KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Perception of Fairness in Group Music Recommender Systems

Elisa Lecluse

Thesis voorgedragen tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: computerwetenschappen, hoofdoptie Mens-machine communicatie

Promotor:
Prof. dr. K. Verbert

Academiejaar 2019 – 2020

Master of Science in de ingenieurswetenschappen: computerwetenschappen

KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Perception of Fairness in Group Music Recommender Systems

Elisa Lecluse

Thesis voorgedragen tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: computerwetenschappen, hoofdoptie Mens-machine communicatie

Promotor:
Prof. dr. K. Verbert

Assessoren:
Dr. Y. Dauxais
Ir. P. Bartels

Begeleider:
Dr. N. N. Htun

Academiejaar 2019 – 2020

Preface

This last piece of work was one of the most interesting during my five years of studying Engineering Science. I rediscovered my passion for design and loved to see how the application developed from scratch. I would like to thank my thesis mentor Nyi-Nyi for assisting me every step of the way. I would also like to thank my thesis promotor Prof. dr. Katrien Verbert and the Augment research group for their valuable feedback. The time and effort of every participant to evaluate the system is very much appreciated as well.

Graag zou ik iedereen willen bedanken die in mij geloofde wanneer ik mezelf onderschatte. Hierbij denk ik vooral aan mijn ouders die me altijd gesteund hebben in zoveel meer dan ik zelf besef, mijn vriend Mathieu die me heeft leren relativeren en altijd vrolijk klaar staat om een dartpijltje te gooien, mijn grootouders die enthousiast de telefoon opnemen tijdens mijn thesispauzes en mijn fantastische vrienden die deze vijf jaar onvergetelijk hebben gemaakt.

Elisa Lecluse

Contents

Abstract

Fairness is an important aspect arising in group recommender systems. To satisfy all users, their preferences should be taken into account equally. Especially if a set of items is recommended to a group, the system should try to balance the user utilities inside the group. In group music recommender systems fairness ensures that even the songs of users with music taste diverging from the group are included in the recommended playlist. This research is unique as it is the first to consider fairness in group music recommender systems. After a few design stages, the objective of this research was defined. It focuses in particular on the influence of personality on the perceived fairness, the influence of the ranking algorithm on the perceived fairness and the overall usability of the system. A web application was developed to assist a group of users in creating a playlist together. A user can search songs and for every newly added song individual and group recommendations are provided using the Spotify Web API. All selected songs are ranked based on the probability that other group members will like them. The lowest ranked songs are not liked by the others and will thus not make it to the final playlist. In this fashion, the system can recommend a set of songs to the group that satisfies most group members. Here, the question can be asked if the users perceive this method as fair. Two versions of the ranking algorithm were integrated in the system: one based on the time the song was selected (time-based version) and one related to content-based filtering and aggregated predictions (dissimilarity-based version). Both algorithms first rank the playlist based on the number of likes (votes) a song received. 45 participants were recruited to conduct a within-subjects experiment in groups of 3. The results show that the dissimilarity-based version performs significantly better than the time-based version in terms of perceived fairness, but not in terms of predicting the popularity of the song. The participant's song ratings indicate that only the voting mechanism effectively ranks the songs. The responses to the fairness questionnaire were highly dependent on the number of songs each individual got included in the final playlist over the number of songs this individual selected. This makes it difficult to draw conclusions related to the influence of personality. Finally, the application was enthusiastically received by the participants and the overall usability scored excellently high. Therefore, this research can help to improve commercially available music applications.

Samenvatting

Fairness is een belangrijk onderdeel bij aanbevelingssystemen voor groepen. Om alle gebruikers tevreden te stellen, zou er gelijkmatig rekening moeten gehouden worden met de voorkeuren van elke gebruiker. Voornamelijk wanneer een verzameling van aanbevelingen wordt voorgelegd aan een groep, zou het systeem moeten proberen om per gebruiker evenveel relevante aanbevelingen op te nemen. In muziekaanbevelingssystemen voor groepen zorgt fairness ervoor dat zelfs de liedjes van gebruikers met een muzieksmaak verschillend van de rest van de groep toegevoegd worden aan de aanbevolen playlist. Dit onderzoek is uniek aangezien het als eerste fairness bestudeert in muziekaanbevelingssystemen voor groepen. Het doel van dit onderzoek werd gedefinieerd na enkele ontwerpfases. Het focust in het bijzonder op de invloed die persoonlijkheid kan hebben op fairness perceptie, de invloed van het rangschikalgoritme op fairness perceptie en de gebruiksvriendelijkheid van het systeem. Een web applicatie werd ontwikkeld om een groep gebruikers te helpen bij het maken van een gezamelijke afspeellijst. Een gebruiker kan liedjes opzoeken en bij elk nieuw toegevoegd liedje worden individuele en groepsaanbevelingen getoond met behulp van de Spotify Web API. Alle geselecteerde liedjes worden gerangschikt naargelang de kans dat andere groepsgenoten het liedje graag zullen horen. De laagst gerangschikte liedjes zullen de anderen niet graag horen en bijgevolg de finale afspeellijst dus niet halen. Op deze manier beveelt het systeem een reeks liedjes aan die de meeste groepsleden zullen appreciëren. De vraag kan hier gesteld worden of de gebruikers deze methode fair vinden. Het systeem bestaat uit twee versies: het eerste is gebaseerd op het tijdstip dat een liedje werd toegevoegd (versie 1) en het tweede is gelijkaardig aan content-based filtering en aggregated predictions (versie 2). Beide algoritmes rangschikken de afspeellijst eerst volgens het aantal likes/stemmen een liedje kreeg. In groepjes van drie ondergingen 45 deelnemers een within-subjects experiment. De resultaten tonen aan dat versie 2 het significant beter doet dan versie 1 wat fairness betreft, maar is niet beter in het voorspellen van de populariteit van de liedjes. De antwoorden van de fairness vragenlijst waren erg afhankelijk van het aantal liedjes van een bepaalde gebruiker in de finale afspeellijst ten opzichte van het totaal aantal geselecteerde liedjes van deze gebruiker. Dit maakt het moeilijk om conclusies te trekken omtrent de invloed van persoonlijkheid. Tenslotte werd de applicatie enthousiast ontvangen door de deelnemers en de gebruiksvriendelijkheid kreeg een uitstekende score. Dit onderzoek kan dus een unieke bijdrage leveren aan het verbeteren van commercieel beschikbare muziekapplicaties.

List of Figures and Tables

List of Figures

List of Tables

List of Abbreviations

BFI	Big Five Inventory
NEO PI-R	NEO Personality Inventory - Revised
Gold-MSI	Goldsmiths Musical Sophistication Index
HCI	Human Computer Interaction
API	Application Programming Interface
MEAN	MongoDB, Express.js, Angular and Node.js software stack
SDK	Software Development Kit
DSP	Digital signal processing
NLP	Natural language processing
SPA	Single-page application
SUS	System Usability Scale
ResQue	Recommender systems' Quality of user experience
GIF	Graphic Interchange Format

Chapter 1

Introduction

1.1 Motivation

As individual recommender systems gain in popularity and technologies facilitating remote communication are improving rapidly, the interest in group recommender systems clearly increases too [1]. One could think of many situations where an item can be recommended to a group of users. If a group of friends plan to watch a movie, a recommender system could support these users in their decision making process by taking into account each group member's preferences. A recommender system could in this case not only be used to select a movie, but also to schedule the movie night. Even in software engineering a set of requirements could be recommended in the early stages of development which satisfies all stakeholders [2]. In this case, item-to-group recommendations are further extended to package-to-group recommendations [3].

Recommending to groups is an even more complicated matter than recommending to individuals. Individual models need to be aggregated while ensuring all preferences are taken into account and while maintaining consistency [4]. It is much harder to please a group of users than an individual. Additionally, psychological factors play an important role in group recommender systems. Personality, emotions, and group dynamics should be considered [5]. Social interactions and relationships influence the preferences of group members and invalidate the assumption that preferences are independent. In sequential recommendations the system should remember which group members were not satisfied with the previously given recommendation [6]. If personality is taken into account, more weights could be assigned to a more assertive individual than to a more altruistic one. Especially in sequential and package-to-group recommendations the system could aim to maximize both the perceived fairness and the overall user satisfaction. After all, users tend to be more satisfied when they perceive a higher level of fairness [7].

In the music domain, popular streaming services like Apple Music, Spotify and YouTube Music are adding more and more features for groups. At the time of writing, Spotify launched a new feature, Group Session[1]. By scanning a code, someone can play music on the device of the person who shared the code. A shared queue

[1] support.spotify.com/is/using_spotify/features/group-session

allows multiple users to play and share music without needing to request a song. Users can now not only create collaborative playlists but also add songs to a shared queue real-time. Group recommendations are also emerging in music applications. Customers sharing a plan receive group recommendations in their 'Family Mix'[2] based on the songs they individually listen to. As recommending a set of songs to a group can in particular be considered as package-to-group recommendations, it gets more interesting when these group recommendations would take fairness into account and maybe even personalities. This new perspective could contribute to optimizing group recommendations in music applications.

1.2 Problem statement

To the best of our knowledge, perceived fairness in group music recommender system has not been studied before. Fairness is an important issue that arises in group recommender systems, especially when a set of items is recommended. Consider a group wanting to compose a playlist for a party, dinner, road trip etc. A convenient way of doing so could be to select a large amount of songs and only keeping the songs appreciated by most users. The question can be asked if the users perceive this method as fair. This research investigates the perception of fairness in a group music recommender system that ranks the songs based on the probability that the other group members will like the song. The study is then broken down into three more specific aspects:

- The influence of personalities
- The influence of the ranking algorithm
- The usability of the system

Accordingly, the goal of this research is to develop a group music recommender system that supports users in creating a playlist together and investigate how different personalities affect user perception in terms of fairness. In addition, the influence of the ranking algorithm is analysed.

1.3 Overview

Figure 1.1 visualizes the overview of the chapters. Group recommender systems and some of their challenges related to this research are presented in the literature review (Chapter 2). The state-of-the-art individual recommendation methods and aggregation strategies are discussed. Existing research on facilitating group decision making processes show that there are many ways to support consensus achievement. Moreover, the multiple objective optimization problem of maximizing both fairness and the overall group satisfaction is proven to be extremely complex. In addition, some group music recommender systems for public spaces or to help a group of friends

[2] support.spotify.com/us/account_payment_help/premium_for_family/family-mix

create a playlist together are presented. To get a better insight into personality psychology, the general Big Five taxonomy and research related to these personality dimensions are discussed. This includes interpersonal implications and the influence of personality on music taste and musical sophistication. Chapter 3 discusses the design stages of this research. Two prototypes preceded the final design. At each iteration, feedback was collected to integrate in the next interface design. The objective changed in this process to eventually get to a more relevant research. For each prototype the objective, components and evaluation are discussed. The final interface design described in Chapter 3 was implemented as a MEAN application (MongoDB, Express.js, Angular, Node.js) using the Spotify Web API to get music recommendations. The implementation is discussed in Chapter 4. The decisions made concerning the ranking algorithm and the recommendations are explained. Socket.IO is used to enable real-time collaboration. As a result, the group members are immediately updated when new songs or recommendations are added. A within-subjects user study was conducted with the developed group music recommendation system. Its method, procedure and results are presented in Chapter 5. The user study was designed to be online conductible without much assistance of the researcher. An interactive tutorial was integrated in the study to make the participants experiment with the system and avoid learning effects during the study. The results from the user study and correlations between them are discussed in more detail in Chapter 6. Perceived fairness, correlations between the Big Five dimensions and perceived fairness, the influence of the ranking algorithm on the perceived fairness and the usability of the system are analysed. This chapter also discusses the limitations of this research. The concluding chapter summarizes the main findings of this research and specifies opportunities for future work.

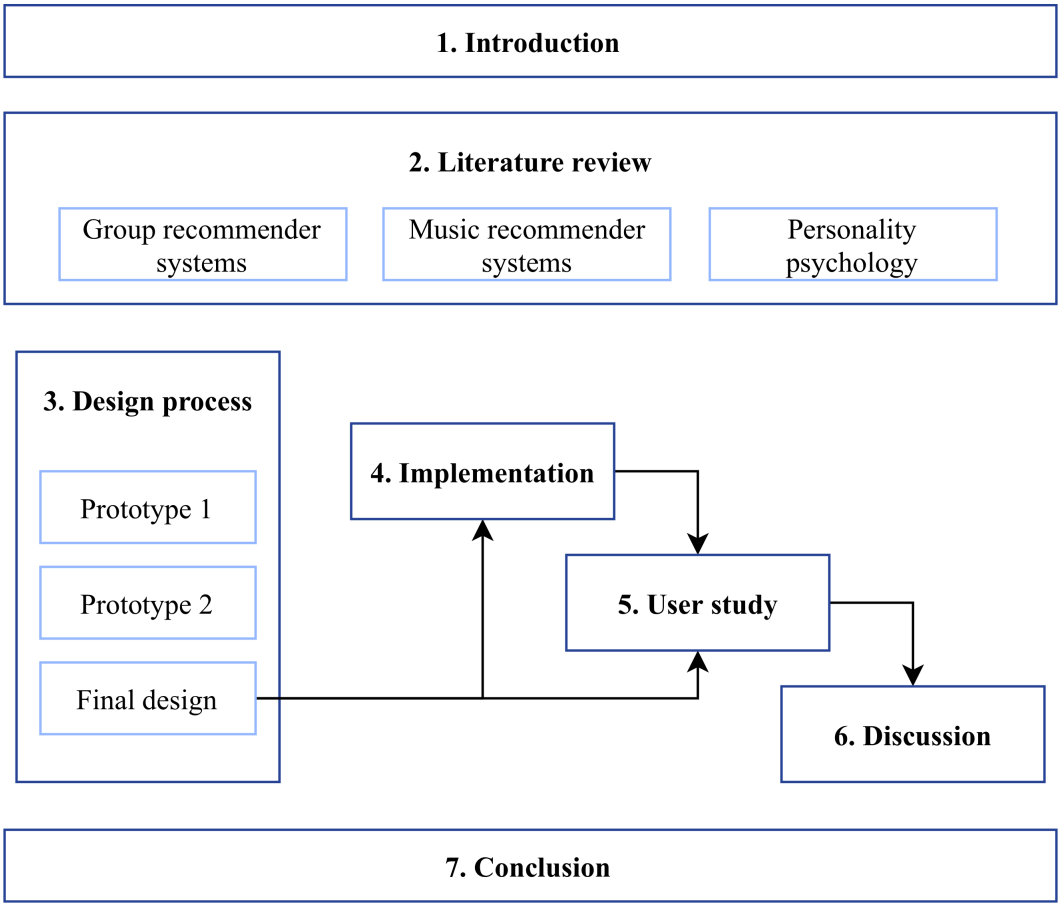


Figure 1.1: Overview of the chapters

Chapter 2

Literature review

This chapter gives an overview of the existing research in the domain of group recommender systems. First, the complicating factors of implementing group recommender systems are discussed. Compared to individual recommender systems, there are some additional aspects that need to be taken into account in group recommender systems. The group recommender systems section discusses three of them most relevant to this research: aggregation strategies (how to combine individual models), achieving consensus (optimizing the level of agreement between group members) and fairness (balancing the relevance of the recommended items to each user inside the group). Second, existing music recommender systems are presented. Third, theories of personality psychology are explained. The Big Five personality dimensions and its application in research domains such as interpersonal functioning and music are discussed in detail.

2.1 Group recommender systems

The basis for group recommender systems are individual recommender systems. Imagine that the perfect individual recommender exists. It knows what the user wants and how to generate good individual recommendations. How do we use the recommendation paradigms for individual users in a group scenario? Not only does the question arise how to combine these individual models, but also how to handle group-specific psychological factors [1]. The research domain of individual recommender systems is already quite diverse. Single-user recommender systems integrate numerous types of user preferences and requirements through well-known methods like collaborative filtering, content-based and knowledge-based recommendation, or more specialized methods tailored to the data domain and context [8]. There are two main strategies to aggregate individual models in group recommender systems [9]. The first strategy, *aggregated predictions*, aggregates individually generated recommendations (recommendation precedes aggregation). The second strategy, *aggregated models*, constructs a group preference profile by aggregating the individual user preferences (aggregation precedes recommendation) [4]. There are many functions for aggregation that try to take into account the group members' preferences. Aggregation functions can

be classified in three categories: majority-based, consensus-based and borderline functions [10]. This section discusses the basic recommendation methods, aggregation functions, how the system can help to achieve consensus and the fairness aspect in group recommender systems.

2.1.1 Recommendation methods

Collaborative filtering The ratings provided by other users are used in collaborative filtering techniques to generate recommendations [8, 11]. Users with the same preferences are the 'nearest neighbours' of the current user and they provide a way to predict what this user may like. Using the aggregated predictions strategy in combination with collaborative filtering, the first step is to determine a predicted rating for each item that a group member did not rate yet [4]. The second step can be approached in two ways. The first option is to use an aggregation function to rank all items based on the individual ratings and take the highest ranked item to recommend to the group. The second option is to merge the individually recommended items. The group is then presented the union of the highest ranked items per group member. As a result, they are responsible for ranking the items if desired. In the aggregated models approach, collaborative filtering is applied to group profiles. This group profile consists of aggregated individual ratings per item. Accordingly, the group recommendation is generated based on the ratings of 'nearest neighbour' groups.

Content-based Items are recommended based on their 'content' in content-based approaches. Closely related items will have similar descriptive attributes [8, 11]. In this fashion, a user can get a recommendation based on a previously liked item. When combining the aggregated predictions approach and content-based filtering to generate group recommendations, the similarity between items not yet rated by the group members and their individual profiles is first determined [4]. These similarities are used as input to an aggregation function to define the probably most liked item by the group. Similar to the collaborative filtering approach, the union of the individually recommended items can also be returned as group recommendation. Hence, per group member the items with the highest similarity are merged. Regarding aggregated models, a group profile is constructed by aggregating the individual preferences in terms of the descriptive attributes. The item most similar to the group profile is given as recommendation.

Knowledge-based In knowledge-based recommenders, the user is expected to have specific domain knowledge [11]. Users need to inform the system about their preferences in terms of certain item features (e.g. number of rooms in an apartment). These systems can be categorized in constraint-based and case-based recommender systems. The former requires the user to be able to specify his/her requirements and preferences whereas the latter generates recommendations based on similarity metrics (the similarity between candidate items and a given case description is calculated). Here, the aggregated predictions and aggregated models approaches are discussed for constraint-based recommendations. For both approaches a user needs to define

requirements and preferences [4]. These preferences can be expressed by assigning weights to interest domains. The aggregated predictions strategy then determines user-specific utility scores per item (displayed in Table 2.1). These scores represent how relevant an item could be to a user. The utility score of items not meeting the user's requirements is set to 0. The utility scores are aggregated to generate recommendations. The items with the highest score per group member can also be merged to return a list of recommended items. The strategy of aggregated models gives the user requirements and preferences as input to an aggregation function to construct a group profile. Utility scores are again calculated, this time based on the group profile utility weights. The item with the highest utility score is selected. Inconsistencies in requirements can occur when constructing a group profile. In this case, the group member will need to alter their requirements.

	Security	Attractiveness	Crowdedness
item (scale 1-5)	4.0	5.0	2.0
user (weights)	0.3	0.6	0.1
	$4.0 \times 0.3 + 5.0 \times 0.6 + 2.0 \times 0.1 = \mathbf{4.4}$		

Table 2.1: Utility score calculation for a specific user and item (e.g. holiday destination)

2.1.2 Aggregation functions

Majority-based Plurality Voting, Borda Count and Copeland Rule are examples of majority-based aggregation functions [4]. These mechanisms recommend the most popular items to the group. Plurality Voting selects the item with the highest number of votes. In contrast to Approval Voting where users show their approval by voting for the approved items, in Plurality Voting every group member votes for one item. For Borda Count, the group members' evaluations are used to rank the items. The position of the ranked items per group member are added together and the item with the best total ranking score is chosen. Copeland Rule recommends the item performing best on pairwise evaluation comparison. Per group member the evaluation of one item is compared to another. A score of -1, 0 or +1 is given when the item has a worse, equal or better evaluation than the other item, respectively.

Consensus-based Preferences of all group members are taken into account in consensus-based aggregation [4]. Additive Utilitarian adds the users' evaluations together and selects the item with the highest sum. Other consensus-based aggregation mechanisms recommend the item with the highest average of the individual evaluations or the item with the highest product of individual evaluations. Taking the highest average and Additive Utilitarian sacrifice the preference of the minority. Their judgment counts less in larger groups. Average without Misery solves this problem by eliminating the items with an individual evaluation below a defined threshold. An alternative is to return a ranked list based on the assumption that every group member would choose an item in turn.

Borderline Borderline functions take a subset of the user preferences to recommend items [4]. Least Misery for instance generates a recommendation based on the lowest evaluations per item. The highest evaluated item of this subset is selected. This option may give too much power to a minority or select items nobody really likes and exclude all the favourites because of one low evaluation. Most Pleasure on the contrary may select items only a few group members really like by selecting the item with the highest individual evaluation (of all evaluations). Majority Voting lists per item the evaluation score that was given by the most group members and selects the item with the highest common evaluation. Another borderline aggregation mechanism is to recommend the item that received the highest evaluation score of the most respected group member (Most Respected Person).

2.1.3 Achieving consensus

Group recommender systems facilitate the decision making process for groups. Generally, a solution to a group decision making problem is found after aggregating the individual preferences and selecting an item or subset of items based on those preferences [12]. However, this solution is not guaranteed to be collectively accepted by the group. To obtain a higher level of agreement, an extra stage should be added to this decision making process [12]. At this stage, the group members discuss and modify their preferences to eventually achieve consensus. The consensus reaching process is shown in Figure 2.1. In the first phase, the individual preferences are compared to each other or to the group profile (the result of aggregating the individual preferences). A consensus degree is determined based on the 'distance' between the compared preferences. The second phase applies a consensus threshold (defined beforehand) to the measured consensus to decide whether consensus is achieved or not. If not, the group proceeds to phase three. In traditional consensus processes, a moderator provides feedback and advises group members farthest from consensus to alter their preferences. In group recommender systems the preferences of group members are automatically updated or weights are given to some preferences [13], replacing the human intervention of a moderator. To not loop forever in this consensus reaching process, a maximum number of discussion rounds can be defined.

Castro et al. [13] developed a consensus-driven group recommender system by

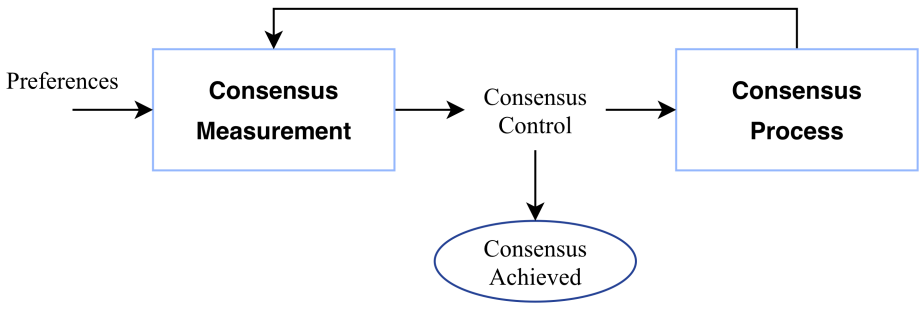


Figure 2.1: General consensus reaching process [12]

implementing the consensus reaching process described above. The system first generates individual recommendations, using the top items as ordered preferences for the consensus phase. The preferences are then used as input to the consensus reaching process to recommend items with a high probability of being accepted by the group. ConsensUs [14], on the contrary, does not automatically update the group member's preferences. Instead, it visualizes points of disagreement (Figure 2.2). A between-subjects experiment reported significantly more changes in user-given scores for the group interface than for the individual interface.

Another way of supporting group decision making is presented by Nguyen et al. [15]. Recommendations and explanations engage users to participate in a discussion (Figure 2.3). A user can browse places of interests using a mobile application. After inviting other users, the user can propose an item to the group. Group members can rate the proposed items by giving a thumbs up, thumbs down or selecting it as a favourite. These ratings are shown to every group member. Users can request recommendations and view the corresponding explanations. Sometimes recommendations are automatically provided. Hints are offered to give more information about the items are the functioning of the system. If the group is unable to select an item, they can ask the system to give a ranked list of all proposed items combined with

Figure 2.2: The ConsensUs independent opinion interface (left); the ConsensUs group visualization interface (right) [14]

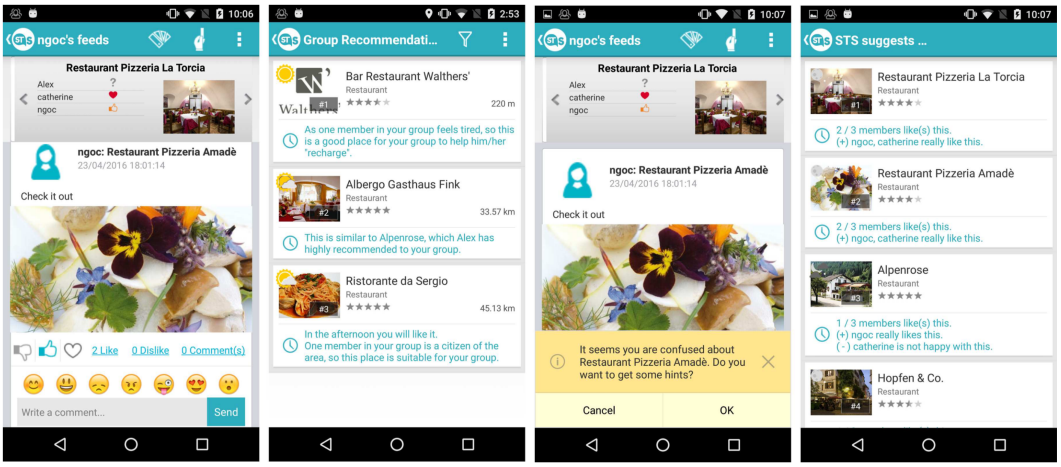


Figure 2.3: Recommendations and explanations to support group decision making: 1) group discussion, 2) recommendations, 3) hints and 4) final choice suggestions [15]

an explanation regarding the item's position in the list. The recommender system thereby helps the group members to agree on a proposed item.

Using an interactive tabletop, CATS [16] combines face-to-face collaboration and a recommender system to help a group of friends in their decision process. The CATS environment assists users to select a ski holiday package. Users sit together around a display showing a map of Europe and its ski resorts (Figure 2.4). Each user can touch one of the mountain icons representing the resorts displayed on the tabletop to get detailed information. They can rate the available hotels and see the preferences of other group members. The most popular hotels are listed first. On the map the users can see which resorts the others are looking at. The users are represented by coloured snow flake icons on the map. The bigger the size of the snow flake, the more the user prefers this resort. The ski resort icon size increases if the overall group preference increases for the resort. The bigger the icons, the more the users will pay attention to it and the faster the group can select a resort for their ski holiday.

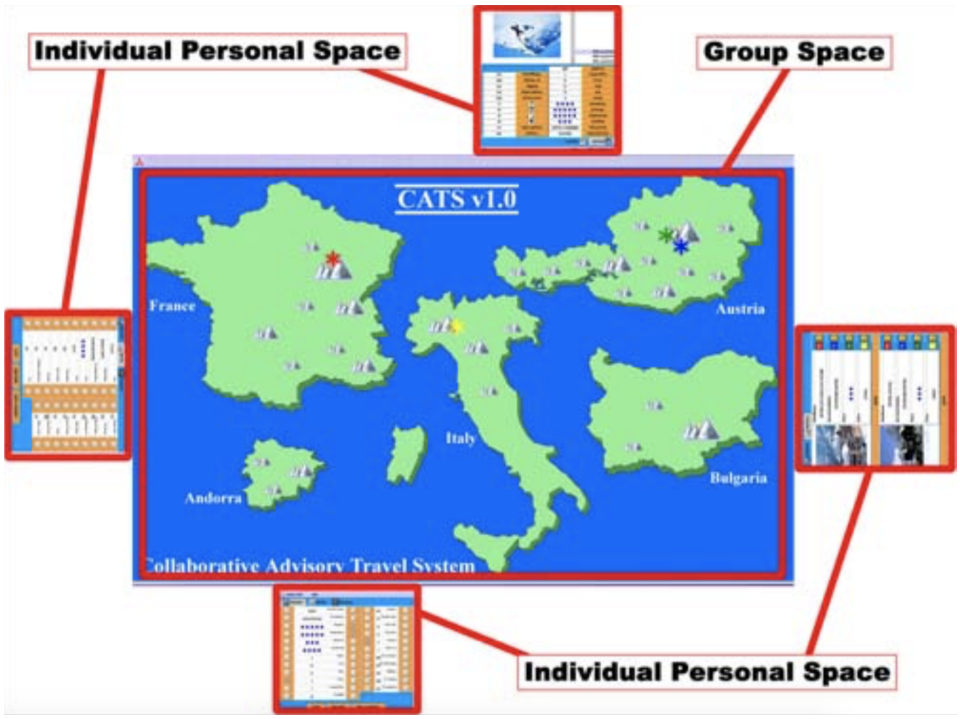


Figure 2.4: CATS interface

2.1.4 Fairness

The term user utility is used to describe the relevance of an item to a user [17]. Increasing utility thus implies increasing user satisfaction. Accordingly, Xiao et al. [17] defined fairness as the "*balance of user utilities inside the group*". It is easy to see that in heterogeneous groups it is very difficult to satisfy every user. In this context, Xiao et al. formally proved that the problem of maximizing both fairness and group member satisfaction is NP-hard[1].

Qi et al. [3] were the first to introduce the problem of recommending packages to groups. They propose probabilistic models to integrate group preferences regarding packages. Fairness is an issue that especially arises in recommending packages. The recommended package could satisfy a majority of the group while some users don't have any of their preferred items in the package. The main objective of Qi et al. is to propose efficient package-to-group recommendation algorithms that prioritize qualitative package recommendations. The work of Serbos et al. [20] focuses more on the fairness aspect in package-to-group recommendations. They propose methods to optimize fairness directly and model it as a set coverage problem. This optimization problem tries to maximize the number of users satisfied by an item of the recommended package (a user satisfied by an item is 'covered' by this item). This problem is also considered NP-hard. It differs from the multiple objective optimization problem described by Xiao et al. [17] in which they not only maximize fairness, but also the overal social welfare (satisfaction of each user). It also differs from fairness in sequential group recommendations [6]. When a user was not satisfied with the previous recommendation, the preferences of this user will prevail in the next recommendation round to generate fair sequential group recommendations.

Burke [21] describes fairness-aware recommender systems from another perspective. The 'group' is considered as a group of multiple stakeholders: consumers, providers and the platform presenting recommendations to the consumers. The recommender system used by the platform serves as a consumer-provider pairing tool. It is beneficial for the platform to balance the consumer and providers utilities in the recommendation process. Tran et al. [7] investigated the perceived fairness in textual explanations. The recommendation explanations were classified in explaining the preference aggregation strategies, explaining both the strategy and the decision history and explaining both the strategy and the future decision plan. The latter showed the strongest positive correlation with perceived fairness. Presenting the previous decisions and corresponding user satisfactions also helped to improve perceived fairness in groups.

[1] In complexity theory, a problem is NP-hard if every decision problem in NP can be reduced in polynomial time to this problem, making it at least as difficult as every problem in NP [18] (e.g. the Travelling Salesman Problem is NP-hard [19]).

2.2 Music recommender systems

Well-known music streaming services like Spotify, Apple Music, YouTube Music have the users and data to apply specialized methods and to generate good music recommendations. Besides, they are releasing more and more group-specific features like collaborative playlists, family recommendations, shared queues etc. Users can invite friends to compose playlists together or to play music on the same device. Customers with a family plan can play group recommendations based on the music they individually listen to. This section discusses other work also related to the group music recommendation research domain.

MusicFX

In a fitness center called Fitness Xchange (FX) with over 600 members, MusicFX [22] improved the music selection by taking into account the preferences of the present members. Preferences for musical genres are collected from every new member and registered members can update their preferences at any time. Based on the aggregated popularity of a specific music genre [2], MusicFX selects a station to play. By scanning a badge or logging in manually, users inform the system of their presence. Each time a user updates his/her preferences, enters or exits the fitness, the shared preferred music genre is updated. 71% of the more than 25% responses reported increased satisfaction. These members liked the music selected by MusicFX better than before this recommender system was used by the fitness center. In addition, most members enjoyed the increased variety of selected songs. The application of MusicFX need not be restricted to fitness centers. It can be used in any public environment in which people stay for some time.

Flytrap

Similar to MusicFX, Flytrap [23] is aware of the presence of its users. The system determines the proximity of the users by receiving radio frequency signals of the wearable ID badges. Flytrap uses an alternative preference elicitation technique. The songs users listen to on their personal computer are recorded and send to the Flytrap database combined with some metadata including artist and genre. Similarity between artists is determined using a network of genres. When a user's presence is detected and a new song needs to be selected, votes are given to the songs the present users would most probably like based on each user's preferences. Consequently, the songs listened to on the personal computers of the users and songs similar to these previously listened songs will get a high vote. High votes increase the probability that a song is played next. The system also avoids to subsequently play songs by the same artist and tries to maintain smooth transitions.

(a) (b) (c)

Figure 2.5: Components of the interactive Jukola Jukebox [24]

Jukola

The interactive Jukebox, Jukola [24], enables its users to vote for the next song to be played in a shared space (e.g. a café bar). The system consists of three components: a public display (Figure 2.5a), handheld clients (Figure 2.5b) and a web page (Figure 2.5c). Customers can search for songs and nominate them using the touch screen public display. A combination of four nominated songs and songs randomly picked from the music collection of the owner of the place are shown on the handheld devices. The handheld devices allow one vote every time a new song needs to be selected. The popularity in terms of the percentage of votes received for each of the four songs is displayed. The most popular song is then played next. A user can visit the Jukola web page to explore the songs played on a specific day. This gives an overview of what kind of music is played at that place and helps users to recollect the songs they listened to.

Adaptive Radio

In Adaptive Radio [25], a group of users can listen to broadcasted songs selected based on their negative preferences (Figure 2.6). Users can 'censor' or 'skip' a song to update their preference profile in the music recommender system. The system will then eliminate similar songs for future selection when this user is present. A song can only be skipped when this user is alone or when it is clear that the other group members don't like the song. This environment tries to find consensus solutions that are implicitly instead of explicitly approved by every group member [2]. This set of solutions is larger than for positive preference elicitation as songs can be played that may not be preferred but are still acceptable by the group members. Besides, when users need to express their positive musical preferences, they will get recommendations they are familiar with. In this way, Adaptive Radio introduces users to more new music they might appreciate.

Figure 2.6: Adaptive Radio interface [25]

Figure 2.7: GroupFun 'home' tab [26]

GroupFun

GroupFun [27, 26] (Figure 2.7) is implemented as a Facebook plugin and recommends group playlists. Users can invite friends to upload and rate music. Based on these ratings and the individual playlists they create, the system recommends a group playlist for specific events. The recommendation algorithm applies a probabilistic weighted sum aggregation function to determine a song's popularity in the group. The higher the popularity the higher the chances the song will be included in the recommended playlist.

2.3 Personality psychology

In personality psychology, the 'Big Five' personality dimensions offer a general model to describe personality traits [28]. These dimensions emerged empirically by analysing how people lexically describe themselves and others. This common framework permits researchers to study how personality influences certain aspects in life and to easily communicate their observations. The factors labeled as Extraversion, Agreeableness, Conscientiousness, Neuroticism and Openness do not reduce differences in personality to only five traits but represent personality at a very broad level of abstraction, hierarchically combining numerous specific traits. Costa and McCrae demonstrate this hierarchy by defining more specific facets per dimension of the Big Five (Table 2.2). Their 240-item NEO Personality Inventory - Revised (NEO PI-R) [29] (cited by [28]) allows to assess an individual's Big Five dimensions in terms of these facets.

To avoid misunderstanding of the traditional labels, John and Srivastava [30] gave the following short conceptual definitions and associations for each of the five dimensions to illustrate the broad range of their meaning:

I. **Extraversion** (Energy, Enthusiasm) "implies an *energetic approach* to the social and material world and includes traits such as sociability, activity, assertiveness, and positive emotionality."

II. **Agreeableness** (Altruism, Affection) "contrasts a *prosocial and communal orientation* toward others with antagonism and includes traits such as altruism, tender-mindedness, trust, and modesty."

III. **Conscientiousness** (Control, Constraint) "describes *socially prescribed impulse control* that facilitates task- and goal-directed behavior, such as thinking before acting, delaying gratification, following norms and rules, and planning, organizing, and prioritizing tasks."

IV. **Neuroticism** (Negative Affectivity, Nervousness) "contrasts emotional stability and even-temperedness with *negative emotionality*, such as feeling anxious, nervous, sad, and tense."

V. **Openness** (Originality, Open-Mindedness) "describes the breadth, depth, originality, and complexity of an individual's *mental and experiential life*."

The full 240-item NEO PI-R would be the most useful instrument to measure an individual's Big Five personality dimensions when the participants' time is not too limited [30]. Unfortunately, this is usually not the case. The 44-item Big Five Inventory (BFI) [31], the 60-item NEO Five-Factor Inventory (NEO-FFI) [29] and Goldberg's 100 trait descriptive adjectives (TDA) [32] serve as good alternatives [30]. The BFI consists of short phrases based on prototypical Big Five trait adjectives. It is chosen for this research because of its efficiency and flexibility in the assessment of the five dimensions. Compared to the NEO-FFI (± 15 min) and the TDA (± 15 min) the BFI takes approximately 10 minutes less to complete [30]. Moreover, using single adjectives as items (TDA) can be ambiguous in their meaning and thereby

Big Five Dimensions	**Facet (and correlated trait adjective)**
Extraversion vs. introversion	Gregariousness (sociable) Assertiveness (forceful) Activity (energetic) Excitement-seeking (adventurous) Positive emotions (enthusiastic) Warmth (outgoing)
Agreeableness vs. antagonism	Trust (forgiving) Straightforwardness (not demanding) Altruism (warm) Compliance (not stubborn) Modesty (not show-off) Tender-mindedness (sympathetic)
Conscientiousness vs. lack of direction	Competence (efficient) Order (organized) Dutifulness (not careless) Achievement striving (thorough) Self-discipline (not lazy) Deliberation (not impulsive)
Neuroticism vs. emotional stability	Anxiety (tense) Angry hostility (irritable) Depression (not contented) Self-consciousness (shy) Impulsiveness (moody) Vulnerability (not self-confident)
Openness vs. closedness to experience	Ideas (curious) Fantasy (imaginative) Aesthetics (artistic) Actions (wide interests) Feelings (excitable) Values (unconventional)

Table 2.2: Costa and McCrae's NEO PI-R facets [29]

cause inconsistencies in participants' answers. Additionaly, the BFI items are easier to understand than the NEO-FFI items.

Soto and John [33] developed scales to assess two specific facet traits per Big Five personality factor using the BFI questionnaire. They successfully attempted to resolve the low fidelity of analysing personality in only five domains by providing an overarching Big Five framework to examine personality hierarchically without using hundreds of items. Starting from NEO PI-R facets [29] closely related to personality characteristics in the BFI item pool, ten preliminary scales were identified: Assertiveness and Activity for Extraversion, Altruism and Compliance for Agreeableness, Order and Self-Discipline for Conscientiousness, Anxiety and Depression for Neuroticism and Aesthetics and Ideas for Openness. Each two facet scales related to the same BFI domain demonstrated noticeable intercorrelations and the ten developed scales showed substantial convergence with the corresponding NEO PI-R facets.

2.3.1 Interpersonal functioning

Du et al. [34] recently found that some social aspects can be deducted from all Big Five personality dimensions and its facets. The Big Five can help to interpret social behaviours and interpersonal functioning. Extraversion and Agreeableness are personality characteristics that are known to be closely related to interpersonal functioning, but the other factors also demonstrate some interpersonal implications. Higher levels of Neuroticism for instance indicate more cold-submissiveness, indifference and interpersonal problems. Self-consciousness, depression and vulnerability are associated with interpersonal difficulties in general and keep people from easily connecting with people. These facets are associated with interpersonal difficulties in general. Anxious individuals pressure themselves too much in properly handling social situations. Conscientiousness is generally associated with few interpersonal problems. Especially its facets competence, dutifulness, and achievement striving are positively related to efficient interpersonal behaviours. Conscientious people are seen as warm and nurturant. Like Conscientiousness, Openness is associated with warm traits. Agreeableness is related to warm and submissive characteristics. Agreeable individuals are trusting and for them it is important that others see them as trusting and cooperative. They need to be careful to not be profited from. Modest people can be bothered by other's domineering behaviours and the facet trust demonstrated sensitivity to cold behaviours. Agreeableness generally shows adaptive interpersonal functioning, meaning that these individuals rather not experience interpersonal problems. Lastly, Extraversion is related to warm-dominant profiles. Extroverted people tend to be good leaders and easily connect with others. For them it is important to have a competent and caring appearance. They are good at handling social situations and also report high interpersonal efficacy. In addition, Du et al. [34] found that people with opposite characters will probably be bothered by each other. Warm-dominant, extroverted people are for example sensitive to the withdrawn behaviour of cold-submissive, neurotic people.

2.3.2 Music and personality

Schäfer and Mehlhorn [35] analysed 28 studies related to music and personality. All studies used the Big Five taxonomy for personality traits and the five-dimensional MUSIC model (mellow, unpretentious, sophisticated, intense, contemporary) to categorize musical style preferences. They concluded that there is no empirical evidence that difference in personality traits is reflected in difference in musical preferences. The only Big Five dimension showing positive result was Openness to Experience. The trivial correlation between Openness and being open to all kinds of music was reported. Individuals who scored high on Openness liked music more in general. Schäfer and Mehlhorn addressed the difficulties arising when categorizing music. They question the reliability of global styles and genres as music preference categorizations. Using musical attributes on the contrary, which are more fundamental and objective, could offer a more appropriate solution to categorizing musical preferences. Furthermore, musical attributes are relatively insusceptible to rapidly evolving musical substyles. Unfortunately, although this is a promising approach for the categorization of musical preferences, Greenberg et al. [36] (cited by [35]) found no predictive potential in personality characteristics for attribute preferences. Based on a pilot study of Ferwerda et al. [37] Schäfer and Mehlhorn propose to rather use personality as a predictor for situation-specific music preference. That is to say, depending on the situation, people listen to different kinds of music. In this context Ferwerda et al. found that individuals open to experience rather browse music by mood, conscientious people rather browse music by activity and Neuroticism increases the chance that these individuals browse by activity or genre.

In another study of Greenberg et al. personality was researched as a predictor for musical sophistication (self-reported sophistication and behavioral test performance). Participants needed to indicate whether they played an instrument, if so, which one they master the most (including voice). The Goldsmiths Musical Sophistication Index (Gold-MSI) [38] was used to self-report their musical skills, abilities and behaviours. More extreme levels of sophistication were measured by a melodic memory test and a beat perception test. Using the BFI and its ten facet scales, the strongest correlation was found for Openness to Aesthetics not only for all musical sophistication domains, but also performance on the musical ability tasks. The Ideas facet of Openness on the other hand did not correlate. Vuoskoski and Eerola [39] studied the predictor role of personality traits for perception of emotion in music. According to their research, personality moderates the consistency between an individual's emotional state and the situation at that time. Their hypotheses related to the congruence of traits showed consistency with the results: higher Neuroticism indicates higher sadness ratings and higher Extraversion indicates lower sadness ratings.

2.4 Conclusion

This chapter listed three of many challenges regarding group recommender systems. Mechanisms to aggregate individual models and studies related to achieving consensus and integrating fairness were addressed as they are most relevant to this

research. Next, some of the existing group music recommender systems were explained (MusicFX, Flytrap, Jukola, Adaptive Radio and Groupfun). Lastly, the Big Five personality taxonomy was explained and will be used in this research to assess personality traits of users and to report observations. Thereafter, this literature review discussed some of the general findings in personality psychology and findings related to interpersonal functioning and music.

Chapter 3

Design process

To design a user-friendly and effective system for creating playlists in group, two iterations preceded the final design. After composing each prototype in Figma[1] (an online interface design tool), feedback was collected from the research group and test users. Processing this feedback resulted in changes in not only the interface design but also the objective of the group recommender system. The design stages in terms of the system's objective are shown in Figure 3.1. This chapter discusses in detail the features of each digital prototype and how the evaluation of one prototype produces the requirements for the other.

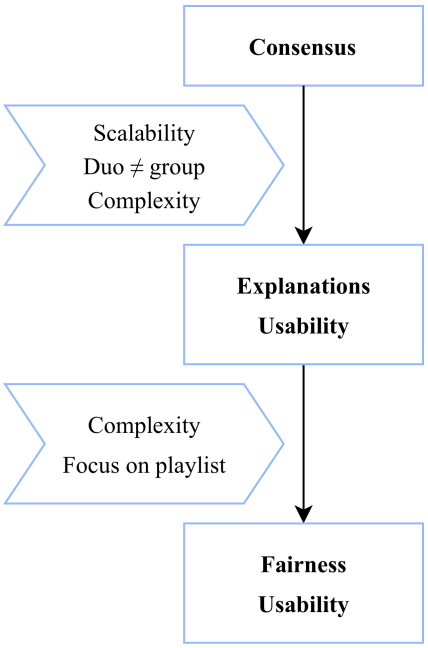


Figure 3.1: Change of objective after evaluating the prototype (stages of the design process)

[1] www.figma.com.

3.1 First prototype

The layout of the individual music recommender system developed by Millecamp et al. [40] and the graph used in the recommender system PeerChooser [41] (Figure 3.2) inspired the design of the first prototype. The prototype consists of three versions of an interface for the group recommender system. The interfaces were presented to the Augment research group of the KU Leuven. Their valuable feedback contributed to the change in focus of the group recommender system. This section describes the objective of the first prototype, the components of the different interface versions and the evaluation.

(a) (b)

Figure 3.2: Inspiration sources for the layout and force-directed graph of the first prototype [40, 41]

3.1.1 Objective

The major question that we want to solve here is how to provide a tool to help users easily create a playlist with other users. As music taste differs per person, it can be hard to meet everyone's needs. This gets even more complicated when people do not have the time to physically meet and discuss their preferences. This first prototype is designed to try to solve this problem using negative preferences. According to the results of Adaptive Radio [25], it is easier to satisfy users by not including their disliked songs than by searching songs that everyone likes. In this manner, users implicitly approve songs by not vetoing them. A consensus solution can then be found by listing all the songs that no one vetoed. The interface is split up in three versions. Each of those versions incorporates the veto mechanism but differs in extra features. In this way the effectiveness of each feature can be researched in terms of achieving consensus in remote collaborations. The next subsections describe these three versions and their components in detail.

3.1.2 Version 1

The interface discussed in Section 3.1.4 was stripped down to a basic version (version 1: Figure 3.3) and an intermediate version (Section 3.1.3) to be able to gauge the effectiveness of each added component. The following paragraphs explain the components of version 1 from the left side to the right side of the interface (Figure 3.3).

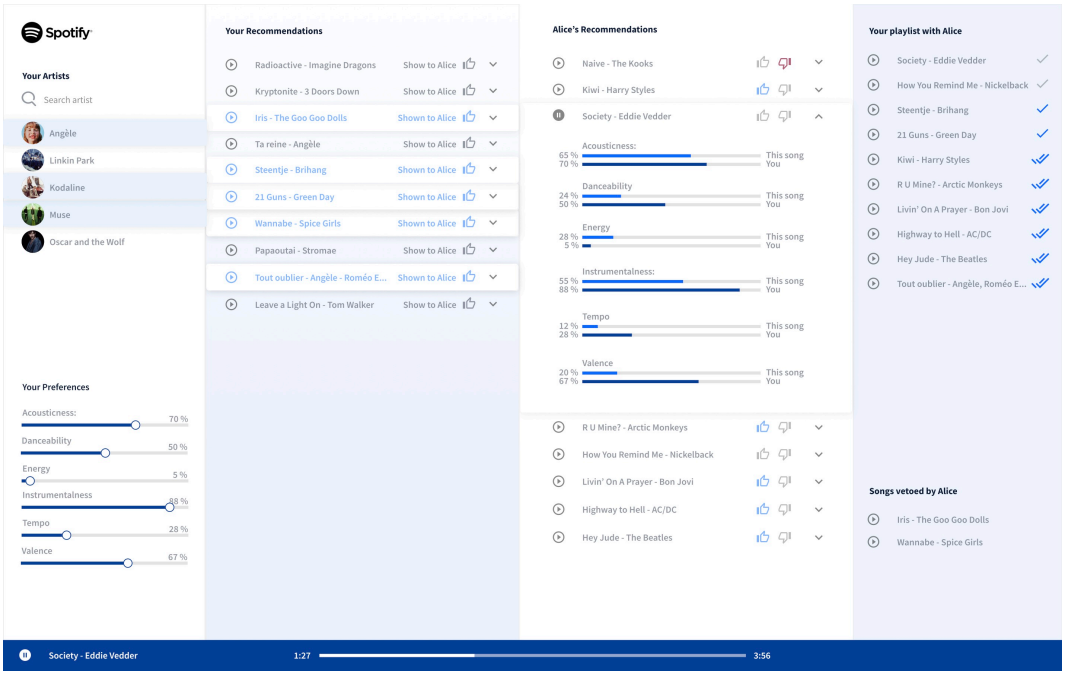


Figure 3.3: First prototype interface 1: basic view

Your artists and preferences The first thing the user needs to do is select some of his/her favourite artists and give in the preferences per track attribute. The track attributes and their definitions according to the Spotify Web API [42] are listed below.

- **Acousticness:** a confidence measure indicating whether electrical amplification is used or not.
- **Danceability:** *"how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity."*
- **Energy:** *"a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. [...] Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy."*

- **Instrumentalness:** *"predicts whether a track contains no vocals. [...] Rap or spoken word tracks are clearly 'vocal'."*
- **Tempo:** *"the overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration."*
- **Valence:** *"musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry)."*

This selection of attributes out of the total 14 track attributes provided by the Spotify Web API is based on the work of Millecamp et al. [43]. Each attribute's value would be converted to a percentage (range from 0% to 100%).

Your recommendations Based on the given artists and preferences, recommendations are generated. The recommended songs are shown one by one with title, artist and a play button ⏵ to listen to the song. When playing the song, it is shown in the footer of the web page together with its progress. If the user likes the song, he/she can click on 'Show to X 👍' where X represents the person the user is making a playlist with. In this example, the other person is named Alice. When clicking on this button, the song will be proposed to the other person (Alice). The ⌄ icon represents the possibility to receive more detailed information on why this song was suggested. The content shown when this button is clicked will be explained in the next paragraph.

Alice's recommendations When a user likes a song by clicking on 'Show to Alice 👍' in the 'Your recommendations' component, it is send to the other user (Alice) to ask for approval. Here, Alice liked Naive by The Kooks. The user can approve the suggested song by clicking on 👍 or veto the song by clicking on 👎 . Analoguous to the 'Your recommendations' component, a user can listen to the proposed songs by clicking on ⏵ and ask for more details about the song via the ⌄ button. The detailed view shows the value of each track attribute (acousticness, danceability, energy, instrumentalness, tempo and valence) of the song. The user can compare these values to his/her preferences given as an input for recommendations. The percentages of each track attribute are presented in a bar chart for both the song and the preference. The detailed view can be closed by clicking on the ⌃ icon.

Your playlist with Alice Every song a user likes by clicking on 'Show to Alice 👍' in the 'Your recommendations' component, is added to the playlist. The playlist is shown on the right in the interface. A grey check mark next to a song in the playlist means that Alice's song is waiting for the user's approval. For example, this user still needs to approve Society by Eddie Vedder. One blue check mark means that the user's selected song (e.g. Steentje by Brihang) is waiting for Alice's approval. Double check mark indicates that this selected song is approved (e.g. Kiwi by Harry Styles).

Songs vetoed by Alice A user has the right to veto a song proposed by the other person. By clicking on the 👎 symbol for a song in the 'Alice's recommendations' component, the song will move from 'Your playlist with Y' to 'Songs vetoed by Y' in Alice's view (Y symbolizes the user interacting with the interface shown in Figure 3.3). In this case, Alice vetoed the songs Iris by The Goo Goo Dolls and Wannabe by the Spice Girls.

3.1.3 Version 2

Version 2 (Figure 3.4) uses all components of the basic interface (Figure 3.3) and adds some extra information about the collaborating user. The added features are explained below.

Extra bar charts In the 'Your preferences' component on the left, the preferences of the other user (personified by 'Alice') are presented as a light blue bar chart below the slider for the user, for each track attribute. This is also the case in the detailed view. If the ⌄ button is clicked in the 'Your recommendations' or 'Alice's recommendations' component, the user can see not only the track attribute's value of the selected song and his/her own preferences, but also the preferences for each track attribute of the other person (Alice). This enables the user to take into account the preferences of the other person when selecting/approving songs, if the user wants to.

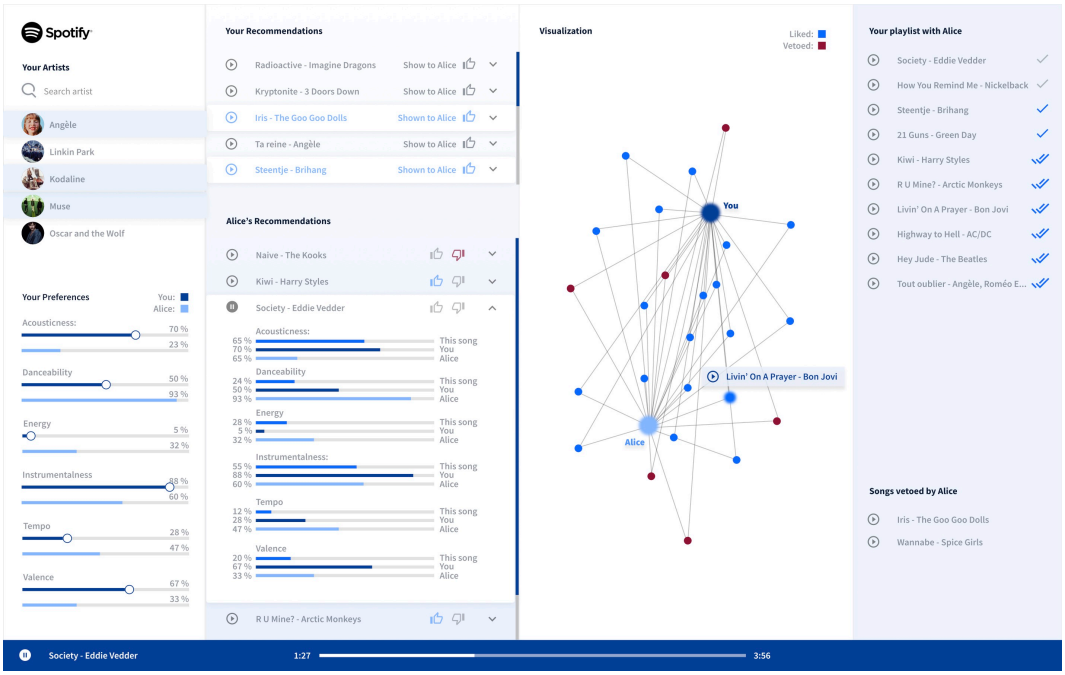


Figure 3.4: First prototype interface 2: bar charts and force-directed graph added

Force-directed graph In this version the 'Visualization' component takes the place of the 'Alice's recommendations' component. The latter is moved below the 'Your recommendations' component. The visualization consists of a slightly modified force-directed graph. Instead of trying to minimize the difference in length between nodes and the amount of crossing edges [44], the length between the nodes (representing songs) is defined by the difference between the songs. This difference can be calculated by taking the Euclidean distance between the track attributes' vectors (an entry for each track attribute value). The big dark blue and light blue nodes represent the profiles/preferences of respectively the user and Alice (the other person). The distance between the song nodes and the profile nodes is the Euclidean distance between the song's attributes vector and the preferences vector. The blue songs are liked/added by at least one user. The dark red songs are vetoed. When hovering over the song nodes, the title and artist of the song are shown with a play button ⏵. This force-directed graph enables the user to quickly view how songs and profiles are related to each other.

3.1.4 Version 3

The last version (Figure 3.5) contains all features of the previous versions plus a chat component. This chat box enables users to communicate and thus facilitates remote collaboration in creating the group playlist. A more detailed description is given in the next paragraph.

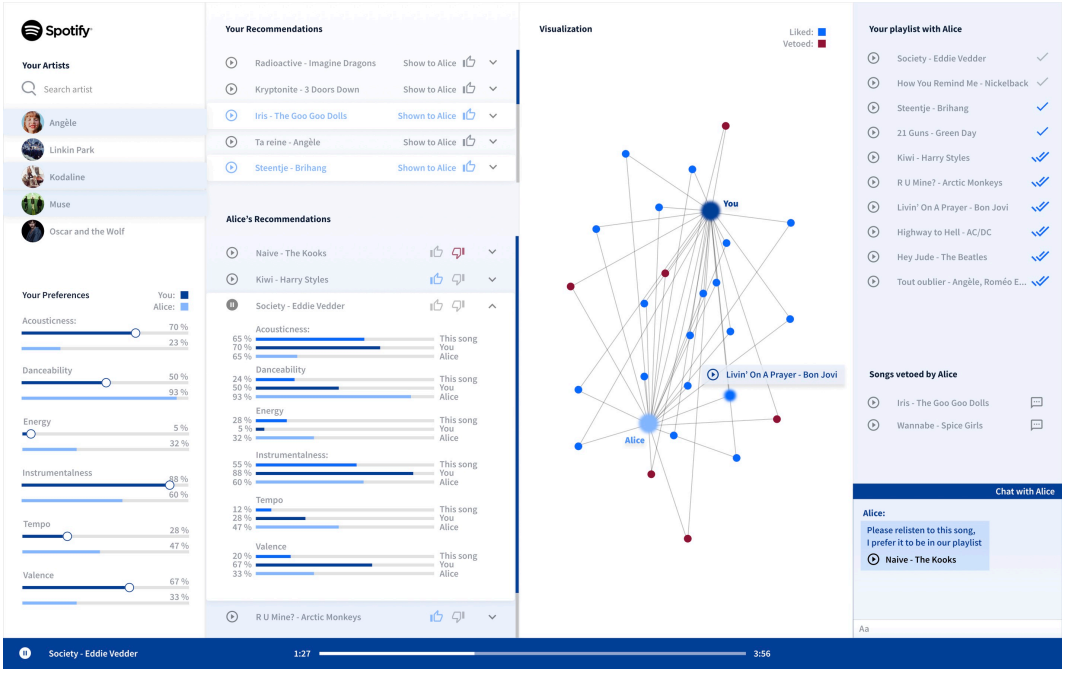


Figure 3.5: First prototype interface 3: chat box added

Chat box Users of the same group can communicate through a chat box. The chat box can be found in the lower right corner of Figure 3.5 below the 'Songs vetoed by Alice' component. This feature is added with the intention to discuss vetoed songs in particular. On the right of each song in the 'Songs vetoed by Alice' component, there is a speech bubble button. By clicking on this button, the user can send a message, with the song (and play button ⏵) attached, and tell the other person to reconsider the song. This can be the case when the user really wants this vetoed song to be in the playlist. As a result, users can try to convince each other to withdraw their veto, thereby adding the song to the playlist again.

3.1.5 Evaluation

The first prototype was one step towards the final design. It still has some major issues, which were addressed through feedback received from researchers in the Augment research group of the Computer Science department. The main conclusions drawn from their feedback are discussed in the following paragraphs.

Scalability The primary concern is scalability. The whole interface facilitates collaboration between only two users. Looking at version 3 (Figure 3.5), it gets challenging to alter the components 'Your preferences', 'Alice's recommendations', 'Visualization' (the force-directed graph), 'Your playlist with Alice' and 'Songs vetoed by Alice' when the group gets bigger. Adding an extra user implies extra bar charts in the preferences and in the detailed view of a song. Users would need to go through a lot more proposed songs, changing 'Alice's recommendations' to 'The group's recommendations'. The check marks in the playlist and the speech bubbles in the vetoed song list would get too confusing, not knowing who still needs to approve your song or who vetoed your song. Moreover, the complexity level of the force-directed graph would increase rapidly for every added group member.

Duo $\neq$ group Related to the scalability problem is the difference between group dynamics and duo dynamics. Two people do not behave in the same way as a group of people, meaning that the researched effectiveness of the interface to achieve consensus is also not scalable. When adding users to the group, the probability of each song being vetoed by at least one person increases. The bigger the group, the more difficult it gets to achieve consensus. Besides, you ask a lot of effort of the users when they need to go over each suggested song to approve or veto it. This gets even more tiring when there are more users proposing songs to you.

Complexity This interface demands strong cognitive skills. There is a lot of information that catches the eye of the user and not all may be self-explanatory. The bar charts and in particular the force-directed graph may be too complex for most users. The nodes and edges of the graph will probably need explanation before users understand their meaning. While some users can be fascinated, others may not look at it or even panic a little when they see the interface.

3.2 Second prototype

The objective of achieving consensus in remote collaborations was difficult to accomplish for groups of more than two users in the interfaces of prototype 1. Therefore, the second prototype does not focus on achieving consensus but on explanations and usability. The Spotify Web API does offer song recommendations based on seeds (artists, tracks or genres) but does not explain how these recommendations were generated. This section explains how this problem was approached and describes the new prototype as a solution to this problem. A small think-aloud study was conducted to evaluate the visualizations and will be discussed at the end of this section.

3.2.1 Objective

Again the question is asked how a recommender system can help users to easily create a good group playlist. One solution to the problem of Spotify not offering recommendation explanations, is developing a new recommendation system. A second option is to use the track attributes per song that the Spotify Web API does offer. The second solution was chosen because it is important to generate good recommendations when testing explanations. Spotify has the data (from users and songs) and fine-tuned algorithms to meet this requirement. Using Spotify's track attributes the songs can be ranked based on how likely the others will enjoy the song. The songs ranked highest will probably be more popular in the group than the low-ranked songs. In this way, the recommendation system can propose a playlist with only the top selected songs in it. The new interface should then help to investigate the usability of this approach and how beneficial explanations of the ranking algorithm can be in this context.

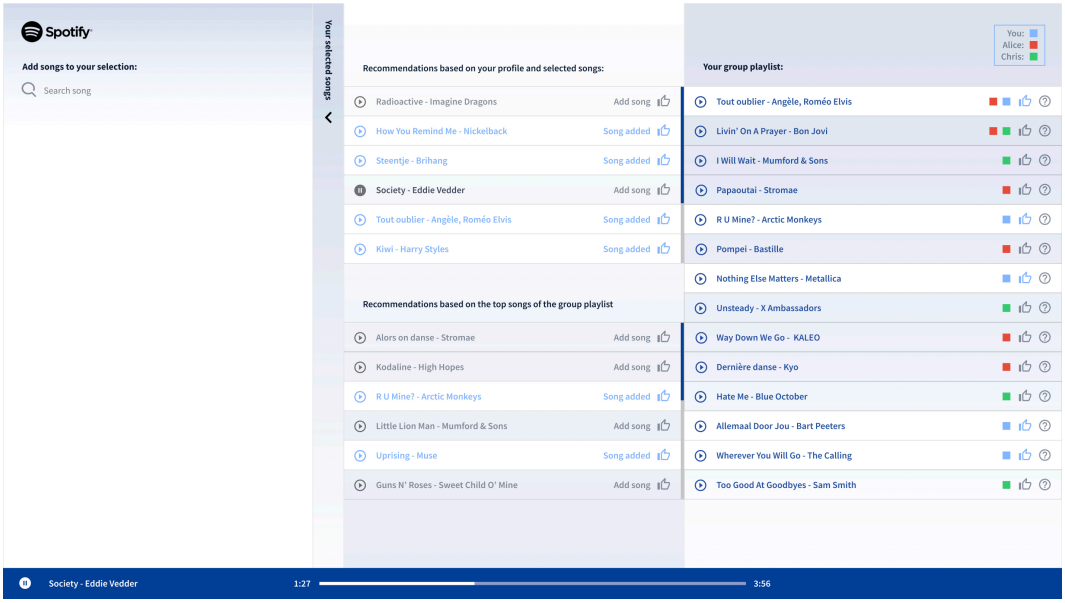
3.2.2 Components

This subsection describes the components of the second prototype from left to right in Figure 3.6a.

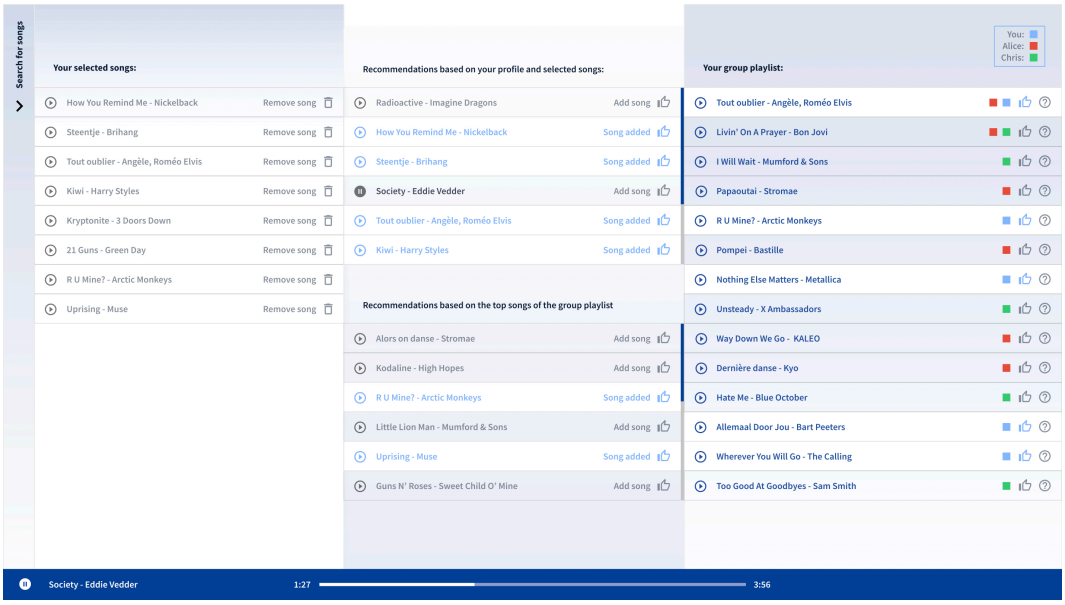
Search

The 'Your artists' component in the first prototype was modified to a search functionality. The primary objective of the system is to help users create group playlists. Discovering new songs is a secondary objective. Using artists as seeds for recommendations enables the user to discover new music, but the user is very limited if he/she can only input artists. Hence, the user can now search songs and the added songs are used as seeds to generate recommendations. Although the user is still able to explore new songs in the recommendations component, the focus has shifted from discovering songs to adding songs to a playlist.

Users can add songs to their selection by clicking on 'Search song' next to 🔍 and entering a title and/or artist of a song. The system will return a list of songs that match the given input. For each returned song, the title and artist is displayed



(a)



(b)

Figure 3.6: Second prototype interface: (a) Search component open; (b) Search component closed, selected songs view open

together with a play button and a button 'Add song '. The songs are shown similarly to the recommendations. This component will be explained in the next paragraph. The user can listen to a song by clicking on . This song will be displayed in the footer with its progress bar. By clicking on 'Add song ' next to a song, the song is added to the user's selection and the group playlist.

The user can see his/her selected songs when clicking on 'Your selected songs ' in the vertical bar between the search and recommendations component. The selected songs' window then slides open and replaces the search component (Figure 3.6b). Here, the user gets an overview of all his/her selected songs. Again, the title and artist of the song are displayed together with a play button. The user can remove a song by clicking on 'Remove song '. This will also remove the song from the group playlist. By clicking on ' Search for songs' in the vertical bar to the left of the selected songs view, the search component will appear again and close the selected songs view.

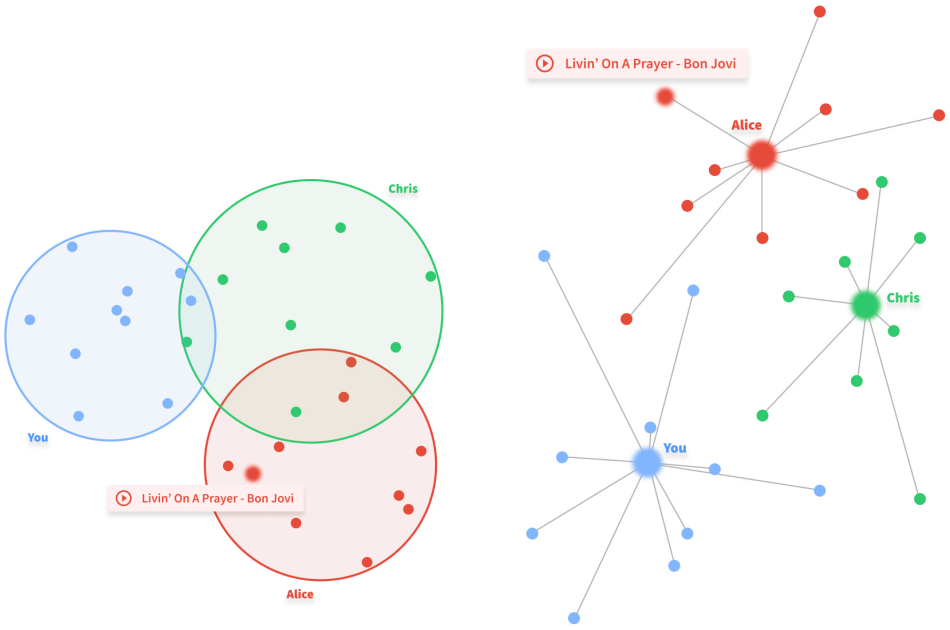
Recommendations

To use the recommender system developed for this research, the user needs to log in to Spotify. Based on the user's top artists and tracks listened in Spotify, recommendations can be generated. Every time a user adds songs to the group playlist, this song is used as an extra seed to generate recommendations. The list of 'Recommendations based on your profile and selected songs' (individual recommendations) is updated dynamically every time an extra song is selected. The 'Recommendations based on the top songs of the group playlist' are shown below the individual recommendations. The selected songs of all group members are ranked. The high-ranked songs have the highest probability that other group members will like the song. Consequently, these songs are given as a seed to generate group recommendations.

Both the individual and group recommendations can help users to find songs to add to the playlist. In addition, if the user adds a group recommendation, the chances are big that the other group members will like the song and as a result, that the song is ranked high. Recommended songs are represented by their title and artist. The user can listen to the song by clicking on and add the song to his/her selection and the group playlist by clicking on 'Add song '.

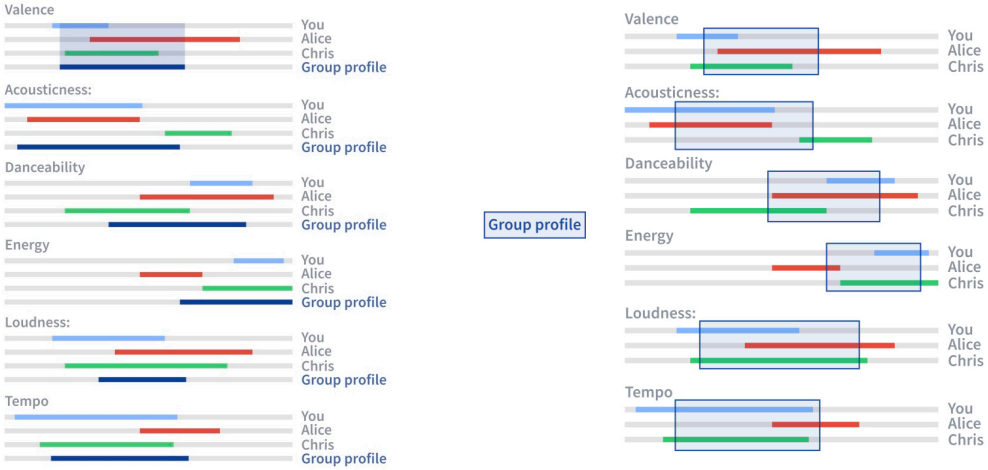
Profiles

One of the visual explanations appears when hovering over the group member's names in the upper right corner of Figure 3.6a. The visualization should give an overview of the individual profiles (and the group profile). In the think-aloud study, the participants were presented different visualizations and were asked which one they would prefer. The visualizations can be split up in 1) giving an overview of the individual profiles and selected songs (Figure 3.7) and 2) using the track attributes to compare the individual profiles and the group profile (Figure 3.8).



(a) (b)

Figure 3.7: Two ways of visualizing the individual profiles and selected songs



(a) (b)

Figure 3.8: Two ways of visualizing the individual profiles and the group profile in terms of the track attributes

In Figure 3.7a and 3.7b the distance between the songs represents the (dis)similarity between the track attributes. The bigger the distance between songs, the less they are similar. The colour of the node (song) indicates who liked/selected the song. When hovering over a node, the title and artist of the song are shown with a play button. Figure 3.7a uses circles to represent the individual profiles with the selected songs in it. Figure 3.7b uses a bigger node to represent the individual profiles and connects the selected songs with an edge in between. The individual profiles can be calculated based on the track attributes of the user's selected songs (taking the mean or median of each track attribute value). These visualizations help users to discover how songs are related to each other and see how this in turn relates to the ranking algorithm (isolated nodes/songs are ranked lower).

The other option for providing an overview is shown in Figure 3.8. For each track attribute, the range of values of the selected songs are shown per group member. Visualizing the values follows the principle of a box plot: it ranges from the lower to the upper quartile. The group profile is an aggregation of these individual profiles. When hovering over the range of values of the group profile in Figure 3.8a, it shows a projection to the individual profiles (this is shown for the valence attribute). In this type of visualization (Figure 3.8) the user can observe these different profiles and how they are related to each other.

Ranked playlist

Every time a user clicks on 'Add song 👍', the corresponding song is added to the group playlist (on the right in Figure 3.6a). The coloured squares next to a song indicate who selected/liked this song. When the background is white and the 👍 is coloured light blue, the user selected this song. As has already been explained, the ranking of songs is based on the similarity of track attributes. This measure is inferior to the likes of users. The similarity of track attributes gives the probability that a user will like the song. If a user explicitly likes a song, it dominates the similarity calculation. In other words, a song with two 'votes' is ranked higher than a song with only one 'vote' but higher similarity to the group profile.

A user can ask for an explanation to understand the position of the song in the ranked playlist by clicking on the question mark (?) next to a song. Three options for visualizing how this song relates to the range of track attribute values of selected songs are shown in Figure 3.9. Figure 3.9a shows the individual profiles compared to the song. Figure 3.9c shows both the individual profiles and the group profile compared to the song and Figure 3.9b compares only the group profile with the group. Obviously, the more information that is presented, the more complex the visualization is.

3.2.3 Evaluation

Six participants evaluated the visualizations and the usability through a think-aloud study. In this study method participants are asked to verbalize their thoughts. The sample of users need not be large as this method offers a wealth of information [45]. To get to know their preferences regarding the visualizations 'active intervention' was used to actively probe and guide the participant [46]. Questions like 'What do you expect to happen when you click on the question mark?' and 'Do you think there is a way to get an overview of the group members' profiles?' were used. The participants were also asked which visualization they preferred. Before the study, the objective of the system was explained to them and they were aware of the fact that the playlist would be ranked. Overall the system was quite clear to all six participants. The next two paragraphs analyse their feedback related to the overview of the profiles and the explanation of the position of a song in the ranked playlist.

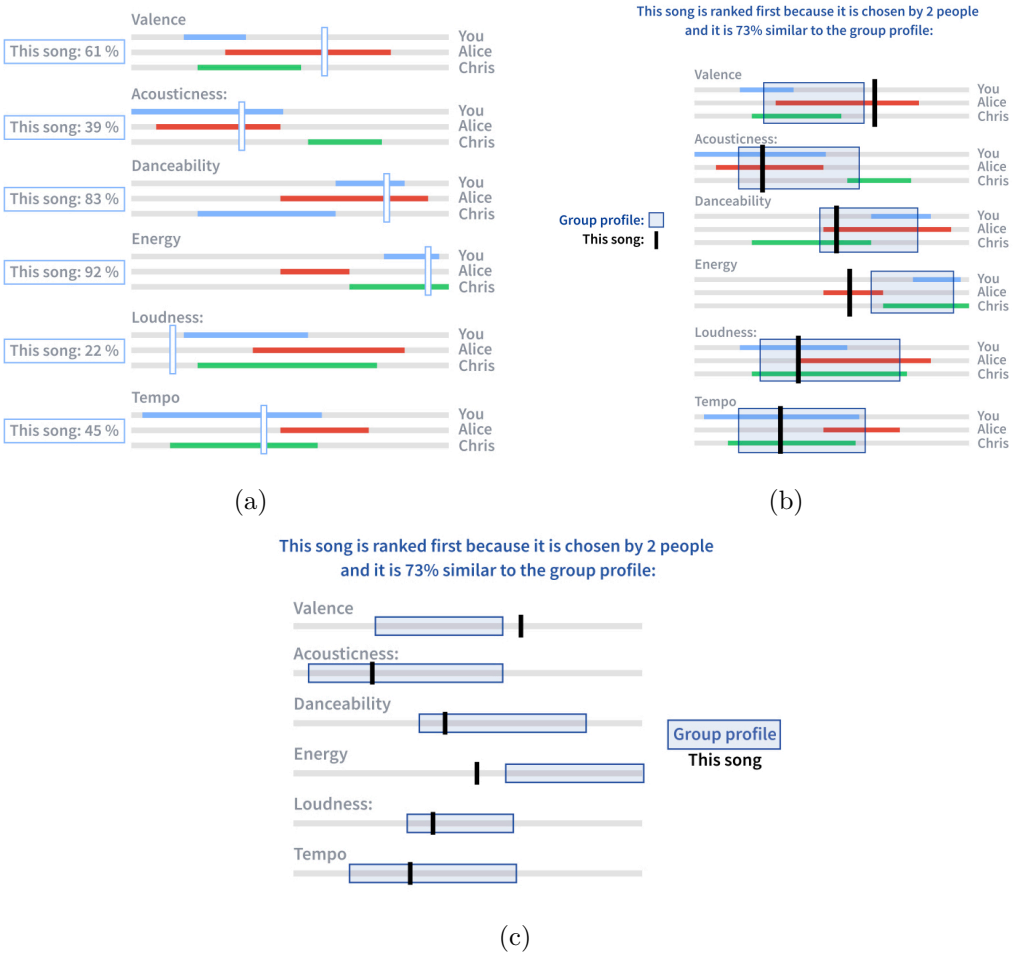


Figure 3.9: Three visual explanations for the position of a song in the ranked playlist

Profiles Initially, none of the participants would click on the question mark or the group members' names. Three participants thought they could perhaps see an overview of the group member's profiles when clicking/hovering over the group member's names. When they were presented Figure 3.7 and 3.8, all six of them preferred the visualizations in Figure 3.7 over the visualizations in Figure 3.8 because the latter was too complex. The preference for 3.7a or 3.7b was equally distributed.

Ranked playlist No participant guessed that an explanation of the position of a song in the ranked playlist would appear when clicking on the question mark. When the visualizations of Figure 3.9 were shown to them, four out of six said that Figure 3.9a and 3.9c were rather complex. However, five participants thought the information about the track attributes is interesting when you understand what is being displayed. One of those five suggested showing Figure 3.9b as default when clicking on the question mark and providing Figure 3.9a and 3.9c as an option (e.g. a toggle button).

Even though five out of six participants found the visualizations interesting, all six of them would mainly concentrate on composing the playlist. Maybe they would look at the explanations once, but after all they prefer a simple and clear interface. This feedback, like the first prototype's feedback, caused a shift of focus for this research.

3.3 Final design

Compared to individual recommender systems, there are extra factors that should be taken into account when developing a group recommender system. Two possible factors were already mentioned in this chapter. First, the focus was to research the facilitation of achieving consensus in group music recommender systems. Nonetheless, the results of this research using the first prototype interface (section 3.1) would be difficult to scale. Second, explanations seemed to be an interesting research topic as explanations for individual users differ from group explanations. However, a small think-aloud study demonstrated that users would not extensively use these explanations in composing a group playlist. They rather like to keep it simple. This leads to a third element in group recommender systems that is perhaps even more relevant in creating playlists and thus became the main subject of this research: fairness. This section discusses the new objective of the final design and the changes made to the design. Its implementation is explained in the next chapter.

3.3.1 Objective

One could argue that a good playlist is a fair playlist, where everyone's preferences are equally taken into account. When ranking the playlist based on popularity and only taking the top selected songs for the final playlist, the level of perceived fairness should definitely be examined. The question is not only if users appreciate this

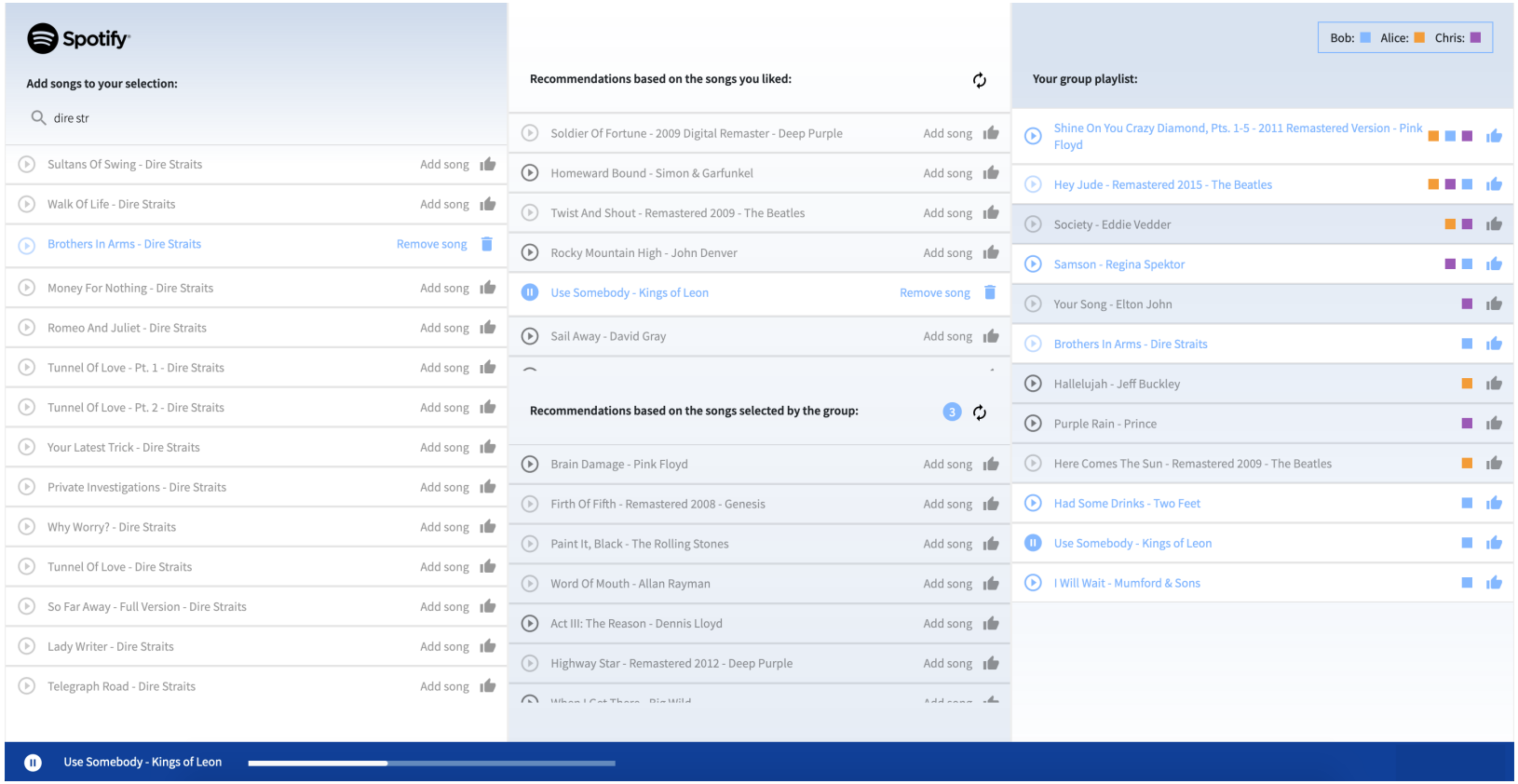


Figure 3.10: Final design of the interface

method of creating a playlist with only the most popular songs in it, but also if some users would feel disadvantaged when their selected songs did not make it to this playlist. Therefore, the final interface (Figure 3.10) is designed to investigate usability and perceived fairness in a group music recommender system.

3.3.2 Altered components

The biggest change compared to the second prototype is the omission of explanations. As has been discussed, most users won't pay attention to explanations and instead just focus on creating a playlist. The other essential changes are described below. The next chapter explains the functionalities of the system in more detail.

Selected songs In the previous prototype the user could view his/her selected songs by opening the 'Your selected songs' window (Figure 3.6b). As none of the participants showed interest in this view, this component is rather redundant. After all, users can see their selected songs in the playlist. As a result, the 'Song added 👍' button is altered to 'Remove song 🗑' to clarify the functionality to remove the song from the playlist.

Recommendations For each song that is added, recommendations are generated. Different to the previous prototype is that not the top songs are used as seeds for group recommendations, but the last selected songs of each group member. In this way, the group recommendations are a bit more 'fair' and are kept alive. Using only the top songs, it is possible that a user's preference is not included as a seed when his/her songs are not highly ranked. In addition, the top selected songs could be quite static. A new selected song is not necessarily highly ranked right away, leaving the top songs unchanged. When adding/updating recommendations each time a song is selected, the user receives a wealth of new music to discover and a disadvantaged user could easily introduce his/her music style.

3.4 Conclusion

This chapter discussed the iterations needed to get to the final interface design. The objective of this research needed to be redefined a few times. The received feedback of HCI (Human Computer Interaction) experts and the participants of the think-aloud study was invaluable to ameliorate the design and focus of the interface. As shown in Figure 3.1, the goal of this research evolved from achieving consensus to examining the use of explanations in a group setting to the perception of fairness in a group music recommender system. The next steps in this master's thesis profited from this gradually increased level of research relevance.

Chapter 4

Implementation

As explained in Chapter 3, the objective and final design for this research were defined after three iterations. This chapter discusses the implementation steps taken to develop this group music recommender system. Thereafter, the system was used to investigate its usability and the perception of fairness. The Spotify Web API[1] and the open-source software stack MEAN were used to build this web application[2]. The main implementation challenges included the ranking algorithm, the recommendations and enabling synchronous collaboration.

4.1 Spotify Web API

To properly test usability, the system should attempt to not be limited by the number of available songs and the quality of the recommendations. Spotify has the data needed to provide a wide variety of songs and good recommendations. It does have some limitations which will be addressed in the next subsection. Nevertheless, the advantages outweigh the limitations. For this reason, using Spotify was chosen over developing our own music recommender system.

4.1.1 Limitations

Play tracks In a web application that uses the Spotify Web API, users can be redirected to the Spotify Web Player[3] to listen to a song. Nevertheless, for most songs a 30 second preview is offered. If the preview is provided, users can listen to it in the application itself. The Web Playback SDK[4] allows to play a song from Spotify in the web application, but it is currently in Beta and only accessible with a valid Spotify Premium (paying) subscription. Moreover, limited time did not favor the implementation of this feature in the web application for this research. Therefore, the option was chosen to redirect the user to the Spotify Web Player when a 30

[1] developer.spotify.com/documentation/web-api
[2] github.com/ElisaLecluse/Group-Playlist-Recommender
[3] open.spotify.com
[4] developer.spotify.com/documentation/web-playback-sdk

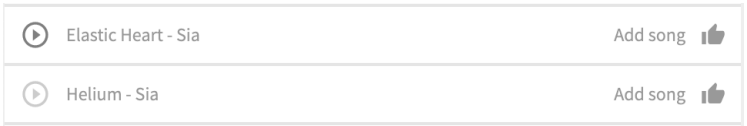


Figure 4.1: The availability of a 30 second preview of a track: the first track can be listened in the app (dark grey play button), for the second track the user will need to be redirected to the Spotify Web Player to listen to the song (light grey play button)

second preview of the song is not available. The color of the play button reveals the availability of the song. If it is dark grey, the application plays the 30 second preview. When clicking on the light grey play button, the user is redirected to the Spotify Web Player. Figure 4.1 shows the difference.

Recommendation seeds To generate recommendations artists, tracks and genres can be given as a seed. However, the input is limited to a combination of maximum five seed values. In other words, only a limited number of songs selected by the user can be given as an input to the Spotify Web API to request recommendations. This influences decisions made regarding the dynamically updated list of individual and group recommendations in the web application.

4.1.2 Advantages

These limitations decrease the user-friendliness of the system, but they are small compared to the advantages the Spotify Web API offers. Spotify is the most popular on-demand music streaming service [47], offers over 50 million tracks and can use data of its 286 million monthly active users to generate good recommendations [48]. Collaborative filtering, machine learning, digital signal processing (DSP) and natural language processing (NLP) techniques are applied to the tremendous amounts of user-item activity data collected every day [49]. They thereby not only provide a wealth of songs, but also promise to generate recommendations of high quality. Moreover, the API is clearly documented, easy to use and free. They even provide some example code snippets to get started.

4.2 MEAN stack application

The acronym MEAN stands for MongoDB, Angular, Express.js and Node.js, a free and open-source technology stack that uses JavaScript end-to-end (Angular is built in TypeScript, which is a superset of JavaScript). The architecture of a MEAN application is shown in Figure 4.2. MongoDB is the database system, Express serves as the backend web framework, Angular is used for the frontend development and Node.js handles the backend runtime environment. The development of a MEAN stack application is fast and simple as it is written in the same language from client-side to server-side. Additionally, JavaScript is a popular, dynamic and easy to use programming language. Another advantage is the transfer of data in JSON

format across the layers of the MEAN stack, bypassing the use of libraries to convert the data during client-server interactions [50]. This is also a major advantage when working with external APIs. The Spotify Web API endpoints namely return all response data as a JSON object. Furthermore, MEAN has the ability to manage concurrent users [51], which is necessary in a group recommender system.

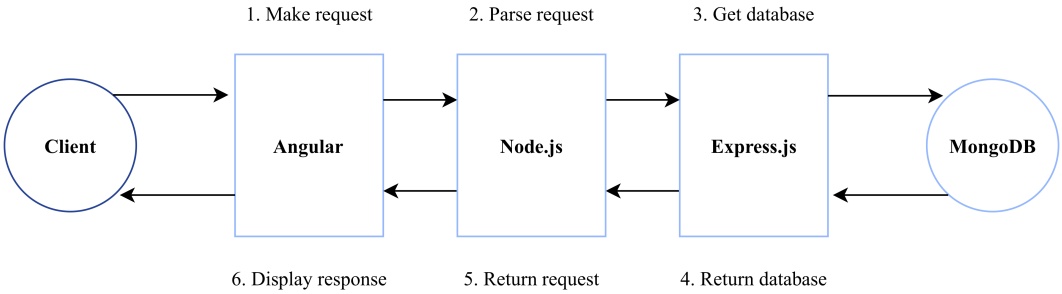


Figure 4.2: MEAN stack architecture (based on [52])

MongoDB MongoDB is a NoSQL database that stores the application's data as JSON-like documents. There is no need to translate objects in code to a relational structure through complex object-relational mapping [53]. The principal asset of MongoDB is its scalability in both storage and performance. The entire table does not need be reloaded when adding fields to the database, and MongoDB is able to manage large volumes of data without diminishing accessibility [51].

Express.js Express is a Node.js web application framework that balances ease of use and a robust set of features for web (and mobile) applications [51]. As backend web framework, Express handles all the frontend requests and easily interacts with the database to guarantee a smooth transfer of data to the end user. The consistent use of JavaScript continues in the backend as Express is designed as a package to work on top of Node.js. In addition, the example code snippets in the Spotify Web API documentation also use Node.js and Express, giving an extra argument to use MEAN for this research.

Angular Angular is a frontend framework ideal for single-page applications (SPAs) in which the browser does not need to load entire new pages. In contrast to AngularJS which primary language is JavaScript, Angular is built in TypeScript, a superset of JavaScript. TypeScript provides static typing, thereby improving performance and avoiding many runtime pitfalls. Due to a superior data binding algorithm and lazy loading (only loading the needed components), Angular applications are faster than AngularJS applications [54]. Moreover, Angular components are reusable as they are quite independent and self-sufficient. In our web application the recommendations component could for example both be used for the individual and group recommendations.

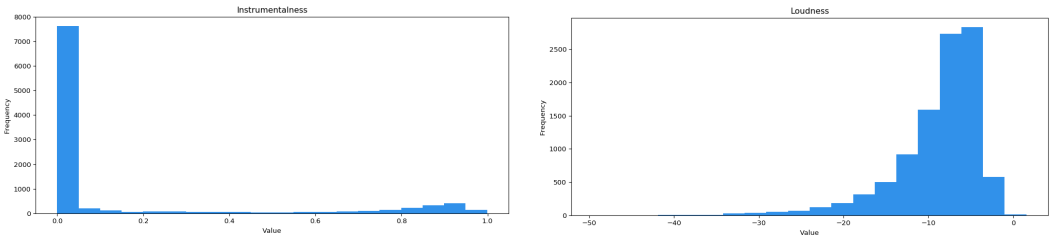
Node.js The backbone of the MEAN stack is Node.js. This server-side JavaScript runtime environment processes multiple connections simultaneously using asynchronous events. It responds quickly to requests from the Angular frontend and has a web server integrated to easily deploy the application with the MongoDB database [51]. As Node.js is non-blocking, it can handle a large number of concurrent incoming requests efficiently. Besides, using web sockets, Node.js can communicate with the client without needing to get requests from the client [50]. As synchronous collaboration (Section 4.5) is needed for this research, Node.js is an ideal choice.

4.3 Ranking algorithm

In this research the influence of the ranking algorithm on perceived fairness is investigated. Therefore, two versions of the system are tested, each version integrating another ranking algorithm. The first algorithm (version 1) ranks the songs based on the time they were added. The most recently added song is appended to the bottom of the list, ranked lowest. The second algorithm (version 2) uses the dissimilarity between the songs and the users' profiles to rank them. To determine this dissimilarity, Spotify's track attributes are used. The selection of track attributes and the algorithm itself are further explained below. Approval voting is integrated in both algorithms as it allows a much greater social engagement with the music [24, 55]. Users can like each others songs in the playlist. Each like counts as a vote. If one user likes the song of another group member, the song has two votes: the user who selected the song obviously likes the song and the other user gave his/her approval vote. These votes indicate 100% probability that these group members like the song. As the ranking algorithm should try to predict the probability that the group will like the songs, both ranking algorithms first rank the songs based on the number of votes. The number of votes is thus superior to the other metric (time and dissimilarity).

4.3.1 Track attributes

Spotify provides track attributes for each song. Section 3.1.2 discussed already the selection of track attributes based on the work of Millecamp et al. [43]. These included acousticness, danceability, energy, instrumentalness, tempo and valence. Unfortunately the distribution of track attribute values suggests that almost all songs have instrumentalness value 0.0 (Figure 4.3a). Hence, this feature would not effectively contribute to the ranking algorithm. Another feasible track attribute is loudness. Figure 4.3b shows that loudness does not have an ideal distribution but it is clearly better than instrumentalness. Loudness is thus better suited for the ranking algorithm. To extract the better part of the loudness and tempo distributions, values lower or higher than the ranges shown in Table 4.1 are set to their minimum or maximum respectively. All track attribute values are converted to fit in a range between 0.0 and 1.0 to facilitate Euclidean distance calculations.



(a) (b)

Figure 4.3: Distribution of values for track attributes (a) instrumentalness and (b) loudness [42]

Acousticness	**Danceability**	**Energy**	**Loudness (dB)**	**Tempo (BPM)**	**Valence**
$[0.0, 1.0]$	$[0.0, 1.0]$	$[0.0, 1.0]$	$[-35.0, 0.0]$	$[40.0, 220.0]$	$[0.0, 1.0]$

Table 4.1: Track attribute ranges

4.3.2 Dissimilarity-based ranking

Since vectors of track attributes are being compared Euclidean distance is taken as a dissimilarity measure. Cosine similarity does not take magnitude into account which makes it inferior to Euclidean distance. If the track attribute values of one song are twice the values of another song, cosine similarity would incorrectly state that the songs are exactly the same. This is not the case for Euclidean distance.

The smaller the calculated Euclidean distance between two songs, the larger their similarity. This allows to compare a new selected song with the songs the group members already selected. How a new song is compared to the existing playlist is discussed in the next paragraphs.

Individual profiles vs group profile One of the primary questions in group recommender systems is whether to aggregate individually generated recommendations or to construct a group preference model (Section 2.1.1). The situation here is slightly different as the system ranks items already selected/liked by the group members instead of ranking candidate items that are new to the group members. Nevertheless, a similar decision needs to be made: is a selected song compared to a group profile representing all selected songs or to individual profiles each representing the selected songs of one group member? To equally treat the preferences of each group member, using individual profiles would be the best choice. If one group member selected five songs and another group member selected ten songs, the preferences of the latter would dominate twice as much if a group profile is constructed from all selected songs. On the contrary, if the result of the comparison using individual profiles is aggregated, it doesn't matter how many songs one selected.

Profile construction There are three options on how to compare the new song to the individual profiles. One option is to calculate the Euclidean distance between every selected song per group member and take the average distance to represent the dissimilarity between the new song and the individual profile. This is the most accurate option but does not scale well. In this case, for every new song, the Euclidean distance needs to be calculated N times (N = number of already selected songs in total). The second option is to calculate the mean for every track attribute of the songs a group member selected. The profile of the group member who selected the song needs to be updated for each new song, but the Euclidean distance only needs to be calculated between the song and the profiles (N = number of group members). Calculating the mean for each track attributes has the drawback that two extremes are smoothed to a result that probably does not present the preference of the user at all. Therefore, a third option was chosen: a group member's profile is represented by the medians of the selected songs for each track attribute.

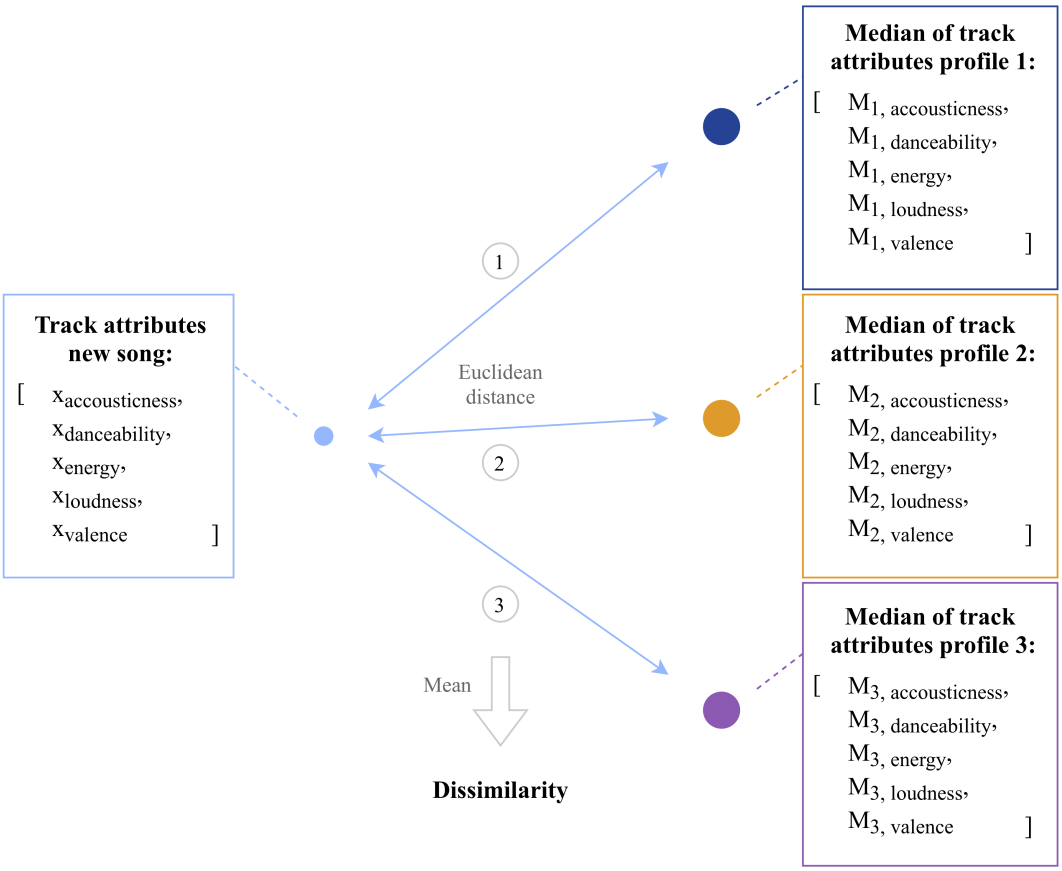


Figure 4.4: Calculation of the dissimilarity between a new selected song and the group members' profiles for a group of three users

Dissimilarity calculation Building upon the decisions made in the previous paragraphs, the dissimilarity between a new song and the individual profiles can now be calculated. Figure 4.4 gives an overview of this process. When a new song is selected, the Euclidean distance is calculated between the track attributes of the new song and the median track attributes of each individual profile. At this stage, the individual profile of the group member who selected the song is already updated with this new selected song. The mean of the Euclidean distances (①, ② and ③ for a group of three users) represents the dissimilarity between the new song and the group members' profiles. For each selected song the dissimilarities are now updated in order to rank the playlist. These updated dissimilarities are determined in the same way as the dissimilarity of the new song. The songs with the lowest calculated dissimilarity are ranked first. These songs should be the most popular among the group members as they are 'closest' to the individual profiles. This ranking technique is similar to content-based filtering in combination with aggregated predictions (Section 2.1.1). Because of the information available per item, content-based recommendation is the best choice out of the three state-of-the-art methods (collaborative, content-based and knowledge-based filtering). The individual 'content' preferences are represented by the median of each track attribute and the average is taken as aggregation function. Euclidean distance dissimilarities are used as a metric for user-item similarities.

4.4 Recommendations

Recommendations are generated by Spotify. The number of seeds is limited to a combination of five tracks, artists or genres. Here, only tracks (the ones selected by the users) are used as input for recommendations. Depending on the component (individual or group recommendations; Figure 3.10) decisions were taken regarding the number of seeds and recommendations. An overview is given in Table 4.2.

4.4.1 Individual recommendations

Each time a user adds a song, three recommendations based on this song are added to the top of the list of individual recommendations. Less than three recommendations would limit the user in discovering new songs. Providing more than three songs for every selected song could overwhelm the user as the length of the recommendations' list increases rapidly. The 🔄 button offers the possibility to replace the list of recommendations by ten new recommendations. Ideally all songs selected by the user are given as a seed, but in this case the five most recently selected songs are given as an input to Spotify. The user can use this button if he/she is not satisfied with the current list of recommendations. Ten recommended songs nicely fill up the space reserved for the individual recommendations (Figure 3.10). The user is required to scroll down to see the last three updated recommendations, so giving more than ten recommendations would not be effective. After all, more recommendations are added to this refreshed list when a new song is selected for the playlist.

4.4.2 Group recommendations

Each time a group member adds a song, three recommendations are added to top of the group recommendations' list. Group recommendations are based on the most recently selected song of each group member. In a group of three users three seeds will thus be given. When a user adds a song the list of most recently selected songs is updated by replacing the previously selected song of this user with the new song. The updated list is then given as a seed to generate three recommendations for every group member. The functionality of refreshing the recommendations' list is implemented in both the individual and group recommendations. Again, each group member's most recently liked song is given as a seed to generate new recommendations to replace the old list. Thus, when the user clicks on 🔄, ten new recommendations are offered, similar to the individual recommendations' component.

	Action	Input	Output
Individual recommendations	song selected	selected song	3 recommendations
	🔄	5 last selected songs	10 recommendations
Group recommendations	song selected	each group member's last selected song	3 recommendations
	🔄		10 recommendations

Table 4.2: Overview of the seeds (input) and the number of generated recommendations

4.5 Synchronous collaboration

Several studies state that by supporting group awareness in multi-user systems, the usability significantly improves [56, 57, 58]. Groupware users should thus receive real-time updates of the actions of others to create awareness of each other. Projects like SearchTogether (remote) [59] and CATS (interactive tabletop) [16] demonstrated the effectiveness of synchronous collaboration. A good software design for synchronous collaboration notifies each group member of the contributions to the process at any time, allows the user to identify the work done by each contributor and supports reverting changes while maintaining consistency [60]. This section explains how this awareness was integrated in the system.

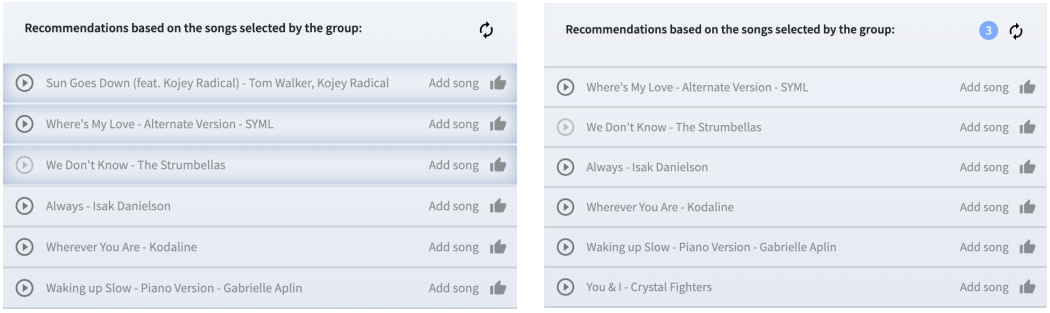
4.5.1 Socket.IO

Socket.IO is a real-time engine for Node.js. This library enables bidirectional and event-based communication between client and server [61]. In traditional web applications, the client sends HTTP requests to the server and the server responds. In our real-time multi-user system, other users need to be updated each time a song is selected. If one client sends its selected song to the server, the server is not able to broadcast this change to the other clients purely by using HTTP requests. In

other words, the server cannot communicate with the client without request. Using Socket.IO, both server and client set up a connection to allow communication in both directions. In this way, when a user likes a song, the other group members can see the updated playlist immediately.

4.5.2 Notifying the user

The playlist gets updated real-time when songs are selected or liked by the group. All users will see the songs that just got selected appearing in the playlist. A user can easily dislike/remove a song by clicking on 'Remove song' in the search or recommendations component or by clicking on next to the song he/she liked in the playlist (Figure 3.10). Again, the playlist is updated for every group member. As has been discussed in Section 4.4, the most recently selected song of each group member is used as a seed to generate group recommendations. Every time a user selects a song this list of seeds is updated to generate new group recommendations. All group members see these new recommendations appearing as shown in Figure 4.5a. They are highlighted for five seconds. If the user scrolls down in the list of recommendations, the content does not jump when new recommendations are added. Instead, a blue circle with the number of new recommendations is displayed (Figure 4.5b). If the user scrolls up, the new songs are highlighted when they become visible one by one. The blue circle, displaying the recommendations not yet seen, counts down for every new song that becomes visible by scrolling.



(a) (b)

Figure 4.5: New recommendations notifications

4.6 Conclusion

This chapter explained how the final design presented in Chapter 3 was implemented. Spotify is chosen to get tracks and recommendations from because its API is free and clearly documented and because it is one of the largest on-line music streaming services. The web application is developed using the MEAN stack technologies (MongoDB, Express.js, Angular and Node.js). The rest of the chapter discussed the implementation of the ranking algorithm, the recommendations and real-time features using Socket.IO.

Chapter 5

User Study

The web application developed for this research was used to investigate the perception of fairness in group music recommender systems. Chapter 3 discussed the iterations needed to design the interface and Chapter 4 discussed its implementation. This chapter describes the conducted user study. First, the method and procedure of the online user study are explained. Thereafter, the results are presented. The next chapter discusses the results in more detail.

5.1 Method

A within-subjects design was chosen for this user study. To compare the time-based and dissimilarity-based ranking algorithms (Section 4.3), the participants tested the two versions of the system. In total, 45 test users were recruited. Their demographic information is presented in the next subsection. Each group consisted of three participants, forming fifteen groups altogether. To avoid learning effects in within-subject experiments, an interactive tutorial taught the participants all the functionalities of the system (Section 5.2.2). Additionally, the counterbalancing approach was used. Eight groups first tested the time-based version. The other seven groups first tested the dissimilarity-based version. The participants were asked to compose playlists suitable for a certain scenario. One task involved creating a playlist for a road trip. The other task's scenario was a dinner. Table 5.1 shows how many groups were assigned to which version and task first. The participants were only informed about the task, not the version. In this way, biased answers to the questionnaire and unnatural interactions with the system are avoided.

5.1.1 Demographic information

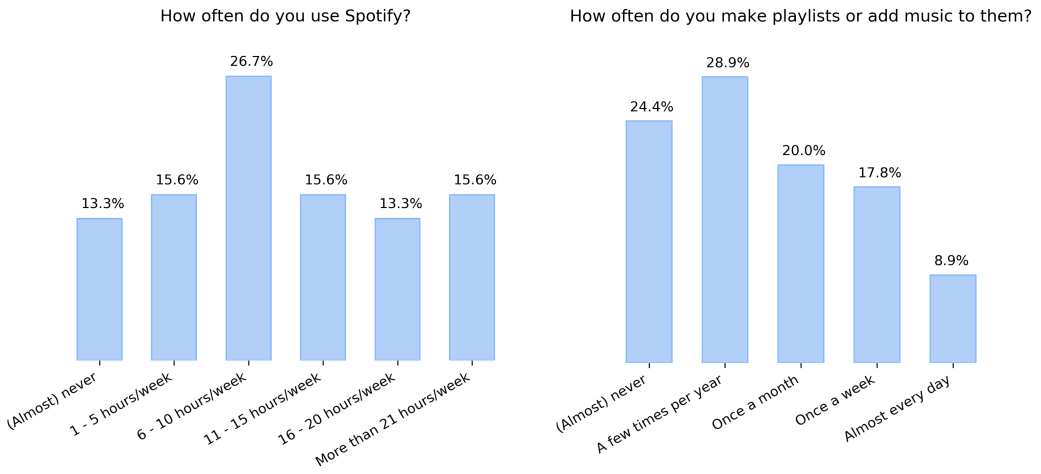
Before testing the system, the participants answered the questions of a demographic questionnaire. Table 5.2 gives an overview of the gender ratio and the ages of the 45 participants. Most of the participants were 18 to 24 years old (64.4%). Figure 5.1 shows how often the participants use Spotify and how often they make playlists or add music to them. 6 to 10 hours per week Spotify usage represents the biggest

	Time	Dissimilarity
Road trip	4	4
Dinner	3	4

Table 5.1: Version-task allocation: the number of groups that started with a certain version of the system and assigned task (three groups started with the time-based version in a dinner scenario and thereafter tested the dissimilarity-based version in a road trip scenario)

Participants	**45**
Groups	15
Female	60 %
Male	40 %
< 18	6.7 %
18 - 24	64.4 %
25 - 34	13.3 %
35 - 44	6.7 %
45 - 54	8.9 %

Table 5.2: Demographic information of the participants



(a) (b)

Figure 5.1: Spotify usage and managing playlists: distribution of the participants' answers

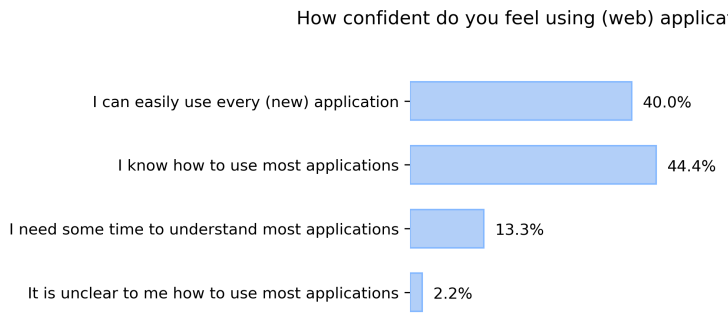


Figure 5.2: Application usage confidence: distribution of the participants' answers

group (26.7%). 75.6% manages playlists at least a few times per year. A big part of the participants feel quite confident using (web) applications (Figure 5.2): 44.4% knows how to use most applications and 40% can easily use every (new) application.

5.2 Procedure

The entire user study needed to be online conductible. Because of the measures taken in response to COVID-19, it was harder for the researcher to assist the participants through the experiment process. The participants should thus be able to test the system and answer the questionnaires without much help. This requires a smooth flow of all elements of the user study. As a precaution, a group chat in Messenger[1] was created to easily communicate and to handle any unexpected problems. The web pages designed for each stage of the user study flow (Figure 5.3) can be found in Appendix A.

As presented in Figure 5.3, the user is first introduced to the objective of the study: research how different personalities have an impact on the perception of fairness in group music recommender systems. The study procedure is briefly explained to them. Second, they are presented the informed consent form in English and Dutch. If they agree to participate, they are redirected to the login page. They need to enter a display name and the group name that was given to them beforehand. The group name was a random code with a prefix to inform the system which version and task to show them first. The group name is also used to filter the recommendations and group songs to ensure that different groups don't interfere. When this data is sent to the database, a random identifier is generated to identify the user's stored actions throughout the whole session. The user is also asked to log in to their Spotify account. A dummy account was provided to participants who did not have one. When the user grants the application permission to access their data, Spotify sends an access token and a refresh token. The access token is used to get for instance recommendations from Spotify. After one hour (3600 seconds) the access token expires and the refresh token is used to get a new access token.

[1] www.messenger.com

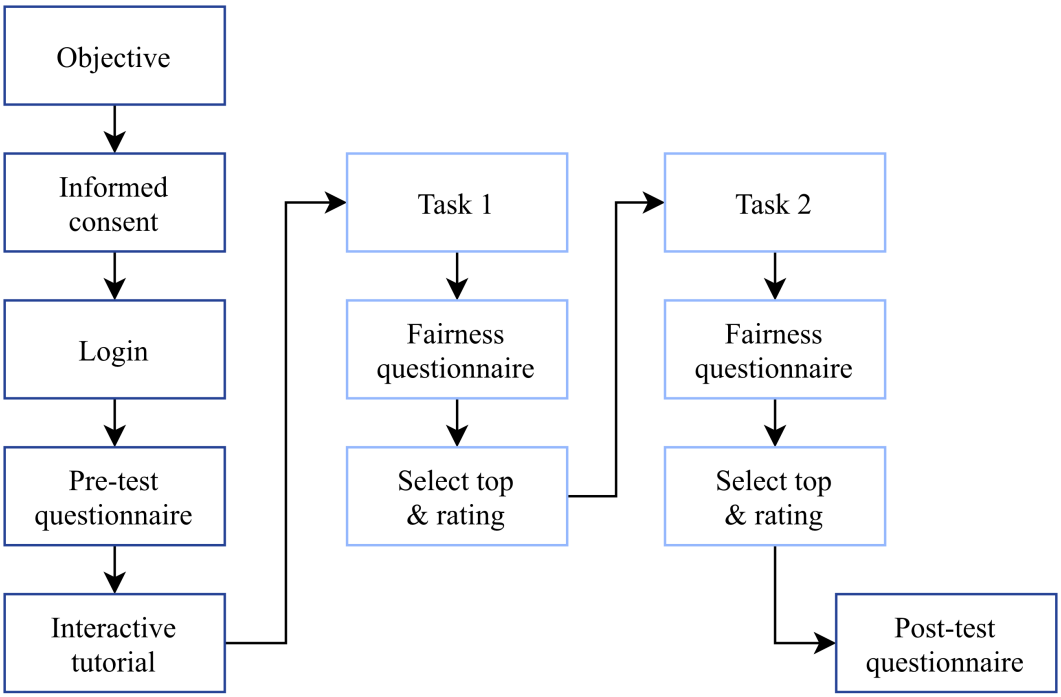


Figure 5.3: User study flow

When the participant successfully entered his/her display name and group name and a Spotify access token was received, the participant was asked to answer the questions in a pre-test questionnaire. The questionnaires and the interactive tutorial subsequent to the pre-test questionnaire are discussed in the next two subsections. Depending on the group name, the user is given one of the two scenarios to create a playlist for: a) road trip; b) dinner (Figure 5.4). Each group member should at least select/like five songs and, in total, at least fifteen songs should be selected by the group. When they click on 'Understood, I'll do my best!', they can use the web application to complete the task.

The participants can click on the 'Finished' button which was added in the footer of the interface (Figure 3.10), in the lower right corner, when they believe that they are finished. If the task appears not to be completed, a pop-up is shown: 'You are not finished yet'. Depending on the case, one of the following three messages (or a combination) are displayed:

- Each group member should like at least 5 songs, please like yours.
- In total, the group playlist should contain at least 15 songs.
- Please wait for X (and Y) to select their songs. Feel free to add more songs to the playlist while waiting.

X and Y represent the names of the group members who did not select five songs yet. When one of the group members clicks on 'Finished' when each of them liked at

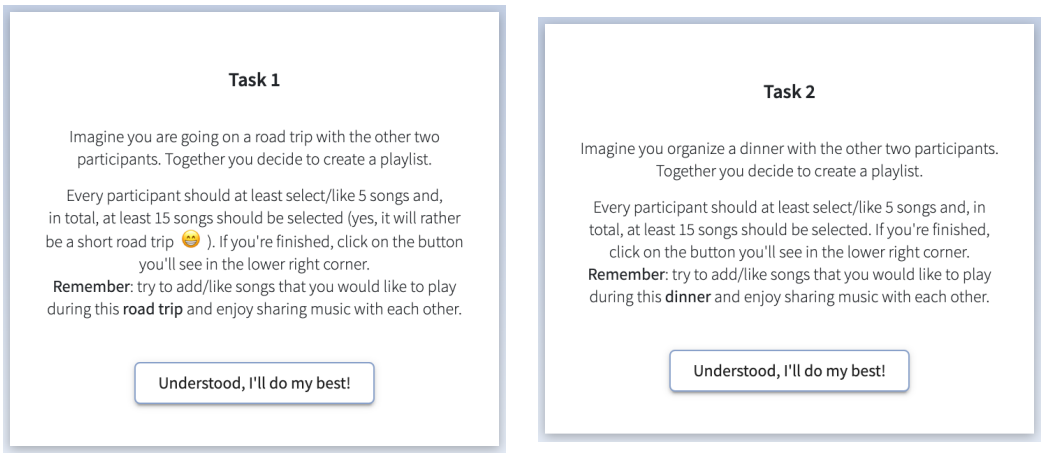


Figure 5.4: Tasks given to test the system (the order of the tasks depends on the given group name)

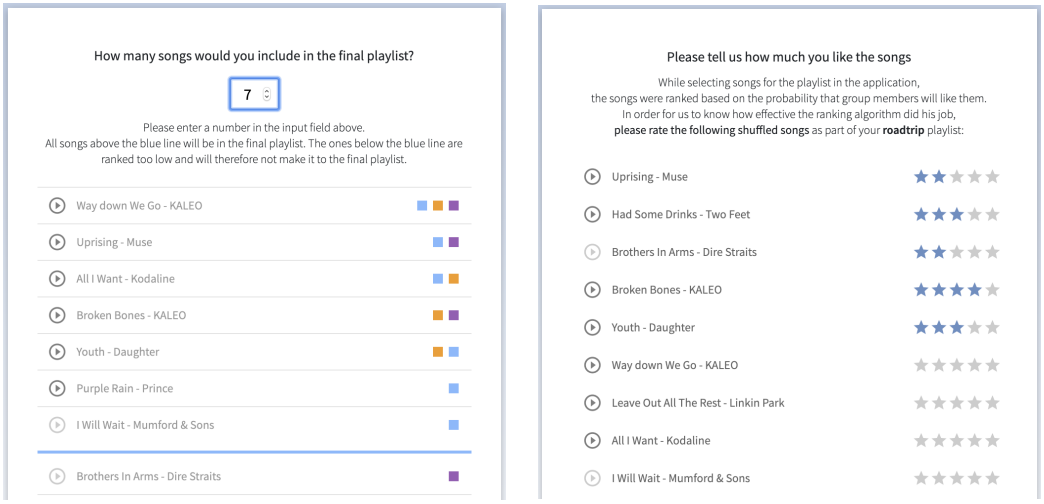
least five songs and, in total, at least fifteen songs were selected, all group members are immediately notified that the task is completed:

Thanks for completing the task!
Now that all users selected at least 5 songs and the group playlist has at least 15 songs in it, you can proceed to the next step.

This pop-up forces all group members to proceed to the next step together. If one user alters the playlist while the others left the web application, they will not answer questions about the same playlist. It is important to note that the 'Finished' button is only added for this user study. It is because of the experiment, to test the interface, that users receive a task to complete.

The next steps consist of answering fairness related questions (Section 5.2.1), entering the number of songs they would want to include in the final playlist and rating each selected song. The ranking algorithm's purpose is to easily select the most popular songs for the group playlist. The group members can decide how many songs they want in the final playlist (Figure 5.5a). To investigate the effectiveness of the ranking algorithm, the participants are also asked to rate each selected song (Figure 5.5b). The songs are listed in random order to avoid biased answers. They can only proceed to the next step if all songs were given at least one star.

Thereafter, the participants receive a second task and follow the same steps as for the first task. When finished, they were asked to answer the last set of questionnaires to evaluate the system as a whole. These post-test questionnaires are discussed in the next subsection. At the end, the participants are thanked and offered a link to the songs they selected together.



(a) (b)

Figure 5.5: The participant was asked to (a) enter the number of top songs to include in the final playlist and to (b) rate each selected song

5.2.1 Questionnaires

With the KU Leuven Qualtrics student account, the online survey software Qualtrics[2] could be used for free and facilitated managing the online questionnaires. The user ID generated at the login stage of the user study was embedded in the survey URL and stored together with the responses. Consequently, the participant's actions in the web application could be linked to the answers given in the Qualtrics questionnaires.

Pre-test questionnaire

The participants were asked to provide some demographic information like age, gender, Spotify usage, playlist management and (web) application usage confidence at the beginning of the experiment. In addition, some music-related 5-point Likert scale questions were asked (1 represents 'disagree strongly', 5 represents 'agree strongly')(Table 5.3). Two questions were extracted from the Goldsmiths Musical Sophistication Index (Gold-MSI) [38], a self-report inventory assessing musical skills and behaviours. The well-established and widely used 44-item Big Five Inventory (BFI) [31, 28] was chosen to assess the Big Five personality dimensions of the participants. Extraversion, Agreeableness, Conscientiousness, Neuroticism and Openness represent personality at the broadest level of abstraction. These five dimensions encapsulate a wide range of distinct, more specific personality characteristics [30]. In this way, this research tries to investigate how different personalities influence the perception of fairness in a group music recommender system. The BFI questionnaire can be found in Appendix C.

[2] www.qualtrics.com

M1.	I spend a lot of my free time doing music-related activities*
M2.	I keep track of new music that I come across*
M3.	I am open to new songs
M4.	I am open to genres that I usually don't listen to
M5.	It lasts a while before I start to like a new song
M6.	I regularly ask friends to recommend new music
M7.	I regularly listen to playlists of friends

Table 5.3: Music-related 5-point Likert scale questions in the pre-test questionnaire (*taken from Gold-MSI [38])

F1.	I feel happy with the number of songs I liked in the top X playlist
F2.	I think someone of the group will feel disadvantaged with the top X playlist
F3.	I think everyone of the group will like the top X group playlist
F4.	I like the top X group playlist
F5.	I feel like the ranking algorithm is fair (i.e. every group member has some songs they like in the top X playlist)
F6.	I feel like the songs are evenly ranked regarding who liked them (i.e. every group member has a song he/she liked highly ranked)
F7.	I would play the top X group playlist with the others of the group
F8.	I would play the top X group playlist even when the others of the group are not present
F9.	I would play the full group playlist with the others of the group
F10.	I would play the full group playlist even when the others of the group are not present
F11.	I think this is a good application to make a fair group playlist
F12.	Fairness is important in group playlists
F13.	It is more important that a majority of the group likes the playlist than that someone's songs are left out
F14.	I want at least some of my songs to be in the playlist, even if no one likes them

Table 5.4: Fairness-related 5-point Likert scale questions (X represents the number of top songs in the final playlist: 2/3 of all selected songs)

Fairness questionnaire

To study the perceived fairness of both the time-based version and the dissimilarity-based version of the system, a set of fourteen 5-point Likert scale questions were constructed (Table 5.4). The first ten questions depend on the playlist the participants just created together. This playlist is shown next to the questionnaire. The participants are told that one third of all selected songs would not make it to the final playlist. The other songs that were ranked higher form the final playlist. A blue line in the list of selected songs indicates which songs made it to the final playlists and which did not. Images showing how this is presented to the participants can be found in Appendix A. Two thirds was chosen to get the best out of the answers to the fairness-related questions. If almost all selected songs are included in the final playlist, fairness does not really matter, since all group members will have some of their songs in it. If too little songs are chosen for the final playlist, then why should the group put time in selecting songs? With regard to the size of the groups in this experiment (three participants), two thirds were chosen. This leaves the case open of not including any of the selected songs of one group member (worst case regarding fairness). The best case regarding fairness is where the group members' songs are evenly ranked. A ratio is preferred over a fixed number as the length of the selection of songs may vary (they may add more than fifteen songs).

Post-test questionnaire

In the last stage of the user study, the participants were asked to evaluate the system by answering questions of two well-established questionnaires. The first one is the System Usability Scale (SUS) [62]. SUS is a quick and easy tool to assess the usability of a system. It is free and provides a single score on an easily understood scale. Moreover, the work of Bangor et al. [63] enable comparing the score with 1180 other web-based interfaces and provide a rule-of-thumb standard for an acceptable SUS score.

Second, the evaluation framework ResQue (Recommender systems' Quality of user experience) [64] was used to measure the quality of the recommender system. The quality of the recommended items, the system's usefulness, the interface and interaction adequacy all contribute to overall user satisfaction with the system. The evaluation questionnaire consists of a set of constructs, thirty two 5-point Likert scale questions for each construct. The eleven most applicable questions were selected. Assessing purchase intention would for instance not be relevant for this research. These questions were also tailored to the use case of the system: words like 'items' were changed to 'songs'. Table 5.6 shows the questions with their constructs. These constructs in turn belong to higher-level constructs: perceived system qualities (R1 to R5), user beliefs (R6 to R9) and behavioral intentions (R10 to R11). Between these domains there exist significant causal effects: increasing perceived system qualities cause improved user beliefs which leads to the intention to use the system (frequently).

S1.	I think that I would like to use this system frequently
S2.	I found the system unnecessarily complex
S3.	I thought the system was easy to use
S4.	I think that I would need the support of a technical person to be able to use this system
S5.	I found that the various functions in this system were well integrated
S6.	I thought that there was too much inconsistency in this system
S7.	I would imagine that most people would learn to use this system very quickly
S8.	I found the system very cumbersome to use
S9.	I felt very confident using the system
S10.	I needed to learn a lot of things before I could get going with this system

Table 5.5: SUS questions [62] (5-point Likert scale)

	Recommendation Accuracy
R1.	This recommender system gave me good suggestions
	Recommendation Novelty
R2.	This recommender system helped me discover new songs
	Recommendation Diversity
R3.	The songs recommended to me are diverse
	Interaction Adequacy
R4.	This recommender system allows me to tell what I like
	Interface Adequacy
R5.	The components of this recommender system are clear to me
	Perceived Ease of Use
R6.	I found it easy to make this recommender system recommend new songs to me
R7.	I became familiar with this recommender system very quickly
	Perceived Usefulness
R8.	This recommender system helped me find good songs
R9.	This recommender system influenced my selection of songs
	Use Intentions
R10.	I will use this recommender system to create group playlists
R11.	I will use this recommender system frequently

Table 5.6: Slightly modified ResQue questions (5-point Likert scale) with their constructs [64]

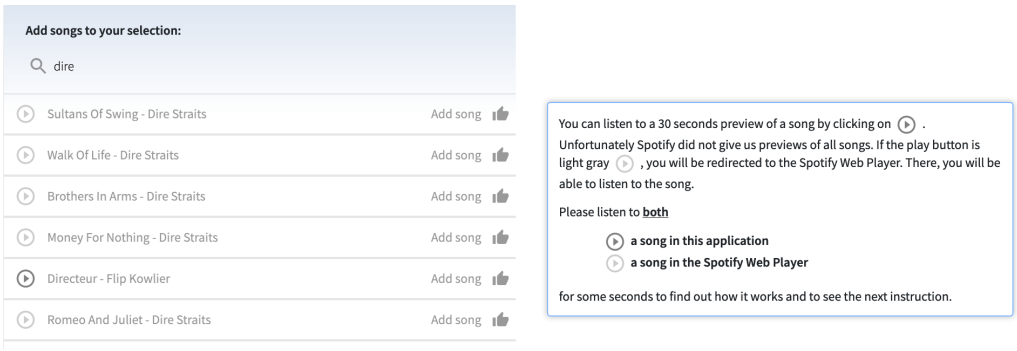


Figure 5.6: The biggest obstacle in the interactive tutorial: step 2

5.2.2 Interactive tutorial

To avoid learning effects in within-subject experiments, participants should know the basic functionalities of the system. Because one learns best by doing, an interactive tutorial was chosen over an explanatory video. Each step is shown in Appendix B and summarized below:

1. Search for a song
2. Listen to a song in the application and the Spotify Web Player
3. Select at least two songs
4. Remove one song from the playlist
5. Use the refresh button to replace the old list of recommendation by ten new ones
6. A GIF explains the new recommendations' notification circle (it displays the number of new recommendations not yet seen by the user)
7. The participant is informed about the dynamically ranked playlist and the possibility to select the top songs for a final playlist at the end

Step one to three only show the search component. The recommendations and playlist component are replaced by a white background to not overwhelm the participants and guide them step by step through the components. When they added at least two songs, the whole interface appears.

Limitations

During the experiment some issues of the interactive tutorial arose. However, these issues were quickly solved as logs were stored per user ID, showing boolean flags for each completed tutorial step. As a result, the researcher could see the progress of the participants and help when necessary. In six of the fifteen groups, one or more group members got stuck in the tutorial. The biggest obstruction occurred at step

two (Figure 5.6). In five groups, some group members started adding songs instead of listening to both a song in the application and a song in the Spotify Web Player. Furthermore, two participants thought the tutorial was over at step four. At this point, the rest of the interface appears (the recommendations and playlist component are not hided anymore). They ignored the tutorial message and started to create their playlist. Eventually every participant completed the tutorial, one sometimes experimenting a bit longer with the system than the others, which in itself is not that bad.

5.3 Results

This section lists the user study results. The distributions of the answers to the questionnaires are presented and briefly discussed. First, the pre-test questionnaire's answers are used to examine the participant's openness to music and their personality. Second, the perceived fairness is compared between the two versions of the system. One version used a basic time-based ranking algorithm, the other version used a dissimilarity-based ranking algorithm based on Euclidean distance. Third, the distribution of the answers to the question 'How many songs would you include in the final playlist', the ratings given to all songs selected by the group and the interaction logs are presented. Fourth, the ResQue and SUS scores are examined and, at the end, the participant's suggestions are discussed. The next chapter analyses these results in more detail.

Openness to music

Seven music-related questions, including two of the Gold-MSI [38], were used to assess the participant's openness to (new) music. The box plots in Figure 5.7 indicate that the participants are generally more open to new songs than to genres they don't listen to, showing that they are probably more open to new songs of genres they like. Overall, M1, M2, M3 and M4 show that participants are open to (new) music as relatively high scores were reported (means: 3.78, 3.60, 4.42, 3.76 respectively).

Personality

The 44-item BFI assesses the the Big Five personality dimensions. For every factor, two related facets can be determined [33]. The box plots in Figure 5.8 show the distribution of the traits among the participants. The Shapiro-Wilk test was applied to determine the normality of the data. All traits, except for Activity, Altruism, Compliance and Order, show a Gaussian distribution. In general, the participants score highest on Altruism, Activity and Ideas (means: 4.05, 3.97 and 3.86, respectively) and lowest on Depression (mean: 2.66). According to Table 2.2 the average participant is rather warm, energetic, curious and contented.

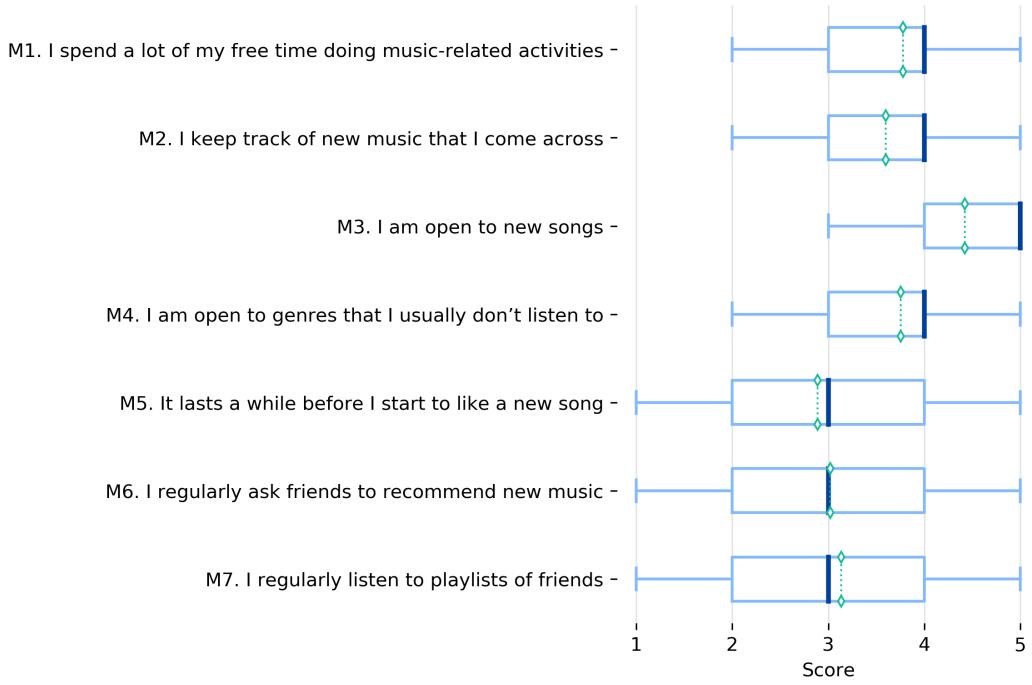


Figure 5.7: Distribution of the answers to the music-related questions; a score of 1 represents 'Disagree strongly', a score of 5 represents 'Agree strongly' (mean: green diamonds; median: dark blue line)

Trait	**W**	**p**	**Gaussian?**
Extraversion	0.978	0.553	✔
Assertiveness	0.982	0.718	✔
Activity	0.931	0.010	no
Agreeableness	0.975	0.420	✔
Altruism	0.936	0.016	no
Compliance	0.947	0.040	no
Conscientiousness	0.981	0.643	✔
Order	0.944	0.030	no
Self-Discipline	0.981	0.676	✔
Neuroticism	0.978	0.534	✔
Anxiety	0.972	0.341	✔
Depression	0.955	0.082	✔
Openness	0.973	0.384	✔
Aesthetics	0.963	0.159	✔
Ideas	0.953	0.067	✔

Table 5.7: Results of the Shapiro-Wilk test for the personality traits distribution (probably Gaussian if $p > 0.05$)

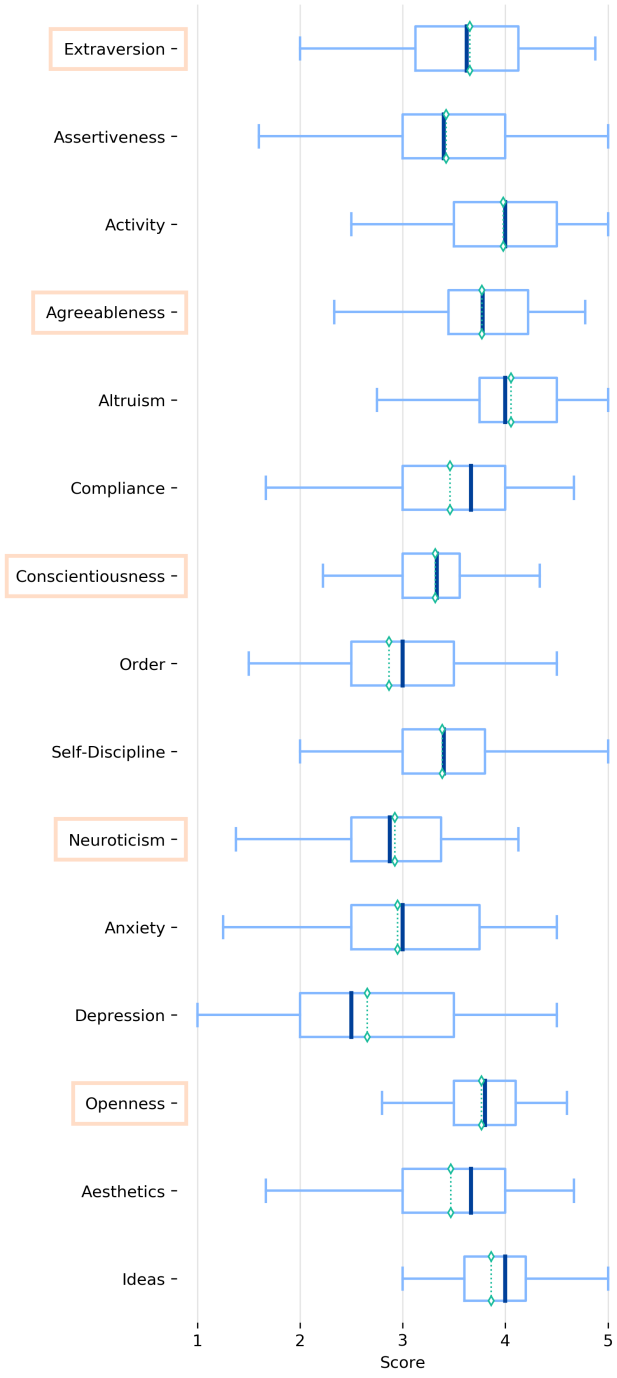


Figure 5.8: Distribution of the Big Five personality dimensions and its ten facets [33] among the participants; the higher the score, the more the participants tend to have this personality trait (mean: green diamonds; median: dark blue line)

Question	W	p	Different?
F1	188.0	0.980	no
F2	150.5	0.001	✔
F3	124.0	0.058	no
F4	138.5	0.724	no
F5	141.0	0.033	✔
F6	76.5	0.001	✔
F7	55.0	0.766	no
F8	125.0	0.671	no
F9	100.5	0.860	no
F10	102.0	0.413	no
F11	83.0	0.015	✔
F12	69.5	0.723	no
F13	37.0	0.302	no
F14	49.0	0.513	no

Table 5.8: Results of the two-sided Wilcoxon signed-rank test for the answers to the fairness questions (the distribution of the answers for the two versions of the system are probably different if $p < 0.05$; W = sum of the signed ranks)

Perceived fairness

The participants were assigned two tasks to evaluate the two versions of the system. Version 1 ranked all selected songs based on the time they were added and version 2 ranked the songs based on their dissimilarity to the individual profiles. Euclidean distance was applied to calculate the dissimilarity between the track attributes of the song and the track attributes representing the profile. Both versions first rank based on the number of likes. When a song has two likes, it obviously has a higher probability of being liked by the group members than a song with only one like. When the participants completed a task by creating a playlist together for a given scenario, they were asked to answer questions about the top X selected songs (X representing two thirds of the number of selected songs). In theory, the most popular songs should be ranked highest and are thus included in the top X final playlist. This assumption is valid for a ranking algorithm that takes into account the preferences of its users. However, a popular playlist does not always imply that the playlist is fair.

A two-sided Wilcoxon signed-rank test was conducted to analyse if the non-parametric answer distributions for version 1 and version 2 are similar. It is an alternative to the paired T-test, which assumes the data is normally distributed. As can be seen in Figure 5.9, questions F2, F5, F6 and F11 have different answer distributions. This is confirmed by Table 5.8. The differences in the answers to F3 were just below the threshold ($p = 0.058$). Nevertheless, the results demonstrate the superiority of version 2 (dissimilarity-based ranking) in terms of fairness. Issues arise with the time-based ranking algorithm if users don't start adding songs at the same time. The slowest user will have almost no songs included in the final playlist

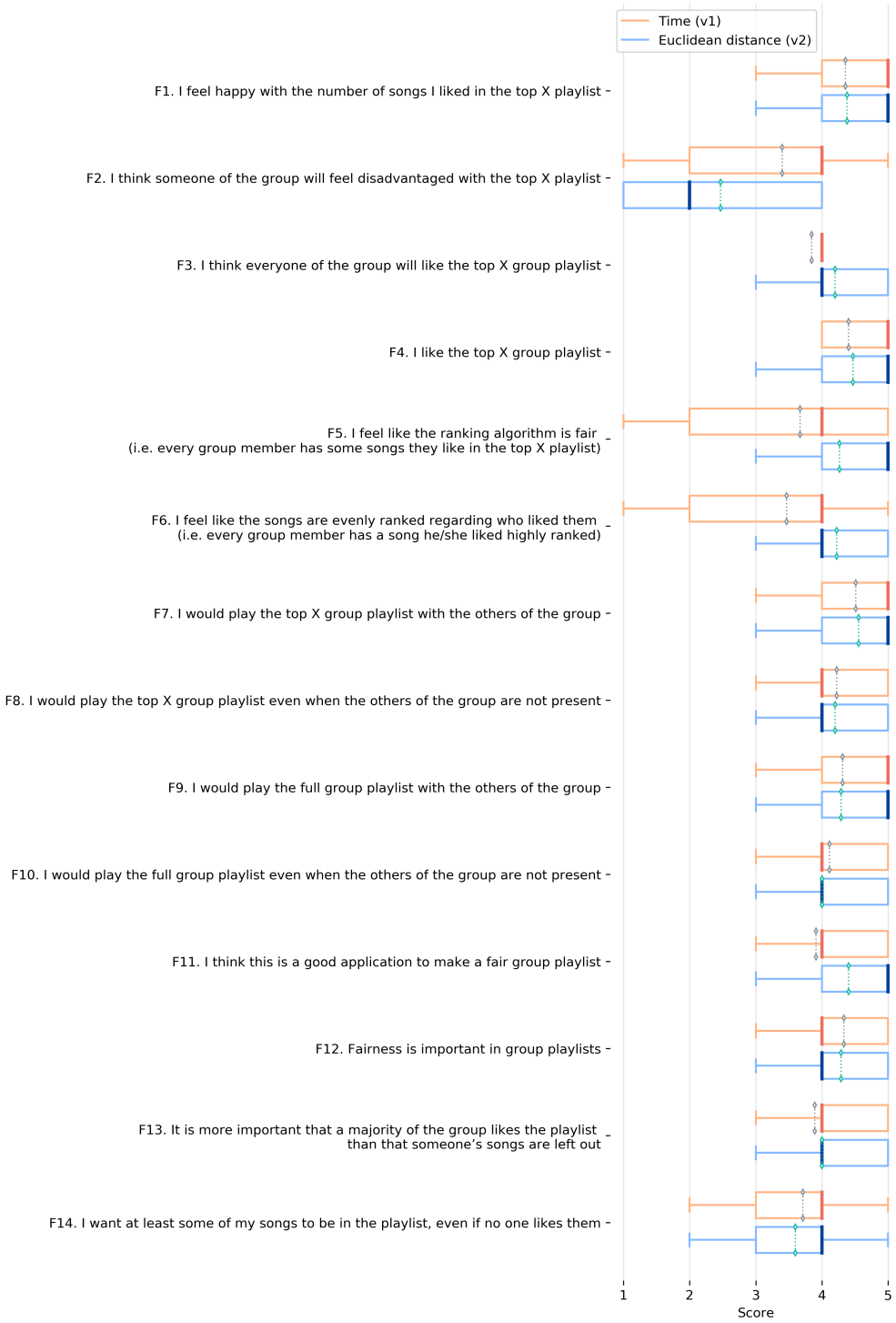


Figure 5.9: Distribution of the answers to the fairness questionnaire for version 1 (v1) and version 2 (v2) of the system; a score of 1 represents 'Disagree strongly', a score of 5 represents 'Agree strongly' (mean: grey and green diamonds; median: red and dark blue line)

minimum ratio	v1:	0.561
	v2:	0.643
mean ratio	v1:	0.736
	v2:	0.754

Table 5.9: The ratio of individual songs included over all individually selected/liked songs (minimum ratio: mean of all lowest ratios per group for version 1 and 2; mean ratio: mean of all ratios for version 1 and 2

as they are ranked lowest. Although all participants started at the same time, some finished their pre-test questionnaire sooner than others. As a result, they could add songs before the others could.

One way of objectively measuring fairness is to determine the number of songs included in the final playlist over the number of selected/liked songs for each individual. Table 5.9 shows the mean of all lowest ratios per group and the mean of all ratios for each version. The minimum ratio for version 1 is smaller than for version 2 ($\Delta = 0.082$), which supports the findings in the distributions of the answers to the fairness questions. The average ratio of included songs over selected songs for an individual is more than two thirds (the total number of songs making it to the final playlist over the total number of selected songs). This can be explained by the fact that group members can like each others songs. This pleases two group members at once when this song is included in the final playlist.

A remarkable observation in the box plots is the difference in distribution between F1 and F2. If people think someone will feel disadvantaged with the top X playlist, why does everyone seem to be happy with the number of songs they liked in the final playlist? There are two possible explanations. People could think someone will feel disadvantaged while this person is in fact satisfied. Another explanation is that if one group member is obviously disadvantaged, all three group members will report this in F2. However, in F1 there will only be one person reporting dissatisfaction. The answers of the other two group members can overshadow the answer of the disadvantaged individual. Correlations between the ratio of included songs per individual and the given answers will be discussed in the next chapter.

The box plots also show that the participants would rather listen to the top X playlist and the full playlist when the group is present. For F7 and F8, the Wilcoxon signed-rank test shows that the distributions are different for both versions (v1: $W = 36.0$, $p = 0.011$; v2: $W = 46.0$, $p = 0.010$) but not for F9 and F10 (v1: $W = 94.5$, $p = 0.095$; v2: $W = 90.5$, $p = 0.078$). F8 and F10 have similar distributions (v1: $W = 35.0$, $p = 0.225$; v2: $W = 40.0$, $p = 0.060$) and F7 and F9 are different only for version two (v1: $W = 60.0$, $p = 0.129$; v2: $W = 26.0$, $p = 0.040$). The last observation discussed here is the small difference in the answers to the general questions F12, F13 and F14 for the two versions. This may indicate that depending on the level of fairness in the created playlist, the participants may change their point of views.

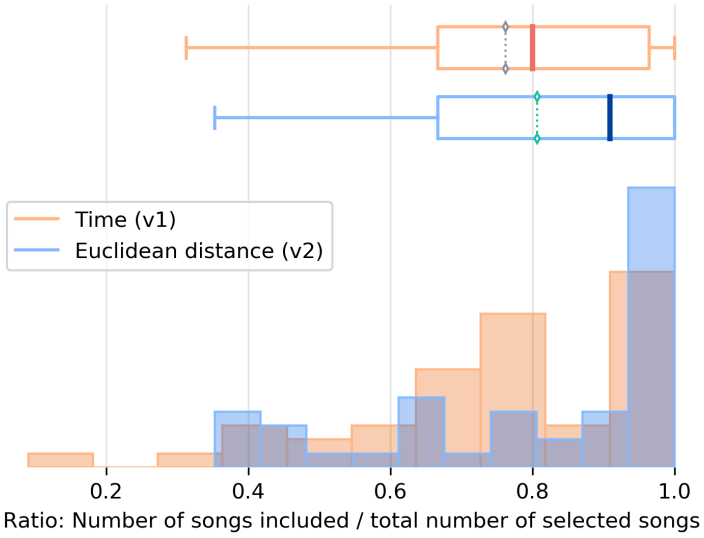


Figure 5.10: The number of songs the participants chose to include in the final playlist over the total number of selected songs for both versions (mean: grey and green diamonds; median: red and dark blue line)

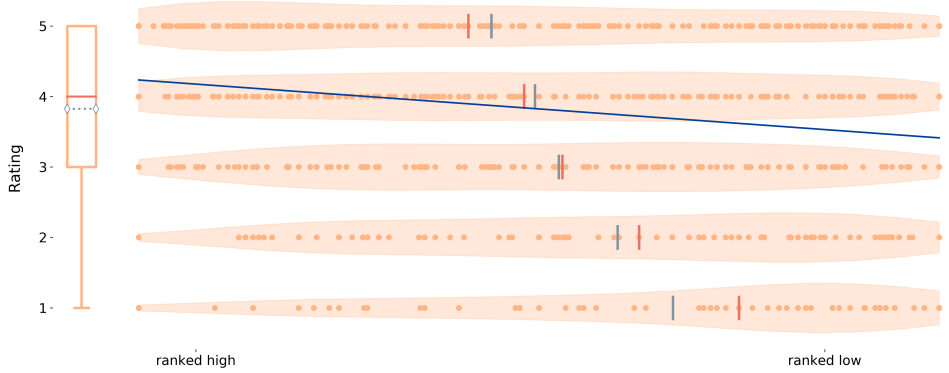
Select top

For the fairness questionnaire, two thirds of all selected songs were included in the final group playlist in order to keep consistency between the answers. In the next step the participants were asked to select the number of songs to include in the final playlist. If the ranking algorithm is qualitatively poor, this number is expected to be lower. Unpopular songs may be ranked high and one could choose not to include them despite the good songs ranked lower. Figure 5.10 shows the distribution of the normalized answers for both versions. Version 2 seems to be somewhat superior to version 1. However, there is no significant difference according to the Wilcoxon signed-rank test ($W = 299.0$, $p = 0.204$). The mean and median percentage of included songs is 76.2% and 80.0% for version 1 and 80.6% and 90.9% for version 2, respectively.

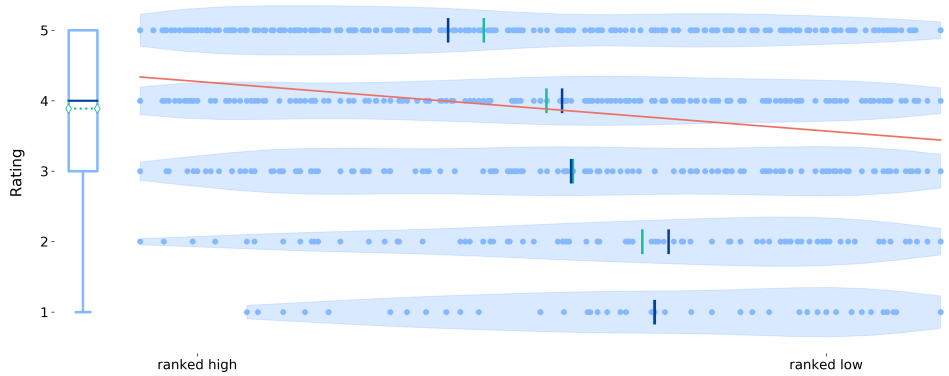
Ratings

The participants' song ratings are a better indicator to analyse the quality of the two ranking algorithms. Figure 5.11 shows scatter plots and its trend lines for both versions of the song's given rating and its position in the playlist. Violin plots per rating show the song's position density. The box plots show the distribution of the ratings. The slope of the trend lines are -0.059 and -0.064 for version 1 and 2 respectively. From this negligible difference it can be concluded that the dissimilarity-based ranking algorithm (explained in Section 4.3) had no positive effect in this user study. The factor dominating the negative slope are the votes as they explicitly express the popularity of the song and are thus ranked highest. For both versions low

ratings increase for lower positions and the opposite is true for the songs receiving five stars. When only the ratings for songs that the participant did not select or like are considered, the difference between the slopes of the trend lines is still insignificant (v1: -0.039; v2: -0.036; plots are shown in Appendix D).



(a) Version 1



(b) Version 2

Figure 5.11: Scatter plot of the song's rating and position in the playlist (the higher the rating, the more the participant likes the song); trend line of the scatter plot (blue and red); violin plots of the song's position density per rating (mean: grey and green line; median: red and dark blue line); box plot of the given ratings distribution (mean: grey and green diamonds; median: red and dark blue line);

Logs

The interactions of the participant with the system were logged during the user study. These logs offer some interesting insight in the popularity of the functionalities of the system. Table 5.10 presents the number of times an interface element was used. The elements are shown in Figure 5.10. In total, the participants selected/liked 1252 songs and removed 66 songs/likes, keeping 1186 likes (if two group members like the same song, this counts as two likes). 35.3% of all likes were given to songs selected by other group members. 51.3% of the searches were effective: 1146 searches were

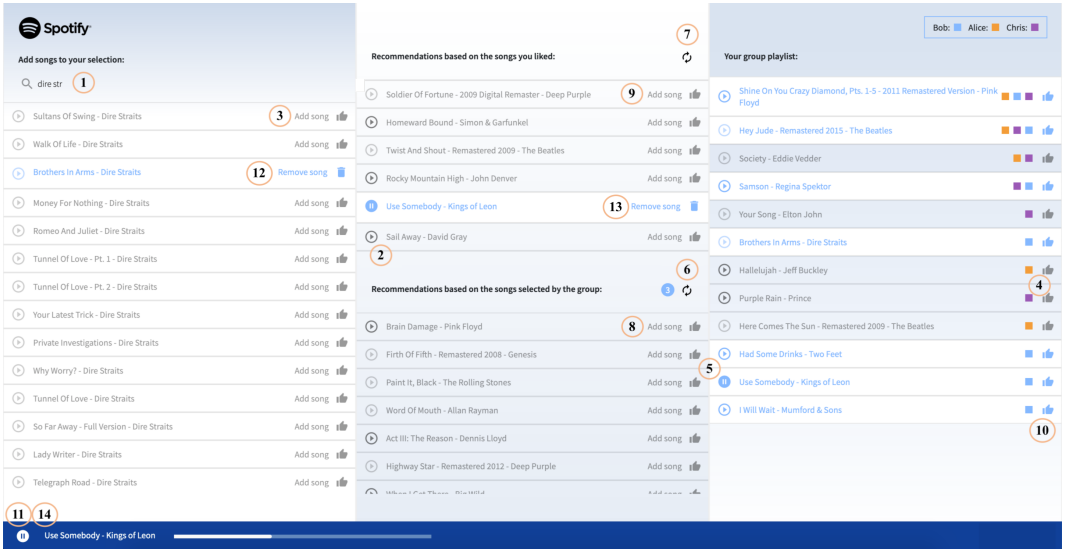


Figure 5.12: Logged interface elements (15 is not displayed as it is similar to 13)

1.	search	1146
2.	play	1009
3.	add song	589
4.	like song (playlist)	443
5.	pause	330
6.	refresh group recommendations	172
7.	refresh user recommendations	140
8.	group recommendation added	111
9.	individual recommendation added	109
10.	remove song (playlist)	52
11.	pause footer	21
12.	remove song	10
13.	individual recommendation removed	3
14.	play footer	3
15.	group recommendation removed	1
total	number of songs added/liked	1252
total	number of songs removed	66
total	number of selected songs	720

Table 5.10: Logged interactions in numbers (Figure 5.12 shows the corresponding interface elements)

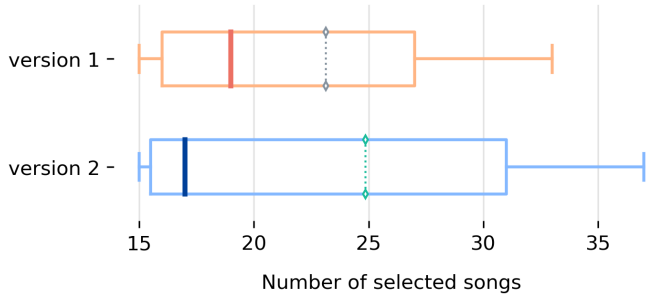


Figure 5.13: Distribution of the number of selected songs per version (mean: grey and green diamonds; median: red and dark blue line)

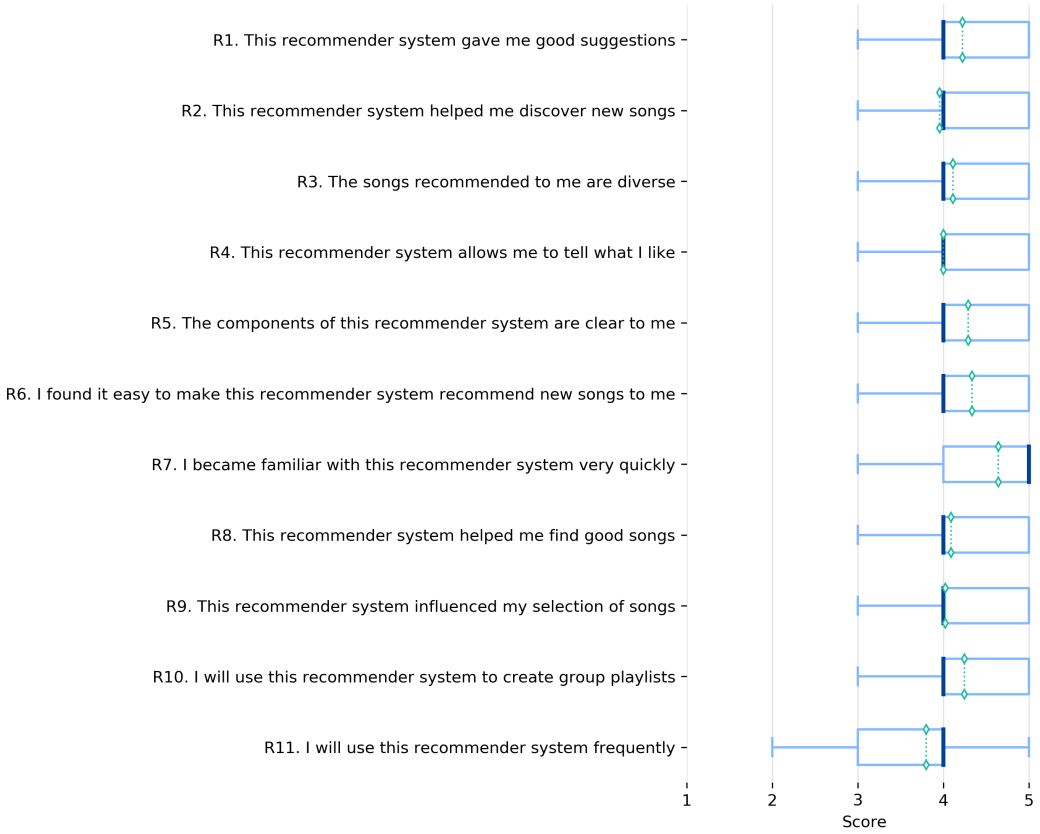


Figure 5.14: Distribution of the answers to the ResQue questions; a score of 1 represents 'Disagree strongly', a score of 5 represents 'Agree strongly' (mean: green diamonds; median: dark blue line)

needed to add 589 songs (search = click on search bar). Eventually, 720 songs were evaluated in the next steps of the user study. The distribution of the number of selected songs per version is shown in Figure 5.13. The average number of selected songs in version 1 was 23 and for version 2 and 25. The maximum number of selected songs was 55 for version 1 and 62 for version 2. Group recommendations seem to be somewhat more favoured than individual recommendations (32 more clicks on the refresh button and 2 more recommended songs selected). This could indicate that participant did their bests to discover and add songs the other group members would also like.

ResQue

The answers to the selected ResQue questions tend to be positive (Figure 5.14). The median score for all questions is 4, except R7 which has an even higher median score of 5 (strongly agree). In other words, the recommender system scores notably high on perceived ease of use and relatively high on recommendation accuracy, novelty and diversity, interaction and interface adequacy and perceived usefulness. Use intentions are represented by the answers to R10 and R11 which also suggest to be positive, although not all participants would use the recommender system frequently.

System Usability Scale (SUS)

The SUS questions were positively answered as well (Figure 5.15). A higher reported score for odd questions and a lower score for the even questions imply higher perceived usability of the system. S4 and S5 score best, indicating that the system is self-explanatory. Bangor et al. [63] published adjective ratings and the mean of 1180 internet-based web pages and applications (mean: 68.05; standard deviation: 21.56) to examine SUS scores. Figure 5.16 demonstrates an excellent usability score for this research's web application with a mean of 81.3 and a median of 82.5. Moreover, this shows the superiority of this web application compared to the average web applications examined by Bangor et al.

Suggestions

The 45 participants were invited to answer some open-ended questions (Table 5.11). The most popular suggestion given in one of these questions was to add a dislike button, reported by five participants. Four others said they would make the group react to all selected songs, for instance by scoring the songs, to take their preferences into account. One participant suggested to offer each group member a 'wild card' for a song they really want in the playlist. Two other participants support this idea by suggesting to give each group member a predefined number of wild cards. This principle of wild cards could be very interesting when applying it also to the dislike button. A person can choose some songs they really want in the final playlist and they can veto some others. The question remains if the dislike or the 'super like' is prioritized. Three participants would like to gain more insight into the ranking

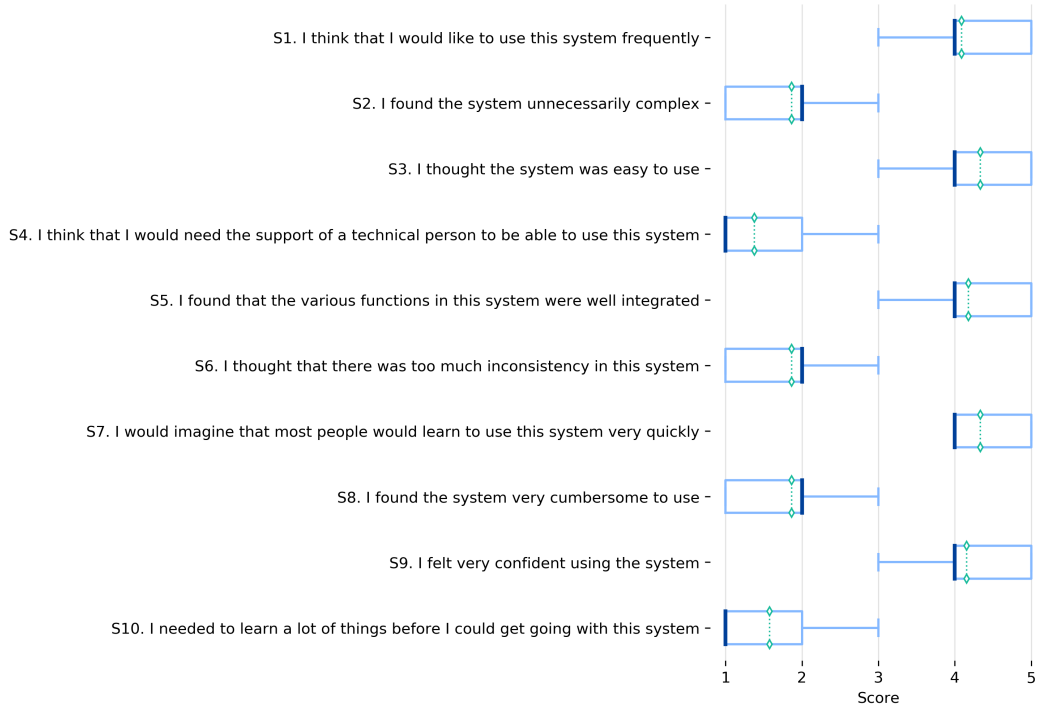


Figure 5.15: Distribution of the answers to the SUS questionnaire; a score of 1 represents 'Disagree strongly', a score of 5 represents 'Agree strongly' (mean: green diamonds; median: dark blue line)

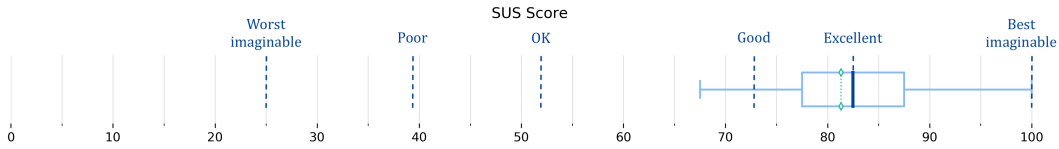


Figure 5.16: Distribution of the SUS score (mean: green diamonds; median: dark blue line); adjective ratings according to Bangor et al. [63]

algorithm. For four other participants the importance of the number of votes was not clear. Some transparency would thus be appreciated by these seven participants.

Four participants answered question 1 by proposing to include an equal amount of songs of each group member in the final playlist. Similarly, one participant proposed to limit a group member who enthusiastically adds much more songs than the others by making him/her wait for the others to be able to keep up. One participant suggested to also take into account the duration of the song to ensure that every group member more or less gets the same amount of 'play time'. Another participant reported that fairness is not important since he/she would use the tool mainly to share and discover music.

One participant reported that the refresh recommendations functionality was

unclear. He/she could not differentiate new recommendations. Two other participants reported that new group recommendations were added too fast making this component less usable. This issue could be solved by not recommending songs for every added song or by giving the recommendations component a bigger window. One participant could not easily read the light grey text. This decreases the system's usability, but is easily fixed. Another participant did not know that a like for a song could be removed by clicking on 👍. This icon could be replaced by 🗑, although the song is not removed from the playlist if other group members liked this song too, which could also lead to confusion. Two participants answered to question 4 by proposing to include album covers. One of them would like to be able to check out the album of a specific song. Hence, one of the songs of the album can easily be selected.

1.	Do you have any suggestions to make the ranking of songs in the playlist more fair?
2.	What aspects of the application were not clear to you?
3.	Do you have any suggestions to make the application more easily usable?
4.	Do you have any suggestions to make a better application to make group playlists?

Table 5.11: Open-ended questions (2, 3 and 4 were asked in the post-test questionnaire, question 1 was asked in the fairness questionnaire)

5.4 Conclusion

This chapter discussed the conducted user study in detail. A within-subjects experiment design was chosen to evaluate the system and examine the perceived fairness in group music recommender systems. 45 participants went trough the process explained in the Procedure section. The results show that the time-based ranking algorithm performs poorly in terms of fairness. Users arriving late at the application will only have a few of their songs highly ranked. This results in a lower score in terms of fairness for the time-based algorithm. Ratings per song show that the dissimilarity-based ranking does not perform better than the time-based ranking in terms of predicting the popularity of the song. ResQue and SUS results show that the system scores high on recommendation quality and usability. Finally, some participants' suggestions to improve the system were discussed. The possibly most interesting suggestion is to use 'wild cards' for songs the user really wants to include in the final playlist. This suggestion can be extended to dislikes. The next chapter analyses the results and correlations between these results.

Chapter 6

Discussion

The results of the user study were presented in Section 5.3 of the previous chapter. This chapter discusses more in depth the perceived fairness, correlations between the Big Five dimensions and perceived fairness, how the ranking algorithm influences perceived fairness and the overall usability of the system. Thereafter the limitations of this study are addressed. The numbers of the fairness questions as presented in Table 5.4 are used to refer to them.

6.1 Perceived fairness

The dominating factors in the fairness questionnaire are 1) the number of songs selected by an individual included in the final playlist over the total number of songs selected by this individual defined as the '*individual ratio*' and 2) the minimum individual ratio per group defined as the '*group minimum ratio*'. Figure 6.1 shows a Spearman correlation matrix which determines the monotonicity of the relationship between (in this case) the responses to the fairness question and the ratios defined above. It is the non-parametric version of the Pearson correlation matrix. This Spearman correlation matrix supports the theory that the individual and group minimum ratio significantly influence the responses to the fairness questionnaire. Version 1 of the system integrated the time-based ranking algorithm and version 2 uses the dissimilarity-based algorithm.

In version 1, the individual ratio correlates strongest with F1 and F4 ('*I feel happy with the number of songs I liked in the top X playlist*' and '*I like the top X group playlist*')($\rho = 0.32$ and 0.35 respectively). The group minimum ratio correlates strongly with F2 ('*I think someone of the group will feel disadvantaged with the top X playlist*'). The minimum individual ratio of a group represents the most disadvantaged group member. This observation is thus in line with the expectations as someone will feel disadvantaged when the minimum individual ratio of the group is low. The positive correlations with F3, F5, F6 and F11 are also as expected (ρ = 0.5, 0.57, 0.59 and 0.43 respectively). When someone feels disadvantaged, not everyone of the group will like the top X playlist, the ranking algorithm is not fair and the songs are probably also not evenly ranked. The positive correlation between

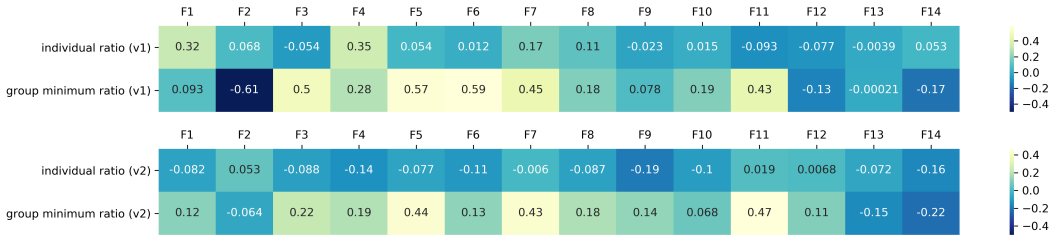


Figure 6.1: Spearman correlation matrix per version for the individual and group minimum ratio and the fairness questionnaire

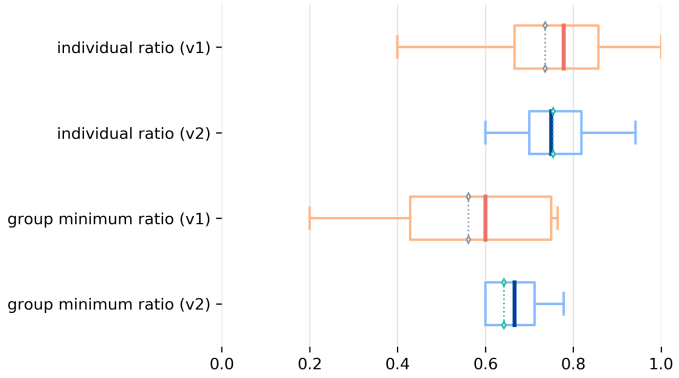


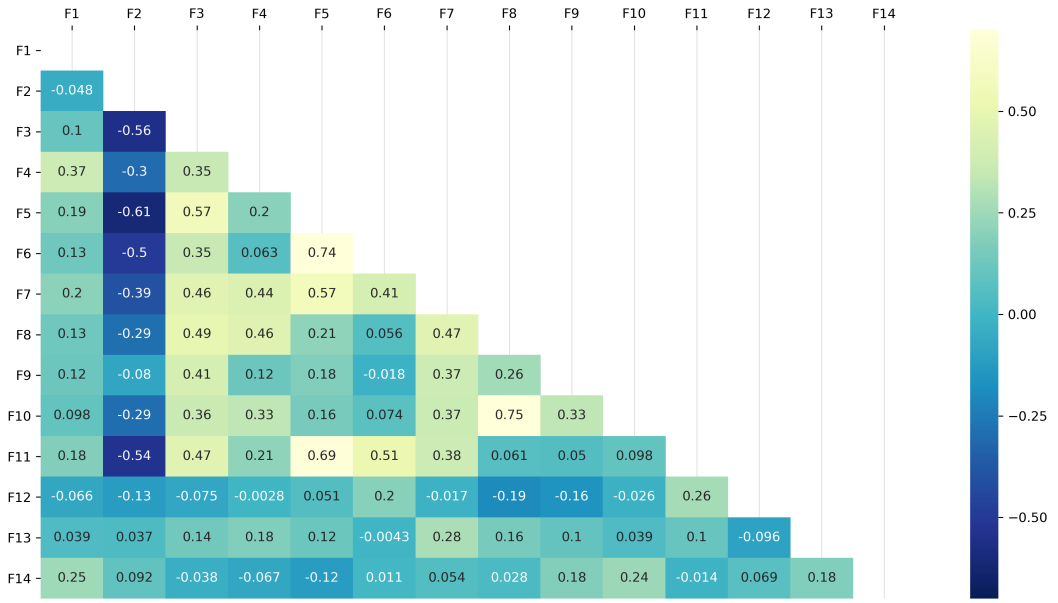
Figure 6.2: Distribution of the ratios per version (mean: grey and green diamonds; median: red and dark blue line)

the group minimum ratio and F7 ($\rho = 0.43$) suggests that the higher the group minimum ratio, the more the participants would play the top X playlist with the group. This is probably out of respect for the disadvantaged group member.

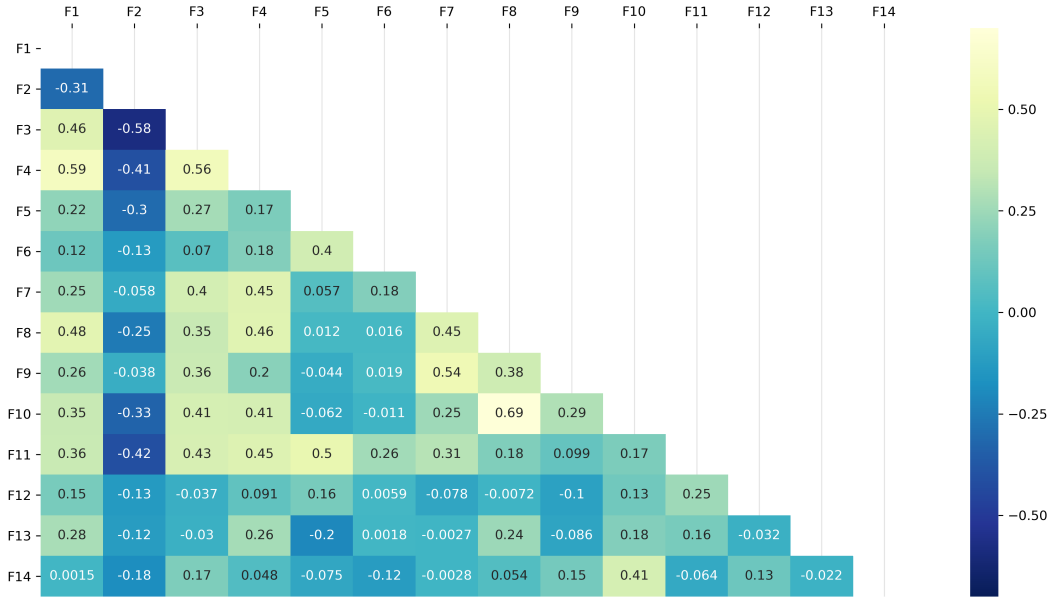
Version 2 shows some more irregularities in the correlations with the fairness questions. These can partly be explained by the distribution of the ratios shown in Figure 6.2. Compared to the first version, the ratios of version 2 are less spread out, which makes it more difficult to determine correlations. This could explain why version 2 shows less significant correlations. The strongest correlations are between the group minimum and F5, F7 and F11 ($\rho = 0.44$, 0.43 and 0.37 respectively). The same argumentation can be given as for version 1 to explain these correlations.

Figure 6.1 clearly highlights that the ratios significantly influence the responses to some fairness questions. This should be taken into account when determining other correlations related to perceived fairness.

Figure 6.3 shows how the answers to the fairness questions correlate with each other. For both versions, F2 is negatively correlated with F3 to F11 as it is a negatively keyed item (its score should be reversed). The strongest correlations are between F5 and F6 ($\rho = 0.74$), F5 and F11 ($\rho = 0.69$) and F8 and F10 ($\rho = 0.75$) in version 1. This substantiates that a ranking algorithm perceived as fair is positively associated with evenly ranked items and with an application perceived as fair. The



(a) Version 1



(b) Version 2

Figure 6.3: Diagonal Spearman correlation matrix for the fairness questionnaire

correlation between F8 and F10 demonstrates that if participants would play the top X playlist even when the others of the group are not present, they would also play the full playlist even when the others of the group are not present. F8 and F10 are also positively associated in version 2 ($\rho = 0.69$). As has been discussed in Section 5.3, version 2 scored better on perceived fairness which could explain the stronger correlations with F1. For a fairly ranked playlist the participant will not only report that he/she is happy with the top X playlist, but also that he/she thinks the others will be happy with the top X playlist. Interestingly, version 2 shows a correlation between F10 and F14 ($\rho = 0.41$). Participants who would play the full group playlist even when the others are not present also indicate to want some of their songs included even if no one likes them. Maybe their songs were not included in the top X playlist (explaining why they would listen to the full playlist) and reported that they rather have them included.

6.2 The Big Five and perceived fairness

F12, F13 and F14 ('*Fairness is important in group playlists*', '*It is more important that a majority of the group likes the playlist than that someone's songs are left out*' and '*I want at least some of my songs to be in the playlist, even if no one likes them*' respectively) are the most general questions of the fairness questionnaire. The answers to these questions should be almost independent of the ranked playlist. 'Almost' should be emphasized, since Figure 6.1 shows that answers to F14 depend on the individual ratio of included songs to a small extend. Figure 6.4 presents the Spearman correlation matrix for the given answers to the three general fairness questions, the SUS score and the Big Five personality dimensions with its corresponding facets. If the responses to the fairness questions differed for version 1 and version 2, the mean was taken for this matrix.

Openness shows the strongest correlation of all traits. It is negatively correlated to F12 (ρ = -0.45). This indicates that people who are more open to experience do not think fairness is very important in group playlists. For them it is probably

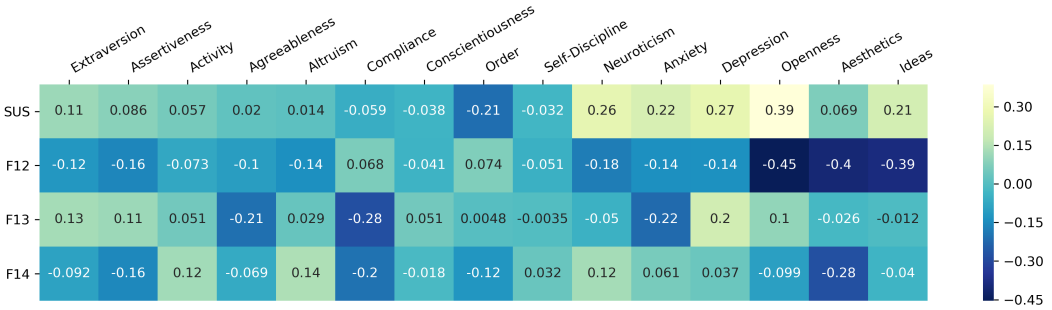


Figure 6.4: Spearman correlation matrix for the SUS scores, the three general fairness question responses and the Big Five personality dimensions and their corresponding facet pairs

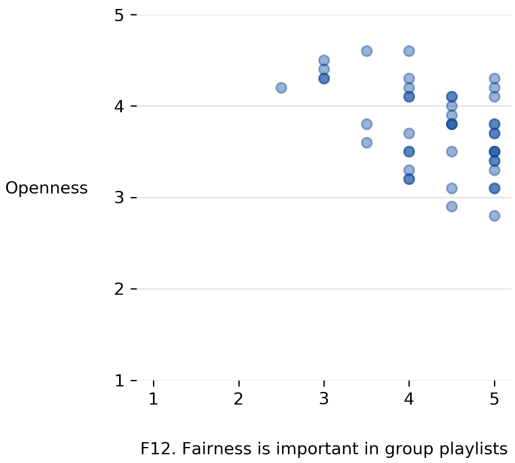
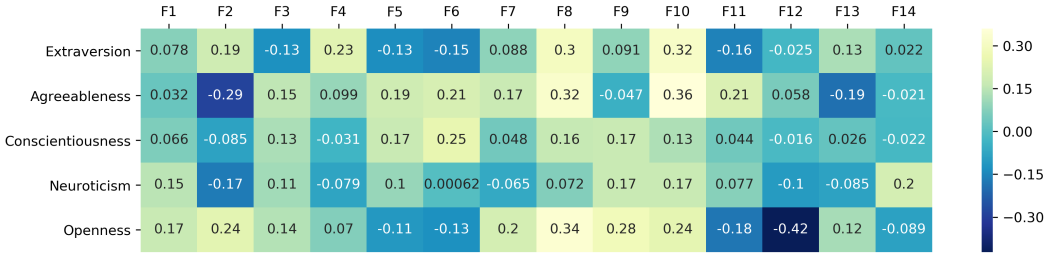


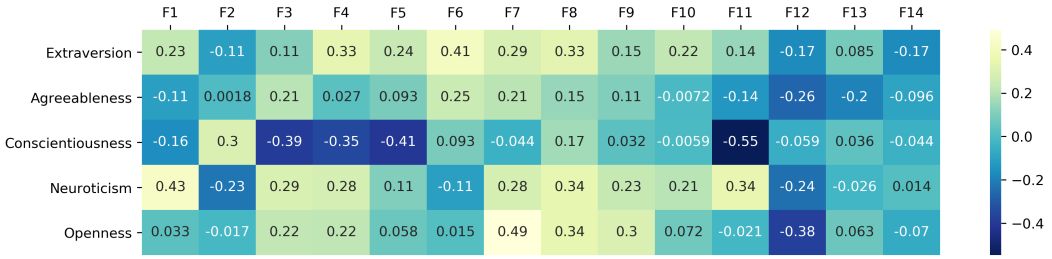
Figure 6.5: Scatter plot of the participant's level of Openness and his/her (average) response to F12

more valuable to discover good songs than that all preferences are equally taken into account. Figure 6.5 shows that this theory is hard to generalize as only a few individuals scoring high on openness reported that to them fairness is not important in group playlists. To substantiate these theories for open individuals, a larger sample size is needed. This also applies to the correlations of its facets Aesthetics and Ideas. In addition, Openness shows a medium association with the SUS score ($\rho = 0.39$). These individuals will have the lowest chance to panic when exploring a new interface. Even if an application is complex, they will be the first to start experimenting with it. It is probably because of their curiosity and wide interests that they find a system more easy to use than others. Ideas, the facet related to Openness, also correlates positively with SUS ($\rho = 0.21$). This upholds the explanation of curious people becoming rapidly familiar with a new interface. But again, a larger sample size is needed to give more accurate results and to confirm this theory.

Compliance, the facet related to Agreeableness, shows a small association with F13 (ρ = -0.28) and F14 (ρ = -0.2). This negative correlation can be expected as people who are less stubborn, will not be the ones to absolutely want their songs to be included in the playlist. They also seem to be open to songs that are liked by only a minority of the group. An unexpected result is the small positive correlation between Altruism and F14 ($\rho = 0.14$). One would normally expect more selfish people to want their own songs included even if no one likes them. The same can be said about the remarkable negative correlation between Assertiveness and F14 (ρ = -0.16). However, this deviation from what is expected could be a coincidence as F14 is also subject to the individual and group minimum ratio (Figure 6.1). Another observation which is difficult to explain is the positive correlation between SUS on the one hand and Neuroticism and its facets on the other (ρ = 0.26, 0.22 and 0.27 respectively). Not contented individuals are not expected to give positive feedback, but apparently in this user study they tend to give higher usability scores.

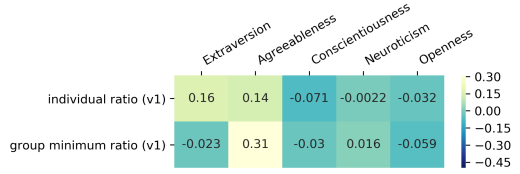


(a) Version 1

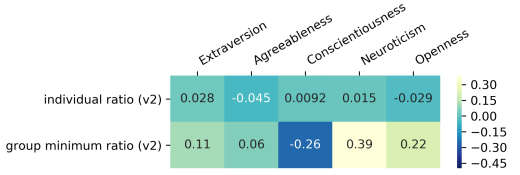


(b) Version 2

Figure 6.6: Spearman correlation matrix for the Big Five personality dimensions and the answers to the fairness questionnaire for both versions



(a) Version 1



(b) Version 2

Figure 6.7: Spearman correlation matrix for the individual and group minimum ratio and the Big Five personality dimensions

Figure 6.6 presents the Spearman correlation matrix for the Big Five personality traits and responses to the fourteen fairness questions. The correlations with F12, F13 and F14 are discussed above (Figure 6.4). One should bear in mind that the answers to the questions F1 to F7 and F11 are highly dependent on the ranked playlist (Figure 6.1). These questions could therefore unintentionally show correlations when individuals with a specific trait were presented an unfair ranked playlist. Figure 6.7 shows the correlations between the Big Five and the ratios as defined in Section 6.1. As can be observed, in version 1 Agreeableness shows a positive correlation of $\rho =$ 0.31 with the group minimum ratio and in version 2 Conscientiousness, Neuroticism and Openness show associations with the group minimum ratio as well ($\rho =$ -0.26, 0.36 and 0.22 respectively). Extraversion is slightly correlated with the individual ratio in version 1 ($\rho = 0.16$) and with the group minimum ratio in version 2 ($\rho =$ 0.11).

The strong correlations of Conscientiousness for version 2 with fairness questions F2, F3, F4, F5 and F11 (Figure 6.6b) can partly be explained by the unfair selection of songs for the final playlist (Figure 6.7b). A similar explanation can be given to the positive correlation between Neuroticism and F11 (Figure 6.6b) and the negative correlation between Agreeableness and F2 (Figure 6.6a). Figure 6.1 in particular shows that the responses to F2 for version 1 and F11 for version 2 are highly dependent on the group minimum ratio. Figure 6.7 suggests to look at Conscientiousness, Neuroticism and Openness in version 1 and to Extraversion and Agreeableness in version 2 as these are least dependent on the ratios. The unexpected correlations between Conscientiousness and F2, F3, F4, F5 and F11 are for instance contradicted by Figure 6.6a, and Figure 6.7 tells us that version 2 reports more reliable results in terms of Conscientiousness. People scoring high on Conscientiousness could see it as their duty to genuinely report the perceived unfairness, but the dominating factor in these correlations is clearly the group minimum ratio.

Looking at the more reliable results, Openness in version 1 seems to show the strongest correlations (Figure 6.6a). Participants scoring high on Openness would play the top X playlist even when the others of the group are not present (F8: $\rho =$ 0.34) and would also listen to the full playlist with or without the group (F9: $\rho = 0.28$; F10: $\rho = 0.24$). This demonstrates that these individuals are open to (all) the songs of their group members. However, although they don't think fairness is important in group playlist, they do report perceived unfairness (F2: $\rho = 0.24$; F11: $\rho =$ -0.18). Hence, it is important to differentiate between perception and values. Agreeable and extrovert participants also tend to report more that they would listen to the playlist ((F7: $\rho = 0.21$ and 0.29; F8: $\rho = 0.15$ and 0.33). Agreeable individuals are expected that they 'go with the flow' in group settings [34], which explains that they like to listen to the playlist when the group is present. Extrovert, social, outgoing individuals who like to connect with others [34] will probably be the ones to play the group playlist not only with the friends they composed the playlist with but also in other situations. The correlations concerning Conscientiousness and Neuroticism are too small to draw conclusions.

The positive correlation between F6 and Extraversion ($\rho = 0.41$) in version 2 is contradicted in version 1 ($\rho =$ -0.15). This indicates that the relatively strong correlation in version 2 likely occured because the individuals evaluated a playlist which was more evenly ranked. The correlations between F6 and Conscientiousness in version 1 ($\rho = 0.25$) and Agreeableness in version 2 ($\rho = 0.25$) could be given the same explanation but further analysis is needed to be able to explain these unexpected observations. If these correlations were generalizable, this would mean that people scoring high on Conscientiousness and Agreeableness easily perceive a playlist as evenly ranked. As there is not enough empirical evidence, this theory cannot be substantiated.

6.3 The ranking algorithm and perceived fairness

Section 5.3 empirically proved the superiority of the dissimilarity-based ranking algorithm over the time-based algorithm in terms of fairness. The ones who add their songs faster will have more songs highly ranked and users arriving late at the application or selecting songs at a slower pace will be disadvantaged. Although the dissimilarity-based ranking algorithm does not integrate fairness, it naturally mixes the selected songs more than the time-based algorithm. In this way, the dissimilarity-based ranking algorithm gets higher scores in the fairness questionnaire. The Wilcoxon signed-rank test demonstrated the difference in answers for F7 and F9 only for version two. This slightly indicates that participants appreciate the ranking algorithm of version 2 since they would rather play the top X playlist with the others of the group than the full playlist. However, plotting the ratings of users combined with their given position in the playlist showed no significant difference between the time-based and dissimilarity-based ranking algorithm. Hence, the dissimilarity-based ranking did not succeed in predicting the preferences of the users.

6.4 Usability

The positive ResQue and SUS scores given by the 45 test users illustrate the high usability and quality of the system. The average total SUS score of 81.3 is very close to the adjective rating 'Excellent' and the median score of 82.5 passed the test to be called 'Excellent' [63]. Some even gave a score of 100 which is 'Best imaginable'. The application scores also remarkably better than the average internet-based web page or application which has a mean of 68.05 [63]. An interactive tutorial helped the participants to understand the system and its components. Most participants were very enthusiastic and found the app fun to use. For some it was a nice distraction during the quarantine period due to COVID-19. Six participants even asked if this application would become publicly available or if Spotify would integrate it. These participants were keen on using the application in the future. Most groups did not limit theirselves to 15 songs. The average number of selected songs in version 1 was 23 and for version 2 and 25. One group even added 55 songs and another 62 (in version 1 and 2 respectively). This confirms that the participants enjoyed using the applications. Four participants explicitly reported that they liked the voting mechanism. Two of them emphasized the benefit in bigger groups. When they organize a party for example, they want to find an effective tool to filter all selected songs and only keep the most popular ones. They preferred this application over the collaborative playlists in Spotify.

6.5 Limitations

This research presented some very interesting results. However, the small sample size of the user study and the large number of dependencies limit the generalizability of the observations. The responses to the fairness questionnaires are highly dependent

on ranking of the songs, which in turn depends on the time they were added (version 1) or on the selected songs' attributes (version 2). This makes it difficult to draw general conclusions related to personality. Personality facets like Activity, Order and Depression were assessed by only two items of the BFI. Furthermore, the age of 18 to 24 is overrepresented in this study. Since 26% of the Spotify users is between 18 and 24 [48], they do represent a reasonable part of the target users. Another limitation arises in the abrupt termination of the user task for some test users. If one participant clicks on 'Finished', the other group members immediately receive a message to proceed to the next step. This ensures that the whole group answers questions about the same selection of songs. Three participants reported that they wished to have more time to go over the songs and perhaps like some more songs of others.

6.6 Conclusion

The results of the user study and correlations between them were further discussed in this chapter. The perceived fairness is highly dependent on the individual ratio (defined as the number of songs selected by an individual included in the final playlist over the total number of songs selected by this individual) and the minimum individual ratio per group defined as the group minimum ratio. It is difficult to draw conclusions from the correlations between perceived fairness and the personality traits since the former is probably more influenced by the individual and group minimum ratio and the sample size of this study is rather small. Nevertheless, Openness shows some moderate to strong associations. Participants scoring high on Openness to experience do not attach much importance to fairness. It is important to differentiate between values and perceptions as some of them do report unfairness when the playlist does not seem to be fairly ranked to them. Open, extrovert and agreeable participants indicated that they would play the top X playlist and the full playlist. Extrovert and open individuals tend to listen to the playlist even if the others of the group are not present. The correlations concerning Conscientiousness and Neuroticism are too small or highly dependent on the ratios to draw conclusions. The results regarding the ranking algorithm and usability presented in the previous chapter were discussed as well. The dissimilarity-based algorithm performs better than the time-based ranking in terms of perceived fairness but not in terms of predicting the popularity of the song. The majority of the participants found the system easy and fun to use, resulting in high ResQue and SUS scores. The main limitations of this study were the small sample size and the dependencies in the responses of the fairness questionnaire.

Chapter 7

Conclusion

This research is the first to investigate perceived fairness in group music recommender systems. Some very interesting results were presented regarding the overall perception of fairness, the influence of personality on the perceived fairness, the influence of the ranking algorithm on the perceived fairness and the usability of the system. 45 participants were recruited to conduct a within-subject experiment with the developed group music recommender system. To get to a suitable interface for this research two prototypes preceded the final design. At each iteration, feedback was collected to integrate in the next interface design. The objective changed in this process to eventually get to a more relevant research. The first prototype aimed at supporting users to achieve consensus. However, the interface was created for a duo and was too complex. The interaction between two people differs from the interaction in the group, which limits the generalizability of the results and the scalability of the interface. The second prototype ranks the songs selected by the group member and shows explanations for the position of the songs and visualizes the group and individual profiles. These visualizations were still perceived as too complex and users indicated that they would rather focus more on just creating the playlist together. This resulted in the final design for this research which objective is to investigate the perceived fairness of the ranked playlist and its usability.

A time-based and dissimilarity-based ranking algorithm were implemented to analyse the quality of the latter and the influence of the ranking algorithm on the perceived fairness. The time-based algorithm ranks the songs based on the time they are added. As a result, the most recently added song is added at the bottom of the list, ranked lowest. The dissimilarity-based method is related to content-based filtering in combination with aggregated predictions as it determines the similarity of a song with the individual profiles to rank the song in the playlist. In this way, it tries to predict the probability that other group members will like the song. Both algorithms first rank on the number of votes. If two users liked a song, it will always be ranked higher than a song liked by only one group member. The dissimilarity-based algorithm showed its superiority over the time-based algorithm in terms of fairness but not in terms of quality. The participant's ratings of the selected songs indicate that only the voting mechanism effectively ranks the songs. The dissimilarity-based ranking

algorithm naturally ranks the songs more evenly than the time-based because in the latter the user arriving latest at the application will almost have no songs ranked high.

It is difficult to draw conclusions from the correlations between the Big Five personality dimensions and the responses to the fairness questions. The responses are highly dependent on the number of songs each individual got included over the number of songs this individual selected and the small sample size of the user study limits the generalizability. Nevertheless, the responses suggest that people open to experiences do not think fairness is very important in group playlists. Some of the participants scoring high on openness did report unfairness if they perceived the group recommender system as unfair. Open, extrovert and agreeable participants indicated that they would play the top X playlist and the full playlist. Extrovert and open individuals tend to listen to the playlist even if the others of the group are not present. Other correlations were too small or not reliable enough to draw conclusions.

The application showed excellent usability and had a unique approach of making users discover and share music. Users can search for songs and every time a song is added some individual and group recommendations are given. The user does not have to explicitly express their preferences in song ratings or genres. For this reason it is more user friendly than other group music recommender systems. The interaction logs showed that refreshing the recommendations based on the songs selected by each group member was more popular than refreshing recommendations based on the individually selected songs. This indicates that users like to discover songs that are in line with the music taste of the other group members. Moreover, the logs demonstrate that both individual and group recommendations helped users to add songs to the playlists. Participants reported that the application was very fun to use. Some even explicitly asked if they could use the system in the future for social gatherings like parties or just to easily share music with friends. Overall, the results of this study suggest that the developed group music recommender system could inspire many commercially available music applications in the way it helps users to discover and share music with others.

Future work

This research lays the foundation for future work in the area of personality and perceived fairness. The dependencies in evaluating a ranked playlist should be decreased to a minimum to research the influence of personality on perceived fairness more in depth. Additionally, a fairness-enhanced algorithm could be compared to the dissimilarity-based algorithm presented in Section 4.3 to investigate if people notice and appreciate the integrated fairness aspect. For this to deliver generalizable results, the sample size should be increased to cover as many different personalities as possible. The developed web application was enthusiastically received by the test users and showed its great potential. The system could be improved by a more sophisticated ranking algorithm and by implementing the 'wild cards' suggested by some participants. This would enable the users to select a limited number of songs

they definitely want to include in the final playlist. These 'wild cards' can also be used to express dislikes and veto songs from the final playlist. Some participants would have benefited from transparency regarding the ranking algorithm. Hence, some explanations could ameliorate the usability of the system. The upgraded version of the application could then analyse the added value of the explanations and further investigate the importance and the perception of fairness.

Every contribution in this field of study will gradually help to ameliorate the overall user experience and is invaluable to the enlarging user community of (group) music recommender systems. In this manner, researchers and developers could provide the optimal environment to listen to, discover and share music.

Appendices

Appendix A

User study

The entire user study needed to be online conductible because of the measures taken in response to COVID-19. This requires a smooth flow of all elements of the user study (Figure A.1). Screenshots of objective, informed consent, login, fairness questionnaire, select-top and rating page are shown below. The screenshots of the interactive interface are shown in Appendix B and the questionnaires are presented in Appendix C.

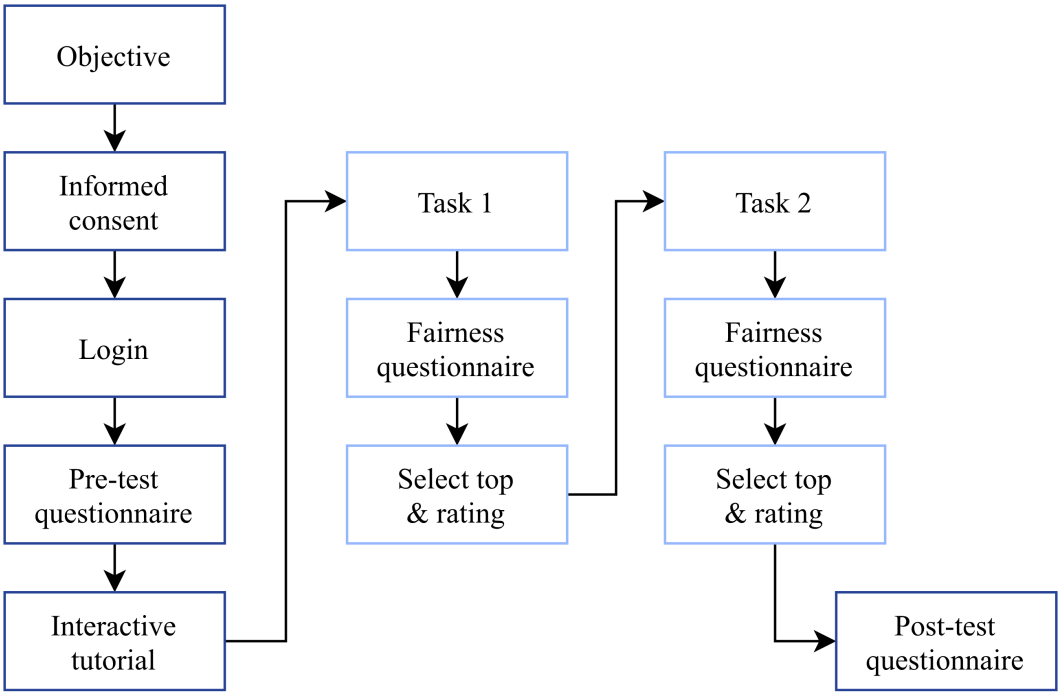


Figure A.1: Overview of the user study

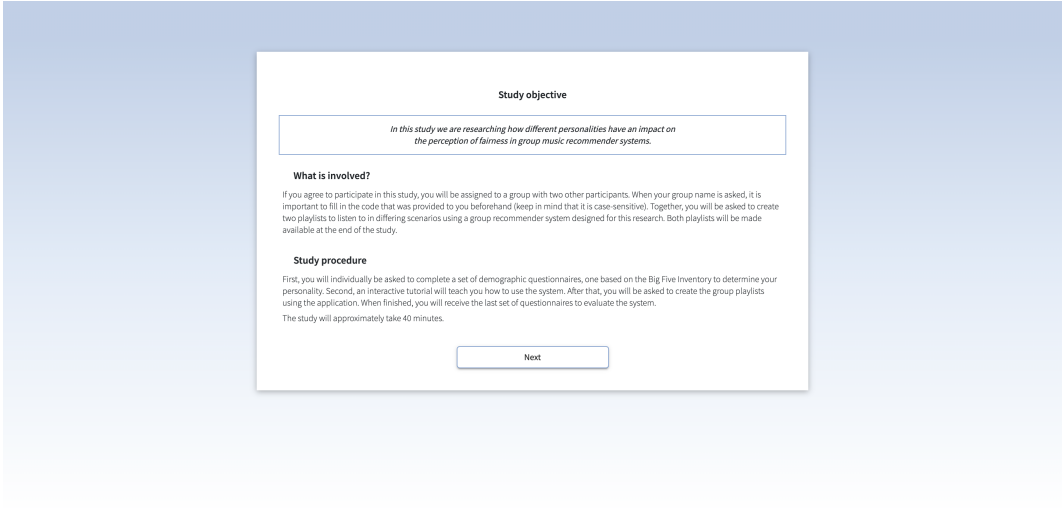
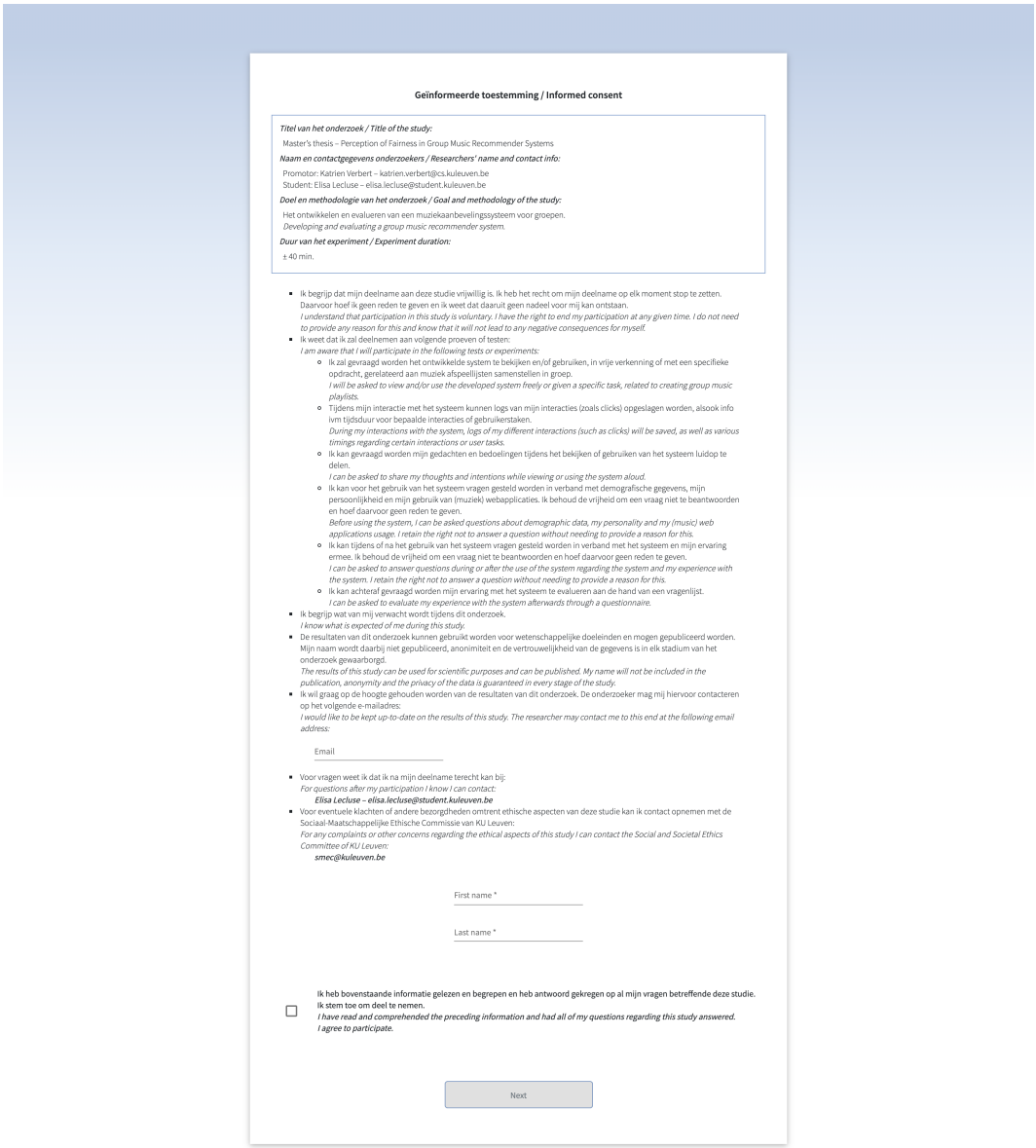


Figure A.2: Objective page



Geïnformeerde toestemming / Informed consent

Titel van het onderzoek / Title of the study:

Master's thesis – Perception of Fairness in Group Music Recommender Systems

Naam en contactgegevens onderzoekers / Researchers' name and contact info:

Promotor: Katrien Verbert – katrien.verbert@cs.kuleuven.be
Student: Elisa Lecluse – elisa.lecluse@student.kuleuven.be

Doel en methodologie van het onderzoek / Goal and methodology of the study:

Het ontwikkelen en evalueren van een muziekaanbevelingssysteem voor groepen.
Developing and evaluating a group music recommender system.

Duur van het experiment / Experiment duration:

± 40 min.

- Ik begrijp dat mijn deelname aan deze studie vrijwillig is. Ik heb het recht om mijn deelname op elk moment stop te zetten. Daarvoor hoef ik geen reden te geven en ik weet dat daaruit geen nadeel voor mij kan ontstaan.
  I understand that participation in this study is voluntary. I have the right to end my participation at any given time. I do not need to provide any reason for this and know that it will not lead to any negative consequences for myself.
- Ik weet dat ik zal deelnemen aan volgende proeven of testen:
  I am aware that I will participate in the following tests or experiments:
  - Ik zal gevraagd worden het ontwikkelde system te bekijken en/of gebruiken, in vrije verkenning of met een specifieke opdracht, gerelateerd aan muziek afspeellijsten samenstellen in groep.
    I will be asked to view and/or use the developed system freely or given a specific task, related to creating group music playlists.
  - Tijdens mijn interactie met het systeem kunnen logs van mijn interacties (zoals clicks) opgeslagen worden, alsook info ivm tijdsduur voor bepaalde interacties of gebruikerstaken.
    During my interactions with the system, logs of my different interactions (such as clicks) will be saved, as well as various timings regarding certain interactions or user tasks.
  - Ik kan gevraagd worden mijn gedachten en bedoelingen tijdens het bekijken of gebruiken van het systeem luidop te delen.
    I can be asked to share my thoughts and intentions while viewing or using the system aloud.
  - Ik kan voor het gebruik van het systeem vragen gesteld worden in verband met demografische gegevens, mijn persoonlijkheid en mijn gebruik van (muziek) webapplicaties. Ik behoud de vrijheid om een vraag niet te beantwoorden en hoef daarvoor geen reden te geven.
    Before using the system, I can be asked questions about demographic data, my personality and my (music) web applications usage. I retain the right not to answer a question without needing to provide a reason for this.
  - Ik kan tijdens of na het gebruik van het systeem vragen gesteld worden in verband met het systeem en mijn ervaring ermee. Ik behoud de vrijheid om een vraag niet te beantwoorden en hoef daarvoor geen reden te geven.
    I can be asked to answer questions during or after the use of the system regarding the system and my experience with the system. I retain the right not to answer a question without needing to provide a reason for this.
  - Ik kan achteraf gevraagd worden mijn ervaring met het systeem te evalueren aan de hand van een vragenlijst.
    I can be asked to evaluate my experience with the system afterwards through a questionnaire.
- Ik begrijp wat van mij verwacht wordt tijdens dit onderzoek.
  I know what is expected of me during this study.
- De resultaten van dit onderzoek kunnen gebruikt worden voor wetenschappelijke doeleinden en mogen gepubliceerd worden. Mijn naam wordt daarbij niet gepubliceerd, anonimiteit en de vertrouwelijkheid van de gegevens is in elk stadium van het onderzoek gewaarborgd.
  The results of this study can be used for scientific purposes and can be published. My name will not be included in the publication, anonymity and the privacy of the data is guaranteed in every stage of the study.
- Ik wil graag op de hoogte gehouden worden van de resultaten van dit onderzoek. De onderzoeker mag mij hiervoor contacteren op het volgende e-mailadres:
  I would like to be kept up-to-date on the results of this study. The researcher may contact me to this end at the following email address:

  Email _____

- Voor vragen weet ik dat ik na mijn deelname terecht kan bij:
  For questions after my participation I know I can contact:
  Elisa Lecluse – elisa.lecluse@student.kuleuven.be
- Voor eventuele klachten of andere bezorgdheden omtrent ethische aspecten van deze studie kan ik contact opnemen met de Sociaal-Maatschappelijke Ethische Commissie van KU Leuven:
  For any complaints or other concerns regarding the ethical aspects of this study I can contact the Social and Societal Ethics Committee of KU Leuven:
  smec@kuleuven.be

First name * _____

Last name * _____

☐ **Ik heb bovenstaande informatie gelezen en begrepen en heb antwoord gekregen op al mijn vragen betreffende deze studie. Ik stem toe om deel te nemen.**
I have read and comprehended the preceding information and had all of my questions regarding this study answered. I agree to participate.

Next

Figure A.3: Consent page

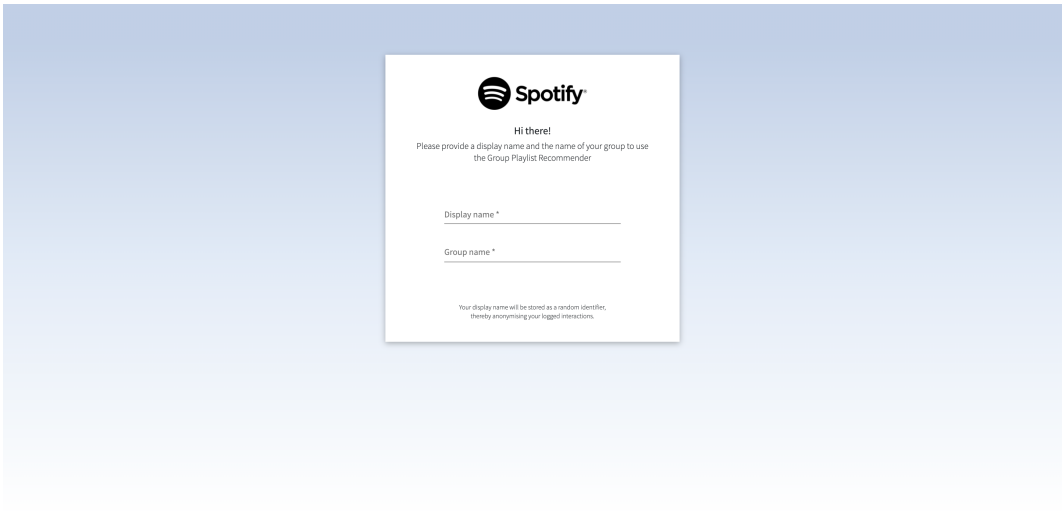


Figure A.4: Login page

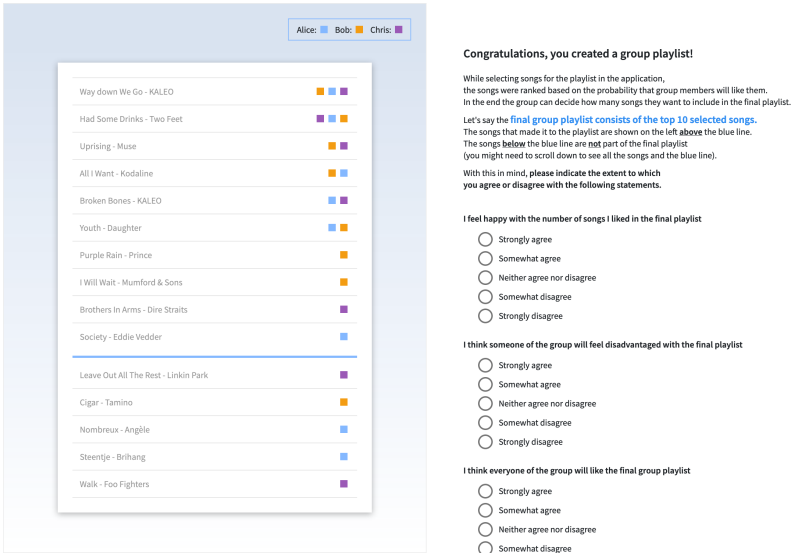


Figure A.5: Fairness questionnaire page

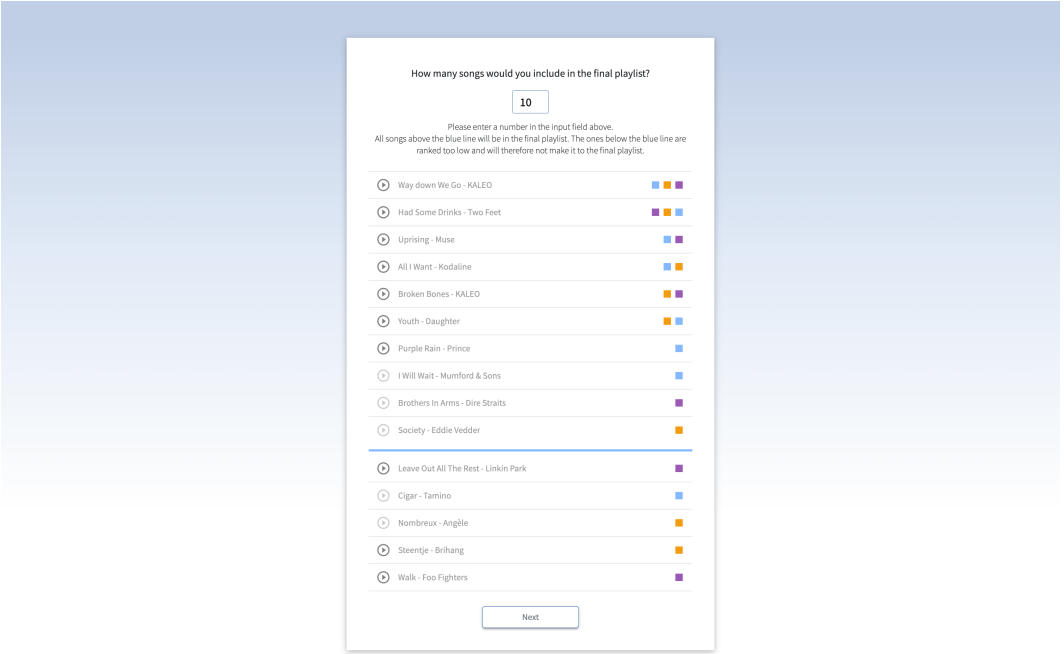


Figure A.6: Select-top page

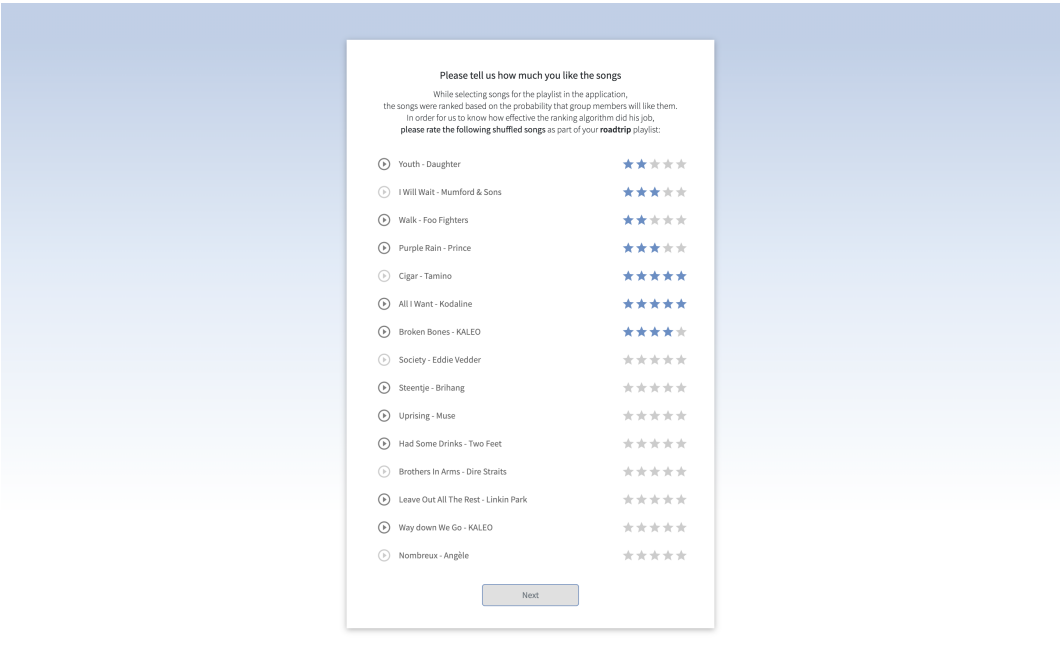


Figure A.7: Rating page

Appendix B

Interactive tutorial

Screenshots of the interactive tutorial are presented below.

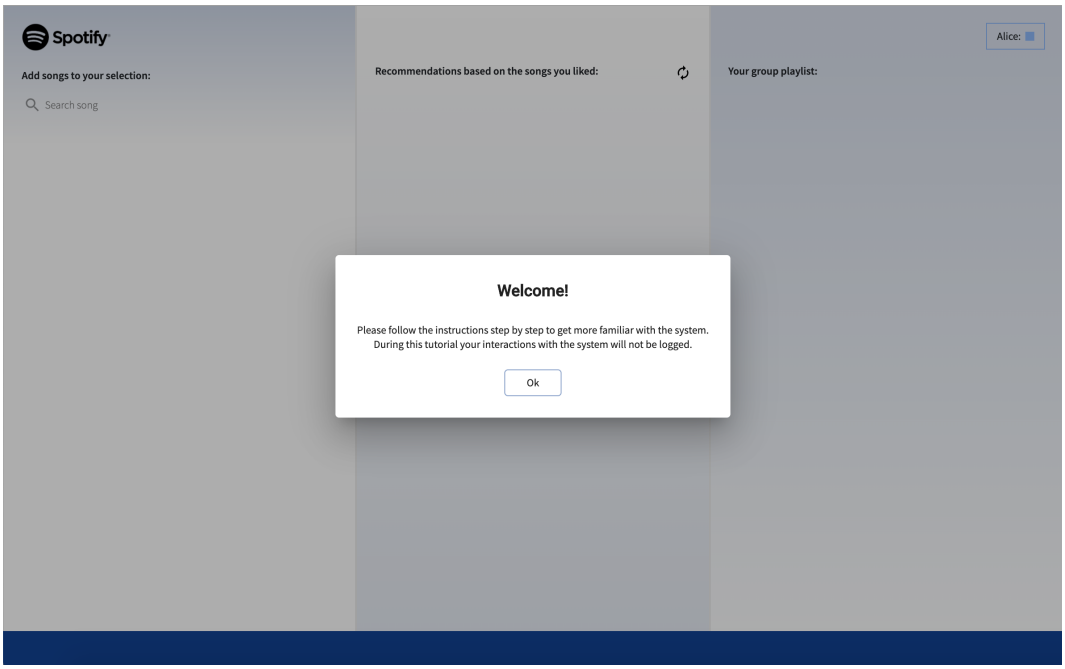


Figure B.1: Interactive tutorial: welcome

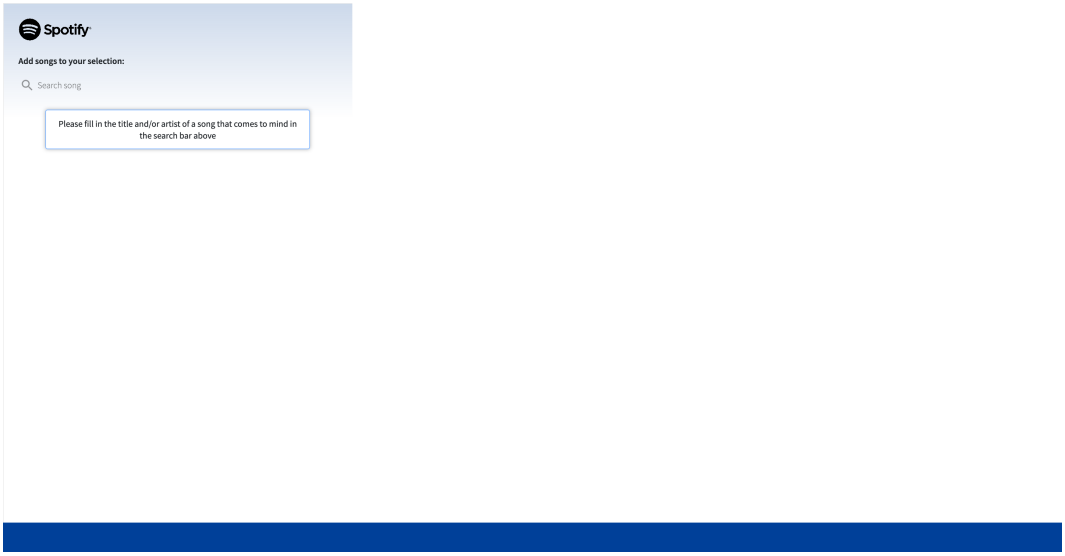


Figure B.2: Interactive tutorial: step 1

Figure B.3: Interactive tutorial: step 2

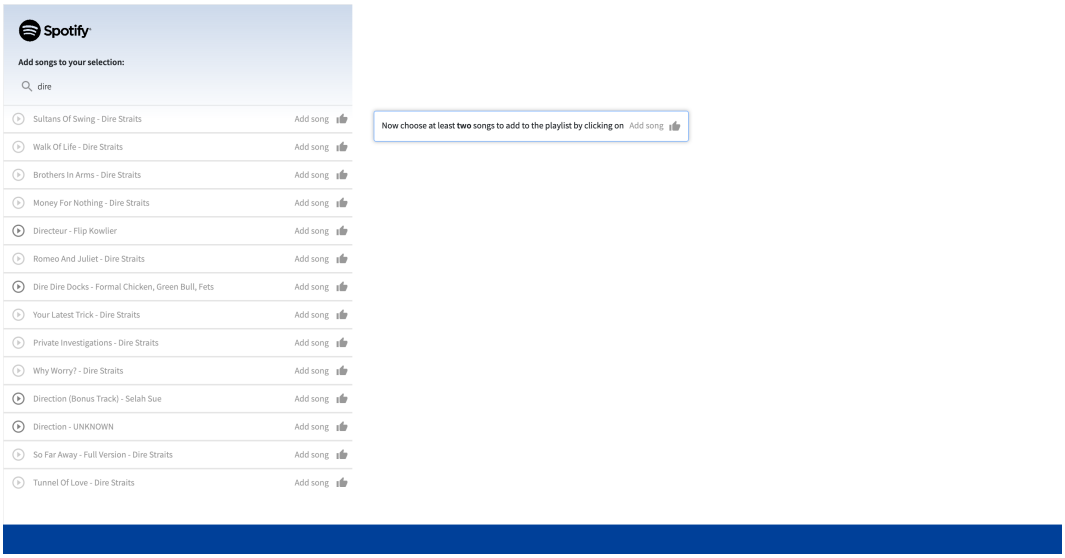


Figure B.4: Interactive tutorial: step 3

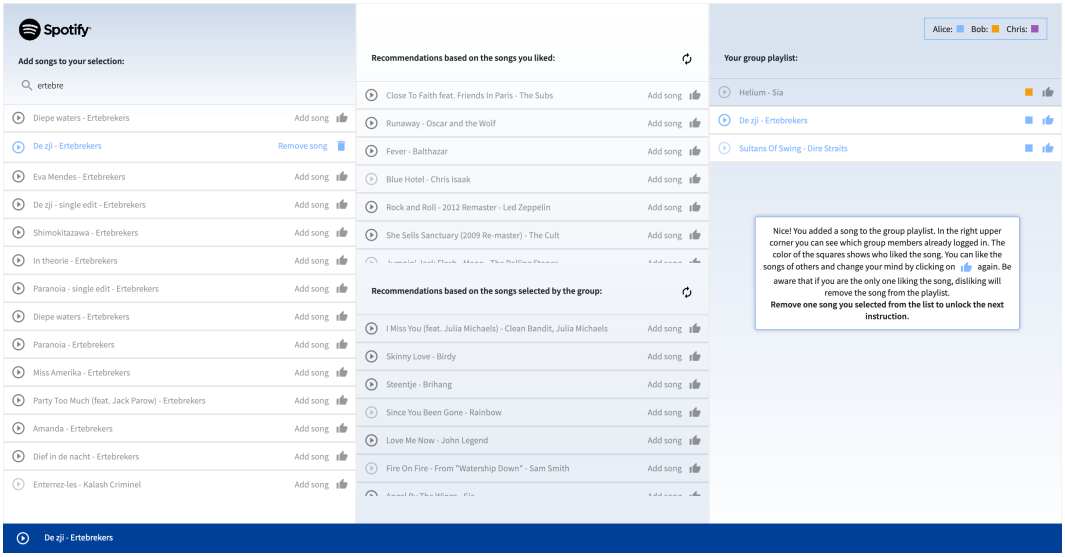


Figure B.5: Interactive tutorial: step 4

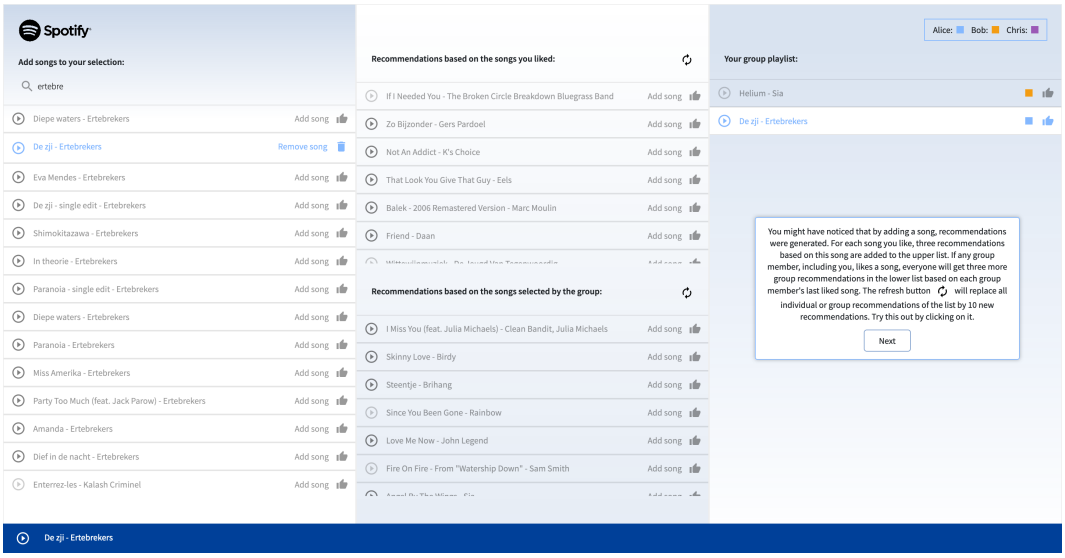


Figure B.6: Interactive tutorial: step 5 ('Next' appears when one of the refresh buttons is clicked)

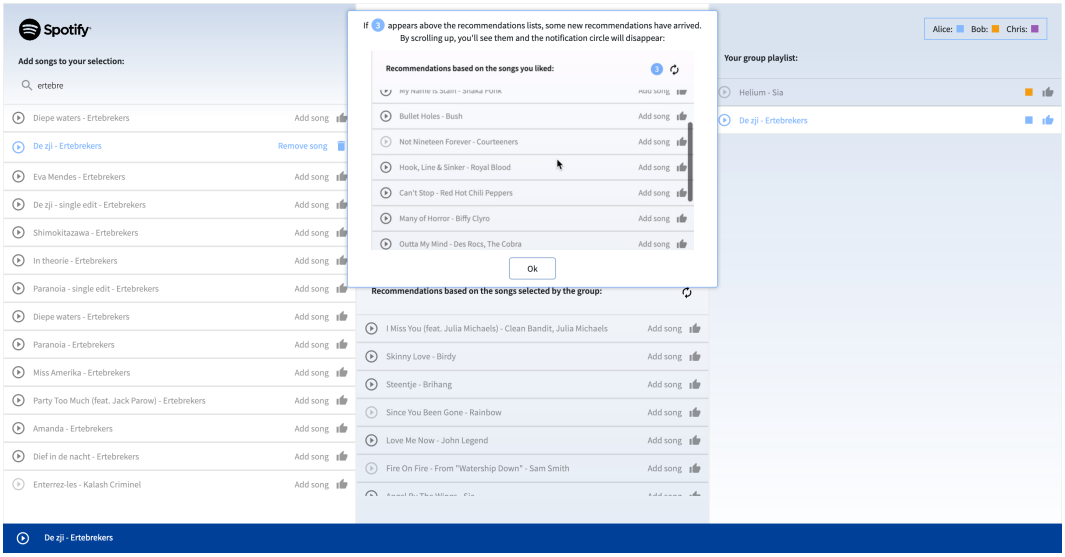


Figure B.7: Interactive tutorial: step 6 (a GIF shows how the new recommendations get highlighted when scrolling up)

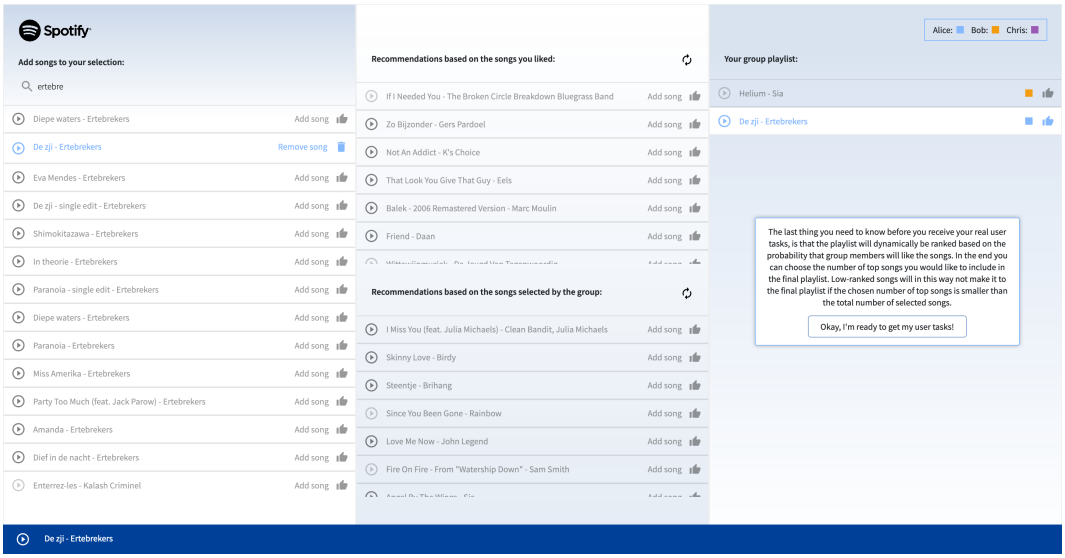


Figure B.8: Interactive tutorial: step 7

Appendix C

Questionnaires

The questionnaires used during the user study are listed below. Pre-test questionnaire includes the demographic questionnaire (question 6 and 7 from the Gold-MSI [38]) and the Big Five Inventory (BFI) [28, 31]. When the participants completed a task, the fairness questionnaire was given. The modified ResQue questionnaire [64] and the System Usability Scale (SUS) [62] belong to the post-test questionnaire.

Demographic Questionnaire

Participants will be asked to answer the following questions:

1. *What is your age?*
   - Younger than 18
   - 18 – 24
   - 25 – 34
   - 35 – 44
   - 45 – 54
   - Older than 55

2. *What is your gender?*
   - Female
   - Male
   - Other

3. *How often do you use Spotify?*
   - (Almost) never
   - 1 – 5 hours/week
   - 6 – 10 hours/week
   - 11 – 15 hours/week
   - 16 – 20 hours/week
   - More than 21 hours/week

4. *How often do you make or add music to playlists?*
   - (Almost) never
   - A few times per year
   - Once a month
   - Once a week
   - Almost every day

5. *How confident do you feel using (web) applications?*
   - I can easily use every (new) application
   - I know how to use most applications
   - I need some time to understand most applications
   - It is unclear to me how to use most applications

The following questions need to be answered with a 5-point Likert scale:
[Options: Disagree strongly, disagree a little, neither agree nor disagree, agree a little, agree strongly]

6. *I spend a lot of my free time doing music-related activities*

7. *I keep track of new music that I come across*

8. *I am open to new songs*

9. *I am open to genres that I usually don't listen to*

10. *It lasts a while before I start to like a new song*

11. *I regularly ask friends to recommend new music*

12. *I regularly listen to playlists of friends*

The Big Five Inventory (BFI)

Here are a number of characteristics that may or may not apply to you. For example, do you agree that you are someone who *likes to spend time with others*? Please write a number next to each statement to indicate the extent to which you agree or disagree with that statement.

I see myself as someone who…	Disagree strongly	Disagree a little	Neither agree nor disagree	Agree a little	Agree strongly
Is talkative					
Tends to find fault in others					
Does a thorough job					
Is depressed, blue					
Is original, comes up with new ideas					
Is reserved					
Is helpful and unselfish with others					
Can be somewhat careless					
Is relaxed, handles stress well					
Is curious about many different things					
Is full of energy					
Starts quarrels with others					
Is a reliable worker					
Can be tense					
Is ingenious, a deep thinker					
Generates a lot of enthusiasm					
Has a forgiving nature					
Tends to be disorganized					
Worries a lot					
Has an active imagination					
Tends to be quiet					
Is generally trusting					
Tends to be lazy					
Is emotionally stable, not easily upset					
Is inventive					
Has an assertive personality					
Can be cold and aloof					
Perseveres until the task is finished					
Can be moody					
Values artistic, aesthetic experiences					
Is sometimes shy, inhibited					
Is considerate and kind to almost everyone					
Does things efficiently					
Remains calm in tense situations					
Prefers work that is routine					
Is outgoing, sociable					

Is sometimes rude to others					
Makes plans and follows through with them					
Gets nervous easily					
Likes to reflect, play with ideas					
Has few artistic interests					
Likes to cooperate with others					
Is easily distracted					
Is sophisticated in art, music, or literature					

Fairness Questionnaire

The following questions need to be answered with a 5-point Likert scale:
[Options: Disagree strongly, disagree a little, neither agree nor disagree, agree a little, agree strongly]
[X = 2/3 of the songs selected by the group]

1. *I feel happy with the number of songs I liked in the top X playlist*
2. *I think someone of the group will feel disadvantaged with the top X playlist*
3. *I think everyone of the group will like the top X group playlist*
4. *I like the top X group playlist*
5. *I feel like the ranking algorithm is fair (i.e. every group member has some songs they like in the top X playlist)*
6. *I feel like the songs are evenly ranked regarding who liked them (i.e. every group member has a song he/she liked highly ranked)*
7. *I would play the top X group playlist with the others of the group*
8. *I would play the top X group playlist even when the others of the group are not present*
9. *I would play the full group playlist with the others of the group*
10. *I would play the full group playlist even when the others of the group are not present*
11. *I think this is a good application to make a fair group playlist*
12. *Fairness is important in group playlists*
13. *It is more important that a majority of the group likes the playlist than that someone's songs are left out*
14. *I want at least some of my songs to be in the playlist, even if no one likes them*

The following question is open-ended

15. *Do you have any suggestions to make the ranking of songs in the playlist more fair?*

Modified ResQue Questionnaire

	Disagree strongly	Disagree a little	Neither agree nor disagree	Agree a little	Agree strongly
This recommender system gave me good suggestions (Recommendation accuracy)					
This recommender system helped me discover new songs (Recommendation novelty)					
The songs recommended to me are diverse (Recommendation Diversity)					
This recommender system allows me to tell what I like (Interaction Adequacy)					
The components of this recommender system are clear to me (Interface Adequacy)					
I found it easy to make this recommender system recommend new songs to me (Perceived Ease of Use)					
I became familiar with this recommender system very quickly (Ease of Initial Learning)					
This recommender system helped me find good songs (Perceived Usefulness)					
This recommender system influenced my selection of songs (Perceived Usefulness)					
I will use this recommender system to create group playlists (Behavioral Intentions: Intention to Use the System)					
I will use this recommender system frequently (Behavioral Intentions: Continuance and Frequency)					

The following question is open-ended

Do you have any suggestions to make a better application to make group playlists?

System Usability Scale (SUS)

	Disagree strongly	Disagree a little	Neither agree nor disagree	Agree a little	Agree strongly
I think that I would like to use this system frequently					
I found the system unnecessarily complex					
I thought the system was easy to use					
I think that I would need the support of a technical person to be able to use this system					
I found that the various functions in this system were well integrated					
I thought that there was too much inconsistency in this system					
I would imagine that most people would learn to use this system very quickly.					
I found the system very cumbersome to use					
I felt very confident using the system					
I needed to learn a lot of things before I could get going with this system					

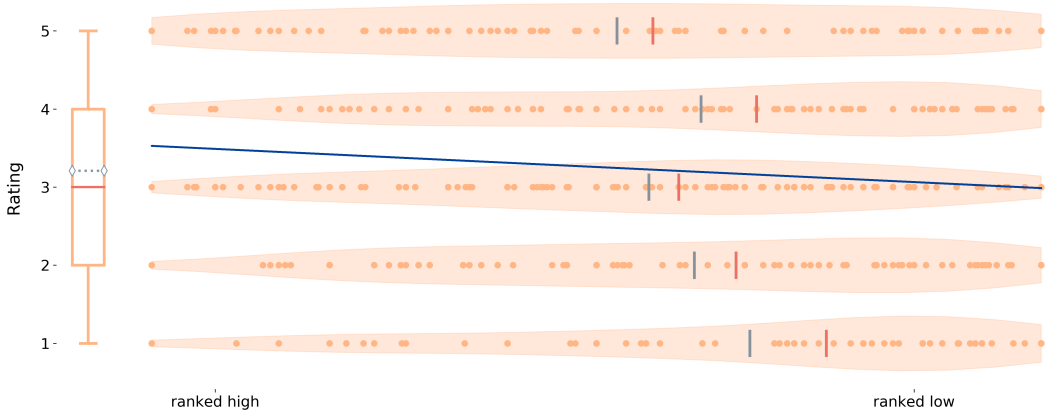
The following questions are open-ended

1. *What aspects of the application were not clear to you?*
2. *Do you have any suggestions to make the application more easily usable?*

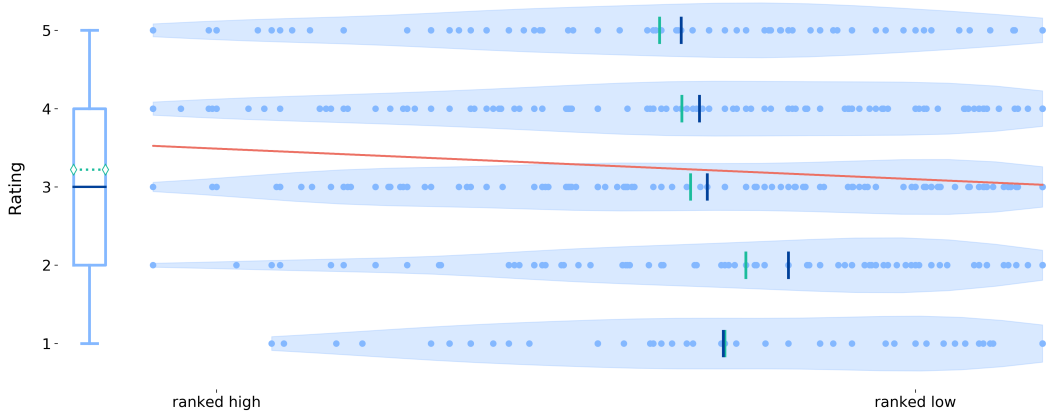
Appendix D

Extra results

This appendix shows some extra plots to analyse the quality of the two ranking algorithms. Figure D.1 shows scatter plots and its trend lines of the given rating and position of the songs not selected/liked by the participant giving the rating in the playlist for both version 1 and 2. Violin plots per rating show the song's position density. The box plots show the distribution of the ratings. The slope of the trend lines are -0.039 and -0.036 for version 1 and 2 respectively.



(a) Version 1



(b) Version 2

Figure D.1: Scatter plot of the rating and position of the songs not selected/liked by the participant giving the rating (the higher the rating, the more the participant likes the song); trend line of the scatter plot (blue and red); violin plots of the song's position density per rating (mean: grey and green line; median: red and dark blue line); box plot of the given ratings distribution (mean: grey and green diamonds; median: red and dark blue line);

Bibliography

[1] Alexander Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalčič. *Group recommender systems: An introduction*. Springer, 2018.

[2] Alexander Felfernig, Müslüm Atas, Martin Stettinger, Thi Ngoc Trang Tran, and Stefan Reiterer. Group recommender applications. In Alexander Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalčič, editors, *Group recommendeer systems: An introduction*, pages 75–89. Springer, 2018.

[3] Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. Recommending packages to groups. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 449–458. IEEE, 2016.

[4] Alexander Felfernig, Müslüm Atas, Denis Helic, Thi Ngoc Trang Tran, Martin Stettinger, and Samer Ralph. Algorithms for group recommendation. In Alexander Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalčič, editors, *Group recommendeer systems: An introduction*, pages 27–58. Springer, 2018.

[5] Marko Tkalčič, Delić Amra, and Alexander Felfernig. Personality, emotions, and group dynamics. In Alexander Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalčič, editors, *Group recommendeer systems: An introduction*, pages 157–167. Springer, 2018.

[6] Maria Stratigi, Jyrki Nummenmaa, Evaggelia Pitoura, and Kostas Stefanidis. Fair sequential group recommendations. In *Proceedings of the 35th ACM/SIGAPP Symposium on Applied Computing, SAC*, 2020.

[7] Thi Ngoc Trang Tran, Müslüm Atas, Alexander Felfernig, Viet Man Le, Ralph Samer, and Martin Stettinger. Towards social choice-based explanations in group recommender systems. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, pages 13–21, 2019.

[8] Charu C. Aggarwal. *Recommender systems*, volume 1. Springer, 2016.

[9] Anthony Jameson and Barry Smyth. Recommendation to groups. In *The adaptive web*, pages 596–627. Springer, 2007.

[10] Judith Masthoff. Group recommender systems: aggregation, satisfaction and group attributes. In *Recommender Systems Handbook*, pages 743–776. Springer, 2015.

[11] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems: introduction and challenges. In *Recommender Systems Handbook*, pages 1–34. Springer, 2015.

[12] Iván Palomares, Francisco J Estrella, Luis Martínez, and Francisco Herrera. Consensus under a fuzzy context: Taxonomy, analysis framework afryca and experimental case of study. *Information Fusion*, 20:252–271, 2014.

[13] Jorge Castro, Francisco J. Quesada, Iván Palomares, and Luis Martínez. A consensus-driven group recommender system. *International Journal of Intelligent Systems*, 30(8):887–906, 2015.

[14] Narges Mahyar, Weichen Liu, Sijia Xiao, Jacob Browne, Ming Yang, and Steven P. Dow. Consesnsus: Visualizing points of disagreement for multi-criteria collaborative decision making. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 17–20, 2017.

[15] T. Nguyen and Francesco Ricci. Supporting group decision making with recommendations and explanations. 2016.

[16] Kevin McCarthy, Maria Salamó, Lorcan Coyle, Lorraine McGinty, Barry Smyth, and Paddy Nixon. CATS: A synchronous approach to collaborative group recommendation. In *Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pages 86–91, 2006.

[17] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. Fairness-aware group recommendation with pareto-efficiency. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 107–115, 2017.

[18] B. Demoen and T. Schrijvers. *Fundamentals for Computer Science - Course Text*. Cursusdienst VTK vzw, 2019.

[19] Wikipedia. Travelling salesman problem. URL: https://en.wikipedia.org/wiki/Travelling_salesman_problem, last accessed 28-05-2020.

[20] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. Fairness in package-to-group recommendations. In *Proceedings of the 26th International Conference on World Wide Web*, pages 371–379, 2017.

[21] Robin Burke. Multisided fairness for recommendation. *arXiv preprint arXiv:1707.00093*, 2017.

[22] Joseph F. McCarthy and Theodore D. Anagnost. MusicFX: an arbiter of group preferences for computer supported collaborative workouts. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 363–372, 1998.

[23] Andrew Crossen, Jay Budzik, and Kristian J. Hammond. Flytrap: intelligent group music recommendation. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 184–185, 2002.

[24] Kenton O'Hara, Matthew Lipson, Marcel Jansen, Axel Unger, Huw Jeffries, and Peter Macer. Jukola: democratic music choice in a public space. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 145–154, 2004.

[25] Dennis L. Chao, Justin Balthrop, and Stephanie Forrest. Adaptive radio: achieving consensus using negative preferences. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 120–123, 2005.

[26] George Popescu. Designing a voting mechanism in the groupfun music recommender system. In *International Conference on Human-Computer Interaction*, pages 383–386. Springer, 2013.

[27] George Popescu and Pearl Pu. What's the best music you have? designing music recommendation for group enjoyment in groupfun. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pages 1673–1678, 2012.

[28] O. P. John, L. P. Naumann, and C. J. Soto. Paradigm shift to the integrative Big Five trait taxonomy: History, measurement, and conceptual issues. In O. P. John, R. W. Robins, and L. A. Pervin, editors, *Handbook of personality: Theory and research*, pages 114–158. Guilford Press, New York, 2008.

[29] Paul T. Costa and Robert R. McCrea. *Revised NEO personality inventory (NEO PI-R) and NEO five-factor inventory (NEO-FFI).* Psychological Assessment Resources, 1992.

[30] O. P. John and S. Srivastava. The Big Five trait taxonomy: History, measurement, and theoretical perspectives. In L. A. Pervin and O. P. John, editors, *Handbook of personality: Theory and research*, pages 102–138. Guilford Press, New York, 1999.

[31] Oliver P. John, Eileen M. Donahue, and Robert L. Kentle. The big five inventory–versions 4a and 54. *Berkeley, CA: University of California, Berkeley, Institute of Personality and Social Research*, 1991.

[32] L. R. Goldberg. The development of markers for the big-five factor structure. *Psychological assessment*, 4:26–42, 1992.

[33] Christopher J. Soto and Oliver P. John. Ten facet scales for the big five inventory: Convergence with neo pi-r facets, self-peer agreement, and discriminant validity. *Journal of research in personality*, 43(1):84–90, 2009.

[34] Tianwei V. Du, Alison E. Yardley, and Katherine M Thomas. Mapping big five personality traits within and across domains of interpersonal functioning. *Assessment*, page 1073191120913952, 2020.

[35] Thomas Schäfer and Claudia Mehlhorn. Can personality traits predict musical style preferences? a meta-analysis. *Personality and Individual Differences*, 116:265–273, 2017.

[36] David M. Greenberg, Michal Kosinski, David J. Stillwell, Brian L. Monteiro, Daniel J. Levitin, and Peter J. Rentfrow. The song is you: Preferences for musical attribute dimensions reflect personality. *Social Psychological and Personality Science*, 7(6):597–605, 2016.

[37] Bruce Ferwerda, Emily Yang, Markus Schedl, and Marko Tkalcic. Personality traits predict music taxonomy preferences. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 2241–2246, 2015.

[38] Daniel Müllensiefen, Bruno Gingras, Jason Musil, and Lauren Stewart. The musicality of non-musicians: an index for assessing musical sophistication in the general population. *PloS one*, 2014.

[39] Jonna K. Vuoskoski and Tuomas Eerola. The role of mood and personality in the perception of emotions represented by music. *Cortex*, 47(9):1099–1106, 2011.

[40] Martijn Millecamp, Nyi Nyi Htun, Cristina Conati, and Katrien Verbert. To explain or not to explain: the effects of personal characteristics when explaining music recommendations. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 397–407, 2019.

[41] John O'Donovan, Barry Smyth, Brynjar Gretarsson, Svetlin Bostandjiev, and Tobias Höllerer. Peerchooser: visual interactive recommendation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1085–1088, 2008.

[42] Spotify for Developers. Get audio features for a track. URL: developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features, last accessed 18-05-2020.

[43] Martijn Millecamp, Nyi Nyi Htun, Yucheng Jin, and Katrien Verbert. Controlling spotify recommendations: effects of personal characteristics on music recommender user interfaces. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, pages 101–109, 2018.

[44] Wikipedia. Force-directed graph drawing. URL: https://en.wikipedia.org/wiki/Force-directed_graph_drawing, last accessed 08-05-2020.

[45] Raquel Benbunan-Fich. Using protocol analysis to evaluate the usability of a commercial web site. *Information & management*, 39(2):151–163, 2001.

[46] Erica L. Olmsted-Hawala, Elizabeth D. Murphy, Sam Hawala, and Kathleen T. Ashenfelter. Think-aloud protocols: a comparison of three think-aloud protocols for use in testing data-dissemination web sites for usability. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2381–2390, 2010.

[47] Matthew Field. The best music streaming services: Apple Music, Spotify, YouTube Music and Amazon Music compared. URL: https://www.telegraph.co.uk/technology/0/best-music-streaming-services-apple-music-spotify-amazon-music, last accessed 16-05-2020.

[48] Mansoor Iqbal. Spotify usage and revenue statistics (2020). URL: https://www.businessofapps.com/data/spotify-statistic, last accessed 16-05-2020.

[49] Kurt Jacobson, Vidhya Murali, Edward Newett, Brian Whitman, and Romain Yon. Music personalization at Spotify. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 373–373, 2016.

[50] Ronak Patel. Why choose mean stack for your web & mobile app development projects? URL: https://medium.com/@ronak8036/why-mean-stack-ec42aa82818, last accessed 17-05-2020.

[51] IBM Cloud Education. Is MEAN right for you? Learn why this end-to-end stack of MongoDB, Express.js, AngularJS and Node.js is gaining popularity for modern web app development. URL: https://www.ibm.com/cloud/learn/mean-stack-explained, last accessed 17-05-2020.

[52] Evince. [Updated] MEAN Stack Architecture: MongoDB, ExpressJS, AngularJS, and NodeJS. URL: https://evincedev.com/blog/mean-stack-architecture, last accessed 16-05-2020.

[53] MongoDB. MongoDB vs. MySQL. URL: https://www.mongodb.com/compare/mongodb-mysql, last accessed 17-05-2020.

[54] Shanal Aggarwal. Angular vs AngularJS - A Complete Comparison Guide 2019. URL: https://www.techaheadcorp.com/blog/angular-vs-angularjs, last accessed 17-05-2020.

[55] Matt Ratto, R. Benjamin Shapiro, Tan Minh Truong, and William G. Griswold. The activeclass project: Experiments in encouraging classroom participation.

In *Designing for change in networked learning environments*, pages 477–486. Springer, 2003.

[56] Carl Gutwin and Saul Greenberg. Workspace awareness for groupware. In *Conference Companion on Human Factors in Computing Systems*, pages 208–209, 1996.

[57] Carl Gutwin and Saul Greenberg. A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work (CSCW)*, 11(3-4):411–446, 2002.

[58] Nicola Yuill and Yvonne Rogers. Mechanisms for collaboration: A design and evaluation framework for multi-user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 19(1):1–25, 2012.

[59] Meredith Ringel Morris and Eric Horvitz. Searchtogether: an interface for collaborative web search. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 3–12, 2007.

[60] Claudia Iacob. Identifying, relating, and evaluating design patterns for the design of software for synchronous collaboration. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, pages 323–326, 2011.

[61] Socket.IO. What Socket.IO is. URL: https://socket.io/docs, last accessed 19-05-2020.

[62] John Brooke et al. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[63] Aaron Bangor, Philip T. Kortum, and James T. Miller. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594, 2008.

[64] Pearl Pu, Li Chen, and Rong Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 157–164, 2011.

Fiche masterproef

Student: Elisa Lecluse

Titel: Perception of Fairness in Group Music Recommender Systems

UDC: 621.3

Korte inhoud:

Fairness is an important aspect arising in group recommender systems. Especially if a set of items is recommended to a group, the system should try to balance the user utilities inside the group. In group music recommender systems fairness ensures that even the songs of users with music taste diverging from the group are included in the recommended playlist. This research is unique as it is the first to consider fairness in group music recommender systems. After a few design stages, the objective of this research was defined. It focuses in particular on the influence of personality on the perceived fairness, the influence of the ranking algorithm on the perceived fairness and the overall usability of the system. A web application was developed to assist a group of users in creating a playlist together. A user can search songs and for every newly added song individual and group recommendations are provided using the Spotify Web API. All selected songs are ranked based on the probability that other group members will like them. The lowest ranked songs are not liked by the others and will thus not make it to the final playlist. In this fashion, the system can recommend a set of songs to the group that satisfies most group members. Here, the question can be asked if the users perceive this method as fair. Two versions of the ranking algorithm were integrated in the system: one based on the time the song was selected (time-based version) and one related to content-based filtering and aggregated predictions (dissimilarity-based version). Both algorithms first rank the playlist based on the number of likes (votes) a song received. 45 participants were recruited to conduct a within-subjects experiment in groups of 3. The results show that the dissimilarity-based version performs significantly better than the time-based version in terms of perceived fairness, but not in terms of predicting the popularity of the song. The participant's song ratings indicate that only the voting mechanism effectively ranks the songs. The responses to the fairness questionnaire were highly dependent on the number of songs each individual got included in the final playlist over the number of songs this individual selected. This makes it difficult to draw conclusions related to the influence of personality. Finally, the application was enthusiastically received by the participants and the overall usability scored excellently high.

Thesis voorgedragen tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: computerwetenschappen, hoofdoptie Mens-machine communicatie

Promotor: Prof. dr. K. Verbert

Assessoren: Dr. Y. Dauxais
Ir. P. Bartels

Begeleider: Dr. N. N. Htun