

Final Report



A 3D Multi-
Purpose Indoor
Navigation
Application



Marija Kruminaitė
Eva van der Laan
Haicheng Liu
Antigoni Makri
Benny Onrust
Karl van Winden
Weilin Xu

Final Version 28-10-2013

Contents

List of Figures.....	3
List of Tables.....	6
Preface	7
Acknowledgements	8
Abstract.....	9
1. Introduction	10
1.1 Motivation	10
1.2 Aim of the Project	14
1.3 Research Questions	15
1.4 Research Approach.....	16
1.5 Structure of the report.....	17
2. User Requirements	18
2.1 User Types.....	19
2.2 User Requirements in DaRen Project.....	20
3. System Architecture	28
3.1 Conceptual Model	28
3.2 System Components	29
3.3 Functional Components.....	31
4. Database Design and Implementation	34
4.1 Conceptual Model	35
4.2 Implementation.....	36
5. Localization	51
5.1 Localization Techniques.....	52
5.2 Wi-Fi Monitors.....	53
5.3 Fingerprinting Technique	67
5.4 Implementation.....	70
5.5 Wi-Fi Monitored Fingerprinting Method and Comparison.....	75
6. Navigation	81
6.1 Navigation Approaches	81

6.2 Generation of the Network	83
6.3 Routing Algorithms	86
6.4 Implementation.....	88
6.5 Limitations of the generated network.....	92
6.6 Determining the shortest path.....	92
7. Visualization: Graphic User Interface	94
7.1 Graphic User Interface Requirements	94
7.2 Mobile 3D Visualization	98
7.3 Implementation of Graphic User Interface.....	101
8. Verification and validation	106
9. Privacy Issues	108
9.1 Privacy.....	108
9.2 Privacy concerns DaRen.....	110
10. Conclusions and Future Work	112
10.1 Conclusions	112
10.2 Future Work.....	114
10.3 Future Implementation	114
References	118
Appendix A – Localization Component.....	123
Appendix B – Navigation Component	126
Appendix C – Web Service	132
Appendix D – User Requirements.....	135
Appendix E – Database Component.....	137

List of Figures

<i>Figure 1:</i> 2D indoor map without and with route.	11
<i>Figure 2:</i> The navigating information displayed on the smartphone is just the arrow indicating the direction the user should go along.	11
<i>Figure 3:</i> The interface for the indoor navigation application for COEX.....	12
<i>Figure 4:</i> 3D visualization.....	13
<i>Figure 5:</i> Use case of indoor Google map for IKEA.	13
<i>Figure 6:</i> Thematic information as well as the indoor map for The Avenues, a shopping mall.	14
<i>Figure 7:</i> Personal information questionnaire results	22
<i>Figure 8:</i> Users' preferences results.	23
<i>Figure 9:</i> User's mobile phone results.....	24
<i>Figure 10:</i> Conceptual Model.	29
<i>Figure 11:</i> Information in the data model.	35
<i>Figure 12:</i> Data model.	37
<i>Figure 13:</i> The 3D model for the first floor of Hubei Provincial Museum.....	38
<i>Figure 14:</i> The individual indoor objects: a) the walls of room; b) the door of room...	38
<i>Figure 15:</i> Floor plan automatically extracted from the object table.....	41
<i>Figure 16:</i> Floor plan of the museum created manually.	42
<i>Figure 17:</i> a) geometries stored in the footprint field, b) 3D geometries stored in the shape field.....	43
<i>Figure 18:</i> The subdivision of the original indoor space.	44
<i>Figure 19:</i> Connecting map of tables through ids.	47
<i>Figure 20:</i> Measured signal strengths of test 1.	55
<i>Figure 21:</i> Measured signal strengths of test 2.	55
<i>Figure 22:</i> Measured signal strengths of test 3.	56
<i>Figure 23:</i> Direction of the flow of people in test 4.....	57
<i>Figure 24:</i> Cross-section showing the measured signal strength and distance from the access point in test 5.	57
<i>Figure 25:</i> Measured signal strength from two different mobile devices.	58
<i>Figure 26:</i> Actual and measured locations of mobile device user in test 7.....	59
<i>Figure 27:</i> Second, third and fourth floor of the Hubei Provincial museum with the location of the Wi-Fi monitor (AP) and the measured signal strength of three different mobile devices at the location of the cross.	61
<i>Figure 28:</i> Correlation between distance (x) and RSSI quality (y) in test 1.	62
<i>Figure 29:</i> Correlation between distance(x) and RSSI quality(y) in test 2.	63

<i>Figure 30: Correlation between distance(x) and RSSI quality(y) in test 3.</i>	63
<i>Figure 31: Test 2 (red) versus test 5 (blue).</i>	64
<i>Figure 32: Samsung Galaxy S (red) versus HTC Desire 6 (blue) in test 6.</i>	65
<i>Figure 33: Mobile device user in the middle of the room, Mobile device user at the edge of the room.</i>	69
<i>Figure 34: Testing Wuhan fingerprinting application in an open space.</i>	70
<i>Figure 35: Hubei Provincial Museum's floor plan with reference points.</i>	71
<i>Figure 36: Histograms of RSS without human presence (Chan and Sohn, 2012).</i>	72
<i>Figure 37: Histograms of RSS with human presence (Chan and Sohn, 2012).</i>	73
<i>Figure 38: Testing the functionality of Wuhan fingerprinting application in the museum.</i>	73
<i>Figure 39: Radio Map of a Wi-Fi monitor (purple dot) in Hubei Provincial Museum.</i>	76
<i>Figure 40: Probability maps of both Wi-Fi monitors.</i>	77
<i>Figure 41: Final probability map of two Wi-Fi monitors containing the calculated location (dark-blue dot) and actual location (light-blue dot).</i>	77
<i>Figure 42: Final probability map with RSSI 13 and 21 returns more than one location.</i>	78
<i>Figure 43: Scatter Plot to derive the relationship between the RSSI measured by the Wi-Fi monitors and the mobile device</i>	79
<i>Figure 44: Cell centres and paths overlaid with a floor Plan (Lorenz et al., 2006)</i>	82
<i>Figure 45: Regular grid (Afyouni et al., 2012).</i>	82
<i>Figure 46: Irregular grid (Afyouni et al., 2012).</i>	83
<i>Figure 47: Network created using Medial Axis Transformation, adapted from Karas et al. (2006).</i>	84
<i>Figure 48: Left: 2D plan. Right: Topological data construction (dash line – adjacency, black line – connectivity) (Choi and Lee, 2009).</i>	85
<i>Figure 49: 3D Geometric Network from 3D Topological Data Structure (Meijers, Zlatanova and Pfeifer).</i>	85
<i>Figure 50: Path with obstacles computation using Dijkstra's Algorithm (available at: http://theory.stanford.edu).</i>	87
<i>Figure 51: Path with obstacles computation using A* Algorithm (available at: http://theory.stanford.edu).</i>	87
<i>Figure 52: Spatial units: rooms, corridors, stairs, escalators and doors, represented as polygons on the first floor of the museum.</i>	88
<i>Figure 53: Vertical link connecting two floors (Spassov, 2007).</i>	89
<i>Figure 54: Left: room triangulation. Right: centre points of triangles.</i>	90
<i>Figure 55: Connection of two individual networks of rooms.</i>	90

<i>Figure 56:</i> Exhibits in the navigation network.....	91
<i>Figure 57:</i> The derived route to the exhibit.	92
<i>Figure 58:</i> The result of path computation.....	93
<i>Figure 59:</i> 3D model of Hubei Provincial Museum.	96
<i>Figure 60:</i> Inside of the Hubei Provincial museum.	96
<i>Figure 61:</i> Left: bird-eye view and right: pedestrian view (<i>Zhai Shumin, et al, 1999</i>) .	97
<i>Figure 62:</i> GUI state map.....	101
<i>Figure 63:</i> Left: main menu. Centre: popular exhibitions. Right: map view.....	102
<i>Figure 64:</i> Implementation of map view.....	103
<i>Figure 65:</i> 3D scene in Unity3D.	103
<i>Figure 66:</i> Visualization of path.	104
<i>Figure 67:</i> DaRen application.	107
<i>Figure 68:</i> Wayfinding problems for passengers in a transportation hub.....	115

List of Tables

<i>Table 1:</i> Key user requirements of DaRen application.	26
<i>Table 2:</i> System architecture.	33
<i>Table 3:</i> Fields in the object table.	38
<i>Table 4:</i> Fields in the indoorspace table.	42
<i>Table 5:</i> Fields in the subspace table.	43
<i>Table 6:</i> Fields in the locations table.	44
<i>Table 7:</i> Fields in the objectinfo table.	45
<i>Table 8:</i> Fields in the networknodes table.	46
<i>Table 9:</i> Fields in the networkedges table.	47
<i>Table 10:</i> Comparison of different approaches, adapted from Afyouni et al. (2012). ...	83
<i>Table 11:</i> Routing techniques (<i>Zlatanova, 2013</i>).	86
<i>Table 12:</i> The test of loading model on a mobile device.	97
<i>Table 13:</i> Comparison of 3D Engines.	100

Preface

Synthesis Project is a part of the MSc Geomatics programme at Delft University of Technology. It combines most of the subjects taught in this programme and allows applying the gained knowledge to a real-world project while working in a team. Synthesis Project aims to experience the entire process of project management from project definition, data gathering, processing and analysis to the final application.

This year Synthesis Project is carried out in cooperation with Wuhan University (China) and focuses on demonstrating the potential of 3D indoor navigation systems and raising awareness of the possibilities provided by the technology for every-day users. The final product of the project is a 3D indoor navigation system for Hubei Provincial Museum (China).

Synthesis project lasts seven weeks and is subdivided into three development phases: preparation, implementation and synthesis. Project is concluded with the Symposium.

Acknowledgements

Our team would like to thank our supervisors Ir. Edward Verbree and Dr. Dipl. Ing. Sisi Zlatanova for guidance, feedback and enthusiastic encouragement during the entire Synthesis Project. We would like to express special gratitude and thanks to the community of Wuhan University: Prof. Qing Zhu, Prof. Xinyan, Assoc. Prof. Haojun Ai, Assoc. Prof. Kehua Su, Ph.D. Qing Xiong, Ph.D. Xiao Xie, Ph.D. Weiping Xu, MSc students Taizhou Li and Tao Hu for their kind cooperation and their valuable technical support on this project as the project could not be executed without their help. We are grateful for the gracious hospitality toward us in our three weeks visitation; the visit is an invaluable and memorable experience. We thank TU Delft, Joint Research Centre between TU Delft and Wuhan University, Delft Infrastructure and Mobility Initiative, Bureau BrowserBeest and Kadaster for their trust in a success of the project and their financial support. Last but not least we would like to thank all others who contributed to this project.

Abstract

Built environment is an important factor in people's daily life as they spend most of their time indoors. It is extremely important for people to find their destination in large complex buildings, such as malls, airports and museums. 3D capabilities are becoming highly demanded in most visualization-related applications. So far most navigation systems use 2D maps to represent the environment. However, these maps do not provide adequate information about object features, such as colours, textures and shapes. A 3D environment provides users with more realistic spatial information. 3D indoor navigation offers more possibilities to represent dynamic changes such as closings of buildings, inaccessible areas or areas that are affected by moving obstacles. Therefore, the goal of this study is to show the capabilities of the 3D indoor navigation system by creating DaRen 3D indoor navigation application for the Hubei Provincial Museum.

As mentioned in a number of previous studies to be able to build 3D indoor navigation application several components are needed: localization method, a digital model, algorithms for path finding and a guidance mechanism. In this study, additionally to these components requirements of museum visitors and managers are analysed. Besides, privacy issues were considered. In the background research possible implementation options for the 3D indoor navigation application were identified and the most appropriate solutions were selected. To locate the user accurately Wi-Fi based fingerprinting technique was selected. To implement the navigation the network approach was used. Using this approach the indoor objects are represented as nodes and edges were linked together according to their connectivity and adjacency. The visualization of 3D model is implemented using Unity3D game engine and location of the user and path information are rendered on 3D model.

Finally, all different components were integrated with each other and application for Android platform was developed. The application has three-tier architecture: client, web server and database.

The created application was successfully tested in the Hubei Provincial Museum: user was accurately located and guided to the point of interest. The research shows the high potential of indoor navigation systems.

1. Introduction

1.1 Motivation

Location based services (LBS) have been appearing as one of the most fast-growing research fields in the last decades. LBS integrate geographic location into services to provide added value for the user (*Gressmann, Klimek, Turau, 2010*). These services are extensively used in outdoor navigation systems. The success of these systems indicates the need of users for navigation and location based information. Nowadays, the ability to navigate people in indoor environments has become increasingly important for a large number of applications.

Built environment is an important factor in people's daily life as they spend most of their time indoors. It is extremely important for them to find their destination in large complex buildings, such as malls or hospitals. Indoor positioning and navigation is not a new issue. Actually, in the past few years it has emerged to one of the most popular and evolving applications of wireless telecommunications, with a great variety of commercial applications in the sectors of industry, health, safety, but also in a lot of other sectors of human activities (*Papaspyrou, 2012*). An important factor for the development of positioning and navigation systems with increased usability and functionality was the technological advances of the last decades, such as GPS, cellular communications and Wi-Fi, Internet enabled mobile devices and innovative operating systems such as Android and networked environments. So far, there is not a single low-cost scalable mobile phone solution available in the market for successfully navigating the user from one position to another in an indoor environment. However, there is a growing interest on the development of indoor navigation systems which is reflected in the increase of the number of researches and implementations in both companies and universities.

In 2009, Le designed 2D indoor navigation system on an Android mobile platform. The system accomplishes three tasks, namely localization, navigation and user interaction. For the localization, two techniques are combined. Location fingerprinting approach is used to determine user's position based on the received signal strengths of wireless access points inside the building. Then an inertial navigation system is adopted to detect changes of user's position by measuring the acceleration and orientation of the device in three dimensions. And the accuracy is reported to be less than 2 meters. As far as navigation component is concerned, the 2D topologic network, in which rooms and hallways are represented by graphic nodes and edges, was initially built and later on

Dijkstra's algorithm was implemented to calculate the shortest route. Figure 1 depicts screenshots from the user interface showing the map and the route.

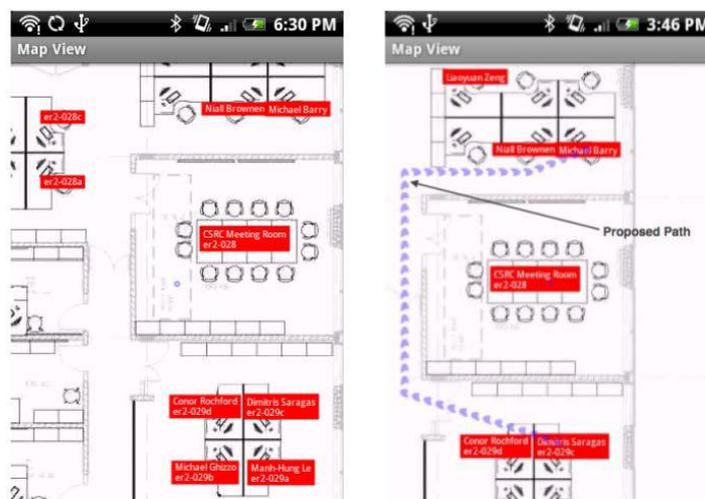


Figure 1: 2D indoor map without and with route.

Chandgadkar constructed another kind of indoor navigation system for smart phones in 2013. It uses the combination of computer vision related techniques and inertial sensors to navigate the users to their destinations. More specifically, the application employs the scanning of custom-made markers to calibrate user's location through the camera. Then a dead reckoning algorithm is used to calculate the user's movements from the last known point. The information acquired is later combined with the obstacle detector to display the correct direction to the user's destination (Figure 2).



Figure 2: The navigating information displayed on the smartphone is just the arrow indicating the direction the user should go along.

Another 2D indoor navigation application was proposed in 2012 by Suh in an indoorGML workshop. In this application for COEX shopping mall, which is illustrated in figure 3, four datasets, i.e. point of interest, topologic network, space data and shape data, were applied. By processing the indoor floor plan which is just the shape data of indoor objects like walls and floor, the navigatable space map can be derived. Then the

topologic network is able to be generated and the optimal route can thus be found. The visualization is supposed to be extended into the 3D space.

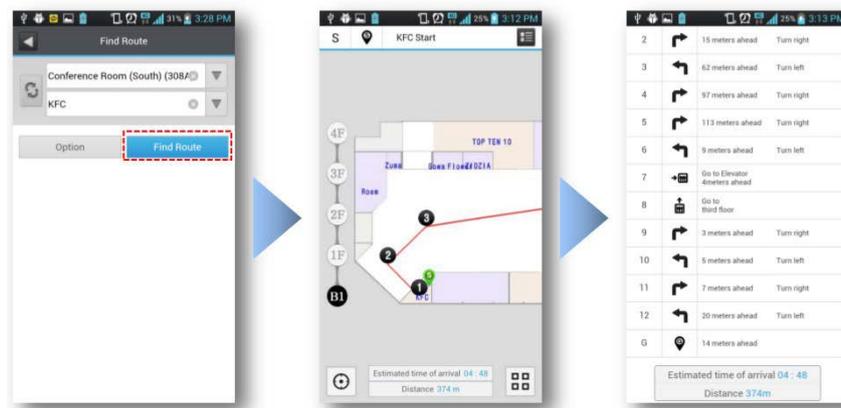


Figure 3: The interface for the indoor navigation application for COEX.

However, so far most outdoor and indoor navigation systems use 2D maps to represent the environment. These maps do not provide adequate information about object features, such as colours, textures and shapes. 3D capabilities are becoming highly demanded in most visualization-related applications. A 3D environment provides users with more realistic spatial information. More specifically for the case of indoor navigation it offers more possibilities to represent dynamic changes such as closings of buildings, inaccessible areas or areas that are affected by moving obstacles.

A prototype of a 3D indoor navigation system without the positioning part was realized in 2008 by Yuan and Zizhang. For this system a BIM (Building Information Model) which includes both the geometric and the semantic information of the indoor 3D model was imported into a 3D GIS system, the 3D model was divided into a predefined navigation-oriented semantic model, the topologic network was derived and the optimal route was computed using Dijkstra's algorithm. The route was displayed in a 3D texture-attached building (Figure 4). Users' profiles and dynamic events such as the close of a certain door had also been considered making the navigation applicable also in cases of emergency.

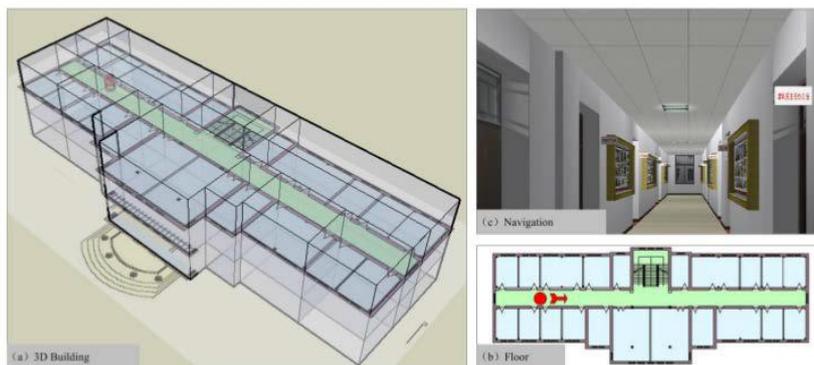


Figure 4: 3D visualization.

In addition to the above mentioned systems which are in a research phase, there are already some commercial systems available for smart phones. Indoor Google maps offer seamless integration of indoor map with outdoor Google map. An indoor map is available if the floor plan is provided. So the navigation is still constructed in the 2D space (Figure 5). Additionally, the route is not displayed, instead the navigating information are the working directions. And the user can switch to different floors inside the app.

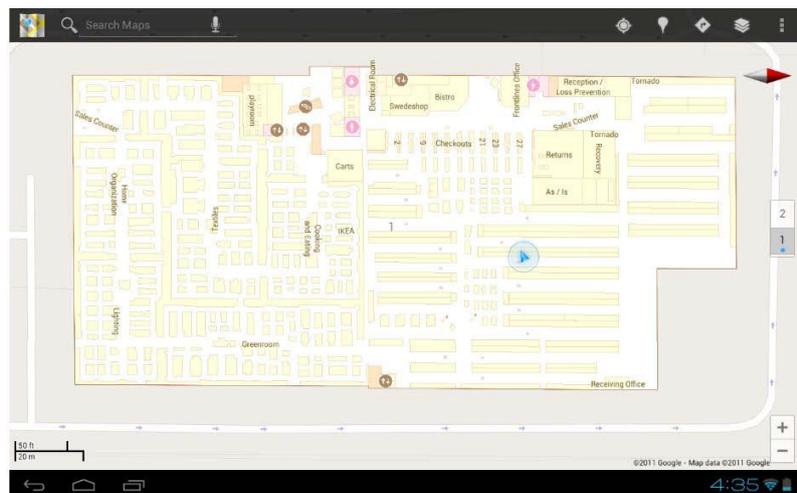


Figure 5: Use case of indoor Google map for IKEA.

Visioglobe is another company dedicated for indoor navigation, which embeds information of services into the indoor map such as real time flight information for airport navigation application or last minute offers for malls. Although the visualization seems to be built in a 3D indoor environment (Figure 6), actually the texture as well as the details of indoor objects like doors are not rendered. So indeed the application is still a 2D navigation system.

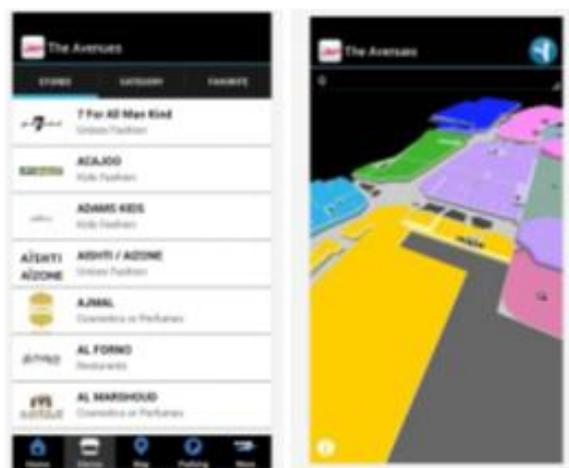


Figure 6: Thematic information as well as the indoor map for The Avenues, a shopping mall.

The presentation of all the previous commercial and non-commercial solutions for indoor navigation systems indicate the need for such kind of systems and the efforts that are undertaken in order to accomplish the aim of a ubiquitous indoor navigation system. Still the proposed solutions rely on the specific application and the implemented technology and methods of localization. More thorough research into the capabilities of indoor navigation systems and especially for 3D systems is necessary.

1.2 Aim of the Project

This project focuses on demonstrating the potential of indoor navigation systems by creating a 3D indoor navigation application for mobile devices. The proposed application is created for the Hubei Provincial Museum in Wuhan. A successful implementation of the proposed system will provide visitors of Hubei Provincial Museum with their current locations displayed in a 3D map of the building on their mobile device and the shortest route plan when searching for an exhibit or an exit. Moreover, additional information on the exhibitions are going to be provided to the user after his request. The research concerns mainly about the Wi-Fi positioning technology, the indoor navigation itself, the 3D model and the 3D visualization on the mobile terminals. All different components i.e. localization, navigation, database storage and 3D visualization should be elaborated, connected to each other and integrated into a mobile application.

In addition to the ultimate goal of the project, which was the development of a 3D indoor navigation application for mobile devices, there were also set other objectives to be achieved while trying to accomplish the main scope.

Objective 1: To create a user interface for the application according to the user preferences and needs.

Objective 2: To add-value to the navigation system by providing context information related to the exhibits on display.

Objective 3: To gain insight on the signal strength scanning by a special kind of Wi-Fi monitors in comparison to the common known fingerprinting purpose.

Objective 4: To explore the most appropriate navigation approach for the Hubei Provincial Museum.

Objective 5: To investigate the possibilities of 3D visualization of the navigation environment.

The objectives had to be achieved within 7 weeks. Since the period is very short some deviations from the initial plan has been decided. As an end product a prototype of only a part of one floor of the museum has been created. The chosen part is the right wing of the second floor, which includes a large exhibition room partitioned in several sections according to the existed exhibits and one side of the corridor that is surrounding the open space connecting all the floors. Moreover, different user profiles were not considered in the design of the interface of the application. The application serves basically the needs of the individual visitor of the museum who is not familiar with the building.

1.3 Research Questions

The main research question that has to be answered in this project is:

Is it possible to develop an indoor navigation application which provides the real time location of the user based on Wi-Fi technology and the route to a final destination in a 3D model visualized on the mobile phone screen?

Other important research questions are:

- Is the final application in accordance to the user requirements?
- How are the needs of museum managers served?
- What kind of system architecture is most appropriate for the application?
- How all data are going to be stored? Locally or on the server?
- Which Wi-Fi based technique is more suitable in order to provide accurately the room in which the user is located?

- Which navigation approach is the most appropriate for implementation of navigation services in the Hubei Provincial Museum?
- How geometry and topology are linked? Where the information about exhibit is stored: by the geometry model or by topology?
- How the 3D model is going to be visualized on the mobile phone screen?

1.4 Research Approach

The project was carried out in collaboration with Wuhan University. The mobile application was created for the Hubei Provincial Museum. The Wi-Fi indoor positioning technology has been selected for the project because of the popularity of the Wi-Fi enabled mobile devices and the availability of existing infrastructures in most indoor environments and among them the Hubei Provincial Museum. In addition to the positioning subsystem, a navigation system and a database were implemented. The navigation subsystem implements a routing algorithm to guide the user to his final destination. The database stores the 3D model, location information, navigation nodes and context information of the museum. The proposed application was developed for Android Operating System. Android is well suited for modern devices and offers an excellent option for localization due to its support of various sensors and hardware, and its higher efficiency in comparison with other platforms (*Martin et al., 2010*).

The research was carried out in three main phases. In the first phase the basic conditions of the research were formulated, the literature study on localization techniques, navigation algorithms and visualization approaches was completed and the project plan was defined. The second phase was the actual implementation phase. It includes all the actions that led to the creation of the mobile application. In this important phase the user requirements were specified, the information about museum exhibits were collected, the determination of the user's location was achieved, a topological model of the museum was obtained, the shortest path between two points was calculated and the route was displayed on the 3D model of the museum on a mobile device. The database in which all data were stored was connected to an external server and a web service was created on the server through which the application was connected to the database to retrieve the data. This phase was completed by testing the application in the museum in order to validate the proposed system architecture. Finally, in the third phase the research on localization techniques was completed and data analysis was verified, the results were validated and the conclusions and recommendations were generated. The culmination of the third phase and the whole Synthesis Project is this report in which the purpose,

objectives, processes and outcomes of the Synthesis Project 2013 are presented in detail.

1.5 Structure of the report

The remainder of the report is structured according to the abovementioned components. Chapter 2 elaborates on the user needs presenting what is required from such system from the user's point of view. The next chapter discusses the system architecture providing detailed information on how the different components of the proposed application interact with each other. The database component is analysed in chapter 4. In this part the selected database is introduced and the way in which all information from the different components is stored in this is thoroughly explained. Chapter 5 contains detailed information about the localization system. The two Wi-Fi based techniques that have been considered and combined are analysing, the results coming out from the tests of the Wi-Fi monitors and fingerprinting application in Architecture and Liesmars building as well as the Hubei Provincial Museum are provided and the implementation phase is described. In chapter 6 navigation options are explained and the implementation of the chosen navigation algorithm in order to provide a route is presented. Chapter 7 provides detailed information on the implemented Graphical User Interface (GUI) and 3D visualization. In chapter 8 the results that came out of the implementation of the application are presented and the system is validated. Privacy issues related to the system are discussed in chapter 9. Finally, the conclusions, future work and possible implementation of the application are provided.

2. User Requirements

User needs and wishes is one of the most important components when designing a new product or system or service on a mobile device. The user requirements for specific applications provide the direction of the research. A mismatch between the users' expectations and the perception of designers on what is important / required can lead to the failure of the product. Therefore, the first consideration when developing a new application is the intended user and what that application should be able to provide to him.

User requirements of a mobile application can be classified into two groups; the requirements of functionality and usability of the application. Functionality refers to the extent the system operates in the way it is structured and is expected to perform while usability refers to the extent a system facilitates users; enhance them utilize the offered functions easily and appropriately (*Calisir et al., 2011*). A balance between those two factors is requested in a user-centered design. In such a design, user behaviour studies must be carried out and the users' environment must be taken into consideration.

According to the abovementioned, for DaRen project the first step in the development of the mobile application for the Hubei Provincial Museum is the conduction of a user requirements analysis. The potential users of the application and their needs and wishes from such an application should be identified. The requirement specification is the main input for the user interface specification and implementation process. The design of the user interface relies on the requirement specification to provide a thorough list of the desired application functionalities and other qualities. The user interface designer refines the requirements into a more detailed form that specifies how the application solves the consumer need.

The conclusions of the user requirements analysis are based on monitoring of peoples' behaviour in the museum's environment, a questionnaire based survey that was conducted in the museum area and in which survey around 130 visitors of the Hubei Provincial Museum answered some questions about their needs from a mobile navigation application in the museum and an "expert interview". Expert interview in the case of the museum is a semi-structured interview with museum employees aimed at understanding the management concept in the museum, the display of information and the planning of guided tour system and visitors' service and mainly at exploring in depth the information needs of the museum visitors.

2.1 User Types

The first step in specifying the user requirements is the determination of the various stakeholders involved in the process. In this project in particular, the proposed application is mainly in the field/range of interest of the users that directly interact with it, which are the visitors of the museum but also the museum employees. Additional to that the museum director and managers needs have to be considered as they decide on the implementation of the application.

Museum visitors can be classified according to their interests and backgrounds or even according to their mobile devices. They can be children or adults, foreigners or domestic visitors, individuals that visit alone, regular visitors, researchers, professionals and groups of people. Each one of the abovementioned groups has different demands regarding the type of information they are interested in. Moreover, the level of details that is requested and the way information will be displayed on the mobile device can be differentiated according to the target group. For example, the information enquiries from regular visitors are mainly concerned about what the museum has to offer on the day of their visit, the items on display and navigation around the museum; while educational visitors require detailed information about the Museum's collections and access to a wide range of information resources both in the museum and elsewhere (*Lin, 2007*). Nevertheless, it can be stated that the main requirements of visitors are related to the provision of information regarding the context of the museum (exhibitions, artefacts on display) based on their locations. This is not the case for museum employees as they are familiar with the exhibitions and the building configuration. They are mostly interested in finding out their way to restricted areas for visitors.

The information presented to the user can be based on the real time location of the user but it is also possible user profiles to be considered. The characteristics of the user type as well as their preferences about the way of presenting this information can be taken into consideration in designing the different tour routes; thereby providing an adaptive system. An important design point of such a system is that it does not eliminate user choices but rather order them based on the stereotypical user model. In this case information related to the users' preferences are used to propose the most interesting parts for the specific user. Knowledge on the previous location of the user can be also used to provide better information (related to their interest). Moreover, for regular visitors the locations of previous visits can be stored and conclusions about their preferences can be extracted.

However, there are many issues and challenges arising from the real-world environment implementation of a mobile application that considers various user types requirements. First of all the application interface should be able to cover the needs of all different type of users. However, this renders the design of an interface a complicated task. Another limitation is that often the user's position alone is not enough to indicate interest in the closest work of art. Thus, the risk is that the system erroneously identifies the user interests and determines the corresponding user model.

In this project, due to the short time span specific user requirements for different groups of users were not considered. A generic user type has been considered for the design of the application based on the information collected from the questionnaires and the interview and also of the monitoring of visitors behaviour. That generic user type is the average individual museum visitor who has no professional interest and is not visiting regularly the museum. Thus, the user model that is used refers to museum visitors not familiar with the place, with increased needs for detailed and precise routing services and for context information.

2.2 User Requirements in DaRen Project

As it has already been stated the goal of this design is to build an application for a mobile device which is able to locate the users and provide them with the path to a desired location within the Hubei Provincial Museum. This should be possible using only the features available on the device. In this case the user requirement analysis is twofold. Firstly, the information requirements of a museum visitor should be investigated and later his requirements of the design of the mobile application should be considered. The main questions that have to be answered in every case are:

- What visitors need to know when they visit the museum?
- What are their basic needs when they use an indoor navigation mobile application?

The final user requirements for this project can be specified by combining the results that came out of the literature research focused on these topics with the in situ monitoring of people's behaviour and the outcome of the questionnaire-based survey.

2.2.1 Needs of the Museum Visitors

Nowadays, it is becoming extremely important for the museum visitors to be able to be navigated in the environment being aware of the context of it. Every industry in the society is trying to investigate and research into the information requested by the public

to understand what people actually need. Museum curators have increasingly been concerned with putting the needs of visitors first, ensuring that museum information resources are available to all users. The information needs can be divided into visit information and service information (*Lin, 2007*). Visit information concerns the functions of an exhibition such as exhibit theme, exhibit content, exhibit time and display location. Additionally, visit information provides all relevant details for the main exhibitions. Service information includes the services provided in the museum for the convenience of the visitors during their visit such as checkrooms, dressing rooms and gift shops. All of this information is important to the visitors. Furthermore, applications for smartphones and tablets for offering museum information resources to visitors are becoming more and more popular and visitors demand them. The evolution of position tracking technologies and location-based services offers new possibilities for indoor and outdoor context-aware navigation services (*Zafeiropoulos et al., 2004*). Through a context-aware navigation system the visitors in a museum can retrieve location-based content immediately in their mobile devices in order to enhance visiting experience. The context information may concern the different exhibits in a certain floor or the most important artefacts of the museum or information about the scheduled events.

2.2.2 Design Requirements for the Mobile Indoor Navigation Application

When developing a mobile indoor navigation application there are certain design criteria aiming at the fulfilment of basic needs important for all different groups, such as:

- accuracy of location determination
- robustness of the application, which refers to the level of consistency in providing reliable and accurate positioning values in areas where signal strength and/or data acquisition may be compromised
- easy navigation in the indoor environment
- orientation support in the surrounding environment
- user-friendly interface

2.2.3 Questionnaires-based survey

The survey consists of questions regarding three type of information, i.e. personal information, user needs of navigation systems and information about user's mobile devices. The questionnaire is cited at Appendix D. The personal information part aimed at gaining insight on the visitor's profile. The questions of the second information

category investigate the user requirements of a museum navigation system, focusing especially on the needs regarding the functionality of the user interface. Finally, the third type of questions aimed at finding out the most popular mobile operating system. Both paper questionnaires and online questionnaires (132 in total) were distributed. Among them, 85.61 per cent are from visitors of the museum and the other 14.39 per cent are from internet.

The statistics of personal information are shown in Figure 7. It is notable that most people investigated are aged between 21 and 25, and 70 per cent of participants are younger than 30. The number of male participants is a little greater than female participants. Since the navigation system is developed for a mobile platform, the survey results of Chinese mobile user in 2012 (*IiMedia, 2013*) were also studied to discover whether the sample of survey participants is representative. The abovementioned survey demonstrates that 64 per cent of the smartphone owners in China are aged between 10 - 29 and 56 per cent of them are male. In that way it is proved that the participants in the survey can represent the possible users of the proposed navigation system.

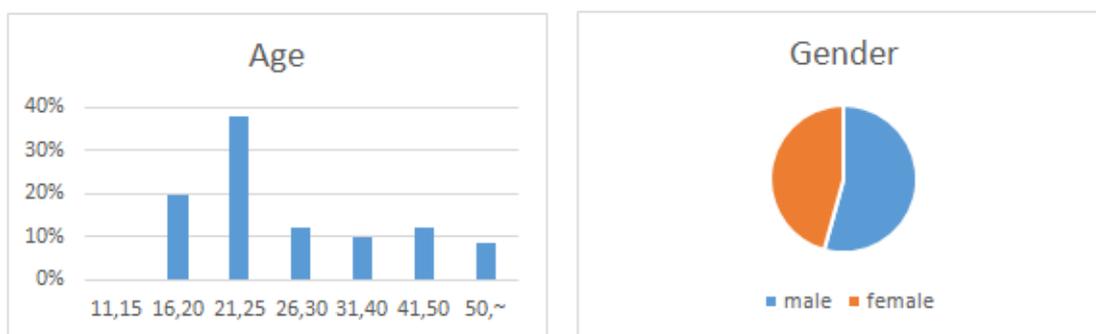


Figure 7: Personal information questionnaire results

Figure 8 depicts the results of the users' needs questions. Based on the statistics, the following conclusions can be drawn. Firstly, bird's eye view of 3d model is the preferred navigation environment since it is most welcomed by users. Secondly the system should allow users to find their destinations dynamically which means the path is computed based on user's current location and the path can be updated when user change the destination. And it is better to provide both "type" and "select" methods to entry destination. In addition, the exhibition information is necessary. The voice guide and compass function should be included. Finally, bilingual interface (Chinese and English) is optimal.

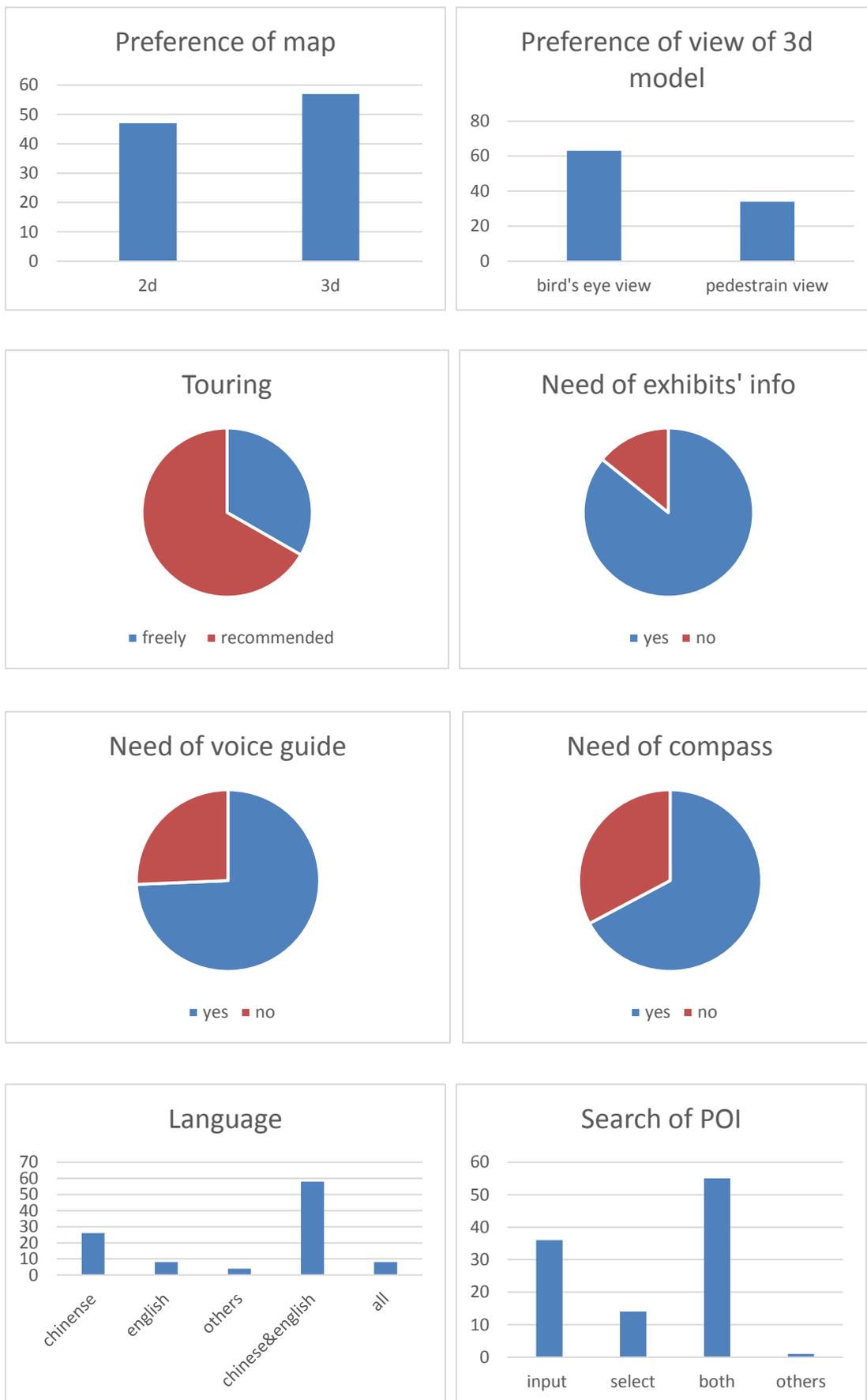


Figure 8: Users' preferences results.

The last part of questionnaire looks at the mobile phone information. Figure 9 shows the statistic of mobile brand and operate system. Samsung phone is proved to be most common used and Android system beat all other operate system becoming the most popular OS. When looking at another survey of Chinese mobile user carried out in 2012 (Wang, 2012), similar conclusion can be made. The top 3 mobile brand in China are Nokia (21.4 per cent), Samsung (14.2 per cent) and HTC (12.7 per cent). Android system has much more user than others with 58.2 per cent of total users. Therefore the Android platform should be the first choice for the navigation system.

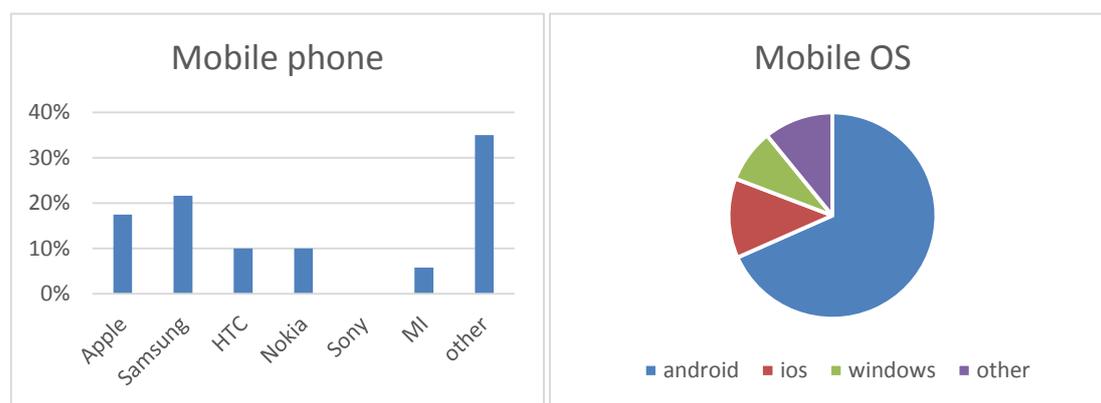


Figure 9: User's mobile phone results.

2.2.4 Specific User Requirements in DaRen Project

The main condition that museum visitors require from an indoor navigation mobile application is the accuracy of the location determination. The accuracy refers to the error between the system's approximated location with reference to the true value of user location at that given time. The accuracy can be significantly differentiated according to the particular application. For DaRen application the required accuracy is refers to room level. The exact position determination is definitely preferable but at least the room in which the visitor is located is a prerequisite in order to have a starting point for the navigation. For this application also the response time is important for the users. There is a need for real-time location information.

Moreover, one of the main requests of Hubei Provincial museum visitors is the need on additional information about the context of the museum. Therefore, the combination of localization technique with location-awareness service to provide the surrounding information for the user has been implemented. The information that is provided to the museum visitors is related to:

- information about the museum's exhibitions and description of the main exhibits

- the most important artefacts of the museum in case that the visitor has not enough time to tour around the whole museum
- location information of points of interest such as museum shop, restrooms, information desk
- the most aesthetic routes
- information about the scheduled performances and their timetable
- parking and transportation information
- opening hours

That information is available to the users based on their current location. Other type of information is important for people when entering the museum and other information for people entering an exhibition room. The users need not only to be able to search for the exhibit of their interest but also important objects to be proposed to them by the application.

The user model that is considered has no or limited information on the indoor environment. Based on that fact there are increased needs on clear and explicitly detailed representation of the route to the final destination. Furthermore, in the museum environment the locations of the exhibits have to be considered in such a way that the navigation route includes them. The output format is also extremely important for the user. According to the results coming out from the survey museum visitors are interested in experimenting with 3D model as the mean of displaying their location and the navigation routes. Apart from the visual representation of the information on their screen they are also extremely interested in voice guidance.

Users should be able to be efficient when using the DaRen application. In order to do so they need an interface that allows them to operate the application without experiencing problems and difficulties. The provided options should be well-stated and the icons should be straightforward in their meaning in order not to allow misinterpretations. Furthermore, an overview of how the application works and a comprehensive help function should be provided in order to improve the usefulness of the system for novice users.

Finally, users have exacting expectations of any system that involves their location. It is essential to provide the user with confidence in the design and implementation of the system. Security requirements include the need to protect authorization information from unauthorized access, the maintenance of customer confidentiality and tracking of location in progress.

The following table summarizes the main results from the user requirements analysis for the proposed application:

Parameter	Requirements
Accuracy of localization	<ul style="list-style-type: none"> • Room level • Real-time
Easy navigation	<ul style="list-style-type: none"> • Clear and detailed route • Exhibits to be considered
Information Output	<ul style="list-style-type: none"> • 3D map • Voice guidance
Context information	<ul style="list-style-type: none"> • Based on current location • To be proposed to them
User Interface	<ul style="list-style-type: none"> • Ease of use • Overview of how the application works

Table 1: Key user requirements of DaRen application.

Finally, it is necessary to refer briefly to the museum managers requirements which are extremely important for making feasible the implementation of the system. The application should be beneficial for the museum. One of the major desires of the museum directors is the reduction of the costs. This can be achieved by using visitors' mobile phones. In that way a museum decreases the resources necessary to implement and maintain the solution, as in any other case, the museum would have to provide handheld devices, such as headphones, PDAs, palmtops. The museum would also minimize maintenance expenses. Moreover, the acquisition of information related to the number of visitors, the most visited exhibitions and the visitors' behaviour are of great importance as it can contribute to the improvement of the management of its resources. So far the monitoring of visitors behaviour is realized through museum cameras, the presence of security guards in all exhibition rooms and the visitors book. These methods are not adequate for the extraction of proper information as the museum employees have been admitted. A more comprehensive way for gaining insight on visitors' behaviour is necessary.

The main benefits of such an application for the museum are:

- attraction of more visitors as it enhances the ability to be navigated in the museum and provides information about exhibits
- reduction of costs for equipment such as headphones with recorded instructions
- reduction of the number of visitors that are addressed to the information desk for information about exhibitions or POI

- obtain information related to the number of visitors and artefacts of interest.

All the abovementioned parameters had to be considered during all the phases of development of the proposed system. The success of the system is based on its ability to fulfil these requirements. Due to this fact all the decisions of all the components of the system were made taking into account the results of the user requirements analysis.

3. System Architecture

Different components of a system and interaction of these components are defined in the system architecture. System architecture explains reasons for certain decisions that are made to create the DaRen application taking user requirements into account. Essential elements such as software languages, Database management systems (DBMS) and operating systems (OS) are explained in detail.

3.1 Conceptual Model

The system architecture can be divided into different components which interact with each other. In the DaRen architecture two types of components are distinguished: functional and system components. Functional components perform specific functions while system components support functional components to carry out these tasks.

Based on the key user requirements stated in Table 1, three functional components are identified for the DaRen application: localization, navigation and 3D model visualization. Localization component is extracted from the "Accuracy of localization" parameter, navigation on the "Easy navigation" parameter, and 3D model visualization on "Information Output" and "Context information". The following descriptions are given to these components:

- **Localization component** calculates the location of the mobile device carried by the user in the Hubei Provincial Museum.
- **Navigation component** contains two functions: network extraction from the floor plan of the museum to obtain a topological model and path computation from point A to B based on the built topological model.
- **3D model visualization component** is responsible for visualization of the 3D model of the museum and calculated route on a mobile device.

In the DaRen system architecture the system components are database, web service and application. These components are selected based on the combination of the requirements of functional components and requirements of the museum manager for the application.

The conceptual model of the DaRen system is provided in Figure 10 and gives an overview of the different components and their interaction. In the model it is made visible which system components are used by the functional components. Additionally,

the workflow of the system is indicated by numbers on arrows between each component. All components are described in detail below.

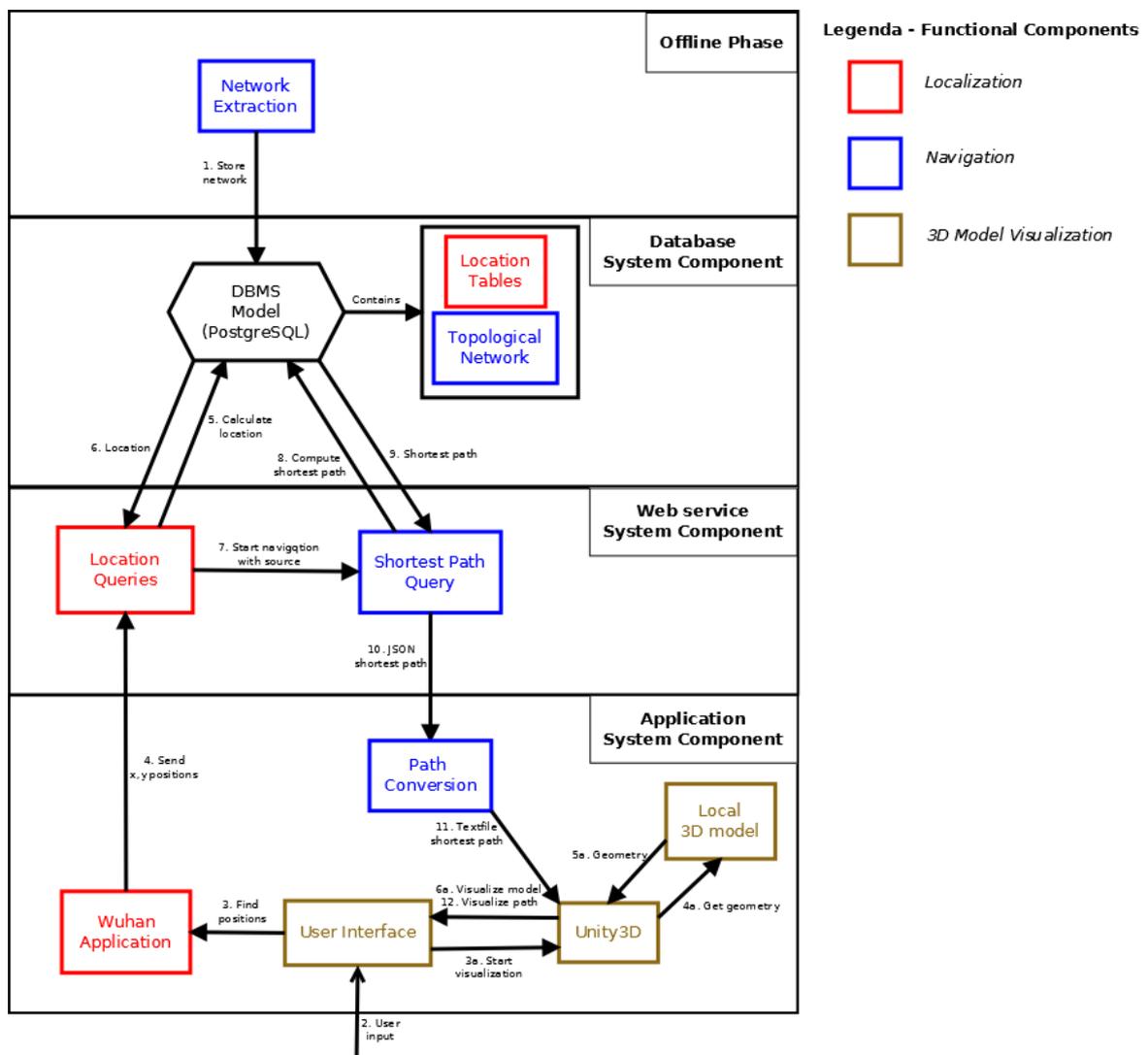


Figure 10: Conceptual Model.

3.2 System Components

As it was mentioned above system components are very important and can be identified as a backbone of the system since they support the functional components.

Application is required to be able to guide the visitor of the museum while showing the navigation path on a mobile device. This functionality is established by creating a mobile Android application. The choice of using mobile devices is based on the museum manager requirements. The requirements express that the implementation and maintenance costs of the system should be as low as possible. Thus use of mobile devices can reduce the costs as museum does not have to provide hardware to the

visitors. A mobile application is implemented instead of a mobile website since mobile applications provide interactive environment and offers personalization. Further, Android OS is used to implement the mobile application due to its popularity: worldwide 60 percent of the mobile devices use Android OS and even 80 percent in China (*Arthur, 2012*). Another reason for using Android OS is that most of the members of the project team have Android mobile devices which can be used in testing phases.

Database. In the proposed Android application users should be able to indicate their destination target and receive the route to it from their starting point and the route should be represented using 3D model. Therefore, data required to provide these results should be stored either locally on a mobile device or externally on a database. In this system the data is stored on a database on an external service, because large datasets need to be stored and processed thus mobile devices cannot handle this very well. Moreover, updating data is more efficient using an external database as the data only has to be updated in one place which means that every user can use the most recent data. While, in case of local storage, the user has to install the new updated data himself. Thus, using an external database reduces the costs of system maintenance. Overall, the database can be seen as one of the core components of the system, since it stores all the vital data and it allows calculations on this data. Data of navigation and 3D model components, also information about the museum and its exhibits is stored in this database.

Further, calculations of user's position and/or the shortest path to destination are done in the database by using SQL language of the DBMS. The SQL language depends on the DBMS that is used and in this project PostgreSQL database with PostGIS extension is used. PostgreSQL is chosen because it supports spatial data storage and spatial calculations. Also, there is an extension for PostGIS that supports shortest path calculations in the database based on the stored spatial data. The shortest path calculation is necessary for the navigation to obtain the route between the start point of the user and the destination target.

Web service. To perform the above mentioned calculations on the database, it is necessary that the Android application would give SQL calls to the database. This might be done directly on the database, so that the application makes a direct connection to the database or it can be done through a web service which runs on an external server. A web service is basically a middle layer between a client and a server where the client can ask services to the server. A web service ensures better performance and security of the system, thus it is decided to use web service for DaRen. In DaRen project, the

Android application makes a call to the web service and then the web service makes the appropriate SQL calls to the database. Theoretically it might seem that the use of web service decreases the performance of the system, because there is an extra hub in the connection between the application and the database and this means a longer round-trip time, however in practice the direct connection from a mobile application to a database is very unstable due to low network connectivity of mobile devices. A web service strengthens the stability of a connection between an application and database. In addition, with a web service it is possible to do calculations on the server itself instead of the mobile device, thus the computation power is saved on the mobile devices. The other reason for using a web service is better security of the system. When a direct connection is used, the log-in credentials for the database are stored on each mobile device. These credentials can be extracted by people to get unauthorized access to the database. However, using a web service, the credentials are stored at one place on the external server which makes it much harder for people to obtain these log-in credentials. Therefore, based on performance and security reasons a web service is used in the DaRen system.

The web service is created using open source program language PHP which is the most popular language for creating web services. As a result, it is widely supported by many different software packages. The web service in DaRen receives calls from the Android application and based on the message in these calls the web service chooses the appropriate action and runs SQL statements on the database. The connection and the calls on the database are made by the PDO library in PHP. The PDO library offers a simple way to connect and retrieve data from a database. Later the result obtained by PDO is either used for calculations on the web service or the result is sent to the Android application in JSON format. JSON is a data-interchange format for many different programming languages, thus it allows interchanging data between the PHP and JAVA programming languages. Other advantages of JSON are that it is easy to read and write by humans, and that JSON can be simply parsed and generated by machines. The complete code of the web service is placed in Appendix C.

3.3 Functional Components

As it was previously indicated three functional components are identified for the DaRen application: localization, navigation and 3D model visualization. To implement their functionalities they use several system components.

Localization component operates on all three system components. The localization starts in the application component where several possible positions for the user are calculated. Then, this information is sent to the web service, so that the web service can perform a SQL query on the database to calculate the actual position. The web service receives this result and this is used as input data for the navigation component.

Navigation is the second functional component which operates on web service and database. It receives the user's location from the localization component and the destination target from the application. Based on this input data, start and target points, SQL query is made and is run on the database. The result of the query is the shortest path between these two points. This shortest path is calculated using the navigation network which is stored in the database. The navigation network is extracted from the floor plan of the building in off-line phase. The extraction is achieved by creating a script using the open source programming language Python. Python is selected, because it has many libraries that support spatial types and geometric calculations, and it has well-supported integration with the PostgreSQL DBMS. The created Python script has to be run once and then the network is stored in the database. If the floor plan has been updated, the script is executed again and the information in the database is updated.

3D model visualization is the last functional component and this functionality covers only one the system component – Android application. In the application the 3D model is stored and visualized using Unity3D game engine which has a library that offers support for Android OS. To visualize the path in the 3D model, the 3D visualization component needs a text file with a sequence of points (nodes) that represents the path. This input is delivered by the navigation component. It should be noted that originally it was planned that 3D model of the museum would be stored in the database. In such case the 3D model does not need to be stored locally on each device, but parts of the model are extracted from the database based on the user location and navigation path. Thus, in the original plan 3D visualization component would cover all the system components. However, due to time constraints this could not be completely achieved.

System architecture is summarized in Table 2. In the next chapter the database, one of the system components, is explained in detail.

	Components	Description	Architecture	Functions
System Components	Database	Stores data on external server	PostgreSQL + PostGIS	<ul style="list-style-type: none"> - Stores the navigation network - Provides spatial in-built database calculations
	Web service	Provides connection between application and database	PHP	<ul style="list-style-type: none"> - Uses (spatial) SQL functions on database - Combines SQL results - Combines localization and navigation parts
	Application	Provides interaction with user and visualizes results	Android	<ul style="list-style-type: none"> - Processes user input - Makes calls to web service - Combines navigation and 3D model parts - Visualizes end result on screen by using Unity3D
Functional Components	Localization	Calculates the location of the mobile device	Database, Web service, Application	<ul style="list-style-type: none"> - Measures several time the approximate position - Results are sent to web service to calculate through database true location.
	Navigation	Extract network from floor plan Calculates path between point A and B	Python (network extraction), Database, Web service, Application	<ul style="list-style-type: none"> - Creates navigation network and stores it in the database - Uses output from localization to execute shortest path query - Converts output query in application to text file for visualization
	3D model	Visualize 3D model and path	Unity3D, Application	<ul style="list-style-type: none"> - Visualizes 3D model which is stored on mobile device - Visualizes navigable path by using the output from navigation part

Table 2: System architecture.

4. Database Design and Implementation

It is important to understand how the crucial information which is needed for DaRen 3D indoor navigation application to serve its functionalities: localization, navigation and 3D visualization is stored in the database. In addition, context information of the museum which includes visitor's information and exhibition's information is also stored in the database. To store the data for the DaRen application, a data model which is applicable for the museum indoor navigation system with newly designed classes is constructed and then the data model is implemented in the database.

The new data model is designed because currently most of the 3D models have problems to express spatial information required by indoor navigation (*Kim et al., 2013; Li and Lee, 2010*). For example, in CityGML classes of geometric model is constructed from the lowest level of geometry - point and gradually aggregates to solid which is represented by 3D boundary representation (B-rep). In this way, the topology is implicitly embedded. However, research conducted by Li and Lee (*2010*) shows that for the navigation purposes, storing geometry and topologic network individually is more efficient than topology enabled geometric model. To achieve an efficient structure for indoor navigation, the approach to store geometry and topologic network separately is employed. Thus classes for geometric visualization and topology are designed.

Lately a number of studies on IndoorGML are performed. IndoorGML is a standard which deals with specialties of indoor navigation model. It is aimed at representing and exchanging the spatial information that is required to build and operate indoor navigation systems (*Munkres, 1984*). The data model for DaRen project is designed according to ideas expressed by IndoorGML. In its original data model, geometric classes and topologic classes are treated separately, but the topology derived from the geometries is first realized as induced topology (*Munkres, 1984*) using TP_Solid and TP_Face classes (*Nagel et al., 2010*). After it the topology is mapped in edges and nodes in the dual space. While in the DaRen case, the procedure is simplified and the topology extracted from geometric model is delicately expressed in the dual space.

Another option to store the information of indoor spatial model is XML based formats as many developing libraries support XML file operation. Moreover, most of 3D visualization software are able to render the 3D model stored in XML based formats like GML, COLLADA.

In DaRen project, database is selected mainly due to two reasons. Firstly, the indoor navigation environment is complex; thus system should be efficient to manage data in large volumes. XML based formats usually require more storage space than the actual data needs due to redundant tags and attributes which are not always necessary. Also spatial index plays a crucial role if large amount of spatial records are involved. In this case SDBMS (Spatial Database Management System) such as PostGIS is the most appropriate solution, because it ensures high performance, it does not require lot of storage space and supports spatial indices. In addition, it can provide native geometric data type and plenty spatial analysis functions. All in all, database is considered to be the preferable option for storage of data for indoor navigation.

The main research question regarding the database implementation is:

How to organize and store all the information used in the DaRen navigation application, i.e. geometry, topology and context information of the museum in the database to serve main functionalities provided by DaRen, namely localization, navigation, 3D visualization and context-aware service?

4.1 Data Model

The focus of the DaRen application is navigation, thus the data model should provide efficient structure for spatial queries and route calculations. It has been proved that storing geometry and topology separately decreases the time costs to process spatial queries although it takes more space than the topology-enabled geometric model (*Lee and Lee, 2010*). Therefore, the indoor geometric data and topologic network are also stored individually inside the database. Besides, according to users' requirement, the information about the exhibition and general information about the museum have to be considered and structured in the database. To ease the maintenance, context information should be processed separately. However, all three separate parts need to be linked (Figure 11).

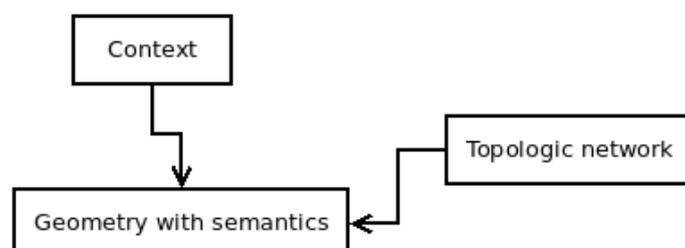


Figure 11: Information in the data model.

Figure 12 shows the UML diagram of the data model for DaRen which consists of nine classes. All classes except “Recroute” which refers to recommended routes and “General_info” are created and stored as tables in the database. Geometry information as well as semantic information is stored in the object table. Records of users' positions are elaborated in the locations table and topology is encoded into “Networknodes” and “Networkedges” tables. Finally, the context information of exhibits is stored in the “Objectinfo” table. Other tables are created to serve localization and navigation.

4.2 Implementation

As it was mentioned above, the PostgreSQL with PostGIS extension is chosen as the platform for DaRen since PostGIS provides native geometric data type and abundant spatial analysis functions. Additionally it supports spatial index for both 2D and 3D geometries. Moreover, an extended module of PostGIS, pgRouting provides functions for path calculation which is necessary for the navigation to obtain the shortest path between the start point of the user and the point of interest. It should be noted that although the DaRen indoor navigation system is expected to be constructed in the 3D space, in fact its main functionalities localization and navigation are based on the 2D floor plans while the 3D model is only used for the visualization. In the UML diagram (Figure 12), it can be observed that apart from 3D geometries, 2D geometries also appear in these classes. As has been mentioned, classes in the data model are realized as tables and in the following parts, they are described respectively.

Object table. The object table is designed to store the geometric and semantic information encoded in the original 3D model. And this information is organized under each individual indoor object such as wall or floor, so each object is stored as a record in this table. 3D scenes on a mobile device are constructed retrieving information from this table. The following SQL command to create the object table is executed and Table 3 describes the fields of the table:

```
CREATE TABLE object (
  objectid bigint NOT NULL,
  spaceid bigint,
  objtype character varying(20),
  geometry geometry(MultiPolygonZ),
  footprint geometry(Polygon),
  navigatable boolean);
ALTER TABLE ONLY object
ADD CONSTRAINT object_pkey PRIMARY KEY (objectid);
CREATE INDEX geomidx ON object USING gist (geometry gist_geometry_ops_nd);
```

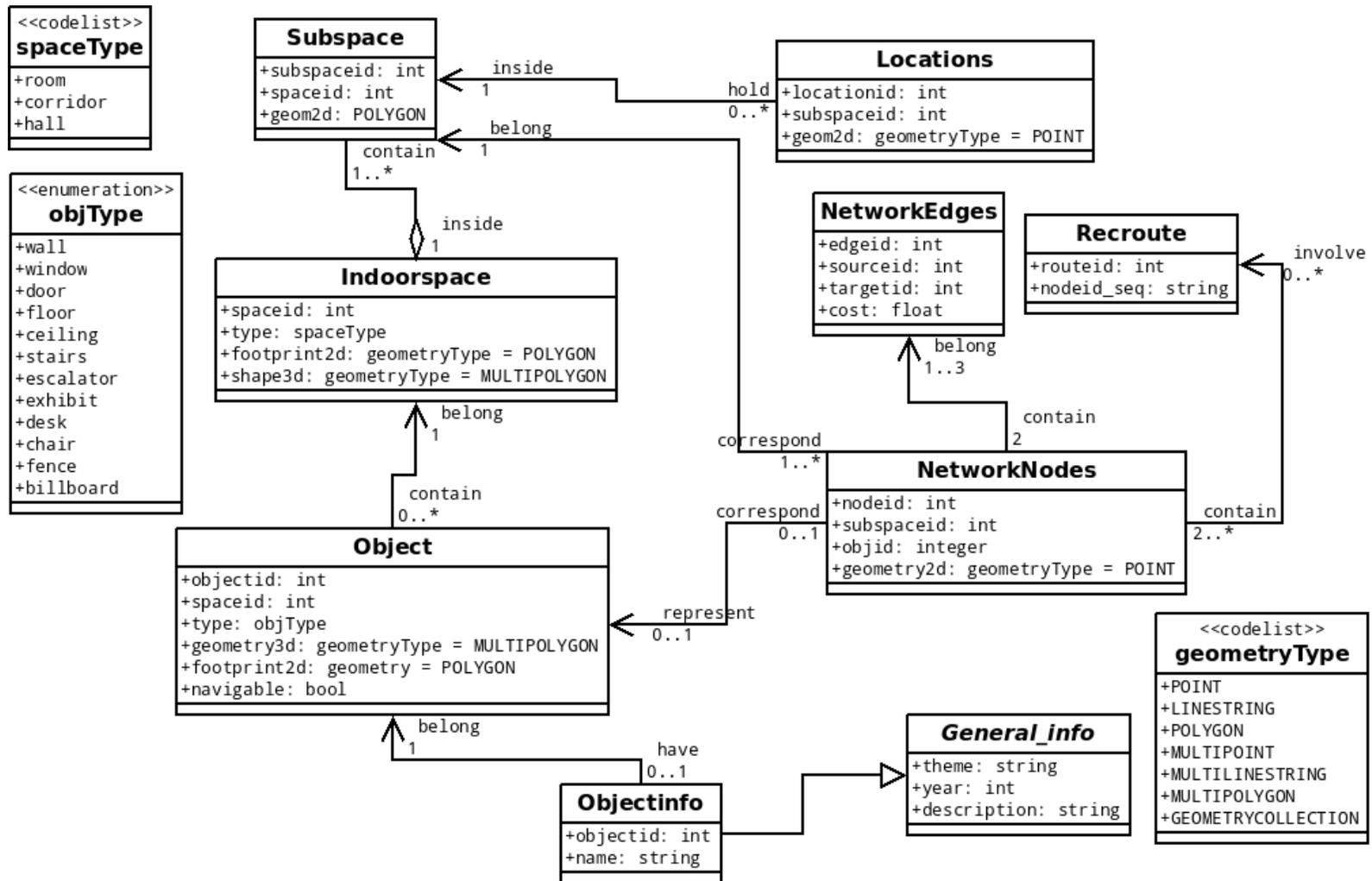


Figure 12: Data model.

Field	Meaning
objectid	A unique integer number identifying each object.
spaceid	Integer number identifying which room, corridor or hall the object is located
objtype	The type name for each object, it can be “door”, “exhibit”, for example.
geometry	3D Geometry in B-rep for the object and it is stored as MultiPolygon in the database.
footprint	2D polygon representing the footprint of 3D object.
navigable	A Boolean value indicating whether the object can let the route pass through, for instance door and window are navigable.

Table 3: Fields in the object table.

The original data source to populate this table is the 3D model of the first floor of Hubei Provincial Museum (Figure 13). The original 3D model is encoded using Wavefront OBJ format. It is an ASCII format and information is grouped with respect to specific features, i.e. objects. Under each object, the shape is expressed using coordinates of vertices (v), vertex normal vectors (vn), coordinates of the texture (vp) and faces constructed by the indices of vertices (v/vt/vn). For each object, the only semantic information is just the name of the object, for example “Room_203_Ceiling_01”. Sample individual objects are shown in Figure 14.

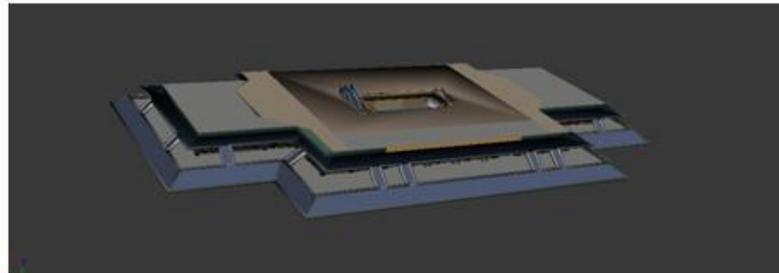


Figure 13: The 3D model for the first floor of Hubei Provincial Museum.

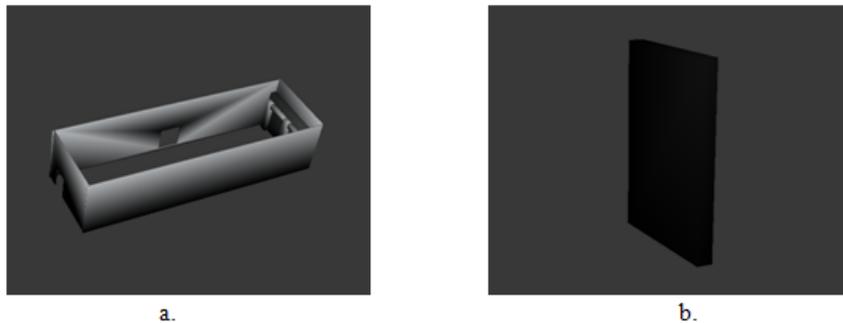


Figure 14: The individual indoor objects: a) the walls of room; b) the door of room.

To import these objects into database, a script in Python was created (Appendix E). In practice, the OBJ file is firstly transformed into COLLADA format and then processed by the Python script since there are limited libraries handling OBJ file. The libraries used are *pycollada* which is capable to parse COLLADA file, *ppygis* which can transform normal arrays of coordinates to

PostGIS geometric data types and *psycopg2* which is a Python interface of PostgreSQL. The door which is illustrated in Figure 14b is used to demonstrate how the importer works.

First, the name of the object and the geometry of it are extracted. Through the name, i.e. “Room_102_Door_01”, the spaceid can be retrieved and the value is 1102. The first number 1 represents the type of the space - room. Currently, three spaces are available, room (1), corridor (2) and hall (3). The second number 1 means the object is located at the first floor while the last two numbers indicate the room number with respect to a certain floor. Here it is assumed that the maximum number of rooms, corridors or halls is 99. In fact, there are only six rooms and one corridor on the first floor of Hubei Provincial Museum and the spaces are divided by walls. The value of objectid can also be acquired and it is 1102xx where 1102 is just the spaceid it belongs to and “xx” means the object number with respect to all the objects existing on the first floor and hence should not be “01”. This is done for the sake of uniqueness because many “01” can exist in a room, for instance, “Room_102_Floor_01”. It should also be noted that xx does not mean that only two digits are available, as long as the whole id is within the constraint of the big int type, it can be as large as possible. It may be argued that since spaceid is stored, why does it still appear in the objectid. This is due to the fact that when later the objectinfo table is constructing, the objectid is used as the only identifier without spaceid. The objtype in this case is just “door” and the navigable is true since the route can pass through the door. The 3D geometry of the door is also extracted. In the original OBJ file, each face of the door is composed by triangles and when the geometry is imported into the database each triangle is treated as a polygon and the final 3D geometry is just a multipolygon, i.e. the combination of these triangular polygons. Footprint is the 2D polygon representing the footprint of the 3D geometry. This field is reserved because the navigation system is actually implemented on the floor plan and by providing the footprints of objects, the related calculation can be simplified.

However, there are several issues which should be addressed. As far as the objtype field is concerned, it currently supports only the twelve types shown in Figure 12 (objType). More types can be added later but for the indoor environment of the museum these twelve types are enough. For DaRen navigation, the exhibits play an important role as not only locations should be determined to guide visitors to them, but also the context information required by users. So exhibits' geometries should be well defined. But in the original 3D model, the exhibits are not expressed, so both the geometric and thematic information are manually collected and depicted. And later the exhibits records are imported individually. In the object table, the wall is also listed as the available type, but actually a wall can be shared by more than one room. So for a specific wall, the spaceid may not be unique. In practice, the semantic of a wall is defined with respect to only one room or corridor in the original model, which means, for some rooms, they are not closed since a part of its walls are attached to other rooms. For those doors which are also shared by two rooms, the same principle is adapted. Besides, so far only the footprints of doors and exhibits are stored in the object table since they are navigable and appear as nodes in the topologic network. And the navigable field is kept for the intension to automatically extract topologic network from the footprints of indoor space and objects stored in the database. As to

the original 3D model, it has textures attached to objects, as it is clearly indicated in Figures 13 and 14. However, since the database mainly performs the data processing task, the textures are not imported into the database. Constructing another “texture” table which is linked to the object table through “objectid” or “objtype” might be a solution. Apart from that, Figure 14a indicates the slight gaps that exist at the corners of walls, this may cause a potential problem for spatial computation later. But now it does not influence the DaRen application since the navigation system is built on the floor plan.

The object table also serves as the data source to construct other tables, for example indoorspace, subspace and topologicnodes because the geometries involved in both tables come from the floor plan. And thus the focus here is how to extract the floor plan automatically from the object table. Since PostGIS provides plenty spatial operating functions, the extraction of floor plan can be achieved inside the database using SQL commands. The algorithm is described below and at the end the SQL commands are provided.

Firstly, the bottom face, i.e. footprint of each indoor object should be extracted. As faces of objects are all organized by triangles whose vertices have 3 dimensions - x, y and z - a height value has to be found as the threshold to filter all the triangles at bottom faces. According to that, if the z values of three vertices of a certain triangle are below the threshold, then the triangle is considered as a part forming the floor plan. The determination of the threshold is a complicated task as it is assumed that all the doors are aligned on the floor so the taken approach is to list the lowest points of all the doors and calculate their average height values. After calculation, the mean is 18.5. In order to avoid the omission of some parts, the threshold is set to 19. Using this value, the bottom triangles of walls, doors and floors can be extracted. After that the z value is removed from all the coordinates and 3D triangles are turned into 2D polygons. During the extraction of bottom triangles, their corresponding “objectids” are recorded, so bottom triangles which share the same “objectid” are then combined into a single polygon.

In the 3D model, all the doors and walls are built on floors, so the footprints of doors and walls are contained in those of floors. Consequently, by removing footprints of walls and doors from floors' footprints, the footprints of indoor space which are just geometries stored in the footprint field in the indoorspace table can be acquired. Then by integrating the footprints of doors, a draft floor plan can be extracted (Figure15).

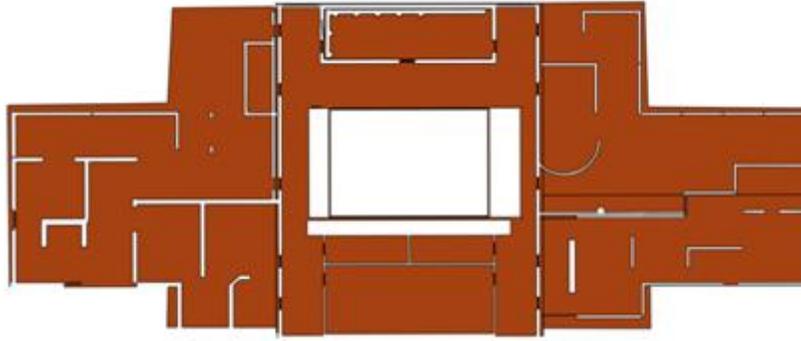


Figure 15: Floor plan automatically extracted from the object table.

It should be noted that in Figure 15, footprints of escalators and stairs are missing compared to the manually created floor plan in Figure 16. In Figure 16 red rectangles indicate exhibits, green polygons represent stairs or escalators and blue polygons indicate doors. The yellow polygons are just the footprint of indoor spaces. This is because escalator and stairs are objects across the entire 3D space and bottom face under the threshold are only a small part of the total footprint. Hence such objects should be treated separately in order to be added into the floor plan. Also the derived floor plan has invalid geometries like spikes but actually by decomposing each polygon and checking the validity, those invalid polygons can be eliminated or rectified inside the database.

The approach is implemented using SQL commands below:

```
CREATE table room as select * from (select objectid, objtype, ST_Force_2D(poly) polygon from (select
objectid, objtype, ST_GeometryN(geometry,n) as poly from object cross join generate_series (1,300) n
where n<=ST_NumGeometries(geometry) and objtype in ('floor','door','wall')) as a where
ST_ZMax(a.poly)<19) as b where ST_IsValid(polygon);
CREATE table floorplan as select objectid, objtype, ST_Difference(geom,out) as geom from (select
objectid, objtype,ST_MemUnion(polygon) as geom from room where objtype='floor' group by
objectid,objtype) as floor, (select ST_Union(polygon) as out from room where objtype in ('wall','door')) as b
union select objectid, objtype,ST_MemUnion(polygon) from room where objtype='door' group by objectid,
objtype;
DROP table room;
```

It should also be noted that such functions, i.e. automatic extraction of floor plan from 3D object model can decrease the work of maintainers since once the 3D model is modified, the floor plan can be easily updated and then the update of relevant tables is coming.

Indoorspace table. The indoor space represents independent rooms divided by walls, doors, floors and ceilings. It is a concept of the space surrounded by boundaries which can be the inner faces of walls, upper faces of floor and lower faces of ceilings. The division of space in this way can be regarded as the realization of the symbolic space concept of IndoorGML. In the table, not only the geometry of each space is stored, but also the semantic information.

SQL command to create the table:

```
CREATE TABLE indoorspace (
  spaceid bigint NOT NULL,
  type character varying,
  footprint geometry(Polygon),
  shape geometry(MultiPolygonZ)
);
ALTER TABLE ONLY indoorspace
  ADD CONSTRAINT indoorspace_pkey PRIMARY KEY (spaceid);
CREATE INDEX spaceidx ON indoorspace USING gist (shape gist_geometry_ops_nd);
```

In Table 4, the meaning of these fields are given.

Field	Meaning
spaceid	A unique integer number identifying each space.
type	The property of the space and currently the value can only be “room”, “corridor” or “hall”
footprint	Footprint of each space and the geometry is 2D polygon
shape	B-rep shape of the indoor space and the geometry is multipolygon

Table 4: Fields in the indoorspace table.

As is was discussed above, the data source to populate this table is the floor plan table extracted from the object table. It should be noted that in the floor plan table, apart from footprints of indoor spaces and doors, the objectid and objtype is also stored. For each indoor space, its objectid is just the objectid of the floor making up that indoor space and objtype is just floor. Then the spaceid which is just the first four numbers in objectid can be derived. And after populating the indoorspace table with all the corresponding information of floor which is stored in the floor plan table, only the shape field is left without value. However, by extruding the footprints of indoor space using the height value, the 3D B-rep geometries can be acquired and used to fill in the shape field.

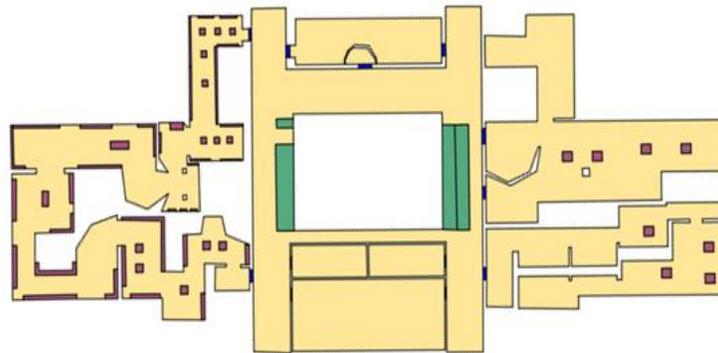


Figure 16: Floor plan of the museum created manually.

However, since the automatically generated floor plan still contains invalid geometries, in practice, a manually created floor plan is adopted (Figure 16). This floor plan is firstly extracted from the 3D model manually in 3ds Max and later is modified in QGIS. The attributes like type and id are also attached to the manually created floor plan. And the indoor space in this case corresponding to type room and corridor. In a similar way, all the data needed by the indoor space table can be populated. Figure 17 shows the extrusion of footprints to 3D shape of indoor spaces.

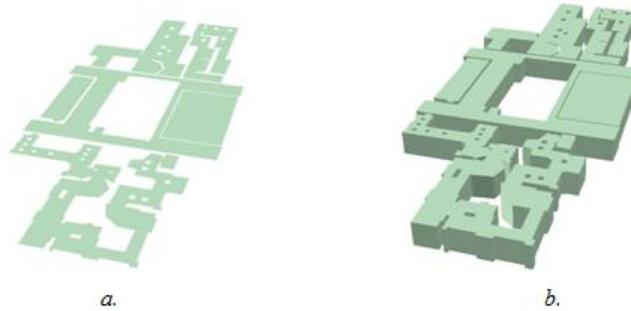


Figure 17: a) geometries stored in the footprint field, b) 3D geometries stored in the shape field.

It should be mentioned that semantic information stored in this table serves crucial context with regard to users' locations, i.e. which room is the visitor located. Also the 3D shapes stored in this table do not have any impact for the DaRen application, the field's existence is just to indicate the idea of 3D indoor space and might be useful for the 3D localization in the future.

Subspace table. This table is designed for the purpose of determining the closest node in the topologic network to the user's location so that the closest node is used as the starting point of the path. Briefly, using the original indoor space, the closest node can be found but by decomposing the original indoor space into smaller ones (Figure 18), the result can be more accurate, and specific procedure is given in the Chapter 5. Since this table is dedicated for the localization and navigation which are based on floor plan, only the footprints of these subspaces are stored. SQL command below creates the table and Table 5 describes the fields.

```
CREATE TABLE subspace (
  subspaceid integer NOT NULL,
  spaceid bigint,
  the_geom geometry(Polygon)
);
ALTER TABLE ONLY subspace ADD CONSTRAINT subspace_pkey PRIMARY KEY (subspaceid);
```

Field	Meaning
subspaceid	A unique integer number identifying each subspace
spaceid	Integer identifier indicating which original indoor space the subspace belongs to
the_geom	2D polygon representing the footprint of the subspace

Table 5: Fields in the subspace table.

In practice, the input data for this table is the shapefile of the manually divided floor plan (Figure 16) and it should be noted that doors and exhibits in original floor plan are integrated into subspaces. In the shapefile, there is an id attribute which starts from 0 and is used to distinguish each feature. Through *shp2pgsql*, the shapefile is imported into PostGIS as a table while the table only has the subspaceid which is the original id field with the *_geom* column. The spaceid is later added into the table by performing intersection operation between the *_geom* column and the footprint column stored in the indoorspace table.

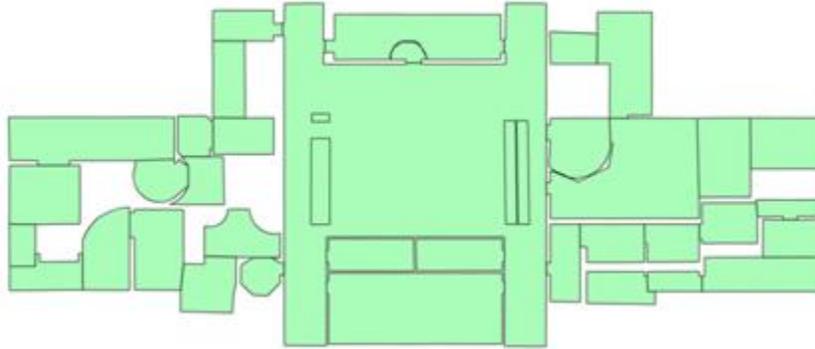


Figure 18: The subdivision of the original indoor space.

It should be addressed here that there is no specific rule for the subdivision, basically each subspace is kept as a concave polygon.

Locations table. This class is realized as a temporary table storing users' locations as points on floor plan and corresponding subspaces which they are in. Once user starts to navigate, the corresponding location table is constructed to store the position records. And once the request ends, i.e. the connection with database is over, the table is removed from the database.

SQL command to create the table:

```
CREATE TEMPORARY TABLE locations (locationid integer, subspaceid integer, geom geometry(Point));
```

In table 6, fields of the Locations table are explained.

Field	Meaning
locationid	The identifier of each location record
subspaceid	The identifier of the subspace in which the position is located
geom	2D point geometry representing user's location with respect to the floor plan

Table 6: Fields in the locations table.

The data inserted in this temporal table are x, y positions derived from fingerprinting method as is introduced in the Chapter 5. Later to determine the starting point in the topologic network for route searching, ten records are collected and used in order to minimize the influence of outliers.

And the geom column is used to store these ten positions while the locationid ranges from 1 to 10 with respect to corresponding position. And subspaceid value is later calculated using “within” operation to judge which polygon stored in the subspace table contains the position.

Objectinfo table. Objectinfo class is intended to include the context information of every indoor object which might provide service or visit information, i.e. exhibits information for visitors.

SQL commands to create the table.

```
CREATE TABLE objectinfo (
  objectid integer NOT NULL,
  theme character varying(256),
  name character varying(256),
  description character varying(512),
  year integer);
ALTER TABLE ONLY objectinfo ADD CONSTRAINT object_pkey PRIMARY KEY (objectid);
```

Field	Meaning
objectid	The unique identifier indicating which object the context is about
theme	The theme of the context and if the description is about exhibits, then the theme is exhibit
name	The specific name of the corresponding object
description	Detailed description of the corresponding object
year	The year when the information is collected

Table 7: Fields in the objectinfo table.

Since in the DaRen navigation system, the context focuses on the visit information, i.e. the descriptions of exhibits. Consequently, only the description of a group of exhibits are noted manually and stored in this table. The objectid is manually added when making notes of the context information. The theme is exhibit for all records. The name and description is collected from the annotation in the museum while the year is set to 2013 to all because the information is collected in 2013. In the future, service information needed by visitors like “help desk” can also be stored in this table.

Networknodes table. This table stores all the locations of topologic nodes with respect to the floor plan. So the geometries are expressed in 2D space. SQL command to create the table:

```
CREATE TABLE networknodes (
  nodeid integer NOT NULL,
  subspaceid integer,
  objid bigint,
  geom geometry(Point));
ALTER TABLE ONLY networknodes
ADD CONSTRAINT networknodes_pkey PRIMARY KEY (nodeid);
CREATE INDEX nodeidx ON networknodes USING gist (geom);
ALTER TABLE networknodes ADD CONSTRAINT nodefkey FOREIGN KEY (objid) REFERENCES
object (objectid)
```

Field	Meaning
nodeid	Unique integer identifier of each node starting from 0
subspaceid	The integer identifier showing which subspace the node belongs to with respect to the subspace table
objid	The same as objectid defined in object table and only the value can be null for a node
geometry	2D Point data type representing the location of the node on the floor plan

Table 8: Fields in the networknodes table.

Table 8 describes the meaning of fields in Networknodes table. In the museum rooms and corridors are large thus for navigation purposes they contain many nodes not one (Figure 17a). Each space in Figure 17a is subdivided into triangles and each of these triangles corresponds to a node which is centre point of triangle. For these nodes, the value of objid is null since they do not represent any indoor objects and the subspaceid is just the indoor subspace in which they are located. Besides since the exhibits and doors are navigable, they are also represented by nodes in the topologic network, and then corresponding objid values are attached to them. And because exhibits and doors also belong to a certain space, therefore these nodes also have subspaceids attached. This table is a modified version of nodes table which is generated semi-automatically from the floor plan by a Python script in the navigation part (Chapter 6). More specifically, the nodes table finishes the nodeid, subspaceid and geometry columns while the objid is later acquired from the intersection results between the geometry column and the footprint of object table.

It should also be noted that a foreign key is constructed on this table to conform a certain node cannot represent an object which does not appear in the object table. But actually the constraint is very weak because if the object table updates, certain nodes may become invalid and there is no constraint to verify that.

Networkedges table. This table stores the line segments of the topologic network and each edge connects two network nodes. The length of each edge is stored, which is later used for shortest path calculations.

SQL command to create the table:

```
CREATE TABLE networkedges (
  edgeid integer NOT NULL,
  source integer,
  target integer,
  cost double precision
);
ALTER TABLE ONLY networkedges
ADD CONSTRAINT networkedges_pkey PRIMARY KEY (edgeid);
```

The attributes are concluded in table 9.

Field	Explanation
edgeid	Unique integer identifier of each edge starting from 0
source	Nodeid of the first vertex
target	Nodeid of the last vertex
cost	The length of the edge on the floor plan

Table 9: Fields in the networkedges table.

The data source to populate this table is the networknodes table. As has been mentioned before, each indoor space is split into several triangles and each triangle is represented by a node which is just its centroid in the network. For each triangle, it can have at most three neighbors and the edge is just the connection between the centroids of two neighbour triangles. In addition, the nodes representing doors are added lately, so the edge can also be between the centroids of the door and a triangle. And this table is built using the same Python script for the networknodes. The distance is calculated using functions provided by shapely library.

Recroute table. This table is expected to store the information of recommended routes since there is a group of people who want to visit exhibits following a certain route according to the survey. There are two fields, i.e. routeid and nodeid_seq in the table. Routeid is used to identify each recommended route. And each of the routes is consisted of a sequence of nodeid, which means the route is actually built on the topologic network. However currently this class has not been realized as a table due to time limitations.

Relationships between tables. Although how tables are connected with each other has more or less been elaborated in the table explanation parts, the exact relationships still need to be analysed clearly since unique point of the data model is that all the information, i.e. geometry, topology and context are integrated together. In Figure 19 below, the connecting map is show.

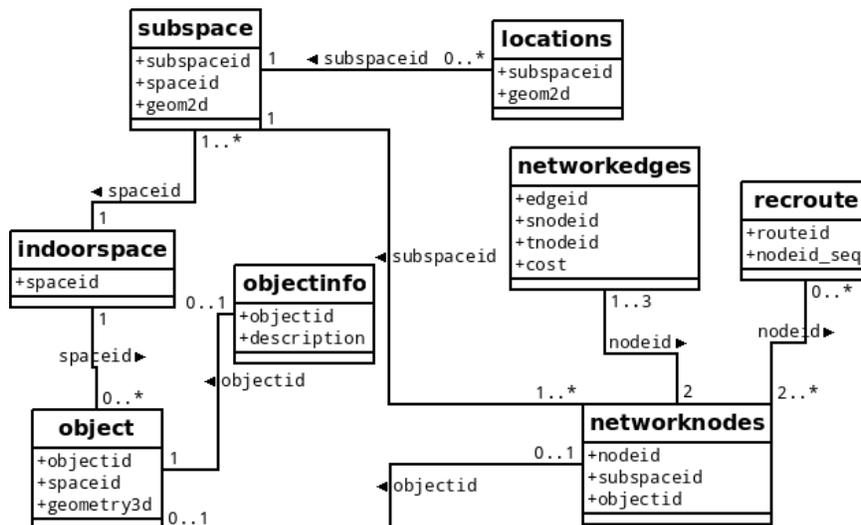


Figure 19: Connecting map of tables through ids.

It should be noted that in the above figure, some of ids' names are different from the ones realized in tables, but they have the same meaning. There are totally 8 relationships and they are concluded below,

1. Object and indoorspace. Through spaceid, each indoor object connects to a certain indoor space which can be room, corridor or hall. As has been mentioned, the wall is a special case since in the original 3D model, each wall is only attached to one indoorspace as is shown in Figure 14a. Conversely, in a certain indoor space, in principle, there can be no objects consisting it since the indoor space is an idea of the space which is formed by boundary faces.
2. Subspace and indoorspace. Through spaceid, each indoor space contains at least 1 subspace while each subspace must belong to a certain indoor space. Comparison between Figure 17a and Figure 18 shows this.
3. Locations and subspace. Through subspaceid, a subspace can hold zero or many visitors, i.e. locations. For the other way around, an x,y position should be within one subspace except outliers which may be located outside the floor plan but this is not taken into account.
4. Objectinfo and object. Through objected context information of an object is attached to the object while it is not always possible for an object to have context information. In DaRen application, only exhibits have context information.
5. Networknodes and object. Through objectid, navigable objects, doors and exhibits in DaRen case have corresponding nodes in the topologic network. While each node as is mentioned in the networknodes table can represent a part of the indoor space or the navigable objects, so not every node has an objectid attached.
6. Networknodes and subspace. Through subspaceid, each topologic node should be within a certain subspace and since each subspace can be navigated, so one subspace at least has one node in the network.
7. Networknodes and networkedges. Through nodeid, each edge is the connection of two network nodes. Conversely, because nodes represent the centroids of triangles as is discussed in the networkedges table, it can maximum have three neighbours. Thus at most, a node can join to construct three edges. For those nodes representing doors, since doors form constraints in the triangulation, it is connected with two neighbours as shown in Figure 54 in the next Chapter.
8. Recroute and networknodes. Through nodeid, each recommended route is at least forms by 2 network nodes. Reversely, not every node has the chance to be included in the route. While since the node can also represent exhibit, for some famous exhibits, many recommended route may pass through them.

Realization of main functionalities. There are mainly three functionalities that can be served by DaRen application and they are localization, navigation and 3D visualization which are described in detail in the next chapters. For the localization, it is implemented through locations table and subspace table in the database. When a user starts to localize himself, his position can be acquired

through fingerprinting approach and then the coordinates are sent to a temporarily constructed locations table. There the spatial computation is performed to identify in which room the user is. As to the navigation function, it is based on the networkedges table. The start node of the route is determined by localization subsystem and the end node is defined by the user of the application. Then the shortest path is calculated using pgRouting functions. For the 3D visualization, it is supposed to be built on the object table since the geometric information of original 3D model is directly imported into this table. But since the exporter is not ready yet, the visualization is still based on the original OBJ file. Additionally, for the context information which might be interesting to the visitors, objectinfo table is created to store the descriptions of exhibits.

Database structure of the whole DaRen navigation system is presented. And through the middle-layer, web service, it can communicate with user interface to process users' queries on both spatial and context information. The proposed and described data model treats geometry, topology and context information separately in order to achieve high efficiency to handle users' queries. Then the data model is realized inside PostGIS database making use of its native geometric data types and spatial index with each class implemented as a table. After creation of these tables, the data needed to support main functionalities of DaRen application, namely, localization, navigation, 3D visualization as well as the context information of exhibits can then be stored in the database. Another unique point in the implementation of database is that all tables are related to each other using ids. This on the one hand, can help solve complex queries since all tables are aggregated together, and on the other hand, these ids provide possibility to update all the tables consistently. Nevertheless, to make the maintaining work easy, an algorithm which can automatically extract floor plan from the 3D geometric model is also implemented inside the database. Therefore once the 3D model is updated, the floor plan can be generated automatically and tables which are derived from the floor plan are also updated.

In practice, due to time limitation, the localization and route calculations are conducted on the floor plan. However, the focus is still on the 3D geometries in the database, not only for the 3D visualization, but also for the potential localization and spatial calculation and analysis in the 3D space.

Nevertheless, notable shortcomings should also draw attention for the database.

1. When the geometric data is imported into the database, the database should be able to check the validity of each geometric object and may even rectify the invalid ones. Because invalid geometry can cause errors of spatial computation and analysis.
2. Although identifiers connect all the functionalities as well as different kinds of information together to enable the database response to queries efficiently and it can also facilitate the maintenance of the whole database, but still some issues might occur since many constraints need to be satisfied to update the whole database. For instance, if a navigable object is added into the object table, it should have a corresponding record in the networknodes table and also, more edges should appear in the networkedges table. And for the other way around, if a certain topologic node represents an object, then the

navigable value of this object must be true. All these constraints exceed the capability of simple native constraints like foreign key provided by the database. However, using a high level programming language, these constraints can be easily implemented. Consequently, it is required that an admin tool should be developed for the maintaining work of the database to make all the tables keep consistency with each other.

3. As is mentioned in the chapter, the 3D visualization is not fully realized based on geometries stored in the database and thus the visualization of DaRen project is still file based. To display 3D objects stored in the database on the mobile rendering engine, a specific exporter still needs to be developed to transfer the data stream. And this will be an important part in the future work.

In conclusions due to the complexity and inefficiency of previous data models for indoor navigation, a new data model is designed with focus on localization, navigation, 3D visualization and context service provided by DaRen application. The localization component is described in detail in the next chapter.

5. Localization

Positioning technologies has been integrated into mobile devices providing users the ability to easily determine their current location and find their way to their point of interest. Global Positioning System (GPS) is the predominant technology being used for outdoor navigation applications. However, GPS is not reliable for indoor environments due to signal interferences caused by walls, floors and other objects. This is a technical challenge for the system designers. So far, a large number of indoor navigation systems have been proposed but a standard technology for indoor localization is still missing. More thorough research into indoor positioning technologies and different techniques is necessary for developing a ubiquitous system for indoor navigation. The capability to determine a user's position in an indoor environment as accurate as possible is a key requirement for the success of every navigation system. This is due to the fact that the localization component provides the system with the starting point for every routing algorithm that calculates the point to a final destination.

In this project Wi-Fi localization technology using access points has been used. The main features of Wi-Fi technology that make it the most appropriate approach for the indoor navigation case of the Hubei Provincial Museum are the low cost technology, the cost effective scalability and the possibility to use existing infrastructures (WLAN - wireless local area network). An additional benefit for the user is that Wi-Fi technology does not rely on the use of extra hardware devices apart from his smartphone. All these factors will increase the possibilities of real-world implementation of the research. The research was focused on two Wi-Fi based localization methods, Wi-Fi monitoring and Wi-Fi fingerprinting using the existing WLAN infrastructure. The first method detects the beacon frames send out by mobile devices, laptops and other Wi-Fi enabled devices in range using Libelium Meshlium Xtreme monitors - Wi-Fi monitors (*Libelium, 2013*) while in the second one a mobile application receives signal strength measurements from multiple access points and compares them with the values obtained in a previous (training) phase.

The main question that needs to be investigated is how suitable these methods are for a context-aware 3D indoor navigation system for the Hubei Provincial museum. The system demands a few requirements of the localization part. The most important requirement is to provide the location of the user to the application so that the right starting point in the navigation network to be assigned. A second requirement is to provide the room in which the user is located to be able to inform the user on the exhibits in that room. Moreover, the system has to be able to provide the real time location of the user. Besides these user- and application- requirements, there is also a requirement from the museum point-of-view. The museum should be given insight on the behaviour of people and the amount of people present in the museum. The fact that the requirements of both visitors and museum managers were considered makes the realization of one mobile device application which takes into account both user types extremely challenging especially for the short time span for completing the project. Due to that two different subsystems should be considered. The first one regards the museum visitors and aims at navigating them in

the environment while the second one regards the museum managers and aims at the provision of information. The two sub-systems are necessary for meeting all users' needs.

In this section two research questions will be investigated:

1. *What is the most suitable localization method for DaRen system inside the time span of the project according to the relevant user type and how can this method be implemented in the project?*
2. *What are the possibilities of Wi-Fi monitoring in comparison to the fingerprinting method for the purpose of a 3D indoor navigation system?*

5.1 Localization Techniques

There is a vast number of different localization technologies which makes it a difficult process to investigate the possibilities of all of them. The most suitable localization technology for a specific application can be selected based on the requirements of the application and the performance parameters of the technology. At the highest level, all technologies can be divided into categories employing three different physical principles: inertial navigation (accelerometers and gyroscopes maintaining angular momentum), mechanical waves (i.e. audible and ultra-sound) and electromagnetic waves (i.e. using the visible, infrared, microwave and radio spectrum). Most technologies rely on electromagnetic waves and a few on mechanical (sound) waves (Mautz, 2012). Some of the most popular technologies are Wi-Fi, Bluetooth, RFID, Infrared, Ultrasound, Ultra-Wideband and Inertial Navigation Systems.

In this research, an indoor localization technology was implemented in order to locate and navigate visitors of the Hubei Provincial museum. A distinction has to be made between the terms positioning and localization. Positioning refers to the determination of global world coordinates while localization is the determination of relative coordinates (Chan and Sohn, 2012). Localization is succeeded by determining the place of the museum (room, corridor, session) that the visitor is located at. According to that, the requirements of the application in terms of accuracy are not very high as relative locations are used. However, the scalability of the system is an important parameter. The system should be able to be extended without influencing other parameters. Moreover the latency of the system should be low enough in order to achieve a real time localization and the amount of required infrastructure should be the minimum that is necessary.

Considering the above mentioned system requirements, the limited time for testing the technology and the restrictions in terms of equipment and funding, Wi-Fi is selected as the best solution. Wi-Fi localization technology using access points is used for deriving the position of a Wi-Fi enabled mobile device user. Results of conducted experiments reveal that a wireless access point-based indoor positioning system is feasible and a positioning accuracy of 1-3m can be achieved (Wang et al., 2003). The main advantages of Wi-Fi technology are:

- existing infrastructures (WLAN - wireless local area network)
- access points almost everywhere
- low cost technology
- quick installation
- cost effective scalability - ability to start with a small deployment and later to be expanded as the number of access points increases
- low latency - delay with which the requested information is delivered to the user

The most important reasons for rejecting other positioning technologies were the increased needs in infrastructures, i.e. tags, readers, beacons (RFID, IR), the high cost of devices and high set-up and maintenance costs (RFID, Ultra-Wideband, Infrared), the high scalability costs (Bluetooth), the high latency-time of response (Bluetooth, Inertial Navigation Systems) and the decrease of usability (RFID, Ultrasound) (*Martin et al., 2010*).

Having selected Wi-Fi as the most appropriate localization technology for DaRen project, the different Wi-Fi based techniques had to be considered. Two Wi-Fi based approaches that are based on signal strength measurements were considered. The decision on those techniques was made based on the existing equipment. For the first approach the MAC addresses and the signal strength of the mobile enabled devices is measured by the Meshlium Xtreme monitor and stored on an external database (*Libelium, 2013*). This method is called Wi-Fi monitoring. The main advantage of this method is that all mobile device types can be detected which makes this method accessible for all users. The second approach is the fingerprinting technique, which compares the unique signal data from a source send at a particular location with a map of data. This technique requires a training phase, in which the received signal strength at different locations are acquired and then stored in a database to create a map (*Fallah et al., 2012*). This mapping can be done by on-site measurements or processing limited measurements into a propagation model (*Evennou and Marx, 2006*). In the next phase, when the user is navigating, the received signal strength is measured and compared with the map to find the closest match. Both methods were tested in order to gain an insight on their performance parameters and to be decided the way they will be implemented in the project.

5.2 Wi-Fi Monitors

A prerequisite for access points based Wi-Fi localization of a mobile device user is the Wi-Fi monitor. In this chapter the capabilities of the Wi-Fi monitor and Wi-Fi monitoring as a method are specified by tests and possible localization methods for the 3D indoor navigation system are discussed.

5.2.1 Meshlium Xtreme

The Meshlium Xtreme router launched by Libelium company is the Wi-Fi monitor that is used in the current project. Meshlium Xtreme is a multiprotocol router for wireless sensor networks designed to connect ZigBee, Wi-Fi and Bluetooth sensors to the Internet through 3G

connectivity. It allows the detection of any device working with Wi-Fi or Bluetooth interfaces, such as iPhone and Android devices. In this research the device is going to be used to detect mobile devices through their Wi-Fi connection. The information read from each user by the Wi-Fi monitor contains:

- The MAC address of the wireless interface, which allows identifying it uniquely.
- The strength of the signal (in RSSI quality (0-100%) from signal strength -90 dBm and -40 dBm)
- The vendor of the Mobile device (Apple, Nokia, etc)
- The Wi-Fi Access Point where the user is connected (if any). Users not connected to an AP will be showed as "free users".

The recorded data can be stored in a local or an external database. The Wi-Fi monitor is plugged into the power, which allows the administrator to read the data when connected to the device. However, if the Wi-Fi monitor is connected to the internet, the records can be sent to an external database. This makes it possible for the administrator to access the data from any place when provided with Internet connection (*Libelium, 2013*).

5.2.2 Capabilities of the Wi-Fi Monitor

The capabilities of the Wi-Fi monitor have been investigated by performing a number of tests to determine the localization possibilities of it and to get acquainted with the type of information that can be provided to the visitors and the museum. In all tests a Samsung Galaxy S mobile device has been used. The aim of test 1 and 2 was to explore the reach/range and the signal strength of the device. Test number 3 was intended to look at how the signal strength and reach behaves in conditions where the two devices have an overlapping reach. In test 4 the flow of people was determined. Finally, in test 5 and 6 the stability of the device was explored. More specifically, test 5 targeted in discovering the influence of walls and ceilings on the signal strength and reach, while test 6 was performed in order to record the behaviour of the monitor when using two different mobile devices, the Samsung Galaxy S and the HTC Desire S. Tests 1 to 6 were conducted in the Architecture building of the TU Delft while test 7 has taken place in the LIESMARS building located at the campus of Wuhan University and test 8 was conducted at the Hubei Provincial Museum in China. A more detailed description on the deployment of the Wi-Fi monitor(s) and the outcomes of the tests are elaborated in the next section.

Test 1: *What is the range of the Wi-Fi monitors in different environments?*

For the first test the device was placed behind the counter of @hok at the ground floor. One of the testers carried a mobile device, which is detected by the Wi-Fi monitor. Also the signal strength was measured. The measurement was taken approximately every 5 meter until the mobile device was no longer in reach of the Wi-Fi monitor. With the following simple query containing the MAC-address of the mobile device, the tester could be immediately detected in the external database:

```
SELECT *
FROM Wi-Fiscan_1
WHERE MAC = "xx:xx:xx:xx:xx:xx";
```

Figure 20 depicts an overview of the measured signal strengths and the position of the user within the Architecture building. NS means 'No Signal'.



Figure 20: Measured signal strengths of test 1.

Test 2: *What is the range of Wi-Fi monitors in different environments?*

The same mobile device was also tracked on the first floor of the building by the same Wi-Fi scanner. Figure 21 shows the measured signal strength in approximately every 5 meters. An important difference between the test performed on the ground floor and the test on the first floor is that in the second case the device was placed in the corridor rather than behind a counter within a room. The results suggest that in the second test the devices have an increased reach compared to the reach of the first test.



Figure 21: Measured signal strengths of test 2.

Test 3: *How does the range of both Wi-Fi monitors overlap and how does the RSSI react?*

The third test was different from the first and second tests because in this one two Wi-Fi monitors were used instead of one. Figure 22 depicts the locations where the Wi-Fi monitors were placed and the values of the measured signal strengths at these certain locations. The area where the mobile device was detected by both Wi-Fi monitors is covered by red.



Figure 22: Measured signal strengths of test 3.

Test 4: *Can the direction be derived from the data of the Wi-Fi monitors?*

In the fourth test the two Wi-Fi monitors were placed in a room at the beginning of the BK City "Street" and in a room at the end of the BK City "Street" (Figure 23). The purpose of this test was to define the flow of the people passing the "street". It takes about 30 seconds to walk from one scanner to the other. To make sure that only the people walking from one Wi-Fi monitor to the other are in the results, a query was created to select only the entries on a certain day, with the same MAC address and that are scanned no longer than 2 minutes between the first and the last detection:

```
SELECT *, TIMESTAMPDIFF(MINUTE, Wi-Fiscan_1.TimeStamp, Wi-Fiscan_2.TimeStamp)
FROM Wi-Fiscan_1
JOIN Wi-Fiscan_2
ON Wi-Fiscan_1.MAC = Wi-Fiscan_2.MAC
WHERE DAY(Wi-Fiscan_1.TimeStamp) = 06
AND DAY(Wi-Fiscan_2.TimeStamp) = 06
AND TIMESTAMPDIFF(MINUTE, Wi-Fiscan_1.TimeStamp, Wi-Fiscan_2.TimeStamp) <= 2
AND TIMESTAMPDIFF(MINUTE, Wi-Fiscan_1.TimeStamp, Wi-Fiscan_2.TimeStamp) >= -2
AND HOUR(Wi-Fiscan_1.TimeStamp) >= 6
AND HOUR(Wi-Fiscan_2.TimeStamp) >= 6
GROUP BY Wi-Fiscan_1.TimeStamp;
```

The results in the *Direction* column show the direction in which the people are moving. When the result is a negative integer, the people are moving from the second Wi-Fi monitor to the first (the green arrow in Figure 23). If the result is positive, the people are moving from the first Wi-Fi monitor to the second (the orange arrow).

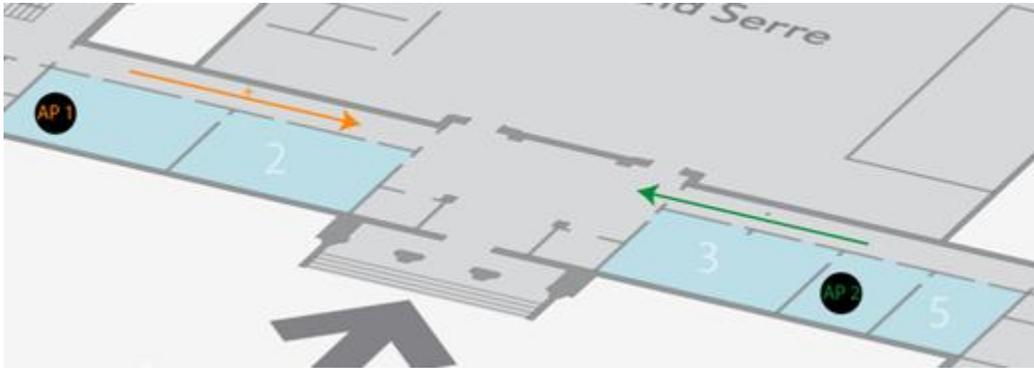


Figure 23: Direction of the flow of people in test 4.

Test 5: *How do walls influence the RSSI of the Wi-Fi monitors?*

Test 5 investigates the interference of walls and ceilings with respect to the signal strength. In this test the Wi-Fi monitor was placed in a corridor on the second floor which has adjacent rooms. This allowed the mobile device user to test the interference of walls by measuring the RSSI in every room. The result of this test is visualized in Figure 24, in which the signal strength values are depicted in red. Note that the rooms were relatively small and connected to an open corridor which could have caused some signals to get to the mobile device via multipath. This could have influenced the measured signal strength value.

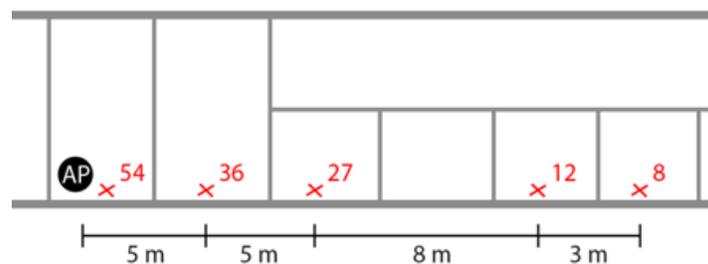


Figure 24: Cross-section showing the measured signal strength and distance from the access point in test 5.

Test 6: *How does the RSSI react on using different mobile devices?*

In test 6, both the Samsung Galaxy S and the HTC Desire S mobile devices have been used to indicate the differences in signal strength values on the same location. This test proves the

influence of different models of mobile devices on the Wi-Fi monitor scanning results. The same setup was used as in test 1. In Figure 25 the measurements of the test are illustrated.

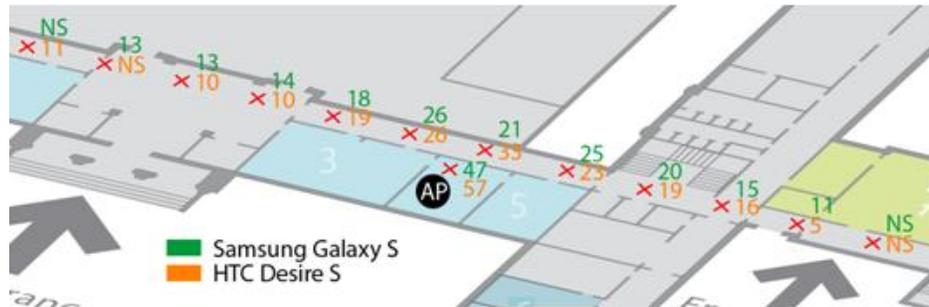


Figure 25: Measured signal strength from two different mobile devices.

Test 7: *Is it possible to derive a location using only two Wi-Fi monitors with a propagation-based localization method?*

The goal of this test is to find out the extend up to which it is possible to derive the location of a mobile device user when only two Wi-Fi monitors are available. This test was conducted at the first floor of the LIESMARS building at Wuhan University using the two Wi-Fi monitors which were available for this project. Figure 26 shows the location of the two Wi-Fi monitors (indicated with AP). The mobile device user chose 11 random locations (depicted by the green crosses in Figure 26) and stayed there until the mobile device was detected by both Wi-Fi monitors (if both of them where in reach of the mobile device). In order to derive the possible location of the mobile device user the maximum reach of the AP had firstly to be determined. The second floor of the LIESMARS building contains a lot of glass. This influences the reach of the Wi-Fi monitor and it is therefore necessary to determine the maximum reach in the LIESMARS building. Location nr 8 gives a good indication of the maximum reach as the signal strength for AP1 is 0. The signal strength at the scanner is 55% and from former tests a correlation of close to one is determined between distance and signal strength. Using this information it is possible to determine an expected location. However, an error of 10 in signal strength quality is assumed which leads to the spaces where the user is expected to be located (see the blue spaces in Figure 26). The blue spaces are the measured locations with their associated nr. while with green colour the corresponding actual locations of the mobile device user are indicated. Some of the expected locations are in the vicinity of the actual locations, which shows the potential of the possible correlation between RSSI and distance. However, with two Wi-Fi monitors it remains difficult to determine a unique location.

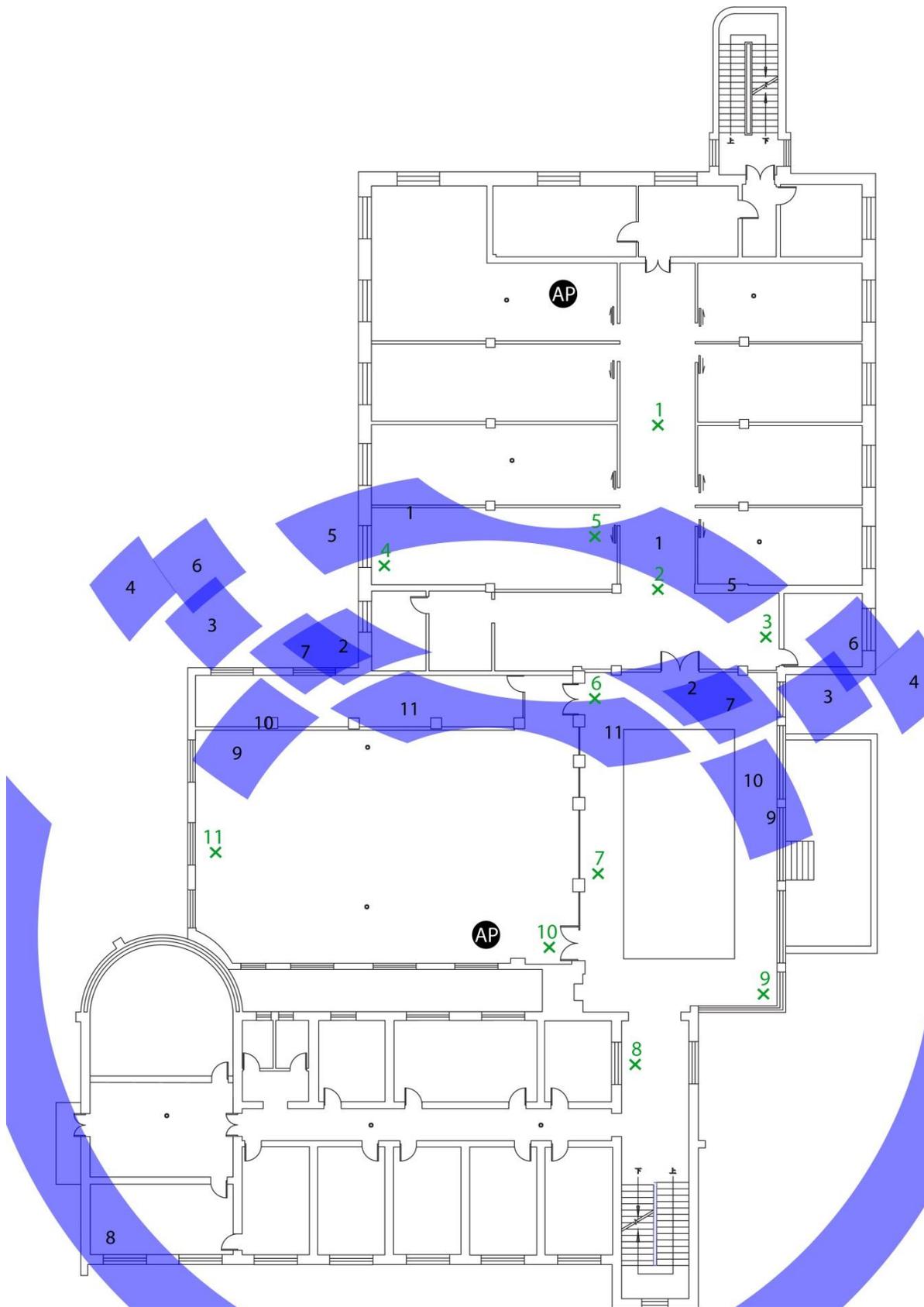


Figure 26: Actual and measured locations of mobile device user in test 7.

Test 8: *How do the Wi-Fi monitors operate in a multi-floor open space?*

In this test the signal strength of a Wi-Fi monitor is tested in the Hubei Provincial Museum. The goal of this test is to gain an insight on how the Wi-Fi monitors act in the museum environment where the storeys are connected by an open space in the middle of the floors. The Wi-Fi monitor is installed at the second floor. In Figure 27 it is depicted as a AP. Three different mobile devices (Samsung Galaxy S, Samsung Galaxy SIII and a HTC Desire S) were scanned in this test. The signal strength of each one of the devices was measured in four locations in every floor. It was expected that the different floors would have presented a clear difference in signal strength values, since every additional floor is even further from the Wi-Fi monitor. In Figure 27 the different floors with the tested locations and the signal strength values of every mobile device are depicted. The measurement results show no clear link between the distance and the signal strength. Presumably, this is due to the large open space (around 30 meters) that connects all the floors.

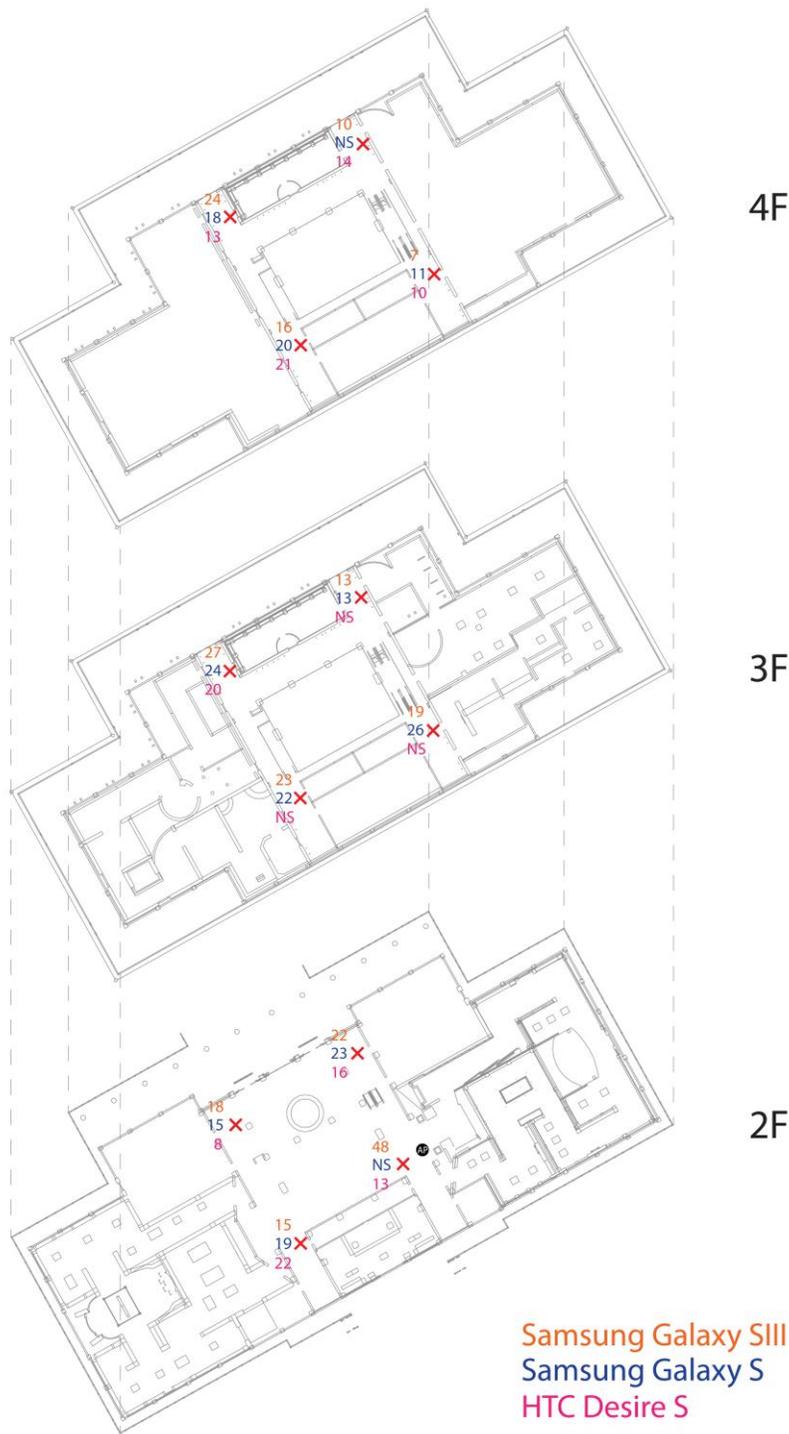


Figure 27: Second, third and fourth floor of the Hubei Provincial museum with the location of the Wi-Fi monitor (AP) and the measured signal strength of three different mobile devices at the location of the cross.

5.2.3 Results

After completing the tests, the correlation between distance and RSSI could be calculated. This contributes in estimating the signal strength attenuation when the mobile device moves away from the Wi-Fi monitor. For every test and in every direction from the Wi-Fi monitor, the correlation coefficient was calculated and a scatter plot was made using MatLab. The results are shown in Figures 28-30.

The correlation between distance and RSSI that came out of the first test is 0,9480 for the left path and 0,9442 for the right path.

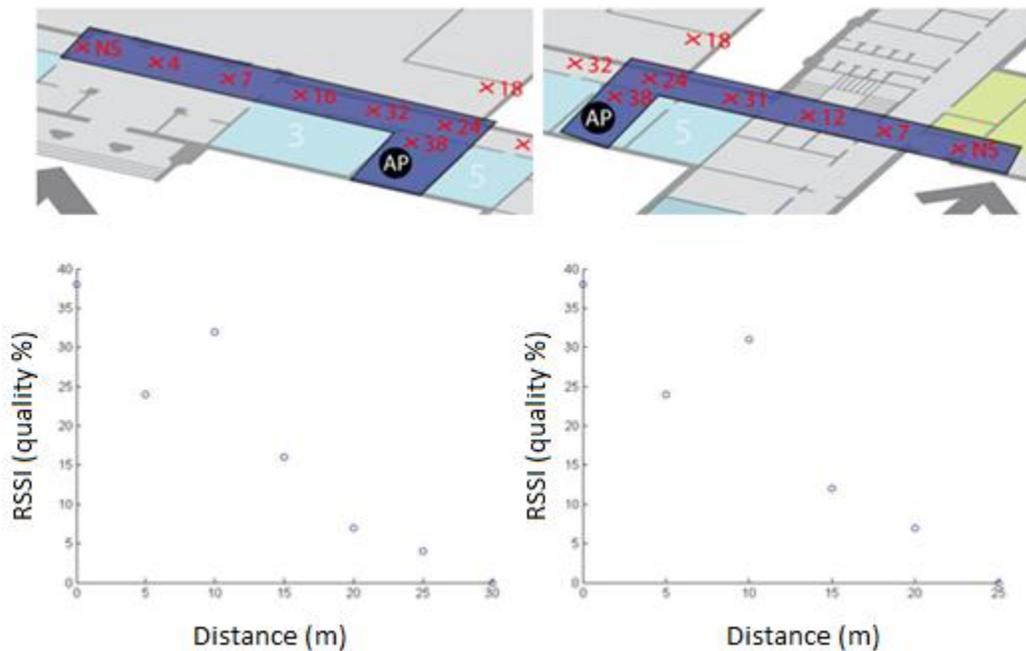


Figure 28: Correlation between distance (x) and RSSI quality (y) in test 1.

The correlation in the second test is 0,8913 for the left path and 0,8721 for the right path.

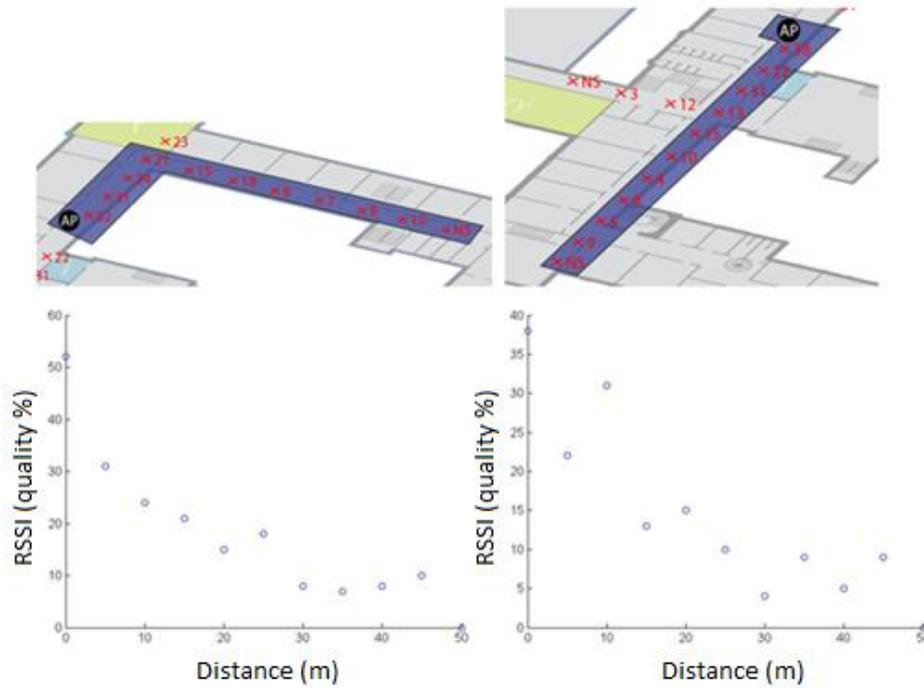


Figure 29: Correlation between distance(x) and RSSI quality(y) in test 2.

The correlation in the third test is 0,9542 for the path from AP1 and 0,9208 for the path from AP2.

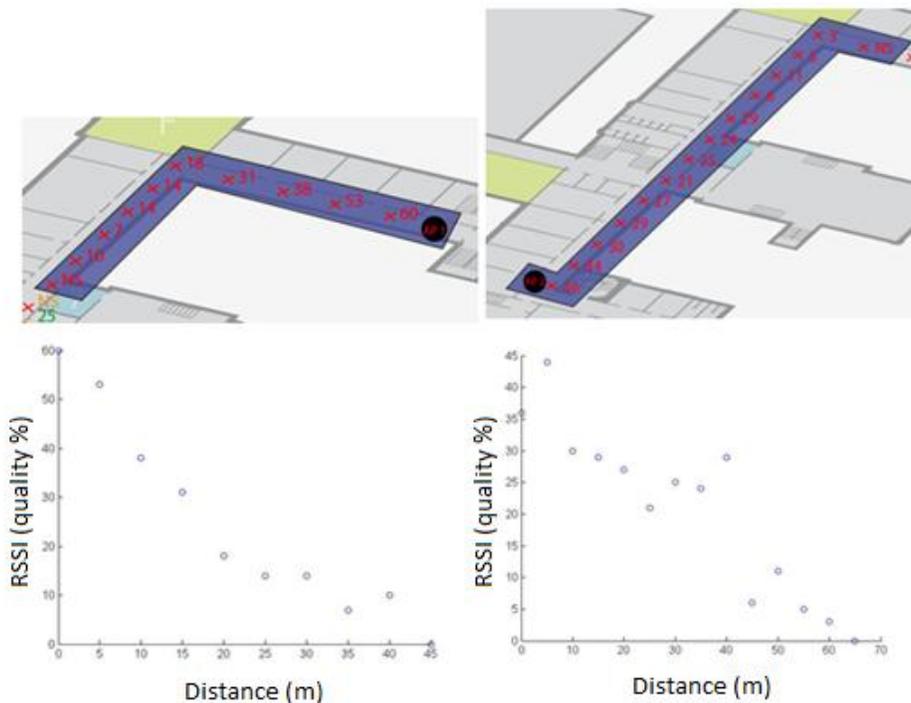


Figure 30: Correlation between distance(x) and RSSI quality(y) in test 3.

The differences in the correlation between distance and RSSI that came out from the different tests can be explained in various ways. The high correlation in the first test is partly attributed to the low number of received signals, which increases the possibility of getting a higher correlation value. The slightly lower correlation in the second test is attributed to the environment. The doors in the corridor are sometimes open and other times closed, which leads to differences in RSSI due to interference. Also when the mobile device moved around the corner, it was located sometimes behind a wall and sometimes behind a window which can also cause a difference in interference. However, when the results from the third test are also considered, there is an average correlation of 0,9218. This is a decent result and can therefore help to localize the user more efficient.

After comparing the results of test 5 with the results of test 2 (Figure 31), it becomes clear that the walls cause a small interference in the signal strength. The signal strength seems to reduce faster when it is blocked by walls. However, it should be taken into account the fact that the rooms were small and adjacent to a corridor which is directly connected to the corridor in which the Wi-Fi monitor was placed. This could cause the signal to bypass the walls (e.g. via multipath) and therefore influence the measured signal strength.

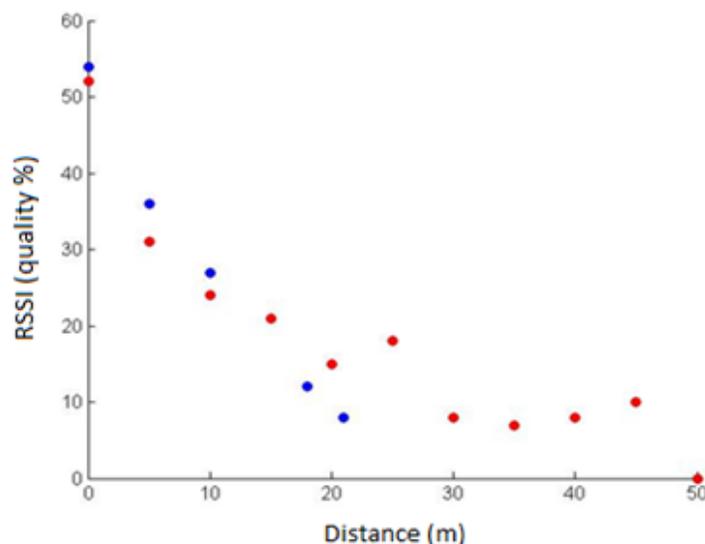


Figure 31: Test 2 (red) versus test 5 (blue).

With test 6 the stability of the Wi-Fi monitor was tested by using different mobile devices. The signal strengths of both mobile devices are depicted in a scatter plot (Figure 32). The overall trend demonstrates that there is little difference between the two devices.

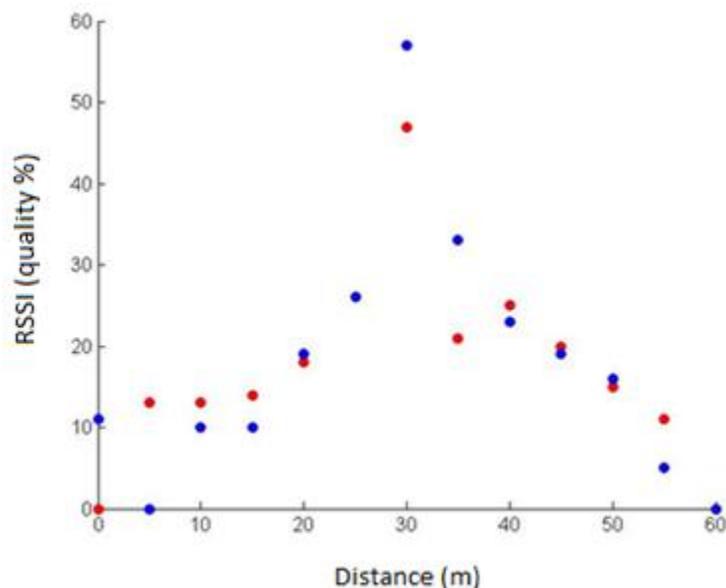


Figure 32: Samsung Galaxy S (red) versus HTC Desire 6 (blue) in test 6.

5.2.4 Problems and solutions

During the testing phase the benefits and drawbacks of the Wi-Fi monitors were revealed. The major issue with the Wi-Fi monitors is that the interval between two successful scans of the same mobile device is large (a couple of minutes). The scanning cycle of the Wi-Fi monitors is set at 5 seconds in most cases and even 2 seconds at times, but still the interval between two scans is between a minute and a couple of minutes. This is most likely due to the active scanning of the mobile devices and the interval between probe requests. Probe requests are the active scans by the mobile device. "During active scan, the client tunes its IEEE 802.11 radio to the channel being scanned and broadcasts a probe request in order to listen to any available probe responses from APs on the specific channel with the matching SSID" (*Sidiropoulos et al., 2012*). A directed probe requests let the client send a name-specific SSID request on which the specific SSID will response. A broadcast probe with a null SSID will cause all the APs to respond and these requests can be picked up by the Wi-Fi monitor. For a mobile device that is asleep, it sends out a probe request every minute. When the mobile device is in stand-by it sends out a request every 4-6 seconds. Only manual scanning for networks by the user can increase the interval (*Cisco, 2013*). The interval between the probe requests and the scanning cycle of the Wi-Fi monitors causes the Wi-Fi monitors to miss some requests, which leads to larger intervals than necessary. For localization purposes the current interval between two scans is too big, because within a minute a user can already be moved from one room to the other. Also it is desirable for a localization system that all Wi-Fi monitors, in our case two, to scan the user at the same time. This cannot be guaranteed with our Wi-Fi monitors. A smaller interval between consecutive probe requests would be a solution to scan the mobile devices more often. According to Musa and Eriksson (*2012*) a smaller interval can be achieved by "opportunistic AP emulation". Besides the broadcast probes a mobile device sends out frequently, it also frequently sends out directed probe requests

for a specific SSID that it has been associated to in the past. The idea is to change the SSID of the Wi-Fi monitor in a SSID that the user has used in the past. In general this SSID can be well-known providers that are in the memory of many mobile devices. When the mobile device tries to associate with the "known" SSID, it will encounter a different security protocol than it has remembered for that SSID. Because it cannot connect to the AP with the unknown security protocol it will continuously sending out null packets to see if the AP is still there. In that way the theory on how the Wi-Fi monitor will receive additional probe requests is explained. In practice, the SSID of one Wi-Fi monitor is changed to 'tmobile', a well-known provider that is in the memory of the mobile device. The other Wi-Fi monitor was given a different name. For 25 minutes both Wi-Fi monitors where scanning the same mobile device. The Wi-Fi monitor with SSID 'tmobile' scanned the mobile device 27 times, every 55,5 seconds. The Wi-Fi monitor with a different SSID scanned the same mobile device during the same 25 minutes 25 times, every 60 seconds. The change in the amount of scans was not as expected. The performance of the Wi-Fi monitor with SSID 'tmobile' was not significantly improved, the Wi-Fi scanners still detect a mobile device on average once in 11,5 scanning cycles.

The second issue was that not all mobile devices were scanned by the Wi-Fi monitor. Especially newer devices tended to be invisible in the logs. At a certain time, one of these newer devices did show up in a log of one Wi-Fi monitor but not on the other. Apparently, one of the Wi-Fi monitors had been previously updated and the other was not. After updating the Wi-Fi monitor, the newer mobile devices were scanned. The last issue is a practical one. For the scope of this project, only two Wi-Fi monitors were available. Localization with only two scanners is a difficult task. For example, lateration-based methods require at least three APs for localization and a fourth is recommended to minimize the error.

5.2.5 Possible Implementation of the Wi-Fi Monitors

Being aware of the disadvantages of the Wi-Fi monitors, for determining the implementation of these devices the advantages are equally important. The advantages of Wi-Fi monitoring are that there is a clear correlation between distance and signal strength. This makes it unnecessary to make a radio map, only a local inventory of the signal strengths and distance is advised since many local factors influence the signal strength and reach (e.g. walls, windows etc.). Another advantage is that all mobile device types are detected which makes this method accessible for all users, including users of Apple devices. Also the possibility to store all the data externally on a database is a big advantage. In this way all the data can be reached from any location without the need for information from the mobile device of the user.

This last advantage can be used for providing information to the museum. As it has been demonstrated in test 4, the direction of the movement of people can be computed and also the amount of people can be calculated by using information provided by the Wi-Fi monitors. The museum managers can retrieve this information every moment of the day from their computer, without specific information from the user. The question remains on how the two Wi-Fi monitors can be used for localization. As it has already been mentioned, two Wi-Fi monitors are not

sufficient for a proper lateration-based localization method. With a fingerprint-based method two Wi-Fi monitors might be used to localize with a certain error, but the uncertainty of this method and the short time span of the project made feasible only an experimentation with this possibility. In section 6 this method is applied and compared to the implemented in the application method, which is original fingerprinting. The strong points of the Wi-Fi monitors for providing information to user and museum were considered and therefore the available two Wi-Fi monitors were implemented in the existing WLAN-infrastructure. In this way the Wi-Fi monitors were used for localization, provision of information and experimentation with a new fingerprinting method with only two Wi-Fi monitors.

5.3 Fingerprinting Technique

Fingerprinting is a commonly used indoor localization method due to the high accuracy of location determination and low costs of implementation as it is usually possible to use (a part of) the existing infrastructure (*Haojun et al., 2011*). Another advantage is that this technique can have a high location calculation frequency, also called location granularity. This means that the location calculation frequency can be adjusted to what is necessary for the purpose of the indoor localization. This makes it employable for a wide range of purposes. However, there are also some important disadvantages of Wi-Fi fingerprinting. Firstly, implementation is extremely time consuming since the offline phase takes a lot of time and effort (*Machaj and Brida, 2012*). Secondly, fingerprinting does not work well in open spaces due to the lack of unique signal strengths in such spaces (*Lu et al., 2013*). Thirdly, there are many external circumstances that influence the accuracy of the localization. For example, the presence of public or using different kind of mobile devices influences the measured signal strength and thus the accuracy of the measurements (*Ma et al., 2008*). Finally, an important disadvantage of this technique is that Apple devices are excluded from this localization method because they are not able to carry out a Wi-Fi scan since all such sniffer applications are banned. Fingerprinting is based on comparing the unique signal data from a source send at a particular location with a radio map of data (*Fallah et al., 2012*). Localization based on fingerprinting technique includes two phases, an offline phase and online phase. The offline or training phase is about collecting the data for the radio map and in that way obtaining a unique signature of signal strengths of different access points at multiple locations at the research area (*Rehim, 2004*). The online phase includes the comparison of the received signal strength values to the radio map and in that way give the approximate location of the user (*Rehim, 2004*).

5.3.1 Theory

In this research a Wi-Fi fingerprinting application created in the Wuhan University was used. This application makes use of the Parametric Signal Strength Distributions for location determination described by *Rehim (2004)*. As it has already been stated, the process consists of two phases, the offline and online phase.

Offline Phase. In the offline phase a radio map is created, signal strengths of the present access points are measured and processed. It is assumed that the signal strength is normally distributed. The location of the various access points is unknown, which reduces the problem to defining the parametric distribution fitting the signal strength as if it would come from a single access point. The parametric distribution is defined by calculating the probability density function (pdf) of the Gaussian distribution.

Online Phase. In the online phase the users measurement (signal strengths s) is compared to the previously created radio map and a location x with the highest probability P is selected and returned. To find the highest probability of certain signal strengths at a certain location, once again the Gaussian distribution is used (*Rehim, 2004*).

5.3.2 Tests and Results

The fingerprinting application of Wuhan University was tested in order to investigate its basic functionalities.

Test 1: *How accurate does the Wuhan fingerprinting application locate a user in a room?*

The aim of the first test was to estimate the accuracy of the Wi-Fi fingerprinting application of Wuhan. The test was carried out in the LIESMARS building at the Wuhan University. The main requirement of this test was to find out the percentage of locations which appear to be in the application outside of the room the user is actually located at. The importance of this test lies on the fact that DaRen system employs the Wi-Fi fingerprinting system of Wuhan to determine the right starting point for the navigation. Also, the percentage of correct room assignments claims the accuracy of a location in terms of spatiality than the accuracy in terms of meters. The test has been carried out twice. The first time the user was in the middle of the room with the application running for 10 minutes. The Wi-Fi fingerprinting application of Wuhan derives a location every 0,5 seconds which means that 1200 locations were obtained within the duration of this test. Figure 33a illustrates the result of the test. The blue dots are the locations derived by the application. A maximum of 10 times the user was presented to be outside of the room. This means that 10/1200 times the user is mis-located, that is 0,83 per cent. The rest 99,17 per cent the user location is correctly assigned to the room.

The second time this test was carried out the user was standing at the edge of the room. The results are depicted in Figure 33b. This time the user was mis-located 216 times out of the total of 1200 measurements. That means that an erroneously assigned room occurs 216/1200, which is translated in 18 per cent. This is important due to the fact that in order for the application to achieve a better accuracy the user will have to wait a little for the application to determine with reasonable certainty the room the user is located at.

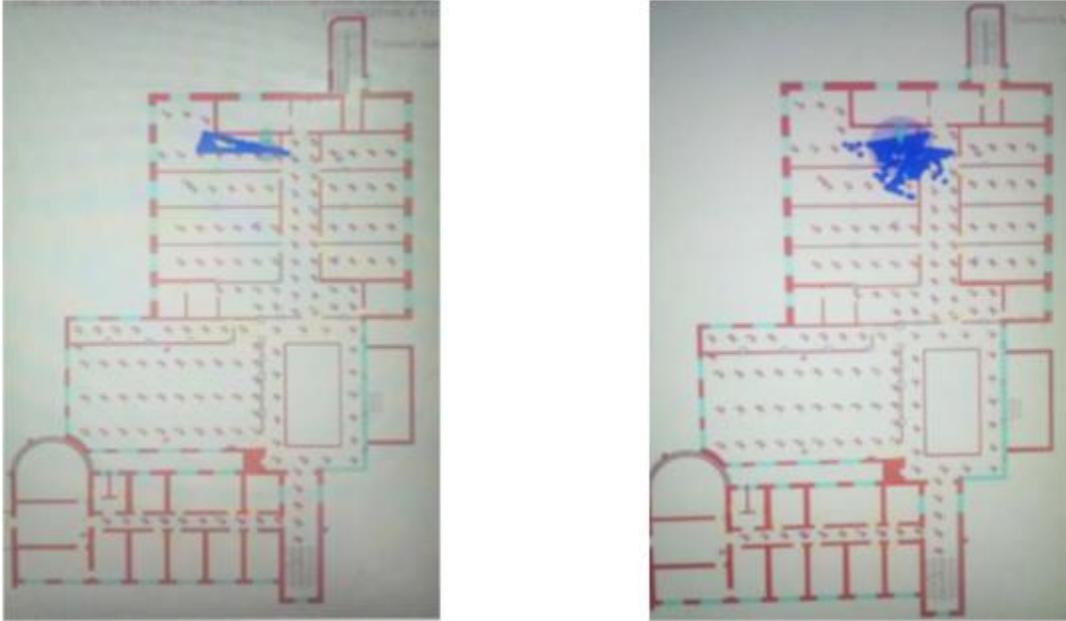


Figure 33: Mobile device user in the middle of the room, Mobile device user at the edge of the room.

Test 2: *How does the Wuhan fingerprinting application function in open space?*

Localization based on fingerprinting technique in a large open space is challenging. Due to the lack of a unique fingerprint in open spaces it is difficult to derive the location of the user. In order to prove this statement for the Wuhan fingerprinting application, a test was conducted in which the user walked around an open space where all floors are connected. Figure 34 depicts the result of this test. The blue dots are the locations according to the application while the user was walked from the top to the bottom, around the hole and then back to the top. While conducting this test it was very clear that in the corridor the accuracy of localization was quite good, but as soon as the mobile device user reached the open space the blue dots appeared all over the place. This is an important conclusion since it shows that localizing in open space is not trustworthy and that should be noted with regard to using the DaRen application.

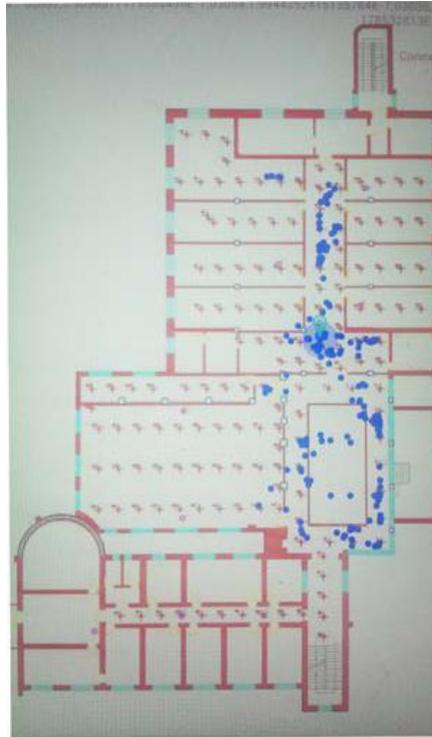


Figure 34: Testing Wuhan fingerprinting application in an open space.

5.4 Implementation

The Wi-Fi fingerprinting application of Wuhan University was used to determine a proximate location of the user which is a key requirement for the functionality of the DaRen 3D indoor navigation application. The Wi-Fi monitoring was used to provide the museum with additional information about the visitors' behaviour. In this chapter first the implementation of the Wuhan fingerprinting application in the DaRen application is described and secondly Wi-Fi monitoring is assessed.

5.4.1 Offline Phase

In the offline phase of Wi-Fi fingerprinting the radio map of the Hubei Provincial Museum was created. In the following subchapters the different components of the implementation of the offline phase are described.

5.4.2 Reference Points and Floor Plan

The first step to create a radio map was to determine reference points in the locations of interest. The reference points were placed in the floor plan of the first floor of the Hubei Provincial Museum. All points are located in the west wing (Figure 35). Only the west wing of the first floor was used, because creating a radio map for the entire floor/museum would be extremely time consuming. 166 reference points in total were used. Sixty (60) different MAC addresses of access points were detected. However, these access points also included laptops and mobile devices that

were turned into local access points for the time being. Around 30 access points were static and could therefore be used to create the radio map, including the two Wi-Fi monitors. The distance between the reference points in open space is around 4 meters. In practice this corresponds to the corridors that lead the visitors from one side to the other, passing the open space in the middle that connects all the floors together. In open spaces the variation of signal strengths is low in contrast to closed spaces where the variation is high (Lu *et al.*, 2013). Therefore it is not necessary to have a dense grid of reference points in open spaces. In closed spaces reference points were placed with approximately 2,5 meters distance, depending on the shape of the room. The latter was used for the areas with the exhibitions.

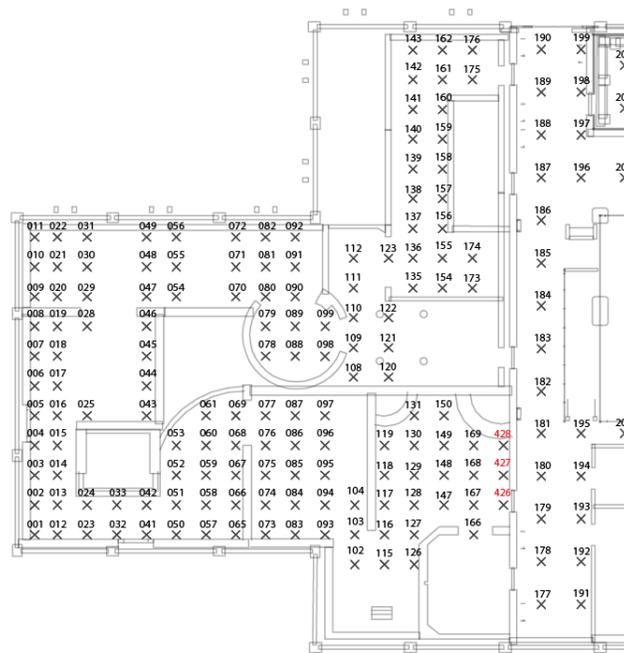


Figure 35: Hubei Provincial Museum's floor plan with reference points.

5.4.3 Creating Radio Map

Every reference point was measured into four different directions (north, east, south and west), using the Wuhan University application. This was necessary because the body of the mobile device user can influence the signals of different access points coming from different directions. The museum is oriented towards the north east (29 degrees) and to correct this, the measurements should all have been done turning with a 29 degrees with respect to the museums orientation; thus the measurements were done facing the four directions. The signal strength of each direction was measured for one minute which means that it was taking at least four minutes to collect the measurements from each reference point. It should be noted that the time required was usually longer due to documentation and processing. As noted before, this method is time consuming; for collecting all the data to create the radio map a bit less than 17 hours of measuring with the same device are needed. This is the case only when no problems or mistakes are encountered.

Xml and Processing. All measurements were saved locally at the mobile device in xml files. In every xml file measurements of one direction were saved. The xml files contain the following information about every measured access point: time of the measurement, MAC address of an access point, SSID which is the name of the access point, the measured signal strength and the bandwidth of the signal. The name of the xml file refers to the location (reference point) and direction of measuring. The measurements taken from the xml files were automatically processed.

Improvements. Several difficulties were encountered during the offline phase. Firstly, due to time limitations, the measurements were collected by multiple persons at the same time carrying different mobile devices. Every mobile device has different characteristics that can influence the measurements of the signal strengths (*Haojun et al., 2011*) therefore measuring and testing with the same type of device is considered to be more accurate. Second, during the data collection phase, the museum was exceptionally busy due to holidays. Figures 36 and 37 show the influence of human body on the signal strength. The presence of a large number of people during the measurement phase influenced the signal strengths measured by the mobile devices consequently biasing the radio map which was based on these measurements. The online phase took place on a quiet day. The difference between the offline and online phase can influence the accuracy of the localization.

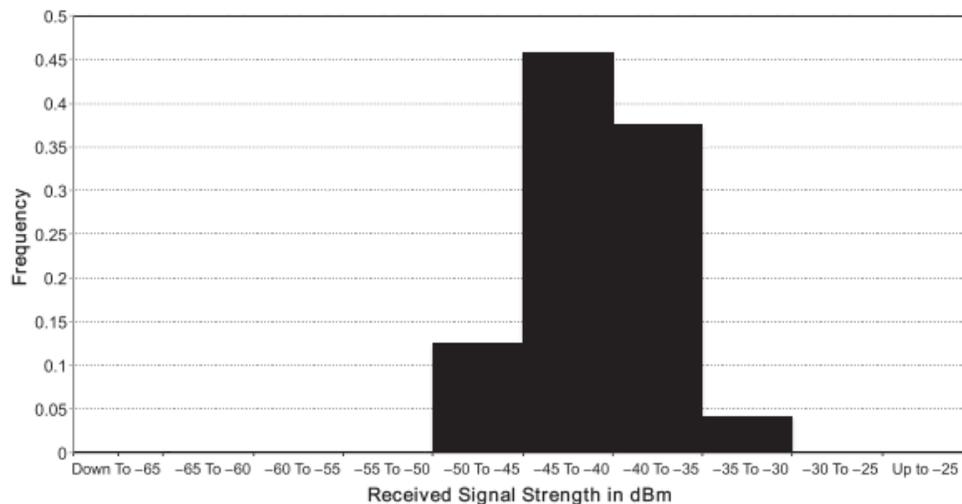


Figure 36: Histograms of RSS without human presence (*Chan and Sohn, 2012*).

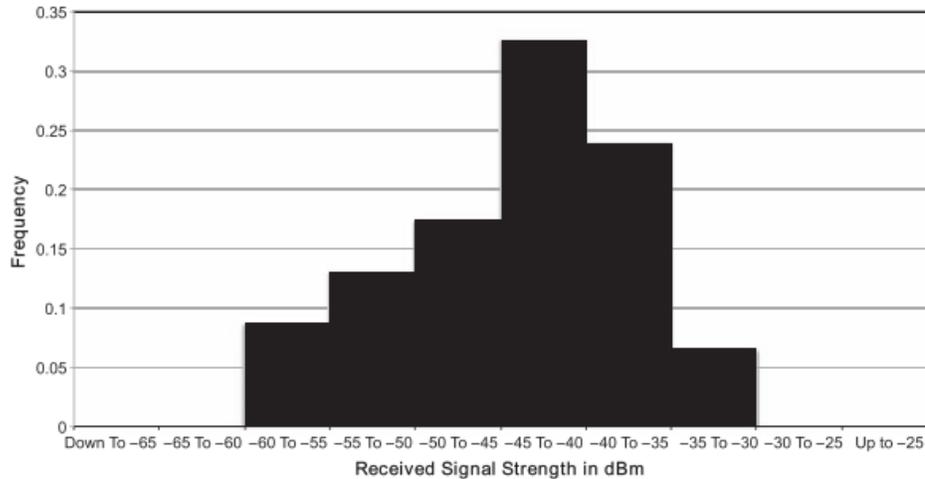


Figure 37: Histograms of RSS with human presence (Chan and Sohn, 2012).

5.4.4 Online Phase

After the offline phase was completed the functionality of the online phase was tested. In the online phase the location of a mobile device user is determined. The x, y coordinates of the user are determined using the Wuhan fingerprinting application. A test was conducted to estimate the accuracy of the location determination returned by the application and to evaluate the functionality of the application in open space. The mobile device user walked in a straight line through the hallway (blue line in Figure 38) requesting a location regularly for twenty-one times and labelled in consecutive order. The twenty-one location that were returned by the application are plotted in Figure 38. The points show the locations given by the fingerprinting application.



Figure 38: Testing the functionality of Wuhan fingerprinting application in the museum.

As shown in Figure 38, the locations returned by the application are relatively close to the blue line. However, the numbers are not always in a consecutive order, i.e. the first location calculated (number 1), is located after number two, three and four. The latter highlights the importance of taking into account multiple x, y positions for the calculation of an average location where the outliers are averaged out.

5.4.5 Additional Processing

In order to determine the starting point of the navigation in DaRen application, the location of the user is essential. The determination of the starting point can be done by determining the users' current location and then searching for the closest node in the NetworkNodes table in the database so that the navigation to start from this node. However, as already noted, a single x, y position is not sufficient to determine the users position due to outliers. Therefore ten x, y positions was decided to be collected at the mobile device of the user and then processed as follows:

1. The first step contains a point in polygon operation. For each of the ten points (x, y positions) it is checked in which room the point is located. The room that contains most of the ten points is considered the room in which the user is located. Therefore, only the nodes that are located in that room are considered to be a possible starting point for the indoor navigation.
2. In second step, the distance of each point to each node of room is determined, (including the distance from the points that are located outside of the room). This is important as the points that are located outside of the room might be close to the actual location of the user (for example when the user is close to the edge of the room). Not taking these points in consideration provides an incomplete image of the location of the user.
3. In the third step, the average distance from the points to each node is calculated. The node with the smallest average distance is considered to be the closest node to the user and is used as a starting point for the navigation.

The code for the processing steps described above is provided in Appendix A.

Wi-Fi monitoring is used to provide the museum with additional information about its visitors. Data is collected from the two Wi-Fi monitors that are placed in the museum during the measuring hours on the 4th of October 2013. Based on this data the following information was obtained:

1. Number of unique visitors with a Wi-Fi enabled mobile device in the exhibition is obtained through counting the unique MAC addresses in the database.
2. Time these visitors spent at the exhibition.
3. Direction the visitors walked through the exhibition.

The collected data was processed using MySQL (Appendix A) with 1327 unique MAC addresses scanned by the Wi-Fi monitors during the day. According to the data collected, 1012 visitors visited the exhibition at least for one minute and 727 visitors stayed there longer than 10 min.

Lastly, the direction was determined by calculating the TimeStamp difference between the first and last time a unique MAC address was scanned (largest scanned interval). All measurements in between were deleted since a mobile device could have been scanned multiple times by both Wi-Fi monitors. For practical reasons the largest and smallest scanned interval per MAC address were extracted. To determine the direction the largest scanned interval is interesting. To obtain this a threshold of 20 minutes was taken into account in order to filter the smallest scanned intervals. Although this method is not ideal since some smallest intervals might be bigger than 20 minutes, it does give an idea of the percentage of people that walk through an exhibition the intended way or not. It turned out that approximately six out of seven visitors followed the exhibition as intended by the museum. One out of seven followed the exhibition in opposite direction.

5.5 Wi-Fi Monitored Fingerprinting Method and Comparison

Wi-Fi Monitored Fingerprinting. "Fingerprinting techniques are especially appropriate for the range of frequencies in which GSM and Wi-Fi networks operate (aprox. 850 MHz to 2.4 GHz) because of two main reasons: the signal strength at those frequencies presents an important spatial variability, and also a reliably consistency in time (despite the variable nature of radio signals)" (*Martin et al., 2010*). Additionally, the authors state that the fingerprinting methods already proved to be able to deliver better accuracies compared to propagation-based methods.

As seen in test 7, the propagation-based method for only two Wi-Fi monitors was not sufficient for localization. Fingerprinting with these two monitors was not applied. Due to the uncertainty of this method for two monitors it is not implemented in the application, however in this section an attempt has been made for localization with only two Wi-Fi monitors. The Wi-Fi monitors were already taken into account in the WLAN-infrastructure in the fingerprinting method for the DaRen application. Therefore the data needed for fingerprinting is already present in the database of the Wi-Fi monitors. The data is stored at the local database, since it is not possible to connect the Wi-Fi monitors to the internet. The data was derived from the local database, stored in .csv files and inserted into the database.

Similar as with normal fingerprinting an offline phase is needed to create the radio map. All the reference points are already measured for the actual fingerprinting method of the DaRen application, however a radio map has to be created. The RSSI measured by both Wi-Fi monitors for every reference point is required to create a radio map of this method. The exact time of every measured reference point is known, which is at least 4 minutes, including the MAC-address of the mobile device that was used for measurement. A query in the database can filter the data, so that the different RSSI values remain for the given reference point and Wi-Fi monitor. The averages of these values are taken into account. This has been done for every reference point for

both monitors until a table is created with X, Y and RSSI thus creating the radio map. By rasterizing the data, this also can be visualized (Figure 39).



Figure 39: Radio Map of a Wi-Fi monitor (purple dot) in Hubei Provincial Museum.

The green colour indicates a high RSSI and the red colour a low RSSI. The purple dot represents the position of the AP. The radio map gives a clear view of the correlation between the distance and RSSI (as depicted by the colours). It should be noted that, the white cells in the floor plan are primarily the consequence of the fact that not all reference points are measured in the museum due to different circumstances.

When the user is scanned in the online phase, the RSSI is stored in the database of both Wi-Fi monitors. These can be used to determine a location. In this Wi-Fi monitoring fingerprinting method the RSSI is used as follows (for a cell in the raster):

$$(RSSI \text{ Radio Map} - RSSI \text{ User})^2$$

Equation returns a value that gives an idea of the probability of where the user is located. The higher the number, the more unlikely it is that the user is in that location. The lowest number the more probable that the user is at that location. For example, the measured RSSI of the user is 13 and in the radio map there are cells with values 13, 21 and 32. The value returned by the equation will be respectively, 0, 64 and 361. Thus the user is most likely to be at the cell with value 13, then 21 and less likely 32. Applying this method for both Wi-Fi monitors will create two different probability maps (Figure 40). A known position and the associated measurements (RSSI of 13 and 15) are used for these figures.

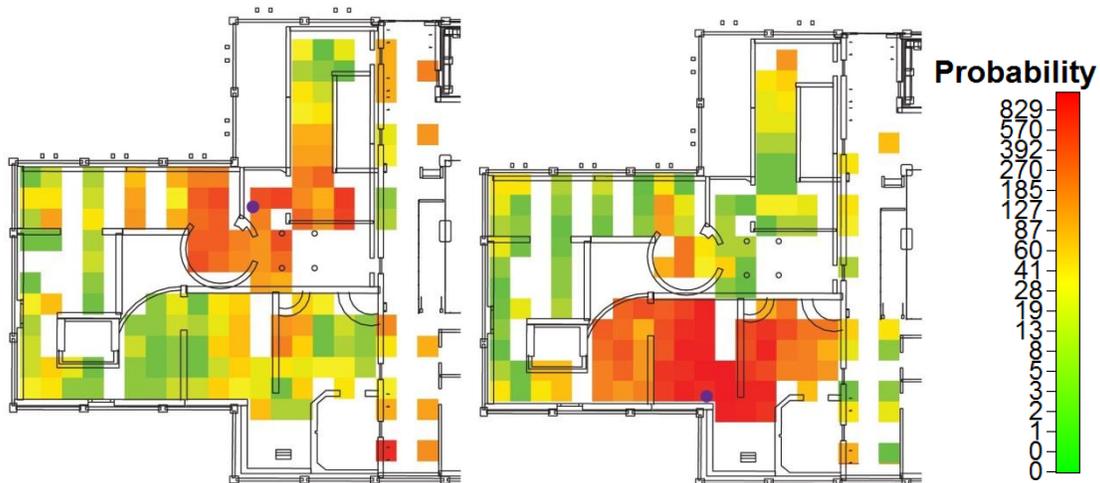


Figure 40: Probability maps of both Wi-Fi monitors.

The probability maps show the chance of the user being at a specific reference point or cell. Green colour indicates high probability while red – low probability. These maps are still separate and only show the probability per Wi-Fi monitor. By combining these maps, an approximation of the location by two Wi-Fi monitors is given. Figure 41 provides a visualization of the final map containing the most-probable location of the user represented by the dark-blue dot, calculated by selecting the cell with the lowest value in the raster, as explained previously. The light blue dot represents the actual location of the user. The difference between these two locations is approximately 5 meters.

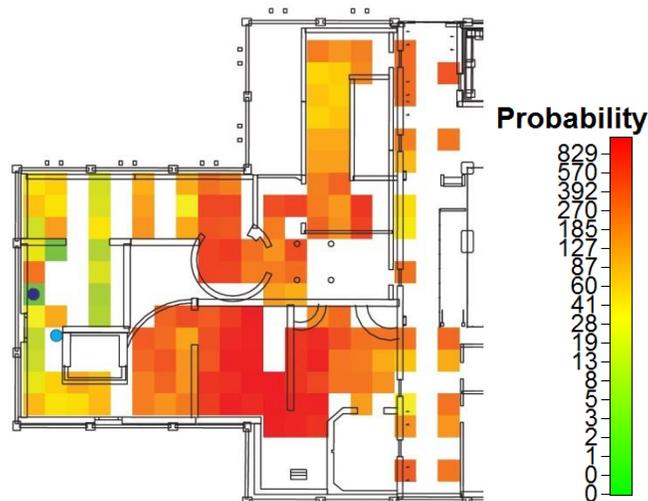


Figure 41: Final probability map of two Wi-Fi monitors containing the calculated location (dark-blue dot) and actual location (light-blue dot).

However, there are cases where more than one location is given by this method as provided in Figure 42, where the location is calculated by an RSSI of 13 for one Wi-Fi monitor and 21 for the

other. In this case, more than two Wi-Fi monitors could increase the chance of deriving only one location.



Figure 42: Final probability map with RSSI 13 and 21 returns more than one location.

Comparison. There are some advantages and disadvantages of the method as compared to the original fingerprinting which are discussed below:

1. The advantage of Wi-Fi monitoring fingerprinting stems from the fact that no additional application is required to scan the APs and derive the RSSI. On the other hand, Wi-Fi monitors are more expensive than the regular APs used for original fingerprinting.
2. Additional advantage is that all the data is stored directly in the database. The data is accessible every moment of the day and in thus can provide real-time information to the museum. Adding additional Wi-Fi monitors will enrich the information provided thus improving the quality. Furthermore, when the data is stored in the database, it is possible to provide information to the user about the location of other people thus enabling him to avoid crowded exhibits. This also includes people who are not using the application, but who only have their Wi-Fi enabled on their mobile device. On the other hand, the downside of storing data in the database related to privacy issues as users should give permission to let the application use their location for different purposes.
3. The interval between two consecutive scans with the Wi-Fi monitors is relatively big and a smaller interval is required for localization. Visitors of the museums tend to walk slow and therefore give the Wi-Fi monitors the opportunity to scan the users while they are still in the same place. However, this is not a suitable for localization purposes.
4. Besides the differences in method, there is also a difference in signal strength. A comparison has been made between the signal strengths measured by the Wi-Fi monitors and measured by the mobile device. Measurements are used from both devices scanning each other at the same time. A conversion for the signal strength from the mobile device

is necessary. The mobile device measures in dBm, while the Wi-Fi monitor measures quality in percentage where the dBm is between -90 and -40.

To derive a transformation equation, a scatter plot has been made (Figure 43). The x-axis contains the RSSI of the mobile device in dBm, where the y-axis contains the RSSI of the Wi-Fi monitor in percentage of quality. A line was fitted through these points to calculate the regression. The formula calculated from this line is $0,55x + 60,86$ where x is the RSSI in dBm and the result is the transformation to percentage of quality. However, the low number of measurements may question the equation calculated by the regression. More measurements could give a better indication of the regression.

By using the calculated equation it can be stated that the RSSI scanned by the mobile device is higher than the RSSI scanned by the Wi-Fi monitor when the RSSI is lower than 20%. However, when the RSSI of the Wi-Fi monitor exceeds 20%, the RSSI scanned by the mobile device is lower. Additionally, it should be noted that the difference seems to be linear. The lowest RSSIs of the Wi-Fi monitors result in the highest difference (-13 to -17%) with the RSSIs scanned by the mobile device. After the turning point of an RSSI of 20% by the Wi-Fi monitor where the differences are approximately 0%, the differences between the two measurements will increase again to 16% at the highest RSSI scanned by the Wi-Fi monitor.

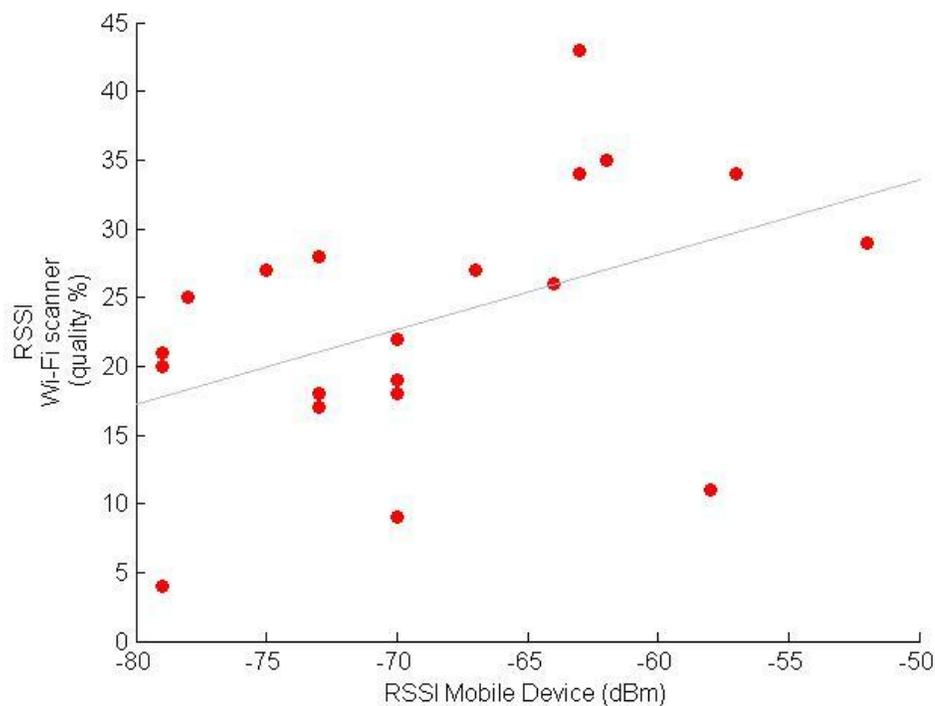


Figure 43: Scatter Plot to derive the relationship between the RSSI measured by the Wi-Fi monitors and the mobile device

In short the answers in the two research questions that were introduced at the beginning of the chapter:

What is the most suitable localization method for DaRen system inside the time span of the project according to the relevant user type and how can this method be implemented in the project?

In this chapter an overview on possible localization methods was given. Localization using Wi-Fi technology was chosen as the most suitable technology for the DaRen application. Wi-Fi in combination with a propagation-based localization method was tested, but not proved sufficient for the application. Instead of a propagation-based method, fingerprinting was the most suitable method for the DaRen application. However, this fingerprinting method has an important 'extension'. Both Wi-Fi monitors are implemented in the existing WLAN-infrastructure in Hubei Provincial Museum. The Wi-Fi monitors do not only contribute to the localization, but also provide valuable information about visitors' behaviour to the museum managers. Besides the implementation of the two Wi-Fi monitors in the fingerprinting method, also the way of localizing is special. The x, y position from the original fingerprinting method is not directly used, but inserted in a database with 9 other x, y positions derived from fingerprinting to minimize the influence of outliers. From these 10 positions the room ID where the user is located can be derived with a high probability that the user is indeed in that room. With this room ID, the user can be provided with extra information on the exhibits in that room.

What are the possibilities of Wi-Fi monitoring in comparison to the commonly known fingerprinting method for the purpose of a 3D indoor navigation system?

In this chapter a different way of fingerprinting has been introduced, namely Wi-Fi monitored fingerprinting method. This method uses rasterized radio maps to calculate the probabilities of the user being at a certain cell in the raster. The minimum value can be derived from the raster which is the location with the highest probability where the user can be located. This method has its advantages and disadvantages compared to common fingerprinting, but the results with only two Wi-Fi monitors are promising for the future and really show the potential of this method.

6. Navigation

The other important component of the DaRen application is navigation subsystem which is responsible for guiding the user from the current location to the user-specified destination inside the Hubei Provincial Museum. The route guidance is preceded by computation of the optimal path from the user's actual location to the point of interest, referred to a map. In general once the user's position is known it is easy to compute the path to his destination and to give directions (*Spasov, 2007*). The navigation system consists of two components: representation of navigable environment and route computation. However, the indoor navigation differs from outdoor navigation substantially as the indoor environment is far more complex. The orientation in the building is complicated by the existence of multiple levels, relatively smaller spaces and the more difficult overview of the entire indoor environment. Moreover, people have the option to move freely within rooms and corridors in contrast to streets. Additionally indoors spaces have many obstacles such as furniture, columns and podiums which can be avoided in various ways. Therefore, the composition of navigable indoor environment is difficult and to be able to perform correct indoor navigation the requirements for every case need to be estimated. The aim of the navigation subsystem is to assure the continuity of the navigation process. This section investigates the possible methods to derive the navigation component for the mobile navigation application for the Hubei Provincial Museum. Thus, the following research questions are raised regarding the generation of the indoor navigation and its performance in the Hubei Provincial Museum:

- What navigation approach is the most appropriate for implementation of navigation services in the Hubei Provincial Museum?
- How to present the relations between spatial units?
- What kind of geometrical and topological information is required?

6.1 Navigation Approaches

To generate the navigation system the grid and network approaches might be used (*Zlatanova, 2013*). These approaches are discussed in more detail below.

Network approaches reflect the indoor environments using topological-based structures which describe the connectivity and adjacency between spatial units (*Afyouni, Ray and Claramunt, 2012*). A topological graph is a system of nodes and edges. Nodes represent specific locations such as rooms while edges stand for the connections between different locations (hallways) (Figure 44). Moving from one node to the other is allowed only when there is a link between them. The link indicates the distance or travel time between nodes. Moreover the node can contain additional information about the location that might be useful to the user i.e. descriptions, other location-related information.

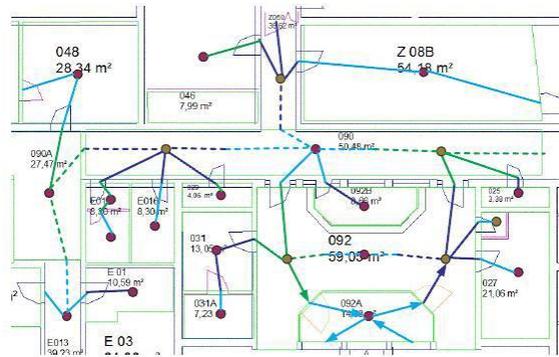


Figure 44: Cell centres and paths overlaid with a floor Plan (Lorenz et al., 2006)

The network can be designed manually or using Poincare Duality, Medial Axis Transformation, Visibility Graph, triangulation, Voronoi diagram or combinations of them (Zlatanova, 2013).

Network approaches are generally considered to be less accurate since they lack geometric details on entities and places indoors. However, network approaches have a high level of flexibility – new data can be easily inserted, modified or deleted to update the network.

Grid approaches subdivide the environment into a finite number of non-overlapping areas, thus creating a partition that covers the entire space (Afyouni et al., 2012). There are two types of tessellations (subdivisions): regular and irregular. Regular grid contains cells that have exactly the same shape and size such as rectangles, hexagons, octagons (Figure 45) while cells of irregular grid differ in shape and size (Figure 46). The main disadvantage of regular grids is that they do not represent the shape (boundaries) of objects precisely due to their fixed shape. Irregular tessellations allow for adaptive decomposition of space to represent the complexity of environment (Afyouni et al., 2012).

Grid approach provides spatial models with continuous geometric properties, supporting different kinds of geometric-based queries as well as cell-level interactions. The main advantage of grid-based approaches is that they allow very precise movement in space. However, the performance depends on the grid resolution and cell size. If the grid is too coarse important information might be lost while overly fine grid might disproportionately increase processing time (Zlatanova, 2013).

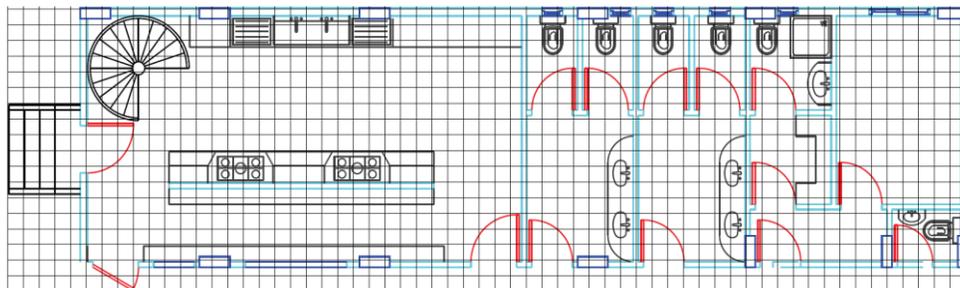


Figure 45: Regular grid (Afyouni et al., 2012).

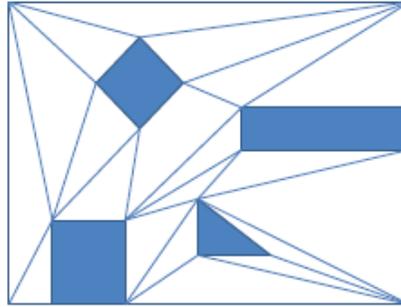


Figure 46: Irregular grid (Afyouni et al., 2012).

The advantages and disadvantages of both approaches are summarized in Table 10.

Approach	Advantages	Disadvantages
Network	<ul style="list-style-type: none"> – Easy to design and maintain – Efficient due to its compactness – High level of flexibility 	<ul style="list-style-type: none"> – Not a high accuracy location data
Grid	<ul style="list-style-type: none"> – Accurate location data – High quality data – Good for navigation queries as geometric information is accurate – Easy to design and maintain 	<ul style="list-style-type: none"> – Consumes excessive amounts of memory and processor time in large spaces

Table 10: Comparison of different approaches, adapted from Afyouni et al. (2012).

The network approach was selected to design navigation subsystem due to several reasons:

- Easy implementation and maintenance is considered an advantage since Hubei Provincial Museum is a large and complex building. Moreover, the implementation time is limited.
- Low data processing time is essential for the navigation as the user is most likely not willing to wait long for the guidance.
- Part of expositions is temporal thus the navigation system should enable easy updates.
- The need for high level of accuracy is limited as it is generally considered that the intelligence and common sense of the user can compensate for minor deviations.

6.2 Generation of the Network

As it was mentioned above to develop the network Poincare Duality, Medial Axis Transformation, Visibility Graph and Voronoi diagram might be used. In order to create the navigation network for the Hubei Provincial Museum, three methods: manual network extraction, topological network extraction using Medial Axis Transformation and geometric network extraction using topological model, were considered.

The navigation network might be drawn manually using CAD software. The advantages of this method are that the developer of the system defines how to avoid the obstacles and which sub-

spaces of the environment can be entered. However, this method is time consuming and provides static network thus the update of the network is complex.

Essentially the Medial Axis Transformation (MAT) is a thinning algorithm which provides the skeleton of the polygon. There are several MAT approaches for extracting a contour line from a polygon. A skeleton of a polygon can be extracted using Delaunay triangulation. One or two edges of the Delaunay triangles are inside the input polygon. Connecting the midpoints of the inside edges of the triangles produces the skeleton of the polygon (*Choi and Lee, 2009*). Another MAT approach derives the skeleton of the polygon using the Voronoi-Diagram. *Lee (2004)* described the method in detail. Using this method firstly Voronoi edges for each vertex of a polygon are generated. Later polygon edges are grouped into chains at the concave vertexes. In the next step Voronoi edges are generated for each chain and merged with each successive chain. Finally, the medial axis is created by merging the final two chains and by removing the Voronoi edges incident at the vertexes of the polygon.

Karas et al. (2006) derived the navigation network automatically using MAT. MAT is be used to generate the middle line of the corridors and determine the middle points of the rooms which are represented as nodes. Room nodes are generated calculating the centre point of the polygons of rooms. Finally, the nodes representing rooms are linked with corridors. The created navigation network is presented in Figure 47. The network is derived automatically and shows the connectivity between spatial units but it lacks the realistic representation of the environment and unreasonable routes might be provided by this network. For example, in some cases in order to guide the person to one room in the corridor the person must reach the end of the corridor where the node is located and come back to enter the room.

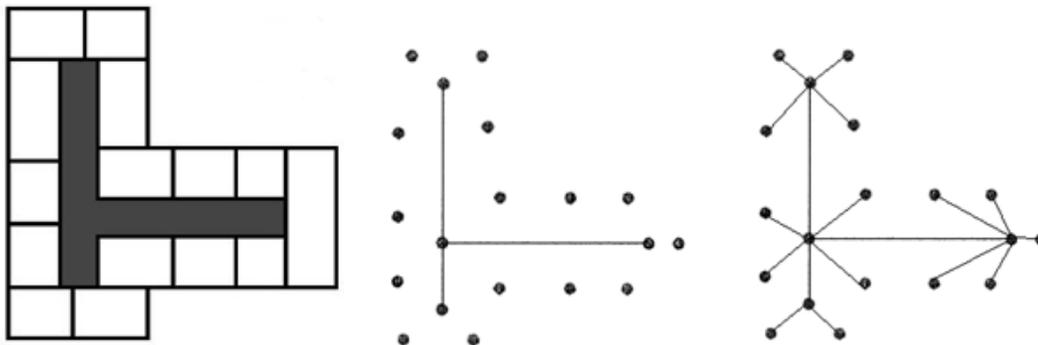


Figure 47: Network created using Medial Axis Transformation, adapted from Karas et al. (2006).

Lee (2004) describes the method where geometric network of 3D indoor environment is created using 3D topological model. In this method a 3D topological building structure is created where nodes represent centre points of rooms and corridors and edges show the relationships: adjacency and connectivity (Figure 48).

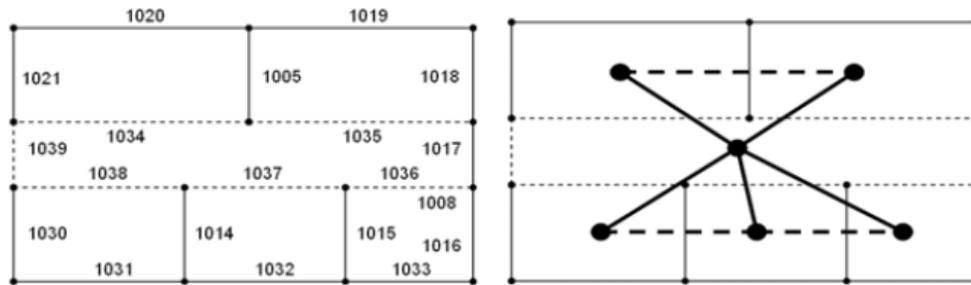


Figure 48: Left: 2D plan. Right: Topological data construction (dash line – adjacency, black line – connectivity) (Choi and Lee, 2009).

When the topological data structure is created the geometric network needs to be built. Firstly, the nodes representing corridors are converted into edges using MAT algorithm. Secondly, all edges from the other nodes need to be adjusted perpendicularly to the corridor edges in order to maintain shortest connectivity in 3D topological data (Figure 49). One main advantage of this approach is that the network is generated automatically. Additionally, this approach shows the navigable environment realistically and provides rational routes. However, this approach is not suitable for spacious rooms as the room is represented only by one node and navigation network of the room is not created. Therefore the navigation might be inaccurate as the obstacles (objects in the room) are not taken into account unless all objects in the room are represented as nodes. Additionally, some issues might appear if the room is a concave polygon, then the centre point of the polygon is placed outside the polygon. Thus, the environment might be represented incorrectly.

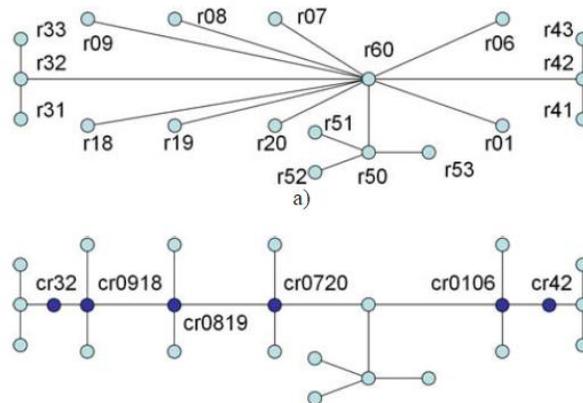


Figure 49: 3D Geometric Network from 3D Topological Data Structure (Meijers, Zlatanova and Pfeifer).

After performing the analysis of possible methods to generate the network, it is decided that concepts of different approaches are going to be combined. Since the Hubei Provincial Museum is a complex building where rooms are spacious and are not shaped as regular rectangles, also some rooms contain open spaces – holes, the true MAT algorithm would be difficult to implement. The MAT might generate unrealistic routes between rooms and inside the rooms. As the rooms are large and part of them is shaped as concave polygons, the Constrained Delaunay

Triangulation is used to derive the network in rooms and corridors. Applying the geometric model of the building the navigable environment is represented realistically. The use of geometric and topologic information can derive the navigation network which represents the indoor environment accurately and provides cost efficient path computations.

6.3 Routing Algorithms

Routing techniques use algorithms that find a particular path. Table 11 presents the most common strategies for navigation.

Path Type	Advantages	Disadvantages
Shortest path	With minimum distance	If obstacles are considered may result in following a difficult to explain path
Shortest time	With estimated minimum travel time	Speed should be considered
Simplest path	Minimum turns	Could be longer distance/travel time
Least-space-visited	Least-number of passed room	May lead more travelling time and distance
Least obstruction	Least degree of blockage	Need accurate information about (dynamic) obstacles
Safest path	Avoid specific areas	May result in long detours
Floor strategy	Better performance of homing users regarding time, similar with Least-Effort path	May not with the minimum distance

Table 11: Routing techniques (Zlatanova, 2013).

However, it should be noted that the most commonly-used strategies are the shortest path and shortest time but for some applications such as emergency applications they might not be essential (Zlatanova 2013). In this project the shortest path algorithm is going to be implemented considering that the user of application has limited amount of time to visit in the museum. The most commonly used shortest path algorithms are Dijkstra and A* which are explained in more detail below.

Dijkstra's Algorithm finds the path with lowest cost (shortest path) from one node to the other. The distance values are assigned to all nodes from the perspective of starting node. All the nodes connected to the starting node are examined and the one with the lowest cost value is selected. The algorithm works by advancing a single node at a time, starting from the source node. At each step the node with the lowest value from the starting node is selected until the destination node is reached. Thus, the algorithm finds the shortest path to the point of interest. However, the Dijkstra's algorithm has one main disadvantage that it explores all the possible nodes ("blind

search”) that might consume lots of time. The Figure 50 shows the Dijkstra’s search area and the dashed line indicates the found shortest path.

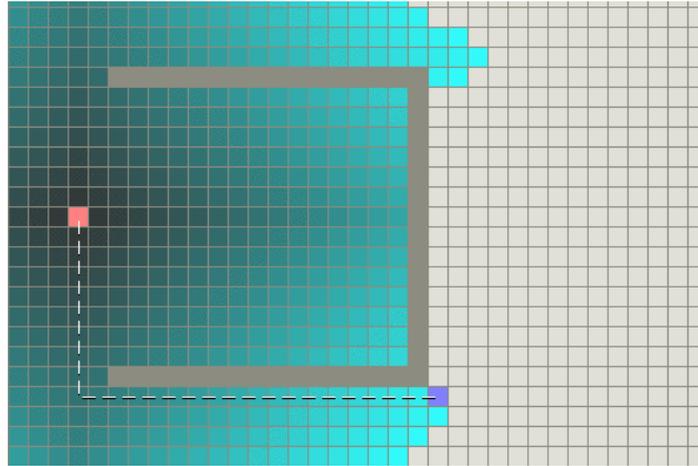


Figure 50: Path with obstacles computation using Dijkstra’s Algorithm (*available at: <http://theory.stanford.edu>.*)

A* Algorithm is another algorithm widely used to find the shortest path. A* algorithm combines Dijkstra’s algorithm and the Best-First-Search. Dijkstra selects nodes that are closest to the source node and the Best-First-Search uses heuristic function for guidance (favouring nodes that are closest to the destination). Therefore, the A* is faster than Dijkstra. However, the optimal performance of the A* algorithm depends on the heuristic model. A* finds the optimal path only if the heuristic function is admissible. Figure 51 shows the A* search area and the generated route. As it can be seen from Figures 50 and 51 both algorithms find very similar paths to the destination but the A* algorithm minimises the search area thus performs the calculations faster.

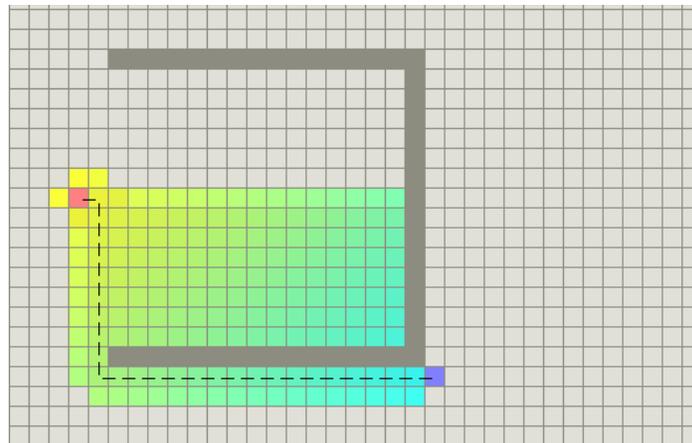


Figure 51: Path with obstacles computation using A* Algorithm (*available at: <http://theory.stanford.edu>.*)

Comparing the two possible the shortest path algorithms, the A* algorithm is chosen for the implementation of this project. The selected algorithm allows fast path computations which are important to ensure real time navigation as the visitor of the museum is not willing to wait.

6.4 Implementation

3D geometric network for building can be created through 3D topological data extraction (*Choi and Lee, 2009*). In order to build navigation network for the Hubei Provincial Museum firstly data for the processing needs to be prepared and then the topology and geometry extraction for the network has to be executed.

In the data preparation phase 2D floor plans of Hubei Provincial Museum are extracted from the existing 3D model. The extracted plans are encoded in CAD file format. However, CAD dataset represents features using lines not polygons which are needed for the network creation. Moreover, this format does not support geo-referencing. For these reasons the CAD drawings are transformed to shapefiles and geo-referenced in WGS84 for further processing. The generated vector shapefiles are uploaded to Quantum GIS software to check data consistency (if all needed spatial units are represented in the dataset) and to eliminate observed errors such as crossing boundaries and overlapping features. In order to extract topology and geometry of the building in an automatic way spatial objects have to be represented as polygons (Figure 52). However, due to the complexity of the floor plans, *lines to polygons* command in QGIS cannot be performed. Moreover, semantic information of indoor objects is lost during the extraction. Therefore, the derivation of navigation network is done semi-automatically. Using the Quantum GIS tools the objects (rooms, corridors, stairs, escalators and doors) are manually drawn as polygons. All object polygons are given meaning with respect to their role in the building. Additionally, object IDs are attached to each polygon. These entire attribute values are entered manually.

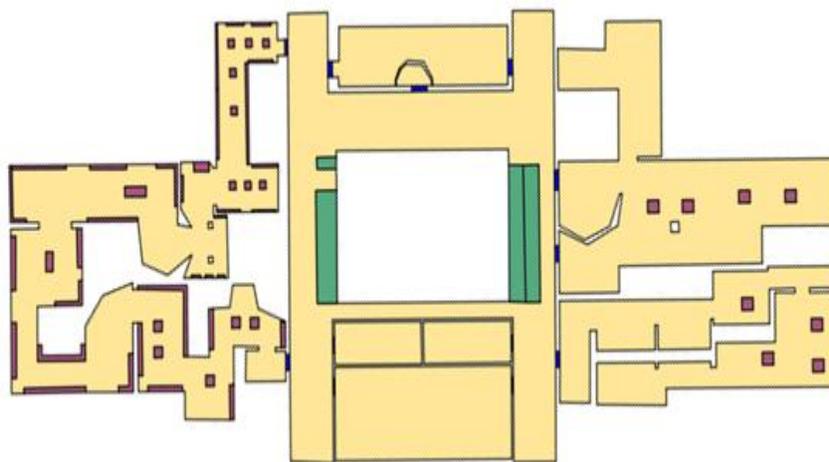


Figure 52: Spatial units: rooms, corridors, stairs, escalators and doors, represented as polygons on the first floor of the museum.

In order to derive the navigation network of the whole building firstly the network of every floor has to be extracted. 3D navigation network to imitate 3D environment can be developed by stacking multiple layers of 2D maps. In such a case the vertical connections such as stairs and elevators between different layers are incorporated. Height information is assigned to nodes representing stairs and escalators (Figure 53).

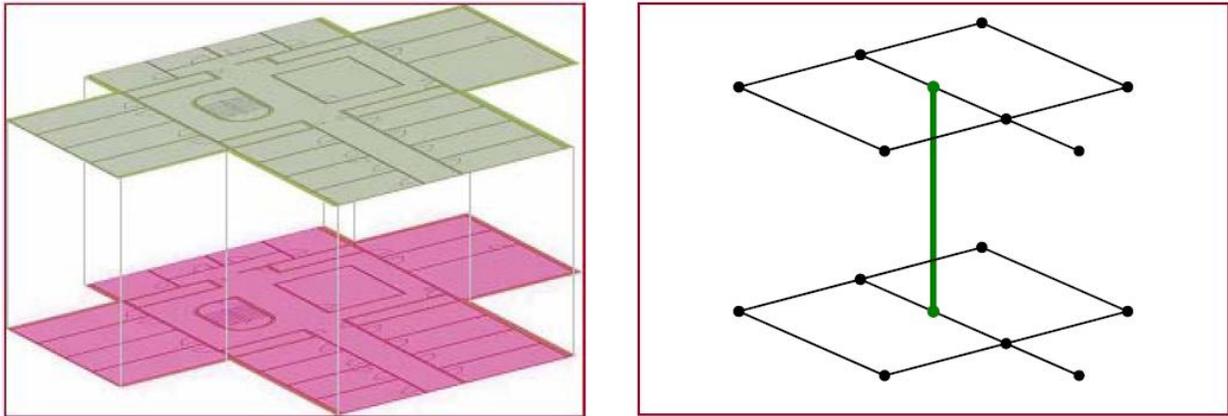


Figure 53: Vertical link connecting two floors (Spassov, 2007).

In order to read shape files *fiona* library is installed in Python. Additionally, to use the geometry of the polygons in the shapefiles the *shapely* library which contains spatial object types and functions is installed. To generate the network of the whole floor, in the first place network of each room and corridor has to be derived. To extract the navigation network of each room firstly constrained Delaunay triangulation is built in every polygon. To perform the constrained Delaunay triangulation another Python library *Triangle* is added as it contains the required triangulation algorithm. Constrained Delaunay triangulation allows decomposing the polygons into non-overlapping triangles. In order to enable fast data processing, the connectivity of nodes is limited. Room node has maximum 3 connections, door node – 2 and exhibit node can have only one link. When the polygons of rooms and corridors are subdivided into triangles, the centre points of every triangle are calculated. In this stage such objects as exhibits are represented as holes and are not taken into account. The exhibits are not considered in this step in order to provide more realistic (straight) paths between the objects. Finally the calculated centre points of the triangles which have common edges are connected (Figure 54).

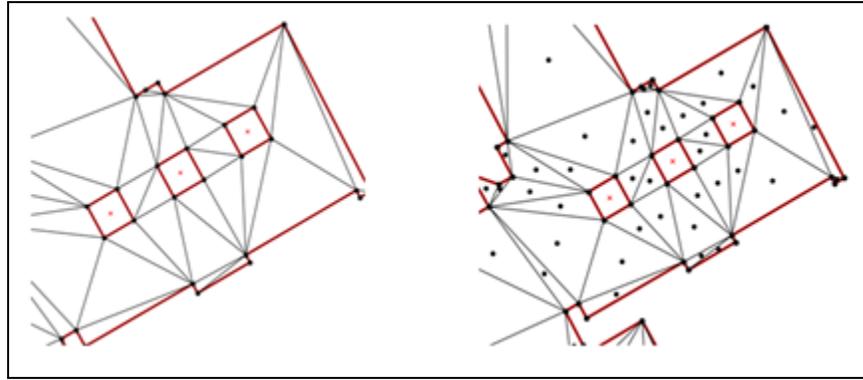


Figure 54: Left: room triangulation. Right: centre points of triangles.

When the separate room networks are created the links between them need to be built. Centre points of door polygons are calculated and linked with the centre point of adjacent triangles (Figure 55).

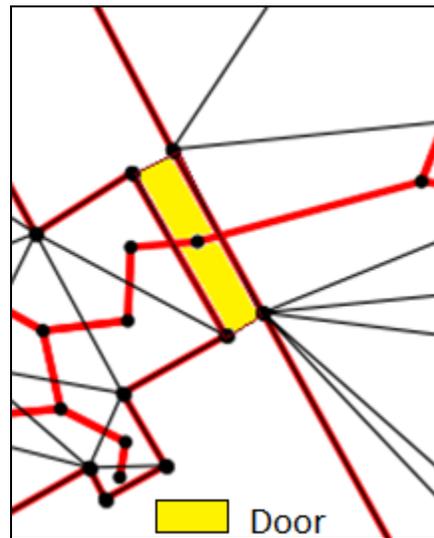


Figure 55: Connection of two individual networks of rooms.

To expand the network the nodes representing exhibits are added (Figure 56). As in previous step the centre point of the exhibit polygon is calculated and attached to the closest point in the existing network. The exhibition nodes contain additional attributes such as name and description of the exhibit.

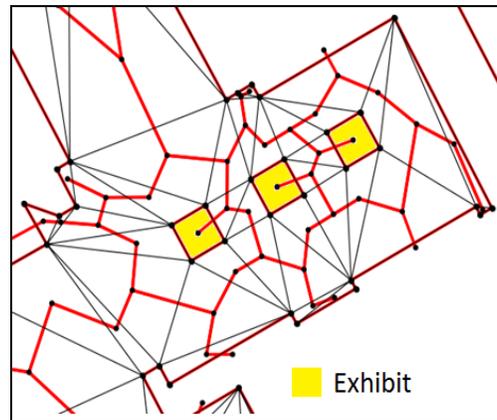


Figure 56: Exhibits in the navigation network.

Due to the time limitations the navigation network only for the first floor of the Hubei Provincial Museum is extracted. The navigable environment is represented by nodes and edges. The nodes indicate spatial units such as rooms, corridors, doors, stairs and escalators while edges represent the connections and distances between them. The database should be organized so that the spatial objects could be recognized by their coordinates and path computations and guidance would be allowed. Therefore, the generated network is stored in four tables in the PostGIS database which supports spatial objects:

- Node table: nodeID, point geometry, roomID;
- Edge table: edgeID, source, target, cost;
- Room table: roomID, polygon geometry;
- Exhibit table: exhibitID, nodeID, name, description.

Node table contains all the nodes of the network with their IDs, x and y coordinates and identifier of room which they belong to. The edge table consists of links between nodes. It provides ID of every edge, start and end point of the edge and cost value which is traveling distance. Room table includes room IDs and coordinates of room polygon. Localization subsystem computes the coordinates of the location of user of the application, according to these coordinates *point in polygon* command is processed to identify in which room the user is. When the room is identified the nearest node of the network in that particular room is set as the starting point of the route. The exhibit table consists of IDs of exhibitions their node IDs and also provides additional information such as name and description of the exhibit. The created tables correspond to the data model introduced in Chapter 4. “Node” table relates to “Network nodes”, “Edge” table corresponds to “Networkedges”, “Room” – “Subspace” and “Exhibit” table corresponds to “Objectinfo”.

The navigation network is generated by running the Python script and is stored in the database. Psycpg2 is a PostgreSQL database adapter for the Python programming language installed to make the connection to the database and store the results of the Python script. In case the floor

plan or model has been changed the script is executed once more and the information in the database is updated.

6.5 Limitations of the generated network

The navigation network is generated semi-automatically, contains semantics and provides the path directly to the exhibit which user is interested in. However, some limitations of this network are observed.

The generated path does not always provide the smooth line although the straight line is expected. For instance, the route leads through the corridor where no obstacles are, thus the straight line segment in the route is expected to appear. However, since the centre points of triangles are not placed on a straight line the jagged path is composed (Figure 57). The jagged path issue can be solved increasing the number of triangles or applying line smoothing algorithm.



Figure 57: The derived route to the exhibit.

The other drawback of the generated network is that the exhibit nodes have only one link. As a consequence, the route to the specific exhibit with not reasonable turns might be provided (Figure 56). This issue might be solved by increasing the connectivity of the exhibit nodes.

6.6 Determining the shortest path

When the generated navigation network is stored in the database, the path from one location to the other can be determined. In order to calculate the route only the connectivity information is needed. The distance is reflected by the cost associated with each link. A* algorithm is implemented which provides the shortest path. A* selects nodes that are closest to the source node and gives priority to the nodes that are closest to the destination. Thus, the computation costs are very low. A path from a source node to a destination node is said to be the shortest path

if its total cost is minimum among all paths (*Spasov, 2007*). The path calculations are performed in PostGIS database as pgRouting extension for various paths computations is installed.

The start node of the route is determined either by location subsystem and the end node is defined by the user of the application. When start point and destination node are known the following general query is processed:

```
SELECT seq, id1 AS node, id2 AS edge, cost
FROM pgr_astar(
    'SELECT id, source, target, cost, x1, y1, x2, y2 FROM edge_table',
    4, 1, false, false
);
```

Where: id is identifier of the edge

source is identifier of the source vertex

target is identifier of the target vertex

cost is the edge traversal cost

x1 is x coordinate of the start point of the edge

y1 is y coordinate of the start point of the edge

x2 is x coordinate of the end point of the edge

y2 is y coordinate of the end point of the edge

The full used path computation query is provided in the appendix B. The result of the routing analysis is an ordered list of nodes with x and y coordinates (Figure 58). The result array is sent to the mobile device and on the mobile device text file is created to visualize the route.

	st_x double precision	st_y double precision
1	5033.9702148	-4774.890136
2	5033.3676757	-4776.368815
3	5031.7226562	-4776.162109
4	5031.1023763	-4775.189127
5	5030.3977864	-4775.384765
6	5028.7545387	-4775.161511
7	5026.6464913	-4775.560514
8	5026.7193036	-4775.687815
9	5027.1157624	-4776.267002

Figure 58: The result of path computation.

As the literature review and previous research study show that the most appropriate navigation approach for the Hubei Provincial Museum is network representation, the dense context aware navigation network for the first floor of the museum is created. The network is generated semi-automatically where indoor objects such as rooms, doors and exhibits are represented as nodes and linked together according to their connectivity and adjacency. A high number of nodes allows finding reasonable route to the destination and the implemented A* algorithm allows finding the shortest path ensures the high performance of the navigation component of the DaRen application.

7. Visualization: Graphic User Interface

The graphic user interface (GUI) is essential component of human navigation system. According to many guides of user interface design, user requirements are considered to be one of the most critical issues to be taken into account when designing a user-friendly application interface (Stone, 2012). A GUI of a navigation system usually consists of three parts: main menu, destination entry and map view. Apparently the map view is the most important component of the GUI. The research of Fallah et al. (2003) states 2D maps are commonly used to represent indoor environment. However, the 2D map shows many restrictions when navigating in a very complicate indoor environment like Hubei Provincial Museum. Therefore this research aims to explore the 3D visualization of indoor space. But there are challenges to visualize the 3D model on mobile devices due to the limitations of mobile display (Junghare, Thakare and Dharaskar, 2011). Thus the following questions are discussed in order to implement an efficient and effective GUI of navigation system:

1. What are the requirements of graphic user interface for the navigation system of museum?
2. Why is 3D visualisation necessary for the navigation in museum?
3. Is it preferable to visualise the whole model of the museum or parts of it every time? If parts, then how should the model be divided?
4. What is needed in order to visualise the 3D model on a mobile device? What technologies and mobile platform should be chosen?

7.1 Graphic User Interface Requirements

7.1.1 General Graphic User Interface design criteria

A sufficient level of knowledge and understanding of user needs is required to design a user-friendly application interface (Lauren, 2012). ISO 9241 is particularly useful as it presents the ergonomic requirements for designing software. According to the part 10 of the mentioned standard, there are seven dialogue principles, which represent the dynamic aspects of the interface and can be regarded as the "feel" of the interface:

1. Suitability for the task: the dialogue is suitable for a task when it supports the user in the effective and efficient completion of the task.
2. Self-descriptiveness: the dialogue is self-descriptive when each dialogue step is immediately comprehensible through feedback from the system or is explained to the user on request.
3. Controllability: the dialogue is controllable when the user is able to initiate and control the direction and pace of the interaction until the point at which the goal has been met.
4. Conformity with user expectations: the dialogue conforms with user expectations when it is consistent and corresponds to the user characteristics, such as task knowledge, education, experience, and to commonly accepted conventions.
5. Error tolerance: the dialogue is error tolerant if despite evident errors in input, the intended result may be achieved with either no or minimal action by the user.

6. Suitability for individualization: the dialogue is capable of individualization when the interface software can be modified to suit the task needs, individual preferences, and skills of the user.
7. Suitability for learning: the dialogue is suitable for learning when it supports and guides the user in learning to use the system.

In order to design the GUI of DaRen mobile navigation system, the principles of ISO 9241 are applied. The GUI operates on a mobile platform and avoids any unnecessary complexities that require excessive time to learn how to use the system. In addition, GUI is able to display user's current location as well as the direction to the destination. Moreover, GUI enables the user to type their destination or select it from directories of places in the museum. It also allows the user to change and reselect their destination. Besides, the information (pictures and texts) of exhibits can be viewed via the interface. Finally, the GUI should display the location and routes on a map properly and allow basic interactions with the map such as move, rotate, zoom in and zoom out.

7.1.2 Map View

The map view is the core part of GUI of the navigation system. Therefore, to find the most appropriate representation of the map is vital to the success of GUI design. In general there are two types of map: 2D map and 3D map. In previous navigation applications, 2D maps are commonly used since they are very easy to generate and they provide simple implementation of routing. However, there are apparent limitations of 2D maps when the indoor environment is complicated and the vertical structure needs to be considered. Thus 3D maps are used, to overcome the limitations of 2D representation in complex environments (*Fallah et al, 2003*). The 3D map provides more detailed representation of indoor space by including the location of various objects such as doors, furniture and stairs between floors. However, representing detailed environment carries its costs in terms of storage and processing power required for these thus making them computationally expensive. Additional mean of representing environment is provided by a hybrid model, where both the 3D model and the 2D map can be represented. The 2D map and 3D model are linked to allow the user to switch the map view according to their preference at any given time. University of Technology Vienna has carried out a test of presentation forms of indoor navigation. The test results show that floor plans are found acceptable by 82 percent of the participants as "supporting navigation by finding the way immediately", while animated 3D graphics had only 55 percent acceptance level (*Meng and Reichenbacher, 2005*). However according to the survey carried out in this project (introduced in Chapter 2), majority of respondent prefer the 3D view (54%) and other part of respondents prefer 2D map (46%) as the navigation environment of museum.

Besides user's preference, there are more arguments to support that the 3D model should be used to represent navigation environment of the Hubei Provincial Museum. Firstly, the museum has multiple floors with a number of theme exhibition rooms (Figure 59), the 3D model provides visitors with better understanding of the vertical paths linking different floors. Secondly, the 3D model can provide visitors with information on specific locations of exhibits. For example, 3D

model is necessary to guide visitors to various exhibits inside the show room (Figure 60). Lastly, 3D model allows more semantic annotations to be presented, thus it is possible to provide additional information on the items on the exhibit.

Another important reason for using 3D model as the navigation environment of Hubei Provincial Museum is that the 3D model of Hubei museum is available and is free to use. With all advantages mentioned above, 3D visualization is considered to be necessary in this application.



Figure 59: 3D model of Hubei Provincial Museum.



Figure 60: Inside of the Hubei Provincial museum.

In addition, it should be noted that the combination of 2D and 3D maps provides the most opportunities and therefore can be considered to be the ideal solution to represent the indoor navigation environment. However, due to external restrictions and time limitations, the hybrid representation is not going to be implemented.

Since the 3D model of the museum is going to be used to represent the navigation environment, the analysis how to represent the 3D model on a mobile device is performed. The possibilities to visualize the whole and part of the 3D model on the mobile device are tested and the loading time

is recorded. Unity3D game engine is used to perform the test on the HTC Desire X smartphone (more details about Unity3D are introduced in the later section). Test results are presented in Table 12 which shows that the whole model of museum can be visualized successfully on the mobile device but the loading time is up to 9 seconds while to upload one floor takes 3 seconds. Considering that the goal of this application is navigating the visitors of museum who want to know their location immediately, 9 seconds are too long to wait. Therefore it is decided to visualize a part of museum rather than the whole model at every time.

Model	Format	Size	Time	Rendering
Museum 1 (the whole model)	.dae	40.9 mb	9~10s	successfully
Museum 2 (second floor)	.dae	13.3 mb	≤3s	successfully

Table 12: The test of loading model on a mobile device.

The view mode of the 3D model needs to be decided in order to know how to subdivide the model. The 3D map is able to provide the user two views: bird's eye view (flying level) and pedestrian view (walking level) (Figure 61) (Meng and Reichenbacher, 2005). The study of Radoczky (2004) researched user preferences of different view modes. Overview maps for showing the whole route and information about distance and estimated time of walking have been found necessary. An acceptance level of 64 per cent of all participants has been reached who indicated that while using bird's eye view they were "always sure to be on the right way". In another study (Coors et al., 2003), the flying mode was found much easier for navigational purposes in a 3D environment. The same results of the survey performed in this projects also shows that 64 percent of visitors prefer the bird's eye view mode.

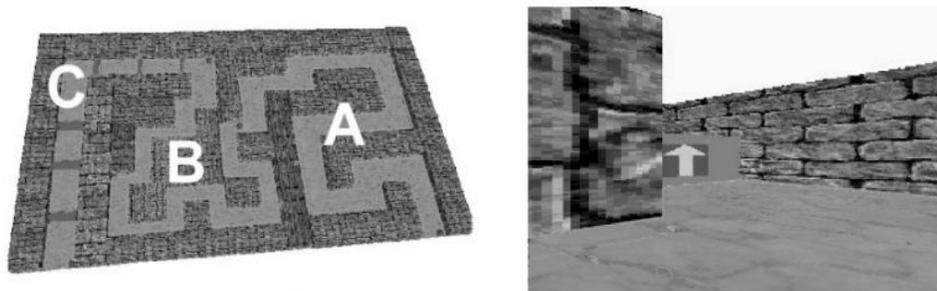


Figure 61: Left: bird-eye view and right: pedestrian view (Zhai Shumin, et al, 1999)

The previous researches demonstrate that the bird's eye view is most accepted by users and its main advantage is enabling the user to have a clear view of the whole route. It is decided to divide the model into different floors and visualize one floor per time; it is the most optimal

solution as performed test proves that the loading speed of one floor model on mobile device is very fast.

7.2 Mobile 3D Visualization

Compared to other devices, mobile devices have many restrictions to consider when performing effective and efficient 3D visualization (*Fonseca et al., 2011*):

1. The characteristics of displays are limited and restrictive due to smaller size, different width/height aspect ratio, lower resolution, limited range of colours, etc.
2. Onboard hardware, including the CPU, processing memory and graphic hardware are generally less powerful and have limited processing power.
3. Connectivity is slower, affecting interactivity when a significant quantity of data is stored on remote databases.
4. There is a lack of powerful, high-level graphic libraries (currently available tools tend to be low level or limited).

Due to these restrictions, it is important to find an efficient and effective method to realize the visualization of 3D model on mobile device.

7.2.1 Technologies of 3D Mobile Visualization

There are two approaches to visualize the 3D model on mobile platform: local and remote visualization. The former method relies on local device resource to display the 3D scene, while the later method uses remote hardware to render the scene to be displayed on the mobile device (*Lamberti and Sanna, 2007*).

OpenGL ES and Mobile 3D Graphics (also known as M3G) are the most widely used mobile graphic application programming interface (API) and can visualize the model locally. OpenGL ES is a royalty-free, cross-platform low-level API for full-function 2D and 3D graphics on embedded systems. M3G is a high-level API for Java mobile 3D graphics and it is designed to provide an efficient 3D graphics API suitable for the J2ME platform (*Chehimi, Coulton and Edwards, 2006*). The developer can implement their own 3D visualization functions based on these two API. However these APIs require the developer has advanced programming skills and experience on using graphic library.

Another possible method to locally visualize 3D model is using game engine. Fritsch and Martin (*2004*) state that game engines are powerful to visualize texture mapped 3D objects, scenes and 3D worlds in real time by efficiently using rendering pipelines, special data structures and speed-up techniques. Usually the game engines also include the 3D graphics libraries mentioned before but they have more functions. The common components of a game engine are list below (*Al-Najdawi, 2007*):

- A graphics module for 2D or 3D

- A physics module
- A collision detection module
- An input/output module
- A sound module
- An artificial intelligence module
- A network module
- A database module
- A Graphical User Interface module (GUI)

The high rendering performance of game engines makes it be increasingly used in various fields for visualization, such as real estate (*Miliano, 1999*) and route instructions (*Kray et al., 2003*).

Nevertheless, remote methods can also be used for mobile visualization. Especially driven by the technologies of HTML5 and WebGL, web 3D visualizations are becoming more popular recently. WebGL is a JavaScript API based on OpenGL ES 2.0 for rendering 2D and 3D graphics within Web browsers without plug-ins. It is exposed on webpages through HTML canvas elements. However this method lacks in supporting very complicate 3D objects' visualization and has high requirements for mobile devices.

With comparisons given above the game engine was chosen to visualize the 3D model on mobile device since its high performance and ease of implementation.

7.2.2 Game Engine

As introduced before, the game engine is used in this research. To find the most suitable game engine to visualize the 3D model on a mobile device, several game engines are studied and compared below. An analysis and comparison of several different game engines in terms of their suitability for visualizing 3D model on mobile device is presented in Table 13.

According to the comparison of 4 game engines (Table 13), it should be noted that Unity3D is one of the most powerful 3D engines. It support most of 3D formats and can be scripted in multiple programming languages. There are also sufficient tutorials of Unity3D available online. Despite the fact that Unity3D is not open source software, the free version of it can be used for non-profit purposes. To conclude, comparison of different engines provides definite proof that the Unity3D model should be chosen due to its superior processing power, ease of scripting and availability of supporting materials (i.e. tutorials).

3D Engine	Description	Platforms	3D formats	Scripting	Advantages	Disadvantages
Unity3D	Unity3D is a game development ecosystem: a powerful rendering engine fully integrated with a complete set of intuitive tools and rapid workflows to create interactive 3D content; easy multiplatform publishing; thousands of quality, ready-made assets in the Asset Store and a knowledge-sharing Community.	iOS, Android, windows phone 8, blackberry	.FBX, .dae, .3DS, .dxf,.obj, .Max, .Bl	C#, JavaScript, Boo	multiple platform, high performance, good tutorials	commercial software pro version is charged
jPCT-AE	jPCT-AE with its easy to learn API offers the developer all the features needed to write a cool looking, mobile 3D game, simulation or business application for Google's Android platform in a short time. jPCT-AE is light weight and only loosely coupled to the OpenGL context. It does not force the developer into a specific code structure and can be used in any Activity.	Android 1.5 +	3DS,OBJ, MD2, ASC	Java	open source, optimized for the Android	lack tutorials
Minko	Minko is a free and open source real-time 3D framework created by Aerys. Minko is designed to create fast and robust 3D applications for the web, desktop and mobile devices.	Android 2.2+, iOS,	.dae .obj, .pc2	ActionScript	open source, good tutorials, and many examples	scripting in ActionScript
Libgdx	Libgdx is a cross-platform game development library written in Java, with some JNI code for performance hungry sections. It abstracts away the differences between writing desktop, Android, iOS and HTML5 games based on standards like OpenGL ES/WebGL. Applications can be prototyped and developed entirely on the desktop, then only 6 lines of code are needed to run the application on Android or HTML5.	Android 2.0+, iOS	.obj MD5	Java	open source, the application can target at different platforms	supports less 3D formats

Table 13: Comparison of 3D Engines.

7.3 Implementation of Graphic User Interface

Figure 62 depicts the main modules of the GUI and how they are connected. The modules in orange and blue colours are designed using graphic layout tool inside Android application. The implementation of Map module is more complicated and is firstly developed in Unity3D and after that integrated with Android application.

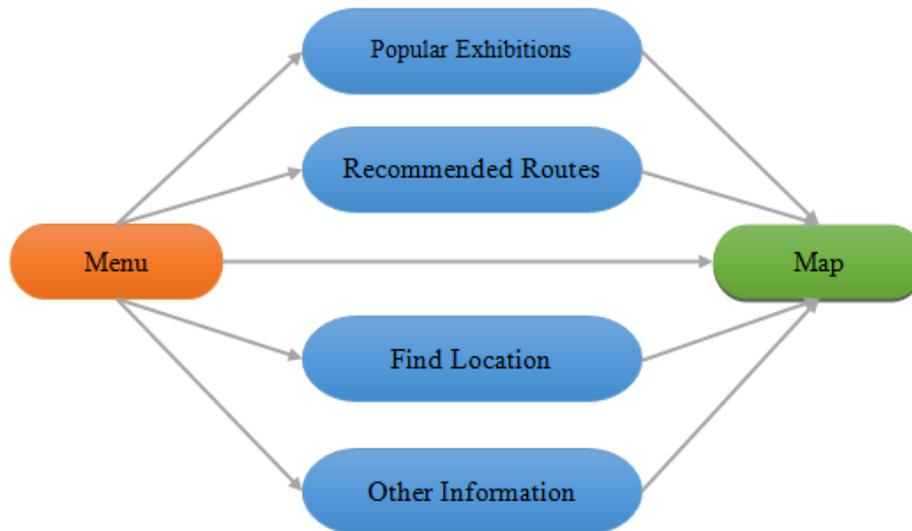


Figure 62: GUI state map.

7.3.1 Interface of DaRen application

A new layout can be created by adding a new Android xml file and setting its root element as layout. For each module, at least one layout is created. Firstly in the menu (Figure 63, left), five buttons were added to provide options: Popular Exhibitions, Recommended Routes, Find Location, Map and Other information. These functions can be started by clicking the button click and user can go to a new module.

“Popular Exhibitions” (Figure 63, centre) shows a list of most visited exhibitions. By selecting the picture of an object, the user is directed to the map view (Figure 63, right) where the user’s current location and the shortest path to the interested object are displayed. The user can also go back to the main menu by clicking the top right button.

“Recommended Routes” provides several visiting routes recommended by the museum. The map view is presented after selecting a certain route. The user is first guided to the start point of the recommended route then he or she can follow the pre-defined route.

“Find Location” enables the user to search or select a place they want to go such as exhibitions, washroom, and exit. The path to the selected location is visualized on map view.

”Other information” gives non-spatial information of the museum such as the opening time of museum and activities in museum. However, the last three modules are not finished yet due to the limitation of time and information.

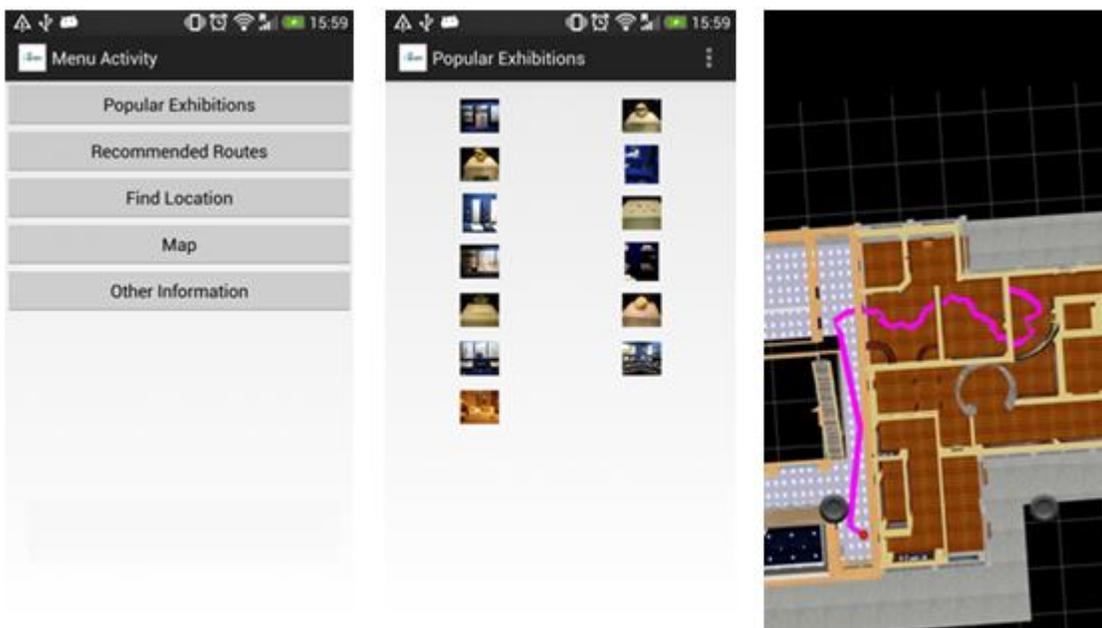


Figure 63: Left: main menu. Centre: popular exhibitions. Right: map view.

7.3.2 Map view

The visualisation process is explained in Figure 64.

Import of 3D model and scene creation. The 3D model of museum is stored in database, but due to the restriction of time, the model used for visualization is still a COLLADA file (.dae). The model files can be easily imported by dragging the files into the “Assets” folder in project window of Unity3D. The original imported model is only first floor of the museum. In order to create bird’s eye view, the ceiling and outer walls of the model are removed. In the next step camera (represents the view point of the user) and controls of camera are added to allow the user to move and rotate the model. The camera is placed above the model with perspective projection. The controls of camera were based on two black buttons in Figure 65: the left one can move camera horizontally and the right button can rotate the camera. Besides a red sphere is added to represent the user’s current location. 3D scene for navigation is built by finishing these steps.

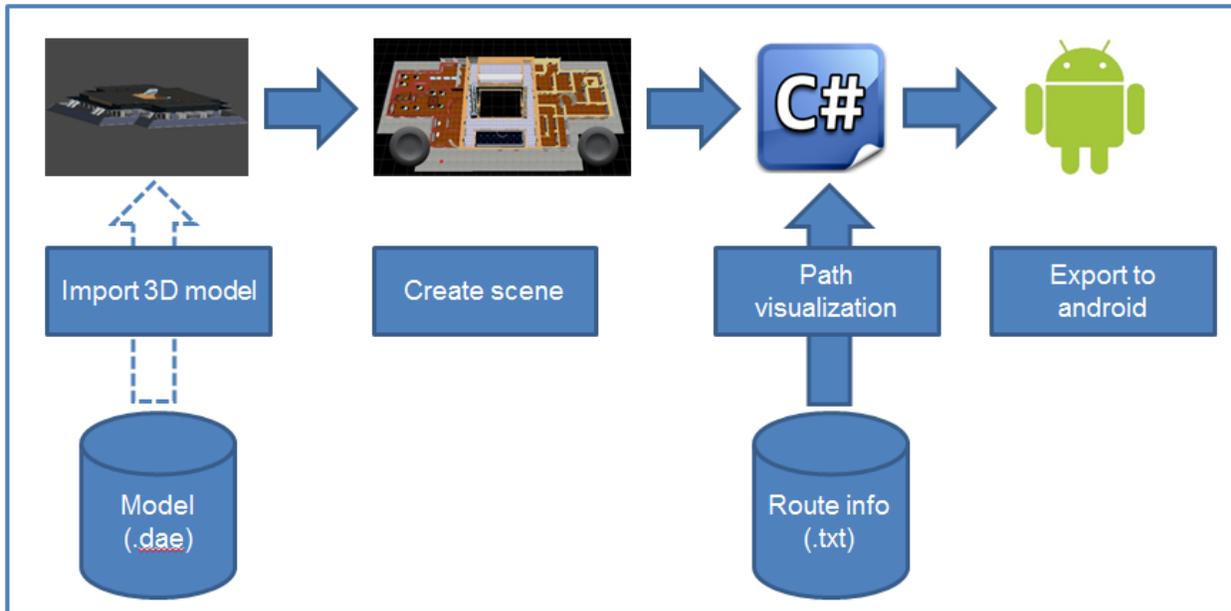


Figure 64: Implementation of map view.

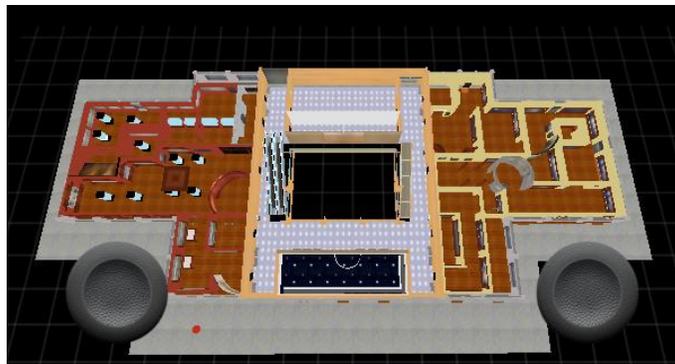


Figure 65: 3D scene in Unity3D.

Route rendering. To visualize the user's location and the route dynamically, a behaviour script needs to be written. One of the main advantages of Unity3D is supporting uncluttered, straightforward and fast scripting in multiple languages. In this project a C# script is created to control the location of the red sphere (represents the user's location) and draw the line of the calculated path. The route information is queried by web service from database and written into a text file dynamically. The text file contains a series of x , y coordinates of route nodes (the first two coordinates are the user's location). The heights of these nodes are uniformly set as a little higher than the floor. Since the coordinate system of the model is different from the Unity3D, the input coordinates have to be transformed to the local coordinate system by using the formula below:

$$x = x' \cos t - y' \sin t + x_0$$

$$y = y' \sin t + x' \cos t + y_0$$

where: x and y are coordinates in Unity3D system,

x' and y' are coordinates in original system,

x_0 and y_0 are coordinates of the origin of 3D model system in Unity3D system.

Later using “LineRenderer” module of Unity3D the route is drawn. To put the sphere in the user’s location “transform.position” command is used. In order to ensure the path and user’s location can be updated dynamically, it is necessary to write the codes for the two mentioned steps under the “Update ()” function of the behaviour script. An example of path rendering is given in Figure 66.

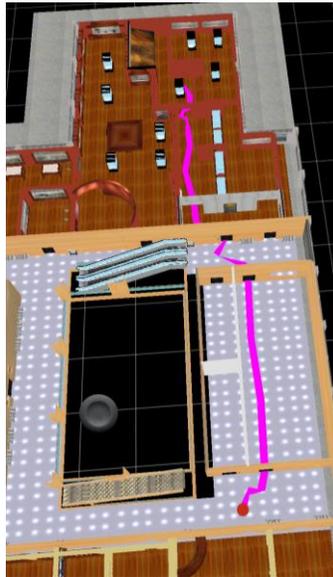


Figure 66: Visualization of path.

Export to android. The first step is to build the program in Unity3D, then export it to the main android project as a library and move the project assets of Unity3D to the assets of android project. Besides, the class of “androidplayer” is also imported as a library. As shown in codes below the map view activity class should extend UnityPlayerActivity.

```
import com.unity3d.player.UnityPlayerActivity;

public class MapViewActivity extends UnityPlayerActivity{
...}
```

By this way the Unity3D activity can be triggered automatically when starting the activity

To sum up it can be stated that based on the former research user friendly GUI is designed based on user requirements. The 3D model is considered to be the optimal representation of navigation environment of Hubei Provincial Museum since it is the preference of users and can provide more information about indoor space. However due to the loading speed of 3D model on mobile

device and the fact that most users prefer the birds' eye view mode, the one floor of the museum instead the entire 3D model is visualized at once. Unity3D game engine is proved to be one of most powerful game engines to support 3D mobile visualization with high performance and ease of implementation.

Although the 3D model is visualised on the mobile device successfully some limitations are observed. Currently the model is stored locally on the mobile device which may encounter issues when the model is too large or multiple models are needed because of the limitation of mobile storage. Hence loading the model on runtime is recommended. There are two ways to load the model on runtime for Unity3D: use Asset Bundles function of Unity3D pro and write a model loader where the geometry and materials of model can be loaded via URLs. Another drawback is the little user interaction; more interactions with the 3D model need to be added. Last but not least some annotations could be added to the model such the name of each exhibition room and simple introduction about it to enrich the model.

8. Verification and validation

The DaRen application was made operational on the 10th of October 2013. The application is tested multiple times at the Hubei Provincial Museum. In Figure 67 the results of three separate navigation operations are depicted. Clearly visible is the 3D model in bird's eye view with the pink line being the navigation route to the final destination. The green circles indicate the actual location of the DaRen application user at the time of the navigation request. The red dot represents the calculated location of the user.

It is noticeable that the starting point of the navigation is in most cases in the same room as the user is located. However it seems that the navigation does not always start at the node that is located closest to the user. Also the user has to wait about five seconds before the starting point is determined due to the multiple locations acquired by the system to determine the starting point of the navigation. Although these issues are present, they do not seem to influence the navigation experience negatively.

The navigation route visualized in the 3D viewer is the actual shortest path calculated by the application. The visualization of the path is according to the network which is automatically extracted. Therefore the line is not straight as expected in a navigation system.

The visualization of the 3D model is very clear and it is relatively easy to find your location in the 3D model using recognizable features. Although the panning and rotating in the 3D viewer is not as convenient as wanted, it does give the necessary options to adjust the view to the users preference. Also a zoom-function would increase the movability and therefore also the performance of the visualization of the 3D model.

However, despite some minor flaws the application can be used as a navigation application. Visitors can find their way with the DaRen application and that is the ultimate goal of this project. Also, it is possible for the museum to derive information from the Wi-Fi monitors to get an insight on the usage of the exhibitions and the museum in general.



Figure 67: DaRen application.

9. Privacy Issues

The 3D indoor navigation system DaRen falls in the category of location-based services (LBS). LBS uses the user location to deliver a service. Privacy is a serious issue for LBS and it can have a great impact on the success of these services. An example of this, are the recycling bins of the company Renew (*Datoo, 2013*). This company installed recycling bins with screens in the city of London that can show advertisements to people that are close. In some of these bins Renew has installed tracking devices to detect and track mobile devices, so that the advertisements to be personalized for each mobile device. The problem with this system was that MAC-addresses are tracked without the consent of the user and therefore the tracking system in the bins was not allowed by the city of London.

Besides the legal concerns, there are also the user's concerns about their privacy, such as who receives this location information, for which purpose and for how long is it used and stored (*Xu et al., 2009*). For DaRen project, it is important to investigate both legal and user concerns so that users' privacy to be protected. So, that then these concerns can be taken care off in the system, if they apply. In this section, first privacy is discussed where both user concerns and legal concerns are addressed. After that, it is investigates whether these concerns are applicable for the DaRen project and if these concerns are applicable how these concerns should be taken care off.

9.1 Privacy

Often, privacy is described as the right to be left alone (*Warren and Brandeis, 1890*). This right is also been recognized by the United Nations as a fundamental right in The UN Universal Declaration of Human Rights (*Loenen and de Jong, 2007*). There are different kinds of privacy. In this case information privacy, in which information is regarded as personal data, is on interest. Personal data is regulated by the privacy laws of each country.

9.1.1 User concerns

Lot of research concerning the privacy concerns of users using a LBS has been conducted. Those studies reveal that users are concerned about loss of privacy when using LBS (*Xu et al., 2009*). The different concerns can be categorized in four groups: collection, unauthorized secondary use, errors and improper access. Collection reflects the concern about the extensive amount of personally identifiable data collected and stored in databases. Secondly, unauthorized secondary use is about the concern that information is collected from individuals for one purpose but is used for another secondary use without consent. Errors reflect the concern that protection against deliberate and accidental errors in personal data is inadequate. Improper access is the concern that data about individuals are readily available to people not properly authorized view or work with data. The abovementioned concerns should be considered in the design and implementation of the application. Especially the proper communication of privacy issues related to the application should be of primary importance. Users should be aware that their privacy is protected.

9.1.2 Legal Concerns

In many countries the privacy concerns are addressed by applying regulation. This means that regulations can differ per country. Therefore, in this subsection the current legislation current legislation regarding protection of privacy in the Netherlands and China are investigated.

Netherlands. According to the Dutch law information is personal when it can identify or has a possibility to identify a specific person. If information is personal, regulations should be applied on this information as described in the "Wet Bescherming Persoonsgegevens" (*WBP, 2000*). According to this law the user should give prior consent in order for his personal information to be used. Furthermore, it should be stated which information is used and for what purpose. Also, it is not allowed this information to be later used for other purposes or to be sold to a third party without consent. Finally, personal information is only stored till the point that information does not serve its purpose anymore and the stored information should be sufficient protected to prevent people from gaining unauthorized access.

Further, this year the Article 29-working group (*Article 29 Working Party, 2013*) which are the privacy supervisors for the European Union gave their opinion on the area of privacy and mobile applications/LBS. They provide mobile application developers with some obligations and recommendations. Most of the obligations are more-or-less implemented in the Dutch Law. On the other hand, the recommendations such as informing the public when there is a data leak and use in the privacy condition information special for European users, are more interesting.

China. In China there is currently no specific legislation for protecting personal information. Only a guide of personal information protection was published in November 2012 (*Shi, 2012*), which became a national standard since February 2013 (*Zhang, 2013*).

This guide divides the personal information into personal general information and personal sensitive information. The personal sensitive information is the information that can lead to harmful effect on the person when it is revealed or changed, such as the ID number, phone number and fingerprint. The personal general information is all other personal information excluding sensitive information. According to this definition, the writer thinks the anonymous personal location information belongs to the personal general information, which is allowed to be collected and processed under tacit or expressed consent of the information source. That means that if the person does not explicitly express his or her dissent on being tracked, then his or her location can be recorded and used. The guide also gives the general rules to use personal information, which are listed below:

1. *Clear purpose:* The purpose of processing the personal information must be clear, specific and reasonable. It's not allowed to change the purpose without informing the subject of personal information.
2. *As less as possible:* The least personal information should be collected and processed for the purpose clarified and the information should be deleted as soon as possible.

3. *Public informed*: The administrator of personal information should inform people about the purpose of processing the personal information, how the data will be collected and what measures will be taken to protect the data.
4. *Consent*: The information must be processed only with the consent of the subject of personal information.
5. *Quality*: The personal information should be correct, secured and updated.
6. *Security*: Measures should be taken to protect the security of personal information, such as preventing unauthorized retrieval, reveal and change of the information.
7. *Loyalty*: The administrators of personal information should comply their commitment that they made when collecting the information, and shouldn't use the data for other purposes which are not clarified.
8. *Responsibility*: The responsibility of each party in processing the personal information should be clarified and the process should be well recorded.

Two cases of using personal location information in China are discussed below. Case 1 is a research carried out by a software company and case 2 is a mobile app widely used in China.

- *Case 1, monitoring traffic flows via mobile phone signal*: In this case the researchers monitored the traffic flow by tracking the passenger's mobile phone signal. However, the way that they deal with the privacy issues of their research is not mentioned (*Cheng, et al., 2012*).
- *Case 2, WeChat*: WeChat is a very popular chatting mobile application in China. Two functions of this application need to access the location data of the user. The first one is "shake" which aims at finding the people who are shaking their phone, either right next to the user or across the globe. The second one is "look around" which is to find people using WeChat with Look Around enabled nearby. Both functions ask for a permission to access the user's location data when they are initiated (*WeChat, 2013*). Moreover, the service agreement of this application points out that the application company has responsibility to protect the personal information and is not allowed to reveal the information to others without the user's permission (*Tencent, 2013*).

Finally, after studying the regulations and former cases about collecting and processing personal information (location information) in China, a conclusion can be reached that it is allowed to collect the anonymous location information in China with consent from the source of information and security guarantees of the information.

9.2 Privacy concerns DaRen

The first step is to identify the occurrence of personal information in the DaRen system in order to find out if data protection regulation has to be applied on the system. Since the main difference in the privacy regulation of the Netherlands and China is that in the Netherlands the implementation is regulated by law while the substance is more-or-less the same, the Dutch privacy regulation is used to detect the occurrence of personal information. As it has already

mentioned, according to the Dutch law information is personal when it can identify or has a possibility to identify a specific person. Normally in LBS during the localization or tracking of a mobile device, personal information is collected as the MAC address of the device is scanned. In DaRen project the user does not have to provide any personal information while using the mobile phone application. However, the localization component includes two sub-systems. The sub-system that provide the museum visitors' location in the application does not make use of the MAC address but the sub-system that provides the information to the museum managers makes use of it and therefore the data falls under the data protection regulation. In case that the two sub-systems will be integrated into one, the implementation of privacy terms and conditions would be indisputable. The application should ask for the user's permission before being installed on the device and the purpose of collecting the MAC address of the device should be clearly stated. Moreover, personal data should be sufficiently protected from unauthorized use.

10. Conclusions and Future Work

10.1 Conclusions

The output of the project is a 3D indoor navigation application for the Hubei Provincial Museum which provides the real time location of the user based on Wi-Fi technology and the route to a final destination in a 3D model visualized on the mobile device. In the process of achieving this ultimate goal, the possibilities of indoor navigation system on mobile devices were investigated. Separate objectives were set for each one of the components constituting the system. Firstly, possible solutions were identified and selected and secondly the components were developed and integrated with each other. In order to verify that the objectives have been met tests were carried out at the final product.

The main conclusions that came out of the project results are discussed below:

1. The created application fulfils most of the user requirements which were identified through performed survey and interviews with employees of the museum. The location of the user of the application can be identified in real time with object-level accuracy. Moreover, the information about the exhibits is added in the navigation system and in that way DaRen application is differentiated from other indoor navigation systems that provide only the route to a final destination. Additional information is important to the visitors of the museum that are not familiar with the environment and want to be aware of what the museum can offer to them. Moreover, the application provides recommendations about the most important exhibits and some interesting routes. The application provides easy to use interface where the route is represented clearly in 3D space. The main features the system is missing are voice guidance and orientation support which were considered as important parameters for the application from the museum visitors point of view. Additionally, the application meets the needs of the managers of the museum as it can provide the museum with the information about the number of visitors and the most visited exhibitions. Moreover, the system does not require additional resources (hardware) to implement and maintain the system.
2. In the system architecture of DaRen two types of components are distinguished: system (application, database and web service) and functional (localization, navigation and 3D visualization). This type of system architecture ensures the high performance and security of data. All data that are necessary to make the application functional are stored in a database on an external server. The database contains information about the museum and the exhibits, navigation nodes and the subdivision of the 3D model. Use of database allows easy updates and spatial calculations while web service strengthens the connection between an application and database. Application provides interaction with user and visualizes the 3D model and navigation path using Unity3D game engine.
3. For the localization of the user two different techniques based on Wi-Fi signal strength are combined. Localization using fingerprinting was selected as the most appropriate

technique for the DaRen application. The fingerprinting application that has been created in Wuhan University was integrated in the DaRen application. The value add of DaRen application is that ten positions instead of one are calculated to minimise the influence of outliers and determine the room in which the visitor is located with high accuracy. The determined location is used as an input for the navigation algorithm. The second technique, the Wi-Fi Monitoring based on the Meshlium Xtreme monitors, is used complementary to the fingerprinting. The two Wi-Fi monitors are implemented in the existing WLAN-infrastructure in Hubei Provincial Museum. The main contribution of this method in the project is the provision of important information to the museum managers as mobile devices are detected by the Wi-Fi monitors without the need of user's permission. The provided information is related to the number of people visiting the museum or a specific exhibition as well as their behaviour in the museum area. One important factor is also that this information can be retrieved by the museum managers every time of the day from their computer. Another important aspect of comparing and using the abovementioned techniques is that significant information were obtained about the Wi-Fi signal behaviour and the sources that cause interferences were discovered.

4. The most appropriate navigation approach for the Hubei Provincial Museum is network. Network is easy to implement and maintain since the Hubei Provincial Museum is large and complex building these characteristics are advantages. Additionally, network allows fast computations which are essential for the navigation as the user is most likely not willing to wait long for the guidance. Moreover, the network enables easy updates which are needed since exhibitions might change during the time. To implement the navigation services in DaRen application the dense context aware navigation network based on topological and geometrical information for the first floor of the museum was created. In the created network spatial objects such as rooms, doors and exhibits are represented as nodes and linked together according to their connectivity and adjacency. The developed navigation system can not only guide the user to the exhibition rooms but also generate the route to the specific exhibit and provide its description. Additional information concerning every object is attached to the node. The adapted A* algorithm allows shortest path calculations with very low computation costs. The other important advantage of the extracted navigation network is that it was generated semi-automatically therefore the navigation network can be updated easily if the indoor environment is changed. Additionally, the algorithm to derive the network is versatile and can be adapted in other cases (shopping malls, airports or other museums) with minor changes.
5. The 3D model of the Hubei Provincial Museum is visualised on a mobile device using Unity3D game engine which ensures high level of performance, enables texture representation of spatial objects and provides graphical user interface. The user-friendly interface of the DaRen application was designed which allows easy, natural and engaging interaction between the user and the system. Based on the results of the performed survey 3D model visualization with Bird's eye view which provides user with good overview of the scene while emphasizing the closest objects is implemented in the application.

10.2 Future Work

Although DaRen application demonstrated promising results, several shortcomings and recommendations could be addressed.

- Different user types in the application should be considered.
- The two localization sub-systems to be integrated into one application that serves both visitors and museum managers' needs.
- The 3D navigation network for the whole building should be generated.
- The 3D model should be stored in the database to save storing space of the application.
- The important semantic information should be attached to the 3D model and visualized on the screen of mobile device.

10.3 Future Implementation

The possibilities for implementing DaRen application in order to solve real world problems should be investigated. One of the possible fields of implementation are transportation hubs. "Transit centers should offer well-designed guidance signs to help passengers find their way to desired destinations easily and quickly" (*Zhang et al., 2010*). Transit centers or transportation hubs, such as airports and train stations are the nodes of public transport. Planes, trains, trams, subways and busses are all connected and brought together in these transportation hubs. In the Netherlands, yearly 8,9 million people [a] use the train at least once and 51 million people go through Schiphol airport (*Schiphol Group, 2012*) to start, end or continue their journey to work, home, family etc.

Way-finding is a key aspect in these transportation hubs, since the transfer time is preferred to be as little as possible. Entire studies are dedicated to improve it. For example, Farr et al. (2013) investigated effective way-finding in airports using a Bayesian network approach and the concept and process behind this (*Farr et al., 2012*). Zhang et al. (2010) found a set of key problems concerning way-finding in transportation hubs in their research. They investigated three different areas in a transportation hub where train, subway and bus come together to define the most problems users experience while finding their ways. These three areas are called TA-1, TA-2 and TA-3 in Figure 68 below.

Problem for Passengers	TA-1	TA-2	TA-3
Difficult to identify the current position	√	√	√
Information about other function areas out of subway station is insufficient	√		
Difficult to determine the route	√	√	√
Difficult to identify where is north, south, east, or west	√	√	√
Difficult to choose next steps: turning to right or left, going straight or backward	√	√	√
Difficult to determine position of destination	√	√	√
Difficult to understand the comparative spatial relation between underground layer and ground layer		√	
Difficult to understand the information in the existing regional sign	√	√	√
Difficult to relate the graphic sign to the real environment			√

Figure 68: Wayfinding problems for passengers in a transportation hub.

These problem definitions can be used to determine the potential of the DaRen application to improve wayfinding in transportation hubs. The increasing usage of mobile devices and the implementation of Wi-Fi in public areas like these transportation hubs are all in the advantage of the application. For every problem, the potential of the DaRen application is discussed below.

Difficult to identify the current position

The actual location of the user is not visualized in the application, however the application shows the closest node in the network as the starting point of the navigation. The average distance between two nodes in the network of Hubei Provincial Museum is 2 meters, the maximum distance is 9,5 meters. These distances depend on the extraction of the network from the floor plan, different floor plans give different distances. It can be stated that, with an average node-distance of 2 meters, the starting point visualized in the 3D model of the DaRen application is a rough estimation (in the order of magnitude of meters) of the location of the user.

Information about other function areas out of subway station is insufficient

DaRen is a multi-purpose 3D indoor navigation application which also provides detailed information about the room the user is located. The implementation of information about function areas out of subway can be easily implemented in the system.

Difficult to determine the route

The DaRen application automatically calculates the navigation route to the desired destination using the A* shortest-path algorithm. This route is visualized in the 3D model.

Difficult to identify where is north, south, east or west

This is not implemented in the DaRen application. However, the inertial sensors of mobile devices provide the opportunity to derive this information easily. The information could be displayed in the 3D viewer as a compass.

Difficult to choose next steps: turning to right or left, going straight or backward

The choice between going left or right won't have to be made anymore, when the entire navigation route is visualized in the 3D model. However, the 3D model should contain enough recognizable objects to determine where the path is going.

Difficult to determine position of destination

The application is able to store different destinations, by name and by picture. With only one click on a destination, the route to this destination is visualized.

Difficult to understand the comparative spatial relation between underground layer and ground layer

In the current version of the DaRen application it isn't possible to navigate between different floors. This is something for future work. When it would be possible, information about the current floor can be provided.

Difficult to understand the information in the existing regional sign

An advantage of the DaRen application is that everyone can understand the visualization of the path. Also the possible destinations can be given using their names and pictures. Therefore this method excludes the use of regional signs.

Difficult to relate the graphic sign to the real environment

It is important that the 3D model of the desired transportation hub is as enriched as possible. The more details, the easier it is for the user to recognize objects in the 3D model. If the user is able to relate the real environment to the 3D model, the navigation will be easier.

The DaRen application is able to solve most of these problems and in the future maybe even all. But why is this method not implemented in most transportation hubs then? This is because there are a few downsides on the implementation of the DaRen application. Firstly, the user is expected to be the owner of a mobile device suitable for the application. Where most people already own these devices, it still not covers all of the users of the transportation hubs. Secondly, the transportation hub should provide a suitable existing WLAN-infrastructure or create one. The last option is expensive, however nowadays most transportation hubs like airports already provide a existing WLAN-infrastructure. Thirdly, the implementation time needed for creating the radio map is long. The creation of the radio map of only one wing on one floor of the Hubei Provincial Museum took approximately 17 hours. The time needed for an entire airport or multiple train stations will be much higher. Finally, the existing DaRen application is not capable of navigating on different floors. Most airports only contain one floor, but transportation hubs like train- and subway stations contain multiple floors.

However, even with these downsides the DaRen application shows potential concerning transportation hubs. Reviewing on the problems people experience on wayfinding, there is a need for such an application. People can already navigate from hub to hub using different navigation systems, but an application for navigation inside the hub would make the route complete.

References

- Afyouni, I., Ray, C. and Claramunt, C., 2012. Spatial models for context-aware indoor navigation systems: A survey. *Journal of Spatial Information Science*, 4, pp. 85–123.
- Al-Najdawi, N., 2007. Introduction to visualization using game engines. In: AHRC Methods Network Workshop.
- Arthur, C., (2012, August). "Android is winning - if you're writing apps for China. Elsewhere, though...". <http://www.theguardian.com/technology/appsblog/2012/aug/16/android-winning-apps-china-smartphone>
- Calisir, F., Gumussoy, C.A., Bayraktaroglu, A.E., Saygivar, E, 2011. Usability and Functionality:A comparison of Key Project Personnel's and Potential Users' Evaluations, World Academy of Science, Engineering and Technology.
- Chan, E.C.L., Baciú, G., 2012. Introduction to Wireless Localization: With iPhone SDK Example
- Chan, S., Sohn, G., 2012. *Indoor Localization Using Wi-Fi based fingerprinting and trilateration techniques for LBS applications*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-4/C26.
- Chandgadkar A. , 2013. An Indoor Navigation System For Smartphones. Imperial College London.
- Chehimi, F., Coulton, P., Edwards, R. 2006. Advances in 3D graphics for Smartphones. In: *Information and Communication Technologies. ICTTA'06. 2nd*. IEEE, 2006. p. 99-104
- Choi, J., & Lee, J., 2009. 3D Geo-Network for Agent-based Building Evacuation Simulation. *3D Geo-Information Sciences*. Springer Berlin Heidelberg, pp. 283-299.
- Cisco, 2013. *White Paper: Presence Analytics* [pdf] Cisco Meraki. Available at: https://meraki.cisco.com/lib/pdf/meraki_whitepaper_presence.pdf [Accessed 25 October 2013].
- Evennou, F. and Marx, F., 2006. *Advanced Integration of WiFi and Inertial Navigation Systems for Indoor Mobile Positioning*. EURASIP Journal on Applied Signal Processing . Volume 2006, Article ID 86706, Pages 1–11.
- Fallah, N., Apostolopoulos, I., Bekris, K., and Folmer, E., 2013. Indoor Human Navigation Systems: A Survey. *Interacting with Computers*, 25.1: 21-33.
- Farr, A.C., Kleinschmidt, T., Yarlagadda, P.K.D.V., Mengersen, K.L., 2012. *Wayfinding: A simple concept, a complex process*.

- Farr, A.C., Kleinschmidt, T., Johnson, S., Yarlagadda, P.K.D.V., Mengersen, K.L., 2013. *Investigate Effective Wayfinding in Airports: a Bayesian Network Approach*. Transport
- Fritsch D, Kada M, 2004. Visualization using game engines. *Archiwum ISPRS*, 35: B5.
- Gressman, B., Klimek, H., Turau, V., 2010. Towards Ubiquitous Indoor Location Based Services and Indoor Navigation, Hamburg University of Technology, Hamburg, Germany
- Haojun, A., Taizhou, L., Jianjian, W., Menglei, Z., 2011. *A New Wi-Fi Fingerprint Indoor Localization Method for Smart Phone's Heterogeneity*. School of Computer Science, Wuhan University.
- Henniges, R., 2012. *Current approaches of Wifi Positioning*. Service-Centric Networking Seminar.
- Junghare U S, Thakare V M and Dharaskar R V, 2012. Efficient and Effective 3D Visualization Techniques for Rendering on Mobile Devices. *World Journal of Science and Technology*, 1.12.
- Karas, I. R., Batuk, F., Akay, A. E., and Baz, I., 2006. Automatically extracting 3D models and network analysis for indoors. *Innovations in 3D Geo Information Systems*. Springer Berlin Heidelberg, pp. 395-404.
- Kim, Y.J., Kang, H.Y., Lee, J., 2013. Development of Indoor Spatial Data Model Using CityGML ADE. *Journal of Korea Spatial Information Society*. Vol. 21. nr. 2, pp. 11-21
- Kray, C., Laakso, K., Elting, C., and Coors, V, 2003. Presenting route instructions on mobile devices. In: *Proceedings of the 8th international conference on intelligent user interfaces*. ACM, p. 117-124.
- Lamberti F., Sanna A., 2007. A streaming-based solution for remote visualization of 3D graphics on mobile devices. *Visualization and Computer Graphics, IEEE Transactions on* 13.2: 247-260.
- Le, M.H.V., Saragas, D., Webb, N., 2009. *Indoor Navigation System for Handheld Devices*. Worcester Polytechnic Institute, Worcester, Massachusetts, USA
- Lee, J., 2004. A Spatial Access-Oriented Implementation of a 3-D GIS Topological Data Model for Urban Entities. *GeoInformatica*, 8(3), pp. 93-113.
- Lee, S., Lee, Y.J., 2010. Efficient topological data models for spatial queries in 3D GIS, in: *A special joint symposium of ISPRSTechnical Commission IV & AutoCartoin conjunction with ASPRS/CaGIS 2010 Fall Specialty Conference*. Available at: <[http://www.isprs.org/proceedings/XXXVIII/part4/files/Lee Seokho.pdf](http://www.isprs.org/proceedings/XXXVIII/part4/files/Lee%20Seokho.pdf)> [Accessed 25 October 2013].

Li, J.K.J., Lee, J., 2010. Indoor spatial awareness initiative and standard for indoor spatial data, in: *Proceedings of IROS 2010 Workshop on Standardization for Service Robot*.

Libelium, 2013. *Meshlium Technical Guide v4.0* [pdf] Libelium. Available at: <<http://www.libelium.com/development/Meshlium/documentation/Meshlium-technical-guide/>> [Accessed 25 October 2013].

Lin, F.S., 2007. The study of domestic and foreign visitors' information needs in the national palace museum in Taiwan, *International Association of Societies of Design Research 2007*

Lorenz, B., Ohlbach, H.J. and Stoffel, E.P., 2006. A Hybrid Spatial Model for Representing Indoor Environments. In *Web and Wireless Geographical Information Systems*. Springer Berlin Heidelberg, pp. 102-112.

Lu, B., Niu, J., Juny, J., Cheng, L., Guy Y., 2013. *WiFi Fingerprint Localization in Open Space*. State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China and Singapore University of Technology and Design, Singapore

Ma, J., Li, X., Tao, X., and LU, J., 2008. *Cluster Filtered KNN: A WLAN-Based Indoor Positioning Scheme*. State Key Laboratory for Novel Software Technology

Machaj, J., Brida, P., 2012. *Impact of Radio Fingerprints Processing on Localization Accuracy of Fingerprinting Algorithms*. Electronics and electrical engineering. No7 vol . 123

Martin, E., Liu, L, Covington, M., Pest, P., Weber, M., 2010. *Location-Based Services Handbook: Applications, Technologies and Security*

Martin, E., Vinyals, O., Friedland, G., Bajcsy, R., 2010. *Precise Indoor Localization Using Smart Phones*.

Mautz, R. , 2012. Habilitation Thesis in Indoor Positioning Technologies, Institute of Geodesy and Photogrammetry, Department of Civil, Environmental and Geomatics Engineering, ETH Zurich.

Meng L., Reichenbacher T., 2005. *Map-based mobile services*. Springer Berlin Heidelberg.

Miliano, V., 1999. Unrealty: application of a 3D game engine to enhance the design, visualization and presentation of commercial real estate. In: *Proceedings of 1999 International Conference on Virtual Systems and Multimedia (VSMM'99)*. p. 508-513.

Munkres, J. R., 1984. *Elements of Algebraic Topology*. Addison-Wesley, Menlo Park, CA OGC, (October 2013). "IndoorGML SWG". *Open Geospatial Consortium Official Web Page*. Available at: <<http://www.opengeospatial.org/projects/groups/indoorgmlswg>> [Accessed 25 October 2013].

- Musa, A.B.M., Eriksson, J., 2012. *Tracking Unmodified Smartphones Using Wi-Fi Monitors*.
- Nagel, C., Becker B., Kaden, R.; Li, K.-J.; Lee, J., and Kolbe, T. H. 2010. Requirements and Space-Event Modelling for Indoor Navigation, OGC 10-191rl.
- Papaspyrou, D., 2012. Development of an Indoor Navigation Application for Mobile Terminals, National Technical University of Athens, Greece
- Radoczky, V., 2005. Karten für Fußgängernavigation–kann die Bildung des mentalen Modells beeinflusst werden. *Strobl, Blaschke, Griesebner (eds.): Angewandte Geoinformatik*, 568-573.
- Rehim, M.A.A.A.Y.A., 2004. *Horus: A WLAN-Based Indoor Location Determination System*.
- Shi, J., 2012. The personal information protection legislation under Web environment. Available at: http://d.g.wanfangdata.com.cn/Periodical_szdxxb-zxshkxb201206013.aspx
- Schiphol Group, 2012. *Feiten & Cijfers 2012*. [pdf] Available at: <<http://www.schiphol.nl/Reizigers/Nieuws/2012Jaarverslag1.htm>> [Accessed at 25 October 2013]
- Sidiropoulos, N., Mioduszewski, M., Oljasz, P., Schaap, E., 2012. *Open Wifi SSID Broadcast vulnerability*.
- Spasov, I., 2007. Algorithms for Map-Aided Autonomous Indoor Pedestrian Positioning and Navigation, Ph.D. Ecole Polytechnique Federale de Lausanne.
- Stone D, Jarrett C, Woodroffe M, and Minocha, S, 2005. User interface design and evaluation. Morgan Kaufmann.
- Suh, D., 2012. Introduction of ISO/NP17438 and COEX Indoor App Case. Available at: <http://stem.cs.pusan.ac.kr/indoorGMLWorkshop/slides/3>
- Van Loenen, B., de Jong, J., 2007. 'SDIs and Privacy: Conflicting Interests of the Spatially Enabled Society', Chapter 21, In A. Rajabifard (ed.), *Towards a Spatially Enabled Society*, ISBN 978-0-7325-1620-8, University of Melbourne, pp. 271-284
- Warren, S. D., & Brandeis, L. D., 1890. The right to privacy. *Harvard law review*, 4(5), 193-220.
- Xu, H., Gupta, S., & Shi, P., 2009. Balancing User Privacy Concerns in the Adoption of Location-Based Services: An Empirical Analysis across Pull-Based and Push-Based Applications. *iConference (iSociety: Research, Education, and Engagement, University of North Carolina-Chapel Hill*.

Yuan, L., Zizhang, H.E., 2008. 3D indoor navigation: A framework of combining BIM with 3D GIS, in: 44th ISOCARP Congress.

Zafeiropoulos, A., Solidakis, E., Zoi, S., Konstantinou, N., Stathopoulos, P., Mitrou, N., 2006. Location Based Guidance Services in a Museum Environment: Deployment Issues and a Proposed Architectural Approach, National Technical University of Athens, Greece

Zhai, S., Kandogan, E., Smith, B. A., & Selker, T., 1999. In search of the ‘magic carpet’: Design and experimentation of a bimanual 3D navigation interface. *Journal of Visual Languages & Computing*, 1999, 10.1: 3-17.

Zhang, Y., Chen, X., Jiang, J., 2010. *Wayfinding-Oriented Design for Passenger Guidance Signs in Large-Scale Transit Center in China*. Transportation Research Record: Journal of the Transportation Research Board, No. 2144, Transportation Research Board of the National Academies, Washington, D.C., 2010, pp. 150–160.

Zlatanova, S., 2013. Space subdivision for indoor applications. Overview of 2D and 3D approaches: preliminary study. TU Delft, the Netherlands.

Websites:

[a] http://www.treinreiziger.nl/kennisnet/feiten_en_cijfers:_ns_in_2010-143388

<http://theory.stanford.edu>

<http://wlanbook.com/iphone-wifi-monitor-apps-banned-by-apple/>
WeChat, 2013. http://www.wechat.com/en/features.html#live_chat

Tencent, 2013. <http://weixin.qq.com/agreement>

Article 29 Working Party, (2013). *Opinion 02.2013 on apps on smart phones*.

http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2013/wp202_en.pdf

WBP. (2000), *Wet Bescherming Persoongegevens*.

http://www.cbpreweb.nl/Pages/pb_20111115_google.aspx

Datoo, S. (2013), *This recycling bin is following you*.

<http://qz.com/112873/this-recycling-bin-is-following-you/>

Appendix A – Localization Component

Code for the determination of the starting point for the navigation:

```

DROP TABLE IF EXISTS locations CASCADE;
DROP VIEW IF EXISTS closestnode CASCADE;

CREATE TEMPORARY TABLE locations (locationid int, the_geom geometry);
INSERT INTO locations VALUES(values Wuhan Application)
ALTER TABLE locations ADD COLUMN roomid int;

UPDATE locations
SET roomid = rooms.roomid
FROM rooms
WHERE ST_Within(locations.the_geom, rooms.the_geom);

CREATE TEMPORARY VIEW closestnode (locationid, nodeid)
AS (SELECT locationid, nodeid
FROM nodes, locations
WHERE ST_Distance(nodes.the_geom, locations.the_geom) IN (
SELECT MIN(ST_Distance(nodes.the_geom, locations.the_geom))
FROM nodes, locations
WHERE nodes.roomid = (SELECT roomid
FROM (SELECT roomid, COUNT(locations.roomid) as k FROM locations GROUP BY roomid) as t
WHERE k = (SELECT MAX(k) FROM (SELECT COUNT(locations.roomid) as k
FROM locations GROUP BY roomid) l) LIMIT 1)
GROUP BY locationid))
;

SELECT nodeid
FROM (SELECT nodeid, COUNT(nodeid) AS r FROM closestnode GROUP BY nodeid) as q WHERE r=(
SELECT MAX(r) FROM (SELECT COUNT(nodeid) AS r FROM closestnode GROUP BY nodeid) l);

```

Code for extraction of information about museum visitors' behaviour in the museum

Number of visitors and time they spend in the exhibition:

```

LOAD DATA LOCAL INFILE 'C:/TU2013_14/GE02000/MeshliumDB_kpn_2.csv'
INTO TABLE MESHLIUMDB.kpn FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY '\r\n';

DROP table if exists meshliumdb.merged;

Drop table if exists meshliumdb.tabelTimedifference;

create table meshliumdb.MERGED as select *
from meshliumdb.tmobile where DAY(TimeStamp)=4 AND MONTH(TimeStamp)=10
UNION select * from meshliumdb.kpn where DAY(TimeStamp)=4 AND MONTH(TimeStamp)=10;

DROP table if exists meshliumdb.min;
Drop table if exists meshliumdb.max;

create table meshliumdb.min as select Device, min(TimeStamp) as minimal, MAC as MAC_min
from meshliumdb.merged
group by MAC_min;

create table meshliumdb.max as select Device, max(TimeStamp) as maximal, MAC as MAC_max
from meshliumdb.merged
group by MAC_max;

CREATE table meshliumdb.tabelTimedifference as
SELECT *, TIMESTAMPDIFF(minute, min.minimal, max.maximal) as Timedifference
FROM meshliumdb.min
JOIN meshliumdb.max
ON max.MAC_max = min.MAC_min;

/*time spend at the exhibition*/
SELECT count(mac_min)
from meshliumdb.tabelTimedifference
where tabelTimedifference.Timedifference>0
order by Timedifference DESC;

SELECT count(mac_min)
from meshliumdb.tabelTimedifference
where tabelTimedifference.Timedifference>10
order by Timedifference DESC;

/*unique mac addresses*/
select count(distinct(mac))
from meshliumdb.merged;

```

Direction of visitors in the exhibition:

```
DROP table if exists meshliumdb.tableDirection;
DROP table if exists meshliumdb.tableKPN;
DROP table if exists meshliumdb.tableTMOBILE;

create table meshliumdb.tableKPN as select MAC, Max(TimeStamp) as TimeStamp
from meshliumdb.kpn
where DAY(TimeStamp)=4 AND MONTH(TimeStamp)=10
group by MAC
UNION select MAC, Min(TimeStamp) as TimeStamp from meshliumdb.kpn
where DAY(TimeStamp)=4 AND MONTH(TimeStamp)=10
group by MAC;

create table meshliumdb.tableTMOBILE as select MAC, Max(TimeStamp) as TimeStamp
from meshliumdb.tmobile
where DAY(TimeStamp)=4 AND MONTH(TimeStamp)=10
group by MAC
UNION select MAC, Min(TimeStamp) as TimeStamp from meshliumdb.tmobile
where DAY(TimeStamp)=4 AND MONTH(TimeStamp)=10
group by MAC;

CREATE table meshliumdb.tableDirection as
SELECT TIMESTAMPDIFF(minute, tableKPN.TimeStamp, tableTMOBILE.TimeStamp) as Direction
FROM meshliumdb.tableKPN
JOIN meshliumdb.tableTMOBILE
ON tableKPN.MAC = tableTMOBILE.MAC
GROUP BY tableKPN.TimeStamp;

/*right direction from kpn to tmobile*/
select count(Direction)
from meshliumdb.tableDirection
where Direction>20;

/*wrong direction from tmobile to kpn*/
select count(Direction)
from meshliumdb.tableDirection
where Direction<(-20);
```

Appendix B – Navigation Component

Network Extraction

```

import fiona
import triangle
from numpy import array
import itertools
import psycopg2
import xlrd

from shapely.geometry import MultiPolygon
from shapely.geometry import Polygon
from shapely.geometry import Point
from shapely.geometry import asPoint

# Files used to extract network
FILE = 'Polygons_Floor3_v4.shp'
FILE_EXHIBITS = 'Exhibits.shp'
FILE_ROOMS = 'Rooms.shp'
FILE_INFORMATION = 'Exhibits Information.xlsx'
BUFFER = 0.0005

# load files into Python
data = fiona.open(FILE)
exhibits = fiona.open(FILE_EXHIBITS)
rooms = fiona.open(FILE_ROOMS)
book = xlrd.open_workbook(FILE_INFORMATION)
information = book.sheet_by_index(0)

dictGeom = []
triList = []
doors = []
polygonList = []

# Process the data obtained from the shp file.
# Corridors and rooms polygons should be stored as vertices, segments and holes,
# So that it can be used as input for the Delanauy triangulation
# For doors we only need the shapely Polygon
for each in data:
    prop = each['properties']
    geom = each['geometry']
    v = geom['coordinates']

    # Only rooms and corridors that needs pre-processing
    if not ((prop['Type'] == 'Door') or (prop['Type'] == 'Stairs') or (prop['Type']
== 'Escalator')):
        edges = []
        holePoints = []
        start = 0

        # Determine outerring and innerring
        # For every ring make a connected segment
        # For every inner ring determine a point inside the inner ring (hole)
        for ind, ring in enumerate(v):
            if start > 0:

```

```

        pg = Polygon(ring)
        hole = pg.representative_point().coords[:]
        holePoints.append(hole)
    elif start == 0:
        polygonList.append(Polygon(ring))

# remove the end point of each ring, since it is the same as the start
point
v[ind] = ring[0:-1]
length = len(v[ind])

for i in range(0, length):
    edges.append([start + i, start + ((i + 1) % length)])
    start = start + length

# Merge all the points in the inner and outer ring in one list
pointsMerged = list(itertools.chain(*v))

# Store the results
if holePoints:
    dictGeom.append(dict(vertices=array(pointsMerged), segments=array(edges),
holes=array(holePoints)))
else:
    dictGeom.append(dict(vertices=array(pointsMerged),
segments=array(edges)))

# Store doors as shapely Polygons
elif prop['Type'] == 'Door':
    doors.append(Polygon(v[0]))

# Apply Delaunay Triangulation on every polygon and store resulting triangles in list
for ind, d in enumerate(dictGeom):
    t = triangle.triangulate(d, 'p')
    triList.append(t)

centroidPointsList = []
neighbors = []
triPolygons = []
# Calculates for every triangle created by the triangulation the central point
for tri in triList:
    points = []
    polyTriList = []
    vertList = tri['vertices']
    triInd = tri['triangles']
    centroidPoints = []
    for t in triInd:
        points = vertList[t].tolist()
        points.append(points[0])
        polygon = Polygon(points)
        polyTriList.append(polygon)
        cPoint = polygon.centroid
        c = cPoint.coords[:]
        centroidPoints.append(c[0])
    triPolygons.append(polyTriList)
    centroidPointsList.append(dict(vertices=array(centroidPoints)))
ng = []
# Calculates the neighbors of each triangle which can be a maximum of three.

```

```

# The central points of the neighboring triangles are the neighbors of the
central point
for ind, polygon in enumerate(polyTriList):
    for otherInd, otherPolygon in enumerate(polyTriList):
        if not (ind == otherInd):
            mPolygon = MultiPolygon([polygon, otherPolygon])
            if not mPolygon.is_valid:
                ng.append([ind, otherInd])

    neighbors.append(ng)

networkPoints = []
networkEdges = []
# All the centroid points are put together in one list, this are the network points.
# In the network edge list every central point gets an edge to its neighbors
for ind, each in enumerate(centroidPointsList):
    lengthNetwork = len(networkPoints)
    vertices = each['vertices'].tolist()
    edges = neighbors[ind]
    for e in edges:
        e[0] = e[0] + lengthNetwork
        e[1] = e[1] + lengthNetwork
        networkEdges.append(e)
    networkPoints.extend(vertices)

exhibitsPoints = []
# Every exhibit polygon will be represented by a point.
for each in exhibits:
    geom = each['geometry']
    v = geom['coordinates']
    exhibitsPolygon = Polygon(v[0])
    exp = exhibitsPolygon.representative_point().coords[:]
    exhibitsPoints.append(exp)

exhibitNodeIDs = []
lengthNetwork = len(networkPoints)
# For every exhibit point the closest network point is calculated
# Then an edge is created between these points and exhibit point is added to network
points list
for ind,exp in enumerate(exhibitsPoints):
    shortestDistance = float("inf")
    winner = 0
    for ind2 in range(0, lengthNetwork):
        np = networkPoints[ind2]
        dis = Point(exp).distance(Point(np))
        if dis < shortestDistance:
            winner = ind2
            shortestDistance = dis
    index = ind+lengthNetwork
    networkEdges.append([index, winner])
    networkPoints.extend(exp)
    exhibitNodeIDs.append(index)

doorPoints = []
# Now, the rooms needs to be connected.
# This is done by finding the neighboring triangles of each door.

```

```

# Then an edge is created from the door point to both the central points of the
# neighboring triangle
# Finally door point is added to network point list
for door in doors:
    index = 0
    doorPoint = door.centroid
    for ind, poly in enumerate(polygonList):
        trip = triPolygons[ind]
        if not door.disjoint(poly.buffer(BUFFER)):
            for ind2, t in enumerate(trip):
                mPolygon = MultiPolygon([door, t])
                if not mPolygon.is_valid:
                    networkEdges.append([len(networkPoints), index+ind2])
            index = index + len(trip)
    networkPoints.extend(doorPoint.coords[:])

# Calculate the cost of each edge by using the distance between two points
costEdges = []
for e in networkEdges:
    sPoint = Point(networkPoints[e[0]])
    tPoint = Point(networkPoints[e[1]])
    cost = sPoint.distance(tPoint)
    costEdges.append(cost)

roomList = []
# Get rooms
for each in rooms:
    geom = each['geometry']
    cood = geom['coordinates']
    roomPolygon = Polygon(cood[0])
    roomList.append(roomPolygon)

nodeInRoomList = []
# Calculate for every network point in which room it is
for ind,eachNode in enumerate(networkPoints):
    for ind2,rPolygon in enumerate(roomList):
        if Point(eachNode).within(rPolygon):
            nodeInRoomList.append(ind2)
            break

exhibitsInformationList = []
# exhibit information is extracted
for index in range(1,information.nrows):
    cells = information.row(index)
    info = []
    for ind, cell in enumerate(cells):
        if not (ind == 1):
            info.append(cell.value)
        elif ind == 1:
            info.append(exhibitNodeIDs[index])
    exhibitsInformationList.append(info)

# Log in credentials database
HOST = 'xxx.xxx.xxx.xx'
PORT = "xxxx"
DBNAME = "*****"

```

```
USERNAME = "*****"
PASSWORD = "*****"

# Make connection to database
connection = psycopg2.connect(database = DBNAME, user = USERNAME,
                              password = PASSWORD, host = HOST, port=PORT)

cursor = connection.cursor()
# Clean tables before inserting new data
cursor.execute('TRUNCATE exhibits')
cursor.execute('TRUNCATE rooms')
cursor.execute('TRUNCATE edges')
cursor.execute('TRUNCATE nodes')

# Insert the network points
for nodeID, np in enumerate(networkPoints):
    SQL = "INSERT INTO nodes (nodeID, the_geom, roomID) VALUES (%s,
ST_PointFromText(%s, 4326), %s)"
    pointText = asPoint(array(np)).wkt
    data = [nodeID, pointText, nodeInRoomList[nodeID]]
    cursor.execute(SQL, data)

# Insert the network edges
for edgeID, ne in enumerate(networkEdges):
    SQL = "INSERT INTO edges (edgeID, source, target, cost) VALUES (%s, %s, %s, %s)"
    data = [edgeID, ne[0], ne[1], costEdges[edgeID]]
    cursor.execute(SQL, data)

# Insert the rooms
for roomID, rp in enumerate(roomList):
    SQL = "INSERT INTO rooms (roomID, the_geom) VALUES (%s, ST_PolygonFromText(%s,
4326))"
    polygonText = rp.wkt
    data = [roomID, polygonText]
    cursor.execute(SQL, data)

# Insert exhibit information
for info in exhibitsInformationList:
    SQL = "INSERT INTO exhibits (exhibitID, nodeID, name, description) VALUES(%s, %s,
%s, %s)"
    data = [info[0], info[1], info[2], info[3]]
    cursor.execute(SQL, data)

connection.commit()
cursor.close()
connection.close()
```

Network Storing

```
DROP TABLE exhibits;
DROP TABLE rooms;
DROP TABLE edges;
DROP TABLE nodes;

CREATE TABLE nodes (
    nodeID          INTEGER,
    the_geom        GEOMETRY (POINT, 4326),
    roomID          INTEGER,
    PRIMARY KEY (nodeID)
);

CREATE TABLE edges (
    edgeID  INTEGER,
    source  INTEGER,
    target  INTEGER,
    cost    DOUBLE PRECISION,
    PRIMARY KEY (edgeID)
);

CREATE TABLE rooms (
    roomID          INTEGER,
    the_geom        GEOMETRY (POLYGON, 4326),
    PRIMARY KEY (roomID)
);

CREATE TABLE exhibits (
    exhibitID       INTEGER,
    nodeID          INTEGER,
    name            VARCHAR (256),
    description     VARCHAR (512),
    PRIMARY KEY (exhibitID)
);
```

Appendix C – Web Service

```
<?php
```

```
// Log-in credentials database
$SERVER = "xxx.xxx.xxx.xx";
$DBNAME = "*****";
$USERNAME = "*****";
$PASSWORD = "*****";

//Create connection to database
$db = new PDO("pgsql:host=$SERVER;dbname=$DBNAME;user=$USERNAME;password=$PASSWORD");

//Fix for older php versions
if (function_exists('get_magic_quotes_gpc') && get_magic_quotes_gpc()) {

    function undo_magic_quotes_gpc(&$array) {
        foreach ($array as &$value) {
            if (is_array($value)) {
                undo_magic_quotes_gpc($value);
            } else {
                $value = stripslashes($value);
            }
        }
    }

    undo_magic_quotes_gpc($_POST);
    undo_magic_quotes_gpc($_GET);
    undo_magic_quotes_gpc($_COOKIE);
}

$locationData = [];
// Obtain the data from the header
// Where service and target are always given in the header
foreach ($_GET as $key => $data) {
    if ($key == "service") {
        $service = $data;
    } else if ($key == "target") {
        $target = $data;
    } else {
        $dataNew = strtr($data, array('_' => ' '));
        $locationData[] = $dataNew;
    }
}

// if service is navigation then first the user's position is calculated
// After that a position is given back and that can be used as source point for the a* algorithm
if ($service == "navigation") {
    $query1 = "DROP TABLE IF EXISTS locations CASCADE;";
    $query2 = "DROP VIEW IF EXISTS closestnode CASCADE;";
    $query3 = "CREATE TEMPORARY TABLE locations (locationid int, the_geom geometry);";
    $query4 = "INSERT INTO locations VALUES (1, ST_PointFromText(:data0, 4326)), (2,
ST_PointFromText(:data1, 4326)),
(3, ST_PointFromText(:data2, 4326)), (4, ST_PointFromText(:data3, 4326)), (5, ST_PointFromText(:data4,
4326)),
(6, ST_PointFromText(:data5, 4326)), (7, ST_PointFromText(:data6, 4326)), (8, ST_PointFromText(:data7,
4326)),
```

```

(9, ST_PointFromText(:data8, 4326)), (10, ST_PointFromText(:data9, 4326));
$query5 = "ALTER TABLE locations ADD COLUMN roomid int;";
$query6 = "UPDATE locations SET roomid = rooms.roomid FROM rooms WHERE
ST_Within(locations.the_geom, rooms.the_geom);";
$query7 = "CREATE TEMPORARY VIEW closestnode (locationid, nodeid) AS (
SELECT locationid, nodeid
FROM nodes, locations
WHERE ST_Distance(nodes.the_geom, locations.the_geom) IN (
SELECT MIN(ST_Distance(nodes.the_geom, locations.the_geom))
FROM nodes, locations
WHERE nodes.roomid = (SELECT roomid
FROM (SELECT roomid, COUNT(locations.roomid) as k FROM locations GROUP BY roomid) as t
WHERE k = (SELECT MAX(k) FROM (SELECT COUNT(locations.roomid) as k FROM locations
GROUP BY roomid) l) LIMIT 1)
GROUP BY locationid));";
$query8 = "SELECT nodeid
FROM (SELECT nodeid, COUNT(nodeid) AS r FROM closestnode GROUP BY nodeid) as q WHERE r=(
SELECT MAX(r) FROM (SELECT COUNT(nodeid) AS r FROM closestnode GROUP BY nodeid) l);";

$stmt1 = $db->prepare($query1);
$stmt1->execute();
$stmt2 = $db->prepare($query2);
$stmt2->execute();
$stmt3 = $db->prepare($query3);
$stmt3->execute();
$stmt4 = $db->prepare($query4);
$stmt4->bindParam(":data0", $locationData[0], PDO::PARAM_STR);
$stmt4->bindParam(":data1", $locationData[1], PDO::PARAM_STR);
$stmt4->bindParam(":data2", $locationData[2], PDO::PARAM_STR);
$stmt4->bindParam(":data3", $locationData[3], PDO::PARAM_STR);
$stmt4->bindParam(":data4", $locationData[4], PDO::PARAM_STR);
$stmt4->bindParam(":data5", $locationData[5], PDO::PARAM_STR);
$stmt4->bindParam(":data6", $locationData[6], PDO::PARAM_STR);
$stmt4->bindParam(":data7", $locationData[7], PDO::PARAM_STR);
$stmt4->bindParam(":data8", $locationData[8], PDO::PARAM_STR);
$stmt4->bindParam(":data9", $locationData[9], PDO::PARAM_STR);
$stmt4->execute();
$stmt5 = $db->prepare($query5);
$stmt5->execute();
$stmt6 = $db->prepare($query6);
$stmt6->execute();
$stmt7 = $db->prepare($query7);
$stmt7->execute();
$stmt8 = $db->prepare($query8);
$stmt8->execute();

$result2 = $stmt8->fetchAll();
$temp = $result2[0];
// User position
$source = $temp[0];

// Calculate shortest path
$query = "SELECT ST_X(the_geom), ST_Y(the_geom) FROM (SELECT id1 FROM pgr_astar('SELECT edgeID
as id, source, target, cost, x1, y1, x2, y2 FROM (SELECT edgeid, source, target, cost, x1, y1, st_x(the_geom) as x2,
st_y(the_geom) as y2 FROM (SELECT edgeid, source, target, cost, st_x(the_geom) as x1, st_y(the_geom) as y1
FROM edges JOIN nodes ON source = nodeID) AS test JOIN nodes ON target = nodeID) AS test2', :source,
(SELECT nodeID FROM exhibits WHERE exhibitID = :target), false, false)) AS test3 JOIN nodes ON id1 =
nodeID;";

```

```
$stmt = $db->prepare($query);
$stmt->bindParam(":source", $source, PDO::PARAM_STR);
$stmt->bindParam(":target", $target, PDO::PARAM_STR);
$stmt->execute();
$result = $stmt->fetchAll();
$response["message"] = $result;

// Return shortest path in JSON format
echo json_encode($response);

// In the case service of testnavigation, then the user position is not calculated,
// but a fixed source point is used for the navigation
} else if ($service == "testnavigation") {
    $query = "SELECT ST_X(the_geom), ST_Y(the_geom) FROM (SELECT id1 FROM pgr_astar('SELECT edgeID
as id, source, target, cost, x1, y1, x2, y2 FROM (SELECT edgeid, source, target, cost, x1, y1, st_x(the_geom) as x2,
st_y(the_geom) as y2 FROM (SELECT edgeid, source, target, cost, st_x(the_geom) as x1, st_y(the_geom) as y1
FROM edges JOIN nodes ON source = nodeID) AS test JOIN nodes ON target = nodeID) AS test2', :source,
(SELECT nodeID FROM exhibits WHERE exhibitID = :target), false, false)) AS test3 JOIN nodes ON id1 =
nodeID;";
    $stmt = $db->prepare($query);
    $source = "0";
    $stmt->bindParam(":source", $source, PDO::PARAM_STR);
    $stmt->bindParam(":target", $target, PDO::PARAM_STR);
    $stmt->execute();
    $result = $stmt->fetchAll();
    $response["message"] = $result;

    echo json_encode($response);
}
```


of the exhibit.

Yes

No

11. Is necessary to have a compass in the app?

Yes

No

12. What language do you think should be provided?

Chinese

English

Others

13. What's your preferred way to enter the destination?

A. Type

B. select

C. both A and B

D. others

14. What's the brand of your mobile phone?

Apple

Samsung

HTC

Nokia

Sony

Mi

Others

15. What's the OS of your mobile phone?

Android

IOS

Windows

Others

16. Are there other functions you hope to be included in the app? Please write at least one.

Appendix E – Database Component

Object import into database

Objimporter.py

```
# DaRen
from collada import *
import psycopg2
import ppygis
def geompro(colladageom):
    """
    union all the triangles belonging to a certain object to a multipolygon
    """
    polylist=[]
    for lpolygon in geom.primitives: #large polygon organized by texture
        ls=list(lpolygon)
        for each in ls: #small triangles inside each large polygon
            triangle=each.vertices.tolist()
            poly=ppygis.Polygon([ppygis.LineString([ppygis.Point(triangle[0][0],triangle[0][2],-triangle[0][1]),
                                                            ppygis.Point(triangle[1][0],triangle[1][2],-triangle[1][1]),
                                                            ppygis.Point(triangle[2][0],triangle[2][2],-triangle[2][1]),
                                                            ppygis.Point(triangle[0][0],triangle[0][2],-triangle[0][1])])]
                                polylist.append(poly)
            multipoly=ppygis.MultiPolygon(polylist) #multipolygon representing a certain object
        return multipoly
if __name__ == '__main__':
    mesh=Collada('F2.DAE')
    roomid=1201 #the first 1 represents room, 201 means the first room at the second floor
    corridrid=2201 #the first 2 represents corridor, 201 means the first corridor at the second floor
    hallid=3201 #the first 3 represents hall, 201 means the first hall at the second floor
    conn = psycopg2.connect(host='localhost',database='daren',
                            user='lhc',port='5433',password='916563')

    cur = conn.cursor()
    cur.execute('create table if not exists object (objectid int8 PRIMARY KEY, spaceid int8, nodeid int8,\
objtype varchar(20), geometry GEOMETRY(\'MULTIPOLYGONZ\'), navigatable boolean)')
    objnum = 1 #number for the objects
    for geom in mesh.geometries:
        name = geom.id
        nodeid = None #defined in another table, thus not applicable, needs to be edited later
        if 'Room' in name:
            spaceid = int('1'+[s for s in name.split('_') if s.isdigit()][0])
            objectid = int(str(spaceid) + str(objnum))
            objnum += 1
            multipoly = geompro(geom)
            if 'Floor' in name:
                objtype = 'floor'
                navigatable = False
            elif 'Ceiling' in name:
```

```
objtype = 'ceiling'

navigatable = False

elif 'Wall' in name:
    objtype = 'wall'
    navigatable = False
elif 'Desk' in name:
    objtype = 'desk'
    navigatable = False
elif 'Door' in name:
    objtype = 'door'
    navigatable = True
elif 'Window' in name:
    objtype = 'window'
    navigatable = True
else:
    objtype = 'unknown'
    navigatable = False
elif 'Corridor' in name:
    spaceid = int('2'+[s for s in name.split('_') if s.isdigit()][0])
    objectid = int(str(spaceid) + str(objnum))
    objnum += 1

if 'Wall' in name or 'Parapet' in name:
    objtype = 'wall'
    multipoly = geompro(geom)
    navigatable = False
elif 'Floor' in name:
    objtype = 'floor'
    multipoly = geompro(geom)
    navigatable = False

elif 'Ceiling' in name:
    objtype = 'ceiling'
    multipoly = geompro(geom)
    navigatable = False

elif 'Billboard' in name:
    objtype = 'billboard'

    multipoly = geompro(geom)
    navigatable = False

elif 'Chairs' in name:
    objtype = 'chairs'
    multipoly = geompro(geom)
    navigatable = False

elif 'Desk' in name or 'Reception' in name:
    objtype = 'desk'
    multipoly = geompro(geom)
    navigatable = False

elif 'Fence' in name:
    objtype = 'fence'
    multipoly = geompro(geom)
    navigatable = False
```

```
else: continue

elif 'Stairs' in name:
    spaceid = hallid
    objectid = int(str(spaceid) + str(objnum))
    objnum += 1
    objtype = 'stairs'
    multipoly = geompro(geom)
    navigatable = True

elif 'Escalator' in name:
    spaceid = hallid
    objectid = int(str(spaceid) + str(objnum))
    objnum += 1
    objtype = 'escalator'
    multipoly = geompro(geom)
    navigatable = True
else:
    continue

cur.execute('insert into object values (%s, %s, %s, %s, %s, %s)',
           (objectid, spaceid, nodeid, objtype, multipoly, navigatable))

conn.commit()
cur.close()
conn.close()
```