

RASD myTaxiService

Marco Giuliano

6 November 2015

Contents

1	Introduction	7
1.1	Purpose	7
1.2	Scope	7
1.2.1	Goals	8
1.3	Definitions, acronyms and abbreviations	8
1.3.1	Definitions	8
1.3.2	Acronyms	9
1.3.3	Abbreviations	9
1.4	References	10
1.5	Overview	10
2	Overall Description	11
2.1	Product perspective	11
2.1.1	System interfaces	11
2.1.2	User interfaces	11
2.1.3	Hardware interfaces	12
2.1.4	Software interfaces	12
2.1.5	Communications interfaces	12
2.1.6	Memory constraints	12
2.2	Product functions	12
2.2.1	City Government	13
2.2.2	Driver	13
2.2.3	User	13
2.3	User characteristics	13
2.4	Constraints	14
2.4.1	Regulatory policies	14
2.4.2	Hardware limitations	14
2.4.3	Interfaces to other applications	14
2.4.4	Parallel operations	14
2.4.5	Higher-order language requirements	14
2.4.6	Signal handshake protocols	14

2.4.7	Reliability requirements	15
2.4.8	Safety and security considerations	15
2.5	Assumptions and dependencies	15
2.5.1	Assumptions	15
2.6	Apportioning of requirements	15
3	Specific Requirements	17
3.1	External interfaces	17
3.1.1	User Interfaces	18
3.1.2	APIs	23
3.1.3	External DBMS	23
3.1.4	Wired Network	23
3.1.5	Mobile Radio Network	24
3.1.6	Software (OS and DBMS)	24
3.2	Functions	24
3.2.1	[G.1] City Government can activate taxis into the system	24
3.2.2	[G.2] Every user in the city area can use the mobile application or the web application to reserve a taxi . .	25
3.2.3	[G.3] Every user, after reserving a taxi, receive an acknowledge by the system communicating the number of the incoming taxi and the ETA	25
3.2.4	[G.4] It is guaranteed a fair distribution of runs	26
3.2.5	[G.5] Drivers have the discretion to accept or refuse a call	26
3.2.6	[G.6] When a registered driver turns on the transponder the system automatically recognize the corresponding taxi as available to the service and inserts it in the corresponding queue based on its position	26
3.2.7	[G.7] When a registered driver turns off the transponder the system removes the corresponding taxi from the queue	27
3.2.8	[G.8] The system maintain a QoS of 90% of users served in max 8 minutes	27
3.2.9	[G.9] In case of congestion of the system a message is prompted to the user	27
3.3	Performance requirements	28
3.4	Logical database requirements	28
3.5	Software system attributes	28
3.5.1	Reliability	28
3.5.2	Availability	28
3.5.3	Security	28

<i>CONTENTS</i>	5
3.5.4 Maintainability	28
3.5.5 Portability	29
4 Appendix	31
4.1 UML Models	31
4.1.1 Use Cases	31
5 Revision	35

Chapter 1

Introduction

1.1 Purpose

This Requirements Analysis and Specification Document (RASD) is intended to describe the *myTaxiService* system and define an agreement on expected functionalities, as specified on the contract, between both the customer and the supplier.

1.2 Scope

The project aims to develop a solution to optimize and manage the taxi service in the city customer. The solution should be composed of different modules, each with a specific function. The main component is the *myTaxi_Server*; this is the heart of the application, it stores data about taxis and customers and it manages the business logic for the correct operativity of the whole system. There should be also different front-end components: 2 for customers (*myTaxi_Web* and *myTaxi_Mobile*), one for the taxi driver (*myTaxi_Driver*) and one for the city administration (*myTaxi_Administration*). *myTaxi_Web* and *myTaxi_Mobile* are used by the customers to reserve a taxi and receive and acknowledge message by the system communicating the estimated time of arrival of the car. *myTaxi_Driver* is the interface module in use of the taxi drivers, it enables them to receive calls by the customers and indication about the location they have to reach to pick the customer up. *myTaxi_Administration* is used by the administrator of the system: it enables them to add or remove taxi drivers and to monitor the state of the system. Overall this solution should increase the level of service provided by the city taxi's system both for customers and drivers: customers would be served more precisely and in shorter time and market share for drivers would

be fairer because of the automated management of the queues.

1.2.1 Goals

Here the list of the goals of the myTaxiService application:

- [G.1] City Government can activate taxis into the system;
- [G.2] Every user in the city area can use the mobile application or the web application to reserve a taxi;
- [G.3] Every user, after reserving a taxi, receive an acknowledge by the system communicating the number of the incoming taxi and the ETA;
- [G.4] It is guaranteed a fair distribution of runs;
- [G.5] Drivers have the discretion to accept or refuse a call;
- [G.6] When a registered driver turns on the transponder the system automatically recognize the corresponding taxi as available to the service and inserts it in the corresponding queue based on its position;
- [G.7] When a registered driver turns off the transponder the system removes the corresponding taxi from the queue;
- [G.8] The system maintain a QoS of 90% of users served in max 8 minutes;
- [G.9] In case of congestion of the system a message is prompted to the user;

1.3 Definitions, acronyms and abbreviations

1.3.1 Definitions

- [User] They are considered the end-users of the myTaxiService application: people who need to reserve a taxi to travel in the city.
- [Authorized Driver] They are the drivers of taxi with a regular license who are authorized to work in the city. They are the only one who can be inputed into the database of the myTaxiService system.
- [Registered Taxi] Cars with a regular license, as stated by the law, that can be used by authorized drivers for the service.

- [City Area] Zone in which the city is divided to better distribute cars and users in order to offer a better service and a fairer distribution of work between taxis.
- [Mobile Application] It's a software to download on a mobile device. In our specific case it's a module of the system-to-be in use to the Users to make reservations of taxis.
- [Transponder] A radio device that transmit an unique ID and a position to the server of the mmyTaxiService.
- [Congestion of the system] A situation in which most of the taxi actually in service are busy.
- [Internet] The global network for data transmission.

1.3.2 Acronyms

- [ETA] Estimated time on Arrival. It represents the estimated time, calculated by the system, that the user should wait until the incoming taxi will reach to pick him up.
- [FIFO] First In First Out. It is a methodology in queue management where the first element entered in a queue is the first element served.
- [RASD] Requirement Analysis and Specification Document. It is the document written to specify the requirements of a system to implement.
- [GUI] Graphic User Interface. It's an interface used by users to interact with a system.
- [LAN] Local Area Network. It's a local and private network for the transmission of data.

1.3.3 Abbreviations

- [App] Abbreviation used to indicate a mobile application.
- [Ack] Abbreviation used to indicate an acknowledge message.
- [Driver] Used in this document to indicate an Authorized Driver
- [Taxi] Used in this document to indicate a Registered Taxi

- [Run] Used in this document to indicate a service of transport made by a Taxi for an User
- [G.*n*] Goal number "n"
- [D.*n*] Domain property number "n"
- [R.*n*] Requirement number "n"

1.4 References

- Specification document **Assignments 1 and 2 (RASD and DD).pdf** for the project *myTaxiService*
- IEEE Std 830-1998 **IEEE Standard For Requirement Specification.pdf**
- National, regional and local laws and regulamentation about public transportation and rent of cars with driver;

1.5 Overview

This RASD document will be organized, as the standard IEEE 830-1998 suggest, in four different parts:

- [Introduction]
- [Overall Description]
- [Specific Requirements]
- [Appendix]

Chapter 2

Overall Description

This chapter will describe how the system to be developed will be integrated with the actual environment, how it will interact with the stakeholders and which are the assumptions to be made before specifying the requirements that it must satisfy.

2.1 Product perspective

The myTaxiService system will be the only technology application in use to the city for the management of the taxi service. It won't need to be integrated with any other existing product.

2.1.1 System interfaces

The system should provide an API that exposes the main services used for reservation (the same used by the Mobile and Web module). This interface will be used by developers to provide in the future further services based on the myTaxiService such as taxi sharing or advanced reservation.

2.1.2 User interfaces

As mentioned before, the system will provide four different kind of user interfaces: two are destined for the Users, one for the drivers and one for the city administration. The two interfaces for Users (web and mobile) must be easy to use: no more than one message at time should be provided on the screen and no more than a double choiche should be asked (all requests should be composed with the "Yes/No" pattern). The taxi's interface should be developed for a 11 inch monitor (the monitor will be integrated on the

dashboard) and the information showed on the screen should be limited in order to keep the use of a big font to avoid the distraction of the drivers. The administrator's interface should contain every information about the current situation of the system such as incoming calls, busy taxis, available taxis, state of the queue of every zone and list of authorized taxis.

2.1.3 Hardware interfaces

The system should provide an interface to the navigation system of the taxis, to help drivers to transfer GPS data of the destination to their own navigation system. The User interface module should work only if authorized to use the GPS sensor of the device where is installed (and, of course, the GPS sensor must be present).

2.1.4 Software interfaces

The system should provide internal interfaces for the different modules that are going to be implemented: every GUI module should be able to communicate with the server using specific services provided by this.

2.1.5 Communications interfaces

The main communication interface is the Internet. Mobile, Web and Driver's module will use it for information exchange with the Server. The Administrator's interface communicates with the server using the LAN of the City Government.

2.1.6 Memory constraints

The system shouldn't need to access the memory of Users devices. The system should need to access the memory of Drivers devices to store data about the next serving User. Data of Users, Drivers and runs should be stored on a permanent memory on the Server for at least 5 years for statistical research.

2.2 Product functions

The system will provide the following functions for each category of end user

2.2.1 City Government

- Add a Taxi to the list of the registered Taxis
- Remove a Taxi from the list of the registered Taxis
- Inquiry the system about the status of the taxi service (number of busy cars, number of free cars, number of waiting customers, number of served customers)

2.2.2 Driver

- Join the queue of the corresponding zone as a registered taxi turning on the transponder
- Leave the queue of the corresponding zone as a registered taxi turning off the transponder
- Receive an incoming call
- Accept an incoming call
- Refuse an incoming call
- Signal that the User is on board and the run has begun
- Signal that the User left the car and the run has ended

2.2.3 User

- Download the app
- Get authenticated into the system
- Reserve a taxi

2.3 User characteristics

The typical users of the system shouldn't necessary have a technology background. The only required competency is a certain familiarity with modern technology, touchscreens and mobile phone / tablet. In addition, drivers should be trained in the use of the *myTaxi_Driver* module and the city government's user should be trained in the use of *myTaxi_Administration* module for their peculiarities.

2.4 Constraints

The functional analysis and the development phases of the project may be subject to the following constraints:

2.4.1 Regulatory policies

All the studies and the implementations should be subject to the national, regional and local laws in the field of public transportation and renting of cars with driver.

2.4.2 Hardware limitations

The following hardware limitations should be considered in the design phase:

- Mobile network availability for the Users and the Drivers;
- Computational capacity in order to allow a real time dispatching of calls under the condition of the maximum number of taxi in service
- Issues retrieving the position with the GPS sensor in use of the Users

2.4.3 Interfaces to other applications

It should be possible to transmit the position of the User to serve from the Driver module to the GPS navigation system installed on the taxi. The analyst should consider an interface compatible to all the navigation system installed on the cars.

2.4.4 Parallel operations

It should be considered, in the analysis phase, that in the peak hours the system could receive an average of 20 calls every minute.

2.4.5 Higher-order language requirements

Consider the use of a common programming language in order to encourage external developers to use APIs and improve with new functions the system.

2.4.6 Signal handshake protocols

Every operation made by a user should receive an answer by the system considering the use of mobile radio network.

2.4.7 Reliability requirements

A taxi should be always driven at the correct destination. User that received and ack shouldn't wait an indefinite time because the driver can't find them. Every run should be traced and monitored by the city administration.

2.4.8 Safety and security considerations

Consider, during the analysis phase, a methodology to make the Driver's module compatible to a safe guide.

2.5 Assumptions and dependencies

2.5.1 Assumptions

The following assumption should be considered before formalizing the requirements:

- When a user reserves a taxi it's sure that he needs it and he will be waiting at the place from where he made the reservation;
- There is always an available route between the taxi and the calling User;
- Taxis and Users are equally distributed between the zones;
- Every time a taxi begins a run with a passenger it will bring the User to the destination required. A run can't be interrupted.
- Coordinates communicated by the User's app are always correct
- If a run ends in a zone different than the origin then the taxi will enter the queue of the new zone;

2.6 Apportioning of requirements

Many possible improvement could be considered for further versions of the product, such as:

- The possibility to cancel a reservation by the User;
- The possibility, for the City Administration, to suggest a Driver to relocate his taxi in a different zone considering the amount of calls received and the length of taxi's queues;

- The possibility, for a Driver, to freeze his position in queue for a certain amount of time in order to take a break avoiding the possibility to jump to the end of the queue;
- The possibility to pay the run with the mobile app at the end of it;
- The possibility to specify the destination of the run at the moment of the reservation in order to improve the route calculation engine and optimize the use of the taxi and the queue;

Chapter 3

Specific Requirements

3.1 External interfaces

To list the External interfaces of the system it should be important to distinguish between interfaces with external stakeholders and interfaces between internal modules. There are 2 kind of interfaces between external stakeholders:

- GUIs;
- APIs;
- External DBMS; description

Interfaces between internal components are the following:

- Wired network;
- Mobile radio network;
- Software (OS and internal DBMS);

3.1.1 User Interfaces

Administrator's Interfaces

Administration Page – Taxis List

Unactive Taxis	Remove	Transponder ID:	Add
Alpha1			
Beta2			
Gamma3			

Zones	Taxi for Zone
Zone1	Alpha2
Zone2	Alpha3
Zone3	Beta2

Messages:

Driver's Interfaces

myTaxi_Driver

Status Transponder:

Date:

Hour:

This interface is shown when the transponder is turned off and the taxi is not in service (not in the queue). This give the possibility to turn on the transponder and join the queue of taxis in service.

The interface is titled "myTaxi_Driver". It contains the following elements:

- Status Transponder:** A text input field containing "ON".
- Zone:** A text input field containing "1".
- Position in Queue:** A text input field containing "5".
- Turn OFF:** A button with a grey gradient.
- Date:** A text input field.
- Hour:** A text input field.

This interface is shown when the driver turns on the transponder. It shows the zone in which the taxi is (depending on the position measured by the GPS sensor) and the position in the corresponding queue. It gives the possibility to turn off the transponder and leave the service.

The interface is titled "myTaxi_Driver". It contains the following elements:

- Status Transponder:** A text input field containing "ON".
- Zone:** A text input field containing "1".
- Position in Queue:** A text input field containing "1".
- Turn OFF:** A button with a grey gradient.
- Date:** A text input field.
- Hour:** A text input field.
- Incoming reservation from:** A text input field containing "+39 333 3333333".
- Countdown:** A text input field containing "30s".
- Address:** A text input field containing "Via M. Rossi, 55".
- Coordinates:** Two text input fields containing "11,000000000" and "13,000000000".
- ACCEPT:** A button with a grey gradient.
- DECLINE:** A button with a grey gradient.

This interface is shown when the taxi receives an incoming call. It signals information about the calling User (ID, address and position) and gives the possibility to accept or decline the call.

myTaxi_Driver

Status Transponder:

Zone:

Position in Queue:

Date:

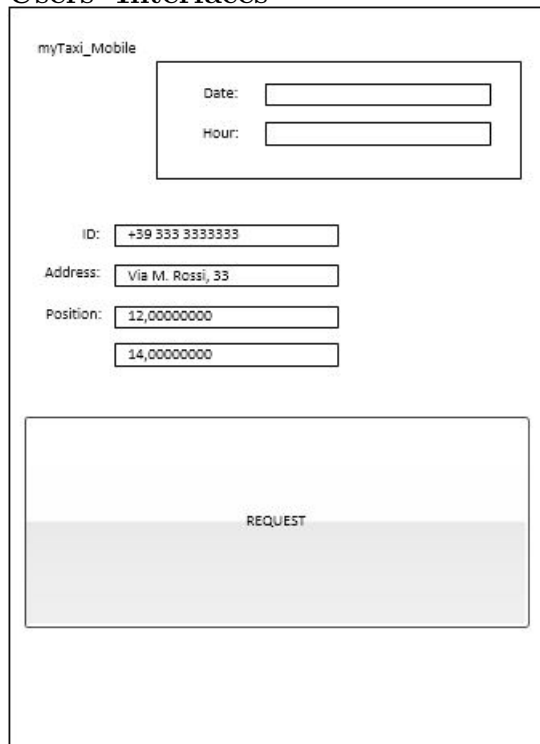
Hour:

Serving user:

Address:

Coordinates:

This interface is shown if the driver accepts the incoming request. It shows information about the serving User, it gives the possibility to send information about the position to the car navigator and the possibility to communicate the end of the run (to rejoin the queue). During the service there's not the possibility to turn off the transponder.

Users' Interfaces

The diagram illustrates the 'myTaxi_Mobile' interface. It features a header 'myTaxi_Mobile' in the top left. Below it, a box contains 'Date:' and 'Hour:' labels, each followed by an empty input field. Further down, the 'ID:' label is followed by an input field containing '+39 333 3333333'. The 'Address:' label is followed by an input field containing 'Via M. Rossi, 33'. The 'Position:' label is followed by two stacked input fields; the top one contains '12,00000000' and the bottom one contains '14,00000000'. At the bottom, a large rectangular area is divided horizontally, with the word 'REQUEST' centered in the upper half.

This is the main interface of the myTaxi-Mobile. When opened it shows information about the logged user and the position. It offers the possibility to reserve a taxi.

myTaxi_Mobile

Date:

Hour:

ID:

Address:

Position:

REQUEST

Error during the request: please try again.

This screen is shown if, after trying to reserve a taxi, an error has occurred.

The screenshot displays the 'myTaxi_Mobile' application interface. At the top, there is a section for reservation details with input fields for 'Date:' and 'Hour:'. Below this, the reservation information is listed: 'ID: +39 333 3333333', 'Address: Via M. Rossi, 33', and 'Position: 12,00000000' and '14,00000000'. A 'Confirmation' box contains the message: 'Your request has been processed, a taxi is approaching. Please maintain your position'. Below the confirmation, the 'Car ID: Alpha1' and 'Estimated Time on Arrival: 4m 30s' are displayed.

This screen is shown after a reservation is made. It shows the ID of the incoming taxi and the estimated time on arrival. At this point the user can close the application.

3.1.2 APIs

The system should expose every function used to reserve a taxi using an API to allow external programmer to develop new functionalities based on the core of the system. Functionalities about the administration panel and the Driver's interface should remain hidden in exclusive use of the city government and his supplier.

3.1.3 External DBMS

The system should be able to connect to the Database, managed by the city, containing the list of all licensed taxis.

3.1.4 Wired Network

The server hosting the application should reside in the LAN of the City Government. The administration module should be connected by this LAN

to the application core. The application server should be connected, by a wired Internet connection, to the three main mobile providers in order to reach every possible user and driver connected by a mobile network.

3.1.5 Mobile Radio Network

User's and Driver's modules should connect to the application core using a mobile radio network. This interface should allow the transmission of small packet (maximum 16kb of data at time). Availability of the network must be guarantee only during the reservation and acceptance phases. After that the system can work "offline".

3.1.6 Software (OS and DBMS)

There are not specific requirements for OS. The system should work on any platform in commerce for mobile devices and on any OS for the application server. The DBMS should be transactional and should manage autonomously read/write concurrency. One of the most common commercial packages should work fine for our purpose.

3.2 Functions

3.2.1 [G.1] City Government can activate taxis into the system

- [D.1] Only authorized users can log into the system with privileges provided to the "City Government";
- [D.2] There is a list in a Database of the City government containing all the taxis with a regular license;
- [R.1] The taxi to be added should have a tablet device where to install the Driver's module and it should be connected to the GPS system and the transponder;
- [R.2] When activated into the system, a taxi should be inserted into the list of authorized Taxis with a specific ID (the transponder ID);
- [R.3] When an administrative user activates a taxi into the system the operation should be logged indicating which user did it, the date, the hour of the operation and the taxi's ID activated;

3.2.2 [G.2] Every user in the city area can use the mobile application or the web application to reserve a taxi

- [D.1] The User has the app installed on his mobile device or opens the web page on his browser;
- [D.2] The app or the website are authorized to retrieve the mobile phone number and to use it as unique ID of the User;
- [D.3] The position of the User is retrieved using the GPS sensor installed on his mobile device;
- [R.1] When a User makes a reservation, a data packet is sent to the system indicating the exact position, the date and the hour of reservation;
- [R.2] In case of absence of radio network the User's module should wait for 30 seconds and try to send another packet;
- [R.3] After 4 try (2 minutes) a message should appear on the User's device to alert that the request has not been sent due to absence of radio signal;
- [R.4] When the request is received by the system an ack message should be sent to the User's device. This message makes the User on hold;

3.2.3 [G.3] Every user, after reserving a taxi, receive an acknowledge by the system communicating the number of the incoming taxi and the ETA

- [D.1] The User doesn't change his position after a reservation request is sent;
- [D.2] The User doesn't close the App or the website on his device until he receives the notification of acceptance of the reservation;
- [R.1] When a request reaches the system the server calculates which taxi should serve the request and the time spent to reach the customer using algorithms based on history;
- [R.2] A message is sent from the server to the User's device communicating the ID of the taxi and the ETA;

- [R.3] Data received by the User's device are visualized on the display and an ack message is sent from the User's device to the server;

3.2.4 [G.4] It is guaranteed a fair distribution of runs

- [D.1] Every taxi has a position in the queue of taxis for a certain zone;
- [R.1] The incoming reservation for a certain zone is assigned to the first taxi in queue for that zone;
- [R.2] The taxi that received the last reservation request jumps to the last position of the queue;

3.2.5 [G.5] Drivers have the discretion to accept or refuse a call

- [D.1] The reservation call is received by the taxi in the first position of the queue;
- [R.1] The Driver's display show the address of the incoming call and a countdown timer indicating 30 seconds;
- [R.2] 2 Big buttons occupies the lower half of the screen indicating "accept" or "refuse";
- [R.4] If the countdown ends until the driver made his choiche, "refuse" is automatically chosen;
- [R.5] The choiche of the driver is sent to the system;

3.2.6 [G.6] When a registered driver turns on the transponder the system automatically recognize the corresponding taxi as available to the service and inserts it in the corresponding queue based on its position

- [D.1] Every taxi has a transponder with an unique ID;
- [R.1] When the driver turns on the transponder a radio signal is sent to the server with the ID and the actual position;
- [R.2] When the server receives the "activation" signal checks that the taxi is registered into the system;

- [R.3] If the taxi is registered is inserted in the last position of the queue of the zone corresponding the actual position of the taxi;
- [R.4] If the ID is not present in the list of registered taxis then is ignored;

3.2.7 [G.7] When a registered driver turns off the transponder the system removes the corresponding taxi from the queue

- [D.1] Every taxi has a transponder with an unique ID;
- [R.1] When the taxi turns off the transponder the ID is sent to the server;
- [R.2] When the server receives the ID of the turning off taxi, the car is removed from the corresponding queue;

3.2.8 [G.8] The system maintain a QoS of 90% of users served in max 8 minutes

- [D.1] The number of taxis is limited;
- [D.2] The number of requests is potentially much larger than the number of taxis
- [R.1] When the number of busy taxis reaches the 90% in the majority of the zones an alert should be sent to the city administration;

3.2.9 [G.9] In case of congestion of the system a message is prompted to the user

- [R.1] The city government has the discretion to send a broadcast message to all the Users informing that a congestion of the service happened;
- [R.2] The city government has the discretion to block any possible new reservation until the QoS is back to an acceptable level;

3.3 Performance requirements

In general, the system should work with real time interaction. The designer should consider an amount of 200 cars, an average number of Users of 3000 each day. Every request made by a User should receive an answer, even if the connection with the server isn't available as previously specified in the requirements. If an exception is raised by the application it should be notified to the users, giving them the possibility to make another reservation.

3.4 Logical database requirements

The system should log every operation made by every function already detailed in this document for a period of 5 year, for statistical purposes;

3.5 Software system attributes

3.5.1 Reliability

The system should allow every time a taxi accepts a reservation to make it reach the corresponding User. The User shouldn't wait for an indefinite time until a taxi serves his request if he receives the ack message;

3.5.2 Availability

The system should be available 99,5% of the time every year. Downtime for maintenance, for the rest of the time, should be always scheduled in the nighttime between 3am and 4am;

3.5.3 Security

Informations on the mobile radio network should be encrypted; It should be impossible to log into the system as a Driver without a recognized transponder ID;

3.5.4 Maintainability

The interaction between modules should be guaranteed by webservices so that every module follows a maintenance and versioning autonomous path;

3.5.5 Portability

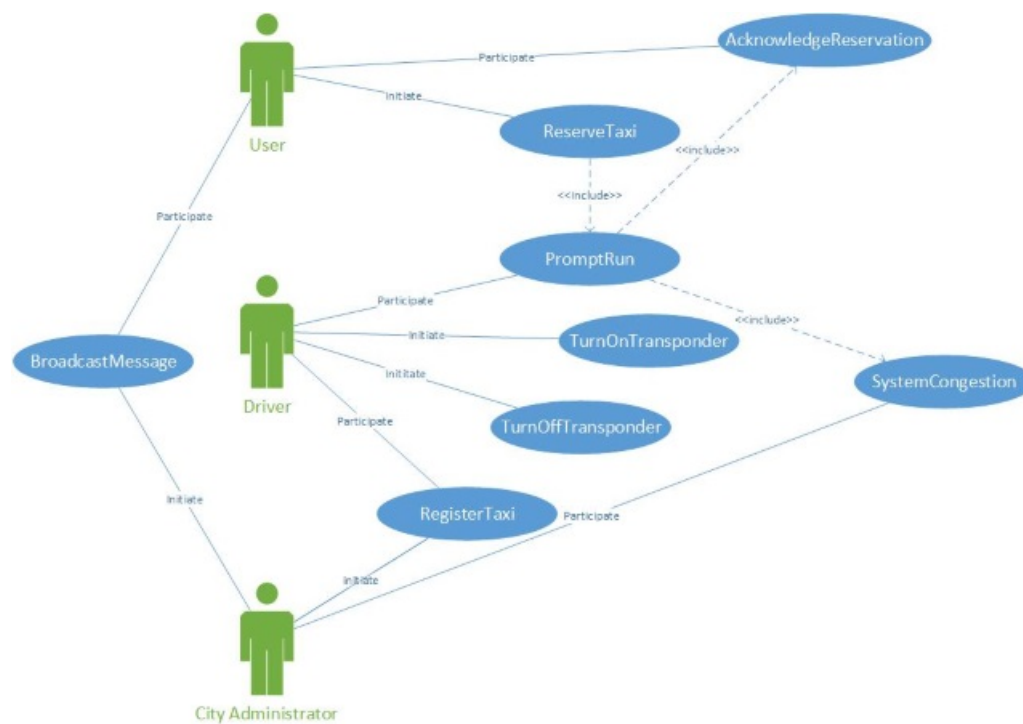
The mobile modules should be multi-platform. They should be developed for iOS, Android and WP8.

Chapter 4

Appendix

4.1 UML Models

4.1.1 Use Cases



RegisterTaxi

- [Actors] City Administrator (initialize), Driver
- [Exceptions] The taxi isn't present in the DB of licensed cars;
- [Flow of Events]
 - 1 The administrator write in the textbox the code of the taxi
 - 2 The systems accepts the code and write a record in the db
- [Special requirements] The taxi should be present in a specific DB

TurnOnTransponder

- [Actors] Driver (initialize)
- [Exceptions] The taxi has not been registered in the system. The taxi is already in service.
- [Flow of Events]
 - 1 The Driver push the button on the main screen of the Driver's module
 - 2 A radio message is sent to the server with an ID and a position.
 - 3 The server checks if the ID is present in the database;
 - 4 The server computes the zone in which the car is and calculates the position in the queue;
 - 5 A message is returned to the taxi with the position in the queue
- [Special requirements] The taxi should be present in the system's db. The taxi should be not in service.

TurnOffTransponder

- [Actors] Driver (initialize)
- [Exceptions] The taxi is not in service
- [Flow of Events]
 - 1 The Driver push the button on the main screen of the Driver's module
 - 2 A radio message is sent to the server with an ID;

- 3 The server checks if the taxi isn't serving any User;
 - 4 The server acknowledge to the taxi;
- [Special requirements] The taxi should be in service. The taxi shouldn't be serving any User.

ReserveTaxi

- [Actors] User (initialize)
- [Exceptions] No mobile radio available. Impossible to retrieve the position
- [Flow of Events]
 - 1 The User opens the app
 - 2 The user push the button to reserve a taxi
 - 3 The system receives the message with the ID and the position of the User
 - 4 The system checks if there is a free taxi in the corresponding zone;
 - 5 The system assign a taxi to the service and executes the "PromptRun" function;
- [Special requirements] Mobile radio should be available. GPS position retrievable.

PromptRun

- [Actors] Driver
- [Exceptions] No mobile radio available.
- [Flow of Events]
 - 1 A message appear on the Driver's device communicating information about a User.
 - 2 The Driver accept or refuse the run
 - 3 The choice is sent to the system
 - 4 If the driver don't make any choice the run is automatically refused after 30 seconds
- [Special requirements] Mobile radio should be available.

AcknowledgeReservation

- [Actors] User
- [Exceptions] No mobile radio available.
- [Flow of Events]
 - 1 The system calculates the route between the accepting taxi and the User
 - 2 The system send a message to the User communicating the incoming taxi and the ETA
 - 3 The message is displayed on the User's device
- [Special requirements] Mobile radio should be available.

SystemCongestion

- [Actors] City Administrator
- [Exceptions] None
- [Flow of Events]
 - 1 The system calculates that less than 10% of taxis are available
 - 2 The system prompt a message on the administrator's interface
- [Special requirements] None

BroadcastMessage

- [Actors] City Administrator (initiate), User
- [Exceptions] No mobile radio available
- [Flow of Events]
 - 1 The city administrator push a button on his interface indicating to broadcast a message
 - 2 The message is broadcasted to all active users indicating that the service is temporary suspended due to a congestion
- [Special requirements] Mobile radio network should be available

Chapter 5

Revision

This is the first revision of the document. All the sections as is are written in the first version.