DOIT: AN ANDROID APPLICATION

Andres Acevedo

MAY 15, 2020

CSC400

Andres Acevedo
5/14/20

DoIt – Final Report

Introduction and project inception

DoIt is an organizational Android application targeted toward college students. This application aims to improve productivity and organizational habits of the users through the implementation of several features. These features include allowing users to set reminders for themselves, create and edit textual notes, and create quizzes which they can use to study for subjects in their college courses. Although organizational applications already exist, most of them are targeted toward professional individuals, and not college students. DoIt has the advantage of combining the features from those existing applications and making them accessible all in one place. Users will not need to switch between different applications in order to complete their intended tasks. The user interface involves a minimalist design and will place an emphasis on being user-friendly, as opposed to containing a wide variety of features.

To explain the application in further detail: DoIt is an Android application, created within Android Studio. It can be accessed by Android phones and tablets running on Android version 6.0 and above, using API 23 and above. The interface for DoIt currently consists of a home screen which contains a main menu. The main menu includes individual tabs corresponding to the main features of the application: “Notes”, “Reminders”, and “Quiz Flash Cards”. Users can navigate to and from the different sections of this Android application easily and with seamless transitions. This application makes use of different tools such as the Google Firebase console, which is used for data storage. Another tool used is the Android AlarmManager library, which is used for the implementation of the “Reminders” feature. This library will allow for the creation of timers and alarms and notification associations within a user’s Android mobile device.

Project Implementation

In order to describe DoIt’s features in further detail, we can analyze them one by one. We can use the home screen (figure 1) as an initial reference point. To begin the analysis, we can talk about what exactly the “notes” feature is comprised of. For this feature, users have the ability to create text-only notes. After a user clicks on the “notes” section of the application, a user can then click on the circular “plus” symbol to initiate the creation of a note (figure 2). There will then appear a square canvas in the center of the screen in which users will be able to input any

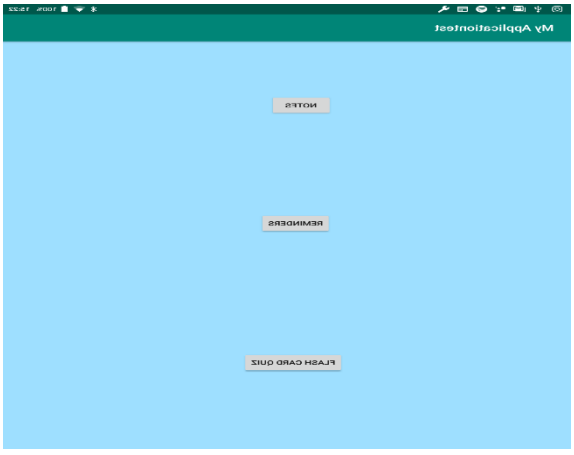


Figure 1:DoIt Home Screen

Figure 2:Note creation screen

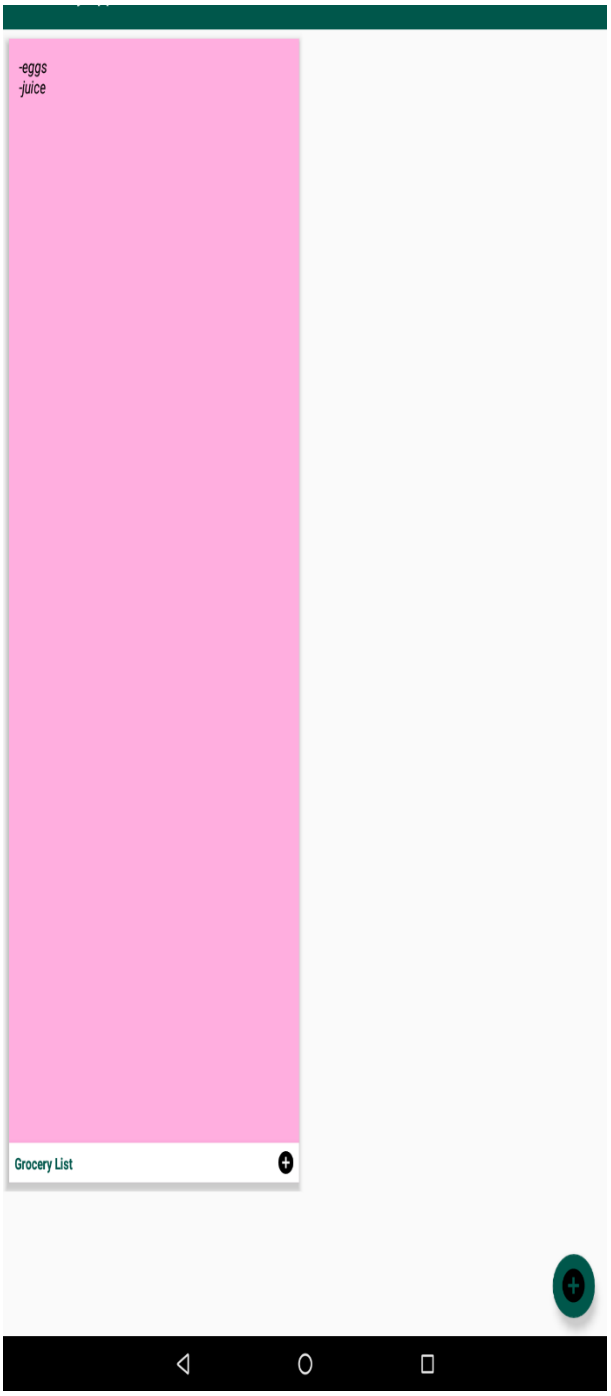
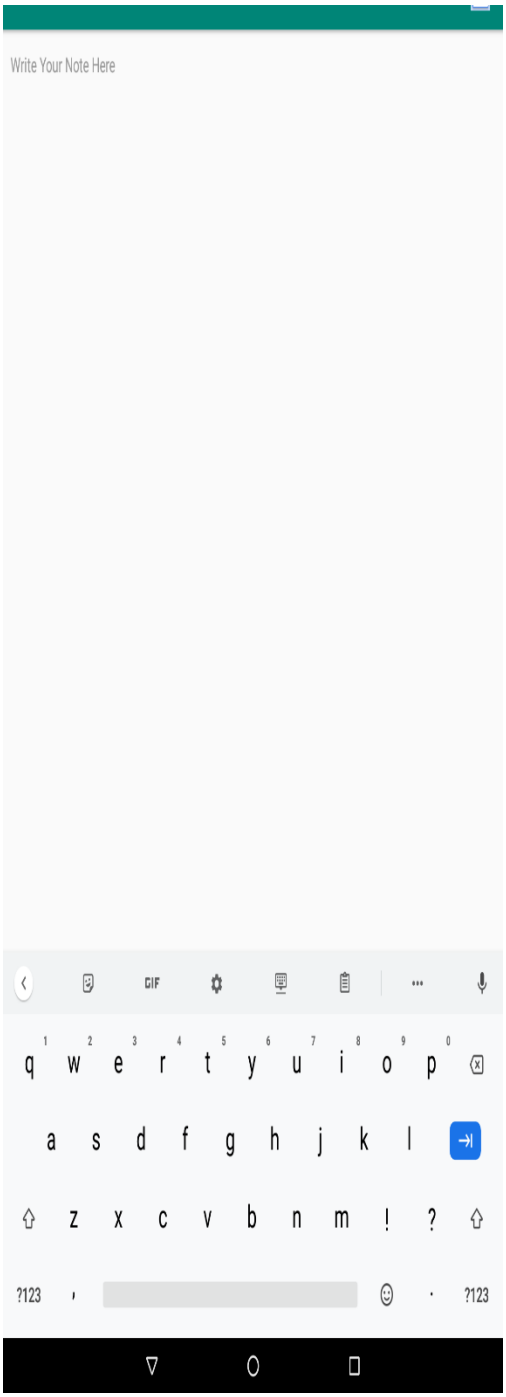
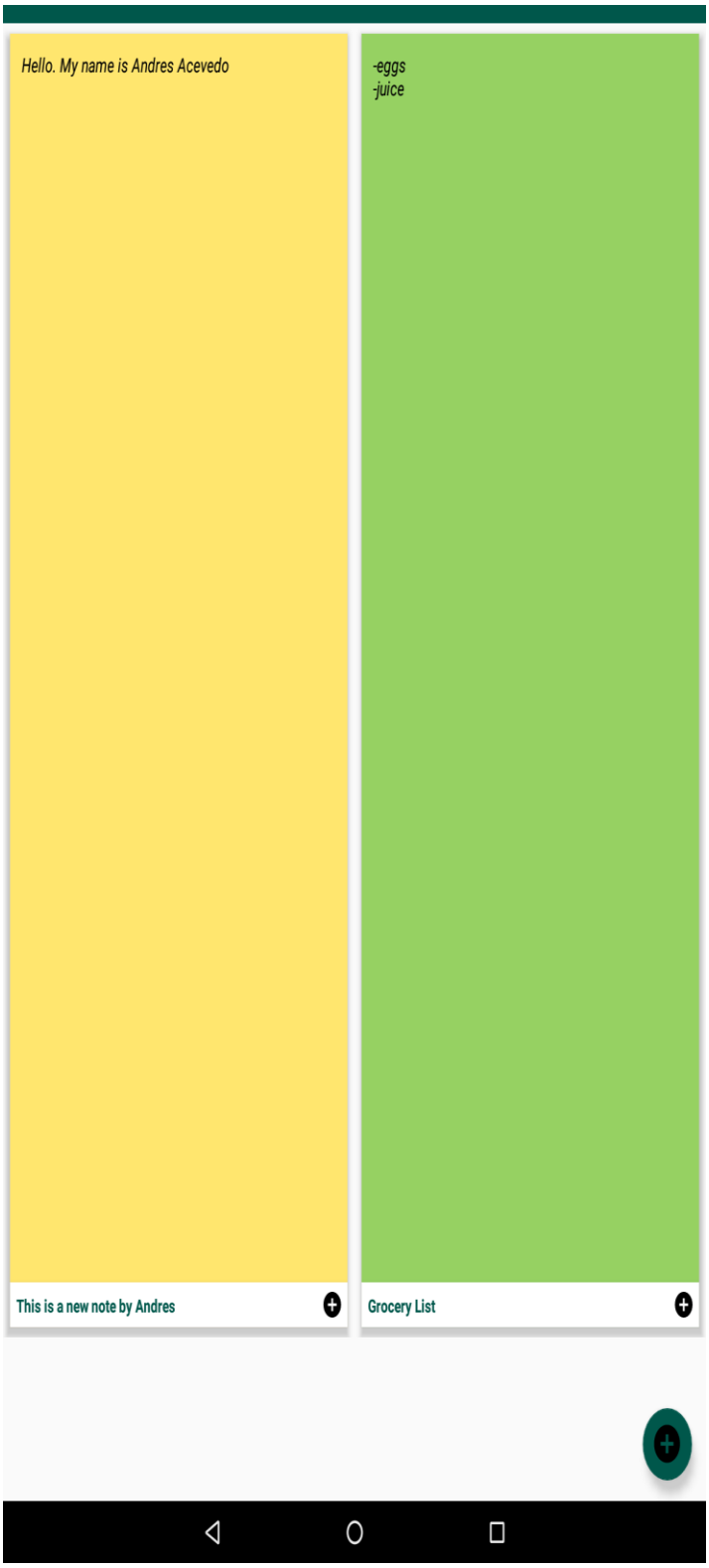


Figure 3:Note editing screen



textual notes they desire (figure 3). Users will be able to input notes without a character limit as long as the total size of the created notes does not exceed 1gb (the free limit for Firebase accounts). Users will then click on the Google symbol on the bottom right corner of the screen to save their notes. The created notes will then be linked to a Firebase account, so that users will not lose their notes. Firebase is a service provided by Google. It is a real-time database

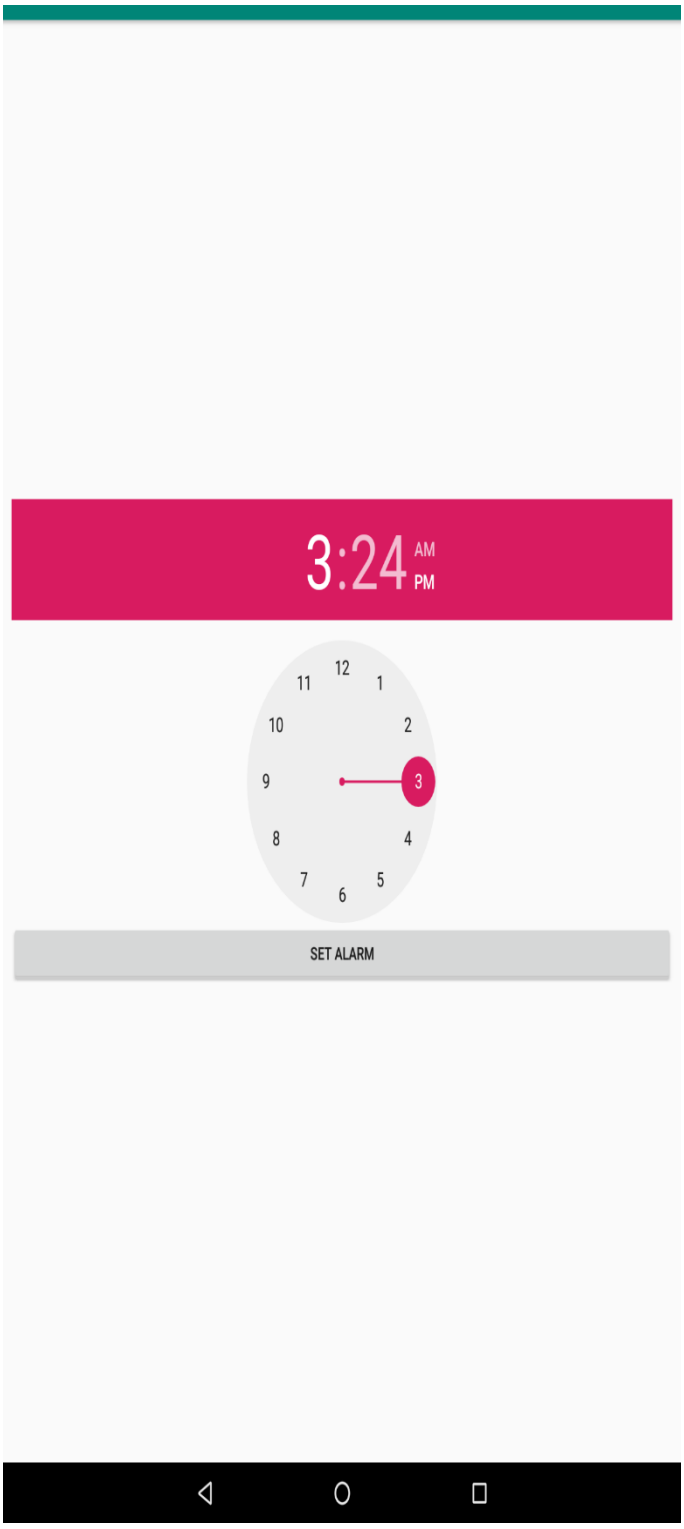
Figure 4: New note



which allows users to save their data to it. Users have the option of saving their data anonymously, with the disadvantage of having their data deleted upon uninstalling the application. When a user creates an account with Firebase, their data will always be linked to the Firebase account. If a user decides to uninstall the application, their notes would not be deleted. They could simply reinstall the application at a later time and still gain access to their previously created notes. Notes are displayed in order of recency, from left to right. This means the newest created notes will show up first (figure 4). The section of code that queries all of the user's current files is as follows: Query query = fStore.collection("notes").

Next, we can talk about the "Reminders" feature. Upon opening the "Reminders" section of the application a clock will be displayed in the center of the screen (figure 5). Users will be able to adjust this clock to match their desired time (AM or PM) for receiving their reminder. There is a "Back" button on the bottom left side of the screen so that users can return to the main menu when they are ready to do so. On the day and time set for the reminder, a user's mobile Android device will play an audible alarm or a

Figure 5: Reminders



vibration, depending on the user's current volume settings. The reminders feature currently allows alarms to repeat at the same time every day, as long as the application is still running in the background. This process is handled in the setAlarm function of the project. This function calls an instance of the MyAlarm class which contains the ringtone sound. If the alarm is set to go off on the current day, even if the user were to close the DoIt application, the alarm would still be activated. As of right now, users can only set one reminder at a time. Future versions of the application would allow for simultaneous reminders to be set and activated within the same period of time.

Next, we can talk about what the "Flash Card" feature was proposed to be comprised of. For this feature, users will be able to input text on a square canvas, in a way which is similar to the notes feature. However, this flash card canvas will be double sided. Users will be able to write text on one side of the canvas, flip the canvas over, and write text on the other side as well. Once a flash card is completed, users will be able to flip the flash card back and forth to their liking. This can be useful for students trying to memorize concepts for a test, because this feature simulates having a physical flash card in front of you.

Andres Acevedo
5/14/20

Testing and Evaluation

In order to test my final project, the first thing I checked was to make sure that there were no apparent error messages in any of my Java classes. I noticed that I had some warning messages for some undeclared strings corresponding to my buttons. However, this is not a big deal and does not affect the functionality of the application. The next thing I checked was to make sure that each and every button successfully completed its intended action upon click on it. Once I verified that all the buttons did what I intended, I checked certain things in the individual features of the application. For example, I checked the input and storing limit of notes in the "Notes" feature. What I mean by this is testing the character limit and testing the size limit of the notes. For the character limit, there are no apparent constraints. The canvas in which the notes are written contains a large space in which many characters can be written. As far as the size constraints, I found that Firebase allows a storage capacity up to 1GB in size for free-tier Firebase accounts.

Lessons learned and Reflection

This project was single-handedly the most challenging university project is have ever taken part in. After taking an online course on the basics of Android development using Java several years ago, I honestly thought completing this project would be a relatively easy feat. It wasn't until I began the implementation stage of my project that I realized the general difficulties of designing mobile applications and the ever-changing Android Studio software. This project has seriously given me a newfound respect and admiration for Android development.

Before I speak on the things that went well during my project, I would first like to speak about some of the many challenges I faced throughout the process of this project. The very first error I encountered while coding the "Notes" section of the application was a "Manifest merger" error in my Gradle file inside Android Studio. The Gradle file is where all dependencies are stored for your project. I was unable to find a resolution to this issue for a while because the various sources I used for troubleshooting Android Studio errors all offered conflicting advice. This caused me a lot of frustration because it was a clear obstacle in getting the Notes feature to work correctly. After a lot of digging around on the internet, I found that this issue was being caused due to having conflicting libraries in my Gradle file. I learned that a dependency I used to integrate my project with Google Firebase was conflicting with Android Studio's stock support library (AppCompat).

The next issue I encountered while working on my project was being able to convert the project into AndroidX. AndroidX is a replacement to the stock support library offered in Android Studio. Initially, I didn't think using the stock library would be an issue at all since this is the default library you get when you first install Android Studio on your computer. However, I soon came to find that the stock support library is mostly outdated, and it is generally recommended

to convert your project to AndroidX. Android Studio does have a dedicated button to convert your project to AndroidX, but there are certain declarations you must also make in your Gradle file to complete the conversion. I learned that the official documentation guide for Android Studio does not explain in a good way how to successfully make this conversion. Solving this issue required me to gather advice from multiple sources.

The most important thing that went awry with my project was being able to successfully implement the Flash Card Quiz feature of the DoIt application. I had a feeling from the very beginning of the project creation process that this would be the most difficult feature. I brainstormed several possibilities for completing this feature, but a lack of time and a lack of useful tutorials online prevented me from implementing this feature.

Now, I can speak on some of the things that went well with my project. The main thing I am proud of is that I was able to get the “Notes” feature of the application to work without any errors. This was the most difficult section of the project for me because it involved integrating the project with the Firebase database and querying the saved notes into my Firebase account. Firebase is a hot topic right now in Android development and I am glad I was able to successfully integrate it with my project. Even though I have only scratched the surface of Firebase, I am excited to see what other ways I can use it to manipulate data.

Another thing that went well with my final project is being able to successfully play a ringtone sound in the “Reminders” section of my application. I was able to successfully call an instance of the Android TimePicker which will allow you to display a clock on the screen which can take time/date input from a user. The application can then perform certain activities when the actual time on the device is equal to the time input by the user on the TimePicker. In the case of my application, when a user sets a time on the TimePicker, the resulting activity is that an alarm “ringtone” will activate once the device time is equal to the user’s set time.

The final thing I would like to talk about that went well is my successful understanding and implementation of button clicks and the activities that these button clicks trigger. I was able to successfully implement a home screen from which a user can click on a button corresponding to a specific activity. In the case of my final project, this involved allowing the user to click on one of the application features such as “Notes”, and the corresponding activity would be opened/triggered depending on which button the user clicked. This was one of my favorite parts of the implementation stage of my final project. This is because it was very satisfying to link a button to a specific activity and then running the application and realizing that the button did exactly what you wanted it to do after being clicked.

If I were to do this project again, there are several things I would do differently. I would work extremely hard to improve my time management skills and meet deadlines. My initial proposal was considerably more ambitious and sophisticated than the actual implemented design of the project. For example, I honestly believe that having better time management would have allowed me to successfully develop the “Flash Card Quiz” feature in the way I

originally intended it to be. The resources I used while working on my project, along with my personal knowledge turned out to be insufficient with regards to successfully completing this feature.

Version 2

As I keep working on the DoIt application and further develop its features, I have thought about several features and changes I can implement in a second version of the app. Based on my initial proposal, it is still my intention to eventually implement a flash card feature in the application which will allow users to store sets of flash cards, which they could then shuffle and use to test themselves on vocabulary and concepts relating to their university courses. Although my original idea involved the user creating their own quiz, an alternative idea I could pursue would involve somehow gathering a large amount of existing quiz questions on a variety of subjects taught in school. I would find a way to integrate all these quiz questions into my app to be accessed by users. Users would be able to choose their desired subject and take a pre-existing quiz based on that subject. This is another ambitious idea, but it is still interesting to think about implementing something like this in a future version of the application.

For the "Notes" feature of the application, I would like to eventually add the ability for users to create voice memos in addition to textual notes. With regards to memorizing a topic, sometimes students might benefit from hearing their note instead of just reading it. Another possibility would involve giving users drawing tools from which they could draw out their notes. They could use these drawing tools to create diagrams and other figures. Lastly, I would like to add FireBase sign-up and login capabilities to the application. As previously stated, the Notes feature only allows for anonymous users to create notes. No account creation is currently required. However, I would like to give users the option to create a FireBase account so that even when they delete the DoIt application, they can still have their created notes linked to their account. If they choose to re-download the application, they will regain access to the notes.

The last thing I would work on in future versions of the application involves a dramatic user interface overhaul. I am not much of an artistic person, so I would take the time to learn about colors and what interface features work best on mobile devices so that users can have the most pleasant experience possible while using the application. I previously took a human-computer interaction course in which we learned about the concept of interfaces and the experience a user has with any given system. However, we didn't go into too much detail about the artistic aspects of the interfaces. I personally know what kind of designs I like to see in systems I interact with. For example, I really admire the designs of Apple products such as the iPhone. I really like the sleekness and simplicity of the interface, and how user-friendly it is. Lastly, I want to learn more about how to make your interface layout work well with any devices, regardless of screen size. In my experience of working on websites, I have found that it

is relatively simple to create a website layout that can work well with screens and devices of varying shapes and sizes. However, I found that it was not so easy for me to accomplish this feat while testing my application. The layout I designed for my application truly works best on the device I originally used for testing purposes: A Lenovo Nexus tablet.

Professional development and lifelong learning

As I reflect on the process for developing skills and new knowledge to achieve project goals, I can't help but compare it to my collegiate experience as a whole. When I first entered college, it was quite difficult getting accustomed to the rhythm of my courses. I quickly found that college was a lot more fast-paced than high school and the courses I was taking were considerably more challenging. As I moved forward and progressed through college, I was forced to adapt and develop new skills and new knowledge in order to complete my college courses. As I gained more experience as a college student, my methods of gaining knowledge improved as did my overall qualities as a student. Once I figured out how one studies and how one prepares to pass a college course, I felt more comfortable when faced with the increasingly challenging courses involved in my Computer Science program.

My college experience relates to the process of developing my final project because I also had to adjust and get accustomed to the pace of this project. Knowing that this capstone course was the final step to completing my collegiate career invoked a lot of pressure and frustration for me. I had many setbacks and shortcomings during the development process which tested my full capabilities as a student. The turning point in my frustration began when I was able to do something in my project as simple as launching a new activity when a button is clicked without getting a dreaded error message. Being able to do this gave me a sense of hope, because it showed me that I could do something right and it showed me that I could adapt. As I progressed on this final project, I would often think about the long journey it took to get to this point, and the adversity I faced both in and out of college. These are the things that motivated me to learn the skills I lacked.

In my experience of working on this final project, I had to use multiple resources. These resources ranged from the Android official documentation, to YouTube and Udemy tutorials, to written tutorials. I needed all these resources to implement the specific features required in my project. To learn about Firebase, I had some success looking at YouTube tutorials as there is a growing community of students and developers dedicated to teaching and learning about Firebase. I used various written tutorials to learn about Gradle files and resolving conflicting dependencies, along with converting a project from the stock support library to AndroidX. I also took a look at many YouTube tutorials on the various visual layouts available in Android Studio.

Working on this project has really shown me just how popular Android development is, especially with people who live outside of the United States. In fact, it was sometimes hard to follow along with Android tutorials because most of the video creators often spoke with heavily

accented English. This actually makes me consider making YouTube tutorials of my own at some point. I would like to make tutorials on the same concepts I had trouble with while working on this project, in case other people are experiencing the same issues. I would even consider making tutorials in my native language (Spanish).

Overall, this incredibly challenging experience has taught me a lot. I am grateful for the many opportunities I was given to complete this project, and I am satisfied with the new skills I have learned as a result of working on it.