<Українська музична платформа>

Software Requirements Specification

<1.0>

<3/04/2024>

<Климчук Анастасія, Лимар Вікторія, Соя Марʼяна, Юрків Христина, Вітрук Іван>

Lead Software Engineer

Prepared for

WSU-TC CptS 322—Software Engineering Principles I

Instructor: Дорош В.І.

Spring 2024

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 03.04.2024 | Версія 1 | Климчук Анастасія, Лимар Вікторія, Соя Марʼяна, Юрків Христина, Вітрук Іван | Перша ревізія |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Лимар Вікторія | Project Director | 3.04.2024 |
|  | Климчук Анастасія | Project Manager | 3.04.2024 |
|  | Дорош В.І. | Instructor | 3.04.2024 |

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

2. General Description 1

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

3. Specific Requirements 2

3.1 External Interface Requirements 3

3.1.1 User Interfaces 3

3.1.2 Hardware Interfaces 3

3.1.3 Software Interfaces 3

3.1.4 Communications Interfaces 3

3.2 Functional Requirements 3

3.2.1 <Functional Requirement or Feature #1> 3

3.2.2 <Functional Requirement or Feature #2> 3

3.3 Use Cases 3

3.3.1 Use Case #1 3

3.3.2 Use Case #2 3

3.4 Classes / Objects 3

3.4.1 <Class / Object #1> 3

3.4.2 <Class / Object #2> 3

3.5 Non-Functional Requirements 4

3.5.1 Performance 4

3.5.2 Reliability 4

3.5.3 Availability 4

3.5.4 Security 4

3.5.5 Maintainability 4

3.5.6 Portability 4

3.6 Inverse Requirements 4

3.7 Design Constraints 4

3.8 Logical Database Requirements 4

3.9 Other Requirements 4

4. Analysis Models 4

4.1 Sequence Diagrams 5

4.3 Data Flow Diagrams (DFD) 5

4.2 State-Transition Diagrams (STD) 5

5. Change Management Process 5

A. Appendices 5

A.1 Appendix 1 5

A.2 Appendix 2 5

# 1. Introduction

Проектна карта проекту "Музична платформа" документує та відстежує необхідну інформацію, необхідну для прийняття рішення приймальниками рішень щодо затвердження проекту на фінансування. Проектна карта повинна містити потреби, обсяг, обґрунтування, зобов'язання з ресурсами, а також рішення спонсора (спонсорів) про продовження або припинення проекту. Вона створюється під час фази Ініціювання проекту.

## 1.1 Purpose

Специфікація вимог до програмного забезпечення мобільного додатку "Українська музична платформа" для забезпечення виконання всіх бізнес-вимог та технічних завдань.

## 1.2 Scope

Документ описує вимоги до мобільного додатку, який буде використовуватись для потокової трансляції музики, створення плейлистів, перегляду біографій виконавців та інших функцій, що стосуються української музики.

## 1.3 Definitions, Acronyms, and Abbreviations

SRS - Специфікація вимог до програмного забезпечення,

UI - користувацький інтерфейс.

## 1.4 References

(1) (готовий) CDC\_UP\_Project\_Charter\_Template.

(2) (готовий) M101-project-vison-statement-template.

(3) sample-test-plan-template.

## 1.5 Overview

Документ складається з вступу, загального опису, конкретних вимог та додатків.

# 2. General Description

## 2.1 Product Perspective

"Українська музична платформа" буде інноваційним мобільним додатком, який надасть унікальну можливість широкому колу користувачів досліджувати та насолоджуватися українською музикою. Цей додаток заповнить прогалину на ринку, де бракує представлення української музики, і створить єдиний доступ до музичних творів українських виконавців.

## 2.2 Product Functions

Основні функції включатимуть управління акаунтом користувача, стрімінг музики, створення та управління плейлистами, інтеграцію з соціальними мережами, пошук за різними критеріями (наприклад, виконавцем, жанром), а також персоналізовані рекомендації.

## 2.3 User Characteristics

Цільова аудиторія включає любителів української музики всіх вікових категорій. Додаток буде легкий у використанні для недосвідчених користувачів, але з достатньою глибиною та функціональністю для досвідчених шанувальників музики.

## 2.4 General Constraints

Розробка ведеться з урахуванням бюджетних обмежень та обмеженого часу. Додаток має працювати на iOS та Android.

## 2.5 Assumptions and Dependencies

Проект робить припущення, що попит на спеціалізовану музичну платформу з українською музикою є значним. Він залежить від участі та вкладу українських музикантів та наявності достатнього обсягу контенту для забезпечення багатого користувацького досвіду.

Цей розділ повинен також враховувати організаційні та технічні наслідки створення такої платформи, у тому числі потребу у створенні ІТ-інфраструктури для розробки, тестування та хостингу додатку.

Також мають бути описані ризики, включно з недостатньою доступністю даних, низькою зацікавленістю користувачів та потенційними технічними проблемами розробки. Кожен ризик повинен мати стратегії пом’якшення, такі як розробка спільноти користувачів в додатку для збільшення залученості та використання перевірених технологій та методологій тестування для забезпечення якості продукту.

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

* Має бути інтуїтивно зрозумілим та привабливим, забезпечувати легке навігування.
* Повинен відображати інформацію про виконавців, треки, та плейлисти.

### 3.1.2 Hardware Interfaces

Сумісність з сучасними смартфонами, що працюють на Android та iOS.

### 3.1.3 Software Interfaces

* Інтеграція з сервісами для авторизації (наприклад, Google Accounts).
* Використання API для забезпечення стрімінгу музики.

### 3.1.4 Communications Interfaces

Використання HTTPS для безпечної передачі даних.

## 3.2 Functional Requirements

### 3.2.1 <Functional Requirement or Feature #1>

3.2.1.1 Introduction

Функціонал реєстрації користувачів дозволяє слухачам та виконавцям створювати особистий акаунт для доступу до повного спектру можливостей платформи.

3.2.1.2 Inputs

* Ім'я користувача
* Електронна пошта
* Пароль
* Підтвердження паролю
* Дата народження

3.2.1.3 Processing

* Перевірка унікальності електронної пошти
* Валідація введених даних
* Хешування пароля перед збереженням
* Генерація токена для підтвердження акаунту через електронну пошту

3.2.1.4 Outputs

* Підтвердження реєстрації користувача
* Електронний лист із посиланням для активації акаунту

3.2.1.5 Error Handling

* Якщо електронна пошта вже використовується, відображати повідомлення про помилку
* Якщо паролі не співпадають, запитати користувача перевірити введені дані

### 3.2. Функціональні вимоги (Functional Requirements):

### 3.2.2 Система пошуку (Search System)

3.2.2.1 Вступ (Introduction):

Система пошуку дозволяє користувачам ефективно знаходити музичні треки, виконавців, або плейлисти через інтерактивний інтерфейс.

3.2.2.2 Введення (Inputs):

* Пошуковий запит введений користувачем.
* Фільтри, такі як жанр, рейтинг, дата випуску.

3.2.2.3 Обробка (Processing):

* Відфільтрування бази даних за заданими критеріями.
* Ранжування результатів за релевантністю до запиту.

3.2.2.4 Виводи (Outputs):

* Список відфільтрованих та впорядкованих результатів, що відображається користувачу.

3.2.2.5 Обробка помилок (Error Handling):

* Повідомлення про помилку, якщо результати не можуть бути завантажені.
* Підказки користувачам щодо оптимізації запиту для кращих результатів.

## 3.3 Use Cases

### 3.3.1 Реєстрація виконавця (Artist Registration):

* Виконавець вводить свої дані для створення профілю.
* Система перевіряє та затверджує профіль виконавця.

### 3.3.2 Створення плейлисту користувачем (User Playlist Creation):

* Користувач обирає треки для створення нового плейлисту.
* Система зберігає плейлист у профілі користувача.

## 3.4 Classes / Objects

### 3.4.1 <Class / Object #1>

3.4.1.1 Attributes

* Ім'я користувача (Username)
* Електронна пошта (Email)
* Список плейлистів (Playlists)

3.4.1.2 Functions

* Реєстрація (Register)
* Вхід в систему (Login)
* Створення плейлисту (Create Playlist)

### 3.4.2 <Class / Object #2>

3.4.2.1 Атрибути (Attributes):

* Ім'я виконавця (Artist Name)
* Біографія (Biography)
* Дискографія (Discography)

3.4.2.2 Функції (Functions):

* Створення профілю виконавця (Create Artist Profile)
* Додавання музики (Add Music)
* Оновлення інформації профілю (Update Profile Information)
* Посилання на вимоги: 3.2.1, 3.3.1

## 3.5 Non-Functional Requirements

### 3.5.1 Performance

* Система має обробляти 95% транзакцій менше ніж за 2 секунди.
* Стрімінг музики має починатися не пізніше ніж за 3 секунди після вибору користувачем треку.

### 3.5.2 Reliability

* Система має мати час безвідмовної роботи (MTBF - Mean Time Between Failures) понад 30 днів.
* Всі критичні оновлення системи повинні проводитися з можливістю відновлення до попереднього стабільного стану (rollbacks).

### 3.5.3 Availability

* Система має бути доступною 99.9% часу, не враховуючи заплановане технічне обслуговування.
* Дозволено не більше 1 хвилини незапланованого простою на день.

### 3.5.4 Security

* Усі користувацькі дані мають бути зашифровані відповідно до стандарту.
* Система автентифікації має підтримувати двофакторну автентифікацію.

### 3.5.5 Maintainability

* Код додатку має бути написаний таким чином, щоб його можна було легко підтримувати та оновлювати.
* Документація коду має бути повною та актуальною.

### 3.5.6 Portability

* Додаток має бути крос-платформним і підтримувати останні дві версії операційних систем iOS та Android.
* Компоненти додатку мають бути модульними, щоб спростити міграцію на нові платформи у майбутньому.

## Inverse Requirements

1. **Система не повинна надавати доступ до редагування чужих профілів**. Це захищає приватність інформації користувачів.
2. **Додаток не повинен дозволяти автоматичне відтворення музики при запуску.** Запобігає небажаному споживанню даних користувача.

## 3.7 Design Constraints

1. **Система повинна бути розроблена з урахуванням обмеження ресурсів мобільних пристроїв.** Мінімізація споживання батареї і використання даних.

## 3.8 Logical Database Requirements

1. Використання реляційної бази даних для зберігання користувацьких даних, інформації про виконавців та треки.
2. Формат даних має відповідати JSON для інтеграції з API.
3. Забезпечення інтегритетності даних через реляційні зв'язки та обмеження цілісності.

## 3.9 Other Requirements

1. **Підтримка багатомовності:** Система повинна підтримувати українську мову як основну з можливістю додавання інших мов.
2. **Інтеграція з соціальними медіа:** Забезпечити можливість публікації плейлистів в соціальних мережах.
3. **Адаптивність до різних мережевих умов:** Додаток повинен бути ефективним навіть при повільному інтернет-з'єднанні або у роумінгу.

# 4. Change Management Process

Для забезпечення актуальності SRS у відповідності до змін в проекті або вимогах, наступний процес буде використовуватись:

1. Ініціація зміни:

Будь-який член проектної команди може ініціювати зміну, подавши формальний запит на зміну (Change Request - CR) через проектну систему управління.

1. Оцінка зміни та ухвалення рішення:

Project Manager і Project Director оцінюють вплив запропонованої зміни на проект.

1. Реалізація зміни:

Після затвердження, проектна команда вносить зміни в SRS.

1. Сповіщення та документування:

Всі зміни документуються, а стейкхолдери сповіщаються про оновлення SRS.

1. Версіонування SRS:

Оновлений SRS отримує новий номер версії та дату релізу.