Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2023), B.Sc. in CSE (Day)

Authentication Notebook Application

Course Title: Mobile Application Development
Course Code: CSE 426
Section: 201 D3

Students Details

Name	**ID**
Saiful Islam	173002049
MD Mahinul Alam	201002257

Submission Date: 18-06-2023
Course Teacher's Name: Dr. Nazib Abdun Nasir

[For teachers use only: Don't write anything inside this box]

Lab Project Status

Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

The Authentication Notebook Android application that allows users to login to the app using their email and password. It uses Firebase Authentication tp users and redirects them to the notesactivity class once they are authentic. If a user has not verified their email, they will not be able to login. Here, Using XML layout file for login screen, details screen, edit screen, password forgot screen. The layout consists of a Relative layout, Image layout, Linear Layout with various child views such as text views, edit text fields, and buttons.

1.2 Motivation

The motivation authentication notebook app could be to provide a platform for users to store their notes, ideas, or any other important information in one place. Additionally, this app could provide a secure way for users to store their information and access it whenever needed.

1.3 Problem Definition

The problem being addressed is the need for a secure platform for users to store their important information. With the increasing use of smartphones, people tend to store their notes on their devices, which may not be secure. This app aims to provide a secure platform for users to store their notes and access them whenever needed. [1]

1.3.1 Problem Statement

The problem statement of this code is to implement a login system for an Android notes app using Firebase authentication. The app allows users to log in with their email and password and provides functionality for signing up and resetting forgotten passwords. The motivation behind this code is to provide a secure and easy-to-use login system for users of the notes app. The complex engineering problem involves integrating Firebase authentication with the Android app and handling user authentication and verification in a secure and efficient manner.

1.4 Design Goals/Objectives

1.4.1 Design

The authentication notebook application code is an implementation of the MainActivity class in an Android application using Java and the Firebase authentication API. The design consists of a login screen with an email and password field, a login button, a "forgot password" button, and a "go to sign up" button. The activity also contains a progress bar that becomes visible during the login process.

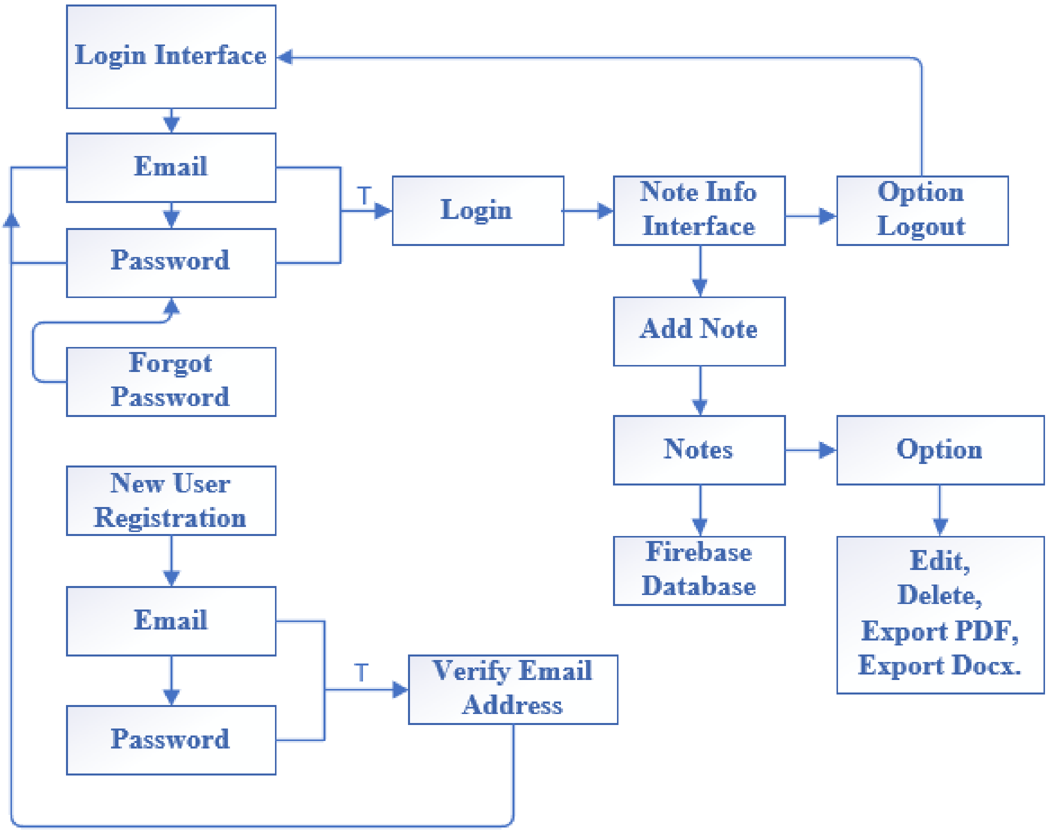


Figure 1.1: Authentication Notebook Application Model

1.4.2 Activity Design

Figure: 1.2 is the activity design of authentication notebook application.

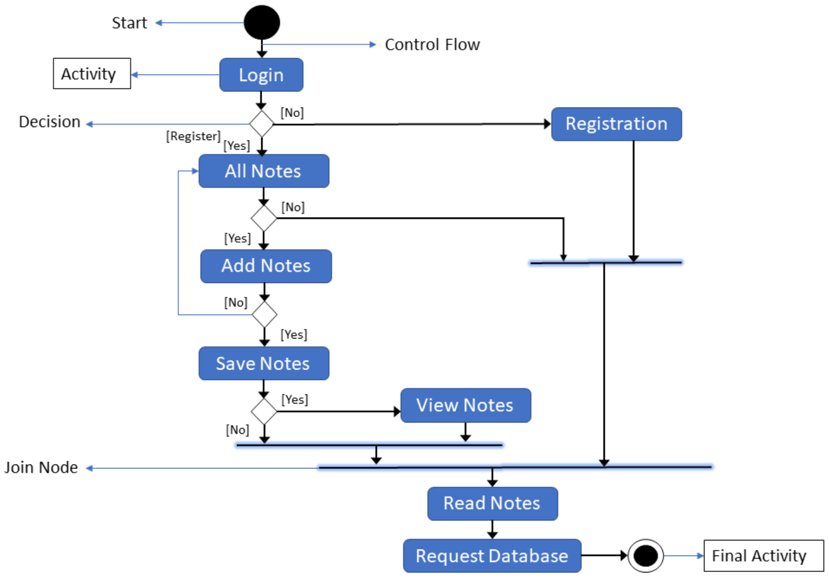


Figure 1.2: Activity Design Authentication Notebook Application

1.4.3 Goal Objective

The goal of this code is to provide a login functionality for an Android application. The user enters their email and password, and if the information is correct, they are directed to the notesactivity screen. If the email is not verified, the user is asked to verify their email before proceeding. The "go to sign up" button directs the user to the sign-up screen, and the "forgot password" button allows the user to reset their password. The progress bar indicates to the user that the login process is ongoing.

1.5 Application

The application being developed is a authentication notes application for Android. The MainActivity class is responsible for handling the login functionality, allowing users to sign in to their accounts and access their notes. The notesactivity class, which is not shown in the provided code, would handle the creation, editing, and deletion of notes. The signup class and fogotpassword class, which are also not shown, would handle the sign-up and password reset functionalities, respectively. Overall, the application allows users to manage their notes and keep them secure with user authentication. Working progress running so I did not attach screenshot.

Chapter 2

Design/Development/Implementation of the Project

2.1 Project Details

The Authentication Notes App is an Android application that allows users to create and manage notes. The app uses Firebase for user authentication and Firestore as a cloud-based NoSQL database for storing and retrieving notes.

Key Features:

(1) User Registration and Login:

- Users can create a new account by providing their name, email, and password.
- Existing users can log in using their registered email and password.

(2) Note Creation and Management:

- Users can create new notes by entering a title and content for each note.
- The app provides a user-friendly interface for managing and organizing notes.
- Notes are stored in the Firestore database, allowing users to access their notes from multiple devices.

(3) Email Verification:

- After successful registration, a verification email is sent to the user's provided email address.
- Users need to verify their email to ensure the security and validity of their account.

(4) User Interface:

- The app features a clean and intuitive user interface, allowing users to navigate and interact with ease.
- Custom layouts and UI elements are used to enhance the user experience.

(5) Firebase Integration:

- Firebase Authentication is utilized for user registration, login, and email verification.
- Firebase is used as a real-time cloud database for storing and retrieving notes.

2.2 Implementation

MainActivity.java:

- This file contains the code for the main activity of the Notes App.
- It handles user authentication using Firebase Authentication.
- It includes methods for user login, registration, and password reset.
- The onCreate method sets the layout and initializes the Firebase Authentication instance.

NotesActivity.java:

- This file represents the activity that displays the user's notes.
- It retrieves the user's notes from the Firestore database and displays them in a RecyclerView.
- The onCreate method sets the layout, initializes the Firebase Firestore instance, and retrieves the user's notes.
- It includes a method for deleting a note.

NoteDetails.java:

- This file represents the activity that displays the details of a specific note.
- It retrieves the note's details from an intent and displays them.
- The onCreate method sets the layout, initializes the UI elements, and sets a click listener for the "Edit" button.
- It includes a method for setting a random background image for the note.

ForgotPassword.java:

- This file represents the activity for password recovery.
- It allows the user to enter their email to receive a password reset email.
- The onCreate method sets the layout and initializes the Firebase Authentication instance.
- It includes a click listener for the "Go back to login" button and the "Password Recover" button.

FirebaseModel.java:

- This file defines the data model for a note.
- It includes a constructor and getter/setter methods for the title and content of a note.

EditNoteActivity.java:

- This file represents the activity for editing a note.
- It allows the user to edit the title and content of a note and save the changes to the Firestore database.
- The onCreate method sets the layout, initializes the UI elements, and retrieves the note's details from an intent.
- It includes a click listener for the "Save" button.

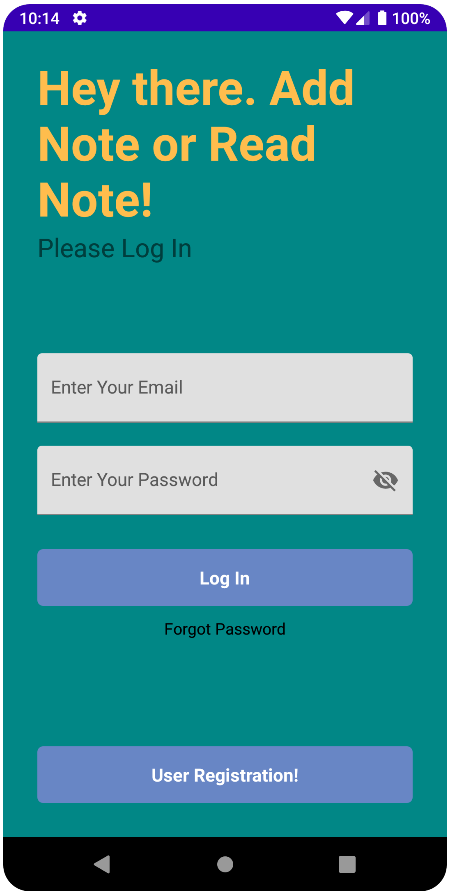
CreateNote.java:

- This file represents the activity for creating a new note.
- It allows the user to enter a title and content for the new note and save it to the Firestore database.
- The onCreate method sets the layout, initializes the UI elements, and includes a click listener for the "Save" button.

Signup.java:

- This file represents the activity for user registration.
- It enables users to register by providing their name, email, and password.
- The registration process is handled using Firebase Authentication.
- Upon successful registration, a verification email is sent to the user.
- The activity includes methods for sending the verification email and handling user interface interactions.

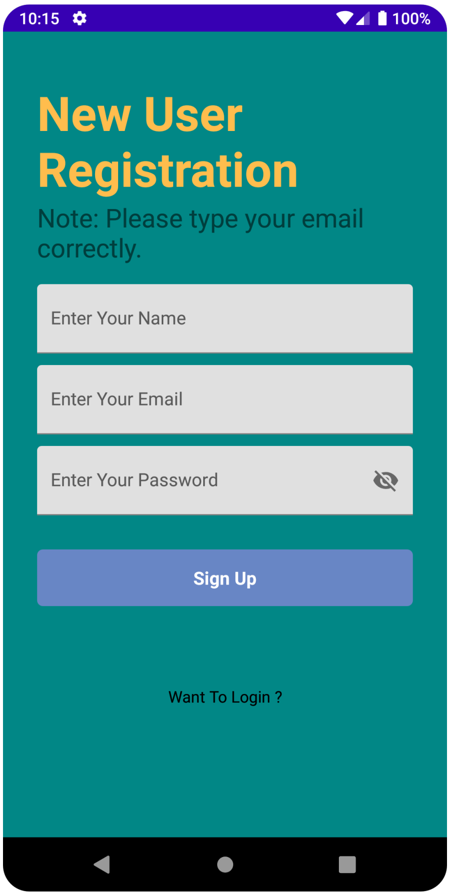
2.3 Application Interface



Hey there. Add Note or Read Note!

Please Log In

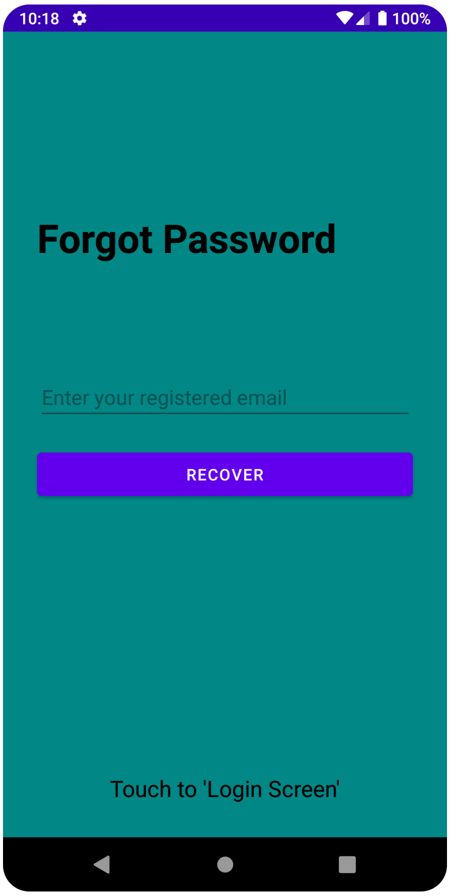
Figure 1



New User Registration

Note: Please type your email correctly.

Figure 2



Forgot Password

Figure 3

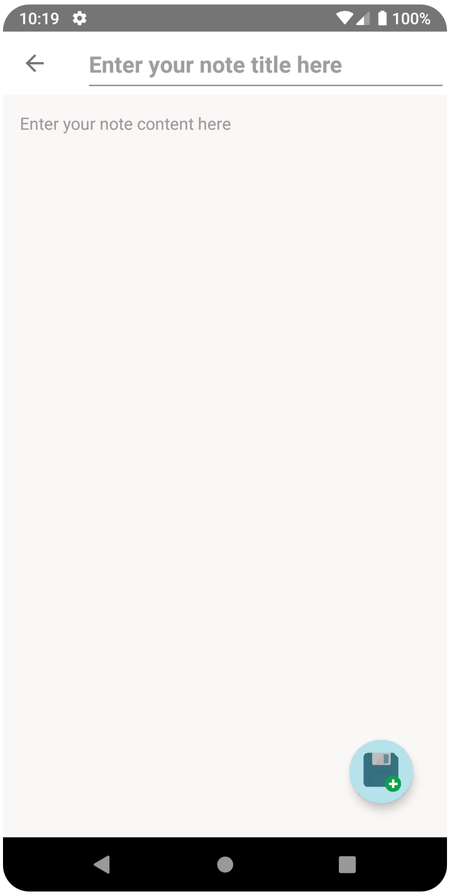
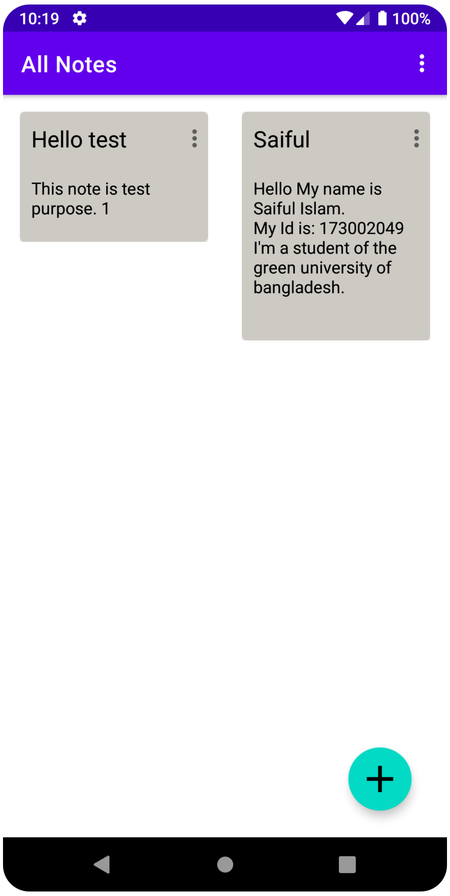


Figure 4



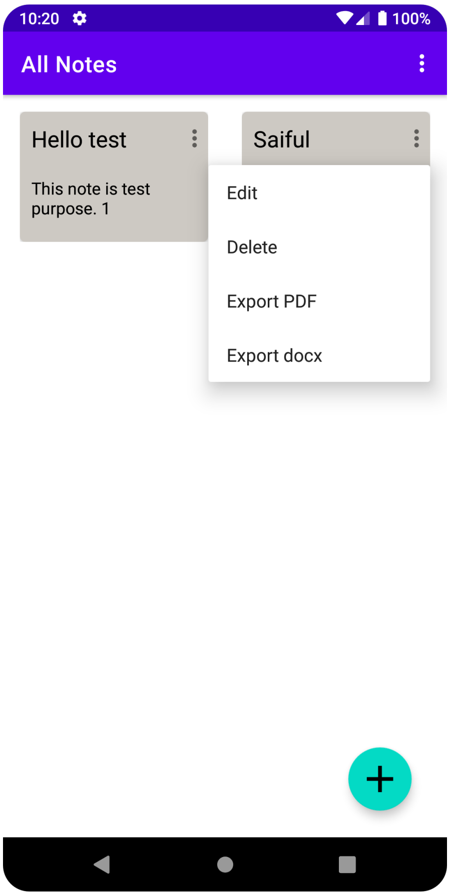
Hello test

This note is test purpose. 1

Saiful

Hello My name is Saiful Islam.
My Id is: 173002049
I'm a student of the green university of bangladesh.

Figure 5

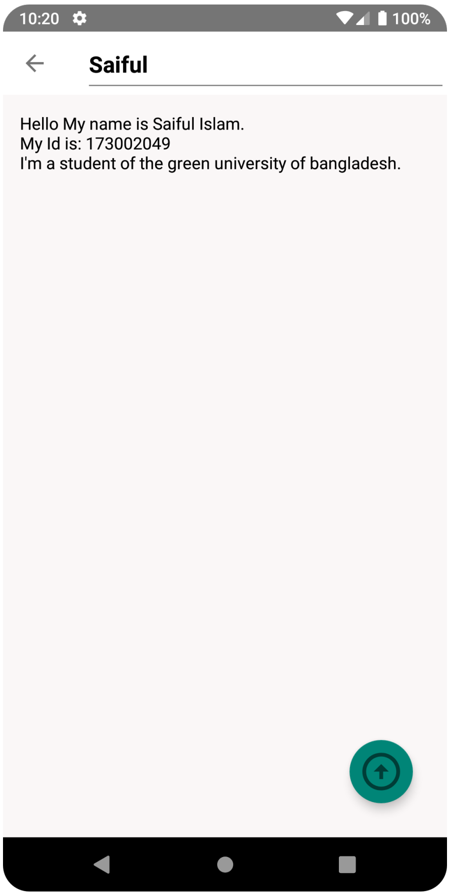


Hello test

This note is test purpose. 1

Saiful

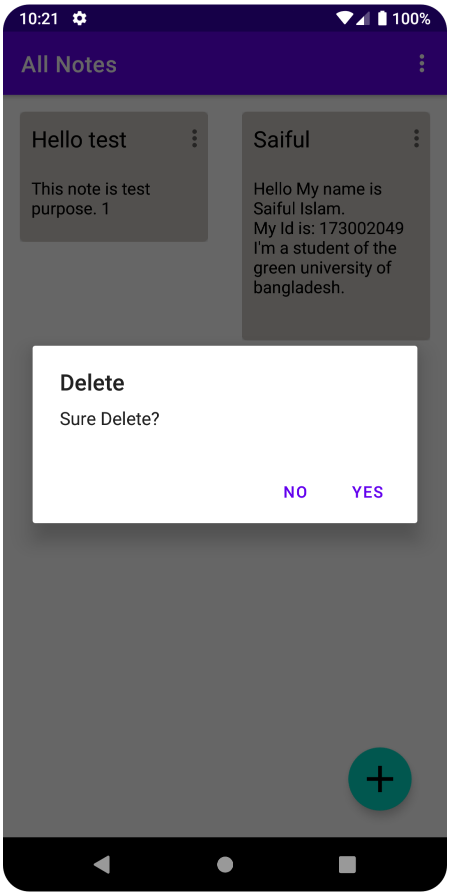
Figure 6



Saiful

Hello My name is Saiful Islam.
My Id is: 173002049
I'm a student of the green university of bangladesh.

Figure 7



Hello test

This note is test purpose. 1

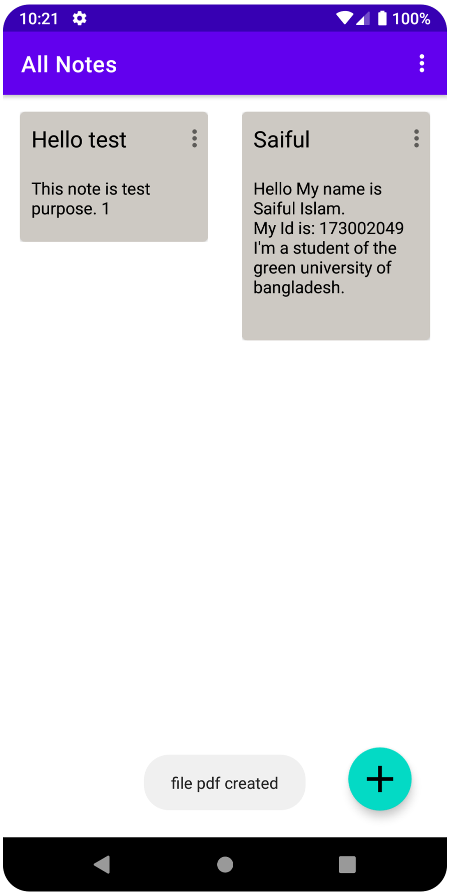
Saiful

Hello My name is Saiful Islam.
My Id is: 173002049
I'm a student of the green university of bangladesh.

Delete

Sure Delete?

Figure 8



10:21 100%

All Notes

Hello test

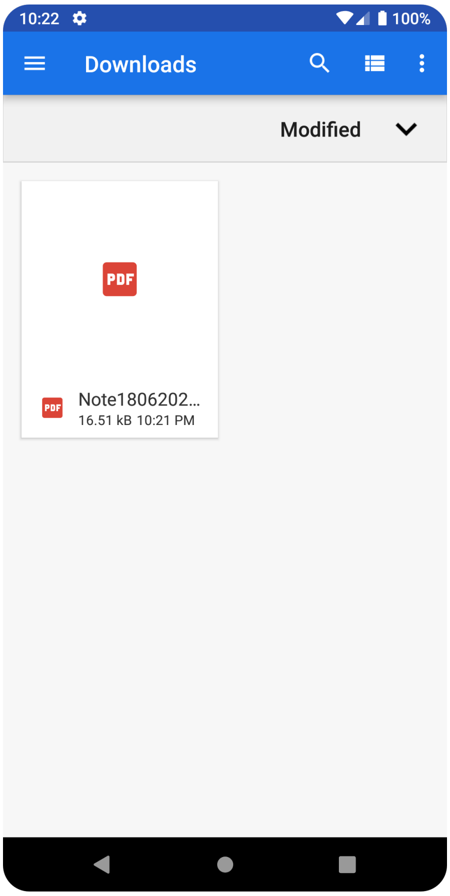
This note is test purpose. 1

Saiful

Hello My name is Saiful Islam.
My Id is: 173002049
I'm a student of the green university of bangladesh.

file pdf created

Figure 9



10:22 100%

Downloads

Modified

Note1806202...
16.51 kB 10:21 PM

Figure 10

Figure 2.1: The Figure 1 to 9 are Application Interface and Figure 10 are Output.

Chapter 3

Conclusion

3.1 Discussion

Authentication Notebook Android application that allows users to sign up and log in to access a note-taking feature. The app utilizes Firebase Authentication to register and authentic users. The code includes functions to check if the user's email is verified, and if not, prompts the user to verify their email before logging in, and store their notes on Firebase cloud storage.

3.2 Limitations

It is difficult to identify potential limitations. However, some general limitations of the code may include:
Lack of input validation: Authentication notes app code only checks if the email and password fields are empty and whether the password length is greater than 7 digits. It is recommended to add more input validation to ensure the user provides valid and secure input.
Lack of scalability: Authentication notes app code is a simple implementation of Firebase Authentication, which may not be scalable for larger applications. Additional security measures, such as implementing two-factor authentication, may need to be added as the app scales.

3.3 Scope of Future Work

In future work, the code could be extended to include the Sharing option. The user interface could be improved with better design elements and improved usability.

3.4 Complex Engineering Problem

The following Table 3.1 must be completed according to your above discussion in detail.

Table 3.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributes	Explain how to address
P1: Depth of knowledge required	Java programming (syntax, data types, control structure), Android development (Android application architecture, activity lifecycle, user interface components, and storage options), User interface design (easy to navigate), Data storage (SQLite databases, Cloud Storage), Security (secure authentication).
P2: Range of conflicting requirements	Offline vs. Online: The application needs to offer online access to notes synchronize them with the cloud, but it's not allow users to access their notes offline.
P3: Depth of analysis required	The analysis required is also relatively simple, as it involves verifying user credentials and checking their email verification status.
P4: Familiarity of issues	Email verification status issues involved are also relatively familiar, as the app uses a common authentication service and interface.
P5: Extent of applicable codes	These are not likely to be extensive.
P6: Extent of stakeholder involvement and conflicting requirements	—-
P7: Interdependence	The interdependence of authentication notebook app with other systems or components is also likely to be limited.

References

[1] https://securitygladiators.com/security/software/note/.