

IN3405 BACHELORPROJECT

The Friendly Project

Preface

This report is the result of the work completed for the course IN3405 Bachelorproject by Anne van Ee. This course is the final project of the Bachelor of Science in Computer Science at the Delft University of Technology.

This project was carried out at Collins Kumarasinghe Associates in Singapore from April 13th 2009 to June 12th 2009. The project consisted of two smaller assignments. The first assignment was to assess the user friendliness of their GMAT practice web application, the Test Zone at www.gmat-zone.com. The second assignment was to gather the requirements for a car parts website, www.parts.sg. The latter assignment required a lot of interaction with clients which was quite challenging but very fun and rewarding. Readers who are mostly interested in the user experience research can find this in chapter three. In chapter four the car parts website is discussed.

I owe many thanks to Micheal Collins at Collins Kumarasinghe Associates for giving me this opportunity to work on this project in Singapore. I would also like to thank Stijn Oomes of the TU Delft for agreeing to be my supervisor during this project. Sadly he became sick during the course of this project. Matthijs Sepers took his task and his comments on my documentation were very useful. I also owe Jan de Vries appreciation for supporting my goal to go to Singapore to carry out this project. Finally, I want to thank Hans Vriens for getting me in touch with Micheal and offering me a place to stay during my first week in Singapore.

Summary

The first assignment of this bachelorproject was to enhance the user experience of the Test Zone on www.gmat-zone.com. This web application helps users practice tests before taking the GMAT. This project required that research was done about the usability of the system using several techniques.

First, an online questionnaire was sent to students of the Rotterdam School of Management asking them to participate. They were asked to look at the Test Zone and share their thoughts by answering the questions in the questionnaire. Only one person filled in half of the questionnaire. A paper-based version was handed out to people who attended the GMAT training courses at Collins Kumarasinghe Associates and who had worked with the application before. Hardly anyone had used the application because they had just started studying for the GMAT. Most people started practicing with the application after they had finished the courses. The paper-based questionnaire was not a success either.

Since most people attending the courses would be using the application later, they were perfect users for the think-aloud protocol. This technique is used to identify any usability problems by letting users, who have never used the application before, play around with the system and talk about what they are thinking. This makes it possible to identify what people do not understand and what parts of the application are unclear. This technique proved very effective and a lot of data was gathered from multiple sessions with different users.

Most problems found using the think-aloud protocol were easily fixed. Most functionality was already available in the application but users had trouble finding it. Some pages were renamed and some bugs were fixed. Some new functionality was proposed during the second research phase.

This involved adding a progress bar to show users how much time they were spending on each question, the possibility to rate and flag questions while taking the tests and being able to send questions to the trainers through the website. Users were also asked about the new functionality and they thought that each extra function was a good idea. These functions were also implemented and tested, and did not show any other major problems with the application.

The second assignment was to gather the requirements for a car parts website, www.parts.sg, and implementing a part of the website. This assignment was not the project described in the Orientation Report. At first the assignment was to create an environment in which the web developer at Collins Kumarasinghe Associates could program websites easily and quickly. Research was done to see if another framework could support the company in a more efficient way. Collins Kumarasinghe Associates use Joomla to create websites. The project changed when the car parts website was ordered by the new clients. This was during the research phase and the idea became to pick a framework that would fit this project. This framework ended up being Drupal.

Drupal is a good framework to help create community based websites. It is possible to create many different types of users with different permissions. This was exactly what was needed for the car parts website. Normal users are allowed to view and edit content such as parts or car brand descriptions and these users can also offer used parts for sell. Workshops are allowed to sell new parts and offer services. Workshops are run by other types of users, namely Workshop Moderators and Mechanics. They have other permissions than users who are not part of a Workshop.

To gather the requirements for the website, multiple meetings were held with the clients. First, use cases and UML diagrams were created after meeting the clients for the first time. This highlighted what parts of the system were still unclear. After the second meeting, a PowerPoint presentation was made. This functioned as the prototype of the website and it contained almost all the functionality that would be found in the final version of the car parts website. This prototype proved to be very useful. The clients were happy to see the project taking shape and the prototype made it easy to discuss the different functions and to find missing functions. The prototype was also used to explain the website to the other developer. After the prototype was revised a couple of times, it was approved by the clients. After that the UML class diagram was finished and a functionality matrix was created. A Gantt chart was made to plan the implementation phase of the website.

To implement the website, it was decided to use custom and existing modules. Modules add certain functionality to the web application. Many useful existing modules were found but the other programmer had a preference for his own modules. This meant that he decided to code a lot of modules himself even though existing modules could create the same functionality. One existing module called Modr8 contained some useful features for the Moderator function. Code was added to this module to adhere to the specifications of the Moderator. A rating system and personal messaging system were both implemented using existing modules. Other functionality was built using a combination of different modules or custom modules.

Glossary

GMAT (Graduate Management Admission Test)

A computer adaptive standardized test in the English language for measuring the ability to succeed academically in graduate business studies.

JavaScript

A scripting language which makes it possible to create dynamic websites.

MVC (Model View Controller)

An architectural pattern used in software engineering to divide the back end of an application and the user interface (the front end).

OOP (Object-oriented programming)

OOP is a programming paradigm that uses ‘objects’ and their interactions to design applications.

PHP (PHP: Hypertext Preprocessor)

A general-purpose scripting language that is especially suited for Web development.

Think aloud protocol

Users who have never seen a certain system before are asked to use the system and while doing so talk about what they are thinking.

Table of Contents

Preface.....	iii
Summary	iii
Glossary.....	v
1 Introduction.....	1
2 Project Organization.....	2
2.1 Assignment.....	2
2.1.1 GMAT Test Zone.....	2
2.1.2 Parts.sg website.....	2
2.2 Project Planning.....	2
3 User Experience of the Test Zone.....	4
3.1 First Research Phase	4
3.1.1 Online Questionnaire	4
3.1.2 Paper-based Questionnaire	4
3.1.3 Think-aloud Interviews	5
3.1.4 Interview	6
3.1.5 Conclusion.....	7
3.2 Second Research Phase.....	7
3.2.1 Proposed Changes	7
3.2.2 Think-aloud Protocol.....	8
3.2.3 Interviews	9
3.2.4 Conclusion.....	10
3.3 Final Research Phase.....	10
3.3.1 Interview	11
3.3.2 Conclusion.....	11
3.4 Implementation	11
3.4.1 Files.....	11
3.4.2 Bug Fixes	12
3.4.3 Usability Changes	12
3.4.4 New Features.....	13
4 Parts.sg website	17
4.1 Research.....	17
4.1.1 Global Specifications.....	17
4.1.2 Prototype	21
4.1.3 Requirements	23
4.2 Choosing a Framework.....	24
4.2.1 Frameworks.....	25
4.2.2 Drupal	26
4.3 Implementation.....	28
4.3.1 Modules.....	28
4.3.2 Moderator.....	32
4.3.3 Conclusion.....	34
5 Conclusions and Recommendations.....	35
5.1 Conclusions.....	35
5.2 Recommendations	36
References.....	38
Appendices	39
Appendix A: Questionnaire	39
Appendix B: The Result of the Online Questionnaire.....	42
Appendix C: The Result of the Paper-based Questionnaire	43
Appendix D: Parts.sg specifications	44
Appendix E: Review of the Prototype	48
Appendix F: Planning of Parts.sg.....	50
Appendix G: The Functionality Matrix for Parts.sg.....	51

1 Introduction

Collins Kumarasinghe Associates is a small company based in Singapore. They focus on two markets: standardized testing preparation courses and web development. The focus is mostly on the GMAT (Graduate Management Admission Test). On their website for the GMAT, www.gmat-zone.com, they offer the possibility to practice for the test using an online testing application. This part of the website is called the Test Zone. The Test Zone had been created by their in-house developer but was in need of some improvement. He was very busy with ongoing projects of the web development side of the company and did not have time to work on the Test Zone. This became the first part of the bachelorproject: researching the user experience of the Test Zone, making the appropriate changes and adding new functionality.

A second part of the project was introduced because the scope of the first project was not large enough. This part would focus on the web development aspect of the company. At first the idea was to create a standard code library for the company to use in future projects. An existing framework would be chosen and a setup of a basic website would be made. During the research phase of the second project, a new website was ordered by clients. This website would make it possible for people to search for car parts and add information about these parts. It was decided that the second part of the bachelorproject would now consist of gathering the requirements for the new website and helping implement a part of the website. The goal was to make the programming task easier and faster by correctly defining the requirements and choosing a framework that would be a good match for this website.

This report focuses on the process of this project and on how the design decisions were made. Since a part of the project was changed during the time spent at Collins Kumarasinghe Associates, this will receive some extra attention because it makes the process of the second part of the project a bit vague. The research for the user experience of the Test Zone will be discussed to explain why the changes were made to the application.

After this introduction, one can find chapter 2 devoted to explaining the assignment of this project and the planning made at the beginning of the orientation phase. In chapter 3 the Test Zone is discussed. This chapter is divided into separate sections describing the different research phases. The last section is about the implementation of the changes in the Test Zone. The car parts website research and implementation phases are discussed in chapter 4. Finally, chapter 5 contains some final thoughts about the project.

2 Project Organization

In this chapter the project assignment and the planning of the project is described. It explains why the project assignment was changed after the orientation phase and how this fit into the planning.

2.1 Assignment

The project at Collins Kumarasinghe Associates was divided into two smaller projects. The first project was enhancing the user experience of the Test Zone on www.gmat-zone.com. This involved gathering information from users and implementing the necessary changes. The second project was setting up a car parts website: www.parts.sg. This website was created for clients and at the start of this bachelorproject it was not certain yet that Collins Kumarasinghe Associates would get the assignment. That is why in the Orientation Report the second project was defined as creating a standard code library for websites containing functionality which many of their clients require. This was abandoned once it was clear that the car parts website had to be made.

2.1.1 GMAT Test Zone

The Test Zone on www.gmat-zone.com lets people study for the GMAT by taking mock tests, viewing their results and creating their own tests made out of previously completed questions. This web application needed its user experience improved. To enhance the user experience, the faults had to be found. Users were interviewed and also the think-aloud protocol was used to find any usability problems. Questionnaires did not return a lot of data because the users, who were easy to get in touch with, usually had not used the application yet.

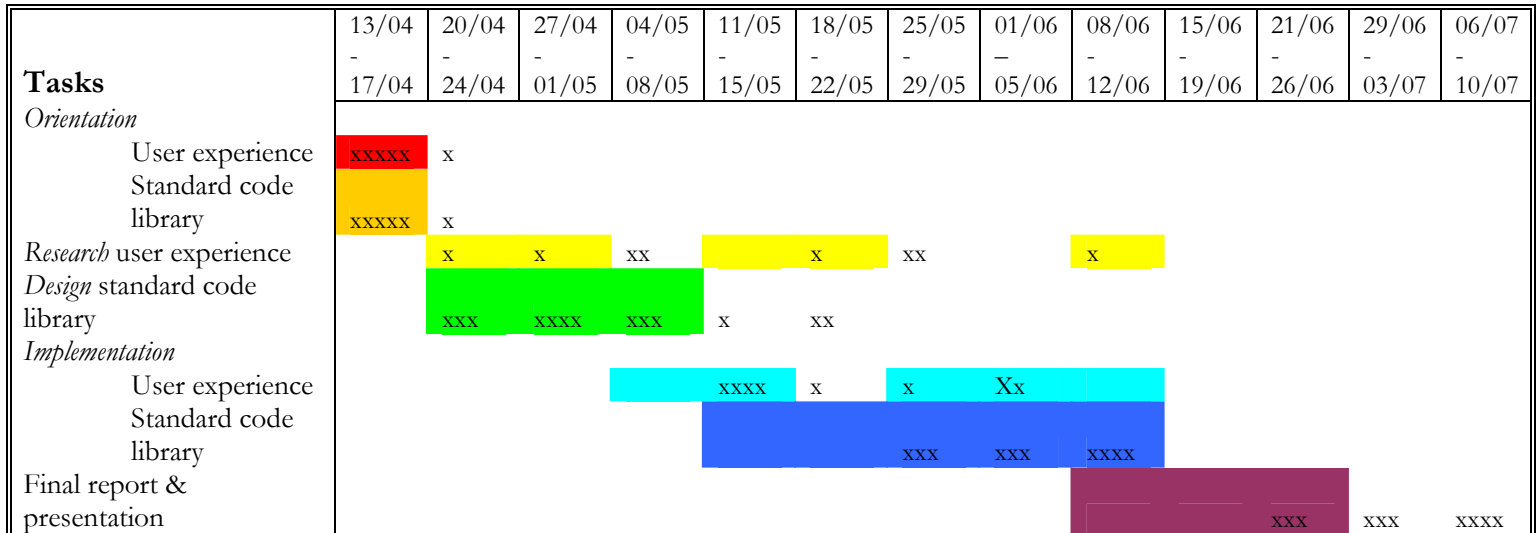
2.1.2 Parts.sg website

Two men wanted a car parts website to make it easy to find car parts and offer these for sale. The task was to gather the requirements of the website and help implement a part of it. The clients had made a document containing what they wanted the website to do. They were interviewed twice. The first interview session was to identify any major differences in ideas. For the second meeting a PowerPoint prototype was made to clear up the smaller issues and to check if everything was understood correctly. This part of the project also required some research on using a different framework.

2.2 Project Planning

The planning of this project was not very easy because two different subjects had to be focused on at the same time. By creating a clear plan during the orientation phase, it was possible to keep an eye on things. A Gantt chart was created and during the project it was updates with x marks to keep track of progress (see figure 2.1 below). The scope of the parts.sg project was a bit bigger than the user experience of the Test Zone project and this is clear to see in the chart. The user experience of the Test Zone took less time because there was already a whole system available while the parts.sg website had to be created from scratch. Creating interviews for users did not take a lot of time either

because most users were interviewed using the think-aloud protocol. The web application did not contain a large set of functions, so most interviews took fifteen to thirty minutes. The implementation of changes also did not require as much programming work as the parts.sg website did.



x means about a day spent on that part of the project

Figure 2.1: the Gantt chart containing the time spent on each part of the project.

Most tasks were not finished in the time planned. This was not a huge problem because the implementation of the Test Zone took less time than expected. The implementation phase of both projects was finished before the end of the internship. The implementation of the parts.sg started two weeks later than initially hoped. This happened because halfway through the design phase of the 'standard code library', as it was described in the Orientation Report and Action plan, the project changed into setting up the car parts website. While the work done for the standard code library could be partially used for the car parts website, some work had become useless and therefore some time had been lost.

3 User Experience of the Test Zone

The first project that will be discussed is the user experience of the Test Zone web application. This was the original task for the bachelorproject but it did not require as much work to count as a complete bachelorproject should. This project was very interesting because it involved a lot of user testing. Most usability problems were found using the think-aloud protocol. A few interviews were conducted with users who had used the web application before. These users were more difficult to get a hold of.

In this chapter the work done during the three research phases will be explained and also the conclusions drawn from the research can be found here. At the end of the chapter the implementation of the changes made to the system will be discussed.

3.1 First Research Phase

The goal of the first research phase was to identify the main problems of the Test Zone on www.gmat-zone.com. At first the idea was to gather data using an online questionnaire. The Rotterdam School of Management helped to find students who had previously taken the GMAT. An email was sent to these students asking them to take a look at the Test Zone web application and to fill out a questionnaire. A paper-based questionnaire was made for the students who attend the GMAT prep sessions at Collins Kumarasinghe Associates. This also did not go as well as planned because most people who attended the sessions had not used the system yet. This did make it easy to find users for think aloud sessions. In the end, one interview was conducted with someone who had used the Test Zone to study for the GMAT.

3.1.1 Online Questionnaire

The online questionnaire was created during the Orientation Phase (see Appendix A) and was sent to the Rotterdam School of Management during the second week. The questionnaire was sent to Master students who had taken the GMAT. An email was sent, asking them to look at the Test Zone web application and to fill in an online questionnaire. While eleven people followed the link to the online questionnaire, only one person actually looked at the system and answered part of the questionnaire (he/she did not even finish all the questions!). Thankfully, the most important table of small questions was filled in (see Appendix B). The person did not disagree with any questions, but answered neutral at only two questions. One of these questions was “I expected to be able to review questions at the ‘Review Section’.” This question is one of the more important questions because later it would appear that most first time users had problems with the ‘Review Section’.

3.1.2 Paper-based Questionnaire

The paper-based questionnaire was handed out to students who attended the prep classes for the GMAT. Most of these students had not used the web application yet, because it is usually used after attending all the classes and when the students are in the last phase of studying for the GMAT. Only one person filled in the questionnaire. The result can be found in Appendix C. This slow start lead to applying the think-aloud protocol, which is discussed in the next section.

3.1.3 Think-aloud Interviews

It seemed most students had not used the Test Zone before, making them good candidates for the think-aloud protocol sessions [17]. They are the perfect users, seeing how they will be using the web application later during their preparations for the GMAT. Three students were interviewed separately. All started on the first page when logged in: the Dash Board. This page shows pie charts which should give the user an overview of their progress. Currently, these pie charts do not work therefore only empty pie charts are shown. Since this is a known defect, it was fixed during the first implementation phase. Below are the three think-aloud protocol transcripts.

"I am trying to find the tests so I am going to go to 'GMAT Tests'. So I see the different tests here and they have different types. They make sense to me." She clicks on a link to one of the tests. "Yeah, this is what I would expect from GMAT practice tests. Do I have to finish the whole test?" (Answer is no, the account she logged into already contained a few completed tests.) She exits the test. "So now I will click on Test History to view my previously taken tests. Okay, looks fine. I see all the tests here. Now I want to see how I did on the tests so I go to 'Review Section'. [Oh, what is this? I see a few questions here, but I am not sure what to do with them. These are not all the questions I took right? There are not many questions on this page." **UP 1.1**] Clicks on the arrows at the top of the screen to go through the different categories but does not find what she is looking for. "Okay, I don't know. I am going to click on 'Math Player' now. [Huh? I have to download something? I don't know why." **UP 1.8**] She clicks cancel and goes to 'GMAT Musings'. "So GMAT Musing is for personal comments, I think. And now I am going to 'Dashboard', so we go back to the first page. Okay." She is showed how to find the answers of the tests she made (at Test History and then clicking on Solution). She is asked to try to review a few questions. ["Oh, I feel like you would review questions at the Review Section. That is what I thought. The title of the page is just misleading, so maybe it should be changed. **UP 1.1**] ['Solution' is also misleading because you review there. **UP 1.2**] Okay, now I can see all the question numbers categorized by correct, incorrect and skipped." She clicks on one of the questions. "So now I see the solution below and I can review here at the top." She clicks on the red X icon and it changed to a green check mark icon. "Oh, this indicated that it is reviewed. What I like is that on the left it groups all the questions in the category chosen. I can choose between all, correct, incorrect and skipped so that is good."

Female 30-39

"These pie charts seem very useful, yes." He scrolls down to the bottom of the Dashboard page. ["The bars at the bottom, what is it? It is unclear to me. Two ways to see the result? But then what would be the difference." **UP 1.4**] He clicks on 'Test History' in the side menu. ["Take these tests again? So I can take the same test again? Would the questions be the same when you take it the second time?" He is told that the questions would be the same. "Oh. I guess I expected different questions because why would you want to answer the same questions twice." **UP 1.5**] He clicks on the 'GMAT tests' link and lists all the different tests. ["I am missing some tests. Is there no reading comprehension test? Because that is a part of the GMAT. Data sufficiency is also not there? I think those two sections are also important so it would help to have more of those tests." **UP 1.6**] He stares at the page for a while. ["Is there not more information?" He is shown how to mouse-over the links to the tests. "Yes. That was what I was looking for." **UP 1.7**] He navigates to the 'Review Section'. ["What I expected to see here were the answers to questions that I had not answered correctly. So you would review these questions here. But I do not see that here." **UP 1.1**] He is shown where he can review the questions at 'Solution' at 'Test History'. ["I see all the questions of the test here and I can click on them. Then I am not sure what to do. I can look at the answers. How do I review? This does not make sense to me **UP 1.3**]. I would rather flag questions for review later while taking the test instead of doing it all after the test has finished because I would not remember which questions I wondered about for a long time." He is shown that by pressing on the 'red X', he flags the question for review later. "This type of reviewing, it does not make sense. I understand how it works now, but when you asked me to try myself, I could not find out what I should do." He then proceeds to click on 'MathPlayer'. ["This is not clear. I do not know what the point of this is." **UP 1.8**] He goes to 'GMAT Musings'. ["What are musings? This is also not clear what it is. Can I send my comments to the gmat-zone trainers here?" **UP 1.9**] He clicks on the 'Dashboard' and is brought back to the first page. "Okay so now we are back at the beginning." [He is asked if he thought he would be brought back to the first page now. "No, I did not expect it. It also does not really make sense." **UP 1.10**]

Male 20-29

"The first page looks all right." She clicks on the 'GMAT Tests' link. ["Uh, I do not see everything right away. I prefer the tests placed not as randomly as it seems now. So start with easy then medium then hard for each type of test." **UP 1.11**] She goes to the 'Test History'. "This looks good. Yes, it is what I expect." [She goes to the 'Review Section' and skips the 'Solution' link. "Fine," but when asked where she would be able to see the answers to the questions, she does not know. **UP 1.3**] [She also does not understand why the certain questions appear here. **UP 1.1**] [She clicks on the 'MathPlayer' link. "Should I press okay? No, I do not know what it is for." **UP 1.8**] [She presses cancel and goes to the 'GMAT Musings' page. "Can I send in questions here? What are musings?" **UP 1.9**]

Female 30-39

In these three transcripts, different usability problems are highlighted and displayed below.

Usability Problems

- 1.1 Review Section did not show what was expected. The users thought they would see the answers to the questions here.
- 1.2 The Solution page does not invite users to review the questions there.
- 1.3 Confusion about how to review the questions on the Solution pages.
- 1.4 The pie charts and bars at the bottom of the Dash Board confuse the user.
- 1.5 Confusion about 'Take this Test again'. Thinks it might be a test with different questions.
- 1.6 Misses some test categories or does not understand them.
- 1.7 Mouse-over to see the information about the tests is not found.
- 1.8 It is unclear what the 'Download MathPlayer' is for.
- 1.9 It is not clear that GMAT Musing are comments for yourself. Thinks the comments are sent to trainers.
- 1.10 Does not make sense that the Dash Board is the last link in the menu, but the first page of the web application.
- 1.11 Prefers the tests to be ordered differently.

3.1.4 Interview

One interview was conducted with a woman who had been using the Test Zone. It had been a long time since she used it, so she was unable to answer all the questions. The interview, shown below, did not result in a great deal of useful information. However, it does show that most people are able to find most functionality after trying to use the program. The questionnaires also support this view. It is not clear right away what is expected of the user, but after looking at parts of the web application for a little while, they are able to understand how to use it.

1. *Do you use the Test Zone on www.gmat-zone.com to practice for the GMAT?*
Yes.
2. *How many tests have you taken on the website?*
4
3. *Did you have any problems when you first used it?*
Yes, I could not see the Math Figures (still cannot). The MathPlayer was downloaded and installed, but it still would not work.
4. *Was anything confusing at first?*
No, it was pretty straightforward.
5. *If so, what was?*
NA.
6. *Are any parts of the website still confusing? Which ones?*
NA.
7. *What feature(s) do you use?*

- GMAT tests and viewing the answers at the Solution. Flagging questions for review are used, but no new tests are created.
8. *What is your favorite feature of the website and why?*
You can do it during your own time.
 9. *Which features don't you use? Why not?*
I do not create my own tests. No particular reason for this.
 10. *Do you like the review feature?*
Maybe, yes.
 11. *Did you find it easy to review questions?*
I only reviewed questions when I did something wrong.
 12. *Did it make sense to you that you have to go to the Test History page to review?*
I do not remember because it was too long ago.
 13. *Are there any other features you think the website should have?*
It's fine.
 14. *Do you have anything else you would like to say or discuss about the website?*
I would like a reset password feature.

Female 30-39

3.1.5 Conclusion

The web application seems to have a few issues when a user interacts with it for the first time. The Test Zone is not very intuitive and lacks any kind of explanation. The functionality of the website is there, but sometimes the users cannot find it. Eventually the users do seem to understand how the web application works. The user experience of first time users can be greatly improved. More seasoned users seem happy with the Test Zone, so they do not call for many changes.

The biggest issue most first time users encounter is understanding what the functions of each page are. 'GMAT Musings' and 'Download MathPlayer' caused some confusion. Also, the 'Review Section' can be interpreted two ways: the answers to the questions can be reviewed to study later, or the questions can be viewed again for studying purposes. Both actions can be seen as reviewing which is why the 'Review Section' should be renamed.

All in all, most issues revolve around the fact that things should be displayed a bit clearer. By adding little bits of text at the top of each page, explaining what the page is for, would probably already help considerably. There are no core functionality changes needed. The only extra option required is that users can get a new password for their account.

3.2 Second Research Phase

During the previous research phase some users noted how some extra functions might be useful. These functions will be discussed in more detail here and other users will be asked if they would like to see these functions in the application. During this phase the changes made to the system will also be tested. These changes consisted of adding page titles to every page and some text to explain the purpose of the specific page. The pie charts were fixed, the 'Review Section' was renamed to 'Practice Section' and the 'Download Mathplayer' link was moved to the 'GMAT Tests' page by adding a button with the text 'Install Mathplayer' on it.

3.2.1 Proposed Changes

A plan for the future is that full length GMAT tests will be available on the website. The tests that are currently on the website allow users to focus on specific aspects of the tests, such as Verbal or Math. Some tests have comments indicating how much time a student should spend on each

question. These tests are only for practice, so they do not have to simulate the GMAT perfectly. Some functions can be added to enhance the learning experience of the users.

1. A button allowing users to flag a question for later review can be placed in the tests. This is probably better than flagging the questions after taking the test, because people tend to forget which questions they had problems with.
2. Add a progress bar indicating how long they are spending on each question. For instance, easy questions should be answered in 2 minutes so a bar counting showing the 2 minutes fill up would add some pressure to the users. This might be good, because they will be under a lot of pressure during the real GMAT test too.
3. Currently it is not clear that users are actually able to submit questions to the trainers of the courses. Would they like this option? If they do, it should be clearer in the web application.

These functions had been discussed with the company and they agreed that it would add to the experience. Of course, only interviewing the real users would reveal if these plans were any good.

3.2.2 Think-aloud Protocol

To see if the changes made any difference to the users, new users had to be observed again. Testing the new website should show if the changes caused any new problems and if they solved the previous ones. Below the transcripts of the think-aloud protocol can be found.

The user starts at the first page: 'My Dashboard'. "[I am trying to see what sources the pie charts are built up from. I want to know if the pie charts show all the results. **UP 2.1**]" He scrolls down. "What are these bars? Does this mean other results are here? Oh, these are the best results. [What does the dark blue bar mean? Why does the Sentence Correction bar not have this darker color? I think I understand that I scored 80/100 on the Critical Reading, but I do not understand what happened at Sentence Correction. **UP 2.2**] So now you wanted to know what I expect from the next page I go to? I will go to GMAT Tests. What I think will be there are sample questions for each category. Also explanations, about what to do at each category, will be there and also some practice questions. So now I see the page and well, it was not quite as expected but it does make sense. I can take tests here, as the practice questions I said before. [I miss the number of questions each test has to be able to calculate how much time I should spend per question. **UP 2.3**]" He is asked about the 'Install MathPlayer' button. "I saw it and I guess it is clear, but maybe [more focus should be on 'install' because you do not really see it in the text. **UP 2.4**] I will go to 'Test History' now. It looks like I would think it should look like. Ranking and flagging questions look fine. [So what I do not like is that I have to close the Solution page and click on it again to go to the correct answers if I was viewing the incorrect answers. **UP 2.5**] What I expect of the 'Practice Section' is to view tutorials and in detail what should be done on each test section. What I see here is different, but I do like it. I find this very good. 'Personal Notes' I would not use, but I suppose others might. I just will not."

Male 20-29

The next person was very quiet and only said "hmm" and "uhu" about the pages. However, he did ask questions if some functionality would be possible. These questions will be listed here and discussed with the company.

1. Can the pie chart break down the question categories even further? Like say for quantitative tests, it would show if you are weak in algebra or something?
2. I would like to see more information about the math problems such as how much percent is statistics or algebra, etc.
3. Is there an option to ask questions? Or like, if I am late for a Master class because I cannot go the first hour, then I can watch the class online and ask questions while the class is going on and the trainer would respond?

Male 20-29

"Quite clear this first page that I am looking at. In terms of layout, why does it say 'Hult School'?" It is explained that this is part of the main website. She proceeds to the 'Test History' page. "Pretty straight forward for me. The Practice Section... also not really any problems. [Student Profile, but there is no profile? Maybe they should add some profile fields or something. **UP 2.6**]" She goes to the 'Personal Notes' page. "No personal notes. Oh, I can key them in. Would I see it if I do?" She is told she would see them. She is then asked to find the answers to questions of tests. She goes to 'Test History' and then to 'Solution' without a problem. She is asked to look at the answers. "I think the question details should look more like the real GMAT test."

Female 20-29

"[Would there also be a Reading Comprehension section too? (He is referring to the pie charts) **UP 2.7**] [Hm, one thing actually. It would help if this page would show the different difficulty level too in the pie charts. So you would know if you are able to do the hard questions too. **UP 2.8**]" He goes to the 'GMAT Tests' page and clicks on a test and exists again. "It is self guiding, how it should be right?" He goes to the 'Test History' page. "Okay, so you can leave your tests where ever it is, right?" He views the 'Student Profile' and 'Personal Notes'. "Looks fine to me. I would expect more analysis on 'My Dashboard', but the rest is fine."

Male 20-29

Usability Problems

- 2.1 Not directly clear what information is used to fill in the pie charts.
- 2.2 The graph bars at the bottom do not show the best results in a clear manner.
- 2.3 User would like to see the number of questions to know how much time should be spent on each question.
- 2.4 The user thinks there is not enough emphasis on the 'install Mathplayer' function.
- 2.5 Confusion about how to switch between correct and incorrect answers on the 'Solutions' page.
- 2.6 User would like more information on the 'Student Profile' page.
- 2.7 Confusion about the pie chart categories.
- 2.8 User would like the pie charts to be built up differently by also using difficulty levels.

3.2.3 Interviews

A few people, who had used the application before or who had seen it during the think-aloud protocol session, were asked three questions about the new features that could be added to the application. The questions and the responses are displayed below.

Would you like having the option to flag questions for later review while taking a test on the Test Zone?

"This would be okay. Maybe it would be useful to have different levels for people who want to get different grades (intermediate users, expert users, etc)." Female 30-39

"Yes, this would be good." Male 20-29

"It would be helpful if there was the possibility to see how long I took on a question and such, so this would add to that." Male 20-29

"No, I want it to mirror the real tests as closely as possible." Female 20-29

"Yes" Male 30-39

Would you like seeing a progress bar indicating the amount of time you should spend on a question? For example, you should spend two minutes on an easy sentence correction question. The bar would show filling up the two minutes.

"You mean like an indicator? Yes, that would also be okay." Female 30-39

"Yes, this would be very nice as long as students are not forced to answer the question before the bar fills up." Male 20-29

"Yes, that would be helpful." Male 20-29

"No, it should follow as closely to the real test as possible." Female 20-29

“Ah, yeah, that would be useful.” Male 30-39

Would you like to be able to ask questions through the website? The trainers would receive these and email or call you back.

“Yes, this would also be good. If I had known the GMAT Musings had that function, I would have certainly used it.” Female 30-39

“Yes, that would be useful. Could you also select which test and which question you are referring to? Then the student can add the question and the trainer could see both.” Male 20-29

“Yeah, I think that would be helpful.” Male 20-29

“Yes, okay that is fine.” Female 20-29

“Uhu.” Male 30-39

A woman who was sitting next to the last man who was interviewed agreed with him on every question. From the interview results it becomes clear that the users of the web application would welcome the proposed changes. One woman would prefer the tests to correspond to the real GMAT tests as closely as possible. The company is currently working on creating ‘full length’ tests that simulate the real GMAT test as stated before. The tests on the ‘GMAT Tests’ will be merely for practicing certain types of questions of the GMAT and therefore it will not have to mimic the GMAT tests.

3.2.4 Conclusion

The problems with the old ‘Review Section’ are gone and the users do seem to understand the function of that page a lot better. This was the biggest problem for most first time users and it is clear that it is no longer an issue in the application.

Since the pie charts work, a lot of users are still a bit confused by them. Users miss the ‘Reading Comprehension’ test category. The problem persists because there are no Reading Comprehension tests in the database yet. Also, some users would like more detailed pie charts. Usability Problem 2.8 shows that a user would like to see the different difficulty levels in the pie charts on the ‘My Dashboard’ page. These pie charts can be found on the ‘Test History’ page when they mouse over their attempts. Since the users started at the ‘My Dashboard’ page, they did not know that this function exists yet. However, on the ‘Test History’ page, most users did not discover this functionality, so it would be useful to inform the user about it.

The new functionality was well received by the users and was therefore implemented. The problems on the ‘My Dashboard’ page are likely to stem from the fact that the pie charts were filled before the users started using the web application. When they take a test, they will see the result in the pie charts. This helps them understand how the pie charts are set up and therefore problems 2.1, 2.2 and 2.7 are all related to that issue. Problem 2.3 should be fixed by adding the progress time bar on the testing pages. Problems 2.4 and 2.6 are not of extreme importance, but problem 2.5 is a little bit more important. No solution was found for this problem during the project.

3.3 Final Research Phase

During the last research phase the goal was to discover if the users liked the placement of the added functionality during the tests. This contained the option to rate the questions while taking the test and also seeing the time spent on the question. Users were asked if ‘time spent’ and ‘flag question’ should be displayed elsewhere or if they liked the current layout.

3.3.1 Interview

To examine the placement of the new functions during testing, users were shown the current situation and asked where they would prefer to see these options. The results can be viewed below in figure 3.1. ‘Time spent’ is indicated with a T and ‘flag question’ with a F.

00:19:24		2 of 10		TT		Exit	
Rank this Question: FFF ☐ Easy ☐ Medium ☐ Hard Flag this question for review: ☒		Machine A produces 30 widgets an hour, Machine B produces 50 widgets an hour, Machine C produces 70 widgets an hour. How long will it take all three working together to produce 50 widgets? ☐ 15 minutes ☐ 20 minutes ☐ 10 minutes ☐ 1 hour exactly ☐ half a hour				Answer Confirm Next	
		Time spent on question: 00:00:25 T					

Figure 3.1: The preferred placement of the new features.

3.3.2 Conclusion

The users were happy with the placement of the ‘flag question’ option. The ‘time spent’ option should be displayed in the top black bar. If the display of the ‘time spent’ is changed into a progress bar, it would not fit into the top bar. In that case, the current placement is sufficient. The progress bar was not implemented due to time constraints caused by the www.parts.sg website project and was left as a task for the current developer at Collins Kumarasinghe Associates.

3.4 Implementation

The Test Zone web application required several changes to be made. Before these changes to the code could be made, the set up of the application had to be understood. Eventually only three files needed to be edited. A few bugs were fixed, changes required to enhance usability were made and also some new features were added.

3.4.1 Files

The Test Zone consists of a few files. One set of files contains all the functionality for the administrators of the website. The administrators are in charge of placing students in classes, allowing them to view tests, and adding tests to the website. This part of the application was not researched. The files that were important to change were the php file containing the functions to get the correct information from the database (testzone.php), the php file that used the results of testzone.php file to display the html pages (testzone.html.php), and the language files that contain the different

variables with the correct text. In the php code only the variables are referenced when text is needed to ensure that different languages can be used easily.

An example of a part of a language file:

```
DEFINE('_TZ_EXIT', 'Exit');
DEFINE('_TZ_EXIT_TEST', 'Exit this test and return to GMAT Test Zone?');
DEFINE('_TZ_START_TEST', 'Start this test?');
DEFINE('_TZ_FINISH', 'The Test is over, click the button to return to your scores.');
```

The testzone.php file and testzone.html.php contain all the functions for the Test Zone application. The use of comments was lacking but thankfully the developer was still aware of what functionality could be found where. Each php function related to a page of the Test Zone, but this did not mean that they always had the same name as the page! For instance the function `show` gathers the information for the 'My Dashboard' page.

3.4.2 Bug Fixes

Before the project started, some time was spent going through the web application to find faults and things that could be improved. Two faults were found and discussed with the company. These faults were identified as bugs and were the first things to be fixed. The bugs were:

1. The Pie Charts on My Dashboard did not work.
2. When the mouse-over information was longer than the webpage, the text could not be seen.

To fix the first problem, the My Dashboard code had to be reviewed. What the code does is gather all the tests results of a user per test category. These results are then used to calculate how many questions were answered correctly and how many were incorrect. The skipped questions could be calculated with the previously obtained numbers because the total number of questions is already available. These numbers are then used to fill in the pie charts using a Google application. This resulted in about 300 lines of undocumented code. The programmer, who had created this, helped to explain the code.

The fault ended up being that the system checked for a variable, `$latest_testing_str`, but this was never defined. This variable keeps track of which tests had been taken by the user. It seemed the programmer had forgotten to finish this function because the only thing that needed to be added was the following for-loop.

```
for($k = 0;$k<count($testings);$k++) {
    $latest_testing_str .= $testings[$k]->id.', ';
}
```

The second bug was that the mouse-over texts explaining the tests would sometimes fall outside the visible area of the page. This was easily fixed by changing the JavaScript variable that placed the text below the link to the variable that places it below the link as long as there is enough room left; if this is not the case, the text is placed above the link.

3.4.3 Usability Changes

The problems the users encountered during the first think-aloud protocol sessions are listed below. To remove these problems from the application, different changes were made.

Usability Problem	Solution
Review Section did not show what was expected. The users thought they would see the answers to the questions here.	The Review Section was renamed to 'Practice Section' to remove the confusion about the word review.
Solution page does not invite users to review the questions there.	A text was added at the top of the 'Test History' page indicating that users can flag questions for later review there.
Confusion about how to review the questions on the Solution pages.	The word 'Review' was replaced by: 'Flag this question for later review:'.
The pie charts and bars at the bottom of the Dash Board confuse the user.	A text was added to the top of the page indicating that the pie charts represent the total scores, and that the bars show the best scores.
Confusion about 'Take this Test again'. Thinks it might be a test with different questions.	No changes made to correct this.
Misses some test categories or does not understand them.	Content related issue. No changes made because the company has to create more tests.
Mouse-over to see the information about the tests is not found.	A little text on the GMAT Tests page was added explaining that more information about the tests can be found when you mouse over the names.
It is unclear what the 'Download MathPlayer' is for.	A button was added to the GMAT Tests page and a text was placed next to it explaining why it was important to download this. The link in the menu was removed.
It is not clear that GMAT Musings are comments for yourself. Thinks the comments are sent to trainers.	GMAT Musings was renamed to 'Personal Notes' and a bit of text was added to indicate that the notes are only for the user.
Does not make sense that the Dash Board is the last link in the menu, but the first page of the web application.	This was easily changed at the administrative back end of the website.
Prefers the tests to be ordered differently.	This is quite difficult to change because the tests do not have a certain order in the database and the IDs of the tests are also not ordered in any way.

Almost all changes were made to the testzone.html.php file because as noted before, most of the functionality of the system already existed and the display of it just needed to be tweaked. All pages were given a page title, so the users would know on which page they are. Also, a small text was added to explain what functionality can be found on the page. The 'Review Section' was renamed to 'Practice Section' and some other texts referring to 'review' were changed to 'practice'.

3.4.4 New Features

The proposed features were added to the application during the last implementation phase of the project. The two features for the testing component of the application, keeping track of the time spent on the question and being able to rank the question while taking the test, marked the first time JavaScript had to be used.

Before implementing the progress bar, a simple timer was used to display the main functionality. The progress bar was never implemented but the timer was enough to test the function. To create the timer a few things had to be programmed. First, Javascript had to keep track of the minutes and update this on the webpage. This was similar to the code that kept track of the time left of the test.

```
function startQuestionTimer()
{
    timeSpentQuestion = 0;
    timerQuestionID = setTimeout("updateTimerQuestion()", 1000);
    document.getElementById('timeQuestion').innerHTML =
        displayTimeString(timeSpentQuestion);
}

function updateTimerQuestion()
{
    timeSpentQuestion = timeSpentQuestion + 1;
    document.getElementById('timeQuestion').innerHTML =
        displayTimeString(timeSpentQuestion);
    timerQuestionID = setTimeout("updateTimerQuestion()", 1000);
}
```

Next, the timer had to be displayed using only the following bit of code in the testzone.html.php:

```
<td>Time spent on question: <div id='timeQuestion'></div></td>
```

The timeQuestion div was set in the Javascript functions so this was very simple to display.

The time spent on a question also had to be displayed on the Solution page. The time was saved in seconds as a whole number in the database so it needed to be rewritten to a human readable format. This code transforms the seconds to hours, minutes and seconds. Questions that had been completed before this function was added, had " as their entry in the database. This was checked to be able to show that the time spent on the question was unknown.

```
if ($row->timeSpent == '') {
    $row->timeSpent = 'unknown';
}
else { //display the time correctly
    $seconds = $row->timeSpent;
    $dSeconds = $seconds%60;
    $minutes = ($seconds-$dSeconds)/60;
    $dMinutes = $minutes%60;
    $dHours = ($minutes-$dMinutes)/60;
    $row->timeSpent = (($dHours<10?'0':'').$dHours ).':'.(
        ($dMinutes<10?'0':'').$dMinutes ).':'.(
        ($dSeconds<10?'0':'').$dSeconds );
}
```

To get the results from the database, they had to be saved there first. This was done in the testzone.php file. In the function called **testing**, the results of the answers to the questions were added to the database. A field called timeSpent was added to the **testzone_testing_detail** table in the database. The queries referencing that table had to be updated. The time spent on a question was added to the form that sent all the data about the question. This could then be used to place in the database using the query below.

```
$query = "INSERT INTO #__testzone_testing_detail(testingId, quesId,
selectedOption, questionResult, quesNo, timeSpent)
VALUES('.$testingId.', '.$questionId.', '.$rdoAnswer.', 'CORRECT',
'.$currentQuestion.', '.$timeSpentQuestion.')"
```

If the time of the test ran out, the questions that had not been finished yet had to be marked as skipped by setting the time spent on the question to 0. The code below shows how this was implemented.

```
<input type='radio' name='ranking' id='ranking' value='2'  
onclick='make_ranking(\"\"".$mosConfig_live_site."\", \"\"".$option.\""\",  
"$testingId.", "$question->quesId.", this.value, -1);' " ".$medium."  
>"._TZ_MEDIUM."&nbsp;&nbsp;&nbsp;&nbsp;&br>  
  
<input type='radio' name='ranking' id='ranking' value='3'  
onclick='make_ranking(\"\"".$mosConfig_live_site."\", \"\"".$option.\""\",  
"$testingId.", "$question->quesId.", this.value, -1);' " ".$hard."  
>"._TZ_HARD."<br>  
  
"._TZ_PRACTICE." <a href='#><img id='review_img' src='".$simg.'" border='0' onclick=\"set_review('".$mosConfig_live_site."', '$option.', '$testingId.', '$question->quesId.', '$flag.');\">  
></a>;  
?>  
</td>
```

Javascript was used to save the ranking. This was also used on the Solution page but it did not work right away. The Solution page used a certain variable to save the ranking and flag with but this was not available in the testing function. The variables were changed and the flagging and ranking worked perfectly.

The final function was actually already in the system. The GMAT Musings page actually sent what people filled in to the trainers of the GMAT Zone. Some miscommunications lead to the belief that this was not the case. Now only the title of the page had to be changed from Personal Notes to Ask a Question. The Personal Notes page was created by adding an extra function to the testzone.html.php and testzone.php called **showPersonalNotes**. It contains the same functionality as Ask a Question but the part where the email is sent to the trainers is deleted.

4 Parts.sg website

The second project of this bachelorproject was partly implementing www.parts.sg, a car parts website. This website had to make it easy for people to search for car parts and to find workshops where they could buy these parts. The goal was to define the requirements for the website and start programming with the developer at the company. After two meetings with the two clients a PowerPoint prototype was made to check if everything was understood correctly. After that a functionality matrix was created and more detailed UML class diagrams were made.

Another important part of this project was choosing a framework to create this website with. The company had already used Joomla for various projects and was interested in trying something new. Drupal was selected and this required understanding how to use the framework. Some functionality of the website could be implemented using existing code grouped in something called a module. Some modules were edited to fit the exact needs of the clients.

4.1 Research

The car parts website was ordered by two men who are big car fans. These clients wanted a website that would make it possible to easily find car parts. The website divides parts into different search categories: car brand, part brand and part type. People interested in car parts as a hobby but also people working in the car part industry would visit the site. Car part workshops could advertise the parts they offer and normal users could list the used parts they have for sale. The user community would supply the information about the car parts, much like a wiki.

How to create this website had to be researched. The clients created a document containing their view of the website which can be found in Appendix D. This document did not contain enough information to proceed with implementing right away. UML diagrams had to be created and some extra meetings with the clients were needed to assess their exact wishes. A PowerPoint presentation was created to simulate the website. This prototype made it easy to evaluate if the system would fulfill the clients' requirements.

4.1.1 Global Specifications

After the initial meeting with the clients, there was a good idea of what they envisioned. They had already made a document containing their view of the website. They had also presented an Excel prototype to show the main actions of the website. Before the second meeting it was important to define more specific requirements for the website. This was done using Use Cases and very simple UML diagrams. First, the different types of users and their actions on the website were identified and added to use cases.

There are a few different types of users of the website. Visitors of the website, who are not logged in, do not have very many actions. Logging in, registering, searching and viewing content are possible. Viewing all the parts is available to visitors. They can also view comments, workshops and reviews. They cannot, however, edit parts or comment on any content of the website.

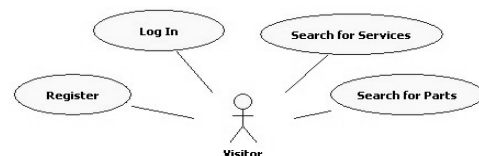


Figure 4.1: the use case of the Visitor.

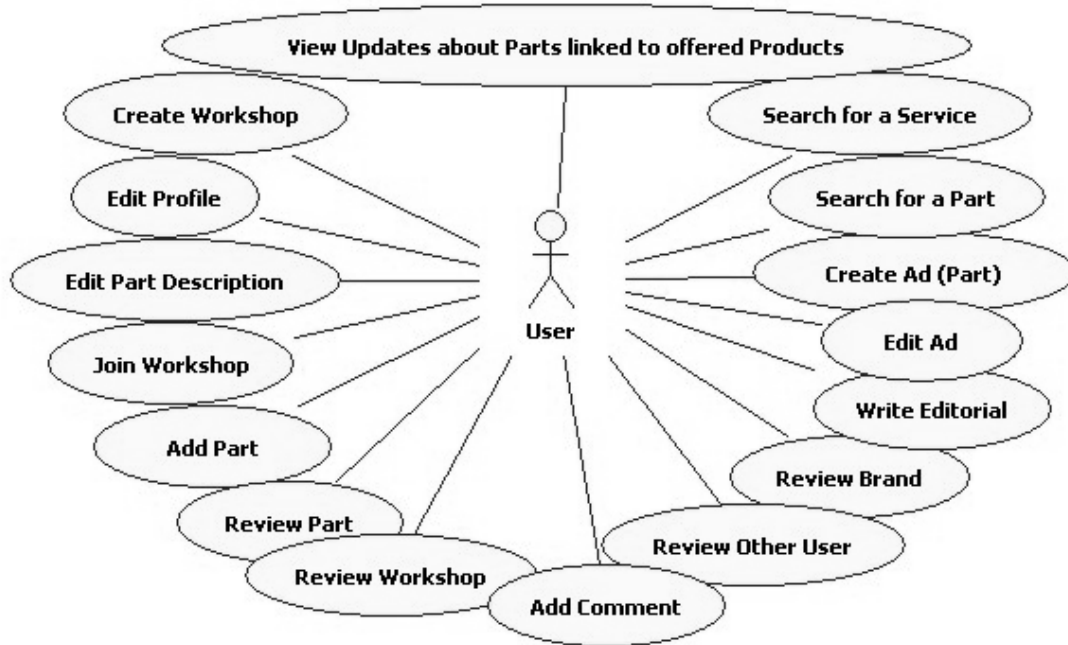


Figure 4.2: the use case of the Registered User.

Registered users have more options once logged in than visitors do. They are able to add comments and rate workshops, review other users offering products, review parts and brands. They can add comments about workshops and parts. They can edit the description of parts and create new parts. Other users are notified of these actions if they have the edited content in their 'Watch List'. The registered users can also join an existing workshop or create a new one. Joining a workshop requires that the users with the appropriate permissions in the workshop accept or decline the new user request. Users can also write editorials but these sent to moderators before being published on the website.

In addition to editing parts, the other important difference between the registered user and the visitor is that registered users can offer parts for sale. These secondhand parts would be linked to the main part being described on the website.

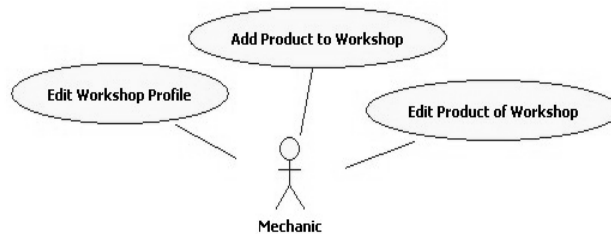


Figure 4.3: the first use case of the Mechanic.

Mechanics are the registered users who have been accepted into a Workshop and who offer certain services. Other users are allowed to rate and place comments about mechanics. After the first meeting it was not clear if a mechanic always had to be part of a workshop. Perhaps freelance mechanics were also allowed on the website. This was discussed during the second meeting with the clients and it was decided that mechanics always had to join a workshop. If they wanted to be freelancers, they should add that to their workshop description. At first it seemed that mechanics also

had permission to edit what parts or services a workshop offers. Later it was decided that they could have this option, but it is not the default. Mechanics can also be managers of a workshop as described below.

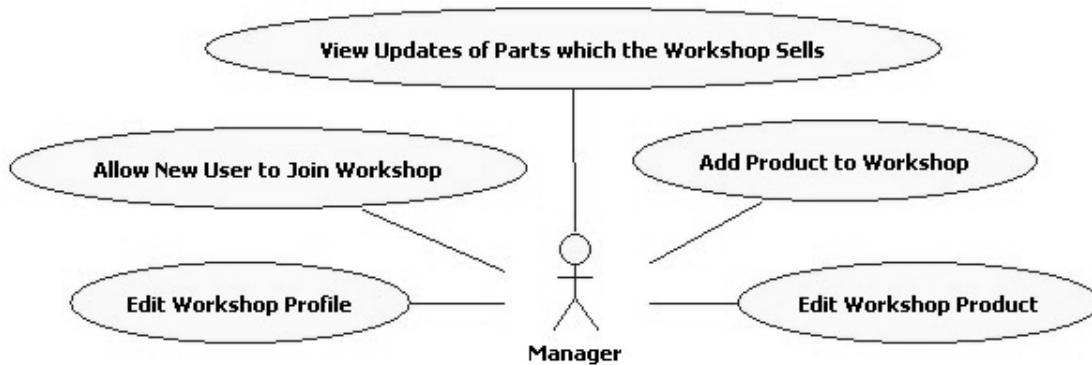


Figure 4.4: the use case of the Manager, later this was changed to Moderator and extra cases were added.

A Manager of a Workshop is really a moderator. The manager is in charge of accepting users as mechanics of a workshop and receives updates about the parts which the Workshop offers for sale. Managers can add parts to the workshop and edit the prices. They can also edit the workshop profile. These permissions are not connected so a manager can have only one of these options or all options. The creator of the workshop has all the options and can give permissions to other members of the workshop. It was unclear if a workshop had only one or many managers but this was discussed and multiple members could have moderator permissions. At first it also seemed like managers were a kind of super mechanics, but after the second meeting this view was changed. Later on during the research the name Manager was changed to Workshop Moderator.

A Workshop is a group of mechanics and normal users. Anyone in the workshop can have zero to many permissions. Only the creator of the workshop has all the permissions by default. A workshop can be rated and comments can be placed about the workshop. A workshop can also offer services. In the beginning the idea was to have one global entity of a Service or of a Part. These would both be classified as Products when a user or workshop offered them. Registered Users and Workshops can own Products, which are ads for Parts or Services. These ads refer to the real Part or Service. When the real Part or Service is changed, anyone who has a Product referring to it will get an update about the change.

This was not exactly what the clients wanted. It was decided that users can offer secondhand parts but no services. A workshop can offer new parts for sale, but they only need to fill in the price of that part. Services are very different per workshop and should have many editing options. Because these services vary so much, only a description and name of the service is maintained globally. Secondhand parts also have more attributes than just a price. The age of the part, the state of the part and the description are examples of attributes a secondhand part has but a new part offered by a workshop does not need.

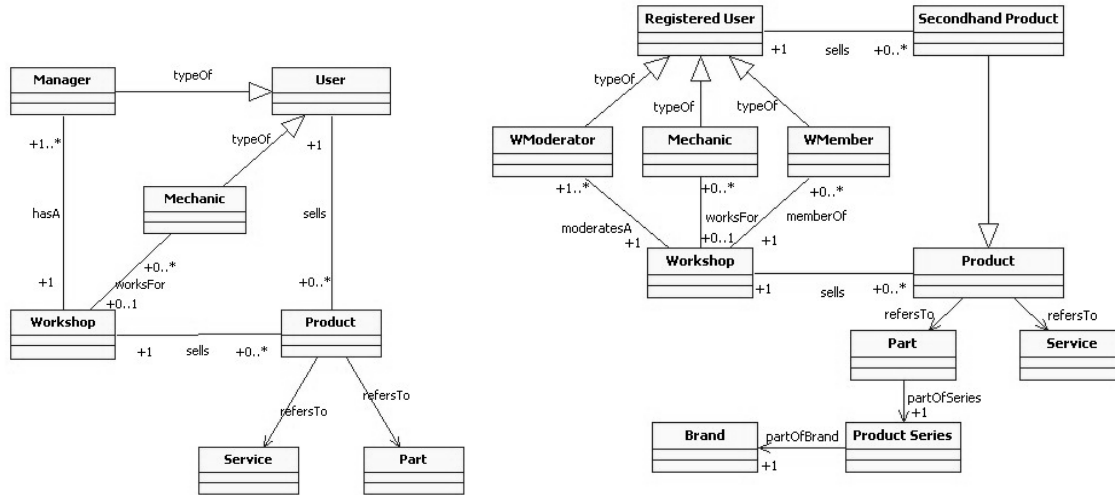


Figure 4.5: the transition of the UML diagram for parts.sg.

The two UML diagrams above (figure 4.5) show the changes made after the second meeting. As discussed before, a Manager was renamed to WModerator (Workshop Moderator). It was also decided that other users were allowed to join a workshop but these users would not have to be a mechanic or have moderator permissions. These users are called WMembers (Workshop Members). In the second UML diagram it is also clear that a Secondhand Product cannot be offered by a Workshop. Later the whole notion of a Product would disappear because users are not allowed to offer services.

After a third meeting the UML diagram was transformed into the diagram on the next page (figure 4.6). Here it is clearer that users cannot offer services. The Secondhand Product was renamed to Used Part and the parts a Workshop offers are called WParts. Likewise, the services offered by workshops are named WService. Services had not been defined very clearly in the previous stages of the research. This was discussed in more detail during the third meeting. The clients explained that usually services are offered in a package. Let's take a car grooming service as an example. Workshops often have different prices for differently sized cars. The grooming of a small car is cheaper than that of a large car. Maybe different waxes are available for the grooming service. This should all be displayed on the website by using a Package. This is a small data item containing a name, description and price. These packages are displayed on the WService page.

This UML diagram also shows more details about the Parts. Parts can be used in different cars and these cars can have different brands. A Car is always linked to a Brand. The same goes for a Product Series. Parts are usually produced by a company, in this case the Part Brand. These companies have Product Series to match parts that work together or are similar. A Part is always part of a Product Series and a Product Series is always part of a Part Brand.

Finally, the Workshop Member was removed from the UML diagram. It was decided that if a user in a workshop is not a mechanic, the user is seen as a moderator. A normal user would have no permissions therefore the user would be a moderator without any moderator tasks. Users can always be granted permissions later.

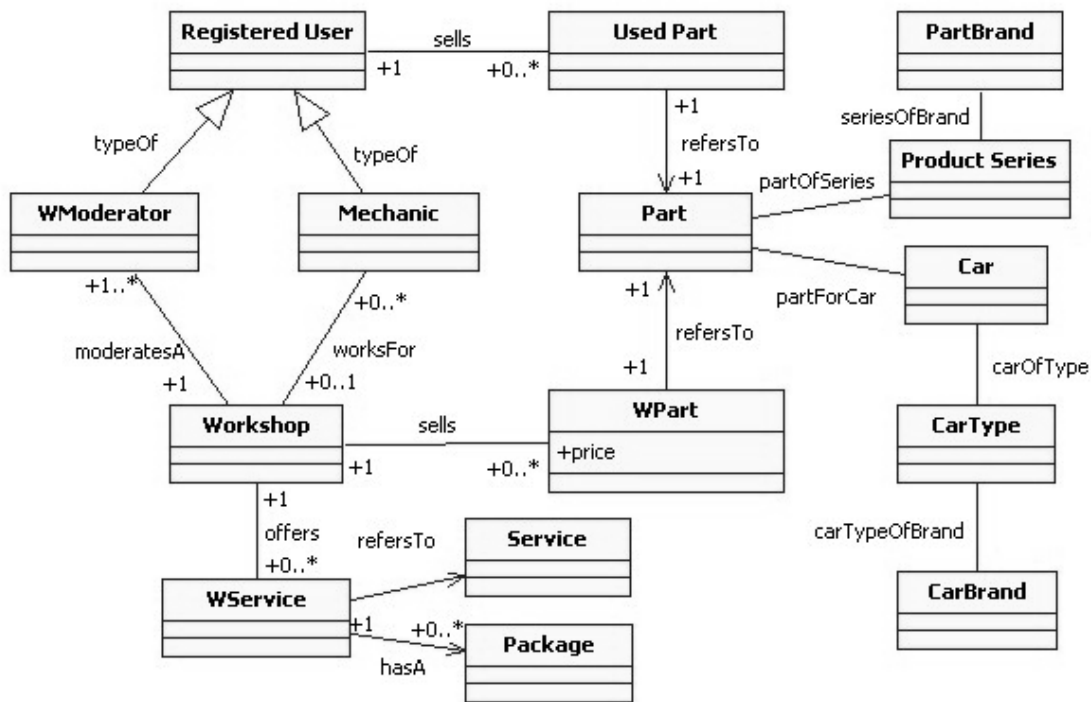


Figure 4.6: The final UML diagram.

What is not shown in the UML diagram is the Watch List. The clients thought the users might be interested in keeping track of certain parts or other content on the website. The watch list is a small list placed on the web pages showing a collection of parts, services and workshops. The users can then add a content item to their watch list. If they want, an email is sent to them when any part of the content changes. Perhaps they are interested in a used engine. If it is sold, the interested users would be notified. If they would not like to receive emails, the changes would only be visible by accessing the website. All changes made to items on the watch list will be displayed on the Recent Updates page.

Another important feature of the website is that when new parts or services are added, it is checked that it does not already exist. This would be done using Ajax to show a list of parts that match the description of the new part supplied by the user. The user is then able to check if the parts listed match the part he/she is adding. If the list is not empty when the user clicks on the 'submit' button, the user will be asked if he/she is sure that this part is really a new part.

4.1.2 Prototype

To check if what the clients wanted was correctly understood, a PowerPoint prototype of the website was made. It was shown to the client during the third meeting and contains almost all the functionality the site would have once implemented. The prototype was not focused on any layout design aspects. Someone else at the company was in charge of the look and feel of the website.

The clients had created an Excel document which showed how they thought the user would browse the website. It had many comments about the functionality and, combined with Appendix D, it was very useful. The prototype was based on those two documents and the things discussed with the clients during the first and second meetings.

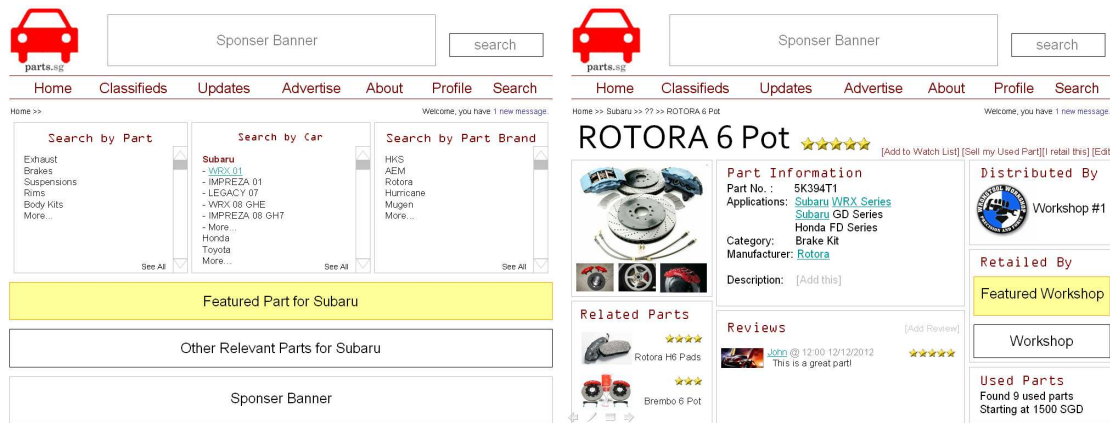


Figure 4.7: Screenshots of the prototype.

Creating the prototype highlighted what parts of the system had not been defined very well yet. The global parts had been discussed during every meeting and what was expected by the clients was obvious. They wanted something like a wiki where users are able to edit the different attributes of the parts. A few examples of these attributes are the cars the part can be used in, the description of the part and the part number. After the first meeting a lot of time was spent on defining the different user types and how the workshop would function. The services seemed to be forgotten until the prototype was made. It showed that services had not been defined very well by the clients and there were some things still unclear about them. Would services also have a wiki-like page? Did workshops have their own service page? These things were added to the prototype to spark the discussion.

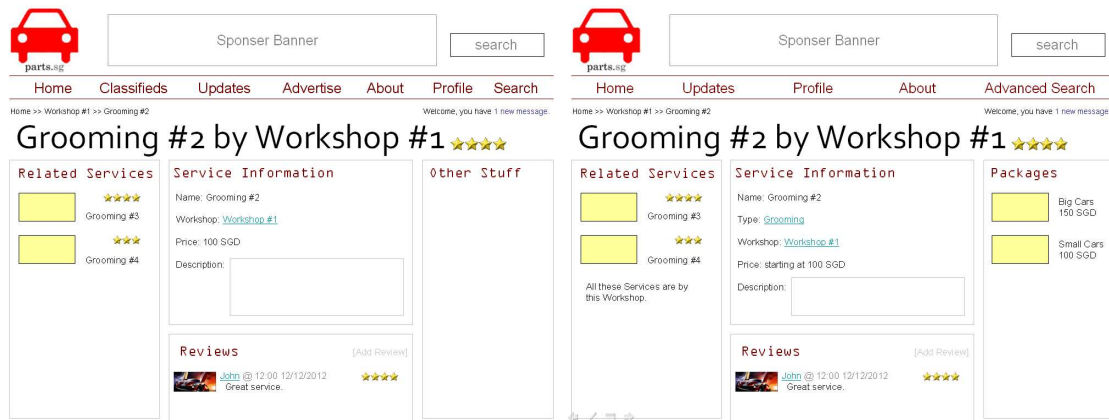


Figure 4.8: the first and final version of the Service by Workshop page.

The clients explained their view of the services in more detail and the prototype was adjusted accordingly (see figure 4.8 above). It had become clear that services can vary a lot between different workshops. As explained in section 4.1.1, services could have different packages. Packages could have different prices and as a result services should have an attribute 'price starting at' instead of merely 'price'. The main service page would be similar to the car brand page. One part of the page is a main description containing what can basically be expected from that service. The rest shows the lists of services offered by workshops and also a top 10 best rated services.

The prototype worked very well because the clients were a bit unsure if the company understood what their wishes were. They were delighted to see their website taking shape and studied every slide very carefully. PowerPoint makes it possible to recreate links so the clients could have an idea of

really clicking through the website. This made it easy to discuss the final functionality. Once this was completed, no new major functionality changes could be made. In previous projects of Collins Kumarasinghe Associates there were a lot of issues involving clients who wanted extra functionality when a website was almost finished. This prototype should help get a good overview right from the start so the company does not face any big surprises when the clients view the site after implementation.

The services page was just one of the many issues resolved using the prototype. The other changes can be viewed in Appendix E. These changes were also made to the prototype to mimic the website as closely as possible. The other developer was then able to see exactly what the clients wanted and by also using the UML diagram and use cases he was able to grasp the idea of the website fairly quickly and accurately.

4.1.3 Requirements

After the prototype had been approved by the clients it was time to finish the UML diagrams, create a functionality matrix and make a planning for the remaining period of the project. First, the UML class diagram was expanded significantly. The functionality matrix was created to get a better overview of the scope of the project. This made it easy to create a planning because the different functions could be grouped and were allotted a certain time period.

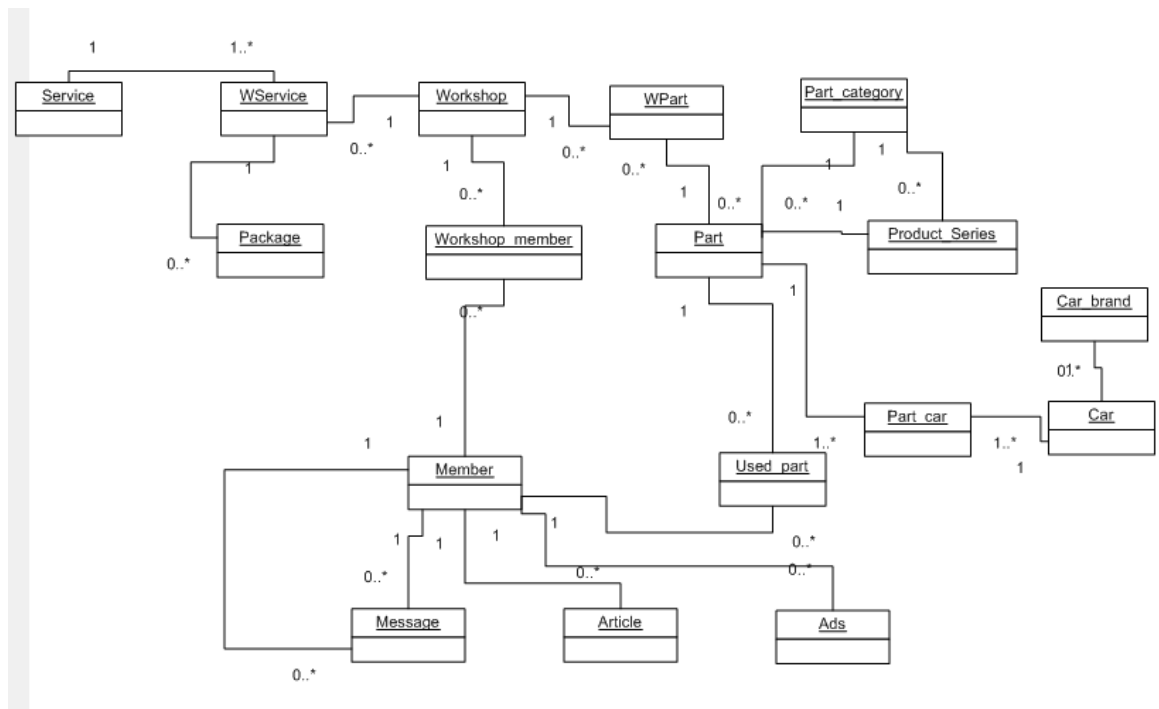


Figure 4.9: the final UML diagram for parts.sg.

The final UML diagram, shown above in figure 4.9, was made to illustrate the relations between the different content items on the website. The Watch List was not added because it would clutter the diagram. The Watch List keeps track of the changes of Workshops, WServices, Services, Parts and Used Parts. This diagram was the basis for setting up the functionality matrix.

The functionality matrix lists all the actions that are possible on the website. Each function has a small description and it lists the objects it interacts with. The type of data needed for this function is also added to the matrix. A few functions can be viewed below in table 4.1. The whole functionality matrix is available in Appendix G.

Name	Description	Object	Data
login	User log into the site	User	username password
createUser	A visitor registers on the site	User	username password email
searchServices	A user can search for services offered by workshops.	Search Service	category Keywords
searchParts	Finds global parts	Search Part CarBrand PartBrand	Category Carbrand Partbrand Keywords

Table 4.1: A small part of the functionality matrix.

Three weeks were left for the implementation of the parts.sg website. During these last weeks the workload was shared with the other developer. This meant a good overview of what should be implemented was needed. Using the functionality matrix, the different functions were grouped into sets as shown below in table 4.2. In Appendix F the Gantt chart made for the implementation of this project can be found. As can be seen in the Gantt chart, hardly anything went as planned. The other developer had gotten sick during the three weeks and the user experience project also required some time. Also, the other developer liked to code everything by himself while I had tried to find modules which already took care of a lot of functionality. This meant a lot of work was done twice.

After three weeks the Users, Moderators, Displays, Watch List and Updates, and Personal Message system had been implemented. Workshops had been created but most functions were still missing. Searching had not even been started on yet. The advertising system would be implemented and discussed when Micheal Collins returned from Europe, which was a day after I had left Singapore.

Functions	Time needed (days)
Users	3
<i>Register & Log In & Edit profile</i>	1
<i>Post Article/Part/Used Part/Comment</i>	2
Moderator	2
Search	2
Display Part/Used Part/Car Brand/Etc	1
Workshop	5
<i>Manage Workshop</i>	2
<i>Workshop Users</i>	2
<i>Workshop Parts/Services</i>	1
Watch List & Updates	2
Personal Message System	1
Advertising System	Unknown

Table 4.2: The overview of the time needed to implement most of the parts.sg functions.

4.2 Choosing a Framework

During the orientation phase of the Bachelorproject there was a whole section about creating a standard code library for Collins Kumarasinghe Associates. This was later changed to setting up the car parts website. This did not mean that all the work done for the standard code library was a waste of time! The company thought it would be interesting to see if this project, which was quite different from their normal projects, could be benefit more from a different framework than Joomla. After

some research, it was decided that Drupal would be the best option for the car parts website. In section 4.2.2 the main features and fundamental ideas of Drupal are explained.

4.2.1 Frameworks

The company uses Joomla, a content management framework, to create websites for their clients. For most of their clients this framework has worked well, but the parts.sg website is very different from their other websites. While Joomla does work well for many web applications, it has its limitations. The car parts project needs different user permissions and in Joomla this requirement is not very customizable. User permissions were very important during the process of choosing a new framework.

There are different types of frameworks available to create websites with using PHP. Each framework tries to make a certain part of the development or maintenance easier. Some of the big open source frameworks are the Zend Framework, CakePHP, Symfony, Joomla and Drupal. In this section these different frameworks will be discussed.

Zend Framework

The Zend Framework tries to make developing websites easier for programmers. Object-oriented programming and the Model View Controller (MVC) design pattern are used as the basis of the framework. The Zend Framework is more like a collection of components that take the dull repetitive programming work out of the hands of the developer. It is not a full programming environment in which a developer is forced to program in a certain way like the other frameworks require. It is flexible to use and can easily be incorporated with existing websites. PHP Data Objects (PDO) extension is used as a data-access abstraction. [10]

CakePHP

CakePHP is a framework modeled after Ruby on Rails, a popular framework for programming in Ruby. There are many tutorials and documentation available to start working with CakePHP. It also uses the MVC design pattern, just like most other frameworks discussed here. It is very strict with naming conventions and enforces people to use the MVC pattern. It uses PEAR for database abstraction. It uses script.aculo.us as extra JavaScript libraries. CakePHP is strict about how to use the framework but makes it easy to create websites very fast. [14]

Symfony

Symfony is also a framework based on the MVC design pattern and incorporates OOP. It stands apart from the other frameworks with the database abstraction. It uses YAML (YAML Ain't Markup Language), a language that has a very human readable format. Symfony supports Propel and Doctrine, two Object-Relational Mapping (ORM) frameworks for PHP. Symfony has the control over the MySQL data tables and the developer can change them by just editing the YAML file. However, installing Symfony and learning how to use it requires a lot of time. The learning curve is high but when the developer understands what Symfony wants, it can be very useful. It works well for large complicated web applications. [11]

Joomla

Joomla is a Content Management System (CMS) that is based on the MVC design pattern and also uses OOP. It makes it easy to manage content for non technical users and has many built-in modules such as calendars and polls. It has many add-ons but also some restrictions. It is difficult to allow more user permissions than already defined by Joomla. It's nice for small businesses or personal websites. There are many templates available to design a Joomla website with. Joomla also uses Mootools, a JavaScript framework, to help write JavaScript code. [12]

Drupal

Drupal is another CMS, but unlike most other frameworks it is not based on the MVC pattern. Using Drupal only differs from Joomla slightly. The administrator interface of Drupal uses Ajax, making it possible to slide and drop content within the website. It does not use as many buttons as Joomla does, but some people find it more user-friendly. The community of Drupal seems a bit smaller and fewer templates are available. It works best as a CMS for community websites but can be used for any website. User permissions are more versatile than the Joomla user permissions are. Extra groups can be created to satisfy the needs of a website. jQuery is used to apply JavaScript with Drupal. Drupal seems easier to customize than Joomla, but not as much as CakePHP, Symfony and the Zend Framework are. [13]

	Structured Code	CMS	User Permissions
CakePHP	++	-	++
Drupal	+	++	++
Joomla	-/+	++	-
Symfony	++	-	++
Zend Framework	-	--	++

Table 4.3: Overview of the different frameworks and the important aspects for the car parts website.

Choosing between five different frameworks is not easy. One by one each framework was removed as an option. First, the Zend Framework did not seem to fit the way the company works. They do not need to use it with existing sites, because these are made using Joomla. The Zend framework is very flexible, but we were looking for something structured (see table 4.3). Symfony was the second framework that was reviewed. Getting it to work took a very long time and it was difficult to understand. It has a steep learning curve and while it seems very powerful, you really needed to have a good understanding of how Symfony wants you to use it.

Now we have three frameworks left: Joomla, Drupal and CakePHP. At first, Drupal was not going to be used because Joomla has about the same functionality. Joomla, however, has severe limitations with user permissions. CakePHP made it possible to create any type of user you want with any type of permission! Creating these users was fairly easy and developing with CakePHP can be done very fast. The code was very structured, OOP used very nicely and the MVC architecture was implemented very well. The problem is that a content management system would have to be created and a lot of functions made available in the CMS frameworks would have to be implemented too. Drupal was then evaluated because it is a CMS framework which is good for creating community websites.

For www.parts.sg it was important to choose a framework that had enough functionality built in or options to use add-ons. While CakePHP can be used to create websites quickly, the time spent creating a CMS would cancel out the time gained. The other requirement was that the framework should allow very flexible user permissions. Joomla is a lot less flexible than Drupal when it comes to user permissions. This is why Drupal was chosen to develop the car parts website with. While it did require the developer to learn a new framework, this would be useful for the company in the long run.

4.2.2 Drupal

After Drupal was picked as the framework to use for the parts.sg website, some further research had to be done. Each framework is based on a set of concepts and when those are understood, it is easy to work with. The concepts of Drupal are explained in this section.

Drupal treats almost all content the same based on one concept: a Node. Nodes are data which can be displayed on the website. This can range from simple Pages (a webpage with a title and some text) to blog posts to news items. By using one concept for almost all the data, things such as comments can be added to any node and it is not limited to only blog posts. All nodes are saved together in the database, guaranteeing that all nodes are handled the same way. Every node has a Content Type (such as a Page), a Node ID, a Title, a creation date and an author. Some nodes also have a Body or other attributes. This is all saved a database table. [1][2]

Modules are bits of code that add extra functionality to a Drupal website. Many modules are available on www.drupal.org but they can also be coded by the developer of the website. Core modules are the modules that are included when downloading Drupal. It is possible to disable or enable modules. The Blog module is a core module but does not have to be enabled when using Drupal. Some core modules are always required to keep the website running and therefore cannot be disabled. An example of such a core module is the Node module. [2]

The next important concept is Blocks. Blocks are used to display whatever a module wants to have displayed. These blocks can be placed in different regions of the website. It is possible to show certain blocks to a select group of users. For example, only logged in users can see special news items. Some of the regions a block can be placed in are the left sidebar, right sidebar, header, footer and more. [1][2]

Block	Region	Operations
Left sidebar		
Navigation	Left sidebar	configure
User login	Left sidebar	configure
watchlist: Block	Left sidebar	configure
Modr8 moderator's block	Left sidebar	configure
Right sidebar		
No blocks in this region		
Content		
No blocks in this region		
Header		
No blocks in this region		
Footer		
Powered by Drupal	Footer	configure
Disabled		
Author information	<none>	configure
Primary links	<none>	configure

Figure 4.10: the Blocks administration page.

Many different types of users can be created using Roles. When users are assigned a role, they can have different permissions on the website. For instance, a role called Moderator can be created. Moderators would be able to change and view all content. Permissions are used to define what actions a user with a certain role can or cannot do. A user can have multiple roles. The two default roles are: anonymous user and authenticated user. All logged in users have the authenticated user role. [1][2]

On the User Permissions page the admin can determine which things different user types have access to. Permissions are assigned to various roles, and in turn, users are associated with those

various roles in order to grant them the associated permissions. These permissions are different for each module and for example can viewing some type of content or editing it. Users might be able to view content but can only edit what they created on the website. [1][2]

Finally, the concept Theme describes how the website can be displayed. A theme controls how the website looks. XHTML and CSS are used to create a theme and sometimes a bit of PHP may be used to show content in the correct places. Different themes can be created to show different colors of the same website. Maybe some user roles see a different website layout. Theme can be assigned to roles and it is also possible to let users choose whatever theme they would like. [1][2]

Figure 4.11 shows the different concepts and how each concept requires other concepts. A module cannot work without content which is provided by the nodes.

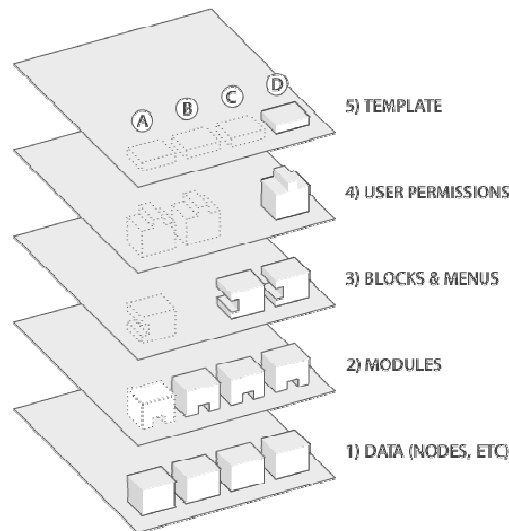


Figure 4.11: the five Drupal concepts.

4.3 Implementation

Before starting the serious programming, a lot of existing modules were found. During the implementation phase it became clear what modules did and did not work well with the rest of the system. Some modules worked but were missing a tiny bit of functionality. In the case of programming the Moderator functions, an existing module was used but some functionality was added to it.

While I tried to find as many high quality and useful modules, the other developer only liked using what he programmed. Only what I have created will be discussed in this section. First, the different modules used during development will be discussed. Second, the special case of the Moderator function will be highlighted. This chapter ends with some final thoughts about the implementation phase.

4.3.1 Modules

Drupal uses the concept of Modules to group functions. There is a User module that handles the registration and logging in/out of users. Another module, Profile, adds the functionality that users can have their own profile on the website. For each module many different options are available such as only allowing users to register after receiving an email invitation or letting anyone join who wants to.

This first module that was downloaded and explored was Organic Groups. This module was proposed as a solution for the Workshop functionality. Organic Groups lets users create and manage groups. These groups each have a homepage which can be edited by the users in the group. The workshop functionality ended up being much more complicated than this module could handle and therefore Organic Groups were not used. A module called Workshop was created by the other developer. [7]

To create the content types such as parts, used parts, services and more, the CCK (Content Construction Kit) module could be used. CCK makes it possible to add custom fields to nodes using the administrative interface and requires no programming. As can be seen in figure 4.12, it is very easy to add new fields. The type of fields can be references to other nodes, like a part fits into a category, or just plain text. Fields can also contain files or images. It is possible to choose the number of values that can be assigned for each field. The field color could have 10 different values if we think a part can have a maximum of 10 colors. The default is that there is only one value. [3]

Label	Name	Type	Operations
Name	Node module form.		
Category	field_cat	Node reference	Configure Remove
Part Number	field_number	Text	Configure Remove
Product Series	field_productseries	Node reference	Configure Remove
Manufacturer	field_manufacturer	Node reference	Configure Remove
Car	field_car	Node reference	Configure Remove
Car Series	field_carseries	Node reference	Configure Remove
Car Brand	field_carbrand	Node reference	Configure Remove
Colors	field_colors	Text	Configure Remove
Description	field_description	Text	Configure Remove
Menu settings	Menu module form.		

Add

☐ **New field**

Label: Name: Type: Widget:

☐ **Existing field**

Figure 4.12: Example of the fields that can be added using the CCK module.

After creating the content types, it was time to focus on the moderator functionality. The plan was to use the modules Flag, Rules and Views. The module Flag is a flagging system that can provide flags for nodes, comments or users. It could be used for keeping track of bookmarks, but for this project it can be used to keep track of content that has not been moderated yet. [5]

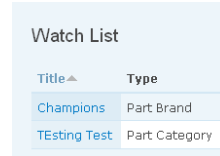
The module Rules made it possible to create rules in the system such as: when new content is created, flag content for moderation. Another rule was: when content is modified, flag it for moderation.

The Views module was used to create lists of the content that was flagged for moderation. This could be done without any programming. The Views module is a query builder and can create lists of whatever content is needed. Different outputs can be chosen for the lists. The list can be output as a page, which is what was used for the moderator functionality. It can also be output as a block and for example be placed in a sidebar. This was used for the Watch List, which will be discussed later. [4]

This setup was not used because it was not possible to add a field in the list to block a user right away. This is something the clients wanted, so another solution had to be found. Two other modules, Moderator and Modr8, were examined to see if these did contain the needed functionality.

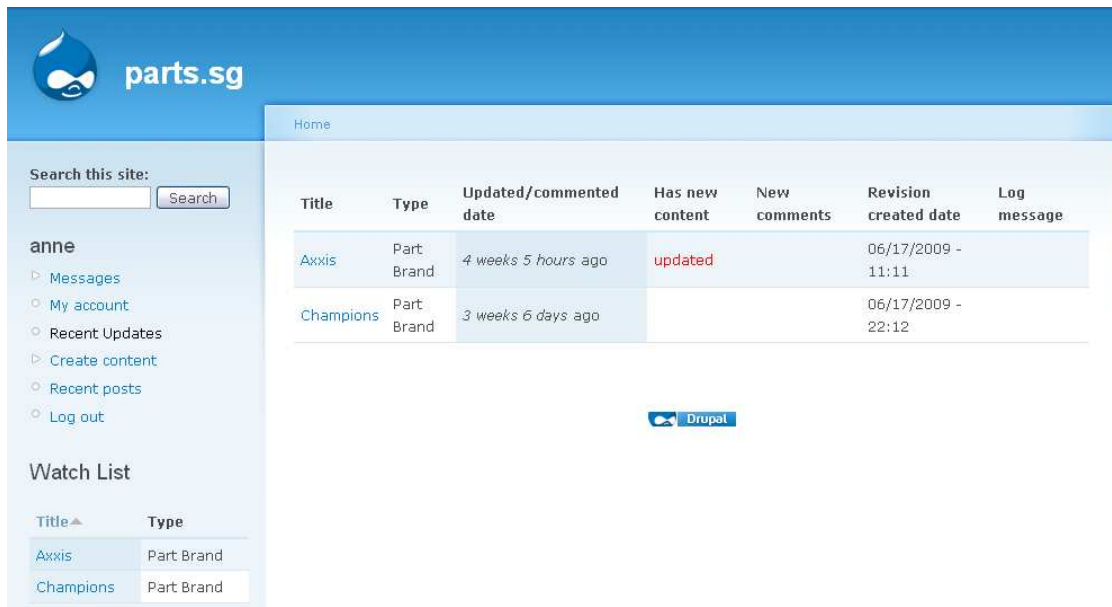
The Moderator module looked very promising, but it could not approve or deny content. It only showed content that had not been moderated yet and still needed to be published. It therefore could not be used for content that already existed and was edited. The Moder8 module did not have these options either, but has a simpler look to it. This module seemed easier to extend than the Moderator module. This will be discussed in more detail in 4.3.2. [8][9]

The Flag and Views modules were also used to create the Watch List functionality. This was quite easy. Users could flag content they wanted on their Watch List and the list was displayed using Views. It was output to a block in the left sidebar, see figure 4.13. Changes made to items in the Watch List could also be displayed using Views (see figure 4.15). This time the output was a page showing the content items, content types and when it was last created/edited as can be seen in figure 4.14. This was implemented in less than a day which is a lot less than the two days planned for this function.



Watch List	
Title ▲	Type
Champions	Part Brand
Testing Test	Part Category

Figure 4.13: the block in the left sidebar containing the Watch List.



Title	Type	Updated/commented date	Has new content	New comments	Revision created date	Log message
Axxis	Part Brand	4 weeks 5 hours ago	updated		06/17/2009 - 11:11	
Champions	Part Brand	3 weeks 6 days ago			06/17/2009 - 22:12	

Figure 4.14: the Recent Updates page showing the items in the Watch List.

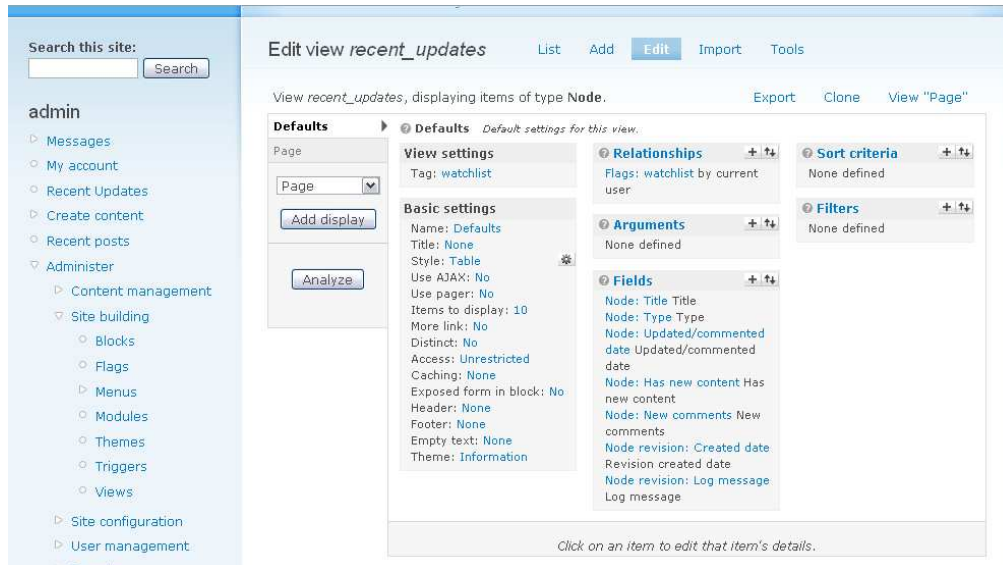


Figure 4.15: the Views page for the Recent Updates.

The clients wanted a system that would allow users to rate content on the website. The module Fivestar was perfect for this functionality. It allows users to add comments with a rating between 1 and 5. The rating is showed on content page as shown below in figure 4.16. When comments are added, the rating the user gave is also shown. Some content items do not have comments enabled but they can be rated anyway. The module allows content to be rating with or without comments and this can be configured separately for each content type. [6]

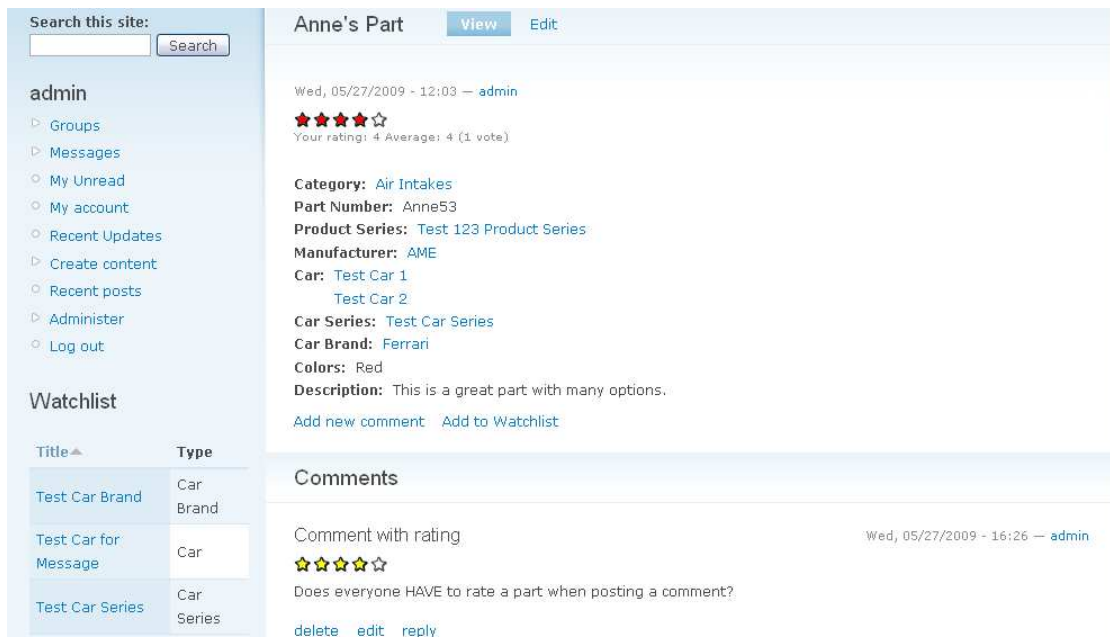


Figure 4.16: A Part page containing the Fivestar Rating module.

The last module that was used was Privatmsg. The clients wanted a messaging system but did not really specify anything about it. This module took care of standard messaging functionality such as sending and receiving messages from and to other users. Emails can also be sent to users who have

received a message on the website. Other possible functionality such as blocking users was also enabled. The clients used it on the temporary test website and approved it. [16]

4.3.2 Moderator

The moderator functionality was not available in any existing modules on the official Drupal website. Still, some modules came close to achieving the desired functionality. These modules showed new content items, which had to be moderated, in a list. Then the action ‘moderate’ could be chosen and the item would no longer appear. The clients also wanted to be able to block users right away on the moderation page and it was also important that when content is edited, it appears in the moderation list again. Eventually, only being able to block users on the moderator page was implemented. The content types such as parts had not been finished by the other programmer in time to work with.

The module Modr8 was used to take care of a part of the necessary functionality for the moderator. As said in section 4.3.1, this module was easiest to add functionality to. Modr8 shows a page with the items that still need to be moderated. Actions that could be chosen are: approve, delete or no action. The content of the item is also displayed to view the changes quickly.

Being able to block a user was added to the Modr8 module. It shows up on the overview page of all the items waiting to be moderated as can be seen in figure 4.17. It was also important that this option showed up on the settings page of the Modr8 module. This module also allows the administrator to have emails sent to the users if their content was approved or deleted. This function was also added for when a user was blocked. This is all displayed on the settings page as seen below in figure 4.18. This page also allows the administrator to choose the default settings for approve/delete/no action and block/no action.

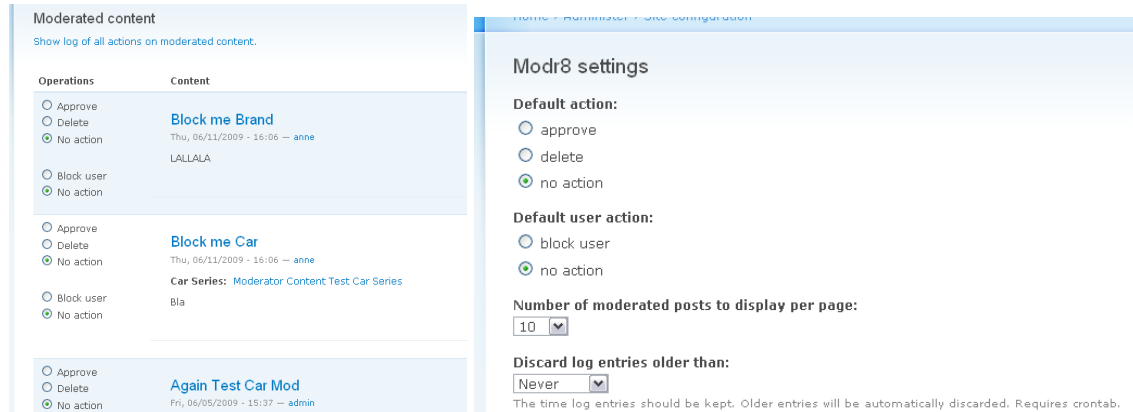


Figure 4.17: the Moderator page and figure 4.18: the Moderator Settings page.

To add these two options, the admin settings file had to be edited. To the function `modr8_settings_form` the code in figure 4.20 was added to enable the settings for blocking users. The code to show the email options, as pictured in figure 4.19, was the same as other code already in the module and only needed some words to be changed. Actually sending the email was done in the function `modr8_form_submit` and is displayed in figure 4.21. To display the functionality on the moderator page, the code in figure 4.22 was added to the function `modr8_form`. All these things were not very difficult and are pretty straightforward. It is worth noting that only moderators who have permission to block users can see the block user option.

Title	Type
Test Car Brand	Car Brand
Test Car for Message	Car
Test Car Series	Car Series

Moderation queue
3 posts in moderation

Recent additions:

- Block me Brand
- Block me Car
- Again Test Car Mod

No action e-mail note:

Your %type entry entitled "%title" has been reviewed by our content moderator, but not yet approved.

%note

To respond to the moderator, you can visit %response_url

Replacement strings are: %title, %teaser, %body, %short_date, %medium_date, %long_date, %type, %node_url, %nid, %author_name, %author_mail, %author_url, %site, %note, %response_url

☐ Send user blocked messages

Blocked user e-mail subject:

[%site] note to author about %title

Blocked user e-mail note:

Your %type entry entitled "%title" has been reviewed by our content moderator and has gotten your account blocked from %site.

%note

Regards,

Replacement strings are: %title, %teaser, %body, %short_date, %medium_date, %long_date, %type, %node_url, %nid, %author_name, %author_mail, %author_url, %site, %note, %response_url

Figure 4.19: editing the email text on the Moderator Settings page.

```

4 function modr8_settings_form($form_state) {
5
6     $form['modr8_default_option'] = array(
7         '#type' => 'radios',
8         '#title' => t('Default action'),
9         '#options' => array(
10             'approve' => t('approve'),
11             'delete' => t('delete'),
12             'nada' => t('no action'),
13         ),
14         '#default_value' => variable_get('modr8_default_option', 'nada'),
15     );
16
17     $form['modr8_default_user'] = array(
18         '#type' => 'radios',
19         '#title' => t('Default user action'),
20         '#options' => array(
21             'block' => t('block user'),
22             'nada' => t('no action'),
23         ),
24         '#default_value' => variable_get('modr8_default_user', 'nada'),
25     );

```

Figure 4.20: the modr8_setting_form code.

```

415 }
416 // Do the same for blocking the user
417 switch ($values['ops_user']) {
418     case 'block':
419         if (variable_get('modr8_send_approve', FALSE)) {
420             $message = modr8_usermail('block', $nid, $values);
421         }
422         if (user_access('administer users')) {
423             $block = 'status = 0';
424         }
425         db_query('UPDATE (users) SET '. $block .' WHERE uid = %d', $values['author_uid']);
426         drupal_set_message(t('The user %user has been blocked.', array('%user' => $values['author_uid'])));
427         cache_clear_all();
428         //modr8_log_action('blocked', $author_uid, $values, $message);
429         break;
430     case 'nada':
431         if (variable_get('modr8_send_noact', FALSE) && !empty($values['note'])) {
432             $message = modr8_usermail('nada', $author_uid, $values);
433             //modr8_log_action('nada', $nid, $values, $message);
434         }
435         break;
436 }
437 }

```

Figure 4.21: making it possible to send emails to users once they are blocked.

```
278 //if the moderator has permission to block users, they will see this
279 //option at the moderation page too.
280 if (user_access("administer users")) {
281     $op_user_options['block'] = t('Block user');
282 }
283 $op_user_options['nada'] = t('No action');
284
285 $form[$node->nid]['ops_user'] = array(
286     '#type' => 'radios',
287     '#options' => $op_user_options,
288     '#default_value' => variable_get('modr8_default_option', 'nada'),
289 );
290
```

Figure 4.22: show the blocking option on the Moderator page and check if the user has the correct permissions.

Some functionality still needs to be implemented but was omitted due to lack of time and insufficient resources. The Modr8 module logs all moderator actions so it would be good if it also logs when a user is blocked. Another bigger function that still needed to be added was that when content is updated, it should also be placed in the moderator list. Since the content types such as parts and services had not been completed yet by the other developer, it was too difficult to create the updates functionality. This was because it was not clear if he wanted to use revisions to keep track of changes, or create his own function for keeping track of the changes of a node.

4.3.3 Conclusion

The implementation phase was not as successful as hoped. While the idea was to use existing modules to take care of as much functionality as possible, this did not end up being the case. The other developer preferred creating his own modules from scratch. A few existing modules were used because these functions were mostly extra features for the website such as the private messaging system. Core functions such as the content types were implemented by the other developer with his own modules.

The modules that were used are Privatemsg for private messaging on the website, Fivestar for rating content, and Views and Flags to create the Watch List functionality. Using CCK to create the different content types was not used and the developer created his own modules for this. It is unclear if he decided to use the Modr8 module as a basis for the moderator functionality but he most likely created his own module for this too. The workshop functionality was implemented by a custom module because this is a very specific and complicated feature.

The goal of using Drupal as the framework for the car parts website was to make programming easier for the developer. While he did create most modules himself, he did say that creating modules went quickly. He would have needed less time to implement the website using Joomla because he had been using that for the past two years. He still liked working with a new framework and it will be interesting to see if the company will use Drupal more often.

Drupal is great for creating websites quickly without having to program at all. For less complicated websites than the car parts website, it is not hard to imagine that someone can create a quite large website without having to look at any code. Drupal has a lot of available modules which are very flexible and can almost fit any need. Drupal is recommended for anyone who is looking for a customizable framework. To create a simple blog website Drupal can be used but it would be easier to use a specialized blog website such as www.blogspot.com for this. However, Drupal is the right choice when someone wants a site where their users can each have their own personal blog on the website.

5 Conclusions and Recommendations

The idea of this project was to enhance the user experience for the Test Zone and making the programmer's life easier by using a framework that fit the current project well and by researching the requirements of the project in more depth than the company usually does. This chapter will explain if this goal was reached and which parts of the project went well or not as well as hoped. First the Test Zone will be discussed, then the car parts website and finally the overall project.

Things that could have gone better during the project and things that still need some work will be discussed in 5.2. These recommendations vary from advice for Collins Kumarasinghe Associates to some final thoughts about this bachelorproject.

5.1 Conclusions

The conclusions here focus on the user experience research, the car parts website and the overall project. Especially using questionnaires, the think-aloud protocol, a prototype and Drupal will be discussed.

To assess the user friendliness of the Test Zone questionnaires, interviews and the think-aloud protocol were used. The online and paper-based questionnaires were not a success to say the least. It is obvious that people will not fill in a questionnaire if there is no reward and if it takes too much time. Users had to go through the web application first and after that they had to answer questions. This was too difficult and people were not willing to spend time on it.

The paper-based questionnaires failed because hardly anyone that showed up at the training sessions at Collins Kumarasinghe Associates had used the Test Zone before. This was also the reason why it was hard to find people for the interviews.

Thankfully the think-aloud sessions were a big success. Every week around 5 people came in who could take part in the research. In a place where many potential users are available for testing, the think-aloud protocol should be the preferred method for gathering data.

The three research phases showed how the issues with the Test Zone slowly decreased. After the first and second research phases most large usability problems had been solved and no new large problems arose. Some problems still linger in the web application and these are discussed in paragraph 5.2. These problems are mostly due to lack of content on the website and are known issues at the company.

The second project of the bachelorproject, the car parts website, was a lot more work than assessing the user experience of the Test Zone. The project specifications changed during the first half of the project which meant some work done had become useless. Thankfully some work could be carried over. This was mainly choosing a correct framework. To choose a suitable framework a lot of time had to be spent researching and testing the different possibilities.

Gathering requirements for the car parts website was very interesting because it was really important to understand the clients' wishes correctly and at the same time the other programmer had to understand everything too. While understanding the clients' wishes went quite well, it was more difficult to explain everything to the other programmer. Using use cases, UML class diagrams and a prototype, the requirements were explained. Using a prototype is an extremely effective way to determine the requirements with the clients and to show the programmer how the system should

work. The prototype was used to create the functionality matrix. This overview was very useful when the Gantt chart was created.

The conclusion of the research phase was that Drupal was the best framework for this website, with Joomla as a close runner up. CakePHP would have been the best choice if the clients did not require a CMS but this was not the case. The Zend framework and Symfony were both not a good match with the current way the company develops websites.

Drupal is a very flexible CMS and there are many extra modules that can be used to extend the functionality of a website. It is not mandatory that other modules are used and it is easy enough to create your own, but it is easier to use modules that have been coded and tested already. It is a shame that the other programmer did not like using other people's modules. Drupal was a perfect fit for this project. The user permissions were very easy to modify which is exactly what was needed.

Both projects were very interesting and it was great that experience could be gained in two different fields: interviewing users and interviews clients. There is quite a difference between the two. The users were not really sure what they wanted while the clients knew what they wanted but not exactly how. It was a challenge to understand these two different types of stakeholders.

Completing a bachelorproject on your own makes you realize how slow work is done when not working in a team. I now understand why it is required that a minimum of two people should work on a project together. Just creating the documentation takes twice the time it usually would which is a shame when more interesting things, such as research, can be done during that time.

5.2 Recommendations

The recommendations made focus on the two different projects but also on the overall bachelorproject. Every project has some points that it could have improved or things left unfinished. These are discussed here.

During the user experience research, it became clear that people will not spend time on questionnaires when there is no reward given in return. While one person filled in part of the questionnaire, ten others stopped right after reading what they would have to do. If a questionnaire is necessary for research, it is wise to offer people a small reward.

If more people are available who do not have experience with the system, it is much easier to use the think-aloud protocol. The think-aloud protocol is very effective to gather data with. It does, however, take up quite a lot of time. Questionnaires take less time, but when online questionnaires are sent to users it may take some time before they are filled in.

The usability of the Test Zone was increased by only making a few small changes. However, it should be noted that a few bigger issues still linger in the application. A few users missed testing categories such as Reading Comprehension which lead to confusion about the pie charts on the My Dashboard page. The trainers at Collins Kumarasinghe Associates need to create tests for each category. This is already a goal at the company.

Another issue is that some users did not like that the tests did not resemble the GMAT tests perfectly. The full length tests would be based on a real GMAT tests, but users do not know that these tests are in the works. It would be a great improvement if these are added to the system to reduce the confusion between the current tests that are meant for practice and the full length tests that mimic the real GMAT.

The progress bar would be a good enhancement for the testing system. It is recommended that this is added to the application but the timer works well too. The progress bar can increase the stress level of the users which is a good thing because they will be stressed when taking the GMAT.

Communicating with the developer and the clients using UML diagrams, use cases and prototypes is very useful. Especially using prototypes proved to be very effective when talking to people who do not have any idea of computer applications. While the clients were quite aware of what was possible on the web, they were unaware of the time it would take to create the website. The prototype helped define the different functions and showed that it would require more time than they expected. The prototype does show them that their wishes are understood. It is recommended that a small prototype is always made to discuss an application with clients.

During the second project some things did not go as smoothly as hoped either. The focus had been to find modules that would reduce the time spent programming. The other programmer did not like using other people's work because he preferred his own. While this led to some frustrations during the project, it was nice to know that this was a known issue with the company. It is difficult to join a company and tell another programmer what to do while he has been working there for a longer time. This is something that was underestimated. It would have been better if more time was spent communicating with the other programmer even if he was not really open to this.

Drupal was a good choice for this project. It is a very flexible framework and is recommended for anyone who wants to create a website quickly. While someone should not choose Drupal if they only want a blog website, Drupal is a good choice if someone wants multiple users on a website who each have their own blog. Drupal is great for multi-user websites with its flexible user permissions and different web community options such as blogs, forums and user profiles.

During the whole bachelorproject it was pretty difficult to spend time on both projects. It is better to focus on one specific project but this situation can also be seen as an extra challenge. It is recommended that only one project is chosen. Also, people should work in groups because the documentation required for the bachelorproject takes too much time when only one person is working on it.

References

- [1] *The Drupal overview*. <http://drupal.org/getting-started/before/overview> accessed on the 7th of July, 2009.
- [2] *General Concepts*. <http://drupal.org/node/19828> accessed on the 7th of July, 2009.
- [3] *Content Construction Kit (CCK)*. <http://drupal.org/project/cck> accessed on the 8th of July, 2009.
- [4] *Views*. <http://drupal.org/project/views> accessed on the 8th of July, 2009.
- [5] *Flag*. <http://drupal.org/project/flag> accessed on the 8th of July, 2009.
- [6] *Fivestar*. <http://drupal.org/project/fivestar> accessed on the 8th of July, 2009.
- [7] *Organic Groups*. <http://drupal.org/project/og> accessed on the 8th of July, 2009.
- [8] *Modr8*. <http://drupal.org/project/modr8> accessed on the 8th of July, 2009.
- [9] *Moderation*. <http://drupal.org/project/moderation> accessed on the 8th of July, 2009.
- [10] *About Zend Framework*. <http://framework.zend.com/about/overview> accessed on the 29th of April, 2009.
- [11] *What's symfony?* <http://www.symfony-project.org/about> accessed on the 27th of April, 2009.
- [12] *What is Joomla?* <http://www.joomla.org/about-joomla.html> accessed on the 15th of April, 2009.
- [13] *About Drupal*. <http://drupal.org/about> accessed on the 30th of April, 2009.
- [14] *CakePHP: Basic Features*. <http://cakephp.org/pages/features> accessed on the 28th of April, 2009.
- [15] *Comparison of web application frameworks*.
http://en.wikipedia.org/wiki/Comparison_of_web_application_frameworks accessed on the 29th of April, 2009.
- [16] *Privatemsg*. <http://drupal.org/project/privatemsg> accessed on the 8th of July, 2009.
- [17] Preece, Jenny, Rogers, Yvonne and Sharp, Helen. *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 2007.

Appendices

Appendix A: Questionnaire

Welcome! I am trying to find out what aspects of a web application, which helps you practice for the GMAT, are in need of improvement. The important part is that the application is userfriendly and that people can easily understand how to use it. That is why I need your help! I would like you to tell me what was unclear, annoying, or anything else negative, but also tell me what you liked about the application and what should definantly not be changed!

Please fill in your age before we get started:

- Under 20
- 20-29
- 30-39
- 40-50
- Over 50

You may explore the website, but no questions will be asked about it. The Test Zone is the focus of the study. Please spend around 10 minutes exploring the application. You should try view every page and use the functionality available.

You are trying to study for the GMAT and would like to use this application to practice some tests. Try to go about the website in that manner. You have already completed some tests in the system (this makes it possible to test extra features and you will not have to complete a full practice test).

Please log in at http://www.gmat-zone.com/component/option,com_testzone/Itemid,45/ with the following information:

Username:

Password:

(If you are unable to log in, please email anne.vanee@colkum.com).

Please rate (i.e. check the box to show) agreement or disagreement with the following statements:

	Strongly Disagree	Disagree	Okay	Agree	Strongly Agree	NA
The side menu (Test Zone Menu) was easily located as the main navigation between pages.						
Each link in the menu gave me the kind of page I expected to see.						
The practice tests represent real GMAT questions.						
Reviewing the questions when seeing the solutions was intuitive.						
Clicking on the red X to show that you reviewed a question was intuitive.						
I expected to be able to review questions at the 'Review Section'.						
Creating my own tests containing previously answered questions is useful.						
The 'My Dashboard' is a good first page.						
If the pie charts worked, they would be helpful.						
I think viewing your past results on the tests is good.						

The list of previous statements to make it more readable in the questionnaire document. This was not used in the online survey.

1. The side menu ("Test Zone Menu") was easily located as the main navigation between pages.
 2. Most links in the menu gave me the type of page I expected to see.
 3. The practice tests are representative real GMAT questions.
 4. Reviewing the questions when seeing the solutions was intuitive.
 5. Clicking on the red X to show that you reviewed a question was intuitive.
 6. I expected to be able to review questions at the 'Review Section'.
 7. Creating my own tests containing previously answered questions is useful.
 8. The 'My Dashboard' is a good first page.
 9. If the pie charts worked, they would be helpful.
 10. I think viewing your past results on the tests is good.
-

This is the last page. Please give complete answers to the open questions.

I downloaded the Mathplayer.

Yes/no

Do you understand why you would have to download it? If yes, please explain; if no, just fill in 'no'.

What was the best feature of the application? Explain why.

Please name what parts of the application confused you.

If you have any extra comments or thoughts about the application, place them here.

Thank you for participating in this survey!

Appendix B: The Result of the Online Questionnaire

1. Please rate (i.e. check the box to show) agreement or disagreement with the following statements.							
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	NA	Response Count
The side menu ('Test Zone Menu') was easily located as the main navigation between pages.	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	100.0% (1)	0.0% (0)	1
Most links in the menu gave me the type of page I expected to see.	0.0% (0)	0.0% (0)	0.0% (0)	100.0% (1)	0.0% (0)	0.0% (0)	1
The practice tests are representative real GMAT questions.	0.0% (0)	0.0% (0)	0.0% (0)	100.0% (1)	0.0% (0)	0.0% (0)	1
Reviewing the questions when seeing the solutions was intuitive.	0.0% (0)	0.0% (0)	0.0% (0)	100.0% (1)	0.0% (0)	0.0% (0)	1
Clicking on the red X to show that you reviewed a question was intuitive.	0.0% (0)	0.0% (0)	0.0% (0)	100.0% (1)	0.0% (0)	0.0% (0)	1
I expected to be able to review questions at the 'Review Section'.	0.0% (0)	0.0% (0)	100.0% (1)	0.0% (0)	0.0% (0)	0.0% (0)	1
Creating my own tests containing previously answered questions is useful.	0.0% (0)	0.0% (0)	100.0% (1)	0.0% (0)	0.0% (0)	0.0% (0)	1
The 'My Dashboard' is a good first page.	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	100.0% (1)	0.0% (0)	1
If the pie charts worked, they would be helpful.	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	100.0% (1)	0.0% (0)	1
I think viewing your past results on the tests is good.	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	100.0% (1)	0.0% (0)	1
	<i>answered question</i>						1
	<i>skipped question</i>						10

Appendix C: The Result of the Paper-based Questionnaire

	Strongly Disagree	Disagree	Okay	Agree	Strongly Agree	NA
The side menu (Test Zone Menu) was easily located as the main navigation between pages.				1		
Each link in the menu gave me the kind of page I expected to see.				1		
The practice tests represent real GMAT questions.			1			
Reviewing the questions when seeing the solutions was intuitive.				1		
Clicking on the red X to show that you reviewed a question was intuitive.				1		
I expected to be able to review questions at the 'Review Section'.				1		
Creating my own tests containing previously answered questions is useful.			1			
The 'My Dashboard' is a good first page.				1		
If the pie charts worked, they would be helpful.				1		
I think viewing your past results on the tests is good.				1		

Appendix D: Parts.sg specifications



WEB APPLICATION SPECIFICATIONS VERSION 1.0

Aimless blabber on what the site should, and will be.

FOREWORD

1. We believe the script has to be as smart as possible, reducing the need for a human in the loop. Ideally, the script should be smart enough to find out if there are repeated products etc etc.
2. We believe in spending on a good script, hence reducing or eliminating the need for manpower to keep the website or database organized.
3. Loyal or regular users to our site should be our biggest “employees” concept.

PROGRAMMING AND HOSTING BASICS

1. Built based on PHP and MYSQL.
2. Website can be built, uploaded and tested on a server that already has Parts.sg being pointed to. However, do note the server is already running a few other websites, so care must be taken not to delete the database and files of the other websites.

MAIN FEATURES

1. Users search for parts based on Car Brand/Make, Part Types or Part Brands.
2. Integrated user profile database, workshop database, parts database, and mechanic database.
3. Integrated user profile editing, workshop details and products editing.
4. Integrated classified ads section.
5. Integrated reviews function. Reviews parts (reviews specific to car types), workshops, mechanics and other users (in the case of classified ads transaction).
6. Integrated rating system of everything reviewable.
7. Integrated advertising system.
8. Integrated article system for editorials and advertorials.
9. RSS inputs from local forums? (What’s the buzz?)

USER UPDATE AS FAR AS POSSIBLE

1. The entire database should as far as possible be registered user updatable. The entire process can be moderated by admin. Think Wikipedia concept (this is a bonus feature if possible).
 - a. A user updates their own workshop details
 - b. A user updates their own product details (do note the difference between a distributor, a sole agent, in which case both should be allowed to edit the details of the product. Of course permission should be given by the first claimed owner of the product)
 - c. Normal registered user can make suggestions to changes, which can be reviewed by either the admin or the owner.
 - d. Normal users should be able to make changes to general details like Car brand summary, car make summary, part brand summary etc etc, those things that have no owner linked to them. Of course, the admin can edit it.

2. Should there be a product not listed yet, a user can be allowed to add a product. The script should use some logic to see if there is any repeated product. If there is, the details entered should be merged with the original product. It's usually for the 1st entry by a sincere user of a new product is not complete. So subsequent users should be able to add to those details. Falling back to the wiki user added details. Of course, these can be moderated until we realize that there is no more need, ie when the community proved itself to be mature enough to self-run.
 - a. Once this is done, a workshop user login to the site will be shown "The following products have been added since your last visit. Do you sell them?" and the list of new products are shown, allowing them to quickly add these onto their sell list if they do carry them at their workshop.

SEARCH RESULTS

1. Upon selection of 1st level search criteria, any relevant results are immediately displayed. Ie Users don't need to go thru several levels of selection before results are displayed. This concept aids "window shopping" browsing.
2. Because certain services are not product defined, ie custom bodykit service, car grooming, workshops should be allowed to define the services they provide, and have custom page where they can update their pricing packaging etc. Users can then have the option of searching by services. They will be brought to a page "Car Grooming" where they will see all the service providers that do it, with price listed as starting from \$50, and clicking on it will show users details of the service package.

USER PROFILE

1. Single log-in detail to edit user profile or workshop details and products (if user has staked claim on the product)
2. A user can stake claim on a workshop or product (thru a user or a workshop), which the admin can then verify.
3. Once a user has a verified ownership of a product or a workshop, they become the "moderator" of it, and are able to give permission to anyone to edit it.
4. A workshop moderator is able to easily add products that they sell. We're open to how to make this as easy as possible. See product detail point 7, intended to make it easier to add products for sale in a workshop.
5. User don't need to key in detailed info like HP and address. However if he happens to click on the "sell my used part" button, these details may be asked. And if he enters the details in, his profile is automatically updated.

PRODUCT DETAILS

1. Shows a basic summary of the product, and can show the details if the user choose to.
2. Users can easily add their used product for sale here, which will be inserted into the classified ads section.
3. Any product that is in the used section will be shown here, a few randomly selected and shown, with a "more" button if people are interested to see more.
4. Replaceable parts can be shown here too.
5. Related parts can be shown here too. (ie brake fluids shown in a brake kit section)
6. Reviews of the product, car specific can be shown here.
7. If a user has a workshop ownership, and they do not have the product listed for sale in the workshop yet, they can click on a button like "I have this in my showroom too!" kinda concept.

8. Workshops that sell this product will be shown on this page. Contact details are immediately shown for people to call if required. Ranking as follows
 - a. Featured
 - b. Priced product (to encourage people to price their products)
 - c. Rating
9. A user who has staked claim on this product is able to choose whether to show users if they are the distributor. Otherwise nothing is shown.
10. Some pricing is not realistic to be a fixed price (ie car grooming with several packages for sale). So something like \$50 and up with a link for users to click for detailed pricing is displayed.

USER PROFILE PANEL

1. A normal user will only see a Parts.sg user profile tab.
2. He can “claim” and unclaimed workshop. The admin will then verify that and allow or disallow it.
3. The user who has a workshop claimed will see a workshop details tab to edit.
4. After that, he is now the moderator for that workshop’s details page. He will then be allowed to define other moderators who will then be allowed to edit the page. Those users will also see a new tab in the user profile panel.
5. There should be a place for users to see what’s been updated or added or removed. A change in “part no” in a product, or a change in workshop address in the workshop page, or a workshop listing a product for sale in a workshop will be shown. Basically any logical change in the database should be shown. Think facebook status update page.
 - a. This will allow users who login to be updated on what’s going on in the site.

WORKSHOP DETAILS

1. A summarized product list and pricelist.
2. Shows who the “moderators” are, shows who the mechanics are here.
3. A summary of what services they provide, and pricing of the services if able.
4. Reviews and ratings

PARTS BRAND DETAILS

1. A summarized detail on what the parts brand is about.
2. Reviews and ratings.

MECHANIC DETAILS

1. Shows a history of where he worked, name, number or as much as he wants to show.
2. Specialises in what car?
3. Specialises in what service?
4. Reviews and ratings.

CLASSIFIED ADS

1. Basically a collection of all used parts for sale, categorized.
2. User can choose how they should be contacted. Email or HP (in which case HP must be entered into his profile, or automatically updated when he lists the item for sale)
3. We don’t really want a PM system.

REVIEWS AND RATING

1. Review anything you can think of. Car brand, car makes, parts brand, parts, workshop, mechanic, other user, advertorials, and editorials.
2. Ratings can be of 1-5 star type (for most cases) or Thumbs up/down format for advertorials and editorials.
3. All ratings must be justified with some words. Maybe 50 characters will be able to sieve out jealous competitors from defaming someone.
4. Disputing system should be in place.
5. Admin should have the option of making all ratings review before post, post and then review or no review required.

ADVERTISING MODULE

1. Fully integrated payment and implementation, with setting of the payment gateway Hands off approach as far as possible.
2. Zoned advertisements. User can purchase zone feature of products or workshops.
3. Zoning levels are as follows.
 - a. Home page feature
 - b. Car brand feature
 - c. Car make feature
 - d. Car parts feature
4. Workshop users can purchase featured workshop for the same zoning level as above.
5. An example of pricing for zoning as follows. A more zoomed in zoning, cheaper pricing.
 - a. Home page feature \$X
 - b. Subaru feature \$X-B
 - c. Subaru WRX feature \$X-B-C
 - d. Subaru WRX Exhaust feature \$X-B-C-D
 - e. Exhaust Feature \$X-B
 - f. Subaru Exhaust feature \$X-B-C
 - g. Product specific feature \$???User can basically select several criteria for their campaigns.
6. Zoned Banner advertising should be managed thru here too. Google adsense integrated please. Admin editable at all times.
7. Workshops can pay to have their logo displayed in the product listing results. Otherwise no logo is shown.
8. If too many people are bidding on a set of criteria, they should be able to auction for the limited space available. This will ensure maximum monetizing of the site.
9. Integrated purchasing of advertorials, with all details collected from the advertiser. Article provided to us? Cheaper. We write for them? Either pay more for our time (loan us parts for us to review) or negotiate if they want to.
10. Because we can't possibly charge a large workshop \$100 to feature their workshop and get their name featured in 1000 of their products they sell while we charge \$100 for a workshop that sells only 1 product and hence featured only in 1 product, we'll rather charge ie \$10 for workshops to feature themselves in 1 product. Of course we should have the capability to charge people by packages ie \$500 to feature in all their products regardless of how many they have.

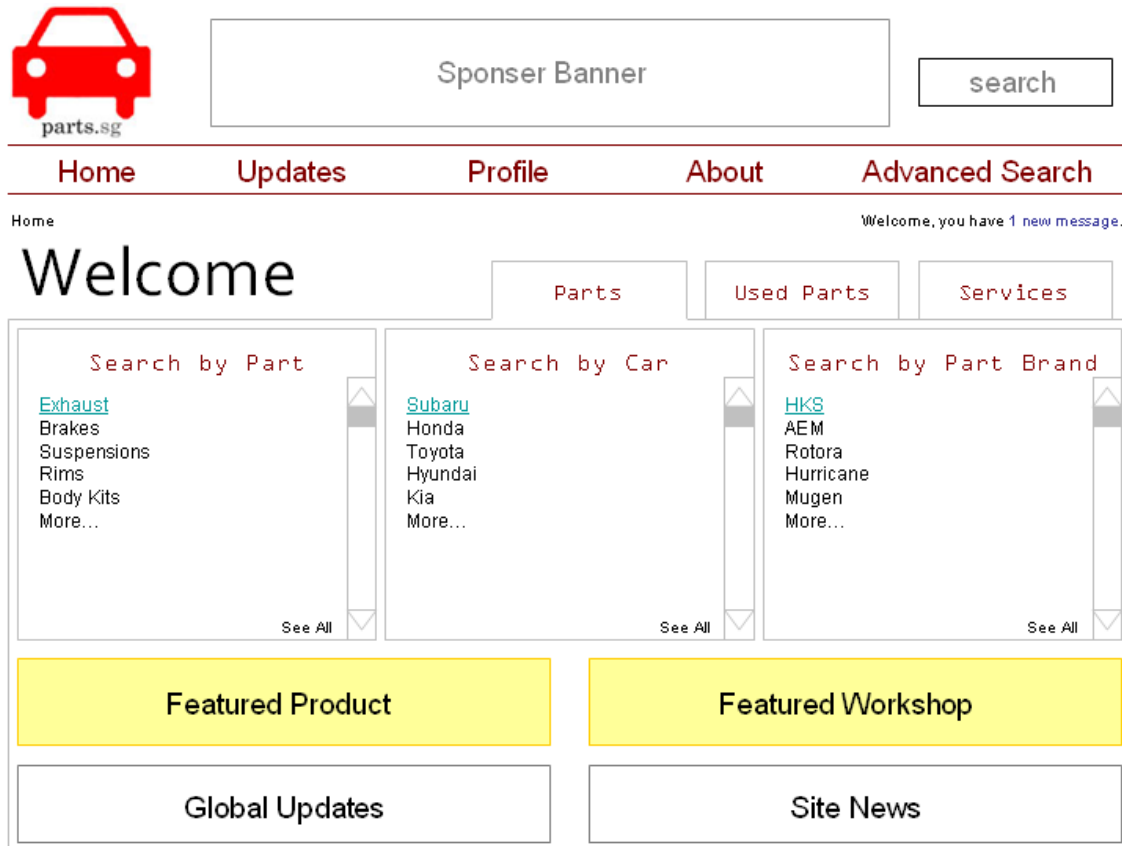
That's all we can think of so far! Looking forward to a quote. Thank you for listening to our blabber.

Gerald Tan
Gerald@parts.sg

Ong Pe Hon
pehon@parts.sg

Appendix E: Review of the Prototype

To search for Services, there should be a submenu to make it easier to find where to search for Parts, Services and Used Parts. There are different ways to implement this in the design, but that will not be discussed here. Below an example of the new view can be seen.



The Advertise main menu item will now be accessible through the “About” page. Near the different banners a small link should be placed to notify people that they can also advertise on this website.

Users can claim ownership of a Part by adding it to their Watchlist. Workshops claim ownership when they click on “I retail this” or “I distribute this” when adding the Parts to their Workshop. This will make it possible to see the updates made to the parts right away when on the Updates page.

A Service such as Grooming will have a wiki-like page just like Car Brands, etc, have these pages. A Workshop will also have their own page for the Service because the services can vary a bit between Workshops.

A Workshop can be a Distributor of a Part and then would also appear in the Retailer section. However, if the Workshop did not pay to appear at the top or with a logo, the Workshop will probably be below other Workshops.

A Mechanic can choose his/her specialty from the Car Brands, Car Types and Cars.

The Watchlist will be moved to the Profile page (it is currently on the Updates page).

On the Profile page the Most Active List will only show the most active users from the last 7 days. All the activity updates (today, last 7 days, this month) will be shown on the Updates page.

If a User is part of a Workshop but is not a Mechanic, he/she will be a Moderator.

Members will be Users who like the Workshop and visit them frequently. So you can see them as the Regulars (this name will be changed, probably). These people will also be listed on the Workshop page.

Mechanics will not see the Workshop at their Profile page unless they have Moderator tasks.

On the Service page of a Workshop, the Related Services will only be Services of that Workshop.

“Something else here” on the Updates page for Workshops will be: “Create new update” of “Create news item”.

To feature Parts and Services, another option behind the items will be [Feature This].

A [Retail All]/[Distribute All] option will be shown on the wiki-like pages of Brands, etc. Then an overview of all the parts will be shown and the Workshop can still delete one or two Parts if need be.

The Featured Part option and the whole advertising aspect will need to be discussed in more detail later with Micheal and Sherry because they have more expertise on this area. The details of the advertising system en featured system are still unclear at this point.

Appendix F: Planning of Parts.sg

Tasks	5/25	5/26	5/27	5/28	5/29	6/1	6/2	6/3	6/4	6/5	6/8	6/9	6/10	6/11	6/12	6/15	6/16
Users	Users																
Register	x																
Login	x																
Edit Profile	x																
Post article			x														
Post Part		x															
Post Used Part		x															
Post Comment		x															
Log			x														
Moderator					x				x	x			x				
Search																	
Display																	
Part							x		x	x							
Used Part							x		x	x							
Car Brand							x		x	x							
Part Brand							x		x	x							
Product Series							x		x	x							
Car Type							x		x	x							
Workshop																	
Create Workshop Users																	
Create Workshop Parts																	
Create Workshop Services																	
Management System																	
Watchlist																	
Create Watchlist function											x						
Display Recent Activity										x							
PM system																	
Advertising system																	

Appendix G: The Functionality Matrix for Parts.sg

Name	Description	Object	Data
logIn	User log into the site	User	username password
createUser	A visitor registers on the site	User	username password email
searchServices	A user can search for services offered by workshops.	Search Service	category keywords
searchParts	Finds global parts	Search Part CarBrand PartBrand	category carbrand partbrand keywords
searchUsedParts	Finds Used Parts offered by users.	Search UsedPart	category carbrand partbrand condition other attributes keywords
editProfile	Change the User attributes.	User	set attributes
createUsedPart	Create a new Used Part object. Notify users who have the related global part on their WatchList (WL). The status is available.	UsedPart	set attributes condition price
editUsedPart	Notify the users who have this used part on their WL. Edit the attributes of a used part.	UsedPart	set attributes
createPart	Create a new Part object. Notify the Moderators.	Part Temp	set attributes
editPart	Set the attributes of the part. Notify the users who have this part on their WL. Notify the Moderators and show them who changed what exactly.	Part Temp	set attributes
addEditorial	Create the text and title for an Editorial. Notify the Moderators to publish it. Update users Activity Rating.	Article User Temp	author -> user title date text body
addReview	Notify the Object owner: Wmoderator users who have the item on their WL Mechanic User	Workshop Part Mechanic User WService UsedPart	text body (comment) rating
createWorkshop	Create a Workshop object. Notify the Moderators.	Workshop WModerator	set attributes set permissions

	Make the user Chief Moderator of this Workshop (has all permissions).	Temp	
editWorkshop	Edit the Workshop information attributes. Only allowed by a Wmoderator with the correct permissions.	Workshop	set attributes
manageWUsers	Notify the user. Accept, decline, remove and add users of a Workshop. Change their permissions if they are Wmoderators.	User WModerator Mechanic Workshop	user id Role
manageParts	Notify the users who have the workshop on their WL. Notify the Wmoderators on Updates page. Can change price, add and remove parts.	Workshop Wpart	Price status in Workshop
manageServices	Notify the users who have (the Service of) The Workshop on their WL. Notify the Wmoderators on Updates page. Can add, remove and edit the information.	WService User Workshop	Service attributes status in Workshop
manageAdvertise	Buy different advertising options. Show logo on parts pages. Show as a featured workshop on homepage. Show as a featured workshop on part page.	Workshop	Paypal selected options
joinWorkshop	Adds the user to 'pending'. Notify the Wmoderators.	User Workshop	user id workshop id Status Permissions isMechanic isModerator
createMechanic	After a users wants to join a Workshop, the user is asked for more information if they are a Mechanic. If the request is approved by a Wmoderator, the new Mechanic object will be created.	Mechanic	set attributes
editMechanic	Edit the attributes of a mechanic.	Mechanic	set attributes
createWModerator	When a user is accepted to the Workshop, he/she can get permissions. Then a Wmoderator object is created with the given permissions.	WModerator	set permissions
edit Wmoderator	Edit the permissions of the Wmoderator. Can only be done by another Wmoderator with the correct permissions.	WModerator	set permissions
manageWL	Add and remove items from the WatchList.	User Part UsedPart Workshop Service WService	id of each item
recentActivity	Keeps track of all the changes made to the items in the WL.	User Part UsedPart	

		Workshop Service WService	
latestPart	Display the latest parts.	Parts	part name part date added
activitiesRecord	Keep track of all the activity levels of all the users.	LogTable User	activity rating
manageMessages	PM system which lets users send, receive, view, remove, forward and reply messages.	User Message	to userid from userid title message text message body
manageModerators	Add new global moderators. Removing mods can only be done by the admins.	Moderator	
viewChanges	See the changes made to the content of the site and accept or deny them. Also shows which users did what, making it possible to ban users.	Moderator Temp User Part Service etc	set changes
manageContent	Delete or add editorials, new brands, categories, news, etc.		
manageUsers	Ban, delete, create users.	User	user id