Content

1. Introduction

1. Purpose

In this document, more technical details will be presented than the RASD about the PowerEnjoy system.

This document aims to present how we implement the system specifically.

2. Architecture design

1. Overview

2. High level components and their interaction

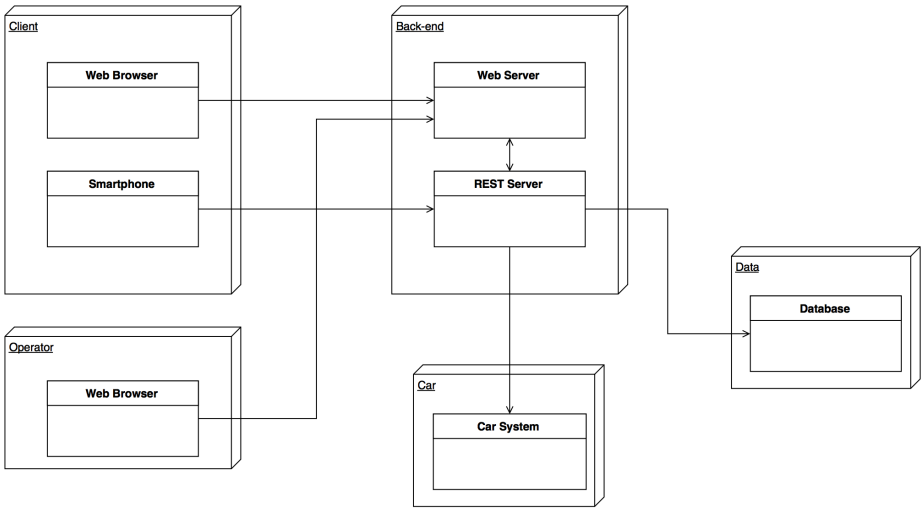

Figure 1: High Level Components

The figure above describes the high level components and their interaction. It is based on the application architecture presented in the RASD Document // TODO Add section //. The REST Server is the main component in our application as it is the central point between the GUI (Web and mobile), the database and the cars. It holds the business logic of our system. This component will be detailed in a lower level in the following section.

3.Component view

Notification Helper : Manage notifications, noticing the user that they are already close to the car.

Ride Controller : manage rides,

Reservation Controller : manage reservation,

Bill Controller : manage payment method and bills,

Economic Controller : manage money saving request,

Car Controller : manage the status and availability of cars,

Router : route the request to related controller,

Clients : mobile application based on Android and web application (browser),

User Controller : manage user, access log in or sign in request.

4. Deploying view

5. Run-time view

Sequence diagrams for - Login process W - Reservation process RA
- Billing process RA - Check-in car process J - Check-out car process J - Money saving process W

7. Selected architectural styles and pattern

Application architecture

As stated in the RASD Document, we will be using the 3-tier client-server architecture. The presentation tier is composed of the mobile application and the website. The application layer is composed of two parts. The REST Server that exposes the REST API and holds the business logic. It can be consumed by the web server or the mobile application. In addition, it is a security barrier between the client and the database as it prevents direct accesses to the database by the user. The other component of the application later is the web server. The web server takes care of formatting the data in webpages and communicating with the web browser. The last tier is the data tier which is, in this case, composed of only one database that takes care of persisting the data of the whole application. The fact that the business logic is held at the level of our servers, the client-side of the application is kept as light as possible. Therefore, users can quickly access the application by installing it on their device or browsing the website.

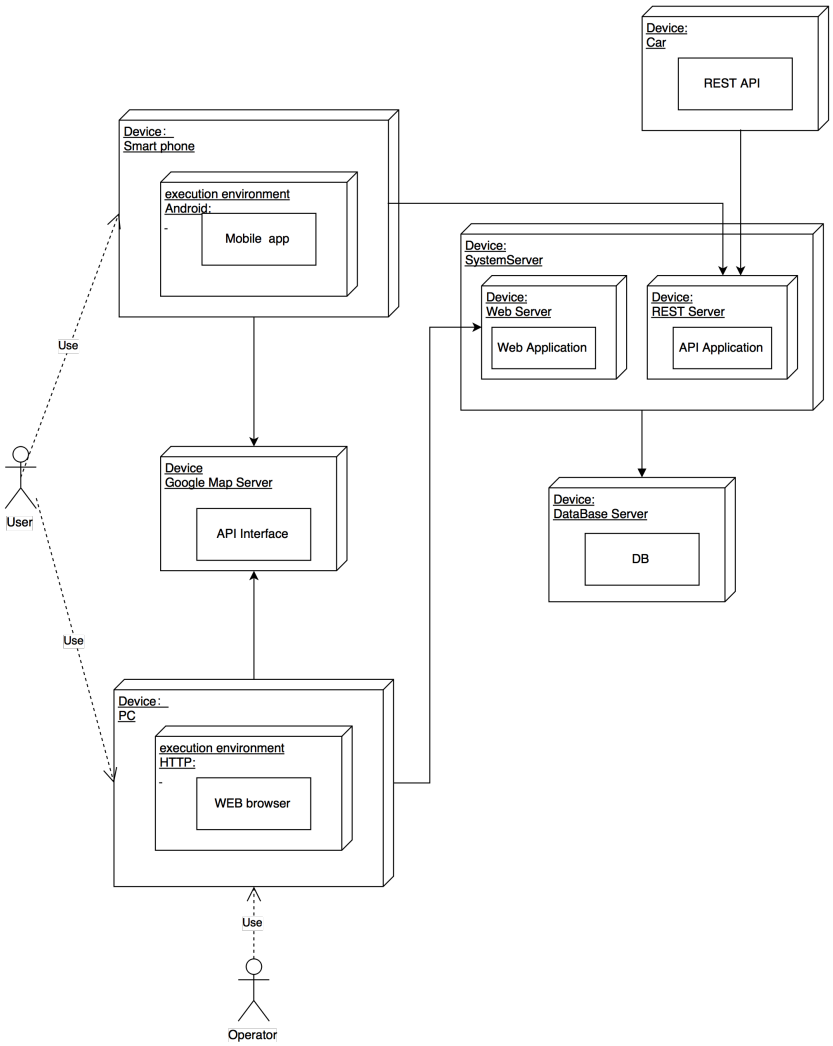

Figure 2: Deploying view

Servers application

Model-View-Controller patter is used in both the web server and the application server.