Politecnico di Milano
A.A. 2015-2016
Software Engineering 2: "TAXInseconds"
Requirements **A**nalysis
and
Specifications **D**ocument

Roberto Clapis, Erica Stella

October 28, 2015

Contents

1 Introduction **3**
- 1.1 Purpose . . . 3
- 1.2 Actual System . . . 3
- 1.3 Scope . . . 3
- 1.4 Goals . . . 4
- 1.5 Definition and Acronyms . . . 4
  - 1.5.1 Definitions . . . 4
  - 1.5.2 Acronyms . . . 4
- 1.6 Actors . . . 5
- 1.7 References . . . 5
- 1.8 Overview . . . 5

2 Overall description **5**
- 2.1 Product perspective . . . 5
  - 2.1.1 System interfaces . . . 6
  - 2.1.2 User Interfaces . . . 6
  - 2.1.3 Hardware Interfaces . . . 6
  - 2.1.4 Software Interfaces . . . 6
  - 2.1.5 Communication Interfaces . . . 6
  - 2.1.6 Operations . . . 6
- 2.2 Product functions . . . 6
- 2.3 User characteristics . . . 6
- 2.4 Constraints . . . 6
- 2.5 Assumptions and dependencies . . . 6
  - 2.5.1 Domain Assumptions . . . 6

3 Specific requirements **7**
- 3.1 Functional Requirements . . . 7
- 3.2 Non functional requirements . . . 8

1 Introduction

1.1 Purpose

This document represent the Requirement Analysis and Specification Document (RASD). It aims at explaining the domain of the system to be developed and the system itself in terms of functional requirements, nonfunctional requirements and constraints. It also provides several models of the system and typical use cases. It is intended for all the developers who will have to implement the system, the testers who will have to determine if the requirements have been met and the system analysts who will have to write specifications for other system that will relate to this one. It is also intended as a contractual basis thus being legally binding.

1.2 Actual System

The government of the city wants to optimize its taxi service with a completely new application. Therefore, we assume there are no previous systems to take into account.

1.3 Scope

The aim of the project TAXINSECONDS is to provide a new application to optimize the taxi service of the city.
The application will be available to the users in web and mobile forms and will have public APIs in order to expand and improve the service with additional features.
The city managed by TAXINSECONDS is divided in zones of 2 km each and every zone has its own queue of taxis. The queues are automatically computed by the system with the information it receives from the GPS of the taxis. Taxi drivers can be available or not. Only available taxi drivers can be in a queue. When a taxi driver changes her state from not available to available the system automatically adds her to the queue of the zone she is currently in, based on the information of the GPS of the taxi.
Users that are not registered can only see the estimated time of arrival of the nearest taxi with TAXINSECONDS.
Registered users can also request a taxi or make a reservation for a taxi. Reservations can only be made at least two hours before the ride and must be done specifying the starting location, the destination and the meeting time. Requests, instead, only need the starting location and the destination.
When a request is made, the first taxi driver of the queue of the starting location's zone is notified for accepting or rejecting the request. If a taxi driver rejects a request her state is put on unavailable until her changes it back to available. If a taxi driver doesn't accept or reject a request within 1 minute, the request will be passed on to the next taxi driver in the queue and the first one will be moved to the end of the queue. If there are no available taxis in

the zone of the request the system will propagate the request to the closest not empty queue.
When a request is accepted the registered user that has made the request receives a notification from the system informing her of the code of the incoming taxi and the estimated waiting time.
When a reservation is made, the system confirms it to the user and allocates a taxi 10 minutes before the meeting time. If a taxi for that zone is not available the request is then handled by finding the closest available taxi.
If a taxi does not reach the client's location in an amount of time determined by the system using traffic and position information the call is by default cancelled without penalties and both parts are notified.

1.4 Goals

- Provide an easy way to request a taxi.
- Provide an easy way to reserve a taxi.
- Guarantee a fair management of the taxis queue.

1.5 Definition and Acronyms

1.5.1 Definitions

- registered user
- guest
- administrator
- request
- reservation
- state of a taxi driver
- meeting time
- nearest taxi

1.5.2 Acronyms

- ETA: estimated time of arrival: the time, estimated by the system, that the nearest taxi to the location of the call will take to get to the client's position.

1.6 Actors

- Taxi drivers: after successfully logging in, taxi drivers are able to set their current state as available or not and to accept or refuse requests.
- Users: user are able to login or to ask the system for an ETA.
- Registered users: after successfully logging in, registered users can request or reserve taxis.

1.7 References

1.8 Overview

This document is structured in three parts:

- Introduction: gives an high-level description of the software purposes and context.
- Overall Description: gives a general description of the application, focusing on the context of the system, going in details about domain assumptions and constraint. The aim of this section is to provide a context to the whole project and show its integration with the real world.
- Specific Requirements: this section contains all of the software requirements to a level of detail aimed to be enough to design a system to satisfy the said requirements, and testers to test that the system actually satisfies them. It also contains the detailed description of the possible interactions between the system and the world with a simulation and preview of the expected response of the system with given stimulation. (Details are given with Alloy specifications and UML diagrams)

2 Overall description

2.1 Product perspective

The TAXINSECONDS application will be released as a web application and as a mobile application. There are no existing systems to integrate it with.
It will provide a total of 4 main interfaces:

- for the users in both the
  - registered user mode
  - guest mode
- for taxi drivers
- for administrators
- a lower level interface for APIs

2.1.1 System interfaces

2.1.2 User Interfaces

2.1.3 Hardware Interfaces

DOPO NEI REQUIREMENTS

2.1.4 Software Interfaces

2.1.5 Communication Interfaces

2.1.6 Operations

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions and dependencies

2.5.1 Domain Assumptions

- All taxi drivers who intend to use the service will have a mobile phone with one of the supported mobile OSs
- All taxi drivers will have a phone with GPS functionality
- The taxi drivers will have to grant the system the rights to handle their taxi codes
- Taxi drivers' phones will have an internet connection
- Clients will have access to the internet and a device with GPS integrated
- Clients have a valid email address
- Clients have a credit/debt card
- Clients allow the app to access their credit in order to pay for the service
- When accessing from the website the clients will have to grant permission to the application to read the device's position
-

3 Specific requirements

3.1 Functional Requirements

On the user side:

- For the non logged in user (on both the WEB and Mobile interface):
  - Provide information about taxis queues and availability
  - Registration in order to become a registered user
  - Login for registered users
  - "change personal data" procedure for registered users
- For the logged in user (on both the WEB and Mobile interface):
  - Provide information about taxis queues and availability
  - Place a call for a taxi at the current position
  - Make a reservation for later
  - Logout
- Through the API:
  - Get the queue status for a given position
  - Get the ETA for a taxi for a given position
  - Establish a Secure Channel
  - Obtain a login-token using valid credentials through a previously established Secure Channel
  - Using a valid login-token:
    - Place a call for a given position
    - Place a reservation for a given position
    - Require the system to send a push message for updates about the status of a previous request
    - Logout (this will close the secure channel and invalidate the login token)

On the taxi driver side (there will be no Web interface in this case):

- For the non logged in driver:
  - Login for a registered driver
  - "Reset password" procedure for registered drivers
- For the logged in driver:
  - If credentials have been invalidated a "reset password" procedure will be mandatory

  - Toggle the availability of the driver
  - Notify for new calls
  - Allow to accept or refuse a call
  - Handle disconnection
  - Handle timeout (if reaching the client takes too much time)
  - Log out
- Through the API:
  - Establish a Secure Channel
  - Obtain a login-token using valid credentials through a previously established Secure Channel
  - Using a valid login-token:
    * Toggle the driver state
    * Send a notification when a call is made for the driver
    * Accept the answer to the call, whether the answer is Acceptance or Refusal
    * Handle disconnection
    * Handle timeout (if no taxi reach the client in time)
    * Log out (invalidation of the token)
  -

On the admin side:

- For the non logged in admin:
  - Log in
- For the logged in admin:
  - Add a new taxi driver
  - Remove a taxi driver
  - Invalidate a taxi driver's credentials
  - Set the area of the map handled by the system
  - Change credentials for the admin
  - Log out

3.2 Non functional requirements

- *Cross platform*
  - There will be a UI for mobile and one for the Web platform
  - At least Android¿=4.0 and iOS will be supported for the mobile version

  - At least Edge, Chrome and Firefox will ahve to be supported
  - The web application will have to use HTML5
- *Responsiveness and availability*
  - The application must always respond, even if no taxi are available
  - If a query is taking some time a spinner will be shown, the app must never freeze
- *Usability*
  - The UIs have to be user friendly and with a few, clear functionalities
- *Security*
  - In no situation sensible data will pass through an insecure channel
  -
- The service must be always