MSA SRS Document
Concert Booking System

Maulik Masrani, B037
Nikita Melam, B040
Prajakta Mungekar, B041

Introduction

Purpose:
The purpose of this document is to provide a comprehensive overview of the requirements for the development of a microservices-based system for booking tickets for concerts. It outlines the functional and non-functional requirements, as well as the constraints and specifications necessary for the successful implementation of the system.

Scope:
The system will allow users to log in, view the list of upcoming concerts, and book tickets for the chosen event. Upon booking, the system will update payment details in the payments database and the desired number of tickets for that event will be reserved for the user. This SRS focuses on the core functionality of the ticket booking system and secondary features such as user management or concert management are to be handled separately by the administrators.

Overview:
The ticket booking system will be implemented using microservices architecture, providing scalability, flexibility, and modularity. Users will interact with the system through a web interface, accessing concert information and making bookings securely. The system will ensure smooth payment processing and database management.

General Description

The ticket booking system will consist of the following modules:

1. **User Authentication**: Users will log in using their credentials (ID and password) to access the system. Error Handling has also been catered to, if in case a user enters an incorrect password or User ID. Thus, this module brings security to the system.

2. **Concert Listing**: Upon successful login, users will be presented with a list of details pertaining to available upcoming concerts. Details such as singer name, concert ID, zones, price per ticket and available number of tickets are fetched from the database and displayed.

3. **Ticket Booking**: Users will select a concert from the list and book tickets for the desired event. Once seats for the user are reserved, the new number of available tickets are accordingly updated in the concerts database.

4. **Payment Processin**g: Upon booking, the system will update payment details in the payments database. Requisite details like payment ID, time of payment, amount paid, user ID of payer are added to the database, providing for effective logging of payments and keeping a track record of all the transactions.

Functional Requirements

User Authentication:
Users must provide valid credentials (ID and password) to log in.
These credentials must be passed by the user through the URL to the system.

Concert Listing:
The system must fetch and display a list of upcoming concerts.
Concert information should include details such as date, time, venue, and available tickets. Upon seeing the concert details, the user decides which concert they wish to book tickets for.

Ticket Booking:
Users must specify the number of tickets they wish to book in the URL.
Upon booking, the corresponding number of seats are reserved for the user.

Payment Processing:
Payment date and time are noted when payment is done.
Payment details should be stored securely in the payments database.

Design Constraints

Microservices Architecture:
The system must be implemented using microservices architecture for modularity and scalability.
Each microservice should have a single responsibility and communicate with others via well-defined APIs.

Database Management:
Each microservice must have their own database and the datatypes must correlate between the database and microservices.

Non-Functional Attributes

Authentication

Our application provides the authentication feature by comparing the user IDs and passwords with the ones stored in the authentication database.

ER Diagram