ANalysis document

Jordan Radushev – Semester 6

# Functional requirements

In the table below you will see the functional requirements of my project.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Name** | **Description** | **MoSCoW** |
| 1 | Authentication & Authorization | JWT & Role-based | Must |
| 2 | Standard login | Secure login by generating a JWT | Must |
| 3 | Register | Encrypt personal data such as password and payment details | Must |
| 4 | Login with Google, Apple, etc. | - | Could |
| 5 | Core playlist-related functionality | Creating, modifying, and displaying user’s playlists and their content | Must |
| 6 | Possible playlist-related functionality | Public/private playlists, like other users’ playlists | Should |
| 7 | Song searching | Full-text search of the available songs | Must |
| 8 | Song uploading | A separate role (Artist) should be able to upload songs in the platform | Could |
| 9 | Song streaming | Backend service, responsible for streaming music files over the network | Must |
| 10 | Offline mode | The user should be able to save a concrete song/playlist locally, so the user can listen to music offline | Should |
| 11 | Mock Payment functionality | The backend service should be able to track if the user has paid or not | Must |
| 12 | Real payment functionality | The user should be able to make payments for subscription securely | Could |

# User Scenario

The main scenario is when a user has created its own account to start searching for music, listen to hers/his favourite songs and she/he should be able to create playlists out of the available music in the platform. As the payment is secondary requirement it will be part of the main scenario in later phase of the project and then the user should have already paid hers/his subscription before using the platform with its account.

# NON-FUNCTIONAL Requirements

This section is about the non-functional requirements, regarding my project, which are:

* The system needs to be able to handle a huge number of users (e.g., 600 k simultaneous users).
* The system’s users are from all around the globe.
* The system must be available 24/7 uninterrupted.
* The application must not have huge loading time (no more than 3-4 sec.).
* The application must encrypt the sensitive data before storing it in the database.
* The application must be open for modifications and easy for maintenance and extensions.