**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.
* Notifications that display when the app is not in use or display as a Toast when the app is in use. The notifications are configured inside the app.
* The User can display a personal QR code or scan others to add them to the subscription list

**Future TODOs:**

The app cannot currently show all attachments.

There is no PGP or MIME decryption implemented.

Make secure… Google Blocked my account because our app is not as secure. This is due to the fact that when we access Gmail is it through email and password not OAuth2.0. If we implemented it it would be usable to the standard google account. When enabling the app to use OAuth we will need a client ID and possibly a client secret which is gotten from Google when you register the app with them. Contents will need to be added to the AuthenticationClass. I added some commented out variables into AuthenticationClass for this in the future. Hopefully, this will be able to be wrapped into a class/module.

**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.

-Notifications now work on a timer system. Push notifications don’t, however the user can specify for the app to check for new emails every minute if that is desired.

**Ideas/Suggestions/Recommendations ( BY DSHADE Summer 2018)**

**Ideas and Suggestions**

**To be implemented**

**Not there yet**

**-Sounds with Notifications**

There is a bug in the android code for this

**-Attachments**

Currently, some photo attachments are taken care of. Other attachments are just downloaded.

-**Sent folder**

Pulling from the sent folder to add to the other emails in the chat would be a great thing, for the messaging app.

**-Encryption**

Encryption is not currently handled in the app

**Implemented**

**-Longer emails - IMPLEMENTED**

Currently all emails got straight into the database and there is no limit to how long. This could be painful, to need to scroll past a very long email.

**-Change sent\_by\_me from bool to int**

The idea behind this is that we can add system messages like a time stamp to new messages in a third category that the app places in the center

**-Advanced Login**

The idea here is a button/checkbox in the login screen that allows the user to specify IMAP and POP server names that are not implemented in the main section.

**Recommendations**

-Rebrand EmailHelper to EmailToText

* I think this will give better visibility and slogans and emphasis on what the app does.

|  |  |  |
| --- | --- | --- |
| API | Cumlative distribution | Note |
| 10 | 0.3% |  |
| 15 | 0.7% |  |
| 16 | 2.2% | Current Lowest excluding 0.7% of devices |
| 17 | 4.4% |  |
| 18 | 5.0% |  |
| 19 | 15.3% | New Lowest excluding only 5% of devices. |
| 21 | 20.1% |  |
| 22 | 37.7% |  |

-Update lowest API from 16 to 19. View chart below for details