

# Technical Specs Document

## Technical Specifications Document

### 1. Introduction

#### **Purpose:**

This document outlines the technical specifications for the KSU Rec Center app, detailing its objectives, functionalities, and design requirements.

While functional, KSU's current Rec Center app lacks the modernity in design and usability in function expected of today's applications. Issues such as outdated UI design and overall complexity of use repels users from the app and inhibits them from its current offerings. Our project seeks to address these issues by developing a new KSU Rec Center app. The primary focus of our app is to enhance the user experience and maximize accesibility. This will be done by having a modern UI that simplifies navigation and interaction.

#### **Scope:**

The scope includes the development of a new rec center app with key features such as personal profiles, event registrations, easy access to facility information(such as maps) and the ability to submit the necessary forms. This app will be built with modern techonologies such as React Native and a Node.js backend to handle data securely. We will also integrate APIs to manage student information, real time scheduling, and notifications. The maintanance of the existing app is not covered by this project.

#### **Audience:**

Our targets are KSU students,faculty, and staff who use the Rec Center facilities regularly and need a seamless way to interect with the rec center's services. This document will serve as a guide for developers involved in the development of this new app.

### 2. System Overview

#### **System Purpose:**

The new KSU Rec Center app aims to provide a modern and user-friendly platform for students, faculty, and staff to access and engage with Rec Center services.

## Architecture Overview:

The KSU Rec Center app will utilize a client-server architecture. The Rec Center app will act as the client, handling user interaction and presenting data while sending requests to the secure backend server. This server will manage data and business logic.

## Key Components:

### Client (Mobile App):

**Technology:** This app will be built using React Native, enabling the development of a user interface that allows users to access and interact with the Rec Center's services.

### Features:

- Register and log in to personalized accounts.
- Receive notifications, view, and register for upcoming events.
- Access Rec Center information such as operating hours and map locations.
- Submit forms for consent, class registrations, and feedback.

### Backend Server:

**Technology:** The backend will be developed using Node.js and Express to handle HTTP requests and business logic.

### Responsibilities:

- **Data Management:** Operations on user profiles, event registrations, and facility data.
- **User Authentication:** Implement secure authentication to manage user sessions and access control.
- **API Integration:** The backend will connect to third-party APIs for real-time scheduling of events and push notifications for events and reminders.

### Database:

**Technology:** MongoDB will be used to store and access data.

**Security Measures:** All data transmitted between the client and server will be managed by the school's cybersecurity systems.

### User Management:

**Purpose:** Manage user accounts and ensure secure authentication.

### Key Functionalities:

- **Account Creation:** Users will sign up using their Owl Express information.
- **User Authentication:** Secure login will be implemented with KSU's DUO system.
- **Profile Management:** Users will be able to edit their personal profiles.

### **Event Management:**

**Purpose:** Assist in the discovery and registration of Rec Center events.

#### **Key Functionalities:**

- **Event Listings:** Display upcoming events and their details, including date, time, location, and description.
- **Event Registration:** Allow users to register for events and manage their registrations.
- **Notifications:** Send reminders and updates about registered events.

### **Facility Information:**

**Purpose:** Provide essential information about the KSU Rec Center facilities.

#### **Key Functionalities:**

- **Facility Overview:** Description of both the Marietta Campus and Kennesaw Campus, including their facilities and features.
- **Operating Hours:** Display hours for both Marietta Campus and Kennesaw Campus Rec Centers.
- **Maps:** Help users navigate the Rec Centers.

### **Feedback and Support:**

**Purpose:** Allow users to provide feedback and receive support.

#### **Key Functionality:**

- **Feedback Submission:** Allow users to write comments about the service at each location.

### **User Notifications:**

**Purpose:** Keep users informed about updates and alerts.

#### **Key Functionality:**

- **In-App Notification:** Display updates concerning classes and events the user is enrolled in.

### **User Roles:**

#### **General Users:**

- Primary users of the app who utilize the Rec Center facilities and services.

**Key Responsibilities:**

- Log in.
- Create and manage personal profiles.
- Access facility information such as hours, locations, and amenities.
- Submit forms.
- Receive notifications.

- **Admin:**

Staff responsible for managing the app's user accounts and content.

**Key Responsibilities:**

- Update facility information and resources.
- Handle user feedback and inquiries.
- Monitor user activity and engagement with the app.

- **Event Coordinators:**

Staff members who plan and oversee events and classes at the Rec Centers.

- **Key Responsibilities:**

- Update facility information and resources.
- Handle user feedback and inquiries.
- Monitor user activity and engagement with the app.

- **Facility Managers:**

Staff responsible for overseeing the Rec Center facilities and ensuring maintenance availability.

- **Key Responsibilities:**

- Update operating hours and availability of facilities.
- Provide information on facility usage policies.
- Communicate with users about facility-related matters.

### 3. Functional Requirements

In this section, we list the functionalities that will be present in the new Rec Center app and describe how they will operate.

## 3.1 User Interface

### Login Page:

- The app shall display a message prompting the user to enter their credentials.
- The app shall display input fields for both school email and password.
- The app shall display a submission button for the input fields.

### Today Tab:

- The app shall display a welcome message and a daily reminder.
- The app shall display the events for the current day as well as their respective times.
- The app shall display a calendar view for monthly events.
- The app shall display Today, Map, Services, and Profile tabs to allow users to change tabs.

### Interactive Maps Tab:

- The app shall display "Interactive Maps" text.
- The app shall display two tabs allowing the user to alternate between Marietta and Kennesaw Rec Center map locations.
- The app shall display individual maps with area descriptions.
- The app shall display Today, Map, Services, and Profile tabs to allow users to change tabs.

### Services Tab:

- The app shall display "Services" text.
- The app shall display icons pertaining to individual KSU Rec Center classes.
- The app shall display a pin icon to allow users to pin their preferred classes.
- The app shall display a button to allow users to subscribe and unsubscribe to the newsletter.
- The app shall display a button to allow activity registration.
- The app shall display Today, Map, Services, and Profile tabs to allow users to change tabs.

### Profile Tab:

- The app shall display the user's profile image from their student ID.
- The app shall display the user's full name and their current level.
- The app shall display a dynamic streak tracker.
- The app shall display a dynamic weekly log.
- The app shall display a button to redirect the user to favorites (workouts, classes).
- The app shall display a button to redirect the user to hours of operation.

- The app shall display a button to redirect the user to the waiver sign.
- The app shall display Today, Map, Services, and Profile tabs to allow users to change tabs.

#### **Comment Card:**

- The app shall display a toggable thumbs up icon.
- The app shall display a toggable thumbs down icon.
- The app shall display a "What can we improve?" message.
- The app shall display an input field for user comments and complaints.
- The app shall display a submission button for the input field.

### **3.2 Application Features**

#### **Login/Authentication:**

- The app will implement the already established DUO credential validation system.
- If the user becomes locked out of the DUO login system, the app will not allow the user to enter their credentials.
- Successful login will direct the user to the Today tab, and their session will be authenticated with the time described by the DUO system.

#### **Event Registration:**

- Users can view upcoming events in the calendar and register through the app directly.
- After the user registers for an event, they will be prompted with a confirmation message.
- In the case of a full event or registration error, the user will receive a notification regarding the issue.
- Users will be able to cancel or modify their registrations within the app, and the app will update their profile dynamically.

#### **Profile Management:**

- The streak tracker will update automatically every time the user taps into the Rec Center. In the case of Rec Center inactivity, the streak will reset to 0.
- The weekly log will update with the user's completed activities, displaying a summary at the end of each week.

#### **Notifications:**

- Users will receive push notifications reminding them of upcoming events or new Rec Center updates.
  - The app will allow users to customize which notifications they will receive.
- Notifications will be shown in-app as push notifications, which users can enable or disable.

### Interactive Maps:

- The maps will allow users to zoom in and out. They will also be able to click and zoom into specific areas and view detailed descriptions of each facility.
- Users will be able to toggle between Kennesaw and Marietta locations seamlessly.

### Comment Card:

- Users can leave feedback directly from the comment card tab section.
- The toggleable thumbs up and thumbs down feature will allow users to indicate positive and negative experiences while being allowed to go into detail in the input field.
- The app shall display a submission confirmation once feedback is sent, and the comments will be logged for review by admins. The user will be thanked for their submission.

## 4. Technical Design

### System Architecture

The KSU Rec Center application will follow a Client-Server architecture, where the frontend interfaces with the users and the backend manages data processing, storage, and retrieval.

### Frontend

- The UI of the Rec Center app will be developed using the cross-platform framework, React Native, to cater to both iOS and Android platforms.
- The app will utilize the MVVM (Model-View-ViewModel) architecture to bind UI components with data models through reactive programming.
- The core UI design will be created in Figma and later implemented in React Native.

### Backend

- The MongoDB will handle the backend systems. It will manage user data, events, and gym information while interfacing with external services for authentication and notifications.
- Node.js will be used to implement the backend logic.
- External systems such as DUO will be integrated for authentication, and third-party services like the Google Maps API may be used for handling interactive maps and push notifications.

### Data Flow Diagram

```
plaintext
Copy code
graph LR
  A[User Interface] -->|User Requests| B(API Gateway)
  B --> C{Backend Services}
```

```
C -->|Authentication| D[User Profile Service]
C -->|Events Handling| E[Event Management Service]
C -->|Equipment Info| F[Gym Layout Service]
D --> G[Database]
E --> G
F --> G
```

The app will also interface with external systems like DUO for two-factor authentication and may use third-party services for sending notifications or handling maps.

## Module Descriptions

### User Modules

- **Login/Registration Module:**
  - Handles authentication methods including OAuth.
  - Integrates with DUO for two-factor authentication, ensuring secure login for users.
  - Manages user sessions and credentials.
- **Profile Management Module:**
  - Allows users to view and update their profiles.

### Event Modules

- **Today's Feature Module:**
  - Pulls daily quotes and event information from the server.
- **Event Calendar Module:**
  - Manages the display and interaction with the monthly calendar.
- **Booking System Module:**
  - Handles registration for different recreational services, including real-time updates when users register, cancel, or modify bookings.

### Gym Information Module

- **Interactive Map Module:**
  - The app will integrate a map AI to describe the KSU Rec Center layout, potentially using the Google Maps API to handle map data.
  - Users can zoom in on specific areas of the map and view facility details.

## Performance Considerations

- Use of lazy loading and pagination for displaying events and activities.
- Implement caching strategies for frequently accessed data, potentially using services like Redis or client-side caching mechanisms.



- Minimize API calls through efficient data fetching and updating mechanisms, possibly using state management tools like Redux or React Query to reduce unnecessary re-fetching of data.

## Best Practices

- Ensure code modularity for easier updates and feature expansions.
- Adhere to platform-specific UI/UX design guidelines.
- Conduct thorough testing, including unit tests, integration tests, and UI tests.
- Set up a Continuous Integration/Continuous Deployment (CI/CD) pipeline for streamlined development.

## Security Best Practices

- Follow security guidelines, including encryption for sensitive user data, secure authentication processes (e.g., OAuth and DUO integration), and regular security audits to ensure system integrity.

# 5. Hardware/Software Requirements

### Development Tools:

- **IDE:** Visual Studio Code
  - Recommended Extensions:
    - ESLint for JavaScript linting.
    - Prettier for code formatting.
- **Cross-platform Framework:** React Native
- **Version Control:** Git with GitHub

### Server Infrastructure:

- **KSU Pre-established APIs and Servers**
- **Containerization:** Docker containers for environment standardization

### Users:

- **Supported Devices:**
  - Current Android and iOS versions
  - Minimum Requirements:
    - **RAM:** Minimum 2 GB
    - **Storage:** Minimum 100 MB available for app installation

- **OS Versions:**

- Android 8.0 (Oreo) and above
- iOS 12 and above

- **Not Compatible On:** Windows, Linux, or Mac

### Testing Tools:

- **Unit Testing:** Jest or Mocha
- **UI Testing:** Selenium

### Deployment Tools:

- **Hosting Services:** Heroku or AWS for backend deployment

### Security Tools:

- **Vulnerability Scanning:** OWASP ZAP for identifying security vulnerabilities

## 6. Security Considerations

- **DUO Authentication System:**

- Implement the DUO authentication system to ensure secure logins through two-factor authentication, adding an additional layer of security.

- **Secure Token Management:**

- Ensure that all authentication tokens are securely stored and transmitted using HTTPS to prevent interception during transmission.

- **Role-Based Access Control:**

- Utilize role-based access control (RBAC) to manage user permissions, ensuring that users only have access to features and data relevant to their roles.

- **Data Encryption:**

- Implement data encryption both at rest and in transit using KSU's encryption services to protect sensitive information from unauthorized access.

- **Input Validation:**

- Validate data submitted through forms to avoid malicious input and prevent common vulnerabilities such as SQL injection and cross-site scripting (XSS).

- **API Security:**

- Secure the API Gateway by ensuring proper authentication and authorization for API endpoints.
  - Use API keys for secure API access, preventing unauthorized usage of the APIs.
  - **Session Management:**
    - Implement secure session management by using session tokens that will expire after a set period of time to reduce the risk of session hijacking.
  - **Data Privacy Compliance:**
    - Comply with data privacy regulations (such as GDPR or HIPAA, as applicable) to ensure that user information is handled properly and transparently.
  - **Audit Logging:**
    - Maintain audit logs to track critical system events, ensuring these logs are accessible to authorized personnel for monitoring and analysis.
  - **Data Backup:**
    - Implement regular data backups for critical user data to prevent loss in case of data corruption or system failure.
  - **Disaster Recovery Plan:**
    - Ensure the system has a disaster recovery plan in place to restore data quickly in the event of a system failure, minimizing downtime.
  - **Access Controls:**
    - Enforce strict access controls and authentication mechanisms to protect sensitive data and application features from unauthorized access.
  - **Session Expiry:**
    - Implement session expiry policies to log users out after a period of inactivity, further enhancing security.
- 

Document Revision: 2.0

Date: [10/22/2024]

*Note: This document is subject to change as project requirements evolve.*