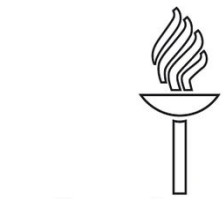
MUSIC ADVISER – EMOTION-DRIVEN MUSIC RECOMMENDATION ECOSYSTEM

Jyväskylä University

Master's thesis

2017

Mikhail Rumiantcev
Department of Mathematical Information Technology
Oleksiy Khriyenko



JYVÄSKYLÄN YLIOPISTO

ABSTRACT

Author
Mikhail Rumiantcev

Title of thesis
Music adviser - Emotion-driven music recommendation ecosystem

Discipline	Type of work Master's thesis
Time (month/year) February 2017	Number of pages

Abstract

In respect of the big amounts of music available in the web, people met the problem of choice. From another side, practically unlimited resources can bring us new opportunities in the music context. Efficient data management engines which are smart and self managed are in demand nowadays in the music industry to handle music sources amounts of which are coming towards to infinity continuously. This study demonstrates feasibility of the emotional based personalization of music recommendation system. There is still gap between human and artificial intelligence, robotics do not have intuition and emotions which represent critical point of recommendations. Taking into account significant influence of music to human emotions, we can notice that it can be a strong chain between human emotions and machines. This work provides the novel implementation of the music recommendation system based on emotional personalization, which manages human emotions by selecting and delivering music tracks based on their previous personal listening experience, collaborative and classification filtering.

Keywords
recommendation system, music, web services, machine learning

Location : Jyväskylä University Library

ACKNOWLEDGEMENTS

Prima facie, I would like to thank my parents Tatyana and Sergey and grandparents Galina and Igor, they raised me, provided motivation for my studies since my childhood, and supported me in all undertakings.

I am grateful to the Department of Mathematical Information Technology of the University of Jyvaskyla and particularly the Web Intelligence and Services Engineering (WISE) program where I did my master studies. I appreciate the structure of studies within this program which combines theoretical and practical knowledge in cutting edge artificial intelligence and web technologies. I have never regret that I chose this program, because throughout the whole studying process I felt the growth of knowledge and gained skills useful in professional business.

I would like to express my sincere gratitude to my supervisor Oleksiy Khriyenko. He is friendly and open person and always welcome for new proposals and suggestions. Through constructive dialogues and brainstorms he elaborated great ideas and conducted the topic of this thesis from everyday life use cases. Also I would like to thank my supervisor for valuable advice and insightful comments which he gave me throughout the writing process of my thesis.

During my studies I took part in the Music Psychology (MuPsych) project. I would like to acknowledge the docent of Music Psychology at the University of Jyvaskyla Suvi Saarikallio, who provided the opportunity to take part in the MuPsych project. My sincere thanks goes to the postdoctoral researcher of Music Psychology at the University of Jyvaskyla Will M Randall, who was the project coordinator of MuPsych. Initially, my task at that project was to develop the system which collects personal psychological reflections during music listening. It was a mobile player which interacted with users, collected their feedback and sent to the server side. Later, based on that project, my supervisor Oleksiy Khriyenko offered me the idea of the music recommendation engine. After some time, Suvi and Will described the idea of

the ongoing research which relates to support young people with music. I found that the research idea and the idea of my thesis topic are on the same page, and this study will have broad implications in the future. Also I am grateful to Will for proofreading of my thesis and collaboration from the psychological point of view which helped to investigate music influence to emotions and wellbeing of humans.

Jyväskylä
February 2017
Mikhail Rumiantcev

ABBREVIATIONS

AI - Artificial Intelligence

API - Application Programming Interface

APK - Android Package Kit

BI - Business Intelligence

DNS - Domain Name Servers

DSL - Domain Specific Language

DTO - Data Transfer Object

EB - Emotional Business

EI - Emotional Intelligence

EBI - Emotional Business intelligence

FTP - File Transfer Protocol

GPS - Global Positioning System

GUI - Graphical User Interface

HTTP - Hypertext Transfer Protocol

IBM - International Federation of the Phonographic Industry

IoT - Internet of Things

IP - Internet Protocol

JSON - JavaScript Object Notation

OSI - Open System Interconnection

RDF - Resource Description Framework

REST - Representative State Transfer

SDK - Software Development Kit

SPARQL - recursive acronym for SPARQL Protocol and RDF Query Language

URI - Uniform Resource Identifier

URL - Uniform Resource Locator

XML - Extensible Markup Language

LIST OF FIGURES

CONTENT

1 INTRODUCTION

The contemporary world is filled with a huge amount of machines. According to the Cambridge dictionary[1], the term machine relates to an entity which is targeted to perform specific tasks, and usually contains parts with separated functions. Computers are machines which are able to do tasks by following exact instructions which should be prepared by people beforehand. Software development has strictly followed this paradigm for the last several decades, and it is still true today but with some additions. Machines are now faster than humans, and able to process large scales of information. Even in 1997 the Deep Blue[2] chess robot gave a perfect result, when it won against the world chess champion Garry Kasparov. Of course that machine was specially prepared for that player and decisions made by the program relied to conditions occurred at the game.

The cyber world has already reached the point of the machine learning, and now there are many software systems which use computational experience which was already performed by them to improve the further work. Software has become smarter than it was earlier. Programs contain instructions of the learning processes. We can see machine learning in software which is related to source recognition, such as texts, voice, colors etc. The larger the recognized number of sources, the better output engines provide. For example the Google Speech recognition API (2017) applies neural networks to support continuous improvements of the work. The Alpha Go (2016) computer program won the professional player at the intellectual board game Go[3], it utilized the machine learning at the prepare state. IBM Research Labs introduced the cognitive supercomputer which outperformed humans on the Jeopardy show[4]. Currently we can say that artificial intellect is already able to compete with humans.

Machines have become smart, but they still are robots without feelings, emotions and intuition. Indeed they do not need to have their own opinion to work for people at the current

[1] Cambridge dictionary - dictionary published by the Cambridge University Press, has over 140, 000 words and descriptions.
[2] Deep Blue - computer developed by IBM, programmed with playing logic of chess game.
[3] Go - traditional Chinese intellectual board game.
[4] Jeopardy show - American television show based on the math game with the same name.

state of the cyber development. However, there are systems which perform source analyzes as naturally as it could be done by human brain. For example, IBM tone analyzer (IBM, 2016) which determines text attributes in emotional, linguistic style and social contexts. Another very powerful example is Cloud Natural Language API (Google, 2016) which also provides detailed set of text features taking into account different analysing aspects. How do we push the software systems to learn about human emotions and reflect their preferences on the fly? Let us consider this question within the context of music. If we ask people about their music preferences, the most popular answer will be: "Depending of the situation or mood." The idea for a solution to this issue is loosely based on this typical answer. Music can not only reflect human emotions but can also influence to them. It is therefore important to design and develop the service which will understand feelings and mood of the people and provide them music which will match their preferences and perform music based emotional treatments by bringing them to the target emotional statements.

Recommendation systems are widely used in trading systems, web services advertisements and other spheres. Youtube[5] is quite nice example of recommendation system. Based on user experience during the video watching session on Youtube it selects videos with similar topic meanings and puts them into a stack near the main video player widget. As result user spend on this web resource longer time. According to watched content and particular location this service shows appropriate advertisements.Similar strategy is used in other systems, however we are going to talk more about emotional personalisation rather than other recommendation techniques.

A lot of research has been done with respect to a music driven influence on emotional state of a human. According to Levitin and Chanada (2013, 179-193) music can have significant impact to humans, it is able even to regulate their blood pressure, heart rate and condition of the whole body and well-being. There is wide range of use cases where such technologies of music regulation could be used. Sportsmen need to maintain their heart rate at some time periods of their trainings, make it faster or slower to show better results and keep health in good condition. Intellectual workers need to keep fresh their brains to be able to continue their work well. Drivers need to keep their attention sharply throughout their travelling and be

[5] Youtube - video sharing web resource, was introduced in 2005.

awake. Poets, composers and other people somehow related to art activities can retrieve new ideas from music listening. Taking into account this significant effect of the music to the body conditions, we can notice that with right approach music can improve performance if humans in different fields such as sport, work, study, art and many others. Music helps us to relax or otherwise to focus, to entertain, to feel sadness or happiness when we need it. Also this feature of music can be used for treatment purposes, indeed nowadays psychologists use similar methodologies in their expertise, however there is no tool which performs this kind of treatment in automated manner. It would be great to have the tool which would make analyses, filter tracks and deliver them to people in such manner that their well being and work efficiency would be improved.

1.1 Digital music industry

Science and industry always go together. I venture to note that each of these entities does not make much sense without the other, because development is not possible without theoretical foundations and calculations, which, in turn, have motivation only if their output can be applied to the real life or in other words implemented. However a transitional forward movement involves hard utilization of existing knowledge and technologies, it is required to go further and find something innovative. Let us examine the current situation on the music market to determine what issues are still missed and which opportunities occur today.

The digital music market has significant influence on the global music industry, which already crossed the point where digital formats have a greater market share than physical formats. In this situation we can also observe that music streaming services take almost half of the digital music market and sales of these services continue to gain momentum. (IFPI Report, 2016)

The internet has drastically changed the lives of humans, with related technologies allowing for the efficient and rapid spread of innovations through the population from over the world. It connects people and keeps them in contact with each other and with the latest news continuously. Social and media networks allow new artists to share their music and video clips with others, avoiding extra overhead, and some of them become famous quicker and

easier than it was few decades ago. The significant growth of mobile technologies has dramatically increased this interconnection and data sharing, because usually people keep their mobile devices on their person, and often keep them online almost without interruptions whenever they explore new music tracks. (How Digital Marketing Is Changing the Music Industry, 2016)

The above mentioned achievements in Internet technologies have solved many of the limitations of music exploration. There are many music streaming platforms such as Spotify, Apple Music, Pandora, Last FM, Mix clouds and others. The key principle of their work is establishing constant music streams for the users through server-client connection. According to the high popularity of these services we can note that the majority of people prefer to pay a reasonable price for convenient services and enjoy large scale music selection instead of illegal downloading and listening limited set of tracks. Music platforms usually present millions of tracks, which are structured and have metadata, yet users faced with this amount of music may be compared with the situation when we look up at the starry sky. Certainly, most music platforms have search engines which are based on keyword searching, for example artist or song name. Some of these services include music advising add-ons. However they perform just selection of artists and songs which can be considered as similar to the listened ones based on genre and era at all. Indeed we cannot dive too deep into these technologies according to the production non-disclosure rules. We can just define from the user side that services which exist on the market do not perform a real time interaction with humans to define what they feel and how personal preferences vary depending of the context.

1.2 Music based emotion management

Taking into account the significance of the influence of music on the mental and physical health of people, and lack of web services that pay attention to this issue, I propose that the topic of a music curation engine has a great number prospects for development. The main output of this study is the full descriptive design implementation of a music curation service with the kernel based on emotional personalization. This will begin with the introduction of the main goals and working principle directions of the service.

1.2.1 Decision making support

Have you ever encountered the decision making difficulty problem? What product to buy, what movie to watch, what music to listen to. Routine life becomes boring very fast and everyone wants to enjoy it by opening new horizons instead of repeating the same things day by day.

When we talk about emotional component in interactive recommendation systems, we also imply intelligence which should be applied to these systems. The domain of Emotional Business intelligence (EBI) is targeted to support emotional and nonemotional decisions related to business. There are various intelligence types are composed in this domain: Business Intelligence (BI) which can be can considered as traditional business domain, Emotional Business (EB) which is targeted to emotional regulations and methodologies mostly related to the psychological expertise and finally the Emotional Intelligence (EI) which is aimed to ways of implementation of the EB methodologies and applying them into practical systems. EBI is based mostly on capturing of the experience, it represents the structure of the organization which is smart and continuously learning. One of the most important component of the EBI is the explorer of emotions and feelings, which can be used for further analyzes. The purpose of that kind of technologies is in demand because currently companies and services have complex structure when it is not possible to handle resources efficiently in a manual manner and get all work done in time. From another side this kind of intelligence can bring as more implications for the product promotion because web based systems are able to handle and process more information about subjects and hypotheses in comparison with humans and non intelligent tools. (Terziyan et. al. 2014)

A wide range of music resources makes the problem of choice within the music context somewhat controversial. From one side, this provides possibilities to solve the limitation issue and remove listening restrictions. However, risk of an unsuccessful choice rises dramatically in this case. Sometimes it requires additional effort or even can prevent us from exploring new music. When we want to watch a movie which we have never seen, websites with movie trailers or with user's feedbacks could help significantly. Imagine the situation when we pay

the same attention when want to listen to new music, it would be frustrating to make such an effort for every song, album or artist, furthermore not every track on an album would match the habits or preferences of the user, even music fans do not love all songs of their favourite bands. A music adviser service related to this study should capture emotional and activity conditions of the user and music tracks which they listen to. Based on these data the system has to generate playlists by exploring new music tracks, which will match personalised preferences and particular emotional conditions, taking into account the activity performed by the user in their current state. The system is aimed to explore new music which can be considered similar to tracks which were listened by the user and marked as suitable for personal preferences. The main working sense of the service at this scope is that the user gets the phone, presses the play button and immediately starts to listen new music suitable for him or her. All predictions of the emotional conditions and activities could be established by utilizing sensor systems of mobile devices in an automatic manner. This feature makes the music player extremely convenient to use. In this case, not only does the user listen to the player, but the player also listens the user to satisfy all personalized music desires.

1.2.2 Music therapy

Music has a significant effect on our well-being. It is not a secret that it can help to boost energy, to relax, to cope with disease, speed up the rehabilitation and many others things. Very often health changes partly depend on the psychological conditions of the person. Stresses influence the nervous system, which in turn affects all internal organs, especially the cardiovascular system. Mohhamedreza et al. (2013, 167-171) provides evidence of the stress influence on blood pressure and heart rate. These experiments include cardiovascular system measurements of people in different states who have been affected by negative and positive emotions. Emotional stress can have positive or negative outcomes, and balancing these factors can help to keep the well being of the person at the desired level. Getting tired of work very often depends not on the difficulty of the tasks, but from the emotional pressure which comes from unresolved tasks. Relaxing is more efficient and takes less time when it occurs before tiredness. Music helps to relax, without distraction from working process, it brings higher productivity and avoiding the brain overloading. Mental health plays a significant role

in human life, because all factors and conditions of the personal well being are interconnected.

The main target of the music therapy domain is support in emotional management. The system learning phase is expected to be performed continuously at a run time and it includes diagnostics of the user's emotional states, detecting their particular activity, and identifying the correlation of these two attributes with the music preference changes. Taking into account the data collected from all available sources, the system should perform the music selection based on collaborative and classification based filtering principles.

Knowledge about the personal mood, activity and music preference interconnections makes it possible to push people smoothly from one mood state to another by providing music tracks with specific attributes. In this case music playlists are expected to be updated on the fly during the listening session depending of the user's behaviour. The client part which is represented as a music player should include settings which define the current and desired emotional states for the ongoing listening session. Sensor systems of the mobile device and the social media data are expected to be utilized for performing diagnostics in an automatic manner, as to avoid extra questionnaires. The system aims to enhance the mental and physiological well being of people from different social and age groups.

The issue described in the previous section relates to psychological therapy, however sometimes people are not able adequately aware of their emotional states, even if the situation takes more serious forms. For example it could be hard to define real sources of stress by ourself. Medicine and psychology sciences have a wide range of therapy approaches and methodologies of exploring causes of mental and physiological diseases. With knowledge in these areas, professionals perform diagnostics more precisely in comparison with self treatment. According to these diagnostics, they chose the best ways of addressing these problems. In this way, timely, explicit and clear definition of the problem leads to a successful solution. In this case the music curation service could be a great tool of psychology therapists. The data management of the user account in this service form can be managed externally. The system should have a feature which allows therapists to push some recommendations to the system to help their patients to tackle problems.

If we observe the youth part of the service audience, we should pay attention to the specific issues related to adolescence, such as socialization aspects, critical points of transition in life, issues in communication with peers, etc. Sometimes teenagers can have misunderstandings of their real emotional states, for example aggression might not always look like a problem for themselves, and they may even get pleasure from boosting their aggression. The same thing can happen with other feelings, which may be harmful for health such as longing or even euphoria. In this situation, parents and psychology supervisors at schools can take care of managing the settings and recommendations of the music therapy system. By this way, parents also are able to monitor the psychological dynamics of their children, which can help occasionally to avoid significant problems in social life.

1.2.3 Personal safety

In addition to music based therapy, we can highlight the safety domain. It is targeted in advance to the situations when people have to keep sharpen their attention and be focused on critically important things. A big number of traffic incidents were caused by the reasons of sleepy drivers, big percent of them had fatal end. Even when drivers sleep enough, they could feel sleepy, there is wide set of reasons for that: weather, sickness, continuous driving on an open area, not enough rest, high emotional charging and many others. Music is very suitable to discharge emotions and keep well. Taking into account all these facts and high influence of music to a human arousal, we can notice that music recommendation system can be used in safety domain. The system can filter music tracks which rise energy and keep the driver awake and brain in a good untired condition. If the vehicle has sensors, face recognition or some other detectors, all of them can be used as enforcements of the recommendation system. For example when the system detected that the driver is tired or begins to feel sleepy, it can offer tracks with higher energy and temp to increase the arousal of the person.

1.3 Objectives

Breaking limitations usually causes new issues and growth always generates new opportunities. The existence of large scales of music data on the web has a lot of implications.

For my study I want to highlight two of them: the problem of choice and what usefulness we can retrieve from these huge data sources.

The kernel of this study is directed to the architecture development of the music recommendation ecosystem in respect of the personal music preferences at specific emotional context. It is based on the user reflections and constant music metadata. Firstly, I want to design an efficient music exploring engine, which will perform filtering of music tracks which will match music preferences. Then I want to investigate the possibility of the tool implementation which is able to push a person to emotional states different from the current one.

The goals mentioned above are important because their achievement helps to resolve the problem of choice with efficient music exploring in an automated manner. Also, this research aims to maximize the service personalization by enforcing it within the emotional context.

I study this topic because I want to investigate how common recommendation methodologies can be applied to web services, and how to teach machines to understand user preferences through self-managed data retrieval. In other words, I aim to design a service which will interact with users by itself and will learn through the retrieved data.

At first I need to review literature sources related to this study, as this is helpful to familiarize the audience with general approaches of recommendations which are expected to be applied to the service. For people and especially for engineers it is critically important to know the working flow of the technology to understand it. To prove that the theoretical background is relevant and this kind of system could exist I develop the working prototype of the music recommendation engine.

1.4 Thesis structure

In the first chapter I introduce the general idea of the thesis topic to provide background to the study, determine purposes of this research and define goals. In the second chapter I talk about

recommendation methodologies briefly, as it will help us to understand fundamental approaches on which we are going to build our music curation ecosystem. Chapter 3 explores the technologies with which the services related to this study are expected to be built. In the fourth chapter I talk about approaches of the data collection and how to process the collected data. Chapter 5 illustrates the overall picture of the music recommendation ecosystem, and describes how technologies and approaches can be organized to work as the modular service. The implementation of the prepared ecosystem design is presented in Chapter 6. The purpose of the prototype implementation is the practical confirmation of the theory which is developed in previous chapters. Comparisons with existing recommendation systems at the music context are provided in the seventh chapter. The final result of this study, definitions of issues which are addressed at this thesis and possibilities for further researches in this topic are described in Chapter 8.

2 METHODOLOGIES OF RECOMMENDATION SYSTEMS

A great number of musical resources push cloud music services to find new efficient solutions for the issue of music exploration. Companies use various approaches of music recommendation. In this chapter I describe my investigation of existing methodologies in this area. It is good idea at first to define what is already known and can be utilized for making progress instead of reinventing the wheel.

Generally, if we need to recommend something we have to gather information about items which we consider as potentials for the recommendation, and get know at least the type of person to whom we are recommending, in other words, we need to make some initial investigations. The same situation happens in digital recommender systems: they start by gathering data, then process that data, and provide the output represented as recommendations to users.

The first phase of the recommendation process is represented as the information collection step. The system gathers all relevant data about users to determine their preferences, habits and behaviours, all of these data come to the data storage under the particular data model structure. Feedback from the user side is the main source of data for user profile creation, they could be implicit, explicit or hybrid. Explicit feedback mainly includes evaluations of items provided by users, which can also be considered as rating feedback, as it reflects opinions of the person about products or services. Implicit feedback describes a user's behaviour by collecting data from their previous experience, for example their purchase or service usage history. These two types of feedback can be mixed into hybrid feedback, which includes both: user's ratings and general behaviour data. Combination of explicit and implicit feedback decreases the effects of the disadvantages of each, because they complement each other. This approach improves the quality of recommendations and accuracy of matching of user's preferences. When all relevant data are collected, algorithms which are incorporated in a system can be applied to this data, this step is also known as learning phase. Finally, items should be filtered by applying rules which were generated in the learning phase. Selected

items can be presented to users as final output of the recommendation process. (Isinkaye et al. 2015, 260-268).

In this chapter I describe common approaches of recommendation systems by focusing attention on collaborative filtering, classification based filtering and how these two methodologies can be combined to improve the recommendation performance.

2.1 Collaborative filtering

All humans are unique, however we have a lot of common in terms of physiological and psychological characteristics. From one side, based on these characteristics people can be aggregated into groups with similar habits and behaviours, however there could exist other factors which could have influences on personal preferences. Interaction is the best way to understand the thoughts of someone. Information systems are able to automatically capture user behaviour through observation of how they use their services and make choices, without additional activities from the user's side. Users with intersections in characteristics and preferences can be considered as similar or belonging to the same group. In this case the system can make new recommendations to the user, based on the experience history of other users from the same group.

According to Schafer et al. 2007, collaborative filtering is based on analysing point of views of different people about a particular entity. The key principle of this filtering is sharing thoughts between people, which has been part of human nature since ancient periods of history, and in the last few decades has been applied by information systems.

The model of collaborative filtering involves the grouping of humans and entities related by their choices into clusters. This model includes an estimation of the preference matching probability between these groups. Estimation process is based not only on preference choices of people but also on their characteristics such as gender, age, country and features of items related to clusters. It is also known as user based collaborative filtering. However, the clustering model has a dynamic nature and classification should be performed in a more flexible way. (Ungar & Foster 1998, 115–122.)

According to Saric (2009, 302), collaborative filtering can be based on neighbourhood method, where similarities between users can be predicted according to product evaluation ratings which they provide. Based on the experience of other people with similarities, we can detect and evaluate the correlation between users which can help us to guess if the item is suitable for person or not. To illustrate this, Table 1 contains names of users and music tracks, where "1" indicates if user likes the song and "0" if not. Weighted similarity between Suvi and Bob can be considered as high because they have a lot of overlapping in preferences. Based on this similarity we can say that Suvi most probably will like the track "Lazy days", because Bob also likes it. Figure 1 illustrates the formula by which the similarity between two users (M and N) can be calculated, where M' and N' represent mean values of item evaluations made by each user.

	Bob	Ashley	Reijo	**Suvi**	Ted
For the better	**1**	0	0	**1**	0
Root of soul	**0**	1	1	**0**	1
London bridge	**1**	1	0	**1**	0
Ave Maria	**1**	0	1	**1**	0
Let us gather	**0**	1	0	**0**	0
Drawn on stone	**0**	1	1	**0**	0
Lazy days	**1**	0	0	**?**	0

Table 1. Matrix of user ratings

$$r_{MN} = \frac{\sum_k (M_k - \overline{M})(N_k - \overline{N})}{\sqrt{\sum_k (M_k - \overline{M})^2} \sqrt{\sum_k (N_k - \overline{N})^2}}$$

Figure 1. Similarity prediction formula

Collaborative filtering in the context of person to person evaluation is effective, however there is lack of scalability, as when the amount of users increases, the calculation process of similarities requires more computational power to iterate every time new users come to the system, moreover preferences have a dynamic nature and periodical recalculations are also needed. In this case, an item to item approach of collaborative filtering takes place. If items are rated positively by some people it means that this group of users have similarities in preferences and further offers can be established inside this particular group. Conditional probability and cosine similarity approaches can be applied for item based collaborative filtering in similarity calculations. (Ekstrand et al. 2010, 90-100)

Collaborative filtering recommendation approach has high efficiency and it is widely used by major companies such as Google, Amazon, YouTube and others. Despite all of its advantages, this methodology has one serious challenge, known as the “cold start problem”. It happens when the system just started and no user interactions occurred. Other methodologies can therefore be involved in the recommendation process during the earliest periods when entities are not rated yet and no or few collaborations have occurred. Zhao (2016, 15-30) in her study describes approaches of addressing the cold start issue of the collaborative filtering. For example, sequentially offering items one by one or groups of items for individuals and subdividing the audience into smaller groups for making smaller samples. All approaches presented in her work are united into the consistent framework which is targeted to decrease effects of the cold start problem in recommendation systems.

Sahebi and Cohen (2011, 1-5) offer the community based way for addressing the lack of data during initial periods of recommendation processes. The key principle of this method is collecting the data from social networks and media for further user evaluations and comparisons. This method has many implications because social networks are rich in information about the users and various products.

2.2 Classification-based filtering

Recommendation based on the evaluation of physical characteristics of entities is named classification-based filtering. It is a powerful alternative for collaborative filtering when the data from users is not yet collected. Depending on the item specifications, different features and classifications such as content, size, sound, colour etc. can be considered for evaluation.

Large numbers of items and data require structuring to avoid chaotic allocations and the impossibilities of quick retrieval of resources. In comparison with the data model of feedbacks and ratings coming from the user's side, the classification-based approach has an objective nature, in that characteristics of items are constant and do not depend on domains instead of external factors. This approach combines profiles of users with specific settings which are related or match attributes of items. Machine learning in recommendation systems is represented by the data-model formation which describes user's preferences from their experience of using the system. There are different algorithms which are used for the data processing to bind constant item attributes to user profiles. One of these is based on data partitioning and building logical trees depending on service usage behaviours of users; nested structure of these decision trees include accurate correlations and dependencies between user view and machine data. Nearest neighbour approach involves allocation of the data about the user experience and items, when new items come to the system their descriptions are compared with the data which have already been learned. Summarising of item's characteristics into particular tags and keywords speeds up the queueing process and improves efficiency of navigation in large scale data sets. Attributes of items can be processed with linear functions, for example defining borders or thresholds of characteristics in a user's profile for further filtering based on previous experience. Probability algorithms are also commonly used in classification-based data models. (Pazzani et al., 2007, 325-341)

According to Kim et al. (2006, 463-472), conditions under which music recommendation is performed can be defined at one place and from the recommendation context. This includes factors which might have effect to the user's preferences such as age, gender, country, weather at the current state, mood, activity, even blood pressure and heart beat if the measurements of these biometrics are available. In other words, the context describes the

condition of the person according to which the system should perform recommendations, because preferences typically are dynamic depending on many factors. Music fragments have characteristics which can be subdivided into different complexity levels: basic and advanced. The basic music metadata includes track, artist and album names. Higher level of the metadata might include music features and lyric analyses.

2.4 Hybrid recommendation systems

To get maximum efficiency, recommendation performance systems can combine collaborative and classification-based filtering. In the ideal case, the system should dynamically switch between approaches or use them together according to the requirements and measurements of the service efficiency. Combination of several techniques minimises – and sometimes eliminates – the disadvantages of each, because they complement each other and fill the gap in knowledge due to a lack of data.

Collaborative and classification-based recommendation techniques are most popular nowadays, however there are other methodologies which can be useful in hybrid recommendation systems. According to their working principles and specifications, ways of merging them into one system should be adopted to a particular context. Demographic-based recommendation aims to define similar users based on their particular demographic attributes, recommendations are then performed according to the type of person. Utility-based recommendations make calculations on the relevance of the item. Knowledge-based systems perform matching between needs of users and features of items. Mixed hybridization method involves a combination of several recommendation service providers. Weighted approach aims to utilize recommendation methods in different proportions at the same time, and the final output is calculated by linear function applied to the results provided by the recommendation techniques involved in the process. Another way to hybridize recommendation techniques is to switch between them depending on the current requirements. Outputs from different methodologies can be integrated and processed together by a specific algorithm. If the output of one recommender is used by another as input, it means that cascade hybridisation method takes place; this method can be applied on the meta-level when inputs and outputs are represented as data models. (Burke, 2002, 5-20).

Recommendation systems should take into account different aspects of service consumption, factors which describe how often users invoke services and how much money they spend on these services have significant effects for further recommendations. Accuracy of ongoing outputs coming from the recommendation service depends on recent feedback from the user and time passed since last use of the service, because user's preferences have a dynamic nature. Shih and Liu in their research (2005, 1-5) describe how user's demands can be combined with the potential profit evaluations estimated from relationships with them. Possible solutions include the weighted hybridization method, which combines customer demands and processing of data related to recency, frequency and monetary values related to the service consumptions by users.

2.6 Conclusion

Millions of music tracks are available in the web nowadays. Efficient data management is required, including resource allocation, classification and selection algorithms. Companies which provide music streaming services apply their forces to address music filtering issues: this is necessary for the efficient exploration of music by users, otherwise these huge amounts of tracks are useless. Companies pay significant attention to this problem and many high qualified professionals work on improvements of music recommendation services. Music services provide attractive results related to music exploration. All music recommendations, which users can get from web services today, are personalized, however emotional conditions of users are often not considered. There are many scientific studies about music recommendation approaches related to technology, psychology and business research areas. The question of music curation within the psychological context is not so innovative today, but it is still fresh and there is no fully implemented service on the market which takes into account emotional conditions of users to dynamically perform recommendations.

Personalization plays a key role in recommendation system workflow because tastes and preferences have very different flavours through all of people, indeed there are a lot of similarities but in some detailed aspects preferences are unique. Emotional aspect takes one of the most important positions in personalisation. The human mind balances between logic and

emotions continuously, many philosophers such as Schopenhauer[6] adhere these views in general. Usually we need time to process the information about facts, incidents environment around as, we need time to make some conclusions, when first impressions are based mostly on emotions, this nature of the human mind confirms the significance of emotional bases at recommendations. Decisions, behaviours and view of the world depend of the particular type of the person and particular preferences are tuned by the emotional contexts, today we love something, tomorrow or even today at the different mood we hate it. The future of recommendation systems will go hand in hand with emotional fundamentals.

[6] Arthur Schopenhauer - German philosopher (1788 - 1860).

3 TECHNOLOGIES RELATED TO THE STUDY

With the general idea and the kernel of the recommendation system which is related to this study defined, it is a good time to determine the best way to make the described service real and the requirements for the implementation. In this chapter I analyse which tools and technologies are suitable for the recommendation system and should be considered in this study. We are going to get familiar with fundamental technologies which are expected to be used in the recommendation system developments related to this study. The idea is to describe the overall image of techniques and explain how each of them can be implemented. However, I am not going to detail all of these technologies, firstly because we are limited in space and secondly, it will be redundant as there is wide range of study sources which would explain all technical details deeper. In other words, I will introduce the brief outlook of technologies which we are going to use in the ongoing developments.

3.1 Client – server communication

As I defined in previous chapters, the Music adviser system is represented as the server application which performs music data processing and the mobile player which is integrated with music streaming services to deliver selected music tracks to the end users. Logically it is clear that we need to establish an interaction between them to perform the data exchange. This section is focused on fundamental concepts and working principles of communication protocols, web service development, and mobile application development.

3.1.1 Hypertext Transfer Protocol

The acronym HTTP stands for the Hypertext Transfer Protocol, which is the basis protocol of the data exchange through the web. It is like a bridge which interconnects servers and clients in the world wide network.

Terabytes of data travel through the web daily. Originally it is hosted on the server side or can be transferred as peer to peer. Servers support HTTP protocols, clients send requests to servers and then receive reposes with the response status and data according to the particular context. The data which is hosted on services are also called web resources. This digital data can have a heterogeneous nature and structure such as media files, documents, web pages, metadata etc. Depending on the concrete technology which relies on HTTP, the data can be encoded and formatted in different ways. All web servers have web addresses to identify themselves and make them accessible them on the web. Internet Protocol (IP) consists of numerical values of web server addresses, which are not convenient for memorizing them. Domain Name Servers (DNS) contain user friendly equivalents for them and take care about address translations. HTTP transactions are represented as exchanging of message blocks. Each message has text line structure which contains a start line, header and body. Figure 2 illustrates detailed examples of HTTP request and response messages. Start line contains information which defines the type of the request method and the status of the response from the server, headers include additional specifications of the request and response such as the response structure or setting of the authentication. The body includes extra data which tends to be transferred to the server side, and can contain not only text but also binary data which represents files. (Gourley et al., 2002, 8-24.)

Following patterns of HTTP we can transfer the data over the web with different structures. For example we can design our client and server applications to send and receive messages with XML or JSON based formats. The number of data types available for transferring through the HTTP protocol is not limited by these formats. There is possibility to transfer media files with applying the encoding to them and send as binary data, it is also known as the “Multipart form data” format. If there is need of some advanced non repudiability and security, this data can be encoded by other ways, such as hexadecimal encoding with hash calculation. The payload of the media files can be transferred through the HTTP by attaching it to the body of request.

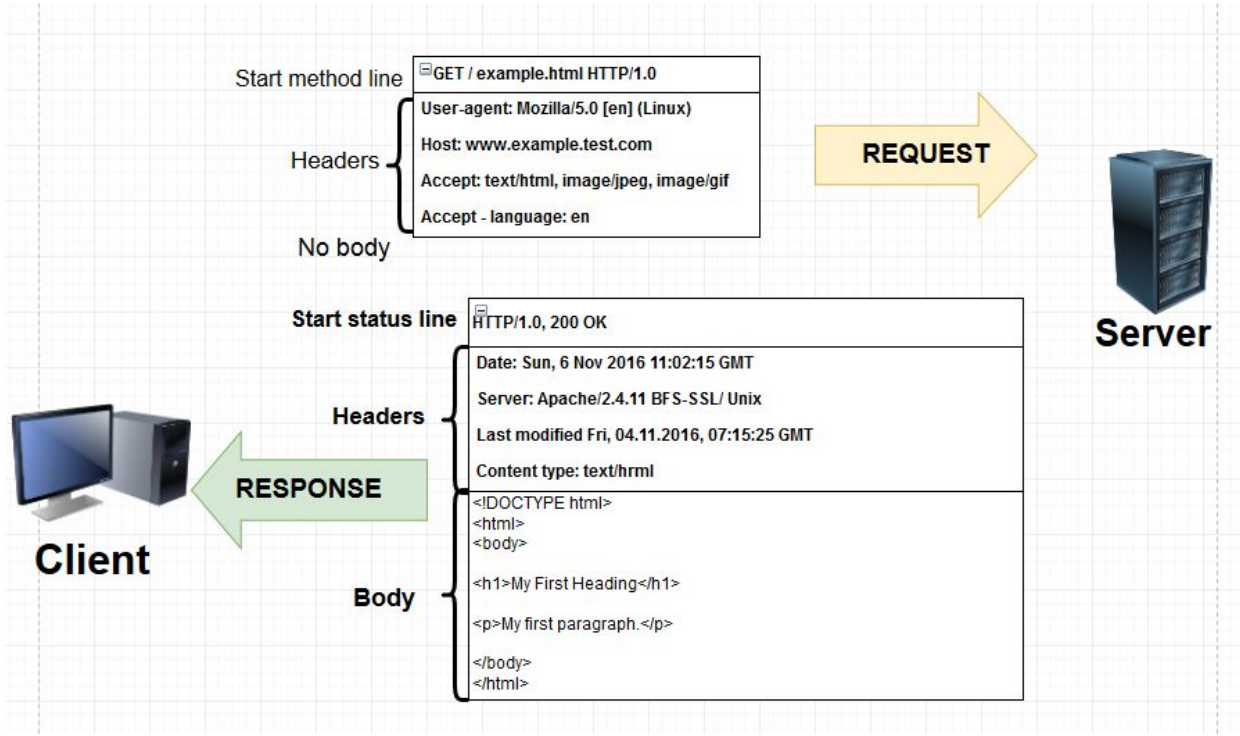


Figure 2. HTTP message structures

HTTP version 1.1 is the protocol of the application layer of the Open System Interconnection (OSI) model. It means that it operates directly with software and tends to have interactions with users. It has various methods which define restrictions and functional sense of requests. GET and HEAD methods tend to retrieve the data without making significant changes in the structure and the file system of the server. The main difference between these requests is that a response to HEAD does not contain the body part of the message, it is suitable for requesting the metadata only. The POST method includes the additional attached data in requests, processing functionality depends of the logical service design, it might be stored or modified according to requirements at the server side, all changes of the resource can be applied to the existing address. The PUT method is used to store content on the server on new specific address, there is also a method which has the opposite functionality named DELETE. For establishing tunnelling connection, the CONNECT method can be used. If the loop back of the request is needed to be defined the TRACE method can be requested. (Fielding et al., 2004.)

3.1.2 Representative State Transfer

There are several standards which work on top of HTTP, and we will discuss one of the most popular of them in this section. Representative State Transfer (REST) also known as RESTful is the resource oriented standard which relies on HTTP.

REST standard describes the design of services which are targeted to the management of resources on web services. Applications and services which meet REST criteria usually have simple structure of part which are responsible for endpoints publishing. One of the main characteristics of REST is addressability, in other words, resources and methods which belong to the service can be accessed with HTTP addresses, in some cases by applying additional parameters. Each HTTP request is independent from others or totally isolated, it is also known as stateless. The client should attach all required parameters to each request, as the service does not keep and use the data received by old requests. Resource management using REST principles can be performed by HTTP methods described in the previous section. All methods should be designed on the server side, clients should satisfy these defined requirements to successfully make requests. Methods which are used only for data reading are also called as safe methods, they are GET and HEAD, with no functionality to change the state of the server at all. Methods which have the same effect on the server and resources regardless if they were requested once or multiple times are also known as idempotent, they are HEAD, GET, DELETE and PUT. The standard supports exchanging of messages with structures represented by the Extensible Markup Language (XML) and the JavaScript Object Notation (JSON). (Richardson & Ruby, 2007, 23-47.)

REST standard provides simple representation of resources at the server side and provides efficient and scalable data access and management. It is convenient for the development and nowadays it remains as the most common standard for web development. This technology is quite lightweight, however it is powerful. RESTful services support wide range of the data formats described in the HTTP protocol. At the same time this technology does not keep transaction states from session to session, it is fully stateless. For some purposes it could be considered as disadvantage and require building extra staff on top of it.

3.1.3 Simple Object Access Protocol

This protocol is optimized for transferring structured data over networks, commonly it is referred to using the SOAP acronym. Here I will introduce fundamental features and working principles of this technology.

SOAP specification represents exchanging of XML messages over HTTP. Originally SOAP utilized the Web Services Description Language and the Universal Description Discovery and Integration standards, common acronyms WSDL and UDDI respectively. Both of these standards are used to describe services to make their discovery more efficient, however all SOAP functionality can be built programmatically using libraries, in some cases using these descriptions is optional, as most programming libraries which are optimised for SOAP endpoint publishing include automatic generating feature and service providers may avoid writing of extra service metadata manually. XML standard is not bound to a specific platform or software vendor, which makes it a cross-platform technology. With XML, it is possible to format the heterogeneous data and transfer it following the strict structuring rules. Paying attention to the data structuring should take place on both sides over which the interaction is expected to be established. Figure 3 gives example various ways to structure the same data. Different cases are acceptable, however the receiver application should expect the variant to be defined by the sender. Each message is also named as the envelope contains the header and the body. The header consists of blocks which define the data about message and describe processing methods. The body is the container for the transferred data encoded with XML, it consists of parent and child entities also named nodes. (Tidwell et. al., 2001, 21-36.)

SOAP based services have a strict structure, each method which is included in the service is published separately from others. This technology is suitable for the situations in which the data flow has to follow strict format rules. Furthermore, SOAP envelopes have all the metadata about the message which gives detailed instructions about how it should be processed. However, SOAP technology is more complex in comparison with sending the data over the HTTP directly, as the development and maintenance can take more effort.

Figure 3. XML structuring

3.2 Web service development with Java

In my experiments and service developments related to this study I use the Java programming language. This language is fully consistent with principles of object-oriented programming. According to IBM (2015) the architecture of this language is designed to be convenient for development, compilation and debugging. Also it is optimized for the building of modular software, which allows for a high possibility of code reusability. Java is an independent platform and has an understandable structure. Nowadays this language has extremely high popularity amongst software developers. Contemporary Java versions also have a great amount of libraries. The repository of libraries is rich, referring to them instead of manual creation of methods accelerates the development process drastically.

The Java platform has high execution speed, and it's robust design provides fast installation and delivery to the market of the developed software. This platform is suitable for long term usage, has high resource and cost efficiency. Java follows the principle of 'compile once and run anywhere', this feature makes it possible to update the software without changing the code or compiling it multiple times. For example, developers can move hard-coded parameters to the external properties file, which can be updated after the software installation,

or several property files could be created for different package types such as testing and production packages. High performance of Java also allows for efficient data management, with a particular engine named garbage collector taking care of resources which are no longer used by the program. The garbage collector of Java simplifies the establishing of data management during the development process. (Cinnober Financial Technology AB, 2012.)

There are building tools for Java applications which help to build ready executable files and import external libraries and files to projects. According to Loughran & Hatcher (2007, 5-19) the first building tool is named Ant, which was presented in 2000, and soon became very popular. Developers using this tool can declare Java libraries in the XML file instead of downloading files and manually adding them to the classpath of their applications. However Ant has disadvantages, building scripts have big sizes even if applications are small and simple, also it relies on the Apache Ivy dependency manager. Varanasi & Belida (2014, 1-15) provide a very detailed description of another building tool named Maven. It is the improved follow-up of Ant. Maven still uses XML files, however with simpler structure. The main advantage of Maven in comparison with Ant is that it performs library imports with a network instead of Apache Ivy. Java building tools can be separated into two main types: imperative and declarative. The first of these provides instructions and process definitions to the system, of which Ant is a good example. The second type helps to find the best way in the goals achievement, which is the principle followed by Maven. Based on a book by Mintra (2015, 4), we can say that Gradle building manager combines the flexibility and powerful abilities of Ant and the convenience of use and the efficient software life cycle management of Maven. Gradle uses the Domain Specific Language (DSL) named Groovy instead of XML.
In this study I need to implement both REST and SOAP web services.
The recommendation system relating to research has different purposes of the data exchange, for external API endpoints I will make REST web services, and for the mobile-server interconnection I will develop SOAP services. There are several ways of designing, building and deploying web services developed with Java, here I would like to describe two common ways of Java web service development. The main difference in designs is publishing of the service endpoints. Publishing can be established by the web servlet container if we have deal with a web application. Also there are specific frameworks which have inbuilt publishing

features; a good example of this kind of tool is the JAX-WS, it is the Java library which has a lot of useful components for the web service development.

3.2.1 SOAP services

We can encapsulate many methods with heterogeneous functionalities inside one service, so SOAP projects can have just several services which are full of methods, which means that we do not need multiple resource addresses and can use a single publisher.

First, we need to build a new Java project which will use the Maven building tool for development. We can choose a Maven framework during the creation process or set up it later in the project settings. When the project is created, all identifications and names are set, we can add dependencies which we will use in further developments, to do that we just need to declare them in the Project Object Model (POM) document. Figure 4 shows declared dependencies in the POM file: the first one can be used to establish the interaction with the PostgreSQL database, the second for reading, writing and processing CSV files, and the last one we can use for the web service building.

```
<dependencies>
    <dependency>
        <groupId>org.postgresql</groupId>
        <artifactId>postgresql</artifactId>
        <version>9.4.1208</version>
    </dependency>

    <dependency>
        <groupId>com.opencsv</groupId>
        <artifactId>opencsv</artifactId>
        <version>3.3</version>
    </dependency>

    <dependency>
        <groupId>javax.jws</groupId>
        <artifactId>jsr181-api</artifactId>
        <version>1.0-MR1</version>
    </dependency>
</dependencies>
```

Figure 4. Maven dependencies example

Following the object oriented principles, we can define the generic interface with methods which have standard characteristics, in the example it looks useless, however in a complex project we can implement similar interfaces and reuse defined parameters instead of recreating them. Then we need to create the class which has the same structure but is customised; in Java this methodology is named class implementation. In a simple Java application we need to have the method from which the running process starts, it is also named as the main method. In this method we place the component which publishes the endpoints and binds them to the service. Figure 5 illustrates the implementation of the web service publisher and methods.

Figure 5. SOAP web service example.

This kind of web service allows us to structure our data properly, where each parameter has its own place in the request envelope. Additionally, with Java we can enforce our service with extra security protection, there are many libraries for the security purposes which help us to with data loss and compromising. Services with this type of protocol are more organized from the data management view, however it brings additional complexity.

3.2.2 RESTful services

This type of web service has a simpler structure: each method of the data management should be declared at a specific address, usually services which are directed to the same business strategy are located at the same domain and separated from each other by subdomains. Let me introduce an example of the RESTful web service running in the container of the Java web application. To do this we need to create a new project as a Java web application and add the Maven framework to it. In this case all required configuration files and folders should appear at the project structure. The “web.xml” file should be placed under the “WEB-INF” folder. This document contains main settings of the web servlet component which is responsible for publishing the web resources and services. Figure 6 shows the structure of the Java web application. The “java” directory contains regular Java classes for the logic implementation, the “webapp” folder includes configurations and web pages which represent the graphical interface, services do not usually have any graphics. The “resources” directory usually contains files required for the application, for example images, files with properties, documents and extra settings. When the application is compiled, files for deploying are located in the target folder, for example bain built file with .”war” extension, this acronym stands for the web archive.

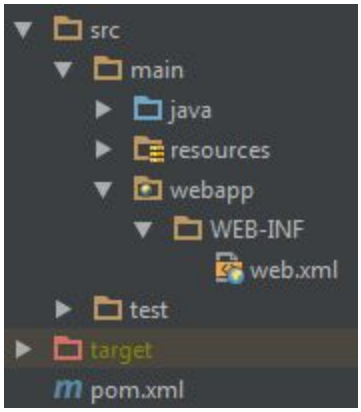


Figure 6. Structure of the Java web application.

GET and POST methods of RESTful web service are of interest because they are used more frequently than others in general and in my experiments, both of them are illustrated in Figure 7. The GET method receives parameters from the address path directly, in the example it gets the input value of degrees celsius and returns back the XML document with the fahrenheit converted value. Services can consume different input formats such as XML, JSON, text, mixed multipart or from data. Input, which comes with the request, can be mocked or converted to the Data Transfer Object (DTO) for convenience of use, as is shown in the example.

```
@Path("{cels}")
@GET
@Produces("application/xml")
public String getMethodExample( @PathParam("cels") Double cels )
{
    Double fahrenheit;
    Double celsius = cels;
    fahrenheit = ((celsius * 9) / 5) + 32;
    String result = "@Produces( \"application/xml\" ) Output: \n\nThe output: \n\n" + fahrenheit;
    return "<testservice>" + "<input>" + celsius + "</input>" + "<output>" + result + "</output>" + "</testservice>";
}

@Path("/json/example")
@POST
@Consumes( MediaType.APPLICATION_JSON )
@Produces( MediaType.APPLICATION_JSON + ";charset=utf-8" )
public Response postMethodExample ( Model obj )
{
    String result = testMethod( obj );
    Integer result_code = 200;
    return Response.status( result_code ).entity( result ).build();
}
```

Figure 7. GET and POST RESTful service examples

3.3 Android development

The development of applications for the Android system is mainly based on the Java programming language. Another option is C# language, however in most cases it is used for the game development with Unity platform. Android provides software development kit (SDK) tools for the Java code compiling and building of Android package kit files (APK), which have ".apk" extensions and are expected to be installed on mobile devices. The kernel of Android relies mainly on the Linux operation system which means that the platform follows functional patterns of that operation system. Every application receives it's own space on the device, and gets a separate process for execution. Processes are grouped in stacks, where each has its own place according to the usage order. Applications which are not used come to the end of the stack and get less priority, if there are not sufficient resources, the system will kill the processes with low priority in first order. (Android, 2016).

Applications can reserve the storage space on devices and store their data in two ways: internally, which means that only the application which owns the data has permission to use it, or externally, which means that other applications can gain access to the data and modify it. According to the security requirements we consider the internal data storage type. Every application obtains its own permissions of resources usage. Basic settings of the application such as definition of the application start and permissions are defined at the "AndroidManifest.xml" file. For example, these permissions could be: sensors management, keeping screen awake, Global Positioning System (GPS), access to media files and others. When the user attempts to install the application on their mobile device, they are prompted about these permissions, which should be accepted for proceeding to the installation as it is shown on Figure 8.

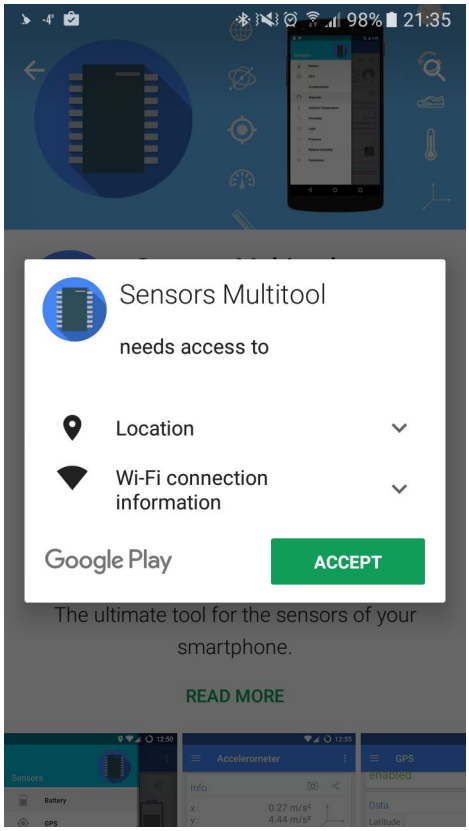


Figure 8. Permissions prompt.

There are several patterns within the Android GUI development. The key question of the GUI is usability; graphics should be understandable and convenient to use. Users should receive main functionalities of applications immediately after installation avoiding time wasting. The graphical user interface (GUI) of the Android application is defined by the XML file, also known as the layout. Allocation of layout parts can be done in different ways depending on the design requirements. Approaches differ from each other in allocation relations between the different child elements, within the parent element, also known as container. For instance, the linear layout places elements inside it following vertical or horizontal directions. Relative layout defines the positions of child elements by relatively separating the space between them. In absolute layout, all positions are strictly predefined. For more details about layout types you can follow the official Android documentation. There are different types of elements which could be declared in the layout document, for instance, text, button, edit text, checkbox, radio button, timepicker and many other views. Each of them should be declared with a particular type. Each view should have the identification value which is needed to refer to and

manage this view from other application parts. In the following example we can see the declaration of the relative layout and simple text view inside it. Using the layout tag, we can define how much space should be kept between it and each edge of the screen, which programmatic management context corresponds to the layout and which tools should be imported to the document.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.mrumiantcev.testapplication.MainActivity">

    <TextView
        android:id="@+id/my_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
</RelativeLayout>
```

Figure 9. Android layout example.

The basic GUI controller is the activity which is represented as Java class with reference for a particular layout. Activities have a definite life cycle, which begins from the activity creation, at this point we can define functionalities and declare views which should exist at the initial state, there is “onCreate” method for this purpose, it is also known as initial callback of the activity. For instance, we can define the layout and views which we are going to manage. The following example illustrates the implementation of the button with the text “PROCEED” on it, the function which is declared by the “onclickListener” starts another activity, or simply redirects us to another application screen.

```
public class Activity extends AppCompatActivity {

    Button button;
    Intent intent;
    Context context;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button = (Button) findViewById(R.id.startButton);
        button.setText("PROCEED");

        button.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View v) {

                intent = new Intent(Activity.this, MainActivity.class);

                startActivity(intent);
            }
        });
    }
}
```

Figure 10. Android activity example.

When the activity comes to the foreground and becomes visible for the user, this state is called the start of the activity. Here the application prepares all features for the interaction. The resume state occurs directly after the start, the interaction between user and activity takes place at this state. There are callbacks which logically have the opposite effect on the activity, they are needed to pause, stop or destroy the activity. Other methods can also be implemented in the activity following in same way that it can be done in the regular Java class.

Android service is the component which does not manage any activity, it performs all functions in the background. The data management in Android is supported by specific components, also known as content providers, which provide access to the internal device memory, where we can use the file system or the database for the storage of data.

Android platform provides the wide range of customized controllers for the GUI and resource management. There are many corresponding libraries which make the software development process efficient and convenient.

3.4 Semantic Web

Data sources on the web can relate to each other by some criterias and meanings however at the same time these sources might not be interconnected. Semantic Web is represented by approaches, frameworks and specific descriptive languages goaled to the data linking, making it understandable for both humans and machines. This technology is built on top of the web and is able to use data from different sources as if it would be placed in the same data model. Semantic web aims to automate the processes related to integration establishing of data sources. Resource metadata annotations, which can be processed by machines, pushes web technologies to work more independently, smartly and in a self managed manner. Semantic methodologies make significant sense when used in data exploration by humans, and for specific tools or engines. It also provides the possibility of processing this interconnected data by heterogeneous entities such as people, different flavours of software and services. (Khriyenko 2008, 23.)

Contemporary web systems became complex and sometimes it would not be possible to manage resources in a manual manner. There is need of smart and self managed software. Semantic web plays one of the most important roles in the future of the internet, tripple based technology describes subjects more naturally and provides possibility to link data between systems and integrate them. According to Khriyenko et al. (2008) Internet of Things (IoT) devices can be managed with centralized middleware based on semantic web technologies, efficiency of storing and retrieving the data in this case rise drastically. Values from sensors of IoT can be classified and described in a meaningful manner instead of indexing approach.

Contemporary web are consists of a great number of resources, all of them produce big amount of data continuously. These resources such as social networks newspapers and media sharing web sites exchange the data between each other. Some resources at particular time periods have inappropriate or not completed data, in case if this data would be extended, other sources which rely on it might need hard updates which is not cost and time efficient. With semantic web we do not need to store the data which we retrieve externally, we only refer to it with links and do not need to handle updates of it. This technology is very suitable for building informational modular ecosystems for the data management. (Khriyenko, 2011).

Web resources have to be located at specific addresses, these unique identifications are required because resources tend to be fetched and managed via networks. A Uniform Resource Identifier (URI) is an address which identifies specific resources by their locations and names them separately or in combination. The definition of the representation is not obligatory in this type of identifier, however it might be applied. Uniform Resource Locator (URL) is a type of URI, which defines the exact location of a particular resource and includes instructions describing how to access the resource. URL, simply put, is URI which includes the extension of the resource. The third type of the identification of web resources is Uniform Resource Name. It is focused on the namespace of the resource. The idea of the URN is for unique identification, obtaining instructions are not considered for this type. Examples of identification types are shown in Figure 11. (W3C Notes 2001.)

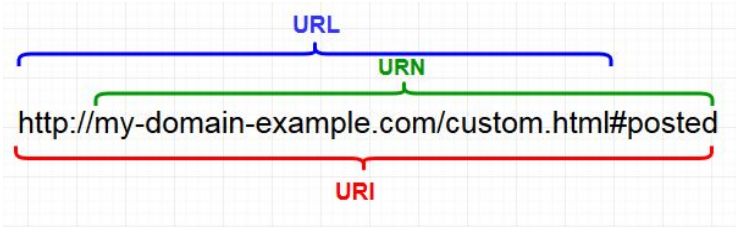


Figure 11. Resource identification types.

The main principle of the Semantic web is annotation of web resources with machine readable metadata, which is achieved by representing them with the Resource Description Framework (RDF) data. The structure of the semantic data is defined in ontologies written with Web Ontology Language. Ontology is like a "spine" for resource annotations, it defines properties, resource types and relationships. Semantic data entities are represented as subject, predicate and object triples. The subject part contains the resource definition. Object stands for another web resource or the literal value. Predicate is used to describe the relationship between subject and object. There are different syntax languages of the RDF data representation such as Turtle, Notation 3, N-Triples, Trig and RDF/XML. Cochez (2012) in his study provides detailed description of the languages which are used for the management and representation of the semantic linked data.

Linking the data approach of semantic web technologies brings as to interconnect and integrate heterogeneous data sources. It is hardly used for integrations of systems with different data models, because the linking approach is very flexible and able to handle as much conditions at a time as it would be required. The number of properties presented in the semantic data model does not have great influence to the reasoning process, of course the iteration between them and processing might take longer time in case if there are too much of them, however in comparison with SQL data storing approaches it will not require handling of huge tables and databases, just sets of links and properties.

To conclude this section I want to say that semantic web technologies bring flexibility, interoperability and extensibility for data processing systems and their features will be applicable for many systems in the future.

3.5 Artificial intelligence

Nowadays big players in IT such as Google, IBM, Microsoft and others are pretty much focused and progressing in the direction of cognitive computing heavily applying Machine Learning, Neural Networks, and other Artificial Intelligence (AI) methods. Intelligent machines and services are expected to manage themselves and make proactive actions according to the changeable environment to address upcoming issues and increase the performance efficiency in achieving their goals. A system which takes into account opinions and preferences of thousands of people has to be self-managed and flexible, because the emotional factors of people which drive these services have a dynamic nature and in this case the manual service and data management would require significant efforts or even could be impossible. Establishment of a machine self-learning process is one of the most important goals of this study. To achieve it we need to design our services that way so they will be able to retrieve data from the web for further proceedings on demand and on the fly without our help. However, as data growth has extremely high speed nowadays, unfortunately there are a lot of irrelevant data for machines. This data irrelevance is caused by interoperability problems and lack of interconnections between data sources. In this section I describe approaches and technologies which are targeted to address these issues and bring efficiency to

the data retrieving and management processes. Smart and self managed software is is in demand nowadays and will will continue gather momentum in the future.

3.5.1 Machine learning

Learning process of machines means gaining experience of devices which will have effect to further behaviour and data processings. There are different types of machine learning algorithms. I want to highlight two main categories of machine learning: supervised and unsupervised.

First group is represented by algorithms which handle particular sets of the data with specific functions. Supervised learning algorithms are targeted to the mapping inputs and outputs of functions. For example, we have a function and input, supervised learning algorithm will predict how the output will change depending on changes occurring in input. This methodology is called supervised because it requires a teacher which would provide the data sets to inputs.

Unsupervised machine learning means processing of the data with heterogeneous structures and unlabeled values. The behavior of unsupervised learning systems rely mostly on gained experience rather than variations of inputs. This type of learning is widely used in text analysing, face and voice recognitions. Google made great research in this topic and hardy utilizes principles of unsupervised machine learning in its services.

Good example of machine learning are neural networks. They are represented as set of components which exchange input and outputs and affect to each other. These components simulate behaviors of natural human neurons. This technology is used for various purposes such as seismic signal processing and recognitions and classifying heterogeneous data sources. Neural networks include learning processes, validations, measurements of uncertainties and errors and many other things.Processing unit of neural network is neuron which has several inputs, processing unit and output. These components are linked together neurons use outputs of others as an inputs. Natural neurons receive inputs and trigger outputs depending on magnitude and strength of inputs. (Soares & Souza 2016)

4 DATA PROCESSING

According to key domains of the music adviser service related to this study, we may highlight two fundamental modules of the data model. The first one relates to the information about users, it is targeted to define states of their emotions and activities. The second module includes the music metadata. Analyses of these data entities allows us to define correlations between human feelings and music to which users listen. In this chapter I am going to describe the data collection approaches from heterogeneous sources which build the knowledge base of our service.

4.1 Personal data

If we are going to make personalized recommendations for someone within a particular life situation context, we need to know about that person. Let's investigate how to retrieve this kind of data.

4.1.1 General personal data

This type of data does not depend on any context, it represents a constant physical or real description of the person. It can be gathered from the person by offering forms at the initial stage of using the service. Basic metadata might not require any confidential information which identifies the person, however authentication data is still needed to process uniquely personal profiles. According to ethical rules and permissions provided by the user, some extra data can be included in their personal system account. General data forms could have fields asking about gender, age and location such as country name and postcode. In the case of the mobile version, a frontend type authentication could be established based on the device ID, account migration through different devices requires updating the system with multiple device numbers, for example phone and smart watches could be bound to a single user profile. Transition to the web frontend application still requires providing identifiers such as email or phone number in combination with password. The mobile only authentication process avoids

filling login fields even if the application data is erased on the device, backend can handle all steps based on earlier provided device ID.
Mobile authentication process:

1. Client: Send authentication request (with unique device ID)
2. Server: Check internal matching of the device ID.
    A) If matches => sign-in accepted.
    B) If not => sign-in is not accepted.
3. Client: Receive response from server.
    A) If authenticated => store received data on client device, redirect to the player screen.
    B) If not => redirect to sign-up screen => sign-up request.
4. Server: In sign-up request case store the user data internally => authentication accepted response.
5. Client: Repeat step 3.

4.1.2 Sensor-based data

The front end of the service is based on the mobile platform, contemporary mobile devices have a wide range of sensors, from which we can enrich the system knowledge with the collected sensor data. In this section I want to investigate how this kind of data is useful for the emotion and activity recognition process. The Android platform is considered for this study, so I will first introduce how sensors of Android-based devices can be utilized to predict the emotional state and activity of the person. Technical descriptions are tightly based on the official Android documentations for developers. (Android developers, 2016)

In general all Android sensors can be separated into three groups. There are motion, position and environmental sensors. Motion sensors are targeted to measure rotation of the device along axes of the three dimensional model. Environmental sensors are related to defining of the environmental conditions such as temperature, humidity, pressure etc. Position sensors aim to determine the orientation of the device and measure magnetic parameters to define its position according to compass points. Sensors from each category can be hardware or

software based, which means that sensors of the first category provide output results directly from physical measurements and others are based on computational processing of the data.

At first we need to define which sensors are available, and we can do this programmatically. Our data model should be flexible, according to the existence of sensors it should adapt itself to further computations. This part is important because different models of mobile devices might have different sensors. Each sensor has a unique type, using this definition we can declare each sensor. To get all available sensors we use “TYPE_ALL” declaration to retrieve the list of sensors. Figure 12 illustrates the method for sensor listing and the output at console.

```
SensorManager m_sensor_manager;

m_sensor_manager = ( SensorManager ) getSystemService( SENSOR_SERVICE );

List<Sensor> list = m_sensor_manager.getSensorList( Sensor.TYPE_ALL );
for( Sensor sensor: list )
{
    System.out.println( "SENSOR : " + sensor.getType() );
}
```

```
I/System.out: SENSOR : K2HH Acceleration
I/System.out: SENSOR : GP2A002 Proximity Sensor
I/System.out: SENSOR : Screen Orientation Sensor
```

Figure 12. Get list of available sensors

Using data from sensors, we are even able to determine physical and health states of the person. Some devices such as smartwatches support heart rate and even blood pressure sensors. Retrieving these data we can monitor the well-being of the user continuously, which provides a great tool for traditional treatment purpose, from one side the system can be interconnected with health care center and share the data with personal doctors, from another side it enforces our music adviser system and can make the curation process more effective.

The accelerometer measures acceleration which occurs with the device in a three dimensional coordinate system. Using the data coming from the accelerometer we can detect specificities of the device movement for example shaking. There are other devices which have similar

purposes. Gravity sensor measures gravity force at the coordinate system. Gyroscope captures rotations around each axis in radians. If we need to work with a single axis there is linear type of the acceleration sensor for this purpose.

Smartphones and smartwatches are devices which we keep with us almost all the time and by these devices we can perform measurements continuously. Particular activities of the user are represented by specific motions of different body parts. For example if the phone is in the pocket during walking it follows movements of the leg and these movements are common for all people with minor differences depending of the walking style. Following the same logic devices can detect sports and exercises, specific movements of hands on a wheel during driving, and specific vibrations if the device is attached to the glass of the vehicle.

Figure 13. Sensor - based activity detection

When the application has information about the available sensors it can call them by specific sensor type definition. To monitor the data from sensors we need to implement a particular method which will listen to selected sensors and capture changes of the data which occur continuously, on the Android platform this kind of method is called the listener method. Figure 14 shows the implementation of the method, following patterns of the clear code we can put declaration of the variables which hold values received from sensors to a separate

method, where we print them to the console. This is a fundamental implementation of the accelerometer listener just to show how sensors can be used programmatically.

```
@Override
public void onSensorChanged(SensorEvent event)
{
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER)
    {
        getAccelerometer(event);
    }
}

private void getAccelerometer(SensorEvent event)
{
    float[] values = event.values;

    float x = values[0];
    float y = values[1];
    float z = values[2];

    long actualTime = event.timestamp;

    System.out.println( "X : "+ x + "; Y : " + y + "; Z : " + z + " - Change time : " + actualTime );
}
```

Figure 14. Sensor listener implementation

Below we can observe the result of the application running process with the code which is illustrated above, here we can see acceleration values relating to each of the axes and timestamp of the value capturing in milliseconds which can be converted into regular datetime format.

```
I/System.out: X : -0.18315643; Y : -0.81163436; Z : 9.758766 - Change time : 717744307752236
I/System.out: X : -0.22146365; Y : -0.59735334; Z : 9.244013 - Change time : 717744507752496
I/System.out: X : -0.19033903; Y : -0.6500258; Z : 9.903615 - Change time : 717744707748121
I/System.out: X : -0.17597382; Y : -0.36631286; Z : 9.502586 - Change time : 717744907802444
I/System.out: X : -0.11133038; Y : -0.095768064; Z : 9.828198 - Change time : 717745107748798
I/System.out: X : 0.077811554; Y : 0.49200845; Z : 9.317036 - Change time : 717745307669891
I/System.out: X : -0.2813187; Y : 1.2856863; Z : 9.194932 - Change time : 717745507745881
I/System.out: X : 0.92057055; Y : 1.5981296; Z : 10.509348 - Change time : 717745707778173
I/System.out: X : 0.9720459; Y : 2.5246856; Z : 9.126697 - Change time : 717745907681975
I/System.out: X : -0.10654198; Y : 4.4675803; Z : 7.547721 - Change time : 717746107758016
I/System.out: X : 0.48602295; Y : 4.7764325; Z : 8.073248 - Change time : 717746307780568
```

Figure 15. Accelerometer output

Taking into account that particular activities have common movements with minor uncertainties, using multiple samples and measurements we can create common activity templates, further application will compare ongoing movements of the person with these templates and check their matching to predict activity cases. There are a lot of approaches to compare these kinds of data. For example, we can represent collected data as graphs and perform alignments or analyses of their correlations. Visualized content of the sensor data is represented in Figure 16.

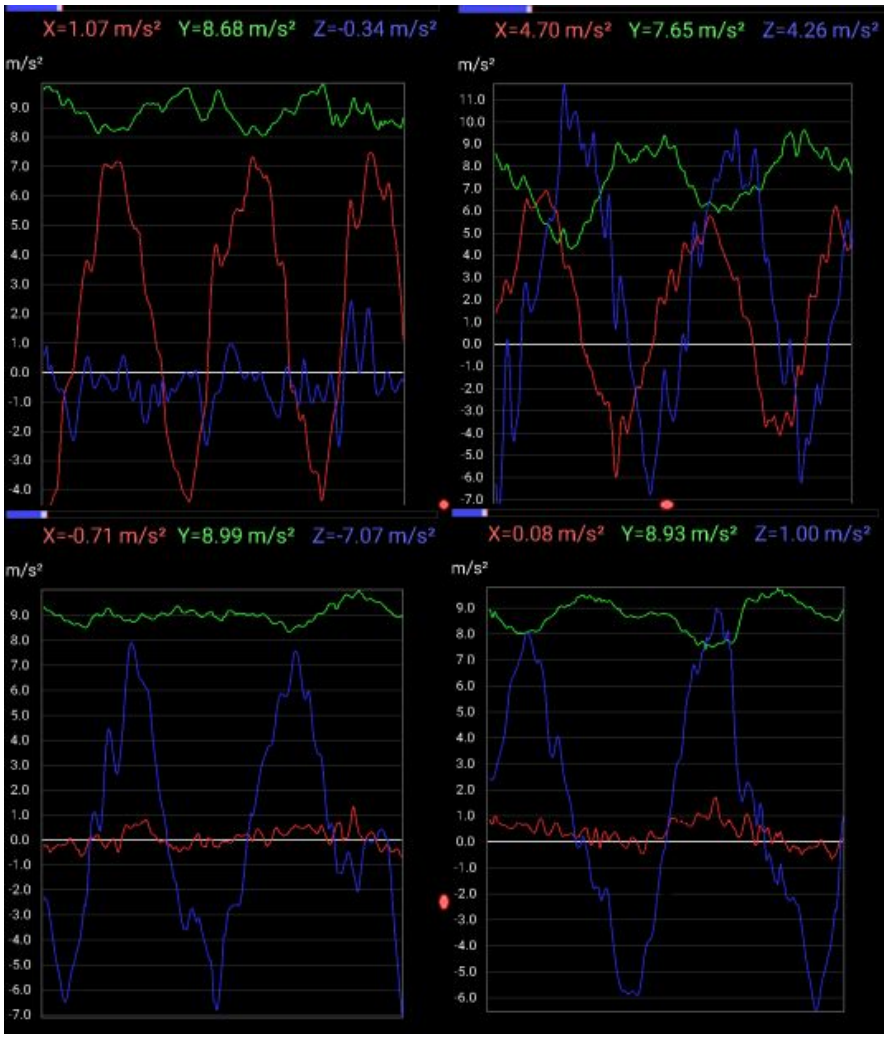


Figure 16. Accelerometer graphs

Most contemporary devices have Global Positioning System (GPS), which can significantly increase the activity prediction ability of our system. With this feature we can get the current location of the device and determine the movement speed. By using map APIs we can define

what is located near the device. For example, if the location matches a highway and the speed is high the system can guess that the user is travelling at the current moment. Following the same logic we can check if the person stays at cafe, gym, museum or at some other public places which have relevant activities.

To work with the geolocation data on Android devices we need to declare the location manager object which is included in the standard Android SDK and retrieve the location entity. Location can be determined using a two dimensional coordinate system and we can get latitude and longitude of the particular point on the map. Fundamental logic of the method implementation is shown in Figure 17. In real development we might need to add some extra features such as an Internet connection check and a GPS enabling feature.

```
public Location getLocation() {

    LocationManager locationManager = (LocationManager) mContext
            .getSystemService(LOCATION_SERVICE);

    locationManager.requestLocationUpdates(
            LocationManager.GPS_PROVIDER,
            MIN_TIME_BW_UPDATES,
            MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
    if (locationManager != null)
    {
        location = locationManager
                .getLastKnownLocation(LocationManager.GPS_PROVIDER);
    }
    return location;
```

Figure 17. Getlocation method implementation

To track changes in location we need to implement location listener method where we can continuously receive location updates. Example of this kind of listener is shown in Figure 18.

```
@Override
public void onLocationChanged(Location location)
{
    location.getLatitude();
    location.getLongitude();
}
```

Figure 18. Location listener method

To implement GUI of the map we need to add “MapView” to the XML layout file which is related to this logic implementation. Figure 19 shows the output of getting current location of the device and nearby objects on the map.

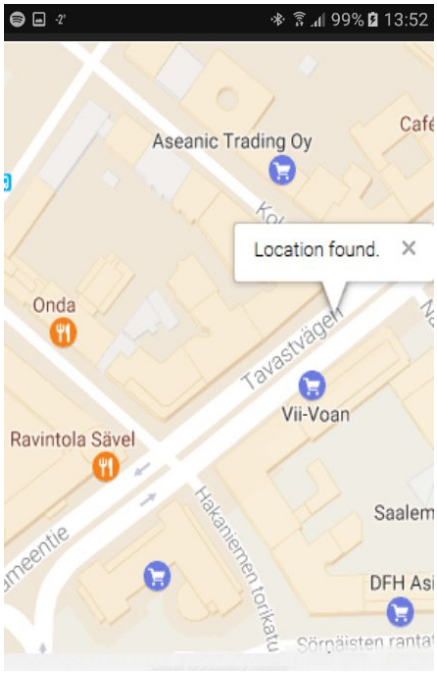


Figure 19. Current location view

4.1.3 Social-based data

Another way of collecting personal data is available through binding other accounts of the user in social and media networks. We can emphasize two great advantages of this method. Firstly, the user is not bothered by filling out extra forms and are able to enjoy the services simpler, logica. Secondly, the system can gather more personal data in an automatic manner, according to permissions provided by the user.

Social networks provide endpoints for implementing login features integrated with their services. One of the most popular social networks, Facebook, provides a wide range of services which support integration with personal data of their participants. Facebook API login provides identification of individuals with their public accounts which have reliable

data. This service is available on most common platforms such as IOS, Android, Windows phone and Web. Through this approach we can retrieve more data in comparison with providing basic web forms. At the same time all personal data sharing from Facebook to external services can be established only under permissions provided directly by users. Facebook API supports about 30 permissions over the data related to friends, relationships, public data, messaging, actions of users in communities, some user experience and preferences and many others. More detailed API instructions can be found on the official company page. (Facebook 2016).

Comments and texts which users share on the web are able to reflect emotional conditions at a particular time period or personal point of view about some topic. For example, we can analyse written text and define the main mood of it. For these purposes it is not necessary to develop and apply our own text analyzing algorithms, there are a lot of APIs which provide endpoints of text processing services. The set of tools and applications for the emotional states detection is not limited by text analyzing approaches, there are systems which are expected to be interconnected with sensor systems, most APIs have algorithm-based architectures and perform sound, face recognitions based on request inputs coming from the client side.

One of the most powerful text analysing tools is provided by IBM Watson, it is named Tone Analyser. It is designed to determine emotional keys, social bases and language styles of the text. With this API we can receive psychological values from text sources such as happiness, disgust, angry and others. We can test this analyzer for free, however if the number of calls for this API will not exceed one thousand per month. Extra features such as security, extra API calls require payment. Endpoints of this service are represented as RESTful based services which can be accessed via HTTP GET and POST requests. It accepts JSON request attachments where we can send our texts to get response with analysed values. This means that for the text analysing purpose we need to develop only the client application. (IBM Watson, 2016).

The existence of such kind of systems is not limited by the IBM vendor, Google and Microsoft corporations provide APIs with similar functionalities and services.

Figure 20 shows the response which this service returns for the request with attached text. It includes emotional, social and language based analytical results. Texts of comments and chats can reflect the current emotional state of the person. However we need to pay attention to the ethical clearance during establishing of these analytical processes in respect to the personal data confidentiality and nondisclosure. Logically this approach expects retrieving personal data from social networks and provide it for external analyser service, it might require complex agreements between different parties. In this study I demonstrate only the possibility of this methodology implementation.

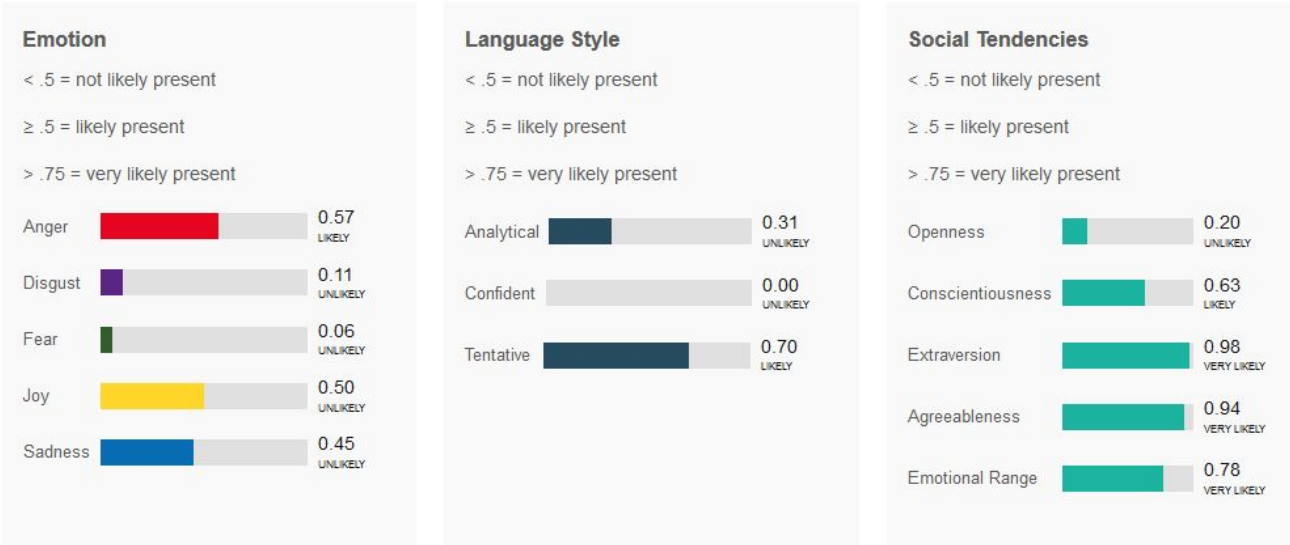


Figure 20. Text analyses with IBM Watson

4.1.4 Facial emotion recognition

People can easily recognise emotions in others when they look at faces, these skills are learned from early childhood. Could we train machines to do the same work? Facial expressions are powerful reflections of emotional state of a person. In this section I will discuss how images with human faces can be processed in order to detect emotions presented in them. Taking into account that nowadays a lot of people like to photograph themselves

everytime and everywhere they can suggests that taking a face snapshot instantly will be more convenient in comparison with filling in surveys. However, grimaces can spoil the analytical process, and in this case the output data can be irrelevant. When people make selfies, usually they make grimaces, these face expressions are not relevant that is why face capturing should be done in automated manner. Here I want to describe the general approach implementation, for proper analyses much more effort is required in respect to the detailed recognition of different facial parts. Generally, users understand that they take pictures to get better music curation services and logically there is no motivation to make grimaces, especially when they are sad, stressed and feel that need a psychological support. We need to remember that face capturings should be done in a sudden manner to avoid posings and irrelevant facial emotions, this aspect has to be reflected in the ethical clearance of the software in advance. In other words, we want to push the system from manual capturing of facial emotions to the environment base automated detection of facial expressions.

Human emotional state which is reflected by facial expressions can be determined with attributes such as happiness, sadness, disgust, anger, neutral or surprising. All of these in weighted percentages can describe the whole psychological picture at a single point in time. Different groups of muscles of a human face are responsible for emotional reflections. Face changes for specific emotions are common for most people, when we are happy we smile, when we are surprised our brows move upwards, otherwise when we are angry or sad. With color and shade analyses we can get face features and create key vectors of the human face. Figure 21 shows main points of smiley and sad faces. This approach is mainly based on the color recognition and comparisons of the image. However we do not need to create this feature manually and from scratch to detect facial expressions, instead of that we can use some ready solutions available in the internet. There are a lot of projects and even APIs which provide face recognition services.

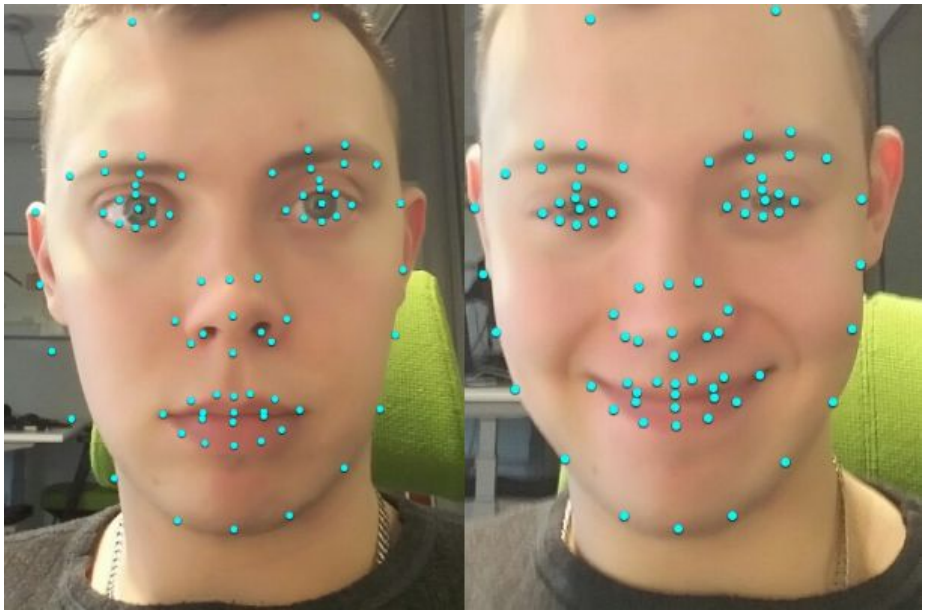


Figure 21. Facial expression key points.

Taking into account the scale of the image we can compose these points with points of emotion-specific templates and calculate metric spaces between them, then we calculate similarities between actual face points and points of the templates. Using this approach we define how near a face is to sad, happy, surprise, or anger templates.

However these analyses are complex and involve a lot of development and design effort, we can rely on existing services which provide emotional measurements of faces. Most of these are very simple in use, for example, according to the official documentation of the (Emotion Analysis API, 2016) we just need to attach the payload of the picture to the JSON message of the HTTP request and to receive emotional values in response. Figure 22 illustrates JSON messages of request and response samples. To transfer media data we need to encode it before transmitting, as the encoded value of the image is a long string to contain the picture.

```
JSON request :

"image_data": {
  "name": 523,
  "extension": 293,
  "width": 417,
  "height": 417,
  "payload": "encoded payload of the image"
}

JSON response :

"success": {
  "anger": 0.1354654,
  "contempt": 0.00011054,
  "disgust": 0.016444,
  "fear": 0.0000334655,
  "happiness": 0.000554555,
  "neutral": 0.67216755,
  "sadness": 0.10064454,
  "surprise": 0.000041111
}
```

Figure 22. Request and response for the facial analyses API.

4.2 Music data

The recommendation system should know as much as possible about music tracks, otherwise it would be impossible to make recommendations for users in case of classification based recommendation approach. For collaborative filtering all physical and constant music features do not have too much sense. In this part of the study I discuss existing data sources and information collecting approaches related to music fragments. The main question of this section is how to establish machine learning with music.

4.2.1 General data about music tracks

Fundamental music metadata describes music entities. Regular keyword searches can be established on this data. It includes basic descriptions such as track name, artist or lyrics author, album and the year of release. There are many web sources where our system can retrieve all required data. Large music data sources, such as The Echo Nest, contain metadata of billions of music tracks. We can retrieve all these metadata through HTTP API. The system

should be adopted to this kind of data sources and request the music data on demand, for example, when it encounters tracks which are not annotated internally in the system. However some parts of the Echo Nest services migrated to the Spotify music platform and currently are available there, later I will describe Spotify use cases. (The Echo Nest, 2016).

The DBPedia services represent the whole web resource ecosystem, which can be used to query and interconnect data from the Internet. It is designed for providing open and free data for users. Gathering semantic annotations of music entities from the DBPedia allows us to teach our machines with more human-friendly and accurate descriptions. Music curation for humans requires external support at the data collection. With DBPedia services we can enforce our curation services with a human knowledge base and significantly increase the data driven sense of the whole ecosystem. DBPedia is a linked data service and if we want to query endpoints of it we should utilyze SPARQL querying approach. (DBPedia, 2016).

4.2.2 Music features

The content filtering approach to music is based on the physical attributes of music. These features are constant and more objective than human feelings. This kind of data should be calculated by music data retrieving professionals with respect to commonly accepted rules and constants. Music analyzing is a complicated process and can be considered as a separate topic to this research. Instead, we can utilize existing services which already have databases with huge amounts of tracks annotated with music features.

One of the most popular and powerful music platforms is Spotify. It supports all common contemporary platforms, we can use software development kits (SDK) adopted for Android, IOS, Web and Applescript. I will now introduce the music features we can retrieve from the API endpoints of this service. The fundamental music feature is the duration of the track. The presence of acoustic instruments instead of electronic music is described by “acousticness” feature. At the same time presence of instrumentals in general should be paid attention, it is described by “instrumentalness”, the scale of this feature is a float value from 0 to 1, which tells us how a track combines vocal and instruments. Additionally, we can get “speechness” of the track, tracks with high amount of reading have a greater value of this feature, for

example, rap tracks or audio books. “Energy” and “danceability” determine how suited the track is for dancing, and how much energy it brings to human feelings, in some cases, for example, rock and roll music is very energetic and good to dance. Another similar variable is “tempo” music attribute. The tonality of the music can be determined by music value such as pitch key. However, tonality mode does not always determine the mood key of the music, track with the minor tonality key could be very positive in emotions and otherwise. The “valence” attribute relates to the emotional value of the music. (Spotify, 2016).

Spotify web API endpoints can be accessed with RESTful HTTP requests. The type of the request is GET, which might require an authorization header with service token, which should be requested beforehand. Figure 23 illustrates basic Spotify API utilization for retrieving music features.

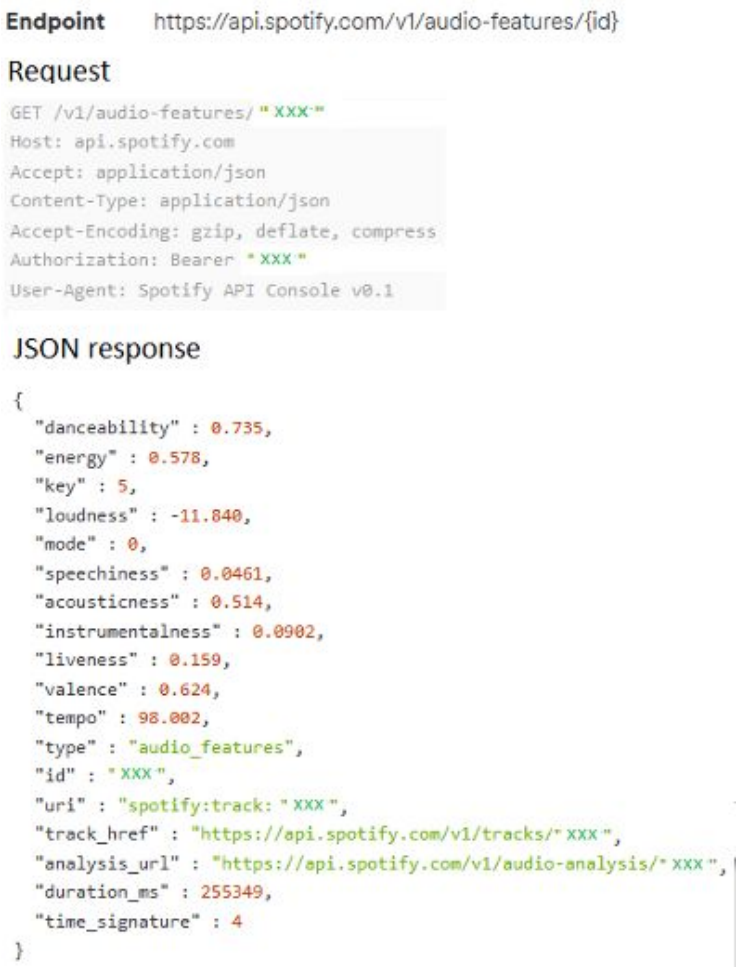


Figure 23. Music features Spotify endpoint

There is set of other sources and services where we can extract music features. Good example of such kind of services is Music Analysis and Feature Extraction Service introduced by Centre for Digital Music[7]. It has semantic web driven service endpoints of API. The research group also presented the software which works as client of these services, it is called Sonic Visualiser[8]. This service would allow recommender system to handle wide ranges of music files, it has extra plugins and services for music track annotation such as Sonic Annotator. This service accepts files from the file system and links for web audio resources as input and provides result sets of audio features as the RDF data.

4.2.3 User feedback

Feedback from users plays a significant role in the whole curation process, as this data is the main management part of the recommendation processes. We already discussed user feedback in relation to their emotional statements, and these approaches apply to users sharing their opinion about particular music tracks to which they listened.

The fundamental approach of getting music feedback is the like – dislike system. The main advantage of this is that it is not annoying and accurately defines what tracks and music features are suitable for the person in their specific situation. However, answers in this case could be caused by different reasons, for example, music features might fit desires within a situation, but the track may have been heard too many times for the user to enjoy it. Like – dislike questions could be asked only when user skips music tracks, but we then need them to determine the reason of skipping, because if user listens to a track from the beginning to the ent it means that all requirements of music are satisfied. User streaming services use unique identification numbers for music fragments, our system should keep links between music metadata and these identifications to allow users immediately use music services instead of simple recommendations.

---

[7] Centre for Digital Music - research group in the area of Audio Music Technologies. Was organised in 2001 on the base of Queen Mary University of London

[8] Sonic Visualiser - software for extracting music content from audio files.

Instead of storing all of the data internally the semantic driven engine aims to interconnect existing data and operate with links between data sources. In this case we do not waste resources with storing large data sets, instead we just teach our machine how to retrieve the

data on demand. This means there is no reason to store all of the metadata internally, instead we annotate tracks with links to DBPedia and other sources. The system should contain identification numbers for music tracks at different services instead of storing media files. But the learning process should be automatic, and the machine should learn by itself. Music streaming services provide a wide range of music, users can share the data of music which they listen to with the system and then based on received data, the machine can query different web resources to learn additional music data. This learning process could be established in continuous manner and on the fly. At the end of the listening session the user's device can send the following information to the server side.

1) User ID
2) Emotional and activity case definition
3) Music exploring case (basic search, offered list or mixed)
4) Set of played tracks with basic metadata
   4.1) Track name
   4.2) Artist name
   4.3) Music service related track ID
   4.4) Like or dislike signature. In case of track skipping.

Figure 24 shows the fundamental feedback processing and machine learning algorithm. At first the system receives a set of tracks with basic metadata, iterates through them and checks if they are familiar to the system. If the detailed metadata of the track is found in the internal storage everything is fine and the system then processes these data. If the track is unfamiliar to the system, it queries the required metadata for external sources and annotates the track internally for further data processing. The idea is to enforce the feedback with extra music metadata to create a particular profile for the specific emotion and activity case.

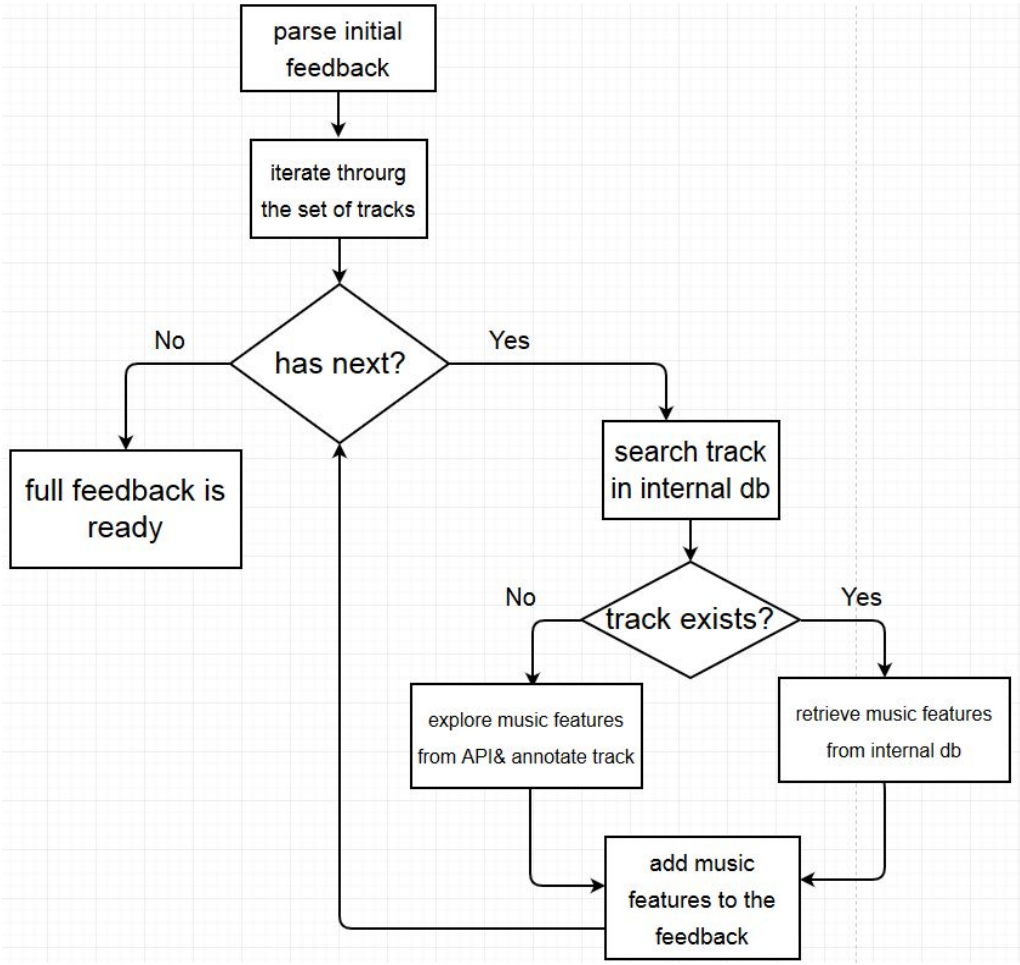


Figure 24. Feedback processing algorithm.

4.2.4 Music track filtering

When feedback is received and processed the system then knows the music features of all tracks which were listened. In this section I will discuss how to operate with these music features to select similar music tracks to those that were given in the feedback.

This involves content-based filtering, in other words, selecting tracks based on their physical constant attributes. Most of music features which I described earlier have float values in scale from zero to one, logically we can set up our filtering algorithm to capture ranges of each feature. Music tracks which have attributes with values within these ranges can be considered

as accepted and suitable for the particular context. Figure 25 presents the visualization of the content-based filtering process.

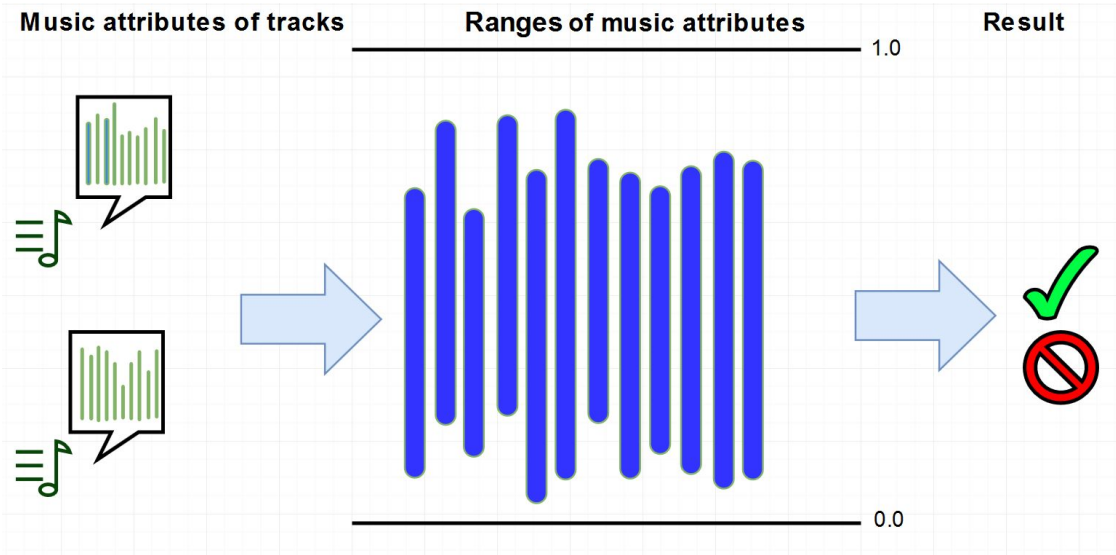


Figure 25. Music filtering process.

These ranges could be defined by minimum and maximum values, which were retrieved from the metadata of listened tracks. This approach could make sense if tracks are similar by their nature such as style. However, if most of them are similar in their musical features and few differ significantly, these few tracks could spoil the whole picture and the system will receive inaccurate data, which will not reflect real music preferences of the current profile. Another way to calculate filtering ranges of music attributes is to find weighted average values from all samples for each music feature and set thresholds around these values. In other words, we can calculate averages of music features and setup ranges of the filtering tolerance.

Another approach to track filtering is based on an interaction between users and the system, it is known as collaborative filtering, fundamentals of this approach are explained in the chapter of this study related to recommendation methodologies. Following patterns of this method we capture subjective ratings coming from the client side. By collecting user feedback we can determine the opinion which is generated by a particular type of person about music tracks which were listened to when he or she was experiencing a specific mood and activity case.

Later, in respect to this feedback, the system will be able to make recommendations for users with similar preferences and at same mood and activity case, simply by offering these tracks

if another person has not listened them. In semantic data management the service should just add links which define that users belong to a particular cluster based on their personal data and preferences. As the service does not initially have any collected data from users, this is a cold start problem of collaborative filtering. However, we should provide recommendations for users immediately when they start to use our service, otherwise our system will not be robust and competitive. To solve this problem, first recommendations could be performed based only on activity cases. In respect of music features there are groups of music tracks which are suitable for particular activities. For example, music with high temp and energy is good for running or other kinds of sports, otherwise soft nature tracks are associated with relaxing, classical music is good for concentration. Some music streaming services have already implemented recommendations based on the activity case alone. Using spotify API we can retrieve music playlists which are prepared for specific activities. Therefore, in summary, at first the system has to make recommendations based on the activity of a person, in respect to the lack of data and we have two options of data processing design of this service part: generate lists by our forces or retrieve ready playlists from web services. Using external APIs should be done in respect to the commercial rules and agreements provided by these services.

The amount of music tracks provided by music services is huge and it will take too much time to iterate through all of these songs. To avoid that we have to establish lower level filtering of tracks before processing music features. This is possible by exploring music which is associated to similar artists and styles. Artists also have the metadata based on which we can consider similarities between them. This kind of similarity can be determined by intersections of main music styles of artists, era or period of their peak popularity, and some other variables. This approach is outlined in Figure 26.

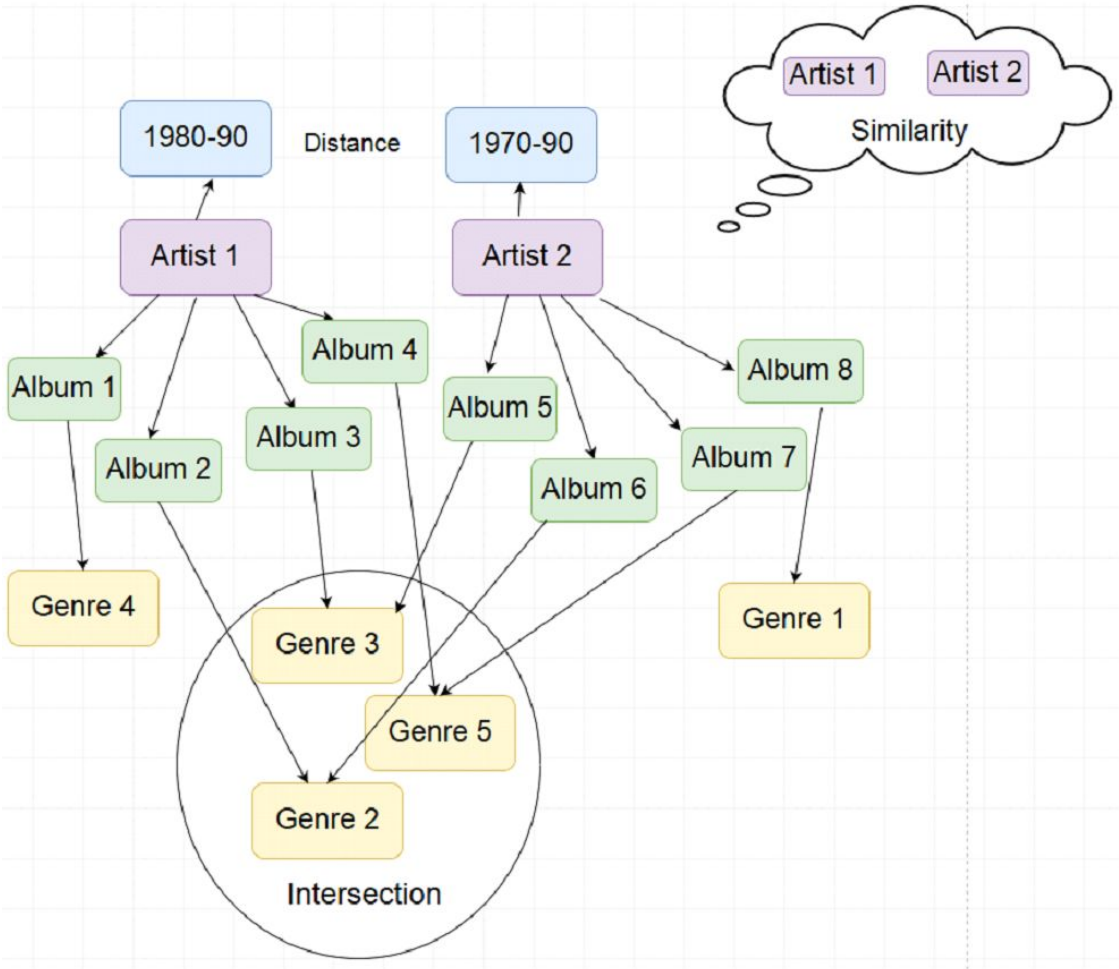


Figure 26. Similarity between artist.

It is very promising approach to apply Neural Networks to build a personalized emotion-based music preference model based on variety of music related and contextual sensor data. However, due to lack of big enough training set of data, which is usually required for Neural Networks, it was not feasible to apply such approach now, it can be considered for a future improvement of this system when collection of more data for training can be established.

4.2.5 MuPsych research

The cold start problem of the collaborative filtering can be addressed by the data existence before the start of the service. Will M. Randall in his music psychology (MuPsych, 2016)

research created studies which allowed for the collection of music related feedback from respondents. One of the most important aims of this study was to investigate relationships between well-being and emotional states of people and music to which they listen. In other words, the core of that research is to understand how music affects humans. I took part in that project as software developer and the fundamental idea of the topic of this thesis was based on that research.

The MuPsych project consists of a mobile application which detects when a person starts listening to music on their mobile device, and sends notifications of surveys, prompting the user to answer the given questions. The application captures all music tracks which are listened to by the user. Additionally, it includes questionnaires which are not related to the music, but which are needed to determine the overall emotional well-being and the traits of the person. This feedback is collected by the web service and can be downloaded by the researcher with a web application. The application sends notifications at three times during a music listening session: at the beginning, after five minutes and after ten minutes. Answers for the first questionnaire determine initial arousal, energy, emotional and activity conditions. The second questionnaire is the extended version of the first one. It includes questions about reasons for listening, how familiar the user is with the music and how it affects their emotional state. The third questionnaire is needed to define final results of the listening session and determine how mood was changed. MuPsych studies expect very detailed interviewing which could be annoying for people. This chapter is targeted to investigate approaches to automate questionnaires of MuPsych studies by retrieving the data from sensors and web resources. However the data which is already collected by the psychological research can be used for further collaborative filtering, because it describes different types of person, gives information about their music preferences, and helps determine what influence music has on humans. We can process the data of new users in respect to the data collected from MuPsych respondents. Based on this knowledge base we can generate generic playlists for users and provide them as initial recommendations. In summary, the MuPsych project fully solves the cold start problem of the collaborative filtering of the music context.

4.2.6 Mood and activity related profile creation

We want to provide personalized music recommendations with respect to specific emotion and situation cases, to do that our service has to operate at different customized profiles. In this section I will explain what data should be included in this kind of profile.

The main idea of this feature is binding emotional aspects to particular music attributes, which will allow us to understand what a person wants or even needs to listen to at the current time to change or maintain their emotional state and keep well. Our service is expected to be semantic driven and the data model should be formed as semantic ontology. Figure 27 shows the main elements of the ontology which holds data of services related to assigning music data to personal preferences. With this data the system can perform track filtering and music curation in respect to particular emotional and activity states.

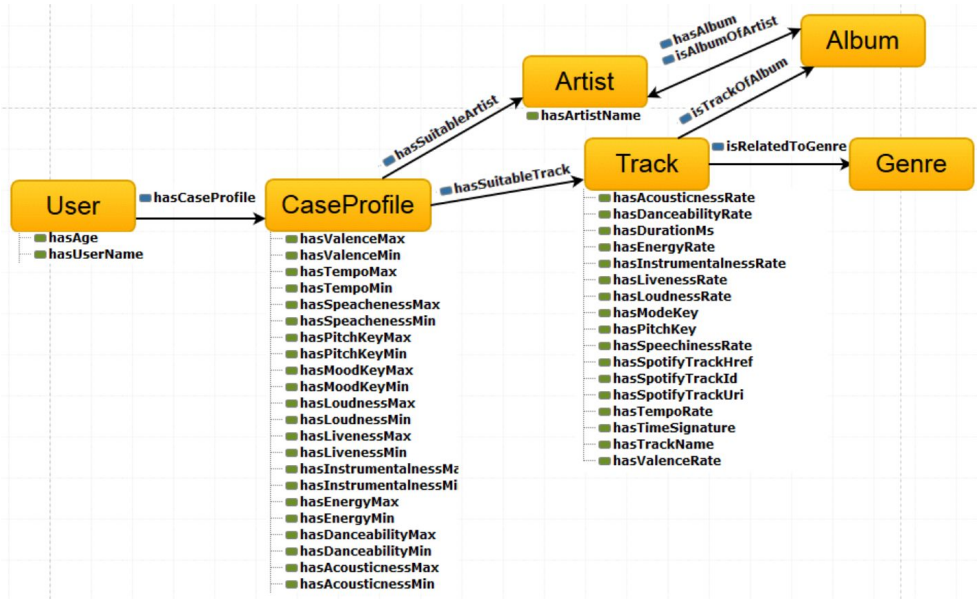


Figure 27. Emotional profile ontology.

5 MUSIC ADVISER SERVICE ARCHITECTURE

In previous chapters we talked about techniques and methodologies related to music curation services, now that we are familiar with all aspects of this issue separately, it is good time to unite them together. In this part of the study I will discuss the whole structure of the music curation engine from the technical point of view. The service design represents separated service entities which are interconnected and work as the united ecosystem to change or maintain emotional state of users with respect to personal music preferences. Figure 28 illustrates the overall picture of the music adviser system. Modular service design allows it to make the system objective consistent by isolating internal logic, external support services and the data space. Each part should have a particular set of responsibilities and work closely with other parts. Flexible and dynamic exploration of data sources has to be established by the system for successful achievement of defined benchmarks.

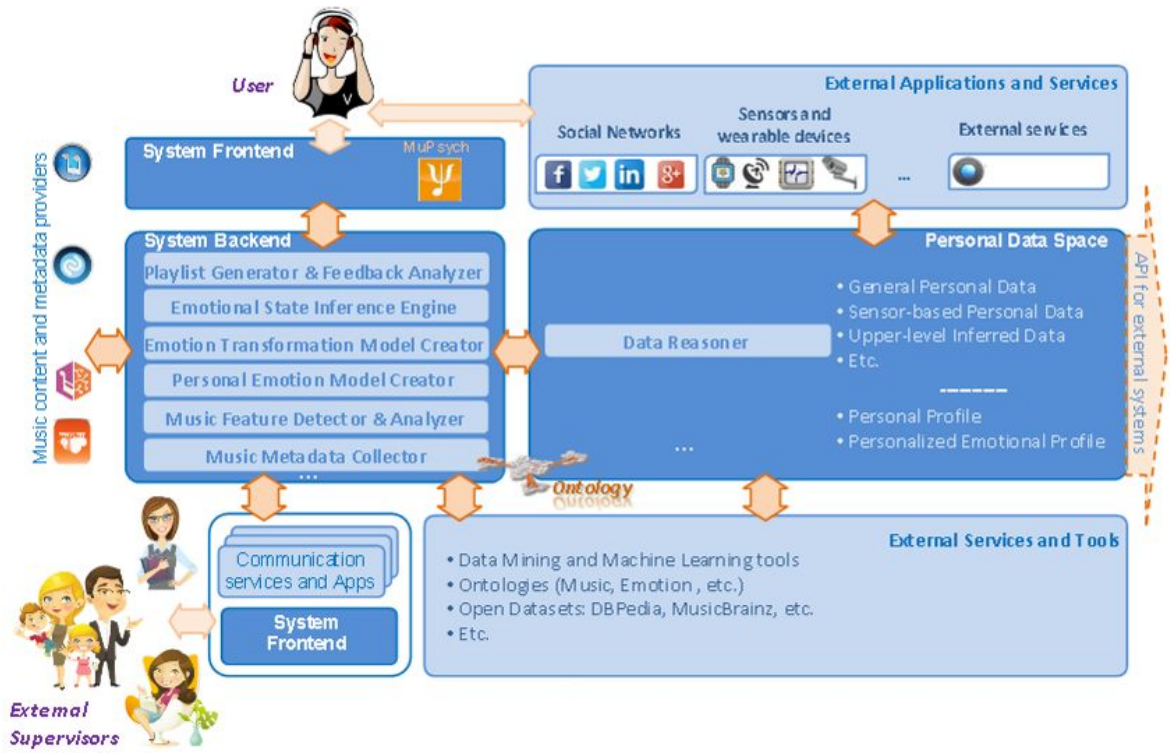


Figure 28. General service architecture.

5.1 Backend

The system kernel is represented as a set of enterprise applications with unique functionalities and responsibilities. The main purpose of the backend is data management and processing. It is interconnected with other parts and manages them to collect the appropriate data from them. At first it generates a main user profile with the general personal data which is received as a result of initial questionnaires, then it creates an emotional model of the person in respect to music preferences and psychological states captured during listening sessions. Based on gathered information from the user and web resources, it then generates music playlists by applying music track filtering described in the fifth chapter. The emotional state transformation module performs comparisons between the data of the profile which corresponds to the current state and the one which is related to the mood which was considered by the user as desired. Then music tracks should be selected in a way that their features smoothly and consistently come from ranges defined from the current profile towards one of the desired mood. To establish a connection with the client side, we need to implement SOAP or RESTful endpoints, or their combination according to the system requirements. RESTful technology is simpler from the development and maintaining issues, however SOAP provides better data structuring and allows us to bind frontend more tightly to the backend. The data management and storing in our case is implemented with Resource Description Framework. The Java platform has a wide range of convenient tools and libraries to utilize SPARQL querying of graph databases with linked data.

5.2 Frontend

The main purpose of the frontend in general is for the direct interaction with users and some external services for data retrieval and service output delivery. It is represented as a mobile application based on the Android platform. This application includes a data collection module which manages various heterogeneous data sources such as sensors and user reflections of their music desires. The main part of the frontend is the music player, integrated with music streaming services. Integration with music services can be done in several ways, some services provide particular development tools also called as service development kits (SDKs),

in some cases we need to implement client applications for the generic APIs based on HTTP requests. We need this kind of integration to gain access to music track pools and establish music streaming on the fly. The more music services that are integrated with our application, the more robust our system will be, because in that case it will cover a wider audience and gain more data. Fundamentals of the Android development are described in the chapter of project-related technologies. From one side, hard utilization of sensors and external data sources at the frontend requires more power and resources, however this approach will make the service more user-friendly and natural looking instead of a strict scientific tool which is full of annoying surveys and notifications. The main idea of the automatization of the emotional states and activities detecting processes is to push machines to learn how to understand users. In the ideal case, the user should just take their phone into their hands and the phone should guess what music to recommend based on gathered information in an automatic manner, instead of manual form filling.

5.3 Data reasoner

The main responsibility of the data reasoner module is the arranging of the data retrieving and processing. This ontology driven engine fully relies on web semantic technologies. The approach of the linked data utilization brings us extra flexibility of the data. With the linked data technologies the system can apply rules of data management dynamically. In general, the reasoning term at this context means data retrieval by following predefined patterns of querying. According to the data model we might have a very nested structure of human and music entities interconnection, and therefore a linked data approach is very efficient technique which is suitable for the the management of large scales of data with a huge number of relationships between different parts. For example, some music tracks, albums and artists can be related to each other somehow, even some of them can have similar metadata, with linked data we can handle it more objectively, while paying attention to different aspects of interest.

To retrieve the data on demand the data reasoner has to be connected with external data sources such as social networks and media sources where the data about the person and their music preferences can be retrieved. For that we need to implement adapters for various external service APIs. I described approaches of binding user profiles of social network

personal accounts to our system in the chapter related to data collection, and more detailed API descriptions can be retrieved from official service documentations. The personal data space logically is the part of the whole data model which is described in detail in the fifth chapter.

A major responsibility of the data space component is the data arrangement which is directed by the backend management. This part includes the skeleton of the data model, also known as ontology. As we already defined, the data model consists of two main parts: music and personal data. There is no necessity to create ontologies for these purposes from scratch and reinvent the wheel, as there are a lot of ready ontologies and data sets corresponding to the personal and music data maintenance.

Music ontology is the tool which helps efficiently to arrange and publish the music data on our own sources. It provides a wide range of services such as API integration of external web sources with the music data model which is defined by the platform. There is a possibility to integrate heterogeneous data sources with this ontology. The data model is very flexible and we can design the ontology in different ways according to the purposes of our services. In other words, we can build our own ontology on top of the music ontology. For example, there is an ontology which is directed mainly on music features (Music features ontology 2010), it concentrates classes from the music ontology which describe music features. Another example represents the extension of the basic ontology with features related to the music engineering (Studio ontology, 2009). Most music features which were described in the section 5.2.2 are presented in the music ontology, however we can extend the ontology on demand. The whole set of classes and annotations of the ontology can be found at the official documentation web resource. (Music ontology, 2016).

Another ontology driven service is directed at retrieving and management of the data sources from different web resources. In other words, this ontology aims to integrate various services by applying the linking data approach of the semantic technologies. It covers many services and data sets such as DBpedia, Echo Nest, MySpace, BBC Playcount data and many others. (DBTune, 2016).

To keep all the data which relates to the psychological phenomena of humans we need to utilize the emotional related ontology. It should reflect as much as possible all subjective personal thoughts and feelings. The correspondence of the emotions to music features is important, however the detailed description of the psychological state is one of the main questions of this study. The set of emotional values which are provided by the data model of the MuPsych research can be held with the emotion ontology. Following the same way as we did with music features, the system can perform annotations of emotional values, such as levels of happiness, anger, surprising and many others. (Emotion Ontology, 2017).

6 PROTOTYPE OF THE MUSIC CURATION SERVICE

Taking into account all descriptions of technologies and parts of the project related to this study, we implemented the design of the music recommendation ecosystem. When all aspects of the system design are completely clear, it is good time to try the practical implementation of the service and show how it would be realised and visualized, because it is better to see something in practice. Descriptions in this thesis section are directed more to the graphical user interface (GUI) and guides of using. I do not provide too much development details at this point because the internet is full of technical documentations, tutorials and other related information which can be easily accessed.

6.1 Authentication

Most web services which expect the data personalization usually start their functionality from authentication. On their first start of the application, the user should provide some basic data about him/herself. The user is expected to fill out a few simple forms and accept agreements of using the service. Figure 29 illustrates a fundamental demographic form which is provided at first application start. These values will be received by the server side and securely stored. Later, the authentication process should be done automatically based on the device id or provided email or phone identificators in combination with password. Current version of the music recommender application uses device identification values, they are constant till the force recovery and data wiping from the device. This method is very convenient, because in this case, users even do not need to re enter login and password when the application data is erased. From another side it brings as limitations which binds user profile to a particular device. This limitation motivates as to roll back our application to the traditional login and password method in the future. However, the data which is illustrated on the figure is the fundamental sample, it is not enough to the personal profile creation for further classification and collaborative filterings. Extra data can be added on the fly, according to the choice of users or on demand of the system. The system requires extra data about users to assign them to particular groups or clusters more precisely, classify them for collaborative filtering.

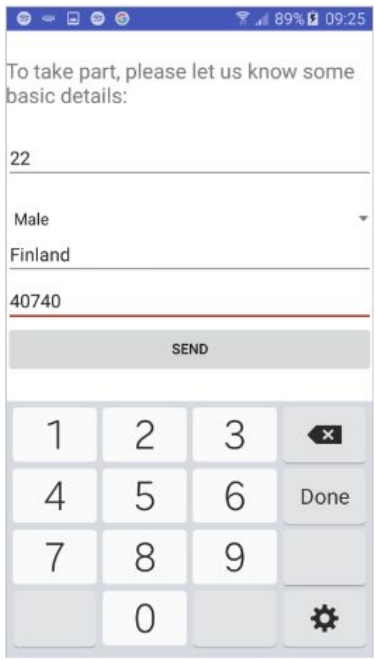


Figure 29. First start personal data forms.

6.2 Detecting of the mood and activity conditions

In this study I explained the approach of the sensor-based and external data retrieving and processing for the automatic detection of emotional states and activities. Practical implementation of this approach requires additional effort and a longer timeframe, which will require further developments, instead of that we can simply use a few small questionnaires and proceed to descriptions of the system kernel.

According to the data model, presented at (MuPsych, 2016) research, the personalized case of the listening session can be implemented with respect to the following attributes:

1. Mood
2. Listening reason
3. Activity

When the user is authenticated the system asks about the mood and activity of the person. Logically, this process requires a questionnaire represented by two single choice lists.

However the range of different mood and activity types could really huge, according to the graphical interface usability, lists should not have too large scale. These lists can be shortened, but uncertainty could increase drastically in this case. Another solution of this problem is adding extra criteria to filter values for these lists. Emotional condition can be represented by valence of the mood and arousal, we can scale these factors by numerical values which allow to generate simpler lists. The same approach can be applied to the list of activities, defined by the location detection of the person. Figure 30 illustrates the mood filtering based on the mood valence and energy levels. For example if the person has low energy and positive mood, the emotional condition could be: tired, love, interest, nostalgia or happiness. Otherwise, if user is in a negative mood and with low arousal, tiredness is an option. Different combinations of these two values determine next emotional status lists. The graphical interface of the application related to the mood detection questionnaire is represented in Figure 31. Psychological science backgrounds of this filtering were provided by the postgraduate researcher at the University of Jyvaskyla Will M. Randall.

Mood	mood	energy	
Anger	low	high	Show IF mood = '-3,-2' AND IF energy = '0,1,2,3'
Anxiety	low		Show IF mood = '-3,-2,-1,0'
Boredom	mid	low	Show IF mood = '-1,0,1' AND IF energy = '-3,-2,-1,0'
Calm	mid	low	Show IF mood = '-1,0,1' AND IF energy = '-3,-2,-1,0'
Delight	high	high	Show IF mood = '2,3' AND IF energy = '0,1,2,3'
Depression	low	low	Show IF mood = '-3,-2' AND IF energy = '-3,-2'
Disgust	low		Show IF mood = '-3,-2'
Excitement	high	high	Show IF mood = '0,1,2,3' AND IF energy = '2,3'
Fear	low	high	Show IF mood = '-3,-2' AND IF energy = '0,1,2,3'
Happiness	high		Show IF mood = '0,1,2,3'
Interest	high		Show IF mood = '0,1,2,3'
Love	high		Show IF mood = '0,1,2,3'
Nostalgia		low	Show IF mood = '-3,-2,-1,0'
Pride	high		Show IF mood = '0,1,2,3'
Sadness	low		Show IF mood = '-3,-2,-1,0'
Shame	low		Show IF mood = '-3,-2,-1,0'
Stress	low	high	Show IF mood = '-3,-2,-1,0' AND IF energy = '2,3'
Surprise		high	Show IF mood = '2,3'
Tiredness		low	Show IF mood = '-3,-2'

Figure 30. Emotional state conditions

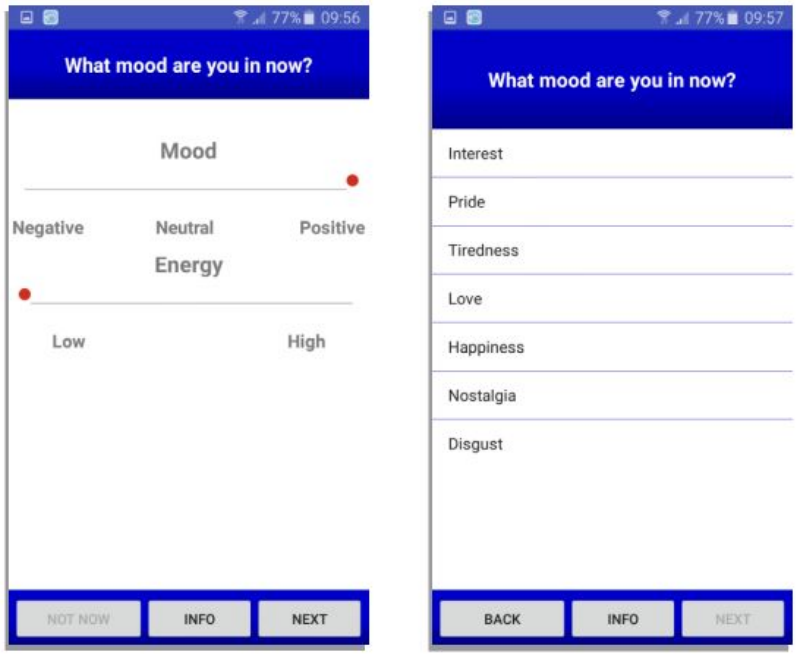


Figure 31. GUI of the emotion related form

A similar logic can be applied to the activity lists. There are sets of activities which correspond to specific places. The number of place types is smaller than the number of activities and can be asked at the first step. Then, a refined list of activities can be presented. If the user chooses location “travelling”, suitable activities such as passenger of some kind of transport, cycling, jogging will be shown at the activity list. Further listening session will be considered for the particular case, and feedback of user will have effect to a specific emotional profile.

Figure 32 shows places as columns and corresponding activities as rows. If mark “1” is placed at crossing point of two values it means that they are related to each other. This matrix allows us to increase the efficiency and usability of surveys, in that case people would easier find descriptions which match their life use cases, there are more chances to get relevant answers. In further developments this part should be automated following methodologies of data retrieving and processing described in the fourth chapter of this study.

	Home	Someone else's home	Work	Travelling	School / University	Shops / market	Restaurant / cafe	Other (inside)	Other (outside)
Activity	18	18	13	6	13	7	8	14	17
Other	1	1	1	1	1	1	1	1	1
Eating	1	1	1		1	1	1	1	1
Gaming / entertainment	1	1	1		1	1	1	1	1
Nothing / relaxing	1	1	1		1	1	1	1	1
Socialising (casual)	1	1	1		1	1	1	1	1
Web browsing	1	1	1		1	1	1	1	1
Reading	1	1	1		1		1	1	1
Working (mental) / studying	1	1	1		1		1	1	1
Shopping						1		1	1
Dancing	1	1	1		1			1	1
Focused music listening	1	1	1		1			1	1
Working (physical)	1	1	1		1			1	1
Exercising / sports	1	1			1			1	1
Socialising (party)	1	1	1					1	1
Cycling				1					1
Jogging / running				1					1
Walking				1					1
In a meeting			1		1				
On bus / train / plane				1					
Passenger in car / taxi				1					
Cooking	1	1							
Getting intimate	1	1							
Going to sleep	1	1							
Housework	1	1							
Waking up	1	1							

Figure 32. Place based activity filtering

The graphical interface is represented in Figure 33. As we can see these lists are single option, which means they do not require a 'next' button, as when the user presses on an option, the app automatically proceeds to the next screen. Following this approach we use the positive feature of the bottle neck case, coming from general and abstract definitions towards more precise values and statements. However, this part is still manual in current version and the

application still looks like scientific psychological tool instead of natural looking interactive player. Further implementations will eliminate these annoying questionnaires.

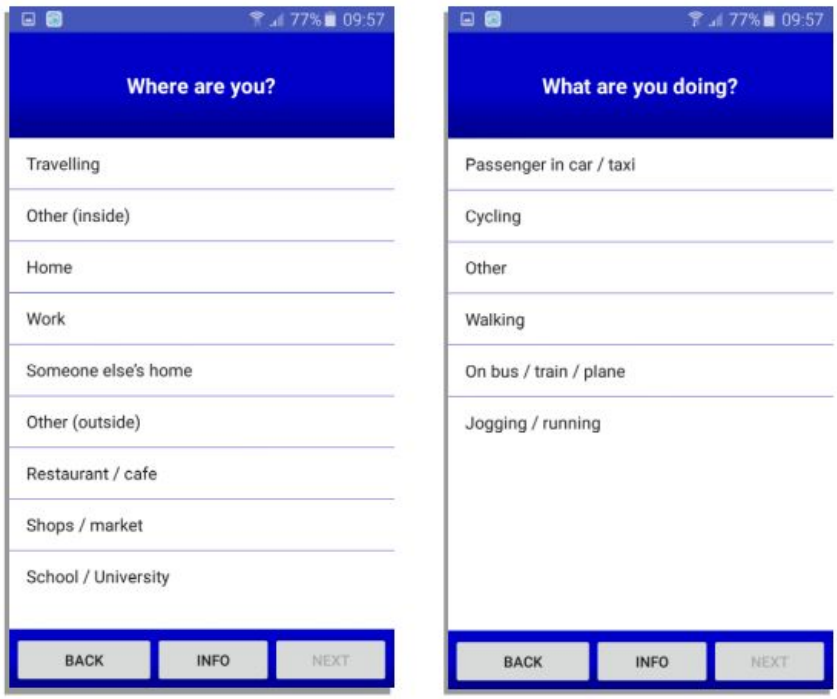


Figure 33. GUI of the activity detection feature.

If the user already has some experience with this player and the system already generated personalized playlists for him, the history of emotional cases should be shown at the initial state. In this case the user can choose existing case from history and proceed directly to the generated playlist avoiding questionnaires, or create new case if the current situation does not match any recent specifications. Figure 34 shows the screen with history of previous emotional/activity cases. However it does not mean that personalised playlists are updated from session to session, they are updated continuously and on the fly, when required amount of feedbacks comes to the server to make objective calculations. The main criteria of the feedback is like or dislike given for the song by users. A minimum number of likes for songs is set to five, minimum amount of dislikes is three. These thresholds initially were set for testing purposes, later this feature could be tuned according to ongoing requirements.

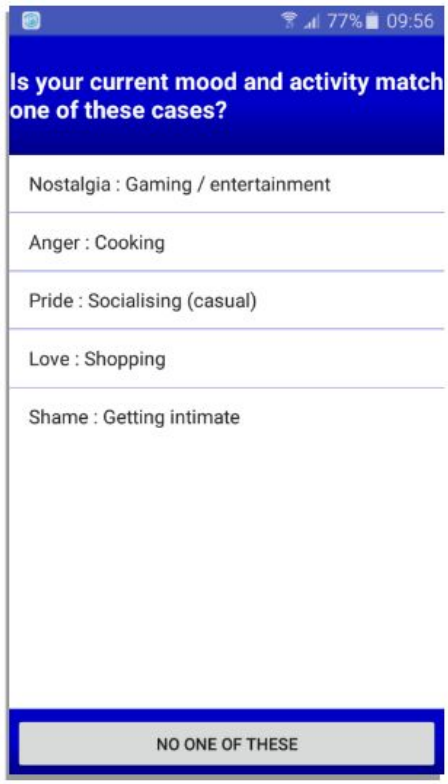


Figure 34. Recent mood/activity cases

6.3 Listening session

Here I will discuss the main part of the service when using it for music exploring and listening. There are two options to explore new music tracks: use the basic keyword search or start with offered playlists. I want to note that even if the user creates a totally new emotion related profile, the system should provide some recommendations immediately. At this stage we have only activity related playlists, which were retrieved from music services and generated according to the data collected by MuPsych research. However at this initial period preferences which are calculated for MuPsych participants are common for a particular cluster of users. So we have to deal with activity corresponding playlists and generated with collaborative filtering approach. Figure 35 shows screens with the search form and generated playlists. For example, if the user selects an activity case related to eating or cooking, the system will provide lists with music which was selected for people with similar characteristics at similar mood and activity case.

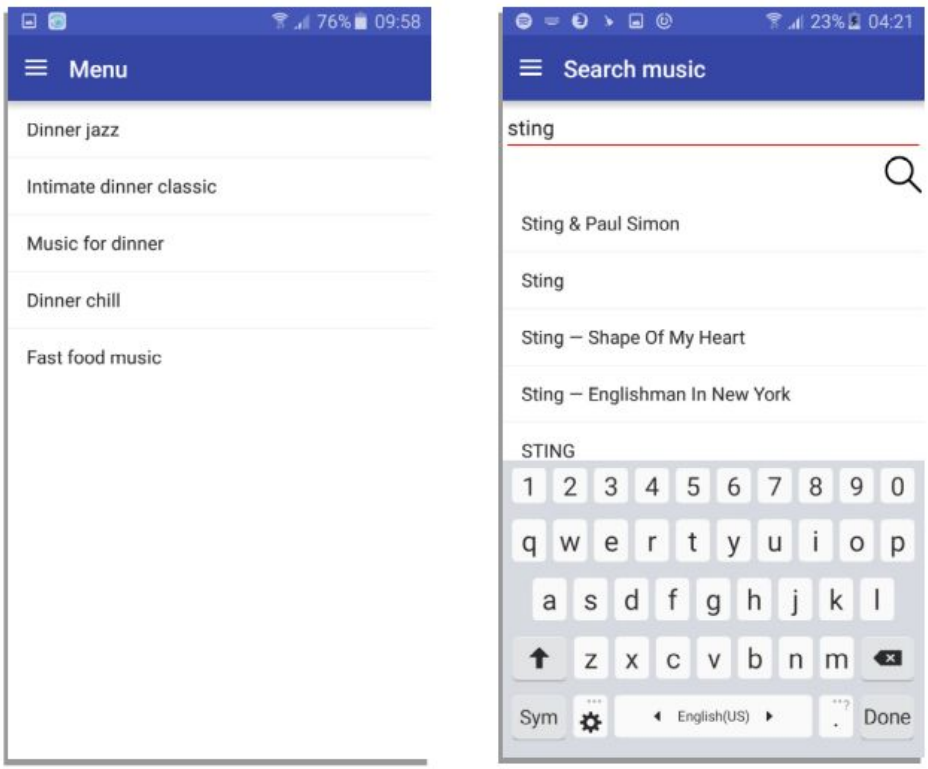


Figure 35. Music exploring screens

If a user presses one of these elements, listening will start immediately and the current listening screen should appear. During a listening session, the application generates the feedback about how tracks are listened. In case of skipping the track, the player asks about the reason for skipping, and the reason is simplified to the “like/dislike” form. Later, music features of full listened and liked tracks will allow for the tuning of tracks selected to the specific person and case. Disliked tracks will have the opposite effect for further selection process, tracks which were listened to recently should be removed from recommendation lists. The interface of the music player contains control buttons, which allow switching between tracks, play, pause or stop listening and setup the repeating of the track or shuffle playing. In addition to these, the picture of the related album, name of the track and related artist are presented. Figure 36 illustrates the main player screen and the notification in case of skipping of a track. After the interaction between user and server, the system performs filtering and provides the playlist which is suitable for the defined mood and activity, and this playlist is updated continuously according to the upcoming user interaction with the player.

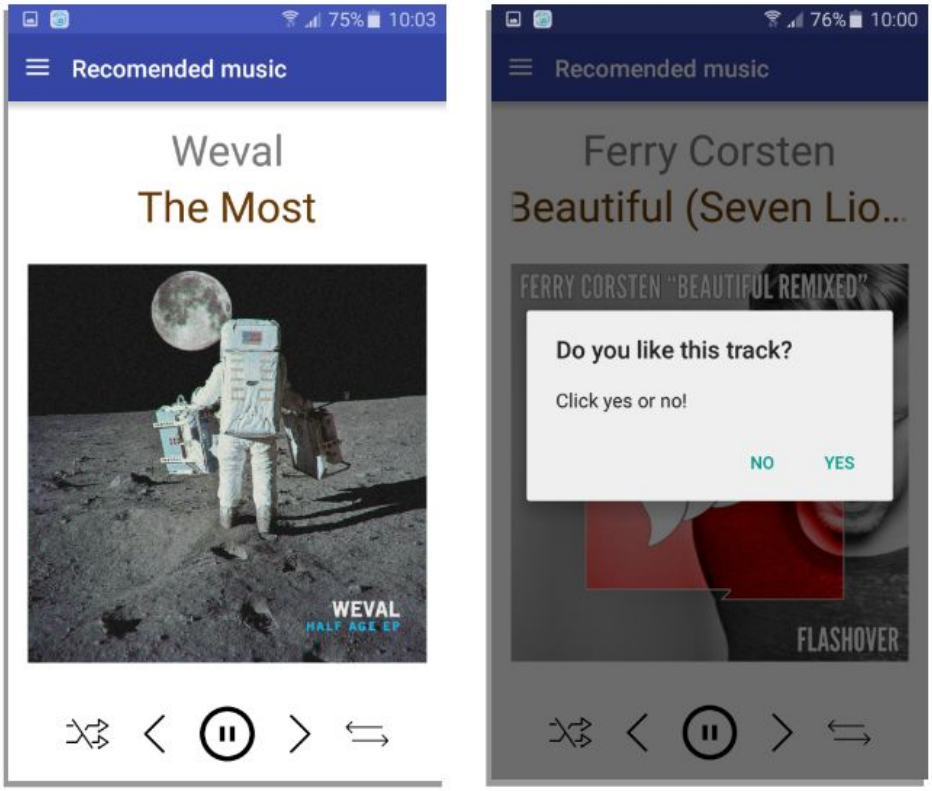


Figure 36. Music player screens

If the user needs to change emotional state, the system will filter music tracks with features similar to music preferences calculated for the emotional based profile which corresponds to the desired emotional state. Collaborative filtering is partly used here, for example, if music with particular features affect the specific person type, other users from the same cluster might get a similar effect. Mood transition has to be performed smoothly, the tuning of music feature ranges has to be done very smoothly to avoid jumps, otherwise affect of the music curation could be insufficient. Tuning of music feature ranges could be done in respect to recommendations provided by the third part such as personal doctor or psychologist. All generic tracks are replaced by a personalized list when the system performs filtering of music tracks. In Figure 37 you can see the playlist which was recently generated and provided for the mobile application, by tapping on items from this list we can start to enjoy of listening them. More psychological investigations of issues related to this topic are expected to enforce the system with more precise data and apply stronger data processing algorithms.

Figure 37. Personalized playlist

6.4 Web service Endpoints

The service publishes SOAP WSDL endpoint for the mobile client side, all of the data exchange such as feedback sending or playlist transferring is performed with this service interface. However, internally the service manages all data in RDF format, some of this data can be published as RDF/XML resource which is understandable by machines. Only the data which is not confidential and was generated by the system internally can be published. The data retrieved from external services is not considered for publishing or additional agreements should be made.

I would like to present the service feature which receives the name of the artist as input and selects other artists which are similar to that artist with respect of their metadata. Finally it

publishes all of them in RDF/XML format. A very similar feature is provided by the Spotify music service, which provides the API for this feature, including a version of it with GUI. In some ways, this could be considered as reinventing the wheel. Firstly, I want to check and prove that the approach of artist similarities based on intersections of their metadata works. Secondly, I want to test how the system can search and populate the music data without human management. Just imagine, we have the system which receives only artist names, if they are new for the system it searches all possible metadata about them from various web sources, including photos, albums, their songs and other information. Then it searches similar artists and performs similar data searches for them. Finally it populates all collected data which is understandable for other machines. With web semantic technologies we get higher probability of the collected data relevance. This kind of self management in data retrieving and publishing approach can also be applied in other industry and art areas. Furthermore, it is possible to make an engine which will enforce this data in a human readable format, so XHTML in combination with RDF data would be understandable for both humans and machines. As a result we can develop self-managed and self-updated music database similar to Wikipedia. And one step could be made towards the realisation of futuristic dreams about robotic composers and artists. Figure 38 illustrates methods of the SOAP service. They are used to register a user in the system, authenticate it, retrieve music, process feedbacks and explore similar artists.

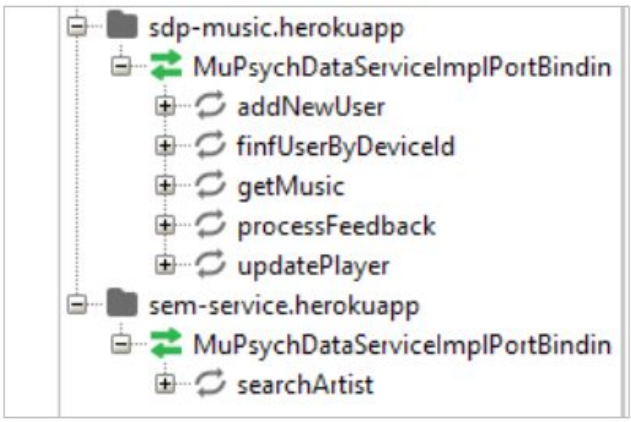


Figure 38. Methods of the SOAP service

According to the SOAP standards, requests should be wrapped in envelopes and have predefined structure of parameters which are passed for methods of the web service. Figure 38 shows the main actions which can be managed on the server side by client devices. As shown, we can manage the personal account data and control the music filtering and playlist creating processes.

Figure 39 shows the service where we need to pass the name of the artists as a request input parameter. Then our semantic service performs calculations and returns the response with names of artists and their metadata. With this type of service we can transfer different data types, such as numerical values, strings and others. In this case the service returns the list of strings, it is convenient because the number of explored artists could be different and the list keeps them in a well structured stack. Given figures which are related to SOAP services represent screenshots of the SOAPUI testing tool, which allows us to efficiently test web services without writing scripts of the client part for services. We need to create a SOAP specific request envelope and pass required parameters, in this case we need just to define the name of the artist.

At the same time the system populates explored music, albums and artists metadata as linked data annotated sources. The current version updates the RDF source on the File Transfer Protocol (FTP) server. Figure 40 illustrates sample data from the RDF web source which was published by the music recommendation system. These data can be accessed and processed by other services which utilize web semantic data processing approaches. So far we have self managed and updated linked data source fully supported by recommendation service. Now it is understandable for machines, if it would be visualized with user friendly view it could be great music pedia source for humans. Of course, there are existing similar sources such as DBTune[9], DBPedia[10] and others and we are not going to reinvent the wheel, the purpose of this part of the recommendation system is to offer own approach of the data collection and processing for establishing learning process of machines and automated resource exploring. Also I would like to notice that copyrights should be paid attention in this feature, sources

[9] DBTune - non commercial web source. Publishes music RDF data since 2007.
[10] DBPedia - global storage of the semantic and linked data.

which are explored by the system for further data processing and publishing should be checked properly and before the data collection to prevent any copyright violations.

Request :

```
:soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/enve
   <soapenv:Header/>
   <soapenv:Body>
      <sem:searchArtist>
         <!--Optional:-->
         <artist_name>Sting</artist_name>
      </sem:searchArtist>
   </soapenv:Body>
:/soapenv:Envelope>
```

Response :

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:searchArtistResponse xmlns:ns2="http://SemanticService
         <return> Sting#https://i.scdn.co/image/ ...
         <return> The Police#https://i.scdn.co/image/ ...
         <return> Peter Gabriel#https://i.scdn.co/image/ ...
         <return> Annie Lennox#https://i.scdn.co/image/ ...
         <return> Simply Red#https://i.scdn.co/image/ ...
         <return> Seal#https://i.scdn.co/image/ ...
         <return> Genesis#https://i.scdn.co/image/
         <return> Mark Knopfler#https://i.scdn.co/image/ ...
         <return> Joe Cocker#https://i.scdn.co/image/ ...
         <return> The Alan Parsons Project#https://i.scdn.co/
         <return> David Gilmour#https://i.scdn.co/image/ ...
         <return> Simple Minds#https://i.scdn.co/image/ ...
         <return> Level 42#https://i.scdn.co/image/ ...
         <return> Bryan Ferry#https://i.scdn.co/image/ ...
         <return> Bruce Hornsby#https://i.scdn.co/image/ ...
         <return> Pat Metheny#https://i.scdn.co/image/ ..
         <return> Supertramp#https://i.scdn.co/image/ ...
         <return> Toto#https://i.scdn.co/image/ ...
         <return> Marillion#https://i.scdn.co/image/ ...
         <return> Don Henley#https://i.scdn.co/image/ ...
         <return> Chris Rea#https://i.scdn.co/image/...
      </ns2:searchArtistResponse>
   </S:Body>
</S:Envelope>
```

Figure 39. Input and output of the SOAP service

```
<rdf:RDF>
 - <rdf:Description rdf:about="http://users.jyu.fi/~mirumian/music.rdf#Eric Clapton">
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Paul McCartney"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Bryan Adams"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Joe Cocker"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#David Bowie"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#U2"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Rod Stewart"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Elton John"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#The Police"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Billy Joel"/>
      <rdf:type rdf:resource="http://users.jyu.fi/~mirumian/mo.owl#Artist"/>
      <ont:hasIdentification> ^^xsd:string</ont:hasIdentification>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Chris Isaak"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Mick Jagger"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Phil Collins"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Chris de Burgh"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Bruce Springsteen"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Simply Red"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Annie Lennox"/>
      <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#NamedIndividual"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Peter Gabriel"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Freddie Mercury"/>
    - <ont:hasImage>
        https://i.scdn.co/image/ ^^xsd:string
      </ont:hasImage>
      <ont:hasPopularity>73^^xsd:string</ont:hasPopularity>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Sting"/>
      <ont:hasRelatedArtist rdf:resource="http://users.jyu.fi/~mirumian/music.rdf#Robert Plant"/>
   </rdf:Description>
 - <rdf:Description rdf:about="http://users.jyu.fi/~mirumian/music.rdf#Atlantic City">
      <rdf:type rdf:resource="http://users.jyu.fi/~mirumian/mo.owl#Track"/>
      <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#NamedIndividual"/>
      <ont:hasIdentification> ^^xsd:string</ont:hasIdentification>
      <ont:hasAlbum>Nebraska</ont:hasAlbum>
   </rdf:Description>
```

Figure 40. Published RDF music metadata

Publishing of these linked data over the web allows as to extend our system with other independent services, this approach allows to establish interconnection between isolated services and provide the data for external parts which can use it in other domains

6.5 System performance evaluation

Always, results play the most significant role of any work. This section is focused on the evaluation of the recommendation system prototype which was implemented throughout of this thesis. Performance analyzing is the logical conclusion of the prototype implementation part of this study. I examine the music recommendation system on the group of volunteers, they answer extended version of music reflection feedback consisting of three parts, provided at the beginning of the music listening session, after five and after ten minutes. Initial questionnaire is targeted to determine initial activity and emotional states and define how user wants to change or maintain the current emotional condition. Second and third surveys are focused to detect changes of personal emotions and get satisfaction rate from the listening process.

All evaluations are performed in automated manner, the application sends push notifications of surveys when users start to listen music. Therefore, these use case results give as the real evaluations and special pretendings. Main subject of the examination is based on likes and dislikes, user is prompted to answer such kind of question in case of the track skipping, if the whole track is listened it can be considered as marked with like. Another thing which should be paid attention is how music effect to emotional conditions matches desired music listening purposes of users and how the system fulfills their estimations.

Major methodologies of recommendations are: collaborative and classification filtering. Collaborative filtering relies on the MuPsych project which has over one hundred participants at the current state. Music classification filtering method is based on the music features metadata retrieved from DBTune and Spotify services. Mo copyrights are violated, all the data was used for non commercial scientific purposes.

Sample 1:

Gender: female

Age: 20

Location: Finland

Time: 11:44 - 11:57 pm

Initial emotional state of the person was considered as aggressive and tired in respect of high arousal and lower mood rates.User chose the activity related to the mental work. Person was expected that music would help to unwind, relax and concentrate on work. The situation is clear and motivations are logical, if person is aggressive it would be hard to focus on something. According to captured tracks which were recommended by the system we can notice that there were selected mostly instrumental music with no vocal, middle tempo and with clear marked rithm. Second survey showed a bit lower aggression, however tiredness was the same. The third part of the feedback presented the emotional state described as nostalgy, it means that aggression were gone, however, tiredness were still presented. Within fifteen minutes user listened eleven tracks, six of them were skipped and four were marked as disliked. According to these values we can say that approximately the system made 64% of success recommendations. The most skips and dislikes were made closer to the beginning of listening session.

Sample 2:

Gender: male

Age: 23

Location: Finland

Time: 9:37 - 9:49 am

Initial emotional state of the user was described as boring. At that time person walked. The desired effect from the music was to increase the arousal and change the mood to happiness. The most of tracks had middle tempo, high vocal and liveness of the music in general, with few exceptions, because tracks were selected from the set of suitable music for users from the same cluster and the system accepts some divergence between music features templates in

personal emotional profile and selected tracks caused by collaborative filtering.. The desired result was partly achieved only at the third part of the feedback, after ten minutes of listening user evaluate his emotional condition as pride, not happiness at all, but at least not boring. There were listened seven music tracks and only two of them were skipped and marked as disliked. In this case the recommender had 71% success. It is a bit higher than in sample 1, dislikes were made in the middle and at the end of the listening session. During next few days at the measurement period, user listened music to reduce boring emotional states and mostly in public transport. The success of further recommendations had not changed rapidly, but the average value finally was 61% .

Sample 3:

Gender: male

Age: 23

Location: Finland

Time: 10:01 - 10:07 am (not completed feedback)

The user was excited at the beginning of the listening session. The activity was: reading. The person wanted to maintain that emotional state. Recommender offered tracks which showed good satisfaction rates and were considered as suitable for reading according to user's feedbacks.Then the system tuned them in respect of the music features ranges defined in the personal profile, because the user already had experience using this application for similar activities. Only two tracks from ten were disliked, it means that 80% of recommendations were successful. Changes of the emotional state of the person were not detected because the final questionnaire was ignored, the second part of the feedback had not showed any changes in activity and in emotional conditions. This person used the recommender mostly to have music background for reading. Satisfaction rates of further listening sessions fluctuated around 70%.

Sample 4:

Gender: female

Age: 25

Location: France

Time: 6:54 - 7:07 pm

The initial emotional state was described as high stressed and the person wanted to eliminate worries. The activity was studying. During the listening session user was offered with tracks which showed good results for stress relieving during MuPsych research. Partly there we tracts which matched preferences of users in similar situations from the same cluster. Second questionnaire did not showed any changes in emotional state, third one reflected small pushing from stress to neutral emotional state. Unfortunately, the result of the recommendation in this case was not sufficient, only 35% of recommended tracks were marked as suitable, and there were too much skipped cases.

These samples showed that recommendation methods which were implemented in this study represent an effective way to manage emotional statements of people. However, from these results we can also see that some parts of the system are inferior and need further improvements. Taking into account satisfaction rates and changes of emotion states we also should pay attention to other factors which have influences to preferences, mood, behaviours and decisions. In other words, we can not argue that emotion pushing was caused only by music. In general, the structure of the music recommendation ecosystem is implemented well and has a lot of implications for further developments. It is clear that this limited number of use case samples can not be considered for the comprehensive evaluation of the system. We need more tests with real users, however, due to lack of opportunity to study more cases, I present just this limited set.

7 RELATED WORK

The topic of music recommendations is not totally innovative, however it is still very fresh and we deal with cutting edge technologies when design our recommendation ecosystem. In this chapter I want to review music filtering approaches which are already in industrial use and compare them with results of my efforts.

Recommendation technologies are used nowadays in many business branches, the music digital market is no exception and there are many existing music streaming services which perform music recommendations and automated playlist generating. According to non-disclosure rules companies do not publish information about their business and technology approaches, we can just take a look them as ready made production to define what nature they have based on inputs and outputs with which they operate.

Almost all music streaming services include a feature of simple recommendation also known as radio adviser. The working principle of this approach is based on capturing artists listened by the user, then the system retrieves tracks of these artists, mixes them and provide plays in turn, usually updates of playlists occur periodically.

Collaborative and content based filtering is widely used by big music streaming services such as Last FM, Pandora, Spotify, Soundcloud, Google music and many others. Approaches of music recommendation systems depend directly on business strategies of these individual music service providers.

Pandora streaming service performs recommendations based on the data provided by the Music Genome Project, where music track detailed annotations are presented, which contain about 400 variables per each fragment, creation and support of the system's knowledge base takes significant effort from music professionals. Logically, with access to such significant amounts of the data and operating with very detailed music metadata structures, the system has to apply very strong algorithms to the data processing. This suggest that Pandora is mostly algorithm targeted system.

Spotify music service has adviser feature called “Discover weekly”, which based on detection of the music data listened by users, applying collaborative filtering methods. As result of listed efforts Spotify provides personalised playlist updated every week. Echonest music platform includes huge resources of the metadata about music fragments including values of their sound attributes. Some of these music services were migrated from Echonest to Spotify application programming interfaces. According to this, it seems that Spotify hardly utilises content based filtering approach in its adviser system. Johnson in his research (2014) describes how logistic matrix factorisation can be used for processing implicit feedbacks at filtering context such as web services service such as Spotify.

I do not have full access to music streaming technologies and for deeper service specifications, however according to the functionality all services which I used do not perform emotional-based music recommendations, because most of them do not have any related surveys. We can accept that they retrieve all of the data automatically, but in that case they should spend more hardware resources, it can be very noticeable especially on old powerless devices such I have, so I did not encounter these kind of services. The main indicator that the emotional personalization of music is not set up in industry is the fact that recommendations are same from mood to mood. My approach is directed to maximize the recommendation personalization.

We can find research related to this topic and it means that the idea is relevant. For instance, Jacobson et al. (2009) describes the approach of the open world ecosystem for music similarity detecting. For example, Ghatak (2009) designed the music recommendation system based on emotions and patented, in the patent application document we can get the general overview of such kind of systems. Zhang and Chong in their patent application (2014) described the music recommendation system which relies on the biometric data retrieved from sensors of mobile devices. However the most of patent applications are written briefly and very abstractive. At the same time scientists keep work on this topic. Logically, patent applications are done for innovative things which are not exist, for ideas which are not implemented and for stuff which is not patented yet. It means that the topic of emotion driven

recommendation system is still fresh and has a lot of unanswered questions, that facts increase the relevance of this study.

Music psychology department of the university of Jyväskylä makes researches related to the investigation of the music influence to the emotional conditions of humans. In this year they are going to start project related to the design and development of the music curation system to support young people in difficult life and psychological situations. It is named My Music - My Life. I hope that the research will start successfully and I will participate in it as researcher and service developer. Parts of this study can be used as preliminary developments for that project and further research.

8 CONCLUSION

This study has shown that the wide range of musical tracks provided by music streaming services creates the problem of choice, which requires new approaches that take into account music influence on human emotional states. The main goal of the present research was to design an automated system of emotion-driven music management. The fundamental purpose of the system was to change or maintain the emotional state of the user and match personal music preferences by exploring music tracks with specific attributes. The theoretical background of this thesis familiarized readers with common recommendation methodologies, which were explained in the second chapter, and related technologies, which were examined in the third chapter. The principal problem of the music recommendation system is the data collection and processing, which was described in the fourth chapter. The design of the whole music recommendation ecosystem was presented in the fifth chapter. In chapter 6, the implementation of the working music adviser prototype was presented. This study suggested the solution of the automated emotion-driven music management problem. Findings presented in this study were confirmed by the practical implementation.

The major limitation of this study is closely tied to the complexity of the related system. Due to this complexity, it is impossible to cover all details of the data collection and processing of the recommendation system within the framework of a master thesis. There were not enough experiments involving the retrieval of sensor-based data, which caused limitations in implementing a prototype, as it relied on manual questionnaire forms of emotional and activity states instead of automatic detection. Furthermore, the current user interface looks like a scientific tool, while it should look more naturally and user-friendly. Functionalities of the backend and frontend prototype applications were targeted to confirm the possibility of enforcing smart machines with common recommendation methodologies within the music context, and only the kernel of the thesis topic was implemented. The design of the mobile part of the prototype is limited, and currently has a very basic style, while the usability is not near the quality it should be at for production. At the current state the music adviser system is

integrated with the popular Spotify music service, however to gain a larger audience other music streaming services should be integrated with the system.

This research provided a broad description of a solution for the music recommendation problem, and highlighted a range of implications for further investigations. It makes several noteworthy contributions to move machines to a new, smarter operational level. Taking into account successful results of music recommendations performed by the working prototype of the emotion driven recommendation system, we can confirm that music can be a strong link between human emotions and electronic systems. Further investigation into the connections between music and emotions will allow us to improve the current recommendation algorithms and invent and develop new approaches. Since this study goes hand in hand with the idea of the ongoing My Music - My Life research I believe that it will meet a lot of future implications in science and subsequently in industry. Nowadays the growth of IoT technologies is gaining the momentum, in the future we expect great number of devices producing data such as sensors and measurement devices. Flexibility of the system based on semantic web paradigms will allow to handle extensions with new devices without much effort. Integration of IoT to the recommendation ecosystem will bring extra opportunities in automatization and recommendation improvements. It is very forward-looking and efficient approach of applying Neural Networks to the data processing and building of emotional profiles related to personalized music preferences relying on the data coming from heterogeneous resources such as sensors. However, it requires huge training data sets, which can not be collected due to the limitations of the study represented as lack of the system participants and limited periods of the research. Neural Network applying is considered for future improvement of the recommendation system when more opportunities of the data collection will occur and more people will participate in the music recommender system.

REFERENCES

Books and publications:

Burke, Robin (2002): Hybrid Recommender Systems: Survey and Experiments. Massachusetts. Kluwer Academic Publishers. Hingham.

Chanda, Mona Lisa; Levitin, Daniel (2013). The neurochemistry of music. Department of Psychology, McGill University. Montreal, Quebec.

Chong, Chia-Chin ; Zhang, Jianyu (2014): Music Recommendation Based on Biometric and Motion Sensors on Mobile Device. Google publication of the patent application. Available at <https://www.google.com/patents/US20140277649>, accessed at 18.1.2017.

Cinnober Financial Technology AB (2012): The benefits of using Java as a high-performance language for mission critical financial applications. A Cinnober white paper. Stockholm. Available at <http://www.cinnober.com/sites/default/files/news/The-benefits-of-Java-white-paper-1.pdf>, accessed at 7.11.2016.

Ekstrand, Michael; Riedl, John; Konstan, Joseph (2011): Collaborative filtering recommender systems. Boston. Now publishers. Available at < http://files.grouplens.org/papers/FnT%20CF%20Recsys%20Survey.pdf >, accessed 12 October 2016.

Fielding, Roy; Gettis, Jim; Mogul, Jeffrey; Nielsen, Henrik; Masinter, Larry; Leach, Paul; Berners-Lee, Tim (2004): Hypertext Transfer Protocol, HTTP/1.1. Network Working Group. Available at <https://www.w3.org/Protocols/rfc2616/rfc2616.html>, accessed at 5.11.2016.

Ghatak, Kausik (2009): Mood based music recommendation method and system. Google publication of the patent application. Available at <https://www.google.com/patents/US20090182736>, accessed at 18.1.2017.

Gourley, David; Totty, Brian; Sayer, Marjorie; Aggarwal , Anshu, Reddy, Sailu (2002): HTTP: The Definitive Guide. Sebastopol, O'Reilly & Associates. Available at <http://www.staroceans.org/e-book/O'Reilly%20-%20HTTP%20-%20The%20Definitive%20Guide.pdf>, accessed at 5.11.2016.

Isinkaye, F.O.; Folajimi, Y.O.; Ojokoh, B.A. (2015): Recommendation systems: Principles, methods and evaluation. Cairo University Egyptian Informatics Journal. Available at <http://www.sciencedirect.com/science/article/pii/S1110866515000341>, accessed at 20 October 2016.

Jacobson, Kurt; Raimond, Yuves; Sandler, Mark (2009): An ecosystem for transparent music similarity in an open world. 10th International Society for Music Information Retrieval Conference. Kobe.

Jonson, Christofer (2014): Logistic Matrix Factorization for Implicit Feedback Data. New York. Spotify documentation. Available at <https://web.stanford.edu/~rezab/nips2014workshop/submits/logmat.pdf>, Accessed at 20 October 2016.

Khriyenko, Oleksiy (2008): Adaptive Semantic Web based Environment for Web Resources. Jyväskylä University Printing House.

Khriyenko, Oleksiy; Nagy, Michal (2011): Semantic Web-driven Agent-based Ecosystem for Linked Data and Services. Industrial Ontologies Group. Jyvaskyla, Finland.

Khriyenko, Oleksiy; Terziyan, Vagan; Kaikova, Olena; Helfenstein, Sacha (2014): Emotional Business Intelligence. 7th International Conference on Human System Interaction. Lisbon.

Kim, Jong-Hun; Un-Gu, Kang; Lee, Jung-Hyun (2006): Content based filtering for music recommendation based on ubiquitous computing. IFIP International Federation for Infonnation Processing (463-472).

Mohammadreza, Sadeghi; Hossein, Namdar; Shahram, Vahedi; Naser,Aslanabadi; Davoud Ezzati; Babak, Sadeghi (2013): Effects of Emotional Stimuli on Cardiovascular Responses in Patients with Essential Hypertension Based on Brain/Behavioral Systems. Journal of Cardiovascular and Thoracic research (167-171). Available at <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3883541/>, accessed at 1.11.2016.

Pazzani, Michael; Billsus, Daniel (2007): The Adaptive Web. Chapter: Content-Based Recommendation Systems. Berlin. Springer Berlin Heidelberg.

Richardson, Leonard; Ruby, Sam (2007): RESTful Web Services. Sebastopol. O'Rely Media. Available at <https://www.crummy.com/writing/RESTful-Web-Services/RESTful_Web_Services.pdf>, accessed at 6.11.2016.

Sahebi, Shaghayegh; Cohen , William (2011): Community-Based Recommendations: a Solution to the Cold Start Problem. Conference or Workshop Item (Paper). University of Pittsburgh. Available at <http://www.dcs.warwick.ac.uk/~ssanand/RSWeb11/7Sahebi.pdf>, accessed 17 October 2016.

Saric, Amar; Hadzikadic, Mirsad; Wilson, David (2009): Alternative formula for rating prediction using collaborative filtering. Berlin. Springer Berlin Heidelberg.

Schafer, Ben; Frankowski, Dan; Herlocker, Jon; Sen, Shilad (2007): The Adaptive Web. Springer Berlin Heidelberg.

Shih, Ya-Yueh; Liu, Duen-Ren (2005): Hybrid recommendation approaches: collaborative filtering. Hawaii. 38th Hawaii International Conference on System Sciences. Available at

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.789&rep=rep1&type=pdf> accessed at 19 October 2016.

Soares, Fabio; Souza, Alan (2016): Neural Network Programming with Java. Packt publishing Ltd. Birmingham, UK.

Terziyan, Vagan; Khriyenko, Oleksiy; Nikitin, Sergiy; Katasonov, Artem (2008): Smart Semantic Middleware for the Internet of Things. Agora Center. Jyvaskyla, Finland.

Ungar, Lyle; Foster, Dean (1998): Clustering methods for collaborative filtering. Technical report. CIS dept. and dept. of Statistics University of Pennsylvania Philadelphia. Available at <http://www.aaai.org/Papers/Workshops/1998/WS-98-08/WS98-08-029.pdf>, accessed 11 October 2016.

Will M. Randall (2016): MuPsych research. Study description documentation. Jyväsjylä. Available at <http://www.mupsych.com/docs/MuPsychResearchinfo.pdf>, accessed at 7.1.2016.

Zhao, Xiaoxue (2016): Cold-Start Collaborative Filtering. Doctor of Philosophy dissertation. University College London. Available at <http://discovery.ucl.ac.uk/1474118/1/Thesis_final_revision.pdf>, Accessed at 17 October 2016.

Web pages:

Android developers (2016). Available at <https://developer.android.com/index.html>, accessed at 17.12.2016.

Cloud Natural Language API (2016). Available at <https://cloud.google.com/natural-language/>, accessed at 01.01.2017.

Cloud Speech API (2016). Available at <https://cloud.google.com/speech/>. accessed at 15.02.2017.

DBPedia (2016). Available at <wiki.dbpedia.org/>, accessed at 29.12.2016.

DBTune (2016). Queen Mary, University of London. Avaliable at <http://dbtune.org/>, accessed at 15.1.2017.

Emotion analyses API (2016). The android arsenal. Available at <https://android-arsenal.com/details/1/3070>, accessed at 23.12.2016.

Emotion Ontology (2017): Bioportal article. The National Center for Biomedical Ontology. Available at <https://bioportal.bioontology.org/ontologies/MFOEM>, accessed at 15.01.2017.

Google documentation (2016). Cloud speech API. Available at: <https://chrome.google.com/webstore/detail/voice-recognition/ikjmfindklfaonkodbnidahohdfbdhkn?hl=en>, accessed at 1.10.2016.

IBM knowledge center (2015): Java performance monitoring. New York. Available at <http://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/com.ibm.aix.performance/java_perf_mon.htm>, accessed at 7.11.2016.

IBM Watson API (2016). Available at <https://www.ibm.com/watson/developercloud/tone-analyzer.html>, accessed at 21.12.2016.

IBM Watson pages (2016): The inside story of how the Jeopardy-winning supercomputer was born, and what it wants to do next. Available at <http://www.techrepublic.com/article/ibm-watson-the-inside-story-of-how-the-jeopardy-winning-supercomputer-was-born-and-what-it-wants-to-do-next/>, accessed at 12.01.2017.

IFPI. Global music report (2016). Available at: <http://www.ifpi.org/downloads/GMR2016.pdf>, accessed: 12.08.2016.

Inc journal. How Digital Marketing Is Changing the Music Industry (2016). Available at : http://www.inc.com/aj-agrawal/how-digital-marketing-is-changing-the-music-industry.html, accessed: 12.08.2016.

Music ontology (2016). Available at <http://musicontology.com/>, accessed at 15.1.2017.

The Echo Nest official documentation (2016), Avaliable at <http://the.echonest.com>, accessed at 28.12.2016.

The OMRAS2 Music Analysis and Feature Extraction Service documentation (2015). Avaliable at <http://www.isophonics.net/sawa/>, accessed at 17.01.2017.

Tone Analyzer (2016). Available at <https://www.ibm.com/watson/developercloud/tone-analyzer.html> , accessed at 14.01.2017.