ImpliciTunes

A learning-free novel music recommender system using contextual sensor data

Abhishek Sen

Master of Science Thesis

Delft University of Technology

Multimedia Computing Group, Intelligent Systems

ImpliciTunes

A learning-free novel music recommender system using contextual sensor data

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Electrical Engineering at Delft University of Technology

Abhishek Sen

October 22, 2015

Faculty of Electrical Engineering, Mathematics and Computer Science (EWI) · Delft University of Technology

Table of Contents

List of Figures

List of Tables

Preface

November 21, 2014—The day I conceived the idea for my thesis. I was taking part in the Agile Product Roadmapping course at Universidad Politecnica de Madrid as part of the Athens Program with two other friends from TU Delft. The Apple Watch had just been announced a couple of months back and one of my dorm mates and I were sparring over the usual topics that geeks talk about—technology. We asked ourselves—What applications could the Apple Watch be used for? We knew it would have an accelerometer, GPS and most notably, a pulse sensor to measure the user's heartbeat. The sparring continued and mid-way through the conversation, the idea just came to me. Why can't we use our own heartbeat to measure the speed at which we walk and use that information to recommend music? Interesting. I discussed this idea with some other people and at first it seemed like just another one of those ideas that come and go as fleeting thoughts. I thought—let's park it and keep looking.

Upon returning to Delft, I still had no idea what to do my thesis on. A few days after I was working out at the Sports Centre on campus, and was getting really tired of listening to my music. I had to manually scroll through my 'likes' on SoundCloud with sweaty hands and pick and choose songs that I thought would uplift my spirits and help me have a good workout. I cannot recall how that workout went but it was on the way back home on my bike it hit me again—Why could my phone not know where I was and what I was doing and recommend me music without me having to enter any information? I am pretty sure my smartphone (iPhone 5S) had all the inbuilt technology needed to do this and I discover lo...—And boom, just like that, I knew my thesis topic!

I sent an e-mail to Prof. Alan Hanjalic on November 28, 2015 requesting a meeting and we agreed to meet on December 3 together with Martha Larson (my future supervisor). At the meeting I enthusiastically pitched my overall vision for the thesis and thankfully, both Alan and Martha shared my enthusiasm and agreed to let me join their group as a Master's student to work on my idea. Fast forward to January 5, 2015—the journey officially started!

I have a lot of people to thank for helping me in different ways. First off, I have to say a big thank you to my awesome supervisor Martha who was always encouraging, supportive and continually challenging me to keep performing on all cylinders. Even though she was in California for a good chunk of my thesis, we somehow managed to work out the 18:00

time slot on Thursdays for most weeks. I appreciated the opportunity to travel to Vienna to present our paper at the ACM RecSys LocalRec 2015 Workshop. I was able to meet and interact with a lot of smart people and most importantly, got some very good feedback about my thesis. I would like to thank Mark Melenhorst for his feedback on the user surveys and the focus group. I would also like to thank Alessandro Bozzon for agreeing to be on my thesis committee and most importantly, for giving me critical and constructive feedback on the first draft of my thesis—very helpful! And then Alan of course for allowing me to join his group as a Master's student. Finally, my work in this thesis also benefited from collaboration and discussions with other members of the EC-funded CrowdRec project.

These last 9 months and 17 days have been an incredible and rewarding journey personally. I consider myself super lucky to have found something to work on that I was (and still am) so passionate about that I honestly did not mind working on it day and night (my roommates can confirm this). There were numerous nights where I just kept staring at my whiteboard, scribbling notes, looking for inspiration and at times completely lost in the whole process. My double screen (laptop + external monitor) and I became best friends. What kept me going forward was that I really wanted to build something for myself.

These last two years in Delft have been a really amazing experience for me. I have met some amazing people, made friendships that will last a lifetime, laughed a lot ... like a lot! Every time I came back to Delft after visiting some place, it really felt like I was coming back home. Thank you to all the people listed below for helping make these last couple of years so memorable.

> "Matteo Ciocca, Dean Gioutsos, Alessandro Cattapan, Sam Ryan, Trine Gedde, Mar Sole, Jennifer Lee Brown, Rafed Mohammed, Matthijs Heeres, Stijn Jochems, Javier James, Arundhati Radhakrishnan, Ruud Visser, Martina Karner, Arnold Schutter, Gagan Reddy, Fouad Saleem, Justin Burford, Viki Pavlic, Sara Eszter Kis, Osman Ay, Laura Bell, Sreejith Chandrasekhar, Marina Nano, Rodolfo Solera, Aman Kalsi, He Ming Zhang, Michiel Gerlach, Joost de Haan, Evan Shorachi, Xin Wang, Luis Alfonso Gonzalez, Harbir Gill, Naji Shafi, Dennis Willaert, Laura Zambelli, Laura Amsing, Babak Loni, Karthik Yadati ... and many more"

I would also like to thank my girlfriend Gwendoline Mommeja for being there for me these last two years. And finally, I would really like give a big shout out to my parents for everything that they have done for me. Enjoy the read!

Cheers,
Abhi. Thursday, October 22, 2015.

Chapter 1

Introduction

People love listening to music and so do we. Today, music streams are readily available through services such as YouTube, Spotify and Apple Music. The digitization of music has eased the way for users to gain access to large collections of music through the Internet. With all this music around us, it would be expected that music discovery is a settled field of research when in fact it is quite the contrary. Novel music discovery remains a challenging and a tedious task for users. Existing techniques such as content-based and collaborative filtering approaches recommend new music based on the past listening history of users. While there are obvious merits to such approaches, they also have some serious limitations with regards to music discovery. By focusing on the past, it becomes increasingly difficult for users to discover music that is different from what they have listened to in the past. Using accuracy as an evaluation metric, as the system become more 'accurate' at giving you the same kind of music recommendations, serendipity and novelty often take a backseat. Cold-start is a well-known challenge that affects content-based and collaborative-filtering based music recommender systems. When a new user joins the system, because they have not yet rated or listened to anything yet, there is very little available information based on which any credible recommendations can be made. Lastly, traditional approaches to music recommender systems do not take into account the user's surroundings for music recommendations—thereby missing out on an opportunity to overcome the challenges put forth above. Our goal for this thesis is to design, implement and evaluate a learning-free novel music recommender system that allows users to easily discover new music based on their current surroundings.

1-1 Current relevant technology trends

We think it is very important to consider current technology trends before addressing some of the challenges laid out in the previous section. Given that these challenges have direct user-facing consequences, it is critical to identify and understand how users listen to music using the technologies they use on a daily basis. By understanding users and their problems better, we make sure that the solution proposed by this thesis is relevant in today's fast paced technology industry.

1-1-1 Internet Of Things (IoT)

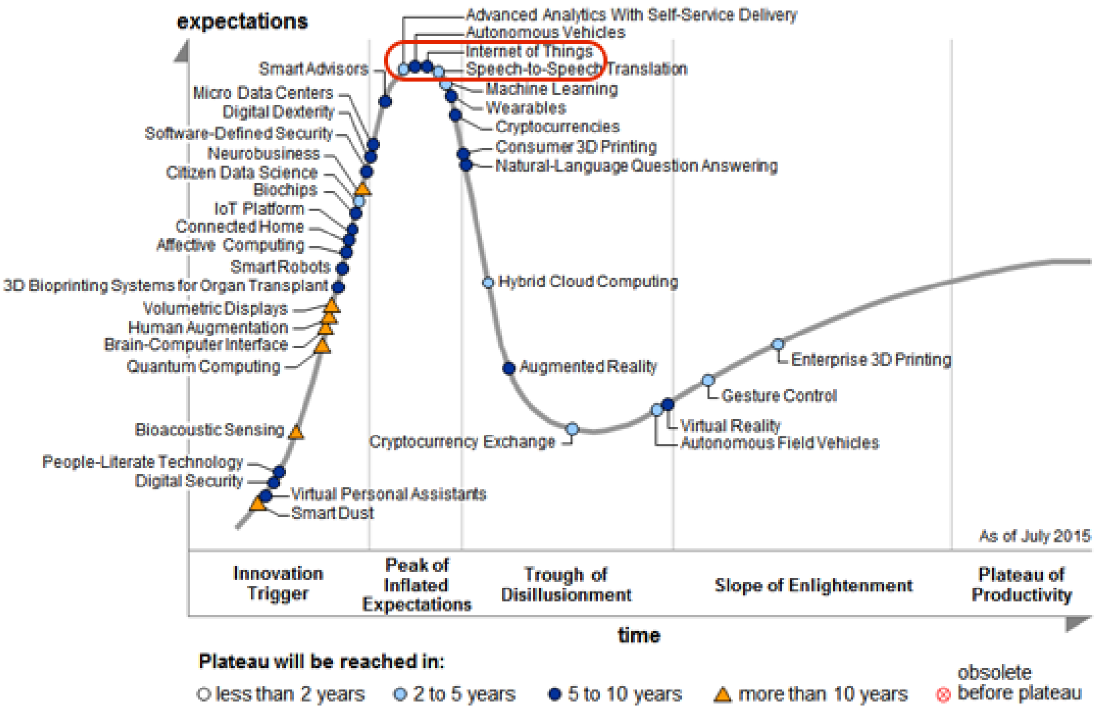


Figure 1-1: Gartner's 2015 Hype Cycle for Emerging Technologies [1]

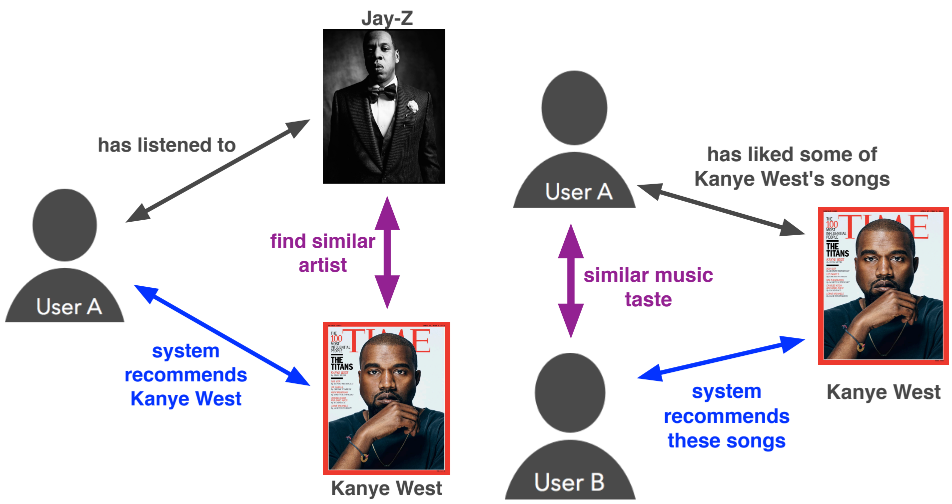
The IoT paradigm promises to connect human beings to their environment and aid in building a world where all objects around us are connected to the Internet and communicate with each other with minimal human intervention. These objects know what we like, want, and need and can act accordingly without the need for explicit instructions. From Gartner's 2015 hype cycle of emerging technologies in Figure 1-1, we see that IoT is on the cusp of breakout user adoption. Future forecasts predict usage estimates of almost 50-100 billion connected devices by 2020 [2]. With all this technology around us, it is natural for users to expect smarter technology without compromising on privacy. Ever since the release of the first Apple iPhone, smartphones have become central to how we connect to the Internet and in doing so, to the world. In future, more novel and efficient ways of interacting with our environment will emerge. With this inevitability on the horizon, we think that the music recommender systems community really ought to take notice and start incorporating contextual sensory data into their algorithms and systems to improve recommendation novelty.

1-1-2 Streaming Music

Fifteen years into the 21st century, we find ourselves at a crossroads where music streaming has overtaken downloading as a means for music consumption. A recent study conducted by Nielsen shows that of 3305 American consumers surveyed, 75% of them listened to music online while 44% reported to using smartphones to listen to music during a typical week [3]. In terms of music discovery however, radio (AM/FM or satellite) is still the most popular source of discovering new music (67%) followed friends/relatives (45%) in second place and online audio streaming services at fourth with 27%. With a general trend towards streaming music, many music services are trying to learn user music tastes and their listening behaviors to provide the best personalized music recommendations. We believe that it is important for the academic world to recognize this behavioral change in how users listen to their music and this thesis aims to develop algorithms and technologies that follow these current user trends.

1-2 Traditional approaches to music recommender systems

Traditional approaches to music recommendation include content-based filtering, collaborative filtering, demographic filtering or a combination of these three approaches. Figure 1-2 shows the differences between content-based filtering and collaborative filtering. Content-based filtering looks at the content of the song and the user's listening history to make novel music recommendations. For example, in Figure 1-2a we see that User A has listened to Jay-Z which prompts the system to find artists similar to Jay-Z to which the system finds Kanye West and recommends him as a potential artist recommendation that User A might also enjoy listening to.



(a) Content-Based Filtering **(b)** Collaborative Filtering

Figure 1-2: Traditional approaches for music recommendation

In the context of music, content-based filtering looks at things such as genre, audio signal characteristics and country of origin. From this analysis, music that is similar to what the user has listened to or followed in the past is then recommended to the user as new music [4]. Collaborative filtering on the other hand looks at music taste similarities between users for music recommendation. It allows users to give ratings about songs, artists and playlists in a way that when enough information is available, the system can make recommendations for each user based on information provided by those users who have the most items in common.

We see an example of this in Figure 1-2b where User A liked some of Kanye West's songs and then the system finds out Users A and B have similar music tastes and thus recommends the same songs that User A liked to User B.

1-3 Challenges with traditional approaches

A major issue with current music recommender systems lies in the assumption of the learning approach, namely, that past behavior is a good predictor of future behavior. Such services certainly satisfy users looking for a highly predictable music experience. However, users interested in expanding their music horizon will not be satisfied by algorithms that rely solely on past listening history since they do not support new music discovery. Bonnin and Jannach mention the importance of context in automatic playlist generation and how similarity-based algorithms are an obvious approach when the system's goal is to maximize the homogeneity of the playlist [5]. As a downside however, serendipity and diversity are negatively affected since most songs recommended will be of a similar type. They suggest future research works to assess multiple criteria at the same time and explore the trade-offs between homogeneity and diversity in music recommender systems. Traditional approaches can only take us so far and they fail to provide the serendipity that is extremely important for users to discover music that is new, but is also not completely alien to them. Sen and Larson mention in [6] that if users are limited in their music discovery by what they did in the past, any future music discovery will also inherently be limited unless users take a drastic step that forces us outside of our comfort zone and experience something new that they might never have discovered with traditional music recommender systems. Zhang et. al add further weight to this misplaced importance placed on accuracy in recommender systems and the dangers of such an approach in [7]. They describe the extreme concept of a “filter bubble" where users are trapped in a self-reinforcing cycle of opinion and never being pushed to discover new music and truly expand one's music horizon. In particular, they conclude that users being willing to sacrifice some amount of accuracy for improved novelty, diversity and serendipity and this results in higher overall user satisfation.

Another challenge that traditional music recommender systems face is that of cold-start. With content-based approaches, when a new song is added to the system, since there is no prior history, the recommendations cannot be personalized very easily and often go unnoticed by users simply because it has never been played. In collaborative filtering, when a new user enters the system, since they have no music history, the system fails because the user-item matrix is devoid of any items. This makes it extremely difficult for new recommendations to be made. Some known approaches to overcome this issue have been to use hybrid approaches, collaborative tagging and extracting meaningful information from user profiles to jump start the recommendation process [4]. Commercial music application services use onboarding techniques such as asking the user their favorite music genres, artist types and other relevant information to overcome this cold-start problem.

Lastly, the process of discovering new music remains a tedious and cumbersome experience for users. We think that music discovery happens either actively or passively. When active, users make a conscious effort to discover new music by searching on the Internet or music applications and browse through multitudes of music trying to find new music. This process can often be tedious and time consuming. In the passive mode of music discovery, users simply go about their day and find out about new music from their surroundings—friends, movie trailers, cafes and nightclubs amongst others. Thus, we rely on our surroundings to help us discover new music. In a way, we leave it up to chance. We want to address all three of these challenges and make novel music discovery easy, yet, serendipitous.

1-4 Motivation

As we read in Section 1-1 about the current related technology trends, we could not help but ask ourselves—"Why is it that in this day and age of technology when we have so much computing power at our fingerprints, can our devices not work for us and provide music recommendations by responding to our surroundings? And why is it that finding new music is still a tedious and cumbersome task, even with all the music available to us through multiple sources?". We know that the technologies required to make this happen already exist in most current smartphones. We also know that the challenges facing current music discovery systems are far from over as discussed in the previous section. So the real question of motivation is not to ask why this topic is interesting and worthy of solving, rather to ask, why not?

1-5 Formulating the research goals

In order to research, design, build and evaluate a learning-free novel context-aware music recommender system, we started by making a new and important assumption. We considered music listening to be independent of the past (history) or the future (prediction) and instead consider it as a function of the present (current context). We use the term *context* to refer to the sum of a user's experience at a given moment, including place, surroundings, activities that the user is currently pursuing and weather conditions. We assume that listeners have similar expectations of which music fits a particular context. We also rely on the idea that this collective conception of 'music that fits a moment' will provide users with a sense that the recommendations of our system fit their current needs, and at the same time allows them to discover music that they would not have otherwise found themselves. With these goals in mind, we set about formulating our main research question:

> **"Can novel music be recommended from contextual situations inferred from sensory data?"**

To help answer this research question, it became important to break things down into sub-questions. We realized the idea of using sensors for music recommendation is too abstract thus we came up with the following sub-questions:

- "What information shown on the user interface conveys a sense of context to users?"
- "Which playlist type (situational or atmospheric) do users prefer to listen to?"
- "Is content-based song re-ranking relevant and/or necessary for novel music recommendations?"

Through the various sections in this thesis we will answer these questions, especially in the evaluation and results and discussion chapters (Chapters 6 and 7).

1-6 Defining the scope

Having presented our motivations and research questions for this thesis, it is important to define the scope of this thesis from the onset. Our goal is to research, design, build and evaluate a system that is able to recommend novel music to users by using contextual information from sensors within the given time frame of 9-10 months. Some of items presented below are outside the scope of this thesis because we want our system to be as close to production-ready as possible and this means saying no to some things. We are not building a framework that will be hacked together without paying attention to the user experience; rather we want to involve users in the design process so that when users evaluate the system, they find it useful and easy-to-use.

The items below were considered in our initial design but eventually left out as future extensions to this thesis:

- The Apple Watch was our initial launching platform of choice but since the first version of the software did not allow developers access to any of the sensors (pulse sensor, accelerometer, GPS), we decided not to wait and continued our system implementation for smartphones. We could not wait for the Apple Watch to release and then perform this research because smartphones already have all the technology needed to design a music recommender system based on sensors and we did not want to fall behind on the technology curve.
- iPhone 5S and newer models contain a MX motion detection microchip that allows developers to take get information about the device's motion activity without having to develop their own detection algorithm. Developing a new activity detection algorithm was not the goal of our thesis and also given the time limitations, we would not have had adequate time to evaluate and ensure consistent performance between iPhones prior to 5S and newer. Hence, our system will support iPhone models 5S and newer.
- We decided to limit our application environment to smartphones because unlike tablets, smartphones are a lot more mobile and users carry them around at all times which allows us to infer their current situation more accurately.

With the scope of this thesis identified, the last section in this chapter explains the structure of this thesis for our readers.

1-7 Thesis structure

This thesis is divided into 9 main chapters. Following this introduction, we present the relevant related work in the field of context-aware music recommender systems, human psychology, emotion classification and other areas that helped us identify potential knowledge gaps in research and formulate our research sub-questions (Chapter 2). Having identified the technology gaps, we then state our design principles or guidelines which we will be abiding throughout the journey of this thesis (Chapter 3). Following the design principles, we discuss the overall design methodology for our system and how we tackled different aspects of the problem associated with designing a context-aware music recommender system (Chapter 4). The next chapter discusses in detail our proposed design and all the design decisions that went into building the final application (Chapter 5). The next two chapters deal with formulating an evaluation strategy and then discussing the results from our user evaluation (Chapters 6 and 7). Based on the experimental results and user feedback, we present the overall conclusions for the thesis in the next chapter with the subsequent chapter discussing areas of future extensions of this thesis (Chapters 8 and 9).

Chapter 2

Related Work

This chapter gives the reader an overview of relevant academic material we perused and inspired the design process, implementation and evaluation of our system. It was important during the initial months of the thesis to aim for a breadth-first approach in order to understand current research techniques and identify any knowledge and technology gaps. The material presented here is not in a chronological manner, rather in topical groups that cover different aspects of this thesis. We conclude the chapter by presenting the overall outcomes from the background literature study and present the knowledge and technology gaps that were identified in the process.

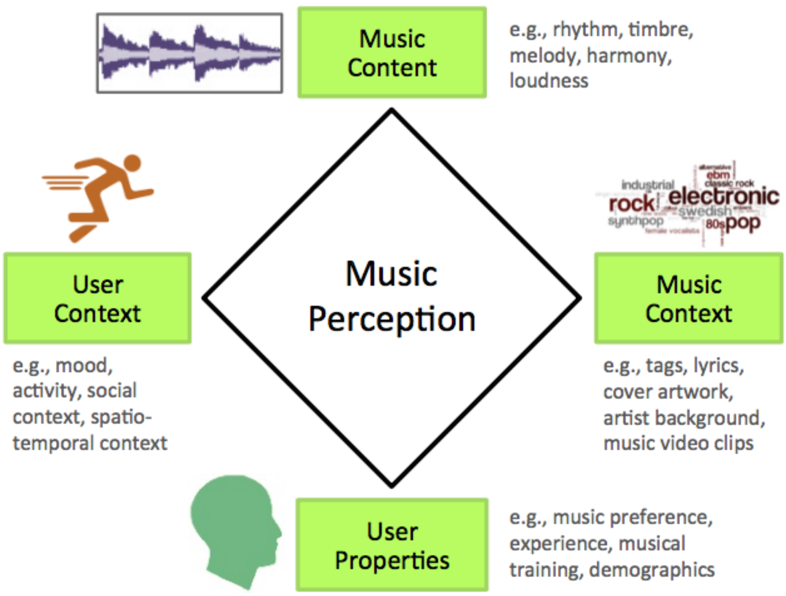


Figure 2-1: Four different categories of factors influencing music perception [8]

According to Knees and Schedl [8], music perception is comprised of four categories—music content (e.g. rhythm, tempo, loudness), music context (e.g. tags, lyrics, cover artwork), user context (e.g. mood, activity, social context) and user properties (e.g. music preferences, demographics). Figure 2-1 shows the model that they propose as the factors that influence music perception. We too adopt an approach wherein multiple factors besides user preference and past history play into the process of music recommendations.

2-1 Context-aware music recommender systems

One of the key problems with music recommender systems today is that they are not context-aware. Even with all the smartphones that we carry around with us at all times we still use them in very limited ways. Traditional music recommender systems do not take advantage of these resourceful data sources. The inherent capability of sensors is to 'sense' their environment to gather information (e.g. location, acceleration, motion), thereby making them an ideal choice for interpreting user context. Dey defined context as

> *"Any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"* [9].

We think that incorporating contextual information into recommender systems will change the way music is discovered. This process begins by understanding the situation where the user is listening to their music. Sensors along with web-accessible contextual data form the core of these additional sources of information that can be used to greatly enhance the user's listening experience. In this section we discuss some of the context-aware music recommender systems that have inspired us.

Wang et. al propose a system that is context-aware, probabilistic and learns the user's listening habits over time for better recommendations [10]. Their system utilizes contextual sensor data and integrates this information with music content analysis to provide relevant music recommendations for each context. Their system uses data from an accelerometer, gyroscope, time data and microphone as contextual sensors based on which 6 activities were classified. The context classification was performed by collecting in-field test data followed by offline classification methods. There were 1200 songs used by the authors; these were annotated by users prior to the test. The paper explains how they avoid the cold-start problem because the proposed system is context-aware. The authors mentioned that even though the contextual annotations could have been subjective between users, their test users generally agreed with the annotations. Although in the right direction of building a context-aware system, their system has some limitations such as the requirement of user annotation of the music and a necessary training phase for context detection.

Cunningham et. al propose a rule-based fuzzy logic system to infer the user's context and then use that information to recommend music [11]. They demonstrated their application with a Wiimote and used contextual information such as activity states, temperature, light sensor and weather conditions. The system's output is an emotive state which then maps to one of the 10 songs used for testing. Their test bench was a Wiimote which means the user was restricted to listening to music at home. This in turn also means that all the other contexts which were tested could not have been tested in a real-world situation.

Nirjon et. al proposed a biofeedback-based, context-aware, automated music recommendation system for smartphones, including a new wearable sensing platform consisting of a pair of sensor-equipped earphones that could communicate with the smartphone via the audio jack [12]. The goal of their system was to recommend music to match the user's heart rate with music from a similar tempo by using data from sensors such as a pulse sensor and accelerometer. However, since their system's goal was to recommend songs for a specific heart rate, their music is limited to only the exercising context which is where they evaluated the application with users. Yu et. al present an interesting system where they performed location-to-mood tag mapping and then retrieved songs from a database built from Last.fm and YouTube. The categories used for mood mapping were adopted from Foursquare's location categories and then based on a subjective user study, each of these categories were then mapped with certain moods [13].

Lee et. al present a case-based reasoning system for a context-aware music recommender that takes into account the listening context of the user (weather, date, time, season), listening history and music content for recommendations [14]. Their system takes in information about the user's context, tries to find users who have listened to music in the similar context, retrieve similar users that have listened to music in a similar context and finally select and retrieve songs from the music database. Liu et. al present a heart rate controlled music recommender system that recommends calming or uplifting music with a target heart rate and try to bring the user's heart rate back to normal ranges [15]. They evaluated their system by simulating a virtual in-flight experience for passengers on a real airplane with 12 participants. A major takeaway from their paper was that heart rate variability is a better indicator than heart rate for more accurate measurements and that it could indeed be used for music recommendations to reduce user stress.

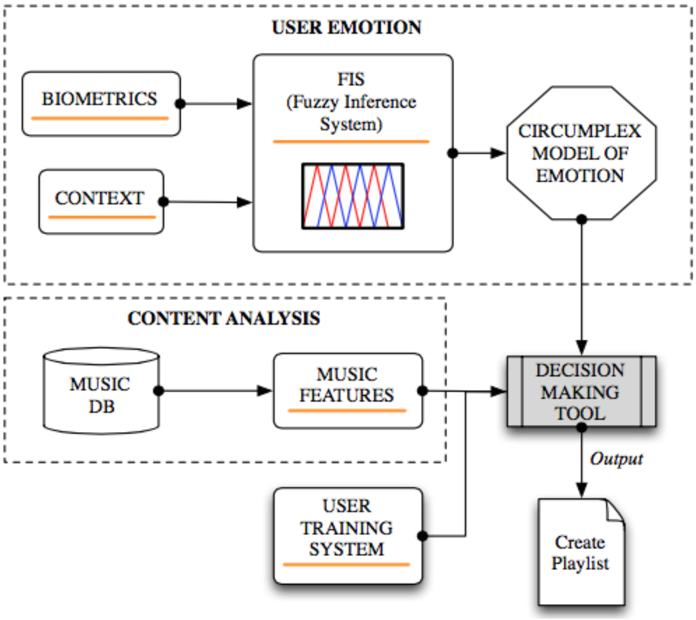


Figure 2-2: Griffiths et. al proposed system of using contextual and physiological to quantify human emotion and automatically generate music playlists [16]

Griffiths et. al propose a system that combines content-based music analysis with context-awareness to quantify human emotion and then use this information to automatically generate music playlists [16]. They use biometric sensors to define the user's emotional state by measuring high-level emotional properties such as valence, arousal and body temperature. In addition, they also propose to use environmental sensors to gather data such as outdoor light, sound, temperature and humidity. Figure 2-3 shows how their proposed system is expected to work. The fuzzy logic inference system generates values for valence and arousal using Russell's circumplex model of affect and then songs with similar affective parameters are retrieved from the music database and recommended to the user [17]. The authors try to incorporate the aspect of human emotion modeling and use that to recommend music and this effort is interesting albeit incomplete and leaves room for improvement in the areas of context-awareness. There were no evaluation details discussed in this paper.

Kaminskas et. al in their paper titled "Location-Aware Music Recommendation Using Auto-Tagging and Hybrid Matching" propose a context-aware music recommender system for points-of-interest (POI) [18]. They selected 25 well-known POIs from 17 cities around the world for the evaluation. For each POI, the top-5 artists were retrieved and then the top-3 songs for the artist were used to build the dataset. The hybrid approach presented in the paper combines the tagging of POI and music and the semantic relationship between the two to generate context-aware recommendations. The songs were tagged by users and also through an auto-tagging technique which used a controlled tagging vocabulary. They then used genre-based and tag-based approaches for the music recommendation. One of the shortcomings here was that the authors performed all the evaluation offline i.e. without requiring the testers to actually be at the points-of-interest in order to confirm if the music recommendations were good or not. The synthesis of the user-based tagging and the auto-tagging techniques were an interesting approach to annotate new songs.

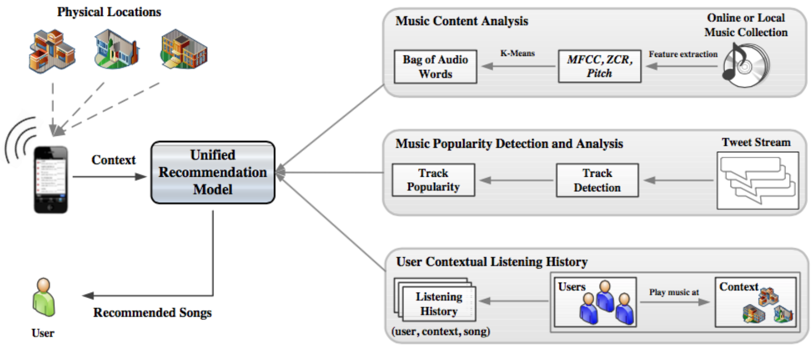


Figure 2-3: Cheng et. al's music recommender system architecture [19]

Cheng et. al present a novel recommender system by considering the user's location and the global music popularity trends for that location to recommend music [19]. They claim that while a user's listening behaviors can affect music popularity trends, the selection of a song is highly related to the user's preference of music content for that context. In their model, the user can be listening to a music with some specific 'audio words' at a given location—called the three-way aspect model. We tend to agree with this approach to building a context-aware recommender system, especially with the integration of contextual information with content-based and social recommendations. It is interesting how they try to learn a user's behavior for a given location context and then use that history for future music recommendations—only for that user. Their conclusions show a strong correlation between music popularities of tracks and user listening behaviors. A possible shortcoming of selecting songs based on popular music is that unpopular or new songs might not ever get recommended because they might not be mentioned on social networking sites like Twitter, thus limiting the music choices for users looking to discover music.

Okada et. al present a system that focuses on the user interface aspects of context-aware music recommender systems, an area often ignored by researchers [20]. One of their core objectives was to explore how context plays a key role in a user's listening behavior and how this information can be conveyed to the user. They defined context as a finite set of sensed conditions collected from a mobile device that could affect a given user's music-listening behaviour. It gave us inspiration to explore this area of user experience and how it affects the user's music listening experience.

As we have tried to show through our background literature study of context-aware music recommender systems, there has been a lot of recent work in this area which validates the relevance of this thesis. From all the papers presented in this section, we could derive some overall conclusions. While some papers had a clear objective in mind (matching user's heart rate, recommending songs that matched user's emotion and/or situation such as POIs), others involved the use of custom hardware (eg. Wiimote, external sensors, headphones) that would not really make for practical user applications in their current form. Another common trend with some of the papers was that the music recommendations were often performed with a controlled dataset of songs and not live production systems. A possible reason for such a choice could have been to reduce the number of variables while conducting a study but it makes the system less realistic since we know that real-world scenarios are often more challenging where network delays and latency issues are often the norm. As we see more and more sensors being integrated into the devices we carry (eg. smartphones, smartwatches), it becomes increasingly important to not simply correlate similar matching sensor readings amongst users but to use the raw sensory data to infer higher-levels of contextual meaning which would then allow for more meaningful music retrieval strategies. Going through these academic papers have opened up some knowledge gaps in this area and we will identify them at the end of this chapter.

2-2 Affect and music

We discuss affect as an introduction to this discussion of the impact that our surroundings have on our 'emotion' and 'mood' and how this in turn can influence our music listening behavior. This is by no means an extensive foray into the complex field of mood classification but an important factor that needs addressing in the context of this thesis.

2-2-1 Differentiating emotion from mood

We need to state the difference between emotion and mood as these two are often interchanged in daily conversation. The key difference between the two is that while mood is an affective state that stretches over longer periods of time and is not tied to a specific object, emotions are felt over shorter periods of time, often bursts and with higher intensities [21]. When it comes to music, we think given that our mood sustains over a longer period of time and not emotional spurts and that our current mood has a greater influence on the type of music that we choose to play.

2-2-2 Exploring the relationships between emotion and music and other influencing factors

Music is one of those art forms like video, paintings, sculpture and others that can trigger lasting or momentary impressions and emotions. Figure 2-4 shows a list of musical features and how a combination of them can be used to emote discrete emotions.

Emotion	Musical features
Happiness	Fast tempo, small tempo variability, major mode, simple and consonant harmony, medium-high sound level, small sound level variability, high pitch, much pitch variability, wide pitch range, ascending pitch, perfect 4th and 5th intervals, rising micro intonation, raised singer's formant, staccato articulation, large articulation variability, smooth and fluent rhythm, bright timbre, fast tone attacks, small timing varibility, sharp contrasts between "long" and "short" notes, medium-fast vibrato rate, medium vibrato extent, micro-structural regularity
Sadness	Slow tempo, minor mode, dissonance, low sound level, moderate sound level variability, low pitch, narrow pitch range, descending pitch, "flat" (or falling) intonation, small intervals (e.g., minor 2nd), lowered singer's formant, legato articulation, small articulation variability, dull timbre, slow tone attacks, large timing variability (e.g., rubato), soft contrasts between "long" and "short" notes, pauses, slow vibrato, small vibrato extent, ritardando, micro-structural irregularity
Anger	Fast tempo, small tempo variability, minor mode, atonality, dissonance, high sound level, small loudness variability, high pitch, small pitch variability, ascending pitch, major 7th and augmented 4th intervals, raised singer's formant, staccato articulation, moderate articulation variability, complex rhythm, sudden rhythmic changes (e.g., syncopations), sharp timbre, spectral noise, fast tone attacks/decays, small timing variability, accents on tonally unstable notes, sharp contrasts between "long" and "short" notes, accelerando, medium-fast vibrato rate, large vibrato extent, micro-structural irregularity
Fear	Fast tempo, large tempo variability, minor mode, dissonance, low sound level, large sound level variability, rapid changes in sound level, high pitch, ascending pitch, wide pitch range, large pitch contrasts, staccato articulation, large articulation variability, jerky rhythms, soft timbre, very large timing variability, pauses, soft tone attacks, fast vibrato rate, small vibrato extent, micro-structural irregularity
Tenderness	Slow tempo, major mode, consonance, medium-low sound level, small sound level variability, low pitch, fairly narrow pitch range, lowered singer's formant, legato articulation, small articulation variability, slow tone attacks, soft timbre, moderate timing variability, soft contrasts between long and short notes, accents on tonally stable notes, medium fast vibrato, small vibrato extent, micro-structural regularity

Figure 2-4: Musical features correlated with discrete emotions in musical expression [22]

	Never (%)	Sometimes (%)	Often (%)
When I wake up	43	37	20
While bathing	58	28	14
While exercising	25	36	39
While working	22	47	32
While doing housework	4	31	64
When relaxing	12	51	37
While eating	25	61	15
As background when socializing	6	48	46
As background to romantic company	17	59	25
While reading	56	36	–
Going to sleep	64	25	11
While driving, cycling, or running	8	46	46
On the train, bus, or plane	34	47	19

Figure 2-5: Situations during which users listened to music [22]

Juslin et. al conducted a study to explore expression, perception, and induction of emotion from the perspective of the average music listener in their everyday life [22]. Figure 2-5 shows the list of situations that users mainly listened to music—this is very useful to us because we can then focus on the most popular situations where users listen to music the most. According to them, there is an important distinction between perceived and induced emotion with respect to music. A variety of emotions can be conveyed by music features such tempo, mode, harmony, tonality, pitch, rhythm and others. Induced emotions are the emotions that we feel in response to listening to music whereas perceived emotion is when we perceive emotions in the music. This is an important distinction to make because according to the authors, measuring induced emotion is more difficult to measure than perceived emotion since the latter is relatively easily measured by asking a person what they felt after listening to a piece of music. One of their findings was that social context has a definite impact on the user's perception of the music with the strongest emotional experiences often occurring occur while listening to music alone as well as in a social or emotionally charged situation. Some of the basic emotions (happiness, sadness, tenderness, anger, fear) seemed to be easier to induce than some of the more complex emotions.

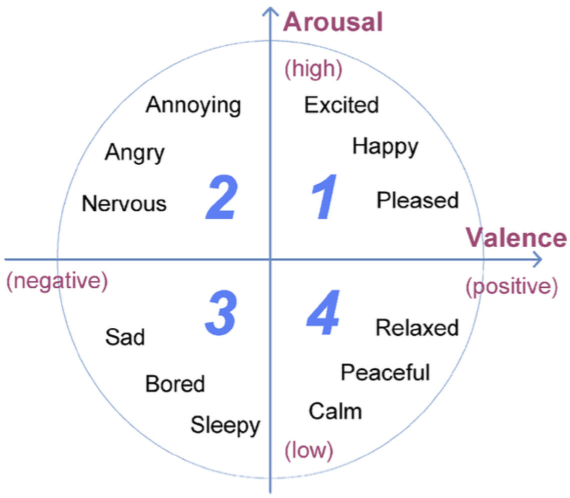


Figure 2-6: The 2D valence-arousal emotion space with approximate positions for the affective terms. [17]

Music information retrieval researchers have empirically validated the use of valence and arousal as the two emotion dimensions (borrowed from psychology). Figure 2-6 shows the two-dimensional valence-arousal emotion space from Russell's seminal paper in 1980 [17]. Laurier's contribution on the automatic classification of musical moods by content-based analysis validates Russell's two-dimensional emotion space model through the use of social music tags and the semantic emotion space they span [23]. According to Yang et. al in [24] social tagging of music through platforms such as *Last.fm* mitigate annotation subjectivity but the quality of annotations have been found to be relatively lower than subjective annotations where users rate songs on the valence-arousal scale between -1.0 to 1.0. One of the challenges they identify for future work in music emotion recognition is to consider the situational factors of emotion perception. Gabrielsson too concludes that our emotion response to music is dependent on an interplay between musical, personal, and situational factors [25] such as listening mood and environment, a person's emotion perception of the same song could vary a lot.

A lot of factors affect our daily mood, both internal and external. It is often difficult for a person to guess the reason for another person's mood because most often the cause is unknown. According to Hume [26], some of the common sources that influence our emotions and moods include—personality, day of week and time of day, weather, stress, social activities, sleep, exercise, age, gender, organizational and cultural influences. It is true that while there are no hard and fast rules as to what exactly influences our emotions and moods, we can see that there are multiple factors that have an effect. Two different people in the exact same situation could be experiencing completely different moods because of some internal or external factors that the outside world has no idea about. In our research, we wanted to see if there were certain factors that affect everyone, in small or big ways, regardless of their back story.

Even if we focused only on the type of influences the outside world could have on a person, there were a lot of unknown variables involved. The only thing that we knew was constantly affecting all users in given geographical area uniformly was the weather, day of week and time of day. Unlike other factors such as sleep, exercise and age, we are constantly exposed to weather conditions on a daily basis. We decided that since weather had an impact on everyone in a given area uniformly (with big or small effects) that we needed to conduct some literature into finding what effects do weather conditions really have on our mood.

2-2-3 Influence of weather on our mood

Denissen et. al examined the effects of six weather parameters (temperature, wind power, sunlight, precipitation, air pressure, and photoperiod) on mood (positive affect, negative affect, and tiredness) [27]. They interviewed 1233 participants and collected data through an online diary which focused on the determinants of individual daily well-being of the participants between July 2005 and February 2007. The results showed that none of the six weather parameters had any significant main effects of on positive mood. The association between sunlight and tiredness was significant. The less sunlight people were exposed to, the more they exhibited depression-like symptoms. Vitamin D3, which is produced in skin exposed to the hormone of sunlight, is known to change serotonin levels in the brain, which could account for changes in mood.

Goldstein explored relationships between mood and weather variables (temperature, humidity, barometric pressure, clearness, temperature deviation from normal, wind speed) by interviewing 22 students in an introductory psychology course at a community college [28]. He asked them to rate the concept "How I Feel Now" on scales representing the semantic differential factors of evaluation, potency, and activity. Ratings were to be made on 11 consecutive Tuesdays and Thursdays, beginning in late October 1971. Students were requested to make their ratings at 2:00 PM, regardless of where they might be on the campus at the time. Positive mood was found be to related to low humidity, high barometric pressure and days being cooler than normal. The students also reports level of high activity on days with low humidity, high barometric pressure and low wind speed.

Harmatz et. al examined the effect of season of the year on depression and other moods in [29]. Their study employed a longitudinal design and a large sample drawn from a normal population (163 males, 159 females). The results demonstrated strong seasonal effects with depression rates measured through Beck Depression Inventory (BDI) being the highest in winter and lowest in summer. The ratings on the scales of hostility, anger, irritability, and anxiety also showed very strong seasonal effects.

Howarth et. al [30] conducted a study between November 10 1980 and December 10 1980 with 24 male subjects to take a multidimensional approach in understanding the relationship between mood and weather. The mood variables were—concentration, cooperation, anxiety, potency, aggression, depression, sleepiness, scepticism, control and optimism. The weather variables included—hours of sunshine, precipitation, temperature, wind direction, wind velocity, humidity, change in barometric pressure and absolute barometric pressure. The subjects were given 11 copies of the Howarth Multiple Adjective Checklist (HMACL-3) which they were asked to fill out every day prior to their evening meal for 11 consecutive days. The results from this study showed that humidity, temperature and hours of sunshine had the greatest effect on the subjects' mood. High levels of humidity lowered scores on concentration while increasing reports of sleepiness. Rising temperature lowered anxiety and skepticism mood scores. One of the shortcomings of this study was that the test duration was very short and thus a seasonal impact on subjects could not be studied. It was useful however to see the study incorporate a multidimensional approach to both mood and weather and understanding their correlations.

Hannak et. al performed a Twitter-specific sentiment extraction methodology to explore patterns of sentiment present in a corpus of over 1.5 billion tweets to understand the effect of the weather and time on aggregated sentiment [31]. Using machine learning techniques on the Twitter corpus correlated with the weather at the time and location of the tweets, they found that aggregate sentiment followed distinct climate, temporal and seasonal patterns. The sentiment analysis was performed by filtering tweets with positive and negative emoticons. The sentiment inference accuracy was demonstrated to have a high correlation with a subset of tweets that were also rated by humans using Amazon Mechanical Turk (each human participant paid $0.10 to rate 10 tweets). The trends observed that as the humidity increased, the predicted sentiment score decreased for all values of temperature, with the decrease more pronounced at higher temperatures (shown in Figure 2-7).

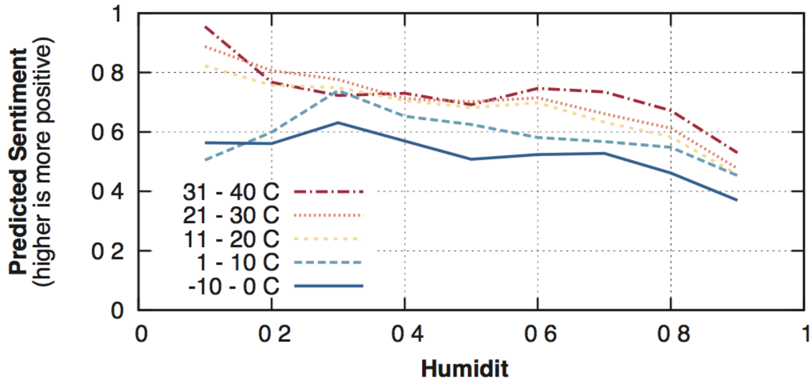


Figure 2-7: Partial dependence plot of predicted sentiment score showing that as humidity increases the predicted sentiment score decreases (with a more pronounced effect at higher temperatures) [31]

Lastly, Sanders et. al conducted a study on 30 psychology students (14 male, 16 female) to understand the relationships between weather and mood [32]. Across the 25 day testing period, the students were asked to complete of the Mood Adjective Checklist developed by Nowlis [33] before every class. After performing a multivariate analysis of weather variables and mood measures, a significant correlation between mood and weather was found. Specifically, decreased vigor, social affection and elation were correlated with higher relative humidity. The authors suggested that the pattern of inverse relationships could be characterized by feelings of diminished physical energy, reduced interest in social interactions and a overall flattened affect amongst the students. As we have seen in prior studies as well, humidity and temperature seem to be having a stronger impact on users than the other weather variables. One of the risks of this type of analysis however is that the authors provide a conclusion based on certain observed correlations and this does not necessarily imply causation.

As we can see from literature, identifying the relationships between human emotion and music is extremely complex, especially because the results can so easily be dismissed as being so subjective. Imagine a music recommender system that decides the user's mood as angry and recommends dark and heavy music when in fact the user is not angry at all. This would create a very bad user experience for the user because it could in turn make the user doubt not only such a system, but also lose trust in such a system. The literature papers studied in this section informs the reader of the differences between emotion and mood and how there are several factors that can affect it. The reader is also presented with a collection of papers that show the influences (or rather lack of in some cases) of weather and seasonal conditions on users. We think that in future, there are research opportunities in the areas of implicit emotion detection through the use of sensors that are embedded in our bodies constantly monitoring our vitals thereby giving a better understanding and quantifiable metrics of how we feel and emote when listening to music.

2-3 Music retrieval techniques

Audio files stored in the public or private domain usually contain a lot of metadata that can be used for information retrieval purposes. This metadata may contain information such as genre, favorite count, artist, tags, creation date amongst others. When recommender systems query these databases to retrieve relevant music or artists, the right combination of querying mechanisms plays an important role in the recommendation process because it is from this subset of retrieved songs that further song analysis is performed. Often times when new music or artists are added to an existing database, there is a lot of missing information because the new item might not have been played yet.

Eck et. al define social music tags as user-generated keywords assigned to music resources (eg. songs, playlists, artists) that have become an important retrieval component for recommender systems, allowing users to query music based on terms such as *lounge or workout* [34]. They propose an auto-tagging method for predicting such social tags directly from audio files by looking at the song's music features. Their main motivation is to auto-tag untagged music thus allowing previously unheard music to become retrievable by the social recommender system thereby overcoming the cold-start from generally associated with collaborative filtering.

They evaluated their system with a set of 60 social music tags scraped from Last.fm. However, their algorithm requires the raw signal-level features of all new songs which can become computationally challenging if the recommendations are to happen in real-time.

Traditionally, content-based and collaborative filtering systems look at things such as audio features, user playback history, user-item rating preferences amongst others. Knees and Schedl present a comprehensive survey of music similarity and recommendations from contextual music data [35]. In their paper they review two forms of music similarity estimation—text-based and co-occurrence based. Basic concepts of text-based approaches includes Latent Semantic Analysis (LSA), similarity calculations (eg. cosine/euclidean distance), term-frequency inverse-document-frequency (TF-IDF) schemes amongst others. Other text-based approaches include web-terms, collaborative and social tags [36], lyrics and comments analysis [37]. Co-occurrence based techniques includes finding similar artists and songs in playlists (eg. user-generated playlists), page counts, using web-services for artist recommendations, peer-to-peer and social networks.

All the above approaches break from the traditional form of music information retrieval where we would analyze either the music content, the user's listening history or look at their listening preferences in order to recommend similar yet novel music. These new contextual techniques open up future possibilities of recommender systems in ways that would allow them to provide more diversity and novelty in the recommendations without having to rely so much on past user data.

2-4 Knowledge and technology gaps

Coming to the end of this section, we see that while there is already a lot of existing work done in this area of research, there is room for improvement. The first technology gap that we have identified is that most of the context-aware music recommender systems presented require some form of offline training before the system can begin to be used. There is also an over-reliance on user preferences and past history. This dependence on past history and user preferences in our belief can limit the user's chances of exploring new music that is outside of their current listening activity. We also did not see any of the systems work with live streaming music and often times dealt with offline music databases. In practice, a solid design framework should be able to work in the real-world of applications where the music recommendation takes place in real-time. While some authors did speak about contextual tag-based queries, we did not see any retrieval mechanisms that discussed the critical link between understanding the user's current situation and using this contextual piece of information to query relevant music. Finally, in terms of user evaluation, we only saw a handful of papers where the proposed recommender systems were tested with real users and even when they were, they was no focus on the user experience of the test applications. As we all know, user experience plays a critical role in evaluating music recommendations and it is imperative to design a system that not only answers our research questions, but also takes into account the impact on the perceived emotion of people using our system. With these knowledge gaps identified, in the next chapter we outline our guiding design principles which evolved from this literature study.

Chapter 3

The Five Design Principles

The identification of the knowledge and technology gaps in Section 2-4 help us formulate a of set of guiding principles for our design. They act as cornerstones around which our design, implementation and evaluation will be based on and help answer our research questions put forth in Section 1-5. Furthermore, they will differentiate our system from other work and guide us to ensure that it is innovative and technologically relevant yet forward-thinking for a end-user point of view.

Design for a streaming future

As we read earlier in the introduction Section 1-1-2, more than 75% of people stream their music from online music services. Music services such as Spotify, Apple Music, Rdio and others have made music library management easy for users with multiple devices and cross-platform support, offline storage capabilities and huge music libraries. We expect this trend to continue in future. Thus, it is very important to identify this growing trend and focus our efforts on coming up with a system that is designed for streaming music.

Context-awareness

Today there are billions of sensors in spaces around us—buildings, airports, office spaces, wearable objects, cars and factories, easily outpacing the total human population on our planet [2]. Most of them sensors work together with very tiny computing devices (microcontrollers) to ‘sense‘ environmental features such as air pressure, light intensity, acceleration, speed and location. Smartphones and tablets are powerful computers that contain a number of these sensors enabling new applications in a wide variety of fields [38]. In such a climate, it is extremely important to acknowledge and incorporate sensors and other ways to identify user context to help solve the current challenges facing music recommender systems and design a system for the future.

Music discovery

Any music recommender system that relies solely on past user history will end up in the user listening to the same type of music that they have always listened to in the past, thus ending up in the “filter bubble". Our focus in this thesis is not to try and re-create the users existing playlist, rather, to generate novel and serendipitous music recommendations for users and help them expand their music horizon.

Learning-free

This principle is very closely related to our focus on music discovery. In order to elicit the desired surprise and delight factor from users, we should not rely on learning about the user’s past listening history as the only way to model all future music recommendations. We believe this non-learning characteristic of the system to be a radical departure from current music recommender systems. By inferring the user’s music preferences based on collection-wide user experiences for a context, we think that such a system will achieve a level of personalization that is ideal for music recommender systems without the need to learn everything about the user’s past listening history. We hypothesize however that placing this learning-free constraint for our system makes it more challenging to generate serendipitous music recommendations but extremely important for users.

User experience

Music is enjoyed and listened to by people and not machines. For this reason, any proposed system that aims to generate novel and serendipitous music recommendations must involve users. The focus on user experience is the most important guiding principle and it should guide everything from the internal workings of our proposed system all the way to the user interface presented to users. It should be simple, intuitive yet functional and evince a feeling of joy and excitement for users.

Having put forth these principles, we envision an application that is extremely simple and easy to use, expands the user’s music horizons and helps them discover new music that they would never have found on their own, thus making the music discovery process less tedious and cumbersome.

Chapter 4

ImpliciTunes Design

With the guiding principles from Chapter 3 as our foundation, this chapter talks about how we went about designing a system to provide novel music recommendations by incorporating contextual user information from sensors. To keep the reader oriented as we discuss the various stages of the design process, Figure 4-1 shows a timeline of major events during the thesis.

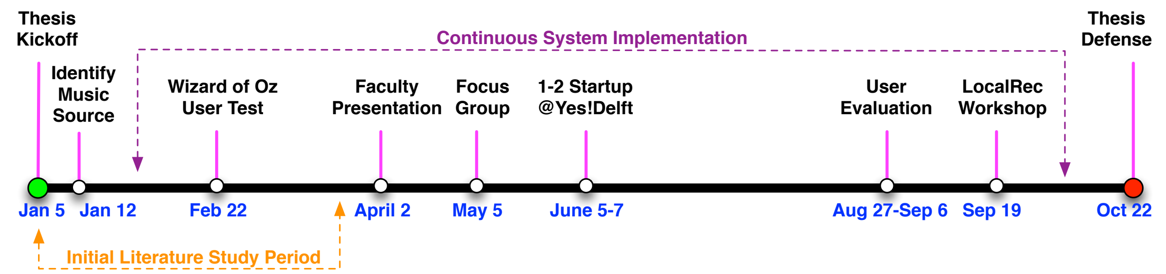


Figure 4-1: ImpliciTunes design timeline

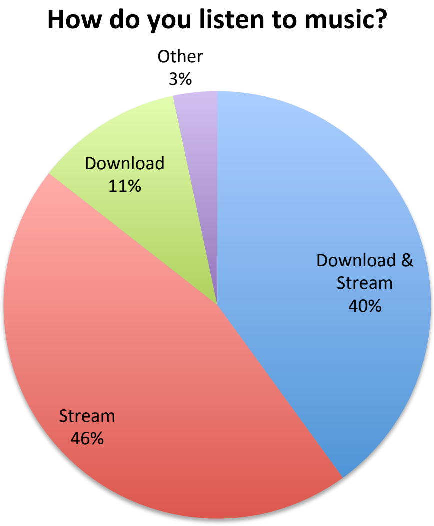
This chapter is divided into six sections—how we plan on adhering to each of the five design principles followed by a conclusions section to reorganize our thoughts before discussing the system implementation details in Chapter 5.

4-1 Design Principle #1: Focus on streaming music

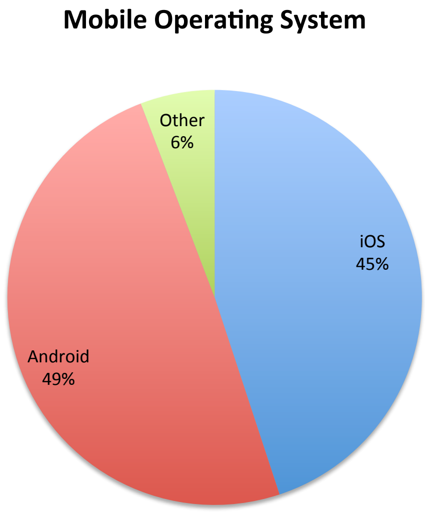
This section refers to the “Identifying Music Souce" item on the design timeline shown in Figure 4-1. As per the statistics released by Nielsen in September 2015 [3], almost 75% of people in the United States listened to music through streaming services such as Spotify, Apple Music, iTunes, satellite radio stations. It is important for us to identify a suitable streaming music source that is developer friendly so that we can build our system on top of it.

4-1-1 Identifying a streaming music source

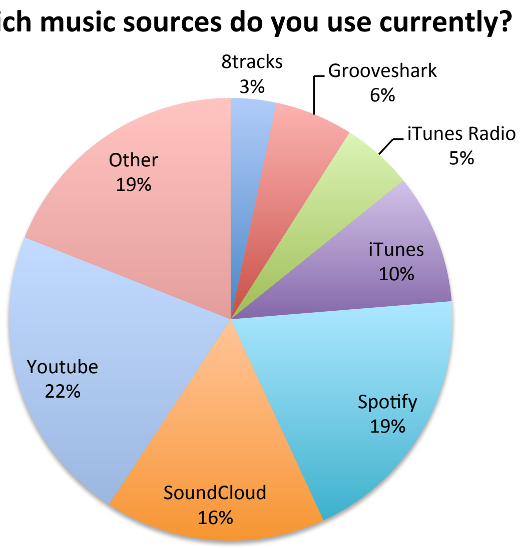
Since Nielsen's study was released much later than when this thesis started, we had conducted our own survey to get more insights about how people listened to music. In addition to identifying how users were listening to music, we also wanted to find out which software development platform to choose and most importantly, which service to choose as our music source. We did not want to design a system that would have a small and controlled music database because that would not be a very realistic scenario—most commercial music streaming services offer their users millions of songs to choose from. Appendix F shows the survey form that 71 participants on Facebook between the ages of 18-45 completed and the results are shown in Figure 4-2.



(a) User Music Listening Behavior



(b) User Smartphone Software Platform



(c) Primary Music Services Used

Figure 4-2: Online music survey results

Based on the survey results shown in Figure 4-2, the following conclusions were drawn with respect to our design:

- Almost 80% of participants stream music from online music sources.
- Of the three most popular music services used, we decided to use SoundCloud[1] because they have a public HTTP API that allows developers to stream any song from their 125 million song database (as long as the song's uploader has made it available for streaming). Even though Spotify[2] turned out to be the most popular music streaming source, in order to write applications for Spotify, developers and all users need to be 'Premium' subscribers which at the time we thought could limit our test group. We did not want to use YouTube[3] as a potential music source because of the lack of publicly available APIs for audio only streaming.
- The results show a close tie between Android[4] and iOS[5]. Our primary smartphone in possession is an iPhone 5S and this made iOS a convenient and sensible choice as the development platform.

The intentions and objective of this short survey were to find out which music sources we should pick as our music source and this turned out to be SoundCloud for the reasons stated above.

4-1-2 SoundCloud as a streaming music source

With more than 125 million in their database, SoundCloud is the world's largest online music service available for users. Exploring new music on SoundCloud is quite challenging however for a new user. A few ways in which users discover new music are:

- Following new artists
- Listening to the 'Stream' section on SoundCloud for trending music and/or like genres
- Re-posting songs
- Sharing songs to a social media platform such as Facebook[6] and/or Twitter[7]
- Finding songs from SoundCloud's recommended songs - not available on the mobile application

All the above approaches requires the user to have been using the system for quite some time before the system can start providing relevant music for the user because there is no prior user history. Their public API allows developers to start streaming music from their database without any user credentials and this is very useful for us. Table 4-1 below shows the query filters for retrieving songs through their 'tracks' API endpoint.

[1] https://developers.soundcloud.com
[2] https://developer.spotify.com
[3] https://www.youtube.com/yt/dev/
[4] https://developer.android.com
[5] https://developer.apple.com
[6] https://www.facebook.com
[7] https://www.twitter.com

Parameter	Type	Description
q	string	string to search for
tags	list	comma separated list of tags
filter	enumeration	(all,public,private)
license	enumeration	filter on license
bpm[from]	number	tracks with at least this bpm value
bpm[to]	number	tracks with at most this bpm value
duration[from]	number	tracks with at least this duration (in ms)
duration[to]	number	tracks with at most this duration (in ms)
created_at[from]	date	tracks created at this date or later
created_at[to]	date	tracks created at this date or earlier
ids	list	comma separated list of track ids to filter on
genres	list	comma separated list of genres
types	enumeration	comma separated list of types

Table 4-1: SoundCloud tracks query filters

From Table 4-1 we can see that for contextual music retrieval, the most useful query filters are the 'q' and 'tags' filter fields. The 'tags' filter performs a Boolean AND operation on all tags. This means that if we query with a filter tag list of *'happy, cheerful,funny'*, the songs that would be returned are the ones that have been tagged with all three tags. The 'bpm' field unfortunately is user-filled which meant that the values were not always present in the song's metadata. Having identified our streaming source in SoundCloud, we can now move on to the next section where we present how we designed our system to be context-aware.

4-2 Design Principle #2: Being context-aware

In order to design a context-aware music recommender system, we needed to identify what types of contextual information we wanted to use in our system. Before we decided on the contextual information, we first needed to decide on what types of contexts did we want our system to recognize. These contexts would help in recommending music so we needed to find out what were some common situations where users listened to music. What we did here is that found out what our users experienced on a daily basis and then worked our way back to the types of sensors that we would need in order to infer these situations. Figure 2-5 shows a list that Juslin et. al found to be the most common situations during which user's listened to music. From that list, we selected the ones that would cover a day in the life of users and decided to focus on recommending contextual music for the following situations:

1. 'Waking Up'
2. 'Commuting'
3. 'Working'
4. 'Relaxing'
5. 'Exercising'
6. 'Housework'
7. 'Sleeping'

We decided that the situations in the above list would be used to generate situational-based playlists for users.

In Section 2-2-3, we investigated the influence of weather conditions on our mood and decided that our system should also recommend a 'Mood' playlist that would contain songs based on the effect current weather conditions have on our mood. The 'Mood' playlist would be termed as an atmospheric-based playlist.

Thus, we identified the contexts that we wanted to recommend music for—either one of the seven situational-based playlists or the atmospheric-based playlist. In order to achieve this, we needed to infer the user's context and this is where the use of sensors is important.

We did not want users to have to wear additional sensors on their body as this is an inconvenience and we did not see much value in adding more hardware complexities. We conducted some initial prototyping work in considering the feasibility of using a pulse sensor as a means to get the user's heart beat, which could then have been used as a sensory input to our recommender system. This was not feasible however and the reasons for this are explained in more detail in the Appendix E. Instead, we decided to design our system for existing iOS-compatible smartphones. As we know, ever since the release of the first iPhone, numerous sensors have become readily available to developers to take advantage of. This is of course true not just with iOS, but also on other platforms like Android and Windows Mobile.

Sensors	**4S**	**5**	**5S**	**6**	**6+**
Proximity	x	x	x	x	x
Ambient Light	x	x	x	x	x
Accelerometer	x	x	x	x	x
Gyroscope	x	x	x	x	x
Magnetometer	x	x	x	x	x
Dedicated Motion Sensor			x	x	x
GPS	x	x	x	x	x
Barometer				x	x
Microphone	x	x	x	x	x
Speakers	x	x	x	x	x
Fingerprint			x	x	x

Table 4-2: Physical sensors present in iPhones 4S and onwards

From the sensor choices in Table 4-2 it is clear that most of the devices have all the sensors that we could possibly consider using in our system. Of the three major differentiating factors, the dedicated motion sensor was the most important choice to consider because it would provide us with pre-classified activity states such as *stationary, walking and running*. This would make our task of using activity in our system a lot simpler and also provide consistent results across all supported devices. We could not find an use case for the fingerprint sensor. Finally, the barometer was introduced on iPhone 6 and iPhone 6+ smartphones and can be really useful for detecting activities such as running up and down stairs, detecting the device's elevation and correlate that with temperature to infer certain weather parameters amongst others. However, we thought that adding a barometer to our system would limit the compatibility of our system only to users with iPhones 6 and 6+ and thus, chose not to include it in our design. In the end, we decided that given the importance of the dedicated motion sensor, our system would support iPhone 5S, 6 and 6+ for the purposes of this thesis. Future versions of the application would integrate support for iPhones 4S, 5 and 5C by developing activity classifiers using the raw accelerometer and gyroscopes sensor values and also integrate the barometer.

Apart from physical sensors, logical sensors can also be used to retrieve useful information from the web and locally on the smartphone such as:

- Date, time and season information
- Weather information
- Location categories
- User calendar

For our system, we decided to use date, time and season information locally from the smartphone and weather and location category information from external web sources. These inputs would give us critical information about the user's context and help us provide relevant music recommendations for all the eight different playlists that we decided to focus on. We decided to forgo using the user's calendar since that required the user to create explicit appointments in their calendar and we did not want to place any such restrictions on our users. With our sensor choices made, in the next section we talk about how our design plans to help users discover novel music for different contexts.

4-3 Design Principle #3: Music discovery

Our goals for the music discovery part of the system is to help users discover new music through the means of a learning-free contextual music recommender system. This means that we cannot simply look at what the user has listened to in the past or base our recommendations on items they have liked. Additionally, we do not want to re-create the user's own music, rather, we want to provide a continuous stream of new songs through contextual playlists throughout their day. The system should allow the user to discover new music that they would never have otherwise listened to if the recommendations were only based on their past listening history.

4-3-1 Ask users what they dislike

We want to help users expand their music horizons and for that reason, in addition to asking what genres of music they enjoy listening to, more importantly, we also want our system to be aware of what music genres that are absolute deal breakers for them. We think this is quite a radical departure from current music recommender systems that only focus on the type of music users like and not on what they do not like consistently. This in essence narrows down the user's listening choices dramatically and keeps them from discovering new music that they would never have listened to otherwise.

4-3-2 Contextual music retrieval from SoundCloud

In order to retrieve contextual music from SoundCloud, we investigated SoundCloud's public API to see what query fields could be of use. We referred to Knees and Schedl's survey paper [35] where they discuss music similarity and recommendations from music context. They identified a number of possible sources of contextual data for music recommendation besides the usual content-based and collaborative filtering approaches and they are summarized their beliefs very succintly in Figure 4-3, each with its advantages and shortcomings.

	Tags	Web-terms	Lyrics	Co-occ.
Source	web service	web pages	lyrics portals	web (search)
Community Req.	yes	depends	no	no
Level	artist, track	artist	track (artist)	artist
Feature Dim.	moderate	very high	possibly high	item×item
Specific Bias	community	low	none	low
Potential Noise	moderate	high	low	high

	Microblogs	Playlists	P2P	Ratings
Source	API, feeds	radio, CDs, web	shared folders	users
Community Req.	yes	depends	yes	yes
Level	artist (track)	artist, track	artist, track	all
Feature Dim.	item×item	item×item	item×item	user×item
Specific Bias	community	low	community	community
Potential Noise	high	low	high	yes

Figure 4-3: Overview of Different Context-Based Sources[35]

Looking at Figure 4-3, we see that of all the music context data sources presented, the ones available for retrieval from SoundCloud are tags, playlists and ratings (e.g. user like counts, play counts). The way tags and playlists work on SoundCloud is that the artist or song creator uploads their song to SoundCloud and fills in all the required metadata—including information such as title, tags, genre and others. These user-generated tags can describe anything—situation, artist, mood, location, event or others. As more people tag the music they upload, the likelihood of songs with similar tags increases. What this means is that multiple people can essentially tag different songs with the same tag or vice-versa.

We want to exploit this communal behaviour of music listening across different contexts and hence we will choose to focus on generating contextual tags to retrieve music from SoundCloud. This step is crucial because it means that users can essentially start listening to contextual music recommendation immediately without having to wait for the system to learn their music taste and listening history. With over 125 million songs in the SoundCloud database, a lot of the songs are already richly annotated with tags. Tags of a track that are related to the context provide us with evidence that listeners generally associate the track with that context. Another reason to use tags for the music retrieval portion of our system is that it will introduce users to completely different songs, songs that they might not have every heard before. Given the time frame required to develop our system (9 months), we left the options to using user-generated playlists as a means to retrieve contextual songs as very likely future extension of our system along with similar music discoveries through web sources, microblogs and co-occurrences.

4-3-3 Looking at the song's signal content

In addition to looking at the song metadata to query relevant and contextual songs from SoundCloud, we also investigated the possibility of using a third-party solution to extract more information from the song's audio signal content. The two alternatives were Essentia[8] and EchoNest[9]; both solutions provide low-level signal information for a given piece of audio which can then be used for higher level recommendation re-ranking or filtering of songs depending on the contextual relevancy.

Essentia is an open-source C++ library used for audio signal analysis and contains an extensive set of algorithms. The library comes packaged as a binary that can be installed locally on a device and an algorithm of choice can be used to extract a wide range of music descriptors. The only downside of the software is that they do not have an API endpoint that can accept external URL links to perform the content analysis since everything needs to be done locally. Albeit a very good open-source solution, this was unsuitable for the needs of our application as our music source is stored on the SoundCloud servers. If we would try to incorporate Essentia into our design we would need to write our own web service API that would be able to handle SoundCloud streaming URLs for song analysis, download the song over the internet and then execute the content-based analysis operations. We estimated the time and effort for this operation to be outside the scope of this thesis as a lot of implementation time would be spent on hosting a server that would expose public APIs for developers to get access to Essentia and we did not have the resources for that given the time frame of 9-10 months. For this reason we decided not use Essentia.

EchoNest on the other hand offers a similar degree of song-content analysis and have an open web API endpoint for developers. Some of the song parameters that can be retrieved include high-level features such as energy (valence), arousal, tempo, and danceability [10] and are explained further in [39]. The key advantage of using EchoNest is that it can analyze remote tracks based on URLs. This is perfect for our application because all the music is stored on SoundCloud and we do not need to maintain a server to host the algorithms as

[8] http://essentia.upf.edu/home
[9] http://developer.echonest.com
[10] http://developer.echonest.com/docs/v4/song.html#profile

in the case of Essentia. The only downside is that there are restrictions as to how many API requests an application can make per minute. If the application exceeds that limit, all future API requests get returned with an error until the minute passes. This could potentially cause user-facing performance and/or usability issues, but because of its ability to analyze remote song URLs, we select EchoNest as our content-based analysis service for all songs post retrieval from SoundCloud. We will have multiple playlists recommended to the user at any given situation (from the choice of the eight playlists listed above) to give the user enough choice of music. To circumvent the API usage limitations of EchoNest and any foreseeable user performance issues, we will perform content-based analysis only if there are at least two playlists that the user can choose from. We will see the implementation details explained in more detail in the next chapter.

4-4 Design Principle #4: Learning-free—recommend music for the present, not the past

One of the important design principles that we established was that our system would not try to learn from the user's past listening history. This makes the system design challenging yet at the same time novel because we cannot use any machine learning techniques to 'improve' the music recommendations over time by relying on the user's past listening history. Since the recommendations will be based on their current context, they will always be contextual thereby giving a level of serendipity often absent from current music recommender systems. The important task for the sensors will be to infer the user's context as accurately as possible. We could use machine learning algorithms to 'learn' about the user's context and see if there is any associated pattern in their daily lives. This could be useful for example if we know that a user arrives at home from work during weekdays between 18:00 and 19:00, this pattern could be useful in predicting the user's context within a few days of using the system. After this initial training period, the system might not need to rely so heavily on sensors and thereby conserve battery life.

Music discovery on the other hand does not necessarily happen in the same regularized manner as our daily lives wherein we don't discover new music every day. At the same time, user expectation during music discovery is that every new song recommended should be good. With that premise, it becomes extremely important for the user's contextual information to lean more towards a more reactive system than one that needs some time to learn about their whereabouts and then start recommending the right music. We concur with Zhang et. al's conclusions in [7] where in order to achieve a better level of user satisfaction, the accuracy of our recommendations might be reduced. We want to avoid our users from being trapped inside this "filter bubble" of listening to the same types of music that we are already listening to and for this reason, we decided not to incorporate any learning behaviour to our system and leave it as a future extension to this topic in select cases such as predicting location information. We placed the following requirements for whichever data modeling technique we eventually selected:

- Lightweight to be able to run efficiently on smartphones
- Able to work immediately as the user starts using the system without any training period

- Handle multi-modal sensor data
- Model the vagueness and uncertainties in the way different users describe their context for the exact same external conditions

The above list is crucial for the relevance and practicality of our system in a real-world environment. This list was formulated up by the author from his past experience of developing real-world software systems and understanding the limitations of technologies outside of the research realm. We do not want to design a system that does not meet all of the above criteria because otherwise, such a system will have a extremely negative impact on the overall user experience. Looking at these above four requirements, we decided to use Fuzzy logic control mechanisms to 'make sense' of the multiple sensor data streams and infer the user's context. Fuzzy logic was first introduced to the engineering world in 1965 by Lotfi Zadeh in his seminal paper titled 'Fuzzy Sets' [40]. It was a radical idea at the time because it dared to allow vagueness and imprecision in a field that was obsessed with precision and exactness. Since then Fuzzy logic has been applied to numerous practical applications in control systems by commercial companies such as Hitachi and Fuji for appliances such as automatic train control, washing machines and others [41]. As listed in the related works chapter, Fuzzy logic has also been recently making its way into the field of recommender systems.

Drawing from the field of control systems where fuzzy logic control systems need to infer an output from a set of inputs which could often be vague and yet be describable in human linguistic terms, we can use a similar approach for our system. Babuska describes fuzzy logic systems as being a suitable framework for representing qualitative knowledge that is provided either by human experts or acquired from data [42]. Such a rule-based system is then able to generate a compact, elegant and highly transparent qualitative summary of a large amount of possibly high-dimensional data. Our system has similar requirements where we need estimate the user's most likely contexts from a wide range of sensory inputs and the best way to create decision points would be generate plausible outputs. Fuzzy logic gives the design a level of flexibility where the same set of sensory inputs could generate multiple outputs depending on how the fuzzy rules are designed. Additionally, since the system will be described in qualitative terms, it is highly transparent thus making it very easily modifiable and due to this, we will know exactly how our system works unlike other techniques which are often treated as black boxes. Motivated by our non-learning design concept, Fuzzy logic makes it possible to translate user-supplied human language rules into mathematical values that can be used for making decisions and this makes the system logic easily understandable. Given the computational challenges of fusing multi-modal sensor data, fuzzy logic provides an extremely light-weight and efficient technique. The implementation details of contextual inference will be discussed in further detail in the next chapter.

4-5 Design Principle #5: Importance of user experience

Our last design principle also happens to be the most important—user experience. It is easy to design a complicated yet technically advanced system, yet very challenging to do both. Our goal for this thesis is to recommend novel music based on contextual sensor data to users. We wanted to start testing our design goals and propositions from a very early stage in the thesis so that whatever we built was relevant and useful for users. In this section we briefly discuss two of the experiments that we conducted with users and received valuable feedback which helped improve our understanding of the system.

4-5-1 Testing user perception of contextual music - Wizard of Oz Test

This sub-section refers to the "Wizard of Oz User Test" item on the design timeline shown in Figure 4-1. Early on in the thesis it was important to check with users if our research goal of recommending novel music based on contextual sensor data was relevant and noticeable to them. More importantly, if given a set of songs to listen to, could users distinguish differences between randomly generated music and that music that is algorithmically generated for a specific context. This was an important test to find out whether users would notice any difference at all between the two. The Wizard of Oz experiment is a very well established and accepted method of evaluating human behavior and performance when using a hypothetical or incomplete technology [43]. We used this approach to test the user perception of contextual music in our own tailored version of the experiment.

Since we were conducting this test very early on in the thesis, we had not yet started any of the real sensor-based implementation or even knew which sensors to use for the music recommendations. What we had completed was that we had identified SoundCloud as our music source and iOS as our development platform. This gave us all the required information to design our own version of the Wizard of Oz test. Since we had not yet defined what sensors we would be using in our system, instead we focused on identifying a particular context for which our users could listen to music and provide feedback. The context statement for which we decided to perform the test for was 'commuting' because we thought it would the easiest and most common and unambigious for users. More specifically, we came up with the following context statement—"Commuting between two places on a nice sunny day while listening to music on my smartphone."

Test design & procedure

It should be mentioned at the onset that in hindsight, we realized that there was a flaw in the test design but is still included in the thesis because we learned some key insights from the process. We recruited 7 users (students between the ages of 23 and 28) who had iOS smartphones to participate in the test. The test was designed such that users would use a test application that would contain two playlists with about 5-6 songs in each and they would have to listen to both and then provide feedback. One of the playlists would contain randomly retrieved songs while the other contained songs retrieved based on contextual information. Users would not know which playlist was which in advance.

The first step was to create a random playlist superset. This was fairly straightforward because we simply retrieved a list of 20 songs from SoundCloud using their public API without specifying any filters. In order to generate a superset for the playlist based on contextual information for 'commuting', we used Last.fm APIs[11] to find tags related to the 'commuting'

[11] http://www.last.fm/api/show/tag.getSimilar

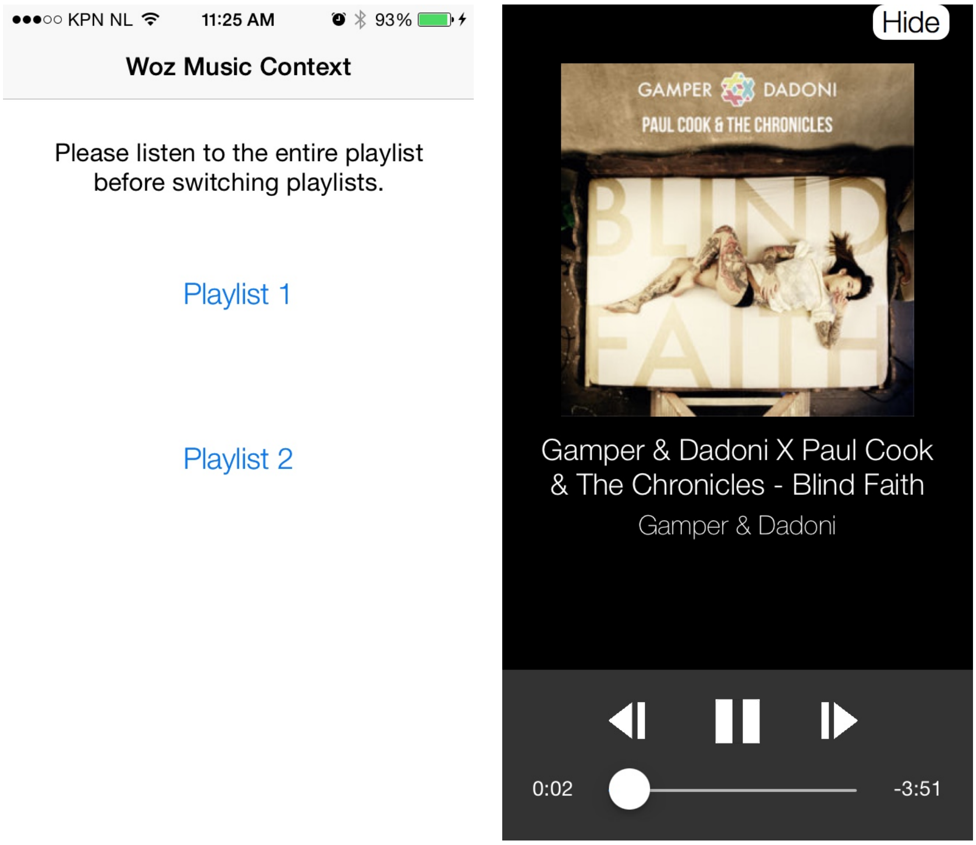
playlist along with the weather information. The seeds tags used to find similar tags were *sunny, roadtrip* and *commuting.* Through this process we settled on using the following tags to query contextual songs from SoundCloud—*sunny, sun, cycling, bike, biking, upbeat, fresh.* Using these tags, we then retrieved the top 20 songs in terms of play count from SoundCloud.

With the music database created, we shuffled both playlists (20 random + 20 contextual) and created a public playlist on SoundCloud [12]. Our 7 users (same testers as above) then listened to all 40 songs from this public playlist and in shuffled order and selected the ones they liked via a survey form. In addition to selecting which songs they liked, they also selected the songs that they had already heard before. At the end of this song selection process, all 7 users had selected from a list of 40 songs (20 random + 20 contextual) the songs they liked and ones that they had already heard before.

To create the two playlists for the test application, we used the user inputs from the song selection stage and selected 6 songs. During this selection process, we ignored any songs that users said that they had already listened to before. Thus, to build the contextual playlist consisting of 6 songs, we started with 20 songs, removed the songs that users had already listened to before and then selected the top-6 most liked from the remaining contextual songs and used these songs in the playlist. For the random playlist, we queried a new random set of 6 songs from SoundCloud.

An iOS application was developed that was pre-loaded with the two playlists and deployed to all 7 users. Users did not know which playlist was which and were asked to listen to both playlists during their regular commute. Figure 4-4 shows the screenshots for the test application. At the end of the test, all 7 users were asked to complete a short survey for their feedback as shown in Appendix G.

[12] https://soundcloud.com/abhishek-sen-5/sets/woztest



(a) Two playlist choices **(b)** Currently playing song

Figure 4-4: Wizard of Oz test application user interface

Feedback Analysis

We had one confounding variable in our design and that was the fact that when selecting 6 songs for the random playlist, we picked brand new songs from SoundCloud and not the random songs that users listened to during the initial selection process from 40 shuffled songs. This is the reason why our test design was flawed. In hindsight, the random playlist should have been created from a list of the top-6 songs that were new for the user and were retrieved randomly from SoundCloud.

The results from this experiment show that about 72% of the testers preferred listening to the contextual playlist and 92% of testers felt that the contextual playlist matched their activity at the time of the test. Most importantly, the participants could perceive a difference between the two playlists which was the goal of this test. When asked if users could perceive a difference between playlist 1 (random) or playlist 2 (contextual), they responded with:

- *"Playlist 1 was almost nothing I would prefer to listen to unlike playlist 2, which had a music style that I enjoyed"*
- *"The tracks in playlist1 sounded amateuristic. It sounded like demo's of kids who are trying to make beats for the first time. The second playlist was okay to listen to"*

- *"Playlist 2 contained songs I have heard in part one of the test. Playlist 2 contained more calm electronic remixes of original songs. They were also less up beat compared to playlist 1"*
- *"Playlists 2 seemed a bit happier, but not by a landslide"*
- *"2 was more like my music taste"*

The test was supposed to have been performed when the weather was sunny but some participants mentioned that they ran the test when it was cloudy or at night so this could have affected the overall results (in a negative way) because the contextual playlist was curated keeping in mind that the weather was sunny. From some of the comments provided in the survey, it seemed that users preferred to provide their music preferences before starting to receive recommendations because some clearly did not like electronic music. Overall this test validated the idea that context could indeed be used to recommend novel music that users could enjoy listening to. Due to the feedback from users, even though we had a confounding variable in our test where we re-selected the songs for the random playlist, our users told us that they could definitely perceive a difference between the two playlists and that they preferred listening to the contextual playlist.

4-5-2 What do users expect from sensors - Focus Group

This sub-section refers to the "Focus Group" item on the design timeline shown in Figure 4-1. During the fourth month of our thesis, after we had finished with the first version of our overall system architecture, we knew the types of sensors we were going to use in our system. We also knew that we wanted to recommend contextual music for the 8 playlists as mentioned in Section 4-2. We knew how to retrieve contextual music from SoundCloud, analyze them using EchoNest and build playlists for users. What was not clear was how to convey the contextual information that was used to retrive the songs and generate the playlists, if at all. Most music services have the standard user interfaces where users see categories such as genre names, artist groups and user created playlists. However, in our case since our music recommendations are all being generated by sensory data, conveying that contextual information becomes very important for the user experience. To answer this question, we conducted a focus group study with 6 Masters students from different faculties at the Delft University of Technology. The 6 participants were recruited through Facebook and signed up voluntarily for the two-hour focus group session. Appendix H shows the agenda used by the author for the focus group session. To ensure a productive focus group session, we laid out the following key objectives for the meeting:

- What is your experience using music recommender systems?
- If sensors are going to be used to recommend your music, what would you expect from such a system?
- Given a list of playlist recommendations using sensors, which playlist(s) would you select and why?
- How would you convey contextual information to the user to help them select their music?

We did not prime the participants with any prior information besides telling them briefly what the topic was about in the initial welcome e-mail. All the priming took place during the focus group session itself. We collected informed consent forms from all the participants before the meeting started. The meeting was video taped and transcribed once it ended. The main role of author during the focus group was to guide the participants according to the agenda and get as much information from the users as possible to help answer the objectives laid out.

Focus Group Outcome

The focus group lasted for about two hours during which the participants were encouraged to think outside of their head and either say things out loud or write them down on paper. Some of the final concepts that the participants came up with are shown in Figure 4-5.

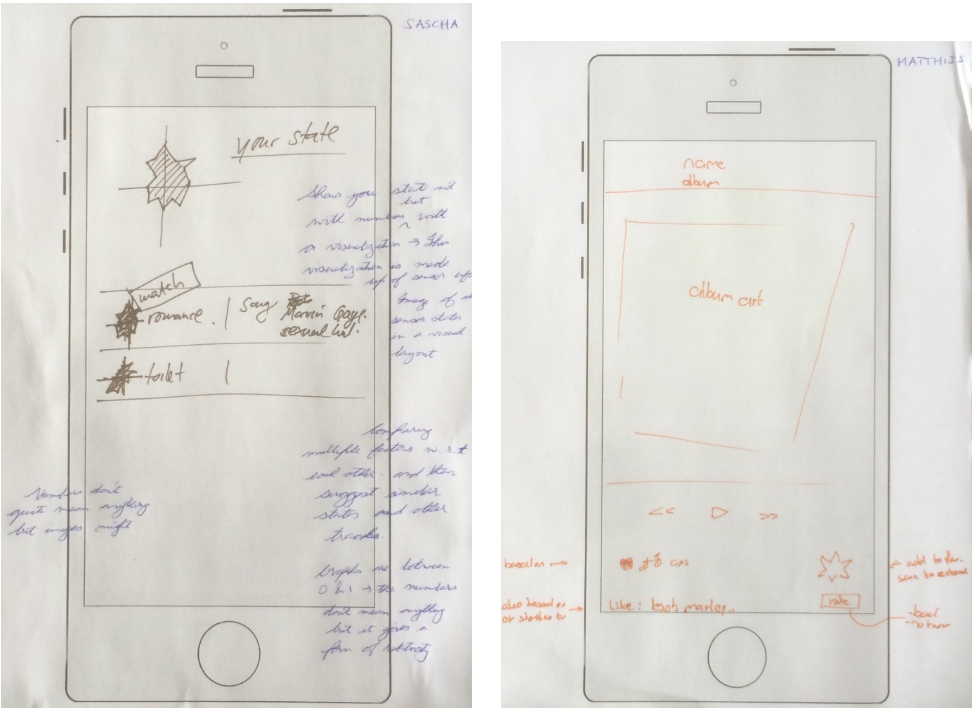


Figure 4-5: Examples of application user interface concepts developed by participants

The feedback gave us some key insights for the system design. All of them described music discovery as a tedious and challenging activity even with all the music available on the web. They described their ideal music recommender system would know which song to play for any given situation and not just based on their past history. One of their main complaints about current music recommender systems was that most systems tend to repeat the same type of songs unless the user has explicitly made a different selection. They were also of the opinion that even though such a system might provide 'bad' recommendations at times, they would simply move on to the next song and continue listening. This insight suggested that our system does not have to infer the user context perfectly and that we could hedge our predictions by providing the user a choice of playlists for the most likely contexts. The group also mentioned that they all had different music tastes and each had their own music preferences for different contexts—–this led us to include a genre preferences block so that in addition to knowing what the user enjoyed listening to of late, more importantly, the system “knows” the kind of music the user really does not enjoy listening.

4-6 Conclusions

By the end of the design process, we had come up with a very feasible plan of all the technologies that we wanted to use in our system and could immediately start with our design implementation. The process of iterating the design process in short feedback cycles proved to be very helpful in validating the design concepts and also ensuring the technical feasibilities for this project. Our design principles from the previous chapter helped guide all of the decisions and choices taken during our design process. In the next chapter we discuss the system’s implementation in greater detail.

Chapter 5

The Proposed System

On the basis of the design developed in the previous chapter, this chapter presents a detailed overview of our final system. There were a number of iterations along the way, which were informed by field testing and the focus group feedback. Each section in this chapter describes a part of the overall system shown in Figure 5-1.

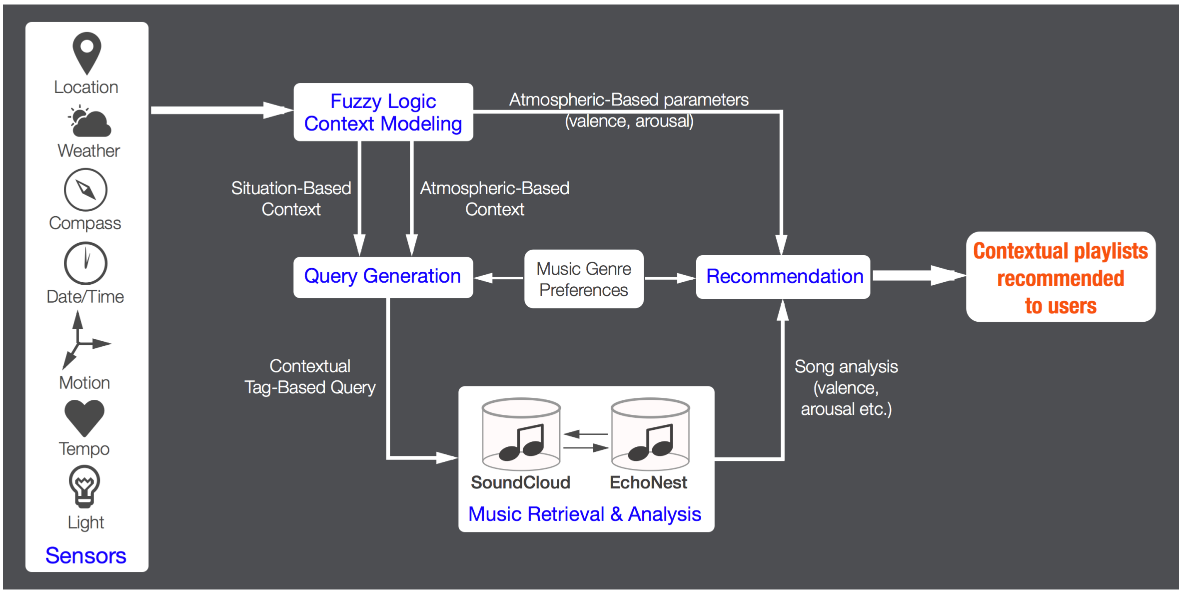


Figure 5-1: ImpliciTunes playlist generation system architecture

Unlike traditional music recommender systems that rely on user history or listening behavior (Figure 1-2), our system does not require any past information about the user, making it ready for use as soon as the user installs the application. The system design is such that it can respond to external sensory events at all times without any explicit user intervention needed. As new sensory data (e.g. location) is received, this information is then pre-processed before being sent to the fuzzy logic context modeling block for processing. The fuzzy logic block determines the most likely contexts that the user is currently in and then for each of them, retrieves songs from SoundCloud. Each of the songs retrieved per context are then analyzed by EchoNest's content-based analysis to determine the audio characteristics for that song. Once all songs have been retrieved from SoundCloud and analyzed by EchoNest, they are then filtered based on the analysis from EchoNest and then organized into contextual playlists and recommended to the user.

The music genre preferences block filters out any songs whose genres the user does not like. By receiving inputs from sensors, all cold-start issues that usually challenge collaborative filtering based systems are avoided. The system design is also very transparent and flexible for modifications, extensions and debugging purposes. New sensors can easily be integrated into the design and new playlists can also be generated by creating additional rules in the fuzzy logic block. The system runs entirely on a smartphone thus enabling the user to discover new music at any time during their day without any special effort on their part. There is a clear demarcation between each of the major sections in our architecture and we go through each one of them in greater detail.

5-1 Sensors

As discussed in the previous chapter, we chose iOS as our development platform and decided to build a mobile application targeted for iPhones 5S, 6 and 6+. Each of these devices contain a number of sensors which we use for context inference. While there are a number of individual sensors present on the smartphone, we decided to focus on the following high-level sensor categories which would be used in our system:

- Place
- Activity
- Weather
- Indoor/Outdoor
- Date/Time

5-1-1 Place

Conventionally, location information is information about a specific geo-location. It involves using GPS and reverse geo-coding the latitude and longitude coordinates to a geographic address in order to identify a location. Location-aware systems typically tend to be very power hungry when run on smartphones due to the nature of GPS. However, we wanted our system to use GPS wisely in order to conserve battery life and at the same time provide useful contextual information to our system. The system currently identifies seven place categories—home, office, library, gym, shops, beach and other. These categories are annotated without users having to explicitly enter any information. We use the Foursquare Venues API[1] to reverse geo-code geographic coordinates to either of the above places categories—except for other. If the coordinates being queried do not match with any of the categories, they are then

[1] https://developer.foursquare.com/overview/venues.html

sent to the Situational Fuzzy Logic Context Model for further processing. Figure 5-2 shows the algorithm on how we detect new location categories from visited locations.

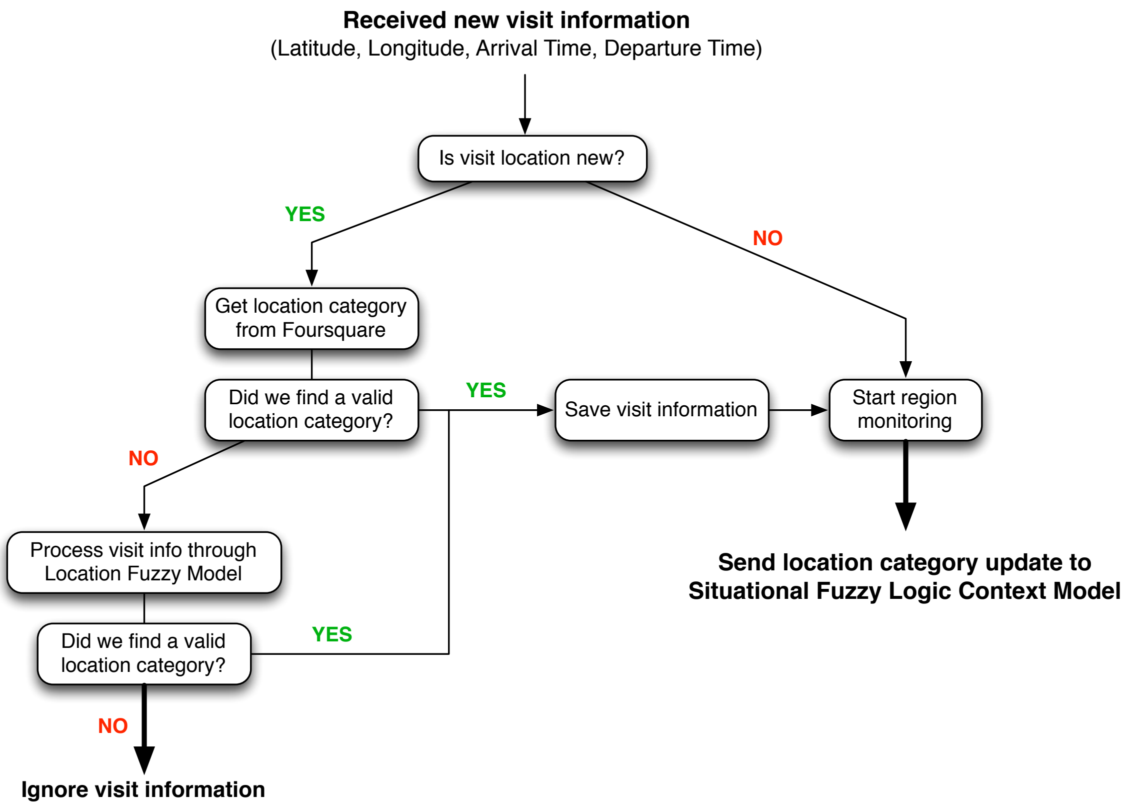


Figure 5-2: Algorithm for detecting new location category

Once region monitoring has been enabled for a place, the system then responds to events when a user enters or leaves the region. Any time the user enters or leaves this region, a place context change event is triggered and sent for processing to the Fuzzy Context Modeling Block. The proposed technique of monitoring places ensures that we do not drain the smartphone's battery by only using the GPS when needed.

5-1-2 Activity

Determining the user's activity states helps us determine what the user is doing in a current location and this is very important in understanding the user's overall context. The activity states that our system identifies are stationary, walking, running and driving. These activity states are provided by the iOS platform which are then pre-filtered to avoid being overtly responsive to any changes in the user's motion. To prevent triggering activity updates every time a user changes their motion for a brief period, we had to ensure that there was indeed an activity state change and we did this by introducing timers in our implementation to add some hysteresis. To accurately distinguish between the stationary and driving state, we perform a quick GPS query to get the user's speed and make a decision accordingly for confirmation.

Figure 5-3 shows the activity state machine transitions and the conditions under which they change. Once a destination state condition has been fulfilled, the activity state changes and the newly updated activity state is then sent to the Situational Fuzzy Logic Context Model for further processing.

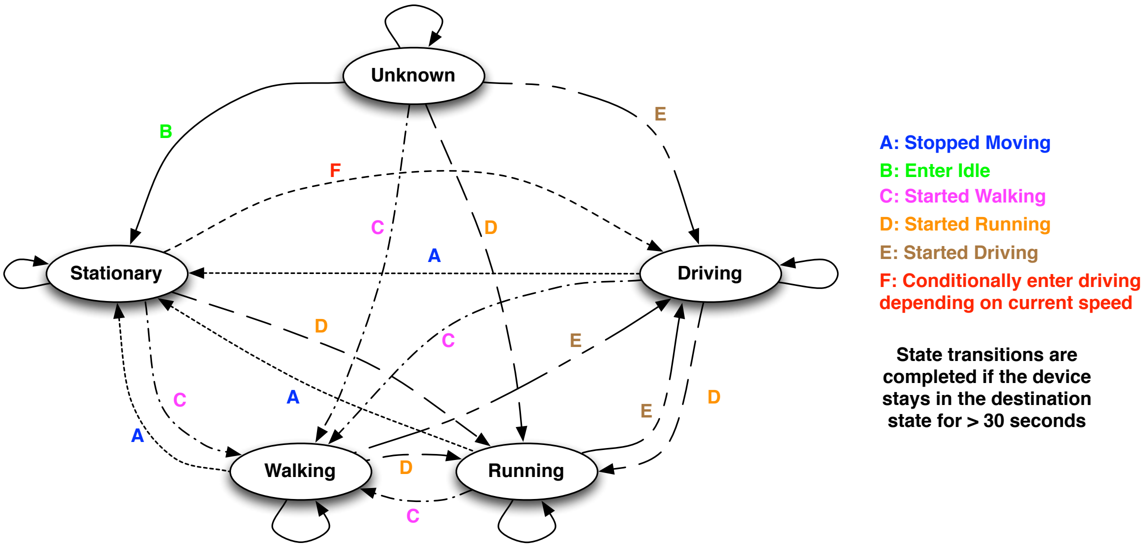


Figure 5-3: Activity state machine with transition descriptions

5-1-3 Weather

Psychology research has shown that there is a certain degree of influence that weather factors have on our moods. Based on the prior works as described in Section 2-2-3, we found that the most influential weather factors to influence a user's mood to be the humidity, temperature, air pressure, sunshine, sunrise/sunset times and cloud cover. There were also some seasonal impacts on user moods but those were not taken into account. We retrieved the weather information using the device's location coordinates from Yahoo's Weather API[2]. Typically, these values are queried once every hour to keep the device up-to-date with any drastic weather changes.

5-1-4 Indoor/Outdoor

In addition to using GPS, determining whether the user is indoor or outdoor gives us a better sense of the situation that the user is in. For example, simply relying on activity to determine whether a user is commuting is not enough unless we also know that they are actually outside and also moving. Zhou et. al proposed a novel technique to determine whether a user carrying a smartphone is indoors or outdoors by compass and light sensors along with cellular radio frequency measurements [44]. Unfortunately on iOS, developers do not have access to the current cellular signal values from the radio towers without jail-breaking the device and so we had to rely on making the decision between indoor and outdoor by using the compass and

[2]https://developer.yahoo.com/weather/

light sensor. In addition, another restriction on iOS APIs is that there is no direct access to the light sensor values. However, we bypassed this limitation by normalizing the screen brightness values between 0 and 1 and use these as brightness or light sensor inputs to the Fuzzy Context Modeling Block. As with the other sensor inputs, we had to add levels of timer hysteresis to ensure that we do not toggle between indoor and outdoor states in case of sudden changes in the sensor data. Figure 5-4 shows an overview of the pre-processing performed on the compass values before they are sent to the Indoor/Outdoor Fuzzy Context Modeling Block for further processing.

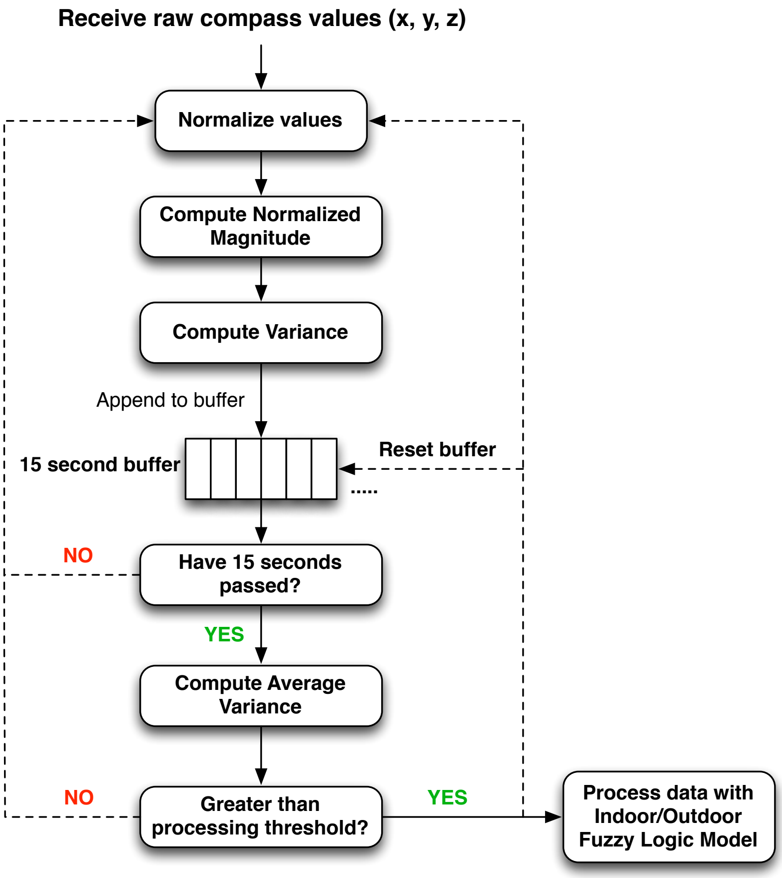


Figure 5-4: Pre-filtering of raw compass data for indoor/outdoor detection logic

For the compass values, once the computed average variance values have been sent for processing to the Indoor/Outdoor Fuzzy Logic Model, the output of that block in turns returns one of three values with varying degrees of membership: Unknown, Indoor and Outdoor. It should be noted that when the device is stationary, compass value variances do not change much unless another electromagnetic device is brought near it causing magnetic variance fluctuations. Similarly, the brightness values are also only valid when the screen is turned on. We ensure that that device does not constantly toggle between indoor and outdoor data due to 'noisy' signal data by waiting for a period of at least 30 seconds before transitioning to the new indoor or outdoor state.

5-1-5 Date/Time

We retrieve the date and time parameters iOS APIs and passed this information to the Fuzzy Logic Context Modeling Block. Some of the information derived from the date and time values were weekday and weekend and location visit arrival and departure times. The date and time inputs help us determine a number of situations such as what the user could possibly doing at 2:00 AM in the morning when he's not at his home on a weekend.

5-2 Fuzzy Logic Context Modeling

Given that all the sensory data could change at any time independently, our system had to be designed in a way as to be responsive enough to handle these environmental changes. Figure 5-1 shows the Fuzzy Context Modeling Block as one but internally there are four fuzzy logic control blocks that help in determining the situational-based and atmospheric-based contextual outputs. Figure 5-5 shows the inner workings of all four fuzzy logic models, including the inputs and outputs for each model.

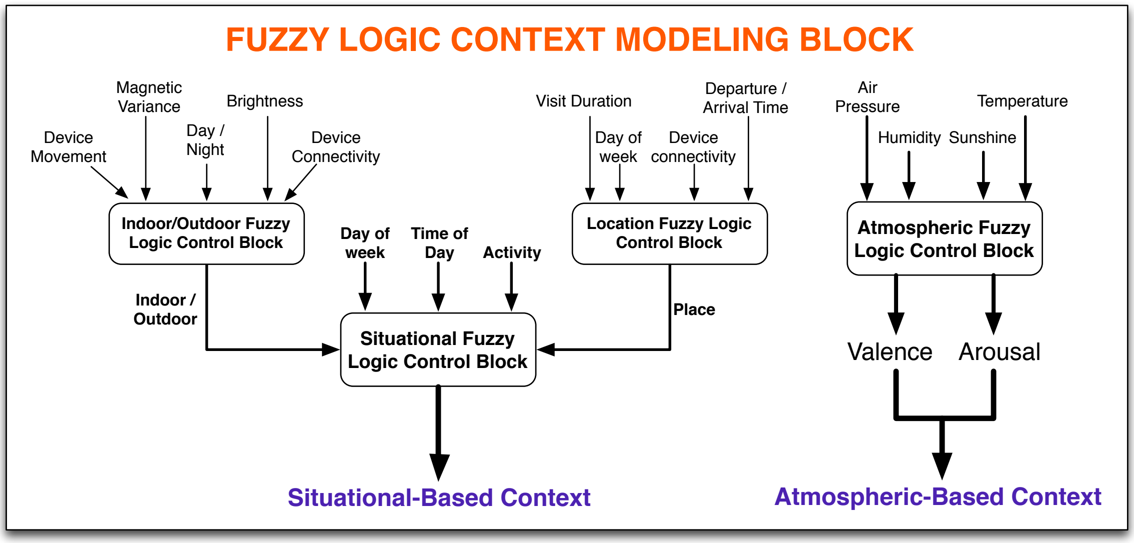


Figure 5-5: Internal workings of Fuzzy Logic Context Modeling Block

All the fuzzy logic control blocks were built using the 'FuzzyLite' library which is an open-source fuzzy logic control library developed by Juan Rada-Vilela [45]. 'FuzzyLite' allows the creation of fuzzy models with an easy to use graphical user interface and contains many fuzzy logic features such as controller options, membership functions, hedge terms amongst others. The rules for each model were created based on the author's own field testing results.

5-2-1 Fuzzy Logic Terminology & Settings used in model designs

In order to explain each of the four fuzzy models in more detail, we need to provide background information about some of the terminology associated with fuzzy logic models. The definitions below describe some of the major concepts of fuzzy sets and fuzzy logic modeling [46].

- Crisp Set - in a crisp set, an element is either a member of the set or not
- Fuzzy Set - an element can partially be a member of a set
- Membership Function - relationship between values of an element and its degree of membership in a set
- Fuzzy Inference System - system that uses fuzzy set theory to map inputs to outputs. Examples include Mamdani [47] & Takagi-Sugeno [48].
- Fuzzy Rules - collection of linguistic statements that allows the fuzzy inference system to make decisions regarding controlling an output
- Fuzzification - process of mapping a set of inputs to values between 0 and 1 (fuzzy values) using a set of membership functions
- Fuzzy combinations - Set of operators such as AND, NOT & OR that are used to make up fuzzy rules
- Defuzzification - the operation needed to convert a fuzzy values into crisp or definite values

Fuzzy Inference System	Mamdani [47]
Conjunction	Minimum
Activation	Minimum
Disjunction	Maximum
Accumulation	Maximum
Defuzzifier	Centroid
Hedges used	Not, very, extremely

Table 5-1: Fuzzy logic settings used for all models

Table 5-1 lists the global settings for all the fuzzy logic models we used in our system. For all the models presented in later sub-sections, these settings stayed consistent for all the fuzzy rules, fuzzification and defuzzification processes. These operators were not optimized for performance but rather for intuitiveness, design flexibilities and simplicity.

5-2-2 Atmospheric Fuzzy Logic Control Block

This block generates values for valence and arousal based on prior psychology research on the impacts of different weather factors on people's mood (e.g., [31], [30]). We decided to use the most important weather condition factors that are thought to most universally affect people in a certain geographic area to get a rough estimate of which quadrant of Russell's widely accepted circumplex model of affect the user might be in [17]. The objective here is not to accurately determine the user's mood but to get a general idea depending on the impacts of weather on their mood. The output values are used to retrieve songs for the 'mood' playlist. The four different categories of the mood playlist include happy, sad, calm and angry—as per Russell's circumplex model of affect.

Inputs	Humidity, temperature, air pressure, sunshine
Outputs	Valence, arousal
Membership Functions	Triangle, Trapezoid
# of Fuzzy Rules	90

Table 5-2: Atmospheric-Based Fuzzy Logic Control Model Inputs & Outputs

Example fuzzy rule: *IF Temperature IS VeryHot AND Humidity IS High AND Sunshine IS High THEN Arousal IS Positive*

5-2-3 Indoor/Outdoor Fuzzy Logic Control Block

The outputs from this block are used to determine which of the two input sensors (brightness or compass) provides a higher confidence value for indoor/outdoor. We take the value that has the highest membership value for a given state and use it as the indoor/outdoor state for the user. This value is then input to the situational fuzzy logic control block as an input. The fuzzy logic membership values for both the brightness and compass rule blocks are adopted from Zhou et. al's threshold values as shown in their paper. Since we cannot get access to the cellular radio signals, we had to rely on additional hysteresis timer values to ensure that we do not keep toggling between the indoor and outdoor states.

Inputs	Magnetic variance, brightness, daytime/nighttime, connectivity, movement
Outputs	Indoor, outdoor, unknown
Membership Functions	Triangle, Trapezoid
# of Fuzzy Rules	9

Table 5-3: Indoor-Outdoor Fuzzy Logic Control Model Inputs & Outputs

Example fuzzy rule: *IF CompassVariance IS very High AND Device IS Moving then State IS Indoor*

5-2-4 Location Fuzzy Logic Control Block

Most of the place categories are decided by doing a reverse geo-lookup from Foursquare, including home and office. In the event that none of the known place categories are found, the location visit details are then passed through this model which outputs some known place categories. If might still be that the category is still other, but that is the final fallback place category. This value is then passed to the situational fuzzy logic control as an input for further processing.

Inputs	Visit duration, departure/arrival time, weekday/weekend, connectivity
Outputs	Home, office, other
Membership Functions	Triangle, Sigmoid, Trapezoid
# of Fuzzy Rules	28

Table 5-4: Location Fuzzy Logic Control Model Inputs & Outputs

Example fuzzy rule: *IF Visit Duration IS Short AND Arrival Time IS Distant Past AND Departure Time IS Morning AND DayOfWeek IS Weekday THEN Location IS Home*

5-2-5 Situational Fuzzy Logic Control Block

This block takes in multiple inputs and as the outputs provides the membership degree for each of the seven situational-contexts for that moment. These contexts are then used to query songs for the situational context-based playlists. This is the main control model in terms of interpreting data from various sensor sources. The system design is such that additional sensors can easily be added to create more complex rules for context inference.

Inputs	Activity, place, time of day, weekend/weekday, indoor-outdoor
Outputs	Wake Up, commuting, working, relaxing, exercising, housework, sleeping
Membership Functions	Triangle, Trapezoid, Sigmoid
# of Fuzzy Rules	56

Table 5-5: Situational Fuzzy Logic Control Model Inputs & Outputs

Example fuzzy rule: *IF Activity IS Driving AND DayOfWeek IS Weekday AND TimeOfDay IS NOT EarlyHours AND TimeOfDay IS NOT very Night AND Place IS Other THEN Context IS Commuting*

5-3 Query Generation & Music Retrieval

The final outputs from the Fuzzy Logic Context Modeling Blocks are a set of the most likely contexts which we decided as the most common situations when users listen to music. The following were introduced to the reader in Section 4-2:

1. 'Waking Up'
2. 'Commuting'
3. 'Working'
4. 'Relaxing'
5. 'Exercising'
6. 'Housework'
7. 'Sleeping'

8. 'Mood'

The 'Mood' playlist is always present because it is based on the ambient weather and changes based on the user's surrounding atmospheric conditions. There are four moods for the 'Mood' playlist: happy, angry, sad, calm. The system is designed such that no two playlists will be recommended at the same time. If the fuzzy control logic generates an output of a context that is already present, that context is then skipped.

5-3-1 Generating contextual tags

In order to generate contextual queries based on the context outputs from the fuzzy logic control blocks, we inspected the SoundCloud filter query mechanisms shown earlier and found the tags field to be the most useful in this regard. Knowing the context, we then performed a tag expansion for all eight contexts using Last.FM APIs[3] and 8tracks[4]. For the weather-to-mood tag generation process, social mood tags from [23] were used as tag seeds for the query expansion. Through this process of query expansion and the ever-evolving music community on SoundCloud, the chances of retrieving novel music is very high. To maintain sufficient diversity in the tags, we handpicked 10-15 tags for each of the contexts and saved them offline in an XML file within the application. From these tags, for each contextual query, we select 6 tags at random from the selected tag list and perform individual queries for each tag to retrieve relevant songs from the SoundCloud song database. For future extensions to these retrieval approaches, please refer to the Chapter 9.

5-3-2 Retrieving songs based on contextual tags

To further enforce recommendation diversity, we also randomly select varying creation dates for the song retrieval so that we do not only retrieve the latest songs. Upon experimentation with the SoundCloud search APIs, we observed that in order to do a multi-tag search for songs, we would have to search for songs per tag individually and then concatenate the results to generate the overall playlist for the context. This was explained as part of the system design in Section section:scmusicsource. For this reason, we limited the song query limit for each tag to 5 songs. Since we select 6 tags per context, that makes it a total of 30 songs that are returned on average per context. Since SoundCloud APIs are RESTful, the application does not need to wait for results and can simply provide a callback function to be triggered when the response is ready with the result for each query. Tables 5-6 and 5-7 gives examples of some of the tags that we crawled from Last.fm for the respective contexts. In future we intend to dynamically incorporate retrieved tags to add to the diversity but for the purposes of this thesis we had a large pool of tags hard-coded in the application to select from for the various context. We explain future extensions in Chapter 9.

[3] http://www.last.fm/api
[4] http://8tracks.com/tags

Wake Up	*summer, upbeat, feel good, awesome, chill*
Commuting	*steampunk, dj, driving, progressive, groovy*
Working	*chill, classical, piano, instrumental, new age*
Relaxing	*relaxing, atmospheric, easy listening, soothing, lounge*
Housework	*energetic, catchy, happy, sweet, smile*
Exercising	*dance, electronic, gym, house, motivation*
Sleeping	*calm, mellow, dreamy, soothing, ballad*

Table 5-6: Examples of tags used for each of the situational-based contexts

Happy	*happy, energetic, catchy, feel good, cheerful*
Angry	*aggressive, angry, energetic , loud, fast*
Sad	*sad, emotional, melancholy, slow, tragic*
Calm	*relaxing, atmospheric, easy listening, soothing, lounge*

Table 5-7: Examples of tags used for each of the atmospheric-based contexts

Once all the songs are retrieved from SoundCloud, the first song filtering happens during the JSON object parsing procedure where any songs whose titles, genres or tags contain matching keywords to any of the genres that the user dislikes are skipped. This step generally reduces the playlist by a sizeable chunk. For each song, we store the following SoundCloud metadata:

- Title
- Username/Userid
- Song Duration
- Streaming URL
- Song Play Count
- Song Favoritings Count
- Artwork URL
- Genre

From here on in, if there are no playlists available to the user yet, a new playlist meta-structure is created and returned to the user immediately. If the user already has a playlist that he/she is viewing or listening to, then the retrieved songs undergo content-analysis which is described in the next section.

5-4 Content-Based Analysis

After retrieving songs from SoundCloud and filtering them based on the user's music dislike preferences, the JSON content for each song is parsed and stored in a temporary location. If there is at least one playlist available for the user to start listening, all of the recommended songs are then uploaded to EchoNest via the latter's APIs for content analysis. The EchoNest analysis process is two-step; first the song's streaming URL is uploaded to EchoNest for analysis and then we query again to check if the analysis is complete. Once the song analysis is complete, the JSON response contains some of following high-level parameters such as tempo, energy (arousal), valence, speechiness and danceability. We skip very long songs from being sent to EchoNest (> 20 minutes) as it would take too long to analyze and keep the user waiting.

The publicly available EchoNest APIs do not offer a solution to set up callback functions after uploading a song for analysis. What this means is that our system polls every 10 seconds per song to check whether the analysis has completed or not. This workaround solution was confirmed by EchoNest as the best possible solution but this adds some analysis delay per song. In addition, because we used a test developer account with EchoNest, the API key we used has API restrictions which means that we could only process 60 simultaneous API calls per minute. After this, all further API requests would return a nil error.

5-5 Recommendation & Playlist Generation

The function of the recommendation block in our system is to filter out songs that are possibly unsuitable for that context. For each analyzed song, we filter out songs depending on the accepted value ranges for all of the related high-level features. For example, if the context is sleeping, we want to make sure that the tempo for the song is not too fast. The values used for each of the high-level features were selected experimentally after multiple iterations per context. Once the recommended playlist has been filtered by the recommendation block, the remainder playlist is then displayed to the user who can then start listening to it. Each song's analyzed parameters are then stored along with its SoundCloud metadata in the same song data structure and once all the songs in a playlist have completed analysis, the recommended playlist is then sent to the recommendation block for further analysis depending on the context.

Song Attribute	Recommender Settings
ValenceMin	0.3
ValenceMax	-
EnergyMin	0.3
EnergyMax	-
DanceabilityMin	0.4
DanceabilityMax	-
TempoMin	110
TempoMax	-
SpeechinessMin	-
SpeechinessMax	0.75

Table 5-8: Content-Based recommender settings for the 'Exercising' context

Table 5-8 shows the recommender settings used to filter out tracks for the 'Exercising' context. The algorithms EchoNest uses to compute these values are unknown. Besides the reference information available at [39] for the song parameters, no other details are known. With this information at hand, we tuned the recommender settings for each context and field tested the outcome by physically testing out the each of the situations and listening to the outcomes.

Future extensions to this section will be discussed in Chapter 9.

5-6 User Interface

We wanted to give the user minimal control over the music recommendation and these are listed below:

- Playback controls—Play, pause, previous track, next track
- Favorite song—Add song to offline favorites or save it on the SoundCloud Likes playlist
- Blacklist song—Skip song and recommendation for user-defined period (5/10/15 days)
- Allow users to select music genres they do not enjoy listening to
- Refresh all playlists by pulling down the screen
- Explicitly send user feedback in case of bugs

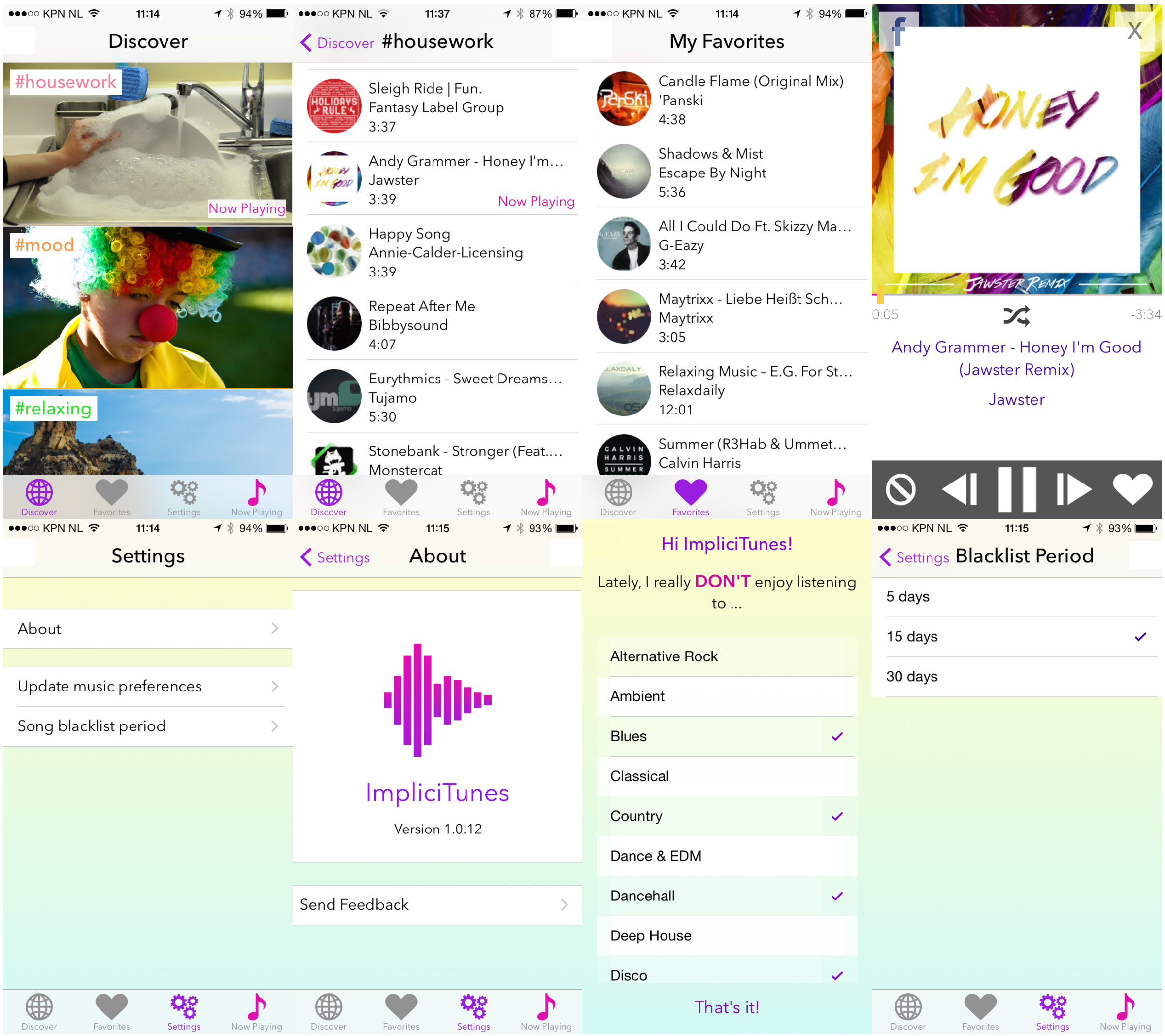


Figure 5-6: ImpliciTunes application user interface

Figure 5-6 shows all the screenshots of the final application. The goal for taking this user interface design approach was to ensure a level of simplicity not seen with other music applications. We also tried to design the user interface in such a way that it would be extremely quick and simple for the user to start listening to music. In addition, all the playlists were shown to the users via playlist images and contextual playlist titles (shown as hashtags) so that users could determine the playlist type easily.

The playlist images were selected from the Creative Commons image databases to match the context[5]. We did not conduct a poll to see how whether all the images we selected were in acceptance with the contexts that we were trying to match. This was outside the scope of our research. All the images used to convey the contextual meaning for both the situational and atmospheric based playlists are shown in Appendices C-1 and D-1. Every time a new contextual playlist is recommended by the system, one of the shown images for that context are selected in random order as the playlist album cover.

5-7 Key Design Decisions

Some of the design decisions required us to re-think some very critical aspects about our system whereas others were more modifications that needed to be made along the way.

5-7-1 Location category inference without reverse geo-coding

Our initial plan for inferring the user's location was to base it completely off of the date, time, raw GPS coordinates and activity monitoring. The algorithm was such that we would try to identify a pattern in a user's day and infer the location (such as home, office and gym) based on the level and type of activity that they took part in at that location with geo-coding it to an place category. Our first fuzzy logic control block for location was built around the logic that we would create rules for a person's daily routine and then use these simple triggers to infer location categories. Once a location category was identified, we would not need to do any more inference because we would set up location geofences and start monitoring these regions.

After field testing this approach we found it to be quite inefficient and inaccurate because the rules were written and could only be applied to a single user's routine. Writing these rules for every person and not use any third-party reverse geo-coding API service meant that we were relying completely on the logic of our fuzzy rules to decide if we were at a specific location category or not. Given the inefficiencies of such an approach, we had to abandon it and switched to using the Foursquare APIs to reverse geo-code a visited location's coordinate to a location category with a certain degree of certainty. This also meant that our fuzzy rules could be more flexible and not so rigid. Once we adopted the Foursquare APIs, we saw an immediate improvement in the location context's detection logic and hence we adopted this approach for our final system design.

[5] https://search.creativecommons.org

5-7-2 Choosing FuzzyLite as the Fuzzy Logic Control Library

Even though we had decided to use fuzzy logic as our context modeling technique, finding the right library to implement the models was not easy. Moreover, the choices were quite limited. One of the choices was MATLAB but the interface and library was quite intuitive. We were looking to find other alternative open-source software that provided the fuzzy logic functionalities and during this search we found FuzzyLite and we decided to give it a try. It turned out to be a very simple and intuitively program and allowed us to integrate multiple inputs and start building our own models very quickly. Integrating the library to work with iOS was quite challenging because nowhere on the online forums had anyone integrated Swift[6] with the FuzzyLite library and even the author of the library was unsure about the steps. We eventually ended up integrating it after some engineering effort and even created a sample project on Github so that the author could reference that to all potential users of the library[7]. This turned out to be key point because once we realized that we could use it reliably for the context modeling, it became feasible to use Fuzzy Logic in our system.

5-7-3 Automatic user logging to gather user feedback

In order to gather quantitative feedback from our test users during the evaluation stage, it as very important to have a logging framework in place for the application. This worked out great because we started collecting user logs if the device was connected to the laptop and synced. However, we realized that users would find it very inconvenient if they had to sync their smartphones to their laptops and have to manually upload their device logs every time. This would have severely limited our ability to collect valuable user feedback information. We looked at a variety of options to overcome this issue. In the end we settled on using Amazon's S3[8] as a storage location for all user logs. All user logs were designed to be uploaded once every day to our Amazon S3 storage—all done in the background, without the user ever having to manually send any logs. Of course we left a manual feedback option for the user in case they noticed something strange on the user interface or an incorrect behavior with the application while testing it. This approach proved to be greatly beneficial for gathering runtime user logs for post-analysis.

5-8 Conclusions

This chapter provides the reader with an overview of how the system design presented in Chapter 4 was implemented and how each sub-component plays its role in determining the final output—generating novel music playlists for users. The system is scalable and flexible enough to incorporate additional sensory data and generate more playlists as outputs. We list below some of the key benefits of the application:

- Auto-curation and generation of contextual music playlists
- Highly flexible and scalable context modeling block
- No learning data necessary to start recommending novel music

[6] https://developer.apple.com/swift/
[7] https://github.com/senabhishek/fuzzyliteAndSwift
[8] https://aws.amazon.com/s3/

- Very simple user interface where content is the most important

The overall implementation of this application was a massive engineering effort which involved learning a number of new techniques and technologies including fuzzy logic modeling, Swift (Programming language), iOS Programming Framework amongst others. To give the reader a scope of the project—there are currently 10296 lines of Swift code and 4 fuzzy logic models. Since the system implementation followed so closely with the design put forth in Chapter 4, there were no major design changes during the implementation phase. We did however underestimate the amount of time that needed for field testing, especially since we were using so many sensory inputs and that is something that we should have factored into the planning process. Looking back at how we envisioned our application in the design principles, we think the end product is exactly what we had in mind for users. Overall, we fulfilled all the requirements set forth by the guiding principles and the design and in the next chapter we discuss how we went about evaluating our system.

Chapter 6

System Evaluation

This chapter discusses how the system described in Chapter 5 was evaluated with users. Given that we involved users in our design process throughout the thesis and also because the final outcome of our proposed system is a user-facing service i.e. music, it was very important to deploy an application to users to gather feedback and use these results for the basis of our analysis. The engineering effort invested into developing this application and all the infrastructure needed to support it was quite immense given that the author was the only person involved—10296 lines of code, user interface design, setting up mechanisms for data collection, conducting surveys, getting continuous feedback from people, field testing the application in different locations, debugging complex technical and usability issues and more. Even though we followed a continuous build-measure-learn loop to iteratively improve our system, this chapter focuses on the final user test we conducted and to give the reader more details about the objectives and strategies used in the evaluation process.

6-1 Evaluation Objectives

The goals for the user test were two-fold: answer the research questions and also gather additional qualitative and quantitative insights about the application. Each research question has its own evaluation methodology which is explained below. In addition to the research questions, the exit survey also contained a number of questions to measure qualitative feedback with regards to the user interface, context, music recommendations and other general feedback. Figure 6-1 shows the evaluation timeline.

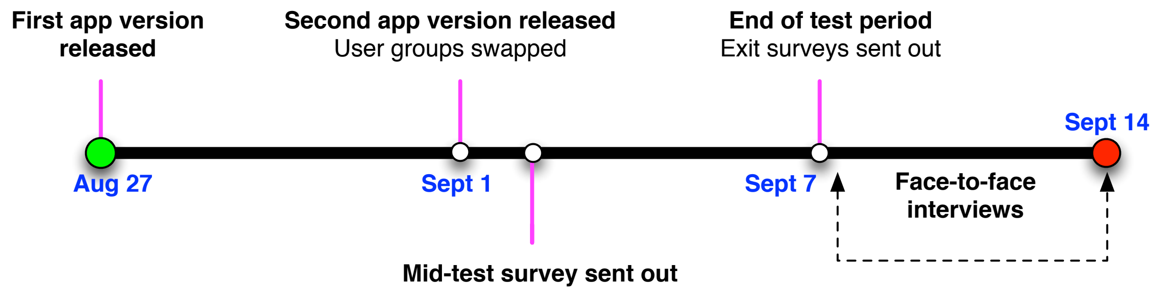


Figure 6-1: Evaluation process timeline

Both the surveys (mid-test and exit survey) have been attached to this thesis in Appendices I and J. In the following sections we will discuss more about the details related to the timeline, the different application version and the user surveys.

6-2 Formulating strategies to answer research questions

The following research questions were introduced to the reader in the introduction. Through the evaluation process, we wanted to ensure that through our strategies we would be able to answer all three research questions and additional information.

What information shown on the user interface conveys a sense of context to users?

Given the importance of providing explanations in recommender systems, it was important for the system to convey this critical piece of information to users. The use of sensors to understand the user's situation is unique to our system. From the focus group discussions, participant responses were mixed on whether they wanted to see raw sensor information (activity type, tempo, weather etc.) or to simply see music content on the application's user interface. They did agree however that they would like to know how the recommendations were made and provide them information about the playlist without giving too much visual information.

With these insights, our approach was to develop two versions of the applications where in one we showed contextual images along with some tag text that would convey the situation to the user. The images were selected from the Creative Commons image databases to match the context[1]. In order to try and answer this question, two versions of the applications were tested. The first version showed contextual playlist images and contextual playlist titles whereas the second version only showed contextual playlist images. To form the contextual playlist title, we appended a '#' character to the name of the context and showed that in text box on top of the contextual playlist image. The idea of this evaluation strategy was to see if users got a sense of the context based on only the images or did they also need the playlist title along with the image to aid in the process of playlist selection. This test might seem at first glance that it is only a test for the user interface, but the real purpose behind it is to convey a sense of context to the user through images and text. Figure 6-2 shows

[1] https://search.creativecommons.org

screen captures from the application that were used in the exit interview survey and we asked users which of the 3 screen captures they found to be the most intuitive in terms of playlist selection.

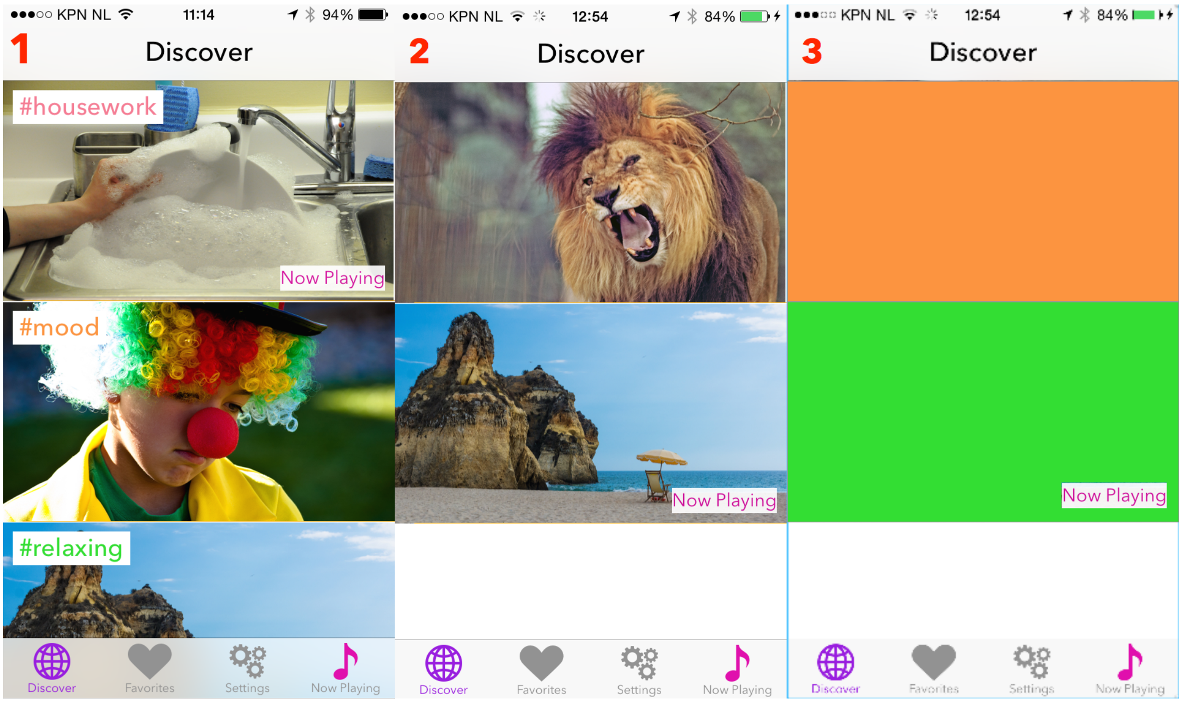


Figure 6-2: Exit interview question to see which screen capture users found most intuitive in terms of playlist selection

Which playlist type (situational or atmospheric) do users prefer to listen to?

As shown in Appendices C-1 and D-1, users will get to view two types of playlists through the application. The first, situational-based, consists of the 7 most common situations that happen daily—wake up, commuting, working, relaxing, exercising, housework and sleeping. These playlists are recommended to users by first inferring the situation that users might find themselves in using the sensors present on their smartphones and then using this situational information to retrieve and recommend relevant music from SoundCloud.

The second type of playlist that users will see is the atmospheric-based playlist that will always be present on the application's user interface. The reason for its omnipresence is that we use the ambient weather conditions (temperature, humidity, air pressure, sunshine) from the user's current location and use this information to map it one of four moods—happy, angry, calm and sad. Similar to the situational-based playlists, the moods will then be used to query relevant music from SoundCloud and then curated for the user.

Through this research question, we want to know which type of playlist do users prefer to listen to the most. On one hand, the atmospheric playlist will always be present but on the other hand different situational playlists will be available throughout the day for the user to listen to. We answer this question with a two-fold approach. First, since we collect information from the user's smartphone at all times, we want to then post-process the data to see what type of playlists they listened to most. In addition, we also ask user's through our exit survey about the playlist type they prefer listening to. We realized that it would be important to see whether what users said they did and what they actually did from the user logs would corroborate. The results for this question will also be shown and discussed in the next chapter.

Is content-based song re-ranking relevant and/or necessary for novel music recommendations?

Content-based song re-ranking refers to the process of re-ranking all the songs retrieved from SoundCloud based on the song's high-level acoustic attributes (e.g. tempo, energy, valence). This process has been elaborated in Section 5-4. These acoustic attribute values are used in the recommendation process to make sure that the contextual songs retrieved from SoundCloud do indeed match the current situation of the user. For example, if the user situation is reported as 'Exercising', we would not want any low tempo songs to present in the final playlist as this does not correspond with the context. We were not sure how often users would use the application and if we would have enough user data to answer this question that is so fundamental in current music recommender systems. Secondly, the EchoNest APIs which were used to perform the content-based analysis had an API limit constraint which meant that if an excess of 60 API calls were made to their servers in a minute, all subsequent API calls would be returned with an error. In addition, performing content-based analysis at runtime is quite slow because all the songs need to be analyzed and this round-trip time from SoundCloud to EchoNest and then back to the user's smartphone was often 5-6 seconds per song.

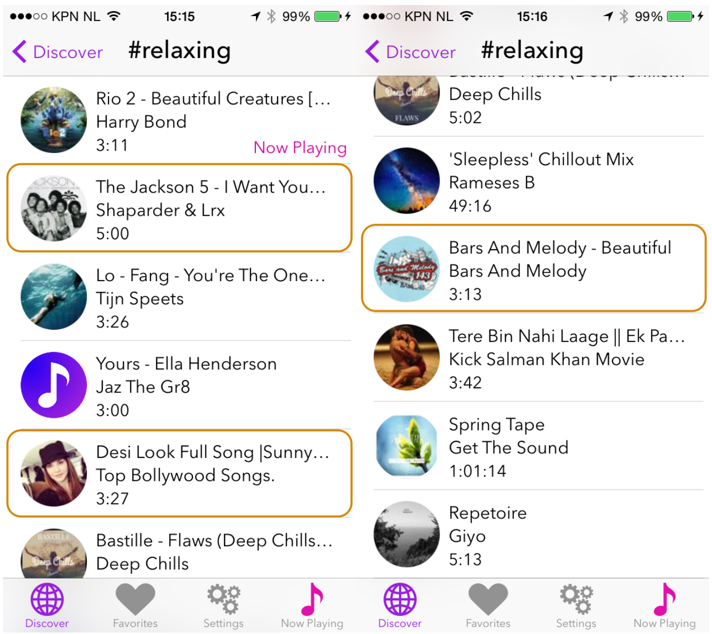


Figure 6-3: Example application screenshot for content-based re-ranking showing the songs that would be skipped if content-based re-ranking were to be used for the recommendation process

Thus, to make sure that we collected as much user data as possible, we always had content-based analysis enabled for all users during the evaluation period. The intention is that once the testing period ends that we would analyze all the user logs to check what percentage of the songs played were re-ranked. There would also be two surveys (one at the end of the 5-day test period and the other at the end of the entire test period) as seen in Figure 6-1 where users would be shown a playlist highlighting songs which would be removed by the content-based analysis. Users would then be asked their preference of whether they wanted to see the songs removed depending on whether they deemed the songs 'unsuitable' for the context. During the evaluation period, we hypothesized that most users would listen to the 'relaxing' playlist because every day people would go back home and the possibility of using the application in the evening could be high. This was a hypothesis and thus we decided to use the 'relaxing' playlist as the playlist of choice to answer this choice in the user surveys. Figure 6-3 shows an example of the application screenshots where the user is shown a playlist showing songs that would be skipped if content-based re-ranking was employed. The follow-up questions are present in the mid-test survey in Appendix I and also in questions 31-33 of the music recommendation section of the exit survey in Appendix J.

6-3 Participants

There were 22 users between the ages of 21 and 45 who used and evaluated the application over a 10 day period (August 27, 2015 - September 6, 2015). 20% of them were females and the rest males. We found these users through Facebook and through our social connections. They were a good mix of working professionals and students and were split into 2 groups of 11 where each group tested one version of the application for 5 days. All the users had either an iPhone 5S/6/6+ as their smartphone with a minimum software requirement of iOS 8.1. The groups were divided such that there was an even distribution of smartphone types in both groups. After the 5-day period, the groups were swapped so that all users got to use both versions of the application as discussed earlier and shown in Figure 6-1.

Users were unpaid and participated voluntarily for the evaluation. In order to have an effective testing period, it was important to keep users engaged and have the feedback process as simple as possible. For this reason, all information collected from the application was done in the background without requiring the user to explicitly do anything besides listen to music. Given that our system is a context-aware that responds to our environment to provide novel music recommendations, it was important that our application was tested in different locations. Figure 6-4 shows map of where all the user were located globally for the test. The application was deployed to all users through iTunesConnect (Apple Beta App Deployment Platform)[2] and users received all updates through the same interface.

[2] https://itunesconnect.apple.com

Figure 6-4: Global geographic spread of user for the 10-day testing period

User participants were asked to sign an informed consent form which means that users were aware that during the evaluation process, their application data would be collected and analyzed for research purposes in the background.

6-4 Face-to-face Interviews

Following the end of the test period, we also intend to sit down with our users for a face-to-face briefing of the overall test process, their thoughts on the application and how it could be improved in future. The structure for these interviews was more of an informal type where we would try to get as much qualitative information from the users as possible.

6-5 Conclusions

This section explained to the reader the evaluation used to gather qualitative and quantitative information about the user during the test period. The next chapter discusses in details the results and also the conclusions that resulted from the overall evaluation period.

Chapter 7

Results & Discussion

Finally we present the results from the user evaluation with additional discussions. Each of evaluation techniques (user logs, mid-test survey, exit survey, face-to-face interviews) gave us valuable insights about the system and we received lots of constructive feedback for the application. The evaluation period ran from August 27, 2015 to September 6, 2015. The infographic in Figure 7-1 shows some of the user statistics gathered from the user application logs during the course of the evaluation period. For the user's reference, both the mid-test and exit interview surveys have been added as appendices to this thesis (Appendix I & J).

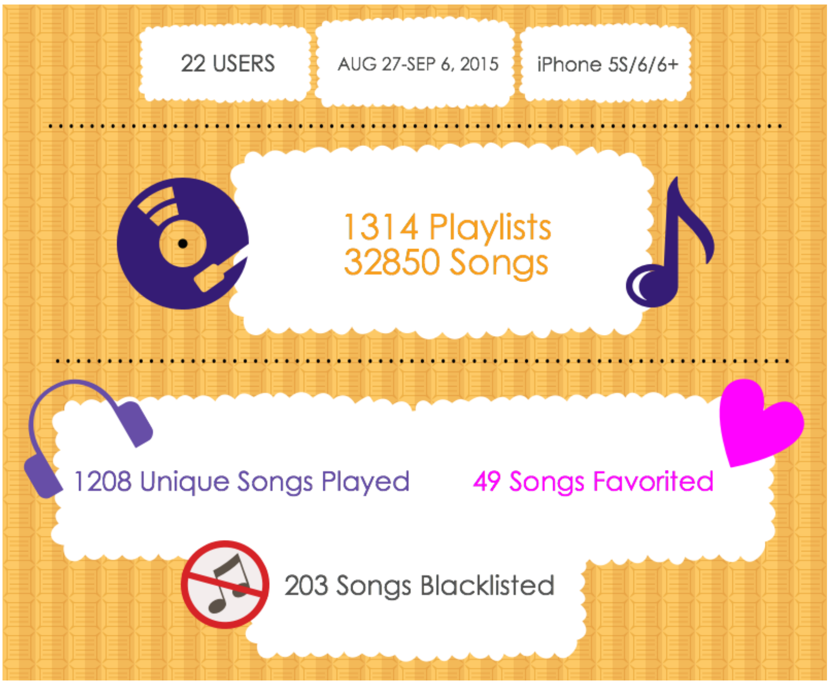


Figure 7-1: Evaluation period statistics

We will first answer all the research questions, follow it up with additional insights that we gathered from our users and then conclude the chapter by listing the key take-away points from the evaluation period that will help us provide some concrete conclusions in the next chapter.

7-1 Answering the research questions

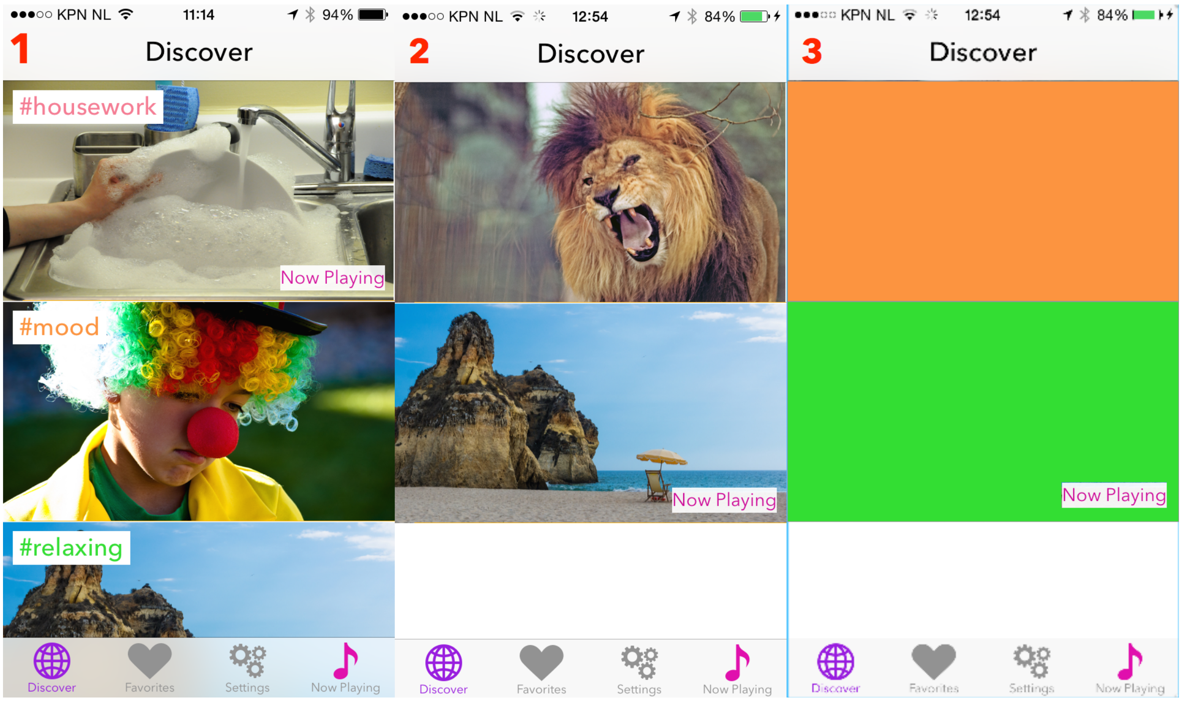


Figure 7-2: Exit interview question to see which screen capture users found most intuitive in terms of playlist selection

7-1-1 Research Question 1: What information shown on the user interface conveys a sense of context to users?

In order to answer our first research question, users used 2 versions of the application during the evaluation period—one with only the playlist images and another with the playlist images and the playlist tags. Screen captures 1 and 2 in Figure 7-2 asked users to provide feedback as to which one of the 3 (the third one was to just use colors instead of tags or images) made the playlist selection process the most intuitive. It should be noted though that users did not test an application version where only colored tiles for playlist choices and that this was only asked of users in the exit survey. We asked this question to see if users could in fact understand simply from colors the contextual information that we were trying to convey. More than 90% of users preferred the first screen capture in Figure 7-2 with the playlist image and the playlist tag as it helped make the playlist selection more intuitive. This answered our first research question which was to find out what information shown on the user interface would convey a sense of context to users.

7-1-2 Research Question 2: Which playlist type (situational or atmospheric) do users prefer to listen to?

We employed two strategies to answer the second research question, which was to find out which playlist type (atmospheric or situational) users listened to most often during the evaluation period. A small caveat here was that while the atmospheric or 'mood' playlist was always available for users to listen to, there was only one 'mood' playlist available at all times. In contrast, there were seven situational playlists ('wake up', 'commuting', 'working', 'relaxing', 'exercising', 'housework' and 'sleeping') available at different times during the day depending on the user's situation. Even though not all these playlists were available for users at all times, they would get a larger variety of playlists to listen from. So, on one hand we had the 'mood' playlist available at all times and on the other we had seven situational playlists available for different user situations during the day.

Figure 7-3 shows the results from the exit survey when we asked users to recall how often they listened to both situational and atmospheric playlists. As we can see, since the 'mood' or atmospheric playlist was always available for users to listen to, a larger percentage of users answered to having listened to the atmospheric playlist more often than the situational playlist. There were unsure users who were not really sure and this can be explained by the fact that most times users would be listening to music and not look at the screen directly to check which playlist was currently active. The margin of difference is not really big however.

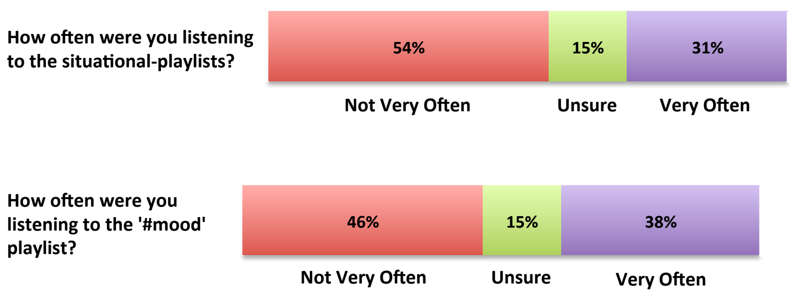


Figure 7-3: Playlist listening behavior based on exit interview feedback

To make sure that users responded accurately about their playlist listening behaviors, it was also important to look at all the user logs collected during the evaluation period too see if what users said they did and what they actually did made sense. The user log data presented in Figure 7-4 shows the playlist types of all the songs played by users during the evaluation period. The logs were collected by the application in the background without any user steps involved in this process. For this data, we only look at all the unique songs played by all users and check to see what was that song's playlist category. One thing to note however is that since we did not track every time a user skipped a song, we are not really sure if a user skipped a song as soon as they started listening to it or towards the end of the song. To avoid any data overlapping issues, we only selected uniquely played songs for this user log analysis.

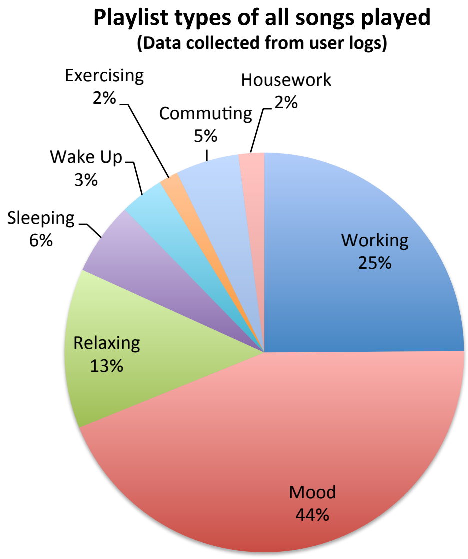


Figure 7-4: Playlist types of all unique songs played from user logs

Looking at the user log data shown in Figure 7-4, we can see that 44% of the time users listened to songs from the atmospheric playlist. There is a 6% discrepancy from the qualitative feedback data we collected from the exit survey shown in Figure 7-3. If we compare these numbers with the raw user logs, the user feedback data corroborates and this is important because we can then say that the feedback from the exit survey and trusted.

Figure 7-5 shows the user preference of which playlist type they preferred listening to in general. We can see that a majority of the users preferred the situational-based playlists over the atmospheric-based playlist. This finding can be because of a couple of reasons. As we saw earlier, users found the playlist title of 'mood' to be confusing as they were expecting something else at times. Also, the mood playlist contained too much song variety in the playlist as per the users. Regardless, the results in Figure 7-5 help answer the second research question where we see that users preferred situational-based playlists more than atmospheric-based playlists.

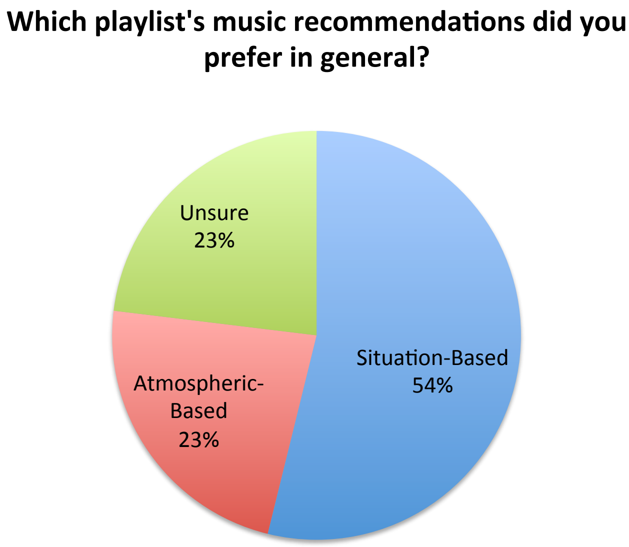


Figure 7-5: Overall user playlist type preference

7-1-3 Research Question 3: Is content-based song re-ranking relevant and/or necessary for novel music recommendations?

Of the 1208 unique songs that users played during the evaluation period, only 3% of the songs were analyzed by EchoNest's content-based libraries. The rest of the songs did not undergo content-based analysis using the EchoNest APIs. After further user log analysis, we find the results shown in Table 7-1.

Total Songs Recommended	32850
Total Songs Sent For Analysis	8667
Total Songs Skipped Analysis (Duration >20 minutes)	1086 (13%)
Total Songs Uploaded To EchoNest For Analysis	7581 (87%)
Total Songs Analyzed Successfully	1058 (14%)
Total Songs with Analysis Error	6523 (86%)

Table 7-1: Content-Based analysis results from EchoNest

As we can see, of the 8667 songs from SoundCloud, 1086 of them were skipped because they were longer than 20 minutes. We put this threshold in the application because of two reasons—songs greater than 20 minutes tend to be very long with lots of musical attributes and analyzing such long songs would take a very long time. Of the 7581 songs that were uploaded to EchoNest for content-based analysis and of these, only 1058 (14%) of them completed a successfully analysis summary. This means that 6523 (86%) of the songs could not be analyzed during to an API error with EchoNest. Our application's EchoNest API key had a cap of 60 API calls per minute but it seems like this limitation might have been exceeded causing a lot of subsequent API requests made to EchoNest to fail.

The implications of the data presented is that we need to rework the algorithm in order to allow for more songs to undergo EchoNest's content-based analysis without compromising user experience and to answer decisively whether or not content-based analysis would have improved the recommendations. In addition, since only about 3% of all the played unique songs were analyzed, we cannot rely on the user logs fully to answer this research question. The reader might ask as to why only 8667 songs were available for analysis when 32850 songs were recommended in total. The reason for this is that we had a check in the application such that at least 2 playlists needed to be present at all times for user viewing and only subsequent playlists would be eligible for content-based analysis from EchoNest. This was done to ensure a responsive user experience otherwise users would be stuck on the main application page without any recommended playlists especially because the analysis usually took a while and often times would error out as we see that almost 86% of songs uploaded for analysis failed for reasons beyond our control. We explained this possibility in Section 6-2.

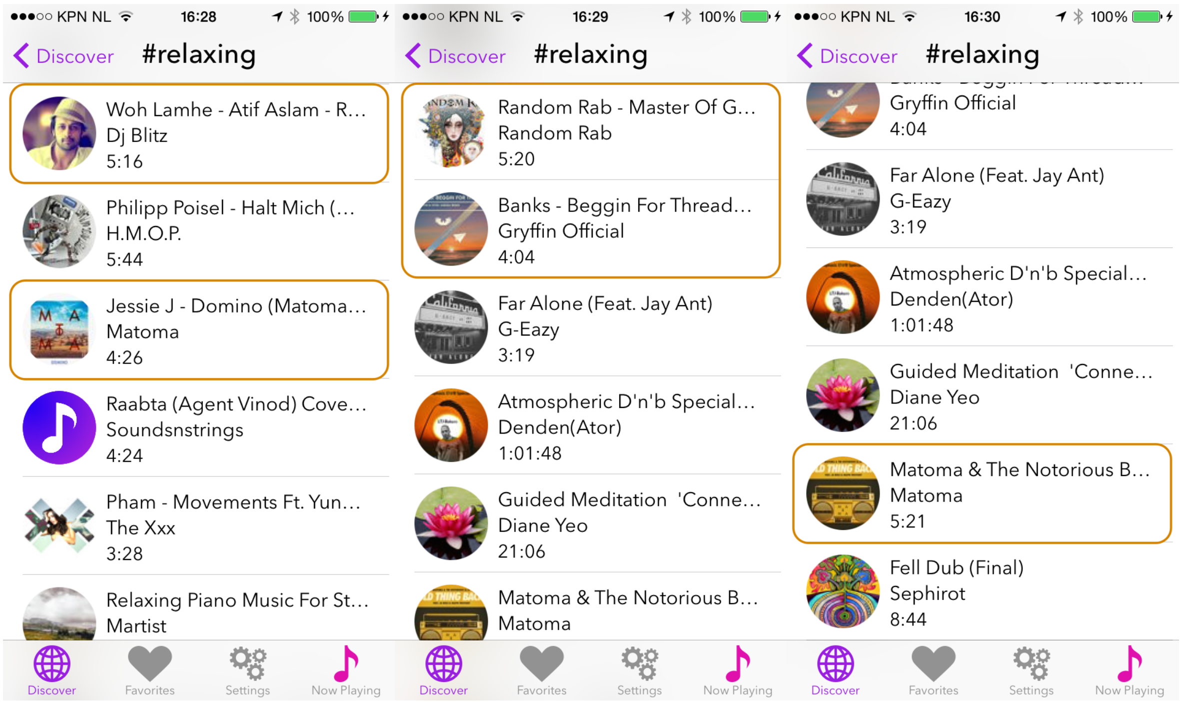


Figure 7-6: Sample 'relaxing' playlist used to ask users about their opinions on re-ranking songs based on content-based analysis

We look to the survey results where we showed users' screen captures of a 'relaxing' playlist along with boxed titles of songs that would have been skipped for recommendation based on the EchoNest analysis. Users were then asked how pleased they would be if the system automatically decided to skip the songs shown in the highlighted boxes in Figure 7-6 with their responses shown in Figure 7-7 and Table 7-2.

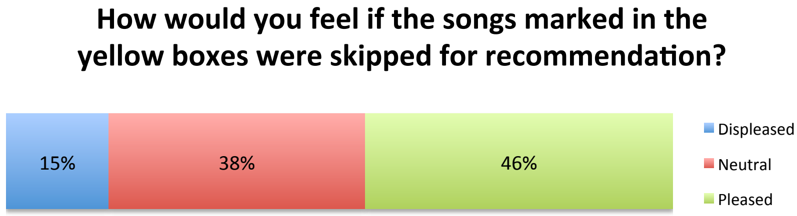


Figure 7-7: User responses to system skipping potential playlist mismatching from recommendation

Feedback	Frequency
Don't know enough about artists/songs in the playlist to comment	4
Trust system to make make the right music recommendations for me	2
Don't mind system curating based on my situations and preferences	2
Don't understand the question	2
Like songs that would've been skipped	2
So much music to listen to, glad system makes choice a bit easier	1
Happy with the choices the system would've made	1
Would've listened to different 'relaxing' music	1

Table 7-2: User feedback about content-based playlists that were ranked

Looking at the user feedback results, it seems that users do not seem to mind the system trying to help them out in filtering out music given the abundance of choice that they are always inundated him. As we can also see, not a lot of users would really be upset if a song was skipped from being recommended due to the system but they want their music preferences and their situation to be taken into account. Given the diversity in the recommendations it was also a little difficult for some users to give a definitive answer to this question because all they could base their decision was based on the song's artist, title, artwork and possibly the duration.

Thus, after conducting this experiment, we can see that content-based music recommendation system can help filter out unwanted songs from a playlist and also help matching songs to their situations. The quantitative data from the user logs was not enough to suffice a definitive answer to this research question. We can conclude by saying that even though we did not get a clear cut answer about this question from users, the logical feedback would be users appreciated the fact that the application was trying to help them by curating their music based on preferences and situations. So, content-based recommender systems can help improve the quality and relevancy of the recommendations. Thus we think that content-based playlist re-ranking is not mandatory but definitely a relevant area for possible future plans.

7-2 Additional Insights

In this section we present more detailed results and analysis of the data and feedback that was collected as part of the evaluation period. There are four sub-sections here–user interface, context recognition, music recommendations and general feedback. Each one of these sub-sections form a critical part of the overall application and contributed to a very successful user evaluation period.

7-2-1 User Interface

As stated in the design principles in Chapter 3, we wanted to make the application's user interface as simple and intuitive as possible. Very often applications tend to confuse users with unnecessary features and user interface elements that distract users and prevent them from focusing on enjoying the core propositions of the system which in our case is the music. Appendix B shows all the screenshots of the application. The user interface elements went through multiple iterations upon receiving feedback from the focus group, pre-beta user trials and general feedback from discussions with people. We showed all the screen captures to our users in the exit survey (Appendix J) and asked them questions with regards to the user interface. Tables 7-3, 7-4, 7-5 and 7-6 present the qualitative feedback with regards to the user interface elements of the application.

Feedback	Frequency
Selecting music dislike preferences	5
Dislike/Blacklist button unclear	4
Crosses/Strikethrough for music dislike preferences	4
Blacklisting	3
Don't know some of the genres to dislike	3
How and when playlists are generated	3
Some playlist images	2
Viewing song playlist	2
How to influence playlist generation	2
Mood playlist text (Expecting it to convey user's mood)	1
Different font colors for playlist	1
Favorite button	1
Lack of home screen	1
How are playlists influenced when song is liked/disliked	1
Unsure of what information was being used for music recommendation	1

Table 7-3: What users FOUND CONFUSING about the user interface

Feedback	Frequency
Simple, clean & easy to use	8
Clear navigation	2
Icons in the tab bar	2
Now Playing Screen with big buttons	1
No login needed to listen to music	1

Table 7-4: What users LIKED MOST about the user interface

Feedback	Frequency
Limited playlist choices - sometimes only 1	3
Playlist screen could be more attractive	2
Uninviting user interface, especially now playing screen	2
Wasn't sure when a new playlist was available	1
Progress bar seeking in a song was difficult	1
Now Playing screen very different from rest of the application	1
Selecting playlist button twice to see song list	1

Table 7-5: What users LIKED LEAST about the user interface

Feedback	Frequency
'Forever' blacklist period	2
Playlist tags too blocky - maybe use blur	1
Favorite a playlist	1
Request button to get more playlists	1
Add to queue feature	1
More information why a song was picked (genre, activity etc.)	1
Dislike button	1
Logical user guide for the application using visual cues	1
Music Preferences & Blacklist in Settings foreground	1
More Info about App in About screen	1
More information about privacy	1
Want to see Menu Bar in Now Playing Screen	1

Table 7-6: What controls users were MISSING from the user interface?

Looking at the feedback presented in Tables 7-3, 7-4, 7-5 and 7-6, we see that users overwhelmingly appreciated the fact that the application's user interface was clear, simple and intuitive to use. We also see that users found it confusing and unnatural to select and checkmark their music 'dislike' preferences—something that they not used to with most of activities. In addition, some users also complained about the limited playlist choices available to them at all times and this is another thing to be improved upon in future. Users also seemed a little confused about the blacklisting feature and we think the application could have done a better job at explaining this user control feature to them. Most of the items in Table 7-6 are valid and we will try to incorporate them into the application in future.

Continuing the user interface feedback and analysis, we also wanted to find out from our users about the interplay between the playlist tags, images and the meaningfulness of them both. Figure 7-8 shows what users thought about the relationship between the playlist images and tags.

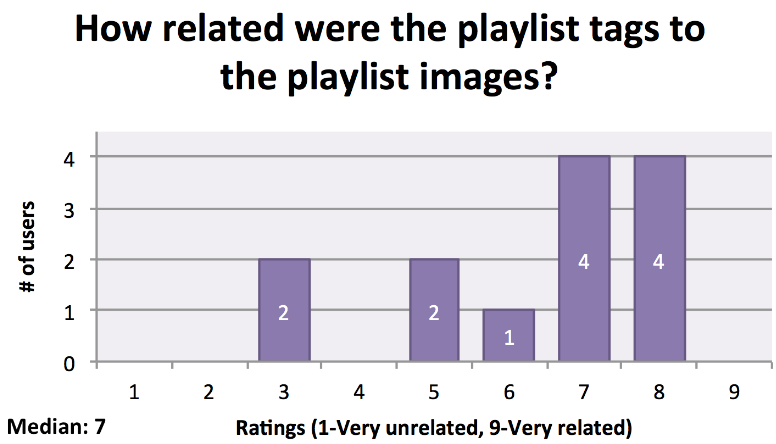


Figure 7-8: What users thought about the relationship between the playlist tags and the playlist images.

The user feedback results from Figure 7-8 tell us that even though contextual image search was not a focus point of the research, from the ones selected for the different playlists, with a median value of 7 we see that users found the playlist tags more related to the playlist images than not. This adds further credence to our findings for the first research question because we can now say that both playlist tags and images, if selected appropriately, can provide users with a sense of context that will help them draw a connection between the situation and the type of music they can expect listening to that playlist.

Feedback	Frequency
Confused by the 'mood' tag	6
Did not understand difference between 'work' and 'housework'	2
Playlist tag did not correspond with playlist content	1
Images for 'mood' playlist were unrelated	1

Table 7-7: What users found CONFUSING about playlist tags and playlist images

Table 7-7 shows some of the qualitative feedback that users gave with regards to the playlist tags and playlist images. Users did not get quite understand the 'mood' tag and were confused whether the application was trying to guess their mood or something else. Also, some users mentioned that some of the images for the 'mood' playlist did not quite make sense. For example, there was a picture of sad clown for the 'mood' playlist which was supposed to be playing sad songs. However, one user said that the playlist instead contained quite happy songs. We think that in future revisions of the application, we will need to re-address on how we can convey the atmospheric conditions better and possibly changing the 'mood' title to something else. This was all the user feedback that we collected with regards to the application's user interface and we shall now move on to discussing the results about the user's context.

7-2-2 Context Recognition

This section of the thesis deals with presenting the findings from the exit interview and also the user logs that were collected during the evaluation period with regards to the contextual inference modeling part of our system. As stated earlier, users are recommended one of the following seven situational playlists ('wake up', 'commuting', 'working', 'exercising', 'relaxing', 'housework', 'sleeping') or the atmospheric-based 'mood' playlist. As the user went about their day and experienced different situations, they would be recommended either one of the above seven playlists. The 'mood' or atmospheric-based playlist would always be available to the user because it is based on the surrounding weather conditions which is omnipresent. It was important for us to get feedback on whether or not our users could relate to these playlist choices hence we asked this question as shown in the Context section of the exit interview in Appendix J. Figure 7-9 shows the distribution of which playlists users could most relate to. From the data it seems like users could relate to most of the playlists excepting for 'housework'. We also saw this feedback in Table 7-7 so this is something that we can improve upon and make more transparent to the user.

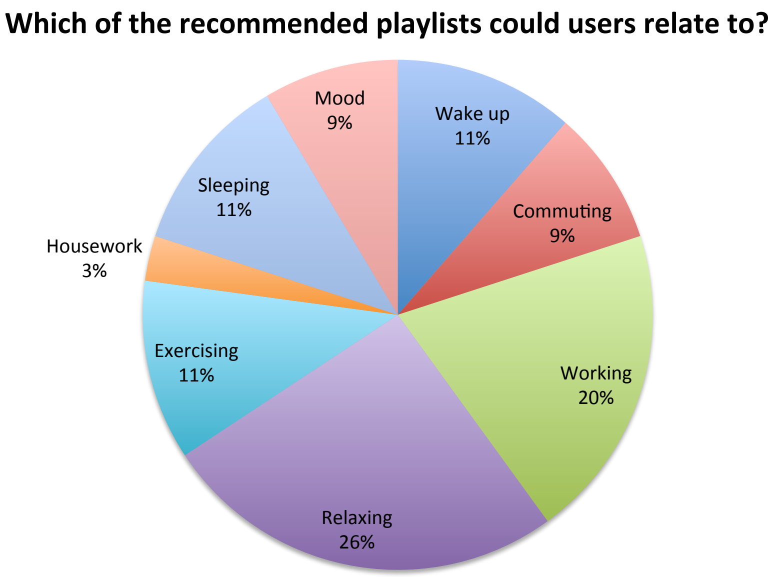


Figure 7-9: The playlists that users could relate to

One of the outcomes from our focus group session was to limit the amount of sensor information that was used to model the context to be shown on the user interface to the user. As a result of this, we took the extreme approach of conveying a sense of context to the user only through using images and tags. Interestingly however, our users told us in overwhelming majority that they would like to know more of why the playlists/songs were recommended for them as shown in Figure 7-10. This result tells us that 85% of them wanted to know more of why certain playlists were recommended to them. From the face-to-face interviews, another insight that came out was that some users wanted to know what hash-tags and/or keywords were being used for specific songs/playlists because they wanted to possibly find similar songs. Also, given that the system's recommendation core is purely coming from retrieving songs from contextual sensor data, it makes sense that users want to know how and why exactly certain playlists/songs were being recommended to them. This is another area of future work for this thesis.

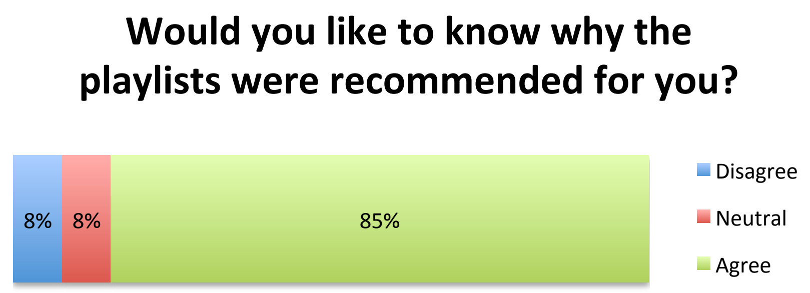


Figure 7-10: Users wanting to know more of why playlists were recommended for them

We now need to see how users could see that the playlists matched up to what they were doing while testing the application. Figure 7-11 shows the results where we see 46% of participants saying that they found the recommended playlists matching the situation they found themselves in. However, there are also 31% of people that tended to disagree with this question which means that this quantitative data might not be sufficient to answer this question.

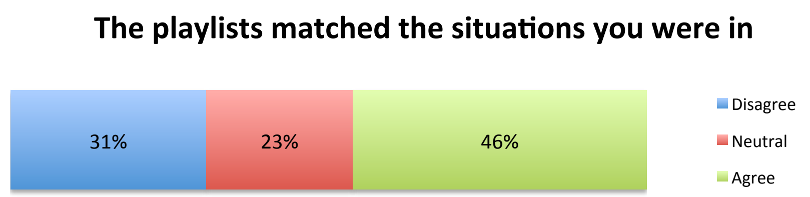


Figure 7-11: Users who found playlists matching their situation

To justify why we had 31% of people disagreeing with the question that the recommended playlists matched their current situation, the following qualitative feedback provides more insight:

- One user did not use cellular data so all of their testing was done from a home location. This meant that they also only saw the 'mood' or the 'working' playlists and none of the others.

- Another mentioned that one of times when the playlist was not matching his situation was when they were getting ready to go out in the evening to a party or bar and then the situation did not match.
- Anothere instance where user drove to work very early in the morning and they still got the 'wake up' playlist instead of the 'commuting'.

The results of these questions mean that the context recognition in terms of situation can definitely be improved to accommodate for additional situations. However, given that fuzzy logic allows for the handling of vague data very effectively, we think that this issue of doing better context recognition can definitely be improved over time by looking at additional contextual data and addition/modification of fuzzy rules offline.

Having found that our users could relate to most of the recommended playlists, we thought it would be interesting to see what other playlists they would be interested in listening to. This question allowed us to gather valuable information for future development of our system. Figure 7-12 shows all the new playlists that users would want to listen to on top of the ones they were already recommended. Some even suggested that we change some of the names of the current playlists, for example, from 'relaxing' to 'chill', 'exercising' to 'workout'. Looking at this feedback, we see that we could easily incorporate these additional playlists into our system by adding new outputs to our Situational Fuzzy Logic Control Block and retrieve contextual music accordingly. In addition, during our face-to-face interviews, another feedback that we received was that even if the user's situation is somewhat static (working for example), users would like to see more choice in the playlists—at least 3-4 playlists at all times.

Figure 7-12: Users wanted to additionally listen to these playlists

This ends the results analysis for the Context section and we will discuss the results from the user generated logs and also the exit survey results with regards to music recommendations in the next section.

7-2-3 Music Recommendations

This section of the analysis deals with the user feedback results and analysis regarding the contextual music recommendations that users listened to. As a refresher, the application performs context modeling by inferring the user's situation from the sensor data, retrieves relevant contextual songs through the use of tags from SoundCloud and then curates them into playlists based on situational or atmospheric playlists for the user. We asked users a range of questions in order to better understand how well the contextual music recommendations performed and also collected their device logs showing application usage. Both these will be discussed in this section.

A key difference between this application and most other music applications for music discovery is that we asked users what music genres they did not enjoy listening to in order to recommend music that is beyond what they listen to every day. We learned from our focus group meeting that in addition to knowing what users prefer listening to, most users know what types of music they really do not enjoy listening to. Our application incorporates this user behavior by asking them at the onset all the genre preferences that they don't enjoy listening to. What the application does with this information is that any songs from these 'disliked' genres are skipped during the recommendation step. The goal then is to find and recommend songs that are still relevant to the user, and at the same time contextually influenced. To ensure that the user's music dislike preferences or not, we asked them about the relevance of the music recommendations based on their 'disliked' music preferences—the results for which are shown in Figure 7-13.

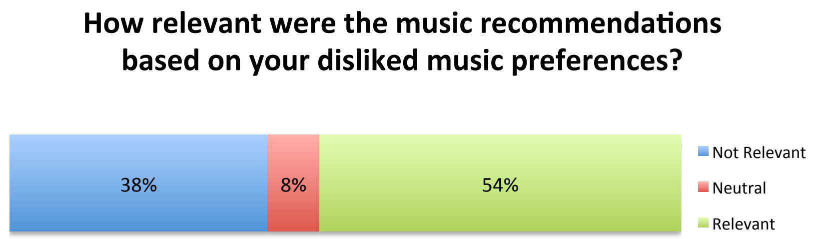
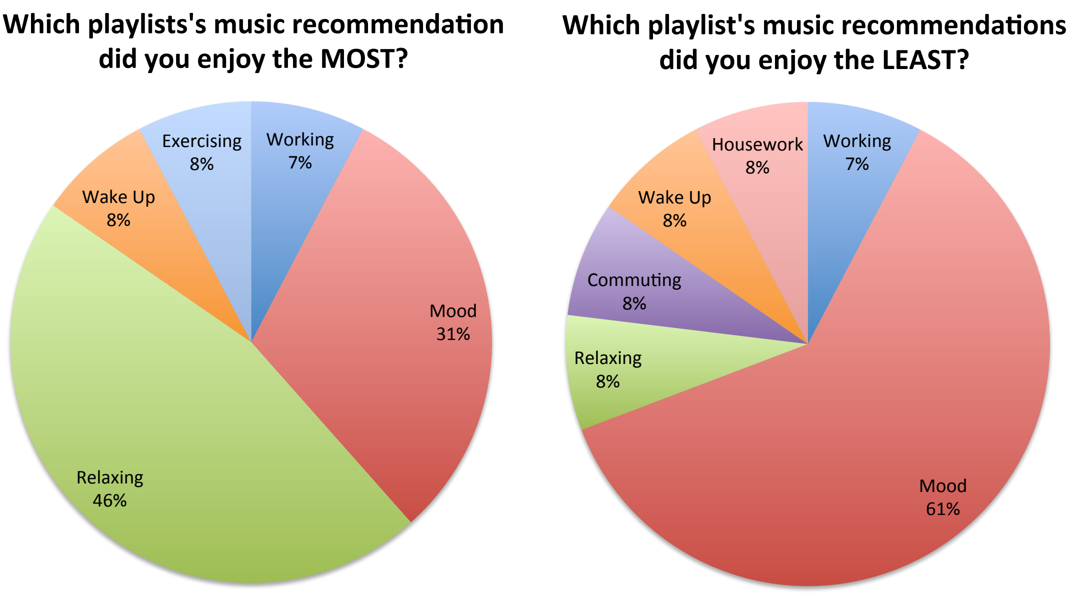


Figure 7-13: Relevance of music recommendations based on disliked music preferences

As we can see from Figure 7-14, 54% of users found the music recommendations relevant based on the disliked music preferences that they set through the application. At the same time, about 38% of people also disagreed with this notion. From the face-to-face interviews, one of the feedback comments about music relevancy was that sometimes in order to get more relevant tracks or songs that they liked, users 'blacklisted' almost all the genres besides the ones that they wanted to listen to but this greatly reduced the playlist size and did not quite provide the recommendation relevancy they were looking for. In addition, user's missed having any control over the recommendations besides updating what they do not like and 'blacklisting' songs. Some were also in the opinion that sometimes they ended up skipping multiple songs in a row in order to find something that they liked listening to. Given this feedback, one of the take-aways from this user evaluation is that the music relevance for the recommendations definitely needs improvement so that users spend less time skipping songs and more time listening to them.



(a) Most enjoyed playlist type

(b) Least enjoyed playlist type

Figure 7-14: Playlists enjoyed by users

Figures 7-14a & 7-14b show results from the exit interview where we asked users which playlists they like the most and the least and asked them to try and remember any specific incidents when either of the two happened. For reasons when they liked a playlist the most, users said:

- *"Music best fitted to the situation"*
- *"The 'mood' playlist had lots of variety and not one specific style/genre"*
- *"When at home and ready to go out, the playlist was all fun to dance to. Also when I was eating, I got some new music."*
- *"Listened to the 'commuting' playlist on the train."*
- *"It got working immediately."*

Some of the criticisms for the application in terms of contextual music relevancy from our users were that:

- *"Maybe because I did not ticked enough music genres I did not like, but there were quite a lot of numbers I did not like. So during work or something I switched it off."*
- *"I tried listening to work a few times but it didn't match my mood at the time. I noticed in general (not specific to a playlist) was that it would sometimes play non-musical audio, like an interview, podcast, or similar. That was a bit of a buzz kill."*
- *"Quite random music."*
- *"The recommendations were not obvious for me."*
- *"I was at home or work, relaxing/ working but the music recommended was too slow and sad."*

We see that about 46% of users enjoyed listening to the 'relaxing' playlist the most and almost 61% of them liked the 'mood' playlist the least. From this statistic we can say a couple of important observations. Firstly, even from earlier comments we have seen that users often did not quite understand the 'mood' playlist and what it was supposed to represent. Some of them mentioned that given the playlist title, they were expecting songs that exactly matched their mood. Others mentioned that the images used to convey mood were sometimes 'too cheesy' or misinforming in predicting their own mood. Another important finding was some users found that certain users seemed to have varying music tastes for different situations. For example, some of them mentioned that while working, they would prefer listening to upbeat music as opposed to calming and laid back music. Since the system currently relies on social tags for music retrieval, a change in the querying mechanisms would help, meaning that we do not need to change the tags but how the tags are selected. Other contexts such as sleeping and exercising are more universal in generalizing because of the nature of the activities.

7-2-4 General Feedback

Having investigated, researched and analyzed all the three research questions, we also received a lot of general feedback from our users which we present in this section. Albeit not quantitative, the qualitative data is extremely useful feedback we reckon and sometime almost more important because there is limited scope for interpreting the data incorrectly. Figure 7-15 shows how users discovered new music during the 10-day evaluation period. As it shows, while users found the application overwhelmingly useful to discover new music, they still went back to using other music sources such as YouTube, Spotify and other. This can be attributed to the fact that this application is solely for music discovery whereas a lot of other music services offer other services such as offline storage, multi-device access amongst others.

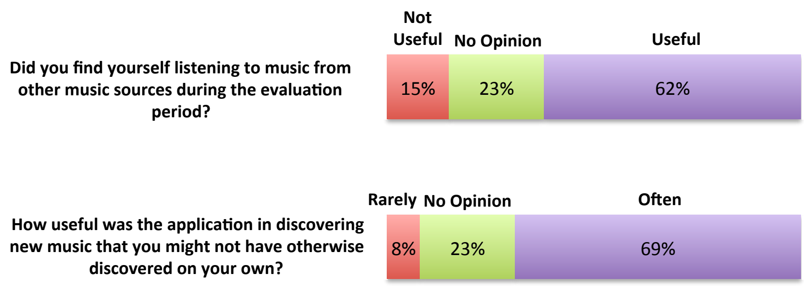


Figure 7-15: Music discovery behavior during evaluation Period

When we asked users how they thought this system was different, we received a broad range of answers—shown in Table 7-8. The feedback comments really show how this application is so different from most other commercially available music services.

Feedback	Frequency
Auto-curation playlists functionality was neat	3
Simple and easy to use for direct music playback	3
Expanded my music horizon	2
Recommended music user was not expecting	2
Recommendations based on situations	2
Different because only for discovery	1
Users had no influence on playlist generation	1
Not too many choices	1
Other music services also have discovery	1
Combination between automatic playlist selection and music preferences was nice and novel	1

Table 7-8: How users found this application to be different

As with any user evaluation, there are times when things go right and also when things go wrong. It was important to see if our users remembered instances of when they really enjoyed using the application and others when not so much. This is of course very valuable information for future development and commercialization plans of this system. Table 7-9 presents a tabulation of all the times when users really enjoyed and/or disliked using the application. This kind of feedback helps us gauge what were some of the pain points for users and also identify instances when they really enjoyed using the application so that they too can be improved.

Really enjoyed using the application	Really disliked using the application
“I found some techno music that I like"	*“Mood playlist started annoying me after some time"*
“I discovered a song!"	*“I got Ariana Grande as a recommendation"*
“Workout music was quite stimulating"	*“While working I had to skip a lot of songs"*
“Got some cool Kygo mixes while at work"	*“I wanted some chill music, I selected all except piano and still did not take my preferences into account"*
“While in the kitchen and wanted to listen to random music"	*“Was relaxing and listening to the 'mood' playlist and kept skipping songs"*
“Getting ready in the morning with some background noise"	*“Listening to songs that were not very relevant for me"*
“Listening to music I've never heard before"	*“Would've liked the ability to select my situation and/or mood"*

Table 7-9: User feedback of times when the really enjoyed/disliked using the application

Feedback	Frequency
More explanation about why playlists were recommended	4
Better user interface design and pictures	4
More playlist choices	3
More relevant and similar tracks	2
Improved battery life	2
Using location all the time	2
Ability to make custom tags in favorites	1
No login should be required	1
Smaller and more subtle buttons in Now Playing screen	1
Useful to tag a song/style and associate it to a situation for future use	1
Possibly a seeding mechanism to provide baseline set of recommendations	1
More control over playlist selection	1
Offline music	1
Quick reaction time to preferences change	1
View tags for a given song	1

Table 7-10: User suggested improvements for future work

Table 7-10 presents a final set of suggestions that users provided to improve the application in future. Getting feedback directly from users and to have them comment (positively/negatively) about your system is a very rewarding and informative experience because you get real opinions and are not left inferring things data. As we can see from their feedback, they definitely seem to want to know where and why the playlists are being recommended and also want more playlist choices. Moreover they want to get more relevant recommendations, even in discovery mode. These are very positive signs from a future development point of view because it shows that we were able to capture the user's imagination into what a truly novel and unique form of music discovery could be like in future—one where the technology works for us and not the other way round.

The final statistic that we want to show for this section is one that really matters for us—do users want to continue using the application now that the evaluation period has ended? Music recommendation as we all know is a very competitive field with numerous commercial services already available for users so capturing the user's attention and keeping them engaged is absolutely imperative. Figure 7-16 shows that 54% of users want to continue using the application for novel music discovery. This is an important outcome for us because it means that we have been able to satisfy our user group (albeit marginally) and have received great feedback to improve our system for future versions.

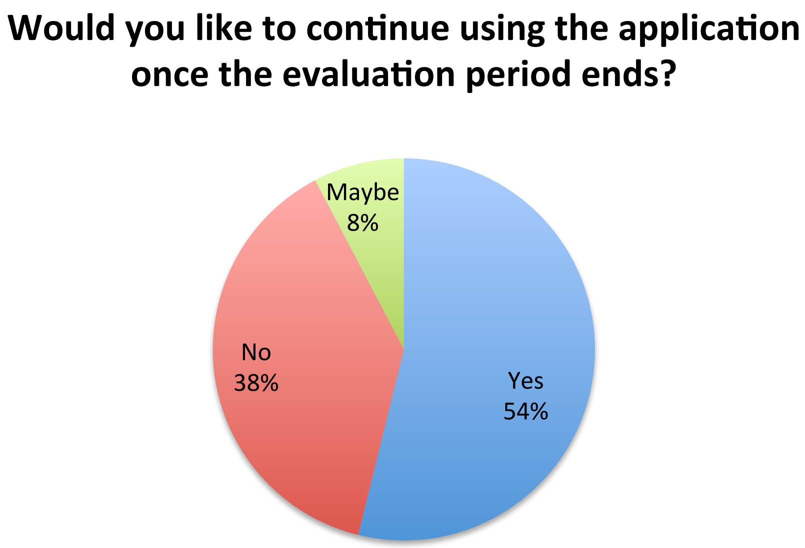


Figure 7-16: Users willing to continue using application

7-3 Conclusions

Through this section we have presented all the results and analyzed them to the best of our ability. We have gone through the four main sections of results namely—user interface, context recognition, music recommendations and general feedback. During the course of presenting and analyzing the results, we have also answered all our three research questions. The user feedback we received was extremely helpful in understanding core issues that our users faced and it really gave us key insights of how to improve the system in future. Since there was such an emphasis on users throughout this thesis, it was great to see how all the users gave us so much valuable feedback for every aspect of our system. We will not be repeating all of the conclusions from the individual sections here, rather, the next chapter will present an overarching set of conclusions for the entire thesis.

Chapter 8

Conclusions

In this thesis we have developed a learning-free novel music recommender system using contextual sensor data. Conventional approaches to music recommender systems look at either the user's past listening history or preferences based on which future recommendations are made. With these approaches, a great deal of emphasis is placed on accuracy as an important evaluation metric to measure new systems. This kind of an approach works well for users who only want to listen to more of what they already are currently listening to or have listened to in the past. However, for users that want to expand their music horizon and explore different types of music, existing approaches do not produce the level of novelty and serendipity that they would expect from their music recommender.

The goal of our thesis is to take a radical approach to music recommender systems where we say that in order to recommend novel music to users, we do not need any information about their past listening history or behaviors. In order to better understand the user's experiences at a given moment, we decided to use sensors present within our smartphones to infer the user's current context and use this information to retrieve novel music. After reviewing background literature in the areas of context-aware music recommender systems, the relationships between music and emotion and music retrieval techniques, we identified some knowledge and technology gaps that we wanted to fill through this research. One of the knowledge gaps was an over reliance on past user data to recommend novel music for users. We also did not find any solutions where the music recommender focused on streaming music and not offline music with a limited music dataset. We then set forth to establish some guiding principles that would help guide our design process—focus on streaming music, context-awareness, music discovery, learning-free and user experience.

These guiding principles helped us structure our design very clearly right from the beginning. We conducted a Wizard of Oz test with users to see if users could perceive the difference between random and contextual music recommendation. Even though there was a design flaw in the test, the user feedback gave us enough confidence to continue with the original idea because a majority of users perceived a difference between the randomly generated playlist and the contextual playlist. We started our design process and three months into the thesis we had developed our initial framework for the system architecture and then transitioned straight into system implementation. After the first version of the system was developed and all technical aspects of the thesis identified, we went back to our users in the form of a focus group study which gave us valuable feedback on how we should convey the contextual sensor data to users. Due to the structured and organized manner in which we had conducted the design process, there were no major hiccups during the system implementation except for the fact that it took a long time—unsurprising given the multi-faceted complexities of the system. The iOS we developed for recommending novel music based on contextual sensor data was deployed and evaluated by 22 users around the globe over a 10-day period.

Our first research question was designed to find out which application user interface was the most meaningful in terms of making a playlist choice. Users overwhelmingly (>90%) preferred viewing the playlist with tags and images together as opposed to images only or color only. They found that only with images it was too confusing to figure out the context type. For this question, only feedback from the survey was used to qualify the results. The second research question was to find out whether users preferred listening to the situation-based or the atmospheric-based playlist. We relied on data collected from user logs as well as qualitative survey feedback data. We found a 6% discrepancy in the results when users were asked which playlist they listened to the most and the data collected from the user logs. About 54% of users preferred listening to the situation-based playlist compared to the 23% for both the atmospheric-based playlist and the unsure group. The use of both user log data and qualitative data gives us confidence to say that users preferred listening to situation-based playlists more than atmospheric-based playlist. Our third research question was to find out if content-based re-ranking of songs was necessary or relevant for novel music recommendations. We used both user log data and qualitative user feedback from the exit survey to try and answer this question. Unfortunately, only 3% of the overall songs played during the evaluation period were analyzed so users did not get enough opportunity to decide whether content-based re-ranking made a difference. When users were shown a list of songs that were to be skipped by the applications, they mentioned that either they did not have enough information to answer the question or that they trusted the system to make the right music choices for them. In general, users appreciated that the application was trying to help them by curating their music based on preferences and situations. Even though we did not obtain a conclusive answer for this question, we think that content-based playlist re-ranking is not mandatory but definitely beneficial to the process.

We also found out from users that the application really pushed them to listening to music that they would not have discovered on their own. At the same time, users also said that sometimes the recommendations were too far away from what they would normally listen to and this caused them to go back to using other music applications during the evaluation period or skip quite a few songs. They really appreciated the auto-curation of playlists based on situations and that the user interface was so simple and easy to use. They would also have liked to know more about why the playlists were recommended and wanted to application to respond better to changes in the music dislike preferences. A positive observation was that about 60% of users continued to use the application even after the evaluation period ended and that shows that users did enjoy using the application to discover new music. In retrospect, we would have liked to have had a longer evaluation period and of course more users to test the application. We received a good deal of valuable feedback from our users and we thank them for testing out this application.

We set out initially to recommend novel music from contextual sensor data and in the process designed, developed and evaluated a full application that is almost ready for production. Our solid design principles allowed us to focus and scrap any approaches that did not adhere to these principles. We have shown that a context-aware music recommender system such as the one proposed here does not face the same cold-start issues that conventional music recommender systems face. We have also learned that designing with a clear set of design principles allows us to applying technologies and techniques from other fields of study. In our case, fuzzy logic context modeling was the perfect approach in hindsight to handle multi-modal sensor data with ease. Additionally, our system opens up the world up music discovery to users in today's realm of music recommender system where accuracy is still synonymous with overall user experience and allows users to really expand their music horizons. The next chapter details the implications of the results from our evaluation period and also identifies future research possibilities.

Chapter 9

Recommendations for future research

The beauty of the system proposed in this thesis is that it forms an umbrella for multiple future research initiatives, not just in multimedia computing but also in other faculties and disciplines. We will first consider the implications of the results from our research questions followed by other areas of future work resulting from this thesis. All the recommendations for future work in this section have been made to improve the current system from both system and user perspectives.

9-1 Implications of research questions on future research

Even though we answered our first research question by showing the user three different screenshots of possible views of contextual playlists that would help them make a choice as to which playlist they should play, we think there is room for more research in this area. We solved the problem of conveying the contextual information using images and tags but that does not mean that these are the most intuitive and informative ways from a user perspective. Our users told us that they wanted to know more of how their recommendations were generated. This means that our original question—“"What information shown on the user interface would convey a sense of context to users?" still remains an open area for future researchers and students in human-computer interaction or design.

Another potential area for research is to conduct a study for dynamic image retrieval techniques for different contexts in real-time. This question also arises from our users who mentioned that some of the pictures for the different playlists did not quite match the playlist tag. Future work in this area could involve matching different context with the best possible images while keeping the user informed and surprised. Similarly, future research could also be done on automatically generating contextual playlist tags so as to inform the user about the playlist in a fun yet informative manner to grab the user's attention. It could also be that the playlist tag gets automatically generated from the dynamically retrieved image thereby creating a fully automated model for informing the user with contextual information through images and tags.

Since we could not answer our third research question in its entirety, we think it still remains valid for future research. A suggestion would be to expose a Web-based API that services content-based requests for streaming URLs, thereby bypassing the issues that we faced using EchoNest in terms of API limits. With more songs being analyzed by such a Web service, a controlled user study can be conducted wherein we compare how users listen to music when an equal number of songs are recommended using music retrieval techniques such as tags and playlist titles, and through content-based filtering. We can then conclusively say whether or not content-based filtering is necessary for novel music recommendations. Until then we stand by our conclusion that it is not mandatory, but helpful.

9-2 Additional future research opportunities

In our current system, to maintain diversity while retrieving contextual music from SoundCloud, we handpicked 10-15 tags for each of the contexts and then select 5 tags at random to perform the SoundCloud query. Future research in this area could be to build up a personalized tag cloud as users listen to more and more music and then dynamically select social music tags from multiple sources (SoundCloud, Last.fm, web music blogs) to create a tag dictionary for the user. This information can then be visualized and shown to users so that not only can they can view their listening habits, but also explore more songs as their tag cloud expands. A caution should be made to distinguish tag queries between contexts or else all contexts will end up with a superset of the same tags. A suggestion would be to limit the number of tags per context but have they change dynamically based on the user's listening behaviors.

As we showed with our system, instead of modeling the user's genre preferences with what they like, we flipped it and asked user's what they do not like and this broadened the music space from where we could retrieve songs. Being averse to any form of machine learning when it comes to music discovery, we agree however that learning the user's dislike preferences (explicit or implicit based on skips and blacklist), could in fact be helpful in recommending songs that the system knows the user does not like given his dislike behaviors. This should be done with caution again otherwise we would end up in the same "filter bubble" as we do with current music recommender systems.

Another area where machine learning could be beneficial would be to learn the user's context but solely for the purposes of being able to predict the user's whereabouts accurately. The use cases for this would be to improve the user experience by prefetching playlists depending on the predictions of the user's context, thus reducing any wait times for the user. Another application of learning the user's context could be to suggest new playlists not currently supported by the system depending on the user's contextual history, thereby creating a different music experience for every user from playlists to songs.

Lastly, during the analysis from the user log data, we developed three evaluation metrics that we think will be good measures of judging such a system's user experience—'Favorites To Unique Songs Played Ratio (FSR - our system is currently at 4%)', 'Blacklist To Unique Songs Played Ratio (BSR - our system is currently at 17%)' and 'Songs Skipped Ratio (SSR)'. We would ideally like to see the FSR ratio increase and BSR and SSR decrease. It would be interesting to see some research done in this area of new user-facing evaluation metrics instead of the traditional approaches such as measuring novelty, accuracy, diversity because they do not take user interactions into account. After all, the recommendations can be very novel or extremely accurate but if the user skips the song every time it is recommended, these metrics make no difference to the overall user experience. Thus a very interesting area of research would to come up with new metrics to evaluate music recommender systems from the user perspective, and not simply from the system perspective.

Appendix A

Appendix A: Conference paper presented at LocalRec 2015 Workshop, ACM RecSys Vienna

From Sensors to Songs: A learning-free novel music recommendation system using contextual sensor data

Abhishek Sen, Martha Larson
Intelligent Systems
Delft University of Technology, Netherlands
a.sen@student.tudelft.nl, m.a.larson@tudelft.nl

ABSTRACT

Traditional approaches for music recommender systems face the known challenges of providing new recommendations that users perceive as novel and serendipitous discoveries. Even with all the music content available on the web and commercial music streaming services, discovering new music remains a time consuming and taxing activity for the average user. The goal for our proposed system is to provide novel music recommendations based on contextual sensor information. For example, contextual place information can be inferred with intelligent use of techniques such as geo-fencing and using lightweight sensors like accelerometers and compass to monitor location. The inspiration behind our system is that music is not in the past, neither in the future, but rather enjoyed in the present. For this reason, the system does not rely on learning the user's listening history. Raw sensor data is fused with information from the web, passed through a cascade of Fuzzy Logic models to infer the user's context, which is then used to recommend music from an online music streaming service (SoundCloud) after filtering out songs based on genre preferences that the user dislikes. This paper motivates and describes the design for a mobile application along with a description of tests that will be carried out for validation.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search, Retrieval and Filtering

Keywords

Context-aware, Music recommender systems, Fuzzy logic, Sensor data fusion

1. INTRODUCTION

Music plays a central role in the daily lives of many people. Today, music streams are readily available through services such as YouTube, Spotify and Apple Music. However, the process of discovering music remains a tedious and unpleasant experience. With a general trend towards streaming music as opposed to downloading music, many music services aim to learn user music tastes and listening behaviors to provide personalized music recommendations. The assumption of the learning approach, namely, that past behavior is a good predictor of future behavior, is certainly not ill founded. Such services will certainly satisfy users looking for a highly predictable music experience. However, users interested in expanding their music horizons will not be satisfied by algorithms that rely on previous listening history or preferences (artist/genre), since they do not support new music discovery. Such algorithms fail to provide the serendipity that is extremely important for users to discover music that is new, but is also not completely alien to them. Instead, to design the system presented here, we make a new assumption. We consider music listening to be independent of the past (history) or the future (prediction) and instead consider it as a function of the present (current context). We use the term *context* to refer to the sum of a user's experience at a given moment, including place, surroundings, activities that the user is currently pursuing and atmospheric effects on the user's mood. We assume that listeners have similar expectations of which music fits a particular context. We rely on the idea that this collective conception of 'music that fits a moment' will provide users with a sense that the recommendations of our system fit their current needs, and at the same time allow them to discover music that they would not have otherwise found themselves.

2. RELATED WORK

There is a large volume of prior research in the field of context-aware music recommender systems (e.g., [2], [10], [7]). Bonnin and Jannach present a comprehensive literature survey on automated playlist generation and categorize existing approaches in [1]. They mention the importance of context in automatic playlist generation and also how similarity-based algorithms are an obvious approach when the system's goal is to maximize the homogeneity of the playlist. As a downside, serendipity and diversity are negatively affected since most songs recommended will be of a similar type, i.e., the same with respect to artist or genre. One of their core recommendations for future research is to assess multiple criteria at the same time and explore the trade-offs between homogeneity and diversity of playlists. Our system, explained further in Section 5, addresses these recommendations by balancing diversity and homogeneity and does not rely on learning the user's past listening be-

LocalRec'15, September 19, 2015, Vienna, Austria.

haviors.

In [10], Wang et. al propose a system that is context-aware, probabilistic and learns the user's listening habits over time for better recommendations. Their system utilizes contextual sensor data and integrates this information with music content analysis to provide relevant music recommendations per context. However, the study requires the musical signal of the songs to be pre-analyzed by music analysis and was also evaluated with offline music. In the version presented in this paper, our system instead focuses on music metadata that is directly available and does not rely on learning the user's listening behaviors. Okada et. al present a system in [8] that focuses on the user interface aspects of context-aware music recommender systems, an area often ignored by researchers. One of their core objectives is to explore how context plays a key role in a user's listening behavior and how this information can be conveyed to the user. In the next sections, we will see how this prior work inspired key design choices in our system.

3. DESIGN CONCEPT

Our main design concept is—as the title states—from sensors to songs. We want to recommend novel music to users by inferring their context from sensory data. To achieve the desired surprise and delight factor, the system should not have to learn the user's music tastes and listening behavior. We believe this non-learning characteristic of the system to be, currently, a quite radical approach to music recommendation. It allows users to discover new music continually without any impediments, such as the need to interact frequently with the system. Through inference of user preferences based on collection-wide user experiences of context, we think the system will achieve a level of personalization that is ideal for music recommender systems—without the need to learn everything about the user's listening history.

We are aware that user music preferences are also highly personal. However, instead of making the assumption that music recommendation is "all about personalization", our system strives to integrate "minimum necessary personalization". We do this in two ways. First, we rely on the idea of the context as mentioned above. The situations in which users find themselves can be expected to reflect their lifestyles and overall music preferences for places and activities. A system like ours that relies on collective music preferences of users for specific contexts, is actually providing a level of personalization, albeit indirectly. Second, we allow users a minimum degree of control, e.g., in excluding songs from genres that the user dislikes.

4. DESIGN METHODOLOGY

Inspired by the design concept, our system focuses on providing novel music recommendations with an emphasis on incorporating contextual user information. Our design methodology aims to inform the possibilities for a sensor-based music recommender, with a user centered approach. The goal of sensors embedded within any device is to 'sense' the environment for information such as temperature, acceleration etc. This inherent capability of sensors makes them an ideal choice for use in interpreting user context, especially since most users carry 'smart' devices such as smartphones close to them at all times. This allows the system to respond to major context changes implicitly without requiring any user action. It is also important for our system to be lightweight and run efficiently and not drain the device's battery during normal usage.

Given that the context inference might not be perfect due to 'noisy' sensory data, we want to give the user a choice of playlists. As discussed in our design concept, to exploit the communal behaviors of music listening across different contexts, we will generate contextual tags to retrieve music from SoundCloud[1]. Knees and Schedl [5] discuss tags as a form of text-based approach given their community-based characteristics. SoundCloud has a music database of over 100 million songs, which are richly annotated with tags. Tags of a track that are related to the context provide us with evidence that listeners generally associate the track with that context. Contextual tag-based queries then allow us to retrieve songs from SoundCloud that both, fit contexts and allow users to discover new music.

We conducted an intensive focus group study with 6 Master's students from different faculties at the Delft University of Technology and the feedback gave us key insights for our design process. All of them described music discovery as a tedious and challenging activity even with all the music available on the web. They described their ideal music recommender system would know which song to play for any given situation and not just based on their past history.

One of their main complaints about current music recommender systems was that most systems tend to repeat the same type of songs unless the user has explicitly made a different selection. They were also of the opinion that even though such a system might provide 'bad' recommendations at times, they would simply move on to the next song and continue listening. This insight suggested that our system does not have to infer the user context perfectly and that we could hedge our predictions by providing the user a choice of playlists for the most likely contexts. The group also mentioned that they all had different music tastes and each had their own music preferences for different contexts—this led us to include a genre preferences block as shown in Figure 1 so that in addition to knowing what the user enjoyed listening to of late, more importantly, the system "knows" the kind of music the user really does not enjoy hearing.

5. PROPOSED SYSTEM

The proposed system architecture as shown in Figure 1 is the materialization of our design concept, methodology, and the focus group feedback. The system is divided into three main components: context inference, music retrieval/analysis and music recommendation.

5.1 Context Inference

Context inference as shown in Figure 1 is done by fusing sensor data and passing it through fuzzy logic models.

5.1.1 Sensors

Table 1 shows a list of contextual information categories and the sensors used for their inference. All the sensors used in the system are embedded inside most smartphones and this trend is likely to continue with future 'smart' devices such as smartwatches and other wearables. The system is scalable and additional sensors can be easily integrated to further improve the context inference process.

[1] https://developers.soundcloud.com/docs/api/reference

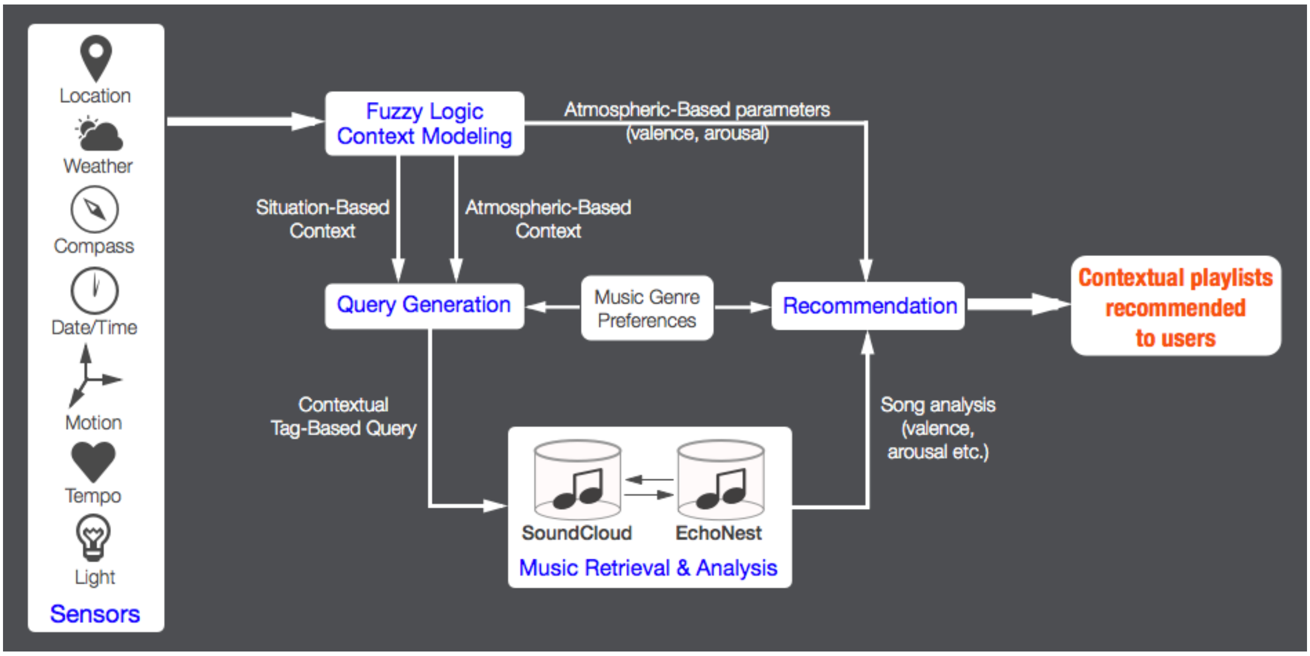


Figure 1: Proposed playlist generation system architecture: Sensor-based novel music recommender

Table 1: Contextual Information

Category	**Sensors**
Location	Wifi, GPS, Accelerometer, Compass, Cellular
Indoor/Outdoor	Compass, Light
Activity	Accelerometer, Gyroscope
Date/Time	System Clock
Weather	Temperature, Humidity, Pressure, Sunshine

5.1.2 Fuzzy Logic Context Modeling

Using fuzzy logic for context inference makes the system extremely flexible and easy-to-understand, and allows it to process imprecise sensor data with ease. Motivated by our non-learning design concept, fuzzy logic makes it possible to translate user-supplied human language rules into mathematical values that can be used for making decisions, thus making the system logic easily understandable. Given the computational challenges of fusing multi-modal sensor data, fuzzy logic provides an extremely light-weight and efficient technique. The Fuzzy Logic Context Modeling block comprises two main internal models as shown in Table 2.

Table 2: Fuzzy Logic Context Models

Category	**Inputs**
Atmospheric-Based	Temperature, Humidity, Pressure, Sunshine
Situation-Based	Activity, Day of week, Time of day, Indoor/Outdoor, Place

5.1.3 Atmospheric-Based Context Model

The Atmospheric-based model generates values for valence and arousal based on prior psychology research on the impacts of different weather factors on people's mood (e.g., [3], [4]). Weather information is integrated into the system using Yahoo API[2]. Mood is a very difficult characteristic to judge on a personal level—especially since everyone's mood could be influenced by a multitude of factors. For this reason, we decided to use the most important weather condition factors that are thought to most universally affect people in a certain geographic area to get a rough estimate of which quadrant of Russell's widely accepted circumplex model of affect the user might be in [9]. The objective here is not to accurately determine the user's mood but to get a general idea depending on the impacts of weather on their mood.

5.1.4 Situation-Based Context Model

For the situation-based context model, the focus group results informed us of the most common situations in which participants listen to music and we chose to pick the top 7 for our system: waking up, commuting, working/studying, exercising, relaxing, housework and sleeping. To determine the situation, we use fuzzy rules such as the following:

IF *Activity* IS *Stationary* AND *DayOfWeek* IS *Weekday* AND *TimeOfDay* IS *Afternoon* AND *Indoor/Outdoor* IS *Indoor* AND *Place* IS *Office* THEN *Context* IS *Working or Studying*

The activity states that our system identifies are stationary, walking, running and driving. These activity states are provided by the iOS platform. To accurately distinguish between the stationary and driving state, we utilize GPS to get the user's speed and make a decision accordingly. The indoor/outdoor sensor inputs to this model determine whether the user is indoors or outdoors using sensors such as light and compass and is adapted from Zhou et. al's proposed system in [11]—we do not use cellular signal strength in our system due to lack of development support on iOS.

[2] https://developer.yahoo.com/weather/

Our system is currently able to identify five general place categories for users—home, office, library, gym and other. These areas are recognized without the user having to explicitly enter information. The system monitors significant location updates and marks any visited locations as possible candidates for any of the above five places in a two-step process.

First, using the Foursquare Venues API we reverse geocode the location's coordinates to the library or gym place categories. If no results are returned, the visit information is then passed through an internal fuzzy model to determine the home and office place categories based on fuzzy rules. Once a place has been annotated with a category (not always), the system sets up a geofence around it for a specified radius. From this point on, any time the user enters or leaves this place, a place context change event is triggered and the user's context is recomputed by processing all the other sensory inputs as shown in the Situation-based Context Model in Table 2. If a change in user context is detected, a new contextual song query is formulated to request a new set of songs from SoundCloud. The proposed technique of monitoring places ensures that we do not drain the smartphone's battery by only using the GPS when needed.

5.2 Music Retrieval & Recommendation

Once the user's context has been analyzed, the next step is to retrieve songs from SoundCloud based on this information. The system performs query expansion using Last.fm APIs[3] to translate the fuzzy model's output into query tags. For example, for the 'Exercising' tag, Last.FM returns a set of similar tags such as 'fitness', 'workout' and 'motivation'. These results are aggregated and the top-ten tags are used by the system for this context. For the weather-to-mood tag generation process, social mood tags from [6] were used as tag seeds for the query expansion. Through this process of query expansion and the ever-evolving music community on SoundCloud, the chances of retrieving novel music is very high.

Post song retrieval, the system then performs music content analysis using the EchoNest APIs[4] for each of the retrieved songs by looking at parameters such as energy and valence. However, since each song is analyzed at runtime by uploading the tracks from SoundCloud to EchoNest, this process takes some time and increases as the song duration increases. For this reason, this analysis is done in the background while the user is listening to music. After the songs have been retrieved from SoundCloud and/or EchoNest, the system then generates situational-based and atmospheric-based playlists. It is important at this step that the user identifies the subtle differences between the recommended playlists and we plan on using visual imagery and colors to convey the contextual differences to the user.

6. OUTLOOK & CONCLUSIONS

The system is currently under development as a mobile application on the iOS platform on iPhone models 5S and newer running iOS 8[5]. The evaluation plan is to have about 20-25 participants use the application for a week and answer the following research questions: "What information shown on the user interface would make the music recommendations transparent for users?", "Which modality (mood-based-context or activity-based-context) influences the user more?", "Is the content-based re-ranking for song relevancy necessary for recommendations?".

Future work in this topic includes a number of challenges such as removing the hard-coding of contextual tags and making the tag generation process dynamic. Other alternatives would be to include playlist titles and tracks within the recommendations for playlists. Our design concept and motivations for this system however remain the same—to expand the musical horizons of users while making the music discovery process less tedious and more serendipitous.

7. ACKNOWLEDGEMENTS

The contribution of the second author was funded in part by CrowdRec (EC FP7 Project 610594).

8. REFERENCES

[1] G. Bonnin and D. Jannach. Automated generation of music playlists: Survey and experiments. *ACM Comput. Surv.*, 47(2):26:1–26:35, 2014.

[2] D. Griffiths, S. Cunningham, and J. Weinel. A discussion of musical features for automatic music playlist generation using affective technologies. In *Audio Mostly Conference*, AM '13, 2013.

[3] A. Hannak, E. Anderson, L. F. Barrett, S. Lehmann, A. Mislove, and M. Riedewald. Tweetin' in the rain: Exploring societal-scale effects of weather on mood. In *International AAAI Conference on Weblogs and Social Media*, ICWSM '12, 2012.

[4] E. Howarth and M. S. Hoffman. A multidimensional approach to the relationship between mood and weather. *British Journal of Psychology*, 75(1):15–23, 1984.

[5] P. Knees and M. Schedl. A survey of music similarity and recommendation from music context data. *ACM Trans. Multimedia Comput. Commun. Appl.*, 10(1):2:1–2:21, 2013.

[6] C. Laurier. *Automatic Classification of musical mood by content-based analysis.* PhD thesis, Universitat Pompeu Fabra, 2011.

[7] S. Nirjon, R. F. Dickerson, Q. Li, P. Asare, J. A. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao. Musicalheart: A hearty way of listening to music. ACM SynSys '12, 2012.

[8] K. Okada, B. F. Karlsson, L. Sardinha, and T. Noleto. Contextplayer: Learning contextual music preferences for situational recommendations. In *ACM SIGGRAPH '13*, 2013.

[9] J. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161–1178, 1980.

[10] X. Wang, D. Rosenblum, and Y. Wang. Context-aware mobile music recommendation for daily activities. In *ACM Multimedia '12*, 2012.

[11] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen. Iodetector: A generic service for indoor outdoor detection. In *ACM SenSys '12*, 2012.

[3] http://www.last.fm/api
[4] http://developer.echonest.com/docs/v4
[5] https://developer.apple.com/library/ios/navigation/

Appendix B

Appendix B: ImpliciTunes Application User Interface

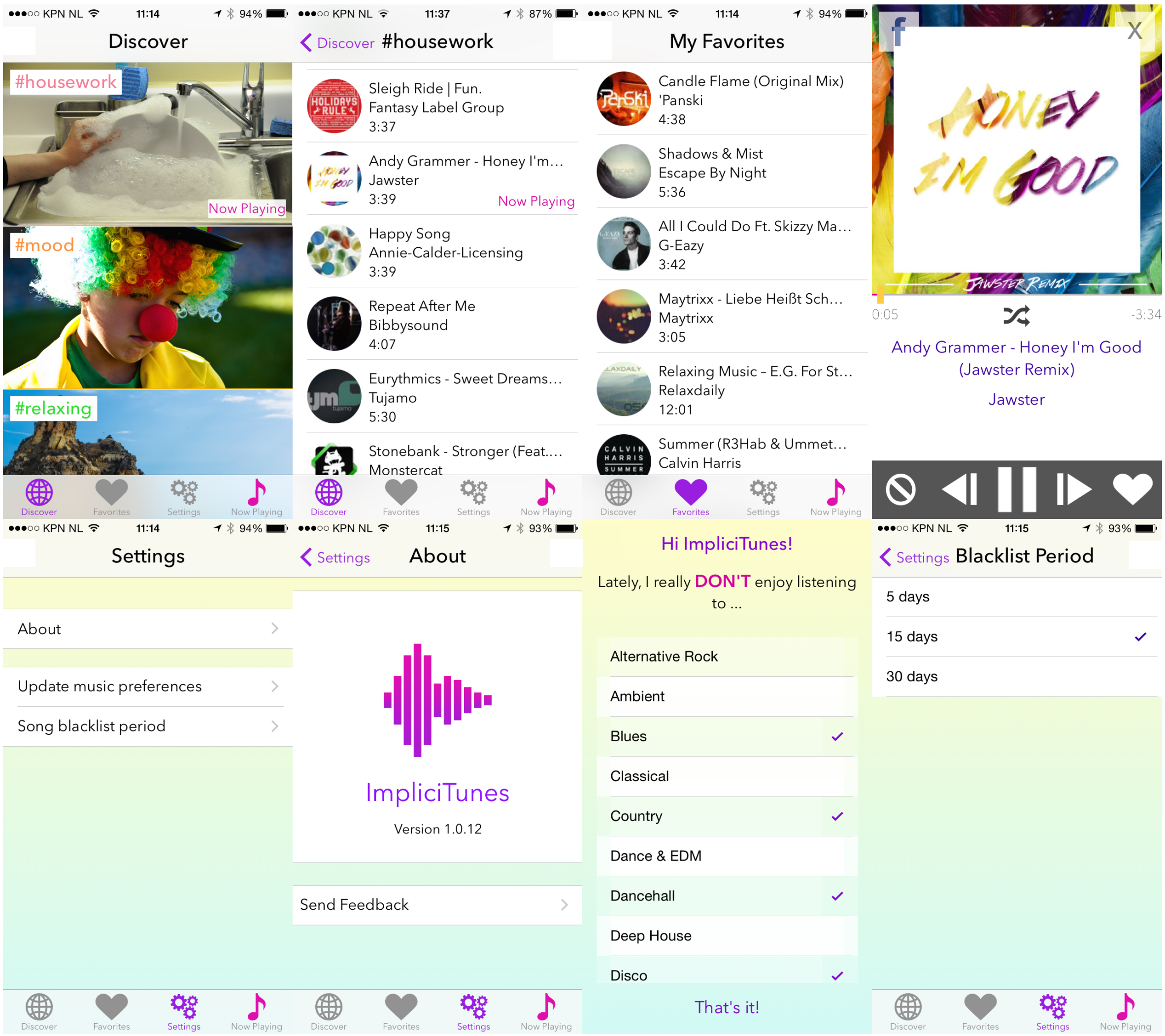


Figure B-1: ImpliciTunes application user interface screenshots

Appendix C

Appendix C: Situational Playlist Images

Figure C-1: Images used to convey contextual information for the situational playlists

Appendix D

Appendix D: Atmospheric Playlist Images

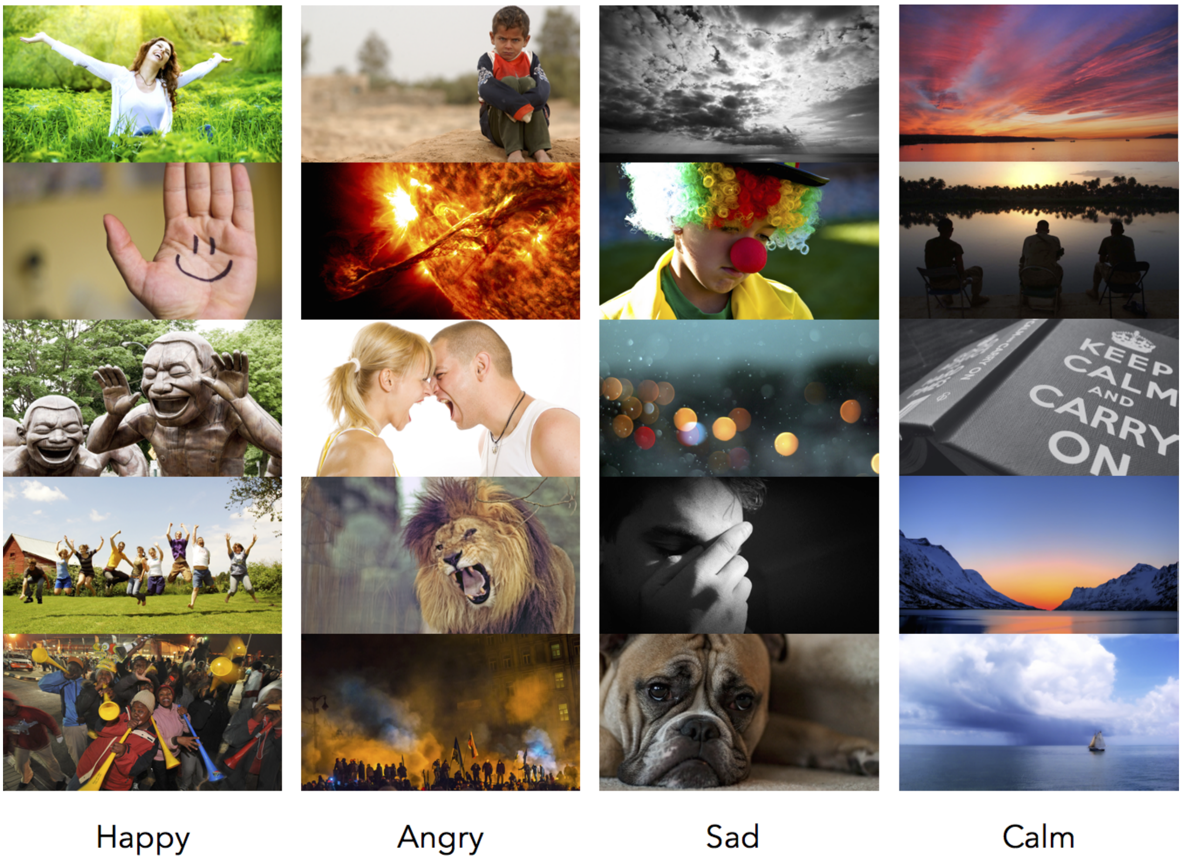


Figure D-1: Images used to convey contextual information for the atmospheric playlists

Appendix E

Appendix E: Prototyping the pulse sensor

We started this thesis on January 5, 2015 and at the time, Apple had already made an product announcement for the Apple Watch[1]. The Apple Watch was slated to be released on April 25, 2015. We were quite excited about the possibilities of being able to use the on-device sensors (accelerometer, GPS, pulse sensor) in order to model the user's movements and activity. The pulse sensor in particular was of interest to us because we could measure the user's heartbeat information and use it in our system for music recommendations. To prototype the pulse sensor, we developed a test iOS application using an Arduino Uno[2], an Arduino compatible pulse sensor[3], a Bluetooth Low Energy Shield from RedBearLab[4] and an iPhone 5S to display the pulse sensor readings sent from the Arduino to the smartphone over Bluetooth. The readings were a bit noisy which meant that we would have had to do some additional filtering to smooth out the raw sensor data. Figure E-1 below shows a version of the implemented prototype to extract the heartbeat signals from the pulse sensor.

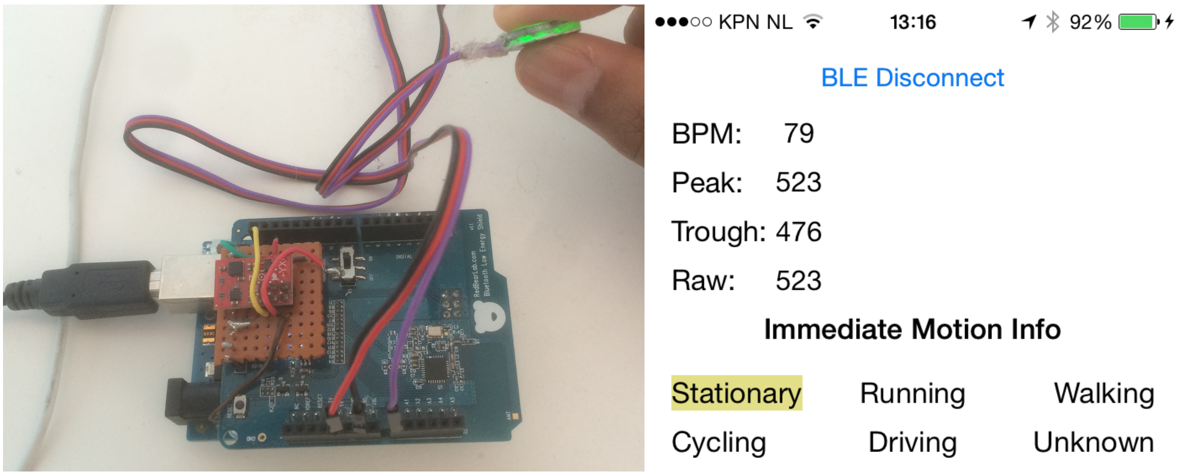


Figure E-1: Pulse sensor reading from Arduino over Bluetooth on test iOS application while sitting stationary

Unfortunately, at the launch of Apple Watch, the first version of the software development kit (SDK), access to the on-device sensors on the watch was restricted and not publicly available. This meant that we could not build our application and have it compatible on the Apple Watch. Given the nascent smartwatch market (in comparison to the massive smartphone market), this setback was actually good because we could then focus on sensors that were already present in devices of a majority of users. We shelved our idea to use the pulse sensor as a means to get the user's heartbeat and decided to readdress it in future when the developer kit opened up access to the on-device sensors on the Apple Watch. No further data readings were taken from the pulse sensor prototype.

[1] http://www.apple.com/watch/
[2] https://www.arduino.cc/en/Main/ArduinoBoardUno
[3] http://pulsesensor.com
[4] http://redbearlab.com/bleshield/

Appendix F

Appendix F: Online Music Source Survey

Online streaming music source

* Required

1. **Your smartphone platform** *

   Mark only one oval.

   - iOS
   - Android
   - BlackBerry
   - Windows Mobile
   - Other: ____________

2. **How do you listen to music?** *

   Check all that apply.

   - Download
   - Stream
   - Other: ____________

3. **Where do you listen to music the most?** *

   Mark only one oval.

   - Laptop/Desktop
   - Smartphone
   - Radio
   - Other: ____________

4. **Do you pay for your music? (Subscription or from an online music store)** *

   Mark only one oval.

   - Yes
   - No
   - Used to

5. **Which of these music streaming services do you use currently?** *

   (Please check all that apply. If you have multiple others, please add them with commas)

   Check all that apply.

   - 8tracks
   - Amazon MP3 Store
   - Anghami

- [ ] Aupeo
- [ ] Beats Music
- [ ] Deezer
- [ ] Dhingana
- [ ] Earbits
- [ ] Fizy
- [ ] Grooveshark
- [ ] Google Play Music All Access
- [ ] Guvera
- [ ] iTunes Radio
- [ ] iTunes
- [ ] Jango
- [ ] Xbox Music
- [ ] Zune Marketplace
- [ ] Mflow
- [ ] MOG
- [ ] Mixcloud
- [ ] Musicovery
- [ ] Myspace (music)
- [ ] Napster
- [ ] Pandora
- [ ] Last.fm
- [ ] Rdio
- [ ] Rhapsody
- [ ] Saavn
- [ ] Simfy
- [ ] Simfy
- [ ] Slacker Radio
- [ ] Songza
- [ ] Spotify
- [ ] Sony Music Unlimited
- [ ] SoundCloud
- [ ] Thumbplay
- [ ] TuneIn Radio
- [ ] We7
- [ ] rara.com
- [ ] WiMP

- [ ] Kollekt.FM
- [ ] Youtube
- [ ] Ubuntu One Music
- [ ] Other: ____________________

Appendix G

Appendix J: Wizard of Oz Feedback Survey

Woz SuperTracklist Feedback

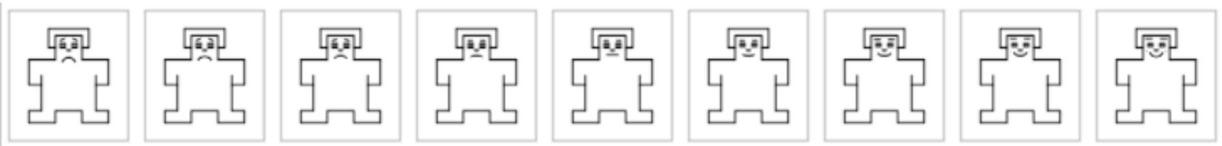
* Required

Playlist1

Tracks in playlist1:

Dj Cam - Wayfarer
youngBARON - oneTHIN //prod. drip-133
Back To The Money (Prod. By M-Press Live)
06 - Time Machine
Eventually (Prod.Topdxg)
Migos - Charles Barkley (prod. Zaytoven)

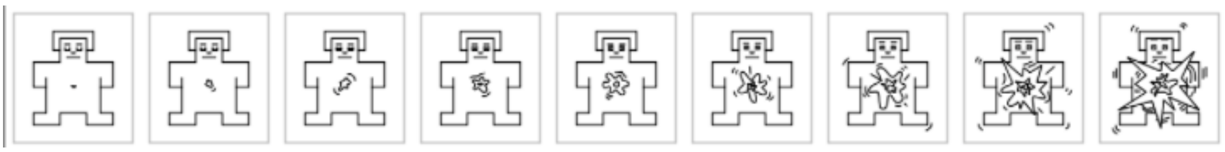
Valence - Feelings that range from completely unhappy or sad (on the left) to completely happy or elated (on the right).



1. **How do you rate your valence during playlist1?** *
   Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Completely Unhappy/Sad	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely Happy/Elated

Arousal - Feelings that range from very calm or bored (on the left) to very stimulated or involved (on the right).



2. **How do you rate your arousal during playlist1?** *
   Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Very Calm/Bored	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very Stimulated/Involved

3. **How well did the tracks in playlist1 'match up' to what you were doing when you listened to playlist1?** *
Mark only one oval.

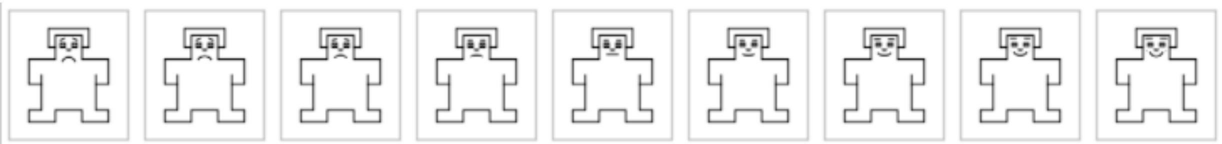
	1	2	3	4	5	6	7	8	9	
Completely mismatched	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely matched

Playlist2

Tracks in playlist2:

Henri Pfr & Ofenbach - Don't Worry Be Happy !
Sons Of Maria - You & I (Original Mix)
Cee-Roo - I'm So Happy
Marv & Philipp Dittberner - Wolke 4 (Hagen Stoklossa Bootleg)
Sebastian Ingrosso | Kidsos (Kiddcat Remix)
Unmissable (Le P Remix) - Gorgon City

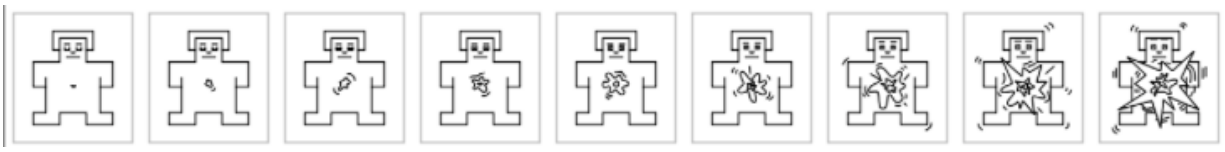
Valence - Feelings that range from completely unhappy or sad (on the left) to completely happy or elated (on the right).



4. **How do you rate your valence during playlist2?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Completely Unhappy/Sad	◯	◯	◯	◯	◯	◯	◯	◯	◯	Completely Happy/Elated

Arousal - Feelings that range from very calm or bored (on the left) to very stimulated or involved (on the right).



5. **How do you rate your arousal during playlist2?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Very Calm/Bored	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very Stimulated/Involved

6. **How well did the tracks in playlist2 'match up' to what you were doing when you listened to playlist1?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Completely mismatched	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Completely matched

Preferences & Differences

7. **Which playlist did you prefer listening to?** *
Mark only one oval.

⬭ Playlist 1
⬭ Playlist 2
⬭ Both
⬭ Neither

8. **Did you perceive any difference between playlist1 and playlist2?** *

....
....
....
....
....

9. **What was the context/situation of when you listened to the music?** *
Going from home to school, library to the gym etc.

....
....
....
....
....

10. **What was your mode of transportation?** *

Mark only one oval.

- ⬭ Bike
- ⬭ Walk
- ⬭ Bus
- ⬭ Train
- ⬭ Car
- ⬭ Motorcycle
- ⬭ Other: ____________

11. **What was the weather like when you listened to the playlists (in a few words)?** *

Sunny, raining, cold ...

12. **What other situations/contexts could you think of where you would've liked to listen to the right type of music?**

13. **Any general comments, improvements and/or suggestions for the app?**

Appendix H

Appendix H: Focus Group Session Plan

ImpliciTunes Focus Group Session Overview

Number of participants: 6
Study Duration: 2 hours
Location: TU Delft Library, Steve Jobs
Time: Monday, May 4, 2015, 17:00
Equipment: Post-Its, Markers, Pen, Paper, Phone Screen Template

1. **Introduction & Project Overview (3-5 minutes)**
    a. About Me
    b. Brief Project Description - we're going to talk about music!
    c. User Agreement Form Signature
    d. Rules:
        i. Be respectful of everyone's opinions
        ii. Write/draw anything you don't feel comfortable sharing or if it's easier to express on paper
        iii. My role as a moderator - Guide the conversation and provide brief inputs and/or ask for opinions once in a while, but mostly it'll be you'll discussing
        iv. Very informal so have fun!

2. **Music Recommender Services Divergence exercise (3 minutes)**
    a. Write on post-its everything and anything that comes to your mind when you see these pictures of music streaming services: *Spotify, iHeartRadio, iTunes Radio, SoundCloud, Pandora, Your Favorite Music Player*

3. **I: What is your experience of using a music recommender system? (10 minutes)**
    a. Under what situations is listening to 'new' music important to you?
    b. What is your opinion on listening to music from up-and-coming artists as opposed to major recording artists/labels?
    c. **OBJECTIVE: Gather user information on what could potentially be used as benefits/advantages of my proposed model (novelty, pros/cons of existing systems), also priming them for future sections**

4. **II: If sensors (accelerometer, GPS, compass) are going to be used to recommend your music, what would you expect from such a service? (15-20 minutes)**
    a. What things should be taken into consideration?
    b. What do you currently do if you receive an incorrect or out-of-context track recommendation? How do you feel?
    c. How would this behavior change if your music is now being recommended using sensors? If at all …
    d. How important is context or situation with regards to recommending the right music for you?

e. **OBJECTIVE: Design decisions for 30-day block period in case of an incorrect recommendation, key design choices as to when to trigger a recommendation based on context change, model learning behavior implementation (if any), logging and user feedback design choices and any other design red flags**

5. **III: Given a list of music playlist recommendations using sensors, which playlist(s) would you select and why? (15-20 minutes)**
    a. What visual indicators would help you make this decision?
    b. How important is it to know why this playlist was recommended for you?
    c. Would you prefer to be notified about a prospective change in context based on which the playlist choices will change? Or would you prefer the system to do this automatically without any notification? Why?
    d. Would you rather prefer 1 consolidated playlist every time the context changes or multiple playlists?
    e. With other applications such as Netflix, Amazon, Bol, what factors help you make a decision whether or not to buy or start using/watching an item? Do you ever rate items, write reviews and post any form of feedback for purchased items? Why or why not?
    f. **OBJECTIVE: How to convey contextual information to users, whether they prefer playlist or track recommendations, auto-pilot vs manual mode for recommendation control, how do they like to provide feedback (if any)**

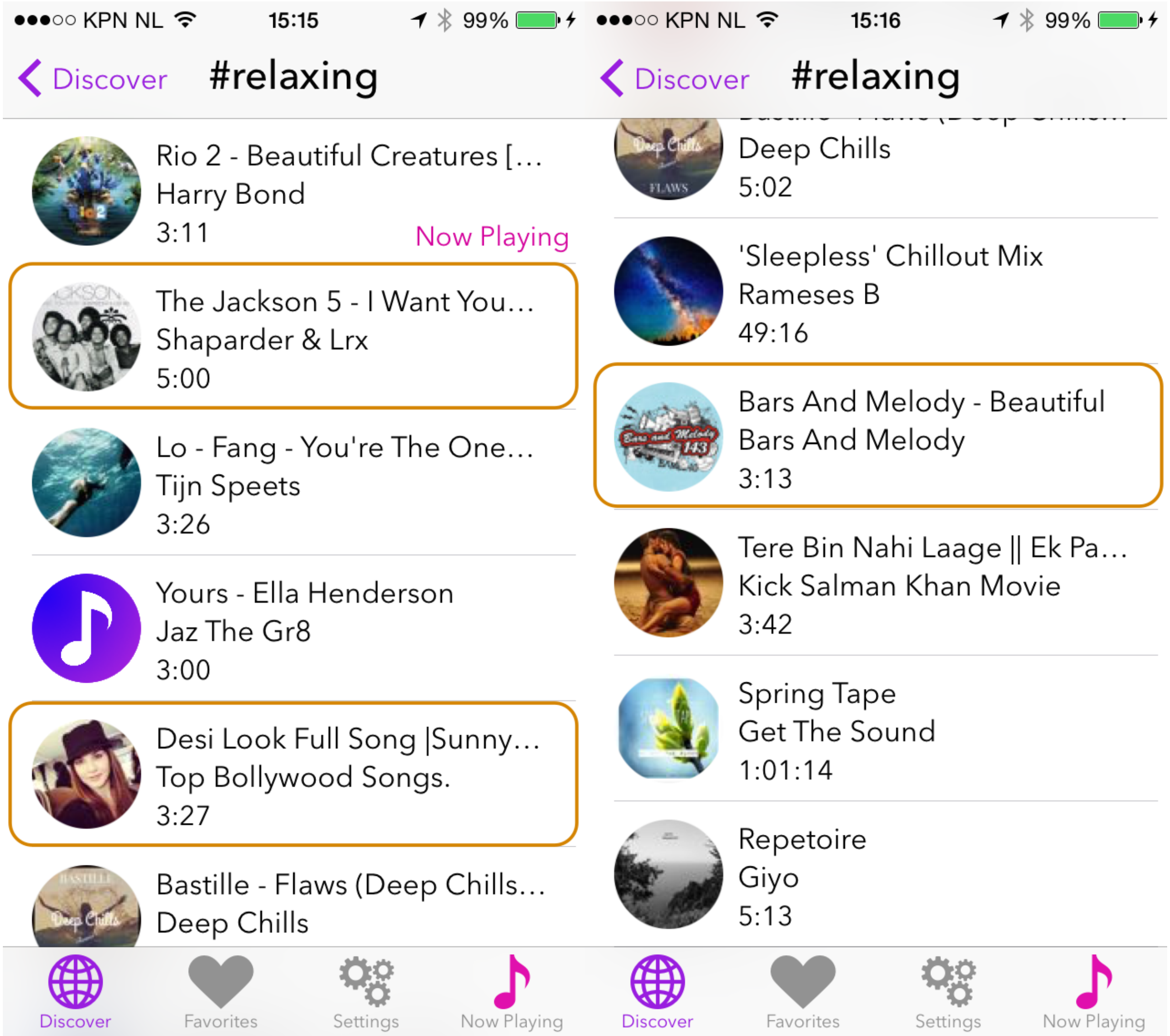
6. **IV: Concept generation for a sensor-based smartphone music application (groups) - How would you convey contextual information to the user to help him/her in selecting their music? Think of the why in your design. (10 minutes)**
    a. Present your interface to the rest of the members to improve and/or receive feedback
    b. Quality over quantity
    c. Groups: Osman/Luis, Sara/Viki, Sascha/Matthijs
    d. **OBJECTIVE: User interface concept design**

Appendix I

Appendix I: Mid-Test User Evaluation Survey

ImpliciTunes Mid-Test Feedback

* Required



1. **1. If you were listening to the #RELAXING playlist and the songs shown in the yellow boxes above WERE SKIPPED FROM BEING RECOMMENDED BY THE SYSTEM, you would be** *

   Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Very displeased	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very pleased

2. **2. What made you pleased/displeased by the system's decision to remove these song choices as shown in the yellow box above from the #relaxing playlist?** *
Your answer should explain why you made the choice you did in question 1

3. **3. Are there any other songs from the above playlist that you've would've rather preferred the system to skip? Why?**
Please list the song titles below (If any)

Application Usage Details

4. **4. So far, which of the following playlists have you listened to (even once)?** *
Check all that apply.

- [ ] Wake Up
- [ ] Commuting
- [ ] Working
- [ ] Exercising
- [ ] Relaxing
- [ ] Housework
- [ ] Sleeping
- [ ] Mood

5. **5. Have you discovered any new music in the last 5 days of testing?** *
Mark only one oval.

- ( ) Yes
- ( ) No

6. **6. Have you 'favorited' any tracks thus far?** *

Mark only one oval.

- Yes
- No

7. **7. Any interim comments/questions/suggestions regarding the application?**

8. **8. Your e-mail address** *

Appendix J

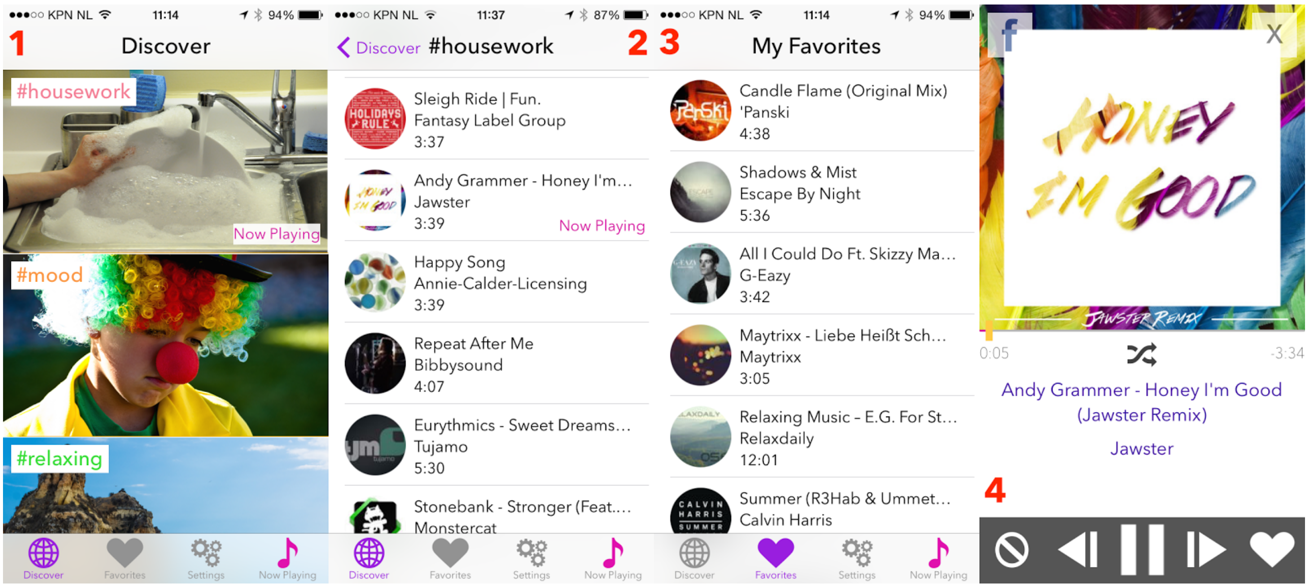
Appendix J: Exit User Evaluation Survey

ImpliciTunes Final Exit Survey

This is the final exit survey for the testing phase. The survey should take approximately 15-20 minutes and it is very important that you answer these questions as accurately as possible. Please don't be reluctant to answer frankly :)

* Required

User Interface



1. **Were any user interface elements shown in the images 1-4 unintuitive and/or confusing for you? If so, please explain? ***

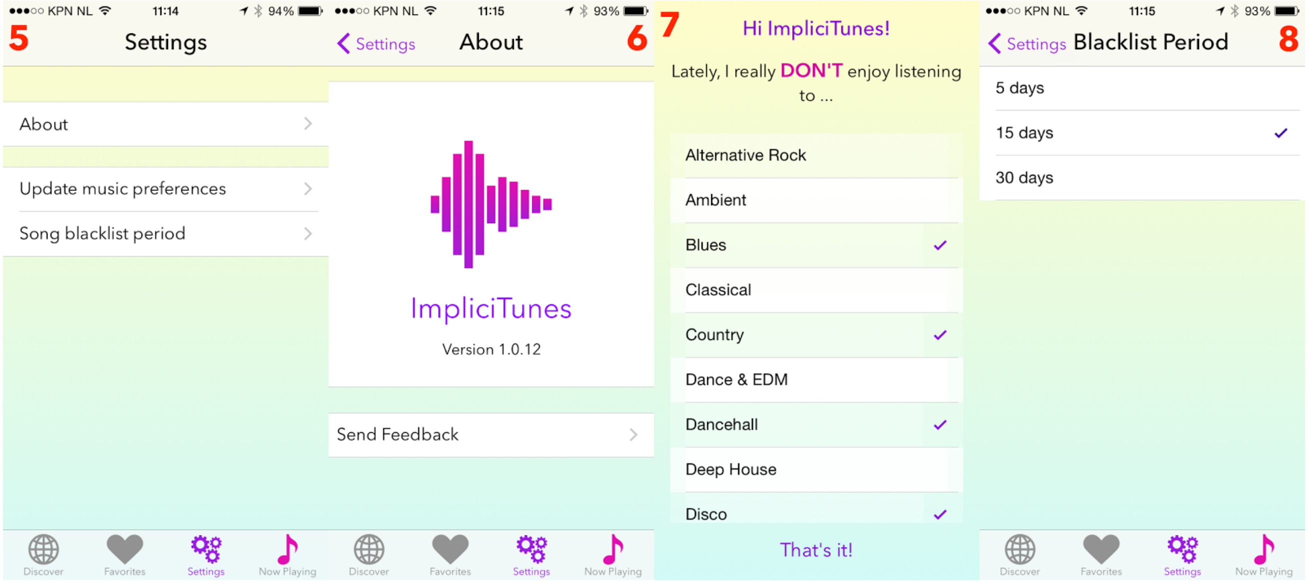
....

....

....

....

....



2. **Were any user interface elements shown in the images 5-8 unintuitive and/or confusing for you? If so, please explain? ***

....

....

....

....

....

3. **How meaningful were the playlist tags? ***

#wake up, #commuting, #working, #relaxing, #exercising, #housework, #sleeping, #mood

Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Did not make sense	◯	◯	◯	◯	◯	◯	◯	◯	◯	Made perfect sense

4. **If any playlist tag did not make sense, please explain.**

....

....

....

....

....

5. **Would you have preferred a different text for any of the playlist tags? If yes, please explain.**

....

....

....

....

....

6. **How closely related were the playlist tags to the playlist images?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Completely unrelated	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very close related

7. **If there were any specific examples of playlist tag and image mismatch, please comment on it.**

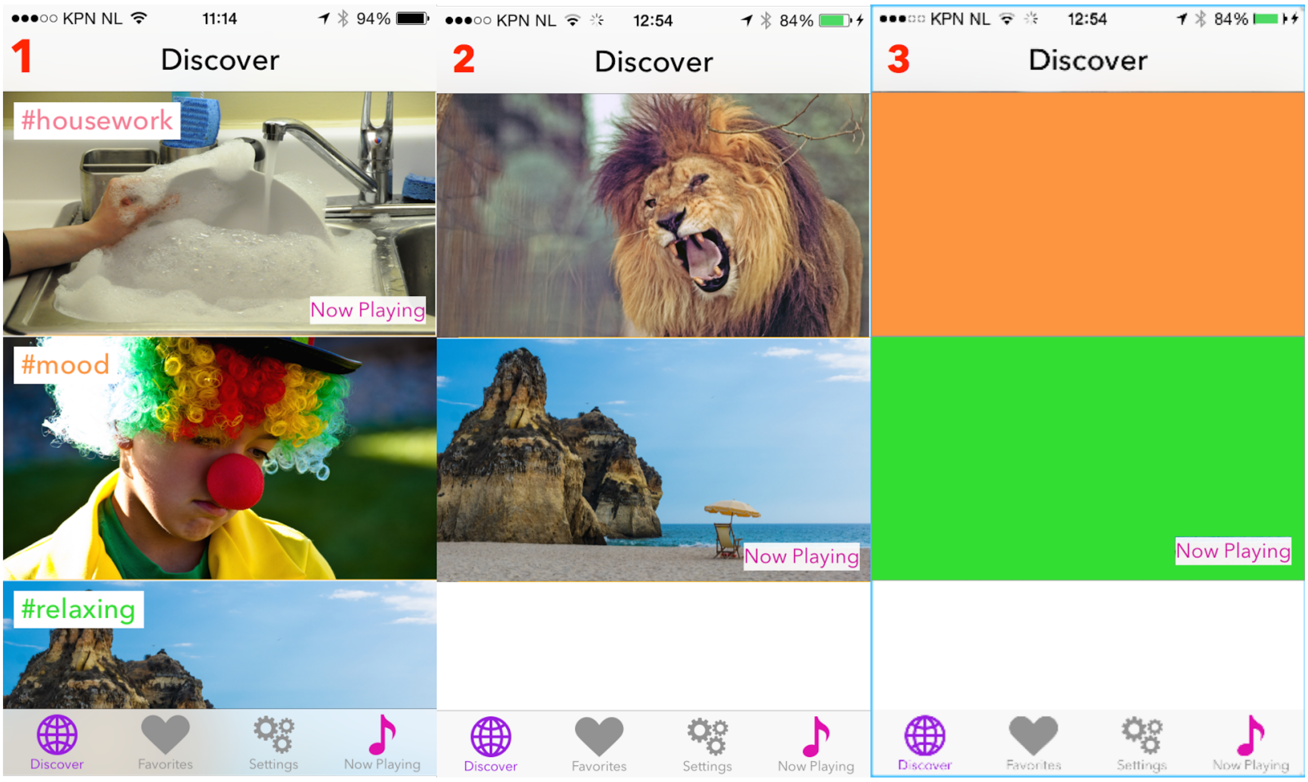
....

....

....

....

....



8. **Which screenshot of the user interface from the playlist view is the most intuitive from a playlist selection point of view?** *

If you have any other suggestions, please select other and explain your answer.

Mark only one oval.

- ( ) 1
- ( ) 2
- ( ) 3
- ( ) Other: ____________________

9. **How closely associated were your playlist selections to the situation you were in at the given moment?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Not close at all	( )	( )	( )	( )	( )	( )	( )	( )	( )	Very closely associated

10. **What did you like MOST about the application's user interface?**

____

____

____

____

____

11. **What did you like LEAST about the application's user interface?**

____

____

____

____

____

12. **Which aspects about the application's user interface were confusing for you?**

____

____

____

____

____

13. **Any user interface controls (buttons, screens) that you were missing from the user interface?**

....

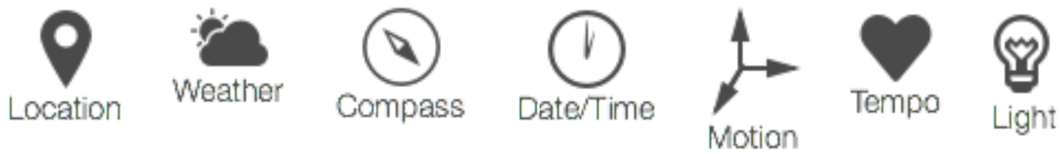
....

....

....

....

Context/Situation Recognition



14. **Which of the following playlist tags could you relate to?** *

Check all that apply.

- [ ] #wake up
- [ ] #commuting
- [ ] #working
- [ ] #relaxing
- [ ] #exercising
- [ ] #housework
- [ ] #sleeping
- [ ] #mood
- [ ] Other: ....

15. **Select all the playlists that you listened to at least once** *

Check all that apply.

- [ ] #wake up
- [ ] #commuting
- [ ] #working
- [ ] #relaxing
- [ ] #exercising
- [ ] #housework
- [ ] #sleeping
- [ ] #mood

16. **What other types of playlists would you have liked to see?**

....

....

....

....

....

17. **Would you have liked to know why the playlists/songs were recommended for you?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Not at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	Absolutely

18. **Was the use of sensory data noticeable in the music recommendations?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Very noticeable	◯	◯	◯	◯	◯	◯	◯	◯	◯	Not noticeable at all

19. **The playlists mostly matched the situations you were in.** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Strongly disagree	◯	◯	◯	◯	◯	◯	◯	◯	◯	Strongly agree

20. **Were there any situations where the playlist did not match your situation?**

....

....

....

....

....

21. **Did you feel like you had sufficient playlist/music choice at all times?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Strongly disagree	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Strongly agree

22. **If you disagree, can you please elaborate?**

____

____

____

____

____

Music Recommendations

23. **How relevant were the music recommendations based on your disliked music preferences?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Not relevant at all	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very relevant

24. **Which playlist's music recommendations did you enjoy the MOST?** *

Mark only one oval.

- #wake up
- #commuting
- #working
- #relaxing
- #exercising
- #housework
- #sleeping
- #mood

25. **Can you explain any specific instances and/or situations?**

26. **Which playlist's music recommendations did you enjoy the LEAST?** *

Mark only one oval.

- #wake up
- #commuting
- #working
- #relaxing
- #exercising
- #housework
- #sleeping
- #mood

27. **Can you explain any specific instances and/or situations?**

28. **How often did you find yourself listening to songs from the '#mood' playlist?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Rarely	○	○	○	○	○	○	○	○	○	Very often

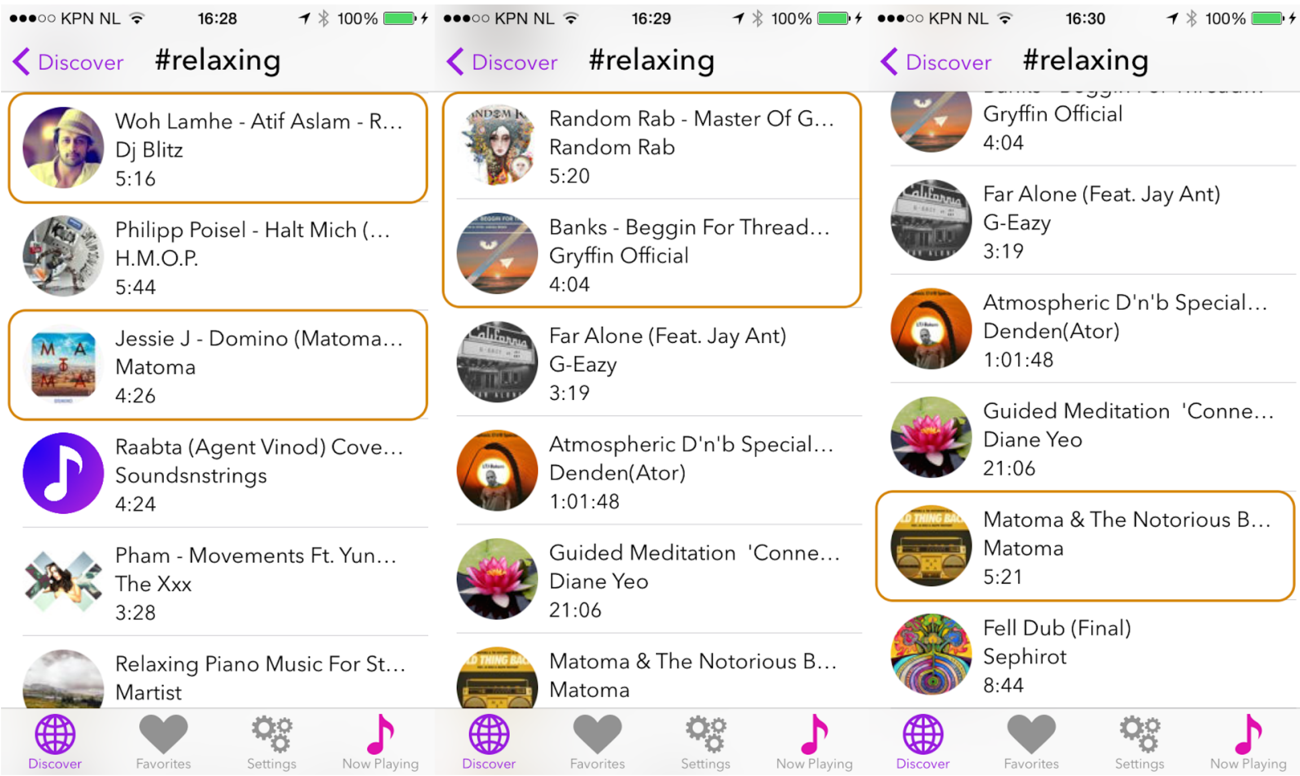
29. **How often did you find yourself listening to songs from the other playlists?** *
Other playlists: #wake up, #commuting, #working, #relaxing, #exercising, #housework, #sleeping
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Rarely	○	○	○	○	○	○	○	○	○	Very often

30. **Which playlist's music recommendations did you prefer in general?** *
Mark only one oval.

- ○ #mood
- ○ #wake up / #commuting / #working / #relaxing / #exercising / #housework / #sleeping
- ○ Unsure
- ○ Other: ....

Imaging that you were listening to #relaxing playlists. The picture below shows three examples. Please look at the examples.



31. **Now imagine that the songs shown in the yellow boxes did not appear in the playlist (in other words, they were skipped because the system did not recommend this). How pleased would you be with the playlist minus the songs in the yellow boxes?** *

The recommender could skip these songs because it deems that the songs in question could have music content that could possibly not matched your situation*

Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Very displeased	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very pleased

32. **What made you pleased/displeased by the system's decision to remove these song choices as shown in the yellow box above from the #relaxing playlist?** *

Your answer should explain why you made the choice you made above.

____

____

____

____

____

33. **Are there any other songs from the above #relaxing playlist that you've would've rather preferred the system to skip? Why?**

____

____

____

____

____

34. **How useful was the application in discovering new music that you might not have otherwise have discovered to on your own?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Not very useful	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very useful

35. **Which online music service(s) do you use primarily for music discovery?** *

Check all that apply.

- [ ] Spotify
- [ ] Apple Music
- [ ] YouTube
- [ ] Kollekt.FM
- [ ] Rdio
- [ ] Deezer
- [ ] 22tracks
- [ ] SoundCloud
- [ ] Other: ____________________

36. **During the test period, how often did you find yourself listening to music from other online music sources?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Rarely	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very often

37. **If you answered OFTEN to the previous question, can you please elaborate why?**

____

____

____

____

____

General Feedback

38. **How was ImpliciTunes similar and/or different from other online music services?**

39. **Can you describe an instance where you REALLY ENJOYED using the application (given the situation you were in)?**

40. **Can you describe an instance where you REALLY DISLIKED using the application (given the situation you were in)?**

41. **How likely are you to recommend this application to your friends for new music discovery?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Very unlikely	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very likely

42. **If you answered unlikely to the previous question, can you please elaborate why?**

____
____
____
____
____

43. **How satisfied are you in general with the application?** *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Very unsatisfied	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	⬭	Very satisfied

44. **What were some features that you sorely missed which would have improved your overall user experience?**

____
____
____
____
____

45. **What other improvements would you suggest for future development?** *

____
____
____
____
____

46. **How was your overall testing experience?** *
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
I didn't enjoy it at all	◯	◯	◯	◯	◯	◯	◯	◯	◯	Very enjoyable

47. **Would you like to continue using ImpliciTunes for new music discovery now that the test period has ended?** *
Mark only one oval.

◯ Yes
◯ No
◯ Other: ______________________

48. **If you did not enjoy the testing experience, can you please explain why?**

____
____
____
____
____

49. **Finally - for one final time :), can you please enter your e-mail?** *

____

Bibliography

[1] "Gartner's 2015 hype cycle for emerging technologies identifies the computing innovations that organizations should monitor." http://www.gartner.com/newsroom/id/3114217, 2015. [Online; accessed 2015-09-14].

[2] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelffle, "Vision and challenges for realising the internet of things." http://www.internet-of-things-research.eu/pdf/IoT_Clusterbook_March_2010.pdf, 2010. [Online; accessed 2015-09-01].

[3] "Music 360 - 2015 highlights." http://www.nielsen.com/us/en/insights/reports/2015/music-360-2015-highlights.html. Accessed: 2015-09-19.

[4] J. Bobadilla, F. Ortega, A. Hernando, and A. GutiéRrez, "Recommender systems survey," *Know.-Based Syst.*, vol. 46, pp. 109–132, July 2013.

[5] G. Bonnin and D. Jannach, "Automated generation of music playlists: Survey and experiments," *ACM Comput. Surv.*, vol. 47, pp. 26:1–26:35, Nov. 2014.

[6] A. Sen and M. Larson, "From sensors to songs: A learning-free novel music recommendation system using contextual sensor data," *LocalRec 2015*, September 2015.

[7] Y. C. Zhang, D. O. Séaghdha, D. Quercia, and T. Jambor, "Auralist: Introducing serendipity into music recommendation," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, (New York, NY, USA), pp. 13–22, ACM, 2012.

[8] P. Knees and M. Schedl, "Music retrieval and recommendation: A tutorial overview," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, (New York, NY, USA), pp. 1133–1136, ACM, 2015.

[9] A. K. Dey, "Understanding and using context," *Personal Ubiquitous Comput.*, vol. 5, pp. 4–7, Jan. 2001.

[10] X. Wang, D. Rosenblum, and Y. Wang, "Context-aware mobile music recommendation for daily activities," in *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, (New York, NY, USA), pp. 99–108, ACM, 2012.

[11] S. Cunningham, S. Caulder, and V. Grout, "Saturday night or fever? context-aware music playlists," in *In: Proc. Audio Mostly*, 2008.

[12] S. Nirjon, R. F. Dickerson, Q. Li, P. Asare, J. A. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao, "Musicalheart: A hearty way of listening to music," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, (New York, NY, USA), pp. 43–56, ACM, 2012.

[13] Y. Yu, Z. Shen, and R. Zimmermann, "Automatic music soundtrack generation for outdoor videos from contextual sensor information," in *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, (New York, NY, USA), pp. 1377–1378, ACM, 2012.

[14] J. S. Lee and J. C. Lee, "Context awareness by case-based reasoning in a music recommendation system," in *Ubiquitous Computing Systems*, pp. 45–58, Springer, 2007.

[15] H. Liu, J. Hu, and M. Rauterberg, "iheartrate: A heart rate controlled in-flight music recommendation system," in *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research*, MB '10, (New York, NY, USA), pp. 26:1–26:4, ACM, 2010.

[16] D. Griffiths, S. Cunningham, and J. Weinel, "A discussion of musical features for automatic music playlist generation using affective technologies," in *Proceedings of the 8th Audio Mostly Conference*, AM '13, (New York, NY, USA), pp. 13:1–13:4, ACM, 2013.

[17] J. Russell, "A circumplex model of affect," *Journal of personality and social psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[18] M. Kaminskas, F. Ricci, and M. Schedl, "Location-aware music recommendation using auto-tagging and hybrid matching," in *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, (New York, NY, USA), pp. 17–24, ACM, 2013.

[19] Z. Cheng and J. Shen, "Just-for-me: An adaptive personalization system for location-aware social music recommendation," in *Proceedings of International Conference on Multimedia Retrieval*, ICMR '14, (New York, NY, USA), pp. 185:185–185:192, ACM, 2014.

[20] K. Okada, B. F. Karlsson, L. Sardinha, and T. Noleto, "Contextplayer: Learning contextual music preferences for situational recommendations," in *SIGGRAPH Asia 2013 Symposium on Mobile Graphics and Interactive Applications*, SA '13, (New York, NY, USA), pp. 6:1–6:7, ACM, 2013.

[21] M. Soleymani, M. Larson, T. Pun, and A. Hanjalic, "Corpus development for affective video indexing," *CoRR*, vol. abs/1211.5492, 2012.

[22] P. N. Juslin and P. Laukka, "Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening," *Journal of New Music Research*, vol. 33, no. 3, pp. 217–238, 2004.

[23] C. Laurier, *Automatic Classification of musical mood by content-based analysis*. PhD thesis, Universitat Pompeu Fabra, 2011.

[24] Y.-H. Yang and H. H. Chen, "Machine recognition of music emotion: A review," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, p. 40, 2012.

[25] A. Gabrielsson, "Emotion perceived and emotion felt: Same or different?," *Musicae Scientiae*, vol. 5, no. 1 suppl, pp. 123–147, 2002.

[26] D. Hume, "Emotions and moods," *Organizational Behavior*, pp. 258–297, 2012.

[27] J. J. Denissen, L. Butalid, L. Penke, and M. A. Van Aken, "The effects of weather on daily mood: a multilevel approach.," *Emotion*, vol. 8, no. 5, p. 662, 2008.

[28] K. M. Goldstein, "Weather, mood, and internal-external control," *Perceptual and Motor skills*, vol. 35, no. 3, pp. 786–786, 1972.

[29] M. G. Harmatz, A. D. Well, C. E. Overtree, K. Y. Kawamura, M. Rosal, and I. S. Ockene, "Seasonal variation of depression and other moods: a longitudinal approach," *Journal of Biological Rhythms*, vol. 15, no. 4, pp. 344–350, 2000.

[30] E. Howarth and M. S. Hoffman, "A multidimensional approach to the relationship between mood and weather," *British Journal of Psychology*, vol. 75, no. 1, pp. 15–23, 1984.

[31] A. Hannak, E. Anderson, L. F. Barrett, S. Lehmann, A. Mislove, and M. Riedewald, "Tweetin'in the rain: Exploring societal-scale effects of weather on mood.," in *ICWSM*, 2012.

[32] J. L. Sanders and M. S. Brizzolara, "Relationships between weather and mood," *The Journal of General Psychology*, vol. 107, no. 1, pp. 155–156, 1982.

[33] V. Nowlis, "Research with the mood adjective check list," *S. S. Tomkins & C.E. Izard (Eds.), Affect, cognition, and personality*, pp. 352–389, 1965.

[34] D. Eck, P. Lamere, T. B. Mahieux, and S. Green, "Automatic Generation of Social Tags for Music Recommendation," in *Advances in Neural Information Processing Systems* (J. C. Platt, D. Koller, Y. Singer, and S. Roweis, eds.), Cambridge, MA: MIT Press, 2008.

[35] P. Knees and M. Schedl, "A survey of music similarity and recommendation from music context data," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 1, pp. 2:1–2:21, 2013.

[36] M. Levy and M. Sandler, "A semantic space for music derived from social tags," in *8th International Conference on Music Information Retrieval (ISMIR 2007)*, 2007.

[37] Karthik Yadati, Martha Larson, Cynthia C. S. Liem, and Alan Hanjalic, "Detecting drops in electronic dance music: Content based approaches to a socially significant music event," in *The 15th International Society for Music Information Retieval Conference (ISMIR)*, 2014.

[38] M. Turck, "Making sense of the internet of things." http://techcrunch.com/2013/05/25/making-sense-of-the-internet-of-things/, 2013. [Online; accessed 2014-11-03].

[39] "Echonest - acoustic attributes overview." http://developer.echonest.com/acoustic-attributes.html. Accessed: 2015-03-30.

[40] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[41] K. Tanaka, *An introduction to fuzzy logic for practical applications.* Springer, 1997.

[42] R. Babuska, *Knowledge-Based Control Systems.* Delft Center for Systems and Control, 2010.

[43] J. F. Kelley, "An iterative design methodology for user-friendly natural language office information applications," *ACM Trans. Inf. Syst.*, vol. 2, pp. 26–41, Jan. 1984.

[44] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen, "Iodetector: A generic service for indoor outdoor detection," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, (New York, NY, USA), pp. 113–126, ACM, 2012.

[45] J. Rada-Vilela, "fuzzylite: a fuzzy logic control library," 2014.

[46] R. B. Knapp, "Fuzzy sets and pattern recognition." http://www.cs.princeton.edu/courses/archive/fall07/cos436/HIDDEN/Knapp/fuzzy.htm, 2004. [Online; accessed 2015-09-26].

[47] E. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Electrical Engineers, Proceedings of the Institution of*, vol. 121, pp. 1585–1588, December 1974.

[48] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *Systems, Man and Cybernetics, IEEE Transactions on*, no. 1, pp. 116–132, 1985.