Student Name: Olufiyin Ben-Stowe

Student ID: 0100623

Requirement Analysis for RSVP'd (Fisk Event Management and RSVP System)

Introduction:

This project aims to make event management for clubs on the Fisk University campus more efficient. This web-based platform allows users club admins to create and manage events, track RSVPs, and handle room bookings, while students can RSVP to events, view their event schedules on a calendar and offer feedback post events. The project will prioritize fundamental functionality to ensure usability and scalability, focusing on necessary features with potential for future extensions. The user requirements, functional requirements, and non-functional requirements necessary for the project's successful development are outlined in this thorough analysis.

Stakeholder Identification:

1. **Club Admins:** Students/ club e-board members in charge of running campus clubs and organizing events
2. **Students (End Users):** Students (also primary users) who RSVP, attend events, and provide feedback for the club admins
3. **Room Booking Managers:** University staff who verify the availability of rooms/ event spaces
4. **Project Developer (Me):** Responsible for building and maintaining the web platform

User Requirements:

1. **Club Admins:**
    - Ability to create events with details such as event title, event type, description, time, and location
    - Option to track RSVPs (Yes, No, Maybe)
    - View Upcoming and past events

- Ability to reserve rooms and avoid double booking

2. **Students (End User):**
    - Easy registration and RSVP process for events
    - Ability to view upcoming events on a calendar
    - Option to receive reminders/ notifications about RSVP'd events
    - Provide event feedback post event.

Functional Requirements:

1. **Event Management:**
    - Club admins can create, edit, and delete events
    - Students have the option to RSVP to events with options (Yes, No, Maybe)
    - Club admins can access RSVP data for planning and budgeting purposes
2. **RSVP Tracking:**
    - Club admins can instantly view RSVP responses
    - Club admins can use RSVP responses to more accurately budget and make provisions for events
3. **Room Booking:**
    - Club admins can book rooms while creating events
    - The system verifies room availability to prevent double-booking
4. **Calendar View:**
    - Display of events (past and upcoming) in a monthly, weekly, and daily view
    - Club admins can view all created events by date
5. **Feedback System/ User Engagement:**
    - Students can leave feedback on events after attending
    - Club admins can view feedback and ratings to improve future events

Stretch Goals:

1. **Map Integration:**
    - Integration with Google Maps API or Mapbox to show event locations on the campus map
    - Users should be able to filter events based on location

Non-Functional Requirements:

1. **Performance:**
    - The system should handle multiple users without noticeable degradation
    - Calendar displays should update instantly and events should load within 2 to 3 seconds
2. **Usability:**
    - A user-friendly interface for club admins and students alike
    - Browser compatibility and responsiveness
3. **Security:**
    - Implement user authentication and secure password storage
    - Use SSL for secure data dispatching
    - Prevent unauthorized access to administrative tasks like event creation and room reservation
4. **Scalability:**
    - Design database and back-end architecture with scalability in mind
    - Future scalability for integrating map view and feedback system
5. **Reliability:**
    - High availability with little maintenance downtime
    - Automatic backup of event and RSVP data to prevent data loss
6. **Maintainability:**
    - Validated code to facilitate upgrades and future developments
    - Version control using Git to track changes/ modifications