



Mobile2App Project Proposal
Option 2: Event Tracking Application

Proposal Revision History	1
Project Overview	1
Expected Users and User-needs	2
Features and Screens	3
Sample Screens	4
User Interface Connections and Data Flow	5

Proposal Revision History

Version	Date	Author	Comments
1.0	July 13, 2022	Lawrence Artl	Initial Proposal

Project Overview

For the development of the Event Tracking application the team has kept one thing in focus: simplicity. While the functions found in this application can likely be found in other applications, the intentional lack of depth is designed for users who want to make a simple event without worrying about extraneous information or options. The inspiration for such an app originates in the currently available Google “Keep” app, a task / reminder application that keeps things simple for users. However, the Keep app lacks a certain visual aspect that some users may find plain or difficult to navigate. The tasks and reminders that are created are kept in a long list that must be scrolled through, and each item inspected to see dates created and projected dates for notifying the user of events. This can make summarizing the data visually somewhat difficult for users.

For the Mobile2App Event Tracking application, the development team sought to design a user interface that combines the simplicity of the Keep application while adding a certain visual component to the main page to help users see at a glance what events are set to go off on a particular date. To accomplish this, the main interface page is designed with a calendar widget that takes up the majority of the screen (see figure 2 in Sample Screens, below). This calendar will show users through colored icons which dates events are scheduled for. Clicking on the dates of the calendar will open up a fragment page that displays the events for that date. The calendar can be expanded to fill the screen with a quick swipe-up (figure 3), or minimized to a single action button by swiping down (figure 4). This layout and motion-set keeps with the intuitive nature of the standard Material design as outlined by the Android Design and Quality documentation.

Events themselves can be edited, but the amount of details that can be added is limited; the application is designed to create quick reminders and events on the fly, similar to writing a note on a sticky pad. Users can set a date and time to remind them with a pop-up notification. Users can also add information about the event itself, and color code it if they choose. The events will display in a truncated list on the layer below the calendar widget on the main page (figure 2), but expands when the user swipes the calendar away (figure 4). The list on this screen is scrollable, with the next-up event showing at the top of the screen and older events below that. Events on this list can be edited by clicking on them individually, when they open a page fragment to fill the screen (figure 5).

The application will require users to create an account and log-in in order to see their personal event calendars. They can create multiple accounts (for work and personal use, for example) and then share their events with others. Sharing events will display a user’s events on another user’s calendar, and vice versa. An option will be available to hide other user’s calendars. User accounts will be stored in a server database, and each user’s events will be stored in a separate server database. Both databases will be accessible with an internet connection in the event the user changes phones or wishes to access their events from a desktop computer. User information can also be stored locally in a limited capacity so that functionality remains even when there is no connection.

Expected Users and User-needs

Since the type of event is left ambiguous, this application can serve a variety of users, both casual and business oriented. With the log-in feature businesses can potentially set up business accounts that link all users in a team to one event calendar, allowing users to track team events. Casual users may find the application a better alternative to the Google Keep and Google Calendar app, as it will be designed to be a simplified version of both applications, combined into one. Secondary and other higher education schools would find the app useful for classrooms to manage event tracking, or administrators to keep teachers' events tracked.

Various users may have one, some, all, of the following goals in mind when creating an account to use this application:

- Teams may need to track events within a project
- Casual users would track personal events and goals
- Business users can disseminated event information to their teams and workers through syncing with those users' accounts
- Teachers can set events for due dates, students can set event reminders for themselves about deadlines
- One single user may use one account to track both personal and business events, or create multiple accounts to switch between

In order to achieve the above goals, the application will need to provide users with an easy to use interface that allows quickly adding and removing events, editing apps, and sharing their calendar of events with other users. The application will also provide an intuitive interface that just about anyone can navigate. The interface will be simple, not cluttered or overly complex. Actions that are common in other applications will be present in this application (swiping up or away, clicking to open more detailed views, floatingActionButton for adding items, etc). Finally, the application will have options to quickly create, use, and switch accounts for single users, and allow easy sharing between users.

Features and Screens

The first thing a user should feel when they open the application is comfort in the simplicity of the application's presentation. Users have grown accustomed to performing certain actions within mobile applications, and that standard will be reflected in how they navigate this application.

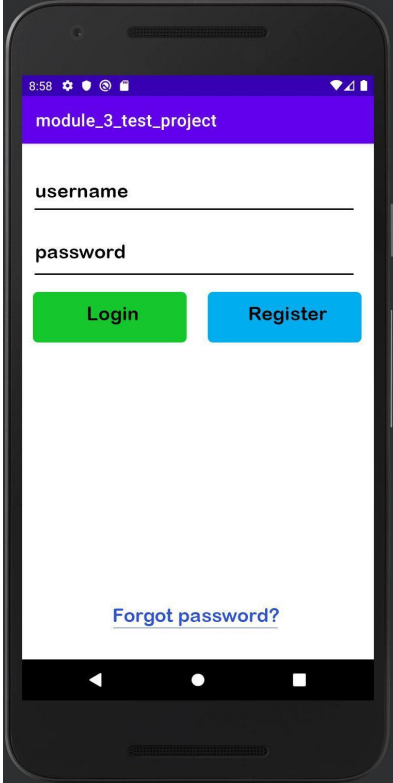
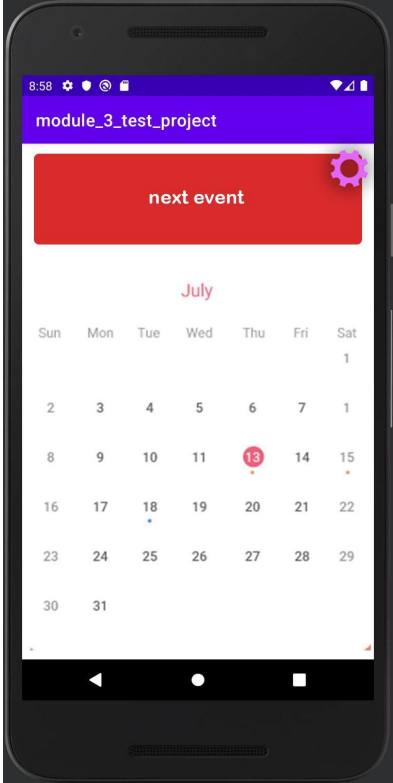
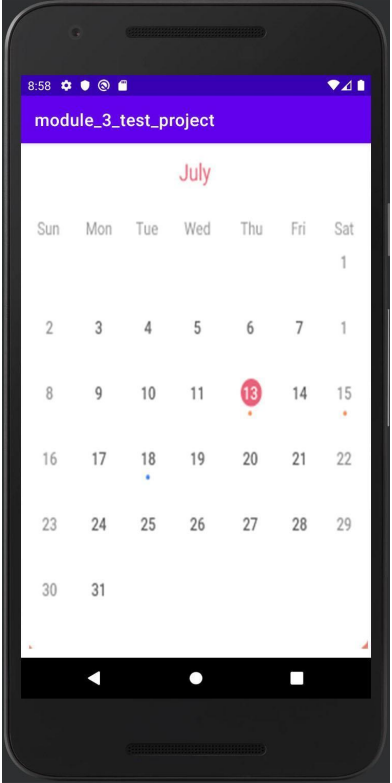
The initial login screen (figure 1, below) will offer only three options: to log-in, register as a new user, or recover a lost password. Registration will require users to enter some PII, but not much (username and email address, as well as choosing a password). Once a user has registered, they may log-in and use the service.

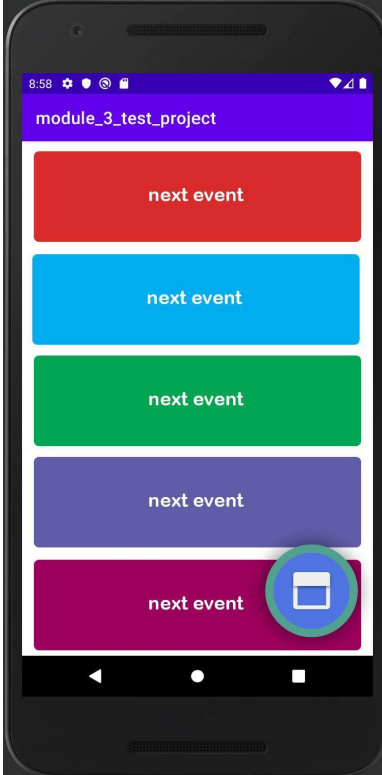
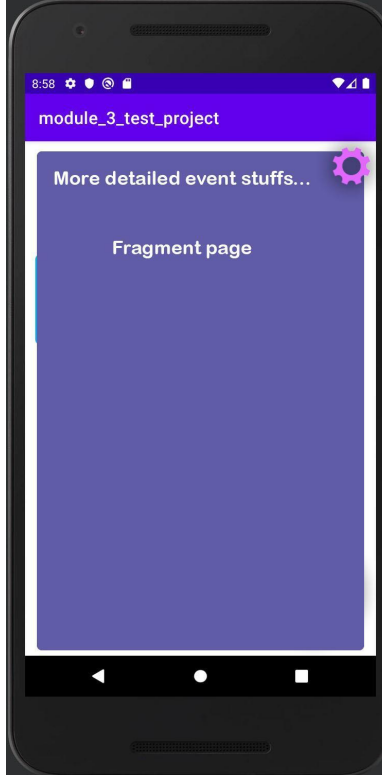
The application's main page default (figure 2) will show a calendar widget of events for the current month, filling up the lower 75% of the screen. The upper 25% will show the next scheduled upcoming event. Should the user swipe up on the calendar, the calendar will expand to fill the screen (figure 3), covering up the next event at the top. Swiping down will return the

user to the default view (figure 2). Should the user click a date on the calendar, a view of events scheduled for that date will appear (similar to figure 4).

On the main page view, if a user swipes down on the calendar, the calendar widget will compress itself to a small round action button in the lower right of the screen, and the remainder of the screen will be taken up by a list of upcoming events (figure 4). The animation for this transition will be smooth, and the topmost event will remain in place to give the impression that the calendar widget was simply a layer on top of the list. The list can be scrolled up (events further in the future) or down (past events), and events can be clicked on to view more details or edit (figure 5). Clicking the action button at the lower right returns the user to the default view of the calendar on the main page.

Sample Screens

 The login screen features a purple header with the text 'module_3_test_project'. Below the header are two input fields labeled 'username' and 'password'. Under the 'password' field are two buttons: a green 'Login' button and a blue 'Register' button. At the bottom of the screen is a blue link that says 'Forgot password?'. The screen is framed by a black border representing a mobile device.	 The main screen has a purple header with 'module_3_test_project'. Below the header is a red rectangular widget with the text 'next event' and a small gear icon in its top right corner. Below this widget is a calendar for the month of July. The calendar shows days of the week (Sun to Sat) and dates. The date '13' is highlighted with a red circle. The screen is framed by a black border representing a mobile device.	 This screen shows the main interface with the calendar widget expanded. It has a purple header with 'module_3_test_project'. Below the header is a full calendar for July. The days of the week are listed at the top, and dates are arranged in a grid. The date '13' is highlighted with a red circle. The screen is framed by a black border representing a mobile device.
Figure 1: Login Screen	Figure 2: Main Screen	Figure 3: Main Screen with expanded calendar widget

		<p>blank</p>
<p>Figure 4: Calendar minimized to action button and event list expanded</p>	<p>Figure 5: Single event expanded for editing purposes</p>	<p>blank</p>

User Interface Connections and Data Flow

To keep local data to a minimum, user information and event tracking should be stored in a database. Considering this, the application will need to be able to connect to the internet to retrieve the data, read the data, and load the data into the appropriate page view. User data (username, password, and PII) will reside in a separate database from user events to enhance security.

When the user opens the application, a small cache of locally stored event data should be loaded first, but in the background the updated event data will be fetched, compared with what is stored locally, and (if newer) replace the local content both in memory and in the app.

User event data will be entered by users in the event fragments. A floating action button (not pictured above) will exist to add a new event from the main page view. Clicking this button will open a new fragment page with the appropriate text-data entry points for the user to fill out (event name and day / time to fire the event). This fragment page will create a temporary event that will become permanent once the user finalizes their inputs. To minimize data retention issues and development on the UI, the application will, for now, operate only in portrait mode. Once Landscape mode is introduced, though, the “bundle” class should be implemented to retain user data.

All data that is displayed in the application will be available on a database. Once a user creates an event, that event's details will be added to the local database for caching, and that database will be uploaded and merged with the database in the cloud.