PRODUCT DESCRIPTION

# Product Title

ISongs – A music application

# Product Scope

# ISongs is an application that provides a list of songs around the world contains lyrics and sheet of songs allows users to search varieties of the songs and bookmark their favorite songs. The best highlight of this application is identifying music instantly by singing or humming. There are two kinds of users: member and VIP member. At a VIP member, they can download and listen the songs offline without advertisement interruptions and access sheet of songs that support them with playing musical instruments. For normal member, they have to wait an ad after finishing three songs in the sequent queue and cannot download their songs.

**KEY OBJECTIVES:**

**Guest:**

* Register
* Login (Email/Facebook/Phone)

**Member:**

* Login
* Logout
* Reset forgotten password
* Get notification when their favorite singers announce a new song and album
* Search the title the songs/lyrics/singers
* Identifying music instantly by singing or humming
* Bookmark their favorite songs and singers
* Create his own playlists
* Activate ‘Luminance’ pack for member to become a VIP member.

**VIP Member:**

* Login
* Logout
* Reset forgotten password
* Get notification when their favorite singers announce a new song and album
* Search the title the songs/lyrics/singers
* Identifying music instantly by singing or humming
* Bookmark their favorite songs and singers
* Create his own playlists
* Access the sheets of songs
* Listen to music offline
* Download the songs
* Listen the songs without advertisements

**Administrator:**

* Insert/Edit/Delete the songs included in lyrics and sheet of them and singers
* View member’s profile

# Stakeholders

## Customers

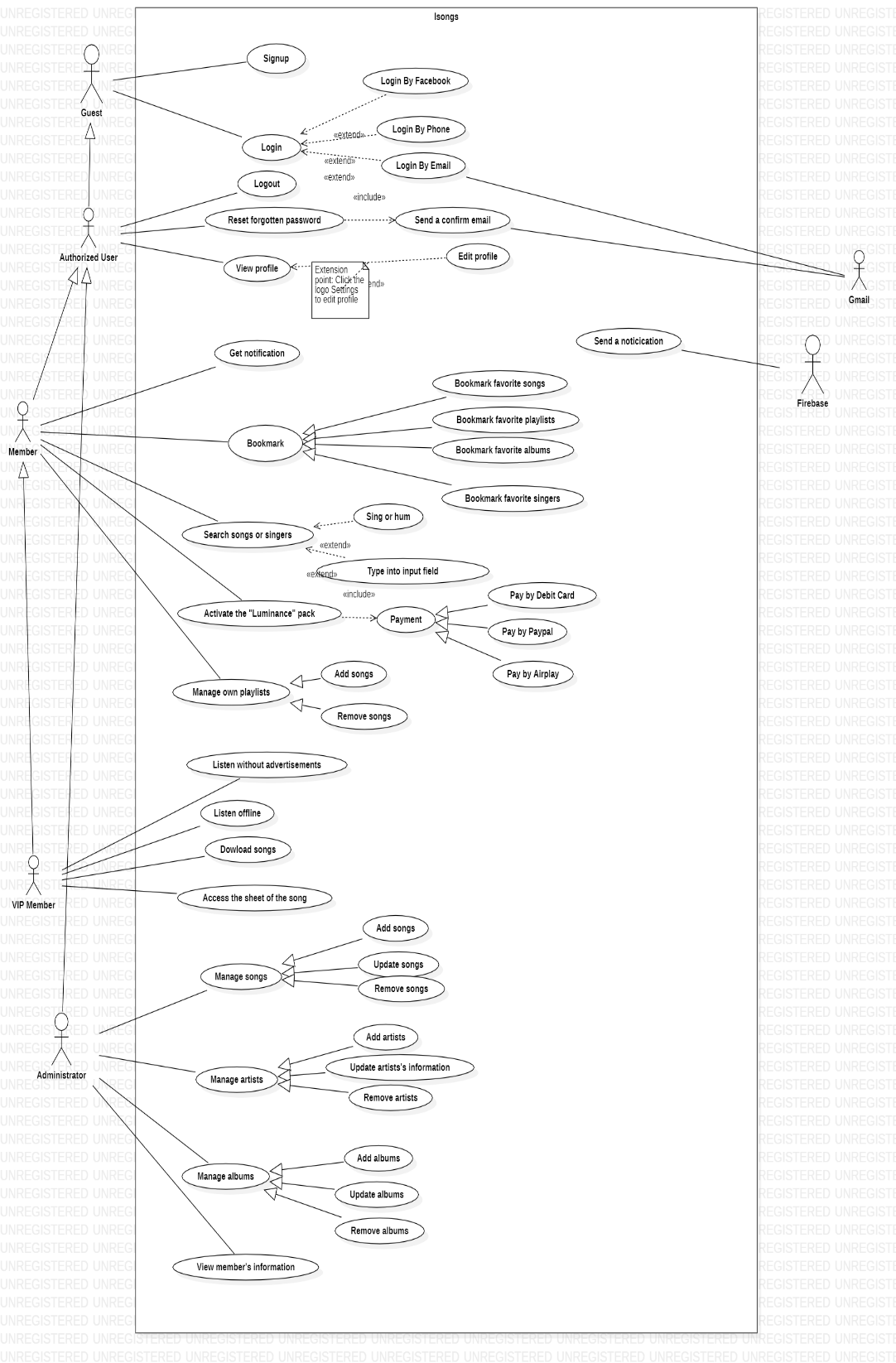
|  |  |  |
| --- | --- | --- |
| Customer Class | Expectation | Represented By |
| Producers | Want to advertise their songwriting to general public | Artists, music composers,.. |
| Media | Want to collaborate with their marketing | Tiktok, Facebook, Instagram |
|  |  |  |

## Users

|  |  |  |
| --- | --- | --- |
| User Class | Expectation | Represented By |
| People | Listen to the quality music and seek out their songs easily. Besides that, they can download and listen the songs offline whenever they want. | All of the generations |
|  |  |  |
|  |  |  |

# Functional Requirements

[*Describe the functional requirements, as black-box with following details:*

* + *Who are actors*
  + *Who will receive outputs*
  + *Which are input, output*
  + *Condition to start (optional)]*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Functional Requirement No. | Description | Actor | Who will receive outputs | Input / Output | Condition to start |
| FR 1 | Search song, album, singer, playlist | Member, VIP member | Member, VIP member | Input: query string  Output: list of result |  |
| FR 2 | Bookmark the favorite song, album, singer, playlist | Member, VIP member | Member, VIP member | Input: User click to bookmark button  Output: Favorite song, album, singer, playlist are added to member library | User is authenticated |
| FR 3 | Get notification when favorite singers release a new song or album | Member, VIP member | Member, VIP member | Input: User marks his/her favorite singers.  Output: System notices to the user when his/her favorite singer releases a new song, album | User is authenticated |
| FR 4 | Activate the “Luminance” pack for member to become VIP member | Member | Member | Input: Member register VIP account  Output: Member become VIP member | User is authenticated and not a VIP member |
| FR 5 | Access the sheet of the song | VIP member | VIP member | Input: Member click to the song’s sheet  Output: Display the song’s sheet | User is VIP member |
| FR 6 | Listen offline | VIP member | VIP member | Input: User choose the song in user his/her library  Output: The song is played | - User is VIP member  - The song was downloaded before |
| FR 7 | Download song | VIP member | VIP member | Input: User click to download button  Output: The song is added to user offline library | User is VIP member |
| FR 8 | Add, Update, Delete the song and related information | Administrator | All actors | Input: admin add, update or delete the song and its related information  Output: the song and its related information is added, updated or deleted | The administrator role is identified |
| FR 10 | Sign up | Guest | Guest | Input: user register new account  Output: new account is created | Account is not created before |
| FR 11 | Login | Member, VIP member, Administrator | Member, VIP member, Administrator | Input: user submit login form with his/her account  Output: return login result | Account is available |
| FR 12 | Reset password | Member, VIP member, Administrator | Member, VIP member, Administrator | Input: enter old password, new password  Output: password is changes | User is authenticated |
| FR 13 | Reset forgotten password | Member, VIP member | Member, VIP member | Input: user click to forgot password button  Output: system will contact will user to chance password | User account is available |
| FR 14 | Add/Remove own playlists | Member, VIP member | Member, VIP member | Input: type the song in the input field and click the button to add the playlist. If user want to remove, they click the button “Remove” in the playlist  Output: all the songs in the playlists are added or deleted. | User account is available |
| FR 15 | Identifying music instantly by singing or humming | Member, VIP member | Member, VIP member | Input: user let the app listen to the song’s melody  Output: song name and information |  |

# Non-functional Requirements

*[Describes testable NF requirements]*

## Targeted Platforms

TP1: Application development on we support on web app and mobile. About web app platforms, we use framework ReactJS for Front End and Java Spring Boot for Back End. In order for users to make use of multi-platforms (Android and IOS), we resolve frame work Flutter

## Performance

PR1: The interface is ready to the user in no more than 3 seconds from the time of action.

PR2: Minimum transmission speed between Client and Server is 500Kb/s.

PR3: Maximum response time of a transaction is 3 seconds.

PR4: Use API to retrieve data, up to 30 songs each time.

## Availability

AR1: The application is guaranteed to work all the time.

AR2: Application maintenance and upgrade maximum 1 time in 1 month, no more than 3 hours each time.

AR3: In the case of application development that needs to be linked with 3rd parties, the application ensures independent operation, not interrupted by 3rd party outages.

## Reliability

RR1: Error probability when using the application is less than 1/1000.

ER2: Song is moderated by Admin before posting to the application.

# Release Method

*[How product is delivered:*

* *One-shot (waterfall)*
* *Iterations/ incremental*
* *Agile/ Scrum]*

*The application is developed according to the* ***Agile*** *software development model.*

*The development process is divided into several sprints, each sprint completed in 2 weeks.*

*After each sprint, application will be released a version for review until the completion of the application.*

*Main Sprint:*

* *Sprint 1: Development interface and function for Guest role.*
* *Sprint 2: Development interface and function for Member role.*
* *Sprint 3: Development interface and function for VIP Member role.*
* *Sprint 4: Development interface and function for Admin role.*

**Group 6:**

Le Nhat Hanh Lan - SE140961

Doan Nhat Quang - SE140968

Nguyen Xuan Linh Tam - SE141130

Nguyen Thanh Nhan - SE140587