Index

GitHub Username: raghuchandan1

Chatify Gmail

Description

Chatify tries to free you from the pain of searching through hundreds of emails looking for important ones. In Chatify you can add an email address and whenever you receive a new email from that sender Chatify notifies you. Chatify also organizes your mail by sender similar to chats making the communication much easier.

Intended User

Any Gmail user

Features

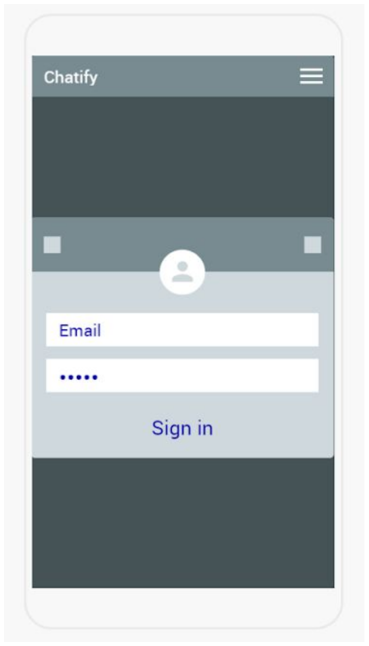
Main features of the app:

- The app will be completely written in Java
- The app uses the latest available stable versions of all the libraries used
- All the strings would placed in the strings.xml file and RTL would be enabled for the required languages
- The app would include content descriptions where ever necessary
- The app uses a SyncAdapter to regularly check for new emails for the registered sender
- Notifies when mail received from sender registered by user
- Organizes mail by sender

User Interface Mocks

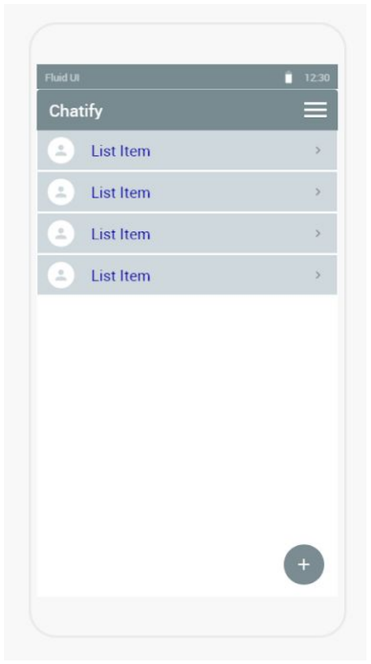
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



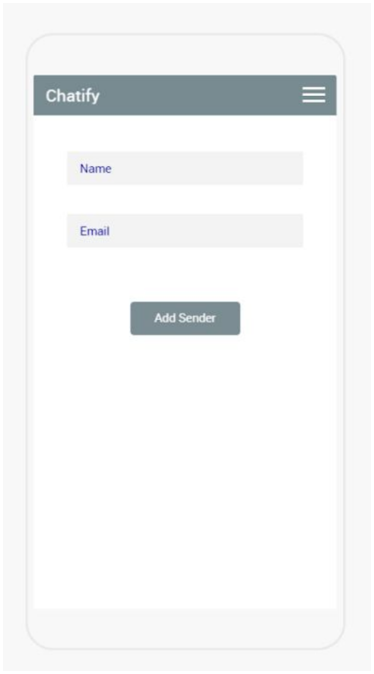
Login page allows a user to login and authenticate with his Google account

Screen 2



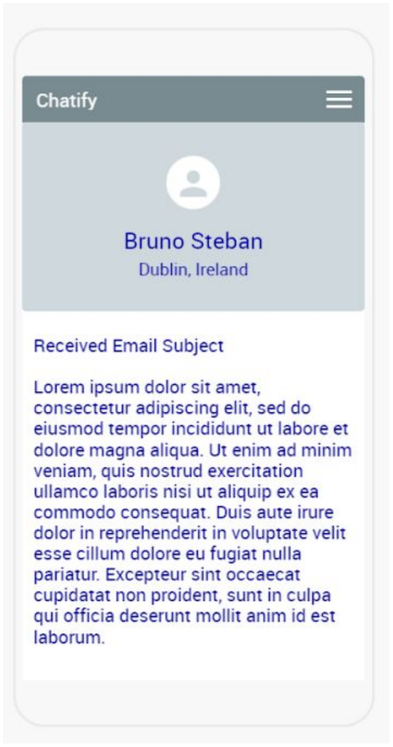
Home screen showing the registered senders

Screen 3



Screen to add a new registered sender

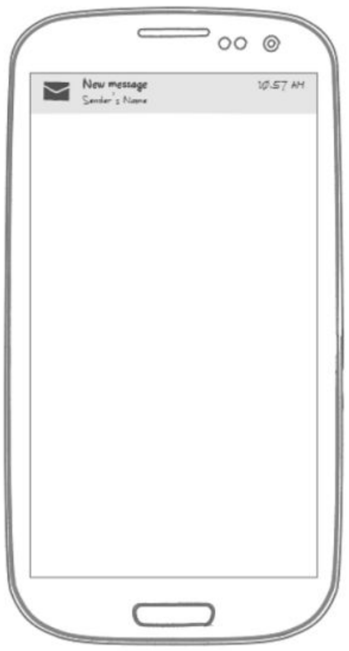
Screen 4



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

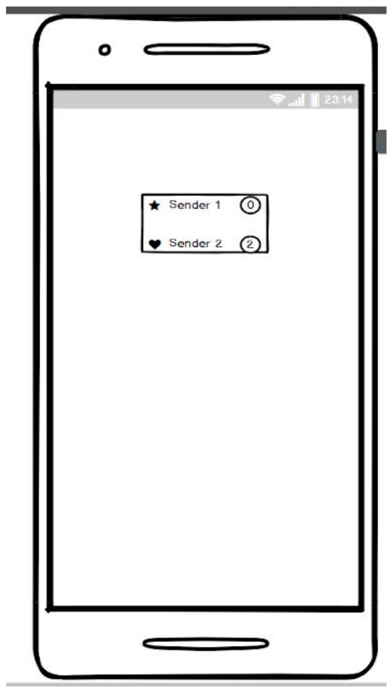
UI showing the new email

Screen 5



Notification when a new message is received

Screen 6



A widget showing the number of new emails from the registered senders

Key Considerations

How will your app handle data persistence?

Stores the messages for a determined amount of time in a local database. It uses the Room Persistence library for storing the data locally using SQLite.

Describe any edge or corner cases in the UX.

Handling app crashes when the user is offline and tries to sync

Describe any libraries you'll be using and share your reasoning for including them.

Gmail API for retrieving the messages

Describe how you will implement Google Play Services or other external services.

The app would make use of Google Sign-in and Google Analytics

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

First the credentials required for the Gmail API are created and stored in the build.gradle. Other required dependencies for the project are added

Task 2: Implement UI for Each Activity and Fragment

Subtasks:

- Build UI for Login page
- Build UI for Home screen
- Build UI for adding a new sender
- Build UI for displaying the unread mails

Task 3: Add functionalities for the above activities

- Make the Login page to check if user is already logged in else display the login form
- Display the count of unread mails for each registered sender in MainActivity
- Make Add Sender Activity to add the sender to the database
- Make Show Mails Activity to read from the database and display the mails correctly

Task 4: Connect the app with the local database using Room library

- Retrieve and store the mails from the web using the JavaMail API and store them in the local database
- Connect the app to the database and retrieve the stored data to present to the user

Task 5: Make App Notifications

- Make the app send out notifications when a mail arrives from one of the registered sender

Task 6: Make an App Widget

- Make the app widget to update every 30 minutes and show the number of unread mails from each sender