

MicroLink Information Technology College

Department of Computer Science

**TVC**

**(TELE VOUCHER CHECKER)**

A PROJECT SUBMITTED TO

MICROLINK INFORMATION TECHNOLOGY COLLEGE

IN PARTIAL FULFILLMENT FOR THE BACHELOR DEGREE

IN COMPUTER SCIENCE

**Prepared By**

**1. Hashim Abdrehman                  14,216/19**

**2. Natnael Amsalu                       13,075/19**

**3. Robel Mamushet                        14,455/19**

**4. Melkamu Maazahaymanot                      13,226/19**

**5. Amanuel Negussie                     13,331/19**

**Advisor:**

**Abdurezak Kemil**

**July 2022, Addis Ababa**

**Acknowledgments**

We (the team members) believe we’ve come up with a new idea of a project that could be used in a daily basis and benefits the society, especially the ones who use a voucher-based airtime users and sellers. We hope that in near future this project would be added in telecom services. Finally, we would like to thank our advisor Mr. Abdurezak Kemil for he has given us a good guidance and support we needed.

Addis Ababa, July 2022

Table of Contents

Contents

[CHAPTER ONE 4](#_Toc108382007)

[1. INTRODUCTION 4](#_Toc108382008)

[1.1 Background 4](#_Toc108382009)

[1.2.2 Problems with the Existing System 5](#_Toc108382010)

[1.3 The Proposed System 5](#_Toc108382011)

[**1.3.1** **General Objectives** 5](#_Toc108382012)

[**1.3.2** **Specific Objective** 6](#_Toc108382013)

[**1.3.3** **Significance of the Project** 6](#_Toc108382014)

[**1.3.4 Scope of the Project** 6](#_Toc108382015)

[1.4 Methodologies and Techniques 7](#_Toc108382016)

[**1.4.1** **Data Collection Techniques and Methods** 7](#_Toc108382017)

[**1.4.2** **System Analysis Design and Development Tools** 7](#_Toc108382018)

[2. SYSTEM ANALYSIS 8](#_Toc108382019)

[2.1 Introduction 8](#_Toc108382020)

[**2.1.1** **Purpose** 8](#_Toc108382021)

[**2.1.2** **Scope** 9](#_Toc108382022)

[**2.1.3** **Overview** 9](#_Toc108382023)

[2.2 Current System 10](#_Toc108382024)

[2.3 Proposed System 11](#_Toc108382025)

[**2.3.1** **Overview** 12](#_Toc108382026)

[**2.3.2** **Functional Requirements** 12](#_Toc108382027)

[**2.3.3** **Nonfunctional Requirements** 13](#_Toc108382028)

[**2.3.4** **Constraints (“Pseudo requirements”)** 15](#_Toc108382029)

[**2.3.5** **System models** 15](#_Toc108382030)

[CHAPTER THREE 33](#_Toc108382031)

[3. SYSTEM DESIGN 34](#_Toc108382032)

[3.1 Introduction 34](#_Toc108382033)

[**3.1.1** **Purpose** 34](#_Toc108382034)

[**3.1.2** **Scope** 34](#_Toc108382035)

[3.2 Goals and Trade-offs 35](#_Toc108382036)

[3.3 System Decomposition 35](#_Toc108382037)

[**3.3.1** **Layers and Partitions** 35](#_Toc108382038)

[3.4 Concurrency Identification 36](#_Toc108382039)

[3.5 Hardware/Software Allocation 36](#_Toc108382040)

[**3.5.1** **System Performance** 37](#_Toc108382041)

[**3.5.2** **Connectivity** 37](#_Toc108382042)

[**3.5.3** **Network architecture** 37](#_Toc108382043)

[3.6 Data Management 37](#_Toc108382044)

[3.7 Global Resource Handling 37](#_Toc108382045)

[3.8 Software Control Implementation 38](#_Toc108382046)

[**3.8.1** **External control flow (between subsystems)** 38](#_Toc108382047)

[**3.8.2** **Concurrent control** 38](#_Toc108382048)

[**3.8.3** **Internal control (within a single process)** 38](#_Toc108382049)

[**3.8.4** **User Interface** 38](#_Toc108382050)

[3.9 Boundary Conditions 43](#_Toc108382051)

[**3.9.1** **Initialization** 43](#_Toc108382052)

[**3.9.2** **Termination** 43](#_Toc108382053)

[**3.9.3** **Failure** 43](#_Toc108382054)

[3.10 Design Rationale 43](#_Toc108382055)

[CHAPTER FOUR 44](#_Toc108382056)

[4. IMPLEMENTATION 44](#_Toc108382057)

[4.1 Overview 44](#_Toc108382058)

[4.2 Tools and Technology Utilized for Implementation 44](#_Toc108382059)

[4.3 System Implementation 44](#_Toc108382060)

[**4.3.1** **System Interfaces** 44](#_Toc108382061)

[**4.3.2** **Source Code** 44](#_Toc108382062)

[CHAPTER FIVE 45](#_Toc108382063)

[5. Conclusion and Future Work 45](#_Toc108382064)

[5.1 Conclusion 45](#_Toc108382065)

[5.2 Future Work 45](#_Toc108382066)

[GLOSSARY 46](#_Toc108382067)

[REFERENCES 48](#_Toc108382068)

# CHAPTER ONE

# INTRODUCTION

## Background

Telecommunications services were first established in Ethiopia in 1894 when Ethiopian King Menelik II brought telephone technology to the country. His relative Las Makonnen, however, was the first Ethiopian phone pioneer. He went home with a phone and established a business after visiting Italy in 1889. In the early twentieth century, the corporation was taken over by the government and later supported by the Ministry of Posts and Communications. Telecommunications services were separated from post office services in 1952 and put under the Ministry of Transport and Communications. Ethiopian Telecommunications Corporation is the continent's oldest public telecommunications company (PTO). ETC provides landline, internet, and cell phone service. Learn how coupons are used to pay for calls on mobile phone top-up systems and others d It has a fixed monetary value and can only be used for certain activities or products.

ETC uses a mobile voucher to charge a customer's phone number to replenish a SIM card with money and improve the card's availability. The majority of vouchers are sold at retail stores, telecommunication system providers, dealers, grocery stores, random shops, etc. In various countries, such as Italy, Spain, and Ethiopia, where more than 90% of customers use vouchers, and in the United Kingdom, where more than 60% of consumers buy coupons at retail outlets, vouchers are prepaid mobile phones. It is the most common type of payment. Cards that do not give charge alternatives are not accepted in other countries, including the United States, Ireland, and several Scandinavian countries cards without charge alternatives, such as clients making online purchases or calling operators and contacts using mobile phones (CSR). It is becoming increasingly popular. Alternatively, you can use the IVR (Interactive Voice Response) system. In Ireland, the number of prepaid mobile phone carriers such as Meteor and T-Mobile USA is increasing, allowing users to send SMS (Text to Pay) and use mobile phone applications such as WAP and BREW technologies.

**1.2 The Existing System**

**1.2.1 Overview of the existing system**

**E-voucher**

The tech-savvy customers today don’t want to deal with the hassle of manual recharge systems. The actors of the electronic voucher distribution ecosystem are a telecom operator, a distribution network, multiple access points (channels) through which resellers request and receive a response, a processing system to validate and respond to queries, and lastly, a customer.

The subscribers can easily get their accounts recharged by requesting a reseller to top up their accounts through an electronic voucher. Then the reseller can forward this request to the telecom operator via an authorized channel like USSD, web channel, or SMS for the desired voucher denomination. After validation, the system approves the request of the reseller and delivers a code via SMS to the subscriber or pushes a printable voucher to the reseller end where the reseller prints the voucher and presents it to the subscriber. The resellers may use a Wireless POS terminal with a print functionality to download the voucher, print it, and sell it to the subscriber. Alternatively, the subscriber receives a voucher pin code directly on the mobile that they can use to top up their account.

**Electronic Voucher Distribution (EVD) System**

The electronic Voucher Distribution (EVD) System is a way of distributing prepaid voucher cards electronically to end users through various sales channels. Telecom operators use this service as an alternative option to distribute airtime pin cards and related services.

Ethio Telecom, the leading telecom operator in Ethiopia, has recently launched an EVD service to distribute electronic voucher cards in partnership with private companies. The primary requirements set by Ethio-telecom to engage with the EVD service are a legal VAS license, server and related infrastructure, EVD software, and 250+ Point of Sale (POS) or Bluetooth machines. In addition, the partner is required to have a minimum of 1 million Birr as registered capital to run the business.

## 1.2.2 Problems with the Existing System

As we know in Ethiopia, we have a shortage of foreign currency so we cannot risk that little currency for importing scratch cards. In addition to this, to keep pace with the highly competitive and dynamic environment, telecom companies need to optimize their distribution and logistics costs as well.

In the current situation,

* POS with print functionality is more used and that printed paper is distributed to retailers than to customers which are not very secure. As the code is visible there may be open sides with this chain of distribution and complaints are rising about used vouchers being sold to the users.
* The time it takes to call a customer service center is frustrating which makes it difficult to check a used voucher in the desired time.
* From the seller’s side, there will be a loss in their business and a lack of trust in the company and in the business itself.
* From the company side a lot of complaints via their customer service center which means they have to deploy more operators to handle those phone calls.
* Users will be unable to charge their account to consume telecom services which makes them very disappointed and lose trust in the company.

## 1.3 The Proposed System

### **1.3.1 General Objectives**

The project’s general objective is:

* To check the validity of generated recharging vouchers.
* To make it possible that airtime should be changed to real money or cash out the airtime.

### **1.3.2 Specific Objective**

* To develop a system that builds trust between the sellers and the customers.
* To make the recharging voucher checking system simple and time-saving.
* To keep track of every generated voucher whether they are recharged or not.
* To eliminate fake compliances that put the sellers’ trustworthiness they have in society on the line.
* To modernize the existing way of voucher checking system.
* Keeps history every time the vouchers are recharged.
* Customers will be able to cash out the air time via a connected banking system just like it’s possible to recharge from our bank account.

### **1.3.3 Significance of the Project**

* Will allow the registered seller to check the overall generated voucher from the central system through our mobile application.
* Will save time and energy much more than the existing system.
* Will be able to store all the generated vouchers on the central database under the distributor they purchased from so that all the registered sellers can have the privilege of checking them.
* Will be simple and user-friendly for almost all kinds of sellers.
* Customers can convert air time to birr or cashing out.
* By unlocking the possibility of changing air time to birr this can open the door for new transaction methods and new business opportunities along with technological growth for the country.

### **1.3.4 Scope of the Project**

The system we will be implemented with this project enables customers to have their voucher recharging history data stored in a safe and central place in which Ethio Telecommunication controls as the main service provider. This project initially focuses on the integrity of the service between the seller and the customer. Also, on controlling the sellers who sell fake vouchers and the fake compliances of the customers.

In addition to the described feature above, our system will unlock the possibility of changing airtime to birr. This will open the door to a new transaction system and new business opportunities. This will benefit the banks that are integrated into this system.

Generally, our system will store all voucher-based recharging history of the generated vouchers by registered sellers and will also be able to provide the desired information to the seller in less than a minute. The system will keep a limited record of reported sellers with levels of warnings. The last level of warning will be terminating the service with punishment. With this kind of method, our system will automatically control the safety and integrity of our service to our customers.

Our system’s functionalities are:

* Sellers can generate voucher codes and sell them more safely.
* Customers can cash out their airtime to birr.
* Will have all the features of the existing Ethio-telecom system provides like:
* Recharging
* Buying packages
* Credit services, etc.

**Limitation of the Project**

Our system will be limited to controlling the reprinted vouchers that will be generated by sellers from the generating machine.

## 1.4 Methodologies and Techniques

To make our general and specific objective happen we will divide our project into three main stages:

* **Requirement and system analysis**: we will analyze the requirements that the new system should have. Preparing surveys on side effects of how non-scratch voucher pin retailing is affecting business owners. We will capture requirements on gathered data from sellers that have dealt with Ethio-telecom company airtime voucher pin distributors and retailers, the company itself, and also some customers of Ethio-telecom. Then we will prepare a SyRS document.
* **System design and development**: in this stage, we will specify the detailed necessary features and to:
* build an architectural model of the system
* prepare logical and physical models
* Components (hardware and software)
* Working procedures of the system to accomplish its main objective
* We will design a database system
* **Implementation and testing**
* Implementing and testing on specific hard wares
* Prepare documentation
* Carry out unit testing and integration testing.

### **1.4.1 Data Collection Techniques and Methods**

Interviews with sellers that have dealt with Ethio-telecom company airtime voucher pin distributors and retailers, the company itself, and also some customers of Ethio-telecom and data analysis will be presented descriptively. We will make statistics based on our observations. Inspecting how they are encountering problems during selling voucher pins. And identifying open sides of these non-scratch vouchers.

### **1.4.2 System Analysis Design and Development Tools**

By using Visual Paradigm 16.3 we describe:

* **Use Cases**: This tool helps us to describe how a system will be used. Also, help us to refine system features. We also use this tool throughout the project life cycle, especially during testing.
* **Flow of events**: we can use this tool to take a process and describe it as a diagram. This helps us to describe a series of steps or decisions in visual form in a manner that facilitates communication.

CHAPTER TWO

# SYSTEM ANALYSIS

## Introduction

### **Purpose**

These days in Ethio-telecom there are visible voucher numbers on the market that are used for airtime recharge that is being sold to customers. This concerns most buyers every time they buy an airtime voucher; whether the visible codes on the papers are valid or invalid. If the codes are invalid, they don't have proof to show to the seller that the voucher they bought is used or not. So, in order to check the validity of the code, they have to call customer service, which is challenging to reach in the desired time. This problem also concerns the seller. Because it's very difficult to reach customer service to check whether the buyer is faking or not. The existing system consumes time, energy, and money and our project will be able to solve these current problems that frequently occur in the existing system. Some sellers reprint vouchers of the same code multiple times and sell them to customers for their own profit. The system will be able to track down the corruptions and mark them by levels of their crime to easily give warnings and fair punishments. When compared to the current system, it will be fast and time-saving for the voucher validity checking both for the customer and the seller.

The purpose of this project is to check the validity of the vouchers that people buy from sellers and to eliminate corruption from the sellers and fake complaints from customers. In other words, this project mainly helps to ensure the integrity of the service between both sides (the seller and the customer). In addition to the integrity of our service, this project will enable the sellers to directly buy airtime from the service provider without any distributor involvement which the current system doesn’t support and this system will enable anybody who satisfies the requirements needed to be a seller.

The voucher codes will be given serial numbers every time they are generated by sellers. These serial numbers will be unique numbers that describe the generators. If a seller gets reported by a customer our system will check for the voucher's serial number to check the identity. If the seller is responsible after checking the serial number, the seller will be sent a warning notification. This method also works for the customer. Our system will analyze and check the validity of the reports and send warnings to the responsible ones.

In order to use our system, the buyer and the seller must have to register. For each seller and customer, there will be a unique identification number to easily track their identity. With this method, it's easier to control and manage users as a categorized entity to ensure the safety of the service. The categorized entities will be added to a central database to record and keep track of the generated voucher codes.

Since most people use smartphones, our system will be a mobile application that will run on androids and iPhones. So, to use our app the user’s phone android version must be at least 5.0 and above.

### **Scope**

TVC is the name of our app which we named after its main objective and stands for Tele-Voucher-Checker. We planned to work on this project after observing several concerns from the people on the service provider - Ethio-telecom. Then we narrowed down what and where the problem is, and we came up with a solution that benefits both entities. This project satisfies almost all the concerns of the people. Which is checking the validity of voucher codes that customers buy from roadside sellers, shops, and any other retailers. The vouchers are small papers that have visible recharging codes on them that lack security and make them more vulnerable to fraud activities (by the sellers and customers). But this project is intended to solve this problem by creating a manageable environment for the service provider, the seller, and also for the customer.

Our app will have the following features:

* **It checks**:
* Voucher codes validity
* Recharged or not
* Usage time
* **Supports most of the existing Ethio-telecom services**:
* Recharge
* Buy package services
* Borrow airtime
* Sends gift package, airtime, call me back request
* Divert call services
* CRBT service
* Customer service
* **Generate Voucher codes (seller-based feature)**:
* Keep records of generated and used codes along with the used phone number and time.
* **Reporting (customer-based feature)**:
* Customers can report used vouchers.
* Then the service provider blocks the seller after 0.1% reports of overall voucher sales.
* Notify the seller of the remaining percentage to block.
* **Cash out (customer-based feature):**
* Customers can cash out (withdraw) their airtime to their bank accounts.

### **Overview**

In this document, we have provided detailed explanations and required information about our project. It contains how the existing system works and its current problems, the proposed system, and the problems that need to be fixed, how the proposed system works, what our system requires to fully function, how the data will be stored, and who and how our system will be used, etc. with clear visualizations that are supported by UML diagrams like:

* Use case diagram,
* Class diagram,
* Object diagram and
* Sequence diagram.

It will also provide definitions of all terms, acronyms, and abbreviations required to properly interpret the RAD so that the readers can easily understand the document without any confusion.

## Current System

**E-voucher**

The actors of the electronic voucher distribution ecosystem are a telecom operator, a distribution network, multiple access points (channels) through which resellers request and receive a response, a processing system to validate and respond to queries, and lastly, a customer.

* The agents with huge capital can directly deal with the telecom company, by issuing business proposals and acting as agents on the side with the company reaching to the seller.
* Sellers will have to deal with agents by issuing the business license and depositing money to the agent's account after that the air time will be transferred to the reseller account with their profit included.
* Seller can then sell that air time with direct recharge to customers, or convert airtime to voucher code and print it on POS machine by using Bluetooth from the app developed by the company or on POS machine pre-installed software prepared by the agent and
* The subscribers can easily get their accounts recharged by requesting a seller to top up their accounts through an electronic voucher. Then the seller can forward this request to the telecom operator via an authorized channel like USSD, web channel, or SMS for the desired voucher denomination. After validation, the system approves the request of the seller and delivers a code via SMS to the subscriber or pushes a printable voucher to the seller’s end where the seller prints the voucher and presents it to the subscriber. The sellers may use a Wireless POS terminal with a print functionality to download the voucher, print it, and sell it to the subscriber. Alternatively, the subscriber receives a voucher pin code directly on the mobile that they can use to top up their account.

**Electronic Voucher Distribution (EVD) System**

The Electronic Voucher Distribution (EVD) System is a way of distributing prepaid voucher cards electronically to end users through various sales channels. Telecom operators use this service as an alternative option to distribute airtime pin cards and related services.

Ethio-telecom, the leading telecom operator in Ethiopia, has recently launched an EVD service to distribute electronic voucher cards in partnership with private companies. The primary requirements set by Ethio-telecom to engage with the EVD service are a legal VAS license, server and related infrastructure, EVD software, and 250+ Point of Sale (POS) or Bluetooth machines. In addition, the partner is required to have a minimum of 1 million birr as registered capital to run the business.

**Problems of the Existing System**

As we know in Ethiopia, we have a shortage of foreign currency so we cannot risk that little currency for importing scratch cards. In addition to this, to keep pace with the highly competitive and dynamic environment, telecom companies need to optimize their distribution and logistics costs as well.

In the current situation, POS with print functionality is more used and that printed paper is distributed to retailers than to customers which are not very secure. As the code is visible there may be open sides with this chain of distribution and complaints are rising about used vouchers being sold to the users.

The time it takes to call a customer service center is very long and almost boring, which makes it difficult to check a used voucher. From the sellers’ side there will be a loss in their business and a lack of trust in the company and in the business itself. From the company side a lot of complaints via their customer service center which means they have to deploy more operators to handle those phone calls. Users will be unable to charge their account to consume telecom services which makes them very disappointed and lose trust in the company.

## Proposed System

The proposed system is intended to work integrated with the Ethio-telecom to make it secure and more accessible. The system can be used to exchange airtime and create accountability, save history, make payments by using airtime and punish those who try to scam. To access the system, the user must have an account or must register. After logging in, they can choose different services. The user can buy a package, report a scam voucher card, make payment by using airtime, recharge, and cash out the airtime. Depending on the choice of the user and according to company specifications the user will get a simpler service. Any person who fulfills the requirements can print the voucher card and sell it. Customers can send any complaint by choosing from the reporting reasons list. In addition, customers can withdraw their airtime money into cash through their bank accounts. The Administrator of the System shall set qualifications for buying the airtime and print, by giving the serial numbers when the airtime voucher is printed. And controls when it gets the administrator to cut off any fraud sellers after giving them a warning.

### **Overview**

The system consists of an additional feature to the existing system, which is used to check voucher code validity before recharging them. The system is designed to help manage airtime vouchers for a seller and a customer. In addition to checking the voucher codes, the system also provides reporting, feedback, and cash-out (withdrawal) services which the current system does not provide in an easy way.

Such as when sellers print used vouchers and if the distributor sells printed serial numbers. Analyzing each sequence of data makes it more reliable and holds accountability. It is transferred from one user to another user and keeps track of the exchange. Furthermore, in Ethiopia, there is no small fee like cloud system, gaming and etc. service example to unlock this stage you have to pay 2 birrs. If some company suggests that without any bank system or an online payment system users can get service by using airtime payment.

### **Functional Requirements**

Functional requirements describe the interactions between the system and its environment independent of its implementation. The environment includes the user and any other external system with which the system interacts. The following points list down the functional requirements of our system:

**Register customer:** The person who uses the airtime voucher to get airtime and check the validity of the voucher card and who can report.

**Register seller:** The person who buys airtime from the service provider and who sells recharging vouchers to the customers.

**Authorization check:** our system will check whether it’s the customer or the seller accessing the right user information and give the privileges as required.

**Cast seller:** when the seller sells used vouchers our system will give 50 warnings which are categorized by levels from simple to hard warnings. When the warning reaches 50 the SP will stop its service automatically and the seller will not be able to sell vouchers anymore.

**Report seller:** the customer will be able to report fraud sellers to the SP.

**Feedback**: the users can give feedback that will help the developers with maintenance and better performance.

**Admin authentication:** our system will have its own credentials to store data and keep track of the transactions which will be maintained by the developers.

**Counting**: the system will count the users who registered, logged in, and who made transactions.

### **Nonfunctional Requirements**

Non-functional requirements (also known as quality requirements) impose constraints on the design or implementation (such as performance requirements, security, or reliability).

* **Usability**: it is expected that the user should be able to use the airtime easily (like buying airtime, transferring air, time, and other services with the press of a button). Users should get fast, clear service and we provide guidance for users.
* **Reliability**: the app should be reliable and secured because the user trusts to use our service, so the app should satisfy the requirement and pass Ethio-telecom security standards. if the seller lost his/her phone the airtime must not be lost.
* **Performance**: There might be many users accessing the server simultaneously. As online recharge airtime, Voucher codes validity, and existing system Recharged or not Usage time tool performance shouldn't be affected much and response time for submitted pages should be less than a minute.
* **Security**: The system should provide basic security features like password authentication. All the passwords should have 8 characters. That combination of alphabet and numbers generated and communicated to the users should be stored in the server only in an encrypted form to prevent unauthorized access from unauthorized users.
* **Safety**: In order to prevent data loss in case of system failure, the process will terminate any kind of air time exchange if the seller is trying to find out if the voucher card is valid or not, he/she will get a notification that said “please try again later, when service is available”, the system will be terminated any kind of connection. The system should be capable of gracefully recovering from earlier crashes and continuing the service > Speed: The system needs a fast or fair internet connection for smooth use. The system should satisfy the user's expectations.

#### **User Interface and Human Factors**

There are two types of users, sellers, and customers. Customers can cash out the airtime to money. The system is used by any user with fair knowledge of how to use a mobile phone; the project team prepares will put a user manual on how to use the app on the help icon, other users such as sellers need some kind of training like how to check if the air time is valid or not, some users to if they want to cash out the air time to real money, as a system administrator will disable the exchanging rate of air time with real money. The project team will try to make the interface user-friendly. Since the system is deployed on mobile phones, users who own these devices can use the system. During deploying the system on mobile phones in the form of an app the user is going to check how to use the app in less than 2 minutes.

#### **Documentation (manual)**

There will be two types of documentation. The first one is for the seller about how to check if the airtime is valid. The second one will be for customers on how to check voucher code validity and also how to convert from airtime to real money.

#### **Hardware considerations**

Since most people use smartphones, our system will be a mobile application that will run on android and iOS. So, to use our app the user’s phone must be at least 5.0 android version or above.

#### **Performance characteristics**

The system will respond immediately by applying a high-speed processor in all subsystems and the hardware requirements are fully operating the speed of the system will increase. The system memory size will be increased in the future according to the number of users of the system.

#### **Error handling and extreme conditions**

Error handling the system should handle run-time errors and invalid inputs from users by giving appropriate error messages whenever necessary. If the users are committing an error, they are aware of being told by the system that they commit an error. The other case is the extreme condition and if an error happens during operation the operation will be totally terminated and will show an error message “there is no network, the service is not available right now, try again later.”

#### **System interfacing**

In the current implementation, there will be neither input coming from outside nor output going to systems outside our system. Our system will be connected to a real-time database which will fully be controlled by us (the developers) and it will be able to automatically update multiple user information simultaneously. But, for future modification, our system will be connected to outside systems. In addition, the app interface will be simple in which users can follow the instructions and interact.

#### **Quality issues**

The app that is developed by the project team is expected to be fault-tolerant. As much as possible the team tries to develop an error-free application, can't totally remove errors, but minimizes to the acceptable standard. If some components of the systems are failing Ethio-telecom has different server farms so if one server fails there will be another one.

#### **System modifications**

Some features could be added for the next release. We have a plan to pay the desisted person who has been scammed by fraud sellers and punish the fraud seller by the law. Also, buy and sell goods by using airtime as a digital currency.

#### **Physical environment**

The target machine which is going to serve as a server will be placed in the telecom server farm, expectedly in the general administrator's office, anywhere in Ethiopia where it has sufficient network for connection to the main server. The server room has to be secured from any physical connection unless we are going to jeopardize user money and identity.

#### **Security issues**

The system will block unauthorized users from accessing the database to get private information and data and we will be creating different security methods for different types of users and for users who are authorized to access the data or the information. Users that are members of the system will be asked for a username and password to access some information from the system database.

#### **Resources and management issues**

Backup is expected to be taken every day and the administrator is responsible for this because continued day-to-day activities will be performed in the registration, printing of airtime vouchers, at what time the airtime voucher is used, and cash outs. Our app will be released on both leading platform stores. That means Play Store for Android devices and App Store for Apple devices. The application could be installed by any user and can be used without any special knowledge. In case of maintenance, our app will let the users give feedback to the developer’s team.

### **Constraints (“Pseudo requirements”)**

* The main requirement is a fair internet speed.
* If the user doesn't have basic knowledge of mobile phone apps it wouldn’t be hard to get on with our user-friendly interface. To develop our system the requirements that are going to use are:
* Flutter
* Dart
* Firebase
* The user of this app should have any kind of mobile phone with an android version of 5.0 and above.

### **System models**

In the system model section analyze any RAD document scenarios, and use case models, and data dictionary dynamic models one by one sequentially.

#### **Scenarios**

1. **Login/Logout Scenario**

|  |  |
| --- | --- |
| **Scenario name** | User Login/ Logout |
| **Actors** | Seller and Customer |
| **Flow of events** | The user initiates the application   1. The system redirects the user to the login page and clicks on the login button. 2. The users enter their phone number and password. 3. The system authenticates the login request and redirects to the homepage based on their entered credentials. 4. The system will log out after use. |

**2. Customer Voucher Checking Scenario**

|  |  |
| --- | --- |
| **Scenario name** | Customer Voucher Checking |
| **Actors** | Customers |
| **Flow of events** | The user initiates the application.   1. The system redirects the user to the login page. 2. The customer enters a phone number and password and clicks on the login button. 3. The system authenticates the login request and redirects to the homepage. 4. Customer clicks on the check voucher code button. 5. The system redirects to the voucher checking page. 6. The customers enter a voucher code and click the check button. 7. The system checks the validity of the voucher code. 8. The system redirects to the voucher checking page with limited information if valid. 9. The system will log out after use. |

**3. Seller Voucher Checking Scenario**

|  |  |
| --- | --- |
| **Scenario name** | Seller Voucher Checking |
| **Actors** | Sellers |
| **Flow of events** | 1. The system redirects the user to the login page. 2. The seller enters a phone number and password, and clicks on the login button. 3. The system authenticates the login request and redirects to the homepage. 4. The seller clicks on the check voucher code button. 5. The system redirects to the voucher checking page. 6. The customers enter a voucher code and click the check button. 7. The system checks the validity of the voucher code. 8. The system redirects to the voucher information page with detailed information. 9. The system will log out after use. |

**4. Report Scenario**

|  |  |
| --- | --- |
| **Scenario name** | Report |
| **Actors** | Customer |
| **Flow of events** | The customer initiates the application.   1. The system redirects the user to the login page. 2. The customer enters a phone number and password, and clicks on the login button. 3. The system authenticates the login request and redirects to the homepage. 4. The customer clicks on the check voucher code button. 5. The system redirects to the voucher checking page. 6. The customers enter a voucher code and click the check button. 7. The system checks the validity of the voucher code. 8. The system redirects to the report page if invalid. 9. The customer enters the voucher code, chooses the issue from a list, and clicks on the report button. 10. The system will store the reported data and send a warning notification to the generator of the code. 11. The system will log out after use. |

**5. Registration Scenario**

|  |  |
| --- | --- |
| **Scenario name** | Registration |
| **Actors** | Sellers and customers |
| **Flow of events** | The user initiates the application.   1. The system redirects the user to the login page. 2. The user clicks on the create account button. 3. The system redirects to the create account page. 4. The user enters the required information and clicks on the create account button. 5. The system authenticates the user’s registration request and redirects to the homepage. 6. The system will log out after use. |

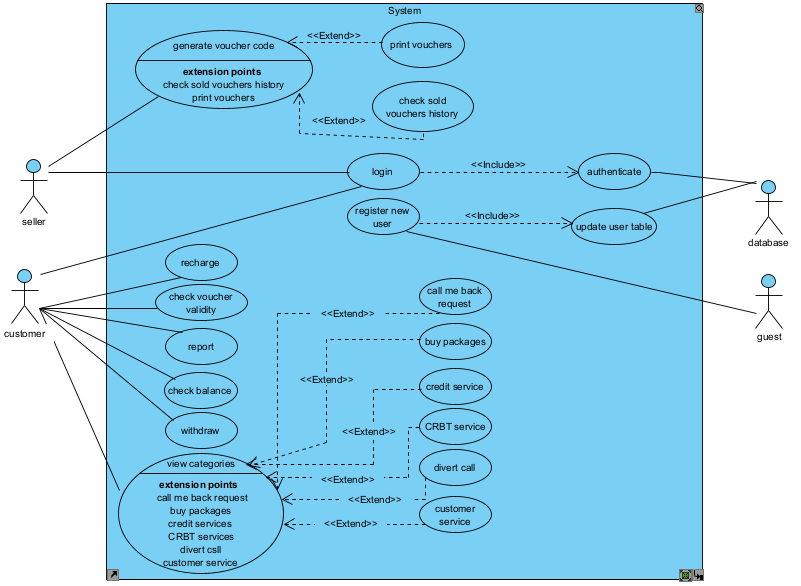
**6. Withdraw Airtime Scenario**

|  |  |
| --- | --- |
| **Scenario name** | Withdraw airtime |
| **Actors** | Customers |
| **Flow of events** | The customer initiates the application.   1. The system redirects the user to the login page. 2. The customer enters a phone number and password, and clicks on the login button. 3. The system authenticates the login request and redirects to the homepage. 4. The customer clicks on the withdraw button and the system redirects to the withdraw page. 5. The customer chooses the bank, enters the bank account number, enters the amount and clicks on the withdraw button. 6. The system authenticates the request and withdraws the entered amount of the airtime. 7. The system will log out after use. |

**7. Generate Voucher Code Scenario**

|  |  |
| --- | --- |
| **Scenario name** | Generate voucher code |
| **Actors** | Sellers |
| **Flow of events** | The seller initiates the application.   1. The system redirects the user to the login page. 2. The seller enters a phone number and password, and clicks on the login button. 3. The system authenticates the login request and redirects to the homepage. 4. The seller clicks on the generate voucher code button and the system redirects to the voucher code generation page. 5. The seller enters quantity, amount and clicks on the generate button. 6. The system authenticates the request and redirects to the printing page with generated voucher codes. 7. The seller clicks on the print button and the POS machine starts to print the vouchers. 8. The system will log out after use. |

#### **Use Case Model**



**Figure 1.1 main use case diagram**

1. **Use case name:** login

**Description:** used to authenticate the user

**Actors:** administer, customer, seller

**Precondition:** the actor should activate the system

**Basic courses of action:**

1. The actor instant login the page
2. Login page will be activated
3. Actor will enter the username and password
4. The system will check the validity of the data
5. If data is valid login according to its privilege
6. Lastly show home from displayed

**Alternative courses of action**

1. The actor instant login the page
2. Login page will be activated
3. Actor will enter the username and password
4. The system will check the validity of the data
5. If data is invalid will display an error message

1. **Use case name:** check voucher validity

**Actors:** seller and customer

**Description:** the seller can request voucher code validity and get full information of the code. On the other hand, the customer can only check whether the voucher code is used or not.

1. Write the serial number
2. Write SN (serial number)
3. Ask the user to enterthe serial number
4. Check if the number of digits on the serial number is a valid amount
5. Check the validity of the voucher card
6. Display a message that tells the time and who used it

**Alternative courses of action**

1. Write the serial number
2. Write SN (serial number)
3. Ask the user to enterthe serial number
4. Check if the number of digits the on serial number is invalid (more or less digit) amount
5. Display a message that said wrong SN
6. **Use case name:** withdraw

**Actors:** customer, database

**Description:** This use case is to cash-out the airtime for real money by exchanging airtime for real money when integrated.

**Precondition:** must have valid user name and password and exchange rate for e.g., customer will change 10-birr air time with 9.00 real currency.

**Alternative courses of action**

1. First the customer has to enter their username and password correctly
2. Customer will change the air time to real money
3. To do that first he has to use an exchange rate (like 10 birrs of air time can exchange by 9.00real money)
4. Money will transfer to the customer’s bank account.
5. **Use case name:** check voucher history

**Actors:** seller, database

**Description:** This use case is initiated by the administrator; this use case will enable access to the database. to access the database for the history of whether the voucher card is used or not by users.

1. **Use case name:** report

**Actor:**  user

**Description:** the user can report when the seller sells a used voucher card.

1. **Use case name:** print

**Actor:** seller

**Description:** when the seller prints a voucher card.by giving it a serial number

1. **Use case name:** buy package

**Actor:** user

**Description:** when the user buys a package.

1. **Use case name:** credit service

**Actor:** user

**Description:** when the user gets credit service.

1. **Use case:** CRBT

**Actor:** user

**Description:** when the user ordered music for waiting in the call.

1. **Use case:** divert call

**Actor:** user

**Description:** when the user diverts the call from one cellphone to another.

1. **Use case:** customer service

**Actor:** user

**Description:** calling 994

1. **Use case:** recharge

**Actor:** user

**Description:** when the user recharges airtime voucher by our application

1. **Use case:** generate voucher code

**Actor:** seller, database

**Description:** The seller will buy the airtime from us and print it into a voucher card by giving it SN. The SN can be generated by the database and the seller will print the SN in sequence.

1. **Use case:** view category

**Actor:** user

**Description:** list the category of the existing system

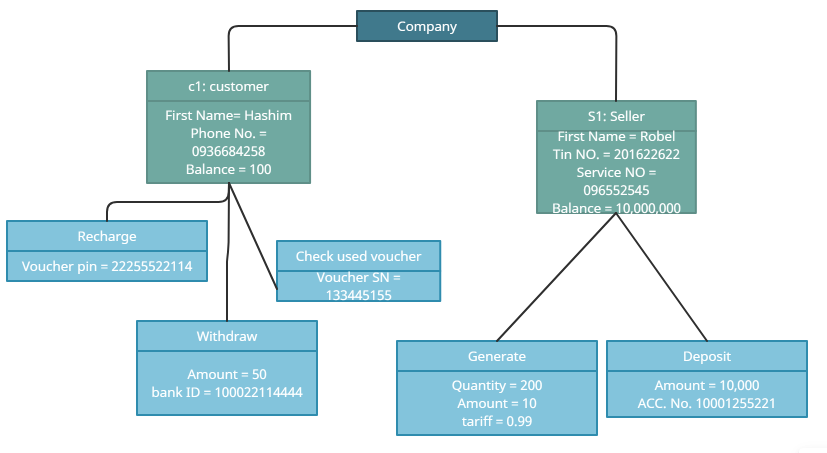
* Buying package
* Getting credit
* Calling customer service
* CRBT service
* Divert call
* Call me back request

1. **Use case:** call me back

**Actor:** user

**Description:** when the user doesn't have time to forward the call the app will send a callback request.

#### **Object diagram**

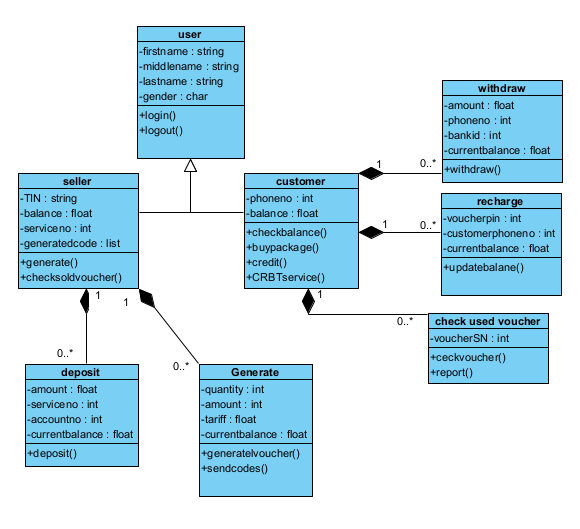


#### **Data dictionary**

| Seller Class | | | | |
| --- | --- | --- | --- | --- |
| Attributes | Data types | Index | Field size | Null value |
| First\_name | String |  | 20 | Not |
| Second\_name | String |  | 20 | Not |
| Last\_name | String |  | 20 | Not |
| Balance | Float | Foreign key | 15 | Not |
| Reg\_date | Date |  | 15 | Not |
| Generated\_code | List |  | 25 | Not |
| Email | String |  | 20 | Not |
| Tin | String |  | 20 | Not |
| Phone\_no | int | Primary key | 10 | Not |
| sex | char |  | 5 | Not |
| Password | String |  | 20 | Not |
| Confirm\_password | String |  | 20 | Not |
| Security\_answer | String |  | 20 | Not |

| Customer Class | | | | |
| --- | --- | --- | --- | --- |
| Attributes | Data types | Index | Field size | Null value |
| First\_name | String |  | 20 | Not |
| Second\_name | String |  | 20 | Not |
| Last\_name | String |  | 20 | Not |
| Balance | Float | Foreign key | 15 | Not |
| Reg\_date | Date |  | 15 | Not |
| Email | String |  | 20 | Not |
| Phone\_no | int | Primary key | 10 | Not |
| sex | char |  | 5 | Not |
| Password | String |  | 20 | Not |
| Confirm\_password | String |  | 20 | Not |
| Security\_answer | String |  | 20 | Not |

#### **Class diagram**

UI

#### **Dynamic models**

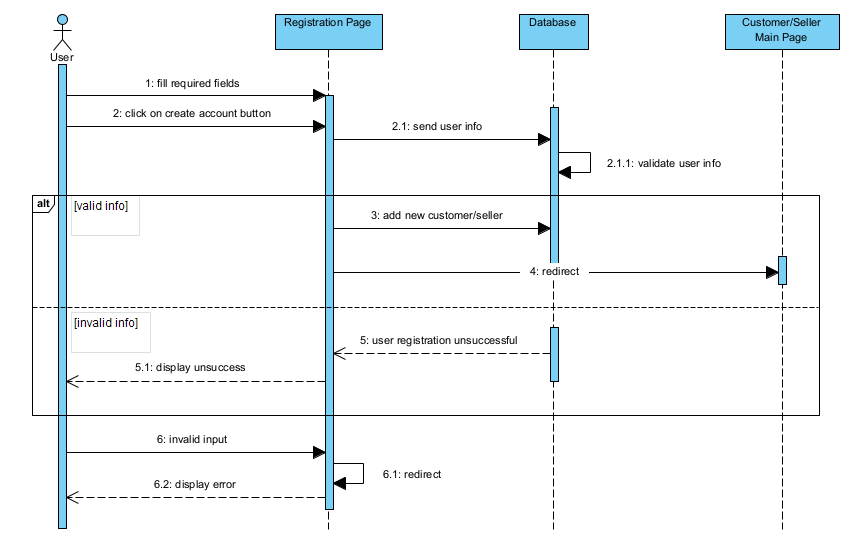
**Sequence Diagrams**

A sequence diagram depicts the dynamic diagram behavior of an entity via a temporal focus on interaction and how the elements that collaborate to constitute an interaction over time provide the functionality of the entity using classifier roles and massage or using instance and stimuli.

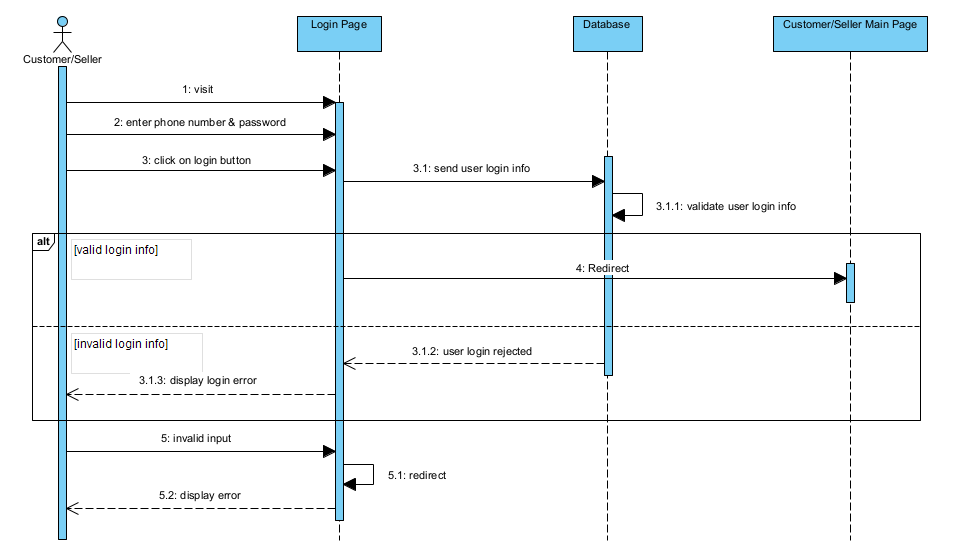
Sequence diagrams are also known as interaction diagrams. A sequence diagram is used to duplicate one or more interactions within a single collaboration. In this part of the documentation, we try to show the integration of all systems. Like:

* registering
* login
* report
* voucher checking (customer)
* voucher checking (seller)

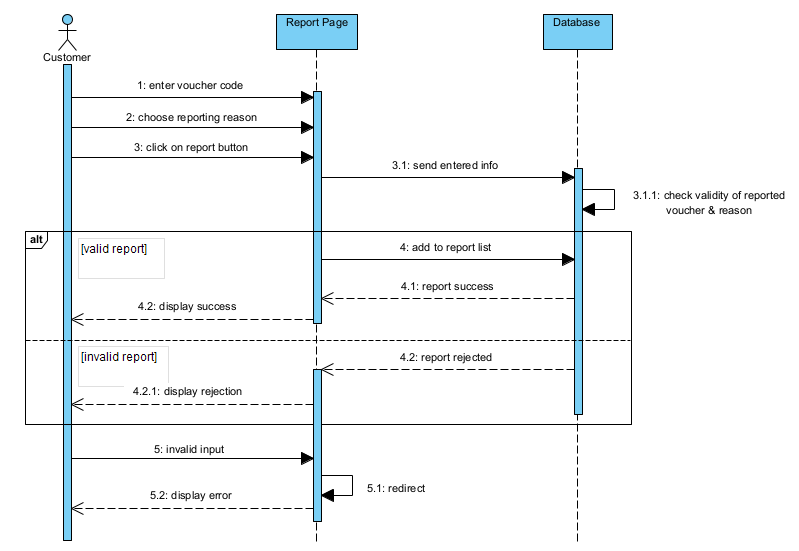
1. **Sequence Diagram for Registering**



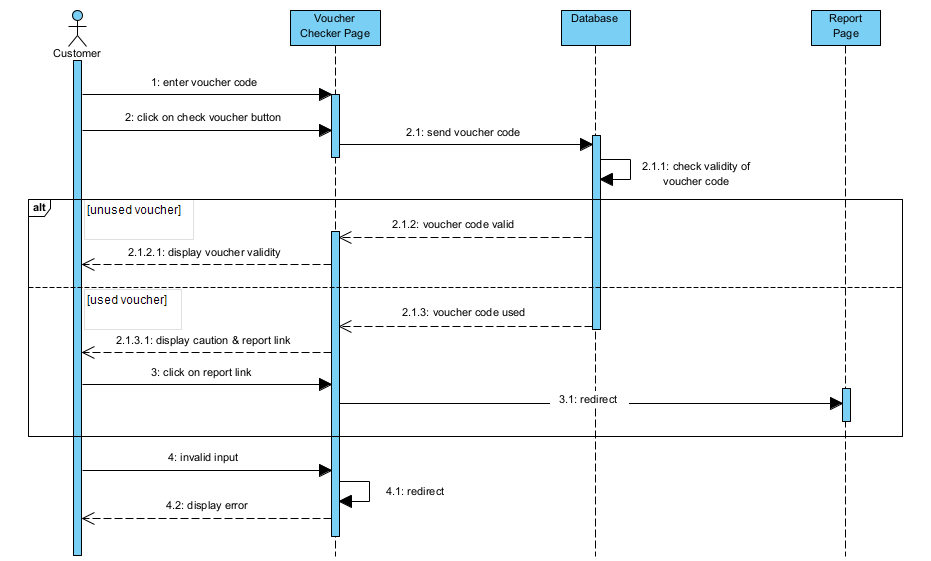
1. **Sequence Diagram for Login**

****

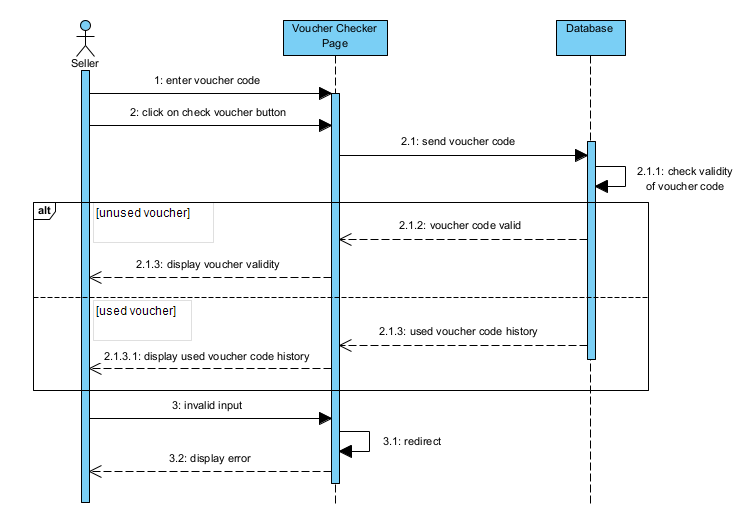
1. **Sequence Diagram for Report**

****

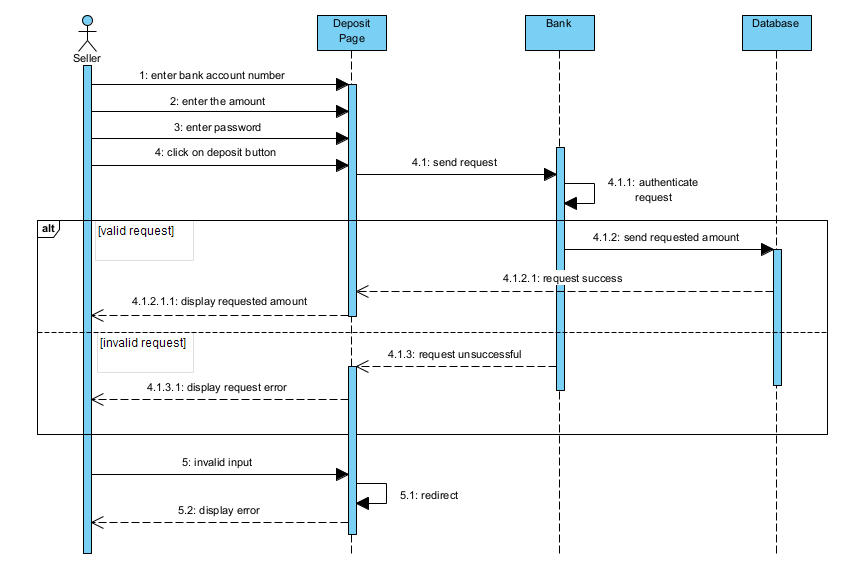
1. **Sequence Diagram for Voucher Checking Customer**

****

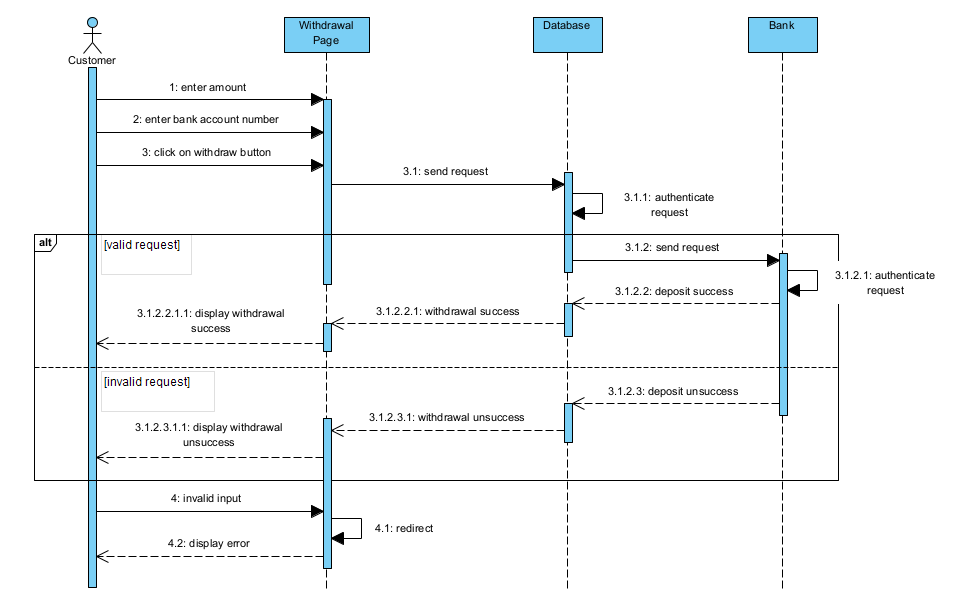
1. **Sequence Diagram for Voucher Checking Seller**

****

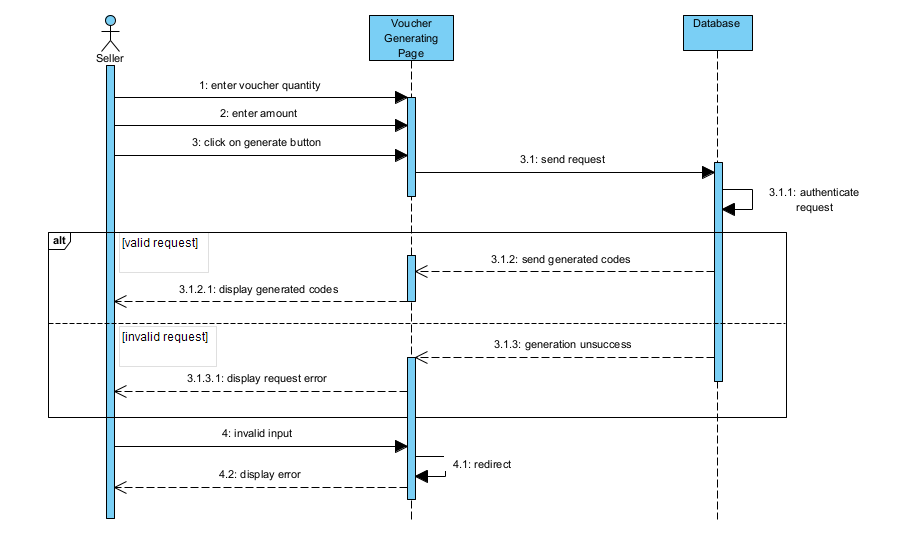
1. **Sequence Diagram for Deposit**

****

1. **Sequence Diagram for Cash-Out**

****

1. **Sequence Diagram for Serial Number Generation**

****

# CHAPTER THREE

# SYSTEM DESIGN

## Introduction

This chapter describes the designs developed for the proposed system whose requirements were collected and analyzed and presented in the previous chapter. It briefly shows how the system will operate and benefit from the proposed system from the manual systems.

Our proposed system is an additional feature that can solve the problems of the currently existing system which is used and controlled by the service provider Ethio-telecom. This project will be used by registered sellers and customers for voucher validity checking, buying airtime, depositing, withdrawing, reporting, and feedback plus the existing system functionalities like recharging, buying packages, sending and receiving (packages, airtime), credit services, etc.

### **Purpose**

The general purpose of this project is to check the validity of the vouchers that people buy from sellers and to eliminate the act of fraud sellers and also fake complaints from the customers. In other words, our project mainly helps to ensure the integrity of the service between both sides (the seller and the customer). In addition to the integrity of our service, this project will enable the sellers to directly buy airtime from the service provider without any distributor involvement and this system will enable anybody who satisfies the system requirements to be a seller.

### **Scope**

We planned to work on this project after observing several concerns from the people regarding the service provider Ethio-telecom. Then we narrowed down what and where the problem is, and we came up with a solution that benefits both entities. This project satisfies almost all the concerns of the people. Which is checking the validity of voucher codes that customers buy from roadside sellers, shops, and any other retailers. The vouchers are small papers that have visible recharging codes on them that lack security and make them more vulnerable to fraud activities (by the sellers and customers). But this project is intended to solve this problem by creating a manageable environment for the service provider, the seller, and also for the customer.

This project will be released as a mobile application that will be easy for the users (sellers and customers) to check the vouchers. The app will save time and energy much more than the current system. In the app, we were also planning to add some features like making transactions with online games, withdrawing money, and depositing to our system from the user's bank account which needs integration with bank accounts and online games. But due to limited time, we are not able to add the described features above which will now be the project's limitation and also made it difficult to widen the scope of our project. For the current release, customers can use the app for:

* voucher validity checking,
* recharging,
* reporting and feedback,
* cash-out (withdrawal)
* buying packages, etc. (current Ethio-telecom services)

## Goals and Trade-offs

* **Rapid prototyping vs. Completeness of Functionality**

Our priority focuses on the completeness of the functionality of the system following the rapid development. The design clarifies the total functionality of the system and how they are implemented for Android and iOS devices. Even though there are some functionalities left unadded because of the shortage of time, we demonstrated all the current functionalities our app provides.

* **Usability vs. functionality**

The way the app’s user interface is developed is simple and user-friendly, in which each functionality is divided and classified in a clearer way to be used by anyone with a basic knowledge of using smartphones.

* **Efficiency vs. portability**

The app will run on almost all Android and iOS devices. Also, it can be sent from one phone to another with good use of a processor and memory.

* **Cost vs. Reliability**

The app will be free to download and also to create an account. In addition, the app will meet the needs and expectations of everyone it serves, by being accessible when needed and by responding within the time frame needed.

* **Reusability vs. Cost**

To make the development cost affordable, we reused the classes and functions in the code during the implementation process. This saves development time and also decreases the development cost.

## System Decomposition

The voucher card checking mobile application is developed to enhance the existing manual platform, have user registration, user login, reporting used voucher cards, can cash-out (withdraw) airtime to real money, and checking system for the seller, and do the existing system functionality like (recharging, sending airtime to the other person, checking how much the user airtime have, etc.) This will make the system fulfill the functionalities of the designed system.

* User Login subsystem: This subsystem helps users to authenticate themselves and distinguish what to access and what to display on their home page.
* **User Registration subsystem:** This system allows users to register and create an account
* **Cash-out:** this system will convert the airtime to real money and cash it out (e.g.: 10-birr air time can convert to 9.5 birrs)
* **Report:** To report the invalid card by showing the time by which the air time is used. And report the invalid one who used its history on the database and report the fraud seller.
* **Checking the voucher validity:** checking the validity of the voucher card and whether it is used or not.

### **Layers and Partitions**

* **User Registration**

The user registration needs storage to store information about users (whether who is the client or seller) and the system should check the validation of the account created from the database.

* **Login**

The system authenticates the user; if the user failed to enter the right information, the system replies with a custom error message for the user. This sub-system depends on the User Registration sub-system.

* **Checking Voucher Code Validity**

This subsystem allows the customer to check only the validity of the voucher card and allows the seller to be provided all information about the voucher codes from the database.

* **Cash Out (Withdraw)**

This subsystem allows the customer to withdraw airtime to real money by transferring it to the customer’s bank account.

* **Generate Voucher Code**

This subsystem allows the seller to generate voucher codes and print them through a POS machine.

* **Report**

This subsystem allows the customers to report fraud activity made by the sellers.

## Concurrency Identification

The system provides access to multiple users. Users on mobile will be able to access the system with an app that we develop to access the system. The database servers are multi-threaded and, in this way, can handle demands from several users at a time. Consequently, the network should be strong enough to reach such an expectation.

A single query consists of two or more queries since a job is composed of several entities. Such as recharge, getting credit buying package, getting credit, calling customer service, CRBT service, divert call, call me back request, sending airtime, checking validation of the voucher card, report the fraud seller, and seller are stored in different sections. The execution of queries will not be handled in parallel by different subsystems. The rationale for this kind of design is that we do not have the necessary hardware or the traffic load to necessitate this. Since Ethio-telecom will not allow us to use its server and it would also take a long time to develop this kind of system.

## Hardware/Software Allocation

The app we are developing will run on android 5.0 above, and Apple iOS 10 above smartphones. We have chosen these hardware platforms because of their availability and must-have devices in the community. So, everyone can get resources from our app. We have implemented our app on the google-backend application development software called firebase which is a fast and reliable database system

### **System Performance**

#### **General system performance**

The system is required to respond to requests less than 5-10 seconds not executed within those seconds. the system will terminate, for tasks that may take longer it is desired to respond in 25 seconds else the system will terminate (stop) the process and ask the user to start again**.**

#### **Input/Output Performance**

There is no need for extra hardware to handle the data generation rate of the system. The existing communication bandwidth is sufficient to support all the communication between subsystems.

It will have the same response rate as any notification.

#### **Processor allocation**

It's a normal utility app. It doesn't need that much processing power; it can run on any simple smartphone but the app can perform better than the rest of the applications that the deployment system’s process can handle normally.

#### **Memory allocation**

Yes, when the system is deployed the system has to buffer all aids of requests.

### **Connectivity**

The app is connected to a real-time database called Firebase. So, the only connection the app will have through the internet connection is to the online database of Firebase. This means there is no physical connection to our system.

### **Network architecture**

There is no unique transmission canal; the user uses a public canal. The general interaction mechanism and protocol for the channels and the communication are synchronous communication so that the server should be available for the user, otherwise it will be disconnected. The expected bandwidth for the networking is 20MB/S.

## Data Management

Data generated by the application will be managed by a non-relational database management system (NoSQL) which stores data in documents rather than relational tables. Firebase is a real-time database that is cloud-based those syncs and stores data between users. The database is accessed every time a user uses the app with an internet connection. The data storage is centralized. The database can only handle 200,000 simultaneous connections which can be considered the worst case since a massive number of users will be using our app.

## Global Resource Handling

The app provides authentication. The user session is handled by writing the authentication that the system will use, which is a basic authentication scheme that sends a Base64-encoded string that contains a user name and password for the client. Base64 is not a form of encryption and should be considered the same as sending the username and password in clear text. The user interface for authentication is for a user to input his/her username and password, and there are no hardware mechanisms to control access. The system will run on a network whose nodes are assigned an IP address that is leased yearly based on a period. Authentication endured that only authenticated and authorized users access the space.

## Software Control Implementation

MVC (model view controller) the model is always responsible for the data handling and the view is responsible for displaying the data. And the controller is responsible for controlling the application logic and acts as the coordinator between the View and the Model. So, there are procedures that require input, one example deposit. Since the deposit is connected to the network takes some time, therefore, the cashing out method needs time for the system until the exchange airtime for real money. after showing the exchange rate it will proceed.

### **External control flow (between subsystems)**

### **Concurrent control**

Some subsystems can run concurrently such as:

* Getting credit
* Buying package
* Calling customer service
* CRBT service
* Divert call
* Call me back request

Until one user requests to view information, the user can view other accessible list pages. Tasks that need fetching or submitting to the request are done asynchronously, other tasks can be done in a parallel direction.

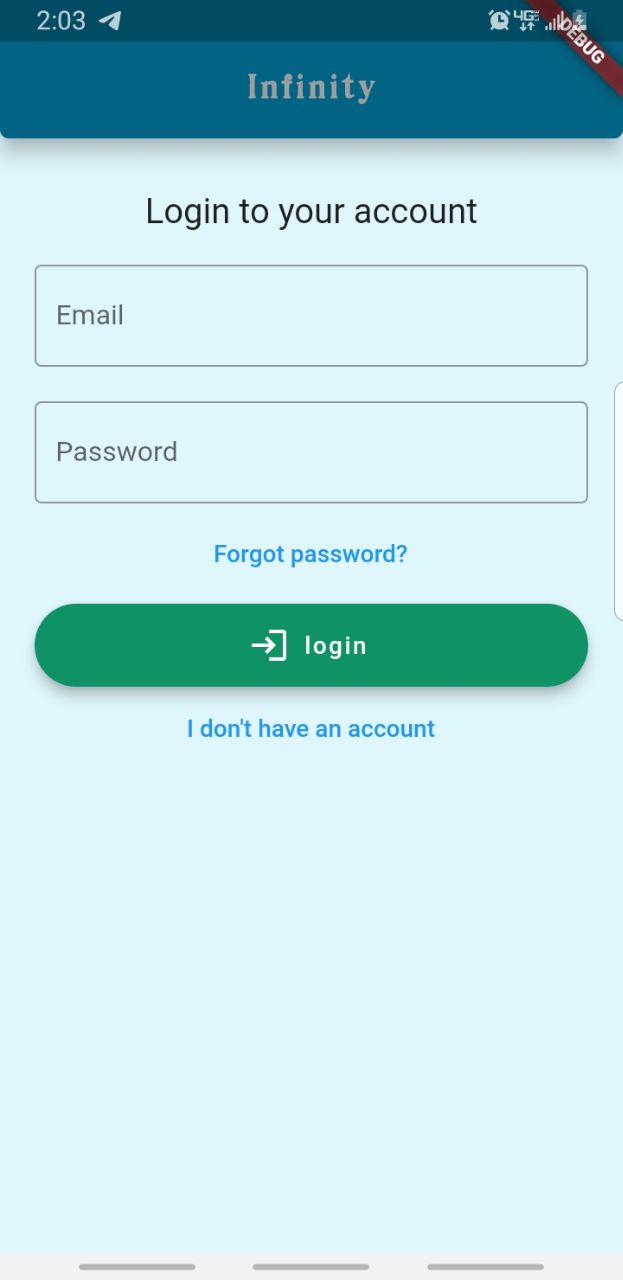
### **Internal control (within a single process)**

The process control system is implemented by a system called the underlying script execution engine. Whenever the user clicks one button from each screen the system displays the destination page for the user.

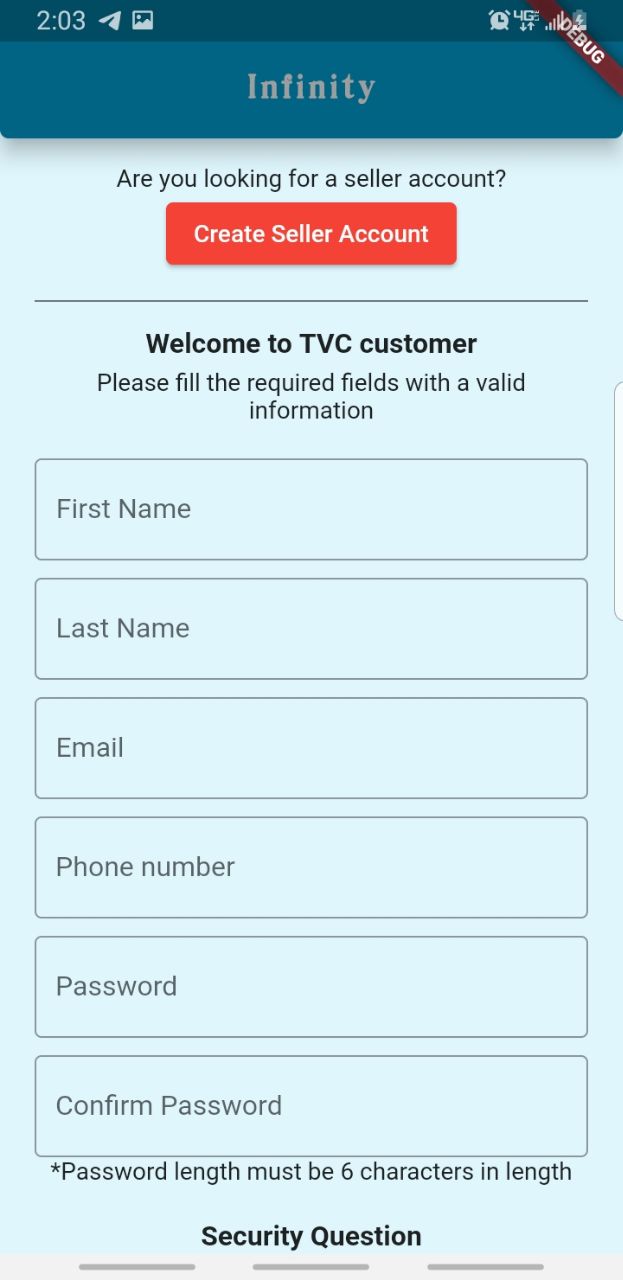
### **User Interface**

These are some of the user interfaces. They have their loop.

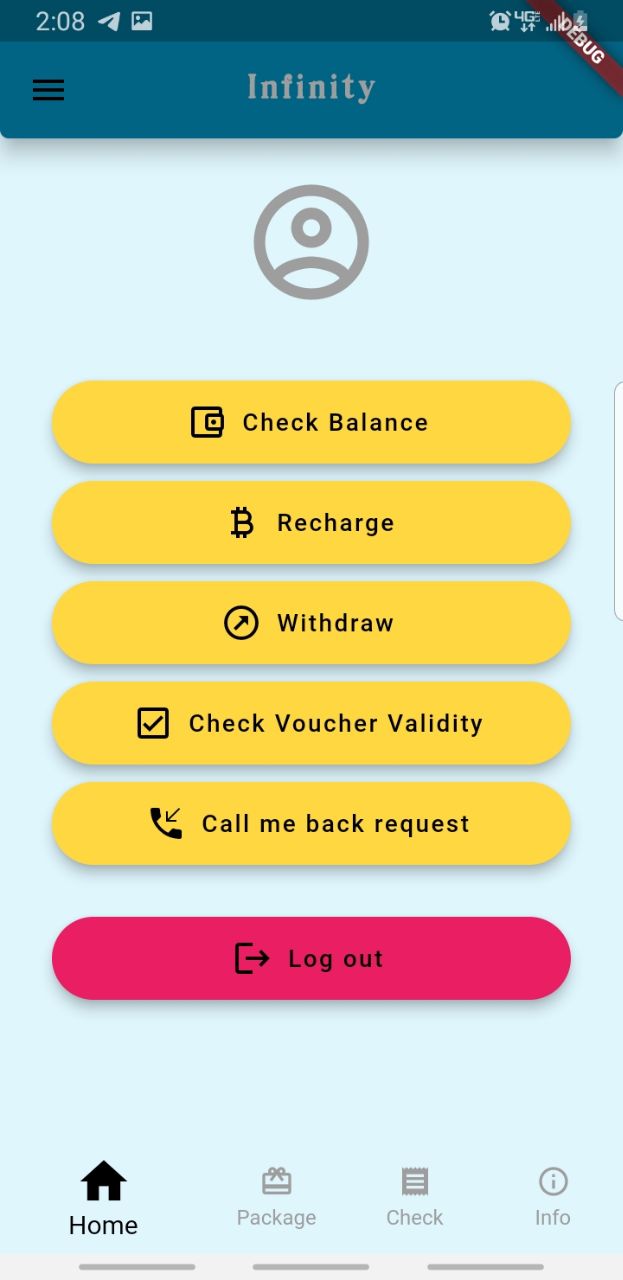
1. **Login UI**

****

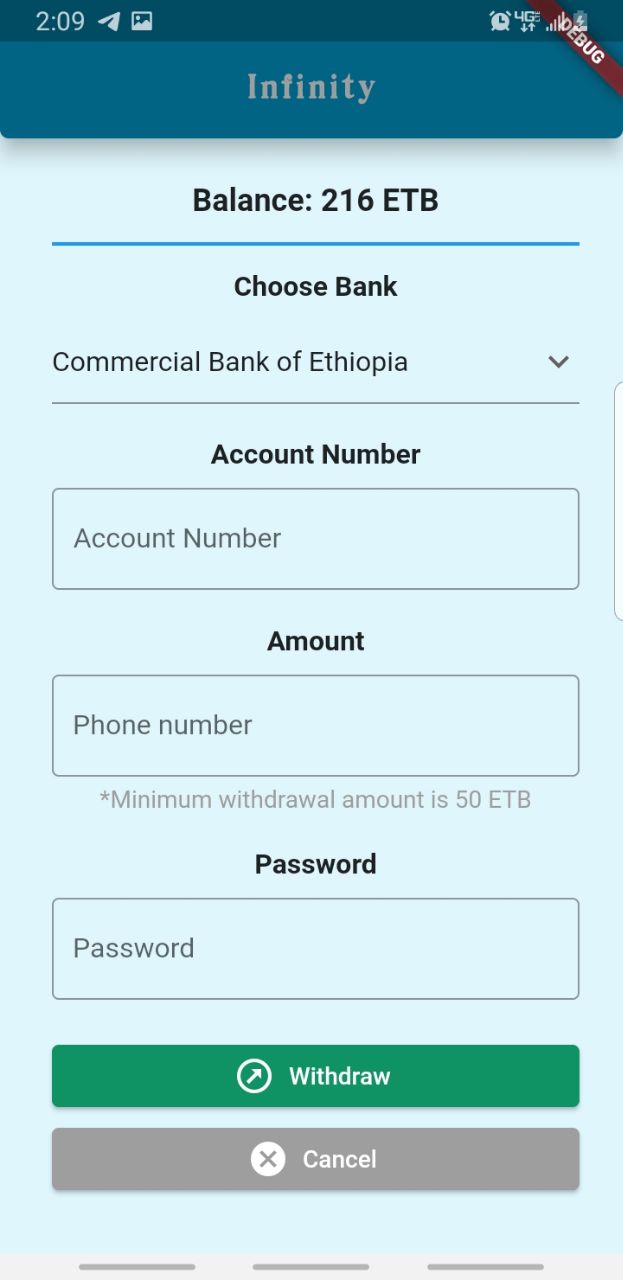
1. **Customer Registration UI**

****

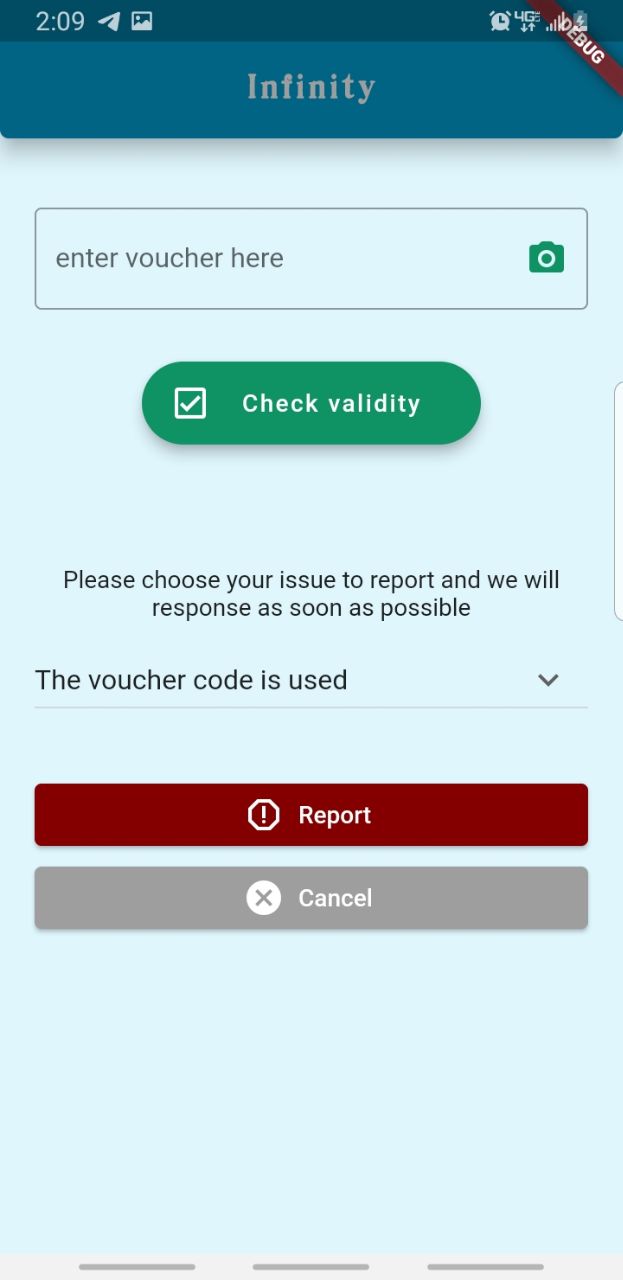
1. **Customer UI**



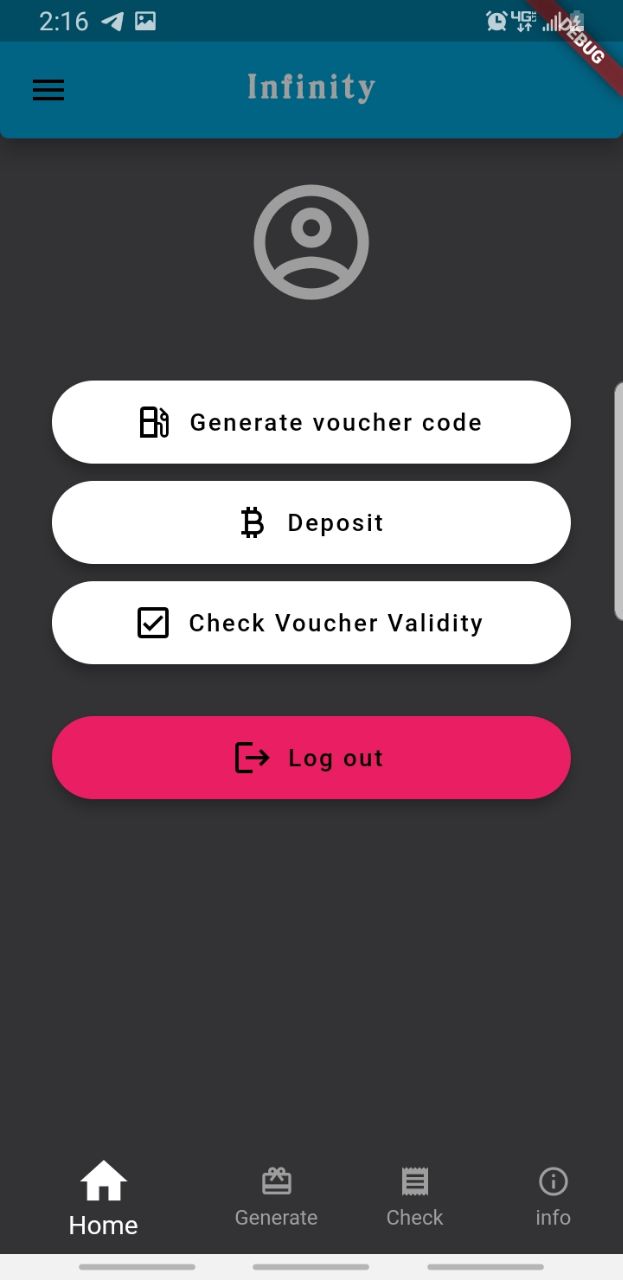
1. **Customer Cash Out (Withdraw) UI**

****

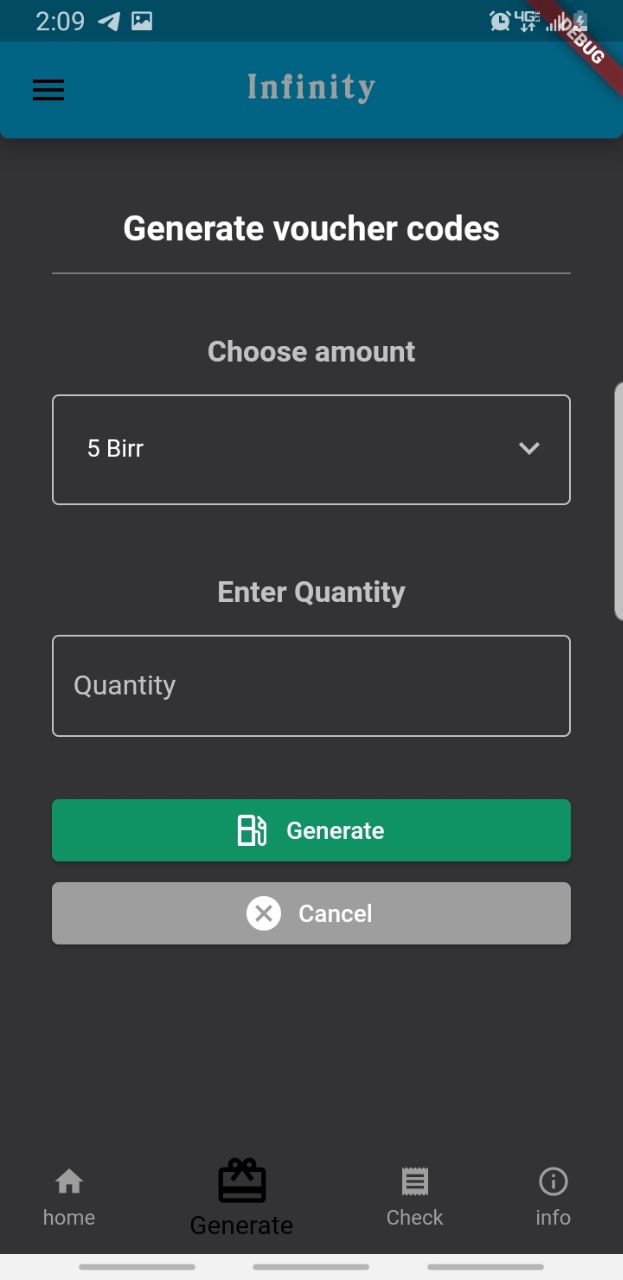
1. **Customer Voucher Validity Check UI**

****

1. **Seller UI**

****

1. **Seller Voucher Code Generation UI**

****

## Boundary Conditions

### **Initialization**

The system can be viewed from a mobile phone. It’s an app-based system. So, the user has to register then the user can log in, after the first login, the data to be user token data should be accessed at the start-up time. and ask the password of every user live. First, when the user opens the app, it will ask for a username and password and then show the first age.

### **Termination**

The termination conditions for the system are as soon as the users sign out or destroy the lifecycle of the application. At the time of termination, the seller database connections will be closed. Database commits will be done by the subsystems when the system terminates the user will terminate any kind of transaction that occurs if they log out.

### **Failure**

Any incomplete transaction will be rolled back to the point where it was before.

## Design Rationale

The reason we choose flutter over the other design architecture patterns are:

* **One code base**

Cross-platform development is a trending development style with one code base, and with Flutter, it becomes cooler in our development we get one code base for Android and IOS apps. from the nature of our app, we want to address smartphone users as of biggest smartphones are IOS and android in the market.

* **Fast development**

With flutter we can develop very faster than ever and with its hot reload feature it becomes extremely fast. In addition, flutter is everything widget and that diminishes the UI development struggle to just a simple win.

* **Testability**

Two types of testing will be performed on the mobile app. These tests were mainly performance testing and functionality testing.

* **Performance Testing**

The mobile app is tested first if it works with different mobile devices. After that check, if the database supports multitasking by 5 of my Tiamat accesses at the same time.

* **Functionality Testing**

All functions in the application, database connection, forms used to enter data for giving airtime, generating airtime voucher, getting information from database whether the voucher is valid or not (checking voucher) can cash out the air time to real money, can access developers performed the test of some functionality requirements were tested during the test of security.

* **Extensibility**

We will have a better chance of making any of those parts more reusable. It has also the ability to replace or add new pieces of code that do similar things in the right places in the architecture.

# CHAPTER FOUR

# IMPLEMENTATION

## Overview

In order to develop the project which is a mobile app, we used a real-time NoSQL cloud database for data storage called Firebase. Also, data is synced across all clients in real-time, and remains available when the app goes offline. It also has a better security & speed. So, the backend is developed using Firebase and Dart, plus the front-end is developed using Flutter.

## Tools and Technology Utilized for Implementation

* Visual Studio
* Android Studio
* Web Browser (Chrome).

## System Implementation

### **System Interfaces**

**4.3.2 System Algorithm**

Here in our project, we use an algorithm, which will check each user role and assign them to their default page.

### **Source Code**

* **Sign Up**

Future signUp(String Email, String password, String Username,int phoneNo,double Balance) async {  
 try {  
 await \_auth.createUserWithEmailAndPassword(  
 email: Email, password: password);  
 User user = \_auth.currentUser!;  
 String uid = user.uid.toString();  
 await FirebaseFirestore.*instance*.collection('Customers').doc(uid).set({  
 'Full Name': Username,  
 'Phone No': phoneNo,  
 'Balance': Balance,  
 }).then((\_){  
 print("collection created");  
 }).catchError((\_){  
 print("an error occured");  
 });  
 print (uid);  
 return user;  
 } on FirebaseAuthException catch (e) {  
 print(e.code);  
 setState(() {  
 loading = false;  
 });  
 return showMyDialog(context);  
 }  
}

* **Sign In**

try {  
 await \_auth.signInWithEmailAndPassword(  
 email: Email, password: password);  
 User user = \_auth.currentUser!;  
 String uid = user.uid.toString();  
 print (uid);  
 return user;  
} on FirebaseAuthException catch (e) {  
 print(e.code);

* **Data retrieve**

class DataRetrieve extends StatelessWidget {  
 final String documentId;  
  
 DataRetrieve(this.documentId);  
  
 @override  
 Widget build(BuildContext context) {  
 final CollectionReference users = FirebaseFirestore.*instance*.collection('Customers');  
  
 return FutureBuilder<DocumentSnapshot>(  
 future: users.doc(documentId).get(),  
 builder:  
 (BuildContext context, AsyncSnapshot<DocumentSnapshot> snapshot) {  
  
 if (snapshot.hasError) {  
 return Text("Something went wrong");  
 }  
  
 if (snapshot.hasData && !snapshot.data!.exists) {  
 return Text("Document does not exist");  
 }  
  
 if (snapshot.connectionState == ConnectionState.done) {  
 Map<String, dynamic> data = snapshot.data!.data() as Map<String, dynamic>;  
 return Text('''Full Name: ${data['Full Name']}   
Balance: ${data['Balance']}''');  
 }  
  
 return Text("loading");  
 },  
 );  
 }  
}

* **USSD Codes**

**\_**callNumber() async{  
 const number = '<USSD Code>'; //set the number here  
 bool? res = await FlutterPhoneDirectCaller.*callNumber*(number);  
}

* **Generator**

import 'dart:math';

import 'dart:core';

class Generated{ //modal class for Person object

  String voucher="", serial="";

  Generated({required this.voucher, required this.serial,});

}

main(){

  var x=3;

  List<Generated> generatedd = [];

  var genserial = "10020000000";

  var gener = 10020000000;

  var number = "";

  var randomnumber=  Random();

  //change i < 15 on your digits need

  for (var i = 0; i < x; i++) {

    for (var i = 0; i < 14; i++) {

      number = number + randomnumber.nextInt(9).toString();

      gener= gener+1;

      genserial = gener.toString();

  }

  print(number);

  generatedd.add(Generated(voucher:number, serial:genserial));

  number = "";

}

  print (generatedd);

}

# CHAPTER FIVE

# 5. Conclusion and Future Work

## 5.1 Conclusion

This project mainly focuses on the problems and concerns among people who use airtime vouchers for recharge, in which the vouchers code is printed visibly in bold numbers. And our project provides a way to check the validity of those vulnerable airtime voucher codes and also it will enable the customers to report fraud activity. Our project is a multiplatform mobile app that supports the current Ethio-telecom services with a simple and user-friendly UI using Flutter. It can be used by Android and iOS devices and uses a real-time NoSQL database called Firebase, which can handle 200,000 simultaneous user requests at a time.

## 5.2 Future Work

For future work, we planned to integrate our system with outside systems like banks and online games for withdrawal and making payments with the recharged airtime.

# GLOSSARY

| **Terms** | **Definitions** |
| --- | --- |
| **Actor** | represents roles which may include human users, external hardware or other systems. |
| **Class diagram** | a type of static structure diagram that describes the structure of a system by showing the system’s classes, attributes, operations, and the relationship among objects. |
| **CRBT** | Caller Ring Back Tone |
| **Object diagram** | a diagram that represents a specific instance of a class diagram at a certain moment in time. |
| **POS machine** | a hardware system for processing card payments at retail locations. |
| **Scenario** | a formal description of the flow of events that occur during the execution of a use case instance. |
| **Sequence diagram** | s diagram that illustrates the sequence of messages between objects in an interaction. |
| **SMS** | Short Message Service |
| **UML** | Unified Modeling Language |
| **UML diagram** | a diagram based on the UML with the purpose of visually representing a system. |
| **Use case diagram** | a graphical depiction of user’s possible interactions with a system. |
| **UI** | User Interface |
| **NoSQL** | a type database which provides a mechanism for storage and retrieval of data that modeled in means other tabular relations used in relational databases. |
| **Flutter** | an open-source framework by Google for building natively compiled, multi-platform applications from a single codebase. |
| **Dart** | a client-optimized language for fast apps on any platform. |
| **Firebase** | a Google-backed application development software that enables developers to develop iOS, Android and Web apps. |
| **System** | a set of inter-connected computer equipment and/or programs used together for a particular purpose and operating together. |
| **ETC** | Ethiopian Telecommunication Corporation |
| **Database** | an organized collection of data stored and accessed electronically. |
| **Real-time Database** | a database system which uses real-time processing to handle workloads whose state is constantly changing. |
| **Voucher** | a small printed piece of paper or a receipt that may be exchanged for goods or services. |
| **Authentication** | the process or action of verifying the identity of a user or process. |
| **Source Code** | a text listing of commands to be compiled or assembled into an executable computer program. |

# REFERENCES

* Wikipedia.
* Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)
* The Unified Modeling Language Reference Manual, (paperback) (2nd Edition)
* UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition)
* Flutter Complete Reference: Create Beautiful, Fast and Native Apps for any Device
* Beginning Flutter: A Hands On Guide to App Development

REFERENCES

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Declaration** | |  | | We, the undersigned, declare that this project is our original work and has not been presented for a degree in any other university or college, and that all sources of materials used for the project have been acknowledged. | |  | |  | | **Declared By:**   |  |  |  |  | | --- | --- | --- | --- | | **No.** | **Student Name** | **Signature** | **Date** | | 1. | <Student 1> | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_ | | 2. | <Student 2> | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_ | | 3. | <Student 3> | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_ | | 4. | <Student 4> | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_ | | 5. | <Student 5> | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_ | | |  | |  | | **Confirmed By Advisor:** | | Name: <Advisor Name> . . | | Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |  | | **Place and date of submission:** | | MicroLink IT College, <Month Year> | |  | |  | |  | |