॥ सिद्धिः भूषयते विद्याम् ॥

avantika
UNIVERSITY

Discipline Core

Design Project: Mobile Application

CS4201

Project Documentation

Sheikh Muhammed Tadeeb (AU19B1014)

❖ Abstract:

- **Problem Statement:**

An easy and simple mobile based application for learning or exchanging the courses/topics from seniors or colleagues of the same field or college with minimum payable amount.

Domain
Student oriented Learning based application.

- **Objective:**

The following are the objectives of this application:

S.nos	Objectives
1	To provide students a secured platform where they can learn from their choice of seniors or colleagues based on the ratings and command on particular topic or subjects of seniors.
2	Knowledge transfer in a simple manner.
3	Learning portions of topics/subject which were unclear to the student in class.
4	Avoiding wastage of time by giving preferences to topics which are not that important.
5	Getting the returns of your knowledge.

❖ Introduction:

- **Need of the project:**

Well, it’s a very common problem with the students of college or university that they are unaware about two things i.e., what should they study and how much to study. Even when they understand that what to study in their college or university lectures, they still mess with the second part i.e., how much to study, generally students refer to online sources or google for exploring the topics that they didn’t understood in the class, but its very common that they will either learn from paid courses provided by existing solutions in the market or from some open free source like YouTube but both these have one drawback that they teach you many extra things which are of no use for the current time, secondly you don’t know which video or reference is the best. So, its better if students learn or share their knowledge with their colleagues or seniors.

- **Scope of project:**

1) This app provides you with a great starting point for building a strong knowledge foundation.
2) In the app a secured user/student needs to register with a username and password for accessing the details of seniors and their contents and to interact with him.
3) Courses can be filtered based on rating, topics and their cost.
4) So, students can communicate with the teacher (here colleagues or seniors) of their choice and can negotiate either on payment or on skill sharing. This will make learning interesting!

❖ Ideation:

- **Brain-Storming:**

After the brainstorming session, the list of following problems came out in existing learning base applications:

1) Unavailability or very less chances of interacting with local students.
2) Loss of interest after some time.
3) High charges for the course or subjects.
4) Less flexibility.
5) No feature of learning a particular or part of a topic from a full course/ subject.
6) Doubt about ratings.
7) Unnecessary or Extra things that dooms the main things to be covered.
8) No guarantee of in-time learning.
9) No chance of extra income.

10) Unavailability of references on projects.

11) Greater or larger audience-based learning which hinders one-on-one attention.

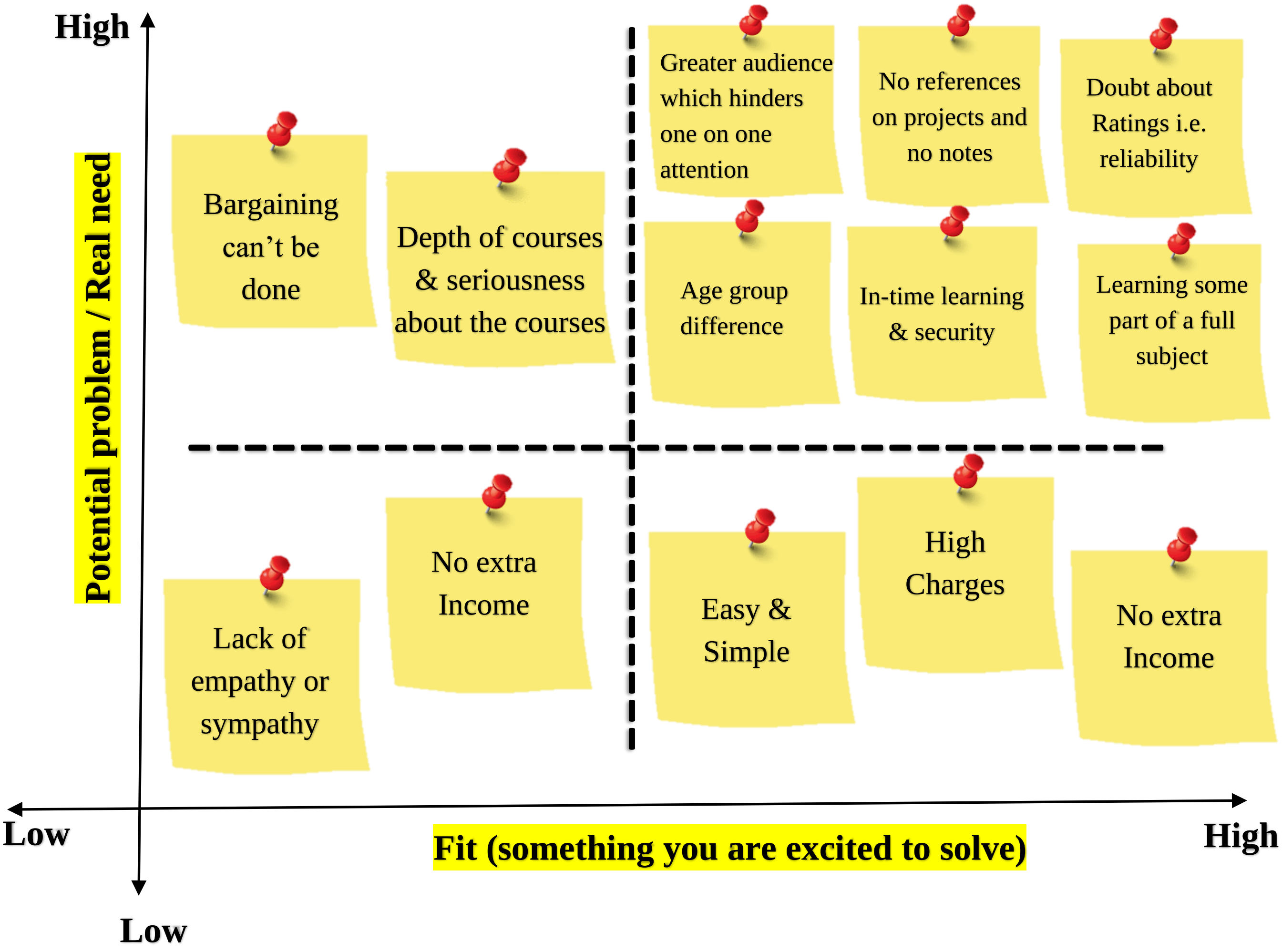
12) Lack of empathy and sympathy.

13) Age group difference which makes it hard for the students to open up with the teachers.

14) Bargaining can't be done i.e., if some student is learning from his/her senior then instead of paying him, he may simply ask him to compensate the amount if he helps the senior with his small stuffs.

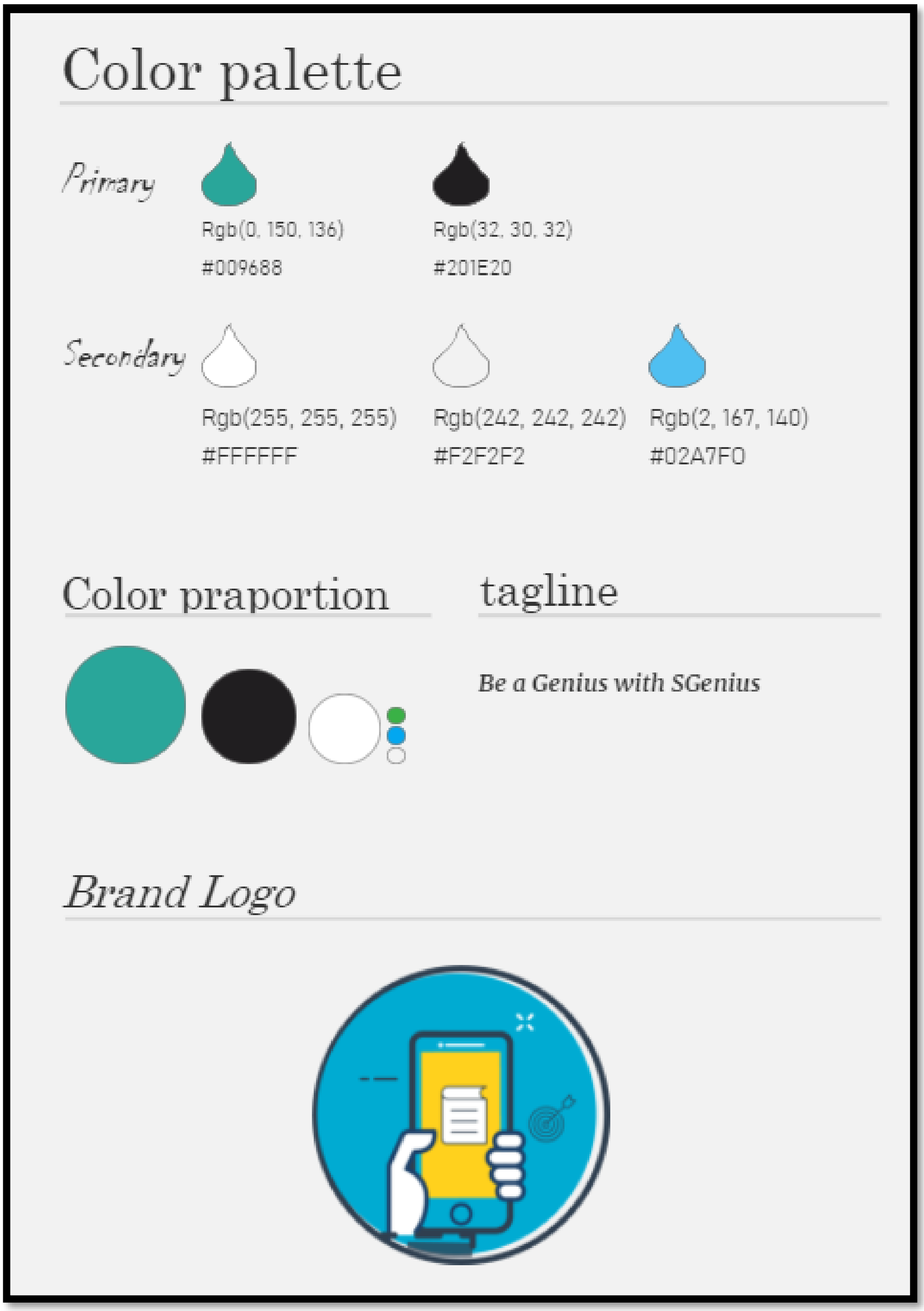
15) Depth of courses or complexity can't be predicted on other language platforms as you don't rely easily on a stranger teacher.

- **Filtration (of problems):**


High
Potential problem / Real need
Low
Bargaining can't be done
Depth of courses & seriousness about the courses
Greater audience which hinders one on one attention
No references on projects and no notes
Doubt about Ratings i.e. reliability
Age group difference
In-time learning & security
Learning some part of a full subject
Lack of empathy or sympathy
No extra Income
Easy & Simple
High Charges
No extra Income
Low
Fit (something you are excited to solve)
High

❖ Design & Methodology:

- **Style Guide:**



Color palette

Primary

Rgb(0, 150, 136)
#009688

Rgb(32, 30, 32)
#201E20

Secondary

Rgb(255, 255, 255)
#FFFFFF

Rgb(242, 242, 242)
#F2F2F2

Rgb(2, 167, 140)
#02A7F0

Color praportion

tagline

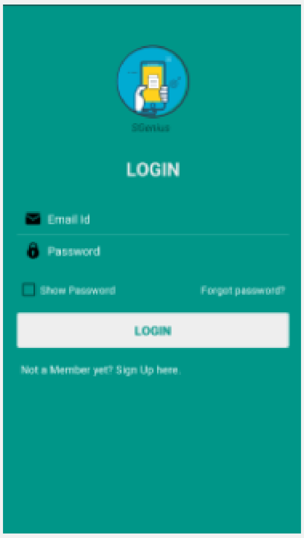
Be a Genius with SGenius

Brand Logo

typography

	Information	*Voice*
Name & Use	*Aa*	"Century" family text is used to clearly display the Headings text.
	Aa	"Bell MT" family text is used to for H2 and Buttons.
	Aa	Bahnschrift family text is used to clearly display the Body.
Colors	Aa Aa	Preferred is Black (darker shade)
	Aa ***Aa***	Preferred is Black (darker shade)
	Aa Aa Aa Aa	Preferred is Black (lighter shade)

Forms



LOGIN

✉ Email Id

🔒 Password

□ Show Password Forgot password?

LOGIN

Not a Member yet? Sign Up here.

Buttons

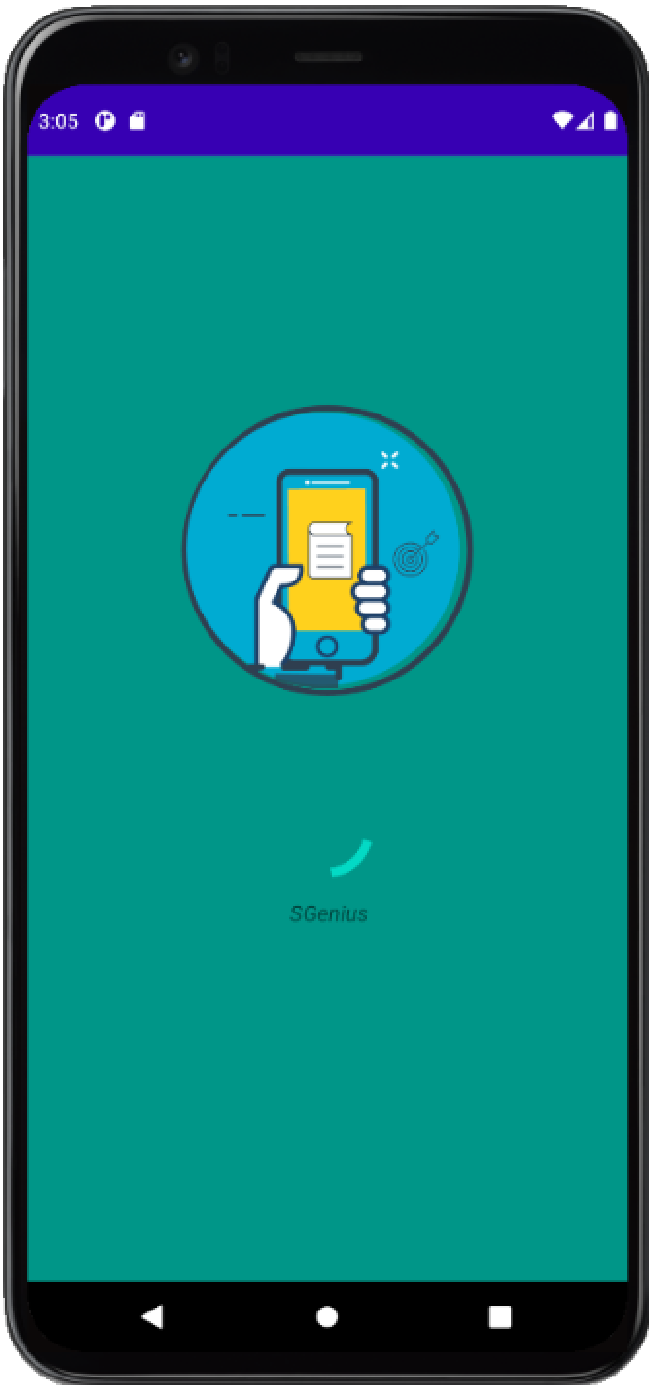
LOGIN SIGNUP

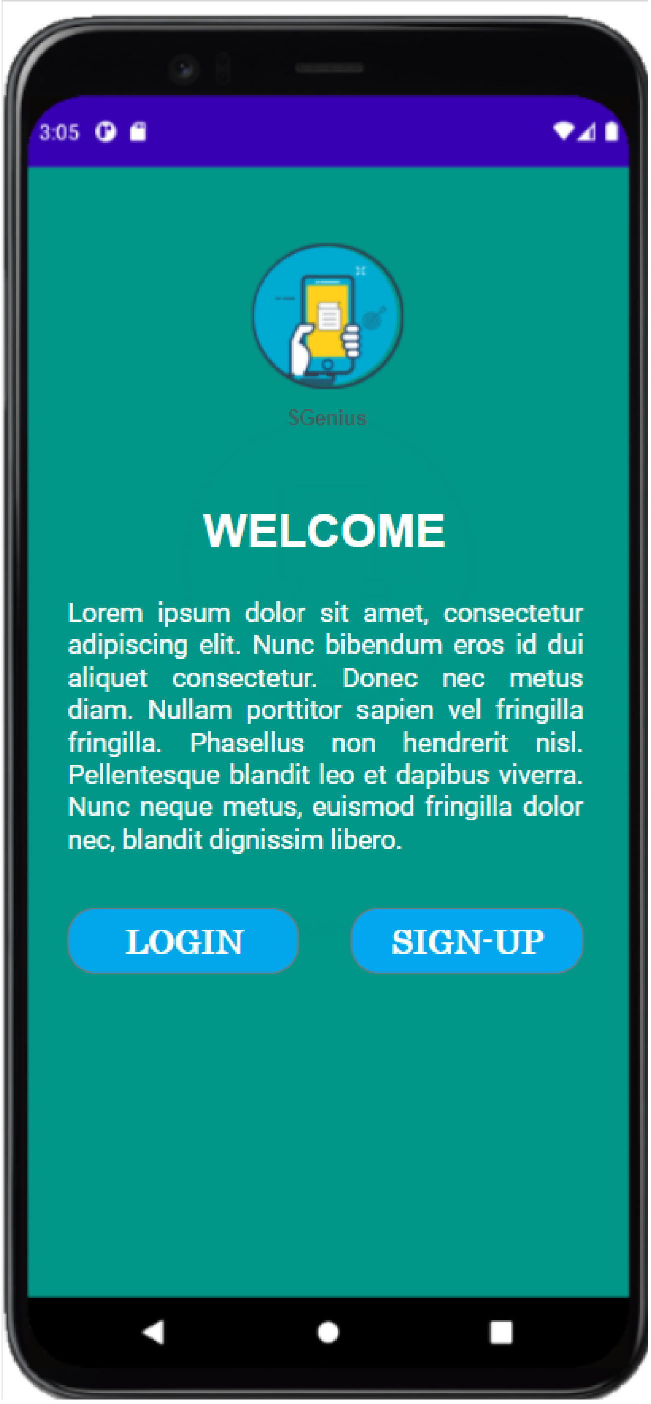
Icons

⊖ 🏛 🔒 📞 👤 ⊗

🔒 ✉

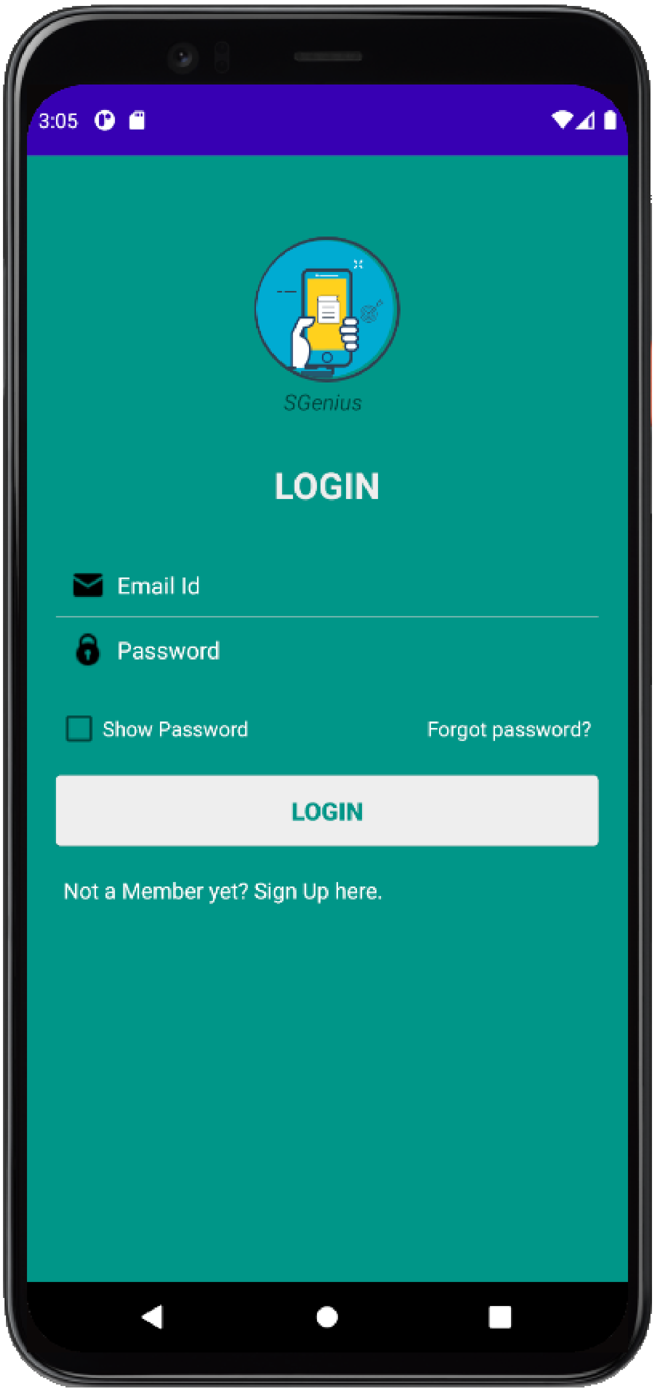
- **Wireframe:**

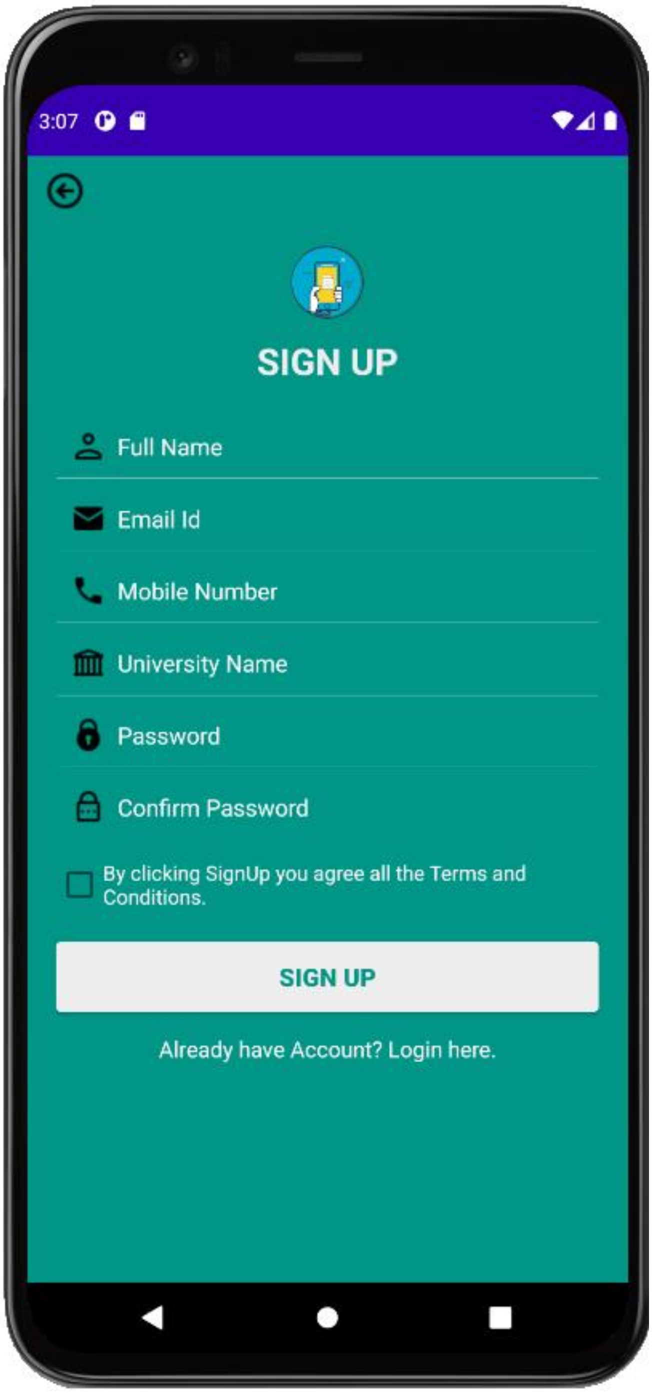


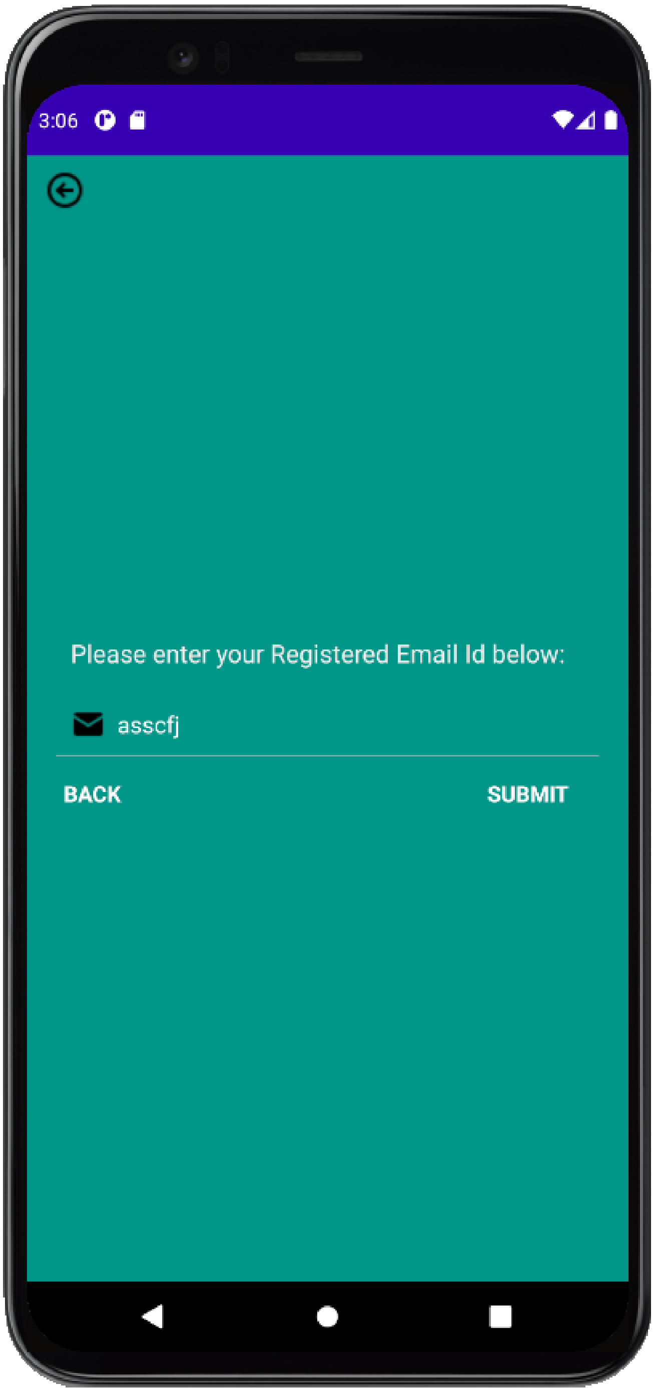


WELCOME

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc bibendum eros id dui aliquet consectetur. Donec nec metus diam. Nullam porttitor sapien vel fringilla fringilla. Phasellus non hendrerit nisl. Pellentesque blandit leo et dapibus viverra. Nunc neque metus, euismod fringilla dolor nec, blandit dignissim libero.

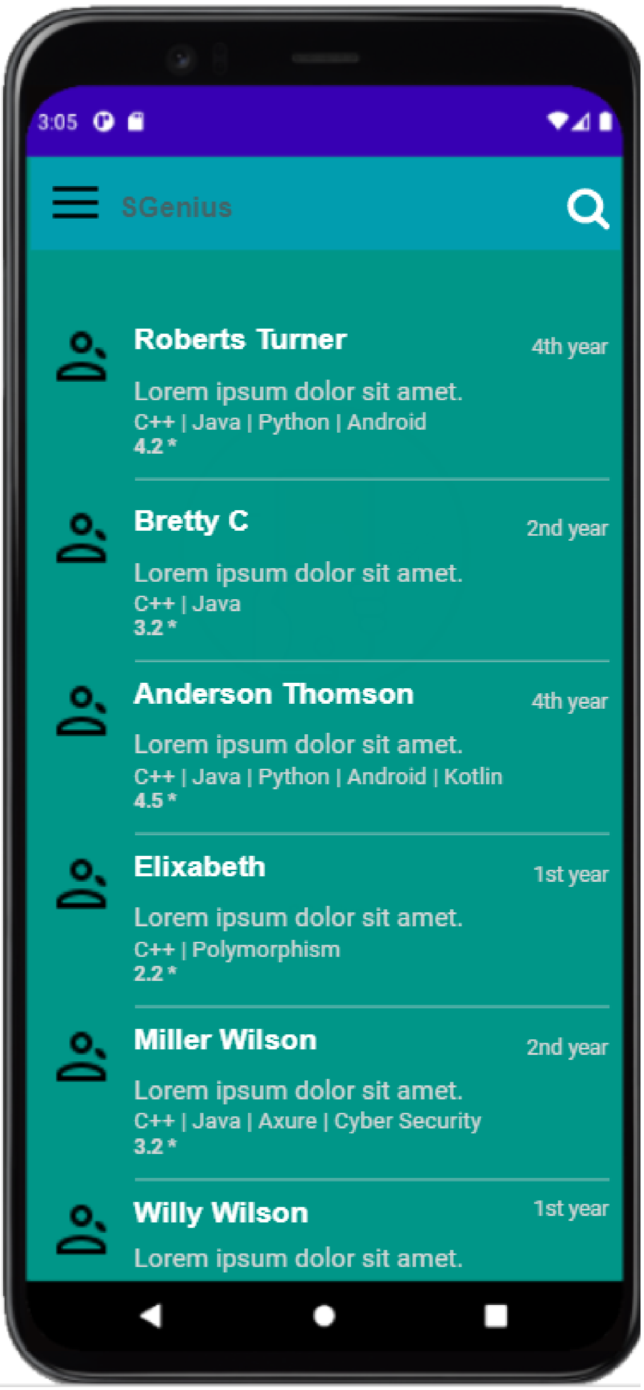

3:05
SGenius
LOGIN
Email Id
Password
□ Show Password
Forgot password?
LOGIN
Not a Member yet? Sign Up here.


3:07
SIGN UP
Full Name
Email Id
Mobile Number
University Name
Password
Confirm Password
□ By clicking SignUp you agree all the Terms and Conditions.
SIGN UP
Already have Account? Login here.



Please enter your Registered Email Id below:

asscfj



Roberts Turner 4th year
Lorem ipsum dolor sit amet.
C++ | Java | Python | Android
4.2 *

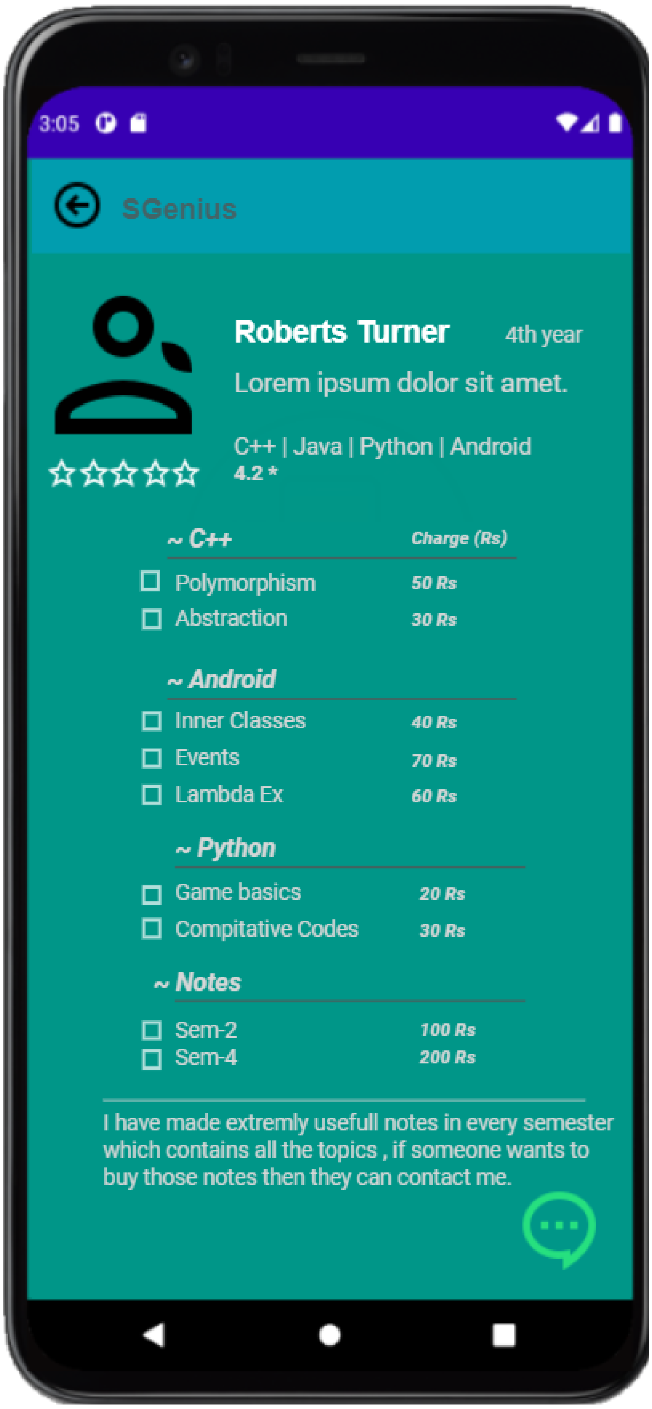
Bretty C 2nd year
Lorem ipsum dolor sit amet.
C++ | Java
3.2 *

Anderson Thomson 4th year
Lorem ipsum dolor sit amet.
C++ | Java | Python | Android | Kotlin
4.5 *

Elixabeth 1st year
Lorem ipsum dolor sit amet.
C++ | Polymorphism
2.2 *

Miller Wilson 2nd year
Lorem ipsum dolor sit amet.
C++ | Java | Axure | Cyber Security
3.2 *

Willy Wilson 1st year
Lorem ipsum dolor sit amet.



SGenius

Roberts Turner 4th year

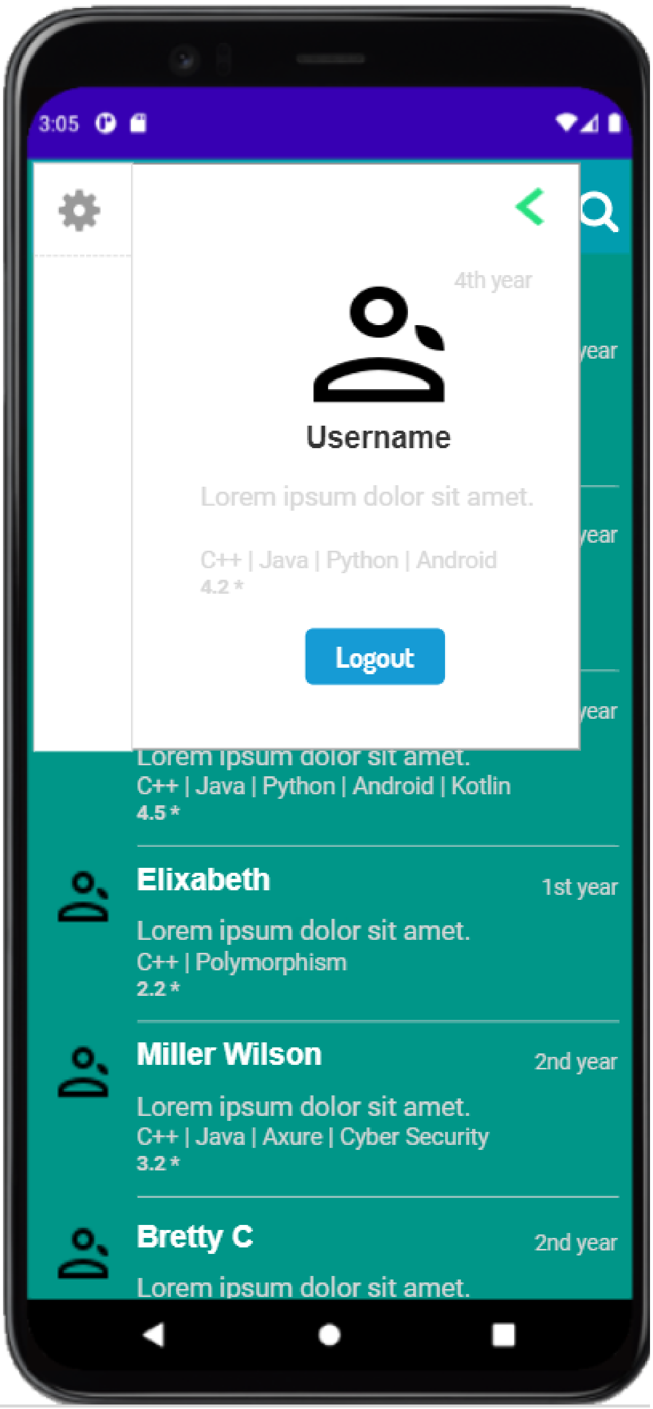
Lorem ipsum dolor sit amet.

C++ | Java | Python | Android
4.2 *

☆☆☆☆☆

~ C++	Charge (Rs)
□ Polymorphism	50 Rs
□ Abstraction	30 Rs
~ Android	
□ Inner Classes	40 Rs
□ Events	70 Rs
□ Lambda Ex	60 Rs
~ Python	
□ Game basics	20 Rs
□ Compitative Codes	30 Rs
~ Notes	
□ Sem-2	100 Rs
□ Sem-4	200 Rs

I have made extremly usefull notes in every semester which contains all the topics , if someone wants to buy those notes then they can contact me.



4th year

Username

Lorem ipsum dolor sit amet.

C++ | Java | Python | Android
4.2 *

Logout

year

year

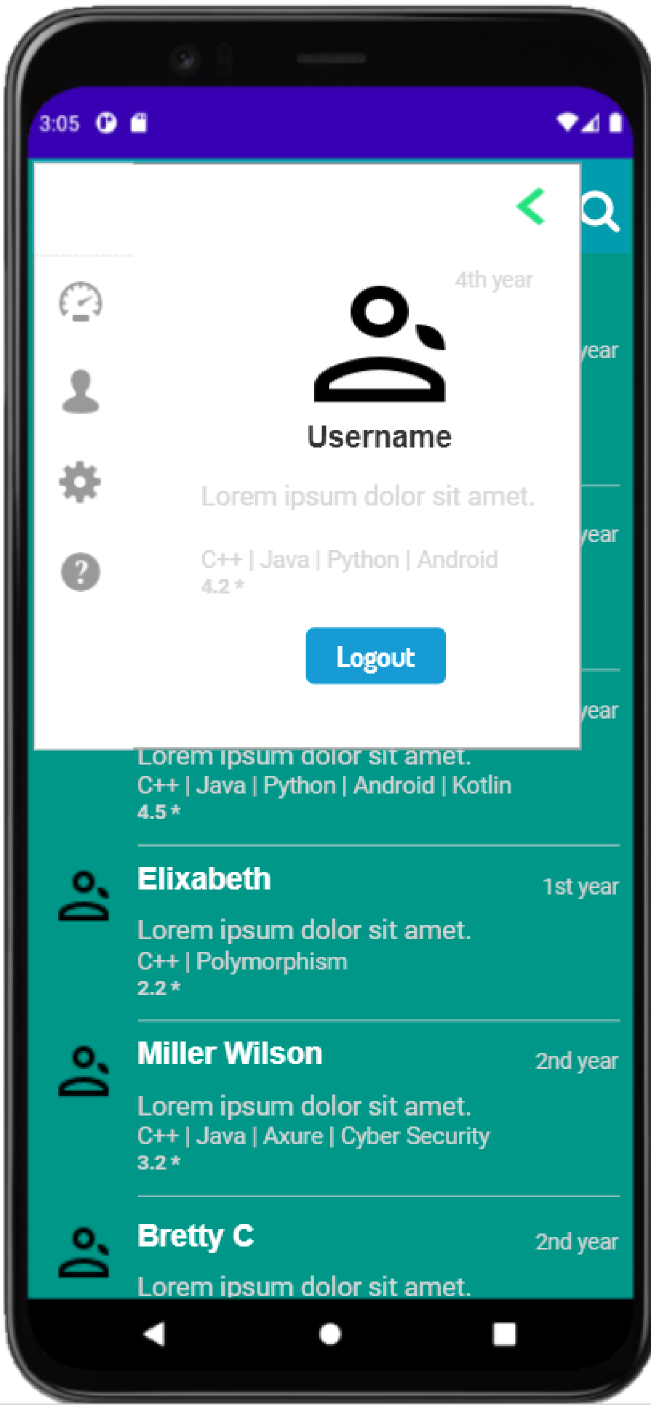
year

Lorem ipsum dolor sit amet.
C++ | Java | Python | Android | Kotlin
4.5 *

Elixabeth 1st year
Lorem ipsum dolor sit amet.
C++ | Polymorphism
2.2 *

Miller Wilson 2nd year
Lorem ipsum dolor sit amet.
C++ | Java | Axure | Cyber Security
3.2 *

Bretty C 2nd year
Lorem ipsum dolor sit amet.



3:05

4th year

Username

Lorem ipsum dolor sit amet.

C++ | Java | Python | Android
4.2 *

Logout

year

year

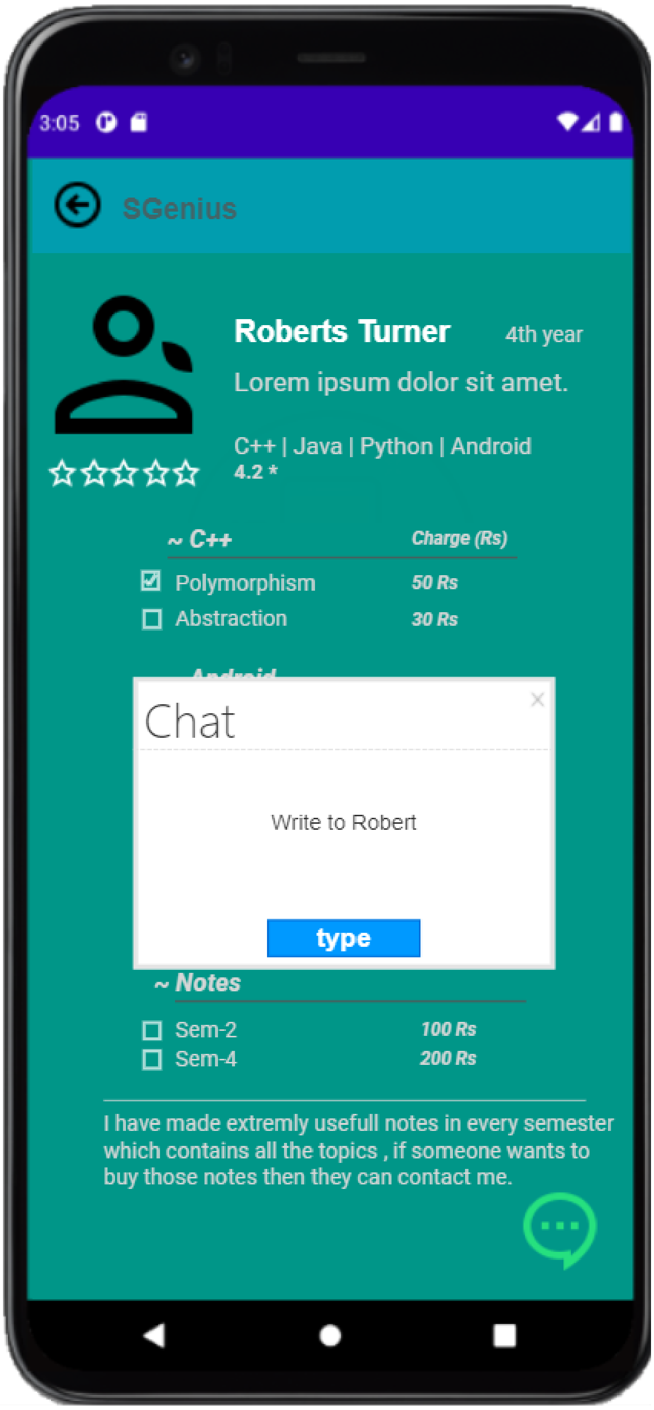
year

Lorem ipsum dolor sit amet.
C++ | Java | Python | Android | Kotlin
4.5 *

Elixabeth 1st year
Lorem ipsum dolor sit amet.
C++ | Polymorphism
2.2 *

Miller Wilson 2nd year
Lorem ipsum dolor sit amet.
C++ | Java | Axure | Cyber Security
3.2 *

Bretty C 2nd year
Lorem ipsum dolor sit amet.



3:05

SGenius

☆☆☆☆☆

Roberts Turner 4th year
Lorem ipsum dolor sit amet.
C++ | Java | Python | Android
4.2 *

~ C++	Charge (Rs)
☑ Polymorphism	50 Rs
□ Abstraction	30 Rs

Android

Chat ×

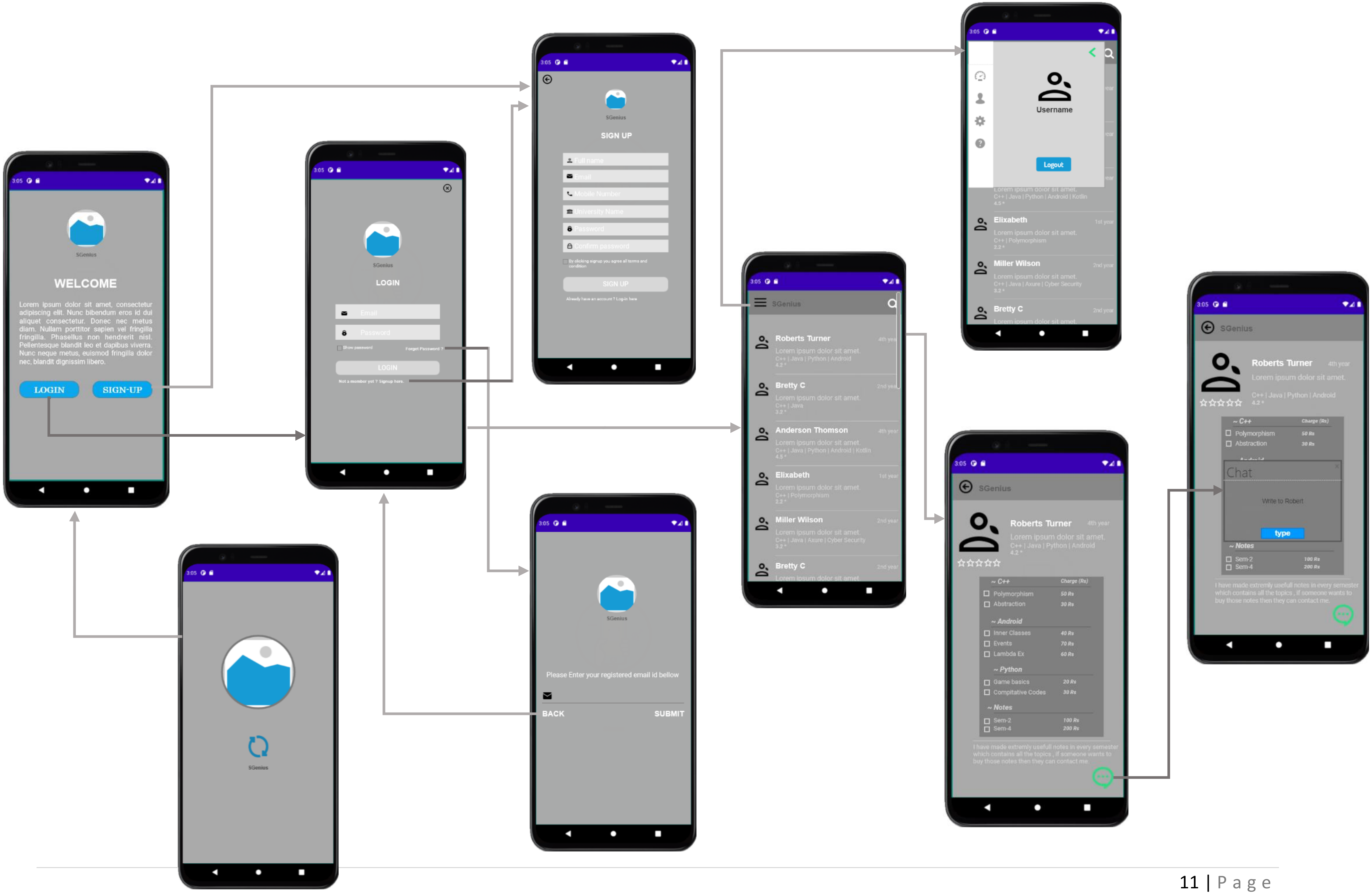
Write to Robert

type

~ Notes	
□ Sem-2	100 Rs
□ Sem-4	200 Rs

I have made extremly usefull notes in every semester which contains all the topics , if someone wants to buy those notes then they can contact me.

Screen Flow

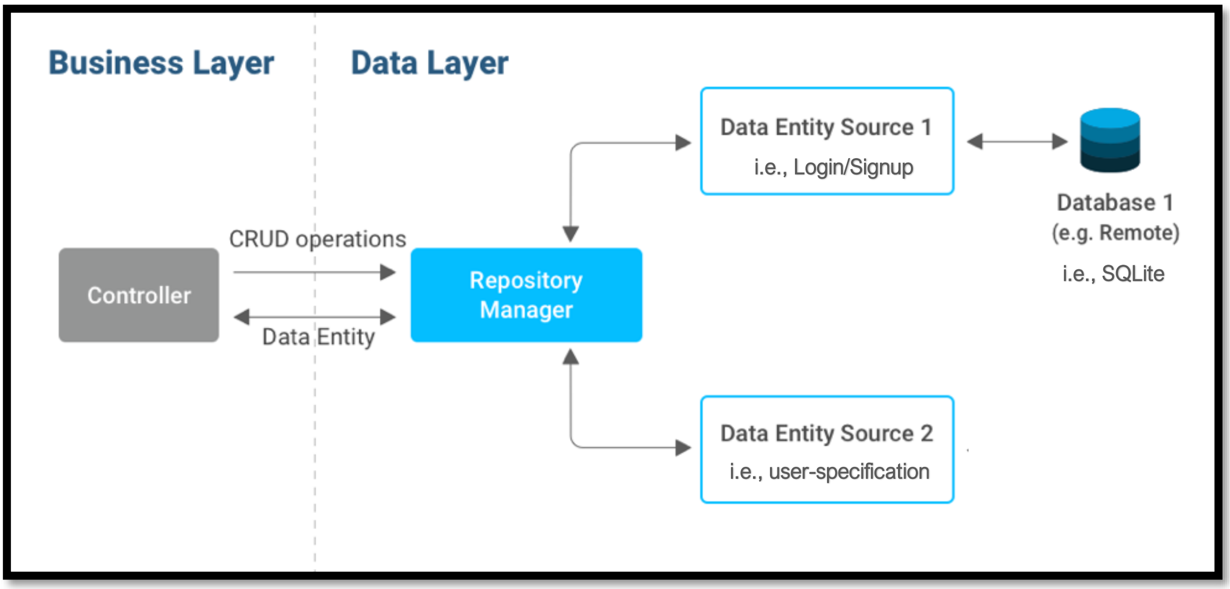


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc bibendum eros id dui aliquet consectetur. Donec nec metus diam. Nullam porttitor sapien vel fringilla fringilla. Phasellus non hendrerit nisl. Pellentesque blandit leo et dapibus viverra. Nunc neque metus, euismod fringilla dolor nec, blandit dignissim libero.

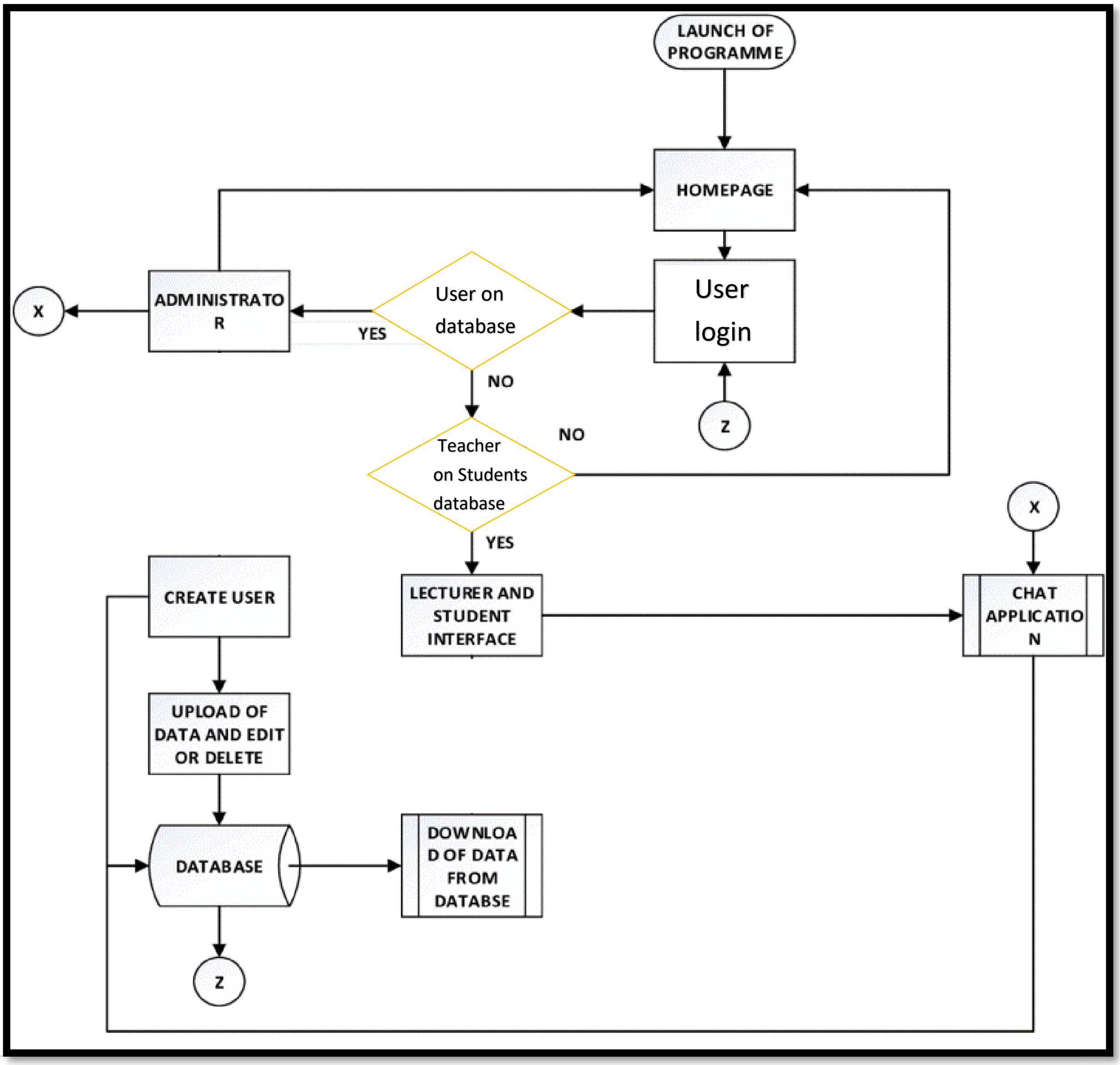
Please Enter your registered email id bellow

I have made extremly usefull notes in every semester which contains all the topics , if someone wants to buy those notes then they can contact me.

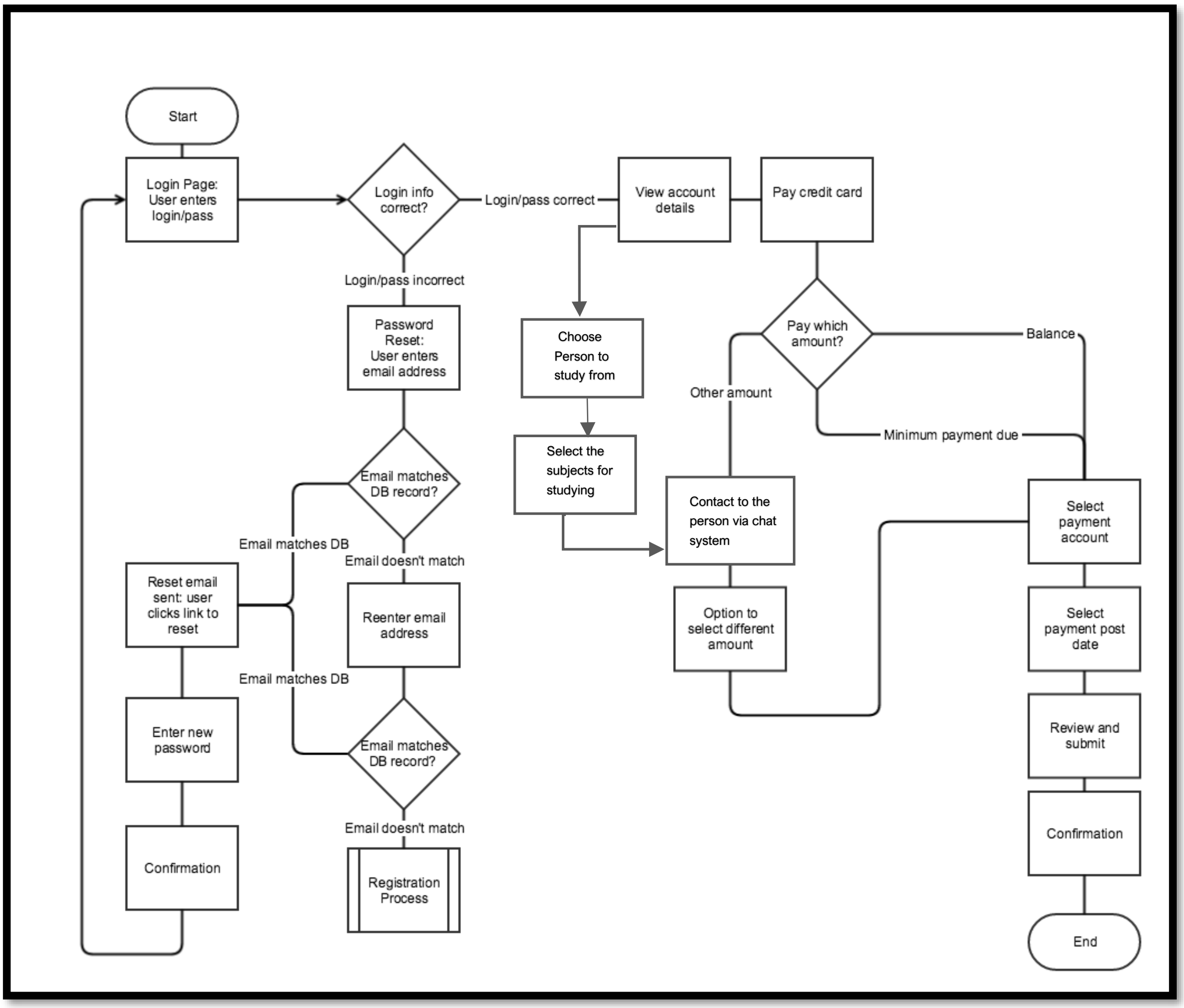
❖ App Architecture:



❖ Data-Flow diagram:


LAUNCH OF PROGRAMME
HOMEPAGE
User login
Z
User on database
YES
NO
ADMINISTRATO R
X
Teacher on Students database
NO
YES
LECTURER AND STUDENT INTERFACE
X
CHAT APPLICATIO N
CREATE USER
UPLOAD OF DATA AND EDIT OR DELETE
DATABASE
DOWNLOA D OF DATA FROM DATABSE
Z

❖ Flowchart:

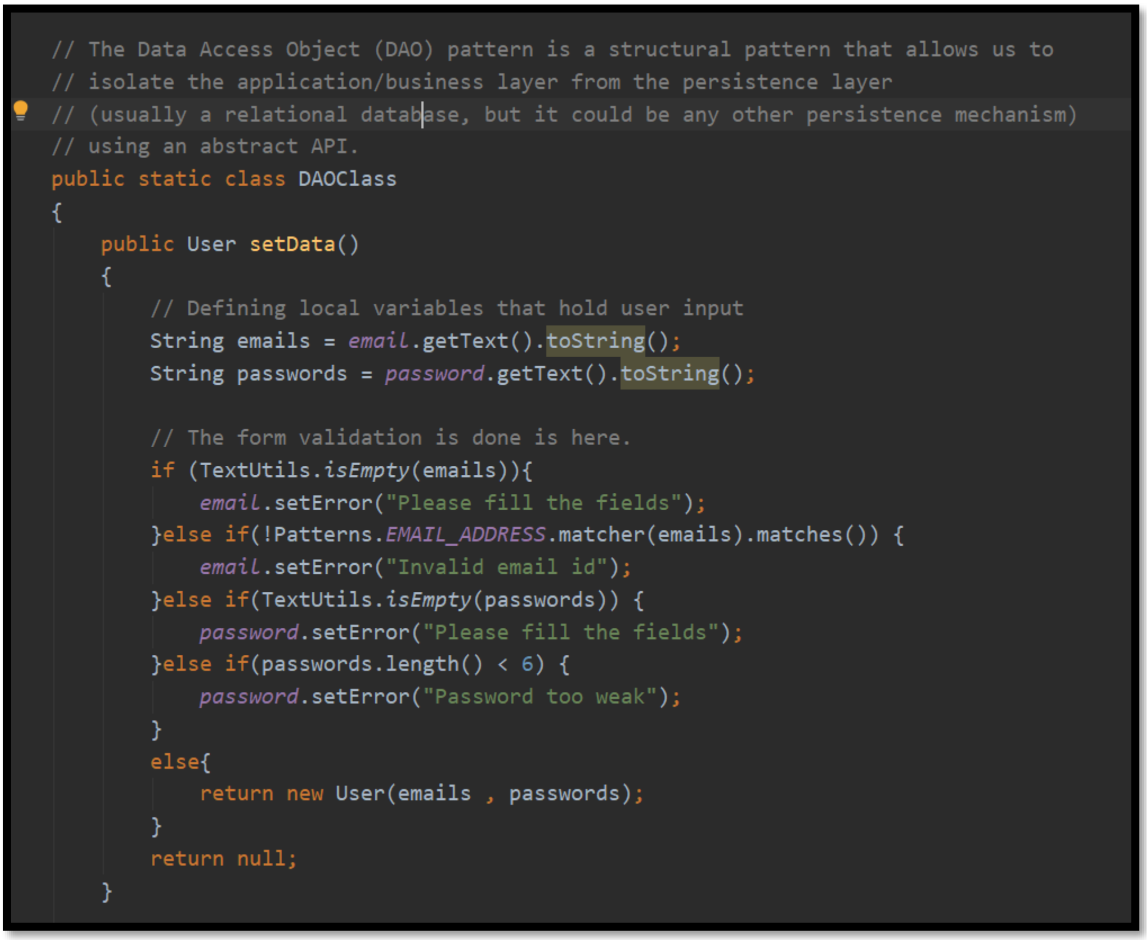


❖ Implementation:

The main functions and their functionality are mentioned in detail below:

- **Function 1:**
  Form Validation and DOA class

  The Data Access Object (DAO) pattern is a structural pattern that allows us to isolate the application/business layer from the persistence layer (usually a relational database, but it could be any other persistence mechanism) using an abstract API.



```java
// The Data Access Object (DAO) pattern is a structural pattern that allows us to
// isolate the application/business layer from the persistence layer
// (usually a relational database, but it could be any other persistence mechanism)
// using an abstract API.
public static class DAOClass
{
    public User setData()
    {
        // Defining local variables that hold user input
        String emails = email.getText().toString();
        String passwords = password.getText().toString();

        // The form validation is done is here.
        if (TextUtils.isEmpty(emails)){
            email.setError("Please fill the fields");
        }else if(!Patterns.EMAIL_ADDRESS.matcher(emails).matches()) {
            email.setError("Invalid email id");
        }else if(TextUtils.isEmpty(passwords)) {
            password.setError("Please fill the fields");
        }else if(passwords.length() < 6) {
            password.setError("Password too weak");
        }
        else{
            return new User(emails , passwords);
        }
        return null;
    }
```

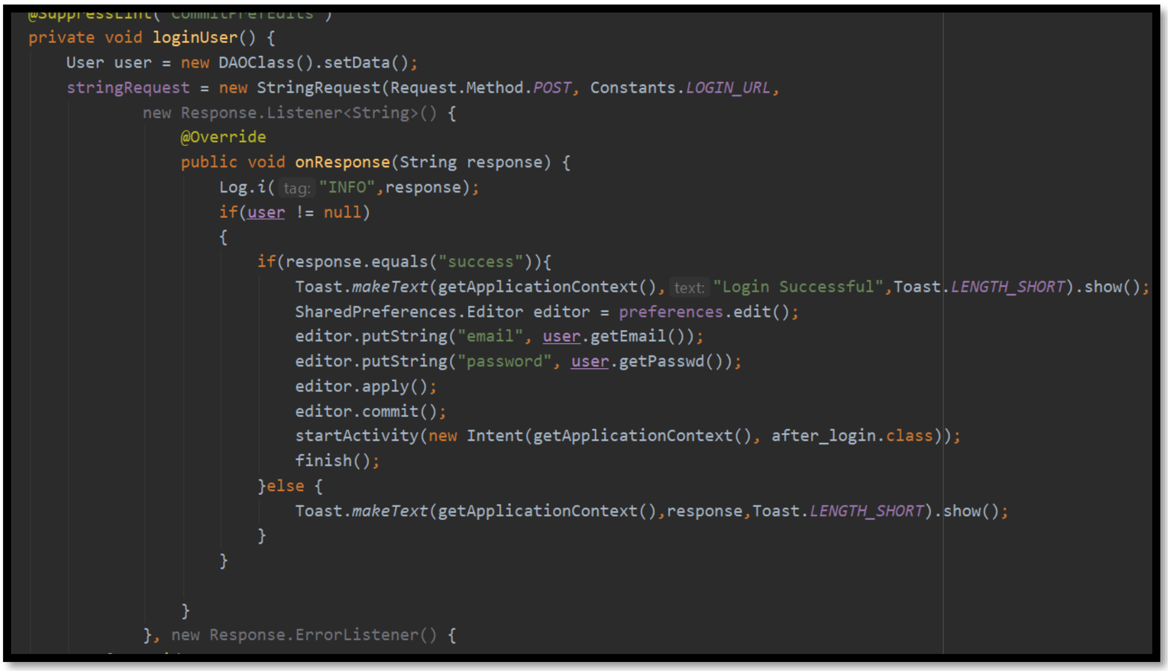
- **Function 2:**
  Shared Preferences

  Shared preferences allow us to store small amounts of primitive data as key/value pairs in a file on the device. To get a handle to a preference file, and to read, write, and manage preference data, I have used the Shared Preferences class.

  When we want get the values, call the getSharedPreferences() method. Shared Preferences provide modes of storing the data (private mode and public mode). It is for backward compatibility- use only MODE_PRIVATE to be secure.

  Nested Classes of Shared Preference:

1. SharedPreferences.Editor: Interface used to write(edit) data in the SP file. Once editing has been done, one must commit () or apply () the changes made to the file.
2. SharedPreferences.OnSharedPreferenceChangeListener(): Called when a shared preference is changed, added, or removed. This may be called even if a preference is set to its existing value. This call-back will be run on your main thread.

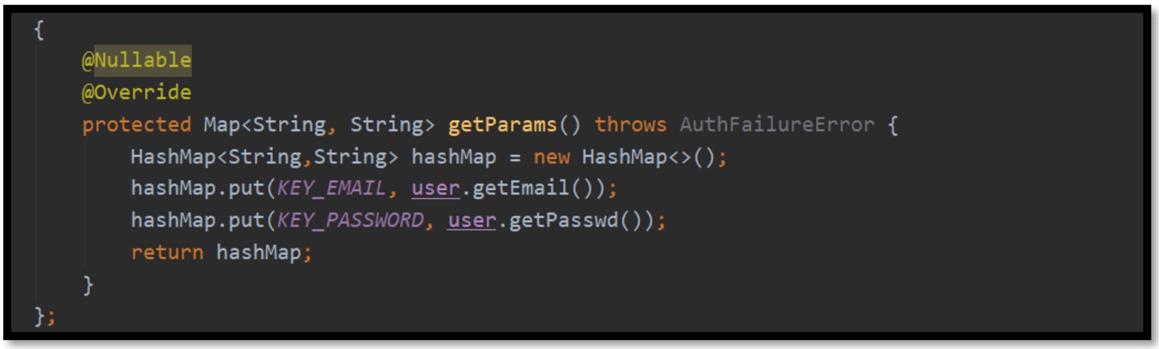


```
@SuppressLint("CommitPrefEdits")
private void loginUser() {
    User user = new DAOClass().setData();
    stringRequest = new StringRequest(Request.Method.POST, Constants.LOGIN_URL,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    Log.i( tag: "INFO",response);
                    if(user != null)
                    {
                        if(response.equals("success")){
                            Toast.makeText(getApplicationContext(), text: "Login Successful",Toast.LENGTH_SHORT).show();
                            SharedPreferences.Editor editor = preferences.edit();
                            editor.putString("email", user.getEmail());
                            editor.putString("password", user.getPasswd());
                            editor.apply();
                            editor.commit();
                            startActivity(new Intent(getApplicationContext(), after_login.class));
                            finish();
                        }else {
                            Toast.makeText(getApplicationContext(),response,Toast.LENGTH_SHORT).show();
                        }
                    }

                }
            }, new Response.ErrorListener() {
```

- **Function 3:**
  HashMap's

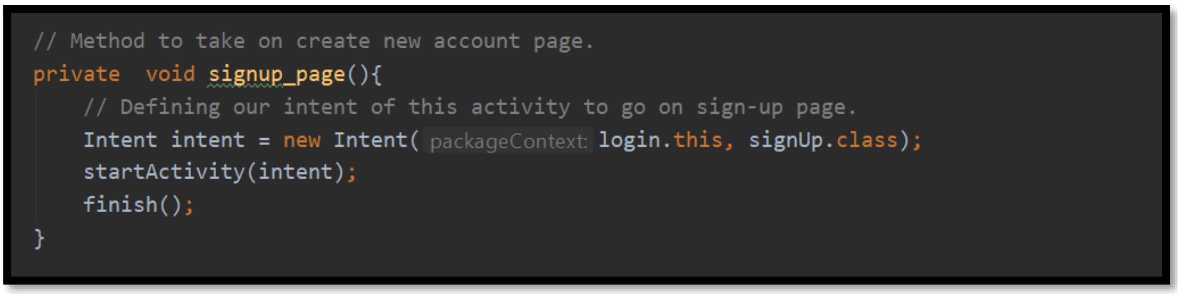
  A HashMap is a structure allowing one to store (key, value) items. A hash function pairs each key to an array index where the value will be stored. Complexity-wise, searching for an item stored in a HashMap is done in constant time on average. Moreover, it does not allow you to have primitive types as key or value (int, long, etc.).



```
{
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        HashMap<String,String> hashMap = new HashMap<>();
        hashMap.put(KEY_EMAIL, user.getEmail());
        hashMap.put(KEY_PASSWORD, user.getPasswd());
        return hashMap;
    }
};
```

- **Function 4:**
  Intents

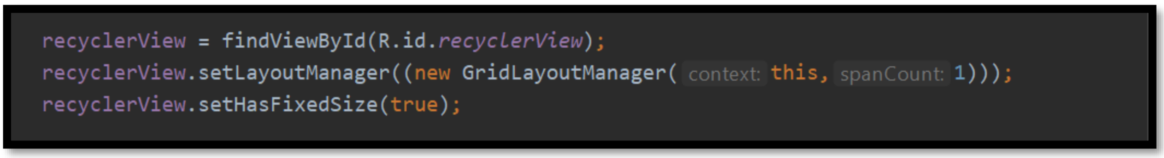
  Intent are the objects which is used in android for passing the information among Activities in an Application and from one app to another also. Intent are used for communicating between the Application components and it also provides the connectivity between two apps.



```
// Method to take on create new account page.
private  void signup_page(){
    // Defining our intent of this activity to go on sign-up page.
    Intent intent = new Intent( packageContext: login.this, signUp.class);
    startActivity(intent);
    finish();
}
```

- **Function 5:**
  RecyclerView

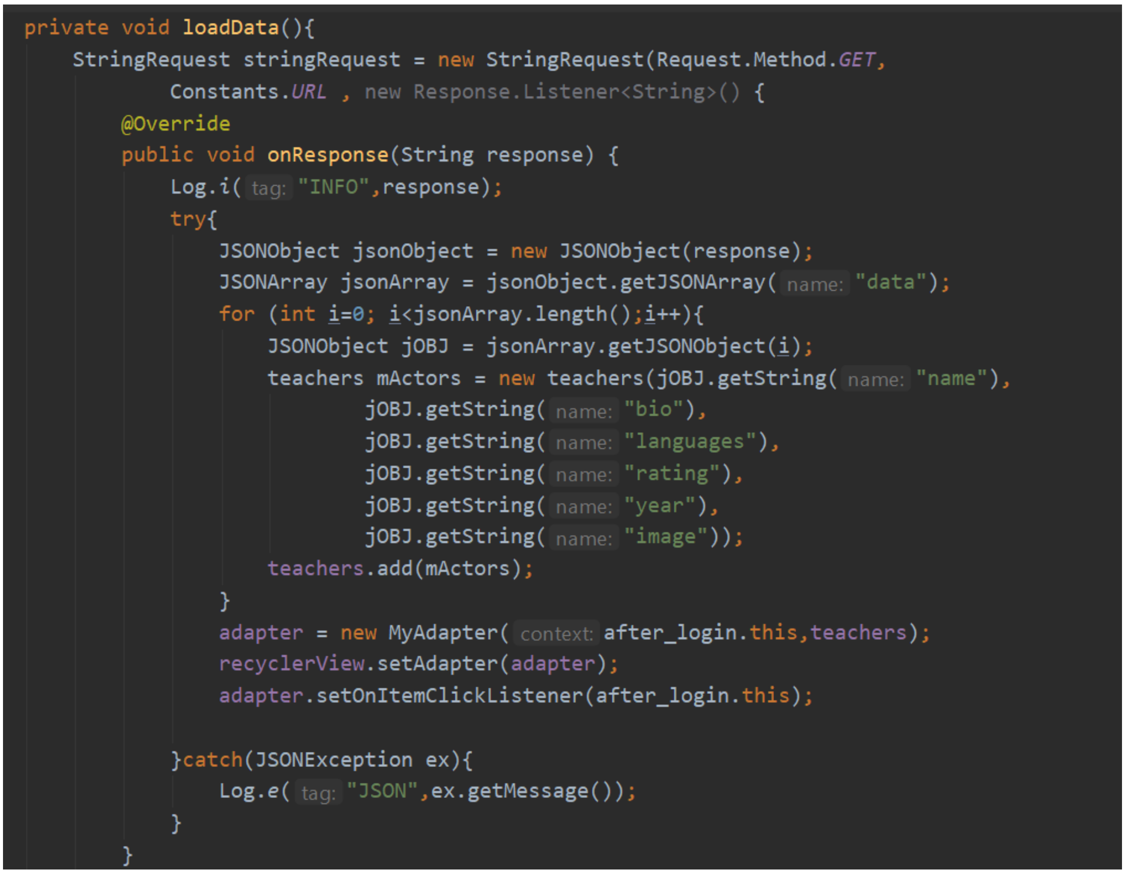
  The RecyclerView is a widget that is more flexible and advanced version of GridView and ListView. It is a container for displaying large datasets which can be scrolled efficiently by maintaining limited number of views. You can use RecyclerView widget when you have data collections whose elements change at runtime depend on network event or user action.



```
recyclerView = findViewById(R.id.recyclerView);
recyclerView.setLayoutManager((new GridLayoutManager( context: this, spanCount: 1)));
recyclerView.setHasFixedSize(true);
```

- **Function 6:**
  JSONObject

  JSON is used for data interchange (posting and retrieving) from the server.



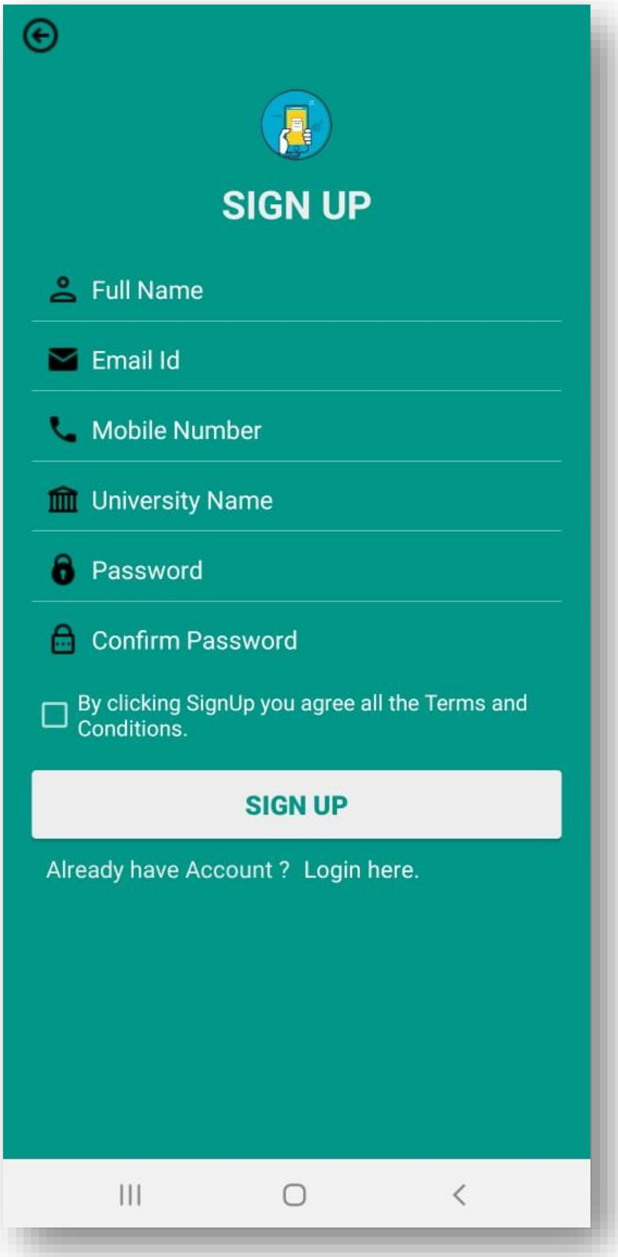
```
private void loadData(){
    StringRequest stringRequest = new StringRequest(Request.Method.GET,
            Constants.URL , new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.i( tag: "INFO",response);
            try{
                JSONObject jsonObject = new JSONObject(response);
                JSONArray jsonArray = jsonObject.getJSONArray( name: "data");
                for (int i=0; i<jsonArray.length();i++){
                    JSONObject jOBJ = jsonArray.getJSONObject(i);
                    teachers mActors = new teachers(jOBJ.getString( name: "name"),
                            jOBJ.getString( name: "bio"),
                            jOBJ.getString( name: "languages"),
                            jOBJ.getString( name: "rating"),
                            jOBJ.getString( name: "year"),
                            jOBJ.getString( name: "image"));
                    teachers.add(mActors);
                }
                adapter = new MyAdapter( context: after_login.this,teachers);
                recyclerView.setAdapter(adapter);
                adapter.setOnItemClickListener(after_login.this);

            }catch(JSONException ex){
                Log.e( tag: "JSON",ex.getMessage());
            }
        }
```

❖ Module-level testing (white box):

1) Checking for the 1st constraint i.e. only admin is given authoritical access to the system.

   ➢ Result:



SIGN UP

Full Name

Email Id

Mobile Number

University Name

Password

Confirm Password

☐ By clicking SignUp you agree all the Terms and Conditions.

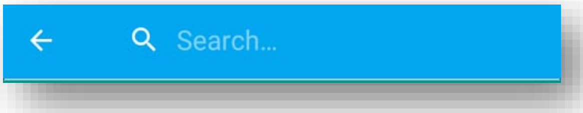
SIGN UP

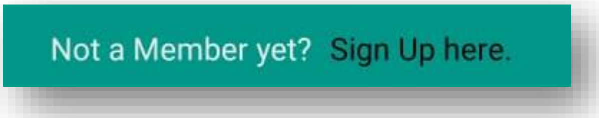
Already have Account ? Login here.

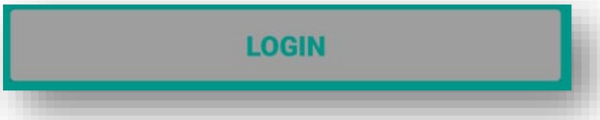
➢ Status:

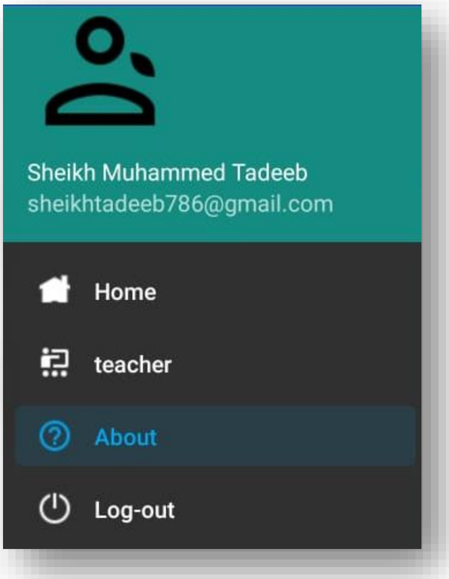
PASSED

2) Checking 2nd constraint i.e. Visibility of system status.







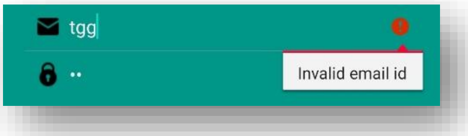


Clearly from above figs we can see the system is well updated i.e. Visibility of system status could be found.

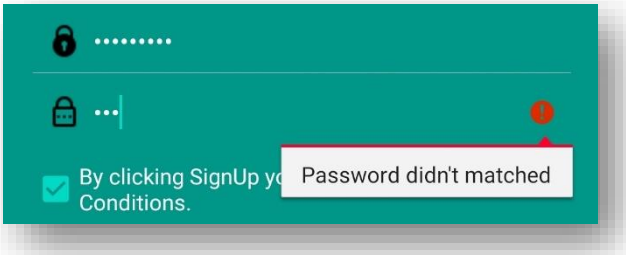
- Status:

PASSED

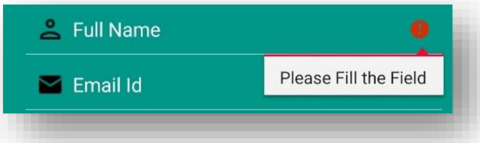
3) Checking the 3rd constraint i.e. Error-Prevention.



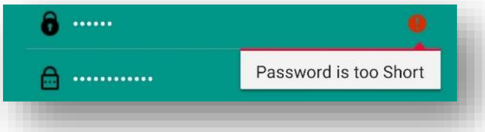
Invalid email id



Password didn't matched



Please Fill the Field

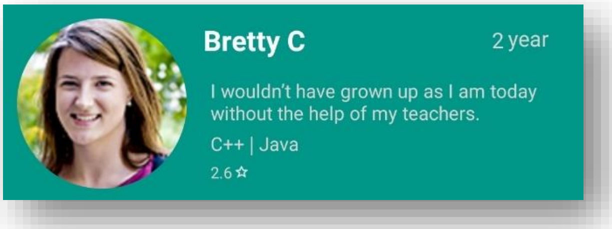


Password is too Short

- Status:

PASSED

4) Re-use rather than recall

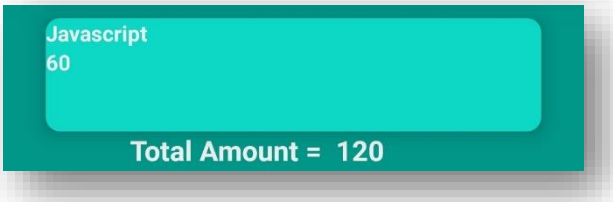

Bretty C

2 year

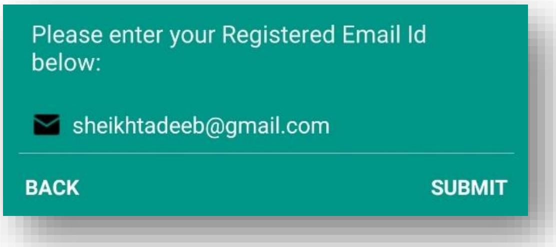
I wouldn't have grown up as I am today without the help of my teachers.

C++ | Java

2.6 ☆


Javascript
60

Total Amount = 120

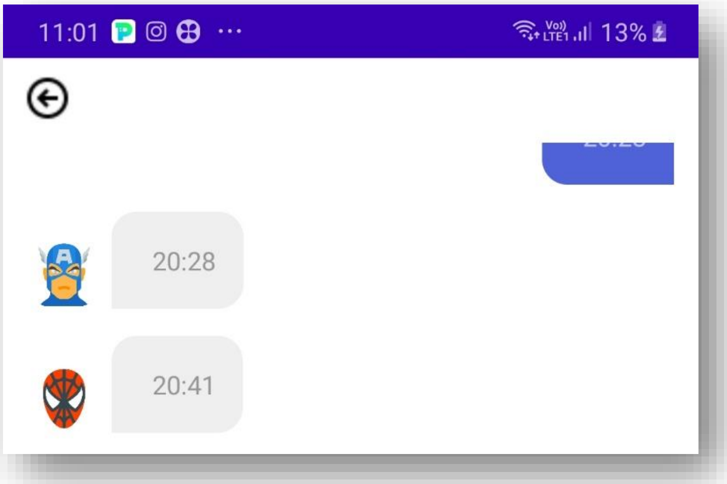

Please enter your Registered Email Id below:

sheikhtadeeb@gmail.com

- Status:

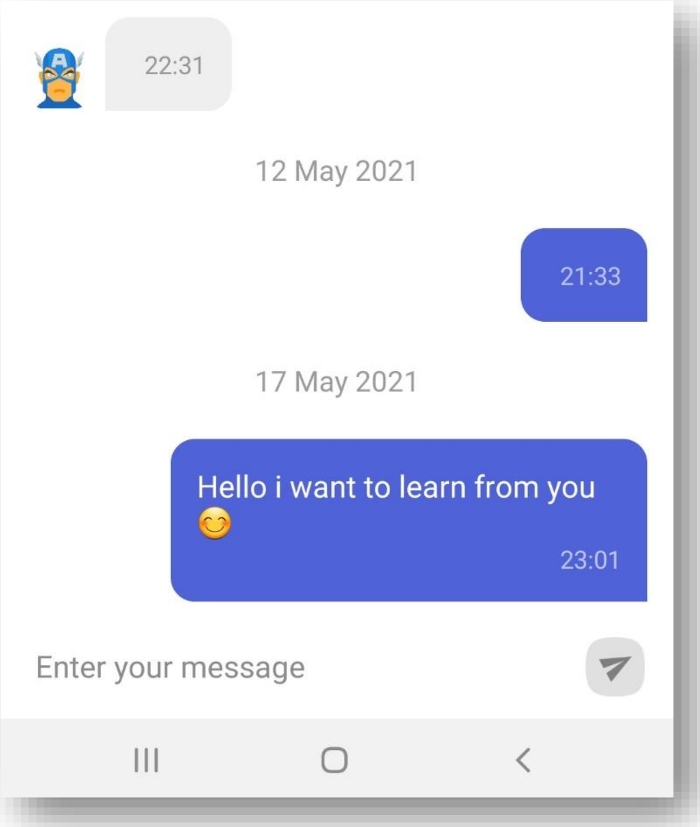
PASSED

5) The user can look up about the person and send the message to the person from whom the user wants to study.



20:28

20:41



22:31

12 May 2021

21:33

17 May 2021

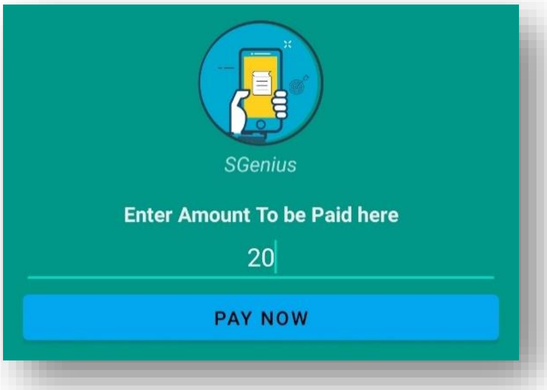
Hello i want to learn from you 😊

23:01

- Status:

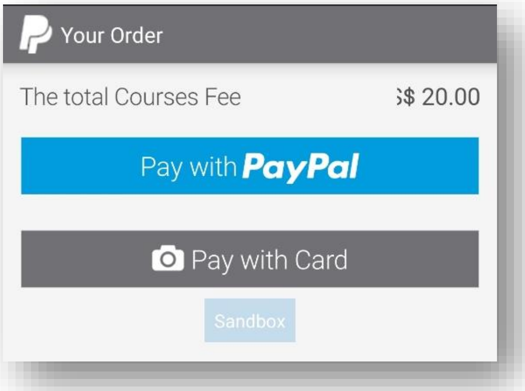
PASSED

6) The user can select the topics which he wants to study and then pay the teacher via the app.



Enter Amount To be Paid here

20



Your Order

The total Courses Fee $ 20.00

- Status:

✓ PASSED

Features and Operations of Application:

Snos.	Features & Operations	Implementation Status
1.	It will be on when internet connection is there.	✓
2.	Google Sign-in.	✓
3.	Visibility of System Status.	✓
4.	Aesthetics and Minimalist design.	✓
5.	Match Between system and real-world.	✓
6.	Easy to use.	✓
7.	User control and Freedom	✓
8.	Error Prevention	✓
9.	Reuse rather than recall	✓
10.	Help and documentation	✓
11.	CRUD operation	✓
12.	Search based on year, rating and name	✓
13.	Payment Integration	✓
14.	Real-time Chat between users	✓

15.	Profile Upload	✓
16.	Register as tutor	✓
17.	Logout (Sessions)	✓
18.	JSON data dynamic calling from server	✓
19.	About us page	✓
20.	Inline link integration	✓
21.	Scanning codes	✓
22.	Deployment on Play store	✓

❖ Advance Technology:

1) Real Time Chat-System:

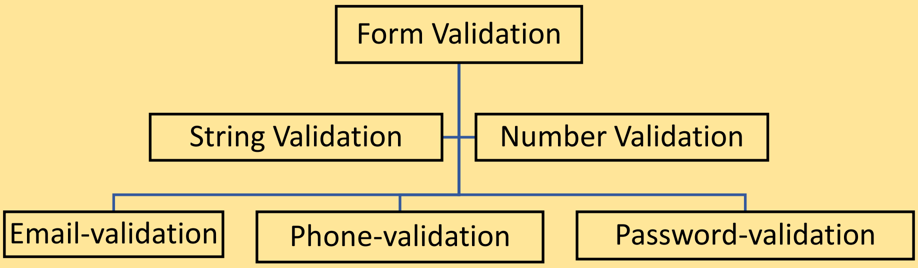
CometChat is a platform for integrating voice, video & text messaging experiences into your websites, web apps and mobile apps. We provide SDKs, APIs and UIs to enable you to quickly build a full-fledged chat solution.

2) Sandbox-Payment Integration:

Sandbox provides a comprehensive secure-test environment where you can test your technical integration with Worldpay. Sandbox supports the testing of both card payments and alternative payment methods. You can submit orders and test the payment life cycle - from initial order submission through to the SETTLED state. Sandbox simulates a production experience but in a shielded secure-test environment. I have integrated PayPal sandbox.

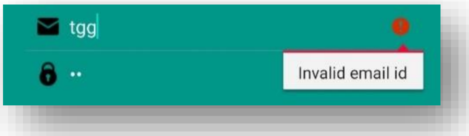
❖ Testing & Deployment:

- **Black Box testing points:**

S.nos	Black Box Testing Points
1	 Form Validation String Validation Number Validation Email-validation Phone-validation Password-validation
2	Maintained-Session.
3	Filtration of list based on names, rating and year.
4	Inline Payment integration.
5	Scanning of QR codes.
6	Inline opening of links.
7	Email generation.
8	Maintained Session with Google.
9	Enabling Multiple Roles. {i.e., teacher and student}.
10	Functional Cart.

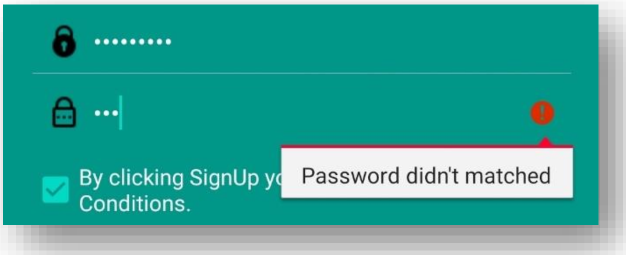
❖ Black Box Testing Validation:

➢ *Point 1: Form Validation*


tgg

··

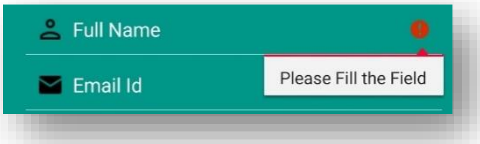
Invalid email id


··········

···

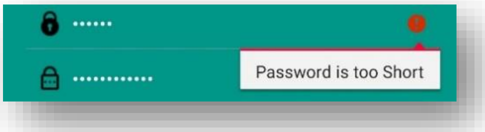
☑ By clicking SignUp y Conditions.

Password didn't matched


Full Name

Email Id

Please Fill the Field

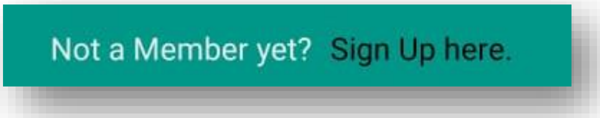

······

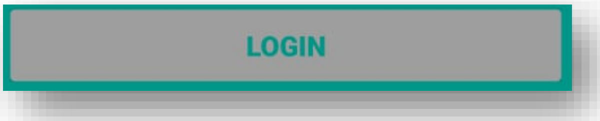
·············

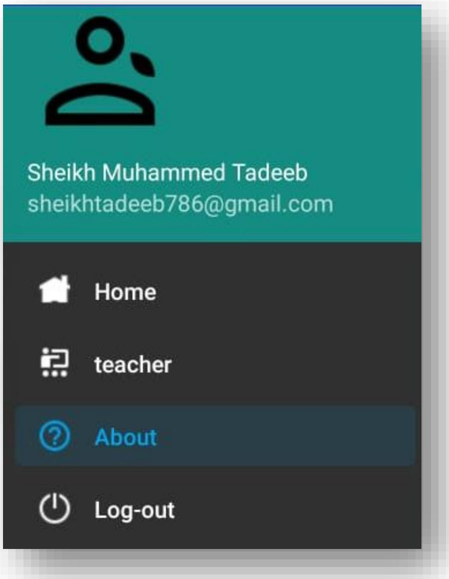
Password is too Short

➢ *Point 2: Maintained-Session*

Session are being maintained in the application i.e., there is proper login and logout carried out for a specific user and once the user logs-out he/she can’t return to the home-page by clicking the back button.

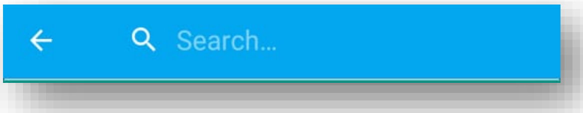




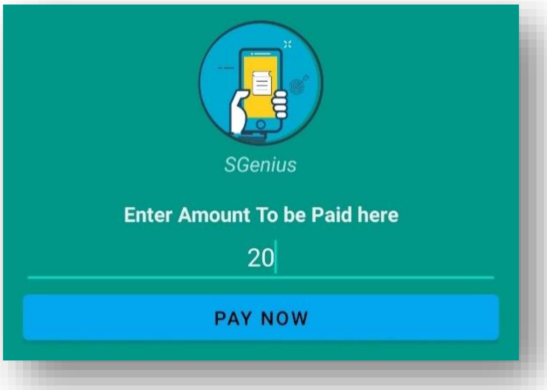

Sheikh Muhammed Tadeeb
sheikhtadeeb786@gmail.com

- ***Point 3: Filtration of list***

User can easily search people based on their names, year and rating from the search bar provided at the top of navigation page.

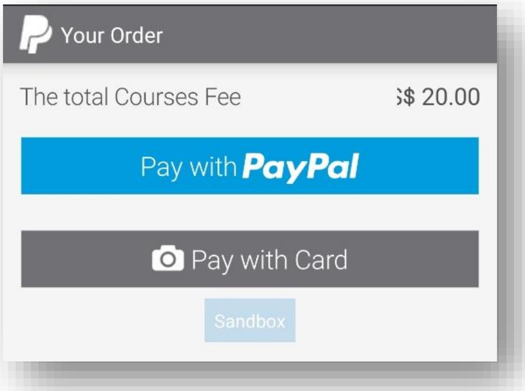


- ***Point 4: Inline-payment Integration***



Enter Amount To be Paid here

20

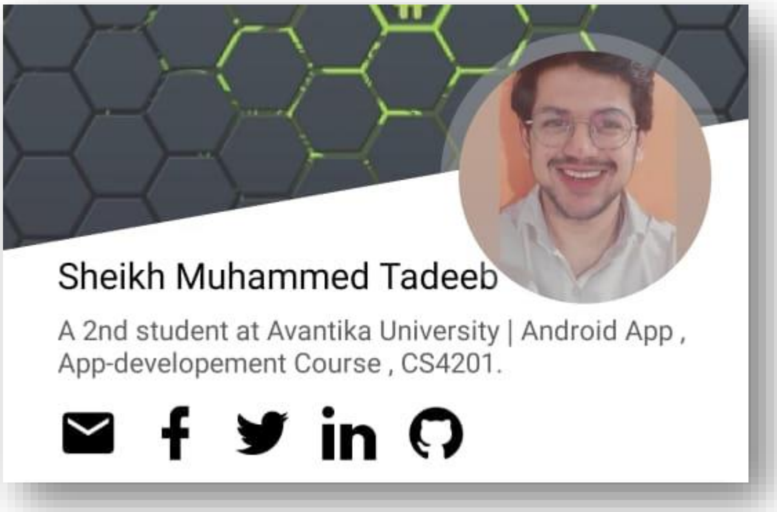


The total Courses Fee $\quad$ $ 20.00

- ***Point 5: QR Code Scanner***

The payment can also be made using a QR code scanner which is helpful in fast and easy transactions. Due to privacy policy of PayPal I have not included the screenshot here.

- ***Point 6: Inline-opening of links***

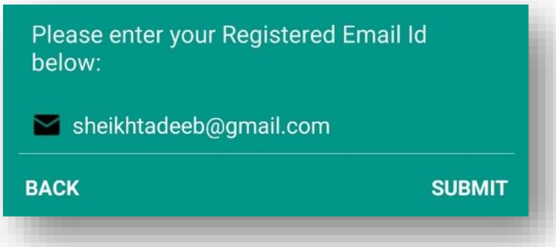


Sheikh Muhammed Tadeeb

A 2nd student at Avantika University | Android App , App-developement Course , CS4201.

- ***Point 7: Email-Generation***

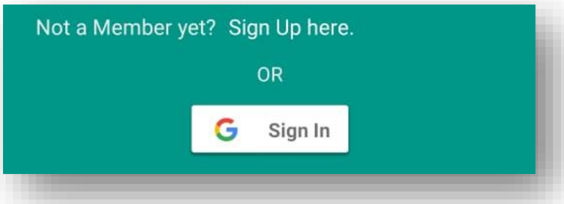
If user forgets the password, then the email will be sent to him with the password attached.



Please enter your Registered Email Id below:

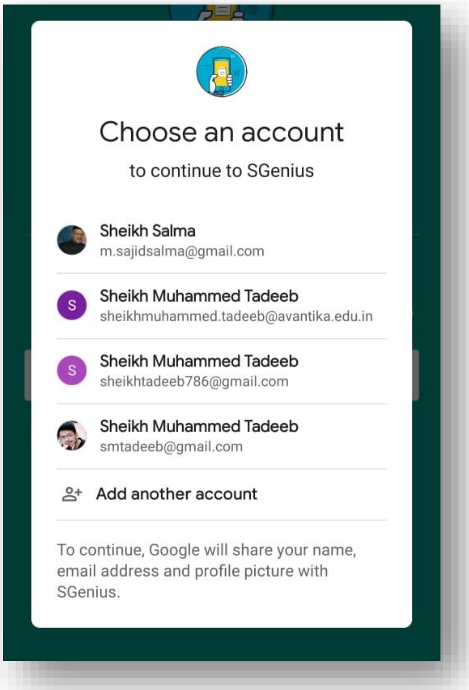
sheikhtadeeb@gmail.com

- ***Point 8: Maintained-Session with Google***



Not a Member yet? Sign Up here.

OR



Choose an account

to continue to SGenius

Sheikh Salma
m.sajidsalma@gmail.com

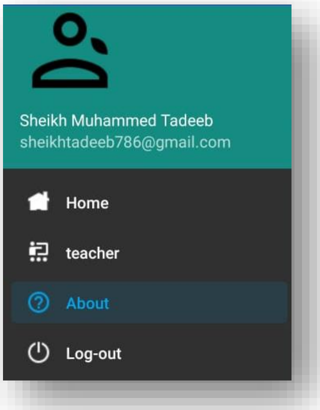
Sheikh Muhammed Tadeeb
sheikhmuhammed.tadeeb@avantika.edu.in

Sheikh Muhammed Tadeeb
sheikhtadeeb786@gmail.com

Sheikh Muhammed Tadeeb
smtadeeb@gmail.com

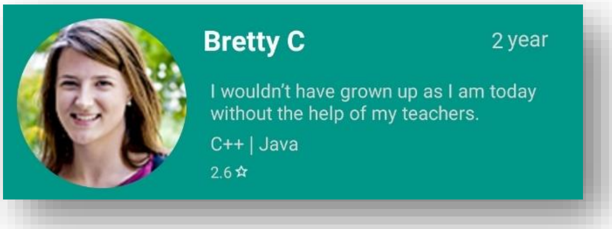
Add another account

To continue, Google will share your name, email address and profile picture with SGenius.



Sheikh Muhammed Tadeeb
sheikhtadeeb786@gmail.com

- ***Point 9:*** Enabling Multiple Roles. {i.e., teacher and student}.

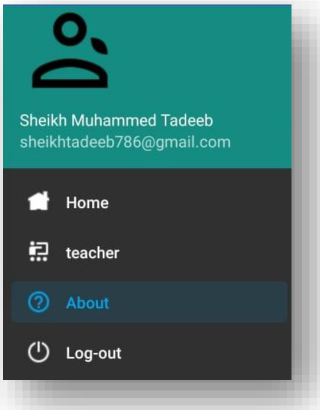


Bretty C 2 year

I wouldn't have grown up as I am today without the help of my teachers.

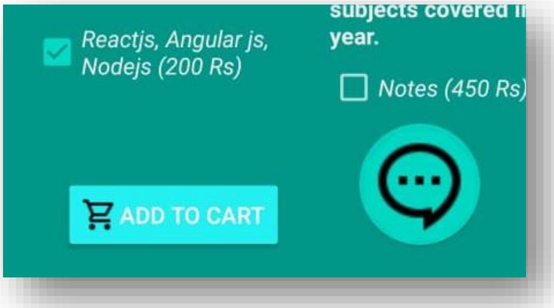
C++ | Java

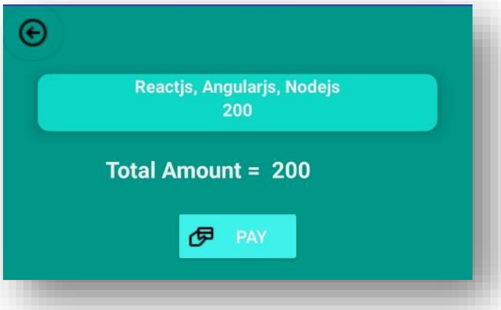
2.6 ☆



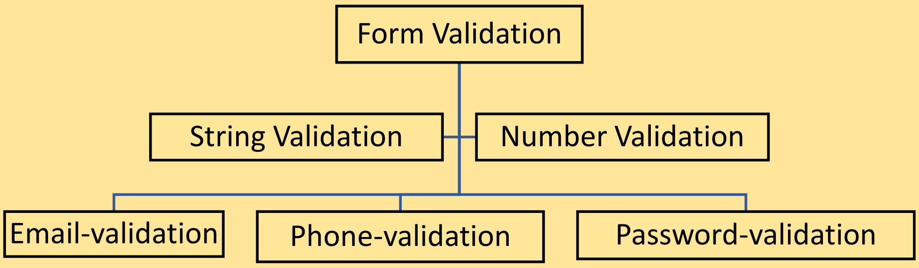
Sheikh Muhammed Tadeeb
sheikhtadeeb786@gmail.com

- ***Point 10:*** Functional Cart





❖ Black-Box testing result:

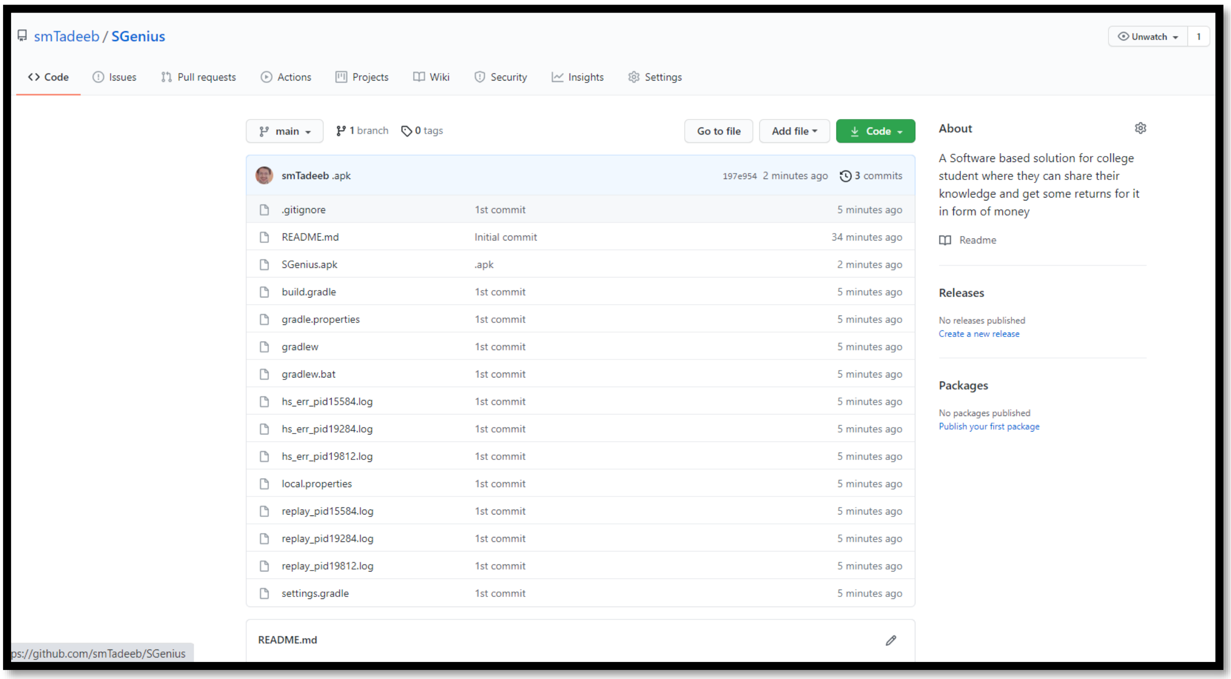
S.nos	Black Box Testing Points	Result
1	 Form Validation String Validation Number Validation Email-validation Phone-validation Password-validation	✓
2	Maintained-Session.	✓
3	Filtration of list based on names, rating and year.	✓
4	Inline Payment integration.	✓
5	Scanning of QR codes.	✓
6	Inline opening of links.	✓
7	Email generation.	✓
8	Maintained Session with Google.	✓
9	Enabling Multiple Roles. {i.e., teacher and student}.	✓
10	Functional Cart.	✓

❖ Result & Conclusion:

Initially I decided to make an application which could boost the teaching industry by providing a way where the juniors or learners can interact with their colleagues or seniors and can know their strong points and could learn from them.

I was successfully able to implement all the initial thoughts and parameters I had decided in my project. In future I will try to integrate university data also in the application

The Application has been deployed on GitHub. Click on the link.



smTadeeb .apk		197e954 2 minutes ago 3 commits
.gitignore	1st commit	5 minutes ago
README.md	Initial commit	34 minutes ago
SGenius.apk	.apk	2 minutes ago
build.gradle	1st commit	5 minutes ago
gradle.properties	1st commit	5 minutes ago
gradlew	1st commit	5 minutes ago
gradlew.bat	1st commit	5 minutes ago
hs_err_pid15584.log	1st commit	5 minutes ago
hs_err_pid19284.log	1st commit	5 minutes ago
hs_err_pid19812.log	1st commit	5 minutes ago
local.properties	1st commit	5 minutes ago
replay_pid15584.log	1st commit	5 minutes ago
replay_pid19284.log	1st commit	5 minutes ago
replay_pid19812.log	1st commit	5 minutes ago
settings.gradle	1st commit	5 minutes ago

README.md

About

A Software based solution for college student where they can share their knowledge and get some returns for it in form of money

Readme

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Hence the testing has been successfully done for various inputs and Sessions which is working perfectly fine.

❖ References:

1. https://www.journaldev.com/

2. https://developer.android.com/docs

3. https://richestsoft.com/mobile-application-development.html

4. http://www.vogella.com/tutorials/android.html

5. https://www.codementor.io/community/topic/android