**Email Helper App Documentation**

**Overview:**

Most of the overview and information about the goals for the app can be found in the Email Helper app design pdf.

**Outside Features:**

This app was implemented using Javamail, which offers many features for searching, sending, and receiving emails using the Java programming language. It also uses SQLite databases to store the emails in the app private data.

**Main Features:**

* The app is able to make connections using IMAP to Google Mail accounts to retrieve emails, both when the app is opened and with a “pull to refresh” feature on the ConversationFragment
* The app can send emails using SMTP with Google Mail accounts
* The app stores the connection data in the Shared Preferences of the app, so that they are persistent.
* The user is able to change their email address to a different Google Mail account
* The user is able to “subscribe” to email addresses and only receive emails from those subscribed accounts.
* The user is able to see their email conversations similar to an instant messaging app, where messages are organized by email address in their own fragments.
* The user is able to send an email using the SMS style messaging at the bottom of the conversation list, and these messages will show up on the screen and save to the database.

**Future TODOs:**

Currently, there is no sort of polling or push notifications that reliably works. There is a ServiceActivity class that would sometimes prevent the user from receiving emails on app start up, so it is currently unused.

QR codes still will need to be implemented.

The app cannot currently show attachments. There is also no PGP or MIME decryption implemented.

Currently the app can only work with Google Mail accounts. For other accounts to work, either the user will need to specify an IMAP host, or the app will need to recognize the email and use the correct host.

Other QoL improvements such as better feedback for deleting messages, a cleaner contacts screen, more settings pages, etc.

**Issues/Bugs:**

The biggest issue was getting polling to work. I could not get a background service to connect to Javamail, and a foreground service would cause the app to not get emails at all occasionally. I attempted to implement Firebase Cloud Functions, but that would require a server side setup. Firebase looks like the best option for push notifications, if that ends up being the route we choose to go over polling.

I had tried to implement a MessageCountChangedListener in the MainActivity, but that would cause the “pull to refresh” to be unable to make a connection and find new emails. I think this is because they were running on the same thread and it would cause interference.