Requirements Analysis and Specification Document: myTaxiService

Chitti Eleonora, De Nicolao Pietro, Delbono Alex

October 25, 2015

Contents

1 Introduction **2**
- 1.1 Purpose . . . 2
- 1.2 Scope . . . 2
- 1.3 Definitions, acronyms, and abbreviations . . . 3
- 1.4 References . . . 3
- 1.5 Overview . . . 4

2 Overall description **5**
- 2.1 Product perspective . . . 5
  - 2.1.1 User interfaces . . . 5
  - 2.1.2 Hardware interfaces . . . 6
  - 2.1.3 Software interfaces . . . 6
- 2.2 Product functions . . . 6
- 2.3 User characteristics . . . 7
- 2.4 Constraints . . . 8
  - 2.4.1 Regulatory policies . . . 8
  - 2.4.2 Hardware limitations . . . 8
  - 2.4.3 Interfaces to other applications . . . 8
  - 2.4.4 Reliability requirements . . . 8
  - 2.4.5 Criticality of the application . . . 8
  - 2.4.6 Safety and security considerations . . . 9
- 2.5 Assumptions and dependencies . . . 9
- 2.6 Future extensions . . . 9

3 Specific requirements **11**
- 3.1 External interface requirements . . . 11
  - 3.1.1 User interfaces . . . 11
  - 3.1.2 Hardware interfaces . . . 11

3.1.3 Software interfaces 11
3.1.4 Communications interfaces 11
3.2 System features 11
3.2.1 User registration 11
3.2.2 User login 14
3.2.3 Standard taxi call 14
3.2.4 Ride request notification to the taxi driver 17
3.2.5 Taxi availability handling 19
3.2.6 Taxi reservation 19
3.2.7 Ride sharing 20
3.2.8 User profile management 20
3.3 Performance requirements 20
3.4 Design constraints 21
3.5 Software system attributes 21
3.5.1 Reliability 21
3.5.2 Availability 21
3.5.3 Security 21
3.5.4 Maintainability 21
3.5.5 Portability 21
3.6 Other requirements 21

Chapter 1

Introduction

1.1 Purpose

This document is the Requirement Analysis and Specification Document for the myTaxiService application. Its aim is to completely describe the system, its components, functional and non-functional requirements, constraints, and relationships with the external world and to provide typical use cases and scenarios for all the users involved. Further, this document will provide formal specification of some features of the applications.

This document is written for project managers, developers, testers and Quality Assurance. It may be useful also to users. It may be used in a contractual requirement.

1.2 Scope

The system is a taxi reservation and dispatching system for large cities. Its aim is to simplify the access of passengers to the service and to guarantee a fair management of taxi queues.

The system consists in a back-end server application (*myTaxi Server*), a web application front-end (*myTaxi Web*) and in a mobile application (*myTaxi Mobile*).

The system has 2 types of users: passengers and taxi drivers; it should allow the users to sign up and login with their credentials. The system has to know the location of both the passengers and the taxi drivers.

The system allows any passenger to request a taxi, informing him o her of the incoming taxi code and the estimated waiting time.

The system knows about the available taxi drivers and, when a request is incoming, informs one of them about the location of the available passenger; the taxi driver can either accept or deny the ride. If the taxi driver accepts the ride, the system sends a confirmation to the passenger, together with the estimated waiting time. If the taxi driver rejects the ride, the system looks for another taxi driver in the same way.

The system offers programmatic interfaces (APIs) to enable the development of additional services on top of the basic one.

The system is provided with two optional modules:

Taxi reservation allows the passenger to reserve a taxi by specifying the origin and the destination of the ride.

Taxi sharing allows the passengers to share a ride together, dividing the costs.

1.3 Definitions, acronyms, and abbreviations

RASD: Requirements Analysis and Specification Document.

System: the whole software system to be developed, comprensive of all its parts and modules.

Module: an optional software component which uses the core system APIs to provide additional features.

Passenger: the registered user who uses the service for a taxi ride.

Taxi driver: any taxi driver subscribed to the service.

User: any use (passenger or taxi driver) subscribed to the service.

1.4 References

- Project rules: "AA 2015-2016 Software Engineering 2 - Project goal, schedule and rules"

- Assignment: "Software Engineering 2 Project, AA 2015/2016 - Assignments 1 and 2"
- IEEE Std 830-1998: "IEEE Recommended Practice for Software Requirements Specifications"

1.5 Overview

This document is structured in three parts:

Chapter 1: Introduction. It provides an overall description of the system scope and purpose, together with some information on this document.

Chapter 2: Overall description. Provides a broad perspective over the principal system features, constraints, and assumptions about the users and the environment.

Chapter 3: Specific requirements. Goes into detail about functional and nonfunctional requirements. This chapter is arranged by feature.

Chapter 2

Overall description

2.1 Product perspective

The back-end stores its data in a RDBMS and can run on every platform that supports the JVM. The web applications works on any web server that supports PHP. The mobile application is supported by Android, iOS. The system provides APIs to extend its functionalities, e.g. :

- taxi reservation
- ride sharing
- online payments
- ...

2.1.1 User interfaces

The user interfaces must provide the following logical characteristics both in the mobile app and in the web interface:

- The possibility to choose the language used in the contents in every moment during every operation.
- A first screen in which the user must login to begin operations.
- A dashboard with links to every function in order to show the user the capabilities of the system and allow him to save time.

- A link to the dashboard in every screen.
- A reminder in the top bar to show the last taxi service called, with a link to a screen which displays the reserved taxi history

2.1.2 Hardware interfaces

The system has to deal with the dichotomy of the web user interaction and the mobile one. It is necessary to provide a common look and feel, without losing simplicity with the different hardware interfaces. For instance, the compilation of data fields has to be made with multiple choices in order to simplify the user's experience of the app. Same goes for the dimension of buttons that can not be too small.

2.1.3 Software interfaces

The required software products used by the systems are:

- MySQL 5.7 `http://dev.mysql.com`
- Java SE 8 `http://java.com`

2.2 Product functions

The system allows passengers to book a taxi and notify taxi drivers of the request.

In particular it lets users to:

- Passengers:
  - create an account
  - request a taxi
  - share a taxi with other passengers
- Taxi drivers:
  - create an account
  - accept or deny a lift request

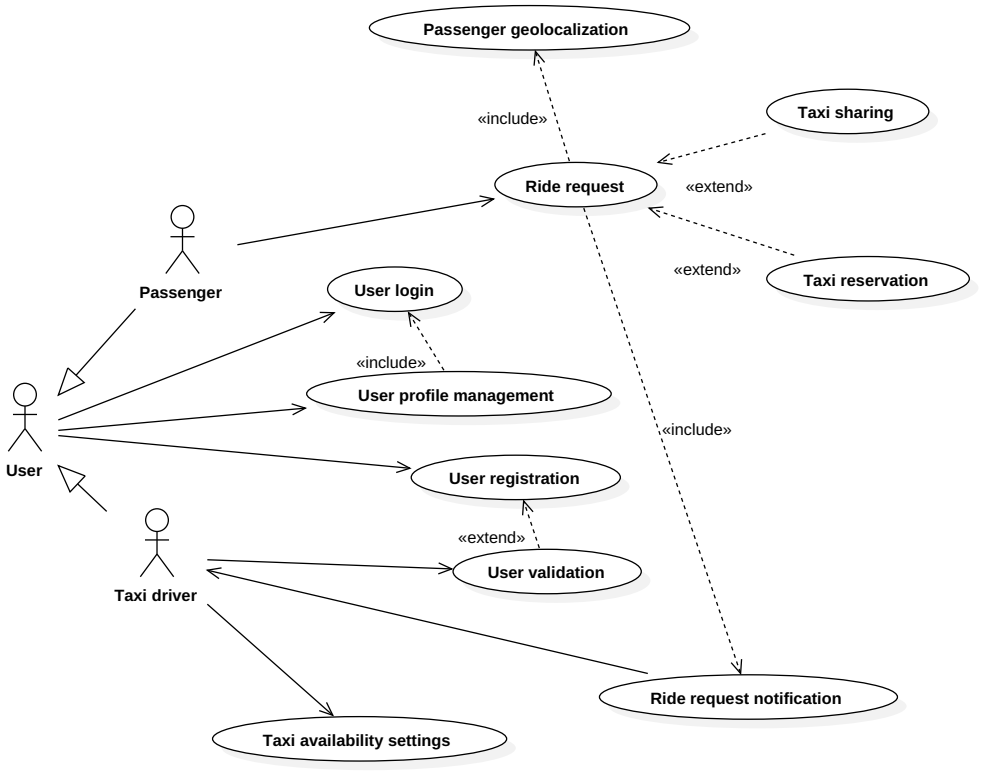

Figure 2.1: The comprehensive use-case diagram of all the functionalities implemented by the system.

2.3 User characteristics

The two kinds of users are *passengers* and *taxi drivers*. We suppose that both kinds of users have access to Internet.

Taxi drivers must be able to install and use the mobile application on their cellphone to answer the ride requests.

Passengers have to use the browser application or the mobile app.

2.4 Constraints

2.4.1 Regulatory policies

It's user responsibility to ensure that the use of the system complies with the local laws and policies.

The system must ask the user for the permission to acquire, store and process personal data and web cookies. The system must offer to the user the possibility to delete all the personal data.

2.4.2 Hardware limitations

the system has to run under the following conditions:

- App
  - 3G connection, at 2 Mb/s
  - 100 MB of free space
  - 2 GB of RAM
- Web application
  - Support for current version of IE, Firefox, Chrome, Safari, Opera as of 2013
  - 2 Mb/s Internet connections
  - 800x600 resolutions

2.4.3 Interfaces to other applications

2.4.4 Reliability requirements

The system must have a minimum availability of 98%.

2.4.5 Criticality of the application

The system is not employed in life-critical applications.

2.4.6 Safety and security considerations

The locations of the passenger and its destinations must be kept private unless the passenger chooses to share rides.

Only taxi drivers with a valid license must be able to use the service for security reasons.

2.5 Assumptions and dependencies

We assume that:

- the taxi is provided with a GPS navigator
- the taxi driver is able to reach the meeting point within 10 minutes from the given hour 90% of the times
- the taxi driver is able to reach the meeting point within 20 minutes from the given hour 100% of the times
- the passenger waits in the location until the taxi arrives
- the taxi driver picks up the correct passenger
- the taxi driver correctly updates his status (off shift, available, busy)
- the passenger specifies the correct location

2.6 Future extensions

The system will be implemented in order to offer the possibility of further extensions:

- Provide secure and reliable methods for ride payments. These functionalities will give the users the opportunity to pay for a taxi service using credit card informations stored in a secure way in their profiles.
- Offer a taxi rating system which allows the users to evaluate the specific taxi service provided.

- Monitor user reliability, recording when they do not show up at the meeting point or how many minutes later they arrive, in order to limit their access to the service when their reliability is bad.
- Create a fidelity score for every user, which gives the opportunity to benefit from special offers. The score increases every time the user makes use of the taxi service, proportionally with the distance covered.

Chapter 3

Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

3.1.2 Hardware interfaces

3.1.3 Software interfaces

3.1.4 Communications interfaces

3.2 System features

3.2.1 User registration

Purpose

Any user car subscribe through the web application or the mobile one.

In both cases user has to fill a registration form and must agree the personal data treatment according his/her country privacy policy, otherwise the registration request is denied.

The system registers the user: As soon as he/she has submitted the privacy policy agreement, the system verifies the consistence of the informations submitted and a confirmation mail is sent to verify the availability of the user mail-address. After this last check the registration ends successfully.

The system has two kind of registration forms (so two kind of registered users) one for passenger and one for taxi drivers.

Passenger registration form: the system registers the user as a passenger; once the user has subscribed the system allows him or her to rent a taxi. The form requires the following informations:

- mail-address
- username
- password
- Name
- Surname
- address (optional filling)
- phone number (optional filling)

Taxi driver registration form: the taxi driver registration is more restrictive and requires more user's informations.

The system registers a user that claims to be a taxi driver only if he is able to demonstrate it with a taxi license; also a medical certificate that ensures the good health of the driver is required. The form requires the following informations:

- mail-address
- username
- password
- Name
- Surname
- address
- phone number
- medical certificate
- taxi license
- taxi id-number
- taxi number-plate

Scenario 1

Alice, a normal citizen without a car, has just discovered the existence of myTaxiService web application and she wants to use it. She opens the home of myTaxiService page on web and clicks on "passenger registration".

She gives all the informations required and she authorizes the personal data treatment.

The system verifies the informations submitted and sends a confirmation mail. So Alice checks the mailbox, opens the mail and clicks on the button "confirm e-mail".

The system informs Alice that the registration has ended with success.

Scenario 2

Bob is a taxi driver that wants to subscribe to myTaxiService application. He downloads the mobile application from his phone app-store and once he opens it, he selects "taxi driver registration". He fills the form, luckily he has gone to the doctor some days before and has a certificate, and authorizes the personal data treatment. However he forgot to write his phone number on the form so the system warns him about the forgetfulness. Only after the complete and correct filling of the form, and the personal data treatment authorization, the registration is one step near the successfully end.

The system verifies, one more time, the informations submitted and sends a confirmation mail. So Bob checks the mailbox, opens the mail and clicks on the button "confirm e-mail".

The system informs Bob that the registration has ended with success.

Associated functional requirements

1. The system must verify the consistence of the user's Name, Surname and nickname. There mustn't be another one already subscribed with the same credentials.

2. The system accepts a password that contains al least one number and one capital letter and that has a minimum length of eight characters.

3. If the personal data treatment isn't authorized the subscription is canceled.

4. The system must abort the registration at each instant if it is user's wish.

5. The subscription isn't ended successfully until the e-mail isn't confirmed. After two days without answer the systems sends anther confirmation e-mail. After seven days user's registration info are deleted.

3.2.2 User login

3.2.3 Standard taxi call

Purpose

Any subscribed passenger shall be able to request a taxi either through the web application or the mobile app. After the request, the passenger is informed by the system about the waiting time and the code of the incoming taxi.

Requests shall be forwarded to available and active taxi drivers in the same zone of the passengers. Taxi drivers shall be able to accept or reject an incoming request.

Scenario 1

Alice needs a taxi. She opens the myTaxiService mobile app on her phone, and selects "Call a taxi". She authorizes the application to access her GPS data, checks on the map that her position is correct and confirms the request.

The system forwards the request to Bob, the first taxi driver in Alice's taxi zone. Bob decides to accept the call: a map of Alice's position gets displayed on Bob's phone and the navigator starts.

The position of Bob is transmitted from Bob's phone to the system, which computes the ETA for the incoming taxi and shows it to Alice. Bob arrives and picks Alice up. Then he confirms that the passenger is on board.

When the ride is over, Bob taps on "Finish ride" so that the system knows that Bob is ready for another ride.

Scenario 2

Luke needs a taxi. He requests it and the request is forwarded to the first taxi driver in queue, Chewbacca. Chewbacca decides to reject the incoming request, so the system puts him at the bottom of the queue.

Luke's request gets forwarded to the new first taxi driver in queue, Han Solo. Han accepts the request and comes to pick up Luke.

Response sequence

The response sequence is illustrated in figure 3.2.3.

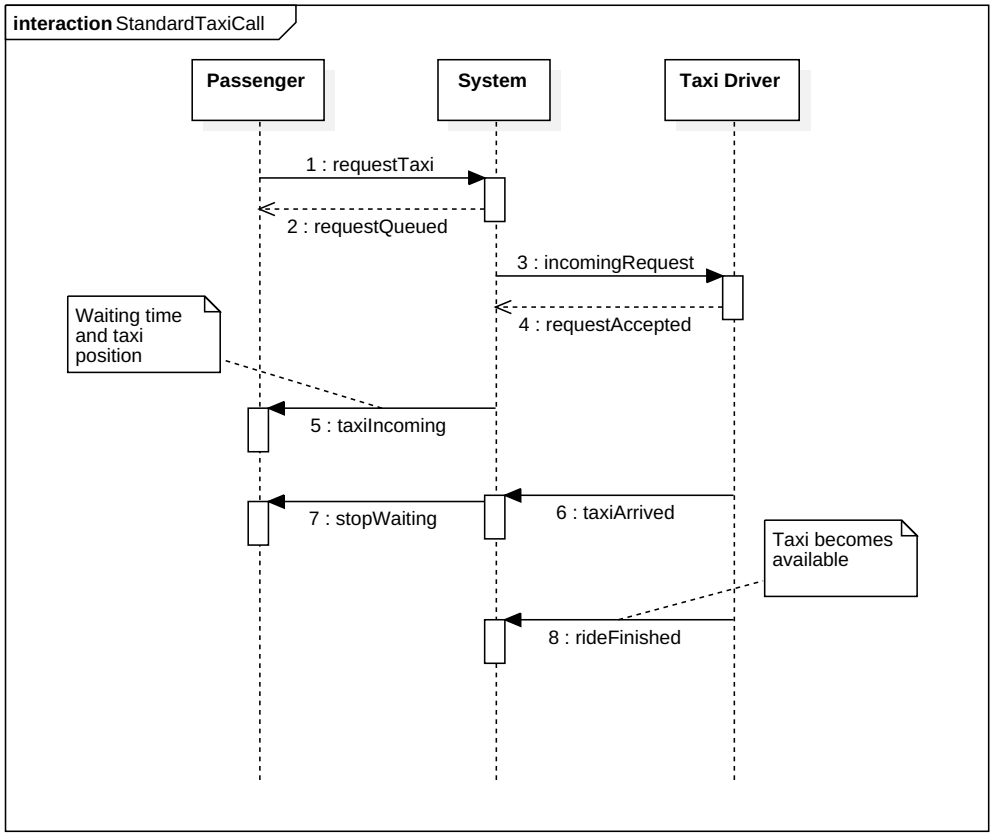

Figure 3.1: Sequence diagram of a successful taxi call.

Associated functional requirements

1. The system must localize the passenger before he or she makes a taxi request.

(a) (App) If GPS info is available and the passenger can be tracked within a radius of 50 m, then the passenger is presented with the option of using the current GPS position.

(b) (App) If GPS info is not available or the precision is less than 50 m, then the app requests the passenger to insert a valid address. Then the address is shown on a map and the passenger can confirm its position.

(c) (Web) In the web application the user is always requested to insert a valid address. Then the address is shown on a map and the user can confirm its position.

2. When the system knows the passenger's position, it presents the passenger the option to request a taxi.

3. The system must ask the passenger for confirmation before delivering the taxi request.

4. After the passenger's confirmation of a taxi request, the request is delivered to the first taxi in the queue for the taxi zone in which the user is located.

5. Taxi requests are forwarded only to active taxi drivers which are located in the same taxi zone, who are not currently busy.

6. Taxi requests are processed in order of arrival.

7. When a taxi driver receives a request, the mobile application shows a notification and emits a sound.

8. When receiving a request, the taxi driver is presented with the possibility of accepting or denying it.

9. After having accepted a request, the taxi driver is automatically marked as "busy".

10. After having accepted a request, the mobile application shows to the taxi driver a map with the location of the passenger, and automatically starts navigation instructions on the app preferred by the taxi driver (Google Maps, Waze, TomTom, etc.).

11. After the taxi request is accepted, the app shows the passenger a map with the current position of the incoming taxi.

12. After the taxi request is accepted, the passenger is informed about the ETA for the incoming taxi.

13. The ETA for the incoming taxi is fetched from the server every 90 s.

14. The ETA for the incoming taxi is computed by the system considering the distance from the taxi to the passenger and the current traffic conditions.

15. When the taxi is within 20 m from the passenger, the app asks the taxi driver to confirm that the passenger is aboard.

16. When the taxi driver confirms that the passenger is aboard, the taxi request is marked as fulfilled and the system stops to show the waiting time to the passenger.

17. The taxi driver is presented by the mobile application with the option to end the ride.

18. When the taxi driver ends the ride, he or she is marked as available and gets re-inserted in the taxi queue of the taxi zone where he or she is located.

3.2.4 Ride request notification to the taxi driver

Purpose

After a taxi driver has informed the system about his/her availability, he/she will be able to receive ride request notifications. The taxi driver receives the notification on his/her cell phone and has one minute to accept or reject the request. If after one minute the request is not answered, the system will consider the request as refused.

Scenario 1

Travis, who suffers from chronic insomnia, every night waits in his taxi for an incoming request. When a user requests a taxi, the system processes the request and decides which taxi driver to send.

For a new incoming request, the system chooses Travis, who is on the top of the queue and quite near the location. It sends him a notification on his mobile app. Travis reads the message which reports location and hour of the meeting and decides to accept the request pushing the appropriate button on the display.

The system acknowledges the decision and waits for Travis to notify that the passenger is on board (last part of 3.2.3).

Scenario 2

The system receives a new incoming request from a user and selects Frank Martin as designated taxi driver. The system sends him a notification and waits for the reply.

Frank has done a very demanding ride and decides to refuse the request using the interface of the app. The system receives Frank's decision and looks for another available taxi on the defined area.

Response sequence

The use case associated to the response sequence is shown in figure.

Associated functional requirements

1. The taxi has to be in the system queue.
2. The GPS on the taxi is available.
3. The GPS position of the taxi must be in the same area of the user's.
4. The app must show the request to the taxi driver.
5. The taxi driver can accept or decline the request using the app.
6. The app must produce sounds or visual notifications as the taxi driver has chosen.
7. The system must use timeouts in order to prevent infinite pending request.
8. The system has to provide the taxi driver with the meeting information.

9. The taxi driver has to reply in one minute or the request is automatically refused.

3.2.5 Taxi availability handling

Purpose

When a taxi driver is ready for a ride, he shall be albe to notify the system using his mobile phone. With the app he can change his status from "not in service" to "in service".

When the system receives the status update from the taxi driver, it inserts him/her in the right queue using the information from the taxi GPS if the new status is "in service", or it removes him/her from the queue if the new status is "not in service".

After a ride, a taxi driver has to notify, using the dedicated section of the app, that the ride is over. The system shall reinsert the taxi into the right queue.

The queues are FIFO and when a taxi driver refuses a ride, the system moves the taxi to the bottom of the right queue.

When a taxi driver accepts a ride and notifies it, the system marks the taxi as busy and removes it from the top of the queue.

Scenario 1

Ernie gets in his taxi, ready to start his working day. He takes out his phone from his pocket and after logging in he changes his status in "in service".

The system acquires the changing and retrieves the GPS position of Ernie's taxi. It analyses the data and sets the taxi in the right queue.

After thirty minutes the system has an incoming request from the Ernie's zone. The first taxi on the top of the queue is Ernie's. The system sends a notification to Ernie and waits for a reply.

Ernie receives the request and accepts the meeting. The system pops Ernie from the top of the queue and marks it as busy.

When Ernie accomplishes the ride, he notifies the system, which retrieves the new GPS position and inserts Ernie's in the right queue.

Response sequence

The response sequence associated with this functionality is shown in figure.

Associated functional requirements

1. The system knows the informations about the GPS installed on the taxi.
2. The mobile app has to offer to the taxi driver the "change status" function and the "ride complete" function.
3. The system must notify the taxi when a request incomes.
4. The system inserts the taxi in the right queue and changes it if the taxi changes area.
5. The system chooses the taxi on the top of the queue.
6. The taxi driver has to reply in one minute to the request, or the ride is automatically refused.
7. The system manages the queue using the FIFO policy.

3.2.6 Taxi reservation

Purpose

Any subscribed passenger shall be able to reserve a taxi for a ride at a predefined time. The passenger has to specify in advance the origin and the destination of the ride, along with the starting date and time.

Scenario 1

John McClane will need a taxi to get to the airport tomorrow morning. He opens the web application of myTaxiService and decides to book a taxi for 6:00 AM for a ride from his home to the airport. He confirms the request.

The morning after, at 5:50 AM, the first taxi driver in the queue gets McClane's request and accepts it. He comes to pick up McClane and brings him to the airport.

Response sequence

Associated functional requirements

1. The system presents the passenger with the option to reserve a taxi.

2. The system asks the passenger the origin and the destination of the ride.

3. Origin and destination must be valid addresses.

4. If GPS info is present and accurate within 50 m, the passenger can specify "current position" as the destination of the ride.

5. The system asks passenger for the date and time of the ride.

6. The system lets the passenger enter only valid dates and times.

7. The system lets the passenger reserve a taxi from 48 hours to 2 hours before the actual ride time.

8. 10 minutes before the specified arriving time, the system allocates a taxi for the passenger by putting a request in the queue as described in subsection 3.2.3.

9. After the request is accepted, the passenger gets notified with the ETA of the incoming taxi along with its position.

3.2.7 Ride sharing

3.2.8 User profile management

Purpose

Any subscribed user can view or update his profile informations.

The system consider taxi drivers accessible for ride only if they load a new medical certificate every two years. A recall mail is sent to the taxi driver three months before the timeline.

Logged passengers can:

- view user profile
- modify an information
- view the latest taxi request
- view taxi requests list

Logged taxi drivers can:

- view user profile
- modify an information
- view the latest taxi request accepted
- view requests accepted list

Scenario 1

Scenario 2

3.3 Performance requirements

The system must support at least 1000 connected passengers at once, and at least 500 simultaneously active taxi drivers at any given time. 95% of requests shall be processed in less than 5 s; 100% of requests shall be processed in less than 10 s.

There is no limit on the total number of registered users.

3.4 Design constraints

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