Victor Udeh

CS360 Module 3

July 20th, 2024

3-3 Project 1

**App Development Proposal for Event-Tracking App**

**Project Goals**

The primary goal of the Event-Tracking App is to provide users with an efficient and user-friendly tool to manage their upcoming events. The app will allow users to log in, view, add, remove, and receive notifications for events. This app aims to help users stay organized and ensure they do not miss important events.

**Application Description**

The Event-Tracking App will consist of the following major components:

1. **User Authentication System**: A secure system for users to log in and create new accounts.
2. **Event Management System**: A feature that allows users to add, remove, and edit event details.
3. **Event Notification System**: A notification feature to remind users of upcoming events.

**Users and Assumptions**

The primary users of the Event-Tracking App are individuals who need to manage their schedules and track important dates. We assume that these users need a straightforward and reliable way to organize their events and receive timely reminders. The users might range from students and professionals to homemakers and retirees.

**Types of Users:**

1. **Regular Users**: Individuals managing personal events.
2. **Professionals**: Users managing work-related events and deadlines.
3. **Students**: Users tracking academic deadlines and social events.

**User Goals:**

1. **View Events**: Users need to quickly see a list of their upcoming events.
2. **Add/Remove Events**: Users need to easily add new events or remove past or canceled events.
3. **Receive Notifications**: Users need timely reminders about upcoming events.

**User-Centered UI Design**

**Screens and Features:**

1. **Login/Create Account Screen**:
   * Fields: Username, Password
   * Buttons: Login, Create Account
   * Navigation: Directs to the Event List screen upon successful login or account creation.
2. **Event List Screen**:
   * Displays a grid of upcoming events.
   * Buttons: Add Event, Remove Event
   * Navigation: Directs to the Add/Edit Event screen when adding or editing an event.
3. **Add/Edit Event Screen**:
   * Fields: Event Name, Date, Time, Description
   * Buttons: Save, Cancel
   * Navigation: Returns to the Event List screen upon saving or canceling.
4. **Event Notification System**:
   * Automatically notifies users on the day of an event.

**User Flow:**

1. **Login/Create Account**: User logs in or creates an account, then is redirected to the Event List screen.
2. **View Events**: User views all upcoming events in a grid format.
3. **Add/Edit Event**: User clicks on 'Add Event' to enter event details and save, which redirects back to the Event List screen.
4. **Remove Event**: User clicks on 'Remove Event' to delete an event from the list.
5. **Notifications**: Users receive notifications on the day of an event.

**Code Design and UI Integration**

**Data Flow:**

1. **Login/Create Account**:
   * **UI Components**: Text fields for Username and Password, buttons for Login and Create Account.
   * **Data Flow**: User input is sent to the authentication system, which checks credentials or creates a new account and returns the status to the UI.
2. **Event List**:
   * **UI Components**: Grid view displaying events, buttons for Add Event and Remove Event.
   * **Data Flow**: The event list is fetched from the database and displayed. User actions trigger add or remove functions.
3. **Add/Edit Event**:
   * **UI Components**: Text fields for event details, buttons for Save and Cancel.
   * **Data Flow**: User inputs are sent to the event management system to be saved in the database.
4. **Event Notification**:
   * **UI Components**: None (background service).
   * **Data Flow**: The system checks the event dates daily and sends notifications to users about upcoming events.

**Major UI Components and Data**

1. **Login/Create Account Screen**:
   * Components: Text fields for Username and Password, buttons for Login and Create Account.
   * Data: User credentials from input fields.
2. **Event List Screen**:
   * Components: Grid view for displaying events, buttons for Add Event and Remove Event.
   * Data: Event details fetched from the database.
3. **Add/Edit Event Screen**:
   * Components: Text fields for Event Name, Date, Time, Description, buttons for Save and Cancel.
   * Data: User inputs for event details.
4. **Event Notification System**:
   * Components: Background service.
   * Data: Event dates and times from the database.

**Justification**

Following the Android Design and Quality Guidelines ensures a consistent and user-friendly experience. The guidelines recommend clear navigation, intuitive input fields, and timely notifications, all of which are incorporated in this proposal.

**Conclusion**

The Event-Tracking App is designed to help users efficiently manage and track their events with a user-centered design. The integration of the authentication system, event management, and notification features will ensure a seamless and practical user experience.