

DATABOOK - Data Maintaining Application

A Project Report

Submitted in partial fulfillment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (COMPUTER SCIENCE)

By

Gautam Chandrakant Mandaliya

Seat Number – 516

Under the esteemed guidance of

Mrs. Niramaye Deshpande

Assistant Professor



NAGINDAS KHANDWALA COLLEGE (Autonomous)

(Affiliated to University of Mumbai)

MUMBAI, 400 064

MAHARASHTRA

2022-23

(Original Copy of the Approved Proforma of the Project Proposal)

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

PNR No.:

Roll No.: **516**

1. Name of the Student: Mr. Gautam Chandrakant Mandaliya
2. Title of the Project: DATABOOK - Data Maintaining Application
3. Name of the Guide: Asst. Prof. Niramaye Deshpande
4. Teaching/Industry experience of the Guide: 16 years
5. Is this your first submission? Yes No

Signature of the Student:

Signature of the Guide:

Date:

Date:

Signature of the Coordinator:

Date:

NAGINDAS KHANDWALA COLLEGE (Autonomous)

(Affiliated to University of Mumbai)

MUMBAI, 400 064

MAHARASHTRA

**DEPARTMENT OF COMPUTER AND INFORMATION
SCIENCE**



CERTIFICATE

This is to certify that the project titled, "**DATABOOK - Data Maintaining Application**", is bonafied work of **Mr. GAUTAM CHANDRAKANT MANDALIYA** bearing Seat No: **(516)** submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in COMPUTER SCIENCE from University of Mumbai.

Internal Guide

Coordinator

External Examiner

Date:

College Seal

Abstract

“DataBook” is an Android Application for maintaining a user’s daily data and also help user’s to have a balance between personal and professional life. In this project I’ve tried to make an application that helps people to keep their daily data at one place and also to help them to maintain their data on daily basis. In order to provide users one platform (application) where they can carry everywhere instead of carrying dairy everywhere and then opening it to maintain data, users can easily take out the cell-phone, open the application and can note down important tasks and meeting which they have to attend, their daily spending and earnings in the money tracker activity. The main aim of the project is to help people to have a place where they can maintain their daily plans and follow-up the things on daily basis without having anything to carry.

This project was developed with the help of Android Studio and the languages used are Java and XML with Firebase Database which is used to save user data and make it available as per their need. This project have overcome many challenges such as gathering information knowing the users requirements, how to make friendly UI so that users are happy to use it on daily basis. The system is then designed in accordance with specifications to fulfill the requirements. The application has all the validation, updating and saving the data in real-time scenario and also to save changes once the work has been done. Users need to just create an account, enter the required credentials and application is good to go.

ACKNOWLEDGEMENT

Apart from my efforts, the success of my project depends on the encouragement and guidelines of my professors. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project. It would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I would like to acknowledge the enormous help given to me in creating this project. I am grateful to the Director of M.K.E.S Dr. Mrs. Ancy Jose, Principal Prof. Dr. Moushumi Datta, Vice Principal of Self-Finance Dr. Mona Mehta, Co-coordinator of Department of Computer and Information Science Dr. Sindhu P.M., for their kind cooperation in the completion of my project.

I am highly indebted to my Project Guide – Asst. Prof. Niramaye Deshpande for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support and suggestion in completing the project.

I would also like to thank all the faculties who helped me in choosing the project and assist me with what technology I need to use and how the project should proceed further. The project has meant a lot to me, the bonding and the exercises I had with my professors was truly amazing. All the professors have been really appreciative and cheerful. This has been a joyful, uplifting, and an excellent educational experience. It is truly an honor being a student of Nagindas Khandwala College. The faculties of our college are steadfast, and studying under them is truly an enriching process.

As this is my last year of college, I always wanted to have such bonding with all the department members and my peers as well. As the memories are the only thing, we'll be able to cherish in the future. I would also like to thank my friends, family and other benefactors who have been a constant support throughout the development of the project. I wish to thank my parents as well for their undivided support and interest who inspired me and encouraged me to go my own way.

DECLARATION

I hereby declare that the project entitled, “**DATABOOK - Data Maintaining Application**” done at **Nagindas Khandwala College Malad (West)**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (COMPUTER SCIENCE)** to be submitted as final semester project as part of our curriculum.

Mr. Gautam Chandrakant Mandaliya

TABLE OF CONTENTS

Sr. No.	TITLE	Page No.
1	Chapter 1: INTRODUCTION 1.1 Background 1.2 Objectives 1.3 Applicability	1 - 2
2	Chapter 2: GAP ANALYSIS / DRAWBACK OF EXISTING SYSTEM	3
3	Chapter 3: REQUIREMENT AND ANALYSIS 3.1 Problem Definition 3.2 Requirement Specification 3.3 Planning and Scheduling 3.4 Software and Hardware Requirements 3.5 Preliminary Product Description	4 - 6
4	Chapter 4: SYSTEM DESIGN 4.1 Basic Modules 4.2 Schema Design 4.3 UML Diagrams 4.4 User Interface Design 4.5 Security Issues	7 - 18
5	Chapter 5: IMPLEMENTATION AND TESTING 5.1 Code 5.2 Testing Approach and Test Cases	19 – 30

6	Chapter 6: RESULTS AND DISCUSSION 6.1 Test Reports 6.2 User Documentation	31 - 32
7	Chapter 7: CONCLUSIONS 7.1 Conclusion 7.1.1 Significance of the System 7.2 Limitations of the System 7.3 Future Scope of the Project	33
	REFERENCES	34

List of Tables

Table Number	Table Name	Table Page Number
1.	Schema Design	7, 8
2.	Login Test Cases	27
3.	Register Test Cases	28, 29

List of Figures

Figure Number	Figure Name	Figure Page Number
1.	Gantt Chart	5
2.	Entity Relationship Diagram	9
3.	Class Diagram	9
4.	Use Case Diagram	10
5.	Activity Diagram	10
6.	State Machine Diagram	11
7.	Component Diagram	11
8.	Sequence Diagram	12, 13

CHAPTER 1: INTRODUCTION

1.1 Background

The main objective of the project is to develop an application for users. In the present scenario, users are manually saving their daily plans and also saving their expenses and income in the dairy or somewhere else. So I wanted to save daily plans and activities to perform on a daily basis, so I used to manually write them in a book and carry it everywhere, but now I want to develop an application for everyone where they can have trust in the system, save their important daily activities, keep track of their money, and be stress-free by simply entering the data on their phones. The services of the application are quite simple: create an account, enter the record, follow the activities, and have a balanced life.

1.2 Objectives

The main objective of the application is to maintain the data of the users in one place through activities such as adding notes, managing tasks, and keeping track of money by entering expenses and income. Just enter the required details, select the date, and save it to your phone. This will not immediately help the users in maintaining data, but in the long term, this will help them track their activities and have a way to maintain the data.

- Helps you stay organized and financially stable by keeping track of all your money and expenses.
- Helps you manage your time more effectively, and increase your productivity.
- Make a review of money and activities on a monthly basis.
- To maintain the data on a daily basis.
- Saving the important notes.
- Save tasks by date.

1.3 Applicability

The best applicability of this application is for those users who like to maintain their data and use it on a daily basis.

- Easily applicable for those who want to have a planner application and work according to their work life while finishing the work ahead of deadlines.
- The project's scope is for users who have a habit of keeping track of their time and money and working on time in order to live a balanced life.
- Applications will help you improve your financial and time management skills, and lead a more organized and productive life.

CHAPTER 2: GAP ANALYSIS / DRAWBACK OF EXISTING SYSTEM

There is no such application available in the market with all three features, such as note saving, task management, and money tracking. So users need to download three different applications to engage in such activities. This application will be free to use for all users, and all the other applications available in the market will charge them after a one-month free trial.

Drawbacks and missing features in the existing application:-

- The applications available don't have features for all three activities.
- Don't have the feature of manually saving the income and expense data.
- After one month of free trial, they will be charged.
- Some applications don't have proper security for data.
- Some have glitches in the application and can't resolve them.

CHAPTER 3: REQUIREMENT AND ANALYSIS

3.1 Problem Definition

The Data Maintaining Application is rarely found on the internet. The existing application that has been found contains drawbacks such as not properly maintaining data or not properly functioning. Also, the application that is available doesn't have all three features that my application will provide and also doesn't have proper timing for the task. The application has one more major drawback: it has a 30-day free trial, but if the users want to use it, they have to pay for it, and the cost ranges from \$20 to \$30 per month. So I thought of developing an application that has all the features at a minimum cost for a long-term scenario. Also, the GUI of the existing system lacks in terms of a user's connections and interactions with the system. If the user interface is not user friendly, the chances of the application being deleted increase. Interacting with users and understanding their needs is paramount, and developers must provide them with what they require.

3.2 Requirement Specifications

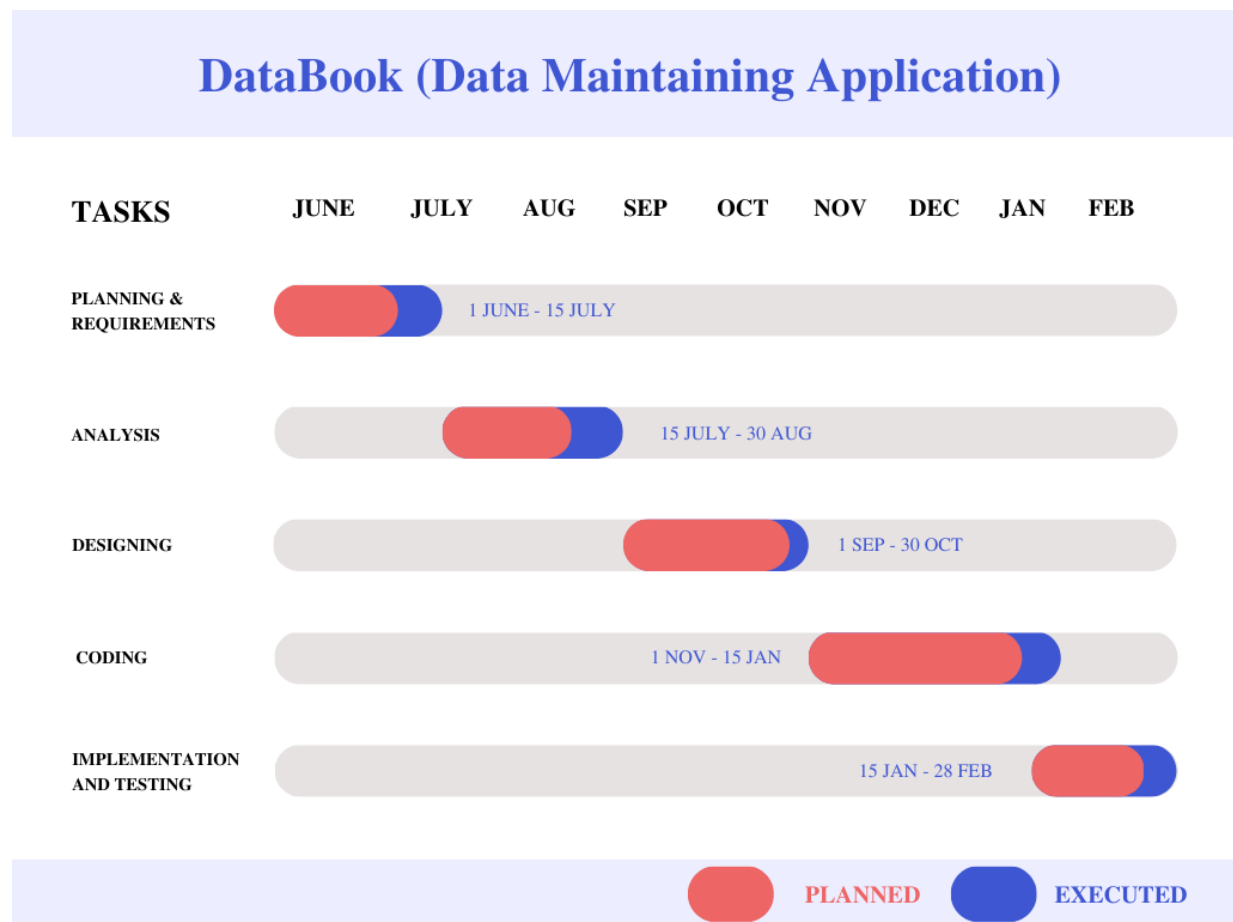
Functional Requirements:

- Creating an account on the application if the user is new.
- After creating an account, new users will be directed to feature activities.
- Existing users can navigate directly to feature activities, where all necessary activities will be displayed.
- Select the activity you want to proceed with. Activities have the options of note saving, task management, and money tracking.
- Select the required function of Update, Deletion, or Addition activity is completely dependent on the user.
- Enter all the details requested by the application.
- Select the date if necessary.
- "Mask as done" when the task is finished.
- It is also possible to delete the note after it has been used.

System Requirements:

- It only works on Android devices and must be downloaded from the Google Play Store.
- Have a stable internet connection only while using the application.
- All Android devices above API 21 (Lollipop) will be able to download this application.
- You will need at least 100 MB of storage space and a decent amount of RAM to download and use this application.

3.3 Planning and Scheduling



3.4 Software and Hardware Requirement

Software Requirements:

- To run the application on mobile, the minimum API is 21 (Lollipop) and above.
- Play Store to download the application.
- Storage minimum of 100 MB to download and save the daily data inserted in the application.
- Internet access is required while using the app and saving data to Firebase.
- The languages used in developing this application are Android Studio, Java, XML, JSON, and Firebase.

Hardware Requirements:

- Only Android devices with more than 2 GB of RAM can be UI-friendly and run applications smoothly.
- Permission to save the data in local storage.

3.5 Preliminary Product Description

- To replace maintaining a book with an online digital medium and minimise the user's workflow.
- One platform where users can find all features related to maintaining important daily data and scheduling meetings and work.
- It will also help users track what they have done throughout the day, how much time they have wasted, and so on.
- To help users maintain their whole day's data according to the time they have worked on it.
- Save the important work schedule based on dates and times in advance.
- All data-saving features, including money tracking, are available in a single application.
- Keeping track of money, expenses, and unnecessary spending with budget planning.
- A platform (application) that maintains their data in one place.
- Help in achieving goals and executing daily tasks, and also save time and energy.

CHAPTER 4: SYSTEM DESIGN

4.1 Basic Module

The core functional modules which will be available are:

- Login
- Register
- Feature Activity
- Notebook (Note Saving)
- Task Manager (To-Do)
- Money Tracker (Income, Expense, Budget)
- User Profile
- Forgot Password
- Profile Picture
- Update Profile
- Update Email I'd
- Change Password
- Delete Account

To cover all of the main activities, further divide into some sub-activities with different application design and functionality.

4.2 Schema Design

Table 1: Login Activity (Existing Users)

Column Name	Input-type	Data-type	Constraints	Allow Null
Email I'd	EmailAddress	String (100)	Primary Key	No
Password	InputPassword	String (100)	Unique Key	No

Table 2: Register Activity (New Users)

Column Name	Input-type	Data-type	Constraints	Allow Null
Full Name	TextPersonName	Character (100)	-	No
Email I'd	EmailAddress	String (100)	Primary Key	No
Date of Birth	Date	Long	-	No
Gender	RadioButton	-	-	No
Mobile Number	Phone	Long Integer (10)	Unique Key	No
Password	InputPassword	Varchar (50)	-	No
Confirm Password	InputPassword	Varchar (50)	-	No

DRMA ▾

Authentication

Users Sign-in method Templates Usage Settings

Add user

Identifier	Providers	Created ↓	Signed in	User UID
gmandaliya2@gmail.com	✉	24 Feb 2023	24 Feb 2023	2Gp4C0eQsGMXLiruRZQwkRj02P...
mandaliyatoral05@gmail.c...	✉	15 Feb 2023	15 Feb 2023	v4bzuY1P6afT1oHGJcorTbv17gA2
gautamman2906@gmail.c...	✉	14 Feb 2023	23 Feb 2023	i5UN3RU4bjbAftUBduDc61JtIW03

Rows per page 50 1 - 3 of 3

DRMA ▾

Realtime Database

Data Rules Backups Usage

<https://drma-29c73-default-rtdb.firebaseio.com>

```

https://drma-29c73-default-rtdb.firebaseio.com/
├── Registered Users
│   ├── 2Gp4C0eQsGMXLiruRZQwkRj02Pa2
│   │   ├── dob: "29/6/2002"
│   │   ├── fullName: "Gautam Chandrakant Mandaliya"
│   │   ├── gender: "Male"
│   │   └── mobile: "9326600407"
│   ├── i5UN3RU4bjbAftUBduDc61JtIW03
│   └── v4bzuY1P6afT1oHGJcorTbv17gA2

```

4.3 UML Diagrams

Diagram 1: Entity Relationship Diagram

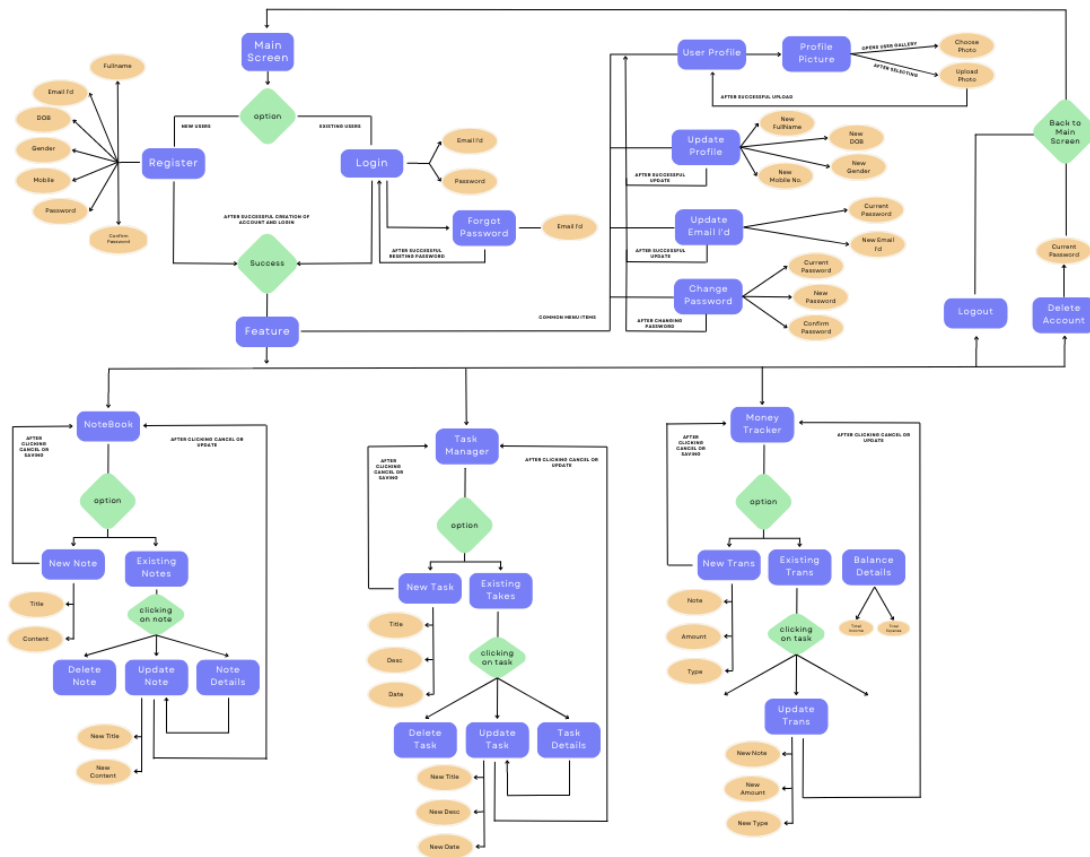


Diagram 2: Class Diagram

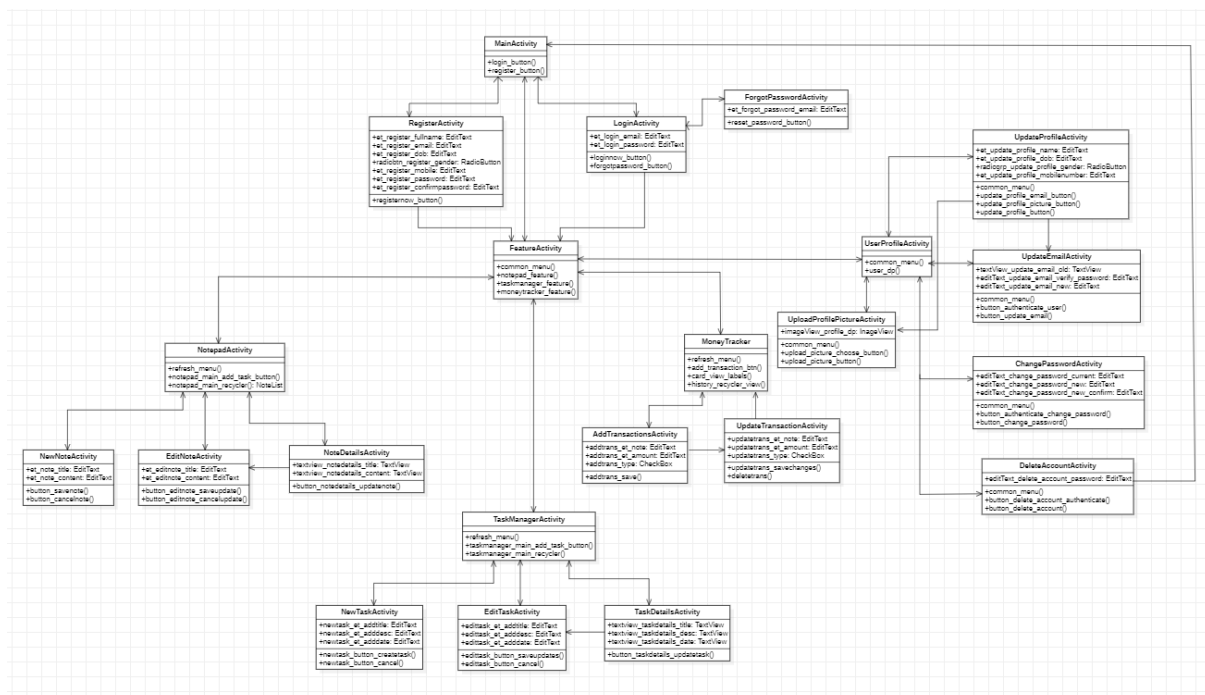


Diagram 3: Use case Diagram

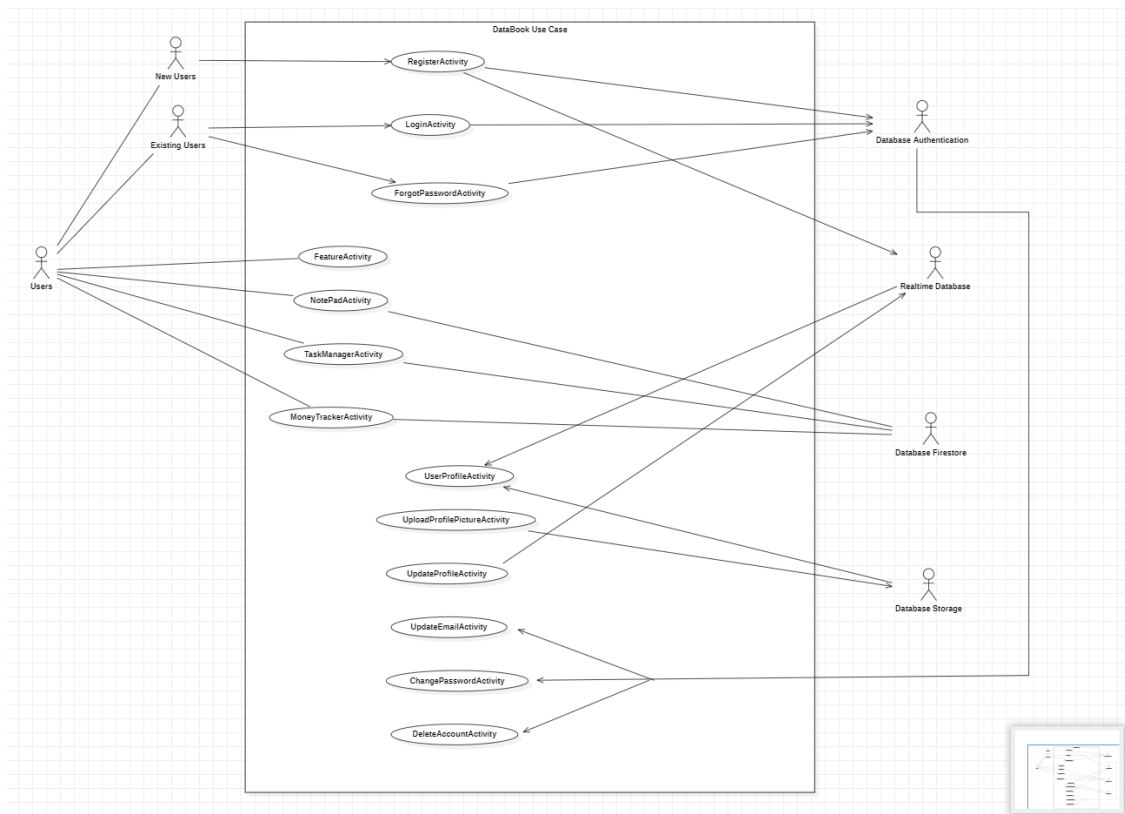


Diagram 4: Activity Diagram

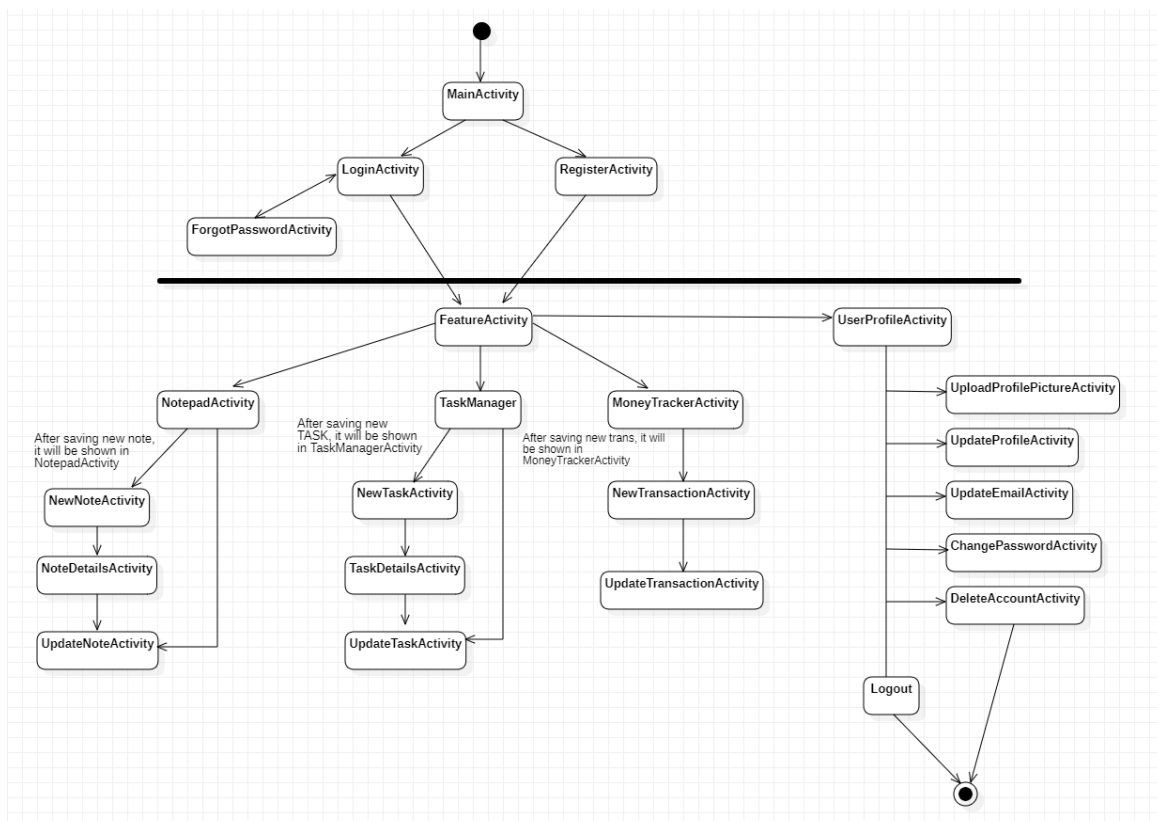


Diagram 5: State Machine Diagram

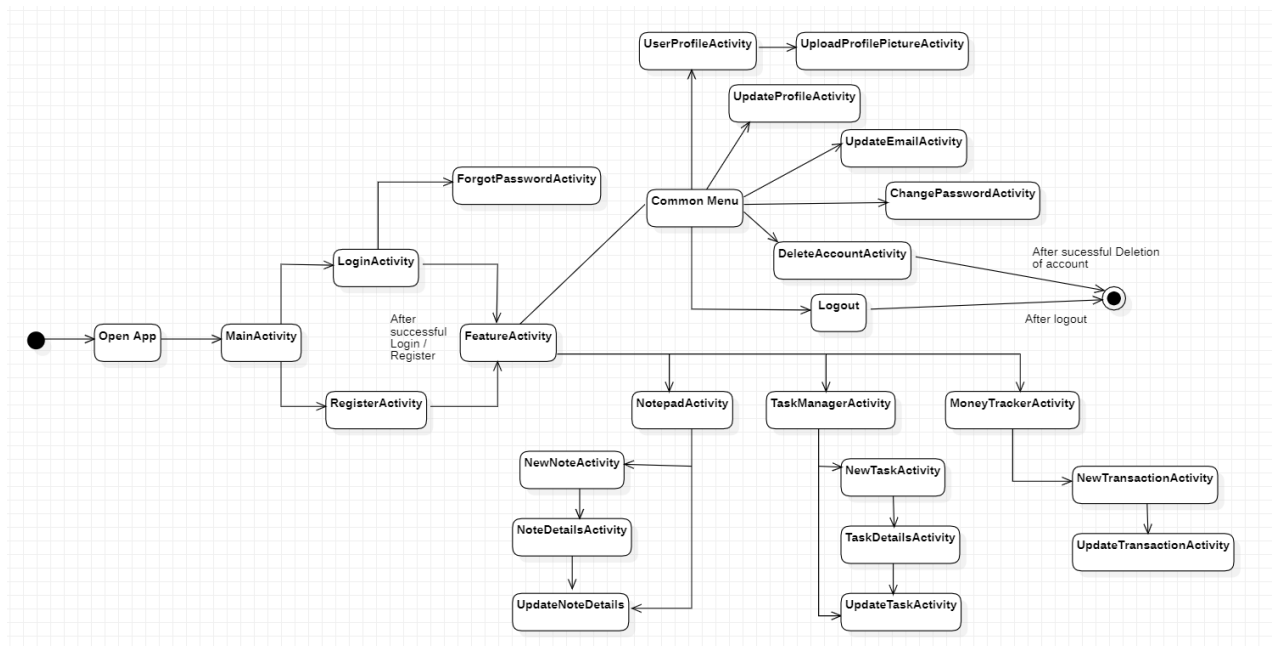


Diagram 6: Component Diagram

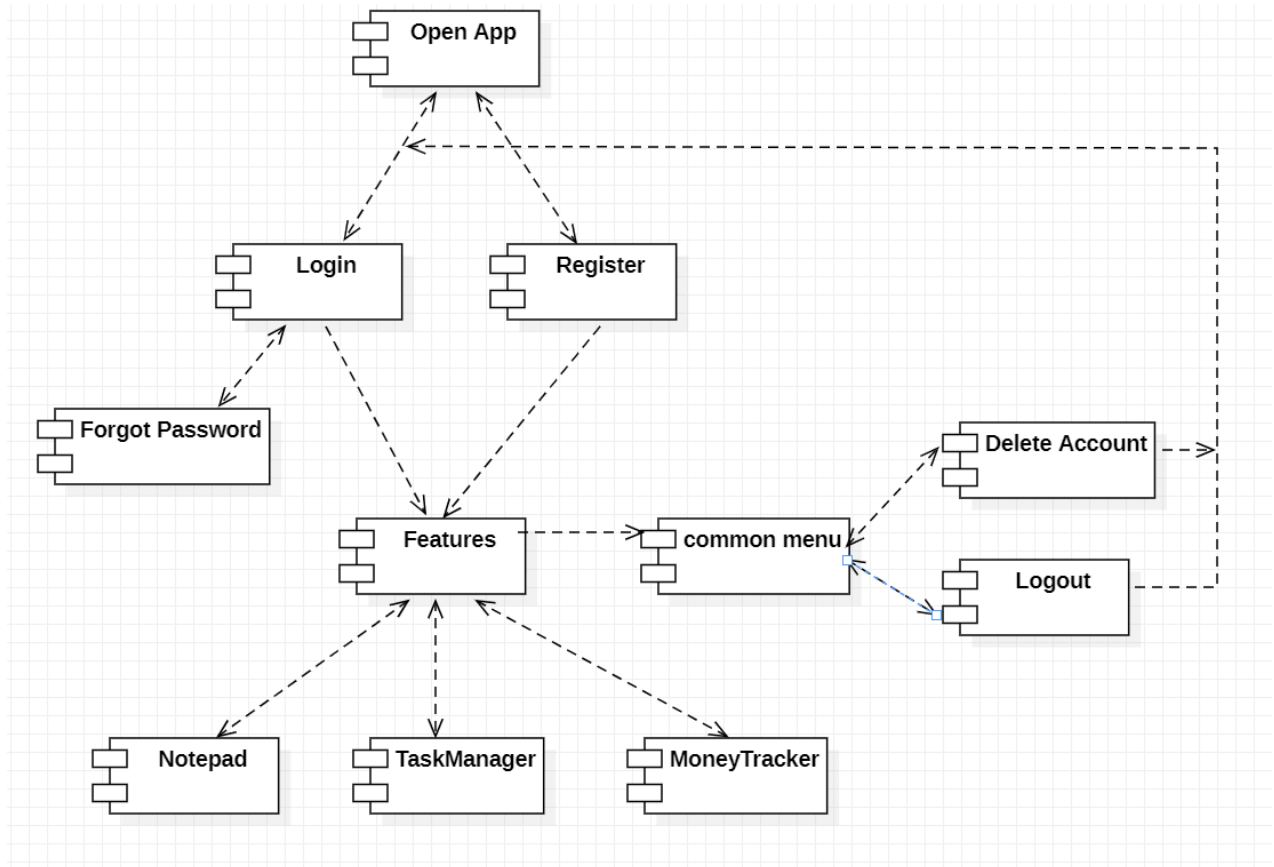
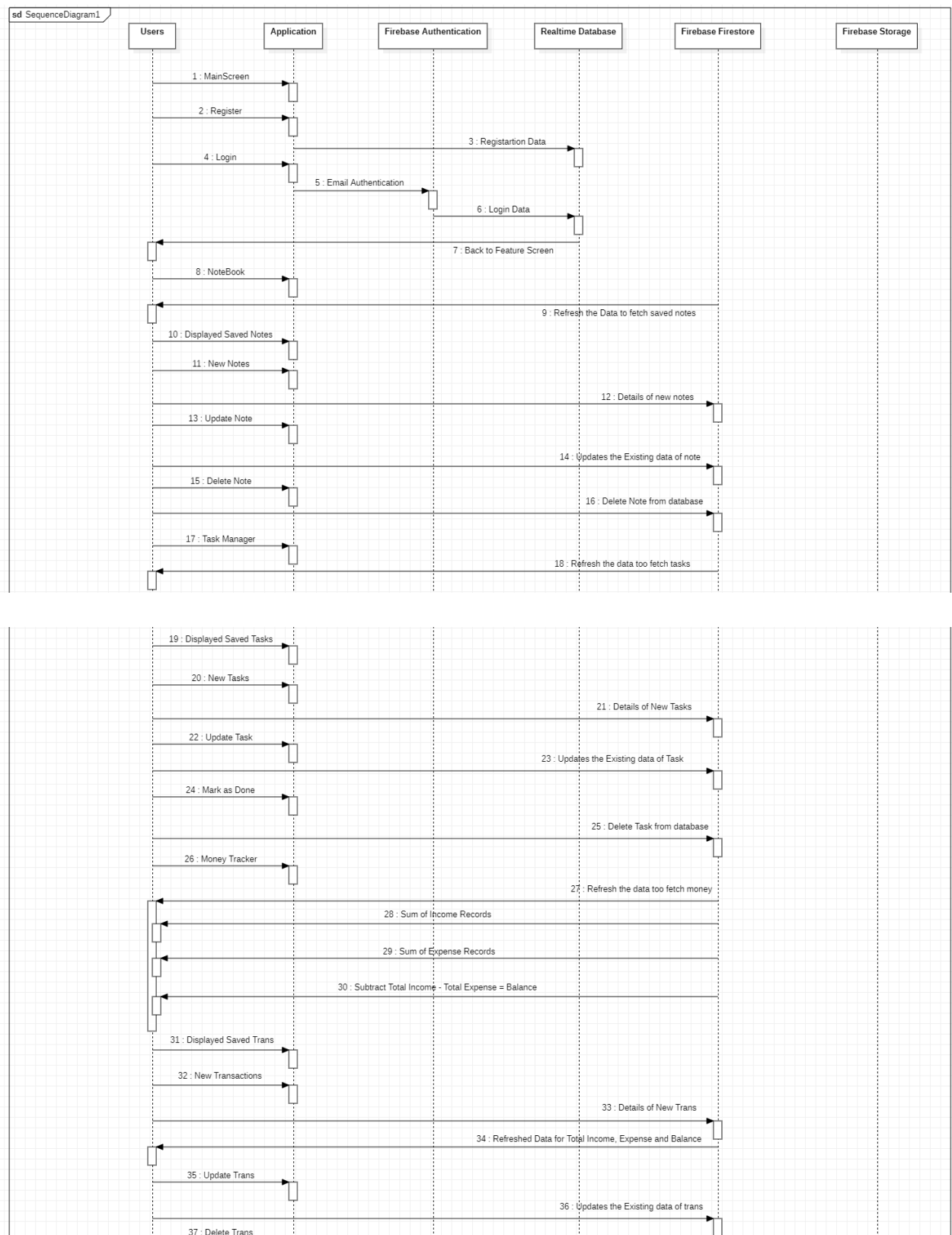
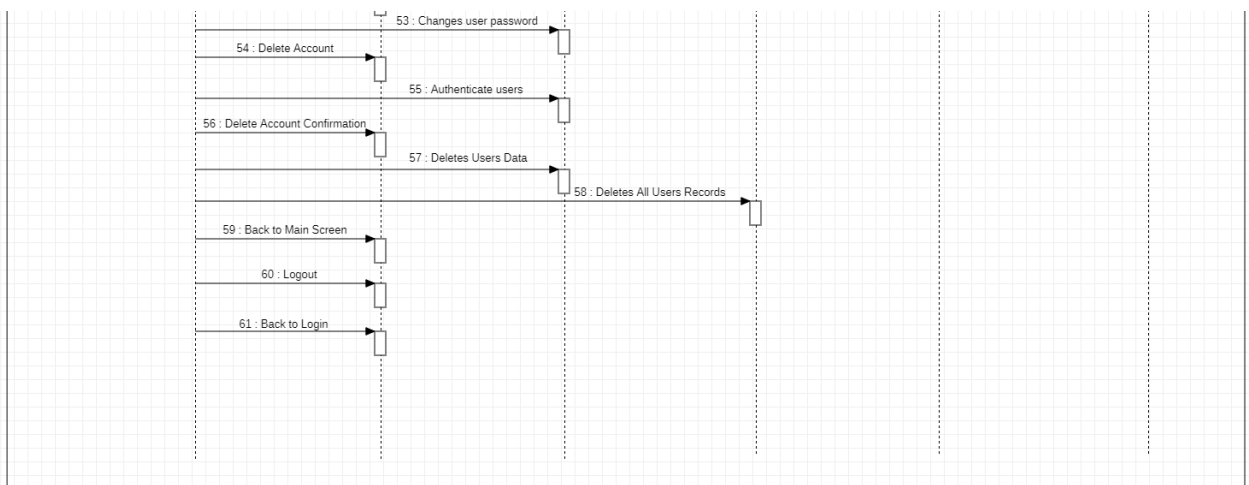
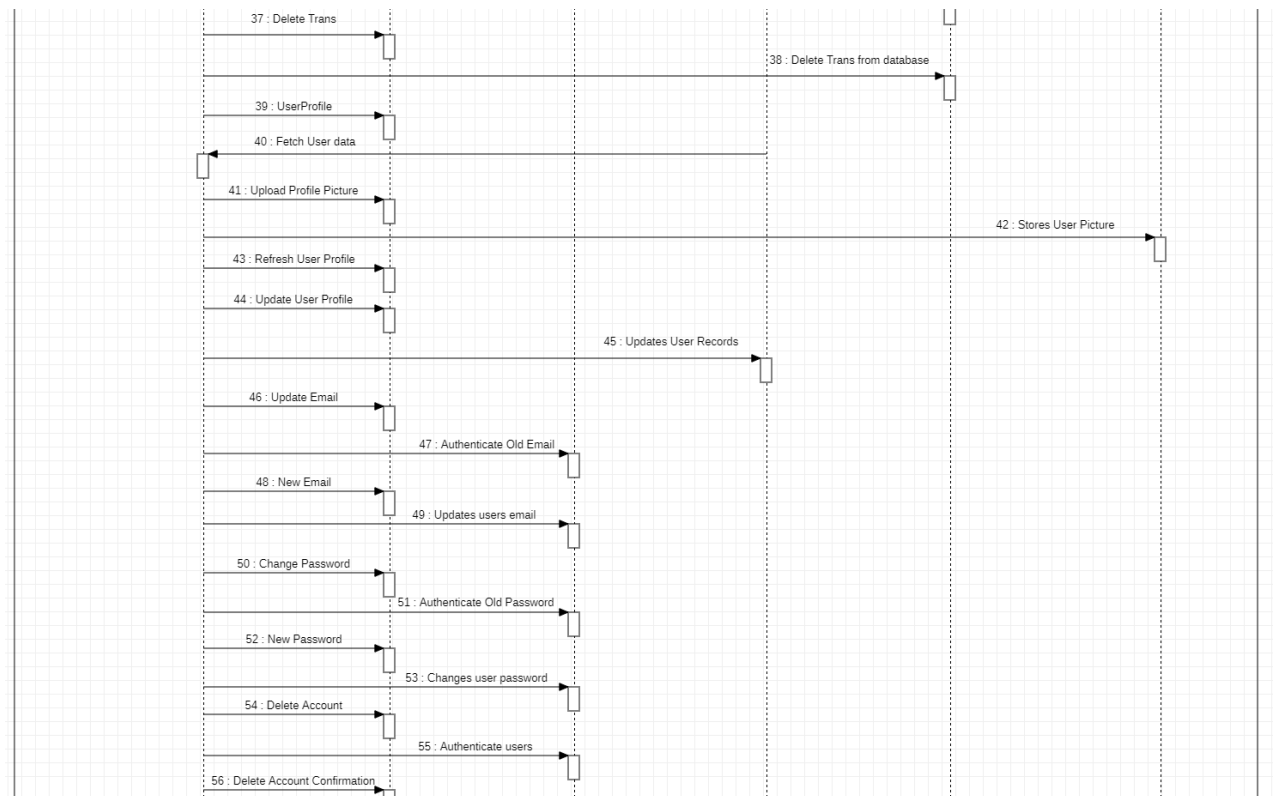
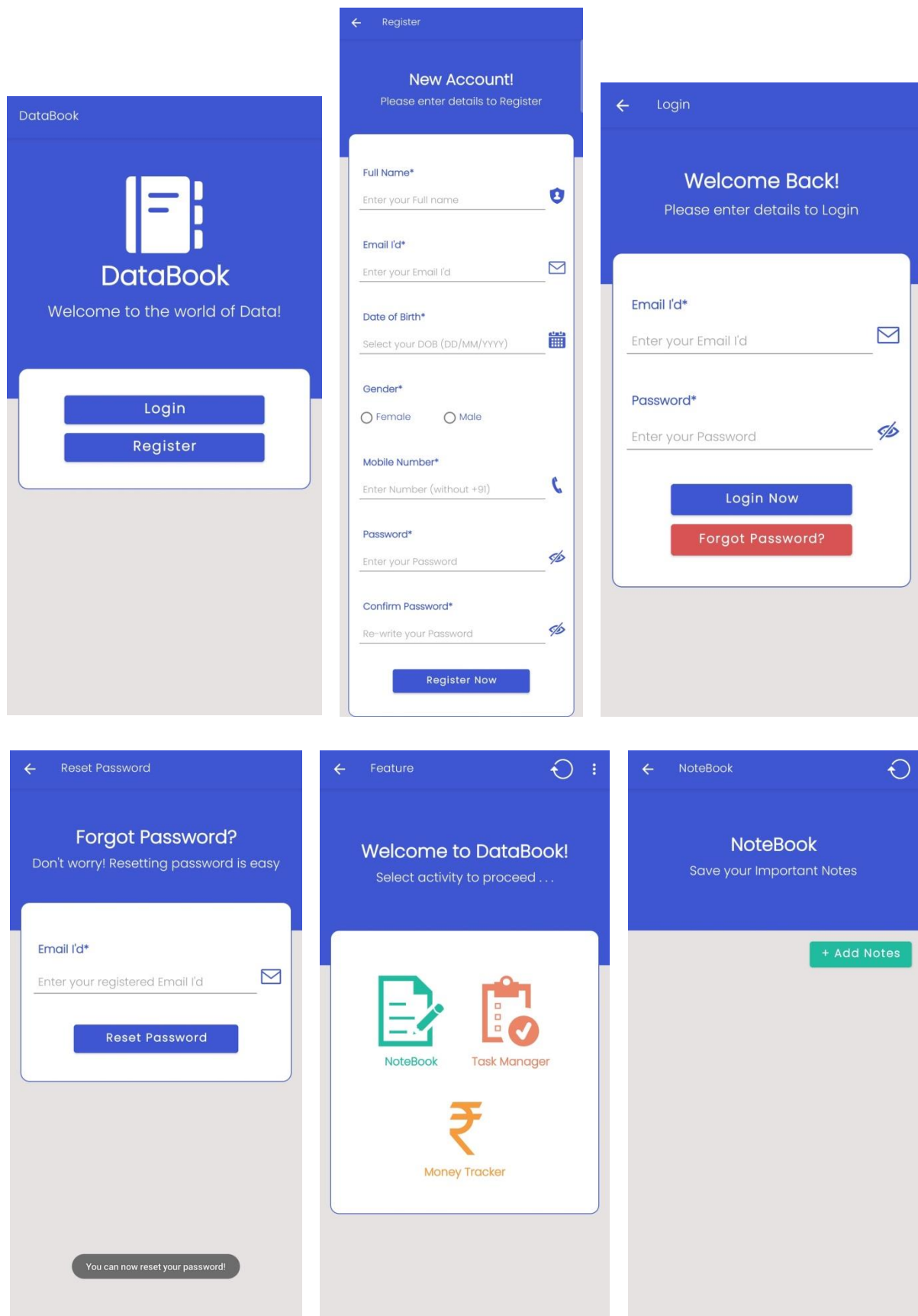


Diagram 7: Sequence Diagram





4.4 User Interface Design



←

New Note

Add New Note

Please enter Note Details

Enter the Title for Note

Enter the Content for Note

Save Note

Cancel

←

Note Details

Gautam Mandaliya Profile Details

Name : Mr. Gautam Chandrakant Mandaliya

College : Nagindas Khandwala College, Malad west

Degree : Bachelors of Science in Computer science

Location : Borivali, Mumbai

This project is for final year of my degree.

Thank You!!!

Edit Note

←

NoteBook

↺

NoteBook

Save your Important Notes

+ Add Notes

Gautam Mandaliya Profile

Update Note

Delete Note

←

TaskBook

↺

Task Manager

Save your Upcoming Tasks

+ Add Task

←

New Task

Add New Task

Please enter Task Details

Add Title*

Enter the title for task

Description*

Enter the description for task

Date of Completion*

📅

Enter the date for task

Create Task

Cancel

←

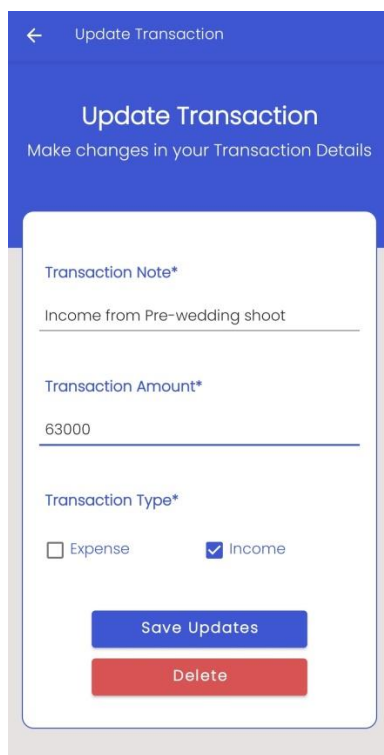
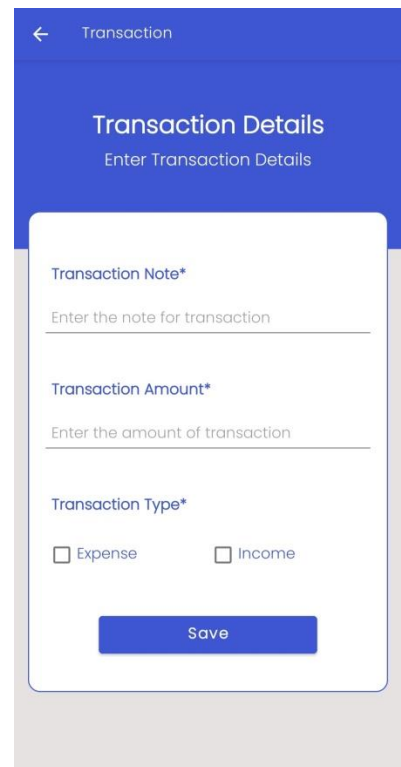
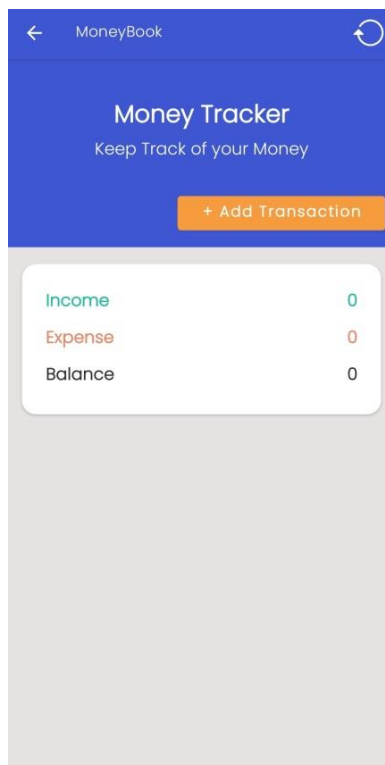
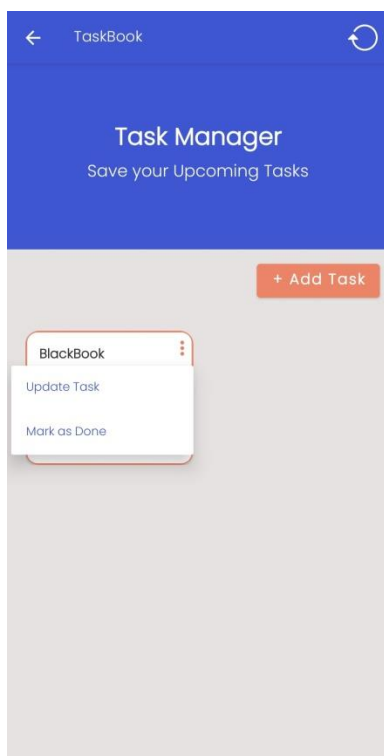
Task Details

BlackBook Submission

Next week monday or tuesday in college submission of blackbook to project guide

28/2/2023

Edit Task



Profile Picture

Upload your Profile Picture

Choose Picture

Upload

User Profile

Welcome, Gautam Mandaliya!

Gautam Mandaliya

gmandaliyal23@gmail.com

29/6/2002

Male

7700038043

Update Profile

Update your Personal Details

Update Email I'd

Upload Profile Picture

Full Name

Gautam Mandaliya

Date of Birth

29/6/2002

Gender

☐ Female
☒ Male

Mobile Number

7700038043

Update Profile

Update Email

Update your Email I'd

Please enter your current password and verify before continuing!

Current Email I'd

gmandaliyal23@gmail.com

Password*

Enter your current password

Authenticate

Your profile is not Authenticated yet!

New Email I'd*

Enter your new Email I'd

Update Email I'd

Change Password

Change your Password

Please enter your current password and verify before continuing!

Current Password*

Enter your current password

Authenticate

Your profile is not Authenticated yet!

New Password*

Enter your new password

New Confirm Password*

Confirm your new password

Change Password

Delete Account

Delete Account

Please enter your current password and verify before continuing!

Current Password*

Enter your current password

Authenticate

Your profile is not Authenticated yet!

Delete Account

4.5 Security Issues

- In this project, the database used is Firebase, which is the most secure database for an application because it was developed by Google, and the authentication and encryption used to protect the data are the top priorities of the developer.
- This application can be used by any person as long as they are authenticated by the developer, and the interchanging of the data is not allowed by any users, so users have to login to view their data.
- Every user has a unique ID to fetch their data from the database, so viewing another user's data is not a problem.
- An encryption pattern has been added to the database to retrieve the data.
- Data leaks are the most common issue that occurs on online applications, as the data of the user is stored on the servers, which can be hacked by hackers but can be kept safe if the proper steps are taken.
- Money fraud is extremely rare, but it can happen to anyone if hackers hack the data server and defraud people in a variety of ways.

CHAPTER 5: IMPLEMENTATION & TESTING

5.1 Code

MainActivity.java

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //Set the Title
    getSupportActionBar().setTitle("DataBook");
    //Open Login Activity
    Button buttonlogin = findViewById(R.id.login_button);
    buttonlogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(MainActivity.this, LoginActivity.class);
            startActivity(intent);
        }
    });
    //Open Register Activity
    Button buttonregister = findViewById(R.id.register_button);
    buttonregister.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(MainActivity.this, RegisterActivity.class);
            startActivity(intent);
        }
    });
}
```

RegisterActivity.java

```
public class RegisterActivity extends AppCompatActivity {
    private EditText edittextRegisterFullName, edittextRegisterEmail, edittextRegisterDOB,
    edittextRegisterMobile, edittextRegisterPassword, edittextRegisterConfirmPassword;
    private ProgressBar progressBar;
    private RadioGroup radioGroupRegisterGender;
    private RadioButton radioButtonRegisterGenderSelected;
    private DatePickerDialog picker;
    private static final String TAG = "RegisterActivity";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        getSupportActionBar().setTitle("Register");
        edittextRegisterDOB = findViewById(R.id.et_register_dob);
        edittextRegisterMobile = findViewById(R.id.et_register_mobile);
        edittextRegisterPassword = findViewById(R.id.et_register_password);
        edittextRegisterConfirmPassword = findViewById(R.id.et_register_confirmpassword);
        radioGroupRegisterGender = findViewById(R.id.radiobtn_register_gender);
        radioGroupRegisterGender.clearCheck();
        edittextRegisterDOB.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                final Calendar calendar = Calendar.getInstance();
                int day = calendar.get(Calendar.DAY_OF_MONTH);
                int month = calendar.get(Calendar.MONTH);
                int year = calendar.get(Calendar.YEAR);
                picker = new DatePickerDialog(RegisterActivity.this, new
                DatePickerDialog.OnDateSetListener() {
                    @Override
                    public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
                        edittextRegisterDOB.setText(dayOfMonth + "/" + (month + 1) + "/" + year);
                    }
                }, year, month, day);
                calendar.add(Calendar.YEAR, -16);
                picker.getDatePicker().setMaxDate(calendar.getTimeInMillis());
                picker.show();
            }
        });
        //Show Hide Password using Eye Icon
        ImageView imageViewShowHidePassword =
        findViewById(R.id.register_showhidepassword);
        imageViewShowHidePassword.setImageResource(R.drawable.hidepassword_maincolor);
    }
}
```

LoginActivity.java

```
public class LoginActivity extends AppCompatActivity {
    private EditText edittextLoginEmail, edittextLoginPassword;
    private ProgressBar progressBar;
    private FirebaseAuth authProfile;
    private static final String TAG = "LoginActivity";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        getSupportActionBar().setTitle("Login");
        edittextLoginEmail = findViewById(R.id.et_login_email);
        edittextLoginPassword = findViewById(R.id.et_login_password);
        progressBar = findViewById(R.id.login_progressbar);
        authProfile = FirebaseAuth.getInstance();
        //Reset Password
        Button buttonForgotPassword = findViewById(R.id.forgotpassword_button);
        buttonForgotPassword.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(LoginActivity.this, "You can now reset your password!",
                Toast.LENGTH_SHORT).show();
                startActivity(new Intent(LoginActivity.this, ForgotPasswordActivity.class));
            }
        });
    }
}
```

FeatureActivity.java

```
public class FeatureActivity extends AppCompatActivity {
    private FirebaseAuth authProfile;
    private ProgressBar progressBar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_feature);
        getSupportActionBar().setTitle("Feature");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        progressBar = findViewById(R.id.feature_progressbar);
        authProfile = FirebaseAuth.getInstance();
        progressBar.setVisibility(View.VISIBLE);
        //Open Notepad Activity using Image
        ImageView notepadImage = findViewById(R.id.notepad_feature);
        notepadImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(FeatureActivity.this, NotepadMainActivity.class);
                startActivity(intent);
            }
        });
        //Open Notepad Activity using TextView
        TextView notepadText = findViewById(R.id.notepad_feature_title);
        notepadText.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(FeatureActivity.this, NotepadMainActivity.class);
                startActivity(intent);
            }
        });
        //Open TaskManager Activity using Image
        ImageView taskmanagerImage = findViewById(R.id.taskmanager_feature);
```



```

taskmanagerImage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(FutureActivity.this, TaskManagerMainActivity.class);
        startActivity(intent);
    }
});

//Open TaskManager Activity using TextView
TextView taskmanagerText = findViewById(R.id.taskmanager_feature_title);
taskmanagerText.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(FutureActivity.this, TaskManagerMainActivity.class);
        startActivity(intent);
    }
});

//Open MoneyTracker Activity using Image
ImageView moneytrackerImage = findViewById(R.id.moneytracker_feature);
moneytrackerImage.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(FutureActivity.this, MoneyTrackerMainActivity.class);
        startActivity(intent);
    }
});

//Open MoneyTracker Activity using TextView
TextView moneytrackerText = findViewById(R.id.moneytracker_feature_title);
moneytrackerText.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(FutureActivity.this, MoneyTrackerMainActivity.class);
        startActivity(intent);
    }
});

```

UserProfileActivity.java

```
public class UserProfileActivity extends AppCompatActivity {
    private TextView textviewWelcome, textFullName, textViewEmail, textViewDOB,
    textViewGender, textViewPhone;

    private ProgressBar progressBar;
    private String fullName, email, DOB, gender, phone;
    private ImageView imageView;
    private FirebaseAuth authProfile;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user_profile);
        Objects.requireNonNull(getSupportActionBar()).setTitle("User Profile");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        textviewWelcome = findViewById(R.id.textview_show_welcome);
        textFullName = findViewById(R.id.user_show_fullname);
        textViewEmail = findViewById(R.id.user_show_email);
        textViewDOB = findViewById(R.id.user_show_dob);
        textViewGender = findViewById(R.id.user_show_gender);
        textViewPhone = findViewById(R.id.user_show_phone);
        progressBar = findViewById(R.id.user_progressbar);
        //Set onClickListener on ImageView to Open UploadProfilePictureActivity
        imageView = findViewById(R.id.user_dp);
        imageView.setOnClickListener(view -> {
            Intent intent = new Intent(UserProfileActivity.this,
UploadProfilePictureActivity.class);
            startActivity(intent);
        });
        authProfile = FirebaseAuth.getInstance();
        FirebaseUser firebaseUser = authProfile.getCurrentUser();
        if(firebaseUser == null) {
            Toast.makeText(UserProfileActivity.this, "Something went wrong. User details are
not available at the moment!", Toast.LENGTH_SHORT).show();
        }
    }
}
```

```

    } else {
        progressBar.setVisibility(View.VISIBLE);
        showUserProfile(firebaseUser);
    }
}

private void showUserProfile(FirebaseUser firebaseUser) {
    String userID = firebaseUser.getUid();
    //Extracting User Reference from Database for "Registered Users"
    DatabaseReference referenceProfile =
FirebaseDatabase.getInstance().getReference("Registered Users");
    referenceProfile.child(userID).addListenerForSingleValueEvent(new
ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            Register_ReadWriteUserDetails readUserDetails =
snapshot.getValue(Register_ReadWriteUserDetails.class);
            if (readUserDetails != null) {
                fullName = firebaseUser.getDisplayName();
                email = firebaseUser.getEmail();
                DOB = readUserDetails.dob;
                gender = readUserDetails.gender;
                phone = readUserDetails.mobile;
                textViewWelcome.setText("Welcome, " +fullName+ "!");
                textFullName.setText(fullName);
                textViewEmail.setText(email);
                textViewDOB.setText(DOB);
                textViewGender.setText(gender);
                textViewPhone.setText(phone);
                //Set User DP (After User has Uploaded)
                Uri uri = firebaseUser.getPhotoUrl();
                //ImageViewer setImageURI() should not be used with regular URIs. So we are
using Picasso
                Picasso.with(UserProfileActivity.this).load(uri).into(imageView);
            } else {

```

```

        Toast.makeText(UserProfileActivity.this, "Something went wrong!",
Toast.LENGTH_SHORT).show();
    }
    progressBar.setVisibility(View.GONE);
}
@Override
public void onCancelled(@NonNull DatabaseError error) {
    Toast.makeText(UserProfileActivity.this, "Something went wrong!",
Toast.LENGTH_SHORT).show();
    progressBar.setVisibility(View.GONE);
}
});
}

```

5.2 Testing Approach and Test Cases

In software development, testing is a crucial phase to ensure that the application meets the requirements and is functioning correctly. A testing approach is a plan or strategy that outlines how testing will be performed during the project. It defines the types of tests to be conducted, the level of testing, and the testing tools and techniques that will be used to ensure the quality of the software product.

The testing approach typically includes the following components:-

- Test objectives: They outline what the testing team aims to achieve through testing.
- Test scope: It defines the boundaries of the testing and what areas of the application will be tested.
- Test strategy: It defines the overall testing approach, including the types of testing, the testing environment, and the testing tools to be used.
- Test schedule: It outlines the timeline for the testing phase, including start and end dates.
- Test deliverables: It lists the documents and artifacts that will be produced during the testing phase, including test plans, test cases, test reports, and defect reports.

Test cases are the individual steps or conditions that are executed to verify that the software is functioning as expected. A test case typically consists of a set of inputs, expected outputs, and the steps to be executed to achieve the expected output.

Test cases help to ensure that the software is tested thoroughly and all the requirements are met. They also help to identify defects and issues in the application, which can be reported to the development team for correction. Test cases are typically documented in a test case repository and can be executed manually or automatically using testing tools.

Table 1: Login Test Cases

Test Case ID	Description	Expected Result	Actual Result	Remark
TC1	Email : (null) Password : (null)	Both Fields are required	Please fill the details	Pass
TC2	Email : gmandaliya Password : (null)	Invalid Email	Valid Email is required	Pass
TC3	Email : gmandaliya@gmail Password : (null)	Invalid Email	Valid Email is required	Pass
TC4	Email : gmandaliya@gmail.com Password : (null)	Password needed	Password is required	Pass
TC5	Email : gmandaliya@gmail.com Password : gautam	Password too short	Password should be atleast 8 characters	Pass
TC6	Email : (null) Password : gautam@123	Email needed	Email I'd is required	Pass
TC7	Email : gmandaliya@gmail.com Password : gautam@123	Login Success	You have logged in successfully	Pass

Table 2: Register Test Cases

Test Case ID	Description	Expected Result	Actual Result	Remark
TC1	Name : (null) Email : (null) DOB : (null) Gender : (null) Mobile Number : (null) Password : (null) Confirm Password : (null)	Fields cannot be empty	All fields are required, No fields can be empty	Pass
TC2	Name : Gautam786 Email : (null) DOB : (null) Gender : (null) Mobile Number : (null) Password : (null) Confirm Password : (null)	Name does not have number	Name contains only alphabets	Pass
TC3	Name : Gautam Mandaliya Email : gautam@gmail DOB : (null) Gender : (null) Mobile Number : (null) Password : (null) Confirm Password : (null)	Invalid Email	Valid Email I'd is required	Pass
TC3	Name : Gautam Mandaliya Email : gautam@gmail.com DOB : 29/06/2022 Gender : Male Mobile Number : 123456789000 Password : (null) Confirm Password : (null)	Number is more than 10 digits	Number should be 0 digits	Pass

TC4	Name : Gautam Mandaliya Email : gautam@gmail.com DOB : 29/06/2022 Gender : Male Mobile Number : 1234567890 Password : (null) Confirm Password : (null)	Mobile Number is invalid	Mobile Number is invalid	Pass
TC5	Name : Gautam Mandaliya Email : gautam@gmail.com DOB : 29/06/2022 Gender : Male Mobile Number : 7700038043 Password : gautam Confirm Password : (null)	Password too short	Password should be atleast 8 characters	Pass
TC6	Name : Gautam Mandaliya Email : gautam@gmail.com DOB : 29/06/2022 Gender : Male Mobile Number : 7700038043 Password : gautam@123 Confirm Password : gautam	Confirm Password should be same	Same Password is required	Pass
TC7	Name : Gautam Mandaliya Email : gautam@gmail.com DOB : 29/06/2022 Gender : Male Mobile Number : 7700038043 Password : gautam@123 Confirm Password : gautam@123	Registered Successfully	Account has been created successfully	Pass

Condition on Login Activity

- Both fields (Email I'd and Password) are required.
- Email I'd should be in this format (characters@abc.abc).
- Password should be atleast 8 characters

Condition on Register Activity

- All fields are required.
- Fullname can only contain alphabets.
- Email I'd should be in this format (characters@abc.abc).
- Mobile Number should be 10 digits and start with 7, 8, 9 for Indian number.
- Password should be atleast 8 characters.
- Confirm password should be same as password.

CHAPTER 6: RESULTS AND DISCUSSION

6.1 Test Reports

Our tests on the login and register activities as well as all the activities that have forms were successful. We were able to create accounts, log in, and use the activity features of the application. We encountered no major issues or errors during our testing, indicating that the activities are functioning as intended. We tested the application on various devices, including small phones, normal phones, and even a tablet, and it performed smoothly across all devices. Along with the activities, other functionality of the application, such as connection between various activities and the fetching of data from a real-time database, was successful. The application is fully functional.

6.2 User Documentation

Our application offers a simple and convenient way for users to save data of all kinds, from big notes to small tasks to even tracking the money.

Here's a step-by-step guide on how to use the application:

- Download our application from the Play Store.
- First, you need to create an account and login with the same credentials.
- Verify your email address.
- After successful login and registration, you will be navigated to a featured activity that has three main components: the notebook, task manager, and money tracker.
- Save large and important notes in the notebook section, this displays all of your saved notes from the application's inception in alphabetical order.
- To add a new note, click on the "new notes" button, enter all the required details, and just save.
- To view the note, just click on the particular saved note.
- To update the notes, click on the 3 dots beside the note title, update the note details, and click on save.

- To delete the notes, click the three dots and then "delete" to remove them from the note screen.
- For saving important tasks, go to the task manager section. Here you will see all of your saved tasks from the start of the application in ascending order by task date.
- To add a new task, click on the "new tasks" button, enter all the required details, and just save.
- To view the task, just click on the particular saved task.
- To update the tasks, click on the 3 dots beside the task title, update the task details, and click on save.
- After completion of the task, click on the 3 dots and click "mark as done." After completion, it will be deleted from the task screen.
- For saving money-related things, go to the money tracker section. Here you will see all your saved income, expense, and balance information from the start of the application in the order it was created.
- To add a new transaction, click on the "new transaction" button, enter all the required details, and then save.
- To update the tasks, simply click on the desired transaction, update the transaction details, and save.
- To delete the transaction, just click on the particular transaction that you want and click delete, it will be deleted from the transaction screen.
- To see user data, go to "User Profile Activity" from the common menu.
- To upload your profile picture, go to user profile activity and click on user profile picture. Choose the photo and click on "upload."
- Go to the "update profile," "update email," and "change password" activities to update your profile details, email id, and password, respectively.
- To delete the account, go to Delete Account Activity and note that this process is irreversible; once deleted, you can't access your account data.
- To log out of your account, click the logout button, and you will be returned to the main activity.

That's it! You have taken a step ahead in your life by noting everything and making it easier for yourself to balance your time, work, and money.

CHAPTER 7: CONCLUSIONS

7.1 Conclusions

7.1.1 Significance of the System

In conclusion, this project report has detailed the development and implementation of a data-maintaining Android application. The application provides a user-friendly interface for users to save notes and tasks ahead of time, as well as keep track of their income and expenses and a secure way to save their data. Throughout the development process, careful consideration was given to the user experience and security, resulting in an application that is both convenient and reliable.

7.2 Limitation of the System

- Using Firebase as the main database support, we will change the database and use our own database management system so that the privacy of the users' data can be increased. Admins will have to go to the Firebase database to keep track of the records.
- Applications require an internet connection to function properly. This can be problematic in areas with poor or no connectivity, making it difficult to access records or update them in real-time. Changing the support for low internet connectivity to update the data more easily and even at a faster rate.

7.3 Future Scope of the Project

- Notification for upcoming tasks and important work schedules.
- The money tracking system will include classification and graphical representation features, allowing users to easily understand their expenses, create monthly budgets, and spend according to their needs. One major feature of the money tracker will be the ratio of income to expenses and savings at the end of the month.

References

Links:

- <http://developer.android.com/docs> (Official Android Studio Documentation)
- <https://developer.android.com/reference/classes> (Official Android Development Classes for Android Developers)

Website:

- Stack Overflow (Error and Doubt Solving)
- Whimsical (Wire Framing)
- CodePath Android Cliffnotes (Tutorials and Guide)
- AndroidHive (Tips and Tricks)

Book:

- Beginning Android 4 Application Development (Author: Wei-Meng Lee)