

**AN ANDROID BASED APPLICATION “Amar Hisab- A PERSONAL
FINANCIAL MANAGER”**

BY

**MD. MAINUR RAHMAN MURAD
ID: 171-15-8758**

**MD. SIYAMUL ISLAM SIYAM
ID: 171-15-8865**

**MD. EUSUF UDDIN
ID: 171-15-8926**

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

SHADAAB KAWNAIN BASHIR
Lecturer
Department of CSE
Daffodil International University

Co-Supervised By

MD. JUEAL MIA
Senior Lecturer
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

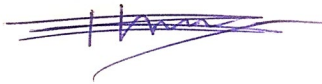
DHAKA, BANGLADESH

APRIL 2021

APPROVAL

This Project titled “**Amar Hisab -A Personal Financial Manager**”, submitted by Md. Mainur Rahman Murad, ID No: 171-15-8758; Md. Siyamul Islam Siyam, ID No: 171-15-8865 and Md. Eusuf Uddin, ID No: 171-15-8926 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 02/06/2021.

BOARD OF EXAMINERS



Dr. Touhid Bhuiyan
Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



Moushumi Zaman Bonny
Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Md. Sazzadur Ahamed

Senior Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Md Arshad Ali

Associate Professor

Department of Computer Science and Engineering
Hajee Mohammad Danesh Science and Technology
University

External Examiner

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Shadaab Kawnain Bashir, Lecturer, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



Shadaab Kawnain Bashir
Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:



Md. Jueal Mia
Senior Lecturer
Department of CSE
Daffodil International University

Submitted by:



Md. Mainur Rahman Murad
ID: 171-15-8758
Department of CSE
Daffodil International University



Md. Siyamul Islam Siyam
ID: 171-15-8865
Department of CSE
Daffodil International University



Md. Eusuf Uddin
ID: 171-15-8926
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to Almighty God for His divine blessing that makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Shadaab Kawnain Bashir, Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “Android Application” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Touhid Bhuiyan**, Professor and Head, Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

"**Amar Hisab** -A Personal Financial Manager" is an android application where we can calculate and store all the necessary accounts of personal life. In our application, the main features are debtors-creditors, daily income-expenses, savings and income tax. Using this application, user can keep track of our debtors and creditors, keep track of daily income and expenses, keep track of weekly and monthly savings and calculate the amount of government income tax. Users can also see the history of monthly and yearly income, expenses, savings, debtors and creditors. Users can directly make a call or message to their debtors/creditors from our application. Users can find easily a debtor/creditor by searching and can see his/her transaction history. Users can monitor their expense sectors and can maintain income and expense balance.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	ii
Declaration	iii-iv
Acknowledgments	v
Abstract	vi
CHAPTER	
CHAPTER 1: INTRODUCTION	1-2
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcome	2
CHAPTER 2: BACKGROUND	3-5
2.1 Preliminaries	3
2.2 Related Works	3
2.3 Comparative Studies	4
2.4 Scope of the Problem	4
2.5 Challenges	5

CHAPTER 3: REQUIREMENT SPECIFICATION	6-11
3.1 Business Process Modeling	6
3.2 Requirement Gathering and Analysis	7
3.3 Use Case Modeling and Description	7
3.4 Data Flow Diagram	8-9
3.5 ER Diagram	10
3.6 Design Requirements	11
CHAPTER 4: DESIGN SPECIFICATION	12-25
4.1 Front-End Design	12-22
4.2 Back-End Design	23-24
4.3 User Interface and Experience	25
4.4 Implementation Requirements	25
CHAPTER 5: IMPLEMENTATION AND TESTING	26-33
5.1 Implementation of Database	26-30
5.2 Implementation of Front-End Design	30-31
5.3 Testing Implementation	31-32
5.4 Testing Report	33

CHAPTER 6: IMPACT ON SOCIETY, ENVIRONMENT & SUSTAINABILITY	34
6.1 Impact on Society	34
6.2 Ethical Aspects	34
6.3 Sustainability Plan	34
CHAPTER 7: CONCLUSION AND FUTURE SCOPE	35-36
7.1 Discussion and Conclusion	35
7.2 Limitations	35-36
7.3 Scope for Further Developments	36
REFERENCES	37

LIST OF FIGURES

FIGURES	PAGE
Figure 3.1: Business Process Model	6
Figure 3.3: Use Case Diagram	7
Figure 3.4.1: Data Flow Diagram (Level 0)	8
Figure 3.4.2: Data Flow Diagram (Level 1)	9
Figure 3.5: ER Diagram	10
Figure 4.1.1: (a) Login Activity (b) Registration Activity	12
Figure 4.1.2: (a) Home Screen (b) Navigation drawer menu	13
Figure 4.1.3: Dena-Pawna main screen (b) Customer details history screen.	14
Figure 4.1.4: Dena-Pawna insert transition (b) Dena-Pawna update & delete transitions.	15
Figure 4.1.5: (a) Add customer screen (b) Add customer input screen.	16
Figure 4.1.6: (a) Income-Expense main screen (b) Main screen with the button.	17
Figure 4.1.7: (a) Savings main screen (b) Savings input screen.	18
Figure 4.1.8: (a) Receipt main screen (b) Receipt input screen.	19
Figure 4.1.9: (a) Income Tax main screen (b) Income Tax guideline screen.	20
Figure 4.1.10: (a) Others main screen (b) Others content screen.	21
Figure 4.1.11: (a) Profile activity (b) Settings activity.	22
Figure 4.2.1: Show registered user's information	23
Figure 4.2.2: Show specific user's information	24
Figure 5.1.1: show Register User List	27
Figure 5.1.2: Show Valid Signing Method for users	28
Figure 5.1.3: Show Dena-Pawna Transaction for user	29
Figure 5.1.4: All Joma-Khoroch Transaction post	30

CHAPTER 1

Introduction

1.1 Introduction

Every day's accounting is a part of our lives. From the beginning of the day to the time we go to sleep at night. People face many calculations daily. Sometimes it is too small and sometimes it is too big.

We sometimes use ledgers, diaries or notes to calculate these accounts. We have to control multiple ledgers or diaries to calculate these accounts. E.g. debt account, daily income-expenditure account, savings account, etc.

Keeping all this in mind, we have set up "**Amar Hisab**" application to solve and make issues easier. Its simple interface will make the calculation easier and save a lot of time as all account areas are together.

1.2 Motivation

People face a lot of reckoning daily. Fine accounting brings prosperity to our life and carelessness destroys us. People often forget to keep track of the small expenses and even the big expenses. If these expenses are not properly accounted, they will face many shortcomings in personal life. Although some people calculate their daily accounts through ledgers or dairies in this modern era.

Keeping all this in mind, we made "**Amar Hisab**" application to solve these problems. Its simple interface will make the calculation easier and save a lot of time as all account areas are together.

So we are motivated to create such an application to be careful about accounting and to manage accounts properly.

1.3 Objectives

Our project goal is to build all the accounting arrangements of daily life easier. Proper use of technology instead of ledgers, diaries. To make the system easy-peasy by coming up with all personal accounting areas on one platform. Basically, the main objectives of our project are:

- View all the accounts of the debtors-creditors in detail, find them easily and keep those as proof.
- Maintaining balance through income and expenditure.
- Saving comparatively time and labor.
- Reduce waste by verifying spending areas.

1.4 Expected Outcome

This application will facilitate all the personal accounting systems of the people. Reduce the use of ledger for income-expenditure, debt-accounting. Will help reduce waste of time. Accounting in the account will eliminate the hassle of saving the previous accounts. Its user-friendly interface will save time at a significant rate and will eliminate reluctance towards accounting. We hope that our application will play a role as a trusted accountant in accounting for daily income expenses, debts, savings, etc.

CHAPTER 2

BACKGROUND

2.1 Preliminaries

In modern civilization, Android is a signifying conventional platform. Nowadays everything becomes digitized. Previously a person used a physical notebook or diary to store daily transactions. It was difficult to carry a notebook everywhere and tough to calculate the final result. That's why we have decided to develop such a daily personal transaction Application. That Application helps a user to store loan, payable, revenue, expense, savings, withdrawal, essential paper with a date. Users can also calculate income Tax using this Application. In this application, we are trying to Store all kinds of daily needed Information in a single application.

2.2 Related Works

Google Play Store has already many applications for calculating daily personal transactions like Hisaber Khata, Bakir khata, Sohoj Hisab, etc. Hisaber Khata Application tries to store daily Expense/Revenue. This application stores data using local storage and users have to back up their transaction data manually via google storage. Bakir khata application tries to store the only total due and total payable and this application gives more Ads that are disappointing for users. Sohoj Hisab also stores daily revenue lists and daily expense lists and stores necessary money borrow, money landing, etc.

2.3 Comparative Studies

Currently, most people used a physical notebook/diary to save daily transactions & mostly needed information. Businessmen also use a register book system to save their daily transactions and took a long time to calculate the final outcome.

At present, few applications already exist in the play store to solve those problems. But those applications only perform a single task. For example, some applications only store users' daily revenue and expenses. But did not calculate user total transactions and did not show the Net Balance. Some applications also store savings and costs on a daily basis. The existing application contains some functionality & UI-related problems and is being crushed a few times.

For this reason, it generates a negative user experience. The Personal Money Management (Amar Hisab) will improve the functionality of the existing application also includes additional functionality like debt/payable, Revenue/cost, savings/withdrawal, store cash memo and calculate the income tax within a single application. Users will be able to handle his/her online data anytime with another device from anywhere using user-specific email and password. There is no way to lose user data. User data will keep safe & secure in our system. We have designed a user-friendly UI for user satisfaction.

2.4 Scope of the Problem

This application uses a real-time cloud storage database to store user data. That's why when a user goes offline mode he/she will not be able to see the transaction dashboard and unable to perform any transactions. Users are also unable to store graphical data in memo-roshid and profile activity in offline mode.

When a user inserts a new customer then this system will not be able to send any notification to the customer. Offline mode and push customer notification handle is a major problem in this application. Which we will solve this problem in the future.

2.5 Challenges

User-friendly UI and responsive design is a big challenge to apply in Android applications which we have overcome into our Application. We are designed our user interface using Adobe XD for getting a perfect GUI.

- To reduce application size we have used vector images. Vector Image takes little space and provides better resolution which helps to create lightweight application sizes.
- To build user-friendly interfaces we have used Material Components which are a little bit difficult to implement in this application.
- Real-time dashboard summary has been calculated using all customer transaction amount. It was too complex to show in a real-time dashboard.
- To Reduce Application Crush probability we have used valid input fields to take all user inputs.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Business Process Modeling

The business process model (BPM) is a technique/tool which is used to manage business virtue. It is a graphical structure of a company's or organization's business process. It is very important to analyze a business step by step. The following figure 3.1 exposes the Business Process Model (BPM) of this application.

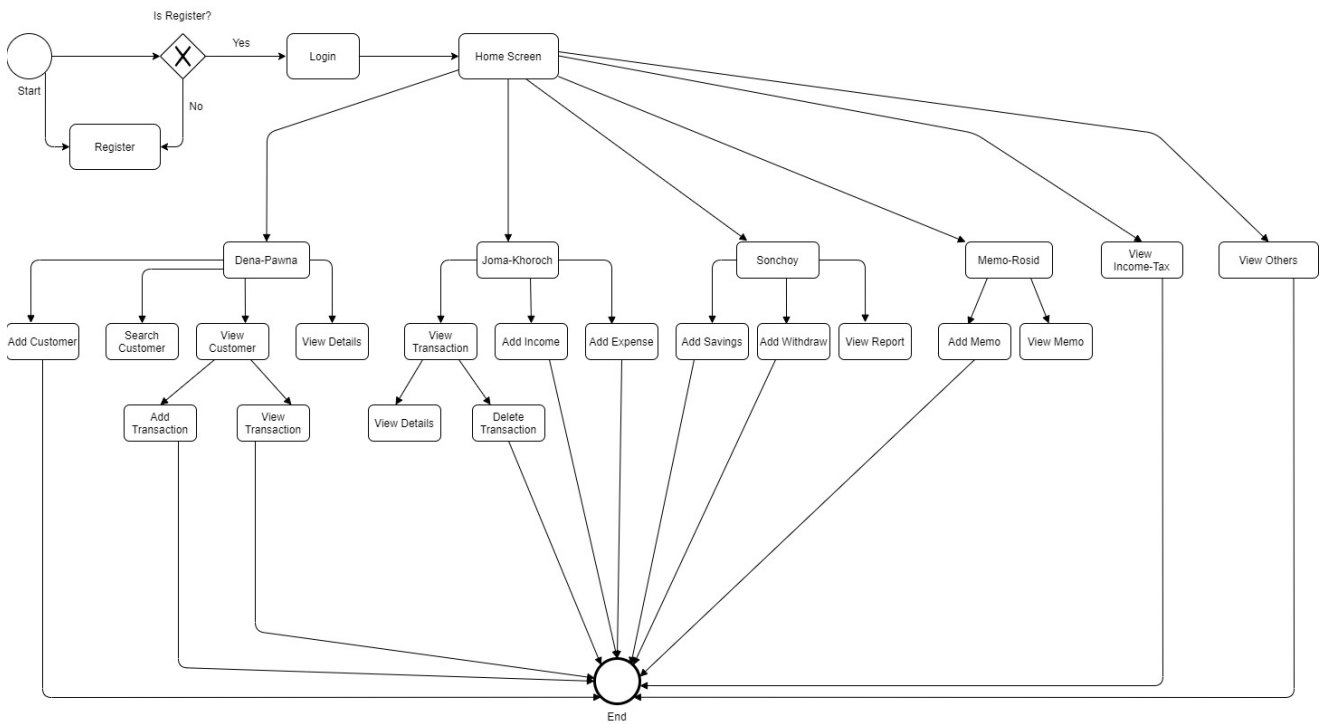


Figure 3.1: Business Process Model

3.2 Requirement Gathering and Analysis

The quality of an application depends on the gathering of the requirement of a project. It is the most important part of a project. We used various techniques to collect the requirement of our project. Our application is designed for all kinds of people. So we tried to reach various types of people one by one to understand their expectations from this type of application. We made a survey form where set some questionnaire to collect requirement of such an application. The brainstorming phase was that when we analyzed the problems that were raised by people. Finally based on their suggestion and our analysis we prepared a prototype of our application.

3.3 Use Case Modeling and Description

The use case represents the pictorial of user action. It is shown user activity step by step.

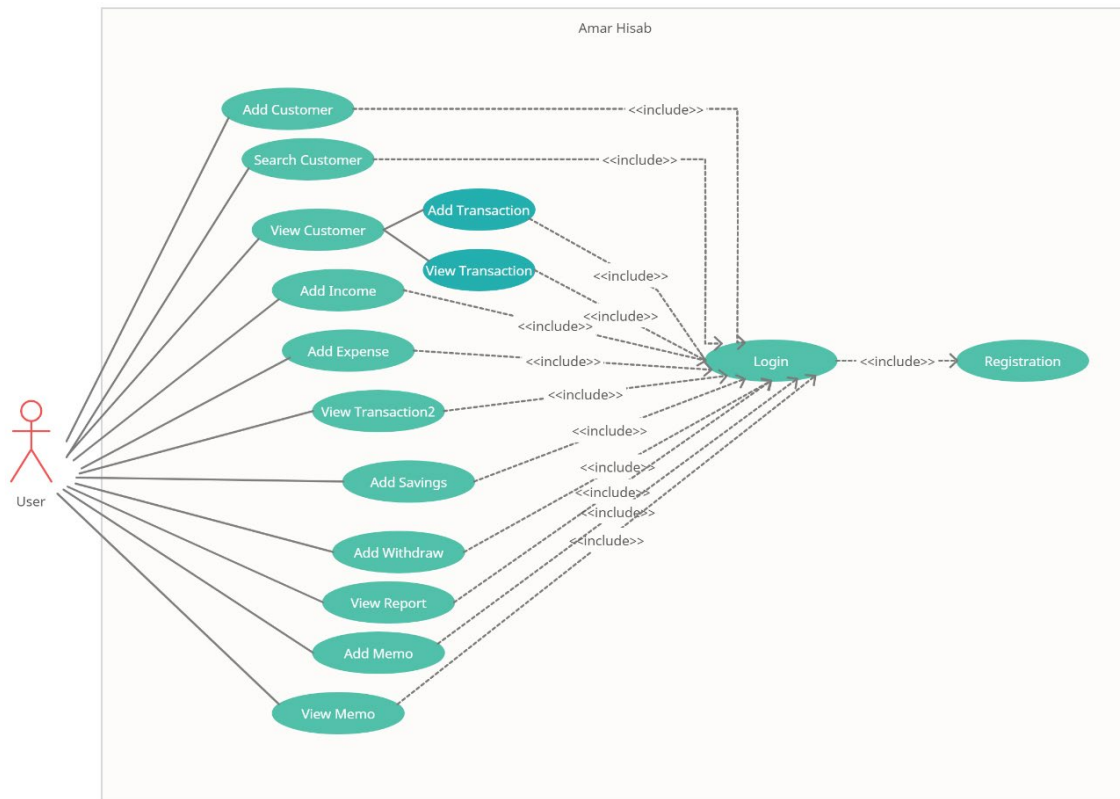


Figure 3.3: Use Case Diagram

3.4 Data Flow Diagram

Data Flow Diagram defines the user input and output information of a system. It describes how to transfer data from user to storage. The following figure 3.4.1 and figure 3.4.2 show the DFD of our project.

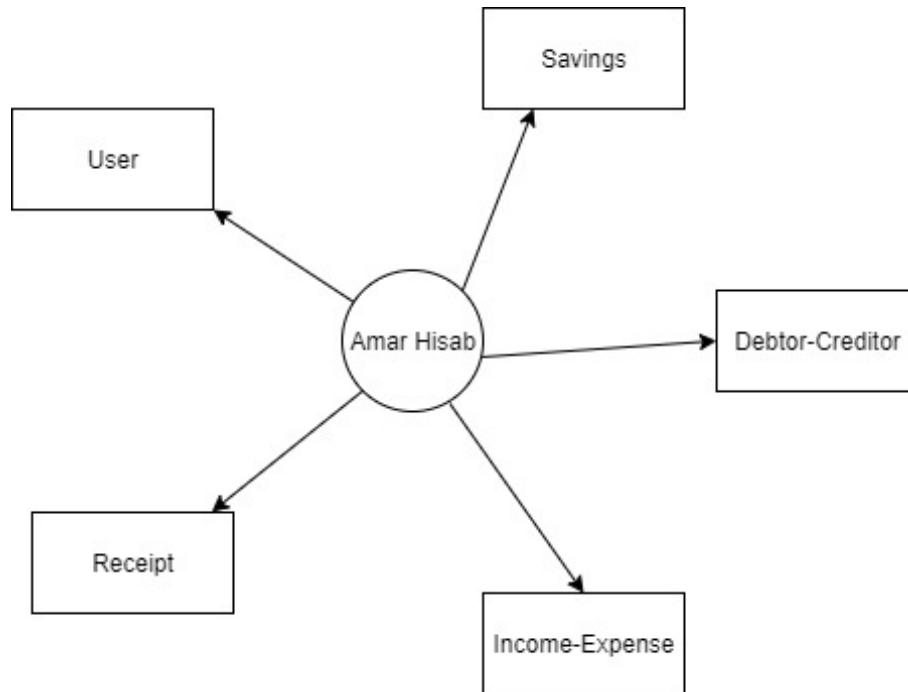


Figure 3.4.1: Data Flow Diagram (Level 0)



Figure 3.4.2: Data Flow Diagram (Level 1)

3.5 ER Diagram

Entity Relationship Diagram (ER Diagram) is a graphical structure of a database. It is one kind of blueprint of a database before implementation. ER diagram contains entity sets and relationship sets. Our ER diagram contains 8 entity sets. Like, user, customer, transaction, etc. The following figure 3.5 shows the ER diagram of our application.

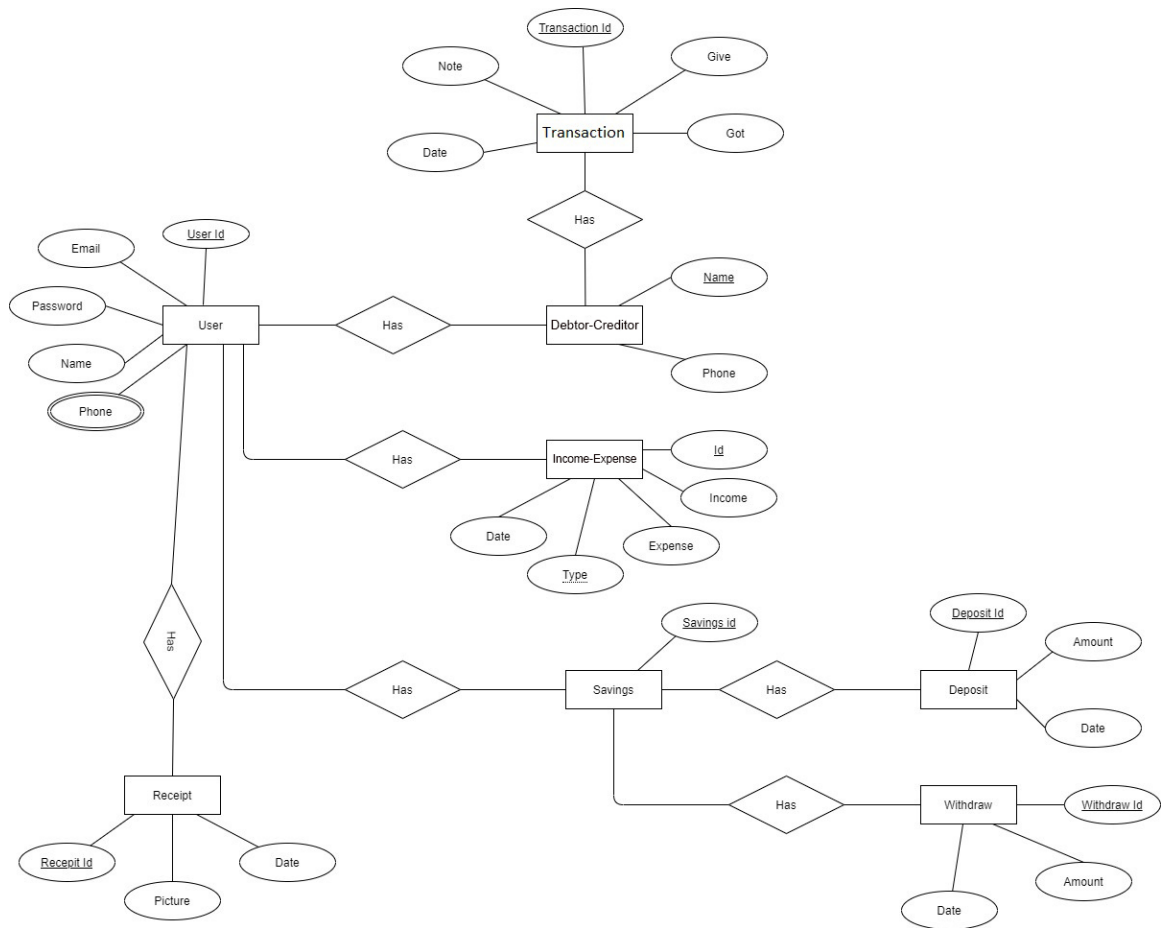


Figure 3.5: ER Diagram

3.6 Design Requirements

Design requirement defines the action of an application and gives clear idea about the functionality of the system. The design requirement for users of our project:

- Able to Registration
- Able to Login
- Can calculate debtor and creditor accounts
- Can calculate daily accounts
- Can calculate savings and withdraw
- Can store receipt
- Can view Income tax
- Can view others
- Exit App.

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-End Design

User interface refers visual representation of an application. The quality and popularity of an application largely depend on front-end design. It usually expresses the interaction design.

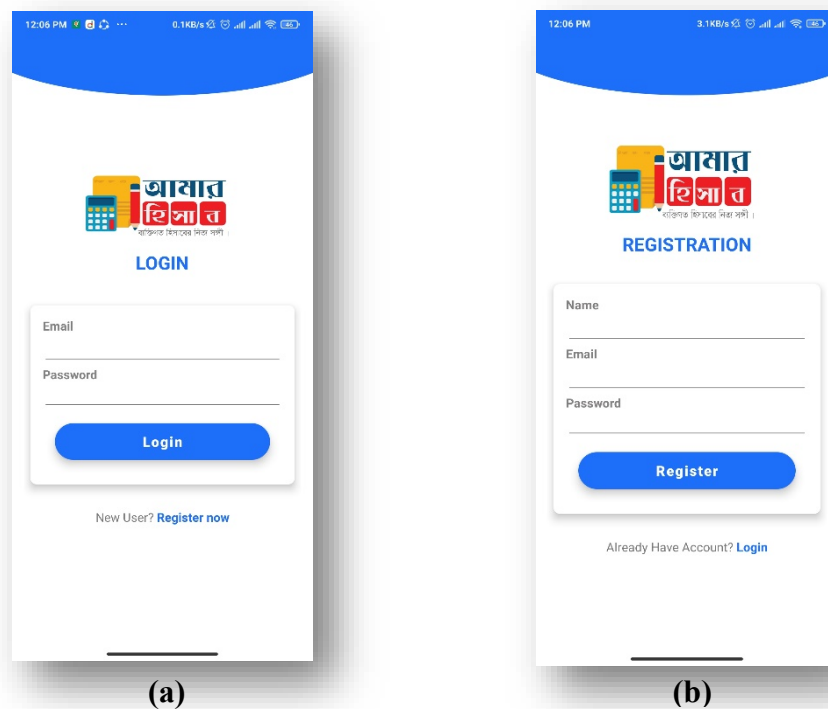
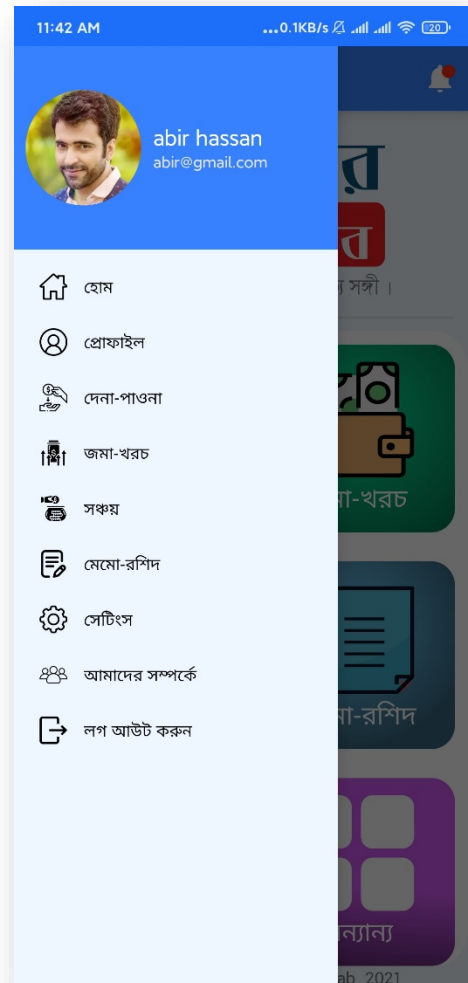


Figure 4.1.1: (a) Login Activity (b) Registration Activity

Figure 4.1.1 (a) shows the login screen and (b) shows the registration screen of our app.



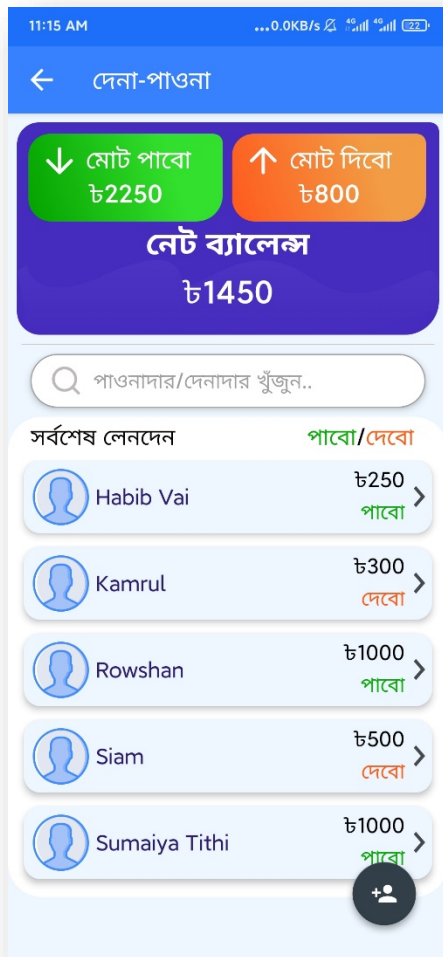
(a)



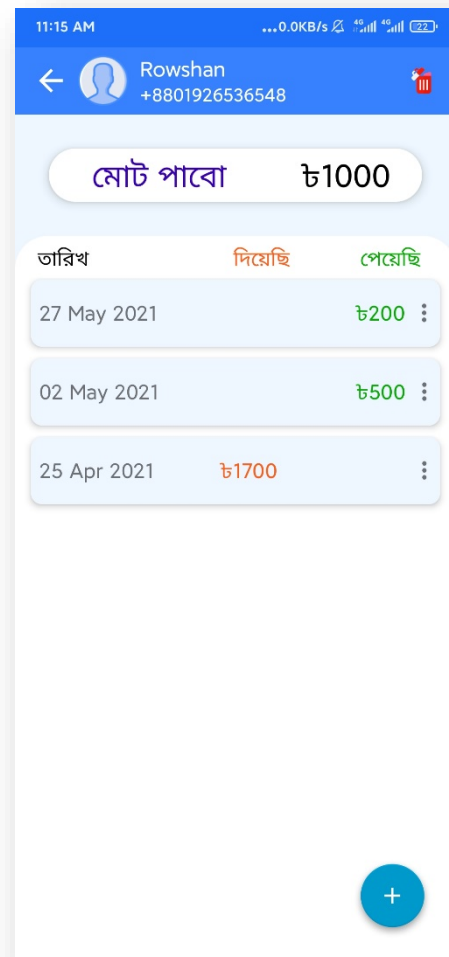
(b)

Figure 4.1.2: (a) Home Screen (b) Navigation drawer menu

Figure 4.1.2: (a) shows the home screen which shows six main features of our application and (b) shows the navigation drawer menu with a profile. This menu contains profiles, settings, about us, logout, etc.



(a)



(b)

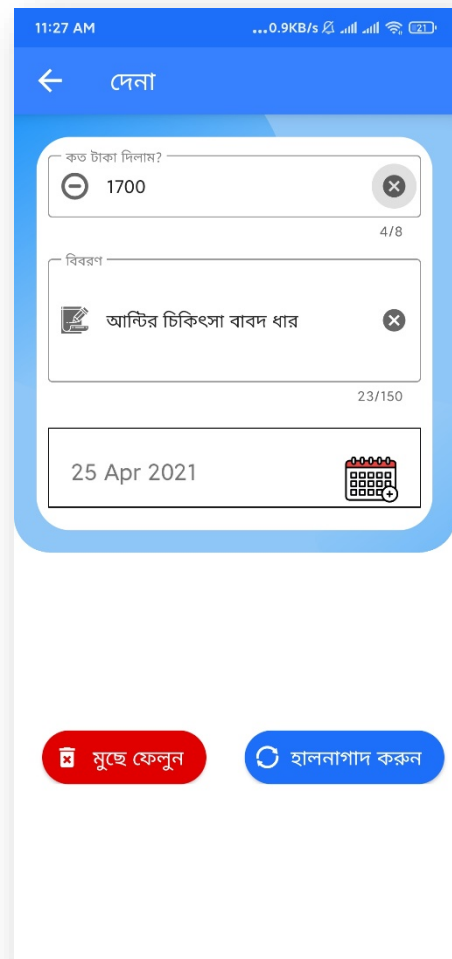
Figure 4.1.3: Dena-Pawna main screen (b) Customer details history screen.

Figure 4.1.3 (a) shows Dena-Pawna's main screen where at a glance users can see debit, credit and net balance. User can see their debtors and creditors and also find them by searching.

Figure 4.1.3 (b) shows the customer history screen where the user can see how much amount money gets or give and newly add money as to give or get.



(a)

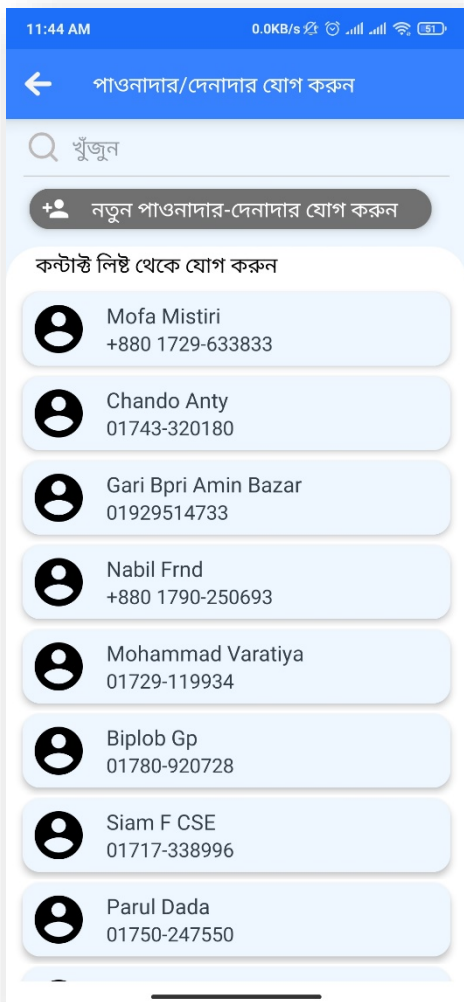


(b)

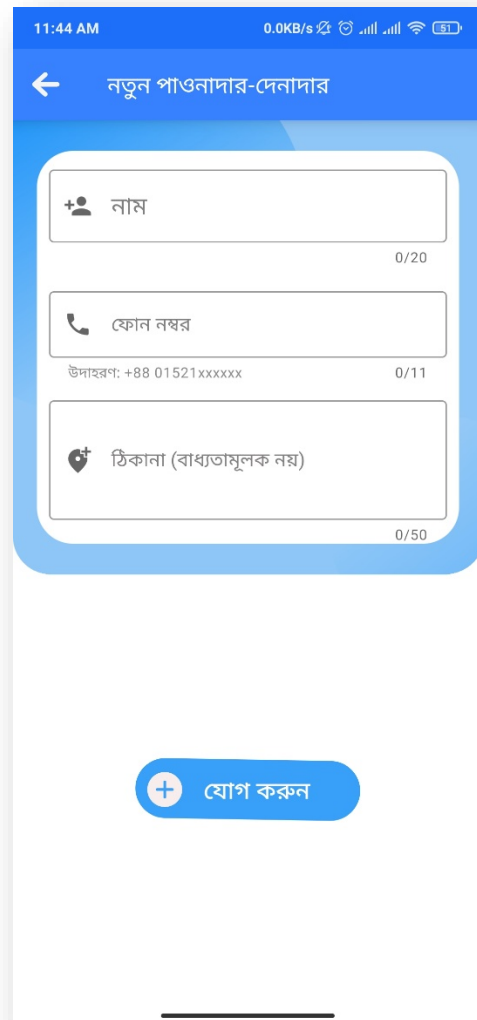
Figure 4.1.4: Dena-Pawna insert transition (b) Dena-Pawna update & delete transitions.

Figure 4.1.4 (a) shows Dena-Pawna's insert transition activity, where users can insert debit, credit amounts.

Figure 4.1.4 (b) shows the update/delete transitions activity.



(a)



(b)

Figure 4.1.5: (a) Add customer screen (b) Add customer input screen.

Figure 4.1.5 (a) Shows add customer screen where the user can add customers manually or add from the contact list.

Figure 4.1.5 (b) Shows add customer input screen where the user can add customer manually by filling up the customer's information.



(a)



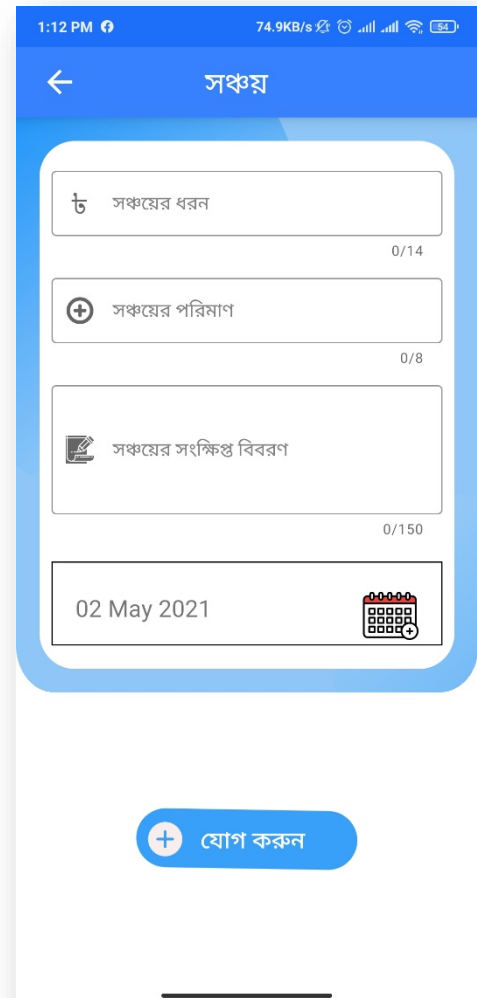
(b)

Figure 4.1.6: (a) Income-Expense main screen (b) Main screen with the button.

Figure 4.1.6 (a) & (b) shows the income-expense main screen where user can add their income and daily expense also see income, expense and net balance.



(a)



(b)

Figure 4.1.7: (a) Savings main screen (b) Savings input screen.

Figure 4.1.7 (a) Shows savings main screen where user can add, delete and update savings and withdraw. Also can see savings and withdrawal amount on the dashboard, date to date savings and withdrawal history.

Figure 4.1.7 (b) shows a savings input screen where users can add savings by filling up the information.



(a)

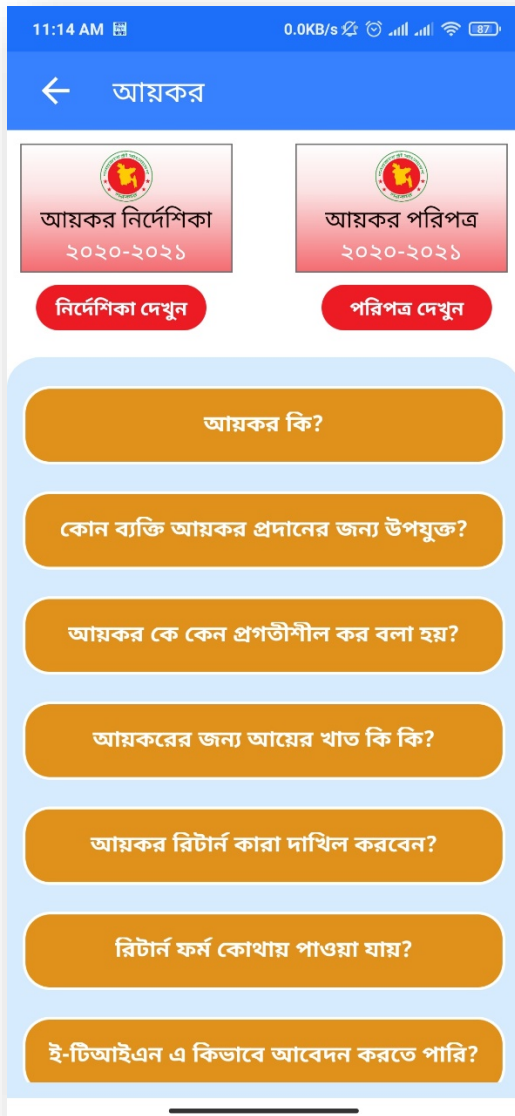


(b)

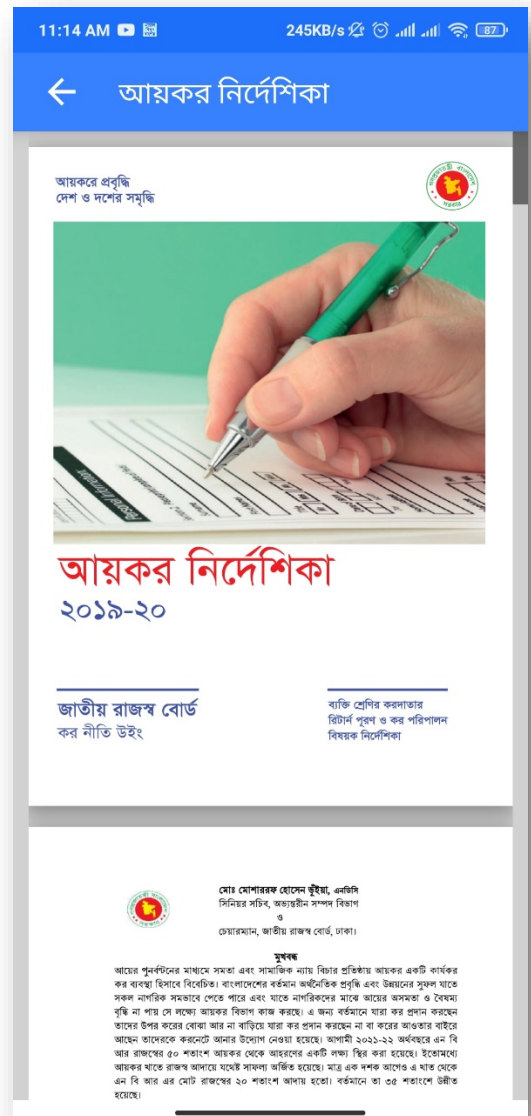
Figure 4.1.8: (a) Receipt main screen (b) Receipt input screen.

Figure 4.1.8 (a) shows the receipt main screen where user can see their uploaded receipt, cards, memos, etc.

Figure 4.1.8 (b) shows a receipt input screen where users can upload receipts, cards, memos, etc.



(a)

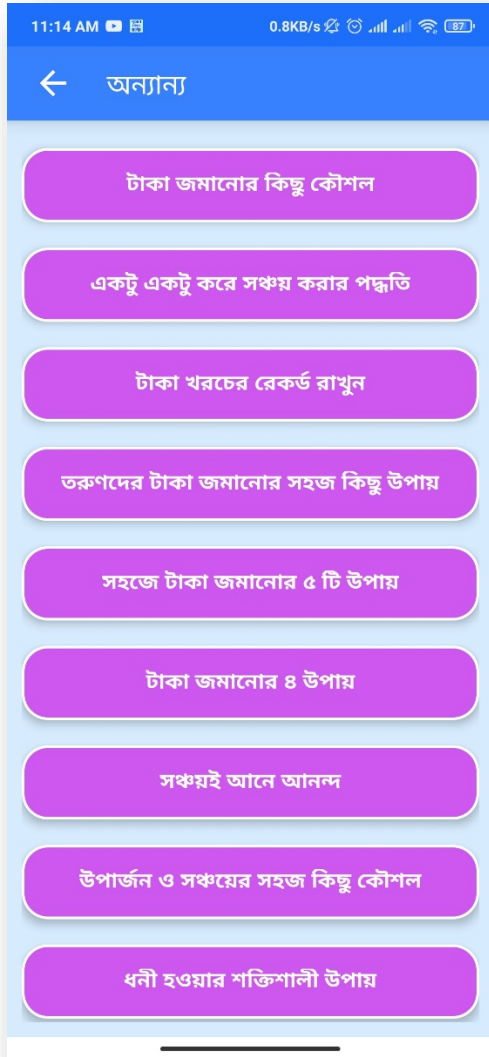


(b)

Figure 4.1.9: (a) Income Tax main screen (b) Income Tax guideline screen.

Figure 4.1.9 (a) shows the income tax main screen where user can read some information about income tax.

Figure 4.1.9 (b) shows the income tax guideline screen.



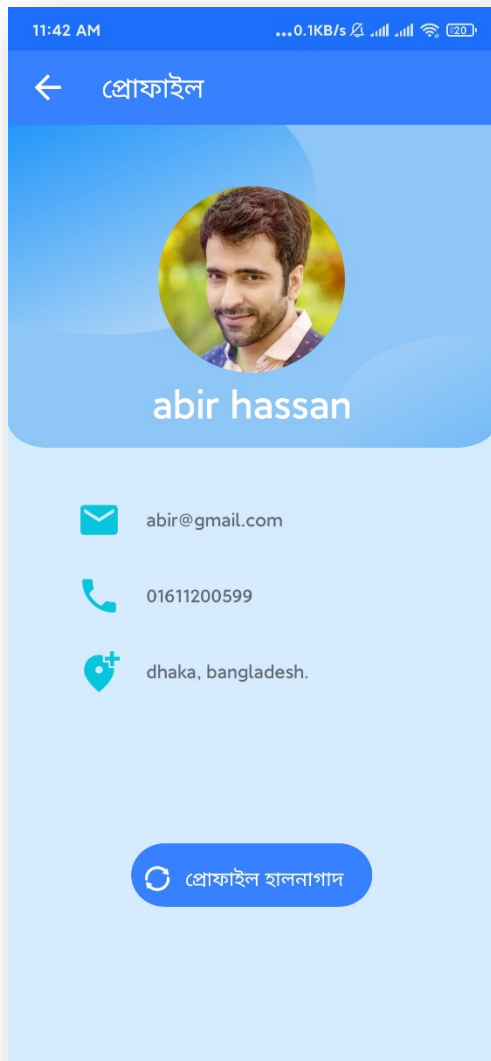
(a)



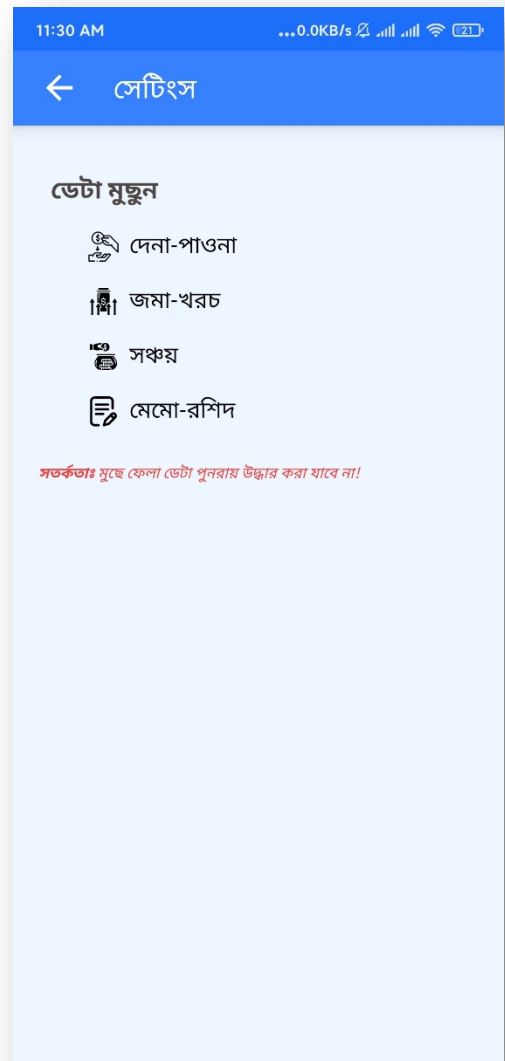
(b)

Figure 4.1.10: (a) Others main screen (b) Others content screen.

Figure 4.1.10 (a) shows the others main screen and (b) shows the content screen.



(a)



(b)

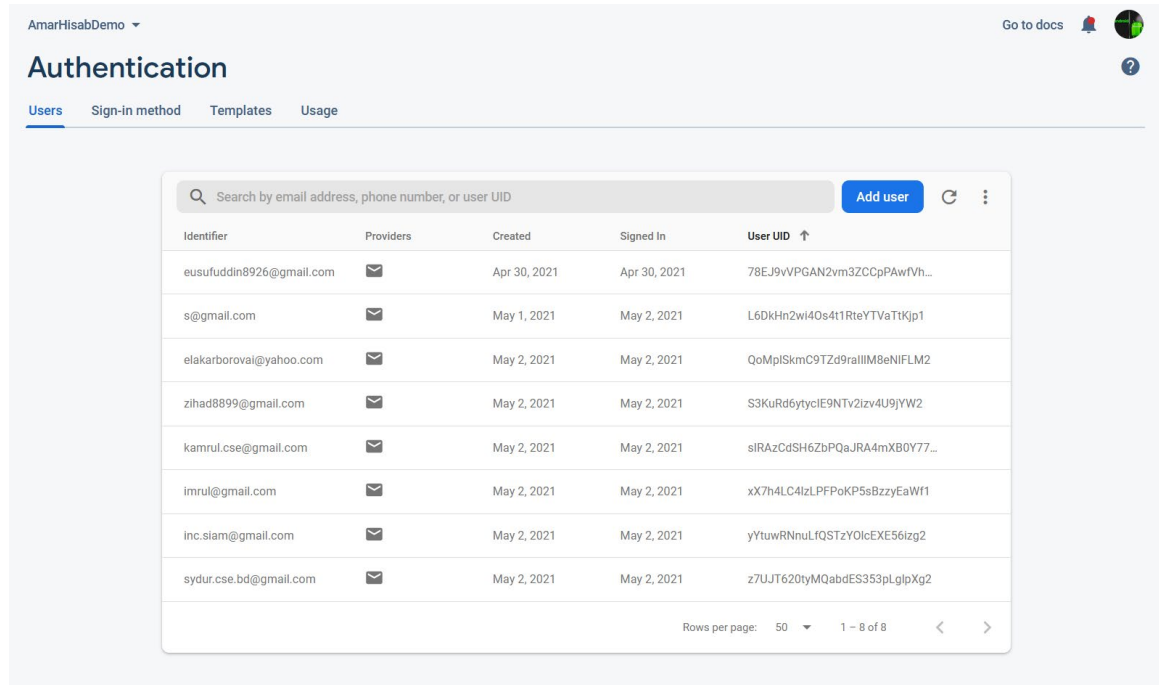
Figure 4.1.11: (a) Profile activity (b) Settings activity.

Figure 4.1.11 (a) Profile activity where can view or update profile information and (b) shows the settings activity.

4.2 Back-End Design

Backend design is an essential section of a dynamic android application. In our project, we used the Firebase database to store user login information and user transactions. When a user creates his/her account then this system saves data in firebase authentication. Firebase Authentication creates or generates a unique id to identify a user. Firebase gives some extra features like a real-time database where we stored user transaction data. We can also store the user's profile image and Memo-Roshid image in firebase storage.

User's list is shown in Firebase Authentication who are registered in our Application.



The screenshot shows the Firebase Authentication console for a project named 'AmarHisabDemo'. The 'Users' tab is selected, displaying a table of registered users. The table has columns for Identifier, Providers, Created, Signed In, and User UID. There are 8 users listed, all with email providers. The 'Signed In' column shows the date they last signed in, which is either April 30, 2021, or May 2, 2021. The 'User UID' column shows the unique identifier for each user.

Identifier	Providers	Created	Signed In	User UID ↑
eusufuddin8926@gmail.com	✉	Apr 30, 2021	Apr 30, 2021	78EJ9vVPGAN2vm3ZCCpPAwfVh...
s@gmail.com	✉	May 1, 2021	May 2, 2021	L6DkHn2wl40s4t1RteYTVaTtKjp1
elakarboroval@yahoo.com	✉	May 2, 2021	May 2, 2021	QoMplSkmC9TZd9raIIIM8eNIFLM2
zihad8899@gmail.com	✉	May 2, 2021	May 2, 2021	S3KuRd6tytciE9NTv2izv4U9jYW2
kamrul.cse@gmail.com	✉	May 2, 2021	May 2, 2021	siRAzCdSH6ZbPQaJRA4mXB0Y77...
imnul@gmail.com	✉	May 2, 2021	May 2, 2021	xx7h4LC4zLPPFoKP5sBzzyEaWf1
inc.siam@gmail.com	✉	May 2, 2021	May 2, 2021	yYtuwRNnuLfqStzYoiCExE56izg2
sydur.cse.bd@gmail.com	✉	May 2, 2021	May 2, 2021	z7UJT620tyMQabdES353pLgIpXg2

Figure 4.2.1: Show registered user's information

Backend database design for specific user's information.

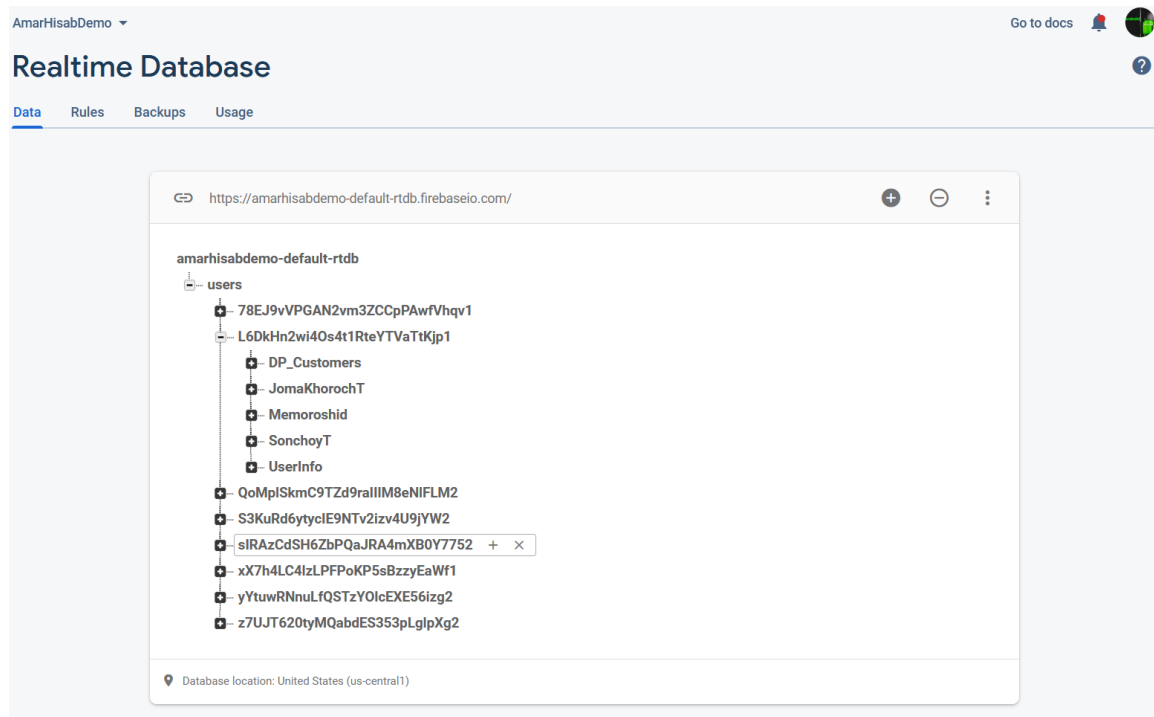


Figure 4.2.2: Show specific user's information

This database structure has a different type of data for specific Users. Users can read/write data by using their unique Id.

4.3 User Interface and Experience

The popularity of an application massively depends on its interaction design. Interaction design is a connector between user and application. A user-friendly interface helps a user to use it easily and saves more time. So we try to create an easy interaction design that's why user can easily connect with our application. Our application is for all kinds of people. With that in mind, we design a user interface that is very user-friendly and attractive.

4.4 Implementation Requirements

We used various types of tools/software to implement our application. We used Adobe XD to design the user interface and also make a prototype of our application. To implement the design we used android XML. All of the tools/software those we used are given below:

- Design : Adobe XD, Android XML, Adobe Illustrator.
- Development IDE : Android Studio
- Language : Java
- Database : Firebase
- Testing : Android Device & Emulator.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of Database

Here we are discussing the backend implementation of this application. To store user's data, we used Firebase Database. We know the firebase database is more secure to keep data and give well Structure. Firebase database works with offline and online in both mode if the user doesn't have internet, user able to store Transaction data in locally when he/she come back in online this data will be keep synced with the real-time database. This Application used only the firebase database to store all kinds of information that's why there is no need for any offline database or SQL.

Storing user Transaction, the User needs to first register with this Application using user information. To register with this application User, need to provide a username, user email and 6 digits of password for security reason. When the user gives all information successfully then it will be store in a real-time database to access from profile Activity.

Authentication

[Users](#) [Sign-in method](#) [Templates](#) [Usage](#)

Add user ↻ ⋮

Identifier	Providers	Created	Signed In	User UID ↑
eusufuddin8926@gmail.com	✉	Apr 30, 2021	Apr 30, 2021	78EJ9vVPGAN2vm3ZCCpPAwfVh...
s@gmail.com	✉	May 1, 2021	May 2, 2021	L6DkHn2wi40s4t1RteYTVaTtKjp1
elakarborovai@yahoo.com	✉	May 2, 2021	May 2, 2021	QoMplSkmC9TZd9raIIIM8eNIFLM2
zihad8899@gmail.com	✉	May 2, 2021	May 2, 2021	S3KuRd6tyclE9NTv2izv4U9jYW2
kamrul.cse@gmail.com	✉	May 2, 2021	May 2, 2021	sIRAzCdSH6ZbPQaJRA4mXB0Y77...
imrul@gmail.com	✉	May 2, 2021	May 2, 2021	xX7h4LC4IzLPPPoKP5sBzzyEaWf1
inc.siam@gmail.com	✉	May 2, 2021	May 2, 2021	yYtuwRNnuLfqSTzYOlcEXE56Izg2
sydur.cse.bd@gmail.com	✉	May 2, 2021	May 2, 2021	z7UJT620tyMQabdES353pLglpXg2 📄 ⋮

Rows per page: 50 1 – 8 of 8 ⏪ ⏩

Figure 5.1.1: show Register User List

Then user login to need a valid email and password that will be stored in firebase authentication. Firebase Authentication creates automatically user Id for every specific user that's is the most important part of our Application. When a user gives an invalid user email and/or password user will unable to login and the application will be shown specific message for this.

Here is our valid signing method

Authentication

Users [Sign-in method](#) Templates Usage

Sign-in providers











Provider	Status
 Email/Password	Enabled
 Phone	Disabled
 Google	Disabled
 Play Games	Disabled
 Game Center	Disabled
 Facebook	Disabled
 Twitter	Disabled
 GitHub	Disabled
 Yahoo	Disabled
 Microsoft	Disabled

Figure 5.1.2: Show Valid Signing Method for users

There will be a user Dena-Pawna information. The user will be able to record all transactions with a specific customer. User will save customer-wise transactions what exactly the user will give or the user will get. Here also added some extra information like transaction notes, transaction type, and specific date of transaction.

Realtime Database

[Data](#) [Rules](#) [Backups](#) [Usage](#)



Figure 5.1.3: Show Dena-Pawna Transaction for user

User stores income-expense record with a specific date, the specific type of income/expense and with a date.

Realtime Database

Data Rules Backups Usage

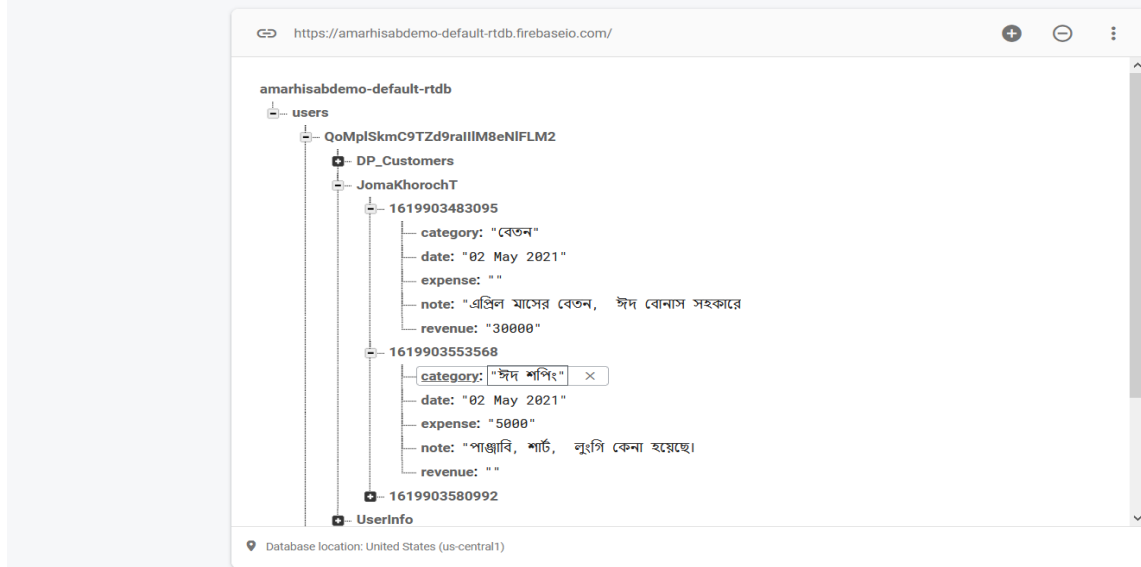


Figure 5.1.4: All Joma-Khoroch Transaction post

5.2 Implementation of Front-End Design

The application goal is to store all kinds of transactions that we are doing in our daily life. We are trying to store daily essential transactions with a person who is connected with us. We also store daily revenue and expense, saving and daily essential memo/receipts in a single application. In this application front-end design, we are trying to keep it simple and user-friendly. When a user first opens our app, a screen will show that has included our App icon and then open will be login activity. But doing log in users need to first registration then the user will be able to log in and show then our app main activity. In this activity, the user will able to store his/her debit-credit, income/expense, savings related transactions and also store essential paper daily. Users will also be knowing about how to calculate Income-Tax from our application.

In the front-end section, we have used the Bangla language for better understanding so that everything will be easy for the user to understand.

5.3 Testing Implementation

For any single error, an application may be cursed or throw unexpected output, also generating risk factors for end-users. By testing an application there is scope to detect and remain bugs. For testing purposes, we use different types of Android devices/emulators. Our selected devices are API levels 23 to 29. Which is covered more than 94% of users. There are few steps to testing an application. So in this section discussing few strategies which we follow to test our Application.

5.3.1 Unit Testing:

Here we check the individual functionality of our application in different android devices (Android version 6.0 to 10).

- At first, In Authentication, we checked the user's input email & password which the user provided during complete his/her registration. After successfully matched the user email and password the user gets access to use this application. Users can update his/her profile information. Which passed successfully.
- In Dena-Pawna, the user is able to add a new customer via phone contact or create a new customer by filling in customer name, phone & address. Users can make transactions with this customer and see a real-time summary of credential status & amount in the dashboard. Users can delete customers and also delete/update customer transactions. Which passed successfully.
- In Joma-Khoroch, the user is able to insert revenue & cost amount with valid data and see real-time Net Balance. Users can also delete/update Joma-Khoroch transactions. Which passed successfully.

- In sonchoy, the user able to insert savings & withdrawal amount with valid information and see real-time Net Savings. Users can also delete/update sonchoy transactions. Which passed successfully.
- In Memo-Roshid, the user able to upload a receipt image with the valid title and view all receipts in Memo-Roshid Gallery. Users can also delete/update receipts. Which passed successfully.
- From settings, the user can delete Dena-Pawna, Joma-Khoroch, sonchoy Memo-Roshid data individually. Which passed successfully.

After unit testing, the usability of our application is improved.

5.3.2 Verification Testing:

In verification testing, we match the expected outcome of our application. In Dena-Pawna, adding a new customer by fetching phone contact performs perfectly also users can add new customers using input fields. After adding a new customer user can make transactions. After all successful transactions, the user can see the final debt, credit & net balance in a real-time dashboard. Users can also delete & update any transactions. After successfully delete/update the real-time dashboard shows the correct credential status & amount. Similarly, Joma-Khoroch, Sonchoy fulfilled all requirement functionality & showing expected output in the real-time dashboard. In Memo-Roshid User can upload memo/receipt via camera or local storage with memo title & date. After successfully upload a memo/receipt the user can see his/her all stored receipts. Users can also delete/update unnecessary memo/receipts, which is performed correctly. So we ensure that the application fully matched the expected design & requirements.

5.3.3 Validation Testing:

It is a dynamic evaluation process. After successful verification testing, we checked user usability. For different types of input like, In Dena-Pawna, when a user inputs debit/credit amount to a specific customer then the final credential status & amount changed successfully. The system calculation & credential status fully matched with manual calculation in a notebook. Delete & update any transactions that work properly & changed the credential status & amount correctly. After back to the main Dena-Pawna dashboard, it shows all customer's final credential status & amount correctly. Also shows total debit/credit & net balance correctly. Where all system calculation & output matches with manual calculation. Similarly Joma-Khorch, Sonchoy calculation provided expected calculation, which matched with manual calculation. In Memo-Roshid Insert delete/update function works properly. Also, registration, login, update profile, & logout functions perform correctly. Overall, our application provides the expected outcome without any issue.

5.3.4 System Testing:

Before deploying our apps, we use system testing knowledge to recheck our system bugs & errors. After properly tested, we can ensure that our system can produce security, reliability. Which will satisfy our end users.

5.4 Testing Report

After different ways of testing, we can ensure that our application is ready to use. Our application will provide reliability and proper security to users. By using Firebase real-time database there is no way to losing/leaked user data. Though firebase is an online database our system has the offline capability to store data locally after the device became online the data automatically keep synced. For testing purposes, we deploy our application's demo version to some users. They give us positive feedback and eager to use this application. Till now we didn't notice any bugs/errors. Howbeit we will monitor testing & kept updates to fulfill user satisfaction.

CHAPTER 6

IMPACT ON SOCIETY, ENVIRONMENT & SUSTAINABILITY

6.1 Impact on Society

This application is developed for all kinds of people in which people can easily calculate their daily needed accounts. It will save their time. Its user-friendly interface inspires people to be accountable. As a result money waste will be reduced, use of register or diary also will be reduced. Users are able to know about income tax. As a result, they inspire to pay their income tax properly.

6.2 Ethical Aspects

The main ethical aspect of our application is it helps a person to be a frugal person. It also creates awareness and responsibilities about debtors and creditors. As a result, this can reduce money laundering fraud.

6.3 Sustainability Plan

This application will provide data security. Users can retrieve data where the user login successfully if any user lost phone. It is very easy to maintain our application. Developing this application requires some simple algorithms. It is very easy to maintain the server's functionality. No need for more employees to maintain it. That's why operational costs are not so high. For this reason, it shows very good sustainability.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Discussion and Conclusion

After a lot of pursuit and effort, we were able to create our application. This is a daily accounting application. It contains some important features that a person needs to perform every day. Like Dena-Pawna (Debtor-Creditor) which help the user to monitor their debtors and creditors, Joma-Khorch (Income-Expense) which help the user to balance their income and expense, Sonchoy (Savings) which help the user to monitor their savings and withdrawal, Memo-Rosid (Receipt) which help the user to store their receipts. Users can also know about income tax. We hope our application will make the user's day-to-day accounting system easier. It will reduce the use of registers and dairies. And its user-friendly interface saves time on a large scale.

7.2 Limitations

Every system has some limitations. Our system is not distinct from that. Our system contains some minor limitations. We tried to best to reduce limitations but some limitations remain in our application. Limitations of our system:

- Dena-Pawna Activity Search-bar has few limitations, like Sometimes it may not work properly to Search Transaction. When the user searches for a specific transaction, amount or credential status it doesn't show an updated value.
- When the user adds debtor/creditor but doesn't give any Transaction, this time transaction details will not work properly.

- User has limitations to give any transaction in the input field, we are taking Limited numbers or Characters because of UI-related problems.
- In income tax activity we added two pdf links, sometimes it Opens Slowly because of the net Problem or other reasons.

7.3 Scope for Further Developments

We have some plans with our application. We can't add some features right now because of knowledge and time limitation. In the future, we add these features to our application. We will add income tax calculator which calculates income tax automatically and user can calculate income tax separately. We will add an expense bounded system where the user can set a limit of expense and when expenses exceed the limit app will show an alert message. Also, we will try to add a scanner that will scan the receipt and automatically add it to the expense or income list. We hope these will increase the efficiency of our application and save users time massively.

REFERENCES

[1] A project report on “CreditInfo” by Mr. Momin Asad Ahmad, Imran Ahmad, Mr. Shaikh Mohammad Osman, Anique Ahmad; BCA (2019-2020).

[2] An Android Based Application "INCOME COUNTER" by MARZIA HAQUE TUMPA, MD. HASSAIN IBNE SALAM and SHAMSUN NAHAR SUMONA.

<<<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/2622/P11732%20%2826%25%29.pdf?sequence=1&isAllowed=y>>>

[3] “Online Income and Expense Tracker” by S. Chandini, T. Poojitha, D. Ranjith, V.J. Mohammed Akram, M.S. Vani, V. Rajyalakshmi.

<<<https://www.academia.edu/download/60555255/IRJET-V6I3111020190910-38299-1elg6et.pdf>>>

[4] "Daily Expense Tracker" An Android Based Mobile Application by Md. Abdul Karim AND Taslima Yesmin Orin.

<<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/4026/P14361%20%2814_%29.pdf?sequence=1&isAllowed=y>>

[5] Learn about Database (Firebase Database) from internet at:

<<[Documentation | Firebase \(google.com\)](#)>>, last accessed on 26-05-2021 at 02:40 PM.

[6] Learn about Back-end and Front-end design from the internet available at:

<<[Developer Guides | Android Developers](#)>>, last accessed on 01-06-2021 at 11:15 PM.

[7] Draw and learn about ER, DFD, Use-Case diagram using:

<<<https://app.diagrams.net/>>>, last accessed on 26-05-2021 at 02:40 PM.

AN ANDROID BASED APPLICATION “Amar Hisab - A PERSONAL FINANCIAL MANAGER”

ORIGINALITY REPORT

19%

SIMILARITY INDEX

18%

INTERNET SOURCES

0%

PUBLICATIONS

16%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	12%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	7%
3	ethesis.nitrkl.ac.in Internet Source	<1%
4	Submitted to Columbia High School Student Paper	<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography On