



Politecnico di Milano, A.A. 2015/2016

Software Engineering 2: My Taxi Service
Requirements **A**nalysis and **S**pecification
Document

Belotti Nicola
Chioso Emanuele 791621
Colombo Andrea

October 28, 2015

Contents

1	INTRODUCTION	2
1.1	Description of the given problem	2
1.2	Actors	2
1.3	Goals	3
1.4	Glossary	3
1.5	Domain Properties	4
1.6	Assumptions	4
1.7	Identifying Stakeholders	5
1.8	Proposed System	5
2	REQUIREMENTS	6
2.1	Functional Requirements	6
3	SCENARIO IDENTIFYING	9
4	UML MODELS	10
4.1	Use case diagram	10
4.2	Use case Description	10
4.3	Class Diagram	10
4.3.1	Log In	10
4.3.2	Reserve a taxi	10
4.3.3	altra roba altra roba	10
5	ALLOY MODELLING	11
5.1	Alloy Code	11
5.2	Alloy Worlds	11
5.3	non so cosa sia ahahahalloy	11
6	USED TOOLS	12

1 INTRODUCTION

1.1 Description of the given problem

We will project My Taxi Service, an online service that will provide passengers an easy and reliable way to access taxi service, while allowing taxi drivers to organize themselves and make their job simple.

There will be two types of users: passengers and taxi drivers.

Passengers who want to access the service should register in the system by providing information like name, address, phone number and e-mail. Once registered passengers will be able to:

- Request a taxi by specifying a valid location, the system will confirm the request by sending the user a code;
- Make a reservation for a taxi, in this case the passenger must provide the destination too, the reservation must be done at least 2 hours before the ride. If the reservation is successful the system will start searching for a taxi ten minutes before the meeting time.
- Enable taxi sharing option, meaning that he wants to share the ride with others and thus the cost of the ride. When making a reservation the user can decide to share the ride. Other passengers decide to join the ride the system will create a path for a taxi.

To taxi drivers will be provided a different account with different functionalities. They will be able to set their availability, if they are available the system can call them to go to a specified location to pick up a passenger, a call that drivers can accept or refuse.

The system will optimize taxi queues by dividing the city in different zones of 2 km square size, and will assign to each zone a certain number of taxis organized in a queue. Each request is forwarded to the first taxi in the queue associated to the zone from which the request comes.

1.2 Actors

- Visitor: a non registered user can only see the main page, log-in page and registration page. He can register himself by the completion of the registration form.
- Passenger: a passenger has already an account, he can log-in in the system. After that he can call for a taxi using the apposite form.

- Taxi-Driver: a taxi driver has an account provided by the company. From this account he can manage his profile, set his availability, accept or refuse a call from the system.

1.3 Goals

Here's the list of the goals of our application

- [G1] Allow a visitor to register in the system and adding/managing his information.
- [G2] Allow a user to log in to application, either he is a passenger or a taxi driver.
- [G3] Allow a passenger to make a request for a taxi.
- [G4] Allow a passenger to make a reservation for a taxi.
- [G5] Allow a passenger to enable the taxi sharing option and create a shared ride.
- [G6] Allow a passenger to cancel a request or reservation for a taxi or disabling the taxi sharing option.
- [G7] Allow a taxi driver to set his availability.
- [G8] Allow a taxi driver to accept a call for a taxi-request from the system.
- [G9] Allow a taxi driver to refuse a call for a taxi-request from the system.
- [G10] Provide a fair management of taxi queues.

1.4 Glossary

We will give a specific definition of some crucial terms that we are going to often use in our documentation of the project to prevent some ambiguity of the natural language.

Visitor Every person that visits the website or downloads the application before registration. Registered users are seen as Visitors before the login.

Users Every single person registered in the Database of the service. It includes Passengers and Drivers

Passengers Clients registered in the database, they can only request a taxi and reserve one.

Drivers Taxi Owners registered in the database, they can set their availability depending on their needs and answer or refuse calls from the system.

Request When a Passenger uses the service to find a ride, he makes a Request. He must insert where he needs to go. The Request is sent to the available drivers who can accept or refuse it. It's the interaction that connects passenger and driver.

Reservation When a passenger uses the service to reserve a taxi so he's sure that the taxi will be available when he will need it.

1.5 Domain Properties

We suppose that —andreciao—

- Once a request is done, it cannot be deleted.
- If a passenger makes a request and the request is accepted, he will show up at the established location in time.
- If a driver accepts a request, he will show up at the established location in time.
- If a passenger joins a shared ride, he will take part to the ride.
- A passenger will not ask for a taxi in any way at a certain time if he knows there will be a conflict in schedules.

1.6 Assumptions

- Users cannot have more than one request open at the same time.
- Users can't create or join more than one shared ride at the same time.
- Reservations must be done at least two hours before the ride.
- Users cannot make requests if a reserved ride is taking place within 30 minutes.

- Reservations can be cancelled at most 30 minutes before the scheduled time.
- There will be a notification by e-mail or sms and through the application 10 minutes before a reserved ride.
- There will be a notification sent to the current members of a shared ride when a new passenger joins the ride.
- Every taxi has an attribute that shows the maximum capacity of the vehicle.

1.7 Identifying Stakeholders

–da vedere. possibili stakeholders: - grosse compagnie di taxi in una città ed il governo di una determinata città.

1.8 Proposed System

The application we will project can be implemented as an enterprise application based on the web, with a Client-Server architecture. The server will run the logic and generates web pages, a database system will be used to record information of the users. On the other side there will be several clients connecting using a web browser and a graphical user interface, or using a mobile application.

2 REQUIREMENTS

2.1 Functional Requirements

By analyzing the goals we came up with a list of requirements in order to achieve them:

- [G1] Allow a visitor to register in the system and adding/managing his information.
 - [R1] The system will provide a registration functionality.
 - [R2] System should check that username must be unique, there cannot be two users with the same username in the system.
 - [R3] System will not allow visitors to see other pages than the login page.
 - [R4] System will grant visitors access only to registration functionality.
- [G2] Allow a user to log in to application, either he is a passenger or a taxi driver.
 - [R1] The system will provide a log-in functionality.
 - [R2] System will check that the tuple username-password inserted by the user exists in the database.
- [G3] Allow a passenger to make a request for a taxi.
 - [R1] The system will not grant access to this functionality if the user is not logged in.
 - [R2] The system will forward a taxi request to a driver only if:
 - * The passenger provides a valid location for a taxi.
 - * Passenger is not waiting for another taxi called by a previous request.
 - * Passenger does not have a reserved ride occurring within thirty minutes.
- [G4] Allow a passenger to make a reservation for a taxi.
 - [R1] The system won't grant access to this functionality if the user is not logged in.
 - [R2] The system will accept the reservation if the passenger:

- * Specifies starting position, destination and leaving time of the ride
 - * Completes the reservation two hours before the ride occurs.
 - * Did not make a reservation for a ride that occurs thirty minutes before or after the requested time.
- [R3] If the reservation is succesful the system will call for a taxi 10 minutes before the scheduled time and send a notification to the passenger.
- [G5] Allow a passenger to enable the taxi sharing option and create/join a shared ride.
 - [R1] The system won't grant access to this functionality if the user is not logged in.
 - [R2] If a user wants to share a ride the system will ask him to enable sharing option before starting position, destination and leaving time of the ride, then he will wait until the taxi is full or until 10 minutes before the scheduled time for other users to join the ride and finally compute a path for the ride.
 - [R4] If a passenger wants to join a shared ride the system will ask him the starting position and destination he's headed, show all the possible non full rides heading in the same direction, wait for user decision and then add the user to the shared ride.
 - [R5] The system will also calculate the distance and the fee of a ride and communicate it to both passengers and driver.
- [G6] Allow a passenger to cancel a reservation.
 - [R1] System will not allow a passenger to cancel a ride if it will occur in less than thirty minutes.
- [G7] Allow a user to disable taxi sharing option and cancel or leave a shared ride.
 - [R1] The system will permit the user to disable the taxi sharing option via the same interface used to enable it.
 - [R2] The system will reject a cancel request of a shared ride if someone has already joined it.
 - [R3] The system will not allow a passenger to leave a shared ride if it will occur in less than thirty minutes.

- [G7] Allow a taxi driver to set his availability.
- [G8] Allow a taxi driver to accept or refuse a call for a taxi-request from the system.
- [G9] Allow a taxi driver to accept or refuse a reservation.
- [G10] Provide a fair management of taxi queues.

3 SCENARIO IDENTIFYING

4 UML MODELS

4.1 Use case diagram

4.2 Usa case Description

4.3 Class Diagram

4.3.1 Log In

4.3.2 Reserve a taxi

4.3.3 altra roba altra roba

5 ALLOY MODELLING

5.1 Alloy Code

5.2 Alloy Worlds

5.3 non so cosa sia ahahahalloy

6 USED TOOLS