

myTaxiService

Design Document

Belluschi Marco 791878, Cerri Stefano 849945, Di Febbo Francesco 852389

November 25, 2015

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Definitions, Acronyms, Abbreviations	3
1.4	Reference Documents	3
1.5	Document Structure	4
2	Architectural Design	5
2.1	Overview	5
2.2	High level components and their interaction	5
2.3	Component view	6
2.4	Deployment view	6
2.5	Runtime view	6
2.6	Component interfaces	8
2.7	Selected architectural styles and patterns	8
2.8	Other design decisions	8
3	Algorithm Design	9
4	User Interface Design	10
4.1	Purpose	10
4.2	UX diagram	10
4.3	Mockups	12
5	Requirements Traceability	17
6	References	18
A	Appendix	19
A.1	Software and tool used	19
A.2	Working hours	19

Chapter 1

Introduction

1.1 Purpose

This document contains the complete design description of myTaxiService. This includes the architectural features of the system down through details of what operations each code module will perform and the database layout. It also shows how the use cases detailed in the RASD will be implemented in the system using this design. The primary audiences of this document are the software developers.

1.2 Scope

MyTaxiService is a taxi service for a large city. The main goals of the system are:

- simplify the access of passengers to the service
- guarantee a fair management of taxi queues

The system architecture will be a three-tier architecture: client, server application and server database. It will be created by using the MVC architectural pattern.

The system will be divided into components with respect to the principles leading to good design:

- Each individual component will be smaller in order to be easier to understand
- Coupling will be reduce where possible
- Reusability and flexibility will be increase in order to make easier future implementation

The system will have efficient algorithm in order to increase its performance; in the document will be given special attention to the sharing algorithm.

1.3 Definitions, Acronyms, Abbreviations

Definitions

- User: person that uses the service applications
- Visitor: user that has not registered nor logged in
- Registered user: user that has registered to the service
- Passenger: passenger registered to the service
- Taxi driver: taxi driver registered to the service
- System: the union of software and hardware to be developed and implemented

Acronyms

- RASD: requirements analysis and specification document
- AES: Advanced Encryption Standard
- FIFO: First In First Out
- ETA: estimated time of arrival
- API: application programming interface
- GPS: Global Positioning System
- MVC: Model View Controller
- UX: User Experience

1.4 Reference Documents

- Software Engineering 2 Project AA 2015/2016: Project Description And Rules
- Software Engineering 2 Project AA 2015/2016: Assignments 1 and 2 (RASD and DD)
- Software Engineering 2 Project AA 2015/2016: Structure of the design document

1.5 Document Structure

This document is essentially structured in seven parts:

- Introduction: it gives a description of the document and some basic information about the system design and architecture.
- Architectural Design: This is the core of the document. It gives general information about the architectural design. It also describes how the system will be divided into components and how the components communicate. It also has a description of the design pattern and architectural styles that will be used.
- Algorithm Design: it gives a description of the main algorithm that will be implemented. More focus will be given in the sharing algorithm.
- User Interface Design: it gives a description of the user interfaces of the system and the flow from one interface to another.
- Requirements Traceability: this section documents the life of a requirement and provides bi-directional traceability between various associated requirements.
- References: it gives information on the guidelines used in order to redact this document.
- Appendix: it provides information that are not considered part of the actual DD. It includes: software and tools used, project group organization.

Chapter 2

Architectural Design

2.1 Overview

This chapter describes the software system, the relationships between software components and the relationship to actors with the system. Each component is described by a specification and an interface design. The specification is a description of its purpose, its functionality, its attributes (including dependency on other components) and the constraints under which it must operate. It also describes resources, that is, any elements used by the component which are external to the design such as physical devices and software services. The interface design is the list of the services that it provides to clients. These services are methods (procedures and functions), each carefully documented.

Each component in turn may provide its services by having an internal architectural design with its own set of subordinate components. These components may be called sub-components. The decomposition of a higher-level component into subordinate component must be explicit. The algorithm that shows how each method of the larger component is performed by these components must be explicit. Any data stored in an component must be explicitly described.

2.2 High level components and their interaction

DBMS: this component provides access to all of the data contained in the database. It provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized.

TaxiQueue this component manages the queue of available taxis for each city zone.

Request this component manages the incoming requests from passengers and responses from taxi drivers. It manages all types of requests, including standard, shared and reserved rides thanks to the interaction with TaxiQueues component. It takes care of messages dispatching to users.

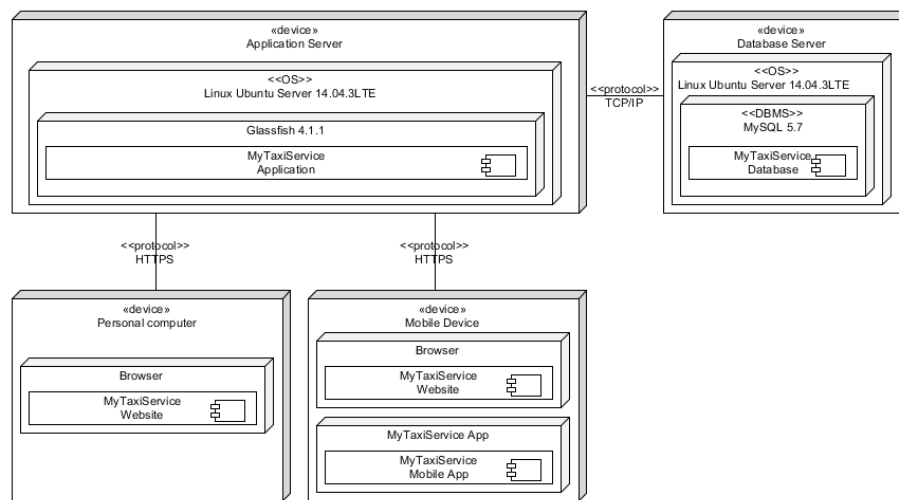
MapServices

UserManager

2.3 Component view

2.4 Deployment view

Here is the deployment diagram for MyTaxiService:



2.5 Runtime view

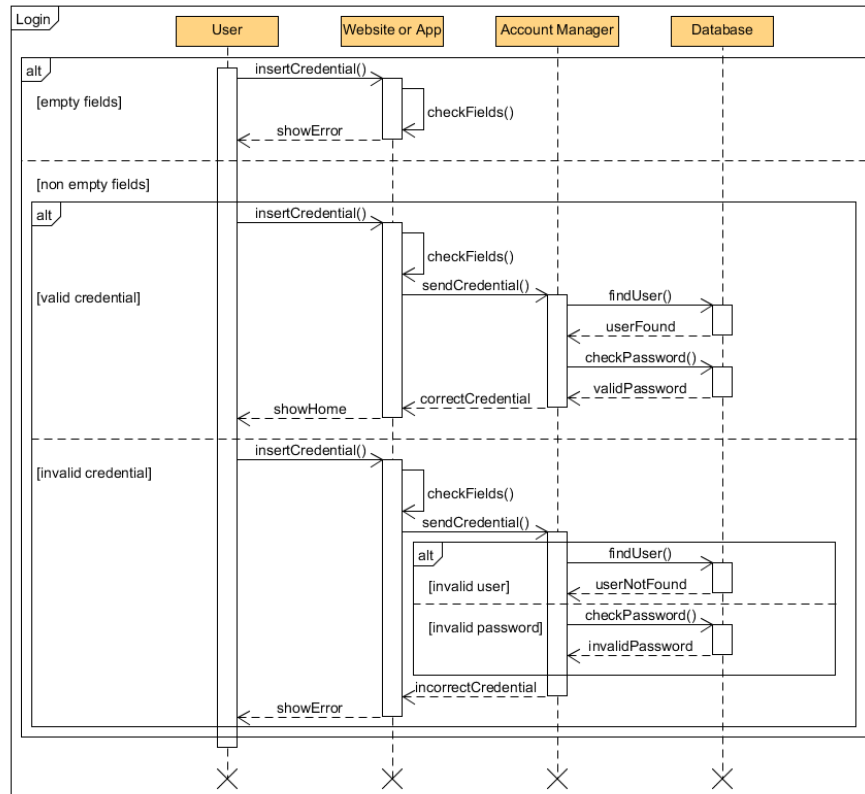
In this section will be presented the interactions between components using some sequence diagram.

Registration

Passenger Registration

Taxi Driver Registration

Login



Find Available Driver

Taxi Request

Taxi Request Client Side

Taxi Request Server Side

Taxi Reserve

Taxi Reserve Client Side

Taxi Reserve Server Side

2.6 Component interfaces

2.7 Selected architectural styles and patterns

2.8 Other design decisions

Chapter 3

Algorithm Design

Chapter 4

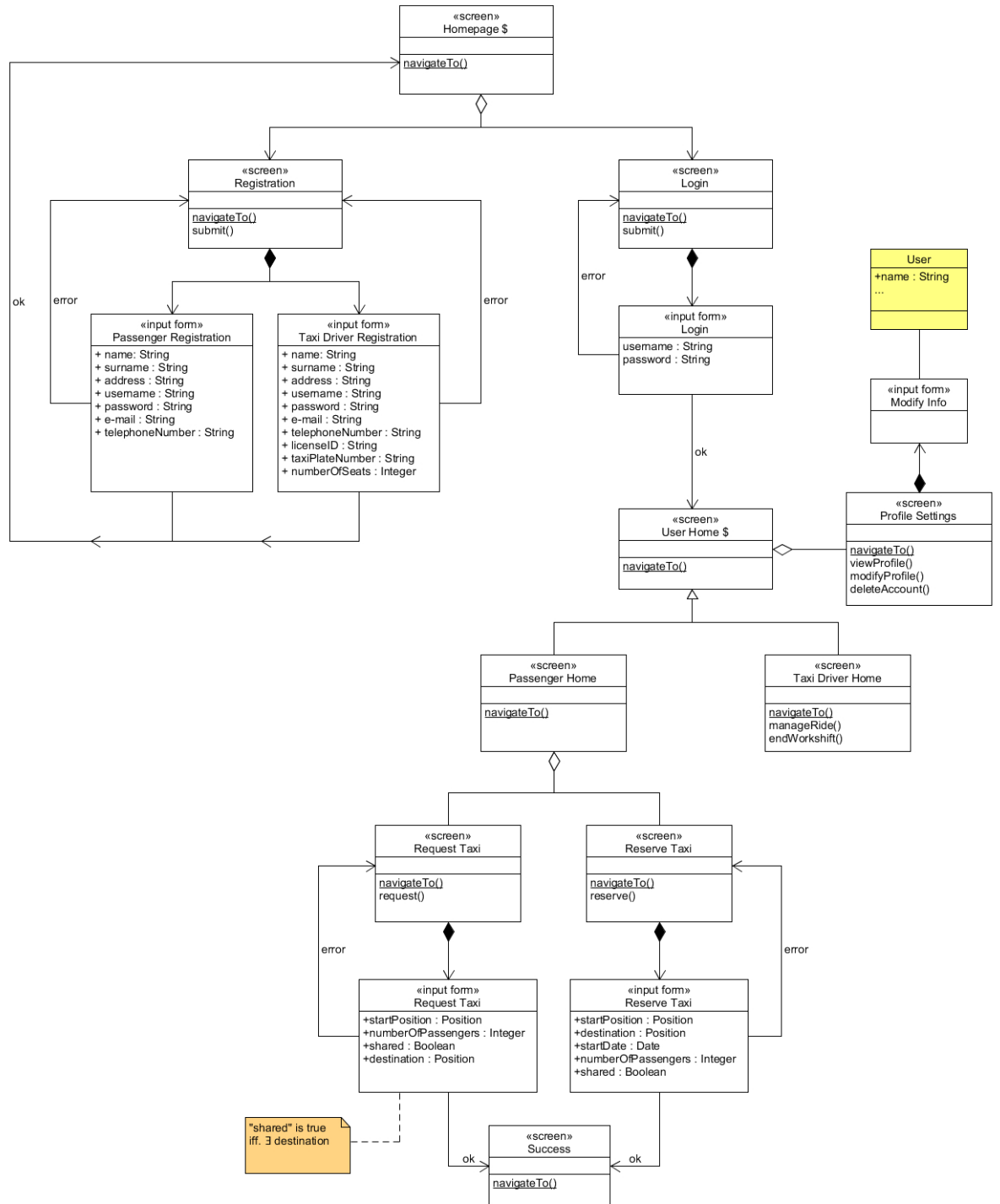
User Interface Design

4.1 Purpose

This chapter gives a description of the user interfaces of the system and the flow from one interface to another. The description includes an UX diagram and some mockups in order to understand how a user can do actions using the interface given by the system.

4.2 UX diagram

Here is the UX diagram:



4.3 Mockups

Here we will present some mockups of MyTaxiService. Some of them referring to the RASD document Section 3.1.1 User Interface.

Log in: In the figure below is shown MyTaxiService's homepage

The desktop mockup shows a web browser window with the URL <http://www.myTaxiService.com>. The page title is "MyTaxiService". The login form includes fields for "username" and "password", a "Log in" button, and a link for "Forgot password?". Below the login form, there are two registration links: "New user? Register now!" with a "Register" button, and "New taxi driver? Register now!" with a "Register" button. The footer contains links for "What's MyTaxiService", "Help", and "FAQ".

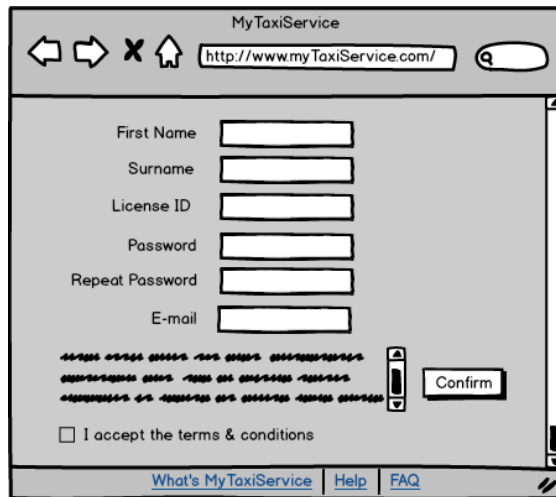
The mobile mockup shows the same login form adapted for a smartphone screen. It includes fields for "username" and "password", a "Log in" button, and a link for "Forgot password?". Below the login form, there are two registration links: "New user? Register now!" with a "Register" button, and "New taxi driver? Register now!" with a "Register" button.

Registration passenger: View of the visitor that wants to register as a passenger

The desktop mockup shows a web browser window with the URL <http://www.myTaxiService.com/>. The page title is "MyTaxiService". The registration form includes fields for "First Name", "Surname", "Username", "Password", "Repeat Password", and "E-mail". Below the form, there is a "Confirm" button and a checkbox for "I accept the terms & conditions". The footer contains links for "What's MyTaxiService", "Help", and "FAQ".

The mobile mockup shows the same registration form adapted for a smartphone screen. It includes fields for "Username", "Password", "Repeat Password", and "E-mail". Below the form, there is a "Confirm" button and a checkbox for "I accept the terms & conditions".

Registration taxi Driver: View of the visitor that wants to register as a taxi driver



MyTaxiService

http://www.myTaxiService.com/

First Name

Surname

License ID

Password

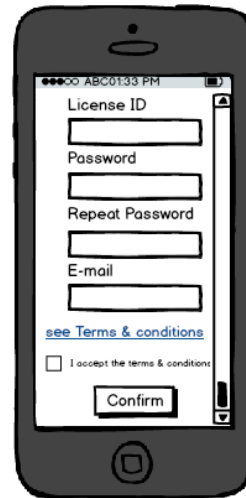
Repeat Password

E-mail

☐ I accept the terms & conditions

[see Terms & conditions](#)

[What's MyTaxiService](#) [Help](#) [FAQ](#)



ABC01:33 PM

License ID

Password

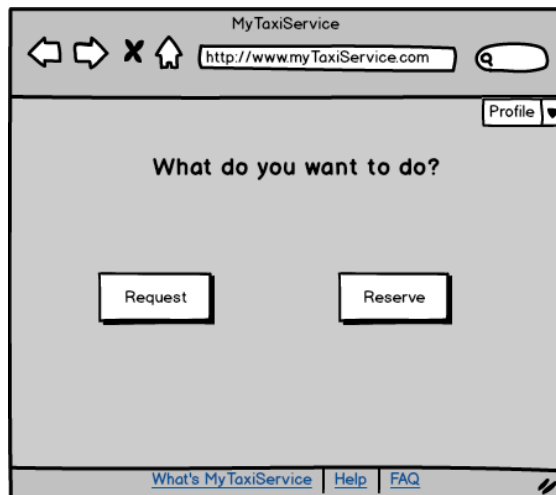
Repeat Password

E-mail

☐ I accept the terms & conditions

[see Terms & conditions](#)

Passenger view: View of the passenger



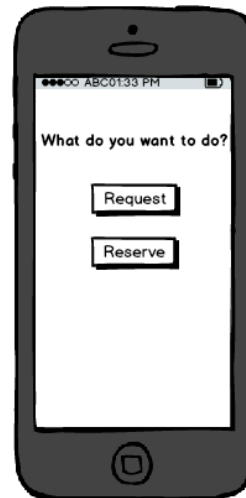
MyTaxiService

http://www.myTaxiService.com/

Profile ▾

What do you want to do?

[What's MyTaxiService](#) [Help](#) [FAQ](#)



ABC01:33 PM

What do you want to do?

Profile: View of the profile of the user

MyTaxiService
http://www.myTaxiService.com/

First Name:
Surname:
username: Heisenberg
Password:
New Password:
Confirm Password:
e-mail:

[What's MyTaxiService](#) [Help](#) [FAQ](#)

ABC11:29 AM

License ID:
Password:
New Password:
Confirm Password:
e-mail:

Request a taxi: View of the passenger when he/she requests a taxi

MyTaxiService
http://www.myTaxiService.com/

Request a taxi Profile

Origin:
☒ Share
Destination:
Number of passenger:

Please remember to wait the taxi in your position

[What's MyTaxiService](#) [Help](#) [FAQ](#)

ABC01:33 PM

Request a taxi

Origin:
☐ Share
Number of passenger:

Please remember to wait the taxi in your position

Reserve a taxi: View of the passenger when he/she reserves a taxi

The image shows two versions of a 'Reserve a taxi' form. On the left is a desktop browser window for 'MyTaxiService' at 'http://www.myTaxiService.com/'. It features a 'Profile' dropdown, a title 'Reserve a taxi', and input fields for 'Origin' and 'Destination', each with a 'use maps' button. Below these are 'Time: hours/minute' (16 and 50) and 'Number of passenger' (1), both with up/down arrows. A 'Share' checkbox is present. A note states 'Remember that you can reserve a taxi 2 hours before the ride'. At the bottom are links for 'What's MyTaxiService', 'Help', and 'FAQ', and a 'Confirm' button. On the right is a smartphone screen showing the same form, adapted for a smaller screen with a status bar at the top showing 'ABC01:34 PM'.

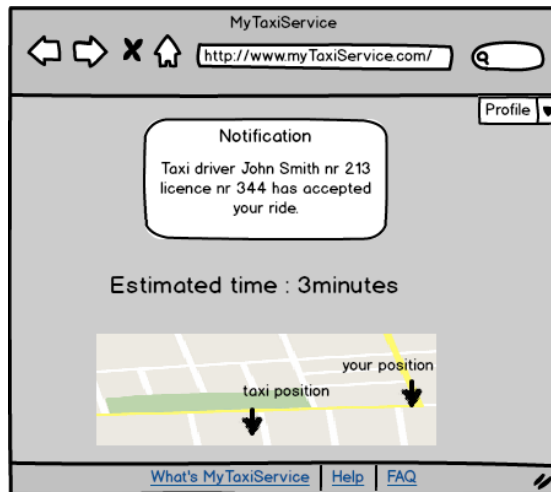
Taxi driver view: View of the taxi driver

The image shows a smartphone screen for a taxi driver. The status bar at the top shows 'ABC01:34 PM'. The main heading is 'What do you want to do?'. Below it, the text 'Your current status is : busy' is displayed. There are two buttons: 'Finish current ride' and 'Finish your workshift'.

Taxi driver notification: Notification that the taxi driver, chosen by the system, sees when a passenger request a ride.



Passenger notification : Notification that the passenger see when a taxi accept the ride



Chapter 5

Requirements Traceability

Chapter 6

References

Appendix A

Appendix

A.1 Software and tool used

- LaTeX (<http://www.latex-project.org/>) : to redact and to format this document
- Balsamiq Mockups (<http://balsamiq.com/products/mockups/>): to create mockups
- Microsoft Office Visio 2013 (<https://products.office.com/it-it/Visio/flowchart-software>): to create sequence diagrams and state-charts
- Eclipse Luna (<https://eclipse.org/luna/>): to draw global use case, class diagrams and UX diagram

A.2 Working hours

This is the time spent for redact the document

- Belluschi Marco : xx hours
- Cerri Stefano : xx hours
- Di Febbo Francesco : xx hours