Abstract

Go-Grocer is a modern web-based platform created to transform the traditional grocery shopping experience into a seamless, digital process. As lifestyles become increasingly fast-paced, consumers demand convenience, efficiency, and reliability — all of which Go-Grocer delivers. The platform offers a comprehensive catalog of grocery items, including fresh produce, dairy, bakery products, frozen foods, beverages, and household essentials. With an intuitive interface and robust search functionality, users can easily find what they need, add items to their cart, and place orders within minutes. The website is designed to be mobile-responsive, ensuring accessibility on all devices, while features such as account creation, wishlist management, and order history offer a personalized experience.

To further enhance convenience, Go-Grocer integrates real-time inventory management, allowing customers to see which products are in stock. The platform also supports multiple secure payment methods, including credit/debit cards, digital wallets, and cash-on-delivery, catering to diverse user preferences. Customers can choose from various delivery slots, and real-time order tracking keeps them informed from checkout to doorstep delivery. For added value, Go-Grocer includes a recommendation engine based on past purchases and browsing behavior, helping users discover new products while saving time. Additionally, the site promotes customer engagement through special offers, loyalty programs, and subscription services for frequently purchased items.

Behind the scenes, Go-Grocer is powered by a scalable backend that ensures smooth performance, even during high-traffic periods. For vendors and administrators, the site includes a management dashboard for inventory, order processing, analytics, and customer support. Environmental consciousness is also a part of Go-Grocer’s mission, with eco-friendly packaging and local sourcing where possible. Overall, Go-Grocer not only caters to modern consumer needs but also supports the growth of local grocery businesses by offering them a digital storefront. With its combination of functionality, user experience, and operational efficiency, Go-Grocer is positioned as a next-generation grocery solution — bringing convenience, reliability, and quality to every home.

Keywords: E-commerce Platform, Online Grocery Shopping, Responsive Design, PHP Backend

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Go-Grocer is an advanced e-commerce website developed to modernize the traditional grocery shopping experience. As consumer behaviour shifts toward online purchasing, Go-Grocer offers a user-friendly digital platform where customers can shop for groceries anytime, anywhere. The website provides a wide range of grocery items including fresh produce, dairy, meat, bakery goods, beverages, snacks, and household products. It is designed to meet the everyday needs of users by combining the convenience of online shopping with the reliability of local grocery stores.

The website features a clean and responsive design, ensuring seamless usability across desktops, tablets, and mobile devices. Customers can easily navigate the platform using smart search functionality, categorized listings, and personalized product recommendations. Real-time inventory updates prevent users from ordering out-of-stock items, improving overall satisfaction and reducing order errors.

One of the standout features of Go-Grocer is its focus on delivery efficiency and user control. Customers can select their preferred delivery time slots, track their orders in real time, and receive notifications about delivery progress. Subscription options are also available for frequently purchased items, allowing users to schedule repeat deliveries without manual reordering. Promotions, discounts, and loyalty programs are integrated into the platform to encourage user engagement and reward repeat customers.

On the backend, Go-Grocer offers a robust management system for administrators and vendors. Store managers can easily update product listings, monitor inventory levels, view order details, and access sales analytics to make informed business decisions. The platform is scalable, allowing it to support small neighbourhood grocery stores as well as larger supermarket chains. With its combination of powerful e-commerce tools, intuitive design, and customer-focused features, Go-Grocer is positioned as a complete solution for modern grocery shopping in the digital age.

1.2 Aim

- To develop a responsive and user-friendly e-commerce website for online grocery shopping.
- To build a secure system for user registration, login, and payment processing.
- To implement real-time product and inventory management features.
- To enable smooth order placement, tracking, and delivery scheduling.
- To design admin and vendor dashboards for efficient backend operations.
- To ensure the platform is scalable, secure, and optimized for performance.

1.3 Objective

The main objective of the Go-Grocer project is to develop a user-friendly and efficient e-commerce platform that allows customers to shop for groceries online with ease and convenience. The platform aims to offer a seamless shopping experience through features such as product categorization, search and filter options, real-time inventory updates, and personalized user accounts. In addition, the website will support order tracking, delivery scheduling, and notifications to enhance the overall customer experience.

From the backend perspective, the project seeks to build robust admin and vendor dashboards for managing product listings, tracking orders, monitoring inventory, and analysing customer data. The system will be designed to be scalable, secure, and optimized for high performance to support future growth and increase user traffic. Furthermore, the project emphasizes responsive design to ensure full functionality across desktop and mobile devices. Ultimately, the objective is to deliver a complete, modern grocery e-commerce solution that benefits both consumers and retailers alike.

Problem Statement: Many consumers face challenges with time-consuming trips to physical grocery stores and limited access to fresh products during busy schedules. There is a need for a reliable, user- friendly online platform that allows convenient, efficient, and secure grocery shopping from home.

1.4 Organization of Report

- **Introduction (Chapter 1):** Introduction introduces the Go-Grocer project, explaining its purpose, scope, and the motivation behind developing an online grocery shopping platform.
- **Problem Statement (Chapter 2):** Problem Statement defines the real-world challenges faced by consumers and highlights the need for a convenient and reliable e-commerce grocery solution.
- **Software Requirements Specifications (Chapter 3):** Software Requirements Specifications (SRS) outlines the functional and non-functional requirements, system features, and the hardware/software needed for development.
- **Literature Survey (Chapter 4):** Literature Survey reviews existing grocery platforms, compares their features, and identifies the gaps that Go-Grocer aims to address.
- **System Design (Chapter 5):** System Design presents the architecture of the system, including diagrams, database design, and the overall structure of the frontend and backend.
- **Risk Analysis (Chapter 6):** Risk Analysis identifies potential risks during development and deployment, along with strategies to minimize or mitigate them.
- **Testing (Chapter 7):** Testing describes the testing methods used, including unit and system testing, and provides an analysis of test results and bug reports.
- **Conclusion (Chapter 6):** Conclusion summarizes the project’s outcomes, discusses the achievements, and suggests future improvements or enhancements.

Chapter 2

Problem Statement

In recent years, the need for digital solutions has significantly increased as people strive for convenience and time-saving alternatives in their daily lives. Grocery shopping, a fundamental routine for every household, has not evolved at the same pace as other online services. Many people still depend on traditional shopping methods, which can be time-consuming, physically demanding, and inefficient, especially during busy or critical times.

The primary group affected by this problem includes working professionals, students, the elderly, and families with tight schedules. These groups often find it difficult to visit grocery stores regularly due to time constraints, transportation issues, or personal limitations. The lack of a practical and accessible online grocery shopping system limits their ability to shop at their convenience.

The issue becomes more critical during weekends, public holidays, or emergency situations such as pandemics and lockdowns. During such times, physical stores become overcrowded, product availability becomes uncertain, and consumers are at higher risk of inconvenience or exposure to health risks. Traditional shopping simply cannot meet the demand for speed, safety, and flexibility during these periods.

Another major challenge is geographical. In urban areas, heavy traffic and long distances between home and stores make grocery trips frustrating and inefficient. In suburban or semi-urban regions, while stores may be nearby, options are often limited, and store hours may not match the shopper's availability. The absence of a dependable digital platform increases the inconvenience for people living in such locations.

From a business perspective, small grocery vendors and local shops struggle to stay competitive against large supermarket chains that can afford in-house delivery systems and online platforms. These local vendors lack the technical infrastructure to manage inventory, handle online orders, or provide delivery services. As a result, they lose customers to bigger competitors and miss out on the opportunity to scale their operations.

Furthermore, many existing online grocery solutions suffer from limitations such as poor user experience, outdated interfaces, unreliable delivery systems, and lack of real-time inventory updates. These shortcomings reduce user trust and satisfaction, preventing mass adoption of digital grocery services. This reveals a gap in the market for a more optimized, user-friendly solution.

The absence of a well-designed platform not only affects consumer convenience but also contributes to increased time wastage, missed opportunities for businesses, and added pressure on physical stores. Customers are left without an efficient way to shop for essentials, and small vendors lose visibility and sales potential. Addressing this problem would benefit both sides of the transaction.

The core requirement is a scalable and secure e-commerce solution that allows users to browse, order, and receive groceries from the comfort of their homes. The platform should include real-time inventory tracking, delivery scheduling, secure payment options, and a smooth user interface. Additionally, it should offer backend management tools for vendors to easily handle orders, update products, and track performance.

Go-Grocer aims to address this problem by providing a complete online grocery shopping platform that caters to both customers and vendors. It seeks to offer convenience, time-efficiency, and improved accessibility for users, while empowering local stores with digital tools to reach more customers. This problem is not only relevant in today’s world but also highly solvable through thoughtful, user-focused software development.

Chapter 3

Software Requirements Specification

3.1 Introduction

The purpose of this document is to present a detailed description of Go-Grocer Website. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system. This document is the Software Requirements Specifications for Go-Grocer Website. The following document provides the complete Software Requirements Specification (SRS) for the "Go-Grocer".

3.1.1 Purpose

The main purpose of developing Go-Grocer Website is to help the customers to purchase the groceries online. After the ordering, customers can place the order in the cart or make the online payment.

The Go-Grocer Website enables vendors to set up online applications where customers can order the grocery through the online and a system administrator to maintain the set up.

The main agenda of the Online Super Market is to set up a portal where the customer can choose their vegetables, fruits, bakery products, branded products, etc online without having to visit the shop physically. Our Go-Grocer Website will use the internet as the sole method for selling groceries to its consumers.

The current system is an offline system. More physical interaction takes place in this current system between Customer and administrator. Before ordering the product, customer preview cannot be done in the existing system. The new online system will solve all the issues because online designers are implemented in the project and company can expand their business all over the world.

3.1.2 Document Conventions

This Software Requirements Specification (SRS) document follows a standardized structure to ensure clarity, consistency, and ease of understanding for all readers and stakeholders. The following conventions and formatting rules have been applied throughout the document:

- **Headings and subheadings are numbered hierarchically (e.g., 1, 1.1, 1.1.1) and presented in bold to indicate section levels and improve navigability.**
- **Bold text is used to emphasize key terms, section titles, and important requirements.**
- **Monospaced font is used to display code snippets, command-line instructions, or system messages.**

Functional and non-functional requirements are listed with unique identifiers (e.g., FR-1, NFR-2) to ensure traceability and ease of reference.

Each requirement is assigned a priority level: High (H), Medium (M), or Low (L). These are explicitly stated and not inherited from higher-level sections.

All diagrams (UML, flowcharts, etc.) are labelled and referenced within the relevant text for clarity. Tables and lists are used wherever necessary to organize complex information in a readable format**.**

3.1.3 Intended Audience and Reading Suggestions

This document is for Admin and Customers. The document aims to explain in an easy manner the basic idea behind this application. Document helps developers and clients easily understand. Future development objectives are also discussed in this document.

3.1.4 Product Scope

The main scope of this project is:

1. The system is convenient and flexible to use. It saves their time, efforts, money and resources.
2. User Friendliness is provided in the application with various controls provided by the Rich User Interface system.
3. The user information files can be stored in a centralized database which can be maintained by the system.
4. Customers can order the products online.
5. Once the customer orders the products, they can add the groceries to the cart for later purchase or can make payment through online and purchase the products.

3.1.5 References

[1] An, S., & Kim, S. (2021). Development of a Multivendor E-commerce Platform Using PHP and MySQL. International Journal of Engineering Research & Technology (IJERT), 10(11), 316-324.

[2] Gupta, R., & Sharma, A. (2020). Design and Implementation of a Multivendor E-commerce Website Using PHP and Laravel Framework. International Journal of Engineering and Advanced Technology (IJEAT), 9(5), 11695- 11702.

[3] Patel, D., Patel, S., & Rana, R. (2020). Implementation of Multivendor E-commerce Platform Using PHP and MySQL. International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), 8(3), 4156-4162.

[4] Rahman, M. M., & Khan, M. A. (2019). Design and Development of a Multivendor E-commerce Platform Using PHP and CodeIgniter Framework. International Journal of Computer Applications (0975 – 8887), 182(50), 19-24.

[5] Singh, N., & Kumar, N. (2018). Implementation of a Multivendor E-commerce Website Using PHP and MySQL. International Journal of Advanced Research in Computer Science, 9(2), 145-151

3.2 Overall Description

The following subsections provide complete overview of the SRS documentation for the product "Go-Grocer Website". For the rest of the document, we first define the overall product. Then, we give the external interface requirements, followed by a brief description of the product components and features. In the last section, we have provided the non-functional requirements of the product.

3.2.1 Product Perspective

With the ever-increasing popularity and accessibility of the internet, it is only natural that the educational community should want to make use of this tremendous resource. The administrative user interface concentrates on the consistent information that is practically part of the organizational activities, and which needs proper authentication for the data collection. The Interface helps the administration with all the transactional states like data insertion, data deletion, and data updating along with executive data search capabilities.

The operational and generic user interface helps the users with the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

3.2.2 Product Functions

The product has the following functions:

- The product works online with the help of internet connection.
- This website is secure and each customer needs to login the system to access account.

3.2.3 User Classes and Characteristics:

It is considered that the user does have the basic knowledge of operating the internet and having access to it. The administrator is expected to be familiar with the interface of the tech support system.

3.2.4 Operating Environment

These are the major components to run this software. The server needs the following tools to run the website:

- Apache Server
- MySQL Server

3.2.5 Design and Implementation Constraints

CO-1: The system's design, code, and maintenance documentation shall conform to the Process Impact Intranet Development Standard, Version
CO-2: The system shall use the current corporate standard MySQL database engine.
CO-3: All HTML code shall conform to the HTML 4.0 standard.
CO-4: All scripts shall be written in PHP.

3.2.6 User Documentation

The Go-Grocer e-commerce platform will be delivered with comprehensive user documentation to assist both end-users and administrators in effectively using the system. The following documentation components will be provided:

- **User Manual (PDF Format):** A detailed manual covering all user functionalities such as account creation, browsing products, placing orders, payment procedures, and tracking deliveries. This will be written in simple language with screenshots to guide users' step-by- step.
- **Admin & Vendor Guide (PDF Format):** A separate guide for administrators and store vendors explaining how to manage products, view and process orders, update inventory, and generate reports.
- **Quick Start Guide (Printable and Online):** A brief, easy-to-follow guide that helps new users get started quickly with essential tasks.
- **In-App Tooltips and On-Screen Instructions:** Built-in help text and hover tooltips embedded in the user interface to provide real-time guidance while using the platform.
- **Online Help Center (Web-Based):** A searchable, browser-accessible help portal containing FAQs, troubleshooting tips, and tutorials.
- **Video Tutorials (Optional - Web-Based or Embedded**): Short video walkthroughs explaining key features and usage scenarios, ideal for users who prefer visual learning.

3.2.7 Assumptions and Dependencies

- Internet connection required.
- 24X7 uptime server connection required.
- The browser version should be used which has html5 support.

- One assumption about the product is that it will always be used on mobile phones that have enough performance.
- The user should have basic computer knowledge. They should be trained to handle the features provided by the system.

3.3 External Interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface

.

3.3.1 Communication interface:

The project shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

3.3.2 Functional Requirements:

The project has the following modules:

- **Customer account module:**

  Public customers or non-registered visitors can view the groceries online, but to purchase the grocery customers need to login to the website. New customers can register to the system by entering profile details. A registered customer can post reviews/comments, and he can like or dislike the product. After the login customer can change his password or profile details. Sub modules are:

  - ➢ Customer Registration
  - ➢ Customer Login
  - ➢ Change password
  - ➢ Customer profile
  - ➢ Forgot password / Recover password.

- **Online Order module:**

  The customer can select the items and put it into a cart and as soon as he clicks on finish, the total amount to be paid is displayed. The view cart screen shows all the selected items, here quantity as to be entered. And also, option is provided to deselect the products.The customer can pay cash on delivery or else he can pay by entering his credit card number and address to which the items have to be shipped. The system generates bills after making payment.

  - ➢ Add to cart and view cart
  - ➢ Purchase form
  - ➢ Billing form
  - ➢ Payment module

- **Payment module:**

Here the customer can see the payment options available to him. When customer clicks the "Order" link, it will go straight to the "Paid" orders.

  - ➢ Credit card
  - ➢ Debit card
  - ➢ Net Banking

- **Reports module:**

Here a customer can track his order details and reports of the orders placed by him in the past. The administrator can view the order details and order reports of all the customers.

  - ➢ Sales report
  - ➢ Payment report
  - ➢ Customer report

- **Dashboard Module**:

This module is used by the administrator. Admin has full authority of the website who will monitor complete features of the website.

  - o Add category
  - o View category
  - o Add Subcategory
  - o View Subcategory
  - o Add items
  - o View items
  - o Admin profile
  - o Change password

- **Items module:**

In items module, customers can make purchase the items in just a few mouse clicks. The customer can choose the required groceries and place an order for products.

  - o Buy Items
  - o Add to cart
  - o Make Payment.

- **Category module:**

In category module, admin can add categories based on the items. Customer can view the items based on category.

  - o Add Category
  - o View Category

- **Sub-category module:**

In sub-category module, admin can add subcategories based category added in the category module. Customers can view the items based on category and sub-category.

  - Add Subcategory
  - View Subcategory

3.4.1 Performance Requirements:

This section lists the performance requirements expected from this Project:

- The users shall be able to add any records fewer than 5 seconds.
- The users shall be able to view any records fewer than 10 seconds.
- The users shall be able to download videos within 20 -40 seconds.
- The navigation between pages shall take fewer than 5 seconds.
- The application shall be able to do a validation check on the information provided in the user- authentication form and the place-order form to avoid false or incomplete information.

3.4.2 Safety Requirements:

No safety requirements have been identified.

3.4.3 Security Requirements:

- A customer needs to require logging in to the System for all operations except viewing a item.
- The system shall permit only the administrator to create or edit items.

3.4.4 Software Quality Attributes:

If the connection between the user and the system is broken prior to an order being either confirmed or canceled, the System shall enable the user to recover an incomplete order.

- **RELIABILITY**

  It is tested for all the constraints at development stage

- **AVAILABILITY**

  This system will only be available till the system on which it is installed is running.

- **SECURITY**

  This system is provided with authentication without which no customer can pass. So only legitimate customers are allowed to use the application. If the legitimate customers share the authentication information, then the system is open to outsiders.

- **MAINTAINABILITY**

  There will be no maintenance required for the website. The database is provided by the end user and therefore is maintained by this customer.

- **PORTABILITY**

  The system works anywhere with an internet connection.

Chapter 4

Literature Survey

1. **E-Commerce: Merits and Demerits – A Review Paper**
   - **Authors:** Ms. Ayushi Y. Vadwala, Ms. Maitri S. Vadwala
   - **Publication Year:** 2017
   - **Introduction:** This paper discusses the rapid growth of e-commerce due to its efficiency in exchanging goods and services globally. It emphasizes India's potential as a burgeoning market for e-commerce business models.
   - **Objectives:**
     1) To explain the concept of e-commerce.
     2) To study the merits of e-commerce.
     3) To study the demerits of e-commerce.
   - **Proposed Solution:** The study provides a conceptual understanding of e-commerce, detailing various business models and highlighting the benefits and limitations associated with e-commerce in India.
   - **Disadvantages**:
   - **Technical:**
     - System security and reliability issues.
     - Insufficient telecommunication bandwidth.
     - Integration challenges with existing applications.
     - Software/hardware compatibility issues.
   - **Non-Technical:**
     - High initial setup costs.
     - User resistance due to mistrust.
     - Security and privacy concerns.
     - Lack of physical interaction with products.
     - Limited internet access in remote areas.

2. **A Narrative Literature Review and E-Commerce Website Research**
   - **Author:** Khan Mashiur Rahman
   - **Publication Year:** 2018
   - **Introduction:** The study explores the relationship between cultural aspects and e-commerce website design, emphasizing the importance of culturally friendly designs for global e-commerce success.
   - **Objective:** To investigate how cultural factors influence e-commerce website design and to provide a reference for future research in this domain.
   - **Proposed Solution:** The research indicates correlations between e-commerce, culture, and website design, suggesting that cultural considerations are crucial for designing effective e-commerce platforms.
   - **Disadvantages:** The paper does not explicitly list disadvantages but implies that neglecting cultural aspects can hinder the success of e-commerce websites in diverse markets.

3. **Recommender Systems in E-commerce**
    - **Authors:** Tanmayee Salunke, Unnati Nichite
    - **Publication Year:** 2022
    - **Introduction:** The paper discusses the growing importance of recommender systems in e-commerce for personalizing user experiences and boosting business revenue.
    - **Objective:** To analyze the challenges associated with big data-based e-commerce recommender systems and propose potential solutions.
    - **Proposed Solution:** The study reviews different types of e-commerce recommender systems, their advantages, and disadvantages, and suggests future research directions to enhance their performance.
    - **Disadvantages:**
        - Limited resources.
        - Data validity period issues.
        - Cold start problem.
        - Long tail problem.
        - Scalability challenges.

4. **Machine Learning for Fraud Detection in E-Commerce: A Research Agenda**
    - **Authors:** Niek Tax, Kees Jan de Vries, Mathijs de Jong, et al.
    - **Publication Year:** 2021
    - **Introduction:** The paper emphasizes the importance of fraud detection and prevention in ensuring the sustained operation of e-commerce businesses, highlighting the role of machine learning (ML) in these operations.
    - **Objective:** To formulate an operational model of anti-fraud departments in e-commerce organizations and derive research topics and practical challenges for fraud detection.
    - **Proposed Solution:** The study summarizes the state of the literature for each research topic, discusses potential solutions to practical challenges, and identifies open research challenges in fraud detection.
    - **Disadvantages:** The paper outlines challenges such as integrating ML models within organizational contexts, dealing with evolving fraud tactics, and ensuring data privacy and security.

4. **The Continuous Cold Start Problem in E-Commerce Recommender Systems**
    - **Authors:** Lucas Bernardi, Jaap Kamps, Julia Kiseleva, Melanie JI Müller
    - **Publication Year:** 2015
    - **Introduction:** The paper addresses the 'Continuous Cold Start' (CoCoS) problem in e-commerce recommender systems, where users exhibit changing interests or visit the website infrequently, leading to recurring cold-start issues.
    - **Objective:** To expose the CoCoS problem and its consequences for content- and context-based recommendation in e-commerce applications.
    - **Proposed Solution:** The study illustrates the CoCoS problem with examples from a major travel recommendation website and discusses its impact on recommendation effectiveness.
    - **Disadvantages:** Challenges include dealing with users' evolving interests.

Chapter 5

System Design

4.1 Project Block Diagram

- **Introduction**

  The detailed design document is designed to explain the complete design structure of the alumni management system. In this document the overall working of each module are shown. The working of the module is represented in the form of flowchart and the description about each of the module is explained in this section. Overall the complete structure of each of the modules of the system is designed by referring the software requirement specification and database design documents.

- **Applicable documents**

  Detailed design is performed using the following documents:

  - System Requirements Specification
  - System Design
  - Database Design

- **Structure of the software package**

  Includes the following components:

  - Customer account module
  - Online Order module
  - Payment module
  - Reports module
  - Dashboard Module
  - Items module
  - Category module
  - Sub-category module

MODULAR DECOMPOSITION OF COMPONENTS:

1. Modules of Customer account:

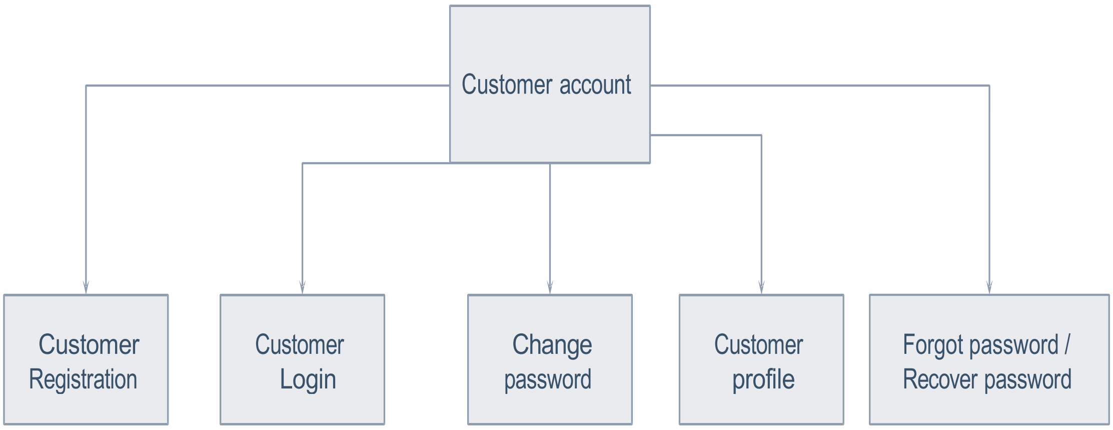
1.1. Design Assumption:

Public customers or non-registered visitors can view the groceries online, but to purchase the grocery customer needs to login to the website. New customers can register to the system by entering profile details. A registered customer can post reviews/comments, and he can like or dislike the product. After the login customer can change his password or profile details.

1.2. Identification of Modules

- Customer Registration
- Customer Login
- Change password
- Customer profile
- Forgot password / Recover password.

1.3. Structure Chart



1.4. Module Design for Customer account

custid, custname, email, mob_no, cpassword, status, addressid, custid, city_id, address, state pincode, contact no

2. Modules of Online Order:

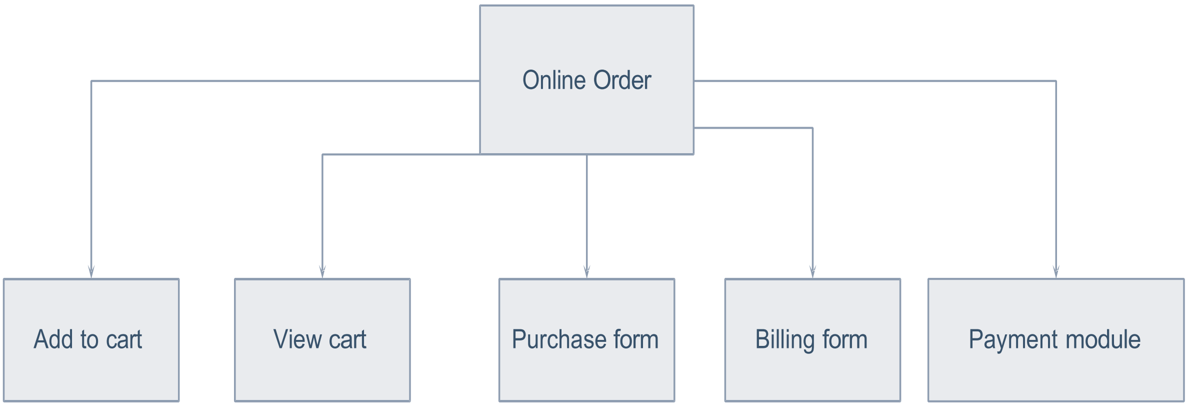
2.1. Design Assumption:

The customer can select the items and put it into a cart and as soon as he clicks on finish, the total amount to be paid is displayed. The view cart screen shows all the selected items, here quantity as to be entered. And also, an option is provided to deselect the products. The customer can pay cash on delivery or else he can pay by entering his credit card number and address to which the items have to be shipped. The system generates bills after making payment.

2.2. Identification of Modules

- Add to cart and view cart
- Purchase form
- Billing form
- Payment module

2.3. Structure Chart



2.4. Module Design for Online Order

purchid, prodid, typeid, custid, bilid, qty, price, discount_price, comment, purchasestatus

3. Modules of Payment:

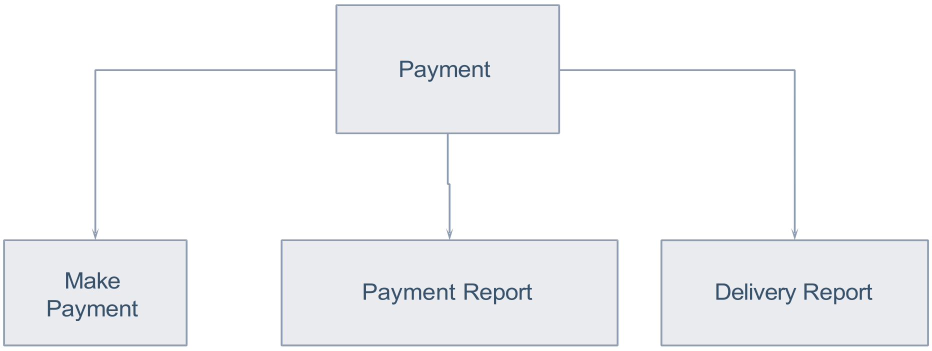
3.1. Design Assumption:

Here the customer can see the payment options available to him. When customer clicks the "Order link, it will go straight to the "Paid" orders.

3.2. Identification of Modules

- Make Payment
- Credit card
- Debit card
- Net Banking

3.3. Structure Chart



3.4. Module Design for Payment

bilid, custid, addressid, city_id, staffid, purchdate, delivdate, total_amt, cardtype, cardno, cvvno, expirydate, comment, status

4. Modules of Reports:

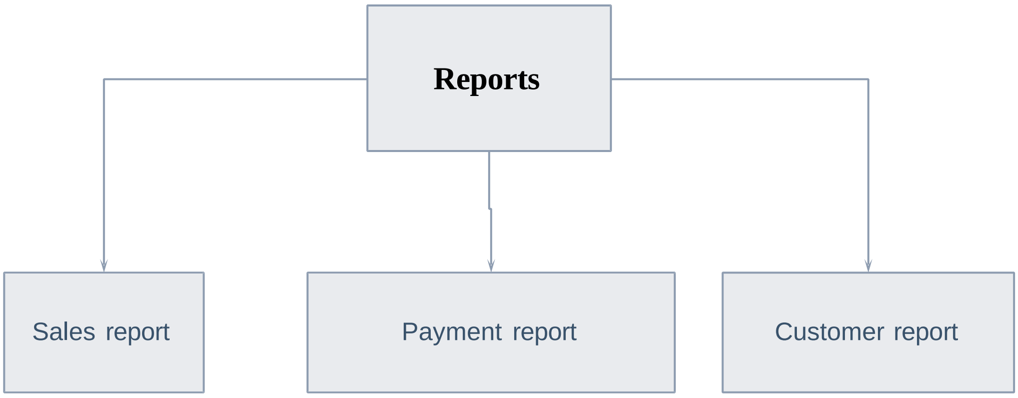
4.4. Design Assumption:

Here a customer can track his order details and reports of the orders placed by him in the past. The administrator can view the order details and order reports of all the customers.

4.5. Identification of Modules

- Sales report
- Payment report
- Customer report

4.6. Structure Chart



4.7. Module Design for Reports

purchid, prodid, typeid, custid, bilid, qty, price, discount_price, comment, purchasestatus, bilid, custid, addressid, city_id, staffid, purchdate, delivdate, total_amt, cardtype, cardno, cvvno, expirydate, comment, status

5. Modules of Dashboard:

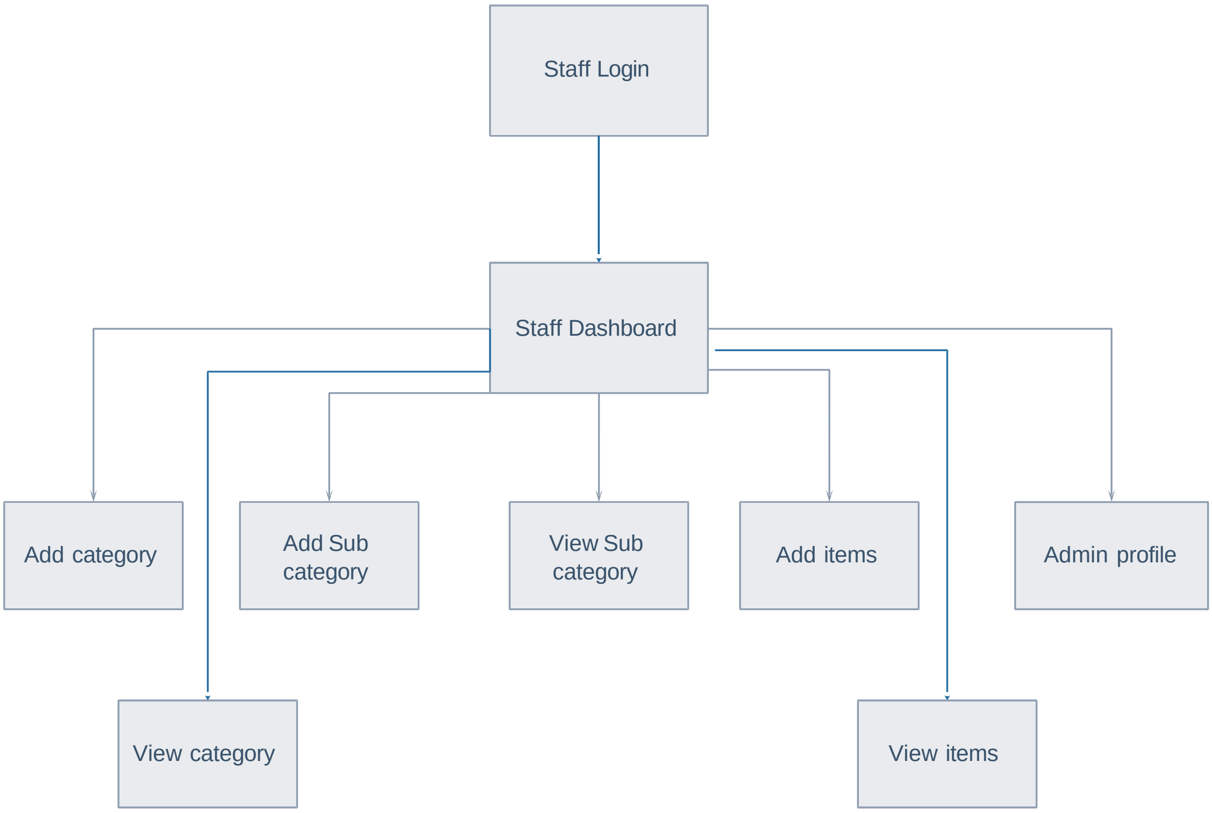
5.4. Design Assumption:

This module is used by the administrator. Admin has full authority of the website who will monitor complete features of the website.

5.5. Identification of Modules

- Add category
- View category
- Add Subcategory
- View Subcategory
- Add items
- View items

5.6. Structure Chart



5.7. Module Design for Dashboard

staffid, city_id, staff_type, staffname, loginid, apassword, emailid, contactno, status

6. Modules of Items:

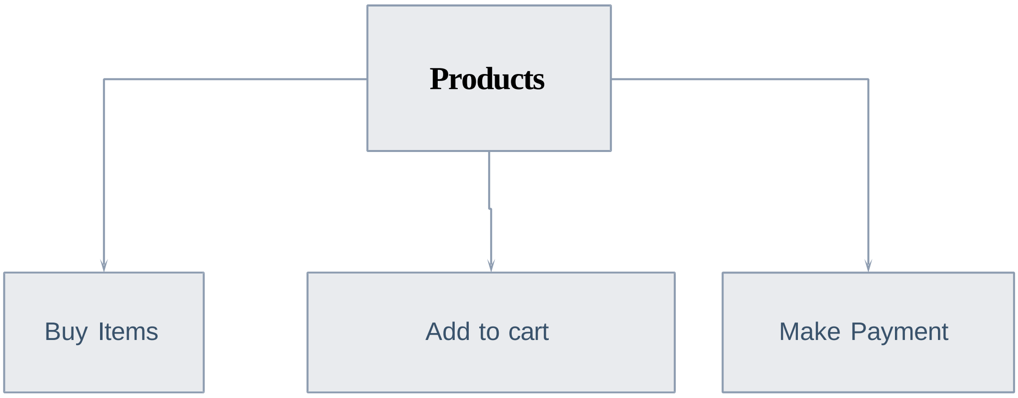
6.4. Design Assumption:

In items module, customer can purchase the items in just a few mouse clicks. The customer can choose the required groceries and place an order for products.

6.5. Identification of Modules

- Buy Items
- Add to cart
- Make Payment.

6.6. Structure Chart



6.7. Module Design for Items

prodid, catid, prodname, price, discount, unit, stockstatus, prodspecif, images, status

7. Modules of Category:

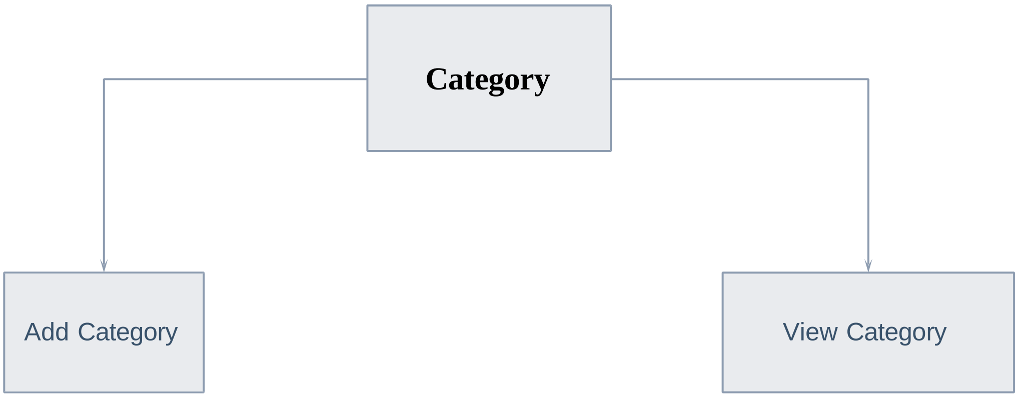
7.4. Design Assumption:

In category module, admin can add categories based on the items. Customer can view the items based on category.

7.5. Identification of Modules

- Add Category
- View Category

7.6. Structure Chart



7.7. Module Design for Category

catid, sub_catid, catgory_title, description, status

8. Modules of Sub-category:

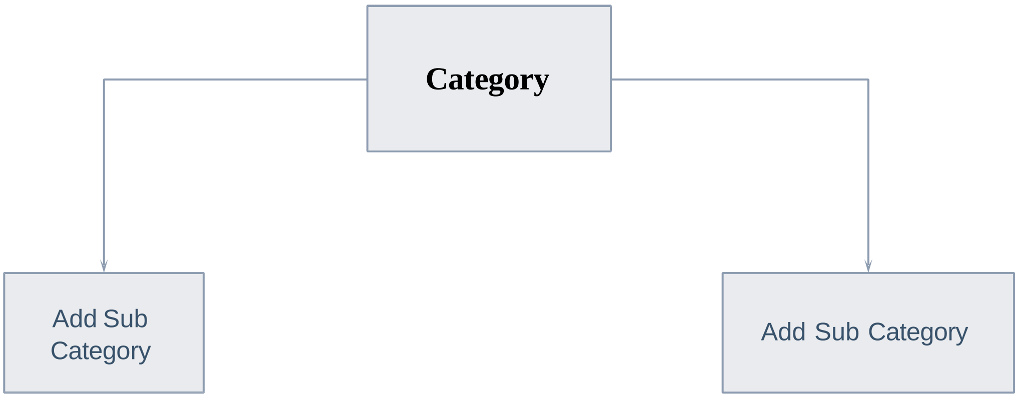
8.4. Design Assumption:

In sub-category module,admin can addsub categories based category added in the category module. Customers can view the items based on category and sub-category.

8.5. Identification of Modules

- Add Sub category
- View Sub category

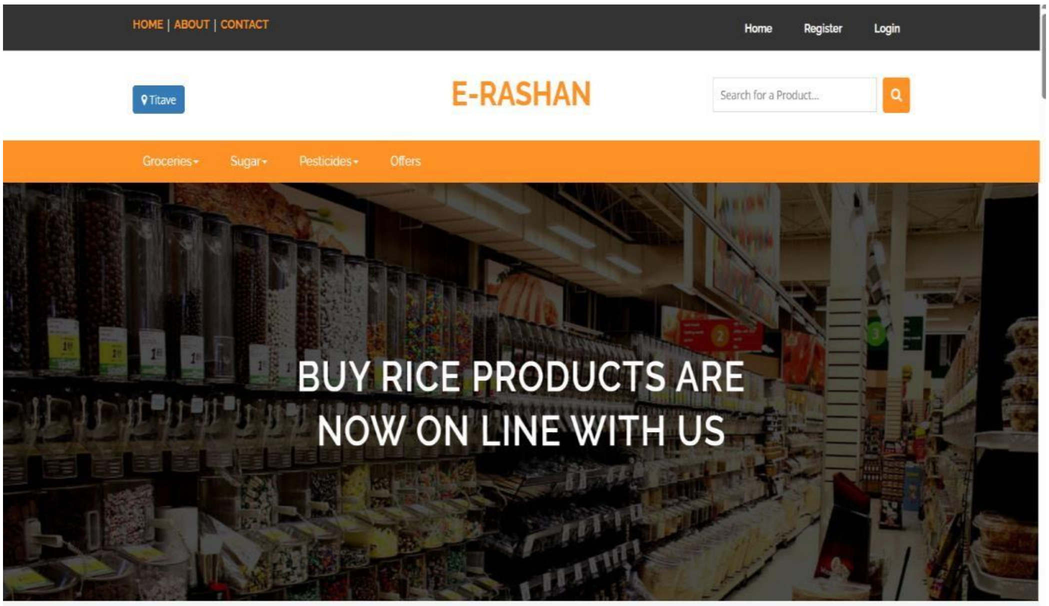
8.6. Structure Chart



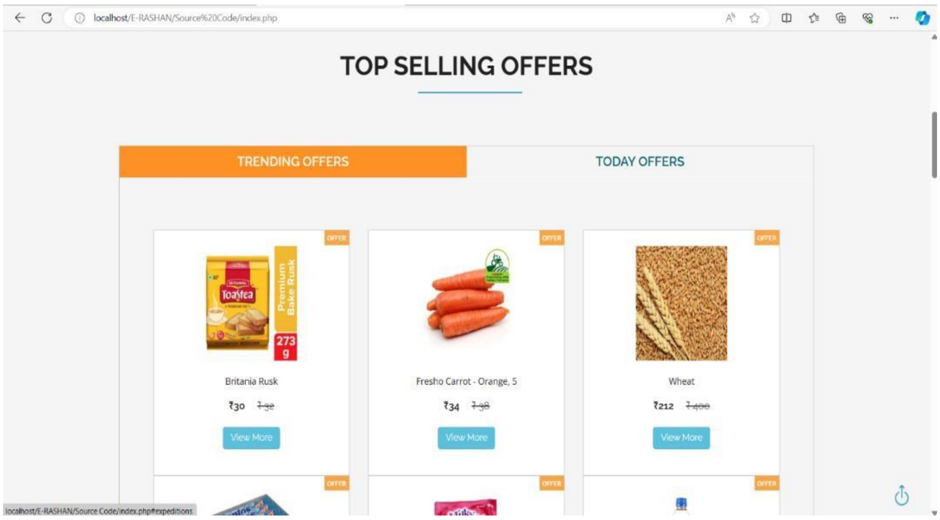
Module Design for Sub-category

catid, sub_catid, catgory_title, description, status

4.2 GUI of Working System



4.2.1 Home page for Customer


localhost/E-RASHAN/Source%20Code/index.php

TOP SELLING OFFERS

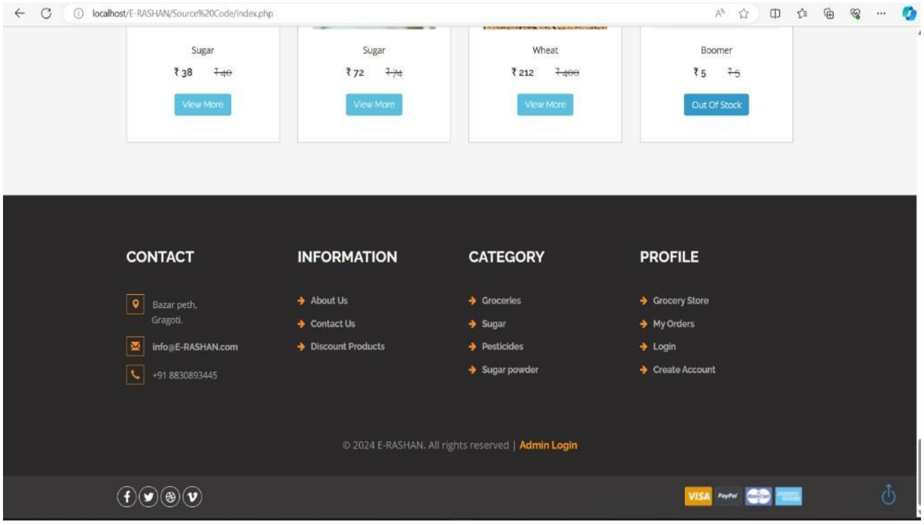
Britania Rusk
₹30 ~~₹32~~

Fresho Carrot - Orange, 5
₹34 ~~₹38~~

Wheat
₹212 ~~₹400~~

localhost/E-RASHAN/Source Code/index.php#expeditions

4.2.2 : Products


localhost/E-RASHAN/Source%20Code/index.php

Sugar
₹ 38 ~~₹40~~

Sugar
₹ 72 ~~₹74~~

Wheat
₹ 212 ~~₹400~~

Boomer
₹ 5 ~~₹5~~

CONTACT

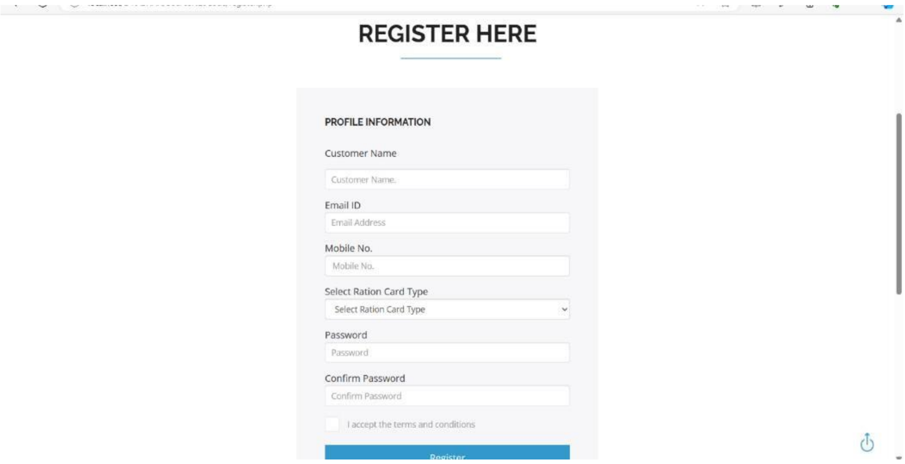
Bazar peth,
Gragoti.

info@E-RASHAN.com

+91 8830893445

© 2024 E-RASHAN. All rights reserved | Admin Login

4.2.3 : Contact info


REGISTER HERE

PROFILE INFORMATION

Customer Name
Customer Name.

Email ID
Email Address

Mobile No.
Mobile No.

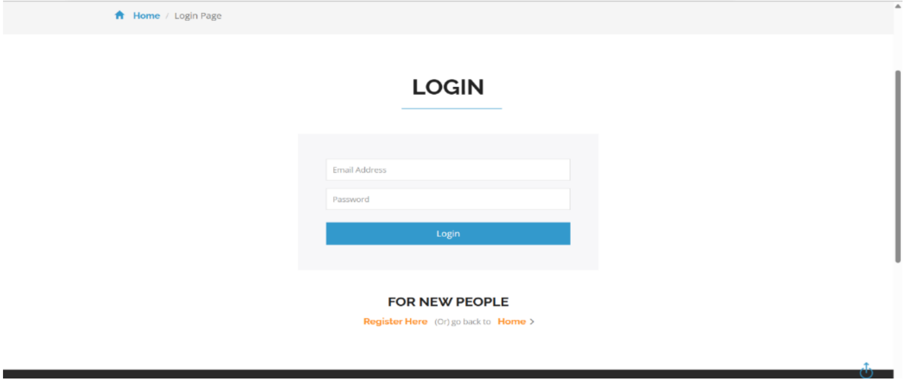
Select Ration Card Type
Select Ration Card Type

Password
Password

Confirm Password
Confirm Password

□ I accept the terms and conditions

4.2.4 : Registration Page



Home / Login Page

LOGIN

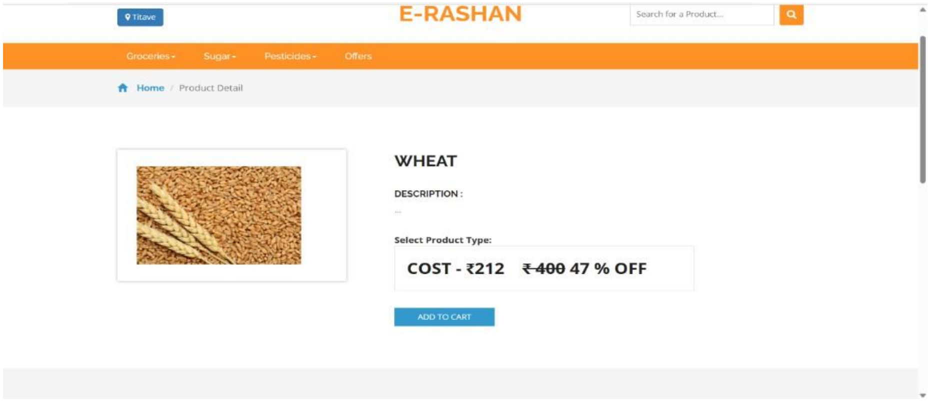
Email Address

Password

FOR NEW PEOPLE

Register Here (Or) go back to Home >

4.2.5 : Login Page



Home / Product Detail

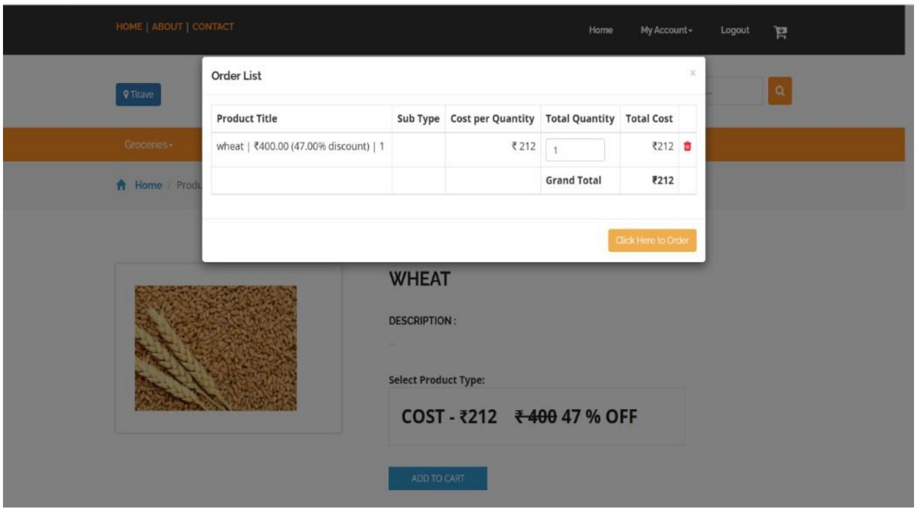
WHEAT

DESCRIPTION :

Select Product Type:

COST - ₹212 ~~₹400~~ 47 % OFF

4.2.6 : Add to cart



Order List

Product Title	Sub Type	Cost per Quantity	Total Quantity	Total Cost	
wheat | ₹400.00 (47.00% discount) | 1		₹ 212	1	₹212	🗑
			Grand Total	₹212	

WHEAT

DESCRIPTION :

Select Product Type:

COST - ₹212 ~~₹400~~ 47 % OFF

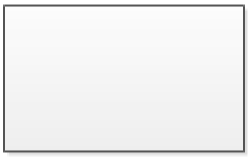
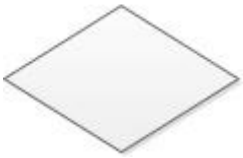
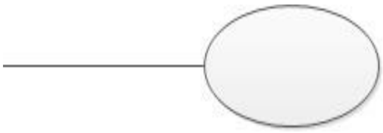
4.2.7 : Cart info

4.3UML Diagram

4.3.1ER Diagram

An entity-relationship (ER) diagram is a specialized graphic that illustrates the relationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

The Symbols are shown in below table:

Name	**Notation**	**Description**
Entity		Entity is represented by a box within the ERD. Entities are abstract concepts, each representing one or more instances of the concept in question. An entity might be considered a container that holds all of the instances of a particular thing in a system. Entities are equivalent to database tables in a relational database, with each row of the table representing an instance of that entity.
Relationship		Relationships are represented by Diamonds. A relationship is a named collection or association between entities or used to relate to two or more entities with some common attributes or meaningful interaction between the objects.
Attributes		Attributes are represented by Oval. An attribute is a single data item related to a database object. The database schema associates one or more attributes with each database entity.

ER-Diagram:

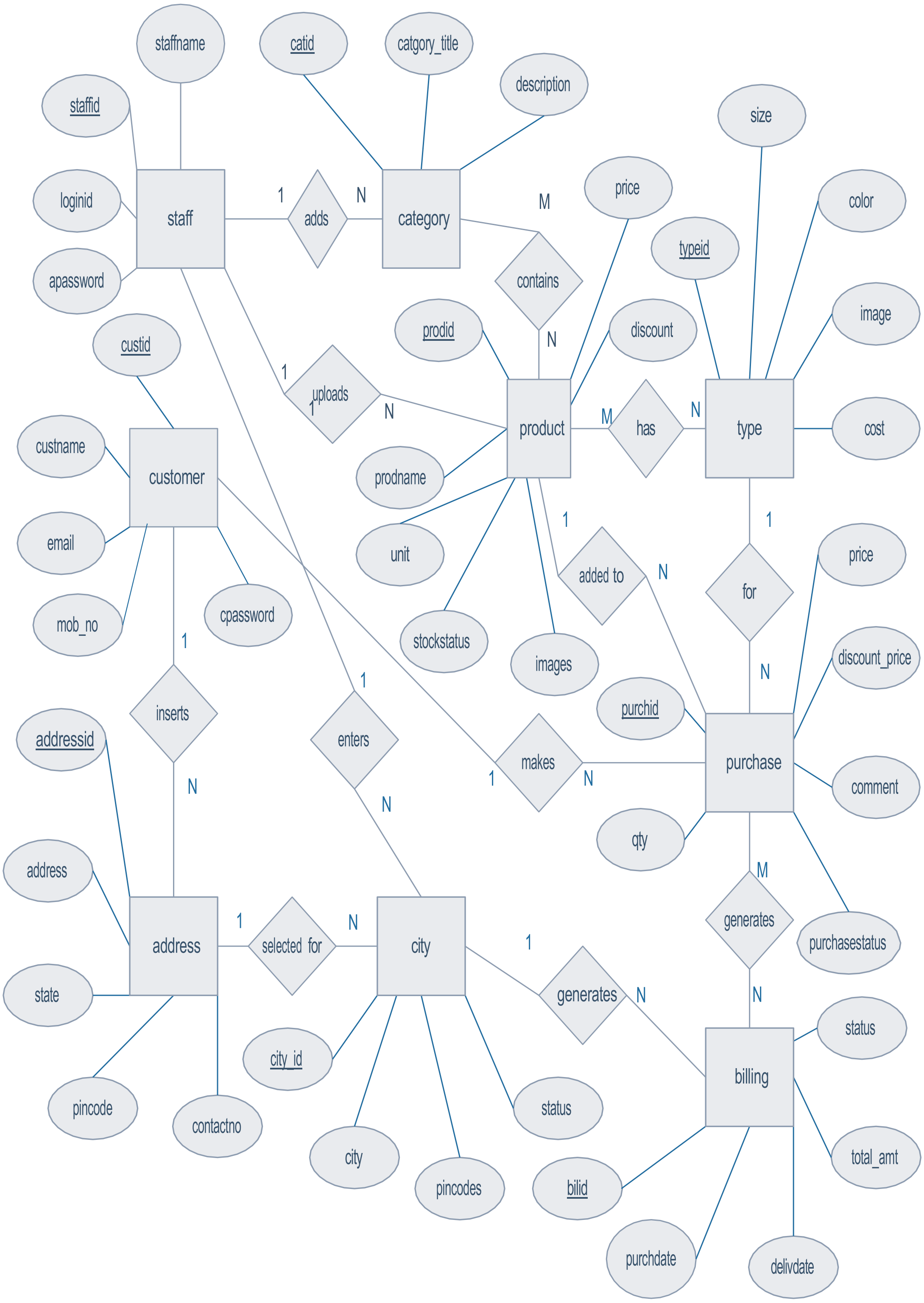


FIG 1: ER- Diagram

4.3.2 Use Case Diagram

A Use Case Diagram is a type of Unified Modelling Language (UML) diagram that visually represents the functional requirements of a system — in other words, what the system should do from the user's perspective.

Purpose of a Use Case Diagram:

- To identify and show interactions between users (called actors) and the system.
- To visualize system functionalities (called use cases) and how different users interact with them.
- It helps developers and stakeholders understand the scope and requirements of the system.

Key Components of a Use Case Diagram:

Element	Description
Actor	A user or another system that interacts with your system. It could be a human (like "Customer" or "Admin") or another application.
Use Case	A function or service the system performs (e.g., "Login", "Place Order", "Add Product"). Usually shown as an oval.
System Boundary	A box that defines the scope of the system; all use cases appear inside it.
Associations	Lines that connect actors to use cases to show interaction.
Include/Extend Relationships	Special relationships that show reusable or optional behavior between use cases.

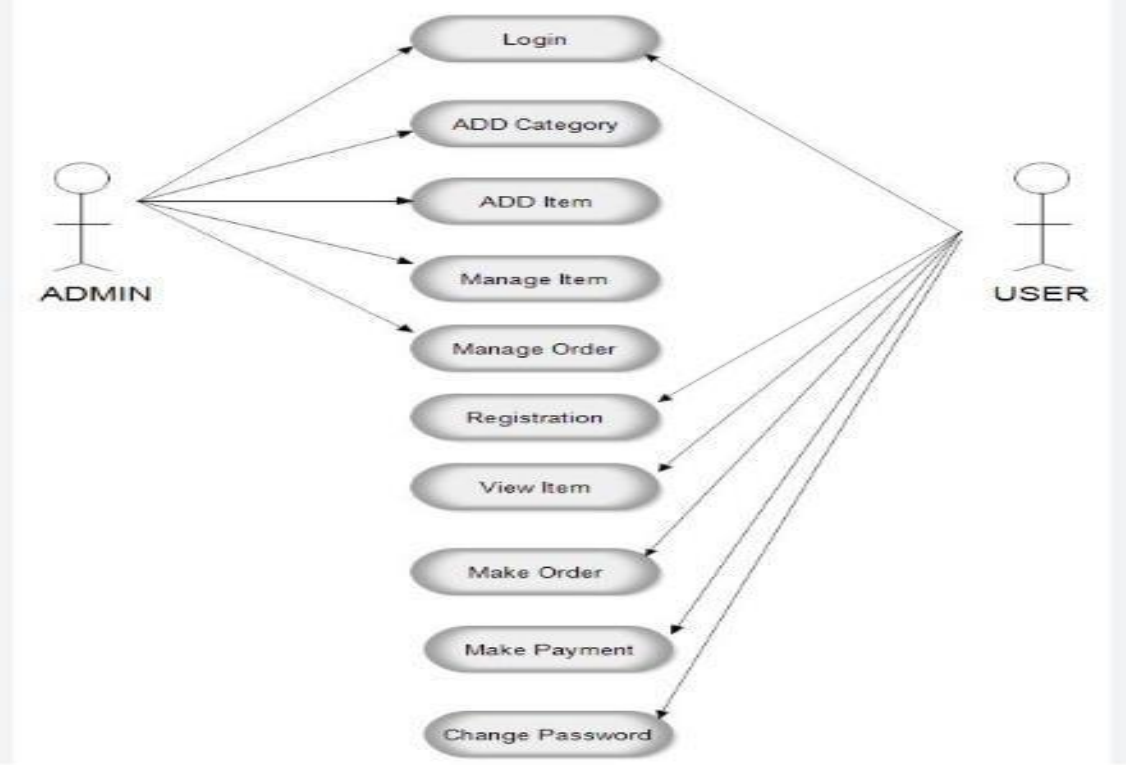


Fig 2: Use Case Diagram

4.3.3 Sequence Diagram

Sequence diagram is the pictorial representation of the interactions of the system. In this diagram using the actors and the arrows show how the system interacts with each of its actors is pictured. In an order in which the system behaves with its actors and other elements is drawn in the sequence diagram.

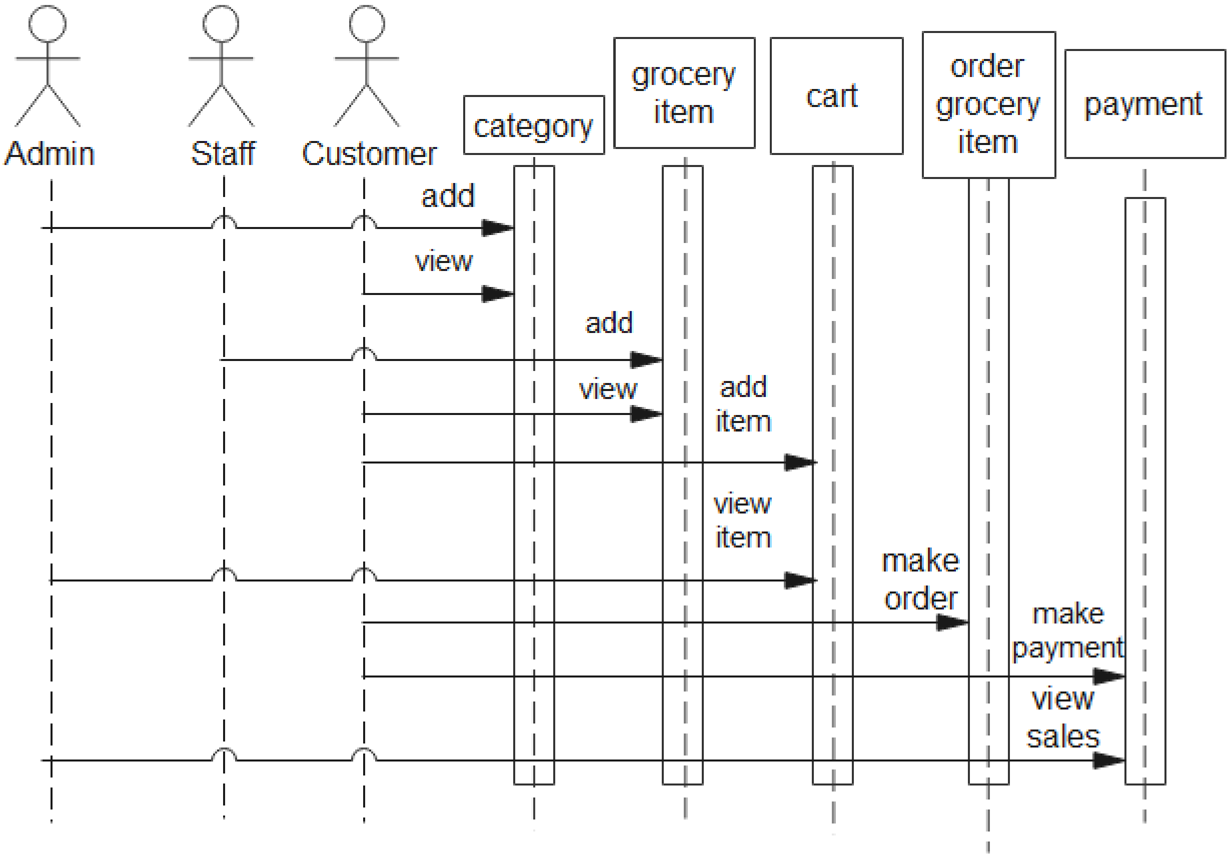


Fig 3: Sequence Diagram

4.3.4Class Diagram

Class diagram shows the relationship between the class, it's objects and attributes of the system. The relationship between each class with another class along with its attributes is depicted here. It illustrates the connection of each class of the system with one another**.**

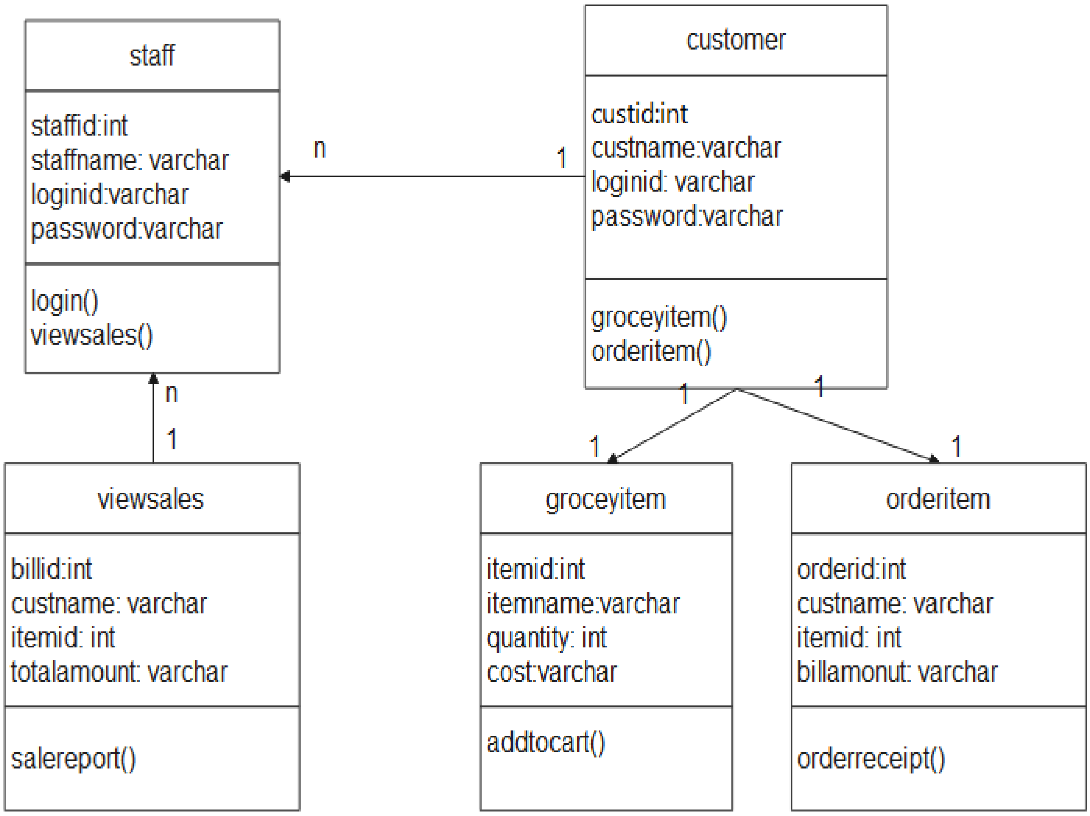


Fig 4: Class Diagram

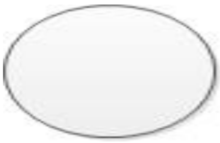
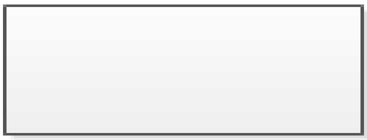
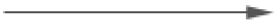
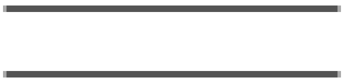
4.3.5DFD Diagram

Data flow diagram

The system design will take the system from theoretical phase to pictorial phase. The system is designed using the data flow diagrams. The data flow diagram is nothing but the representation of the flow of controls in the form of diagrams. The diagram used here is the flowchart which can be drawn using some of the symbols. The data flow diagram illustrates the working of each modules of the system, it also represents the connection between various modules. Use of DFD will make the user understand the way in which the system communicates with each of its modules.

The DFD not only represents the modules of the system, it also includes the database used in the system. This diagram shows the connection between the various modules and different tables of the database with each other. It is the DFD using which the developers of the system can design the system interface through the coding.

Notations in DFD

Symbol	Description
	An oval or a circle symbol is used which will represent the process. The process includes the main factors or the working of the system.
	A rectangular box which is used to represent the source or sink. The source or sink describes the users of the system and this is connected to the process to represent the working of system with the users.
	A straight line represents the flow of control. This is very important as it shows the connection between the process and the source or sink.
	Open box or the parallel line symbol is used to represent the tables of the database. It is used to show the connection between each of the modules and the tables of the database in the system.

Context flow diagram

Before the DFD is drawn, Context flow diagram must be drawn, that is because this gives the brief description of the working of the system. The DFD illustrates the working of each module, where as the context flow diagram illustrates the communication between the different actors of the system with each other as well as with the database. The main function of each actor of the system is highlighted here and followed by which the DFDs are designed. Context flow diagram is also a DFF, but since it gives the overall description it is known as CFD.

Context flow diagram (Level 0)

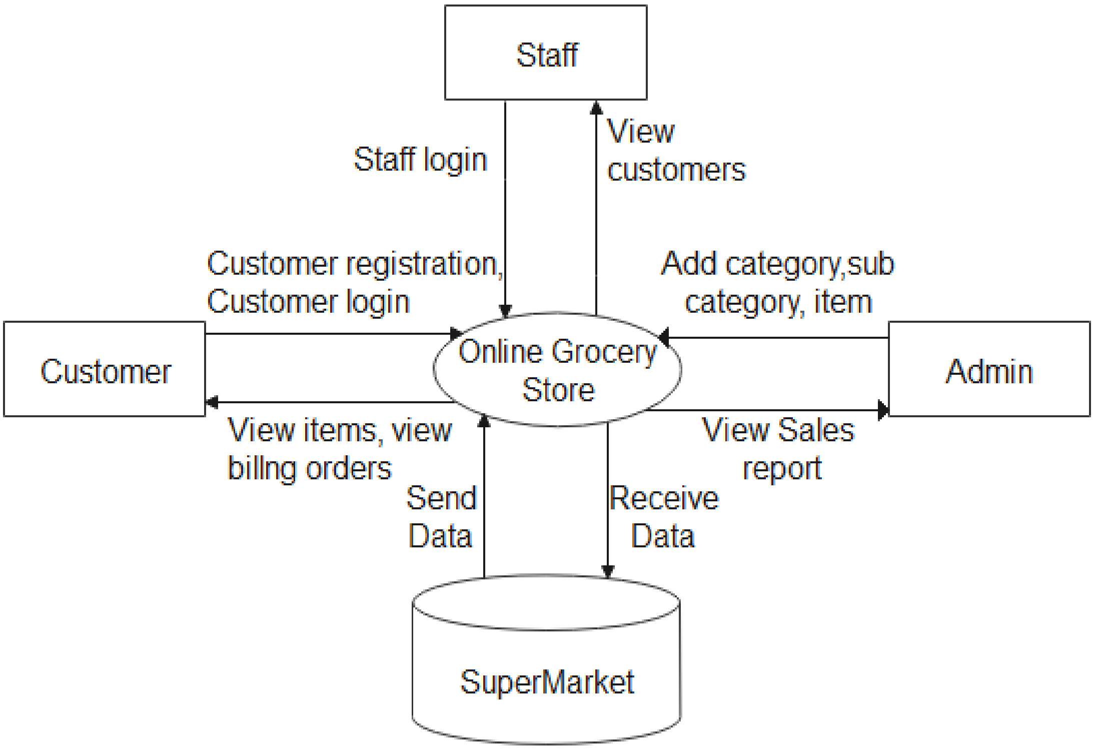


Fig 5: DFD Level 1

Top Level: DFD (Level 1):

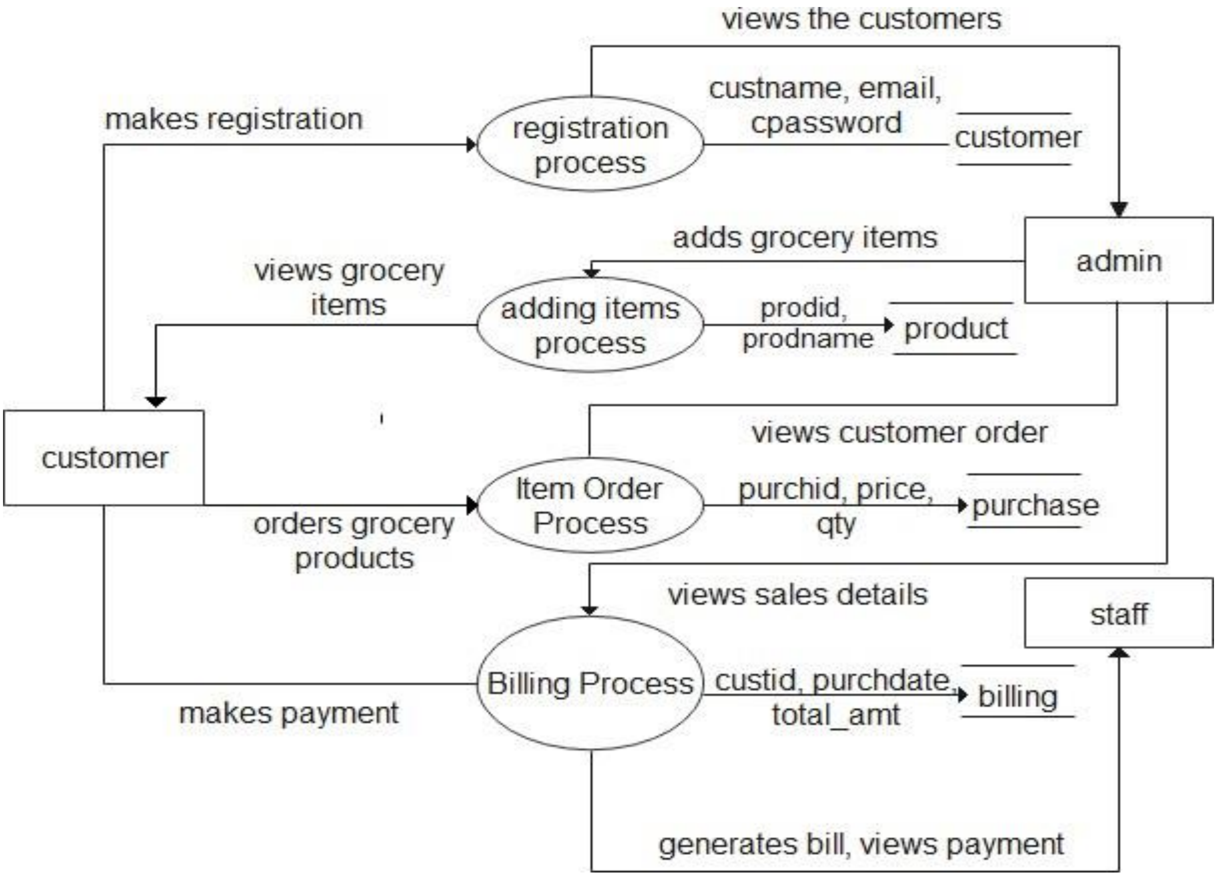


Fig 6: DFD Level 1

5.1.1 Activity Diagram

Activity diagram illustrates the entire activity of the system. All the actions that must happen in the system from the start till the end is shown in the activity diagram. Activity diagram shows all the decisions that is made in every step of the system

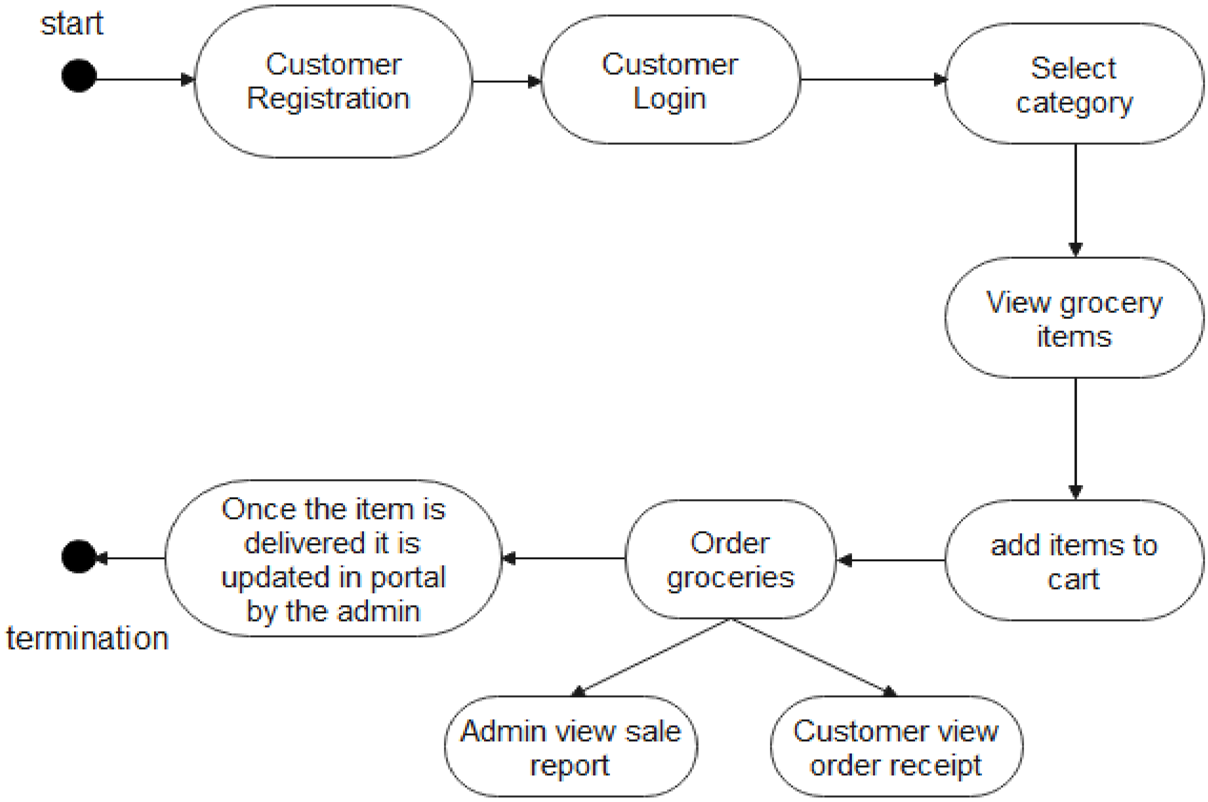


Fig 7: Activity Diagram

Chapter 6

Risk Analysis

6.1 RISK MANAGEMENT W.R.T. NP HARD ANALYSIS

This section outlines the risk management strategy adopted for the Go-Grocer project. It highlights how potential risks are identified, analyzed, mitigated, and monitored. The goal is to reduce project delays and maintain product quality within the set timelines and budget.

6.1.1 Risk Identification

Risk identification was conducted through a detailed review of the scope document, requirement specifications, design documentation, and team feedback. A risk identification questionnaire was used, addressing key areas like team commitment, user involvement, requirements clarity, and resource adequacy.

Key risk sources:

- Software Requirements Specification (SRS)
- Design Specification Review
- Early development feedback

Risk Identification Questionnaire:

1. Are key stakeholders committed to the Go-Grocer project?
2. Are vendors and users fully engaged in the platform design?
3. Does the development team clearly understand all system requirements?
4. Are there stable and well-documented requirements?
5. Is the team size and skillset adequate for timely delivery?
6. Do all users and vendors align with the project’s goals and expectations?

ID	Risk Description	Probability	Impact (Schedule)	Impact (Quality)	Overall Impact
1	Misinterpretation of vendor requirements	Low	Low	High	High
2	Unclear admin panel functionalities	Low	Low	High	High
3	Third-party payment gateway integration	Low	Medium	Very High	High

6.1.2 Risk Analysis

The following table outlines the key risks identified for Go-Grocer, their probability, and expected impact:

Risk Probability Definitions

Probability	Value	Description
High	> 75%	Likely to occur frequently
Medium	26% – 75%	Possible under certain conditions
Low	< 25%	Rare, but still a consideration

Risk Impact Definitions

Impact	Value	Description
Very High	> 10%	Major schedule delays or unacceptable product quality
High	5 – 10%	Some delay or partial failure in features or usability
Medium	< 5%	Minimal delays or manageable quality compromises
Low	Negligible	Easily fixable with minimal impact

Overview of Risk Mitigation, Monitoring, Management:

Risk ID	Risk Description	Category	Source	Probability	Impact	Response	Strategy	Risk Status
1	Misinterpretation of vendor needs	Development Environment	SRS Document	Low	High	Mitigate	Conduct regular vendor interviews and feedback loops	Occurred
2	Unclear admin panel functionalities	Requirements	Design Review	Low	High	Mitigate	Improve UI mockups and involve admin stakeholders	Identified
3	Payment gateway issues	Technology	Early testing	Low	Very High	Accept	Use sandbox testing and fallback gateways	Identified

6.2 PROJECT SCHEDULE

6.2.1 Project Task Set

- Task 1: Requirement gathering and analysis
- Task 2: Design of database and UI prototypes
- Task 3: Backend and frontend development
- Task 4: Integration of vendor and admin modules
- Task 5: Testing, debugging, and deployment

6.2.2 Task Network

A network diagram showing task dependencies and timelines will be illustrated in the project planner (Annex C). Please refer to Annex C for the full Gantt chart and task breakdown.

6.2.3 Timeline Chart

The timeline will detail key project phases across weeks or months, highlighting milestones such as:

- Completion of UI/UX design
- Admin panel integration
- Final testing
- Live deployment

6.3 TEAM ORGANIZATION

6.3.1 Team Structure

The Go-Grocer team is organized as follows:

- Project Manager: Oversees the development and delivery timeline
- Frontend Developer: Designs user-facing components
- Backend Developer: Implements server logic and database interactions
- UI/UX Designer: Designs user interfaces for web and mobile
- QA Engineer: Ensures the quality of the system via testing
- Documentation & Support Lead: Prepares manuals and handles user queries

6.3.2Management Reporting and Communication

- Weekly stand-ups and progress meetings
- Shared progress tracking via project management tools (e.g., Trello, Jira)
- Regular documentation reviez
- Email updates to stakeholders and internal reports
- Lab sessions and online collaboration tools (e.g., Google Meet, GitHub) to ensure smooth development and communication

Chapter 7

Testing

Introduction

Once the system is designed it is necessary to perform the testing as to assure whether all the modules of the system is designed correctly. Testing is the prominent document in the problem-solving methodology. In this document the different validations are checked and tested for the correctness of the system.

Levels of testing

Unit testing

In this testing, each unit is tested separately or individually to know it's performance when checked solely. Any faults in each unit is corrected in this phase.

Integration testing

In this type of testing, two or more modules are combined and tested for their performance. When two or more modules are integrated, the working of those modules along with each other is identified. If the modules don't perform well in this phase they are tested again individually and then modified and integrated again.

System testing

The system testing is the last level of testing, here all modules in system are put together and tested for any errors and ambiguities. If the system performs well, then it is further processed and the system is approved otherwise the testing phase is performed again until the system performs correctly.

Test Reports:

- **Test Unit: Admin component**
- **Admin login:**

Serial No.	Condition to be tested	Test Data	Expected output	Remarks
1.	If login ID is not entered	loginid	Please enter your login ID.	SUCCESS
2.	If password is not entered	apassword	Please enter the password.	SUCCESS
3.	If login ID and password are not valid	loginid, apassword	Invalid login credentials.	SUCCESS

- **Test Unit: Staff component**
- **Add Staff**

Serial No.	**Condition to be tested**	**Test Data**	**Expected output**	**Remarks**
1.	If city is not selected	city_id	Kindly select city	SUCCES
2.	If staff type is not selected	staff_type	Kindly select staff type	SUCCES
3.	If staff name contains other than characters.	staffname	Staff name is not valid	SUCCES
4.	If staff name is not entered	staffname	Staff name should not be empty	SUCCES
5.	If login ID is not entered	loginid	Login ID should not be empty	SUCCES
6.	If password is less than 6 characters	spassword	Password should contain more than 6 characters	SUCCES
7.	If password is not entered.	spassword	Password should not be empty	SUCCESS
8.	If password and confirm password does not match	spassword	Password and confirm password are not matching	SUCCESS
9.	If confirm password is not entered	Confirmpassword	Confirm password should not be empty	SUCCESS
10.	If staff email id is not entered	emailid	Staff email id should not be empty.	SUCCESS

- **Staff login:**

Serial No.	**Condition to be tested**	**Test Data**	**Expected output**	**Remarks**
1.	If login ID is not entered	loginid	Please enter your login ID.	SUCCESS
2.	If password is not entered	apassword	Please enter the password.	SUCCESS
3.	If login ID and password are not valid	ginid, apassword	Invalid login credentials.	SUCCESS

- **Test Unit: Customer component**
- **Customer registration:**

Serial No.	**Condition to be tested**	**Test Data**	**Expected output**	**Remarks**
1.	If customer name is not entered.	custname	stomer name should not be empty	SUCCESS
2.	f customer name contains other than characters.	custname	Customer name not valid	SUCCESS
7.	If email id is invalid	email	Entered Email Id is not valid	SUCCESS
8.	If email id is not entered.	email	Email id should not be empty	SUCCESS
4.	mobile number is not equal to 10 digits.	mob_no	Entered Mobile Number should contain 10 digits	SUCCESS

- **Customer login:**

Serial No.	**Condition to be tested**	**Test Data**	**Expected output**	**Remarks**
1.	If email id is invalid	email	Entered Email Id is not valid	SUCCESS
2.	If email id is not entered.	email	Email id should not be empty	SUCCES

- **Test Unit: Product component**
- **Add City:**

Serial No.	Condition to be tested	Test Data	Expected output	Remarks
1.	If city is not entered	city	City should not be empty	SUCCESS
2.	If pincode is not entered	pincodes	Pincode should not be empty	SUCCESS
3.	description is not entered	description	Description should not be empty	SUCCESS
4.	If status is not selected.	status	Kindly select status	SUCCESS

- **Add Category:**

Serial No.	Condition to be tested	Test Data	Expected output	Remarks
1.	If main category is not entered.	sub_catid	Kindly select main category	SUCCESS
2.	If category is not entered.	catgory_title	Category should not be empty	SUCCESS
3.	If category description is not entered.	description	Category description should not be empty	SUCCESS

- **Add Product:**

Serial No.	Condition to be tested	Test Data	Expected output	Remarks
1.	If product name is not entered	prodname	Product name should not be empty	SUCCESS
2.	If price is not entered	price	Price should not be empty	SUCCESS

4.	If image is not selected	file	Kindly select image	SUCCESS
5.	If product specification is not entered	prodspecif	Product specification should not be empty	SUCCESS
6.	If stock status is not selected	stockstatus	Kindly select stock status	SUCCESS
7.	If status is not selected	status	select status	SUCCESS

- **Test Unit: Stock component**
- **Add Stock entry:**

Serial No.	**Condition to be tested**	**Test Data**	**Expected output**	**Remarks**
1.	If bill number is not entered.	bill_no	Bill number not be empty	SUCCESS
2.	If purchase date is not entered.	purchdate	Category description should not be empty	SUCCESS
3.	If city is not selected.	city_id	select city	SUCCESS
4.	If seller is not selected	cust_id	select seller	SUCCESS
5.	comment is not entered	comment	Comment should not be empty	SUCCESS
6.	If product title is not selected	prodid	Kindly select product title	SUCCESS
7.	f quantity is not entered	qty	Quantity should not be empty	SUCCESS
8.	If cost is not entered	price	Cost should not be empty	SUCCESS

Chapter 8

Conclusion

The project work titled “Go-Grocer Website” has been designed using PHP (Hyper Text Pre Processor) where in many user friendly from controls have been added in order to make it a user interactive application. The system developed in such a way that the user with common knowledge of computers can handle it easily.
More than anything this project has given us great satisfaction in having designed an application, and in one place, which saves precious time and transfers the user’s needs into a software solution.

FUTURE ENHANCEMENT

- This application can be further implemented for Android and iPhones.
- Tracking system (GPS) can be further used for delivery purpose.

References

[1] An, S., & Kim, S. (2021). Development of a Multivendor E-commerce Platform Using PHP and MySQL. International Journal of Engineering Research & Technology (IJERT), 10(11), 316-324.

[2] Gupta, R., & Sharma, A. (2020). Design and Implementation of a Multivendor E-commerce Website Using PHP and Laravel Framework. International Journal of Engineering and Advanced Technology (IJEAT), 9(5), 11695- 11702.

[3] Patel, D., Patel, S., & Rana, R. (2020). Implementation of Multivendor E-commerce Platform Using PHP and MySQL. International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), 8(3), 4156-4162.

[4] Rahman, M. M., & Khan, M. A. (2019). Design and Development of a Multivendor Ecommerce Platform Using PHP and CodeIgniter Framework. International Journal of Computer Applications (0975 – 8887), 182(50), 19-24.

[5] Singh, N., & Kumar, N. (2018). Implementation of a Multivendor E-commerce Website Using PHP and MySQL. International Journal of Advanced Research in Computer Science, 9(2), 145151.