**Podcast hosting and listening platform**

**Software Requirements Specification**

**INT-219**

**Front End Web Developer**

**Kamakshi Aggarwal- 12306189**

**Vedika -12326838**

**Vivek Kumar- 12305176**

**Shivam Kumar- 12309992**

Prepared for

Continuous Assessment 3

Spring 2025

**Table of Contents**

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 2

1.4 References 2

1.5 Overview 2

2. General Description 2

2.1 Product Perspective 2

2.2 Product Functions 3

2.3 User Characteristics 3

2.4 General Constraints 3

2.5 Assumptions and Dependencies 3

3. Specific Requirements 4

3.1 External Interface Requirements 4

3.1.1 User Interfaces 4

3.1.2 Hardware Interfaces 4

3.1.3 Software Interfaces 4

3.1.4 Communications Interfaces 4

3.2 Functional Requirements 5

3.2.1 <Functional Requirement or Feature #1> 5

3.2.2 <Functional Requirement or Feature #2> 5

3.5 Non-Functional Requirements 5

3.5.1 Performance 5

3.5.2 Reliability 5

3.5.3 Availability 6

3.5.4 Security 6

3.5.5 Maintainability 6

3.5.6 Portability 6

3.7 Design Constraints 7

3.9 Other Requirements 7

4. Analysis Models 7

4.1 Data Flow Diagrams (DFD) 7

4.2 UML Diagrams 9

5. Github link…………………………………………………………………………………………………...10

A. Appendices 11

A.1 Appendix 1 11

# 1. Introduction

## *The introduction to the Software Requirement Specification (SRS) document provides a complete overview of the Podcast Hosting and Listening Platform, explaining purpose, scope, and context of podcast creation and consumption.*

## 1.1 Purpose

*The purpose of this Software Requirements Specification (SRS) is to provide the functionality of the Podcast Hosting and Listening Platform including design constraints, and interactions. This document defines a reference for developers, testers, and project stakeholders to comprehend the scope and implementation of the web application. It makes sure that all components of the project are well-structured and documented and approved by all parties involved.*

## 1.2 Scope

## *The Podcast Hosting and Listening Platform is a web application built using PHP and SQL. It provides a centralized space for podcast creators to upload, schedule, manage, and analyse their content and also allows users to explore and listen to podcasts uploaded by various artists.*

*(1) The software product to be produced is a web application named the Podcast Hosting and Listening Platform.*

*(2) The platform's scope consists of the following key features:*

* ***User Authentication:*** *management of user accounts for not just podcast creators but also for listeners, including registration, login, and password recovery to ensure security.*
* ***Podcast Management:*** *provides features for creators to upload, schedule, and manage their podcasts and episodes, including metadata management, and format support.*
* ***Podcast Exploration:*** *allows listeners to explore latest podcasts by searching and categorization.*
* ***Analytics:*** *Provides detailed analytics of podcasts uploaded by artists, their insights into listener engagement, like count, and podcast performance statistics.*
* ***Social Impact Integration:*** *to connect podcasts with community projects and social causes, ensuring social responsibility and engagement.*

*The platform's scope excludes:*

* *Advanced audio editing and production capabilities within the platform.*
* *Direct monetization or payment processing features for creators.*
* *Native mobile applications for iOS and Android.*

*(3) The application of this software aims to create a centralized and efficient platform for podcast creation and use. It aims to:*

* *Encourage creators with necessary features they need to upload, manage, schedule and share their content successfully.*
* *Ensures that listeners have enjoyable experience to explore, listen, and engage with podcasts.*
* *Allows a various community around podcasts to connect with creators and listeners that promotes engagement.*

## 1.3 Definitions, Acronyms, and Abbreviations

*The following section lists all terms, acronyms, and abbreviations used in this SRS to ensure clarity and consistency.*

* *SRS: Software Requirements Specification*
* *UI: User Interface*
* *UX: User Experience*
* *API: Application Programming Interface*
* *SQL: Structured Query Language*
* *PHP: Hypertext Preprocessor*
* *HTTP: Hypertext Transfer Protocol*
* *HTTPS: Hypertext Transfer Protocol Secure*

## 1.4 References

* *IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)*
* *PHP Documentation –* [*https://www.php.net*](https://www.php.net)
* *MySQL Documentation –* [*https://dev.mysql.com/doc/*](https://dev.mysql.com/doc/)

## 1.5 Overview

# *This SRS document delivers a complete breakdown of the system’s requirements, design environment, functionalities, and user interactions. It is organized as follows:*

* *Section 2, "General Description," provides a general overview of the project like its functions, user characteristics, constraints, and assumptions.*
* *Section 3, "Specific Requirements," delivers the specific functional and non-functional requirements that the software must adhere to strictly.*
* *Section 4, "Analysis Models," consists of the analysis models, like Data Flow Diagrams, used to show the system's requirements.*
* *The Appendices provide additional information to support the requirements.*

# 2. General Description

*This section of the SRS describes the general aspects that affect the product and its requirements.*

## 2.1 Product Perspective

## *The Podcast Hosting and Listening Platform is a web application that will be built as a standalone product. It networks with users (creators and listeners) and the database (SQL) but does not directly interact with other external systems except for potential social media integration for sharing.*

## 2.2 Product Functions

The software will provide the following functions:

* User authentication for security and authenticity (registration and login)
* Podcast management (create, edit, delete podcasts and episodes)
* Explore Podcast (search, filter, recommendations)
* Detailed Analytics (track plays, likes, engagement)
* Social impact (connect podcasts with social causes)

## 2.3 User Characteristics

## *Podcast Creators: Users who want to upload, manage, and schedule podcasts. They require the tools to manage content, track performance, and engage with listeners.*

## *Listeners: Users who explore and listen to podcasts uploaded by creators. They need to be able to search for content based on various categories, engage with podcasts by liking the content.*

## 2.4 General Constraints

## *Developed using PHP and SQL*

## *Requires a web server (XAMPP)*

## *Must be responsive*

## *Compatible with modern browsers*

## 2.5 Assumptions and Dependencies

* Users should have internet access
* It is assumed that for PHP and SQL is supported, the server infrastructure will be available and configured.
* It is also assumed that third-party APIs for social media integration will be available.

# 3. Specific Requirements

**3.1 External Interface Requirements**

**3.1.1 User Interfaces**

* A responsive and user friendly platform with navigation menus for creators and listeners.
* Login/Sign up page with input validation.
* Home page for podcast creators to manage episodes.
* Explore page with search through categories.
* Analytics page with charts/graphs to show user engagement.
* Social impact section with linked campaigns.

### 3.1.2 Hardware Interfaces

* Server hosting the application must have at least:
* 4 GB RAM
* 2-core CPU
* PHP & MySQL support
* End-user devices require:
* A modern browser (Chrome, Firefox, Safari)

### 3.1.3 Software Interfaces

* PHP for server-side scripting.
* MySQL for database storage.
* HTML/tailwind CSS/JavaScript for frontend.
* Optional integration with social media APIs.

### 3.1.4 Communications Interfaces

* HTTP/HTTPS protocols for all communications.
* Secure session handling using PHP session management.

## 3.2 Functional Requirements

### *This section provides the specific features of the Podcast Hosting and Listening Platform. It offers the system's behaviour and the tasks that users will be able to perform.*

### 3.2.1 User Authentication

3.2.1.1 Introduction

This feature ensures user sign up, login, and account recovery processes, for both podcast creators and listeners to ensure secure and authentic access to the platform. It includes all functionalities related to user identity management.

3.2.1.2 Inputs

* Registration:
* Username (unique identifier)
* Email address (for verification and communication)
* Password (minimum 8 characters, including a mix of uppercase and lowercase letters, numbers, and special characters)
* Optional: Profile picture, display name
* Login:
  + Username or Email address
  + Password

3.2.1.3 Processing

* Registration:
  + Validation of user input for format, authenticity, and complexity.
  + Store the user's information (including password and verification status) in the database.
  + Upon successful email verification, mark the user's account as active.
* Login:
  + Fetch the user's stored credentials (username/email) from the database.
  + If the credentials are valid and the account is active, secure session is created for the user.
  + Session information (user ID, login time) is stored.

3.2.1.4 Outputs

* Registration:
  + Successful registration:
  + Failed registration:
    - In case of failure, an error message is displayed (e.g., invalid input, username already exists).
* Login:
  + Successful login:
    - user is directed to their dashboard.
  + Failed login:
    - An error message is displayed (e.g., "Invalid credentials").

3.2.1.5 Error Handling

Invalid credentials, existing usernames, SQL errors.

### 3.2.2 Podcast Management

3.2.2.1 Introduction

This feature allows artists to manage their podcasts and episodes.

3.2.2.2 Inputs

Metadata of podcast(title, caption, category), episode details (title, description, audio file), audio or video files.

3.2.2.3 Processing

* Edit Podcast: Update metadata of episode in database.
* Delete Podcast: deletes podcast by the artist.
* Upload Episode: Store episode information and file path.

3.2.2.4 Outputs

* Successful operations: Confirmation messages.
* Podcast/episode listings.

3.2.2.5 Error Handling

## Invalid input: Display error messages.

## File upload errors: Handle file size limits and issues related to format type.

**3.2.3 Podcast Exploration**

3.2.3.1 Introduction: This feature allows users to explore and search podcasts.

3.2.3.2 Inputs: Search, filter through (category, keywords).

3.2.3.3 Processing:

* Search: Fetch podcasts.
* Filter: podcasts based on categories.

3.2.3.4 Outputs:

* List of podcasts matching the search/filter criteria.
* Podcast details.

3.2.3.5 Error Handling:

* No results found: Display appropriate message.

**3.2.4 Analytics**

3.2.4.1 Introduction: This feature provides creators with insights into their podcast performance.

3.2.4.2 Inputs: User interactions (number of plays, likes), listener’s location.

3.2.4.3 Processing:

* Track plays: Counts the number of times an episode is watched.
* Track likes: Counts the number of likes for an episode.
* Calculate engagement statistics.
* Generate reports and visualizations.

3.2.4.4 Outputs:

* Analytics dashboards with statistics on plays, likes, listener demographics, etc.

3.2.4.5 Error Handling:

* Data unavailable: Displays error message.

**3.2.5 Social Impact Integration**

3.2.5.1 Introduction: This feature allows creators to connect their podcasts with social causes.

3.2.5.2 Inputs: Information about community projects/social causes.

3.2.5.3 Processing:

* Allow creators to associate their podcasts with specific causes.
* Display information about associated causes to listeners.
* Potentially provide tools for creators to promote or raise awareness for these causes.

3.2.5.4 Outputs:

* Display of social cause information on podcast pages.
* Tools for creators to engage with social initiatives.

3.2.5.5 Error Handling:

* Invalid input: Display error messages.
* API errors (if integrating with external platforms): Handle API connection and data retrieval issues.

## 3.5 Non-Functional Requirements

### 3.5.1 Performance

* The platform should respond to user quickly.
* The platform should be able to handle atleast 100 concurrent users without significant performance degradation.

### 3.5.2 Reliability

* The platform should have an uptime of 99%.
* The platform should handle errors gracefully and provide suitable error messages.

### 3.5.3 Availability

* The platform should be available 24/7, except for scheduled maintenance.

### 3.5.4 Security

* User passwords should be stored securely.
* The platform should be protected against common web vulnerabilities (e.g., SQL injection).
* User data should be protected in accordance with privacy regulations.

### 3.5.5 Maintainability

* The codebase should be well-organized and documented to facilitate future modifications and updates.

### 3.5.6 Portability

* The platform should be compatible with major web browsers (Chrome, Firefox, Safari, Edge).

## 3.7 Design Constraints

## *PHP must be used as the backend language.*

## *SQL database must store all persistent data.*

## *Web application must be deployable on XAMPP/WAMP stack.*

## 3.9 Other Requirements

* All episodes must be less than 10MB in size.

# 

# 4. Analysis Models

## 4.1 Data Flow Diagrams (DFD)

**Level 0**

A yellow circle with black text

AI-generated content may be incorrect.

**Level 1**

A diagram of a podcast

AI-generated content may be incorrect.

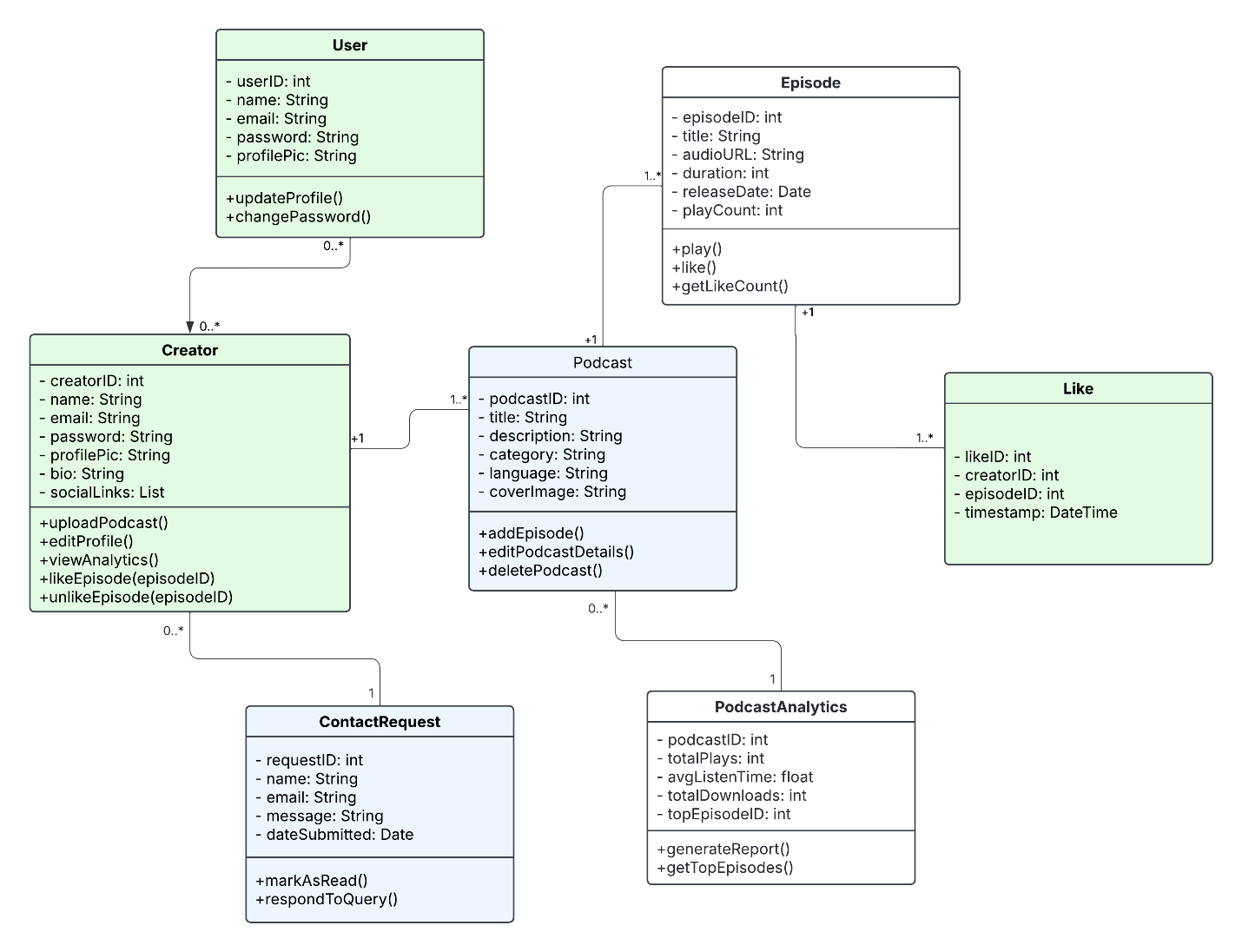
**Level 2**

A diagram of a podcast

AI-generated content may be incorrect.

## 

## 4.2 UML Diagram



# A. Appendices

## A.1 Appendix

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.