Summary

**1. INTRODUCTION ............................................................................................3**

1.1 DESCRIPTION OF THE GIVEN PROBLEM .........................................................................5

1.2 GOALS ................................................................................ ..............................................5

1.3 DOMAIN PROPERTIES ................................................................................................... 6

1.4 GLOSSARY ....................................................................................................................... 6

1.5 ASSUMPTIONS ................................................................................................................. 8

1.6 PROPOSED SYSTEM ........................................................................................................ 9

1.7 IDENTIFYING STAKEHOLDERS ..................................................................................... 9

**2. ACTORS IDENTIFYING ............................................................................. 11**

**3. REQUIREMENTS ........................................................................................ 11**

3.1 FUNCTIONAL REQUIREMENTS ................................................................................... 12

3.2 NON FUNCTIONAL REQUIREMENTS .......................................................................... 13

*3.2.1 User Interface ......................................................................................................... 13*

*3.2.2 Documentation ...................................................................................................... 16*

*3.2.3 Architectural considerations ........................................................................... 16*

**4. SCENARIOS IDENTIFYING ....................................................................... 18**

**5. UML MODELS .............................................................................................. 21**

5.1 USE CASE DIAGRAM .................................................................................................... 21

5.2 USE CASES DESCRIPTION ........................................................................................... 23

*5.2.1 Help Requests Managing ................................................................................... 24*

*5.2.2 Messaging ................................................................... .......................................... 29*

*5.2.3 Profile Managing ................................................................................................... 31*

*5.2.4 User Browsing and Friendship Managing .................................................. 36*

5.3 CLASS DIAGRAM .......................................................................................................... 40

5.4 SEQUENCE DIAGRAMS ................................................................................................ 41

*5.4.1 Log in ......................................................................................................................... 41*

*5.4.2 Search for help requests ..................................................................................... 42*

*5.4.3 Post a help request ................................................................................................ 43*

*5.4.4 Reply to a help request ....................................................................................... 44*

*5.4.5 Choose the best answer to a help request .................................................. 45*

*5.4.6 Provide a feedback ................................................................................................ 46*

5.5 STATE CHART DIAGRAMS ........................................................................................... 47

*5.5.1 Help request Class .................................................................................................. 47*

*5.5.2 Friendship Request Class ................................................................................... 48*

6. ALLOY MODELLING .................................................................... 42

6.1 ALLOY CODE ....................................................................................................... 42

6.2 ALLOY WORLDS .................................................................................................. 47

*5.5.1 General World* ................................................................................................ 47

*5.5.1 Invitation Properties* ....................................................................................... 48

*5.5.1 Notification Properties* ..................................................................................... 49

**8. USED TOOLS ....................................................................................... 57**

**1. Introduction**

**1.1 Description of the given problem**

The aim of this project is to improve the management of the taxi service in a large city, mostly by simplifying the access of passengers to the service and by simplifying the access of passengers to the service and by guaranteeing a fair management of taxi queues.

To further ensure the latter, passengers can also make a reservation by making a request at least 2 hours earlier and by specifying the origin and destination of the ride. To adapt to all situations, the system allows passengers to make requests either through a web application or a mobile app.

The requests will be passed on to the taxi drivers by the system and they will answer through an app.

The taxi drivers will be asked whether they want to accept a call only if they have previously notified the system about their availability.

**1.2 Goals**

We thought about the possible behaviors of those who will interact with the system and what myTaxiService should provide them, so we planned to give our system these features:

 Users should be able to:

• Sign up into the system;

• Log into the system;

• Request a taxi;

• Cancel a request;

• Reserve a taxi;

• View the reservations;

• Modify a reservation.

• Cancel a reservation.

 Taxi drivers should be able to:

• Inform the system about their availability;

• Accept a call from the system;

• Refuse a call from the system;

• Inform the system they can no longer take care of a call they’d previously accepted.

**1.3 Domain Properties**

We suppose that these conditions hold in the analyzed world:

• GPS localization is used to know where the taxis are;

• Taxi drivers must be inside the city;

• Taxi drivers should be able to use the necessary technology to communicate with the system.

• Users should be able to use the necessary technology to communicate with the system.

• Almeno un tassista disponibile that going to take care of a certain call per ogni zona.

**1.4 Glossary**

Here are the definitions of some words that will be used in our documents.

• USER: by user we mean a person who is already registered in the system. This means they have a profile, through which they can make requests and reservations, and manage them.

• GUEST: a guest is a person who hasn’t signed up yet.

Unlike registered users, they cannot make requests and reservations, but they can still access the app, either to sign up or to read about the available features they will be able to access once they register.

• TAXI DRIVER: by taxi driver we mean a person who will interact with the system with the purpose of serving users. Taxi drivers can inform the system about their availability and they can confirm that they are going to take care of a call that the system is trying to assign to them.

• QUEUE: by queue we mean a queue of available taxis in a zone.

• REQUEST: a request can be made by a registered user when they want a taxi right away. Once a user has made a request, a taxi will come pick them up shortly after, at the location specified by the user.

• RESERVATION: a reservation can be made by a registered user when they want to book a taxi at a certain time. If a user wants to make a reservation, they will have to provide additional details, namely, the origin and destination of the ride, along with the meeting time. Reservations must be made at least 2 hours before the desired meeting time.

**1.5 Assumptions**

There are a few points that are not very clear in the specification document, so we will have to assume some facts.

We assume that:

• The user can cancel requests and reservations.

• The user can modify reservations. If a user wants to modify a reservation, they must notify the system at least 10 minutes before the meeting time, otherwise the reservation will be cancelled.

• If a taxi driver informs the system that he is available and then they decide to move from a zone to another, then the system will remove its identifier from the queue of the previous zone and move the taxi in the last position of the queue of the new zone.

• Once a call has been forwarded to an available taxi driver, a timer starts. If the taxi driver doesn’t confirm by the time the timer runs out, the system assumes the taxi driver is no longer available, it moves them to the last position in the queue and it forwards the request to the next available driver in the queue.

• If a taxi driver accepts a call, the system will automatically set the taxi driver as unavailable and therefore remove from the queue.

• bisogna dire cosa fa il driver nel caso abbia un incidente mentre sta andando a prendere un passenger? 🡪 Si, assumiamo che appunto :

• Inform the system they can no longer take care of a call they’d previously accepted.

**1.6 Proposed system**

piattaforma web, applicazione mobile per passeggeri e applicazione mobile per tassisti.

BASATI SU QUESTO

We will implement an enterprise application based on the web using the platform JEE. It will be composed of a server, which runs the business logic, generates dynamic web pages and access to data sources. On the other side there will be several clients that interact with the server using a web browser and a graphical user interface.

Weather forecasts are taken form an online web application that provides free to use APIs to access a data base containing weather information from the whole world, both historical data and forecasts.

**1.7 Identifying stakeholders**

Our main stakeholder is the government, who gave us the delivery of the project to optimizing its taxi service.

Gli altri stakeholder sono gli utenti e I tassiti.

**2. Actors identifying**

The actors of our system are basically three:

 Guest: a guest is someone who hasn’t signed up yet. A guest can either sign up or read about the possible features they will be able to enjoy once they sign up.

User: a user is someone who has already signed up and therefore can use the service to book a taxi, either through a request or through a reservation.

Taxi Driver: a taxi driver is an employee of our service who gets paid for driving customers to their desired destinations upon request. Taxi drivers actively interact with the system by exchanging information about their availability.

**3. Requirements**

We’ve determined the following requirements according to the Jackson and Zave analysis, so assuming that the domain properties, which are described above, holds for our purposes, we’ve written the requirements in order to satisfy goals.

• Registration of a person to the system

− The system has to provide a sign up functionality.

• Log in of a person to the system

− The system has to provide a log in functionality.

• Request a taxi

− The system has to provide a function to allow a user to create a request inside of the city.

− The system forwards a request that comes from a certain zone to the first taxi queuing in that zone.

• Cancel a request

− The system has to provide a function to allow a user to cancel a request.

− The system has to provide a function to show to the taxi driver that the call has been cancelled.

− The system has to provide a function to chidere conferma della sua disponibilità dopo esser stato avvisato della chiamata cancellata.

• Reserve a taxi

− The system has to provide a function to allow a user to make a reservation

− The system forwards a reservation to the first taxi queuing in the zone of the origin of the ride, 10 minutes before the meeting time with the user.

• View the reservations

− The system has to provide a function that allows the user to see their reservations.

• Modify the reservations

− The system has to provide a function that allows the user to modify a reservation.

− The system blocca la funzione modifica, se mancano meno di 2 ore alla corsa.

• Cancel a reservation

− The system has to provide a function to allow a user to cancel a reservation.

− The system has to provide a function to show to the taxi driver that the call has been cancelled, se è gia stato chiamato.

− The system has to provide a function to chidere conferma della sua disponibilità dopo esser stato avvisato della chiamata cancellata. (se è già stato chiamato)

• Taxi driver inform the system about their availability;

− The system has to provide a function to store the identifier of the available taxi in the queue of taxis in the corresponding zone.

• Taxi driver accept a call from the system;

− The system toglie il taxi dalla coda

− The system lo mette come occupato

− The system calcola il tempo di atteso sulla base delle informazioni tratte dal gps

− The system answers to the request by informing the user about the code of the incoming taxi and the waiting time.

• Taxi driver refuse a call from the system;

− The system forward the request to the second in the queue

− The system move the first taxi in the last position in the queue.

• Taxi driver inform the system they can no longer take care of a call they’d previously accepted;

− The system forward the request to the second in the queue

− The system mette in automatico che il tassista non è piu dispon

− The system rimuove il taxi dalla coda

**3.1 Functional requirements**

Now that we have defined the main feature of mytaxiService, we can find some functional requirement concerning each defined actor:

 Guest: he can

• Sign up.

 User: he can

• Log in;

• Modify his profile information;

• Request a taxi;

• Cancel a request;

• Reserve a taxi;

• View the reservations;

• Modify a reservation.

• Cancel a reservation.

 Taxi Driver: he can

• Inform the system about their availability;

• Accept a call from the system;

• Refuse a call from the system;

• Inform the system they can no longer take care of a call they’d previously accepted.

**3.2 non-functional requirements**

**3.2.1 user interface**

**3.2.2 documentation**

**4. Scenarios identifying**

Basic scenario

1-Bob needs to go to work but his car broke down the previous evening, therefore he decides to go by taxi. After having breakfast, he gets on his computer and logs in on his myTaxiService account to request a taxi, by filling a form where he opts for a simple request. Since he's making a simple request he only has to fill in a form saying the address where he lives. He receives a confirmation of the request which contains the code of the incoming taxi and the waiting time. Some minutes later, a taxi shows up, Bob notices the code of the taxi is the one he's waiting for, so gets on it and he tells the driver the address of his workplace. The taxi driver takes Bob to work on time.

**(manca il tassista come attore, che visualizza e accetta e dopo averlo accompagnato, torna disponibile)**

scenario with reservation

2-Alice is on a business trip, she will arrive in the city by train and she will have to stay at a hotel overnight. The train ride will be 4 hours long and she will arrive at the station late at night. Because she knows she will be tired, she wants to make sure she will find a taxi waiting for her right outside the station. Therefore, as soon as she gets on the train, she uses her phone to log in on her myTaxiService account to make a reservation. She fills in all the necessary fields, setting the station as the origin of the ride, the hotel as the destination, and specifying she wants the meeting time to be shortly after the train gets to the station. Around 4 hours later, just before getting off the train, Alice gets a notification from the system which tells her the code of the taxi that will pick her up at the station. She meets the taxi driver just outside the station and he drives her to the hotel. **(manca il tassista come attore, che visualizza e accetta e dopo averlo accompagnato, torna disponibile)**

3-On a weekday Carl and his girlfriend are both stressed and tired from work and they agree on spending some relaxing time together during the weekend, therefore Carl suggests to go for a picnic on the beach this Saturday. Carl doesn't want to go to the beach by car because he knows finding a car park will stress him even more, so he goes on myTaxiService and makes a reservation for a taxi on Saturday morning. Friday night, right before going to sleep, Carl watches the weather forecast on TV, which warns of an incoming heavy storm that is going to make the weather rainy all through the weekend. Knowing that he will have to cancel his plans, Carl goes on myTaxiService application and cancels the reservation he had made a few days earlier.

4-John needs to go to the mall by taxi, so he requests a taxi on myTaxiService. The system checks the queue associated to the taxi zone where John lives and finds the first available cab in the queue, which is number 358. Mark, the taxi driver driving cab number 358, accepts the call and John receives a confirmation from the system which says a cab will be there to pick him up in 10 minutes. A couple of minutes later, Mark hears a noise coming from his cab and he notices that he got a flat tyre. Because of this, he decides he should give up the call and let the system assign the call to another cab, so he notifies the system of what's happened. The system assigns the call to the next available taxi driver and moves Mark to the last position in the queue. After doing this, the system also notifies John that there will be a delay of about 10 minutes; moreover John receives the code of the new cab that is coming to pick him up. Eventually, the new cab arrives and takes John to the mall.

5- chiama il taxi, ma passa un suo amico a prenderlo e quindi cancella la richiesta.

6- un tassista che non accetta la chiamata per un motivo bo

7- uno prova a mettere una via non all interno della città, il sistema lo blocca e quindi il tipo capisce di aver scaricato l app sbagliata ehehe...ed esce dal sistema

8- uno fa una prenotazione, si accorge di aver sbagliato l indirizzo di destinazione e gli viene permessa la modifica perché entro le due ore dalla corsa.

9- prenota e cancella quando ormai il tassista è già stato chiamato.