



POLITECNICO MILANO 1863

Politecnico di Milano
A.A. 2015/2016
Software Engineering 2: “*myTaxiService*”

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

October 28, 2015

Contents

	Page
1 Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 Identifying Stakeholders	2
1.4 Identifying Actors	2
1.5 Goals	2
1.6 Proposed system.....	3
1.7 Definitions, acronyms, and abbreviations.....	4
1.7.1 Definitions	4
1.7.2 Acronyms	4
1.7.3 Abbreviations	5
1.8 Reference Documents	5
1.9 Overview	5
2 Overall description	6
2.1 Product perspective	6
2.2 Product functions	6
2.3 User characteristics.....	6
2.4 Constraints.....	6
2.4.1 Regulatory Policies & Safety and security	6
2.4.2 Hardware limitations	6
2.4.3 Interfaces to other applications	6
2.4.4 Parallel operation	7
2.5 Assumptions and dependencies.....	7
3 Specific requirements.....	7
3.1 External Interface Requirements	7
3.1.1 User Interfaces	7
3.1.1.1 Sign-up	8
3.1.1.2 Login.....	9
3.1.1.3 Call a taxi	10
3.1.1.4 Plan a ride	11
3.1.1.5 Your reservations	12
3.1.1.6 Your profile.....	13
3.1.1.7 Manage jobs.....	14
3.1.1.8 Administrator panel	15

	3.1.2	Hardware Interfaces	16
	3.1.3	Software Interfaces.....	16
	3.1.4	Communication Interfaces.....	16
3.2		Functional Requirements	16
	3.2.1	[G1] The system has to guarantee to the user the possibility to access the service either through a web application or a mobile ap- plication.....	16
	3.2.2	[G2] The system has to guarantee a fair man- agement of the queues in each taxi zone.	16
	3.2.3	[G3] The system has to assign each taxi to the correct taxi zone using the GPS coordi- nates that receives from each taxi.	17
	3.2.4	[G4] The system has to allocate a taxi for each request or reservation.	17
	3.2.5	[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.	17
	3.2.6	[G6] The system hat to offer public APIs to enable the possibility to develop additional services on top of the basic ones.	17
	3.2.7	[G7] The passenger shall be able to sign up to the service.	18
	3.2.8	[G8] The passenger shall be able to log in to the service.	18
	3.2.9	[G9] The passenger shall be able to request a taxi.....	18
	3.2.10	[G10] The passenger shall be able to delete a taxi request.....	18
	3.2.11	[G11] The passenger shall be able to create a reservation for a taxi ride.	19
	3.2.12	[G12] The passenger shall be able to modify and delete a taxi reservation.	19
	3.2.13	[G13] The passenger shall be able to enable taxi sharing option.....	19
	3.2.14	[G14] The passenger shall be able to join a shared reservation.....	19
	3.2.15	[G15] The passenger shall be able to see his- torical data on his taxi rides.	19
	3.2.16	[G16] Taxi drivers shall be able to log in to the service.	20

3.2.17	[G17] Taxi drivers shall use a mobile application to inform the system about their availability.	20
3.2.18	[G18] Taxi drivers shall use a mobile application to confirm that they are going to take care of a certain call.	20
3.2.19	[G19] The administrator shall be able to add, edit and delete taxi drivers in the system DB.	20
3.2.20	[G20] The administrator shall be able to manage and supervise the operation of the whole system, including the real-time situation of all the queues and of the taxis.	20
3.3	The world and the machine	21
3.4	Scenarios	21
3.4.1	Scenario 1	21
3.4.2	Scenario 2	21
3.4.3	Scenario 3	22
3.4.4	Scenario 4	22
3.4.5	Scenario 5	22
3.5	UML Models.....	23
3.5.1	Use Case.....	23
3.5.1.1	Visitor registers to <i>myTaxiService</i> ..	23
3.6	Non Functional Requirements	24
3.6.1	Performance Requirements.....	24
3.6.2	Design Constraints.....	24
3.6.3	Standards compliance	24
3.6.4	Hardware limitations	24
3.6.5	Software System Attributes	24
3.6.5.1	Availability & Reliability.....	24
3.6.5.2	Security	24
3.6.5.3	Maintainability	24
3.6.5.4	Portability	24
3.7	Other Requirements.....	25
4	Alloy	26

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 fare for every passenger informing all the users involved.

1.3 Identifying Stakeholders

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.
- Administrator: is the system manager. He supervises the city zones and their taxi queue. He is able to manage the list of taxi drivers in the system.

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 has to offer public APIs to enable the possibility to develop additional services on top of the basic ones.

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.

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.

List of the goals of *myTaxiService* application for the administrator:

- [G19] The administrator shall be able to add, edit and delete taxi drivers in the system DB.
- [G19] The administrator shall be able to manage and supervise the operation of the whole system, including the real-time situation of all the queues and of the taxis.

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.
- DMZ: Demilitarized Zone.

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 the interface for an administration, except for that feature it is entirely user based. The application does provide some APIs 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 and the browser can request to detect the user's position. The mobile application only requires basic permissions.

2.4.2 Hardware limitations

myTaxiService requires a computer or a smartphone connected to the internet. The mobile application requires the GPS module.

2.4.3 Interfaces to other applications

myTaxiService requires to access the Internet and the Google Maps APIs to provide map visualizations and map-related services.

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

- Entitled taxi drivers are inserted in the system by the administrator.
- The rides that can be shared are only the reserved ones.
- 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 their role they will be automatically redirected to the appropriate page for the passenger or the employee.
- There are no dependencies between users.
- Passengers are not allowed to book overlapping taxi rides.
- Passengers pay only using their valid credit or debit card by linking it to their own profile.
- Passengers always have enough money to pay the taxi fare.
- The web application and the mobile counterpart share the same functionalities.
- The waiting time can vary depending on the current traffic condition.

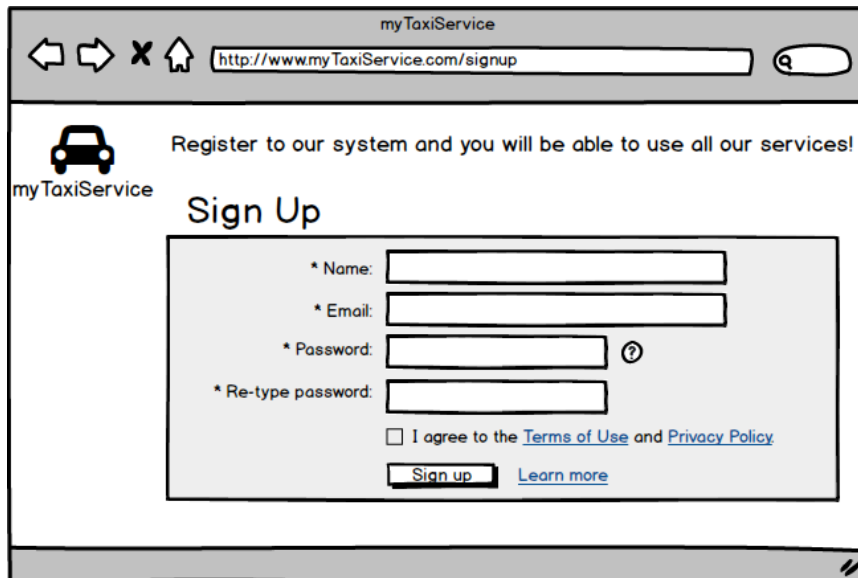
3 Specific requirements

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 presents the sign up form for the clients. The user will need to provide his personal data, home and email addresses, phone number, favorite payment option.

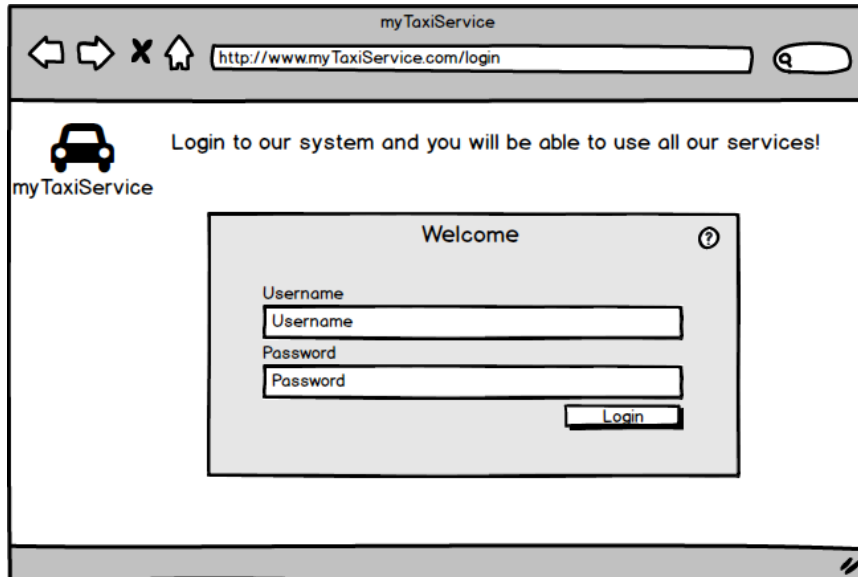


A desktop browser window showing the 'myTaxiService' sign-up page. The browser's address bar displays 'http://www.myTaxiService.com/signup'. The page features a car icon and the text 'myTaxiService' on the left. A heading 'Sign Up' is centered. Below it, a form contains four labeled input fields: '* Name:', '* Email:', '* Password:', and '* Re-type password:'. The password field has a question mark icon to its right. Below the fields is a checkbox labeled 'I agree to the [Terms of Use](#) and [Privacy Policy](#)'. At the bottom of the form are two buttons: 'Sign up' and 'Learn more'.




A mobile phone screen displaying the 'myTaxiService' sign-up page. The status bar at the top shows 'ABC' and '04:44 PM'. The app icon (a car) and 'myTaxiService' are at the top. Below is the 'Sign up' heading. The form has three labeled input fields: 'Name:*', 'Password:*', and 'Email:*'. A 'Sign up' button is at the bottom.

3.1.1.2 Login This page shows the login form for the final users.



A web browser window titled "myTaxiService" with the address bar showing "http://www.myTaxiService.com/login". The page features a car icon and the text "myTaxiService" on the left. A message says "Login to our system and you will be able to use all our services!". A central "Welcome" box contains a "Username" field, a "Password" field, and a "Login" button.



A mobile app interface for "myTaxiService" showing a "Login" screen. It includes a car icon and a hamburger menu icon. The form has "Name:*" and "Password:*" labels, each followed by an input field. A "Login" button is at the bottom.

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 automatic detection of his/her location thanks to the browser or the GPS. The user can also choose from an history of “recent addresses”.

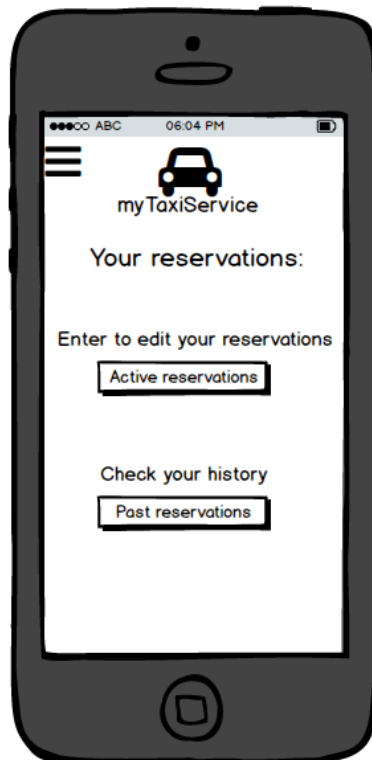
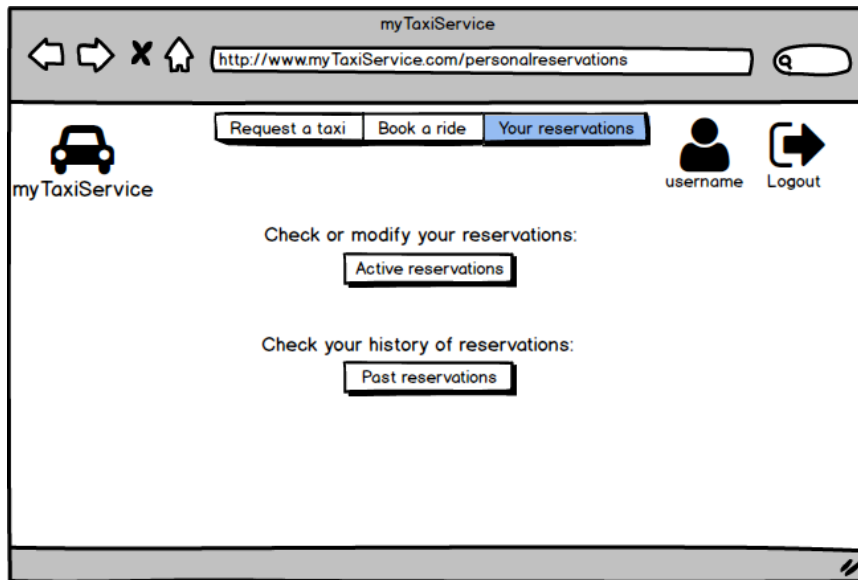


3.1.1.4 Plan a ride The user can plan a ride in advance providing the pickup and drop off locations, the date and time, the willingness to share the ride.

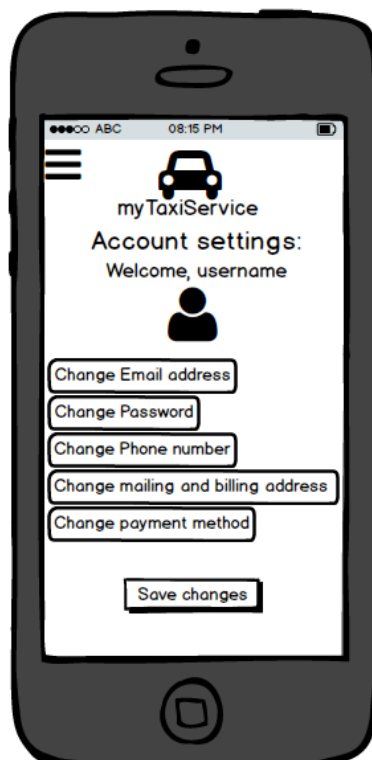
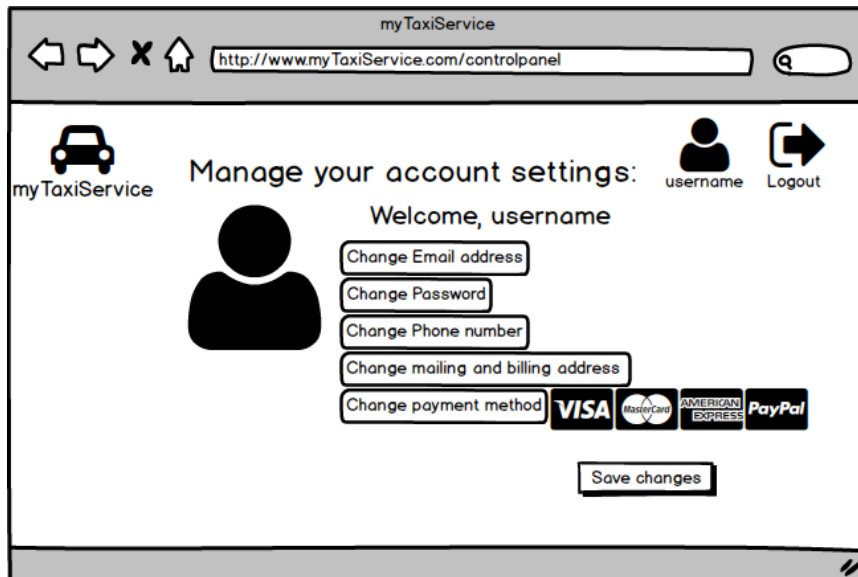
The image shows a web browser window for 'myTaxiService'. The address bar displays 'http://www.myTaxiService.com/bookride'. The page has a navigation bar with 'Request a taxi', 'Book a ride' (highlighted), and 'Your reservations'. A user profile icon shows 'username' and a 'Logout' button. The main heading is 'Make a reservation:'. The form includes: 'Pickup Address:*' with an input field 'Enter a location', a link 'or select from Recent Addresses', and a note 'or use your location'; 'Drop off Address:*' with a similar input field and link; 'Date:*' with a 'dd/mm/yyyy' input; 'Time:*' with a '12:00' input; and a checked checkbox 'Include shared rides'. A map on the right shows a route between two points.

The image shows a mobile app interface for 'myTaxiService'. The status bar at the top shows 'ABC' and '10:38 PM'. The app has a hamburger menu icon and a car icon. The main heading is 'Make a reservation:'. The form includes: 'Pickup Address:*' with an input field 'Enter a location' and a location pin icon; 'Drop off Address:*' with an input field 'Enter a location'; 'Date:*' with a 'dd/mm/yyyy' input; 'Time:*' with a '12:00' input; and a checked checkbox 'Include shared rides'.

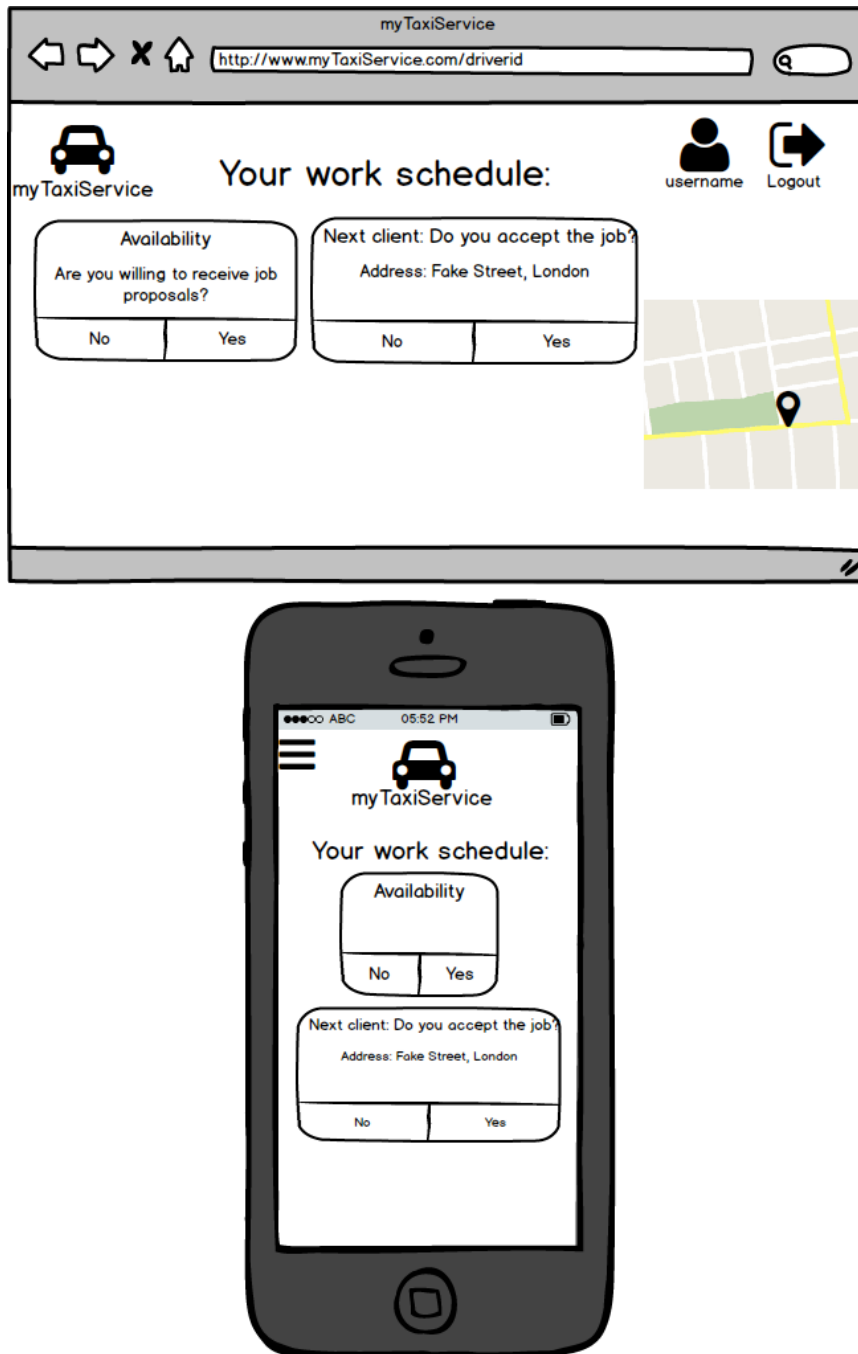
3.1.1.5 Your reservations The user can see both the active reservations and the past ones. S/he can edit the active reservations within the established time frame before the meeting time.



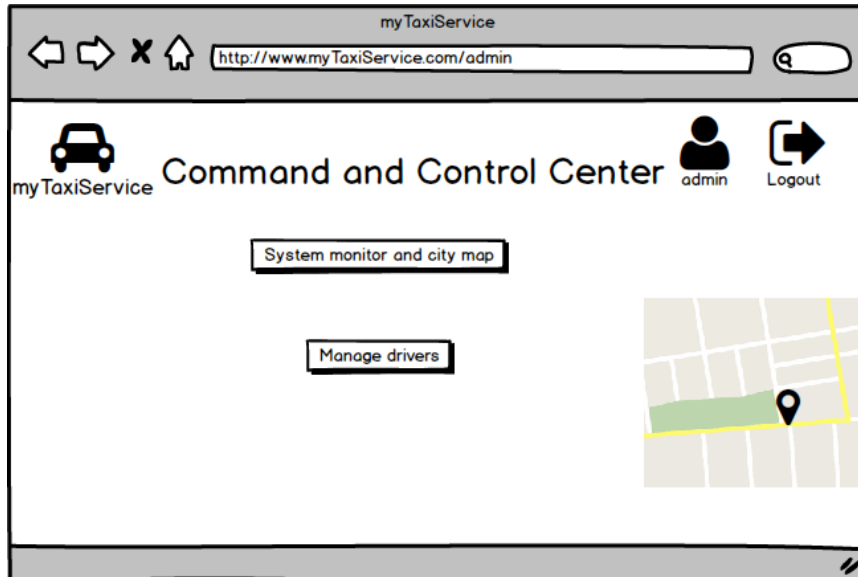
3.1.1.6 Your profile The user can edit the personal profile modifying the password, phone number, email address, permanent address, payment method.



3.1.1.7 Manage jobs The taxi driver's home page: s/he can accept or reject the requests forwarded by the system and set her/his own "availability". 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?



3.1.1.8 Administrator panel The administrator's home page: s/he can monitor the whole system or manage taxi drivers inserting new employees in the system DB or modifying/deleting existing ones.



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/HTTPS 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 [G1] The system has to guarantee to the user the possibility to access the service either through a web application or a mobile application.

- [R1] The system must offer the same functionalities in both the cases.
- [R2]
- [D1]

3.2.2 [G2] The system has to guarantee a fair management of the queues in each taxi zone.

- [R1] 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.
- [R2] If the taxi driver on top of the queue accepts a job then the system allocates the taxi for the specific request and therefore updates the queue.
- [R3] If the taxi driver on top of the queue rejects the job then the system updates the queue placing the second taxi as first and the one that rejected the job in the last position of the queue of the same zone.
- [D1]

3.2.3 [G3] The system has to assign each taxi to the correct taxi zone using the GPS coordinates that receives from each taxi.

- [R1] The mobile phone must have the GPS.
- [R2] 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.
- [D1] The GPS coordinates are always available and correct.

3.2.4 [G4] The system has to allocate a taxi for each request or reservation.

- [R1] In the event of a request the system allocates a taxi as soon as a driver in the corresponding zone is available and accepts the job.
- [R2] In the event of a reservation the system allocates the taxi 10 minutes before the meeting time with the client.
- [D1]

3.2.5 [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.

- [R1] The awaiting passenger is notified by the system if the taxi is delayed for any reason.
- [R2] 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 recomputed taxi fare. Which is the type of notification? SMS? email? app functionality?
- [R3] The system notifies the taxi driver allocated to a request if the correspondent passenger has deleted the reservation.
- [D1]

3.2.6 [G6] The system hat to offer public APIs to enable the possibility to develop additional services on top of the basic ones.

- [R1]
- [R2]
- [D1]

3.2.7 [G7] The passenger shall be able to sign up to the service.

- [R1] The user cannot register more than one time.
- [R2] The user must choose a username not already used by someone else.
- [R3] Unregistered users can only see the sign up and login page.
- [R4] The user must fill in all the mandatory fields of the proposed form to successfully complete the registration process, some of those are the phone number and the data of the credit card.
- [D1] The email address used for the registration must be valid.

3.2.8 [G8] The passenger shall be able to log in to the service.

- [R1] If the user has not the correct login information cannot access the system.
- [R2]
- [D1]

3.2.9 [G9] The passenger shall be able to request a taxi.

- [R1] The user can ask for a taxi simply providing his/her own position.
- [R2] The system has to answer to a taxi request by informing the passenger about the code of the taxi and the waiting time.
- [R3] The system is able to map the requests of the clients according to their location.
- [R4] When a passenger books a ride s/he is given a code that identifies her/him.
- [D1]

3.2.10 [G10] The passenger shall be able to delete a taxi request.

- [R1]
- [R2]
- [D1] There is the assumption (and hope) that the user will eventually wait for the taxi, since the taxi is usually immediately allocated and there is no handle-time in which the user can make up his/her own mind and retire the request without penalty fee like in the case of the reservation in advance; therefore in this case there is no penalty fee.

3.2.11 [G11] The passenger shall be able to create a reservation for a taxi ride.

- [R1] The user can customize the reservation, specifying the date and time of the ride, the origin and destination, the willingness to share the ride.
- [R2] The user can plan the trip and preview the fare of the ride and decide whether to call a taxi.
- [D1]

3.2.12 [G12] The passenger shall be able to modify and delete a taxi reservation.

- [R1] The user can check his/her own reservations and possibly edit or delete a reservation for a taxi.
- [R2]
- [D1] If the user deletes a reservation in the last 10 minutes before the meeting time (the taxi has already been allocated), then there is a fixed penalty fee automatically deducted from the user's credit card.

3.2.13 [G13] The passenger shall be able to enable taxi sharing option.

- [R1]
- [R2]
- [D1] This is only valid for reservations booked in advance.

3.2.14 [G14] The passenger shall be able to join a shared reservation.

- [R1]
- [R2]
- [D1] This is only valid for reservations booked in advance.

3.2.15 [G15] The passenger shall be able to see historical data on his taxi rides.

- [R1] The user can keep track of the recent routes and addresses.
- [R2] The user can select locations from past rides to plan new ones.
- [D1]

3.2.16 [G16] Taxi drivers shall be able to log in to the service.

- [R1]
- [R2]
- [D1] Only legally employed taxi drivers have the login information to access the system.

3.2.17 [G17] Taxi drivers shall use a mobile application to inform the system about their availability.

- [R1] The user is notified by the system when there is a client nearby waiting for a taxi.
- [R2]
- [D1]

3.2.18 [G18] Taxi drivers shall use a mobile application to confirm that they are going to take care of a certain call.

- [R1] The user can take in charge or reject the requests received as notification.
- [R2] 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).
- [D1]

3.2.19 [G19] The administrator shall be able to add, edit and delete taxi drivers in the system DB.

- [R1]
- [R2]
- [D1] The administrator only inserts in the system already verified entitled taxi drivers.

3.2.20 [G20] The administrator shall be able to manage and supervise the operation of the whole system, including the real-time situation of all the queues and of the taxis.

- [R1]
- [R2]
- [D1]

3.3 The world and the machine

“The World & The Machine” model by M. Jackson & P. Zave has been used as an additional domain analysis of *myTaxiService* application. This approach identifies the portion of “The World” affected by “The Machine”, the portion of system to be developed, and the intersection (“Shared Phenomena”) between the world and the application, that are all world information known or managed directly by the application.

INSERT SCHEMA HERE

3.4 Scenarios

3.4.1 Scenario 1

Bob lives in Milan, a very crowded and traffic congested city, for this reason he decided not to buy a car. When he needs to move around the city he finds very convenient the taxi service, less crowded than other public transport means. By now he has learned how to use all the different functionalities offered by *myTaxiService* mobile application. Whenever and wherever he needs he simply start the application on his mobile phone and he requests a taxi in a matter of seconds simply providing his location through the built-in GPS of his smartphone. In a very short time, usually no more than a couple of minutes, he receives the notification that confirms the take on responsibility of his call with the code of the incoming taxi and the estimated waiting time. Meanwhile the system reserves on the line of credit of the credit card he linked the minimum fee in case of cancellation. Once the taxi arrives Bob is taken to his destination and the system automatically computes the fare based on the distance covered and deducts the correct amount of money from his credit line.

3.4.2 Scenario 2

Alice has just moved to Milan and hasnt got any car. For this reason, she looks on the Internet to understand which options she has to move around the city with public transports and she finds *myTaxiService*’s website that presents the taxi service. She decides to sign up to be able to make use of the service, so she provides her personal data including the data of her credit card. Once the 2-step verification process is successfully completed she logs in with the credentials chosen during the registration process. She is now operative and wants to try out the service, so she enters the booking page to make a reservation for the next day. She fills in the pickup address with her own and the destination field with the museum she wants to visit and sets the date and time, enabling the shared ride choice and finally submits the request. The system confirms the reservation process went through.

3.4.3 Scenario 3

Alice is invited to her friends Bob birthday party held the next evening at Bobs house, so she logs in to the *myTaxiService*'s website to reserve a taxi for tomorrow night. Unfortunately, she doesn't recall her friends address, but she remembers that the previous week they took a taxi together to his house, so she browses the history of her own reservation and from there recognizes Bobs house address, then Alice uses it as drop off location and completes the reservation. Unluckily the following morning Bob phones to Alice and warns her that the party is postponed indefinitely due to bad weather forecast. Alice then decides to delete the reservation for taxi for the night, so she logs in to the website and cancel the booking without paying any fees.

3.4.4 Scenario 4

Mike is a taxi driver in Milan and his company relies on *myTaxiService* for the management of the work. He received the access codes to the system once he had been employed. Now every morning he starts his daily work routine by starting the *myTaxiService* app on his phone and heading to the assigned zone with his taxi to pick up the first client. He switches on the "availability to work" and starts receiving job proposals. Mike accepts them almost always; he only takes the liberty to switch off the system for a 30-minute lunch break. Often it happens that he has to pick up two clients in the same area for a shared ride because they are headed in the same direction. The shared ride feature has been having a huge approval among the clients since the implementation of the *myTaxiService* application. At the end of the day Mike stops the flow of job proposals setting the "availability" back again to off and returns to his family at home.

3.4.5 Scenario 5

Jack is the manager of the taxi company in Milan and therefore he is the administrator of the *myTaxiService* system. He spends his days monitoring the activity of the system thanks to the supervision functionality of the system. He can access it anytime anywhere thanks to the double interface for web browser and mobile application. Of course he prefers the bigger monitor of his computer when he is in the office with respect to the small display of the smartphone, but he is more relaxed knowing that he is notified if something goes wrong even when he is outside the office. Often it happens that he needs to insert in the system new just employed drivers; in that case he collects all the information about the person and registers the user with the specific functionalities for the drivers.

3.5 UML Models

3.5.1 Use Case

3.5.1.1 Visitor registers to *myTaxiService*

Actor	Visitor
Goal	[G7]
Input Condition	NULL
Event Flow	<ol style="list-style-type: none">1. Visitor on the home page clicks on the “Sign up” button to start the registration process.2. Visitor fills in at least all mandatory fields.3. Visitors clicks on the “confirm” button.4. The application will save the data in the DB.
Output Condition	Visitor successfully ends the registration process and becomes a registered user. From now on s/he can login to the application using her/his credentials and start using <i>myTaxiService</i> .
Exceptions	<ol style="list-style-type: none">1. The visitor is already a registered user.2. One or more mandatory fields are not valid or missing.3. The username chosen is already used by another user.4. The email chosen is already associated to another user. <p>All exceptions are handled alerting the visitor of the problem and the application goes back to the point 2 of event flow</p>

3.6 Non Functional Requirements

3.6.1 Performance Requirements

The service must guarantee an high level of usability, also in terms of processing and response time of the several functions. The objective is to avoid the situation in which the server represents the bottleneck of the entire system when there are many connections at the same time. To prevent this the server must be correctly chosen and calibrated to handle a large amount of connections in situations of heavy loads.

3.6.2 Design Constraints

3.6.3 Standards compliance

3.6.4 Hardware limitations

3.6.5 Software System Attributes

3.6.5.1 Availability & Reliability *myTaxiService* system must be always accessible. The server that hosts the service is online 24/7 and is equipped with an autonomous power supply backup system.

3.6.5.2 Security *myTaxiService* offers a high grade of privacy and security thanks to the protocols that it enforces. As already stated above it requires a 2-factor verification process during the registration; in this way the account is confirmed by two codes that are sent to both the provided email address and phone number. In this way it's verified that all the means used to communicate with the user really belong to him/her. The application also guarantees an appropriate level of confidentiality and integrity of the sensitive data exchanged on the Web with the server. The DB on the server is behind a DMZ in which is located the web server; firewalls with different rules are used to separate the different "zones" of access rights and permissions in the network.

3.6.5.3 Maintainability The software itself does not require any particular periodic update, unless the taxi company wants to add some functionalities. The only feature that could require some maintenance are the APIs offered by the system, but again, in this document and all the other ones related to *myTaxiService* that will be attached, future developers can find all what they could need to understand how the system works and then how to update some sections of the software.

3.6.5.4 Portability

3.7 Other Requirements

4 Alloy