Discipline Core

Design Project: Mobile Application

CS4201

Assignment 3

Sheikh Muhammed Tadeeb (AU19B1014)

❖ 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.

❖ Scope of project:

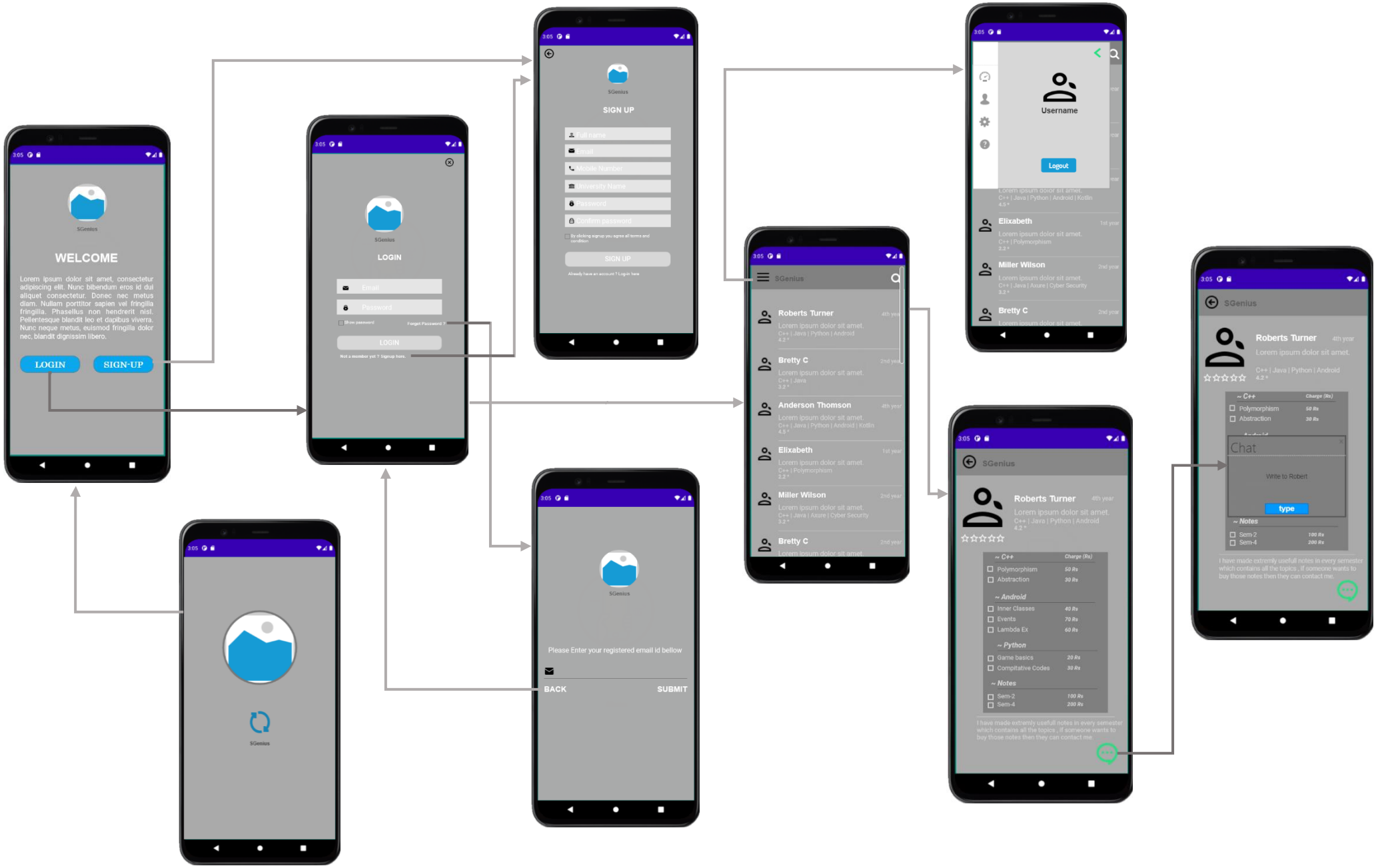
1) This app provides you with a great starting point for building a strong knowledge foundation.
2) In the app a secured user/student needs to register with a username and password for accessing the details of seniors and their contents and to interact with him.
3) Courses can be filtered based on rating, Names and their year.
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!

❖ 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.

Screen Flow and Wireframe decided in initial Phase



❖ 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.

```
// 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.

```
@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.

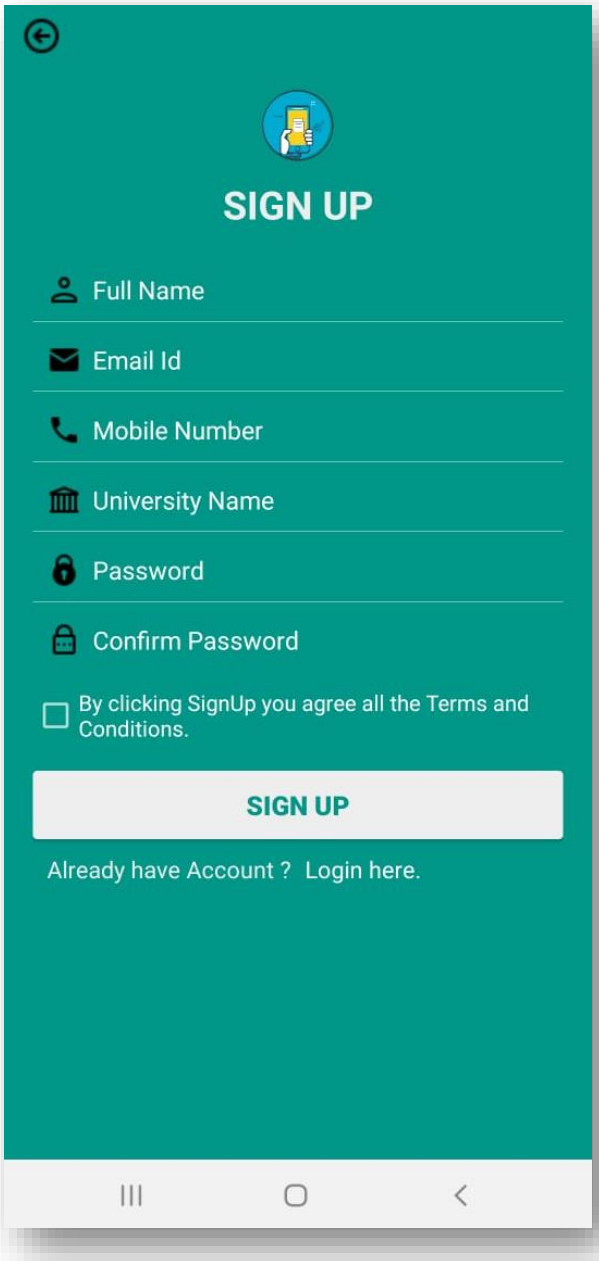
```
private void loadData(){
    StringRequest stringRequest = new StringRequest(Request.Method.GET,
            Constants.URL , new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.i( tag: "INFO",response);
            try{
                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):

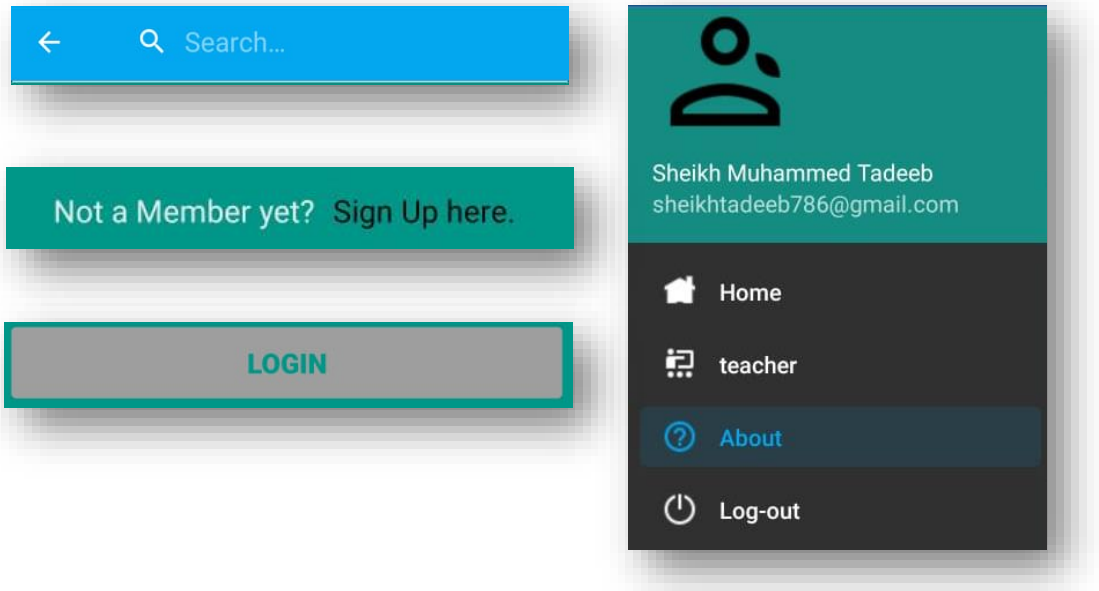
1) Checking for the 1st constraint i.e. only admin is given authoritical access to the system.

   ➢ Result:



   ➢ Status:

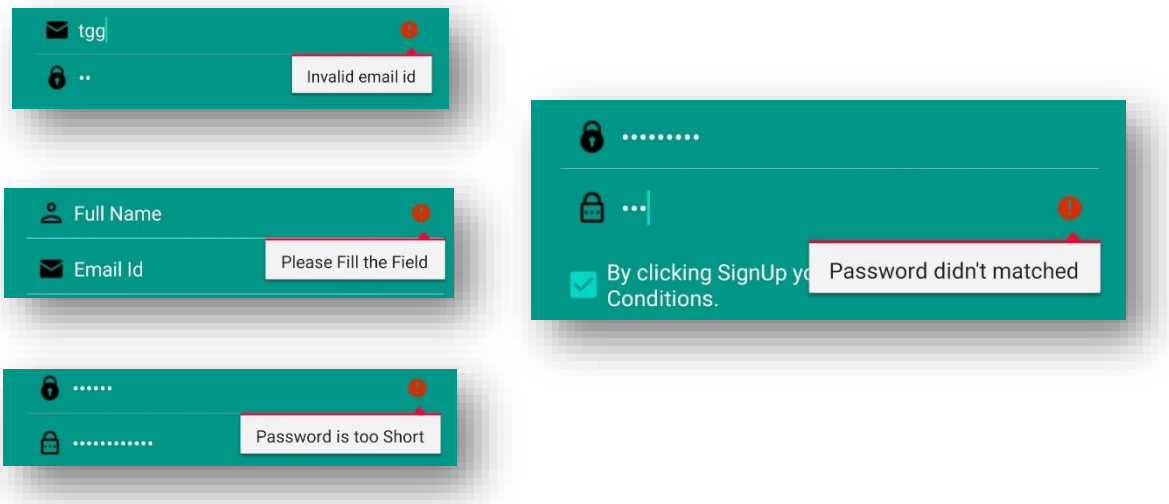
2) Checking 2nd constraint i.e. Visibility of system status.



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

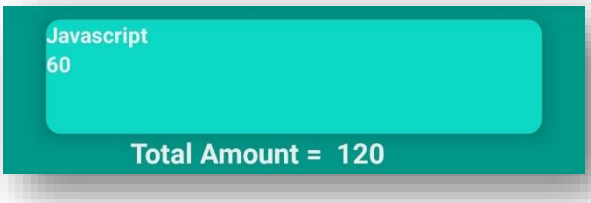
➢ Status:

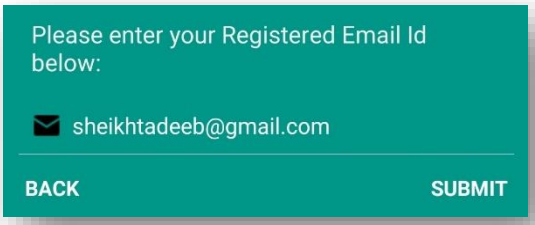
3) Checking the 3rd constraint i.e. Error-Prevention.



➢ Status:

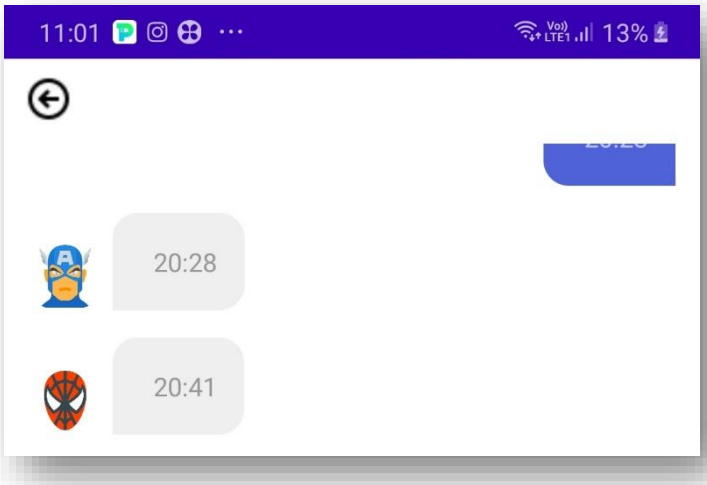
4) Re-use rather than recall

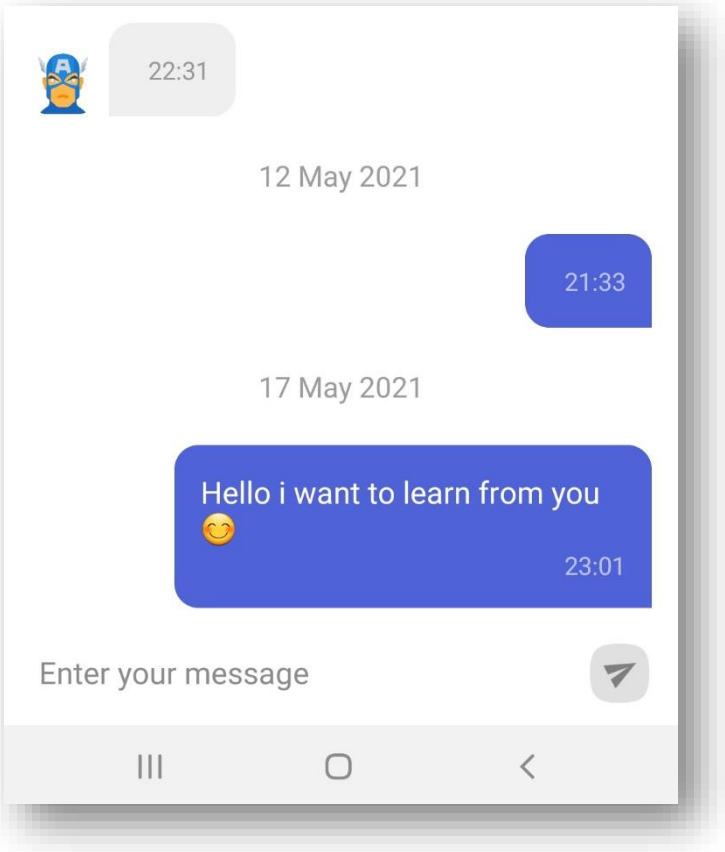




➢ Status:

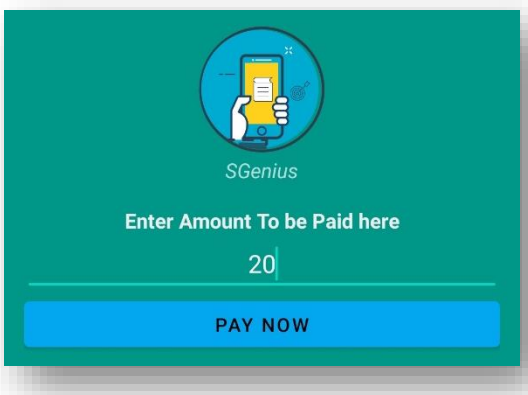
5) The user can look up about the person and send the message to the person from whom the user wants to study.

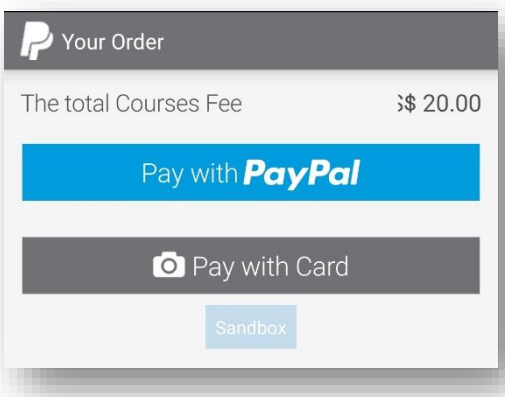




- Status:

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





- Status:

❖ 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.

❖ 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

❖ **Drive link (this is the link of my Project):**

Course_project_link (CLICK HERE)