GRABBO

ONLINE DEPARTMENTAL STORE

TABLE OF CONTENTS

INTRODUCTION

Grabbo was born out of a simple idea: to make shopping easier for university students. We noticed that students often struggled with knowing what was available at the campus store. So, we created Grabbo to solve this problem. From the beginning, our goal was clear: provide real-time information about product availability to streamline the shopping process for students and administrators.

As we planned Grabbo's development, we focused on essential features like real-time updates, easy navigation, and quick search. As the project progressed, we continued to refine Grabbo's capabilities. One significant addition was allowing students to buy products directly through Grabbo using UPI payments. We also added features to help administrators verify transactions quickly and efficiently.

This report tells the story of Grabbo, from its inception to its current state. It showcases how Grabbo has improved the university shopping experience for students and administrators alike. Through Grabbo, we've made shopping on campus simpler and more convenient for everyone involved.

1.1 Problem Statement

The university shopping experience is often hindered by the lack of real-time information on product availability, leading to wasted time and inefficiencies for both students and administrators. These challenges underscore the need for a solution that provides real-time access to product availability for students. Grabbo seeks to address these issues by offering real-time updates to enhance the university shopping experience.

BACKGROUND RESEARCH

With the increasing reliance on online shopping platforms for convenience and accessibility, the idea of establishing an online departmental store for campus students has gained significant traction. The project, named GRABBO, aims to address the needs and preferences of students by providing a seamless online shopping experience tailored specifically for their campus community.

Research conducted among fellow students revealed overwhelming support for the concept of an online departmental store, with 78% expressing a need for such a service. This highlights the demand for convenient shopping options that cater to the busy schedules and lifestyle preferences of students.

Furthermore, insights into delivery preferences indicate that a majority of students (70%) prefer the convenience of home delivery, while a smaller percentage (30%) opt for the pickup option. This information is crucial for designing the logistics and delivery framework of GRABBO to ensure maximum satisfaction among users.

The project's flow involves a user-friendly online webstore where students can easily navigate, select items, and add them to their carts. Upon checkout, users have the flexibility to choose between home delivery and pickup options, providing them with a convenient shopping experience tailored to their preferences. On the administrative side, functionalities include item management, such as adding, updating, and deleting products, as well as verifying user purchases. This ensures smooth operation and efficient management of the online departmental store, enhancing the overall user experience.

In summary, GRABBO aims to revolutionize campus shopping by leveraging online technology to provide students with a convenient and efficient way to access essential items without the need for physical visits to the departmental store. Through careful consideration of user preferences and needs, the project endeavours to enhance student life and streamline the campus shopping experience.

PROPOSED SYSTEM

The proposed system, Grabbo, aims to revolutionize the university shopping experience by introducing real-time product availability updates and streamlined purchasing options for students. By leveraging modern technology and user-centric design principles, Grabbo seeks to address the challenges associated with traditional campus shopping, offering a more efficient and convenient solution for students and administrators alike.

3.1 Goals and Objectives

1. Provide students with accurate and up-to-date information on product availability within campus stores.
2. Simplify the purchasing process for students by offering a user-friendly interface for direct transactions within the platform.
3. Streamline transaction management for administrators with tools for prompt verification and real-time inventory updates.
4. Create an intuitive and user-friendly platform to enhance satisfaction and engagement for both students and administrators.
5. Ensure Grabbo is accessible across various devices, including desktops, tablets, and smartphones, to cater to diverse user preferences and needs.

PROJECT PLANNING

4.1 Project Setup

Decision Description	Description
Technology Stack	▪ Frontend: HTML, CSS, JavaScript ▪ Backend: Node.js, Express.js ▪ Database: MySQL ▪ Deployment: Firebase Hosting
Standards and Conduct	Adherence to coding standards defined by the project team and university guidelines.
Access Privileges	Establishment of access privileges for user authentication and adherence to legal regulations.
Feedback Mechanism	Deployment of feedback forms on the cloud for user input, facilitating application updates and scaling.
Business Model	Development of a sustainable business model for long-term market presence and collaboration with industries.

4.2 Stakeholders and their Roles

Stakeholder	Role
Priyansh Singhal	Frontend & Backend Developer
Yagay Khatri	Database Administrator
Prashast Upadhyay	Tester

4.3 Resource Description

Resource	Resource Description	Quantity
Database Server	MySQL	1
Grabbo Team	Primary developers of this project.	2
Workstations	Workstations for the developers in which they will be developing the product.	2
Android Phone	An Android phone to be used as test hardware for the mobile version of the Application and Feedback for further improvements.	1

4.4 Assumptions

#	Assumptions
A1	The team members will be able to communicate anytime during the week.
A2	Scope of the project will remain constant throughout.
A3	The team will write the backend code in Node.js and deploy it on a Express.js server.
A4	The team will use MySQL as database server.
A5	Project costs will stay the same as initially budgeted costs.
A6	Project will follow AGILE development methodology throughout.

SYSTEM ANALYSIS AND DESIGN

5.1 Overall Description

Grabbo is an innovative online webstore designed specifically for university students, providing a convenient solution for purchasing items from the campus departmental store. With Grabbo, students can effortlessly browse through available products, add them to their cart, and proceed to checkout, all from the comfort of their hostels. The platform offers seamless integration between frontend and backend components, ensuring a smooth user experience. Administrators have access to functionalities for managing inventory, verifying purchases, and maintaining the store's operations. With real-time updates and seamless integration between frontend and backend components, Grabbo sets out to redefine the university shopping landscape, providing students and administrators alike with a convenient, efficient, and modern solution for their shopping needs.

5.2 Users and Roles

User	Description
Developer (Frontend & Backend)	Responsible for designing and implementing both the frontend and backend components of the platform, ensuring seamless integration and optimal user experience.
Database Administrator	Responsible for designing, implementing, and maintaining the database structure, optimizing performance, and troubleshooting any database-related issues to ensure smooth operation of the application.
Tester	Runs thorough unit tests to confirm that software components function correctly, ensuring they meet specified requirements and operate as expected in the application environment.
Application User	This is you, the person using the app! You'll be tapping and swiping through the screens, ordering groceries, and selecting delivery options—all to make your life a little easier.

5.3 User Stories

1. As a student, I want to be able to log in to the webstore using my university credentials so that I can access the platform securely.
   Acceptance Criteria:
   - The webstore should have a login page where students can enter their university credentials.
   - Upon successful authentication, students should be redirected to the homepage of the webstore.

2. As a student, I want to search for products and add them to my cart so that I can easily shop for items I need.
   Acceptance Criteria:
   - The webstore should have a search bar where students can enter keywords to find products.
   - Students should be able to view product details and add items to their cart with a single click.

3. As a student, I want to choose between home delivery and pickup options during checkout so that I can select the most convenient delivery method.
   Acceptance Criteria:
   - During the checkout process, students should be presented with options for home delivery and pickup.
   - Students should be able to select their preferred delivery method before proceeding to payment.

4. As a student, I want to make payments using UPI so that I can complete my purchase securely and quickly.
   Acceptance Criteria:
   - The webstore should provide a UPI payment option at checkout.
   - Students should be able to scan a QR code or enter UPI details to initiate the payment process.

5. As an admin, I want to add new items to the webstore so that I can update the product inventory.
   Acceptance Criteria:
   - The admin dashboard should have an option to add new items to the product catalogue.
   - Admins should be able to enter details such as product name, description, price, and quantity.

6. As an admin, I want to verify pending orders made by users so that I can confirm payments and process orders efficiently.
   Acceptance Criteria:
   - The admin dashboard should display a list of pending orders awaiting verification.
   - Admins should be able to review payment details and confirm orders with a single click.

7. As an admin, I want to update existing items in the webstore so that I can make changes to product information as needed.
   Acceptance Criteria:
   - The admin dashboard should provide options to edit product details such as price, and quantity.
   - Changes made by admins should be reflected in the product listings on the webstore.

8. As an admin, I want to delete items from the webstore if they are no longer available or relevant so that I can maintain an accurate product inventory.
   Acceptance Criteria:
   - The admin dashboard should have a feature to delete items from the product catalogue.

5.4 Sprints

Sprint 1
Start Date: 4/03/2024
End Date: 10/03/2024
Duration: 1 week

ID	Added	Description	Status	% Completed
100	Onset	*As a* student, *I want to* be able to log in to the webstore using my university credentials *So that* I can access the platform securely.	C	100
Acceptance Criteria			**Verification**	
110	The webstore should have a login page where students can enter their university credentials.		Manual testing by the Tester.	
111	Upon successful authentication, students should be redirected to the homepage of the webstore.		Manual testing by the Tester.	
ID	**Tasks**		**Resource**	
1	Set up project environment (Frontend & Backend)		Developer	
2	Implement user authentication		Developer	
3	Create homepage layout		Developer	
4	Test user authentication functionality		Tester	

Sprint 2
Start Date: 11/03/2024
End Date: 24/03/2024
Duration: 2 weeks

ID	Added	Description	Status	% Completed
200	Onset	*As a* student, *I want to* search for products and add them to my cart, *So that* I can easily shop for items I need.	C	100
201	Onset	*As a* student, *I want to* choose between home delivery and pickup options during checkout, *So that* I can select the most convenient delivery method.	C	100
Acceptance Criteria			**Verification**	
210	The webstore should have a search bar where students can enter keywords to find products.		Manual testing by the Tester.	
211	Students should be able to view product details and add items to their cart with a single click.		Create automated test cases to test.	
212	During the checkout process, students should be presented with options for home delivery and pickup.		Manual testing by the Tester.	
213	Students should be able to select their preferred delivery method before proceeding to payment.		Manual testing by the Tester.	

ID	Tasks	Resource
1	Implement product search functionality	Developer
2	Implement add to cart functionality	Developer
3	Implement checkout process	Developer
4	Test product search and add to cart functionality	Tester

Sprint 3
Start Date: 25/03/2024
End Date: 31/03/2024
Duration: 1 week

ID	Added	Description	Status	% Completed
300	Onset	*As a* student, *I want to* make payments using UPI, *So that* I can complete my purchase securely and quickly.	C	100
Acceptance Criteria			**Verification**	
310	The webstore should provide a UPI payment option at checkout.		Manual testing by the Tester.	
311	Students should be able to scan a QR code or enter UPI details to initiate the payment process.		Manual testing by the Tester.	
ID	**Tasks**		**Resource**	
1	Integrate UPI payment option		Developer	
2	Test UPI payment functionality		Tester	

Sprint 4
Start Date: 1/04/2024
End Date: 14/04/2024
Duration: 2 weeks

ID	Added	Description	Status	% Completed
400	Onset	*As an* admin, *I want to* add new items to the webstore, *So that* I can update the product inventory.	C	100
401	Onset	*As an* admin, *I want to* verify pending orders made by users, *So that* I can confirm payments and process orders efficiently.	C	100
402	Onset	*As an* admin, *I want to* update existing items in the webstore, *So that* I can make changes to product information as needed.	C	100
403	Onset	*As an* admin,	C	100

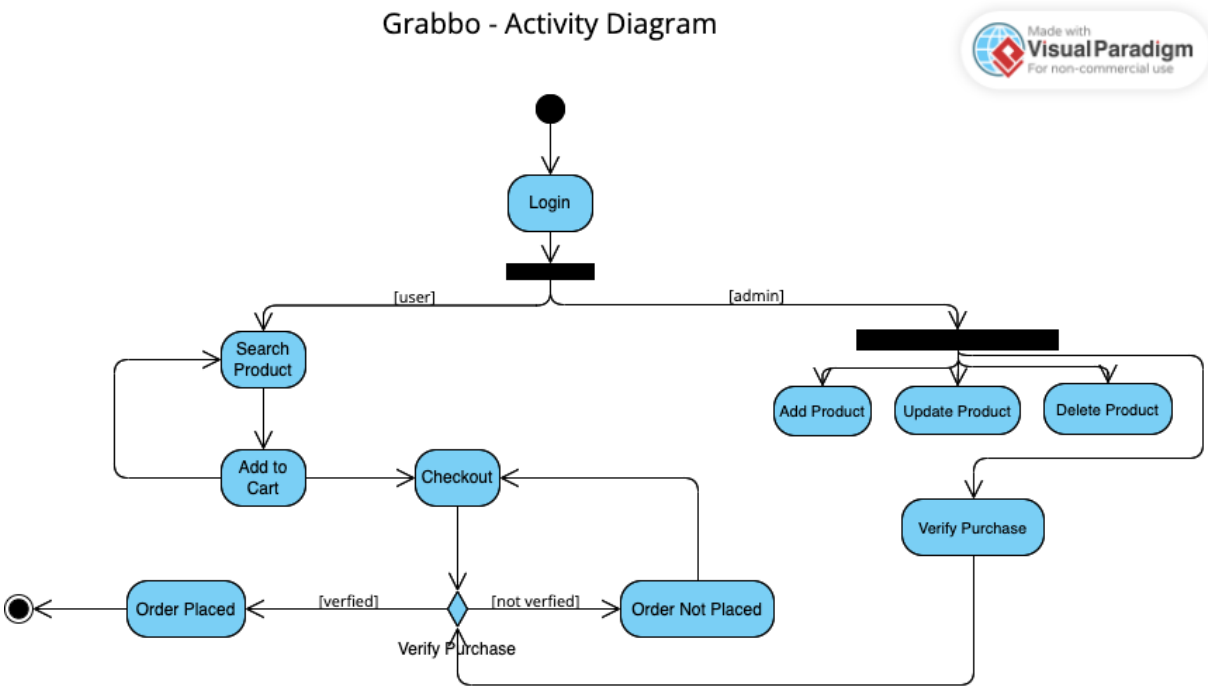
		I want to delete items from the webstore if they are no longer available or relevant *So that* I can maintain an accurate product inventory.		
Acceptance Criteria			**Verification**	
410	The admin dashboard should have an option to add new items to the product catalogue.		Manual testing by the Tester.	
411	Admins should be able to enter details such as product name, description, price, and quantity.		Manual testing by the Tester.	
412	The admin dashboard should display a list of pending orders awaiting verification.		Create automated test cases to test.	
413	Admins should be able to review payment details and confirm orders with a single click.		Create automated test cases to test.	
414	The admin dashboard should provide options to edit product details such as price and quantity.		Manual testing by the Tester.	
415	Changes made by admins should be reflected in the product listings on the webstore.		Manual testing by the Tester.	
416	The admin dashboard should have a feature to delete items from the product catalogue.		Manual testing by the Tester.	
ID	**Tasks**		**Resource**	
1	Implement add new items functionality		Developer	
2	Implement verify orders functionality		Developer	
3	Implement update items functionality		Developer	
4	Implement delete items functionality		Developer	
5	Test admin dashboard functionalities		Tester	

Sprint 5
Start Date: 15/04/2024
End Date: 21/04/2024
Duration: 1 week

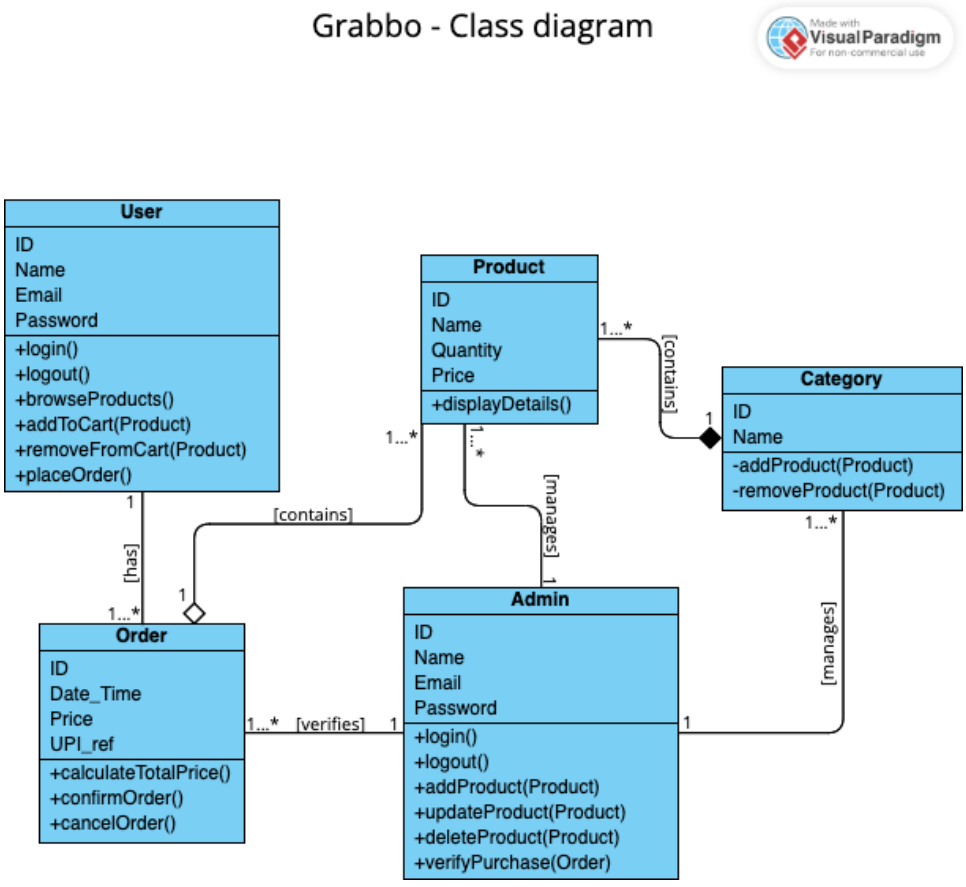
ID	Added	Description	Status	% Completed
500	Onset	*Integrating* all functionalities and performing final testing.	C	100
Acceptance Criteria			**Verification**	
510	All functionalities are integrated seamlessly.		Manual testing by the Tester.	
511	All user stories have been completed and verified.		Manual testing by the Tester.	
ID	**Tasks**		**Resource**	
1	Integrate all functionalities		Developer	
2	Perform final testing		Tester	
3	Refine and fix any issues found during testing		Developer	

UML DIAGRAMS

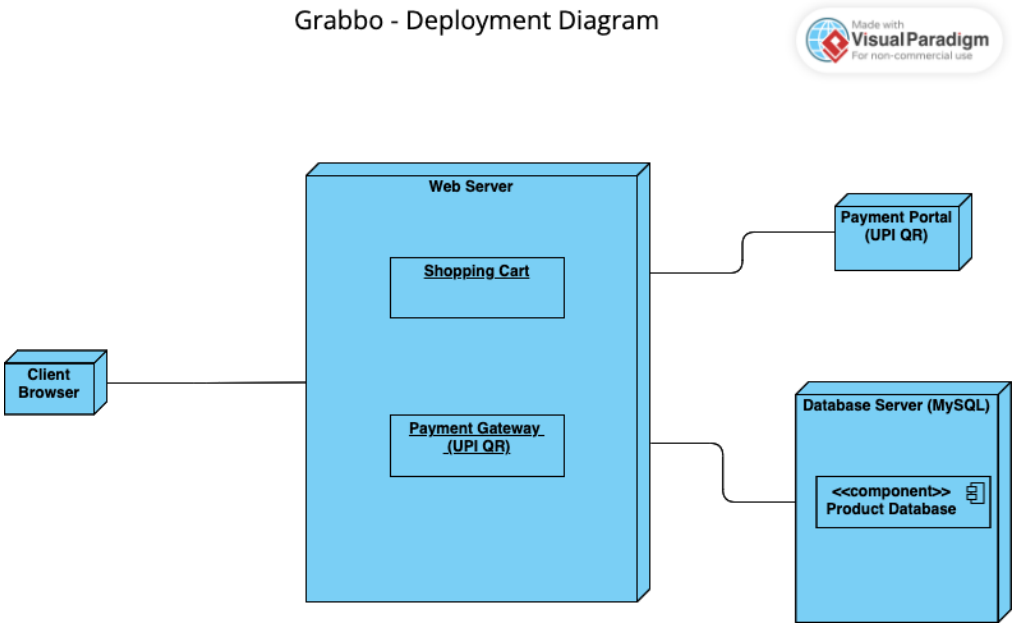
6.1 Activity Diagram



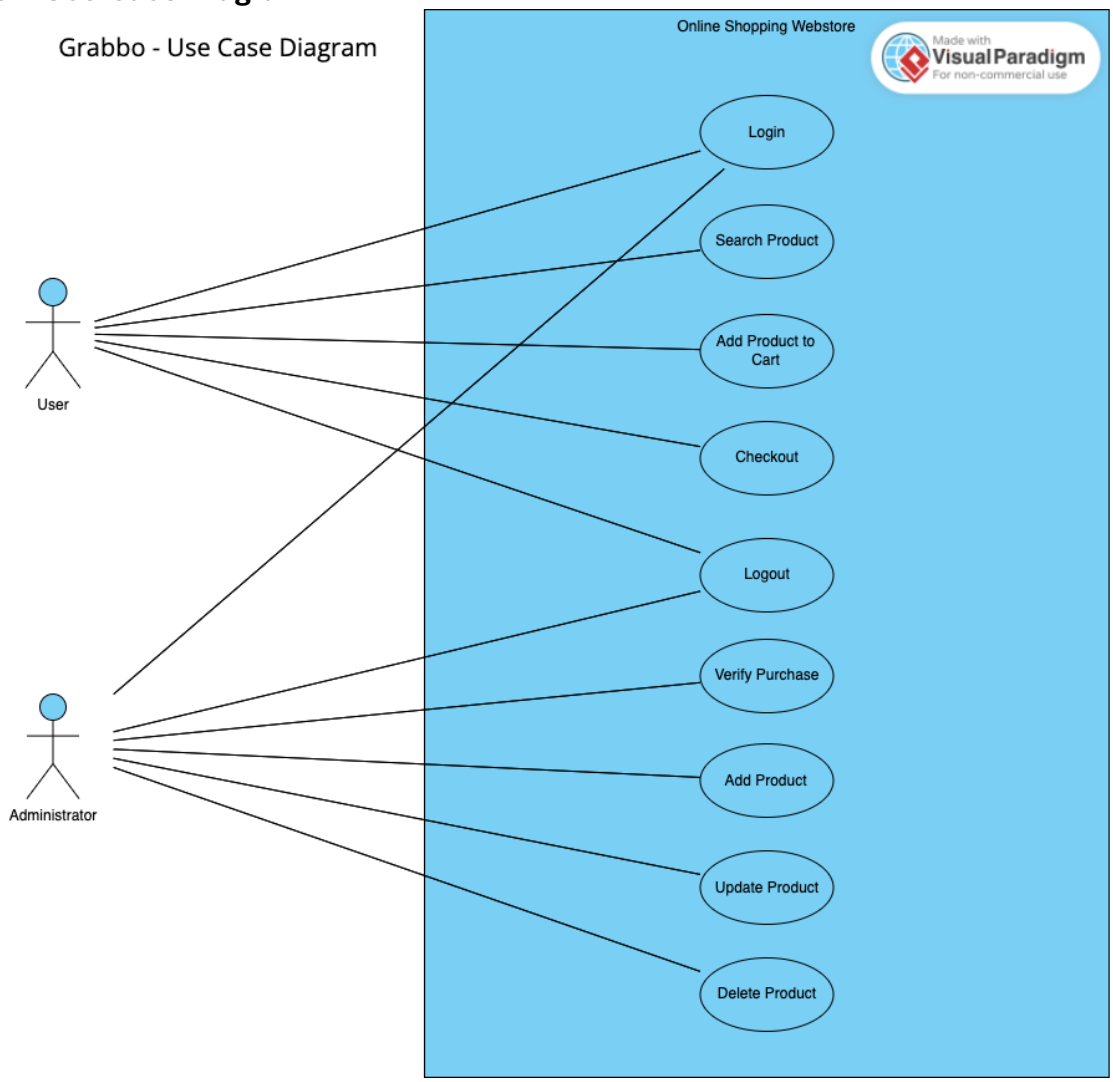
6.2 Class Diagram



6.3 Deployment Diagram



6.4 Use-case Diagram



USER INTERFACE