Faculty of Engineering, Alexandria University
Computers and Systems Engineering Department

Software Engineering
Milestone 4

Music Zone: An Online Virtual Music Listening Platform

Prepared By

Bishoy Nader Fathy	**[21]**
Amr Mohamed Nasreldin	**[47]**
Marc Magdi Kamel	**[55]**
Michael Raafat Mikhail	**[57]**

User Stories Covered in this Milestone

1. User can change his information, i.e. name, email
2. User can change his selected prefered tags.
3. User can search for music based on the tags and title.
4. User can choose a music track to play.
5. User can like or add a track to his favourite library.

Requirement Elicitation

User can change his information

1. User should have the ability to change his main information, name or email.
2. Each user should have a unique email from other user's mail in the system.
3. Trying to edit the user data and leaving one of the mandatory fields will result in an error.
4. Password should be provided with the edit of the user info.
5. Editing the data successfully should redirect to the home page.

User can change his selected prefered tags

1. The user should have a link to a page that allows him to edit his preferred tags.
2. The page should list all tags present in the system with the user prefered tags selected.
3. The user should have the ability to swap a tag from being prefered to not and vice versa by simply clicking on it.
4. The user should be able to select multiple tags.
5. In case user submits without any tags being selected he should stays in the same page to choose at least one tag.
6. In case user submits after choosing at least one tag he should be redirected to the site home page.

User can like or add a track to his favourite library

1. When the user opens the home page, he should be able to see tracks that have the same tags that he has chosen as favorite tags before,
2. Each Track should have a like button that is toggled between on and off by clicking on it.
3. Each Track should have an add to favourites button that is toggled between on and off by clicking on it.
4. The user should have a link to a page that allows him to see his liked tracks, This page should be similar in design and functionality to the home page but showing liked tracks instead of tracks belonging to prefered tags.
5. The user should have a link to a page that allows him to see his favourite tracks, This page should be similar in design and functionality to the home page but showing favourite tracks instead of tracks belonging to prefered tags.

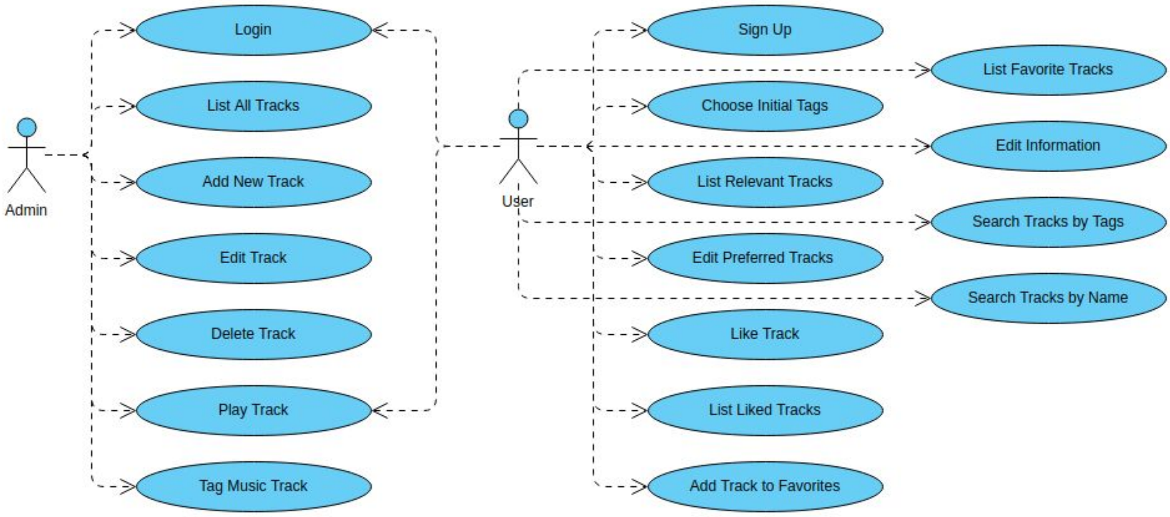
User can search tracks by title.

1. When the user opens the home page, he should be able to see a search box at the top of the page.
2. The user should be able to enter a substring of a title he is searching for.
3. The search results should show all tracks with titles containing the searched substring in a case insensitive matter from the whole database.

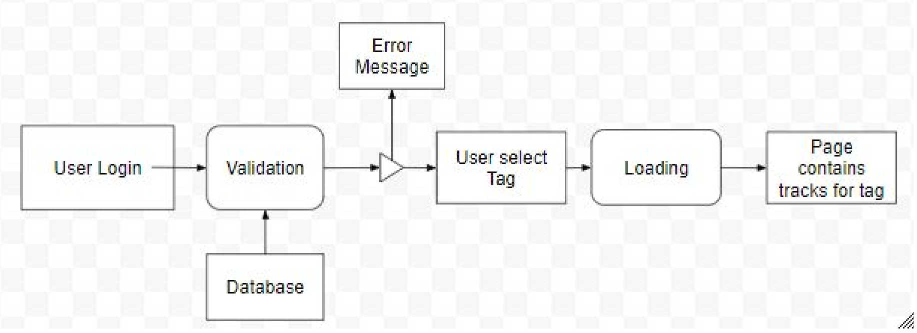
User can search tags.

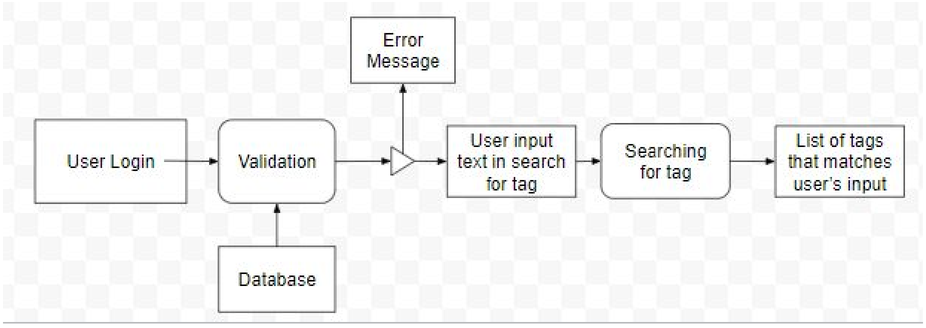
1. When the user opens the home page, he should be able to see a search box at the on the bottom right of the page.
2. The user should be able to enter a substring of a tag he is searching for.
3. The search results should show all tags containing the searched substring in a case insensitive matter from the whole database.

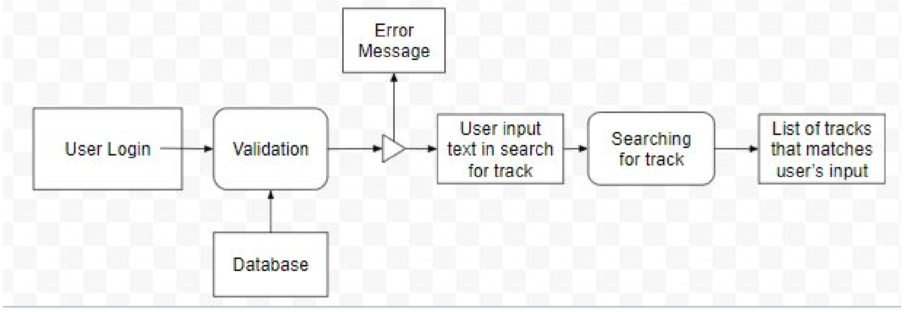
Use Cases in this Milestone

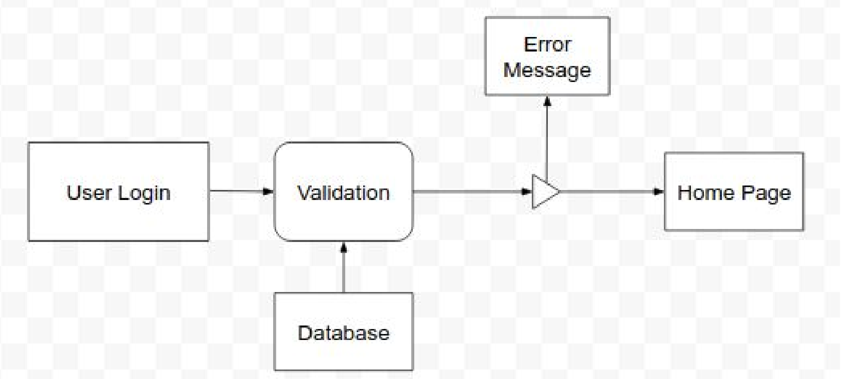


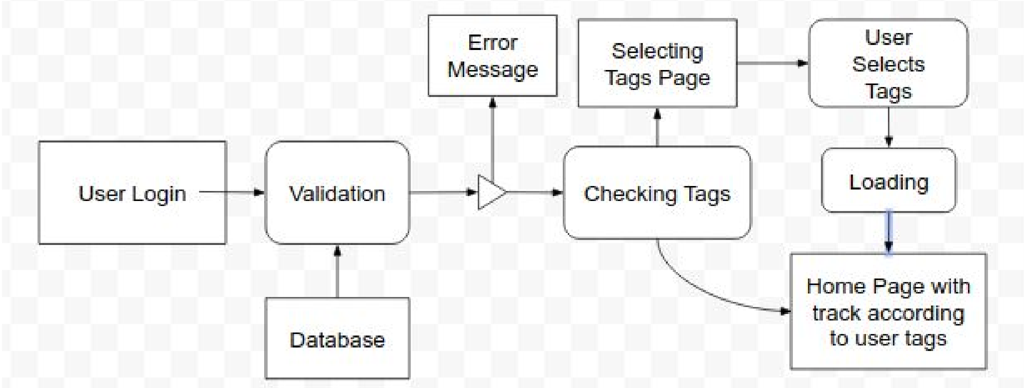
Activity Diagrams of Use Cases in this Milestone




Error Message
User Login
Validation
User input text in search for tag
Searching for tag
List of tags that matches user's input
Database


Error Message
User Login
Validation
User input text in search for track
Searching for track
List of tracks that matches user's input
Database





Test Cases

Task 1: User Change his selected prefered tags
Input: 1- Signed in user clicks open the change tags page. 2- User select new tags or remove selection from tags and then clicks save changes.

Tests:
1- Tests that the right view is returned when a signed in user opens it.
2- Tests that the change tags page doesn't open when user is not signed in.
3- Tests that the user tags are marked selected when the user opens the page.
3- Tests that user click save changes with no tags being selected.
4- Tests that user chooses some tags and then click save changes.
5- Tests that user selects new tags that were not selected, unselect some tags that were selected, keeps some selected tags selected and keep some unselected tags unselected.

Output:
Redirect to the same page again in case no tags are selected and redirecting to the home page in case the some tags were chosen with selected tags updated in the database.

Task 2: User can Like or unlike a track

Input:
1- Signed in user clicks open Home page or Liked tracks page or Favourite tracks page.

Tests:
1- Tests that the like button appears next to each track.
2- Tests that the button is activated for liked tracks and deactivated for unliked tracks.
3- Tests that the button is changed from active to inactive when clicked and vice versa.
4- Tests that when like button is active next to a track then this track appears in the Liked Tracks page and when like button is inactive next to a track then this track doesn't appear in the Liked Tracks page.

Output:
Liked tracks page containing liked tracks or not.

Task 3: Show Liked Tracks Correctly

Input:
1- Signed in user clicks open the Liked tracks page.

Tests:
1- Tests that the right view is returned when a signed in user opens it.
2- Tests that the page doesn't open when user is not signed in.
3- Tests that the liked tracks are shown correctly and same home page functionalities are working.

Output:
Liked Tracks Page with tracks correctly shown or ask user to login or sign up.

Task 4: User add or remove a track from favourites

Input:
1- Signed in user clicks open Home page or Liked tracks page or Favourite tracks page.

Tests:
1- Tests that the Add to favourites button appears next to each track.
2- Tests that the button is activated for favourite tracks and deactivated for non favourite tracks.
3- Tests that the button is changed from active to inactive when clicked and vice versa.
4- Tests that when Add to favourites button is active next to a track then this track appears in the Favourite Tracks page and when Add to favourites button is inactive next to a track then this track doesn't appear in the Favourite Tracks page.

Output:
Favourite tracks page containing Favourite tracks or not.

Task 5: Show Favourite Tracks Correctly

Input:
1- Signed in user clicks open the Favourite tracks page.

Tests:
1- Tests that the right view is returned when a signed in user opens it.
2- Tests that the page doesn't open when user is not signed in.
3- Tests that the favourite tracks are shown correctly and same home page functionalities are working.

Output:
Favourite Tracks Page with tracks correctly shown or ask user to login or sign up.

Task 6: User Edit Information

Input:
1- A String representing the email address of the user.
2- A String representing the current password of the user.
3- A String representing the first name of the user.
4- A String representing the last name of the user.
5- A String representing the new password of the user.
6- A String representing a confirmation for the new password of the user.

Tests:
1- Tests for input when user tries to edit information with valid input.
2- Tests for input when user tries to edit information with valid input with an invalid current password.

3- Tests for input when user tries to edit information with valid input but an email already registered in the database.
4- Tests for input when user tries to change password with valid input.
5- Tests for input when user tries to change password with unmatching wanted passwords.
6- Tests for input when user tries to change password with invalid current password.

Output:
Navigate to User Homepage in case of success or error message in case of failure.

Task 7: Search Tracks By Title

Input:
1- A String representing a substring of the title the user is searching for.

Tests:
1- Tests for input that doesn't match to any track.
2- Tests for input that exactly matches the title of on track.
3- Tests for input that matches a mid substring of a title of a track.
4- Tests for input that matches a prefix substring of a title of a track.
5- Tests for input that matches a suffix substring of a title of a track.
6- Tests for input that matches in a case insensitive way.

Output:
Return all tracks with titles that contains the substring searched by the user.

Task 8: Search Tags

Input:
1- A String representing a substring of the tag the user is searching for.

Tests:
1- Tests for input that doesn't match to any tag.
2- Tests for input that exactly matches the tag.
3- Tests for input that matches a mid substring of a tag.
4- Tests for input that matches a prefix substring of a tag.
5- Tests for input that matches a suffix substring of a tag.
6- Tests for input that matches in a case insensitive way.

Output:
Return all tags that contains the substring searched by the user.

Task 9: Select Tag page

Input:

1- click on certain tag.

Tests:

1- Tests for tag contains many tracks.
2- Tests for tag doesn’t has any track.
3- Tests for tag contains one track.

Output:

Return page of tracks that has this selected tag.