**New App Development**

Lucy Mills

Colorado State University, Global

CSC475 - Platform-Based Development

Luis Gonzalez

April 9, 2023

**New App Development**

Over the past several projects studying Android mobile app development with Android Studio, I had the opportunity to work with various types of technology to enhance the user experience in apps. Expanding upon the project I created for the Critical Thinking 6 assignment, I wanted my portfolio project to be a usable app for EB Outdoors utilizing multiple activities, instances of fragments, and external links to the existing website (Brown, 2022). The main activity serves as the home page for the app and displays buttons for navigating to the other activity pages with activity two being for the company’s mission, activity three being for the programming, and the final activity includes information for the sponsorship program. The fragment used in each one of the pages beyond the home page, HomeFragment, includes a small image view object with the company logo and a grey button labeled ‘home’ that brings the user back to the home page. Additionally, each activity includes a description of the page from the website and a button with the appropriate link to the webpage. Each external URL link is defined in both the strings XML file as an href and in the AndroidManifest XML file as data labeled with a scheme, https, a host, the appropriate webpage link, and path prefix, eboutdoors in the corresponding activity tab (TutorialsPoint, 2023). Also, Toast messages are sent to the user upon opening each activity; a welcome message displays on the home page and a message directing the user to click a button to visit the website is sent on all the other activities (Android, 2022).

**Overview of steps**

Beginning with the activity\_main XML file, I updated the user interface to make this page serve as the home screen for the application. Organized in a linear layout, the top of the activity holds a large text view object with the string “EB Outdoors App: Home Page” and a large image view object to display the company logo in the middle of the screen. Next, three buttons are added for navigating to the other activities in the app: mission, programming, and navigation. A layout margin of 15dp is added to the components in this activity to give them spacing and have the objects fill up the screen. Within the onCreate function in the main activity Java file, the Toast welcome message is sent to the user via the makeText function, and the listeners for the mission, program, and sponsor buttons are set to launch the other activities on click through using intents and the startActivity function (Android, 2022).

For the fragment used through the app, I created the fragment\_layout XML file to define the structure of the components in the fragment. I defined the framework of the app to include the context of the HomeFragment Java file with appropriate padding in a linear layout. It starts with a small image view object containing the company logo, drawable eb, oriented to the left and a lightly colored button object labeled with the text ‘Home’ oriented to the right. The HomeFragment class extends the Fragment class to include the onCreateView function. After using an inflater on the layout, the home button click listener is defined to start the main activity class after being pressed and the view is returned (Abhi, 2017). This fragment was extremely useful to me as it reduced redundancy in my code by allowing me to use the same objects for several different activities throughout the application. Incorporating the fragment into the three other activities proved to be very simple, just like adding an ordinary object into the user interface in the XML files (CodePath, 2019).

To ensure that the various activities were properly included in the app, I had to work in multiple files to accurately define all the elements. In the AndroidManifest XML file, each activity is defined with the name written out in its entirety, for example com.deitel.welcome.ActivityTwo, and the parent name written the same way to ensure the paths are properly connected. In the intent-filter, the categories for the main activity are launcher and browsable, and the category for the other activities is set to default and data fields are included for holding information portraying to each link used in app (Google, 2017). Additionally, all Java files in the project are wrapped in the project ‘com.deitel.welcome’ at the beginning of the code. Lastly, the additional activity Java classes extend the Activity class rather than the AppCompatActivity like the main does.

The XML files for activity two, three, and four are all pretty similar to make EB Outdoors app feel cohesive with identical background colors and layouts. The text view at the top of the activity displays the title for the page, Mission, Programming, or Sponsorship Program, respectively. Next, the description for each page was taken from the website and saved into the strings XML file (Brown, 2022). Then, a button is placed on each activity with the corresponding link to the website embedded. Finally, the fragment, fragment\_layout, is used on each page for displaying the logo in the lower left corner and providing the user with a way back to the home screen with a button in the bottom right corner.

Similarly, the Java files for activity two, three, and four are nearly identical as each page functions the same. The class begins by extending the Activity class, and in the onCreate function a Toast message is sent to the user with the text ‘click the button to be redirected to link’ (Android, 2022). Then, the buttons’ on click listeners are set up to launch the link at the provided URL upon click with the setMovementMethod and LinkMovementMethod (TutorialsPoint, 2023).

Overall, I enjoyed completing this project as I feel it shows many of the applicable topics about Android applications that we covered this term. The thought of making an app with multiple activities intimidated me at first, but now that I was able to make a working project through this assignment, I feel much more confident with the topic. Furthermore, the use of fragments proved to make coding all of the activities much faster and less complicated. The Toast messages prove to be a nice finishing touch to the app that makes the users feel more connected to the app, personifying it in a way.

**SQLite Databases**

As my EB Outdoors app didn’t have the need for a database, I decided to include a refresher on the topic for the portfolio. Android developers can utilize the SQLite database library by extending the SQLiteOpenHelper class in the project’s main Java file for storing and manipulating data pertaining to the app (Chugh, 2022). A table query is created with a table name and other specific data for each entry i.e., types for each column in the table. A dataBaseHelper is created and the onUpgrade function is overridden to call the onCreate function and create the table. Next, an additional Java class has to be made for the database, usually call the database manager or DBManager (Chugh, 2022). Within this class, a DatabaseHelper and SQLitedatabase are created and functions for altering the database are defined for inserting, updating, and deleting entries based off various criteria. With these files properly setup, the user of the app can easily manipulate the data being saved in the database table to best serve their needs.

**References**

Abhi Android. (2017). *Fragment Tutorial With Example In Android Studio*. <https://abhiandroid.com/ui/fragment>

Android Developers. (2022). *Toasts overview*. <https://developer.android.com/guide/topics/ui/notifiers/toasts>

Brown, E. (2022). *EB Outdoors*. EB Outdoors. <https://www.eboutdoors.com/>

Chugh, A. (2022, August 3). *Android SQLite Database Example Tutorial*. DigitalOcean. <https://www.digitalocean.com/community/tutorials/android-sqlite-database-example-tutorial>

CodePath. (2019). *Creating and Using Fragments*. CodePath Android Cliffnotes. <https://guides.codepath.com/android/creating-and-using-fragments>

Google Developer Training. (2017). *2.1: Create and Start Activities*. Github. <https://google-developer-training.github.io/android-developer-fundamentals-course-practicals/en/Unit%201/21_p_create_and_start_activities.html>

McCown, F. (2021). *Mobile App Development with Android and Java.* zyBooks. <https://www.zybooks.com/catalog/mobile-app-development/>

TutorialsPoint. (2023). *How to create clickable links in a textView on Android*. <https://www.tutorialspoint.com/how-to-create-clickable-links-in-a-textview-on-android>