MARMARA UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

CSE 3044 – Software Engineering Term Project

QuickRoute

Software Requirements Specification

Group 17

150119667 Ege Eren Ellez
150119675 Hakan Kenar
150119695 Mustafa Kerem Ekinci
150119703 Mert Akbal
150517059 Özge Saltan
150120531 Hasan Şenyurt
150119906 Zahit Erdem Güzel
150119798 Eşref Emre Koca

14.04.2023

Prepared for

CSE3044 Software Engineering Term Project

Table of Contents

1. INTRODUCTION 3
1.1 Purpose 3
1.2 Scope 3
1.3 Definitions, Acronyms, and Abbreviations 4
1.4 References 5
1.5 Overview 5
2. GENERAL DESCRIPTION 5
2.1 Product Perspective 6
2.2 Product Functions 7
2.3 User Characteristics 8
2.4 General Constraints 9
2.5 Assumptions and Dependencies 9
3. SPECIFIC REQUIREMENTS 10
3.1 External Interface Requirements 10
3.1.1 User Interfaces 10
3.1.2 Hardware Interfaces 11
3.1.3 Software Interfaces 11
3.1.4 Communications Interfaces 11
3.2 Functional Requirements 11
3.2.1 User Registration 11
3.2.2 Order Placement 12
3.2.3 Order Matching 13
3.2.4 Order Acceptance 14
3.2.5 Order Tracking 14
3.2.6 Delivery Confirmation 15
3.2.7 Payment Processing 16
3.2.8 Rating and Feedback 17
3.2.9 System Administration 17
3.2.10 Reporting 18
3.2.11 Courier Management 19
3.2.12 Customer Management 20
3.3 Non-Functional Requirements 21
3.3.1 Performance 21
3.3.2 Reliability 21
3.3.3 Availability 21
3.3.4 Security 21
3.3.5 Maintainability 22
3.3.6 Portability 22
3.4 Inverse Requirements 22
3.5 Design Constraints 23
3.6 Logical Database Requirements 24
3.7 Other Requirements 24
4. UML DIAGRAMS 26
4.1 Use Cases 26
4.2 Classes / Objects 27
4.3 Sequence Diagrams 28
A. APPENDICES 29
A.1 Contribution of the Group Members to Software Requirements Specification Document 29

1. Introduction

1.1 Purpose

The purpose of this document is to explain our project, which is an application of cargo delivery for customers. It is called QuickRoute. This document describes the software requirements of the system, general description, user interfaces and UML diagrams. The goal of this project is to create a mobile application that makes it easier for clients and couriers to receive their packages. Customers are demanding faster and more convenient delivery options in today's fast-paced world, and courier services need to improve their delivery routes to stay competitive. The suggested mobile application provides a response to these two issues.

Customers can quickly order cargo delivery through the mobile application, selecting their chosen delivery location, and paying for the service within the app. They can also receive reminders regarding the status and anticipated arrival time of their delivery, allowing them to follow its progress in real-time.

Overall, this smartphone app strives to increase the convenience and effectiveness of cargo delivery for both clients and couriers, offering a useful resource for both individuals and enterprises.

1.2 Scope

The scope of a mobile application for cargo delivery which is name QuickRoute with the following characteristics is included in the project's scope:

- Users must register an account in the application by providing personal information such as their name, phone number, and email address.
- The software will pair users with the closest available couriers depending on how close their locations are to each other. When an order is accepted, customers can choose a delivery location, which will be displayed to the courier. Users of the application will be able to submit orders for delivery. The order will then be delivered to nearby couriers who are available. Depending on their schedule and workload, the courier may accept or refuse the request.
- Clients can follow the progress of their orders in real time, getting updates on the courier's whereabouts and anticipated delivery time. The app will provide a safe and practical

payment processing system that will let users pay for deliveries right inside the app. Customers will be able to rate the courier and offer feedback on their experience receiving deliveries.

- The software will be created for both the iOS and Android operating systems, and it will work with a variety of mobile devices. Modern software development processes, tools, and technologies will be used throughout the project to provide a high-quality and user-friendly application.
- The project's scope also includes creating an admin panel for managing couriers and customers in addition to the mobile application for cargo delivery. The following functionalities will be available in the admin panel: The admin panel will enable control of the accounts of the couriers, including their availability, personal information, and workload. Additionally, it will enable the evaluation of courier performance, including ratings and delivery timeframes.
- You may manage customer accounts, including their personal data, order history, and payment information, using the admin panel. A consolidated place for managing orders, including the ability to view and update orders, track deliveries, and handle payments, will be made available through the admin panel. A secure login will be required to access the web-based admin panel. Its interface will be simple and contemporary, and it will be made to be user-friendly and intuitive. To provide a top-notch and useful admin panel, the project will make use of contemporary software development processes, tools, and technologies.

1.3 Definitions, Acronyms, and Abbreviations

- UI: User Interface
- Customer: A person that can give an order to courier.
- Courier: A person that will carry the package of customer to given location.
- HTML: Stands for HyperText Markup Language
- CSS: a stylesheet language used to describe the presentation of a document written in HTML or XML.
- TypeScript: A programming language used on the client-side.
- Angular: A JavaScript/Typescript framework.

- MySQL: Open-source relational database management system used to store and manage data for various applications.
- Java Spring Boot: Open-source framework used to quickly and easily create standalone, production-grade Spring-based applications.
- DBMS: Database Management System

1.4 References

[1] IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998, 1998. Available: https://ieeexplore.ieee.org/document/720574. Accessed on: Apr. 9, 2023.

1.5 Overview

A general description, specific requirements, and multiple UML diagrams are included in the project report's various sections. The specific requirements section lists the requirements for external interfaces, functional requirements, non-functional requirements, inverse requirements, design constraints, logical database requirements, and other requirements while the general description gives an overview of the project's goals. Use case and class diagrams, sequence diagrams, data flow diagrams, and state-transition diagrams are all included in the UML diagrams. These illustrations give a visual picture of the structure and operation of the system.

2. General Description

It is a courier application that secures delivery of orders in a simple way by registering to become a courier that anyone can work with.

This application would receive orders from customers of a restaurant or supermarket, for example, and direct those orders to registered couriers within a certain distance. Registered couriers would be able to select orders they wish to accept and communicate with the customer to make the delivery.

This application would help customers receive their orders quickly and safely while also providing flexible job opportunities for registered couriers. The application would also have a tracking system that would allow customers to monitor their orders and for couriers to update their delivery status.

In summary, this type of courier application would meet the needs of both customers and couriers by providing a flexible working environment and convenience for users.

2.1 Product Perspective

Product Name:
QuickRoute

Product Description:
QuickRoute is a courier application that allows customers to have their orders delivered quickly and securely while also providing flexible working opportunities for registered couriers.

Target Audience:

- Restaurants
- Supermarkets
- Customers.

Features:

- Customers can easily create orders through the application.
- Registered couriers can accept and deliver orders within a certain distance.
- A tracking system for couriers to update their delivery status.
- A messaging system for couriers to communicate with customers.
- A notification system for customers to track their orders.
- A payment gateway for managing payments.

Benefits:

- Orders are delivered quickly and securely for customer satisfaction.
- Registered couriers have flexible working opportunities.
- Restaurants and stores can efficiently manage their delivery operations.
- Improved customer satisfaction.
- Easy payment through the application.

Competitive Advantage:

QuickRoute is designed with extra security and privacy measures to ensure the safety of its users. It provides registered couriers with flexible working opportunities while also delivering orders quickly and securely.

Marketing Strategy:

The application can collaborate with restaurants and stores for marketing purposes. It can also offer special discounts and promotions to its customers. Advertising campaigns can be conducted on social media and other online platforms.

This product perspective explains the main features and benefits of the QuickRoute courier application. The application is designed to deliver orders quickly and securely while also providing flexible working opportunities for registered couriers.

2.2 Product Functions

Registration Function:

This function allows couriers to register themselves on the platform by submitting their basic information, including name, contact details, identification.

Order Creation Function:

Customers can create orders by entering the delivery address and selecting the restaurant or supermarket. They can also provide any specific delivery instructions and pay for their orders through the payment gateway.

Order Management Function:

This function allows couriers to manage their orders by accepting or rejecting them based on their availability and distance. Couriers can also update the delivery status of their orders through the tracking system.

Messaging Function:

The messaging function allows couriers to communicate with customers regarding the delivery status of their orders. They can also notify customers of any delays or changes in the delivery schedule.

Notification Function:

The notification function allows customers to receive real-time updates on the delivery status of their orders. They can also receive notifications when their orders are on their way or have been delivered.

Payment Function:

The payment function allows customers to make payments for their orders through the payment gateway. The application also provides a platform for managing payments to couriers.

Verification Function:

The verification function allows the application to verify the identity and vehicle information of couriers to ensure the safety and security of customers.

These product functions are essential to the QuickRoute courier application. They enable the application to provide quick and secure deliveries for customers while also providing flexible working opportunities for registered couriers.

2.3 User Characteristics

Customers:

The main user group of the QuickRoute application. Customers can place orders through the app, make payments, track the delivery status of their orders, and communicate with the couriers.

Couriers:

Couriers are the people who are registered on the app and deliver the orders. Couriers can accept or reject orders, update the delivery status, and communicate with customers.

Administrators:

The administrators of the QuickRoute app are the people who manage the app and the activities and orders of the couriers. Administrators can manage user accounts, manage payments, and monitor the app's performance.

Customer Service Representatives:

Customer service representatives are individuals who can communicate through the app to solve problems experienced by customers or couriers. They can also gather feedback and monitor the app's performance.

These user features demonstrate that the QuickRoute courier application is designed to meet the needs of different user groups.

2.4 General Constraints

Geographical constraints:

The QuickRoute application covers courier services in a specific region. Therefore, the application may not be able to provide services to users who do not have a specific geographic region.

Technology constraints:

The QuickRoute application requires a specific technological infrastructure. Therefore, users need to have a compatible smartphone and internet connection to use the application.

Payment constraints:

The QuickRoute application accepts specific payment methods. Therefore, users need to have a specific payment method to make payments through the application.

Laws and regulations:

The QuickRoute application must comply with specific legal regulations. For example, the application must protect users' personal data during payment transactions.

Security constraints:

The QuickRoute application takes security measures to protect users' personal and financial data. However, any security vulnerability can lead to the misuse of users' data.

These general constraints demonstrate the limitations of the QuickRoute courier application and highlight the need for application developers to consider these constraints in providing the best possible experience for all users of the application.

2.5 Assumptions and Dependencies

Assumptions:

- Users will have a compatible smartphone and internet connection to use the application.
- Courier services will be limited to a specific geographic region.
- Users will have a specific payment method to make payments through the application.
- Users will accurately and completely identify their orders and couriers.
- Users will verify their identity when receiving couriers.
- Couriers will accurately and completely deliver orders and couriers.
- Users will regularly use the application to track the status of their couriers.

Dependencies:

- The application will integrate with a third-party payment processor to perform payment transactions.
- The application will operate within a specific geographic region to provide courier services.
- Users will need to have a specific payment method to make payments.
- Users will need to provide accurate and complete order information.
- Users will require an appropriate identity verification method to verify their identity.
- Couriers will need to have accurate and complete order information.
- The application will need an appropriate location technology to track the courier's location accurately.

These assumptions and dependencies should be considered during the design and development of the QuickRoute courier application.

3. Specific Requirements

While designing the QuickRoute mobile application, we tried to enrich it in many ways. We have added many options and features to give users a good experience. In this section, we will share the functional and non-functional features that we have implemented and hope to realize in the QuickRoute mobile application.

3.1 External Interface Requirements

3.1.1 User Interfaces

In the QuickRoute app, we wanted to design a mobile app that allows customers to place orders and couriers to accept or reject orders based on availability. We want the design of this mobile app to be easy, user-friendly, and comfortable to use.

The application consists of a main screen where users can register or log in. There are "Become a Courier" and "Place Order" options on the main screen. Users who want to be couriers upload their identity verification documents by entering their personal information. Users who want to place an order choose from the product or service catalog and confirm their order by specifying the delivery address. Users can track their orders and view the courier's estimated delivery time.

Payment transactions can be made securely and different payment options are offered. Users can also rate the delivery service and provide feedback.

3.1.2 Hardware Interfaces

As the hardware interface of the QuickRoute application, customers and couriers must have a mobile device. The software of mobile devices must have an up-to-date and reliable internet connection.

3.1.3 Software Interfaces

The software interface for QuickRoute will include APIs and other software components for payment or mapping.

3.1.4 Communications Interfaces

QuickRoute's communication interface will contain the protocols and technologies utilized to allow communication between the system's many components. QuickRoute comprises protocols for data communication between the mobile application and the server, as well as protocols for data transfer between the server and the couriers' mobile devices. To guarantee that orders are received and delivered without problems or vulnerabilities, communication interfaces must be dependable and secure.

3.2 Functional Requirements

3.2.1 User Registration

3.2.1.1 Introduction / Description

This functionality allows users to register as couriers by providing their personal information and relevant documents.

3.2.1.2 Inputs / Display

User input fields must be filled in for personal information such as name, address, phone number and e-mail address. In addition, if the courier will deliver by using a vehicle, it should be uploaded for relevant documents such as driver's license, insurance document.

3.2.1.3 Processing

The system verifies the user's information and documents and creates a new account for the courier.

3.2.1.4 Outputs

A confirmation message or email with the courier's account information and instructions on how to use the system.

3.2.1.5 Constraints

The system must ensure that the user satisfies the minimal standards for being a courier, such as having a valid driver's license and insurance if they deliver by car.

3.2.1.6 Error/Data Handling

The system should provide appropriate error messages if any required information or documents are missing or incorrect.

3.2.2 Order Placement

3.2.2.1 Introduction / Description

This functionality allows customers to place orders by selecting the desired product or service and specifying the delivery location.

3.2.2.2 Inputs / Display

User input fields for the product or service to be delivered, delivery location, and payment information.

3.2.2.3 Processing

The system processes the order and matches it with the nearest available courier.

3.2.2.4 Outputs

A confirmation message or email with the order details and estimated delivery time.

3.2.2.5 Constraints

The system must verify that the delivery location is within the service area and that the payment information is valid.

3.2.2.6 Error/Data Handling

The system should provide appropriate error messages if any required information is missing or incorrect.

3.2.3 Order Matching

3.2.3.1 Introduction / Description

This functionality matches the customer's order with the nearest available courier based on the customer's location and courier's availability.

3.2.3.2 Inputs / Display

Customer's delivery location and courier's availability.

3.2.3.3 Processing

The system uses location-based algorithms to match the order with the nearest available courier based on their availability.

3.2.3.4 Outputs

The courier's account is notified of the new order.

3.2.3.5 Constraints

The system must verify that the courier is available and within a reasonable distance from the delivery location.

3.2.3.6 Error/Data Handling

The system should provide appropriate error messages if no available couriers are found.

3.2.4 Order Acceptance

3.2.4.1 Introduction / Description

This functionality allows couriers to accept or reject orders based on their availability.

3.2.4.2 Inputs / Display

Order details such as delivery location, estimated delivery time, and payment information.

3.2.4.3 Processing

The courier reviews the order details and decides to accept or reject the order.

3.2.4.4 Outputs

The system notifies the customer and other available couriers of the courier's decision.

3.2.4.5 Constraints

The courier must respond within a reasonable time frame and must have a valid account with the system.

3.2.4.6 Error/Data Handling

The system should provide appropriate error messages if the courier is unable to accept the order.

3.2.5 Order Tracking

3.2.5.1 Introduction / Description

The system should allow customers to track the status of their orders, including the courier's location and estimated delivery time.

3.2.5.2 Inputs / Display

The system should display the customer's order details, including the courier's current location and estimated delivery time.

3.2.5.3 Processing

The system should continuously update the courier's location and estimated delivery time based on GPS tracking data.

3.2.5.4 Outputs

The system should display the updated courier location and estimated delivery time to the customer.

3.2.5.5 Constraints

The system should only display the courier's location and estimated delivery time for the customer's order.

3.2.5.6 Error/Data Handling

The system should display an error message if the courier's location cannot be tracked.

3.2.6 Delivery Confirmation

3.2.6.1 Introduction / Description

Couriers should be able to confirm delivery of the order through the system.

3.2.6.2 Inputs / Display

The system should display the courier's current order details, including the customer's name and delivery location.

3.2.6.3 Processing

The system should update the order status to "delivered" once the courier confirms delivery.

3.2.6.4 Outputs

The system should display a confirmation message to the courier and update the order status to "delivered".

3.2.6.5 Constraints

The system should only allow the courier who delivered the order to confirm delivery.

3.2.6.6 Error/Data Handling

The system should display an error message if the courier tries to confirm delivery for the wrong order.

3.2.7 Payment Processing

3.2.7.1 Introduction / Description

The system should process payments for the orders placed by customers and provide payment to the couriers for their services.

3.2.7.2 Inputs / Display

The system should display the order details, including the total amount and payment method.

3.2.7.3 Processing

The system should process the payment using the selected payment method and update the courier's account with their earnings.

3.2.7.4 Outputs

The system should display a confirmation message to the customer and update the courier's account with their earnings.

3.2.7.5 Constraints

The system should only allow payment for completed and confirmed orders.

3.2.7.6 Error/Data Handling

The system should display an error message if the payment cannot be processed.

3.2.8 Rating and Feedback

3.2.8.1 Introduction / Description

The system should allow customers to rate the quality of the courier's service and provide feedback on their experience.

3.2.8.2 Inputs / Display

The system should display the order details and provide a rating scale and text box for feedback.

3.2.8.3 Processing

The system should record the customer's rating and feedback and display it to the courier.

3.2.8.4 Outputs

The system should display the customer's rating and feedback to the courier.

3.2.8.5 Constraints

The system should only allow one rating and feedback per order.

3.2.8.6 Error/Data Handling

The system should display an error message if the rating and feedback cannot be recorded.

3.2.9 System Administration

3.2.9.1 Introduction/Description

The System Administration interface gives administrative users access to the tools they need to manage the system. Manage user accounts, orders, payments, and other system components.

3.2.9.2 Inputs/Display

Information such as user account information, order data, and payment history will be displayed in the System Administration interface. Administrative users will also be able to input and change information such as user account data, order statuses, and payment information.

3.2.9.3 Processing

The System Administration interface will take user input and conduct required actions such as creating or upgrading user accounts, maintaining order statuses, and processing payments.

3.2.9.4 Outputs

The System Administration interface will display information such as updated user account data, order status updates, and payment processing information.

3.2.9.5 Constraints

Only approved administrative users with sufficient permissions will be able to access the System Administration interface. It will also follow data protection rules and regulations to maintain user privacy and security.

3.2.9.6 Error/Data Handling

Error-handling techniques will be included in the System Administration interface to prevent unauthorized access and to manage any problems that may arise during data processing. It will also include data handling methods to maintain the integrity and correctness of the data. Any mistakes or data problems will be recorded and forwarded to the proper persons for remediation.

3.2.10 Reporting

3.2.10.1 Introduction/Description

The reporting capability should offer a thorough picture of system activity over a specified time period, including the number of orders, payments, and ratings.

3.2.10.2 Inputs/Display

Administrators should be able to choose the date range for the report and the type of report they want to create via the reporting interface.

3.2.10.3 Processing

To build the report, the system should collect data from the required modules and process it.

3.2.10.4 Outputs

The system should generate a report in a user-friendly format, such as a PDF or Excel spreadsheet.

3.2.10.5 Constraints

The reporting functionality should be restricted to authorized users with administrative privileges.

3.2.10.6 Error/Data Handling

The system should handle errors gracefully, such as by displaying an error message if the user enters an invalid date range.

3.2.11 Courier Management

3.2.11.1 Introduction/Description

The courier management functionality should allow administrators to manage the courier accounts effectively.

3.2.11.2 Inputs/Display

The courier management interface should allow administrators to view and update courier account information, including their personal information and status.

3.2.11.3 Processing

The system should process administrator requests to update courier accounts, including activating and deactivating accounts and updating personal information.

3.2.11.4 Outputs

The system should update the courier account information and display any relevant notifications to the administrator.

3.2.11.5 Constraints

The courier management functionality should be restricted to authorized users with administrative privileges.

3.2.11.6 Error/Data Handling

The system should handle errors gracefully, such as by displaying an error message if the administrator enters invalid information.

3.2.12 Customer Management

3.2.12.1 Introduction/Description

The customer management functionality should allow administrators to manage the customer accounts effectively.

3.2.12.2 Inputs/Display

The customer management interface should allow administrators to view and update customer account information, including their personal information and status.

3.2.12.3 Processing

The system should process administrator requests to update customer accounts, including activating and deactivating accounts and updating personal information.

3.2.12.4 Outputs

The system should update the customer account information and display any relevant notifications to the administrator.

3.2.12.5 Constraints

The customer management functionality should be restricted to authorized users with administrative privileges.

3.2.12.6 Error/Data Handling

The system should handle errors gracefully, such as by displaying an error message if the administrator enters invalid information.

3.3 Non-Functional Requirements

3.3.1 Performance

This refers to how well the system should perform in terms of speed and response time. In particular, the system should:

- Respond to user requests within a maximum of 2 seconds.
- Support up to 1000 simultaneous user sessions.
- Handle up to 1000 orders per hour.

3.3.2 Reliability

This refers to the system's ability to operate without failure or errors. In particular, the system should:

- Have a 99.9% uptime guarantee.
- Store data in a reliable and fault-tolerant way.
- Have backup and disaster recovery procedures in place.

3.3.3 Availability

This refers to the system's ability to be accessible and usable at all times. In particular, the system should:

- Be accessible 24/7 from any location with internet access.
- Have a maximum downtime of 1 hour per month for maintenance purposes.

3.3.4 Security

This refers to the system's ability to protect data and prevent unauthorized access. In particular, the system should:

- Use secure communication protocols (e.g., HTTPS) to protect user data.
- Store user data (e.g., passwords) in an encrypted format.
- Implement access controls to prevent unauthorized access to the system.

3.3.5 Maintainability

This refers to the system's ability to be easily maintained and updated. In particular, the system should:

- Be modular and scalable, to allow for easy modification and expansion.
- Have clear and well-documented code and system documentation.
- Be easily testable and debuggable.

3.3.6 Portability

This refers to the system's ability to run on different platforms and environments. In particular, the system should:

- Be designed to be platform-independent and work on different operating systems and devices.
- Have clear system requirements and dependencies.
- Be easily deployable to different hosting environments.

3.4 Inverse Requirements

- The system's delivery times shouldn't be excessively long. The system should give preference to local couriers with quick delivery times so that orders are delivered swiftly and effectively.
- The amount of orders that can be placed at once shouldn't be restricted by the system.
- The amount of orders that can be placed at once shouldn't be restricted by the system. Customers should be able to place several orders at once through the system, making it possible for them to buy whatever they require at once.
- The system shouldn't only offer a small number of eateries or meal options. The system should collaborate with a wide variety of eateries and food suppliers to give clients a wide choice of options.
- Orders for items that are out of stock should not be accepted by the system. Customers shouldn't be able to place orders for items that are out of stock thanks to the system's real-time inventory tracking and updating of the item availability status on the app.

- There shouldn't be regional pricing variations in the system. In order to provide transparency and fairness for all customers, the system should use geolocation services to determine the customer's location and present accurate pricing that is uniform across different regions.
- The system should not have a high delivery fee that discourages customers from placing orders. The system should offer competitive delivery fees that are reasonable and affordable for customers, encouraging more orders and repeat business.

3.5 Design Constraints

Payment processing standards:

Mobile app will process payments, strict requirements are required for secure payment processing. May be required to comply with industry standards such as PCI DSS (Payment Card Industry Data Security Standard).

Data protection regulations:

Depending on the location of the users and the company, the mobile app may need to be GDPR (General Data Protection Regulation) bound or, as a Turkish-based business, to comply with consumer law.

Third-party service limitations:

The mobile app relies on third-party services such as mapping or delivery tracking systems, may be subject to limitations such as API usage limits or pricing tiers.

Device compatibility:

The mobile app needs to be designed to run on a wide variety of devices with different screen sizes, processing power and operating systems, which can affect the design and functionality of the app.

Usability testing:

The mobile app needs to undergo extensive usability testing to ensure it is easy to use and navigate, and meets the needs and expectations of its target audience.

Security protocols:

The mobile app needs to implement additional security protocols such as multi-factor authentication to protect user data and prevent unauthorized access.

User feedback:

The mobile application may need to provide a mechanism for users to provide feedback, report bugs, or suggest improvements to the application.

Privacy policies:

The mobile application may need to have a clear and concise privacy policy that explains how user data is collected, used, and protected.

3.6 Logical Database Requirements

The following tables are required for the QuickRoute courier application:

- Users: This table will contain information about the users who register for the application, such as username, password, name, surname, email address, phone number, payment method, and billing information.
- Orders: This table will contain information about the orders placed by users, such as order number, customer ID, courier ID, delivery address, order date, delivery date, order status, list of products, and order amount.
- Couriers: This table will contain information about the couriers, such as courier ID, name, surname, phone number, email address, vehicle registration plate, and courier location.
- Payments: This table will contain information about the payments made by users, such as payment number, customer ID, payment date, payment amount, payment method, and payment status.
- Statistics: This table may contain statistics collected by the application, such as number of orders, number of deliveries, number of cancelled orders, total earnings, and average delivery time.

3.7 Other Requirements

3.7.1 Security

This requirement is critical for ensuring the safety and protection of users' personal and payment information. The application should have secure authentication methods and encryption techniques to prevent unauthorized access or data breaches. It's also important to regularly update and maintain the security features to ensure they remain effective over time.

3.7.2 Usability

The application should be easy to use and navigate to provide a positive user experience. The user interface design should be visually appealing and intuitive, and the menu organization should be logical and easy to follow. Additionally, the application should provide helpful prompts and notifications to guide users through the process of placing orders, tracking deliveries, and making payments.

3.7.3 Performance

The performance of the application is crucial to ensure fast and accurate processing of orders, deliveries, and payments. The application should be designed with efficient algorithms and processing methods to minimize delays and provide accurate tracking of courier locations. Additionally, the application should be optimized to use minimal system resources to avoid performance issues and crashes.

3.7.4 Compatibility

The application should be compatible with a wide range of mobile devices, web browsers, and operating systems to ensure broad accessibility for users. It's important to test the application on different devices and platforms to identify and resolve any compatibility issues that may arise.

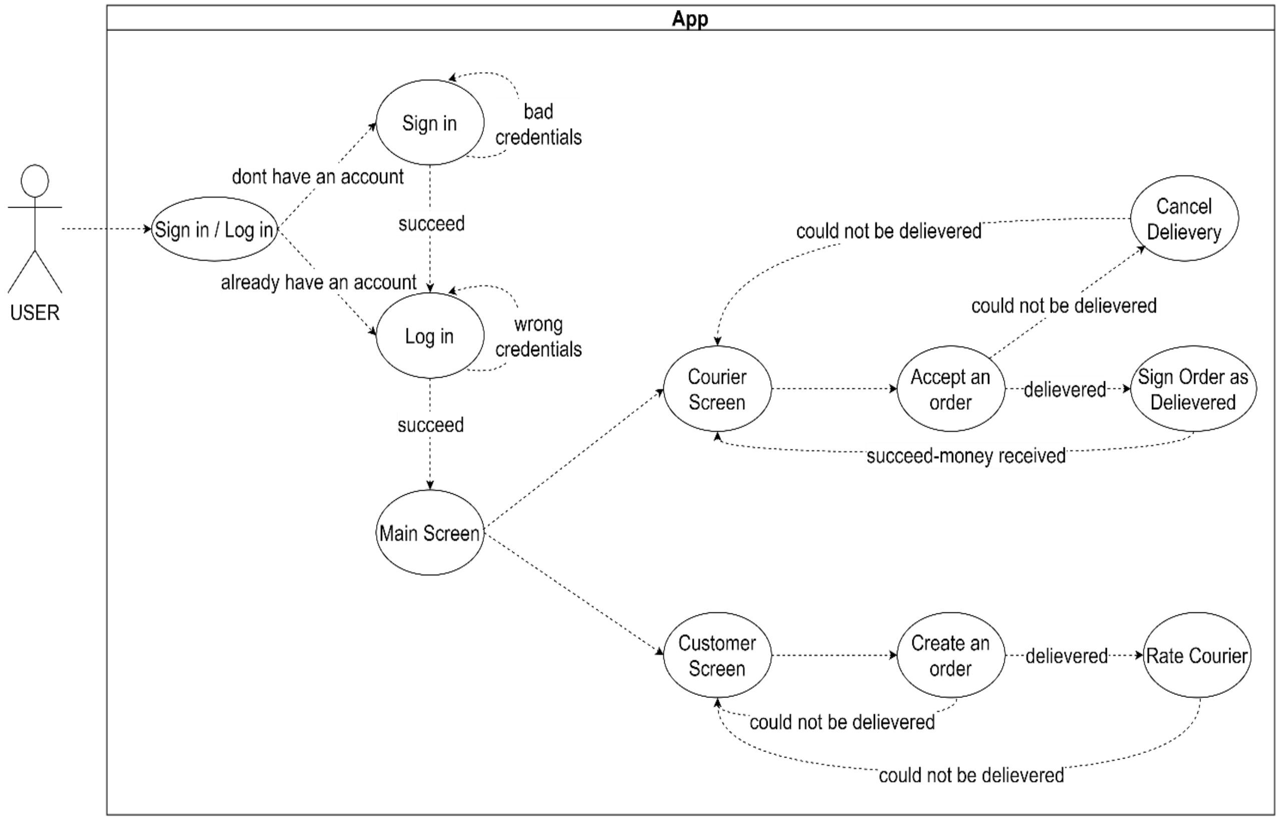
3.7.5 Scalability

The application should be designed with scalability in mind to accommodate a growing user base and increasing demand. This includes having a robust infrastructure and resources to handle increased traffic and orders, as well as implementing efficient and scalable database designs and processing methods.

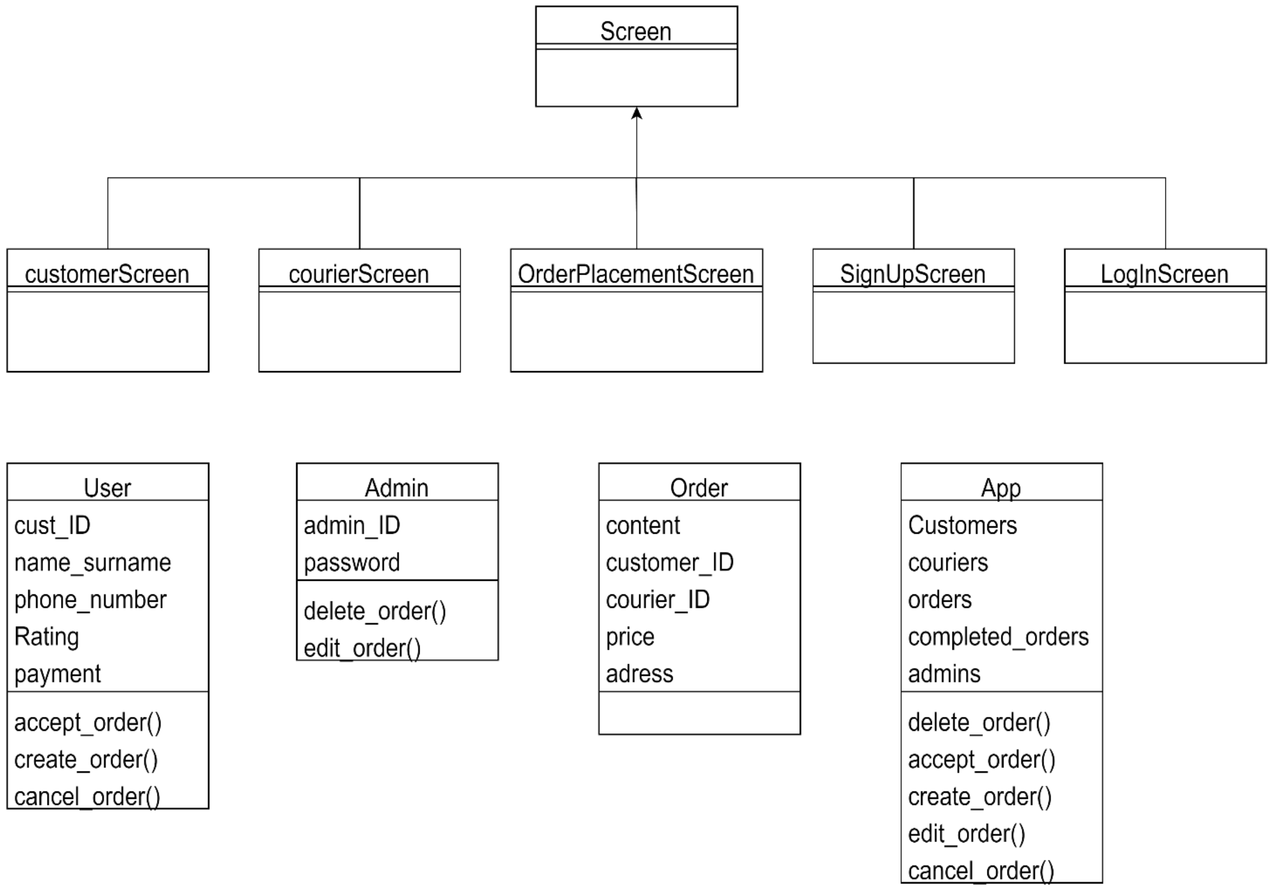
By considering these logical database requirements and additional requirements, the QuickRoute courier application can be developed to provide a secure, user-friendly, high-performance, and scalable solution that meets the needs of both customers and couriers.

4. UML Diagrams

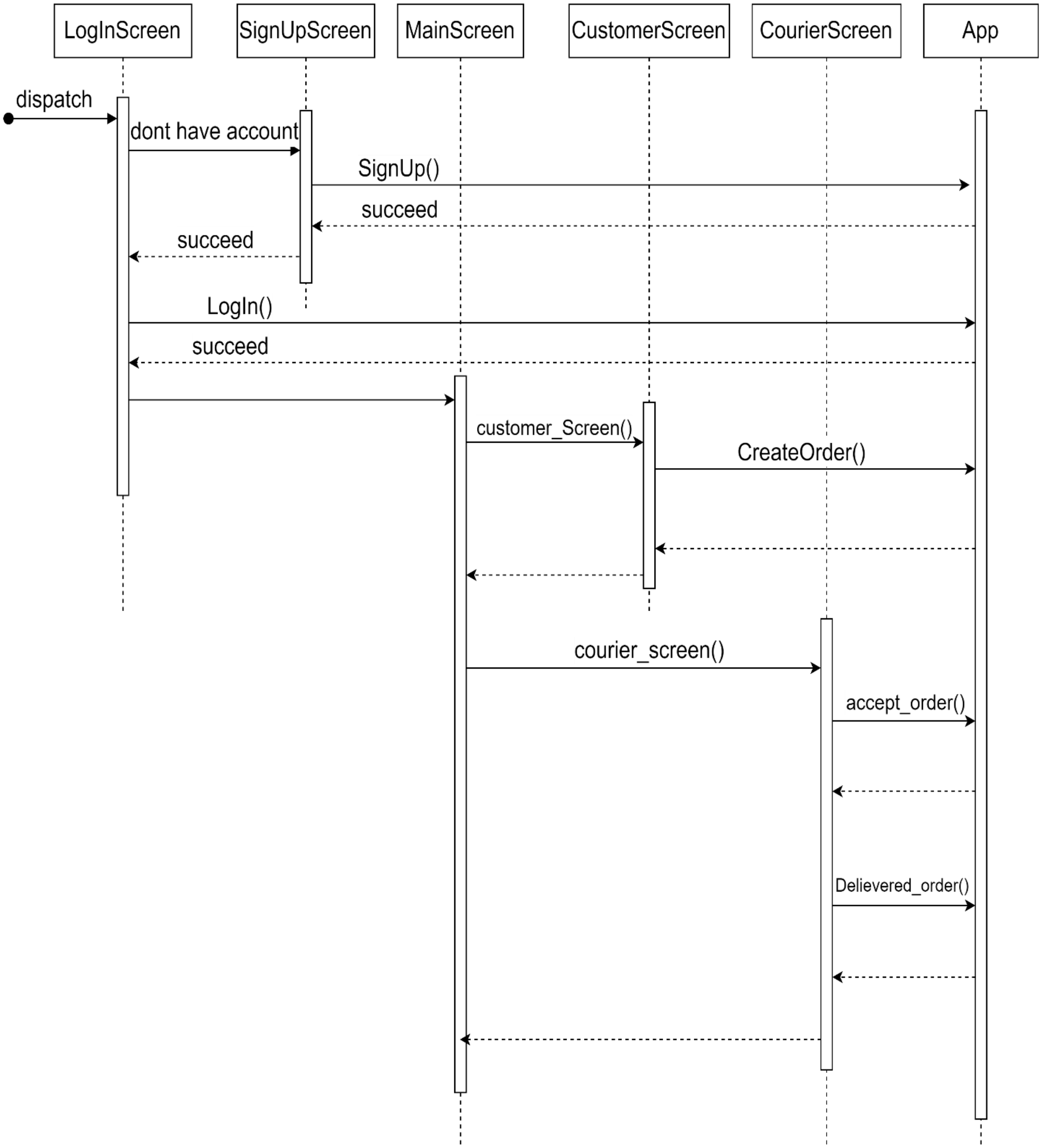
4.1 Use Cases



4.2 Classes / Objects



4.3 Sequence Diagrams



A. Appendices

A.1 Contribution of the Group Members to Software Requirements Specification Document

	Group Member	Contribution
1	Özge Saltan 150517059	External Interface Requirements, Functional Requirements, Non-Functional Requirements
2	Eşref Emre Koca 150119798	Inverse Requirements, Design Constraints
3	Hasan Şenyurt 150120531	Introduction: Purpose, Scope, Definitions, Acronyms, and Abbreviations, References and Overview
4	Hakan Kenar 150119675	Logical Database Requirements, Other Requirements
5	Ege Eren Ellez 150119667	Product Perspective, Product Functions, User Characteristics, General Constraints, Assumptions and Dependencies
6	Mert Akbal 150119703	Use Cases
7	Zahit Erdem Güzel 150119906	Use Cases, Classes/Objects, Sequence Diagrams
8	Mustafa Kerem Ekinci 150119695	Sequence Diagrams