
GitHub Username: https://github.com/huse

Messaging app

Description

Messenger for smart phones, sending text to friends around the world.
App will be written solely in the Java Programming Language.

Intended User

This app is for all smart phones around the world to communicate with each in a single app..

Features

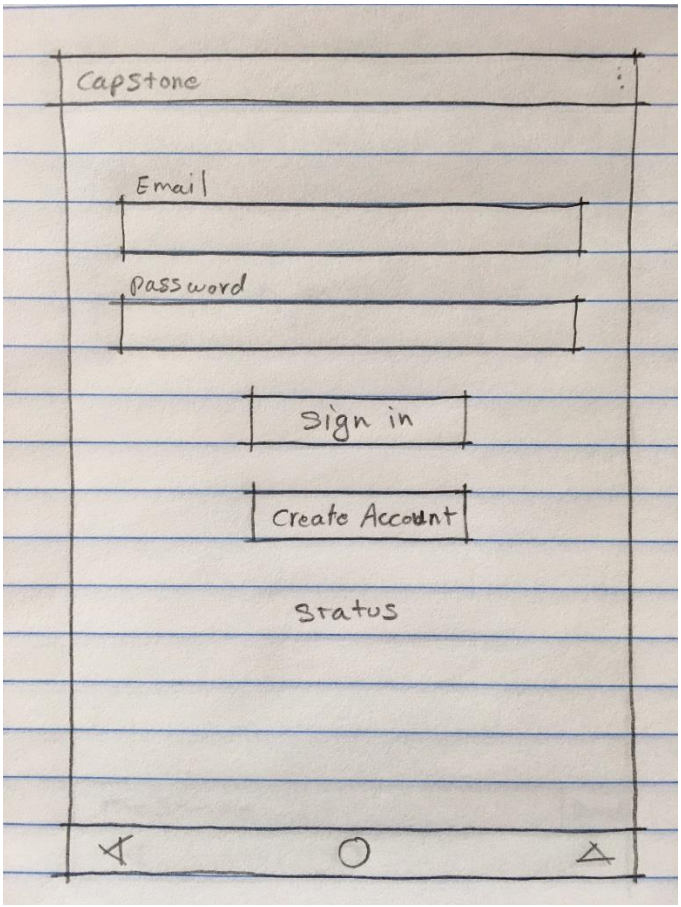
List the main features of your app. For example:

- authenticate information
- Send and receive messages and picture or video
- Show and save messages in recycle view

User Interface Mocks

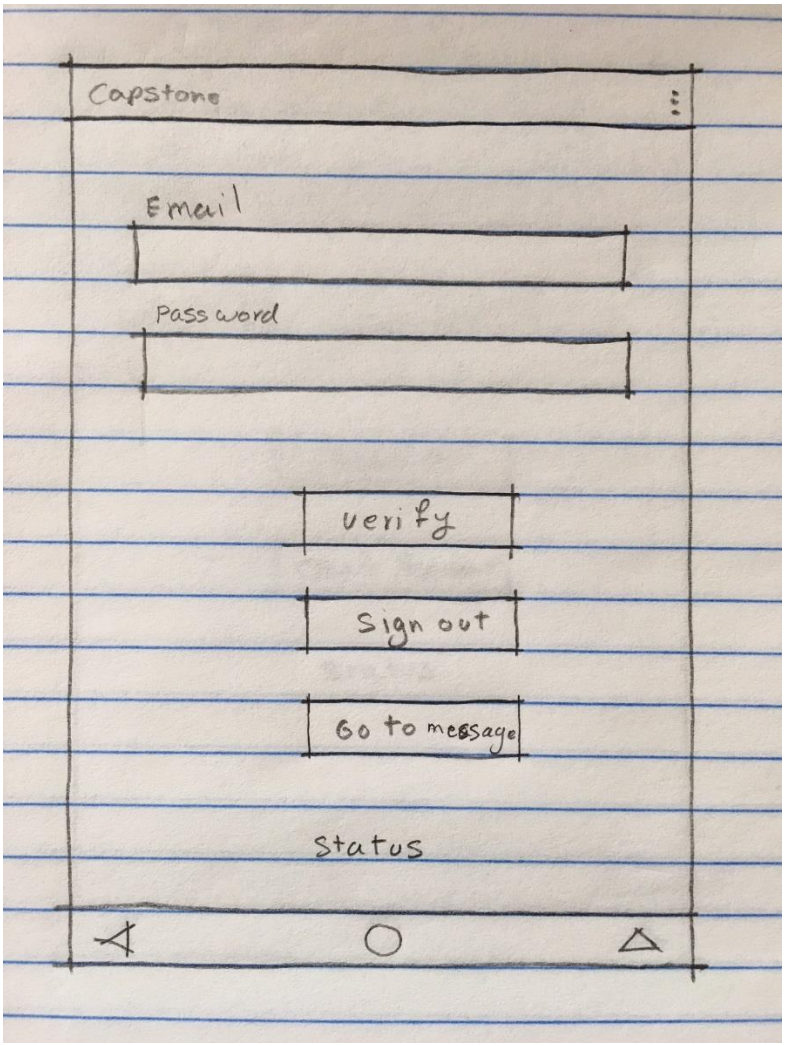
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



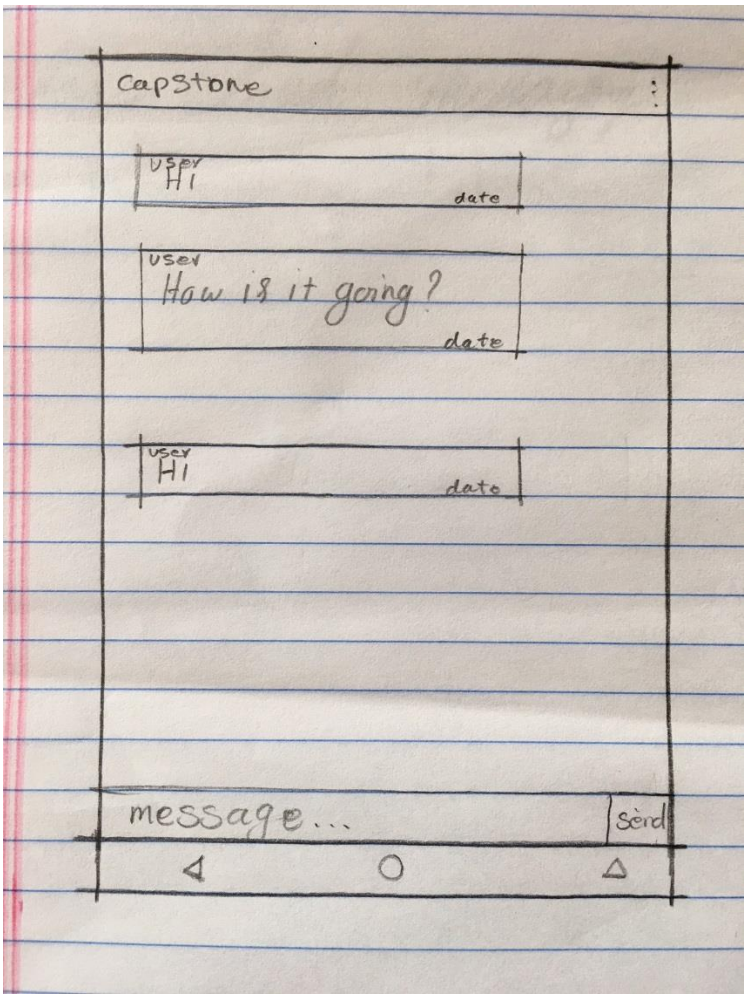
Sign in UI has two textEdit for password and username and one button to sign in and one button to create account. Also has a text view for status.

Screen 2



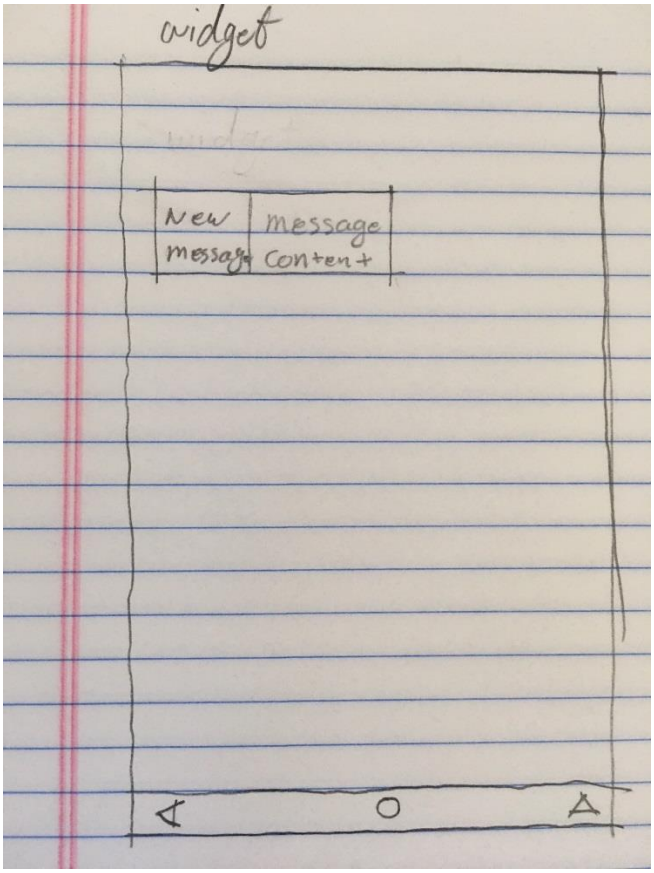
Sign in UI screen will be changed base on user sign in status and if user is signed in, it will show new buttons to verify email, signout and Go to message screen. It's also has two textEdit for password and username and has a text view for status.

Screen 3



This is the message activity UI that has a listview to show messages and a textView for typing messages and a send button.

Screen 4



Widget will be shown on android home screen and shows the status of the app.

Key Considerations

Libraries will be used

	Software / libraries to use	Version / or higher
	android studio	3.1.2
	Gradle	4.4
	Firebase	15.0.0
	firebase-auth	15.1.0
	firebase-core	15.0.2
	firebase-messaging	15.0.2
	firebase-client-android	2.5.2
	firebase-jobdispatcher	0.8.5
	Recyclerview	v7:27.1.1
	constraint:constraint-layout	1.1.0
	firebase-ui-database	1.0.1

How will your app handle data persistence?

For saving and handling data the app ses Firebase Realtime Database.

Describe any edge or corner cases in the UX.

In sign in when sign in pressed the user name and password send to server to authenticate user and return results to app to give access to user.
In message activity messages sync with database.
In message activity messages retrieve from server.
In message activity when send pressed the message content will be sent to other user.

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

App uses Butterknife to reduce coding.
App uses Android support library to design UI.
App uses Firebase library which is used to handle authentication, cloud messaging, Realtime database.

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

Firebase database for storing user information and messages.

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

In first step all libraries will be configured in gradle.

- Configure libraries

- Install dependencies and librares in Gradle.

Task 2: Implement UI for Main activity

- Create layouts for MainActivity,
- Main activity will be used to intent and initiate the loginActivity.

Task 3: Implement UI for Login activity

- Create layout for login activity
- Build UI for login activity.

Task 4: Implement UI for Message activity

- Create layout for message activity
- Build UI message to show each conversation

Task 5: Set up Firebase

- Implementing Firebase Realtime Database
- Creating firebase UI data base for storing data and information.
- Implementing Firebase authentication.
- Implementing Firebase cloud message
- Create MyFirebaseInstanceIDService and MyFirebaseMessagingService classes.
- Implementing methods for sending and showing messages.

Task 6: RTL

- To support RTL
  android:supportsRtl="true"
  will be added to manifest.

Task 7: Managing resources

- Stings will be stored at strings.xml
- Colors will be stored at colors.xml
- Project images will be stored at drawable directory under res.