­­Spotify

TA: Radwa Moustafa

Team ID: 26

|  |  |  |
| --- | --- | --- |
| mainstream | 20191700716 | نورهان عبدالكريم خلف عبدالحافظ |
| mainstream | 20191700701 | نور محمد حسين محمود محمد |
| mainstream | 20191700408 | عمر خالد محمد جلال |
| mainstream | 20191700339 | عبدالرحمن سيد على محمد |
| mainstream | 20191700583 | محمد نورالدين عباس اسماعيل |
| mainstream | 20191700286 | سعيد عبد الناصر سعيد ابراهيم |

Software Requirement Specification

26 – Spotify

TA:Radwa Moustafa

|  |  |
| --- | --- |
| section | Name |
| 5 (sc) | نورهان عبدالكريم خلف عبدالحافظ |
| 5 (sc) | نور محمد حسين محمود محمد |
| 3 (sc) | عمر خالد محمد جلال |
| 3 (sc) | عبدالرحمن سيد على محمد |
| 4 (sc) | محمد نورالدين عباس اسماعيل |
| 3 (sc) | سعيد عبد الناصر سعيد ابراهيم |

1. **Introduction**

Music streaming platform that enables the system users to listen to Music online. In addition to giving them the freedom to create playlists of their favorite songs or search for a specific song. It also allows them to interact with other system users by following them. Moreover, the system enables artists to add songs or create albums on our system. Our goal is to build a comprehensive user-friendly system that provides the best music listening experience.

1. **Context Diagram**

External: listener, artists, database, bank

1. **User Requirements**
   * 1. Spotify shall allow listeners to create profile on the system.
     2. Spotify shall allow listeners to login to the system.
     3. Spotify shall allow listeners to listen to music.
     4. Spotify shall enable listeners to create a playlist.
     5. Spotify shall allow listeners to search for music.
     6. Spotify shall allow listeners to search for other system users.
     7. Spotify shall allow system users to follow each other.
     8. Spotify shall allow listeners to display their profile.
     9. Spotify shall allow listeners to browse music in home page.
     10. Spotify shall allow Artists to Add songs.
     11. Spotify shall allow Artists to create albums.
     12. Spotify shall allow Artists to add events.
2. **Functional Requirements**
   1. **Create profile**

* Description: Register to the system by entering personal information and then create new profile on the system.
* Input requirements:

Username: characters.

Password: set of characters and special characters.

Date of birth: date-time.

Account type: Boolean.

* Source: System user.
* Post-condition: directed to login page.
* Output: display message of success if entered data was in correct format or display message of failure if entered data was in incorrect format.

Add a new record of the new system user in our database.

* 1. **Login**
* Description: Login to the system with username and password.
* Input requirements:

Username: characters.

Password: set of characters and special characters not less than 12 characters.

* Source: System user.
* Post-condition: enter the system if entered data was a correct record.
* Output: display message of success/failure of the login.
  1. **listen to music.**
* Description: listen to music on the system.
* Input requirements:

Music selection: button click.

* Source: System user.
* Pre-conditions: user has already login.
* Post-condition: music streaming.
  1. **Create playlist.**
* Description: create a playlist of desired music by selecting which ones to add from the available ones.
* Input requirements:

Playlist name: characters.

Music selection: button click.

* Source: System user.
* Pre-conditions: user has already login.
* Post-condition: playlist creation. And it will be found in user’s profile.
* Output: insert playlist information into user’s record in the database.
  1. **Search for music.**
* Description: search for music using music name, category.
* Input requirements:

Search key: characters.

* Source: System user.
* Pre-conditions: user has already login.
* Output: display a list of song names.
  1. **Search for system user.**
* Description: search for another listener or artist by name.
* Input requirements:

Search key: characters.

* Source: System user.
* Pre-conditions: user has already login.
* Output: display a list of search results.
  1. **Follow other Spotify user.**
* Description: follow other Spotify user or artist.
* Input requirements:

Follow button: button click.

* Source: System user.
* Pre-conditions: user has already login. And searched for user’s name.
* Post- condition: user’s following list increase by 1.
* Output: update following information in user’s record in the database.
  1. **Display profile.**
* Description: display profile information on our system.
* Input requirements:

Display profile button on home: button click.

* Source: System user.
* Pre-conditions: user has already login.
* Output: display user profile.
  1. **Browse music.**
* Description: browsing various music on home page.
* Pre-conditions: user has already login.
* Output: display a list of all music on our system so that user can browse them.
  1. **Add song.**
* Description: adding a new song to be displayed on our system.
* Input requirements:

Add button: button click.

Song name: characters.

Song length: number.

* Source: artist.
* Pre-conditions: artist has already login.
* Post-condition: song is added on the system.
* Output: song is displayed on the list of music on the system. And song’s information is added in the artists’ record on the database of the system.
  1. **Create album.**
* Description: create a new album on the system and fill it with new songs.
* Input requirements:

Add button: button click.

album name: characters.

Song length: number.

Song number: numbers.

Song name: characters.

* Source: artist.
* Pre-conditions: artist has already login.
* Post-condition: album is added on the system.
* Output: album is displayed on the list of albums on the system. And album’s information is added in the artists’ record on the database of the system.
  1. **Add event.**
* Description: adding a new event to be displayed on our system.
* Input requirements:

Add button: button click.

Event date: date-time.

Event location: characters.

Artist information: characters and numbers.

* Source: artist and database.
* Pre-conditions: artist has already login.
* Post-condition: event is added on the system.
* Output: event is displayed on the home page of the system. And event’s information is added in the artists’ record on the database of the system.

1. **Non-Functional Requirements**

* **Ease of use**

System users shall be able to use all the system functions after 2 hours of training. After this training, the average number of errors made by experienced users shall not exceed two per hour of system use.

* **Speed**

System function shall respond in less than 2 seconds.

* **Size**

size of application won’t exceed 80000 **kb.**

* **Reliability**

System failure rate won’t exceed in 0.009%.

* **Portability**

System operates on various operating systems (windows, macOS, linux).