CS5002NI Software Engineering

35% Individual Coursework

MAIN-SIT

AY 2024-2025

Credit: 30

Student Name: Santoshi Paudel

Londonmet ID: 23048656

College ID: np01ai4a230103@islingtoncollege.edu.np

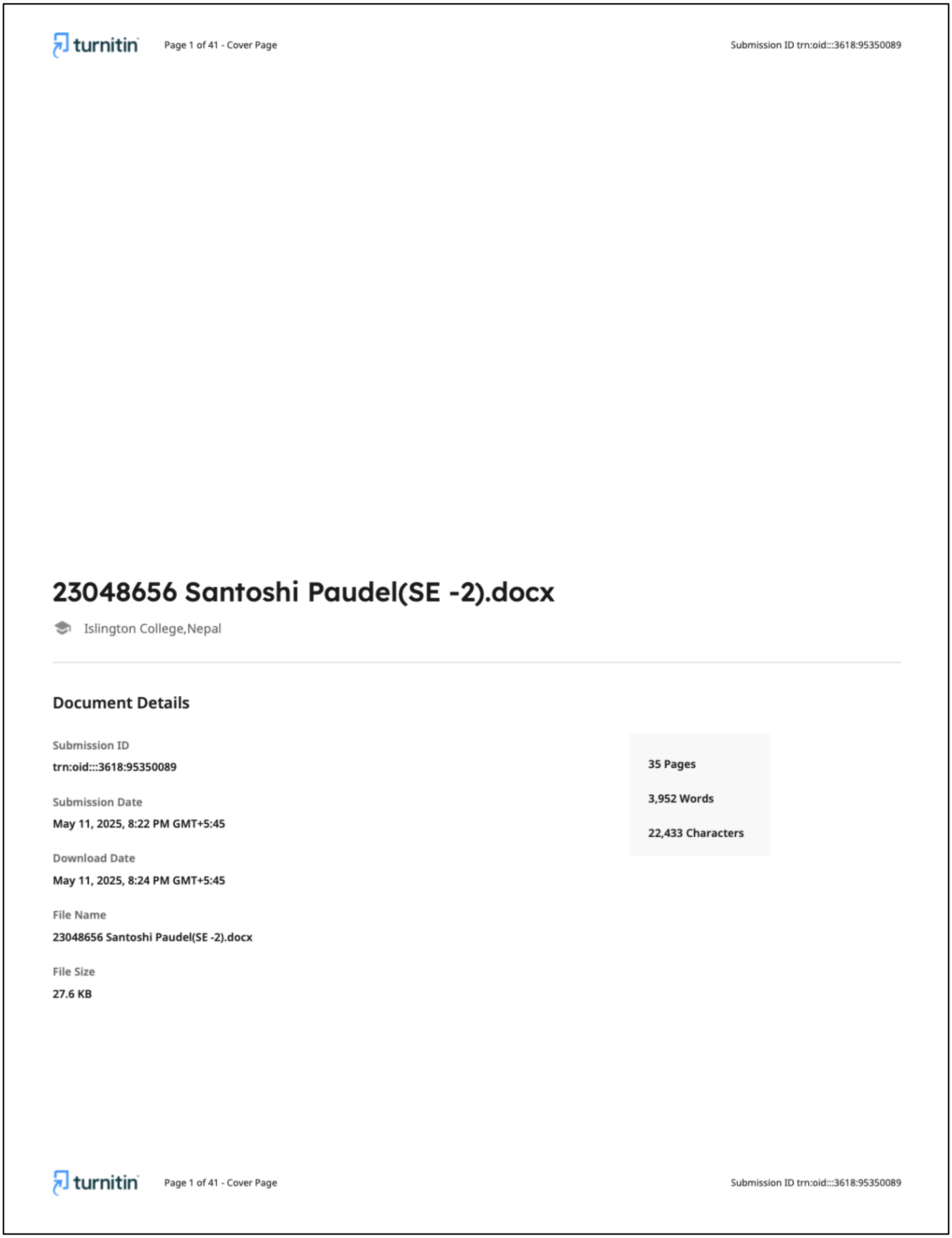
Assignment Due Date: 12th May 2025

Assignment Submission Date: 11th May 2025

Word Count: 4690

Submitted to: Mr. Rubin Thapa Sir

I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.



23048656 Santoshi Paudel(SE -2).docx

Islington College,Nepal

Document Details

Submission ID

trn:oid:::3618:95350089

Submission Date

May 11, 2025, 8:22 PM GMT+5:45

Download Date

May 11, 2025, 8:24 PM GMT+5:45

File Name

23048656 Santoshi Paudel(SE -2).docx

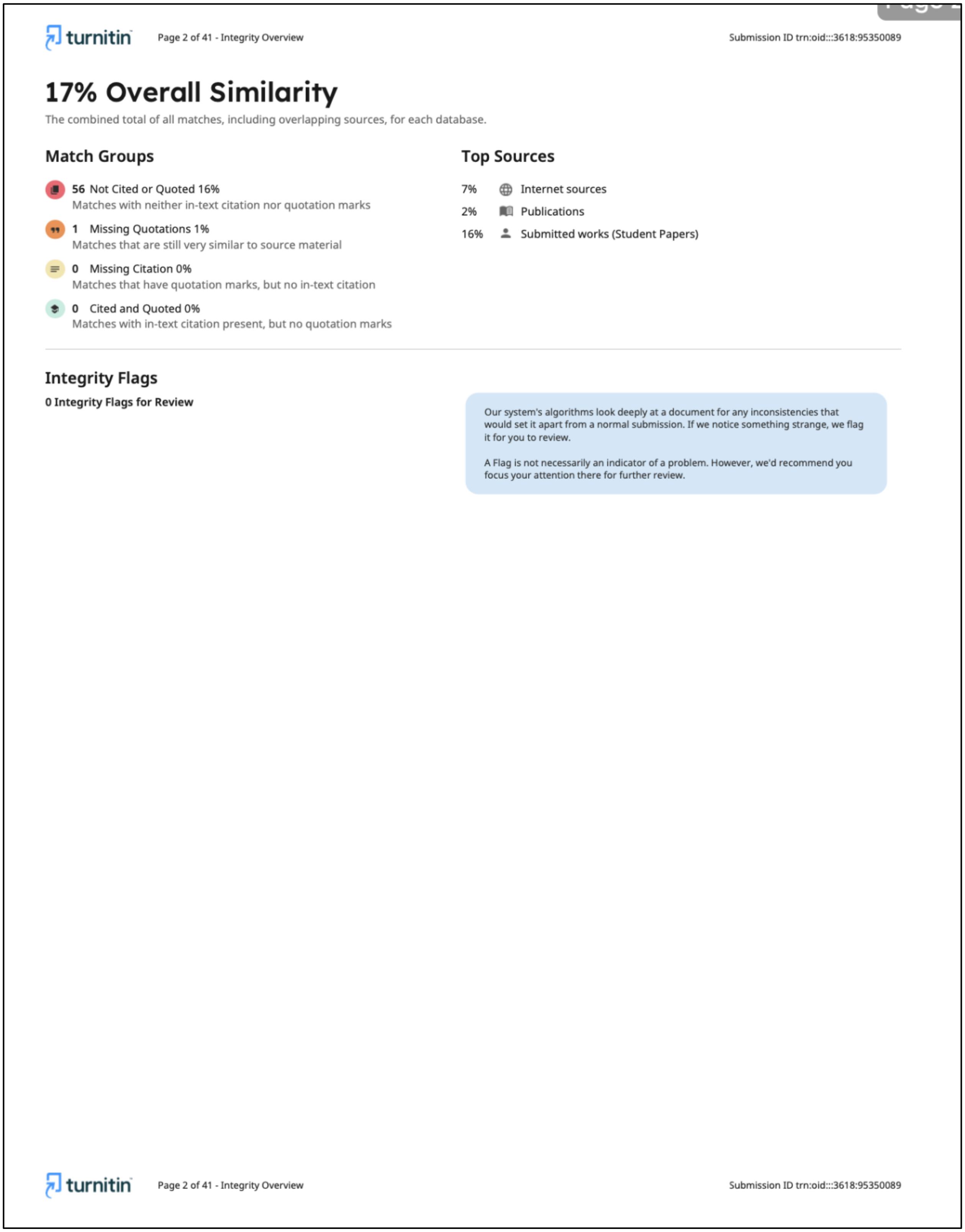
File Size

27.6 KB

35 Pages

3,952 Words

22,433 Characters



17% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **56** Not Cited or Quoted 16%
  Matches with neither in-text citation nor quotation marks
- 1 Missing Quotations 1%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

7%	Internet sources
2%	Publications
16%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

Global Tech Corporation which is a large multinational company, specializes in technology-related products, infrastructure and services. It is a company based on USA and is a technology investment holding company. It was established in the year 1999. It supports areas like artificial intelligence (AI), big data, and telecom. It works with companies around the world to create new tech solutions that help change industries. It now decided to embarked on an ambitious project, which is a next-generation, automated infrastructure system that supports warehouse management. This corporation decided to execute Inventory Management System to enhance efficiency in Nepal. This project primarily aims on minimizing and avoiding the problems that the project has faced earlier such as financial losses, inefficiencies and customer dissatisfaction. These problems were faced due to several causes such as inadequate analysis and system design, absence of clear (Object Oriented Analysis and Design) OOAD, that unfortunately leads to issues like hard to maintain, less scalable, hard to understand and collaborate. The system has function such as Access user to system, Add Purchase and View Purchase, Generate Report, Manage Sales, Manage Products and Make Payment.

The project will involve developing a comprehensive Use Case Diagram covering all necessary use cases. It will also include creating high-level and expanded use case descriptions for key functionalities. Additionally, the process will involve designing Sequence and Collaboration Diagrams alongside a Class Diagram to illustrate system interactions and structure. Ultimately, a prototype of the system will be developed, allowing for iterative feedback**,** which will contribute to the system's long-term functionality and success.

1.1. Aim and Objectives

The main aim of the project is to develop an Inventory Management System with proper use of (Object Oriented Analysis and Design) OOAD to enhance efficiency in Nepal. The objectives of the project are given below.

- To apply OOP principles and explore different software development methodologies.
- To develop Use Case Models, Gantt Chart, Class diagram, Activity diagram, Communication diagram and Functional prototype.
- To minimize and avoid the problems that the project has faced earlier such as financial losses, inefficiencies and customer dissatisfaction.
- To develop the problem-solving skills.
- To enhance time management skills by completing the project within given time period.

2. Work Breakdown Structure

The Work Breakdown Structure (WBS) is basically a common technique used to break down works into smaller tasks to make the work more approachable and manageable. It is one of the very important documents used for project management. A Work Breakdown Structure (WBS) integrates project cost, scope and deliverables into a unified framework (Workbreakdownstructure.com., 2025). For the given coursework, iterative waterfall model is used for developing the Work Breakdown Structure.

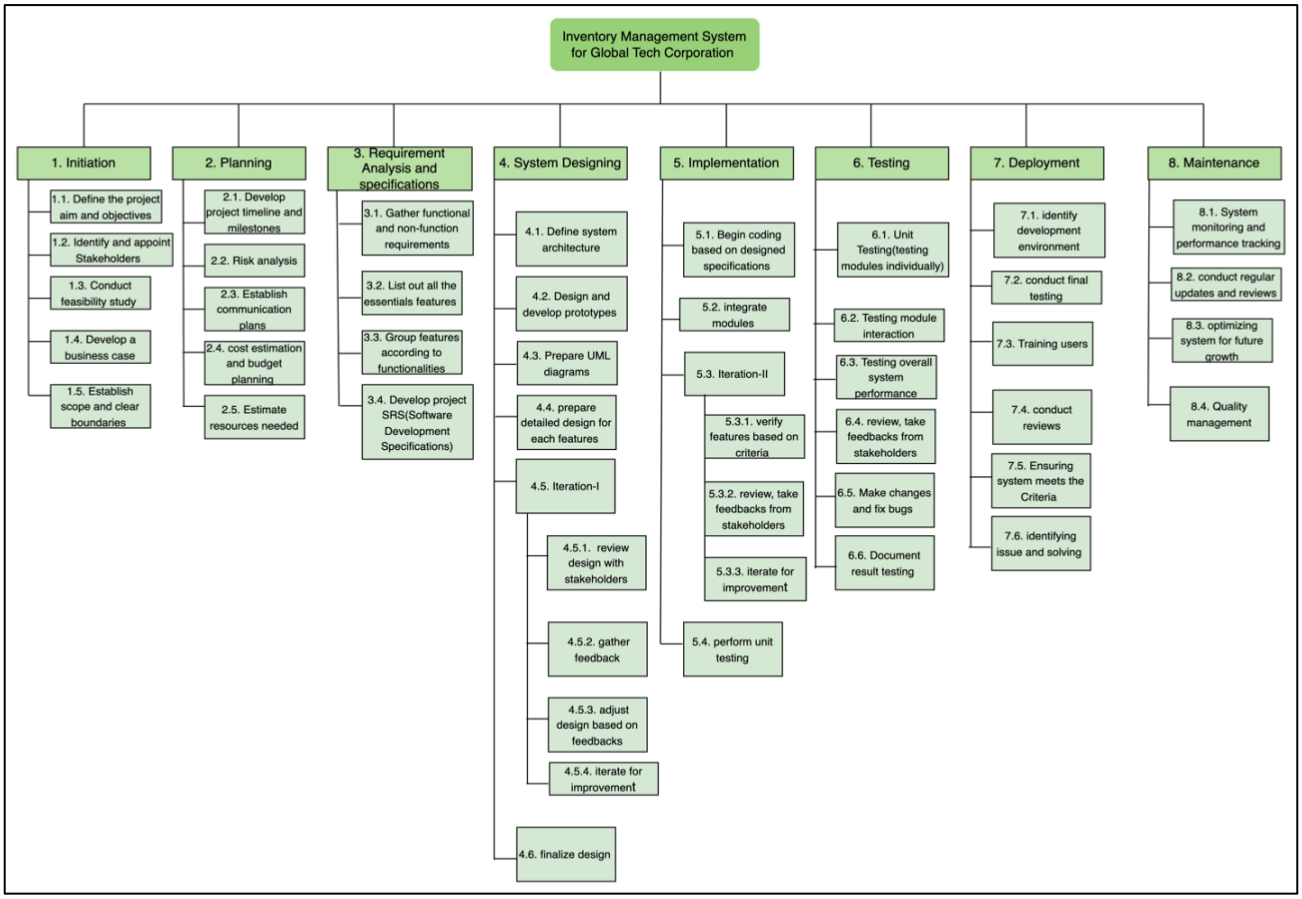

Inventory Management System for Global Tech Corporation
1. Initiation
1.1. Define the project aim and objectives
1.2. Identify and appoint Stakeholders
1.3. Conduct feasibility study
1.4. Develop a business case
1.5. Establish scope and clear boundaries
2. Planning
2.1. Develop project timeline and milestones
2.2. Risk analysis
2.3. Establish communication plans
2.4. cost estimation and budget planning
2.5. Estimate resources needed
3. Requirement Analysis and specifications
3.1. Gather functional and non-function requirements
3.2. List out all the essentials features
3.3. Group features according to functionalities
3.4. Develop project SRS(Software Development Specifications)
4. System Designing
4.1. Define system architecture
4.2. Design and develop prototypes
4.3. Prepare UML diagrams
4.4. prepare detailed design for each features
4.5. Iteration-I
4.5.1. review design with stakeholders
4.5.2. gather feedback
4.5.3. adjust design based on feedbacks
4.5.4. iterate for improvement
4.6. finalize design
5. Implementation
5.1. Begin coding based on designed specifications
5.2. integrate modules
5.3. Iteration-II
5.3.1. verify features based on criteria
5.3.2. review, take feedbacks from stakeholders
5.3.3. iterate for improvement
5.4. perform unit testing
6. Testing
6.1. Unit Testing(testing modules individually)
6.2. Testing module interaction
6.3. Testing overall system performance
6.4. review, take feedbacks from stakeholders
6.5. Make changes and fix bugs
6.6. Document result testing
7. Deployment
7.1. identify development environment
7.2. conduct final testing
7.3. Training users
7.4. conduct reviews
7.5. Ensuring system meets the Criteria
7.6. identifying issue and solving
8. Maintenance
8.1. System monitoring and performance tracking
8.2. conduct regular updates and reviews
8.3. optimizing system for future growth
8.4. Quality management

Figure 1: Work Breakdown Structure (Iterative Waterfall Model)

In the figure above, the Work Breakdown Structure follows the iterative waterfall model. The WBS comprises of eight breakdowns namely, Initiation, planning, requirements analysis and specifications, system designing, implementation, testing, deployment and maintenance. Within each phase certain tasks are performed.

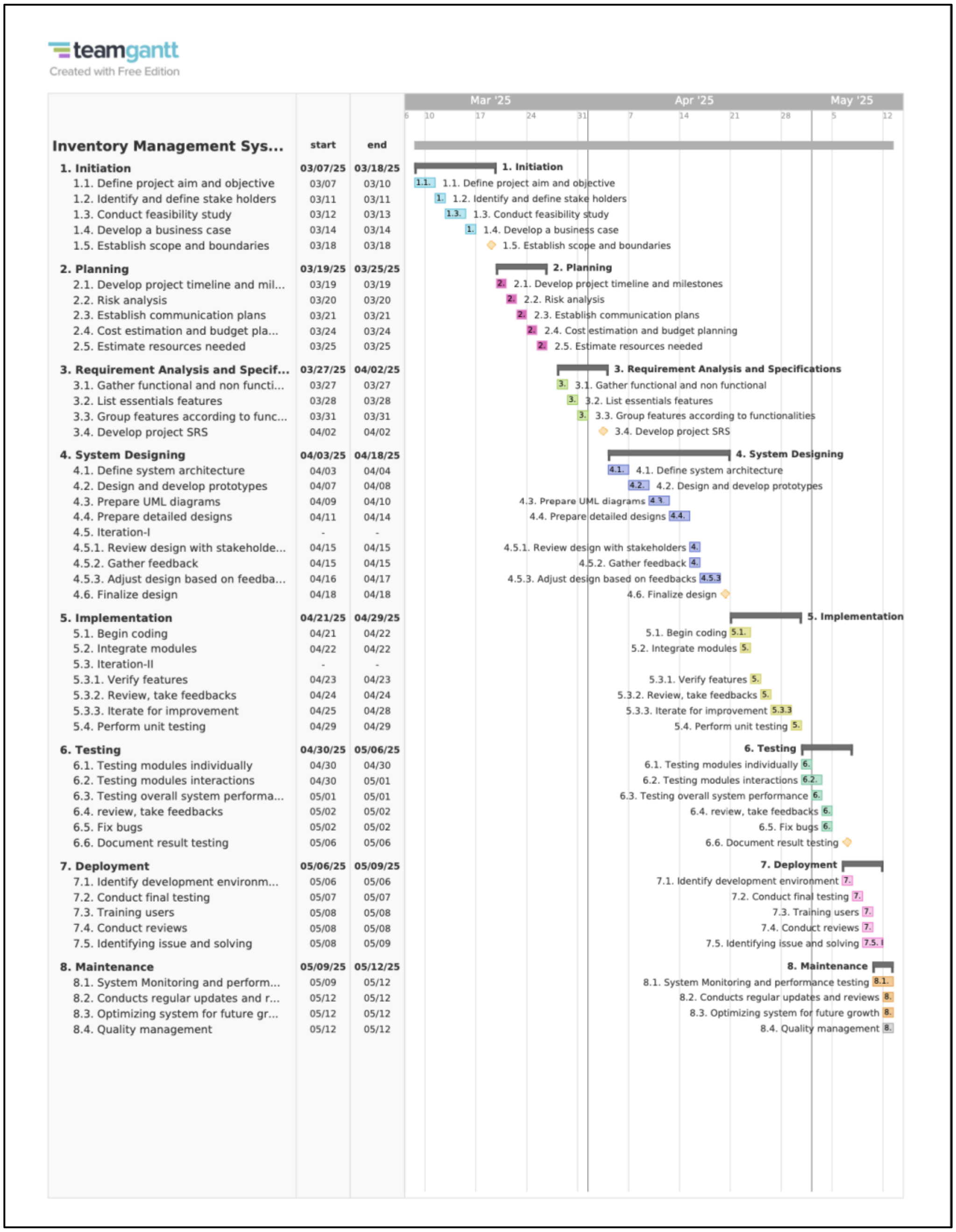
In the Initiation phase, the main goals of the project are decided, the people involved are chosen, a study is done to see if the project can be done, and the work area is clearly explained. In the Planning phase, a schedule is made, possible risks are checked, ways to talk with the team are planned, the cost is guessed, and needed things like tools or people are listed. In the Requirements phase, all the things the system should do are collected, grouped, and written clearly in a document called the SRS.

The System Designing phase is about planning how the system will look and work. Simple early designs are made, diagrams are drawn, and the full design is finished after getting feedback. In the Implementation phase, coding starts, parts of the system are connected, and changes are made if needed. The Testing phase checks if each part works well, tests how they work together, fixes any mistakes, and writes down the results.

In the Deployment phase, the system is set up and used. Final tests are done, users are trained, the system is checked again, and any last problems are fixed. In the Maintenance phase, the system is regularly checked to make sure it works well, updates are added, it is improved when needed, and the quality is kept high.

3. Gantt Chart

A Gantt Chart is a project management chart which was created by Henry Gantt in the early 20th century. It allows project managers and team members to plan projects, create projects schedule and track project progress. It comprises of two main parts, namely a project timeline on right side and a task list on the left side (ProjectManager.com, 2025).



Inventory Management Sys...	start	end
1. Initiation	**03/07/25**	**03/18/25**
1.1. Define project aim and objective	03/07	03/10
1.2. Identify and define stake holders	03/11	03/11
1.3. Conduct feasibility study	03/12	03/13
1.4. Develop a business case	03/14	03/14
1.5. Establish scope and boundaries	03/18	03/18
2. Planning	**03/19/25**	**03/25/25**
2.1. Develop project timeline and mil...	03/19	03/19
2.2. Risk analysis	03/20	03/20
2.3. Establish communication plans	03/21	03/21
2.4. Cost estimation and budget pla...	03/24	03/24
2.5. Estimate resources needed	03/25	03/25
3. Requirement Analysis and Specif...	**03/27/25**	**04/02/25**
3.1. Gather functional and non functi...	03/27	03/27
3.2. List essentials features	03/28	03/28
3.3. Group features according to func...	03/31	03/31
3.4. Develop project SRS	04/02	04/02
4. System Designing	**04/03/25**	**04/18/25**
4.1. Define system architecture	04/03	04/04
4.2. Design and develop prototypes	04/07	04/08
4.3. Prepare UML diagrams	04/09	04/10
4.4. Prepare detailed designs	04/11	04/14
4.5. Iteration-I	-	-
4.5.1. Review design with stakeholde...	04/15	04/15
4.5.2. Gather feedback	04/15	04/15
4.5.3. Adjust design based on feedba...	04/16	04/17
4.6. Finalize design	04/18	04/18
5. Implementation	**04/21/25**	**04/29/25**
5.1. Begin coding	04/21	04/22
5.2. Integrate modules	04/22	04/22
5.3. Iteration-II	-	-
5.3.1. Verify features	04/23	04/23
5.3.2. Review, take feedbacks	04/24	04/24
5.3.3. Iterate for improvement	04/25	04/28
5.4. Perform unit testing	04/29	04/29
6. Testing	**04/30/25**	**05/06/25**
6.1. Testing modules individually	04/30	04/30
6.2. Testing modules interactions	04/30	05/01
6.3. Testing overall system performa...	05/01	05/01
6.4. review, take feedbacks	05/02	05/02
6.5. Fix bugs	05/02	05/02
6.6. Document result testing	05/06	05/06
7. Deployment	**05/06/25**	**05/09/25**
7.1. Identify development environm...	05/06	05/06
7.2. Conduct final testing	05/07	05/07
7.3. Training users	05/08	05/08
7.4. Conduct reviews	05/08	05/08
7.5. Identifying issue and solving	05/08	05/09
8. Maintenance	**05/09/25**	**05/12/25**
8.1. System Monitoring and perform...	05/09	05/12
8.2. Conducts regular updates and r...	05/12	05/12
8.3. Optimizing system for future gr...	05/12	05/12
8.4. Quality management	05/12	05/12

Figure 2: Gantt Chart

4. Use Case Diagram

A Use Case Diagram is a visual representation in Unified Modelling Language (UML) that represents and illustrates the interaction between the System and the actors (users or external system). Use case does not exactly specify the exact method of the system but only specify the expected behavior of the system (Paradigm, 2025).

Symbols used in Use Case Diagram:

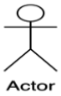
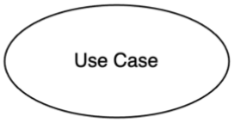
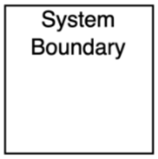
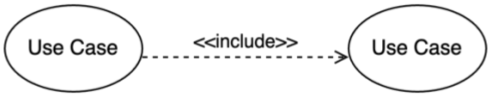
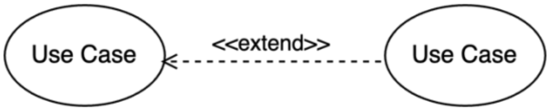
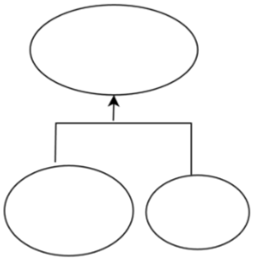
Symbols	Reference Name
	Actor
	Use Case
	System Boundary
	Include
	Extend
———	Association
	Generalization

Table 1: Use Case Diagram Symbols

The use case diagram of the Inventory Management System of Global Tech Corporation is given below.

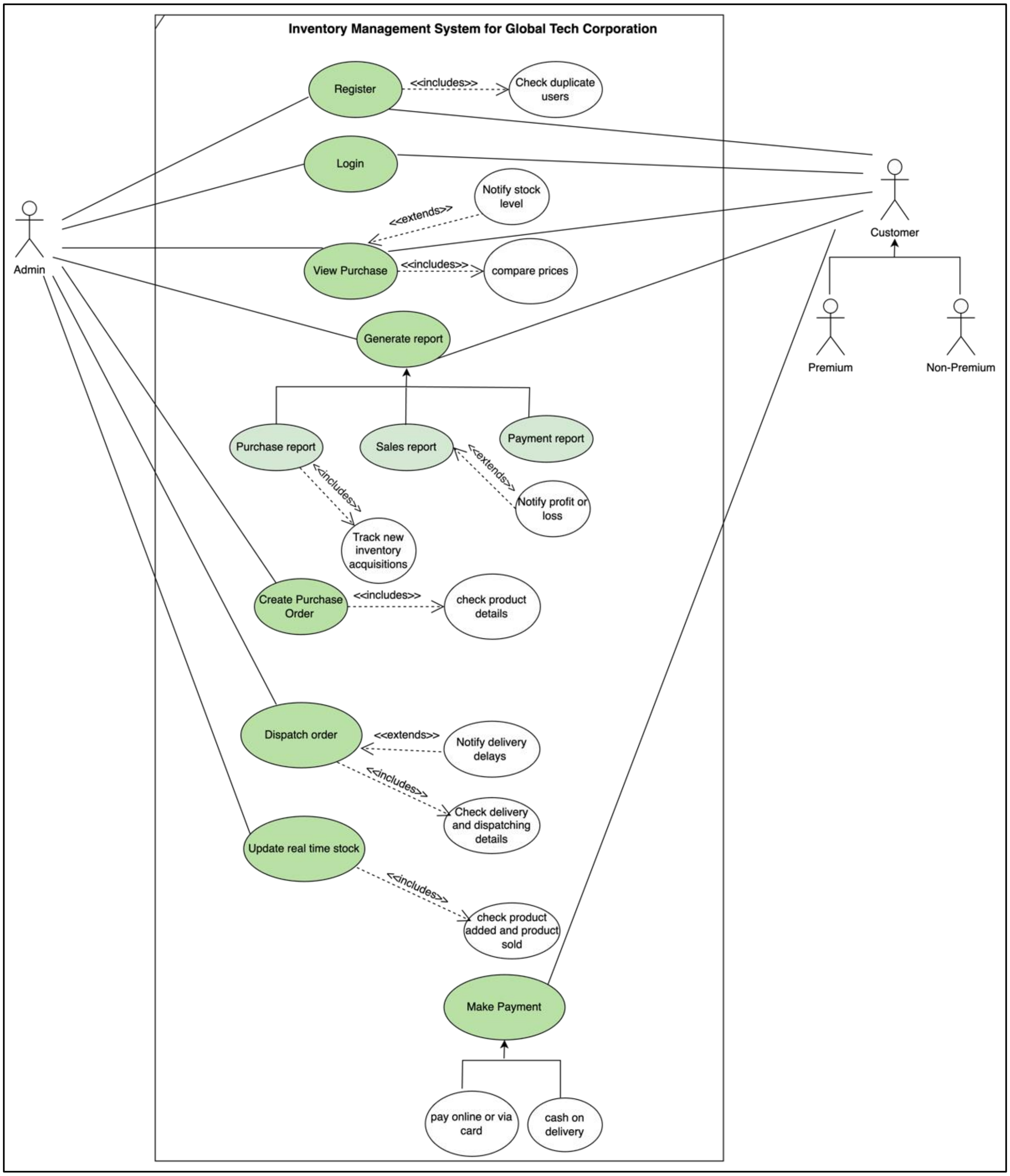


Figure 3: Use Case Diagram

4.1. High Level Description

4.1.1. Register

Use Case:	Register
Actor:	Users (Initiators)
Description:	The new user provides their required personal details, set password for login and get registered to the system.

Table 2: High Level Description for Register

4.1.2. Login

Use Case:	Login
Actor:	Users (Initiators)
Description:	The registered users provide their required details, entered their passwords and get login to the system. Verification process is carried out before the user gets login.

Table 3: High Level Description for Login

4.1.3. View Purchase

Use Case:	View Purchase
Actor:	Users (Initiators)
Description:	The user who has logged into the system has the option to review past purchase. The system fetches the inventories details and purchase details which is used to compare prices of the similar inventories and also get notification of the stock level while viewing the purchase using the view purchase function of the system.

Table 4: High Level Description for View Purchase

4.1.4. Generate Report

Use Case:	Generate report
Actor:	Users (Initiators)
Description:	Registered users can generate reports using the system's report generation function. The system database stores purchase, sales, and payment details, which are used to create reports. The purchase report helps evaluate new inventory acquisitions, while the sales report allows admins to analyze profit and loss. Users can also generate payment reports to review their past purchases and payment history.

Table 5: High Level Description for Generate Report

4.1.5. Create Purchase Order

Use Case:	Create Purchase Order
Actor:	Admin
Description:	The admin use creates purchase order function of the system and create a purchase order by selecting products, specifying quantities and confirming order details.

Table 6: High Level Description for Create Purchase Order

4.1.6. Dispatch Order

Use Case:	Dispatch Order
Actor:	Admin
Description:	This use case allows the admin to manage the sales operations by viewing the sales orders, retrieving customer information, which include contact and address details. It also enables the admin to monitor dispatching details along with sending notification for delay deliveries.

Table 7: High Level Description for Dispatch Order

4.1.7. Update Real Time Stock

Use Case:	Update Real Time Stock
Actor:	Admin
Description:	The admin uses Update Real Time Stock function to update stock level whenever inventory is added or reduced due to sales.

Table 8: High Level Description for Update Real Time Stock

4.1.8. Make Payment

Use Case:	Make payment
Actor:	Customers
Description:	The customers can make payment of their purchased inventories using the make payment function of the system. With each payment, the customer gets their bills which the system generates according to their purchase. Various payment methods are available for making payments. The

	details for each payment made is stored in the system payment database.

Table 9: High Level Description for Make Payment

4.2. Expanded Description

4.2.1. Register

Use Case: Register

Actors: Users (Initiators)

Description: The new user provides their required personal details, set password for login and get registered to the system.

Typical Course of Events:

Actor Action	System Response
1. The users select the “Register” option.	2. Displays the registration form to the new user.
3. The new user provides his/her personal details in the displayed registration form.	4. The system validates the accuracy of the provided information.
5. The new user submits the fully filled registration form.	6. System asks for confirmation to register the new user to the system.
7. The new user confirms the registration.	8. The details get stored in the system database.
	9. System directs the newly registered user to the login page.

Table 10: Expanded Use Case of Register

Alternative Courses:

Line 3: The new user fails to provide the personal details. Use case ends.

Line 5: The new user provides the invalid information. Request to recheck again and redirect to registration form.

Line 7: The user cancels to confirm the registration. Directs to registration forms again.

4.2.2. Make Payment

Use Case: Make Payment

Actors: Customers(initiators)

Description: The customers can make payment of their purchased inventories using the make payment function of the system. With each payment, the customer gets their bills which the system generates according to their purchase. Various payment methods are available for making payments. The details for each payment made is stored in the system payment database.

Typical Course of Events:

Actor Action	System Response
1. The customer selects the 'Make Payment' option.	2. Displays the list of all the ordered product that are unpaid and total amount due.
3. The customer reviews their purchase details.	4. Shows the lists of methods available for payment.
5. The customer selects the preferred payment method.	6. Validates the selected payment method and provides the field to enter payment details.
7. The customer enters the payment details and confirms the transactions.	8. Process the payment and update the payment database.
9. The customer generates the payment bill.	10. Displays the receipt.
11. The customer gets the notification.	

Table 11: Expanded Use Case of Make Payment

Alternative Courses:

Line 1: The customer fails to proceed the payment. Directed to payment option page.

Line 3: The customer is not satisfied with the purchase details and decided to cancel. Use case ends.

Line 5: The selected payment method is unavailable. Selects new payment method.

Line 7: The payment fails. Redirects to make payment page again using the retry option.

5. Communication Diagram

Communication diagram is a kind of interaction diagram that shows how the object interact. This diagram extends the object diagram and shows the objects including the message that travels for one to another (Paradigm, 2025).

Symbols used in Communication Diagram:

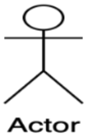
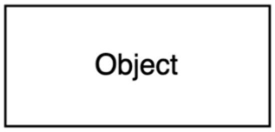
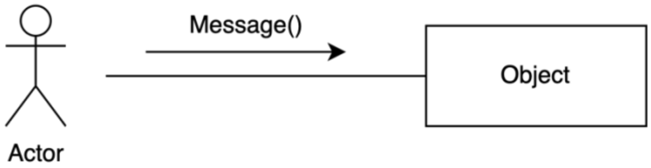
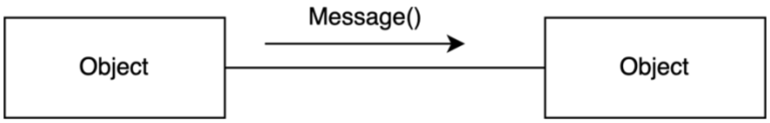
Symbols	Reference Name
	Actor
	Object
	Messages (communication between actor and object).
	Messages (communication between objects).

Table 12: Communication Diagram Symbols

The communication diagram for the use case “Make Payment” is given below.

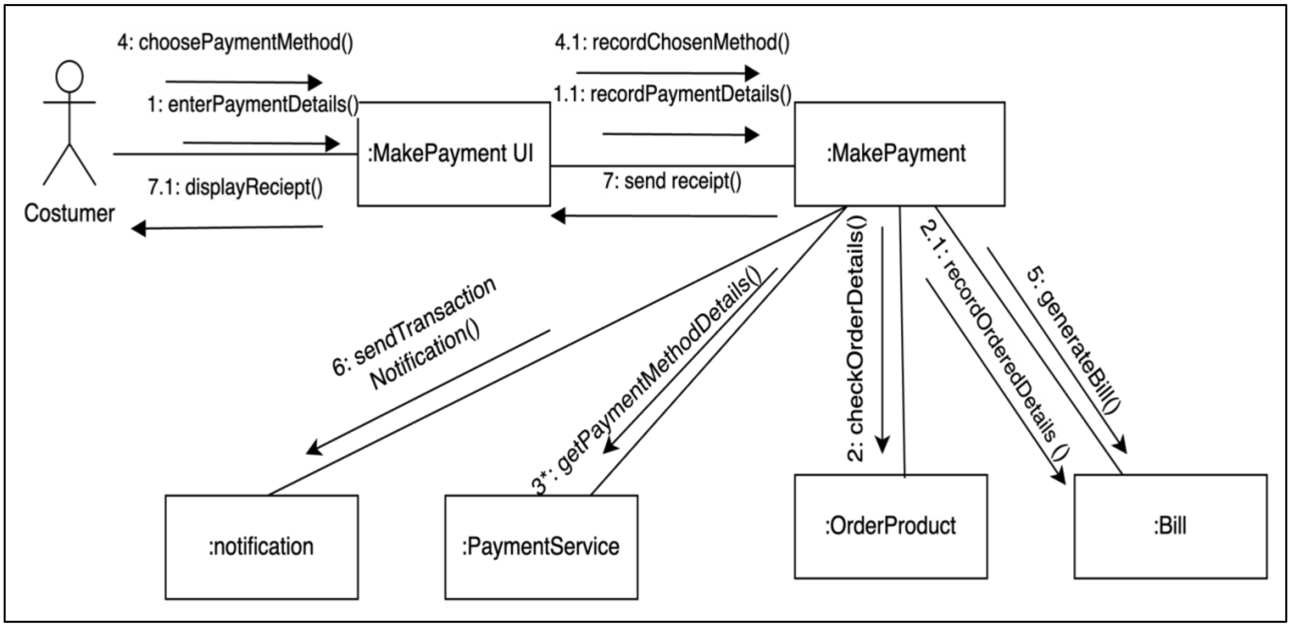


Figure 4: Communication Diagram(Make Payment)

The above communication diagram represents the process of making payment in the system. The object involved are MakePayment UI, MakePayment, notification, customer, PaymentMethod, OrderProduct and Bill.

- The Customer enters the payment details via the PaymentUI.
- The UI records the payment details.
- The main payment processor accesses the payment information. Then, the payment processor checks the order details.
- The payment processor then retrieves the payment method details as well.
- The customer selects the payment method and makes the payment which the payment processor recorded.
- The payment processor then generates the bill and send the transaction notification to the UI.
- The UI then displays the receipt to the customer.

6. Activity Diagram

Activity diagram is an important behavioral diagram that describe dynamic aspect of the system. This diagram is also a kind of flow chart with advanced version that models the flow from one activity to another (Paradigm, 2025).

Symbols used in Activity Diagram:

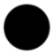
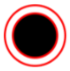
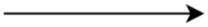
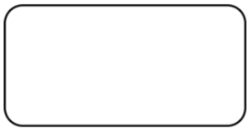
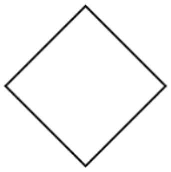
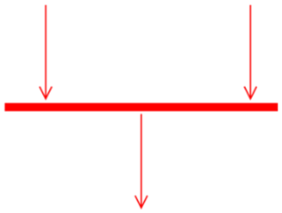
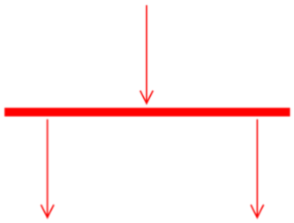
Symbols	Reference Name
	Start
	End
	Control Flow
	Action or Behavior
	Decision
	Join
	Fork

Table 13: Activity Diagram Symbols

The Activity Diagram for the use case “Register” is given below.

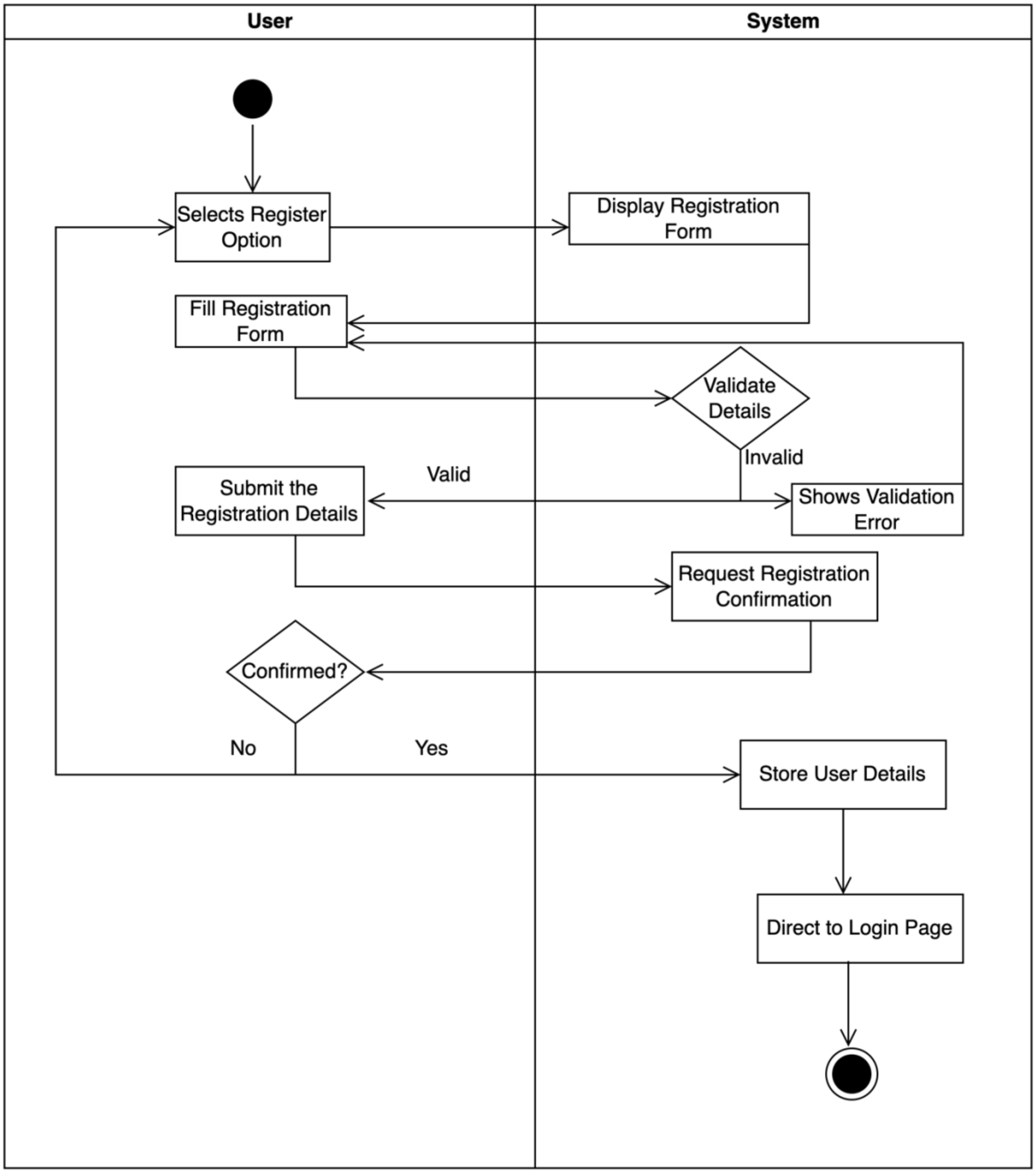


Figure 5: Activity Diagram (Register)

7. Class Diagram

Class diagrams is the structured UML diagram that comprises of attributes and functions. These is a tool that shows how a system is built. It represents system's structure by showing attributes, methods, classes and relationship between them. It helps the designers and developers of the project to understand how its components interact with each other and how the system is organized (GeeksforGeeks, 2025).

Symbols used in Class Diagram:

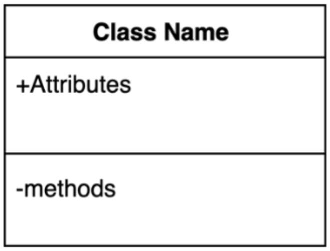
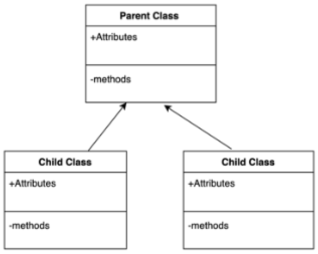
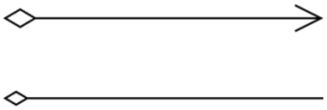
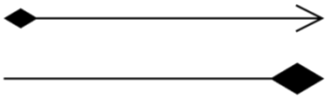
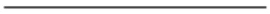
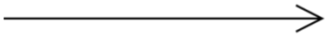
Symbols	Reference Name
	Class
 Parent Class +Attributes -methods Child Class +Attributes -methods Child Class +Attributes -methods	Inheritance
	Aggregation
	Composition
	Association
	Navigable/ Unidirectional Association

Table 14: Class Diagram Symbols

7.1. Table Comprising of Domain Classes from each Use Case

Use Cases	Domain Classes
Register	Admin, Customer, RegistrationService, Profile
Login	Customer, Admin, AuthenticationService
View Purchase	Customer, Admin, Order, OrderItem, Product
Generate Report	Admin, Report, PurchaseReport, SalesReport, PaymentReport
Create Purchase Order	Admin, Order, OrderItem ,Product, inventory
Dispatch Order	Admin, Order, Inventory, DeliveryService
Update Real Time Stock	Admin, Inventory, Product, StockNotification
Make Payment	Customer, Order, Payment

Table 15: Table Comprising of Domain Classes from each Use Case

7.2. Class Diagram

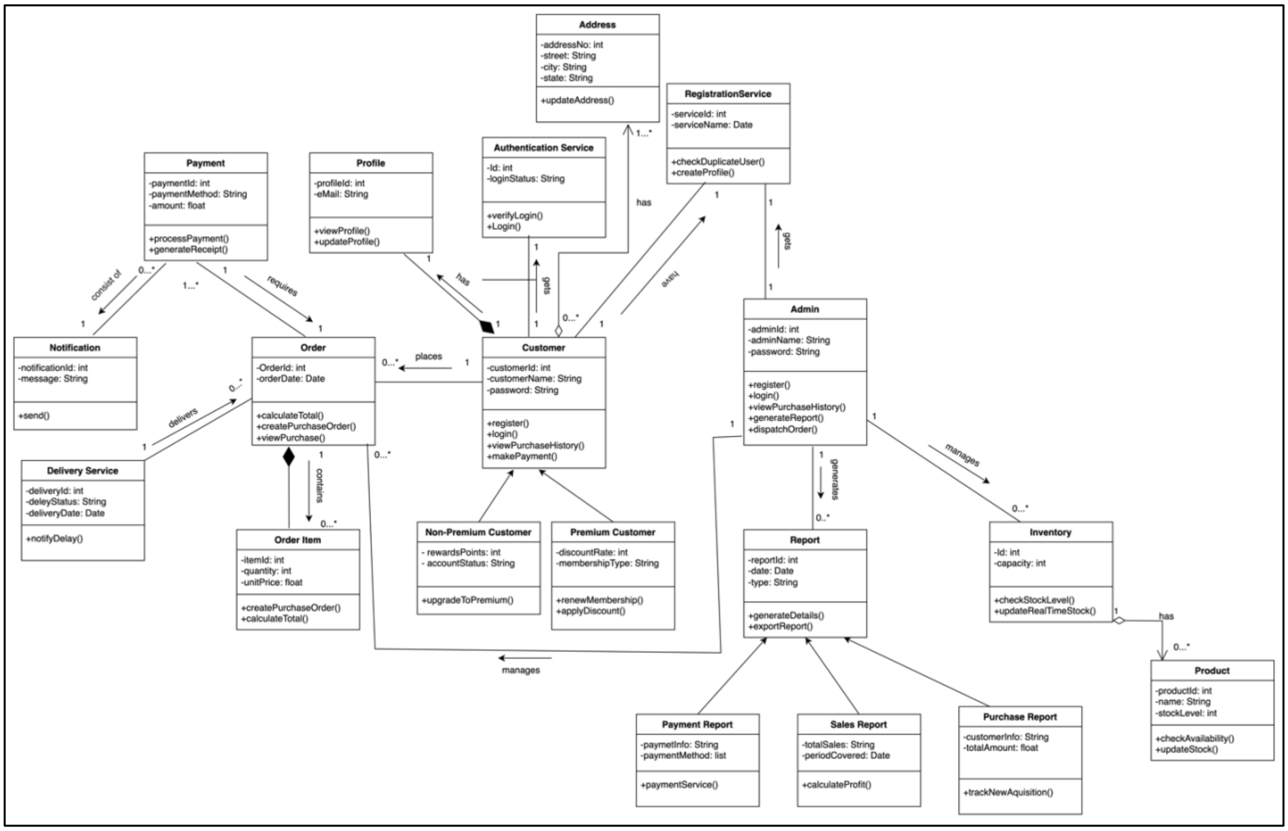


Figure 6: Class Diagram

8. Further Development

In the development phase, Global Tech Corporation worked on turning the idea of the Inventory Management System (IMS) into a real, easy-to-use digital platform to help with warehouse management. This stage focused on building the planned features in a way that was practical and user-friendly. To make sure the system could grow, be easy to fix, and run smoothly, the team designed a solid software structure, used helpful design patterns, and chose the right programming methods.

8.1. Design Pattern

The design of the Inventory Management System (IMS) for Global Tech Corporation follows the Iterative Waterfall Model. It was built using Iterative Waterfall Model because it divides the work into clear steps like gathering requirements, designing, coding, testing, deploying and maintaining. This method was picked because it helps to plan the system properly, work on each part carefully, and fix any problems before moving on to the next step. To build the working model, use case diagrams (both high-level and detailed), sequence diagrams, collaboration diagrams, and class diagrams were created. The design phase took place during the system design stage of the Iterative Waterfall Model, where the most suitable structures were chosen for development.

The Iterative Waterfall Model was used to design each function in the Inventory Management System by breaking the work into small and clear steps, and repeating the steps when needed. For user login and registration, the team first gathered what users needed like separate access for admin and customer, then designed the login forms, tested them, and fixed any issues before moving on. If anything was missing or needed changes, it went back to the design or requirement phase and improved it. For adding and viewing purchases, first the idea of how purchases should be added was planned, then the design was made to include supplier details and price comparison. After testing, small fixes were made and it was improved again before the next step. The same was done for report generation, where the types of reports needed by admin and customer were decided, then the report design and layout were created, tested, and improved again if required.

In sales management, the delivery details and customer information were first decided clearly. Then the parts were designed and tested step by step. For product management, the system was planned to add new products and check how much stock is left. These features were made one at a time and tested to make sure they worked well. In the payment part, things like keeping payment data safe and tracking it properly were planned and tested carefully. This method helped build each part slowly and clearly. If something was wrong or missing, it was fixed before moving forward.

8.2. Development and Testing

While developing and testing the Inventory Management System, each feature was built and checked step by step. In the login and registration part, roles like admin and customer which was first planned clearly, was built and tests were done to make sure both users could log in safely without any problems. In the purchase section, the steps to add product purchases from suppliers which was carefully planned was developed to let users add these details and compare prices of similar products. After building, it was tested to make sure all the data was saved correctly and displayed properly.

For report generation, the goal was to show sales and purchase information to help in decision making. This part was built slowly and tested to check if reports were showing the right totals and useful data. In sales management, details like delivery method, customer address, and contact info were added one by one. Each detail was tested to make sure nothing was missing and everything worked well. In product management, features to add and view products were developed in small steps. Tests were done at each step to check if stock levels and supplier details were correct.

The payment system built in parts and tested to make sure the payment connected correctly to the product and purchase data. At every stage of development, testing was done before moving to the next step. If anything went wrong or was missing, it was fixed immediately.

8.3. Deployment

The software application will be released to the client along with all necessary documentation, test records, and contracts during the deployment phase of the Iterative Waterfall Model. In this phase, regular reviews and post-development testing will be conducted. To ensure client satisfaction and collaboration, several techniques will be applied to refine the deployment process. These techniques include creating concepts that meet user needs, developing working prototypes, analyzing user requirements, allowing clients to test the prototype, gathering feedback, and making improvements based on the feedback received.

8.4. Future Changes

The Iterative Waterfall Model helps in making future changes or improvements to the system because each phase of the development process can be repeated when needed. This is especially useful when new requirements come up after the system is already working. For example, in the user login and registration function, the first step is to gather user requirements such as having separate logins for admin and customer and including secure features like two-factor authentication. If, in the future, there is a need to add new features like social media login, the process can go back to the requirement phase to include this new feature.

In the design step, the look and working of the login system are planned. If later the design is not easy to use or a new security feature is needed, it can be changed in the next round without affecting the whole system. In the coding step, the system is built in small parts, so new features can be added easily without breaking other parts.Testing is very important in this model. Every feature is checked carefully. If there are any problems or something needs to be better, it can be fixed and tested again. Even after the system is finished and being used, more changes or updates can be made by going through the steps again. This makes the system easy to improve over time without starting from scratch.

There are certain features in which changes can be made in the future to make the system better. the login and registration feature can be updated later by adding options like social media login or stronger security. The product list can be improved by showing more details like product pictures, prices, and stock levels. The report section can also be changed by adding new types of reports or using charts to make the data easier to understand. These changes can be made step by step using the Iterative Waterfall Model, without changing the whole system.

8.5. Maintenance

To keep the Inventory Management System (IMS) of Global Tech Corporation working properly, a simple and clear maintenance plan has been made. This plan includes regular system checks and updates to make sure everything runs smoothly. Backups are done on a fixed schedule so that important data is not lost, even if something goes wrong. Security steps like password protection, firewalls, and regular updates help keep the system safe from hackers and other online threats.

The system is built in a way that it can grow when more users join, so it doesn’t slow down. Feedback from users is collected using surveys and simple questions, which helps to find out what can be improved. If needed, old hardware parts are replaced with new ones to keep the system fast and working well. A clear and easy-to-understand user manual is given to help people use the system without confusion. Also, regular security updates are done to fix any problems and keep the system protected.

9. Prototypes

A prototype is a basic model or early version of a product made to test an idea. It helps people see how the product might work and find any problems before building the final version. Prototypes are very useful for checking if a design is good and if users like it (Ramírez, 2018).

9.1. Register Page

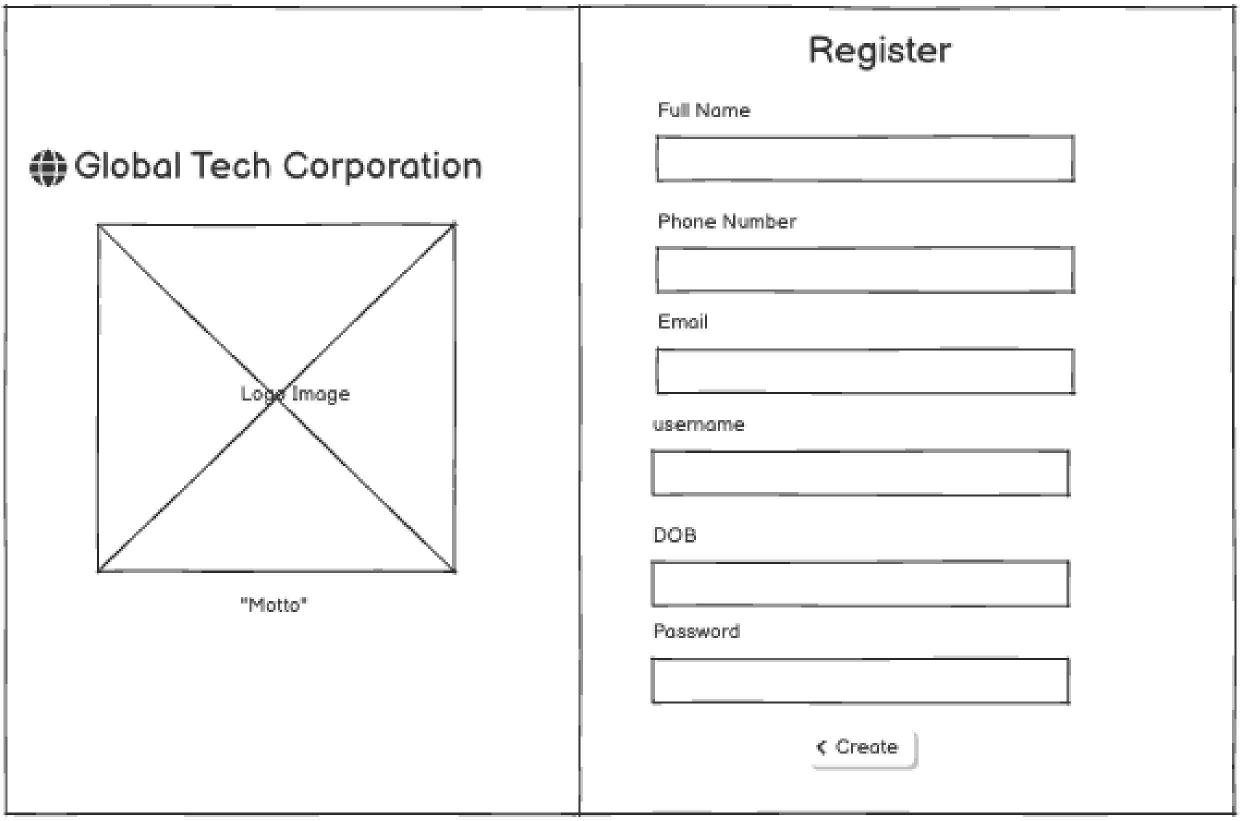


Figure 7: Registration Page

9.2. Login Page

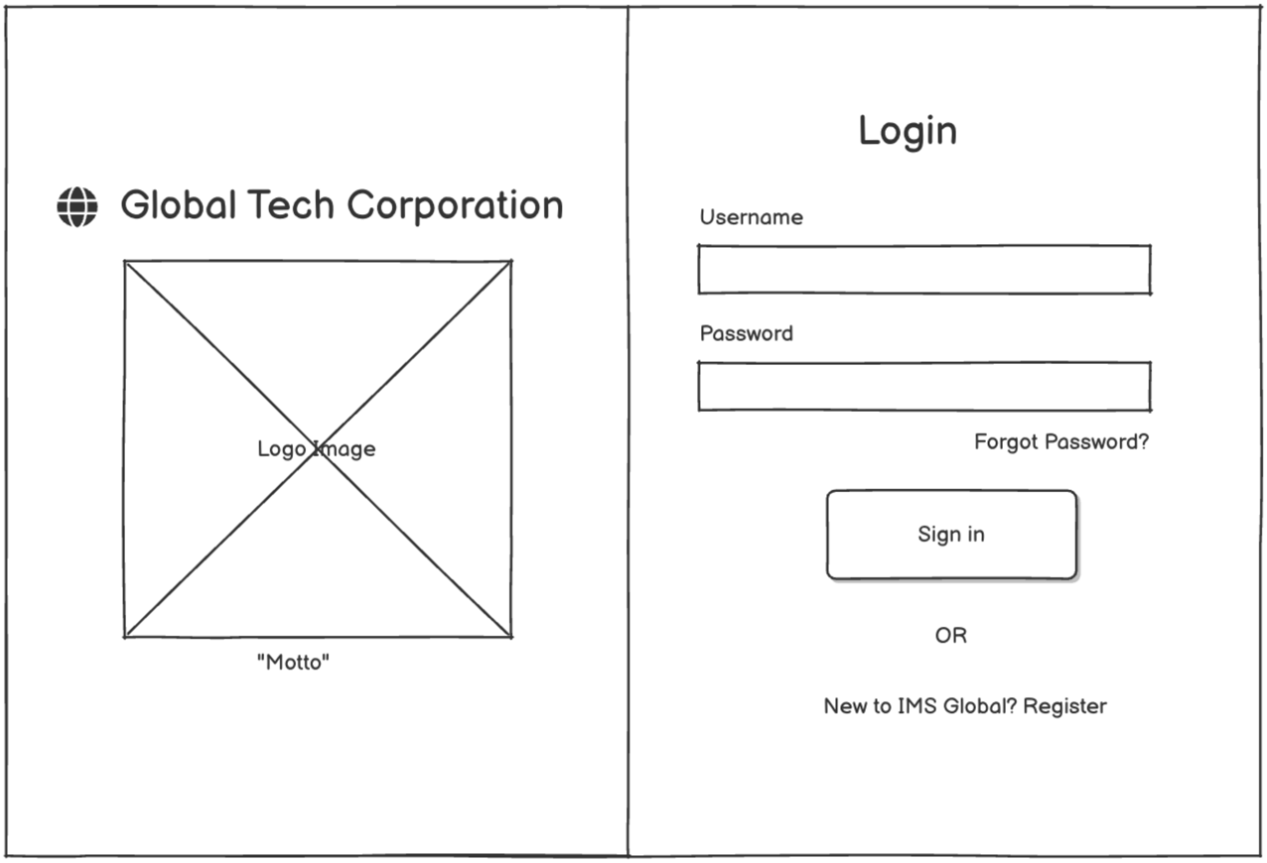
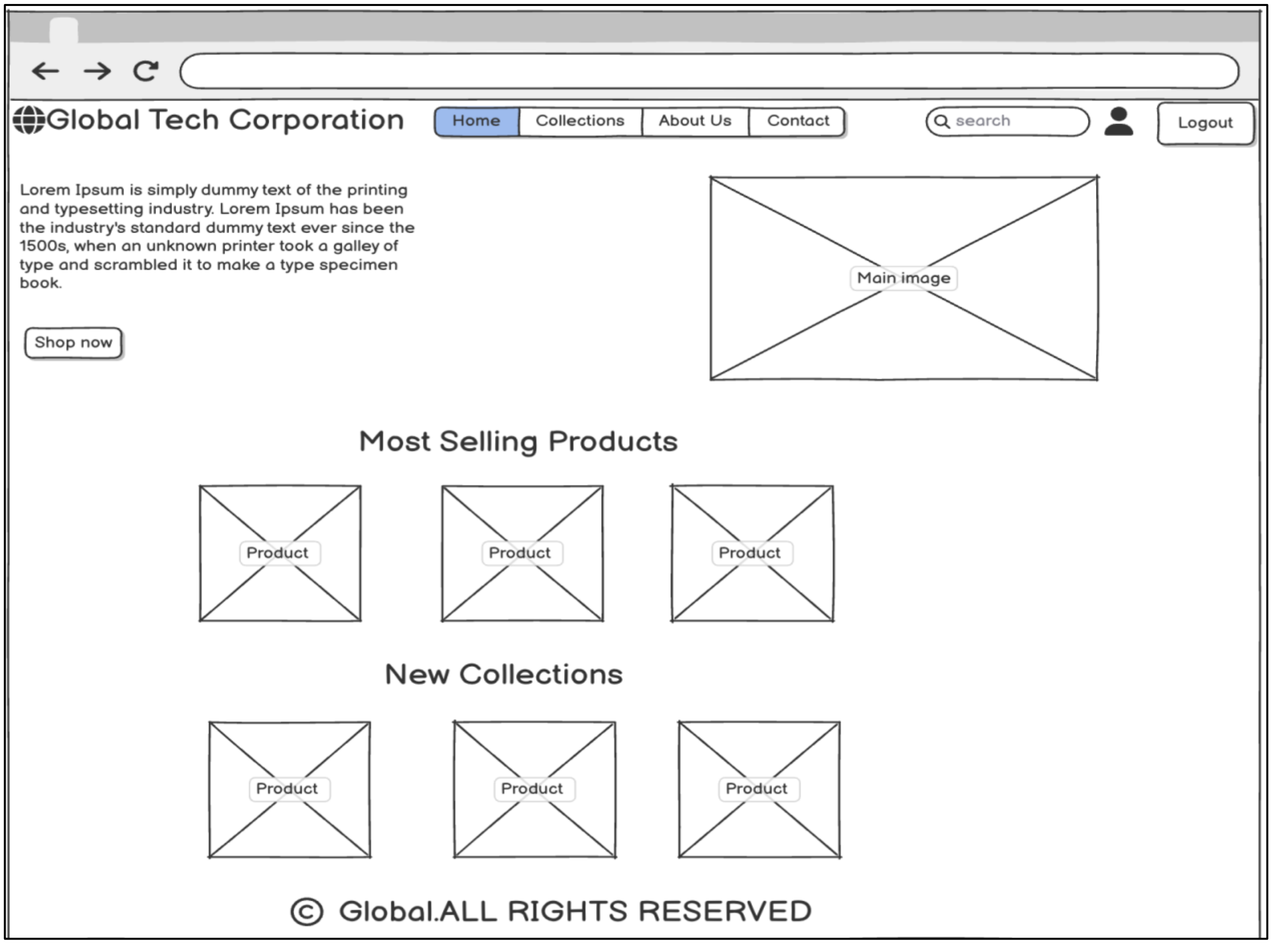


Figure 8: Login Page

9.3. Home Page



Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Figure 9: Home Page

9.4. Contact Us Page

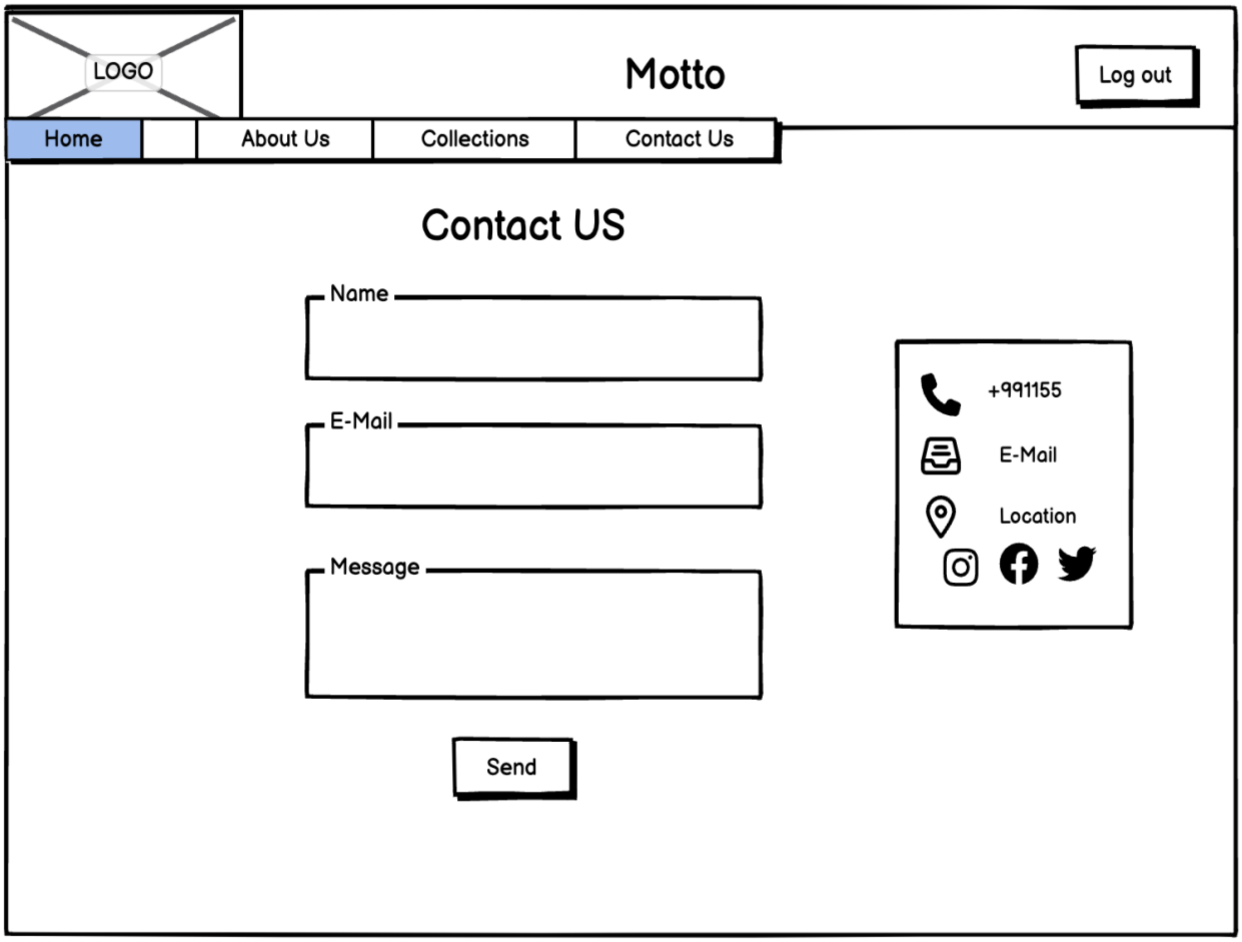


Figure 10: Contact Us Page

9.5. Create Purchase Order Page

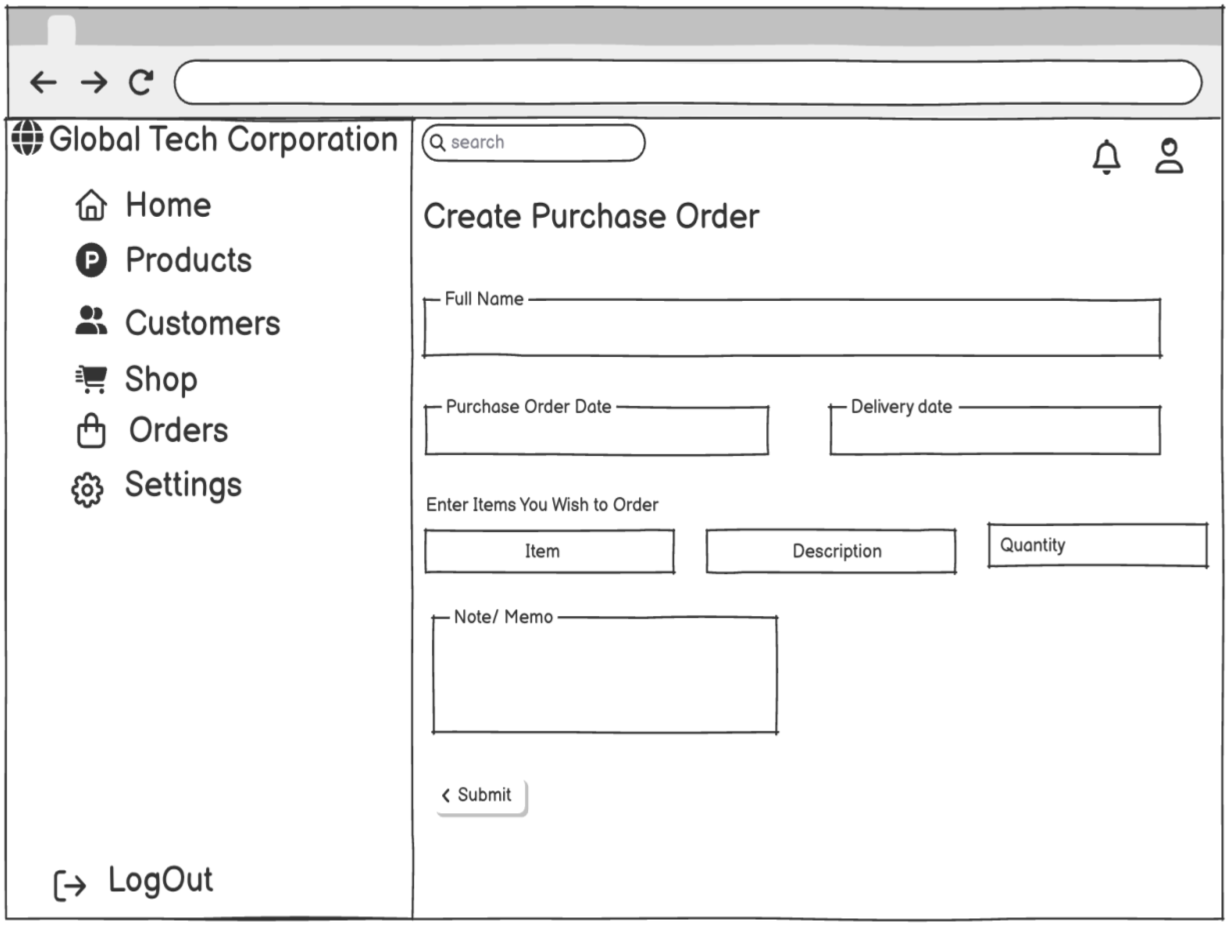
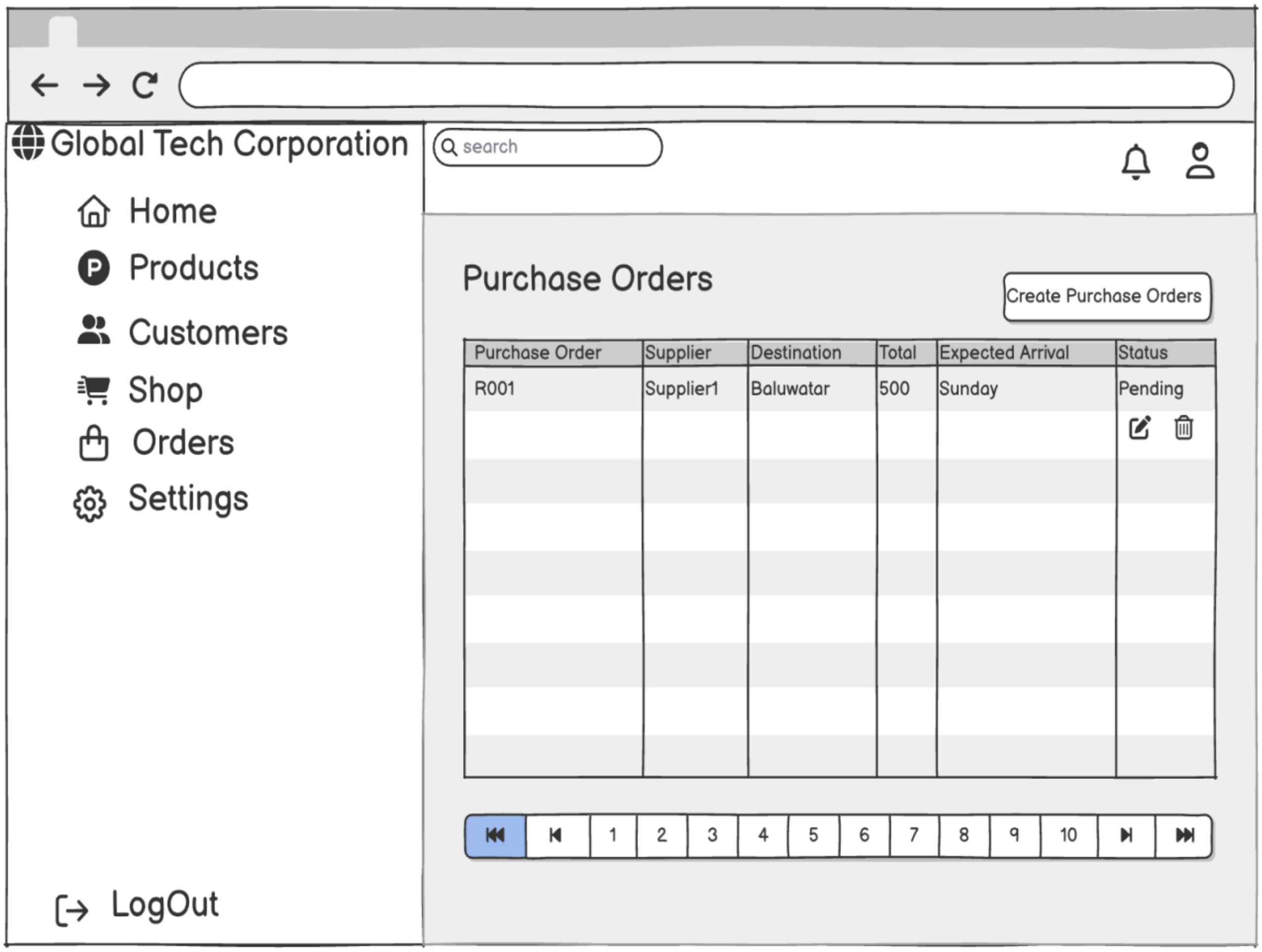


Figure 11: Create Purchase Order Page

9.6. Purchase Order Details Page



Purchase Order	Supplier	Destination	Total	Expected Arrival	Status
R001	Supplier1	Baluwatar	500	Sunday	Pending

Figure 12: Purchase Order Details Page

9.7. Admin Dashboard Page

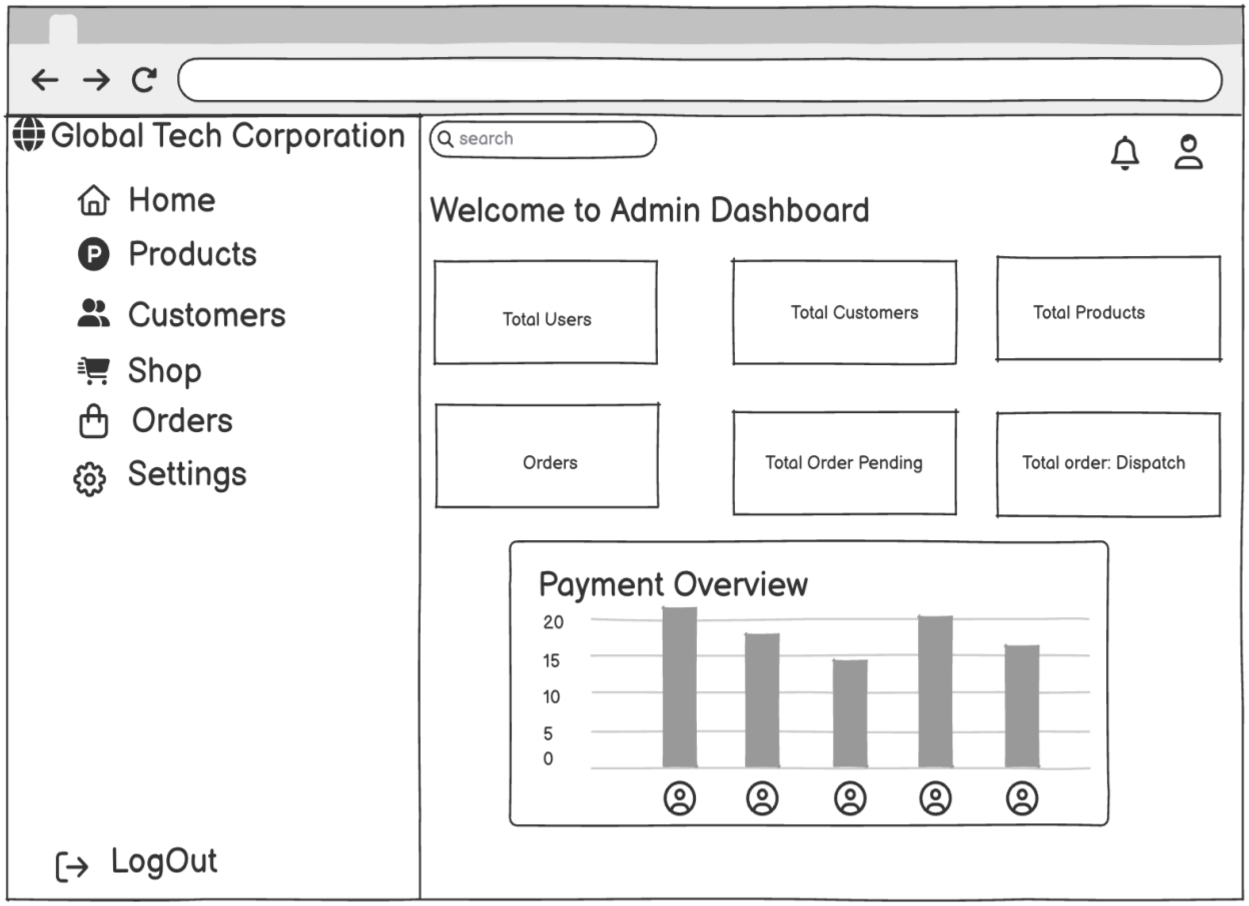
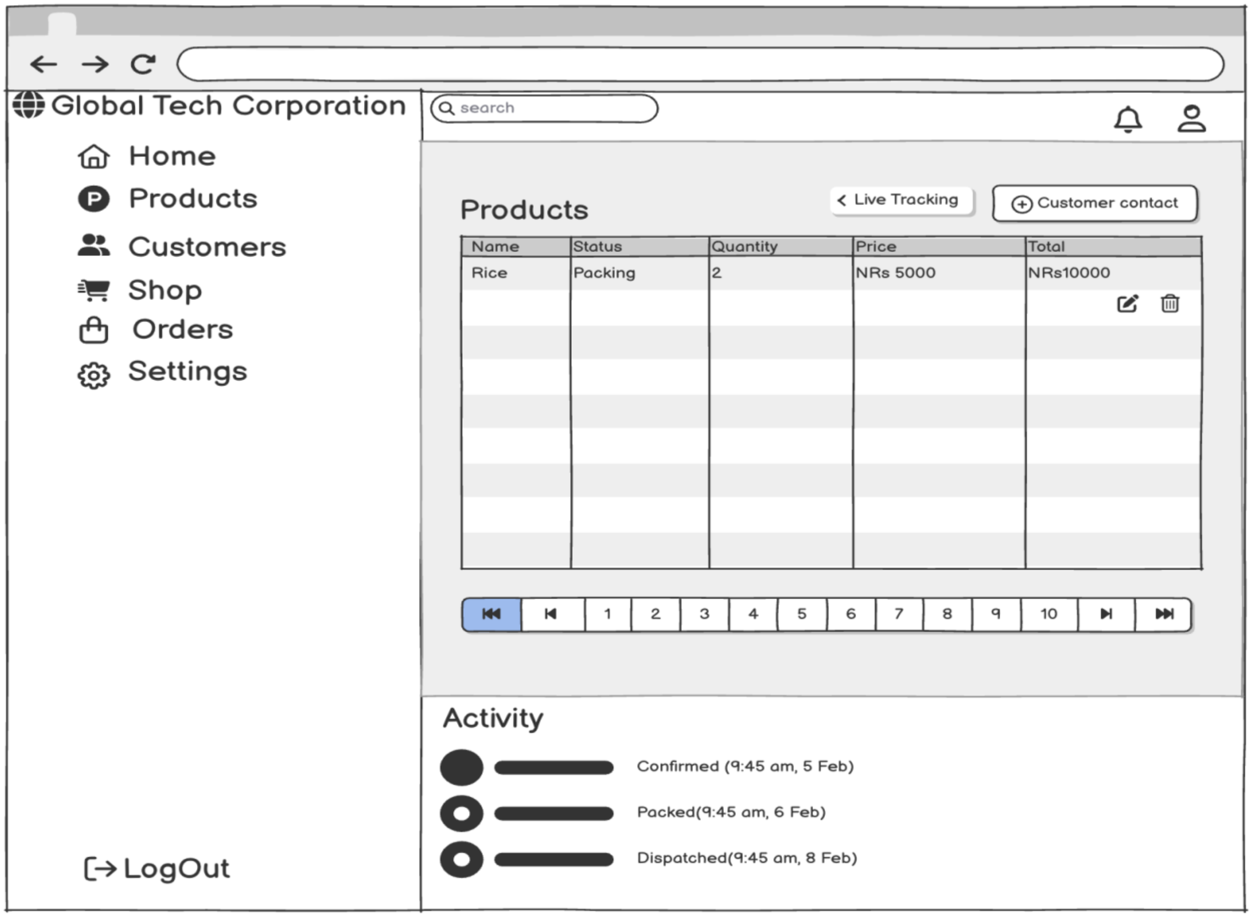


Figure 13: Admin Dashboard Page

9.8. Dispatch Order Page



Name	Status	Quantity	Price	Total
Rice	Packing	2	NRs 5000	NRs10000

Confirmed (9:45 am, 5 Feb)

Packed(9:45 am, 6 Feb)

Dispatched(9:45 am, 8 Feb)

Figure 14: Dispatch Order Page

9.9. Product Page

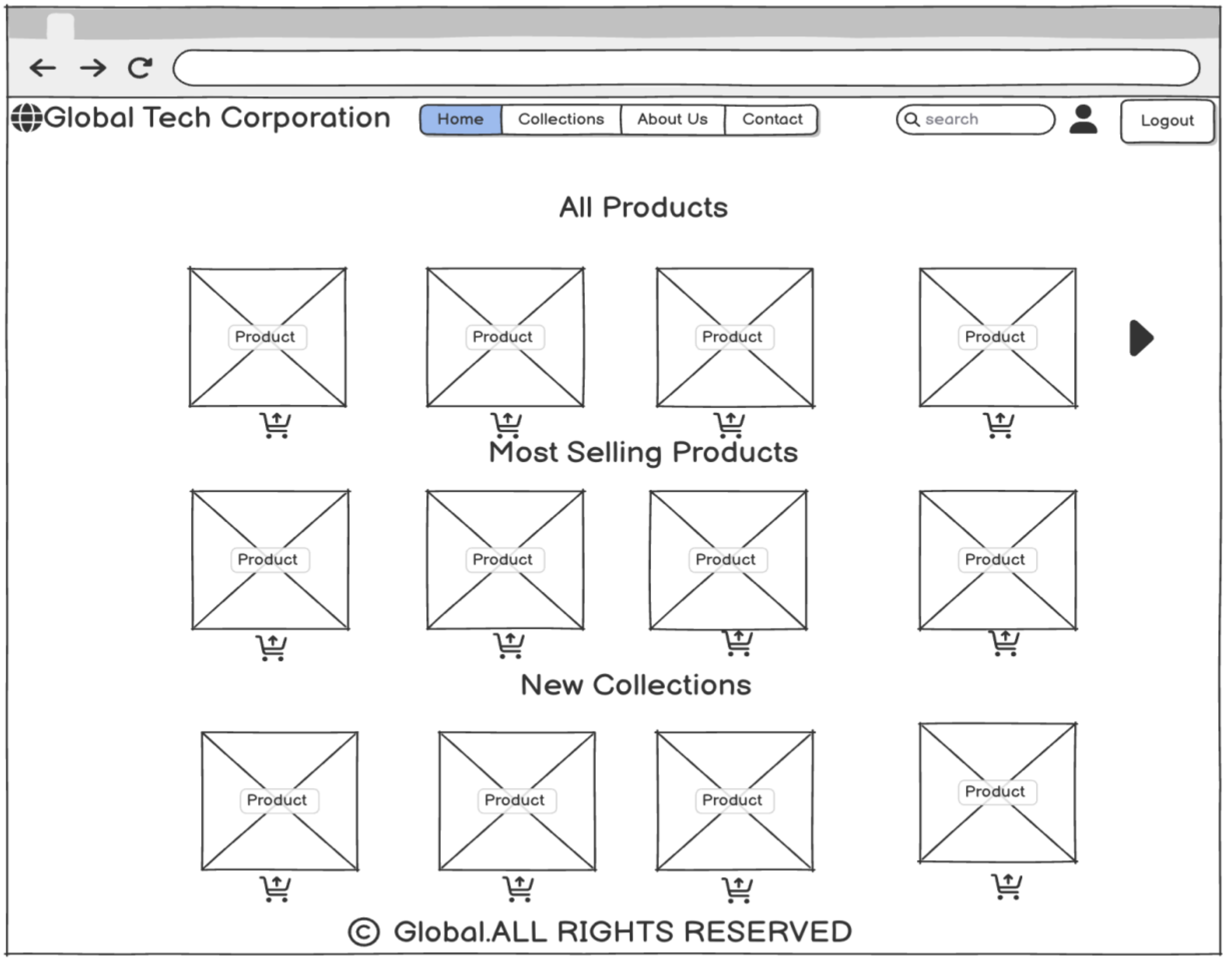
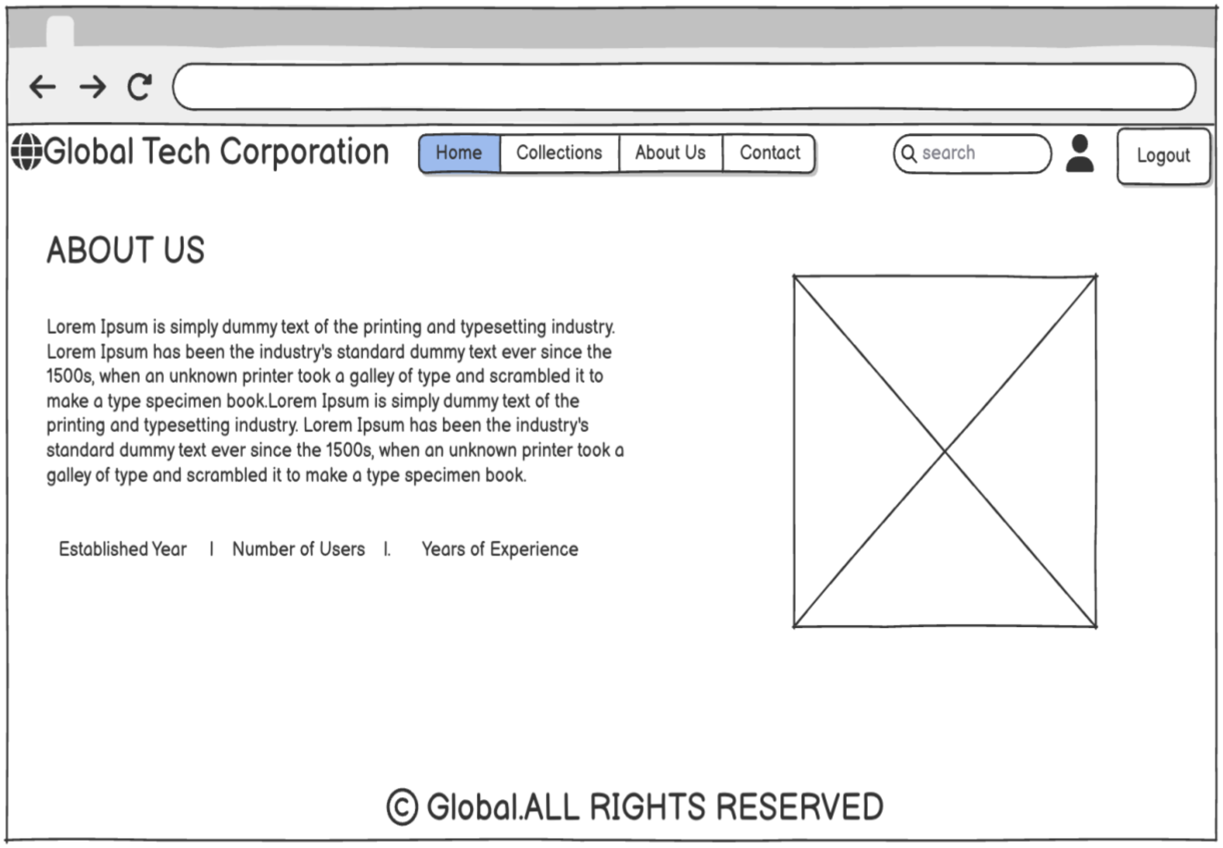


Figure 15: Product Page

9.10. About Us Page



ABOUT US

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Established Year | Number of Users l. Years of Experience

© Global.ALL RIGHTS RESERVED

Figure 16: About Us Page

9.11. Track Order Page (Customer)

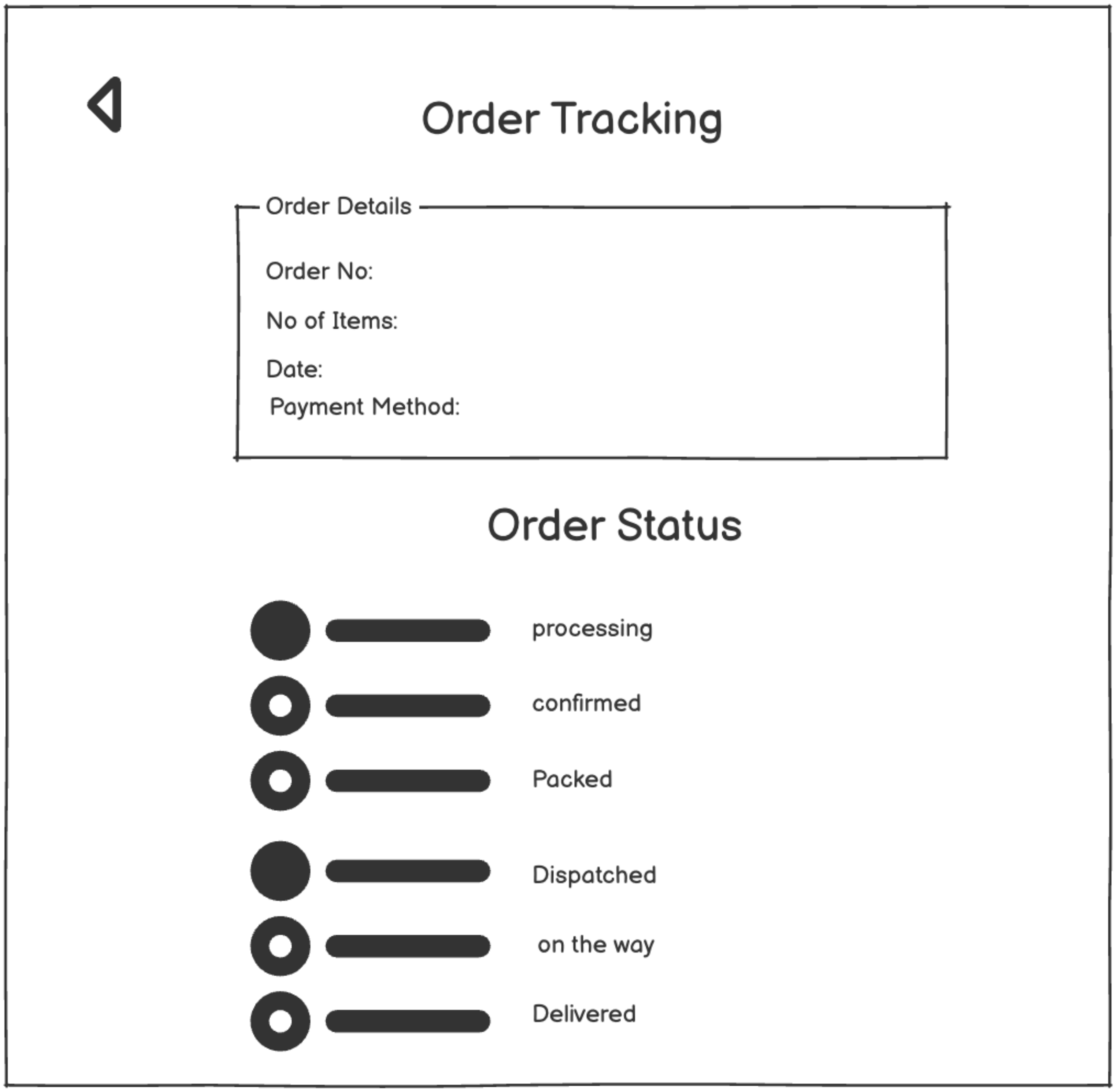


Figure 17: Track Order Page

9.12. Update Stock Page

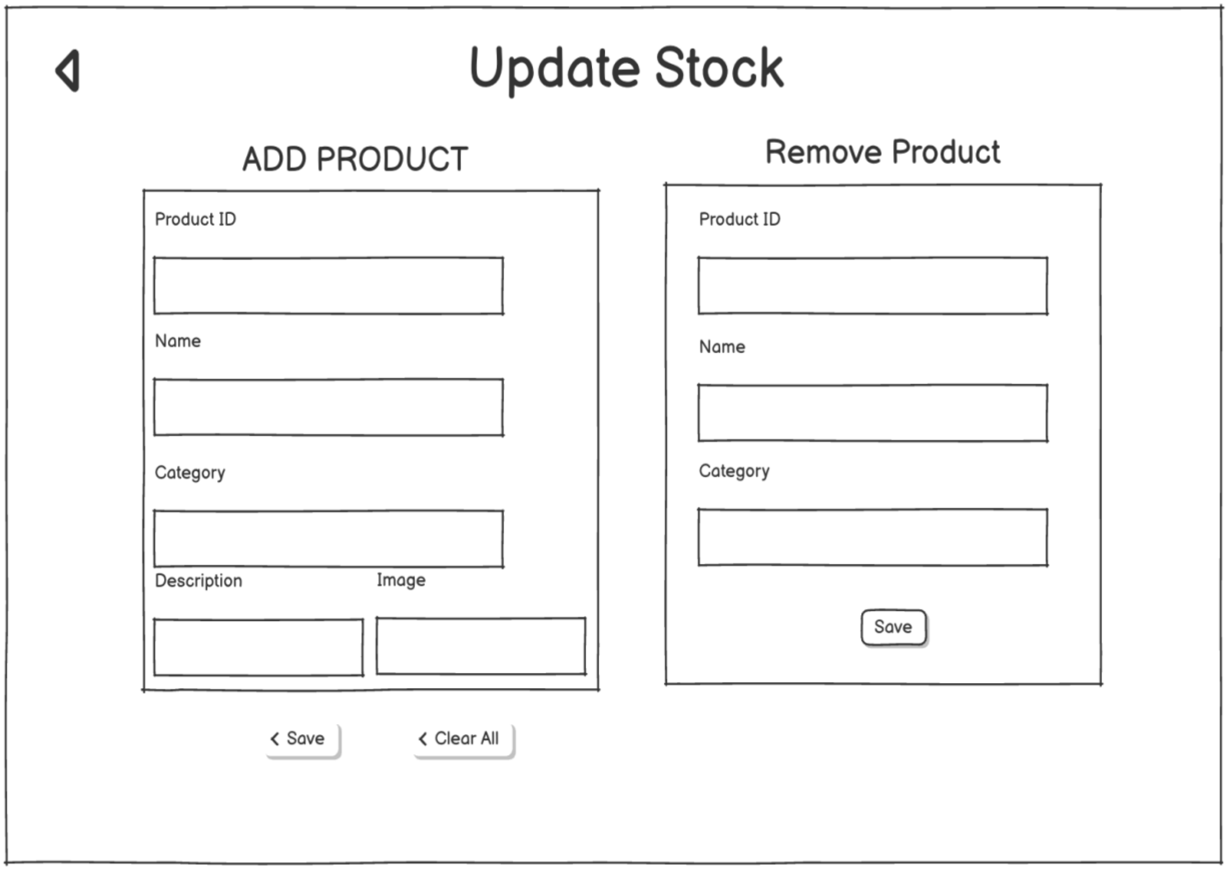


Figure 18: Update Stock Page

9.13. Update Profile Page

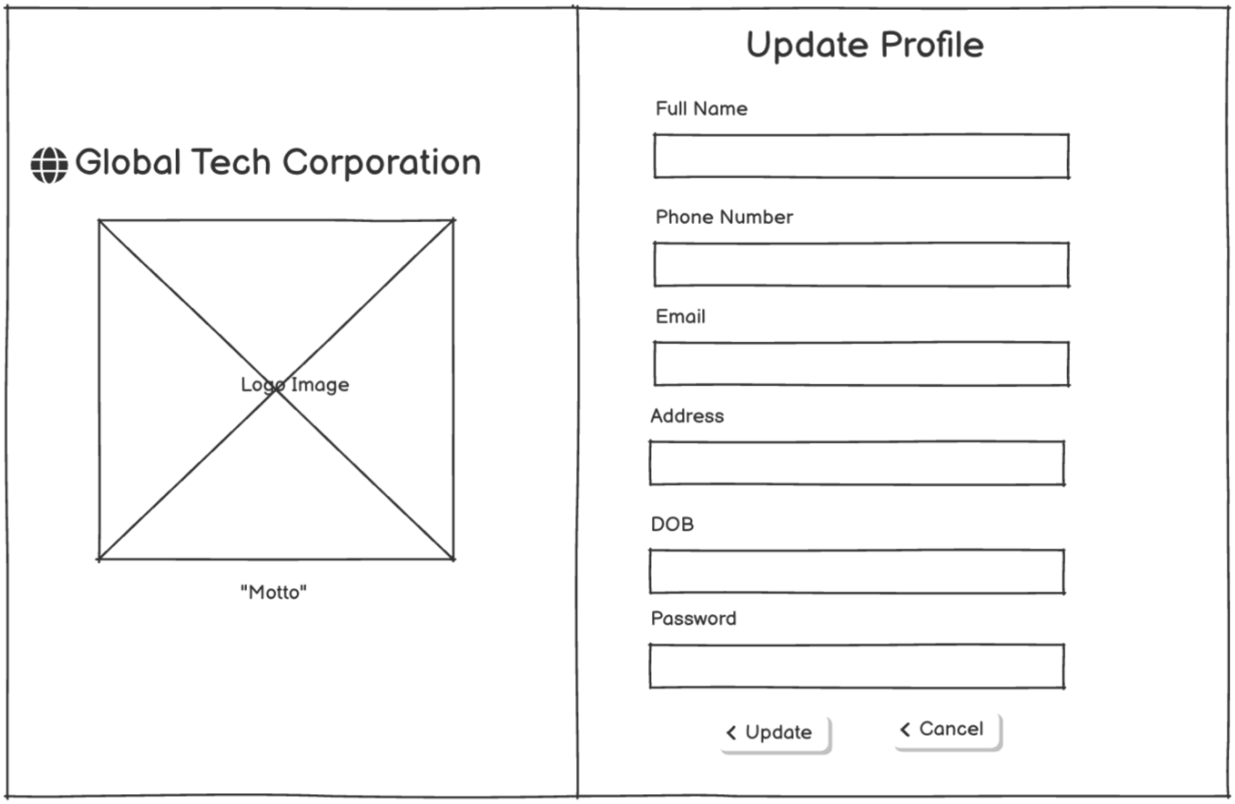
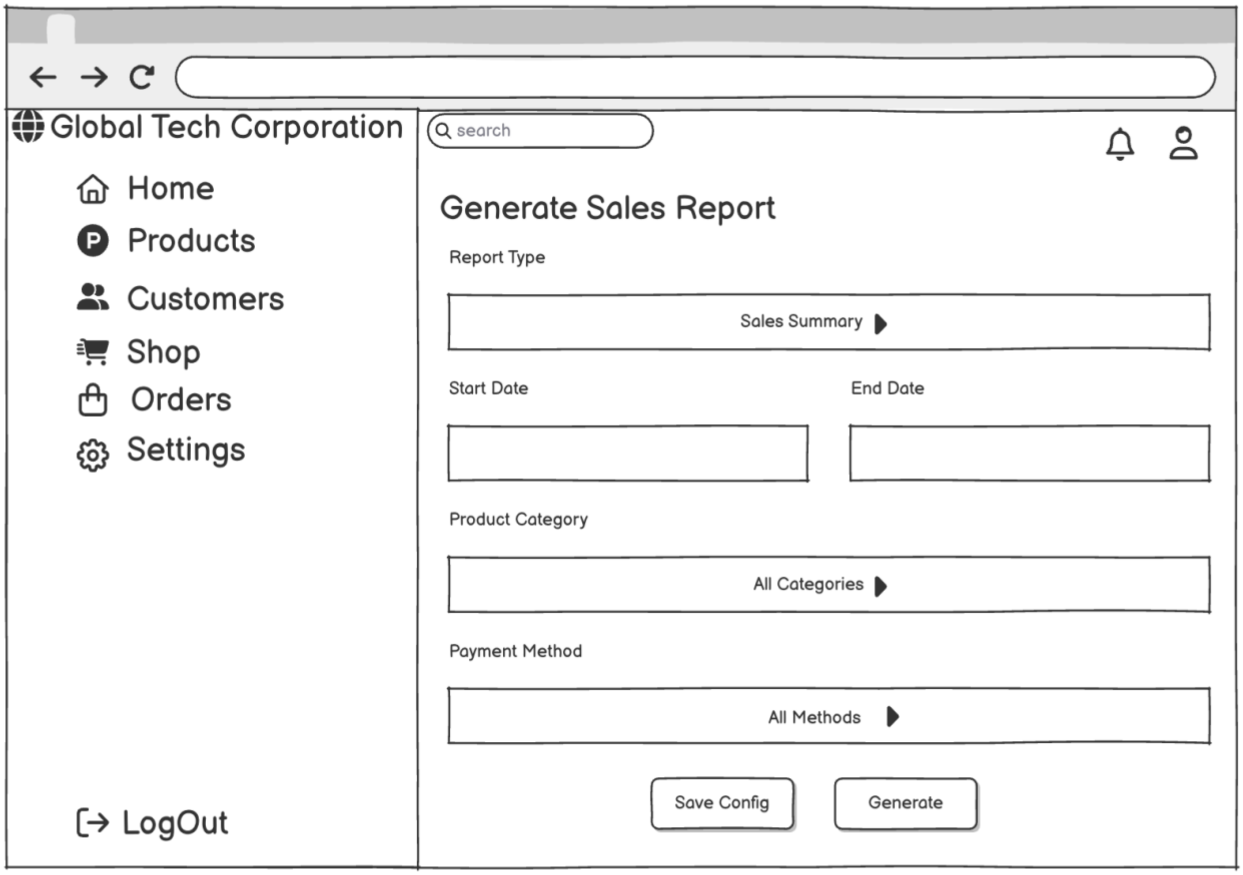


Figure 19: Update Profile Page

9.14. Generate Report Page



Generate Sales Report

Report Type

Sales Summary

Start Date

End Date

Product Category

All Categories

Payment Method

All Methods

Figure 20: Generate Report Page

9.15. Make Payment Page

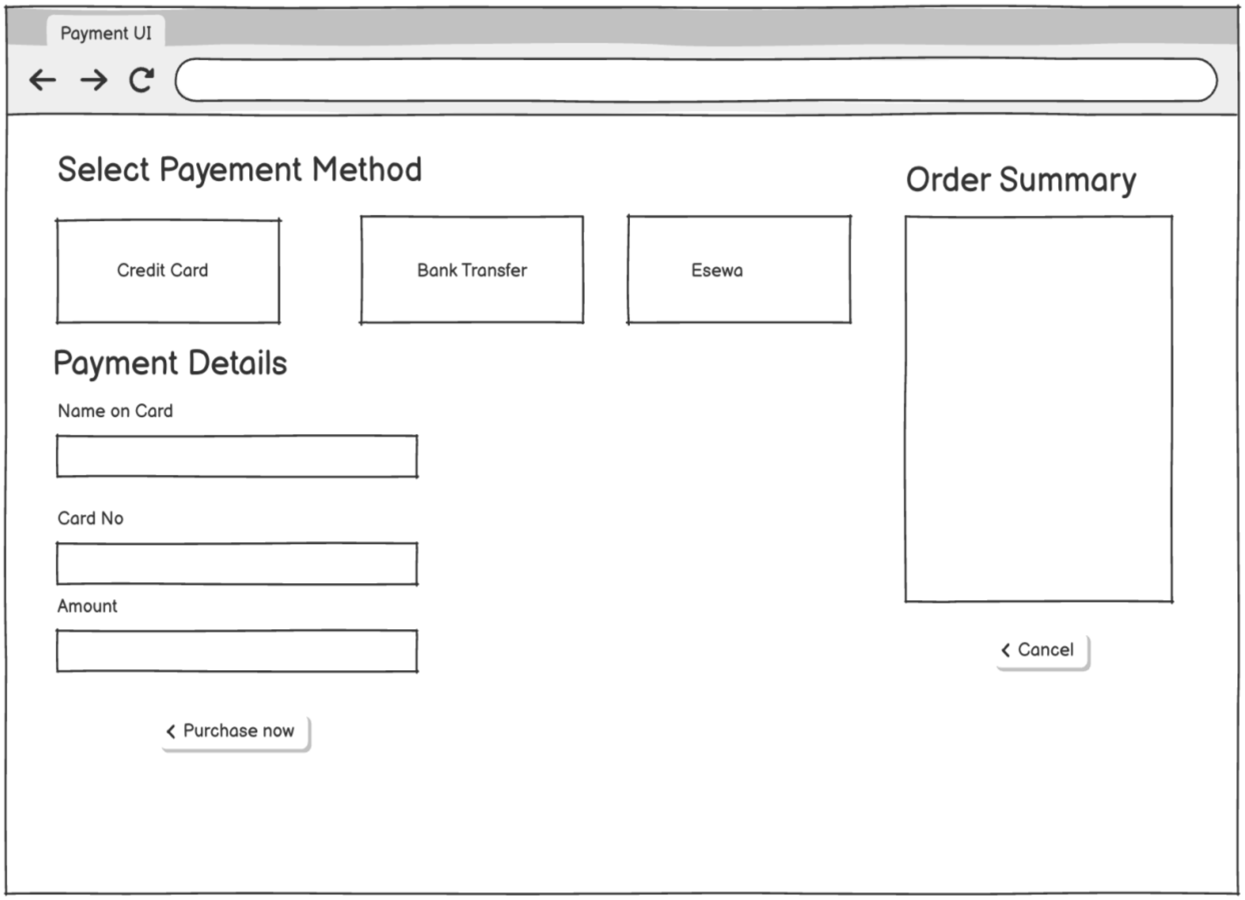


Figure 21: Make payment Page

10. Conclusion

In this project for Global Tech Corporation, an ambitious initiative was undertaken to develop a next-generation automated infrastructure system to enhance warehouse management operations. The primary objective was implementing a next-generation Inventory Management System (IMS) to streamline operations and improve efficiency thorough planning, utilized Gantt charts and UML diagrams to outline project phases and visualize system architecture. The Iterative Waterfall Model was chosen as the development methodology, allowing structured progression through requirements gathering, design, implementation, testing, deployment, and maintenance while enabling iterative refinements. Prototyping and robust testing strategies ensured a high-quality, scalable solution. Despite challenges, the project successfully delivered a reliable IMS with key functionalities like role-based access, purchase order management, reporting tools, and secure payment integration.

Careful planning and clear requirement analysis were key to the project's success. Tools like Work Breakdown Structures (WBS) and Gantt charts were used to set the project’s scope and schedule, while use case and collaboration diagrams helped clearly define system functions. The Iterative Waterfall Model shaped important choices about system design, structure, and deployment. Regular testing and building prototypes allowed for continuous improvements. The final IMS turned out to be a user-friendly and scalable system, showing how well-structured methods can succeed in real-world projects. This project also highlighted the importance of being flexible and planning carefully when working on complex technical solutions.

11. References

Adobe.com, 2024. What is a business case. [Online]
Available at: https://business.adobe.com/blog/basics/business-case
[Accessed 14 December 2024].

Wiegers, K. E., 1999. Software Requirements Specification. [Online]
Available at: https://dspmuranchi.ac.in/pdf/Blog/srs_template-ieee.pdf
[Accessed 15 December 2024].

Workbreakdownstructure.com., 2025. What is a Work Breakdown Structure?. [Online]
Available at: https://www.workbreakdownstructure.com/
[Accessed 6 March 2025].

ProjectManager.com, I., 2025. What Is a Gantt Chart? Definition & Meaning. [Online]
Available at: https://www.projectmanager.com/guides/gantt-chart
[Accessed 8 March 2025].

Paradigm, V., 2025. What is Use Case Diagram?. [Online]
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/
[Accessed 7 March 2025].

Paradigm, V., 2025. What is Communication Diagram?. [Online]
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-communication-diagram/
[Accessed 27 March 2025].

Paradigm, V., 2025. What is Activity Diagram?. [Online]
Available at: http://visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/
[Accessed 27 March 2025].

GeeksforGeeks, S. E. P. L., 2025. Class Diagram | Unified Modeling Language (UML). [Online]
Available at: https://www.geeksforgeeks.org/unified-modeling-language-uml-

class-diagrams/
[Accessed 10 may 2025].

Ramírez, V., 2018. What is a Prototype?. [Online]
Available at: https://medium.com/nyc-design/what-is-a-prototype-924ff9400cfd
[Accessed 10 May 2025].