Quixote Assessment Test App

Laksh Chauhan
+918126681065
iamlakshable@gmail.com

Screens

1. Login Screen

   The login screen lets an already registered user access the app. Upon clicking on the Login button, the provided inputs are first validated and appropriate errors are shown for:

   i) Any of the two input fields being empty, or
   ii) The provided credentials do not match any user in the database.

   In a case where the user is not registered yet, a register button is provided to take the user to the register screen.

2. Register Screen

   The register screen lets any un-registered user create a new account after providing Name, Phone Number, Email Address and Password. Once the user clicks on the Register button, the provided inputs are validated and appropriate error messages are shown in case of a faulty input based on the following constraints:

   i) Email: Name and domain must be 4-25 characters long and must have an '@' symbol and a '.' symbol in the domain.
   iii) Password: password must start with a lowercase letter. Two uppercase letters, two digits and a special character are required. The password cannot contain user's name. Password should be 8-15 characters long.
   iv) Phone Number: The phone number must be Indian.
   v) Name: Cannot be empty

3. Home Screen

   This screen hosts all of the notes stored in our app in a list format. Each row consists of:

   i) A photo associated with the note.
   ii) Title of the note.
   iii) A glance at the summary of the note.

Clicking on any row brings up a details screen.

4. Details Screen

   This screen shows the selected note in detail.
   It shows all of the photos of the note at the top along with the title and the complete description of the note. Users can slide left or right to view other photos.

5. Add note Screen

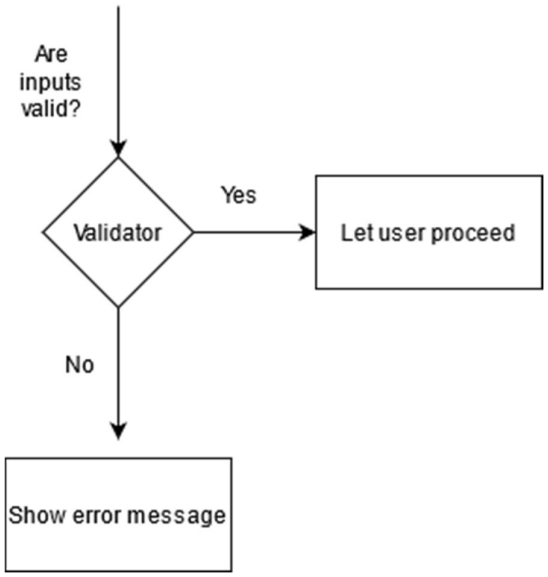
   This screen allows the users to add a new note themselves.
   Users are allowed to add:
   i) A title: 5-100 characters allowed.
   ii) A description: 100-1000 characters allowed.
   iii) Photos from either the gallery or directly the camera. (10 maximum)

   Upon clicking on the 'Done' button in the menu bar, the inputs are validated and appropriate error messages are shown when needed.

   If all inputs are valid, the note is then added to the Home Screen to display.

Validations

Validations are performed using Regular Expression (Regex) pattern matching for Email, Phone number and Password.

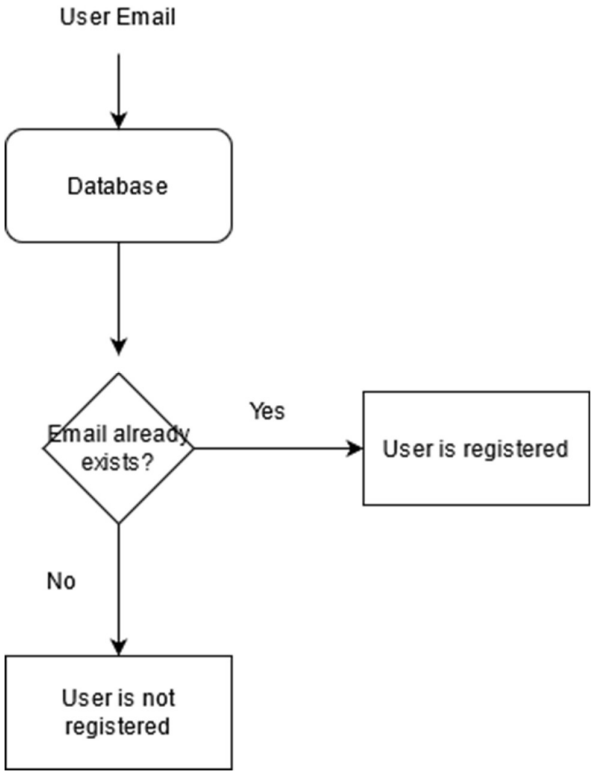


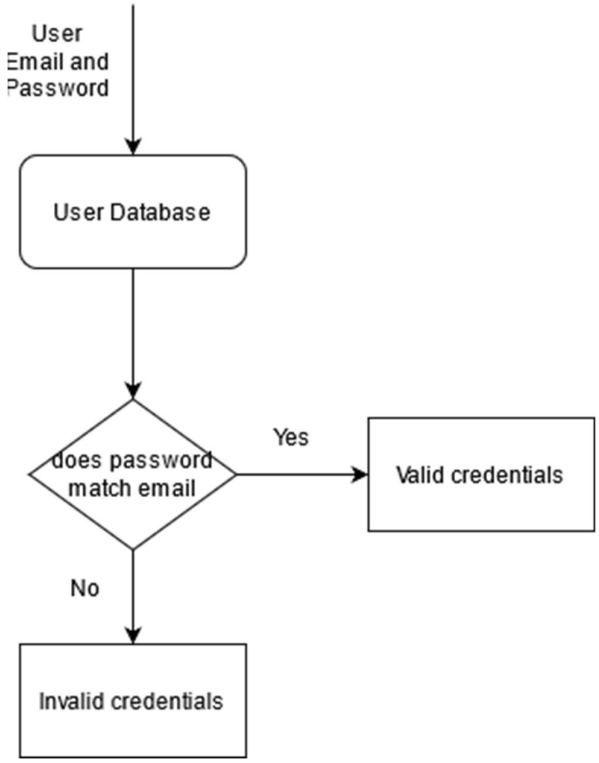
How login and register works

Login and register functions of the app both use SQLite database in the backend to store and check existing user's information

- Login
  The inputs provided by the user are checked if this user is registered or not. If not registered, the user is prompted that this User does not exist. If the user is registered, then the User database is queried for the row where Email and Password match the given inputs.
- Register
  The inputs provided by the user are checked if this user is already registered or not. If yes, then the user is prompted that this User already exists. If not, then a new row is inserted into the User database.

The user password is stored in the database in encrypted form.





Encryption

The app uses the md5 algorithm to store passwords in the database.

Since md5 is a deterministic hashing function, it always produces the same output for a particular input. Given this information, we can check if the provided password is correct by first passing it through the md5 hashing function and then checking if hashed password and the database password match or not.

User Database format

The user database stores the User information in a user table of the following format:

_ID (Primary Key)	Name	Email	Password	Phone Number
1	…	…	…	…
2	…	…	…	…

How home screen works

Home screen uses a RecyclerView and custom Adapter to populate the RecyclerView.

A cursor is queried from the Notes Database for all the notes and is provided to the Adapter as a dataset to populate the views in the list.

Structure of Notes database

Since each note can have multiple photos, the notes database is divided into two table:

- Notes Table: it stores the Note title and Note description along with a unique note id.

_ID (Primary Key)	Title	Description
1	…	…
2	…	…

- Photos table: it stores all the photos for a particular note by referencing the note id which can uniquely identify a note in the Notes Table.

_ID (Primary key)	Note ID (Foreign Key)	Photo Path	Photo Name
1	1	…	…
2	1	…	…

The app stores all photos locally on the device and only stores the path and the name of each image in the database to allow quick and error free Queries since the CursorWindow only allows 1MB of data at a time and if a photo exceeds that limit, the app might crash.

How a query on Notes Database works

In either the Home screen or the details screen, the screen first queries the Notes Table for all the existing notes in the database.

Once all the notes are received, another query for all photos of a particular note is made on the Photos Table.

In case of Home Screen, on the first image of the received images is used and in case of Details screen, all the images are used.

How the Details Screen displays all images

The details screen uses a ViewPager along with a custom adapter in order to display all the images in the top potion of the screen. Users can swipe left or right to view the previous or next image.

How Edit screen works

The edit screen also uses a ViewPager to display all the selected images.

To select an image, the user is provided with a choice to choose an image from the gallery or click a new image from the camera itself. Upon choosing images, they keep getting added to the display section at the bottom.

Once the 'Done' button is clicked on and all the inputs are validated, the Note is then inserted into the database and the Edit screen closes to show the Home screen again.

Testing

Tests are provided for the validator of Email, password and phone number since these three are the most complex validation processes in the app.

Improvements to be made

There are certain improvements this app can go through given more time

- Currently all database operations are performed on the main thread, which might cause lag for larger data sets. This can be resolved by using a CursorLoader or LiveData along with Rooms.
- There could be a delete note functionality implemented.
- One must be able to delete the imported images in the Add images activity if they need to.
- Confirmation before adding new note can be added.
- Using SharedPreferences login/register screen can be made to appear only once when the user first opens the app.