NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA

KNOWLEDGE * CHARACTER * UNITY

MINI PROJECT REPORT

on

Plan With Ease: An Android Travel Planning APP

Submitted in partial fulfilment of the requirement for the award of Degree of

Bachelor of Engineering

in

Information Science and Engineering

Submitted by:

Kuldeep Kumar Raju k	1NT20IS083
Narendra Babu M	1NT20IS098

Under the Guidance of

Mr. Prashanth B S

Assistant Professors, Dept. of ISE, NMIT

NITTE EDUCATION TRUST

Department of Information Science and Engineering

(Accredited by NBA Tier-1)

2022-2023

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

Department of Information Science and Engineering

(Accredited by NBA Tier-1)

CERTIFICATE

This is to certify that the Project Report on "**Plan With Ease: An Android Travel Planning APP"** is an authentic work carried out by **Kuldeep Kumar Raju K(1NT20IS083), Narendra Babu M(1NT20IS098)** Bonafede students of Nitte Meenakshi Institute of Technology, Bangalore in partial fulfilment for the award of the degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during the academic year 2022-2023**.** It is certified that all corrections and suggestions indicated during the internal assessment has been incorporated in the report.

Signature:

_____

Mr. Prashanth B S.

Assistant Professor, Dept. ISE,
NMIT Bangalore

Abstract

"Plan with Ease" is an innovative Android application designed to revolutionize travel planning by prioritizing the often-overlooked task of efficient packing. With predefined categories featuring default items, users can easily tailor their packing lists to specific needs, adding personal items or creating custom categories. The intuitive user interface ensures a seamless experience, even for those new to travel planning. The inclusion of a map feature, integrated with Google Maps, provides users with route planning capabilities from their starting location to the destination. A dedicated "My Selections" page allows users to monitor and manage their chosen items across categories, promoting an organized approach. The app's flexibility is highlighted through individual item deletion and a reset option for a fresh start. "Plan with Ease" aims to redefine travel preparation, offering users a user-centric, organized, and stress-free solution for a more enjoyable journey.

Acknowledgement

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success. I express my sincere gratitude to our Principal **Dr. H. C. Nagaraj**, Nitte Meenakshi Institute of Technology for providing facilities.

We wish to thank our HoD**, Dr. Mohan S. G.** for the excellent environment created to further educational growth in our college. We also thank him for the invaluable guidance provided which has helped in the creation of a better project.

I hereby like to thank our ***Mr. Prashanth B S, Assistant Professor*** Department of Information Science & Engineering on their periodic inspection, time to time evaluation of the project and help to bring the project to the present form.

Thanks to our Departmental Project coordinators. We also thank all our friends, teaching and non-teaching staff at NMIT, Bangalore, for all the direct and indirect help provided in the completion of the project.

Table of Contents

Abstract

Acknowledgement

Sl.no	Chapter Title	Page Number
1	Introduction	1-2
2	Literature Review	3-4
3	Problem Statement	5
4	System Design	6-9
5	Implementation	10-15
6	Results and Snapshots	16-17
7	Conclusion	18

References

List of Figures

Sl.No.	Figure Name	Page No.
1	Android Architecture	1
2	Architectural Design	6
3	Software Architecture Block	7
4	Flowchart	8
5	Use Case Diagram	9
6	Flash screen	16
7	Categories	16
8	Items	16
9	Packing Items	16
10	Deleting Items	16
11	Selected Items	17
12	Custom List	17
13	Location Page	17
14	Maps	17

Chapter-1

INTRODUCTION

"Plan with Ease" is a user-friendly Android application designed to simplify and streamline the travel planning process. Aimed at assisting traveller's in organizing their trips efficiently, the app focuses on the critical aspect of packing. By providing a comprehensive platform for creating packing lists and managing travel essentials, "Plan with Ease" aims to enhance the overall travel experience for users.

1.1 Android Architecture

The Android architecture is a robust framework designed to support diverse device needs. At its core is the open-source Linux Kernel, offering essential operating system functions to smartphones. This kernel interacts with various components, including the Dalvik Virtual Machine (DVM), responsible for executing Android applications. The Android architecture comprises several key components [1]

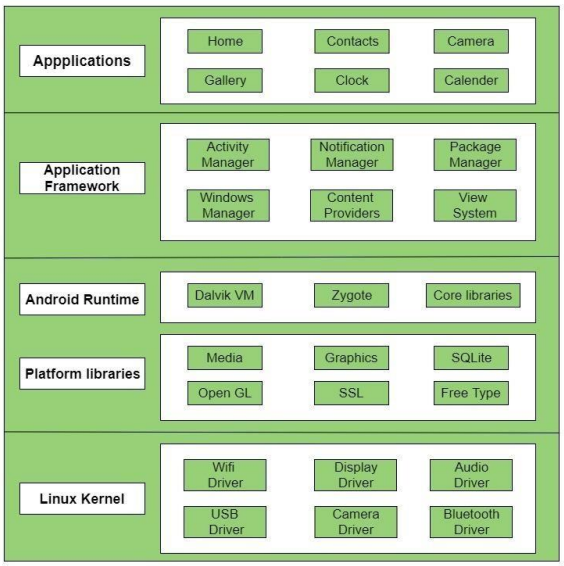


Fig1: Android architecture

1. Applications:

Represents the top layer where user-facing applications reside, providing functionalities and features tailored for end-users.

2. Application Framework

Sits between the applications and the lower-level components, offering a set of services and APIs that facilitate app development. This layer simplifies the development process by providing reusable code and components.

3. Android Runtime

Hosts the Dalvik Virtual Machine (DVM) or, in more recent versions, the Android Runtime (ART). DVM executes Android applications, converting bytecode into machine code at runtime. ART, introduced in later Android versions, pre-compiles the bytecode into native machine code for improved performance.

4. Platform Libraries

Comprises a set of libraries written in C/C++ that provide essential functionalities to developers. These libraries cover various aspects such as graphics rendering, database management, networking, and more.

5. Linux Kernel

Serves as the foundational layer, offering core operating system services such as process management, memory management, device drivers, and security. The kernel is responsible for handling communication between hardware and software components.

Chapter-2

LITERATURE REVIEW AND OBJECTIVES

2.1 Literature Review

Many existing travel planning applications tend to concentrate on managing itineraries, accommodations, and transportation, leaving a gap in addressing the specific requirements of efficient packing—an essential aspect of travel. The inclusion of map features in travel apps is known to improve overall usability, providing users with comprehensive solutions that not only assist in planning but also seamlessly navigate their journeys. Personalized experiences play a crucial role in engaging users, with features like customizable lists and the option to filter user-added items catering to individual preferences, thereby augmenting the app's overall value. Research indicates a positive correlation between efficient packing and overall travel satisfaction, suggesting that an application facilitating the creation of well-organized packing lists has the potential to significantly enhance users' travel experiences [2].

2.1 Objectives

1. Simplify Travel Planning:

- The primary objective is to simplify the travel planning process by focusing on efficient packing, ensuring users have a comprehensive and user-friendly tool to organize their essentials.

2. User-Centric Design:

- Prioritize a user-centric design to enhance the overall user experience. The app should be intuitive, visually appealing, and easy to navigate, catering to users of varying technological proficiency.

3. Comprehensive Packing Lists:

- Provide users with predefined packing categories containing default items. Allow customization by enabling users to add their items and create custom lists tailored to their unique travel needs.

4. Map Integration:

- Integrate a map feature to assist users in planning their routes from the starting location to the destination. This feature should be seamless, requiring minimal input from the user.

5. Search and Filtering:

- Implement a robust search functionality within the packing lists, enabling users to quickly locate specific items. Additionally, provide filters, such as items added by the user, to enhance organization.

6. My Selections Page:

- Create a dedicated page where users can review and manage all selected items from different categories. This consolidated view helps users monitor their packing progress effectively.

7. Reset to Default:

- Offer users the option to reset their packing lists to default settings. This feature ensures flexibility for users who want to start afresh while still retaining the ability to customize.

8. Personalization:

- Enable users to personalize their packing experience by adding their items and creating custom categories. This feature aligns with the trend of providing tailored experiences in mobile applications.

Summary: This chapter talks about the gap in current travel apps regarding efficient packing, while emphasizing the importance of personalized features and project's objectives to create an Android app addressing this gap, providing intuitive packing solutions and a seamless travel experience.

Chapter-3

PROBLEM STATEMENT

Many traveller’s face challenges in efficiently organizing and packing for their trips, often resulting in forgotten essentials and disorganized luggage. Existing travel planning apps primarily focus on itinerary and accommodation, leaving a gap in addressing the critical aspect of streamlined packing. This deficiency leads to a need for a user-friendly mobile application that prioritizes packing, providing a comprehensive solution for creating, managing, and customizing packing lists. "Plan with Ease" aims to bridge this gap by offering an intuitive platform that simplifies the packing process, ensuring traveller’s have a well-organized and personalized checklist for a stress-free travel experience.

Chapter-4

SYSTEM DESIGN

System design talks about the process of designing the architecture, components, and interfaces for a system so that it meets the end-user requirements. Quality system design is essential in developing functional and lasting applications.

4.1 Architecture Design

A system's architecture outlines its main parts, their connections (structures), and how they work together. There are several contributing variables to software architecture and design, including business strategy, quality attributes, human dynamics, design, and IT environment.

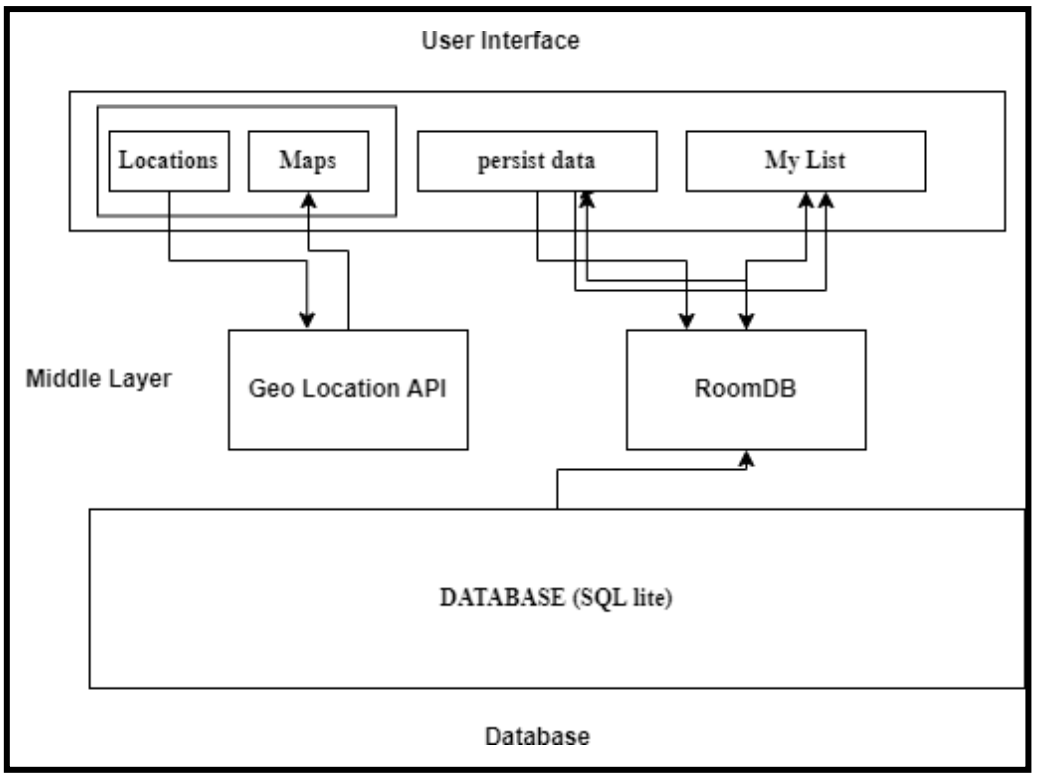


Fig2: architectural Design

This architecture describes a three-layered structure for a location-based Android application with components responsible for user interface, data handling, and database storage. The user interface layer is composed of three key components: "Locations," providing a view of different places; "Maps," facilitating map-related functionalities; and "My List," allowing users to save and manage various locations. The middle layer consists of two components: the "Geo Location API," responsible for acquiring location data from the device, and "RoomDB," handling the persistence of this data. The database layer contains a single component, the "DATABASE" implemented using SQL Lite, which stores the application data in a relational database. The architecture illustrates the seamless flow of data between

these components, portraying a clear structure for the application's functionalities and data management [3].

4.2 Software Architecture Block

A visual depiction that depicts the actual physical implementation of a software system's components is called an architecture diagram. It displays the relationships, constraints, and boundaries between each piece as well as the overall structure of the software system.

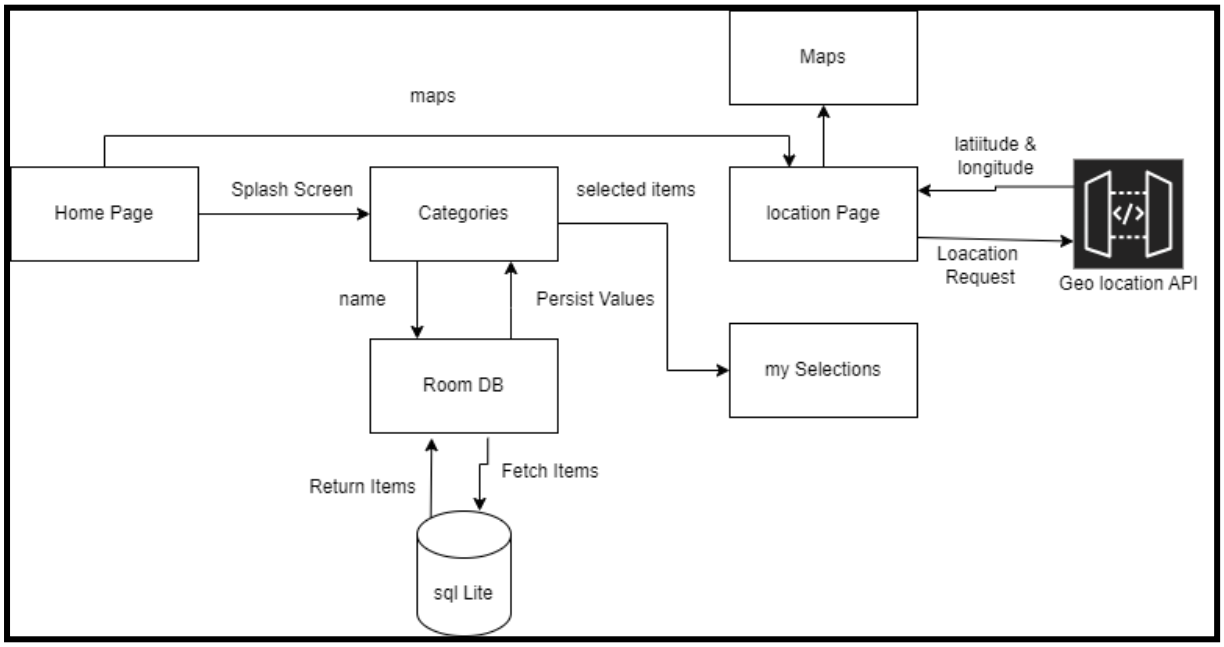


Fig3: Software Architecture Block

The figure shows sequential steps involved in various processes within the application, emphasizing the selection of items, persistence of values, fetching of items, and the utilization of the Geo Location API. The depicted components include the "Home Page," serving as the main interface, "Categories" for item selection, "Location Page" for location-related functionalities, "Maps" for map-related features, and "My Selections" for managing selected items. Additionally, the flowchart incorporates the "Room DB" and "SQL Lite" components, representing the layers responsible for data persistence and storage. The use of rectangular and square boxes with textual content inside, connected by arrows, illustrates the logical sequence of these processes, providing a visual representation of the application's functionality and data flow. This structured flowchart serves as a valuable tool for comprehending the interconnected steps and components involved in the seamless operation of the application. [4]

4.3 Flowchart

An algorithm is graphically represented by a flowchart. It is frequently used by programmers as a technique for planning programs to address issues. It uses interconnected symbols to represent the movement of information and processing. "Flowcharting" is the process of creating a flowchart for an algorithm.

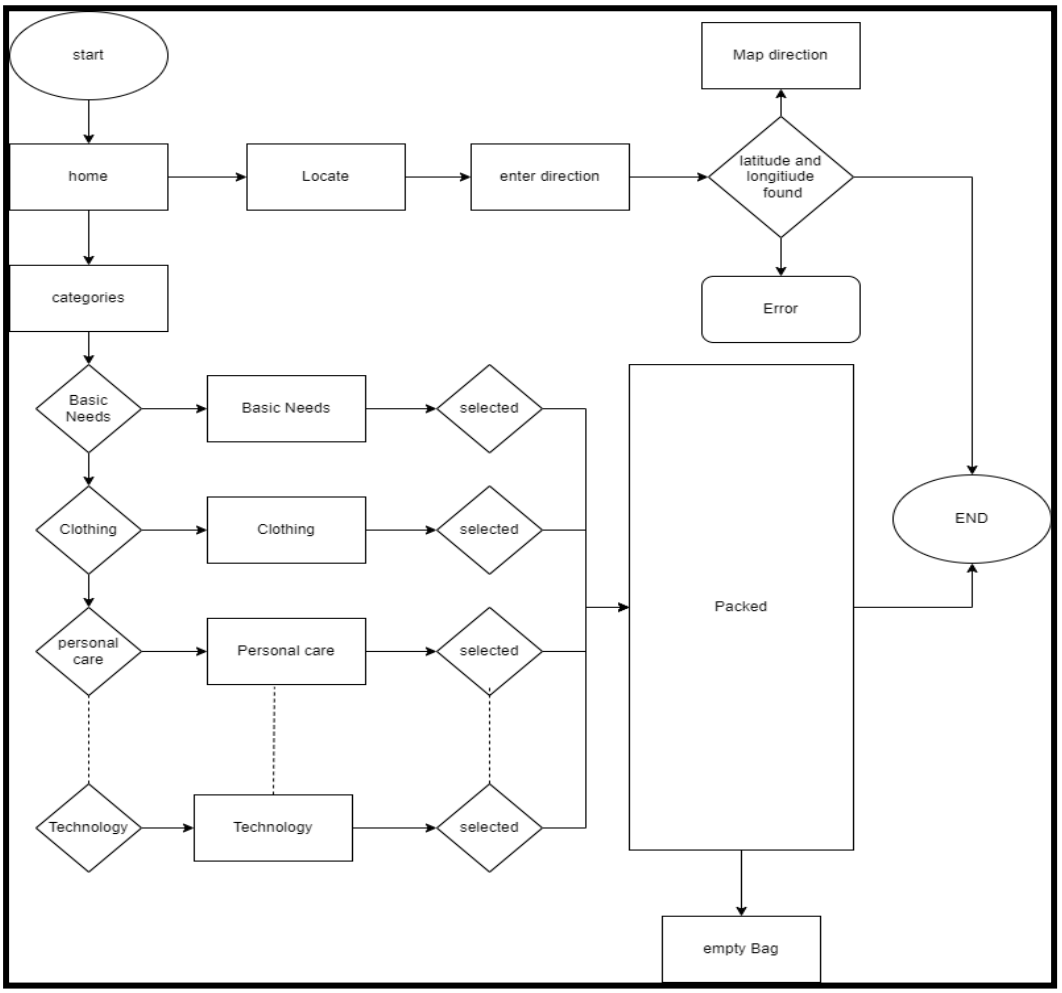


Fig4: Flowchat

The "Plan with Ease" app unfolds with a brief flash screen upon initiation, swiftly transitioning to the Home Page, the application's central hub. Here, users encounter various pre-defined categories encompassing default items essential for travel. The flexibility to personalize their packing experience is offered through the addition of user-specific items or the creation of custom lists. Upon selecting a category, users can view both default and user-added items, confirming their packing progress through a toast message. The consolidated "My Selections" page displays all chosen items across categories, facilitating quick reference. Users can further manage their selections, deleting items individually or resetting to default. A tailored

experience is provided through an "Added by User" option, showcasing only user-contributed items. Additionally, a search functionality streamlines the process of locating specific items. The app integrates a map feature, accessed through a dedicated button, initiating a seamless experience for entering starting and destination locations, ultimately providing directions through Google Maps. This systematic flow ensures users have a comprehensive, personalized, and organized approach to travel planning [2].

4.4 Use case Diagram

The scope and high-level functions of a system are described in use-case diagrams. The interactions between the system and its actors are also depicted in these diagrams. Utilize-case diagrams show what the system does and how the actors use it, but they do not show how the system works within.

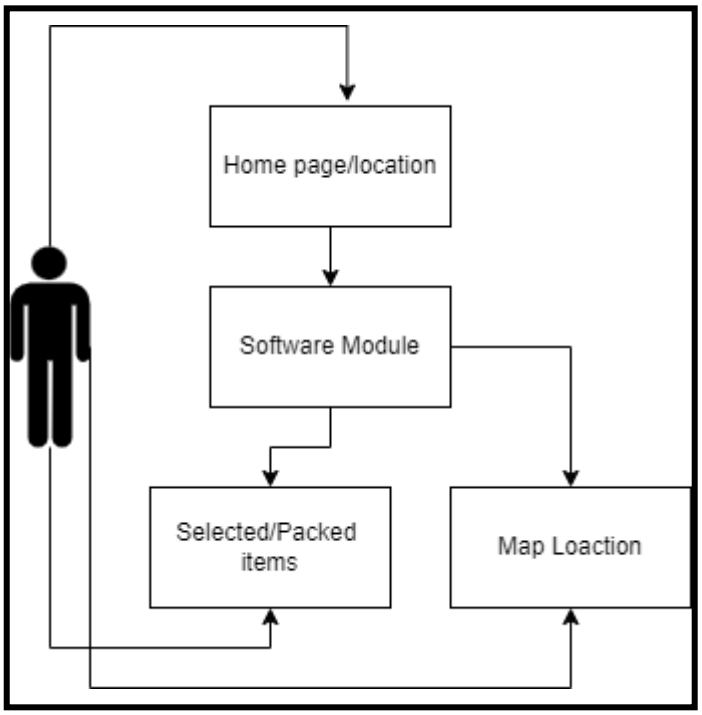


Fig5: use case Diagram

Chapter-5

IMPLEMENTATION

5.1 Important Code Snapshots

5.1.1 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/background3"
    tools:context=".MainActivity">

   <TextView
       android:layout_width="match_parent"
       android:layout_height="80dp"
       android:background="@drawable/wave2"
       android:id="@+id/title_view"
       android:textAlignment="center"
       android:text="Plan With ease"
       android:layout_centerHorizontal="true"
       android:textStyle="bold"
       android:textColor="@color/white"
       android:textSize="35sp"/>

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/recyclerview"
        android:layout_below="@+id/title_view"
        />
```

5.1.2 Mainactivity.java

```
package com.example.travelapp;

import ...

4 usages
public class MainActivity extends AppCompatActivity {

    3 usages
    RecyclerView recyclerView;
    14 usages
    List<String> titles;
    14 usages
    List<Integer> images;
    2 usages
    Adapter adapter;
    5 usages
    RoomDB database;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        getSupportActionBar().hide();

        recyclerView = findViewById(R.id.recyclerview);

        addAllTitles();
        addAllImages();
        persistAppdata();
        database = RoomDB.getInstance( context: this);
        System.out.println("-------->"+database.mainDao().getALlSelected( checked: false).get(0).getItemname());
```

```
    GridLayoutManager gridLayoutManager = new GridLayoutManager( context: this, spanCount: 2,RecyclerView.VERTICAL, reverseLayout: false);

    recyclerView.setLayoutManager(gridLayoutManager);
    recyclerView.setAdapter(adapter);

}

1 usage
private static final int TIME_INTERVAL = 2000;
2 usages
private long mBackPressed;

@Override
public void onBackPressed() {
   if(mBackPressed+TIME_INTERVAL > System.currentTimeMillis()){
       super.onBackPressed();
       return;
   }
   else{
       Toast.makeText( context: this, text: "Double Click to exit",Toast.LENGTH_SHORT).show();
   }
    mBackPressed= System.currentTimeMillis();
}
```

```
private void  addAllTitles(){
    titles = new ArrayList<>();
    titles.add(myconstants.BASIC_NEEDS_CAMEL_CASE);
    titles.add(myconstants.CLOTHING_CAMEL_CASE);
    titles.add(myconstants.PERSONAL_CARE_CAMEL_CASE);
    titles.add(myconstants.BABY_NEEDS_CAMEL_CASE);
    titles.add(myconstants.HEALTH_CAMEL_CASE);
    titles.add(myconstants.TECHNOLOGY_CAMEL_CASE);
    titles.add(myconstants.FOOD_CAMEL_CASE);
    titles.add(myconstants.BEACH_SUPPLIES_CAMEL_CASE);
    titles.add(myconstants.CAR_SUPPLIES_CAMEL_CASE);
    titles.add(myconstants.NEEDS_CAMEL_CASE);
    titles.add(myconstants.MY_LIST_CAMEL_CASE);
    titles.add(myconstants.MY_SELECTIONS_CAMEL_CASE);
}

1 usage
private void addAllImages(){
    images = new ArrayList<>();
    images.add(R.drawable.p1);
    images.add(R.drawable.p2);
    images.add(R.drawable.p3);
    images.add(R.drawable.p4);
    images.add(R.drawable.p5);
    images.add(R.drawable.p6);
    images.add(R.drawable.p7);
    images.add(R.drawable.p8);
    images.add(R.drawable.p9);
    images.add(R.drawable.p10);
    images.add(R.drawable.p11);
    images.add(R.drawable.p12);
```

5.1.3 Adapter.java

```
4 usages
public class Adapter extends RecyclerView.Adapter<Adapter.myViewHolder> {

    5 usages
    List<String> titles;
    2 usages
    List<Integer> images;
    2 usages
    LayoutInflater inflater;
    1 usage
    Activity activity;

    1 usage
    public Adapter(Context context,List<String> titles, List<Integer> images, Activity activity) {
        this.titles = titles;
        this.images = images;
        this.activity = activity;
        this.inflater = LayoutInflater.from(context);
    }

    @NonNull
    @Override
    public myViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = inflater.inflate(R.layout.main_item,parent, attachToRoot: false);
        return new myViewHolder(view);
    }
```

```
@Override
public void onBindViewHolder(@NonNull myViewHolder holder, @SuppressLint("RecyclerView") int position) {
    holder.title.setText(titles.get(position));
    holder.img.setImageResource(images.get(position));
    holder.linearLayout.setAlpha(0.8F);


    holder.linearLayout.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            //Toast.makeText(activity,"Clicked on card",Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(view.getContext(), checkList.class);
            intent.putExtra(myconstants.HEADER_SMALL,titles.get(position));
            if(myconstants.MY_SELECTIONS.equals(titles.get(position))){
                intent.putExtra(myconstants.SHOW_SMALL,myconstants.FALSE_STRING);
            }else {
                intent.putExtra(myconstants.SHOW_SMALL,myconstants.TRUE_STRING);
            }
            view.getContext().startActivity(intent);
        }
    });
}
```

```java
@Override
public int getItemCount() { return titles.size(); }

4 usages
public class myViewHolder extends RecyclerView.ViewHolder{
    2 usages
    TextView title;
    2 usages
    ImageView img;
    3 usages
    LinearLayout linearLayout;
    1 usage
    public myViewHolder(@NonNull View itemView) {
        super(itemView);
        title = itemView.findViewById(R.id.title);
        img = itemView.findViewById(R.id.img);
        linearLayout = itemView.findViewById(R.id.LinearLayout);

    }
}
```

5.1.4 activity_maps.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".maps"
    android:orientation="vertical"
    android:background="@drawable/background3"
    android:padding="20dp">


    <EditText
        android:id="@+id/start"
        android:layout_width="311dp"
        android:layout_height="55dp"
        android:layout_marginStart="4dp"
        android:layout_marginTop="24dp"
        android:ems="10"
        android:hint="Start Location"
        android:inputType="text"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText
    android:id="@+id/destination"
    android:layout_width="307dp"
    android:layout_height="52dp"
    android:layout_marginStart="4dp"
    android:layout_marginTop="16dp"
    android:ems="10"
    android:hint="Destination"
    android:inputType="text"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/start" />

<ImageView
    android:id="@+id/imageView"
    android:layout_width="58dp"
    android:layout_height="41dp"
    android:layout_marginTop="28dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.954"

    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/baseline_album_24" />
```

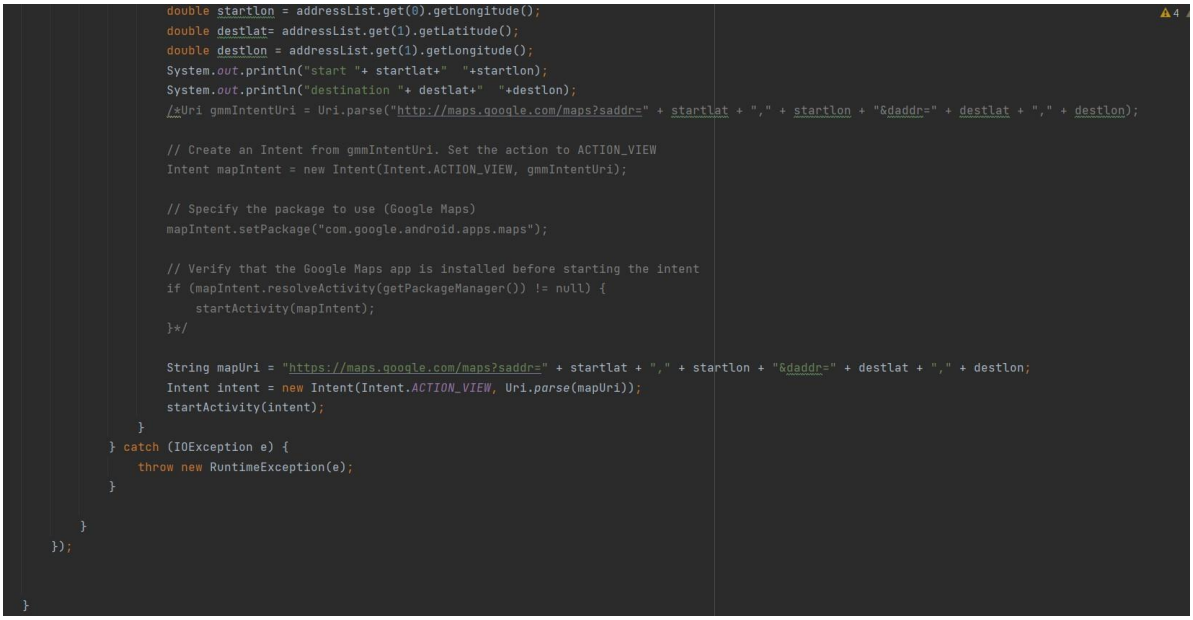
5.1.5 maps.java

```
usages
public class maps extends AppCompatActivity {

    2 usages
    EditText start,destination;
    2 usages
    Button submit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        start = findViewById(R.id.start);
        destination = findViewById(R.id.destination);
        submit = findViewById(R.id.button);

        submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Geocoder geocoder = new Geocoder( context: maps.this);
                List<Address> addressList = new ArrayList<>();
                try {
                    addressList.addAll( geocoder.getFromLocationName(start.getText().toString(), maxResults: 1));
                    addressList.addAll( geocoder.getFromLocationName(destination.getText().toString(), maxResults: 1));
                    if(addressList!=null){
                        double startlat= addressList.get(0).getLatitude();
                        double startlon = addressList.get(0).getLongitude();
                        double destlat= addressList.get(1).getLatitude();
                        double destlon = addressList.get(1).getLongitude();
                        System.out.println("start "+ startlat+"  "+startlon);
```



```
                double startlon = addressList.get(0).getLongitude();
                double destlat= addressList.get(1).getLatitude();
                double destlon = addressList.get(1).getLongitude();
                System.out.println("start "+ startlat+"  "+startlon);
                System.out.println("destination "+ destlat+"  "+destlon);
                /*Uri gmmIntentUri = Uri.parse("http://maps.google.com/maps?saddr=" + startlat + "," + startlon + "&daddr=" + destlat + "," + destlon);

                // Create an Intent from gmmIntentUri. Set the action to ACTION_VIEW
                Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);

                // Specify the package to use (Google Maps)
                mapIntent.setPackage("com.google.android.apps.maps");

                // Verify that the Google Maps app is installed before starting the intent
                if (mapIntent.resolveActivity(getPackageManager()) != null) {
                    startActivity(mapIntent);
                }*/

                String mapUri = "https://maps.google.com/maps?saddr=" + startlat + "," + startlon + "&daddr=" + destlat + "," + destlon;
                Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(mapUri));
                startActivity(intent);
            }
        } catch (IOException e) {
            throw new RuntimeException(e);
        }

    }
});


}
```

Chapter-6

RESULTS AND SNAPHOTS

Fig6: flash screen

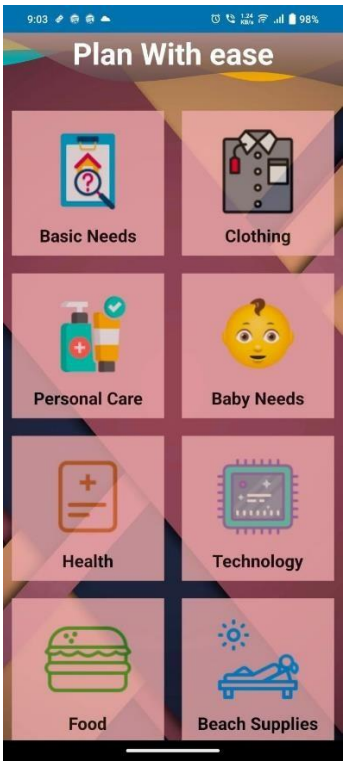


Fig7: categories

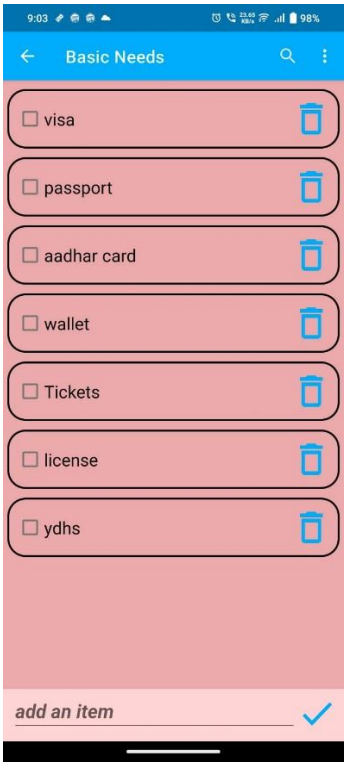


Fig8: items

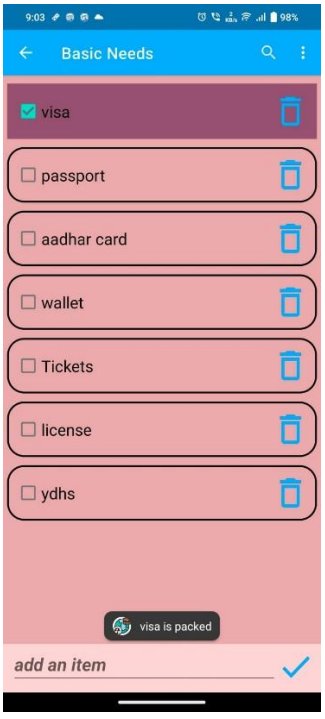


Fig9: packing items

Fig10: delete items

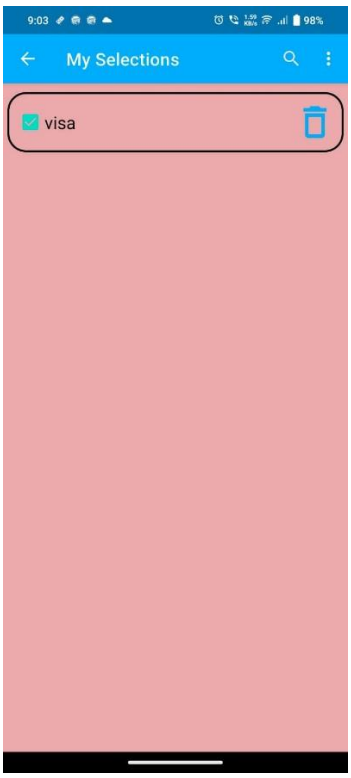


Fig11: Selected Items

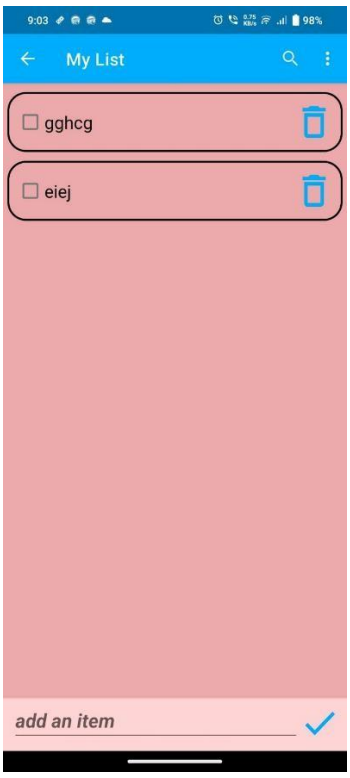


Fig12: custom list

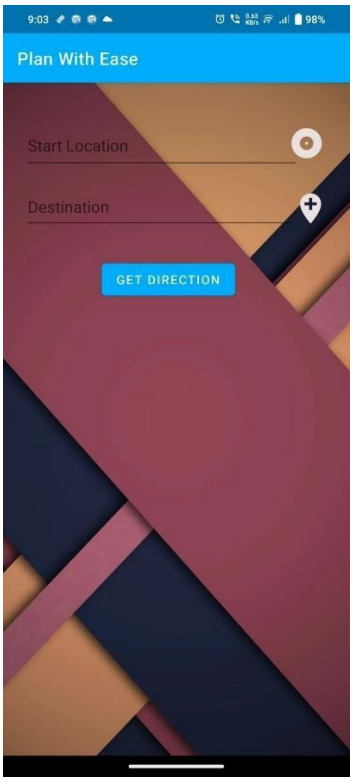


Fig13: location page

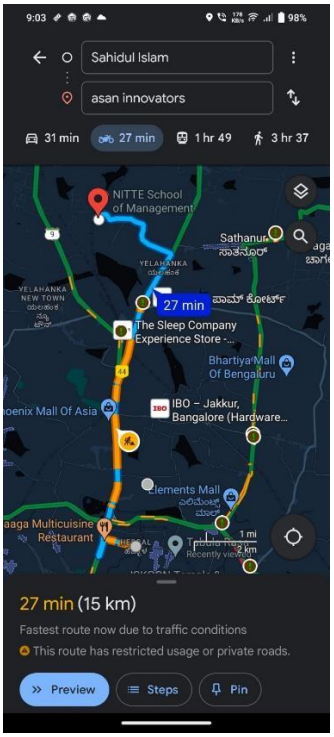


Fig14: Maps

Chapter-7

CONCLUSION

In conclusion, "Plan with Ease" emerges as a valuable solution to enhance the travel planning experience, addressing the often-overlooked aspect of efficient packing. The project's robust architecture and user-friendly interface empower travelers to organize their trips systematically, offering a diverse range of features such as customizable packing lists, map integration for route planning, and personalized item management. The seamless flow from selecting items to viewing consolidated selections, coupled with the flexibility to reset or customize, caters to individual preferences. By leveraging the Android framework, the project provides a comprehensive tool that not only simplifies the packing process but also contributes to a more organized and enjoyable travel experience. Through careful consideration of user feedback, iterative development, and adherence to a user-centric design, "Plan with Ease" endeavors to redefine how users approach and execute their travel preparations, ultimately aiming to become an indispensable companion for travelers worldwide.

REFERENCES

[1] "geekforgeeks," [Online]. Available: https://www.geeksforgeeks.org/android-architecture/.

[2] O. E. R. Emeka Agina, "Influence of destination packaging on tourism performance in Nigeria," 2018.

[3] "android doumnetation," [Online]. Available: https://developer.android.com/docs.

[4] "maps and places," [Online]. Available: https://developer.android.com/develop/sensors-and-location/location/maps-and-places.

[5] "Save data in a local database using Room," [Online]. Available: https://developer.android.com/training/data-storage/room.