“The Email Helper” Android App

1. **Goals of this App**
   - (a) To create an app that would have wider appeal to users whether in the mission community or not.
   - (b) To help missionary supporters be more aware of the communications from the missionaries they support, given today's utilitarian nature of email (vs its use as a communication tool).
2. **Design / Capabilities**
   - (a) Connection:
     - i. The app should allow the user to enter in the connection information for their email account (SSL-based imap server address, optional IMAP port, username, password). Only allow SSL-based imap, not unencrypted IMAP.
     - ii. The app should store the connection information in app private storage.
     - iii. In the future, the app should support token based authentication, like oauth2, so the password does not need to be stored on the device.
     - iv. The app should allow the user to change / re-enter the connection information at a later date.
   - (b) Subscription:
     - i. The app should have the user enter in one or more email addresses to whom the user wishes to “subscribe”.
     - ii. The app should allow the user to add/edit/remove items from this list at any point later on.
   - (c) Polling / Updates:
     - i. The app should regularly poll or receive updates from the email account so that it finds out when a new email has arrived.
     - ii. If a new email is from an subscribed-to address, the app should record that email's information in a sqlite database in app private storage on the android device, and should post a notification to the user.
     - iii. The app should allow the user to configure notifications (whether to notify, and whether to use a sound in addition to the notification). Notifications may be configured by subscribed-to address.
   - (d) Viewing:
     - i. The app should allow the user to view email activity by address (not by date). The first activity here is a list of subscribed-to email addresses that the user has added a la 2(b) above. If new but as of yet unviewed (in this app) emails exist, the subscribed-to address should be in bold and indicate how many unread messages there are.
     - ii. Once the user taps on a subscribed-to address, a conversation list of recent communications from that address should show up (most recent at the bottom). The list should scroll to the bottom automatically, just like an instant messaging client. Each item should be a preview of the text of the email, up to a number of lines of text on the screen (perhaps up to 10 lines). Strip out any quoted emails, multiple empty lines, etc.
     - iii. Email attachments should be shown in the same list, as separate items, directly below the email the attachment belongs to.
     - iv. If the user taps on a communication, the full email text should be shown. If the user taps an attachment, it should open up via Android's Intent feature.
     - v. A “Reply” button should be at the bottom of the list. This should use Android Intents to open up an email message to the subscribed-to email address.

vi. Optionally, the app should show Sent folder correspondence with the subscribed-to address intermixed in the list as well, as if the entire conversation were a chat conversation.

(e) Cryptography:

i. The app should have the ability to import a PGP decryption key for emails from a subscribed-to address. The app should register with Android Intents to handle these types of files, so a user can easily import the key.

ii. Decryption keys should be stored in app private storage.

iii. When an email or its attachment is encrypted using PGP or PGP/MIME formats, the app should automatically decrypt the email and/or attachment, seamlessly to the user. The user should notice no visual difference between encrypted and unencrypted emails, and the email or attachment should be shown in the same manner as an unencrypted email or attachment.

iv. Optionally, the app should allow the user to set a pin or passphrase to protect the decryption keys.

(f) Enhancements:

i. Optionally, the app should have a list of all email addresses in the most recent *n* emails in the user's inbox and sent folder, sorted by number of occurrences, and the user can click (checkmark) those that the user wants to subscribe to.

ii. Optionally, the app should have a way of scanning a QR code that contains an email address to subscribe to, plus an optional decryption key. This will make it easy for the missionary.

iii. Optionally, the app should recognize a special metadata file or coding in an email from a subscribed-to address that sets such things as the color scheme, a header image/text/color, and so forth, for the conversation view activity for the particular subscribed-to address.

3. Phasing / Planning

(a) Phase One: UI design, including proposed Android activity layouts for the following screens:

i. Connection Activity

ii. Subscribed-to Addresses Setup Activity

iii. Configuration Activity (notifications, keys, pin/passphrase, etc.)

iv. Configuration of Decryption Key List Activity

v. Subscribed-to Addresses Viewing Activity

vi. Conversation Activity

vii. Message Detail View Activity

(b) Phase Two: IMAP Connection feature

(c) Phase Three: IMAP Polling & data retrieval

(d) Phase Four: Display of conversations and Reply feature

(e) Phase Five: Cryptography feature