CHANGES :

1. **Objectives:** We redefined the objectives, making him more specific and following the recommendations from the workshop-1. In this way the objectives are the next:
   General objective:
   The main goal of this project is to optimize urban transportation by expanding mobility options beyond traditional public transport (buses, taxis, metro). By leveraging nearby connected devices such as smartphones, the project aims to reduce travel time and improve accessibility and convenience for pedestrians, enhancing overall city mobility.
   Specific Objectives:
   1.Identify limitations of current public transportation systems in terms of accessibility, coverage,and efficiency.
   2.Explore alternative mobility solutions (e.g., micromobility, ride-sharing, dynamic routing) that can be integrated through mobile platforms.
   3.Develop a system that uses real-time data from connected devices to suggest optimal travel routes for pedestrians.
   4.Improve user experience by making transportation services more flexible, intuitive, and accessible via smartphones.
   5.Evaluate the impact of the proposed solution on reducing travel times and increasing mobility options in urban areas.
2. **User stories:**We add a estimated time for the completion of the user stories:

Title: Trip Request	**Priority:** High	**Estimate:** 2 weeks

User Story
As a passanger who has requested a trip, i want to be assigned an available driver so that i can start my trip without delay.

Acceptance criteria
Given that i am using the app
When i enter my pickup and drop-off location and request a trip
Then the system finds available drivers and display the trip details

Title: Driver Assignment	**Priority:** High	**Estimate:** 2 weeks

User Story
As a registered passenger, i want to request a trip by specifying my location and destination, so that i can be transported safely and conveniently

Acceptance criteria
Given that i have requested a trip
When the system finds an available driver
Then i recieve a notification with the driver's details and estimated arrival time

Title: Trip Tracking	**Priority:** Medium	**Estimate:** 1 week

User Story
As a passenger on a trip, i want to see the real-time location of the vehicle so that i can estimate my arrival time at the destination.

Acceptance criteria
Given that my trip has started
When i open the app
Then i can see the driver's location on a real-time map

Title: Service payment	**Priority:** High	**Estimate:** 2 weeks

User Story
As a passanger who has completed a trip, i want to make the payment through the app so that i can complete the transaction quickly and securely

Acceptance criteria
Given that my trip has ended
When i select a payment method and confirm the transaction
Then the payment is processed, and i receive a receipt

Title: Service rating	**Priority:** Medium	**Estimate:** 1 week

User Story
As a passanger who has completed a trip, i want to rate the driver and the service so i can contribute to improving service quality

Acceptance criteria
Given that my trip has ended
When i access the rating option and leavy my feedback
Then the systems saves my rating and comment for future reference

Title: Driver's management	**Priority:** High	**Estimate:** 2 weeks

User Story
As an administrator, i want to check and modify drivers information, so that i make sure they're available and operating correctly

Acceptance criteria
Given that I access the driver dashboard
When i select a driver
Then i should be able to view and edit their information, availability and vehicle.

Title: Trip management	**Priority:** High	**Estimate:** 2 weeks

User Story
As an administrator, i want to have access to detailt about ended and ongoing trips, so that i can monitor and resolve possible incidents.

Acceptance criteria
Given that I access the trips section
When filtered by date, user or driver
Then I can see the current and completed trips with details.

Title: Pay management	**Priority:** High	**Estimate:** 2 weeks

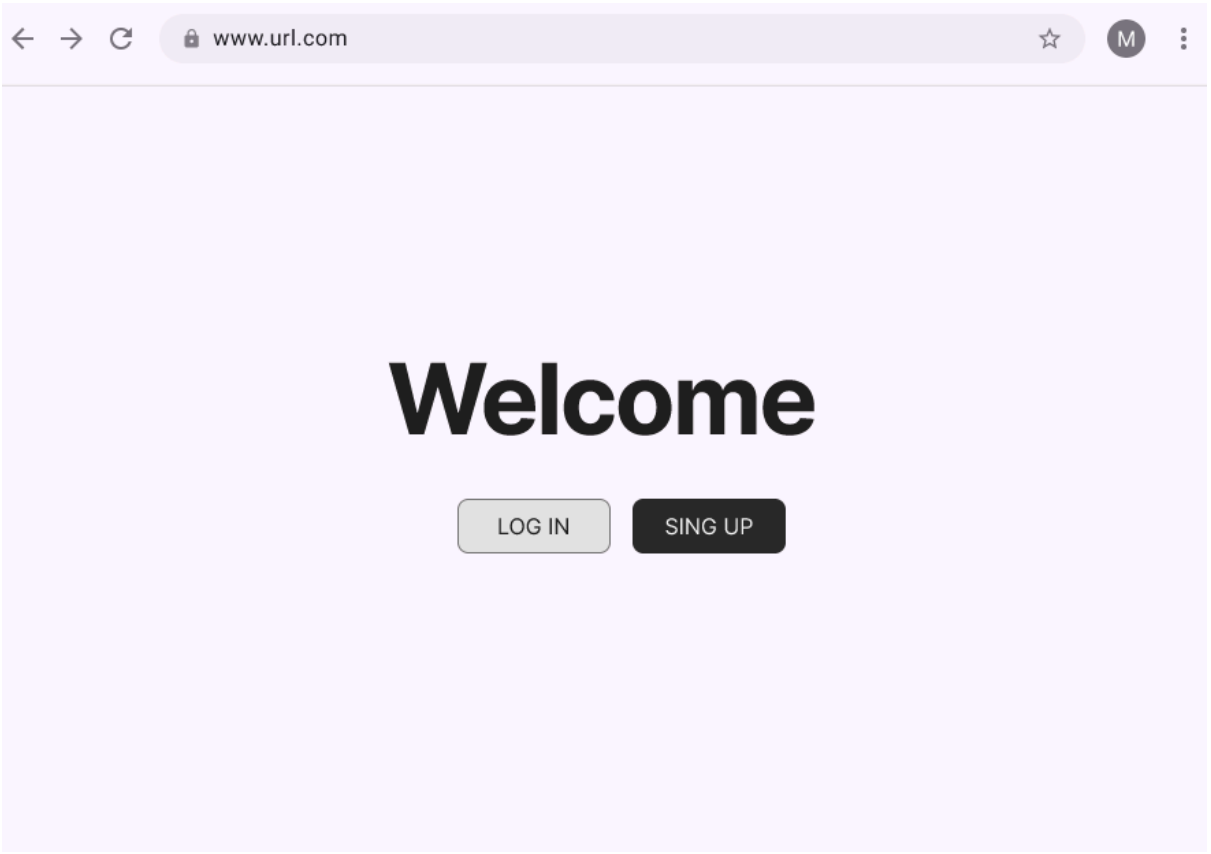
User Story
As an administrator, i want to review and manage payments made by users to ensure that all transactions are in order.

Acceptance criteria
Given that I am in the payments section
When I search by user, date or status
Then I should see the list of corresponding transactions

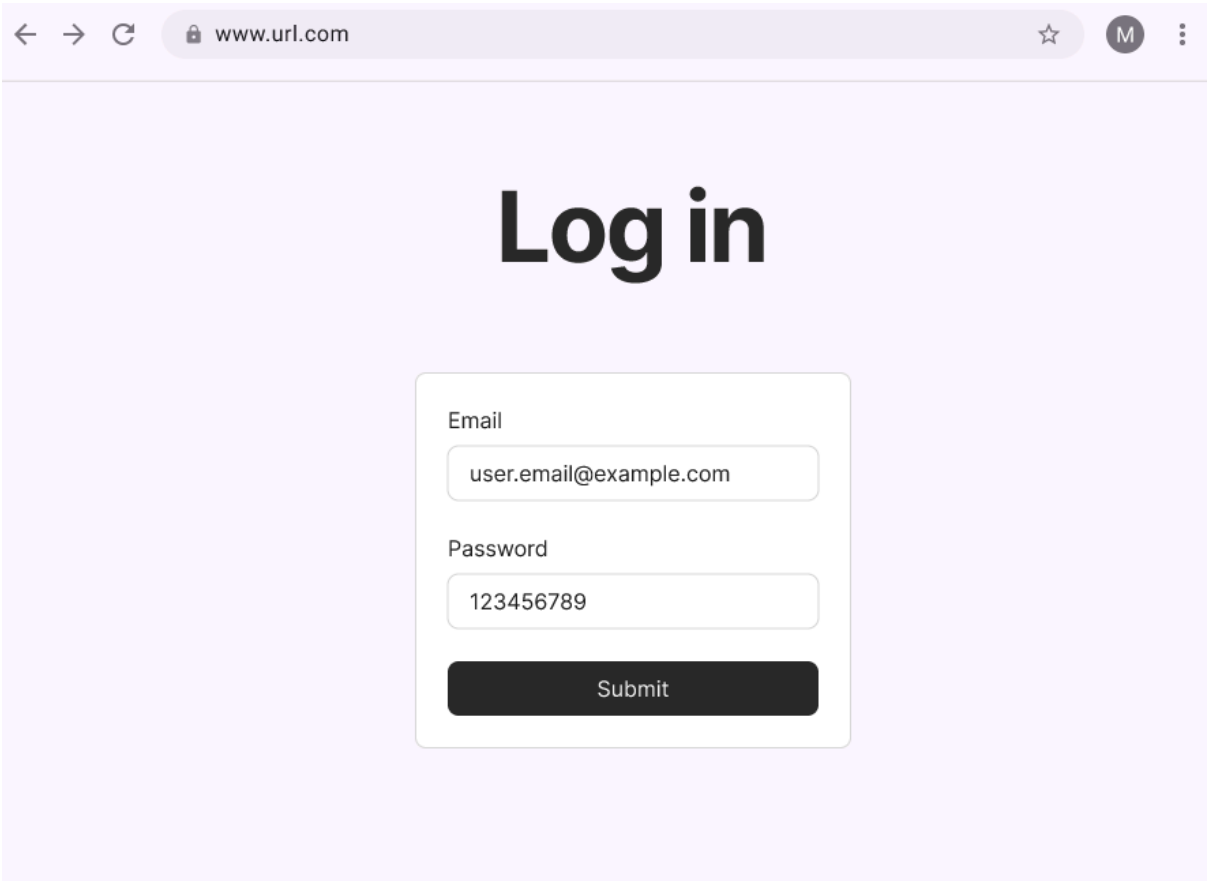
Title: Review management	**Priority:** High	**Estimate:** 2 weeks
User Story As an administrator, I want to be able to review and manage the ratings that users leave for drivers, to take action if there are negative reports.		
Acceptance criteria Given that I access the ratings When I select a driver Then I should see all the ratings and comments received.		

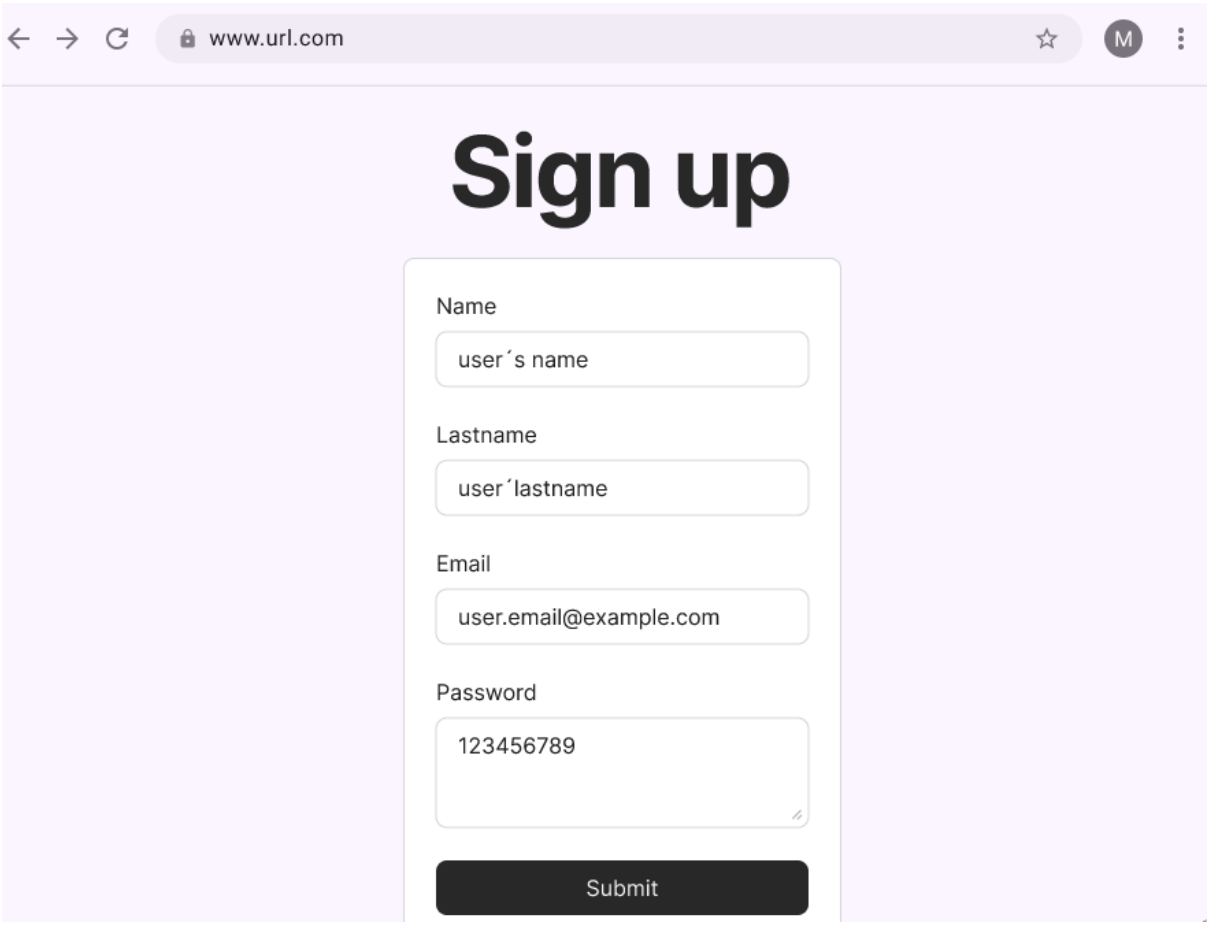
Title: Statistic report	**Priority:** High	**Estimate:** 2 weeks
User Story As an administrator, I want to generate reports about the trips, earnings, user ammount and active drivers, so that i analyze the platform's performance		
Acceptance criteria Given that i need performance information When generating a weekly or monthly report Then the system should display metrics such as number of trips, revenue and active users.		

3. **Mockup:** We remake the mockup because the old design was made for a phone interfaz so we made it for a computer:
    1. **Welcome:** In this one the user will choose between log in and sign up:

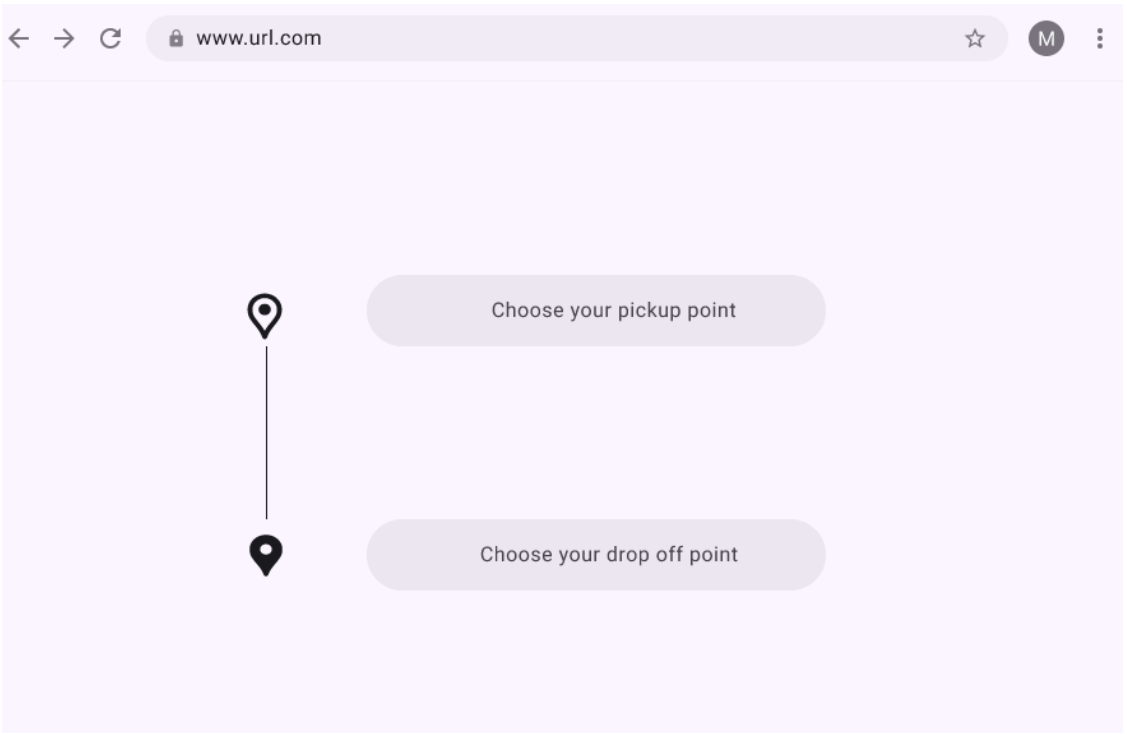


2. Log in and Sign out: After choose between log in and sign up, the user will be redirected to a new page that will request him a information to continue:

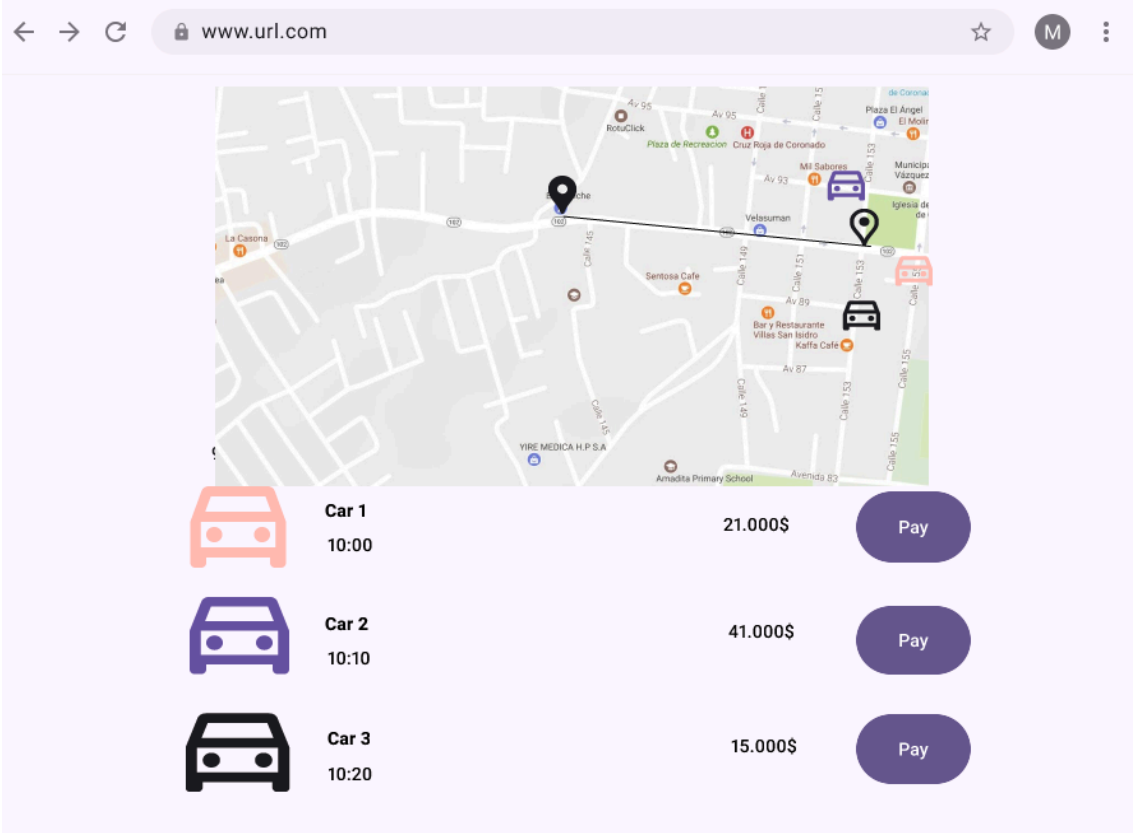




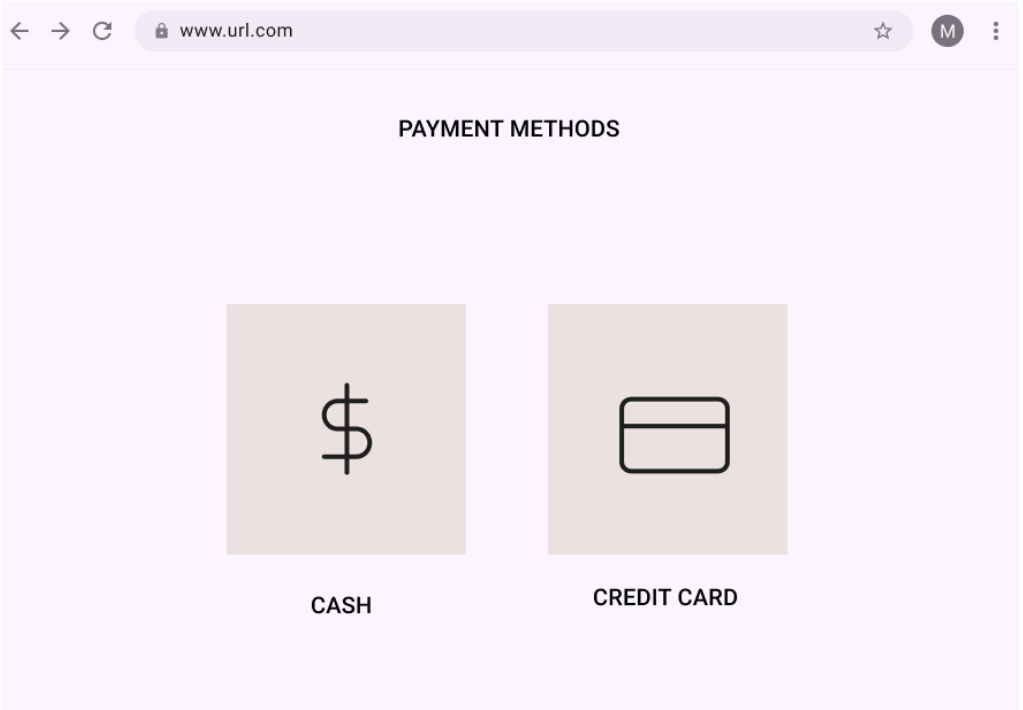
3. Choose trip: After ingrese to the application the user will see a options to put the location where he will be picked up and the location where he want to go:



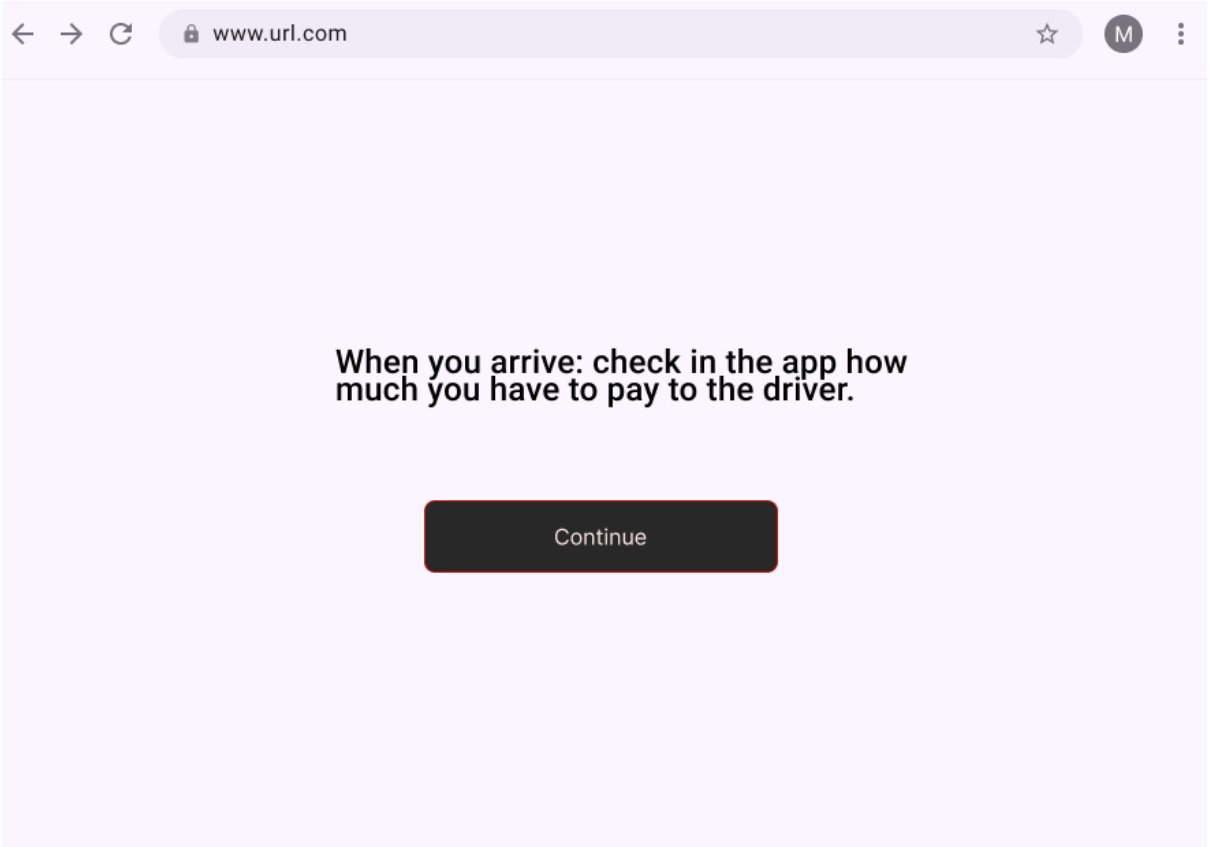
4.Choose driver: When the user put the locations the program show the cars that are around the zone and the value of the trip, that depends of the distance and the type of vehicle:



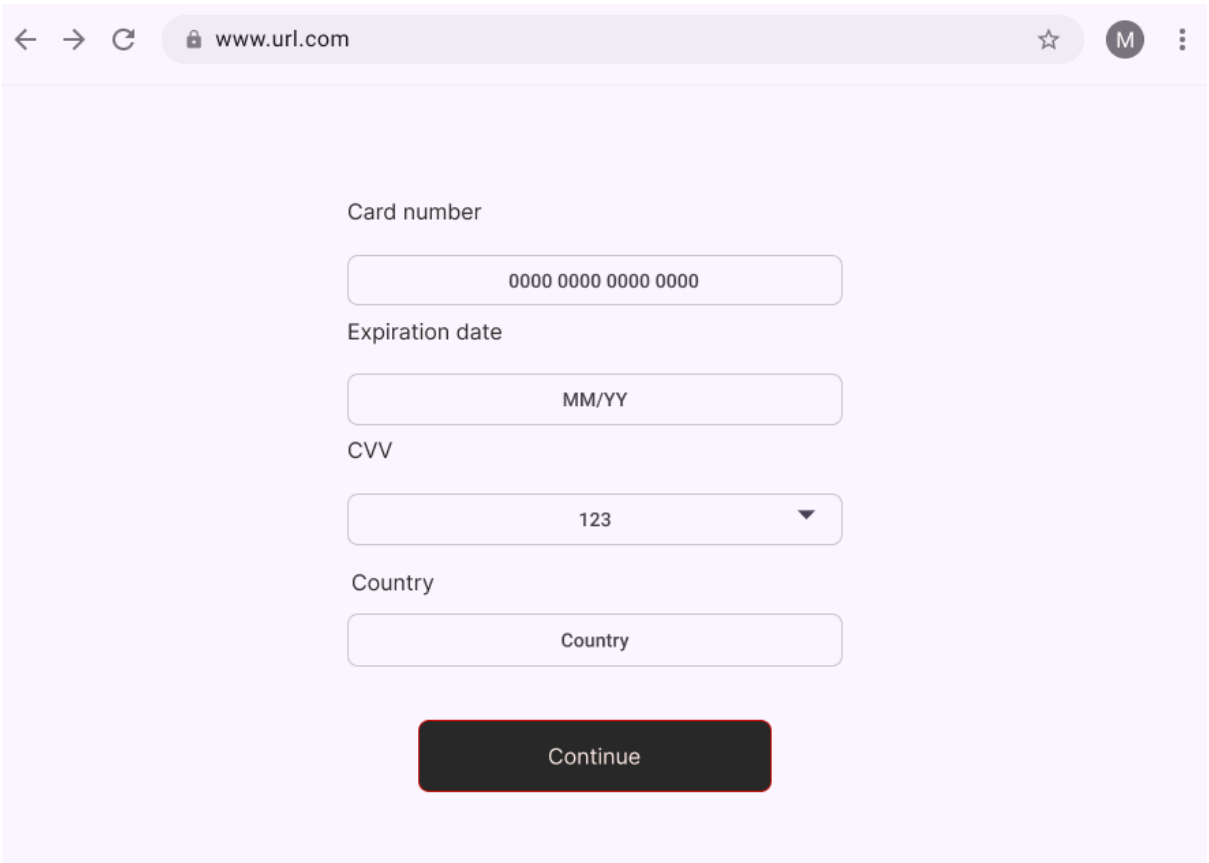
5.Choose Payment method: When the drip is ready, the user has to decide between the payment methods, if he decide the credit card method he have to put the information to complete this method, but if he decide to pay with cash the program will show a message in the screen:



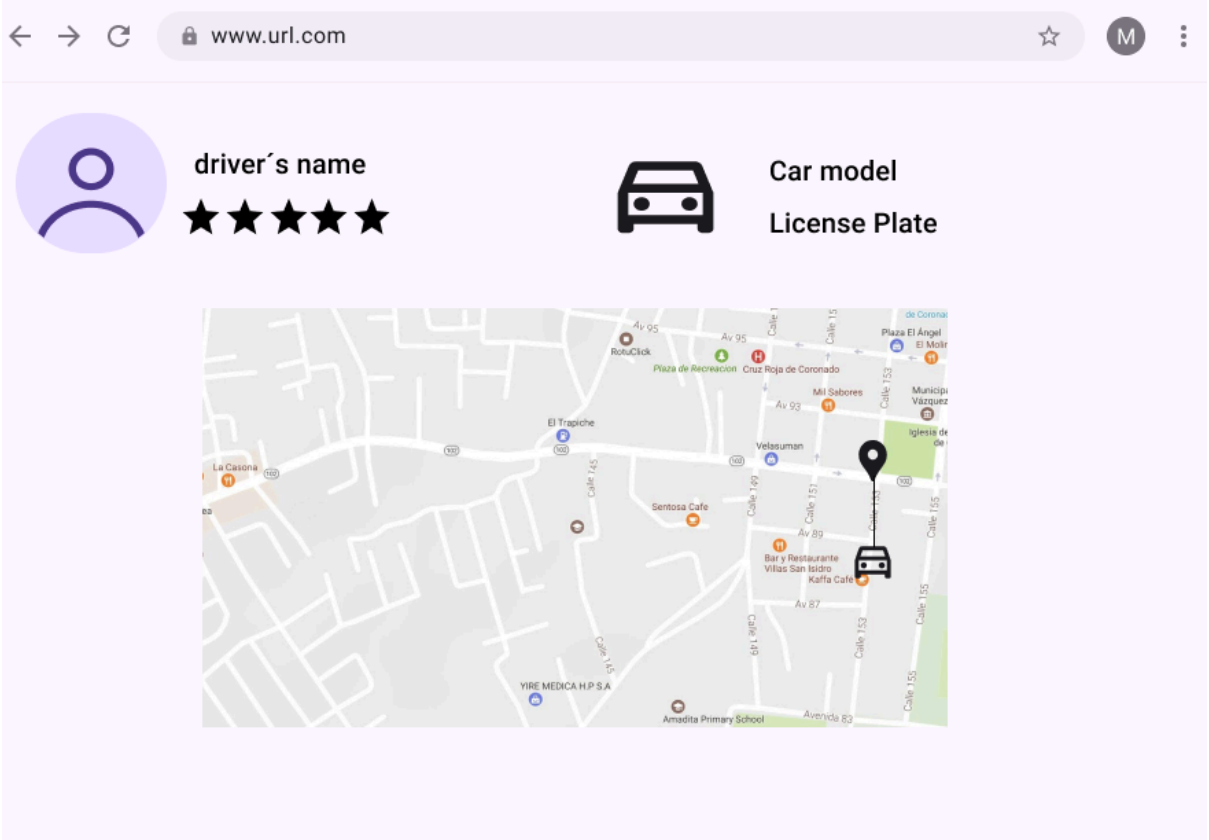
If he pays with cash:



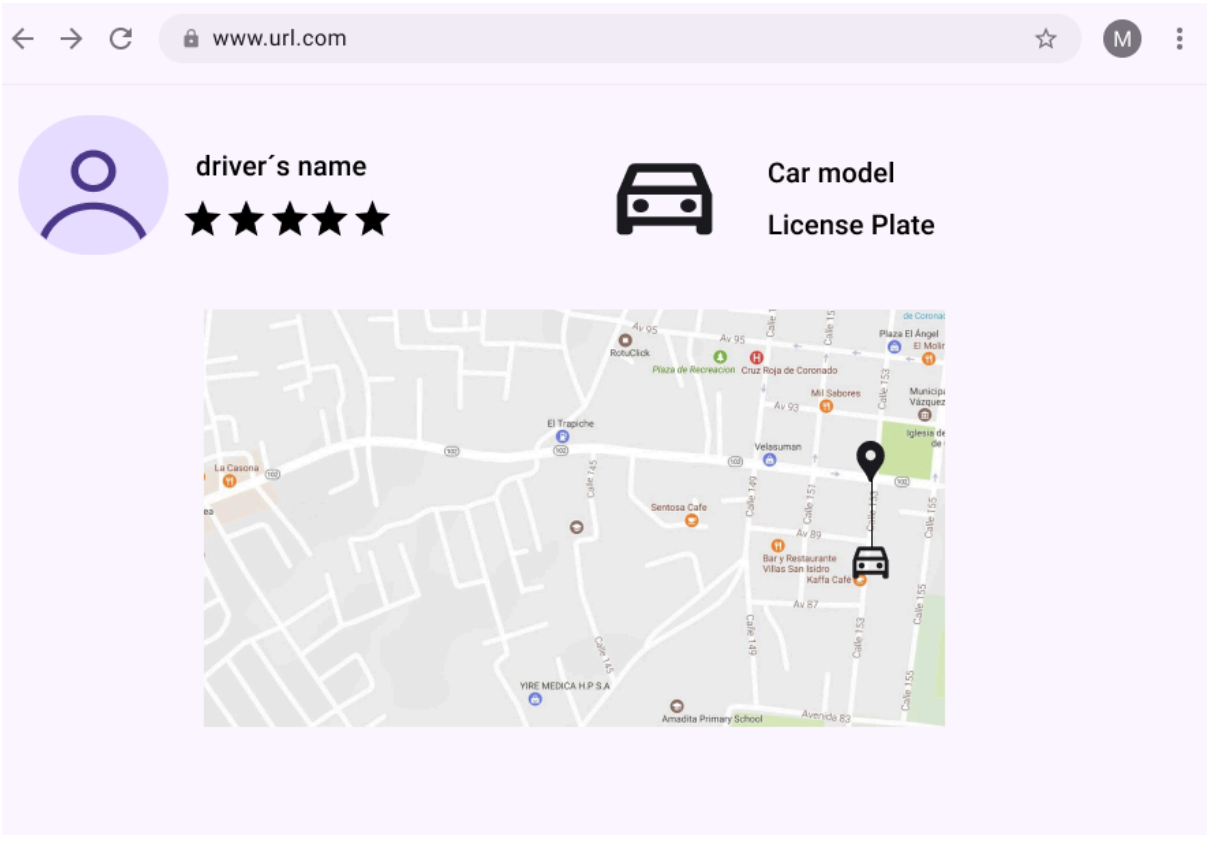
If he pays with credit card:



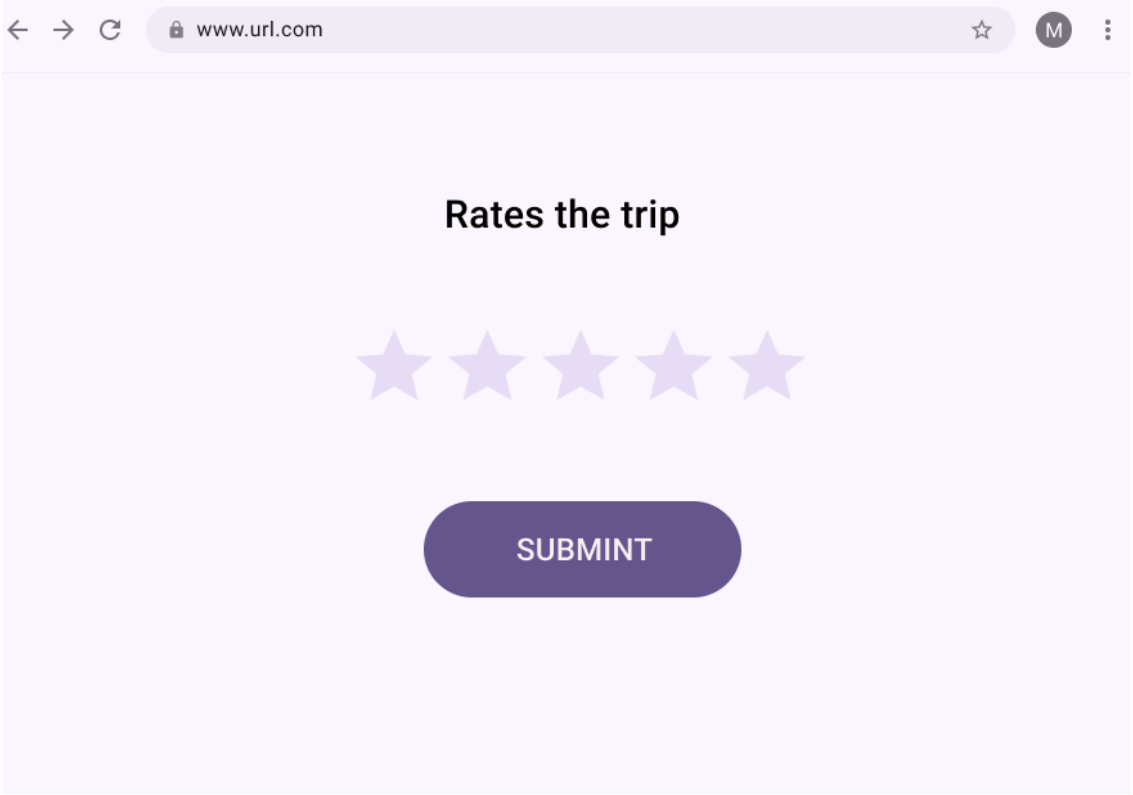
6.Driver´s information and pick up: At this point the user will see the information of the driver and when he will picked up:



7.Trip: The user will see the trip that the driver will make and the estimate time to arrive:



8.Rate the trip: When the trip is completed the program will request to the user rate the experience of the trip:



4. **Update the CRC:**With the implementation of the SOLID we redefined the class diagram in the next way:

Rating	
Responsibillities	**Collaborators**
-Store user rating of drivers and trips -Provide ratings to users and drivers	-User -Driver

LogIn	
Responsibillities	**Collaborators**
-Register and authenticate on the application -Logout unactive users	-User

Car	
Responsibillities	**Collaborators**
-To store vehicle information(License plate, model, color, etc.) -Assign a car to a driver	-Driver

CreditCard	
Responsibillities	**Collaborators**
-Process user payments -Updates user's balance	-User -Trip

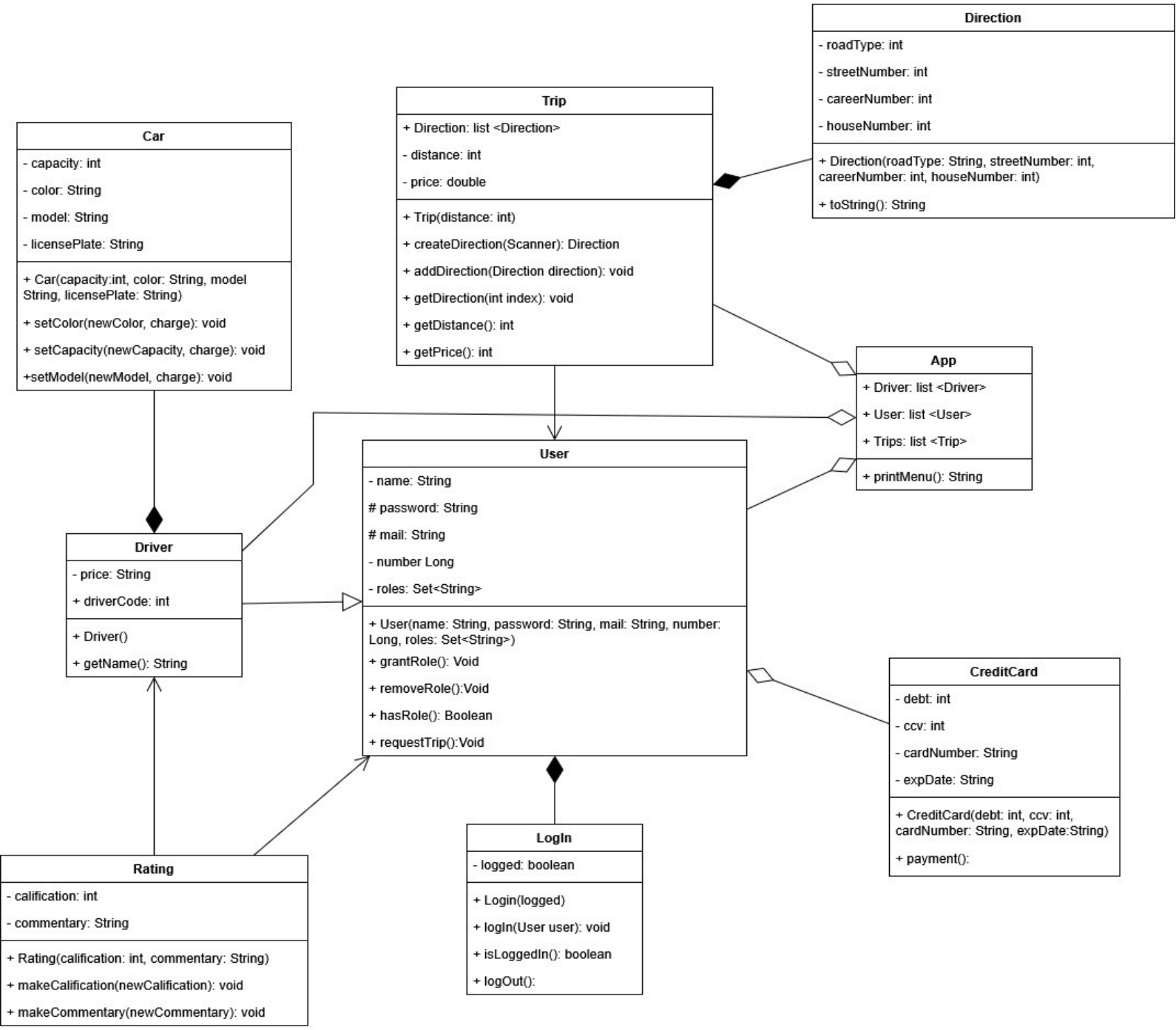
User	
Responsibillities	**Collaborators**
-Request a trip -View driver and trip information -Pay for the trip	-Driver -Trip -LogIn

Driver	
Responsibillities	**Collaborators**
-Receive travel requests -Accept or decline travel requests -Perform trips	-Car -Trip -User

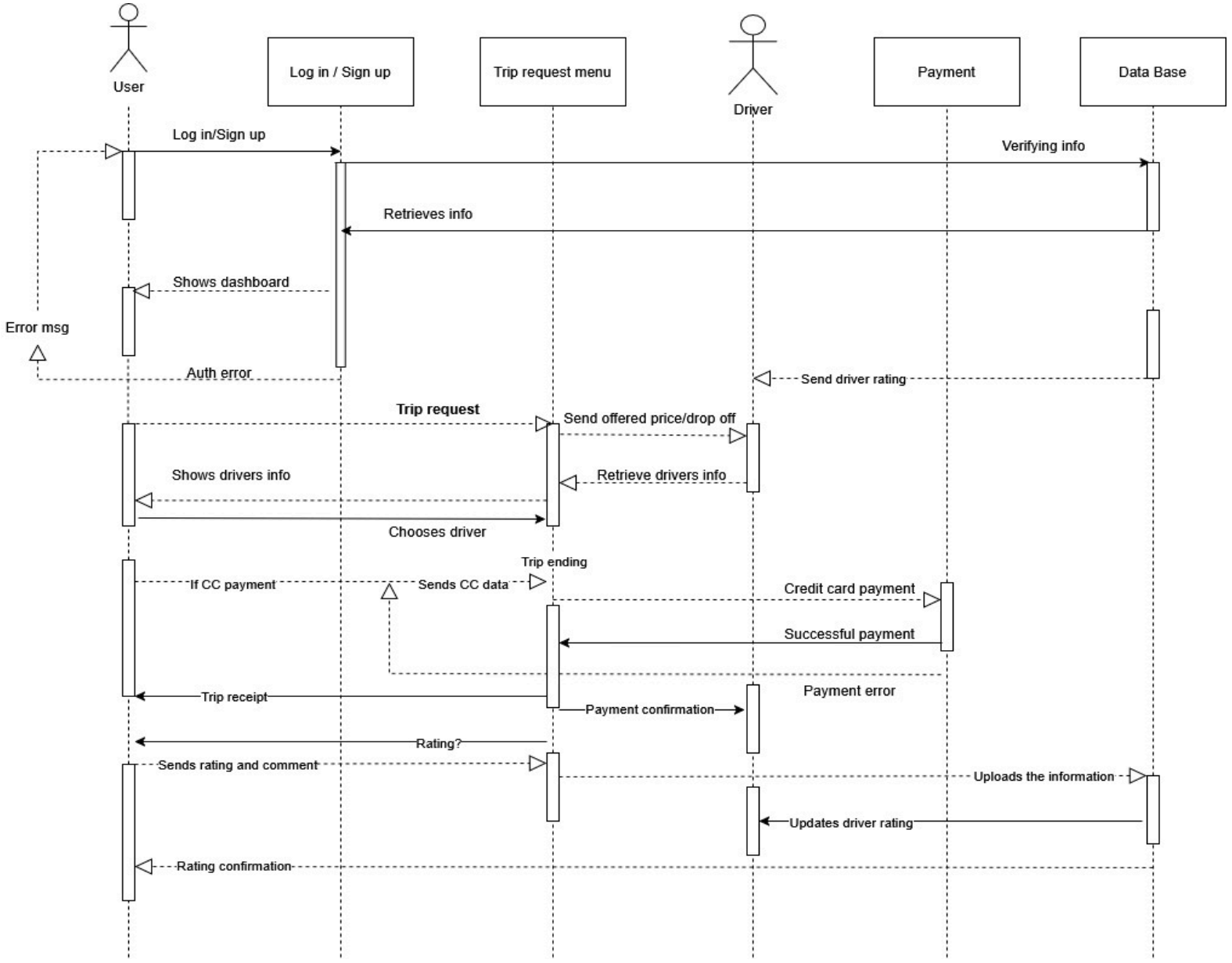
Trip	
Responsibillities	**Collaborators**
-Store trip information(distance, price, etc) -Assign a driver to the trip -Update trip status	-Car -Trip -User -Direction

Direction	
Responsibillities	**Collaborators**
-Request departure point -Request arriving point -Store and transfer ubication	-Trip

5. **Diagram class:** The diagram show the type of interaction between the class:



6. **Sequence diagram:** This diagram show the transaction flows:



SOLID PRINCIPLES:

1. Single Responsibility: We divide the responsibility of trip adding direction, because we consider that the receive and analyze of the input that the user makes to introduce the direction have to be made by the class Direction and not by Trip. This makes the class Trip focus on getting the price and the distance. Also we maintain the class that we introduce in workshop 1 and 2, because they apply this concept.
2. Open/Closed: Keeping with the same example of the first principle we decide to not extend the class Trip with the analysis of the direction and decide to extend him with the class Direction.
3. Liskov Substitution: The liskov substitution is not used in the design because there are no classes that have similar methods. The closest to have similar methods that can be put in an interface is driver and user with the methods that interact with the

role of the user. But these methods have the same logic in driver and user, allowing inheritance and dont use an interface

Following with the argument of the liskov substitution make that the Interface segregation and the Dependency Inversion can be applied.

CODE SNIPPET:

We select the Direction class . This decision was taken because in the process of remake the design taking in care the SOLID we think that the receive and the analyze of the direction have to be made by another class, guarantees the Single Responsibility for the class Trip and the Open/closed because we extend trip with Direction and doesn't extend the class Trip with more methods:

```
import
java.util.Scanner;
public        class
Direction {
    protected int  roadType;
    protected            int
    streetNumber;  protected
    int        careerNumber;
    protected            int
    houseNumber;
    public   Direction(int   roadType,   int   streetNumber,   int
careerNumber, int houseNumber){
        this.roadType =roadType;
        this.streetNumber=street
        Number;
        this.careerNumber=career
        Number;  this.houseNumber
        = houseNumber;
    }
    Scanner input = new Scanner(System.in);
    public int getStreet(int streetElection) {
                if (1 <= streetElection && streetElection <= 246)
                    { streetNumber = streetElection;
            }
        return this.streetNumber;
    }
    public int getCareer(int careerElection){
                if (1 <= careerElection && careerElection <= 161)
                    { careerNumber = careerElection;
            }
        return this.careerNumber;
    }
    public int getHouseNumber(int houseNumerElection) {
                if (1 <= houseNumerElection && houseNumerElection <=
                    99) { houseNumber = houseNumerElection;
            }
        return houseNumber;
```

```
    }
    public String toString(String roadType,int streetNumber ,int
careerNumber,int houseNumber){
        if(this.roadType==1){
            return "This is the direction: Cl" + this.streetNumber + "
#" + this.careerNumber + "-"+this.houseNumber ;
        }
        else {
            return "This is the direction: Cra" +" " +this.careerNumber
+ " #" + this.streetNumber + "-" +this.houseNumber ;
        }
    }
}
```