

**GitHub Username:** kisraval

# Let's chat

## Description

A one to one chat application, which can be used to send text messages to your friends or family. The messages are backed up continuously to Firebase Cloud, so no worries of losing your data.

## Intended User

Anyone having a phone, who would like to chat with their family or friends freely are the intended users of the app.

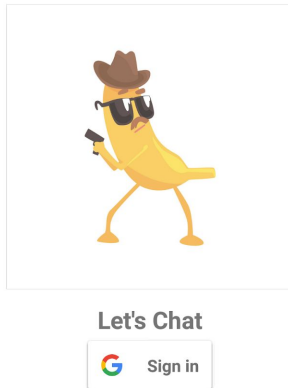
## Features

- One to one chat
- Send texts to intended user
- All data is backed up at cloud

## User Interface Mocks

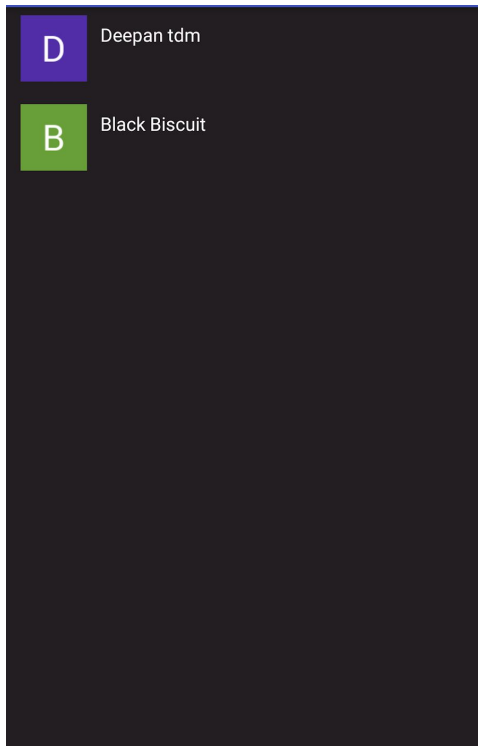
### Screen 1

---



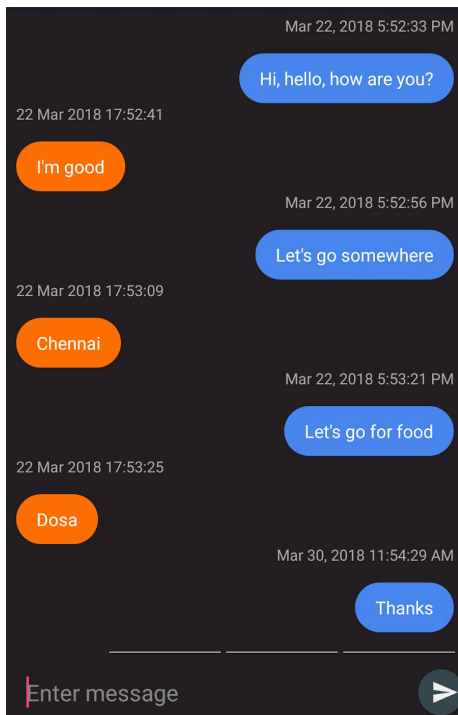
HomeScreen where the user can signIn using Google SignIn button. From the account information, we'll fetch the details like userName, email ID, etc. Upon SignIn, the user will be redirected to People screen where he can see all the people with whom he can chat.

## Screen 2



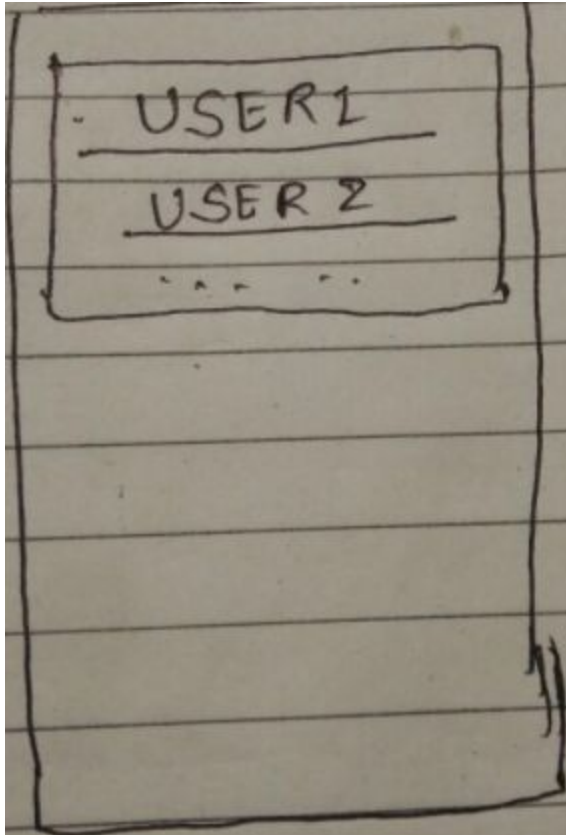
People Screen: All the people with whom the user can chat will be displayed here. Upon selecting the person, user will be redirected to chat screen.

## Screen 3



Chat screen: An Example screen where the messages from both sides can be seen. There is a space from where the user can type the new messages (like usual design).

## Widget Screen



Widget which shows list of all the users to whom the user can chat. Upon selecting any one of the user, the user will be redirected to that selected person's chat screen.

## Key Considerations

How will your app handle data persistence?

I will be using Firebase realtime database to handle data persistence.

Describe any edge or corner cases in the UX.

From people screen, upon pressing the back button, the app will be closed.

From chat screen, upon pressing the back button, user will be redirected to people screen.

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

I'll be using Picasso to load images such as images of people.  
Also, I'll be using Firebase Authentication for signIn purpose.

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

Firebase Authentication: To signIn users with their registered google account.

Firebase database: To store all the data including people's data as well as messages.

I'll taking help of firebase documents for the implementation.

## 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 and adding firebase support

- Starting with setting up project on Android Studio
- Add project to Firebase console and assign SHA1 key.
- Get the json file and add dependencies in Gradle files.
- Check if everything works.

### Task 2: Adding Google SignIn

- Add google SignIn button to UI
- Handle SignIn and make sure that the user reaches to the second screen when signIn is successful.
- Upon successful signIn, check if that user's data is already available on cloud. If not, add required data to cloud.

### Task 3: Adding People Screen

- Make UI for single user, containing photo and username
- Fetch list of all the user's and display the same using adapter.

- Upon selection of the user, make sure that new screen is opened along with passing the user's data to whom the user is going to chat.

## Task 4: Adding Message Screen

- Make UI to display a single message
- Fetch all the past message data and display the same using adapter.
- Make sure that messages from the user are displayed on right side and messages from other person is displayed on right side.

---

### Submission Instructions

- After you've completed all the sections, download this document as a PDF [ File → Download as PDF ]
  - Make sure the PDF is named "**Capstone\_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone\_Stage1.pdf**"