

ADVANCED DATABASE PROJECT

Vinyl Record Collection Manager

Work package 1

GROUP C

Muhammad Ahsan (EPUN)

Pham Hoai Vu (EPUN)

Rosa Tsegaye (ULY2)

Farhad Nooralahzadeh (ULY2)

Punyavee Napalai (ULY2)

November 13, 2011

Document Change History

Version	Date	Responsible	Reason for Change
0.1	10/11/2011	LYON Team	Initial version
0.2	12/11/2011	NANTES Team	Added description for the ERD. Formatting changes.
0.3	13/11/2011	LYON Team	Added Gantt chart. Final formatting.

Table of Contents

Document Change History	ii
Table of Contents	iii
Chapter 1 Project Management Activities	1
1.1 Software Development Model.....	1
1.2 Communication and Team Meetings.....	1
1.3 Software Configuration Management	1
1.4 Work Partitioning and Assignment	1
1.5 Online Collaborative Environment	1
1.6 Issue Solving.....	1
1.7 Progress of each assigned Task	1
1.8 Gantt Chart	2
Chapter 2 Use Case Diagram	3
2.1 User case diagrams	3
Chapter 3 Use Case Description	4
3.1 Owned List.....	4
3.2 Play List.....	6
3.3 Tracked List/Flag.....	7
3.4 Evaluate and Post Comment	7
3.5 Create & Manage Records	8
3.6 Create & Manage Tracks	10
3.7 View Log	11
3.8 Log in.....	12
3.9 Become member	14
3.10 Search/Listen.....	14
Chapter 4 Entity-Relationship Diagram	16
4.1 The Entity-Relationship Diagram	16
4.2 Entity-Relationship Diagram Description	17
Chapter 5 Wireframe	18
Conclusion 	29

Chapter 1 Project Management Activities

1.1 Software Development Model

Concurrent Development is used as software development process. During this process model the entire team member went through planning and analysis phase. After that Lyon and Nantes were divided into sub team. The design and implementation phase is divided into two major subprojects for deliverable 1 including use cases/use case diagram and ER diagram. The Lyon and Nantes team worked on two subprojects respectively. These major subprojects are integrated at the end after their completion.

1.2 Communication and Team Meetings

Regular team meetings with a predefined agenda were conducted every week between Lyon and Nantes team in order to discuss the progress of the work being done. A critical analysis of the work and any possible problems are openly discussed. Any team conflicts are also resolved during the meeting. Any confusion and questions on the provided requirements documents were discussed on the course forum and also emailed directly to professor and tutor in order to clarify the requirement and expectation from of the project.

1.3 Software Configuration Management

“Github” is being used as collaborative version control system in order to share all the documents in order to have consistency and handling of different version by team members distributed across two sites.

1.4 Work Partitioning and Assignment

Deliverable 1 was divided into well defined chunks in order and assign to one of the two teams to work on it while the other team providing support to the first team. Tasks are assigned during meeting session with consensus keeping in view the domain expertise of the team member.

1.5 Online Collaborative Environment

“Zoho” is being used as project management online portal site in order to assign tasks, achieving milestones, monitor project progress, team meetings and forum for discussion within the team.

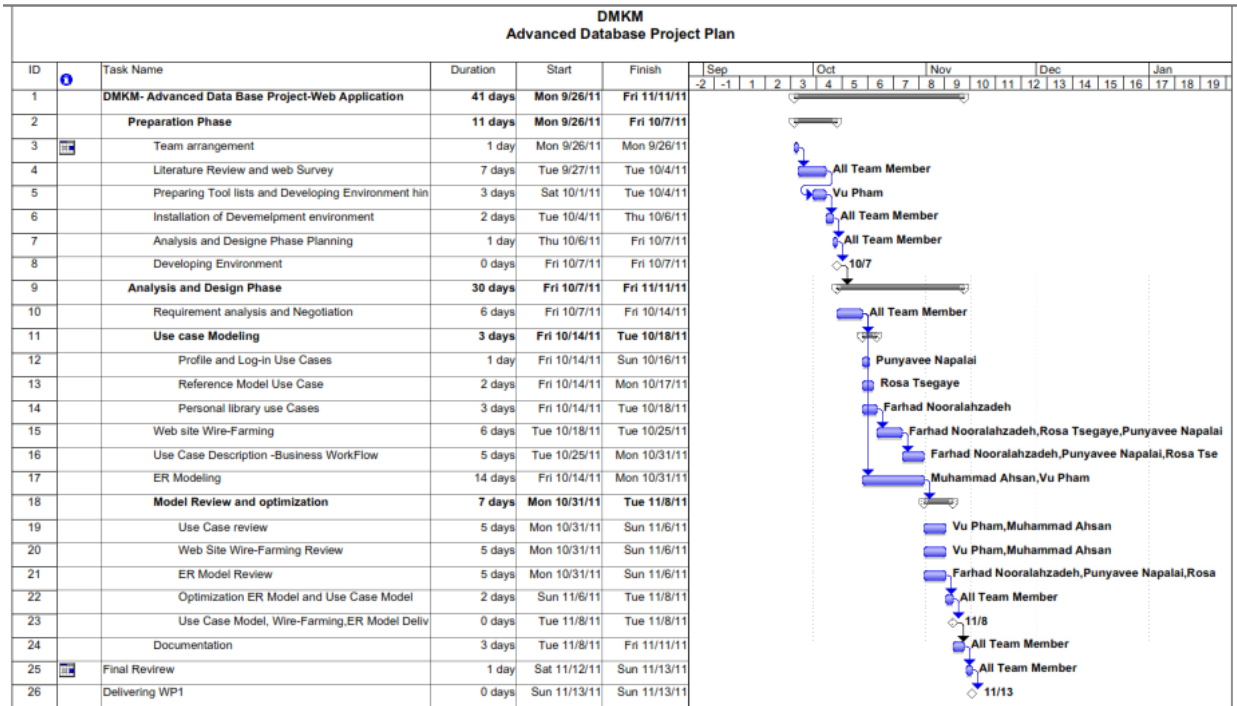
1.6 Issue Solving

Any problematic issues related to any part were presented on the discussion form to get input from each and every member of the team for better understanding and solving the problem in the best possible manner avoiding the team conflicts.

1.7 Progress of each assigned Task

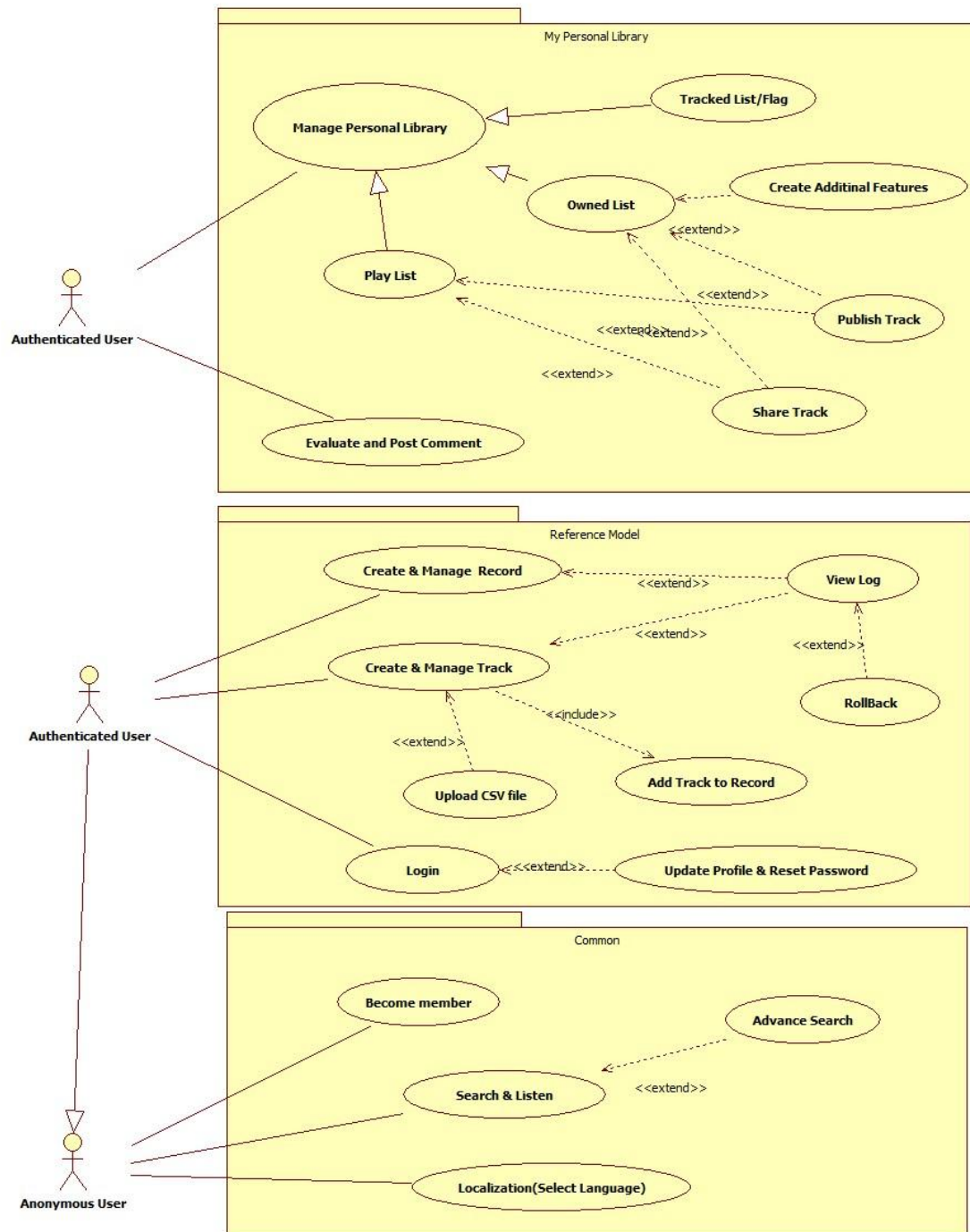
During each meeting the progress of each task was reviewed according to the deadline as specified by the Gantt chart.

1.8 Gantt Chart



Chapter 2 Use Case Diagram

2.1 User case diagrams



Chapter 3 Use Case Description

In this project we have 10 main use cases, which can be listed as follows:

- Owned List
- Play List
- Tracked List/Flag
- Evaluate and Post Comment
- Create & Manage Records
- Create & Manage Tracks
- View Log
- Log in
- Become member
- Search & Listen

Each use case is explained by below description.

3.1 Owned List

3.1.1 Brief Description

This use case allows the authenticated user to manage his/her Owned list.

The actor of this use case is authenticated user.

3.1.2 Flow of Events

The use case begins when the authenticated user selects the "My Personal Library" activity and goes to my owned list.

3.1.2.1. Basic Flow – Play

1. The authenticated user selects "My Personal Library"
2. The system displays a list of music tracks or user can search music tracks.
3. The authenticated user selects music from the lists to the playlist, owned list or Track list.
4. Select "Play" button to listen.

3.1.2.2. Alternative Flows

Add track to owned lists

1. The authenticated user search tracks in "My Personal Library" page.
2. The system displays a list of tracks with their Title, Artist, Record, Style and Rhythm.
3. The authenticated user select music from the displayed list then select "Add To owned list" button.

Create new tracks to my owned list

1. In my owned list page, the authenticated user selects "New".
2. System leads to New owned tracks page. Authenticated user gives information about this new owned track and clicks "Add to my owned list".

3. System adds new track to my owned list

View tracks in my owned list

1. In my owned list page, the authenticated user selects "View/Edit" to go to View/Edit owned track page.
2. Authenticated user can view the owned tracks detail.

Edit tracks in my owned list

1. In my owned list page, the authenticated user selects "View/Edit" to go to View/Edit owned track page.
2. Authenticated user can edit the owned tracks detail.

Share tracks from my owned list

1. In my owned list page, the authenticated user selects "Share" on the track he wants to share.
2. System shares the track.

Remove tracks from my owned list

1. In my owned list page, the authenticated user selects "Remove" on the track he wants to remove.
2. System removes the track.

3.1.2.3. Special Requirements

There are no special requirements associated with this use case.

3.1.2.4. Preconditions

Log In - Before this use case begins the authenticated user has logged onto the system.

3.1.2.5. Extensions

Add new field to my owned list

1. The authenticated user selects "My Owned List" to go to my owned list page.
2. Authenticated user selects add new field.
3. Then he/she can enter Field Caption and, if necessary, add description.

Publish owned list

Share owned list

1. The authenticated user selects playlist from owned list page
2. Then select share link to chosen usernames
3. Then select user names to share with the owned list

3.1.2.6. Post conditions

There are no post conditions for this use case.

3.2 Play List

3.2.1 Brief Description

In this use case, user defines Playlist based on Reference model. The actors of this use case are authenticated user.

3.2.2 Flow of Events

The use case begins when the user selects Playlist from My Personal Library view.

3.2.2.1. Basic Flow – Add Playlist

1. The user selects “New” from playlist page.
2. User enter desired Name for Playlist, privacy and description if it’s necessary
3. Then select ‘Create’ or ‘Cancel’ to not to create.

3.2.2.2. Alternative Flows

View playlist

1. User selects “View/Edit Playlist” from playlist page.
2. The authenticated user enter the name of playlist and the system shows all his/her “Play lists”

Delete tracks from playlist

1. User selects “Remove” from playlist page.
2. The system removes the track from playlist.

Edit Playlists

1. User can change the Name of Playlist and also select tracks from the track list to Playlist
2. User selects “Update” and system saves Playlist.

3.2.2.3. Special Requirements

User can create Play list from Reference Model.

3.2.2.4. Preconditions

User must log-in to system.

3.2.2.5. Extension

Publish owned list

Share owned list

1. The authenticated user selects playlist from owned list page
2. Then select share link to chosen usernames
3. Then select user names to share with the owned list

3.2.2.6. Post conditions

There are no any Post conditions associated with this use case.

3.3 Tracked List/Flag

3.3.1 Brief Description

In this use case, user defines the track as a tracked list. The actors of this use case are authenticated user.

3.3.2 Flow of Events

The use case begins when the user selects My Tracked List from My Personal Library view.

3.3.2.1. *Basic Flow – Add Tracked list*

1. In My Personal Library, user search track.
2. User enter select Add To Track List button.
3. System adds the track to Tracked List.

3.3.2.2. *Alternative Flows*

View Tracked List

1. In My Personal Library page, user selects My Personal Library.
2. System displays Tracked List

Remove tracks from Tracked list

1. User selects “Remove” from My Tracked list page on the track he/she wants to remove.
2. The system removes the track from Tracked list.

Upgrade to Owned List

1. User selects “Add To Owned List” from My Tracked list page on the track he/she wants to upgrade.
2. The system upgrades that track to Owned list.

3.3.2.3. *Special Requirements*

User can create Play list from Reference Model.

3.3.2.4. *Preconditions*

User must log-in to system.

3.3.2.5. *Extensions*

There are no any extensions associated with this use case.

3.3.2.6. *Post conditions*

There are no any Post conditions associated with this use case.

3.4 Evaluate and Post Comment

3.4.1 Brief Description

This use case allows the authenticated user to give a comment and evaluation to each track. The actor of this use case includes the authenticated user.

3.4.2 Flow of Events

The use case begins when the actor selects Comment and Evaluate from My Personal Library.

3.4.2.1. Basic Flow – Comment

1. In the Comment and Evaluate page, user can write the comment in the Your Comment field.
2. The user selects Submit button
3. System displays the comment on the screen.

3.4.2.2. Alternative Flow – Evaluate

1. In the Comment and Evaluate page, user can rate the track by giving stars ranging from 1-star to 5-star.
2. System displays the evaluated rate on the Overall Rate column on the track table.

3.4.3 Special Requirements

There are no special requirements associated with this use case.

3.4.4 Preconditions

The user must log in before giving a comment or evaluation.

3.4.5 Extensions

There are no extensions associated with this use case.

3.4.6 Post conditions

There are no Post conditions associated with this use case.

3.5 Create & Manage Records

3.5.1 Brief Description

This use case allows the authenticated user to create and manage records, which is a collection of tracks. This use case has functionality to create new records, search tracks from created records, assign tracks on created records, view and edit records are included in this use case.

The actors of this use case are the authenticated users.

3.5.2 Flow of Events

The use case begins when the authenticated user selects 'Reference Database' to go to the record interface from the Reference database interface, then select the record from reference data base interface and then reach to record interface. From record interface the authenticated user selects 'New' to create new record or 'Assign track' to assign new track or 'View/Edit' to modify the record.

3.5.2.1. Basic Flow – Create

1. The authenticated user selects "New" button from record interface
2. The authenticated user inputs in the empty field of 'Title', 'Artist', 'Producer', 'Matrix Number', 'Press info' and select about it's type of record from the radio button whether it's 'LP' or 'Double LP' or 'EP' or 'Double EP' or 'Maxi Single' or 'Single' record.
3. Select "Add" and the record is created.

3.5.2.2. Alternative Flows– Manage

Assign tracks to list of records

1. In the List of record page, the authenticated user selects "Assign Tracks" from the music they want to put into this record.
2. On the search and assign track interface page, the authenticated user inputs in the empty field 'Title', 'Artist', 'Label', 'Rhythm' and 'Style' then select 'Search Track' button. Then when the track list display he/she selects 'Assign Track'.
3. Then system adds that track into the record.

View record from list of records

1. In the record page, the authenticated user selects "View" from the music they want to view. This will lead to Record info page.
2. The system display record features.

Edit record information

1. In the List of records page, the authenticated user selects "Edit" from the music they want to edit.
2. The system display latest update record features and the authenticated user can edit them.
3. After editing, they need to select "Save" and the system saves the updated record features and the system also saves the old data as a log.

Remove record from list of records

1. In the List of records page, the authenticated user selects "Remove" from the music they want to delete.
2. The system deletes that track.

3.5.3 Special Requirements

There are no special requirements associated with this use case.

3.5.4 Preconditions

Log In - Before this use case begins the authenticated user or the administrator has to log onto the system.

3.5.5 Extensions

View Log

3.5.6 Post conditions

There is no post conditions associated with this use case.

3.6 Create & Manage Tracks

3.6.1 Brief Description

This use case allows the authenticated user to create and manage track. The track can be added more features after it is created, edited its features or removed.

The actors of this use case are the authenticated user.

3.6.2 Flow of Events

The use case begins when the authenticated user selects the "Search" activity from the Home page. They can select "new" activity to create the track and "edit/view", "delete" activity in the track page to manage it.

3.6.2.1. Basic Flow – Create

1. The authenticated user selects "New" in the track page.
2. The authenticated user selects path of music file and enters Title, Artist and Browse a Track. Then they select "Add to Record" to create a new track.
3. Select "Other features", if they need to add more features to the track
4. In Other features page, they can input Artist, Rhythm, Original Version, Label, Playing Time, Release Date and Style. Then they select "Save" to save the features of the new track.
5. The track is uploaded and the system records the track information.

3.6.2.2. Alternative Flows– Manage

View track from list of track

1. In the track page, the authenticated user selects "edit/view" from the music they want to view.
2. The system displays track features.

Edit track information

1. In the track page, the authenticated user selects "edit/view" from the music they want to edit. This will lead to View/Edit Track page.
2. The system displays latest update track features and the authenticated user can edit them.
3. After editing, they need to select "Update" and the system save the updated features and the system also saves the old data as a log.

Remove track from list of track

1. In the track page, the authenticated user selects "remove" from the music they want to delete.
2. The system removes that track.

3.6.3 Special Requirements

There are no special requirements associated with this use case.

3.6.4 Preconditions

Log In - Before this use case begins the authenticated user or the administrator has to log onto the system.

3.6.5 Extensions

View Log

Upload CSV file

1. In the side bar named "Reference database". User clicks "Upload CSV file"
2. System links to Upload CSV file page and user choose CSV files from his/her directory to upload.

3.6.6 Include

Add Track to Record

1. User selects New Track from Track page.
2. User fills the empty fields of Title, Artist and directory for uploading file and selects Add to Record button.
3. Record list is displayed and user selects records by searching using filter choices.
4. User selects save button. System adds the track to selected records.

3.6.7 Post conditions

There are no Post conditions associated with this use case.

3.7 View Log

3.7.1 Brief Description

As system keeps all creation and update transaction on reference model, user can view and rollback based on this use case.

The actors of this use case are authenticated user.

3.7.2 Flow of Events

The use case begins when the user selects view Log in View/Edit Track and View/Edit Record page.

3.7.2.1. Basic Flow – View Log (History)

The user selects "View Log" from the Track or Record. It will lead the user to View Track Log page or View Record Log page

5. User can search date boundary to view logs.

6. System brings all logs in two categories, first category bases on “Records” in View Record Log page and second one bases on “Tracks” in View Track Log page. The system shows logs for every changed records/Tracks which happened in selected dates. If there is no selected date, system will bring logs information for 2 weeks before.
7. User can select the log row for the records of tracks which wants to see the detail information.
8. System shows the detail information about selected row.

3.7.2.2. Alternative Flows

Show Differences

1. User select “Show differences” at the row which user wants to see the differences.
2. System leads to Show Differences page. In this page, user can see details of changes, including Instance, Date, Changed field, Previous value, New value and User Name.

3.7.3 Special Requirements

System must keep all created and updated logs on Reference Model (Tracks and Records).

3.7.4 Preconditions

User must log-in to system.

3.7.5 Extensions

Rollback

1. User select “Rollback” at the row which user wants to Rollback to .
2. System shows “After Rollback, you will lose information after selected logs” alert to user.
3. If user selects “Yes”, system will rollback selected record/track.
4. If user selects “No”, system will not concern rollback process. And return to logs view.

3.7.6 Post conditions

There are no Post conditions associated with this use case.

3.8 Login

3.8.1 Brief Description

This use case explains how use can login to the system and also how can register user as an Authenticated user.

3.8.2 Flow of Events

The use case begins when the user selects the log in in first page after he put web-site URL in explorer.

3.8.2.1. Basic Flow – log in

1. The user selects “log-in” and system ask him/her user name and password.

2. User enters User name and Password and selects “Log in”.
3. System checks the user and password and if everything are correct , system retrieve “personal view” of related username
4. System shows related “Personal View”

3.8.2.2. Alternative Flows

Update profile

1. User goes to “Profile” from main menu bar.
2. System displays latest data on the Profile page. User can update his/her information in the fields and upload profile picture.
3. User selects OK to confirm his/her update and system saves updated information.

Forget password

1. If user has user name but he/she doesn’t remember the password. He/she must click on “Forget User Name or Password” option on Login form.
2. System shows “forget Password” form and asks user name and email address.
3. User enters user name and email address and selects “ok”.
4. System verifies the user name and email address and if it finds proper information, it will send password to user by email and show “the password has been sent to your email” message to user.
 - a. If system could not find any information based on entered data, system shows “Sorry, there is no any user name based on entered data” to user.

Reset password

1. User goes to “Reset Password” from Profile.
2. User inputs empty field of Old Password, New Password and Retype Password. Then press “OK”.
3. System checks whether all fields are entered and whether New Password and Retype Password are the same.
4. System changes the password.

3.8.3 Special Requirements

There are no special requirements associated with this use case.

3.8.4 Preconditions

There are no preconditions associated with this use case.

3.8.5 Post conditions

After log-in, user will have personal view and also have authorization as a “Authenticated user”

3.9 Become member

3.9.1 Brief Description

This use case explains how use can sign up to the system.

3.9.2 Flow of Events

The use case begins when the user selects Become member in the first page after he put web-site URL in explorer.

3.9.2.1. Basic Flows– become member

User doesn't have User Name."New User"

1. User selects "Become member" from the homepage to define user name.
2. System shows "User definition Form".
3. User follows the Form and desired "user name".
4. System check "user name" to be not redundant.
 - a. If user name was existed, system shows "this username was existed, please change the user name or try to find your user name by "forget password" section.
 - b. If user name was not existed, system shows "acceptable user name "message in green.
5. User follows entering data and fulfills the mandatory fields and selects "Submit"
6. System checks existence of mandatory information and create user name and determine user as an authenticated user.
7. System sends confirmation email based on address which mentioned in user account.

3.9.3 Special Requirements

There are no special requirements associated with this use case.

3.9.4 Preconditions

There are no preconditions associated with this use case.

3.9.5 Post conditions

After log-in, user will have personal view and also have authorization as a "Authenticated user"

3.10 Search/Listen

3.10.1 Brief Description

This use case allows the anonymous user and the authenticated user to use full-text search over reference database and personal libraries. They have multi-criteria search and filter facility and they can switch from Records View, Detail View or Track View.

The actor of this use case includes the anonymous user and authenticated user.

3.10.2 Flow of Events

The use case begins when the actor enter texts into the search input. Search input facility can find anywhere in the webpage.

3.10.2.1. Basic Flow – Search

1. The actor enters texts into the search input.
2. The result will be shown in the result table. The default view of the result is Track view.
3. The actor can listen to the searched music directly.

3.10.2.2. Basic Flow – Switch view

1. The search result can be switched the view between Records View, Detail View or Track View.

3.10.3 Special Requirements

There are no special requirements associated with this use case.

3.10.4 Preconditions

There are no preconditions associated with this use case.

3.10.5 Extensions

Advanced search

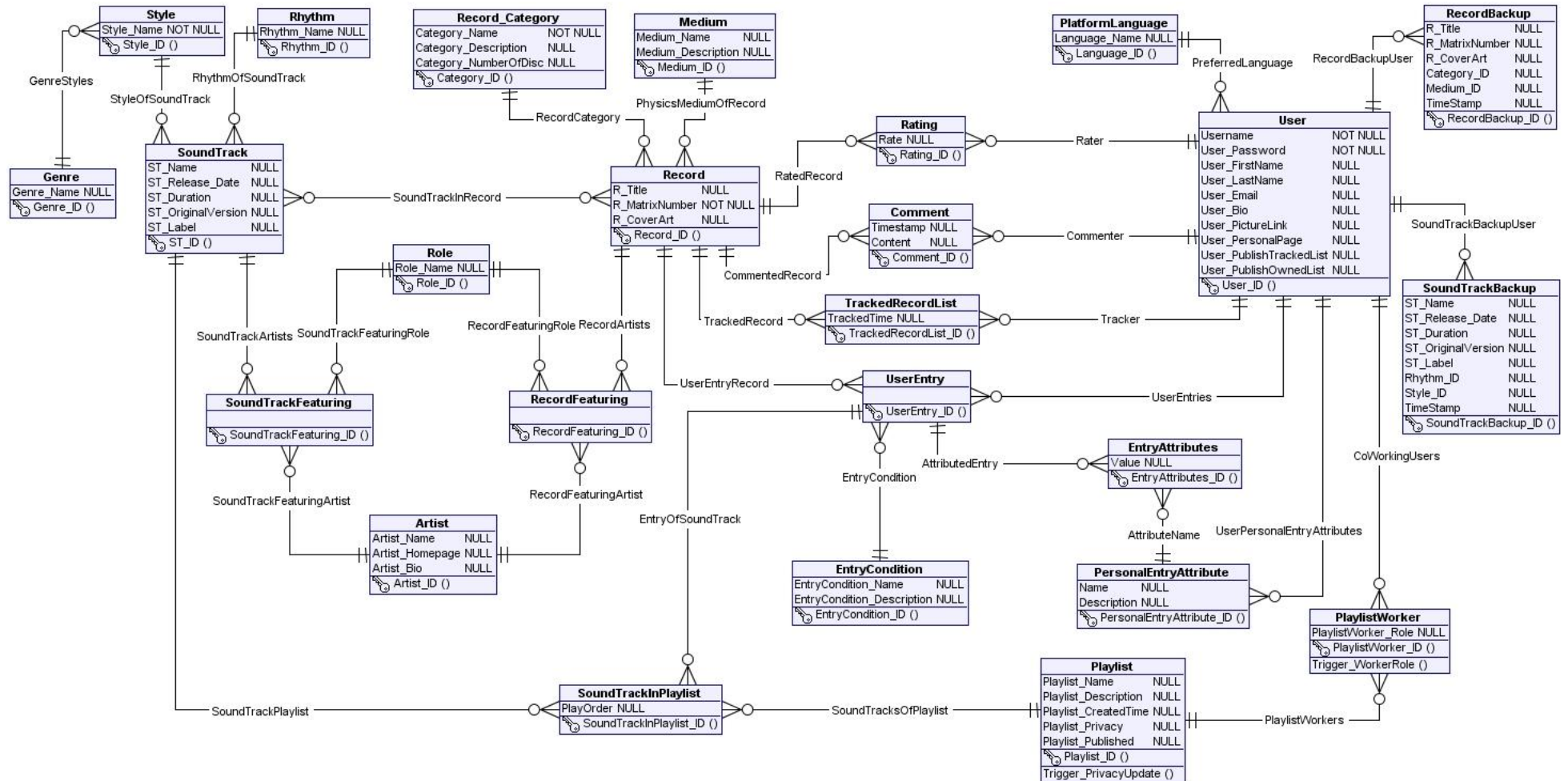
1. Advanced search – multi-criteria search
The user can tick the boxes for multi-criteria search.
2. Advanced search – filter
User can filter search.
3. Advanced search – sorting
User can sorting search.

3.10.6 Post conditions

There are no Post conditions associated with this use case.

Chapter 4 Entity-Relationship Diagram

4.1 The Entity-Relationship Diagram



4.2 Entity-Relationship Diagram Description

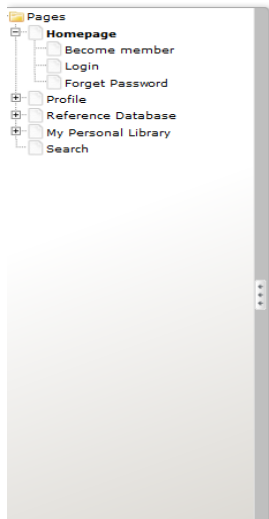
The first-class citizen of the system is stored in the *Record* entity. There is a *many-to-many* relationship between *Record* and *SoundTrack*. Here are some special remarks about this ERD:

- All artists of the system are described by the *Artist* entity. Since an artist can play different roles in a given soundtrack/record, and a soundtrack/record can have different artists, there are *many-to-many* relationships between *Artist - SoundTrack* and *Artist - Record*. On this relationship, however, we need to define the role for each pair of Artist-SoundTrack and Artist-Record, hence we have the *Role* entity. The many-to-many relationships are then described in the *SoundTrackFeaturing* and *RecordFeaturing* entities. Actually *SoundTrackFeaturing* and *RecordFeaturing* should not be entities in an ERD. However, since our tool for drawing ERD does not support attaching attributes on a relationship, we have to create *SoundTrackFeaturing* and *RecordFeaturing* explicitly in the ERD. For a *SoundTrack*, roles can be: artist, invited artist (featuring), music players, audio engineer, lyrics writer, music writer... For a *Record*, roles can be: artist, producer... For simplicity, we don't differentiate between roles for *Record* and roles for *SoundTrack*.
- The user's owned list is described by the *UserEntry* entity. Each user entry is a record in the reference library, or a copy of a record in the reference library. Since users can define their own attributes for their owned entries, all of the custom attributes of a user is stored in the *PersonalEntryAttribute*. A user entry can have many custom attributes, and a custom attribute can be used in many user entry, hence we have a *many-to-many* relationship between *UserEntry* and *PersonalEntryAttribute*. In addition, on this *many-to-many* relationship, we have to define the value of the custom attribute for the given user entry. This many-to-many relationship is then described by the *UserEntry - EntryAttributes - PersonalEntryAttribute* relationship.
- Playlists are stored in the *Playlist* entity. Since it is possible that many users working on the same playlist, and many playlists can belong to a user, we have a many-to-many relationship between *User* and *Playlist*. However there are two possibilities for a user working on a playlist: the user can be the owner of that playlist, or he is invited to work on that playlist. Therefore we have the *PlaylistWorker* entity, with the *PlaylistWorker_Role* attribute which specifies the role for the user in the given playlist. Obviously, all of the users working on the same playlist forms a list of co-workers, this is the reason for the name *PlaylistWorker*.
- Each entry in a playlist is a soundtrack, from the given record. The record is only allowed to choose from the user's owned list. Consequently, we have a *SoundTrackInPlaylist* entity to describe the complicated relationship between *SoundTrack*, *UserEntry* and *Playlist*. In *SoundTrackInPlaylist*, we have the additional attribute named *PlayOrder* identifying the order of a soundtrack inside the playlist.
- For publishing the tracked list and the owned list to the entire community of the system, for each *User*, we have *User_PublishTrackedList* and *User_PublishOwnedList* attributes for controlling these settings. *Playlist* entity also has the *Playlist_Published* attribute to decide whether or not publishing this playlist to the whole community. Furthermore, *Playlist* has a *Playlist_Privacy* attribute, which let the owner decide how to grant the permission for co-workers of the playlist. For simplicity, we only support read and right access for the playlist.
- All of the supported language of the system is stored in the *PlatformLanguage* entity, rather than attached as an attribute of the *User* entity. We do this to maintain the flexibility of the database system, which should not be dependent on the application.

Other entities and relationships are straightforward.

Chapter 5 Wireframe

Homepage



<Site Name> [Become member](#) [Login](#) [Language](#)

Homepage

Listen Music with us.
Become member and Create your Personal Library
(Owned List,PlayList) and Share with your freinds

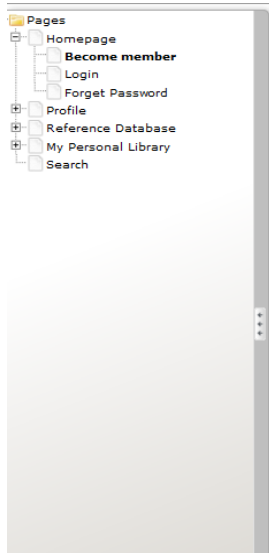
Find your Music

☒ Everything ☒ Artist ☒ Track
☒ Record

Title	Artist	Style	Record
Miami	U2	Pop	AAAA
Do you feel a Love	U2	Pop	AAAA

You are listening to:
<< Track name>>

Become member



Home Page -----> Become member

<Site Name>

Homepage

First Name*

Last Name

User Name*

Password*

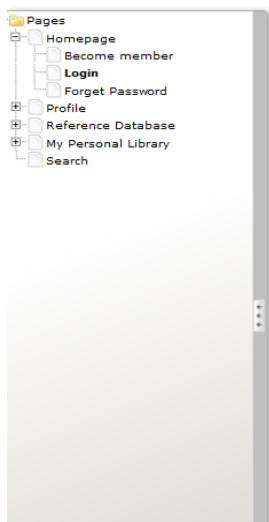
Retype Password*

Email Address*

Preferred

* Required

Login



Home Page -----> Log In

<Site Name> [Community](#) [Profile](#) [Language](#)

Homepage

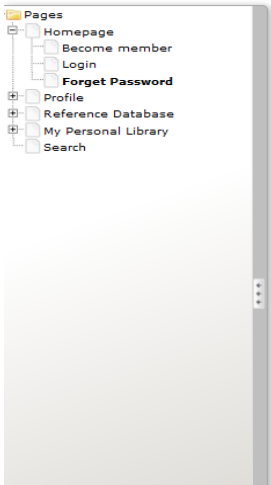
Hi-<<User First Name <Last Name>>

User Name*

Password

[Forget User Name or Password](#)

Forget Password



Home Page -----> Forget Password

Community Profile Language

Homepage	Hi:<<User First Name +Last Name>> Don't worry, we'll send an email with instructions to access your account. Unless you remember? User Name or Email <input type="text"/> <input type="button" value="Close"/> <input type="button" value="Submit"/>
Profile	
Reference Database	
My Personal Library	
Search	

Profile



<Site Name> Community Profile Language

Homepage	Hi:<<User First Name +Last Name>> First Name* <input type="text"/> Profile Picture <input type="text"/> Last Name <input type="text"/> <input type="button" value="Browse"/> User Name* <input type="text"/> Email Address* <input type="text"/> Preferred English <input type="button" value="v"/> Social Network <input type="text"/> Personal Web-Site <input type="text"/> <input type="button" value="Ok"/> <input type="button" value="Cancel"/>
Profile	
Reference Database	
My Personal Library	
Search	

Reset Password

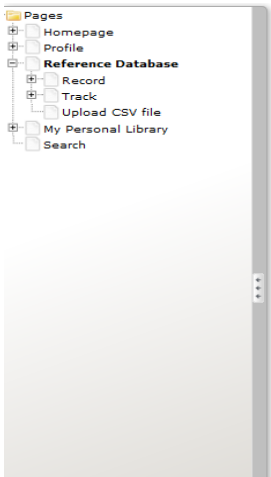


Profile -----> Reset Password

<Site Name> Community Profile Language

Homepage	Hi:<<User First Name +Last Name>> Old Password* <input type="text"/> New Password* <input type="text"/> Retype Password* <input type="text"/> <input type="button" value="OK"/>
Profile	
Reference Database	
My Personal Library	
Search	

Reference Database



Reference Database

<Site Name> Profile Language

Hi:<<User First Name +Last Name>>

Homepage

Profile

Reference Database

My Personal Library

Search

Search/Update Record

Title

Artist

Producer

Matrix Number

Category

LP
Double LP
EP
Double EP
Single
Maxi single

Search/Update Track

Title

Rhythm

Label

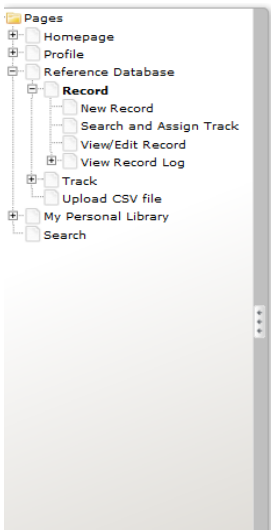
Style

Reggae
Ska
Rock steady
Early digital
Roots
Dancehall

Search Record

Search Track

Record



Reference Database -----> Record

<Site Name> Community Profile Language

Hi:<<User First Name +Last Name>>

Homepage

Profile

Reference Database

My Personal Library

Search

Find Record

☒ Everything ☐ Artist ☐ Track

☒ Record

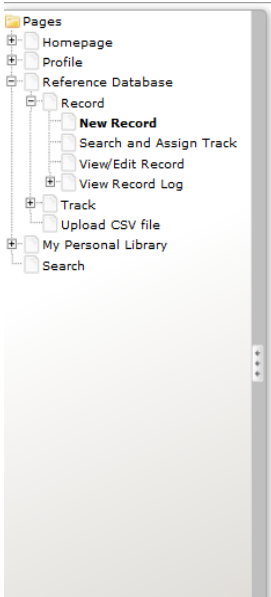
Record List New

Title	Artist	Producer	Category	Matrix No.
....
....

View/Edit Remove Assign Track view Log

View/Edit Remove Assign Track view Log

New Record



Reference Database -----> Record -----> New Record

<Site Name> Community Profile Language

Hi:<<User First Name +Last Name>>

Homepage

Profile

Reference Database

My Personal Library

Search

Title*

Artist

Producer

Matrix Number

Press info

LP
Double LP
EP
Double EP
Maxi Single
Single

Save

Search and Assign Track

Pages

Homepage

Profile

Reference Database

Record

New Record

Search and Assign Track

View/Edit Record

View Record Log

Track

Upload CSV file

My Personal Library

Search

Reference Database -> Record -> Search and Assign Track

<Site Name> Community Profile Language

Homepage

Profile

Reference Database

My Personal Library

Search

Search and assign Track

Search Track and assign to Record << Record Title >>

Title

Artist

Label

Rhythm

Style

Reggae
Ska
Rock steady
Funky dital

Search Track

Track List

Title	Artist	Playing Time	Rhythm	Style
.....
.....

Assign Track

View/Edit Record

Pages

Homepage

Profile

Reference Database

Record

New Record

Search and Assign Track

View/Edit Record

View Record Log

Track

Upload CSV file

My Personal Library

Search

Reference Database -> Record -> View/Edit Record

<Site Name> Community Profile Language

Homepage

Profile

Reference Database

My Personal Library

Search

Hi<<User First Name +Last Name>>

Title*

Artist

Producer

Matrix Number

Press info

LP

Double LP

EP

Double EP

Maxi Single

Single

Find Track

Everything Artist Title Rhythm

Track List

Title	Artist	Playing Time	Rhythm	Style
.....
.....

View/Edit Remove view Log

Save

Show Differences

Pages

Homepage

Profile

Reference Database

Record

New Record

Search and Assign Track

View/Edit Record

Show Differences

View Record Log

Track

Upload CSV file

My Personal Library

Search

Reference Database -> Record -> View Record Log -> Show Differences

<Site Name> Community Profile Language

Homepage

Profile

Reference Database

My Personal Library

Search

Hi<<User First Name +Last Name>>

Instance	Date	Changed Filed	Previous Value	New Value	User Name
2	09/11/2011 13:30	Title	B	A	Farhad Nooralahzadeh
2	09/11/2011 13:30	Artist	BB	AA	Farhad Nooralahzadeh
2	09/11/2011 13:30	Matrix Number	BBB	AAA	Farhad Nooralahzadeh

View Record Log

Pages

Homepage

Profile

Reference Database

Record

New Record

Search and Assign Track

View/Edit Record

View Record Log

Show Differences

Track

Upload CSV file

My Personal Library

Search

Reference Database

Record

View Record Log

<Site Name>

Community

Profile

Language

Homepage

Profile

Reference Database

My Personal Library

Search

Hi<<User First Name +Last Name>>

You are looking at Log history of << Record Title>>

Instance	Date	Created by /Updated by	
1	09/11/2011 12:00	Farhad Nooralahzadeh	Show Differences Rollback
2	09/11/2011 13:30	Farhad Nooralahzadeh	Show Differences Rollback
3	10/11/2011 8:00	Punyavee Napale	Show Differences Rollback

Track

Pages

Homepage

Profile

Reference Database

Record

Track

New Track

Other features

View/Edit Track

View Track Log

Upload CSV file

My Personal Library

Search

Reference Database

Track

<Site Name>

Community

Profile

Language

Homepage

Profile

Reference Database

My Personal Library

Search

Hi<<User First Name +Last Name>>

Find Tack

☒ Everything

☐ Artist

☐ Track

☒ Record

Track List

Title	Artist	Style	Rhythm	Record	
.....	<div><div></div><div>View/Edit</div><div>Remove</div><div>view Log</div></div>
.....	<div><div></div><div>View/Edit</div><div>Remove</div><div>view Log</div></div>

New

New Track

Pages

Homepage

Profile

Reference Database

Record

Track

New Track

Other features

View/Edit Track

View Track Log

Upload CSV file

My Personal Library

Search

Reference Database

Track

New Track

<Site Name>

Homepage

Profile

Reference Database

My Personal Library

Search

Hi<<User First Name +Last Name>>

Title*

Artist

Browse a Track*

Upload

Add to Record

Other Features

Find Record

☒ Everything

☒ Artist

☒ Title

☐ Producer

Record list

Title	Artist	Producer	
.....	<input checked="" type="checkbox"/>
.....	<input type="checkbox"/>

Save

Other features

Pages

Homepage

Profile

Reference Database

Record

Track

New Track

Other features

View/Edit Track

View Track Log

Upload CSV file

My Personal Library

Search

Reference Database —> Track —> Other features

<Site Name>CommunityProfileLanguage

Hi<<User First Name +Last Name>>

Homepage

Profile

Reference Database

My Personal Library

Search

Artist/Music Player/Audio

New

Name	Role
.....

View/EditRemove

Rhythm

Original Version

Label

Playing Time

Release Date

Style

Hyper link

Reggae

Ska

Rock steady

Early digital

Roots

Dancehall

Digital roots

Jazz

Classical

Pop

Save

View/Edit Track

Pages

Homepage

Profile

Reference Database

Record

Track

New Track

Other features

View/Edit Track

View Track Log

Upload CSV file

My Personal Library

Search

Reference Database —> Track —> View/Edit Track

<Site Name>CommunityProfileLanguage

Hi<<User First Name +Last Name>>

Homepage

Profile

Reference Database

My Personal Library

Search

Title*

Artist*

Artist/Music Player/Audio

New

Name	Role
.....	Invited Artist
.....	Invited Artist
.....	Music Player
.....	Audio Engineer
.....	Lyrics Writer
.....	Music Writer

View/EditRemoveView/EditRemove

Original Version

Label

Playing Time

Release Date

Browse a Track*

Rythm

Style

Hyper link

Reggae

Ska

Rock steady

Early digital

Roots

Dancehall

Digital roots

Jazz

Classical

Pop

Update

View Track Log

Pages

Homepage

Profile

Reference Database

Record

Track

New Track

Other features

View/Edit Track

View Track Log

Show Differences

Upload CSV file

My Personal Library

Search

Reference Database —> Track —> View Track Log —> Show Differences

<Site Name>CommunityProfileLanguage

Hi<<User First Name +Last Name>>

Homepage

Profile

Reference Database

My Personal Library

Search

You are looking at Log history of << Track Title>>

Instance	Date	Created by /Updated by
1	09/11/2011 12:00	Farhad Nooralahzadeh
2	09/11/2011 13:30	Farhad Nooralahzadeh
3	10/11/2011 8:00	Punyavee Napale

Show DifferencesRollbackShow DifferencesRollbackShow DifferencesRollback

Show Differences

Pages

Homepage

Profile

Reference Database

Record

Track

New Track

Other features

View/Edit Track

View Track Log

Show Differences

Upload CSV file

My Personal Library

Search

<Site Name>

Community

Profile

Language

Homepage

Profile

Reference Database

My Personal Library

Search

Hi<<User First Name +Last Name>>

Instance	Date	Changed Field	Previous Value	New Value	User Name
3	10/11/2011 13:30	Title	B	A	Punyave Napale
3	10/11/2011 13:30	Artist	BB	AA	Punyave Napale
3	10/11/2011 13:30	Style	BBB	AAA	Punyave Napale

Upload CSV file

Pages

Homepage

Profile

Reference Database

Record

Track

Upload CSV file

My Personal Library

Search

Reference Database

Upload CSV file

<Site Name>

Community

Profile

Language

Homepage

Profile

Reference Database

My Personal Library

Search

Hi<<User First Name +Last Name>>

Browse a CSV

Brows & Upload

My Personal Library

Pages

Homepage

Profile

Reference Database

My Personal Library

My Owned list

Playlist

My Tracked List

Comment and Evaluta

Add to Playlist

Search

<Site Name>

Community

Profile

Language

Homepage

Profile

Reference Database

My Personal Library

Search

Hi<<User First Name +Last Name>>

Find Track

☒ Everything

☐ Artist

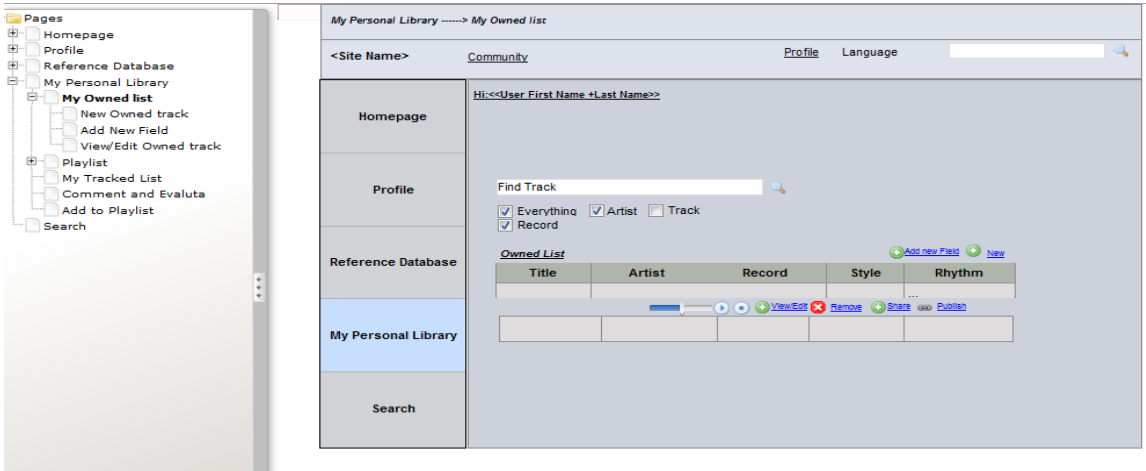
☒ Track

☒ Record

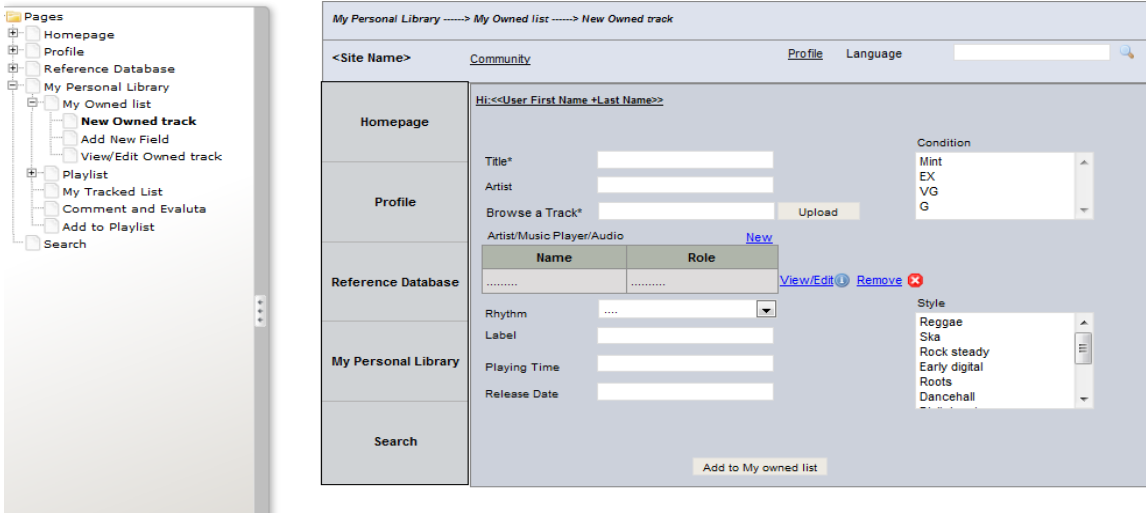
Track List

Title	Artist	Record	Style	Rhythm

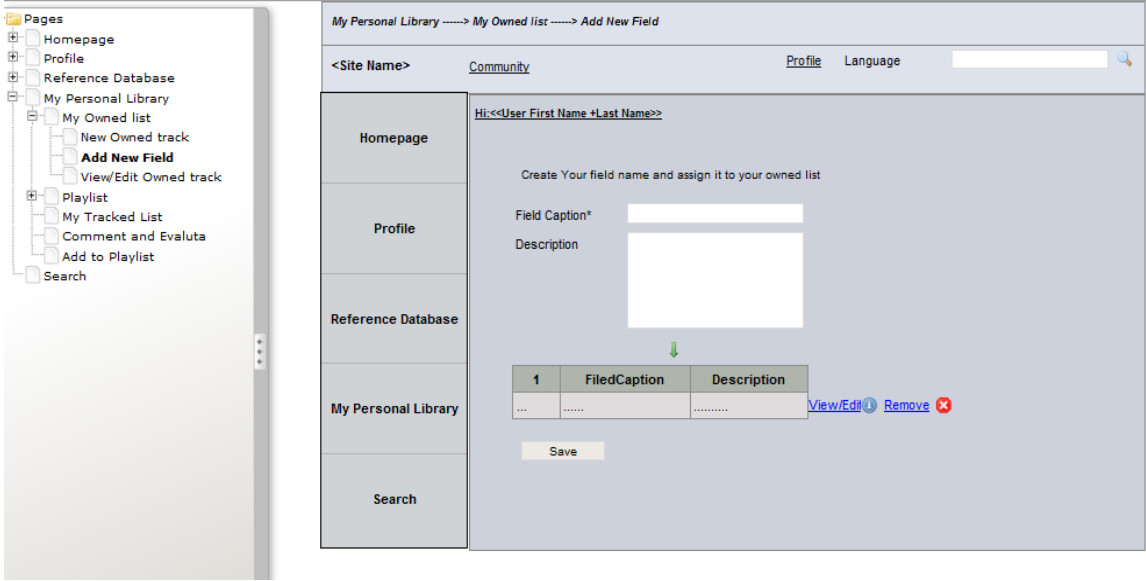
My Owned list



New Owned track



Add New Field



View/Edit Owned track

Pages

Homepage

Profile

Reference Database

My Personal Library

My Owned list

New Owned track

Add New Field

View/Edit Owned track

Playlist

My Tracked List

Comment and Evaluta

Add to Playlist

Search

My Personal Library -----> My Owned list -----> View/ Edit Owned track

<Site Name>CommunityProfileLanguage

Homepage

Profile

Reference Database

My Personal Library

Search

Hi:<<User First Name +Last Name>>

Title*
Artist
Browse a Track*
Artist/Music Player/Audio
Rhythm
Label
Playing Time
Release Date
User Defined Field

Name

Role

View/Edit

Remove

Condition

Style

Update

Playlist

Pages

Homepage

Profile

Reference Database

My Personal Library

My Owned list

Playlist

Create Playlist

View/Edit Play List

Share PlayList

My Tracked List

Comment and Evaluta

Add to Playlist

Search

My Personal Library -----> Playlist

<Site Name>CommunityProfileLanguage

Homepage

Profile

Reference Database

My Personal Library

Search

Hi:<<User First Name +Last Name>>

Playlist view

Playlist name	Description	Creation Date	Privacy
....
....
....

View/Edit

Remove

Share

Public

Remove Sharing

Create Playlist

Pages

Homepage

Profile

Reference Database

My Personal Library

My Owned list

Playlist

Create Playlist

View/Edit Play List

Share PlayList

My Tracked List

Comment and Evaluta

Add to Playlist

Search

My Personal Library -----> Playlist -----> Create Playlist

<Site Name>CommunityProfileLanguage

Homepage

Profile

Reference Database

My Personal Library

Search

Hi:<<User First Name +Last Name>>

Create your Playlist

Playlist Name*

Description

Privacy

Create

Cancel

View/Edit Play List

Pages

Homepage

Profile

Reference Database

My Personal Library

My Owned list

Playlist

Create Playlist

View/Edit Play List

Share PlayList

My Tracked List

Comment and Evaluta

Add to Playlist

Search

My Personal Library -----> Playlist -----> View/Edit Play List

<Site Name>CommunityProfileLanguage

Homepage

Profile

Reference Database

My Personal Library

Search

Hi:<<User First Name +Last Name>>

Name

Description

Privacy

Track List

Title	Artist
.....
.....
.....

Shared List

User Name
.....
.....
.....

Update

Share PlayList

Pages

Homepage

Profile

Reference Database

My Personal Library

My Owned list

Playlist

Create Playlist

View/Edit Play List

Share PlayList

My Tracked List

Comment and Evaluta

Add to Playlist

Search

My Personal Library -----> Playlist -----> Share PlayList

<Site Name>CommunityProfileLanguage

Homepage

Profile

Reference Database

My Personal Library

Search

Hi:<<User First Name +Last Name>>

Share_Playlist <<Playlist Name>>

Select user to Share Playlist with them

User Name

☐

.....

☒

.....

☐

.....

Share

My Tracked List

Pages

Homepage

Profile

Reference Database

My Personal Library

My Owned list

Playlist

My Tracked List

Comment and Evaluta

Add to Playlist

Search

My Personal Library -----> My Tracked List

<Site Name>CommunityProfileLanguage

Homepage

Profile

Reference Database

My Personal Library

Search

Hi:<<User First Name +Last Name>>

Find Track

☒ Everything

☒ Artist

☐ Track

☒ Record

Tracked List

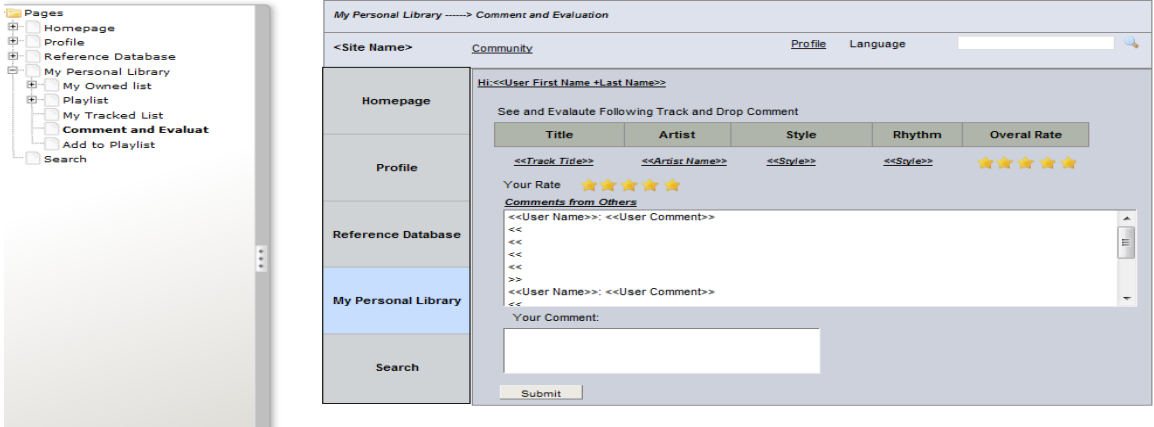
Title	Artist	Record	Style	Rhythm
.....

View

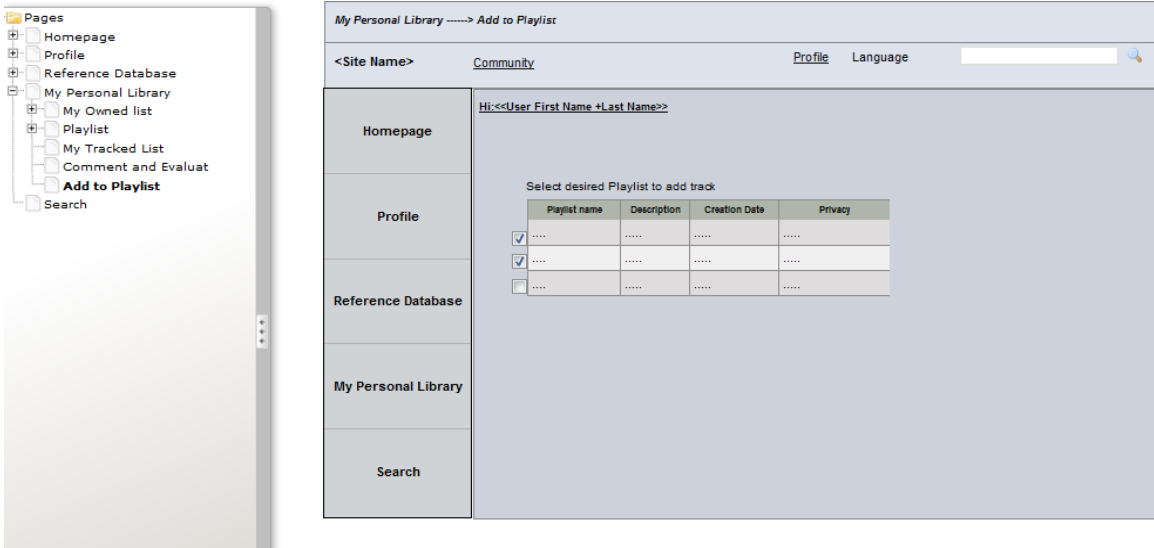
Remove

To Owned List

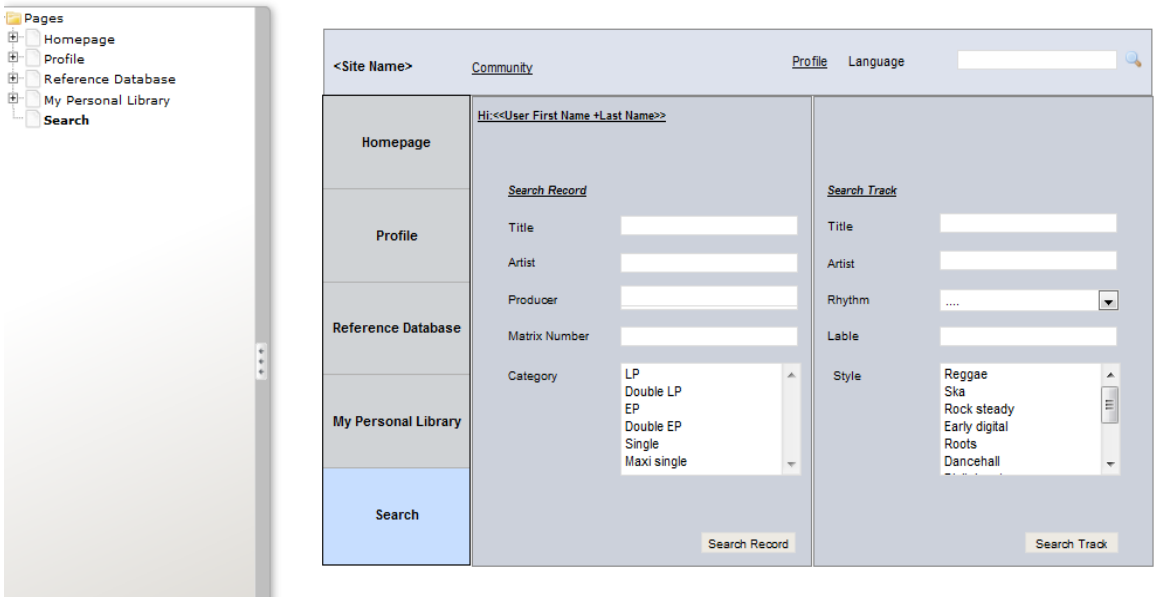
Comment and Evaluate



Add to Playlist



Search



Conclusion

During the preparation of the first phase of project, we try to use all the recommended resource on the project statement. We found it good base to complete our first phase of the project successfully.

We get good experience on the resources that we used as long as most of them were new for us, and also it was good experience doing project and managing distance place meetings together. Additionally, it was good opportunity doing this project with different background colleagues.

Mainly, this project shows us and learns us on how to organize and manage things with tight schedule.