**CEF440: INTERNET PROGRAMMING (J2EE) AND MOBILE PROGRAMMING**

**Software design of a Passenger positioning system (Municipal commuting App)**

Presented by:

| Name | Matricule |
| --- | --- |
| Acha Rha’ah Achubang | FE20A001 |
| Amehmbo Ngewung Sonia | FE20A008 |
| Atem Randy Asong | FE20A014 |
| Tambe Salome Ntoh | FE20A109 |
| Tiokeng Samuel Edward | FE20A110 |

**Table Of Content**

[**Introduction 2**](#_xqxkpu6kgpiy)

[Purpose 2](#_nhj1z3qlnbrs)

[Overview 2](#_gtx8xp4qu3qg)

[Scope 2](#_o5si3uftbvce)

[**Functional Design 3**](#_krbg84ukjd2n)

[Use case Diagram 3](#_tm3a3niuarpv)

[Descriptions of Use cases 3](#_iki9ypy1gfx5)

[Class Diagram 10](#_dg3jstw2lfod)

[Sequence Diagram 10](#_9obpv67xez89)

[Activity Diagram 10](#_b6w95ob7qp9z)

[**Conclusion 13**](#_ia4ucvftly1v)

# **Introduction**

## **Purpose**

This software design document focuses on the development of a mobile application for connecting passengers and drivers. The objective is to create an efficient and secure platform that optimizes fuel consumption for drivers and reduces waiting times for passengers. To achieve this goal, the system will utilize Unified Modeling Language (UML) diagrams to provide a comprehensive view of the system components, their relationships, and behaviors. The document will provide an overview of the system's architecture, requirements, and features, along with detailed UML diagrams for each component. Additionally, the document will outline the testing and deployment procedures to ensure a smooth and successful launch of the application. Overall, this document aims to provide a clear and detailed roadmap for the development of a robust and user-friendly mobile application for connecting passengers and drivers.

## **Overview**

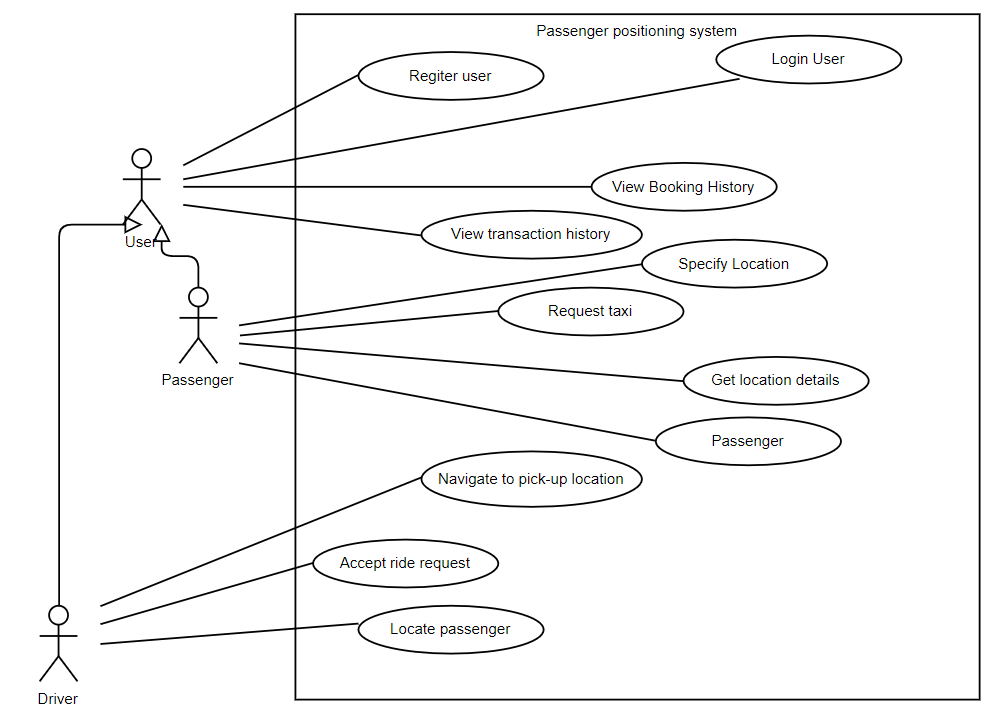
The mobile application that is being developed is a platform for connecting passengers and drivers. It is designed to provide a convenient and efficient way for passengers to hail rides and for drivers to pick up customers. The application aims to optimize fuel consumption for drivers and reduce waiting times for passengers. Users of the application will be able to specify their location, destination, and preferred ride type, and the system will match them with drivers who are available and nearby. The application will provide real-time tracking of the ride to ensure safety and security for both passengers and drivers. Additionally, the application will allow passengers to rate and provide feedback on their driver, and drivers to rate and provide feedback on their passengers. Overall, this is an innovative and user-friendly application that aims to revolutionize the ride-hailing industry.

## **Scope**

The ride-hailing mobile application is designed to provide a platform for passengers to book their rides in a hassle-free manner. The application is developed to work seamlessly on Android and iOS devices, providing easy accessibility to users. The primary functions of the application include matching passengers with nearby drivers, flexible payment methods, real-time ride tracking, and a rating and feedback system to ensure high-quality and accountable services.

# **Functional Design**

## **Use case Diagram**



## 

## **Descriptions of Use cases**

**Register User**

| **UseCase ID** | **01** |
| --- | --- |
| UseCase Name | Register User |
| Actors | User |
| Description | The user enters their personal details into the app and creates an account. The system saves the user's details for future logins. |
| Precondition | None |
| Postcondition | User's account is created |
| Normal Flow | 1. User opens the app  2. User clicks on the "Register" button  3. User enters their personal details (name, email, phone number, password)  4. User selects the account type(Passenger, Driver)  5. User clicks on the "Register" button  6. System saves the user's details  7. System displays a message indicating successful registration |
| Alternative Flows | 3a. User enters an email that is already registered  6a. System displays an error message indicating that the email is already registered |

**Login**

| **UseCase ID** | **02** |
| --- | --- |
| UseCase Name | Login User |
| Actors | User |
| Description | The user enters their registered email and password to log in to the app. If the entered details are correct, the system grants access to the app's features. |
| Precondition | User has an account registered |
| Postcondition | User is logged in and can access app features |
| Normal Flow | 1. User opens the app  2. User enters their registered phone mumber and password  3. User clicks on the "LogIn" button  4. System verifies the user's credentials  5. System grants access to the app's features |
| Alternative Flows | 2a. User enters an invalid email or password  5a. System displays an error message indicating invalid email or password |

**Specify Location:**

| **UseCase ID** | **03** |
| --- | --- |
| UseCase Name | Specify Location |
| Actors | Passenger |
| Description | The passenger specifies their location and publishes to the drivers |
| Precondition | Passenger is logged in |
| Postcondition | Passenger location is available to drivers |
| Normal Flow | 1. Passenger opens the app  2. Passenger clicks on the "Specify Location" button  3. Passenger enters their address or selects their location on the map  4. Passenger clicks on the "Save" button  5. System saves the location  6. System displays a message indicating successful location save |
| Alternative Flows | 2a. User enters an invalid email or password    5a. System displays an error message indicating invalid email or password |

**Request Taxi:**

| **UseCase ID** | **04** |
| --- | --- |
| UseCase Name | Request Taxi |
| Actors | Passenger |
| Description | The passenger requests a taxi by specifying their destination, desired pick-up time, and the number of passengers. The system notifies nearby drivers of the request. |
| Precondition | Passenger has specified their location |
| Postcondition | Request is sent to nearby drivers |
| Normal Flow | 1. Passenger opens the app  2. Passenger clicks on the "Request Taxi" button  3. Passenger specifies their destination, desired pick-up time, and the number of passengers  4. Passenger clicks on the "Request" button  5. Passenger makes payment for the request  6. System notifies nearby drivers of the request  7. System displays a message indicating successful request |
| Alternative Flows | 6a. Payment failed and the system prompts users to try again. |

**Locate Passenger(Check Passenger Density):**

| **UseCase ID** | **05** |
| --- | --- |
| UseCase Name | Locate Passenger |
| Actors | Driver |
| Description | The driver uses the app to locate the passenger who has requested a taxi, based on their specified location or check passenger population density. |
| Precondition | Driver is logged in |
| Postcondition | Driver knows the passenger's location |
| Normal Flow | 1. Driver opens the app  2. Driver goes to the location  3. System displays a map with the passenger's specified location |
| Alternative Flows | None |

**Accept Ride Request:**

| **UseCase ID** | **06** |
| --- | --- |
| UseCase Name | Accept Ride Request |
| Actors | Driver |
| Description | The driver uses the app to locate the passenger who has requested a taxi, based on their specified location or check passenger population density. |
| Precondition | Driver is logged in and a ride request is sent by a passenger |
| Postcondition | Driver is assigned to provide taxi services |
| Normal Flow | 1. Driver receives a notification of a request from a passenger  2. Driver opens the app  3. Driver clicks on the "Accept Request" button  4. System assigns the driver to the passenger's request  5. System displays the passenger's details to the driver  6. Driver confirms the acceptance of the request  7. System sends a notification to the passenger that a driver has accepted the request |
| Alternative Flows | 3a. Driver ignores |

**Navigate to Pick-Up Location:**

| **UseCase ID** | **07** |
| --- | --- |
| UseCase Name | Navigate to Pick-Up Location |
| Actors | Driver |
| Description | The driver uses the app to navigate to the pick-up location of the passenger who has requested a taxi. |
| Precondition | Driver has accepted a request |
| Postcondition | Driver reaches the pick-up location. |
| Normal Flow | 1. Driver opens the app  2. Driver goes to the "Navigate to Pick-Up Location" button  3. System displays a map with the pick-up location and directions to get there  4. Driver follows the directions to get to the pick-up location  5. System displays a message indicating the driver has reached the pick-up location |
| Alternative Flows | None |

**View Driver Location:**

| **UseCase ID** | **08** |
| --- | --- |
| UseCase Name | View Location of Driver |
| Actors | Passenger |
| Description | A passenger wants to view the current location of the driver who is assigned to their ride |
| Precondition | The passenger has requested a ride and a driver has been assigned |
| Postcondition | The passenger is able to see the current location of the driver |
| Normal Flow | 1. The passenger opens the app  2. The passenger selects the "View Current Location" option  3. The app displays the current location of the driver on a map |
| Alternative Flows | 3a. If the driver's location cannot be determined, the app displays a message informing the passenger |

**Get Location Details:**

| **UseCase ID** | **09** |
| --- | --- |
| UseCase Name | Get Location Details |
| Actors | Passenger |
| Description | A passenger inputs details about their current location and their destination by either typing in their address or selecting their location on the map. |
| Precondition | The passenger has opened the app |
| Postcondition | The passenger location is available for the driver |
| Normal Flow | 1. The passenger opens the app  2. The passenger selects the input type (Current location, dropdown)  3. The passenger inputs their current location  4. The passenger inputs their destination. |
| Alternative Flows | None |

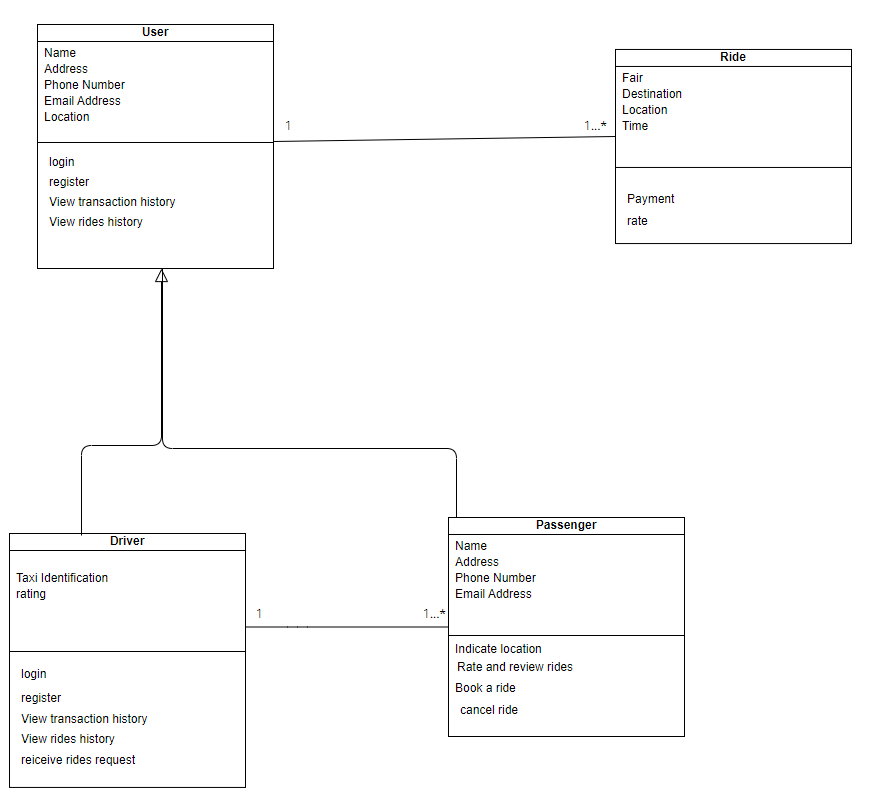
**Make Payment:**

| **UseCase ID** | **10** |
| --- | --- |
| UseCase Name | Make Payment |
| Actors | Passenger |
| Description | A passenger wants to pay for a ride request |
| Precondition | The passenger is making a ride request |
| Postcondition | The payment has been successfully processed |
| Normal Flow | 1. The passenger opens the app  2. The passenger fills in the ride request details  3. The app displays the fare amount and prompts the passenger to select a payment method  4. The passenger enters payment details  5. The app confirms successful payment and updates the transaction history |
| Alternative Flows | 3a. If the payment method is declined, the app displays a message informing the passenger |

**View Transaction History:**

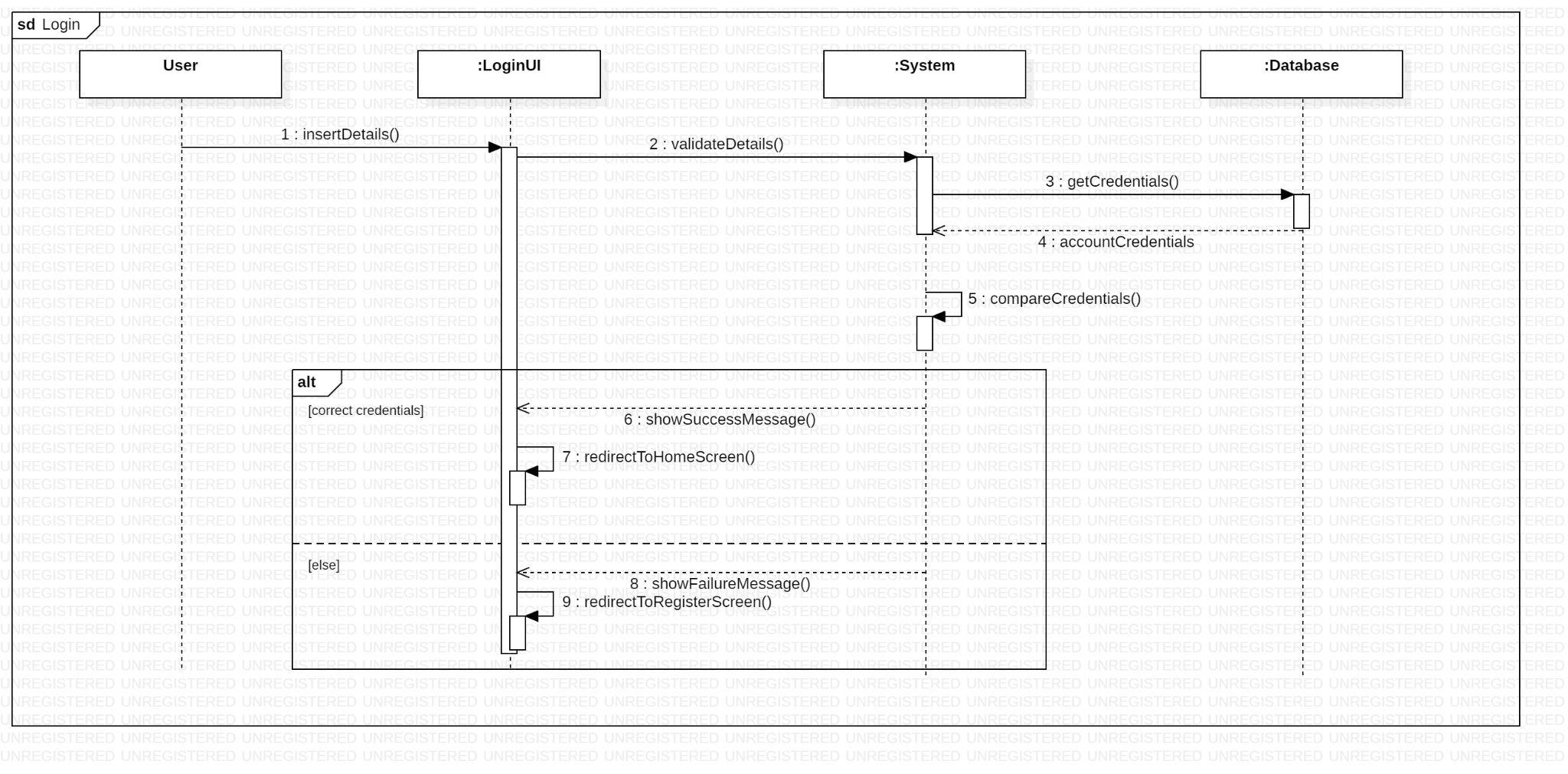
| **UseCase ID** | **11** |
| --- | --- |
| UseCase Name | View Transaction History |
| Actors | User(Passenger, Driver) |
| Description | A user wants to view their transaction history |
| Precondition | None |
| Postcondition | The user is able to view their transaction history |
| Normal Flow | 1. The user opens the app  2. The user selects the "View Transaction History" option  3. The app displays a list of all completed transactions, including the fare amount, date, and payment method |
| Alternative Flows | 3a. If the payment method is declined, the app displays a message informing the passenger |

## **Class Diagram**

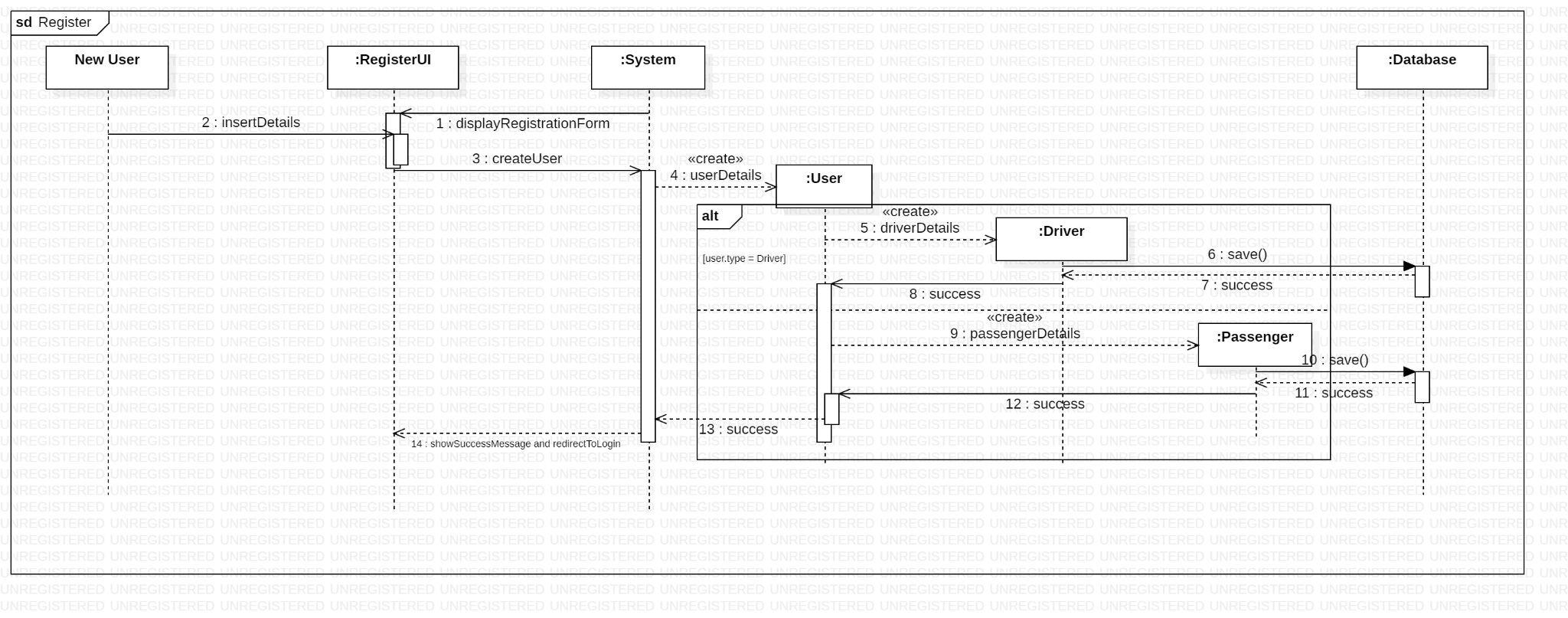


## **Sequence Diagram**

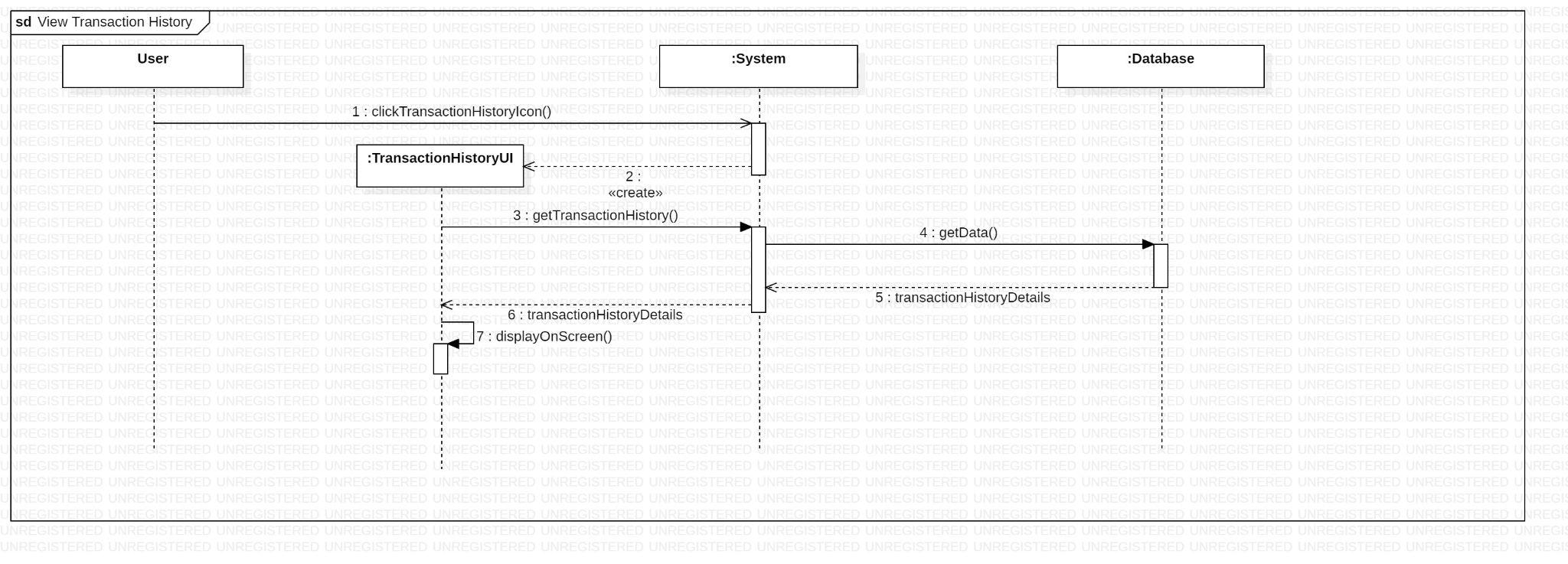
1. **Login**

****

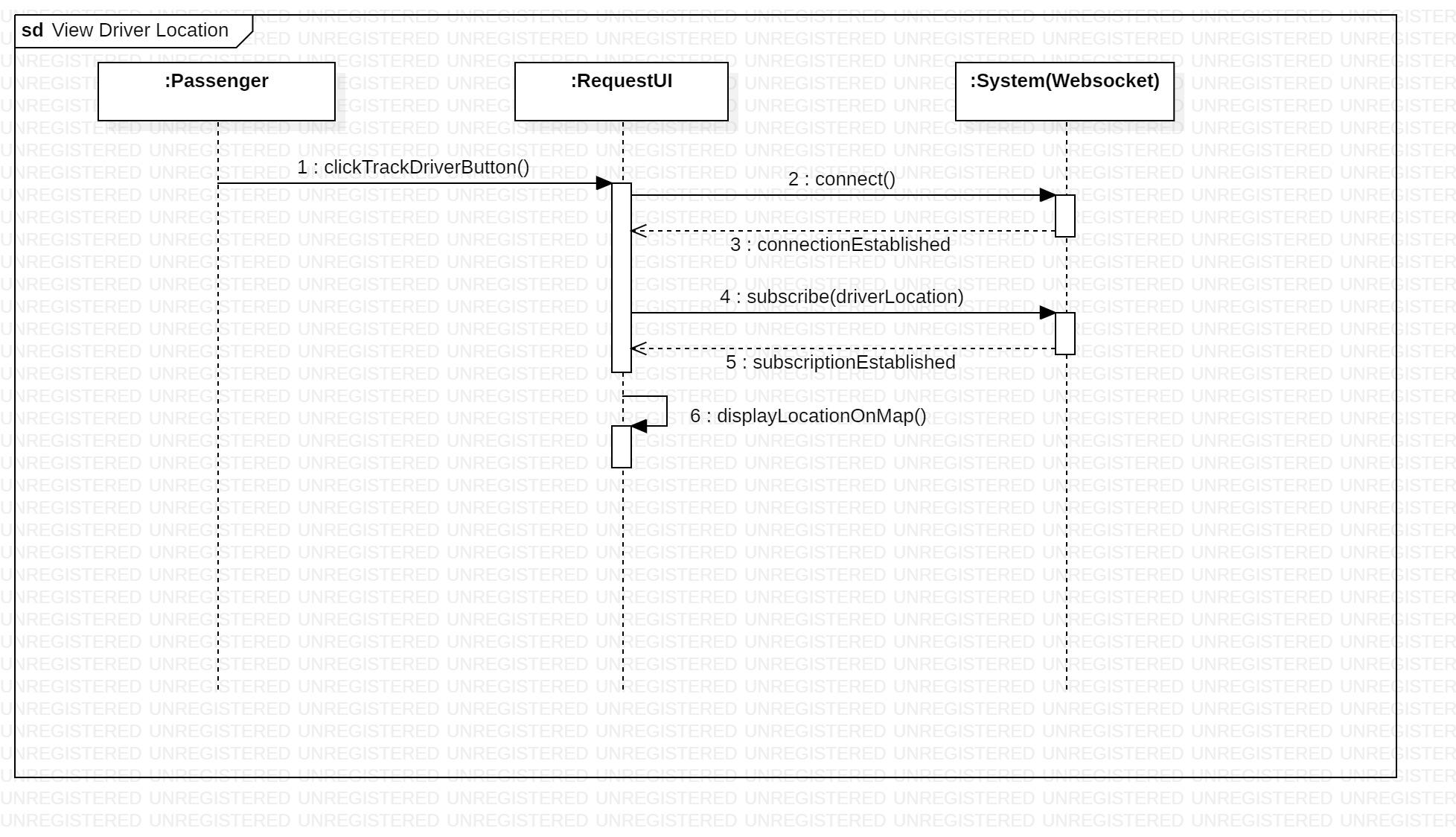
1. **Register**

****

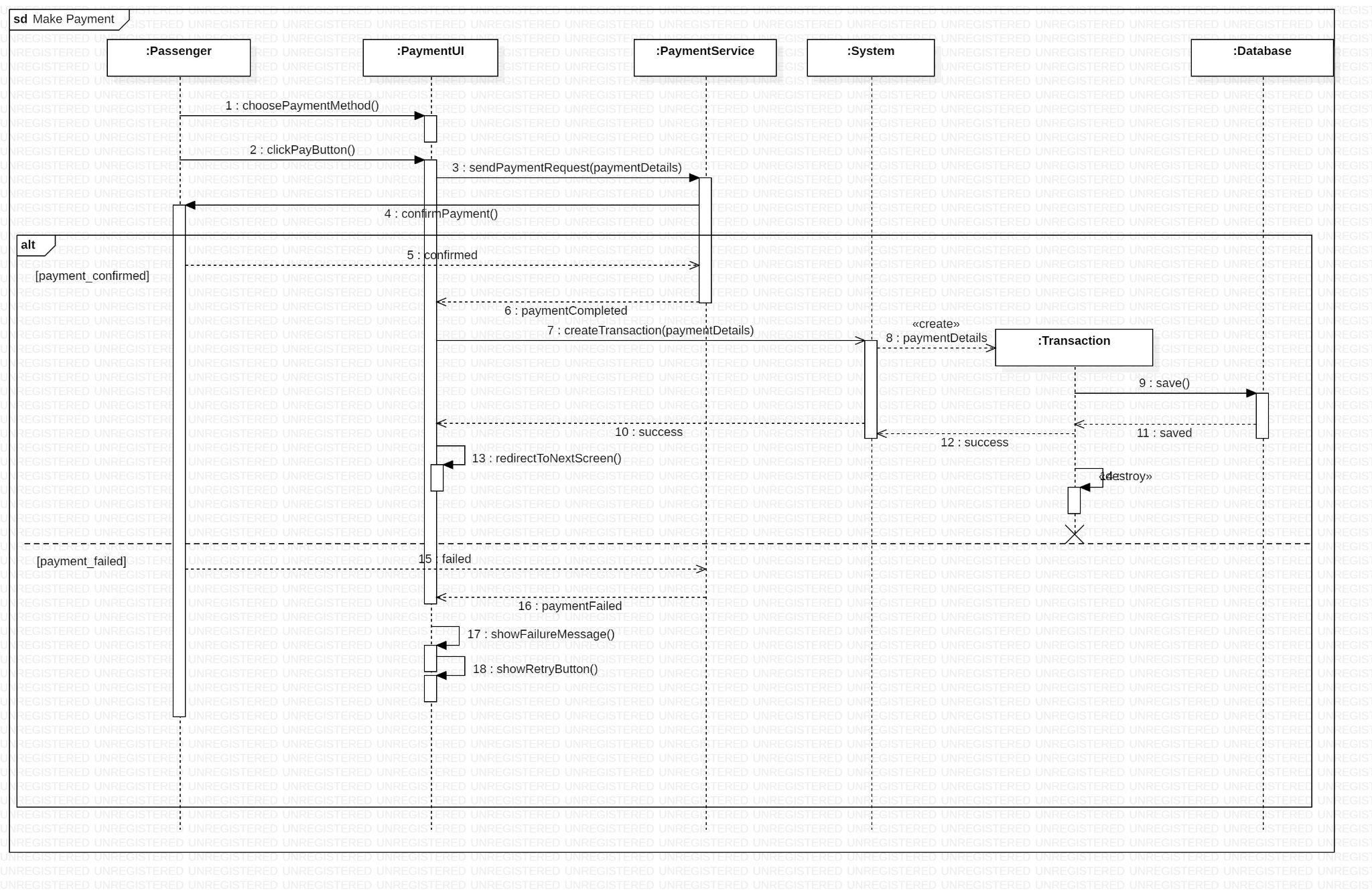
1. **View Transaction History**

****

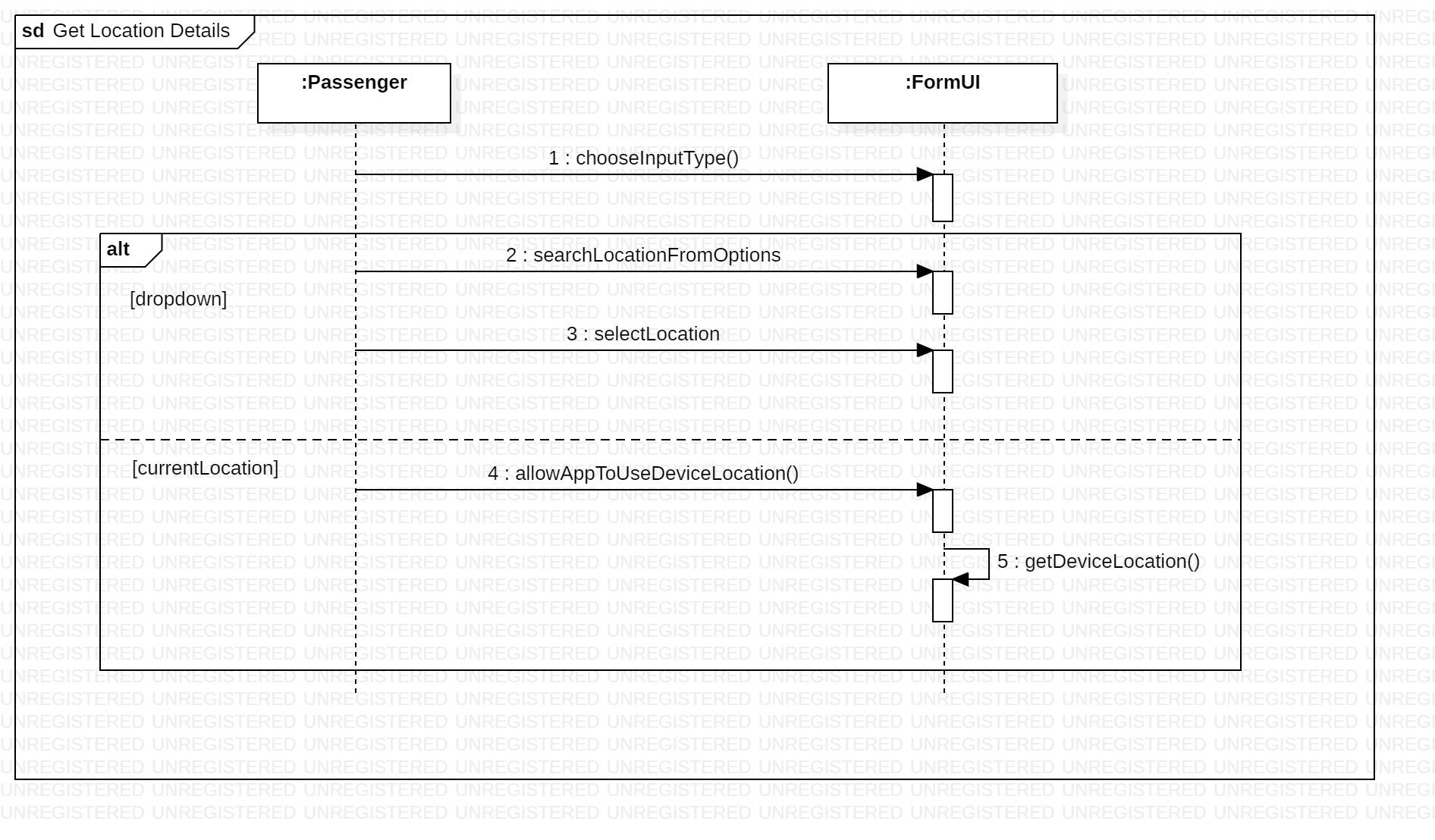
1. **View Driver Location**

****

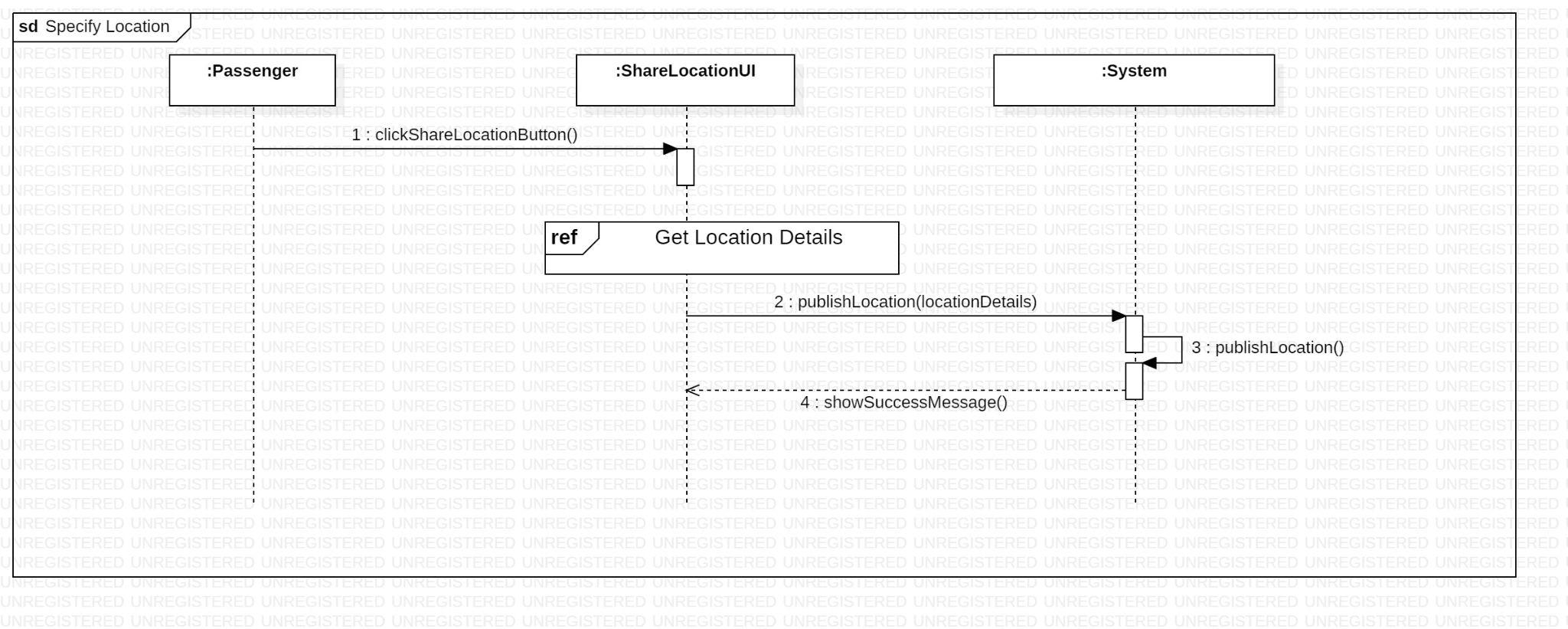
1. **Make Payment**

****

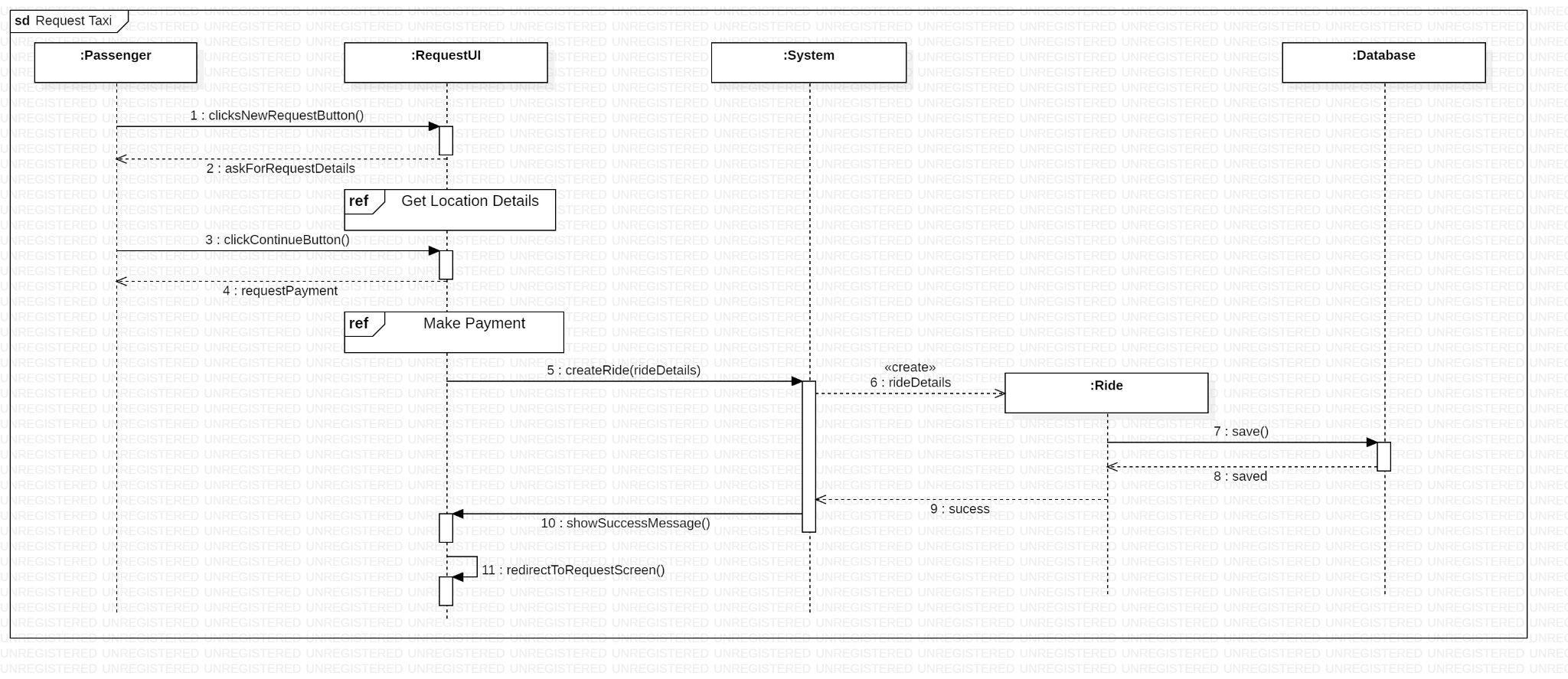
1. **Get Location Details**

****

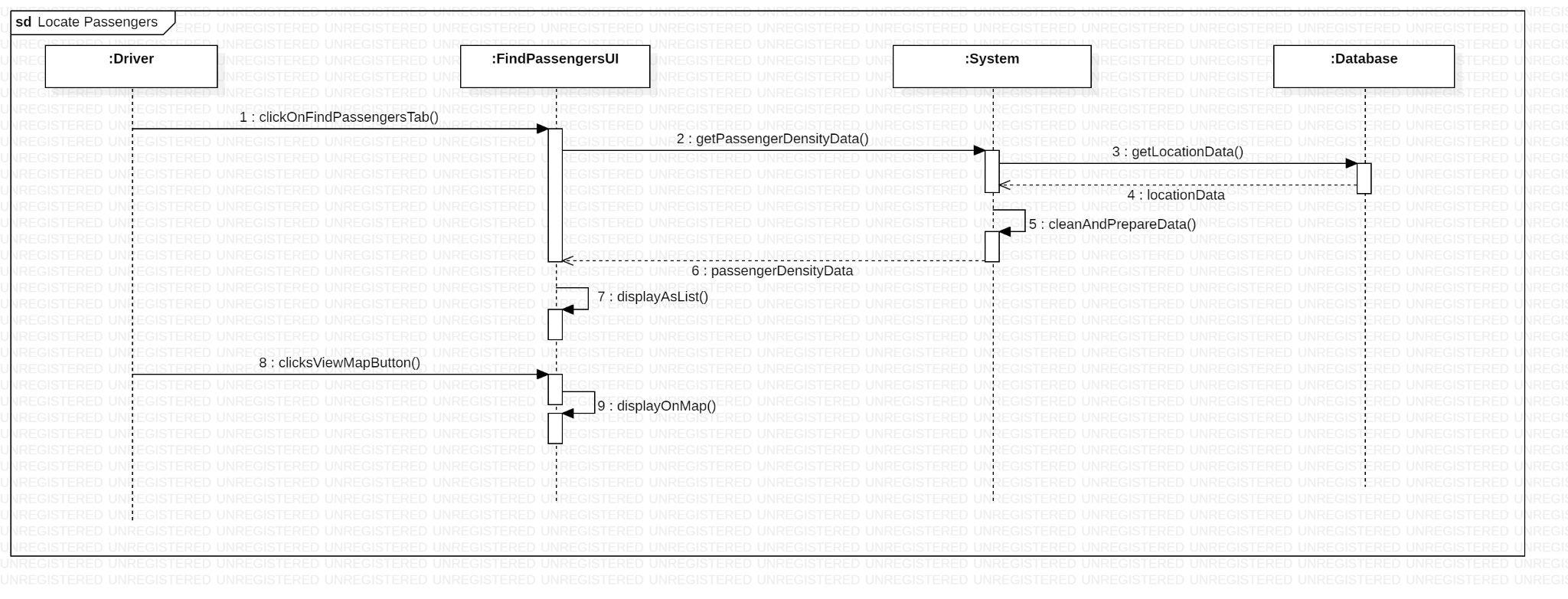
1. **Specify Location**

****

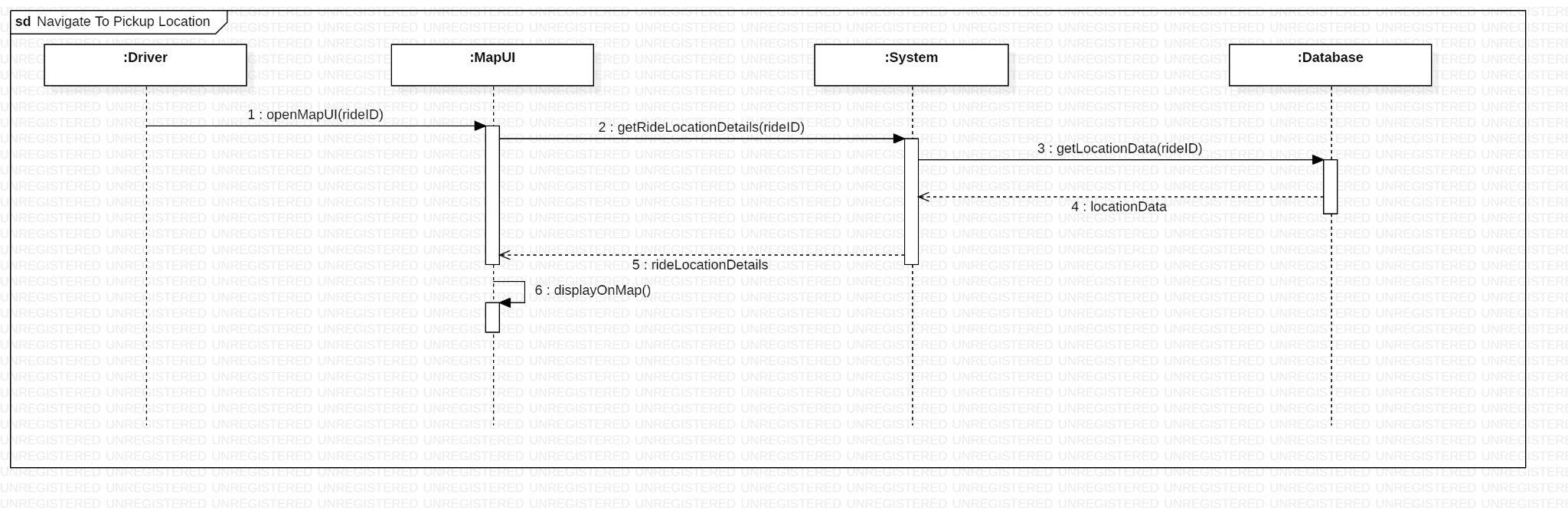
1. **Request Taxi**

****

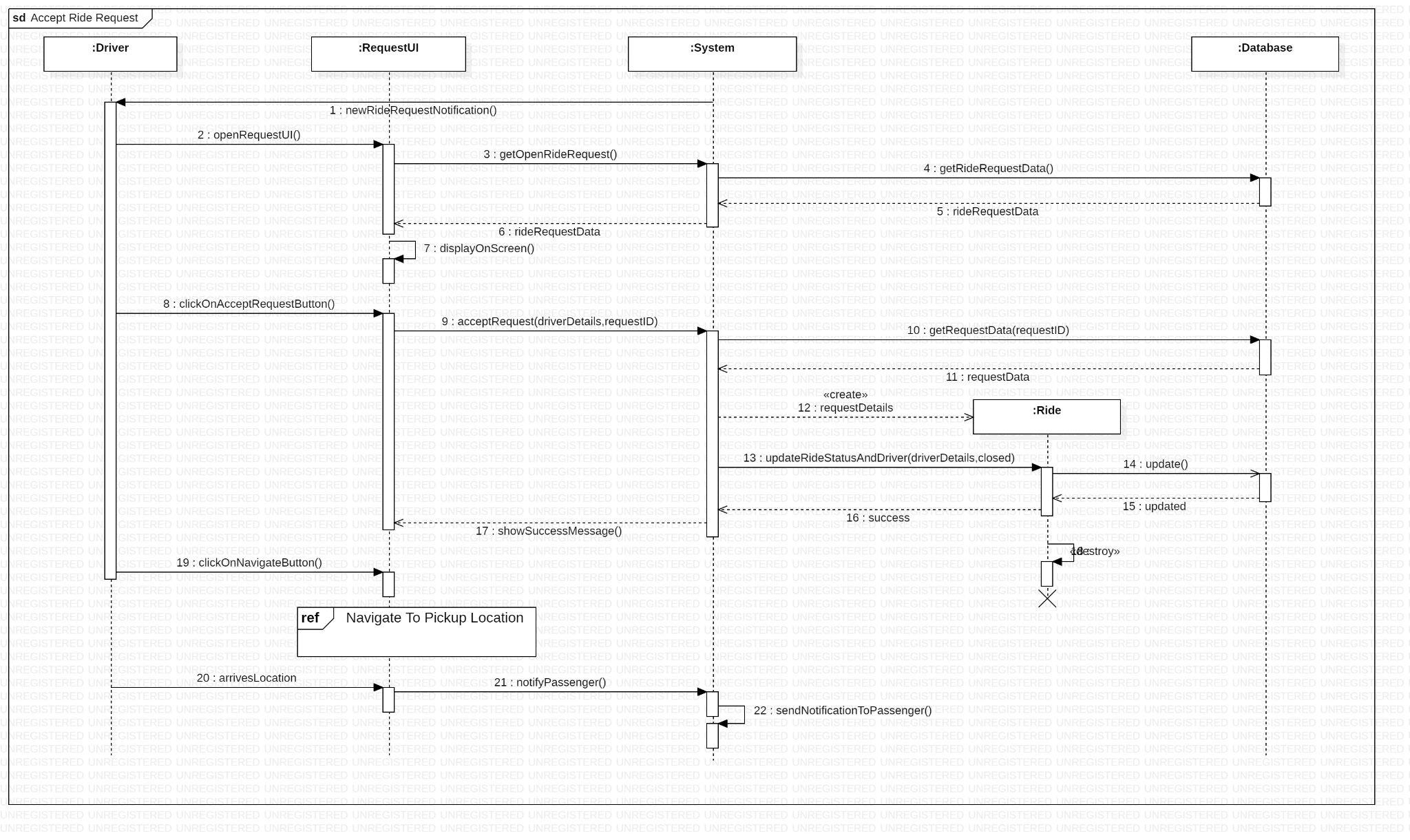
1. **Locate Passengers**

****

1. **Navigate To Pickup Location**

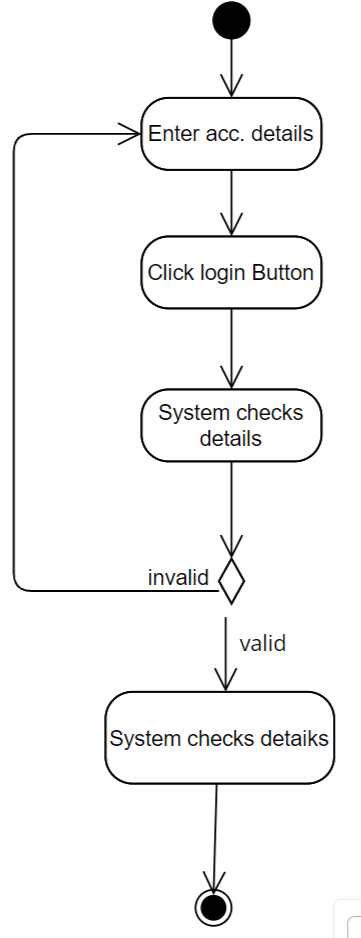
****

1. **Accept Ride Request**

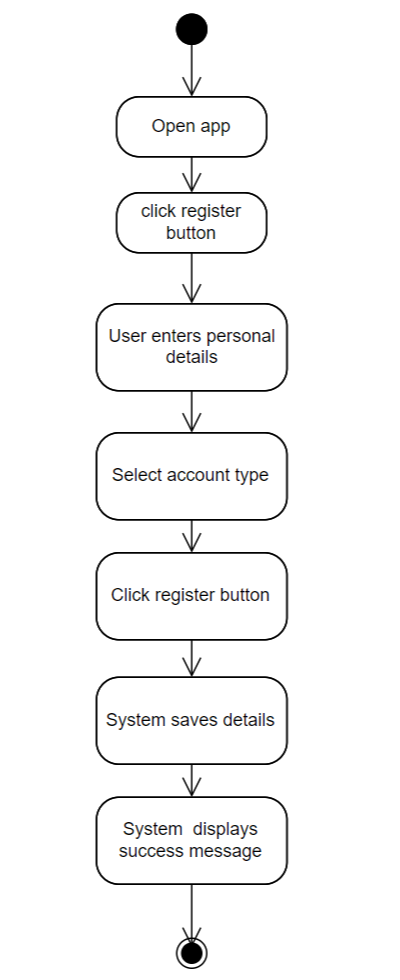
****

## **Activity Diagram**

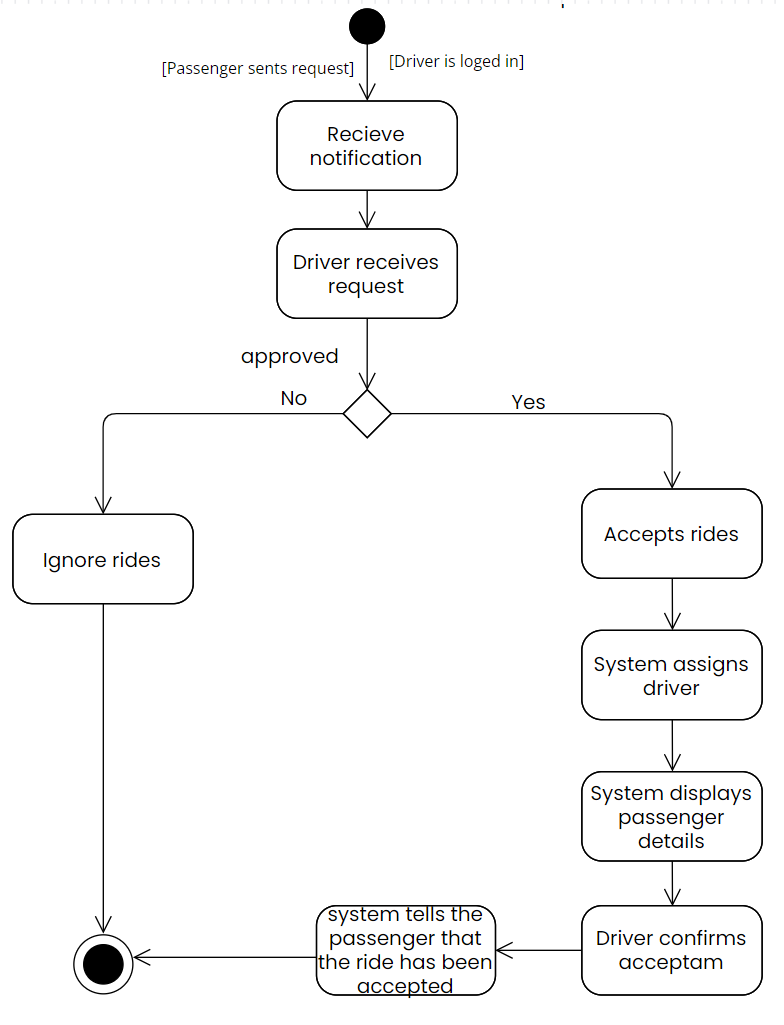
1. **Login**

****

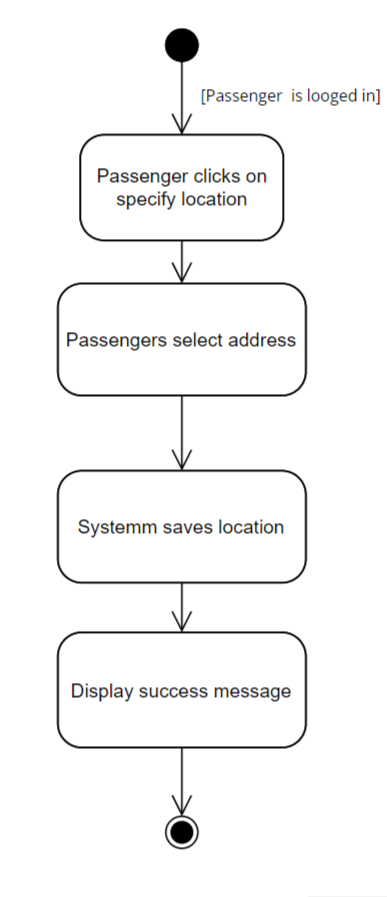
1. **Register**

****

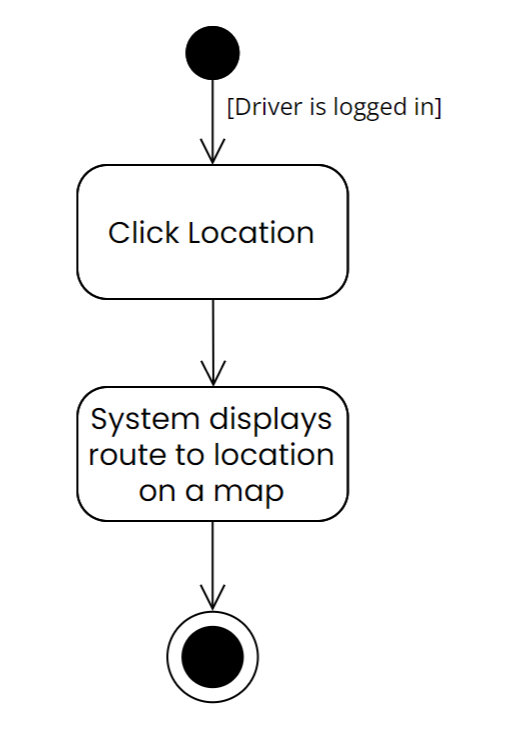
1. **Accept Ride Request**

****

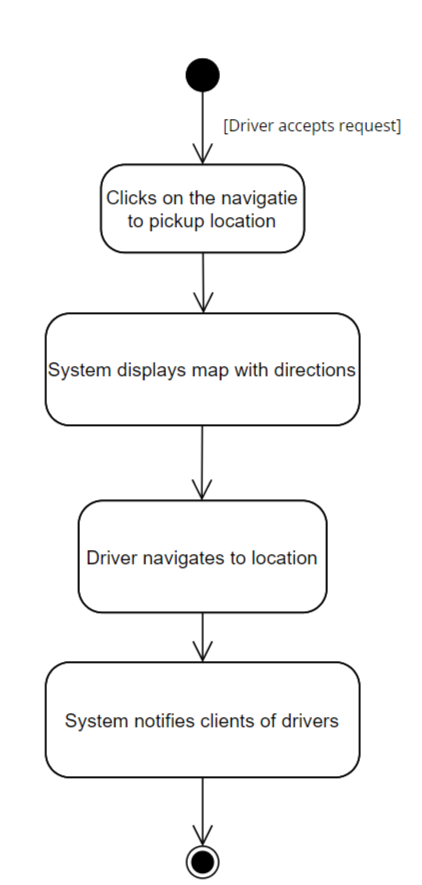
1. **Specify location**

****

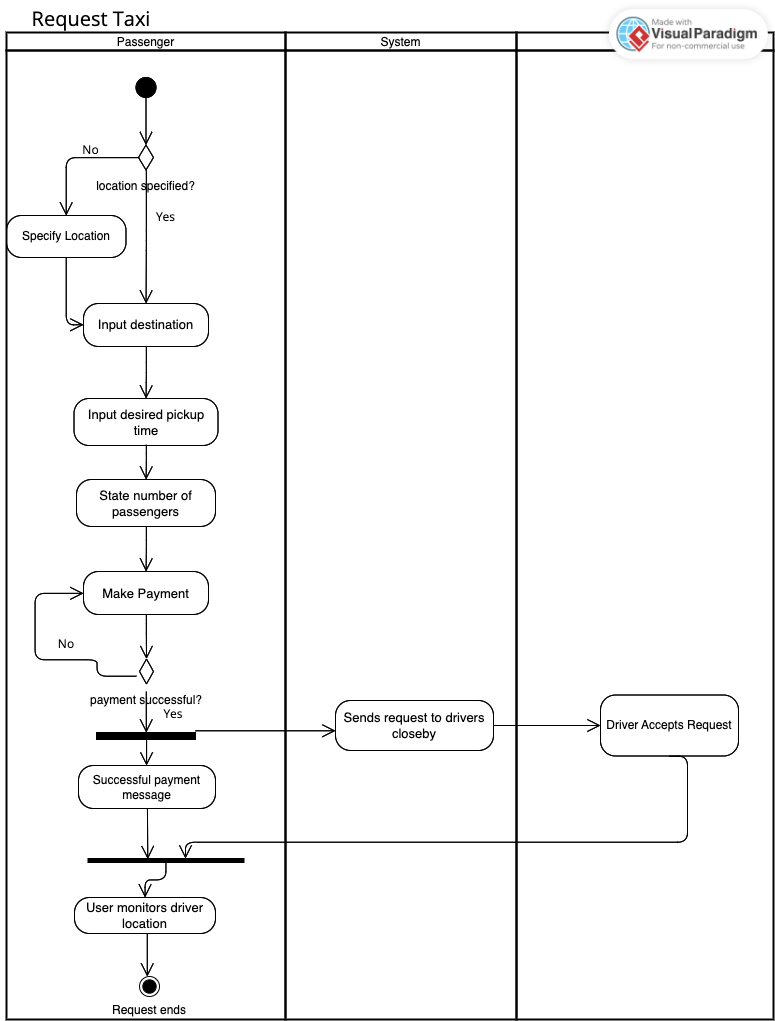
1. **Locate passengers**

****

1. **Navigate to pick up location**

****

1. **Request Taxi**

****

# 

# 

# **Conclusion**

In conclusion, the proposed system's development involves designing and developing a mobile application that utilizes UML diagrams to illustrate its different use cases. The aim is to provide a more efficient and secure taxi-hailing experience for both the drivers and passengers. The project's scope includes testing and deployment of the mobile application to the respective app stores. The proposed system has many benefits for both the passengers and drivers, including increased efficiency, reduced waiting times, and fuel consumption cost optimization. With the projected growth in the market for such mobile applications, the revenue model can be based on commission or subscriptions, resulting in a profitable business venture.