Politecnico di Milano, A.A. 2015/2016

Software Engineering 2: My Taxi Service **D**esign **D**ocument

Belotti Nicola 793419
Chioso Emanuele 791621
Colombo Andrea 853381

November 19, 2015

Contents

1 Introduction

1.1 Purpose

The goal of this Design Document is to explain the design choice made during the architectural analysis and the functionalities that will be developed.

1.2 Scope

The aim of the Design Document is to present the main architecture of My Taxi Service application. The structure will be based on the requirements analysis we made in the RASD document.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

Visitor Someone who visits my taxi service application's website but it's not logged in

User Someone who is registered and logged in my taxi service application

Passenger A type of logged user, who uses the application to call a taxi

Taxi driver Another type of logged user, who uses the application to answer calls from the system

1.3.2 Acronyms

DBMS Database Management System.

JEE Java Enterprise Edition.

API Application Programming Interface.

ER Entity-Relational Model.

EJB Enterprise Java Bean.

HTML Hypertext Markup Language.

HTTP Hypertext Transfer Protocol.

JDBC Java Database Connectivity.

UML Unified Modeling Language.

UX User eXperience.

MVC Model View Controller.

JPA Java Persistence API.

XHTML eXtensible Hypertext Markup Language.

1.3.3 Abbreviations

1.4 Reference Documents

- Structure of the design document.pdf
- ISO/IEC/IEEE 42010 Systems and software engineering Architecture description
- IEEE Std 1016^{tm}-2009 IEEE Standard for Information TechnologySystems Design Software Design Descriptions

1.5 Document Structure

The document is structured in six parts

1. Introduction
2. Architectural Design
3. Algorithm Design
4. User Interface Design
5. Requirements Traceability
6. References

2 Architectural Design

2.1 Overview

2.2 High level components and their interaction

2.3 Component view

2.4 Deployment view

2.5 Runtime view

2.6 Component interfaces

2.7 Selected architectural styles and patterns

2.8 Other design decisions

3 Algorithm Design

Here we will present the algorithm that will be used to create taxi queues:

First we divide the city zone in N areas, approximately equals one to each other. Then we assign a certain number of taxis to each zone, thus meaning that the taxi will work in the area he's assigned to. The number of taxis to assign to each zone will be computed dynamically based on the number of request that are coming from a certain zone, we'll call it request number, it may change from one day to another, or even from one period of time to another. The algorithm will always try to maintain the number of taxis in a zone equals to its request number.

If a taxi, while taking care of a request, leave his initial zone he will be automatically assigned to the zone he's into at the end of the ride, while assigning another taxi to the zone left by the first taxi, in order to have balance between zones. The choice of the taxi to be assigned to the zone will be made by minimizing the distance it will have to travel to enter the designated zone. If a zone's number of taxis drops below its request number the system will assign another taxi to that zone recursively till a point of balance is found. We will take preemptive measures to avoid infinite loops.

The taxis will be organized in queues, each queue assigned to a zone. When a request arrives from a zone it will be forwarded to the first taxi in queues, if he accept the taxi will be removed from the queue and will be placed at the end of the queue in the zone he will end up at the end of the ride. If he refuse it instead he will be placed at the end of the queue. If a taxi is moved to one zone to another he will take the position in the new queue equals to its previous position.

4 User Interface Design

5 Requirements Traceability

6 References