Github Username: instance686(Ayush Srivastava)

LocShout

1)**Description**:

*LocShout(Location Shout) is a location based reminder app which will help a user get the reminders based on a specific location.The reminders instead of being time based will be location based and the user and send other user their reminders which will triggered and notified to the target user when the intended user reaches a specific location for which location reminder is set.

*The main problem statement is that most of the reminders for user a location specific like doing a list activities on reaching college,groceries shopping or at work.

*The user is provided with a chat interface with self chat included where the user can send a note or a list to oneself/otheruser.

*The app requires registration to be done via unique phone number and app accesses the phone contacts which are on the platform system.

Intended Users:
Students,Family,Travellers,people with delivery jobs.

Features:

*Location specific reminders.

*In app reminder sharing to other users.

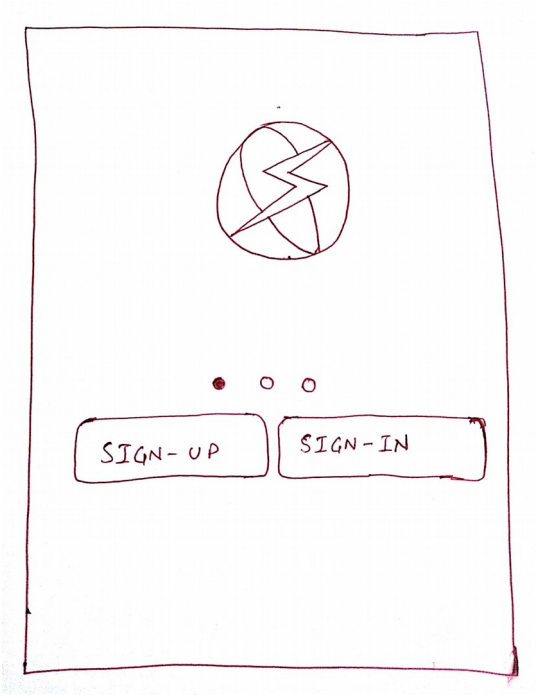
*Persistant Notes and Checklists.

*Color categorization of reminders.

*The app is also suppoted to work on tablets.

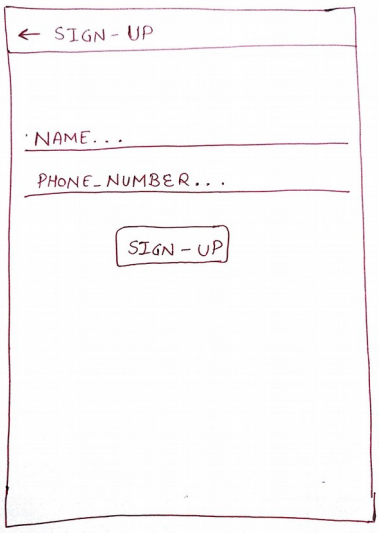
2)User Interface Mocks:

2.1)Intro Screen: The user will sign up or sign in by pressing there respective buttons along with the app walkthrough is given.



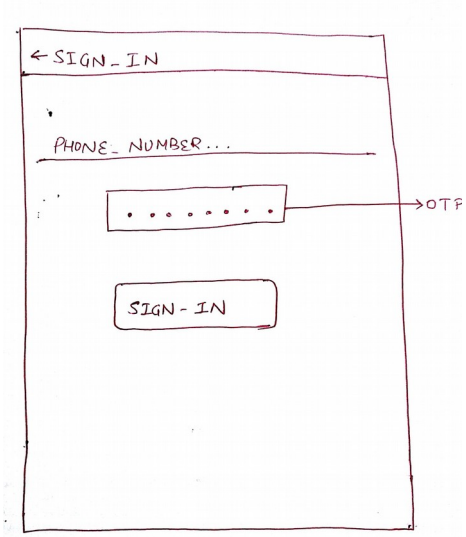
2.1-Intro Screen

2.2)Sign-Up Screen: The user can sign up into the app just by entring the correct mobile number and name.



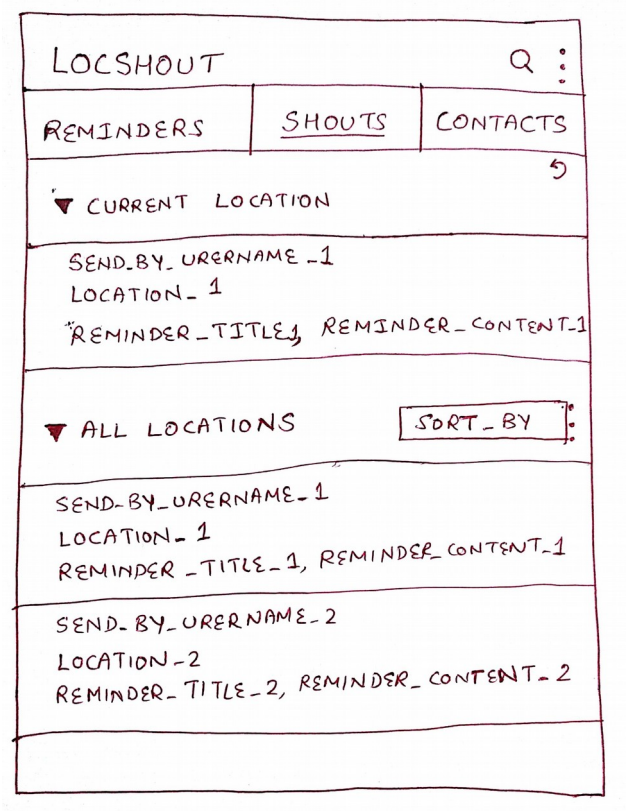
2.2-SignUp

2.3) Sign-In Authenticate-To sign in into the app the user has to aunthenicate their mobile numbers via OTP.

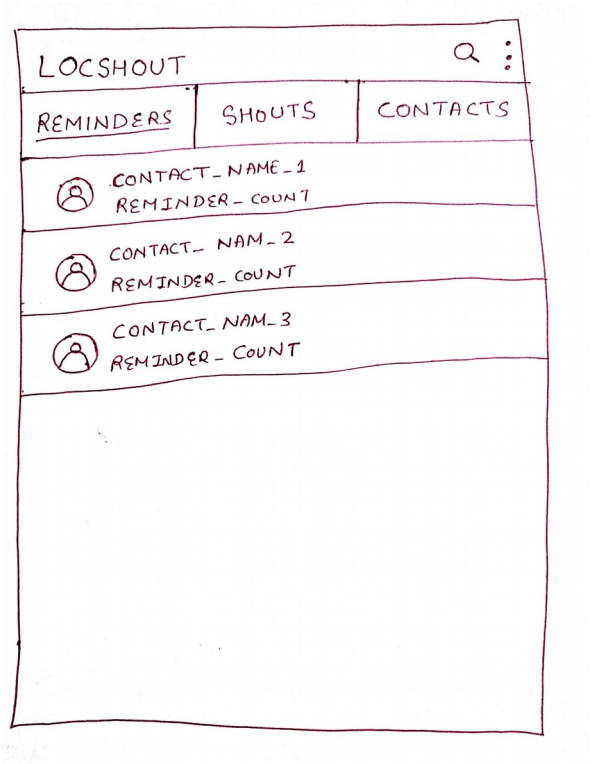


2.3-Sign -In

2.4)MainActivity(Shouts Fragment)-The shouts fragment shows all the reminders for te current location and other locations along with the reminder sender name and location.

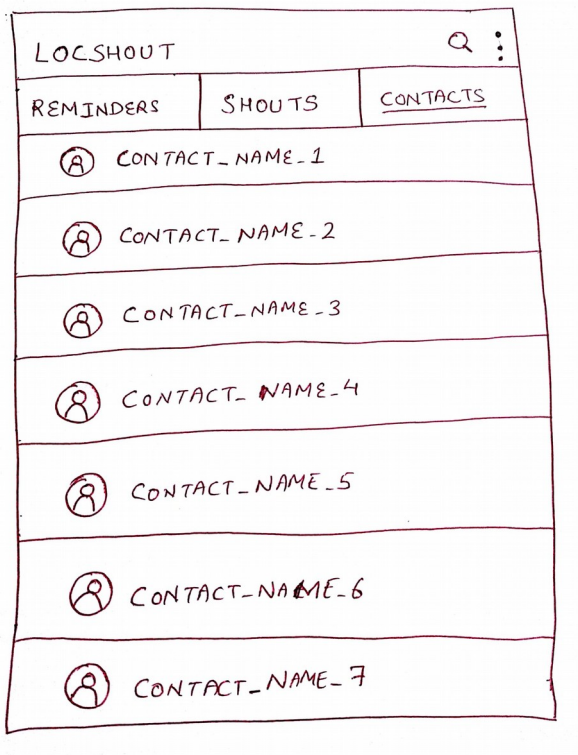


2.5)MainActivity(Reminders Fragment)-The Reminders fragment consists of all the users to whom the current app user has send the reminder.

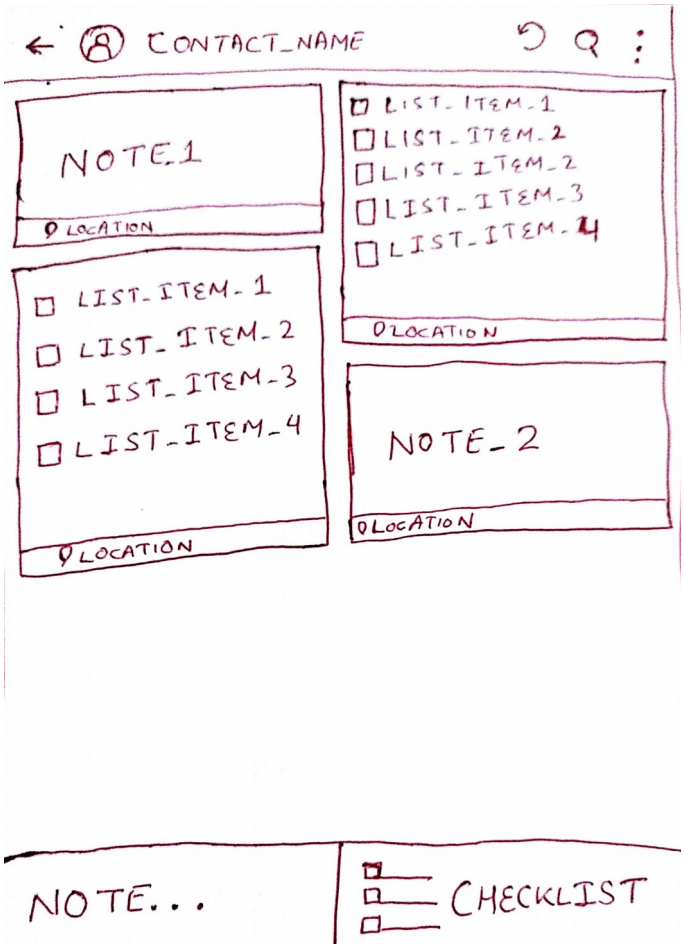


2.5-MainActivity(Reminder Fragment)

2.6)MainActivity(Contacts Fragment)-Contacts fragment consists of list of all the users who have the app installed and are in user's current contact list.

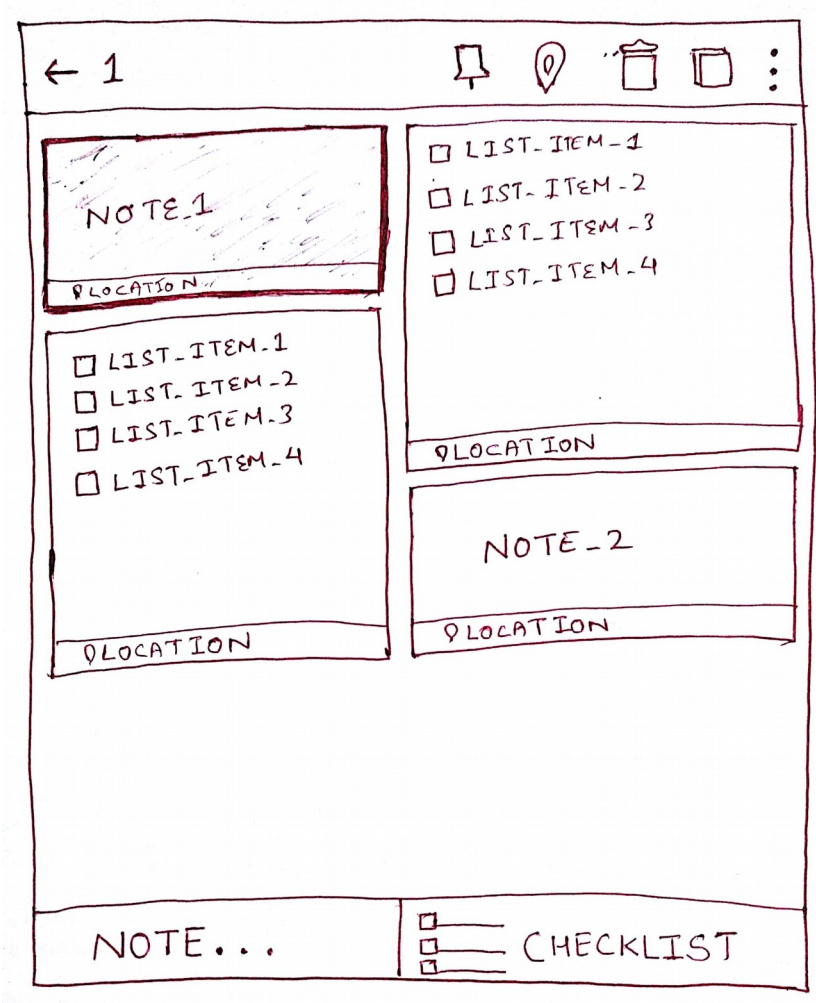


2.7)ReminderChat Activity Layout-This activity consists of all the reminders send to a particular contact in the current user app list.



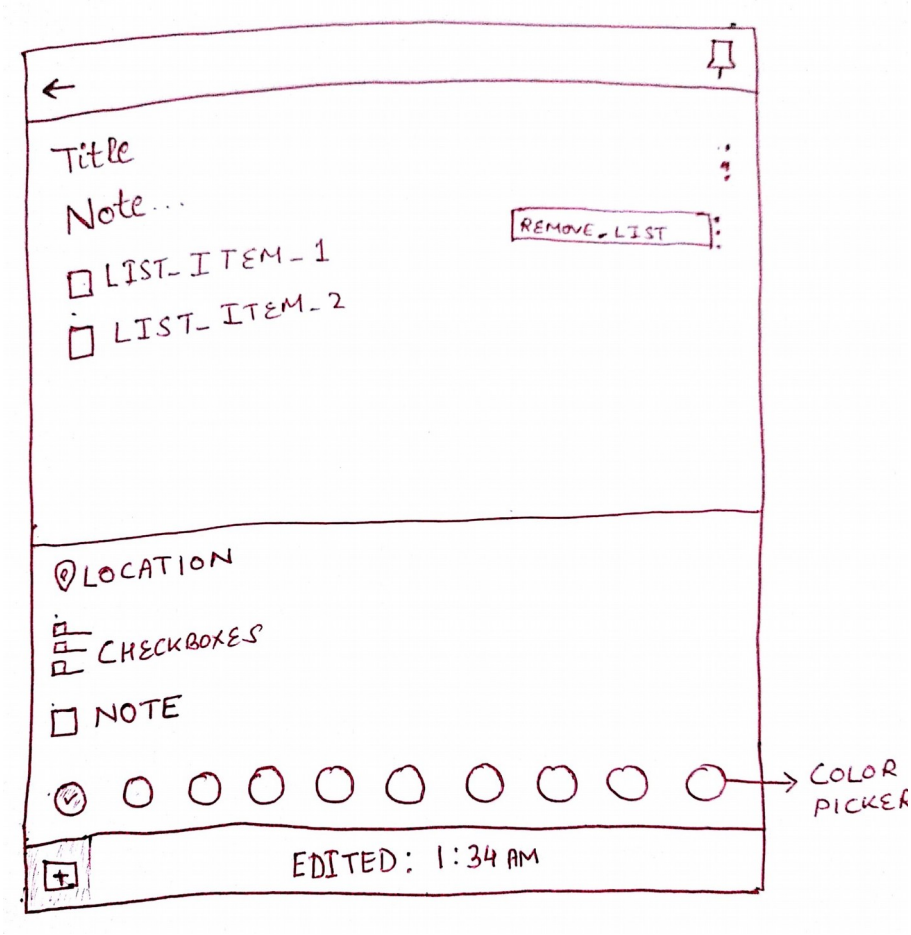
2.7-ReminderChat Activity

2.8)ReminderChat Activity Note Selected-Below layout shows the activity layout on selecting a reminder.



2.8-RecentChat Activity Note Selected

2.9)Notes Activity:-The notes activity shows all the components used in making a note with location changer,add title,add note,add checklist,add color to the reminder.



2.9)Notes Activity

3)Key Considerations :

*Data Persistance:-Data Persistance will be done with the help of a local database(SQLite DB) implemented with the help of rooms library along with a firebase realtime remote DB for sharing location to other users within the in app-chat system.A content provider for this app will also be provided to share notes data to other social media platforms.**The app also uses android app architecture of Model View ViewModel which helps in data persitance between the UI**.

*Location Fetching-The app takes help of Google Places Api to fetch the location of a particular reminder set by a user,the app also uses LocationServices specied in GoogleApiClient with the help of google play servies API.

4)Edge Case:

For choosing a location the app sends and Intent for for launching an activity with a map implemented by Google Places API with a request code in startActivityForResult method which return latitude and longitude of the selected placed picked for a particular reminder.

5)Libraries to be used:-

5.1)Retrofit2:-The app uses retrofit2 library for communication between the remote firebase DB and local app.

5.2)Glide:-The app uses glide to loading and caching of images into the profile section along with the images in contacts section alon

5.3)Google Play Services:-The app will use Google play servies to get location for a particular reminder.

5.4)Cloud FireStore:-The app will use cloud firestore to store and retieve data from a reatime remote databse.

5.5)Room:-The app will use room api for storing and fetching data in the local SQLite Databse.

5.6)LiveData:-The app will use livedata library to persist and show realtime data to the UI.

6)Project Task:-

6.1)Task 1:Project Setup

6.1.1) Configure all the libraries required for building the app in app-->build.gradle.

6.1.2) Define the target and min SDK required to build the app.

6.1.3) Define all the permissions required for the app in the manifest file.

-><uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>.

-><uses-permission android:name="android.permission.INTERNET"/>

-><uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>

-><uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

6.1.4) Create java packages for defining classes Database,Model,View,ViewModel,Network connection,Adapters,Activity,Fragments and Interfaces.

6.1.5) Create dimens,anim,drawable folder under res folders for defining dimensions,animations and drawable images respectively along with styles-v19.xml and style-v21.xml for defining styles on api level 19 and 21 above respectively.

6.1.6) Create initialization classes for databases and network activity.

6.2) Task 2:Build UI

6.2.1) Build UI for IntroActivity with a sign-in authenticator button along with the App walkthrough fragments build inside the same activity.

6.2.2) Build UI NumberAuthenticator activity with a EditText and input type as phone and a OTP check field for signing in into the app after OTP is recieved.

6.2.3) Build UI MainActivity with a custom appbar and a tablayout with three fragments(Reminder,Shouts,Contatcs).

6.2.4) Build UI for Contatcs fragment for showing all the contacts who have the app installed.

6.2.5) Build UI for Reminder fragment for showig all the contacts to whom reminder was send.

6.2.6) Build UI for Shouts fragment for showing all location and current location reminders.

6.2.7) Build UI for ReminderChat Activity with notes and checklist and a appbar on option selection.

6.2.8) Build UI for NotesActivity with the layout given in the section 2.9.

6.2.9) Build UI for UserProfile for managing a users profile with parallex scrolling.

6.3) Task 3:Style the App

6.3.1) Define primary and secondary colors to be used in the app along with all the other colors in the colors.xml.

6.3.2) Define the default dimension sizes in the dimens.xml.

6.3.3) Define the styles.xml with NoActionBar Theme along with the items and styles.

6.3.4) Define the icons to be used in the app under res->drawable.

6.3.5) Defines the strings to be used in the app in res->values->strings.xml.

6.4) Task 4:Build OTP Sign-in

6.4.1) Build OTP sign in and verfication with firebase authentication for phone numbers.

6.4.2) Connect app to firebase console using SHA-1 hash.

6.4.3) Send a verifcation code to entered phone number using PhoneAuthProvider.

6.4.4) On OTP verification send the user to the MainActivity.

6.5) Task 5: Build Contacts fragment.

6.5.1) Use a content provider for accessing all the phone contacts on the users phone and fetch from the firebase realtime database the list of all users contacts registered on the app.

6.5.2) Display the list inside a recyclerView.