Design Document

Michele Madaschi   Lidia Moioli   Luca Martinazzi

November 17, 2015

Contents

Introduction

1.1 Purpose

In this document we aim to provide a description for the architecture and design of MyTaxiService. This document is targeted towards the future developers of the system.

1.2 Scope

The application will be developed using a client-server paradigm. The server-side application must recognize an user (either an unregistered guest, a passenger or a taxi driver), and accordingly signal the client the available actions. The server-side application must manage the city-wide taxi deployment, by the means explained in the RASD document[1] The client-side application must show a UI The application must implement a report system, in order to incentive a good behaviour of the users involved

1.3 Definition, Acronymus, Abbreviation

1.4 Reference Documents

1.5 Document Structure

---

[1]see reference documents

Architectural Design

2.1 Overview

The distributed application is composed by a server side, and a client one. The client side interacts with users (or guests), showing the correct activity, and send requests to the server sdie, when needed. The client side must be able to interact with the GPS, The server side manages requests comeing from the client side, and notify the users (or the guests) involved in the requests. The server side must also interact with a map system, in order to retreives information about the route and the km of the ride.

2.2 High level components and their interaction

The client side is composed by a set of activity composed by one or more actions and displayed through the user interface. The client side has also an interface that manage that manage the interaction with the server. The server side has a controller for each connected client, that manages the requests comeing from the users (or guests), taking data from the ride manager. It also sends messages to the clients in order to resolve the requests. The controller interacts with the clients through a network interface.

2.3 Component view

2.3.1 Client

Activity : an activity is a set of messages and actions, that the application must display to the users. No more than one activity can be displayed at the same time; The default activity is the "guest home" activity. Each time the user taps a button, the application must execute the related action, and select the next activity.

Action : an action is something that a user can do, in order to interact with the application. If an action need some data, the userinterface must display a field ,for each input needed, that allows the human to provide the necessary incormations. Some actions can also select the next

activity that must be shown, or/and send informations to the server, in order to complete their job.

Userinterface : is the component that directly interact with the human. It contains the activity that must be displayed, and read the components of the activity, in order to display them. It also launch the actions selected by the human.

Clientnetworkinterface : is the component that allow the others to exchange messages with the server sdie.

2.3.2 Server

Controller

Ridesmanager

User

2.4 Runtime view

2.5 Component interfaces

2.6 Selected architectural styles and patterns

2.7 Other design decisions

Algorithm Design

User Interface Design

Requirements Traceability

References