My Note Application Report

SUBMITED BY

.....

OM DAUDIA

KHODA KESHWALA

KUSHAL JUNGI

INDEX

PREFACE

AS A STUDENT OF BACHELOR OF COMPUTER APPLICATION ACCORDING TO THE SYLLABUS SUBSCRIBED BY BHAKTA KAVI NARSINH MEHTA UNIVERSITY, JUNAGADH(GUJARAT). IN 6TH SEMESTER OF B.C.A. TO PREPARE A PROJECT ON ANY PARTICULAR SUBJECT OF STUDENT CHOICE. THE PROJECT SHOULD BE SELECTED ON PRECISE SUBJECT THAT WILL FUL FILL THE REQUIREMENTS.

THE MAIN OBJECTIVE OF THE PROJECT IS TO DEVELOP AWARENESS REGARDING APPLICATION OF THEORIES IN THE MODERN WORLD OF INFORMATION TECHNOLOGY.

HERE THE PROJECT REPORT TRAINING IS UNDER TAKEN AT DR.V.R. GODHANIYA INSTITUTE OF INFORMATION TECHNOLOGY IN PORBANDAR. PROJECT REPORT HAS BEEN PREPARED ON THE BASE OF INFORMATION COLLECTED WHICH IS A RESULT OF HARD WORK DONE BY US.

WE HAVE TAKEN ALMOST CARE THAT THE INFORMATION PROVIDED IN THE PROJECT IS TRUE AND PROJECT IS IN SMOOTH RUNNING CONDITION AND ERROR FREE WE SINCERELY APOLOGIES IF ANY WRONG IS GIVEN WE HAVE TRIED OUR LEVEL BEST TO PROVIDE ALL THE NECESSARY INFORMATION AS PER THE SYLLABUS.

ACKNOWLEDGEMENT

IT IS OUT GREAT PLEASURE TO PRESENT OUR PROJECT REPORT ON MY NOTE APPLICATION WHICH WE CONCEIVED DURING THE BCA 6TH SEM AFFILIATED TO BHAKTA KAVI NARSINH MEHTA UNIVERSITY.

WE TAKE THIS OPPORTUNITY TO EXPRESS OUR SINCERE GRATITUDE AND WE FEEL IMMENSE PLEASURE TO THANK OUR FACULTY, PHILOSOPHER AND GUIDE DR. NIRAV DATANI WHO HELPED US AND GAVE FULL SUPPORT IN EACH AND EVERY WAY TO FULL FILL AND ACCOMPLISH OUR PROJECT OF MY NOTE APPLICATION

WE ARE INDEBTED TO OUT COLLAGE DR. V. R. GODHANIYA COLLAGE OF I.T OUR DIRECTOR MR. DHAVAL KHER FOR GIVEN US AN EXCELLENT CHANCE TO PROVE OUR BEST WORK AND EFFORTS.

My Note Application

PROJECT PROFILE

PROJECT NAME	MY NOTE APPLICATION
FRONT END	ANDROID
BACK END	SQLITE
DEVELOPED BY	OM DAUDIA, KHODA KESHWALA AND KUSHAL JUNGI
OBJECTIVE	to develop an easy-to-use application for note-taking.
SCOPE	users can create and store personal notes, to-do lists, and reminders. helps students in organizing lecture notes and study materials.
TECHNOLOGIES USED	ANDROID
TARGET AUDIENCE/USER	NORMAL USERS
TIMELINE	TE PROJECT SPEND OVER ONE WEEK, WITH KEY MILISTONES INCLUDING PLANNING, SYSTEM DESIGN, IMPLIMENTATION AND TESTING.
DEVELOPED AT	DR. V. R. GODHANIYA COLLAGE OF I.T
TESING	Manual
TASK	ANALYSIS, DESIGNING, CODING, TESTING

ABSTRACT OF PROJECT

The My Note Application is designed to provide users with a seamless and efficient way to create, manage, and organize their notes.

This application integrates features such as text formatting, categorization and offline accessibility to enhance user experience.

It is aimed at students, professionals, and general users who require an intuitive and organized digital workspace.

The application is developed using Android for Android user, with SQLite for data storage. Future enhancements may include cloud synchronization, AI-powered suggestions, and voice notes to further improve usability and functionality.

FEASIBILITY STUDY

1. TECHNICAL FEASIBILITY

THIS PART EXAMINES WHETHER THE TECHNOLOGY AND TOOLS USED TO DEVELOP THE SYSTEM ARE ADEQUATE, CONSIDERING FACTORS SUCH AS HARDWARE, SOFTWARE, AND SKILLS.

- **TOOLS AND TECHNOLOGIES**:
  - **PROGRAMMING LANGUAGE**: JAVA
  - **FRAMEWORK**: ANDROID SDK
  - **DATABASE**:SQLITE
  - **development environment:** android studio

- **DEVELOPMENT__SKILLS**:
  EVALUATE THE TEAM'S ABILITY TO WORK WITH JAVA AND ADNDROID AND WHETHER THEY HAVE THE SKILLS TO IMPLEMENT FEATURES LIKE CREATE, EDIT AND STORE.

- **HARDWARE REQUIREMENTS**:
  - **processor:** arm-based or intel processor (for android devices and development pcs)
  - **ram:** minimum 4gb (8gb recommended for development)
  - **storage:** at least 30mb of free space for installation

- **SOFTWARE REQUIREMENTS**:
  - **operating system:** android 7.0 and above for mobile
  - **development environment:** android studio
  - **database:** sqlite (for offline data storage)
  - **programming language:** java

2. ECONOMIC FEASIBILITY

THIS SECTION EVALUATES THE COST-BENEFIT ANALYSIS OF THE SYSTEM. DETERMINE WHETHER THE PROJECT IS FINANCIALLY VIABLE BY COMPARING COSTS AGAINST EXPECTED BENEFITS.

- **INITIAL COSTS**:
  - DEVELOPMENT COST: SALARY OR HOURS SPENT BY DEVELOPERS.
  - HARDWARE: ANDROID MOBILE
  - STORAGE SIZE: 30MB

- **OPERATIONAL COSTS**:

- MAINTENANCE: UPDATING THE APPLICATION, BUG FIXES, AND TECHNICAL SUPPORT.
- SERVER AND HOSTING COSTS: IF CLOUD SERVICES ARE REQUIRED TO STORE DATA OR PROVIDE MULTI-LOCATION ACCESS.

- **EXPECTED BENEFITS**:
  - AUTOMATION OF MANUAL PROCESSES, REDUCING HUMAN ERROR AND SAVING TIME.
  - STREAMLINED OPERATIONS (INVENTORY, BILLING) TO IMPROVE PROFITABILITY.
  - SCALABILITY FOR FUTURE GROWTH (E.G., MULTI-LOCATION CAFÉS).
- **COST-BENEFIT__ANALYSIS**:
  - **development costs**
    - software tools (android studio is free, but premium plugins may be needed)
    - developer time and resources
  - **maintenance costs:**
    - regular updates and bug fixes
    - technical support for users

3. OPERATIONAL FEASIBILITY

the operational feasibility of the my note application determines how well the system will function in a real-world environment and whether it can be maintained efficiently.

- **ease of use:** the application features a simple and intuitive user interface, ensuring that users of all skill levels can operate it without extensive training.
- **performance:** designed to function smoothly on a wide range of android devices with minimal resource consumption.
- **data security:** ensures secure data storage using sqlite for offline storage.

- **maintenance & support:** regular updates and improvements can be managed easily through version control and continuous development.
- **scalability:** the application can handle an increasing number of users and notes without significant performance issues.

6. MARKET FEASIBILITY

the market feasibility of the my notetaking application assesses the potential for user adoption, competition, and revenue generation.

- **target audience:**
  - students, professionals, and general users looking for an efficient digital note-taking solution.
  - businesses and educational institutions seeking an organized note-management tool.
- **market demand:**
  - increasing demand for digital note-taking solutions due to remote work, online education, and paperless workflows.
  - competitor analysis shows the presence of apps like evernote, microsoft onenote, and google keep, but a unique feature set can differentiate the app.
- **revenue generation potential:**
  - freemium model with optional in-app purchases for premium features (cloud storage, ai-based recommendations, etc.).
  - advertisements for free users.
  - subscription-based model for enhanced productivity tools.
- **competitive advantage:**
  - offline functionality with synchronization features.
  - customizable ui and advanced search capabilities.
  - lightweight application optimized for fast performance on all android devices.

7. RISK FEASIBILITY

risk feasibility assesses potential risks associated with the development, deployment, and maintenance of the my note application. below are the key risks and mitigation strategies:

- **technical risks:**
  - **risk:** compatibility issues with different android versions.
  - **mitigation:** regular testing on various devices and os versions to ensure smooth performance.
- **security risks:**
  - **risk:** data breaches or loss of user notes due to security vulnerabilities.
  - **mitigation:** implementing encryption, secure authentication, and backup options to protect user data.
- **financial risks:**
  - **risk:** high maintenance costs, especially for cloud storage and server resources.
  - **mitigation:** optimizing cloud usage and implementing a freemium model to cover operational expenses.
- **user adoption risks:**
  - **risk:** low adoption rate due to competition from existing note-taking apps.
  - **mitigation:** offering unique features like offline access, ai-based suggestions, and an intuitive ui to attract users.
- **operational risks:**
  - **risk:** frequent crashes or performance issues due to high data loads.
  - **mitigation:** efficient database management and performance optimization through regular updates and testing.

SYSTEM ANALYSIS AND DESIGNING

- **REQUIREMENT ANALYSIS**

the requirement analysis for the my note application involves identifying functional and non- functional requirements to ensure a smooth development process.

- **functional requirements:**

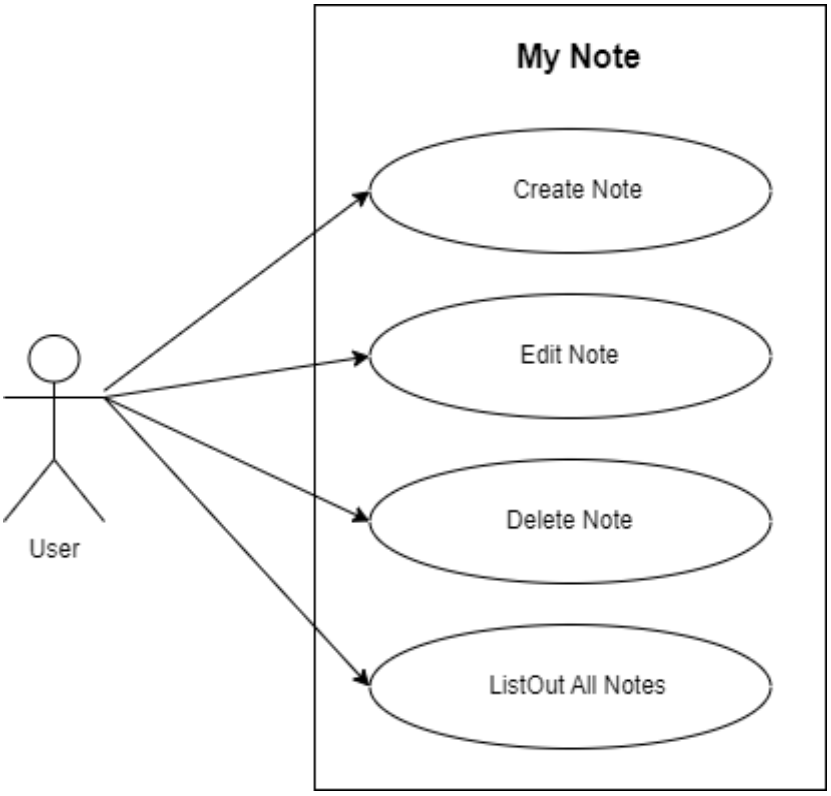
- users should be able to create, edit, and delete notes.
- the app should provide text formatting options like bold, italic, and underline.
- users should have the ability to back up and restore notes.
- the app should function both online and offline.

- **non-functional requirements:**
  - the application should have a user-friendly interface with smooth navigation.
  - performance should be optimized for low-memory devices.
  - security measures should be in place to protect user data.
  - the app should be scalable for future enhancements, such as ai-powered suggestions and cloud synchronization.

- **USE CASE DIAGRAM**

A USE CASE DIAGRAM IS USED TO REPRESENT THE DYNAMIC BEHAVIOR OF A SYSTEM. IT ENCAPSULATES THE SYSTEM'S FUNCTIONALITY BY INCORPORATING USE CASES, ACTORS, AND THEIR RELATIONSHIPS. IT MODELS THE TASKS, SERVICES, AND FUNCTIONS REQUIRED BY A SYSTEM/SUBSYSTEM OF AN APPLICATION. IT DEPICTS THE HIGH-LEVEL FUNCTIONALITY OF A SYSTEM AND ALSO TELLS HOW THE USER HANDLES A SYSTEM.

My Note Application



- **CLASS DIAGRAM (OOP)**

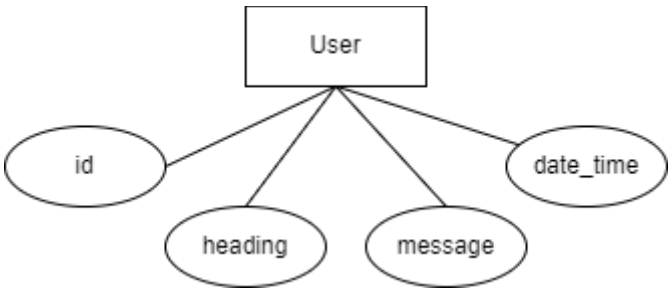
  CLASS DIAGRAMS ARE A TYPE OF UML (UNIFIED MODELING LANGUAGE) DIAGRAM USED IN SOFTWARE ENGINEERING TO VISUALLY REPRESENT THE STRUCTURE AND RELATIONSHIPS OF CLASSES WITHIN A SYSTEM I.E. USED TO CONSTRUCT AND VISUALIZE OBJECT-ORIENTED SYSTEMS.

Note_Table
id INT
heading VARCHAR(50)
message TEXT
date_time DATE

PROJECT LIFE CYCLE

- **E R DIAGRAM**

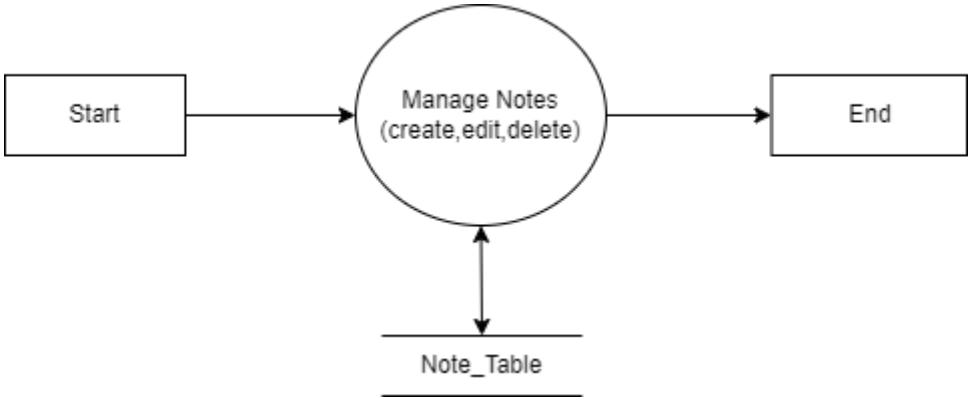
  THE ENTITY RELATIONAL MODEL IS A MODEL FOR IDENTIFYING ENTITIES TO BE REPRESENTED IN THE DATABASE AND REPRESENTATION OF HOW THOSE ENTITIES ARE RELATED. THE ER DATA MODEL SPECIFIES ENTERPRISE SCHEMA THAT REPRESENTS THE OVERALL LOGICAL STRUCTURE OF A DATABASE GRAPHICALLY.



- **DATA FLOW DIAGRAM (DFD)**

  A DATA FLOW DIAGRAM (DFD) IS A TRADITIONAL VISUAL REPRESENTATION OF THE INFORMATION FLOWS WITHIN A SYSTEM. A NEAT AND CLEAR DFD CAN DEPICT THE RIGHT AMOUNT OF THE SYSTEM REQUIREMENT GRAPHICALLY. IT CAN BE MANUAL, AUTOMATED, OR A COMBINATION OF BOTH.

  IT SHOWS HOW DATA ENTERS AND LEAVES THE SYSTEM, WHAT CHANGES THE INFORMATION, AND WHERE DATA IS STORED.

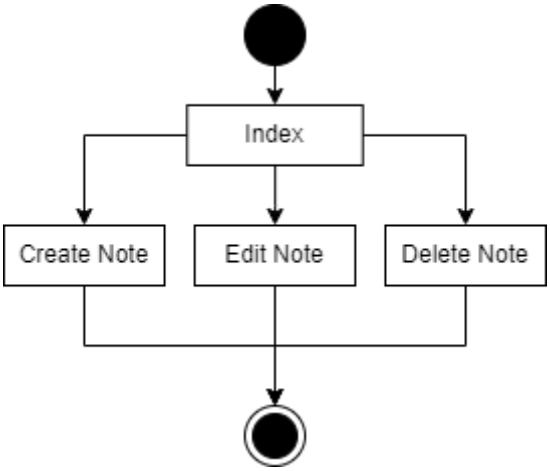


- **ACTIVITY DIAGRAM**

ACTIVITY DIAGRAMS ARE USED TO ILLUSTRATE THE FLOW OF CONTROL IN A SYSTEM AND REFER TO THE STEPS INVOLVED IN THE EXECUTION OF A USE CASE. WE CAN DEPICT BOTH SEQUENTIAL PROCESSING AND CONCURRENT PROCESSING OF ACTIVITIES USING AN ACTIVITY DIAGRAM IE AN ACTIVITY DIAGRAM FOCUSES ON THE CONDITION OF FLOW AND THE SEQUENCE IN WHICH IT HAPPENS.

WE DESCRIBE WHAT CAUSES A PARTICULAR EVENT USING AN ACTIVITY DIAGRAM.
AN ACTIVITY DIAGRAM PORTRAYS THE CONTROL FLOW FROM A START POINT TO A FINISH POINT SHOWING THE VARIOUS DECISION PATHS THAT EXIST WHILE THE ACTIVITY IS BEING EXECUTED.
THEY ARE USED IN BUSINESS AND PROCESS MODELING WHERE THEIR PRIMARY USE IS TO DEPICT THE DYNAMIC ASPECTS OF A SYSTEM.

My Note Application



DATA DICTIONARY

- TABLES

FIELD NAMES	DATA TYPE	CONSTRAINT
ID	INT	P.K A.I
HEADING	VARCHAR(50)	NOTNULL
MESSAGE	TEXT	NOTNULL
DATE_TIME	DATE	NOTNULL

TESTING

- MANUAL (ISSUES & SOLUTIONS) WITH DATA

- MANUAL TESTING IS A METHOD USED BY SOFTWARE DEVELOPERS TO RUN TESTS MANUALLY. THERE ARE MANY MANUAL TESTING TYPES WHICH ARE CARRIED OUT MANUALLY AS WELL AS AUTOMATICALLY.

- BLACK BOX TESTING:
  - IT IS A TESTING METHOD TO TEST FUNCTIONALITIES AND REQUIREMENTS OF THE SYSTEM. IT DOES NOT TEST THE INTERNAL PART OF THE SYSTEM.

- WHITE BOX TESTING:
  - IT IS A TESTING METHOD BASED ON INFORMATION OF THE INTERNAL LOGIC OF AN APPLICATION'S CODE AND ALSO KNOWN AS GLASS BOX TESTING. IT WORKS ON INTERNAL WORKING CODE OF THE SYSTEM.

- SYSTEM TESTING:
  - IT IS A TECHNIQUE TO TEST WHOLE SYSTEM.

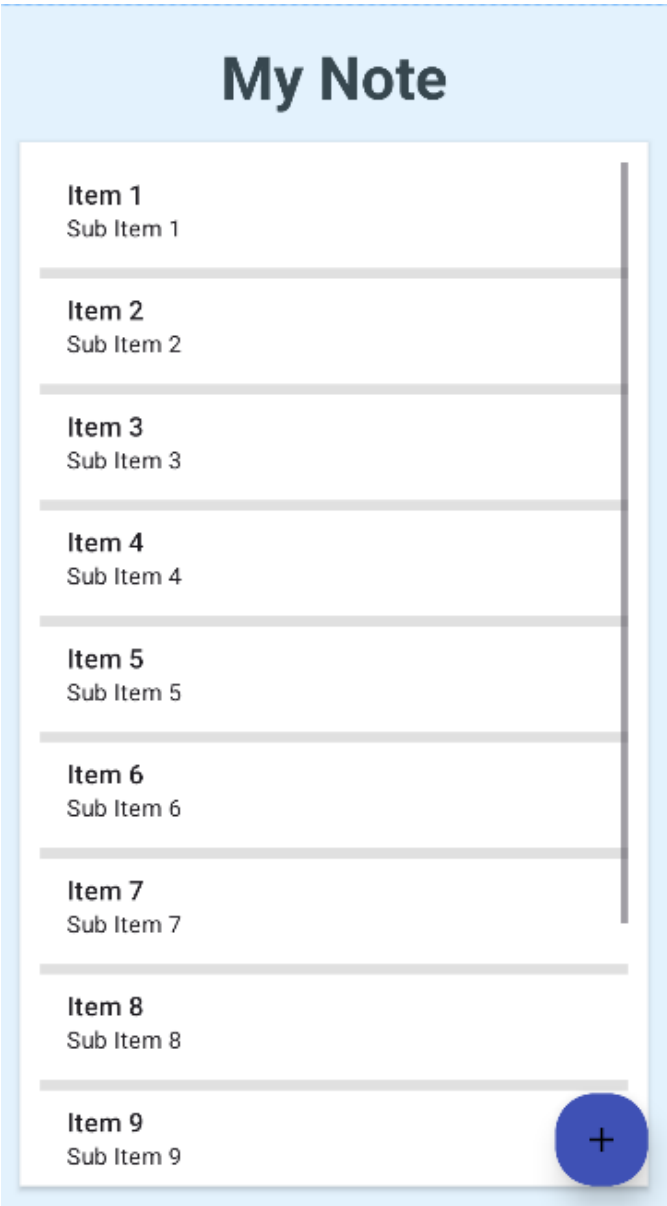
- UNIT TESTING:
  - TESTING METHOD TO TEST SPECIFIC COMPONENT OF SOFTWARE OR MODULE. IT IS SPECIALLY DONE BY PROGRAMMERS AND NOT BY TESTERS, BECAUSE IT NEEDS THOROUGH KNOWLEDGE OF THE INTERNAL PROGRAMMING DESIGN AND CODE.

- ACCEPTANCE TESTING:
  - THIS TYPE OF TESTING VERIFIES THAT THE SYSTEM MEETS THE CUSTOMER SPECIFIED REQUIREMENTS OR NOT. USER OR A CUSTOMER DOES THIS TESTING TO DECIDE WHETHER TO ACCEPT APPLICATION.

USER INTERFACE(SNAPSHOTS) & SOURCE CODE

My Note Application

- MAIN ACTIVITY



  - activity_main.xml

```
<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" android:background="#e3f2fd" tools:context=".mainactivity">

```
<!-- app title -->
<textview
    android:id="@+id/head"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/app_name"
    android:textsize="36sp"
    android:textstyle="bold"
    android:textcolor="#37474f"
    app:layout_constrainttop_totopof="parent"
    app:layout_constraintstart_tostartof="parent"
    app:layout_constraintend_toendof="parent"
    android:paddingtop="20dp"
    android:gravity="center"/>

<!-- notes list -->
<listview
    android:id="@+id/notelist"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_margin="16dp"
    android:background="#ffffff"
    android:cliptopadding="false"
    android:dividerheight="6dp"
    android:elevation="2dp"
    android:padding="12dp"
    android:scrollbars="vertical"
    app:layout_constraintbottom_tobottomof="parent"
    app:layout_constraintend_toendof="parent"
    app:layout_constraintstart_tostartof="parent"
    app:layout_constrainttop_tobottomof="@id/head"
    app:layout_constraintvertical_bias="0.0" />

<!-- floating action button (overlaying listview) -->
<imagebutton
    android:id="@+id/newnote_btn"
    android:layout_width="56dp"
    android:layout_height="56dp"
    android:layout_margin="16dp"
    android:background="@drawable/btn"

    android:contentdescription="add note"
    android:src="@drawable/baseline_add_24"
    android:elevation="10dp"
    app:layout_constraintbottom_tobottomof="parent"
    app:layout_constraintend_toendof="parent" />
</androidx.constraintlayout.widget.constraintlayout>
```

- ManActivity.java

```
package com.example.mynote;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Adapter;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
        import android.widget.ImageButton;
        import android.widget.ListView;
        import android.widget.PopupMenu;
import android.widget.TextView;
import android.widget.Toast;
        import androidx.activity.EdgeToEdge;

import androidx.annotation.NonNull; import androidx.appcompat.app.AppCompatActivity; import
androidx.core.graphics.Insets; import androidx.core.view.ViewCompat; import
androidx.core.view.WindowInsetsCompat;

import java.util.Calendar; import java.util.List;

public class MainActivity extends AppCompatActivity {
```

- ```
  ImageButton newnotebtn;
  ListView itemlist;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
      EdgeToEdge.enable(this);
      setContentView(R.layout.activity_main);

      newnotebtn = findViewById(R.id.newnote_btn);
      itemlist = findViewById(R.id.notelist);

      //intent jump to new note activity from mainactivity
      Intent next = new Intent(MainActivity.this,
         new_note_activity.class);

      //add new note button, used to jump to new note activity
      newnotebtn.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {
              next.putExtra("source", 1);
              startActivity(next);
          }
  ```
  ```

```
    });

    itemlist.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
            item_class sel_item = (item_class)
parent.getItemAtPosition(position);
            Log.d("FLAG", "item - "+sel_item.getId());
            next.putExtra("source",0);
            next.putExtra("id", sel_item.getId());
            startActivity(next);
        }
    });

    itemlist.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
        @Override
        public boolean onItemLongClick(AdapterView<?> parent, View view,
int position, long id) {
            item_class sel_item = (item_class)
parent.getItemAtPosition(position);

            MyConnection con = new MyConnection(MainActivity.this);
            Log.d("Long", sel_item.getTime());
            // Create and show the popup menu
            PopupMenu popupMenu = new PopupMenu(MainActivity.this, view);
            popupMenu.getMenuInflater().inflate(R.menu.dele_option,
popupMenu.getMenu());

            popupMenu.setOnMenuItemClickListener(item -> {
                if (item.getItemId() == R.id.menu_delete) {
                    // Handle delete operation
                    boolean res = con.Delete_Operation(sel_item.getId());
                    if(res){
                        refresh_list();
                        Toast.makeText(MainActivity.this, "Deleted",
Toast.LENGTH_SHORT).show();
                    }
                    else{
                        Toast.makeText(MainActivity.this, "Delete Denied",
Toast.LENGTH_SHORT).show();
                    }
                    refresh_list();
                    return true;
                }
                return false;
            });
            popupMenu.show();
            return true;
        }
```

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

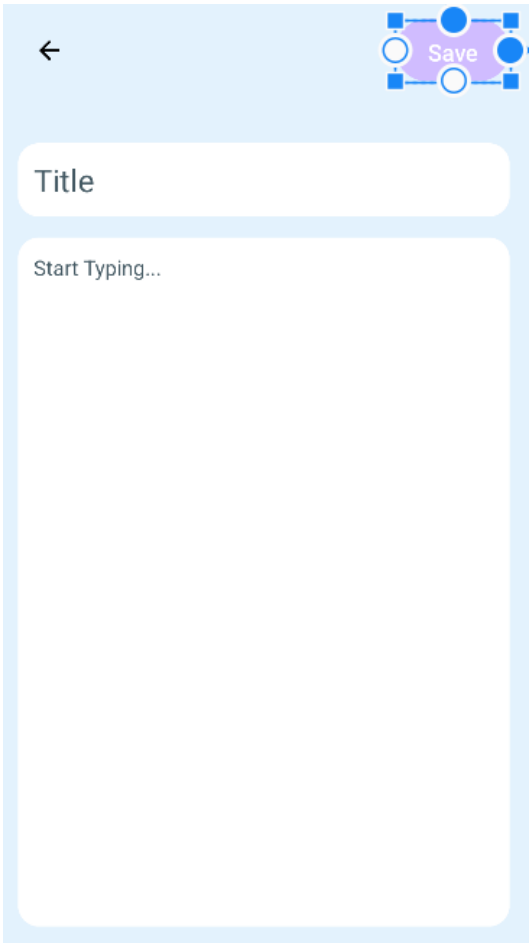
    }

    private void refresh_list(){
        itemlist = findViewById(R.id.notelist);
        MyConnection con = new MyConnection(this);
        List<item_class> items = con.Display_Operation();

        ItemAdapter adapter = new ItemAdapter(this, items);
        itemlist.setAdapter(adapter);
    }

}
```

- NEW NOTE ACTIVITY



- activity_new_note.xml

<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"
android:background="#E3F2FD" tools:context=".new_note_activity">

```
<!-- Back Button -->
<ImageButton
    android:id="@+id/back_btn"
    android:layout_width="48dp"
    android:layout_height="48dp"
    android:background="@android:color/transparent"
    android:contentDescription="@string/backbtn"
    android:src="@drawable/baseline_arrow_back_24"
    android:elevation="6dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_margin="16dp" />

<!-- Save Button -->
<Button
    android:id="@+id/save_btn"
    android:layout_width="wrap_content"
    android:layout_height="48dp"
    android:background="@drawable/btn"
    android:text="@string/savebtn"
    android:textColor="@android:color/white"
    android:textSize="18sp"
    android:padding="10dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_margin="16dp"/>

<!-- Current Time -->
<TextView
    android:id="@+id/txt_currenttime"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="end"
    android:textColor="#455A64"
    android:textSize="16sp"
    android:layout_marginEnd="16dp"
    app:layout_constraintTop_toBottomOf="@+id/save_btn"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="10dp" />

<!-- Note Title -->
<EditText
```

```
    android:id="@+id/txt_title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/rounded_input"
    android:hint="@string/titlehint"
    android:inputType="text"
    android:padding="12dp"
    android:textSize="24sp"
    android:textColorHint="#455A64"
    android:layout_margin="16dp"
    app:layout_constraintTop_toBottomOf="@id/txt_currenttime"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<!-- Note Content -->
<EditText
    android:id="@+id/txt_msg"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:background="@drawable/rounded_input"
    android:hint="@string/msghint"
    android:gravity="start|top"
    android:inputType="textMultiLine"
    android:padding="12dp"
    android:textSize="16sp"
    android:textColorHint="#455A64"
    android:layout_margin="16dp"
    app:layout_constraintTop_toBottomOf="@id/txt_title"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

- new_note_activity.java

```
package com.example.mynote;

import android.content.Intent; import android.os.Bundle; import android.util.Log; import
android.view.View; import android.widget.Button; import android.widget.EditText; import
android.widget.ImageButton; import android.widget.TextView; import android.widget.Toast;

import androidx.activity.EdgeToEdge; import androidx.appcompat.app.AppCompatActivity; import
androidx.core.graphics.Insets; import androidx.core.view.ViewCompat; import
androidx.core.view.WindowInsetsCompat;

import java.util.Calendar; import java.util.Date; import java.util.GregorianCalendar; import java.util.List;

public class new_note_activity extends AppCompatActivity { Button save_btn; EditText title_txt,
msg_txt; TextView time_txt;
```

```
ImageButton back_btn;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_new_note);

    back_btn = (ImageButton) findViewById(R.id.back_btn);
    save_btn = (Button) findViewById(R.id.save_btn);


    title_txt = (EditText) findViewById(R.id.txt_title);
    msg_txt = (EditText) findViewById(R.id.txt_msg);
    time_txt = (TextView) findViewById(R.id.txt_currenttime);

    MyConnection con = new MyConnection(new_note_activity.this);

    Bundle source = getIntent().getExtras();
    int save_or_edit_flag = source.getInt("source");
    int edit_note_id = source.getInt("id");


    if(save_or_edit_flag == 0){
        try {
            List<item_class> edit_note = con.Get_Note(edit_note_id);
            title_txt.setText(edit_note.get(0).title);
            msg_txt.setText(edit_note.get(0).msg);
            time_txt.setText(edit_note.get(0).time);


// Log.d("edited note Error", edit_note.get(0).id+""); // Log.d("edited note Error", "Error -
"+edit_note.get(1).title); // Log.d("edited note Error", "Error - "+edit_note.get(2).msg); // Log.d("edited
note Error", "Error - "+edit_note.get(3).time); } catch (Exception e) { Log.d("edited note Error", "Error -
"+e.toString()); } }

   Calendar cal = Calendar.getInstance();

    Intent previospage = new Intent(this, MainActivity.class);
    back_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            startActivity(previospage);
        }
    });

    save_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(title_txt.getText().toString().equals("") &&
msg_txt.getText().toString().equals("")){
                Toast.makeText(new_note_activity.this, "Access Denied",
```

```
Toast.LENGTH_SHORT).show();
                startActivity(previospage);
            }
            else {
                if(save_or_edit_flag == 1){
                    boolean res =
con.Insert_Operation(title_txt.getText().toString(), msg_txt.getText().toString(),
cal.getTime().toString());
                    if (res) {
                        Toast.makeText(new_note_activity.this, "Saved",
Toast.LENGTH_SHORT).show();
                        startActivity(previospage);
                    } else {
                        Toast.makeText(new_note_activity.this, "Not Saved",
Toast.LENGTH_SHORT).show();
                        startActivity(previospage);
                    }
                }
                else{
                    boolean res = con.Update_Operation(edit_note_id,
title_txt.getText().toString(), msg_txt.getText().toString(),
cal.getTime().toString().replaceFirst("GMT+$", ""));
                    if (res) {
                        Toast.makeText(new_note_activity.this, "Updated",
Toast.LENGTH_SHORT).show();
                        startActivity(previospage);
                    } else {
                        Toast.makeText(new_note_activity.this, "Not Updated",
Toast.LENGTH_SHORT).show();
                        startActivity(previospage);
                    }
                }


            }

        }
    });
}


}
```