

Pak Ancient diggers



BSSE Session:2017-2021

Submitted By:

Umair Akram

(Reg. No: 2017-BS-SE-173)

Hamayun Rashid

(Reg. No: 2017-BS-SE-164)

Supervisor:

Mr. Imran Farooq

Department of Computer Science and Information Technology

Faculty of Basic and Applied Sciences

University of Poonch Rawalakot

Final Approval

Department of Computer Science and Information Technology

This is to certify that we have read the project report submitted by **Umair Akram** Registration No 2016-BSSE-173, **Hamayun rashid** Registration No 2016-BSSE-164 it is our judgment that this report is of sufficient standard to warrant its acceptance by The University of Poonch, Faculty of Basic & Applied Sciences Rawalakot for BS in Software Engineering.

COMMITTEE:

1. External Examiner _____

Dr. Zahid Mahmood

Assistant professor

Department of CS&IT.

University of Kotli AJK.

2. Supervisor _____

Mr. Imran Farooq

Lecturer

Department of Computer Science & IT.

University of Poonch Rawalakot

3. Chairman _____

Pro Dr. Adnan Idris

Department of Computer Science & IT

University of Poonch Rawalakot.

Declaration

I hereby declare that this android application, neither as a whole nor as a part there of has been copied out from any source. It is further declared that I developed this application and this report entirely based on my personal efforts made under the sincere guidance of my project supervisor. If any part of this application proved to be copied or found to a report of some other, I shall stand by the consequences. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning. I further declare that this application and all associated documents, reports, and records are submitted as partial requirement for the Degree of BSSE.

Signature:

Umair Akram

Hamayun Rashid

Dedication

I dedicated this project to my respected and beloved **parents and teachers** who scarifies all the comforts of their lives for my future and whose love, affection and guidance have always guided me to face the challenges of life with patience and courage and their guidance inspired me throughout my life to get my destination, especially **Sir Imran Farooq**, and he helped me throughout the project, encouraged me and make it possible for me, so that I can achieve my goals.

Acknowledgement

This report would not have been possible without the contribution and collaboration of others. Our sincere gratitude:

- To Almighty ALLAH who granted me health and long life, without which i could not have finished this report;
- To my supervisor **Imran Farooq** for his guidance and constant supervision as well as providing necessary information regarding the project and also for his support in completing the project
- To all teachers and parents for their kind cooperation and encouragement in completion of this project.
- To all those who provided us the possibility to complete this project.
- To our colleagues in developing the project and people who have willingly helped out with their abilities.
- Lastly, I also appreciate the guidance given by other supervisors as well as the viva committee especially in our project presentation thanks to their comments and advices.

Name: Umair Akram

Name: Hamayun Rashid

Project in Brief

Project Title	Pak Ancient diggers
Objective	The primary objective of the project is to provide the most effective about the problem statement and its proposed solution. The idea is to develop an android application for better guidance and saving expensive things.
Developed By	Umair Akram 30 Hamayun Rashid 18
Supervised By	Imran Farooq
Tool Used	Visual Studio 2019 Android Studio(version 4.2) MY SQL Database
Operating system	Windows 10

Table of Content

CHAPTER 01	07
INTRODUCTION	08
PURPOSE	08
SCOPE	08
PROBLEM STATEMENT	09
Earlier issues:	
PROPOSED SOLUTION	09
Efficiency	
User Friendly Interface	
Flexibility	
Scalability	
CHAPTER 2	11
SYSTEM ANALYSIS	11
Introduction	12
Requirements Analysis:	13
Functional Requirements:	
Non-Functional Requirements	
Accessibility:	
SYSTEM REQUIRMENTS	14
Softwarerequirements	
HardwareRequirements	
Use case Diagram	15
Use Case Description	
CHAPTER 3	25
System Design	25
Activity Diagrams	26
Sequence Diagrams	34
Class Diagram	38
Entity Relation Diagram	39
CHAPTER 4	40
IMPLEMENTATION	40
Register	
Adapter manage all	

Add account	
Add encyclopedia	
Add payment	
Add tour guide	
Admin dash board	
Admin place adapter	
All assign places data class	
All contact adapter	
All guide activity	
All guide adapter	
All payment activity	
All places activity for user	
All places	
Chat	
Chat activity	
Data class all payment	
Data class encyclopedia	
Data class manage all	
Data class org contact	
Local people dash board	
Manage all	
Message	
Organization contact list	
Organization dashboard	
Payment slip	
Post finding activity	
Report activity	
Search Encyclopedia activity	
Chapter 5.....	151
TESTING.....	147
CHAPTER 6.....	154
Screenshots.....	

List of Figures

Figure 2. 1 Use case diagram.....	15
Figure 3.0 Activity diagram.....	26
Figure 3. 1 login	
Figure 3. 2 Update profile	
Figure 3. 3 Generate report	
Figure 3. 4 Feedback	
Figure 3. 5 Search	
Figure 3. 6 Chat	
Figure 3. 7 Logout	
SEQUENCE DIAGRAM.....	34
Figure 3. 8Login	
Figure 3. 9Update	
Figure 3.10 Generate Report	
Figure 3.11 Search	
Figure 3.12 Chat	
Figure 3.13 Payment	
Figure 3.14 Logout.....	
Figure 3. 15 <i>Class diagram</i>	38
Figure 3. 16 Entity Relation Diagram.....	39
Screenshots.....	154
Figure 6.1 to 6.10.....	

List of Tables

Table 2. 1	FUNCTIONAL REQUIREMENTS	
Table 2. 2	Usecase description (Register).....	16
Table 2. 3	(login).....	17
Table 2. 4	(Delete Ancient Relics Organization).....	18
Table 2. 5	(Block Ancient Relics Organization).....	19
Table 2. 6	(add Encyclopedia)	20
Table 2. 7	(Delete Tourist).....	21
Table 2. 8	(Block Tourist).....	22
Table 2. 9	(Block People)	23
Table 2. 10	(chat).....	24
Table 5.0	TESTING.....	
Table 5.1	user authentication.....	
Table 5.2	Block user.....	
Table 5.3	Update profile.....	

CHAPTER 1

INTRODUCTION

INTRODUCTION

Pak Ancient digger will be an Android app. Which will be a best diggers (if any older discoveries during excavation in an area they can be informed of archeological through) for all ancient relics organization in Pakistan. Local people will contact to Ancient relics organization for informing what they have discovered. It will be a best Wikipedia (include brief history of Pak historical places with pictures in different languages) for users. It will be a best guider for tourist by providing proper information regarding historical places and museums with location. It will provide professional guiders to tourists (which would guide them throughout the journey. It will be a best way of earning for professional guiders through this App. Tourist must have to pay them after selecting the professional guider through different mediums.

PURPOSE

Fundamental objective of purpose project is to further discover the antiquities as well as to make the people outside and abroad aware of the historical places of our country which can make our country economically better. Our own people will also become aware of the history of their country things. It has become easier for archaeologists to decide where and where to dig. Providing a way of earning for professional guiders through this app by representing their experience letter of archaeology.

SCOPE

In our country there is a lot of historical places and museums. People of our country consider these places only picnic points they have no knowledge about history of these places or things. Also our country having lot of expensive things buried under soil but people have no interest or equipment to rescue them. Admin will register Ancient relic's organizations for their further discoveries, and also register the tourists and tourists guiders after log in process but users can also log in to gain more knowledge about ancient civilizations and places, Ancient relics organization will registered the people who provided them information about new places for discoveries through the contact provided by Organization. Tourists who will interested in visiting the historical places and museums with complete guidense they will log in and then visit the profiles of tourists guiders through it and then contact to them which will provided by guiders on their

account. Payment procedure will be included also which includes tourists to pay their guiders. This all payments will be recorded in backend by Admin.

PROBLEM STATEMENT

Before there was no proper App especially about Pak historical places and museums and no proper platform for tourists guiders to earn money safely and securely the more important that also there was no proper platform for archaeologists to know more about their ancient relics from ruins and others smugglings, through this App public contact with them to inform them about new findings and also a great opportunity for users to learn more about Pak historical places and museums history with detail description.

Earlier issues:

Before there was no proper platform for Ancient relics organization to know about new findings throughout this app they will be eligible to find more archaeological things throughout Pakistan with help of public and also no proper way of earning for tourist guiders and also difficult for tourists to know about historical places in details.

PROPOSED SOLUTION

Through this App it will be easy to learn about the past history by viewing encyclopaedia, by visiting the historical places and museums with trained and experienced guiders. Also an easy way of saving more things. Through this app guiders can earn safely and guide people better.

Efficiency

Easy process just sign in with your id, name, password. Click on desired service option and then desired service screen will display to take order/ information. Changing screens will require very little computation and thus will occur very quickly.

User Friendly Interface

The graphical user interface is to be designed with usability as the first priority. The project will be presented and organized in an easy way that is both visually appealing and easy for the user.

Flexibility

Extensive flexibility is provided as we have considered the aspect that the people using our App can easily understand.

Scalability

Our system is scalable if the number of users increases on the network then by optimizing servers we can handle the server they hold i.e. as in case of an increase in the number of users the system remains consistent.

CHAPTER 2
SYSTEM ANALYSIS

SYSTEM ANALYSIS

Introduction

Pak Ancient digger will be an Android app. Which will be a best diggers (if any older discoveries during excavation in an area they can be informed of archeological through) for all ancient relics organization in Pakistan. Local people will contact to Ancient relics organization for informing what they have discovered. It will be a best Wikipedia (include brief history of Pak historical places with pictures in different languages) for users. It will be a best guider for tourist by providing proper information regarding historical places and museums with location. It will provide a professional guiders to tourists (which would guide them throughout the journey. It will be a best way of earning for professional guiders through this App. Tourist must have to pay them after selecting the professional guider through different mediums.

Requirements Analysis:

The following section presents the complete set of functional and non-functional requirements. Functional requirements are listed first, according to their relationship to the overall system. The non-functional requirements that pertain to accuracy, reliability, accessibility, usability, maintenance and performance are subsequently presented.

Functional Requirements:

The functional requirements of the system describe the functionality or services that the system is expected as provide.

Table 2.1 FUNCTIONAL REQUIREMENTS

ID	Requirements	Justification
Admin:		
REQ-1	Register	Should register first
REQ-2	Login	After registration must be login
REQ-3	Manage Ancient Relics Organization	Can manage Ancient Relics Organization
REQ-4	Add Encyclopedia	Can add Encyclopedia
REQ-5	Manage Tourist	Can Manage Tourist, their detail

REQ-6	Manage Local People	Can add , delete/block local people
REQ-7	Chat	Can chat with other
REQ-8	Register Transporter	Can register transports
REQ-9	Generate Report	Can generate repors
REQ-10	Logout	After comlete his work must be logout
Ancient Relics Organization		
REQ-11	Register	Should be register first
REQ-12	Login	Login first
REQ-13	Chat	Can chat with others
REQ-14	Edit profile	Can edit his profile
REQ-15	Feedback	Ca n give feedback to other
REQ-16	Logout	Must logout after done
Local People		
REQ-17	Register	Register first
REQ-18	Login	Must be login
REQ-19	Edit profile	Can edit his profile
REQ-20	View ancient relics organization	Can view ancient relics organizations
REQ-21	View encyclopedia	Can view encyclopedia
Tourists		
REQ-22	Register	Register first
REQ-23	Login	Must be login
REQ-24	View encyclopedia	Can view encyclopedia
REQ-25	Feedback	Can give feedback
REQ-26	Chat	Can chat with others
REQ-27	Logout	Logout
Tourist Guiders		
REQ-28	Register	Register first
REQ-29	Login	Should be login
REQ-30	Edit Profile	Can edit his profile
REQ-31	Receive Payments	Can receive payments from tourists
REQ-32	logout	Logout

Non-Functional Requirements

Non-functional requirements are the quality requirements that stipulate how well a software does what it has to do. These are quality attributes of any system; these can be seen at the execution of the system and they can also be the part of the system architecture.

Accuracy:

The system will be accurate and reliable based on the design architecture. If there is any problem in the accuracy then the system will provide alternative ways to solve the problem.

Usability:

The proposed system will be simple and easy to use by the users. The users will comfort in order to communicate with the system. The user will be provided with an easy interface of the system.

SYSTEM REQUIRMENTS

Softwarerequirements

- Android Studio 4.3.1
- Android Software Development Toolkit (SDK)
- Java 1.6 runtime
- Gradle 5.1.1
- J2me
- Asp.net(MVC5)
- SQLServer12/1

HardwareRequirements

- Operating System: Android
- Processor Speed: 1.0 GHz minimum
- RAM: 512 MB
- Memory: 2 GB
- Smart Phone – Android Honey Comb.

Use case Diagram

This section will provide use case diagrams of the proposed application. Use case diagrams are considered as analysis tool which help in gathering/soliciting requirements of the application. A use case diagram is a representation of a user's interaction with the application that shows the relationship between the user and the different use.

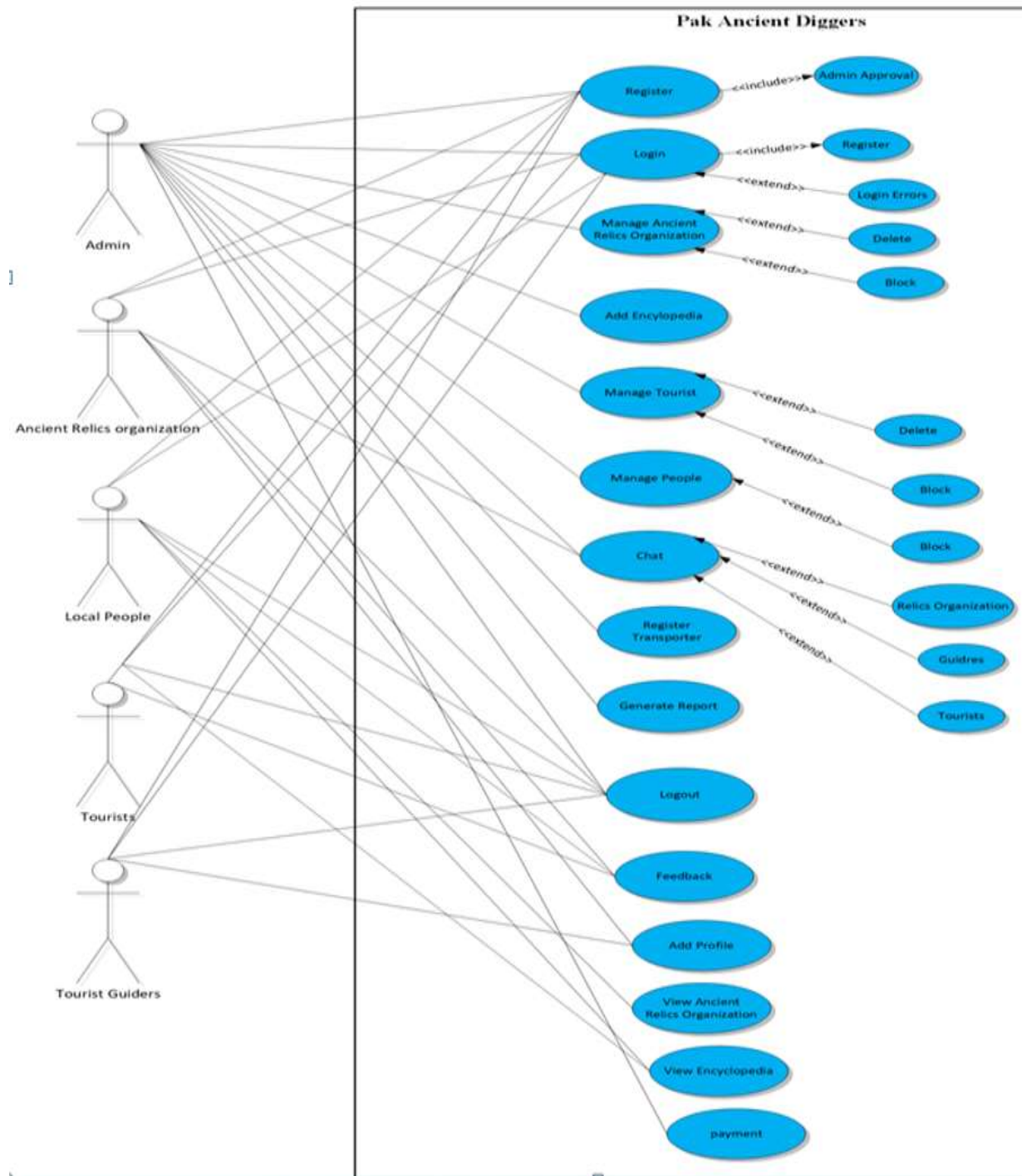


Figure 2.1 Use case diagram

Use Case Description

Table 2.2(Register)

Use Case Section	Comments	
Use Case Name	Register	
Scope	Pak Ancient Diggers	
Primary Actor	Admin,Ancient Relics Organization, local people, tourists, tourist guiders, System	
Level	User goal	
Pre-condition	Actor should be sign up into the system	
Post condition	Actor successfully sign up	
	Actor Action 1. Actor request for signup 3. Actor provides the information and request to sign up.	System Action 1. System as for informations 4. System successfully signup actor
Extension and Alternatives	4.(a) Invalid informations 4.(b) Re-enter user information	

Table 2.3(login)

Use Case Section	Comments	
Use Case Name	Login	
Scope	Pak Ancient Diggers	
Primary Actor	Admin,Ancient Relics Organization, local people, tourists, tourist guiders, System	
Level	User goal	
Pre-condition	Actor should be sign up into the system	
Post condition	Actor successfully login	
	<p>Actor Action</p> <p>1--Actor request for login</p> <p>3--Actor provides the information and request to login.</p>	<p>System Action</p> <p>2-- System as for informations</p> <p>4--System successfully login actor</p>
Extension and Alternatives	<p>4.(a) Invalid informations</p> <p>4.(b) Re-enter user information</p>	

Table 2.4(Delete Ancient Relics Organization)

Use Case Section	Comments	
Use Case Name	Delete Ancient Relics Organization	
Scope	Pak Ancient Diggers	
Primary Actor	Admin, System	
Level	User goal	
Pre-condition	Actor should be authorized to delete personality	
Post condition	Actor successfully delete personality	
	<p>Actor Action</p> <p>1-- admin request to delete personality</p> <p>3-- Admin delete the asked information</p> <p>4-- Admin confirm the deletion</p>	<p>Actor Action</p> <p>1-- admin request to delete personality</p> <p>3-- Admin delete the asked information</p> <p>4-- Admin confirm the deletion</p>
Extension and Alternatives	<p>4.(a) Invalid informations</p> <p>4.(b) Re-enter user information</p>	

Table 2.5(Block Ancient Relics Organization)

Use Case Section	Comments	
Use Case Name	Block Ancient Relics Organization	
Scope	Pak Ancient Diggers	
Primary Actor	Admin, System	
Level	User goal	
Pre-condition	Actor Should be authorised person	
Post condition	Actor successfully Block Ancient Relics Organization	
	<p>Actor Action</p> <p>1-- admin request to Block Ancient Relics Organization</p> <p>3-- Admin update the asked information</p> <p>4-- Admin confirm the Block</p>	<p>System Action</p> <p>2-- System ask for information in editable format.</p> <p>5-- System successfully Blocked Ancient Relics Organization</p>
Extension and Alternatives	<p>(a) Invalid informations</p> <p>(b) Re-enter user information</p>	

Table 2.6(add Encyclopedia)

Use Case Section	Comments	
Use Case Name	add Encyclopedia	
Scope	Pak Ancient Diggers	
Primary Actor	Admin, System	
Level	User goal	
Pre-condition	Actor Should be authorised person	
Post condition	Actor add aptitudesuccessfully	
	Actor Action 1-- Actor request for add Encyclopedia 3-- Actor provides the required information.	System Action 2-- System as for informations 4-- System successfully added Encyclopedia
Extension and Alternatives	(a) Invalid informations (b) Re-enter user information	

Table 2.7 (Delete Tourist)

Use Case Section	Comments	
Use Case Name	Delete Tourist	
Scope	Pak Ancient Diggers	
Primary Actor	Admin, System	
Level	User goal	
Pre-condition	Actor should be authorized to Delete Tourist	
Post condition	Actor successfully Delete Tourist	
	<p>Actor Action</p> <p>1-- admin request to Delete Tourist</p> <p>3-- Admin update the asked information</p> <p>4-- Admin confirm the Delete Tourist</p>	<p>System Action</p> <p>2-- System ask for information in editable format.</p> <p>5-- System successfully Delete Tourist</p>
Extension and Alternatives	<p>4.(a) Invalid informations</p> <p>4.(b) Re-enter user information</p>	

Table 2.8 (Block Tourist)

Use Case Section	Comments	
Use Case Name	Block Tourist	
Scope	Pak Ancient Diggers	
Primary Actor	Admin, System	
Level	User goal	
Pre-condition	Actor Should be authorised person	
Post condition	Actor successfully Block Tourist	
	<p>Actor Action</p> <p>1-- admin request to Block Tourist</p> <p>3-- Admin update the asked information</p> <p>4-- Admin confirm the Block</p>	<p>System Action</p> <p>2-- System ask for information in editable format.</p> <p>5-- System successfully Blocked Tourist</p>
Extension and Alternatives	<p>(a) Invalid informations</p> <p>(b) Re-enter user information</p>	

Table 2.9 (Block People)

Use Case Section	Comments	
Use Case Name	Block People	
Scope	Pak Ancient Diggers	
Primary Actor	Admin, System	
Level	User goal	
Pre-condition	Actor Should be authorised person	
Post condition	Actor successfully Block People	
	<p>Actor Action</p> <p>1-- admin request to Block People</p> <p>3-- Admin update the asked information</p> <p>4-- Admin confirm the Block</p>	<p>System Action</p> <p>2-- System ask for information in editable format.</p> <p>5-- System successfully Block People</p>
Extension and Alternatives	<p>(a) Invalid informations</p> <p>(b) Re-enter user information</p>	

Table 2.10 (chat)

Use Case Section	Comments	
Use Case Name	Chat	
Scope	Personality Prediction Using CV	
Primary Actor	Admin, relics organization, system	
Level	User goal	
Pre-condition	Actor should be authorized to chat	
Post condition	Actor successfully chat	
	<p>Actor Action</p> <p>1-- Actor request to chat</p> <p>3-- Actor provide the information.</p>	<p>System Action</p> <p>2-- System ask for information</p> <p>4-- System successfully chat</p>
Extension and Alternatives	<p>4.(a) Invalid informations</p> <p>4.(b) Re-enter user information</p>	

Chapter 3

SYSTEM DESIGN

System Design

System design is the process of expanding what was learned during domain analysis into a working implementation. Design implementation that will carry out the system charter and lead to reuse among many systems. In design mode, I give shape to my components of implementation. In structure analysis, these are the entities of my system. The design phase is most important part of any software development. Any error at this stage may lead to system failure at the end. Designers have developed a set of design rules that can be reapplied during the design of new applications.

Design Model (Interactive Model)

Interactive modeling emphasis on the illustration of software objects interaction via messages. The interaction diagram/model is generalization of two UML diagrams types, both can be used to express similar messages interaction.

We have included following diagrams:

- Sequence diagram
- Class diagram
- Entity Relationship Diagram

Activity Diagrams

Activity Diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as the operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

Login Activity Diagram

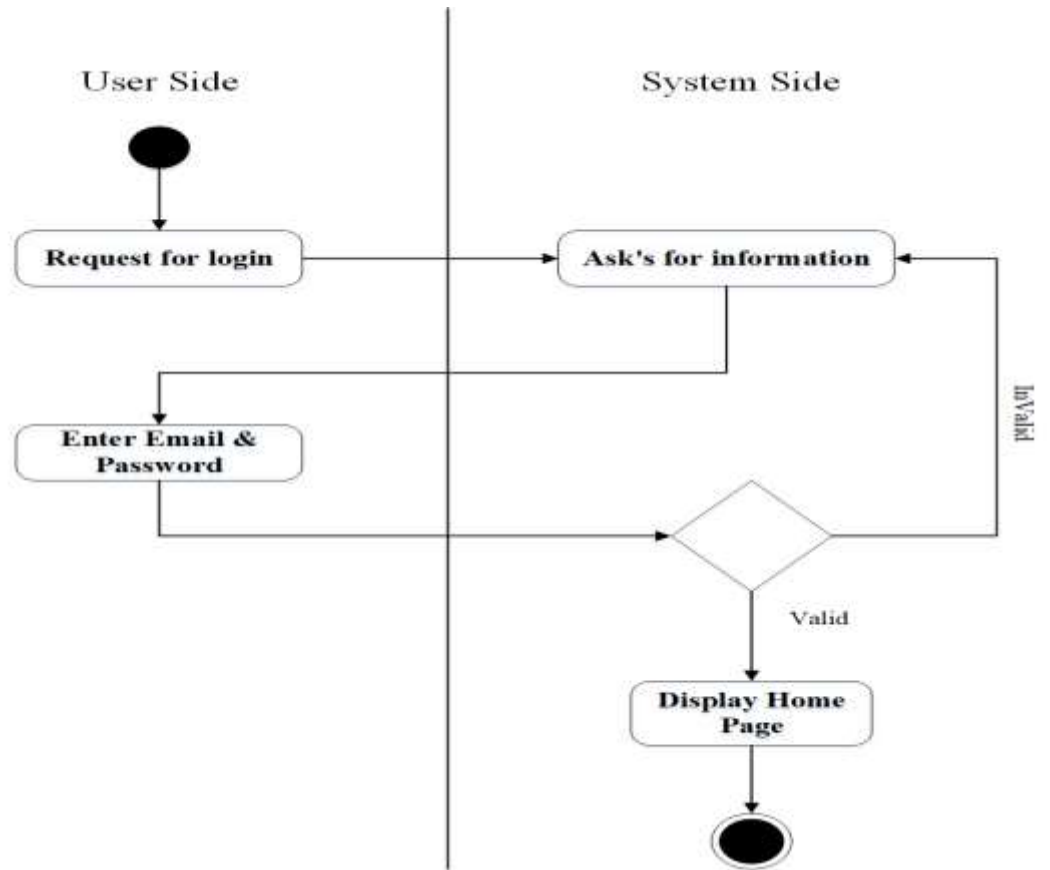


Figure 3.1 : Login activity diagram

Update profile activity diagram

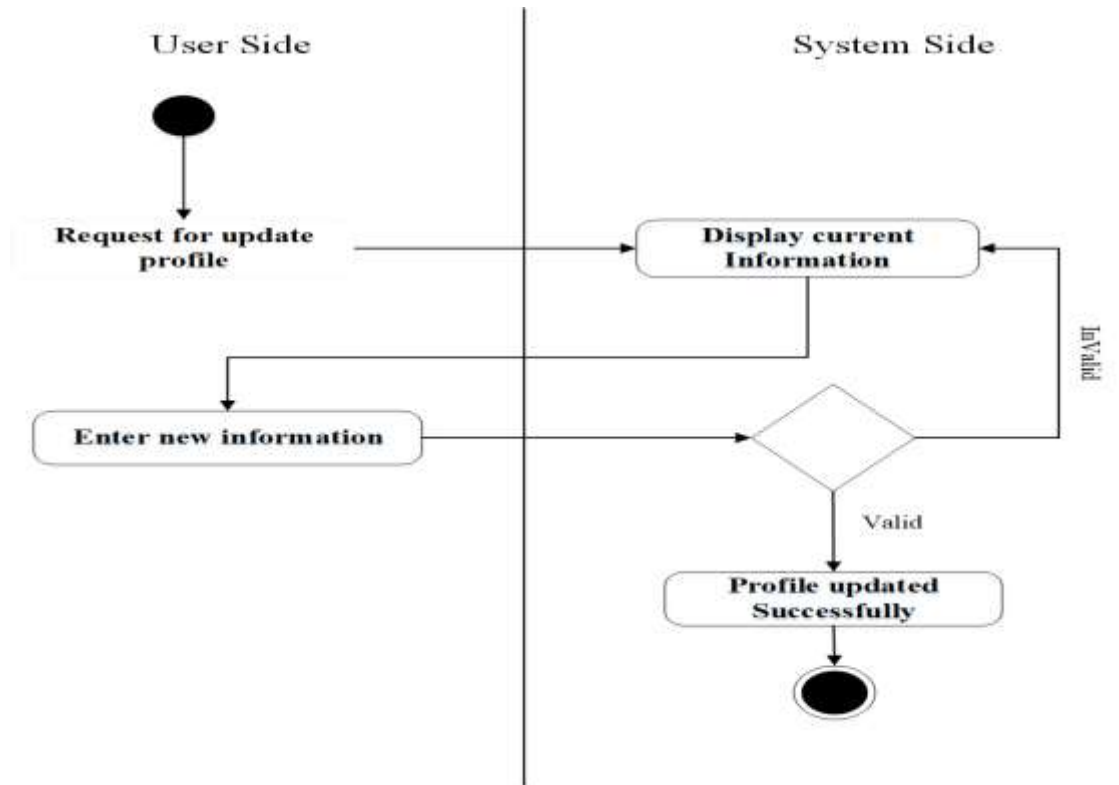


Figure 3.2: Update profile activity diagram

Generate report activity diagram

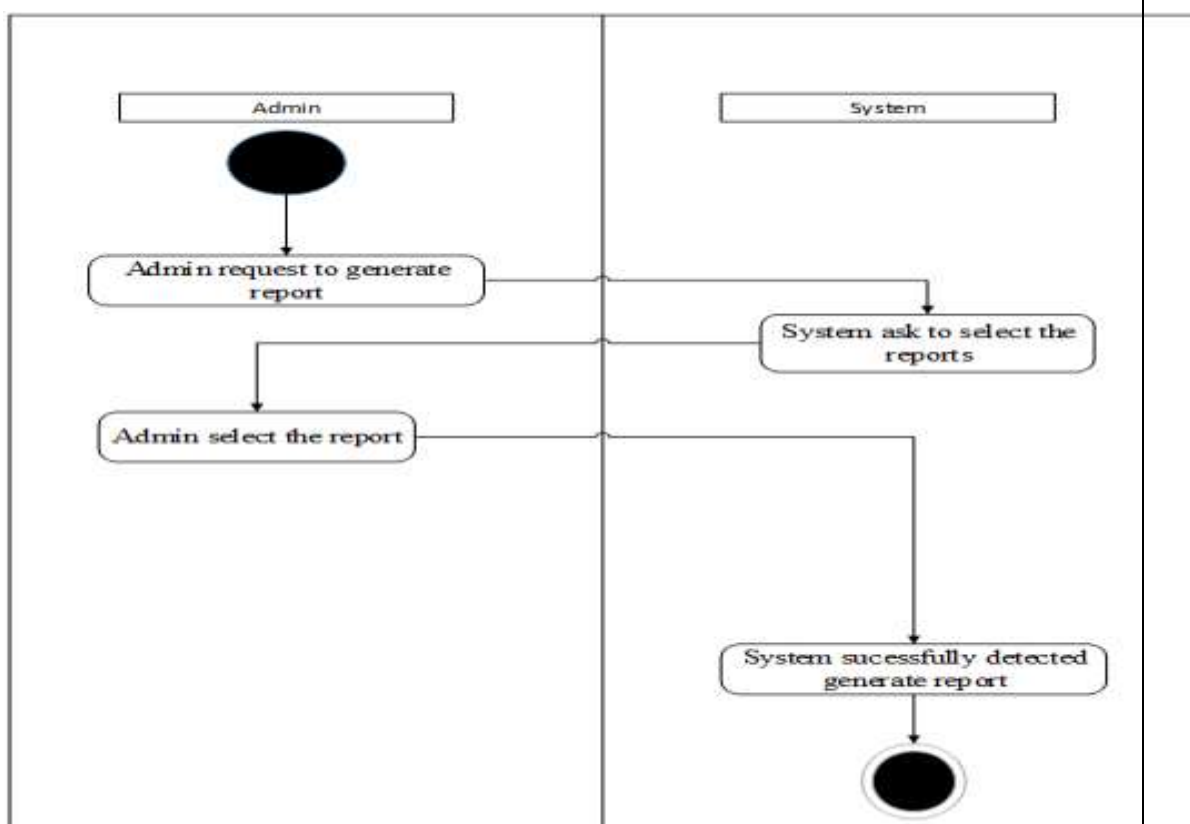


Figure 3.3: Generate report activity diagram

Feedback activity diagram

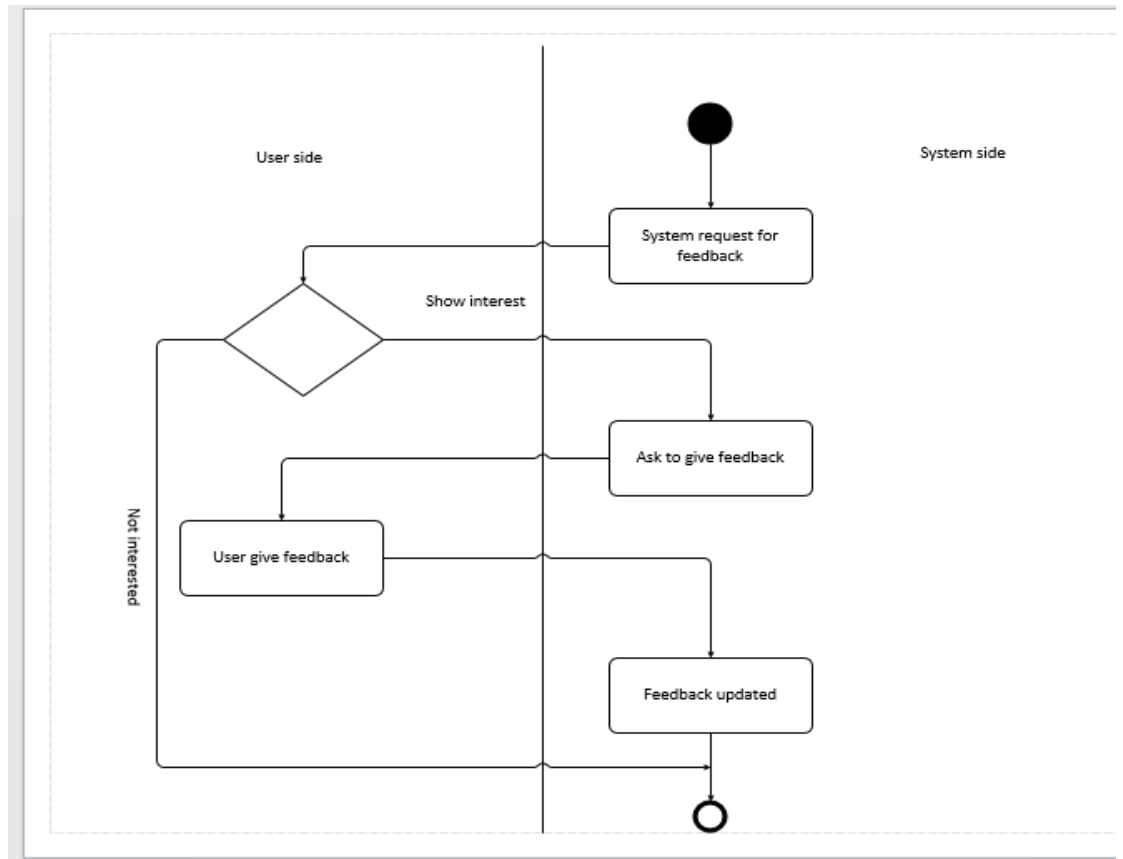


Figure 3.4: Feedback activity diagram

Search activity diagram

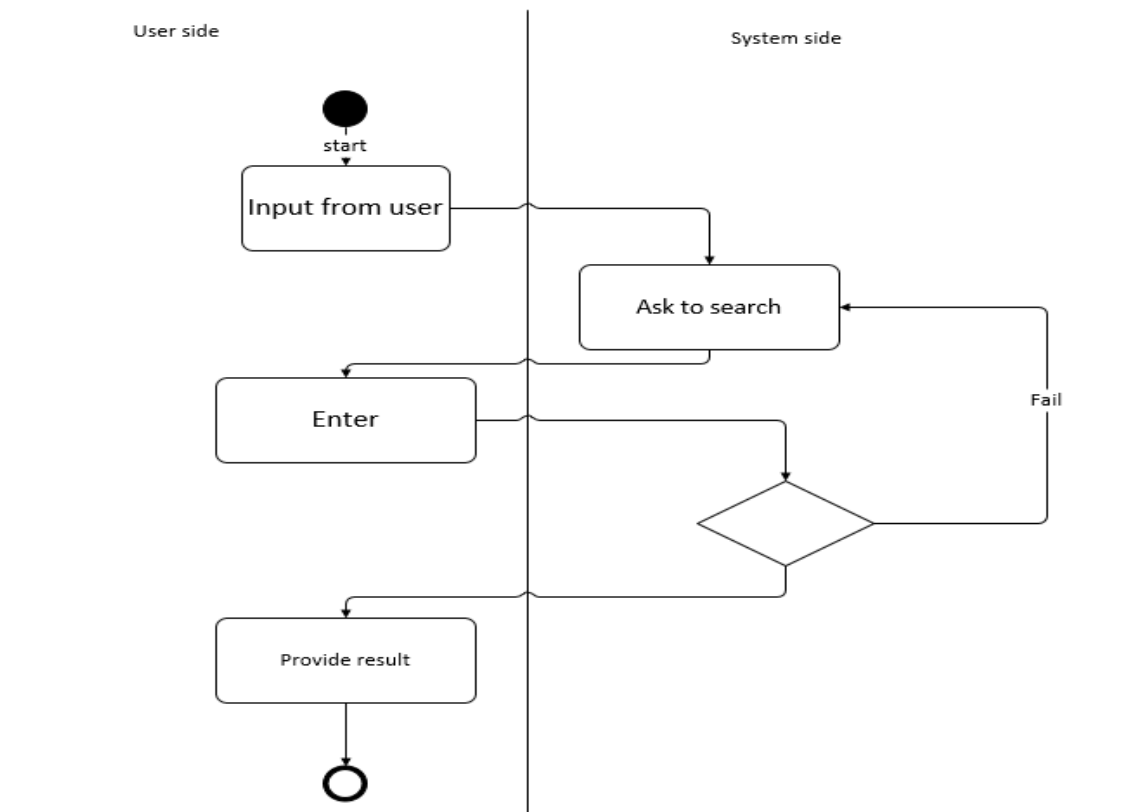


Figure 3.5: Search activity diagram

Chat activity diagram

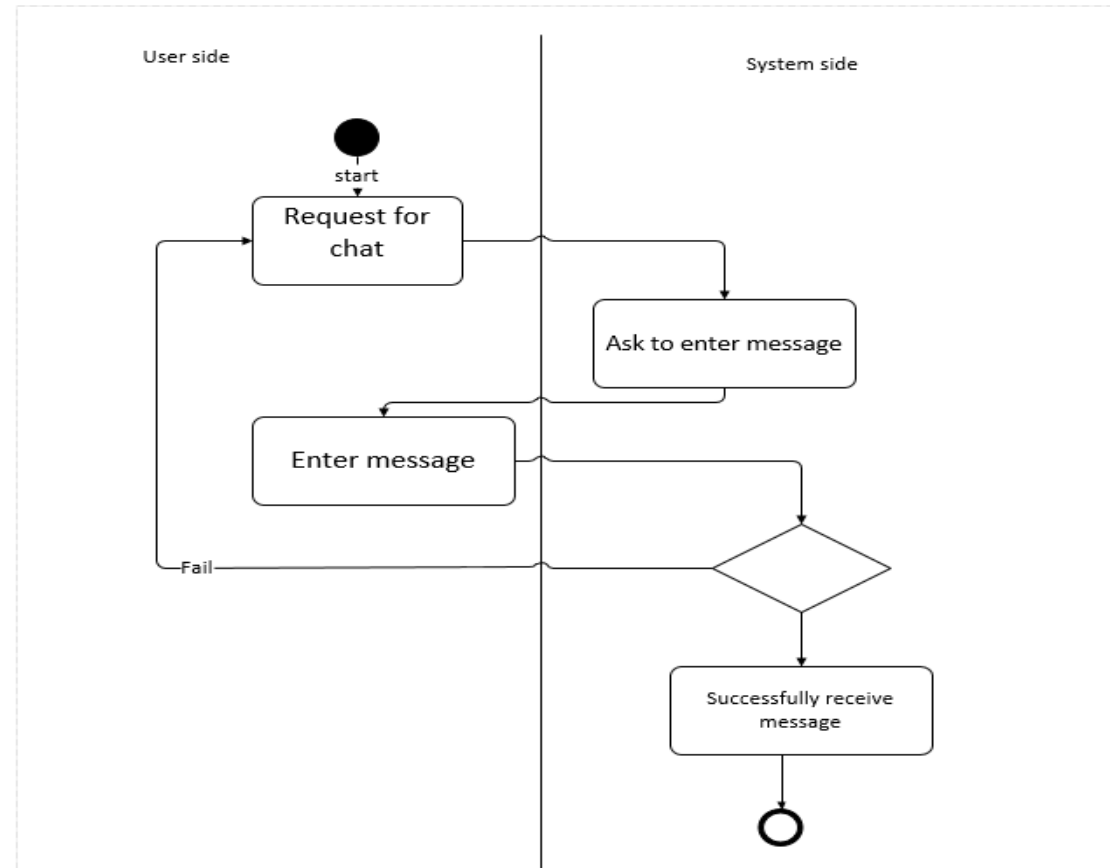


Figure 3.6: Chat activity diagram

Logout activity diagram

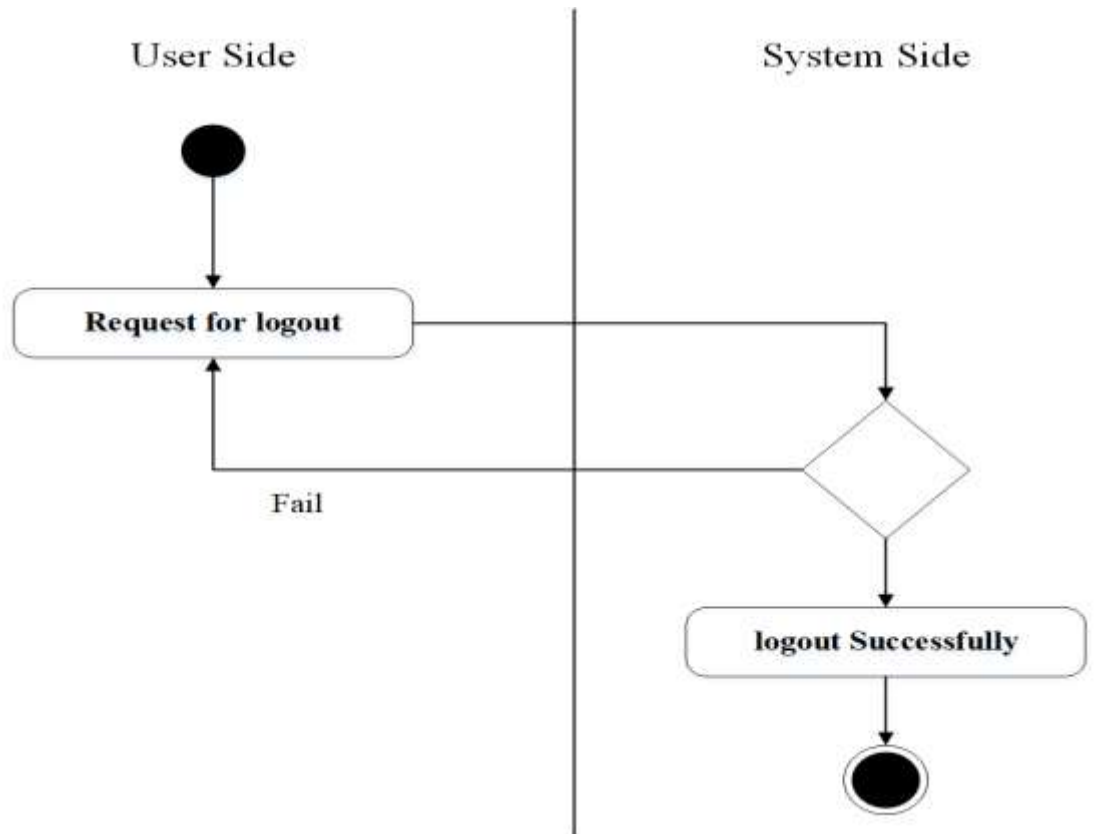


Figure 3.7: Logout activity diagram

Sequence Diagrams

UML sequence diagrams are used to show how objects interact in a given situation. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom (i.e. Lower equals later). A popular use for them is to document the dynamics in an object-oriented system. For each key-collaboration, diagrams are created that show how objects interact in various representative scenarios for that collaboration. Sequence Diagram (SD) is one of the key notations of UML and serves as a well-accepted media among software developers, stakeholders, and tool builders. The appeal can be attributed to the intuitive nature of its graphical representation and its capability to capture scenarios of how the system might be used or how entities interact by transmitting sequences of messages.

Login sequence diagram

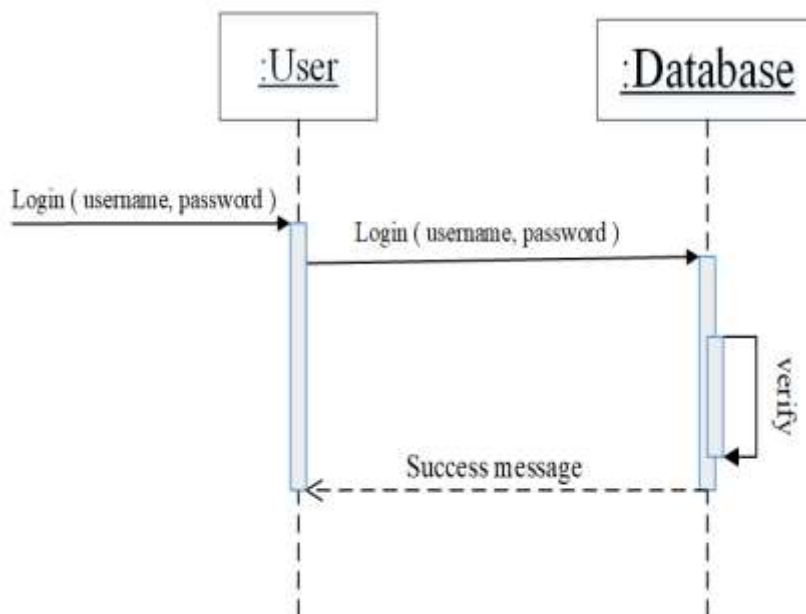


Figure 3.8: Login sequence diagram

Update profile sequence diagram

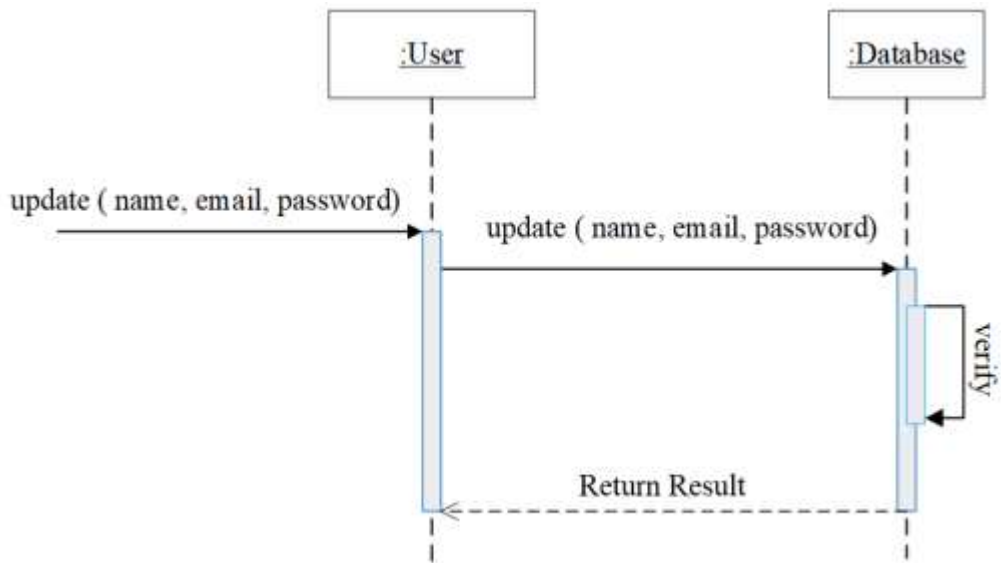


Figure 3.9: Update profile sequence diagram

Generate report sequence diagram

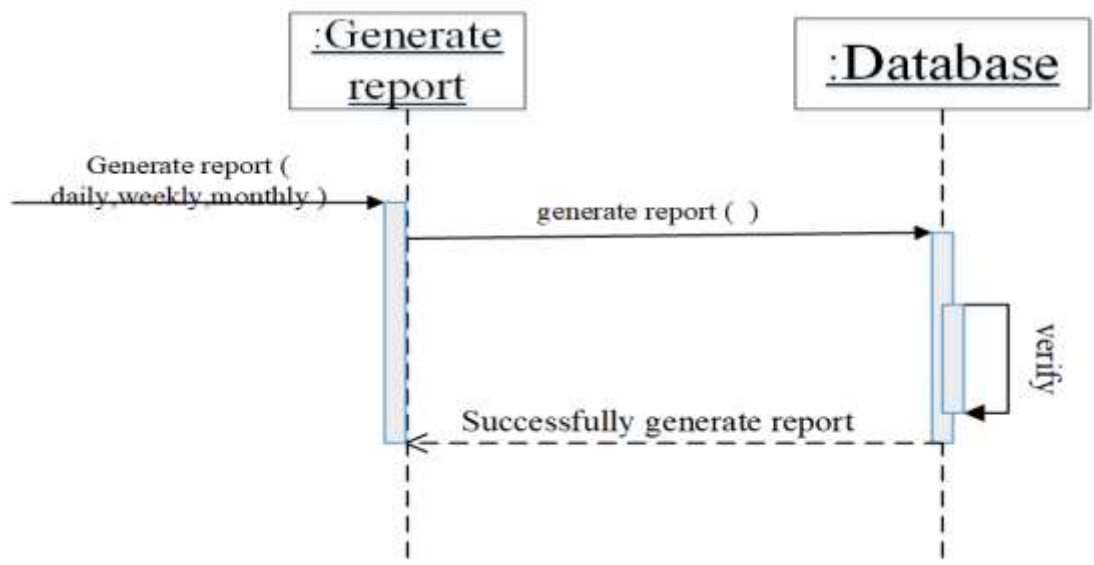


Figure 3.10: Generate report sequence diagram

Searchsequence diagram

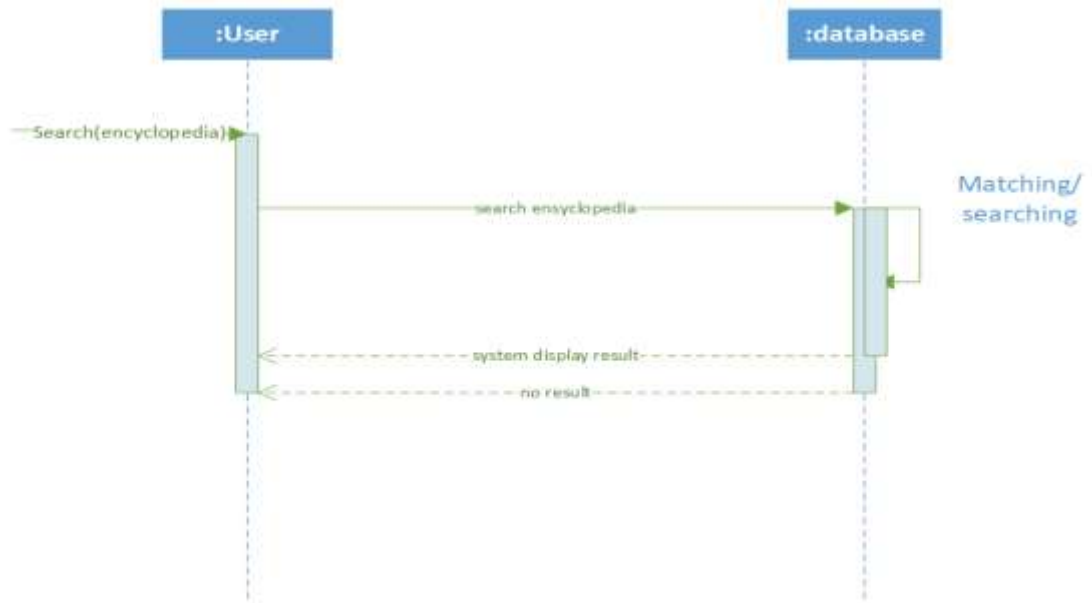


Figure 3.11: Search sequence diagram

Chat sequence diagram

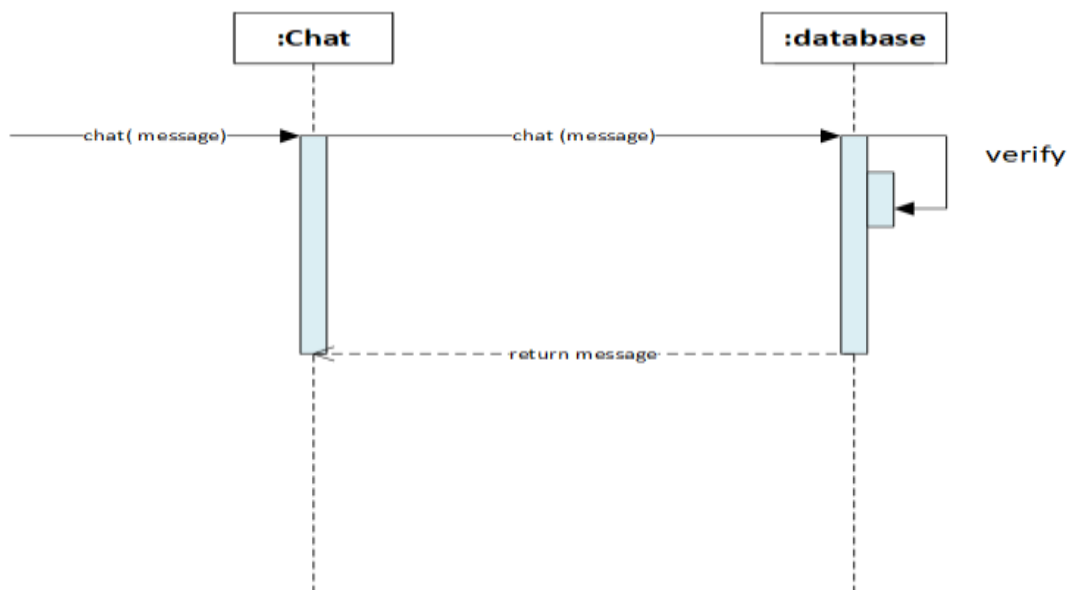


Figure 3.12: Chat sequence diagram

Payment sequence diagram

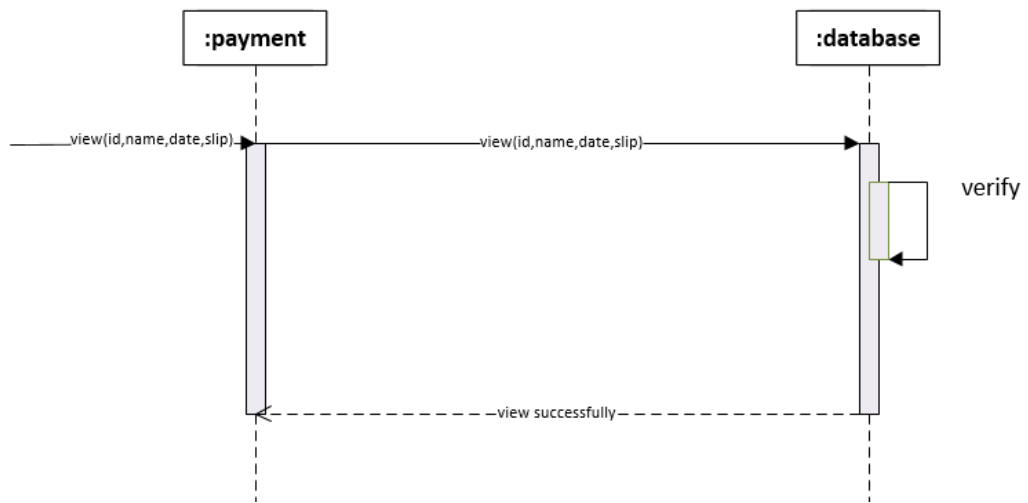


Figure 3.13: Payment sequence diagram

Logout sequence diagram

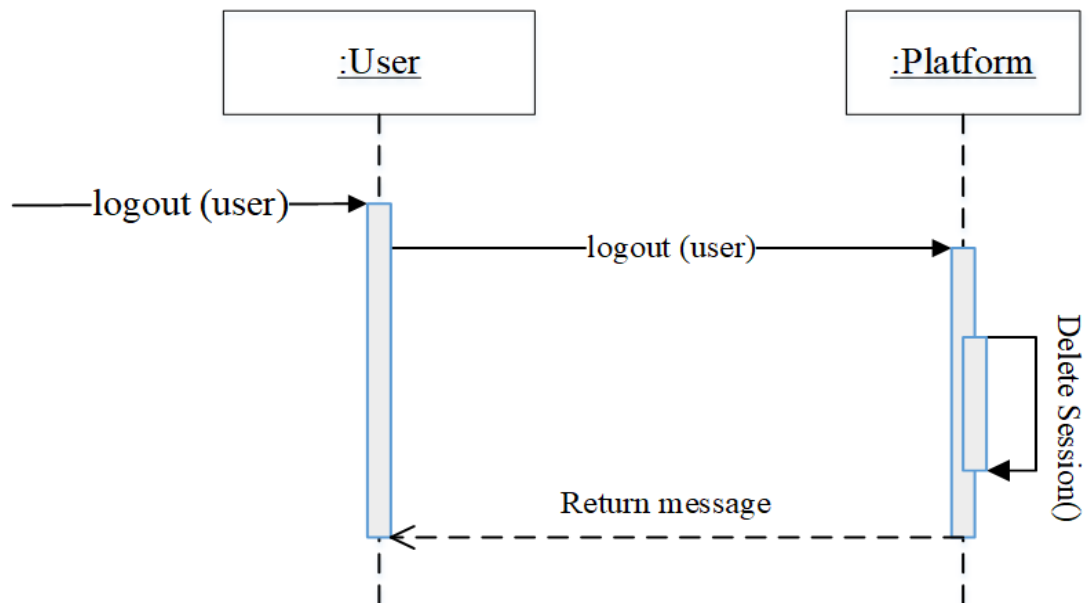


Figure 3.14: Logout sequence diagram

Class Diagram

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram.

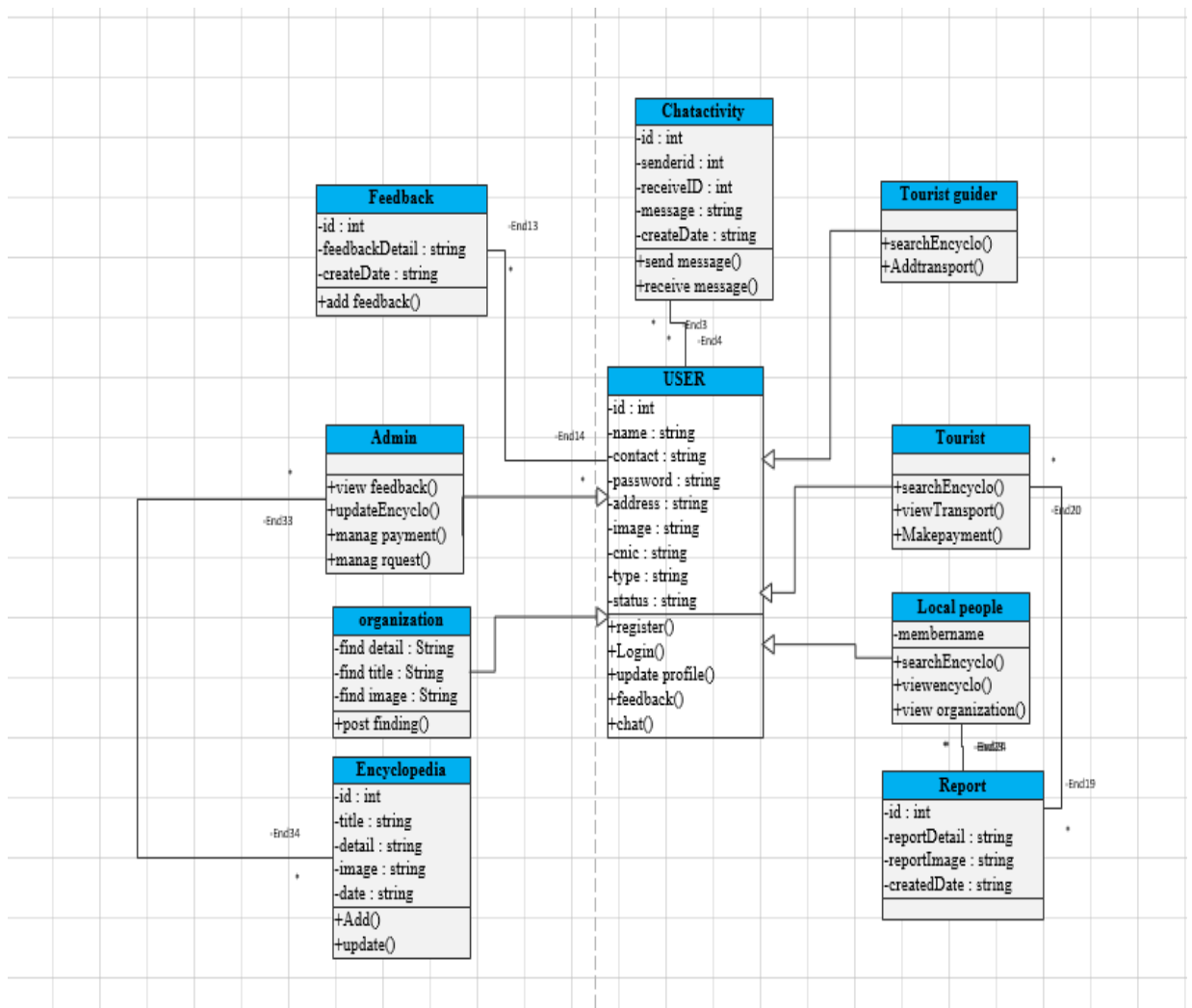


Figure 3.15 Class diagram

Entity Relation Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance, an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

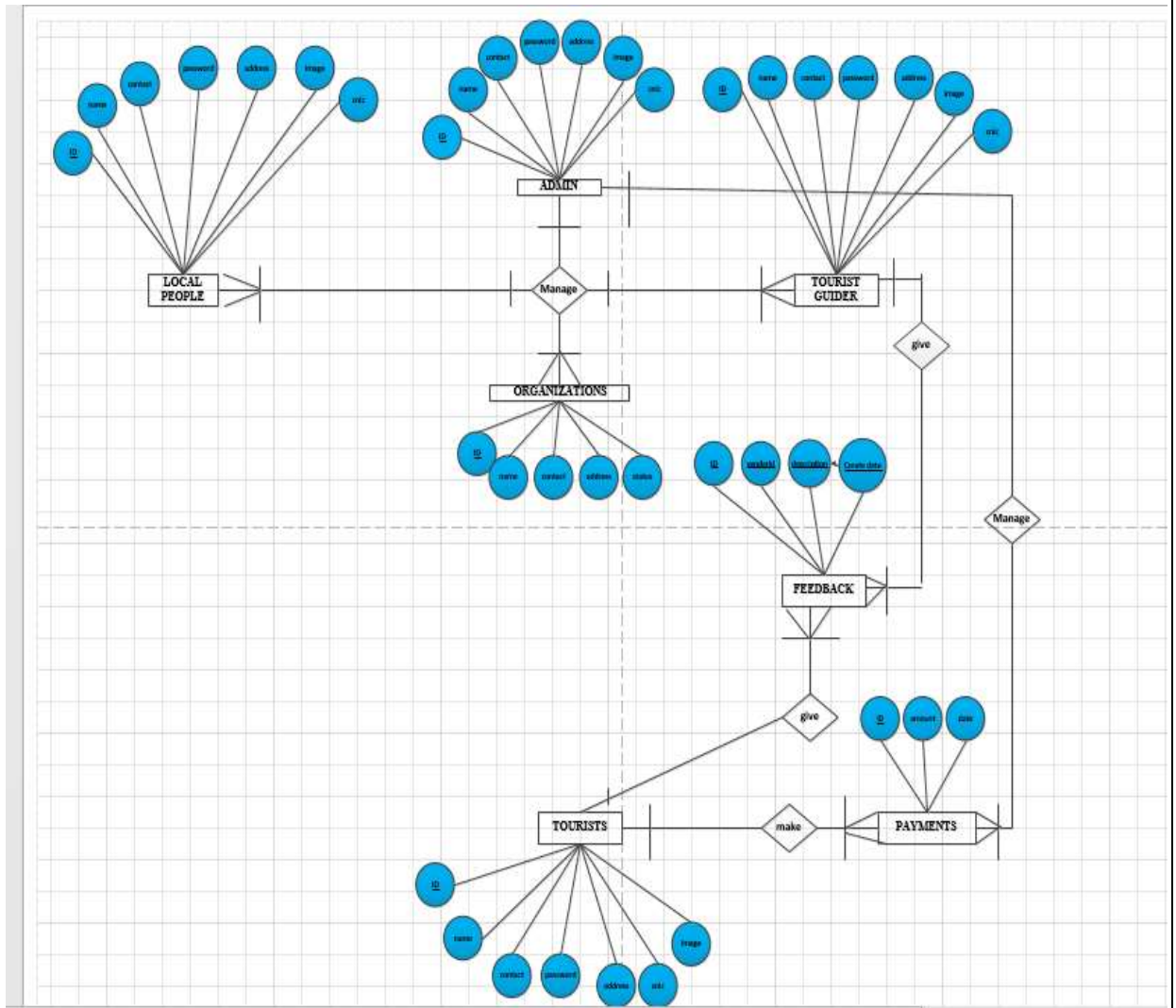


Figure 3.16 Entity Relation Diagram

CHAPTER 4
IMPLEMENTATION

IMPLEMENTATION

The implementation phase of a project starts after design. During this phase a design is converted into working software. The software is developed in such a way so that it can meet the requirements and specifications of the users. The implementation of any system is converted with the tool used in the development work and the component used to implement the system. The process of assuring that the system is operational and then allowing users to take over its operation for the use and evaluation is called implementation. Here, system means new system, which is going to be implemented. The new system needs to be implemented because the old system was manual. Therefore; there is always room for improvement of the system.

System implementation involves following key features:

- Operating system selection
- Tool selection
- Language selection
- Commenting code
- Naming conventions

Register

```
package com.fyp.ancientdiggers.localpeople;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.SessionManager;
import com.fyp.ancientdiggers.Utility;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.util.HashMap;
import java.util.Map;
import androidx.appcompat.app.AppCompatActivity;
public class RegisterUser extends AppCompatActivity {
    Button btnregister;
    private static final int SELECT_PICTURE = 1;
    private static final int CAMERA_REQUEST = 1888;
    int flaging;
    ImageView imageView;
    String url =
    "https://upwardsol.com/AccientTourGuied/Register.php";
    EditText edtname, edtpassword,
    edtPhone_no,address,edtcnic;
    SessionManager sessionManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registeruser);
        sessionManager=new
    SessionManager(RegisterUser.this);
```

```

        edtname = findViewById(R.id.edtname);
        edtpassword = findViewById(R.id.edtpassword);
        address = findViewById(R.id.edtaddress);
        edtPhone_no = findViewById(R.id.edtcontact);
        edtcnic = findViewById(R.id.edtcnic);
        imageView= findViewById(R.id.imaguser);
        /* btnregister.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                });*/
    }
    public void createaccountuser(View view){

        StringRequest request = new
StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Toast.makeText(RegisterUser.this, "" +
response, Toast.LENGTH_SHORT).show();
                try {
                    JSONObject data = new
JSONObject(response);
                    String success =
data.getString("success");
                    if
(success.toString().equalsIgnoreCase("1")) {
                        Toast.makeText(RegisterUser.this,
"Signup Success" + success, Toast.LENGTH_SHORT).show();
                        Intent intent = new
Intent(RegisterUser.this, RegisterUser.class);
                        startActivity(intent);
                    } else if
(success.toString().equalsIgnoreCase("0")){
                        Toast.makeText(RegisterUser.this,
"fill all" + success, Toast.LENGTH_SHORT).show();
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError
error) {

                Toast.makeText(RegisterUser.this, "volly
error " + error, Toast.LENGTH_SHORT).show();
            }
        }
    }

```

```

    }) {
        @Override
        protected Map<String, String> getParams()
throws AuthFailureError {

            Map<String, String> parameter = new
HashMap<String, String>();

parameter.put("type",sessionManager.getType() );
            parameter.put("name",
edtname.getText().toString());
            parameter.put("contact",
edtPhone_no.getText().toString());
            parameter.put("password",
edtpassword.getText().toString());
            parameter.put("address",
address.getText().toString());
            parameter.put("cnic",
edtcnic.getText().toString());
            parameter.put("image",
Utility.bitmapToString(((BitmapDrawable)
imageView.getDrawable()).getBitmap()));
            parameter.put("status", 0+"");
            // parameter.put("user_id",a);
            return parameter;

//                return super.getParams();
        }
    };
    RequestQueue queue =
Volley.newRequestQueue(RegisterUser.this);
    queue.add(request);

}
    public void fromgallery(View view) {
        Intent photoPickerIntent = new
Intent(Intent.ACTION_PICK);
        photoPickerIntent.setType("image/*");
        startActivityForResult(photoPickerIntent,
SELECT_PICTURE); // method 1, get from gallery
        flaging = 0;
    }
    protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode,
data);
        if (flaging == 1) {

```



```

        if (requestCode == CAMERA_REQUEST &&
resultCode == Activity.RESULT_OK) {
            Bitmap photo = (Bitmap)
data.getExtras().get("data");

            imageView.setImageBitmap(photo);
        }
    } // end of if

    else {
        if (resultCode == RESULT_OK) {
            try {
                final Uri imageUri = data.getData();
                final InputStream imageStream =
getContentResolver().openInputStream(imageUri);
                final Bitmap selectedImage =
BitmapFactory.decodeStream(imageStream);

imageView.setImageBitmap(selectedImage);

            } catch (FileNotFoundException e) {
                e.printStackTrace();
                Toast.makeText(RegisterUser.this,
"Something went wrong", Toast.LENGTH_LONG).show();
            }

            } else {
                Toast.makeText(RegisterUser.this, "You
haven't picked Image", Toast.LENGTH_LONG).show();
            }
        } // end of else
    }
}
}

```

Adapter manage all

```

package com.fyp.ancientdiggers.Admin;
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;

```

```

import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.Utility;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
public class AdapterManageAll extends
ArrayAdapter<DataClassManageAll> {
    Context context;
    ArrayList<DataClassManageAll> allRegistrations;

    public AdapterManageAll(Context context,
ArrayList<DataClassManageAll> allRegistrations ) {
        super(context,
R.layout.custom_manage_all,allRegistrations);
        this.allRegistrations=allRegistrations;
        this.context=context;
    }
    @Override
    public View getView(final int position, View
convertView, ViewGroup parent) {
        LayoutInflater
inflater=LayoutInflater.from(context);

convertView=inflater.inflate(R.layout.custom_manage_all,n
ull);
        TextView txtcontact = (TextView)
convertView.findViewById(R.id.txtcontact);
        TextView txtidentity = (TextView)
convertView.findViewById(R.id.txtidentity);
        TextView txtname = (TextView)
convertView.findViewById(R.id.txtname);
        TextView txtcategory = (TextView)
convertView.findViewById(R.id.txtcategory);
        TextView txtstatus = (TextView)
convertView.findViewById(R.id.txtstatus);
        TextView txtaprove = (TextView)
convertView.findViewById(R.id.txtaprove);
        TextView txtreject = (TextView)
convertView.findViewById(R.id.txtreject);
        TextView txtblock = (TextView)
convertView.findViewById(R.id.txtblock);
        TextView txtdelete = (TextView)
convertView.findViewById(R.id.txtdelete);
        ImageView profile_image = (ImageView)
convertView.findViewById(R.id.profile_image);

```

```

txtcontact.setText(allRegistrations.get(position).getCont
act());

txtidentity.setText(allRegistrations.get(position).getCni
c());

txtcategory.setText(allRegistrations.get(position).getTyp
e());

txtname.setText(allRegistrations.get(position).getName()
);

ofile_image.setImageBitmap(Utility.stringToBitmap(allRegi
strations.get(position).getImage()));
    int status=
Integer.parseInt(allRegistrations.get(position).getStatus
());
    if(status==0){
        txtstatus.setText("Request Pending");
    }
    if(status==1){
        txtstatus.setText("Request Accepted");
    }
    if(status==2){
        txtstatus.setText("Request Reject");
    }
    if(status==3){

        txtstatus.setText("Request Block");

    }
    txtaprove.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {

UpdateStatus("1",allRegistrations.get(position).getId());
            Toast.makeText(context, "1",
Toast.LENGTH_SHORT).show();
            Toast.makeText(context,
""+allRegistrations.get(position).getId(),
Toast.LENGTH_SHORT).show();
        }
    });
    txtreject.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {

UpdateStatus("2",allRegistrations.get(position).getId());

```

```

        }
    });

    txtblock.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {

UpdateStatus("3",allRegistrations.get(position).getId());
    }
    });
    txtdelete.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {

Delete(allRegistrations.get(position).getId());
    }
    });
    return convertView;
}

public void UpdateStatus(final String status,final
String id) {
    StringRequest request = new
StringRequest(Request.Method.POST, "
https://upwardsol.com/AccientTourGuied/UpdateStatus.php",
new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        try {
            JSONObject data=new
JSONObject(response);
            String
success=data.getString("success");

            if(success.equalsIgnoreCase("1")){

                Toast.makeText(context, "Record
Hasbeen Updated", Toast.LENGTH_SHORT).show();

            }
            else
if(success.equalsIgnoreCase("0")){

                Toast.makeText(context, "fill
all"+success, Toast.LENGTH_SHORT).show();
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}
}

```

```

        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError
error) {

            Toast.makeText(context, "volly error
"+error, Toast.LENGTH_SHORT).show();
        }
    })
    {
        @Override
        protected Map<String, String> getParams()
throws AuthFailureError {

            Map<String, String> parameter = new
HashMap<String, String>();
            parameter.put("status", status+"");
            parameter.put("id", id+"");
            return parameter;
        }
    }
    ;
    RequestQueue queue =
Volley.newRequestQueue(context);
    queue.add(request);

}

```

```

    public void Delete(final String id) {
        StringRequest request = new
StringRequest(Request.Method.POST, "
https://upwardsol.com/AccientTourGuied/Delete.php", new
Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Toast.makeText(context, ""+response,
Toast.LENGTH_SHORT).show();
                try {
                    JSONObject data=new
JSONObject(response);
                    String
success=data.getString("success");
                    Toast.makeText(context, ""+success,
Toast.LENGTH_SHORT).show();

```

```

        if (success.equalsIgnoreCase("1")) {
            Toast.makeText(context, "Record
Hasbeen Deleted", Toast.LENGTH_SHORT).show();
        }

        else
if(success.equalsIgnoreCase("0")){
            Toast.makeText(context, "fill
all"+success, Toast.LENGTH_SHORT).show();
        }

        } catch (JSONException e) {
            e.printStackTrace();
        }

    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError
error) {

        Toast.makeText(context, "volley error
"+error, Toast.LENGTH_SHORT).show();
    }
})
{
    @Override
    protected Map<String, String> getParams()
throws AuthFailureError {

        Map<String, String> parameter = new
HashMap<String, String>();
        parameter.put("id", id+"");
        return parameter;

    }

}

;

RequestQueue queue =
Volley.newRequestQueue(context);
queue.add(request);
}
}

```

Add account

```
package com.fyp.ancientdiggers.organization;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.SessionManager;
import com.fyp.ancientdiggers.Utility;
import com.fyp.ancientdiggers.localpeople.RegisterUser;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;
import java.util.Map;
public class AddAccountActivity extends AppCompatActivity
{
    SessionManager sessionManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_account);
        sessionManager=new
    SessionManager (AddAccountActivity.this);
    }
    public void adresdetail(View v){

        addaccountuser();

    }
    public void addaccountuser(){

        EditText
edtaccountcontact,edtaccountaddress,edtaccountemail,edtac
countwhatsapnum;
        edtaccountcontact =
findViewById(R.id.edtaccountcontact);
        edtaccountaddress =
findViewById(R.id.edtaccountaddress);
        edtaccountemail =
findViewById(R.id.edtaccountemail);
```

```

        edtaccountwhatsapnum =
findViewById(R.id.edtaccountwhatsapnum);

        StringRequest request = new
StringRequest(Request.Method.POST,
"https://upwardsol.com/AccientTourGuied/AddorganizationCo
ntact.php", new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        Toast.makeText(AddAccountActivity.this,
"" + response, Toast.LENGTH_SHORT).show();
        try {
            JSONObject data = new
JSONObject(response);
            String success =
data.getString("success");

            if
(success.toString().equalsIgnoreCase("1")) {
Toast.makeText(AddAccountActivity.this, "Signup Success"
+ success, Toast.LENGTH_SHORT).show();
                } else if
(success.toString().equalsIgnoreCase("0")){
Toast.makeText(AddAccountActivity.this, "fill all" +
success, Toast.LENGTH_SHORT).show();
                }

            } catch (JSONException e) {
                e.printStackTrace();
            }

        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError
error) {

            Toast.makeText(AddAccountActivity.this,
"volly error " + error, Toast.LENGTH_SHORT).show();
        }
    }) {
        @Override
        protected Map<String, String> getParams()
throws AuthFailureError {

```



```

        Map<String, String> parameter = new
HashMap<String, String>();
        parameter.put("address",
edtaccountaddress.getText().toString());
        parameter.put("email",
edtaccountemail.getText().toString());
        parameter.put("whatsApp",
edtaccountwhatsapnum.getText().toString());
        parameter.put("contact",
edtaccountcontact.getText().toString());
        parameter.put("orgid",
sessionManager.getId()+"");

        // parameter.put("user_id",a);
return parameter;

// return super.getParams();
    }
};
RequestQueue queue =
Volley.newRequestQueue(AddAccountActivity.this);
queue.add(request);

}
}

```

Add encyclopedia

```

package com.fyp.ancientdiggers.Admin;
import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;

```

```

import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.MainActivity;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.SessionManager;
import com.fyp.ancientdiggers.Utility;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.util.HashMap;
import java.util.Map;
import androidx.appcompat.app.AppCompatActivity;
public class AddEncyclopedia extends AppCompatActivity {

    Button btnregister;
    private static final int SELECT_PICTURE = 1;
    private static final int CAMERA_REQUEST = 1888;
    int flaging;
    ImageView imagevieplace;
    EditText edtEncyclopediatitle,edtEncyclopediadetial;
    String url = "
https://upwardsol.com/AccientTourGuied/AddEncyclopedia.ph
p";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_add_encycloedia);
        edtEncyclopediatitle =
findViewById(R.id.edtEncyclopediatitle);
        edtEncyclopediadetial =
findViewById(R.id.edtEncyclopediadetial);
        imagevieplace= findViewById(R.id.imageplace);
    }
    public void addEncyclopedia(View view){

        StringRequest request = new
StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Toast.makeText(AddEncyclopedia.this, "" +
response, Toast.LENGTH_SHORT).show();
                try {
                    JSONObject data = new
JSONObject(response);
                    String success =
data.getString("success");

                    if
(success.toString().equalsIgnoreCase("1")) {

```

```

Toast.makeText(AddEncyclopedia.this, "Sucessfully Add
Encyclopedia" + success, Toast.LENGTH_SHORT).show();

        } else if
(success.toString().equalsIgnoreCase("0")){

Toast.makeText(AddEncyclopedia.this, "fill all" +
success, Toast.LENGTH_SHORT).show();
        }

        } catch (JSONException e) {
            e.printStackTrace();
        }

    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError
error) {

        Toast.makeText(AddEncyclopedia.this,
"volly error " + error, Toast.LENGTH_SHORT).show();
    }
}) {
    @Override
    protected Map<String, String> getParams()
throws AuthFailureError {
        Map<String, String> parameter = new
HashMap<String, String>();
        parameter.put("encyclopediatitle",
edtEncyclopediatitle.getText().toString());
        parameter.put("encyclopediadetail",
edtEncyclopediadetial.getText().toString());
        parameter.put("encyclopediainage",
Utility.bitmapToString(((BitmapDrawable)
imagevieplace.getDrawable()).getBitmap()));
        // parameter.put("user_id", a);
        return parameter;

//            return super.getParams();
    }
};
RequestQueue queue =
Volley.newRequestQueue(AddEncyclopedia.this);
queue.add(request);

}
@Override
public boolean onCreateOptionsMenu(Menu menu) {

```

```

        // Inflate the menu; this adds items to the
        action bar if it is present.
        getMenuInflater().inflate(R.menu.adminmenu,
        menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch(item.getItemId()) {
            case R.id.action_logout:
                Toast.makeText(AddEncyclopedia.this,
                "logout", Toast.LENGTH_SHORT).show();
                SessionManager obj = new
                SessionManager(AddEncyclopedia.this);
                obj.setLogin(false);
                Intent i=new Intent(AddEncyclopedia.this,
                MainActivity.class);

                startActivity(i);
                return(true);

            case R.id.action_updateprofile:

                /* Intent i2=new
                Intent(AddEncyclopedia.this,UpdateProfile.class);
                startActivity(i2);*/

                return(true);

        }
        return(super.onOptionsItemSelected(item));
    }
    public void fromgallery(View view) {
        Intent photoPickerIntent = new
        Intent(Intent.ACTION_PICK);
        photoPickerIntent.setType("image/*");
        startActivityForResult(photoPickerIntent,
        SELECT_PICTURE); // method 1, get from gallery
        flaging = 0;
    }
    protected void onActivityResult(int requestCode, int
    resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode,
        data);
        if (flaging == 1) {
            if (requestCode == CAMERA_REQUEST &&
            resultCode == Activity.RESULT_OK) {

```

```

        Bitmap photo = (Bitmap)
data.getExtras().get("data");

        imagevieplace.setImageBitmap(photo);
    }
} // end of if

else {
    if (resultCode == RESULT_OK) {
        try {
            final Uri imageUri = data.getData();
            final InputStream imageStream =
getContentResolver().openInputStream(imageUri);
            final Bitmap selectedImage =
BitmapFactory.decodeStream(imageStream);

imagevieplace.setImageBitmap(selectedImage);

            } catch (FileNotFoundException e) {
                e.printStackTrace();
                Toast.makeText(AddEncyclopedia.this,
"Something went wrong", Toast.LENGTH_LONG).show();
            }

            } else {
                Toast.makeText(AddEncyclopedia.this, "You
haven't picked Image", Toast.LENGTH_LONG).show();
            }
        } // end of else
    }
}

```

Add payment

```

package com.fyp.ancientdiggers.touriest;
import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;

```

```

import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.Utility;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.util.HashMap;
import java.util.Map;
import androidx.appcompat.app.AppCompatActivity;
public class AddPaymentActivity extends AppCompatActivity
{
    private static final int SELECT_PICTURE = 1;
    private static final int CAMERA_REQUEST = 1888;
    int flaging,sid,rid;
    ImageView imageView;
    EditText edtamount,edtsenderid,edtreciverid;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_payment);
        Intent intent=getIntent();
        sid=intent.getIntExtra("sid",0);
        rid=intent.getIntExtra("rid",0);
        imageView = findViewById(R.id.imagepay);
        edtamount= findViewById(R.id.edtamount);
        edtsenderid= findViewById(R.id.edtsenderid);
        edtreciverid= findViewById(R.id.edtreciverid);

    }

    public void fromgallery(View view) {
        Intent photoPickerIntent = new
Intent(Intent.ACTION_PICK);
        photoPickerIntent.setType("image/*");
        startActivityForResult(photoPickerIntent,
SELECT_PICTURE); // method 1, get from gallery
        flaging = 0;
    }

    public void fromcamera(View view) {
        Intent cameraIntent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(cameraIntent,
CAMERA_REQUEST); // get from camera
        flaging = 1;
    }
}

```

```

        /*public void showdata(View view)
        {
            Intent intent = new
Intent(Register.this,Votter.class);
            startActivity(intent);
        }
*/
        protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
            super.onActivityResult(requestCode, resultCode,
data);
            if (flaging == 1) {
                if (requestCode == CAMERA_REQUEST &&
resultCode == Activity.RESULT_OK) {
                    Bitmap photo = (Bitmap)
data.getExtras().get("data");

                    imageView.setImageBitmap(photo);
                }
            } // end of if

            else {
                if (resultCode == RESULT_OK) {
                    try {
                        final Uri imageUri = data.getData();
                        final InputStream imageStream =
getContentResolver().openInputStream(imageUri);
                        final Bitmap selectedImage =
BitmapFactory.decodeStream(imageStream);

imageView.setImageBitmap(selectedImage);

                    } catch (FileNotFoundException e) {
                        e.printStackTrace();

Toast.makeText(AddPaymentActivity.this, "Something went
wrong", Toast.LENGTH_LONG).show();
                    }

                } else {
                    Toast.makeText(AddPaymentActivity.this,
"You haven't picked Image", Toast.LENGTH_LONG).show();
                }
            } // end of else
        }

        public void sendPay(View view){

```

```

        StringRequest request = new
StringRequest(Request.Method.POST,
"https://upwardsol.com/AccientTourGuied/addpayment.php",
new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        Toast.makeText(AddPaymentActivity.this,
"" + response, Toast.LENGTH_SHORT).show();
        try {
            JSONObject data = new
JSONObject(response);
            String success =
data.getString("success");

            if
(success.toString().equalsIgnoreCase("1")) {

Toast.makeText(AddPaymentActivity.this, "Signup Success"
+ success, Toast.LENGTH_SHORT).show();
                /* Intent intent = new
Intent(AddPaymentActivity.this, LoginActivity.class);
                startActivity(intent);*/

            } else if
(success.toString().equalsIgnoreCase("0")){

Toast.makeText(AddPaymentActivity.this, "fill all" +
success, Toast.LENGTH_SHORT).show();
            }

        } catch (JSONException e) {
            e.printStackTrace();
        }

    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError
error) {

        Toast.makeText(AddPaymentActivity.this,
"volly error " + error, Toast.LENGTH_SHORT).show();
    }
}) {
    @Override
    protected Map<String, String> getParams()
throws AuthFailureError {

        Map<String, String> parameter = new
HashMap<String, String>();

```



```

        parameter.put("senderid", sid+"");
        parameter.put("recieverid", rid+"");
        parameter.put("amount",
edtamount.getText().toString());
        parameter.put("amountslip",
Utility.bitmapToString(((BitmapDrawable)
imageView.getDrawable()).getBitmap()));

        // parameter.put("user_id",a);
        return parameter;

//
        return super.getParams();
    }
};
RequestQueue queue =
Volley.newRequestQueue(AddPaymentActivity.this);
queue.add(request);

}
}

```

Add tour guide

```

package com.fyp.ancientdiggers.Admin;
import androidx.appcompat.app.AppCompatActivity;
import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.Utility;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.FileNotFoundException;
import java.io.InputStream;

```

```

import java.util.HashMap;
import java.util.Map;

public class AddTourGuiedActivity extends
AppCompatActivity {

    Button btnregister;
    private static final int SELECT_PICTURE = 1;
    private static final int CAMERA_REQUEST = 1888;
    int flaging;
    ImageView imageView;
    String url = "
https://upwardsol.com/AccientTourGuied/RegisterTourGuied.
php";
    EditText edtname, edtpassword, edtPhone_no,address;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_tour_guided);

        edtname = findViewById(R.id.edtname);
        edtpassword = findViewById(R.id.edtpassword);
        address = findViewById(R.id.edtaddress);
        edtPhone_no = findViewById(R.id.edtcontact);
        imageView= findViewById(R.id.imaguser);

        /*   btnregister.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

                });*/

    }

    public void addtourguiedtodatbase(View view){

        StringRequest request = new
StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Toast.makeText(AddTourGuiedActivity.this,
"" + response, Toast.LENGTH_SHORT).show();
                try {

```

```

        JSONObject data = new
JSONObject(response);
        String success =
data.getString("success");

        if
(success.toString().equalsIgnoreCase("1")) {

Toast.makeText(AddTourGuidedActivity.this, "Signup
Success" + success, Toast.LENGTH_SHORT).show();
                Intent intent = new
Intent(AddTourGuidedActivity.this,
AddTourGuidedActivity.class);
                startActivity(intent);

        } else if
(success.toString().equalsIgnoreCase("0")){

Toast.makeText(AddTourGuidedActivity.this, "fill all" +
success, Toast.LENGTH_SHORT).show();
        }

        } catch (JSONException e) {
            e.printStackTrace();
        }

    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError
error) {

        Toast.makeText(AddTourGuidedActivity.this,
"volly error " + error, Toast.LENGTH_SHORT).show();
    }
}) {
    @Override
    protected Map<String, String> getParams()
throws AuthFailureError {

        Map<String, String> parameter = new
HashMap<String, String>();
        parameter.put("name",
edtname.getText().toString());
        parameter.put("userpassword",
edtpassword.getText().toString());
        parameter.put("usercontact",
edtPhone_no.getText().toString());
    }
}

```

```

        parameter.put("useraddress",
address.getText().toString());
        parameter.put("userimage",
Utility.bitmapToString(((BitmapDrawable)
imageView.getDrawable()).getBitmap()));
        parameter.put("usertype", "2");

        // parameter.put("user_id",a);
return parameter;

//
return super.getParams();
    }
};
RequestQueue queue =
Volley.newRequestQueue(AddTourGuiedActivity.this);
queue.add(request);

}
public void fromgallery(View view) {
Intent photoPickerIntent = new
Intent(Intent.ACTION_PICK);
photoPickerIntent.setType("image/*");
startActivityForResult(photoPickerIntent,
SELECT_PICTURE); // method 1, get from gallery
flaging = 0;

}
protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
super.onActivityResult(requestCode, resultCode,
data);
if (flaging == 1) {
if (requestCode == CAMERA_REQUEST &&
resultCode == Activity.RESULT_OK) {
Bitmap photo = (Bitmap)
data.getExtras().get("data");

imageView.setImageBitmap(photo);
}
} // end of if

else {
if (resultCode == RESULT_OK) {
try {
final Uri imageUri = data.getData();
final InputStream imageStream =
getContentResolver().openInputStream(imageUri);

```

```

        final Bitmap selectedImage =
BitmapFactory.decodeStream(imageStream);

imageView.setImageBitmap(selectedImage);

        } catch (FileNotFoundException e) {
            e.printStackTrace();

Toast.makeText(AddTourGuiedActivity.this, "Something went
wrong", Toast.LENGTH_LONG).show();
        }

        } else {
            Toast.makeText(AddTourGuiedActivity.this,
"You haven't picked Image", Toast.LENGTH_LONG).show();
        }
    } // end of else
}

}

```

Admin category adapter

```

package com.fyp.ancientdiggers.Admin;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.Utility;
import java.util.ArrayList;
public class AdminCategoryAdapter extends
ArrayAdapter<AllCategoryDataClass> {
    ArrayList<AllCategoryDataClass> thedata;
    Context context;

    public AdminCategoryAdapter(Context context,
ArrayList<AllCategoryDataClass> thedata) {
        super(context, R.layout.custom_hospital_layout,
thedata);
        this.thedata = thedata;
        this.context = context;
    }

    @Override
    public View getView(int position, View convertView,
ViewGroup parent) {

```

```

        LayoutInflater inflater =
LayoutInflater.from(context);
        View view =
inflater.inflate(R.layout.custom_category_layout,
null);

        TextView txtcategoryname =
view.findViewById(R.id.txtcategoryname);
        TextView txtcategorycontact =
view.findViewById(R.id.txtcategorycontact);
        TextView txtcategorydetail =
view.findViewById(R.id.txtcategorydetail);
        TextView txtcategoryid =
view.findViewById(R.id.txtcategoryid);

        ImageView
imageviewhospital=view.findViewById(R.id.imageviewhospita
l);

txtcategoryname.setText(thedata.get(position).getCategory
name());
        txtcategorycontact.setText("Contact
:"+thedata.get(position).getCategorycontact());
        txtcategorydetail.setText("Detail
:"+thedata.get(position).getCategorydetail());
        /*txtcategoryid.setText("Type
:"+thedata.get(position).getCategorytype());*/

        try{

imageviewhospital.setImageBitmap(Utility.stringToBitmap(t
hedata.get(position).getCategoryimage()));
        }
        catch (Exception er)
        {

imageviewhospital.setImageResource(R.mipmap.hunza);
        }

        return view;
    }
}

```

Admin dash board

```

package com.fyp.ancientdiggers.Admin;
import androidx.appcompat.app.AppCompatActivity;

```

```

import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;
import com.fyp.ancientdiggers.AllComplaintActivity;
import com.fyp.ancientdiggers.AllUsersActivity;
import com.fyp.ancientdiggers.MainActivity;
import com.fyp.ancientdiggers.MyInboxActivity;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.SessionManager;
import com.fyp.ancientdiggers.UpdateProfile;
import
com.google.android.material.floatingactionbutton.Floating
ActionButton;

public class AdminDashboard extends AppCompatActivity {

    SessionManager sessionManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_admin_dashboard);
        sessionManager=new
        SessionManager(AdminDashboard.this);

    }
    public void addtourguied(View view){
        Intent intent=new
        Intent(AdminDashboard.this,AddTourGuiedActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);

    }
    public void inbox(View view){
        Intent intent=new
        Intent(AdminDashboard.this,MyInboxActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);

    }

    public void allpayment(View view){
        Intent intent=new
        Intent(AdminDashboard.this,AllPaymentActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);

    }
}

```

```

        public void addencyclopedia(View view){
            Intent intent=new Intent(AdminDashboard.this,
            AddEncyclopedia.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);

        }
        public void alluseractivity(View view){
            Intent intent=new Intent(AdminDashboard.this,
            AllUsersActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);

        }
        public void manageplaces(View view){
            Intent intent=new
            Intent(AdminDashboard.this,ManagePlacesActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);

        }

        public void ManageAll(View view){
            Intent intent=new
            Intent(AdminDashboard.this,ManageAll.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);

        }
        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the
            action bar if it is present.
            getMenuInflater().inflate(R.menu.adminmenu,
            menu);
            return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
            int id = item.getItemId();
            switch (id){
                case R.id.action_logout:

                    Toast.makeText(getApplicationContext(),"Logout",Toast.LEN
                    GTH_LONG).show();
                    sessionManager.setLogin(false);
                    Intent intent=new
                    Intent(AdminDashboard.this, MainActivity.class);

                    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

```



```

        startActivity(intent);

        return true;
    case R.id.action_updateprofile:
        Intent intentupate=new
Intent(getApplicationContext(), UpdateProfile.class);

intentupate.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intentupate);
        return true;

    case R.id.action_complain:

        Intent intentcomplain=new
Intent(AdminDashboard.this, AllComplaintActivity.class);

intentcomplain.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intentcomplain);

        return true;

    default:
        return super.onOptionsItemSelected(item);
    }
}
}

```

Admin place adapter

```

package com.fyp.ancientdiggers.Admin;
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.Utility;
import java.util.ArrayList;
public class AdminPlaceslAdapeter extends
ArrayAdapter<PlacesDataClassadmin> {
    ArrayList<PlacesDataClassadmin> thedata;
    Context context;
}

```

```

        public AdminPlaceslAdapeter(Context context,
        ArrayList<PlacesDataClassadmin> thedata) {
            super(context, R.layout.custom_hospital_layout,
            thedata);
            this.thedata = thedata;
            this.context = context;
        }
        @Override
        public View getView(int position, View convertView,
        ViewGroup parent) {
            LayoutInflater inflater =
            LayoutInflater.from(context);
            View view =
            inflater.inflate(R.layout.custom_place_layout,
            null);

            TextView txtplacename =
            view.findViewById(R.id.txtplacename);
            TextView txtplacedetail =
            view.findViewById(R.id.txtplacedetail);
            TextView txtplaceid =
            view.findViewById(R.id.txtplaceid);
            TextView txtplaceguiedername =
            view.findViewById(R.id.txtplaceguiedername);
            TextView txtassignguied =
            view.findViewById(R.id.txtassignguied);
            txtassignguied.setOnClickListener(new
            View.OnClickListener() {
                @Override
                public void onClick(View v) {

                    Intent intent=new
                    Intent(context,AllGuiedActivity.class);

                    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

                    intent.putExtra("placeid",thedata.get(position).getPlaceid());

                    intent.putExtra("placename",thedata.get(position).getPlacename());

                    intent.putExtra("placeimage",thedata.get(position).getPlaceimage());

                    intent.putExtra("placedetail",thedata.get(position).getPlacedetial());

                    context.startActivity(intent);

                }
            }
        }

```

```

        });

        ImageView
imagevieplace=view.findViewById(R.id.imagevieplace);
        txtplacename.setText("Place Name
:"+thedata.get(position).getPlacename());
        txtplacedetail.setText("Place Detail
:"+thedata.get(position).getPlacedetial());
        txtplaceid.setText("Place Detail
:"+thedata.get(position).getPlaceid());

if(thedata.get(position).getPlaceguiedname().equalsIgnoreCase("Not Guided
Assign")){
        txtplaceguiedername.setText("Not Guided
Assign");
    }
    else {
        txtplaceguiedername.setText("Tour guied Name
:"+thedata.get(position).getPlaceguiedname());
    }
    try{

imagevieplace.setImageBitmap(Utility.stringToBitmap(thedata.get(position).getPlaceimage()));
    }
    catch (Exception er)
    {

imagevieplace.setImageResource(R.mipmap.hunza);
    }

        return view;
    }

}

```

All assign places data class

```

package com.fyp.ancientdiggers.Admin;
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;

```

```

import android.widget.TextView;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.Utility;
import java.util.ArrayList;
public class AdminPlaceslAdapeter extends
ArrayAdapter<PlacesDataClassadmin> {
    ArrayList<PlacesDataClassadmin> thedata;
    Context context;

    public AdminPlaceslAdapeter(Context context,
ArrayList<PlacesDataClassadmin> thedata) {
        super(context, R.layout.custom_hospital_layout,
thedata);
        this.thedata = thedata;
        this.context = context;

    }

    @Override
    public View getView(int position, View convertView,
ViewGroup parent) {
        LayoutInflater inflater =
LayoutInflater.from(context);
        View view =
inflater.inflate(R.layout.custom_place_layout,
null);

        TextView txtplacename =
view.findViewById(R.id.txtplacename);
        TextView txtplacedetail =
view.findViewById(R.id.txtplacedetail);
        TextView txtplaceid =
view.findViewById(R.id.txtplaceid);
        TextView txtplaceguiedername =
view.findViewById(R.id.txtplaceguiedername);
        TextView txtassignguied =
view.findViewById(R.id.txtassignguied);
        txtassignguied.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent=new
Intent(context,AllGuiedActivity.class);

                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

                intent.putExtra("placeid", thedata.get(position).getPlaceid());
            }
        });
    }
}

```

```

intent.putExtra("placename", thedata.get(position).getPlacename());

intent.putExtra("placeimage", thedata.get(position).getPlaceimage());

intent.putExtra("placedetail", thedata.get(position).getPlacedetial());
        context.startActivity(intent);

    }
});

    ImageView
imagevieplace=view.findViewById(R.id.imagevieplace);
    txtplacename.setText("Place Name
:"+thedata.get(position).getPlacename());
    txtplacedetail.setText("Place Detail
:"+thedata.get(position).getPlacedetial());
    txtplaceid.setText("Place Detail
:"+thedata.get(position).getPlaceid());

if(thedata.get(position).getPlaceguiedname().equalsIgnoreCase("")){
    txtplaceguiedername.setText("Not Guided
Assign");
}
else {
    txtplaceguiedername.setText("Tour guied Name
:"+thedata.get(position).getPlaceguiedname());
}

    try{

imagevieplace.setImageBitmap(Utility.stringToBitmap(thedata.get(position).getPlaceimage()));
    }
    catch (Exception er)
    {

imagevieplace.setImageResource(R.mipmap.hunza);
    }

    return view;
}

```

```
}
```

All contact adapter

```
package com.fyp.ancientdiggers.localpeople;
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.ImageView;
import android.widget.TextView;
import com.fyp.ancientdiggers.Admin.MyNewMessages;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.Utility;
import
com.fyp.ancientdiggers.touriest.AllAssingnPlacesDataClass
;
import
com.fyp.ancientdiggers.touriest.UserSelectedPlaceActivity
;
import java.util.ArrayList;
public class AllContactAdapeter extends
ArrayAdapter<DataclassOrgContact> {
    ArrayList<DataclassOrgContact> thedata;
    Context context;

    public AllContactAdapeter(Context context,
ArrayList<DataclassOrgContact> thedata) {
        super(context, R.layout.customcontactlist,
thedata);
        this.thedata = thedata;
        this.context = context;
    }

    @Override
    public View getView(int position, View convertView,
ViewGroup parent) {
        LayoutInflater inflater =
LayoutInflater.from(context);
        View view =
inflater.inflate(R.layout.customcontactlist, null);

        TextView txtwhatsap =
view.findViewById(R.id.txtwhatsap);
        TextView txtaddress =
view.findViewById(R.id.txtaddress);
```

```

        TextView txtemail =
view.findViewById(R.id.txtemail);
        TextView txtcontact =
view.findViewById(R.id.txtcontact);

        txtcontact.setText("Contact
Us"+thedata.get(position).getContact());
        txtwhatsap.setText("Whattsap
Number"+thedata.get(position).getWhatsApp());

txtaddress.setText("Address"+thedata.get(position).getAdd
ress());
        txtemail.setText("Email
"+thedata.get(position).getEmail());

        return view;
    }

}

```

All guide activity

```

package com.fyp.ancientdiggers.Admin;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.Utility;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;

```

```

import java.util.HashMap;
import java.util.Map;
import androidx.appcompat.app.AppCompatActivity;
public class AllGuidedActivity extends AppCompatActivity {

    ListView listviewallguided;
    AllguidedAdapeter hospitalAdapeter;
    ArrayList<AllGuidedDataclass> allGuidedArrayList =new
ArrayList<>();
    String placeimage,placename,placeid,placedetail;
    ImageView imagevieplace;
    TextView txtplaceme,txtplacedetail;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_allguided);

        initializeview();
        getextravalue();
        txtplaceme.setText("Place Name => "+placename);
        txtplacedetail.setText("Place Detail =>
"+placedetail);

        imagevieplace.setImageBitmap(Utility.stringToBitmap(place
image));
        getallhgued(placeimage,placename,placeid);

    }
    public void initializeview(){
        listviewallguided=(ListView)
findViewById(R.id.listviewallguided);
        txtplaceme=(TextView)
findViewById(R.id.txtplacename);
        txtplacedetail=(TextView)
findViewById(R.id.txtplacedetail);
        imagevieplace=(ImageView)
findViewById(R.id.imagevieplace);
    }
    public void getextravalue(){
        Intent intent=getIntent();

        placename=intent.getStringExtra("placename");
        placeimage=intent.getStringExtra("placeimage");
        placeid=intent.getStringExtra("placeid");
        placedetail=intent.getStringExtra("placedetail");
    }
}

```



```

    }
    public void getallhguied(String placeimage,String
    placename,String placeid){

        StringRequest request = new
        StringRequest(Request.Method.GET, "
        https://upwardsol.com/AccientTourGuied/getallguied.php",
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    /*
                    Toast.makeText(View_ALLPost.this, ""+response,
                    Toast.LENGTH_SHORT).show();
                    */    JSONObject object = null;

                                object = new JSONObject(response);
                                JSONArray arrrr =
                                object.getJSONArray("result");

                                if (arrrr != null) {
                                    allGuiedArrayList = new
                                    ArrayList<AllGuiedDataclass>();
                                    for (int i = 0; i <
                                    arrrr.length(); i++) {

                                        JSONObject jsonObject =
                                        arrrr.getJSONObject(i);
                                        int id=
                                        jsonObject.getInt("id");

                                        String name =
                                        jsonObject.getString("name");
                                        String usercontact =
                                        jsonObject.getString("usercontact");
                                        String userpassword =
                                        jsonObject.getString("userpassword");
                                        String userimage =
                                        jsonObject.getString("userimage");

                                        String userassignstatus=
                                        jsonObject.getString("userassignstatus");
                                        String usertype =
                                        jsonObject.getString("usertype");
                                        String useraddress =
                                        jsonObject.getString("useraddress");

```

```

        allGuiedArrayList.add(new
AllGuiedDataclass(name,useraddress,usercontact,userassign
status,userimage,id+""));

    }

        hospitalAdapeter = new
AllguiedAdapeter(AllGuiedActivity.this,
allGuiedArrayList,placename,placeimage,placeid);

listviewallguied.setAdapter(hospitalAdapeter);

    }
    else

Toast.makeText(AllGuiedActivity.this, "You have not
uploaded Work", Toast.LENGTH_SHORT).show();

        } catch (JSONException e) {
            e.printStackTrace();

Toast.makeText(getApplicationContext(), "Json error:Workr
" + e.getMessage(), Toast.LENGTH_LONG).show();
            Log.e("jerror", "Json error: " +
e.getMessage());

        }
    }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError
error) {
            Toast.makeText(AllGuiedActivity.this,
"not success"+error.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    }
    ) {

        protected Map<String, String> getParams()
throws AuthFailureError {
            Map<String, String> parameter = new
HashMap<String, String>();
            /* parameter.put("wid",id+"");*/

```

```

        return parameter;
    }
};
RequestQueue queue =
Volley.newRequestQueue(AllGuiedActivity.this);
queue.add(request);
}
}

```

All guide adapter

```

package com.fyp.ancientdiggers.Admin;
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.Utility;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class AllguiedAdapeter extends
ArrayAdapter<AllGuiedDataclass> {
    ArrayList<AllGuiedDataclass> thedata;
    Context context;
    String placeimage, placename, placeid;

    public AllguiedAdapeter(Context context,
ArrayList<AllGuiedDataclass> thedata, String
placename, String placeimage, String placeid) {
        super(context, R.layout.custom_allguied_layout,
thedata);
        this.thedata = thedata;
        this.context = context;
    }
}

```

```

        this.placeid=placeid;
        this.placeimage=placeimage;
        this.placename=placename;
    }

    @Override
    public View getView(int position, View convertView,
    ViewGroup parent) {
        LayoutInflater inflater =
    LayoutInflater.from(context);
        View view =
    inflater.inflate(R.layout.custom_allguied_layout,
    null);

        TextView txtguiedaddress =
    view.findViewById(R.id.txtguiedaddress);
        TextView txtguiedcontact =
    view.findViewById(R.id.txtguiedcontact);
        TextView txtguiedname =
    view.findViewById(R.id.txtguiedname);
        TextView txtguiedassignplace =
    view.findViewById(R.id.txtguiedassignplace);
        TextView txtassignplace =
    view.findViewById(R.id.txtassignplace);

        ImageView
    imageviewguied=view.findViewById(R.id.imageviewguied);
        txtguiedaddress.setText("Guied
    Address:"+thedata.get(position).getGuiedaddress());
        txtguiedcontact.setText("Guied Contact
    :"+thedata.get(position).getGuiedcontact());
        txtguiedname.setText("Guied Name
    :"+thedata.get(position).getGuiedname());
        txtguiedassignplace.setText("Assign Place
    :"+thedata.get(position).getGuieassignplace());
        txtassignplace.setOnClickListener(new
    View.OnClickListener() {
            @Override
            public void onClick(View v) {

                assignplacetoguied(thedata.get(position).getGuiedid(), the
                data.get(position).getGuiedname(),thedata.get(position).g
                etGuiedcontact(),position,placename,placeimage,placeid);
            }
        });

        try{

```

```

imageviewguied.setImageBitmap(Utility.stringToBitmap(the
ata.get(position).getGuiedimage()));
    }
    catch (Exception er)
    {

imageviewguied.setImageResource(R.mipmap.hunza);
    }

    return view;
}

```

```

    public void assignplacetoguied(String guiedid,String
guiedname,String guiedcontact,int position,String
placename,String placeimage,String placeid){

        StringRequest request = new
StringRequest(Request.Method.POST, "
https://upwardsol.com/AccientTourGuied/insertassignplace.
php", new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Toast.makeText(context, "" + response,
Toast.LENGTH_SHORT).show();
                try {
                    JSONObject data = new
JSONObject(response);
                    String success =
data.getString("success");

                    if
(success.toString().equalsIgnoreCase("1")) {
                        Toast.makeText(context, "Signup
Success" + success, Toast.LENGTH_SHORT).show();
                        Intent intent = new
Intent(context, AddTourGuiedActivity.class);
                        context.startActivity(intent);

                    } else if
(success.toString().equalsIgnoreCase("0")){
                        Toast.makeText(context, "fill
all" + success, Toast.LENGTH_SHORT).show();
                    }

                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}

```

```

        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError
error) {

            Toast.makeText(context, "volley error " +
error, Toast.LENGTH_SHORT).show();
        }
    }) {
        @Override
        protected Map<String, String> getParams()
throws AuthFailureError {

            Map<String, String> parameter = new
HashMap<String, String>();
            parameter.put("guiedid", guiedid+"");
            parameter.put("guiedname", guiedname+"");
            parameter.put("guiedcontact",
guiedcontact+"");
            parameter.put("placename", placename+"");
            parameter.put("placeimage",
placeimage+"");
            parameter.put("placeid", placeid+"");

            // parameter.put("user_id",a);
            return parameter;

//            return super.getParams();
        }
    };
    RequestQueue queue =
Volley.newRequestQueue(context);
    queue.add(request);
}
}

```

All payment activity

```

package com.fyp.ancientdiggers.Admin;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

```

```

import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.EditText;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.touriest.PaymentSlip;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import androidx.appcompat.app.AppCompatActivity;

public class AllPaymentActivity extends AppCompatActivity
{
    ArrayAdapter<DataClassAllpayment> adapter;
    /* access modifiers changed from: private */
    public MyAdapter adapter1;

    String category;
    private EditText etSearch;
    String gender;
    private LinearLayout llContainer;
    /* access modifiers changed from: private */
    public ListView lvProducts;
    private ArrayList<DataClassAllpayment>
mProductArrayList = new ArrayList<>();
    String name,Dist,maridstatus;
    String status;
    int userid;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_allpayment);

    initialize();
    GetallforSearch();

    etSearch.addTextChangedListener(new TextWatcher() {
        public void onTextChanged(CharSequence s, int
start, int before, int count) {
            try {
adapter1.getFilter().filter(s.toString().toLowerCase());
            }
            catch (Exception ex){

            }

        }

        public void beforeTextChanged(CharSequence s,
int start, int count, int after) {

        }

        public void afterTextChanged(Editable s) {

        }

    });

}

public void GetallforSearch(){

    Toast.makeText(this, "hello",
Toast.LENGTH_SHORT).show();

    StringRequest request = new
StringRequest(Request.Method.GET,
"https://upwardsol.com/AccientTourGuied/GetallPayment.php
", new Response.Listener<String>() {

        public void onResponse(String response) {

```



```

        try {
            JSONArray arrrr = new
JSONObject(response).getJSONArray("result");

            if (arrrr != null) {

                mProductArrayList = new
ArrayList<DataClassAllpayment>();

                for (int i = 0; i <
arrrr.length(); i++) {
                    JSONObject data =
arrrr.getJSONObject(i);

                    String
senderid, recieverid, amount, amountslip, created_date, paysta
tus;

                    int
payid=data.getInt("senderid");
                    senderid =
data.getString("senderid");

                    Log.e("allpayimg", senderid+"");
                    recieverid =
data.getString("recieverid");

                    Log.e("allworkercnic", recieverid+"");
                    created_date =
data.getString("created_date");

                    Log.e("allcreated_date", created_date+"");
                    amountslip =
data.getString("amountslip");

                    Log.e("allworkername", amountslip+"");
                    paystatus =
data.getString("paystatus");

                    Log.e("allusername", paystatus+"");
                    amount =
data.getString("amount");

                    Log.e("allamount", amount+"");

                    mProductArrayList.add(new
DataClassAllpayment (payid+"", senderid, recieverid, amount, a
mountslip, created_date, paystatus+""));

```

```

        }
        adapter1 = new
MyAdapter (AllPaymentActivity.this,mProductArrayList);
        lvProducts.setAdapter(adapter1);

    }

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}, new Response.ErrorListener() {
    public void onErrorResponse(VolleyError
error) {

    }
}) {
    /* access modifiers changed from: protected
*/
    public Map<String, String> getParams() throws
AuthFailureError {
        return new HashMap<>();
    }
};
RequestQueue queue =
Volley.newRequestQueue(AllPaymentActivity.this);
queue.add(request);
}

private void initialize() {

    etSearch = (EditText)
findViewById(R.id.inputSearch);
    lvProducts = (ListView)
findViewById(R.id.list_view);
}

@Override
protected void onResume() {
    GetAllforSearch();
    super.onResume();
}

public class MyAdapter extends BaseAdapter implements
Filterable {
    LayoutInflater inflater;
    /* access modifiers changed from: private */
    public ArrayList<DataClassAllpayment>
mDisplayedValues;

```

```

        /* access modifiers changed from: private */
        public ArrayList<DataClassAllpayment>
mOriginalValues;

        private class ViewHolder {
            LinearLayout llContainer;
            TextView txtamount;
            TextView txtpostid;
            TextView txtwid;
            TextView created_date;

            private ViewHolder() {
            }
        }

        public MyAdapter(Context context,
ArrayList<DataClassAllpayment> mProductArrayList) {
            this.mOriginalValues = mProductArrayList;
            this.mDisplayedValues = mProductArrayList;
            this.inflater = LayoutInflater.from(context);
        }

        public int getCount() {
            return this.mDisplayedValues.size();
        }

        public Object getItem(int position) {
            return Integer.valueOf(position);
        }

        public long getItemId(int position) {
            return (long) position;
        }

        public View getView(final int position, View
convertView, ViewGroup parent) {
            ViewHolder holder;
            if (convertView == null) {
                holder = new ViewHolder();
                convertView =
this.inflater.inflate(R.layout.amountrow, null);
                holder.llContainer = (LinearLayout)
convertView.findViewById(R.id.llContainer);
                holder.txtamount = (TextView)
convertView.findViewById(R.id.txtamount);
                holder.txtpostid = (TextView)
convertView.findViewById(R.id.txtpostid);
                holder.created_date = (TextView)
convertView.findViewById(R.id.txtdate);
                holder.txtwid = (TextView)
convertView.findViewById(R.id.txtwid);
            }
        }
    }
}

```

```

        convertView.setTag(holder);
    } else {
        holder = (ViewHolder)
convertView.getTag();
    }
    TextView txtamount = holder.txtamount;
    TextView txtpostid = holder.txtpostid;
    TextView created_date = holder.created_date;
    TextView txtwid = holder.txtwid;

    txtamount.setText("Total Amount-->
"+this.mDisplayedValues.get(position).getAmount()+"Rs");
    txtpostid.setText("Sender Id-->
"+this.mDisplayedValues.get(position).getSenderId());
    created_date.setText("Date -->
"+this.mDisplayedValues.get(position).getCreated_date()+"
");
    txtwid.setText("Reciver Id -->
"+this.mDisplayedValues.get(position).getRecieverid());

    holder.llContainer.setOnClickListener(new
View.OnClickListener() {
        public void onClick(View v) {
            Intent intent=new
Intent(AllPaymentActivity.this, PaymentSlip.class);
            intent.putExtra("recieverid",
String.valueOf(mDisplayedValues.get(position).getReciever
id()+""));
            intent.putExtra("created_date",
String.valueOf(mDisplayedValues.get(position).getCreated_
date()));
            startActivity(intent);

            //Toast.makeText(SearchActivity.this,
((DataSearchClass)
MyAdapter.this.mDisplayedValues.get(position)).getName(),
0).show();
        }
    });
    return convertView;
}

public Filter getFilter() {
    return new Filter() {

        public void publishResults(CharSequence
constraint, FilterResults results) {
            MyAdapter.this.mDisplayedValues =
(ArrayList) results.values;

```

```

MyAdapter.this.notifyDataSetChanged();
    }

    /* access modifiers changed from:
protected */
    public FilterResults
performFiltering(CharSequence constraint) {
    FilterResults results = new
FilterResults();
    ArrayList<DataClassAllpayment>
FilteredArrList = new ArrayList<>();
    if (MyAdapter.this.mOriginalValues ==
null) {
        MyAdapter myAdapter =
MyAdapter.this;
        myAdapter.mOriginalValues = new
ArrayList(myAdapter.mDisplayedValues);
    }
    if (constraint == null ||
constraint.length() == 0) {
        results.count =
MyAdapter.this.mOriginalValues.size();
        results.values =
MyAdapter.this.mOriginalValues;
    } else {
        CharSequence constraint2 =
constraint.toString().toLowerCase();
        for (int i = 0; i <
MyAdapter.this.mOriginalValues.size(); i++) {

            if (((DataClassAllpayment)
MyAdapter.this.mOriginalValues.get(i)).getSenderid().toLo
werCase().startsWith(constraint2.toString())
|| ((DataClassAllpayment)
MyAdapter.this.mOriginalValues.get(i)).getRecieverid().to
LowerCase().startsWith(constraint2.toString())) {

                FilteredArrList.add(new
DataClassAllpayment (MyAdapter.this.mOriginalValues.get(i)
.getPayid()+"" ,
MyAdapter.this.mOriginalValues.get(i).getSenderid()+"" ,
MyAdapter.this.mOriginalValues.get(i).getRecieverid()+"" ,
MyAdapter.this.mOriginalValues.get(i).getAmount(),MyAdapt
er.this.mOriginalValues.get(i).getAmountslip(),MyAdapter.
this.mOriginalValues.get(i).getCreated_date(),MyAdapter.t
his.mOriginalValues.get(i).getPaystatus()));
            }
        }
    }
}
}

```

```

                results.count =
FilteredArrList.size();
                results.values = FilteredArrList;
            }
            return results;
        }
    };
}
}
}
}

```

All places activity for user

```

package com.fyp.ancientdiggers.touriest;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ListView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.LoginUserActivity;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.SessionManager;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
public class AllPlacesActivityForUser extends
AppCompatActivity {
    ListView listViewplaces;
    ArrayList<AllAssingnPlacesDataClass> arrayList;
    AllPlacesAdapeter allPlacesAdapeter;
    SessionManager sessionManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_all_places_for_user);

```

```

        listViewplaces=(ListView)
        findViewById(R.id.listViewplaces);
        sessionManager=new
        SessionManager (AllPlacesActivityForUser.this);

        getallAssignplaces();
        this.setTitle(sessionManager.getName());

    }
    public void getallAssignplaces(){

        StringRequest request = new
        StringRequest(Request.Method.GET, "
        https://upwardsol.com/AccientTourGuided/getallAssingPlaces
        .php", new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    /*
                    Toast.makeText(View_ALLPost.this, ""+response,
                    Toast.LENGTH_SHORT).show();
                    */   JSONObject object = null;

                        object = new JSONObject(response);
                        JSONArray arrrr =
                        object.getJSONArray("result");

                            if (arrrr != null) {
                                arrayList = new
                                ArrayList<AllAssingnPlacesDataClass>();
                                for (int i = 0; i <
                                arrrr.length(); i++) {

                                    JSONObject jsonObject =
                                    arrrr.getJSONObject(i);

                                        // String guidedid=
                                    jsonObject.getString("guidedid");

                                            String guidedname =
                                    jsonObject.getString("guidedname");
                                            String guidedcontact =
                                    jsonObject.getString("guidedcontact");
                                            String placename =
                                    jsonObject.getString("placename");

```

```

                String placeimage =
jsonObject.getString("placeimage");
                String placeid =
jsonObject.getString("placeid");
                /*String hospitalguiedname=
jsonObject.getString("hospitalguiedname");
                String hospitalguiedcontact =
jsonObject.getString("hospitalguiedcontact");
                int hospitalguiedid =
jsonObject.getInt("hospitalguiedid");
*/

                arrayList.add(new
AllAssingnPlacesDataClass("",guiedname,guiedcontact,place
name,placeimage,placeid));

            }

            allPlacesAdapeter = new
AllPlacesAdapeter(AllPlacesActivityForUser.this,
arrayList);

listviewplaces.setAdapter(allPlacesAdapeter);

        }
        else

Toast.makeText(AllPlacesActivityForUser.this, "You have
not uploaded Work", Toast.LENGTH_SHORT).show();

    } catch (JSONException e) {
        e.printStackTrace();

Toast.makeText(getApplicationContext(), "Json error:Workr
" + e.getMessage(), Toast.LENGTH_LONG).show();
        Log.e("jerror", "Json error: " +
e.getMessage());
    }
}
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError
error) {

```



```

Toast.makeText(AllPlacesActivityForUser.this, "not
success"+error.getMessage(), Toast.LENGTH_SHORT).show();
    }
}
) {

        protected Map<String, String> getParams()
throws AuthFailureError {
            Map<String, String> parameter = new
HashMap<String, String>();
            /* parameter.put("wid",id+"");*/
            return parameter;

        }
};
RequestQueue queue =
Volley.newRequestQueue(AllPlacesActivityForUser.this);
queue.add(request);
}

@Override
protected void onResume() {
    super.onResume();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the
action bar if it is present.
    getMenuInflater().inflate(R.menu.adminmenu,
menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    switch (id){
        case R.id.action_logout:

Toast.makeText(getApplicationContext(), "Logout", Toast.LEN
GTH_LONG).show();
            sessionManager.setLogin(false);
            Intent intent=new
Intent(AllPlacesActivityForUser.this,
LoginUserActivity.class);

intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);

```

```

        return true;
        case R.id.action_updateprofile:

Toast.makeText(getApplicationContext(),"Item 2
Selected",Toast.LENGTH_LONG).show();
        return true;

        default:
            return super.onOptionsItemSelected(item);
    }
}
}

```

All places

```

package com.fyp.ancientdiggers.touriest;
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ImageView;
import android.widget.TextView;

import com.fyp.ancientdiggers.Admin.MyNewMessages;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.Utility;
import java.util.ArrayList;
public class AllPlacesAdapeter extends
ArrayAdapter<AllAssingnPlacesDataClass> {
    ArrayList<AllAssingnPlacesDataClass> thedata;
    Context context;

    public AllPlacesAdapeter(Context context,
ArrayList<AllAssingnPlacesDataClass> thedata) {
        super(context,
R.layout.custom_assignplaces_layout, thedata);
        this.thedata = thedata;
        this.context = context;
    }

    @Override
    public View getView(int position, View convertView,
ViewGroup parent) {
        LayoutInflater inflater =
LayoutInflater.from(context);
        View view =
inflater.inflate(R.layout.custom_assignplaces_layou
t, null);

```

```

        TextView txtplacename =
view.findViewById(R.id.txtplacename);
        TextView txtplaceguiedername =
view.findViewById(R.id.txtplaceguiedername);
        TextView txtplaceguiedconatc =
view.findViewById(R.id.txtplaceguiedconatc);

        TextView txtassignguied =
view.findViewById(R.id.txtmoredetail);
        TextView txtnewguiedrequest =
view.findViewById(R.id.txtnewguiedrequest);

        txtassignguied.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent=new Intent(context,
UserSelectedPlaceActivity.class);

intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

intent.putExtra("placeid",thedata.get(position).getPlacei
d());

intent.putExtra("placename",thedata.get(position).getPlac
ename());

intent.putExtra("placeimage",thedata.get(position).getPla
ceimage());

intent.putExtra("guiedname",thedata.get(position).getGuie
dname());

intent.putExtra("guiedcontact",thedata.get(position).getG
uiedcontact());

intent.putExtra("guiedid",Integer.parseInt(thedata.get(po
sition).getGuiedid()));

                context.startActivity(intent);

            }
        });

        txtnewguiedrequest.setOnClickListener(new
View.OnClickListener() {
            @Override

```

```

        public void onClick(View v) {

            Intent intent=new Intent(context,
MyNewMessages.class);

            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

            intent.putExtra("placeid",thedata.get(position).getPlaceid());

            intent.putExtra("placename",thedata.get(position).getPlacename());

            intent.putExtra("placeimage",thedata.get(position).getPlaceimage());

            intent.putExtra("guiedname",thedata.get(position).getGuiedname());

            intent.putExtra("guiedcontact",thedata.get(position).getGuiedcontact());

            context.startActivity(intent);

        }
    });

    ImageView
imagevieplace=view.findViewById(R.id.imagevieplace);
txtplacename.setText("Place Name
:"+thedata.get(position).getPlacename());
txtplaceguiedconatc.setText("Guided Contact
:"+thedata.get(position).getGuiedcontact());
txtplaceguiedername.setText("Guided Name
:"+thedata.get(position).getGuiedname());

    try{

        imagevieplace.setImageBitmap(Utility.stringToBitmap(thedata.get(position).getPlaceimage()));
    }
    catch (Exception er)
    {

        imagevieplace.setImageResource(R.mipmap.hunza);
    }
}

```

```
        return view;
    }
}
```

Chat

```
package com.fyp.ancientdiggers;

public class Chat {
    private String type, message;
    public int getMy_id() {
        return my_id;
    }
    public void setMy_id(int my_id) {
        this.my_id = my_id;
    }
    private int my_id;
    public String getType() {
        return type;
    }
    public String getMessage() {
        return message;
    }
    public Chat(String type, String message, int my_id) {
        this.type = type;
        this.my_id = my_id;
        this.message = message;
    }
}
```

Chat activity

```
package com.fyp.ancientdiggers;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
```

```

import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import
com.fyp.ancientdiggers.touriest.AddPaymentActivity;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
public class ChatActivity extends AppCompatActivity {
    private RecyclerView recyclerView;
    private List<Chat> chatList;
    SessionManager sessionManager;
    private ProgressDialog mProg;

    private int lastsize;

    private EditText chatedittext;
    int id,logintype;
    SessionManager obj;
    String name;
    ImageView btnchatsend;
    TextView textpay;
    Button btnacceptorder;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_chat);
        btnchatsend=findViewById(R.id.btnchatsend);
        textpay=findViewById(R.id.textpay);

```

```

Intent intent=getIntent();
obj = new SessionManager(ChatActivity.this);

id=intent.getIntExtra("recieverid",0);

Toast.makeText(this, "Reciever id is
"+id+obj.getId(), Toast.LENGTH_SHORT).show();

sessionManager=new
SessionManager(ChatActivity.this);

logintype=sessionManager.getLogintpye();
Toast.makeText(this,
"logintype"+sessionManager.getType(),
Toast.LENGTH_SHORT).show();

if(sessionManager.getType().equalsIgnoreCase("Tourist")){
    textpay.setVisibility(View.VISIBLE);
}
else {

    textpay.setVisibility(View.INVISIBLE);

}
textpay.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {

        Intent intentlpy=new Intent(ChatActivity.this,
AddPaymentActivity.class);

intentlpy.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
startActivity(intentlpy);
    }
});

ImageView btnchatsend =
findViewById(R.id.btnchatsend);

    chatedittext = findViewById(R.id.chataddtext);

    recyclerView =
findViewById(R.id.chatrecycleview);

    recyclerView.setLayoutManager(new
LinearLayoutManager(this));

```

```

        mProg = new ProgressDialog(this);
        mProg.setTitle(R.string.app_name);
        mProg.setMessage("Loading...");
        mProg.show();

        btnchatsend.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

                // Toast.makeText(ChatActivity.this, "post
id"+obj.getId()+"", Toast.LENGTH_SHORT).show();
                if
(!chatedittext.getText().toString().trim().equals("") ||
!chatedittext.getText().toString().equals(" ")) {

                    sendmessage(id,obj.getId(),chatedittext.getText().toStrin
g(),obj.getId()+"");

                }
            }
        });
        chatedittext.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                recyclerView.scrollToPosition(lastsize -
1);
            }
        });
        final Handler handleCheckStatus = new Handler();
        //check status every 5 sec
        handleCheckStatus.postDelayed(new Runnable() {
            @Override
            public void run() {
                showusermessages();
                handleCheckStatus.postDelayed(this,
5000);
            }
        }, 5000);
        showusermessages();
    }
    public void insetintoinbox(String msg) {
        StringRequest request = new
StringRequest(Request.Method.POST,
"https://upwardsol.com/AccientTourGuied/insertinbox.php",
new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {

```



```

        JSONObject data=new
JSONObject(response);
        String
success=data.getString("success");

        if(success.equalsIgnoreCase("1")){
            Toast.makeText(
ChatActivity.this, "Insert inbox ",
Toast.LENGTH_SHORT).show();
            /* Intent intent=new
Intent(UpdateCategory.this,ManageAll.class);
startActivity(intent);*/
        }
        else
if(success.equalsIgnoreCase("0")){
            Toast.makeText(
ChatActivity.this, "fill all"+success,
Toast.LENGTH_SHORT).show();
        }

        } catch (JSONException e) {
            e.printStackTrace();
        }

    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError
error) {

        Toast.makeText( ChatActivity.this,
"volly error "+error, Toast.LENGTH_SHORT).show();
    }
})
{
    @Override
    protected Map<String, String> getParams()
throws AuthFailureError {

        Map<String, String> parameter = new
HashMap<String, String>();
        parameter.put("rid", id+"");
        parameter.put("sname", obj.getName()+"");
        parameter.put("message", msg+"");
        parameter.put("sid", obj.getId()+"");

        return parameter;
    }
}

```

```

        }
    }
    ;
    RequestQueue queue = Volley.newRequestQueue(
ChatActivity.this);
    queue.add(request);

}

private class chatadapter extends
RecyclerView.Adapter<chatadapter.chatholder> {
    private Context mContext;
    private List<Chat> chats;
    public chatadapter(Context mContext, List<Chat>
chats) {
        this.mContext = mContext;
        this.chats = chats;
    }
    @NonNull
    @Override
    public chatholder onCreateViewHolder(@NonNull
ViewGroup ViewGroup, int i) {
        View view =
LayoutInflater.from(mContext).inflate(R.layout.recycle_chat_1
ive, ViewGroup, false);
        return new chatholder(view);
    }
    @Override
    public void onBindViewHolder(@NonNull chatholder
holder, int i) {
        Chat chat = chats.get(i);
        int myid = chat.getMy_id();
        if (myid==id) {

holder.receiverlayout.setVisibility(View.VISIBLE);

holder.chatmessage.setText(chat.getMessage());
        } else {

holder.senderlayout.setVisibility(View.VISIBLE);

holder.chatusermessage.setText(chat.getMessage());
        holder.chatusername.setText(name);
        }
    }
    @Override
    public int getItemCount() {
        lastsize = chats.size();
        return chats.size();
    }
}

```

```

class chatholder extends RecyclerView.ViewHolder
{
    ImageView chatuserimg;
    TextView chatusername, chatusermessage,
chatmessage;
    LinearLayout senderlayout, receiverlayout;
    public chatholder(@NonNull View itemView) {
        super(itemView);
        chatuserimg =
itemView.findViewById(R.id.chatuserimg);
        chatusername =
itemView.findViewById(R.id.chatusername);
        chatusermessage =
itemView.findViewById(R.id.chatusermessage);
        chatmessage =
itemView.findViewById(R.id.chatmessage);
        senderlayout =
itemView.findViewById(R.id.senderlayout);
        receiverlayout =
itemView.findViewById(R.id.receiverlayout);
    }
}

public void showusermessages()
{
    Log.d("msg", "msg");
    StringRequest request = new
StringRequest(Request.Method.POST,
"https://upwardsol.com/AccientTourGuied/showuserchat.php"
, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {

            Log.d("doner", response +
obj.getId()+id+"");
            try {
                mProg.hide();
                JSONObject object = new
JSONObject(response);
                JSONArray complaints =
object.getJSONArray("result");
                chatList = new ArrayList<>();
                for (int i = 0; i <
complaints.length(); i++) {
                    JSONObject object1 =
complaints.getJSONObject(i);
                    String message =
object1.getString("message");
                    String date =
object1.getString("createddate");

```

```

        int my_id =
object1.getInt("rid");
        chatList.add(new Chat(date,
message,my_id));
        chatadapter adaper = new
chatadapter( ChatActivity.this, chatList);
        recyclerView.setAdapter(adaper);
    }
    } catch (JSONException e) {
        Toast.makeText(ChatActivity.this,
""+e.getMessage(), Toast.LENGTH_SHORT).show();

        e.printStackTrace();
    }
}
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError
error) {
        Toast.makeText(ChatActivity.this,
""+error.getMessage(), Toast.LENGTH_SHORT).show();
    }
}) {
    @Override
    protected Map<String, String> getParams()
throws AuthFailureError {
//        return super.getParams();
        Map<String, String> params = new
HashMap<String, String>();
        params.put("sid", obj.getId() + "");
        params.put("rid",id+ "");

        return params;
    }
};
RequestQueue rQueue = Volley.newRequestQueue(
ChatActivity.this);
rQueue.add(request);

}

public void sendmessage(final int sid, final int
sid_number, final String message,String postid)
{
    StringRequest request = new
StringRequest(Request.Method.POST,
"https://upwardsol.com/AccientTourGuied/sendmessage.php",
new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {

```

```

        Toast.makeText(ChatActivity.this,
"resbpn "+response, Toast.LENGTH_SHORT).show();
        Log.d("irrespone", response);
        // CustomProgressDialog.hide();
        JSONObject data = new
JSONObject(response);
        int success=data.getInt("success");
        String msg=data.getString("msg");
        if(success==1)
        {

                showusermessages();

insetintoinbox(chatedittext.getText().toString());
                chatedittext.setText("");
        }
        else
        {
                new AlertDialog.Builder(
ChatActivity.this).setMessage(msg).setTitle("Sorry!").set
PositiveButton("Ok", new
DialogInterface.OnClickListener() {
                @Override
                public void
onClick(DialogInterface dialog, int which) {
                        dialog.dismiss();
                }
                }).show();
        }

        } catch (JSONException e) {
                e.printStackTrace();
                // CustomProgressDialog.hide();

                new AlertDialog.Builder(
ChatActivity.this).setMessage("An error Occured! Please
try again!").setTitle("Alert!").setPositiveButton("Ok",
new DialogInterface.OnClickListener() {
                @Override
                public void
onClick(DialogInterface dialog, int which) {
                        dialog.dismiss();
                }
                }).show();

        }

//
Toast.makeText(getApplicationChatActivity.this(), "Json
error: " + e.getMessage(), Toast.LENGTH_LONG).show();
        Log.e("jerror","Json error: " +
e.getMessage());

```

```

        }
    }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError
volleyError) {
//            Toast.makeText(Quiz.this, "Error: ",
Toast.LENGTH_SHORT).show();
            new AlertDialog.Builder(
ChatActivity.this).setMessage("An error Occured! Please
try again!").setTitle("Alert!").setPositiveButton("Ok",
new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface
dialog, int which) {
//                    CustomProgressDialog.hide();
                    dialog.dismiss();
                }
            }).show();
            Log.e("verror", "Error:
"+volleyError.getMessage());
        }
    }) {
        @Override
        protected Map<String, String> getParams()
throws AuthFailureError {
            Map<String, String> parameters = new
HashMap<String, String>();

            parameters.put("message", chatedittext.getText().toString(
));

            parameters.put("rid", id+"");
            parameters.put("sid", obj.getId()+"");

            return parameters;
        }
    };
    RequestQueue rQueue = Volley.newRequestQueue(
ChatActivity.this);
    rQueue.add(request);

}}

```

Data class all payment

```

package com.fyp.ancientdiggers.Admin;
public class DataClassAllpayment {
String
payid, senderid, recieverid, amount, amountslip, created_date,
paystatus;

```

```

    public DataClassAllpayment(String payid, String
senderid, String recieverid, String amount, String
amountslip, String created_date, String paystatus) {
        this.payid = payid;
        this.senderid = senderid;
        this.recieverid = recieverid;
        this.amount = amount;
        this.amountslip = amountslip;
        this.created_date = created_date;
        this.paystatus = paystatus;
    }

    public String getPayid() {
        return payid;
    }

    public void setPayid(String payid) {
        this.payid = payid;
    }

    public String getSenderid() {
        return senderid;
    }

    public void setSenderid(String senderid) {
        this.senderid = senderid;
    }

    public String getRecieverid() {
        return recieverid;
    }

    public void setRecieverid(String recieverid) {
        this.recieverid = recieverid;
    }

    public String getAmount() {
        return amount;
    }

    public void setAmount(String amount) {
        this.amount = amount;
    }

    public String getAmountslip() {
        return amountslip;
    }

    public void setAmountslip(String amountslip) {
        this.amountslip = amountslip;
    }

```

```

    }

    public String getCreated_date() {
        return created_date;
    }

    public void setCreated_date(String created_date) {
        this.created_date = created_date;
    }

    public String getPaystatus() {
        return paystatus;
    }

    public void setPaystatus(String paystatus) {
        this.paystatus = paystatus;
    }
}

```

Data class encyclopedia

```

package com.fyp.ancientdiggers.localpeople;

public class DataClassEncyclopedia {
    String
    encyclopediatitle,encyclopediadetail,encyclopediainage,en
    cyclopediadate;

    public DataClassEncyclopedia(String
    encyclopediatitle, String encyclopediadetail, String
    encyclopediainage, String encyclopediadate) {
        this.encyclopediatitle = encyclopediatitle;
        this.encyclopediadetail = encyclopediadetail;
        this.encyclopediainage = encyclopediainage;
        this.encyclopediadate = encyclopediadate;
    }

    public String getEncyclopediatitle() {
        return encyclopediatitle;
    }

    public void setEncyclopediatitle(String
    encyclopediatitle) {
        this.encyclopediatitle = encyclopediatitle;
    }

    public String getEncyclopediadetail() {
        return encyclopediadetail;
    }
}

```



```

    public void setEncyclopediadetail(String
encyclopediadetail) {
        this.encyclopediadetail = encyclopediadetail;
    }

    public String getEncyclopediainage() {
        return encyclopediainage;
    }

    public void setEncyclopediainage(String
encyclopediainage) {
        this.encyclopediainage = encyclopediainage;
    }

    public String getEncyclopediadate() {
        return encyclopediadate;
    }

    public void setEncyclopediadate(String
encyclopediadate) {
        this.encyclopediadate = encyclopediadate;
    }
}

```

Data class manage all

```

package com.fyp.ancientdiggers.Admin;

public class DataClassManageAll {
    String
    name,contact,password,address,image,cnic,type,status,id;

    public DataClassManageAll(String name, String
contact, String password,
                                String address, String
image, String cnic, String type, String status, String
id) {
        this.name = name;
        this.contact = contact;
        this.password = password;
        this.address = address;
        this.image = image;
        this.cnic = cnic;
        this.type = type;
        this.status = status;
        this.id = id;
    }

    public String getName() {
        return name;
    }
}

```

```
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getContact() {
        return contact;
    }

    public void setContact(String contact) {
        this.contact = contact;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getImage() {
        return image;
    }

    public void setImage(String image) {
        this.image = image;
    }

    public String getCnic() {
        return cnic;
    }

    public void setCnic(String cnic) {
        this.cnic = cnic;
    }

    public String getType() {
        return type;
    }

    public void setType(String type) {
```

```

        this.type = type;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }
}

```

Data class org contact

```

package com.fyp.ancientdiggers.localpeople;

public class DataclassOrgContact {

    String id,orgid,contact,whatsApp,email,address;

    public DataclassOrgContact(String id, String orgid,
String contact, String whatsApp, String email, String
address) {
        this.id = id;
        this.orgid = orgid;
        this.contact = contact;
        this.whatsApp = whatsApp;
        this.email = email;
        this.address = address;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getOrgid() {
        return orgid;
    }
}

```

```

public void setOrgid(String orgid) {
    this.orgid = orgid;
}

public String getContact() {
    return contact;
}

public void setContact(String contact) {
    this.contact = contact;
}

public String getWhatsApp() {
    return whatsapp;
}

public void setWhatsApp(String whatsapp) {
    this.whatsapp = whatsapp;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}
}

```

Local people dash board

```

package com.fyp.ancientdiggers.localpeople;

import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;
import com.fyp.ancientdiggers.AllUsersActivity;
import com.fyp.ancientdiggers.FeedbackActivity;
import com.fyp.ancientdiggers.MainActivity;

```

```

import com.fyp.ancientdiggers.MyInboxActivity;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.SessionManager;
import com.fyp.ancientdiggers.UpdateProfile;
import
com.google.android.material.floatingactionbutton.Floating
ActionButton;

import androidx.appcompat.app.AppCompatActivity;

public class LocelPoeplesDashboard extends
AppCompatActivity {
FloatingActionButton fabchat;
    SessionManager sessionManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_people_dashboard);
        sessionManager=new
SessionManager(LocelPoeplesDashboard.this);
        fabchat=findViewById(R.id.fabchat);

        fabchat.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent=new
Intent(LocelPoeplesDashboard.this,
AllUsersActivity.class);
                startActivity(intent);
            }
        });
    }
    public void searencyclopedia(View view){
        Intent intent=new
Intent(LocelPoeplesDashboard.this,
SearchEncyclopediaActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
    }
    public void allusers(View view){
        Intent intent=new
Intent(LocelPoeplesDashboard.this,
AllUsersActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
    }
}

```

```

        public void feedback(View view){
            Intent intent=new
Intent (LocelPoeplesDashboard.this,
FeedbackActivity.class);
            intent.addFlags (Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);

        }
        public void inbox(View view){
            Intent intentinbox=new
Intent (LocelPoeplesDashboard.this, MyInboxActivity.class);

intentinbox.addFlags (Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intentinbox);

        }

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the
action bar if it is present.
            getMenuInflater().inflate(R.menu.adminmenu,
menu);
            return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
            int id = item.getItemId();
            switch (id){
                case R.id.action_logout:

Toast.makeText(getApplicationContext(), "Logout", Toast.LEN
GTH_LONG).show();
                    sessionManager.setLogin(false);
                    Intent intent=new
Intent (LocelPoeplesDashboard.this, MainActivity.class);

intent.addFlags (Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(intent);

                    return true;
                case R.id.action_updateprofile:

                    Intent intentupate=new
Intent (getApplicationContext(), UpdateProfile.class);

intentupate.addFlags (Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(intentupate);
            }
        }
    }
}

```

```

        return true;

        default:
            return super.onOptionsItemSelected(item);
    }
}

```

Manage all

```

package com.fyp.ancientdiggers.Admin;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
public class ManageAll extends AppCompatActivity {

    String
    name, contact, password, address, image, cnic, type, status;
    ArrayList<DataClassManageAll> allArrayList;
    ListView listView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_manage_all);
        listView=findViewById(R.id.listViewmannageall);
        getAll();
    }
    public void getAll() {

```

```

        StringRequest request = new
StringRequest(Request.Method.GET, "
https://upwardsol.com/AccientTourGuied/getallrequest.php"
, new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        // Toast.makeText(ManageAll.this,
"response pat" + response, Toast.LENGTH_SHORT).show();
        try {
            JSONObject object = null;

            allArrayList=new ArrayList<>();

            object = new JSONObject(response);
            JSONArray arrrr =
object.getJSONArray("result");

                if (arrrr != null) {
                    for (int i = 0; i <
arrrr.length(); i++) {

                                JSONObject jsonObject =
arrrr.getJSONObject(i);

                                    //
Toast.makeText(GetAllApointment.this, "json
obj"+jsonObject, Toast.LENGTH_SHORT).show();

                                type =
jsonObject.getString("type");
                                    name =
jsonObject.getString("name");
                                    cnic =
jsonObject.getString("cnic");
                                    status =
jsonObject.getString("status");
                                    password =
jsonObject.getString("password");
                                    address =
jsonObject.getString("address");
                                    contact =
jsonObject.getString("contact");
                                    int id=
jsonObject.getInt("id");
                                    image =
jsonObject.getString("image");
                                    // trade =
jsonObject.getString("trade");

```



```

                //  exprience =
jsonObject.getString("wExprience");
                //  wimg =
jsonObject.getString("wimg");

                /* logo =
jsonObject.getString("doctorlogo");
                dr_id =
jsonObject.getString("id");*/

                allArrayList.add(new
DataClassManageAll (name, contact, password, address, image, cn
ic, type, status, id+""));

        }

        AdapterManageAll  adapter = new
AdapterManageAll (ManageAll.this, allArrayList);

        listView.setAdapter(adapter);

    }

    } catch (JSONException e) {
        e.printStackTrace();

        Toast.makeText(getApplicationContext(), "Json error:
patientss " + e.getMessage(), Toast.LENGTH_LONG).show();
        Log.e("jerror", "Json error: " +
e.getMessage());
    }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError
error) {
            Toast.makeText(ManageAll.this, "not
success" + error.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    }
    ) {

        protected Map<String, String> getParams()
throws AuthFailureError {
            Map<String, String> parameter = new
HashMap<String, String>();

```

```

        return parameter;
    }
};
RequestQueue queue =
Volley.newRequestQueue(ManageAll.this);
queue.add(request);

listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?>
parent, View view, int position, long id) {

        /* Intent intent = new
Intent(Manage_ALL_Dri_Vehicle.this,
takedata_id_edit_delete_driver.class);

intent.putExtra("id",allapoint.get(position).getId()+"");

intent.putExtra("name",allapoint.get(position).getName()+
"");

intent.putExtra("cnic",allapoint.get(position).getCnic()+
"");

intent.putExtra("password",allapoint.get(position).getPas
sword()+"");

intent.putExtra("contactnum",allapoint.get(position).getC
ontactnum()+"");

intent.putExtra("experience",allapoint.get(position).getE
xperience()+"");

        startActivity(intent);*/
    }
});
}
}

```

Message

```

package com.fyp.ancientdiggers.Admin;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;

```

```

import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.SessionManager;
import
com.fyp.ancientdiggers.tourguied.CustomAdapterNewMessages
;
import
com.fyp.ancientdiggers.tourguied.DataClassNewMsges;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import androidx.appcompat.app.AppCompatActivity;
public class MyNewMessages extends AppCompatActivity {
    ArrayAdapter<DataClassNewMsges> adapter;
    ArrayList<DataClassNewMsges> allapoint;
    String category,region;
    String gender;
    ListView listView;
    String maridstatus;
    String name,myname;
    String placeid;
    int sender;
    SessionManager obj;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_my_new_messages);

        listView = (ListView)
findViewById(R.id.lstmymsges);
        allapoint = new ArrayList<DataClassNewMsges>();
        Intent intent = getIntent();

        placeid = intent.getStringExtra("placeid");

```

```

        Toast.makeText(this, placeid+placeid,
Toast.LENGTH_SHORT).show();

        obj=new SessionManager(MyNewMessages.this);

        myname = obj.getName();

        getNewMessages();
    }

    public void getNewMessages() {
        StringRequest request = new
StringRequest(Request.Method.POST, "
https://upwardsol.com/AccientTourGuied/showinbox.php",
new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject object = null;

                    object = new JSONObject(response);
                    JSONArray arrrr =
object.getJSONArray("result");

                    if (arrrr != null) {
                        for (int i = 0; i <
arrrr.length(); i++) {
                            Log.d("doner", response +
"");
                            JSONObject jsonObject =
arrrr.getJSONObject(i);

                            //
Toast.makeText(GetAllApointment.this, "json
obj"+jsonObject, Toast.LENGTH_SHORT).show();
                            String    sid=
jsonObject.getString("sid");
                            String    rid=
jsonObject.getString("rid");
                            String    message=
jsonObject.getString("messages");
                            String sname =
jsonObject.getString("sname");
                            /* logo =
jsonObject.getString("doctorlogo");
                            dr_id =
jsonObject.getString("id");*/

```

```

                allapoint.add(new
DataClassNewMsges(sid,rid,sname,message));
            }

            adapter = new
CustomAdapterNewMessages(MyNewMessages.this,
allapoint,sender,myname);

            listView.setAdapter(adapter);
        }

        } catch (JSONException e) {
            e.printStackTrace();

Toast.makeText(getApplicationContext(), "Json error:
patientss " + e.getMessage(), Toast.LENGTH_LONG).show();
            Log.e("jerror", "Json error: " +
e.getMessage());
        }
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError
error) {
        Toast.makeText(MyNewMessages.this, "not
success" + error.getMessage(),
Toast.LENGTH_SHORT).show();
    }
}
) {

        protected Map<String, String> getParams()
throws AuthFailureError {
            Map<String, String> parameter = new
HashMap<String, String>();
            parameter.put("rid",obj.getId()+"");

            return parameter;

        }
    };

    RequestQueue queue = Volley.newRequestQueue(
MyNewMessages.this);
    queue.add(request);

    listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override

```

```

        public void onItemClick(AdapterView<?>
parent, View view, int position, long id) {

            /* Intent intent = new
Intent(Manage_ALL_Dri_Vehicle.this,
takedata_id_edit_delete_driver.class);

intent.putExtra("id",allapoint.get(position).getId()+"");

intent.putExtra("name",allapoint.get(position).getName()+
"");

intent.putExtra("cnic",allapoint.get(position).getCnic()+
"");

intent.putExtra("password",allapoint.get(position).getPas
sword()+"");

intent.putExtra("contactnum",allapoint.get(position).getC
ontactnum()+"");

intent.putExtra("experience",allapoint.get(position).getE
xperience()+"");

            startActivity(intent);*/

        }
    });
}
}

```

Organization contact list

```

package com.fyp.ancientdiggers.localpeople;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ListView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

```

```

import com.fyp.ancientdiggers.LoginUserActivity;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.SessionManager;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import androidx.appcompat.app.AppCompatActivity;
public class OrganizationContactlist extends
AppCompatActivity {
    ListView listviewplaces;
    ArrayList<DataclassOrgContact> arrayList;
    AllContactAdapeter allPlacesAdapeter;
    SessionManager sessionManager;
    Intent intent;
    int orgid;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_all_orgcontact);
        listviewplaces=(ListView)
findViewById(R.id.listviewplaces);
        sessionManager=new
SessionManager(OrganizationContactlist.this);
        intent=getIntent();
        orgid=intent.getIntExtra("orgid",0);

        getallAssignplaces();

    }
    public void getallAssignplaces(){

        StringRequest request = new
StringRequest(Request.Method.POST, "
https://upwardsol.com/AccientTourGuied/getorgcontact.php"
, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    /*
Toast.makeText(View_ALLPost.this, ""+response,
Toast.LENGTH_SHORT).show();
                    */
                    JSONObject object = null;

```

```

        object = new JSONObject(response);
        JSONArray arrrr =
object.getJSONArray("result");

        if (arrrr != null) {
            arrayList = new
ArrayList<DataclassOrgContact>();
            for (int i = 0; i <
arrrr.length(); i++) {

                JSONObject jsonObject =
arrrr.getJSONObject(i);

                int id;
                String orgid ,contact
,whatsApp ,email ,address;

                id = jsonObject.getInt("id");
                orgid =
jsonObject.getString("orgid");
                contact =
jsonObject.getString("contact");
                whatsApp =
jsonObject.getString("whatsApp");
                email =
jsonObject.getString("email");
                address =
jsonObject.getString("address");
                /*String hospitalguiedname=
jsonObject.getString("hospitalguiedname");
                String hospitalguiedcontact =
jsonObject.getString("hospitalguiedcontact");
                int hospitalguiedid =
jsonObject.getInt("hospitalguiedid");
                */

                arrayList.add(new
DataclassOrgContact(id+"",orgid ,contact ,whatsApp ,email
,address));

            }

            allPlacesAdapeter = new
AllContactAdapeter(OrganizationContactlist.this,
arrayList);

```



```

listviewplaces.setAdapter(allPlacesAdapeter);

        }
        else

Toast.makeText(OrganizationContactlist.this, "You have
not uploaded Work", Toast.LENGTH_SHORT).show();

        } catch (JSONException e) {
            e.printStackTrace();

Toast.makeText(getApplicationContext(), "Json error:Workr
" + e.getMessage(), Toast.LENGTH_LONG).show();
            Log.e("jerror", "Json error: " +
e.getMessage());
        }
    }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError
error) {

Toast.makeText(OrganizationContactlist.this, "not
success"+error.getMessage(), Toast.LENGTH_SHORT).show();
        }
    }
) {

        protected Map<String, String> getParams()
throws AuthFailureError {
            Map<String, String> parameter = new
HashMap<String, String>();
            parameter.put("orgid",orgid+"");
            return parameter;

        }
    };
    RequestQueue queue =
Volley.newRequestQueue(OrganizationContactlist.this);
    queue.add(request);
}

@Override
protected void onResume() {
    super.onResume();
}
}

```

```

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the
            action bar if it is present.
            getMenuInflater().inflate(R.menu.adminmenu,
            menu);
            return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
            int id = item.getItemId();
            switch (id) {
                case R.id.action_logout:

                Toast.makeText(getApplicationContext(), "Logout", Toast.LEN
                GTH_LONG).show();
                    sessionManager.setLogin(false);
                    Intent intent=new
                Intent(OrganizationContactlist.this,
                LoginActivity.class);

                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(intent);

                    return true;
                case R.id.action_updateprofile:

                Toast.makeText(getApplicationContext(), "Item 2
                Selected", Toast.LENGTH_LONG).show();
                    return true;

                default:
                    return super.onOptionsItemSelected(item);
            }
        }
    }
}

```

Organization dashboard

```

package com.fyp.ancientdiggers.organization;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import com.fyp.ancientdiggers.AllUsersActivity;
import com.fyp.ancientdiggers.FeedbackActivity;
import com.fyp.ancientdiggers.MainActivity;

```

```

import com.fyp.ancientdiggers.MyInboxActivity;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.SessionManager;
import com.fyp.ancientdiggers.UpdateProfile;
import androidx.appcompat.app.AppCompatActivity;
public class OrganizationDashboard extends
AppCompatActivity {
    SessionManager sessionManager;
    TextView txtchat;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_organization_dashboard);
        sessionManager=new
SessionManager(OrganizationDashboard.this);
        txtchat=findViewById(R.id.txtchat);

        txtchat.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent=new
Intent(OrganizationDashboard.this,
AllUsersActivity.class);
                startActivity(intent);
            }
        });

        public void function(View view){
            int id=view.getId();
            if(id==R.id.cardcreataccount){

                Intent intent=new
Intent(OrganizationDashboard.this,
AddAccountActivity.class);

intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(intent);
            }

            if(id==R.id.cardfeedback){

                Intent intent=new
Intent(OrganizationDashboard.this,
FeedbackActivity.class);

```

```

intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);

    }
    if(id==R.id.cardpostfind){

        Intent intent=new
Intent(OrganizationDashboard.this,
PostfindingActivity.class);

intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);

    }
    if(id==R.id.cardreport){

        Intent intent=new
Intent(OrganizationDashboard.this,
AllUsersActivity.class);

intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);

    }
    if(id==R.id.carinbox){

        Intent intentinbox=new
Intent(OrganizationDashboard.this,
MyInboxActivity.class);

intentinbox.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intentinbox);

    }
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the
action bar if it is present.
    getMenuInflater().inflate(R.menu.adminmenu,
menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

```

```

        switch (id) {
            case R.id.action_logout:

Toast.makeText(getApplicationContext(), "Logout", Toast.LENGTH_LONG).show();
                sessionManager.setLogin(false);
                Intent intent=new
Intent(OrganizationDashboard.this, MainActivity.class);

                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(intent);

                return true;
            case R.id.action_updateprofile:
                Intent intentupate=new
Intent(getApplicationContext(), UpdateProfile.class);

                intentupate.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(intentupate);
                return true;

            default:
                return super.onOptionsItemSelected(item);
        }
    }
}

```

Payment slip

```

package com.fyp.ancientdiggers.touriest;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.Utility;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;

```

```

import java.util.Map;
import androidx.appcompat.app.AppCompatActivity;
public class PaymentSlip extends AppCompatActivity {
    ImageView imgpay;
        String recieverid;

        TextView
textViewsenderid, textViewdate, textViewrecieverid, textamou
nt;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_payment_slip);
        textamount= findViewById(R.id.textamount);
        textViewsenderid=
findViewById(R.id.textViewsenderid);
        textViewrecieverid=
findViewById(R.id.textViewrecieverid);
        textViewdate= findViewById(R.id.textViewdate);

        imgpay=(ImageView) findViewById(R.id.imagepay);
        Intent intent=getIntent();
        recieverid=intent.getStringExtra("recieverid");
        Toast.makeText(this, "payimg"+recieverid,
Toast.LENGTH_SHORT).show();

        // );
        Getpaymentslip();
    }
    public void Getpaymentslip(){

        StringRequest request = new
StringRequest(Request.Method.POST,
"https://upwardsol.com/AccientTourGuied/paymentslip.php",
new Response.Listener<String>() {

            @SuppressWarnings("SetTextI18n")
            public void onResponse(String response) {
                try {
                    Log.e("quesns", "Executed" +
response.toString());
                    JSONObject data = new
JSONObject(response); // create JSON obj from string
                    String success =
data.getString("success").toString();

                    if(success.equalsIgnoreCase("1")) {

```

```

        String
        senderid,recieverid,amount,amountslip,created_date,paysta
        tus;

        senderid =
        data.getString("senderid");
        Log.e("allpayimg",senderid+"");
        recieverid =
        data.getString("recieverid");

        Log.e("allworkercnic",recieverid+"");
        created_date =
        data.getString("created_date");

        Log.e("allcreated_date",created_date+"");
        amountslip =
        data.getString("amountslip");

        Log.e("allworkername",amountslip+"");
        paystatus =
        data.getString("paystatus");

        Log.e("allusername",paystatus+"");
        amount =
        data.getString("amount");
        Log.e("allamount",amount+"");

        imgpay.setImageBitmap(Utility.stringToBitmap(amountslip))
        ;
        textamount.setText(amount + "");

        textViewrecieverid.setText(recieverid + "");
        textviewsenderid.setText(senderid
        + "");
        textViewdate.setText(created_date
        + "");

    }

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}, new Response.ErrorListener() {
    public void onErrorResponse(VolleyError
error) {

```

```

    }
    }) {
        /* access modifiers changed from: protected
*/
        public Map<String, String> getParams() throws
AuthFailureError {
            Map<String, String> parameter = new
HashMap<String, String>();

parameter.put("recieverid",recieverid+"");

            return parameter;
        }
    };
    RequestQueue queue =
Volley.newRequestQueue(PaymentSlip.this);
    queue.add(request);
}
}

```

Post finding activity

```

package com.fyp.ancientdiggers.organization;
import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.BitmapDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.SessionManager;
import com.fyp.ancientdiggers.Utility;
import org.json.JSONException;
import org.json.JSONObject;

```



```

import java.io.FileNotFoundException;
import java.io.InputStream;
import java.util.HashMap;
import java.util.Map;
import androidx.appcompat.app.AppCompatActivity;
public class PostfindingActivity extends
AppCompatActivity {

    Button btnregister;
    private static final int SELECT_PICTURE = 1;
    private static final int CAMERA_REQUEST = 1888;
    int flaging;
    ImageView imageView;
    String url =
"https://upwardsol.com/AccientTourGuied/Addfindingthings.
php";
    EditText edtfindingdetail, edtfindingname,
edtfindingplace;
    SessionManager sessionManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_postfinding);
        sessionManager=new
SessionManager(PostfindingActivity.this);

        edtfindingdetail =
findViewById(R.id.edtfindingdetail);
        edtfindingname =
findViewById(R.id.edtfindingname);
        edtfindingplace =
findViewById(R.id.edtfindingplace);

        imageView= findViewById(R.id.imaguser);

        /* btnregister.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

                });*/

    }
    public void createaccountuser(View view){

```

```

        StringRequest request = new
StringRequest(Request.Method.POST, url, new
Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        Toast.makeText(PostfindingActivity.this,
"" + response, Toast.LENGTH_SHORT).show();
        try {
            JSONObject data = new
JSONObject(response);
            String success =
data.getString("success");

            if
(success.toString().equalsIgnoreCase("1")) {

Toast.makeText(PostfindingActivity.this, "Add Success" +
success, Toast.LENGTH_SHORT).show();

                } else if
(success.toString().equalsIgnoreCase("0")){

Toast.makeText(PostfindingActivity.this, "fill " +
success, Toast.LENGTH_SHORT).show();
                }

            } catch (JSONException e) {
                e.printStackTrace();
            }

        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError
error) {

            Toast.makeText(PostfindingActivity.this,
"volly error " + error, Toast.LENGTH_SHORT).show();
        }
    }) {
        @Override
        protected Map<String, String> getParams()
throws AuthFailureError {

```

```

        Map<String, String> parameter = new
HashMap<String, String>();

        parameter.put("orgid",
sessionManager.getId()+"");
        parameter.put("findtname",
edtfindingname.getText().toString());
        parameter.put("findingplace",
edtfindingplace.getText().toString());
        parameter.put("finddetail",
edtfindingdetail.getText().toString());
        parameter.put("findimage",
Utility.bitmapToString(((BitmapDrawable)
imageView.getDrawable()).getBitmap()));

        // parameter.put("user_id",a);
        return parameter;

// return super.getParams();
    }
};
RequestQueue queue =
Volley.newRequestQueue(PostfindingActivity.this);
queue.add(request);

}
public void fromgallery(View view) {
    Intent photoPickerIntent = new
Intent(Intent.ACTION_PICK);
    photoPickerIntent.setType("image/*");
    startActivityForResult(photoPickerIntent,
SELECT_PICTURE); // method 1, get from gallery
    flaging = 0;

}
protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode,
data);
    if (flaging == 1) {
        if (requestCode == CAMERA_REQUEST &&
resultCode == Activity.RESULT_OK) {
            Bitmap photo = (Bitmap)
data.getExtras().get("data");

            imageView.setImageBitmap(photo);

```

```

    }
} // end of if

else {
    if (resultCode == RESULT_OK) {
        try {
            final Uri imageUri = data.getData();
            final InputStream imageStream =
getContentResolver().openInputStream(imageUri);
            final Bitmap selectedImage =
BitmapFactory.decodeStream(imageStream);

imageView.setImageBitmap(selectedImage);

        } catch (FileNotFoundException e) {
            e.printStackTrace();

Toast.makeText(PostfindingActivity.this, "Something went
wrong", Toast.LENGTH_LONG).show();
        }

    } else {
        Toast.makeText(PostfindingActivity.this,
"You haven't picked Image", Toast.LENGTH_LONG).show();
    }
} // end of else
}
}

```

Report activity

```

package com.fyp.ancientdiggers.Admin;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.PopupMenu;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

```

```

import com.fyp.ancientdiggers.R;
import om.fyp.ancientdiggers.SessionManager;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;
import java.util.Map;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
public class ReportActivity extends AppCompatActivity {

    SessionManager sessionManager;
    EditText edtreport;
    Button btnraddreport;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_report);

sessionManager=new SessionManager(ReportActivity.this);

        btnraddreport=(Button) findViewById(R.id.btnraddreport);

        edtreport=(EditText) findViewById(R.id.edtreport);

        btnraddreport.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {

                StringRequest request = new
StringRequest(Request.Method.POST,
"https://upwardsol.com/AccientTourGuied/Reportbyadmin.php
", new Response.Listener<String>() {
                    @Override
                    public void onResponse(String
response) {

```

```

//
Toast.makeText (MapsActivity.this, ""+response,
Toast.LENGTH_SHORT).show();
        try {
            JSONObject data=new
JSONObject (response);
                String
success=data.getString("success");

if (success.toString().equalsIgnoreCase("1")){

Toast.makeText (ReportActivity.this, "Complaint
SucessFull"+success, Toast.LENGTH_SHORT).show();

                }
            else
if (success.toString().equalsIgnoreCase("0")){

Toast.makeText (ReportActivity.this, "fill all"+success,
Toast.LENGTH_SHORT).show();
                }

            } catch (JSONException e) {
                e.printStackTrace();
            }

        }
    }, new Response.ErrorListener() {
        @Override
        public void
onErrorResponse (VolleyError error) {

Toast.makeText (ReportActivity.this, "volly error "+error,
Toast.LENGTH_SHORT).show();
                }
    })
    {
        @Override
        protected Map<String, String>
getParams() throws AuthFailureError {

            Map<String, String> parameter = new
HashMap<String, String>();

```

```

parameter.put("reportdetail",edtreport.getText().toString
()+"");

        return parameter;

    }
}

;
RequestQueue queue =
Volley.newRequestQueue(ReportActivity.this);
queue.add(request);

}
});
}
}

```

Search Encyclopedia activity

```

package com.fyp.ancientdiggers.localpeople;
import android.annotation.SuppressLint;
import android.content.Context;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.fyp.ancientdiggers.R;
import com.fyp.ancientdiggers.SessionManager;
import com.fyp.ancientdiggers.Utility;

```

```

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import androidx.appcompat.app.AppCompatActivity;
import de.hdodenhof.circleimageview.CircleImageView;

public class SearchEncyclopediaActivity extends
AppCompatActivity {

    public MyAdapter adapter1;

    private EditText etSearch;

    public ListView lvProducts;
    private ArrayList<DataClassEncyclopedia>
mProductArrayList = new ArrayList<>();

    SessionManager sessionManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_all_searchencyclopedia);
        sessionManager = new
SessionManager(SearchEncyclopediaActivity.this);

        initialize();
        GetAllforSearch();

        etSearch.addTextChangedListener(new TextWatcher()
{
            public void onTextChanged(CharSequence s, int
start, int before, int count) {

adapter1.getFilter().filter(s.toString().toLowerCase());
            }

            public void beforeTextChanged(CharSequence s,
int start, int count, int after) {

            }

            public void afterTextChanged(Editable s) {

```



```

        }
    });
}
public void searchorg(View view)
{
    sessionManager.setType("Organization");
    GetallforSearch();
}
public class MyAdapter extends BaseAdapter implements
Filterable {
    LayoutInflater inflater;
    /* access modifiers changed from: private */
    public ArrayList<DataClassEncyclopedia>
mDisplayedValues;
    /* access modifiers changed from: private */
    public ArrayList<DataClassEncyclopedia>
mOriginalValues;

    private class ViewHolder {
        LinearLayout llContainer;
        TextView txtname,txtaddress,txtcontact;

        CircleImageView tailorimage;
        Button btnchat,btnreport;

        private ViewHolder() {
        }
    }

    public MyAdapter(Context context,
ArrayList<DataClassEncyclopedia> mProductArrayList) {
        this.mOriginalValues = mProductArrayList;
        this.mDisplayedValues = mProductArrayList;
        this.inflater = LayoutInflater.from(context);
    }

    public int getCount() {
        return this.mDisplayedValues.size();
    }
    public Object getItem(int position) {
        return Integer.valueOf(position);
    }
    public long getItemId(int position) {
        return (long) position;
    }

    @SuppressWarnings("WrongViewCast")
    public View getView(final int position, View
convertView, ViewGroup parent) {
        ViewHolder holder;
        if (convertView == null) {
            holder = new ViewHolder();

```

```

        convertView =
this.inflater.inflate(R.layout.encyclopediarow, null);
        holder.llContainer = (LinearLayout)
convertView.findViewById(R.id.llContainer);
        holder.txtname = (TextView)
convertView.findViewById(R.id.txtname);
        holder.txtaddress = (TextView)
convertView.findViewById(R.id.txtaddress);
        holder.txtcontact = (TextView)
convertView.findViewById(R.id.txtcontact);

        holder.tailorimage = (CircleImageView)
convertView.findViewById(R.id.profile_image);
        holder.btnchat = (Button)
convertView.findViewById(R.id.btnchat);
        holder.btnreport = (Button)
convertView.findViewById(R.id.btnreport);
        convertView.setTag(holder);
    } else {
        holder = (ViewHolder)
convertView.getTag();
    }
    TextView txtname,txtaddres,txtcontact;
    CircleImageView tailorimage;
    Button btnchat,btnreport;
    txtcontact = holder.txtcontact;
    txtname = holder.txtname;
    txtaddres = holder.txtaddress;
    tailorimage = holder.tailorimage;
    btnchat = holder.btnchat;
    btnreport = holder.btnreport;

    txtname.setText("Name->
"+this.mDisplayedValues.get(position).getEncyclopediatitl
e());
    txtaddres.setText("Address->
"+this.mDisplayedValues.get(position).getEncyclopediadeta
il());
    txtcontact.setText("Date->
"+this.mDisplayedValues.get(position).getEncyclopediadate
());

    tailorimage.setImageBitmap(Utility.stringToBitmap(this.mD
isplayedValues.get(position).getEncyclopediaimage()));

    btnchat.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {

```

```

        /* Intent intent=new
Intent(SearchEncyclopediaActivity.this,
ChatActivity.class);

intent.putExtra("recieverid", Integer.parseInt (mDisplayedV
alues.get (position) .getId ());

        startActivity(intent);*/

    }
});
btnreport.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {

        /* Intent intent=new
Intent(SearchEncyclopediaActivity.this,
ComplaintActivity.class);

intent.putExtra("complainid",mDisplayedValues.get (positio
n) .getId ());

intent.putExtra("complainname",mDisplayedValues.get (posit
ion) .getName ());

intent.putExtra("category",mDisplayedValues.get (position)
.getType ());

        startActivity(intent);*/

    }
});

return convertView;
}

public Filter getFilter() {
    return new Filter() {

        public void publishResults(CharSequence
constraint, FilterResults results) {
            MyAdapter.this.mDisplayedValues =
(ArrayList) results.values;

MyAdapter.this.notifyDataSetChanged();
        }
    }
}

```

```

        /* access modifiers changed from:
protected */
        public FilterResults
performFiltering(CharSequence constraint) {
            FilterResults results = new
FilterResults();
            ArrayList<DataClassEncyclopedia>
FilteredArrList = new ArrayList<>();
            if (MyAdapter.this.mOriginalValues ==
null) {
                MyAdapter myAdapter =
MyAdapter.this;
                myAdapter.mOriginalValues = new
ArrayList(myAdapter.mDisplayedValues);
            }
            if (constraint == null ||
constraint.length() == 0) {
                results.count =
MyAdapter.this.mOriginalValues.size();
                results.values =
MyAdapter.this.mOriginalValues;
            } else {
                CharSequence constraint2 =
constraint.toString().toLowerCase();
                for (int i = 0; i <
MyAdapter.this.mOriginalValues.size(); i++) {

                    if (((DataClassEncyclopedia)
MyAdapter.this.mOriginalValues.get(i)).getEncyclopediatit
le().toLowerCase().startsWith(constraint2.toString())
|| ((DataClassEncyclopedia)
MyAdapter.this.mOriginalValues.get(i)).getEncyclopediadet
ail().toLowerCase().startsWith(constraint2.toString())) {
                        FilteredArrList.add(new
DataClassEncyclopedia(MyAdapter.this.mOriginalValues.get(
i).getEncyclopediatitle(),
MyAdapter.this.mOriginalValues.get(i).getEncyclopediadeta
il()+"",
MyAdapter.this.mOriginalValues.get(i).getEncyclopediainmag
e()+"",
MyAdapter.this.mOriginalValues.get(i).getEncyclopediadate
()+""));
                    }
                }
                results.count =
FilteredArrList.size();
                results.values = FilteredArrList;
            }
            return results;
        }
    }
}

```

```

        };
    }
}
private void initialize() {
    etSearch = (EditText)
findViewById(R.id.inputSearch);
    lvProducts = (ListView)
findViewById(R.id.listviewall);
}
@Override
protected void onResume() {

    super.onResume();
}
public void GetAllforSearch(){
    StringRequest request = new
StringRequest(Request.Method.GET,
"https://upwardsol.com/AccientTourGuied/getallencyclopedia
a.php", new Response.Listener<String>() {

        public void onResponse(String response) {
            try {
                JSONArray arrrr = new
JSONObject(response).getJSONArray("result");

                String
encyclopediatitle,encyclopediadetail,encyclopediainage,en
cyclopediadate;

                if (arrrr != null) {
                    mProductArrayList = new
ArrayList<DataClassEncyclopedia>();
                    for (int i = 0; i <
arrrr.length(); i++) {
                        JSONObject json =
arrrr.getJSONObject(i);

encyclopediatitle=json.getString("encyclopediatitle");
encyclopediadetail=json.getString("encyclopediadetail");
encyclopediainage=json.getString("encyclopediainage");
encyclopediadate=json.getString("encyclopediadate");

                            mProductArrayList.add(new
DataClassEncyclopedia(encyclopediatitle,encyclopediadetai
l,encyclopediainage,encyclopediadate));
                        }
                    }
                }
            }
        }
    }
}

```

```

        adapter1 = new
MyAdapter(SearchEncyclopediaActivity.this,mProductArrayLis
t);
        lvProducts.setAdapter(adapter1);
    }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    }, new Response.ErrorListener() {
error) {
        public void onErrorResponse(VolleyError
    }
    }) {
        /* access modifiers changed from: protected
*/
        public Map<String, String> getParams() throws
AuthFailureError {
            Map<String, String> parameter = new
HashMap<String, String>();
            return parameter;
        }
    };
    RequestQueue queue =
Volley.newRequestQueue(SearchEncyclopediaActivity.this);
    queue.add(request);
}
}
}

```

Chapter 5

System Testing

Testing

Software testing is an empirical technical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. System testing is an essential step for the development of the reliable and error free system. Testing is the process executing a program with the explicit intention of finding errors i.e. making the program fail and test cases are devised with the purpose in mind. A test case is a set of data items that the system processes as normal input. A successful test is one that has a high probability of finding an as-yet undiscovered error. Testing involves exercising the program using data like the real processed by the program. Testing may be carried out during the implementation phase, to verify if the software behaves as intended by its designer, or after the completion of its implementation phase. There are many approaches to software testing.

- Unit testing
- Interface testing
- Integration testing

- Module testing

5.1 Unit Testing

Each and every module of the system was tested at individual component level to ensure that system operates correctly. Testing of these modules involved various steps. Validation checks were confirmed on each item after the data was entered. Functionality of each button was successfully tested.

5.2 Interface Testing

Interface testing ensures that proper links are operational and everything is displayed according to the requirement of the system. Interface testing involved many steps: Fonts size and font name were checked against the guidelines laid by the University. It was checked that every meaningful word in prompt/any alert message displayed started with a capital letter.

5.3 Module testing

A module is a collection of dependent components. In module testing, related units are integrated in a module and tested for their proper functionality together. Various modules were tested, by passing data/signal between them.

5.4 Integration Testing

All the components module of the system was put to integrated testing and were found working successfully.

5.5 Evaluation

Evaluation of the system means to examine what goals are achieved by the system and what are the weaknesses and the deficiencies left behind. Evaluation is the final step in the development by any system. This phase is for the purpose of reviewing whether objectives and functional requirements of the Univ. are fulfilled are not.

5.6 Achievements

- Software provides the password protected GUI to facilitate the user to use the system.
- In all modules of the system, data is properly added, delete and update.
- All verification and validation checks work properly.
- System is satisfying almost all of its major specifications and functionality and it can be concluded that system is satisfying its more than 90% scope.

Table 0.1 TC-01 (User Authentication)

Test Case ID: TC-GPD-001	Test Item: User Authentication
Author: Umair Akram	Doc-Date: 15 Nov 2021
Test Type: Unit Testing.	Test Case Name: Login
Component: User authentication facility	Release Version: 1.0
Test Case Description: The test case will determine the current functionality of user authentication.	
Pre-Condition	Post Condition
User should have his own email and password.	System successfully authorized the user to use svstem
Input Data/Event	Expected output Data
Enter username and password. Test Case Result	User should be login successfully.
Actual Input Data/Event	Actual Output Data/Event
Enter “umairakram3636@gmail.com” as username and “123456” as password.Click on “Sign in” button.	Successfully login.
Enter “hamayunk1997@gmail.com” as username and “1234” as password.	Successfully login.
Enter “admin” as username and “12345” as password. Click on “Sign in” button.	Login Failed.
Enter “admin@gmail.com” as username and “abide” as password.Click on “Sign in” button.	Login Failed.
Test Case Result	
Test is performed successfully.	

Table 0.2 TC-02 (Block User)

Test Case ID: TC-GPD-003	Test Item: Manage User
Author: umair akram	Doc-Date: 15 Nov 2021
Test Type: Unit Testing.	Test Case Name: Block User
Component: Add User Facility	Release Version: 1.0
Test Case Description: The test case will determine the current functionality of delete user.	
Pre-Condition	Post Condition
Admin should be login successfully.	User Blocked successfully.
Input Data/Event	Expected output Data
Select any user to Blocked. Click on the “Block” button.	User should be deleted successfully.
Actual Input Data/Event	Actual Output Data/Event
Select the user. Click on “Delete” button.	System Block the user successfully
Test Case Result	
Test is performed successfully.	

Table 0.3 TC-03 (Update profile)

Test Case ID: TC-GPD-004	Test Item: Update profile
Author: Hamayun Rashid	Doc-Date: 20 Nov 2021
Test Type: Unit Testing.	Test Case Name: View/Update Profile
Component: View/Update Facility	Release Version: 1.0
Test Case Description: The test case will determine the current functionality of View/Update Profile.	
Pre-Condition	Post Condition
User should have sign in successfully.	Profile is successfully updated.
Input Data/Event	Expected output Data
User enters the information to view/update profile.	User should be successfully View/Update Profile.
Actual Input Data/Event	Actual Output Data/Event
Click on "Profile" button. Click on field to update profile.	Successfully View Profile
Test Case Result	
Test is performed successfully.	

Chapter 6

Screen Shots

Pak Ancient Diggers

Explore the beauty of world

We all need a little inspiration from time to time. Let these travel quotes motivate you to take that trip you've always dreamt of.



bigstock.com · 322489f

Ancient diggers



Figure 6.1



Figure 6.2

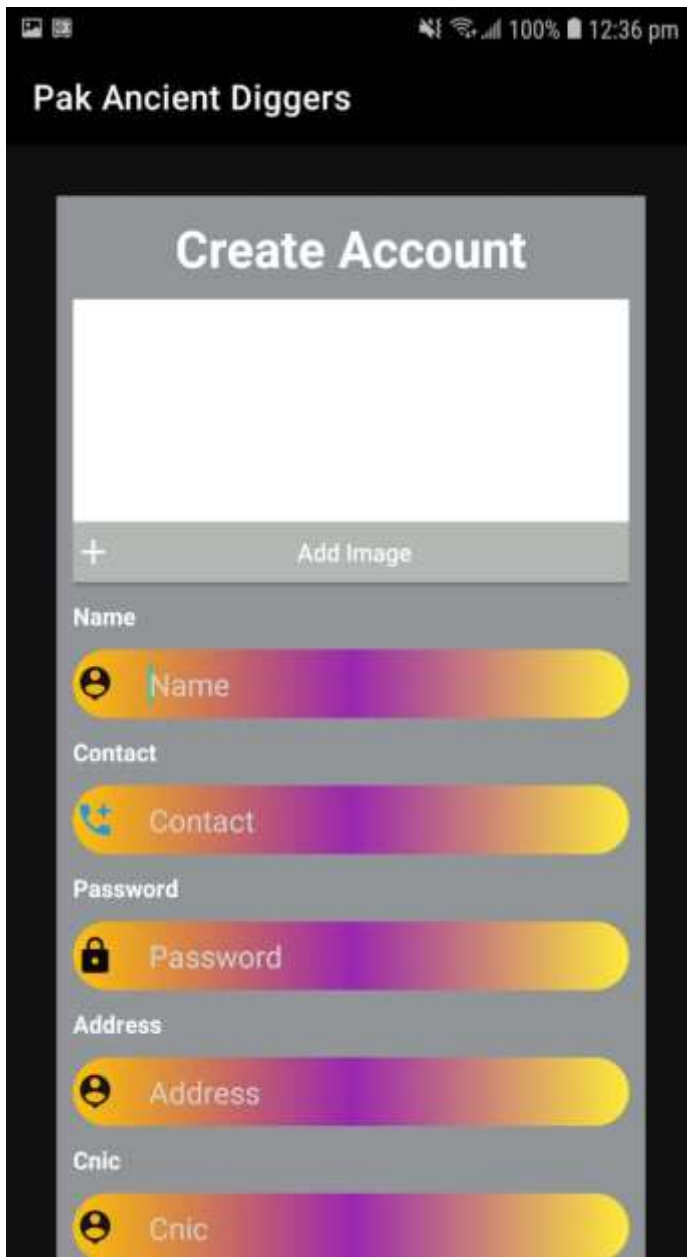


Figure 6.3



Figure 6. 4

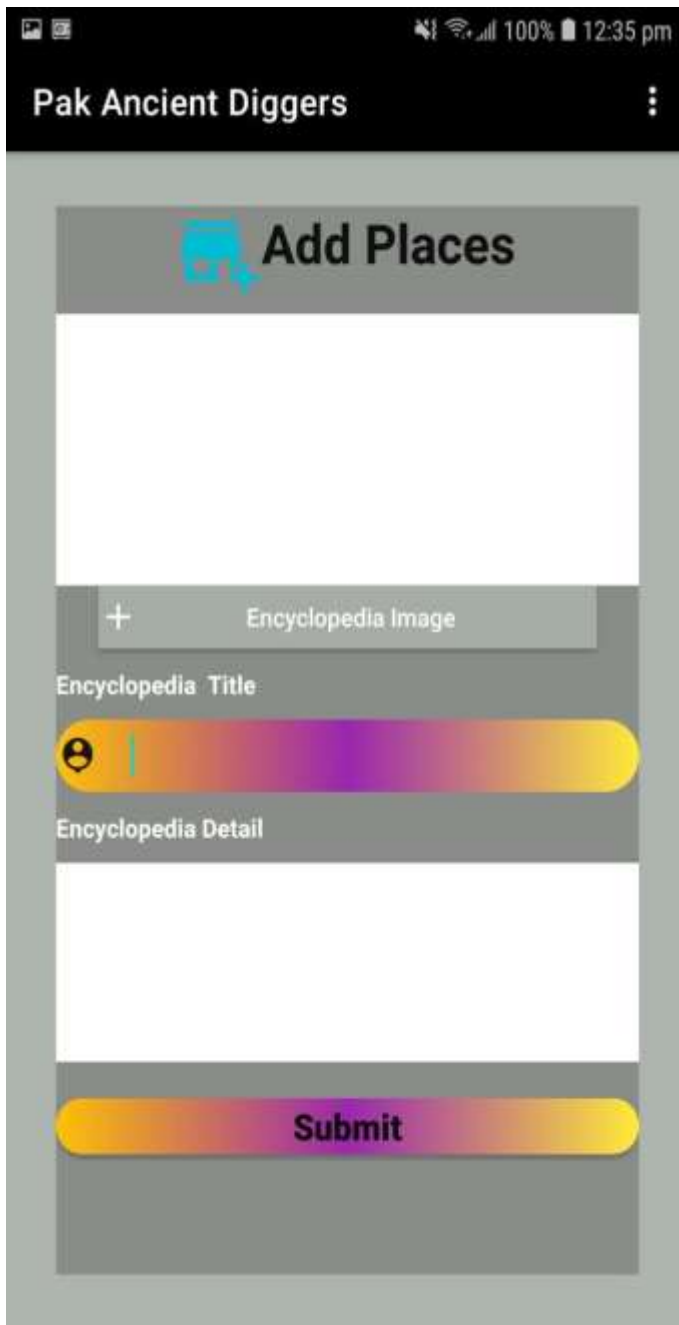


Figure 6. 5

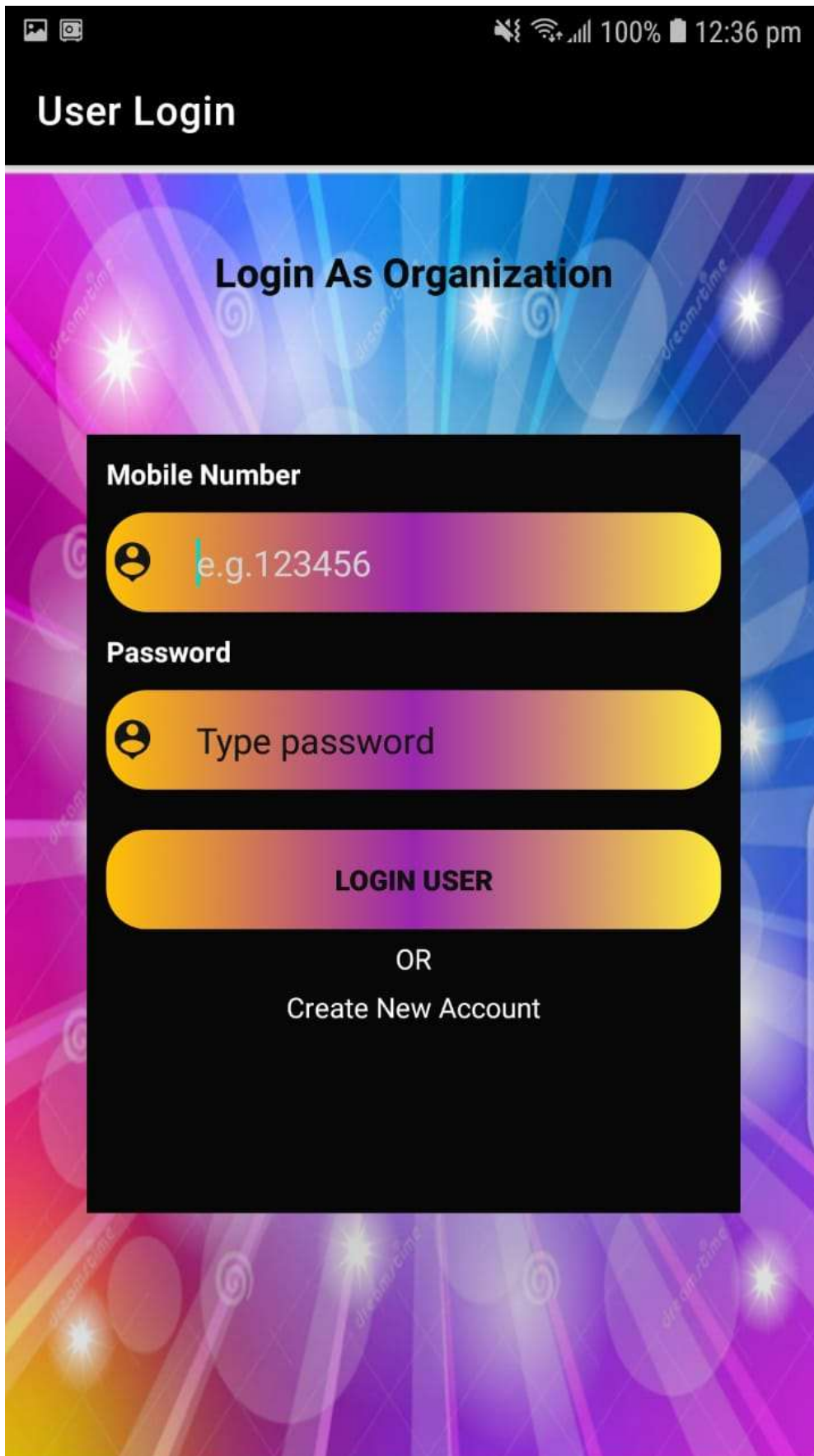


Figure 6. 6

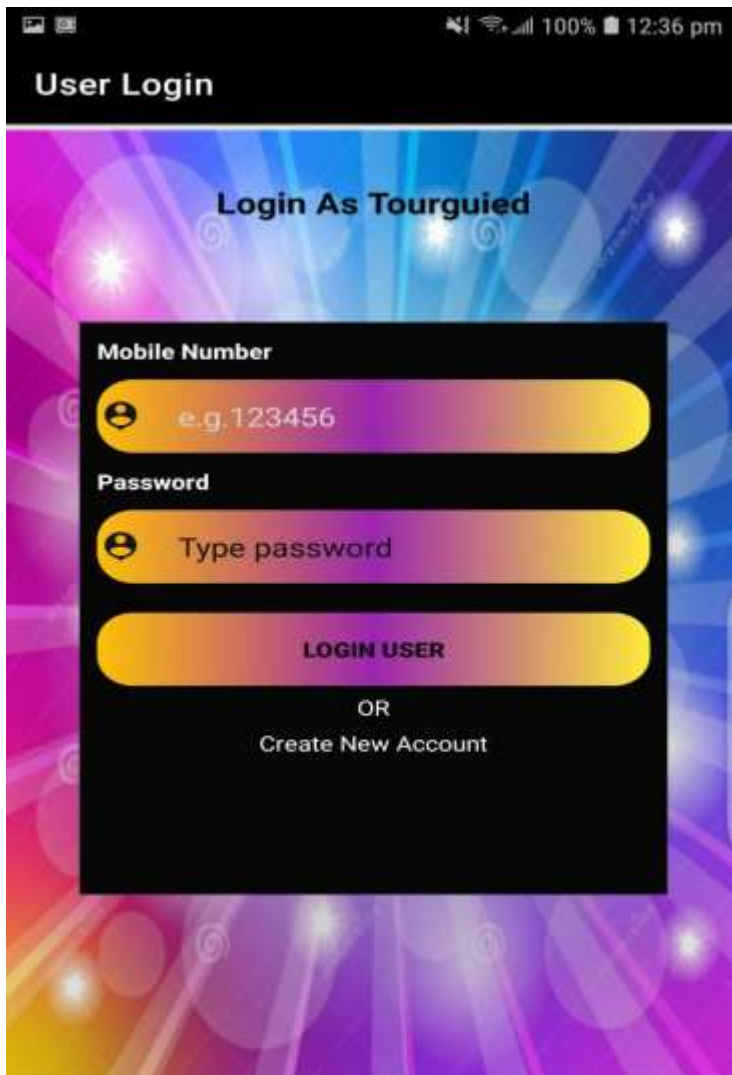


Figure 6. 1

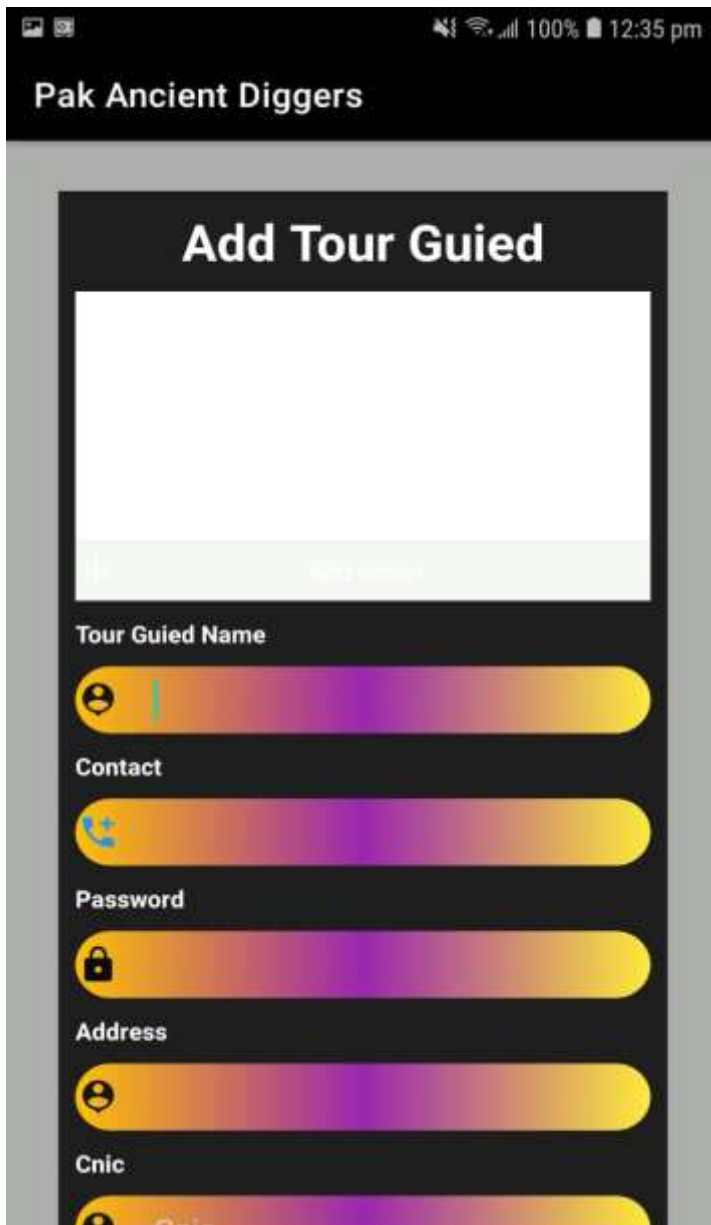


Figure 6. 2

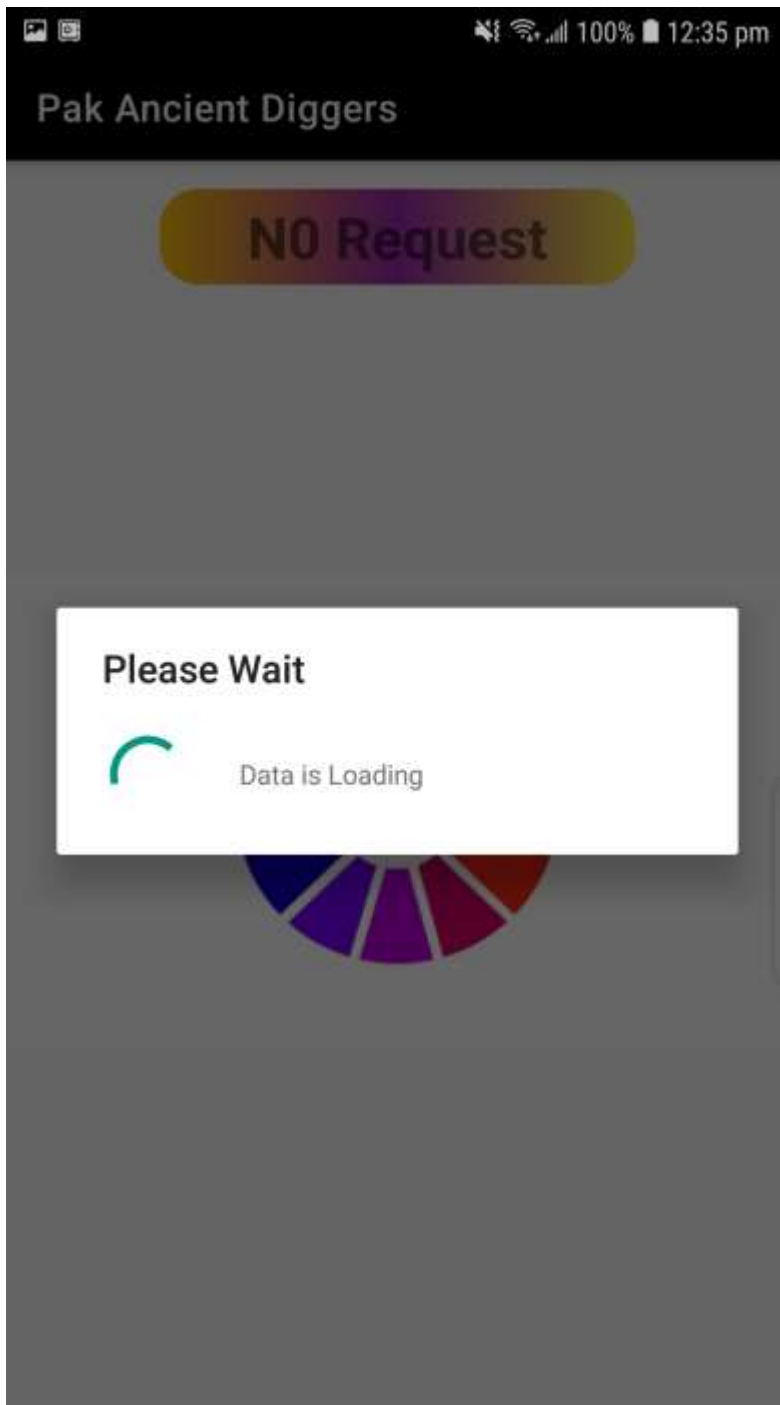


Figure 6. 3

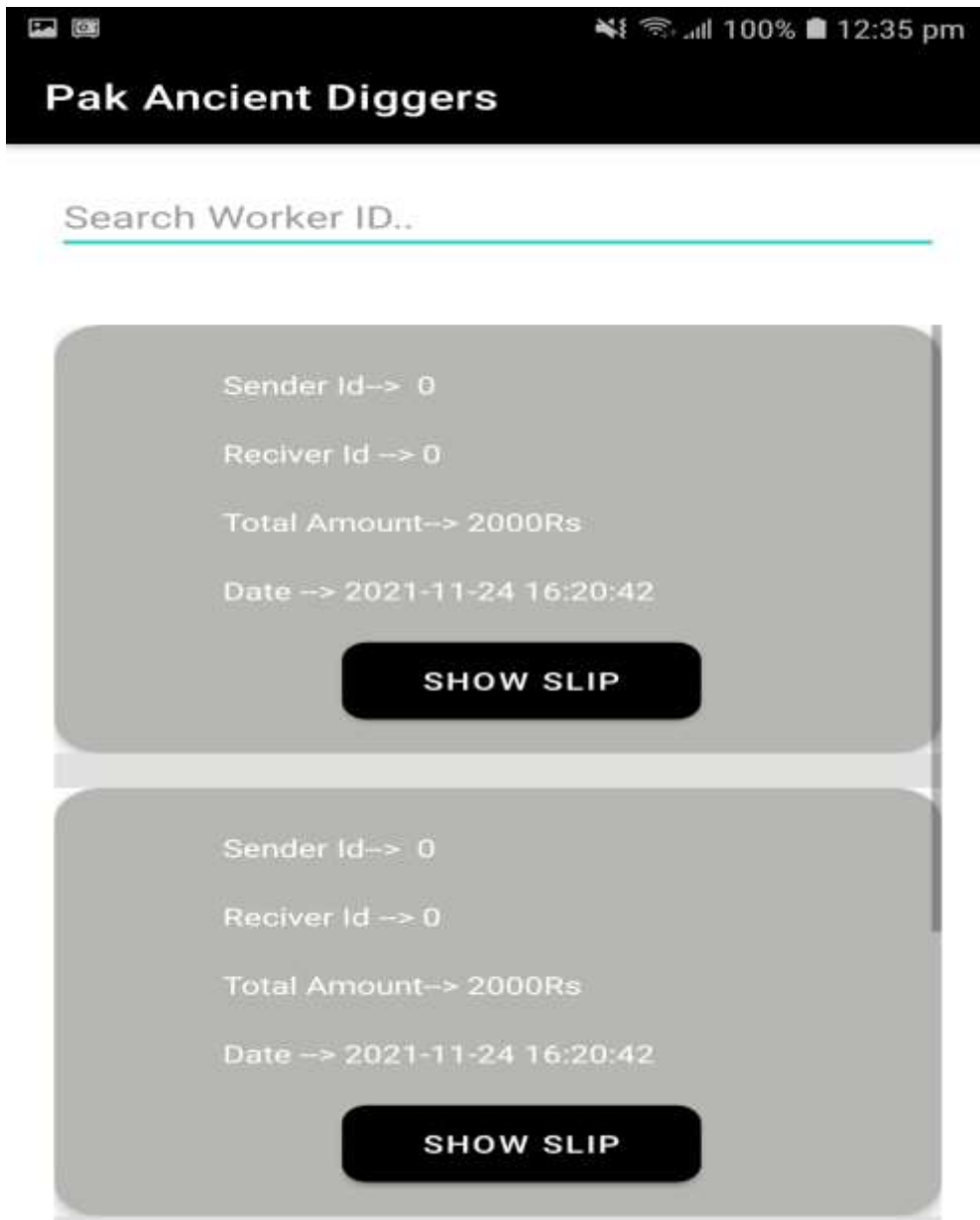


Figure 6. 4

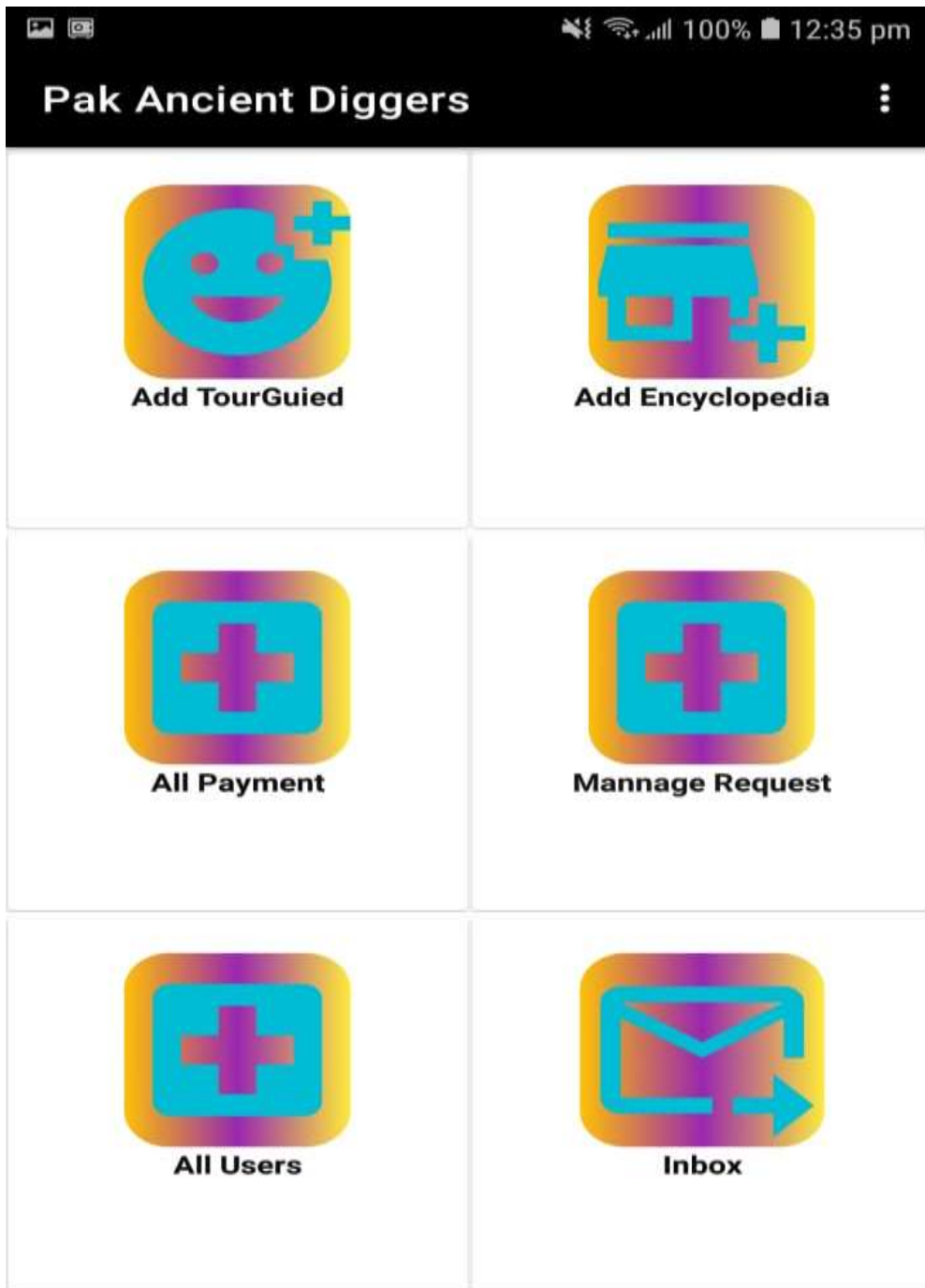


Figure 6.10