1. **Software Requirement Specification**

**Table of Contents**

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

2. Overall Description

2.1 Product Perspective

2.2 Product Features

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 User Documentation

2.7 Assumptions and Dependencies

3.Specific Requirements

3.1 Functional Requirements

3.1.1 User Registration and Authentication

3.1.2 Artist Profile Creation and Management

3.1.3 Browse and Search Artists

3.1.4 Artist Booking

3.1.5 User Booking History and Tracking

3.1.6 Online Talent Hosting for Artists

3.2 Non-Functional Requirements

3.2.1 Performance

3.2.2 Security and Privacy

3.2.3 Usability and User Experience

3.2.4 Availability

3.2.5 Technology Stack

3.3 System Interfaces

3.3.1 Frontend Interface (React)

3.3.2 Backend Interface (Spring Boot)

3.3.3 Database Interface (MySQL)

3.4 Database Requirements

3.4.1 Data Entities and Relationships

3.4.2 Data Storage and Retrieval

3.5 Design Constraints

3.6 Software System Attributes

3.6.1 Reliability

3.6.2 Maintainability

3.6.3 Scalability

4. Appendices

4.1 Glossary

4.2 ER Diagrams

1. **Introduction**
   1. **Purpose**

The purpose of the Talent Hunt web-based application is to provide a platform for users to discover and book artists for various events and venues, while also allowing artists to showcase their talents online. This document outlines the software requirements necessary to develop the Talent Hunt application.

* 1. **Scope**

The Talent Hunt application will enable users to browse, search, and book artists for events including restaurants, pubs, lounges, bistros, and private parties. Artists can create and manage their profiles, showcase their talents, and interact with users. The system will utilize a 3-tier

architecture with Spring Boot as the backend framework, MySQL as the database, and React as the frontend framework.

* 1. **Definitions, Acronyms, and Abbreviations**

- SRS: Software Requirement Specification

- UI: User Interface

- API: Application Programming Interface

- MySQL: A relational database management system

- Spring Boot: A Java-based framework for building backend applications

-React: A JavaScript library for building user interfaces

* 1. **References**

Here is a list of references that might be relevant for the Talent Hunt project:

1. Zhang, J., & Zhang, L. (2020). An Online Booking System for Arts Performances in Rural China. In Proceedings of the 2nd International Conference on Computer Science and Application Engineering (pp. 33-37).

2. Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.

3. Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.

4. Ambler, S. W. (2002). Agile Modeling: Effective Practices for Extreme Programming and the Unified Process. John Wiley & Sons.

5. Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd ed.). Pearson.

6. Booch, G., Rumbaugh, J., & Jacobson, I. (2005). The Unified Modeling Language User Guide (2nd ed.). Addison-Wesley.

7. Oracle Corporation. (n.d.). MySQL. Retrieved from https://www.mysql.com/

8. Spring Boot Documentation. (n.d.). Retrieved from https://spring.io/projects/spring-boot

9. React Documentation. (n.d.). Retrieved from https://reactjs.org/docs/getting-started.html

10. Rozanski, N., & Woods, E. (2011). Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives. Addison-Wesley.

11. Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice (3rd ed.). Addison-Wesley.

12. Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.

13. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.

14. Brown, A. W., & Wallnau, K. C. (2013). Antipatterns: Identification, Refactoring, and Management. CRC Press.

15. Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.

16. Gamma, E. (1995). The State Pattern. Design Patterns: Elements of Reusable Object-Oriented Software.

17. Osterwalder, A., & Pigneur, Y. (2010). Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons.

18. Cusumano, M. A. (2010). Staying Power: Six Enduring Principles for Managing Strategy and Innovation in an Uncertain World. Oxford University Press.

These references cover a range of topics, including software engineering practices, object-oriented analysis and design, technology frameworks (MySQL, Spring Boot, React), design patterns, architecture principles, and business modeling. They can provide valuable insights and knowledge for various aspects of the Talent Hunt project's development and implementation.

* 1. **Overview**

Offering a high-level view of the application's functionalities, user classes, and operating environment. It outlines the user journey, from registration to talent booking, and emphasizes the significance of the platform for both users and artists.

1. **Overall Description**
   1. **Product Perspective**

The Talent Hunt application exists within the context of the broader entertainment and event management ecosystem. It serves as a digital platform that bridges the gap between artists seeking opportunities to showcase their talents and users looking to book artists for their events.

The application acts as a facilitator, connecting artists and users in a seamless manner.

* + 1. **System Interfaces**

The Talent Hunt application interfaces with the following components:

- Front-end (React UI): This is the user-facing interface where users and artists interact with the application. It provides a visually appealing and user-friendly experience for browsing artists, viewing profiles, making bookings, and managing accounts.

- Backend (Spring Boot): The backend serves as the control center of the application. It handles user authentication, artist profile management, booking requests, and interactions with the database. The backend exposes APIs that the frontend interacts with.

- Database (MySQL): The database stores user information, artist profiles, and booking data. It maintains the integrity of the application's data and supports efficient storage and retrieval.

* + 1. **Interaction with External Systems**

The Talent Hunt application may interact with external systems to enhance its functionality and user experience:

- Social Media Platforms: The application could provide the option for users and artists to sign up or log in using their social media accounts, such as Facebook or Google. This integration streamlines the registration process and increases user convenience.

- Payment Gateways: For processing payments related to artist bookings, the application might integrate with third-party payment gateways to securely handle financial transactions.

- Notifications and Communication: The application could utilize email or messaging services to notify users and artists about booking confirmations, updates, or other relevant communications.

* + 1. **Relationship with Existing Systems**

The Talent Hunt application complements and enhances the current event management landscape by offering an efficient and digital solution for artist bookings. Traditional methods of physically finding and booking artists are time-consuming and often lack transparency. The application streamlines this process by providing a centralized platform where users can explore artists' profiles, review their talents, and make bookings based on their preferences and requirements.

* + 1. **Impact on Stakeholders**

- Users: Users benefit from the convenience of finding and booking artists for their events without the need for extensive research or physical visits. They have access to a variety of talented artists, making their event planning experience smoother.

- Artists: Artists gain a broader reach and exposure by showcasing their talents on a digital platform. They can connect with potential clients and receive booking requests, expanding their opportunities for performances.

- Event Venues (Restaurants, Pubs, etc.): Event venues can utilize the application to discoverand book artists for entertainment at their establishments, enhancing the overall custome**r** experience.

- Application Administrators: Administrators manage the application, ensuring smooth operation, monitoring user activities, and addressing any issues that arise.

* + 1. **Future Enhancements**

In the future, the Talent Hunt application could evolve to include features such as:

- Enhanced artist profiles with reviews and ratings from previous bookings.

- Integration with location-based services to suggest artists available in specific regions.

- Collaboration tools for artists to collaborate on joint performances.

- Integration with streaming services to enable live online performances.

- Analytics and insights for artists to track their popularity and bookings.

* 1. **Product Features**

The Talent Hunt application will include the following key features:

- User and Artist Registration

- Artist Profile Creation and Management

- Browse and Search Artists

- Artist Booking for Events

- User Booking History and Tracking

- Online Talent Hosting for Artists

* 1. **User Classes and Characteristics**

- User: Individuals or organizations looking to book artists for events.

- Artist: Individuals with talents to showcase, available for bookings.

* 1. **Operating Environment**

The application will be accessible through modern web browsers on various devices, such as desktops, laptops, tablets, and smartphones.

* 1. **Design and Implementation Constraints**

- Utilize the 3-tier architecture: Frontend (React), Backend (Spring Boot), Database (MySQL).

- Ensure responsive UI design for cross-device compatibility.

* 1. **User Documentation**

**Table of Contents:**

1. Introduction

- About Talent Hunt

- System Overview

2. Getting Started

- User Registration

- User Login

- Forgot Password Recovery

3. Exploring Artists

- Artist Search and Filters

- Viewing Artist Profiles

- Online View of Artist Rates

4. Booking an Artist

- Initiating a Booking Request

- Negotiating Terms

- Making Online Payments

5. Managing Bookings

- Booking History

- Communication with Artists

- Rating and Feedback

6. Artist Features

- Artist Profile Creation

- Managing Availability

- Hosting Talent Online

7. Notifications

- Managing Notifications

8. Admin Controls

- Admin Verification

- Dispute Resolution

* 1. **Assumptions and Dependencies**

- Users have basic knowledge of using web applications.

-Artists have the necessary skills to create and manage profiles.

- Reliable internet connection is assumed for both users and artists.

1. **Specific Requirements**
   1. **Functional Requirements**
      1. **User Registration and Authentication**

- Users can register using email or social media accounts.

- Users can log in securely with their credentials.

* + 1. **Artist Profile Creation and Management**

- Artists can create and edit profiles with personal details and talents.

- Artists can upload multimedia content showcasing their talents.

* + 1. **Browse and Search Artists**

- Users can browse and filter artists based on various criteria (genre, location, availability, etc.).

* + 1. **Artist Booking**

- Users can view artist profiles and select suitable artists for their events.

- Users can send booking requests to artists.

* + 1. **User Booking History and Tracking**

- Users can view their booking history, including past and upcoming events.

- Users can track the status of their booking requests.

**3.1.6 Online Talent Hosting for Artists**

- Artists can upload videos, audio clips, and images to showcase their talents.

- Artists can manage and organize their uploaded content.

* 1. **Non-Functional Requirements**
     1. **Performance**

- The application should respond within 2 seconds for most user interactions.

- Concurrent user handling: 1000 simultaneous users.

* + 1. **Security and Privacy**

- Secure authentication and authorization mechanisms.

-Data encryption for sensitive user information.

- Compliance with data protection regulations.

* + 1. **Usability and User Experience**

- Intuitive and user-friendly UI design.

- Consistent branding and visual elements.

- Responsive design for various screen sizes.

* + 1. **Availability**

- Target availability: 99.9% uptime.

- Regular maintenance and updates without major disruptions.

* + 1. **Technology Stack**

- Frontend: React with responsive UI design.

- Backend: Spring Boot for API development.

- Database: MySQL for data storage and retrieval.

* 1. **System Interfaces**
     1. **Frontend Interface (React)**

- UI components for user registration, artist profile creation, browsing, booking, and user account management.

- Integration with backend APIs for data exchange.

* + 1. **Backend Interface (Spring Boot)**

- APIs for user and artist registration, profile management, browsing, booking, and tracking.

- Integration with frontend and database components.

* + 1. **Database Interface (MySQL)**

- Storage and retrieval of user, artist, and booking-related data

* 1. **Database Requirements**
     1. **Data Entities and Relationships**

- User (ID, username, email, password)

- Artist (ID, username, email, password, talents, media)

- Booking (ID, user ID, artist ID, event details)

* + 1. **Data Storage and Retrieval**

- Efficient data storage and retrieval for user profiles, artist profiles, and booking information.

* 1. **Design Constraints**

- Utilize the 3-tier architecture with separation of concerns.

- Implement responsive design principles for optimal user experience on various devices.

* 1. **Software System Attributes**
     1. **Reliability**

- Minimize downtime through robust coding and regular maintenance.

* + 1. **Maintainability**

- Use of modular code to facilitate future enhancements and bug fixes.

* + 1. **Scalability**

- Design for horizontal scalability to accommodate increasing user load.

1. **Appendices**
   1. **Glossary**

- Talent Hunt Application: The web-based platform that facilitates the booking of artists for various events and allows artists to showcase their talents online.

- User: An individual or entity that uses the Talent Hunt application to discover, browse, and book artists for events.

- Artist: An individual with specific talents, such as musicians, singers, dancers, etc., who uses the Talent Hunt application to showcase their skills and connect with users for potential bookings.

- User Registration: The process by which individuals create accounts within the Talent Hunt application, providing necessary information like email, username, and password.

- Artist Profile: A personalized section within the application where artists present information about themselves, including their talents, multimedia content (videos, audio clips, images), and availability.

- Booking: The process through which users request the services of specific artists for their events, providing event details and preferences.

- Frontend: The user interface (UI) of the application that users interact with directly. In the case of Talent Hunt, the frontend is built using the React framework.

- Backend: The backend of the application handles business logic, database interactions, and serves as an intermediary between the frontend and the database. In Talent Hunt, the backend is implemented using the Spring Boot framework.

- Database: A structured collection of data used to store information related to users, artists, bookings, and other application-specific data. In Talent Hunt, MySQL is the chosen relational database management system.

- API: Application Programming Interface. A set of protocols and tools that allows different software applications to communicate with each other. The Talent Hunt application utilizes APIs to facilitate data exchange between frontend and backend.

- Responsive Design: A design approach that ensures the application's user interface adapts and provides an optimal experience across various devices and screen sizes.

- Social Media Integration: The ability to link and utilize information from social media platforms, such as Facebook or Google, for user authentication and registration.

- Payment Gateway: A service that handles financial transactions securely over the internet,

allowing users to make payments for artist bookings.

- Notification: Automated messages sent to users or artists to provide updates, confirmations,

or other relevant information about their interactions with the application.

- Event Venue: A physical location, such as a restaurant, pub, lounge, or private party setting, where events are hosted and where artists' performances may take place.

- Analytics : Data-driven insights and analysis that provide information about user behaviors, artist popularity, and application performance.

-Location-based Services: Features that utilize a user's geographical location to provide relevant information or suggestions, such as suggesting nearby artists.

- Streaming Services: Platforms that allow artists to broadcast live performances to an online audience in real-time.

- Administrator: A user role responsible for managing and overseeing the operation of the application, including user accounts, content moderation, and addressing technical issues.

* 1. **ER Diagram**

**3.1 purposed system**

The proposed Talent Hunt system introduces a comprehensive web-based application that transforms the landscape of talent booking and showcasing within the entertainment industry. This innovative platform aims to bridge the gap between event organizers, venue owners, and talented artists by offering a streamlined process for discovering, booking, and experiencing artistic performances. The system caters to Users seeking artists for various events and Artists looking to showcase their talents to a broader audience.

**Key Features and Functionalities:**

1. **User and Artist Registration:**

-Users and Artists can register using email or social media accounts, creating personalized profiles.

1. **Artist Profile Creation:**

- Artists can create detailed profiles showcasing their talents, skills, experiences, portfolio, and availability.

3. **Artist Discovery and Search:**

- Users can search for artists based on genres, availability, location, and ratings.

4. **Online Talent Hosting:**

- Artists can showcase their talents through online performances, expanding their reach beyond physical events.

5. **Booking Management:**

- Users can initiate booking requests, negotiate terms, and make secure online payments.

6. **Rating and Reviews:**

- Users can provide ratings and reviews after booking an artist, fostering transparency and accountability.

7. **Admin Controls:**

- Admins verify artist profiles, manage disputes, and ensure platform integrity.

8. **Real-time Notifications:**

- Users and Artists receive notifications for booking requests, confirmations, updates, and reminders.

**Enhancements Over Existing System:**

The proposed system addresses the limitations of the existing process by offering:

1. **Efficiency and Convenience:**

- Users can find and book artists from the comfort of their devices, reducing the need for physical searches.

2. **Transparent Artist Selection:**

- Online view of artist rates and reviews empowers Users to make informed decisions.

3. **Increased Artist Visibility:**

- Artists can showcase their talents online, increasing exposure and opportunities.

4. **Streamlined Booking Process:**

- Online negotiation and secure payments simplify and expedite the booking process.

**Technological Implementation:**

The system is implemented using a 3-tier architecture:

- Backend Database: MySQL for efficient data storage and management.

- Spring Boot Framework: For robust backend logic, APIs, and business processes.

- Frontend React Interface: Ensuring an intuitive and interactive user experience.

**Conclusion:**

The proposed Talent Hunt system promises to revolutionize the way Users discover, book, and interact with artists while providing Artists a platform to showcase their talents and expand their horizons. Through the integration of cutting-edge technologies and a user-centric approach, the system aims to reshape the entertainment industry's talent booking landscape, benefiting event organizers, artists, and audiences alike.

**3.2 Scope**

The Talent Hunt project encompasses a comprehensive solution that caters to both Users and Artists, enhancing the experience for both parties within the entertainment and hospitality industries. The project's scope extends to the following dimensions:

**1. Artist Booking Platform:**

- The application provides Users with a user-friendly platform to search for artists based on specific criteria such as genre, availability, location, and ratings.

- Users can initiate booking requests, negotiate terms, and complete secure online payments.

**2. Artist Profile Management:**

- Artists can create comprehensive profiles that showcase their talents, experiences, portfolio items, and samples of their work.

- Artists can manage their availability, respond to booking requests, and maintain a consistent online presence.

**3. Talent Showcasing:**

- The platform serves as an online stage for artists to host their talents, allowing them to reach a wider audience and expand their opportunities beyond physical events.

**4. Transparency through Ratings and Reviews:**

- The system enhances user experiences by providing an online view of artist rates and reviews.

- Ratings and reviews empower Users to make informed decisions before proceeding with bookings.

**5. Admin Management:**

- Admin controls ensure the verification of artist profiles and efficient dispute resolution.

- Admins manage the overall platform operations and ensure a safe and respectful environment for Users and Artists.

**6. Technology Implementation:**

- The project is implemented using a 3-tier approach, comprising a backend MySQL database, Spring Boot framework, and a frontend React interface.

- The utilization of multi-tiered architecture, server and client-side scripting techniques, and technologies such as Spring Boot, React, and MySQL is fundamental to the application's success.

**7. Object-Oriented Analysis and Design (OOAD):**

- The project employs object-oriented analysis and design methodology to define the interactions and relationships among system components, resulting in a more efficient and intuitive system.

In conclusion, the Talent Hunt project ambitiously seeks to reshape the artist booking and talent showcasing landscape within the hospitality and entertainment sectors. By facilitating seamless collaboration between Users and Artists and leveraging cutting-edge technologies, the project aims to address the existing inefficiencies and uncertainties, ultimately enhancing the experiences of event organizers, artists, and audiences alike.