Addis Ababa University

አዲስ፡አበባ፡ዩኒቨርሲቲ

SEEK WISDOM, ELEVATE YOUR INTELLECT AND SERVE HUMANITY !

**ADDIS ABABA UNIVERSITY COLLEGE OF NATURAL SCIENCE**

**DEPARTMENT OF INFORMATION SYSTEM**

## Industrial project 1

# WEB BASED COMMODITY EXCHANGE SYSTEM

# Group members

| | | |
|---|---|---|
| 1 | Kirubel Asnakew | ugr/3194/12 |
| 2 | Abdulmenan mohammed | ugr/2627/12 |
| 3 | Anteneh Amare | ugr/1571/12 |
| 4 | Biruk Amare | ugr/6400/12 |
| 5 | Haileamlak waleligne | ugr/1406/12 |
| 6 | Samuel Tesfaye | ugr/4893/12 |

# Acknowledgement

Primarily, we would like to thank God for giving us the necessary guidance, energy, support and drive to complete the first semester process of the project.

We would like to express greatest gratitude to our advisor Instructor Melaku Girma to his extensive advice for the project completion. We are also very grateful and would like to extend our sincere thanks to our team members, our friends and students for sharing their ideas, suggestions, support and especially for their commitment. We really give a great respect and credit to everyone who involved in our project tasks.

We would also want to thank the industrial committee for periodically announcing the submission date of each phase of the project, which helped us to strictly adhere to the time limit for supplying deliverables.

# Acronyms

HTML ………………………………. Hyper Text Markup Language

CSS …………………………………………Cascading Style Sheets

AJAX ……………………………………. Asynchronous JavaScript

XML …………………………………. Extensible Markup Language

PHP……………………………………... Hypertext Pre-processor

XAMPP………………. Cross Platform Apache MariaDB PHP Perl

UML……………………………………Unified Modeling Language

WBS……………………………………. Work breakdown structure

CX …………………………………………Commodity Exchange

ECX ………………………………Ethiopian Commodity Exchange

UI----------------------------------------------------User interface

**Table of Contents**

# List of Tables

# List of Figures

# 1. Introduction

## 1.1 Overview

Ethiopia has a large domestic market with a total population of about 110 million people (2020). After an intense period of the COVID-19 pandemic and a prolonged and devastating civil conflict, the Ethiopian macro economy is finding itself on a difficult war-time footing, facing immense humanitarian and security costs, as well as a heavy toll in terms of lost social and physical infrastructure.

Over the 15 years until 2019, Ethiopia's economy had been amongst the fastest growing in the world at an average of 9.5 % per year. Among other factors, growth was led by capital accumulation, in particular, through public infrastructure investments. Ethiopia's real gross domestic product (GDP) growth slowed down to 6.3% in fiscal year (FY) 2020/21 compared to the previous year's historic growth due to COVID-19, with growth in industry and services easing to single digits. However, agriculture was not significantly affected by the pandemic according to the World Bank because around 80.0-85.0% of Ethiopians are engaged in agriculture. It is true that agriculture is the backbone of our country's economy, but even though we do not import agricultural products from abroad, the cost of these products has increased like they are imported.[1]

Ethiopia has been characterized by high costs and high risks of transacting, forcing much of Ethiopia into global isolation. With only one-third of output reaching the market, commodity buyers and sellers tended to trade only with those they knew, to avoid the risk of being cheated or default.

Trade is done based on visual inspection because there was no assurance of product quality or quantity, this droves up market costs, leading to high consumer prices. For their part, small-scale farmers, who produce 95 percent of Ethiopia's output, came to market with little information and are at the mercy of merchants in the nearest and only market they know, unable to negotiate better prices or reduce their market risk.[2]

The Ethiopia Commodity Exchange, (ECX), is a marketplace, where buyers and sellers come together to trade, assured of quality, delivery and payment. The vision of ECX is to transform the Ethiopian economy by becoming a global commodity market of choice. ECX's mission is to connect all buyers and sellers in an efficient, reliable, and transparent market by harnessing innovation and technology, and based on continuous learning, fairness, and commitment to excellence. But, ECX only provides these services for exporters and the commodities provided by ECX are limited in type. There is no web-based or mobile-based Commodity Exchange System for the local Market that provides a modern, efficient, transparent, and reliable market platform and warehousing service, our web-based Commodity Exchange System will be a marketplace, where buyers and farmers (sellers) come together to trade, assured of quality, delivery, and payment, the local government policy provides support to do such activity. We need to provide a modern, efficient, transparent, and reliable market platform and warehousing service through the adaptation of technology, excellence in innovation, and with integrity.[1]

We try to make the Market efficient by operating a trading system where buyers and farmers (sellers) can coordinate seamlessly based on standardized contracts and make the market transparent by disseminating market information in real time to all market players.

## 1.2 Statement of the problem

Below we have classified the problems which the current market is facing into different sections.

- Performance
    - ✓ The current market situation takes time and energy to get information about the marketplace, specific product, amount and quality.
    - ✓ There is product wastage because of the loss of a comfortable warehouse and appropriate marketplace to sell products.
    - ✓ Cultural operation
- Information
    - ✓ Farmers do not have any information about where to sell their products
    - ✓ Buyers also have no information about where to buy the commodity they need
    - ✓ Both the farmer and the buyer have no information about the current price of the goods.
- Economy

    Cost of the products are much more expensive because of
    - ✓ Free market economy
    - ✓ Brokers
- Control/Security
    - ✓ Loss of product from the marketplace
    - ✓ Products are kept hoarding
    - ✓ Can't keep track of the products
- Efficiency
    - ✓ In times of need, finding a product can be challenging
- Service
    - ✓ It is difficult to get a product of preferred quality.
    - ✓ Getting the whole transaction done takes a lot of time.
    - ✓ Backward market operation

# 1.3 Objective of the Project

## 1.3.1 General Objective

The main objective of doing this project is to develop a web-based commodity exchange system that coordinates better, links faster, and protects the interests of both sellers and buyers.

## 1.3.2 Specific Objective

1. Analyze the current market system
2. Specify functional and non-functional requirement
3. Design the new system based on the analysis of the current system
4. Develop the new system based on the design
5. Test and deploy the new system

# 1.4 Feasibility study

A feasibility study is simply an assessment of the practicality of a proposed project plan or method. This is done by analyzing technical, economic, legal, operational and time feasibility factors.

## 1.4.1 Operational feasibility

Operational feasibility is the ability to utilize, support and perform the necessary tasks of a system or program. It includes everyone who creates, operates or uses the system. To be operationally feasible, the system must fulfill a need required by the business. We believe that our system is operationally feasible because:

- The government is going to manage the system.
- We are going to have policy support
- Develop it per user requirements or needs.
- Design it with capability to provide the end users with timely, accurate and reliable information.
- Develop it with ability to provide adequate through put and response time.

## 1.4.2 Economic Feasibility

Economic feasibility is also referred to as cost/benefit analysis. While developing this project we are going to use free software and use our own computers, in addition to this we are going to build our web-based application by ourselves along with the help of our advisor, so our project is economically feasible.

**Cost:** refers to tangible and intangible cost that is incurred to develop the system.

**Tangible cost:** Is the cost that is directly measured in birr.

Estimated cost for the project is listed in the table below.

| **Number** | **Items** | **Description** | **Expected cost** | **Cost type** |
|---|---|---|---|---|
| **1** | Advertisement | The cost we are going to invest for advertisement purpose. | 100000 | recurring |
| **2** | Development | The cost for development of the proposed system. | 50000 | One time |
| 3 | Expert | We will get help from experts on different problems like deployment and test. | 20000 | |
| **3** | Transport | We are planning to go to few product buyers and farmers (who sell products) to do interview to gather the requirement of the system. | 200 | recurring |
| **4** | Deployment | We are going to deploy the proposed system on yegarahost.com (hosting service provider). | 3000 | recurring |
| **5** | computer | We need computers for admins to manage different things. | 160000 | One time |
| 6 | Training | Proper training should be given specially for farmers since they are not familiar with use of computers and systems. | 30000 | One time |
| 7 | Total | | 363000 | |

Table 1.1 cost feasibility

**Intangible Cost:** is the cost that is not directly measured in birr.

- Loss of time and energy

## Benefits

Benefits are paybacks from a business venture. This system will have tangible and intangible benefits.

## Tangible benefits

Tangible benefit is an asset that produces income consistent with its fair market value, which is listed by the wed developer or organization in a quantifiable form. A benefit that we get from using this web-based application can easily measure in the following manner.

- Cost reduction and avoidance.

### Intangible benefits

Intangible benefit also called soft benefits, are the gains attributable of the improvement project that are not reportable formal accounting purposes. In this case, we can see the benefits of using this web-based application that is not easy to measure in terms of money.

- Reduce the response time of activities.
- Reduce number of brokers

## 1.4.3 Technical Feasibility

This assessment focus on the technical resources available to implement the project It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software and other technical requirements of the proposed system.

Our team has learned different requirement gathering techniques, programming languages and development technologies that allow us to convert ideas to working system. We will also get help from experts (web design expert, web development expert, deployment and test expert) to help us finish the project. We believe our system is technically feasible.

Hardware requirement

- ✓ 10GB database space
- ✓ 8GB RAM
- ✓ 3.5GHz CPU

Software requirement

- ✓ Window Operating system.
- ✓ MySQL
- ✓ Wamp server

## 1.4.4 Schedule Feasibility

The development process of this project is dynamic but we can complete all the sections by the time frame allocated in the project schedule in section [1.13].

## 1.5 Significance of the system

- Web-Based Commodity Exchange tries to benefit and modernize the way Ethiopia trades its most valuable assets, its commodities. Ethiopia needed a change from the traditional means of trading to better support the needs of all those involved in trading and production.

- CX creates trust and transparency through aggressive market data dissemination to all market actors, through clearly defined rules of trading, warehousing, payments and delivery, business conduct, and an internal dispute settlement mechanism.

- CX provides market integrity at three important levels:
    - The integrity of the product itself,
    - The integrity of the transaction, and
    - The integrity of the market actors
- Make a bank transaction and people who aren't using banks become bank customers.

- Reduce current market inflation by trying to reduce prices. This is because we believe our system will reduce the number of brokers, who make the market inflation high.
- Accessible: Purchases can be made anywhere.
- Convenient: There is no need to carry cash.
- Reliable: Due to the fact that the payment is made by a bank or tele birr, you can rest assured.

## 1.6 The beneficiary of the project

- Government: Inflation, cash flow, and product flow can all be easily controlled by the government.
- Farmers: will be able to find out where to sell their products without having to deal with any intermediaries
- Customers /consumers can get preferred quality when needed
- Bank: Due to the fact that the transaction is made by the bank, people who do not use the bank become customers of the bank.
- Ethio-telecom: As a result of the Tele Birr, Ethiopian telecom will benefit like banks.
- Delivery service providers.

# 1.7 Methodology

Methodology is defined as a particular procedure or set of procedures used in academic research and scientific experimentation. It refers to the systematic and logical approach used to solve specific problems, typically involving the collection and analysis of data. Methodology can include both qualitative and quantitative methods, such as interviews and surveys, observation, experiments, and fieldwork.

## 1.7.1 Data Collection

Systems analysis is the part of the systems development life cycle in which you determine how a current information system in an organization function. Then you assess what users would like to see in a new system.

The two parts to analysis are determining requirements and structuring. System analysts use different requirement elicitation techniques such as interviews, questionnaires, user observation, workshops, brainstorming, introspection etc.

After tuning it to our particular requirements, we decided to use at least two or all in combination for identifying the requirements of the new system from the above requirements elicitation mechanisms. These are interviewing, introspection, and document analysis.

## Introspection

Introspection is the first and the most obvious method for trying to understand what properties a system should have in order to succeed. In introspection technique, requirement analyst "imagines" what kind of system is required for doing the required job or by using available equipment etc.

### Reasons to use introspection

- Introspection is an easier technique to apply.
- There are no costs for implementing this technique
- It can act as a good initial step to start requirements elicitation

## Document analysis

We will analyze different documents if we think it is necessary and can teach us how we should develop our system. We have analyzed the document of ECX.

### Reasons to use document analysis

- Document analysis is an efficient and effective way of gathering data because documents are manageable and practical resources.
- Obtaining and analyzing documents is often far more cost efficient and time efficient.
- Documents can be read and reviewed multiple times and remain unchanged.

## Interview

Interviewing is one of ways analysts gather information about an information system project.

### Reasons to use Interview

- It provides information to supplement other methods of collecting data.
- It is used to gather data extensively and intensively.
- It is used to exchange ideas and experience.

## 1.7.2 System development methodology

System development methodologies are a standard set of steps used to develop and support information systems in organizations. We need a methodology for analyzing a problem to be solved, planning for the design of the solution and a construction method that minimizes the risk of error.

### Development approach

### Object oriented Approach

The method of system development paradigm that we selected is the object-oriented approach because this approach is helpful for us to represent the different phases of the project through many diagrammatic representations such as activity diagrams, use cases, sequence diagrams and class diagrams. Generally, we choose object-oriented development approach because:

- It has a reusability feature
- These techniques provide greater opportunities for users to participate in the t process
- Increases flexibility
- Improves quality
- It is the latest, powerful, easy and highly usable
- Increase domain and design reuse

**Process Model**

### Iterative process model

The process model we used is iterative process model. we select iterative model because, to develop our project we are required to review and design in each phase iteratively to meet user requirements.

Flexibility: - this model allows as to make changes at any stage without affecting the scope of the project.

Improved Quality; - this model allows as to identify and fix problems early in the development process. this leads to improved quality.

collaborations: - encourages collaboration between our team members, leading to a better overall understanding of the project goals and requirements.

## 1.8 Development tools and technologies

### 1.8.1 Front end technologies

These are technologies to be used to build web pages and user interfaces for web applications.

### HTML

Html (Hypertext Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content.

#### Advantages of using html

It is easy to use and to learn, simple to edit, light weight, user friendly, and free. In addition to these, all browsers support it.

### CSS

CSS (Cascading Style Sheets) is a declarative language that controls how webpages look in the browser.

#### Advantages of using CSS

It saves time, helps to make spontaneous change and consistent change, improve page loading speed, device compatible, and makes search engine better crawl web pages.

### JavaScript

Java script is a programming language used most often for dynamic client-side scripts on webpages and allows you to implement complex things on web pages. It is easy to learn, debug and test, fast no need of compilation, platform independent, and has programming language capabilities.

### Bootstrap

Bootstrap is a free, open-source HTML, CSS, and JavaScript framework for quickly building responsive websites. It saves time, encourages consistency, provides an excellent grid system, creates responsive website.

### jQuery

jQuery is a JavaScript Library that focuses on simplifying DOM (Document Object Model) manipulation. It is open source, makes table sort-able and dynamic, display easily charts, popup overs, modals, tabs, integrated or compatible with bootstrap, highly extensible, cross-browser compatible.

### AJAX

AJAX (Asynchronous JavaScript and XML) allows updating parts of the DOM of a HTML web-page instead of having to reload the entire page. It is the integration of jQuery and PHP. It is easy to learn and use, allows changing content without refresh or reload web-page, makes website faster. Response time is faster so increases performance and speed.

## 1.8.2. Back-end technologies

These technologies used to deal with server, application and database.

### PHP

PHP is a server-side scripting language. That is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Preprocessor, that earlier stood for Personal Home Pages. It is open source. PHP is manly supported by all the operating systems like windows, Unix, and Linux. It can be integrated with other programming language and database easily and there is no requirement of redevelopment. Which means it saves a lot of effort and cost. It is simple and easy to learn and to code. It is one of the fastest languages.

### Laravel

Laravel is an open-source PHP web framework for creating web applications. It is used to develop websites, APIs, and web services quickly and easily. Laravel provides an expressive and elegant syntax for writing efficient, concise code.

## MySQL

MySQL is a free to use, open-source database that facilitates effective management of database by connecting them to the software. Advantages of using MySQL It is more secure and reliable, provides on demand scalability, and open source so reduces cost.

## Apache

Apache is an open-source and free web server software that powers around 46% of websites around the world. The official name is Apache HTTP Server, and it's maintained and developed by the Apache Software Foundation.

### Advantages of using Apache

- It is open source, reliable and stable software.
- It is also easy to configure and has easily available support in case of any problem.

## XAMPP

XAMPP stands for Cross Platform Apache MariaDB (MYSQL), PHP, Perl XAMPP is a free open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

### Advantages of using XAMPP

It is easy to learn and install when compared with other web servers like WAMP. It consists of Apache, and MYSQL, so there is no need of installing them separately. It is also possible to start and stop the web server and database stack with one command.

## 1.8.3 Documentation and modeling tools

At the end of the project, we will deliver system documentation and user documentation. Our system documentation would be the description of detail information about user requirements; systems design specifications, its internal workings and its functionalities. And our user documentation will be in the form of online help with a web connection. It will contain information about the system; how it works and how to use it.

We will use the following tools for our project's documentation:

- Lucid chart to draw UML diagrams and
- Microsoft Excel 2019 to make project schedule using Gantt chart
- Figma as prototyping tool
- Microsoft Word 2019 as word editor.

### 1.8.4 Deployment environment

We use **Yegarahost.com** to store the system and run it on the internet.

## 1.9 Scope

Scope refers to all the work involved in creating the products of the project and the processes used to create them. Defining scope of a project is extremely important to not only make sure no obstacles crop up unexpectedly but also to resolve any potential risks that may arise during its execution.

It is very important to articulate what will be included and what will be excluded in this project. In fact, we will add additional functionality if we found it very necessary to the success of the project. We may also discard functionality if we found it is not really important and affordable.

- The system will only be functional in Ethiopia.
- Products are only agricultural(cereal)
- The system doesn't provide delivery and payment functionality but outsource it.
- Control prices by setting initial prices
- Farmer can add new products
- Involves buyer and seller(farmer) registration

## 1.10 Risks assumptions and constraints

**Risks:**

- The level of awareness users (especially farmers) has to technology may not be good. As a result, people may not adopt the system easily.
- We may not have sufficient time and knowledge to develop all the deliverables of the project.
- Consumers may be reluctant to the reliability of the system.

### Assumptions:

- We assume consumers own and know how to use smart phones and computers.
- We assume consumers will easily understand the advert we make.
- The cost we are going to invest is enough to get the project done.
- We will have improved internet connection or at least it continues like this.
- We assume both the payment and delivery system run smoothly to make the system fully functional.

### Constraint:

- Not all farmers may easily adapt the system.

## 1.11 Phases and Deliverables of the project

By definition phase is the logical division of a project. Like many processes, the development of information systems often follows a life cycle. A project may be single phase or a multiphase project accordingly. Most organizations use system the standard system development life cycle (SDLC). System experts divided SDLC into four major development phases. These phases include System Planning and Selection, System Analysis, System Design and Implementation and operation.

Our new System will have seven logical phases. These are introduction, business area analysis and requirements definition, object-oriented analysis(agile), conclusion of the first three phases, object-oriented design, object-oriented implementation and conclusion of the entire project phases. This report includes up to the conclusion of the first three phases. Each phase of the project will have its own tasks and respective deliverable. The term deliverable describes a product created as part of a project. Deliverable can be product related, such as a piece of hardware or software, or process-related, such as a planning document or meeting minutes. Our project deliverable could be the set of software's and documentations. The table below shows tasks and its corresponding deliverable.

| Title | Deliverable |
|---|---|
| **System Planning and Selection** | Convincing proposal document |
| **Business Area analysis and Requirements Definition** | • Existing systems documentation<br>• Requirements documentation |
| **Object Oriented System Analysis** | • List of User Interfaces<br>• List of Business Rules |

| | |
|---|---|
| | • System Use Case Diagram and Use Case Description<br>• Class Diagram and Class Description<br>• Sequence Diagram<br>• User Interface Prototype |
| **Conclusion** | • Revision of the previous phases of the project and concluding statement. |
| **Object Oriented System Design** | • Class Type Architecture<br>• Class Diagram and Description of Classes<br>• State chart diagram<br>• Component diagram<br>• Collaboration diagram<br>• Deployment diagram<br>• Relational model<br>• Database diagram<br>• User Interface Flow Diagram<br>• User interface design |
| **Object Oriented Implementation** | • Working System Program integrated to database initialized with data |
| **Conclusion** | Revision of the entire project and Concluding paragraph |

Table 1.2 phases and deliverable of the project

## 1.12 Work-break down structure

A work breakdown structure (WBS) is a visual, hierarchical and deliverable oriented deconstruction of a project. It is a helpful diagram for project managers because it allows them to break down their project scope and visualize all the tasks required to complete their projects. All the steps of project work are outlined in the work breakdown structure chart, which makes it an essential project planning tool. The final project deliverable, as well as the tasks and work packages associated with it rest on top of the WBS diagram, and the WBS levels below subdivide the project scope to indicate the tasks, deliverables and work packages that are needed to complete the project from start to finish.

| **phases** | **Tasks** | **Responsible person** |
|---|---|---|
| **1.System planning and selection** | 1.1 Overview | Biruk Amare<br>Haileamlak walelign<br>Kirubel Asnakew |
| | 1.3. Stating Problem statement | |
| | 1.4. defining Objective of the project | |
| | 1.5. Conducting Feasibility study | |
| | 1.6. Analyzing Significance of the system | |
| | 1.7. Defining Beneficiaries of the project | |
| | 1.8. choosing Development methodology | Abdulmenan Mohammed<br>Anteneh Amare<br>Samuel Tesfaye |
| | 1.9. Specifying Development tools and technologies | |
| | 1.10. Scoping of the project | |
| | 1.11. Determining Risks assumptions and constraints | |
| | 1.12 determining phases and deliverable of the project | |
| | 2.1 Overview | |
| | 2.2. Business area analysis | |
| | 2.2.1. Identifying Activities and functions of the organization | |

| | | |
|---|---|---|
| **2.Business area Analysis and requirements dominion** | 2.2.2. Studying Problems of the current system | All members of the group |
| | 2.2.3 Forms and reports of the current system | |
| | 2.2.4. Identifying Players of the current system | |
| | 2.3. Requirements definition | |
| | 2.3.1. Specifying Functional requirements | |
| | 2.3.2. specifying non-functional requirements | |
| **3.Object Oriented Analysis** | 3.1 Overview | All members of the group |
| | 3.2 use case modelling | |
| | 3.2.1 UI identification | |
| | 3.2.2 Business rules identification | |
| | 3.2.3 Actor identification | |
| | 3.2.4 Designing the use case diagram | |
| | 3.2.5 Use case description | |
| | | |
| | 3.3 Conceptual modelling | |
| | 3.3.1 Class diagram | |
| | 3.3.2 Class description | |
| | 3.4. Sequence diagramming | |
| | 3.5. User Interface Prototyping | |

Table 1.3work-break down structure

# 1.13 Project Schedule

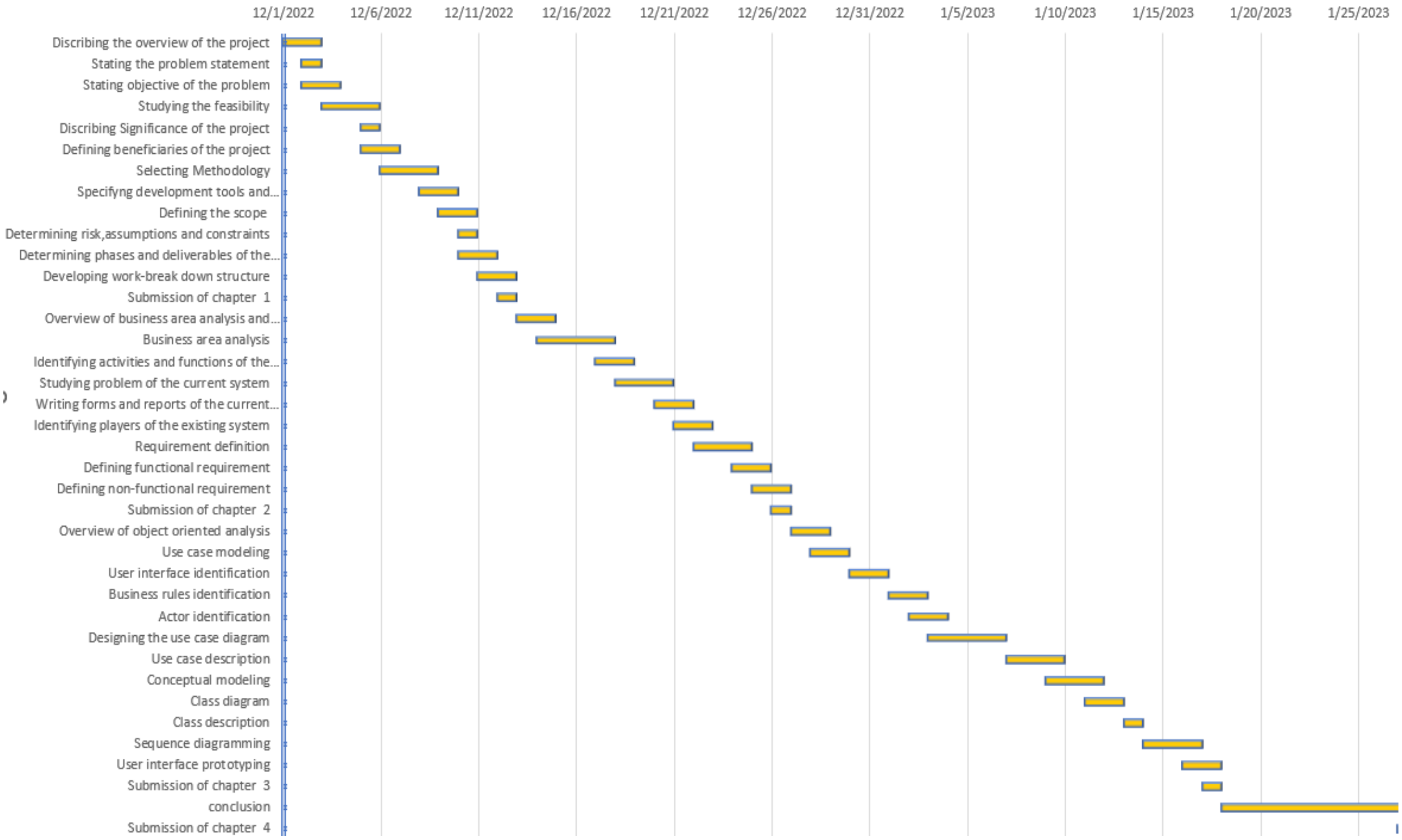


Figure 1.1 project schedule Gantt chart

# 2.Business Area Analysis and Requirement Definition

## 2.1 Overview

So far, the proposal section has covered business requirements. To create a system, we must specify more than just the business requirements. We also need to be aware of what the user requirements are. User requirements will be covered in this chapter. Understanding the system requirements is our top priority because they inform us of the features that our Web based Commodity Exchange System must have in order to fulfill the needs of our users. This chapter also looks at the reports, forms, and concerns about the current Market. Finding the Web based Commodity Exchange System's functional and non-functional requirements will also be of crucial concern.

## 2.2 Business area analysis

In Ethiopia, people buy and sell products in different ways. The most common way of doing this is by using a traditional market where the place is less comfortable, dirty, and jostling. Ethiopian Commodity Exchange is the only company providing commodity exchange services.

The ECX is set up as a private company owned by a partnership of the market actors, members of the exchange, and the Ethiopian government, led by Eleni Gebremedhin a former economist for the International Food Policy Research Institute and the World Bank. As of July 2011, the physical presence of the ECX consists of 55 warehouses in 17 regional locations. It has grown from trading 138,000 tons in its first year to 508,000 tons in its third year, with nearly equal shares of coffee and oilseeds, and pulses. The value of the ECX rose 368% between 2010 and 2011 to reach US$1.1 billion.[1]

As of November 2010, the trading floor in Addis Ababa handled 200 spot contracts in commodities such as coffee, sesame, navy beans, maize, and wheat. It was assessed in July 2011 that membership equaled 243 with clients, who trade through members, about 7,800. Farmer cooperatives represented 2.4 million smallholder farmers, which make up 12% of the membership.

In spite of its remarkable results, ECX does not offer its services to the local market (it focuses solely on exports). Our project will address this challenge.

### 2.2.1 Activities and function of the organization

Farmers are traditionally involved in the production of their goods, and then they sell them through brokers to merchants, and merchants then sell them to customers.

### 2.2.2 Problems of the current system

- **Time**: the current market takes time and energy.
- **Cost**: cost of the product is much more expensive because of free market economy.
- Difficult to compare prices.
- **Information**: farmers and buyers don’t have any information about where to sell and buy products.
- **Service**: Obtaining a product of the desired quality is challenging.
- **Security**: loss of the goods from the marketplace.
- **Efficiency**: It can be very difficult to find a product that you want.

### 2.2.3 Forms and reports of the current system

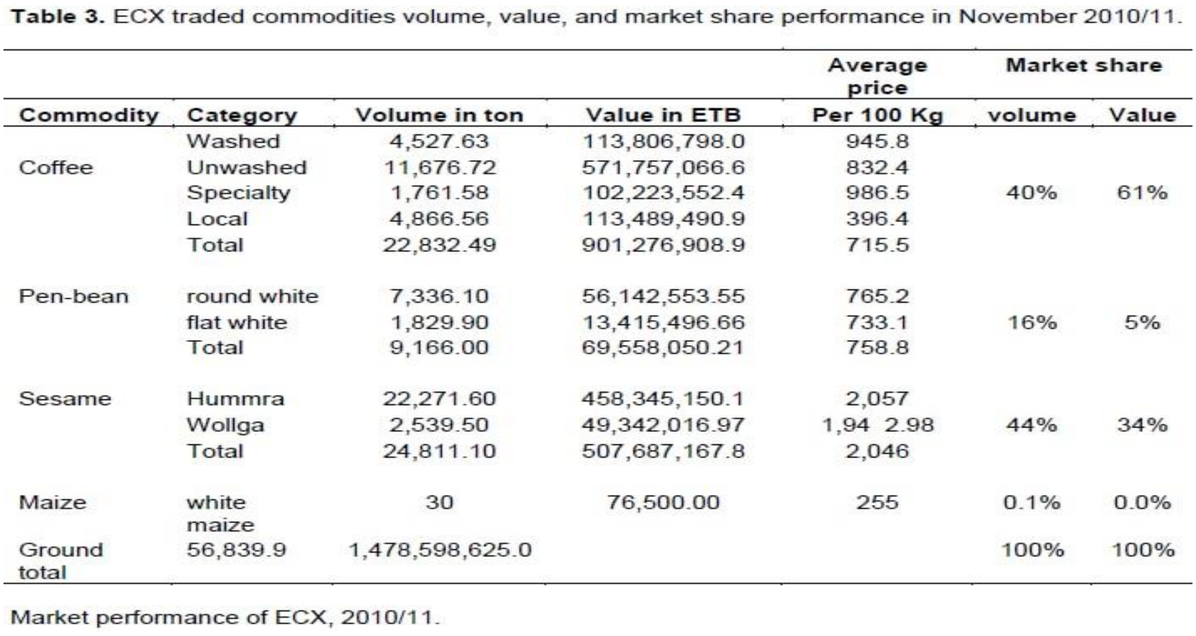


**Table 3.** ECX traded commodities volume, value, and market share performance in November 2010/11.

|  |  |  |  | Average price | Market share |  |
|---|---|---|---|---|---|---|
| **Commodity** | **Category** | **Volume in ton** | **Value in ETB** | **Per 100 Kg** | **volume** | **Value** |
| Coffee | Washed | 4,527.63 | 113,806,798.0 | 945.8 | 40% | 61% |
|  | Unwashed | 11,676.72 | 571,757,066.6 | 832.4 |  |  |
|  | Specialty | 1,761.58 | 102,223,552.4 | 986.5 |  |  |
|  | Local | 4,866.56 | 113,489,490.9 | 396.4 |  |  |
|  | Total | 22,832.49 | 901,276,908.9 | 715.5 |  |  |
| Pen-bean | round white | 7,336.10 | 56,142,553.55 | 765.2 | 16% | 5% |
|  | flat white | 1,829.90 | 13,415,496.66 | 733.1 |  |  |
|  | Total | 9,166.00 | 69,558,050.21 | 758.8 |  |  |
| Sesame | Hummra | 22,271.60 | 458,345,150.1 | 2,057 | 44% | 34% |
|  | Wollga | 2,539.50 | 49,342,016.97 | 1,94 2.98 |  |  |
|  | Total | 24,811.10 | 507,687,167.8 | 2,046 |  |  |
| Maize | white maize | 30 | 76,500.00 | 255 | 0.1% | 0.0% |
| Ground total | 56,839.9 | 1,478,598,625.0 |  |  | 100% | 100% |

Market performance of ECX, 2010/11.

Figure 2.1 Forms and reports of the current system

### 2.2.4 Players of the existing system

**Customer**: A person responsible for purchasing goods from the merchant.
**Farmer**: A person responsible for producing good
**Government**: responsible for managing and controlling the overall market situation.
**Merchant**: responsible for selling goods to the customer.
**Broker**: Facilitate the transaction between farmer and merchant as well as customer and merchant.

## 2.3 Requirements definition

Requirements analysis or requirements engineering is a process used to determine the needs and expectations of a new product. It involves frequent communication with the stakeholders and end-users of the product to define expectations, resolve conflicts, and document all the key requirements[6]. Software system requirements are often classified as functional requirements or non-functional requirements.

### 2.3.1 Functional requirement

**Functional requirements** are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behaviour under specific conditions.[7]

Functional Requirements for the proposed system include:

- User Registration: Web Based Commodity Exchange System have a user registration system that allows users to create accounts, manage their personal information, and view their purchase history.
- Product Catalog: Web Based Commodity Exchange System provide a catalog of products that are available for purchase, including product descriptions, prices, and images.
- Cart: Web Based Commodity Exchange System allow users to add products to a virtual shopping cart and modify the contents of the cart before proceeding to checkout.
- Payment Gateway: Web Based Commodity Exchange integrate with a payment gateway to securely process payments and facilitate the transfer of funds between buyers and sellers(farmers).
- Order Management: Web Based Commodity Exchange System provide an order management system that allows admin to manage orders, track shipments, and generate invoices.
- Delivery Integration: Web Based Commodity Exchange System integrate with delivery providers to provide accurate delivery rates and tracking information to buyers.
- Search Functionality: Web Based Commodity Exchange have a search bar that allows users to search for products by name and category.
- Checkout: Web Based Commodity Exchange System have a secure checkout process that allows users to enter their delivery and billing information.

## 2.3.2 Non-Functional requirement

Non-Functional Requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. [7]

Non-Functional Requirements for the proposed system include:

### Usability

Usability is the degree of ease with which the user will interact with the system to achieve required goals effectively and efficiently.

- We will keep the design simple and consistent.
- We will use product descriptions that are easy to understand.
- We will provide high-quality images that show the product from different angles.
- We will ensure fast load times
- We will use user testing. It allows us to identify usability issues and improve the user experience. We will conduct user testing at different stages of development to ensure that the System is easy to use and meets the needs of your users.

### Reliability

Reliability is defined as the probability of a system or system element performing its intended function under stated conditions without failure for a given period. We will achieve reliability of Web Base Commodity Exchange System with redundant servers and backup systems in place to prevent downtime. We will also develop a disaster recovery plan to ensure business continuity in the event of an unexpected event.

### Performance

One of the benefits of our project is to deliver services quickly and efficiently. To achieve optimal performance the Web Based Commodity Exchange System will be designed with a focus on efficient coding, database optimization, and caching.

### Availability

Availability is the probability that a repairable system or system element is operational at a given point in time under a given set of environmental conditions. Availability is another concern of our system. We will regularly monitor the site’s performance to identify and fix any issues. This includes monitoring server uptime, website response time, and website traffic.

## Scalability

Scalability refers to the ability of a system to perform well under an increased or expanding workload. We will design the Web Based Commodity Exchange System to scale horizontally, with ability to add more servers and resources as traffic grows. We will be achieved this by using cloud-based hosting services that offer auto-scaling capabilities.

## Security

This includes requirements related to the protection of the system and its data from unauthorized access, as well as the ability to detect and recover from security breaches [5]. To achieve the maximum security of Web Based Commodity Exchange System we will use SSL encryption to protect sensitive data during transmission. We will also implement strong password policies, secure user authentication mechanisms, and uses firewalls and intrusion detection systems to prevent unauthorized access.

# 3.Object-oriented analyses

## 3.1 Overview

In the previous chapter, we have clearly identified the problems of the existing system and we proposed an alternative solution. And also, we pointed out the functional and non-functional requirements of the new system. In this chapter, we will apply object-oriented analysis to determine the functions and requirements of the new system, including creating use case diagrams, class diagrams, sequence diagrams, UI designs, and identifying business rules. We will also build prototypes of the system and come up with a concrete visualization of how the new system will look like.

Object-oriented analysis (OOA) looks at the problem domain, with the aim of producing a conceptual model of the information that exists in the area being analyzed without considering any implementation details.

## 3.2 Use case modelling

Use case modelling is a software development technique that is used to facilitate the design and description of a proposed user interface. It involves representing tasks, situations, and interactions between users and the system (computer) as a two-dimensional diagram that defines how the system should respond. The model contains information regarding inputs, outputs, rules, and boundary conditions.

The use case models can help improve the accuracy of requirements analysis, by providing an understanding of what needs to be built in order to satisfy the customer's expectations. They can also assist in improving the process of communication between developers, customers and users, by ensuring all stakeholders have a common understanding of what needs to be implemented. It also helps anticipate risks associated with specific steps within a project or software development initiative.[3]

### 3.2.1 UI Identification

UI identification is a process of identifying user interface (UI) elements within an application so that they can be shown in a user-friendly way. This process involves providing descriptive names, labels, and other information that makes it easier for users to interact with the system.

| ID | User interface | Description | Actors |
|---|---|---|---|
| **UI_01** | Home page | The first page the buyer sees whenever they access the website | Buyer |
| **UI_02** | About us page | The page that says some information about the system | Buyer |
| **UI_03** | Category page | page where products are organized and grouped into categories. It provides customers the ability to quickly find what they are searching for and easily browse through product offerings. | Buyer |
| **UI_04** | Products page | A page listing the various items available for sale. It often includes product photos, descriptions, prices and purchasing options. | Buyer |
| **UI_05** | Sign up/Register page | A page where account is created | Buyer, farmer, admin |
| **UI_06** | Log in page | A page in which a user can fill his log in credentials to access the system | Buyer, farmer, admin, system admin |
| **UI_07** | Cart page | A page where buyer can select the items they would like to purchase and proceed to checkout. | Buyer |
| **UI_08** | Checkout page | A page where buyer review their order and submit payment and delivery information. | Buyer, Bank, Delivery providers |
| **UI_09** | Order confirmation page | is the final page that verifies and confirms the order details and shows payment and shipping information. | Buyer |
| **UI_10** | Admin dashboard page | A page containing representations of business performance and overall insights into the website's activity. | Admin |
| **UI_11** | Contact us page | A page that provides users with contact information and gives them a way to contact by providing a contact form or email address. | Buyer |
| **UI_12** | Manage-category page | A page where admin manages categories | Admin |
| **UI_13** | Manage-product page | A page where admin manages products | Admin |
| **UI_14** | Manage-order page | A page where admin manages orders | Admin |
| **Ul_15** | Manage-admin page | A page where system admin manages admins | System admin |
| **UI_16** | Add-category page | A page where admin add categories | Admin |
| **UI_17** | update-category page | A page where admin update categories | Admin |
| **UI_18** | Add-product page | A page where product is added | Farmer, Admin |
| **Ul_19** | update-product page | A page where admin update a product | Admin |
| **UI_20** | Add-admin page | A page where super admin add admin | System admin |
| **UI_21** | update-admin page | A page where admin update admin | Admin |
| **UI_22** | update-order page | A page where admin update orders | Admin |

Table 3.1 User interface identification

### 3.2.2 Business rule identification

A business rule is a regulation that defines or restricts actions within an organization's operations. They're statements that guide behaviour and determine where, when, why and how to carry out business tasks.[8]

#### List of business rule

| Rules | Name | Description | Constraint |
|---|---|---|---|
| R1 | Free shipping | Buyer gets free shipping | Order must be over 3000 ETB |
| R2 | Payment method | Buyer selects payment options | Buyer must have bank account |
| R3 | Security | Security of the system | Every user must have unique credential which are verified by the system |
| R4 | Order management | Admin delete and update orders | Admin must first login to the system |
| R5 | Prices | Prices of different products | All prices are listed in ETB |

Table 3.2 business rule

### 3.2.3 Actor Identification

Object-oriented system analysis involves the identification of objects and actors for a given system. Actors are users or systems outside an object, but interact with that object. Identification of actors involves studying the external interface of the system and analysing user and other systems' interactions or communications with the object in order to determine the need for actor interaction with the object. Actors could be users, hardware, software, or even automated systems.

**List of actors**

- Buyer
- Farmer(seller)
- Admin
- System admin
- Bank
- Delivery provider

## 3.2.4 Designing the use case diagram

Use case diagram is a behavioural UML diagram type and frequently used to analyse various systems. They enable you to visualize the different types of roles in a system and how those roles interact with the system. It mainly had four parts.

**Actor**-represents an external entity that interacts with a system
**Use case**: A use case represents a function or an action within the system. It’s drawn as an oval and named with the function.
**System**: The system is used to define the scope of the use case and drawn as a rectangle.
**Relationship**: The relationship between actor and the system and it can be.

## List of use case

- Register /sign up
- Login
- Browse products
- Browse categories
- Search products
- Add to cart
- Choose payment method
- Choose delivery option
- Checkout
- Manage product
- Manage category
- Manage admin
- Manage order
- Add product
- Manage balance
- Manage delivery

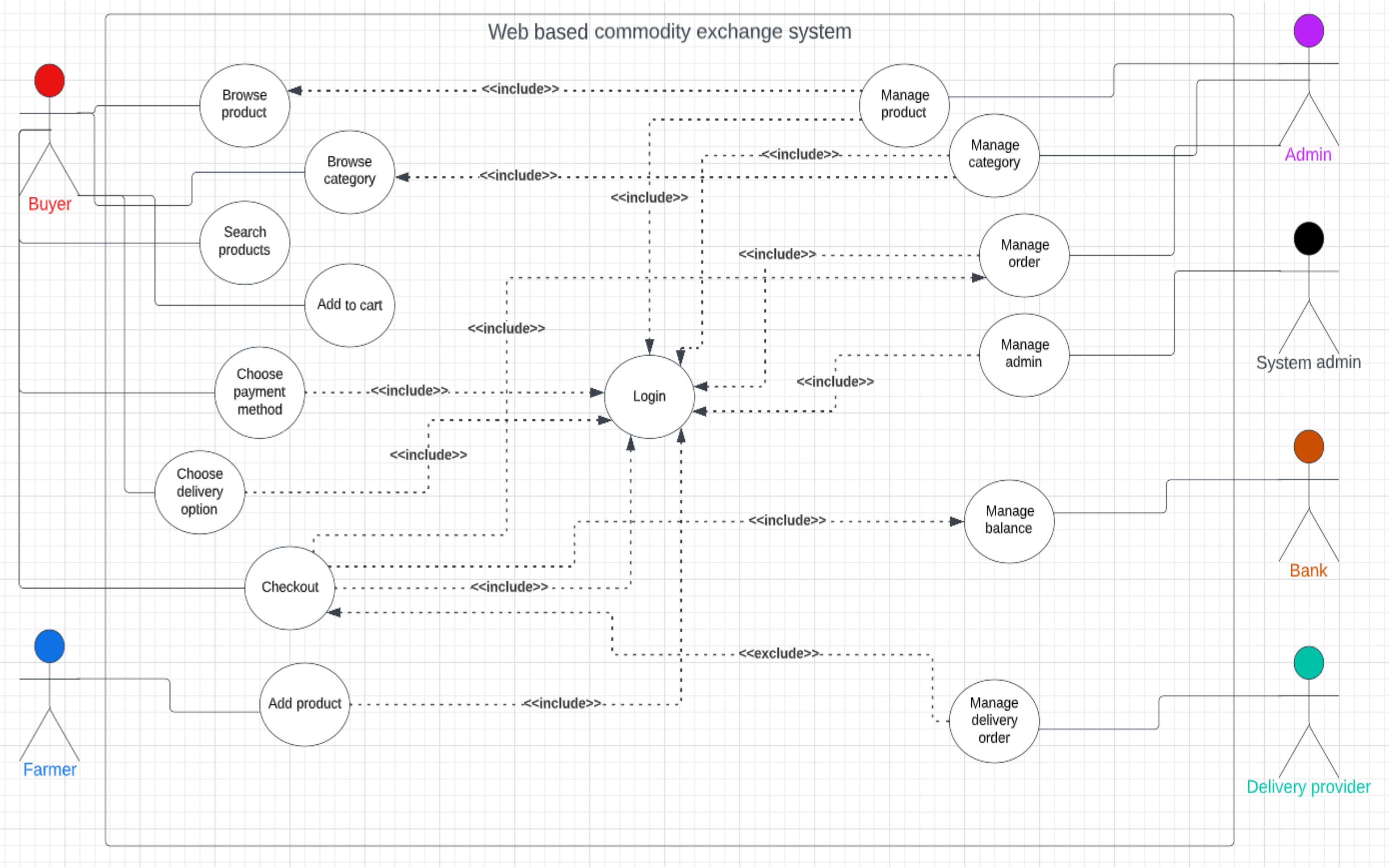


Figure 3.1 use case diagram

## 3.2.5 Use case description

Below are the lists of the use cases that are identified for this. The system use case description contains:

- Use case name
- Use case identification number
- Actor
- Description
- Pre-condition
- Post-condition
- Basic course of Action
- Alternative course of Action

| | |
|---|---|
| Use case name | Register |
| Use case ID | UID_01 |
| Description | it is the act of creating an account by providing personal data |
| Actor | Buyer, farmer |
| Pre-condition | None |
| Post-condition | Buyer registered |
| Basic course of Action | 1.user wants to register<br>2. user opens home page<br>3.user clicks register button<br>3.user fill fullname, email, password, contact and address information<br>4.user clicks “register” button<br>5. System checks and validate if fullname, email, password, contact and address information are valid.<br>6. System displays successfully registered message<br>7. The use case ends |
| Alternative course of Action | A5 System checks and validate if fullname, email, password, contact and address information are valid.<br>A5.1 System checks fullname, email, password, contact and address information are not valid and display invalid input message<br>A5.2 The use case continues at step 4 |

Table 3.3 Use case description for system use case register

| | |
|---|---|
| Use case name | Login |
| Use case ID | UID_02 |
| Description | Login is the way to access the system |
| Actor | Buyer, farmer, admin, system admin |
| Pre-condition | UID_01 for buyer, farmer, admin and None for system admin |
| Post-condition | Buyer, farmer, admin and system admin logged in |
| Basic course of Action | 1.user wants to login<br>2.user opens home page<br>3.User clicks “login” button<br>4.User enters user-name and password, and clicks the “login” button.<br>5. The system validates whether the username and the password are correct.<br>6. The user will be logged into the system.<br>7. The use case ends. |
| Alternative course of Action | A5 System checks and validate if the username and password are valid<br>A5.1 System checks that username and password are not valid and display invalid input message<br>A5.2 The use case continues at step 4 |

Table 3.4 Use case description for system use case login

| | |
|---|---|
| Use case name | Browse Products |
| Use case ID | UID_03 |
| Description | Browsing and viewing different products |
| Actor | Buyer |
| Pre-condition | None |
| Post-condition | Viewed list of products |
| Basic Course of Action | 1.Buyer click the “Products” button found in the home page navigation bar.<br>2. The system displays the available list of products<br>3. The use case ends. |
| Alternative course of Action | None |

Table 3.5 Use case description for system use case browse products

| | |
|---|---|
| Use case name | Browse categories |
| Use case ID | UID_04 |
| Description | Browsing and viewing different categories |
| Actor | Buyer |
| Pre-condition | None |
| Post-condition | Viewed list of categories |
| Basic course of Action | 1. Buyer clicks the “Category” button found in the home page navigation bar.<br>2. The system displays the categories of the products.<br>3. The use case ends. |
| Alternative course of Action | None |

Table 3.6 Use case description for system use case browse categories

| | |
|---|---|
| Use case name | Add to cart |
| Use case ID | UID_05 |
| Description | Adding choice of products to a cart. |
| Actor | Buyer |
| Pre-condition | None |
| Post-condition | Choice of products are added to the cart |
| Basic course of Action | 1.buyer wants to add product to cart<br>2.buyer opens product page<br>3.buyer choose product and click add to cart button |
| Alternative course of Action | None |

Table 3.7 Use case description for system use case add to cart

| | |
|---|---|
| Use case name | Search Products |
| Use case ID | UID_06 |
| Description | Searching choice of product using the search bar |
| Actor | Buyer |
| Pre-condition | None |
| Post-condition | Product searched and system displayed different product based On the search |
| Basic course of Action | 1. Buyer clicks the search bar on the products page and insert the name of the product of their choice<br>2.Buyer click on search button<br>3. The system displays the searched products. |

| | |
|---|---|
| | 3. The use case ends. |
| Alternative course of Action | None |

Table 3.8 Use case description for system use case search product

| | |
|---|---|
| Use case name | Choose payment Method |
| Use case ID | UID_07 |
| Description | Choosing payment method that fits with the buyer. |
| Actor | Buyer |
| Pre-condition | UID_02 for buyer |
| Post-condition | Payment method selected |
| Basic course of Action | 1.The system displays the available ways of payment.<br>2. The customer selects their choice of payment.<br>3, The use case ends. |
| Alternative course of Action | None. |

Table 3.9 Use case description for system use case choose payment method

| | |
|---|---|
| Use case name | Choose Delivery Option |
| Use case ID | UID_08 |
| Description | Choosing delivery option that fits with the buyer. |
| Actor | Buyer |
| Pre-condition | UID_02 for buyer |
| Post-condition | Delivery option selected |
| Basic course of Action | 1. The system displays available delivery options<br>2. The customer selects their choice of delivery method<br>3. The use case ends. |
| Alternative course of Action | None. |

Table 3.10 Use case description for system use case choose delivery option

| | |
|---|---|
| Use case name | Checkout |
| Use case ID | UID_09 |
| Description | Finishing purchase |
| Actor | Buyer |

| | |
|---|---|
| Pre-condition | UID_02 for buyer |
| Post-condition | Products purchased |
| Basic course of Action | 1.buyer wants to buy<br>2.buyer opens product page<br>3.Buyer choose product and click add to cart button<br>2. Buyer clicks proceed to checkout button found in cart page<br>3.Buyer fill billing and delivery informations<br>4.Buyer clicks checkout button<br>5.the System checks and validate if the billing and delivery informations are valid<br>6.Finally the buyer will be provided with detail report of the order |
| Alternative course of Action | 5 System checks and validate if the billing and delivery informations are valid<br>A5.1 System checks that billing and delivery informations are not valid and display invalid input message<br>A5.2 The use case continues at step3 |

Table 3.11 Use case description for system use case checkout

| | |
|---|---|
| Use case name | Manage product |
| Use case ID | UID_10 |
| Description | Add, view, update and delete product |
| Actor | Admin |
| Pre-condition | UID_02 for admin |
| Post-condition | Admin managed product |
| Basic course of Action | 1. The use case starts when the admin navigates to manage-product page<br>2. System displays all products with their description and operations<br>3. Admin manage products<br>4. The use case ends |
| Alternative course of Action | None |

Table 3.12 Use case description for system use case manage product

| | |
|---|---|
| Use case name | Manage category |
| Use case ID | UID_11 |
| Description | Add, view, update and delete category |
| Actor | Admin |
| Pre-condition | UID_02 for admin |
| Post-condition | Admin managed category |
| Basic course of Action | 1. The use case starts when the admin navigates to manage-category page<br>2. System displays all categories with their description and operations<br>3. Admin manage categories<br>4. The use case ends |
| Alternative course of Action | None |

Table 3.13 Use case description for system use case manage category

| | |
|---|---|
| Use case name | Manage order |
| Use case ID | UID_12 |
| Description | View, delete and update orders |
| Actor | Admin |
| Pre-condition | UID_02 for admin |
| Post-condition | Admin managed order |
| Basic course of Action | 1. The use case starts when the admin navigates to manage-order page<br>2. System displays all orders with their description and operations<br>3. Admin manage order<br>4. The use case ends |
| Alternative course of Action | None |

Table 3.14 Use case description for system use case manage order

| | |
|---|---|
| Use case name | Manage admin |
| Use case ID | UID_13 |
| Description | Add, view and delete admins |
| Actor | System admin |
| Pre-condition | UID_02 for system admin |
| Post-condition | System admin managed admin |
| Basic course of Action | 1. The use case starts when the system admin navigates to admins page<br>2. System displays all admins with their description and operations<br>3. System admin manage admins<br>4. The use case ends |
| Alternative course of Action | None |

Table 3.15 Use case description for system use case manage admin

| | |
|---|---|
| Use case name | Add product |
| Use case ID | UID_14 |
| Description | Adding new product to the system |
| Actor | Farmer |
| Pre-condition | UID_02 for farmer |
| Post-condition | Farmer added new product |
| Basic course of Action | 1. The use case starts when the farmer navigates to add product page<br>2. System displays form that allow farmer to fill product detail<br>3. Farmer fill product detail and click add button<br>4. The use case ends |
| Alternative course of Action | A3 System checks and validate if the inputs are valid<br>A3.1 System checks that inputs are not valid and display error message<br>A3.2 The use case continues at step 2 |

Table 3.16 Use case description for system use case add product

| | |
|---|---|
| Use case name | Manage balance |
| Use case ID | UID_15 |
| Description | Check the balance of the buyer |
| Actor | Bank |
| Pre-condition | None |
| Post-condition | Balance of the buyer checked |
| Basic course of Action | 1.wait for buyer to make order<br>2.checks balance of buyer before completing order<br>3.use case ends |
| Alternative course of Action | None |

Table 3.17 Use case description for system use case manage balance

| | |
|---|---|
| Use case name | Manage delivery |
| Use case ID | UID_16 |
| Description | Managing delivery for buyer |
| Actor | Delivery provider |
| Pre-condition | None |
| Post-condition | Delivery managed |
| Basic course of Action | 1.wait for buyer to order delivery<br>2.manage delivery for the buyer<br>3.use case ends |
| Alternative course of Action | None |

Table 3.18 Use case description for system use case manage delivery

## 3.3 Conceptual modeling

**A conceptual model** is a representation of a system. It shows how people, places, and things interact. They show the real-world features and interactions of your design idea. In other words, a conceptual model is an abstraction of a piece of the real world or a design you plan to bring into the real world.

Class diagram contains:

- The name classes
- The interrelationships (including inheritance, aggregation, and association)
- the operations
- attributes of the classes

### 3.3.1 class diagram

**A class diagram** is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, objects, attributes, operations and relationships. They can also be used for visualizing, documenting and constructing components of software or hardware systems.

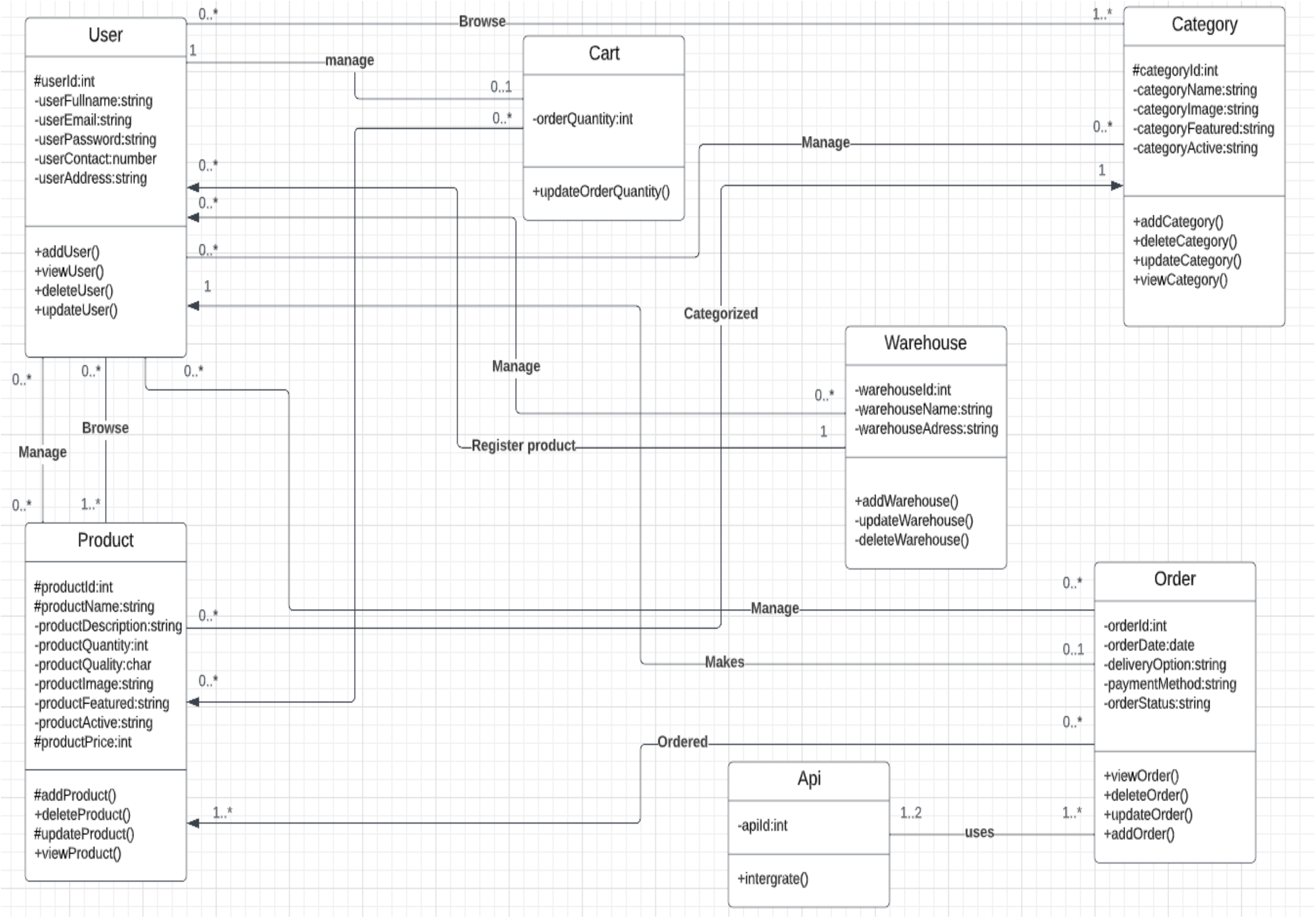


Figure 3.2 class diagram

### 3.3.2 Class description

| Class | User | It includes buyers, farmers and admins |
|---|---|---|
| Attribute | memberId | Type:int<br>Visibility: Protected |
| | memberFullName | Type: string<br>Visibility: private |
| | member Email | Type: string<br>Visibility: private |
| | memberPassword | Type: string<br>Visibility: private |
| Method | Add Member () | Visibility: public<br>Parameters: memberId, memberFullName, memberEmail, memberPassword |
| | ViewMember () | Visibility: public |
| | DeleteMember() | Visibility: public |

Table 3.19 user class description

| Class | Admin | It includes all registered admins |
|---|---|---|
| Attribute | AdminId | Type:int<br>Visibility: private |
| | AdminFullName | Type:string<br>Visibility: private |
| | AdminUserName | Type:string<br>Visibility: private |
| | AdminPassword | Type:string<br>Visibility: private |
| Method | UpdateAdmin() | Visibility: Public<br>Parameters: AdminId, AdminFullName, AdminUserName, AdminPassword |
| | ViewAdmin() | Visibility: Public |
| | DeleteAdmin() | Visibility: Public |
| | AddAdmin() | Visibility: Public<br>Parameters:AdminId, AdminFullName, AdminUserName, AdminPassword |

Table 3.20 admin class description

| **Class** | **Product** | **It includes all registered Products** |
|---|---|---|
| **Attribute** | ProductId | Type:int<br>Visibility: Protected |
| | ProductName | Type:string<br>Visibility: private |
| | ProductDescription | Type:string<br>Visibility: private |
| | ProductQuantity | Type:int<br>Visibility: private |
| | ProductQuality | Type: char<br>Visibility: private |
| | ProductImage | Type:string<br>Visibility: private |
| | CatagoryId | Type:int<br>Visibility: private |
| | ProductFeatured | Type:string<br>Visibility: private |
| | ProductActive | Type:string<br>Visibility: private |
| **Method** | AddProduct() | Visibility: public<br>Parameters: ProductId, ProductName, ProductDescription, ProductQuantity, Product Quality, ProductImage, CatagoryId, ProductFeatured, ProductActive |
| | DeleteProduct() | Visibility: public |
| | UpdateProduct() | Visibility: public<br>Parameters: ProductId, ProductName, ProductDescription, ProductQuantity, Product Quality, ProductImage, CatagoryId, ProductFeatured, ProductActive |
| | ViewProduct() | Visibility: public |

Table 3.21 product class description

| **Class** | **Category** | **It includes all registered categories** |
|---|---|---|
| **Attribute** | CatagoryId | Type:int<br>Visibility: protected |
| | CatagoryName | Type:string<br>Visibility: private |
| | CatagoryImage | Type:string<br>Visibility: private |
| | CatagoryFeatured | Type:string |

| | | |
|---|---|---|
| | | Visibility: private |
| | CatagoryActive | Type: string<br>Visibility: private |
| Method | AddCatagory() | Visibility: public<br>Parameters: CatagoryId, CatagoryName, CatagoryImage, CatagoryFeatured, CatagoryActive |
| | ViewCatagory() | Visibility: public |
| | DeleteCatagory() | Visibility: public |
| | UpdateCatagory() | Visibility: public<br>Parameters: CatagoryId, CatagoryName, CatagoryImage, CatagoryFeatured, CatagoryActive |

Table 3.22 category class description

| **Class** | **Order** | **It includes all orders made by member** |
|---|---|---|
| **Attribute** | OrderId | Type:int<br>Visibility: private |
| | OrderDate | Type: date<br>Visibility: private |
| | MemberId | Type:int<br>Visibility: private |
| | ProductId | Type:int<br>Visibility: private |
| | DeliveryOption | Type: string<br>Visibility: private |
| | PaymentMethod | Type:string<br>Visibility: private |
| | CustomerAddress | Type:string<br>Visibility: private |
| | CustomerContact | Type:number<br>Visibility: private |
| | OrderStatus | Type:string<br>Visibility: private |
| **Method** | ViewOrder() | Visibility: public |
| | DeleteOrder() | Visibility: public |
| | UpdateOrder() | Visibility: public<br>Parameters: orderStatus |

Table 3.23 order class description

# 3.4 sequence diagramming

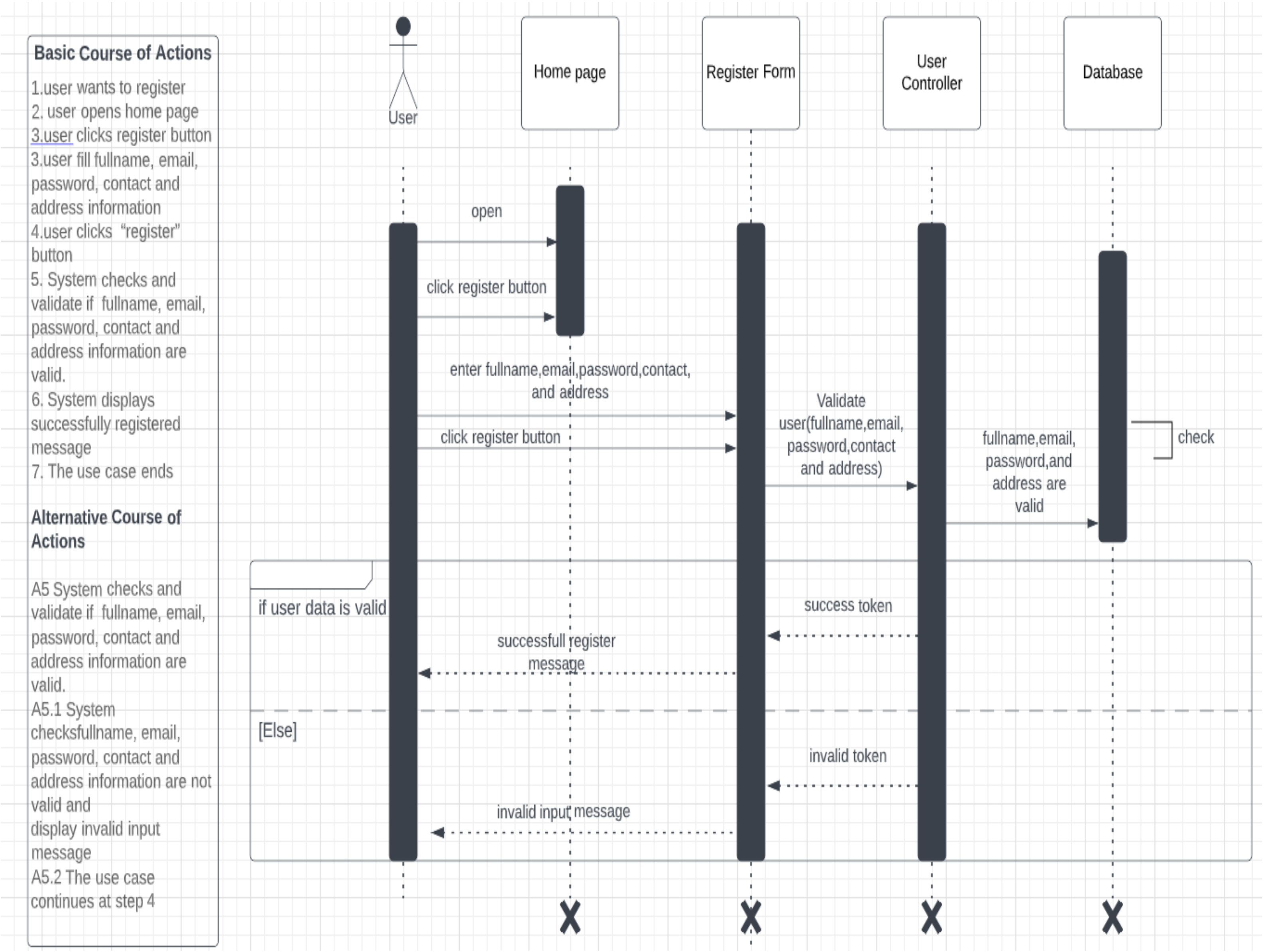
Sequence diagramming is a type of UML (Unified Modeling Language) diagram that shows the order that activities, events, and operations occur in a system or process. It typically employs a vertical timeline that connects multiple lifelines representing different objects or components involved in the system, as well as arrows indicating message/signal flows between them. Sequence diagrams are used to illustrate complex interactions among objects, and can help identify logic errors when developing software systems.

## sequence diagram consists of the following main notations [5]

**Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.

**Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline.

**Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.



Basic Course of Actions

1.user wants to register
2. user opens home page
3.user clicks register button
3.user fill fullname, email, password, contact and address information
4.user clicks "register" button
5. System checks and validate if fullname, email, password, contact and address information are valid.
6. System displays successfully registered message
7. The use case ends

Alternative Course of Actions

A5 System checks and validate if fullname, email, password, contact and address information are valid.
A5.1 System checksfullname, email, password, contact and address information are not valid and display invalid input message
A5.2 The use case continues at step 4

User
Home page
Register Form
User Controller
Database
open
click register button
enter fullname,email,password,contact, and address
click register button
Validate user(fullname,email, password,contact and address)
fullname,email, password,and address are valid
check
if user data is valid
success token
successfull register message
[Else]
invalid token
invalid input message

Figure 3.3 sequence diagram for register

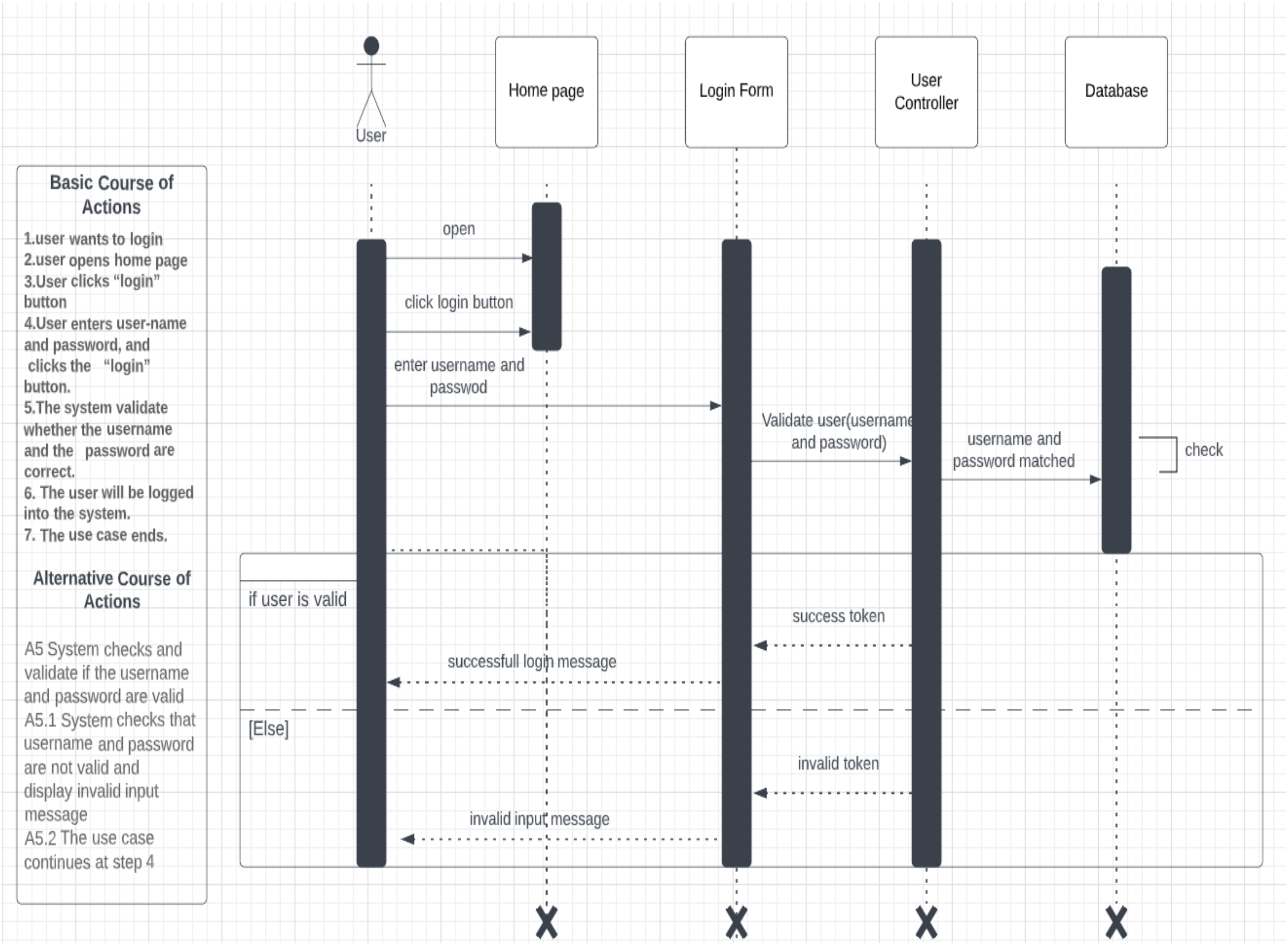
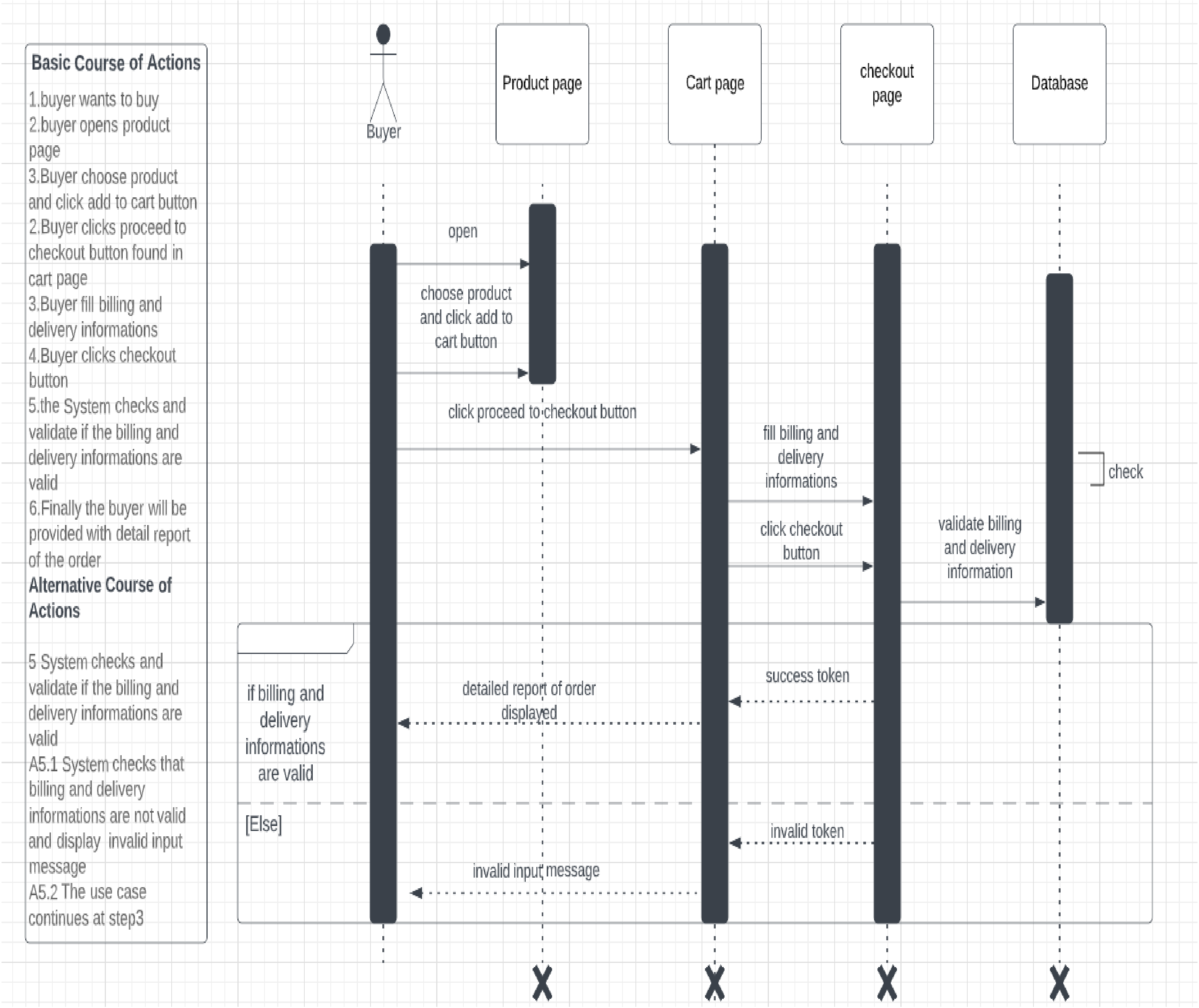

User
Home page
Login Form
User Controller
Database
Basic Course of Actions
1.user wants to login
2.user opens home page
3.User clicks "login" button
4.User enters user-name and password, and clicks the "login" button.
5.The system validate whether the username and the password are correct.
6. The user will be logged into the system.
7. The use case ends.
Alternative Course of Actions
A5 System checks and validate if the username and password are valid
A5.1 System checks that username and password are not valid and display invalid input message
A5.2 The use case continues at step 4
open
click login button
enter username and paswod
Validate user(username and password)
username and password matched
check
if user is valid
success token
successfull login message
[Else]
invalid token
invalid input message

Figure 3.4 sequence diagram for login



**Basic Course of Actions**

1.buyer wants to buy
2.buyer opens product page
3.Buyer choose product and click add to cart button
2.Buyer clicks proceed to checkout button found in cart page
3.Buyer fill billing and delivery informations
4.Buyer clicks checkout button
5.the System checks and validate if the billing and delivery informations are valid
6.Finally the buyer will be provided with detail report of the order

**Alternative Course of Actions**

5 System checks and validate if the billing and delivery informations are valid
A5.1 System checks that billing and delivery informations are not valid and display invalid input message
A5.2 The use case continues at step3

Buyer
Product page
Cart page
checkout page
Database
open
choose product and click add to cart button
click proceed to checkout button
fill billing and delivery informations
click checkout button
validate billing and delivery information
check
if billing and delivery informations are valid
success token
detailed report of order displayed
[Else]
invalid token
invalid input message

Figure 3.5 sequence diagram for checkout

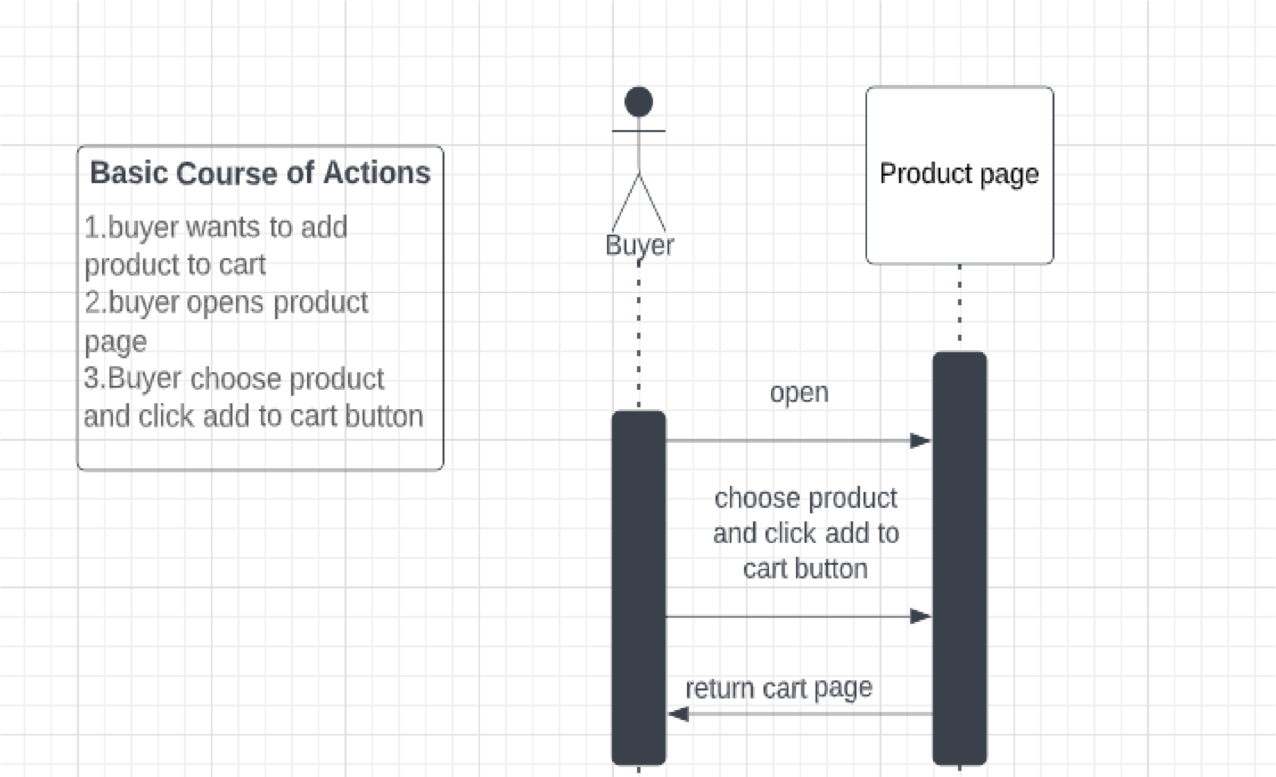

Basic Course of Actions
1.buyer wants to add product to cart
2.buyer opens product page
3.Buyer choose product and click add to cart button
Buyer
Product page
open
choose product and click add to cart button
return cart page

Figure 3.6 sequence diagram for add to cart

## 3.5 User interface prototyping [4]

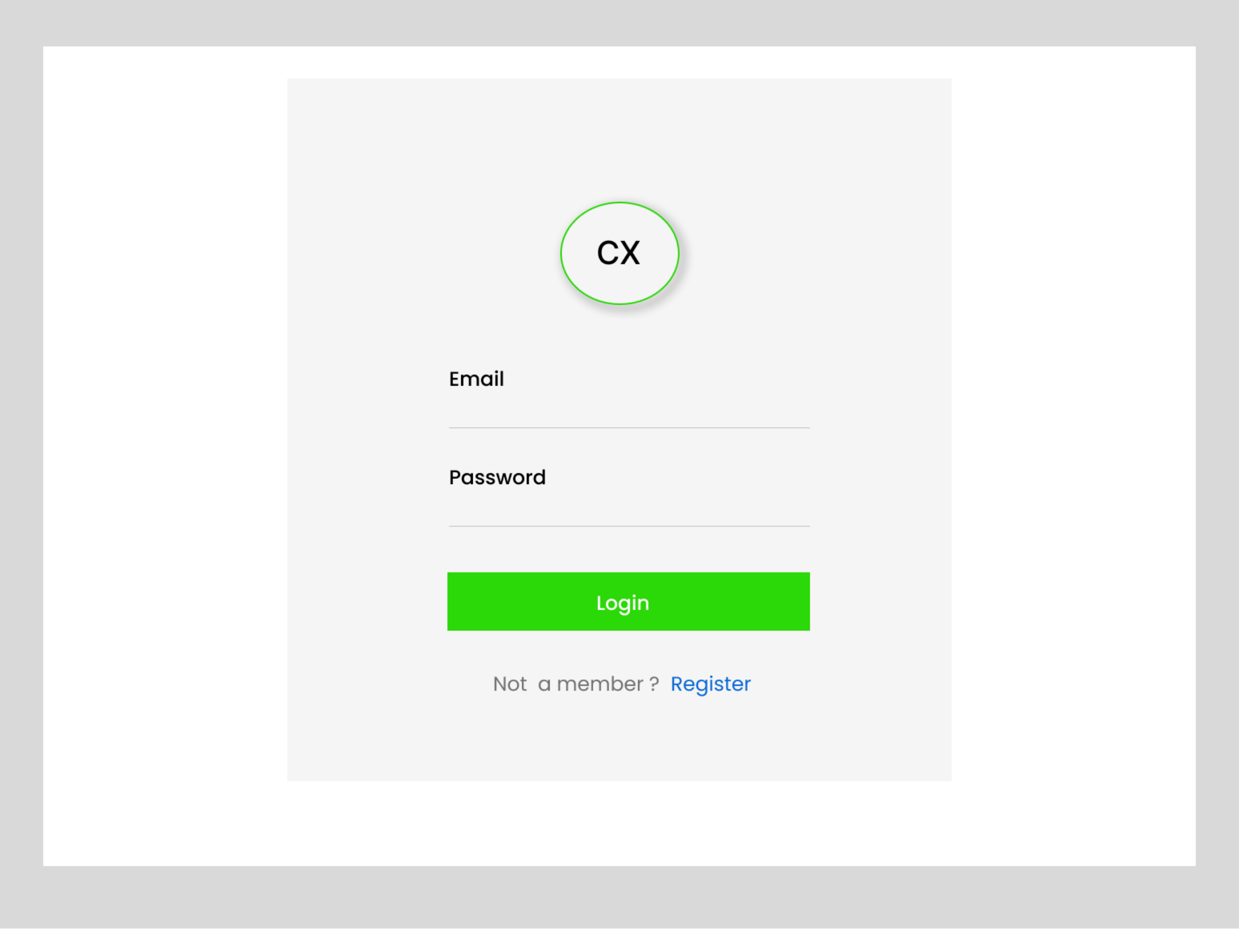
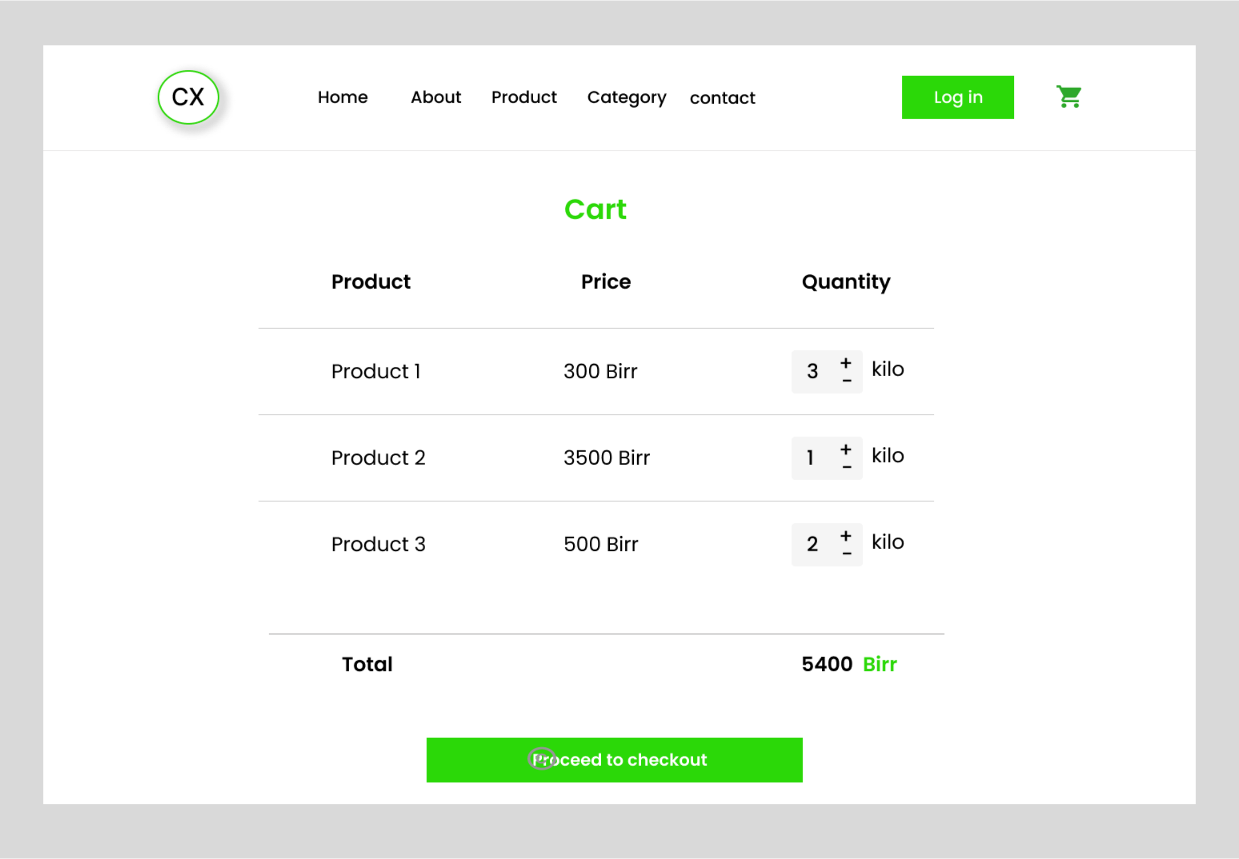


Figure 3.7 Login page

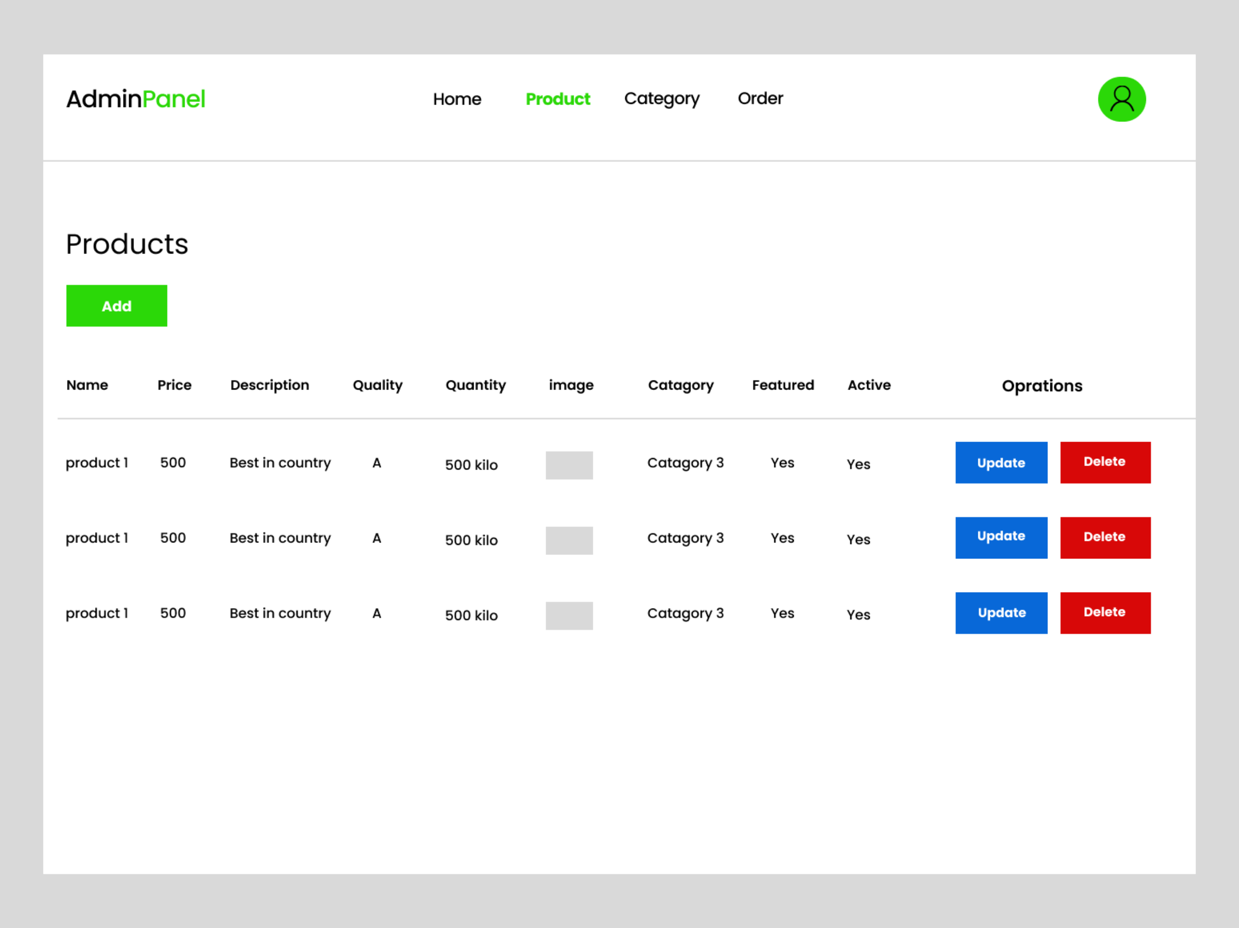


Cart

| Product | Price | Quantity |
|---|---|---|
| Product 1 | 300 Birr | 3 + − kilo |
| Product 2 | 3500 Birr | 1 + − kilo |
| Product 3 | 500 Birr | 2 + − kilo |

| Total | 5400 Birr |
|---|---|

Figure 3.8 Cart page



Products

| Name | Price | Description | Quality | Quantity | image | Catagory | Featured | Active | Oprations |
|---|---|---|---|---|---|---|---|---|---|
| product 1 | 500 | Best in country | A | 500 kilo | ▨ | Catagory 3 | Yes | Yes | Update Delete |
| product 1 | 500 | Best in country | A | 500 kilo | ▨ | Catagory 3 | Yes | Yes | Update Delete |
| product 1 | 500 | Best in country | A | 500 kilo | ▨ | Catagory 3 | Yes | Yes | Update Delete |

Figure 3.9 Manage-product page

# 4.Conclusion

In the previous three chapters of the document, we worked on different issue. Just to remember in the first chapter we discussed about the overview of the project, statement of the problem. we defined who are the key players in the market. Based on the problem we have seen we defined our scope to address the problems. We planned a system that can overcome these problems per our knowledge of problem solving. We believe, the planned system i.e., web-based commodity exchange system can alleviate the problems which exists in the market. We also noted that the benefits of this system and its significant after the deployment of the system.

In the second chapter, we performed business area analysis. We studied how member interact with the current system; what problems do they have in more detail we analysed; who the participants are. we determined our system to fulfil certain requirements. We classified the requirements in to two categories i.e., Functional and non-functional requirements.

In the third chapter we designed conceptual modeling this includes designing use case modeling and we clarify each use case then we designed the class diagram with description we also draw a sequence diagram which show the sequence of action in the proposed system at last we designed user interface design which simulate the proposed system.

In general, in the previous three chapter we understood that how the current market works with its problem and we could propose a new system to solve this problem and we designed use case class and sequence diagram and UI prototype that enables us to go to the next phase of industrial project.

# References

1.International trade administration.(2022).*Ethiopia commercial guide market overview*.[online][Accessed on 4th December 2022] https://www.trade.gov/country-commercial-guides/ethiopia-market-overview

2.The Ethiopia commodity exchange. (No date). *How Ecx works*.[online][Accessed on 4th December 2022] https://www.ecx.com.et/Pages/AboutUs.aspx

3.Bittner, K., & Spence, I. (2003). *Use case modeling. Addison-Wesley Professional*.

4.Galitz, W. O. (2007). *The essential guide to user interface design: an introduction to GUI design principles and techniques. John Wiley & Sons.*

5.Geeksforgeeks.(2022).*UnifiedModelingLanguage(UML)*|*Sequence diagram*.[online][Accessed on 15th January 2023] https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/

6.simplilearn.(2023).*What is Requirement Analysis: Overview, Applications, and More*.[online][Accessed on 23th December 2022] https://www.simplilearn.com/what-is-requirement-analysis-article

7. altexsoft.(2021). *Functional and Nonfunctional Requirements: Specification and Types*.[online][Accessed on 23th December 2022]
https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/

8.indeed (2022).*What is business rule*.[online][Accessed on 2th January 2022] https://www.indeed.com/career-advice/career-development/business-rule