
Shworolipi

A Music Streaming Web
Application

Team Skylark

Institute of Information Technology

University of Dhaka

Music Streaming Web Application

SRS of the Project

Submitted to

Emon Kumar Dey

Assistant Professor

Saeed Siddik

Lecturer

Institute of Information Technology

University of Dhaka

Supervised by

Dr. Zerina Begum

Professor

Institute of Information Technology

University of Dhaka

Submitted by

Team Name:

Skylark

Team Members:

Aquib Azmain (BSSE 0718)

Nishat Tasnim Niloy (BSSE 0723)

BSSE Session: 2014-2015



**Institute of Information Technology
University of Dhaka**

15-05-2017

LETTER OF TRANSMITTAL

15th May 2017

Emon Kumar Dey

Assistant Professor

Institute of Information Technology

University of Dhaka

Subject: Submission of software requirements specifications of Software Project Lab 2.

Dear Sir,

With due respect, we are pleased to submit the final report on software requirements specifications of Music Streaming Web Application. Although this report may have shortcomings we did try our level best to produce an acceptable software requirements specifications. We would be highly obliged if you overlooked our mistakes and accepted our effort we put in this SRS.

Sincerely yours,

Team Name: Skylark

Aquib Azmain (BSSE 0718)

Nishat Tasnim Niloy (BSSE 0723)

BSSE 7th batch

Institute of Information Technology

University of Dhaka

ACKNOWLEDGMENT

At first, we would like to thank almighty for helping us complete this project.

We would like to express our deepest gratitude to all those who provided us the support and encouragement to complete this project. Thanks to our Supervisor teacher Dr. Zerina Begum, Professor, Institute of Information Technology, University of Dhaka, whose continuous suggestions and guidance has been invaluable to us. Also special thanks to Dr. Mohammad Shoyaib sir, Dr. Kazi Muheymin-Us-Sakib sir, Dr. B M Mainul Islam sir, Emon kumar Dey sir and Saeed Siddik sir for their valuable opinions which help us to improve our insight. Without such stimulating suggestions, the project could not have made this much progress.

We are grateful to the Institute of Information Technology for giving us the opportunity to do such a project.

Lastly, we would like to thank our classmates. Although they were not on our team, they have always been helpful and provided valuable insights from time to time.

ABSTRACT

This document contains the software requirements and specifications for the Software Project Lab 2. It contains a scenario-based model, data-based model, class-based model and behavioral model. Using this document as a guide, we are describing the requirements, necessary diagrams, procedures, design for database and working sequence of our project.

Our project is about a music streaming web application. Here we will discuss how we will identify the requirements, how to analyze them and how to present a recommended solution for the system.

This will help to make the software according to the demand of the stakeholders.

CONTENTS

Chapter 1: Introduction of The music streaming web application	1
1.1 Purpose.....	1
1.2 Intended Audience	1
Chapter 2: Inception of The music streaming web application.....	3
2.1 List of Stakeholders.....	3
2.2 Recognizing Multiple Viewpoints.....	5
2.3 Working towards Collaboration.....	6
2.4 Requirements Questionnaire.....	7
Chapter 3: Elicitation of The music streaming web application.....	9
3.1 Quality Function Deployment (QFD).....	9
3.1.1 Normal Requirements	9
3.1.2 Expected Requirements	10
3.1.3 Exciting Requirements.....	10
3.2 Usage Scenario of the music streaming web application	11
3.2.1 HomePage	11
3.2.2 Searching songs.....	11
3.2.3 Music pLayer	11
3.2.4 Registration.....	11
3.2.5 Authentication	12
3.2.6 Personal account	12
3.2.7 Playlist	12
3.2.8 Admin section	12
Chapter 4: Scenario-Based Model of the music streaming web application.....	13
4.1 Definition of Use case.....	13

4.2	Use Case Diagrams	14
4.2.1	Level-0 Use Case Diagram-Shworolipi	14
4.2.2	Level-1 Use Case Diagram-Subsystems of shworolipi	15
4.2.3	Level-1.1 Use Case Diagram-Homepage	16
4.2.4	Level-1.1.1 Use Case Diagram-Registration	17
4.2.5	Level-1.1.2 Use Case Diagram-Authentication	18
4.2.6	Level-1.2 Use Case Diagram-Searching songs	19
4.2.7	Level-1.3 Use Case Diagram-Music player	20
4.2.8	Level-1.4 Use Case Diagram-Personal Account	22
4.2.9	Level-1.4.3 Use Case Diagram-Playlist	23
4.2.9	LEVEL-1.5 USE CASE DIAGRAM- Administration	24
4.3	Activity Diagrams of MSWA	25
4.3.1	Activity diagram of homepage	26
4.3.2	Activity diagram of registration	27
4.3.3	Activity diagram of authentication	28
4.3.4	Activity diagram of Searching songs	29
4.3.5	Activity diagram of music player	30
4.3.6	Activity diagram of personal account	31
4.3.7	Activity diagram of playlist	32
4.3.8	Activity diagram of administration	33
4.4	Swimlane Diagrams of MSWA	34
4.4.1	Swimlane diagram of homepage	34
4.4.2	Swimlane diagram of registration	35
4.4.3	Swimlane diagram of authentication	36
4.4.4	Swimlane diagram of Searching songs	37
4.4.5	Swimlane diagram of music player	38

4.4.6	Swimlane diagram of personal account	39
4.4.7	Swimlane diagram of playlist	40
4.4.8	Swimlane diagram of administration	41
Chapter 5: Data-Based Model of the music streaming web application		42
5.1	Grammatical Parsing and Analysis	42
5.2	Potential Data Objects	44
5.3	Analysis of Potential Data Objects	44
5.4	Final Data Objects	45
5.5	Data Object Relationship	46
5.6	Entity Relation Diagram	47
5.7	Table Translation	48
5.8	Schema Tables	48
Chapter 6: Class-Based Model of the music streaming web application		52
6.1	Grammatical Parsing and Analysis	52
6.1.1	CLASS Identification with General Classification	52
6.1.2	Class Identified with Selection Criteria	54
6.2	Preliminary Classes	56
6.3	Verb Identification	56
6.4	Attributes and Methods of Preliminary Classes	58
6.5	Analysis of Potential Classes	59
6.6	Final Classes:	59
6.7	Attributes and Methods of Final Classes:	60
6.8	Class Responsibilities and Collaborators Card of Final Classes:	63
6.1	Class Diagram	65
Chapter 7: Behavioral Model of the music streaming web application		67
7.1	Event Identification	67

7.2	State Diagram	68
7.2.1	User	69
7.2.2	Member	69
7.2.3	Musicplayer	70
7.2.4	Non-member.....	70
7.2.1	Song.....	71
7.2.2	Playlist	71
7.2.3	Admin	71
7.3	Sequence Diagram	73
7.3.1	Homepage	73
7.3.2	Personal account	74
7.3.3	Administration	75
7.3.4	Full-system (MSWA)	76
Chapter 8: conclusion		77
References		78

LIST OF FIGURES

Figure 1: Level 0 Use Case diagram of MSWA.....	14
Figure 2: Level 1 Use Case Diagram-Subsystems of Shworolipi	15
Figure 3: Level 1.1 Use Case Diagram of Homepage	16
Figure 4: Level 1.2 Use Case Diagram of Registration	17
Figure 5: Level 1.1.1 Use Case Diagram of Authentication	18
Figure 6: Level 1.4.3 Use Case Diagram of Searching songs.....	19
Figure 7: Level 1.3 Use Case Diagram of Music Player	20
Figure 8: Level 1.4 Use Case Diagram of Personal Account	22
Figure 9: Level 1.4.3 Use Case Diagram of Playlist	23
Figure 10: Level 1.5 Use Case Diagram of Administration.....	24
Figure 11: Level 1.1 activity diagram of homepage	26
Figure 12: Level 1.2 activity diagram of registration	27
Figure 13: Level 1.1.1 activity diagram of authentication	28
Figure 14: Level 1.4.3 activity diagram of Search	29
Figure 15: Level 1.3 activity diagram of music player	30
Figure 16: Level 1.4 activity diagram of personal account	31
Figure 17: Level 1.4.4 activity diagram of playlist.....	32
Figure 18: Level 1.5 activity diagram of administration.....	33
Figure 19: Level 1.1 swimlane diagram of homepage.....	34
Figure 20: Level 1.2 swimlane diagram of registration	35
Figure 21: Level 1.1.1 swimlane diagram of authentication.....	36
Figure 22: Level 1.4.3 swimlane diagram of searching songs.....	37

Figure 23: Level 1.3 swimlane diagram of music player.....	38
Figure 24: Level 1.4 swimlane diagram of personal account	39
Figure 25: Level 1.4.4 swimlane diagram of playlist	40
Figure 26: Level 1.5 swimlane diagram of administration	41
Figure 27: Data object relationship of Shworolipi	46
Figure 28: ER diagram of Shworolipi	47
Figure 29: class diagram of MSWA	66
Figure 30: State diagram of user class	69
Figure 31: State diagram of member class	69
Figure 32: State diagram of musicPlayer class	70
Figure 33: State diagram of non-member class	70
Figure 34: State diagram of song class.....	71
Figure 35: State diagram of playlist class	71
Figure 36: State diagram of admin class.....	72
Figure 37: Sequence diagram of homepage	73
Figure 38: Sequence diagram of personal account	74
Figure 39: Sequence diagram of administration	75
Figure 40: Sequence diagram of MSWA	76

LIST OF TABLES

Table 1: Noun Identification.....	42
Table 2: Attributes identification of final data objects.....	45
Table 3: Table translation of ER diagram.....	48
Table 4: Schema table of Song	48
Table 5: Schema table of Member.....	49
Table 6: Schema table of Playlist.....	49
Table 7: Schema table of Admin	50
Table 8: Schema table of adds	50
Table 9: Schema table of deletes	50
Table 10: Schema table of remove	51
Table 11: Class Identification with General Classification	53
Table 12: Class Identification with Selection Criteria	55
Table 13: Verb Identification.....	56
Table 14: Attributes and Methods of Preliminary Classes	58
Table 15: Attributes and Methods of User Class	60
Table 16: Attributes and Methods of Member Class	60
Table 17: Attributes and Methods of non-member Class	61
Table 18: Attributes and Methods of Admin Class.....	61
Table 19: Attributes and Methods of Song Class	61
Table 20: Attributes and Methods of MusicPlayer Class	62
Table 21: Attributes and Methods of Playlist Class.....	62
Table 22: Class Card of User Class.....	63
Table 23: Class Card of Member Class.....	63

Table 24: Class Card of non-member Class.....	64
Table 25: Class Card of Admin Class	64
Table 26: Class Card of MusicPlayer Class	64
Table 27: Class Card of MusicPlayer Class	64
Table 28: Class Card of Playlist Class	65
Table 29: Notations of class diagram.....	65
Table 30: Event identification.....	67

CHAPTER 1: INTRODUCTION OF THE MUSIC STREAMING WEB APPLICATION

1.1 Purpose

This document briefly describes the software requirement analysis of a music streaming web application. It contains normal, expected and exciting requirements and establishes a requirement baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered and organized. The SRS serves as an official means of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 Intended Audience

This SRS is intended for several audiences including the users as well as the admins, designers, developers, and testers.

- The designers will use this SRS as a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
- The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all of the customer's documented requirements.
- The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system

when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

- The customer will use this SRS to verify whether the product created by the developer team is acceptable to the customer.

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

CHAPTER 2: INCEPTION OF THE MUSIC STREAMING WEB APPLICATION

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved are. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of a software project. At project inception, we establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired and the effectiveness of preliminary communication and collaborations between the other stakeholders and the software team.

To establish the groundwork we have worked with the following factors related to the inception phases:

- List of stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Requirements questionnaire

2.1 List of Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that may be affected by its installation. At inception, a list of people who will contribute input as requirements is elicited. The initial list will grow as stakeholders are contacted because every stakeholder will be asked: "Whom else do you think we should talk to?"

To identify the stakeholders we consulted with the Admin of the web application and asked him following questions:

- Who is paying for the project?

- Who will be using the project outcomes?
- Who gets to make the decisions about the project?
- Who have resources we need to get the project done?
- Whose work will affect our project?

We identified the following stakeholders for our project. They are:

1. Admin
2. User
3. Designer
4. Developer

- i) Programmer
- ii) Tester

1. **Admin:** Admin will handle all the server side activities. They will upload and delete songs and their necessary information for the users.
2. **User:** They will use the application. They can search and play songs. They can also have their own account to have their own playlist and friend circle.
3. **Designer:** They will communicate with all the stakeholders and design according to the demands to provide the documents of the software.
4. **Developers:** Developers are one of the stakeholders because they are also affected by this system. They develop this system and work for further development. If there occurs any system interruption, they will find the problem and try to solve it
 - i. **Programmer:** They will code according to the supplied document
 - ii. **Tester:** They will test the software after programmers deliver the software.

2.2 Recognizing Multiple Viewpoints

Different stakeholders achieve different benefits from the system. Consequently, each of them has a different view of the system. So we have to recognize the requirements from multiple points of view, as well as multiple views of requirements. Assumptions are given below:

The viewpoint of Admin:

- Any user can search for desired songs with song's name or genre.
- Non-member users will be provided with a registration process to be members
- Members will be authenticated before logging in to their homepage
- Members will be provided with personal account
- Lyrics will be provided with the songs
- Members can make playlists with favorite songs
- Admin can upload new songs to the server with proper information
- Admin can delete any song
- Only members can create playlists
- There will be no sharing to the social networks
- No forum will be provided
- Only Bengali songs will be uploaded

The viewpoint of User:

- Any user can search for desired songs with song's name or genre.
- Members will be provided with personal account
- Users will get suggestions for more frequent songs in general homepage
- Members will get a suggestion for more frequent songs played by them in their personal Lyrics homepage.
- Members will be provided with the songs
- Members can make playlists with favorite songs
- Users can share songs to social networking sites
- A simple forum will be provided for the members
- Any user can create playlists
- All types of songs will be provided irrespective of language

The viewpoint of Designer:

- Suggestion will be provided during searching songs
- Allow members to add or edit their personal information
- Every confirmation message will be sent to email addresses of the users
- Members will receive email if anyone follows them
- Allow members to delete personal account
- Music player will be provided to play songs
- There will be no sharing to the social networks
- No forum will be provided
- A guideline option will be there for the convenience of the members
- Provide admin a corresponding interface for special access

The viewpoint of Developer:

- Cost within budget
- Detail documentation
- Enough time for development

2.3 Working towards Collaboration

Every stakeholder has their own requirements. There are some common and conflicting requirements of our stakeholder. That's why we followed the following steps to merge these requirements-

- Find the common and conflicting requirements
- Categorize them
- List the requirements based on stakeholder's priority points
- Make final decision about requirements

Common requirements:

- Any user can search for desired songs with song's name or genre.
- Members will be provided with personal account
- Users will get suggestions for more frequent songs in general homepage
- Lyrics will be provided with the songs
- Members can make playlists with favorite songs

Conflicting requirements:

- Any user can create playlists
- Only members can create playlists
- Users can share songs to social networking sites
- no sharing opportunity will be provided to the social networks
- A simple forum will be provided for the members
- No forum will be provided
- Only Bengali songs will be uploaded
- All types of songs will be provided irrespective of language

Final requirements:

We finalize the following requirements based on stakeholder's priority point:

- Any user can search for desired songs with song's name or genre.
- Non-member users will be provided with a registration process to be members
- Members will be provided with personal account
- Members will be authenticated before logging in to their account
- Members will get suggestions for popular songs of this website in their personal homepage.
- Lyrics will be provided with the songs
- Members can make playlists with favorite songs
- Admin can upload new songs to the server with proper information
- Admin can delete any song

2.4 Requirements Questionnaire

We first ask the stakeholder some context-free questions to understand the project's overall performance and goals. These questions are mentioned in section 2.1.1. These questions help us to identify the stakeholders of the project. Then we ask our next set of questions to better understand the problem and take stakeholder's opinion about the solution. The final set of question focused on the effectiveness of the communication activity itself.

The Inception phase helped us to establish a basic understanding of the web application, identify the stakeholders who will be benefited if this system becomes automated, define

the nature of the system and the tasks done by the system, and establish a preliminary communication with our stakeholders.

In our project, we have established a basic understanding of the problem, the nature of the solution that is desired and the effectiveness of preliminary communication and collaboration between the stakeholders and the software team. More studies and communication will help both sides (developer and client) to understand the future prospect of the project. Our team believes that the full functioning document will help us to define that future prospect.

CHAPTER 3: ELICITATION OF THE MUSIC STREAMING WEB APPLICATION

3.1 Quality Function Deployment (QFD)

QFD is the methods or quality management technique that translate the needs of the customer into technical requirements for the software. Ultimately the goal of QFD is to translate often subjective quality criteria into objective ones that can be quantified and measured and which can then be used to design and manufacture the product. It is a method for maximizing customer satisfaction from the software engineering process. We followed this methodology to identify the requirements for the project which are given below:

3.1.1 NORMAL REQUIREMENTS

Normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customers. The presence of these requirements fulfills satisfaction to the customers. These are the normal requirements for our project.

- Any user can search for desired songs with song's name, artist and genre from the database of this website
- Non-member users will be provided with a registration process to become members
- Members will be authenticated before logging in to their account
- Members will be provided with personal account containing name, date of birth, gender and phone number
- Members will get suggestions for popular songs of this website in their personal homepage.
- Members can make playlists with their favorite songs
- Admin can upload new songs to the server with proper information regarding title, artist name, genre, lyric and mood
- Admin can delete any song

3.1.2 EXPECTED REQUIREMENTS

These requirements are intrinsic to the product or system and may be so elementary that the customer does not explicitly state them. Their absence will be a cause for significant dissatisfaction. Below the expected requirements for our project are briefly described.

- Allow members to add or edit their personal information
- Allow members to delete account
- Music player will be provided to play songs
- Provide admin a corresponding interface for special access

3.1.3 EXCITING REQUIREMENTS

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present. Following are some exciting requirements of our project.

- A guideline option will be there for the convenience of the members

3.2 Usage Scenario of the music streaming web application

"Shworolipi" is a music streaming web application. The features of this site are searching Bengali songs, providing a registration process to be a member, managing the accounts, playlists of the members and providing admins with a separate interface for their administrative activities.

3.2.1 HOMEPAGE

Users will find a search option to search songs in the general homepage. The user can find options to sign up and sign in. A guideline option will be there for the convenience of the members.

3.2.2 SEARCHING SONGS

There will be member user and non-member user. Every user can search for the songs in this sites. Users can search songs by- title, artist, genre and mood. They will also get suggestions during searching. The user can play the song after finding it.

3.2.3 MUSIC PLAYER

Necessary information to store any song is- title, artist, genre, mood and lyrics. Also, to keep the track of a popular song we have to store its popularity value. Every song will be in mp3 format. There will be a music player to play those songs. The user can play, pause, repeat, control volume and also select next or previous songs. When next or previous song will be selected, the system will search it in the database. If it is not, then it will replay the songs again. The user can also check the lyric and other information.

3.2.4 REGISTRATION

A user can register to be a member of this website. This procedure will require an email address and password of the user. The system will check whether any account exists

under the same email or not. If any account exists under the same email address, it will prompt the user. If the request is valid, an account will be created for the member. The applicant will receive a confirmation mail to his/her email addresses.

3.2.5 AUTHENTICATION

A member can enter to his/her personal homepage after being authenticated by the email address and password. If the email address and password do not match, he/she will be prompted for the wrong password. One can try again or select forgot password option. If anyone chooses the "forgot password" option, he/she will get an email containing the password. He/she will be prompted to check email.

3.2.6 PERSONAL ACCOUNT

Member's account may contain- name, date of birth, gender and phone number. One may add or edit their personal information. Members can log out after they finished their activities. One may delete one's account from this service permanently.

3.2.7 PLAYLIST

Members can create playlists to store their favorite songs. To create a playlist, they have to enter a name. They will be provided the song list, from where they can search for songs and can add them to the playlists. They can search songs from the playlist and also play songs from their playlists. They can delete songs from the playlist or the whole playlist.

3.2.8 ADMIN SECTION

Admin will have a different interface for their convenience. They will have to login into the system to perform the administrative activities. One will be given a fixed email id to enter as an admin. They can play songs too. They can edit passwords and name. They can remove any user they want. For that, they have to enter the email address of that member and reason behind the removal. They can upload or delete songs and database will be updated accordingly. Also, the email id of that admin will be stored for each of this action. Admins can log out after they finished their activities.

CHAPTER 4: SCENARIO-BASED MODEL OF THE MUSIC STREAMING WEB APPLICATION

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If we understand how end users (and other actors) want to interact with a system, our software team will be able to characterize requirements and build meaningful analysis and design models properly. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swim lane diagrams.

4.1 Definition of Use case

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users.

The first step in writing a Use Case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using the system.

Primary Actor

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

Secondary Actor

Secondary actors support the system so that primary actors can do their work. They either produce or consume information

4.2 Use Case Diagrams

Use Case diagrams give the non-technical view of the overall system.

4.2.1 LEVEL-0 USE CASE DIAGRAM-SHWOROLIPI

In figure 1, the level 0 diagram of MSWA is shown-

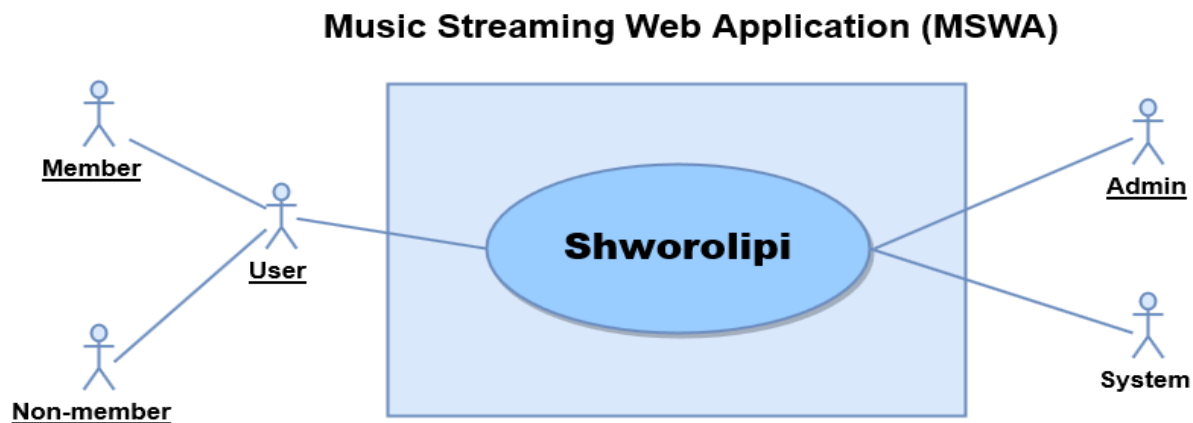


Figure 1: Level 0 Use Case diagram of MSWA

Name	Shworolipi
ID	SHWOROLIPI-L-0
Primary Actors	Member, Non-member, User, Admin
Secondary Actors	System

Description of Use Case Diagram Level 0:

After analyzing the user story we found five actors who will directly use the system as a system operator. Primary actors are those who will play action and get a reply from the system whereas secondary actors only produce or consume information.

Following are the actors of Shworolipi –

1. Member (primary)
2. Non-member (secondary)
3. System (primary)
4. Admin (secondary)
5. User (secondary)

4.2.2 LEVEL-1 USE CASE DIAGRAM-SUBSYSTEMS OF SHWOROLIPi

In figure 2, the level 1 diagram of sub-systems of Shworolipi is shown-

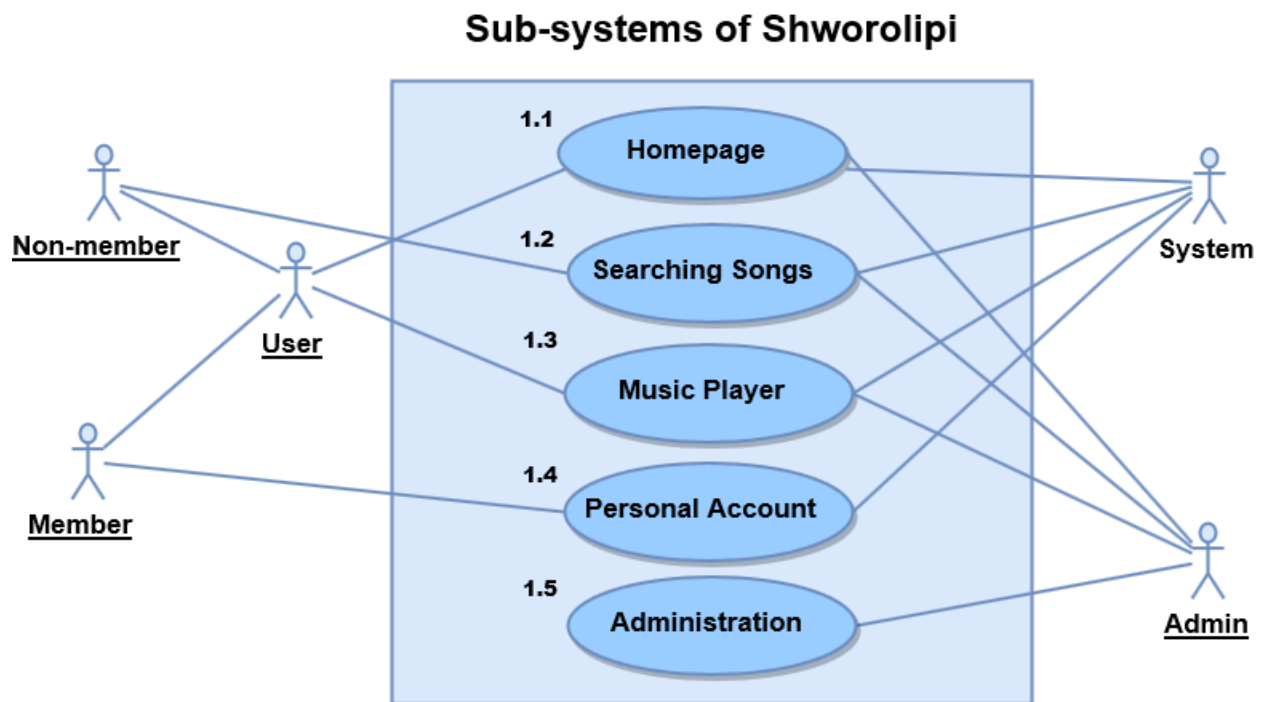


Figure 2: Level 1 Use Case Diagram-Subsystems of Shworolipi

Name	Subsystems of Shworolipi
ID	SHWOROLIPi-L-1
Primary Actors	Member, Non-member, User, Admin
Secondary Actors	System

Description of Use Case Diagram Level 1:

There are five subsystems in the Shworolipi. They are:

1. Homepage
2. Searching songs
3. Music player
4. Personal account
5. Administration

4.2.3 LEVEL-1.1 USE CASE DIAGRAM-HOMEPAGE

In figure 3, the level 1.1 diagram of MSWA is shown-

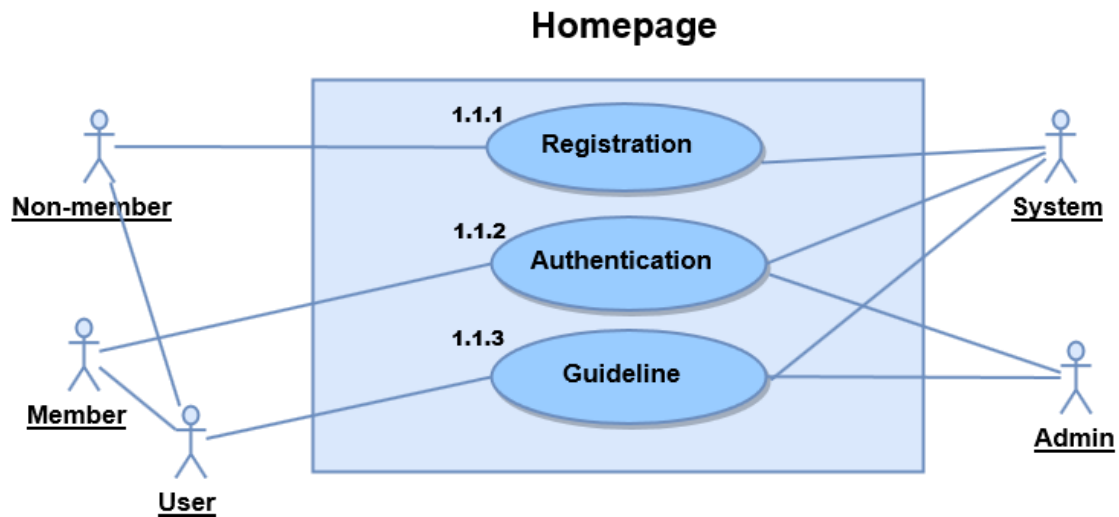


Figure 3: Level 1.1 Use Case Diagram of Homepage

Name	Homepage
ID	SHWOROLIPI-L-1.1
Primary Actors	User, Member, Non-member, System

Description of Use Case Diagram Level 1.1:

Users will find popular songs in the general homepage. The user can find options to sign up and sign in. Also, there will be a search option to search songs. A guideline option will be there for the convenience of the members.

The homepage subsystem of Shworolipi can be divided into three parts. They are:

1. Registration
2. Authentication
3. Guideline

4.2.4 LEVEL-1.1.1 USE CASE DIAGRAM-REGISTRATION

In figure 4, the level 1.1.1 diagram of MSWA is shown-



Figure 4: Level 1.2 Use Case Diagram of Registration

Name	Registration
ID	SHWOROLIPI-L-1.1.1
Primary Actors	System
Secondary Actors	Non-member

Description of Use Case Diagram Level 1.1.1:

The user can register to be a member of this website. This procedure will require an email address and password of the user. The system will check whether any account exists under the same email or not. If any account exists under the same email address, it will prompt the user. If the request is valid, an account will be created for the member. The applicant will receive a confirmation mail to his/her email addresses.

Action-Reply of Use Case Diagram Level 1.1.1:

- Action 1: User will enter email address and password to register.
Reply 1: System will check whether any personal account exists under the same email or not. If the request is valid, the applicant will receive a confirmation mail to his/her email address.

4.2.5 LEVEL-1.1.2 USE CASE DIAGRAM-AUTHENTICATION

In figure 5, the level 1.1.2 diagram of MSWA is shown-

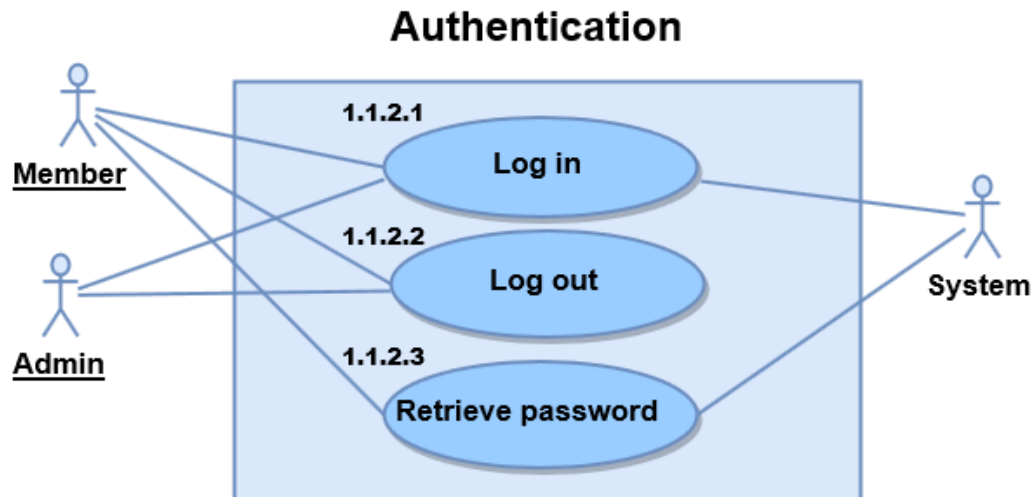


Figure 5: Level 1.1.1 Use Case Diagram of Authentication

Name	Authentication
ID	SHWOROLIP-L-1.1.2
Primary Actors	Member, Admin
Secondary Actors	System

Description of Use Case Diagram Level 1.1.2:

A member can enter to their personal homepage after being authenticated by the email address and password. If the email address and password do not match, he/she will be prompted for the wrong password. One can try again or select forgot password option. If anyone chooses the "forgot password" option, he/she will get an email containing the password. Then, he/she will be prompted to check email.

Action-Reply of Use Case Diagram Level 1.1.2:

- Action 1: Member/Admin will enter an email address and password to log in.
Reply 1: He/she will be allowed to enter into the system upon entering correct credentials.

- Action 2: Member/Admin given password is wrong.
Reply 2: Allow to retrieve the password through email

4.2.6 LEVEL-1.2 USE CASE DIAGRAM-SEARCHING SONGS

In figure 6, the level 1.2 diagram of MSWA is shown-

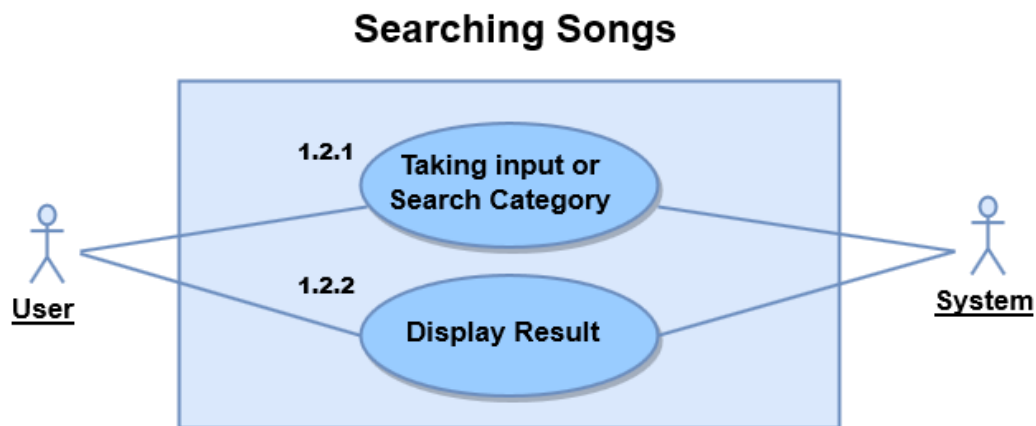


Figure 6: Level 1.4.3 Use Case Diagram of Searching songs

Name	Searching songs
ID	SHWOROLPI-L-1.2
Primary Actors	User, System

Description of Use Case Diagram Level 1.2:

There will be two types of users- member and non-member. Every user can search for the songs in this sites. Users can search songs by- title, artist, genre and mood. They will also get suggestions during searching. The user may play the songs after finding it.

Action-Reply of Use Case Diagram Level 1.2:

- Action 1: Member will search for songs from the database
Reply 1: System will provide result.

4.2.7 LEVEL-1.3 USE CASE DIAGRAM-MUSIC PLAYER

In figure 7, the level 1.3 diagram of MSWA is shown-

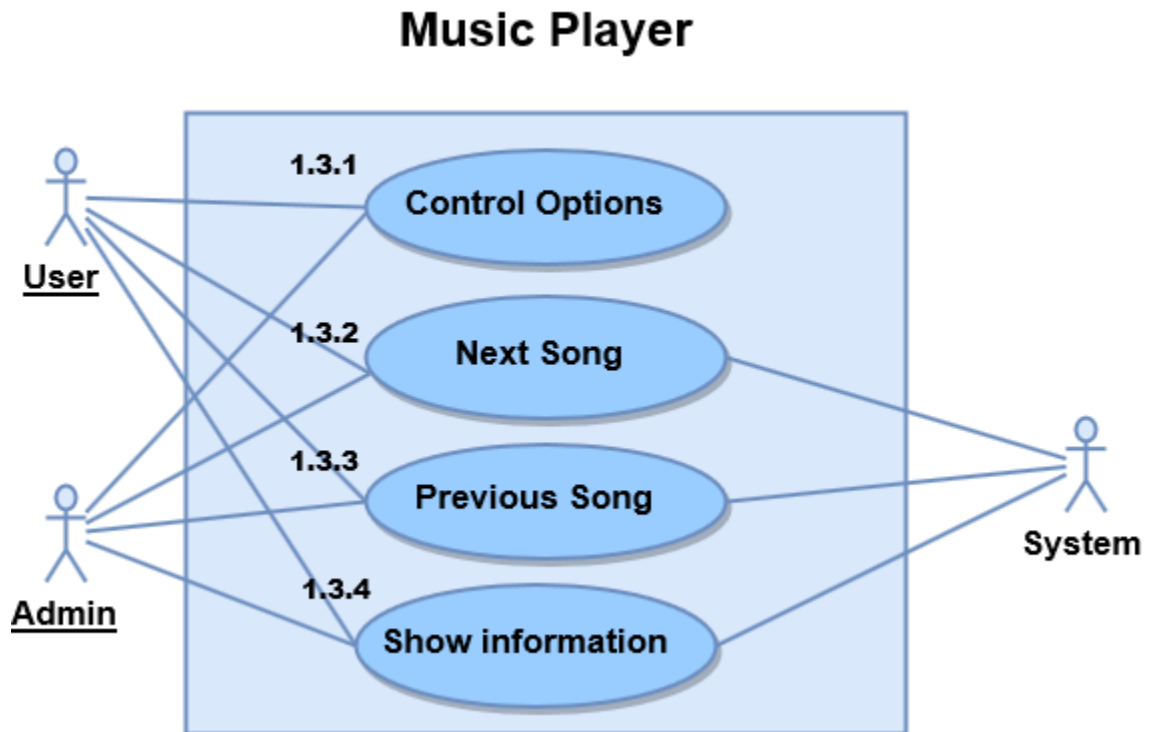


Figure 7: Level 1.3 Use Case Diagram of Music Player

Name	Music player
ID	SHWOROLIP-L-1.3
Primary Actors	User, System
Secondary Actors	Member

Description of Use Case Diagram Level 1.3:

Necessary information to store any song is- title, artist, genre, mood and lyrics. Also, to keep the track of a popular song we have to store its popularity value. Every song will be in mp3 format. There will be a music player to play those songs. When any song will be played, all the information will be shown on the screen. The user can play, pause, repeat, control volume and also select next or previous songs. When next or previous song will be

selected, the system will search it in the database. If it is not, then it will prompt the user. The user can check the lyric of a song.

Action-Reply of Use Case Diagram Level 1.3:

- Action 1: User will play a song.
Reply 1: System will show the information on the screen.
- Action 2: User will select a next or previous song.
Reply 2: System will search it in the database. If it is not, then it will prompt the user.
- Action 3: User can check for the lyric of a song.
Reply 3: System will show the lyric.

4.2.8 LEVEL-1.4 USE CASE DIAGRAM-PERSONAL ACCOUNT

In figure 8, the level 1.4 diagram of MSWA is shown-



Figure 8: Level 1.4 Use Case Diagram of Personal Account

Name	Personal Account
ID	SHWOROLIP-L-1.4
Primary Actors	Member, System

Description of Use Case Diagram Level 1.4:

Member's account may contain- name, date of birth, gender and phone number. One may add or edit their personal information. One may delete one's account from this service permanently. Members can create playlist and manage those.

Action-Reply of Use Case Diagram Level 1.4:

- Action 1: User may add or edit their personal information.
Reply 1: System will provide that interface.
- Action 2: Member can delete his/her personal account.
Reply 2: System will update the database.
- Action 3: Member may manage playlists
Reply 3: System will update database accordingly

4.2.9 LEVEL-1.4.3 USE CASE DIAGRAM-PLAYLIST

In figure 9, the level 1.4.3 diagram of MSWA is shown-

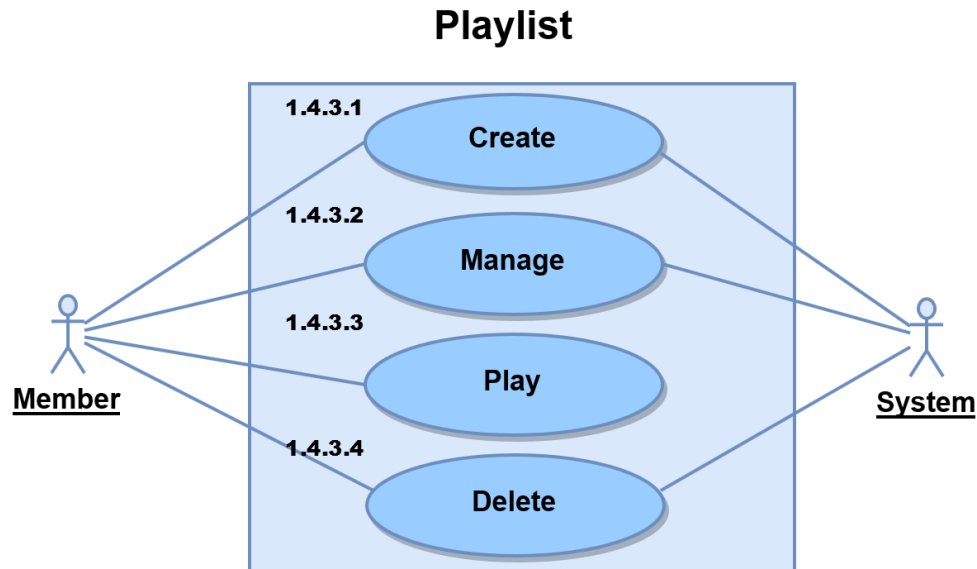


Figure 9: Level 1.4.3 Use Case Diagram of Playlist

Name	Playlist
ID	SHWOROLIPI-L-1.4.3
Primary Actors	Member, System

Description of Use Case Diagram Level 1.4.3:

Members can create playlists for their favorite songs. To create a playlist, they have to enter a name. Members can add songs to any playlist while listening to the songs from the music player or after searching any songs. They can search songs from the playlist and also play songs from their playlists. They can delete songs from the playlist or the whole playlist.

Action-Reply of Use Case Diagram Level 1.4.3:

- Action 1: Member will enter a name of a playlist for a new playlist.
Reply 1: System will update the database.

- Action 2: Member can update the playlist.
Reply 2: System will update the database.
- Action 3: Member can play full playlist.
Reply 3: System will connect the music player.
- Action 4: Member can delete playlist.
Reply 4: System will update the database.

4.2.9 LEVEL-1.5 USE CASE DIAGRAM- ADMINISTRATION

In figure 10, the level 1.5 diagram of MSHA is shown-

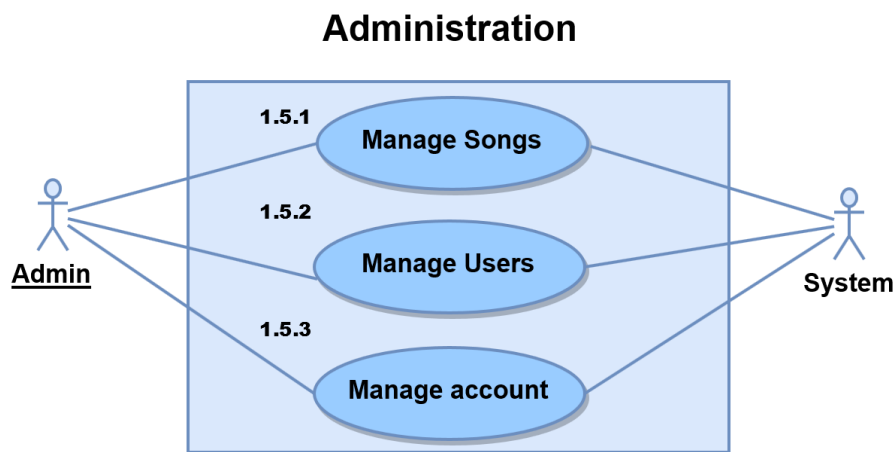


Figure 10: Level 1.5 Use Case Diagram of Administration

Name	Administration
ID	SHWOROLIP-L-1.5
Primary Actors	Admin
Secondary Actors	System

Description of Use Case Diagram Level 1.5:

Admin will have a different interface for their convenience. They will have to login into the system to perform the administrative activities. They can remove any user they want. They can upload or delete songs to the server side and database will be updated accordingly.

Action-Reply of Use Case Diagram Level 1.5.:

- Action 1: Admin will upload or delete songs to the server side.
Reply 1: System will update the database.
- Action 2: Admin can remove members.
Reply 2: System will update the database.

4.3 Activity Diagrams of MSWA

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows).

The activity diagrams of the modules described in the previous chapter is shown in the following figures:

4.3.1 ACTIVITY DIAGRAM OF HOMEPAGE

In figure 11, the activity diagram of Homepage is shown-

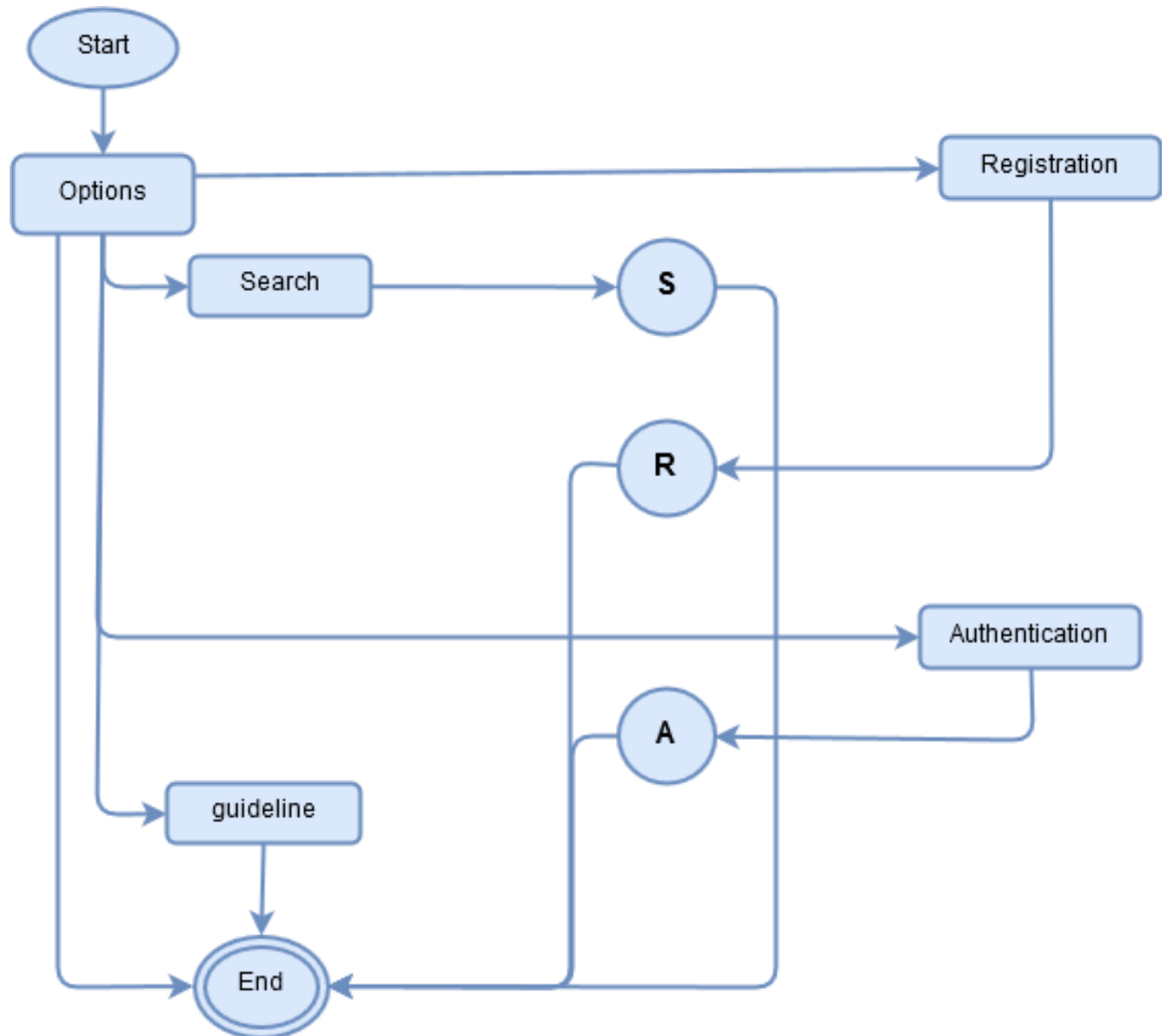


Figure 11: Level 1.1 activity diagram of homepage

4.3.2 ACTIVITY DIAGRAM OF REGISTRATION

In figure 12, the activity diagram of registration is shown-

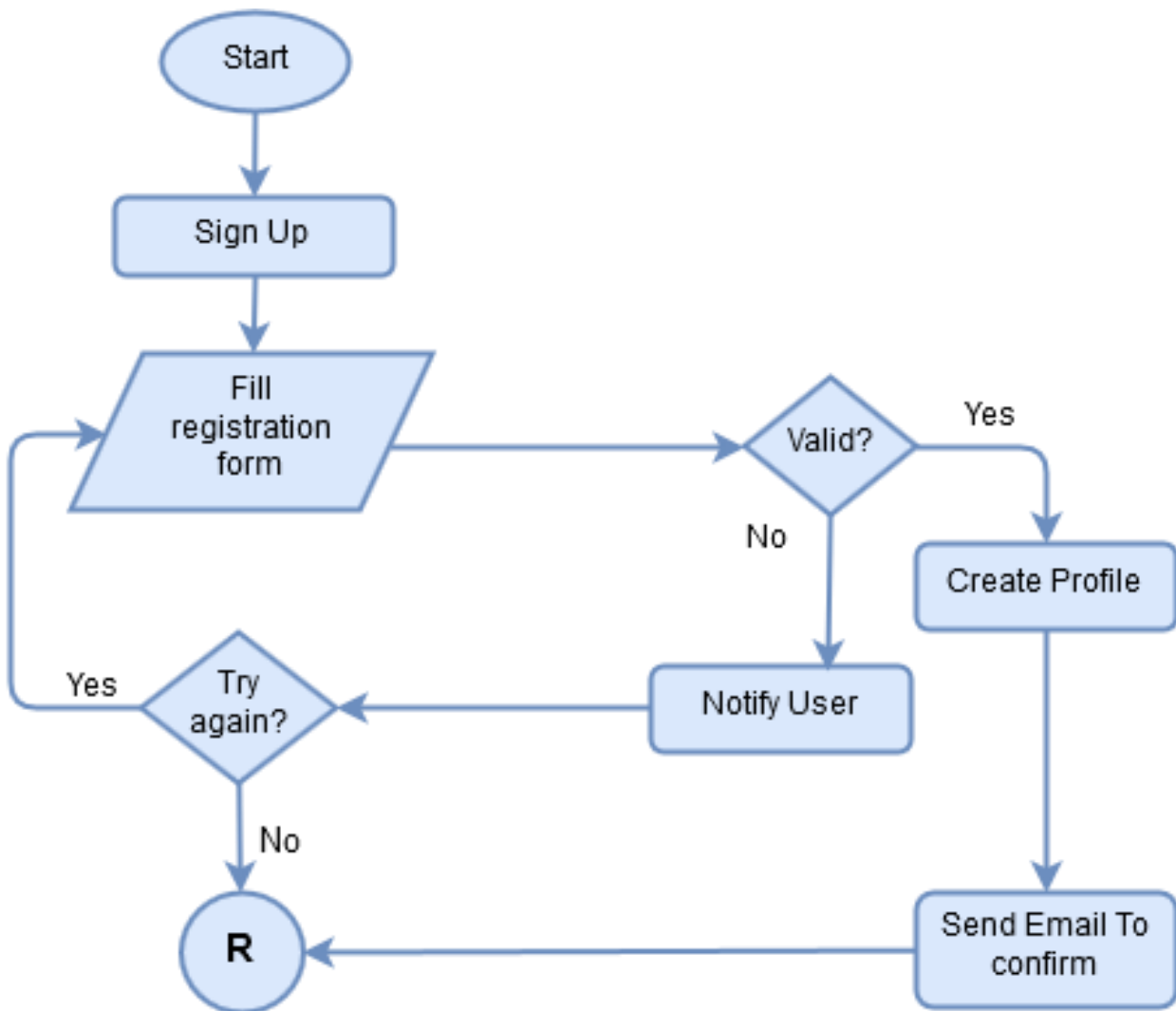


Figure 12: Level 1.2 activity diagram of registration

4.3.3 ACTIVITY DIAGRAM OF AUTHENTICATION

In figure 13, the activity diagram of authentication is shown-

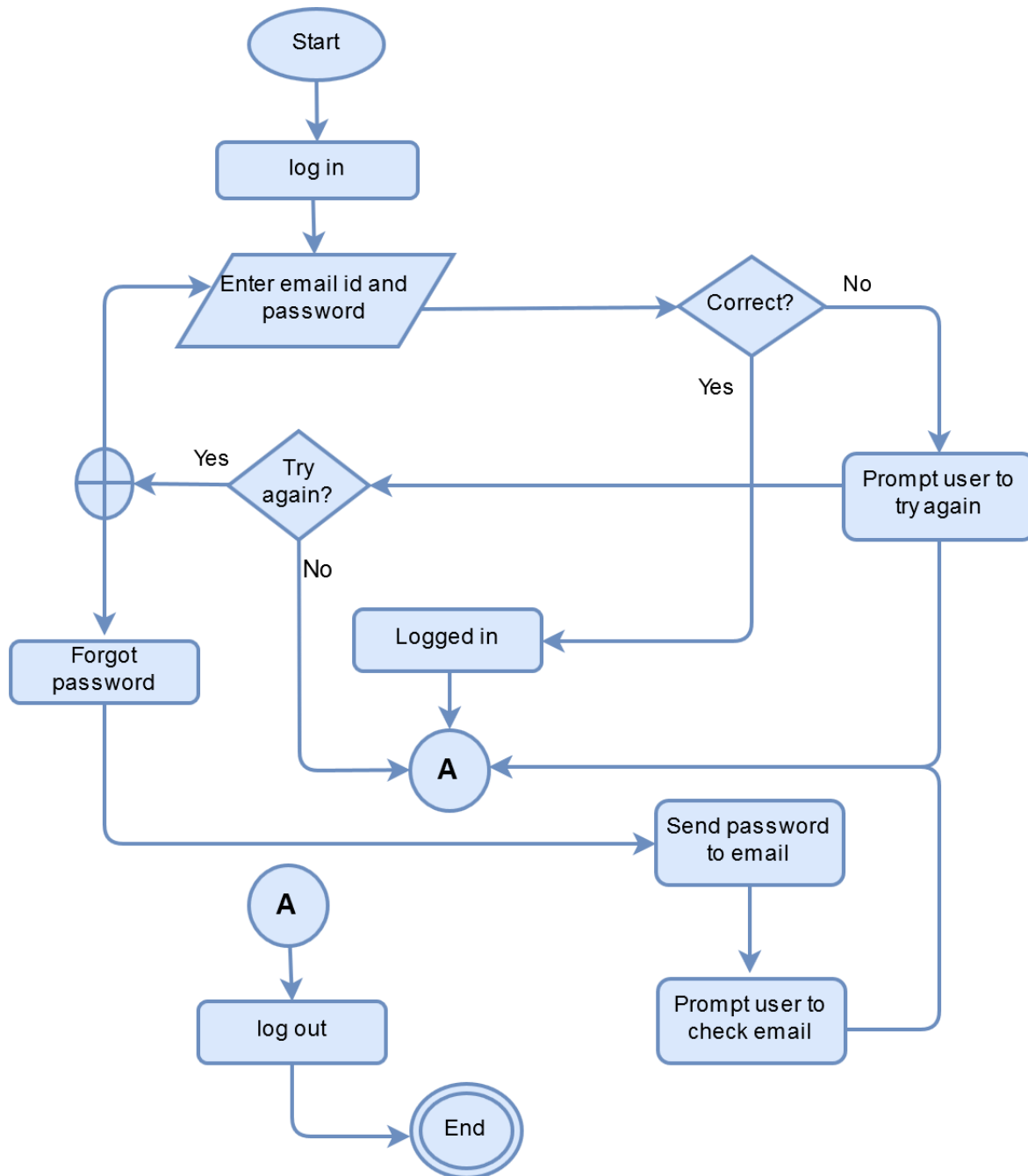


Figure 13: Level 1.1.1 activity diagram of authentication

4.3.4 ACTIVITY DIAGRAM OF SEARCHING SONGS

In figure 14, the activity diagram of searching songs is shown-



Figure 14: Level 1.4.3 activity diagram of Search

4.3.5 ACTIVITY DIAGRAM OF MUSIC PLAYER

In figure 15, the activity diagram of music player is shown-

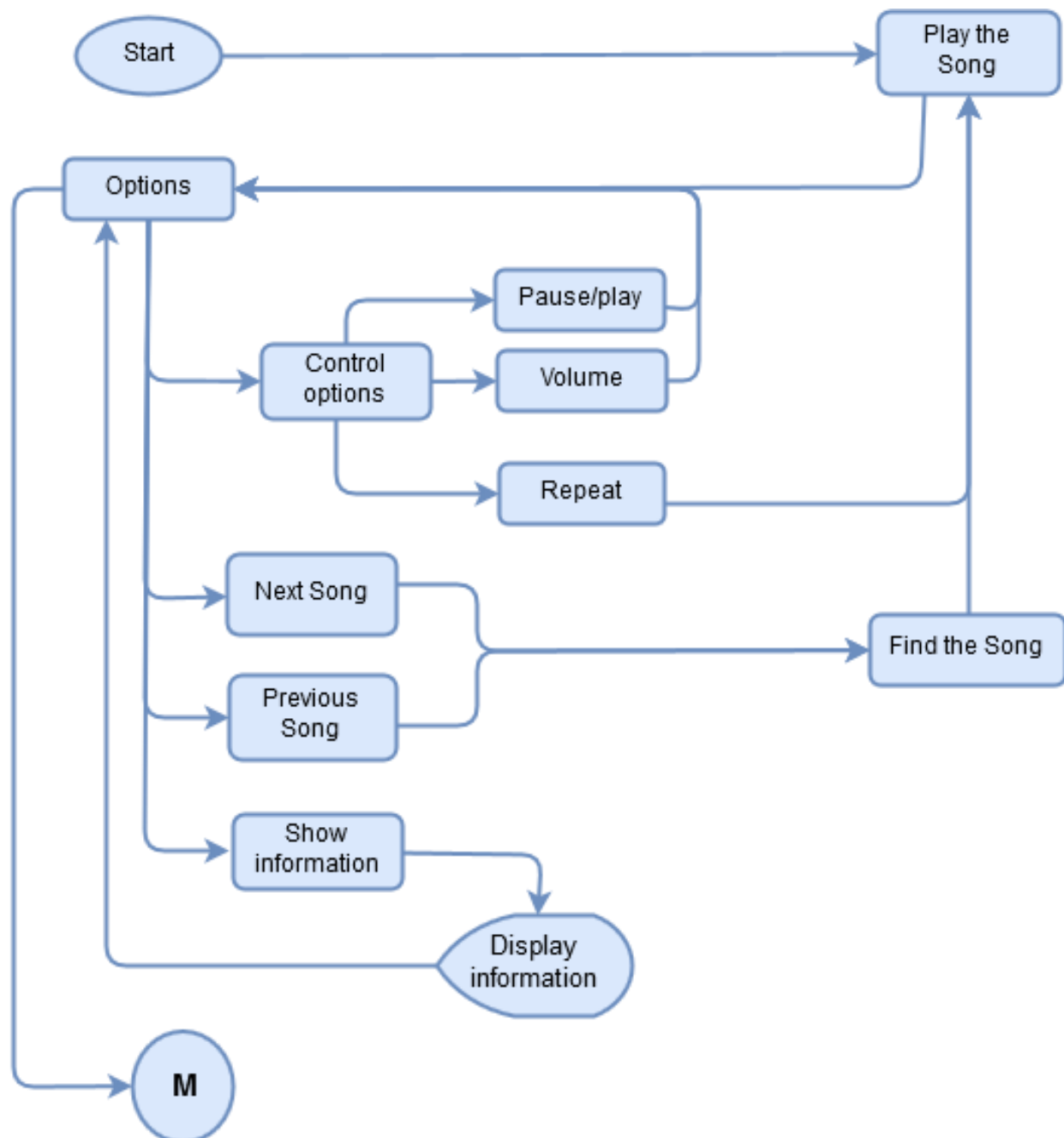


Figure 15: Level 1.3 activity diagram of music player

4.3.6 ACTIVITY DIAGRAM OF PERSONAL ACCOUNT

In figure 16, the activity diagram of personal account is shown-

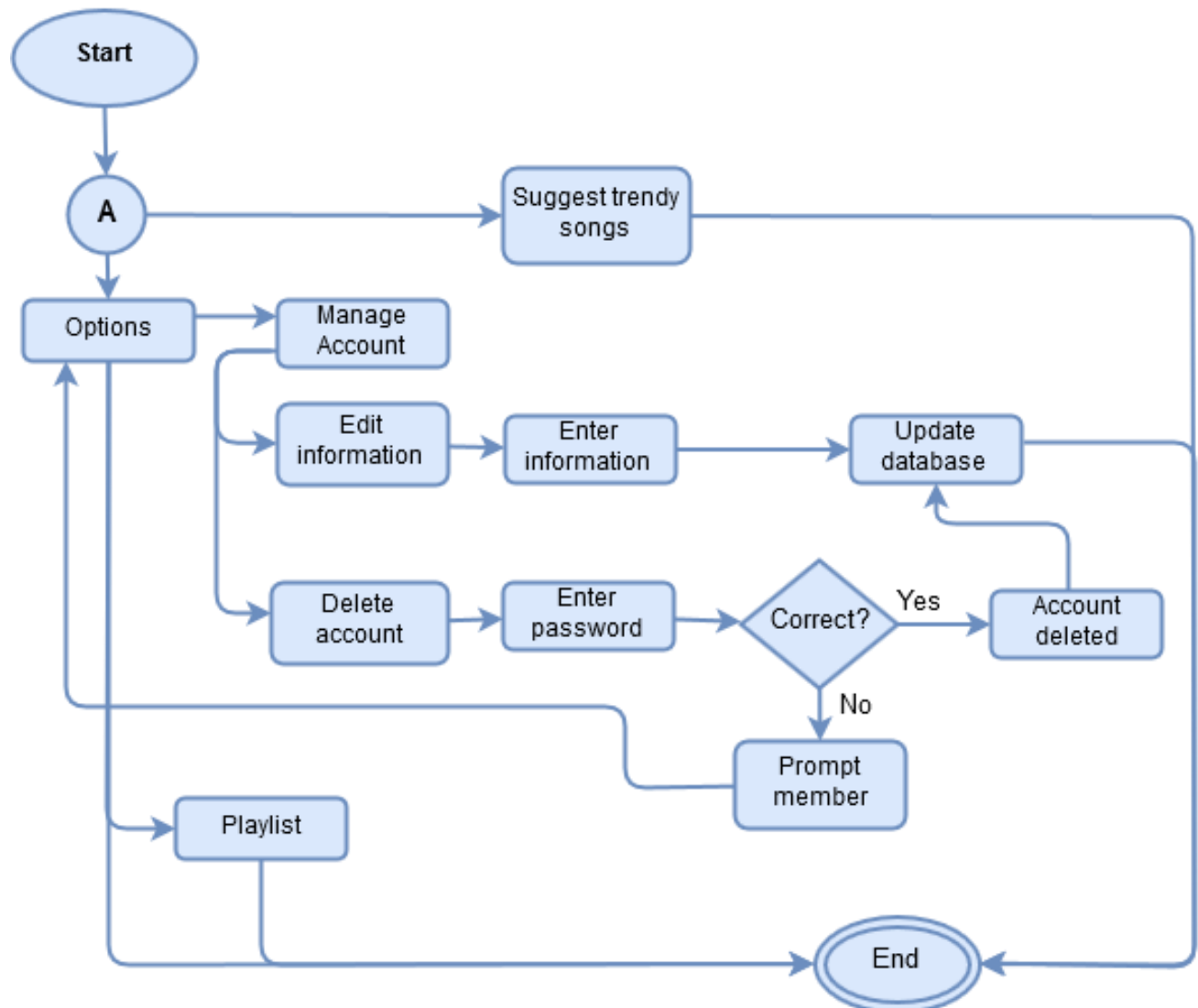


Figure 16: Level 1.4 activity diagram of personal account

4.3.7 ACTIVITY DIAGRAM OF PLAYLIST

In figure 17, the activity diagram of playlist is shown-

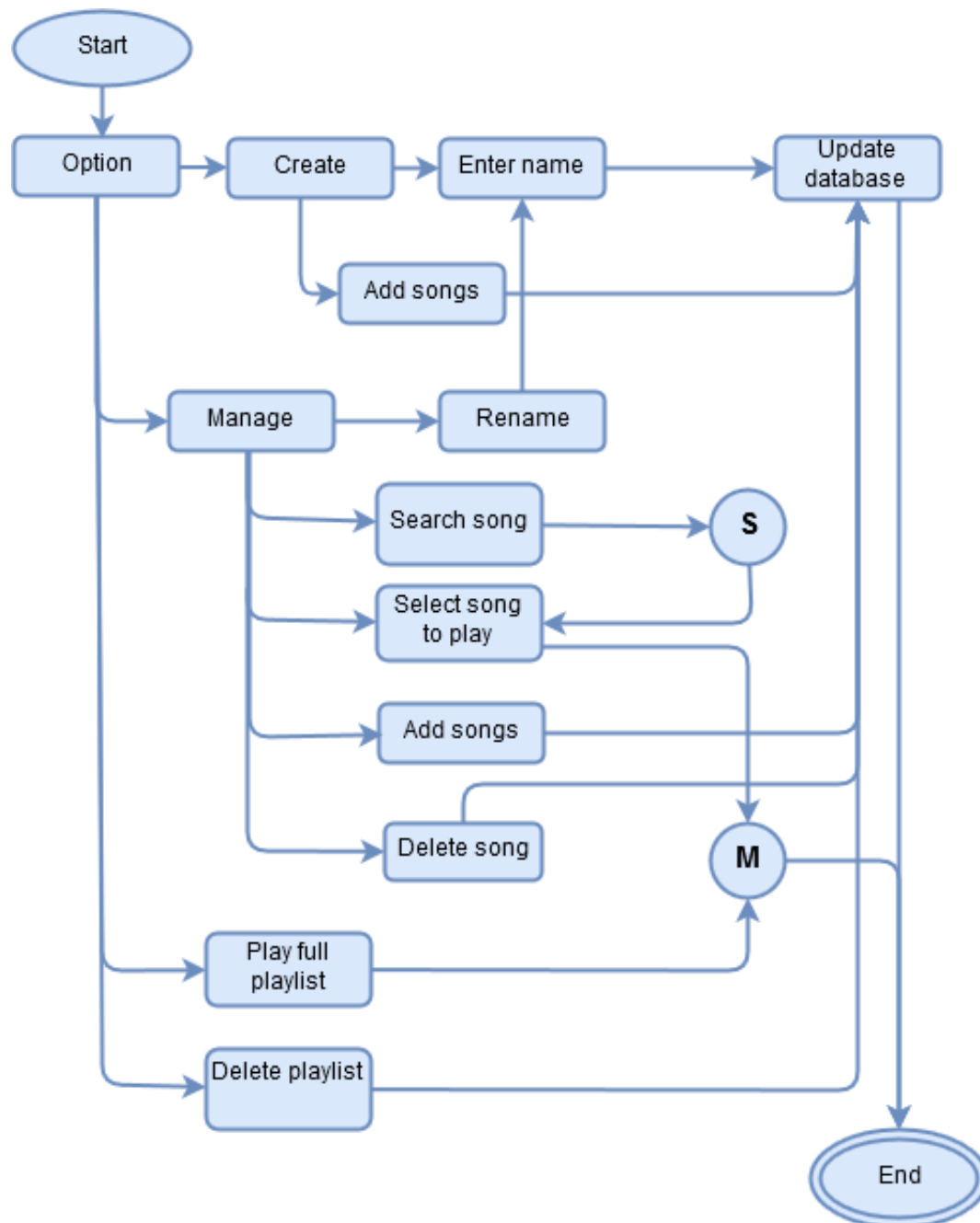


Figure 17: Level 1.4.4 activity diagram of playlist

4.3.8 ACTIVITY DIAGRAM OF ADMINISTRATION

In figure 18, the activity diagram of administration is shown-

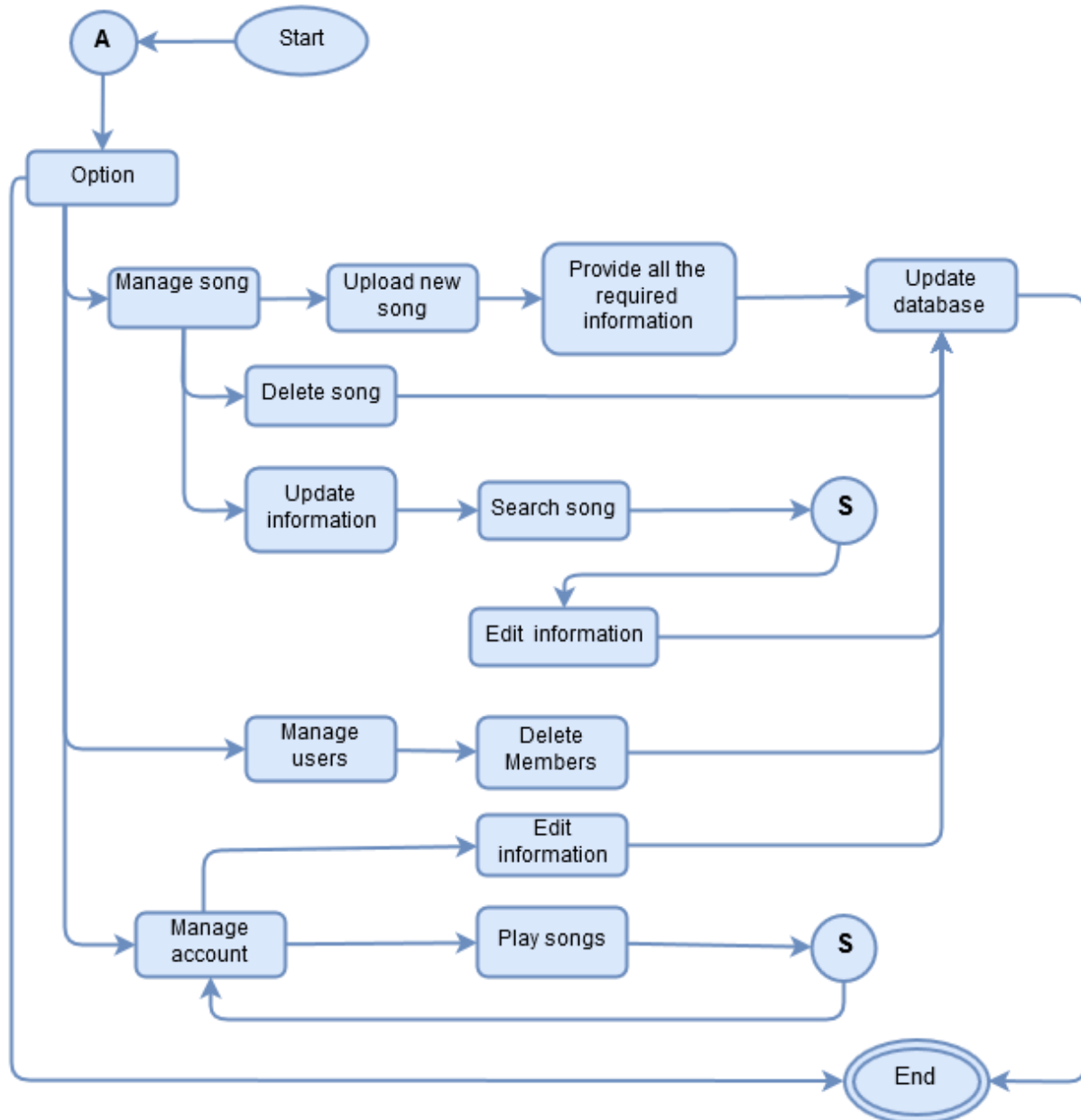


Figure 18: Level 1.5 activity diagram of administration

4.4 Swimlane Diagrams of MSWA

A swimlane diagram is a visual element used in process flow diagrams, or flowcharts, which visually distinguishes job sharing and responsibilities for sub-processes of a business process.

The swimlane diagrams of the modules described in the previous chapter is shown below:

4.4.1 SWIMLANE DIAGRAM OF HOMEPAGE

In figure 19, the swimlane diagram of homepage is shown

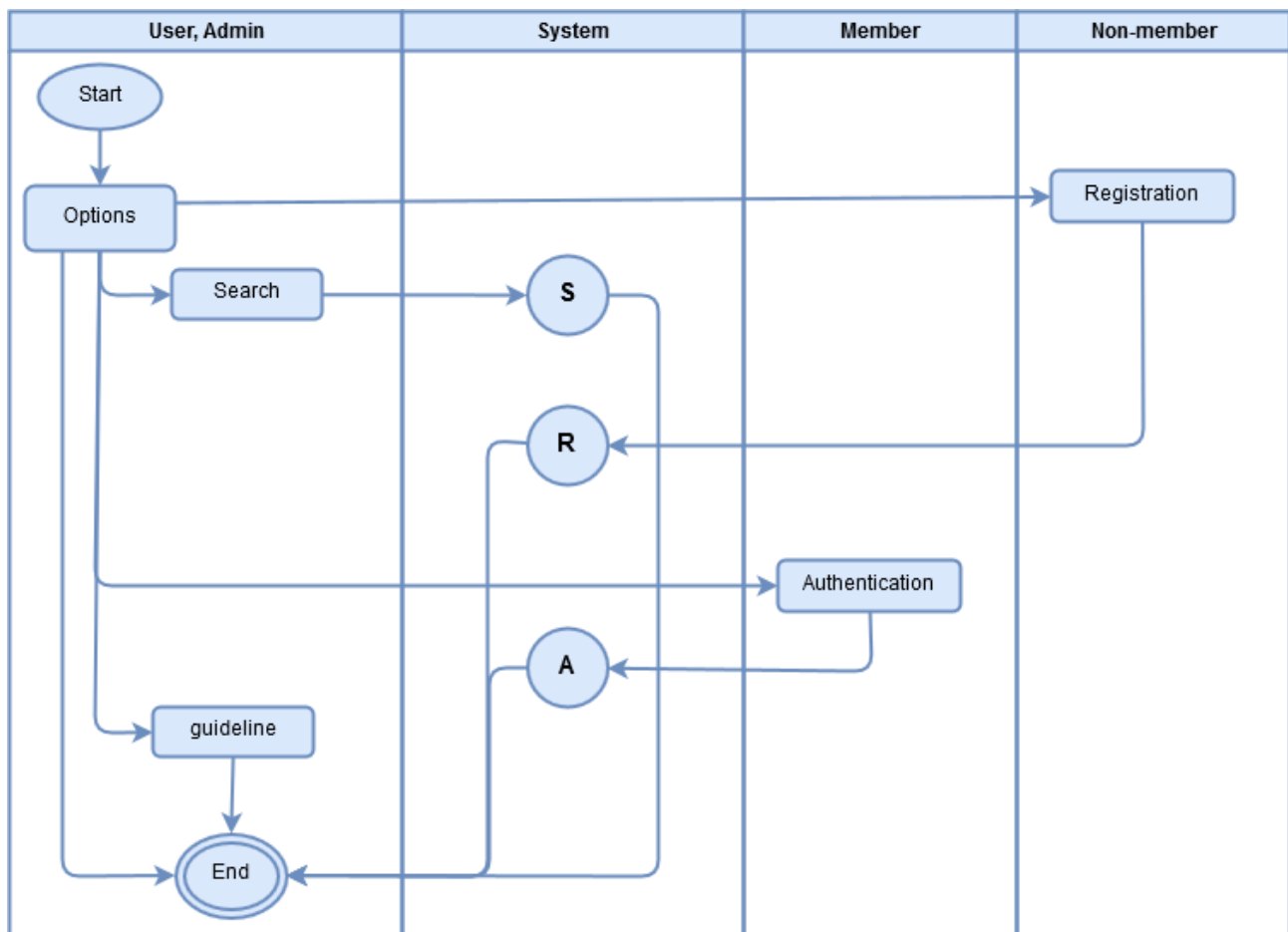


Figure 19: Level 1.1 swimlane diagram of homepage

4.4.2 SWIMLANE DIAGRAM OF REGISTRATION

In figure 20, the swimlane diagram of registration is shown

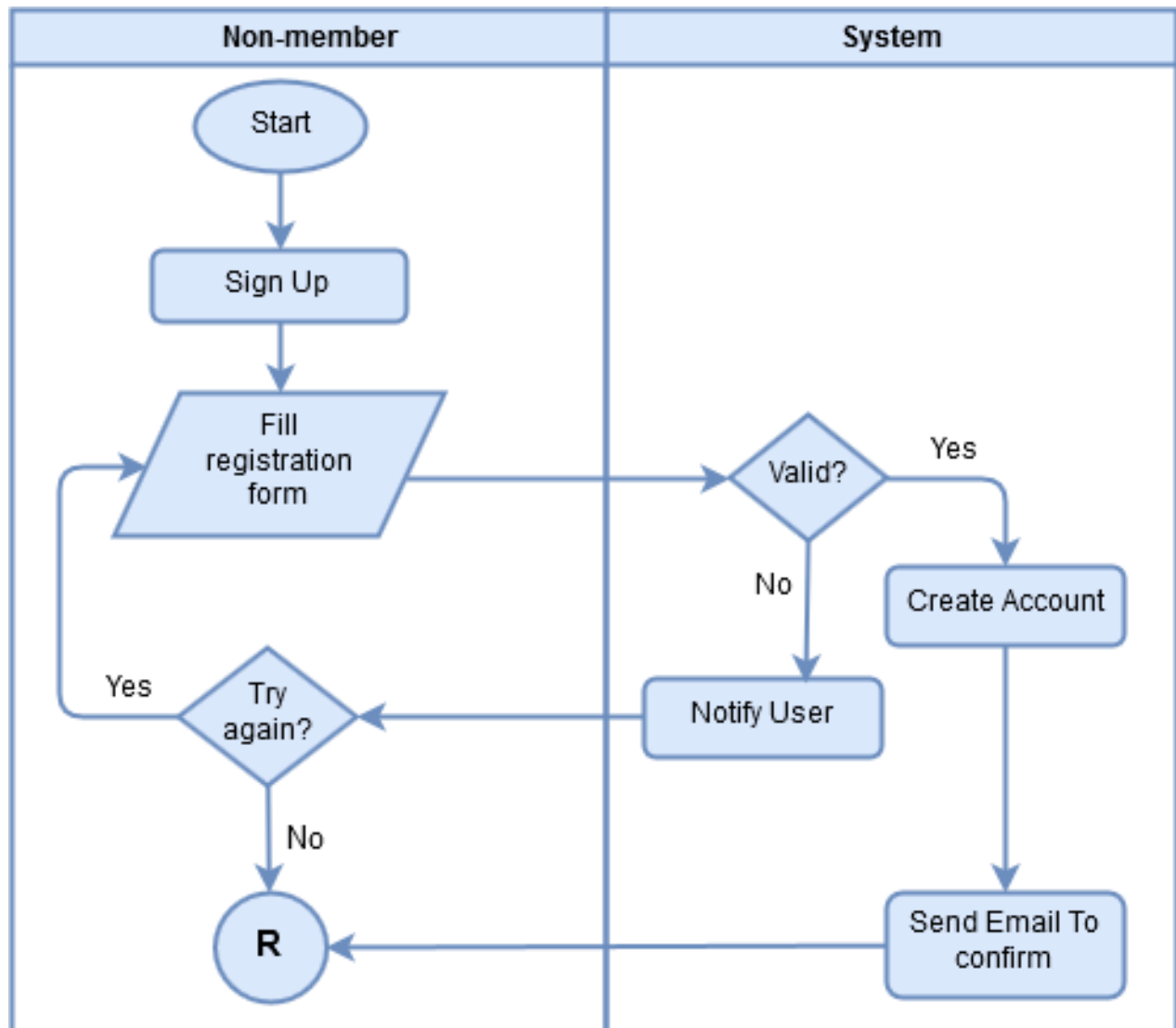


Figure 20: Level 1.2 swimlane diagram of registration

4.4.3 SWIMLANE DIAGRAM OF AUTHENTICATION

In figure 21, the swimlane diagram of authentication is shown

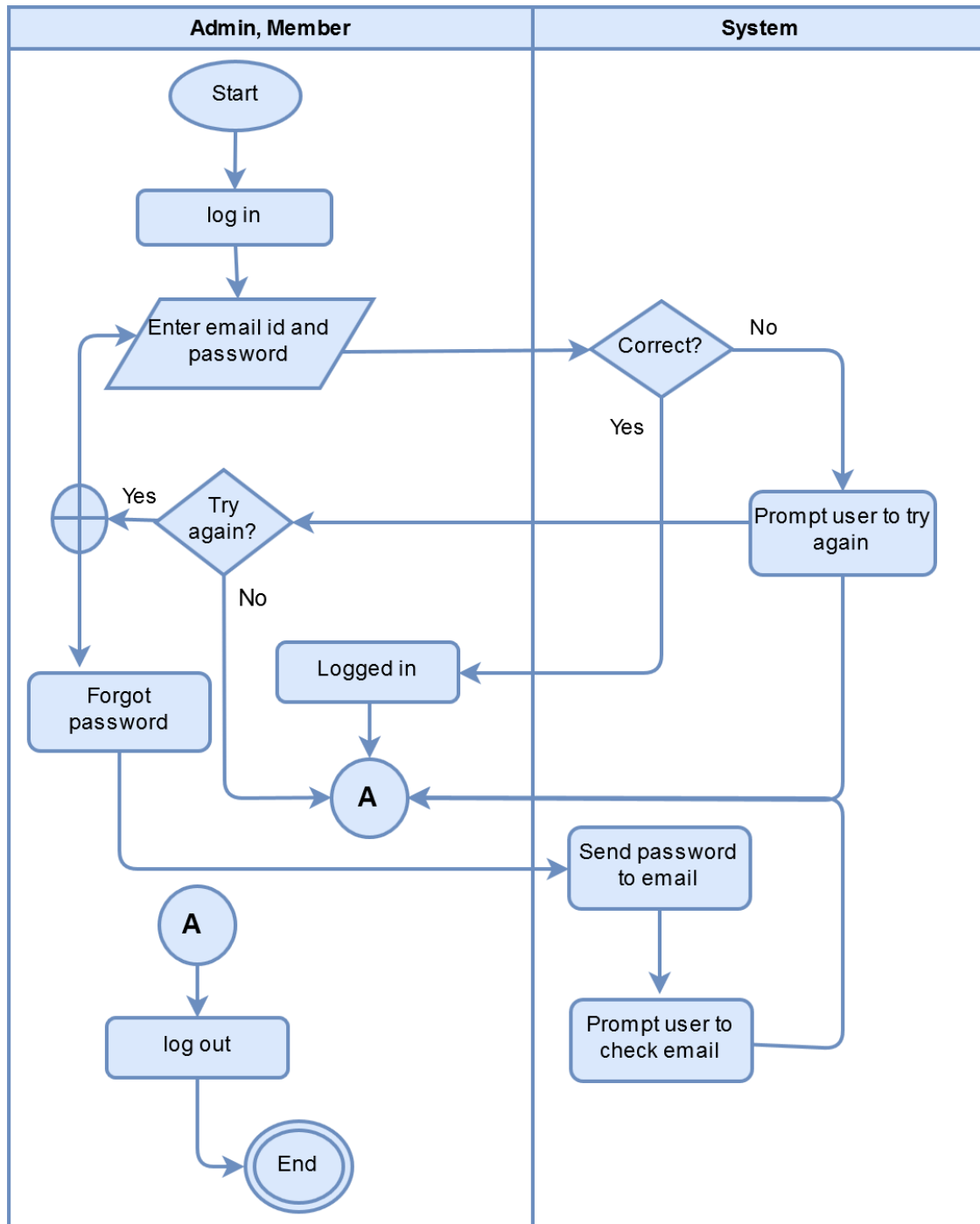


Figure 21: Level 1.1.1 swimlane diagram of authentication

4.4.4 SWIMLANE DIAGRAM OF SEARCHING SONGS

In figure 22, the swimlane diagram of searching songs is shown

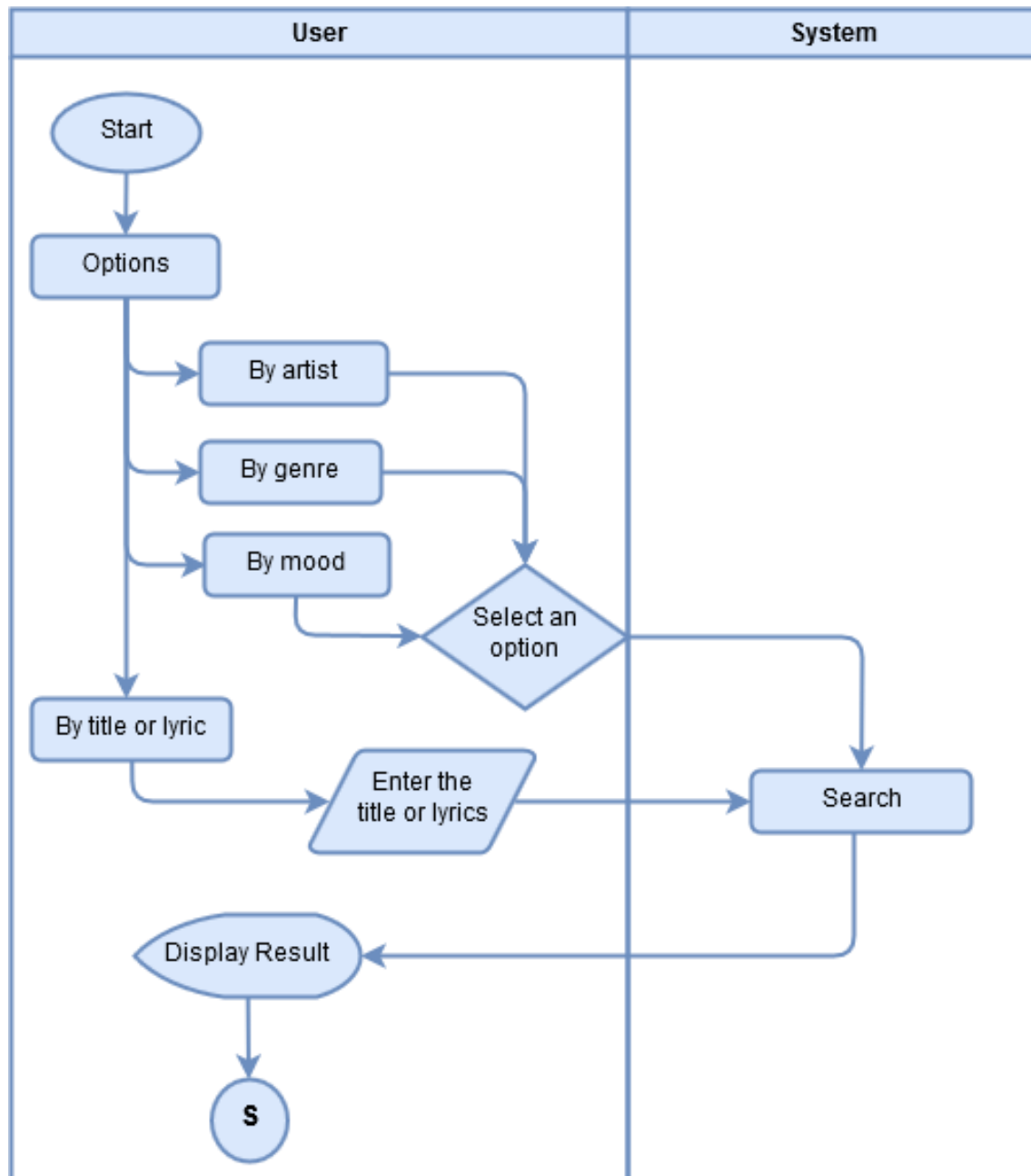


Figure 22: Level 1.4.3 swimlane diagram of searching songs

4.4.5 SWIMLANE DIAGRAM OF MUSIC PLAYER

In figure 23, the swimlane diagram of music player is shown

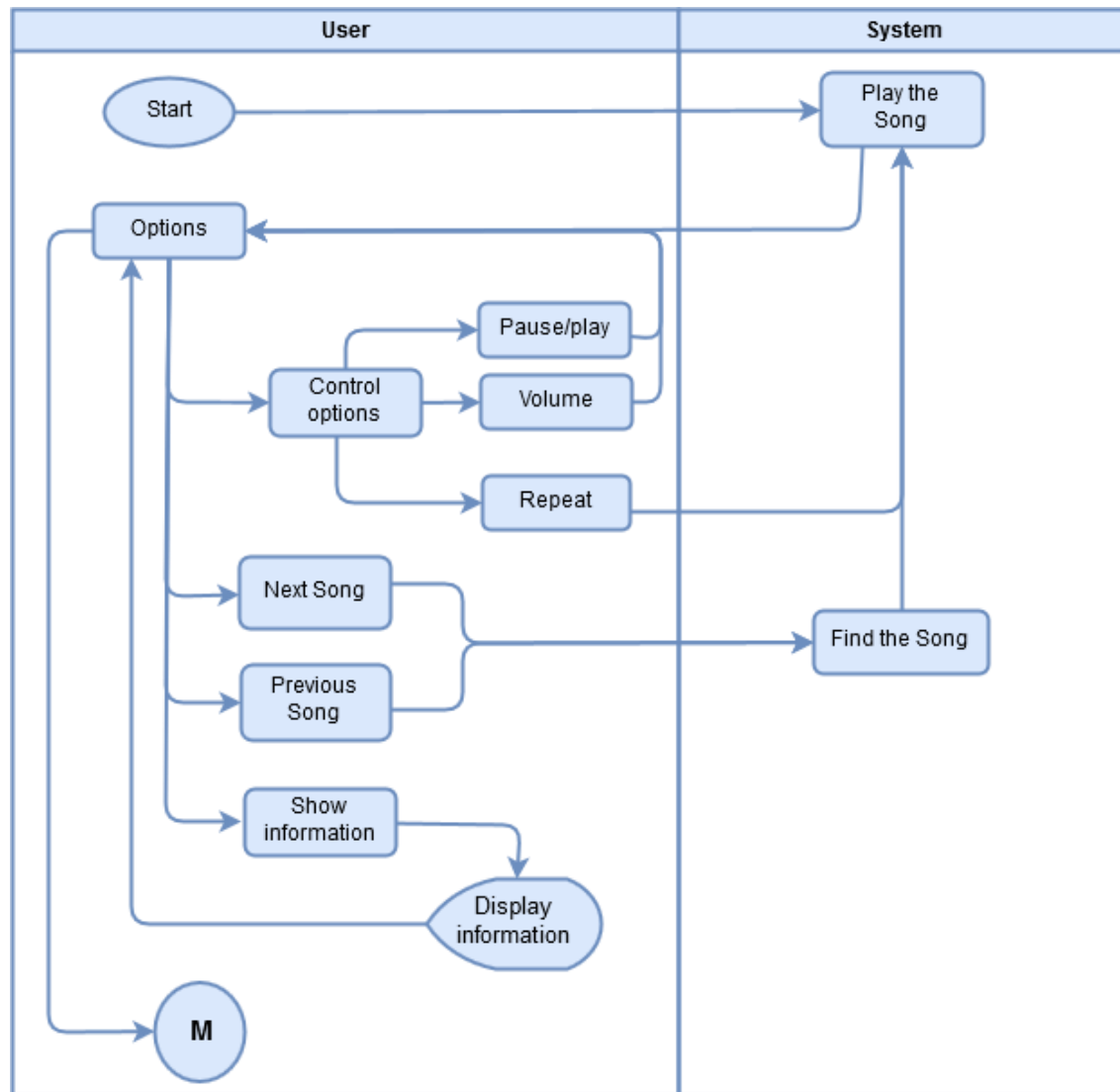


Figure 23: Level 1.3 swimlane diagram of music player

4.4.6 SWIMLANE DIAGRAM OF PERSONAL ACCOUNT

In figure 24, the swimlane diagram of personal account is shown

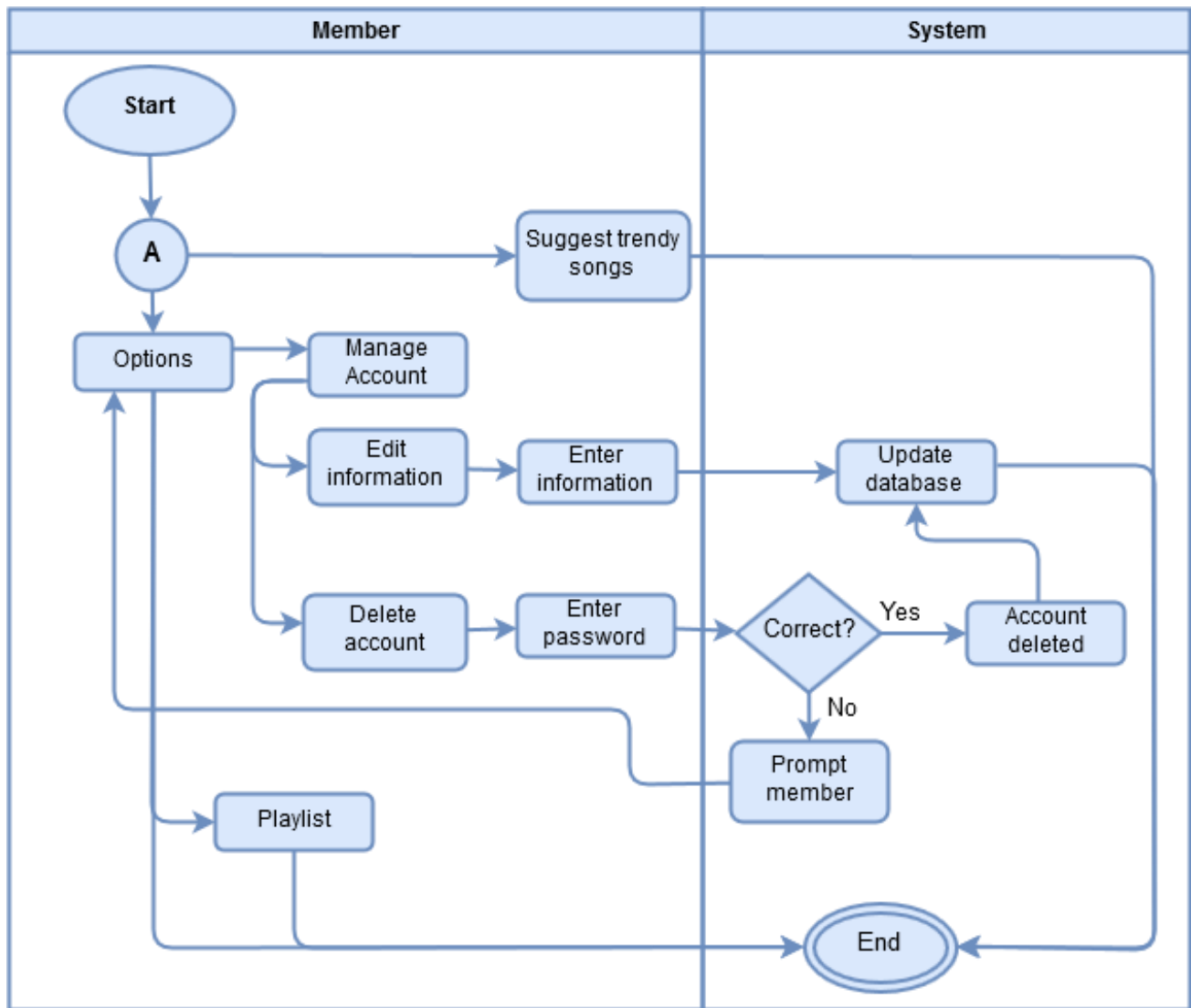


Figure 24: Level 1.4 swimlane diagram of personal account

4.4.7 SWIMLANE DIAGRAM OF PLAYLIST

In figure 25, the swimlane diagram of playlist is shown

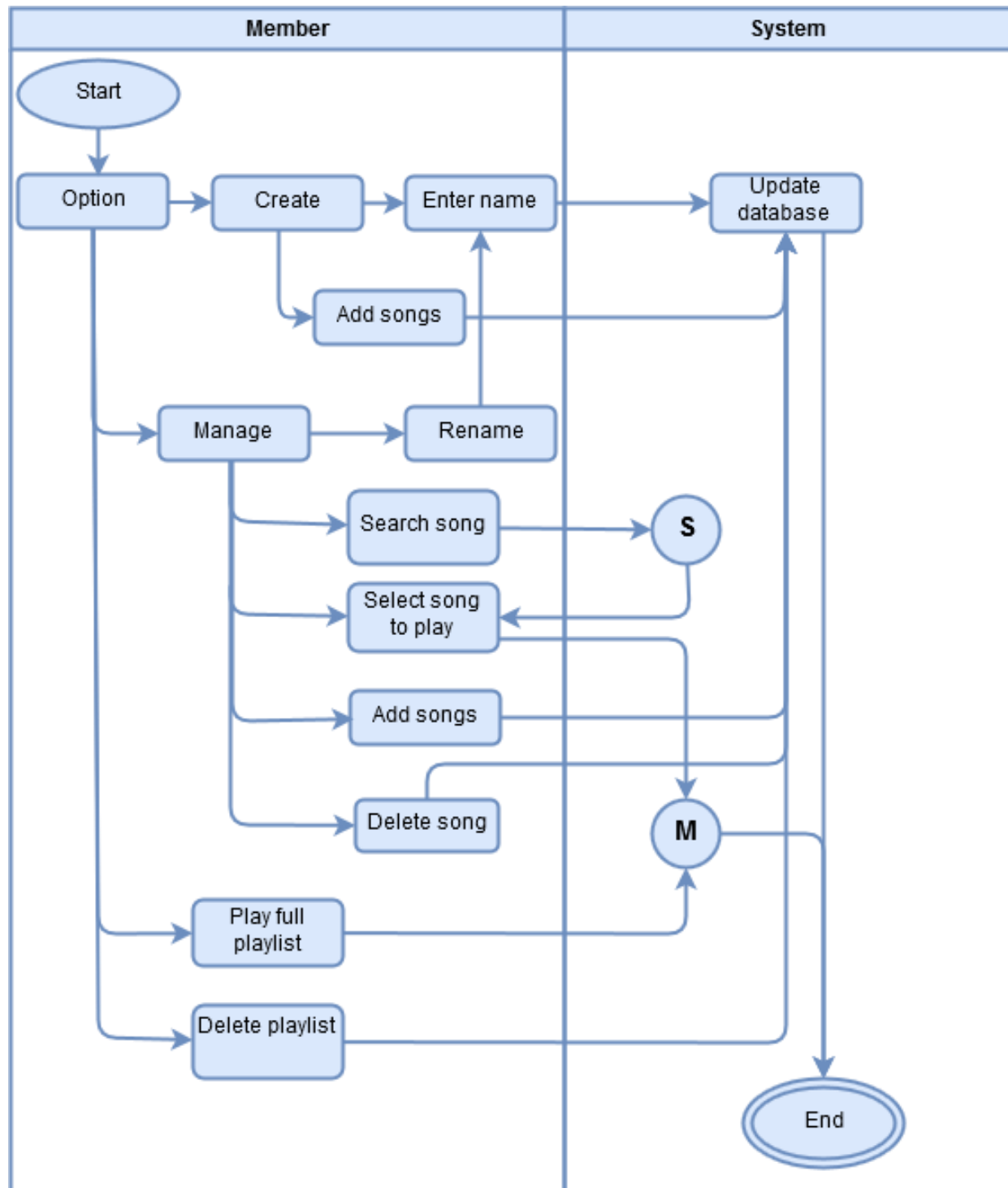


Figure 25: Level 1.4.4 swimlane diagram of playlist

4.4.8 SWIMLANE DIAGRAM OF ADMINISTRATION

In figure 26, the swimlane diagram of administration is shown

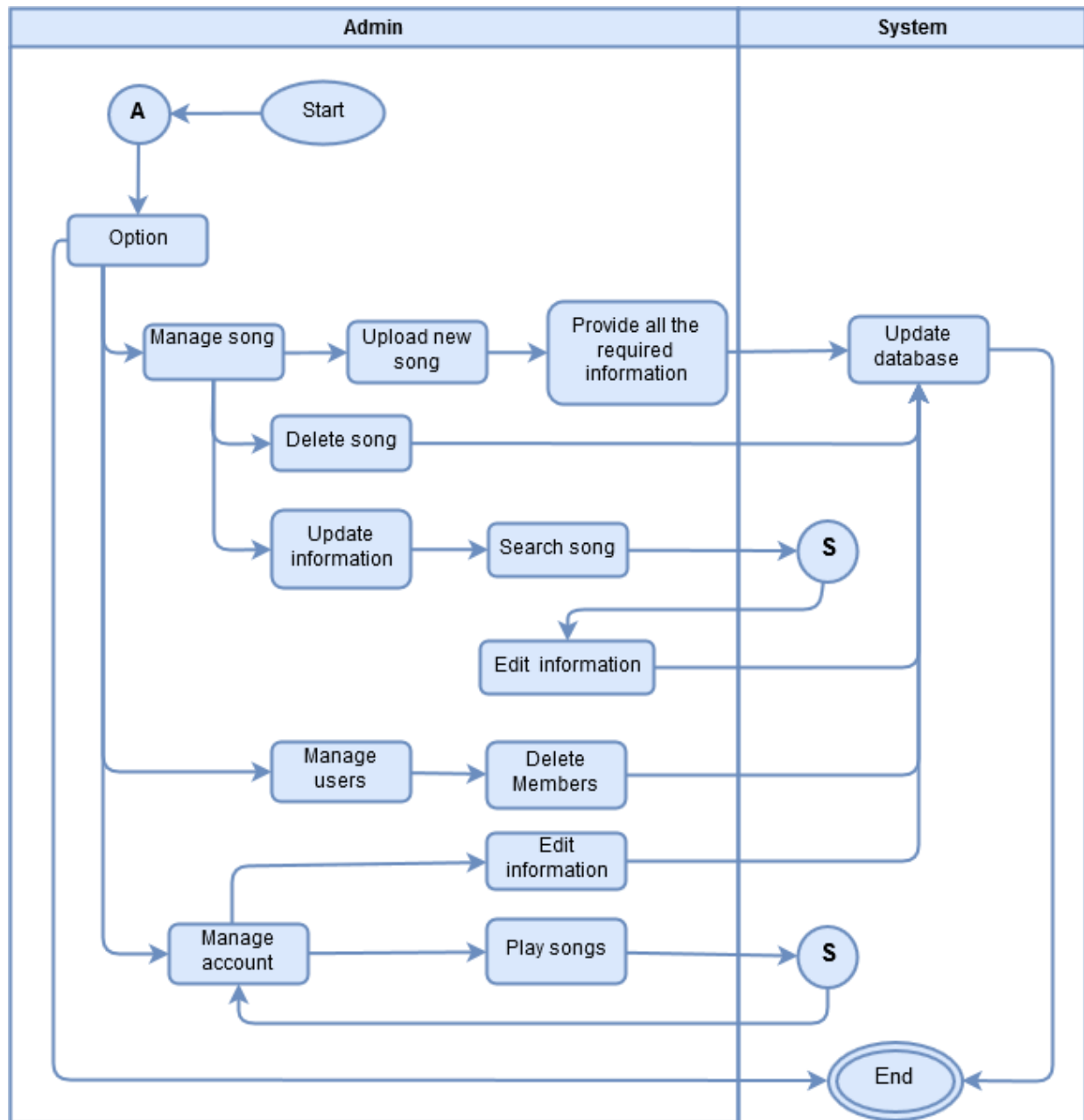


Figure 26: Level 1.5 swimlane diagram of administration

CHAPTER 5: DATA-BASED MODEL OF THE MUSIC STREAMING WEB APPLICATION

If software requirements include the need to create, extend or interface with a database or if complex data structures must be constructed and manipulated, the software team choose to create data model as part of overall requirements modeling. The entity relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects and the information that how the data objects are entered, stored, transformed and produced within the system.

5.1 Grammatical Parsing and Analysis

We identified all the nouns whether they are in problem space or in solution space from our usage scenario and categorized them according to their attributes. In the following table, “P” stands for problem domain and “S” stands for solution space. In table 1, the nouns are identified from the usage scenario of the project-

Table 1: Noun Identification

SL No.	Noun	P/S	Attributes
1.	Shworolipi	P	-
2.	Music	P	-
3.	Web application	P	-
4.	People	P	-
5.	User	S	21,41-44
6.	Feature	P	-
7.	Site	P	-
8.	Song	S	23-26,46,48
9.	Music player	P	-
10.	Registration	P	-

11.	Member	S	37, 38, 41-44
12.	Personal account	P	-
13.	Playlist	S	41
14.	Admin	S	37, 38, 41
15.	Interface	P	-
16.	Administrative activities	P	-
17.	Homepage	P	-
18.	Popular	P	-
19.	General homepage	P	-
20.	Personal homepage	P	-
21.	Type	S	11,22
22.	Non-member	S	42-45
23.	Title	S	-
24.	Artist	S	-
25.	Genre	S	-
26.	Lyric	S	-
27.	Suggestion	P	-
28.	information	P	-
29.	Screen	P	-
30.	Next song	P	-
31.	Previous song	P	-
32.	System	P	-
33.	Database	P	-
34.	option	P	-
35.	Procedure	P	-
36.	Email address	S	-
37.	Password	S	-
38.	Applicant	S	37, 38
39.	Authentication	P	-
40.	Name	S	-
41.	Date of birth	S	-

42.	Gender	S	-
43.	Phone number	S	-
44.	Favorite songs	P	-
45.	mood	S	-
46.	Guideline	P	-
47.	Popularity	S	-
48.	Convenience	P	-
49.	Activity	P	-

5.2 Potential Data Objects

After grammatical parsing, we identified the following data objects that may need in our system for information storage.

1. **User:** type, name, date of birth, gender, phone number
2. **Song:** title, artist, genre, lyric, popularity, mood
3. **Member:** email address, password, name, date of birth, gender, phone number
4. **Playlist:** name
5. **Admin:** email address, password, name
6. **Type:** member, non-member
7. **Non-member:** name, date of birth, gender, phone number
8. **Applicant:** email address, password

5.3 Analysis of Potential Data Objects

1. We do not need to store the data of **users**, **type**, **non-member** and **applicant**.
2. We need to add **member_ID**, **song_ID**, **playlist_ID**, **admin_ID** to **member**, **song**, **playlist** and **admin** respectively
3. We need two types of title for a song during searching. So we need to add **title_English** and **title_Bengali** instead of **title** to **Song**.
4. Since the **mood** attribute of song table can be more than one, it requires a separate table which will contain **song_ID** and **mood_name** variable. It will

depend on song table so its primary key will be both **song_ID** and **mood_name**. It will be a weak entity.

5.4 Final Data Objects

In table 2, the attributes of the final data objects are shown-

Table 2: Attributes identification of final data objects

SL No	Entity	Attributes
1.	Song	<u>song_ID</u> , title_English, title_Bengali, artist, genre, lyric, popularity
2.	Member	<u>member_ID</u> , email address, password, name, date_of_birth, gender, phone number
3.	Playlist	<u>playlist_ID</u> , name
4.	Admin	<u>admin_ID</u> , email address, password, name
5.	Mood	<u>song_ID</u> , <u>mood</u>

5.5 Data Object Relationship

In figure 27, the relationship among the data object member, playlist, admin and songs are shown-

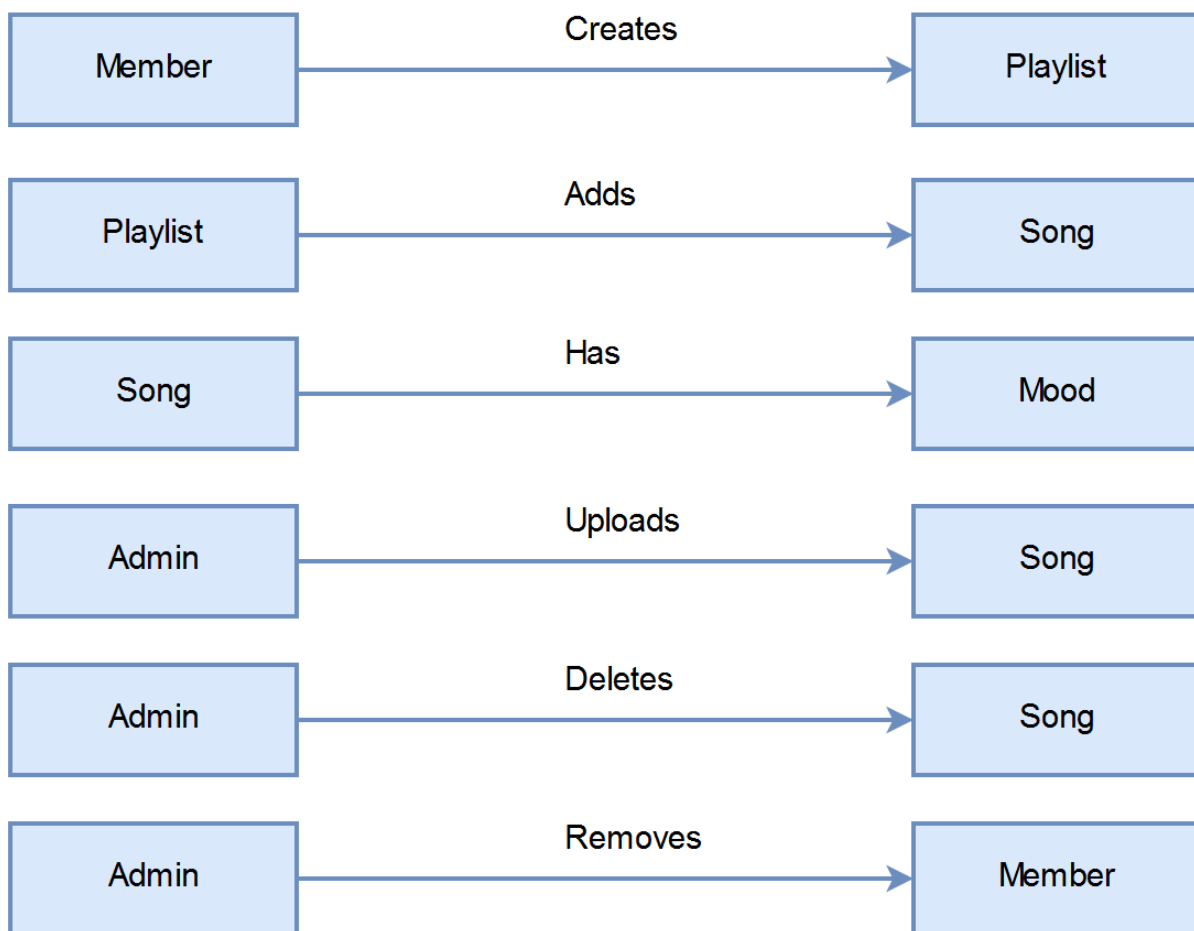


Figure 27: Data object relationship of Shworolipi

5.6 Entity Relation Diagram

An entity-relationship diagram (ER Diagram) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. We have shown the relationship among the entities through this model.

The ER diagram of the system of our application is shown in figure 28:

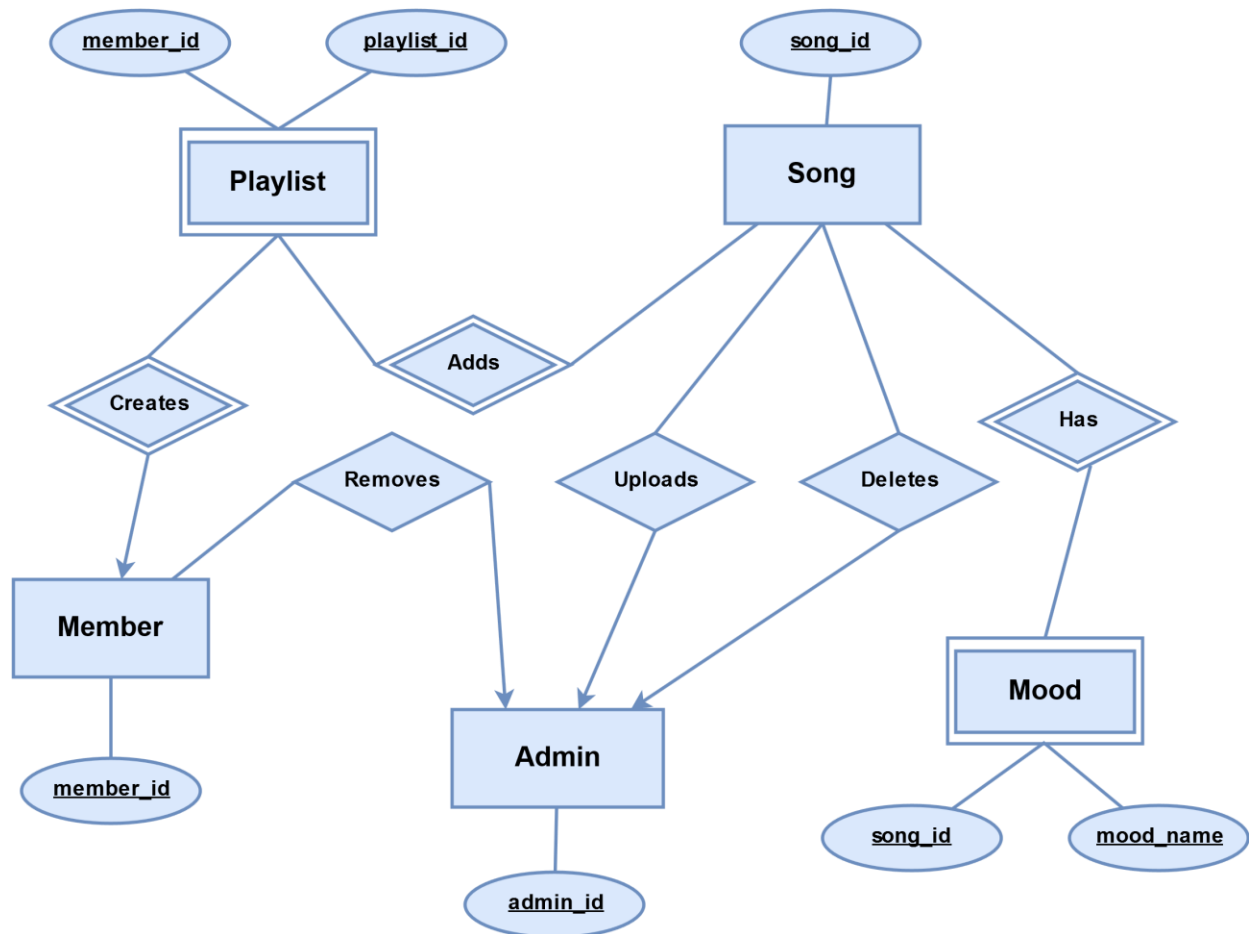


Figure 28: ER diagram of Shworolipi

5.7 Table Translation

From the entity relation diagram, the finalized form of the tables are shown in table 3:

Table 3: Table translation of ER diagram

No	Data Table
1.	Song (<u>song_ID</u> , title_English, title_Bengali, artist, genre, lyric, popularity, mood, admin_ID)
2.	Member (<u>member_ID</u> , email_address, password, name, date_of_birth, gender, phone_number)
3.	Playlist (<u>playlist_ID</u> , name, member_ID)
4.	Admin (<u>admin_ID</u> , email_address, password, name)
5.	Add (<u>song_ID</u> , <u>playlist_ID</u> , date)
6.	Delete(<u>admin_ID</u> , <u>song_ID</u> , title_Bengali, artist, genre, popularity, mood, date)
7.	Remove(<u>admin_ID</u> , <u>member_ID</u> , email_address, date, reason)

5.8 Schema Tables

In table 4, schema table of song is shown-

Table 4: Schema table of Song

1. Song		
Attributes	Types	Size
<u>song_ID</u>	INT	6
title_English	VARCHAR	50
title_Bengali	VARCHAR	60
artist	VARCHAR	30
genre	VARCHAR	15
lyric	VARCHAR	5000
popularity	INT	6
mood	VARCHAR	50
admin_ID	INT	6

In table 5, schema table of member is shown-

Table 5: Schema table of Member

2. Member		
Attributes	Types	Size
<u>member ID</u>	INT	6
mail address	VARCHAR	50
password	VARCHAR	50
name	VARCHAR	50
date of birth	DATE	-
gender	VARCHAR	10
phone number	VARCHAR	50
Hash	VARCHAR	32
Active	INT	1

In table 6, schema table of playlist is shown-

Table 6: Schema table of Playlist

3. Playlist		
Attributes	Types	Size
<u>playlist ID</u>	INT	6
name	VARCHAR	50
member_ID	INT	6

In table 7, schema table of admin is shown-

Table 7: Schema table of Admin

4. Admin		
Attributes	Types	Size
<u>admin_ID</u>	INT	6
email address	VARCHAR	50
password	VARCHAR	50
name	VARCHAR	50

In table 8, schema table of adds is shown-

Table 8: Schema table of adds

5. Adds		
Attributes	Types	Size
<u>song_ID</u>	NUMBER	6
<u>playlist_ID</u>	NUMBER	6

In table 9, schema table of deletes is shown-

Table 9: Schema table of deletes

6. Deletes		
Attributes	Types	Size
<u>song_ID</u>	INT	6
<u>admin_ID</u>	INT	6
title_Bengali	VARCHAR	50
artist	VARCHAR	50
genre	VARCHAR	15
popularity	INT	6
mood	VARCHAR	30

In table 10, schema table of remove is shown-

Table 10: Schema table of remove

7. Remove		
Attributes	Types	Size
<u>admin ID</u>	INT	6
<u>member ID</u>	INT	6
email_address	VARCHAR	20
reason	VARCHAR	500

CHAPTER 6: CLASS-BASED MODEL OF THE MUSIC STREAMING WEB APPLICATION

Class-based modeling represents the objects that the system will manipulate, the operations that will apply to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

6.1 Grammatical Parsing and Analysis

To identify our analysis class, firstly, we grammatically parsed all the nouns and then categorized them according to general classification and selection criteria. We identified potential class by identifying the nouns from the scenery. Then we compared those with the following criteria whether they matched or not. We noted down the number of the fulfilled criteria at the right column.

6.1.1 CLASS IDENTIFICATION WITH GENERAL CLASSIFICATION

In table 11, the nouns from the usage scenario are classified by general classification. Also, here, by "P" we meant a noun is in problem domain and by "S" we meant solution space.

General Classification:

1. External entities
2. Things
3. Occurrence or events
4. Roles
5. Organizational unit
6. Places
7. Structure

Table 11: Class Identification with General Classification

SL No	Noun	P/S	General Classification(GC)
1.	Shworolipi	P	-
2.	Music	P	-
3.	Web application	P	-
4.	People	P	-
5.	User	S	4
6.	Feature	P	-
7.	Site	P	-
8.	song	S	2
9.	Music player	S	4
10.	Registration	S	3
11.	Member	S	4
12.	Personal account	P	-
13.	Playlist	S	2
14.	Admin	S	4
15.	Interface	P	-
16.	Administrative activities	P	-
17.	Homepage	P	-
18.	Popular	P	-
19.	Non-member	S	4
20.	Title	S	2
21.	Artist	S	2
22.	Genre	S	2
23.	Lyric	S	2
24.	Suggestion	P	-
25.	information	P	-
26.	Screen	P	-
27.	Next song	P	-

28.	Previous song	P	-
29.	System	P	-
30.	Database	S	4
31.	Procedure	P	-
32.	Email address	S	2
33.	Password	S	2
34.	Applicant	S	4
35.	Authentication	S	3
36.	Name	S	2
37.	Date	S	2
38.	Gender	S	2
39.	Phone number	S	2
40.	Favorite songs	P	-
41.	mood	S	2
42.	Guideline	P	-
43.	Activity	P	-
44.	Server	P	-
45.	Popularity	S	2

6.1.2 CLASS IDENTIFIED WITH SELECTION CRITERIA

The nouns having two or more than two were selected from the general classification list. After that step, we compared them with the following criteria list. Those are-

1. Retained information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

In table 12, with the help selection criteria we identified whether the noun is accepted for preliminary class or not.

Table 12: Class Identification with Selection Criteria

SL No.	Noun	Special Classification (SC)	
		Accepted	Remarks
1.	User	1,2,3,4,5	Accepted
2.	Song	1,3,4	Accepted
3.	Music player	2,4,5	Accepted
4.	Registration	5	Rejected
5.	Member	1,2,3,4,5	Accepted
6.	Playlist	1,3,4,5	Accepted
7.	Admin	1,2,3,4,5	Accepted
8.	Non-member	3,4,5	Accepted
9.	Title	-	Rejected
10.	Artist	-	Rejected
11.	Genre	-	Rejected
12.	Database	2,5	Accepted
13.	Lyric	-	Rejected
14.	Email address	-	Rejected
15.	Password	-	Rejected
16.	Applicant	3,4,5	Accepted
17.	Authentication	5	Rejected
18.	Name	-	Rejected
19.	Date of birth	-	Rejected
20.	Gender	-	Rejected
21.	Phone number	-	Rejected
22.	Mood	-	Rejected
23.	Popularity	-	Rejected

6.2 Preliminary Classes

From above table, we have taken all the noun who passed three or more accepted criteria. So these are the candidate classes who are selected primarily:

- User
- Song
- Music player
- Member
- Playlist
- Admin
- Non-member
- Database
- Applicant

6.3 Verb Identification

We have identified the verbs from our scenery to find out the necessary methods for the classes. In the table 13, "P" stands for problem domain and "S" stands for solution space. The possibility of verbs in the solution space have higher possibilities to become methods of the class.

Table 13: Verb Identification

SL No.	Verbs	Remarks
1.	Connect	S
2.	Search	S
3.	Provide	P
4.	Register	S
5.	Manage	S
6.	Create	S
7.	Follow	S

8.	Find	S
9.	Get	P
10.	Sign up	S
11.	Sign in	S
12.	Suggest	S
13.	Get trendy song list	P
14.	Play	S
15.	Store	S
16.	Keep	P
17.	Show	P
18.	Pause	S
19.	Repeat	S
20.	Control volume	S
21.	Select next or previous	S
22.	Prompt	S
23.	Check	S
24.	Add song	S
25.	Enlist	P
26.	Require	P
27.	Exist	P
28.	Valid	S
29.	Receive	S
30.	Authenticate	S
31.	Enter	S
32.	Match	S
33.	Try	P
34.	Contain	P
35.	Make playlist	P
36.	delete	S
37.	Update	S

6.4 Attributes and Methods of Preliminary Classes

Analyzing the above table, we have categorized the verbs and convert them into method names. We put them to their respective classes and showed them in the table 14:

Table 14: Attributes and Methods of Preliminary Classes

SL No	Preliminary Class	Nouns	Verbs
1.	User	-	Search, play song, share songs, get guideline
2.	Song	song_ID, title_English, title_Bengali, artist, genre, lyric, popularity, mood, admin_ID	-
3.	Music player	Song	Show information, select next, select previous
4.	Member	member_ID, email_address, password, name, date_of_birth, gender, phone_number, playlist	Search, get trendy song list, play song, sign in, sign out, edit account, create playlist, delete account, get guideline
5.	Playlist	Name, song, playlist_ID, member_ID	add song, delete song, rename, delete playlist
6.	Admin	admin_ID, email_address, password, name	Search, get trendy song list, play song, Manage account, remove user, upload song, delete song, get guideline, update songs

7.	Non-member	-	Search, play song, sign up, get confirmation, share songs, get guideline
8.	Database	-	Update, retrieve, store
9.	Applicant	Email address, password	Sign up, get confirmation

6.5 Analysis of Potential Classes

1. Here **user class** has some similarities with **member, non-member and admin class** in some functionalities. So, those can extend user class.
2. We can remove **applicant class** because its functionalities are already fulfilled by **non-member class**.
3. We can remove database class since those methods can be fulfilled by other class.

6.6 Final Classes:

From above analysis, our final classes are:

1. User
 - a. Member
 - b. Non-member
 - c. admin
2. Song
3. Music player
4. Playlist

6.7 Attributes and Methods of Final Classes:

Attributes and methods of the final classes are shown in the following tables. In table 15, attributes and methods of user class are shown-

Table 15: Attributes and Methods of User Class

1. User	
Attributes	Methods
-	search(), playSongs(), getGuideline()

In table 16, attributes and methods of user member are shown-

Table 16: Attributes and Methods of Member Class

1.a. Member	
Attributes	Methods
member_ID, emailAddress, password, name, dateOfBirth, gender, phoneNumber, playlist	signIn(), getTrendySongs(), createPlaylist(), editPersonalAccount(), deletePersonalAccount(), getGuideline(), signOut()

In table 17, attributes and methods of non-member class are shown-

Table 17: Attributes and Methods of non-member Class

1.b. Non-member	
Attributes	Methods
	signUp(), getConfirmation(), getGuideline()

In table 18, attributes and methods of admin class are shown-

Table 18: Attributes and Methods of Admin Class

1.c. Admin	
Attributes	Methods
Name, email_address, password	signIn(), signOut(), manageAccount(), removeUser(), updateSong(), uploadSongs(), deleteSongs(), getGuideline()

In table 19, attributes and methods of song class are shown-

Table 19: Attributes and Methods of Song Class

2. Song	
Attributes	Methods
title_English, title_Bengali, artist,	viewAllSongId(), viewAllTitleBengali(), viewAllArtist(),

genre, lyric, popularity, mood	
---	--

In table 20, attributes and methods of musicPlayer class are shown-

Table 20: Attributes and Methods of MusicPlayer Class

3. MusicPlayer	
Attributes	Methods
song	getSong(), showInformation(), selectNextSong(), selectPreviousSong()

In table 21, attributes and methods of playlist class are shown-

Table 21: Attributes and Methods of Playlist Class

4. Playlist	
Attributes	Methods
name, Song, playlist_ID, member_ID	view(), search(), playSongs(), addSong(), rename(), deleteSong(), deletePlaylist()

6.8 Class Responsibilities and Collaborators Card of Final Classes:

In the class card, we have shown the responsibilities of a class and the collaborative classes to perform the responsibilities. One responsibility may need one or more methods to carry out. These class cards will be needed to make CRC model in the next step. We will see the association among the classes from their collaboration and the name of the association will be named after the responsibilities.

The class cards of user class is given in table 22-

Table 22: Class Card of User Class

1. User	
Responsibilities	Collaborator
Searching songs	Song
Playing songs	Song, MusicPlayer
Getting guidelines	-

The class cards of member class is given in table 23-

Table 23: Class Card of Member Class

1.a. Member	
Responsibilities	Collaborator
Authenticating	-
Handling playlist	Playlist
Handling personal account	-

The class cards of non-member class is given in table 24-

Table 24: Class Card of non-member Class

1.b. Non-member	
Responsibilities	Collaborator
Getting registered	-

The class cards of admin class is given in table 25-

Table 25: Class Card of Admin Class

1.c. Admin	
Responsibilities	Collaborator
Managing account	-
Removing user	Member
Updating song information	Song

The class cards of Song class is given in table 26-

Table 26: Class Card of Song Class

2. Song	
Responsibilities	Collaborator
View Song list	-

The class cards of musicPlayer class is given in table 27-

Table 277: Class Card of MusicPlayer Class

3. MusicPlayer	
Responsibilities	Collaborator
Getting a song	Song

The class cards of playlist class is given in table 28-




Table 28: Class Card of Playlist Class

4. Playlist	
Responsibilities	Collaborator
Updating playlist	Song, Member
Play songs	Song

6.1 Class Diagram

Class diagram is a diagram where dynamics of object interaction and collaboration are represented through UML diagrams and their networks. Here composition, association and inheritance of the classes are shown in the diagram. The notations are in table 29-

Table 29: Notations of class diagram

Relationship	Notation
Association	
Composition	
Inheritance	

CLASS DIAGRAM

The class diagram shows the relation among the classes. It is shown in figure 29-

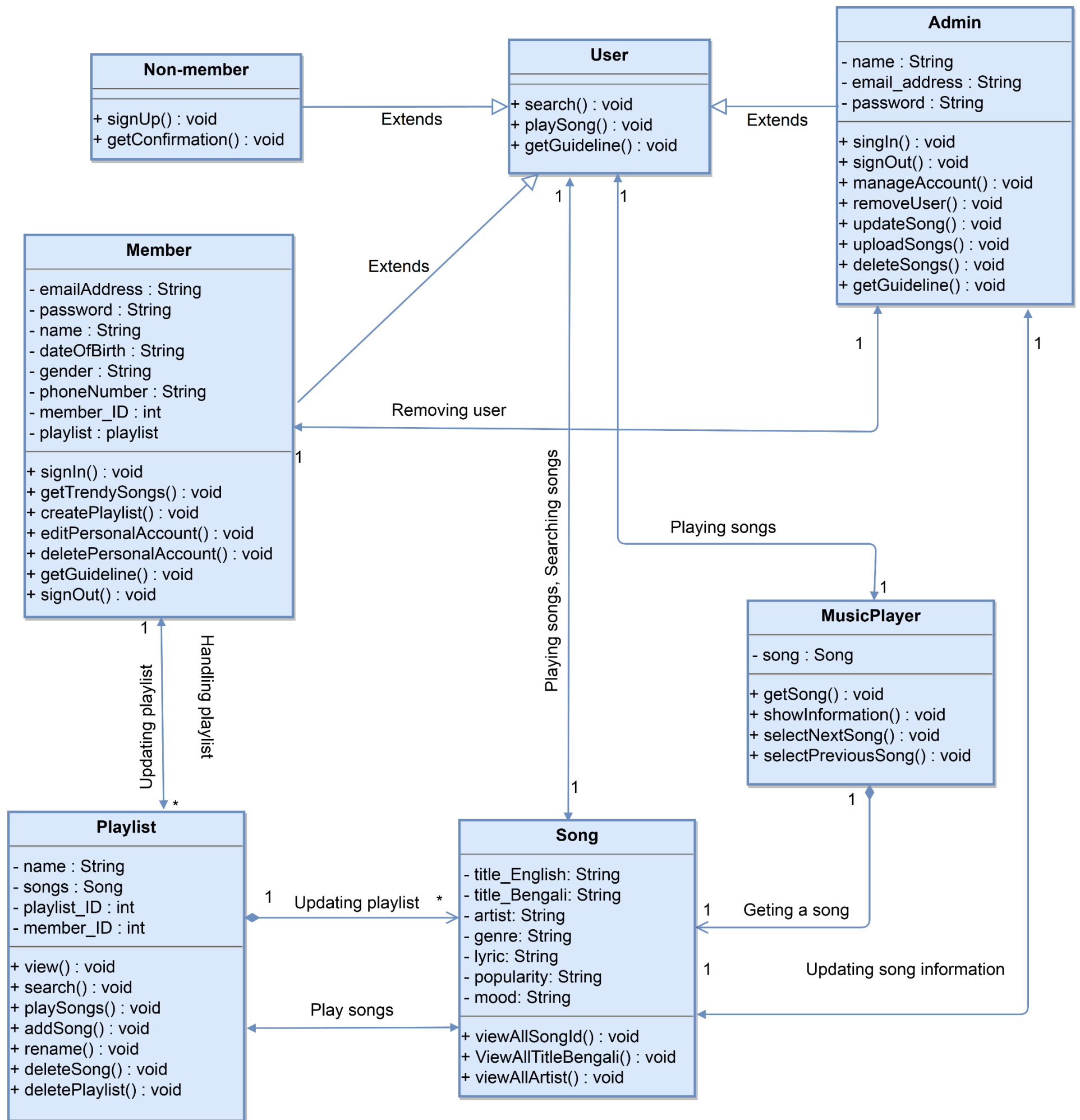


Figure 29: class diagram of MWSA

CHAPTER 7: BEHAVIORAL MODEL OF THE MUSIC STREAMING WEB APPLICATION

The behavioral model indicates how software will respond to external events. Two different behavioral representations are discussed in this chapter. The first indicates how individual class changes state based on external events and the second shows the behavior of the software as a function of time.

7.1 Event Identification

We have identified all the events, their initiators and collaborators of the total system. Those are shown in table 30:

Table 30: Event identification

SL No.	Event	Initiator	Collaborators
1.	Enter to Shworolipi	User	-
2.	Search song	User	-Song
3.	Select song to play	User	-Song
4.	Get music player	User	MusicPlayer
5.	Get guideline	User	-
6.	Close website	User	-
7.	Enter email address and password	Non-member	-
8.	Validate the request	Non-member	-
9.	Get confirmation	Non-member	-
10.	Enter to personal account	Member	-
11.	Handle playlist	Member	Playlist
12.	Edit account	Member	-
13.	Delete account	Member	-
14.	Signed out from account	Member	-
15.	Play song	MusicPlayer	Song
16.	Select next or previous song	MusicPlayer	Song

17.	Close music player	MusicPlayer	-
18.	Checking playlist	Playlist	Member
19.	Update playlist	Playlist	Song, Member
20.	Connect music player	Playlist	MusicPlayer
21.	Close playlist	Playlist	-
22.	Enter to admin account	Admin	-
23.	Manage account	Admin	-
24.	Remove user	Admin	Member
25.	Manage Song	Admin	Song
26.	Sign out from account	Admin	-

7.2 State Diagram

A state diagram is a diagram used in computer science to describe the behavior of a system considering all the possible states of an object when an event occurs. This behavior is represented and analyzed in a series of events that occur in one or more possible states. We have shown the state diagram of the classes in the following diagrams:

7.2.1 USER

In figure 30, the state diagram of user class is shown-

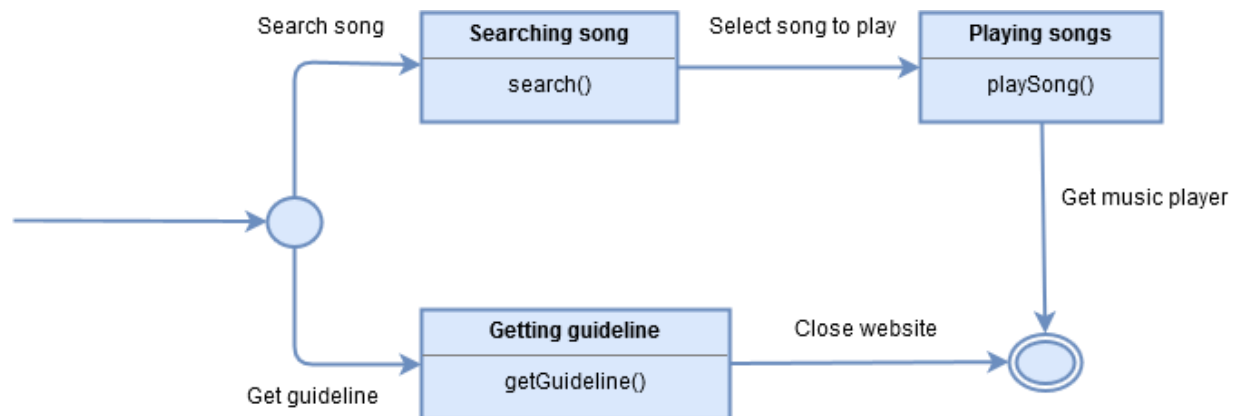


Figure 30: State diagram of user class

7.2.2 MEMBER

In figure 31, the state diagram of member class is shown-

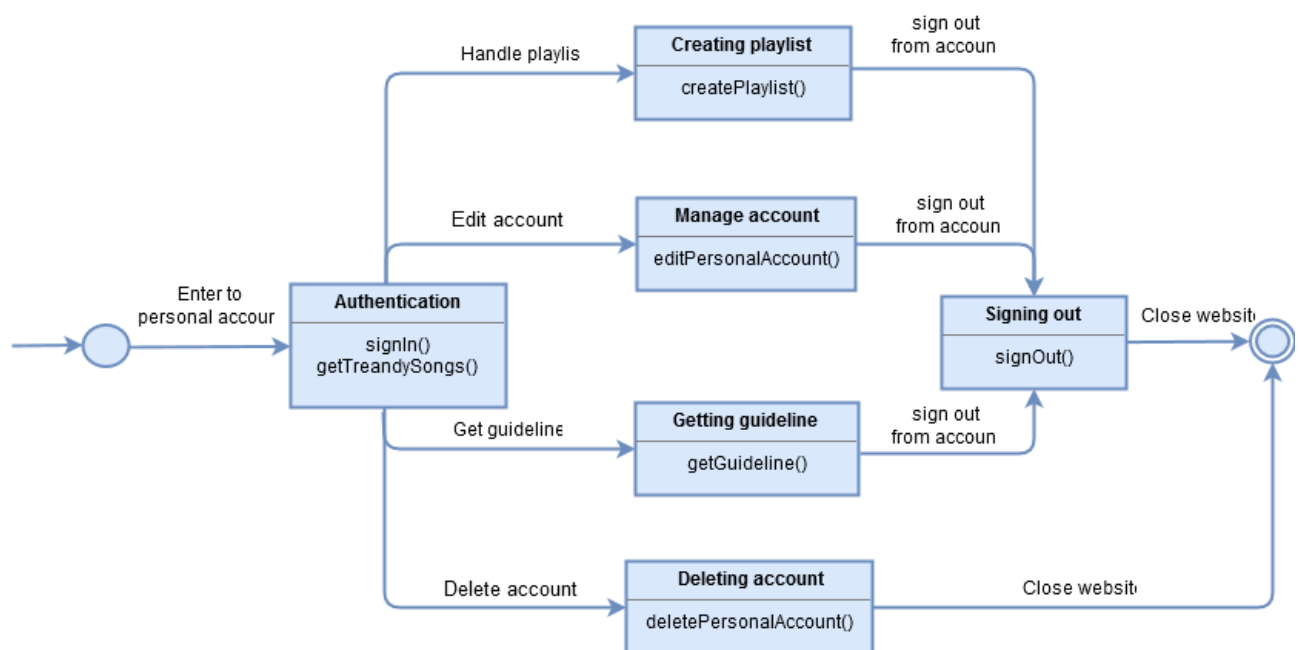


Figure 31: State diagram of member class

7.2.3 MUSICPLAYER

In figure 32, the state diagram of MusicPlayer class is shown-

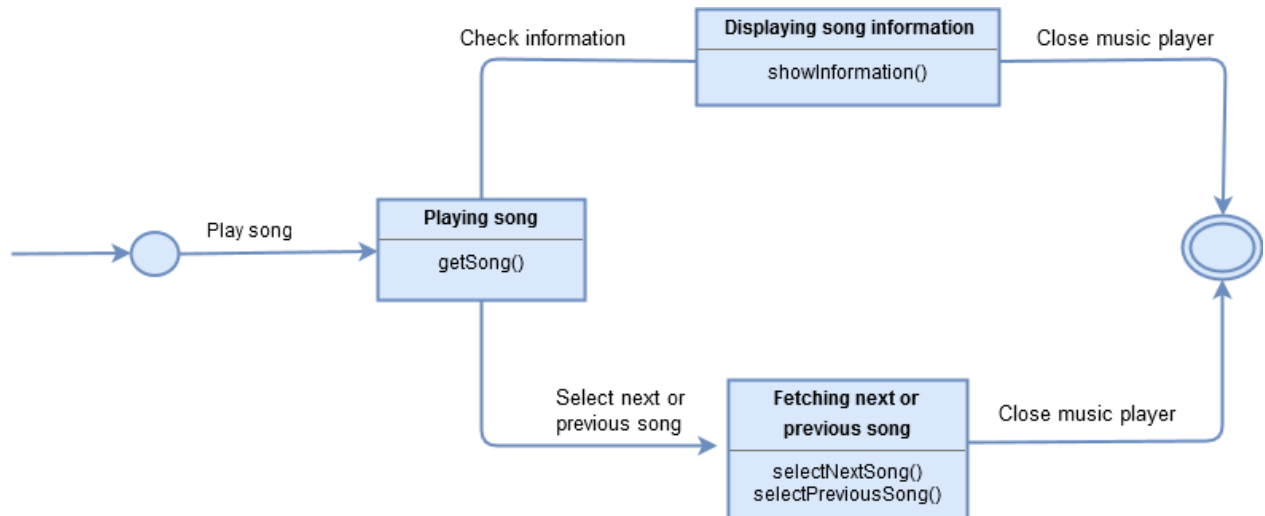


Figure 32: State diagram of musicPlayer class

7.2.4 NON-MEMBER

In figure 33, the state diagram of Non-member class is shown-

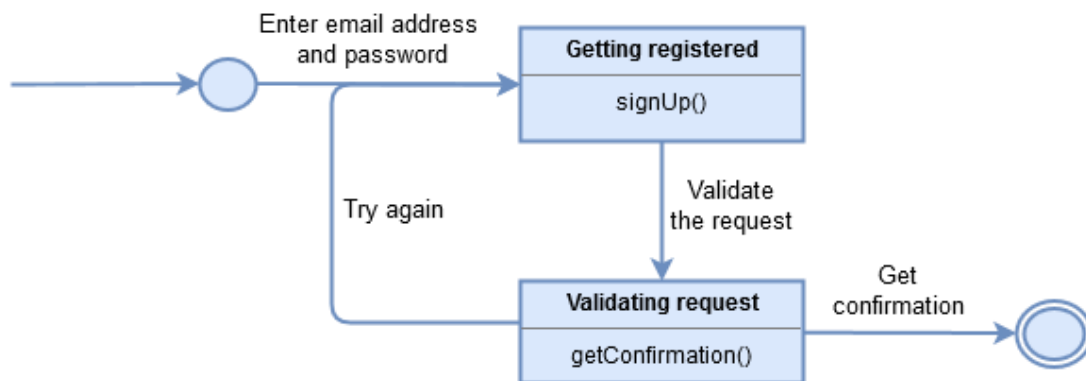


Figure 33: State diagram of non-member class

7.2.1 SONG

In figure 34, the state diagram of Song class is shown-



Figure 34: State diagram of song class

7.2.2 PLAYLIST

In figure 35, the state diagram of Playlist class is shown-

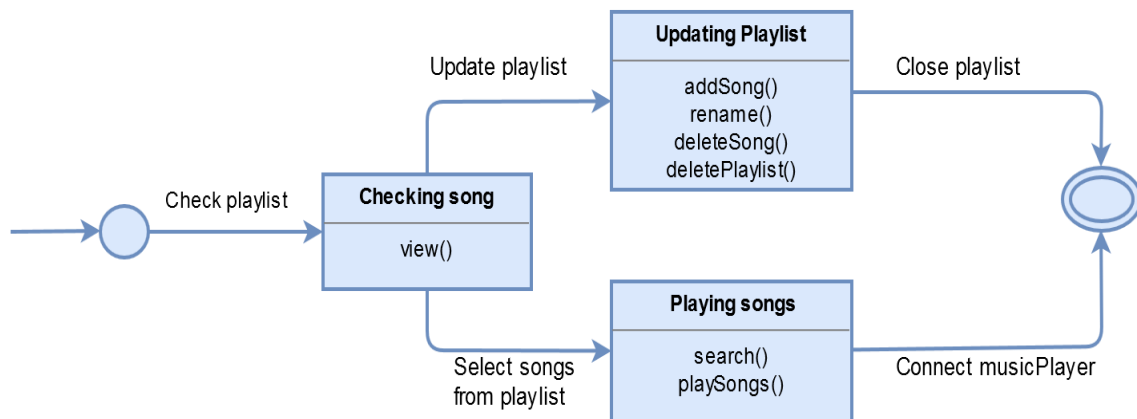


Figure 35: State diagram of playlist class

7.2.3 ADMIN

In figure 31, the state diagram of Admin class is shown-

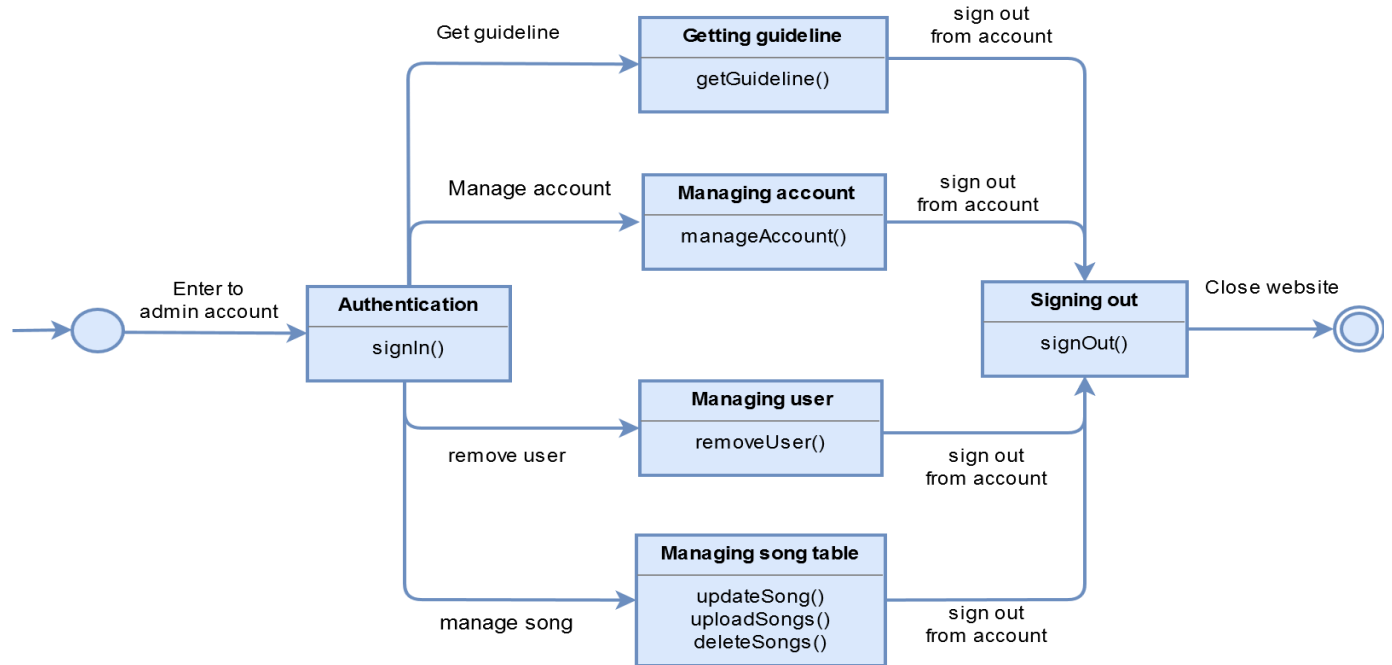


Figure 36: State diagram of admin class

7.3 Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. We have shown the sequence diagram of three modules- Homepage, personal account and administration.

7.3.1 HOMEPAGE

In figure 37, the sequence diagram of Homepage is shown-

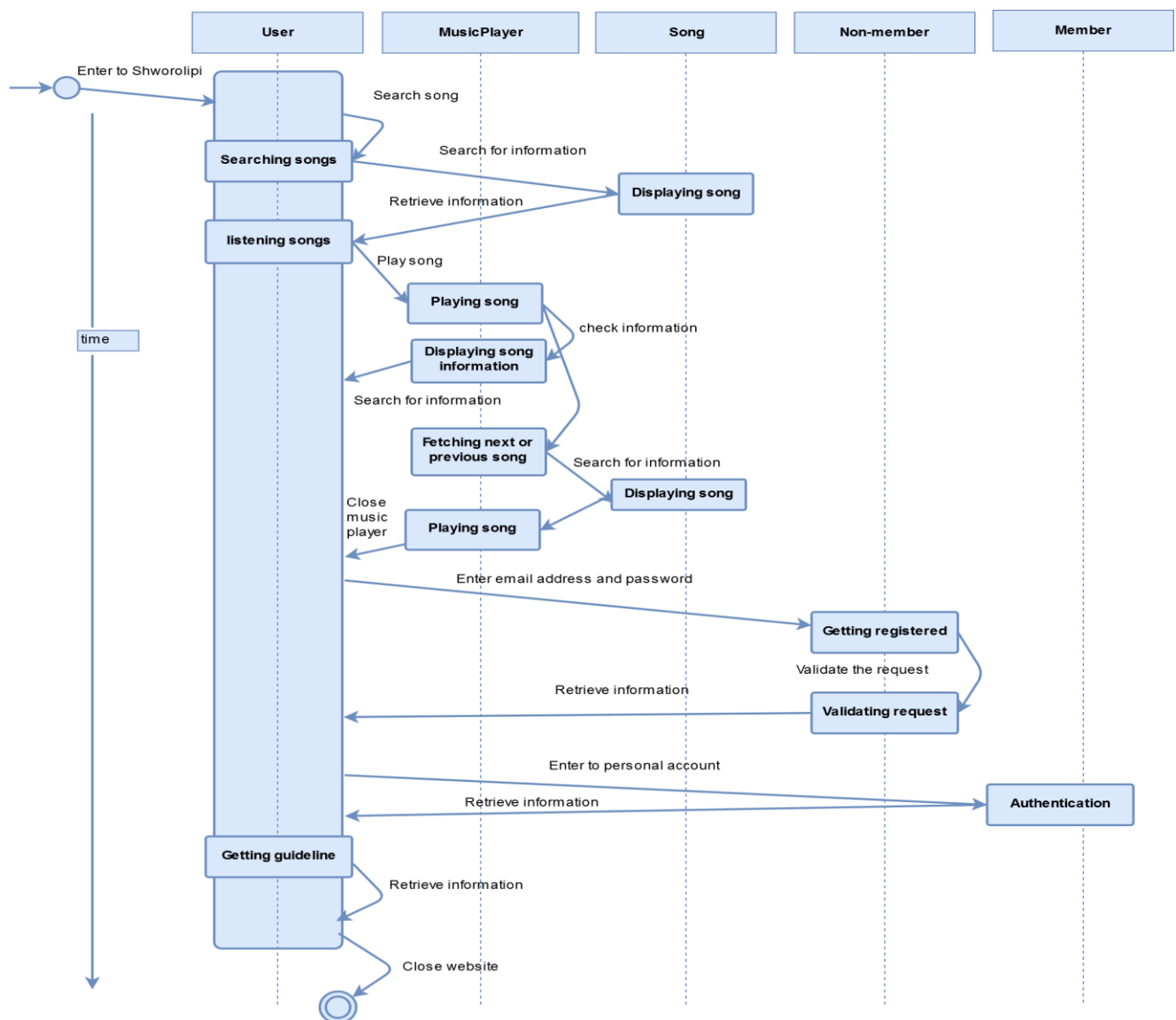


Figure 37: Sequence diagram of homepage

7.3.2 PERSONAL ACCOUNT

In figure 38, the sequence diagram of personal account is shown-

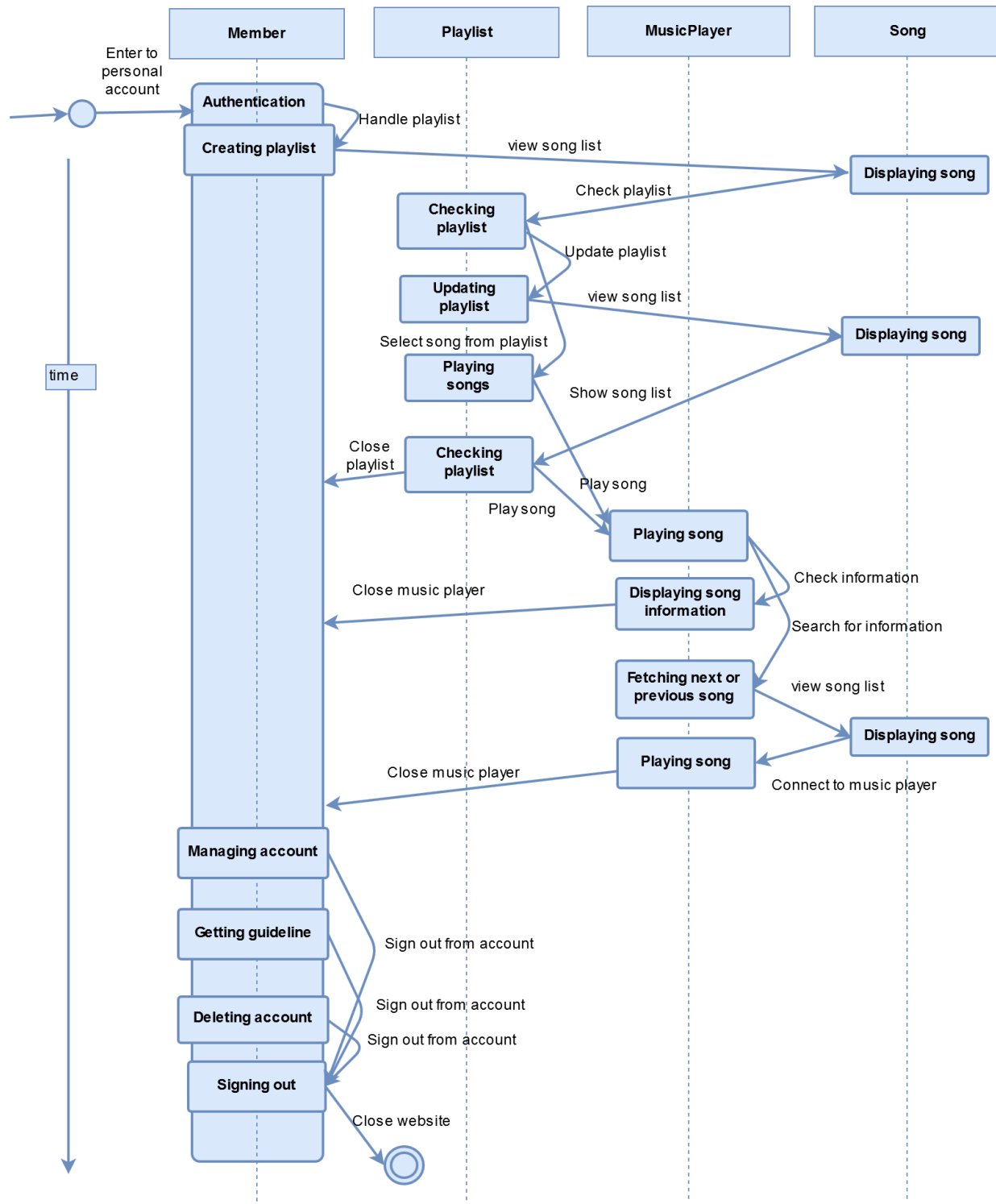


Figure 38: Sequence diagram of personal account

7.3.3 ADMINISTRATION

In figure 38, the sequence diagram of administration is shown-

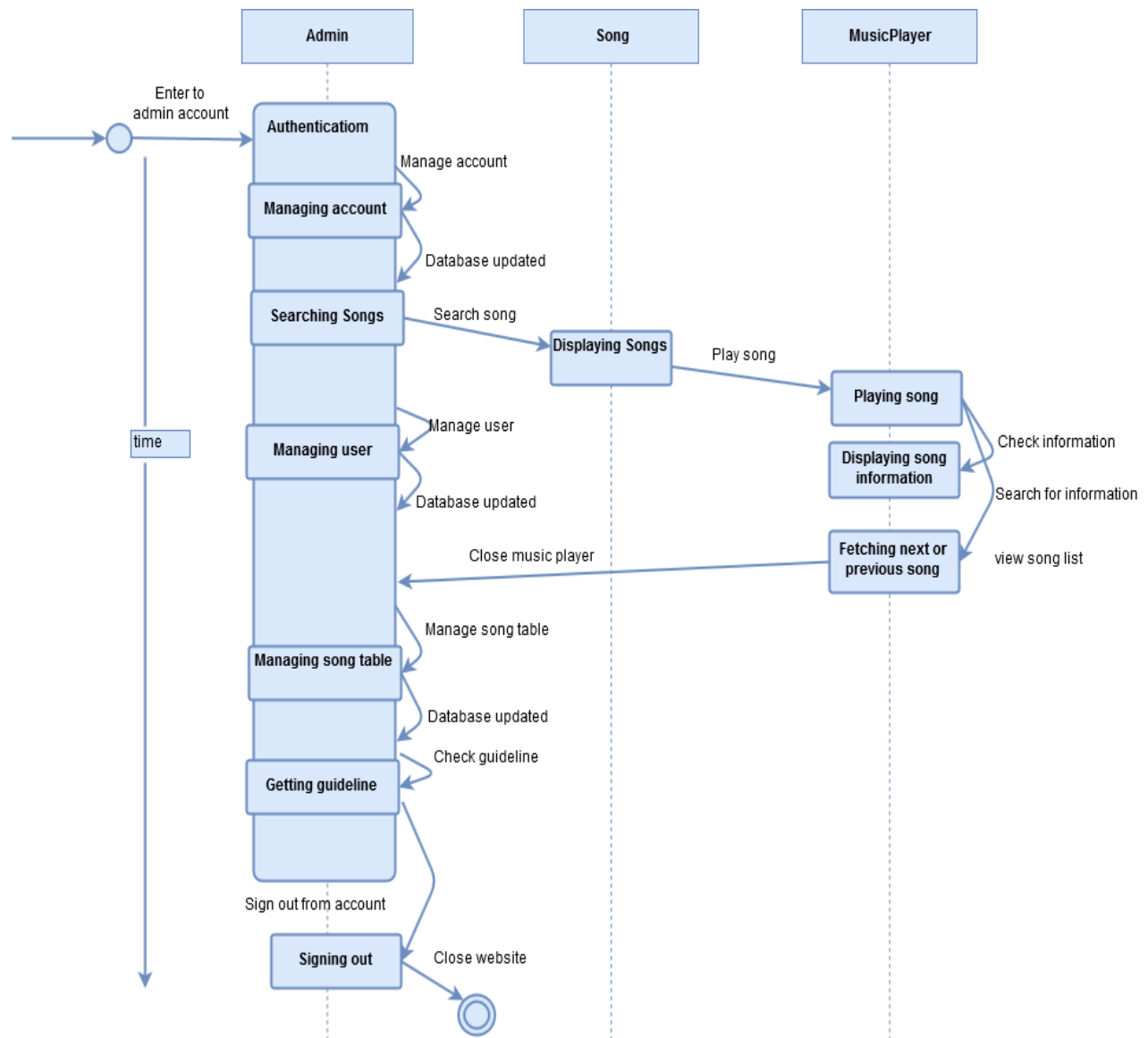


Figure 39: Sequence diagram of administration

7.3.4 FULL-SYSTEM (MSWA)

In figure 40, the sequence diagram of full system is shown-

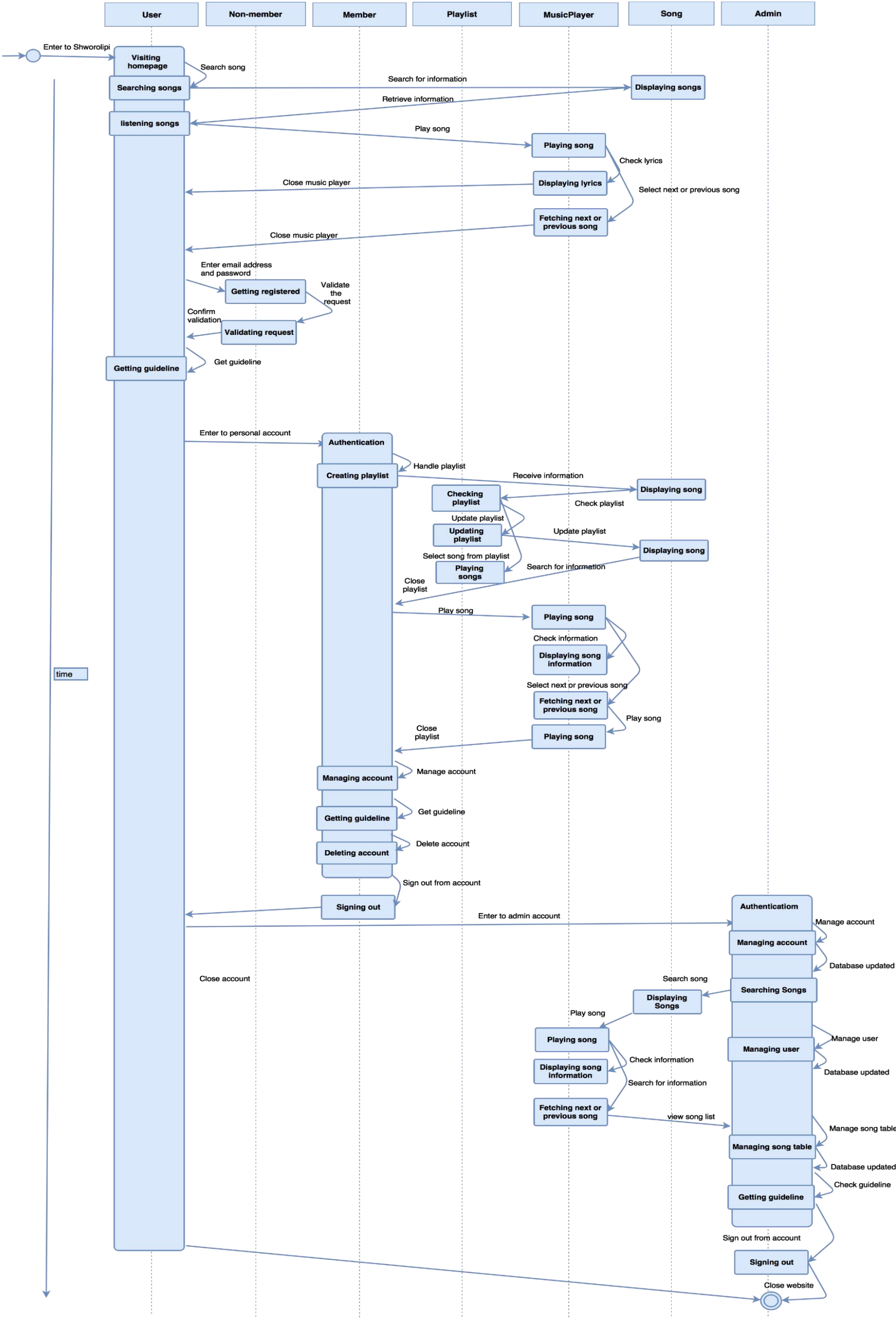


Figure 40: Sequence diagram of MSWA

CHAPTER 8: CONCLUSION

We are pleased to submit the final SRS report on a music streaming web application. From this, the readers will get a clear and easy view of the overall system. This SRS document can be used effectively to maintain the software development cycle. It will be very easy to conduct the whole project using this SRS.

Hopefully, this document can also help our junior BSSE students. We tried our best to remove all dependencies and make an effective and fully designed SRS. We believe that the reader will find it in order.

REFERENCES

1. Book- Pressman, Roger S. Software Engineering: A Practitioner's Approach (7th Edition)
2. ER diagram- <https://www.tutorialcup.com/dbms/er-data-model.htm> [Last accessed: 12/02/2017 4.30 PM]
3. CRC diagram-<http://csis.pace.edu/~marchese/cs615sp/L4New/l4senewf.htm> [Last accessed: 15/02/2017 3.50 PM]