NORTHERN ARIZONA UNIVERSITY
College of Engineering, Forestry & Natural Sciences

SCHOOL OF INFORMATICS, COMPUTING, AND CYBER SYSTEMS
CS 386 – Software Engineering
Prof. Marco Gerosa
Team Project – D.2 Requirements

1. Positioning

1.1. Problem statement

The problem of not having a music player that satisfies all the needs of the modern user affects populations of various taste and age; the impact of which is not having a reliable music streamer and a place to listen to the music of their choice.

1.2. Product Position Statement

For music lovers of all age who are struggling to find a centralized place to put all their music, *MooD* music streaming app that allows the users to play music based on their current mood, ability to create and modify easily and have an easy way to download their songs unlike Spotify or Pandora our product will not rely on premium plans to let the user create playlists or listen to music without having to be interrupted.

1.3. Value proposition

MooD is a music streaming app that allows music listeners to listen to music effectively and with ease in one place. Our music app allows listening to music in various ways; from selecting a mood, or playing a premade playlist to listening to one's own playlist.
Consumer Segment:

2. Stakeholders

Make a list of all stakeholders of the project with a brief description of each one of them, emphasizing any responsibilities with the project. Examples of stakeholders include users, clients, competitors, detractors, developers, etc.

Project Manager - Turan Naimey

Responsible for assigning responsibilities, and managing deadlines and resources to ensure timely completion of assigned projects.

Application Developers - Karsten Nguyen, Austin Collins, Austin Torrence, & Michael Evans

Develop the necessary specifications for software. Developers may also test, debug and improve generally faulty applications for clients.

Testers/Users - Rosalie Nguyen, Max Waltz, & Nick Bury

The uses will test and use the application software developed for its functionality, usability and consistency. They will hopefully give us useable input on the applications, so that will can make it as user friendly as we can.

Competitors - Spotify, Soundcloud, & Pandora

Other applications that are similar to our music application. We will based our platforms to theres, to see the difference and resemblance of the platforms.

3. Functional requirements

Make a numbered list of requirements for your software. Just self-explanatory titles are enough at this point. Avoid design. Keep requirements at a general level. Focus on capabilities needed and why (not how) they should be implemented.

1.) Moods or Genres are entered or picked through the system by the user
2.) Only the user can enter the data into the system
3.) The operation will then play certain tracks or playlist accordingly onto the screen
4.) If the user likes the selected track, they may find more information on them (Tour dates, Albums)
5.) They may look at the upcoming tracks but can not alter the playlist.
6.) But users can create their own playlist, with an add to playlist button
7.) Users may be able to skip certain tracks
8.) Product may be used through a website or mobile device
9.) Must be downloadable

These functions must be implemented so that the application can be able used throughout all platforms and user friendly as possible. If are the functions are implemented correctly, then we can steer people away from the current competitors.

4. Non-functional requirements

Make a numbered list of non-functional requirements that are important for your software. Follow the terminology of ISO/IEC 25010:2011. For each non-functional requirement, give an objective goal/measurement in order to provide verifiability for the requirement.

1. Resource Utilization - To be able to search for a database or station that contains the mood of the music the user selected.
2. Compatibility - The ability to be used across iOS, Android, and websites alike.
3. Co-existence - Being able to function properly with other applications and processes.
4. Interoperability - Effectively share music in a presentable manner that does not conflict with a station or database being accessed.
5. Usability - Produce the mood of music specified by the user to fulfill the users needs.
6. Operability - Make it easy for the user to operate the application in all aspects being searching, downloading, creating a playlist, etc.
7. User Interface Aesthetics - Make the applications running form aesthetically appeal to the mind of the user to produce a more enjoyable and comforting experience.
8. Accessibility - Ability to please all users by producing any genre of music based on the mood specified.
9. Reliability - Play music from stations and databases that fits the mood selected for as long as the user needs.
10. Availability - Use of the application at anytime and anyplace dictated by the user.

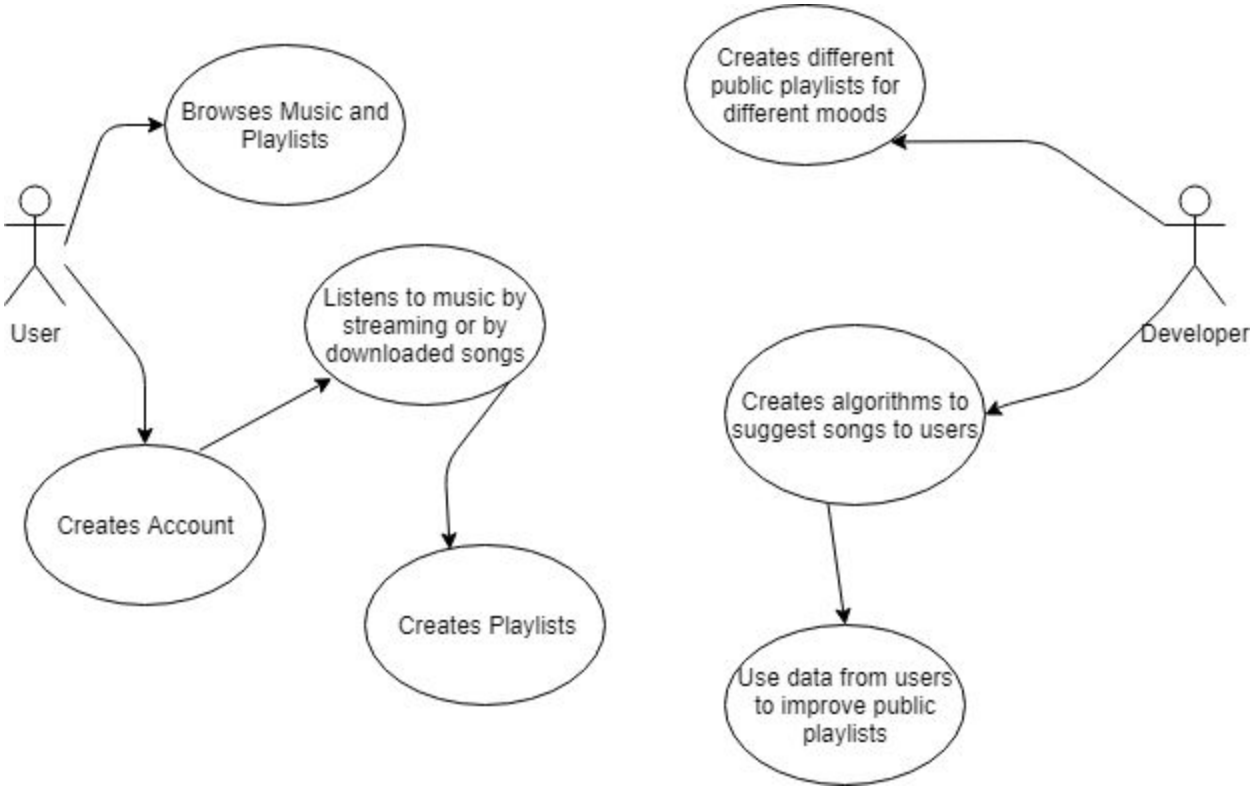
5. MVP

Present the strategy that you will use for developing a MVP. Which features are you going to validate? How?

Designing a minimum viable product for a mood music player will require the most basic of functionality that still satisfies the requirements while not being in its final form. To start, the application will have an icon and base color to represent its beta testing, so to speak. Upon opening the application, the user should be able to use the promised features of selecting a mood, searching and downloading music, streaming music, and creating a playlist for a specific mood. This will be done by having selectable fields of the different features that will direct the user to the respectable destination. In addition, the MVP should include play, pause, next, back, repeat all songs, repeat one song, don't repeat, shuffle, and skip buttons. These buttons will be placed in specific locations of the screen and will have smaller fields. It will also have a bland look to start as functionality success will be the focus point and not aesthetics until testing is complete. It should also be able to play quality sounding music from the given features to verify the song has been downloaded or streamed correctly. The song currently playing will be displayed on screen with its name, artist or band, and its respectable album if applicable.

6. Use cases

6.1. Use case diagram



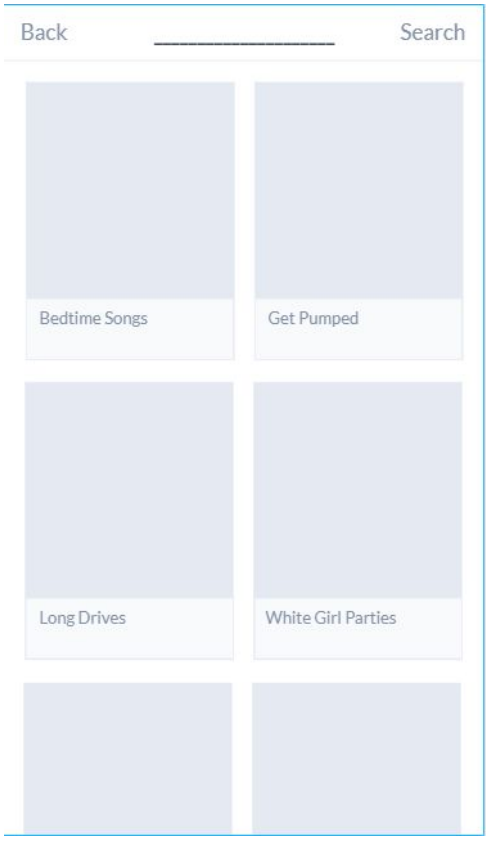
6.2. Use case descriptions

Use Case: Browsing Music
Actor: User
Description: The user browses through playlists or searches for specific songs
Pre-Conditions: The user has an account
Post-Conditions: The now knows about more music than they did before
Main Flow:

1. The user accesses our premade playlists
2. The user can stream the playlist as a whole or single out a song
3. The user can scroll through bunches of premade playlists to fit their mood

Alternative Flows:

1. Users can also use the search bar to search for a specific song, artist/group, album, or genre
2. They will be able to sift through results on a page

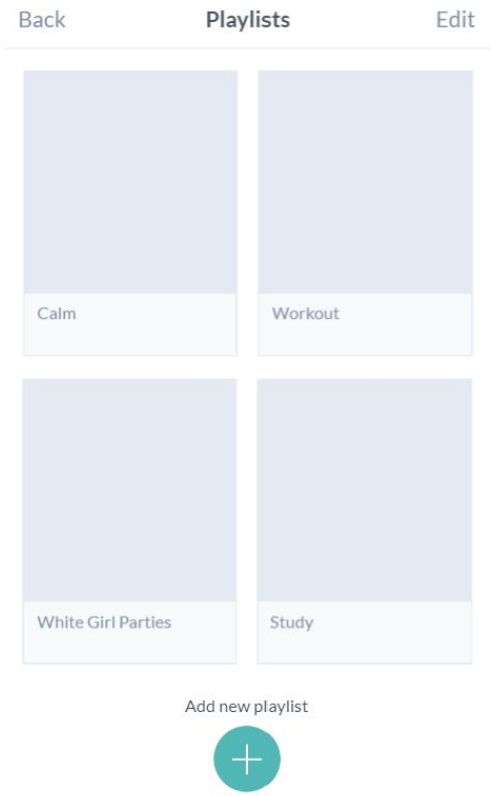


Use Case: Creating a Playlist
Actor: User
Description: The user creates a playlist to fit their mood
Pre-Conditions: The user has an account to save their music
Post-Conditions: The user now has a playlist to their liking
Main Flow:

1. The user can browse through public playlists for certain moods
2. The user can also search for specific songs, albums, or artists/groups
3. The user then creates an empty playlist with a Title and an optional description
4. The user then adds these songs found while browsing to such playlist
5. If there is a certain mood/genre to the playlist, algorithms developed by developers will suggest songs or artists
6. The user can then add suggested songs to their playlists
7. The user can add or delete songs from the playlist at any time
8. Playlists can then be streamed through the internet or downloaded for offline listening

Alternative Flows:

1. The user creates a playlist and does not add anything to it

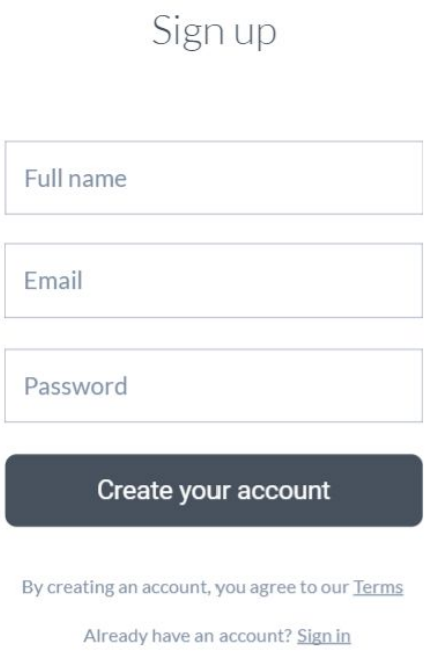


Use Case: Creating an Account
Actor: User
Description: The user creates an account to save music and playlists
Pre-Conditions: The user has an email address and a device capable of downloading our app
Post-Conditions: The user now has an account to save and browse music
Main Flow:

1. The user downloads the app on a capable platform
2. The user opens the app and selects Create Account
3. The user then enters a valid email address and password
4. The user then creates the account
5. The user will then go to their email to register their account by clicking on a link which will activate their account
6. The user can then use their email and password to log into the app

Alternative Flows:

1. The user doesn't register account and cannot log in



Use Case: Creating Mood Playlists
Actor: Developer
Description: The developer creates playlists for certain moods
Pre-Conditions: The developer has data on their users and what they listen to
Post-Conditions: Users can now access these public playlists
Main Flow:

1. The developer collects data on types music their users listen to together
2. The developer then must use a combination of data as well as their own judgement (most likely judgement from other developers as well) to create playlists that fit certain moods
3. The developers then add these songs to these playlists to fit these moods
4. These moods might include things such as “Get Pumped” or “Bedtime Songs”
5. The developers will then publish these playlists so that users can access them
6. The developers will continually collect data on whether or not users like certain playlists and take feedback into account to continually edit these mood playlists

Alternative Flows:

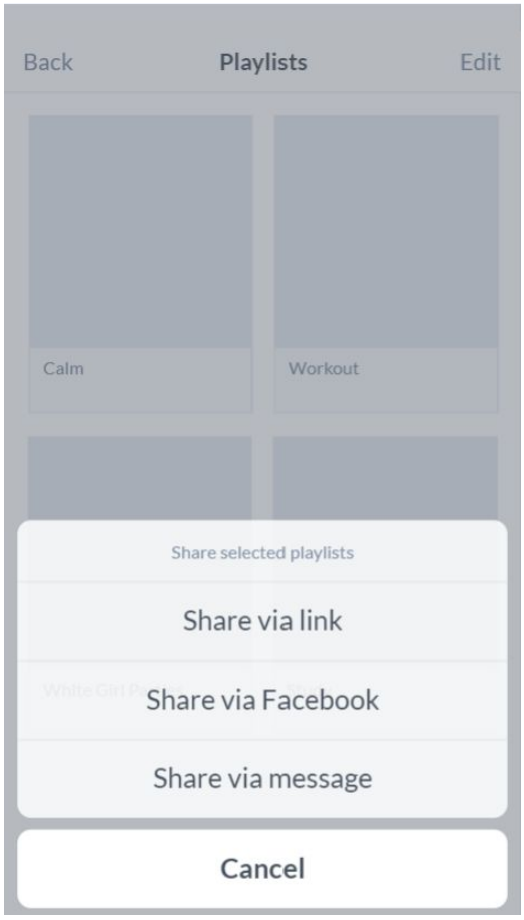
1. No one listens to the mood playlist
2. Developer would then edit and re-release playlist to try and get better result

Use Case: Sharing Music
Actor: User
Description: Sharing your playlists or liked playlists with others
Pre-Conditions: Having an account and optional a playlist
Post-Conditions: Sharing music with people to discover more music
Main Flow:

1. The user will like a playlist or create a playlist
2. Once playlist or liked playlists are available you can select share
3. Once you share you will have a variety of options to share
    a. This would include options such as copy link, share via messages, email, etc
4. You can select one of these options to share a link to the playlist to anyone you want
5. The person receiving the link can open the link which will automatically open the app to the playlist linked

Alternative Flows:

1. Users can also just tell people in person the name of public playlists to look up4



7. User stories

As a developer I want to make it easy for users to find playlists for certain moods so that the user can listen to what they want to when they're feeling a certain mood. (Priority # 2; approximately 20 hours)

As a user I want to be able to share playlists easily so that all my friends and family can listen to the music I like. (Priority # 3; approximately 10 hours)

As a user I want to be able to link my email or social media to my account so I can share playlists and songs easier. (Priority # 6; approximately 5 hours)

As a developer I want to make playlists that match a certain mood so that users can feel the music they're listening to instead of just blankly listening to music. (Priority # 1; approximately 50 hours)

As a shareholder I want the app to do something users want and need so I can advertise it to the public easier. (Priority #4; N/A)

As a developer I want to be able to collect data easily on users (anonymously) so we can make good playlists. (Priority #5; approximately 25-40 hours)

As a user I want to be able to find specific songs to add to any playlist that I create. (Priority #7; approximately 10-20 hours)

As a Developer I want to make general navigation around the app very easy for the user. (Priority #8; approximately 10 hours)

As a user I want to be able to search for music by genre so I can learn about new genres of music that I haven't heard before. (Priority #9; approximately 5 hours)

As a developer I want to make the app look pretty so that people will feel happier while using the app. (Priority #10; approximately 5-10 hours)

8. Trello

The trello link is included here: https://trello.com/b/MjlvEN7h/backlog

9. Group participation

Each member of the group was active in this document. Turan Naimey focused hard on section 1, and task break ups. Turan also helped around multiple sections during the process of this document. (20%). Karsten focused on sections 2 and 3, and assisted others in other sections as well. (20%). Austin C provided help on sections 1 and 2 along with general help on sections 3, 4 and 5. (20%). Austin T worked on sections 6 and 7, while also providing help within 1, 2, 3, and 4. (20%). Mike Ewers worked on sections 8 and 9, while helping with sections 1, 2 and 6 (20%).