Software Requirements Specification

Document For

Online Music Library Management System

Version 1.0 approved

Khushi Jha (2019BTCS036)

Proposed To

Professor Dr. Deepti Chouhan

2020-2021

**Index**

1. Introduction
   1. Purpose
   2. Intended Audience and Reading Suggestions
   3. Product Scope
   4. References
   5. Document Overview
2. Overall Description
   1. Product Perspective

2.1.1 User Interface

2.1.2 Hardware Interface

2.1.3 Software Interface

2.1.4 Communication Interface

* 1. Product Function
     1. Acquire and Index New Items
     2. Get Customer Profile and Statistics
     3. Monitor User Actions/Feedback
     4. View Recommendations
     5. Provide Feedback
  2. User Classes and Characteristics
  3. Constraints
  4. User Documentation
  5. Assumption and Dependencies

1. Specific requirements
   1. Interface Requirements
   2. Functional Requirements
      1. Description:
         1. Register
         2. Sign-In
         3. Sign-Up
         4. Payment
         5. Logout
         6. Favourites (add n remove)
         7. Playlists (diff types)
         8. Shuffle play
         9. Auto play
         10. Repeat
         11. Downloads
         12. Liked songs
         13. Recommendation
         14. Share playlist
         15. Share song
         16. Lyrics
         17. Skip
   3. Non-Functional Requirements
      1. Performance requirements
      2. Design constraints
2. Data Model and Description
   1. Data Description
      1. Data objects
         1. Class Diagram
         2. DFD
         3. E-R Diagram
      2. Data dictionary
3. Behavioural Model and Description
   1. Description for software behaviour
   2. State Transition Diagram
4. Planning
   1. Team Structure
   2. Process Model
5. Conclusion

7.1 Problems and Issues in currents system

7.2 Future extension

1. INTRODUCTION
   1. Purpose

The purpose of this document is to give a detailed description of the requirements for “Music System”. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to the customer company for their approval and a reference for development of the system.

* 1. Intended Audience and Reading suggestions

This project is a prototype for the “music system”. This has been implemented under the guidance of college professors. The different types of reader that the document is intended for are developers, project managers, marketing staff, users, testers, and documentation writers.

* 1. Product Scope

“Music System” is an online music application system, which provides users songs which they may like, based on the songs listed or uploaded on the system. Every logged in user should have access to the all the songs uploaded on the system. The user can go through all genre of the song available online and create their own playlist. The project’s main aim is to provide accurate music playlist to the user. This project is beneficial for the users. For users, they may find songs that they may like without consuming time and even they can encounter new songs which they like from the recommendation. The website has made more attractive, so they draw more users to the website and the system makes the users of the website spend more time online.

* 1. References
* www.Google.com.
* “Fundamentals of Software Engineering” by Rajib Mall.
* Prepared Notes during lectures.

1.5 Document Overview

Following section of this document will focus on describing the system in terms of product perspective, product functions, user characteristics, assumptions and dependencies. In the third section, we will address specific requirements of the system, which will enclose external interface requirements, functional requirements of the system, performance requirements, and other requirements.

1. OVERALL DESCRIPTION
   1. Product Perspective

“Music system” depends on data come from websites users. This program has different type of users, so there is functionality differences between users will occur with respect to item data. Our “music system” should work efficiently according to music data. So, there exists a user interface that is suitable for music system and this interface will be an e-commerce website. Our system will be working in background. Once the “music system” finds an accurate result, it will be shown on the interface. In terms of hardware, music system will be embedded in a website. To use or benefit from “music system”, user should enter from a personal computer, mobile device with internet connection, tablet etc. In terms of software, our “music system” will run on personal computers, smart phones etc. That is, it will run on any device with internet connection. The system will work both Windows and UNIX operating systems. Moreover, it will be implemented making use of database management tools such as MySQL. This brief information of interfaces is explained in more detailed below.

* + 1. User Interface

On the larger system, every user should see a main page with login, register and log-in with Facebook/Google/twitter buttons. If the user is a first time user, he/she should be able to register to the website by clicking the register new user button or be able to directly login with his/her Facebook/Google account by clicking register with Facebook/Google button. If the user chooses to register to the system, he/she should login the system by clicking the login button. The logged in user should see all the genre of songs on the webpages of the website, in which Music System is integrated to.

* + 1. Hardware Interface

Since the web portal does not have any designated hardware, it does not have any direct hardware interfaces. The hardware connection to the database server is managed by the underlying operating system on the web server.

* + 1. Software Interface

The web component communicates with the database in order to get the user rating logs of the larger system. The communication between the database and the web component consists of operation concerning both reading and modifying the data.

* + 1. Communication Interface

The communication between the different parts of the system is important, since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for the web portal.

* 1. Product Function

With the Music System user can create their own playlist, listen to all genre of song which can be of any language, artist, album, etc. The user will be able to see music recommendations made especially for him/her. The recommendations will be based on the user’s previous actions and the actions of the other users who have a similar taste of music as the user, who will get the recommendations.

Since the recommendations are made based on the user, they are likely to be unique.

* + 1. Acquire and Index New Items

This function inserts new items by indexing them like the previously saved items. This functionality may be achieved by a query language. After adding this item to database, this item will also ranked according to its album id for users who listens that album.

* + 1. Get Customer Profile and Statistics

This function provides customer profile information and statistics to supplier of the whole system. Actually, this function provides the whole information needed to make a recommendation.

* + 1. Monitor User Actions/Feedback

The system should monitor user actions as the system receives implicit feedback through it.

* + 1. View Recommendations

To make user tend to make use of an opportunity of a recommendation, monitoring is an important point. Here supplier uses this functionality to get attention of the user on a recommended item by displaying it in an attractive way.

* + 1. Provide Feedback

The users must be able to read reviews and provide feedback to the system. This feedback is crucial to provide more accurate recommendations in time.

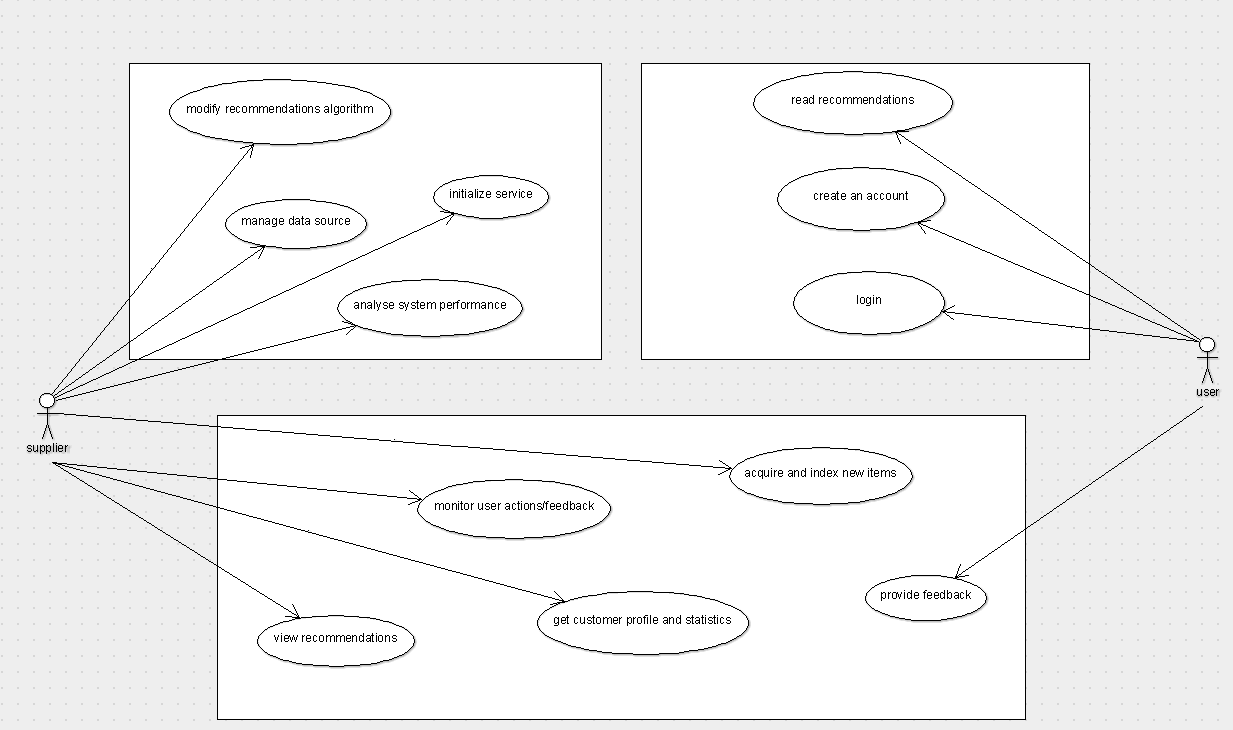


Figure 1. Use case Diagram

* 1. User Classes and Characteristics

The software can be used by the following user categories:

• Simple users that intend to hear a song. These users could be of any age, with no special knowledge. Knowing how to use a computer and mobile with internet is an essential.

• Programmers- Software Developers –Open Source project participants of any age that could understand the program’s source code and expand or improve it. Must have knowledge of the programming language that the software is written in, in order to be able to understand exactly what it does and how it does it. Also some experience on programming will most certainly help on extending or improving the project. General there is no restriction in user’s categories.

* 1. Constraints

●Since we need user profile data while developing the product, to find real time and sufficient data can be a problem for developer because of regulatory policies.

●Millions of data will be needed to test the software. At this stage developers will need huge amount of disk space and clusters.

●We will not design any specific interfaces for the product. It will be suitable for the application that people listen music from. So, developers have to consider common components of these applications so that the software can be integrated to any of this application easily.

●The application gathers real time user profile information from user accounts. Therefore, it must be reliable and keep those data in safe. Moreover, the system will produce new data about users depending on their behaviour on the web. Security of this resulting data must be provided by the software also.

* 1. User Documentation

1. Firstly, the user have to register on the “music system” to use it.
2. When the user completes the registration process then, they have the option to choose their languages in which they will like to listen the songs.
3. Once the user have register no need to register again and only the user have to login in the “music system”.
4. The user can make their own playlist, add and remove a song from favourites, shuffle play, download the song, like the song, get the lyrics and share the songs and their playlist.
5. User will also get recommendation for their song.
6. User can give feedback and rating to “music system”.
   1. Assumption and Dependencies

As stated in the previous section which is constraints, there are several requirements like music data, user data, database management tool etc. To accomplish activating “music system”, these requirements should be provided. However, the case of having all these software and hardware provide, still we might have some difficulties to test the system without strong internet connection. Losing connection is an important problem, because our system will work on online platform. If the internet connection cannot be supplied as requested, the whole sales policy will fail and our system will be useless. We stated that we will use PHP, HTML, CSS, JAVASCRIPT and MYSQL as a programming language to code our system. However, we can also study in other platforms.

1. SPECIFIC REQUIREMENTS
   1. Interface Requirements

The user needs to click the link to the website. Then he/she needs to register to the system with Mobile Number or log-in with Facebook/Google buttons, otherwise he/she won’t be able to use the Recommender System. Then, to benefit from the “music System” he/she needs to be active on the website by listening and downloading music, adding his/her favourites to the list or sharing them on Facebook or twitter.

* 1. Functional Requirements

This section provides requirement overview of the system. Various functional modules that can be implemented by the system will be –

* + 1. Description:
       1. Register

The user can register to the “music system” by Facebook/Google

Or twitter to stay connected to the system for listening songs.

* + - 1. Sign-In

If the user has already sign-up with the following number he can sign-in in the “music system” to stay tuned.

* + - 1. Sign-Up

The user can sign-up in “music system” by simply with their phone number.

* + - 1. Payment

The “music system” is free of cost for all the user. Only some functions of the system need some money to activate it.

* + - 1. Logout

If any user wants to log-out he can log-out through the system.

* + - 1. Favourites (add n remove)

The user can add a song or can remove a song through favourites.

* + - 1. Playlists (diff types)

The user can make their own playlist of the songs.

* + - 1. Shuffle play

This option enable the in-build function to play songs at the random that are present in the queue.

* + - 1. Auto play

This option enable the in-build function to play the songs by their own one by one present in the queue.

* + - 1. Repeat

The user can tap to repeat for listening to the same songs many times.

* + - 1. Downloads

The user can download the songs.

* + - 1. Liked songs

The user can like the songs.

* + - 1. Recommendation

The user will get recommendation of the songs. The recommendation will be according to their searches and recently played songs.

* + - 1. Share playlist

The user can share their playlist through Whatsapp/email or other apps.

* + - 1. Share song

The user can share their songs through Whatsapp/email or other apps.

* + - 1. Lyrics

The user have also option of getting lyrics of the songs.

* + - 1. Skip

We have kept a skip option for the user so that they can simply take fun of listening to the song without getting register to the “music system”.

* + - 1. Add to Queue

From this option user can add any song to the queue.

* 1. Non-Functional Requirements

Following Non-Functional Requirements will be there in the insurance to the internet:

1. Secure access to consumer’s confidential data.
2. Better component design to get better performance at peak time.
3. Flexible service based architecture will be highly desirable for future extension. Non-Functional Requirements define system properties and constraints. Various other Non-Functional Requirements are:

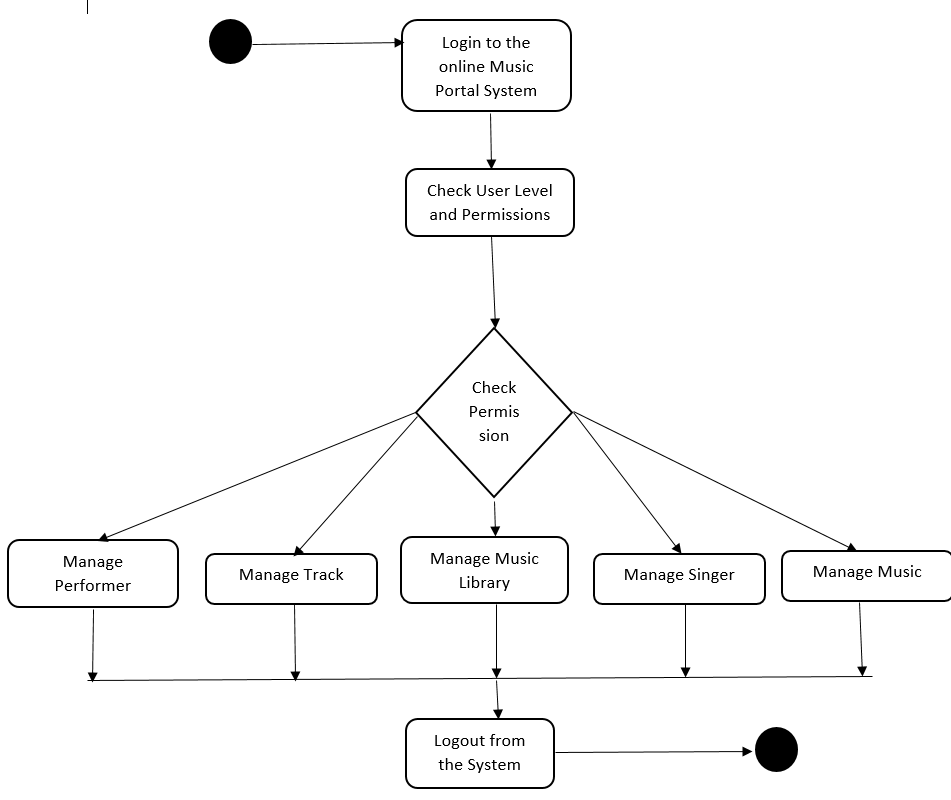
* Security
* Reliability
* Maintainability
* Portability
* Extensibility
* Reusability
* Compatibility
* Resource Utilization
  + 1. Performance requirements

As a larger system, the website has a monthly traffic of over 4 million users. Since music system is planned to be designed for the use of every user in the larger system, it is easy to say that this system will have a monthly traffic of over 4 million users. Every day, over 3 million songs are listened on the larger system. However, not all 3 million songs are logged by the website or not all logged information will be given for the use of the system. By the company, daily 1 million logs will be given to be handled. User information of the past 60 days will be the reference point of the system, that’s why 60 million logs need to be handled in order to implement the whole “music system”.

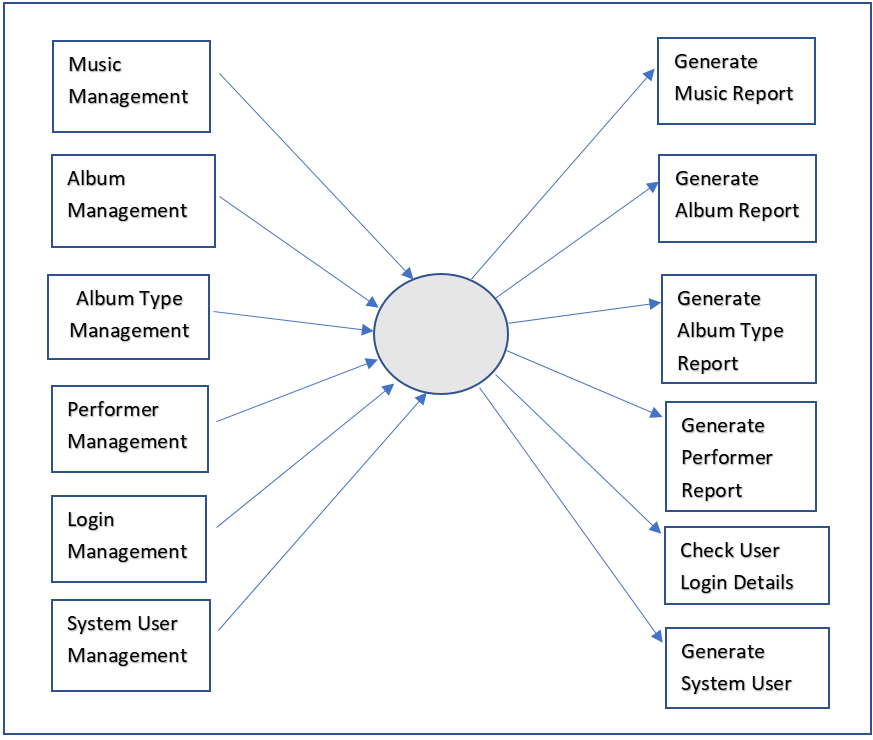
* + 1. Design constraints

In the implementation process of this system, PHP Programming Language will be the main development language. Since PHP is selected to be the main development language but the languages like HTML, JAVASCRIPT, CSS and MYSQL are also chosen as a standard for the development process of the system. In the process of the documentation of the system, IEEE standards will be used and UML standard will be used while designing the diagrams. Since this system will be a part of much larger system, it must be portable to this larger system. That’s why portability is one of the most important attributes of this system. Since the larger system is a website that has the potential of increasing its number of users, user traffic and number of songs, this system needs to be scale up with the website in the correct order. Therefore, scalability must be the number one attribute that system will have

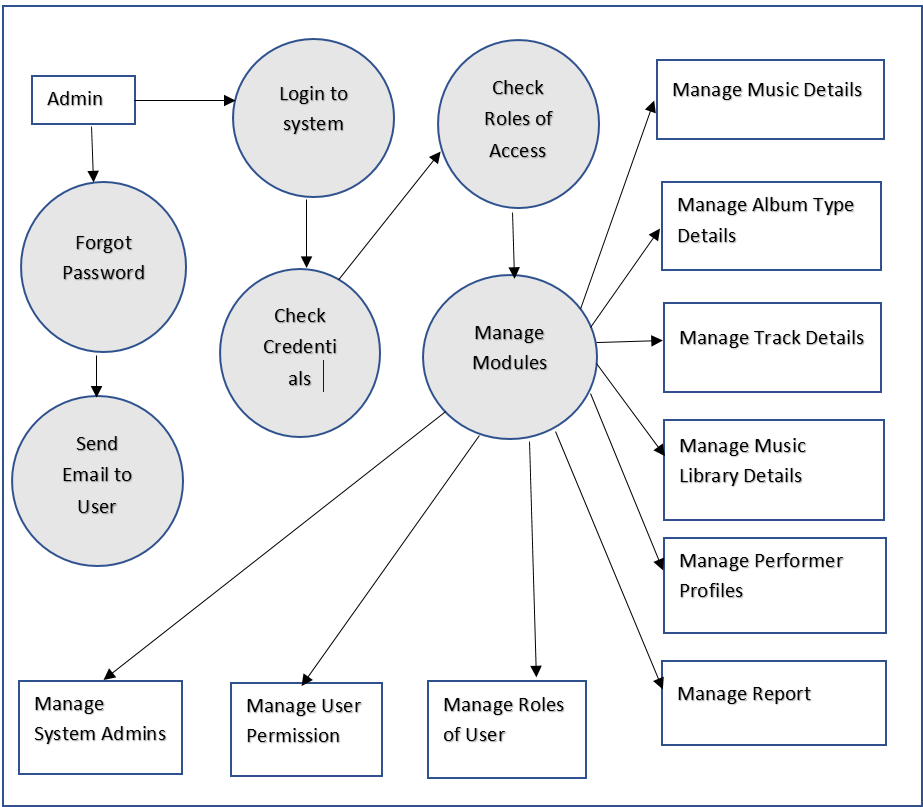
1. DATA MODEL AND DESCRIPTION
   1. Data Description
      1. Data objects
         1. Class Diagram



* + - 1. DFD

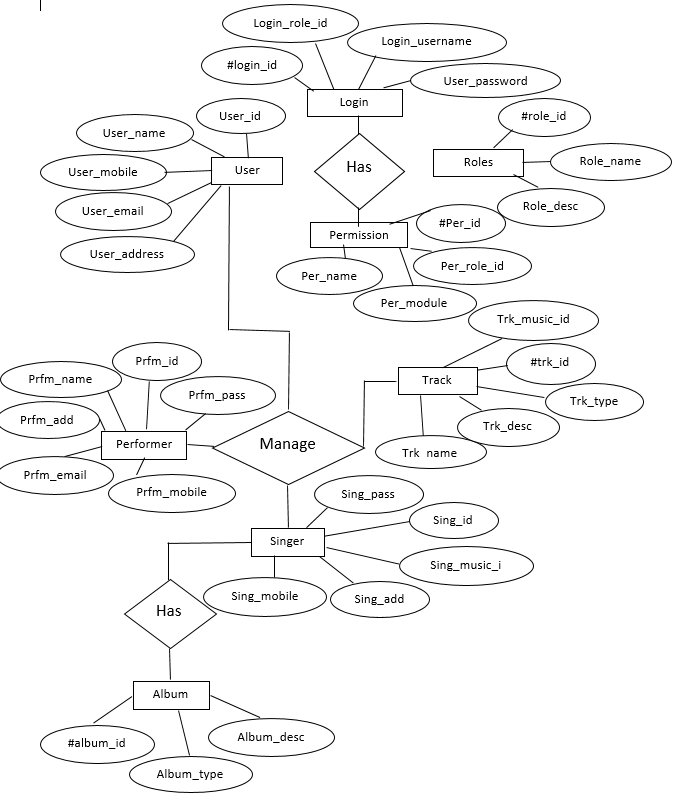


First Level DFD



Second Level DFD

* + - 1. E-R Diagram



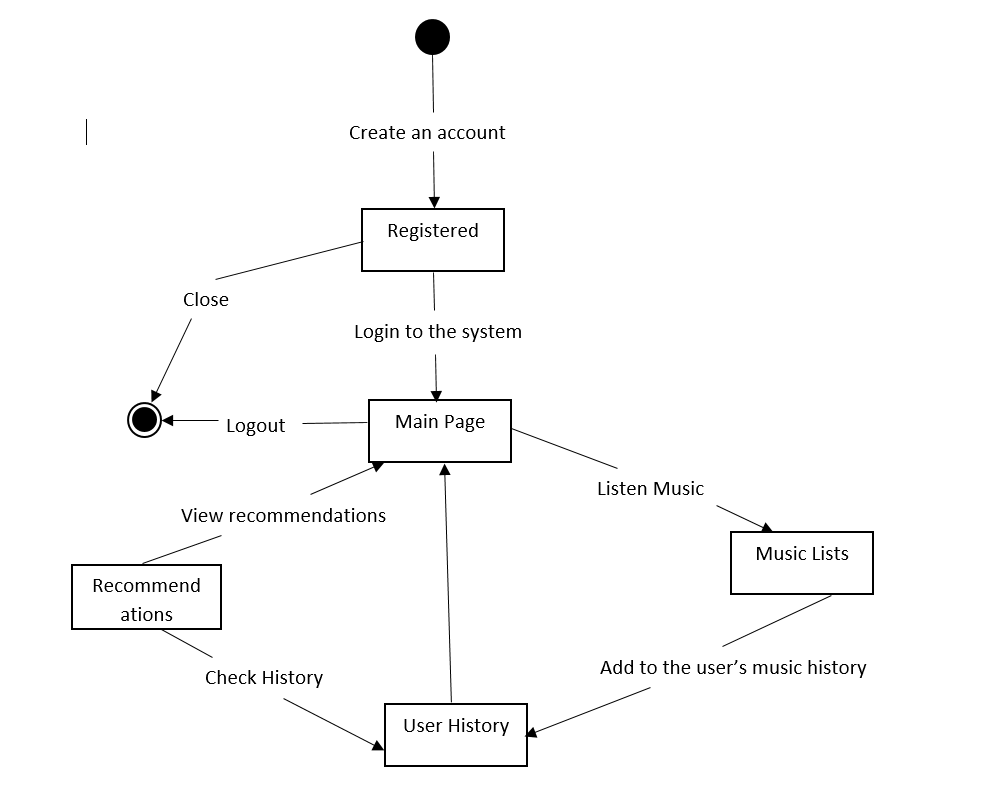
* + 1. Data dictionary

|  |  |  |
| --- | --- | --- |
| **Name** | **TYPE** | **DESCRIPTION** |
| Age | Int | Age of user |
| Gender | Character | Gender of user |
| UserId | Int | Unique Id of User of website. |
| ArtistId | Int | Unique Id of an Artist |
| AlbumId | Int | An Id for the album of the artist whose Id given in the same data set |
| SongId | Int | An Id for the song in the album whose artist ID and album Id given in the same data set |
| TimeOfAction | Date/Time | The time when the user listened the song |
| RatingValue | Float | Rating value is a derived value obtained by a formula depending on user’s actions(listened, downloaded, listened before etc) |
| Channel | String | The abbreviation of where the user listened to the song from(music list, search bar or recommendation list) |

1. BEHAVIOURAL MODEL AND DESCRIPTION
   1. Description for software behaviour

This subsection describes the major events and states of our software. When the user open the webpage at first, user will see a screen in which there will be option of sign-in, sign-up or register with Google/Facebook or Twitter. If user sign-in in the system or register then user will be able to move on to the page where they will decide in which language they would like to listen songs. And then user will be able to open music lists and listen music. Every song that the user listened will be added to the user history. By checking the user history, every user will be able to view recommendations. Users will be able to logout of the system.

* 1. State Transition Diagram



State transition Diagram

1. Planning

In this part of the document, the structure of the team responsible from the project, the basic schedule, and the process model will be presented.

* 1. Team Structure

Development Team

Design

Maintenance

Coding

Testing

Project management

We plan to divide the workload equally at the technical side. The basic structure of workload of team as follows:

1. The whole team is divided into two group
2. Front End Development Team

A front-end developer links together the world of design and technology, packaging up the utility of the back end in an inviting way for users to interact with. Front end team will take website design files and convert them into HTML, JavaScript (JS), PHP and CSS code - the core elements of front-end development.

1. Back End Development Team

The back end runs a website—the user doesn’t see it or interact with it, but it is always running in the background, delivering functionality and a desktop-like experience. Back-end developers play a critical role in web development teams, and make sure that data or services requested by the front end system or software are delivered. In order to make the server, application, and database communicate with each other, back-end developers will use language like PHP to build an application and tools like MySQL, Oracle, and SQL Server to find, save, or change data and serve it to the user in front end code.

* 1. Process Model

As a process model, it is planned to be used Scrum. It is much easier to demonstrate this process model. Here is the visual demonstration of the chosen process model.

Daily cycle

Sprint planning meeting

Release

Product Increment

Update Product Backlog

Sprint Review

Figure11. Process Model

1. Conclusion

Software Requirement Specification is an important part of the process of project development. Moreover, it is a prerequisite for creating the following design documentation. In this document, we have provided the information about general product description, data elements that the product deals with, specific requirements like product's interfaces and the functions will be implemented. In addition general behaviour of the product has been explained in order to make it easy for the user to understand to product's usage clearly. This document has been created through the help of various researches and depending on the demands of company. However, some little specifications are prone to be changed in the future.

* 1. Problems and Issues in currents system

In this system the song will play only if we click to the specific song’s title. There are button for shuffle play, auto play, repeat, next song, previous song but they didn’t work. User cannot make their own playlist and cannot share it. User cannot add music to favourites. User will not get recommendations.

* 1. Future extension

In future we will add functionality to previous button, next button, shuffle play, auto play, and repeat option. User can make their own playlist. Can add their song to the favourites. User get the recommendation of the songs.