# **EventPlan - Event Management & Coordination App**

### **Objective**

EventPlan aims to simplify event planning and management for individuals and small teams. The app facilitates the coordination of tasks, budget tracking, guest management, and real-time communication, providing an end-to-end solution for organizing events efficiently.

### **Key Features**

* **User Accounts & Profiles**: Secure sign-up/login, personal profiles, and the ability to manage multiple events.
* **Event Creation & Dashboard**: Users can create events with basic details and view a dashboard for each event showing key stats, upcoming tasks, and budgets.
* **Task & Checklist Management**: Allows users to add and assign tasks, set deadlines, and track progress.
* **Guest Management**: Invite guests via email or link, send RSVP requests, and track responses.
* **Budget & Expense Tracking**: Budget planning, expense logging, and visual budget tracking (actual vs. planned).
* **Collaborative Chat**: In-app chat for real-time communication with collaborators and guests.
* **Event Timeline & Reminders**: Event countdown, timeline views, and reminders for tasks and key dates.
* **Reports**: Generate reports on budget utilization, RSVPs, and task completion to assess event readiness.

### **Screens**

1. **Login & Registration Screen**: Secure login and onboarding.
2. **Profile & Settings Screen**: Manage user details, app preferences, and security settings.
3. **Event Dashboard Screen**: Overview of all ongoing and past events.
4. **Create Event Screen**: Basic event details including name, type, date, and location.
5. **Event Details Screen**: Event-specific dashboard with budget overview, task progress, and guest list.
6. **Task Management Screen**: Add and assign tasks with due dates and status tracking.
7. **Guest Management Screen**: Invite guests, send RSVPs, and view attendance status.
8. **Budget Tracking Screen**: Set budget goals and record expenses.
9. **Expense Management Screen**: Record, categorize, and track event-related expenses.
10. **Timeline Screen**: Visual event timeline and countdown.
11. **Chat Screen**: Real-time messaging with collaborators.
12. **Reports Screen**: Generate reports on expenses, RSVPs, and task status.
13. **Notifications Center**: Manage reminders and alert settings for event updates.

### **Technical Requirements**

* **Platform**: Native Android
* **Languages**: Java/Kotlin for Android development
* **Database**: Firebase Firestore for real-time data storage
* **Authentication**: Firebase Authentication (email, Google)
* **Storage**: Firebase Storage for image and file sharing (event documents)
* **Push Notifications**: Firebase Cloud Messaging for reminders and updates
* **PDF Generation**: iText for exporting reports
* **Development Environment**: Android Studio

### **Non-Functional Requirements**

* **Performance**: Optimized for real-time updates, particularly for task, budget, and RSVP changes.
* **Usability**: Easy navigation with a clean UI for event management and coordination.
* **Reliability**: Firebase ensures high availability and secure data access.
* **Security**: End-to-end encryption, secure data access via Firebase Rules.
* **Scalability**: Can handle multiple events with varying levels of complexity.
* **Responsiveness**: Adaptive UI for all Android screen sizes.

### **Development Milestones**

1. **Phase 1: Core Features**
   * Tasks: Profile setup, event creation, guest management, and task list functionality.
2. **Phase 2: Budget Tracking and Reports**
   * Tasks: Expense tracking, budget overview, and report generation.
3. **Phase 3: Collaborative Chat and Notifications**
   * Tasks: In-app chat, notifications, and reminders for events and tasks.
4. **Testing and Deployment**
   * Final testing, Play Store deployment, and post-launch support.

### **Additional Libraries and APIs**

* **UI Libraries**:
  + Material Design, RecyclerView for lists, and MPAndroidChart for visual budget tracking.
* **Network**: Retrofit or Volley for any backend interactions.
* **PDF Export**: iText for generating downloadable reports.

### **Security and Data Privacy**

* **Data Protection**: Firebase rules and SSL encryption for data in transit.
* **Privacy Compliance**: Clear data usage terms for user privacy.

### **Analytics and Crash Reporting**

* **Analytics**: Firebase Analytics to track feature usage and engagement.
* **Crash Reporting**: Firebase Crashlytics to monitor app stability.

### **Development and Testing Tools**

* **Android Studio, Postman, JUnit, and Espresso** for testing and debugging.

### **Deployment**

* Google Play Store deployment, CI/CD with GitHub Actions, and regular updates.

### **Summary of Technology Stack**

|  |  |
| --- | --- |
| **Component** | **Technology** |
| Language | Java/Kotlin |
| Database | Firebase Firestore |
| UI/UX | Material Design, RecyclerView, MPAndroidChart |
| Network | Retrofit/Volley |
| PDF Generation | iText |

### **Topics Covered:**

1. Jetpack compose
2. MVI
3. Dependency Injection (Hilt)
4. Unit Testing
5. UI Testing