**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI – 590018**



**A MINI-PROJECT REPORT ON**

**“NOTE APP”**

**MOBILE APPLICATION LABORATORY WITH MINI PROJECT**

**18CSMP68**

Submitted by:

BHARATH M – 4VV19IS013

DARASHAN V – 4VV19IS019

ISHA C – 4VV19IS032

UNDER THE GUIDANCE OF

**Prof. RAKSHITHA M S**  **Prof. R KASTURI RANGAN**

**Assistant Professor**

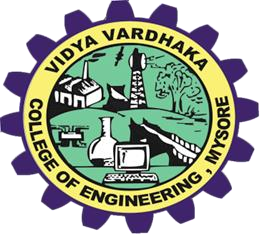
**Department of ISE**

**VVCE, Mysore**

**Assistant Professor**

**Department of ISE**

**VVCE, Mysore**



**2021-2022**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**VIDYAVARDHAKA COLLEGE OF ENGINEERING**

**MYSURU-570002**

Vidyavardhaka College of Engineering

Gokulum 3rd Stage,

Mysuru-570002

**Department of Information Science and Engineering**

**CERTIFICATE**

This is to certify that the mini-project report entitled “**NOTE APP**” is a bona fide work carried out by **BHARATH M**, **DARSHAN V** and **ISHA C**, students of 6th semester Information Science and Engineering, **Vidyavardhaka College of Engineering, Mysuru** in partial fulfilment for the award of the degree of Bachelor of Engineering in **Information Science & Engineering of the Visvesvaraya Technological University, Belagavi**, during the academic year **2021-2022**. It is certified that all the suggestions and corrections indicated for the internal assessment have been incorporated in the report deposited in the department library. The report has been approved as it satisfies the requirements in respect of mini-project work prescribed for the said degree.

**Signature of the Guide Signature of the Guide Signature of the HOD**

**(Prof. Rakshitha M S)**  **(Prof R Kasturi Rangan) (Dr. A B Rajendra)**

**Name of the Examiners Signature with Date**

1.

2.

## ACKNOWLEDGEMENT

The Mini project would not have been possible without the guidance, assistance, and suggestions of many individuals. I would like to express our deep sense of gratitude and indebtedness to each one who has helped me to make this project a success.

We heartily thank our beloved Principal, **Dr. B Sadashive Gowda** for his wholehearted support and for his kind permission to undergo the mini project.

We wish to express our deepest gratitude to **Dr. A B Rajendra**, Head of Department, Information Science and Engineering, VVCE, for his constant encouragement and inspiration in taking up this mini project.

We gracefully thank our mini-project guides, **Prof. Rakshita M S and Prof. R Kasturi Rangan,** Assistant Professor, Dept. of Information Science and Engineering for their encouragement and advice throughout the course of the mini project work

In the end, we are anxious to offer our sincere thanks to our family members and friends for their valuable suggestions and encouragement.

BHARATH M(4VVIS013)

DARSHAN V(4VV19IS019)

ISHA C(4VV19IS032)

### **ABSTRACT**

Taking notes is about staying focused, being interactive, and recalling information that otherwise is lost. Typically when we get an idea or we hear one, we tend to look for a piece of paper to jot it down. However, the times are changing. We now live in an era of information overload. And, sadly, it has made it quite easy for us to lose information on a daily basis. As a result, it becomes even easier to forget to check out the information or share it with your team. Or worse, lose the information that you jotted down. This is where our app comes in handy. It will cover all such issues and make the entire process as relevant user friendly as possible. The required data is stored and retrieved using SQLite database.

The app is coded in Java using Android Studio Taking notes is about staying focused, being interactive, and recalling information that otherwise is lost. Typically when we get an idea or we hear one, we tend to look for a piece of paper to jot it down. However, the times are changing. We now live in an era of information overload. And, sadly, it has made it quite easy for us to lose information on a daily basis. As a result, it becomes even easier to forget to check out the information or share it with your team. Or worse, lose the information that you jotted down. This is where our app comes in handy. It will cover all such issues and make the entire process as relevant user friendly as possible. The required data is stored and retrieved using SQLite database. The app is coded in Java using Android Studio.

**TABLE OF CONTENTS**

**CHAPTERS Page no**

1. **INTRODUCTION**1.1. Motivation 1   
   1.2. Proposed System 1  
   1.3. Related Work 1
2. **REQUIREMENTS ANALYSIS AND SYSTEM SPECIFICATIONS**2.1 Software Requirements Specification Document 2  
   2.2 Validation 2
3. **SYSTEM DESIGN**3.1 Design Approach 3  
   3.2 Layout Approach 4
4. **IMPLEMENTATION**4.1 Introduction to Programming Languages, IDEs, Tools and Technologies 6  
   4.2 Layout Implementation 7
5. **TESTING**5.1 Types of Testing 12  
   5.2 Test cases and its result 12
6. **SNAPSHOTS** 14
7. **FUTURE WORK** 18
8. **CONCLUSION** 19

**BIBLIOGRAPY** 20

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure number** | **Figure name** | **Page number** |
| Fig 3.1.1 | Use case Diagram | 3 |
| Fig 3.2.1 | Layout Main page | 3 |
| Fig 3.2.2 | Layout Note-content page | 4 |
| Fig 3.2.3 | Layout Splash-Screen Page | 4 |
| Fig 3.2.4  Fig 6.1.1  Fig 6.1.2  Fig 6.1.3  Fig 6.1.4  Fig 6.1.5  Fig 6.1.6 | Layout Add-Note Page  Snapshot Activity-Main  Snapshot Activity-Main Dark mode  Snapshot Note-Activity  Snapshot Splash screen  Snapshot Search screen  Snapshot title, description error | 5  12  13  13  14  14  15 |

**CHAPTER-1**

**INTRODUCTION**

## 1.1 MOTIVATION

NOTE APP is a user-friendly application. It is used to digitize the note taking process through an android app so that anyone can take notes and store it from anywhere efficiently, without any hassle.

## 1.2 PROPOSED SYSTEM

NOTE APP: Note taking app allow us to store all notes and important information. With the help of this app, the entire process is digitized. Type notes on device just as one would using pen and paper. The application is implemented on android platform which is linked to the SQLite database for access of data.

Our project will make the note taking procedure easier for the users, by having a user-friendly UI for typing, storing and manage the note according to priority. We can also delete the note with just a swipe. There will be a provision for searching a note. We have proposed an exclusive feature which lets the user add a note based on the priority of the note or work to be done.

**1.3 RELATED WORK**

The existing note taking app has some of the basic functionalities of adding, searching, deleting and modifying an existing note. Our project will make the note taking procedure easier for the users, by having a user-friendly UI for typing, storing and manage the note according to priority. We can also delete the note with just a swipe.

When compared to the normal app, this app can set priority of note and also it fetches the date, time and year of the note created.

# **CHAPTER-2**

# **REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATIONS**

## 2.1 SOFWARE REQUIREMENTS SPECIFICATION DOCUMENT

OS: Android 8 or higher

Memory: 8 GB RAM

Free storage: 2GB

Android Studio

Android Software Development Kit

Java Development Kit

The SDK and AVD Manager

## 2.2 VALIDATION

In this project, when the user creates note he/she should enter the title and also the description. If any of these fields is kept empty, the app displays enter title if title is kept empty and enter description if description is empty.

**CHAPTER-3**

**3.SYSTEM DESIGN**

**3.1 DESIGN APPROACH**

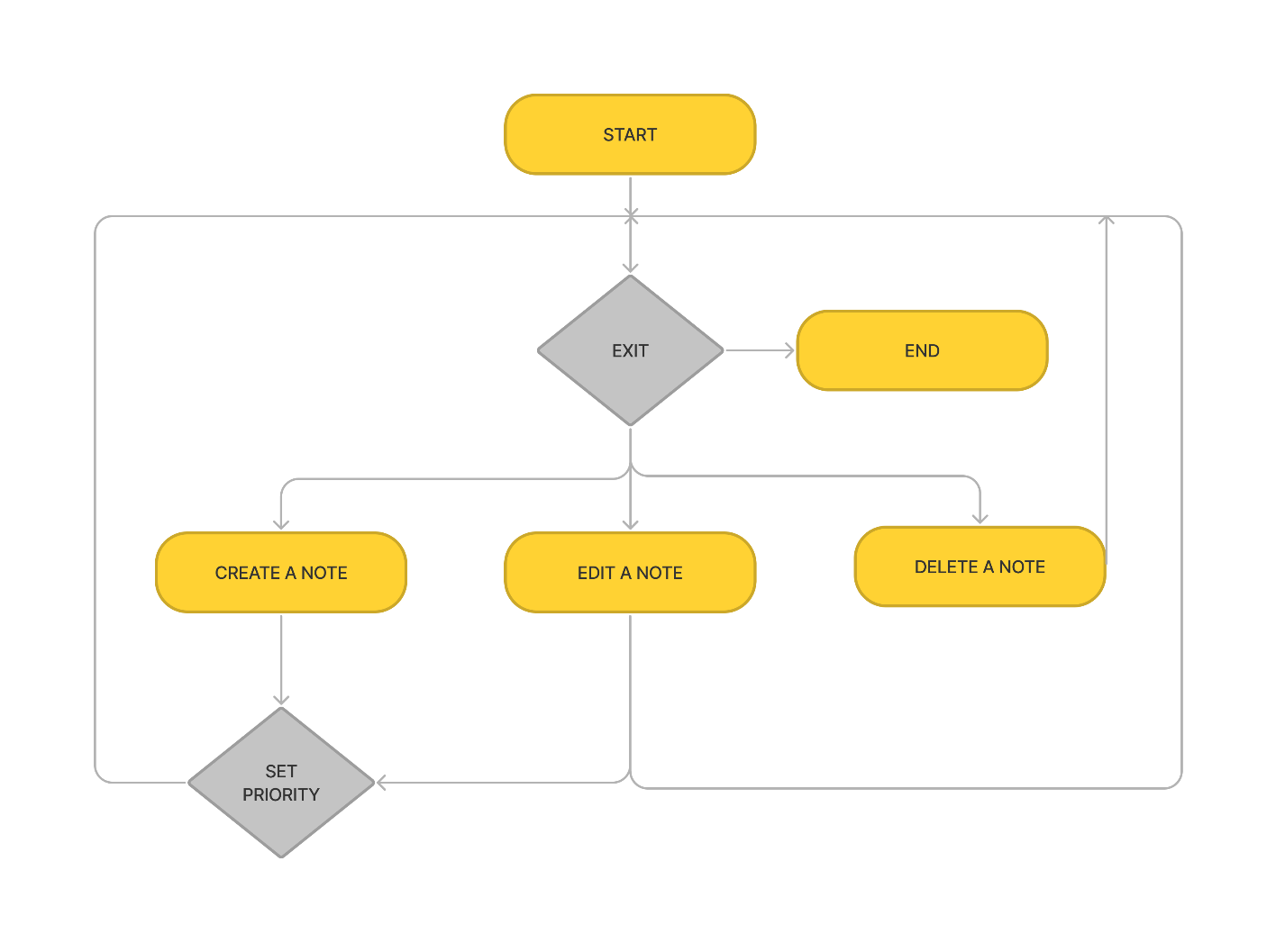
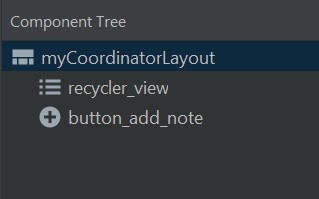
****

Fig 3.1.1: This is the basic design approach of the developed application

**3.2 LAYOUT APPROACH**

**MAIN SCREEN**

****

**NOTE ITEM**

Fig 3.2.1: This is the layout of the main screen which displays the note added and also displays the button to add new note

**NOTE CONTENT**

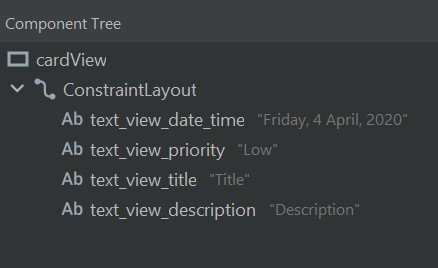
****

Fig 3.2.2: This is the layout which displays the content of the note added, which can further be edited and saved.

**SPLASH SCREEN**

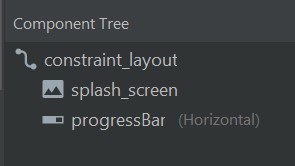
****

Fig 3.2.3: This the layout of the first loading screen when user opens the application

**ADD NOTE**

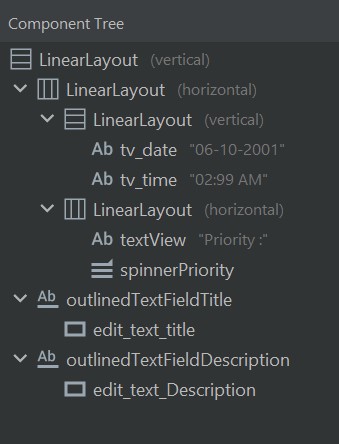
****

Fig 3.2.4: This the layout where the user can add new note and set title, description, priority or modify an existing note

**CHAPTER-4**

**IMPLEMENTATION**

**4.1 INTRODUCTION TO PROGRAMMING LANGUAGES, IDEs, TOOLS AND TECHNOLOGIES**

**ANDROID STUDIO**

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps.

**ANDROID SOFTWARE DEVELOPMENT KIT**

The Android SDK (software development kit) is a set of development tools used to develop applications for the Android platform. It’s a set of software tools and programs used by developers to create applications for specific platforms. The Android SDK includes the following: Required libraries, Debugger.

**JAVA DEVELOPMENT KIT**

Java is one of the most popular coding languages out there, and Java Development Kit (or JDK) is its official development package. Currently, Java Development Kit is also one of the most popular development environments in which to code Java. The Java Development Kit offers a wide range of practical tools like javac. You also have java and jdb which works as the debugger of the system.

**THE SDK AND AVD MANAGER**

The sdk manager is a command line tool that allows you to view, install, update, and uninstall packages for the Android SDK. The avd manager is a command line tool that allows you to create and manage Android Virtual Devices (AVDs) from the command line. An AVD lets you define the characteristics of an Android handset, Wear OS watch, or Android TV device that you want to simulate in the Android Emulator.

**SQLITE-DATABASE**

SQLite is a open source SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation. SQLite supports all the relational database features. In order to access this database, you don't need to establish any kind of connections for it like JDBC,ODBC etc. The main package is android.database.sqlite that contains the classes to manage your own databases In order to create a database you just need to call this method open Or Create Database with your database name and mode as a parameter. It returns an instance of SQLite database which you have to receive in your own object.

**4.2 LAYOUT IMPLEMENTATION**

**ADD NOTE**

**public** **class** **AddEditNoteActivity** **extends** AppCompatActivity {

**public** **static** **final** String EXTRA\_ID = "com.example.achitectureexample.EXTRA\_ID";

**public** **static** **final** String EXTRA\_TITLE = "com.example.achitectureexample.EXTRA\_TITLE";

**public** **static** **final** String EXTRA\_DESCRIPTION = "com.example.achitectureexample.EXTRA\_DESCRIPTION";

**public** **static** **final** String EXTRA\_PRIORITY = "com.example.achitectureexample.EXTRA\_PRIORITY";

**public** **static** **final** String EXTRA\_PRIORITY\_NUMBER = "com.example.achitectureexample.EXTRA\_PRIORITY\_NUMBER";

**public** **static** **final** String EXTRA\_DATE = "com.example.achitectureexample.EXTRA\_DATE";

**public** **static** **final** String EXTRA\_TIME = "com.example.achitectureexample.EXTRA\_TIME";

**private** TextView tvDate, tvTime;

**private** TextInputEditText editTextTitle;

**private** TextInputEditText editTextDescription;

**private** Spinner spinnerPriority;

//private String currentDate;

**@Override**

**protected** **void** **onCreate**(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.activity\_add\_note);

Calendar calendar = Calendar.getInstance();

//currentDate = DateFormat.getDateInstance(DateFormat.MEDIUM).format(calendar.getTime());

SimpleDateFormat dateFormat = **new** SimpleDateFormat("dd MMMM yyyy");

SimpleDateFormat timeFormat = **new** SimpleDateFormat("hh:mm a");

String date = dateFormat.format(calendar.getTime());

String ntime = timeFormat.format(calendar.getTime());

String time = ntime.replace("am", "AM").replace("pm", "PM");

editTextTitle = findViewById(R.id.edit\_text\_title);

editTextDescription = findViewById(R.id.edit\_text\_Description);

spinnerPriority = findViewById(R.id.spinnerPriority);

tvDate = findViewById(R.id.tv\_date);

tvTime = findViewById(R.id.tv\_time);

getSupportActionBar().setHomeAsUpIndicator(R.drawable.ic\_close);

Intent intent = getIntent();

**if** (intent.hasExtra(EXTRA\_ID)) {

setTitle("Edit Note");

editTextTitle.setText(intent.getStringExtra(EXTRA\_TITLE));

editTextDescription.setText(intent.getStringExtra(EXTRA\_DESCRIPTION));

tvDate.setText(intent.getStringExtra(EXTRA\_DATE));

tvTime.setText(intent.getStringExtra(EXTRA\_TIME));

//ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,R.array.priorityList,R.layout.style\_spinner);

String[] array = {"High", "Medium", "Low"};

ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**, R.layout.style\_spinner, array);

spinnerPriority.setAdapter(adapter);

//spinnerPriority.setSelection(intent.getIntExtra(EXTRA\_PRIORITY\_NUMBER,1));

} **else** {

setTitle("Add Note");

tvDate.setText(date);

tvTime.setText(time);

//ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,R.array.priorityList,R.layout.style\_spinner);

String[] array = {"High", "Medium", "Low"};

ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**, R.layout.style\_spinner, array);

spinnerPriority.setAdapter(adapter);

//spinnerPriority.setSelection(intent.getIntExtra(EXTRA\_PRIORITY\_NUMBER,1));

}

}

**SAVE NOTE**

**private** **void** **saveNote**() {

String title = editTextTitle.getText().toString();

String description = editTextDescription.getText().toString();

String priority = spinnerPriority.getSelectedItem().toString();

String date = tvDate.getText().toString();

String time = tvTime.getText().toString();

**if** (title.trim().isEmpty() || description.trim().isEmpty()) {

Toast.makeText(**this**, "Please insert a title and description", Toast.LENGTH\_SHORT).show();

**return**;

}

Intent data = **new** Intent();

data.putExtra(EXTRA\_TITLE, title);

data.putExtra(EXTRA\_DESCRIPTION, description);

data.putExtra(EXTRA\_PRIORITY, priority);

data.putExtra(EXTRA\_PRIORITY\_NUMBER, spinnerPriority.getSelectedItem().toString());

data.putExtra(EXTRA\_DATE, date);

data.putExtra(EXTRA\_TIME, time);

**int** priorityNumber = **0**;

**if** (priority.equals("High")) {

priorityNumber = **3**;

} **else** **if** (priority.equals("Medium")) {

priorityNumber = **2**;

} **else** **if** (priority.equals("Low")) {

priorityNumber = **1**;

}

data.putExtra(EXTRA\_PRIORITY\_NUMBER, priorityNumber);

**int** id = getIntent().getIntExtra(EXTRA\_ID, -**1**);

**if** (id != -**1**) {

data.putExtra(EXTRA\_ID, id);

}

setResult(RESULT\_OK, data);

finish();

}

**@Override**

**public** **boolean** **onCreateOptionsMenu**(Menu menu) {

MenuInflater menuInflater = getMenuInflater();

menuInflater.inflate(R.menu.add\_note\_menu, menu);

**return** **true**;

}

**@Override**

**public** **boolean** **onOptionsItemSelected**(**@NonNull** MenuItem item) {

**switch** (item.getItemId()) {

**case** R.id.save\_note:

saveNote();

**return** **true**;

**default**:

**return** **super**.onOptionsItemSelected(item);

}}}

**DELETE NOTE**

**public** **class** **SwipeToDeleteCallback** **extends** ItemTouchHelper.Callback {

Context mContext;

**private** Paint mClearPaint;

**private** ColorDrawable mBackground;

**private** **int** backgroundColor;

**private** Drawable deleteDrawable;

**private** Drawable deleteDrawable2;

**private** **int** intrinsicWidth;

**private** **int** intrinsicHeight;

**private** **int** intrinsicWidth2;

**private** **int** intrinsicHeight2;

**public** **SwipeToDeleteCallback**(Context context) {

mContext = context;

mBackground = **new** ColorDrawable();

backgroundColor = Color.parseColor("#b80f0a");

mClearPaint = **new** Paint();

mClearPaint.setXfermode(**new** PorterDuffXfermode(PorterDuff.Mode.CLEAR));

deleteDrawable = ContextCompat.getDrawable(mContext, R.drawable.ic\_delete);

deleteDrawable2 = ContextCompat.getDrawable(mContext, R.drawable.ic\_delete);

intrinsicWidth = deleteDrawable.getIntrinsicWidth();

intrinsicHeight = deleteDrawable.getIntrinsicHeight();

intrinsicWidth2 = deleteDrawable2.getIntrinsicWidth();

intrinsicHeight2 = deleteDrawable2.getIntrinsicHeight();

}

**@Override**

**public** **int** **getMovementFlags**(**@NonNull** RecyclerView recyclerView, **@NonNull** RecyclerView.ViewHolder viewHolder) {

**return** **makeMovementFlags**(**0**, ItemTouchHelper.LEFT );

}

**@Override**

**public** **boolean** **onMove**(**@NonNull** RecyclerView recyclerView, **@NonNull** RecyclerView.ViewHolder viewHolder, **@NonNull** RecyclerView.ViewHolder viewHolder1) {

**return** **false**;

}

**@Override**

**public** **void** **onChildDraw**(**@NonNull** Canvas c, **@NonNull** RecyclerView recyclerView, **@NonNull** RecyclerView.ViewHolder viewHolder, **float** dX, **float** dY, **int** actionState, **boolean** isCurrentlyActive) {

**super**.onChildDraw(c, recyclerView, viewHolder, dX, dY, actionState, isCurrentlyActive);

View itemView = viewHolder.itemView;

**int** itemHeight = itemView.getHeight();

**boolean** isCancelled = dX == **0** && !isCurrentlyActive;

**if** (isCancelled) {

clearCanvas(c, itemView.getRight() + dX, (**float**) itemView.getTop(), (**float**) itemView.getRight(), (**float**) itemView.getBottom());

**super**.onChildDraw(c, recyclerView, viewHolder, dX, dY, actionState, isCurrentlyActive);

**return**;

}

mBackground.setColor(backgroundColor);

mBackground.setBounds(itemView.getLeft(), itemView.getTop(), itemView.getRight(), itemView.getBottom());

mBackground.draw(c);

**int** deleteIconTop = itemView.getTop() + (itemHeight - intrinsicHeight) / **2**;

**int** deleteIconMargin = (itemHeight - intrinsicHeight) / **2**;

**int** deleteIconLeft = itemView.getRight() - deleteIconMargin - intrinsicWidth;

**int** deleteIconRight = itemView.getRight() - deleteIconMargin;

**int** deleteIconBottom = deleteIconTop + intrinsicHeight;

**int** deleteIconLeft2 = itemView.getLeft() + deleteIconMargin;

**int** deleteIconRight2 = itemView.getLeft() + deleteIconMargin+intrinsicWidth ;

deleteDrawable.setBounds(deleteIconLeft, deleteIconTop, deleteIconRight, deleteIconBottom);

deleteDrawable.draw(c);

// deleteDrawable2.setBounds(deleteIconLeft2, deleteIconTop, deleteIconRight2, deleteIconBottom);

// deleteDrawable2.draw(c);

**super**.onChildDraw(c, recyclerView, viewHolder, dX, dY, actionState, isCurrentlyActive);

}

**private** **void** **clearCanvas**(Canvas c, Float left, Float top, Float right, Float bottom) {

c.drawRect(left, top, right, bottom, mClearPaint);

}

**@Override**

**public** **float** **getSwipeThreshold**(**@NonNull** RecyclerView.ViewHolder viewHolder) {

**return** **0.7f**;

}

}

**CHAPTER-5**

**TESTING**

**5.1 TYPES OF TESTING**

**5.1.1 UNIT TESTING**

Unit testing involves the testing of each unit or individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validate unit components with its performance. The purpose of unit testing is to test the correctness of isolated code. A unit component is an individual function or code of the application. White box testing approach used for unit testing and usually done by developers.

**5.1.2 INTEGRATION TESTING**

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units. Unit testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The goal of integration testing is to check the correctness of communication among all the modules.

**5.2 TEST CASES AND ITS RESULTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST ID** | **TEST CASE NAME** | **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **RESULT** |
| 1 | Add Note | Enter title, description and set priority | Note created | Note created | Pass |
| 2 | Delete note | Swipe to delete note | Note deleted | Note deleted | pass |
| 3 | Add Note | Enter title and no description | Enter description | Enter description | pass |
| 4 | Edit Note | Edit title, description and set priority | Note updated | Note updated | Pass |
| 5 | Search Note | Enter title | Note searched | Note searched | Pass |
| 6 | New note button | Add note button clicked | New note screen | New note screen | Pass |
| 7 | Add Note | Enter description and no title | Enter title | Enter title | Pass |
| 8 | Dark Mode | Toggle dark mode switch | Screen dark mode | Screen dark mode | Pass |
| 9 | Save note | Click Save note button | Note saved | Note saved | Pass |

**CHAPTER-6**

**SNAPSHOTS**

**ACTIVITY\_MAIN**

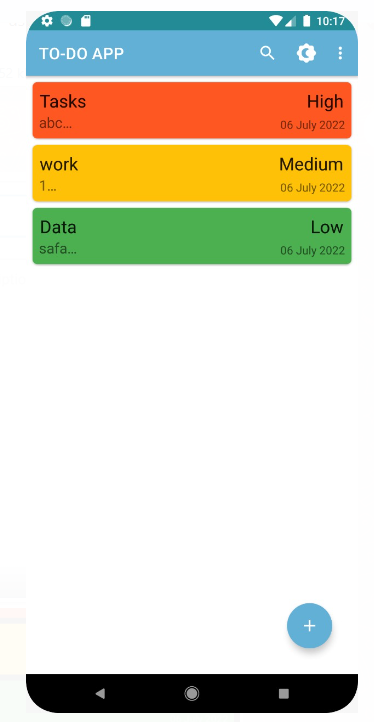
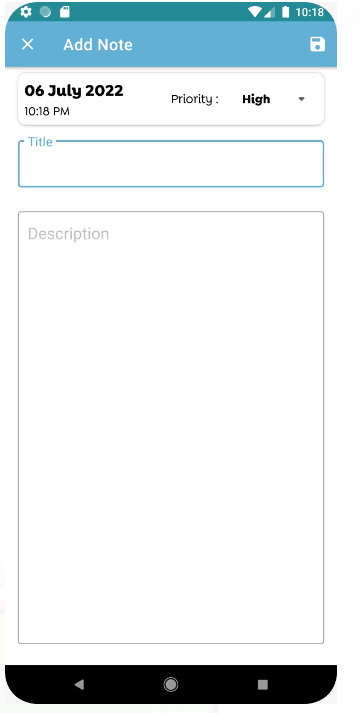


Fig 6.1.1: This the display page which the user can see when he/she visits the app. It displays the note added previously.



**NOTE\_ACTIVITY**

**ACTIVITY\_MAIN DARK MODE**

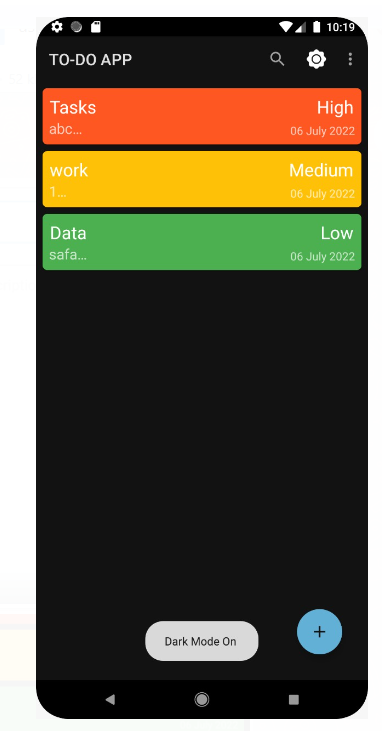
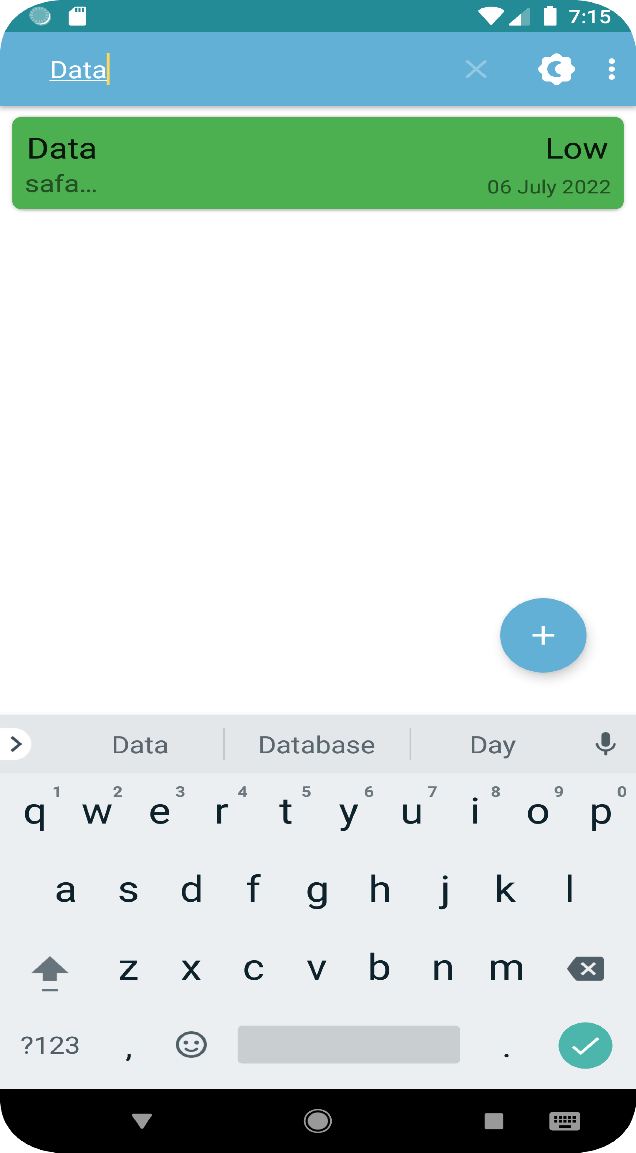


Fig 6.1.3: In this screen the user can set the priority, give a title and add the description

Fig 6.1.2: According to the users preference he/she can change screen into light or dark mode

****

**SEARCH SCREEN**

**SPLASH SCREEN**

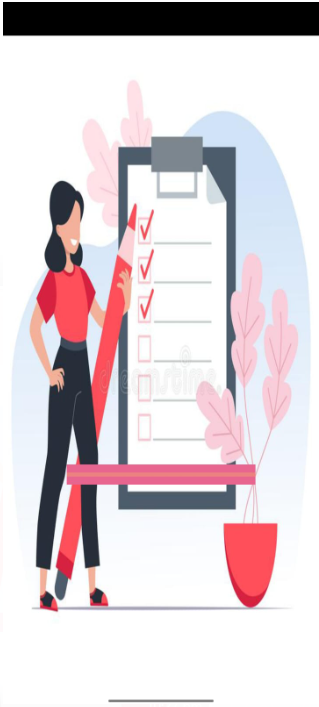


Fig 6.1.4: This is the front page that appears to the user when first open also called Splash Screen

Fig 6.1.5: This is the search bar where the title of the note is used to search the note

**NO DESCRIPTION OR TITLE ERROR SCREEN**

****

Fig 6.1.6: This screen displays toast message when no title or description is entered and note is tried to save

**CHAPTER-7**

**FUTURE WORK**

In the future, we may extend this project by adding extra features to our android app like,

• Adding image to the notes.

• Sharing the notes to all possible apps.

**CHAPTER-8**

**CONCLUSION**

Existing system involves manual intervention in the process of note taking, and high chances of loosing the note. Taking notes using pen and paper. Difficulty in availability of notes or paper at the moment, with noted papers going missing many a times. Recent advances have made it comparatively less tedious by taking notes digitally, yet tedious task to search for a particular note, hence the Note App is developed with the solution to all such problem. Note App is an android application for taking notes, in both light mode and dark mode. The user can create notes with ease. He /She can sort notes based on their priority. User can share the notes to others.

**BIBLIOGRAPHY**

**References**

* Overview Guides Reference for database connection, https://www.geeksforgeeks.org/how-to-view-and-locate-sqlite-database-inandroid-studio/
* <https://www.geeksforgeeks.org/how-to-build-a-simple-notes-app-in-android/>
* https://developer.android.com/docs