



CS491 Senior Design Project - Fall 2024

Analysis and Requirements Report

YOLLA-T2418

Celal Salih Türkmen - 22102498

Sadra M. Hazarabadi- 22101018

Eslim Rana Emiroğlu - 22102692

Elif Şen - 22102144

Halil Tataroğlu - 22102198

Supervisor: Can Alkan

Innovation Expert: Sinan Sarioğlu

Table of Contents

1. Introduction.....	5
2. Current System.....	6
3. Proposed System.....	6
3.1. Overview.....	6
3.2. Functional Requirements.....	6
3.2.1. Main Features.....	6
3.2.1.1. User Registration and Authentication.....	6
3.2.1.2. Cargo Listing & Search.....	6
3.2.1.3. Package Tracking.....	7
3.2.1.4. Receiving Payments.....	7
3.2.1.5. Notifications.....	7
3.2.1.6. Business Cargo Management.....	7
3.2.1.7. Box Measurer.....	7
3.2.1.8. Chatbot Integration.....	7
3.2.2. Secondary Features.....	7
3.2.2.1. Ratings & Reviews.....	7
3.2.2.2. Cargo Insurance Options.....	7
3.2.2.3. Advertisement.....	8
3.3. Non-Functional Requirements.....	8
3.3.1. Usability.....	8
3.3.2. Reliability.....	8
3.3.3. Performance.....	8
3.3.4. Supportability.....	8
3.4.5. Scalability.....	9
2.2.6. Platform Compatibility.....	9
3.4.7. Security.....	9
3.4.8. Integration.....	9
3.4.9. Efficiency.....	10
3.4.10. Flexibility.....	10
3.4.11. Sustainability.....	10
3.4.11.1. Protecting Environment.....	10
3.4.11.2. Supporting Local Economy.....	10
3.4. Pseudo Requirements.....	11
3.5. System Models.....	11
3.5.1. Scenarios.....	11
3.5.2. Use Case Model.....	13
3.5.3. Object and Class Model.....	14
3.5.3.1. User.....	15
3.5.3.2. Cargo.....	15
3.5.3.3. CargoServiceProvider.....	16

3.5.3.4. Promotion.....	16
3.5.3.5. Address.....	16
3.5.3.6. Service.....	17
3.5.3.7. CargoStatus.....	17
3.5.3.8. Insurance.....	17
3.5.3.9. ChatSession.....	17
3.5.3.10. ChatMessage.....	17
3.5.3.11. UserManager.....	18
3.5.3.12. CargoManager.....	18
3.5.3.13. Notification.....	18
3.5.3.14. BoxCargo.....	18
3.5.3.15. BoxProperties.....	19
3.5.3.16. LetterCargo.....	19
3.5.4. Dynamic Models.....	20
3.5.5.1 Activity Diagrams.....	20
3.5.5.1.1 Login/Register.....	20
3.5.5.1.2 View Page.....	20
3.5.5.1.3 Profile.....	21
3.5.5.1.4 Cargo Send/Receive Operations.....	21
3.5.5.2 State Diagrams.....	22
3.5.5.2.1 User.....	22
3.5.5.2.2 Cargo.....	23
3.5.5.3 Sequence Diagrams.....	23
3.5.5.3.1 Search Sequence Diagram.....	23
3.5.5.3.2 Profile Sequence Diagram.....	25
3.5.5.3.3 Receive Sequence Diagram.....	25
3.5.5. User Interface - Navigational Paths and Screen Mock-ups.....	26
3.5.5.1 Mobile Application.....	26
3.5.5.2 Web Application.....	30
4. Other Analysis Elements.....	31
4.1. Consideration of Various Factors in Engineering Design.....	31
4.1.1. Constraints.....	31
1. Accessibility.....	31
2. Aesthetics.....	32
3. Cost.....	32
4. Legal Considerations.....	32
5. Functionality.....	32
6. Sustainability.....	32
7. Marketability.....	32
8. Interoperability.....	33
9. Technical Specifications.....	33
10. Social, Cultural, and Economic Factors.....	33

4.1.2. Standards.....	33
Application of Standards in Yolla Tech.....	33
4.2. Risks and Alternatives.....	34
1. Server Overload During Peak Hours.....	34
2. Insufficient User Adoption.....	34
3. Data Breach.....	34
4. High Server Costs.....	35
5. Regulatory Non-compliance.....	35
6. Loss of Key Team Members.....	35
7. Technical Bugs and Failures.....	35
8. Competition from Established Platforms.....	36
4.3. Project Plan.....	36
4.4. Ensuring Proper Teamwork.....	40
1. Team Roles and Responsibilities.....	40
2. Communication and Collaboration Tools.....	40
3. Teamwork Strategies.....	40
4. Conflict Resolution.....	41
5. Knowledge Sharing.....	41
6. Accountability and Transparency.....	41
4.5. Ethics and Professional Responsibilities.....	41
1. User Privacy and Data Protection.....	41
2. Inclusivity and Accessibility.....	42
3. Environmental Responsibility.....	42
4. Fairness and Transparency.....	42
5. Professional Conduct.....	42
6. Social Responsibility.....	43
7. Legal Compliance.....	43
8. Ethical Development Practices.....	43
4.7. Planning for New Knowledge and Learning Strategies.....	43
5. Glossary.....	44
6. References.....	44

1. Introduction

The industry of logistics and cargo delivery has grown in the last couple of years due to increasing e-commerce and the demand for efficient and reliable services. It is, however, difficult for individuals and SMEs to operate in logistics. Finding the right cargo option, comparing service providers, and price transparency are major issues faced by the customers. SMEs have to balance cost-efficiency with maintaining a seamless logistics process, especially when traditional methods lack modern optimization and user-friendly features.

Yolla is a digital cargo marketplace that will break the rules in the cargo logistics experience in Türkiye. It makes finding, comparing, and booking cargo services easier by providing one platform for many service providers. It enables customers to find cargo options based on location, size, weight, and delivery time to make a well-informed decision for their needs. Yolla offers more than comparing options: advanced features such as real-time package tracking, responsive notifications, and flexible filtering options will allow the experience of using this website to be smooth and user-friendly.

The key concept in Yolla is cargo delivery at ease, more accessible, and more transparent, hence efficient, for both individuals and SMEs. Yolla applies a responsive design to ensure perfect work both on mobile and web platforms for more ease of use. Yolla provides an efficient way to quickly compare various cargo service providers on pricing, speed of delivery, and other areas to find the best means, helping clients and businesses save time and resources toward improving their core operations. Moreover, with Yolla, users get box measurement using smartphone cameras, real-time updates, and promotions for shipments, hence versatile for every logistical need. In this paper, the Yolla system is reviewed in detail. We begin with the review of the existing solutions of logistics, determining gaps in the market. The next step would be to describe in depth the features of Yolla and their benefits compared with traditional systems. The functional and non-functional and pseudo requirements of the system are then made to define the scope and objective of the project. System models will be demonstrated later, supported by architectural diagrams and use case scenarios to show the technical structure and possible applications of Yolla. Conclusively, impacts on users, businesses, and overall impacts in the logistics industry due to the existence of this platform will be deliberated in this report.

This report begins with an analysis of the current system. After that, the proposed system, Yolla, is introduced, together with a detailed description of the features and advantages it will have over the current system. It then defines the scope and objectives of the project by providing a comprehensive breakdown of the functional, non-functional, and pseudo requirements. System models, such as use case models, object and class diagrams, and dynamic models, are given to illustrate the technical structure of Yolla. These are accompanied by real-world scenarios that indicate possible applications of the platform. Additionally, navigational paths and screen mock-ups are included to give an idea about the design of user interface and user experience. Finally, the report mentions the broader implications of Yolla: analyzing its impact on users, businesses, and the logistics industry as

a whole; analyzing constraints, risks, standards, and ethical considerations in its design and development.

2. Current System

We follow an agile workflow in our project. The main functionality, which allows users to search for cargo options, is currently operational for 5 cargo companies. We have integrated their pricing systems into our app. Additionally, we have automated the process of locating cargo branches to store them in our database. Users can also track their packages through our system. We have a basic and easy to use frontend that meets the current needs of the application.

3. Proposed System

3.1. Overview

Yolla is a mobile and web-based cargo marketplace designed to simplify the process of finding, comparing, and booking cargo services in Türkiye. Users can easily search for cargo options based on location, size, weight, and delivery time while comparing prices and services from multiple providers. The platform offers comprehensive service provider comparisons using various filters, real-time package tracking, and notifications, ensuring a smooth and user-friendly experience. With its responsive design, Yolla targets individual customers and SMEs by offering transparency, convenience, and efficiency in cargo delivery. Yolla aims to make cargo logistics accessible to everyone at the best price.

Yolla brings a user-friendly interface and serves as a unified marketplace for various cargo companies. This project not only enables individual users to send their packages effortlessly but also supports small to medium-sized businesses in managing and selling their products more efficiently. The primary goal of our system is to provide an easy-to-use interface while delivering all the essential functionalities our users need.

To achieve this, Yolla uses React for the frontend and FastAPI for the backend. The platform utilizes a database system to manage large volumes of data and is designed to scale for thousands of users. It integrates with third-party APIs for cargo providers.

3.2. Functional Requirements

3.2.1. Main Features

3.2.1.1. User Registration and Authentication

- Users can create an account and log in using an email address, phone number, or third-party authentication services like Google, Microsoft, and Apple.
- Senders can save their favorite delivery addresses for future use.

3.2.1.2. Cargo Listing & Search

- Users can input the source and destination addresses either manually or by selecting them on a map.
- According to these provided addresses, cargo options will be displayed along with their delivery times and prices.
- Users can choose the best option based on their preferences.
- The users can list and search cargo services even if they are not registered.
- Users can compare the prices of different cargo options. There will be filtering and sorting options for users to list the cargo services according to their preferences.
- Users can filter the options by the estimated delivery time. Users will be able to select express, standard, or economy delivery options depending on how quickly they need the cargo to reach its destination.
- Users will be able to set a price range for the cargo service to find options that fit within their budget.
- Users can also sort the list of cargo services by delivery time, price, reviews or distance they will have to deliver the cargo to the related cargo branch.

3.2.1.3. Package Tracking

- Both the sender and the receiver can track their packages using a unique key provided to them. If both the sender and receiver are registered users, the code will be sent to them via SMS, and it will also be available in the app. If either party is not a registered user, only the sender will receive the code via SMS and app notification from the cargo company. The sender must be a registered user.

3.2.1.4. Receiving Payments

- The payment will be received through the payment gateway APIs such as SanalPOS and iyzico.

3.2.1.5. Notifications

- The sender and the receiver will be notified by the package's status.

3.2.1.6. Business Cargo Management

- Our platform will be a great medium for small to medium companies. They will be able track all of their sent products and can track if the products were received by the customer. The businesses can get promotions from the cargo services if they send regular cargos at a certain amount. The businesses can agree with certain companies and get promotions from the cargo companies.

3.2.1.7. Box Measurer

- Users will be able to measure their boxes using their phone's camera. By simply pointing the camera at their box, users will determine the box's dimensions. This will make it easier for them to check if their package meets the size requirements.

3.2.1.8. Chatbot Integration

- The application will provide a chatbot service so that the users can ask their questions while using our application.

3.2.2. Secondary Features

3.2.2.1. Ratings & Reviews

- The users will be able to rate and review the cargo services. They can send their commands to the relative cargo service so that other users can benefit from their feedback.

3.2.2.2. Cargo Insurance Options

- Insurance options will be provided for senders, allowing additional coverage to be added to their cargo to ensure protection during transit.

3.2.2.3. Advertisement

- Product and cargo-related advertisements will be displayed on the Yolla platform. Additionally, promotional tariffs will be offered to enhance user engagement and attract more service providers.

3.3. Non-Functional Requirements

3.3.1. Usability

Yolla prioritizes a user-friendly approach in its design across both its mobile and web platforms. The integration of a camera-based cargo box measurement service allows users to quickly determine the dimensions of their packages, eliminating the need for manual calculations. Yolla further enhances convenience by enabling users to compare different cargo service providers effortlessly, with a single click, based on price, delivery time, and other relevant criteria. Additionally, Yolla offers real-time tracking capabilities with one click from the main menu, allowing users to monitor the location and status of their sent or incoming packages with ease.

3.3.2. Reliability

Yolla aims to provide a highly reliable service with a guaranteed uptime of at least 99.9%, excluding scheduled maintenance periods. The platform will ensure data integrity by accurately processing all transactions and data exchanges, preventing data loss or corruption. Robust error-handling mechanisms will be implemented to gracefully manage any system errors, providing meaningful feedback to users without exposing sensitive system information. Regular data backups and efficient recovery procedures will be established, enabling Yolla to restore services promptly—within 15 minutes—in the event of any failures, thereby minimizing disruption to users.

3.3.3. Performance

Optimizing performance is a key priority for Yolla to ensure swift and efficient user interactions. The application will be engineered to load pages and process user requests within three seconds under normal network conditions. Yolla will be capable of handling at least 10,000 concurrent users without any degradation in service quality. Efficient algorithms and data processing methods will be utilized so that search queries and data retrieval operations return results promptly, typically within two seconds. Resource utilization on both server and client devices will be optimized to ensure smooth and efficient operation, contributing to an overall positive user experience.

3.3.4. Supportability

Yolla is designed to provide comprehensive customer support to address user needs effectively and efficiently. The platform will include a 24/7 chatbot service powered by the Gemini API, ensuring users receive immediate assistance at any time. To manage common inquiries, a detailed FAQ page will be available, offering clear answers to frequently asked questions. Additionally, a dedicated support email will be allocated, and checked regularly to ensure timely responses to user concerns. As the platform grows and reaches a substantial user base, a call center will be established to deliver personalized and real-time support, further enhancing the overall user experience. Through these support channels, Yolla aims to maintain high levels of user satisfaction and trust.

3.4.5. Scalability

Yolla is designed to scale efficiently to meet growing user demands and market expansion. The system will support horizontal scalability, allowing for the addition of more servers or resources to handle increased loads seamlessly. Elastic resource management will enable the platform to dynamically allocate computing resources based on current demand, optimizing both performance and operational costs. Yolla is prepared to accommodate a 100% increase in user base and transaction volume within the next 12 months without requiring significant redesign, ensuring continuous, reliable service as the platform grows.

2.2.6. Platform Compatibility

To maximize accessibility, Yolla will be available as both a mobile application and a web application. The mobile app will support major operating systems, including Android and iOS, catering to a wide range of mobile users. The web application will be compatible with all major browsers such as Chrome, Firefox, Safari, and Edge, ensuring users can access Yolla from any device. A responsive design approach will be adopted so that the web application adjusts seamlessly to different screen sizes and resolutions, providing a consistent and optimized user experience across desktops, tablets, and smartphones.

3.4.7. Security

Security is of utmost importance for Yolla to protect user data and maintain trust. All data transmissions will utilize secure protocols (HTTPS) to ensure data integrity and confidentiality during communication. The platform will comply with all relevant data protection laws and regulations, including GDPR and KVKK if applicable, to safeguard personal information. Robust authentication and authorization mechanisms will be implemented, incorporating multi-factor authentication options to enhance account security. Sensitive user data will be encrypted both in transit and at rest using industry-standard encryption algorithms. Regular security audits and vulnerability assessments will be conducted to proactively identify and mitigate potential risks.

3.4.8. Integration

Yolla will integrate with third-party cargo service providers through well-defined APIs, enabling users to access a wide range of services within a single platform. Secure and reliable integration with multiple payment gateways will be established to facilitate smooth and secure transaction processing.

3.4.9. Efficiency

Efficiency is a core focus for Yolla, aiming to optimize resource consumption and performance. The application will be optimized to minimize CPU and memory usage, ensuring efficient operation on both server and client devices. For mobile users, the application will be designed to minimize battery consumption, enhancing usability on the go. Efficient algorithms and data processing methods will be employed to ensure fast execution of tasks, contributing to overall system efficiency. Load balancing techniques will be utilized to distribute workloads evenly across servers, preventing bottlenecks and maintaining consistent performance.

3.4.10. Flexibility

Flexibility is at the core of Yolla's design philosophy for easy adaptation to changing market conditions and user needs. The system architecture will be extensible; new features, services, and integrations with more cargo service providers should be easily added without major overhauls. Users will have the option to customize specific aspects of their experience, such as notification preferences and interface settings, which will enhance personalization and user engagement.

3.4.11. Sustainability

Yolla will serve to United Nations Sustainability goals in few aspects:

3.4.11.1. Protecting Environment

Yolla will be ranking cargo companies based on the carbon efficiency of their fleets, enabling users to make environmentally conscious choices. In addition to that, Yolla will calculate and show the amount of potential carbon emissions for every cargo option, which would

introduce more transparency and raise awareness among users. To further offset this environmental impact, Yolla is working toward collaboration with NGOs committed to carbon neutrality. By means of this partnership, it will be possible for the users to make a donation of rounding up the final value and donating the difference to NGOs, therefore helping to neutralize the carbon emission that their cargo emits. Such steps reduce not only the environmental footprint of the platform but also encourage a sustainability culture within the logistics sector.

3.4.11.2. Supporting Local Economy

Yolla will contribute to the local economy by providing a platform that promotes small and medium-sized cargo businesses together with larger providers. By allowing fair visibility and ranking opportunities, Yolla enables local businesses to compete on an equal footing, promoting diversity and innovation in the logistics market. Besides, the platform nudges the user toward opting for local service providers, thus reducing transport distances and benefiting the regional economic growth of businesses. It will also give special offers to the SMEs. Yolla will grant special price packages, tailored promotions, and priority visibility for the SMEs.

3.4. Pseudo Requirements

The pseudo requirements of this project are coming from expectations of end-users and cargo providers. The platform must be easily usable, even by the new users. The platform must be trustable and reliable. The other pseudo requirement may be compliance with industrial standards, cultural expectations, and competitors that will make the platform relevant and competitive. Furthermore, the system is supposed to give appropriate messages with some feedback mechanisms, though they are not clearly outlined to ensure there is a good experience for users. Some other widely assumed requirements include no hidden charges, responsiveness, and subtle advertising. These requirements, though pseudo and not documented, are very important in meeting the expectations that are crucial for the success of the platform.

In addition to these, there are some pseudo requirements for our project management:

- Git and GitHub will be used for version management and control [1, 2].
- Trello will be used for task assignment and project tracking [3].
- Google Docs will be used to collaborate on technical reports [4].
- Zoom and Google Meet will be used for synchronous communication [5, 6]
- WhatsApp will be used for asynchronous communication [7].
- Figma will be used to design user interfaces [8].

3.5. System Models

3.5.1. Scenarios

Place Order

- Use-case Name: Place Order
- Actor: Registered User, Unregistered User
- Entry Condition: Actor creates a request for price inquiry
- Exit Condition: Order is sent to the relevant company
- Flow of Events:
 - Requested route by actor is checked by API services
 - Offers are sorted according to the various factors specialized for the actor's profile regarding the factors such as price, environment sensitivity, distance to the nearest cargo point
 - Actor chooses one of the offers made by companies
 - A QR code regarding the details of the order sent to the actor
 - This request is anonymized and saved to database for future commercialization of the data
 - System sends the shipping request to the selected company
- Alternative Flow:
 - Requested route by actor is checked by API services
 - No possible offers are found
 - Actor is notified no possible offers are available
 - This request is anonymized and saved to database for future commercialization of the data
- Alternative Flow:
 - Requested route by actor is checked by API services
 - Actor applies a coupon code or a promotion
 - Offers are sorted according to the various factors specialized for the actor's profile regarding the factors such as price, environment sensitivity, distance to the nearest cargo point
 - Actor chooses one of the offers made by companies
 - A QR code regarding the details of the order sent to the actor
 - This request is anonymized and saved to database for future commercialization of the data
 - System sends the shipping request to the selected company, if applicable data about the discount is also sent to company.

Tracking The Non-Related Cargo

- Use-case Name: Tracking The Non-Related Cargo
- Actor: Registered User, Unregistered User, Admin
- Entry Condition: Actor sends a request to inquire cargo status
- Exit Condition: Cargo status is sent to actor
- Flow of Events:
 - Track code of the cargo is provided by actor to the system
 - System recognizes the cargo company by its code format
 - If valid system requests the cargo status from the relevant company by API service
 - Status is sent to the actor

Tracking The Related Cargo

- Use-case Name: Tracking The Related Cargo
- Actor: Registered User, Admin
- Entry Condition: Actor checks related cargos screen
- Exit Condition: Cargo status is sent to actor
- Flow of Events:
 - For each cargo related with the actor:
 - System checks the cargo company from the database
 - System requests the cargo status from the relevant company by API service
 - Status is sent to user
- Alternative Flow:
 - Actor receives a notification about the cargo status

Order Confirmation

- Use-case Name: Order Confirmation
- Actor: Cargo Service Provider
- Entry Condition: Customer provides the order code for open order
- Exit Condition: Admin confirms order
- Flow of Events:
 - Customer (Registered User or Unregistered User) provides the code either in QR or numeric format to the admin
 - Cargo Service Provider double checks the information of the cargo
 - Admin confirms the offers and it is converted into an accepted order
- Alternative Flow:
 - Customer provides the code either in QR or numeric format to the admin
 - Cargo Service Provider double checks the information of the cargo
 - Cargo Service Provider confirms edits the order and price
 - Customer confirms the order

User Support

- Use-case Name: User Support
- Actor: Everyone
- Entry Condition: Anyone requests a support
- Exit Condition: Ticket is tagged as complete
- Flow of Events:
 - Actor submits a request to customer support
 - System recommends few FAQ similar to the submitted question
 - Actor will tag the ticket as open
 - Ticket will be directed to customer service
 - Until ticket is closed:
 - Customer service can add message to ticket

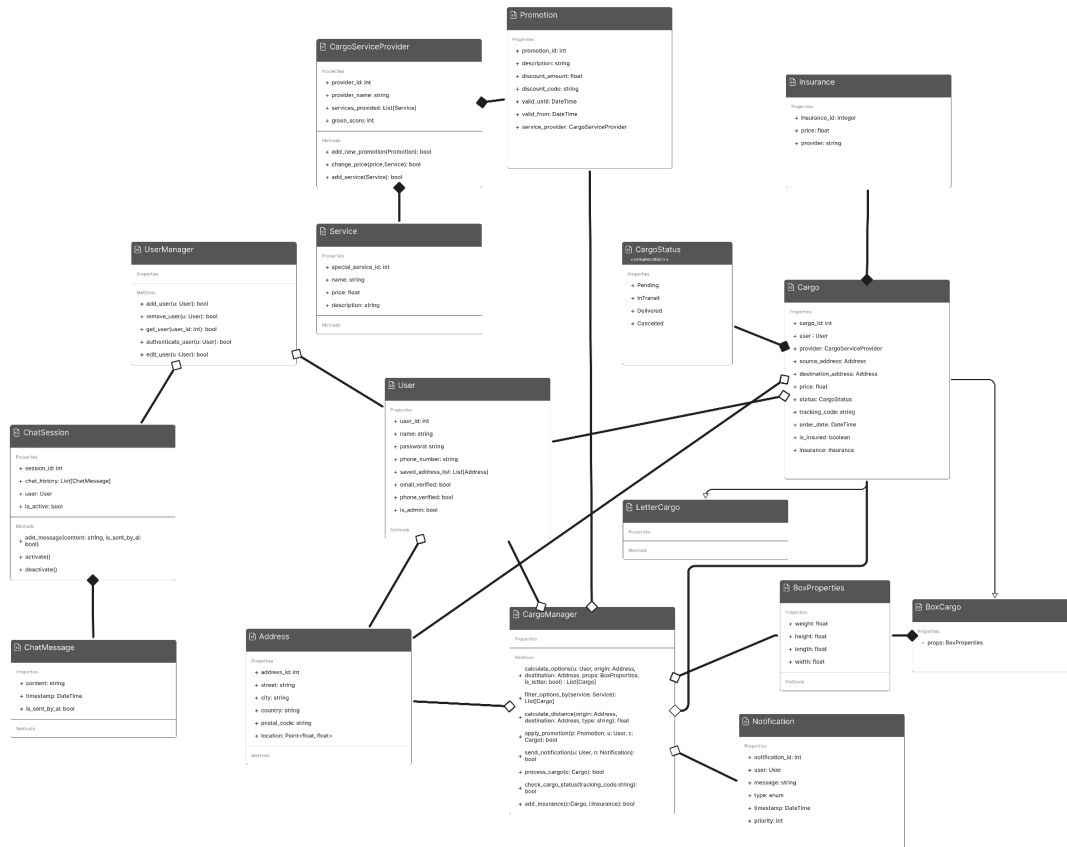
- Actor can add message to ticket
- Ticket will be tagged as closed
- Alternative Flow:
 - Actor submits a request to customer support
 - System recommends few FAQ similar to the submitted question
 - Actor will tag the ticket as closed

3.5.2. Use Case Model

Yolla has the below use case diagram. Main actors are registered and unregistered users where an unregistered user is able to use all the searching and comparison, price calculation, packet tracking capabilities. Registered user on the other hand, in addition to an unregistered user, receives notifications about his/her cargo, sends packages and redeems the promo coupons. Admin has the authorities to manage the system. And various outer actors are provide functionalities such as LLM API, Insurance companies API's, cargo service provider API's and customer service call center.



3.5.3. Object and Class Model



3.5.3.1. User

- Attributes:
 - user_id (int)
 - name (string)
 - phone_number (string)
 - email_address (string)
 - password (string)
 - phone_verified (bool)
 - email_verified (bool)
 - is_admin (bool)
- Relationships:
 - Linked to **Cargo** (1-to-Many)
 - Linked to **Notification** (1-to-Many)

3.5.3.2. Cargo

- Attributes:
 - cargo_id (int)
 - sender (User)
 - provider (CargoServiceProvider)
 - sender_address (Address)
 - receiver_address (Address)
 - status (CargoStatus)
 - creation_date (DateTime)
 - insurance (Insurance)
- Relationships:
 - Linked to **User** (sender)
 - Linked to **CargoServiceProvider**
 - Has **Address** for sender and receiver
 - Has **CargoStatus**
 - Can have **Insurance**

3.5.3.3. CargoServiceProvider

- Attributes:
 - provider_id (int)
 - provider_name (string)
 - service_supported (List[Service])
 - promo_code (string)
- Relationships:
 - Can have multiple **Service**

3.5.3.4. Promotion

- Attributes:
 - promotion_id (int)
 - description (string)
 - discount_amount (float)
 - valid_from (DateTime)
 - valid_till (DateTime)
 - service_provider (CargoServiceProvider)
- Relationships:
 - Associated with **CargoServiceProvider**

3.5.3.5. Address

- Attributes:

- address_id (int)
- street (string)
- city (string)
- country (string)
- postal_code (string)
- Relationships:
 - Used in **Cargo** (sender and receiver addresses)

3.5.3.6. Service

- Attributes:
 - service_id (int)
 - name (string)
 - price (float)
 - description (string)

3.5.3.7. CargoStatus

- Enumeration:
 - Pending
 - Shipped
 - Delivered
 - Cancelled

3.5.3.8. Insurance

- Attributes:
 - insurance_id (int)
 - price (float)
 - provider (string)

3.5.3.9. ChatSession

- Attributes:
 - session_id (int)
 - chat_history (List[ChatMessage])
 - User (linked user)
 - is_active (bool)
- Methods:
 - add_message()
 - end_session()
 - activate()
 - deactivate()

3.5.3.10. ChatMessage

- Attributes:
 - content (string)
 - timestamp (DateTime)
 - is_read (bool)

3.5.3.11. UserManager

- Functions:
 - add_user (u:User) bool
 - remove_user (u:User) bool
 - edit_user (u:User) bool
 - authenticate_user(u:User) bool
- Relationships:
 - Associated with managing **User**

3.5.3.12. CargoManager

- Functions:
 - calculate_options(u: User, origin: Address, destination: Address, props: BoxProperties, is_letter: bool) : List[Cargo]
 - filter_options_by(service: Service): List[Cargo]
 - calculate_distance(origin: Address, destination: Address, type: string): float
 - apply_promotion(p: Promotion, u: User, c: Cargo): bool
 - send_notification(u: User, n: Notification): bool
 - process_cargo(c: Cargo): bool
 - check_cargo_status(tracking_code:string): bool
 - add_insurance(c:Cargo, i:Insurance): bool
- Relationships:
 - Manages **Cargo** operations
 - Sends **Notification**

3.5.3.13. Notification

- Attributes:
 - notification_id (int)
 - user (User)
 - message (string)
 - timestamp (DateTime)
 - priority (int)

- Relationships:
 - Linked to **User**

3.5.3.14. **BoxCargo**

- Attributes:
 - props (BoxProperties)

3.5.3.15. **BoxProperties**

- Attributes:
 - height (float)
 - width (float)
 - length (float)
 - weight (float)

3.5.3.16. **LetterCargo**

- Placeholder class, no attributes specified.

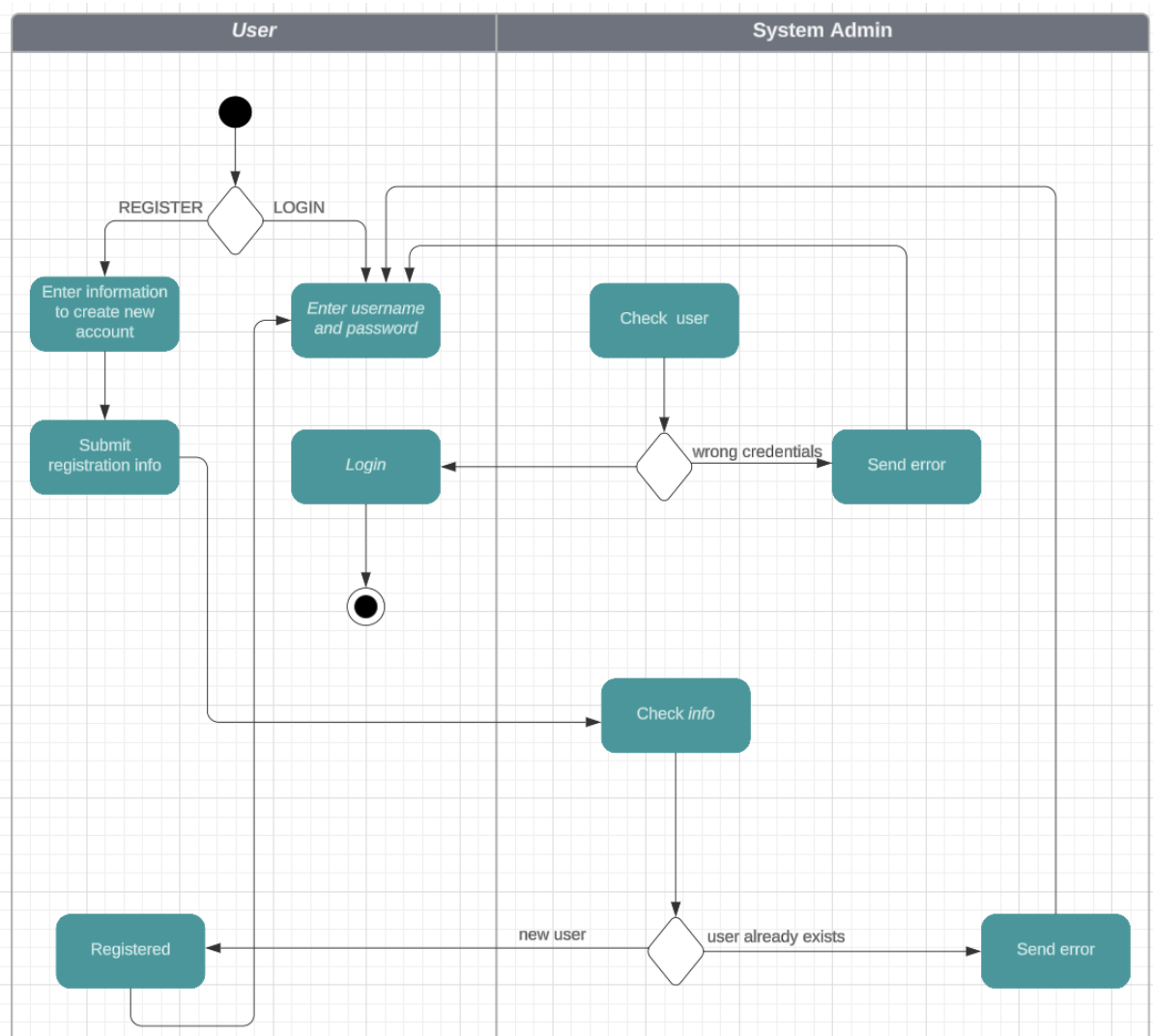
Relationships between some classes:

- **User and Cargo**: A user sends cargo (1-to-many relationship).
- **Cargo and Address**: Cargo includes sender and receiver addresses.
- **Cargo and CargoServiceProvider**: Each cargo has a service provider.
- **Cargo and Insurance**: Cargo can be insured.
- **CargoManager**: Manages the **Cargo** processing and notifications.
- **ChatSession and User**: Each chat session belongs to a user.
- **ChatSession and ChatMessage**: Chat session contains multiple messages.
- **Notification and User**: Notifications are tied to specific users.
- **BoxCargo and BoxProperties**: Box cargo details include dimensions and weight.

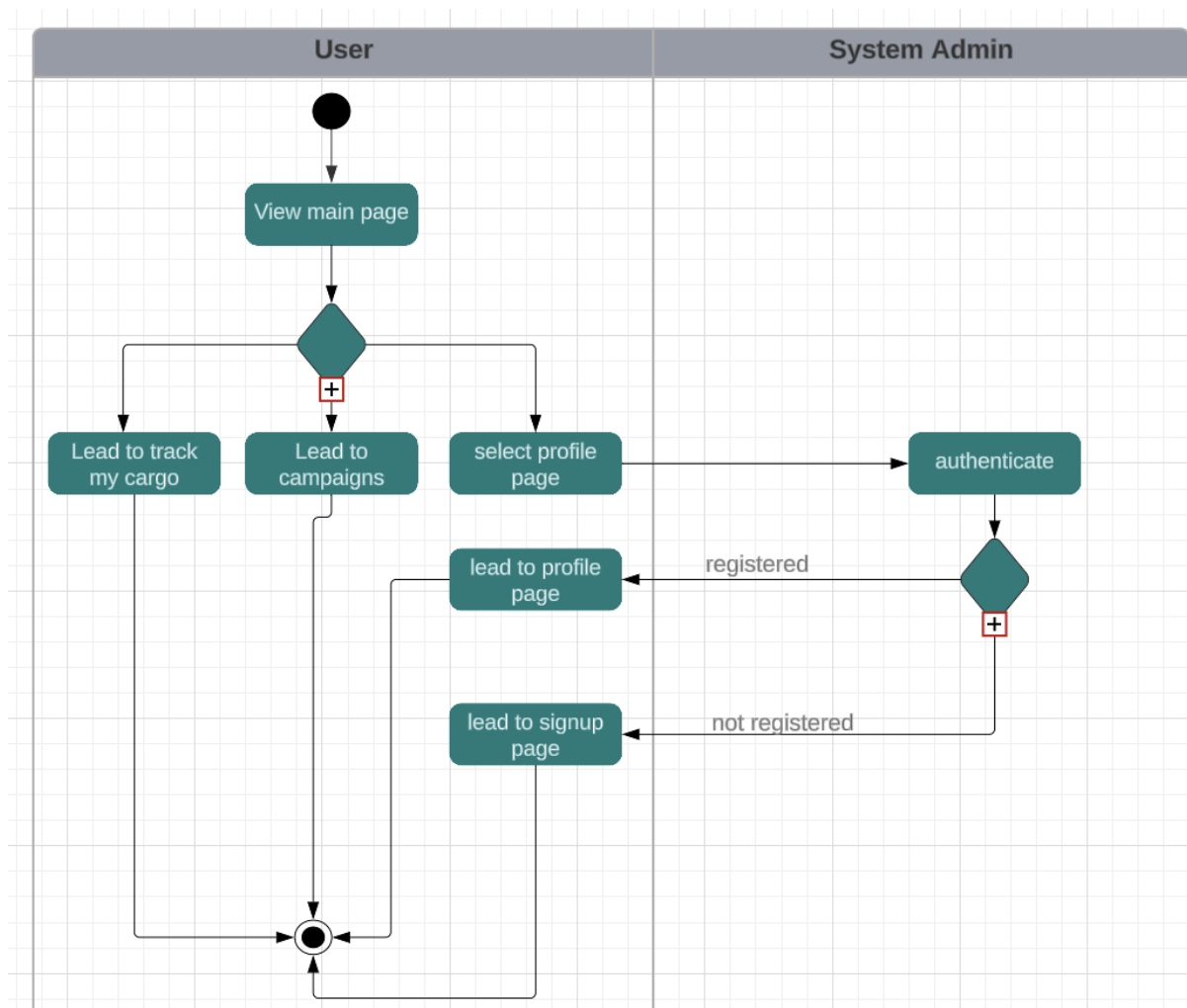
3.5.4. Dynamic Models

3.5.5.1 Activity Diagrams

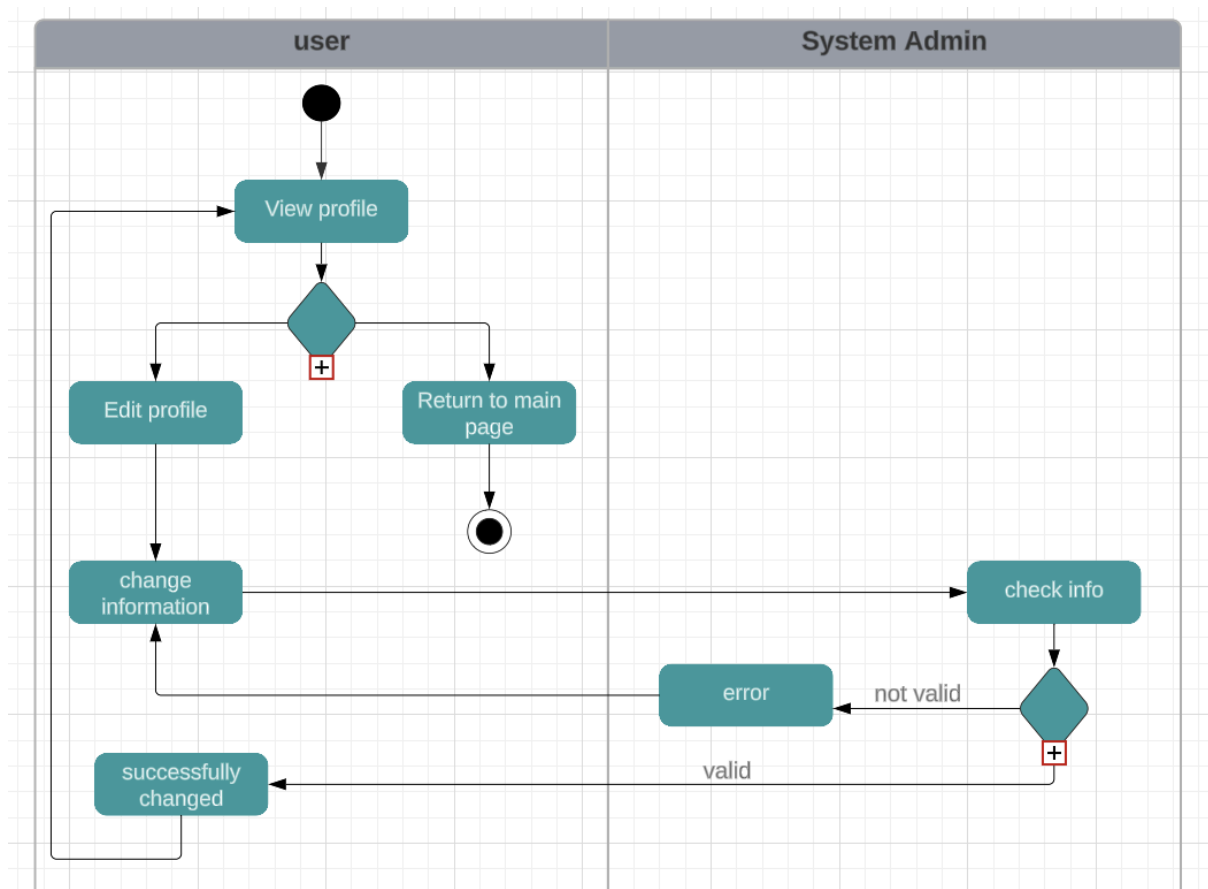
3.5.5.1.1 Login/Register



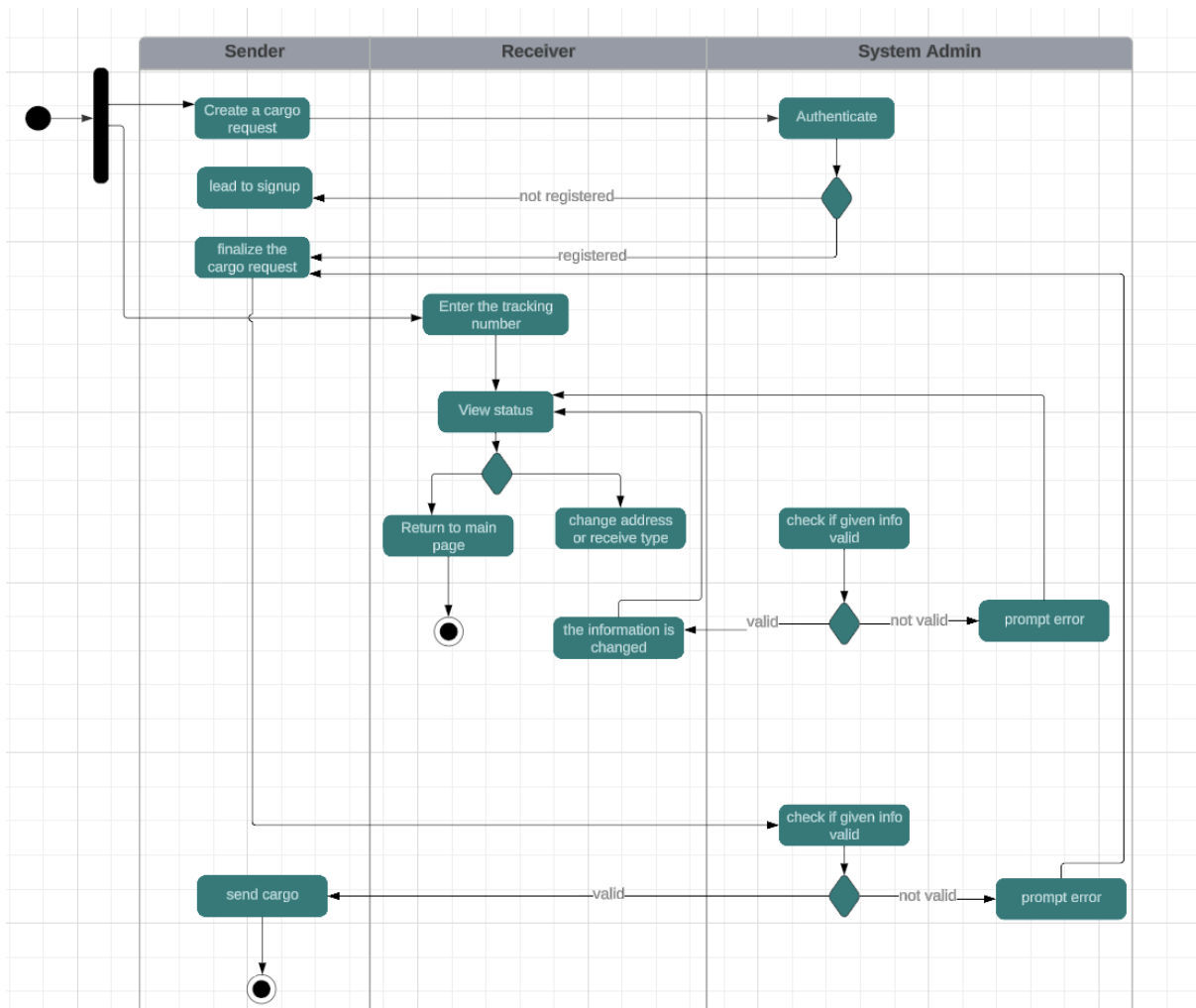
3.5.5.1.2 View Page



3.5.5.1.3 Profile



3.5.5.1.4 Cargo Send/Receive Operations

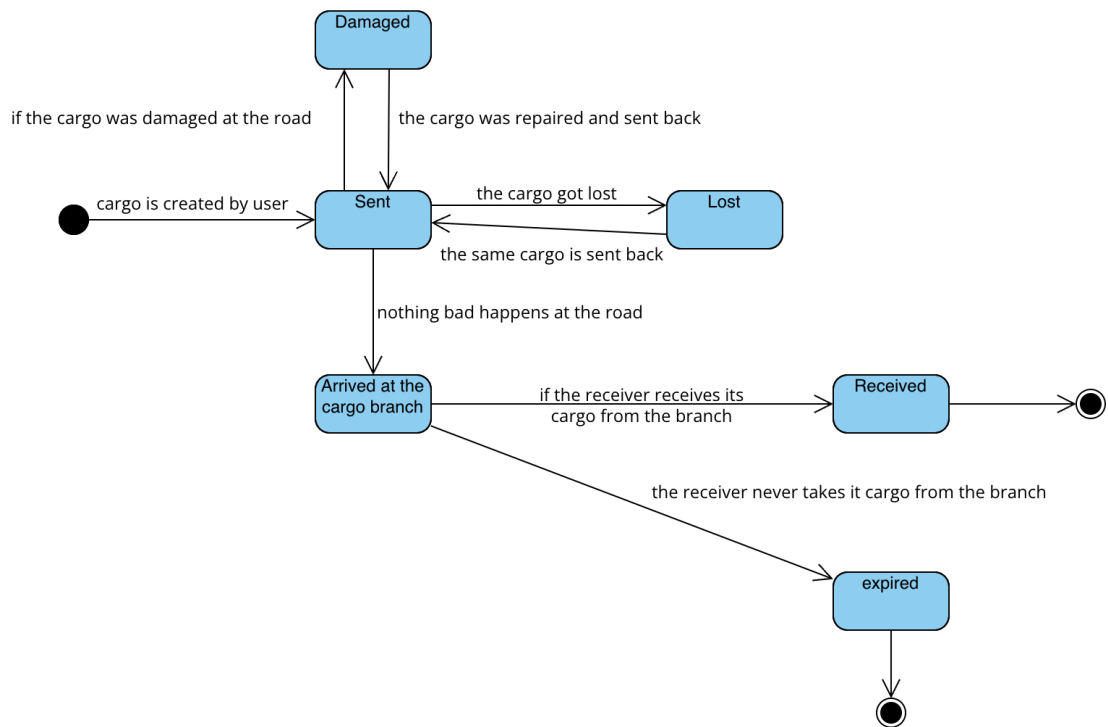


3.5.5.2 State Diagrams

3.5.5.2.1 User

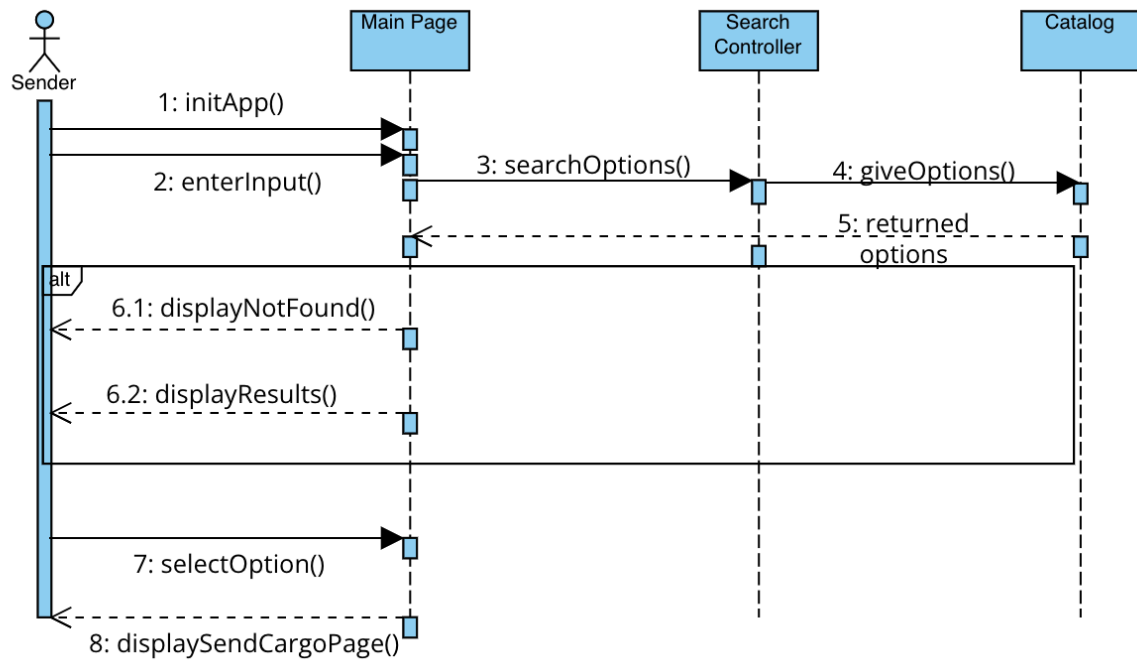


3.5.5.2.2 Cargo

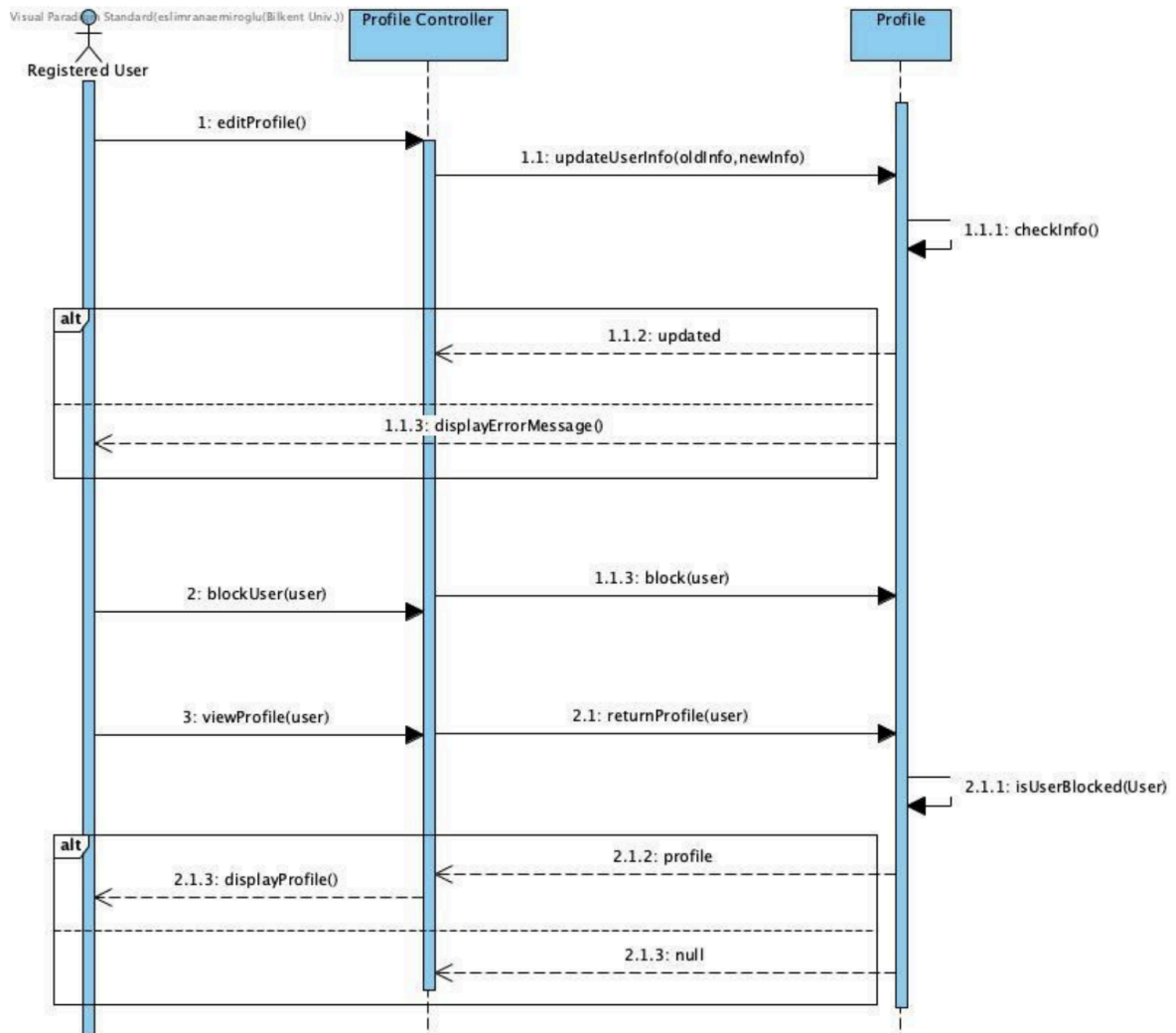


3.5.5.3 Sequence Diagrams

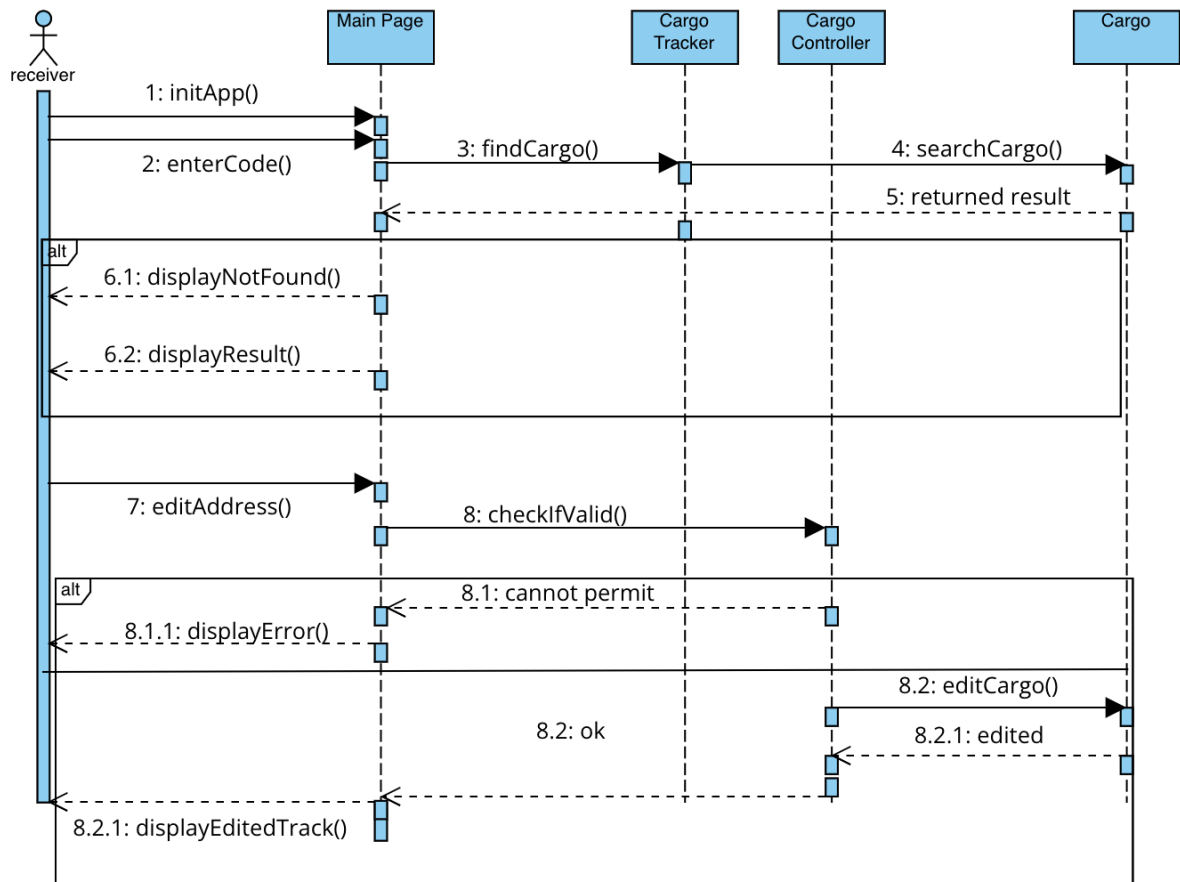
3.5.5.3.1 Search Sequence Diagram



3.5.5.3.2 Profile Sequence Diagram



3.5.5.3.3 Receive Sequence Diagram



3.5.5. User Interface - Navigational Paths and Screen Mock-ups

3.5.5.1 Mobile Application



Screen for settings, support and cargo tracking

Selam 🖐️ Nereye yollayalım?

Yollamak için adres gir

Kargolarım



YY4ED

Teslim edildi!

Teslim eden: Ali K.



L53FVD

Adana Transfer Merkezi'nde

Tahmini teslimat: 2 Aralık



F2R4GDC

Yolladın!

Beşiktaş, İstanbul - 4 gün içinde

Kampanyalar

Öğrencilere
%20 indirim!
MNG Kargo ile okula dönüş başladı!
16 Ekim'e kadar indirimli fırsatı kaçıрма!

İstanbul içi
10% indirimli!
İstanbul içi gönderilerde cazip fırsat!
10% indirimden faydalanmayı unutma!

İlk Kargo Yarış
İlk kargonu Aras ile gönder! Yarısı bizi kazanır!
Bu fırsatı 30 Ekim'e kadar kaçıрма!

Hesabım



8 kere yolladın

210₺ kâr ettin

2 kg CO₂ kurtardın

Halil Tataroğlu 🖐️

Hesap Ayarları ⚙️

Kuponlarım 📄

Davet Et 🎁

Destek

Bir sorun mu var? Her zaman yanındayız.

Sorun nedir, kısaca bahseder misin?

Screen for settings, support and cargo tracking

Selam 🖐️ Nereye yollayalım?

Bilkent Üniversitesi, Ankara

Koç Üniversitesi, İstanbul

Bu bir kutu mu yoksa bir zarf mı?

Emin değilsen [tıkla](#)! Her zaman yanındayız.

Kutu 📦

Zarf ✉️

Nasıl bir kutudan bahsediyoruz?

Ölçmek istersen [tıkla](#)! Kamera ile ölçebilirsin.



Desi:

0.4

En

10 cm

Boy

8 cm

Yükseklik

15 cm

Ağırlık

1.4 kg

Bizim için özelsin 😊

Ekstra hizmetler için [tıkla](#)

Tekliflerimiz



En yeşil

Bu teslimatının karbon salınımı SIFIR!

54.5 ₺

3-5 gün

0 gr CO₂



En ucuz

En yakın tekliften 10.45 ₺ daha ucuz !

21 ₺

6-7 gün

5 gr CO₂



En hızlı

En geç 16 Aralık'ta teslimat

43 ₺

1-3 gün

8 gr CO₂

Search screen, available offers are shown here

3.5.5.2 Web Application

YOLLA

Selam 🖐️ Nereye yollayalım?

Nereden 📍

↔️

Nereye 📍

Gönderi Türü

☒ Kutu ☐ Zarf

Ağırlık _____

En/Boy/Yükseklik _____

ARA



Biz Kimiz?



Home screen, users can search cargo options for the entered source and destination

Ordu



Malatya



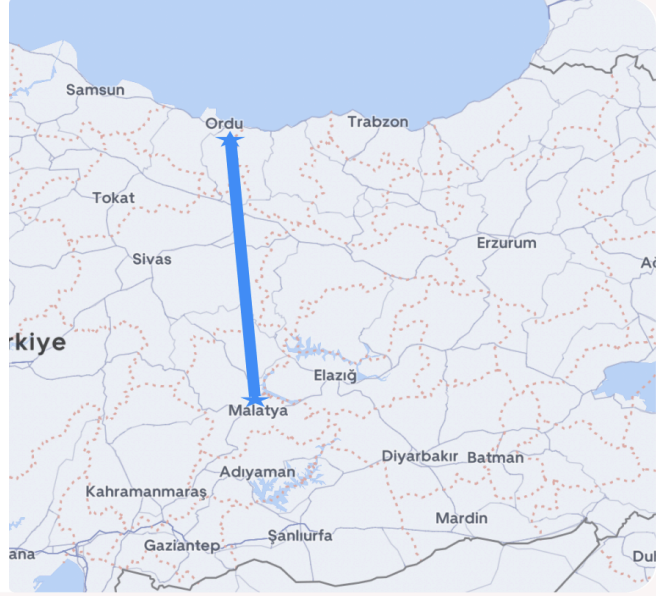
Tekliflerimiz

- **En yeŖil!**
Bu teslimatın karbon salınımı SIFIR!
54.5 ₺
3-5 gün
0 gr CO₂
- **En ucuz**
En yakın tekliften 10.45 ₺ daha ucuz!
21 ₺
6-7 gün
6 gr CO₂
- **En hızlı**
En ge 16 Aralık'ta teslimat
43 ₺
1-3 gün
8 gr CO₂

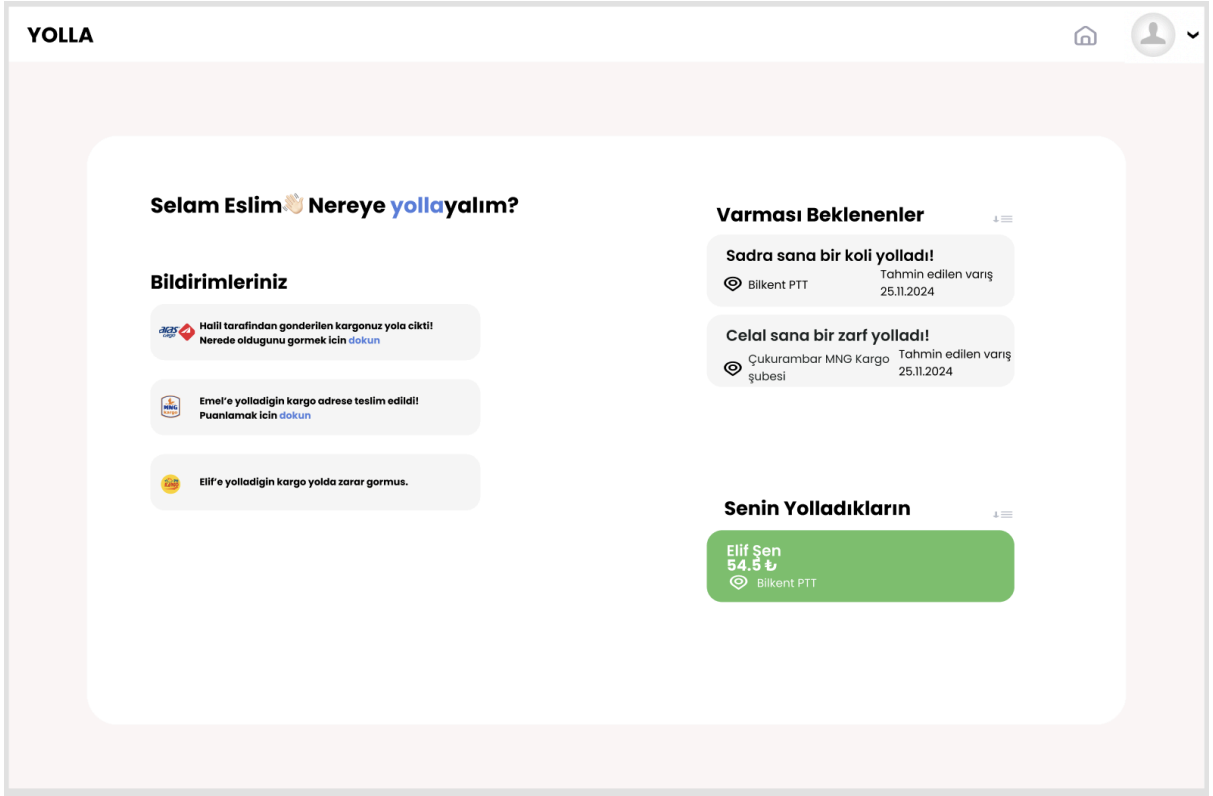
Arama Sonuclari




- Surat Kargo**
54.5 ₺
 En yakın sube: 1km 3-5 gün
- Aras Kargo**
21 ₺
 En yakın sube: 200 m 1-3 gün
- UPS**
43 ₺
 En yakın sube: 3 km 6-7 gün



Search Results, this page is shown after the user presses “ARA” button on the home page



Profile Page, this page is shown after the user presses Profile button on the navigation bar

YOLLA

Detayları gir 🙌 biz yollayalım?

İsim

Eslim

Soyisim

Emiroğlu

Adres

Yasamkent mah. 18No. 9

Kullanıcı Adı

eslimrana

[Şifre Değiştir](#)

KAYDET

Edit Profile Page, this page is shown after the user presses Profile>Edit Profile button on the navigation bar

4. Other Analysis Elements

In this part of the report, dimension and possible risk that may be encountered will be discussed.

4.1. Consideration of Various Factors in Engineering Design

4.1.1. Constraints

Yolla Tech's engineering design takes into account a number of limitations to make sure the platform is viable, usable, and sustainable. These are essential to providing a user-friendly cargo marketplace app that satisfies consumer demands while resolving social, legal, and technical issues.

1. Accessibility

The application must ensure user-friendly navigation and functioning to accommodate users with different degrees of technical literacy. It should comply with Web Content Accessibility Guidelines (WCAG) 2.1 requirements for web accessibility and offer mobile accessibility

features, such as screen reader compatibility for visually impaired users. To increase its diversity and reach, the app must also support many languages, such as English and Turkish.

2. Aesthetics

For the platform to work flawlessly on desktop and mobile devices, it must have a contemporary, tidy, and responsive design. Important factors include improving the user interface (UI) with high contrast designs and readable fonts to increase accessibility and readability, and optimizing the user experience (UX) by simplifying the process for looking for, comparing, and scheduling cargo services.

3. Cost

In order to solve infrastructure and user affordability, cost restrictions need to be carefully handled. Using open-source technologies like React, FastAPI, and PostgreSQL helps reduce development costs. The software should continue to be free to use, guaranteeing that users are not charged extra, in order to increase user attractiveness.

4. Legal Considerations

The platform's operation depends on adherence to both domestic and international regulations. Obtaining user consent for data collection and offering options for data deletion are two ways to comply with GDPR's standards for protecting user data. To guarantee complete legal compliance, the platform must also abide by Turkey's e-commerce and data protection laws.

5. Functionality

Both essential and optional features should work flawlessly in the app, guaranteeing a quick and responsive user experience. This entails keeping high availability and dependability during periods of high demand as well as managing user inquiries, cargo searches, and package tracking with the least amount of latency.

6. Sustainability

One of the top priorities is reducing the environmental impact of cloud infrastructure and hosting. This can be accomplished by choosing cloud hosting companies like AWS or Google Cloud that have made promises to renewable energy. The app should also encourage users to switch from paper to digital receipts and documents in order to improve sustainability.

7. Marketability

The software needs to be able to efficiently promote to small and medium-sized businesses (SMEs) as well as individual consumers. Building user trust requires competent and

transparent branding. Furthermore, features like insurance choices and real-time tracking can help set the app apart from rivals and draw in more users.

8. Interoperability

For the app to improve functionality and user experience, third-party services must be smoothly integrated. To provide seamless and effective operations, this involves connecting payment gateways for safe transactions and APIs for real-time package tracking with cargo suppliers.

9. Technical Specifications

To guarantee compatibility and performance, important limitations determine the hardware and software specifications. While the web app must be completely compatible with contemporary browsers like Chrome, Firefox, Safari, and Edge, the mobile app must function flawlessly on devices running iOS 12+ and Android 8.0+.

10. Social, Cultural, and Economic Factors

For the app to be inclusive and widely used, social, cultural, and economic considerations must be taken into consideration. Socially, no demographic group should be excluded from using the app, and it should continue to be free to use. Culturally speaking, it should enable localized material, like cargo delivery customs in Turkey, and provide adaptable interfaces to meet specific user requirements, like extra fields for regional address forms. For the app to be competitive and desirable in the market, cost-efficiency for both individual users and SMEs must be given top priority.

4.1.2. Standards

Application of Standards in Yolla Tech

Category	Standard	Purpose
Software Development	IEEE 830, ISO/IEC 12207	Define clear requirements and lifecycle processes.
Modeling and Design	UML 2.5.1, IEEE 1016	Create structured visual and textual representations of the system architecture.
Accessibility and UI	WCAG 2.1, ISO 9241	Ensure usability and inclusivity for all users, including those with disabilities.
Security	OWASP, ISO/IEC 27001, PCI DSS	Protect user data, secure transactions, and mitigate vulnerabilities.
Data Protection and Privacy	GDPR, ISO/IEC 27701	Handle personal data securely and comply with privacy laws.

Quality Assurance	ISO 25010, IEEE 29119	Deliver a reliable, maintainable, and high-performing system.
Collaboration	Git Workflow, Semantic Versioning	Maintain organized development and release cycles.
Deployment and Hosting	Docker, ISO/IEC 20000	Ensure consistent deployment environments and service delivery.
API Standards	RESTful API, OpenAPI Specification	Foster easy integration with third-party services and cargo providers.
Localization	Unicode, ISO 639-1	Support seamless multilingual experiences for users in Türkiye and beyond.

By adhering to these standards, Yolla guarantees a robust, user-friendly, and compliant platform, meeting the expectations of users, developers, and regulatory authorities.

4.2. Risks and Alternatives

The development of the Yolla Tech cargo marketplace involves numerous risks that could impact its success and operational efficiency. Identifying these risks early and proposing actionable alternatives is crucial to ensure the project's sustainability and resilience.

1. Server Overload During Peak Hours

There is a chance of server outages, sluggish response times, or crashes when there is a lot of traffic, including during holidays or promotional events. This has a medium chance of happening, but it will have a significant impact. The application should use cloud services like AWS Auto Scaling or Google Cloud Compute Engine to implement auto-scaling in order to dynamically distribute resources in response to real-time demand, thereby reducing this risk. While improving database queries and backend code guarantees effective resource use, load balancers can be utilized to divide traffic evenly among servers, avoiding bottlenecks. If auto-scaling turns out to be too expensive, edge computing can be used as a backup strategy to reduce the strain on central servers by caching frequently accessed data at local servers near users.

2. Insufficient User Adoption

Low engagement and monetary losses could arise from the app's inability to draw in enough users. This risk, which has a medium impact and a high possibility, can be reduced by starting focused marketing initiatives and establishing alliances with nearby companies and SMEs. Providing incentives, such referral schemes or first-time user discounts, can help promote user acquisition even more. Additionally, increasing user retention and happiness will be facilitated by quickly iterating the app and collecting early user input. As a backup

strategy, the app can switch to a business-to-business (B2B) model and offer customized freight logistics capabilities to assist companies in streamlining their processes.

3. Data Breach

A major danger with a low probability but a high impact is the exposure of user data, including payment and personal information, as a result of insufficient security measures. The application should use role-based access control (RBAC) to limit access to sensitive data, encrypt data for transmission and storage, and perform frequent security audits and penetration tests to find and fix flaws in order to reduce this risk. As a backup strategy, the app should immediately hire cybersecurity professionals to limit the harm, alert impacted users, and publish updates to fix any vulnerabilities found.

4. High Server Costs

There is a medium chance and significant impact of going over budget due to high operating costs for server and infrastructure maintenance. Utilizing affordable cloud providers or negotiating long-term agreements with hosting companies for reduced prices are two ways the app might lessen this. Pay-as-you-go models guarantee that resources are only paid for when they are really used. Additionally, costs can be greatly decreased by streamlining database queries, cutting down on pointless calculations, and lowering API call costs. An alternate strategy is to reduce the strain on server infrastructure by offloading non-essential calculations to client-side processing when practical.

5. Regulatory Non-compliance

With a medium likelihood and a high impact, noncompliance with GDPR, PCI DSS, and e-commerce regulations unique to Turkey presents a serious danger of fines or app suspension. In order to address this, the app should be subjected to extensive evaluations of legal and regulatory requirements throughout the design stage. Legal consultants should also be involved to guarantee compliance with both local and international regulations. Compliance will be further supported by the implementation of standardized frameworks, such as GDPR-compliant user consent processes. As a backup strategy, the software can quickly update to comply with legal requirements while temporarily disabling impacted functionalities if compliance problems occur.

6. Loss of Key Team Members

The project may be delayed if a team member leaves during crucial stages of development; this is unlikely but could have a moderate effect. Cross-training should be used to make sure all team members are knowledgeable about different project components in order to reduce this risk. Keeping thorough records of procedures, including code comments and technical designs, will also help to ensure continuity. An alternate strategy would be to reassign the departing member's duties to the remaining members appropriately.

7. Technical Bugs and Failures

Although unusual, a team member leaving during critical phases of development could have a moderate impact on the project's timeline. To lower this risk, cross-training should be performed to ensure that all team members are informed about various project components. Maintaining comprehensive documentation of processes, including technical designs and code comments, will also aid in ensuring continuity. Redistributing the departing member's responsibilities to the remaining members in a suitable manner would be an alternative tactic.

8. Competition from Established Platforms

With a high chance and medium impact, entering a sector that is dominated by well-established logistics platforms is a substantial task. The app should concentrate on differentiating itself by providing special features like real-time tracking, AR-based box measuring, and competitive pricing in order to reduce this risk. Collaborating with regional suppliers can help increase confidence and ease market entry. As a backup strategy, the app might use niche targeting to gain traction by concentrating on underdeveloped areas like regional logistics or SMEs.

4.3. Project Plan

WP#	Work package title	Leader	Members involved
WP1	Backend	Sadra	Halil, Elif, Eslim
WP2	Frontend	Eslim	Sadra, Halil, Celal
WP3	Business	Halil	Elif, Eslim
WP4	DevOps	Celal	Sadra
WP5	Marketing	Halil	Elif, Celal
WP6	Reports	Eslim	Sadra, Halil, Elif, Celal

WP 1: Backend			
Start date: September 2024 End date: December 2024			
Leader:	Sadra M. A.	Members involved:	Halil Tataroğlu Elif Şen Eslim Ranaoğlu
Objectives: The backend in Yolla Tech is responsible for handling the core logic and operations of the platform. It facilitates seamless communication between the frontend and the database, ensuring secure data storage, retrieval, and management. Key objectives include processing user requests, managing cargo-related APIs, enabling real-time package tracking, and providing robust authentication and authorization mechanisms. Additionally, it ensures system scalability, reliability, and security through efficient API design and server optimization.			

Tasks:

Task 1.1 Price Search: Price search takes the user's input on the locations the user is sending their package from and to alongside the necessary filters and returns the most optimal results based on the user's wants.

Task 1.2 Authentication: Develop user authentication mechanisms, including registration, login, and password recovery.

Task 1.3 Package Tracking: The packages of the users will be tracked based on the user orders.

Task 1.4 Notifications: Notifications will be sent on important updates.

Task 1.5 API Integrations: Integrate the API's of the respective cargo systems into the system.

Task 1.6 Distance: Find the distances of the cargo branches respective to the given location.

Task 1.7 Database: Integrate Postgres as the chosen database and design the database schema.

Deliverables

D1.1: Backend Server (FastAPI)

WP 2: Frontend

Start date: September 2024 **End date:** December 2024

Leader:

Eslim Ranaoğlu

Members involved:

Halil Tataroğlu
Celal Salih Türkmen
Sadra M. A.

Objectives: The frontend in Yolla Tech is designed to deliver an intuitive and user-friendly interface for seamless interaction with the platform. Its primary objectives are to display cargo services, enable users to search, compare, and book services, and provide real-time tracking updates. It ensures responsiveness across devices, supporting both web and mobile users. Additionally, the frontend integrates features like notifications, social login, and multilingual support to enhance the overall user experience.

Tasks:

Task 2.1 Admin Dashboard: Develop the admin dashboard to provide administrative users with tools to monitor system performance, manage users, and oversee cargo operations. The dashboard should include analytics and controls for effective management.

Task 2.2 Landing Page: Design and implement the landing page to attract users and provide an overview of the platform's features and benefits. This page serves as the first impression and is critical for user conversion.

Task 2.3 Package Tracking: Create a dedicated page for users to track their packages in real time. This task involves integrating tracking APIs and displaying location updates in a user-friendly format.

Task 2.4 Main Page: Develop the main user interface for interacting with the platform, including cargo search, comparisons, and bookings. It should be intuitive and responsive across devices.

Task 2.5 Web Page: Implement the project's official webpage to provide information about the platform, team, and contact details. This page supports credibility and communication with users.

Task 2.6 Figma Design: Use Figma to create and iterate on the UI/UX designs for all application pages. This ensures a consistent and visually appealing user experience.

Task 2.7 Social Login: Enable users to log in using their social media accounts, such as Google or Facebook. This simplifies the onboarding process and improves user convenience.

Task 2.8 Login and Registration: Develop the login and registration functionality to allow users to create accounts securely and manage their credentials. Includes features like password recovery and validation.

Task 2.9 MVP Page: Create a minimal viable product (MVP) page to demonstrate core features and gather initial user feedback. This page will focus on essential functionalities for testing and refinement.

Deliverables

D2.1: Figma Designs

D2.2: React App

D2.3: MVP App

WP 3: Business			
Start date: September 2024 End date: December 2024			
Leader:	Halil Tataroğlu	Members involved:	Elif Şen Eslim Ranaoğlu
<p>Objectives: The business side focuses on establishing and maintaining relationships with cargo service providers and ensuring the platform aligns with market needs. Its primary objectives include onboarding cargo companies, negotiating partnerships, and managing promotional collaborations. It also involves conducting market and competitor analyses to identify trends and optimize the platform's offerings. Additionally, the business side ensures effective task management, resource preparation, and maintaining communications with academic advisors and stakeholders to guide the project strategically.</p>			
<p>Tasks:</p> <p>Task 3.1 Cargo Service Relations: Establish and maintain partnerships with cargo service providers to ensure a wide range of delivery options for users. This task involves negotiations, onboarding services, and managing ongoing relationships.</p> <p>Task 3.2 Competitor Analysis: Conduct a detailed analysis of competitors to understand their strengths, weaknesses, and market positioning. This information will be used to identify opportunities for differentiation and improvement.</p> <p>Task 3.3 Market Analysis: Research the market to identify user needs, preferences, and emerging trends. This helps align the platform's features and services with market demands for better adoption.</p> <p>Task 3.4 Task Management: Organize and track project tasks using collaborative tools to ensure efficient progress and timely completion. This includes assigning responsibilities, setting deadlines, and monitoring deliverables.</p> <p>Task 3.5 Resource Preparation: Gather and prepare the necessary resources, such as tools, datasets, and documentation, to support development and business operations. This ensures the project runs smoothly and efficiently.</p> <p>Task 3.6 Academic Relations: Maintain communication with academic advisors to gain guidance and feedback throughout the project. This ensures alignment with academic standards and leverages expert insights.</p>			
<p>Deliverables</p> <p>D3.1: Market Analysis Report</p> <p>D3.2: Competitor Analysis Report</p> <p>D3.3: Google Drive Account</p> <p>D3.4: Trello Board</p>			
WP 4: Devops			
Start date: September 2024 End date: December 2024			
Leader:	Celal Salih Türkmen	Members involved:	Sadra M. A.
<p>Objectives: The DevOps side ensures efficient development, deployment, and operation of the platform through continuous integration and delivery pipelines. Its primary objectives include automating infrastructure setup, managing Docker containers, and implementing scalable hosting solutions. DevOps also focuses on maintaining version control with GitHub CI/CD workflows to streamline updates and deployments. Additionally, it ensures system reliability and performance by monitoring server health and optimizing deployment processes for seamless user experiences.</p>			

Tasks: Task 4.1 Domain: Register and configure the project's domain to establish a unique and professional online presence. This includes setting up DNS records and ensuring domain security for smooth accessibility. Task 4.2 Docker: Containerize the application using Docker to ensure consistent development, testing, and deployment environments. This simplifies the process of scaling and deploying the app across various platforms. Task 4.3 GitHub CI/CD: Set up Continuous Integration and Continuous Deployment (CI/CD) pipelines on GitHub to automate testing, building, and deploying the application. This ensures rapid and reliable delivery of updates. Task 4.4 AWS Hosting: Deploy the platform on AWS, leveraging its scalable and secure hosting infrastructure. This involves setting up instances, configuring server environments, and optimizing costs.			
Deliverables D4.1: yolla.tech domain D4.2: Docker Image D4.3: Github Actions D4.4: AWS hosted app			
WP 5: Marketing			
Start date: September 2024 End date: December 2024			
Leader:	Elif Şen	Members involved:	Halil Tataroğlu Eslim Ranaoğlu Celal Salih Türkmen
Objectives: The marketing focuses on promoting the platform to attract users and build brand awareness. Its primary objectives include managing social media campaigns, creating engaging content, and leveraging digital marketing strategies to reach target audiences. It also involves conducting feasibility studies to assess market potential and align promotional efforts with user needs. Additionally, marketing aims to establish a competitive presence by highlighting the platform's unique features, such as real-time tracking and price comparisons, to drive user adoption and retention.			
Tasks: Task 4.1 Social Media: Manage social media platforms to promote the project, engage with the audience, and build brand awareness. This includes creating content, running campaigns, and responding to user interactions. Task 4.2 School Bulletins: Utilize school bulletins to share project updates and announcements with the academic community. This helps raise awareness, gather feedback, and potentially attract collaborators or supporters.			
Deliverables D4.1: Instagram Page			
WP 6: Reports			
Start date: September 2024 End date: December 2024			
Leader:	Eslim Ranaoğlu	Members involved:	Halil Tataroğlu Elif Şen Celal Salih Türkmen Sadra M. A.
Objectives: The reports aim to document the project's progress, requirements, and outcomes systematically. Their primary objectives include detailing the project's description, specifications, and analysis to ensure alignment with stakeholder expectations. Reports serve as a reference for team collaboration, providing clarity on tasks, milestones, and deliverables. Additionally, they facilitate accountability and transparency by highlighting			

completed work, challenges faced, and plans for future development. This ensures a well-documented foundation for the project's success and future scalability.

Tasks:

Task 5.1 Project Description: <briefly explain the purpose of this task (1-3 sentences)>

Task 5.2 Project Specification: <briefly explain the purpose of this task (1-3 sentences)>

Task 5.3 Analysis and Requirements: <briefly explain the purpose of this task (1-3 sentences)>

Deliverables

Task 5.1 Project Description: Provide a clear and concise overview of the project's objectives, scope, and significance. This serves as an introduction for stakeholders to understand the purpose and goals of the platform.

Task 5.2 Project Specification: Document the detailed specifications of the project, including functional and non-functional requirements, to guide development. This ensures alignment between the team's efforts and the project's objectives.

Task 5.3 Analysis and Requirements: Analyze user needs, market demands, and technical feasibility to define the platform's requirements. This task ensures the system design and features meet the expectations of stakeholders and end-users.

4.4. Ensuring Proper Teamwork

For the Yolla Tech initiative to succeed, effective teamwork is crucial. We hope to guarantee a seamless workflow and the timely completion of project objectives by defining defined responsibilities, encouraging communication, and utilizing collaborative tools. The main techniques, approaches, and resources used to sustain effective teamwork are listed below.

1. Team Roles and Responsibilities

Each team member has been assigned roles based on their expertise and experience. However, cross-functional collaboration ensures everyone understands different aspects of the project.

2. Communication and Collaboration Tools

To streamline communication and collaboration, the following tools are being used:

- **Slack:** Real-time communication for discussions, announcements, and resolving issues.
- **Trello:** For tracking tasks using Kanban boards, organizing sprints, and monitoring task progress.
- **GitHub:** Version control and collaboration on code. Issues are used to document and resolve bugs or feature requests.
- **Google Workspace:** Shared documents, spreadsheets, and presentations for collaborative editing and reporting.
- **Figma:** Collaborative design tool for UI/UX mockups and wireframes.

3. Teamwork Strategies

In order to maintain team alignment, share updates, and resolve roadblocks, the development approach integrates weekly sprints and daily stand-up meetings to guarantee a targeted and manageable workload. Pair programming sessions encourage cooperative problem-solving and information sharing, while GitHub code reviews assist detect possible issues and maintain code quality. Team members gain a comprehensive understanding of the project through rotational leadership roles within tasks, which also give each member the chance to lead particular tasks or modules, boosting their confidence and skill sets. For clarity and ease of tracking, large activities are divided into smaller, more manageable subtasks. Assignments are based on competence while promoting skill growth in other areas. Performance is tracked by monitoring GitHub contributions and commits, while sprint points are evaluated using tools like Trello or Google Sheets to ensure efficiency and identify bottlenecks.

4. Conflict Resolution

Overlapping duties, divergent viewpoints, or missed deadlines can all lead to conflicts, but there are procedures in place to deal with these issues successfully. Team members can express issues during meetings or via Slack because open communication is promoted. By mediating disputes and assisting the group in coming to an agreement, the project manager serves as a mediator. All choices are also recorded to guarantee clarity and avoid misunderstandings in the future.

5. Knowledge Sharing

The team uses efficient knowledge-sharing techniques to guarantee continuity and alignment. Thorough documenting of project requirements, processes, and code guarantees development even in the event that a team member is unavailable and serves as a trustworthy reference. Team members are upskilled in critical tools like Docker, FastAPI, and React Native through workshops and training sessions. Peer learning is also promoted, with team members exchanging perspectives from their specialized fields to create a cooperative and information-rich atmosphere.

6. Accountability and Transparency

Each assignment has a designated task owner, who ensures responsibility and clarity by assigning a responsible person to do the task. Weekly reports outlining finished, ongoing, and forthcoming work are used to track progress; obstacles are found early and dealt with during stand-up meetings. Major project milestones are also monitored to make sure they are in accordance with the project's overall objectives and schedule.

4.5. Ethics and Professional Responsibilities

To guarantee that the result is not only useful and effective but also complies with legal standards, environmental sustainability, and societal values, the Yolla Tech project development entails substantial ethical and professional obligations. Respecting moral

standards is essential to gaining the trust of users, staying out of trouble with the law, and making a constructive contribution to society.

1. User Privacy and Data Protection

Given the app's reliance on sensitive personal and transactional data, protecting user privacy is an ethical imperative. By only collecting user data with express consent and restricting it to what is required for functionality, the app complies fully with the General Data Protection Regulation (GDPR). Users still have total control over their data, including the option to view, edit, or remove it. To maintain security, all user data—especially private information like delivery and payment credentials—is encrypted while it's in transit and at rest. Additionally, user data is never monetized through advertising or unapproved partnerships, and it is only ever shared with third parties for confirmed and necessary integrations like payment gateways.

2. Inclusivity and Accessibility

Because Yolla Tech is committed to inclusivity, users from a wide range of backgrounds can utilize the program, regardless of their socioeconomic level, cultural diversity, or physical impairments. Following WCAG 2.1 (online Content Accessibility Guidelines) prioritizes accessibility compliance by including features like voice-assisted navigation, keyboard accessibility for online users, and screen-reader compatibility. The software is free to download and use in order to encourage economic inclusion, giving people and companies—including SMEs—access to affordable shipping solutions without facing financial obstacles.

3. Environmental Responsibility

Yolla Tech prioritizes sustainability by reducing paper waste and promoting energy efficiency. The app encourages the use of digital receipts and invoices, minimizing reliance on paper-based documentation, while digitalizing business operations for SMEs further reduces paper consumption. Energy efficiency is achieved by selecting hosting services committed to renewable energy sources and optimizing backend operations to minimize computational waste and lower energy consumption.

4. Fairness and Transparency

By encouraging energy efficiency and minimizing paper waste, Yolla Tech demonstrates their commitment to sustainability. While digitizing company activities for SMEs further reduces paper usage, the software promotes the use of digital invoices and receipts, reducing dependency on paper-based documentation. Choosing hosting providers that use renewable energy sources and streamlining backend processes to reduce computational waste and energy usage are two ways to achieve energy efficiency.

5. Professional Conduct

Throughout the project, the Yolla Tech team is committed to maintaining professional standards. Regular updates on project progress are recorded and openly distributed to stakeholders thanks to accurate reporting, which guarantees that all deliverables and reports are thorough, accurate, and devoid of false information. By revealing any affiliations or collaborations that might influence project choices, team members proactively prevent conflicts of interest. Additionally, alliances with third-party services and cargo providers are based on honesty, trust, and equity, which strengthens the group's dedication to moral business conduct.

6. Social Responsibility

The purpose of the app is to positively influence society. Yolla Tech enables small and medium-sized businesses (SMEs) to improve their operations and streamline their logistics by offering a platform for comparing and using freight services. Fair pricing, moral working conditions, and responsible data use for all users, service providers, and employees are guaranteed by the app's strict avoidance of exploitative methods.

7. Legal Compliance

To ensure legal and moral operations, the project closely complies with all applicable national and international legislation. The platform is guaranteed to comply with the legal framework for online services if it complies with Turkey's e-commerce legislation. To make sure users are completely aware of their rights and obligations when using the app, a clear and easy-to-read terms and conditions paper is also supplied.

8. Ethical Development Practices

The group uses open-source tools responsibly, following license terms to the letter and giving back to the community whenever they can. In addition, all team members adhere to a code of conduct that promotes ethical behavior, respect, and teamwork at every stage of development.

4.7. Planning for New Knowledge and Learning Strategies

To ensure the success and sustainability of Yolla, it is critical to plan for continuous learning and the acquisition of new knowledge. In order to achieve this, our approach integrates various strategies and resources.

GitHub and Open-Source Contributions

- GitHub will be employed not only as a version control platform but also as a resource for exploring open-source projects.
- Existing repositories related to logistics systems, cargo tracking, and real-time image processing will be reviewed to identify reusable modules and innovative techniques. Customer service chatbot will be investigated in the same manner.

Online Courses and Tutorials

- Training materials from platforms such as Coursera, Udemy will be studied.
- Courses covering Python backend frameworks, React frontend development, and image processing tools like OpenCV and TensorFlow will be prioritized.
- Tutorials on DevOps, containerization, and cloud deployment will be used to enhance backend scalability and maintainability.

Literature Reviews

- Relevant research papers, technical articles, and case studies will be analyzed to understand state-of-the-art techniques in:
 - Object detection and size estimation for box measurement.
 - Logistics optimization algorithms for pricing and delivery time comparisons.
 - Retrieval Augmented Generation for customer support chatbot models.
 - Real-time tracking systems for efficient data integration.

Leveraging Large Language Models

- Tools such as ChatGPT, Gemini, PerplexityAI, Claude and other LLMs will be utilized to:
 - Generate sample code and resolve programming challenges.
 - Summarize and explain technical documentation.
 - Suggest potential optimizations for the backend and frontend systems.

Documentaries and Industry Insights

- Official documentaries of the languages/libraries will be consulted as needed to ensure accurate understanding and implementation.
- Case studies highlighting the challenges and solutions implemented by industry leaders will be reviewed.

Trial and Error Learning

- Prototypes and experiments will be developed to explore potential solutions, refine features, and identify the most effective approaches.
- Through iterative testing and feedback, issues will be identified and addressed, leading to improved system functionalities and better user experiences.

5. Glossary

6. References

[1] Git, "Git," Accessed: Dec. 15, 2024. [Online]. Available: <https://git-scm.com>

[2] GitHub, "GitHub: Where the world builds software," Accessed: Dec. 15, 2024. [Online]. Available: <https://github.com>

[3] Trello, "Trello: Manage your team's projects from anywhere," Accessed: Dec. 15, 2024. [Online]. Available: <https://trello.com>

[4] Google Docs, "Google Docs," Accessed: Dec. 15, 2024. [Online]. Available: <https://docs.google.com>

[5] Zoom, "Zoom Video Communications," Accessed: Dec. 15, 2024. [Online]. Available: <https://zoom.us>

[6] Google Meet, "Google Meet," Accessed: Dec. 15, 2024. [Online]. Available: <https://meet.google.com>

[7] WhatsApp, "WhatsApp," Accessed: Dec. 15, 2024. [Online]. Available: <https://www.whatsapp.com>

[8] Figma, "Figma," Accessed: Dec. 15, 2024. [Online]. Available: <https://www.figma.com>