Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST)

Final Year Project Report

EDistributor

Project Supervisor: Sir Asim Ali
Project Name: EDistributor
Group Members:
Muskan (1812125)
Harmeet Jot (1812150)

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science

The Department of Computer Science

Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST)
100 Clifton Karachi

1.1 Declaration of Authorship

We Muskan (1812125) and Harmeet Jot (1812150) declare that this report “**EDistributor**” and the work presented in this report is our own.

The work has been done completely while in the candidacy for a bachelor’s degree at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology (SZABIST) Karachi, any report previously submitted on this topic in this university or any other institution is clearly mentioned in this report. Everything we used in this report which is submitted by others or belong to any other person or organization is stated in this report.

We have always cited the work we have used. We have acknowledged all source of help we used for this report. The report is based on the research work done by the team members with, we have clearly stated the sources where we took help from to conduct the research.

Signed:

Muskan (1812125)
Harmeet Jot (1812150)

Dated: 1st August 2022

1.2 Project Description

The idea of this project came into our mind when our client approached us for their problem. The main problem faced by client was that all of his business was done manually, and all the records kept were written on paper. He wanted an electronical system for business, all the records of transactions and goods should be stored in database, and so whenever he would like to monitor his sales and purchase, he could easily carry out all activities on his desktop. So we decided to create two platforms one for client and one for customers. For client we created a web portal where he can see all his transactions, inventory record and all users. For customers we developed a mobile application where customers can buy products at ease of their home.

The main idea for this project is to enhance feasibility in business and make ease for customers. Our app will automate everything for client. All you need is a smartphone which is owned by everyone around. Plus to manage the application for any additional functionalities or updating current functions we would need a Personal Computer to alter the app.

The Project Selection had different stages where an extensive amount of research and discussion took place. Multiple ideas were scrutinized but the best and reliable project was to solve a real life problem through our project. This is where EDistributor comes in place. Multiple versions of EDistributor were discussed with our supervisor Sir Asim Ali before finalizing the best. Eventually we came with the perfect solution to prosper a complete platform for our client and his customer, this platform not only digitized the business but will also help to spread his business.

1.3 Acknowledgement

In the name of ALLAH, the most beneficent and merciful who gave us the knowledge and courage to work on this project. We would like to thank SZABIST for giving us an opportunity to be a part of this prestigious institute and pursue our dreams. Also we would like to mention our supervisor Sir Asim Ali who helped us out in completing this project and deliver it on time.

We would like to thank to the teachers at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, who guided us and taught us throughout our time in the university. We would also like to express our gratitude to our parents and family members who helped and encouraged us during this time.

Lastly, we would like to extend our gratitude to everyone at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology for creating and environment for students to thrive in. The quality of education, the cooperative faculty members and the motivation provided by them.

Table of Contents

I. PROPOSAL

1. Introduction

We are designing an application for some business which aims to enhance and monitor sell and purchase and make an ease in billing of orders .After covid, almost everything shifted online and our lifestyle also changed. For common business people it became tough to fulfill demands of customers, so here we are now introducing a platform for a business where distributors will sell their products directly to the end users. As it was a long procedure before to take orders and then generate the bill and go to deliver the stock to the suppliers. All of this work was done manually but now anyone can order the desired product at ease of their home, make payment online and there will be no need to send any employee to take the order from shop to shop. This will help them to save their time and energy and they will get their product at their doorsteps.

2. Objective

- To design an application for a business where distributors will sell their product online and manage their resources.
- Customers will be able to book their order and make payment online.
- Distributors will be able to manage stock online and monitor sale of their products.

3. Problem Description

Even in modern times most of the business in rural Sindh are operating manually and investing so much of their resources, so it’s time for some change, we are targeting a business in rural Sindh where distributors are still utilizing the old methodology of taking orders going from shop to shop to take orders then deliver the stock and generate bill. As whole world is being shifted to work online, why this distribution system is not getting in touch with technological routes?

In this platform, distributors can easily manage all their work like stock management, billing, and dues. Moreover, customers will order by themselves without waiting for any person to come from the distribution firm. They can select any item added by the distributor on the platform. Meanwhile, they can see the stock availability in terms of quantity to match their needs.

A single platform will be working and covering all the work done by the employees. This will cut off the long hectic process into a short and quick process. At first, people may feel difficult to adapt the new process but no sooner they will be at ease.

4. Methodology

We’ll be using agile model for our application development, as our application will be divided into multiple increments where each increment will be developed through iterations in fixed intervals.

For our backend we'll be using NodeJs, ExpressJs and we'll be using MongoDB for our database. The front-end will be created using React-Native.

5. Project Scope

The scope of the project is to help out the distributor and give them a new form of business. As they need to move along with the technology to make the most of their resources. By this application distributors will be able to manage their stock and check most bought product for further purchase of required stock. Our idea is to develop a simple yet effective and smooth application so that every user can easily use it without any difficulties. As in localities, there is a lack of information in terms of technology and stuff. This project may help them in getting in touch with such a technology interface.

6. Feasibility Study

With above defined scope, would you be able to meet your project schedule? Do mention following aspects:

i. **Risks Involved**: There might be a risk while implementing payment gateways as there could be a financial dispute.
ii. **Resource Requirement**: A laptop with 8 GB ram and 256GB SSD for operating Android studio and VS Code.

7. Solution Application Areas

Our application is targeting a business in rural Sindh where distributors will be able to shift their business from manual work to online and manage their stock. We aim to help distributors so that they don't have to go from shop to shop to take orders and do the billing procedure which will not only help them to monitor their sales but it will be ease for them to take orders on their phone. We are here targeting the business of interior Sindh where people are still running things manually. This platform will let them extend their business in nearby cities.

8. Tools/Technology

- NodeJs
- ExpressJs
- MongoDB
- React Native
- Android Studio
- Visual Code Studio
- Postman

9. Expertise of the Team Members

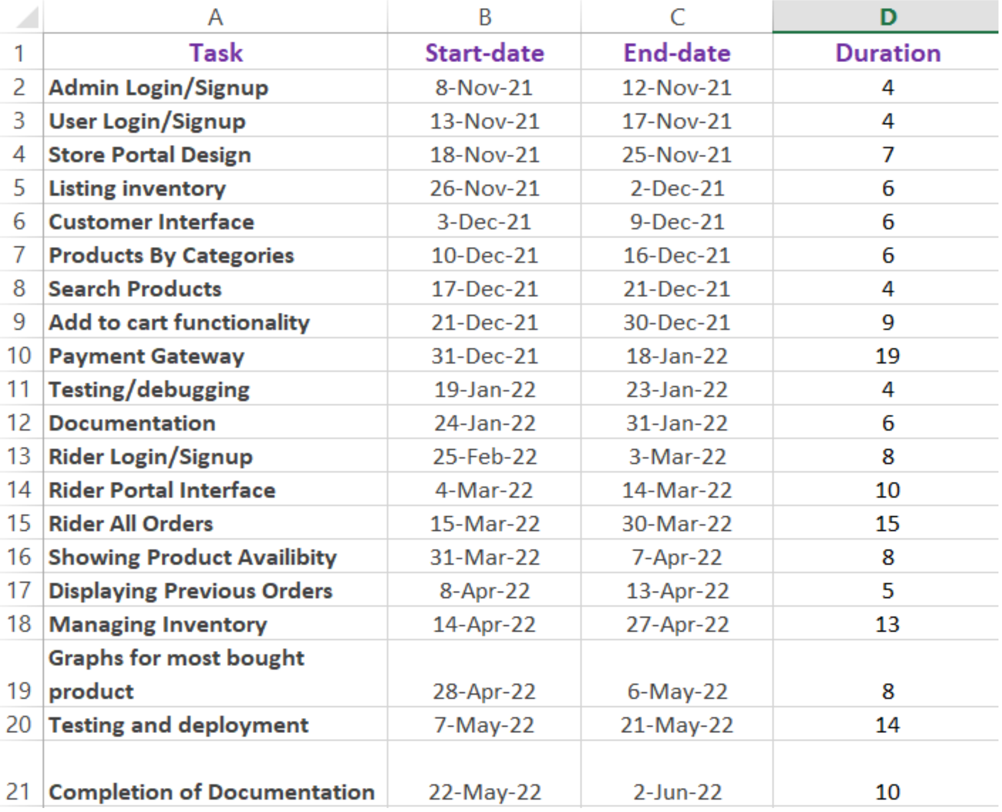
Both of us have quite expertise in MERN Stack. We have covered course for ReactJs and currently we are enrolled in React-Native course, which helped us quite in choosing the platform.

10. Milestones

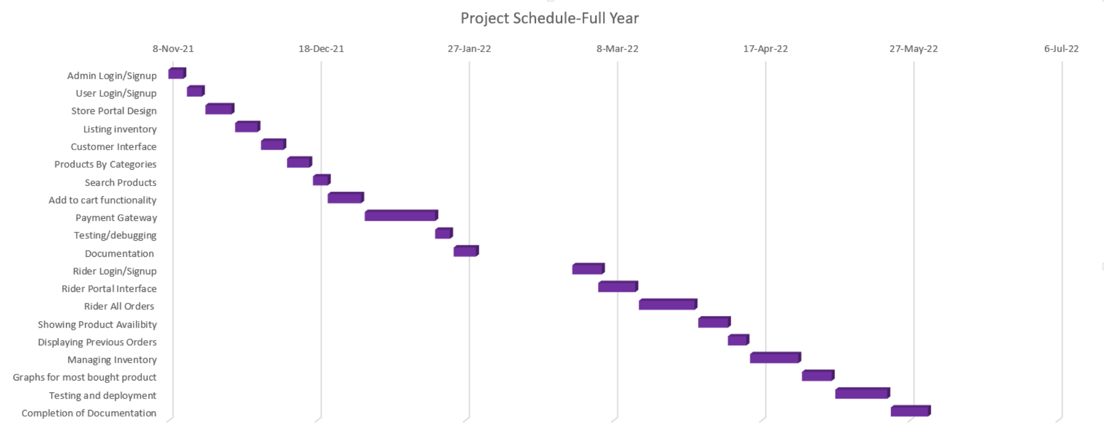
- Admin Login/Signup
- User Login/Signup
- Admin Portal Design
- Store portal designing
- Listing inventory
- Customer Interface
- Displaying products by categories
- Showing products availability i.e. either in stock or not
- Search Products
- Add to Cart interface
- Rider Login/Signup
- Rider portal designing
- Rider All orders tab
- Showing previous completed orders.
- Managing Inventory
- Displaying most bought products in form of charts/Graphs
- Testing and deployment
- Completion of Documentation

11. Project Schedule

Include or attach a preliminary 1-year project **Gantt chart** projecting a project timeline representing all milestones mentioned in para 10 above.



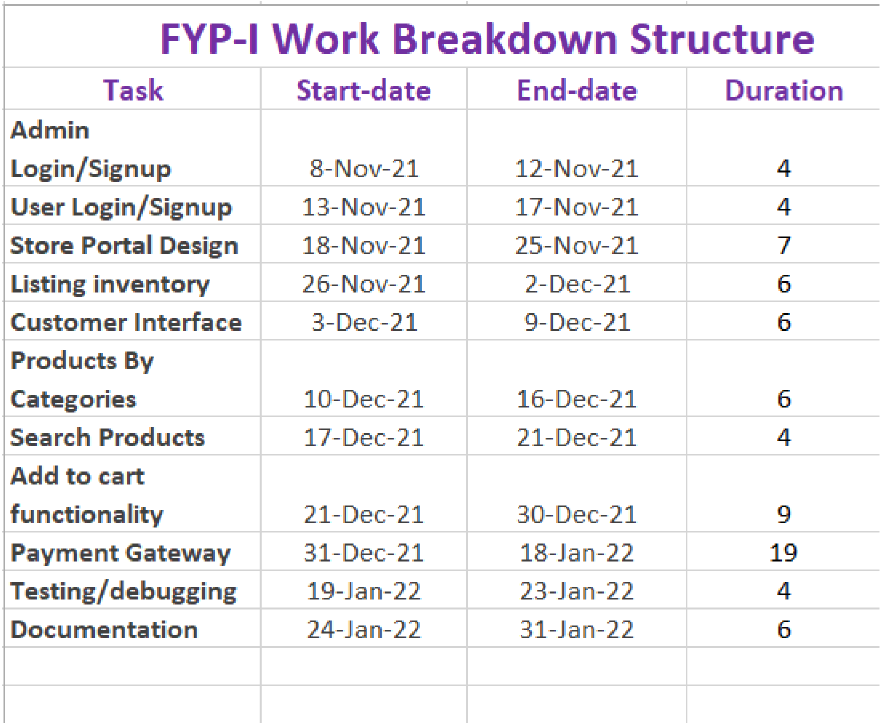
	A	B	C	D
1	**Task**	**Start-date**	**End-date**	**Duration**
2	**Admin Login/Signup**	8-Nov-21	12-Nov-21	4
3	**User Login/Signup**	13-Nov-21	17-Nov-21	4
4	**Store Portal Design**	18-Nov-21	25-Nov-21	7
5	**Listing inventory**	26-Nov-21	2-Dec-21	6
6	**Customer Interface**	3-Dec-21	9-Dec-21	6
7	**Products By Categories**	10-Dec-21	16-Dec-21	6
8	**Search Products**	17-Dec-21	21-Dec-21	4
9	**Add to cart functionality**	21-Dec-21	30-Dec-21	9
10	**Payment Gateway**	31-Dec-21	18-Jan-22	19
11	**Testing/debugging**	19-Jan-22	23-Jan-22	4
12	**Documentation**	24-Jan-22	31-Jan-22	6
13	**Rider Login/Signup**	25-Feb-22	3-Mar-22	8
14	**Rider Portal Interface**	4-Mar-22	14-Mar-22	10
15	**Rider All Orders**	15-Mar-22	30-Mar-22	15
16	**Showing Product Availibity**	31-Mar-22	7-Apr-22	8
17	**Displaying Previous Orders**	8-Apr-22	13-Apr-22	5
18	**Managing Inventory**	14-Apr-22	27-Apr-22	13
19	**Graphs for most bought product**	28-Apr-22	6-May-22	8
20	**Testing and deployment**	7-May-22	21-May-22	14
21	**Completion of Documentation**	22-May-22	2-Jun-22	10



	Completion of Documentation	Testing and deployment	Graphs for most bought product	Managing Inventory	Displaying Previous Orders	Showing Product Availibity	Rider All Orders	Rider Portal Interface	Rider Login/Signup	Documentation	Testing/debugging	Payment Gateway	Add to cart functionality	Search Products	Products By Categories	Customer Interface	Listing inventory	Store Portal Design	User Login/Signup	Admin Login/Signup
Start-date	22-May-22	7-May-22	28-Apr-22	14-Apr-22	8-Apr-22	31-Mar-22	15-Mar-22	4-Mar-22	25-Feb-22	24-Jan-22	19-Jan-22	31-Dec-21	21-Dec-21	17-Dec-21	10-Dec-21	3-Dec-21	26-Nov-21	18-Nov-21	13-Nov-21	8-Nov-21
■ Duration	10	14	8	13	5	8	15	10	8	6	4	19	9	4	6	6	6	7	4	4

12. Work Breakdown Structure

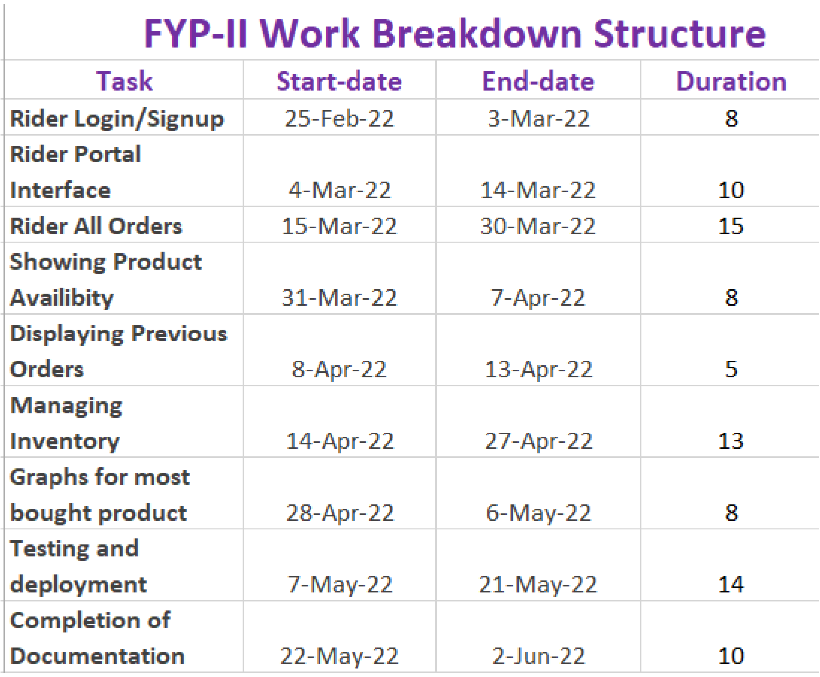
FYP-I WBS:



FYP-I Work Breakdown Structure

Task	Start-date	End-date	Duration
Admin Login/Signup	8-Nov-21	12-Nov-21	4
User Login/Signup	13-Nov-21	17-Nov-21	4
Store Portal Design	18-Nov-21	25-Nov-21	7
Listing inventory	26-Nov-21	2-Dec-21	6
Customer Interface	3-Dec-21	9-Dec-21	6
Products By Categories	10-Dec-21	16-Dec-21	6
Search Products	17-Dec-21	21-Dec-21	4
Add to cart functionality	21-Dec-21	30-Dec-21	9
Payment Gateway	31-Dec-21	18-Jan-22	19
Testing/debugging	19-Jan-22	23-Jan-22	4
Documentation	24-Jan-22	31-Jan-22	6

FYP-II WBS:



FYP-II Work Breakdown Structure

Task	Start-date	End-date	Duration
Rider Login/Signup	25-Feb-22	3-Mar-22	8
Rider Portal Interface	4-Mar-22	14-Mar-22	10
Rider All Orders	15-Mar-22	30-Mar-22	15
Showing Product Availibity	31-Mar-22	7-Apr-22	8
Displaying Previous Orders	8-Apr-22	13-Apr-22	5
Managing Inventory	14-Apr-22	27-Apr-22	13
Graphs for most bought product	28-Apr-22	6-May-22	8
Testing and deployment	7-May-22	21-May-22	14
Completion of Documentation	22-May-22	2-Jun-22	10

13. References

We are using references of below mentioned websites.

• Internet Web page: “Getting-Started with React Native". 10/02/2021
https://reactnative.dev/docs/getting-started

• Internet Web page: “Welcome to the MongoDB Documentation” https://docs.mongodb.com/

•InternetWeb page: “MongoDB NodeJS”
https://www.w3schools.com/nodejs/nodejs_mongodb.asp

II. Software Requirement Specification

1. Introduction

1.1 Purpose

E-Distributor is a platform developed for a client where he will be able to manage his stock and inventory along with selling the products online. For even clear picture use cases are defined that define user interaction with the system. This application will not only let the business manage their stock but will also help them to grow their business.

1.2 Document Conventions

Font style: Arial
Regular Text: Font size 11
Subheading: size 14 with bold
Main Heading: size 18 with bold

1.3 Intended Audience and Reading Suggestions

This document is intended to be read by developer, tester, project advisor, panel members and owner of business. This Software requirement specification contains functional and non-functional requirements, user interface, and features of project and use cases of features to demonstrate working of features.

1.4 Product Scope

The scope of project is develop a platform for our client “Subash Traders” where they can manage inventory and stock of their business i.e. a platform where our client can monitor his store and provide their customers an environment where they can make online purchase of products.

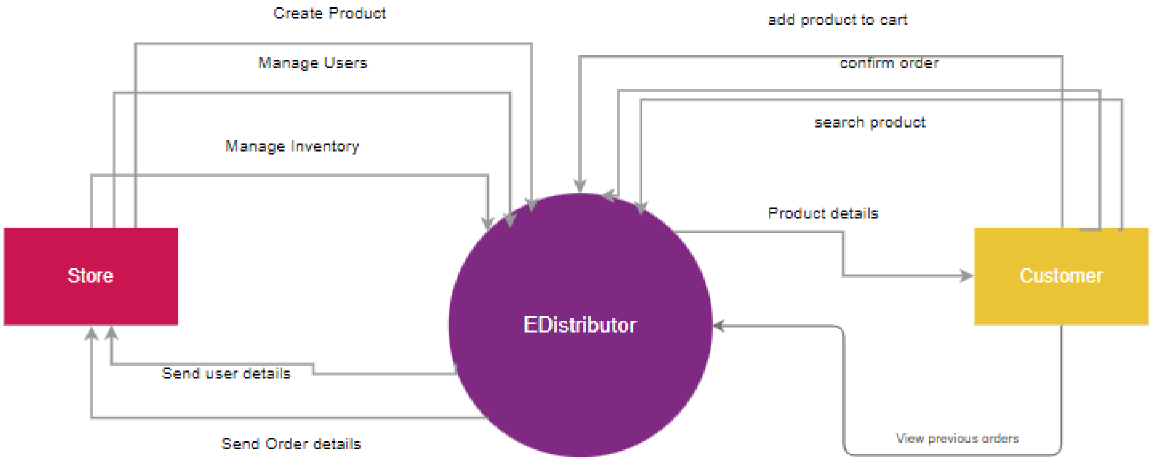
1.5 References

We are using references of below mentioned websites.

- Internet Web page: “Getting-Started with React Native". 10/02/2021 https://reactnative.dev/docs/getting-started
- Internet Web page: “Welcome to the MongoDB Documentation” https://docs.mongodb.com/
- Internet Web page: “MongoDB NodeJS” https://www.w3schools.com/nodejs/nodejs_mongodb.asp
- Internet Web page: “Google Maps Platform” https://developers.google.com/maps

2. Overall Description

2.1 Product Perspective



2.2 Product Functions

- Login/Signup
- Search products
- Add to cart
- View previous orders
- Confirm Order
- Display product category wise
- Store managing users
- Store listing inventory
- Add/Update/Delete inventory(Management of inventory)
- Orders list
- Shipment details
- Cart total calculation
- Graphs for statistics

2.3 User Classes and Characteristics

In our application there are multiple users and are mentioned below.

1. **Store:**
    - Store can login
    - Add products to inventory
    - Update/Delete the products
    - View Orders
    - View Customers
    - Check graphs for most bought inventory
    - Manage users i.e. remove

2. **Customer:**
    - Can login/Signup
    - View products
    - View Categories
    - View previous orders
    - Add product to cart
    - Purchase product
    - Search products

    .

2.4 Operating Environment

Software Platform: Android phone with internet connection for customers and for store any web-browser.
Hardware Platform: Android Mobile and desktop.

2.5 Design and Implementation Constraints

Software Constraints:

Our mobile application is needed to be installed in mobile phones and mobile must have android with working internet and store needs to have any working web browser for management.

Hardware Constraints:

All android phones with basic hardware specifications will be able to use this application.

2.6 User Documentation

The application is self- explanatory, therefore there is no need for a user manual. No video tutorials are included by developers. However, System Requirement

Specification (SRS), System Design Specification (SDS), System Test Description (STD) will be provided to client.

2.7 Assumptions and Dependencies

It is assumed that the project will for now be operated on dummy data, however after completing few phases of application, the application will be using client's real data. It is assumed that customers are well aware of technology and usage of mobile phone. Since the application is android based, an android mobile phone with internet connectivity is mandatory. User must be logged in to application to purchase products. The application is simplified and user friendly, basic knowledge of using applications is required.

3. External Interface Requirements

3.1 User Interfaces

We will be using windows GUI Standards in our application. User can interact with our mobile app through touch screen and in our web application user can interact with mouse and keyboard.

There will be login screen from where user will into the system and perform further functionalities.

3.2 Hardware Interfaces

This application can run on any android phone. The store can access the web portal on computer or laptop. Although for optimal performance and faster results high speed internet connection is required.

3.3 Software Interfaces

This application needs android environment for running. Store can operate the web portal on any browser. Tools used for development are Android Studio, Visual Studio Code and postman. MongoDb, the NoSQL database is used for creating the database. For frontend ReactJs and React native will be used for web-based portal and mobile application respectively. For backend NodeJs will be used.

3.4 Communications Interfaces

There is no specific protocol used for making communication. However API's will be used for communicating frontend with backend.

4. System Features

4.1 Signup

<table>
<tr><td><b>Use case name:</b></td><td>Signup</td></tr>
<tr><td><b>Use case ID:</b></td><td>1</td></tr>
<tr><td><b>Summary:</b></td><td>Signup will help you create your account and get you registered in the database</td></tr>
<tr><td><b>Actors:</b></td><td>Customer.</td></tr>
<tr><td><b>Preconditions:</b></td><td>The account should not be existing before.</td></tr>
<tr><td><b>Basic course of event/happy path</b></td><td>
<table>
<tr><th>Action</th><th>System Response</th></tr>
<tr><td>i. Enter valid credentials</td><td>ii. Check record in database</td></tr>
<tr><td></td><td>iii. If account doesn’t exist create account and registers in database.</td></tr>
<tr><td></td><td></td></tr>
</table>
</td></tr>
<tr><td><b>Alternate Scenario:</b></td><td>1. If entered credentials are invalid or not matching the requirements then “Invalid credentials” message will be displayed.
2. If any field is left empty, then “enter respective credential” message will be displayed.</td></tr>
<tr><td><b>Post Conditions:</b></td><td>Actor’s account will be created and next screen will be displayed</td></tr>
<tr><td><b>Author name:</b></td><td>Muskan,Harmeet</td></tr>
</table>

4.2 Login

Use case name:	Login

<table>
<tr><td><b>Use case ID:</b></td><td>2</td></tr>
<tr><td><b>Summary:</b></td><td>Login will let the respective actor enter into their system.</td></tr>
<tr><td><b>Actors:</b></td><td>Store, Customer</td></tr>
<tr><td><b>Preconditions:</b></td><td>Actor should be registered to the system’s database.</td></tr>
<tr><td><b>Basic course of event/happy path</b></td><td>
<table>
<tr><th>Action</th><th>System Response</th></tr>
<tr><td>i. Actor enters username and password.</td><td>ii. System verifies the record in database.</td></tr>
<tr><td></td><td>iii. If entered credentials match with record then it will let actor enter into system</td></tr>
</table>
</td></tr>
<tr><td><b>Alternate Scenario:</b></td><td>1. If Email and password is invalid then “Login failed or User doesn’t exist” message is displayed.
2. If account details is empty then “Please Enter credentials” message is displayed.</td></tr>
<tr><td><b>Post Conditions:</b></td><td>Actor will be redirected to their respective screens .i.e. User can view products screen</td></tr>
<tr><td><b>Author name:</b></td><td>Muskan,Harmeet</td></tr>
</table>

4.3 Add Product to inventory

<table>
<tr><td><b>Use case name:</b></td><td>Add products to inventory</td></tr>
<tr><td><b>Use case ID:</b></td><td>3</td></tr>
<tr><td><b>Summary:</b></td><td>This feature lets store add products to the inventory</td></tr>
<tr><td><b>Actors:</b></td><td>Store</td></tr>
<tr><td><b>Preconditions:</b></td><td>Authentic data of product should be available</td></tr>
<tr><td><b>Basic course of event/happy path</b></td><td>
<table>
<tr><th>Action</th><th>System Response</th></tr>
<tr><td>I. Actor clicks add product</td><td>II. Display form for product details</td></tr>
<tr><td>III. User click submit button</td><td>IV. Verifies the entered details and submit record to database</td></tr>
</table>
</td></tr>
<tr><td><b>Alternate Scenario:</b></td><td>If invalid details are entered product will be not be store in database</td></tr>
<tr><td><b>Post Conditions:</b></td><td>Actor will be able to see list of added products.</td></tr>
<tr><td><b>Author name:</b></td><td>Muskan,Harmeet</td></tr>
</table>

4.4 Update Inventory

Use case name:	Update inventory
Use case ID:	4
Summary:	This feature lets store update products(i.e. add or remove) to the inventory
Actors:	Store
Preconditions:	Product record should exist in database before update.

<table>
<tr><td><b>Basic course of event/happy path</b></td><td><table><tr><th>Action</th><th>System Response</th></tr><tr><td>I. Actor selects the product</td><td>II. Returns product details for updating product</td></tr><tr><td>III. Select remove or add product</td><td>IV. Update the product in database</td></tr></table></td></tr>
<tr><td><b>Alternate Scenario:</b></td><td>1. If actor doesn’t select update option, product details will not be updated</td></tr>
<tr><td><b>Post Conditions:</b></td><td>Actor will be able to see update product details</td></tr>
<tr><td><b>Author name:</b></td><td>Muskan,Harmeet</td></tr>
</table>

4.5 View Orders

<table>
<tr><td><b>Use case name:</b></td><td>View orders</td></tr>
<tr><td><b>Use case ID:</b></td><td>5</td></tr>
<tr><td><b>Summary:</b></td><td>This feature will let actor check orders</td></tr>
<tr><td><b>Actors:</b></td><td>Store</td></tr>
<tr><td><b>Preconditions:</b></td><td>Order should exist in database.</td></tr>
<tr><td><b>Basic course of event/happy path</b></td><td><table><tr><th>Action</th><th>System Response</th></tr><tr><td>I. Click orders tab</td><td>II. Fetch order details from database</td></tr><tr><td>III. View orders</td><td></td></tr></table></td></tr>
<tr><td><b>Alternate Scenario:</b></td><td>1. If orders doesn’t exist, no orders made message will be displayed</td></tr>
<tr><td><b>Post Conditions:</b></td><td>Actor will see how much orders are made.</td></tr>
<tr><td><b>Author name:</b></td><td>Muskan,Harmeet</td></tr>
</table>

4.6 Search Products

<table>
<tr><td><b>Use case name:</b></td><td>Search Products</td></tr>
<tr><td><b>Use case ID:</b></td><td>6</td></tr>
<tr><td><b>Summary:</b></td><td>Actor will be able to search products</td></tr>
<tr><td><b>Actors:</b></td><td>Customer</td></tr>
<tr><td><b>Preconditions:</b></td><td>Products should be available to search</td></tr>
<tr><td><b>Basic course of event/happy path</b></td><td><table>
<tr><th>Action</th><th>System Response</th></tr>
<tr><td>I. Click search bar</td><td></td></tr>
<tr><td>II. Input the product name and click search button</td><td>III. Match the result with database and display searched product</td></tr>
</table></td></tr>
<tr><td><b>Alternate Scenario:</b></td><td>1. If actor enters invalid name, “Product doesn’t exist” message will be displayed.</td></tr>
<tr><td><b>Post Conditions:</b></td><td>Searched product will be displayed</td></tr>
<tr><td><b>Author name:</b></td><td>Muskan,Harmeet</td></tr>
</table>

4.7 Add to cart

Use case name:	Add to cart
Use case ID:	7
Summary:	This feature will let actor add selected products to cart.
Actors:	Customer
Preconditions:	Product should be added to cart

Basic course of event/happy path	**Action**	**System Response**
	I. User clicks “add to cart” button	II. System will display selected products
		III. Ask for shipping details
		IV. Display total
	V. Click confirm order	VI. Order will be confirmed and saved to orders table.
Alternate Scenario:	1. If user doesn’t add product to cart, order will not be confirmed.	
Post Conditions:	Order confirmation message will be displayed to screen	
Author name:	Muskan,Harmeet	

4.8 Display previous orders

Use case name:	Display previous orders	
Use case ID:	8	
Summary:	It will let actor see their previous orders	
Actors:	Customer	
Preconditions:	Orders should be made in past	
Basic course of event/happy path	**Action**	**System Response**
	I. Actor click orders tab	II. Display previous orders
Alternate Scenario:	1. If previously orders were not made, message “No orders yet” will be displayed.	
Post Conditions:	Order confirmation message will be displayed to screen	

Author name:	Muskan,Harmeet

4.9 Payment Calculation

<table>
<tr><td><b>Use case name:</b></td><td>Payment Calculation</td></tr>
<tr><td><b>Use case ID:</b></td><td>9</td></tr>
<tr><td><b>Summary:</b></td><td>This tab will let actor check total payment based on items added to cart</td></tr>
<tr><td><b>Actors:</b></td><td>Customer</td></tr>
<tr><td><b>Preconditions:</b></td><td>Actor must have added products to carts to check the total sum.</td></tr>
<tr><td><b>Basic course of event/happy path</b></td><td>
<table>
<tr><th>Action</th><th>System Response</th></tr>
<tr><td>I. Actor add products to cart</td><td>II. Display all products in cart</td></tr>
<tr><td></td><td>III. Display total sum of all products in cart.</td></tr>
</table>
</td></tr>
<tr><td><b>Alternate Scenario:</b></td><td>1. If actor don’t add products to cart no total payment will be displayed.</td></tr>
<tr><td><b>Post Conditions:</b></td><td>Actor will be see the total sum of products he/she will be purchasing.</td></tr>
<tr><td><b>Author name:</b></td><td>Muskan,Harmeet</td></tr>
</table>

4.10 Shipping Details

Use case name:	Shipping Details

<table>
<tr><td><b>Use case ID:</b></td><td>10</td></tr>
<tr><td><b>Summary:</b></td><td>This tab will ask actor for shipping details.</td></tr>
<tr><td><b>Actors:</b></td><td>Customer</td></tr>
<tr><td><b>Preconditions:</b></td><td>Actor must have added product to cart to enter the shipping details</td></tr>
<tr><td><b>Basic course of event/happy path</b></td><td><table>
<tr><th>Action</th><th>System Response</th></tr>
<tr><td></td><td>i. Display Shipping details form</td></tr>
<tr><td>ii. Enter shipping details</td><td>iii. Save shipping details to database</td></tr>
</table></td></tr>
<tr><td><b>Alternate Scenario:</b></td><td>1. If products are not added to cart, no shipping details will be asked.</td></tr>
<tr><td><b>Post Conditions:</b></td><td>Actor will be able to get the product delivered at entered destination</td></tr>
<tr><td><b>Author name:</b></td><td>Muskan,Harmeet</td></tr>
</table>

4.11 Graphs

Use case name:	Graphs
Use case ID:	11
Summary:	This tab will let actor see graphs.
Actors:	Store
Preconditions:	Actor must have uploaded the products and products should be sold before to see graphs

Basic course of event/happy path	**Action**	**System Response**
	IV. Actor clicks graphs tab	V. Show statistics by graph
	VI. Will get to know about which products are being sold most	
Alternate Scenario:	1. If products were not sold, no graphs will be displayed	
Post Conditions:	Actor will be able to monitor sell of products.	
Author name:	Muskan,Harmeet	

4.12 Manage Users

Use case name:	Manage users	
Use case ID:	14	
Summary:	By this feature store will manage users on the system	
Actors:	Store	
Preconditions:	User’s record should exist in database	
Basic course of event/happy path	**Action**	**System Response**
	I. Actor will click manage user.	II. Open list of users with options of managing user
	III. Select respective option.	IV. Update in database.
Alternate Scenario:	1. If any option is not selected record will not be updated.	

Post Conditions:	User collection will be updated
Author name:	Muskan,Harmeet

5. Other Nonfunctional Requirements

5.1 Performance Requirements

System requires android operating environment to perform efficiently. Application is considered to perform in real time environment. The application and web-based platform will be available only when there is good internet connection.

5.2 Safety Requirements

For safety reasons user can view his data only when he is registered in the database and logged in to the system. Backup will be made of database in any case of failure or crash to prevent data loss. Furthermore password will be encrypted.

5.3 Security Requirements

User's information will be stored in database. That information cannot be accessed by other user. The application is safe to use and will use encryption for security.

5.4 Software Quality Attributes

Software quality assurance is mandatory for every software to identify errors and flaws in code and design to correct it within time. It ensures the product is competitive and secure. Quality attributes that will be ensured in our systems are:

- Correctness: The application designed will work as expected and it can be achieved through testing.
- Reliability: Application will be reliable as the data will be secured.
- Maintenance: After launch of product any errors, changes and modifications will be ensured to keep the app updated and keep up with customer needs.
- Usability: The application is user-friendly, user won't find its features difficult.

5.5 Business Rules

This application is built for customers and store. Only store can access its interface and there is separate interface for customers where they can search and buy the product. Once customer books the order it will not be cancelled.
Other Requirements

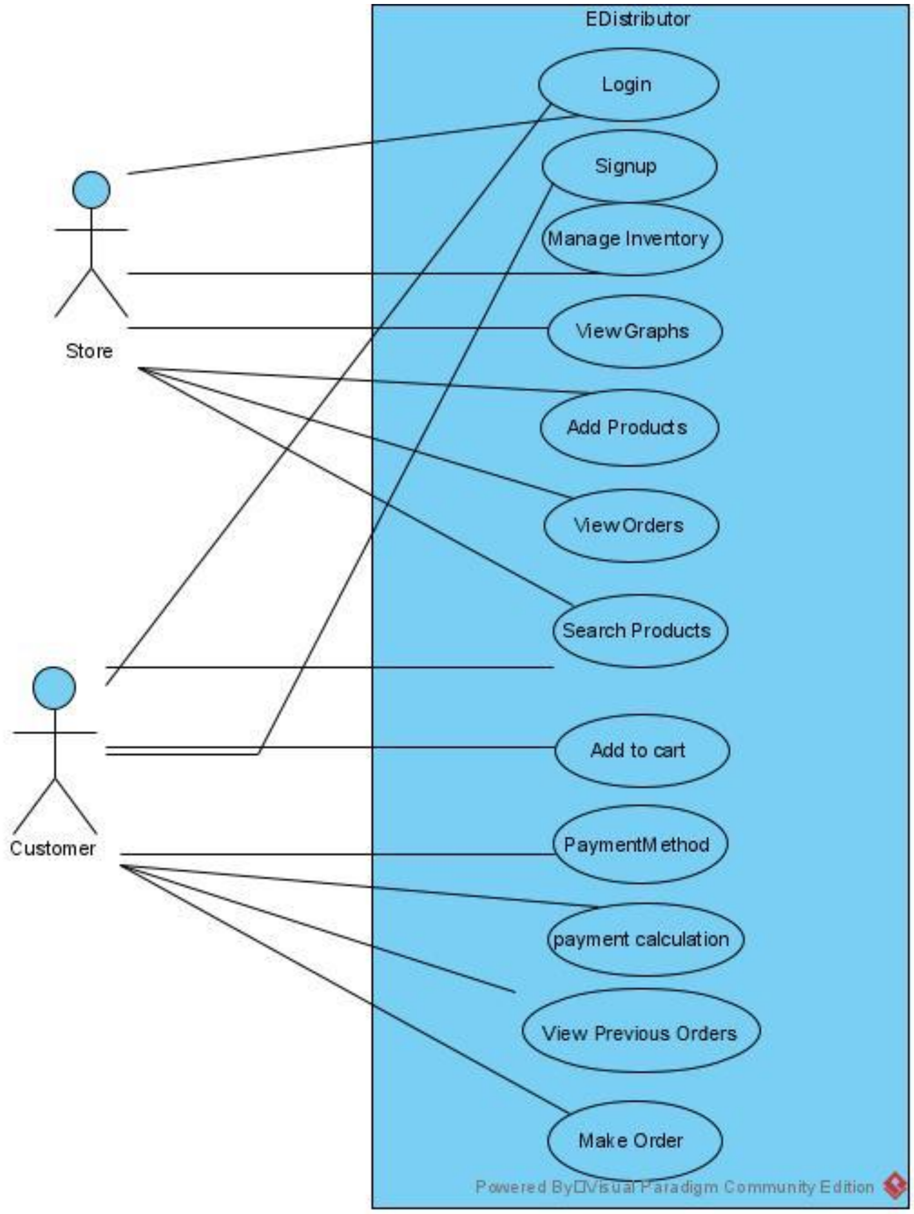
6. Other Requirements

Appendix A: Glossary

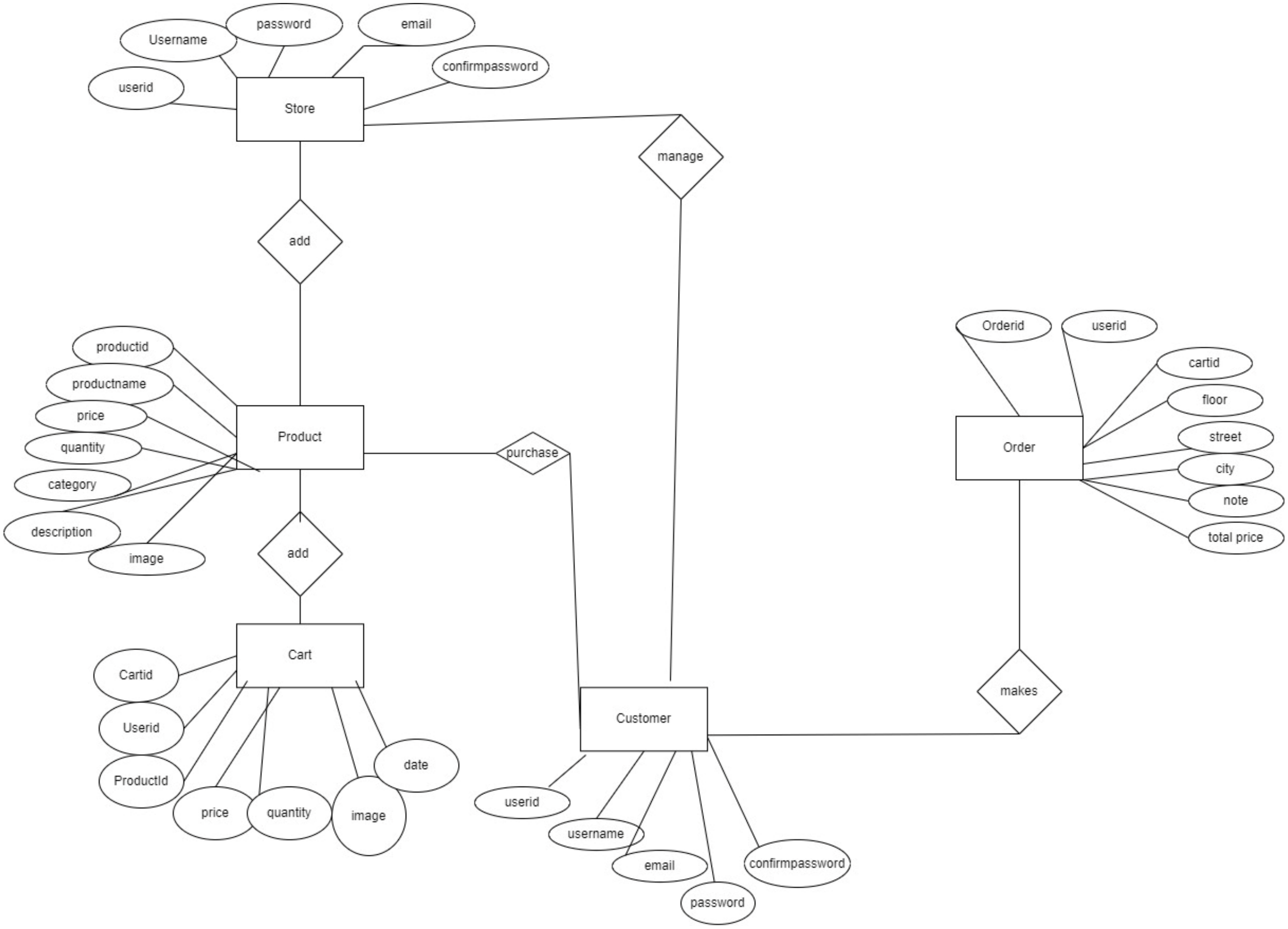
NA

Appendix B: Analysis Models

Use Case Diagram:



ERD:


Username
password
email
userid
confirmpassword
Store
manage
add
productid
productname
price
quantity
category
description
image
Product
purchase
add
Cart
Cartid
Userid
ProductId
price
quantity
image
date
Orderid
userid
cartid
floor
street
city
note
total price
Order
makes
Customer
userid
username
email
password
confirmpassword

III. Software Design Specifications

1. Introduction

1.1 Purpose of this document

This document illustrates the internal working of our project “E-distributors”. The purpose of this document is mainly to break down the product into components to describe in detail what the purpose of each component is and how it will be implemented.

1.2 Scope of the development project

The scope of project is develop a platform for our client “Subash Traders” where they can manage inventory and stock of their business i.e. a platform where our client can monitor his store and provide their customers an environment where they can make online purchase of products.

1.3 Definitions, acronyms, and abbreviations

TERMS	**DESCRIPTION**
API	Application programming Interface
UI	User Interface
No SQL	No Structured query language

1.4 References

We are using references of below mentioned websites.

- Internet Web page: “Getting-Started with React Native". 10/02/2021 https://reactnative.dev/docs/getting-started

- Internet Web page: “Welcome to the MongoDB Documentation” https://docs.mongodb.com/

- Internet Web page: “MongoDB NodeJS” https://www.w3schools.com/nodejs/nodejs_mongodb.asp

1.5 Overview of document

This System Design specification document contains the following information:

Section 1.0: An introduction to the document by explaining the purpose of the document, the scope providing terms and references, and a brief overview of each section within the document.

Section 2.0: An overview of the system architecture, system components and modules, relationships between the various components, and user interface issues without any technical details.

Section 3.0: A detailed description of each software component used in our product.

Section 4.0: Gives detail description of UI and design implementation rules.

Section 5.0: An explanation of the reusability of existing products and relationships within the product.

Section 6.0: A listing of the design decisions and tradeoffs made during the design phase. This section will help readers and users understand the reasoning for these decisions.

Section 7.0: NA

Section 8.0: It contains an appendix, which will contain diagrams and so on

2. System architecture description

This section provides an overview and rationale for the program's data and architectural design decisions.

2.1. Section Overview

This section describes the general constraints and usability of the system. Data design describes the structure of database. It consists of ERD. Program structure describes the complete structure of the software and detail description of each individual features.

2.2. General Constraints

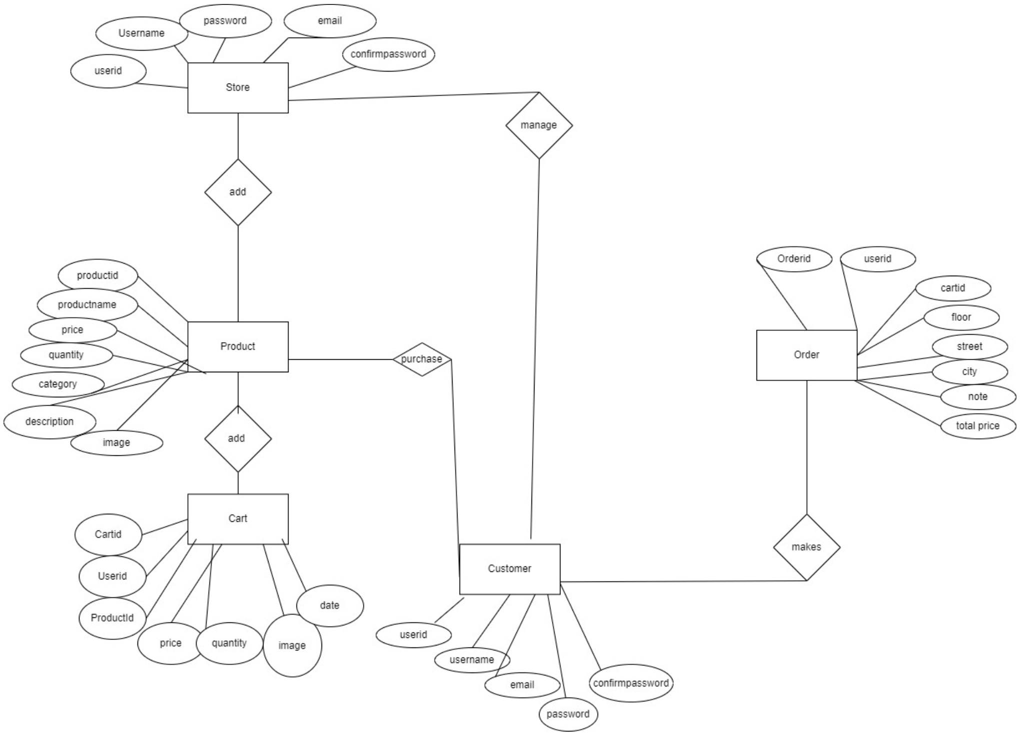
The product is easy to understand, easy to use, secure .The product needs web browser and android mobile for running with a fast internet connection. No external hardware or software is connected to our system directly or indirectly. The database is stored on a cloud based platform mongodb Atlas, which requires internet connection for retrieving and storing data.

- Hardware Constraint: Android phone with internet connection.
- Software Constraint: Software interfaces which are used are as following:
  - Android Studio
  - Mongo DB Atlas
  - Visual Code Studio
  - Postman

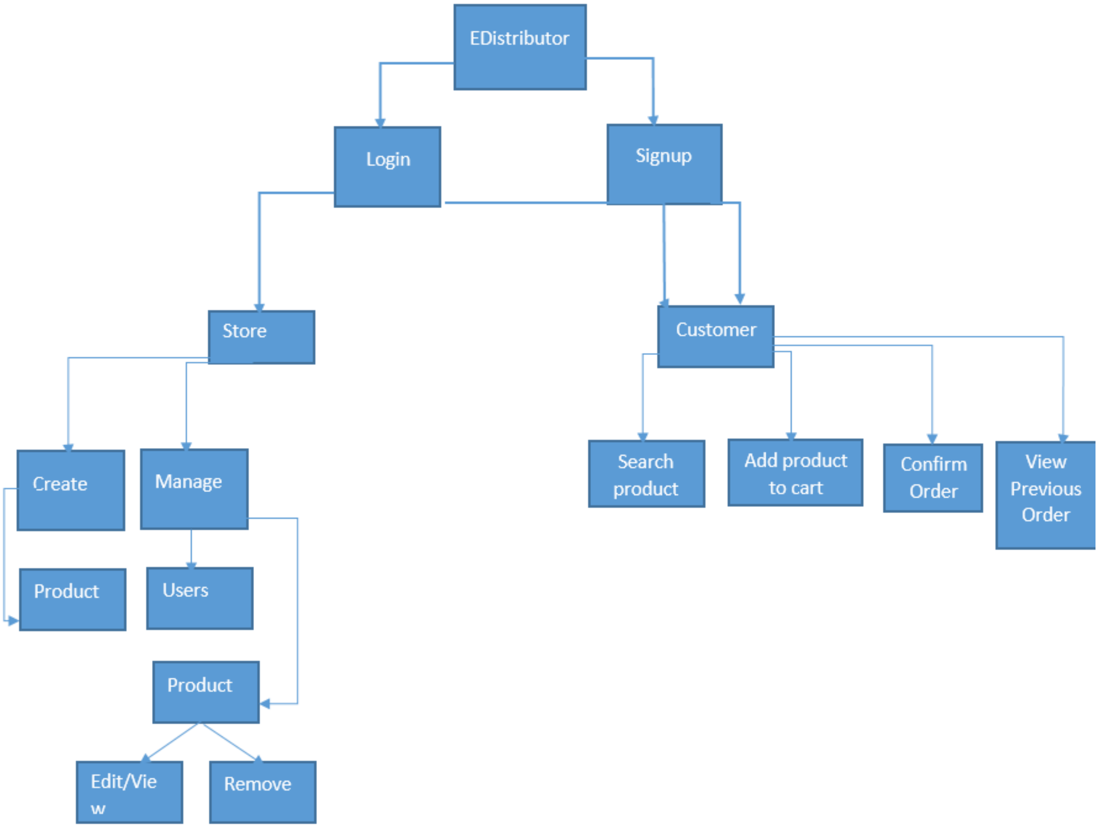
2.3. Data Design

The database of our system is located upon MongoDB Atlas, which needs internet connection for fetching data.

ERD:



2.4. Program Structure



2.5. Alternatives Considered

There was no alternative considered for designing our application. We have followed the MVC pattern Model, View and controller design pattern for our project. Each of these components are built to handle specific development aspects of an application.

3. Detailed description of components

3.1. Section Overview

In this section we'll describe the details of components in the system, so that reader can understand every component easily.

3.2. Component in Detail (include a sub-section for each component)

Identification	Login
Type	Authentication, an API to let user sign into the system.
Purpose	Login API is created as user can log in to the system and have their own environment.
Function	On login screen user enter his/her email and password, system authenticates the details and if details are valid ,user is redirected to his home screen
Subordinates	It contains unique id for every user, along with email and password. It redirects user to their homepage.
Dependencies	It depends upon registration, the user account must be registered in database else login function cannot be executed.
Interfaces	This is main interface of application, which lets user sign in to the system. The data entered on respective fields is verified with records present in database. If any field on form is left empty, respective error will be generated "Fields cannot be left empty", if System do not authenticate user error "Login failed" message will be generated.
Resources	No external hardware or software resources are required.

Processing	User will be asked to fill the fields, if fields are not filled, errors will be generated to fill the form.
Data	Users will be asked about email and password. The datatype for both are String.

Identification	Register
Type	Authentication, an API to let user register into the system.
Purpose	Register API is created so that user can create an account and his record will be stored in database.
Function	On register screen user enter his/her details, system will check if details entered are correct, then his account will be created.
Subordinates	It contains unique id for every user, along with username, email, password and confirm password.
Dependencies	It depends upon that all the fields should be filled correctly with proper format.
Interfaces	This is the interface which will create account for user. If user with existing email is registered before, an error will be generated. If fields are left empty error will be generated.
Resources	No external hardware or software resources are required.
Processing	User will be asked to fill the fields, if fields are not filled, errors will be generated to fill the form.
Data	Users will be asked about username, email, password and confirm password. The datatype for all are String.

Identification	Create Product
Type	A sub program under Store.
Purpose	It is used as store can create details for product and add products to database.
Function	A form will be opened where store can fill respective fields and register the product into database, so product details can be displayed to user.
Subordinates	It contains unique id for every product, product name, price, quantity, category and description.
Dependencies	It depends upon that all the fields should be filled correctly with proper format.
Interfaces	This is the interface which will register product to database. If product with existing name is registered before, an error will be generated. If fields are left empty error will be generated.
Resources	No external hardware or software resources are required.
Processing	Store will be asked to fill the fields, if fields are not filled, errors will be generated to fill the form.
Data	Product name, price, quantity, category and description are the fields. The data type for name, category and description is string while for price and quantity datatype is int.

Identification	List product
Type	A sub program under Store and Customer.
Purpose	It list down all the products on screen.
Function	It displays products available in database.
Subordinates	NA
Dependencies	It is dependent on one thing, which is that the product must exist in database so that it will be displayed on screen.
Interfaces	This is the interface which will display product to user.
Resources	No external hardware or software resources are required.
Processing	NA
Data	Product name, price, category and description will be displayed to customers, while store can see quantity as well.

Identification	Search product
Type	A subprogram and a method which helps in searching products.
Purpose	It is used for searching details of products.
Function	It displays products available in database.

Subordinates	NA
Dependencies	It is dependent on one thing, which is that the product must exist in database so that it will be displayed on screen.
Interfaces	This is the interface which will display product to user.
Resources	No external hardware or software resources are required.
Processing	NA
Data	Product name, price, category and description will be displayed to customers.

Identification	Add product to cart
Type	A subprogram for customer.
Purpose	It helps customer add products to cart.
Function	It generates total sum of all the products present in the cart along with shipping details
Subordinates	NA
Dependencies	It depends on that product must be added to cart

Interfaces	An interface which will let user check the total sum of his products.
Resources	No external hardware or software resources are required.
Processing	User will first select a product, add it to cart ,enter shipping details
Data	Cart contains user ID, product ID, product details and total sum of products.
Identification	Shipping Details
Type	A form for customer.
Purpose	It asks customer about the details of shipment.
Function	It asks shipment details, so that order should be delivered to correct destination.
Subordinates	NA
Dependencies	It depends on that product must be added to cart.
Interfaces	An interface which will ask user about shipment details.
Resources	No external hardware or software resources are required.
Processing	User will first select a product, add it to cart and enter shipping details
Data	Data like address, city will be asked.

Identification	Confirm Order.
Type	A submodule under customer.
Purpose	It asks customer the relevant details and confirm the order
Function	It asks about the shipment details, along with order summary and confirms the order and save it to database.
Subordinates	NA
Dependencies	It depends on that product must be added to cart.
Interfaces	It saves order record to database.
Resources	No external hardware or software resources are required.
Processing	User will first select a product, add it to cart, enter shipping details then order will be confirmed and saved to database.
Data	It contains all the data like shipment details and order summary.

4. User Interface Design

4.1. Section Overview

In this section we describe in detail about the designs of our system, the interface design rules for the purpose of usability, we have provided information about how our system will look, and few screenshots will be attached to preview the UI of application.

4.2. Interface Design Rules

the design rules which we have followed are as below:

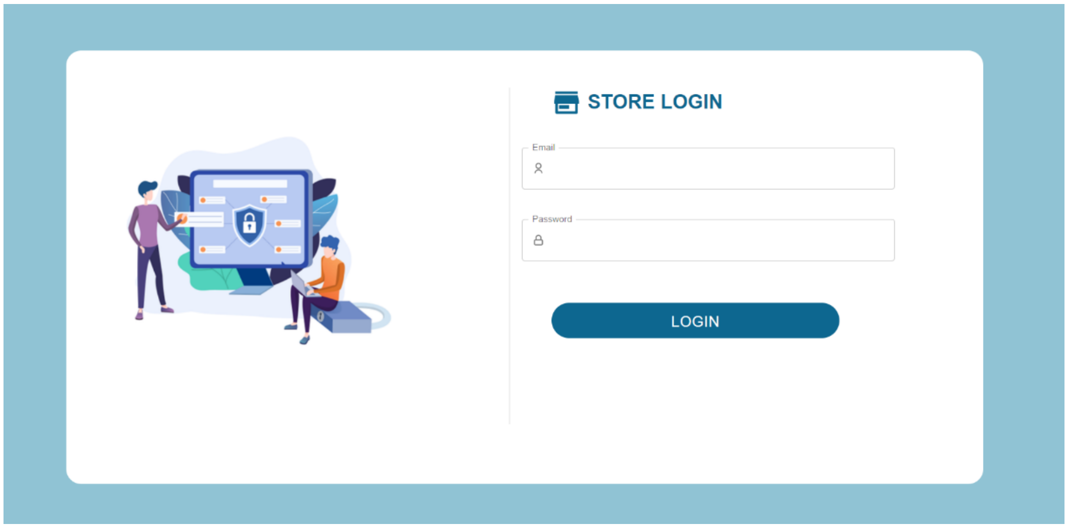
- **Offer simple error handling:** if any error is made by user, system should be able to recognize it and display an error message.
- **The Concept of Simplicity:** The UI of system should be simple, user friendly, with user control over interface, clarified, should generate accurate response according to user action.
- **The Concept of Visibility:** It is about UI of application, i.e. the alignment of screens, text-fields should be in proper order, there should be no spelling mistakes, position of buttons and logo should be on point, and application should be easy to navigate.

4.3. GUI Components

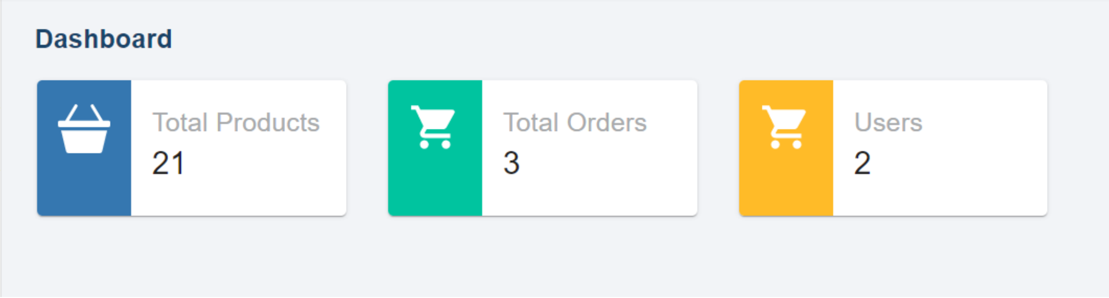
NA

4.4. Detailed Description

Store Login:



Store dashboard:

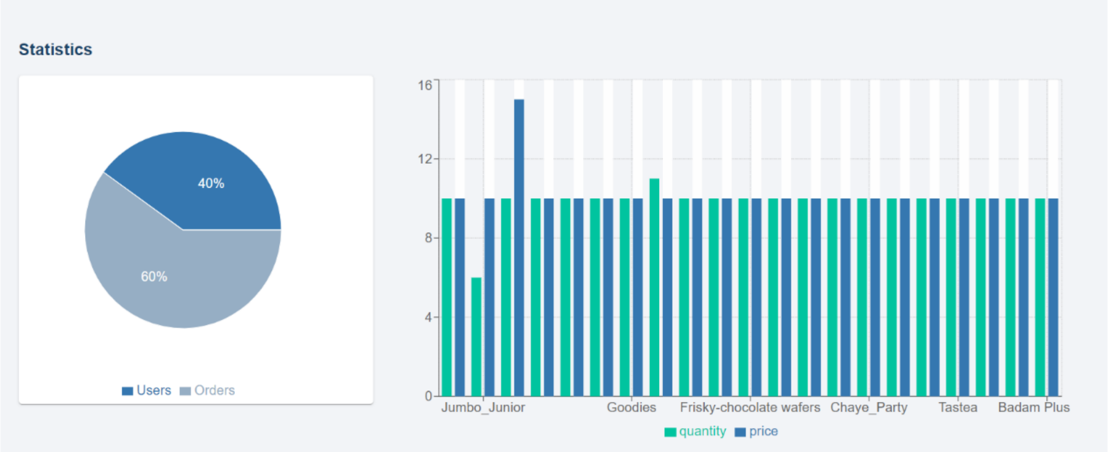


Dashboard

Total Products 21

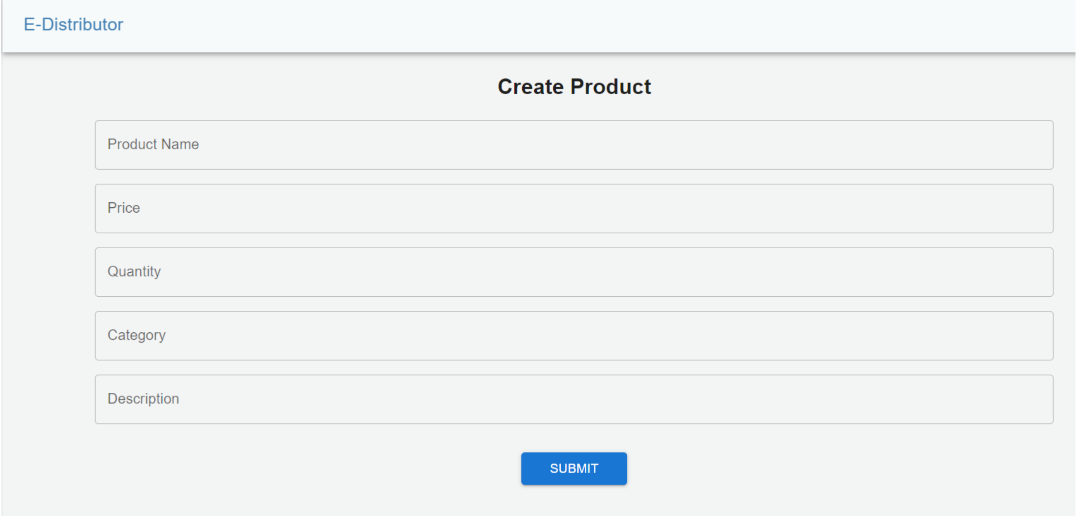
Total Orders 3

Users 2



Statistics

Create Product:



E-Distributor

Create Product

Product Name

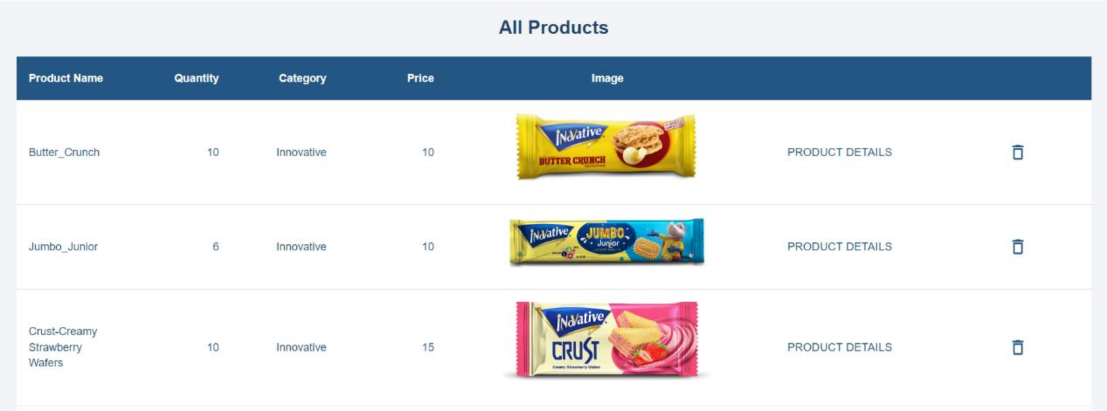
Price

Quantity

Category

Description

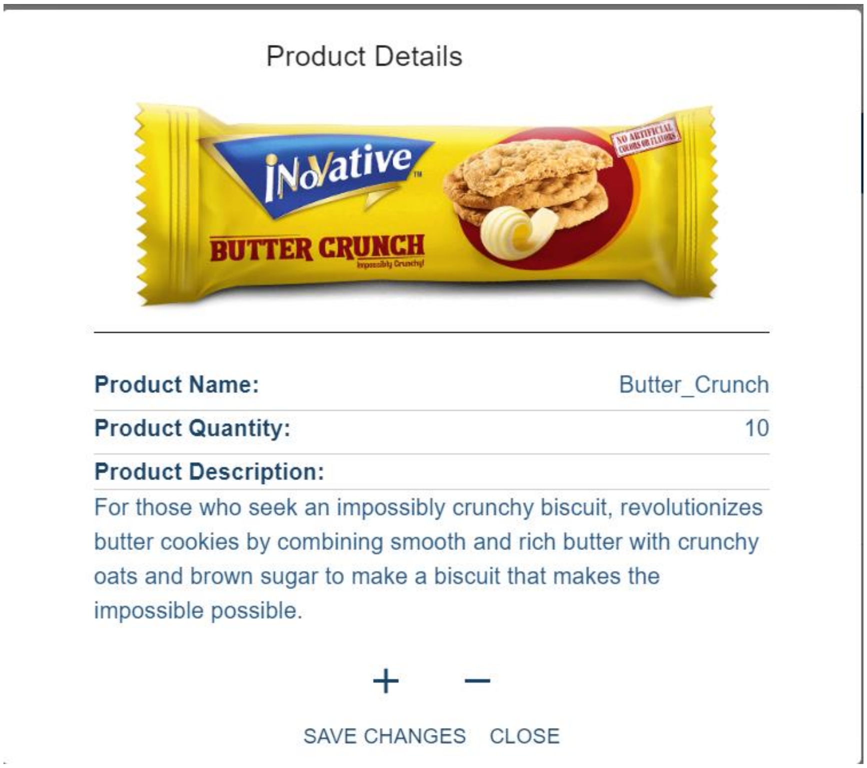
List of Products:



All Products

Product Name	Quantity	Category	Price	Image		
Butter_Crunch	10	Innovative	10		PRODUCT DETAILS	
Jumbo_Junior	6	Innovative	10		PRODUCT DETAILS	
Crust-Creamy Strawberry Wafers	10	Innovative	15		PRODUCT DETAILS	

ProductDetails:



Product Details

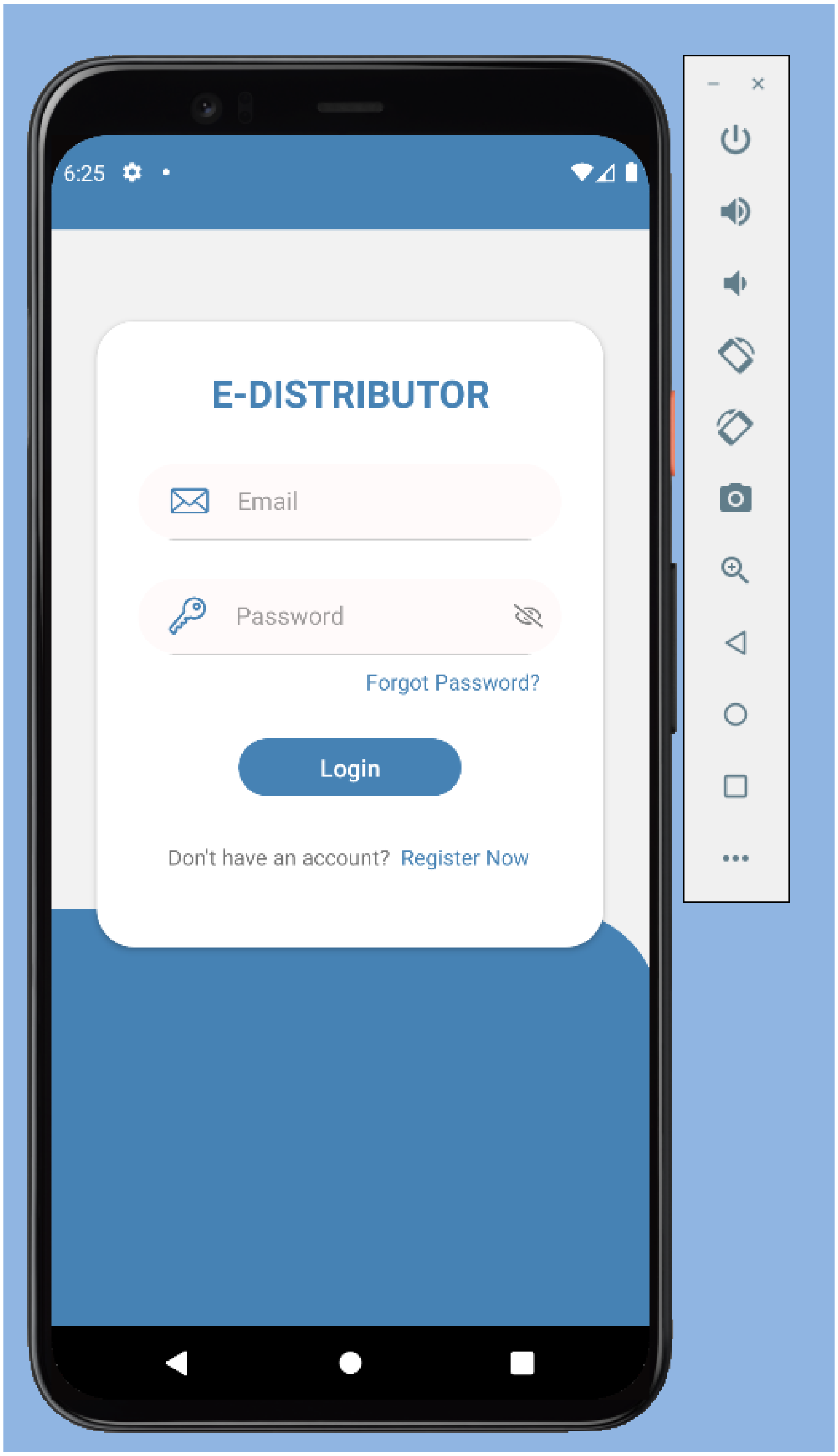
Product Name: Butter_Crunch

Product Quantity: 10

Product Description:

For those who seek an impossibly crunchy biscuit, revolutionizes butter cookies by combining smooth and rich butter with crunchy oats and brown sugar to make a biscuit that makes the impossible possible.

Login:



6:25

E-DISTRIBUTOR

Email

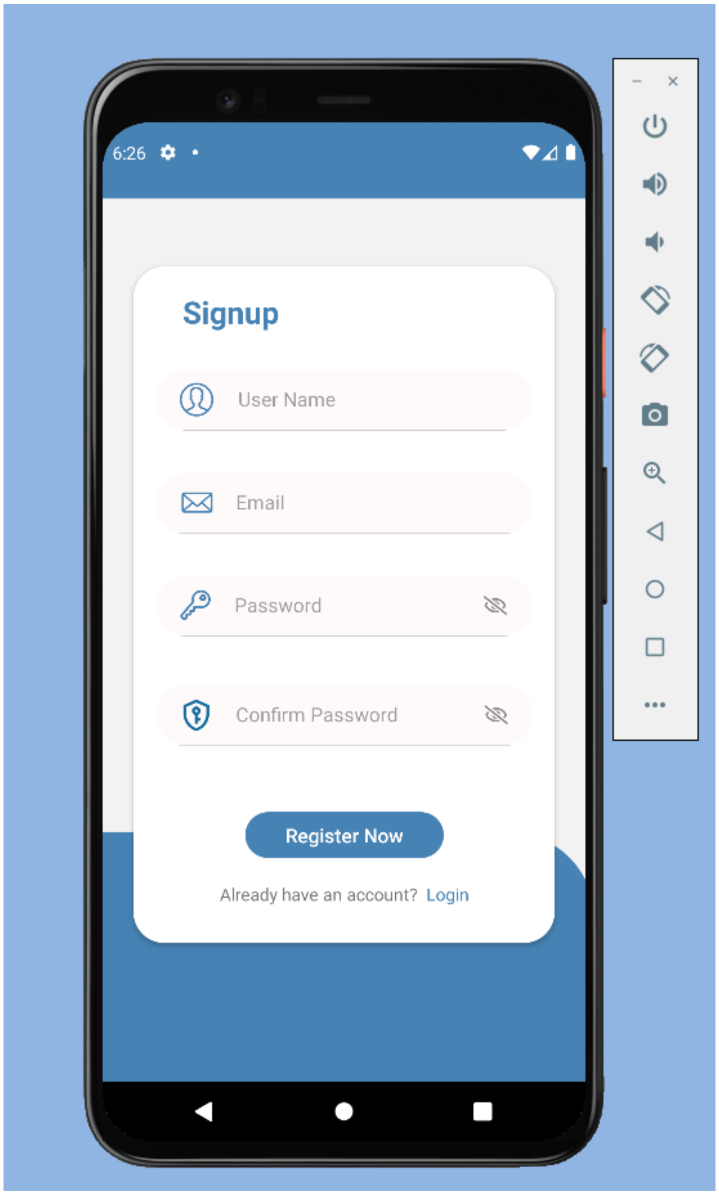
Password

Forgot Password?

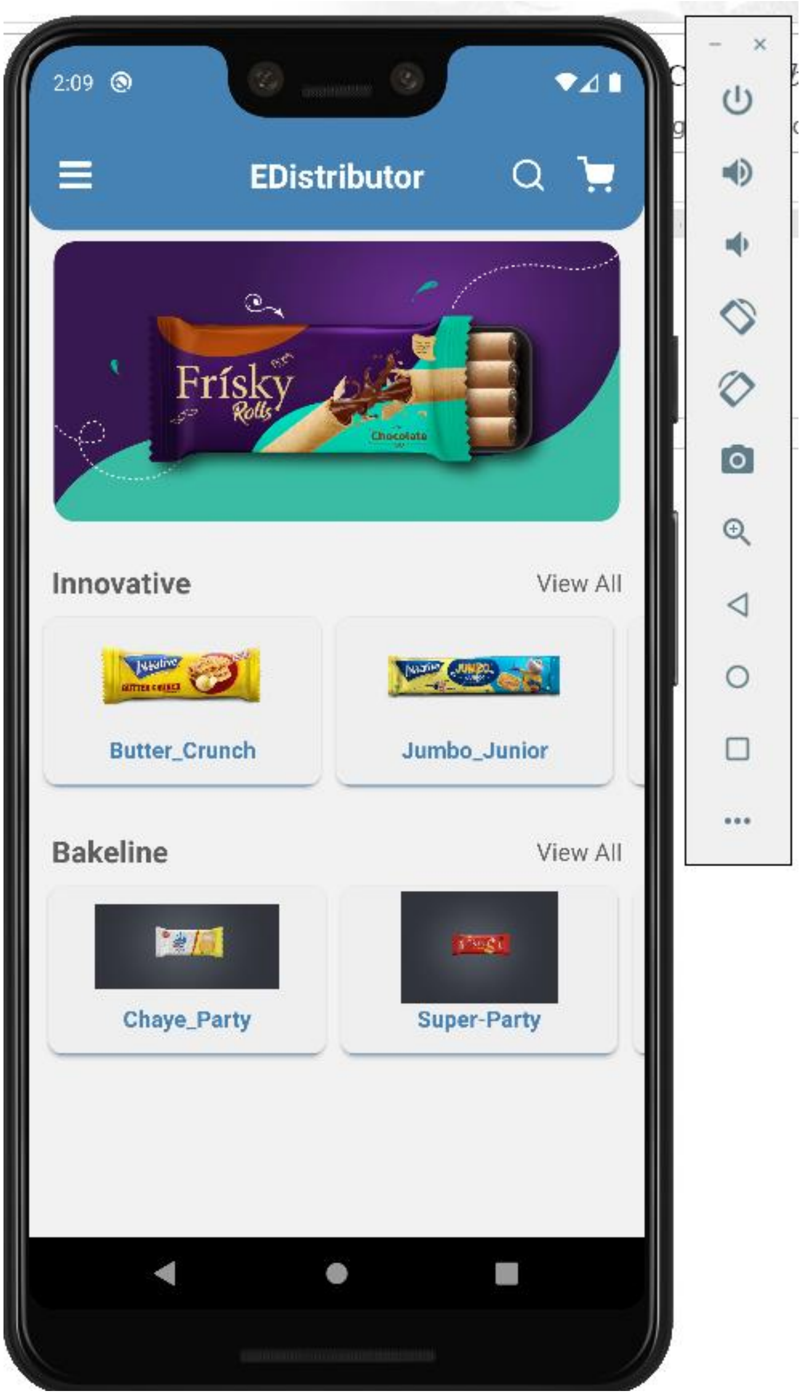
Login

Don't have an account? Register Now

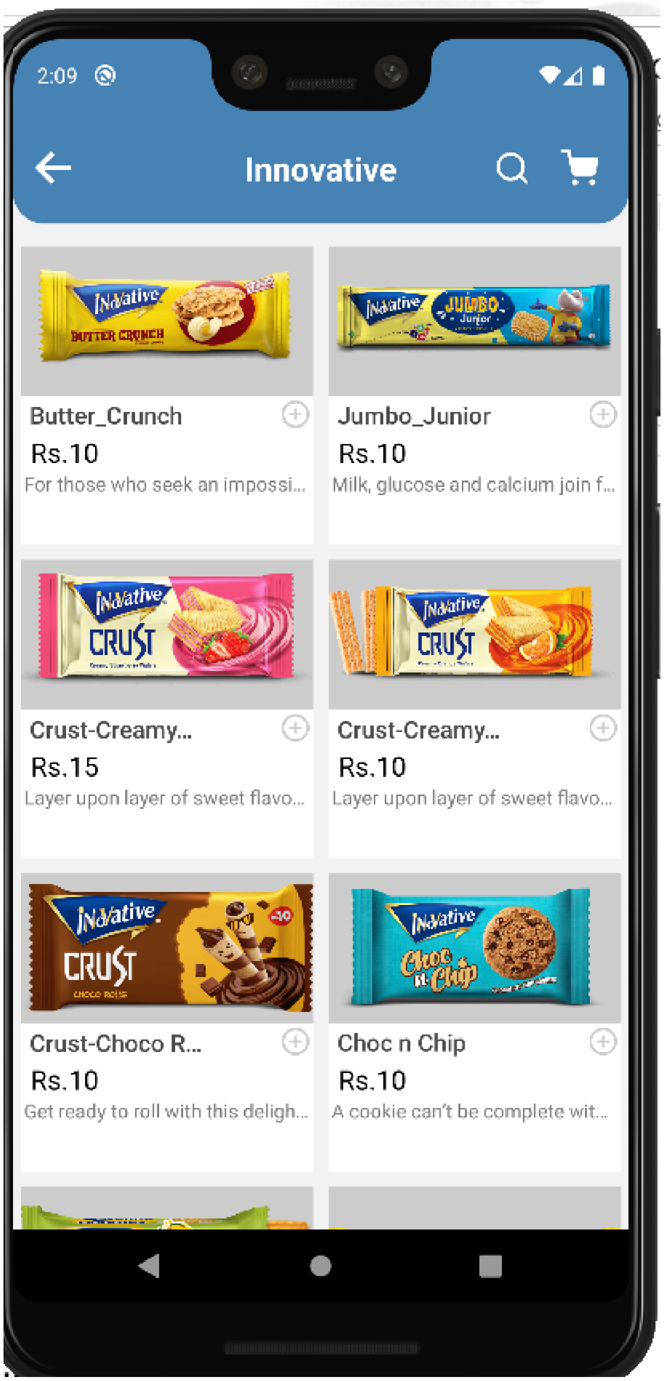
Register:


6:26
Signup
User Name
Email
Password
Confirm Password
Register Now
Already have an account? Login

Home Screen:

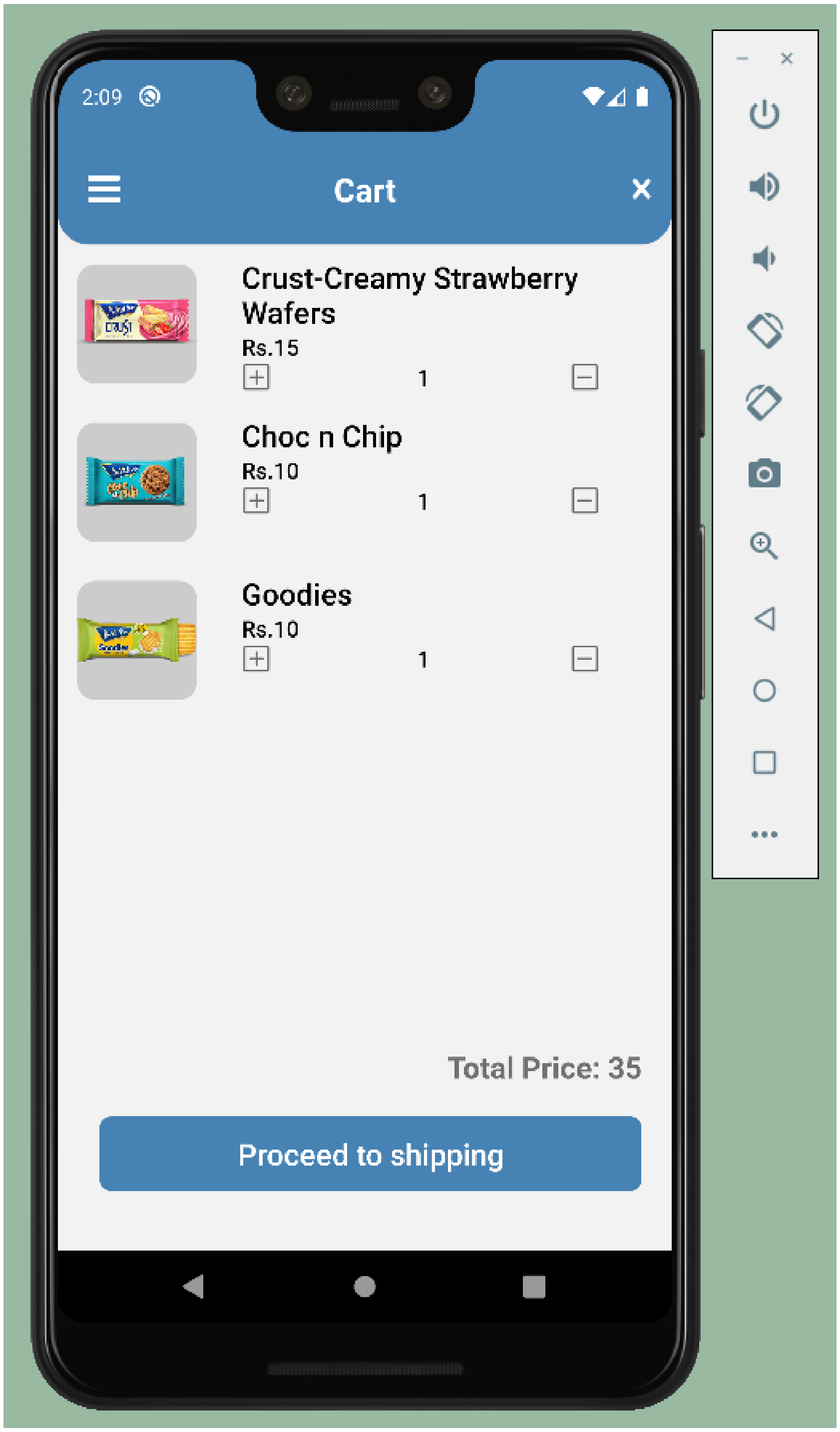

2:09
EDistributor
Frísky Rolls
Chocolate
Innovative
View All
Butter_Crunch
Jumbo_Junior
Bakeline
View All
Chaye_Party
Super-Party

Category:



2:09
Innovative
Butter_Crunch
Rs.10
For those who seek an impossi...
Jumbo_Junior
Rs.10
Milk, glucose and calcium join f...
Crust-Creamy...
Rs.15
Layer upon layer of sweet flavo...
Crust-Creamy...
Rs.10
Layer upon layer of sweet flavo...
Crust-Choco R...
Rs.10
Get ready to roll with this deligh...
Choc n Chip
Rs.10
A cookie can't be complete wit...

Cart:



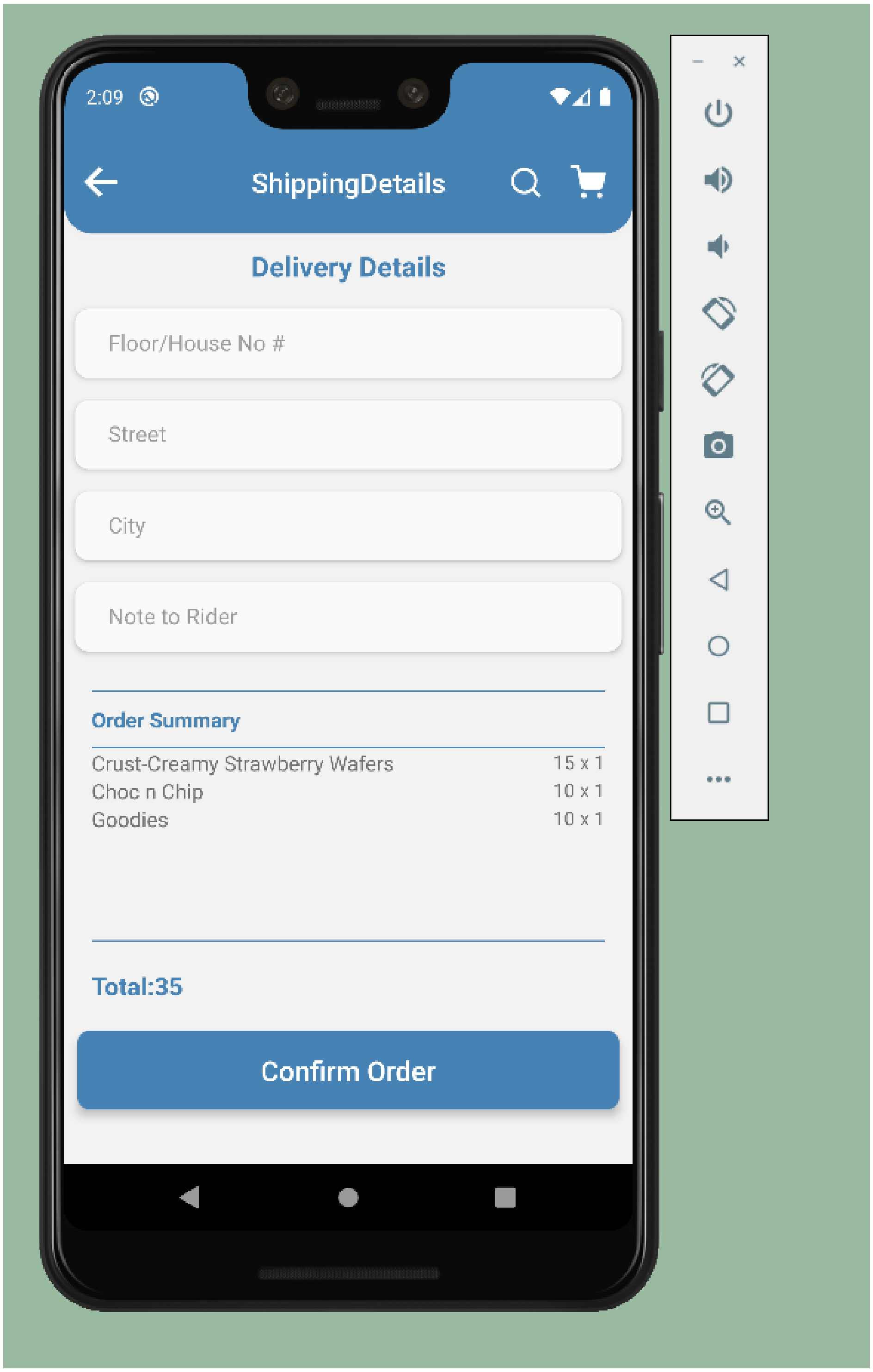
Crust-Creamy Strawberry Wafers
Rs.15
⊞ 1 ⊟

Choc n Chip
Rs.10
⊞ 1 ⊟

Goodies
Rs.10
⊞ 1 ⊟

Total Price: 35

Shipping details:



Delivery Details

Floor/House No #

Street

City

Note to Rider

Order Summary

Crust-Creamy Strawberry Wafers	15 x 1
Choc n Chip	10 x 1
Goodies	10 x 1

Total:35

5.0 Reuse and relationships to other products

Reusability allows developers to be more efficient because the same code can be developed once and used in many different applications. We have used material UI, a customizable, and accessible library of React components for our web application.

We have used some components of native base for designing our mobile application. It helped us to save our time than to customize each component by yourself.

6.0 Design decisions and tradeoffs

We have followed MVC pattern for design, main advantage of using MVC is that it offers separation of concern. It means that it divides the application into Model, Controller and View. We can easily maintain our application because of separation of concern. With help of this different person can work on different components of it like one can work on controllers, while other can work on designing the model.

7.0 Pseudocode for components

NA

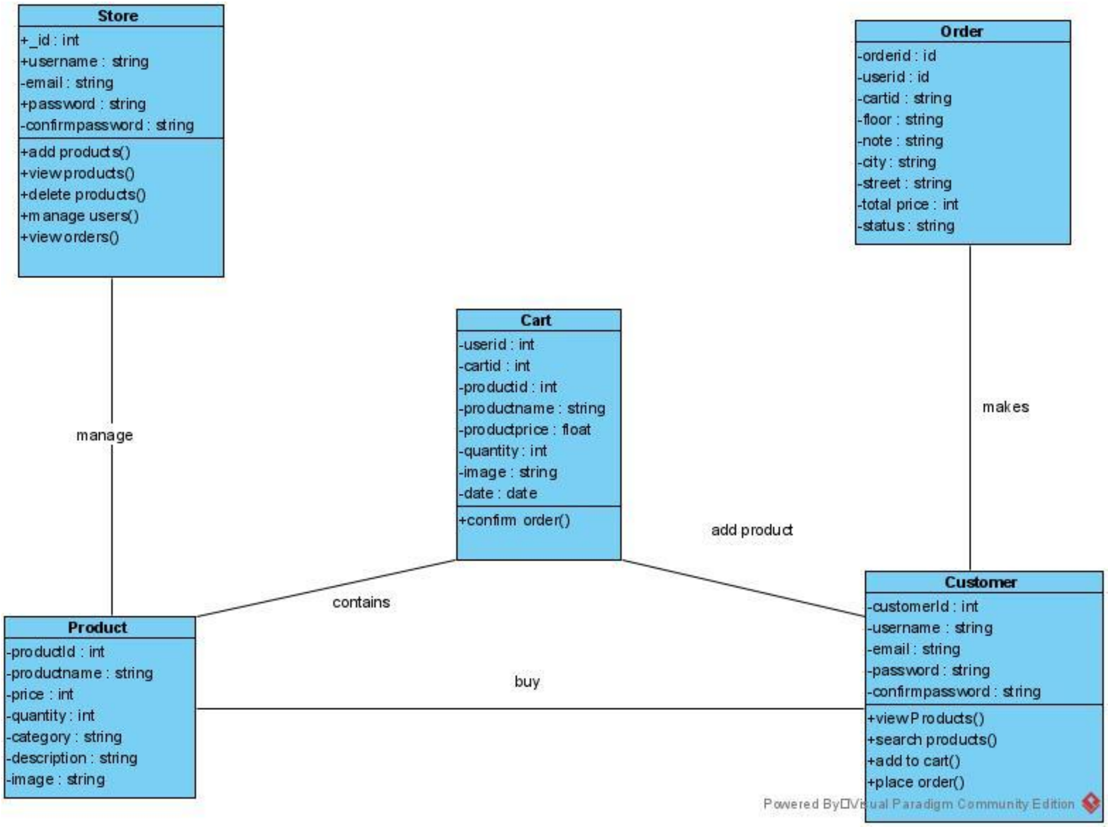
8.0 Appendices

The following list presents the diagrams that should be included at appropriate places

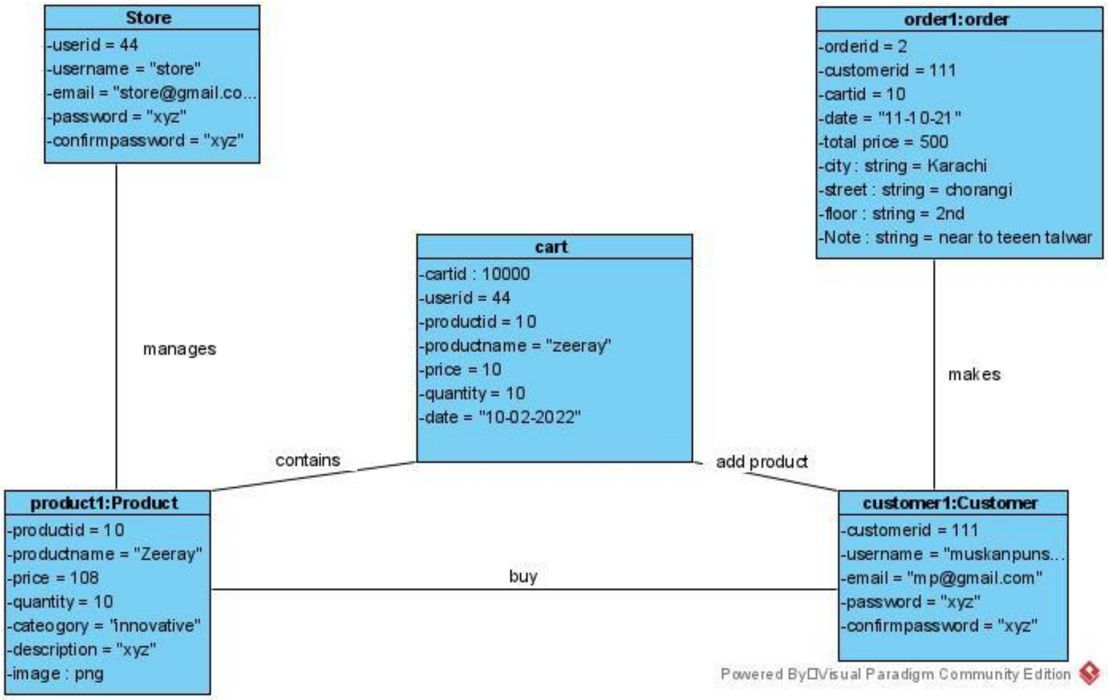
Class Diagram	Describes the structure of a system
Object Diagram	Expresses possible object combinations of a specific Class Diagram
State chart Diagram	Expresses possible states of a class (or a system)

Activity Diagram	Describes activities and actions taking place in a system
Sequence Diagram	Shows one or several sequences of messages sent among a set of objects
Collaboration Diagram	Describes a complete collaboration among a set of objects
Use-case Diagrams	Illustrates the relationships between use cases
Component Diagram	A special case of a Class Diagram used to describe components within a software system
Deployment Diagram	A special case of a Class Diagram used to describe hardware within the overall system architecture
System Block diagram	A diagram showing the major components of the system with its interconnections and external interfaces

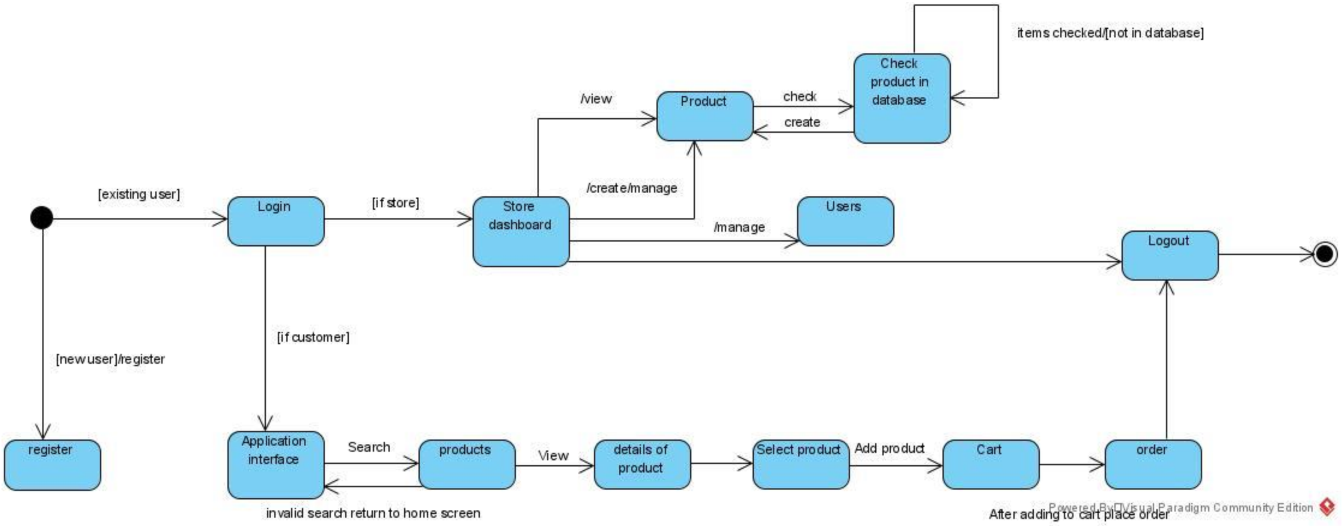
Class Diagram:



Object Diagram:

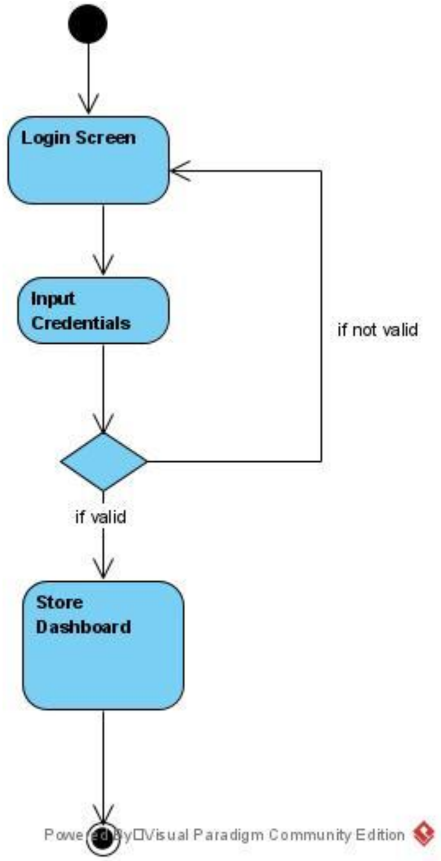


State Chart Diagram:

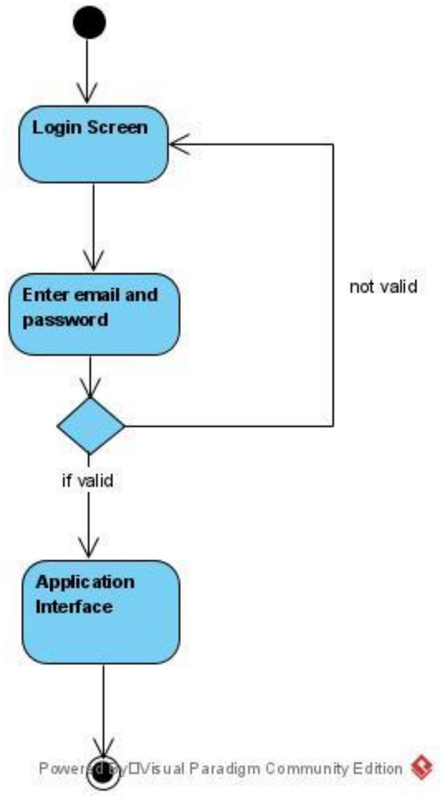


Activity Diagram:

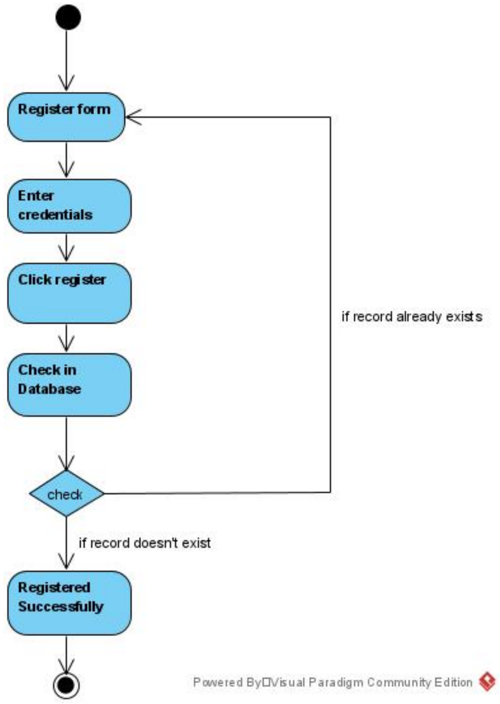
Store Login:


Login Screen
Input Credentials
if not valid
if valid
Store Dashboard
Powered By Visual Paradigm Community Edition

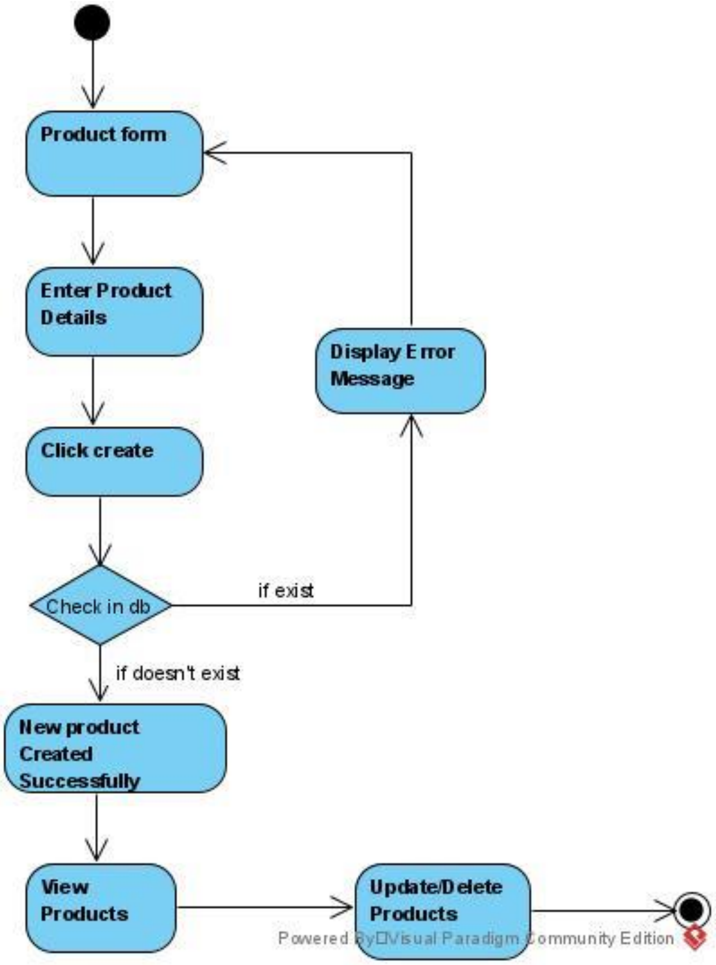
User Login:



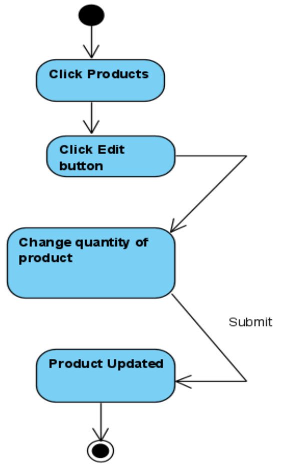
Register:



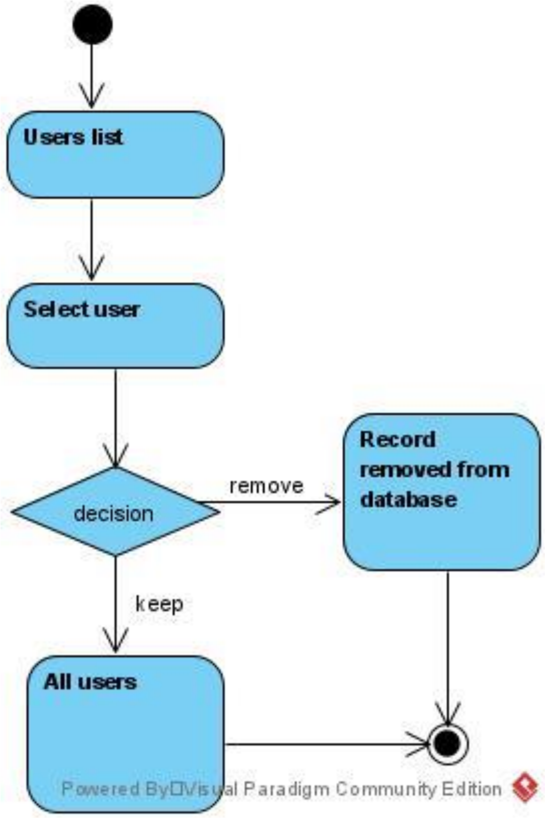
Create Product:



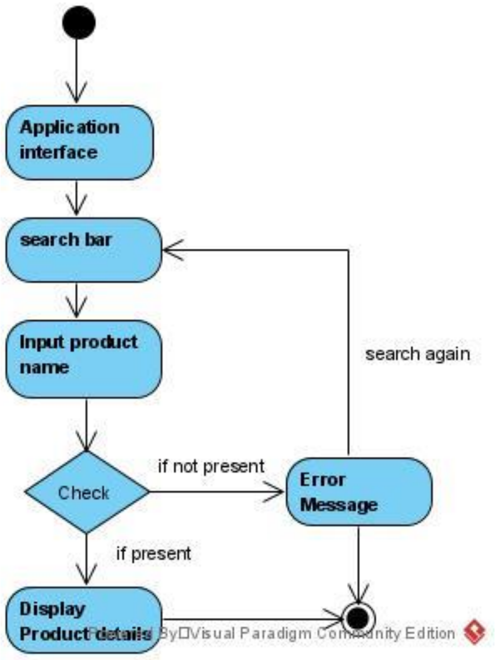
Manage Inventory:



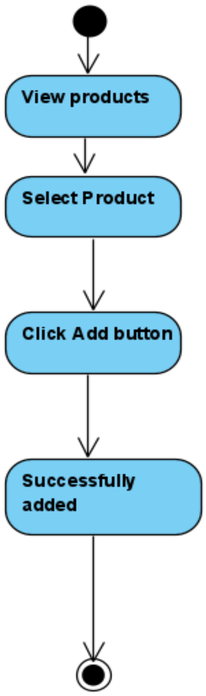
Manage Users:


Users list
Select user
decision
remove
Record removed from database
keep
All users
Powered By Visual Paradigm Community Edition

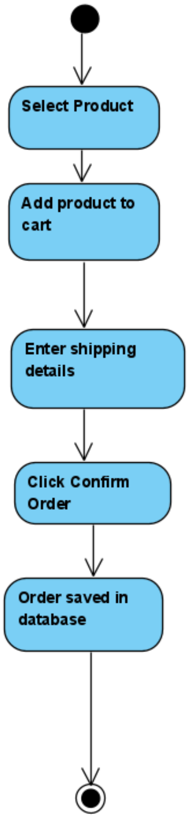
Search Products:



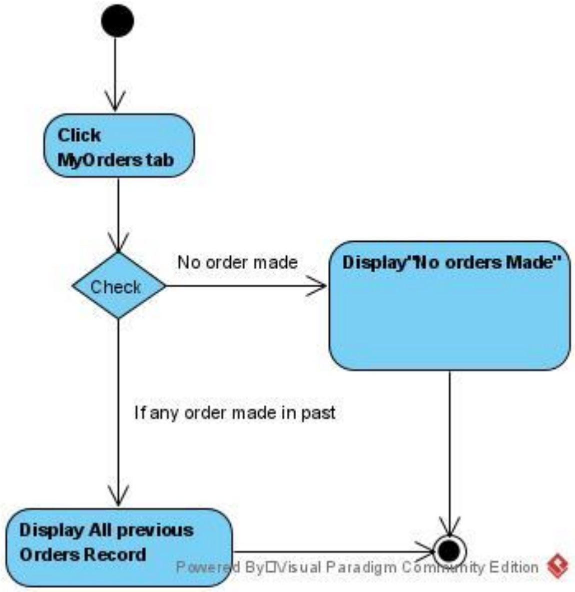
Add to Cart



Confirm Order:

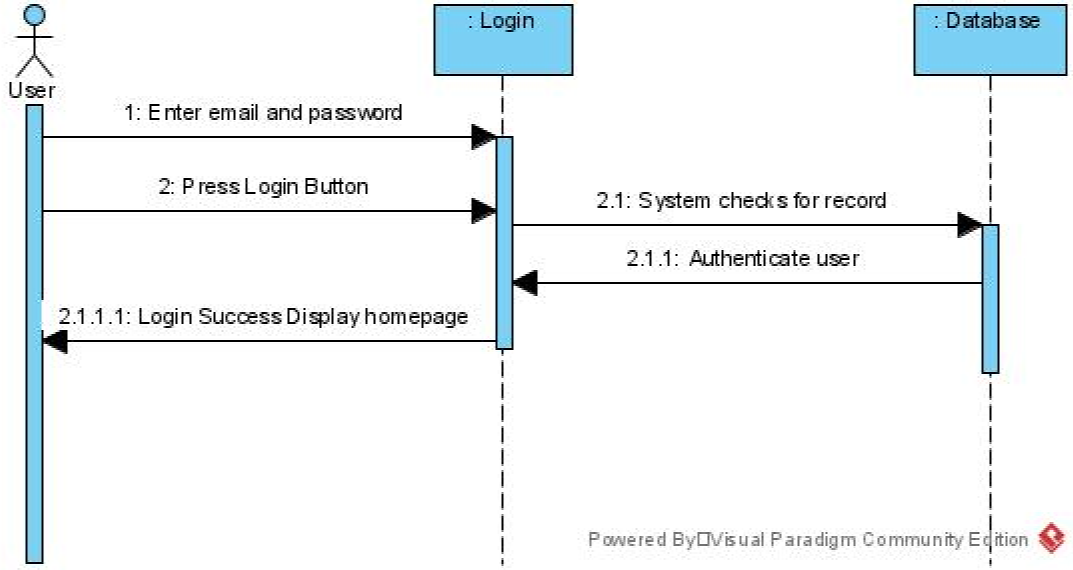


Display Previous Orders:

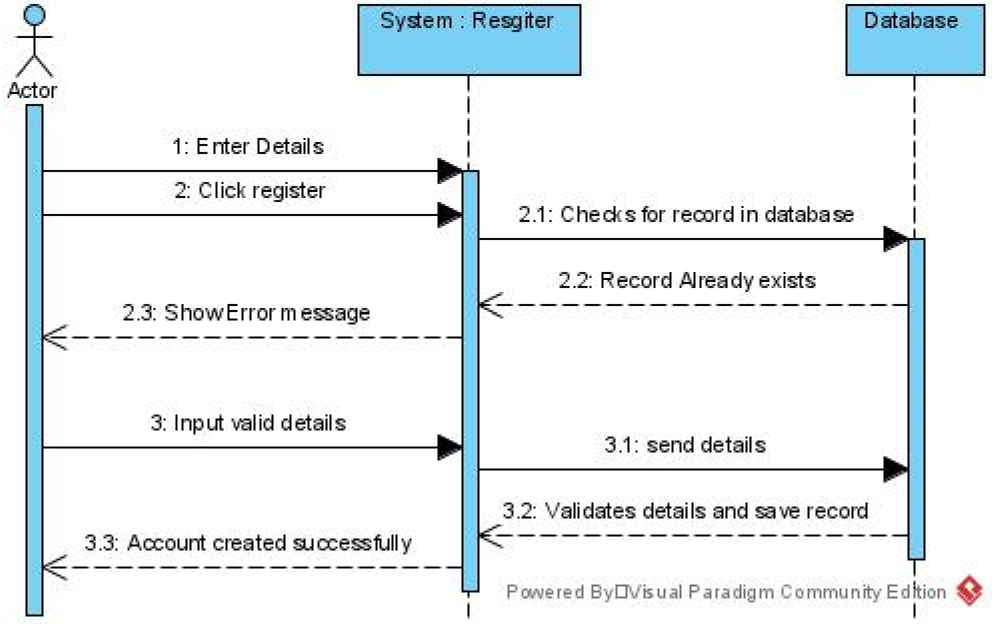

Click MyOrders tab
Check
No order made
Display"No orders Made"
If any order made in past
Display All previous Orders Record
Powered By Visual Paradigm Community Edition

Sequence Diagrams:

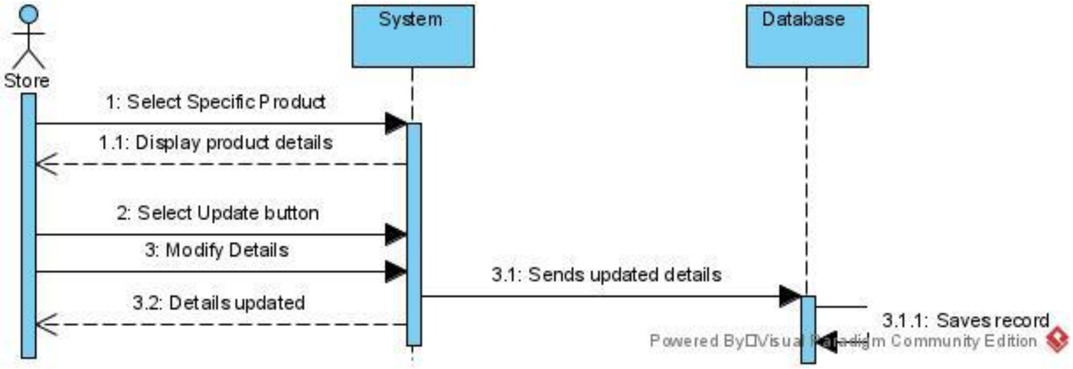
Login:


User
: Login
: Database
1: Enter email and password
2: Press Login Button
2.1: System checks for record
2.1.1: Authenticate user
2.1.1.1: Login Success Display homepage
Powered By Visual Paradigm Community Edition

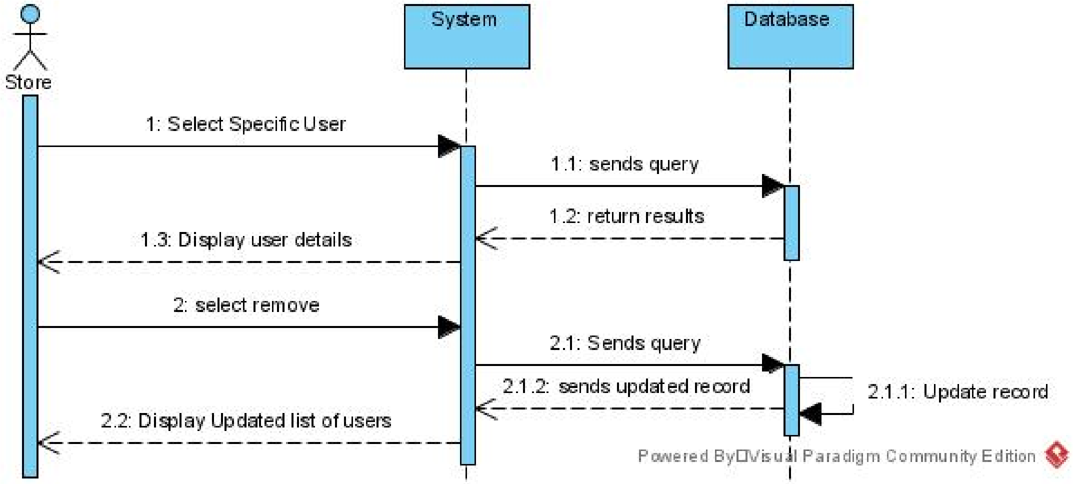
Register:


Actor
System : Resgiter
Database
1: Enter Details
2: Click register
2.1: Checks for record in database
2.2: Record Already exists
2.3: Show Error message
3: Input valid details
3.1: send details
3.2: Validates details and save record
3.3: Account created successfully
Powered By Visual Paradigm Community Edition

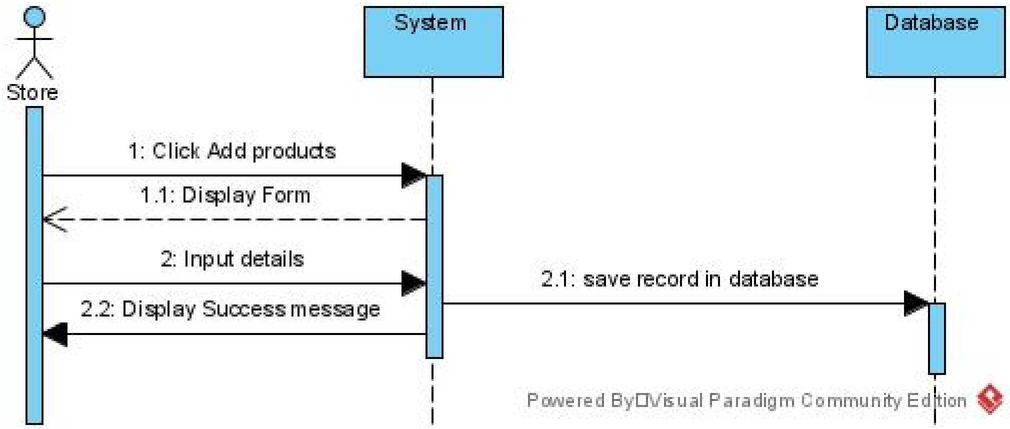
Manage Products:


Store
System
Database
1: Select Specific Product
1.1: Display product details
2: Select Update button
3: Modify Details
3.1: Sends updated details
3.2: Details updated
3.1.1: Saves record
Powered By Visual Paradigm Community Edition

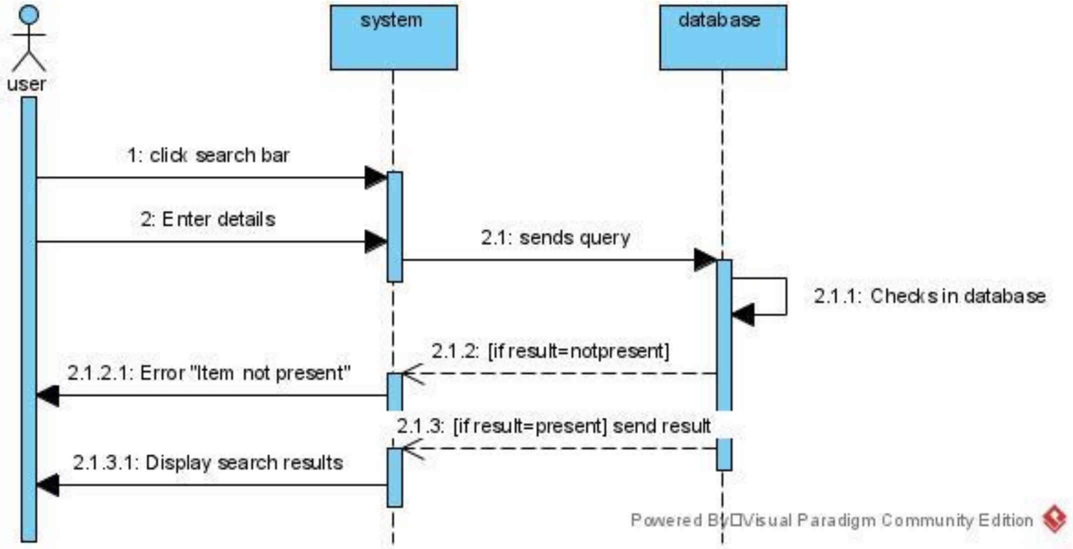
Manage Users:


Store
System
Database
1: Select Specific User
1.1: sends query
1.2: return results
1.3: Display user details
2: select remove
2.1: Sends query
2.1.2: sends updated record
2.1.1: Update record
2.2: Display Updated list of users
Powered By Visual Paradigm Community Edition

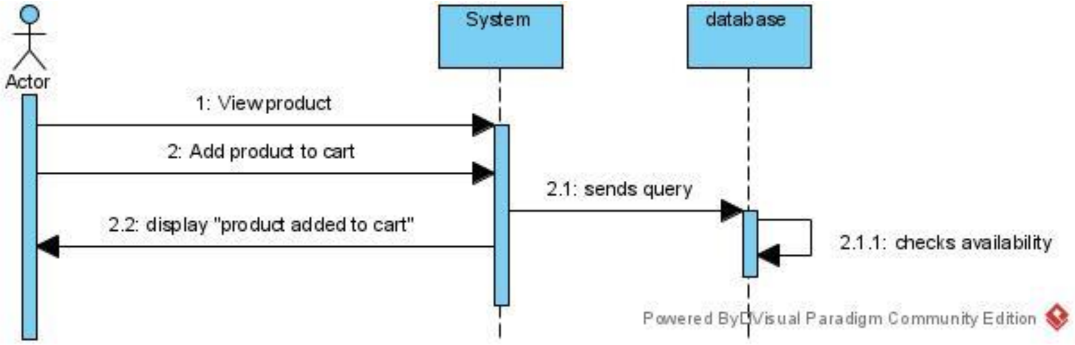
Create Product:


Store
System
Database
1: Click Add products
1.1: Display Form
2: Input details
2.1: save record in database
2.2: Display Success message
Powered By□Visual Paradigm Community Edition

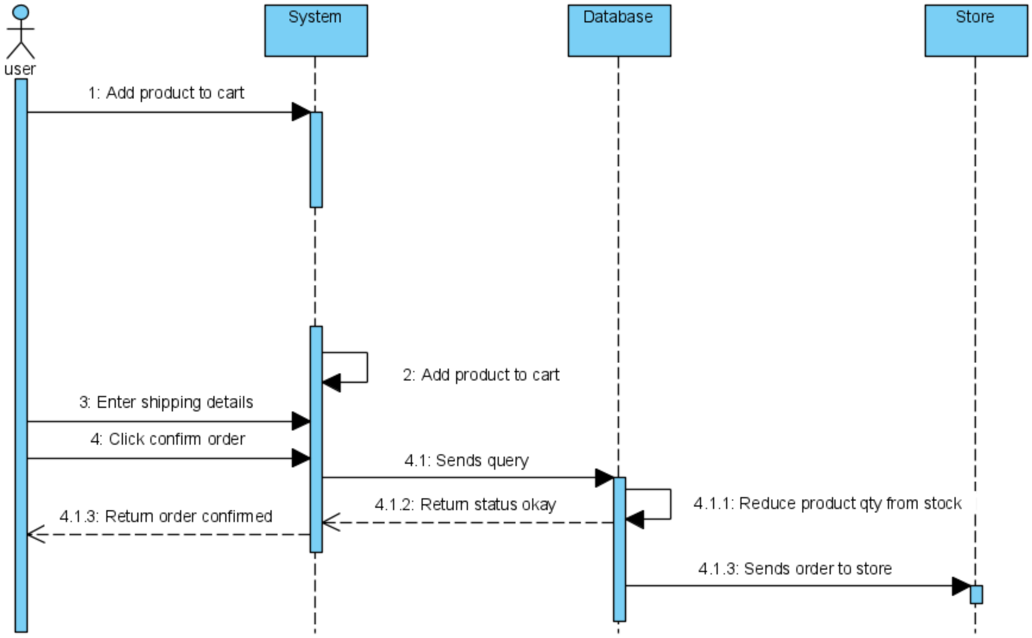
Search Products


user
system
database
1: click search bar
2: Enter details
2.1: sends query
2.1.1: Checks in database
2.1.2: [if result=notpresent]
2.1.2.1: Error "Item not present"
2.1.3: [if result=present] send result
2.1.3.1: Display search results
Powered By□Visual Paradigm Community Edition

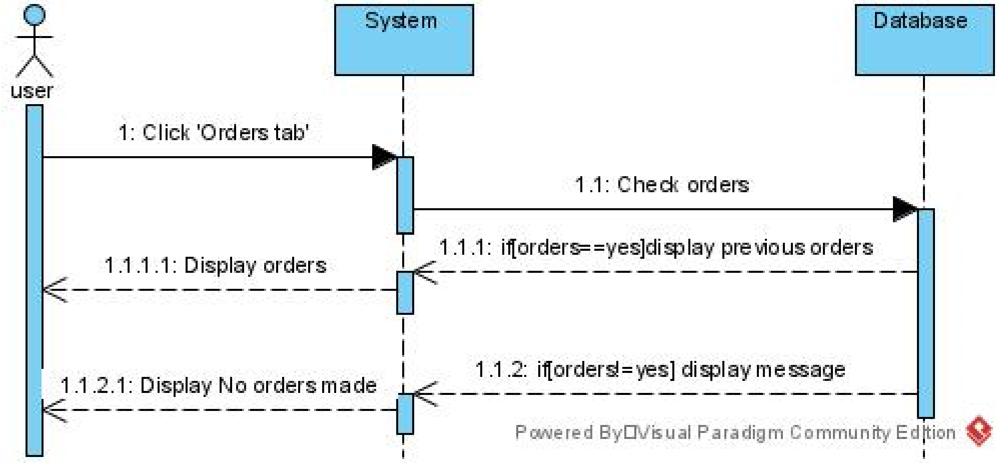
Add products to cart


Actor
System
database
1: View product
2: Add product to cart
2.1: sends query
2.1.1: checks availability
2.2: display "product added to cart"
Powered By Visual Paradigm Community Edition

Confirm Order:

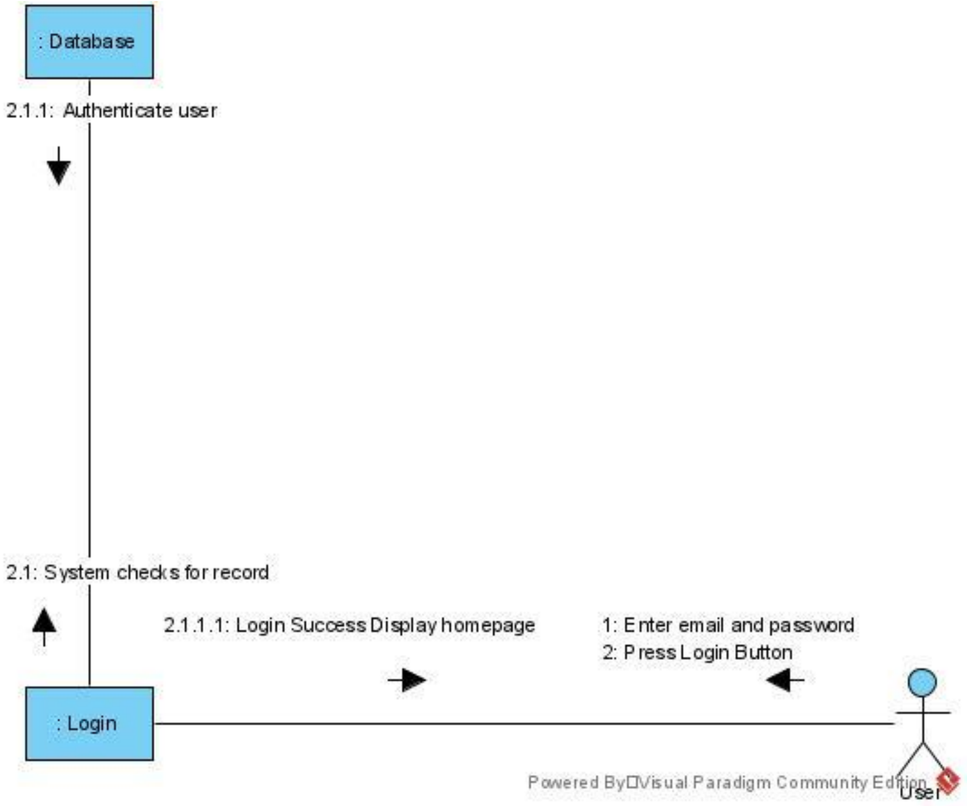

user
System
Database
Store
1: Add product to cart
2: Add product to cart
3: Enter shipping details
4: Click confirm order
4.1: Sends query
4.1.1: Reduce product qty from stock
4.1.2: Return status okay
4.1.3: Return order confirmed
4.1.3: Sends order to store

Show Previous Orders:

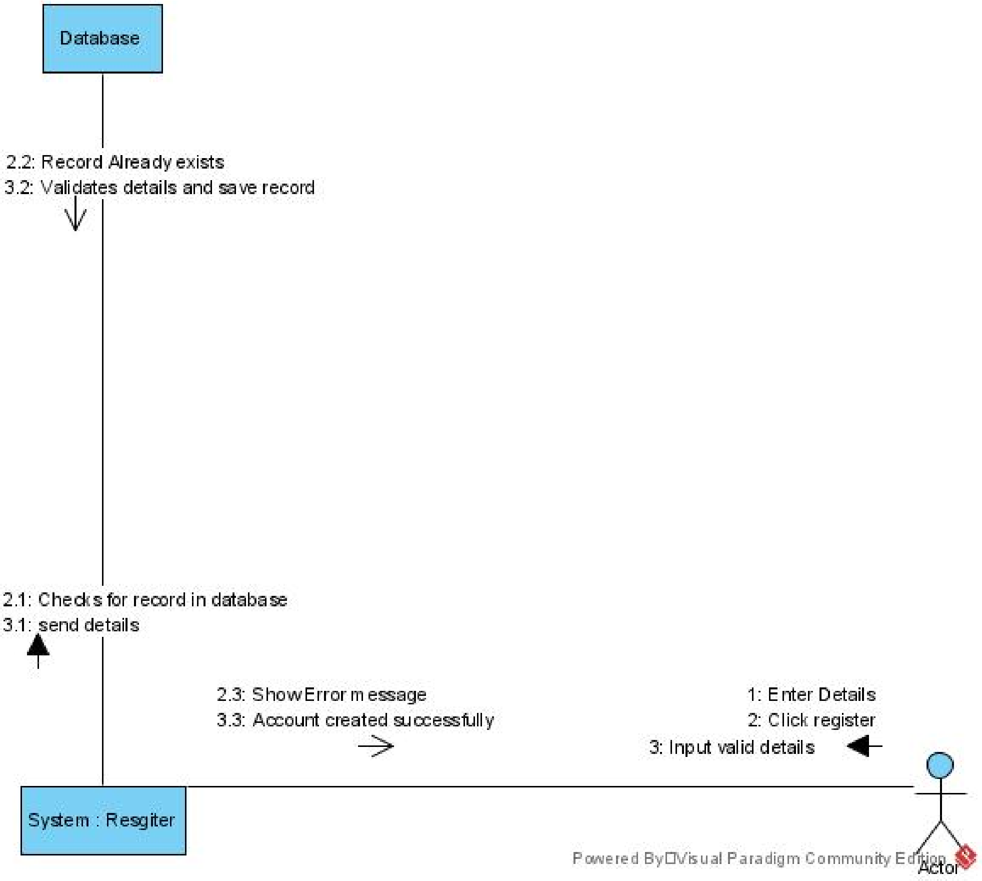

System
Database
user
1: Click 'Orders tab'
1.1: Check orders
1.1.1: if[orders==yes]display previous orders
1.1.1.1: Display orders
1.1.2: if[orders!=yes] display message
1.1.2.1: Display No orders made
Powered By□Visual Paradigm Community Edition

Collaboration Diagram:

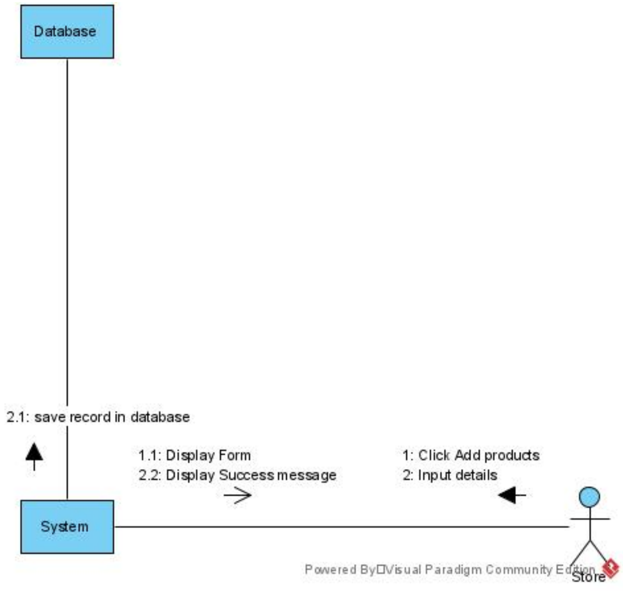
Login:


: Database
2.1.1: Authenticate user
2.1: System checks for record
2.1.1.1: Login Success Display homepage
1: Enter email and password
2: Press Login Button
: Login
User
Powered By□Visual Paradigm Community Edition

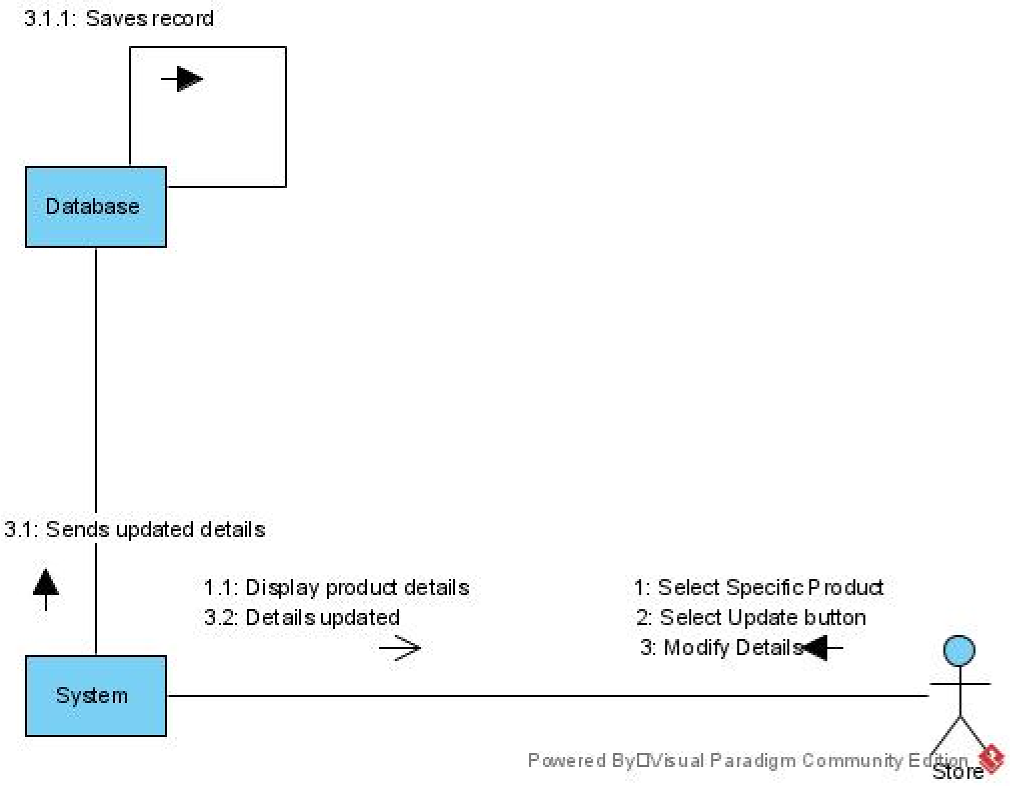
Register:


Database
2.2: Record Already exists
3.2: Validates details and save record
2.1: Checks for record in database
3.1: send details
2.3: Show Error message
3.3: Account created successfully
1: Enter Details
2: Click register
3: Input valid details
System : Resgiter
Actor
Powered By Visual Paradigm Community Edition

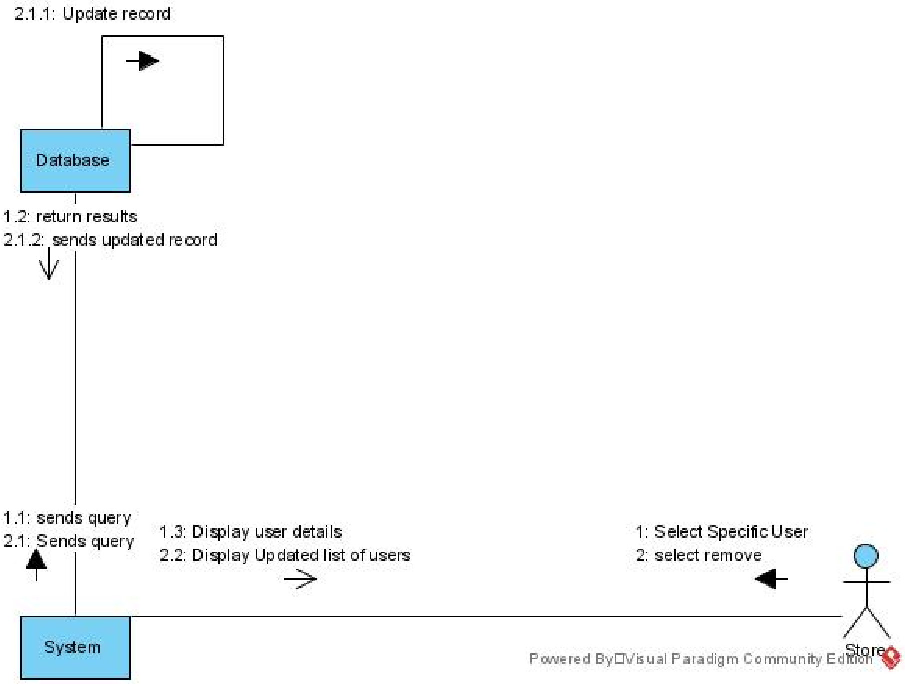
Create product:


Database
2.1: save record in database
1.1: Display Form
2.2: Display Success message
1: Click Add products
2: Input details
System
Store
Powered By Visual Paradigm Community Edition

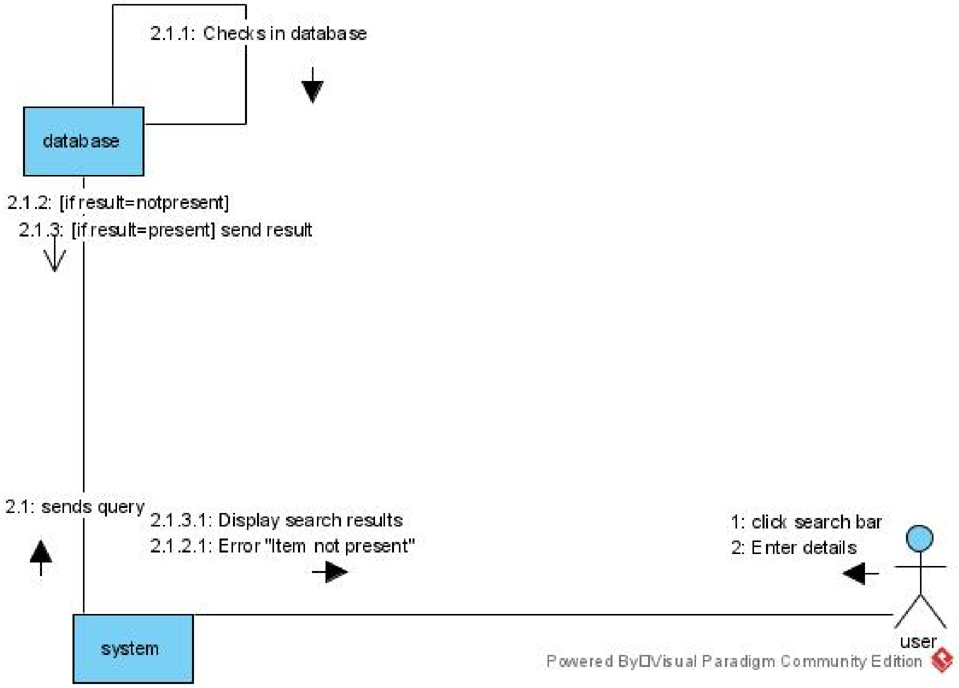
Manage Inventory/Product:


3.1.1: Saves record
Database
3.1: Sends updated details
1.1: Display product details
3.2: Details updated
1: Select Specific Product
2: Select Update button
3: Modify Details
System
Store
Powered By Visual Paradigm Community Edition

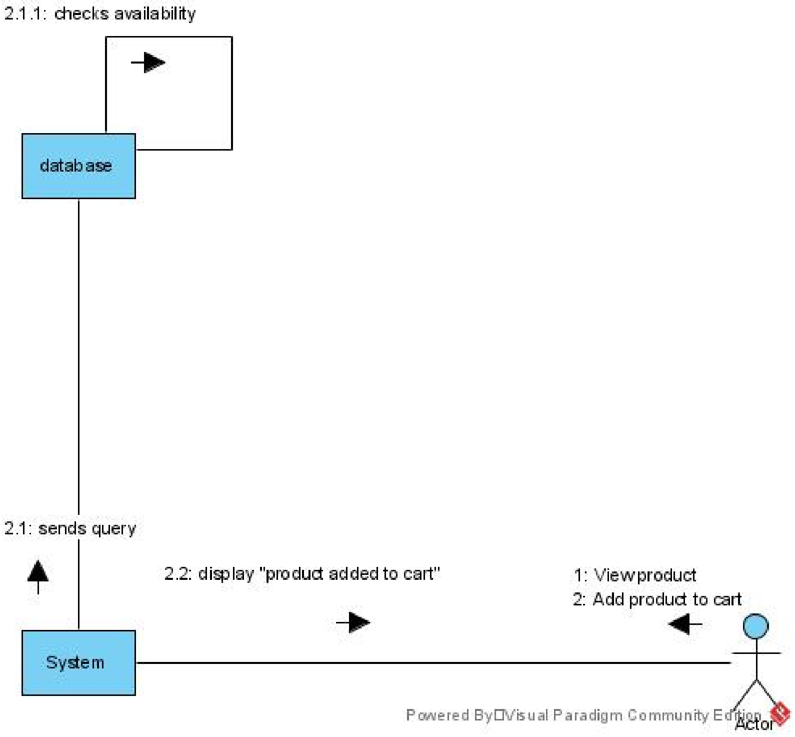
Manage Users:


2.1.1: Update record
Database
1.2: return results
2.1.2: sends updated record
1.1: sends query
2.1: Sends query
1.3: Display user details
2.2: Display Updated list of users
1: Select Specific User
2: select remove
System
Store
Powered By Visual Paradigm Community Edition

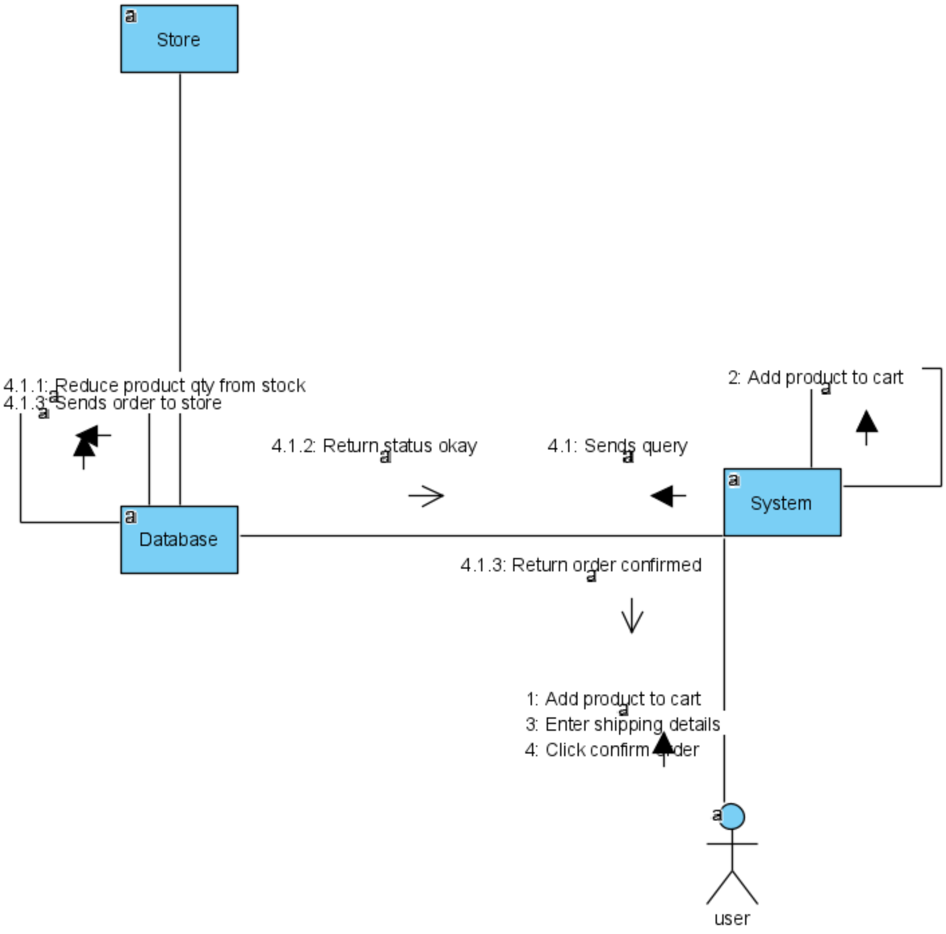
Search Products:


2.1.1: Checks in database
database
2.1.2: [if result=notpresent]
2.1.3: [if result=present] send result
2.1: sends query
2.1.3.1: Display search results
2.1.2.1: Error "Item not present"
1: click search bar
2: Enter details
system
user
Powered By□Visual Paradigm Community Edition

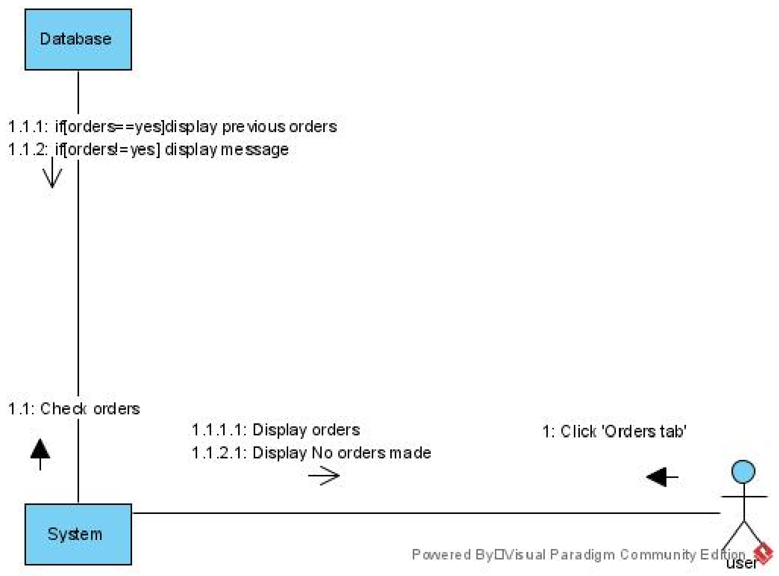
Add to Cart:


2.1.1: checks availability
database
2.1: sends query
2.2: display "product added to cart"
1: View product
2: Add product to cart
System
Actor
Powered By□Visual Paradigm Community Edition

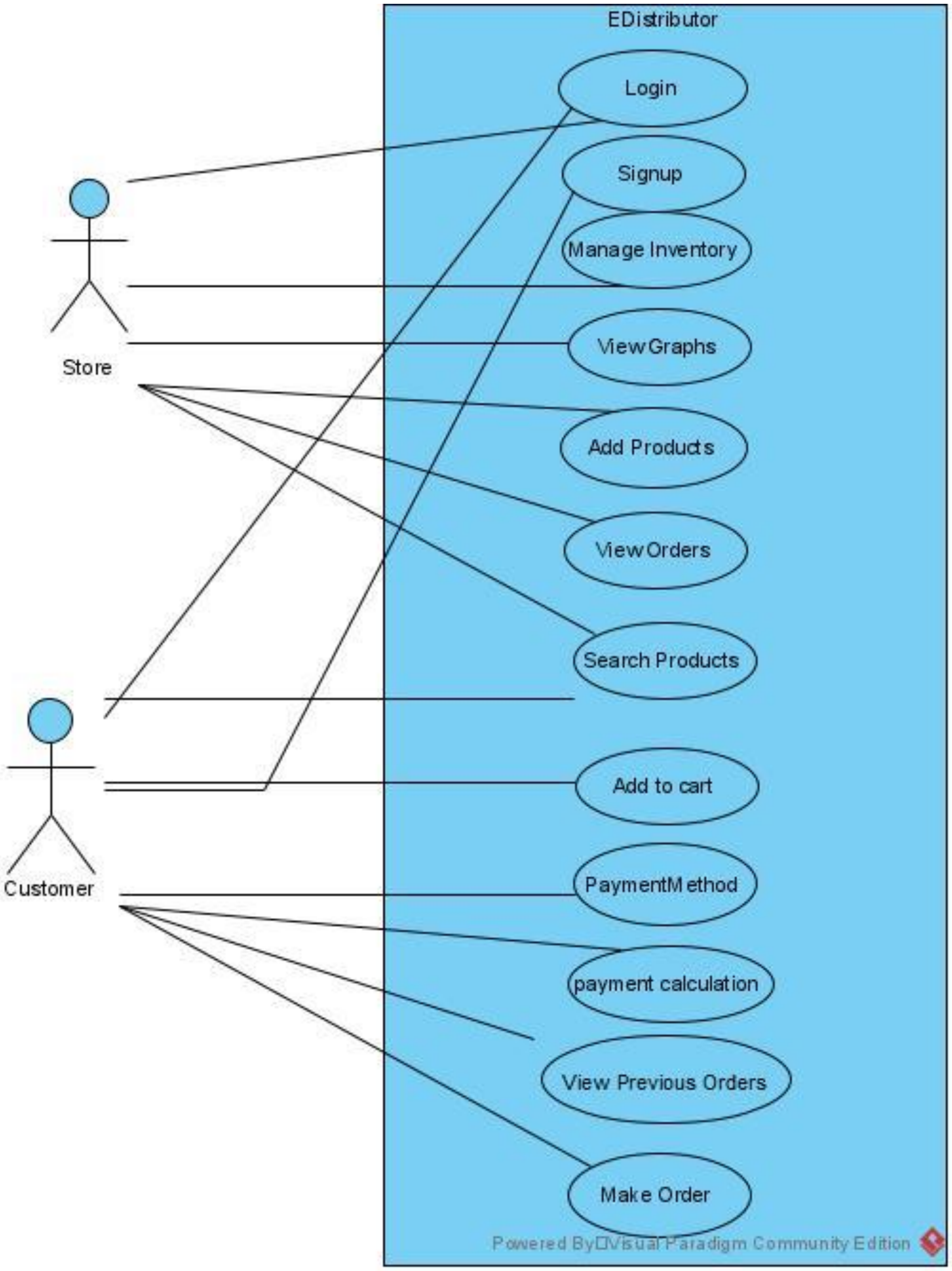
Confirm Order:


Store
4.1.1: Reduce product qty from stock
4.1.3: Sends order to store
4.1.2: Return status okay
4.1: Sends query
2: Add product to cart
System
Database
4.1.3: Return order confirmed
1: Add product to cart
3: Enter shipping details
4: Click confirm order
user

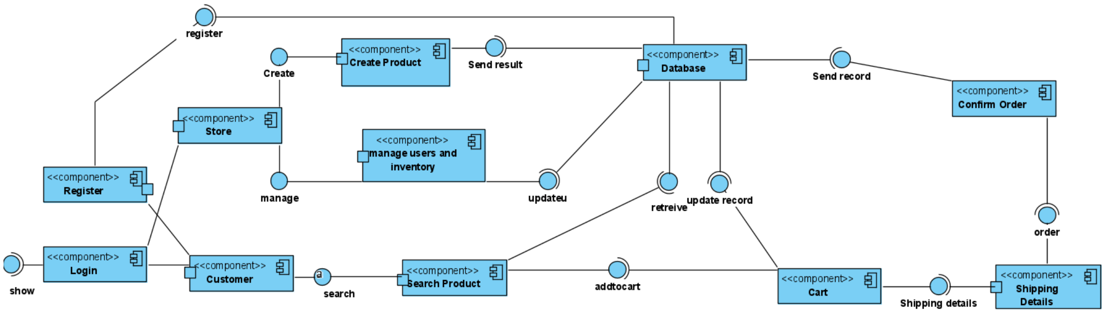
View previous orders:


Database
1.1.1: if[orders==yes]display previous orders
1.1.2: if[orders!=yes] display message
1.1: Check orders
1.1.1.1: Display orders
1.1.2.1: Display No orders made
1: Click 'Orders tab'
System
user
Powered By Visual Paradigm Community Edition

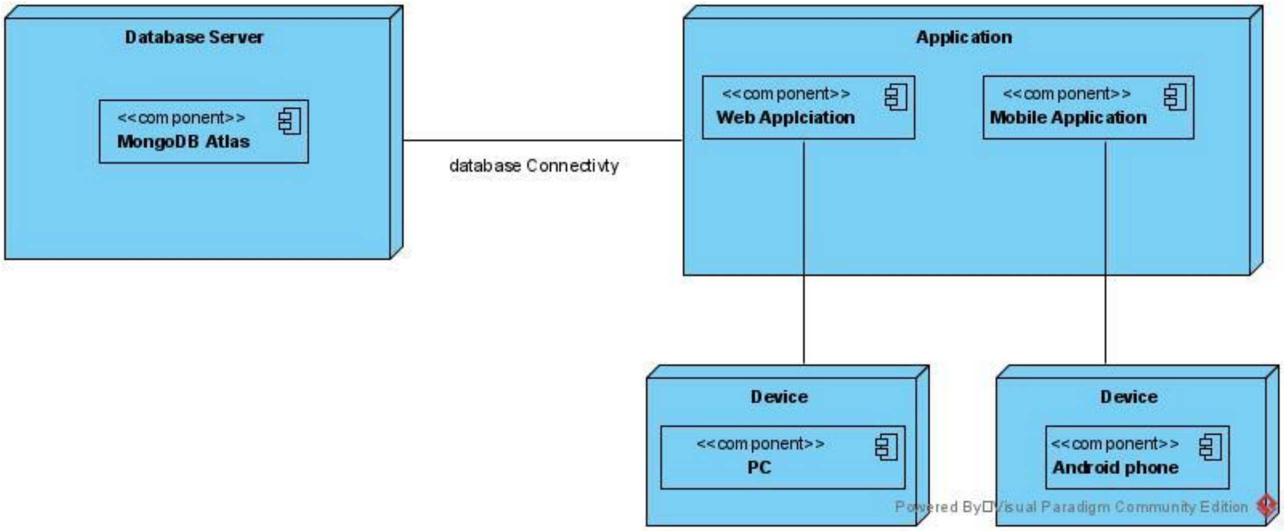
Use Case Diagram:



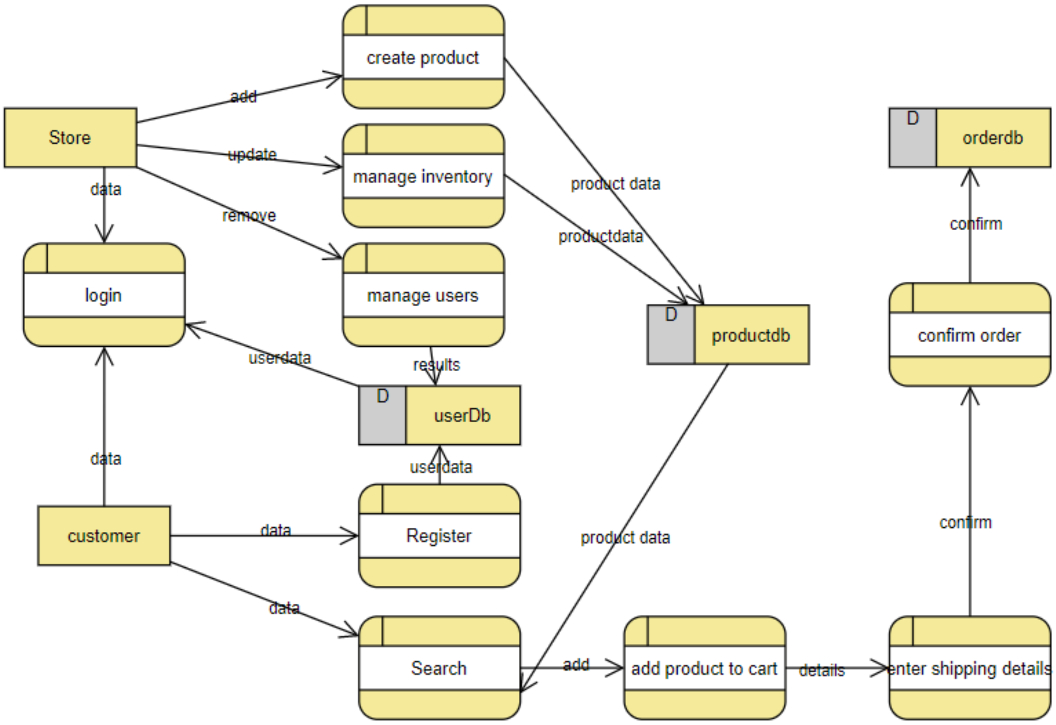
Component Diagram:



Deployment Diagram:



System Block Diagram:



IV. Test Case Documentation

Test Case # 1:

Test case ID	**Test case name**	**Test case summary**	**Test case steps**	**Expected result**	**Actual result**	**Pass/Fail**
1	Login	This test case is used to test the login functionality	i. Enter valid email address and password.	Respective home screen will be shown to user	Respective home screen will be shown to user	Pass
			ii. Enter invalid email and password	Alert (‘Login failed’)	Alert (‘Login failed’)	Pass
			iii. Enter invalid email syntax	Error “enter valid syntax”	Error “enter valid syntax”	Pass
			iv. Enter invalid password	Error “The password you have entered is incorrect”	Error “The password you have entered is incorrect”	Fail
			v. Leave fields empty	Error “This field cannot be empty”	Error “This field cannot be empty”	Pass

Test Case # 2:

Test case ID	Test case name	Test case summary	Test case steps	Expected result	Actual result	Pass/Fail
2	Register	This test case is used to test the registration functionality	i. Fill all input fields	Message ”user created successfully”	Message ”user created successfully”	Pass
			ii. Leave fields empty	Error ”Fields cannot be empty”	Error ”Fields cannot be empty”	Pass
			iii. Enter existing email	Error “User with this email already exists”	Error “User with this email already exists”	Pass
			iv. Enter existing username	Error “This username already exists”	Error “This username already exists”	Fail
			v. Enter invalid syntax of password	Error “Password must be between 4 and 8 digits long and include at least one number”	Error “Password must be between 4 and 8 digits long and include at least one number”	Pass

Test Case # 3:

Test case ID	Test case name	Test case summary	Test case steps	Expected result	Actual result	Pass/Fail
3	Create Product	This test case will test the creation of product	i. Fill all input fields	Message ”Product created successfully”	Message ”Product created successfully”	Pass
			ii. Leave fields empty	Error ”Fields cannot be empty”	Error ”Fields cannot be empty”	Pass
			iii. Enter product with existing name	Error “Product with this name already exists”	Error “Product with this name already exists”	Pass
			iv. Enter Invalid Category	Error “Invalid category”	Error “Invalid category”	Fail

Test Case # 4:

Test case ID	Test case name	Test case summary	Test case steps	Expected result	Actual result	Pass/Fail
4	List Product On Store	This test case will test functionality of list product	i. Display all products	Show all products	Show all products	Pass
			ii. Display product category wise.	Show all products category wise	Show all products category wise	Fail

Test Case # 5:

Test case ID	Test case name	Test case summary	Test case steps	Expected result	Actual result	Pass/Fail
5	List products on application	This test case will test the functionality of list product	i. Click category “Innovative”	Show product of “Innovative”	Show product of “Innovative”	Pass
			ii. Click category “Bakeline”	Show product of “Innovative”	Show product of “Innovative”	Pass

Test Case # 6:

Test case ID	Test case name	Test case summary	Test case steps	Expected result	Actual result	Pass/Fail
6	Search Product	This test case will test functionality of search button	i. Search a product that already exists “Case Sensitive”	Display “Searched product”	Display “Searched product”	Pass
			ii. Search product category wise	Display product according to category	Display product according to category	Fail
			iii. Search product “Case insensitive”	Display “Searched product”	Display “Searched product”	Fail
			iv. Search product that do not exist on database	Error “No results found”	Error “No results found”	Pass

Test Case # 7:

Test case ID	Test case name	Test case summary	Test case steps	Expected result	Actual result	Pass/Fail
7	Add product to cart	This test case will test functionality of adding products to cart	Search a product	Display “Searched product”	Display “Searched product”	Pass
			ii.Select the product	Display product	Display product	pass
			iii. Add product to cart	Product successfully added	Product successfully added	pass
			iv. Modify Cart	Modification	Cannot Modify	Pass

Test Case # 8:

Test case ID	Test case name	Test case summary	Test case steps	Expected result	Actual result	Pass/Fail
8	Confirm order	This test case will test functionality of confirming order	I. Select the product	Display product”	Display product”	Pass
			ii.Add to cart	Added to cart	Added to cart	pass
			iii. Enter shipping details	Input details	Input details	pass
			iv. Show order summary	Order summary with total	Order summary with total	Pass

			V . Confirm Order	Order confirm success message	Order confirm success message	Pass

Test Case # 9:

Test case ID	Test case name	Test case summary	Test case steps	Expected result	Actual result	Pass/Fail
9	Update quantity of product	This test case will test functionality of update quantity	I. Select the product	Display product details	Display product details	Pass
			Ii Increase quantity	Success and updated in database	Success and updated in database	pass
			iii. decrease quantity	Decrease quantity and update in database	Decrease quantity and update in database	pass
			iv. Show updated list	Updated List	Updated list	Pass

Test Case # 10:

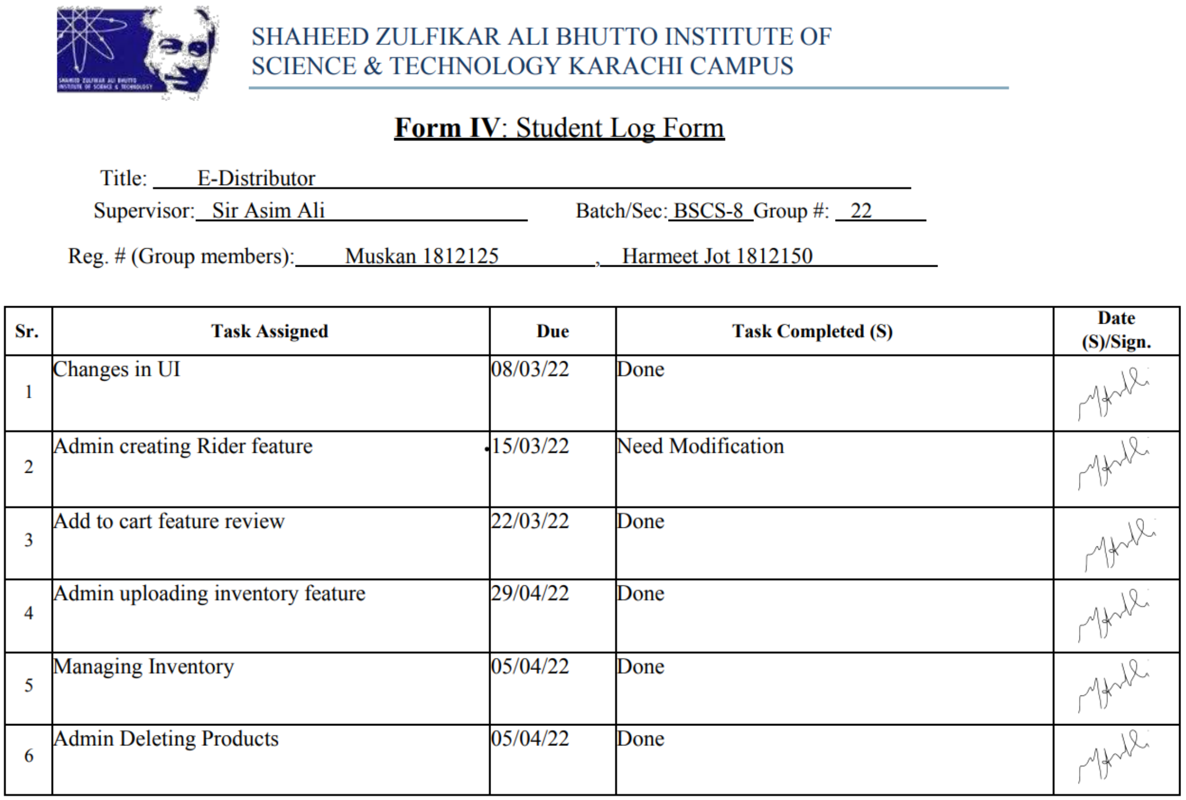
Test case ID	Test case name	Test case summary	Test case steps	Expected result	Actual result	Pass/Fail
10	Manage Users	This test case will test functionality of manage users	I. Select the user	Display details	Display details	Pass
			Ii Click delete button	Success and updated in database	Success and updated in database	pass
			iii. Show updated list	Updated List	Updated list	Pass

Test Case # 11:

Test case ID	**Test case name**	**Test case summary**	**Test case steps**	**Expected result**	**Actual result**	**Pass/Fail**
11	Graphs	This test case will test graphs	i.View graphs	Display graphs	Display graphs	Pass
			ii. Modify database	Updated graphs data	Updated graphs data	pass

i.Meeting Log Sheet

Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST)Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST)Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST)Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST)Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST)



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF
SCIENCE & TECHNOLOGY KARACHI CAMPUS

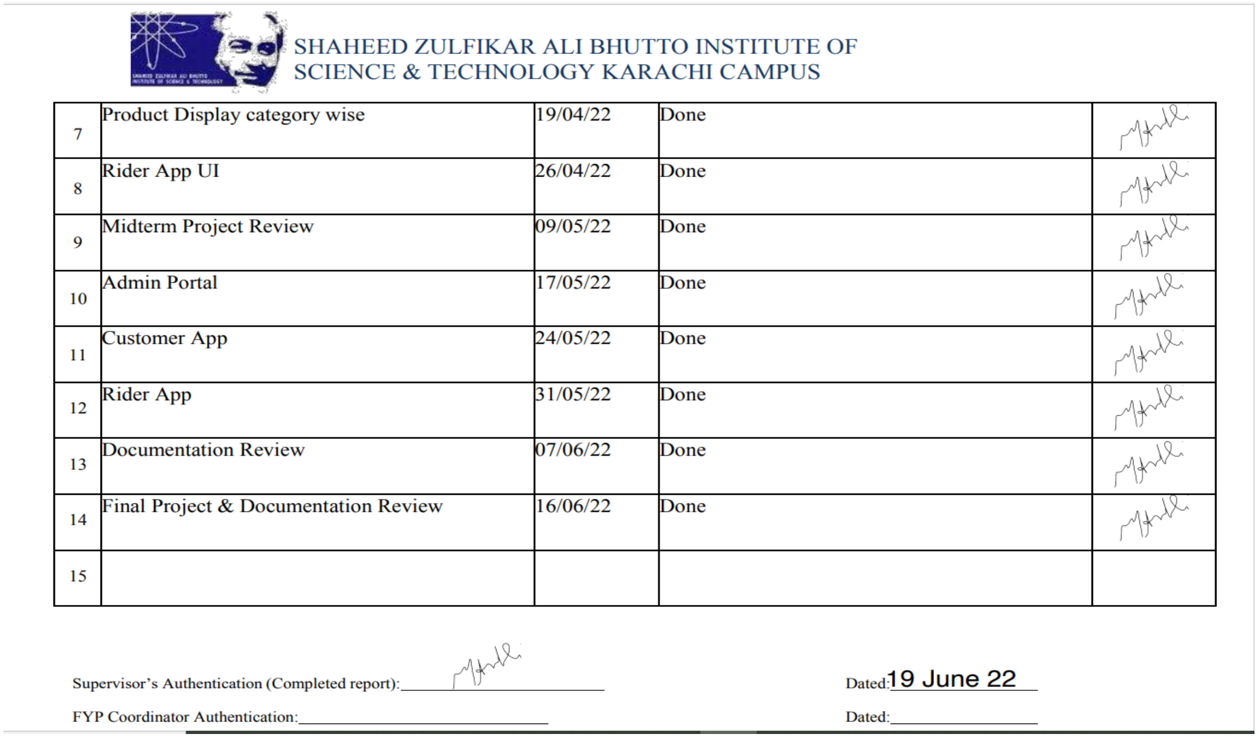
Form IV: Student Log Form

Title: E-Distributor

Supervisor: Sir Asim Ali Batch/Sec: BSCS-8 Group #: 22

Reg. # (Group members): Muskan 1812125 , Harmeet Jot 1812150

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
1	Changes in UI	08/03/22	Done	
2	Admin creating Rider feature	15/03/22	Need Modification	
3	Add to cart feature review	22/03/22	Done	
4	Admin uploading inventory feature	29/04/22	Done	
5	Managing Inventory	05/04/22	Done	
6	Admin Deleting Products	05/04/22	Done	



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF
SCIENCE & TECHNOLOGY KARACHI CAMPUS

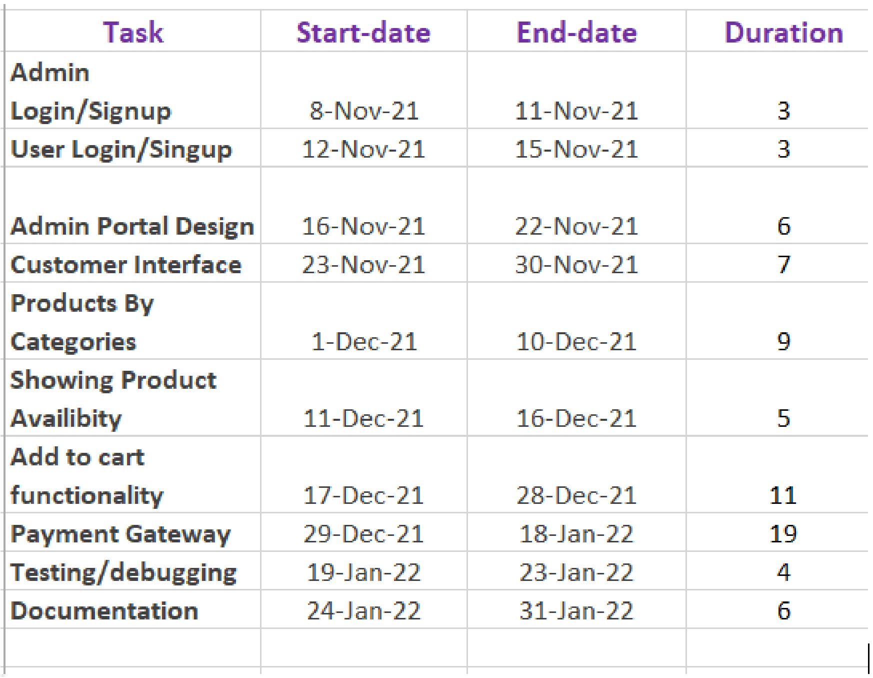
7	Product Display category wise	19/04/22	Done	
8	Rider App UI	26/04/22	Done	
9	Midterm Project Review	09/05/22	Done	
10	Admin Portal	17/05/22	Done	
11	Customer App	24/05/22	Done	
12	Rider App	31/05/22	Done	
13	Documentation Review	07/06/22	Done	
14	Final Project & Documentation Review	16/06/22	Done	
15				

Supervisor's Authentication (Completed report):____________ Dated: 19 June 22

FYP Coordinator Authentication:____________ Dated:____________

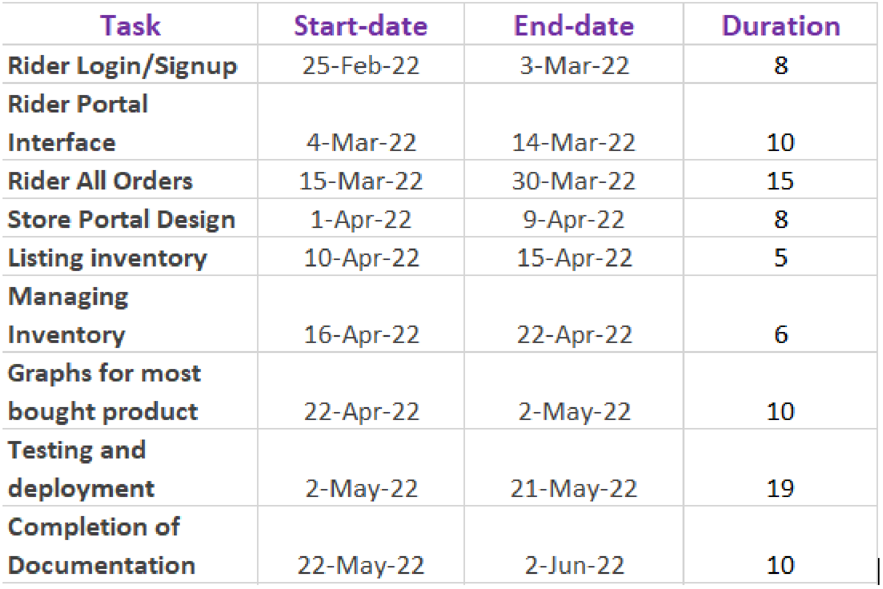
ii.Iteration Plan:

FYP-I



Task	Start-date	End-date	Duration
Admin Login/Signup	8-Nov-21	11-Nov-21	3
User Login/Singup	12-Nov-21	15-Nov-21	3
Admin Portal Design	16-Nov-21	22-Nov-21	6
Customer Interface	23-Nov-21	30-Nov-21	7
Products By Categories	1-Dec-21	10-Dec-21	9
Showing Product Availibity	11-Dec-21	16-Dec-21	5
Add to cart functionality	17-Dec-21	28-Dec-21	11
Payment Gateway	29-Dec-21	18-Jan-22	19
Testing/debugging	19-Jan-22	23-Jan-22	4
Documentation	24-Jan-22	31-Jan-22	6

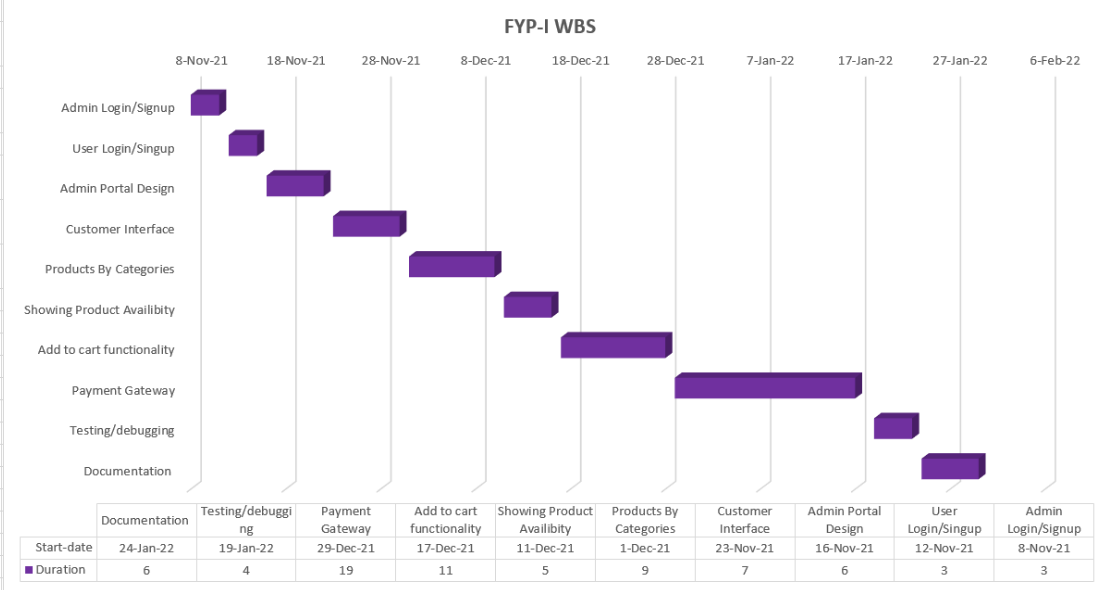
FYP-II



Task	Start-date	End-date	Duration
Rider Login/Signup	25-Feb-22	3-Mar-22	8
Rider Portal Interface	4-Mar-22	14-Mar-22	10
Rider All Orders	15-Mar-22	30-Mar-22	15
Store Portal Design	1-Apr-22	9-Apr-22	8
Listing inventory	10-Apr-22	15-Apr-22	5
Managing Inventory	16-Apr-22	22-Apr-22	6
Graphs for most bought product	22-Apr-22	2-May-22	10
Testing and deployment	2-May-22	21-May-22	19
Completion of Documentation	22-May-22	2-Jun-22	10

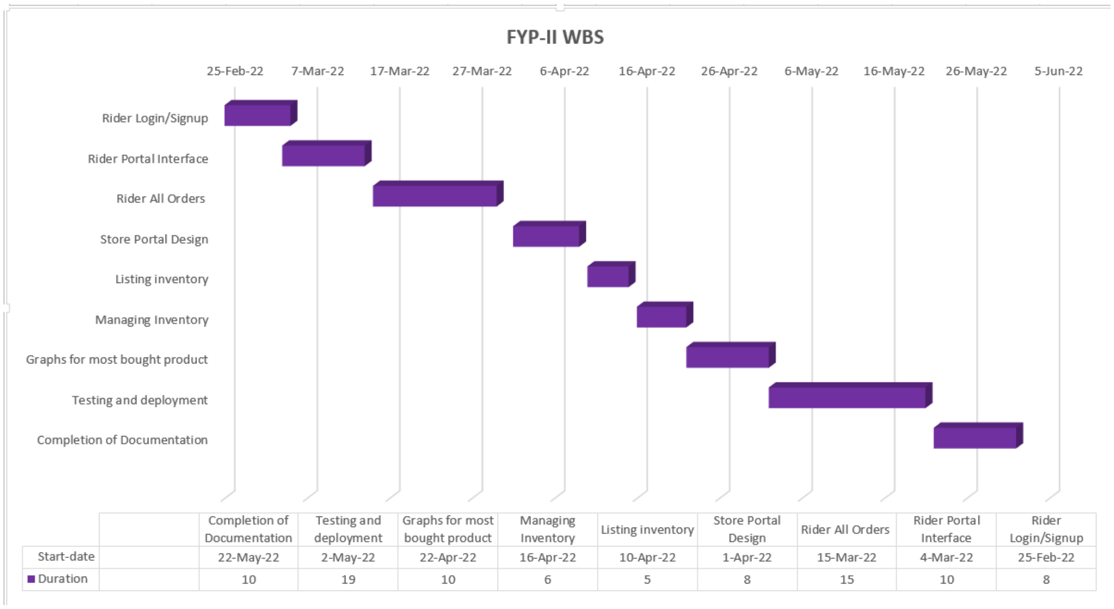
iii.Gantt Chart

FYP-I



	Documentation	Testing/debugging	Payment Gateway	Add to cart functionality	Showing Product Availibity	Products By Categories	Customer Interface	Admin Portal Design	User Login/Singup	Admin Login/Signup
Start-date	24-Jan-22	19-Jan-22	29-Dec-21	17-Dec-21	11-Dec-21	1-Dec-21	23-Nov-21	16-Nov-21	12-Nov-21	8-Nov-21
■ Duration	6	4	19	11	5	9	7	6	3	3

FYP-II



	Completion of Documentation	Testing and deployment	Graphs for most bought product	Managing Inventory	Listing inventory	Store Portal Design	Rider All Orders	Rider Portal Interface	Rider Login/Signup
Start-date	22-May-22	2-May-22	22-Apr-22	16-Apr-22	10-Apr-22	1-Apr-22	15-Mar-22	4-Mar-22	25-Feb-22
■ Duration	10	19	10	6	5	8	15	10	8