

**HATLEY**

# **HATLEY**

(Delivery Website &app)

Graduation Project Report

Prepared by

**Abdullah Salah**

**Abd El-Rahman Omar**

**Ahmed Ali Farahan**

**Santy Osama Mina**

**Rahma Bahaa**

Academic Supervisor

**Dr. Ali Hussein Ahmed**

**Faculty of computers and information (information system),**

**Assiut University**

**2023-2024**

# **Table of Contents:**

|  |
| --- |
| Table of Contents ii  Abstract 4  1. Software Proposal 5  1.1 Introduction 5  1.2 Problem statement 6  1.2.1 User Perspective 6  1.2.2 Delivery Perspective 7  1.3 Objectives 7  2. Software Analysis 9  2.1 Introduction 9  2.1.1 Problem Analysis and Motivation 9  2.1.2 Scope of the Project 10  2.1.3 Target User Group 11  2.2 System Requirement 12  2.2.1 Project Sponser 12  2.2.2 Business Need 12  2.2.3 Business Value 13  2.3 Fesability Study 13  2.3.1 Technical Feasibility 14  2.3.2 Economica Feasibility 15  2.3.3 Organizational Feasibility 16  2.4 The Survey 16  2.5 Definition of Requirements 22  2.5.1 Functional Requirements 22  2.5.2 Non-Functional Requirements 25  2.6 Use Cases 27  2.6.1 Register Use Case 27  2.6.2 Login Use Case 28  2.6.3 Display requests and Interaction Use Case 29  2.6.4 Create Order Use Case 30  2.6.5 Accept offer Use Case 31  2.6.6 Introduce offer Use Case 31  2.6.7 Notification and Alert Use Case 32  2.6.8 Customer Support Use Case 33  2.6.9 Driver Ratings for Customers Use Case 34  2.6.10 Request Tracking and Delivery Use Case 35  2.6.11 User Ratings and Comments Use Case 37  2.6.12 Location and Address Selection Use Case 38  2.6.13 Order Confirmation and Payment Use Case 39  2.6.14 Request Processing and Assignment Use Case 40  2.6.15 Admin control Use Case 42  2.7 Management Plan 43  2.7.1 Current State of the Project 43  2.7.2 Phases 44  2.7.3 Gantt Chart & Timeline 46  2.7.4 Task Plan with Milestones 47  2.7.5 Division of Responsibilities Among Team Members 48  2.7.6 Risk Management 50  3. Software Design  3.1 Introduction  3.1.1 Project Background/Purpose  3.1.2 Hosting Platform  3.2 System Architecture  3.2.1 Architectural Design  3.2.2 Decomposition Description  3.2.3 Design Rationale  3.3 Data Design  3.3.1 Data Description  3.3.2 Data Dictionary  3.4 Components Design  3.4.1 User Authentication Module  3.4.2 Data Storage Module  3.4.3 User Interface Module  3.4.4 Data Processing Module  3.4.5 Data Flow Diagram  3.4.6 Vitalism End-to-End Framework  3.5 User Interface Design  3.6 Testing and Quality Assurance  4. Software Requirements Specification  4.1 Introduction  4.1.1 Intended Audience and Reading Suggestions  4.1.2 Product Scope  4.2 Overall Description  4.2.1 Product Perspective  4.2.2 Product Functions  4.2.3 User Classes and Characteristics  4.2.4 Operating Environment  4.2.5 Design and Implementation Constraints  4.2.6 User Documentation  4.2.7 Assumptions and Dependencies  4.3 External Interface Requirements  4.3.1 Hardware Interfaces  4.3.2 Software Interfaces  4.3.3 Communications Interfaces  4.4 System Features  4.5 Other Non-functional Requirements  4.5.1 Performance Requirements  4.5.2 Safety Requirements  4.5.3 Security Requirements  4.5.4 Software Quality Attributes  4.5.5 Business Rules  5. Conclusion  6. Future Works |

# **Abstract:**

Welcome to Hatley! Hatley is a platform that connects users with delivery personnel, providing a faster and more convenient way to purchase daily essentials in Egypt.

Hatley is a comprehensive delivery platform designed to provide a seamless and convenient delivery experience for our users in Egypt. Our goal is to create a competitive marketplace for delivery drivers, encouraging them to provide faster and more cost-effective delivery options for users.

Hatley is a website that connects users with delivery drivers who compete to fulfill user orders in the shortest amount of time for a competitive price. By allowing users to submit their requests for products and specify a preliminary price and time, the platform creates a competitive marketplace for delivery drivers, encouraging them to provide faster and more cost-effective delivery options for users.

Delivery men can then compete for these orders by either accepting the offer or presenting a competitive alternative in terms of time and cost, it can lead to faster and more efficient deliveries.

At Hatley, we strive to provide a fast, efficient, and transparent delivery service that meets the needs of modern users and delivery personnel. Our platform offers a user-friendly app, real-time notifications and updates, and a rating and review system for both users and delivery personnel.

Hatley can help users save time and effort when it comes to ordering their daily needs. The idea of competition between delivery men based on time and price is an interesting concept that can lead to faster and more efficient deliveries.

# **1. Software Proposal**

## **1.1 Introduction:**

Hatley is a web-based delivery platform that is designed to provide a more competitive and efficient delivery ecosystem in Egypt. The platform connects users with delivery drivers who compete to fulfill user orders in the shortest amount of time for a competitive price. By allowing delivery drivers to take the full price of delivery, Hatley provides them with more earning opportunities and promotes a more transparent and accountable delivery service for users.

The Website includes several key features, such as user registration and account management, product request submission and tracking, competitive delivery pricing and time offers, notification system for order updates, user-friendly interface, payment system, and review and rating system for delivery personnel and user experience. These features are designed to provide a seamless and convenient delivery experience for both users and delivery drivers.

Hatley allows users to order a variety of products by submitting their request through the app, specifying a preliminary service price and time. Delivery personnel compete for these orders by either accepting the offer or presenting a competitive alternative in terms of time and cost. Notifications are specifically sent to the customer, giving them the freedom to choose the best delivery option based on price and time. The delivery personnel then proceed to any nearby or distant store where the product is available, purchase it, and deliver it to the location specified by the customer through the app. This creates job opportunities for young individuals and transforms the Egyptian market into a competitive space, enhancing product quality and increasing revenue for some stores without the need for advertisements.

In this software proposal, we will outline the features and functionality of the Hatley platform, as well as the development process and timeline. We will also provide an overview of the project scope, including project objectives, user requirements, and technical specifications.

## **1.2 Problem statement:**

### 1.2.1 User Perspective:

people often face significant challenges when it comes to purchasing daily essentials. The country's notorious traffic congestion can make it difficult and time-consuming to go to physical stores, especially during peak hours. Additionally, long queues and limited parking spaces can further add to the frustration and inconvenience of in-store shopping.

Moreover, People with busy schedules may face challenges in accessing delivery services due to their limited time and the need to go to multiple stores for their daily needs.

These challenges can lead to a lack of access to a wide range of products and services, as people may not have the time or patience to visit multiple stores to find what they need. This can be particularly challenging for those who live in remote or underserved areas, where access to physical stores may be limited.

Moreover, small and medium-sized stores often struggle to compete with larger stores, who have the resources and scale to offer lower prices and a wider range of products. This can lead to lower revenue and limited product availability for smaller stores, which can further limit access to essential goods and services for consumers.

In addition to the delay of the delivery companies in responding and waiting a long time to confirm your order, then waiting for another period of time until the delivery company arrives at the place of order, purchases it, and then brings it to you. This takes a very long time and is considered a waste of time.

Hatley aims to address these challenges by providing a fast, convenient, and transparent delivery service that connects users with a wide range of products and services.

By allowing delivery personnel to compete for orders based on price and time, the platform can help to incentivize faster and more efficient delivery services, while also providing users with greater choice and flexibility. Additionally, by providing job opportunities for young individuals and helping to transform the Egyptian market into a more competitive space, the platform can help to increase revenue and product availability for smaller stores, while also providing consumers with greater access to essential goods and services.

### 1.2.2 Delivery Perspective:

The traditional delivery market is often characterized by high delivery fees, lack of competition, and undervalued delivery drivers. Delivery companies often take a large percentage of the delivery fee, leaving delivery drivers with a smaller portion of the earnings. This can lead to delivery drivers feeling undervalued and exploited, as well as a lack of competition in the delivery market. Additionally, high delivery fees can be a burden for users and small businesses, limiting their ability to access delivery services.

Hatley aims to address these issues by allowing delivery drivers to take the full price of delivery, providing them with more earning opportunities and promoting a more transparent and accountable delivery service for users. The platform's competitive pricing and time offers can also benefit users and small businesses by providing more cost-effective delivery options.

By promoting a more competitive and efficient delivery ecosystem, Hatley aims to benefit both users and delivery drivers, promoting a more sustainable and profitable delivery market. The platform's unique business model and features can help address the issues of undervalued delivery drivers and high delivery fees, providing a more equitable and affordable delivery solution.

## **1.3 Objectives:**

Hatley has many objectives, this including the following:

* To develop a user-friendly and accessible website and mobile app that allows users to easily submit their requests for daily essentials and track the delivery status.
* To ensure transparency and fair competition, Hatley can implement various measures, such as:

1. Clearly displaying delivery fees and prices for each delivery option on the platform.
2. Providing users with the option to sort and filter delivery options based on price, delivery time, and other relevant factors.
3. Encouraging delivery personnel to provide accurate and up-to-date information about their delivery times, prices, and service quality.
4. Implementing a rating and review system that allows users to rate and review delivery personnel based on their service quality and delivery time.

* To provide a fast and efficient delivery service that connects users with delivery personnel. Hatley aims to provide a fast and efficient delivery service that connects users with delivery personnel. By allowing delivery personnel to compete for orders based on price and time, Hatley can help to incentivize faster and more efficient delivery services, ultimately benefiting both delivery personnel and users.
* To provide job opportunities for young individuals in the delivery industry. Hatley aims to provide job opportunities for young individuals in the delivery industry by connecting them with users who need delivery services. By allowing delivery personnel to compete for orders based on price and time, Hatley can help to create a more level playing field for young and inexperienced delivery personnel, ultimately helping to promote competition and innovation in the delivery industry.
* To help transform the Egyptian market into a more competitive space. Hatley aims to help transform the Egyptian market into a more competitive space by providing a platform for delivery personnel to compete for orders based on price and time. By promoting competition and innovation in the delivery industry, Hatley can help to drive down delivery fees and improve service quality, ultimately benefiting both delivery personnel and users.
* To increase revenue for smaller stores. Hatley aims to increase revenue for smaller stores by connecting them with users who need delivery services. It allows to compete with larger stores, Hatley can help to level the playing field and promote competition.

# **2. Software Analysis**

## **2.1 Introduction:**

### 2.1.1 Problem Analysis and Motivation:

The Problem Analysis and Motivation phases involve understanding the challenges faced by consumers in purchasing daily essentials, the limitations of small and medium-sized stores, and the potential benefits of a delivery service.

**Problem Analysis (PA):**

The Problem Analysis phase involves understanding the problem domain, identifying the requirements, and determining the constraints that the solution must satisfy. This phase involves gathering and analyzing information about the problem, such as user requirements, constraints, and context. It also involves identifying potential solutions and evaluating their feasibility and effectiveness.

Traffic congestion: The country's notorious traffic congestion can make it difficult and time-consuming for people to purchase daily essentials. This can lead to a lack of access to a wide range of products and services.

Limited parking spaces: Limited parking spaces can make it difficult for delivery personnel to access and park their vehicles. This can impact the efficiency of delivery services.

Remote or underserved areas: Some areas may be underserved or remote, making it difficult for delivery personnel to access these areas and deliver orders.

Small and medium-sized stores: Small and medium-sized stores often struggle to compete with larger stores, who have the resources and scale to offer lower prices and a wider range of products.

**Motivation:**

The motivation behind Hatley is to provide a fast, convenient, and transparent delivery service that connects users with a wide range of products and services. By allowing delivery personnel to compete for orders based on price and time, the platform can help to incentivize faster and more efficient delivery services, while also providing users with greater choice and flexibility. Additionally, by providing job opportunities for young individuals and helping to transform the Egyptian market into a more competitive space, the platform can help to increase revenue and product availability for smaller stores, while also providing consumers with greater access to essential goods and services.

### 2.1.2 Scope of the Project:

Delivery of daily essentials: Hatley provides a comprehensive delivery solution for daily essentials, covering a wide range of products and services. This includes groceries, household items, electronics, and more. Hatley can help to simplify the delivery process and save users time and effort.

Competition-based delivery service: Hatley allows delivery personnel to compete for orders by either accepting the offer or presenting a competitive alternative in terms of time and cost. This encourages competition in the delivery, ultimately benefiting both users and delivery personnel.

Real-time tracking: Hatley enables users to track the arrival of their request, ensuring transparency and accountability. By providing real-time updates on the status of the delivery, users can have greater peace of mind and confidence in the delivery process.

Job opportunities: Hatley creates job opportunities for young individuals and transforms the Egyptian market into a competitive space, enhancing product quality and increasing revenue for some stores without the need for advertisements. By providing a platform for delivery personnel to compete for orders, Hatley can help to promote competition in the delivery.

User-friendly platform: Hatley provides a user-friendly and accessible platform for users and delivery personnel. By designing a simple and intuitive user interface, Hatley can help to ensure that users and delivery personnel can easily navigate the platform and access the information they need.

Expansion: Hatley expands the service to new markets and regions, both domestically and internationally. By expanding into new markets, Hatley can help to increase revenue, and provide job opportunities for delivery personnel in new regions.

Overall, the scope of Hatley includes providing a comprehensive delivery solution, promoting competition, ensuring transparency and accountability, creating job opportunities, promoting sustainable and environmentally-friendly practices, ensuring safety and security, providing a user-friendly platform, and expanding to new markets and regions. By focusing on these areas, Hatley can help to meet the needs of its users and transform the Egyptian market into a more competitive space.

### 2.1.3 Target User Group:

Hatley targets the following user Group:

* Busy professionals: Hatley is designed for busy professionals who do not have the time to go to multiple stores to purchase their daily essentials. By providing a comprehensive delivery solution, Hatley can help these individuals save time and effort, allowing them to focus on other important tasks.
* Individuals with mobility issues: Hatley is also suitable for individuals with mobility issues who may have difficulty traveling to physical stores. By providing delivery services, Hatley can help these individuals access a wide range of products and services from the comfort of their own homes.
* Small and medium-sized businesses: Hatley can also be useful for small and medium-sized businesses that require regular deliveries of supplies and equipment. By allowing delivery personnel to compete for these orders, Hatley can help these businesses save time and money, ultimately increasing their revenue and competitiveness.
* Individuals in remote or underserved areas: Hatley can also be a valuable resource for individuals in remote or underserved areas who may have limited access to physical stores. By connecting users with delivery personnel who can purchase and deliver items from any nearby or distant store, Hatley can help to ensure that these individuals have access to a wide range of products and services.

Overall, the target user group for Hatley is broad and diverse, encompassing busy professionals, individuals with mobility issues, small and medium-sized businesses, environmentally-conscious consumers, and individuals in remote or underserved areas. By providing a comprehensive delivery solution that is fast, efficient, and user-friendly, Hatley can meet the needs of these users and help to transform the Egyptian market into a more competitive space.

## **2.2 System Requirement:**

### 2.2.1 Project Sponser:

The Egyptian Ministry of Higher Education in general and the Faculty of Computers and Information (Information System), Assiut University in particular finance the project and financial support and work to meet any needs of the project from the beginning of the project until its completion.

### 2.2.2 Business Need:

In studying the needs of the Egyptian market and the delivery services provided, we identified customer demand for a delivery service that assists them in obtaining their orders without the need to leave their premises, saving time, effort, transportation costs, and avoiding congestion. The problems with existing services include:

1. Delayed Responses: Delivery companies in the Egyptian market, specifically in Assiut Governorate, often take up to 45 minutes to confirm an order, followed by additional time for delivery.
2. Price Increase: Some companies exploit delivery personnel by taking a large percentage, up to 40%, of their earnings.
3. Limited App Coverage: Certain areas, especially restaurants, are not adequately covered by existing delivery apps.
4. Busy Families: Heads of households, occupied with work, require delivery for simple household items such as vegetables, fruits, school supplies, and other family needs.
5. Uncertainty in Delivery: Customers lack real-time information on the delivery time and location.
6. Address Ambiguity: Delivery personnel sometimes struggle to reach the exact address due to a lack of automatic location detection.

In response, we propose the development of a new application allowing users to order a variety of products by writing their request and setting a preliminary service price and time. Delivery personnel can then compete for the order by accepting the offer or providing a competitive alternative in terms of time and cost. Notifications are sent to the customer, allowing them to choose the best delivery option based on their preferences. This approach not only opens up job opportunities for the youth but also enhances market competition, improving product quality and increasing the revenue of local businesses without the need for extensive advertising.

Features:

For users, the application offers:

1. Ordering from anywhere with delivery to any location.
2. Order tracking through real-time location updates.
3. Rating and commenting on delivery services after order completion.
4. The freedom to choose the preferred delivery option based on cost and time.

Features for Delivery Personnel:

1. **Smart Order Matching:**
   * Delivery personnel can choose suitable orders using GPS features that display all nearby orders.
   * Receive notifications when customers post new requests, allowing them to promptly respond.
2. **Competitive Bidding:**
   * Delivery personnel can submit alternative offers to customers who have posted orders, competing based on time and cost.
   * Enhances the chances of securing orders and encourages efficient service.
3. **Efficient Address Navigation:**
   * Simplified access to precise delivery locations through automatic or manual customer location detection.
   * Facilitates accurate and timely deliveries, improving overall customer satisfaction.
4. **Increased Earnings Percentage:**
   * Application allows a higher percentage of the delivery cost to go to the delivery personnel, motivating them to use the app rather than traditional services with high commission rates.
   * Encourages fair compensation for the delivery efforts.
5. **Customer Rating and Feedback:**
   * Delivery personnel can be rated by customers, fostering a trustworthy and transparent system.
   * Customers can leave comments, building a reputation for reliable and efficient delivery personnel.
6. **Flexible Work Zone:**
   * Delivery personnel can set their preferred work area and adjust it when transitioning between different locations or governorates.
   * Provides flexibility and convenience for delivery personnel based on their current location.

### 2.2.3 Business Value:

In our current era, digital services have become indispensable. As the individual's responsibilities increase and the speed of life increases, we thought of providing a home delivery service for all services that an individual may need, in order to save time and effort while ensuring quality service and good treatment. We conducted a survey to see the extent of the project idea’s resonance and quality among the audience, and it was clear that the community needed such a service, as many survey participants explained that long distance and crowding are among the main factors that increase their desire to provide such a service. In addition, the quality of the service and its cost are among the main factors that encourage their use of it and this is what our project aims to achieve and its greatest challenge, as it works to provide the user’s order in the fastest time at the lowest cost and best quality also it provides knowledge of the status of your order and tracking it. In addition, the lack of similar competitors in such a service also makes the idea more desirable. Therefore, it is very expected that there will be a strong demand and response from the community for the service provided

## **2.3 Fesability Study:**

### 2.3.1 Technical Feasibility:

1. familiarity with web application is medium risk:

* The team has some good experience with the web application design and functions.
* The team has low experience with the frameworks and how real application work in the real world.
* there is some new features is still under studying by the team.

1. familiarity with the delivery System is medium risk:

* some of the team has good experience of how the mechanism of the delivery application work and how user and the delivery will contact with each other.
* some of team has good user experience to make a good design and UX to make the app easy to use and fast and have good UI design.

1. size of project is considered low risk:

* in first version of the application; the number of team is good to do this app
* the project time is long so there is low risk in the deadline of the app
* Up till now there is no app in competitive with us.

1. compatibility with the delivery app is good:

* because it’s a new system so we have to get new server and it must be fast and strong to handle the number of users of the app fast.
* in this version the marketing will be good and will scale by time to deliver the application every were

### 2.3.2 Economica Feasibility:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 2023 | 2024 | 2025 | 2026 | Total |
| Benefits |  |  |  |  |  |
| Number of views to the site |  | 5,000 | 15,000 | 30,000 | 50,000 |
| Percentage from each order |  | 0 | 120 | 1,825 | 1,945 |
| Investments |  | 0 | 50,000 | 150,000 | 200,000 |
| Total Benefits |  | 5,000 | 65,120 | 181,825 | 251,945 |
| *PV Total Benefits* |  | 4,319 | 56,253 | 157,067 | 217,640 |
| Development costs |  |  |  |  |  |
| Domain | 500 | 0 | 0 | 0 | 500 |
| Virtual server | 20,000 | 0 | 0 | 0 | 20,000 |
| *Total development costs* | **20,500** | 0 | 0 | 0 | **20,500** |
| Operational costs |  |  |  |  |  |
| Private Server |  | 20,000 | 20,000 | 20,000 | 60,000 |
| Office Space Workplace |  |  | 24,000 | 24,000 | 48,000 |
| Marketing |  | 20,000 | 30,000 | 50,000 | 100,000 |
| *Total operational costs* |  | **40,000** | **74,000** | **94,000** | **208,000** |
| *Total costs* | **20,500** | **40,000** | **74,000** | **94,000** | **228,500** |
| *Present value total costs* | 17,709 | 34,554 | 63,924 | 81,201 | 197,387 |
| *Total benefits – Total costs* | -17,709 | -30,234 | -7,671 | 75,867 | 20,253 |
| *Cumulative Net cash Flow* | -17,709 | -47,943 | -55,614 | 20,253 |  |
| *Return on investment* | 10.26% |  |  |  |  |
| *Break-even point* | 2.733 Years |  |  |  |  |

### 2.3.3 Organizational Feasibility:

* **Champion**: The goal of the Hatley Project is to provide a distinctive delivery service that provides job opportunities for young people away from delivery companies exploiting them, facilitating the service for users so that it saves them the time and cost of ordering, the time required to complete the project is one year.
* **organizational** **Management**: we will support this project with budget **228,500**EGP (Editing in progress).
* **system users:** their role is to make important decisions and they are asked for feedback to improve system.

## **2.4 The Survey:**

**Survey:** We have conducted a survey to gauge the market's response to the idea of a delivery application. This survey has been instrumental in understanding people's needs, existing problems, customer requirements, satisfaction levels, and the anticipated success of the project. The survey results can be summarized as follows:

A total of 58 individuals responded, with 55.2% being female and 44.8% male.

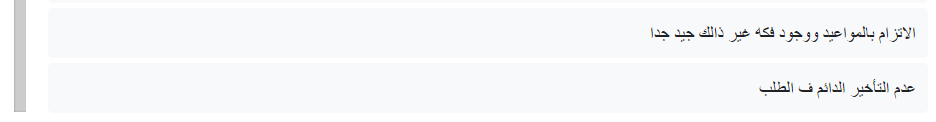
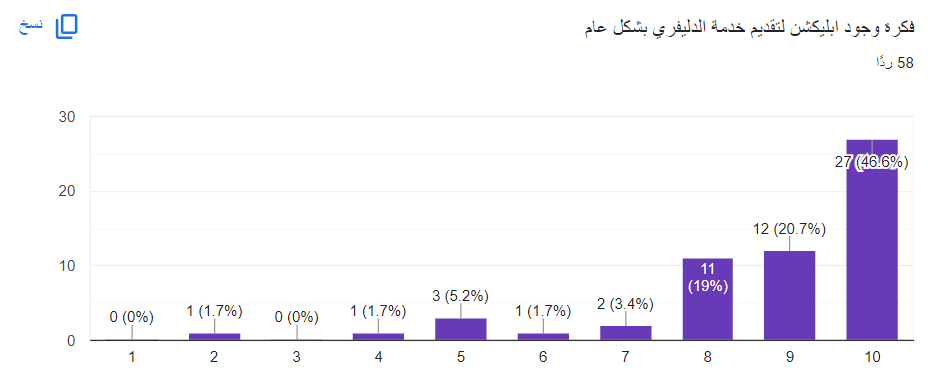
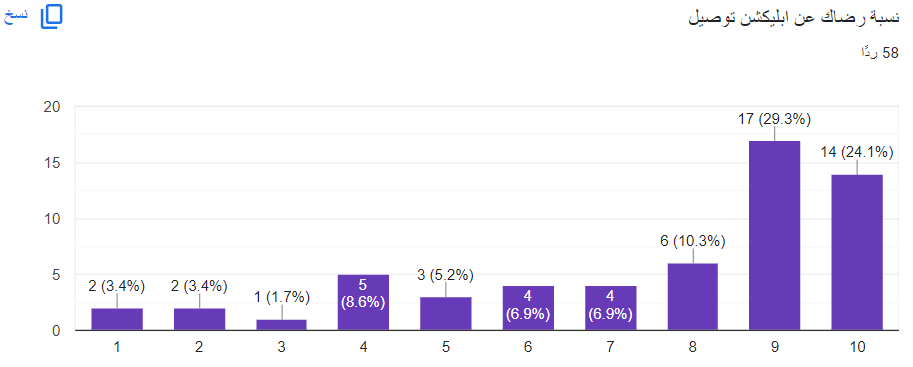
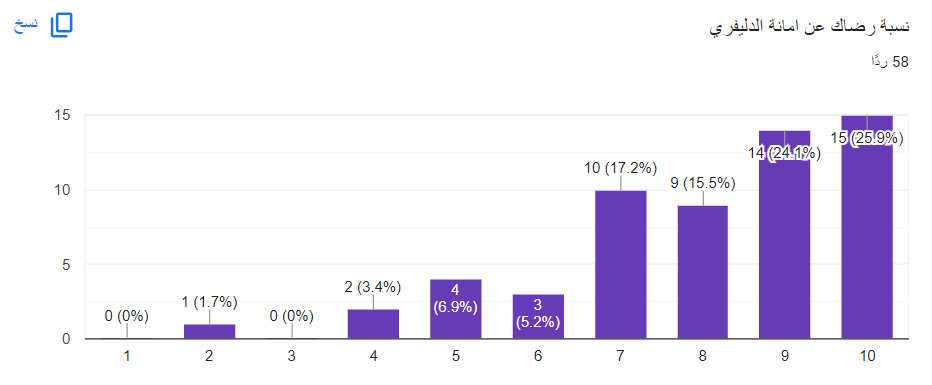
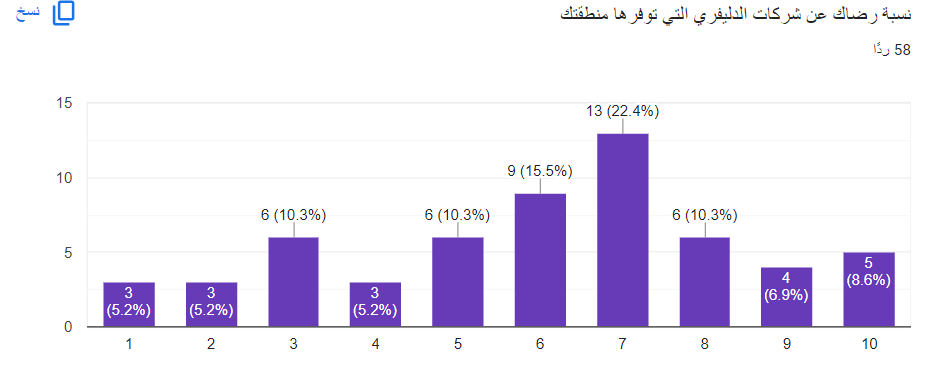
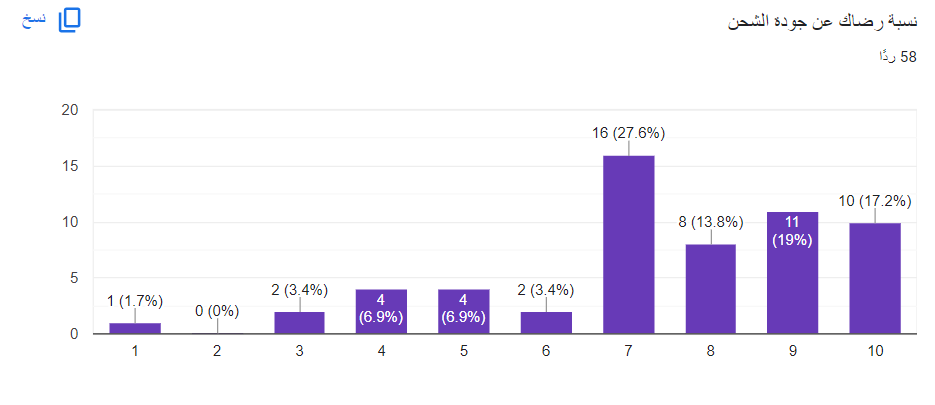
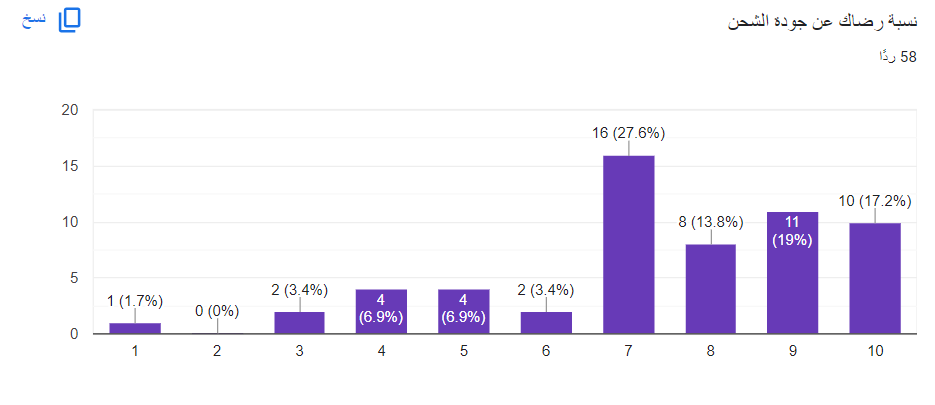
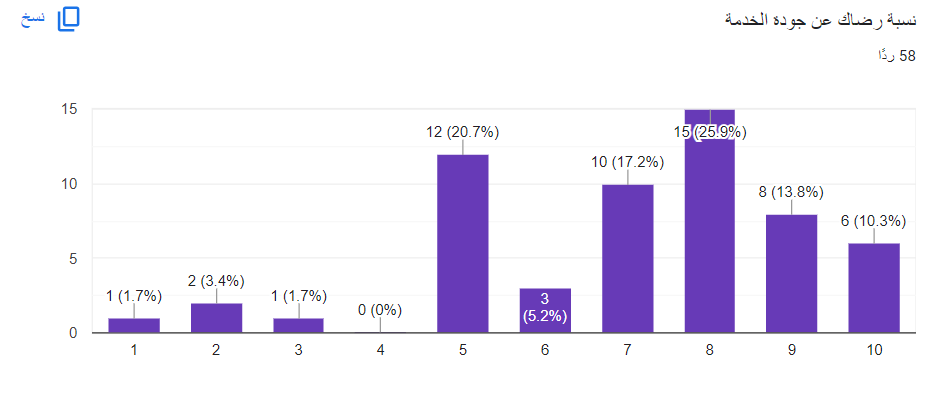
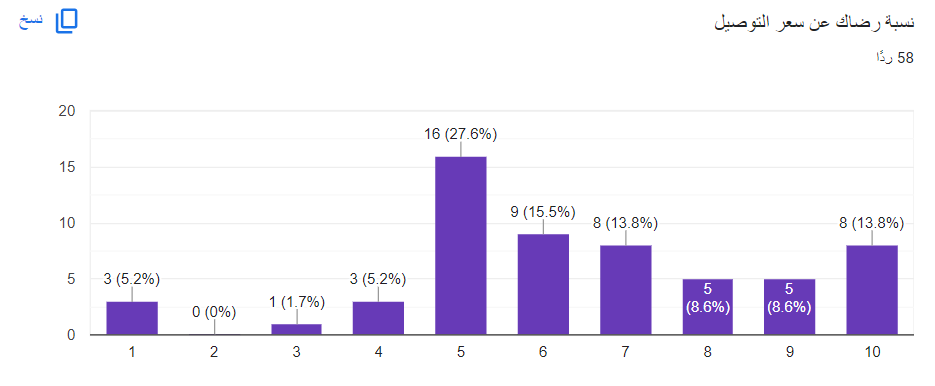
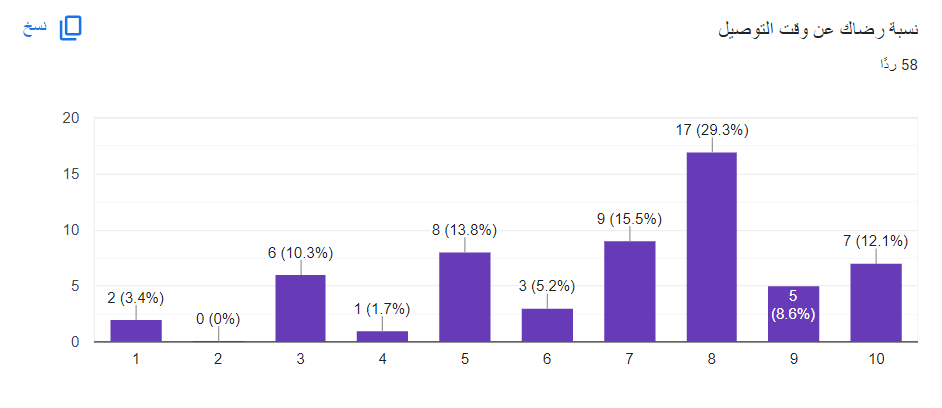
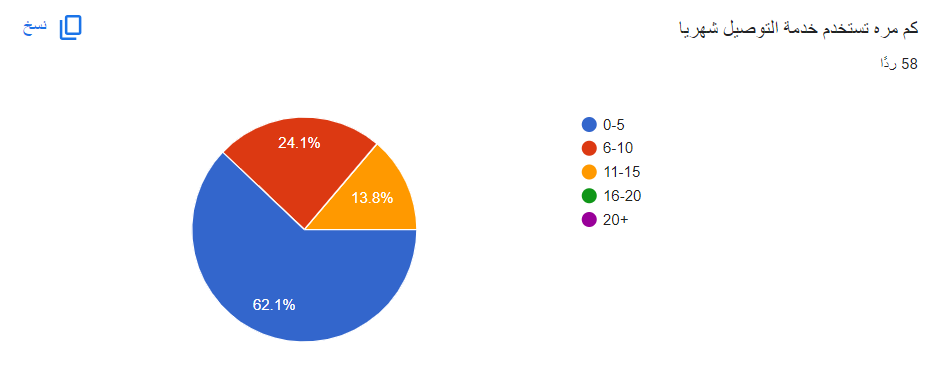
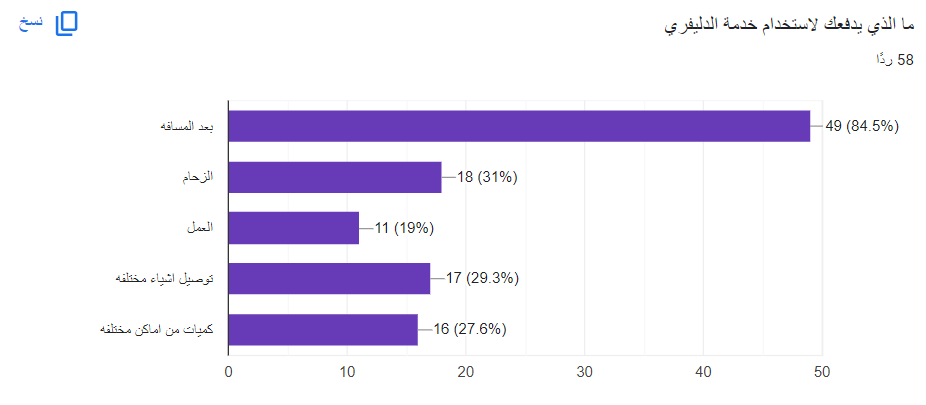
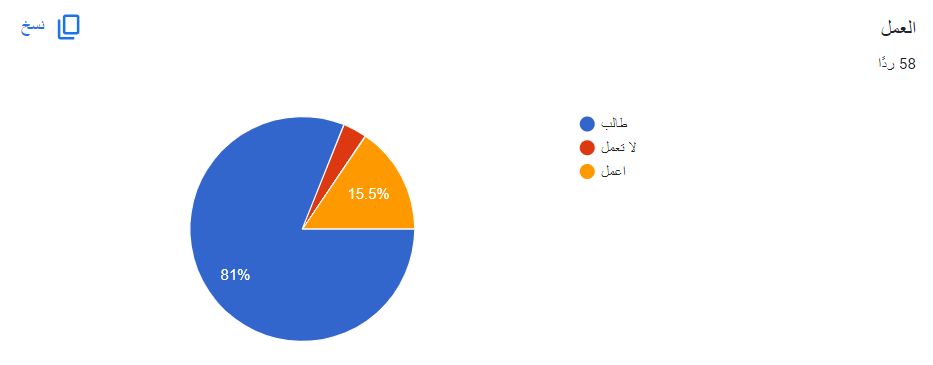
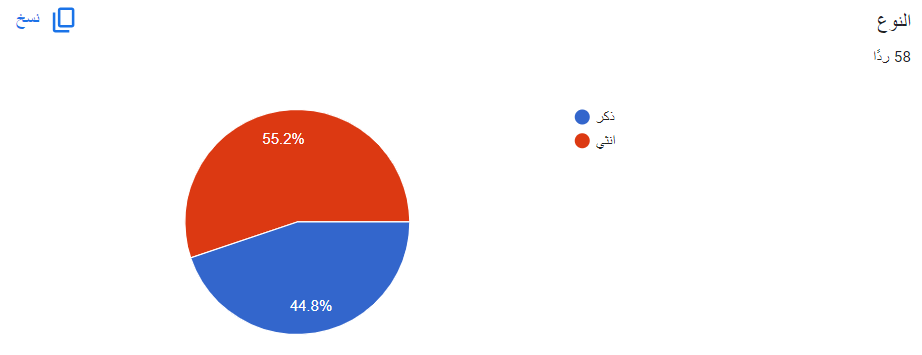
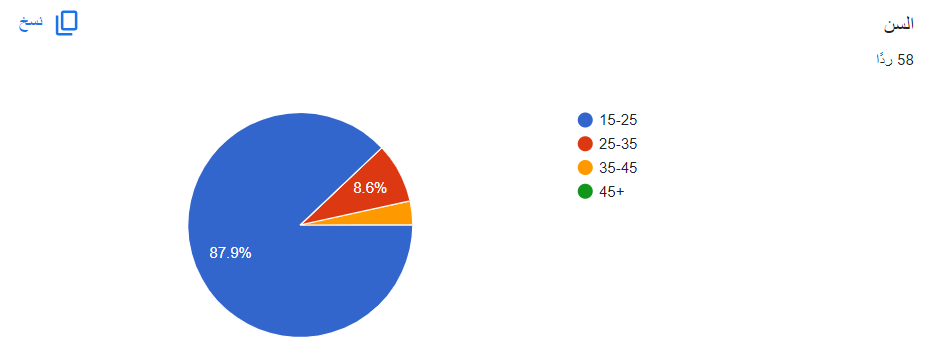
Reasons for using delivery services included distance, traffic, work commitments, delivery of various items, and quantities from different locations.

The satisfaction rate regarding pricing received an average score of 5 out of 10, with 27.6% giving this rating.

Customers expressed an 86% satisfaction rate with the concept of a delivery application.

Additional feedback included suggestions for reducing delivery fees, implementing a rating system for delivery personnel, managing order confusion, offering a reliable GPS tracking feature, addressing delivery time concerns, and ensuring the application is not tied to specific products, allowing flexibility in ordering.

This valuable feedback will guide our project development, addressing specific concerns and enhancing the overall customer experience.

****

## **2.5 Definition of Requirements:**

### 2.5.1 Functional Requirements:

|  |  |
| --- | --- |
| * **Functional requirements** | |
| **User Registration and Authentication** | * Users should be able to register, create accounts, and log in securely. * Implement password recovery and account verification processes. |
| **Order Creation** | * Allow users to create new orders. * Create a simple request form with fields for entering basic information such as the name of the desired item, the quantity needed, available budget, specific specifications for the requested item, and any other relevant information. * We consider additional options, such as attaching images or files to provide more information about the requested item. |
| **Location and Address Selection** | * Provide an interface for users to input or select their delivery location and address. * Include options for saving multiple delivery addresses. |
| **Order Confirmation and Payment** | * Confirm the order details, including the total cost, and prompt users for payment. * Support various payment methods, such as credit cards, digital wallets, or cash on delivery. * The user needs a “Submit Request” button to send the information they have entered. |
| **Display Requests and Interaction** | * Display user requests to service providers who can respond to them. * Providing an interface for service providers to interact with requests by presenting their offers. * Enable service providers to submit their offers and specify prices and terms. |
| **User Selection** **of Preferred Offer** | * Grant users the ability to choose from among the service providers based on the offers and terms presented. * Add a "Select" or "Accept" button for users to choose from among the offers. * Update the status of the request and notify the user of their selection. |
| **Request Processing and Assignment** | * Implement a system to process requests and assign them to selected service providers. * Provide options for service providers to interact with posts by using "Accept" or "Offer" buttons. * Develop a dashboard for managing offers submitted by service providers, including specifying prices and terms. |
| **Notifications and Alerts** | * Implement a real-time notification system to inform users of offer details and request status. * Send notifications to both users and service providers regarding request status. |
| **Request Tracking and Delivery** | * Allow users to track the status of their requests and receive updates on the delivery process. * Send automatic updates on the provider's location and estimated arrival time. |
| **User Ratings and Comments** | * Enable users to provide ratings and comments after the service or delivery is completed. * Display the average ratings for users. |
| **Customer Support** | * Develop a customer support system that allows users and service providers to communicate and resolve issues. |
| **Admin Dashboard** | * Create an administrative control panel that allows administrators to manage posts, offers, and monitor operations. |

### 2.5.2 Non-Functional Requirements:

|  |  |
| --- | --- |
| * **Non-functional requirements** | |
| **Security** | * The website must implement high levels of security to protect sensitive information and prevent security breaches and protect user data from unauthorized access. * User authentication and identity verification mechanisms must be secure. |
| **performance** | * + The website should load quickly and provide good performance to ensure a smooth user experience,   even with a large number of users.   * + Performance can be improved by reducing loading times, optimizing images, and utilizing caching.   + The system supports a large number of concurrent users at all other times. |
| **Availability** | * The website must be available 24/7/365 and maintain a high level of uptime. * Contingency plans and disaster recovery strategies should be in place in case of website failure. |
| **Cross-Browser Compatibility** | * The website should be designed to work consistently across different common web browsers (Chrome, Firefox, Safari, etc.). |
| **Responsive Design** | * The website should be responsive, adapting to various devices and screen sizes. |
| **Ease of Maintenance** | * Code should be well-organized and maintainable to facilitate future maintenance and development. |
| **Scalability** | * The website must be continuously scalable to accommodate the increase in the number of users and the volume of data. |
| **Database Performance** | * The database should be well-structured and optimized for fast response times. |
| **Documentation** | * The project and codebase should be well-documented to ease future handling by development teams. |
| **Data Security** | * Personal data and sensitive information must be protected with strict information security measures. |
| **Continuous Improvement** | * There should be a plan for ongoing management and implementation of improvements to the website. |
| **Compliance with Standards** | * The website must comply with international web standards such as W3C and WCAG for accessibility. |
| **Usability** | * The system must be easy to use and understand. |

## **2.6 Use Cases:**

### 2.6.1 Register Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** Register | | **ID: UC\_1** | **Level:** Primary |
| **Actor** | New User or New Delivery | | |
| Description | New User: Wants to create an account to use the delivery app for ordering goods. New Delivery Person: Wants to create an account to become a delivery service provider | | |
| **Preconditions** | * The user has Navigated to the register page. * The user or delivery person has not registered an account previously. * The website is accessible. | | |
| **Main Success Scenario** | 1. The new user opens the website and selects the "Register" option. 2. The system presents a registration form with fields for personal information, such as name, email, phone number, a password, Photo of ID card and Photo of the user or delivery with ID card. 3. The new user or delivery person enters their information into the form. 4. The new user or delivery person selects the type of account they want to create (either "User" or "Delivery Person"). 5. The system validates the entered information:    * If the information is valid, the system generates a unique ID for the user or delivery person.    * If any information is missing or invalid, the system displays an error message and allows the user to correct it. 6. The new user or delivery person confirms their registration by submitting the form. 7. The system stores the user's or delivery person's information, including their unique ID, in the database. | | |
| **Postconditions** | * If successful, a new user or delivery person account is created with a unique identifier (ID). * The new user or delivery person can log in with their newly created account. * The user's or delivery person's information is stored securely in the system's database | | |
| Exceptions | * **Existing User/Delivery Person (Step 1, Alternative Flow):** If a user or delivery person with the same email or phone number already exists, the system should prompt the user to log in or initiate a password reset if they forgot their password. * **Validation Errors (Step 5, Alternative Flow):** If the system encounters errors during the validation of the registration information (e.g., invalid email format, password requirements not met), it should display error messages and allow the user to correct the information. | | |

### 2.6.2 Login Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case name: Login | | ID: UC\_2 | Priority: |
| Actor | * Customer * Delivery Driver * System | | |
| Description | This use case describes the login mechanism for the user or the delivery or the admin | | |
| Preconditions | The customer must be registered. | | |
| Main Success Scenario (Main Flow) | 1-The user navigates to the login page  2-the system make the user choose between delivery or user or admin  3-The system presents the user with a login form, including fields for username/email and password  4-the user enter their valid username and password  5- the user submit the form by clicking on Login button   * If the credentials are valid, the system grants access to the user and proceeds to the main application. * If the credentials are invalid, the system displays an error message (e.g., "Invalid username or password") and returns to the login form | | |
| Postconditions | The system will navigate the user automatic to the home page  or the last page the user was in it and the page ask him to login before complete | | |
| Alternative Flow | the user make login with account blocked by system. the system will display a message that this account is blocked for bad behavior | | |
| Exceptions | 1-the user login with user account in the delivery account or the opposites 2- the user forget the password 3-invaild data for many times must be locked for five minutes or more cooldown | | |

### 2.6.3 Display requests and Interaction Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Display requests and Interaction | | ID:UC\_3 | Priority: |
| Description | This scenario enables service providers (delivery men) to accept or interact with orders, they also can submit their offers, price and time to customer. | | |
| Actor | * Service provider (Delivery driver) | | |
| Precondition | * Delivery should have active account and logged in to the site * There are active orders posted by users (customer) not be done yet. | | |
| Normal Course | 1. Delivery man open home page. 2. From his dashboard, delivery man can view a list of available requests and its details. 3. Delivery man can select a specific delivery request. 4. Delivery can accept the request by pressing on accept button. | | |
| Alternative Course | 1. Delivery man open home page or notification. 2. Choose a specific order. 3. View its details. 4. Press on “another offer”. 5. New page appear that contain time and price fields. 6. Delivery fills these fields. 7. Press “send” or “confirm” button. | | |
| Postcondition | * Notification contains all offers will be sent to the customer that contain delivery name, rate, picture, price and time. * Customer reviews the offers and compare them each offer has two button accept and reject. * The customer has the authority to accept any offer then when he accept:  1. A notification must be sent to the service provider that the customer accept his offer. 2. The order is being converted or marked as unavailable now.  * The customer and the chosen delivery must be linked together. * If none of the service providers offer for an order for long time, the user can choose to resubmit the order or explore other options | | |
| Exception | * When he offers another by filling time and price and clicks Confirm or Send, it has be agreed by someone else or converted to not available now. | | |

### 2.6.4 Create Order Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Create Order | | ID: UC\_4 | Priority: High |
| Description | This use case describes how the user will make an order. | | |
| Actor | User | | |
| Precondition | * The user opens the system. * The user must log in. | | |
| Normal Course | 1. The user clicks "make an order" button. 2. The system opens the form to the user to fill it out. 3. The user writes the name of items he\she needs. 4. The user writes the quantity he\she needs. 5. The user specifies the available budget he\she can pay. 6. The user can attach a photo or file if he\she wants(optional). 7. The user clicks "Done" button. | | |
| Postcondition | The system will redirect the user to location selection interface. The system will redirect the user to location selection interface. | | |
| Exception | * If the user wrote a strange name for the item he\she needs, an error message will appear to tell him\her to explain what he\ she needs. * If the user wrote a MINS quantity (or any not reasonable amount) for the item he\she needs, an error message will appear. * If the budget were zero, an error message will appear. | | |

### 2.6.5 Accept offer Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Accept offer | | ID: UC\_5 | Priority: High |
| Description | This use case describes how the user will choose from the offers he\she has received. | | |
| Actor | * User | | |
| Precondition | * The user opens the system. * The user must log in. * The user makes an order. * The user determines his\her location. * The user determines payment way.   The user receives offers. | | |
| Normal Course | * 1. The user reviews the offers he\she has received.   2. The user clicks "Accept" button of the offer he\she preferred. | | |
| Postcondition | 1. A success message will appear for the user. 2. The chat will be opened. | | |
| Exception | If the user did not find a suitable offer, he\she can cancel the order. | | |

### 2.6.6 Introduce offer Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name’: Introduce offer | | ID: UC\_6 | Priority: High |
| Description | This use case describes how the service provider will provide an offer. | | |
| Actor | * Service provider (Delivery driver) | | |
| Precondition | * The service provider opens the system. * The service provider must log in. | | |
| Normal Course | * 1. The service provider receives a notification of the order   2. The service provider reviews the orders.   3. The service provider provides an offer for the order he\she wants.   4. The service provider determines the price and duration for the order. | | |
| Postcondition | A notification will be received for the service provider if the user accepted his\ her offer. | | |
| Exception | The order has just started. | | |

### 2.6.7 Notification and Alert Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE NAME: Notification and Alert | | ID: UC\_7 | Priority: medium |
| Actor | System | | |
| Description | This use case describes the notification mechanism of how to notify the user or the delivery man of the request status for the user and new order for delivery. | | |
| Preconditions | * The customer or delivery must be registered and logged into their account. | | |
| Main Success Scenario (Main Flow) | 1. For delivery:  * If any user post an order that match with the delivery location, the system send notification to the delivery that there is new post and brief of the post . * If the delivery sends an offer to a user and the user accept or reject the offer, the system should send notification with the answer of the use. * The messages between the user and delivery have also notification with every message  1. For User:  * If the user post order and start to receive offers from deliveries each one have single notification * The status of the order has notification the describe to the user where its order now * The messages between the user and the delivery also have notification | | |
| Postconditions | Any of them if clicked must open the website on the page that pushed the notification | | |
| Exceptions | No Exception | | |

### 2.6.8 Customer Support Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Customer Support | | ID:UC\_8 | Priority: |
| Description | This scenario enables users and service providers to communicate and resolve problems to the delivery service. | | |
| Actor | * User (Customer) * Service provider (Delivery driver)   Customer support representative. | | |
| Precondition | * The user or service provider is logged into their own accounts. * The user or service provider has encountered a problem or has a question related to system services. | | |
| Normal Course | * Press on “Support System” button. * Choose type of problem with delivery men or with application.  1. If he choose with delivery men (customer) :  * All the delivery men (customers) that the customer (delivery man) deal with must be appear as a list on a new page. * Choose one of them and press on Reporting button * New page should be opened must have several choices and empty box to write details of problem. * Choose type of problem, write details and send some proofs. * Press on “confirm” button.  1. If he chooses with application:  * New page should be opened must have list of some expected problems and its solutions take in consideration he might choose another problem in an empty box. | | |
| Alternative Course | 1. User clicked on “last deals” button. 2. Choose the order that has a problem. 3. Open the account of delivery man (customer) who he wants to complain him. 4. Press on “Reporting” button. 5. New page should be opened must have several choices and empty box to write details of problem. 6. Choose type of problem, write details and send proof if exist. 7. Press on “confirm” button. | | |
| Postcondition | 1. If he chooses specific question, new page appears with answer this question and solve the problem. 2. If he writes a report, it should be successfully stored in database related with the data of user. 3. The system send notification to Admin with the report. 4. An admin checks the report. 5. An admin replay to the user and when he solved this problem he marked as “resolve or done” | | |
| Exception | * No place in server to store report. * The delivery has just deleted his account. | | |

### 2.6.9 Driver Ratings for Customers Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE NAME: Driver Ratings for Customers | | ID: UC\_9 | Priority: Moderate |
| Actor | Delivery Driver | | |
| Description | This Use Case outlines the process of allowing delivery drivers to rate customers based on their interactions and experiences. | | |
| Preconditions | * The customer has successfully completed a delivery request. * The delivery driver has successfully completed a delivery. | | |
| Main Success Scenario (Main Flow) | 1. Driver completes the delivery:  * The delivery driver successfully completes the delivery to the customer's location.  1. Driver rates the customer:  * After completing the delivery, the delivery driver has the option to rate the customer based on their interactions and behavior during the delivery. * The driver provides a rating for the customer (e.g., on a scale of 1 to 5 stars) reflecting their experience.  1. Driver leaves optional comments:  * The delivery driver can also provide optional comments or feedback regarding their experience with the customer, including any specific positive or negative aspects.  1. System records the driver's rating and comments:  * The system records the rating and comments provided by the delivery driver, associating them with the specific delivery and customer. | | |
| Postconditions | * The delivery driver's rating and comments about the customer are stored in the system for reference. * The customer's profile may include the average rating given by drivers. | | |
| Alternative Flow | Driver chooses not to rate or comment:   * If the delivery driver decides not to provide a rating or comments about the customer. * The system records the delivery as completed without specific feedback. | | |
| Exceptions | 1. Inappropriate Comments:  * If the delivery driver leaves inappropriate or offensive comments about the customer. * The system should have a mechanism in place to flag and review such comments Inappropriate comments may be removed, and appropriate actions may be taken.  1. Pattern of Inappropriate Ratings:  * If a delivery driver consistently provides low ratings and negative comments about customers without justifiable reasons. * The system may trigger a review of the driver's behavior and may take corrective actions if necessary. | | |

### 2.6.10 Request Tracking and Delivery Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE NAME: Request Tracking and Delivery | | ID: UC\_10 | Priority: High |
| Actor | * Customer * Delivery Driver * System | | |
| Description | This use case describes the process of tracking and delivering customer orders, it focuses on the interactions between customers, delivery drivers and the system to ensure efficient and transparent order tracking and successful delivery. | | |
| Preconditions | * The customer must be registered and logged into their account. * Orders must be recorded in the system with the required information. | | |
| Main Success Scenario (Main Flow) | 1. Customer places an order:  * The customer logs into their account. * The customer specifies the item he/she want to order. * The customer creates a delivery request post, specifying the item for delivery and delivery details. * The system records the delivery request and sets the initial status to "Request Submitted."  1. Delivery drivers provide offers:  * Delivery drivers who are available to take delivery requests view the posted requests. * Each interested driver submits an offer, including the delivery fee and estimated delivery time. * The customer receives notifications of new delivery offers.  1. Customer selects a delivery offer:  * The customer reviews the delivery offers provided by drivers. * The customer selects a preferred delivery offer based on the driver's fee, estimated delivery time and the service provider's rating. * The system assigns the chosen delivery driver to the request, and the status is updated to "Driver Assigned."  1. Delivery in progress:  * The assigned delivery driver follows the navigation provided by the system to  reach the pickup location if it exists. * The driver collects the specified item for delivery. * The system continuously tracks the driver's location in real-time. * The customer can track the delivery progress in real-time through their account.  1. Delivery driver completes the delivery:  * The driver arrives at the customer's destination and delivers the specified item. * The system updates the request status to "Delivered." * The customer receives an instant notification confirming the successful  delivery of the requested item. * If the customer had chosen to pay in cash, payment is collected from them after the delivery. | | |
| Postconditions | * The request status is updated at each stage of the delivery process. * Customers receive real-time updates on their request's status and delivery progress. * The system records completed deliveries for future reference. * The final state of the request is "Delivered" upon successful delivery. | | |
| Alternative Flow | 1. Customer cancels the request:  * If the customer decides to cancel the request before a driver is assigned, the request is canceled, and no delivery is made. | | |
| Exceptions | 1. Driver Unavailability Exception:  * If there are no available drivers to provide delivery offers when the customer submits a delivery request, the system must notify the customer of the unavailability of drivers at the moment. * The customer can try resubmitting the request later.  1. Driver Offer Rejection Exception:  * If the customer decides to reject all delivery offers provided by drivers, the system must reassign the request and notify the customer that no offers were accepted. * The customer can try again.  1. Driver Assignment Delay:  * If unexpected delays occur in assigning a driver to the request (e.g., due to connectivity issues or driver availability problems) * The system should periodically update the customer about the delay and provide an estimated time for driver assignment.  1. Delivery Delay:  * If the delivery is significantly delayed beyond the expected time due to unforeseen circumstances (e.g., heavy traffic or adverse weather conditions). * The customer or system should update the customer with the delay information and provide a new estimated arrival time. | | |

### 2.6.11 User Ratings and Comments Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| USE CASE NAME: User Ratings and Comments | | ID: UC\_11 | Priority: High |
| Actor | * Customer * System | | |
| Description | This Use Case outlines the process of allowing customers to rate and leave comments on their experiences with the delivery service. | | |
| Preconditions | * The customer has successfully completed a delivery request. * The delivery driver has successfully completed a delivery. | | |
| Main Success Scenario (Main Flow) | 1. Customer rates the delivery service:  * After the successful delivery of an order, the customer has the option to rate the delivery service provider on three criteria: speed of delivery, product quality, and cost-effectiveness. * The customer provides ratings for each of these criteria (e.g., on a scale of 1 to 5 stars).  1. Customer leaves optional comments:  * The customer can also provide optional comments or feedback regarding their delivery experience, including specific comments related to speed, product quality, or cost-effectiveness.  1. System records the ratings and comments:  * The system records the customer's ratings and comments, associating them with the specific delivery request and the involved delivery driver.  1. Driver views the ratings and comments:  * The delivery driver can view the ratings and comments left by the customer for the delivery service, including feedback on speed, product quality, and cost-effectiveness. | | |
| Postconditions | * The customer's ratings and comments, including feedback on speed, product quality, and cost-effectiveness, are stored in the system for reference. * The delivery driver can access and review the detailed feedback provided by the customer. | | |
| Alternative Flow | 1. Customer does not provide ratings or comments:  * If the customer chooses not to provide ratings or comments on speed, product quality, or cost-effectiveness, the system still records the delivery as completed but without specific feedback. | | |
| Exceptions | 1. Inappropriate Comments:  * If the customer leaves inappropriate or offensive comments, the system should have a mechanism in place to flag and review such comments. * Inappropriate comments may be removed, and appropriate actions may be taken. | | |

### 2.6.12 Location and Address Selection Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Location and Address Selection | | ID:UC\_12 | Priority: Primary |
| Description | This use case allows users to choose between two options for providing their delivery location and address: manual input or automatic detection. Users can also save multiple delivery addresses for future orders. | | |
| Actor | User (Customer) | | |
| Precondition | * User is logged in to their account on the Hatley App. * User has one or more saved delivery addresses (optional). | | |
| Normal Course | 1-User selects the option to place an order.  2-The system presents the user with two options for providing the delivery address:   * **Option 1: Input Address Manually**   + User selects this option to manually input their delivery address.   + They provide the following details:     - Street address     - City     - State/Province     - Country * **Option 2: Define Address Automatically**   + User selects this option to have the system automatically detect their delivery location based on their device's geolocation data or IP address.   + The system displays the automatically detected address.   3-User has the option to either:   * + **Select the Automatically Detected Address**:     - The user chooses to use the automatically detected address as the delivery address.     - They proceed to the next steps of the order placement process.   + **Change or Input a Different Address**:     - The user can choose to change the automatically detected address or input a new delivery address manually.     - If they select this option, they provide the following details:   4-User confirms the selected or newly inputted delivery address. | | |
| Postcondition | * The selected or newly inputted delivery address is associated with the user's current order. * The user can proceed with the order placement process. | | |
| Exception | If the user cancels the order or navigates away from the address selection screen:   * The address selection process is abandoned. * No address is associated with the order. * The user may return to this step later if they decide to continue with the order. | | |

### 2.6.13 Order Confirmation and Payment Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Order Confirmation and Payment | | ID:UC\_13 | Priority: Primary |
| Description | This use case covers the process of confirming the order details, including the total cost, and facilitating the payment process. It supports various payment methods such as credit cards, digital wallets, or cash on delivery. | | |
| Actor | User (Customer) | | |
| Precondition | * User has provided a delivery address. * User has chosen a preferred payment method. * User has reviewed the order details. | | |
| Normal Course |  | | |
| Postcondition | * The delivery order is confirmed. * Payment (if applicable) is successfully processed, and a payment confirmation is displayed. | | |
| Exception | 1-If the payment fails (for credit card or digital wallet), the system displays an error message, and the delivery order is not confirmed.  2-If the user cancels the delivery order or navigates away from the payment screen, the delivery order is not confirmed.  3-If there are issues with the selected payment method (e.g., insufficient funds for credit card payment), the system displays an error message and does not confirm the delivery order. | | |

### 2.6.14 Request Processing and Assignment Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Request Processing and Assignment | | ID:UC\_14 | Priority: Primary |
| Description | This use case describes the process of handling user requests, assigning them to selected service providers, and providing options for service providers to interact with requests using "Accept" or "Offer" buttons. It also includes the development of a dashboard for managing offers submitted by service providers, allowing them to specify prices and terms. | | |
| Actor | * User (Customer) * Service Provider | | |
| Precondition | 1. User (Customer) is logged in. 2. Service providers are registered in the system. | | |
| Normal Course | 1. User (Customer) submits a service request, providing details about the request and any specific requirements. 2. The system receives the service request and stores it as a pending request. 3. Service providers registered in the system are notified of the new request. 4. Service providers review the request and decide whether to:    * **Accept the Request**: A service provider clicks the "Accept" button if they are willing to take on the request. This action marks them as the assigned provider for the request.    * **Offer a Proposal**: Alternatively, a service provider can click the "Offer" button to submit a proposal with details such as price and terms. 5. If a service provider accepts the request, the system assigns the request to that provider. 6. If a service provider submits an offer, the system stores the proposal and notifies the user (Customer) of the offer. 7. The user (Customer) reviews the offers and either:    * **Accepts an Offer**: The user (Customer) selects a service provider's offer and marks them as the assigned provider.    * **Rejects Offers**: The user (Customer) can reject one or more offers. | | |
| Postcondition | * The service request is either assigned to a service provider who accepted it or is still pending if no provider was selected. * If an offer is accepted, the service request is associated with the selected service provider. * If offers are rejected, the request remains pending, and other service providers can still make offers. | | |
| Exception | 1. If there are no service providers available to handle the request, the request remains pending. 2. If no service provider accepts the request or submits an offer, the request remains pending. 3. If the user (Customer) cancels the request or navigates away from the request screen, the request remains pending. 4. If the service provider's offer is rejected, the system updates the request status but does not assign it to the rejecting provider. | | |

### 2.6.15 Admin control Use Case:

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Admin control | | ID:UC\_15 | Priority: |
| Description | This scenario explains how the administrator controls the system and this is through interacting with system, managing users, monitoring data and statistics, and making reports | | |
| Actor | Admin (Customer support representative.) | | |
| Precondition | Admin logged in using his own User name and password | | |
| Normal Course | * After correct login or authentication, the admin will be redirected to control panel (Admin Dashboard). * The admin dashboard displays the following:  1. Home page that contains charts of data as how many orders was made last 7 days, information about regions, percentage of increasing on new users (customer, delivery). 2. Emails and reports:   Admin can review all reports or answer all questions from users, first he open email if it is report about someone, the admin read it carefully then he opens the account of bad behavior user and check if this is true or no, if true he open all users search about him then delete or ban his account and send to him a notification with warning.   1. Members:    1. The admin can view the list of registered users (customer or delivery) and all users’ data like pic, phone, email, rate and how many orders he made...etc.    2. The admin can view and modify user account details, including their roles and permission by searching or clicking on any account he also can ban an account for any time with sending a warning notification. 2. Report and analyses:    1. The admin can create reports and analyses about the performance, and more statistics.    2. These reports can include user activities, reports on the service provider’s performance, and more statistic. 3. Forms And Table:   He can control and change the shape of any form, he can add, delete, edit and update any field of each table by clicking on forms and table 🡺edit 🡺 add field or delete. | | |
| Postcondition | * Saving data and changes that the administrator has just made to user, service providers account, or system settings. * If he made a report we can use these data to make decision to improve our system. | | |
| Exception | No exception. | | |

## **Management Plan:**

Hatley is a delivery platform that aims to provide a faster and more convenient way for users to purchase daily essentials in Egypt. With a focus on creating a competitive marketplace for delivery drivers, Hatley encourages them to provide faster and more cost-effective delivery options for users.

To ensure the long-term success of the platform, it is essential to have a well-thought-out management plan in place. This management plan outlines the key strategies and initiatives that will be implemented to drive growth, improve efficiency, and enhance the user experience for both users and delivery drivers.

In this section, we will outline the management plan for Hatley, including Current State of the Project, Phases, Gantt Chart & Timeline, Task Plan with Milestone, Division of Responsibilities Among Team Members and finally Risk Management.

### 2.7.1 Current State of the Project:

The current state of the project highlights the progress made so far, including the completion of documentation, UI/UX for website, UI/UX for mobile application. Additionally, preparations are underway for the website presentation.

### 2.7.2 Phases:

The management plan consists of a series of phases, each with specific tasks and objectives. The goal of each phase is to bring the project closer to completion and ensure that it meets the desired standards as shown in Figure 2.1. The phases include:

**Phase 1:** Project initiation and research. This including Defining the problem and project idea.

**Phase 2:** Planning and documentation. This including Studying the community need for the idea, creating a roadmap, and starting chapter 1 of software proposal.

**Phase 3:** Preparing and design creation. This including Imagining how the website will work, starting Chapter 2 of the software analysis documentation, and creating the UI/UX for mobile and website.

**Phase 4:** Prototype creation. This including Creating databases, finishing Chapters 3 and 4 of the software design documentation, and finishing the documentation.

**Phase 5:** Hosting prototype. This including Finishing the database and backend for the website, hosting the prototype, and preparing for the integration of the system.

**Phase 6:** Testing and integration. This including Testing and launching on real time, preparing the final web and app version, and preparing for the second presentation and discussion phase.

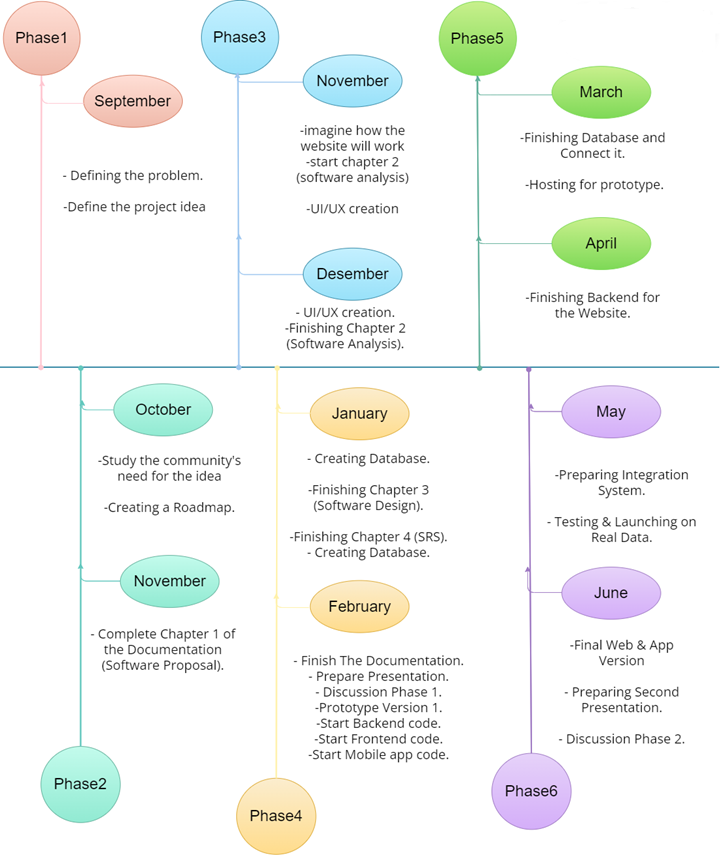


Figure 2.1: Phases and Timeline for Hatley Project.

### 2.7.3 Gantt Chart & Timeline:

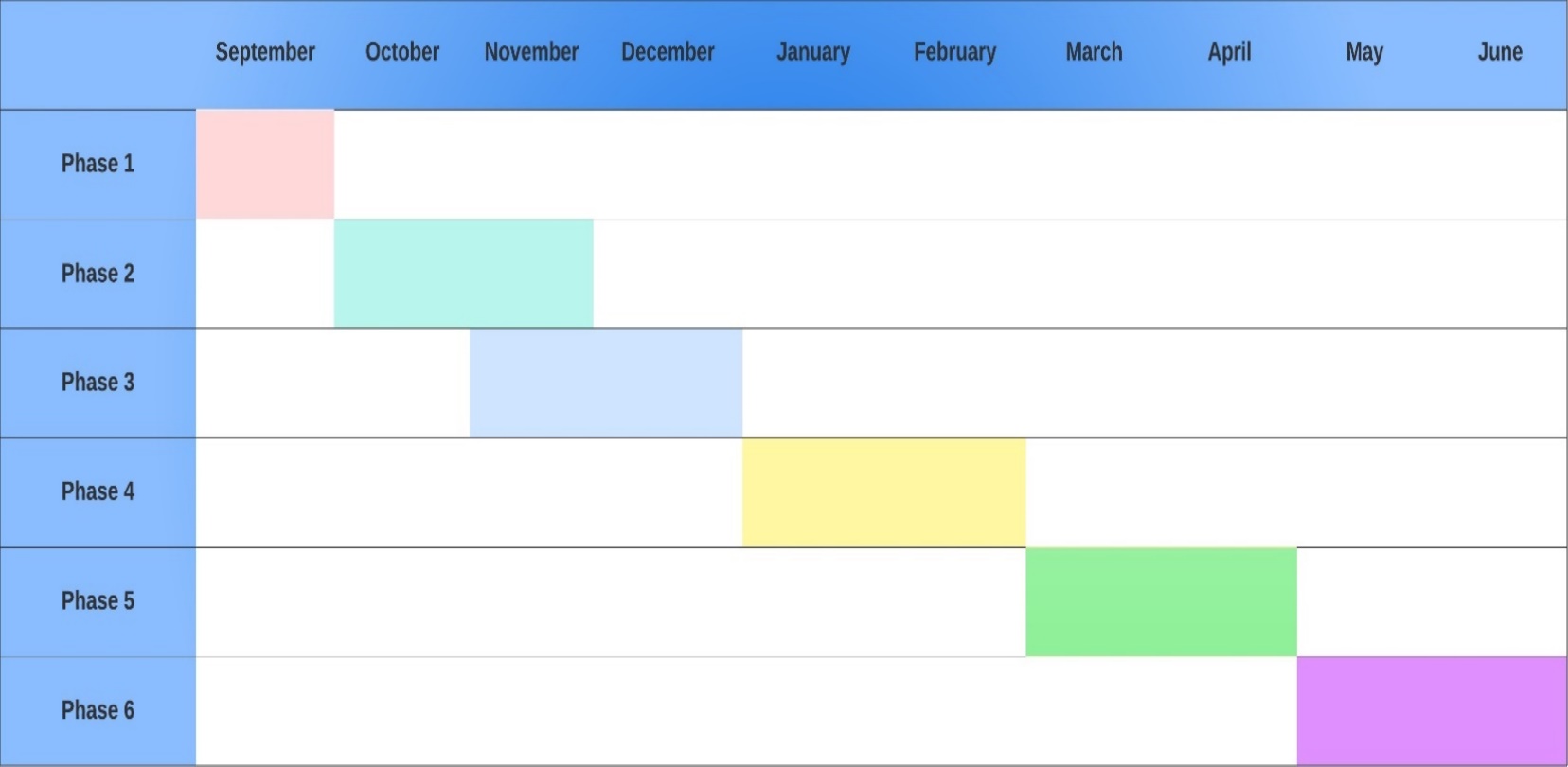


Figure 2.2: The Gantt Chart of General Management and Time Plan of Hatley Project.

### 2.7.4 Task Plan with Milestone:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Task Description | 2023 | | | | 2024 | | | | | | |
| SEP | OCT | NOV | DEC | JAN | FEB | MAR | ABR | MAY | JUN |
| 1 | Define the problem. | Checkmark with solid fill |  |  |  |  |  |  |  |  |  |
| 2 | Define the project idea. | Checkmark with solid fill |  |  |  |  |  |  |  |  |  |
| 3 | Study the community need for the idea. |  | Checkmark with solid fill |  |  |  |  |  |  |  |  |
| 4 | Creating a Roadmap. |  | Checkmark with solid fill |  |  |  |  |  |  |  |  |
| 5 | complete Chapter 1 of the Documentation (Software Proposal). |  |  | Checkmark with solid fill |  |  |  |  |  |  |  |
| 6 | Imagine How the website will work. |  |  | Checkmark with solid fill |  |  |  |  |  |  |  |
| 7 | Start Chapter 2 (Software Analysis). |  |  |  | Checkmark with solid fill |  |  |  |  |  |  |
| 8 | UI/UX creation. |  |  |  | Checkmark with solid fill |  |  |  |  |  |  |
| 9 | Finishing chapter 2 (software Analysis). |  |  |  | Checkmark with solid fill |  |  |  |  |  |  |
| 10 | Creating Databases. |  |  |  |  |  | Checkmark with solid fill |  |  |  |  |
| 11 | Finishing Chapter 3. (Software Design). |  |  |  |  | Checkmark with solid fill |  |  |  |  |  |
| 12 | Finishing Chapter 4 (SRS). |  |  |  |  | Checkmark with solid fill |  |  |  |  |  |
| 13 | finish the documentation. |  |  |  |  |  | Checkmark with solid fill |  |  |  |  |
| 14 | Prepare presentation. |  |  |  |  |  | Checkmark with solid fill |  |  |  |  |
| 15 | Discussion Phase1. |  |  |  |  |  | Checkmark with solid fill |  |  |  |  |
| 16 | Prototype version 1. |  |  |  |  |  | Checkmark with solid fill |  |  |  |  |
| 17 | Start coding. |  |  |  |  |  |  |  |  | Checkmark with solid fill |  |
| 18 | Finishing Database and connect it. |  |  |  |  |  |  | Checkmark with solid fill |  |  |  |
| 19 | Hosting for prototype |  |  |  |  |  |  | Checkmark with solid fill |  |  |  |
| 20 | Finishing backend for website. |  |  |  |  |  |  |  | Checkmark with solid fill |  |  |
| 21 | Preparing integration for system. |  |  |  |  |  |  |  |  | Checkmark with solid fill |  |
| 22 | Testing& launching on real time. |  |  |  |  |  |  |  |  | Checkmark with solid fill |  |
| 23 | Final web &app version. |  |  |  |  |  |  |  |  |  | Checkmark with solid fill |
| 24 | Preparing second presentation. |  |  |  |  |  |  |  |  |  | Checkmark with solid fill |
| 25 | Discussion phase 2. |  |  |  |  |  |  |  |  |  | Checkmark with solid fill |

Figure 2.3: Task Plan with Milestone.

### 2.7.5 Division of Responsibilities Among Team Members:

Division of Responsibilities Among Team Members is an important aspect of the Management Plan for the project. This helps to ensure that every team member understands their role and responsibilities, and it also helps to distribute the workload evenly as shown in Figure 2.4.

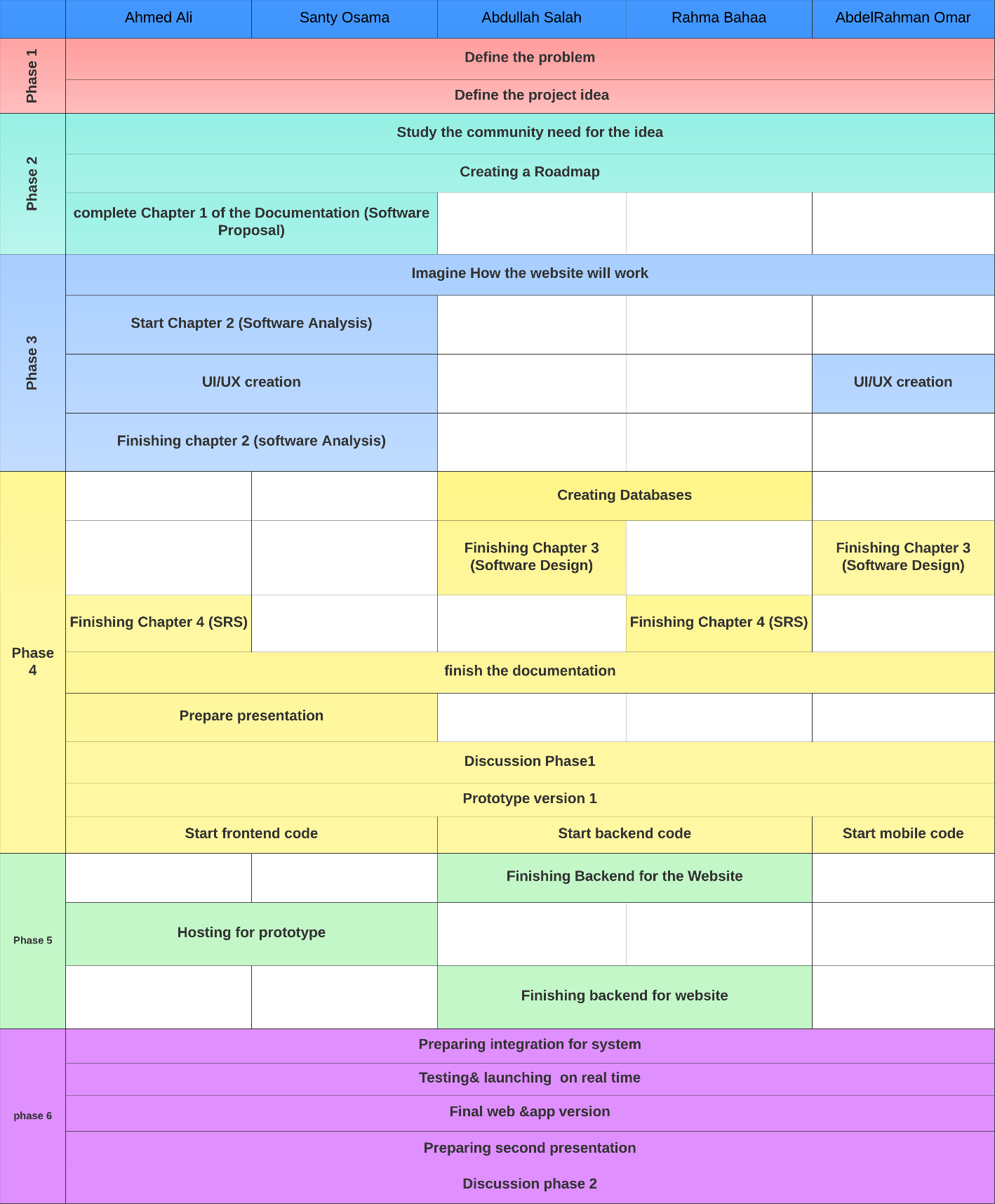


Figure 2.4: Table of Division of Responsibilities for Hatley Project Team.

Figure 2.5: Bar Char of Division of Responsibilities for Hatley Project Team.

### 2.7.6 Risk Management:

Moving forward with the Hatley management plan brings potential obstacles. By smart planning, evaluating risks, and using right strategies, we can navigate through these. Hatley might encounter risks like:

* + 1. **Challenging Market:** There's a chance our service doesn't hit the mark, or perhaps there's an oversupply of similar offerings, making distinction quite difficult.
    2. **Financial Hurdles:** Funding isn't always plentiful and unforeseen expenses might surface throughout the project's course.
    3. **Operational Obstacles:** Occasionally, plans derail and unpredicted hitches materialize.
    4. **Security Concerns:** Upholding the safety and privacy of user information is critical. But, it’s no small task.

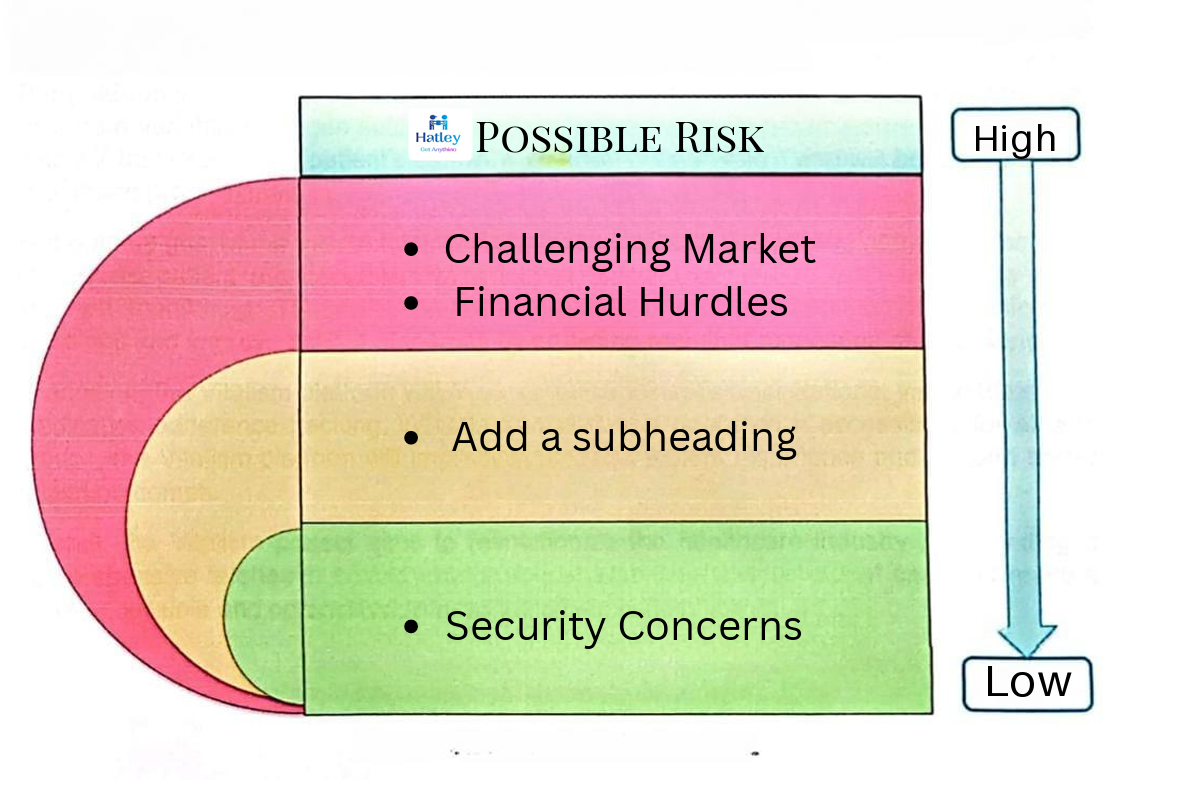


Figure 2.6: The Possible Risks of Hatley.

# 3.Software Design

### 3.1 Introduction

The software design phase for Hatley is crucial in creating a robust delivery service application. It involves defining the application's architecture, modules, and interfaces, considering functional and non-functional requirements. The design must prioritize user-friendliness, security, reliability, and performance while ensuring comprehensive functionality. It must align with stakeholder expectations and provide a clear depiction of the application. After the design, a thorough analysis of requirements traceability documents should show a correlation between design components and requirements, confirming all requirements have been addressed.

### 3.1.1 Project Background/Purpose

The purpose of the Hatley project is to develop an innovative delivery service application that offers convenience and efficiency to users in Egypt. Hatley aims to streamline the delivery process by connecting users with delivery personnel through a user-friendly platform. The primary objective of Hatley is to provide a cost-effective, time of delivery and accessible solution for ordering and delivering various products.

The Hatley application will facilitate seamless order placement and tracking, allowing users to specify their requirements and preferences. Delivery personnel will utilize the platform to receive and fulfill orders promptly, optimizing delivery routes and ensuring timely service. Through its intuitive interface and broad accessibility, Hatley aims to enhance the overall delivery experience and support efficient logistics management.

In addition to standard delivery services, Hatley will also offer features such as competitive bidding and real-time order matching to improve service efficiency and customer satisfaction. By leveraging technology and user-centric design, Hatley seeks to revolutionize the delivery industry in Egypt, making it more convenient, reliable, and accessible to all.

Overall, the Hatley project aims to transform the delivery landscape by providing a comprehensive and user-friendly platform that enhances the delivery experience for both users and delivery personnel, ultimately fostering efficiency and convenience in the delivery process.

### 3.1.2 Hosting Platform

The Hatley app is compatible with multiple cloud hosting platforms, providing scalability and flexibility to meet its requirements. One such platform that offers a range of cloud computing services for hosting web applications is Microsoft Azure. Google Cloud Platform (GCP), which provides a variety of modular cloud services like computation, data storage, analytics, and machine learning, is yet another feasible choice for hosting Hatley. Usually, financial information like a credit card number or bank account is needed to register for GCP. Although Google Cloud Platform and Microsoft Azure offer reliable hosting options, it's important to take into account specific aspects like cost, integration needs, and scalability when selecting Hatley's hosting platform.

### 3.2 System Architecture

The Hatley system architecture, depicted in Figure X, encompasses a comprehensive framework designed to facilitate efficient and seamless delivery services. It consists of various modules and components, including order management, delivery tracking, user authentication, bidding system, and rating feedback mechanism. The architecture prioritizes scalability, reliability, security, and performance to ensure optimal service delivery for users and delivery personnel.

The Hatley system architecture leverages modern technologies such as geolocation services for dynamic order management and efficient delivery routing and cloud computing and real-time communication to deliver a highly responsive and intuitive delivery service platform. By embracing innovative solutions and user-centric design principles, The system aims to optimize the delivery process, minimize delivery times, and maximize customer satisfaction, revolutionizing the Egyptian delivery industry.

### 3.2.1 Architectural Design

The architecture of the Hatley app follows a multi-tier architecture, as depicted in Figure 42, which comprises the following layers:

**1. Front-end layer:** This layer is responsible for presenting the user interface and interacting with users. In the case of Hatley, the front-end layer utilizes HTML, CSS, and JavaScript for web-based applications, and React for creating dynamic and responsive user interfaces. For the mobile application, React Native is employed to develop cross-platform mobile apps that offer consistent user experiences across different devices.

**2. Application layer:** Situated between the front-end and back-end layers, the application layer hosts the business logic of the Hatley app. It processes user inputs, performs necessary computations, and communicates with the back-end layer to fetch or update data. In Hatley, the application layer is implemented using ASP.NET Core API, which provides a robust framework for building scalable and secure web APIs.

**3. Backend layer:** This layer manages the storage and retrieval of data, serving as the backbone of the application. The backend layer of Hatley utilizes SQL Server as the database management system, ensuring efficient data storage and retrieval. SQL Server offers features such as reliability, scalability, and transaction support, making it suitable for handling the data requirements of a delivery service application.

By employing this architectural design, Hatley ensures separation of concerns, scalability, and maintainability of the application. The use of modern technologies such as React and React Native for the front-end, ASP.NET Core API for the application layer, and SQL Server for the backend layer enables Hatley to deliver a seamless and efficient delivery service experience to its users.

### 3.2.2 Decomposition Description

Decomposition involves breaking down a complex system like the Hatley delivery application into smaller, more manageable components. Here's how we can decompose its architecture:

**1. User Interface (UI):** This component focuses on providing a user-friendly interface for customers to interact with the Hatley app. It includes features such as order placement, tracking, and feedback submission. Separating the UI ensures clarity and ease of maintenance.

**2. Order Management:** Responsible for handling the processing of orders, including creation, assignment to delivery personnel, and status updates. This component ensures efficient order handling and scalability as the application grows.

**3. Bidding System:** Enables delivery personnel to bid on orders based on factors like delivery time and cost. Separating this functionality ensures flexibility and optimization without affecting other parts of the app.

**4. User Authentication and Authorization:** Ensures secure access to the app, managing user accounts, authentication, and authorization levels. Separating this component enhances security and simplifies user management.

**5. Notification System:** Facilitates communication between users and delivery personnel, sending notifications for order updates and important events. Isolating this functionality ensures timely and reliable notifications.

**6. Location Services:** Handles location tracking and geocoding functionalities for accurate delivery location specification and real-time order tracking. Separating this component ensures efficient handling of location-related tasks.

**7. Payment Gateway Integration:** Integrates with payment gateways for secure online payments. Segregating this component ensures reliable and secure payment processing.

**8. Rating and Feedback System:** Allows users to rate their delivery experience and provide feedback. This component contributes to service improvement and customer satisfaction without complicating other functionalities.

Each component contributes to the Hatley application's overall functionality and efficiency. Decomposing the system architecture in this way ensures a clear separation of concerns, scalability, and maintainability, facilitating efficient development and deployment processes.

### 3.2.3 Design Rationale

We chose to organize the Hatley app using a multi-tier architecture because it helps us keep things organized and manageable as the app grows. Here's why we made this decision:

1. **Clear Separation of Concerns:** Each part of the app has its own job to do. The front-end layer handles what you see and interact with, like placing orders. The middle layer takes care of the logic behind the scenes, like processing orders. And the back-end layer deals with storing and retrieving data, like your order history. Keeping these parts separate makes it easier to work on and update the app without causing problems in other areas.
2. **Scalability:** Our architecture allows us to grow the app smoothly. We can scale each part independently to handle more users and data as the app becomes more popular. This means we can keep providing a smooth experience even when lots of people are using the app at the same time.
3. **Maintainability:** With everything neatly organized into layers, it's easier for us to keep the app running smoothly over time. If we need to make changes or fix something, we can focus on one layer without worrying about breaking the rest of the app.
4. **Flexibility with APIs:** We use APIs to connect the different layers of the app. This gives us flexibility to add new features or connect with other services in the future. It's like having different parts of the app talk to each other in a way that's easy to understand and change.

Overall, this architecture helps us build and grow Hatley in a way that's reliable, easy to work with, and adaptable to future needs.

### 3.3 Data Design

The Data Design section of the Hatley app documentation outlines the architecture of the software and how the various components interact with each other. It defines the different modules of the system, their functionalities, and the data structures that they use. Additionally, it describes the relationships between the different modules and their data flow. This section provides an overview of the data flow within the system and how the various modules interact with each other to perform the necessary operations.

### 3.3.1 Data Description

The data description of the Hatley app can be divided into three main entities, each representing different aspects of the application:

**1. User Entity:** This entity represents the individuals who request orders through the app. Users can place multiple orders. The user entity includes attributes such as user ID, username, and contact information.

**2. Delivery Entity:** This entity manages the delivery process within the app. It includes information about each order, such as order description, delivery price, and estimated delivery time. The delivery entity establishes a relationship between users and their respective orders, indicating who placed each order and its details.

**3. Order Entity:** This entity contains details about the orders placed by users. Each order is associated with a specific user and includes information such as the items requested, delivery address, and any special instructions. The order entity facilitates the interaction between users and delivery personnel, ensuring smooth order processing and delivery.

### 3.3.2 Data Dictionary

A data dictionary serves as a comprehensive guide to the data objects within the Hatley app, defining their structure, characteristics, and relationships. It ensures consistency and accuracy of data throughout the system. Here are some key data elements in the Hatley app's data dictionary:

**1-User**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Allow Nulls | Description |
| User ID | int | NO | Unique identifier for the user |
| Name | nvarchar(MAX) | NO | Name of the user |
| Phone | nvarchar(MAX) | NO | Phone number of the user |
| Email | nvarchar(MAX) | NO | Phone number of the user |
| Password | nvarchar(MAX) | NO | Password of the user |
| rate | float | Yes | Rate of user |

**2-Delivery**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Allow Nulls | Description |
| Delivery ID | int | NO | Unique identifier for the delivery |
| Name | nvarchar(MAX) | NO | Name of the delivery person |
| Phone | nvarchar(MAX) | NO | Phone number of the delivery person |
| Email | nvarchar(MAX) | NO | Email address of the delivery person |
| Password | nvarchar(MAX) | NO | Password of the delivery person |
| rate | float | Yes | Rating of the delivery person |
| National ID | int | NO | National ID of the delivery person |
| Front National ID Img | nvarchar(MAX) | NO | Image of the front of the delivery person's National ID |
| Back National ID Img | nvarchar(MAX) | NO | Image of the back of the delivery person's National ID |
| Face with National ID Img | nvarchar(MAX) | NO | Image of the delivery person's face with their National ID |
| Governorate ID | int | NO | Foreign key to the Governorates table |

**3-Order**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Allow Nulls | Description |
| Order ID | int | NO | Unique identifier for the order |
| User ID | int | NO | Foreign key to the Users table |
| Description | nvarchar(2000) | NO | Description of the order |
| Location | nvarchar(MAX) | NO | Location of the order pickup |
| Price | int | NO | Price of the order |
| Status | int | NO | Status of the order |
| Delivery ID | int | YES | Foreign key to the Delivers table |

# **4.Software Requirements Specification**

## **4.1 Introduction:**

Think of this document as your project's handbook, guiding you through all the important details about what we're building and why it matters.

### Intended Audience and Reading Suggestions:

This document is for everyone involved in the project, project manager, development team, and quality assurance testers. It gives insights into project requirements and scope, catering to different needs within the project lifecycle.

**Reading Suggestions:**

* Clients are encouraged to review the entire document to understand project requirements and scope thoroughly.
* Developers should focus on sections detailing functional requirements, system architecture, and data design.
* Testers should pay attention to sections outlining system features, testing procedures, and requirements for validation and verification.

## Product Scope:

The product scope defines the boundaries of the software project and outlines the features and functionalities to be included in the final product.

**The project scope encompasses:**

* Essential features and functionalities necessary to achieve project objectives.
* Exclusions and limitations that clarify what is not within the project scope.
* Interfaces with external systems or components essential for software operation.
* Constraints, assumptions, or dependencies that may impact software development or implementation.