POLITECNICO
MILANO 1863

Politecnico di Milano
A.A. 2015/2016
Software Engineering 2: "*MyTaxyService*"

Alessandro Baldassari (mat. 841561)
Alberto Bendin (mat. 841734)
Francesco Giarola (mat. 840554)

October 26, 2015

Contents

Page

1 Introduction 1
- 1.1 Purpose 1
- 1.2 Scope 1
- 1.3 Identifying Stackeholders 2
- 1.4 Identifying Actors 2
- 1.5 Goals 2
- 1.6 Proposed system 4
- 1.7 Definitions, acronyms, and abbreviations 4
  - 1.7.1 Definitions 4
  - 1.7.2 Acronyms 5
  - 1.7.3 Abbreviations 6
- 1.8 Reference Documents 6
- 1.9 Overview 6

2 Overall description 7
- 2.1 Product perspective 7
- 2.2 Product functions 7
- 2.3 User characteristics 7
- 2.4 Constraints 7
  - 2.4.1 Regulatory Policies & Safety and security 7
  - 2.4.2 Hardware limitations 7
  - 2.4.3 Interfaces to other applications 7
  - 2.4.4 Parallel operation 8
- 2.5 Assumptions and dependencies 8

3 Specific requirements 9
- 3.1 External Interface Requirements 9
  - 3.1.1 User Interfaces 9
    - 3.1.1.1 Sign-up 10
    - 3.1.1.2 Login 11
    - 3.1.1.3 Call a taxi 12
    - 3.1.1.4 Plan a ride 13
    - 3.1.1.5 Your reservations 14
    - 3.1.1.6 Your profile 15
    - 3.1.1.7 Your work 16
  - 3.1.2 Hardware Interfaces 17

3.1.3 Software Interfaces 17
3.1.4 Communication Interfaces 17
3.2 Functional Requirements 17
3.2.1 User Class 1 17
3.2.2 User Class 2 17
3.3 Performance Requirements 17
3.4 Design Constraints 17
3.4.1 Standards compliance 17
3.4.2 Hardware limitations 17
3.5 Software System Attributes 17
3.5.1 Reliability 17
3.5.2 Availability 17
3.5.3 Security 17
3.5.4 Maintainability 17
3.5.5 Portability 17
3.6 Other Requirements 17

1 Introduction

1.1 Purpose

The purpose of this document is to describe in a complete and sound way the *MyTaxiService* application that will be developed and the application domain in which it will run.
The intended audience for this document are the developers and programmers who have to implement the application, system and requirement analysts who want to integrate *MyTaxiService* with their system or software, testers who have to determine whether the requirements have been satisfied in the application implementation, projects managers who have to plan, estimate and control the analysis and development processes and finally the users themselves. This document could be used as a contractual agreement between the costumer and the entity who develops the application.

1.2 Scope

MyTaxiService is a new web and mobile application conceived to provide an immediate and user-friendly access to the taxi service of a large city; it aims at an overall improvement of the quality of the service offered.
This optimization is obtained thanks to the real-time interaction and feedback of all the parties involved in the service: taxi passengers can choose and book the ride, and the system will forward the request to the nearest available taxi drivers who can decide to take over the call; in this case the system will notify the client with the code of the incoming taxi and the waiting-time. The system guarantees a fair management of taxi queues. In particular, the city is divided in taxi zones and each zone is associated with its taxi queue. The system automatically computes the distribution of taxis in the various zones based on the GPS information it receives from each taxi. When a taxi is available, its identifier is stored in the queue of taxis in the corresponding zone. When a request arrives from a certain zone, the system forwards it to the taxis in the corresponding zone according to their order in the queue. If the taxi confirms, then the system will send a confirmation to the passenger. If not, then the system will forward the request to the second in queue and will move the first taxi in the last position of the queue. Additional features of the application are the possibility for the passengers to reserve a ride with an advance of at least two hours, choosing the origin and destination, and the option to possibly share the ride with someone else, thus dividing the cost of the service. The system confirms the reservation to the user and allocates a taxi to the request 10 minutes before the meeting time with the user. If more people are willing to share a ride from the same zone going in the same direction, then the system arranges the route for the taxi driver and defines the fee for every passenger informing all the users involved.

1.3 Identifying Stackeholders

The main direct stakeholders for this project are the government of the city and the taxi company which together have promoted the renewal of the software that manages the taxi service in the city. Of course once the system will be up and running the final stakeholders will be the users of the service that will provide an essential feedback on the new system.

1.4 Identifying Actors

- Clients: are the final users the taxi service is offered to. They can book the ride choosing among different options, for instance date and time, origin and destination locations and the possibility of sharing the trip with other customers.
- Taxi Drivers: represent the other category of users of the application, they can accept a call for a service or turn it down, thus allowing the whole system to be synchronized, fast and efficient; moreover the system keeps the coordinates of the taxis automatically updated.

1.5 Goals

System goals of *MyTaxiService* application:

- [G1] The system has to guarantee to the user the possibility to access the service either through a web application or a mobile application.
- [G2] The system has to guarantee a fair management of the queues in each taxi zone.
- [G3] The system has to assign each taxi to the correct taxi zone using the GPS coordinates that receives from each taxi.
- [G4] The system has to allocate a taxi for each request or reservation.
- [G5] The system has to notify the users, both passengers and taxi drivers, about updates on taxi requests and reservations in which they are involved.
- [G6] The system hat to offer public APIs to enable the possibility to develop additional services on top of the basic ones.
- ~~[G?]The system has to answer to a taxi request by informing the passenger about the code of the taxi and the waiting time.~~
- ~~[G?] The system is able to map the requests of the clients according to their location.~~

- ~~[G?] The system is able to map the position of the taxi fleet and assign each taxi to a predetermined zone of the city according to its position.~~
- ~~[G?] The system is able to control the queue of taxis in every zone and enforce the predetermined priority rules to guarantee a fair management of the queues.~~

List of the goals of *MyTaxiService* application for taxi passengers:

- [G7] The passenger shall be able to sign up to the service.
- [G8] The passenger shall be able to log in to the service.
- [G9] The passenger shall be able to request a taxi.
- [G10] The passenger shall be able to delete a taxi request.
- [G11] The passenger shall be able to create a reservation for a taxi ride.
- [G12] The passenger shall be able to modify and delete a taxi reservation.
- [G13] The passenger shall be able to enable taxi sharing option.
- [G14] The passenger shall be able to join a shared reservation.
- [G15] The passenger shall be able to see historical data on his taxi rides.
- ~~[G?] The user can ask for a taxi simply providing his/her own position.~~
- ~~[G?] The user can plan the trip and preview the fare of the ride and decide whether to call a taxi.~~
- ~~[G?] The user can customize the reservation, specifying the date and time of the ride, the origin and destination, the willingness to share the ride.~~
- ~~[G?] The user is notified when a taxi has answered his/her request and is informed about the waiting time and possibly the fee if the ride is shared.~~
- ~~[G?] The user can check his/her own reservations and possibly edit or delete a reservation for a taxi. (???maybe only with a constraint eg. 30 minutes in advance, there is however a booking fee??)~~
- ~~[G?] The user can keep track of the favorite routes, favorite payment method? Link credit card? Maybe to pay the fee in case the user decides to delete a reservation?~~

List of the goals of *MyTaxiService* application for taxi drivers:

- [G16] Taxi drivers shall be able to log in to the service.
- [G17] Taxi drivers shall use a mobile application to inform the system about their availability.
- [G18] Taxi drivers shall use a mobile application to confirm that they are going to take care of a certain call.
- ~~[G?] The user is notified by the system when there is a client nearby waiting for a taxi.~~
- ~~[G?] The user can take in charge or reject the requests received as notification.~~
- ~~[G?] The user can cancel a service that has already taken in charge and notify the system, specifying the reason for the emergency (eg. engine failure).~~

1.6 Proposed system

The enterprise web application is going to be developed from scratch, and will also provide the counterpart mobile version for all the main smartphone OSs on the market nowadays. It will be composed of a server, which runs the business logic, generates dynamic web pages and access to the DBMS and on the other side there will be several clients who interact with the server using a web browser or the mobile application.

1.7 Definitions, acronyms, and abbreviations

1.7.1 Definitions

- Client (or Customer, Taxi passenger): is the user of the application that wants to use the taxi service.
- Taxi driver (or Taxi owner): is the user of the application that together with the back-end system makes the service functional and constantly updated, s/he controls the work which is assigned to herself/himself accepting or rejecting the proposals of clients that the system forwards.
- Ride: is a single taxi ride from a location to another one.
- Shared ride: a ride shared with other passengers arranged by the system; in this way clients that have common taxi lines can shrink the cost of the single ride.

- Location: are the GPS coordinates (or the street address) to unequivocally identify a place. It could be the position of a taxi, the origin or destination of a ride.
- Availability: is the "status" of a taxi driver, s/he can be available to take care of new jobs, thus the system should forward compatible requests, or can be unavailable, or occupied, busy, meaning that the taxi is temporarily off line with respect to the system, meaning that the system should not consider that taxi for requests assignment. The availability is set to "off" also when the taxi is carrying passengers.
- Queue: the queue of the available taxis in each zone of the city; it is updated by the system, based on the interaction with taxi drivers that can accept or reject the proposals forwarded by the system.
- Zone: the city is divided in several zones to better allocate and coordinate the taxi fleet.
- Request: it is intended as a real time call of a passenger that simply provides his/her own pickup position for an immediate service, without previous reservation.
- Reservation: it is a previously planned ride, for which the client can specify the date and time, the origin and destination, the willingness to share the ride.

1.7.2 Acronyms

- RASD: Requirements Analysis and Specification Document.
- DBMS: DataBase Management System.
- DB: DataBase.
- OS: Operating Systems.
- API: Application Programming Interface.
- HW: HardWare.
- TCP: Transmission Control Protocol.
- HTTP: Hypertext Transfer Protocol.
- HTTPS: HTTP over SSL/HTTP Secure.
- SSL: Secure Sockets Layer.

1.7.3 Abbreviations

- [Gn]: n^{th} goal.
- [Rn]: n^{th} functional requirement.
- [Dn]: n^{th} domain assumption.

1.8 Reference Documents

- Specification document: MyTaxiService project
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.

1.9 Overview

- Section 1: Introduction, it gives a brief description of the purpose, functionalities and goals of the application.
- Section 2: Overall Description, focuses more in-depth on features of the software, constraints and assumptions.
- Section 3: Specific Requirements, this part lists requirements, typical scenarios and use cases, together with UML diagrams to provide a more easy-to-read insight at the several functionalities of the software.

2 Overall description

2.1 Product perspective

MyTaxiService is a web and mobile application which is not integrated with any other existing system, it is independent and totally self-contained,. It has no interface for administration because it is entirely user based. The application does not provide any interface or API for integration with future projects.

2.2 Product functions

The function summary that is necessary for this part can be taken directly from the section of the higher-level specification in the first part of this document *Scope* (see section 1.2) and *Goals* (see section 1.5).

2.3 User characteristics

No technical expertise is required to the intended users of the new service. Part of the developers' effort are invested exactly in designing a user-friendly, self-explanatory interface, yet keeping a modern look and feel to assure an innovative user experience that fully exploits the power of present web and mobile technologies.

2.4 Constraints

2.4.1 Regulatory Policies & Safety and security

MyTaxiService fulfills all the required policies that regulate the public transportation, in particular the taxi-related ones. Besides it enforces all the requested security and privacy-connected regulations for web based transmission of information; it can make use of cookies to memorize some users' preferences. The mobile application only requires basic permissions.

2.4.2 Hardware limitations

MyTaxiService does not require any particular HW feature, except for the minimum requirements that the browser and/or the mobile phone must meet to support the latest policies for surfing the web safely.

2.4.3 Interfaces to other applications

MyTaxiService only requires access to Internet, as stated before it meets all the latest regulations for secure transmission of personal data over the web. It may use cookies, if the user accepts it.

2.4.4 Parallel operation

MyTaxiService supports parallel operations from different users when working with the DB and when dealing with all the operations done by the user after the connection.

2.5 Assumptions and dependencies

- Taxi drivers are already inserted in the system by the administrator.
- The rides can be shared only if booked in advance.
- If the taxi availability is set to "off", the system does not consider that particular taxi for jobs assignment.
- The login page for the users is unique, but according to the role of the user they will be offered the page for the passenger or the employee.
- There is neither an administrator nor a privileged user. The only authorized entity able to modify the functionality of the application is the software house itself, possibly to add new functionalities or to fix some problems on behalf of the taxi company. On the contrary, do we want to add a supervisor functionality?? Maybe a sort of command and control center that monitors the whole system??? But is it an administrator or only a passive monitor activity??
- There are no dependencies between users.
- Passengers are not allowed to book overlapping taxi rides.
- The system notifies the taxi driver allocated to a request if the correspondent passenger has deleted the reservation.
- The awaiting passenger is notified by the system if the taxi is delayed for any reason.
- The passenger is notified when other passengers sharing the same ride join the ride or cancel the reservation; the user is also updated with the new taxi fare. Which is the type of notification? SMS? email? app functionality?
- The web application and the mobile counterpart share the same functionalities.
- When a passenger books a ride s/he is given a code that identifies her/him.
- The waiting time for the passenger before the taxi arrival is computed using the actual position of the allocated taxi, but it can vary depending on the traffic.

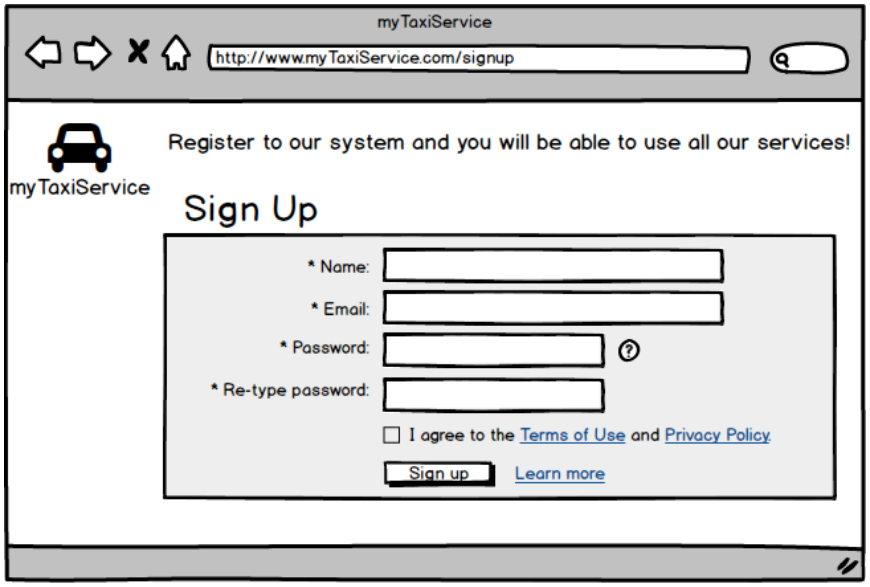
3 Specific requirements

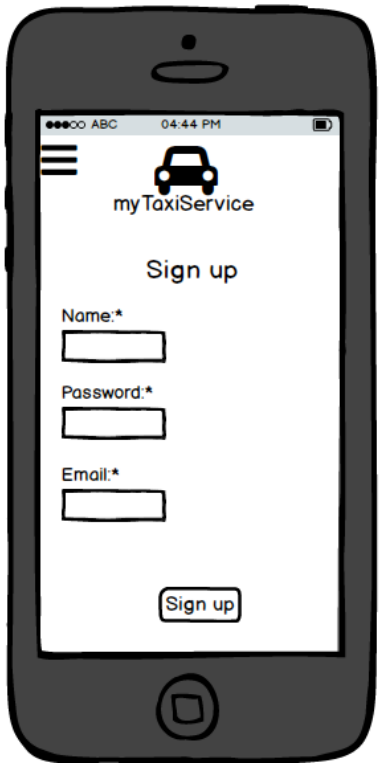
3.1 External Interface Requirements

3.1.1 User Interfaces

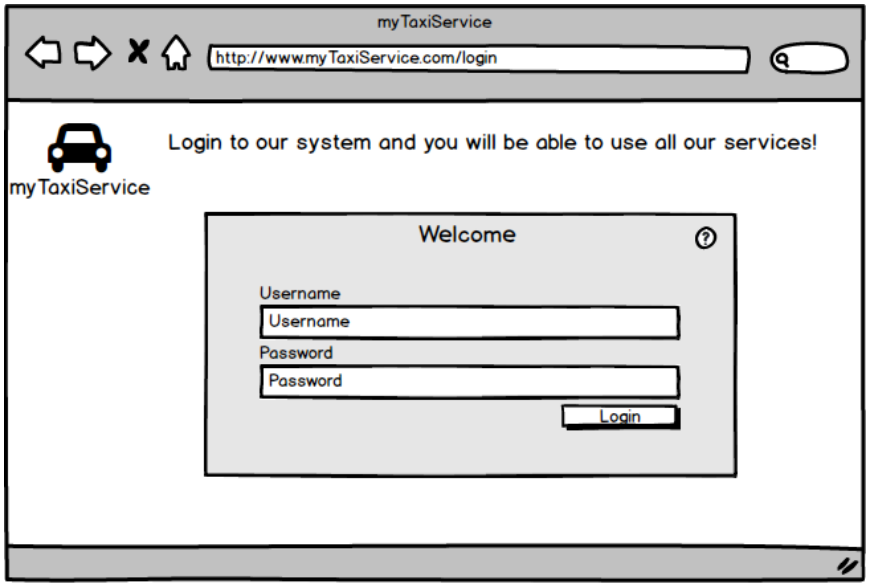
Shown below are some mock-ups that preview the user interface of the main features the system shall provide.

3.1.1.1 Sign-up This page allows the choice between two sections: the sign up as client and the sign up for taxi drivers.



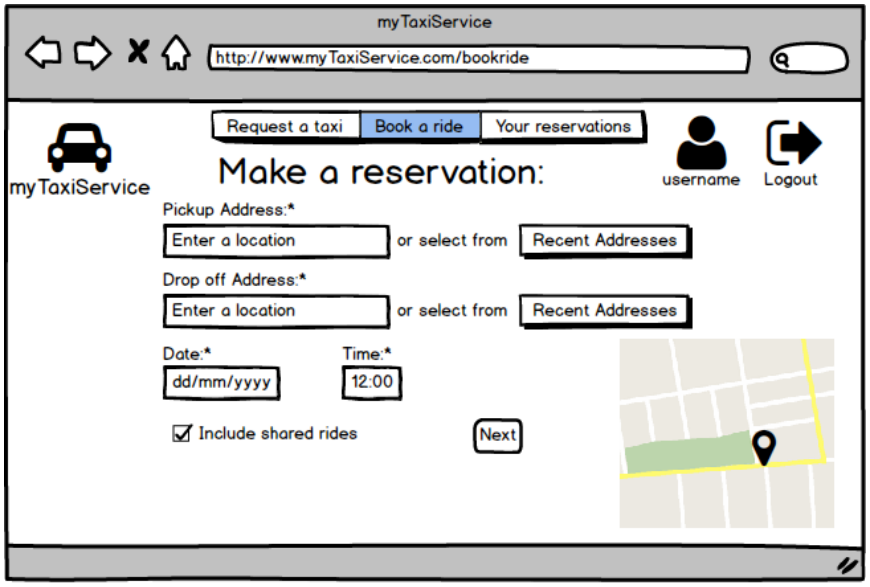


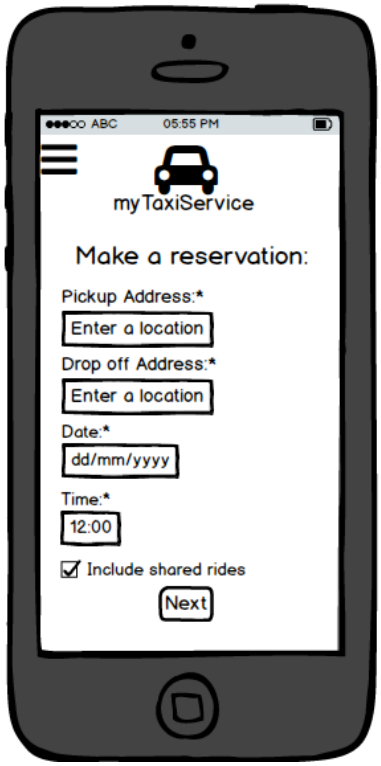
3.1.1.2 Login This page allows the choice between two sections: the login as client and the login for taxi drivers.



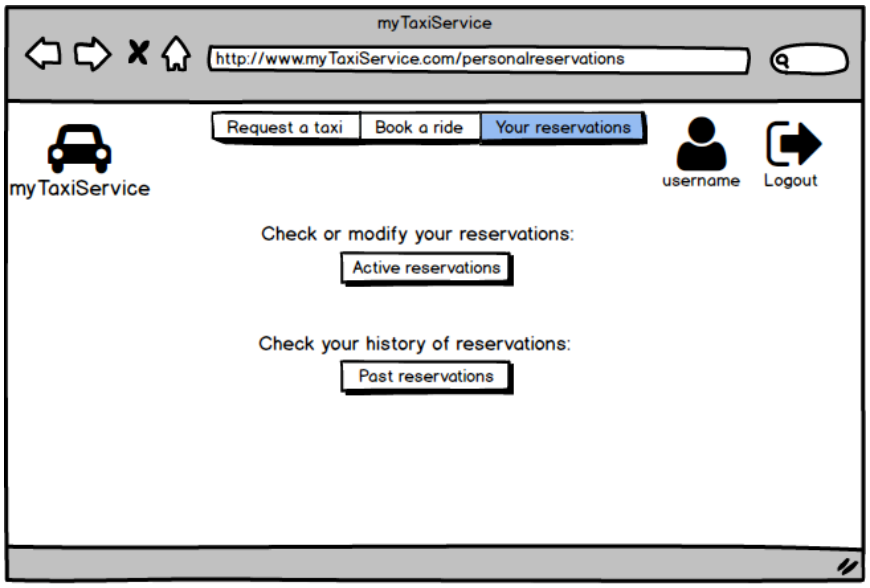
3.1.1.3 Call a taxi The user can ask for a taxi providing the pickup location through a complete address or through the GPS coordinates in the mobile app, in the latter case the phone must have the built-in GPS turned on. The user can also choose from an history of "recent addresses".

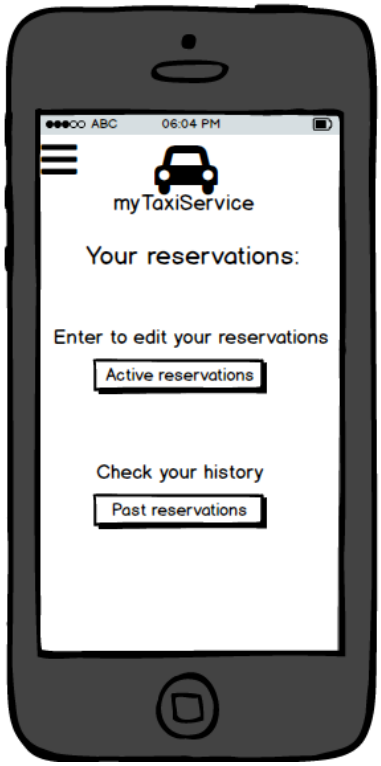
3.1.1.4 Plan a ride The user can plan a ride in advance providing the pickup and drop off locations, the date and time, the willingness to share the ride.





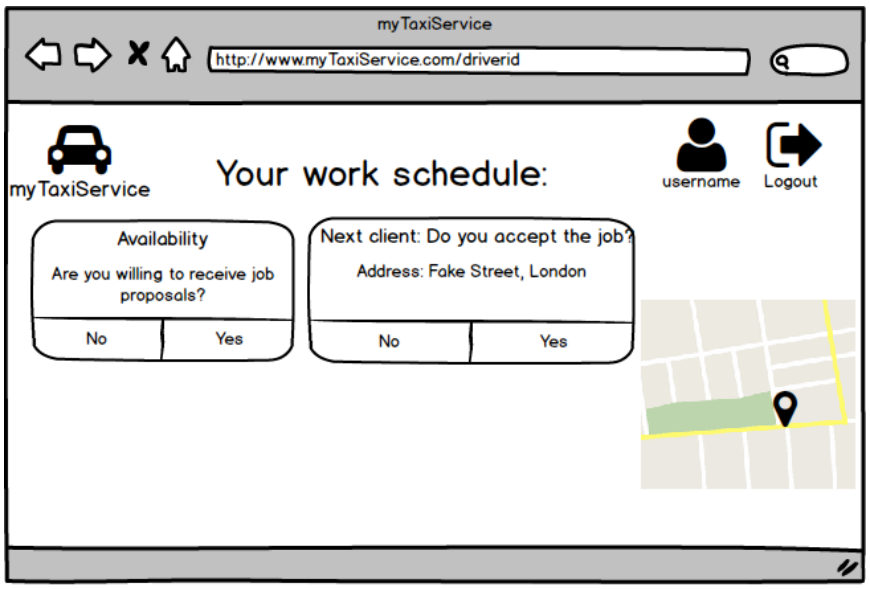
3.1.1.5 Your reservations The user can see both the active reservations and the past ones. The user can edit the active reservations within the established time frame before the meeting time.

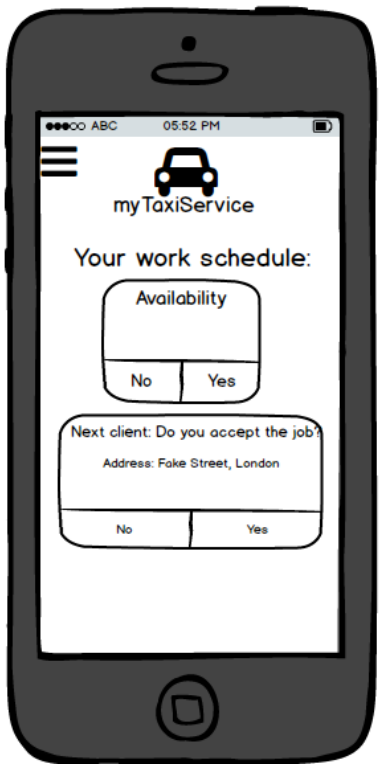




3.1.1.6 Your profile The user can edit the personal profile modifying the password, phone number, email address, payment method.

3.1.1.7 Your work The taxi driver's home page: s/he can accept or reject the requests forwarded by the system. There is also the possibility to inform the system about an "emergency situation" and the impossibility to carry out the scheduled jobs. does the system also offer a sat nav function to the driver? in this case do we need two pages, one to accept the job and one related to the current work and map?





3.1.2 Hardware Interfaces

MyTaxiService does not require and does not support any additional HW interface, it is enough a compatible smartphones or any supported browser on a computer.

3.1.3 Software Interfaces

do we have to pretend to use sql, java ee....???

3.1.4 Communication Interfaces

MyTaxiService uses the TCP transport protocol and HTTP/HTTP over SSL application layer protocol to guarantee the top of the line security on the matter of transmission of private data.

3.2 Functional Requirements

3.2.1 User Class 1

3.2.2 User Class 2

3.3 Performance Requirements

3.4 Design Constraints

3.4.1 Standards compliance

3.4.2 Hardware limitations

3.5 Software System Attributes

3.5.1 Reliability

3.5.2 Availability

3.5.3 Security

3.5.4 Maintainability

3.5.5 Portability

3.6 Other Requirements