

KACHABAZAR

The hassle free market that we have all been dreaming of is finally here.

A Farmer Friendly
Ecommerce

Table of Contents

Chapter 1 - Introduction	5
1.1 Abstract	5
1.2 Overview.....	5
1.3 Ease of Use	6
1.4 Background Study.....	6
1.5 Project Planning.....	6
1.5.1 Software Development Lifecycle Presets	7
Chapter 2 - System Design	9
2.1 Planning.....	9
2.2 Scenario Based Model.....	11
2.2.1 Scenario 1:	11
2.2.2 Scenario 2:	11
2.2.3 Scenario 3:	11
2.2.4 Scenario 4:	11
2.2.5 Use Cases	11
2.2.6 Use Case Diagram.....	19
2.2.7 Activity Diagram.....	20
2.2.8 Swimlane Diagram.....	23
2.3 Flow Oriented Model.....	24
2.3.1 DFD – Level 0.....	24
2.3.2 DFD – Level 1.....	25
2.4 Behavioral Model	26
2.4.1 State Diagram.....	26
2.4.2 Sequence Diagram	27
2.5 Class-Based Model.....	28
2.5.1 Scenario analysis.....	28
2.5.2 Class Declaration Table.....	30
2.5.3 Class Responsibility Collaboration (CRC)	32
Chapter 3 - Technology Requirements.....	35
3.1 Hardware Requirements	35
3.1.1 For Website:	35
3.1.2 For Android App:.....	35
3.1.3 For Server.....	35

3.2	Software Requirements	35
3.2.1	For Website	35
3.2.2	For Android App.....	35
3.2.3	For server	35
3.3	Technologies	36
3.3.1	For Website	36
3.3.2	For android application	36
Chapter 4 - Database and Tables		37
4.1	Database Design	37
4.2	All Table List	38
4.3	Category Table	38
4.4	Product Table	39
4.5	Seller Product Table	39
4.6	Order Table.....	40
Chapter 5 - Project View		41
5.1	Homepage	41
5.2	Single Product Page	41
5.3	Inventory Products	42
5.4	Cart Page	42
5.5	Checkout Page	43
5.6	Payment Gateway page.....	43
5.7	Payment Successful Card.....	44
5.8	Profile Information.....	44
5.9	Android App – Home Page & Category Product page.....	45
5.10	Android App – Cart & Cart Checkout	46
5.11	Android App – Payment Methods.....	47
5.12	Android App – Seller page and Adding product.....	48
5.13	Admin Login Page.....	49
5.14	Admin Homepage	49
5.15	Admin Product Page	50
5.16	Admin Area control page.....	50
5.17	Admin Panel Report Generation	51
Chapter 6 - Software Testing		52
6.1	White Box Testing.....	52

6.2	Black Box Testing	52
6.3	User Acceptance Testing (UAT).....	53
6.3.1	Scope (In Scope – Out of Scope).....	53
6.3.2	Assumptions and Constraints	53
6.3.3	Constraints	53
6.3.4	Risks.....	54
6.3.5	Team Roles & Responsibilities	55
6.3.6	Entry Criteria.....	55
6.3.7	Requirements-Based Test Cases – APP	55
6.3.8	Test Results – APP.....	57
6.3.9	Requirements-Based Test Cases – Website.....	58
6.3.10	Test Results – Website	60
6.3.11	Requirements-Based Test Cases – Website.....	61
6.3.12	Test Results – Website	63
Chapter 7	- Conclusion	64

Chapter 1 - INTRODUCTION

1.1 ABSTRACT

A well-defined software plan, design, architecture can make complex work easy and understandable from every perspective while making the objectives easily attainable. The KachaBazar is a web-based service that offers several new functionalities than regular eCommerce like removing the middleware between consumer and seller, thus lowering the price, pre-ordering for a fragile product like fish. Furthermore, the services have implemented a website and android application. The usage of restful API and servlets (small Java class that handles request/response) makes the implementation-independent thus free of dependencies while the data in all the implementations are perfectly synchronized. In this report, I represent such a document for the project that complies with the state-of-the-art methods recommended by many experts to extract the optimum output by using the resources effectively and efficiently by all possible means. We used the spiral model which is a component-based loosely coupled model, that is flexible to the future changes and friendly with debugging and testing and it makes one of the reasons to be more cost-efficient and holds greater value. The business value of this project among the consumer and businesses in Bangladesh will be greater and the additional cost of the middle man will be cut thus the price should be stable and affordable for both producers and customers. Since people from every ethnicity and social level will use this service, we used the best practices and made sure that the applications are minimum in size and supported in most of the devices if not all of them.

1.2 OVERVIEW

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing websites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

The term 'KachaBazar' refers to the regular groceries for the household to live by. The main problem with this kind of product is that they are often fragile and short-lived. In this project, we tried to solve different strategic and technical problems with the grocery marketplace.

We have used Java server fragments namely, sublets as our logic and request handler, and different design patterns namely, MVC, Null pointer, DAO, etc. to implement higher-level logic. Then the payment method was handled by SSL Commerz integration which made it possible to accept all types of banking from mobile to internet to card. Then the implementation of the android app was done with the security and compatibility consideration in mind.

The delivery process is initially handled by the super and sub-admin who allocated an order to the delivery man and there is another app developed for them to pick the order and change their status accordingly.

1.3 EASE OF USE

The main application is hosted on a CentOS server and with the help of additional services such as the Tomcat, Apache, and PhpMyAdmin. The admin can easily log in to the public IP of the server and get access to the services so it makes it easy for him/her to do maintenance tasks such as taking backup, restarting the server, etc.

The website for consumers is supported on every device that has a browser and it is responsive. There are several options for searching and choosing the preferred product and purchasing them with the help of SSL Commerz. And the android app is developed with SDK 26 to make sure that at least 95% of android phones will support it.

The super admin and sub admin both have their portal with access control and report generation. They can add, remove and update products, categories, sub-categories, etc. They are also able to add seller and delivery man according to application.

The Seller has their portal and app where they can upload the product that will lead them to sale. The delivery man also has an app to get allocated orders and they can change the order status which confirms the admin about the payment and delivery.

The central application is independent and the API is open to the world thus any number of implementations with different technologies are possible.

1.4 BACKGROUND STUDY

There have been other grocery category eCommerce initiatives and businesses in Bangladesh such as chaldal.com. They work as the bridge between consumer and producer. They earn their commission from both the farmer and the customer. It makes the market more expensive but the problem remains such as the farmers are still underpaid and the customer is paying more than the government-determined price. In the KachaBazar there is no middleman. The KachaBazar itself is the bridge and it connects the consumer and the farmer with the delivery man. Here the consumer has to pay only the additional delivery fee and the rest goes to the farmer thus the farmer is not underpaid.

The next problem comes with frozen food. Most of the fragile foods are frozen food because they are short-lived and inconvenient to carry and store because of their very nature. In KachaBazar the consumer can order a fish that hasn't been caught yet. It means the consumer can get a fresh product right out of the production field.

The next advantage of the KachaBazar is the ability of prior estimation. Here the farmer doesn't have to fish everything in the pond and go to the market. Since the farmer already knows the order, he only has to produce the ordered number of products and bring them to the delivery man. It solves one of the major economic problems of estimation.

1.5 PROJECT PLANNING

Project planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. The logical dependencies

between tasks are defined using an activity network diagram that enables the identification of the critical path. Float or slack time in the schedule can be calculated using project management software. Then the necessary resources can be estimated and costs for each activity can be allocated to each resource, giving the total project cost. At this stage, the project plan may be optimized to achieve the appropriate balance between resource usage and project duration to comply with the project objectives. Once established and agreed upon, the plan becomes what is known as the baseline. Progress will be measured against the baseline throughout the life of the project

1.5.1 Software Development Lifecycle Presets

Software Development Life Cycle or the SDLC is a methodology or system with clearly defined processes and tasks for creating high-quality software. In detail, the methods of SDLC focuses on the following phases of software development:

- Requirement analysis
- Planning
- Software design such as architectural design
- Software development
- Testing
- Deployment

A good SDLC followed architecture can achieve the highest level of documentation and management control. All the questions of the developers like what they should build and why gets easily understandable. All stakeholders agree on the goal in an upfront manner and decide a clear plan for arriving at that goal. Everyone understands the costs and resources required.

There are several models developed over time such as:

1. WaterFall
2. Iterative
3. **Spiral**
4. V-model
5. Big Bang
6. Agile

Each of them has some specialties. Since we are going to use the **Spiral SDLC model**, only it will be introduced.

The Spiral Model:

The spiral model is a software development process model that is risk-driven. Depending upon the unique risk patterns of a specific project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping.

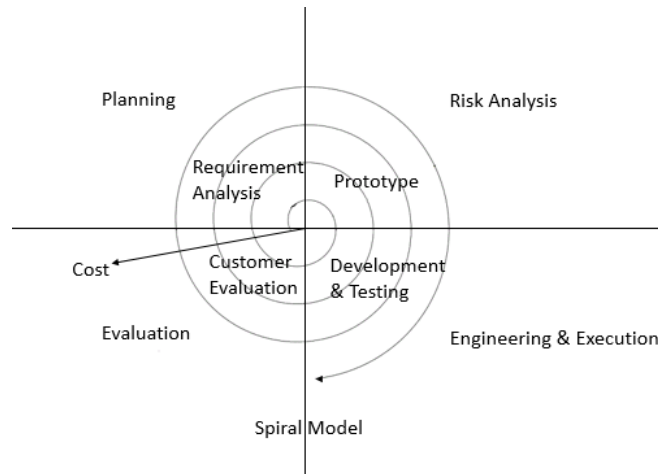


FIG 1 – Spiral Model

We use this model when:

- A Spiral model in software engineering is used when the project is large
- When releases are required to be frequent, the spiral methodology is used
- When changes can occur at any time
- When risk and costs evaluation is important
- Spiral methodology is useful for medium to high-risk projects
- When prototype creation is applicable
- When requirements are unclear and may be complex, the Spiral model in SDLC is useful
- When long-term project commitment is not feasible due to changes in economic priorities.

Chapter 2 - SYSTEM DESIGN

To build a quality design for the system, we followed the Spiral model. Our main objectives were to create the software with the best practices and we decided to go with the spiral model after considering these factors:

1. We need to take clients' feedback into consideration
2. We need to assess the risks
3. We need to launch new services
4. The has to be a flexible and adaptive new type of product that may arise
5. We need to satisfy the clients

2.1 PLANNING

This is the first phase; A major part of planning is requirement analysis. It started with a set of questionnaires and some actors acted as different stakeholders. This gave us an idea of the requirements. After analyzing them, we divided them into two categories, functional and non-functional. Then again, we divided it into three more categories, normal, expected, and exciting. It puts a priority tag with the requirement. Here are a few examples where normal is denoted as 1, expected as 2, exciting as 3.

- Functional
 - Admin – Sign In; [1]
 - Admin – Managing Category; [1] Here 'managing' is adding, editing, deleting
 - Admin – Managing Sub-Category; [1]
 - Admin – Managing Product; [1]
 - Admin – Managing Category; [1]
 - Admin – Managing Customer; [1]
 - Admin – Managing Delivery Man; [1]
 - Admin – Managing Sellers; [1]
 - Admin – Managing Sub-Admins [1];
 - Admin – Managing Areas [1];
 - Admin – Managing Orders [1];
 - Sub-Admin – Report Generation; [2]
 - Sub-Admin – Access Control; [1]
 - Sub-Admin – Partial Admin Role; [1]
 - Delivery Man – Order Processing; [1]
 - Seller – Product Managing; [1]
 - Seller – Order Managing; [1]
 - Customer – Product Viewing; [1]
 - Customer – Ordering; [1]
 - Customer – Searching; [1]
 - Customer – Profile; [1]
 - Customer – Cart; [1]
 - Payment Gateway Integration; [1]
 - Android Apps Setup; [SDK, NDK, and other dev tools]; [2]
 - Restful API construction; [1]
 - Asynchronous HTTP Request Handling; [1]
 - Android – Payment Gateway integration; [2]
 - Android – Cart; [1]

- Android Picture rescale, resize, and compression; [2]
- Server Setting; [1]
- UI/UX Designing; [2]
- Feedback; [2]
- Support; [2]
- Privacy Policy; [2]
- Contact Methods; [2]
- Recommended Product; [3]
- Non-Functional
 - Security – HTTPS [1]
 - Reliability [1]
 - Fast [1]
 - UI UX [2]
 - Support for all browsers [1]
 - Low Downtime [1]

This is a little snapshot of our requirements. After analyzing every requirement we started doing system requirement specification (SRS) that helps understand the requirements better. It all starts with the scenario. The scenario is what a client of user wants to do when he is inside the system. There are four main types of models and some submodels,

1. Scenario Based Model
 - a. Scene
 - b. Use Case
 - c. Use Case Diagram
 - d. Activity Diagram
 - e. Swimlane Diagram
2. Flow Oriented Model
 - a. Data Flow Model – level 0
 - b. Data Flow Model – level 1
 - c. Data Flow Model – level 3
3. Behavioral Model
 - a. State Diagram
 - b. Sequence Diagram
4. Class-Based Model
 - a. Class Detection
 - b. Class Evaluation
 - c. Class Declaration
 - d. Class Responsibility Collaboration (CRC)

Here is a sample is given for every model mentioned above.

2.2 SCENARIO BASED MODEL

2.2.1 Scenario 1:

A happy customer opens the browser and types KachaBazars and opens our website. By clicking on the featured and other products he viewed his preferred product. Then he searched for a specific product with the category type and name. After adding the quantity, he then adds the product into the cart and proceeds to the checkout page. He then pays with his favorite payment method and gets a success page. Now he can see the order status on his profile and observe as the KachaBazar authority will handle the rest.

2.2.2 Scenario 2:

A quality seller is tense about his business as the market becomes competitive and he gets lower prices every day. Then he gets himself registered as a KachaBazar seller and started uploading pictures and other descriptions of his product. Then he waits for the order to come. Suddenly an order came and he prepared the goods and hand it over to the delivery man and receives the payment.

2.2.3 Scenario 3:

A busy deliveryman is checking his Delivery app for KachaBazar to check for new orders and started moving as he gets the closest one. He then took the goods from the seller and deliver them to the customer and get a tip for his good work. By changing the order status, he confirms that the delivery was successful.

2.2.4 Scenario 4:

An admin logs into the KachaBazar admin panel and creates some new categories as well as some new subcategories. Then he adds some products of the added category with all the information about the product. Then he moves on to check the report generated by the sub-admins and do some accounting jobs. He checks for the unallocated order or the new orders and finds a suitable deliveryman who is not too busy or too far from the given location. Then moves onto the seller control tab and adds some new sellers who want to serve the KachaBazar as a seller. The same goes for the deliveryman. He adds the new districts and unions that were recently declared by the government.

2.2.5 Use Cases

Use case no	1
Name	Customer Login & Registration
Primary Actor	Client
Secondary Actor	Database, Rest API, Async HTTP Request
Goal in context	To get access to the website OR app to view or buy some product.
Precondition	Internet, Brower, Email, Phone, 18 years old
Trigger	When someone wants to create an ID or use our service
Events	After clicking the buttons and interactions with the database. In the android app, a restful request was made to the server.
Priority	High
When Available	4th increment
Frequency of Use	Medium

Channel to actor	Through browser OR App
Channel to secondary actor	API / DB connection Class
Scenario	At the beginning
Exception	<ul style="list-style-type: none"> • Client enters a wrong email • Server timeout because of weak network • Clients' browser doesn't support modern JS • Client puts on excessively large text as name • Client is less than 18 years old

Table 2.1 – Use case 1 Login & Registration

This use-case controls all the operations of login and registration and does session management

Use case no	2
Name	View Product
Primary Actor	Client
Secondary Actor	System
Goal in context	Show the full description of a product according to the client's selection
Precondition	No precondition
Trigger	Mouse Click on the website OR touch on the app
Events	Then click or touch is recorded and sent to the server using API and then the servlets render the info from the database and respond with the JSON file that contains all the info.
Priority	High
When Available	4th increment
Frequency of Use	High
Channel to actor	Rest API
Channel to secondary actor	Servlet
Scenario	The client wants to view the description of a product
Exception	<ul style="list-style-type: none"> • Client is trying to click on a product with internet connection off. • The internet connection is too slow • Server is in maintenance

Table 2.2 – Use case 2 – Viewing Product

This use-case serves the product information as a JSON file. Firstly, it was rendered as a Java object and then converted into JSON for cross-platform compatibility.

Use case no	3
Name	Search Product
Primary Actor	Client
Secondary Actor	System
Goal in context	Show the list of products according to the client's search query
Precondition	Name and Category has to be selected
Trigger	Click on the Search tab and search for the product name

Events	The query is taken to the preferred category and then the product is fetched if found and returns null otherwise
Priority	High
When Available	5th increment
Frequency of Use	Moderate-High
Channel to actor	Rest API
Channel to secondary actor	Servlet
Scenario	The client wants to search for a product
Exception	<ul style="list-style-type: none"> • Entering unrelated info • The internet connection is too slow • Server is in maintenance

Table 2.3 – Use case 3 – Search Product

This use-case serves the products that are searched for as a JSON Array file. Firstly, it was rendered as a list of Java objects and then converted into JSON Array for cross-platform compatibility

Use case no	4
Name	Add to Cart
Primary Actor	Client
Secondary Actor	Product or Seller Product
Goal in context	Shopping multiple products at the same time
Precondition	Have to be a registered customer
Trigger	Click on add to cart button on the product view page
Events	The product ID is taken and the quantity is also taken, then the information is added to the cart of the client.
Priority	High
When Available	5th increment
Frequency of Use	Moderate
Channel to actor	Rest API
Channel to secondary actor	Servlet
Scenario	The client wants to add a product to the cart
Exception	<ul style="list-style-type: none"> • Entering quantity that is larger than the stock

Table 2.4 – Use case 4 – Adding product to cart

This use-case shows the factors related to adding the product to the cart

Use case no	5
Name	Checkout
Primary Actor	Client
Secondary Actor	Product or Cart
Goal in context	The client needs to order so he has to pay for the selected product(s).
Precondition	Have to have a complete profile with all the necessary information like address, phone number, etc.
Trigger	Click on the checkout button on the product page or cart page

Events	The product or products that are ordered will be calculated and additional information like the expected delivery date has to be filled in to proceed with the ordering.
Priority	High
When Available	6th increment
Frequency of Use	Moderate
Channel to actor	Rest API
Channel to secondary actor	Servlet
Scenario	The client wants to check out the product and make the payment.
Exception	<ul style="list-style-type: none"> The products are already out of stock

Table 2.5 – Use case 5 – Checkout

This use-case helps to check out both individual and products that are already in the cart

Use case no	6
Name	Cart Page
Primary Actor	Client
Secondary Actor	Cart
Goal in context	The client wants to see what is inside the cart and perform operations on them
Precondition	Has to be registered and it would be good if there are some products already added to the cart
Trigger	Click on the cart icon or button
Events	The cart products are represented and the total cost is calculated for better understanding
Priority	High
When Available	6th increment
Frequency of Use	Moderate
Channel to actor	Rest API
Channel to secondary actor	Servlet
Scenario	The client wants to see the cart and see the amount he has to pay for all the products that are inside that cart.
Exception	<ul style="list-style-type: none"> The cart is empty

Table 2.6 – Use case 6 – Viewing the cart

This use-case serves all the products that were previously added to the cart. The client can additionally clear the cart or perform other operations on the cart.

Use case no	7
Name	Payment Gateway
Primary Actor	Client
Secondary Actor	SSL Commerz, System, database
Goal in context	The client needs to pay online
Precondition	The client must have some kind of banking access like mobile banking (Bkash, Nagad, etc.) or Internet Banking (Prime Bank, Bank Asia, etc.), or card facility (Visa, master, etc.). Then he has to have enough balance to perform the payment

Trigger	Clicking on the make payment button
Events	The client is taken to the SSL Commerz interface to select the payment option and when the payment is complete then the order is finally placed.
Priority	High
When Available	7th increment
Frequency of Use	Low
Channel to actor	Rest API, SSL Commerz API
Channel to secondary actor	Servlet, SSL Commerz Class
Scenario	The client has added the items to the cart and wants to make the payment online OR the client wants to purchase a single product and make the payment online.
Exception	<ul style="list-style-type: none"> • The client doesn't have access to any kind of banking methods • The balance is lower than the price • Connection was cut out in the middle • OTP mismatch

Table 2.7 – Use case 7 – Payment Gateway

This use-case serves all the products that were previously added to the cart. The client can additionally clear the cart or perform other operations on the cart.

Use case no	8
Name	Profile
Primary Actor	Client
Secondary Actor	System
Goal in context	The client wants to check his profile and perform operations
Precondition	The client has to log in to the website or the app to get all the info.
Trigger	Clicking on the avatar or the profile section in the app
Events	The API serves the information about the client and then fields are populated with them.
Priority	Moderate
When Available	4th increment
Frequency of Use	Low
Channel to actor	Rest API
Channel to secondary actor	Servlets
Scenario	The client wants to see his/her full information
Exception	<ul style="list-style-type: none"> • The client doesn't have all the info • The client is not logged in.

Table 2.8 – Use case 8 – Profile

This use-case server all the information and allow the client to perform different operations on them

Use case no	9
Name	Seller Login
Primary Actor	Seller
Secondary Actor	System

Goal in context	The Seller wants to log in to the seller portal or seller section on the app
Precondition	The seller has to have an account and credential beforehand
Trigger	Clicking on the Sell menu on the app
Events	The seller will be confirmed with the Phone and Password
Priority	Moderate
When Available	4th increment
Frequency of Use	Moderate
Channel to actor	Rest API
Channel to secondary actor	Servlets
Scenario	The Seller wants to log in and sell his/her products
Exception	<ul style="list-style-type: none"> The seller is not registered yet

Table 2.9 – Use case 9 – Seller login

This use-case helps the seller to login to his portal where he can see the products he uploaded and also upload new products

Use case no	10
Name	Adding Seller Product
Primary Actor	Seller
Secondary Actor	System
Goal in context	The Seller wants to upload new products
Precondition	The seller has to have an account and credential. And he has to be logged in.
Trigger	Clicking on the add product button
Events	The seller will be asked for information about the product such as image, price, category, etc.
Priority	High
When Available	4th increment
Frequency of Use	Moderate
Channel to actor	Rest API
Channel to secondary actor	Servlets
Scenario	The Seller wants to add new products for sale
Exception	<ul style="list-style-type: none"> The seller is not logged in Some information is missing

Table 2.10 – Use case 10 – Add a new product by the seller

This use-case helps the seller to upload a new product for sale

Use case no	11
Name	Deliveryman Login
Primary Actor	Deliveryman
Secondary Actor	System
Goal in context	The Deliveryman wants to log in to the Deliveryman portal or Deliveryman App
Precondition	The Deliveryman has to have an account and credential beforehand
Trigger	Opening the Deliveryman App

Events	The Deliveryman will be confirmed with the Phone and Password
Priority	Moderate
When Available	7th increment
Frequency of Use	Moderate
Channel to actor	Rest API
Channel to secondary actor	Servlets
Scenario	The Deliveryman wants to log in and deliver his/her orders
Exception	<ul style="list-style-type: none"> The Deliveryman is not registered yet

Table 2.11 – Use case 11 – Deliveryman login

This use-case helps the **Deliveryman** to login to his portal where he can see the orders that he was assigned for both inventory and seller products

Use case no	11
Name	Deliveryman Status Changing
Primary Actor	Deliveryman
Secondary Actor	System
Goal in context	The Deliveryman wants to change the status of an order
Precondition	The Deliveryman has to have an account and credential beforehand
Trigger	Opening the Deliveryman App and logging in with the credentials
Events	The Deliveryman will deliver the product and update the order status according to the payment status
Priority	Moderate
When Available	8th increment
Frequency of Use	Moderate
Channel to actor	Rest API
Channel to secondary actor	Servlets
Scenario	The Deliveryman wants to update the order status
Exception	<ul style="list-style-type: none"> The Deliveryman was not assigned any order

Table 2.11 – Use case 11 – Deliveryman Status Changing

This use-case allows the deliveryman to change the status of an order according to the payment of the order.

Use case no	12
Name	Admin Login
Primary Actor	Admin
Secondary Actor	System
Goal in context	The admin wants to log in to the Admin portal
Precondition	The admin has to have an account and credential beforehand
Trigger	Opening the admin portal
Events	The admin will be confirmed with the Phone and Password
Priority	Moderate

When Available	1st increment
Frequency of Use	Moderate
Channel to actor	Rest API
Channel to secondary actor	Servlets
Scenario	The admin wants to log in and control everything
Exception	<ul style="list-style-type: none"> The admin is not registered yet

Table 2.12 – Use case 12 – admin login

This use-case helps the admin to login into his portal where he can see and control everything.

Use case no	13
Name	Admin – Category Control
Primary Actor	Admin
Secondary Actor	System
Goal in context	The admin wants to manage the categories
Precondition	Admin has to be logged in
Trigger	Opening the category menu
Events	After the admin clicks the category menu, he can perform several operations on the category
Priority	Moderate
When Available	1st increment
Frequency of Use	Moderate
Channel to actor	Rest API
Channel to secondary actor	Servlets
Scenario	The admin creates, updates categories
Exception	<ul style="list-style-type: none"> The admin is not logged in Missing picture of a category

Table 2.13 – Use case 13 – admin – category control

This use-case manages all the categories

There are many similar use cases for the admin and thus they are skipped to make the document precise.

2.2.7 Activity Diagram

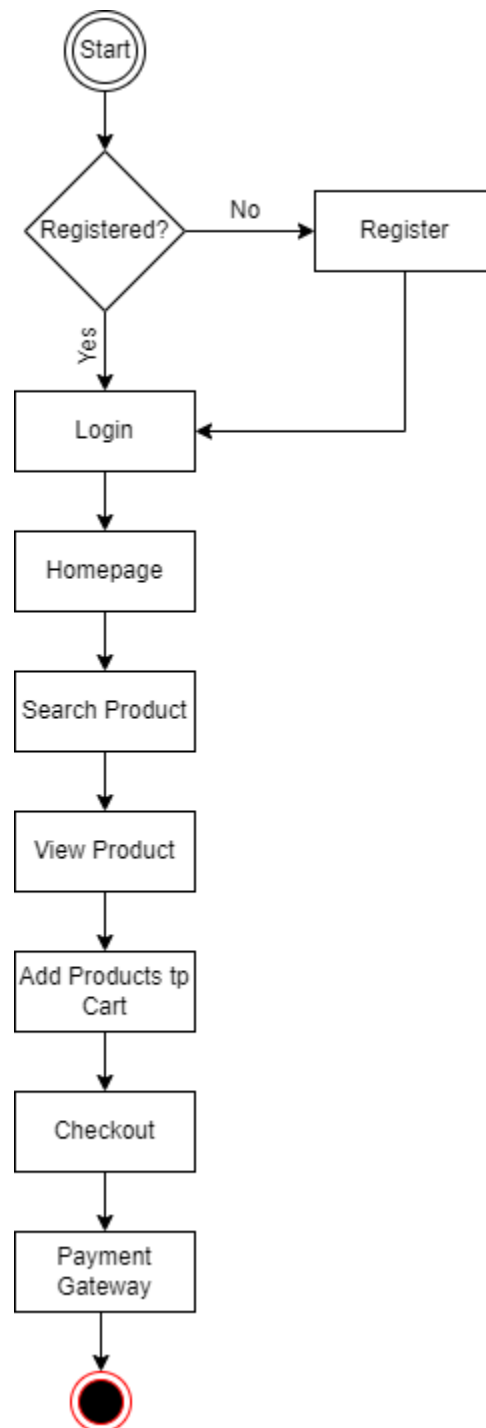


Fig 3 – Activity Diagram for Customer

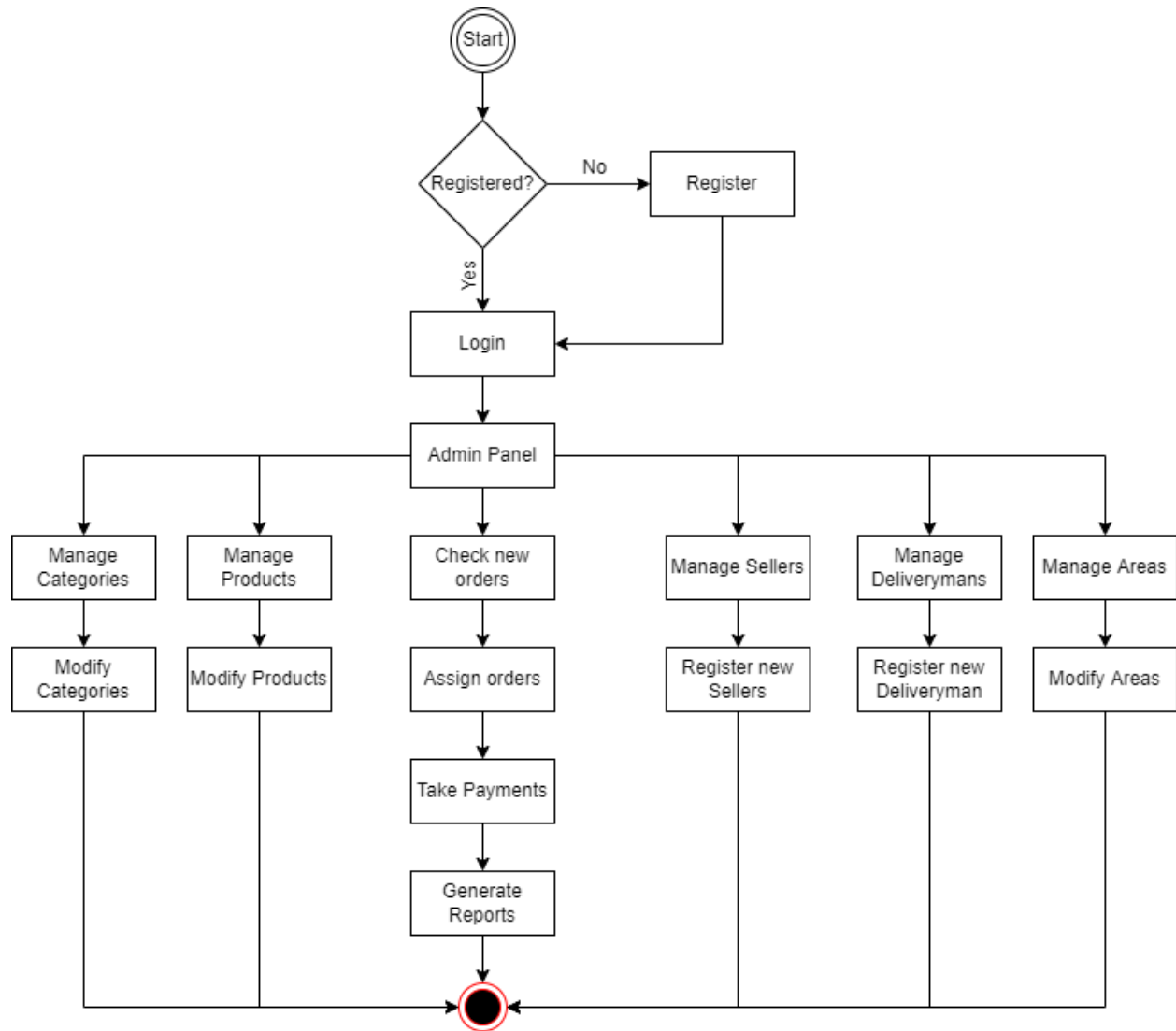


Fig 3 – Activity Diagram for Admin

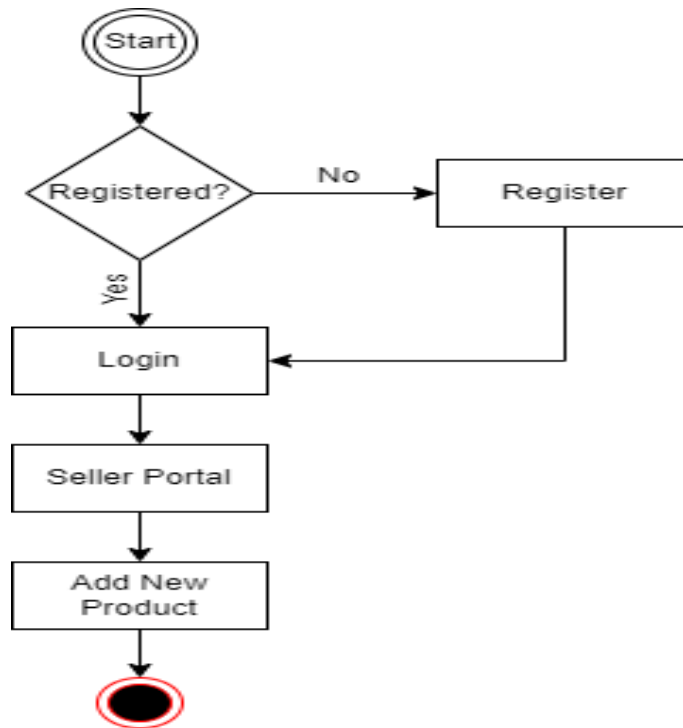


Fig 4 – Activity Diagram for Sellers

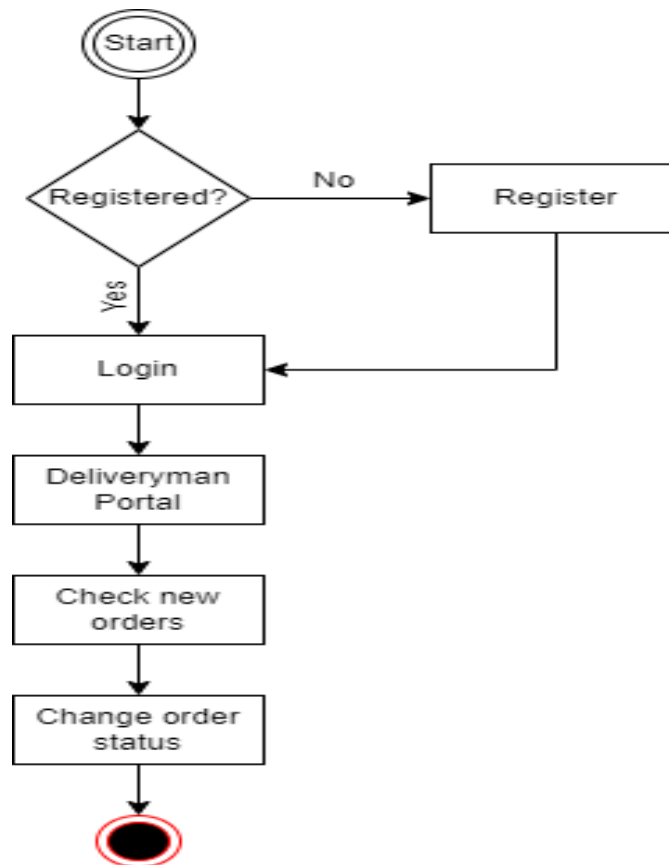


Fig 5 – Activity Diagram for Deliveryman

2.2.8 Swimlane Diagram

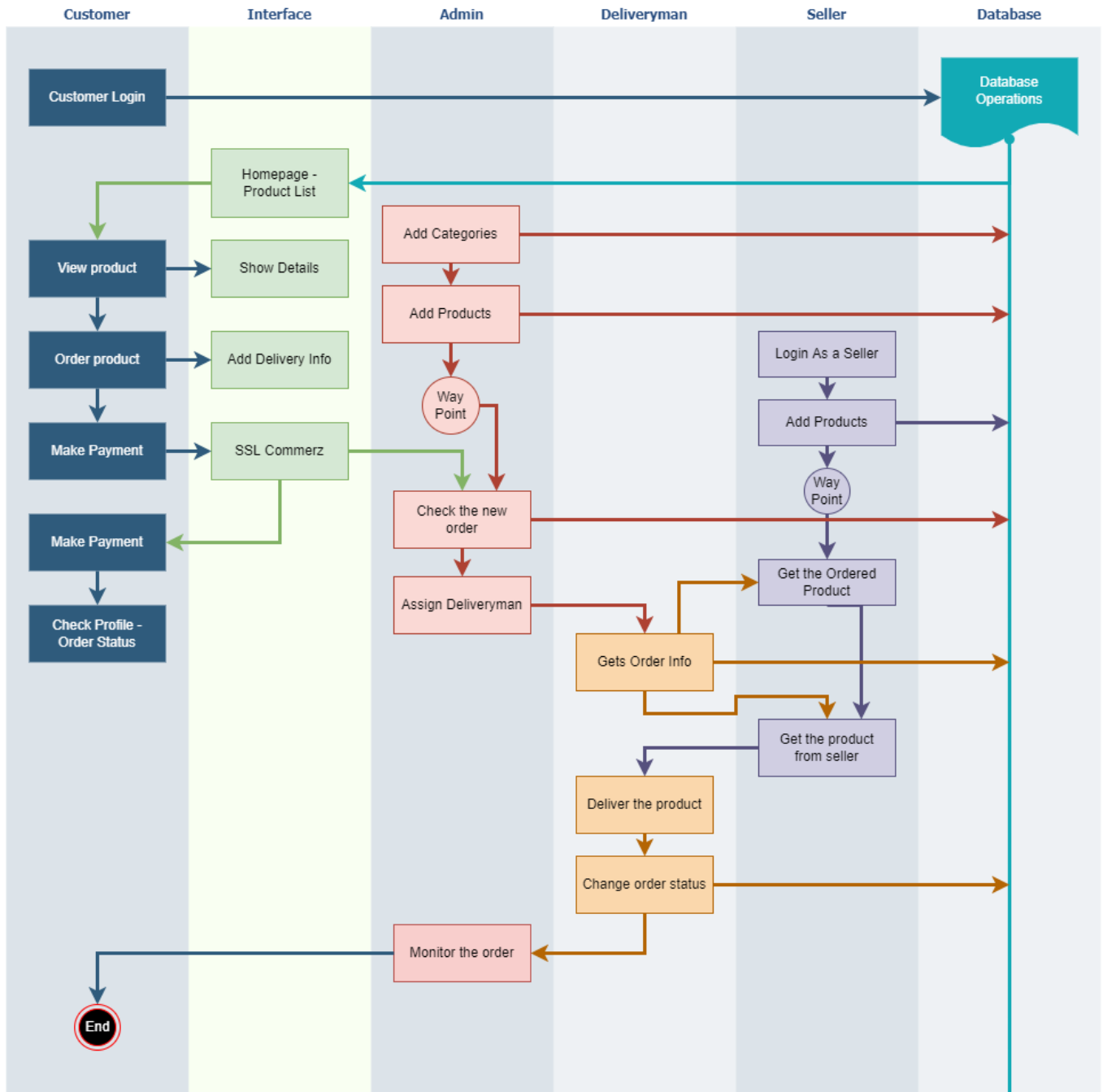


Fig 6 – Swimlane Diagram

2.3 FLOW ORIENTED MODEL

2.3.1 DFD – Level 0

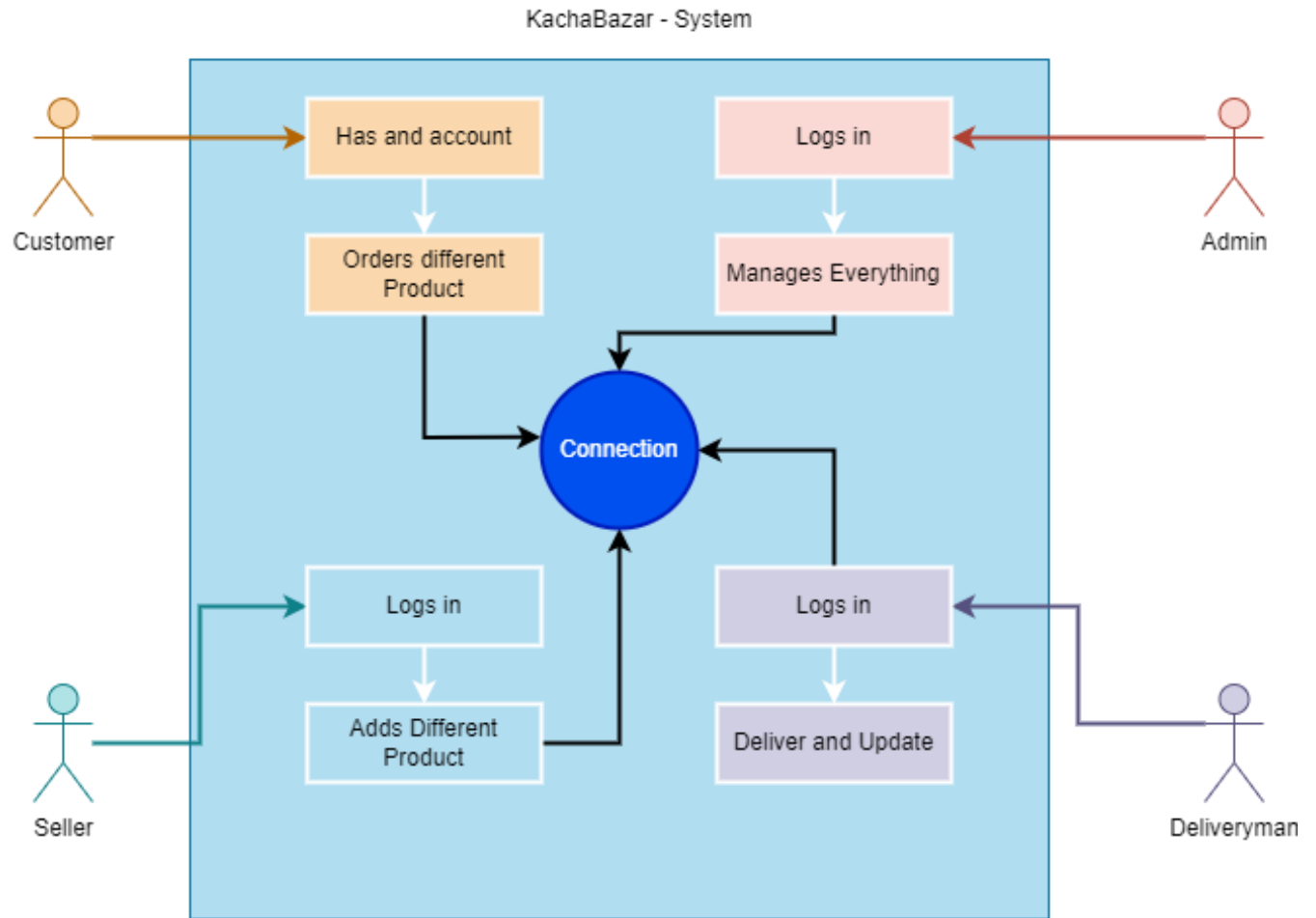


Fig 7 – DFD Level 0

The DFD 0 shows a higher-level architecture which is also an abstract view of the system and all its major parts or components.

2.4 BEHAVIORAL MODEL

2.4.1 State Diagram

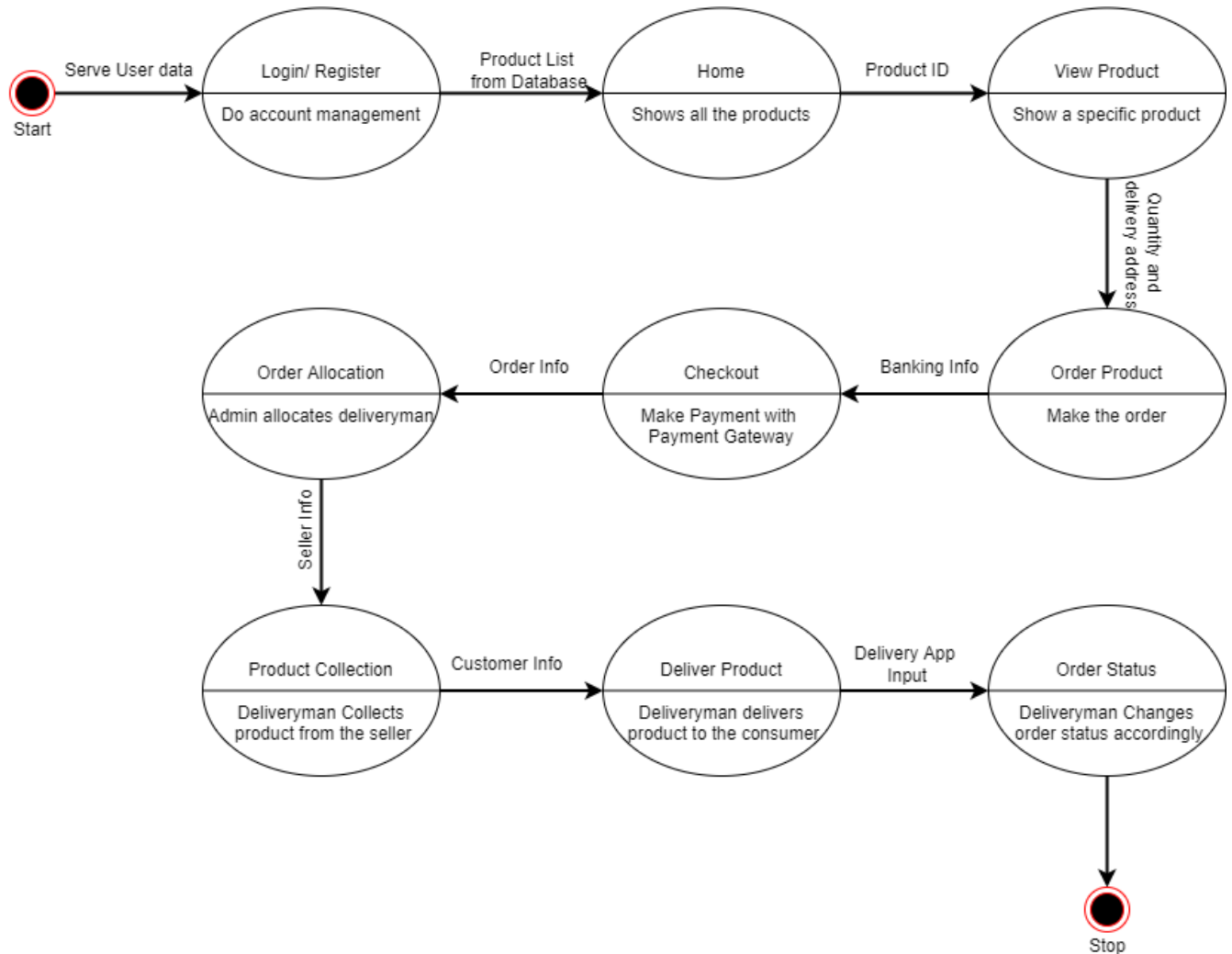


Fig 9 – State Diagram

This diagram shows the state of the procedure as well as jumping from module to module by taking necessary input

2.4.2 Sequence Diagram

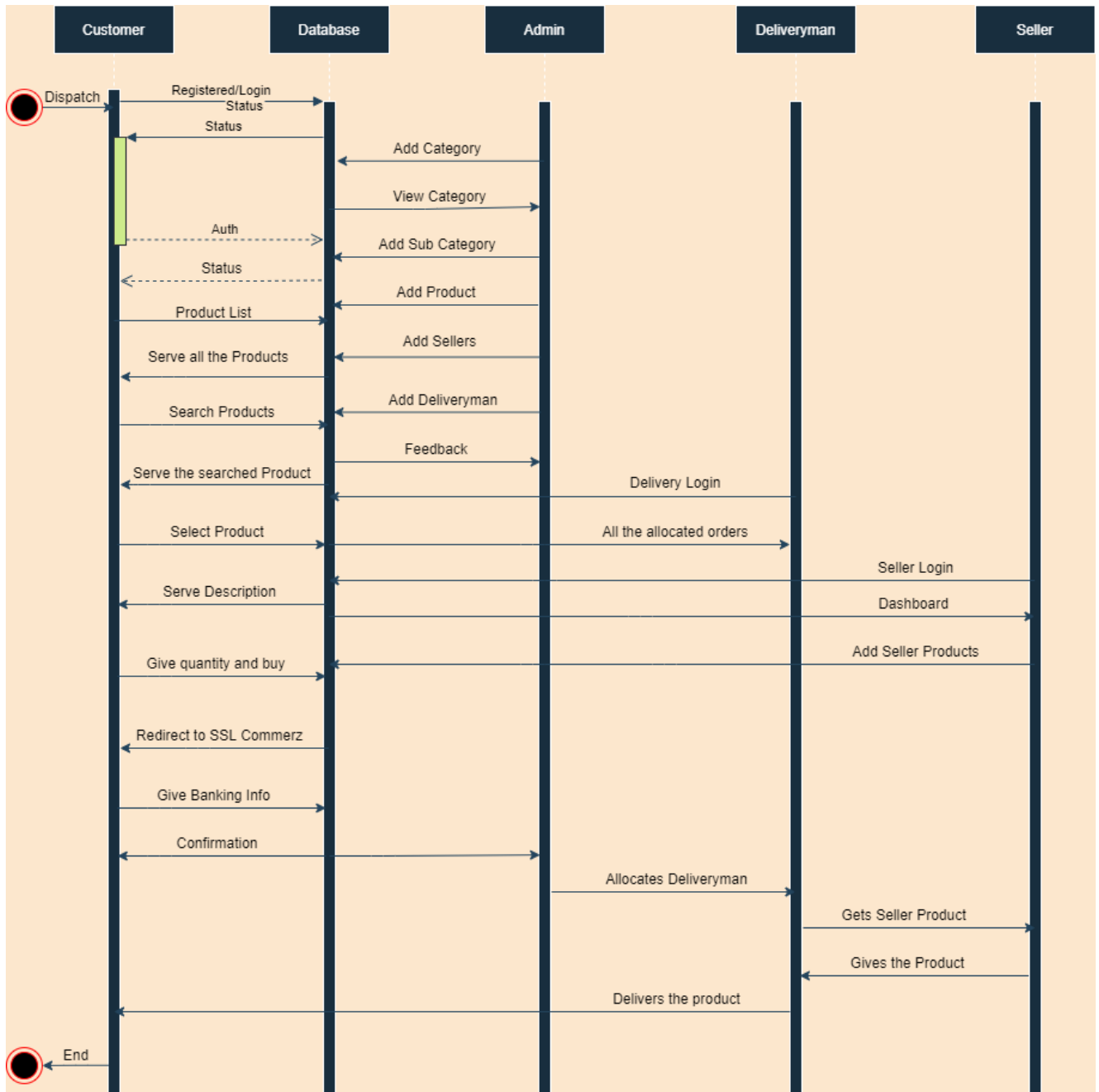


Fig 10 – Sequence Diagram

This shows the workflow of the program with conditional statements

2.5 CLASS-BASED MODEL

Here, we will mark all the nouns and verbs where every noun has the potential to become a class and every verb has the potential to become a function. There is mainly 7 category a class can manifest into. Those are –

1. External entities
2. Things
3. Occurrences Or Events
4. Roles
5. Organizational Units
6. Places
7. Structures

Now, let's mark the nouns and verbs.

2.5.1 Scenario analysis

Scenario 1:

A happy customer opens the browser and types KachaBazars and opens our website. By clicking on the featured and other products he viewed his preferred product. Then he searched for a specific product with the category type and name. After adding the quantity, he then adds the product into the cart and proceeds to the checkout page. He then pays with his favorite payment method and gets a success page. Now he can see the order status on his profile and observe as the KachaBazar authority will handle the rest.

Scenario 2:

A quality seller is tense about his business as the market becomes competitive and he gets lower prices every day. Then he gets himself registered as a KachaBazar seller and started uploading pictures and other descriptions of his product. Then he waits for the order to come. Suddenly an order came and he prepared the goods and hand it over to the delivery man and receives the payment.

Scenario 3:

A busy deliveryman is checking his Delivery app for KachaBazar to check for new orders and started moving as he gets the closest one. He then took the goods from the seller and deliver them to the customer and get a tip for his good work. By changing the order status, he confirms that the delivery was successful.

Scenario 4:

An admin logs into the KachaBazar admin panel and creates some new categories as well as some new subcategories. Then he adds some products of the added category with all the information about the product. Then he moves on to check the report generated by the sub-admins and do some accounting jobs. He checks for the unallocated order or the new orders and finds a suitable deliveryman who is not too busy or too far from the given location. Then moves onto the seller control tab and adds some new sellers who want to serve the KachaBazar as a seller. The same goes for the deliveryman. He adds the new districts and unions that were recently declared by the government.

We found 38 nouns/ potential classes. Now the final classes will be

No.	Name	Satisfies No.	Status
1	Customer	1, 2, 4, 5, 7	Accepted
2	Feature	2, 4, 7	Rejected
3	Product	2, 5	Accepted
4	Search	2, 5	Accepted
5	Category	2, 3, 5	Accepted
6	Sub Category	2, 3, 5	Accepted
7	Name	2, 3, 5	Rejected
8	Quantity	2, 3, 5	Rejected
9	Cart	2, 5	Accepted
10	Checkout	2, 3	Accepted
11	Payment Method	1, 2	Accepted
12	Success	2, 5	Accepted
13	Order	2, 3, 7	Accepted
14	Profile	2, 3, 7	Accepted
15	KachaBazar	2	Accepted
16	Admin (Authority)	1, 2, 4, 5	Accepted
17	Seller	1, 2, 4, 5	Accepted
18	Business	1, 2, 4, 5	Rejected
19	Market	1, 2, 4, 7	Rejected
20	Prices	1, 2, 4, 5, 7	Rejected
21	Seller registration	2, 3, 5, 7	Accepted
22	Seller Product	2, 3	Accepted
23	Seller Product Picture	2, 3, 5, 7	Rejected
24	Seller Product Info	2, 3, 5, 7	Accepted
25	Seller Order	2, 3, 4, 5, 7	Accepted
26	Delivery Man	1, 2, 4, 5, 7	Accepted
27	Delivery App	1	Accepted
28	Order Confirmation	3, 7	Accepted
29	Report	3, 7	Accepted
30	Division	1, 7	Accepted
31	District	1, 7	Accepted
32	Upazilla	1, 7	Accepted
33	Union	1, 7	Accepted
34	Connection	3, 7	Accepted
35	SSL Commerz	1, 2	Accepted
36	Parameter Builder	3	Accepted
37	Request Initiator	3	Accepted
38	Unit	5, 7	Accepted

Table – Class Recognition Table

2.5.2 Class Declaration Table

1. System	
Application ID Application Version Application metaData	StartTheKachaBazar() //maintenance mode or running ClearGC()
Class – 1	
2. Customer	
ID Name Email Phone Picture: TYPE byte[] Password Division: TYPE Class 'Division' District: TYPE Class 'District' Upazilla: TYPE Class 'Upazilla' Union: TYPE Class 'Union' Permissions Role	Init() : self Registration() Login() Logout() getProfile() Cart(ID) //checks the cart SSLCommerz() //For Paying
Class – 2	
3. Seller	
ID Name Email Phone NID Picture: TYPE byte[] Password Division: TYPE Class 'Division' District: TYPE Class 'District' Upazilla: TYPE Class 'Upazilla' Union: TYPE Class 'Union' Permissions Role	Init() : self Login() Logout() getProfile() uploadProduct(ID) //Adding Product uploadPicture () //Image resizing and compression
Class – 3	
4. Deliveryman	
ID Name Email Phone NID Picture: TYPE byte[] Password Division: TYPE Class 'Division' District: TYPE Class 'District' Upazilla: TYPE Class 'Upazilla' Union: TYPE Class 'Union' Permissions Role	Init() : self Login() Logout() getProfile() getOrders(ID) //get assigned order markOrderAsComplete () //changing order status
Class – 4	
4. Admin	

ID Name Email Phone NID Picture: TYPE byte[] Password Permissions Role	Init() : self logIn() logOut() category() subCategory() product() seller() customer() deliveryman() area() inventoryOrders() sellerOrders() generateReport() subAdmin()
---	--

Class – 4

5. Sub-Admin	
ID Name Email Phone NID Picture: TYPE byte[] Password Permissions Role	Init() : self logIn() logOut() product() seller() customer() deliveryman() area() inventoryOrders() sellerOrders() generateReport()

Class – 5

6. Category	
ID Name Picture: TYPE byte[] Description	Init() : self addCategory() deleteCategory(ID) updateCategory() viewCategory()

Class – 6

6. Sub-Category	
ID Name Description Government Price	Init() : self addSubCategory() deleteSubCategory(ID) updateSubCategory() viewSubCategory()

Class – 6

All the accepted classes are declared with appropriate constructors and variables. Those are skipped as the real implementation is too vague to present in one document.

2.5.3 Class Responsibility Collaboration (CRC)

The CRC shows the closely connected classes and their collaboration to fill one or more responsibilities.

Class: Customer	
Responsibility	Collaborator
1. Registration 2. Login 3. Save cart 4. Save all info 5. Record order history	Connection Cart Order Product

CRC – 1

Class: Category	
Responsibility	Collaborator
1. Add category 2. Update category 3. Delete category 4. View category 5. Picture Storing	Connection

CRC – 2

Class: Sub-Category	
Responsibility	Collaborator
1. Add sub-category 2. Update sub-category 3. Delete sub-category 4. View sub-category 5. Store government price	Connection Category

CRC – 3

Class: Product	
Responsibility	Collaborator
1. Add Product 2. Update Product 3. Delete Product 4. View Product 5. Product Type 6. Product Unit	Connection Category Subcategory Unit

CRC – 4

Class: Order	
Responsibility	Collaborator
1. Order a product 2. Saves a record 3. Payment 4. Notify the admins 5. Necessary info for the order 6. Customize and randomize the transaction ID 7. Modify orders 8. Assign Deliveryman 9. Set expected date 10. Get quantity	Connection Cart Customer Product SSLCommerz

11. Subtract quantity from the stock	
--------------------------------------	--

CRC – 5

Class: Order Seller Product	
Responsibility	Collaborator
7. Order a Seller product	Connection
8. Saves a record	Cart
9. Payment	Customer
10. Notify the admins	Seller Product
11. Necessary info for the order	SSLCommerz
12. Customize and randomize the transaction ID	
13. Modify orders	
14. Assign Deliveryman	
15. Set expected date	
16. Get quantity	
17. Subtract quantity from the stock	

CRC – 6

Similarly, all the responsibilities are collaborated between these classes to provide a flawless and simultaneous process synchronization. Thus, they are skipped.

There are more models like Entity-Relationship Diagram (ERD), Dependency diagram, etc. This model simplifies the requirements and helps stakeholders to understand the behavior and the nature of the software.

Then at the second stage risk analysis,

We estimated the time that it may take to complete requirements or modules. It helped us understand the efficiency and the capability of the team and thus we could calculate the cost for each module. The negotiation part gets a great deal of help from this analysis.

We also estimated the lines of code it may need to perform one task and since the variables, functions, and relationships were declared in the Class-based model above, it was easy to estimate a rough number.

Since our software was an eCommerce type of software, we have also performed a competitor analysis and calculated risk factors and budget for the final product. It may be hard and not realistic but it gives an idea and creates a new way. The lifetime and the probable maintenance risk were also estimated.

After taking every risk with careful consideration, we tried our best to minimize the risk. Like:

The line of codes can be reduced by using **reusable objects**. We thought about design patterns that will be discussed in the next stage of the spiral model.

Finally, we have completed step two of the project and started building a prototype.

In the next step, the development is started,

We tried as much as we can to reduce the code smell and reduce the line of codes by using different design patterns.

We have used,

- DTO design pattern
- MVC
- Singleton Pattern
- Factory Pattern
- Strategy Pattern
- Null object Pattern
- Proxy Pattern

Then in the development stage, we did unit testing and environment testing to check out the integrity of the software.

Finally, we deployed the software and get customer feedback and repeat the cycle.

Chapter 3 - TECHNOLOGY REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

3.1.1 For Website:

- Processor – Pentium IV or Above
- RAM – 2 GB or above
- Hard disk – 50 GB or above
- Input Device – Keyboard, Mouse
- Output Device – Monitor

3.1.2 For Android App:

- Processor – 1 GHz or above
- RAM – 512 MB or above
- ROM – 4GB or above
- Camera (any)

3.1.3 For Server

- Intel Xenon or AMD Epyc
- Or shared server
- RAM – 8 GB or above (16GB recommended)
- SSD – 500GB or above

3.2 SOFTWARE REQUIREMENTS

3.2.1 For Website

- Any modern browser that supports JS
- Internet connection > 32Kbps
- Stable connection
- Updated firmware

3.2.2 For Android App

- Android version – 4.2 or above
- No data restriction
- Active internet connection
- Android package installer

3.2.3 For server

- Apache
- Tomcat
- Perl
- MySQL/MariaDB
- PHP
- Eclipse
- Operating System – CentOS, Linux, Windows, Ubuntu, Redhat, Kali

3.3 TECHNOLOGIES

3.3.1 For Website

3.3.1.1 *Frontend*

- JSP
- jQuery
- Ajax
- JavaScript
- HTML
- CSS
- Bootstrap
- SVG

3.3.1.2 *Backend*

- Java
- Servlet
- Hibernate
- HQL
- MySQL drivers
- JSON
- PhpMyAdmin

3.3.1.3 *Localhost*

- Apache
- Tomcat
- MariaDB/MySQL
- PHP

3.3.2 For android application

- Java
- SDK – 24
- Asynchronous HTTP Request
- Android Studio

Chapter 4 - DATABASE AND TABLES

4.1 DATABASE DESIGN

The database is critical for all businesses. A good database does not allow any form of anomalies and stores only relevant information in an ordered manner. If a database has anomalies, it is affecting efficiency and data integrity. For example, a delete anomaly arises upon the deletion of a row which also forces other useful data to be lost. As such, the tables need to be normalized. This fulfills the last objective of ensuring data are accurate and retrieved correctly.

The organization of data in database aims to achieve three major objectives: -

- ✓ Data integration
- ✓ Data integrity
- ✓ Data independence

In this project, we didn't use a traditional or classical database design approach, since it makes the dependencies complex. Instead, we used the hibernate ORM to create and manage a database for us. Hibernate is an open-source object-relational mapping (ORM) tool that provides a framework to map object-oriented domain models to relational databases for web applications. Object-relational mapping is based on the containerization of objects and the abstraction that provides that capacity. Abstraction makes it possible to address, access, and manipulate objects without having to consider how they are related to their data sources. The Hibernate ORM framework guides mapping Java classes to database tables and Java data types to SQL data types and provides querying and retrieval.

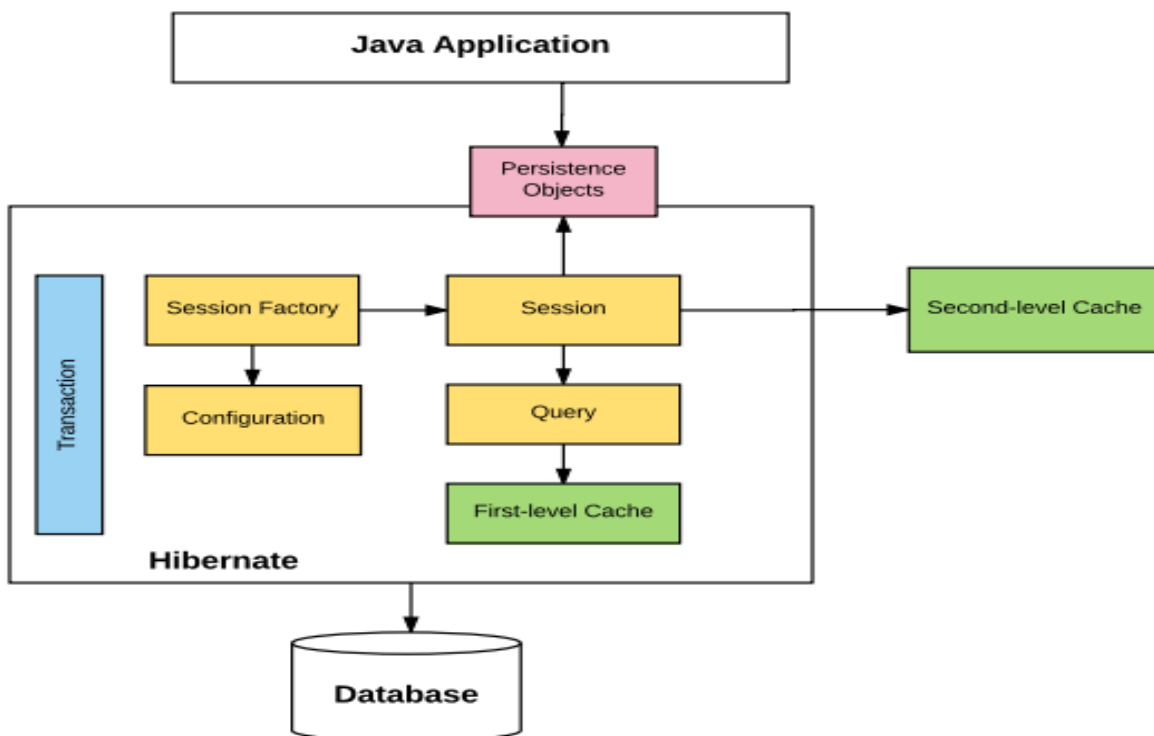


Fig 11 – Hibernate workflow

4.2 ALL TABLE LIST

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> admin	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	16 KiB	-
<input type="checkbox"/> android	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	16 KiB	-
<input type="checkbox"/> bids	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci	32 KiB	-
<input type="checkbox"/> cart	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	32 KiB	-
<input type="checkbox"/> cart_details	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	48 KiB	-
<input type="checkbox"/> category	★ Browse Structure Search Insert Empty Drop	7	InnoDB	utf8_general_ci	1.5 MiB	-
<input type="checkbox"/> customers	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	192 KiB	-
<input type="checkbox"/> delivery_person	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	560 KiB	-
<input type="checkbox"/> districts	★ Browse Structure Search Insert Empty Drop	65	InnoDB	utf8_general_ci	32 KiB	-
<input type="checkbox"/> divisions	★ Browse Structure Search Insert Empty Drop	8	InnoDB	utf8_general_ci	16 KiB	-
<input type="checkbox"/> orders	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	128 KiB	-
<input type="checkbox"/> order_seller_product	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	144 KiB	-
<input type="checkbox"/> product	★ Browse Structure Search Insert Empty Drop	16	InnoDB	utf8_general_ci	3.6 MiB	-
<input type="checkbox"/> recommendation	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci	64 KiB	-
<input type="checkbox"/> sellers	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	528 KiB	-
<input type="checkbox"/> sellers_product	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	176 KiB	-
<input type="checkbox"/> subcategory	★ Browse Structure Search Insert Empty Drop	18	InnoDB	utf8_general_ci	32 KiB	-
<input type="checkbox"/> sub_admin	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	32 KiB	-
<input type="checkbox"/> unions	★ Browse Structure Search Insert Empty Drop	2,351	InnoDB	utf8_general_ci	304 KiB	-
<input type="checkbox"/> unit	★ Browse Structure Search Insert Empty Drop	6	InnoDB	utf8_general_ci	16 KiB	-
<input type="checkbox"/> upazilas	★ Browse Structure Search Insert Empty Drop	491	InnoDB	utf8_general_ci	80 KiB	-
21 tables	Sum	2,978	InnoDB	utf8_general_ci	7.5 MiB	0 B

Fig 12 – All tables of the database

4.3 CATEGORY TABLE

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> 1	category_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/> 2	category_description	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/> 3	category_name	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary Unique Index Spatial Fulltext Distinct values
<input type="checkbox"/> 4	category_image	longblob			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Fig 13 – Category table

4.4 PRODUCT TABLE

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 product_id	int(11)			No	None	AUTO_INCREMENT	Change Drop
<input type="checkbox"/>	2 government_price	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
<input type="checkbox"/>	3 product_image	mediumblob			No	None		Change Drop
<input type="checkbox"/>	4 product_description	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
<input type="checkbox"/>	5 product_name	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
<input type="checkbox"/>	6 product_price	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
<input type="checkbox"/>	7 stock	double			Yes	NULL		Change Drop
<input type="checkbox"/>	8 product_type	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
<input type="checkbox"/>	9 productCategory_category_id	int(11)			Yes	NULL		Change Drop
<input type="checkbox"/>	10 productSubcategory_subcategory_id	int(11)			Yes	NULL		Change Drop
<input type="checkbox"/>	11 productUnit_unit_id	int(11)			Yes	NULL		Change Drop

Fig 14 – Product table

4.5 SELLER PRODUCT TABLE

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 product_id	int(11)			No	None	AUTO_INCREMENT	Change Drop
<input type="checkbox"/>	2 category_image	mediumblob			No	None		Change Drop
<input type="checkbox"/>	3 product_description	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
<input type="checkbox"/>	4 product_name	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop
<input type="checkbox"/>	5 product_price	double			Yes	NULL		Change Drop
<input type="checkbox"/>	6 product_quantity	double			Yes	NULL		Change Drop
<input type="checkbox"/>	7 categoryModel_category_id	int(11)			Yes	NULL		Change Drop
<input type="checkbox"/>	8 sellerModel_seller_id	int(11)			Yes	NULL		Change Drop
<input type="checkbox"/>	9 subcategoryModel_subcategory_id	int(11)			Yes	NULL		Change Drop
<input type="checkbox"/>	10 unitModel_unit_id	int(11)			Yes	NULL		Change Drop

Fig 15 – Seller Product table

4.6 ORDER TABLE

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 order_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary
<input type="checkbox"/>	2 care_of_contact	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
<input type="checkbox"/>	3 expected_delivery_date	date			Yes	NULL		Change Drop Primary
<input type="checkbox"/>	4 order_date	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
<input type="checkbox"/>	5 order_quantity	double			Yes	NULL		Change Drop Primary
<input type="checkbox"/>	6 order_status	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
<input type="checkbox"/>	7 order_street	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
<input type="checkbox"/>	8 order_village	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
<input type="checkbox"/>	9 order_zip_code	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
<input type="checkbox"/>	10 payment_status	bit(1)			Yes	NULL		Change Drop Primary
<input type="checkbox"/>	11 payment_type	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
<input type="checkbox"/>	12 reciever_phone_number	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
<input type="checkbox"/>	13 tran_id	varchar(255)	utf8_general_ci		Yes	NULL		Change Drop Primary
<input type="checkbox"/>	14 customerModel_customer_id	int(11)			Yes	NULL		Change Drop Primary
<input type="checkbox"/>	15 deliveryPersonModel_delivery_person_id	int(11)			Yes	NULL		Change Drop Primary
<input type="checkbox"/>	16 districtModel_id	int(11)			Yes	NULL		Change Drop Primary
<input type="checkbox"/>	17 divisionModel_id	int(11)			Yes	NULL		Change Drop Primary
<input type="checkbox"/>	18 productModel_product_id	int(11)			Yes	NULL		Change Drop Primary
<input type="checkbox"/>	19 unionModel_id	int(11)			Yes	NULL		Change Drop Primary
<input type="checkbox"/>	20 upazillaModel_id	int(11)			Yes	NULL		Change Drop Primary

Fig 16 – Order Table

The rest of the Tables are skipped to make the document shorter.

Chapter 5 - PROJECT VIEW

5.1 HOMEPAGE

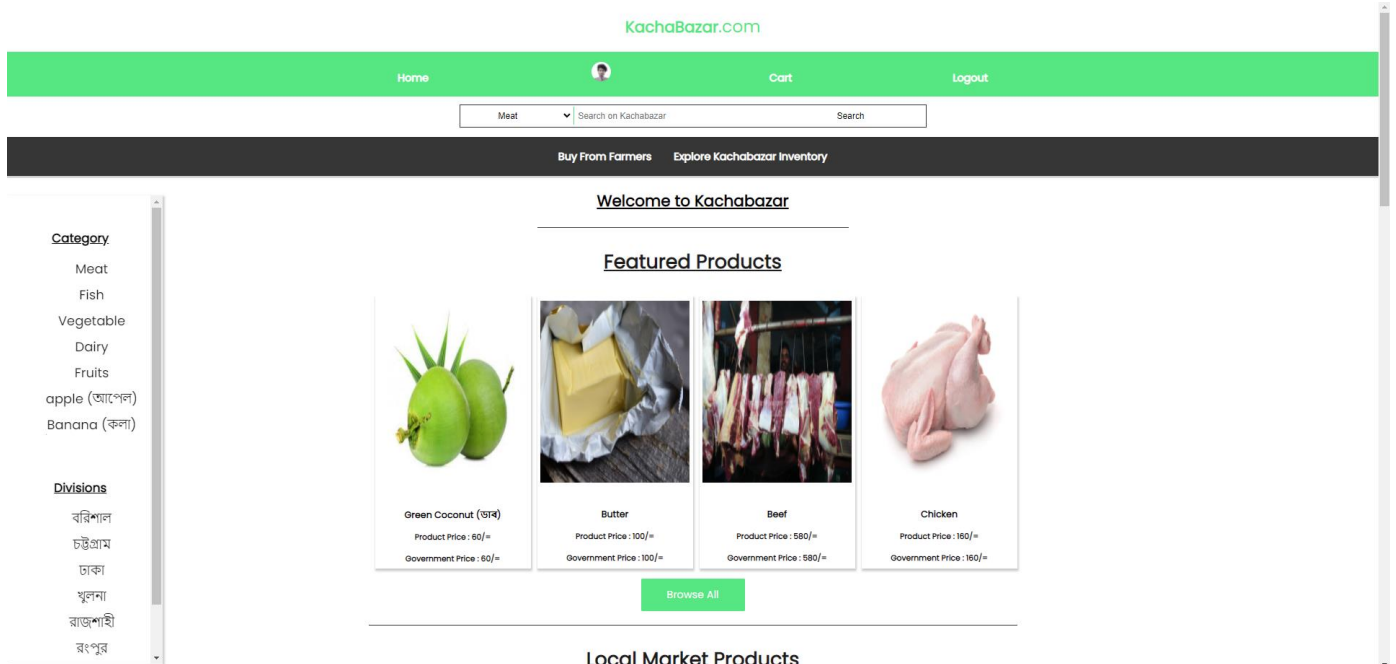


Fig 17 – Homepage

5.2 SINGLE PRODUCT PAGE

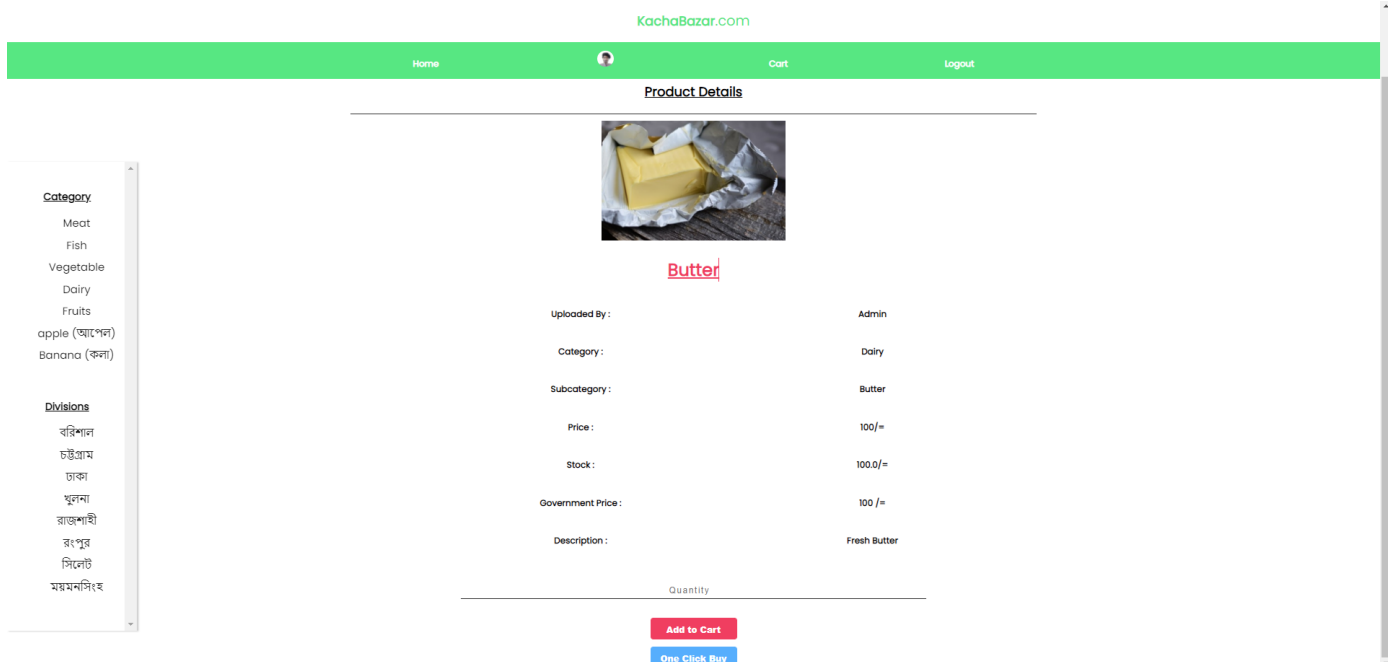


Fig 17 – Single Product Page

5.3 INVENTORY PRODUCTS

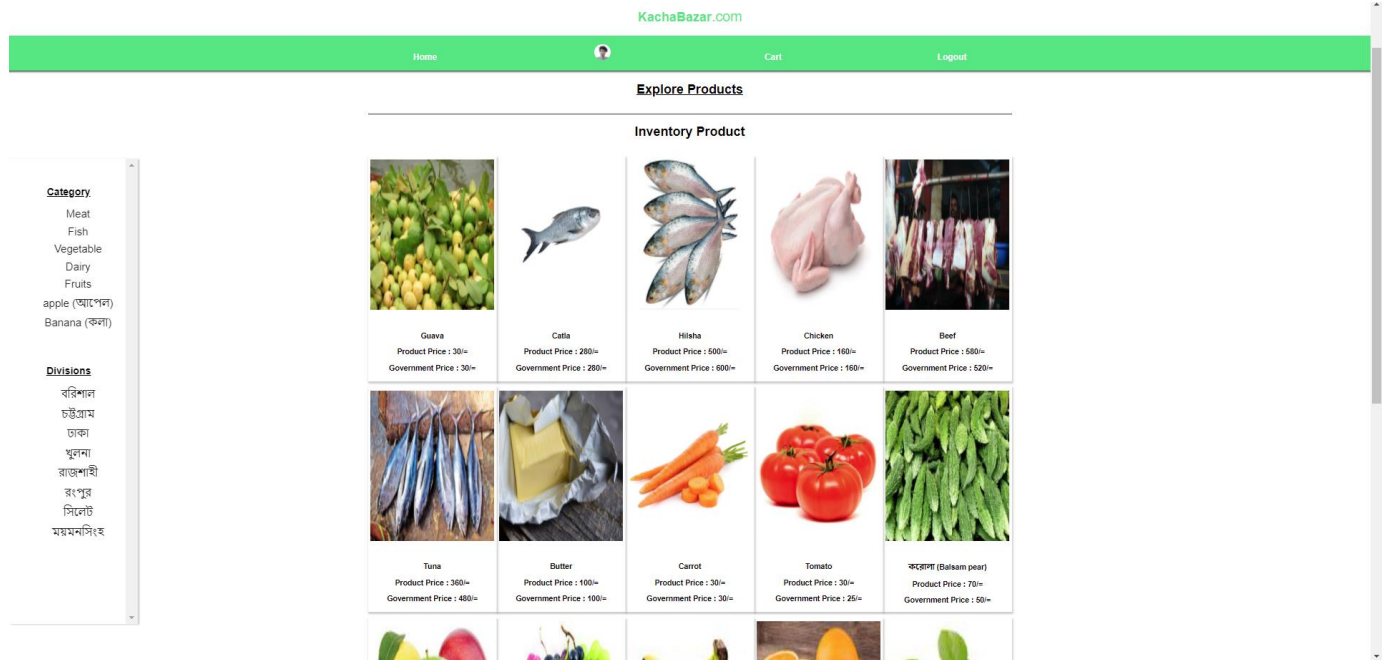


Fig 18 – Inventory Products

5.4 CART PAGE




Cart Details				
No.		Product	Quantity	Subtotal
1	Beef		1.0	580.0/=
2	Butter		3.0	300.0/=
3	Catla		1.0	280.0/=
Total			1160.0 /=	
Checkout				

Fig 19 – Cart Page

5.5 CHECKOUT PAGE

Buy Product


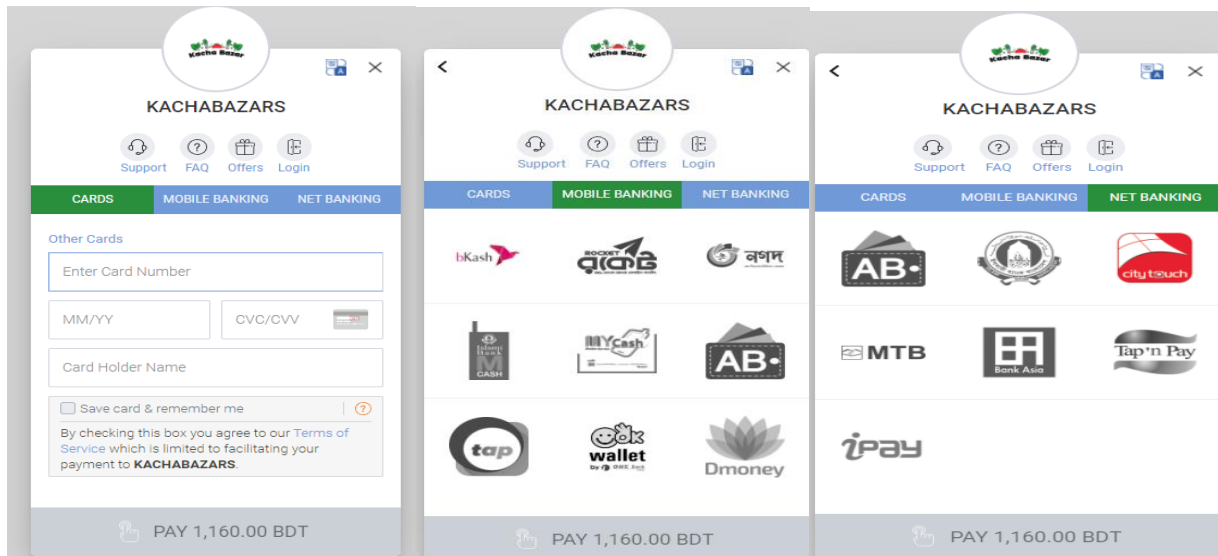
Use your current information to buy	or add new Information
 <p><u>Expected Date</u></p> <p>03 / 22 / 2022</p> <p>Order</p>	<p>Shipping Information</p> <p>Name <input type="text"/></p> <p>Phone <input type="text"/></p> <p>Select Division <input type="text"/></p> <p><input type="text"/></p> <p><input type="text"/></p> <p>Village <input type="text"/></p> <p>Street <input type="text"/></p> <p>ZipCode <input type="text"/></p> <p><u>Expected Date</u></p> <p>mm / dd / yyyy</p> <p>Checkout</p>

Fig 20 – Checkout Page

5.6 PAYMENT GATEWAY PAGE



The figure displays three screenshots of the KACHABAZARS payment gateway page, each showing a different payment method selected in the top navigation bar.

Left Screenshot (Cards): Shows the "Other Cards" section with fields for "Enter Card Number", "MM/YY", "CVC/CVV", and "Card Holder Name". Below these fields is a checkbox for "Save card & remember me" and a note about the service. The bottom bar shows "PAY 1,160.00 BDT".

Middle Screenshot (MOBILE BANKING): Shows a grid of mobile banking logos including bKash, Rocket, bKash, YCash, AB, tap, eK wallet, and Dmoney. The bottom bar shows "PAY 1,160.00 BDT".

Right Screenshot (NET BANKING): Shows a grid of net banking logos including AB, MTB, Bank Asia, Tap'n Pay, and iPay. The bottom bar shows "PAY 1,160.00 BDT".

Fig 21 – Payment Gateway

5.7 PAYMENT SUCCESSFUL CARD

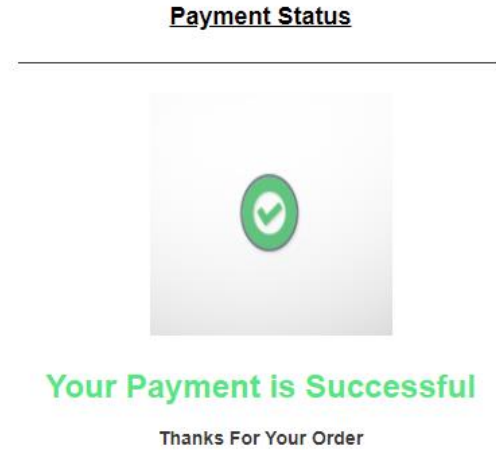



Fig 22 – Successful

5.8 PROFILE INFORMATION

Profile

Basic Information



Minhaj UI Islam

Address : abc , abc
 সিটি কর্পোরেশন, বরিশাল সদর, বরিশাল, বরিশাল

Phone No : +880 12345

Date Of Birth : 1998-01-29

Order History

Orders

From Kacha Bazar

Product	Quantity	Buying from	Order Status	Price per piece
Chicken	1.0 Kilogram	Kacha Bazar	Completed	160/=
Beef	1.0 Kilogram	Kacha Bazar	Unallocated	580/=
Butter	3.0 Kilogram	Kacha Bazar	Unallocated	100/=
Cattle	1.0 Kilogram	Kacha Bazar	Unallocated	290/=

From Seller

Product	Quantity	Buying from	Order Status	Price
---------	----------	-------------	--------------	-------

Fig 23 – Profile

5.9 ANDROID APP – HOME PAGE & CATEGORY PRODUCT PAGE

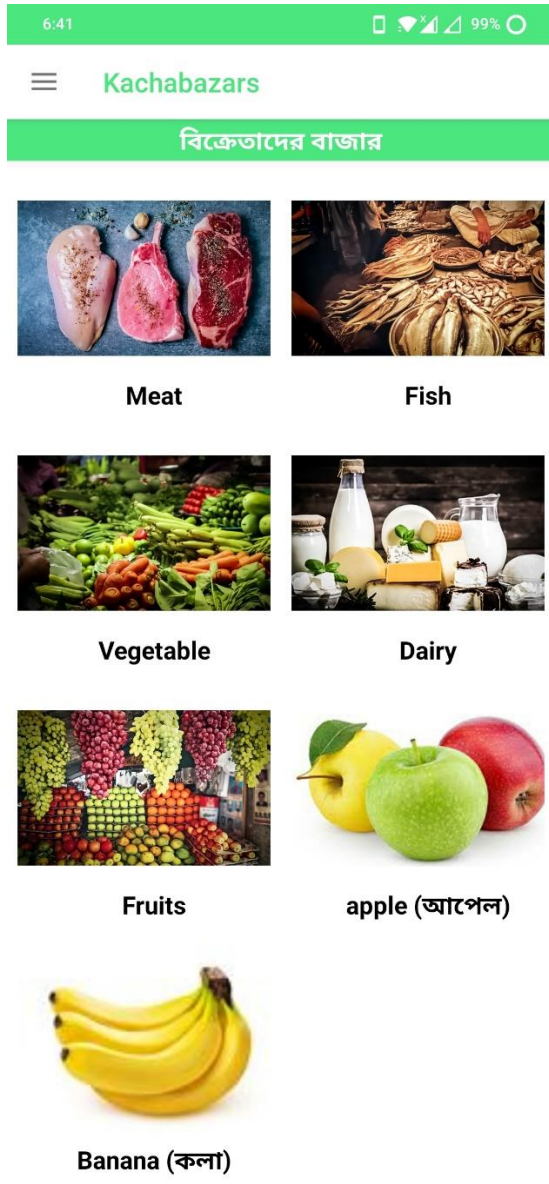


Fig 24 – Homepage



Fig 25 – Product List Page

5.10 ANDROID APP – CART & CART CHECKOUT

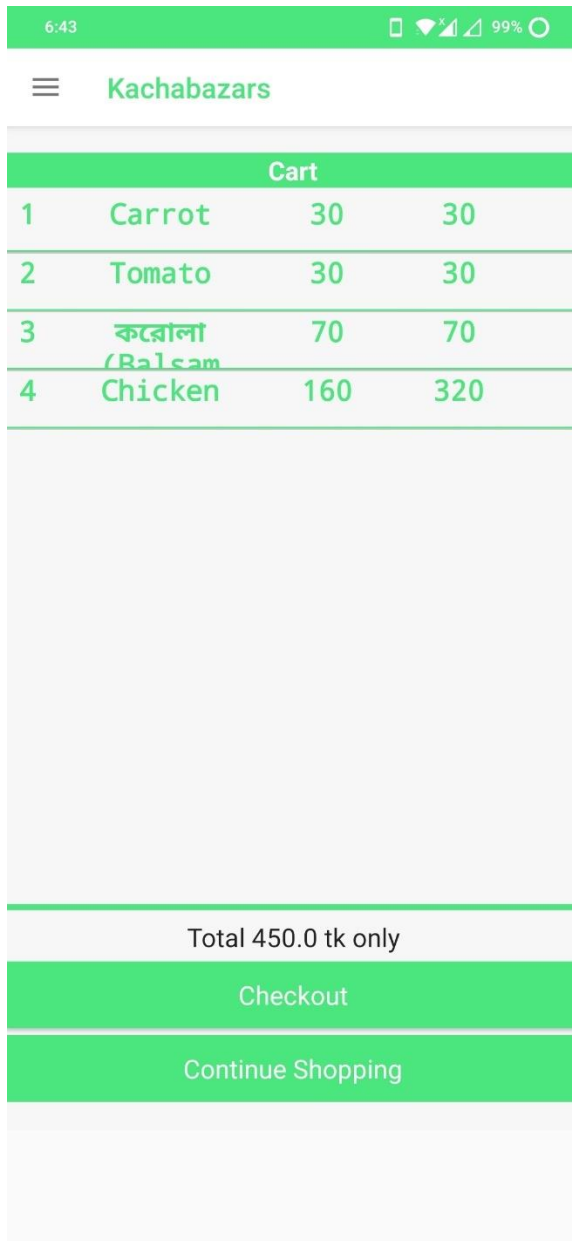


Fig 26 – Cart



Fig 27 – Cart Checkout Page

5.11 ANDROID APP – PAYMENT METHODS

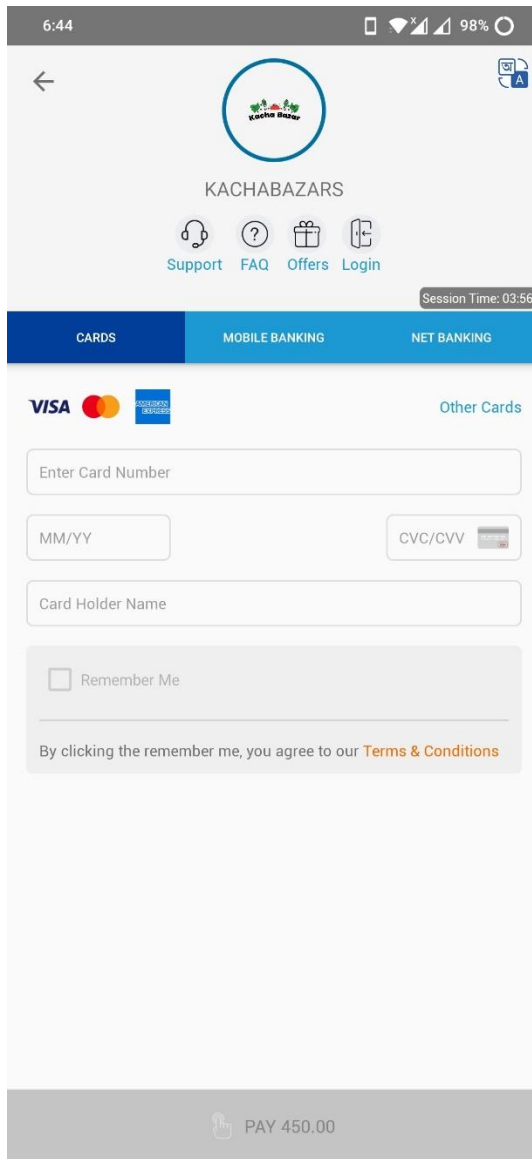


Fig 28 – Payment Card

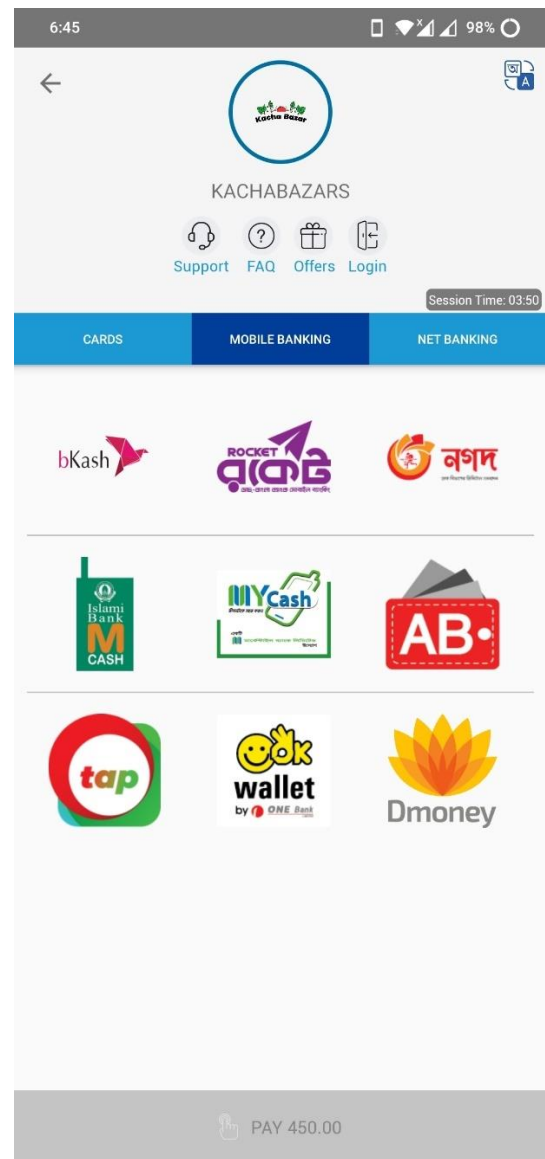


Fig 29 – Payment Mobile Banking

5.12 ANDROID APP – SELLER PAGE AND ADDING PRODUCT



Fig 30 – Seller Product Add

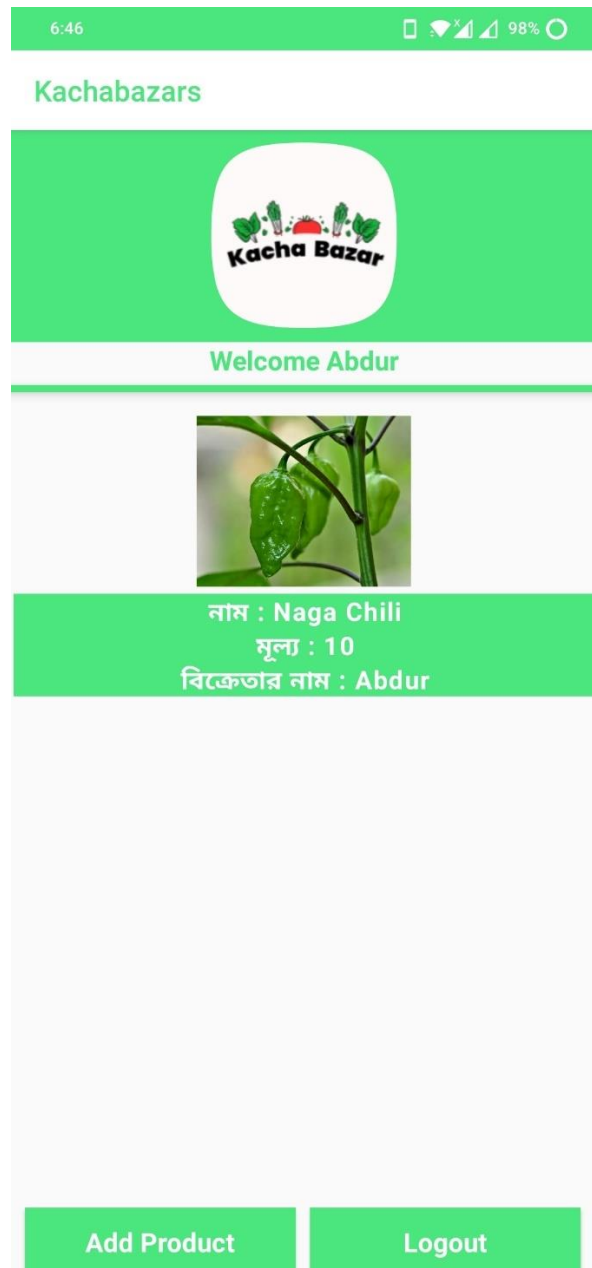
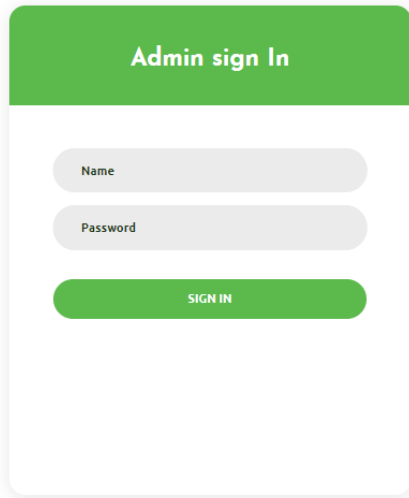


Fig 31 – Seller Home

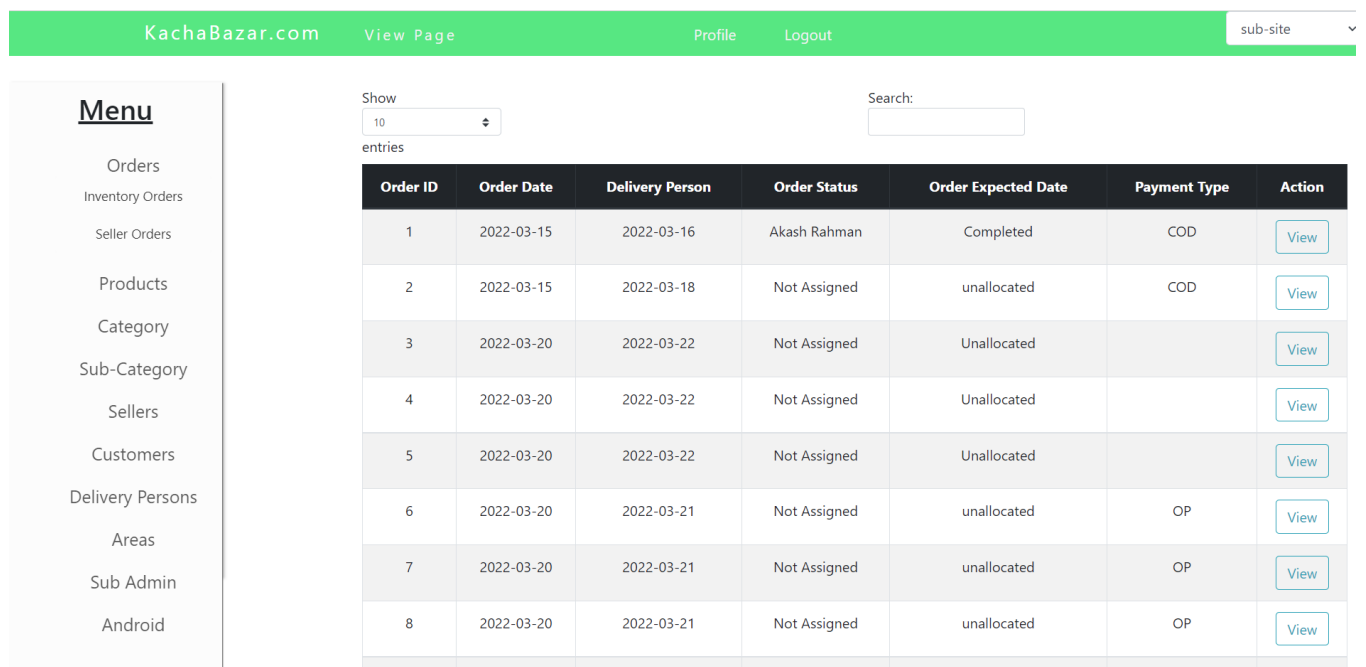
5.13 ADMIN LOGIN PAGE



The image shows a mobile app interface for the Admin sign In page. It features a green header with the text "Admin sign In". Below the header, there are two input fields: "Name" and "Password", both with rounded rectangular borders. At the bottom of the form is a green button with the text "SIGN IN" in white capital letters.

Fig 32 – Admin Login Page

5.14 ADMIN HOMEPAGE



The image shows the Admin dashboard interface. At the top is a green navigation bar with the text "KachaBazar.com", "View Page", "Profile", "Logout", and a dropdown menu labeled "sub-site". Below the navigation bar, on the left, is a "Menu" section with a list of items: Orders, Inventory Orders, Seller Orders, Products, Category, Sub-Category, Sellers, Customers, Delivery Persons, Areas, Sub Admin, and Android. On the right, there is a table displaying orders. Above the table, there is a "Show" dropdown menu set to "10" and a "Search:" input field. The table has columns for Order ID, Order Date, Delivery Person, Order Status, Order Expected Date, Payment Type, and Action. The table contains 8 rows of data, each with a "View" button in the Action column.

Order ID	Order Date	Delivery Person	Order Status	Order Expected Date	Payment Type	Action
1	2022-03-15	2022-03-16	Akash Rahman	Completed	COD	View
2	2022-03-15	2022-03-18	Not Assigned	unallocated	COD	View
3	2022-03-20	2022-03-22	Not Assigned	Unallocated		View
4	2022-03-20	2022-03-22	Not Assigned	Unallocated		View
5	2022-03-20	2022-03-22	Not Assigned	Unallocated		View
6	2022-03-20	2022-03-21	Not Assigned	unallocated	OP	View
7	2022-03-20	2022-03-21	Not Assigned	unallocated	OP	View
8	2022-03-20	2022-03-21	Not Assigned	unallocated	OP	View

Fig 33 – Admin dashboard showing orders with pagination

5.15 ADMIN PRODUCT PAGE









KachaBazar.com <input type="text" value="Type to search"/> <input type="button" value="Search"/> Home Profile Notification									
Menu Orders Products Add Products View Products Category Sub-Category Sellers Customers Delivery Persons Areas Sub Admin	ID	Name	Description	ImageName	Seller's Price	Government Price	Stock	Action	
	1	Guava	Fresh Guava		30	30	100.0	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
	2	Catla	Fresh Catla		280	280	100.0	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
	3	Hilsha	Fresh Hilsha		500	480	60.0	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
	4	Chicken	Processed Chicken		160	160	63.0	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
	5	Beef	Processed Beef		580	570	65.0	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
	6	Tuna	Fresh Tuna		360	360	65.0	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
	7	Butter	Fresh Butter		100	100	100.0	<input type="button" value="Update"/>	<input type="button" value="Delete"/>
	8	Carrot	Fresh Carrot		30	30	50.0	<input type="button" value="Update"/>	<input type="button" value="Delete"/>

Fig 34 – Admin product control page

5.16 ADMIN AREA CONTROL PAGE

KachaBazar.com View Page <input type="text" value="Type to search"/> <input type="button" value="Search"/>												
Menu Orders Products Category Sub-Category Sellers Customers Delivery Persons Areas Division District Upazilla Union Sub Admin	Add Union		<input type="text" value="বরিশাল"/>	<input type="text" value="Select division first"/>	<input type="text" value="Select district first"/>	<input type="text" value="Enter Union name"/>	<input type="button" value="Submit"/>					
	Union List											
	Show <input type="text" value="10"/> entries		Search: <input type="text"/>									
	Union ID	Union Name	Upazilla ID & Name			Action						
2351	সিটি কর্পোরেশন	10 বরিশাল সদর			<input type="button" value="Edit"/>	<input type="button" value="Delete"/>						
1653	ইসলামাবাদ	100 কক্স বাজার সদর			<input type="button" value="Edit"/>	<input type="button" value="Delete"/>						
1654	ইসলামপুর	100 কক্স বাজার সদর			<input type="button" value="Edit"/>	<input type="button" value="Delete"/>						
1655	পোকখালী	100 কক্স বাজার সদর			<input type="button" value="Edit"/>	<input type="button" value="Delete"/>						
1656	ঈদগাঁও	100 কক্স বাজার সদর			<input type="button" value="Edit"/>	<input type="button" value="Delete"/>						

Fig 35 – Admin Area control page

5.17 ADMIN PANEL REPORT GENERATION

Report For 3-2022 From abcd(বরিশাল) created on 2022-03-20		Download PDF
#	Inventory Report	
1	Total Orders = 9	
2	Total Sales = 0.0BDT	
3	Completed Orders = 0	
4	Incomplete Orders = 9	
5	Total Online Payment(0) = 0.0BDT	
6	Total COD(0) = 0.0BDT	
#	Seller Orders Report	
1	Total Orders = 2	
2	Total Sales = 0.0BDT	
3	Completed Orders = 0	
4	Incomplete Orders = 2	
5	Total Online Payment(0) = 0.0BDT	
6	Total COD(0) = 0.0BDT	

Fig 35=6 – Sub-admin report generation

There are a lot more functionalities in this project.

Chapter 6 - SOFTWARE TESTING

There are types of testing that we implement. They are as follows:

While deciding on the focus of testing activities, study project priorities. For example, for an online system, pay more attention to response time. Spend more time on the features used frequently. Decide on the effort required for testing based on the usage of the system. If the system is to be used by a large number of users, evaluate the impact on users due to a system failure before deciding on the effort.

This creates two problems:

- ❖ The time delay between the cause and appearance of the problem.
- ❖ The effect of the system errors on files and records within the system.

The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the systems to limits. The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and on the functional interval is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results. Program level testing, modules level testing integrated and carried out.

There are two major types of testing they are:

- ✓ White Box Testing.
- ✓ Black Box Testing.

6.1 WHITE BOX TESTING

A white box sometimes called “Glass box testing” is a test case design uses that the control structure of the procedural design to drive the test case. Using white box testing methods, the following tests were made on the system

- a) All independent paths within a module have been exercised once. In our system, ensuring that the case was selected and executed checked all case structures. The bugs that were prevailing in some parts of the code were fixed
- b) All logical decisions were checked for the truth and falsity of the values.

6.2 BLACK BOX TESTING

Black box testing focuses on the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box testing rather it is a complementary approach that is likely to uncover a different class of errors than white-box methods.

- ✓ Interface errors.
- ✓ Performance in the data structure.
- ✓ Performance errors.
- ✓ Initializing and termination errors.

6.3 USER ACCEPTANCE TESTING (UAT)

6.3.1 Scope (In Scope – Out of Scope)	
UAT - In Scope	UAT - Out of Scope
Browser Android.	macOS

6.3.2 Assumptions and Constraints
UAT Assumptions
<p>Assumption <i>List the UAT assumptions.</i></p> <ul style="list-style-type: none"> - Test environment: The test cases are conducted by 5 QA Manager and App tester, all communications were blocked, and hard- and software is provided by Project Manager - Test documentation: All UAT test cases are documented within the server. - Error reporting: Errors, failures, and other flaws are reported using apps and website tester tools

6.3.3 Constraints
<p>Constraint <i>List the UAT constraints.</i></p> <ul style="list-style-type: none"> - Resources: <ul style="list-style-type: none"> • Human resources: 5 testers available all time • Provided hardware: Macbook Air, OnePlus, Samsung, Symphony, Xiaomi, Surface Pro 4, HP Spectre, • Tested operating systems: Mac OS X: 10.10, 10.11, 10.12, 10.13 Windows: 8, 8.1, 10 Linux: Arch Linux, Debian, Ubuntu Android 10, 9, 8, 7, 6, 5,

- Browser tests:

Firefox (latest 2 versions)

Chrome (latest 2 versions)

IE (latest 2 versions)

Edge (latest 2 versions)

Safari (latest 2 versions)

Opera (latest 2 versions)

6.3.4 Risks

Description	Probability High Med Low	Impact High Med Low	Mitigation
not properly trained Testers	Low	Med	Ongoing training sessions for testers
Incomplete test environment due to time constraints	Med	Med	Realistic time and resource planning
Error handling: Testers are not aware of how to report bugs	Low	High	Easy-to-use bug reporting solution for UAT test available
UAT test failure	Low	High	Feature-complete development done before UAT test start

6.3.5 Team Roles & Responsibilities

Name	Roles	Responsibilities
MD Minhaj UI Islam	Web and android dev and QA tester	Test management

6.3.6 Entry Criteria

ID	Criteria
5.1	The user must have an internet connection and an account to buy something.

6.3.7 Requirements-Based Test Cases – APP

Tested on OxygenOS, Android 10, and Android 5

ID	Test Cases
1	<p>Launch App.</p> <p><u>Start</u></p> <ul style="list-style-type: none"> • Instruction: Download the app from the play store with any android phone • Keep the internet on • Expected Results: An animated screen with the word Kachabazars • Then the homepage with all the categories will show up. • Unexpected Error: A blank screen
2	<p>App Registration and Login</p> <p><u>Start</u></p> <ul style="list-style-type: none"> • Select (লগ ইন করুন) button at the end of the sidebar • Click “অ্যাকাউন্ট নেই” button and fill up the info and submit for registering • Fill out the box and Click “লগইন” to simply log in with this id: 123 and pass: 123 • Expected Results: Will be redirected to the homepage with categories and a greeting saying “লগইন সফল হয়েছে” • Unexpected Error: Toast message containing errors
3	<p>View Category</p> <p><u>Start</u></p> <ul style="list-style-type: none"> • Click on any category to see all the products listed in that category • Expected Results: A list of products • Unexpected Error: Toast message containing errors
4	View Product

	<p><u>Start</u></p> <ul style="list-style-type: none"> Click on any category to see all the products listed in that category Click one of the products to see their details Expected Results: A page with product image and description Unexpected Error: Toast message containing errors
5	<p>Buy a product</p> <p><u>Start</u></p> <ul style="list-style-type: none"> Click the button “কিনুন” from the product view page A form will appear. Fill up the necessary information and select the delivery method Select the “অনলাইনে পেমেন্ট” button to pay with any card or mobile banking Select the “ক্যাশ অন ডেলিভারি” to pay when you receive the product Expected Results: A pop-up screen will be showing and a success message will be given Unexpected Error: A failed message with the cause will be shown
6	<p>Adding Product to Cart</p> <p><u>Start</u></p> <ul style="list-style-type: none"> Select a product Fill the box called “পরিমাণ” Click “কার্টে রাখুন” button Expected Results: A success message will be given Unexpected Error: A failed message with the cause will be shown
7	<p>Viewing the Cart</p> <p><u>Start</u></p> <ul style="list-style-type: none"> Click the floating Cart Symbol at the right-bottom corner of the screen Expected Results: A page will Cart details will be given Unexpected Error: A failed message with the cause will be shown
8	<p>Checkout the Cart</p> <p><u>Start</u></p> <ul style="list-style-type: none"> After Viewing the cart, click the “CHECKOUT” button Fill out all the boxes and follow the same procedure as Buying a product at 6.5 Expected Results: A success message will appear Unexpected Error: A failed message with the cause will be shown
9	<p>View Seller Product</p> <p><u>Start</u></p> <ul style="list-style-type: none"> Select “বিক্রেতাদের বাজার” from the left sidebar Expected Results: A Category page will show Unexpected Error: A failed message with the cause will be shown
10	<p>View Local Market</p> <p><u>Start</u></p> <ul style="list-style-type: none"> Select “স্থানীয় বাজার” from the left sidebar Select your favorable option to see the products available Expected Results: A list of products will appear Unexpected Error: A failed message with the cause will be shown
11	<p>Search for a Market</p> <p><u>Start</u></p> <ul style="list-style-type: none"> Select “বাজার খুঁজুন” from the left sidebar Select your favorable location from the dropdown menu

	<ul style="list-style-type: none"> Expected Results: A list of products available in that marketplace will appear Unexpected Error: A failed message with the cause will be shown
12	Profile <u>Start</u> <ul style="list-style-type: none"> Select “প্রোফাইল” from the left sidebar You can see and edit your details from here Expected Results: A page with your profile details will appear Unexpected Error: A failed message with the cause will be shown
13	Sell from the app <u>Start</u> <ul style="list-style-type: none"> Select “বিক্রি করুন” from the left sidebar You'll have to log in with your seller ID A page showing your product will appear You can add new products from there Expected Results: A page with your uploaded products will appear Unexpected Error: A failed message with the cause will be shown
14	Contact Us <u>Start</u> <ul style="list-style-type: none"> Select “আমাদের সাথে যোগাযোগ” from the left sidebar You can fill-up the form and email to the official email of KachaBazars Expected Results: A page with your profile details will appear Unexpected Error: A failed message with the cause will be shown
15	About Us <u>Start</u> <ul style="list-style-type: none"> Select “আমাদের সম্পর্কে জানুন” from the left sidebar All the details can be found here Expected Results: A page with KachaBazars details will be shown here <p>Unexpected Error: A failed message with the cause will be shown</p>
16	Log Out <u>Start</u> <ul style="list-style-type: none"> Select “লগ আউট করুন” from the left sidebar You can only see this if you have already logged in Expected Results: You will be redirected to the home page Unexpected Error: A failed message with the cause will be shown

6.3.8 Test Results – APP Tested on OxygenOS, Android 10, and Android 5				
ID	Test Cases	Pass/Fail	Tested By	Date Tested
1	Launch App	Pass	Minhaj	19/03/2022

2	App Registration and Login	Pass	Minhaj	19/03/2022
3	View Category	Pass	Minhaj	19/03/2022
4	View Product	Pass	Minhaj	19/03/2022
5	Buy a product	Pass	Minhaj	19/03/2022
6	Adding Product to Cart	Pass	Minhaj	19/03/2022
7	Viewing the Cart	Pass	Minhaj	19/03/2022
8	Checkout the Cart	Pass	Minhaj	19/03/2022
9	View Seller Product	Pass	Minhaj	19/03/2022
10	View Local Market	Pass	Minhaj	19/03/2022
11	Search for a Market	Pass	Minhaj	19/03/2022
12	Profile	Pass	Minhaj	19/03/2022
13	Sell from the app	Pass	Minhaj	19/03/2022
14	Contact Us	Pass	Minhaj	19/03/2022
15	About Us	Pass	Minhaj	19/03/2022
16	Log Out	Pass	Minhaj	19/03/2022

6.3.9 Requirements-Based Test Cases – Website

Tested on Google Chrome

ID	Test Cases
1	<p>Opening the website</p> <p><u>Start</u></p> <ul style="list-style-type: none"> • Instruction: Open any browser • Keep the internet on • Go to the URL: http://kachabazars.com:8080/ecommerce/ • Expected Results: The homepage of KachaBazars with some product grid. • Unexpected Error: A screen saying error

2	Register as a customer <u>Start</u> <ul style="list-style-type: none"> To register click the Login button at the top right bar. Below the box see the SIGN UP NOW button and click it. Fill in all the boxes. And click register. Expected results: You will be redirected to the homepage and you'll be able to see your picture at the menu bar. That is your profile Unexpected Error: Registration failed
3	Adding product to cart <u>Start</u> <ul style="list-style-type: none"> Hover the mouse over any product. Select ADD TO CART button A product detail page will show up Write the correct quantity in English numbers and click ADD TO CART NOTE: quantity should be less than the stock Expected result: You'll be redirected to all the product marketplace and products that have been added Unexpected Error: A screen showing error
4	Checking out cart <u>Start</u> <ul style="list-style-type: none"> Click the CART button from the menubar. You'll see all the information from CART Click the CHECKOUT button at the bottom It will redirect you to the payment page Fill in all the info or just click order if you want the product to be delivered to your location. Pay the actual amount. Expected result: The display saying Thank You Unexpected Error: A screen showing error
5	One-click buy <u>Start</u> <ul style="list-style-type: none"> Hover the mouse over any product. Select ONE CLICK BUT/ DETAILS button A product detail page will show up Write the correct quantity in English numbers and click ONE CLICK BUY (NOTE: quantity should be less than the stock) It will redirect you to the payment page Fill in all the info or just click order if you want the product to be delivered to your location. Pay the actual amount. Expected result: The display saying Thank You Unexpected Error: A screen showing error
6	Category-wise Product <u>Start</u> <ul style="list-style-type: none"> Hover the mouse over any Category on the Left Sidebar. Click your desired Category Expected result: A product detail page will show up with the selected category Unexpected Error: A screen showing error
7	Division-wise Product <u>Start</u> <ul style="list-style-type: none"> Hover the mouse over any Division on the Left Sidebar. Click your desired Division Expected result: A product detail page will show up with the selected division. Unexpected Error: A screen showing error

8	All Farmers product <u>Start</u> <ul style="list-style-type: none"> • Hover the mouse over BUY FROM FARMER from the secondary menu bar • Click the button. • Excepted result: All products from the farmer. • Unexpected Error: A screen showing error
9	All Product from Inventory <u>Start</u> <ul style="list-style-type: none"> • Hover the mouse over EXPLORE KACHABAZAR INVENTORY from the secondary menu bar. • Click the button. • Excepted result: All products from Inventory. • Unexpected Error: A screen showing error
10	Search in Kachabazar <u>Start</u> <ul style="list-style-type: none"> • Click the search bar located at the top bar • Select a category from the dropdown menu • Type the desired product in the search bar • Click the SEARCH button • Excepted result: All products that match the search criteria. • Unexpected Error: A screen showing error

6.3.10 Test Results – Website Tested on Google Chrome				
ID	Test Cases	Pass/Fail	Tested By	Date Tested
1	Opening the website	Pass	Minhaj	19/03/2022
2	Register as a customer	Pass	Minhaj	19/03/2022
3	Adding product to cart	Pass	Minhaj	19/03/2022
4	Checking out cart	Pass	Minhaj	19/03/2022
5	One-click buy	Pass	Minhaj	19/03/2022
6	Category-wise Product	Pass	Minhaj	19/03/2022
7	Division-wise Product	Pass	Minhaj	19/03/2022
8	All Farmers product	Pass	Minhaj	19/03/2022
9	All Product from Inventory	Pass	Minhaj	19/03/2022

10	Search in KachaBazar	Pass	Minhaj	19/03/2022
----	----------------------	------	--------	------------

6.3.11 Requirements-Based Test Cases – Website ADMIN – Tested on Google Chrome

ID	Test Cases
1	Opening the ADMIN PANEL <u>Start</u> <ul style="list-style-type: none"> • Instruction: Open any browser • Keep the internet on • Go to the URL: http://kachabazars.com:8080/ecommerce/admin-login.jsp • A login interface will show up. Use ID and Password to log in. • ID: admin & PASS: admin123 • Expected Results: Admin Panel with a lot of entries • Unexpected Error: A screen saying error
2	Viewing Orders <u>Start</u> <ul style="list-style-type: none"> • Click at the end of the row "VIEW" to see all the info of an order • Expected results: A page with all the info will appear • Unexpected Error: A page displaying an error
3	Updating Order <u>Start</u> <ul style="list-style-type: none"> • After viewing the order, change the necessary things. • Click the "UPDATE" button at the end of the page • The page will get reloaded and updated • Expected result: A page with the order list will appear • Unexpected Error: A screen showing error
4	Product List <u>Start</u> <ul style="list-style-type: none"> • Click "PRODUCT" from the left sidebar • Click the "VIEW PRODUCT" from the expanded menu • Expected result: A page with all the product lists will appear • Unexpected Error: A screen showing error
5	Add Product <u>Start</u> <ul style="list-style-type: none"> • Click "PRODUCT" from the left sidebar • Click the "ADD PRODUCT" from the expanded menu • Fill up all the info • Click the SUBMIT button to submit OR if you want to add more than one product click ADD MORE button • Expected result: If you click SUBMIT button the product list page will appear with the new product added and If you click ADD MORE button the form will appear again. • Unexpected Error: A screen showing error

6	Update Product <u>Start</u> <ul style="list-style-type: none"> From the Product List page, click the “UPDATE” button to the product that you want to update Fill out all the information and click the “UPDATE” button Expected result: The page with all the product lists will appear Unexpected Error: A screen showing error
7	Delete Product <u>Start</u> <ul style="list-style-type: none"> From the Product List page, click the DELETE button to the product that you want to delete Expected result: The page with all the product lists will appear Unexpected Error: A screen showing error
8	Category List <u>Start</u> <ul style="list-style-type: none"> Click “CATEGORY” from the left sidebar Click the “VIEW CATEGORY” from the expanded menu Expected result: A page with all the Categories will appear Unexpected Error: A screen showing error
9	Add Category <u>Start</u> <ul style="list-style-type: none"> Click “CATEGORY” from the left sidebar Click the “ADD CATEGORY” from the expanded menu Fill up all the info Click the SUBMIT button to submit, Expected result: The Category list page will appear Unexpected Error: A screen showing error
10	Update Category <u>Start</u> <ul style="list-style-type: none"> From the CATEGORY List page, click the “UPDATE” button to the category that you want to update Fill out all the information and click the “UPDATE” button Expected result: The page with all the CATEGORY lists will appear Unexpected Error: A screen showing error
11	Delete Category <u>Start</u> <ul style="list-style-type: none"> From the Category List page, click the DELETE button to the product that you want to delete Expected result: The page with all the Category lists will appear Unexpected Error: A screen showing error
12	Area <u>Start</u> <ul style="list-style-type: none"> From the left sidebar click AREA Select the area you want o audit Do the CRUD operation accordingly Expected result: All the operations will be successful Unexpected Error: A screen showing error

13	Sub-Admin <u>Start</u> <ul style="list-style-type: none"> From the left sidebar click SUB ADMIN Do the CRUD operation accordingly Excepted result: All the operations will be successful Unexpected Error: A screen showing error
14	About us page controlling for Android <u>Start</u> <ul style="list-style-type: none"> From the left sidebar click ANDROID Edit the desired section Click SUBMIT Excepted result: All the operations will be successful Unexpected Error: A screen showing error

6.3.12 Test Results – Website ADMIN PANEL – Tested on Google Chrome				
ID	Test Cases	Pass/Fail	Tested By	Date Tested
1	Opening the ADMIN PANEL	Pass	Minhaj	19/03/2022
2	Viewing Orders	Pass	Minhaj	19/03/2022
3	Updating Order	Pass	Minhaj	19/03/2022
4	Product List	Pass	Minhaj	19/03/2022
5	Add Product	Pass	Minhaj	19/03/2022
6	Update Product	Pass	Minhaj	19/03/2022
7	Delete Product	Pass	Minhaj	19/03/2022
8	Category List	Pass	Minhaj	19/03/2022
9	Add Category	Pass	Minhaj	19/03/2022
10	Update Category	Pass	Minhaj	19/03/2022
11	Delete Category	Pass	Minhaj	19/03/2022
12	Area	Pass	Minhaj	19/03/2022

13	Sub-Admin	Pass	Minhaj	19/03/2022
14	About us page controlling for Android	Pass	Minhaj	19/03/2022

Chapter 7 - CONCLUSION

The drastic change in technology has influenced each sector in our country. While the new tech is gazing at us on the horizon, we strive to solve the economic problem of our root-level people, the farmers, the fishermen, etc. To make a stand for them we have developed the KachaBazar and tried to Bring the consumer and producer closer to each other. The dependency on the middle man is minimized so that the market becomes stable and affordable. To make the implementation work flawlessly, effectively, and efficiently we used best practices and state-of-the-art methods. The change that this project can do to society cannot be observed in one day, but it will surely do some improvements. And we are always up for the future challenges to come.