**Event Planner**

**Goals:**

I have selected the Event-Tracking application for the project and named it “Event Planner.” Its significant features include the ability to schedule the event date and receive a notification before the event. This app provides a way to create a user account and log in using the account details with username and password. Users can create events with essential (event name, date) and optional information (description, location, notification), update events if any modification is required later, or remove the event if canceled. This application's functionality includes speed, reliability, and the ability to navigate the back button and the home screen with the help of user-friendly access buttons and symbols.

**Users:**

This application's targeted users are individuals who need a personal planner app to manage their plans for the day. Users like sports persons, dancers, singers, event managers, personal assistants, etc., can use this app to manage upcoming events. The goals of the users might change for every user; for example, if the sports person uses this app, he/she might use it to set the date, time, and place for practice or match and have the option to add all the details like a jersey, bat, ball, and other kinds of stuff needs for a game in the description field. If the person uses this for parties, they can add the number of people, invitation, event venue, and food. The common feature used for all kinds of users is date and time because it is crucial for event-tracking applications. The main motive of this app is to help users organize their events with all the necessary details and achieve the event as planned without any issues. The notification feature will help users prepare for the event in advance, e.g., doctor appointments are planned long ahead, which can be easily forgotten.

**UI design:**

The Event Planner app's UI design is critical and should be user-centric. A well-designed interface is vital to attracting and retaining users, and our app's interface is intuitive and easy to navigate, making it a pleasure to use. The first screen is the login screen, with App Name (Event Planner) in the top center of the screen, followed by “Username” and “Password” Text Views components with their Edit Text components. Grid Pattern is going to be used to set proper layout so that it can achieve Android UI standards. Password field input type should be marked as “textPassword”; this will make sure to hide the plain text password. The “Submit” button should be placed in the center so you can access it easily after filling in the username and password details. For first-time users to create an account, a “Signup” link will be provided next to the submit button to open an account creation page. Another “Forget/Reset password" link is next to the “Signup” link. Once the “Signup” or “Forget/Reset password” option is completed, the user has navigated automatically to the Login screen instead of the user manually clicking “Login.”

After a successful login, the Home page displays an upcoming events grid. The events are shown in ascending order, helping the user see the most recent event at the top. If more events do not fit the current screen size, the Gesture navigation option is used to scroll and check all the events. Each event is displayed with a fixed height/widget, and spacing between each event is also fixed irrespective of event name length to maintain consistency (UI Design, Figure 3: Don’t disrupt readability).

To create a new Event, the user clicks the “+” icon on the top right of the home page; it will open a “New Event” page to provide all the event details. Once the information is filled in, the user clicks the “Save” button, and it automatically redirects to the home page, where the newly created event will be displayed. On the other hand, if the user decides there is no need to create the event, the “Cancel” button can be used to go back to the home screen without saving.

To edit an existing event, the user taps on the event on the home screen, which opens the event with all its details. Users can update the event details, such as name, description, date/time, etc. To delete the existing event, swipe to the left of the event and confirm the delete operation. It will remove the event and refresh the page, and now the deleted event is not there.

**Code Design:**

In the Login screen, the “Login” button component is defined with the “android:onclick” attribute, which will take data flow from the UI to the code specified in the “MainActivity.java” file. If the login credentials are correct, then by using ActivityResultLauncher, the flow redirects to the Home Page screen; otherwise, the error message is displayed and stays on the same login screen.

To display the home page, the onCreate method calls the database to get all the event details and feeds into the Grid view to display. If no events are created, then the “Upcoming Events” section will appear empty. To update an event, upon an event, click call the database using the event ID to get stored event details and display the event details screen with retrieved event details. Modified event details were sent back to the database for update, and the home page was refreshed with the latest event details. On the event delete event, the delete operation is sent to the database with the event ID; the event gets deleted and removed from the home screen after the page is refreshed.

Screen Components:

Login Screen: TextView, EditView, Button

Home Page: ListView, ListActivity, TextView, Button

New Event/Edit Event Page: Datepicker, TextView, EditView, Button, Spinner

Input Components are EditView (Username, password, event name, event description, event location), Datepicker to provide event date and time, and Spinner to choose the notification period.

Display Components are TextView, which displays labels/captions for components or sections, and the ListView component, which shows all the events received from the database.

**Reference:**

UI Design

<https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-basics>