CS5002NI Software Engineering

MAIN-SIT 35% Coursework

AY 2024-25

Credit: 30

Student Name: Arun Nagarkoti

London Met ID: 23050194

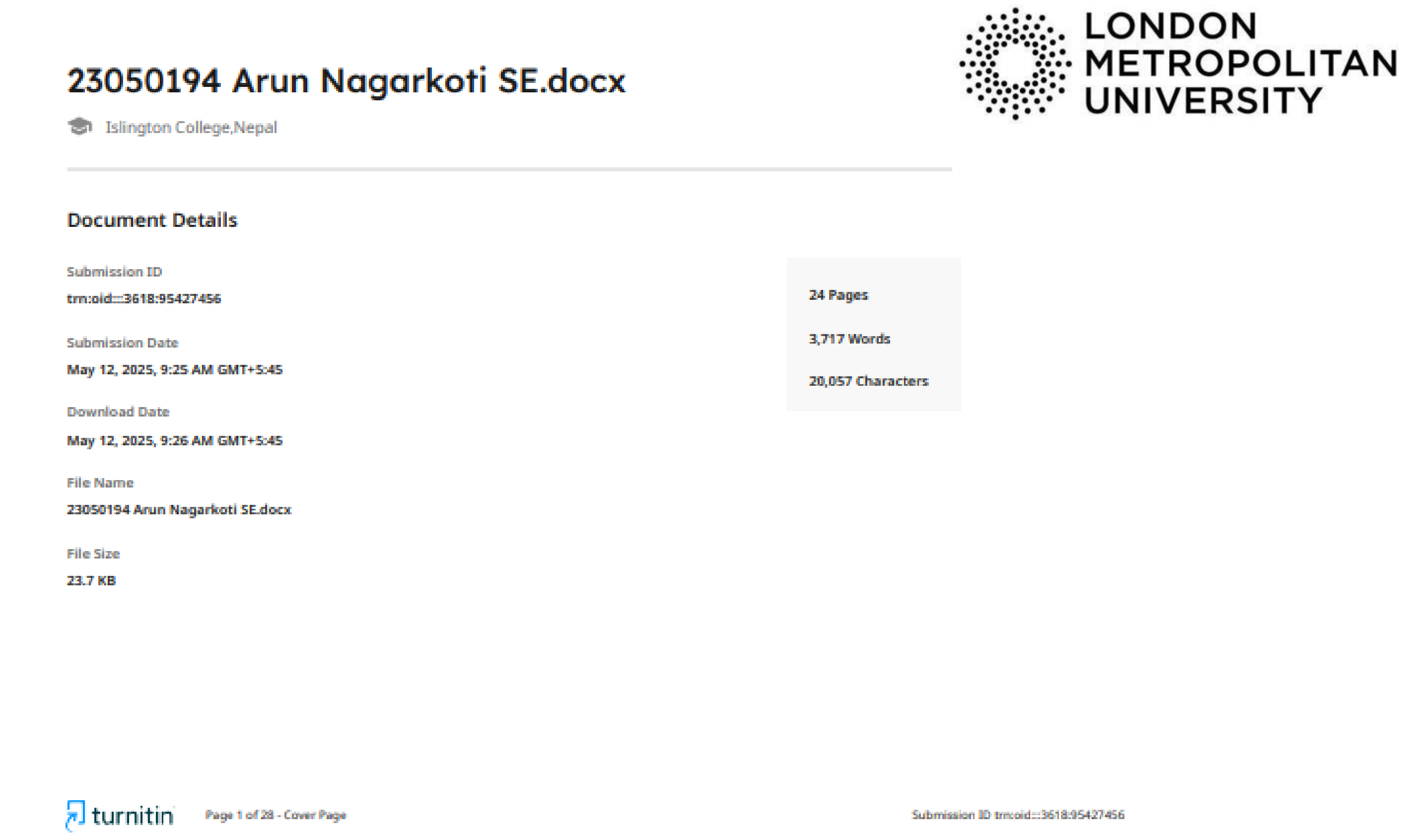
College ID: np01cp4a230252

Assignment Due Date: May 12, 2025

Assignment Submission Date: May 12, 2025

Word Count: 3717

I confirm that I understand my coursework needs to be submitted online via Google 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.



LONDON METROPOLITAN UNIVERSITY

23050194 Arun Nagarkoti SE.docx

Islington College,Nepal

Document Details

Submission ID

trn:oid:::3618:95427456

Submission Date

May 12, 2025, 9:25 AM GMT+5:45

Download Date

May 12, 2025, 9:26 AM GMT+5:45

File Name

23050194 Arun Nagarkoti SE.docx

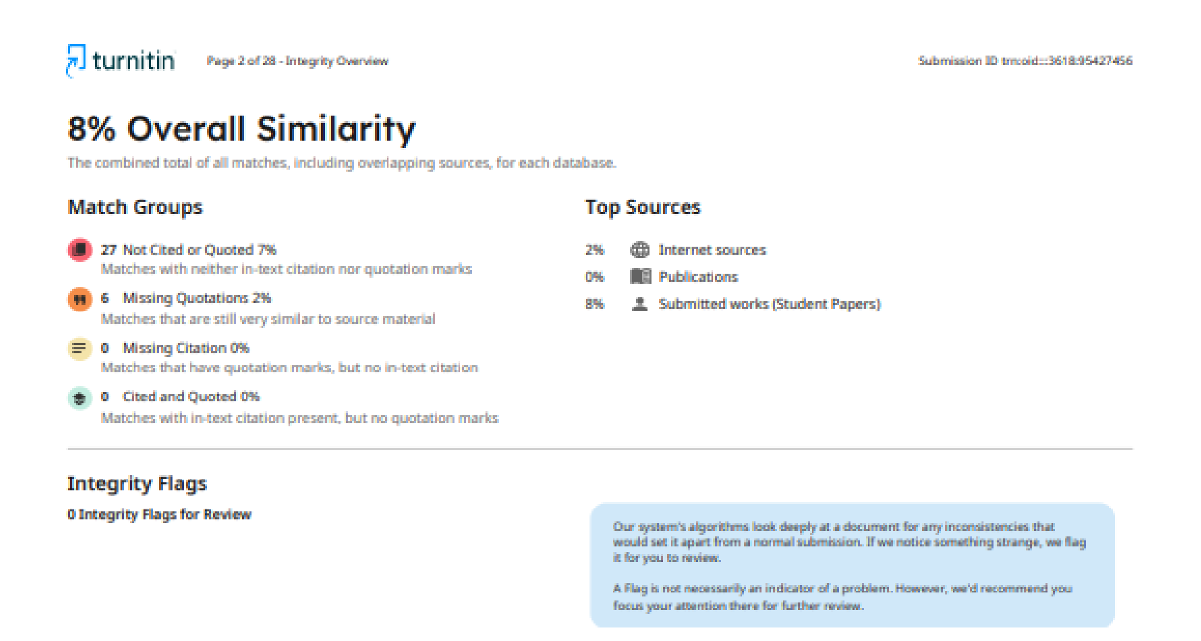
File Size

23.7 KB

24 Pages

3,717 Words

20,057 Characters



8% Overall Similarity

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

Match Groups

- 27 Not Cited or Quoted 7%
  Matches with neither in-text citation nor quotation marks
- 6 Missing Quotations 2%
  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

2%	Internet sources
0%	Publications
8%	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 table

1. Introduction

The Global tech Corporation approached us to create a high-end automated warehouse system management in Nepal. IT strategy is one of them, and quite a bold one at to automate with an IMS (Inventory Management System) process and make improvements. The challenges through the project encountered on the due to the system design being flawed or not well analyzed. Consequently, the company was left with large financial problems, angry customers, and hefty fines on delivery of product. After knowing this limitation, the firm thought a more formal approach was required.

Object-Oriented Analysis and Design (OOAD) was used to provide it a well-structured system with complete necessary guidelines. CRQ (comprehensive report questionnaire) was initiated at the start of this project for effective analysis of user requirements (i.e. All analysis was necessary to meet user needs and expectations. This led to the generation of a work breakdown structure to ensure a design that is more user friendly. Prototyping of the system was done before the coding phase to show part of the system architecture that was represented. After the design was quickly finalized and development was proceeding fast, with extensive testing being done. Regular post-implementation maintenance was performed so as to optimize the system usage and higher the user satisfaction. During the entire process, we leveraged on few really important project tools, services, Word for docs, Turnitin to verify citation originality, and Draw.io for figures. Team Gantt assisted in managing and overseeing projects and in developing schedules that were followed properly. Also, messages diagrams and action sequences were designed to demonstrate interactions between system components and system.

The overall System is to help the corporation to assist with their automation of work done and increase overall efficiency and productivity of the corporation.

2. Work break down system:

WBS which stands for Work breakdown structure acts as a transporter for dividing and breaking an engineering project into sub projects and subtasks as well as work packages and many more. WBS is plays a vital role of linking the objectives and goals with available resources and activities in a logical framework (Tausworthe, 1980).

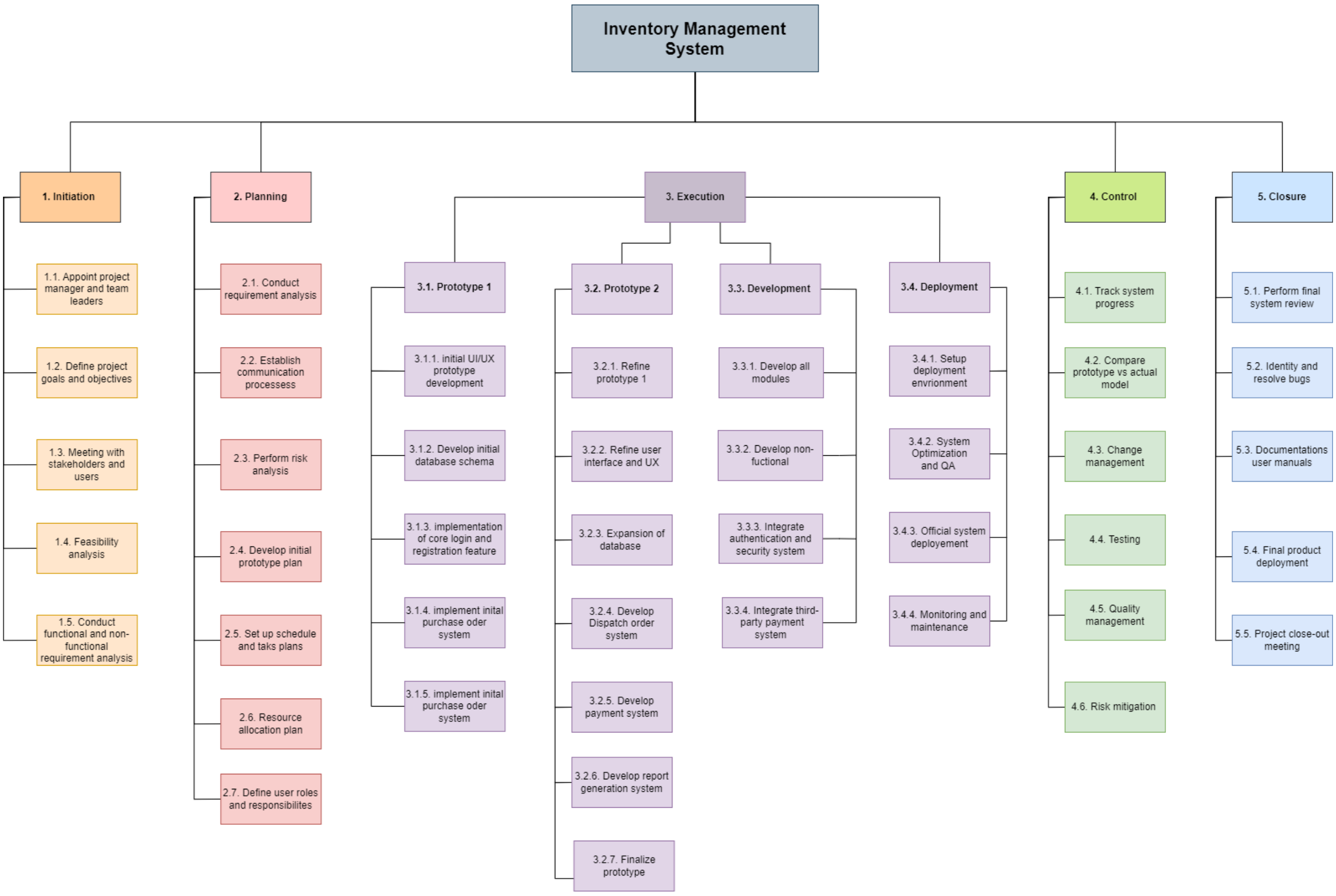

Inventory Management System
1. Initiation
1.1. Appoint project manager and team leaders
1.2. Define project goals and objectives
1.3. Meeting with stakeholders and users
1.4. Feasibility analysis
1.5. Conduct functional and non-functional requirement analysis
2. Planning
2.1. Conduct requirement analysis
2.2. Establish communication processess
2.3. Perform risk analysis
2.4. Develop initial prototype plan
2.5. Set up schedule and taks plans
2.6. Resource allocation plan
2.7. Define user roles and responsibilites
3. Execution
3.1. Prototype 1
3.1.1. initial UI/UX prototype development
3.1.2. Develop initial database schema
3.1.3. implementation of core login and registration feature
3.1.4. implement inital purchase oder system
3.1.5. implement inital purchase oder system
3.2. Prototype 2
3.2.1. Refine prototype 1
3.2.2. Refine user interface and UX
3.2.3. Expansion of database
3.2.4. Develop Dispatch order system
3.2.5. Develop payment system
3.2.6. Develop report generation system
3.2.7. Finalize prototype
3.3. Development
3.3.1. Develop all modules
3.3.2. Develop non-fuctional
3.3.3. Integrate authentication and security system
3.3.4. Integrate third-party payment system
3.4. Deployment
3.4.1. Setup deployment envrionment
3.4.2. System Optimization and QA
3.4.3. Official system deployement
3.4.4. Monitoring and maintenance
4. Control
4.1. Track system progress
4.2. Compare prototype vs actual model
4.3. Change management
4.4. Testing
4.5. Quality management
4.6. Risk mitigation
5. Closure
5.1. Perform final system review
5.2. Identity and resolve bugs
5.3. Documentations user manuals
5.4. Final product deployment
5.5. Project close-out meeting

Figure 1: Work Breakdown System (WBS)

The Work breakdown system (WBS) given above of the Inventory management system project shows the overall works needed to be done for the completion of the project smoothly and before the deadline ensuring the time and cost remains minimal. The WBS is divided into 5 phases which are Initiation, planning, Execution, control and closure which follows the prototype model with iteration process.

The first phase is the Initiation phase where all the necessary team members, project managers are assembled, defining goals and objectives and conducting feasibility analysis as well as requirement analysis is done. Once all the ground work is done, the planning phase starts. During this phase road maps are created for clear view of the project path. Risk assessment, resource allocation and time scheduling are also done in this phase. After all the risk assessment and planning the Execution phase begins. The execution part is divided into 4 section which are prototype 1, prototype 2, the development and the deployment phase. The first step in the execution part is to develop a prototype 1 where initial UI/UX design is developed. Following the UI/UX development, a simple data base schema is developed. The first phase of prototype 1 is completes with implementation of login and registration features including purchase order system. The prototype 1 concludes with gathering feedback. The prototype 2 starts with refining the prototype according to the feedback gained during prototype 1 phase. All the functional features are refined and developed here. The UI/UX is refined more in this step with the expansion of database schema. All the features such as dispatch order, development of payment system and report generation are completed here. In the last phase of execution part which is development and deployment phase, third party payment methods are implemented. All the security and authentication features are also developed here. The deployment environment is then set up with QA. The system is then finally deployed. The control phase then ensures and maintains quality of the system through monitoring the system, risk mitigation and performance testing.

Finally, in the closure phase, All the documentation, training and deployment are done ensuring the system's functionality and readiness for use. Each phase concludes with milestones to track progress and to check if the deadlines are met efficiently.

2.1. Gantt Chart:

A horizontal bar chart which was developed by Henry L. Gantt, an American social scientist who was also an engineer in 1917 that was developed as a production control which came to be known as Gantt Chart. As a Gantt Chart provides graphical illustration of a schedule which can be used to plan time schedule of the tasks needed for project completion it is frequently used in project management. (Lutkevich, 2022)

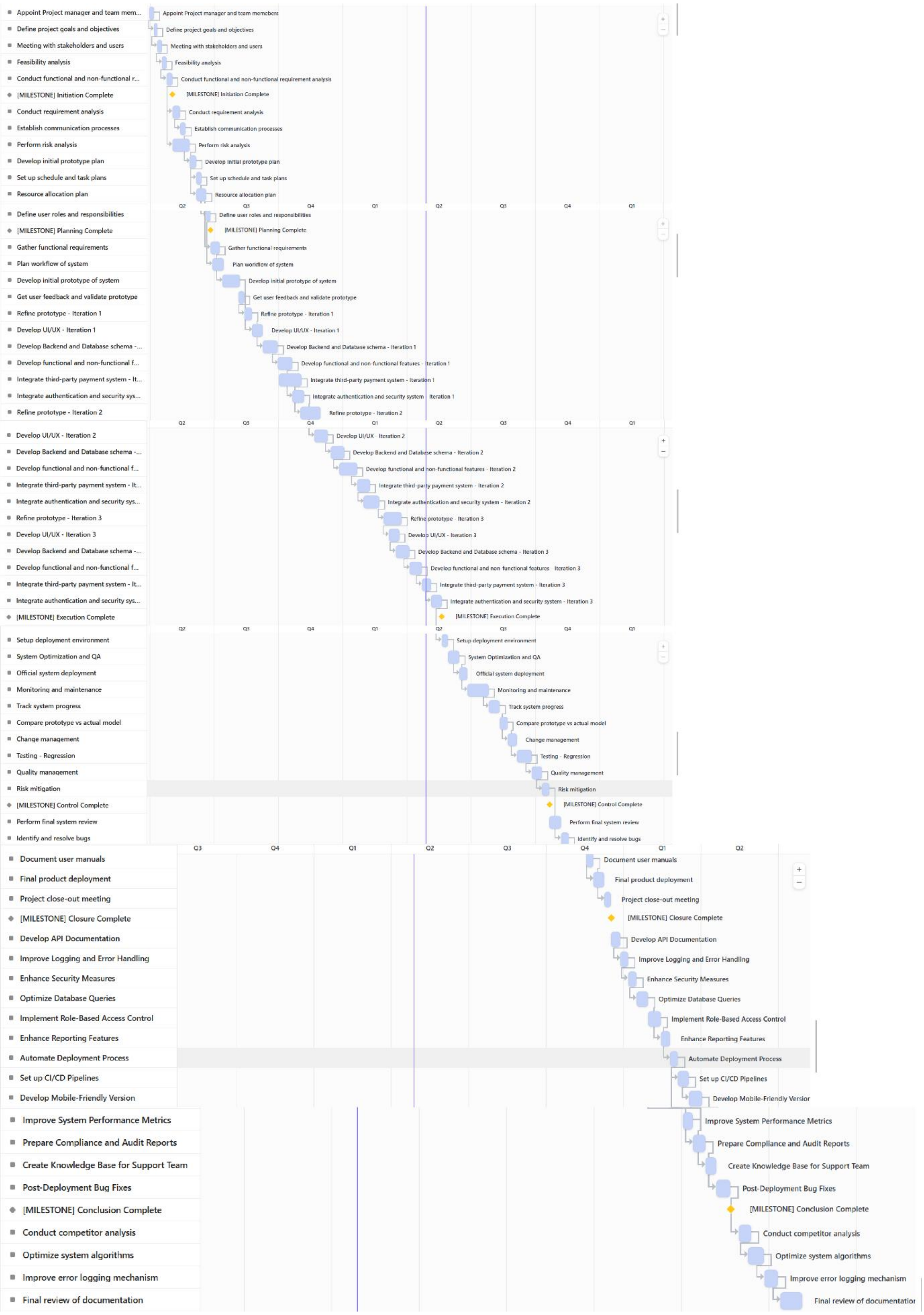


Figure 2: Gantt Chart 1

The Gantt chart above represents a well-structured timeline of a project. This chart outlines various tasks, their dependencies and key milestone. The arrow represents dependency whereas the diamond shape represents the milestones of the overall project that needs to be completed to meet the deadline. The project begins in early April starting with an appointment of a project manager and the team members who will be working together in the project followed by defining project goals and objectives. Each task is dependent on the completion of previous task which is represented by the arrows. This means that the task will only be started with the end of any previous task. The Gantt chart shows the task span over months of time period. Some of the tasks have same date which mean those tasks can be worked on simultaneously to improve productivity. There are few milestone points represented by a diamond shape which indicates the completion of a significant point of project. All the milestones help to keep the record of progress done. The Gantt chart shows all the start date and due date of all the tasks that need to be completed for the completion of project in due date. The Gantt chart serves as a roadmap for project tracking, ensuring deadlines are met and dependencies are properly managed

3. Use Case Diagram:

In 1986 the first use case diagram was proposed by Ivar Jacobson. A use case is a methodology used during system and requirement analysis which helps to in the identification, clarification and organization of system functional requirements. Use case diagram falls under Unified Modeling Language (UML) which is a standard notation for the modeling of real-world entities and objects and systems. The four major components of Use case diagram are Actor, Use cases, Association and System boundary (Aleryani, 2016).

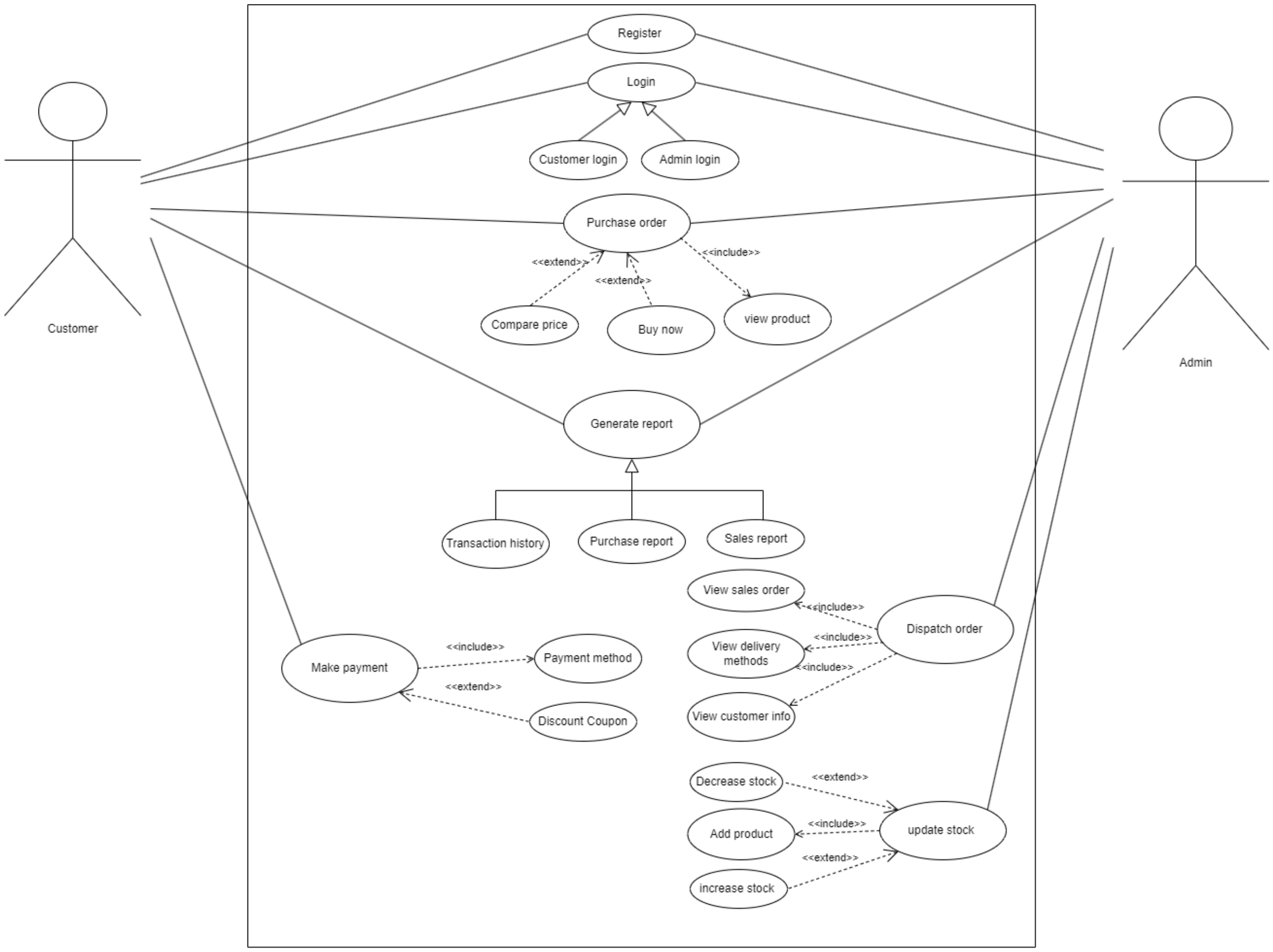


Figure 3: Use case Diagram

The above Use Case Diagram represents the functions and features that the users (Customer/Admin) can use in this Inventory Management system. This diagram outlines the interactions between Customer and Admin. The features are known as use case which are represented my horizontal oval circle which are inside of a system boundary box. Customers can register, login, browser products, compare prices, place orders and make payments through various method, including online payment, cash on delivery and discount coupons. They can also view product details and add different products to their cart before concluding a purchase. With customers, Admin can manage products, process orders and dispatch the products according to the dispatch details. The Admin can manually increase or decrease the stock level as well. Customers can generate their transaction report whereas admin can generate sales report and purchase report and handle customer as well as sales information. In this diagram the “include” is to indicate mandatory functionalities and "extend” indicates optional or conditional functions. The Use Case Diagram effectively demonstrates the all the features available for both customers and the Admin.

3.1. Use Case Description (High-level)

3.1.1. Use case: Login

Actors: Customer/Admin (initiator)

Description: A user provides their personal details to the system. The user can access the system according to their role which is customer/Admin.

3.1.2. Use case: Generate report

Actors: Customer/Admin

Description: The customer can generate a history report which contains the details of all the past purchases and their payment whereas the admin can generate the both the purchase and sales report.

3.1.3. Use case: Register

Actors: Customer/ Admin

Description: The user (Customer/ Admin) provides their credentials and register themselves as either as a customer or an admin.

3.1.4. Use case: Make payment

Actors: Customer

Description: A customer can make payment using different payment method according to their choice as they are given options accordingly. Discount coupons can also be used by the customers.

3.1.5. Use case: Dispatch order

Actors: Admin

Description: The admin gets order from the customer, they can dispatch the order. While dispatching orders the admin can view sales order, delivery methods and customer info to ensure the correct address and customer for order dispatching.

3.1.6. Use case: Purchase order

Actors: Admin/Customer

Description: The admin can make a purchase order of the products they need from the supplier. The admin provides product details for purchase order. A customer can check the details about the details, it’s price and can also compare the price with other products.

3.1.7. Use case: Update stock

Actos: Admin

Description: The admin can manually update the stock. The stock can be whether increased or decreased in the inventory. The stock will also be automatically increase with if the new product is added by the admin.

3.2. Use Case Description (Expanded)

3.2.1. Use Case: Login

Actors: Customer/Admin (initiator)

Description: A user provides their personal details to the system. The user can access the system according to their role which is customer/Admin.

Typical Course of Events:

Table 1: Expanded Description of Login Use Case

Actor Action	System Response
1. The users provide their email and password.	2. If the user credentials are valid, the system sends an authentication code to the user.
3. The user provides the code for the authentication to the system	4. If the code is valid the system grants the access to the user according to their user role if not then the system requests re-entry.

Alternate Courses:

- Line 1 The user is a new user therefore the account doesn’t exist to login to. Use case ends
- Line 4 The OPT code is invalid. Use case ends

3.2.2. Use case: Generate report

Actors: Customer/Admin

Description: The customer can generate a history report which contains the details of all the past purchases and their payment whereas the admin can generate the both the purchase and sales report.

Typical Course of Events:

Table 2: Expanded Use Case description of Generate Report Use Case

Actor Action	System Response
1. The users requests for the report generation. 3. The user selects the report type. 5. The user confirms the report generation request. 7. The user can now view or download the report.	2. The system prompts the user to select the type of report. 4. The system filters the report type and sends a confirmation message. 6. The system generates the report that can be viewed or downloaded as a receipt and sends it to user.

Alternate Courses:

- Line 4 The role is not validated. Use case ends
- Line 5 The user cancels the report generation. Use case ends

4. Communication Diagram:

A communication diagram is a visual representation or a graph that is responsible to denote the behavior of an organization or a structure (Alsaadi, 2006). They are the diagrams that helps to visualize the different interactions that can be happening within a system's architecture. They show how the message transfers between components or objects within a system (Team, 2025).

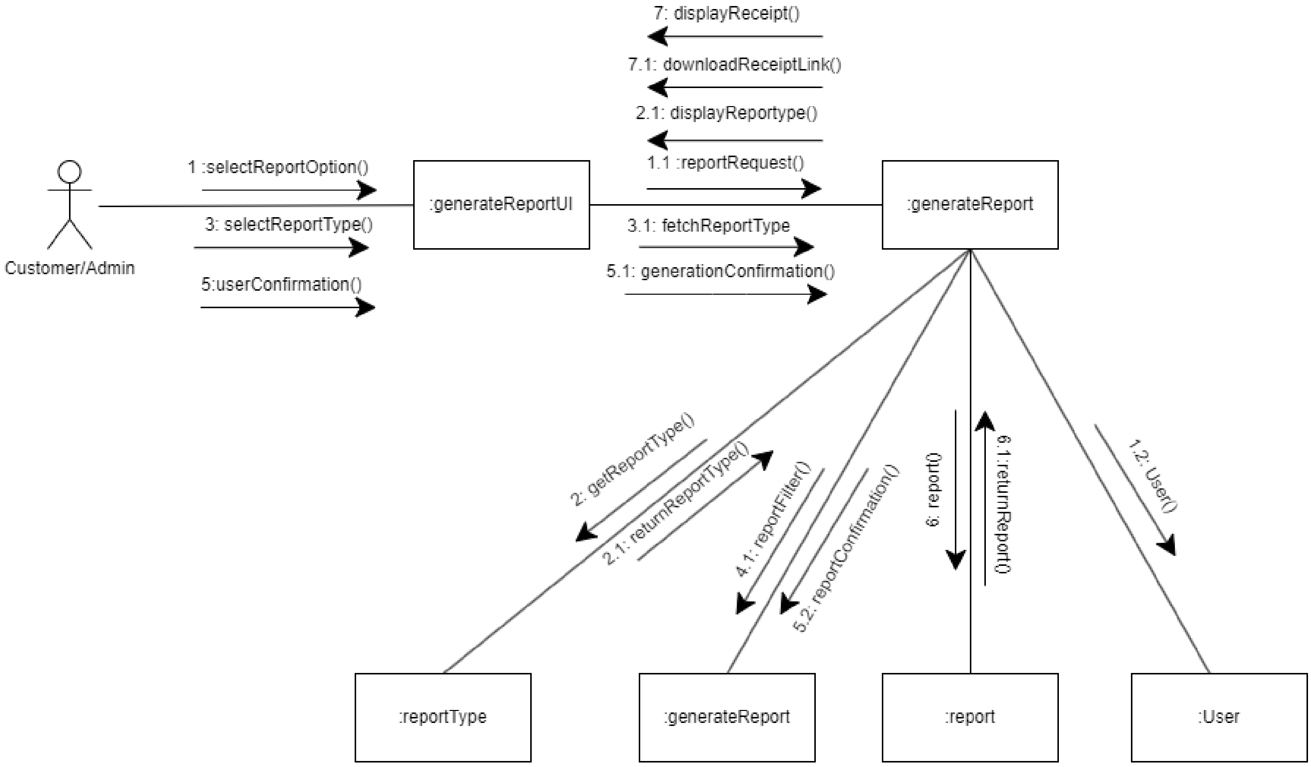


Figure 4: Communication Diagram of Generate Report Use Case

The collaboration diagram given above illustrates the interaction process for generating report (purchase, sales and transaction report) in the inventory system. The user request for a report in the generateReportUI which is then handled by controller object to fetch the information about report types from the domain class. The UI retrieves available report types according to the user role (admin or customer) and displays them for the user to select one of them. The system then filters the report type. Once the filtration is complete, generate report processes the request and retrieves the required report type and report data and generates the final report. The report is then ready to be downloaded to the user.

5. Activity Diagram:

Activity diagrams which is also known as object flow or control flow diagrams. They lie under UML (unified modeling language) behavioral diagrams which provide a graphical representation to define the sequence, condition and composition of lower level behaviors (Baresi, 2009).

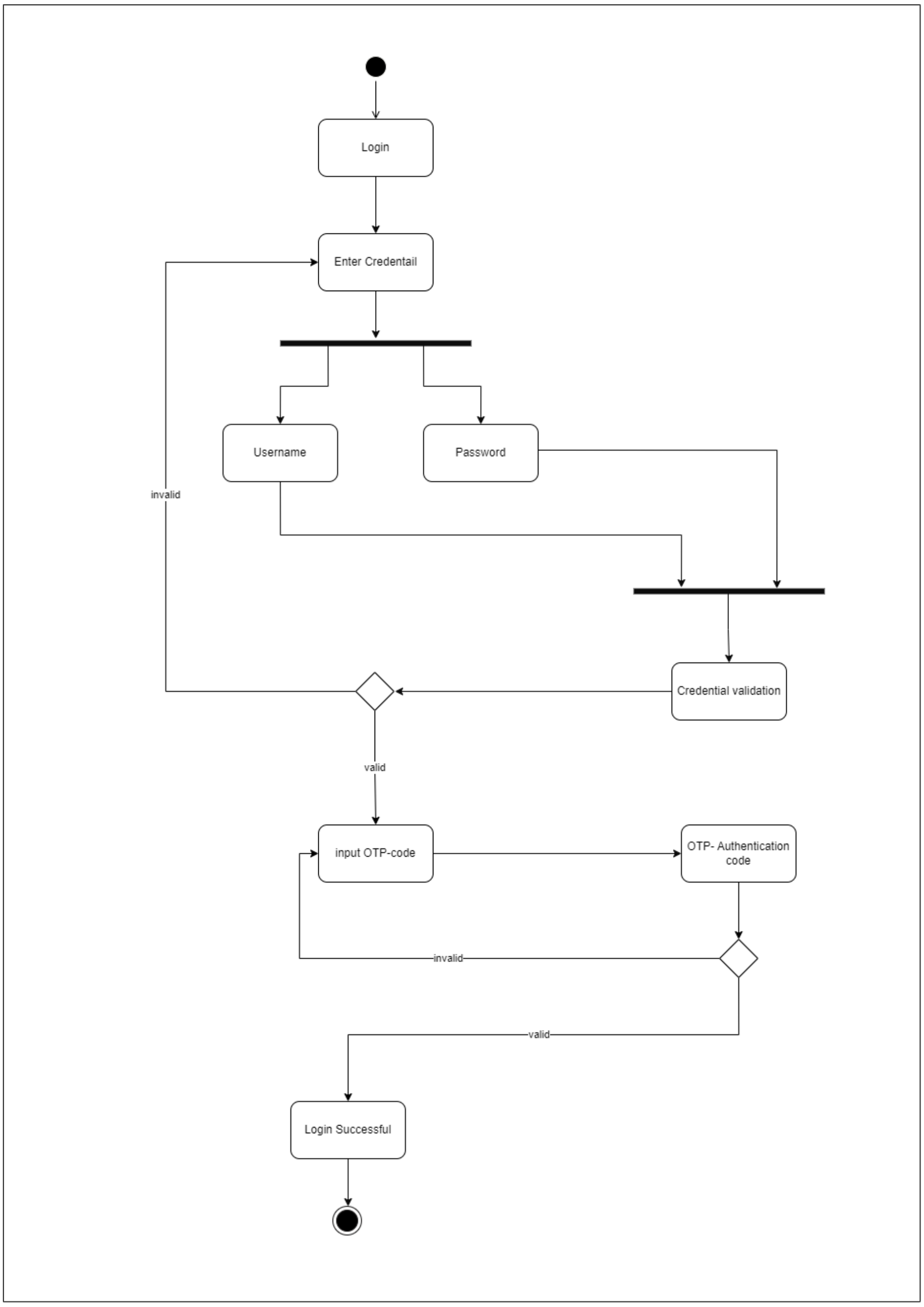


Figure 5: Activity Diagram of Login Use Case

The activity diagram given above gives us the clear idea view of the login process of the inventory management system. It details the sequence of actions and decision points involved during the login process of the user (customer/ admin). The action of user during the login process. It starts with the user entering the role either admin or customer their personal details such as username and password which then undergo the validation process. If the provided credentials are invalid then the user is prompted to re-enter their credentials again but if the credentials are valid then the system proceeds with OTP authentication step, requiring the user to enter a one-time password. If the OTP is correct then the login is successful if not then re-enter the OTP. The login process is then complete, ensuring a secure authentication mechanism.

6. Class Diagram

Class diagram is a part of UML diagram which is most commonly found in object-oriented system. This diagram helps to provide the visual representation of static view point of a system. Different types of classes, interfaces and their relationships in a system, all are shown in this diagram (Janis Osis, 2017).

Table 3: Use case and domain class

Use case	Domain Class
Register	Registration, user
Login	Login, user, session
Create Purchase order	Inventory, supplier, purchase order, Product
Dispatch order	user, inventory, delivery method
Real time stock	Product, inventory, add product
Make payment	Payment method, order, payment
Report Generation	User, Report, report types (Purchase, Sales, transaction history)

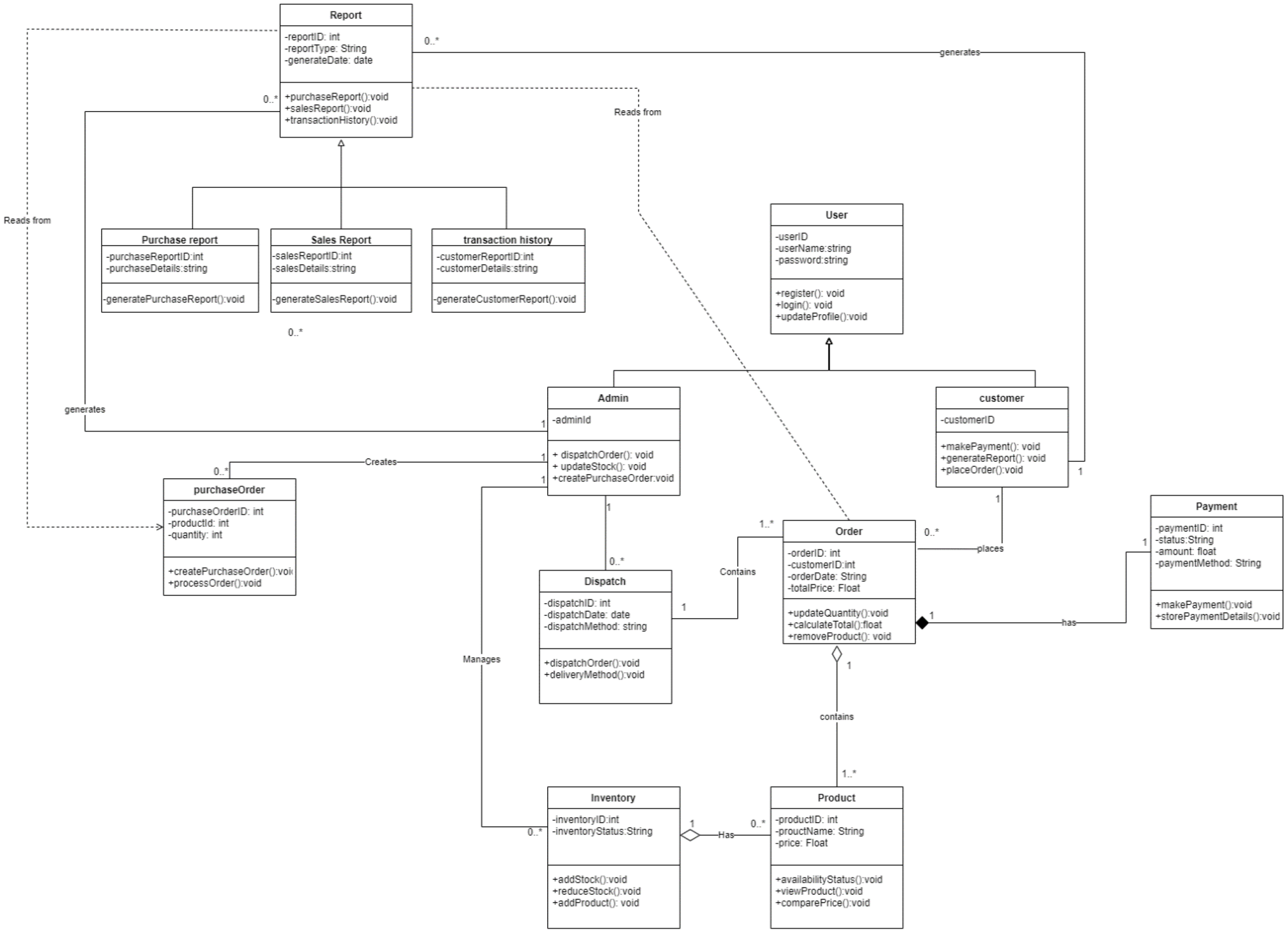


Figure 6: Class diagram of IMS

The given class diagram above represents and shows how all the classes present in a software connect and communicate to each other. There's a special class called UserID that is the main part for two types of users: Admins and Customers. Both of these user types can do things like sign up, log in, and change their profile information. Admins take care of many products in the store, and customers can make multiple payments and place several orders. An order can have many products in it and is linked to a delivery record. Each product has important details like its ID, name, and price, and it connects back to the inventory to keep track of how many items are available. There’s also a class that gives different roles to users, and a Report class that helps the business understand sales and purchases better by creating reports and allowing users to download receipts. Overall, this system is built to manage everything needed for an online shopping platform effectively.

7. Further Development

The currently progressing inventory management system (IMS) for Global Teach Corporation can be even more elevated with the help of adoption of the Agile Scrum Methodology. The developments of different features and the bug fixings will be completed in 2-4-week sprints. Each of the sprint will be helpful for the further refining and development with improved results. The first and initial sprints will be responsible for development of core features such as login, registration and user authentication. Product management and sales management are also developed in this sprint. The last sprints will be responsible for the expansion of report generation and payment system. The normal sprints will be used for planning, daily reports and feedbacks will help to keep the progress on going.

7.1. Architectural choice

The architectural choice for this project of inventory management system will be '**Micro-Services Architecture**'. Reason to choose this architecture are given below:

- **Scalability:** This architecture helps the system to scale the performance as much as needed and prevents overloading the system.

- **Flexibility:** This architecture helps the individualize a component which helps during the development of a single component without having to deploy entire application

- **Fault tolerance:** Failure of single component does not affect the whole service which gives the system fault tolerance feature.

7.2. Design pattern choice: Observer pattern

The observer pattern is a design pattern which allows different sections to observe and respond automatically according to the condition. In our IMS, this pattern helps to ensure when state of a components changes all the related modules changed automatically which helps to keep the system consistent and reduce manual work.

7.3. Development plan:

The development of this inventory management system (IMS) will be progressed with the help of different kind of technology stacks and tools. All the modern tools and technology stacks for the development of front-end, backend, database and cloud will be chosen with consideration to maximize the efficiency and minimize the labor cost as well as time frame. The technology stack and tools for the development of these features are given below:

7.4. Technology Stack and Tools

- Frontend:
  - For the frontend, React.js will be used for dynamic and responsive UI

- Backend:
  - For the backend, Python will be used.
  - For the framework, Django will be used.

- Database:
  - For the database, MongoDb will be used for flexible and strong database management system

- Tools:
  - For the version control, Git and Github will be used
  - For the API testing Postman will be used

- Cloud Services:
  - Hybrid cloud services.
  - AWS cloud services

7.5. Testing Plan:

Software testing plan refers to the evaluation activity where all the system requirements and components are tested manually by a software tester with the help of automation tools to verify if the system is running smoothly free of errors and also know if the system is deliverable to the customers (Hooda, 2015). After the development of the inventory management system (IMS), a proper and comprehensive testing plan will be conducted. The testing will be held to track the progress and ensure the flexibility, scalability and functionality of the system. All the bugs will also be tested during this period.

The different types of testing that will be used during this project are:

- **Unit testing:** Unit testing helps to test the components and modules separately and individually. Features re login, registration, payment and product management will be tested by unit testing. Different tools such as Jest and mocha will be used for this testing

- **Integration testing:** The integration test is a type of test that will help to test and know how the different components are communicating to develop one feature. This testing will be responsible to test the API integration between frontend and the backend of the system. The data flow between database and application are also ensured with the help of this testing. Postman and Supertest are some tools that will be used for API and backend testing.

- **Security testing:** The objective of this testing is to know if the system is strong enough to remain unfazed against different types of cyber-attacks and threats. This test helps to detect different types of attacks such as SQL injection and CSRF attacks. Different types of antivirus and security system will be used such as OWASP ZAP and Burp Suite.

Other different types of testing such as UAT testing, Regression Testing and performance testing will also be implemented for extra evaluation of the system.

7.6. Maintenance Plan:

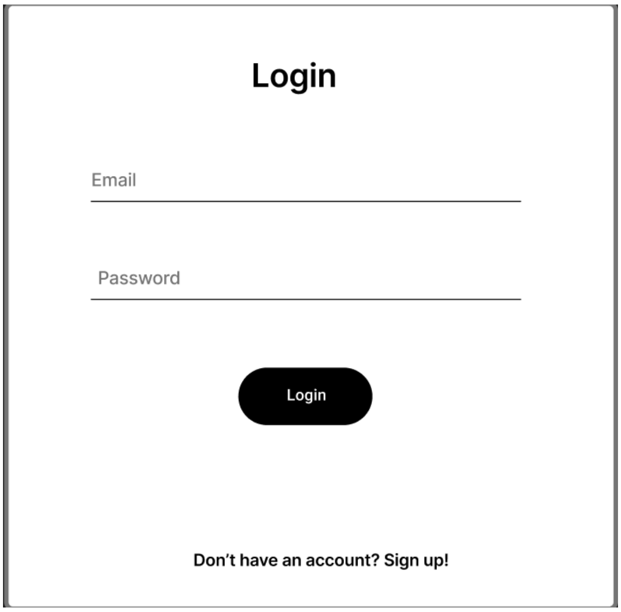
After the deployment of the system, the continuous maintenance of the system is important to keep up the efficiency and productivity of the system. Different types of maintenance plans will be implemented for the further development of the system.

Maintenance plans for the further development:

- Daily monitor the system and fix the bugs if found.
- Continuous optimization of the system to increase the efficiency of the system
- Monitor the security and enhance the system security
- Optimize databases to handle the ever-growing data
- Maintain scalability of the system.
- Get feedbacks from the consumers and stakeholders.
- Updates to implements different improvements few monthly.
- Major updates and develop new features.

8. Prototype Development

Login:



Don’t have an account? Sign up!

Figure 7: Login Prototype

Registration page:

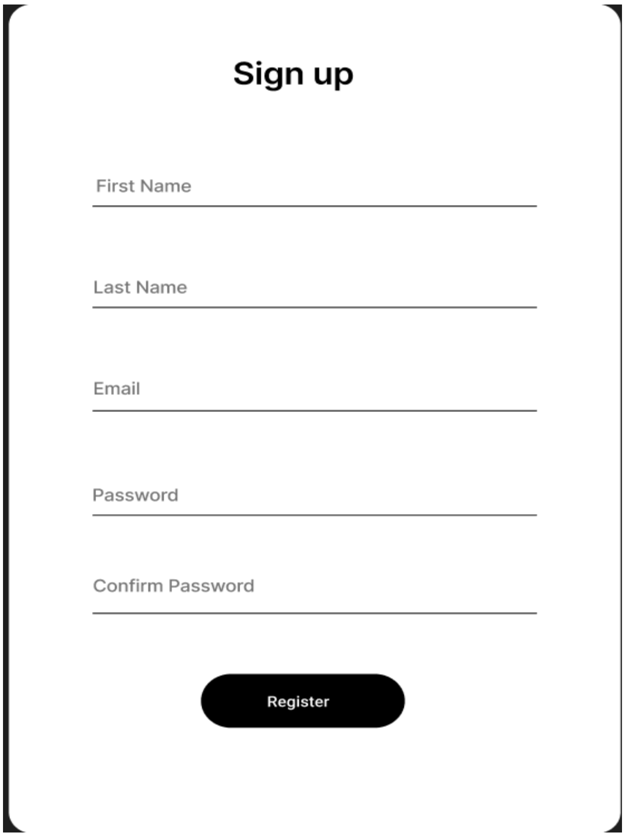
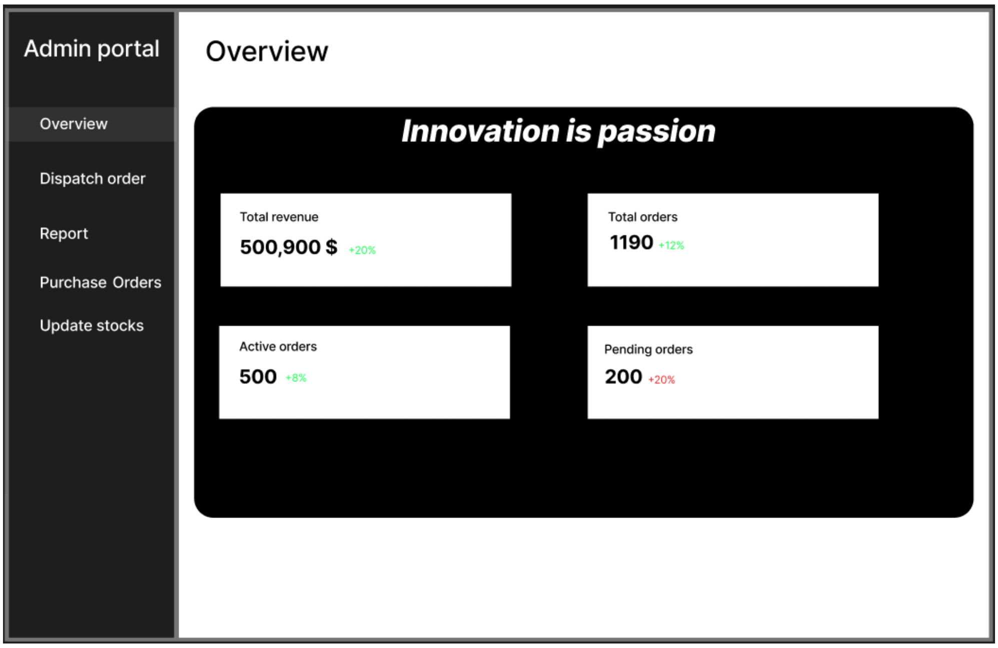


Figure 8:Register Prototype

Admin dashboard (overview):



Overview

Innovation is passion

Total revenue
500,900 $ +20%

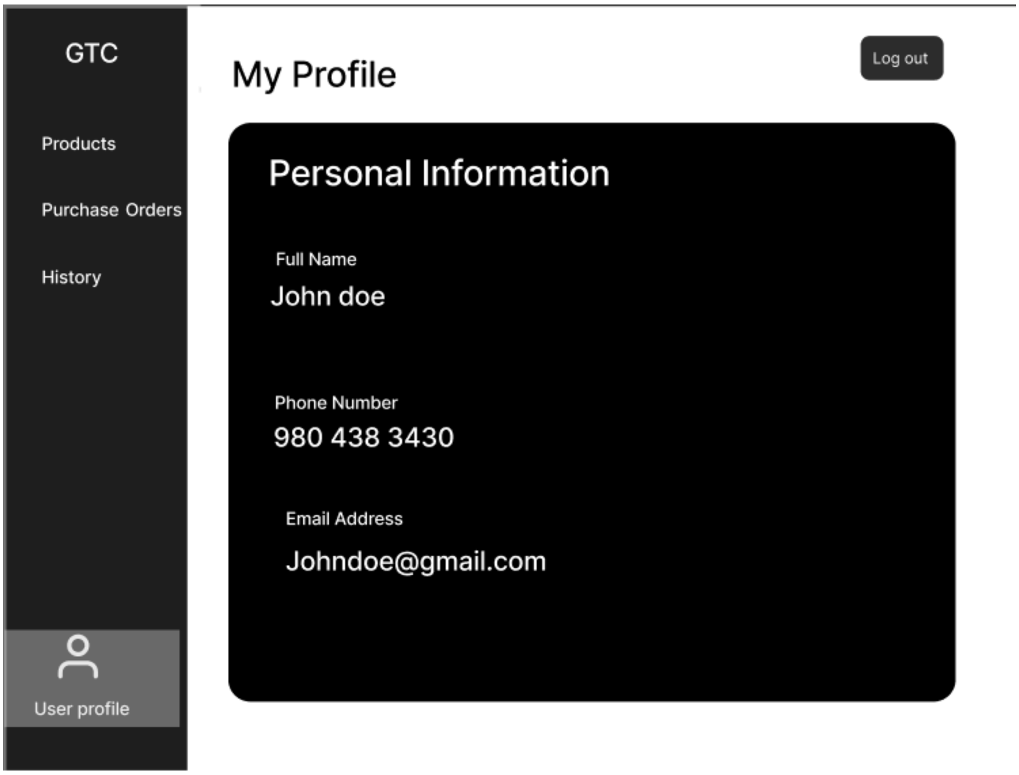
Total orders
1190 +12%

Active orders
500 +8%

Pending orders
200 +20%

Figure 9: Admin Dashboard prototype

User Profile:



My Profile

Personal Information

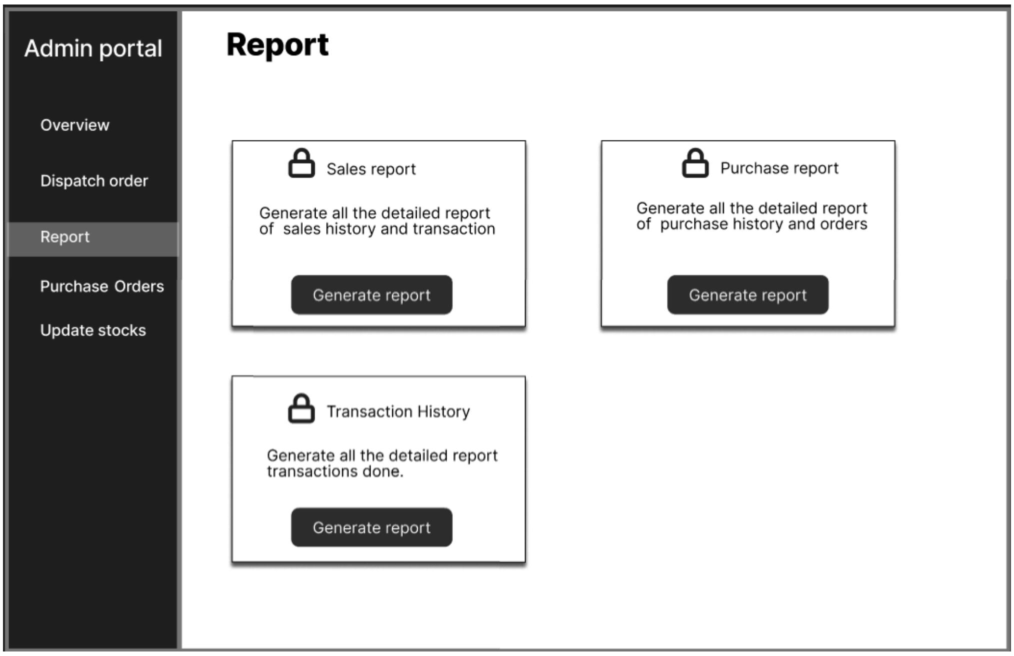
Full Name
John doe

Phone Number
980 438 3430

Email Address
Johndoe@gmail.com

Figure 10: User profile prototype

Report Generation page:



Report

Sales report

Generate all the detailed report of sales history and transaction

Purchase report

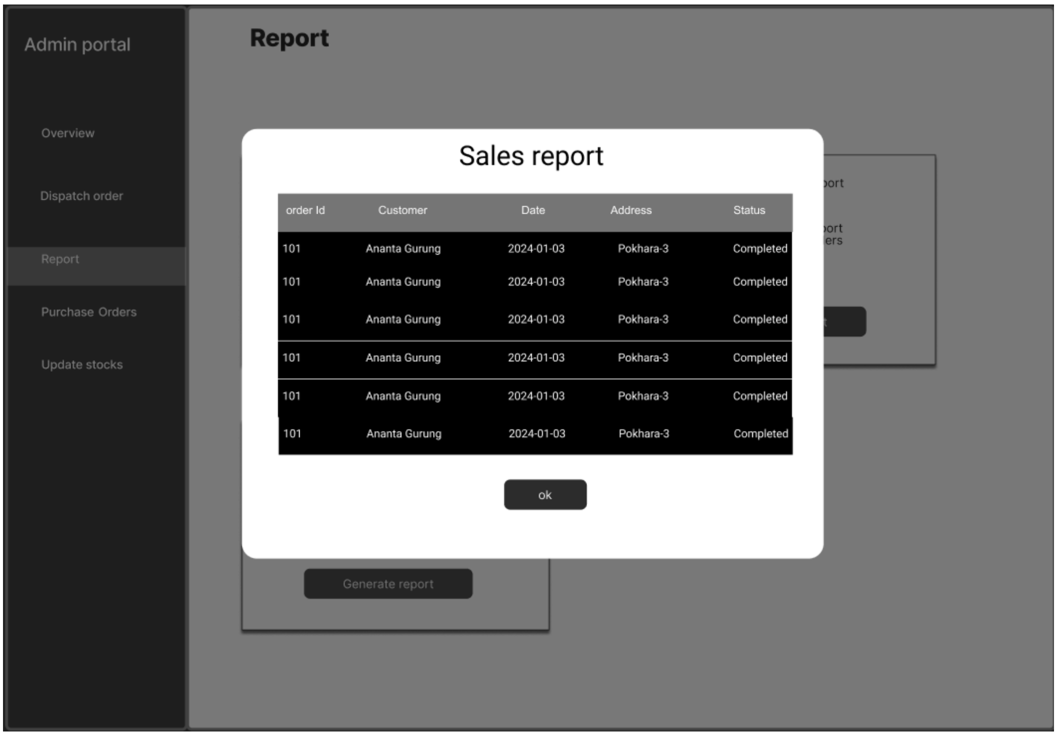
Generate all the detailed report of purchase history and orders

Transaction History

Generate all the detailed report transactions done.

Figure 11:Report Prototype

Sales Report:



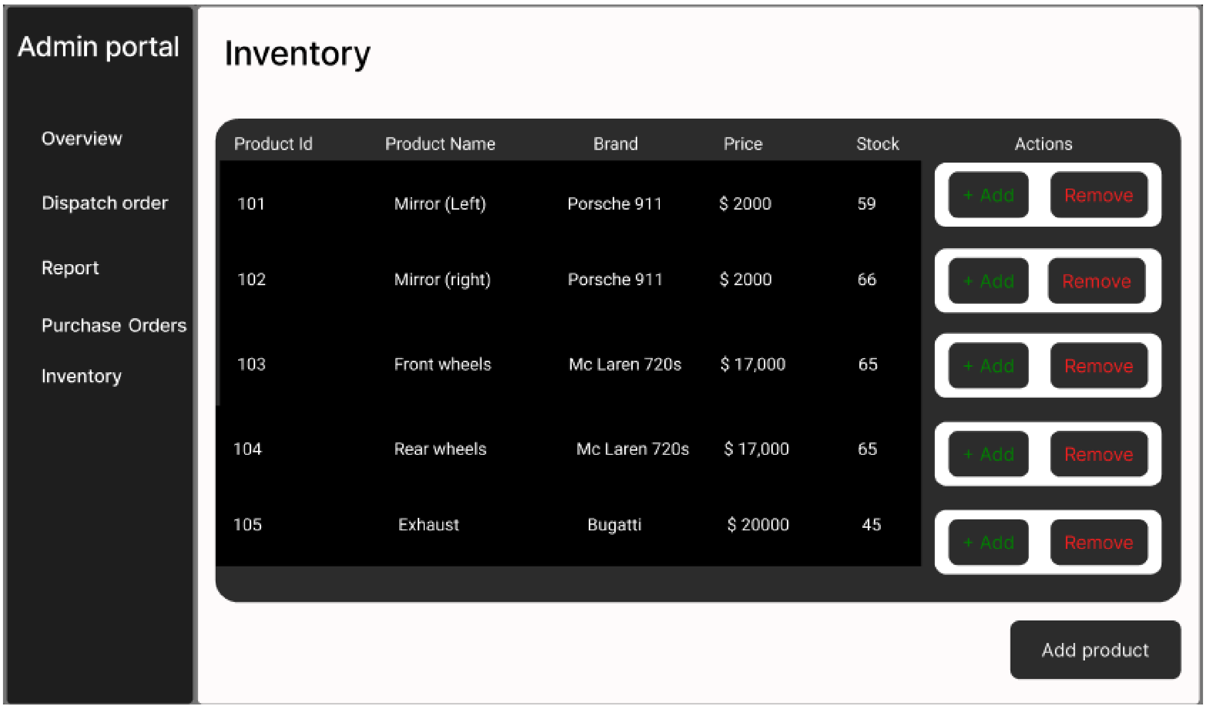
Report

Sales report

order Id	Customer	Date	Address	Status
101	Ananta Gurung	2024-01-03	Pokhara-3	Completed
101	Ananta Gurung	2024-01-03	Pokhara-3	Completed
101	Ananta Gurung	2024-01-03	Pokhara-3	Completed
101	Ananta Gurung	2024-01-03	Pokhara-3	Completed
101	Ananta Gurung	2024-01-03	Pokhara-3	Completed
101	Ananta Gurung	2024-01-03	Pokhara-3	Completed

Figure 12: Sales Report Prototype

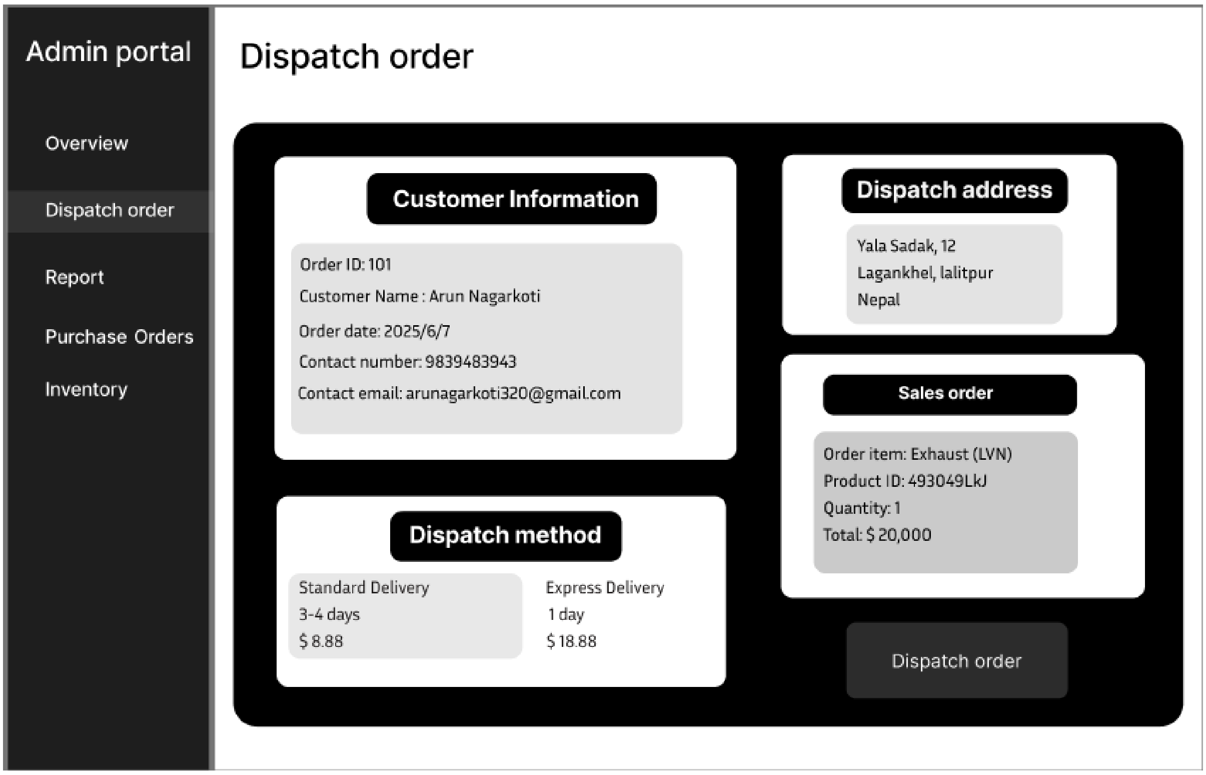
Inventory:



Product Id	Product Name	Brand	Price	Stock	Actions
101	Mirror (Left)	Porsche 911	$ 2000	59	+ Add Remove
102	Mirror (right)	Porsche 911	$ 2000	66	+ Add Remove
103	Front wheels	Mc Laren 720s	$ 17,000	65	+ Add Remove
104	Rear wheels	Mc Laren 720s	$ 17,000	65	+ Add Remove
105	Exhaust	Bugatti	$ 20000	45	+ Add Remove

Figure 13: Inventory prototype

Dispatch order



Customer Information

Order ID: 101
Customer Name : Arun Nagarkoti
Order date: 2025/6/7
Contact number: 9839483943
Contact email: arunagarkoti320@gmail.com

Dispatch address

Yala Sadak, 12
Lagankhel, lalitpur
Nepal

Sales order

Order item: Exhaust (LVN)
Product ID: 493049LkJ
Quantity: 1
Total: $ 20,000

Dispatch method

Standard Delivery
3-4 days
$ 8.88

Express Delivery
1 day
$ 18.88

Figure 14: Dispatch order Prototype

Product page:

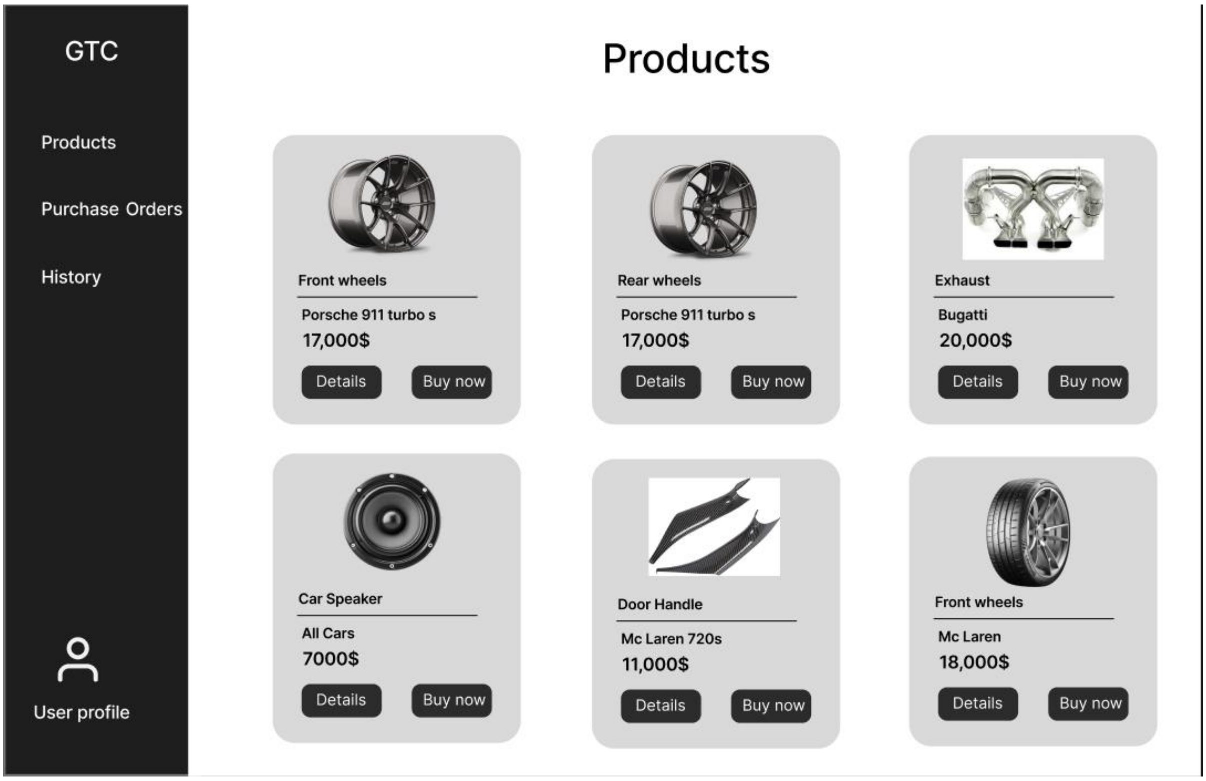


Figure 15: Products Prototype

Product details page:

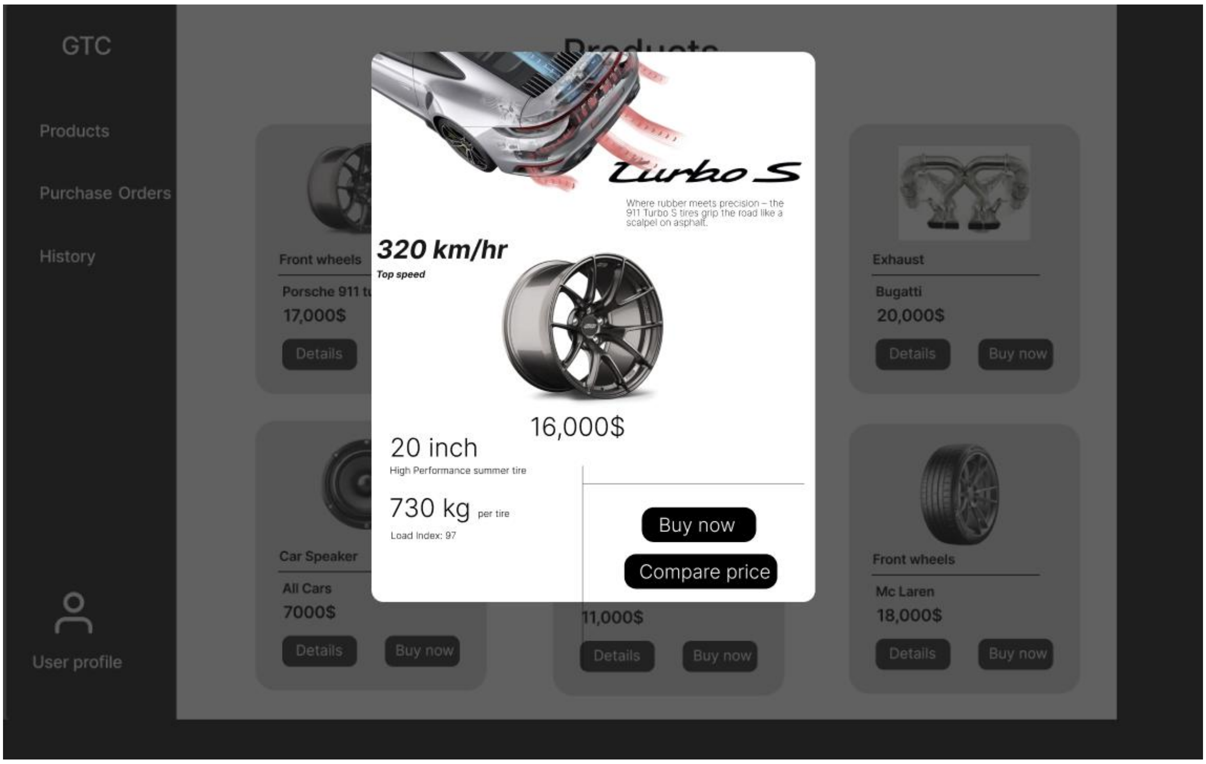
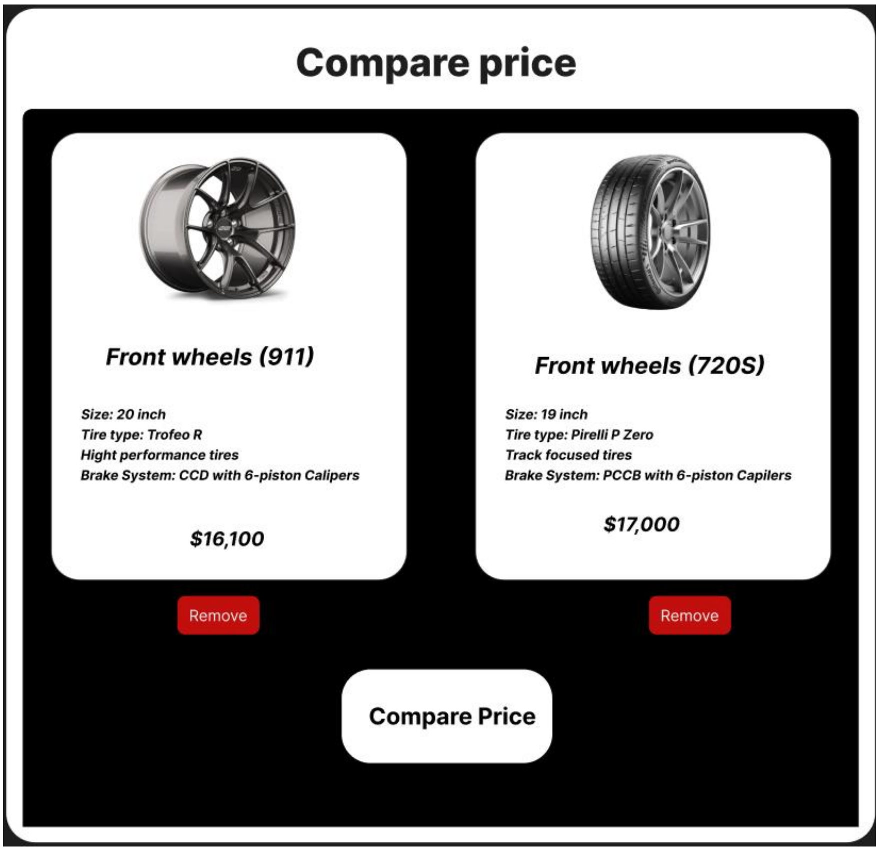


Figure 16: View Product details Prototype

Compare price page:



Compare price

Front wheels (911)

Size: 20 inch
Tire type: Trofeo R
Hight performance tires
Brake System: CCD with 6-piston Calipers

$16,100

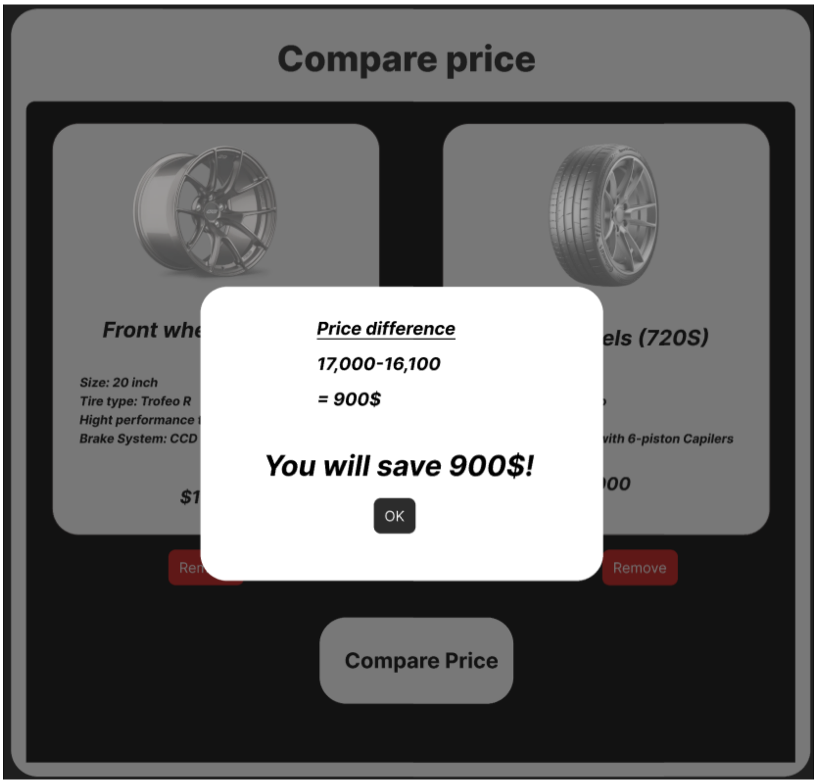
Front wheels (720S)

Size: 19 inch
Tire type: Pirelli P Zero
Track focused tires
Brake System: PCCB with 6-piston Capilers

$17,000

Figure 17: Compare price interface Prototype

Price compared page:

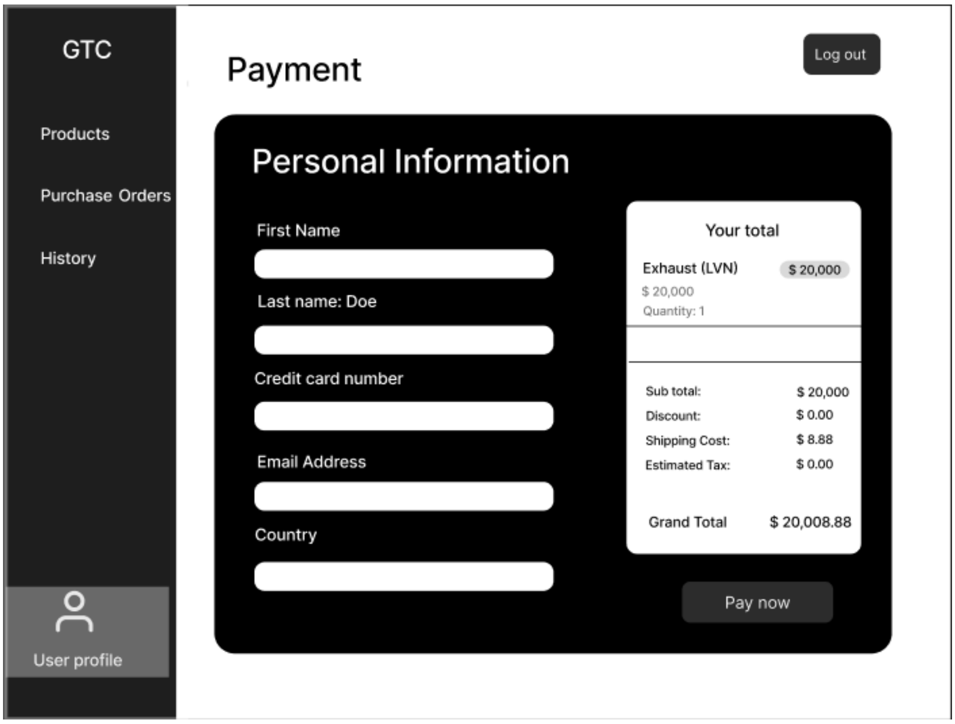


Price difference
17,000-16,100
= 900$

You will save 900$!

Figure 18: Price comparison Prototype

Payment page:

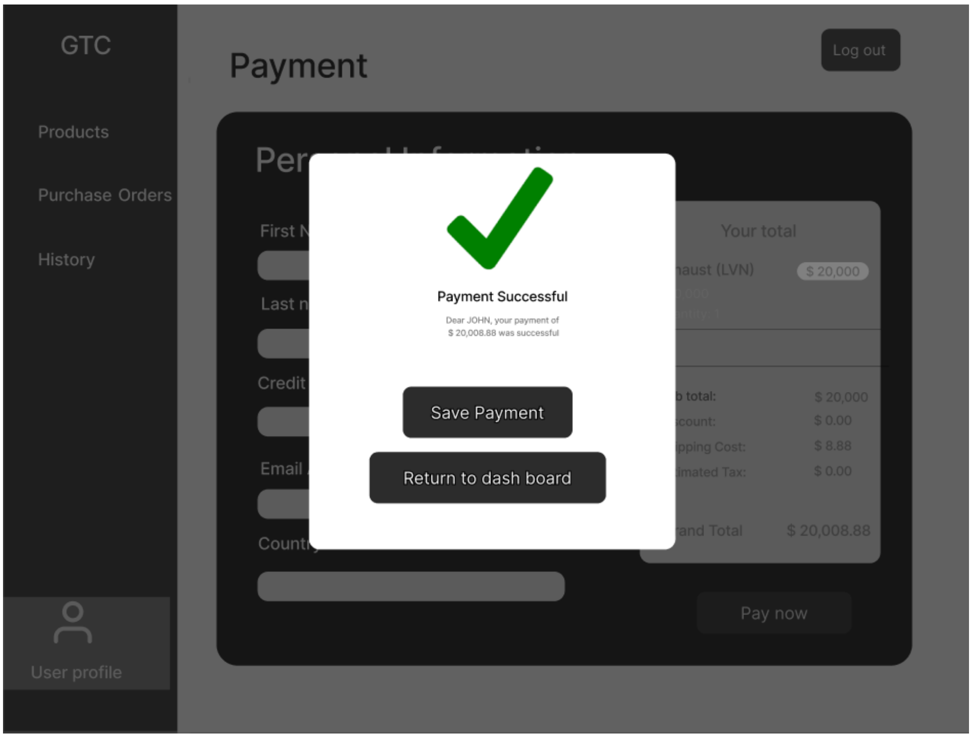


Exhaust (LVN) $ 20,000
$ 20,000
Quantity: 1

Sub total:	$ 20,000
Discount:	$ 0.00
Shipping Cost:	$ 8.88
Estimated Tax:	$ 0.00
Grand Total	$ 20,008.88

Figure 19: Payment Prototype

Payment Successful:

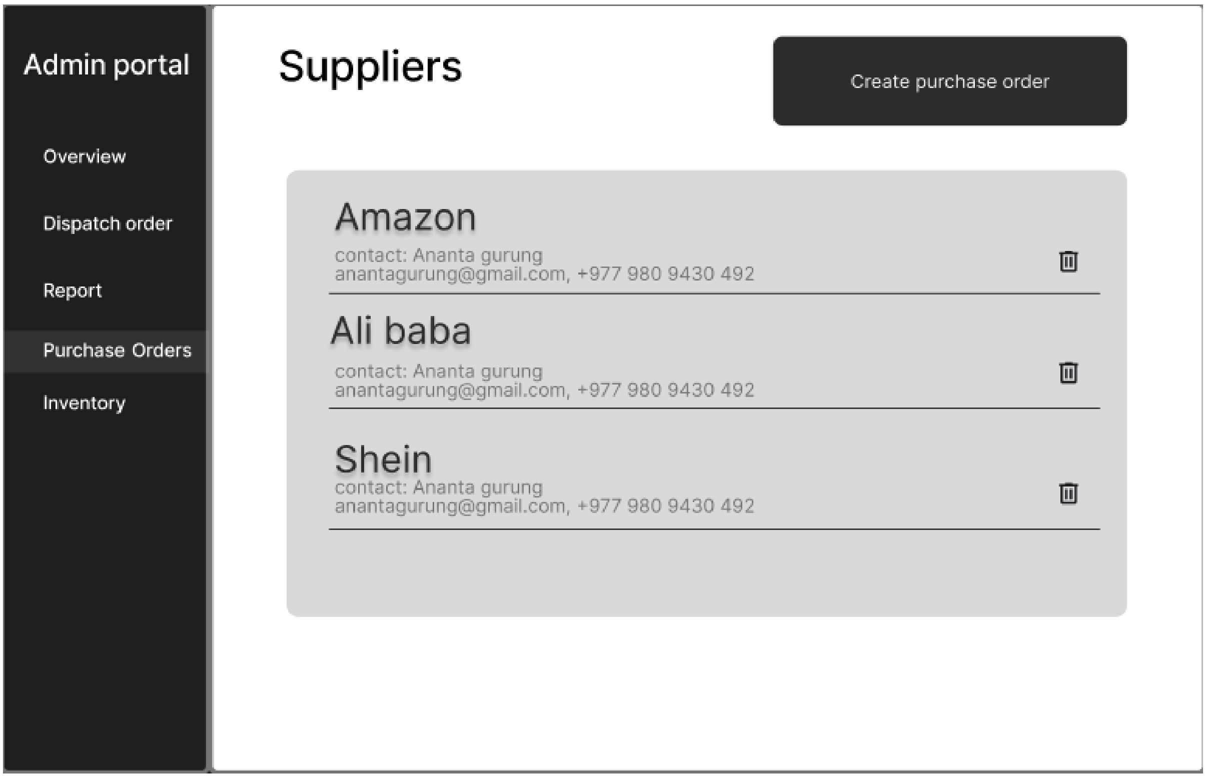


Payment Successful

Dear JOHN, your payment of
$ 20,008.88 was successful

Figure 20: Payment Successful Prototype

Suppliers and Purchase order page:



Amazon
contact: Ananta gurung
anantagurung@gmail.com, +977 980 9430 492

Ali baba
contact: Ananta gurung
anantagurung@gmail.com, +977 980 9430 492

Shein
contact: Ananta gurung
anantagurung@gmail.com, +977 980 9430 492

Figure 21: Create Purchase order

9. Conclusion:

In conclusion, the development of the inventory management system (IMS) for the Global tech company is finally completed. We studied how the company had failed to make any profit due to previous poor management of and design of the system which caused a lot of issues for the company draining them economically. We developed our new modern fully automation inventory management system to address this issue. For this project we used different UML diagrams such as Gantt chart, activity diagram, use-case diagram and communication diagram. We divided the tasks and broke down he works into smaller sub tasks for proper work distribution in our prototyping model.

We used prototyping model for our development and iterative process to further refine the prototype which makes the development of the system more accurate, reliable, strong and flexible which helps us to implements improvisations more easily. Also, we used microservices architecture for smoother development experience and updates. Different technology stack and tools were used during the development of this project for all frontend, backend and databases and used testing plans and maintenance plan. Completing all the milestones, we have finally developed our robust, scalable and flexible inventory management system for the Global Tech company to assist then in their business journey making all the works more efficient and productive.

Finally, we documented every steps we took, every methodology we used and overcoming all the troubles and mistakes, we have concluded this project with a proper documentation.

10. Bibliography

Aleryani, A. Y., 2016. Comparative Study between Data Flow Diagram and Use. *International Journal of Scientific and Research Publications,* 6(3), pp. 124-127.

Alsaadi, A., 2006. *Checking Data Integrity via the UML Class Diagram,* s.l.: IEEE.

Baresi, L., 2009. *springer.* [Online]
Available at: https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_9
[Accessed 31 march 2025].

Hooda, I., 2015. psu.edu. *Software Test Process, Testing Types and Techniques,* 111(13), pp. 60-70.

Janis Osis, U. D., 2017. *Topics/computer.* [Online]
Available at: https://www.sciencedirect.com/topics/computer-science/class-diagram
[Accessed 27 April 2025].

Lutkevich, B., 2022. *Agile.* [Online]
Available at: https://www.techtarget.com/searchsoftwarequality/definition/Gantt-chart
[Accessed 4 March 2025].

Tausworthe, R. C., 1980. science direct. *Journal System and software,* Volume I, pp. 181-186.

Team, I. E., 2025. *indeed.* [Online]
Available at: https://uk.indeed.com/career-advice/career-development/communication-diagram
[Accessed 31 March 2025].