Business Requirements Document (BRD)

# **Project Overview**

The Wealth Game module is an integral part of the Superstar Music Karaoke App, aiming to enhance user engagement by providing an interactive gaming experience. Users can earn virtual wealth through challenges and gifting, fostering a competitive and rewarding environment.

# **Mobile Application Development**

## Needs Analysis

Study and analyze the requirements for developing mobile applications for iOS and Android devices, including tablets and smartphones, to allow users to enjoy the app and to provide the source of income through the app.

## Solution

Design and develop mobile applications that include the following components:

**(1) User Interface (UI) and User Experience (UX) Systems**

* Prestige Room Interface: Design an intuitive interface within the Prestige Room section where users can seamlessly access the Wealth Game and other earning activities.
* Activity Navigation: Implement an easy-to-navigate menu or dashboard displaying all available activities for earning virtual wealth.

**(2) Activity Management System:**

* Challenge and Task Management: Develop a system to create, manage, and display daily challenges and tasks for users within the Wealth Game. Include challenges related to karaoke sessions, live streaming, and other app activities.
* Reward Management: Create a system to allocate virtual wealth rewards to users upon completing challenges and tasks.

**(3) Virtual Currency System:**

* Currency Earning: Implement mechanisms for users to earn virtual currency (Gold/Blue Coins) by participating in challenges, tasks, and live streaming sessions.
* Currency Spending: Develop a system allowing users to spend virtual currency on in-app items, avatar customization, gifting, and other activities.

**(4) Live Streaming System:**

* Live Streaming Features: Enhance the live streaming functionality, enabling users to host live sessions, interact with viewers, and receive virtual gifts from the audience.
* Gift Receiving: Implement a system for users to receive and acknowledge virtual gifts sent by viewers during live streaming.

**(5) Gifting System:**

* Virtual Gift Catalog: Create a catalog of virtual gifts with varying values and visual representations.
* Sending and Receiving Gifts: Develop systems for users to select, send, and receive virtual gifts. Include animations and notifications for a dynamic user experience.

**(6) Leaderboard and Achievement System:**

* Leaderboards: Design leaderboards displaying top users based on their virtual wealth earnings. Implement ranking algorithms to ensure accurate leaderboard representation.
* Achievements: Develop an achievement system recognizing users' accomplishments within the Wealth Game, encouraging continued engagement.

**(7) Backend Systems:**

* User Data Management: Develop secure backend services for managing user data, including profiles, virtual wealth balances, transaction history, and achievements.
* Integration with Superstar App: Ensure seamless integration with the existing Superstar Music Karaoke App infrastructure, allowing data exchange between the Wealth Game module and other app features.

# **Database Design**

Design and develop a scalable database system to support future resource expansion.

# **Image Processing Algorithms**

Design and develop image processing algorithms to improve data accuracy, including Karaoke singing audio and vedio mode, Live streaming, gifting, broadcasting, gaming, song storage, recorded veio-audio song posts.

# **Operation Plan**

Prepare a comprehensive operation plan for employer review, covering all project specifications.

# **Application Deployment**

Update and deploy the developed applications on Google Play Store and Apple Store.

# **Multi-Platform and Multilingual Support**

Develop applications for both iOS and Android, with support for both Hindi and English languages.

Software Requirements Specification (SRS)

**Introduction**

This Software Requirements Specification (SRS) outlines the functional and non-functional requirements of the Food Waste Tracking and Analysis System.

# **Functional Requirements**

## Mobile Applications

## User Authentication

* Users must be able to register and log in to the mobile applications.

## User Management

* Users must be able to register a new account using their phone number, Facebook, or email.
* Users should have the option to customize their avatars upon registration.
* Users must log in securely using their credentials (phone number, Facebook, or email).
* Users should be able to reset their passwords if forgotten.

## Virtual Currency

* Users should earn virtual currency (e.g., Gold Coins, Blue Coins) by participating in challenges, tasks, and live streaming sessions.
* Users must be able to view their virtual currency balance.
* Users should spend virtual currency on in-app items, avatar customization, and gifting.

## Challenges and Tasks

* The system must generate a variety of daily challenges and tasks for users to complete.
* Users should receive virtual currency rewards upon successful completion of challenges and tasks.
* Challenges and tasks must cover different app activities, including karaoke sessions, making friends, and live streaming.

## Live streaming Functionality

* Users should be able to host live streaming sessions where they can showcase their talents.
* Viewers must be able to interact with broadcasters through live chat and virtual gifting.
* Broadcasters should receive virtual gifts from viewers, which are convertible to virtual currency.

## 1.1.6. Gifting System

* Users must have access to a catalog of virtual gifts with different values and visual representations.
* Users should be able to send virtual gifts to broadcasters during live streaming sessions.
* Broadcasters must be able to view and acknowledge received virtual gifts.

## 1.1.7. Leaderboard and Achievement System

* The system should maintain leaderboards displaying top users based on virtual wealth earnings.
* Users should earn achievements for specific accomplishments within the Wealth Game module.

## 1.1.8. User Interface and Experience

* The user interface must be intuitive, ensuring easy navigation and interaction with Wealth Game features.
* Users should receive real-time notifications for completed challenges, received gifts, and leaderboard updates.
* The system must provide clear instructions and guides for users unfamiliar with Wealth Game activities.

## 1.1.9. Security and Data Privacy

* User data, including virtual currency balances and transaction history, must be securely stored and encrypted.
* The system must implement secure authentication and authorization mechanisms to protect user accounts and activities.

## 1.1.10. Integration and Compatibility

* The Wealth Game module should seamlessly integrate with the Superstar Music Karaoke App, ensuring a consistent user experience.
* The module should be compatible with both Android and iOS platforms and support various devices and screen sizes.

# **Non-Functional Requirements**

**2.1 Performance Requirements**

* The Wealth Game module should respond to user interactions within 2 seconds under normal system load conditions.
* The system should support a minimum of 1000 concurrent users without significant performance degradation.
* Image processing operations (e.g., avatar customization) should be completed within 3 seconds for a standard-sized image.
* The system should handle a minimum of 1000 transactions per minute during peak usage hours.

**2.2 Usability Requirements**

* The user interface of the Wealth Game module should be intuitive and easy to navigate, ensuring a positive user experience.
* The app should support multiple languages and provide localization options for global users.
* Clear and concise user documentation should be available within the app, explaining the Wealth Game features and how to participate.

**2.3 Reliability Requirements**

* The Wealth Game module should have a system uptime of at least 99.9% of the time, excluding scheduled maintenance periods.
* Backup and recovery mechanisms should be in place to ensure the safety of user data, including virtual currency balances and transaction history.

**2.4 Security Requirements**

* User data, including personal information, transaction history, and virtual wealth balances, should be stored securely, and encrypted at rest.
* The Wealth Game module should implement secure authentication and authorization mechanisms to prevent unauthorized access.
* Regular security audits and vulnerability assessments should be conducted to identify and address potential security risks.

**2.4 Scalability Requirements**

* The system architecture should be designed to scale horizontally to accommodate a growing user base without significant modifications to the existing infrastructure.
* The Wealth Game module should be able to integrate with third-party services or APIs for future enhancements without disrupting the existing functionality.

**2.4 Compatibility Requirements**

* The Wealth Game module should be compatible with Superstar Music Karaoke App versions 2.0 and above on both Android and iOS platforms.
* The module should be responsive and accessible across various devices and screen sizes, including smartphones and tablets.

**2.4 Maintainability Requirements**

* The codebase should be well-documented and maintainable to facilitate future updates, bug fixes, and feature enhancements.
* A version control system should be implemented to track changes and collaborate effectively among development team members.

# **Image Processing Workflow**

The image processing workflow involves several steps, as illustrated in the following flowchart:

Start

|

|--> Download App.

| |

| v

|--> Sign up.

| |

| v

|--> Login

| |

| v

|--> Home

| |

| v

|--> Profile

| |

| v

|--> Prestige Club

| |

| v

|--> Wealth Game

|

End

**Step 1: Download the App**

* Users need to install the app in their mobile phones using the app store or google store.

**Step 2: Signup**

* Need to sign up into the app either using email-password, Facebook or directly from google.

**Step 3: Login**

* Login into the app using the email-password, Facebook or directly from google.

**Step 4: Home**

* After login into the app user will redirect to the home screen.
* There are two tabs Following and Follower.
* It contains the songs recorded by the user.

**Step 5: Profile**

* This section contains the features, user details, user covers, draft songs, and settings.
* The features it contains are daily tasks, games, my live, recharge, my songs, wallet, gifts, achievements, and Prestige club.

**Step 5: Prestige club**

* In prestige club users will get a lot of options like wealth games, karaoke singing, parties’ room.

**Step 6: Wealth Game**

* In this section, users can play games and perform the tasks through which they can earn money.

# **System Architecture**

The system architecture for the Wealth Gaming and Analysis System can be designed as a distributed system to handle mobile application requests, image processing tasks, and data storage efficiently.

**Components of the System Architecture:**

1. **Front-end:** The front-end of the Wealth Game is a web application that allows users to sign up, complete tasks, watch live streams, and gift to broadcasters. The front-end is implemented using a modern web development framework, such as React or Angular.
2. **Back-end:** The back end of the Wealth Game is a microservices-based architecture. Each microservice is responsible for a specific task, such as user management, task management, live streaming, or gifting. The microservices are implemented using a programming language such as Java, Python, or Node.js.
3. **Messaging:** The Wealth Game uses a messaging queue to communicate between the front-end and back-end, and between the different microservices. The messaging queue is implemented using a message broker such as RabbitMQ or Kafka.
4. **Caching:** The Wealth Game uses a cache to store frequently accessed data, such as user data and task data. This helps to improve the performance of the application. The cache is implemented using a caching service such as Redis or Memcached.
5. **Load balancing:** The Wealth Game uses a load balancer to distribute traffic between the different microservices. This helps to ensure that the application is scalable and can handle many users. The load balancer is implemented using a load balancing service such as Nginx or HAP Proxy.
6. **Monitoring and logging:** The Wealth Game uses a monitoring and logging system to track the performance of the application and to identify any errors. The monitoring and logging system is implemented using a monitoring service such as Datadog or New Relic.
7. **Security:** The Wealth Game uses a variety of security measures to protect user data and to prevent fraud. These security measures include encryption, authentication, and authorization.

# **Database Design**

**User Table**

* This table stores user information and credentials.

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| User\_ID | INT (Primary Key) | Unique identifier for each user. |
| Username | VARCHAR | User's username for login. |
| PhoneNumber | VARCHAR | User’s phone number for registration. |
| Email | VARCHAR | User's email address for registration. |
| Password | VARCHAR | Encrypted password for user login. |

**Challenges Table**

* This table stores information about available challenges.

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| ChallengeID | INT (Primary Key) | Unique identifier for each challenge. |
| Title | VARCHAR | Title of the challenge. |
| Description | TEXT | Description of the challenge. |
| Reward | INT | Virtual wealth reward for completing the challenge. |

**Virtual Currency Table**

* This table tracks the virtual wealth (gold and blue coins) of each user.

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| UserID | Foreign Key, INT | Identifier linking to the user. |
| GoldCoins | INT | Amount of gold coins owned by the user. |
| BlueCoins | INT | Amount of blue coins owned by the user. |

**VirtualGifts Table**

* Records virtual gifts sent and received by users.

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| GiftID | INT (Primary Key) | Unique identifier for each virtual gift. |
| SenderUserID | INT (Foreign Key) | Identifier linking to the user sending the gift. |
| ReceiverUserID | INT (Foreign Key) | Identifier linking to the user receiving the gift. |
| GiftType | VARCHAR | Type or category of the virtual gift. |
| GiftValue | INT | Value of the virtual gift in wealth points. |

**AvatarCustomization Table**

* This table stores customized avatar data for each user.

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| AvatarID | INT (Primary Key) | Unique identifier for each avatar customization. |
| UserID | INT (Foreign Key) | Identifier linking to the user. |
| AvatarImageURL | VARCHAR | URL pointing to the customized avatar image. |
| SelectedElements | TEXT | JSON or serialized data representing selected avatar elements. |

# **Server Architecture Components**

1. **Web servers:** The web servers are responsible for serving the web application to users. The web servers are implemented using web server software such as Apache or Nginx.
2. **Application Servers:**  The application servers are responsible for running the back end microservices. The application servers are implemented using application server software such as Tomcat or Wild Fly.
3. **Messaging queue:** The messaging queue is responsible for communicating between the front-end and back-end, and between the different microservices. The messaging queue is implemented using a message broker such as RabbitMQ or Kafka.
4. **Cache server:** The cache server is responsible for storing frequently accessed data to improve the performance of the application. The cache server is implemented using a caching service such as Redis or Memcached.
5. **Load balancer:** The load balancer is responsible for distributing traffic between the different web servers and application servers. The load balancer is implemented using a load balancing service such as Nginx or HAProxy.
6. **Monitoring and logging server:** The monitoring and logging server is responsible for tracking the performance of the application and identifying any errors. The monitoring and logging server is implemented using a monitoring service such as Datadog or New Relic.
7. **Security server:** The security server is responsible for protecting the application from attacks. The security server is implemented using security solutions such as a firewall, intrusion detection system, and intrusion prevention system.

# **User Story**

## User Persona:

**Name:** Alice Thompson

**Background:** Alice is a 25-year-old graphic designer who loves music and social interactions. She is tech-savvy and enjoys mobile gaming during her free time.

**Demographics:**

* Age: 25
* Gender: Female
* Location: Urban resident
* Occupation: Bachelor's degree in graphic design

**Goals and Motivations:** Alice wants to have a fun and engaging online experience, make new friends, and showcase her creativity through avatar customization.

**Pain Points:** Limited virtual wealth, occasional lack of engagement in other games, looking for a social platform that aligns with her interests.

## User Journey:

**User Persona:** Alice Thompson

**User Journey Stages:**

1. **Discovery and Registration :**

* Alice learns about the Superstar Music Karaoke App through social media.
* She downloads the app and registers using her email.

1. **Exploring Features :**
   * Alice explores the app features, discovering the Wealth Game module.
   * She learns about challenges, avatar customization, and virtual gifts.
2. **Avatar Customization :**
   * Motivated by creativity, Alice goes to the Avatar Customization section.
   * She uploads her own designs and customizes her avatar with unique elements.
   * Alice feels accomplished as her avatar reflects her personality.
3. **Participating in Challenges :**
   * Alice joins a singing challenge in the Wealth Game.
   * She practices and submits her performance.
   * She wins the challenge and earns virtual wealth.
4. **Gifting and Making Friends :**
   * Alice receives a virtual gift from another user as appreciation for her challenge performance.
   * She appreciates the gesture and sends a thank-you gift in return.
   * She starts chatting with the sender and makes a new friend on the platform.
5. **Achieving Goals:**
   * Alice sets a goal to reach the top of the leaderboard in the Wealth Game.
   * She participates in more challenges, customizes her avatar further, and engages in live streaming to earn virtual wealth.
   * Alice achieves her goal and feels a sense of accomplishment.
6. **Long-term Engagement :**
   * Alice continues to participate in challenges, customize her avatar, and interact with others in the Wealth Game.
   * She has become a respected member of the community, helping newcomers and enjoying her time in the app.

# **Test Cases:**

Below are some sample test cases for the Food Waste Tracking and Analysis System:

Test Case 1: User Registration

**Scenario:** Alice attempts to register a new account in the Wealth Game module.

**Test Steps:**

1. Navigate to the registration page.
2. Enter valid credentials (email, password, username).
3. Click on the "Register" button.

**Expected Result:** Alice’s account is successfully registered, and the system redirects the user to the dashboard of the Wealth Game module.

Test Case 2: Avatar Customization

**Scenario:** Alice attempts to customize their avatar using uploaded images and predefined elements.

**Test Steps:**

1. Go to the Avatar Customization section.
2. Upload a valid image for avatar customization.
3. Select predefined elements (hairstyle, clothing, accessory).
4. Adjust colors and other customization options.
5. Save the customized avatar.

**Expected Result:** Customized avatar is saved successfully. The avatar reflects the chosen elements and customization preferences accurately.

Test Case 3: Challenge Participation

**Scenario:** Alice participates in a singing challenge in the Wealth Game module.

**Test Steps:**

1. Select a singing challenge from the available options.
2. Practice and record the performance.
3. Submit the performance for evaluation.

**Expected Result:** Challenge submission is successful. The system processes the performance, evaluates it, and awards virtual wealth to the user based on the performance quality and rules of the challenge.

Test Case 4: Live Streaming Interaction

**Scenario:** Alice interacts with a live streaming session in the Wealth Game module.

**Test Steps:**

1. Join a live streaming session.
2. Participate in the chat with other viewers.
3. Send a virtual gift to the broadcaster.

**Expected Result:** User successfully joins the live streaming session, interacts with others, and sends a virtual gift. The virtual gift is deducted from the user's virtual wealth balance and is displayed in the chat.

Test Case 5: Social Interaction and Friend Making

**Scenario:** Alice interacts with other users and makes a new friend in the Wealth Game module.

**Test Steps:**

1. Receive a virtual gift from another user.
2. Send a thank-you gift in return.
3. Initiate a chat with the sender.
4. Add the sender as a friend.

**Expected Result:** User successfully interacts with the sender, sends, and receives gifts, initiates a chat, and adds the sender as a friend. The friend is added to the user's friend list.