



Uniwersytet  
Wrocławski

## Wroclaw University

Software Engineering

**LEI-TracK**

A system supporting the management of orders to a Delivery  
Company Store (Service Company)



Henrique Alvelos  
(344153)



João Caldas  
(344152)



Júlio Gonçalves  
(344152)

# Contents

<b>1 Application theme selection</b>	<b>3</b>
<b>2 Description of the fragment of reality that the application relates to</b>	<b>4</b>
2.1 Motivation and Goals . . . . .	4
2.2 Justification and Utility of the System . . . . .	4
<b>3 Description of the application functionality in the form of user stories</b>	<b>5</b>
3.1 Actors . . . . .	5
3.2 Worker User Stories . . . . .	5
3.3 Admin User Stories . . . . .	6
3.4 Domain Model . . . . .	6
<b>4 Identification of use cases</b>	<b>7</b>
4.1 Funcional Requirements . . . . .	7
4.2 No Funcional Requirements . . . . .	7
4.3 Use Case Diagram . . . . .	9
4.4 Some Use Case Scenarios . . . . .	9
4.4.1 Login . . . . .	9
4.4.2 Track order . . . . .	9
4.4.3 Change delivery status . . . . .	9
<b>5 Short description of the algorithms planned to be implemented</b>	<b>10</b>
5.1 Login . . . . .	10
5.2 Handle Order . . . . .	11
5.3 Deliver Order . . . . .	11
5.4 Cancel Order . . . . .	12
5.5 Notification . . . . .	12
<b>6 Development of a conceptual database model</b>	<b>13</b>
<b>7 Technical database design</b>	<b>15</b>
<b>8 Database implementation and integration of CRUD operations</b>	<b>16</b>
<b>9 Concept development and frontend design, which is what the end user, including what admin sees</b>	<b>17</b>
9.1 Class Diagram . . . . .	17
9.2 Components Diagram . . . . .	18
9.3 Package Diagram . . . . .	18
9.4 Mockups . . . . .	19
9.4.1 Login . . . . .	19
9.4.2 Register . . . . .	20
9.4.3 Admin when sees Worker Stats . . . . .	20
9.4.4 Worker when registers order . . . . .	21
9.4.5 Client when sees its Notifications . . . . .	21

<b>10 Implementation of selected, non-trivial frontend fragments</b>	<b>22</b>
10.0.1 Login . . . . .	22
10.0.2 Register . . . . .	23
10.0.3 Admin when sees Worker Stats . . . . .	23
10.0.4 Worker when registers order . . . . .	23
10.0.5 Client when sees its Notifications . . . . .	24
<b>11 Conclusion</b>	<b>25</b>

# Chapter 1

## Application theme selection

This work relates to the study and development of a system that will support management of orders inside the service industry. Following the guidelines provided, this development will follow a Waterfall approach with the following stages in scope:

- Study of Domain
- Requirements Analysis
- System Design
- System Implementation

**Note:** In a normal development project, more stages would be required.  
A Gant Diagram can represent the various development phases of this project:

	2023			2024	
	Oct	Nov	Dec	Jan	Feb
Study of Domain					
Requirements and Use Case Analysis					
System Design					
Back-End Development					
Front-End Development					

Figure 1.1: Gant Diagram

# **Chapter 2**

## **Description of the fragment of reality that the application relates to**

The service industry is a vital component of our economy, and effective order management is essential for its success. As service companies continue to grow, there is a rising need for adaptable solutions that prioritize the customer.

### **2.1 Motivation and Goals**

In any economic sector, there's always room for improvement. We aim to create a project that simplifies and improves order management for service companies, especially in the field of delivery. With this app, our goal is to provide them with the tools they need to streamline operations and enhance customer satisfaction.

### **2.2 Justification and Utility of the System**

Efficient order management is essential for service companies. This application is designed to help service companies save time and resources by simplifying order creation, assignment, tracking, and communication with customers. It aims to make the entire order management process more seamless and customer-centric, ultimately improving the overall service experience.

# **Chapter 3**

## **Description of the application functionality in the form of user stories**

### **3.1 Actors**

First of all, we want to define who will be the users of our system. Different types of users will take access to different domains of the application and as a result of this to different functionalities.

We decide to divide the users of our system into three types of users:

- Customer User
- Worker User
- Admin User

#### Customer User Stories

- As a Customer user, I want to be able to track my order delivery progress.
- As a customer user, I want to be notified when my order delivery updates the progress.
- As a Customer user, I want to see the expected delivery date.
- As a Customer user, I want to receive an order confirmation and details.
- As a Customer user, I want to view and access my order history and previous purchases easily.
- As a Customer user, I want to be able some of my profile settings

### **3.2 Worker User Stories**

- As a Worker User, I want to be able to add new deliveries to the system.
- As a Worker User, I want to be able to change the delivery status
- As a Worker User, I want access to a knowledge base or resources to help me handle various delivery scenarios efficiently.
- As a Worker user, I want to view and access my order history and previous purchases easily.
- As a Worker user, I want to be able some of my profile settings.

### 3.3 Admin User Stories

- As an Admin User, I want to manage and maintain the worker user accounts, including creating worker profiles.
- As an Admin User, I want to be able to see some delivery statistics.
- As an Admin user, I want to view and access my order history and previous purchases easily.
- As an Admin user, I want to be able some of my profile settings.

### 3.4 Domain Model

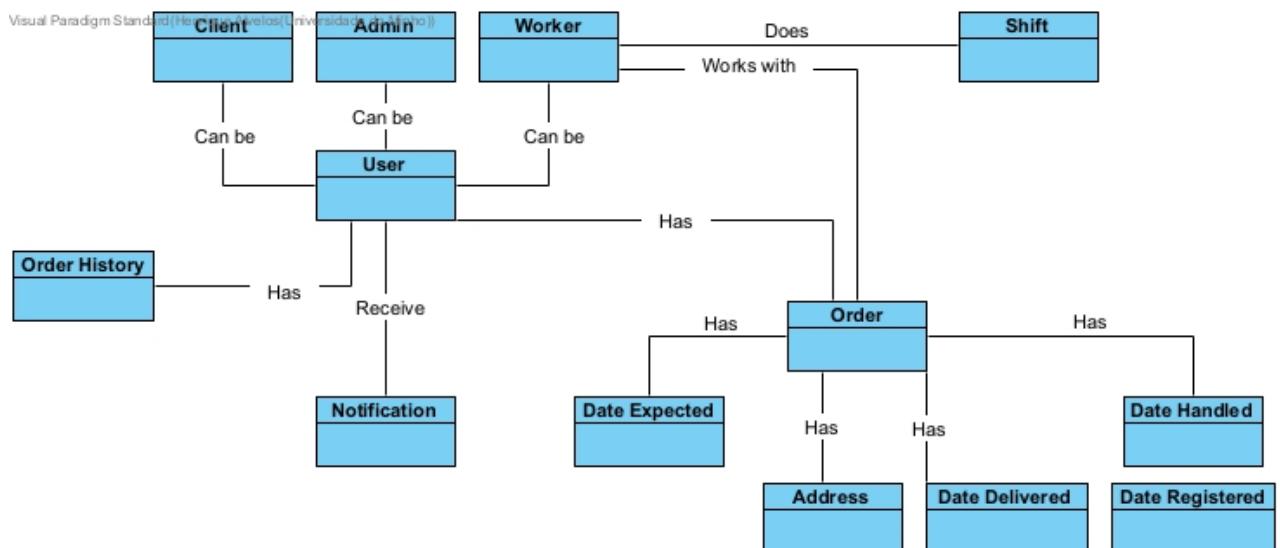


Figure 3.1: Domain Model

# Chapter 4

## Identification of use cases

### 4.1 Funcional Requirements

#### 1. User Registration and Login:

- a) The system shall provide a user registration module where users can create accounts by providing necessary information such as username, email, password, address and phone number.
- b) The system shall include a secure login module allowing registered users to authenticate themselves using their credentials.

#### 2. Order History:

- a) The system shall maintain a centralized database to store order information, including order date, items purchased, and delivery status.
- b) Upon successful login, the system shall retrieve and display the user's order history with detailed information.
- c) The order history shall be presented in a clear and organized format for easy user comprehension.

#### 3. Order Shipment and Cancellation:

- a) The system shall include a module for users to view their active orders.
- b) Workers shall be allowed to initiate order delivery or cancellations.
- c) The system shall update the order status.

### 4.2 No Funcional Requirements

- The technicians/reviewers are previously registered in the database.
- The application must run smoothly and without bugs, always ensuring the security of both the user and the system.
- Application response times should never exceed 15 seconds.
- The application must present an elegant and simple interface, to cover a wide age range.
- The application must be decentralized, always ensuring that the system is online and operational 24 hours a day, 7 days a week.
- The application must be of a level that allows it to run on the greatest number of devices, with hardware limitations not being an inconvenience to its use.

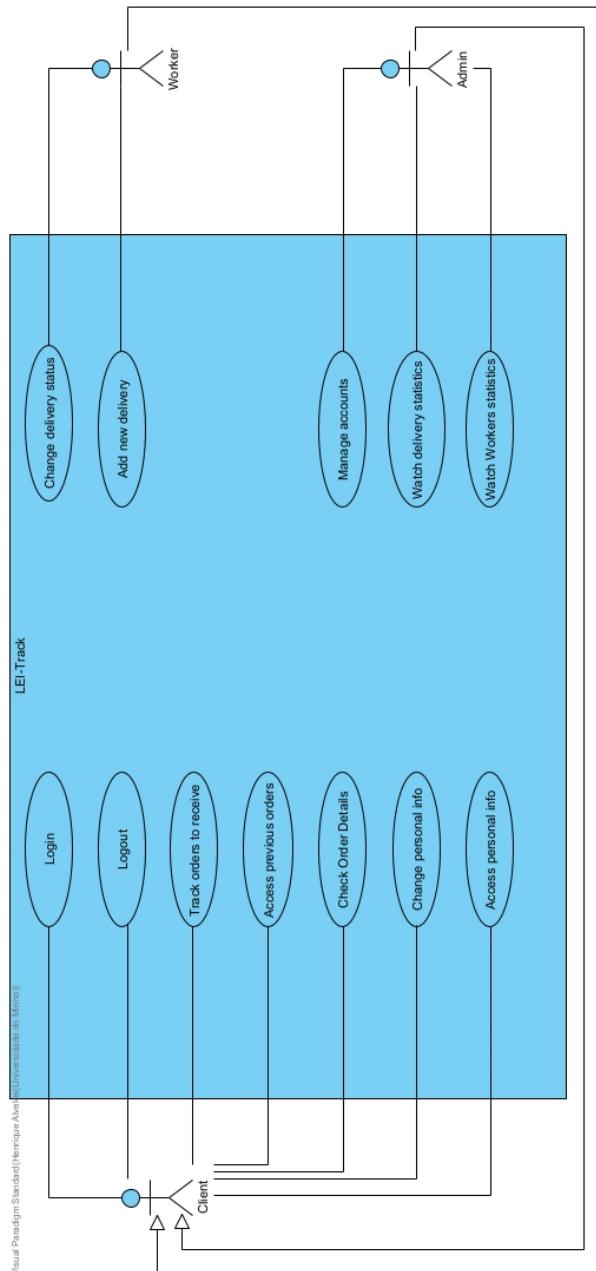


Figure 4.1: Use Case Diagram

### 4.3 Use Case Diagram

### 4.4 Some Use Case Scenarios

#### 4.4.1 Login

Login		
Description	The user wants to login	
Pre-Condition	The user is not logged	
Post-Condition	The user is logged	
Normal Flow	1.	The user tries to log in to the app.
	2.	The user inserts his email and password.
	3.	«include» Autenticate user
	4.	The user is logged in.
Exception Flow	4.1	The user doesn't exist in the database.

#### 4.4.2 Track order

Track Order		
Description	The user wants to see the Order Status	
Pre-Condition	The user is logged in and has an order assigned to his account	
Post-Condition	The user see the order informations	
Normal Flow	1.	The user selects the order button.
	2.	All order information is displayed.

#### 4.4.3 Change delivery status

Change Delivery Status		
Description	The worker updates the delivery status	
Pre-Condition	The user is logged	
Post-Condition	The order status have been updated successfully	
Normal Flow	1.	The worker selects an order to deliver or to handle
	2.	The worker clicks a new status to update.
	3.	The order status has been updated.
	4.	The user is notified.

# Chapter 5

## Short description of the algorithms planned to be implemented

### 5.1 Login

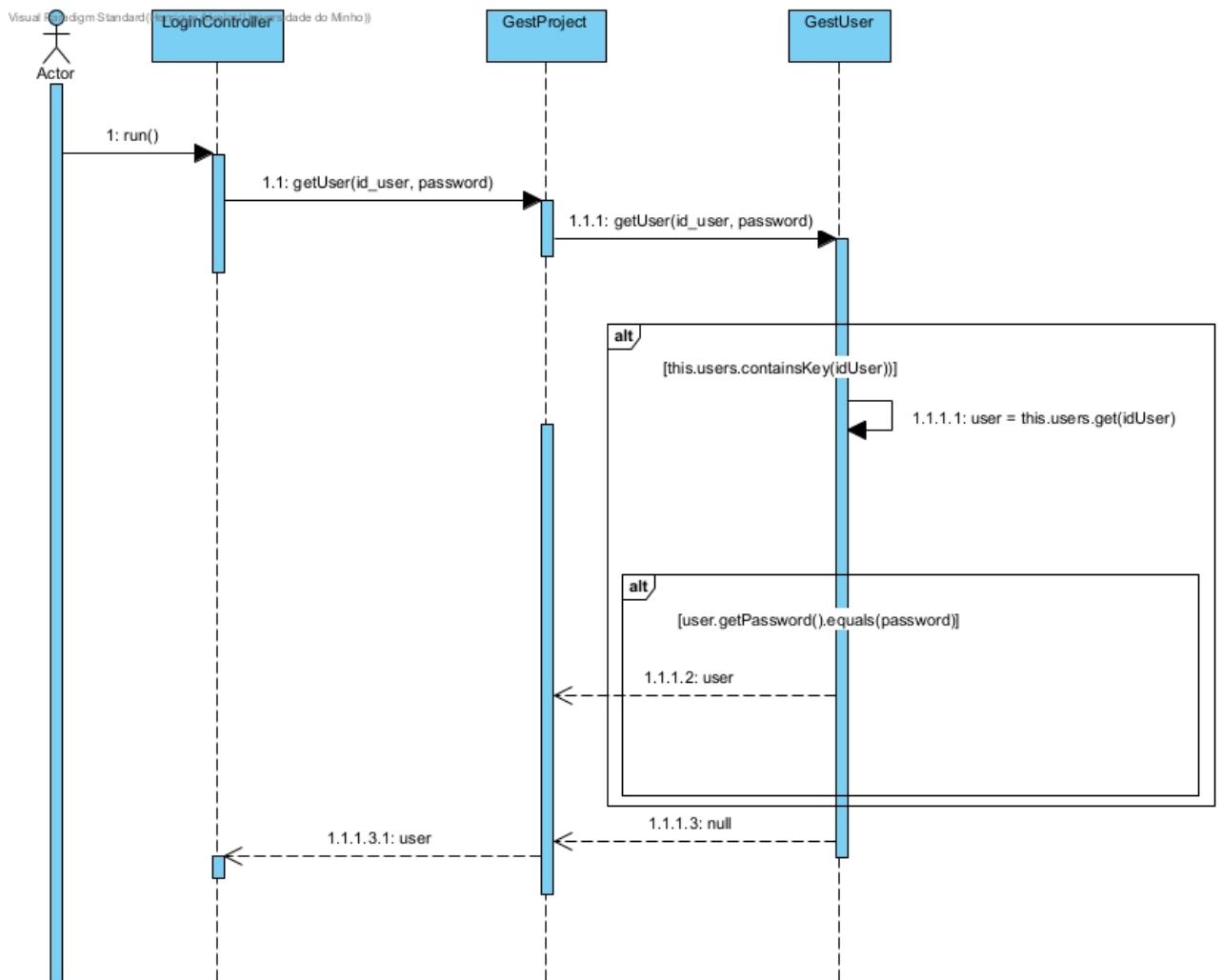


Figure 5.1: Login Sequence Diagram

## 5.2 Handle Order

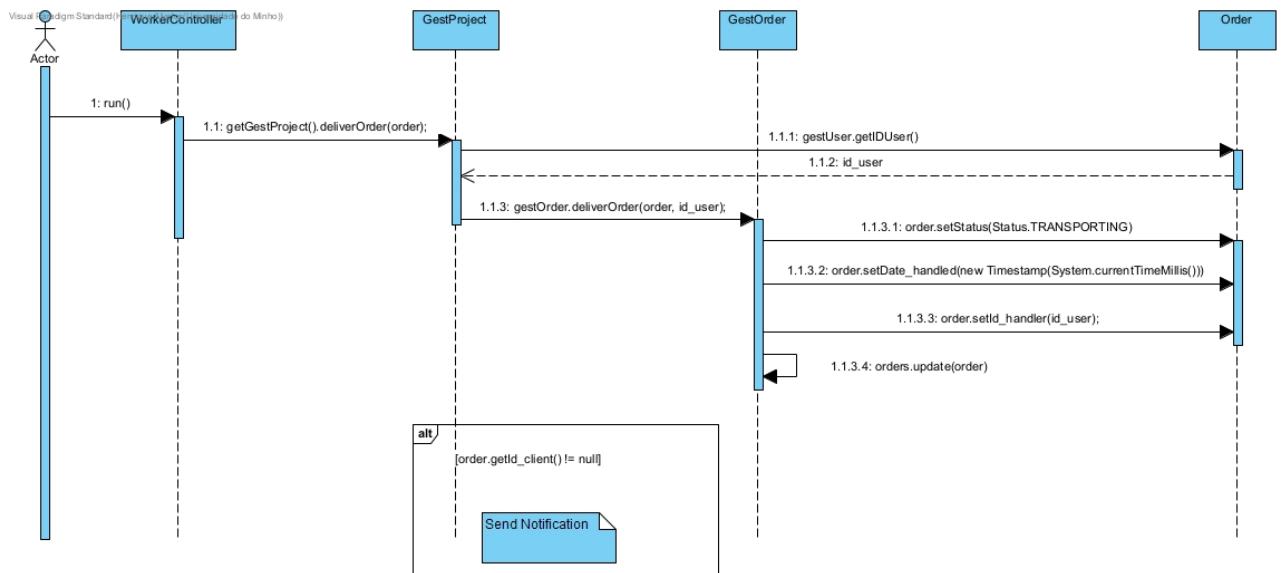


Figure 5.2: Handle Order Sequence Diagram

## 5.3 Deliver Order

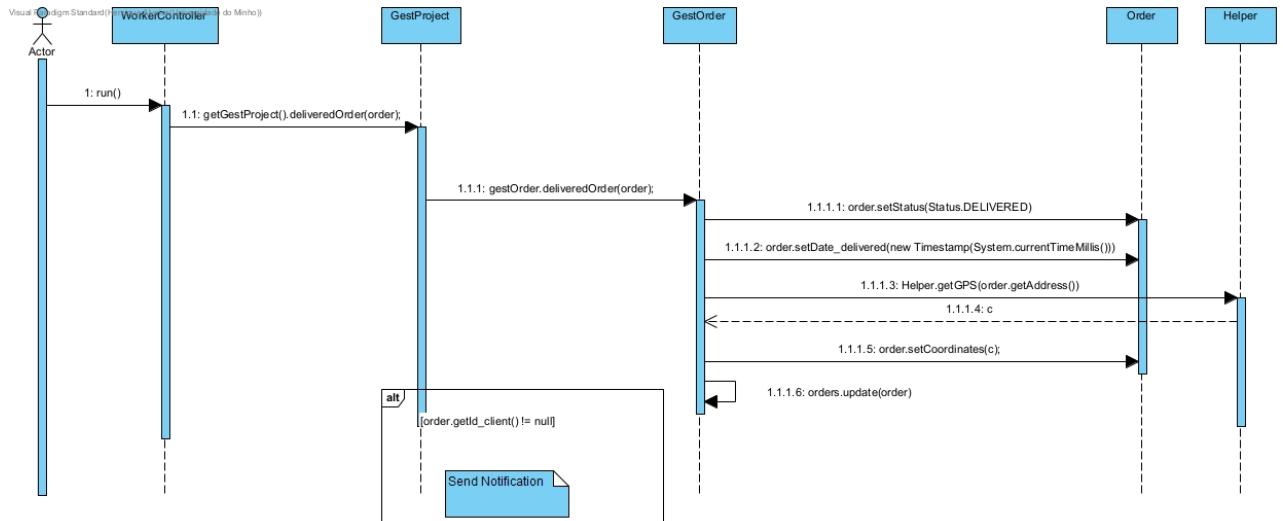


Figure 5.3: Deliver Order Sequence Diagram

## 5.4 Cancel Order

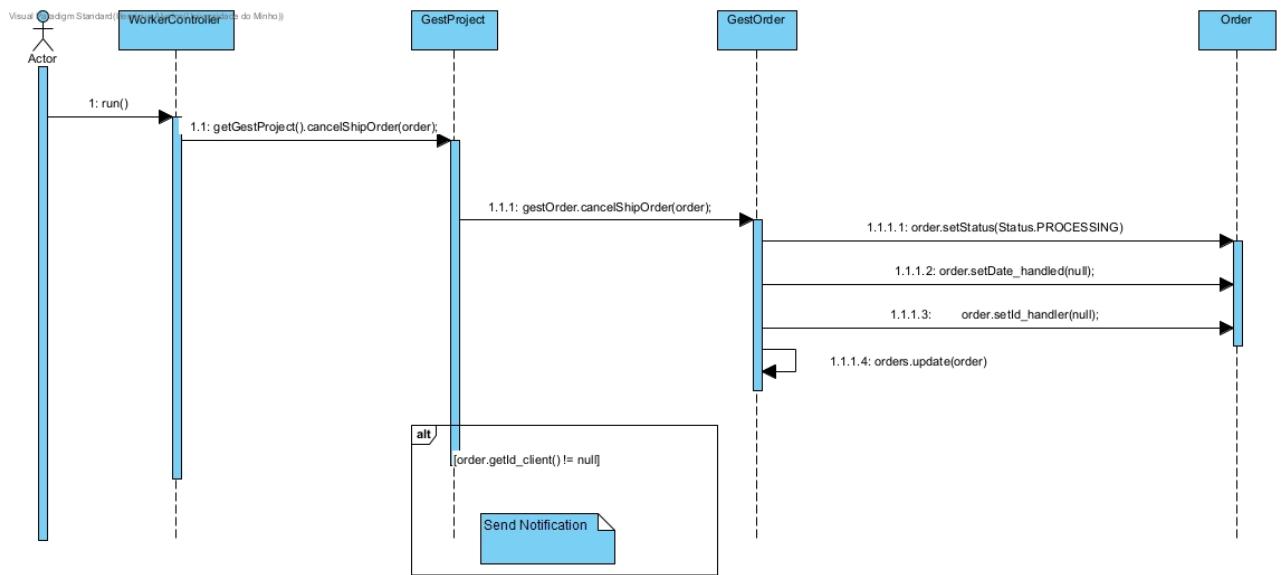


Figure 5.4: Cancel Order Sequence Diagram

## 5.5 Notification

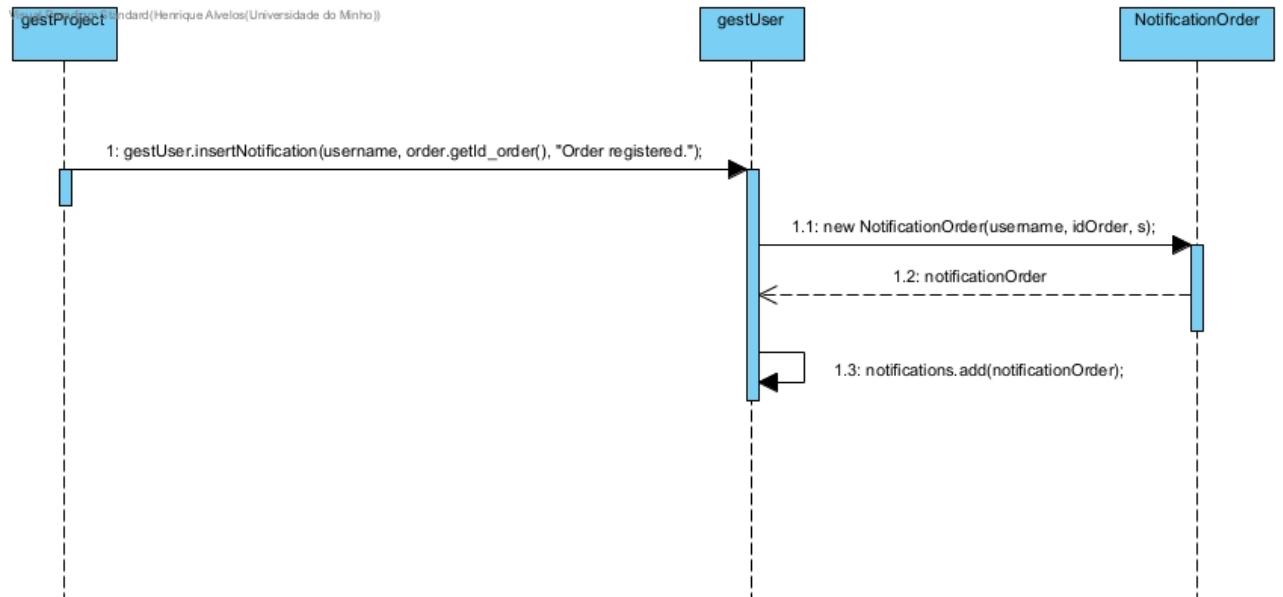


Figure 5.5: Notification Sequence Diagram

## **Chapter 6**

# **Development of a conceptual database model**

This visual representation encapsulates the essence of our database's logical model, showcasing the interconnected entities, relationships, and attributes that form the backbone of our data structure. A glance at this image provides a concise insight into the organized architecture that underpins our information management system.

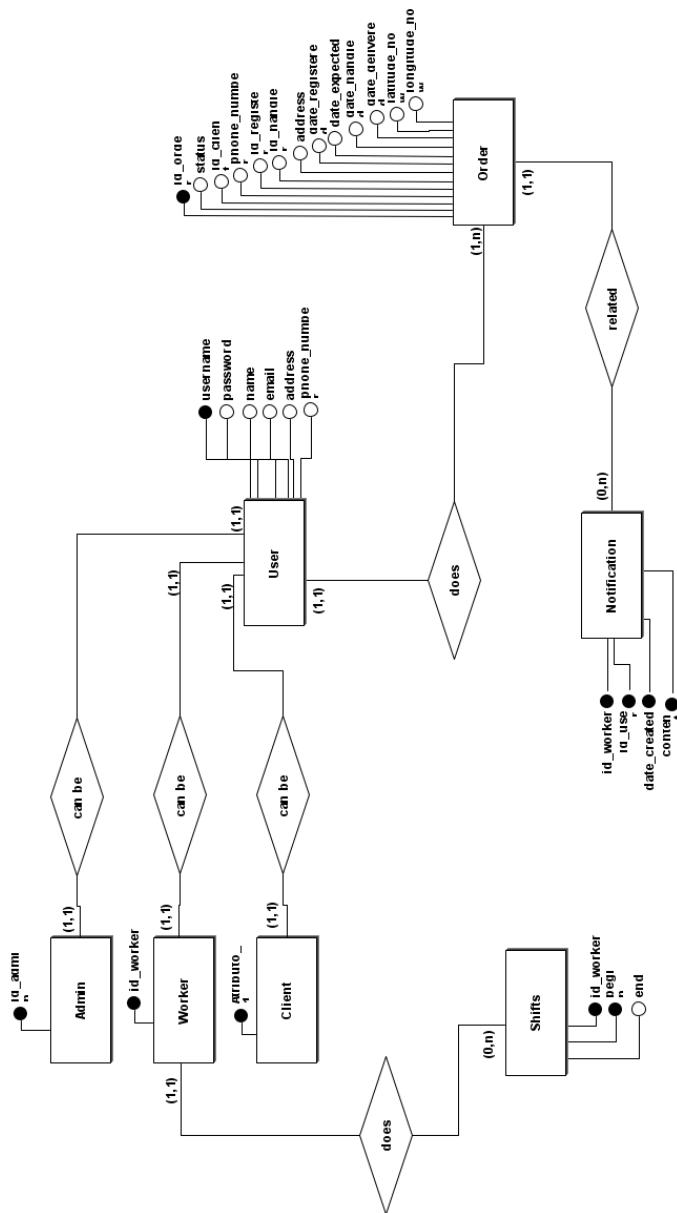


Figure 6.1: Conceptual Model

# Chapter 7

## Technical database design

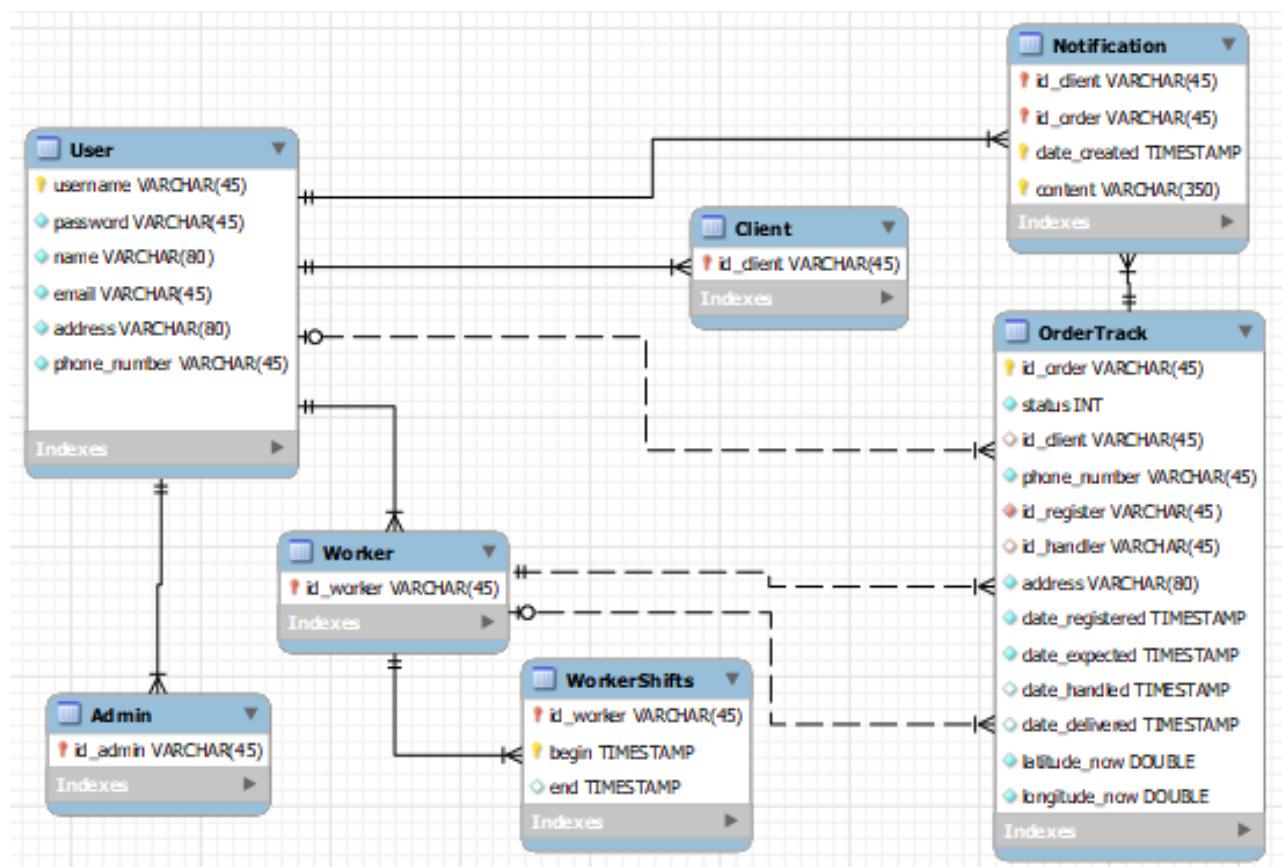


Figure 7.1: Logical Model

## **Chapter 8**

# **Database implementation and integration of CRUD operations**

To connect the database with the program, we created some classes that change data on the database directly. Those are in the DAO package. This package is only accessed by the Model package, which has all the Entities.

# Chapter 9

## Concept development and frontend design, which is what the end user, including what admin sees

### 9.1 Class Diagram

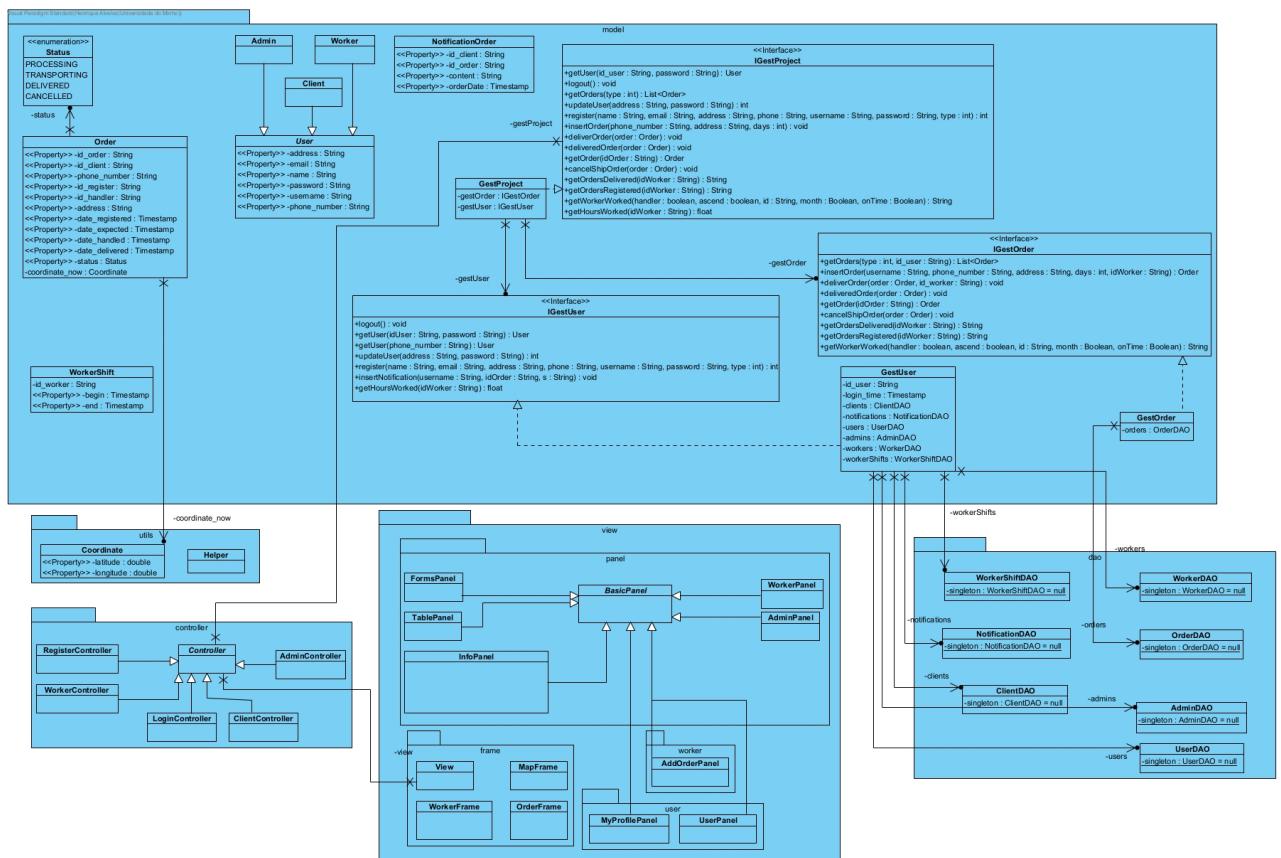


Figure 9.1: Class Diagram

## 9.2 Components Diagram

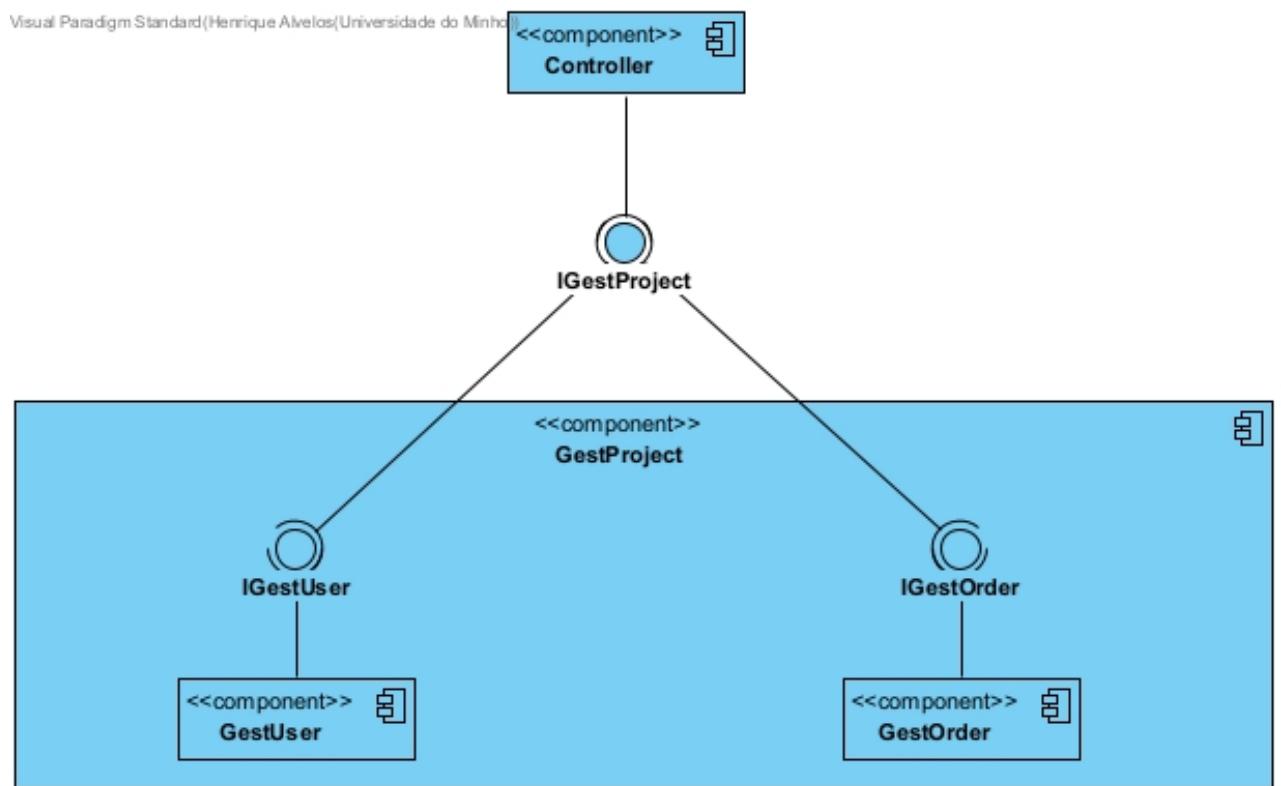


Figure 9.2: Components Diagram

## 9.3 Package Diagram

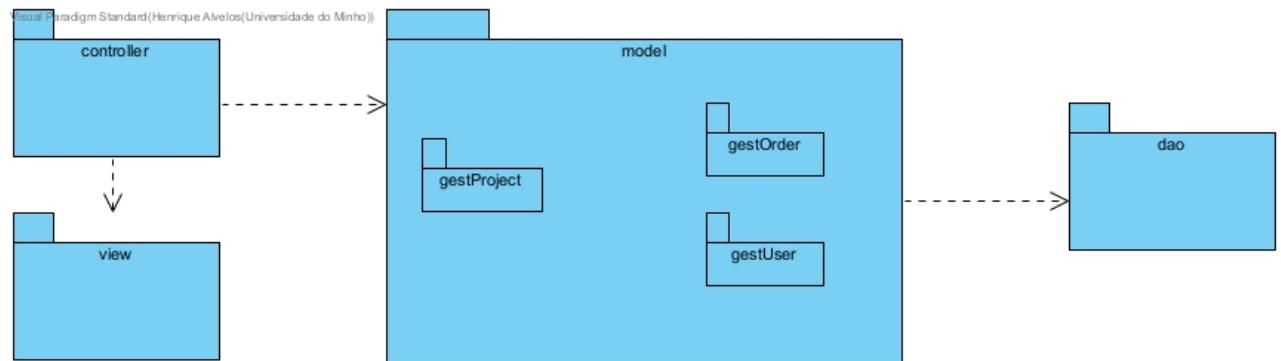


Figure 9.3: Package Diagram

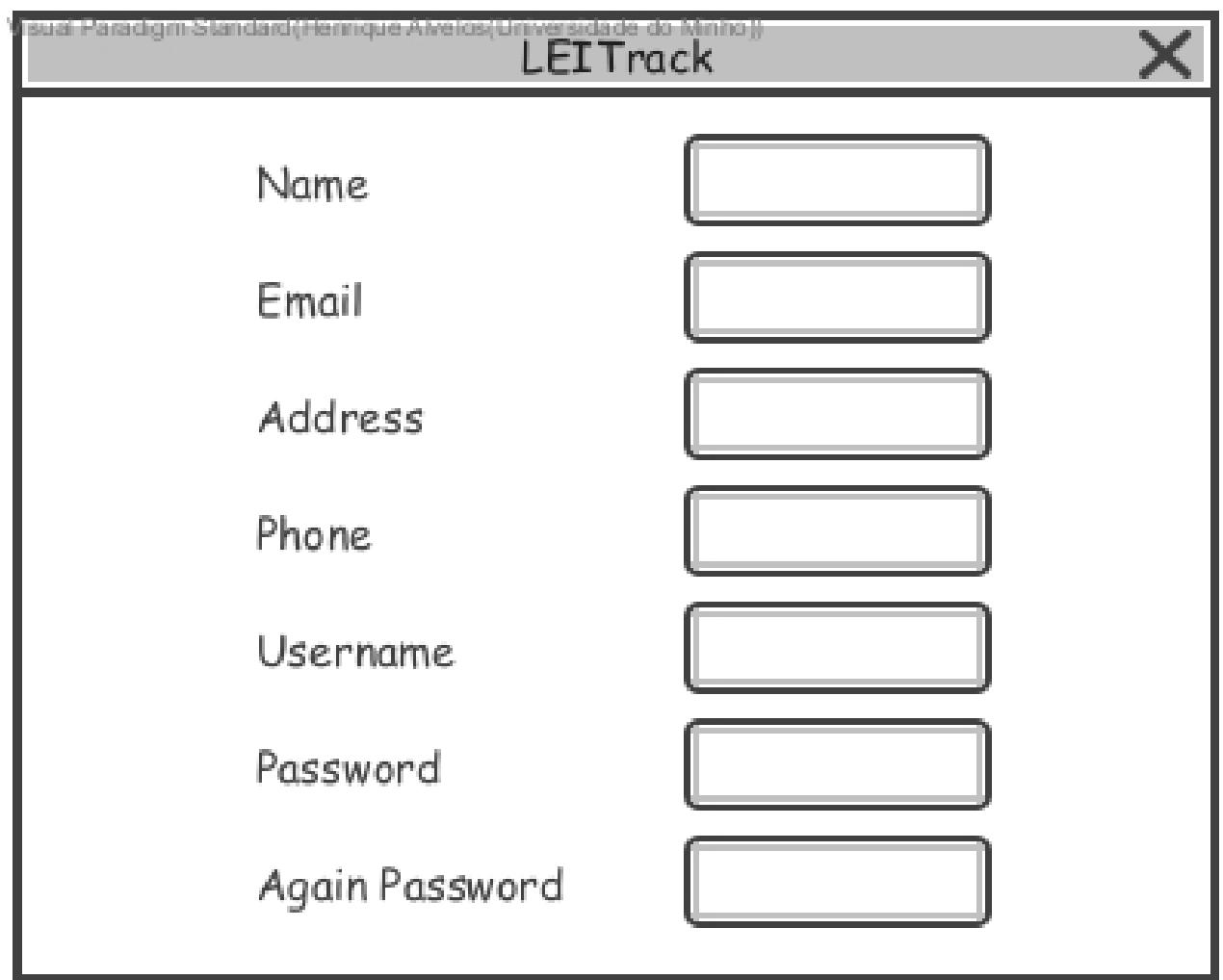
## 9.4 Mockups

### 9.4.1 Login



Figure 9.4: Login Mockup

#### 9.4.2 Register

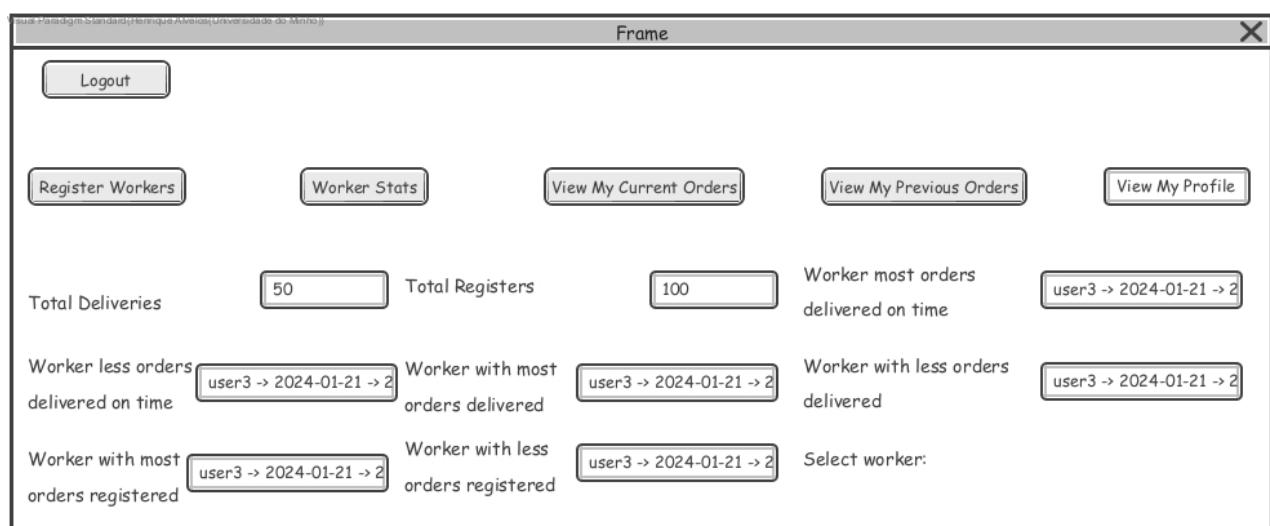


The mockup shows a registration form titled "LEITrack". It contains fields for Name, Email, Address, Phone, Username, Password, and Again Password, each with a corresponding input box.

Name	<input type="text"/>
Email	<input type="text"/>
Address	<input type="text"/>
Phone	<input type="text"/>
Username	<input type="text"/>
Password	<input type="text"/>
Again Password	<input type="text"/>

Figure 9.5: Register Mockup

#### 9.4.3 Admin when sees Worker Stats



The mockup shows an admin interface for worker statistics. It includes a "Logout" button and links for "Register Workers", "Worker Stats", "View My Current Orders", "View My Previous Orders", and "View My Profile". Below these are various statistics and their corresponding values and details.

Total Deliveries	50	Total Registers	100	Worker most orders delivered on time	user3 → 2024-01-21 → 2
Worker less orders delivered on time	user3 → 2024-01-21 → 2	Worker with most orders delivered	user3 → 2024-01-21 → 2	Worker with less orders delivered	user3 → 2024-01-21 → 2
Worker with most orders registered	user3 → 2024-01-21 → 2	Worker with less orders registered	user3 → 2024-01-21 → 2	Select worker:	

Figure 9.6: Worker Stats Mockup

#### 9.4.4 Worker when registers order

The mockup shows a window titled "Frame". At the top left is a "Logout" button. Below it is a horizontal row of buttons: "Register Order", "Start Delivery", "Mark Delivered", "View Orders Delivered", "View my Current Orders", "View My Previous Orders", and "View My Profile". The main area contains input fields for "Phone Number", "Address", and "Days to Deliver", each with an associated text input box. At the bottom are two buttons: "Clear" and "Insert".

Figure 9.7: Register Order Mockup

#### 9.4.5 Client when sees its Notifications

The mockup shows a window titled "Frame". At the top left is a "Logout" button. Below it is a horizontal row of buttons: "Orders To Receive", "Previous Orders", "My Profile", and "Notifications". The main area features a table with three columns: "Date", "OrderID", and "Content". The "Date" column is currently empty.

Figure 9.8: Client Notifications Mockup

# Chapter 10

## Implementation of selected, non-trivial frontend fragments

### 10.0.1 Login

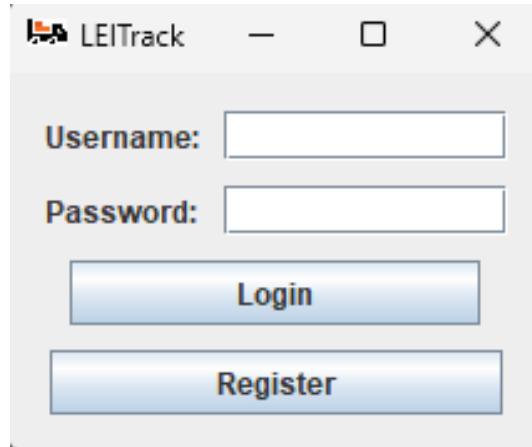


Figure 10.1: Login View

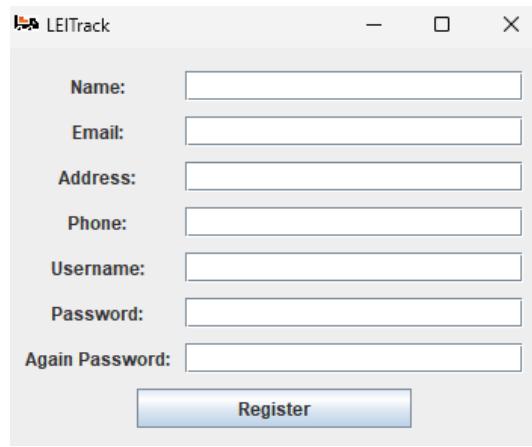


Figure 10.2: Register Client View

### 10.0.2 Register

### 10.0.3 Admin when sees Worker Stats

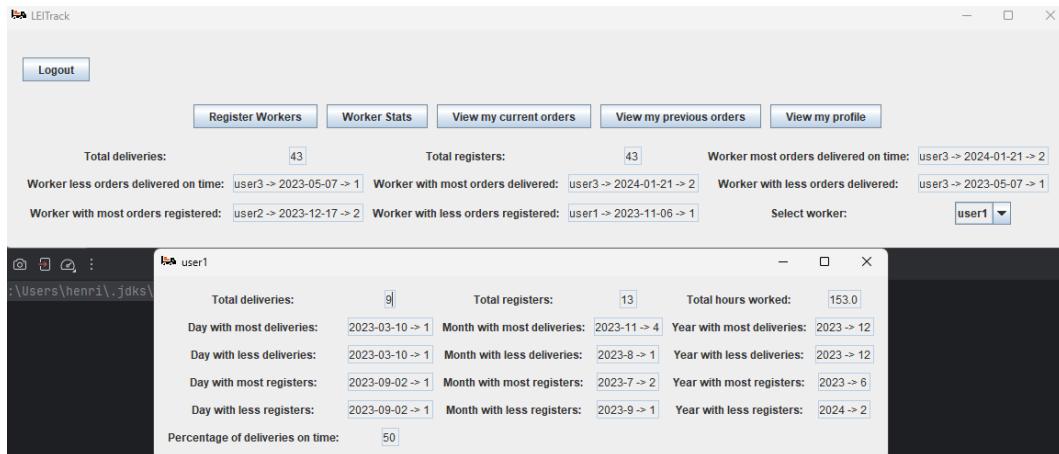


Figure 10.3: Admin View of Worker Stats

### 10.0.4 Worker when registers order

Phone Number:   
 Address:   
 Days to deliver:

Figure 10.4: Register Order View

### 10.0.5 Client when sees its Notifications

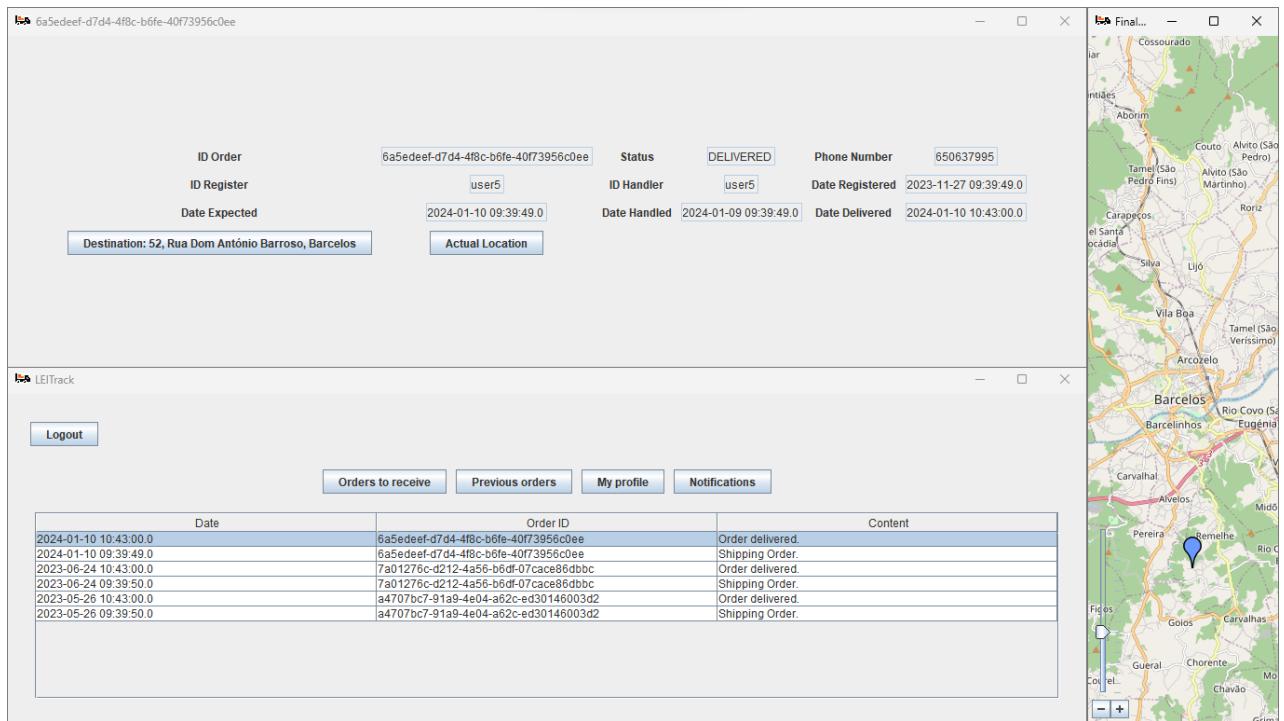


Figure 10.5: Client Notification View with Order Detail and Current Location

# **Chapter 11**

## **Conclusion**

The Order Delivering software engineering project has successfully enhanced order delivery processes through features like tracking. The team demonstrated effective collaboration and adaptability, overcoming integration challenges and evolving user requirements.