**Project 1: Mobile2App Event Calendar Development**

James Porter

Department of Computer Science, Southern New Hampshire University

CS 360 Mobile Architect & Programming

Professor Jerome DiMarzio

January 23, 2022

Mobile2App has a client who needs an app developed to track events and display the information so the user can better keep their schedule straight. The client is also requested to have user logins that can be used to add and remove events. One thing that might be useful to the client would be to allow only the user who created the event access to remove that event. The client has also requested that a grid be used to display the events that have been created. I think that this would be best done using a calendar layout. Each grid would represent a day in the month with the ability to toggle between which month and year is currently displayed. Lastly, the client wants the user to be prompted on their first login to setup a new account. After logging in the for the first time, the user would be able to login to the app like normal.

There are a lot of possible users for this app, and I have the benefit of being a prime candidate for using this app. Being a mobile DJ, it is beneficial to have a calendar of events to make sure that you are never double booking yourself, and I use calendars myself all the time. This wouldn’t limit this app just to mobile DJ’s but could be used by any individual working in the entertainment industry. Another group that would benefit from an event services app would be professionals in an office setting. People who work in an office would benefit from having an event scheduling app by being able to better keep track of their schedule. One last group that would benefit from this app would be parents, especially those who have school aged children. This app would allow these parents to better keep track with the events that they have going on and when they would have to pick up their kids from school, work, or other after school activities.

Although these three groups may seem very different, they all have some things in common. All three groups have schedules that get full quickly and they need to make sure that they can keep the events straight. These groups also don’t always have tons of time to spend organizing their schedule. Depending on who this app is specifically being developed for, small changes in login would need to be made, but the overall functionality would remain the same. Lastly, I would like this app developed so that it would be useful for keeping track of events but take very little time to search for and add new events.

There are multiple screens that would be necessary to accomplish this app and make it as beneficial to the user as possible. The first and easiest choice for a screen would be a login screen. This screen would consist of a username and password EditText objects that would allow the user to input their individual username and password. The last object on the first screen would be a button that would let the user login and see the next screen. After the user gets logged in, they would be directed to the second screen which would be the calendar screen.

The calendar screen is the most complex screen of the entire project. The default display would show a calendar with the current month being displayed. This calendar would have each day represented by a grid square; if the grid is a different color, the user has an event that day. The user could then touch a date on the calendar and be sent to a new screen that would show a detailed list of the events of that day; this list would be displayed in a grid form, but only have one column. From this grid, the user could select an individual event that would send them to another screen. From the event screen, they would see all the details of the event along with a button to delete or edit the event. If the user deleted the event, the event will be removed from all calendars. On the other hand, if the user wants to edit the event, one final screen will pop up with details of the event that could be edited by the user. The editor screen is the same screen that is used to add an event (see the next paragraph for more details) just with the event details already filled in.

There are a couple of more functions that can be done from the calendar screen. There would be an option for the user to advance the screen to the next month or previous month by pushing a button to the top left (previous month) or right (next month) of the calendar. A floating action button (FAB) that would allow the user to add an event to their calendar. The FAB would be how the user adds new events to the calendar and brings the user to a new screen the editor screen (see above paragraph). This screen would have fields that the user can edit that would be for the event title, time, date, description, invited participants, and participants attending. The event title, and description would both be represented by an EditText object so the user can just add text. Date and time would be changed using a drop-down list (spinner) that can be used to choose the specifics of the event. Lastly, the participants would be added using a checklist of other users in the system.

Lastly, there are two final buttons on the calendar screen for the settings and requested events. When the user selects the requested events button, the user is shown a list of events where another user has requested their presence at a meeting. From this list the user can use a toggle button to share whether they are going to the event or not. Once the user says they are going, their name is added to the list of attending individuals. The user can also change their mind and say they are not coming after they have said yes. The settings screen allows the user to change their password, change the colors of their calendar, and logout. All these options are represented by buttons when each button is pushed the appropriate action is completed. If the user selects the change color button the user is sent to a screen with radio buttons for each color. When the user wants to change their password, they can select that button and be redirected to the change password screen. From here they must retype their current password and then type in their new password and confirm the changes. Finally, if the user wants to logout, they can select the logout button and be sent back to the login screen.

I understand this has been all a little confusing so I have attached low level mockup images of each screen that the user can navigate to.

**Login Screen**

**Graphical user interface, application

Description automatically generated**

**Calendar Screen**

**Calendar

Description automatically generatedDay’s Events Screen**

**Graphical user interface

Description automatically generated**

**Event Editor Screen**

**Graphical user interface, application, table

Description automatically generated**

**Settings Screen**

**Graphical user interface, application

Description automatically generated**

**Color Screen**

**Graphical user interface, application, table

Description automatically generated**

**Change Password Screen**

**Graphical user interface, application, Word

Description automatically generated**