VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi: 590 018

A Mobile Application Development Mini Project report on

Notes App

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

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

By

Mr. Pukkalla Suraj Nageswararao (1AY20CS115)
Mr. Soumyajit Biswas (1AY20CS158)

Under the guidance
of

Mr. Ranjan G.
Asst Professor,
Department Of CS&E

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
ACHARYA INSTITUTE OF TECHNOLOGY
(Affiliated to Visvesvaraya Technological University, Belagavi)
2022-2023

ACHARYA INSTITUTE OF TECHNOLOGY
(Affiliated to Visvesvaraya Technological University, Belagavi)
Soladevanahalli, Bangalore – 560090

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Mobil e Application Development Mini Project entitled **"Notes App"** is a bona fide work carried out by **Pukkalla Suraj Nageswararao (USN-1AY20CS115)** and **Soumyajit Biswas (USN-1AY20CS158)** of 6th semester in partial fulfilment for the award of degree of **Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University, Belagavi**, during the year **2022-2023**. It is certified that all corrections/ suggestions indicated for internal assessments have been incorporated in the Report deposited in the departmental library. The Mini Project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the **Bachelor of Engineering Degree**.

Mr. Ranjan G.	**Dr Ajith Padyana**
Assistant Professor CS&E	**Head Of Department**
(Guide)	**(CS&E)**

Name of the examiners	**Signature with date**
1.	
2.	

ABSTRACT

This paper presents a Flutter notes app that allows users to add, delete, and edit notes. The app also includes a basic text editor with undo and search functionality. The app is designed to be simple and easy to use. Users can create new notes by entering a title and description. Notes can be edited by tapping on the title or description. The undo button can be used to reverse the last change made to a note. The search bar can be used to find notes that contain a specific keyword.

The app is built using the Flutter framework. Flutter is a cross-platform framework that allows developers to create apps that run on both Android and iOS devices. This makes the app accessible to a wider range of users. The app has been tested on a variety of devices and has been found to be stable and reliable. The app has also been tested with a variety of different input methods, including keyboards, mouse, and touch screens.

The app is a valuable tool for anyone who wants to keep track of their thoughts and ideas. The app is easy to use and can be customized to meet the needs of individual users. The app is also free and open source and is available on GitHub, which makes it a great option for users who want to save money or who want to contribute to the development of the app.

ACKNOWLEDGEMENT

We express our gratitude to our institution and management for providing us with good infrastructure, laboratory, facilities and inspiring staff whose gratitude was of immense help in completion of this seminar successfully.

We are very thankful to our Principal, **Dr. Rajath Hegde M M** for being kind enough to provide us an opportunity to work on a project in this institution.

Our sincere thanks to **Dr. Ajith Padyana,** Head of the Department, Computer Science and Engineering, Acharya Institute of Technology for his valuable support and for rendering us resources for this mini project work.

We express our gratitude to **Mr. Ranjan G. , and Mr. Srikanth SP** Assistant Professor, Dept. Computer Science and Engineering, Acharya Institute of Technology who guided us with valuable suggestions in completing this mini-project at every stage.

Our gratitude thanks should be rendered to many people who helped me in all possible ways.

Pukkalla Suraj Nageswararao (USN-1AY20CS115)

Soumyajit Biswas (USN-1AY20CS158)

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

Chapter 1

INTRODUCTION

1.1 Introduction to Android Studio

In recent times, Android became the world's most popular operating system for various reasons. Android Studio is an IDE for Google Android Development launched on 16th May 2013, during Google's I/O 2013 event. Android Studio contains all the Android tools to design, test, debug, and profile your application. The Android Studio uses Gradle to manage your project, a Build Automation Tool. Android Studio has many exciting features that can help you to develop your Android application like Powerful code editor with smart editing and code re-factoring, Emulator to show your code output in various resolutions, including Nexus 4, Nexus7, Nexus 10, and many other android phones, Gradle based build support, Maven Support , template-based wizards and also Dracula Theme Environment to enjoy your coding experience. Android Studio supports all the same programming languagesof IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go and Android Studio 3.0 or later supports Kotlin and all Java 7 language features and a subset of Java 8 language features that vary by platform version. External projects backport some Java 9 features. While IntelliJ states that Android Studiosupports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android. Once an app has been compiled with Android Studio, it can be published on the Google Play Store. The application has to be in line with the Google Play Store developer.

1.2 About the Project

The objective of this project is to make a flutter app to add, delete and edit note. The app also includes the basic text editor with undo, redo and search functionality. The app is designed to be simple and easy to use. Users can create new notes by entering the title and description. Notes can be edited by tapping on the title or description. The undo button can be used to resolve the last change made to a note. The search bar can be used to find notes that contains specific keywords.

The app is built using the flutter framework. Flutter is a clause plot platform framework

that allows developers to create apps that run on both Android and iOS device. This makes the app accessible to a wider range of users. The app has been tested on a variety of devices and has been found to be stable and reliable. The app has also been tested with a variety off input methods, including keyboards, mouse and touch screens.

The app is valuable tool for anyone who wants to keep track of their thoughts and ideas. The app is easy to use and can be customised to meet the needs of individual users. The app is also free and open source and is available on GitHub, which makes it a great option for users who want to save money or who want to contribute to the development of the app.

1.3 Project Applications

Notes app can have various applications and can be useful in many different scenarios. Here are some common application areas of a simple notes app:

1. Personal Organization: A notes app can help individuals stay organized by providing a convenient digital space to jot down ideas, reminders, to-do lists, and important information. Users can easily create, edit, and manage their personal notes, ensuring that important details are not forgotten.

2. Task Management: A notes app can serve as a task management tool, allowing users to create task lists, set priorities, and track progress. Notes can be used to outline project plans, assign deadlines, and record updates, making it easier to stay focused and accomplish goals.

3. Education and Learning: Students and learners can utilize a notes app to take lecture notes, jot down key points, summarize readings, and create study guides. The app can support multimedia attachments, allowing users to include images, diagrams, or audio recordings to enhance their learning experience.

4. Meeting and Collaboration: A notes app can be handy during meetings, enabling participants to take notes, capture action items, and record decisions or discussions.

Collaborative features can be incorporated, allowing multiple users to access and contribute to shared notes, facilitating effective teamwork and coordination.

5. Creative Inspiration: Artists, writers, and creative professionals can use a notes app to capture ideas, brainstorm concepts, and store inspirational content. It can serve as a digital sketchbook or a repository for collecting references, links, and images that fuel creativity and serve as a source of inspiration.

6. Travel Planning: A notes app can assist travelers in organizing trip details, such as flight itineraries, hotel reservations, sightseeing plans, and packing lists. Users can easily update their notes, access them offline, and have all the essential travel information in one place.

Chapter 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Hardware Requirements

- Tools: Android Studio (Software), Android Phone or Emulator.
- Platform: Windows 8 / Windows 10 / Windows 11.
- RAM: 8GB ram or more.
- Memory:8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)

2.2 Software Requirements

- Operating system: 64-bit Microsoft® Windows® 8/10/11.
- Front end: Flutter
- Back end: Flutter
- SDK : Android 5.0
- IDE: Android Studio

2.3 Functional Requirements:

Software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. It should only specify the external behavior of the system. Functional requirements in software engineering, a functional requirement defines a function of software system or its component. A function is described as a set of inputs behavior and outputs.

- **Note Creation**: Users should be able to create new notes by entering a title, adding content, and optionally attaching multimedia files like images or audio recordings.
- **Note Editing**: Users should have the ability to edit existing notes, allowing them to

modify the title, content, and attached multimedia files. The app should provide an intuitive interface for editing notes.

- **Note Deletion**: Users should be able to delete unwanted notes. Depending on the requirements, the app may implement a feature that moves deleted notes to a "Trash" or "Recycle Bin" from where they can be permanently deleted or restored.
- **Note Organization**: The app should offer options for organizing notes, such as sorting notes by date, title, or custom categories/tags. Users should be able to create folders or use labels/tags to categorize and search for specific notes easily.
- **Search Functionality**: The app should include a search feature that allows users to find notes based on keywords, titles, tags, or content. The search functionality should provide quick and accurate results.
- **Synchronization and Cloud Storage**: The app should support synchronization across multiple devices, ensuring that users can access their notes from any device connected to their account. It should also provide cloud storage capabilities to securely store and back up notes.
- **Offline Access**: The app should allow users to access and edit their notes even when they are offline.
- **Backup and Restore**: The app should have mechanisms in place to back up user data, allowing users to restore their notes in case of device loss, damage, or accidental deletion.

These functional requirements form the foundation of a simple notes app, providing users with The necessary features to create, edit, organize, and manage their notes efficiently.Additional requirements can be added based on specific use cases and user needs.

2.4 Non-functional Requirements:

Non-functional requirements define the qualities and characteristics of a software system, such as its performance, security, usability, and maintainability. Here are some non-functional requirements that can be applicable to a simple notes app:

- **Performance**: The app should be responsive and provide quick load times for creating, editing, and accessing notes. It should be able to handle a large number of notes without significant performance degradation.
- **Reliability**: The app should be reliable and minimize the risk of data loss or corruption. It should have mechanisms in place to handle unexpected errors or crashes gracefully, ensuring that user data is not compromised.
- **Security**: The app should implement appropriate security measures to protect user data, including notes and user credentials. This may involve data encryption, secure authentication mechanisms, and secure communication protocols.
- **Usability**: The app should have an intuitive and user-friendly interface, making it easy for users to create, edit, organize, and manage notes. The design should follow established usability principles and provide clear instructions and feedback to users.
- **Accessibility**: The app should be designed and developed with accessibility in mind, ensuring that users with disabilities can access and use the app effectively. It should conform to accessibility standards and guidelines, such as providing support for screen readers, keyboard navigation, and adjustable text sizes.
- **Scalability**: The app should be able to handle an increasing number of users and notes without significant degradation in performance. It should be scalable to accommodate growth and increasing demands on the system.
- **Cross-Platform Compatibility**: The app should be compatible with various platforms and operating systems to allow users to access their notes seamlessly across different devices, such as smartphones, tablets, and desktop computers.
- **Data Backup and Recovery**: The app should have mechanisms in place for regular data backups to prevent data loss. It should provide a reliable restore process to recover notes in case of accidental deletion or system failures.
- **Maintainability**: The app should be designed and developed in a way that facilitates easy maintenance and updates. It should use modular and well-structured code, adhere to coding standards, and have proper documentation to support future enhancements

and bug fixes.

These non-functional requirements help ensure that the simple notes app not only meets functional expectations but also delivers a high-quality user experience in terms of performance, security, usability, and maintainability.

Chapter 3

DESIGN

3.1 Data / Work Flow Diagram:

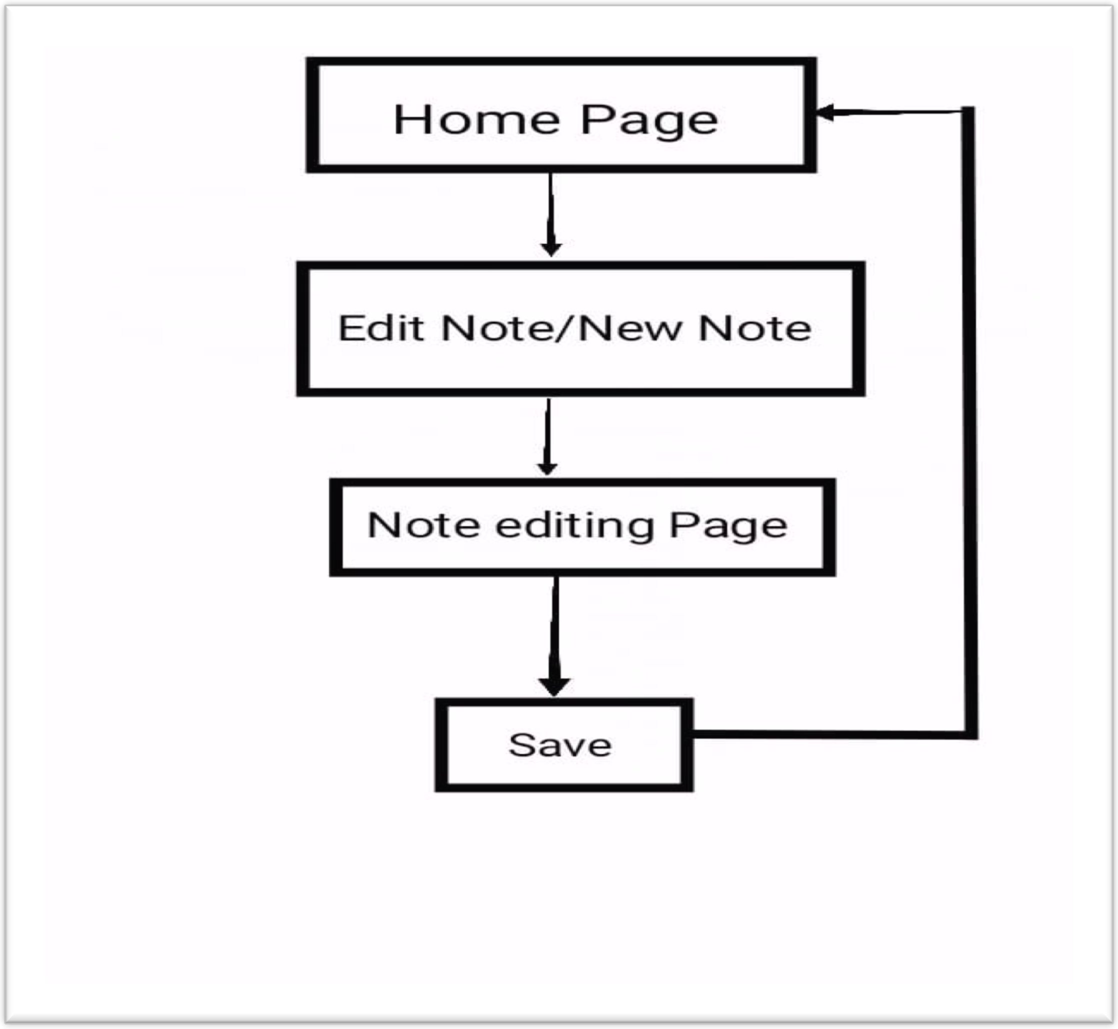

Figure 3.1: Flow representation

This diagram represents the work flow of the application. The process of creation, editing, deletion and save instructions.

The user enters to the home page after clicking on the app icon. The user will be provided.

With 2 options, 1. Add new note, 2. Edit note, and will be able to see all the notes existing in the app.

Then the user can decide whether to edit an existing note or to create a new note, followed by saving the note.

After saving the user will be back to the home screen and then he can either exit the app or do further modifications.

Chapter 4

IMPLEMENTATION

4.1 Source code:

4.1.1 main.dart (Code for starting the app)

```
import 'package:hive/hive.dart';

import 'package:hive_flutter/hive_flutter.dart';

import 'package:notes/pages/models/note_data.dart';

import 'package:flutter/material.dart';

import 'package:provider/provider.dart';

import 'pages/home_page.dart';


void main() async {

  //initialize hive

  await Hive.initFlutter();


  //open a hive  box

  await Hive.openBox('note_database');


  runApp(const MyApp());

}


class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.

  @override

  Widget build(BuildContext context) {

    return ChangeNotifierProvider(

      create: (context) => NoteData(),

      builder: (context, child) => const MaterialApp(



  debugShowCheckedModeBanner: false,

        home: HomePage(),

      ),

    );

  }
}
```

4.1.2 home_page.dart (Home page for the app that comes after we open it)

```
          import 'package:flutter/cupertino.dart';

import 'package:flutter/material.dart';

import 'package:provider/provider.dart';

import 'editing_note_page.dart';

import 'models/note_data.dart';

import 'models/note.dart';


class HomePage extends StatefulWidget {
 const HomePage({super.key});

 @override

 State<HomePage> createState() => _HomePageState();

}


class _HomePageState extends State<HomePage> {

 @override

 void initState() {

   super.initState();

   Provider.of<NoteData>(context, listen: false).initializeNotes();
 }

 //create a new note
 void createNewNote() {
   int id = Provider.of<NoteData>(context, listen: false).getAllNotes().length;
 //create a blank note
   Note newNote = Note(

   id: id,
     text: '',
   );
   //go to edit the note

 goToNotePage(newNote, true);
 }

 //go to note editing page
 void goToNotePage(Note note, bool isNewNote) {
   Navigator.push(
       context,
       MaterialPageRoute(
         builder: (context) => EditingNotePage(
           note: note,
           isNewNote: isNewNote,
         ),
       ));
 }

 //delete note
 void deleteNote(Note note) {
  Provider.of<NoteData>(context, listen: false).deleteNode(note);
 }

 @override
 Widget build(BuildContext context) {
   return Consumer<NoteData>(
     builder: (context, value, child) => Scaffold(
       backgroundColor: CupertinoColors.systemGroupedBackground,
            //floatingActionButtonLocation: FloatingActionButtonLocation.miniEndTop,
       floatingActionButton: FloatingActionButton(
         onPressed: createNewNote,

 elevation: 50,
         backgroundColor: Colors.grey[300],
         child: const Icon(

 Icons.add,
           color: Colors.grey,
         ),
       ),
       body: SingleChildScrollView(
      padding: const EdgeInsets.only(bottom: 70),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            //heading
            const Padding(
              padding: EdgeInsets.only(left: 25.0, top: 75.0),
              child: Text(
                'Notes',
                style: TextStyle(fontSize: 32, fontWeight: FontWeight.bold),
              ),
            ),

            //list of notes
            value.getAllNotes().length == 0
                ? Padding(
                    padding: const EdgeInsets.only(top: 50.0),
                    child: Center(
                      child: Text(
                        'Nothing Here...',
                        style: TextStyle(color: Colors.grey[400]),
                              ),
                    ))
                : CupertinoListSection.insetGrouped(

           children: List.generate(
                      value.getAllNotes().length,

 (index) => CupertinoListTile(
                          title: Text(value.getAllNotes()[index].text),
                          onTap: () => goToNotePage(
                            value.getAllNotes()[index], false),
         trailing: IconButton(
                 icon: Icon(Icons.delete),
                 onPressed: () => showCupertinoDialog(
                   context: context,
                   builder: (BuildContext ctx) {
                     return CupertinoAlertDialog(
                       title: const Text('Please Confirm'),
                       content: const Text(
                         'Are you sure you want to Delete the note?'),
                       actions: [
                         // The "Yes" button
                         CupertinoDialogAction(
                           onPressed: () {
                             setState(() {
                               //_isShown = false;
                               deleteNote(value
                                   .getAllNotes()[index]);
                               Navigator.of(context).pop();
                             });
                           },
                           isDefaultAction: true,
                           isDestructiveAction: true,
                           child: const Text('Yes'),
                         ),
                               // The "No" button
                         CupertinoDialogAction(

                   onPressed: () {

 Navigator.of(context).pop();
                           },
                         isDefaultAction: false,
                         isDestructiveAction: false,
                         child: const Text('No'),
                        )
                       ],
                      );
                     }),
                   //deleteNote(value.getAllNotes()[index]),
                  ),
                 ))),
       ],
      ),
     ),
    ),
   );
  }
 }
```

4.2 Testing:

Test No.	Test case	Expected Output	Observed Output	Result
1.	Add new note functionality, when plus symbol is clicked in home page	A new page for adding note should open.	A new page for adding note opens.	Test case passed
2.	Edit option after Opening already existing note from home page	The opened note should be allowed to be edited.	The opened note is allowed to be edited.	Test case passed
3.	Search functionality, on clicking search icon in the editing page	String should be searched, if present show number of matches else show 0 matches.	String should is searched, if present shows number of matches else shows 0 matches.	Test case passed
4.	Undo functionality	The changes should be reversed one by step.	The changes are being reversed by one step.	Test case passed
5.	Redo functionality	The changes reversed should be undone by one step.	The changes reversed are being undone by one step.	Test case passed
6.	Save functionality	The new note or the changes made should be saved.	The new note or the changes made are being saved.	Test case passed
7.	Deletion functionality	The note should be deleted.	The note is being deleted.	Test case passed

Table 4.2.1 Test Cases

Chapter 5

RESULTS AND DISCUSSION

In this section we will be seeing some of the snapshots of the working of the app

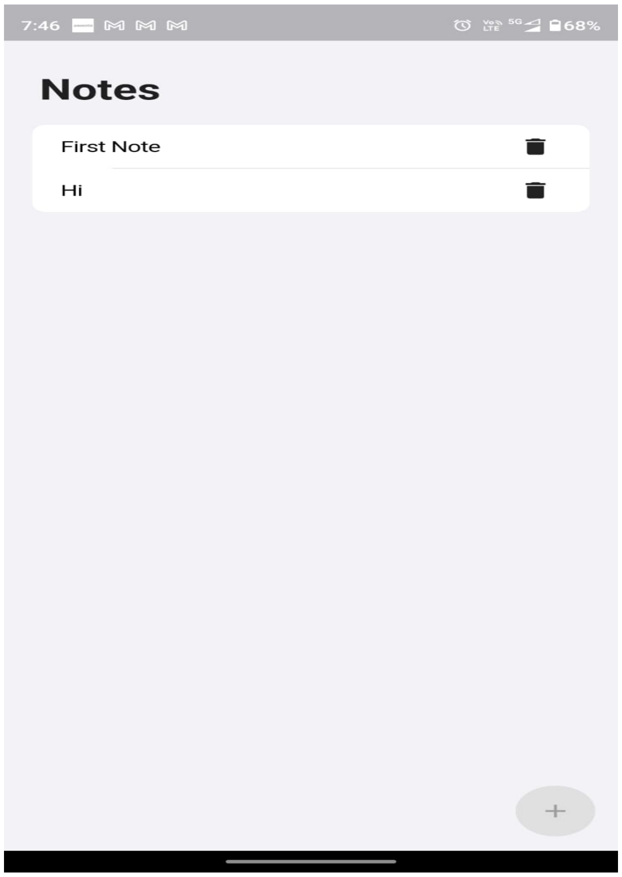

Figure 5.1 : Home page

Figure 5.1 shows the landing or home page after clicking on the icon of the app. We can see the plus symbol which will add a new note, trash bin symbol which is used to delete an existing note. The user can click on the existing note to edit it.

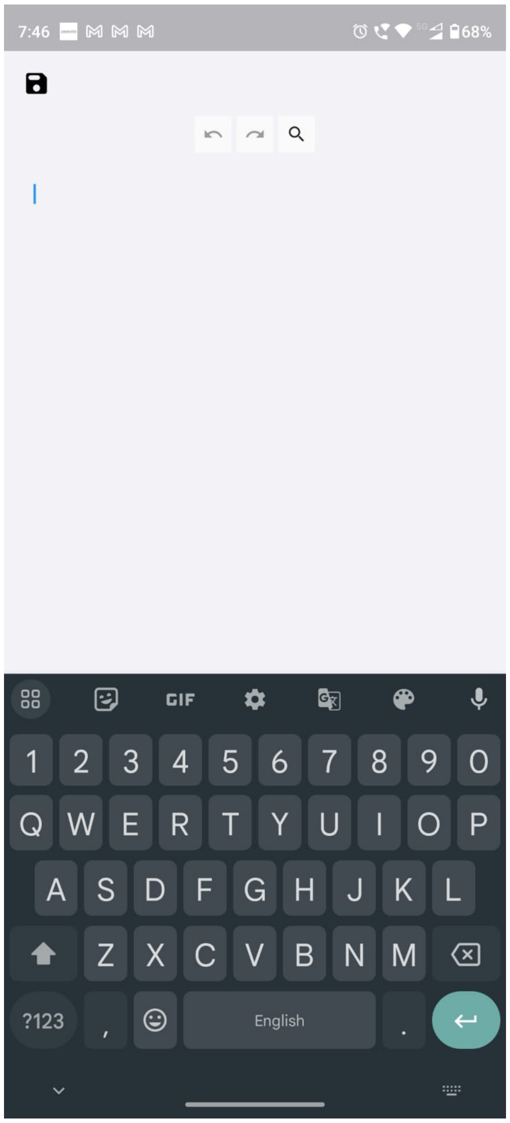

Figure 5.2: Add note page

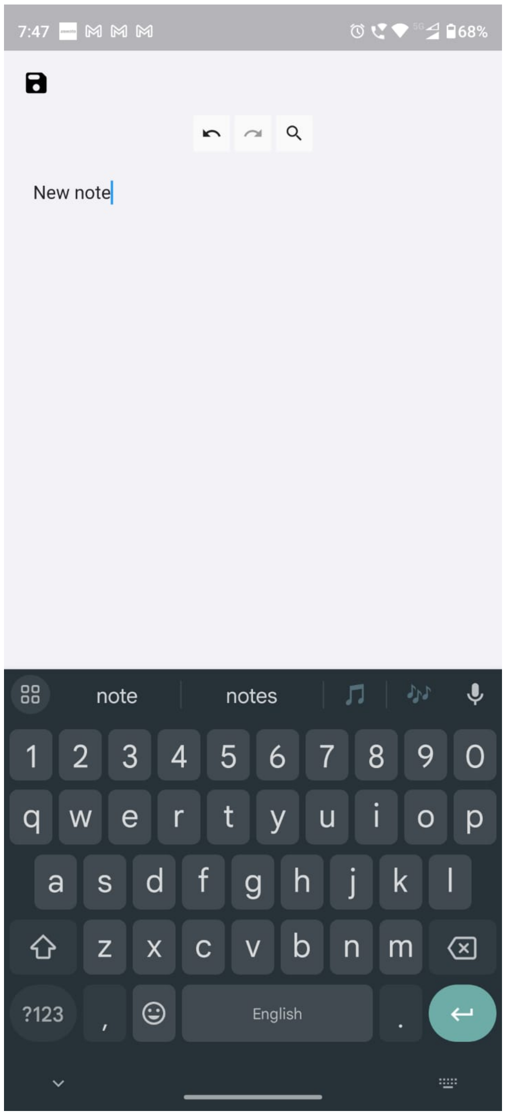
New note

Figure 5.3: Addition of text in the note

Figure 5.2 is the page we get after pressing the plus symbol, i.e. add a new note page. This will add a new note.

Figure 5.3 shows the text being added in the new note

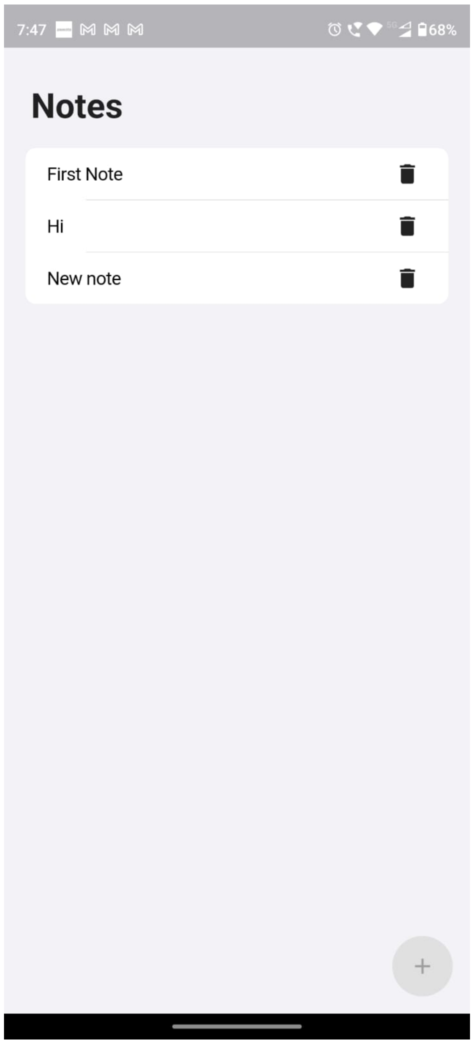

Figure 5.4: Before deletion of note

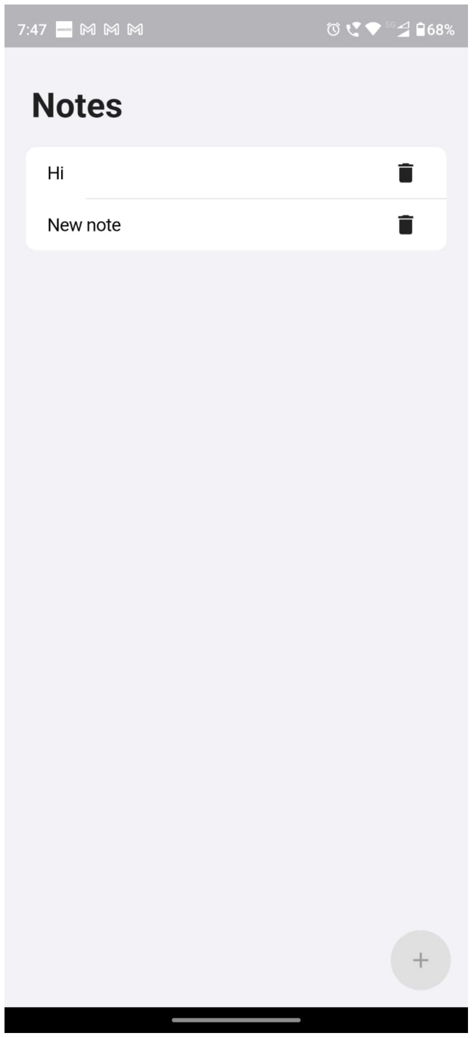

Figure 5.5: After deletion of note 'First Note'

The figure 5.4, shows a snapshot of home page after the new note has been added.

The figure 5.5, shows a snapshot of home page after the note named 'First Note' was deleted. We can see that only 2 notes exist in the home page now, the deletion operation was carried out by pressing the trash button.

CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, the development of a simple notes app with features for creating, editing, organizing, and managing notes is a valuable undertaking. Such an app serves as a versatile tool with numerous applications across personal, educational, professional, and creative domains. By fulfilling the functional requirements of user-friendly note creation, editing, deletion, organization, search, synchronization, and backup, the app provides users with a streamlined and efficient note-taking experience.

Additionally, by addressing non-functional requirements such as performance, reliability, security, usability, accessibility, scalability, cross-platform compatibility, data backup and recovery, privacy, and maintainability, the app ensures a high-quality and reliable user experience. These non-functional requirements contribute to the app's overall performance, security, ease of use, accessibility for all users, compatibility across devices, and ability to handle increasing demands.

In summary, the development of a simple notes app provides users with a digital space to store, manage, and access their notes conveniently. The app's functional and non-functional features work in harmony to deliver an intuitive, secure, and reliable user experience. Whether for personal or professional use, the simple notes app becomes an indispensable tool that enhances organization, productivity, and creativity in the fast-paced digital age.

Future Enhancements:

Here are some potential future enhancements that can be considered for a simple notes app:

1. Collaboration Features: Introduce real-time collaboration capabilities, allowing multiple users to simultaneously edit and view notes. This would facilitate teamwork, brainstorming sessions, and group projects.
2. Advanced Formatting Options: Expand the app's note editing capabilities by incorporating advanced formatting options such as font styles, colors, bullet points, numbering, tables, and rich media embedding. This would enable users to create visually appealing and structured notes.

3. Voice-to-Text Conversion: Implement voice recognition technology to enable users to dictate their notes, which will be automatically converted to text. This feature would enhance accessibility and convenience, especially for users on-the-go or those with limited typing abilities.
4. Reminders and Task Management Integration: Integrate the app with task management platforms or calendar apps to enable users to set reminders, deadlines, and prioritize notes as tasks. This integration would facilitate seamless organization and productivity.
5. Dark Mode and Theming Options: Offer customizable themes, including a dark mode option, to provide users with personalization choices and accommodate different preferences and environments.
6. Cross-Device Clipboard Sync: Enable a clipboard sync feature that allows users to copy and paste content across multiple devices. This would streamline the transfer of information between devices and enhance productivity.

These future enhancements can elevate the functionality, versatility, and user experience of the simple notes app, providing users with more advanced features, improved productivity, and seamless integration with other tools and platforms.

BIBLIOGRAPHY

[1] Erik Hellman, “Android Programming – Pushing the Limits”, 1st Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197

[2] Dawn Griffiths and David Griffiths, “Head First Android Development”, 1st Edition, O‟Reilly SPD Publishers, 2015. ISBN-13: 978-9352131341

[3] Bill Phillips, Chris Stewart and Kristin Marsicano, “Android Programming: The Big Nerd Ranch Guide”, 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054

[4] Data from Global Academy of Technology

[5] http://stackoverflow.com

[6] https://www.w3schools.com

[7] http://www.javapoint.com

[8] https://www.bootply.com

[9] https://www.tutorialspoint.com

[10] http://youtube.com