1. Introduction

1.1 Purpose

The purpose of this document is to provide a complete requirements specification of the "Online Music Recommender System". The document provides a detailed description about the objectives and goals of the application, the different functionalities supported by the system, the interface requirements of the system, the non-functional requirements that help to evaluate the system and the constraints involved in developing this application. The document serves as a useful reference for developers as well as the stakeholders of this system.

1.2 Document Conventions – Definitions and Acronyms

Through-out this document, the term 'user' refers to a generic user. Users of the system fall into 3 categories namely Customers, Viewers and Admin members.

Table 1

GUI	Graphical User Interface. It provides various screens with appropriate icons, hyperlinks etc. to facilitate screen navigation and data entry.
ER	Entity Relationship
HTML	HyperText Mark-up Languages
MySQL	A backend database for storing the data
PHP	HyperText Preprocessor – An interface to provide the connection between the backend database and the frontend GUI

1.3 Intended Audience and Reading Suggestions

The next 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, various external interface requirements are discussed. Further sections describe functional requirements, performance, safety and other requirements of the system. It is advisable for a developer to go through the contents in sequence to understand the complete system. Users may refer to the content under a particular section directly, the documentation is written keeping end-users in mind. The section on functional requirements would be of utmost importance to developers and project managers who would be interested in a similar concept.

1.4 Product Scope

The Online Music Recommendation website suggests songs which the users may like, based on the songs that they have previously listened to. Every logged in user should have access to the recommender system. The system will go through the songs that user has previously listened to and rated, then according to that information it should suggest songs to the user. The project's main aim is to provide accurate music recommendations to the user. This project is beneficial for visually impaired users also, as they can choose songs by the speech recognition feature of the system. The users can find songs that they like without consuming time and they can even explore new songs, which they may like, by means of the recommendations. The other uses of this website include:

1) Accessing the user's favourite music quickly, as there is a playlist feature provided

2) Exploring diverse categories of music through a single platform

3) Helping the user find music that they may like to listen to, by making suggestions through the incorporation of a recommendation system

4) Allowing the user to request for songs which may not be available at that point of time, using the query portal system

1.5 References

[1] Karl E. Weigers, "Software Requirements Specification for <Project>", IEEE, copyright 1999

[2] IEEE, "IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications", IEEE Computer Society, 1998

[3] Ceng History X, "https://senior.ceng.metu.edu.tr/2014/cenghistoryx/"

2. Overall Description

2.1 Product perspective

Recommender systems are a subclass of information filtering systems that seek to predict the "rating" or "preference" that a user would give to an item. It would cater to a wide variety of users, due to which there will be functionality difference between users with respect to item data.

This recommendation system should work efficiently for music data. So, there will be a user interface that is suitable for music recommendations and the interface will be a Music-sharing website. The recommendation algorithm will be running in the background. Once the recommendation system finds an accurate result, it will be displayed on the interface. The website will include a feature for visually challenged people by incorporating a voice search feature.

Our recommender system will run on personal computers, smartphones, etc. That is, it will run on any device with an internet connection. The system will work on Windows, Linux as well as Mac OsX operating systems. Moreover, it will be implemented by making use of database management tools such as MySQL and PHP.

2.2 Product Functions

This website will be designed to serve different categories of peoples. Some of the main features of the website includes:

- Exploring diverse categories of music on a single platform
- Helping users to access favourite music quickly, with the help of playlist that he/she has created
- Recommending songs that customers may like based on their prior ratings
- Allowing customer to request for songs which may not be available at that point of time using the query portal system
- Customer can post his/her cover song in turn making the site more inviting.
- Admin portal will be designed to monitor user’s actions and feedback.
- Admin can access the performance graph of any customer.

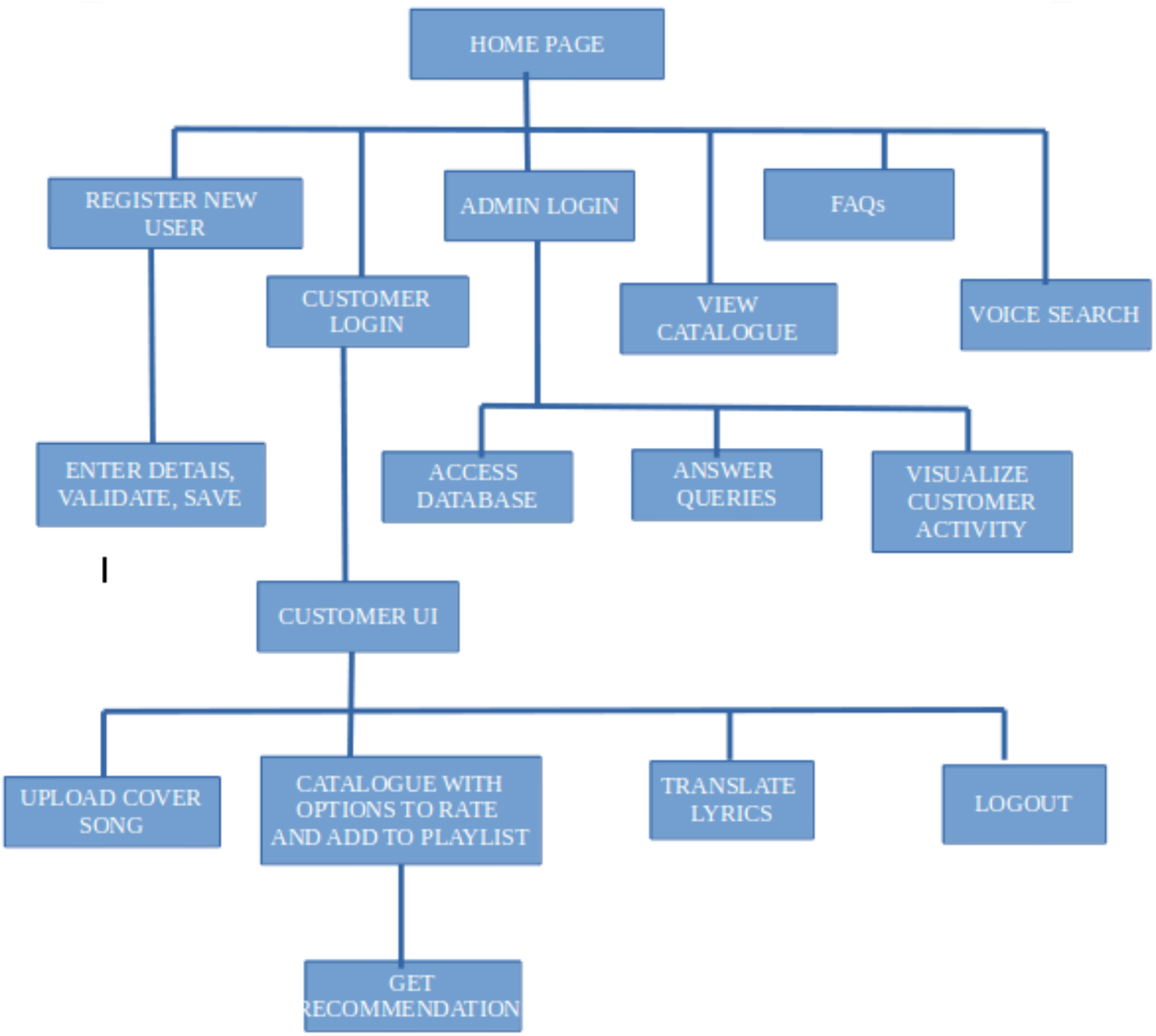


Figure 1: The flow of website

2.3 User Classes and Characteristics

Depending on the different functionalities, the users of this website are classified into three main categories:

1. Visitor

2. Customer

3. Administrator

i. Viewer

- New user can visit the home page of the online music site.
- Search for a particular music item
- View new releases.
- Can see the most trending songs.
- For creating their own playlist, first the visitor has to click the register option in the client login area.

ii. Customer (Registered user)

- Initially the customer has to login into the user interface which will help him/her to avail the services of the website.
- Can create their own playlist.
- Can rate the songs.
- Will get recommendation of music they may like.
- Can upload his/her cover songs.
- Can post the query regarding the site.

iii. Administrator

The website will be designed to provide following functionalities to the admin:

- Get Customer profile and statistics.
- Add/modify/delete songs in the database.
- Monitor user actions/feedback.

- Answer all the questions posted by the users in the query portal.

2.4 Operating Environment

The goal of this website is to provide an automated Web application that allows a user to browse, store his/her favourite music and post questions over the Internet at any point of time as well as to provide effective decision support tools, i.e., recommendation system in order to guide users through the exploration process. The constructed system will have three-tier architecture. Figure 1 shows the system on three- tier architecture.

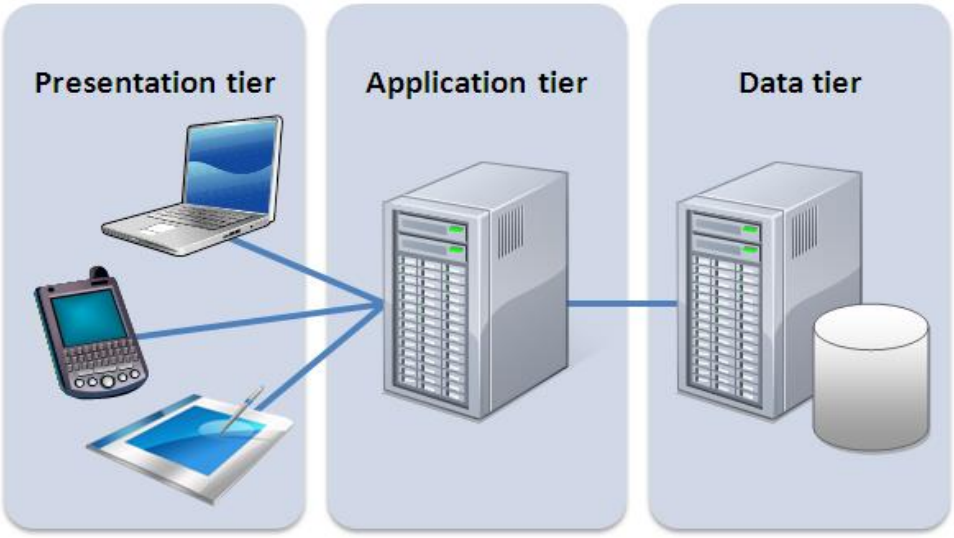


Figure 2: Three tier architecture

2.4.1 Hardware specification

Processor	:	Pentium-III and above

2.4.2 Software specification

Operating System	:	Linux, Windows, MacOSx
Programming language	:	For recommendation system - Python
		For accessing data from database - SQL
Client Side technologies	:	XHTML, CSS, Python, Bootstrap
Server Side technologies	:	PHP
Database support	:	MySQL

Browser support : Firefox, Chrome, IE 7 and above.

2.5 Design and Implementation 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.
- Developers have to consider common components of the applications used to listen music, so that the software can be integrated to any of these applications easily.
- The application gathers real time customer 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.
- Most important concern of the system is producing accurate recommendations. To provide expected accuracy and to handle with sparse and huge data at the same time is critical.

2.6 User Documentation

A successful website must provide a proper user documentation which would allow the site to be used by a person from any domain. Therefore, in-order to help customers get acquainted with all the features of the website, the user documentation will be provided in the following ways:

- User manual
- On-line help via. Query portal system
- Video tutorial

2.7 Assumptions and Dependencies

As stated in the constraints section, there are several requirements like music data, user data, database management tool etc. to accomplish activating recommendation system. In order to get an accurate recommendation, the algorithm needs to learn from large number of examples. Since storing such huge dataset is practically hard to achieve, we will be storing details of all the songs instead of the actual mp3 files itself. It is assumed that the user has the good internet connection while using the website.

3. External Interface Requirements

3.1 User interface

The user interface handles the interactions between the "Music Recommendation System" and the users of the system (admin, customer and viewer). Users who have not created an account will not be allowed to rate songs, create playlists, get recommendations or post queries. They can simply view the entire catalogue and search for a particular song.

Viewer can register on to the website by filling in his/her details in the registration form and the registered user (customer) can login by filling in their username and password.

To get music from the Recommender System customer needs to be active on the website by listening and downloading music, adding his/her favourites to the list .The viewers can also get the lyrics of the song in any of the languages by selecting their desired option from a drop-down menu.

3.2 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.

3.3 Software interface

Many software components will be used in building the "Music Recommender System".

Web development languages like HTML, CSS, JavaScript and PHP. Bootstrap has also been used for the responsive design of the front end so that it is customizable to the devices having different dimension. The backend runs on MySQL provided by XAMPP.

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 updating, removing, reading and modifying the data.

Some software specifications are as follows

Operating System	:	Linux, Windows, Mac OsX
Programming language	:	For recommendation system - Python
		For accessing data from database - SQL
Client Side technologies	:	XHTML, CSS

Server Side technologies : PHP

Database support : MySQL(phpMyadmin)

3.4 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.

4. System Features

The music recommender website has three actors, viz., a Customer, a Viewer and an Administrator. The entire music catalogue and customer details are stored in a SQL database.

4.1 Search music catalogue

4.1.1 Description and priority: This feature is a Customer as well as a Viewer use case. The purpose of this part of the application is to enable the customer/viewer to find the available music/song of his choice without browsing the entire catalogue. This can be a medium priority use case.

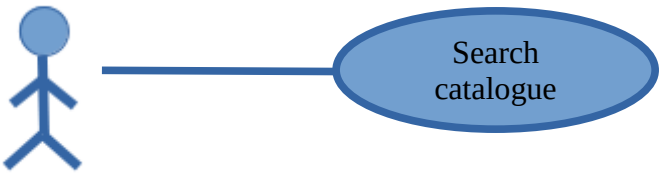
4.1.2 Stimulus or Response Sequences

1) The customer/viewer can enter any text in the search text box and hit the "Search" button.

2) This will redirect the user to the page which will display all the matched items.

3) Otherwise an appropriate message will be displayed.

4.1.3 Functional Requirements



Customer/Viewer

Req. 1) The user inputs should be valid. A valid input is a song name, artist name or a genre name. An invalid input is anything other than that.

Req. 2) A message should be displayed affirming the user's choices along with the appropriate music/song information for the particular search.

Req. 3) The system should display an appropriate message if there are no matches for the particular search.

4.2 Browse music catalogue

4.2.1 Description and priority : This feature is also a Customer as well as a Viewer use case. The customer/user can browse through the different genres, categories and can also view the details of the songs which are listed. Depending upon the genre/category selected, the contents of the table are

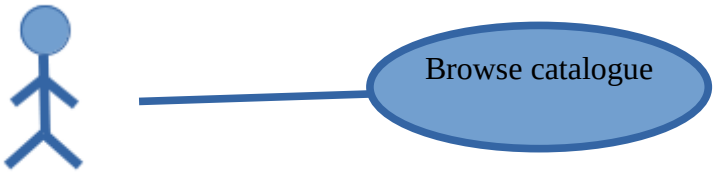
accessed using a select query of SQL database. This can be a low priority use case as many viewers/customers prefer searching for a song if they are particular about it.

4.2.2 Stimulus or Response Sequences

1) The Customer selects one of the genre or a category.

2) Then, he can play the music of his own choice.

4.2.3 Functional Requirements



Customer/Viewer

4.3 View FAQs

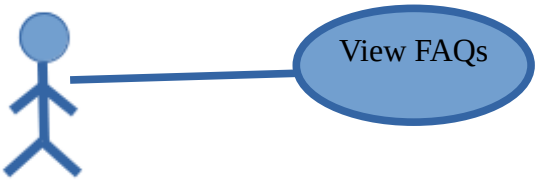
4.3.1 Description and priority : This feature is also the use case of both customers as well as viewers. This use case corresponds to viewing of the frequently asked questions related to the website. They are linked directly through the home page of the website. As many people just tend to browse the site more often and find out answers to their questions rather than going through the FAQ page, this use case has been assigned a low priority.

4.3.2 Stimulus or Response Sequences

1) The viewer/customer clicks on the “View FAQs” button.

2) The system displays the FAQs.

4.3.3. Functional Requirements



Customer/Viewer

4.4 Voice Search

4.4.1 Description and priority : This feature is a use case of both customers as well as viewers. This uses the algorithm of speech to text conversion and then, searching of music with the help of this text. This feature would be assigned medium priority as it is an innovative feature, but not many people use it.

4.4.2 Stimulus or Response Sequences

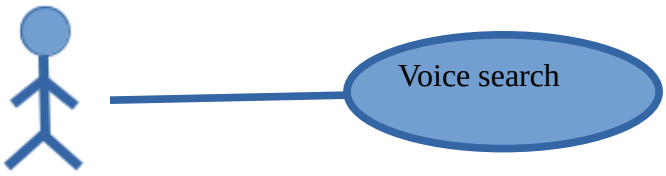
1) The customer/viewer will have to keep a button pressed and speak out the name of the song, genre or artist in the microphone.

2) The system recognizes the speech. If not, gives a prompt back asking the user to speak again.

3) It then converts this speech to text and runs the search algorithm.

4) It finally displays the corresponding results for the search.

4.4.3 Functional Requirements



Customer/Viewer

Req. 1) The system should prompt the user to speak if the voice button is kept pressed for 5 seconds and nothing is spoken.

Req. 2) The system displays an appropriate message notifying the user that the particular song/genre/artist is not currently available in the website if the spoken text does not lead to any successful search.

Req. 3) The system should display search results for successful searches.

4.5 System Registration

4.5.1 Description and priority : This feature is a Viewer use case. It is implemented in order to enable a new user registration to the website. A valid user account must be used for an existing customer or a new customer can register. This would be a high priority use case as the other features of the website can be accessed only after registration and login takes place.

4.5.2 Stimulus or Response Sequences

1) If the customer is a new user, he/she can request to register with the system.

2) The system generates a form which the viewer has to fill out.

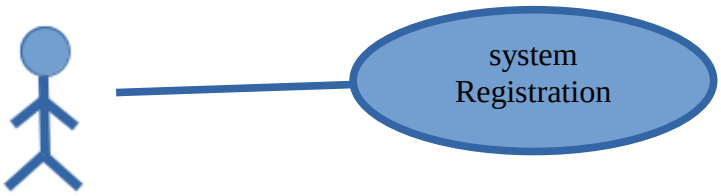
3) The form contains necessary details of the viewer, such as, his/her email id, etc.

4) The user also has to set a password.

5) The system then generates a customer ID which is unique for every customer.

6) The customer has to login to his portal by entering this ID and the corresponding password.

4.5.3 Functional Requirements



Viewer

Req. 1) The system should display a registration page and the customer should enter a user name, password and a valid email id (mandatory fields).

Req. 2) The password and confirm password must match. Otherwise, the user must be notifed.

Req. 3) The password must be of minimum 6 characters and a maximum of 12 characters in length.

Req. 4) The system should display a “Successfully registered” message if all the details entered are valid.

Req. 5) The system should not accept any invalid data.

4.6 System Login

4.6.1 Description and priority : This feature is a Customer and Administrator use case. This is for a customer to login to his/her profile and is implemented to enable user authentication. A valid user account must be used for an existing customer. Similar is the case for an Admin. This feature would receive a high priority and is quite beneficial since, many other important features of this website can be accessed only after a customer/admin logs into his/her profile/dashboard respectively.

4.6.2 Stimulus or Response Sequences

1) The customer clicks on the login button which is displayed on the home page of the website.

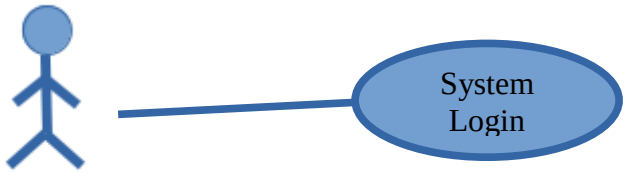
2) This logs the customer to his/her profile after validating for the correct user-id and password.

3) The customer can then explore the various features available.

4) If the customer forgets the password, then, he/she can send an email displayed in the contact section of the home page of this music recommender website.

5) A link would then be sent to their registered email id in order to reset the password. (This link will be a one time login link and will expire after 24 hours).

4.6.3 Functional Requirements



Customer

Req. 1) The system should authenticate user-id and password.

Req. 2) The system displays an appropriate error message if the user-id and/or password is/are invalid. Then, the customer should re-enter his user-id and password.

Req. 3) The corresponding customer profile should be displayed by the system if the user inputs are valid.

4.7 Rate Songs

4.7.1 Description and priority : This feature is a Customer use case. Here, the customer can rate a song of his/her choice. The main purpose of rating these songs is to get song(music) recommendations. This would be given a high priority as getting music recommendation is the main feature of this website and recommending music to other people is possible only when we have certain set of music rated.

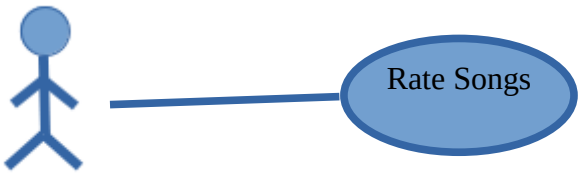
4.7.2 Stimulus or Response Sequences

1) The customer chooses a song.

2) Enters a rating between 1 and 5 (inclusive) depending on his preference.

3) The system stores these ratings in the database.

4.7.3 Functional Requirements



Customer

Req. 1) Rating only between 1 and 5 (inclusive) is allowed. This is a valid rating.

Req. 3) The system should display an appropriate error message if no rating is entered or if the entered rating is an alphabet or a decimal or any other invalid data.

4.8 Get Song Recommendation

4.8.1 Description and priority : This feature is the use case of Customer. It is the "core" feature of the website. The purpose of this part of the application is to enable the customer to find recommendations for the songs of his/her choice. This has been assigned high priority.

4.8.2 Stimulus or Response Sequences

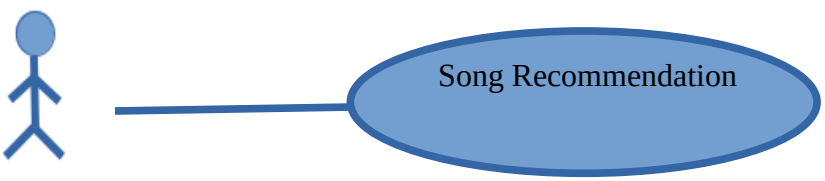
Before this use case can be initiated, the customer should have already rated at least 3 to 5 songs.

1) Customer rates a song.

2) As soon as a song is rated the recommender starts running in the background.

3) The system then displays the top 5 most recommended music/songs to the customer only after the customer has rated at least 3 to 5 songs.

4.8.3 Functional Requirements



Customer

Req. 1) The recommender should start as soon as a song is rated by any customer.

Req. 2) At least 3 to 5 songs must be rated by the customer in order to get accurate recommendations.

Req. 3) The system should display the top 5 song suggestions to the customer if he/she has rated 5 or more songs.

4.9 Create playlist

4.9.1 Description and priority : This feature also is the use case of a Customer. The purpose of this part of the application is to enable the customer to store his/her favourites separately in a playlist in order to access them quickly. It has been assigned a high priority as the user can directly play all the songs of the playlist instead of individually playing all the songs or adding individual songs to queues.

4.9.2 Stimulus or Response Sequences

1) Customer selects a song.

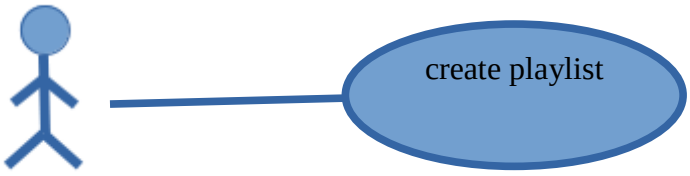
2) Then, he/she clicks on "Add to My Playlist" button.

3) The system adds that particular song to the customer's playlist.

4) Customer also has the option of viewing the playlist.

5) Any number of songs can be added to playlist.

4.9.3 Functional Requirements



Customer

Req. 1) The customer should be able to view his playlist and modify it.

Req. 2) Any customer must be able to access the playlist anywhere, anytime with just an active internet connection on his device.

4.10 Translate lyrics

4.10.1 Description and priority : Every song has its lyrics stored along with it which can also be viewed explicitly on the website. As this is a global site, there are people from all over the world viewing it and enjoying the music. So, there is a feature of translation of lyrics which can translate the lyrics of the current song to another language. This feature has been assigned medium priority.

4.10.2 Stimulus or Response Sequences

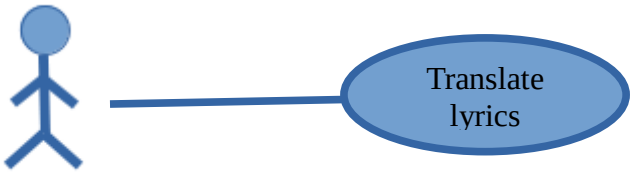
1) The customer selects a particular song.

2) Then, clicks on the corresponding view lyrics option.

3) If the customer wants to view the lyrics in some other language, then, he/she can choose that particular language from the dropdown menu.

4) The system displays the translated lyrics.

4.10.3 Functional Requirements



Customer

Req. 1) The customer should select a song to translate its lyrics.

4.11 Post questions on query portal

4.11.1 Description and priority : If the customer has any queries (for example : guidance regarding the features of the website (in case they are not clear after reading the FAQs, etc.), they may post a query/request on the query portal. This feature has been given a high priority since, if the customers really like the features of the website and they want to know how exactly it works, they can post a query through the query portal.

4.11.2 Stimulus or Response Sequences

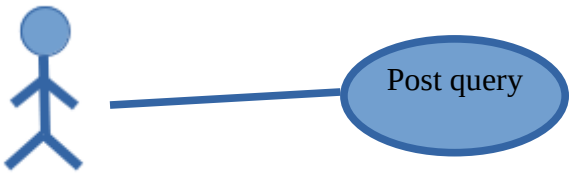
1) The customer enters his registration ID and query in the form available in his profile.

2) The system adds the query to the query portal list and admin has the priviledge to view this.

3) Admin then goes through the query and responds to it accordingly. He/she can either send a mail back to the customer who posted that query and help him out.

4) Or, if the admin thinks that there are chances of the same question being asked by other people too, then, he can add it to the FAQs page too, apart from mailing back the customer.

4.11.3 Functional Requirements



Customer

Req. 1) The customer must ensure that his/her customer ID is displayed correctly while filling the form of query portal.

Req. 2) The system should not accept blank queries.

4.12 Upload cover songs or instrumentals

4.12.1 Description and priority : This is the use case of a Customer. It enables the customer to upload his/her cover songs and/or instrumentals and earn points for it. It has been given a high priority since many people who are interested in listening to music are also interested in singing and playing musical instruments and this website would provide them a good platform to showcase their talent.

4.12.2 Stimulus or Response Sequences

1) The customer clicks on upload cover song button.

2) The system opens the file manager.

3) Customer clicks on the file which has to be attached.

4) A customer can also get goodies after they earn a certain number of points.

4.12.3 Functional Requirements

Customer

Req. 1) The file being uploaded should be in .mp3 format.

Req. 2) File size should not exceed 10 MB.

4.13 Manage database

4.13.1 Description and priority : This is an Administrator use case. This is implemented to allow Administrator to do the following tasks :

- Add new data to the existing database
- Delete data from an existing database
- Modify data from an existing database

This feature has been assigned high priority as it is the central component for backend.

4.13.2 Stimulus or Response Sequences

A) For adding new data to the existing database (This use case is to create and add new genres, categories to the catalogue)

1) Administrator enters the name and necessary details to create a new genre or category in the catalogue.

2) The changes made to the catalogue are updated and saved by the system.

3) The systems also displays a “Success” or a “Error” message accordingly.

B) For deleting data from an existing database (This is to remove genres, categories from the music catalogue and also to delete customers if they are inactive for more than two years)

1) Administrator selects a genre/category/customer that is to be removed from the respective catalogues and clicks the remove button.

2) The changes made to the catalogue are updated and saved by the system.

3) The systems also displays a “Success” or a “Error” message accordingly.

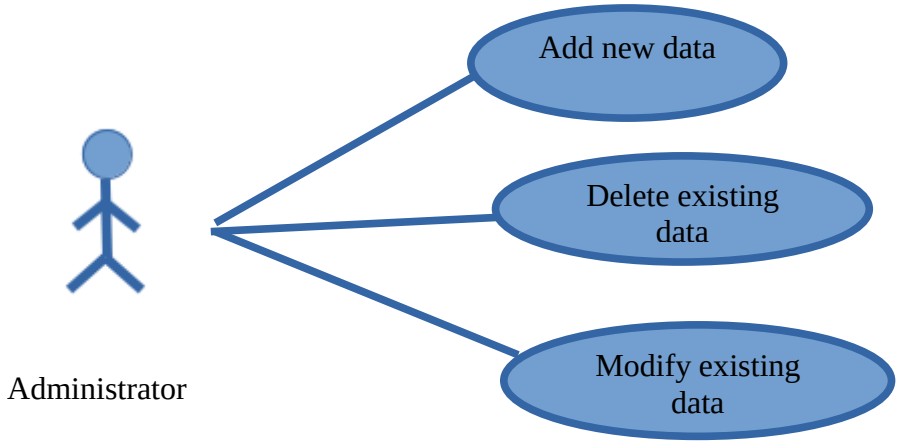
C) Modifying data from an existing database (This is to make some changes in the existing genres, categories. These changes might include modifying the artist name, updating the year in which a particular music was produced, etc. The customer details can also be updated if there is a request from the customer)

1) Administrator selects a genre/category/customer details column which is to be updated/modified and enters the correct details.

2) The changes made to the catalogue are updated and saved by the system.

3) The systems also displays a “Success” or a “Error” message accordingly.

4.13.3 Functional Requirements



A) For adding new data to the existing database

Req. 1) Administrator must be logged in to be able to create and add a new genre or category.

Req. 2) The genre to which the new category is to be associated should exist in catalogue.

B) For deleting data from an existing database

Req. 1) Administrator must be logged in to be able to create and add a new genre or category.

Req. 2) The genre to which the new category is to be associated should exist in catalogue.

Req. 3) A customer registration must exist before deleting a customer.

C) Modifying data from an existing database

Req. 1) Administrator must be logged in to be able to modify a genre or category.

Req. 2) Modification would be done only if the initial entered information was incorrect.

4.14 Answer customers' queries

4.14.1 Description and priority : This is the use case of an admin. Its purpose is to reply the customer's queries/questions and/or address any other requests they make. This has been assigned high priority since this feature is really important in order to cater to the needs of the customers.

4.14.2 Stimulus or Response Sequences

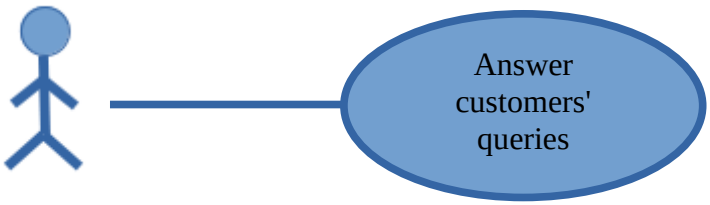
1) Administrator visits the "Query Portal".

2) He/she selects the unanswered queries (one at a time) and updates the answer either in the FAQs page or mails back the customers or both.

3) The changes made to the catalogue are updated and saved by the system.

4) The systems also displays a "Success" or a "Error" message accordingly.

4.14.3 Functional Requirements



Administrator

Req. 1) Administrator must be logged in to be able to answer the customers' queries.

4.15 Visualization through graphs

4.15.1 Description and priority : This is an admin use case. The main purpose of this is to give an overall idea to the admin regarding the taste of the customer (,i.e., what sort of songs/music is a customer listening to). This feature also has been given a high priority since this is used to figure out the current trending songs (which are displayed on the home page of the website).

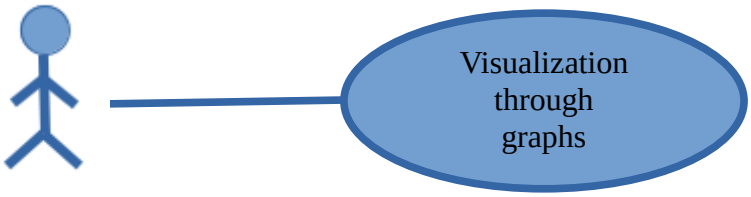
4.15.2 Stimulus or Response Sequences

1) Admin selects a particular customer and clicks on View activity button.

2) The system shows the graph. The graph displays a detailed analysis of what sort of songs is the customer listening to.

3) Admin also updates the trending songs on the home page.

4.15.3 Functional Requirements



Administrator

Req. 1) Admin must be logged in to view customers' activities.

Req. 2) The system should not show any graph if the customer has not listened to any song.

5. Detailed Non-Functional Requirements

The Online Music Recommendation system will be on a server with high speed Internet capability, hosted locally. The website developed assumes the use of XAMPP server for connection between the Web pages and the database.

5.1 Logical Structure of the Data

The logical structure of the data to be stored in the internal Online Music Recommendation Website database is given below in the ER Diagram.

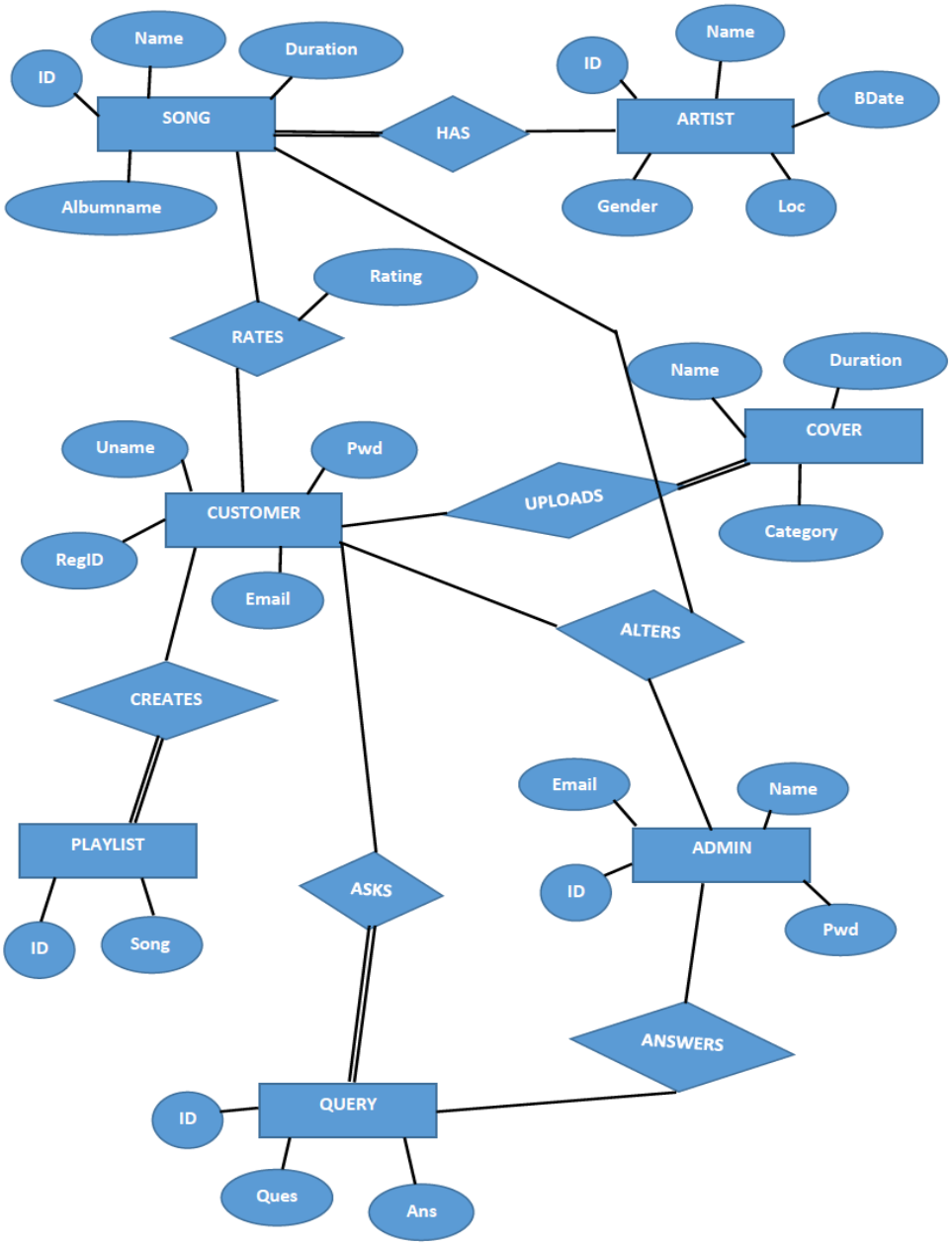


Figure 3: Logical design

6. Other Non-Functional Requirements

6.1 Performance Requirements

Feature 1: Registration and Login

This is a standard feature which is a part of most online portals. But every system must cater to the privacy standards and protect Customers' personal information. Hence database access is provided only to authorised members (Admin) and Customer must also login to ensure that he/she is a genuine person availing the services of the Online Music Recommendation System. The system developed will perform checks to ensure valid data entry.

Feature 2: Rating songs

A user must first register on the website then login to be able to rate songs. Only this will allow him/her to avail the recommendation service (as customer patterns should be observed for accurate suggestions). The system must verify this sequence.

Feature 3: Recommendation

This feature requires newly registered customers to provide ratings for a few (3-5) songs prior to suggesting new tracks. This is because the recommendations will be made based on similarity in likes/interests of the new customer to existing customers.

Feature 4: Admin Privileges

The system must allow for modification of the backend database – a primary performance requirement. There must be an easy-to-use interface for this. Quick and easy access of the internal data will be provided to those who are authorised. Additionally, website maintenance privileges - like answering FAQs and viewing customer information for comparison purposes – are also granted to authorised members of the Admin panel.

6.2 Safety Requirements

There is no loss, damage or harm that could result from the use of this system. The Customer's credentials that are stored in the internal database are protected from other customers - as only authorized members can access the database.

6.3 Security Requirements

The server on which the system resides will have its own security to prevent unauthorized *write*/*delete* access to the database, in the form of a login for the admin personal. Only the authorized admin can modify or update the database. There is no restriction on *read* access. But only authentic customers can avail the services provided by the system - this is ensured by a customer login which checks credentials. Individual customer data is protected from other customers and can be viewed only by the admin.

6.4 Software Quality Attributes

The Online Music Recommendation Website will be available to all users with an internet connection, irrespective of the hardware of the machine they access the site from. It will be reusable as it considers any user's most recent ratings to suggest new ones. It will be easy to use and understand - the Graphical User Interface will be designed such that a new user will be able to quickly grasp the various features of the website.

6.5 Business Rules

There are 3 important types of users for the Online Music Recommendation Website –Customers (registered users), Viewers (unregistered users) and Admin personal. Viewers are only allowed to view the music catalogue and perform a filter-based search to check for a particular song. Customers avail special services like recommendations, option to maintain his/her playlist of favourite songs, option to make suggestions to improve the site and option to upload his/her own cover song/tune. Members of the Admin panel are responsible for maintaining the database and can perform functions like updating, deletion and modification of records and/or tables.