****

**Group Members:**

* Steven Tong
* Justin Meville
* Lifu Zhang
* Zhaoze Wang
* Jack Halberian
* Paul Stephen Hutchinson Maltaghati

**Objective:**

The objective of Scintaxx is an app that provides users with an app that combines different messaging/social media apps into one app. The purpose of our app is to not only provide convenience but to also maximize the space users have in their electronics today.

**Scintaxx:**

In developing the Scintaxx app, we used multiple APIs in order to create an all around messaging app. For the email section of our app, we utilized the JavaMail API. With this API, we are able to write emails within our app provided that a user has a valid email account. Using JavaMail, however, we are unable to read emails through our app. We also were able to connect our app to Facebook using a Facebook developers API. With this API, a user can log into Facebook in Scintaxx, and then proceed to scroll through their feed and like various posts. They cannot, however, access Facebook Messenger since the API is limited. Finally, in order to send SMS messages through our app, we used the SMS Verification APIs for Android. By using this API, we were able to send text messages to authentic numbers.

Within our app, we make use of many different Android Studio activity pages. In order to add functionality to these pages and in order to transition between different activities, we make use of both Kotlin and Java, the two programming languages which Android Studio makes its use of. Within the Kotlin and Java code, we use a variety of classes and Android Studio specific functions in order to make our code operate properly.

At the time of submitting this project documentation, our group is still working on trying to integrate notifications and contacts within our app. They are not currently functions within our app, but they are the main features which we will be trying to implement within the next couple of days.

**Front End:**

In order to create the front end portion of our messaging app, we use XML files in order to create the layout of different portion of our app. The XML files are built into Android Studio, as they are a part of each activity made through the environment. We also made use of online design apps in order to make the different logos which appear on the welcoming pages of our app.

**Back End:**

For developing the back end of our app, we used both Kotlin and Java. These two higher level languages are built into the Android Studio environment. Both Kotlin and Java are object oriented, and we make use of the classes in order to generate the functions in our app. Another major part of developing our back end was the implementation of APIs. As mentioned before, the APIs we made use of were the Facebook developer API, the JavaMail API, and the SMS Verification APIs.

**Our Challenges:**

Throughout this entire project, we came across several difficulties when making our app. One of the main ones was learning Java, Kotlin, and how to use Android Studio. Despite the fact that Java and C++ are similar, learning the syntax took longer than we anticipated. This is simply because we constantly had to look up the syntax when writing our code. The crash courses that were provided on YouTube only covered the basics of Java which did not really help us in the construction of our app. Also, the conversion from Java to Kotlin and Kotlin to Java did take quite a bit of time to get used to since many of us were not familiar with either of those languages. On top of these struggles, finding an open source API for messaging/social media platforms were quite difficult. Many of the apps were either not disclosing enough information about their APIs or their APIs required a very sophisticated back end that we did not have the technical abilities to create or did not have the time to create.

Another struggle we had in this project was combining all the features into our app. We found that it is much more efficient if we created each feature separately then piece it together at the end. However, we found it quite difficult for two main reasons. One is that due to the fact that we were still in the process of familiarizing/learning Java, Kotlin and Android Studio, we had difficult writing modular code that worked well with others. The second reason simply came down to the coding style of each member. Sometimes, it took some time to figure out why another member wrote a certain function, variable or parameter.

Lastly, a struggle we had while developing our app came down to a particular feature that we wanted to add, the notifications feature. We were able to create a button which transitions into a page which shows the notifications from all of the different messaging apps. The button utilized Java code in order to make the transition. When we ran the app, however, whenever we clicked on the button, the app would crash. In the next few days, we will be attempting to fix this transition, and it needed, the notification feature will be removed from Scintaxx.