POLITECNICO DI MILANO

SOFTWARE ENGINEERING II

myTaxiService
REQUIREMENTS ANALYSIS AND SPECIFICATION DOCUMENT

Authors
Giovanni BERI 852984
Biagio FESTA 841988

Friday 6th November, 2015

POLITECNICO DI MILANO

SOFTWARE ENGINEERING II

myTaxiService
REQUIREMENTS ANALYSIS AND SPECIFICATION DOCUMENT

Authors
Giovanni BERI 852984
Biagio FESTA 841988

Thursday 3rd December, 2015

Contents

Contents

myTaxiDriver
He's a registered user for whom all the offered functionalities are needed to accomplish the goal 1 (see subsection Goals at page 4), because they all are complementary to the services provided to passengers. The system allows him:

10. to manage an incoming ride request, by:
    - 10.1 showing him the ride details:
      - 10.1.1 ID of the registered user who sent the request
      - 10.1.2 the GPS coordinates of the picking up place selected by the passenger
      - 10.1.3 estimated time of arrival in the place mentioned above
    - 10.2 letting him choose to accept or decline the incoming request
11. to cancel the confirmation of an already accepted ride. It should avoid the block of the system in situations like a broken engine, an accident or other technical problems that prevent the taxi driver from managing a ride that he has accepted.
12. to signal his state: available or busy.
13. to signal that a ride request he accepted was a *fake request*. The system provides this functionality to the taxi driver only after he has waited 5 minutes for the passenger, in order to avoid false negatives. This functionality is necessary because the system needs to determine which users made the *fake request* and to take measures according to the company policies.
14. to confirm a ride, i.e. signal that he has actually reached the passenger who sent the handled *ride request* or *ride reservation*.

Third Party Developer

15. The system must provide the access to some of its functionalities (basically the *myTaxiUser* ones) using a programmable interface, in order to enable the development of additional services on top of the basic one. The accessible functionalities will be clarified in the next sections.

Administrator User
The system has to provide to a small group of registered users the access to some advanced functionalities, concerning the control of the system itself. So the administrator users will be allowed:

16. to create new *Administrator* account.
17. to visualize informations about position and state of all *myTaxiDrivers*.
18. to activate the *myTaxiDrivers* account.

myTaxiDriver
He's a registered user for whom all the offered functionalities are needed to accomplish the goal 1 (see subsection Goals at page 4), because they all are complementary to the services provided to passengers. The system allows him:

10. to manage an incoming ride request, by:
    - 10.1 showing him the ride details:
      - 10.1.1 the GPS coordinates of the picking up place selected by the passenger
      - 10.1.2 estimated time of arrival in the place mentioned above
    - 10.2 letting him choose to accept or decline the incoming request
11. to cancel the confirmation of an already accepted ride. It should avoid the block of the system in situations like a broken engine, an accident or other technical problems that prevent the taxi driver from managing a ride that he has accepted.
12. to signal his state: available or busy.
13. to signal that a ride request he accepted was a *fake request*. The system provides this functionality to the taxi driver only after he has waited 5 minutes for the passenger, in order to avoid false negatives. This functionality is necessary because the system needs to determine which users made the *fake request* and to take measures according to the company policies.
14. to confirm a ride, i.e. signal that he has actually reached the passenger who sent the handled *ride request* or *ride reservation*.
15. to visualize a map showing the *CityZone* partition of the city, and the distribution of *myTaxiDrivers* in the zones.

Third Party Developer

16. The system must provide the access to some of its functionalities (basically the *myTaxiUser* ones) using a programmable interface, in order to enable the development of additional services on top of the basic one. The accessible functionalities will be clarified in the next sections.

Administrator User
The system has to provide to a small group of registered users the access to some advanced functionalities, concerning the control of the system itself. So the administrator users will be allowed:

17. to create new *Administrator* account.
18. to visualize informations about position and state of all *myTaxiDrivers*.
19. to activate the *myTaxiDrivers* account.

4.3.4 Manage an Incoming Request

Name	Manage an incoming ride request
Actors	myTaxiDriver
Assumptions	• The taxi driver is really available and ready to manage an incoming request.
Pre-Conditions	• The taxi driver is been successfuly authenticated in the system. • The taxi driver is signed as available.
Flow of events	• The taxi driver displays a incoming request through his mobile application running on his phone. • The system provides the taxi driver all information about the incoming request: information about the user who performed the request, and the eventual meeting point. • The system shows to the taxi driver the precise position of the request's starting point, displaying to him a map and an estimation time in order to reach it. • The taxi driver has a limited amount of time to make a decision: the taxi driver can either accept or decline the request. • If the taxi driver accepts the request, he signals if the request is a fake or confirms the ride.
Post-Conditions	• Whether the request has been accepted the taxi becomes not available anymore and the request is solved: a taxi is allocated and he is going to collect the user who made the request. • Whether the request is declined the taxi driver's application will get back a standby status.
Exceptions	• An exceptional case can happen during the taxi driver's decision time, the mobile application cannot communicate anymore for any reasons. In this unlucky case the answer of the taxi driver has to be interpreted as declined.

References: *User requirements* n. 10 (see page 8).

4.3.4 Manage an Incoming Request

Name	Manage an incoming ride request
Actors	myTaxiDriver
Assumptions	• The taxi driver is really available and ready to manage an incoming request.
Pre-Conditions	• The taxi driver is been successfuly authenticated in the system. • The taxi driver is signed as available.
Flow of events	• The taxi driver displays a incoming request through his mobile application running on his phone. • The system provides the taxi driver all information about the incoming request: information about the user who performed the request, and the eventual meeting point. • The system shows to the taxi driver the precise position of the request's starting point, displaying to him a map and an estimation time in order to reach it. • The taxi driver has a limited amount of time to make a decision: the taxi driver can either accept or decline the request. • If the taxi driver accepts the request, he signals if the request is a fake or confirms the ride.
Post-Conditions	• Whether the request has been accepted the taxi becomes not available anymore and the request is solved: a taxi is allocated and he is going to collect the user who made the request. • Whether the request is declined the taxi driver's application will set the taxi driver's state as "*Busy*".
Exceptions	• An exceptional case can happen during the taxi driver's decision time, the mobile application cannot communicate anymore for any reasons. In this unlucky case the answer of the taxi driver has to be interpreted as declined.

References: *User requirements* n. 10 (see page 8).

22

4.3.8 Evaluate a *myTaxiDriver* account activation requests

Name	Evaluate *myTaxiDriver* account activation requests
Actors	Administrator
Pre-Conditions	• Some pending myTaxiDriver account activation requests exist. • The *Administrator* user has already logged in.
Flow of events	• The *Administrator* user visualize the list of all pending requests. • The *Administrator* choose a request to evaluate. • The request data (e.g. licence number) are verified and evaluated, according to the company policy. • If all the required data are evaluated positively, the request is approved; otherwise, the request is rejected. • The operations are repeated until the list is empty.
Post-Conditions	• Every taxi driver who sent an activation request is notified with the evaluation outcome. • All and only the *myTaxiDriver* accounts related to the positively evaluated requests are activated.

References: *User requirements* n. 18 (see page 8). Figure 18.

4.3.8 Evaluate a *myTaxiDriver* account activation requests

Name	Evaluate *myTaxiDriver* account activation requests
Actors	Administrator
Pre-Conditions	• Some pending myTaxiDriver account activation requests exist. • The *Administrator* user has already logged in.
Flow of events	• The *Administrator* user visualize the list of all pending requests. • The *Administrator* choose a request to evaluate. • The request data (e.g. licence number) are verified and evaluated, according to the company policy. • If all the required data are evaluated positively, the request is approved; otherwise, the request is rejected. • The operations are repeated until the list is empty.
Post-Conditions	• Every taxi driver who sent an activation request is notified with the evaluation outcome. • All and only the *myTaxiDriver* accounts related to the positively evaluated requests are activated.

References: *User requirements* n. 19 (see page 8). Figure 19.

27

5 External Interfaces

5.1 Hardware Interface

The mobile application must be authorized to access the device locationing services, either GPS technology and network-based.

5.2 Software Interface

myTaxiService mobile application will use the Google Maps API to provide the map visualization and position insertion services.

5.3 Communication Interface

The mobile application must be authorized to access the internet connection of the device, either Wi-Fi connection or mobile data connection.

5.4 UserInterface

There are few points that are not very clear in the specification document, so we will assume that:

- In the specification document is not clear how the taxi-driver gets the mobile application. For this preliminary document we suggest two different approaches:
  - The system is composed only by *common* applications.

    Both user and taxi-driver mobile applications are stored and maintained in a *digital distribution platform* (such as *Google Play* or *App Store*). It means that everybody can download and use them.

    Although this is not a problem for passenger application (because our purpose is to reach as many people as we can), it could be for the taxi-driver application. Indeed someone could download the mobile application and spoofs others customers passing himself off as licenced taxi-driver.

    In order to avoid this problem, the system shall provide a kind of authentication mechanism. That could be solve by requirement [18] (see 8).

    Moreover this solution could be take into account for a possible selling of the entire system, but a study of this is beyond the scope of this document.
  - The passenger mobile application is stored as the previous point (via common digital market). Instead the taxi-drivers application is distributed through an internal system. *i.e. the city provides every taxi driver a mobile device with the pre-installed application.*
- From the passenger point of view, *myTaxiService* will be accessible from two different interfaces (see figures 11 and 12 for mockups):
  - Mobile Application: the user can access to all functionalities listed in the section 2.1 (see page 7).

5 External Interfaces

5.1 Hardware Interface

The mobile application must be authorized to access the device locationing services, either GPS technology and network-based.

5.2 Software Interface

myTaxiService mobile application will use the Google Maps API to provide the map visualization and position insertion services.

5.3 Communication Interface

The mobile application must be authorized to access the internet connection of the device, either Wi-Fi connection or mobile data connection.

5.4 UserInterface

There are few points that are not very clear in the specification document, so we will assume that:

- In the specification document is not clear how the taxi-driver gets the mobile application. For this preliminary document we suggest two different approaches:
  - The system is composed only by *common* applications.

    Both user and taxi-driver mobile applications are stored and maintained in a *digital distribution platform* (such as *Google Play* or *App Store*). It means that everybody can download and use them.

    Although this is not a problem for passenger application (because our purpose is to reach as many people as we can), it could be for the taxi-driver application. Indeed someone could download the mobile application and spoofs others customers passing himself off as licenced taxi-driver.

    In order to avoid this problem, the system shall provide a kind of authentication mechanism. That could be solve by requirement [19] (see 8).

    Moreover this solution could be take into account for a possible selling of the entire system, but a study of this is beyond the scope of this document.
  - The passenger mobile application is stored as the previous point (via common digital market). Instead the taxi-drivers application is distributed through an internal system. *i.e. the city provides every taxi driver a mobile device with the pre-installed application.*
- From the passenger point of view, *myTaxiService* will be accessible from two different interfaces (see figures 11 and 12 for mockups):
  - Mobile Application: the user can access to all functionalities listed in the section 2.1 (see page 7).

31

– Web Application: not all functionalities are at disposal of the users, because we assume that users access to web applications interfaces using a personal computer, so all informations related to the position detection are not useful. For that reason we decided to allow the user to make only *ride reservation* and deny to make a *simple ride request*.

5.4.1 Mockup User Interface

myTaxiUser Graphic Interface

(a) *Pick Me Up Function.*

(b) *Manual Starting Point Address Insertion.*

(c) *Make a Reservation.*

(d) *Waiting For Taxi Arrival.*

Figure 11: Some mockups user interface for passenger user.

– Web Application: not all functionalities are at disposal of the users, because we assume that users access to web applications interfaces using a personal computer, so all informations related to the position detection are not useful. For that reason we decided to allow the user to make only *ride reservation* and deny to make a *simple ride request*.

- The system will be accessible only via web browser interface. Indeed the administrator's work should be made using a personal computer and the design of another mobile application would have been an overkill.