

Requirements Analysis and Specification Document: myTaxiService

Chitti Eleonora, De Nicolao Pietro, Delbono Alex

October 22, 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
2.2	Product functions	5
2.3	User characteristics	5
2.4	Constraints	5
2.5	Assumptions and dependencies	5
3	Specific requirements	6
3.1	External interface requirements	6
3.1.1	User interfaces	6
3.1.2	Hardware interfaces	6
3.1.3	Software interfaces	6
3.1.4	Communications interfaces	6
3.2	System features	6
3.3	Performance requirements	6
3.4	Design constraints	6
3.5	Software system attributes	6
3.6	Other requirements	6

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 or 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, comprehensive 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.

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

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions and dependencies

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

3.3 Performance requirements

3.4 Design constraints

3.5 Software system attributes

3.6 Other requirements