Politecnico di Milano, A.Y. 2016/2017
M.Sc. Degree Programme in Computer Science and Engineering
Software Engineering 2 Project

Design Document

Philippe Scorsolini,
Lorenzo Semeria,
Gabriele Vanoni

10th December 2016

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to give the detailed structure of PowerEnJoy software system.
So we try to give developers a clear representation of:

- The high level architecture of the system.
- The design patterns applied in order to achieve our specific necessities.
- The main components and the interfaces they provide.
- The Runtime behaviour.

1.2 Scope

PowerEnJoy project aims to provide users and operators with means to use the system services they need and they are supposed to have access to. It also provides an API for external services to access the system with "operator" rights, in order to allow call center operators to interface with the system.
The system allows:

- Users to manage their personal data through both a web and a mobile app and Active Users to manage reservations.
- Callcenter's operators to manage assistance tickets via the API. The tickets will tne be managed internally by PowerEnJoy.
- PowerEnjoy's operators to manage the open assistance tickets, take them in charge and update car's and user's data accordingly.

The system architecture shall guarantee future proof scalability and allow subsequent improvements and general reliability.

1.3 Abbreviations, Definitions and Acronyms

1.3.1 Abbreviations:

- Gn: the n-th Goal
- An: the n-th Assumption
- Rn: the n-th Requirement

1.3.2 Acronyms

- CC: Credit Card
- DL: Driving Licence
- AU: Active User

1.3.3 Definitions

- Visitor: person that may not be registered to the system or not logged in.
- User: a registered and logged in Visitor, that may be still waiting for his information to be verified.
- Active User: a User whose data (CC, DL) have been verified. (Shares all User's characteristics)
- Safe Zone: predefined zones where parking is allowed, parking is forbidden in any other zone.
- Park: park the car in the safe zone and terminate the rental.

2 Architectural Design

2.1 Overview

The system adopts a three tier architecture with a thin client represented by web and mobile app, which allows users and operators to access through a GUI the different functionalities of the system accordingly to the type of client used. These clients are both managed by a specific server-side ClientHandler offering the same interface to other server-side services and handling appropriately the communication with the two clients adopting in both cases an asynchronous implementation of the RESTful APIs over HTTPS in JSON format in order to achieve complete freedom of development-specific choices on both sides.
The second tier adopts a microservice oriented architecture with shared database that, taking into account the unknown but supposedly not massive load of the system in the near future, reduces the need of synchronization between services. This allows to keep the structure as simple as possible, allowing ibetter performing solutions such as “database per service” or “schema per service” to be implemented when needed. This tier will also manage the communication with the cars and the power station around the city through OpenVPN and MQTT protocols.
The third tier provides the previous one the necessary abstraction on the storage technology chosen and will manage the concurrent access to the databases. In case a PaaS hosting service is chosen, this tier will be managed by the provider.
The second and third tier are designed to be deployed on the cloud in order to take advantage of the “scale on need” possibility it gives. For this reason it has been designed as a micro service architecture.

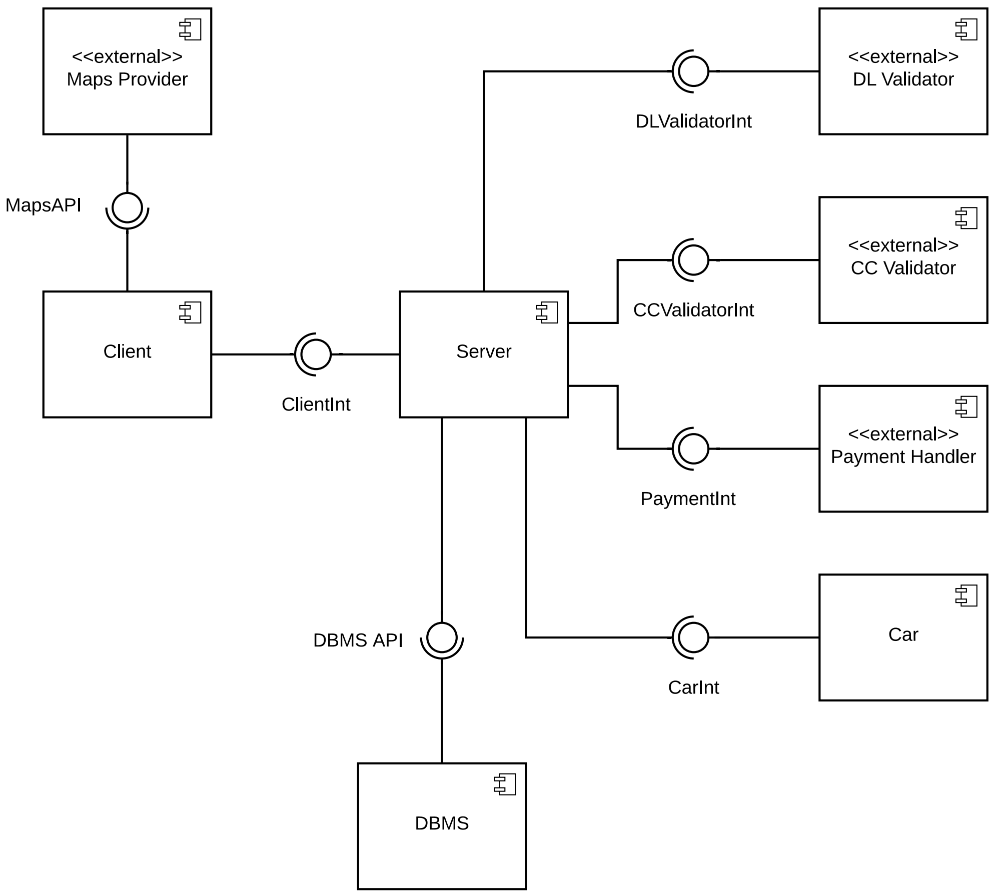


Figure 1: Component View of the server side deployed

2.2 Component View

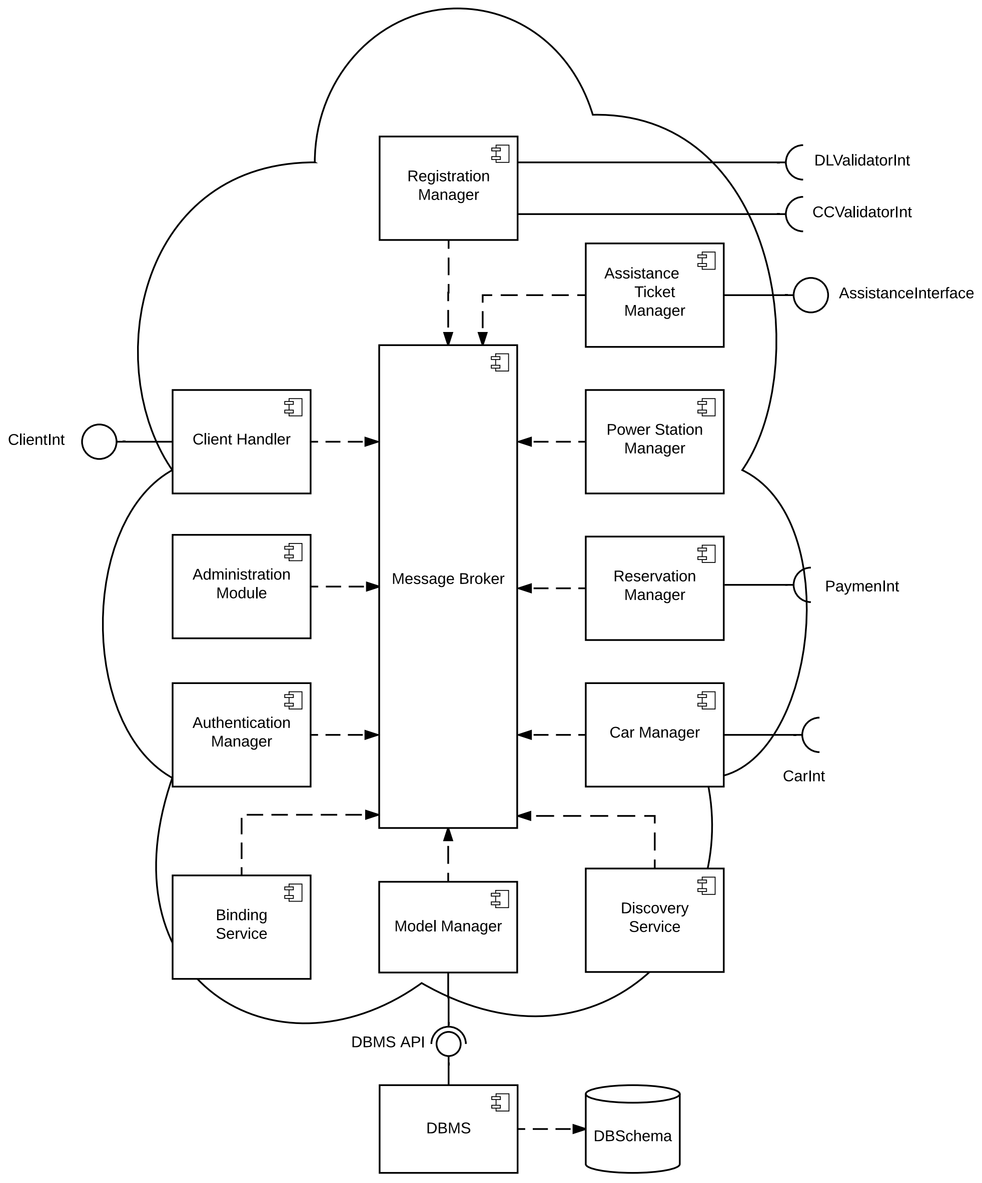


Figure 2: Component View of the server side single services

As previously said the system has been designed in order to respect the principles of microservice architecture. Therefore every component will be stateless, independently deployable, able to deal with a specific aspect of our business domain and to hide to the other services any kind of implementation detail.
The communication between these microservices will use AMQP (Advanced Messaging Queue Protocol) and will be managed by a centralized message broker (such as RabbitMQ or ActiveMQ). When a new instance of a service is created the message broker will be automatically connected to the new service and will publish its existence to the other services through the “discovery service”, then the “binding service” will take charge of setting up the correct queue binding rules with the message broker. That will allow some advanced traffic routing features including per-version weighting and elastic load

balancing on the different instances of the same service.
The hereunder specified components could be in the future divided in much more granular services in order to decouple even more their functionalities:

- Client Handler: orchestrates the needed services for the clients
- Discovery Service: gives the other services the references that allow them to communicate with each other
- Binding Service: manages the queue binding for the services
- Authentication Manager: checks whether or not a certain user is allowed to do a certain action
- Registration Manager: manages the registration of the new users and the update of the registered users data if needed
- Administration Module: grants PowerEnJoy's operators access to some specific features needed to manage the assistance tickets they have to take care of
- Model Manager: grants access to the data to the other components, abstracting any kind of technology specific detail to the other services
- Car Manager: manages the physical cars providing other services with the needed functionalities and informations
- Power Station Manager: manages the physical power stations' data and functionalities
- Assistance Ticket Manager: grants the assistance callcenter, the operators and the other services the APIs to manage the assistance tickets
- Reservation Manager: manages the whole reservation process for an Active User from the reservation intention to the effective payment, interacting with the external payment handler through his APIs

All the communications between the different component will be asynchronous in order to minimise unneeded resource consumption that would result in an increase of the costs for commissioner.
Each of these services can be instantiated and deallocated as many times as needed to handle the momentary load of the system, taking advantage of the queuing system offered by the message broker. This, coupled with a dynamic cloud infrastructure that allows quasi-immediate upscale of computing and networking capability, will allow the system to handle load spikes along with mitigating the risk of a downtime due to the excessive load.

2.3 Deployment View

To reach our goals we have depicted 4 main components to be deployed separately from the numerous services we previously indicated in section 2.2:

- Onboard Car Management system: grants access remotely to the needed data both from the car to the server and vice versa and will grant the system to access functionalities such as locking and unlocking the car remotely.

- PowerStation Data System: deployed inside the single power stations will manage the communication with the central system granting access to it's data to the services that need them.
- Mobile App: will give both the operators and the Active Users, only once they signed in, access to the specific feature they have permission to access, could be implemented in many ways and will comunicate with the client handler server side through RESTful APIs. It will use an external map provider to give users a human friendly visualization of cars' positions.
- Web Page: will give the Visitors access to the map with cars' positions and let the signed in Users modify their personal data

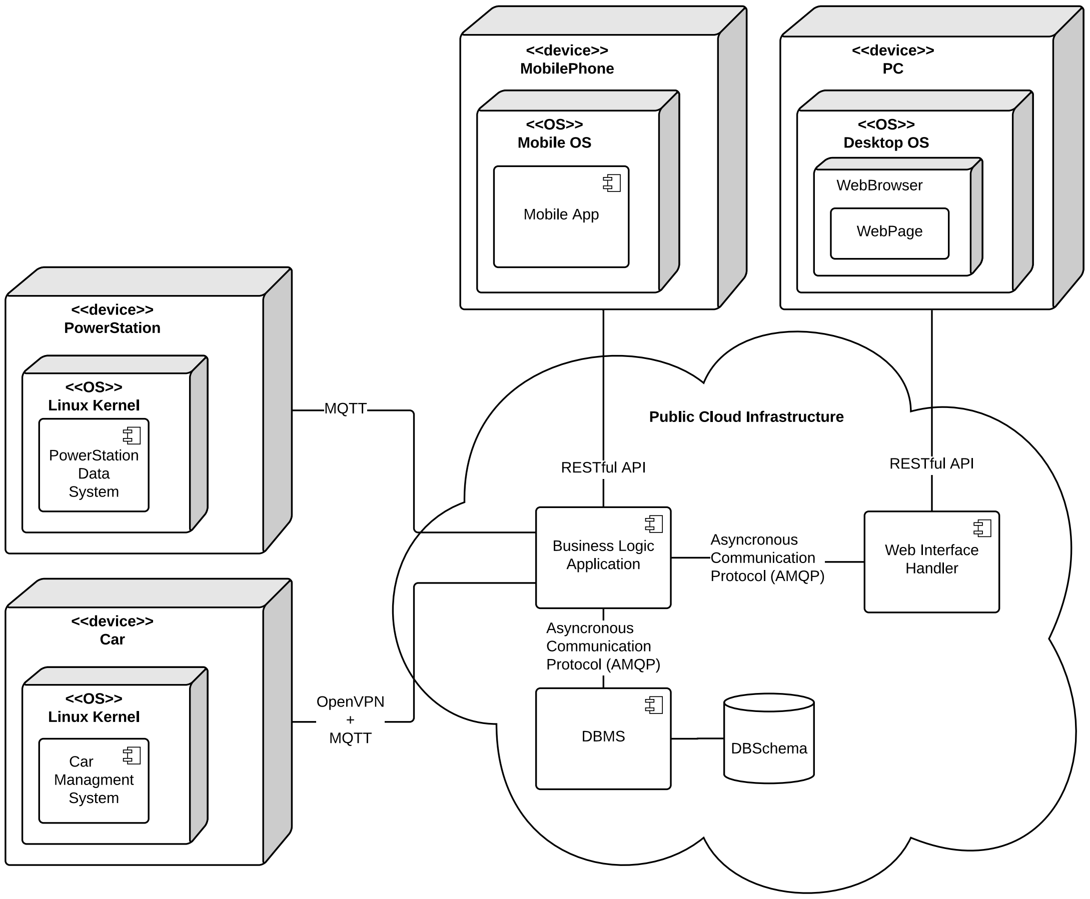


Figure 3: Deployment View of the system

2.4 Runtime View

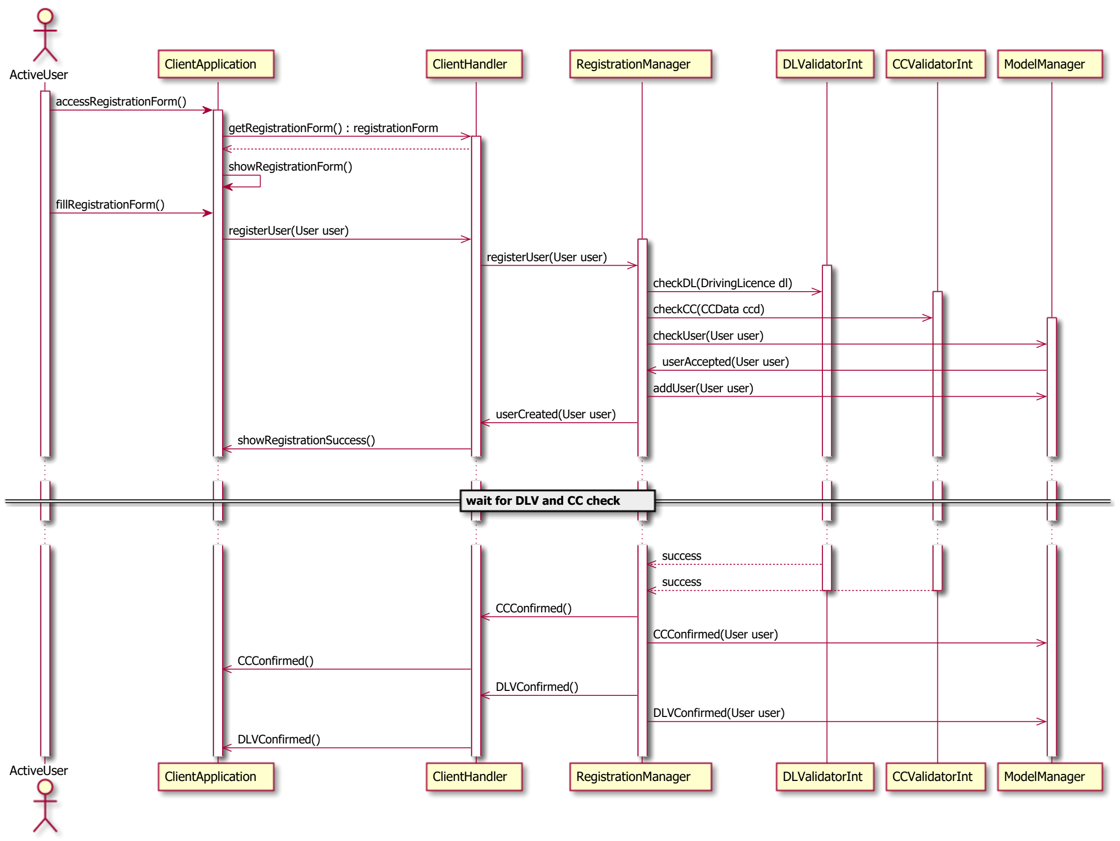


Figure 4: Sign Up sequence diagram

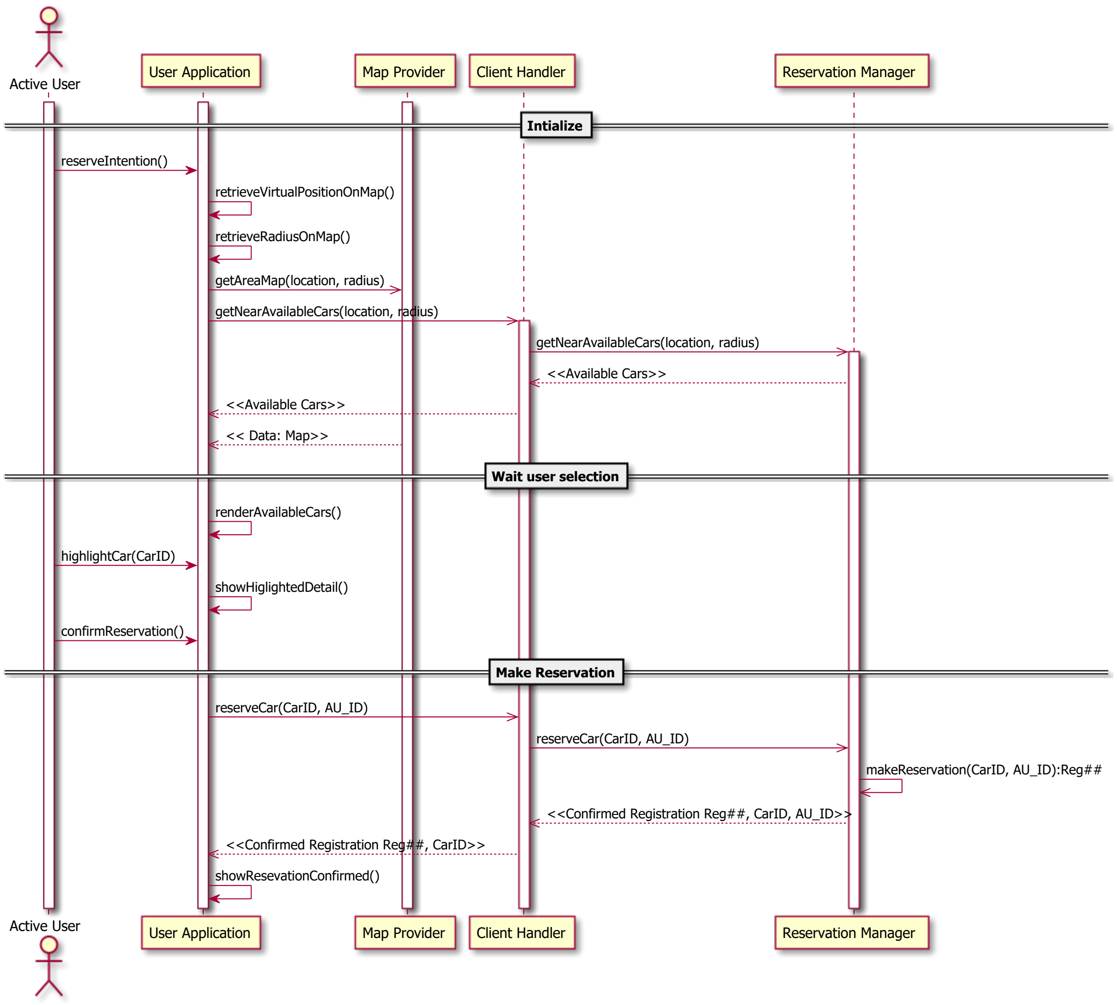


Figure 5: Car reservation sequence diagram

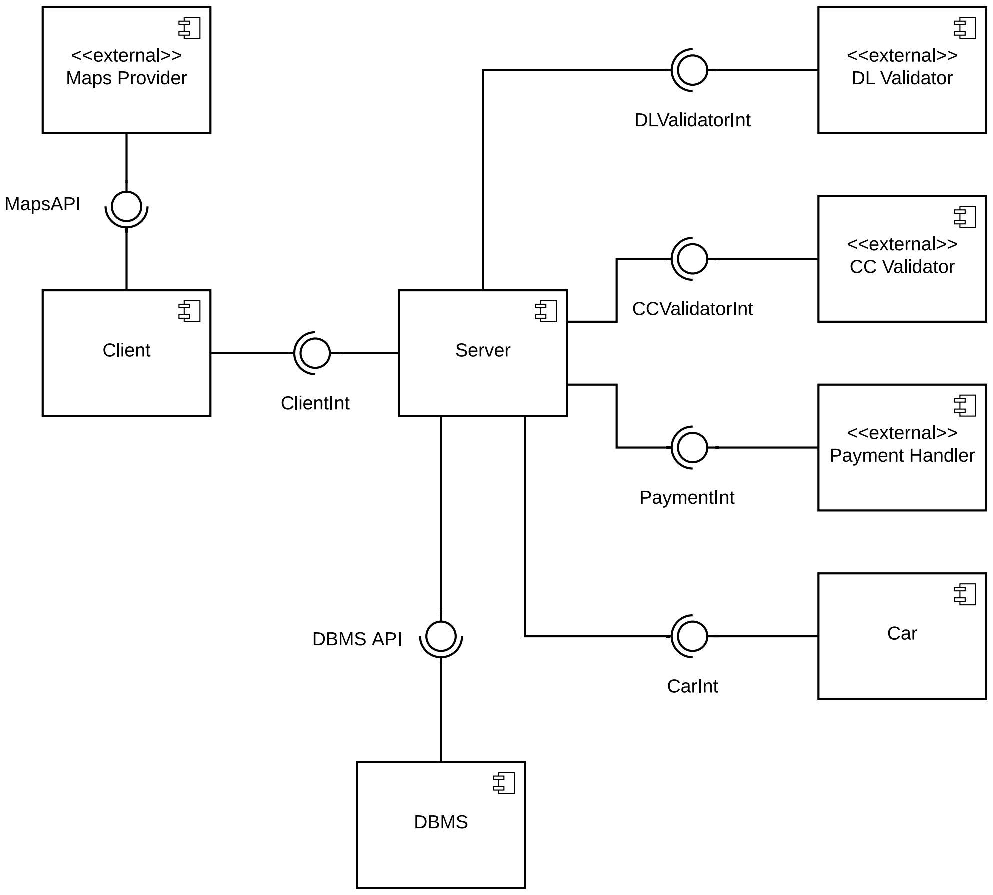


Figure 6: Component View of the server side deployed

2.5 Component Interfaces

2.6 Selected Architectural Styles And Patterns

2.7 Data management

Although we have used a microservice approach for the business logic tier, it doesn't make much sense split data between the different modules. In fact our data model is very small and interconnected and we don't need to use different approaches (eg. SQL and noSQL at the same time). Of course in this way we couple the different modules but it is inevitable since they run for the same macro-functionality. A unique database grant us the possibility to achieve in a simple way ACID properties and to use the standardized SQL language for queries. We provide an Entity-Relationship model for our application in figure 7.

2.8 Other Design Decisions

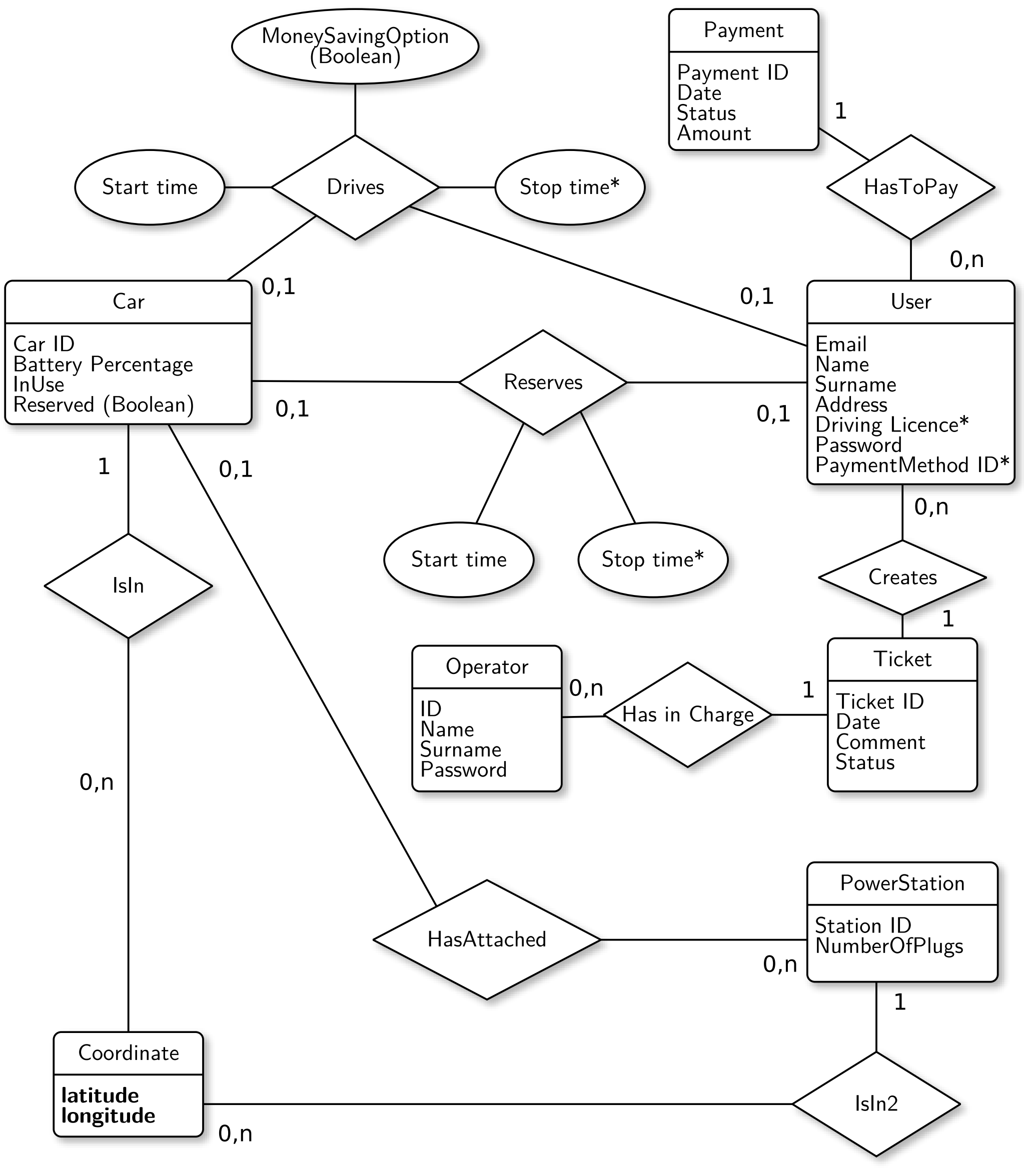


Figure 7: ER data model

3 Algorithm Design

All algorithms needed in the project are trivial but the one dealing with uniform repartition of cars in the city.
Cars are picked up by by users in one location dropped off in another one. Of course the distribution of of picking up and dropping off locations is not uniform. So some operators are needed in order to perform some relocations, in particular during the night, so that in the morning cars are located where there is actual necessity. The money saving option tries to drop the load of relocations, stimulating users to park where there is a deficiency, with a discount. So the global situation have to be monitored in real time, taking into account, the static situation, the reservations and and the current rides.
This problem has been studied a lot and there are in literature various algorithms that solve it. They are mainly based on mixed integer linear programming techniques and in particular [1] presented a complete model. In [2] is presented a greedy algorithm that achieves almost the same result. In [3] a more sofisticated approach is used taking into account a three dimensional objective function and exploiting genetic algorithms and local search methods. [4] offers a sort of classification of the strategies proposed in the past years.
For our purpose the approach described in [2] is the best since it minimizes the number of the operators needed to relocate cars and so the costs.

Riferimenti

[1] A. G. Kek, R. L. Cheu, Q. Meng, and C. H. Fung, “A decision support system for vehicle relocation operations in carsharing systems”, Transportation Research Part E: Logistics and Transportation Review, vol. 45, no. 1, pp. 149–158, 2009.
[2] R. Zakaria, L. Moalic, A. Caminada, M. Dib, “A Greedy Algorithm for relocation problem in one-way carsharing”, 10th International Conference on Modeling, Optimization and Simulation - MOSIM’14 – November 5-7-2014- Nancy – France “Toward circular Economy”.
[3] Moalic, L., Lamrous, S., & Caminada, A. (2013). A Multiobjective Memetic Algorithm for Solving the Carsharing Problem. Proceedings Of The 2013 International Conference On Artificial IntelligenceIcai 2013, Vol. 1, pp. 877-883.
[4] S. Weikl, K. Bogenberger, “Relocation Strategies and Algorithms for free-floating Car Sharing Systems”, 15th International IEEE Conference on Intelligent Transportation Systems Anchorage, Alaska, USA, September 16-19, 2012.

4 User Interface Design

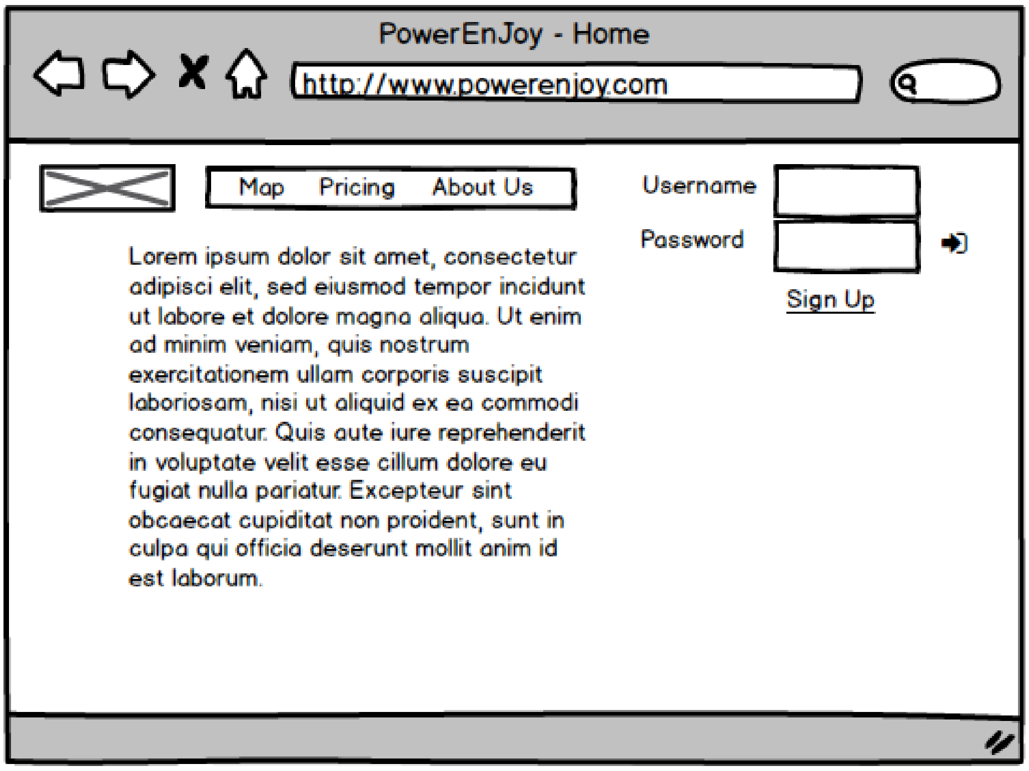
Below are some mockups to show how users will interact with the service. Since PowerEnJoy can be used both from a computer (except for unlocking the car), both Mobile and Web mockups are provided. Moreover, since both Users and Operators have access to the service via browser and app, interfaces for both types of users have been created.

4.1 User Interfaces

As anticipated, in this section all User mockups are analyzed. These mockups show how all actions that can be performed by our users. This section is further split between Web interfaces, imagined for standard browsers, and Mobile interfaces, designed having a smartphone App in mind.

4.1.1 Web Interfaces

4.1.1.1 Home Page (Web) From the home page any user can try to login inserting username and password or they can choose to sign up and go to the registration page. This page will likely show a description of the service as well as providing links to other important part of the website (Map, Pricing, About Us).



Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Figure 8: Web Interface - User's Home Page

4.1.1.2 Registration Page (Web) In this page users must input the core informations to register online: name, surname and email address.

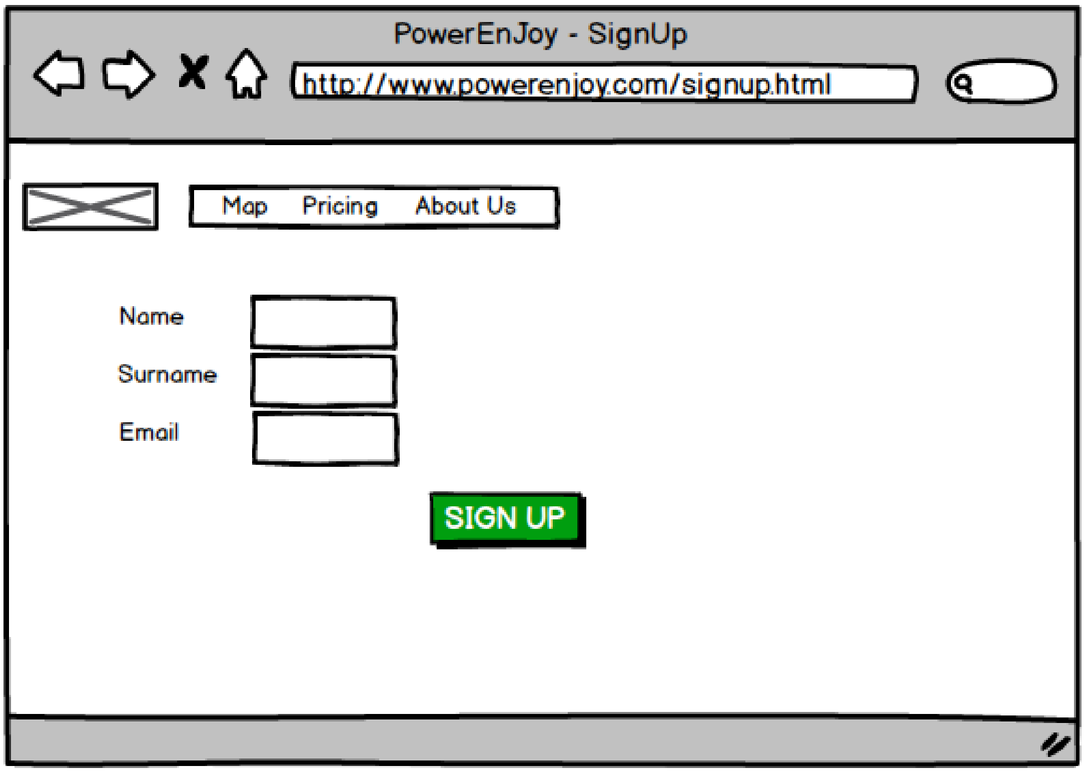
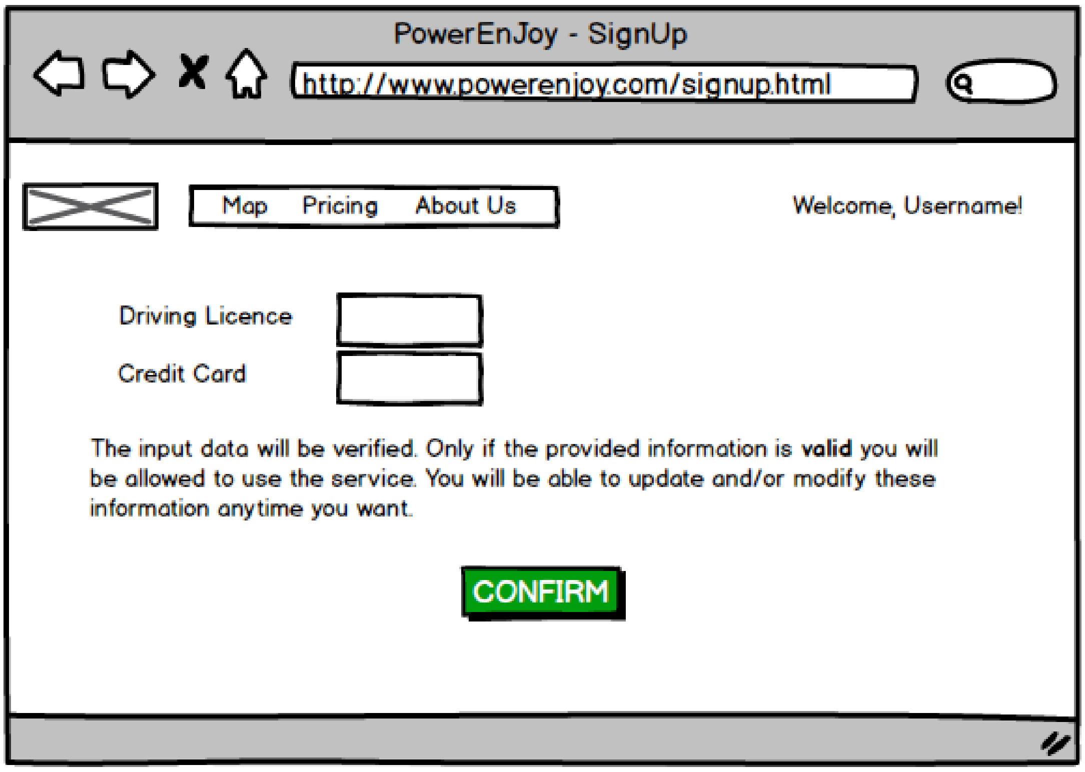


Figure 9: Web Interface - User Registration

4.1.1.3 Further Information (Web) After having registered and logged in, users must input their Licence and Credit Card in order to use the service if they haven't already.



The input data will be verified. Only if the provided information is **valid** you will be allowed to use the service. You will be able to update and/or modify these information anytime you want.

Figure 10: Web Interface - User Further Information

4.1.1.4 Map (Web) All users can view available cars near their position and choose one if they want more info (see next mockup).

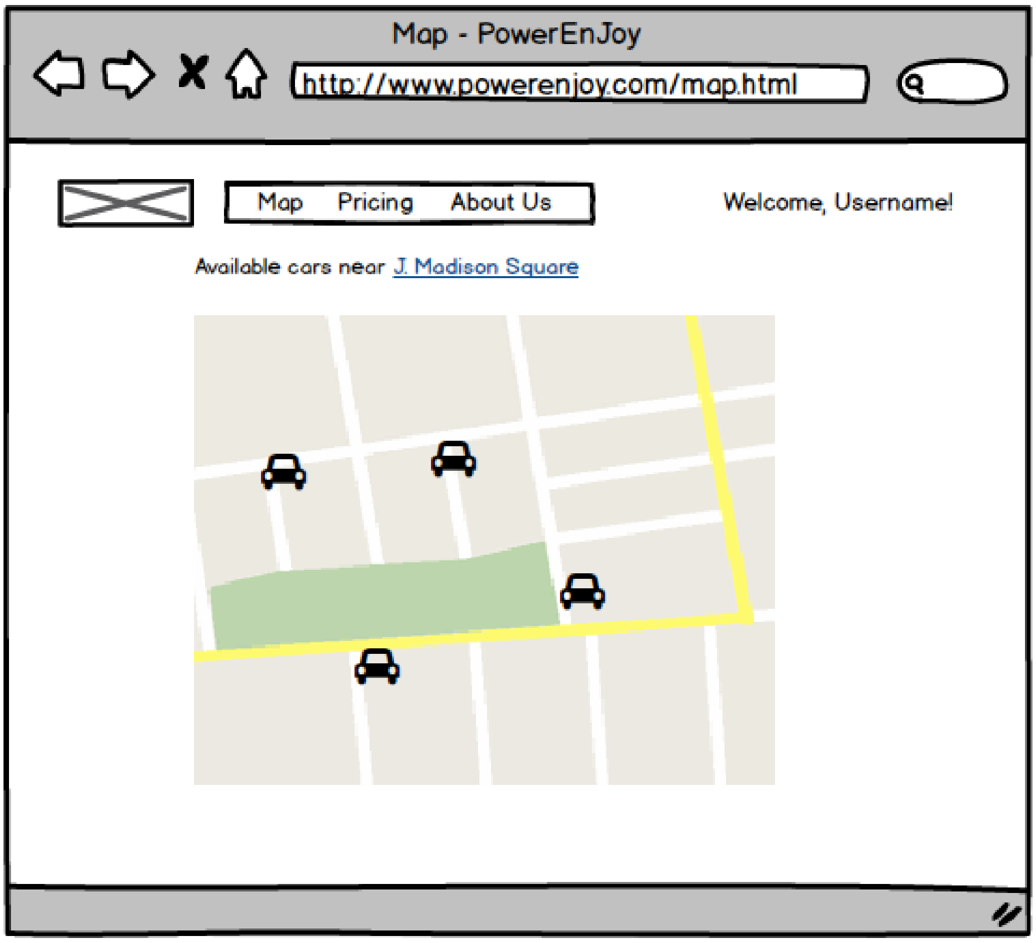

Map - PowerEnJoy
http://www.powerenjoy.com/map.html
Map
Pricing
About Us
Welcome, Username!
Available cars near J. Madison Square

Figure 11: Web Interface - Map

4.1.1.5 Map - Selected Car (Web) Selecting a car provides relevant information about that specific car, in order to give to the user the chance to pick a car that can suit his needs.

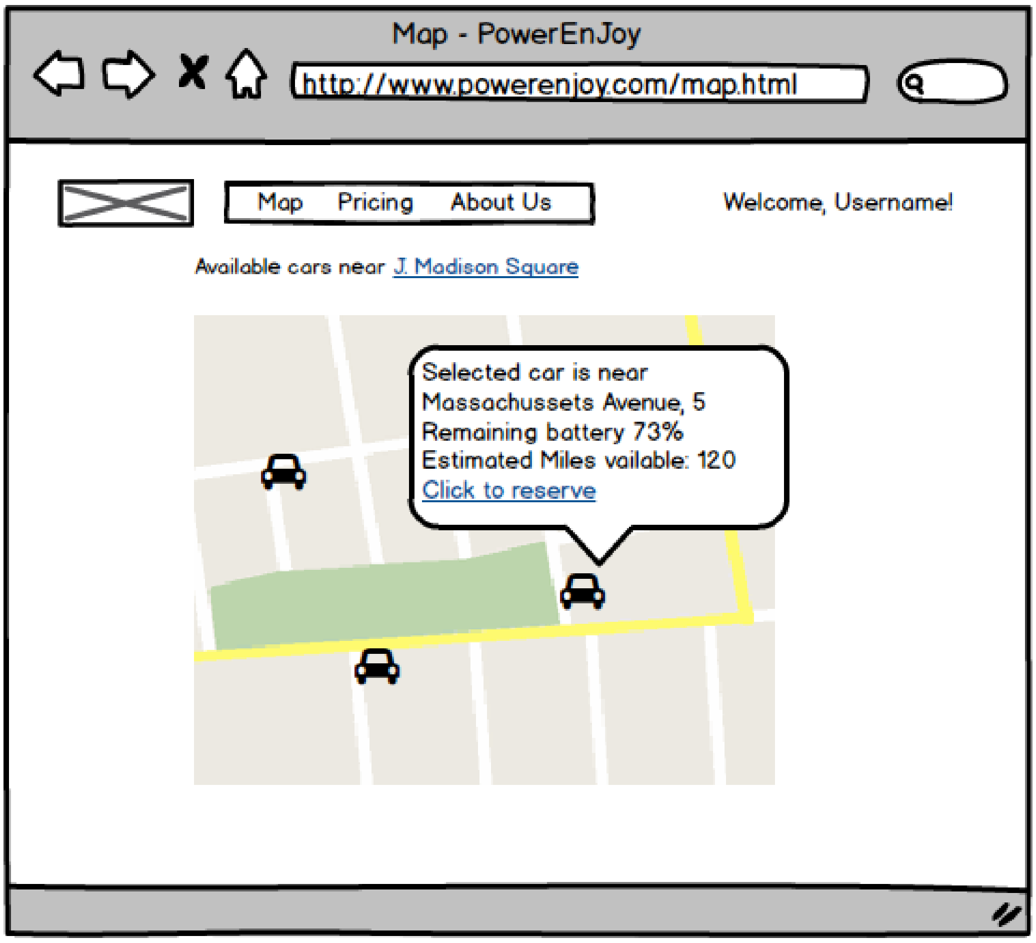
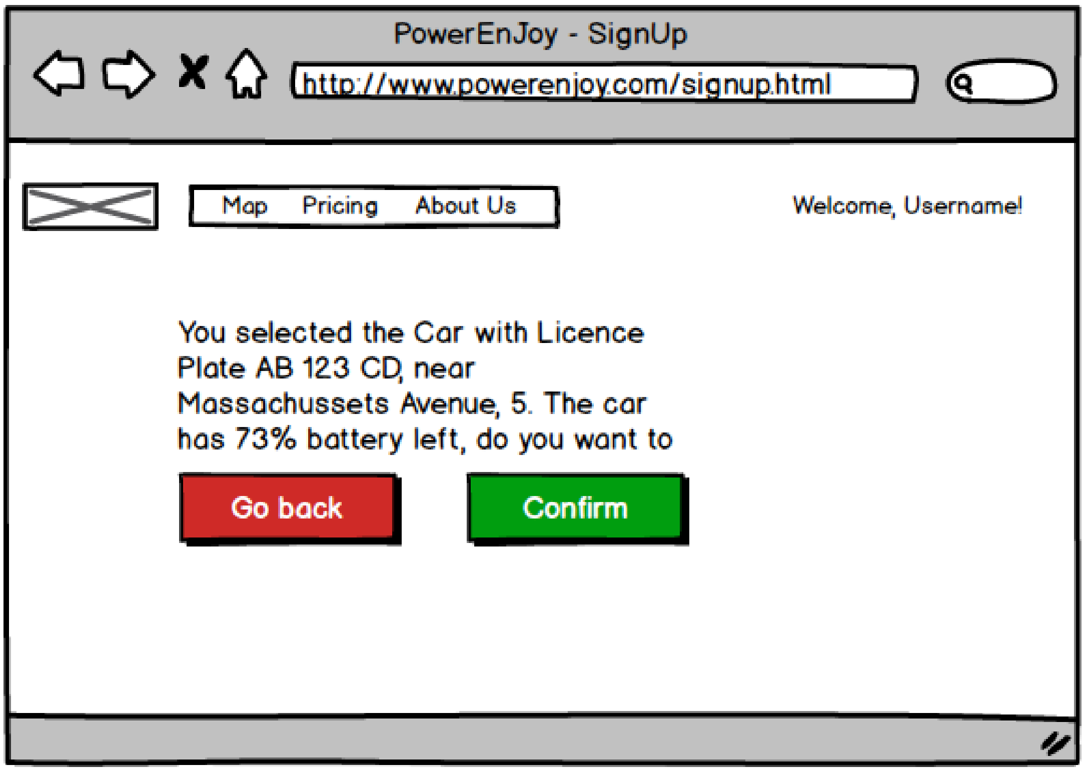


Figure 12: Web Interface - Selected car

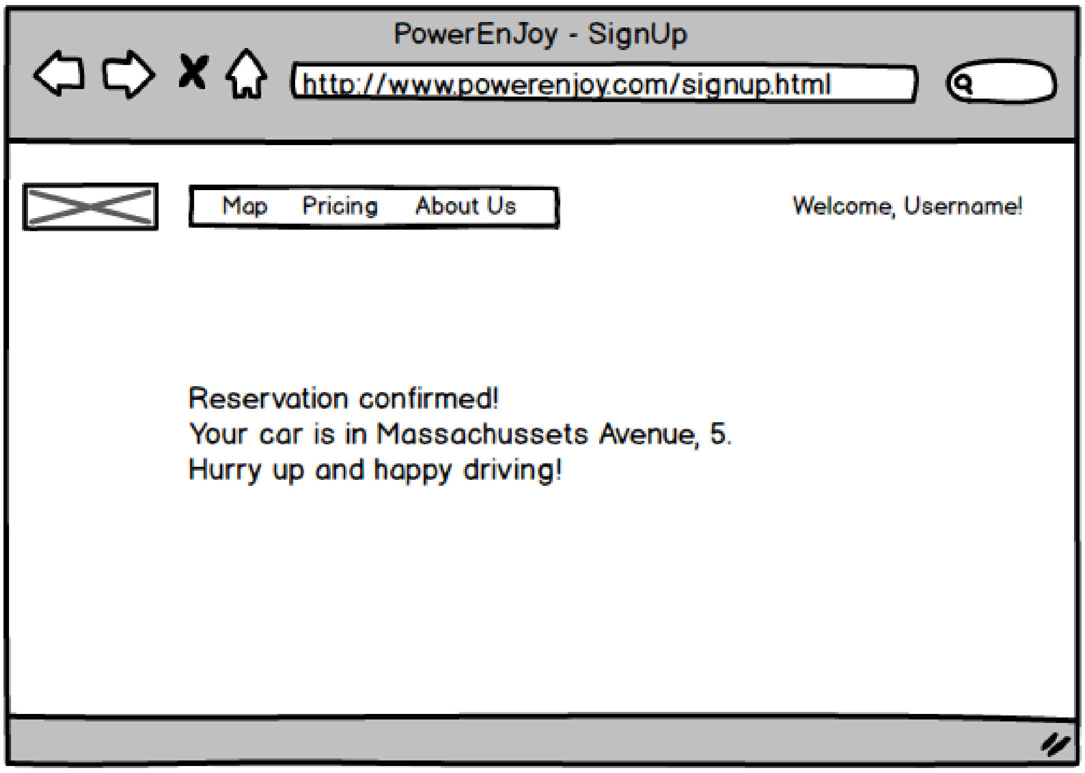
4.1.1.6 Selected Car Confirmation (Web) After having selected a car and having pressed “Click to reserve”, users are asked to confirm their choice one last time, to avoid errors in case of mistakenly pressed links or misread information.



You selected the Car with Licence Plate AB 123 CD, near Massachussets Avenue, 5. The car has 73% battery left, do you want to

Figure 13: Web Interface - Car Confirmation Screen

4.1.1.7 Reservation Confirmed (Web) Confirming a reservation shows a brief summary containing the address at which the car is parked.



Reservation confirmed!
Your car is in Massachussets Avenue, 5.
Hurry up and happy driving!

Figure 14: Web Interface - Reservation Confirmed

4.1.2 Mobile Interfaces

4.1.2.1 Home (Mobile) From the App's home page users can either login or create an account. Login is handled inside the app, while to create an account the user is redirected to the website.

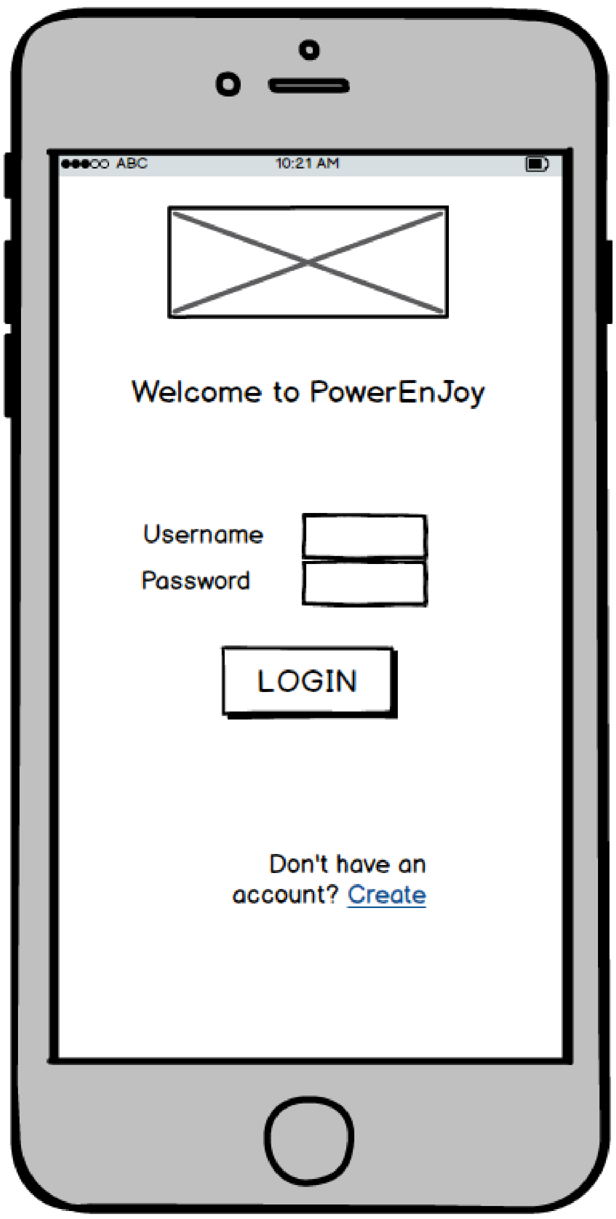


Figure 15: Mobile Interface - Home Page

4.1.2.2 Map (Mobile) The user is shown a map displaying all cars that are nearby. His position can be calculated using the built in GPS receiver or can be manually input.

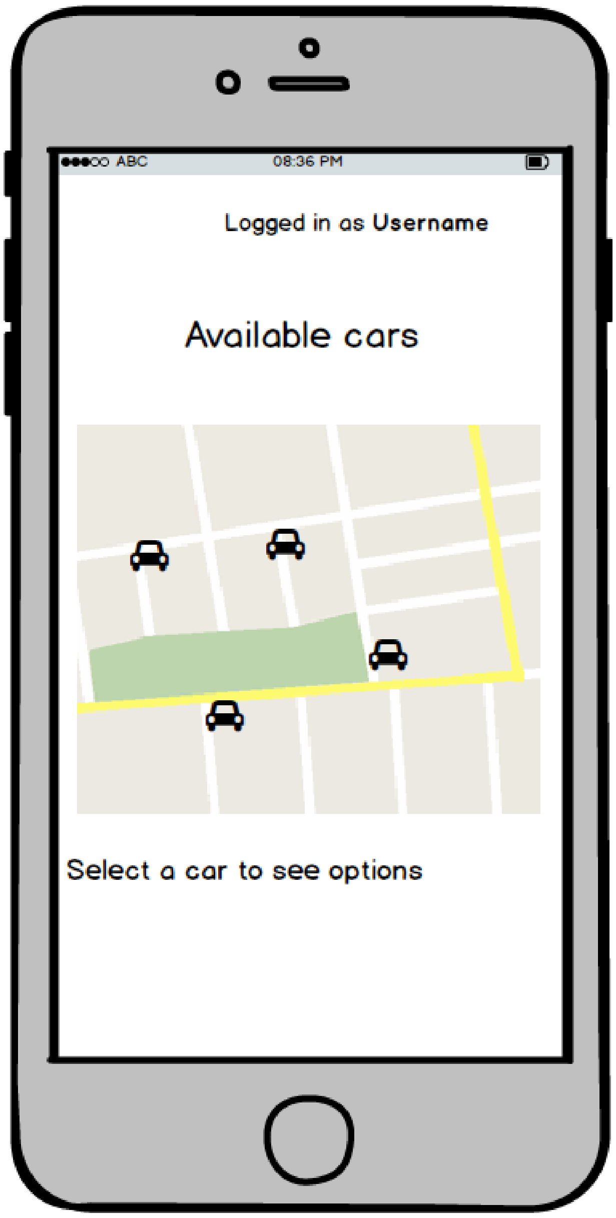


Figure 16: Mobile Interface - Map

4.1.2.3 Selected Car (Mobile) Selecting a car on the map shows relevant information, like it happens on the website.

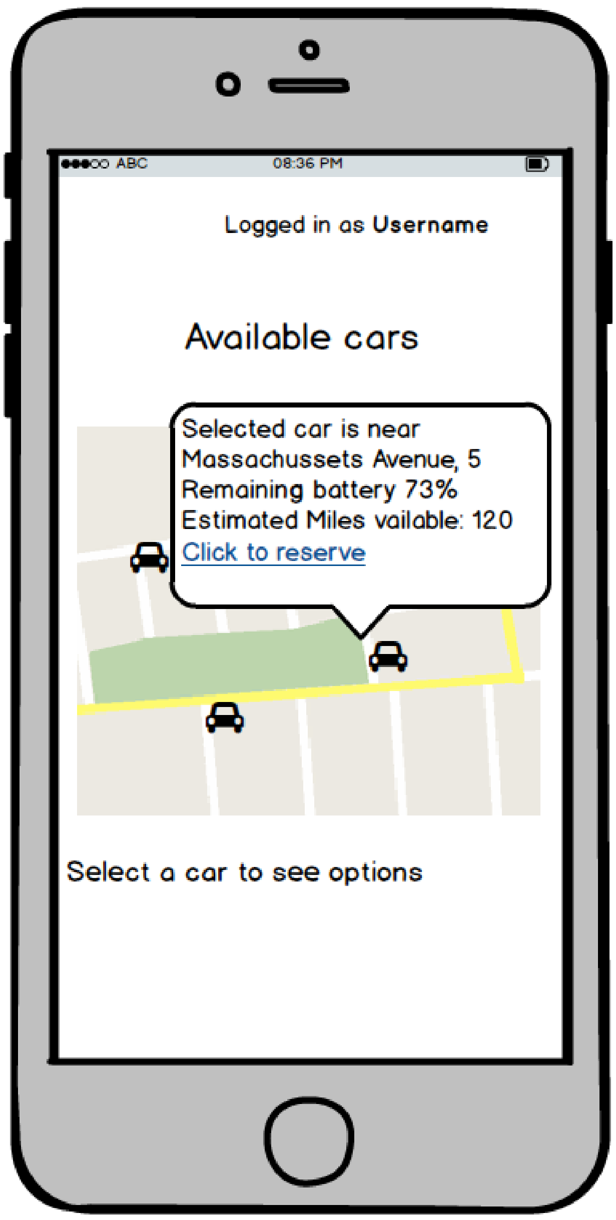
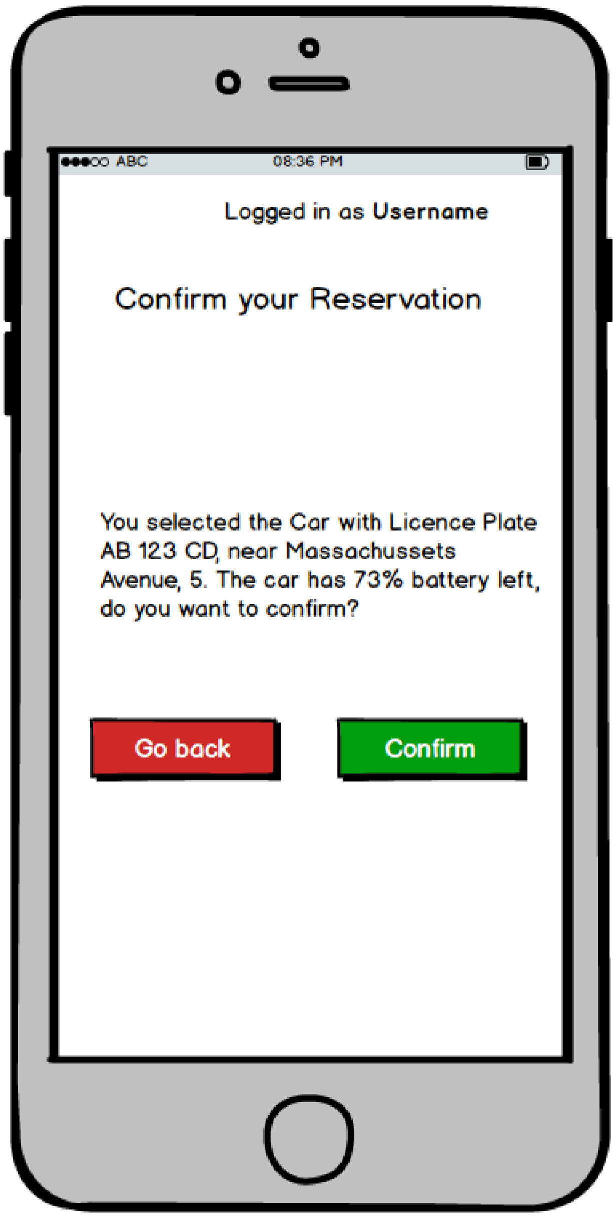

●●●○○ ABC
08:36 PM
Logged in as **Username**
Available cars
Selected car is near Massachussets Avenue, 5
Remaining battery 73%
Estimated Miles vailable: 120
Click to reserve
Select a car to see options

Figure 17: Mobile Interface - Selected Car

4.1.2.4 Car Confirmation (Mobile) A user can confirm a reservation or go back if he chose that car by mistake.



You selected the Car with Licence Plate AB 123 CD, near Massachussets Avenue, 5. The car has 73% battery left, do you want to confirm?

Figure 18: Mobile Interface - Car Confirmation

4.1.2.5 Reservation Confirmed (Mobile) When the reservation is confirmed, the smartphone shows a countdown as well as the car's address.

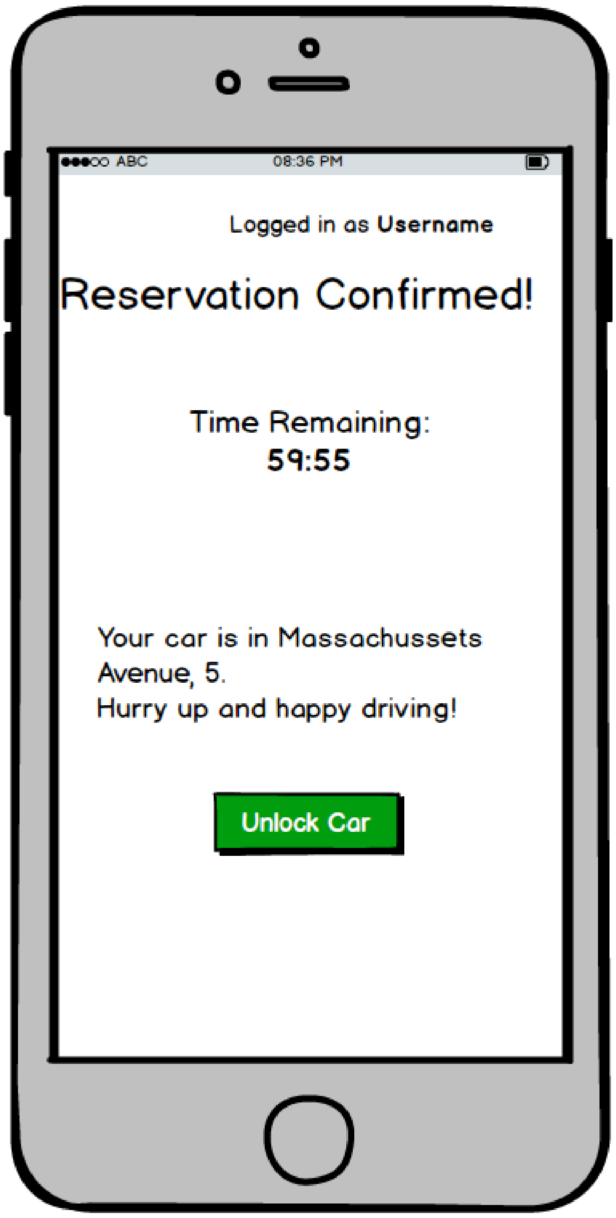
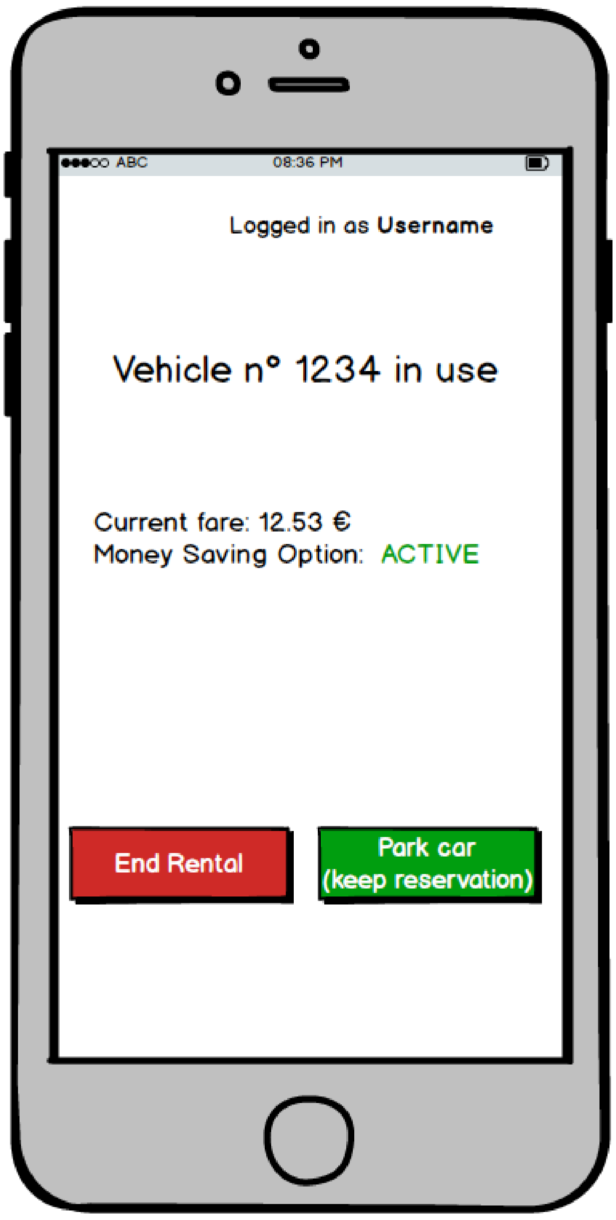

●●●○○ ABC
08:36 PM
Logged in as **Username**
Reservation Confirmed!
Time Remaining:
59:55
Your car is in Massachussets Avenue, 5.
Hurry up and happy driving!
Unlock Car

Figure 19: Mobile Interface - Reservation Confirmed

4.1.2.6 Car In Use (Mobile) While using the car, Users can decide to park it — which signals the system that the car will be picked up by the same user and that the rental is not to be terminated — or to end the rental.

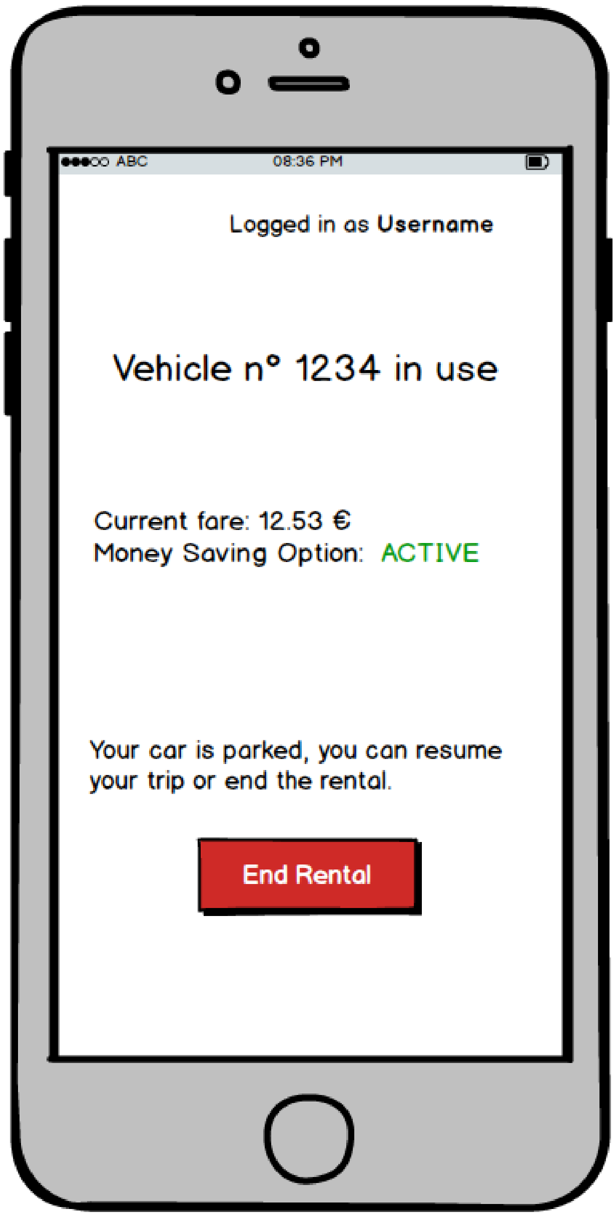


Vehicle n° 1234 in use

Current fare: 12.53 €
Money Saving Option: ACTIVE

Figure 20: Mobile Interface - Car in Use

4.1.2.7 Car Parked (Mobile) If the car is already parked users can decide to end the rental using the app. In case they want to continue their journey, they only have to jump back on the car.



Vehicle n° 1234 in use

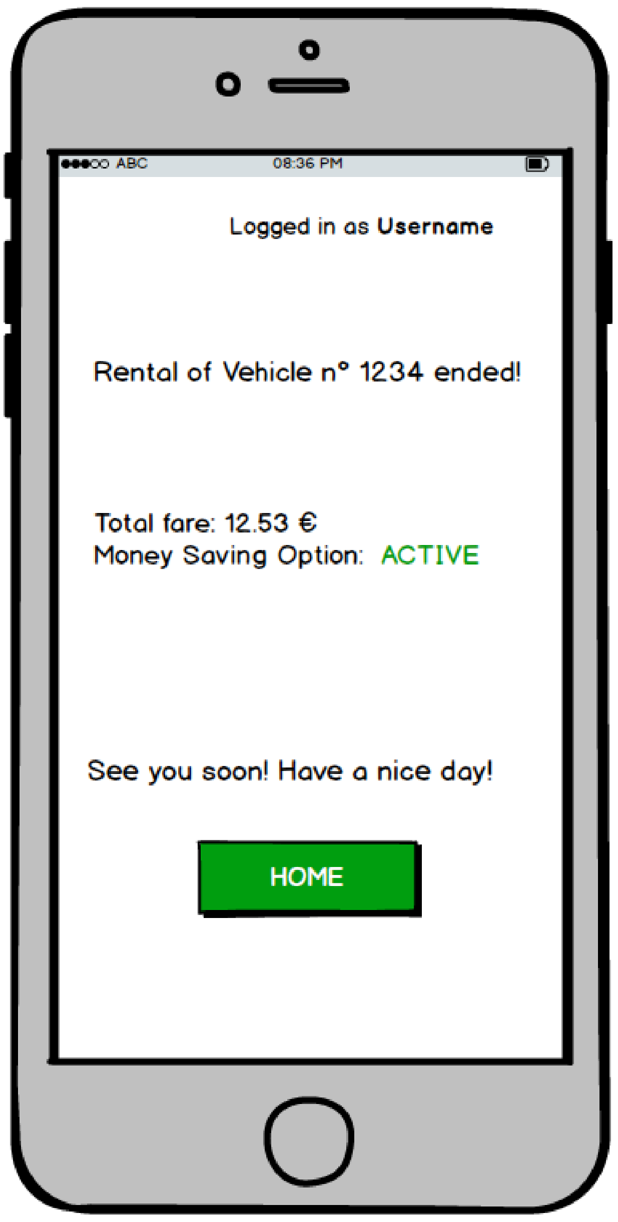
Current fare: 12.53 €
Money Saving Option: ACTIVE

Your car is parked, you can resume your trip or end the rental.

Figure 21: Mobile Interface - Car Parked

4.1.2.8 Rental Ended (Mobile)

Ending a rental shows the total as well as whether the Money Saving Option was active for the trip.



Rental of Vehicle n° 1234 ended!

Total fare: 12.53 €
Money Saving Option: ACTIVE

See you soon! Have a nice day!

Figure 22: Mobile Interface - Rental Ended

4.2 Operator Interfaces

4.2.1 Web Interfaces

4.2.1.1 Operator Main Page (Web) Operators can choose between performing a pending task (a "Todo") or managing a specific car.

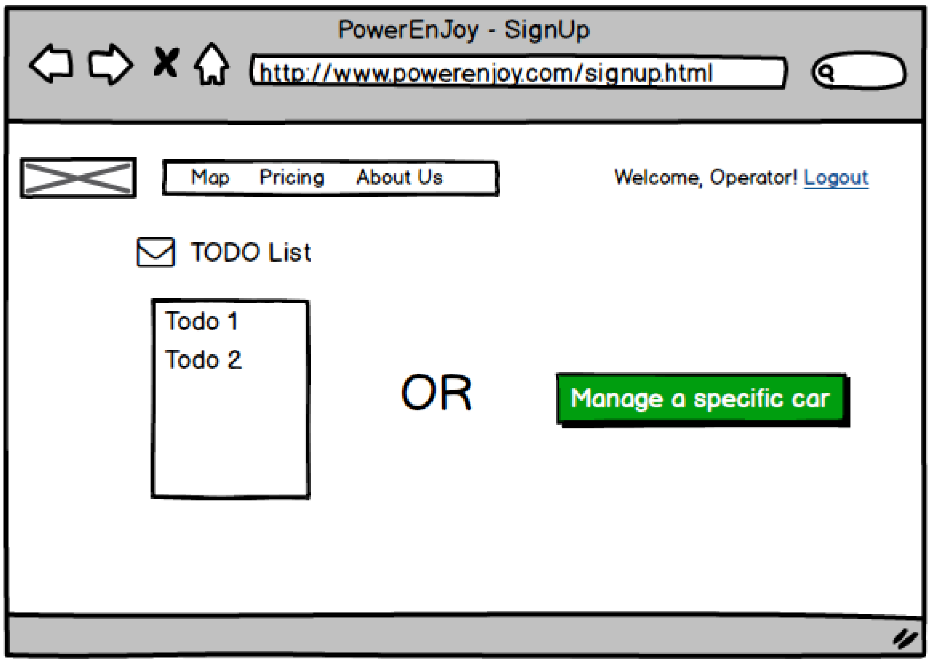
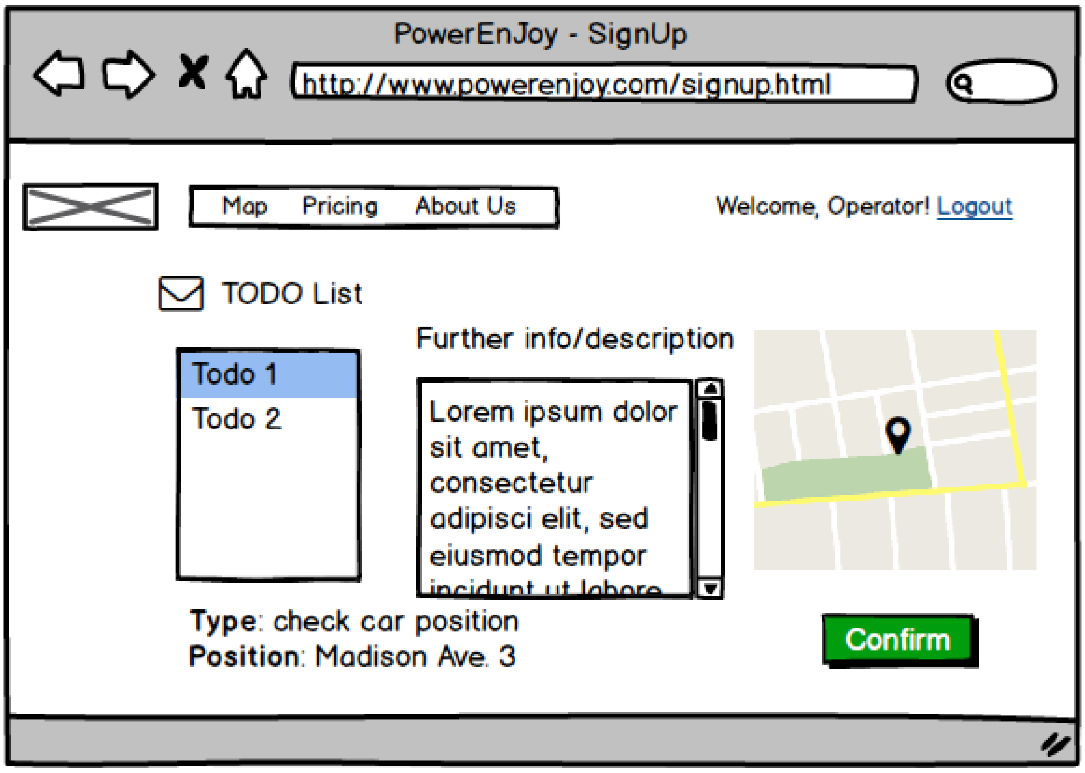


Figure 23: Web Interface - Operator Main Page

4.2.1.2 Operator chose TODO (Web) Choosing a pending operation brings up relevant information about it: its type, the position (if relevant), a short summary and a map. The operator can confirm the task if he will take care of it or go back to the home page(the logo links to the home page).

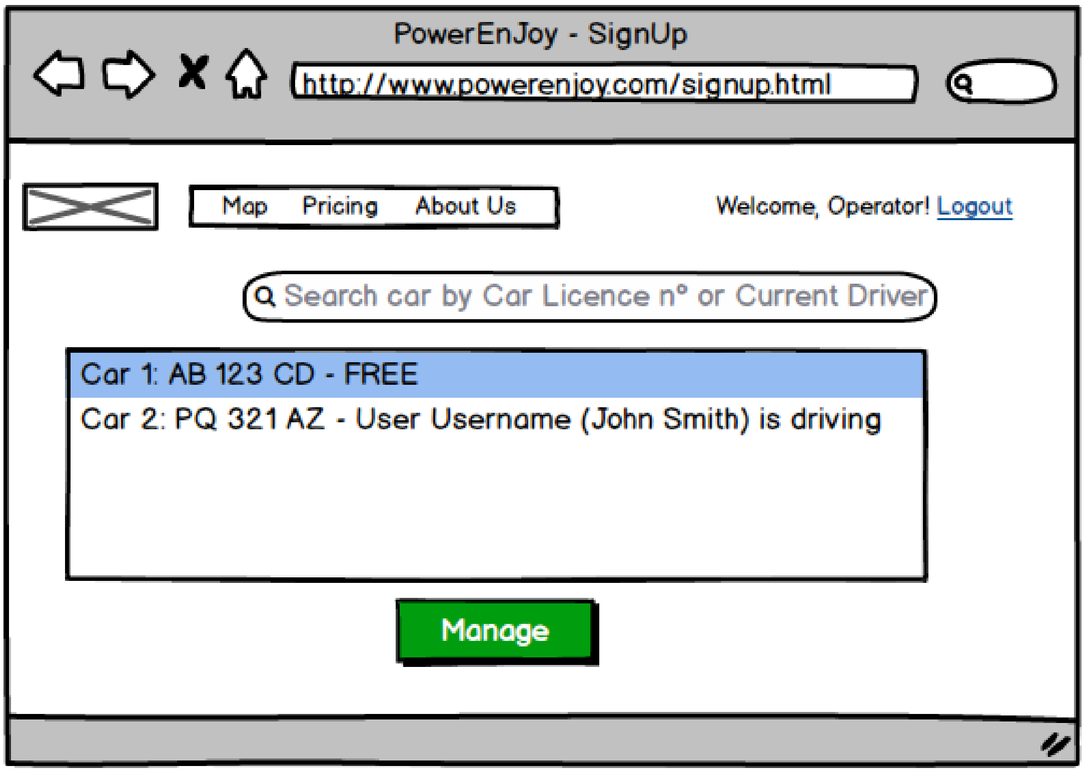


Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore

Type: check car position
Position: Madison Ave. 3

Figure 24: Web Interface - Chosen TODO

4.2.1.3 Operator Searched Car (Web) Choosing to manage a specific car allows the operator to search among all cars, by car Licence Plate or by Current Driver (User) if in use.


PowerEnJoy - SignUp
http://www.powerenjoy.com/signup.html
Map
Pricing
About Us
Welcome, Operator! Logout
Search car by Car Licence n° or Current Driver
Manage

Car 1: AB 123 CD - FREE
Car 2: PQ 321 AZ - User Username (John Smith) is driving

Figure 25: Web Interface - Car Search

4.2.1.4 Car Details (Web) The Car Details page shows all informations available for the chosen car as well as providing buttons to change all editable parameters (for instance the car's status)

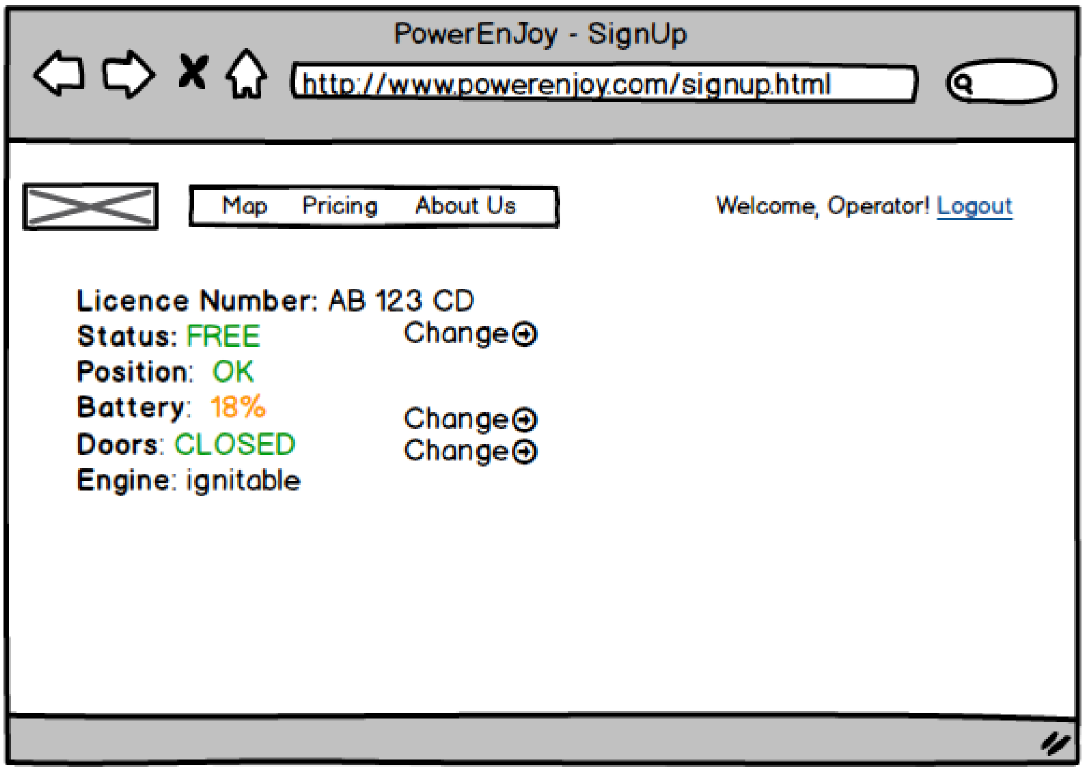


Figure 26: Web Interface - Car Details

4.2.1.5 Changing a Car parameter - Sample (Web)

Choosing to edit a parameter brings up a "pop-up" providing the needed options to edit.

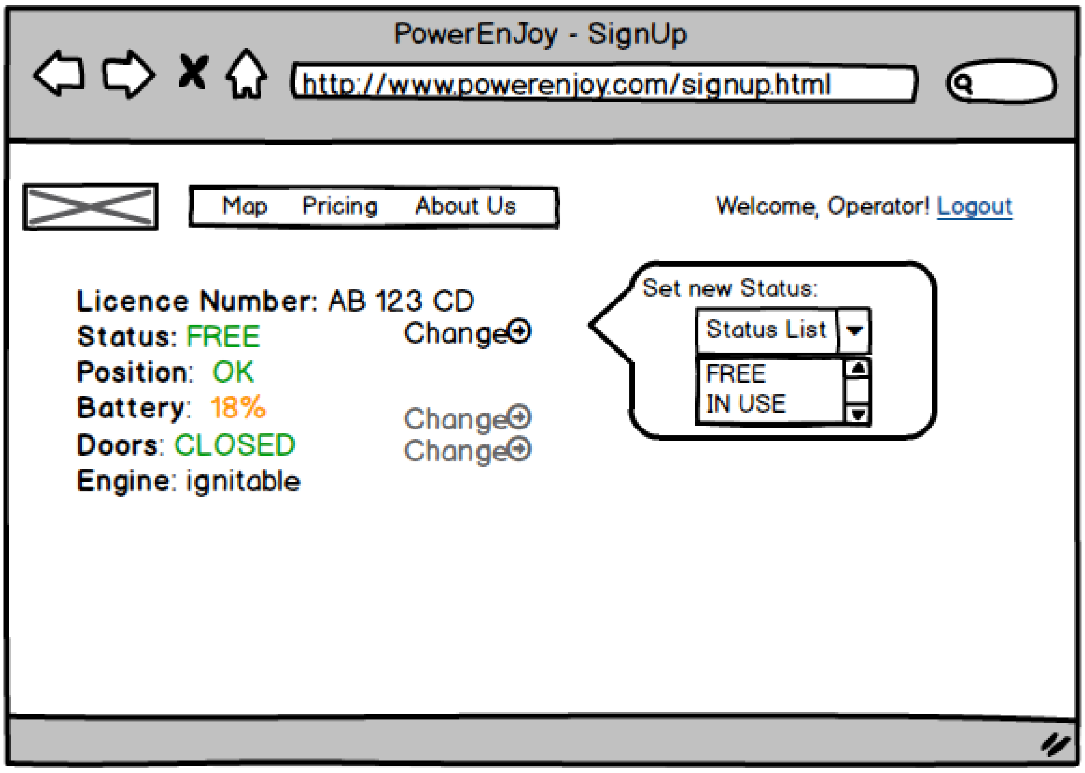


Figure 27: Web Interface - Changing a detail

4.2.2 Mobile Interfaces

4.2.2.1 Main Page (Mobile) The mobile Main page for operators offers the same options of the web one: they caan either choose a pensing task or opt to manage a car.

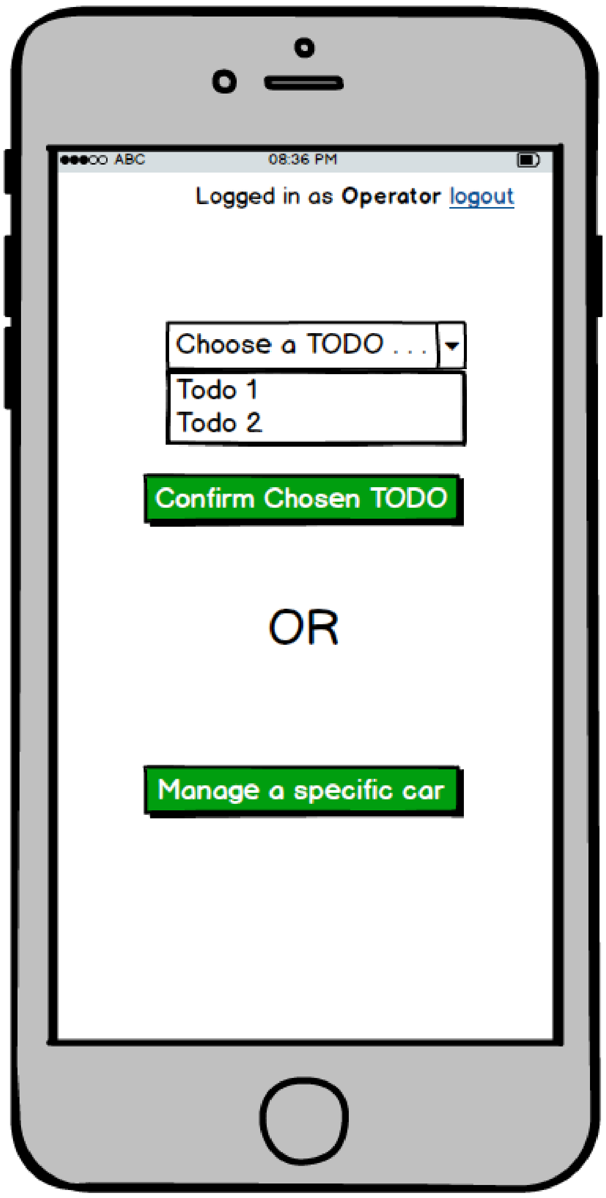
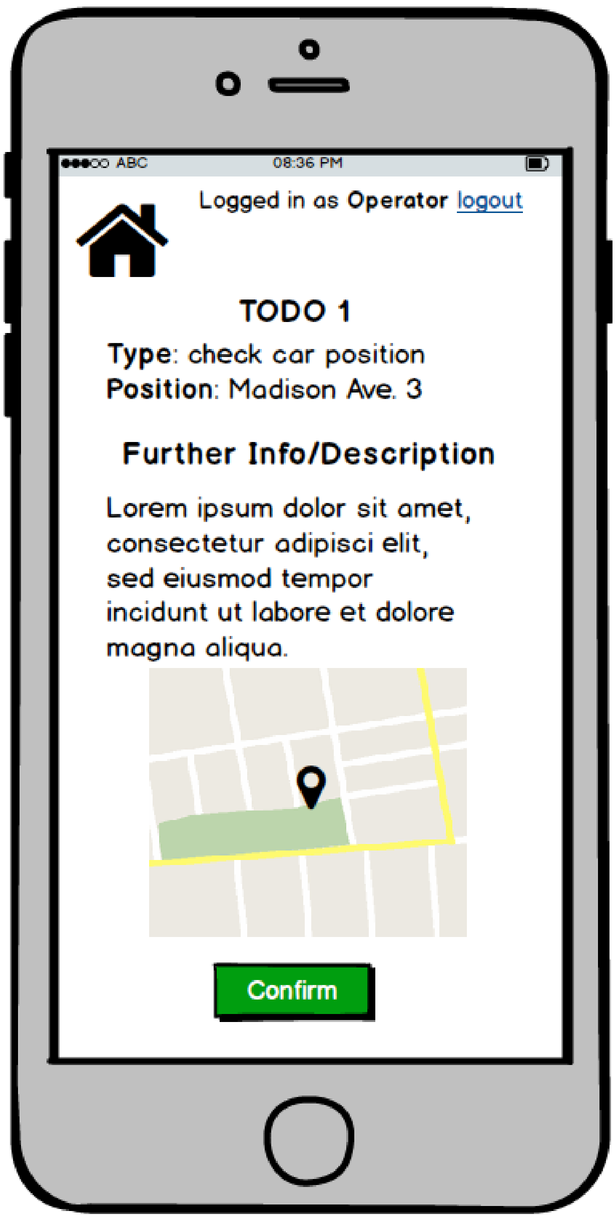


Figure 28: Mobile Interface - Operator Main Page



TODO 1

Type: check car position
Position: Madison Ave. 3

Further Info/Description

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua.

Figure 29: Mobile Interface - Chosen Todo

4.2.2.2 Operator chose TODO (Mobile)

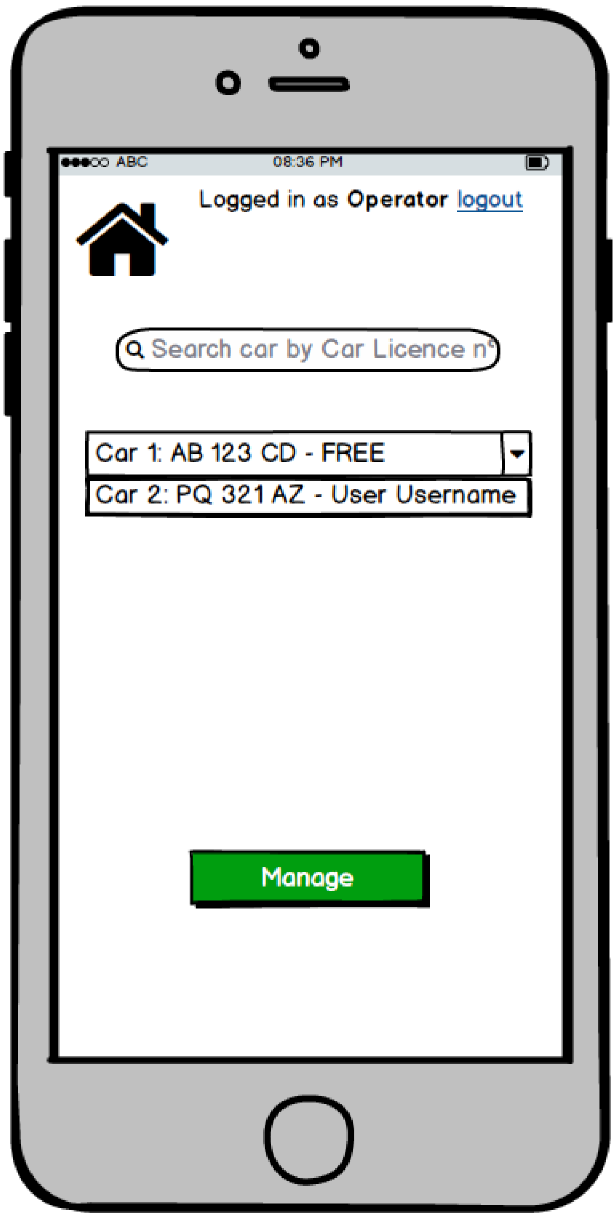

●●●○○ ABC
08:36 PM
Logged in as **Operator** logout
Search car by Car Licence n°
Car 1: AB 123 CD - FREE
Car 2: PQ 321 AZ - User Username
Manage

Figure 30: Mobile Interface - Car Search

4.2.2.3 Operator Searched Car (Mobile)

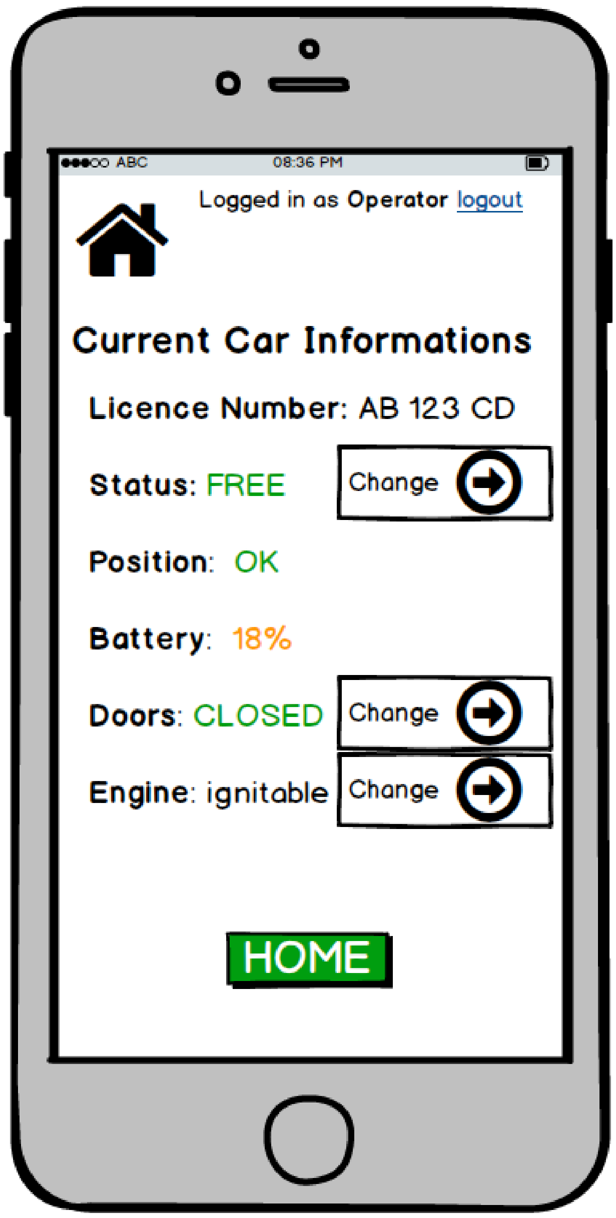


Figure 31: Mobile Interface - Car Details

4.2.2.4 Operator Car Details (Mobile)

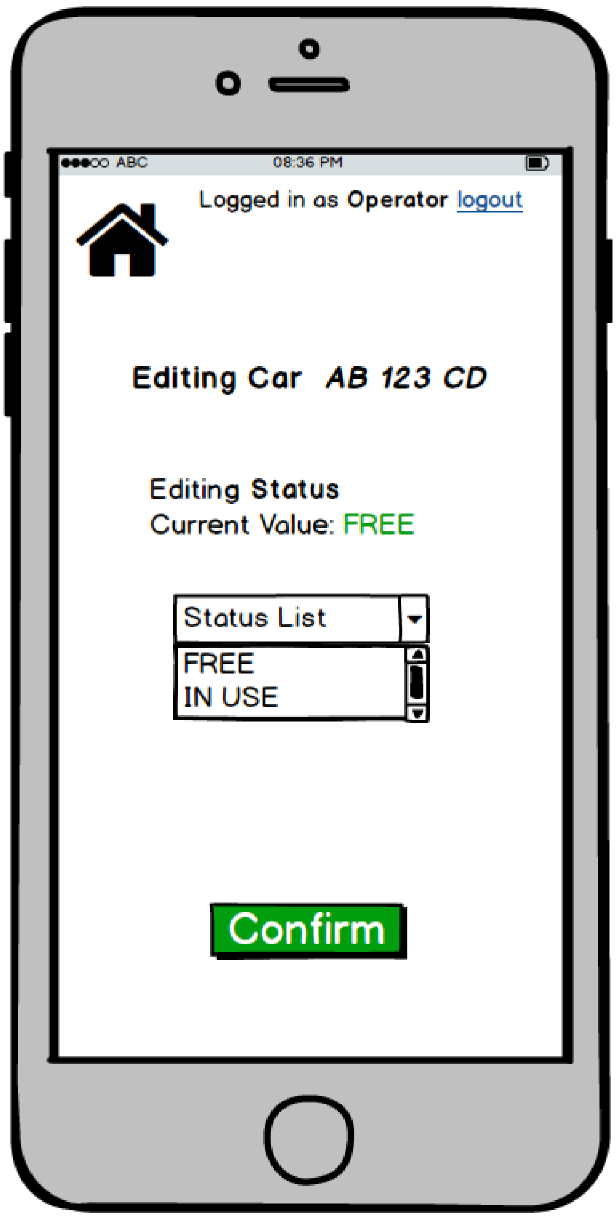


Figure 32: Mobile Interface - Changing a Car's parameter

4.2.2.5 Changing a Car parameter - Sample (Mobile)

4.3 Car Interface

4.3.1 Car Screen

The car has a screen that shows the fare in real time as well as showing whether the Money Saving Option is active or not.

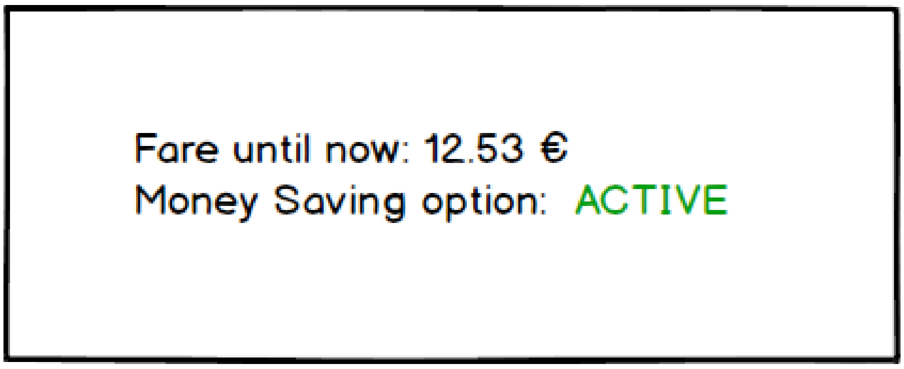

Fare until now: 12.53 €
Money Saving option: ACTIVE

Figure 33: Car Screen

5 Requirements traceability

6 Effort spent

Component	**Time spent (in hour)**
Philippe Scorsolini	23
Lorenzo Semeria	18
Gabriele Vanoni	24.5