POLITECNICO
DI MILANO

SmartCityAdvisor

Design Document

Hasancan Sayılan

Contents

INTRODUCTION

Purpose

This design document gives more information in a technical perspective than RASD and aims to identify the following topics :

- High level architecture
- The design patterns
- The main components and their interfaces within
- The runtime behaviour

Scope

SmartCityAdvisor is a service based on a mobile and web application plus a finite number of large displays throughout the city of Milan. The project aims to serve only one target of actor that is

- The clients.

The clients are the people who interacts with the services using mobile/web applications or large displays. The system requires no registration or login process from the clients since such activities are not necessary for the system to maintain its progress.

SmartCityAdvisor catches the CO_2 level in the air and if it is critical, the system prevents it from going on a high level that could be dangerous for the environment. For this purpose, the system handles the entrances to the city center, allowing or preventing it according to the CO_2 percentage in the air which was captured by the sensors. If city center entrance is allowed, the system also offers the client the available parking areas in the city center. If entrance is not allowed, then the system provides alternative routes for the clients to travel in the city.

Defitinitons, Acronmys, Abbreviations

System : The digital manager of all the service.

Client : Regular people, who will recieve the notifications through web/mobile applications about the current level or carbon dioxide and whether they can enter the city center or not. The clients are not required to register or login to the system.

Traffic Lights : The lights that control the flow of car traffic in the city. With this system, they will also be used in order to balance the level of carbon dioxide.

City Center : The city center of the city Milano.

City Zone : The part of the city Milano. There are 9 city zones in total.

Sensors : The eyes and ears of the system in the city that acquires information send a message to the rest of the system such as the level of carbon dioxide in the air, cars that enter the city center and availability of the parking places in all areas in the city center.

Auctators : The devices that control the traffic lights at the main intersections in the city.

Large Displays : The screens located on the main roads entering in the city from the highways that shows information.

RASD : Requirements analysis and specifications document.

DD : Design document.

ASI : Application service interface.

UX : User experience design.

BCE : Business controller entity.

Client Application : The devices which a client can interact with the system such as web/mobile application and large displays. (ClientAPP)

API : Application programming interface.

Reference Documents

- The RASD for the same project
- Sample Design Deliverable Discussed on Nov. 2.pdf
- Lecture notes and slides

Document Structure

- Introduction: This section gives information about the purpose and scope of the design part of the project.
- Architecture Design: This is the main part of the DD and it is divided into sub topics to explain the architectural design of the project with more details that is:
    1. Overview: This section gives the basics of the design for this project, explaining the main tiers of the architecture.
    2. High Level Componenents and Their Interactions: This section explains the high level components and their interactions within.
    3. Component View: This section explains the component of the system with more details.
    4. Deploying View: This section explains the components which need deployments to be working correctly.
    5. Runtime View: This section shows the sequence diagrams with more details of components.
    6. Component Interfaces: This sections shows the interfaces between the components.
    7. Architectural Styles and Patterns: This section explains the chosen architectural styles and patterns.
    8. Other Design Decisions
- Algorithm Design: This section explains the most important parts with the help of algorithms.
- User Interface Design: This sections shows the user interface designs with mockups, UX and BCE diagrams.
- Requirement Traceability: This section connects the RASD document with the DD.

ARCHITECTURAL DESIGN

Overview

SmartCityAdvisor has three main tiers of architecture.

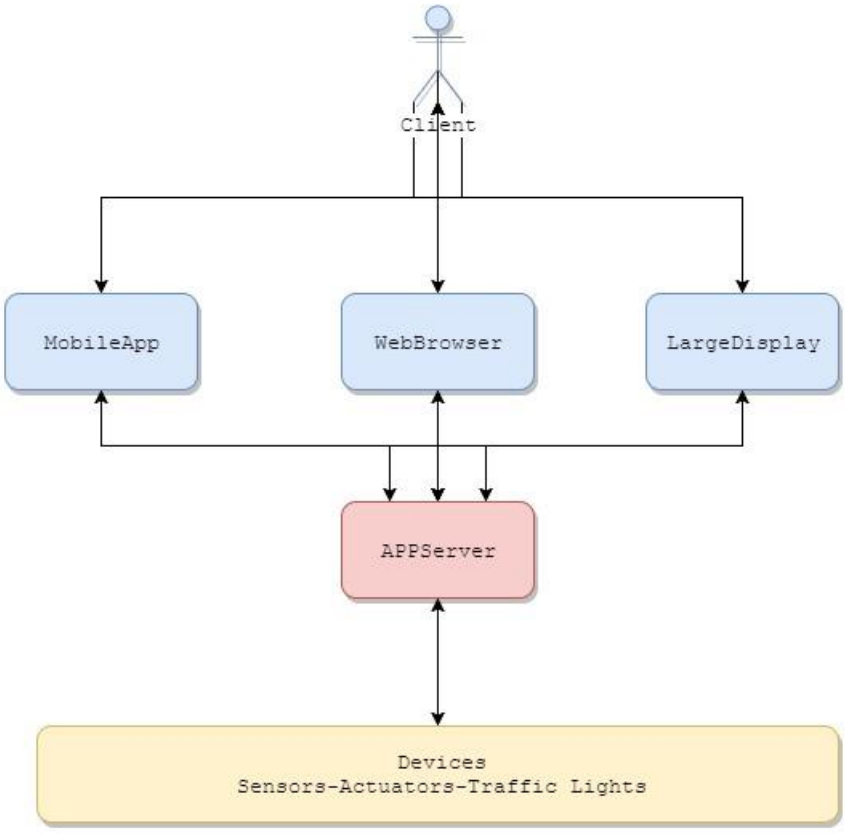

Figure 1 : Main Tiers

First tier is the client and the components that can be interacted directly by the client for the service of the project. The first tier is shown in the color of blue in figure 1.

This architecture needs no registration or login process for it is not essential. By simply downloading the application via a mobile phone, opening the web link with a

browser via computer or interacting with the large displays, the clients can achieve what he/she needs to do with the service.

The second tier which is shown in the color of red is the APPServer of the system that is basically the core of the service and the third tier is shown in the color of yellow which is the devices and their services in the system.

High Level Components and Their Interactions

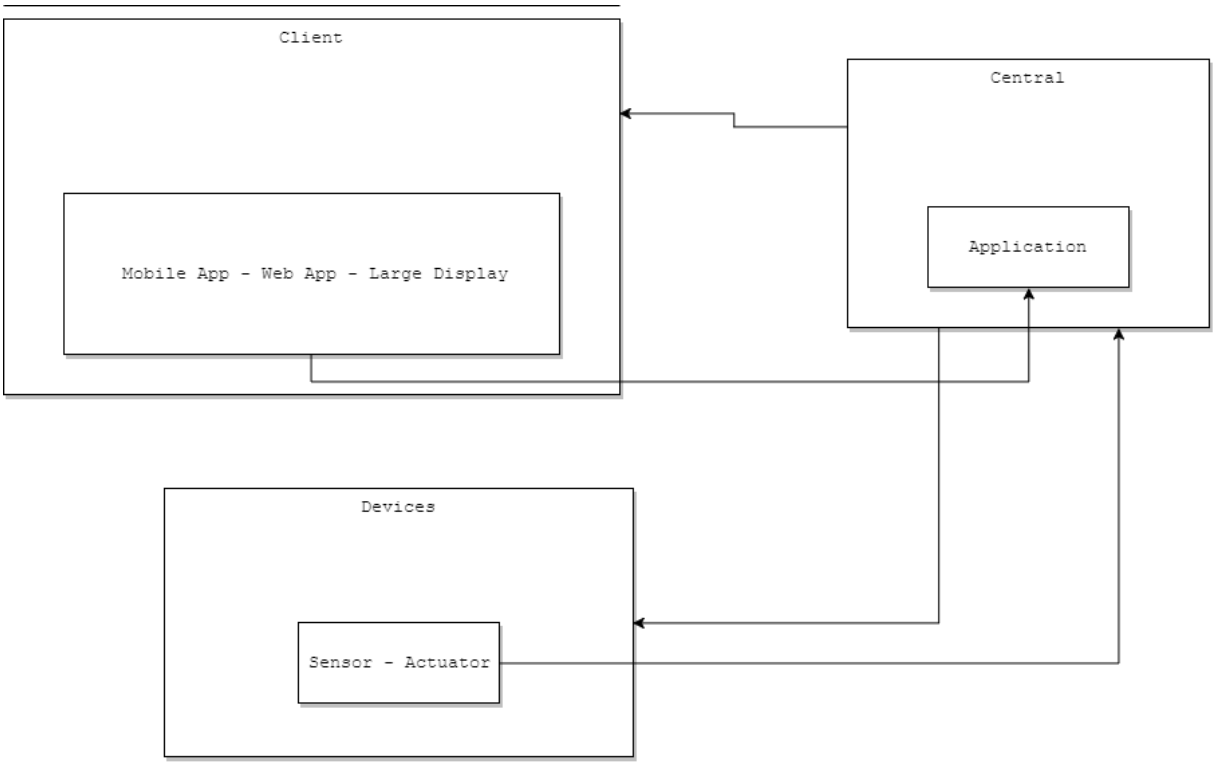

Figure 2 : High Level Components and Their Interactions

As it was discussed in the previos section, the system contains 3 main high level tiers. As a core, the central works as the application server that connects clients and their interactions with the devices and their captured informations.

The system does not start working with a request from a client. In fact it is working on a simple routine but only interacts with the client via his/her request. The client using the mobile/web applications or large displays, connects with the application of the central. Central then, using the information delivered by the devices, gives the correct service to the client.

Component View

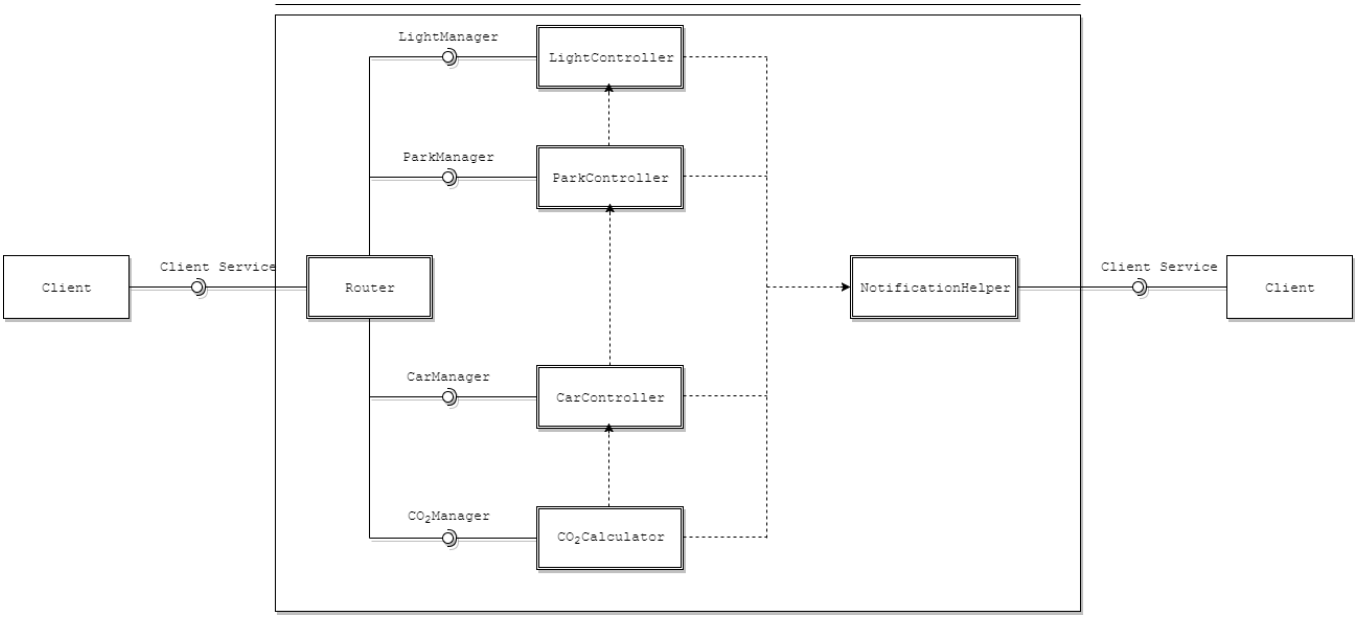

Figure 3 : Component View

- Client : The device which the actual clients uses to interact with the system.
- Router : The service which directs (routes) the requests to the appopriate service of the system.
- CO_2Calculator: The service which captures the current level of CO_2 in the air.
- CarController: The service which captures the current number of cars entering the city center.
- ParkController : The service which captures the availability of the parking areas in the city center.
- LightController: The service which controls the traffic lights depending on the current situation.
- NotificationHelper : The service which handles the notifications to interact with the clients.

The system works automatically. When a client send a request via client applications, it first interacts with the router, then the router transfers the request to the appopriate service. When the procedure is done, the service NotificationHelper interacts with the client via notifications.

Deploying View

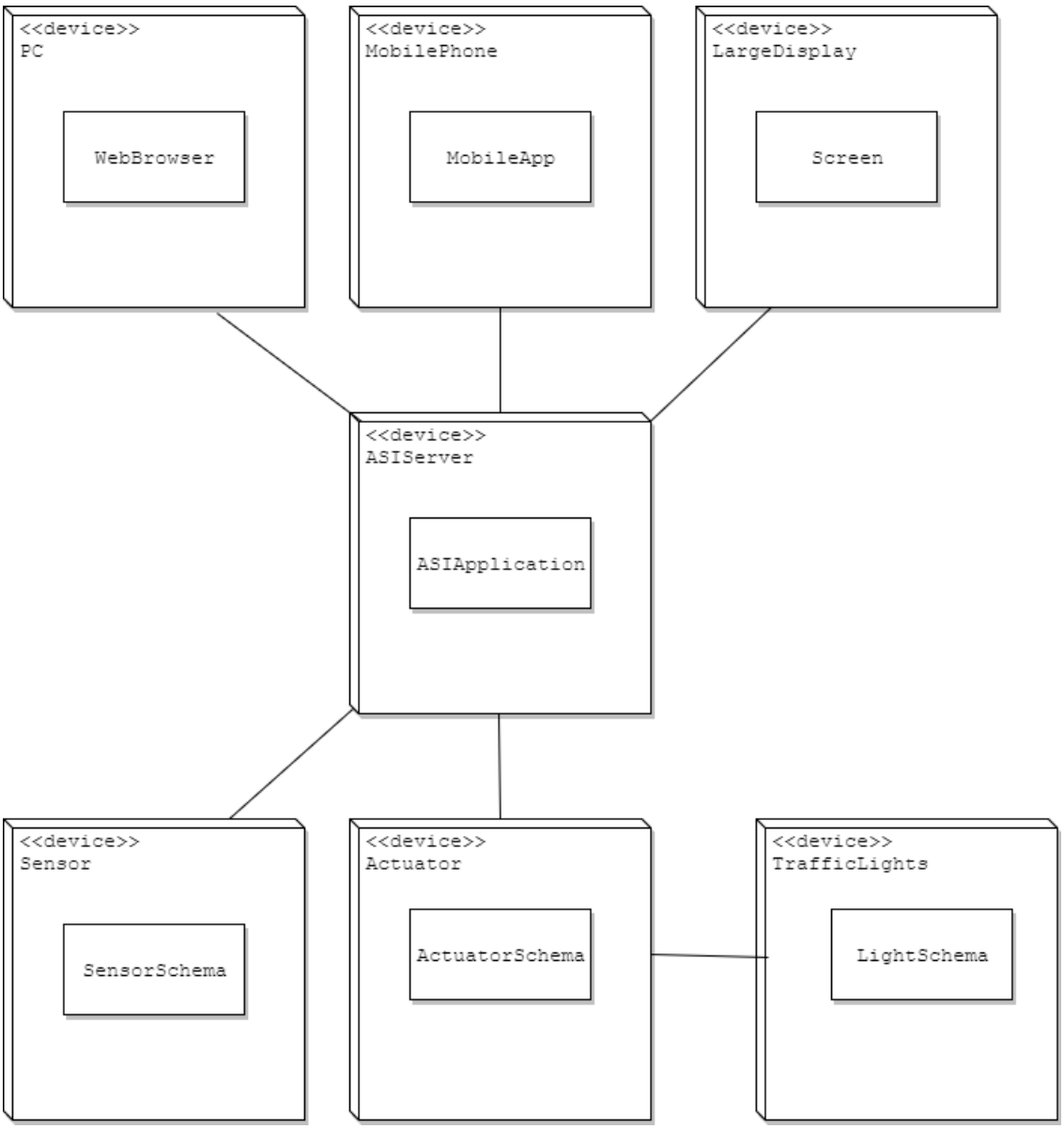

Figure 4 : Deploying View

Runtime View

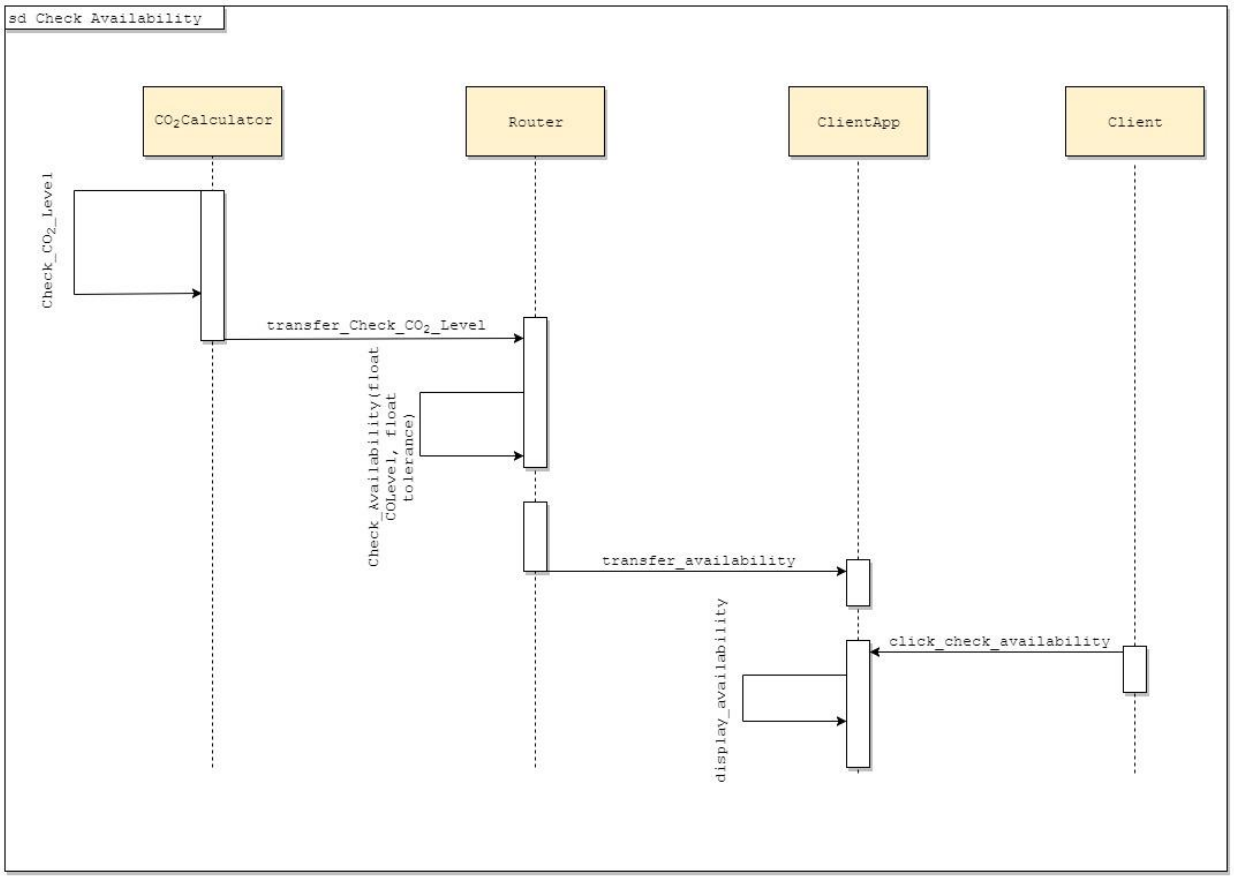

Figure 5 : Runtime View 1 : Checking Availability

For this sequence diagram, it can be seen that the system does not start working with a request from a client. The sensors using the service CO_2Calculator, captures the level of CO_2 level in the air then transfer this information to the router. The router with an appopriate function, using the CO_2 level in the air provided by the previous service, and the tolerance number that was defined before, reaches a final solution which is the availability of the entrance. The router then transfer this boolean answer to the ClientApp service in which can be displayed and showed to the client with the corresponding request.

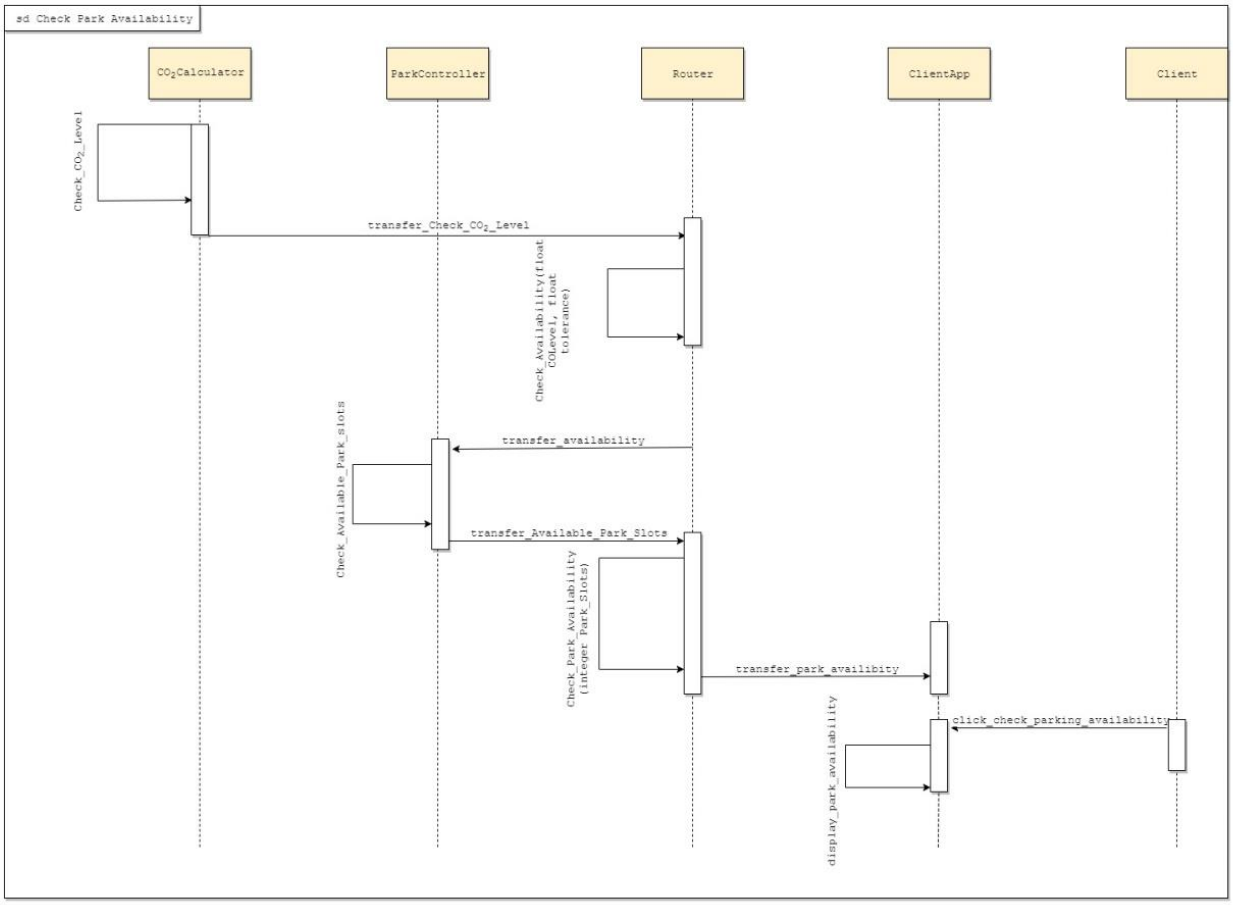

Figure 6 : Runtime View 2 : Checking Parking Slots

This sequence gives the runtime view when a client wants to know the availability of the parking slots in the city center of Milan. For this purpose, the city center entrance must be available first or else the parking places also will not be available. Thus the service CO_2Calculator and Router again decides the availability of the entrance. Then Router transfer this information to the service ParkController. Park controller captures the number of available parking slots in the city enter and transfers the information to the Router. Similar to the previous runtime process, the availability of parking places is decided then the final information is sent to the ClientApp service in which can be displayed to the client.

Component Interfaces

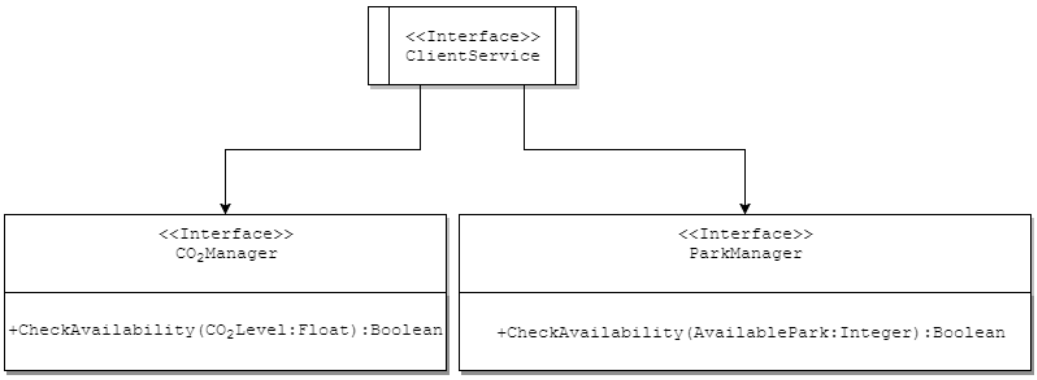

Figure 7 : Component Interfaces

Architectural Styles and Patterns

Overall Architecture

As mentioned before, the system consist of 3 tiers those are :

- Presentation Tier : Interaction tier is the part of the system that includes the client and every service and device that the client can interact with. This tier contains the mobile/web applications and also large displays. Since the system requires no registration or login process, this interfaces will be as simple as possible to increase effectiveness and useability.
- Logic Tier : The tier between the other two. In simple terms, core tier is the mid level that has all the application logic and connects the other tiers.
- Data Tier : The third tier which contains the elements that do what the service provides such as capturing information, gaining data and transfering them to the logic tier.

Protocols

The three tiers mentioned before will communicate via the following protocols :

- SOAP : This is the protocol to communicate with XML format. When the logic tier recieves a request, it will return an XML object.
- Fetch API : This is the protocol that will be used by the first tier of the architecture and also will be usefull using the first protocol since Fetch API applies SOAP.

Other Design Decisions

Beside of the main activities of the service, the system also provides alternative routes if the city center entrance is limited. This means the system must work with a map service which will be included in the service.thug Google Maps API will be used for the map service for clients who wishes to find another routes that do not go through the city center during the limited entrances to the city center.

ALGORITHM DESIGN

Algorithm 1 : Checking Availability

Algorithm 1 : Check Availability

1) **procedure** CheckAvailability(co_2level, tolerance)
2) message ← AVAILABLE
3) **if** co_2level > tolerance then
4) message ← UNAVAILABLE
5) SendNotification(message)
6) **else**
7) SendNotification(message)
8) **end procedure**

Algorithm 2 : Checking Parking Areas

Algorithm 2 : Checking Parking Areas

1) **procedure** CheckingParkingAreas(parknum, slotnum,counter)
2) **for** 0 < counter < parknum **do**
3) **if** slotnum > 0 **then**
4) message ← AVAILABLE
5) SendParkNotification(message,counter)
6) **else**
7) message ← UNAVAILABLE
8) SendParkNotification(message, counter)
9) **End**
10) **End procedure**

USER INTERFACE DESIGN

Mockups

The mockups were introduces in the RASD.

UX Diagram

UX diagrams are given in this section to explain how a client will perform his/her main action with the system.

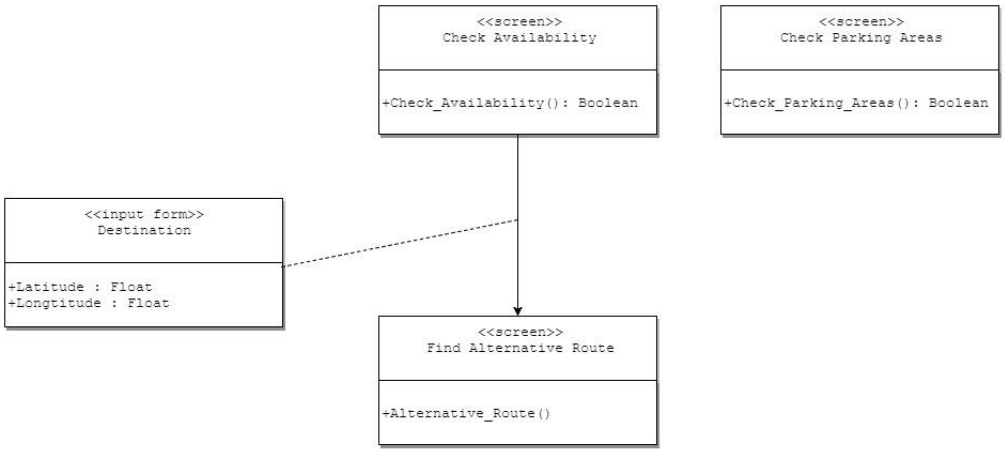

Figure 8 : UX Diagram

BCE DIAGRAM

As it was discussed in the previous chapters, the system has three main tiers that is :

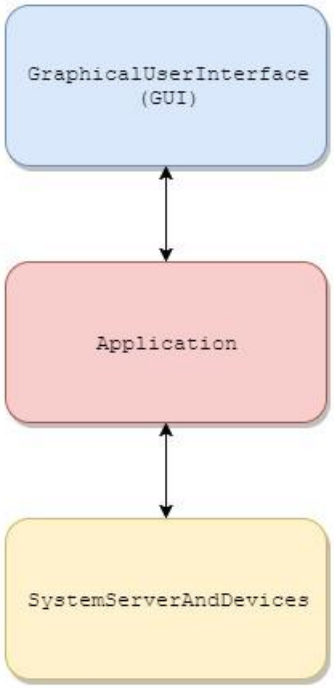

Figure 9 : Main Tiers in Color

With the purpose of explaining how client actions are managed, we use an actor based BCE diagram.

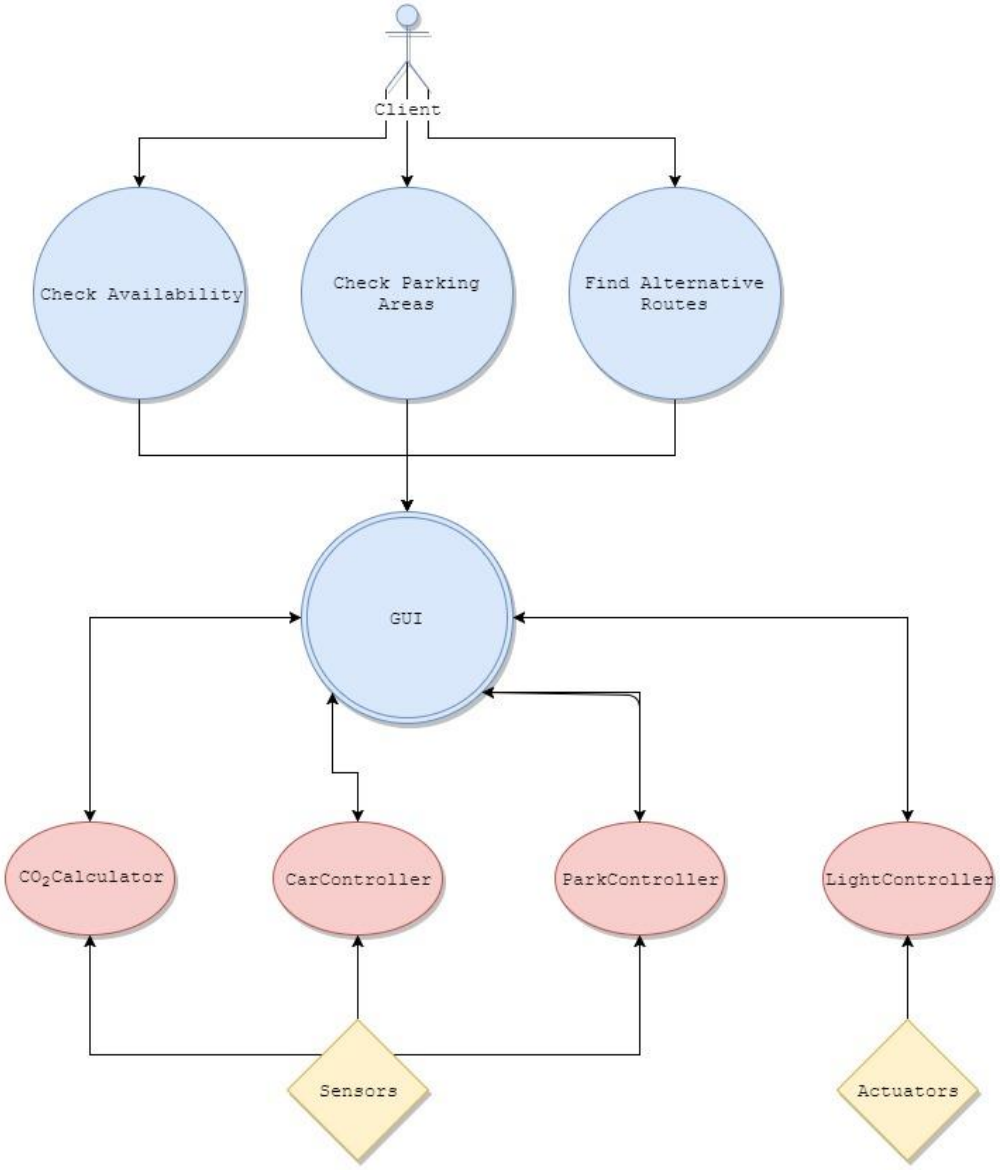

Figure 10 : BCE Diagram

The colors are linked with the BCE Diagram and the Main Tiers Figure to explain the connections by creating a physical image in one's mind.

REQUIREMENT TRACEABILITY

This section connects the DD with the RASD. By RASD here we mean the essentials of RASD to create the project that is the requirements and goals. This section links the components which were described in DD with the requirements and goals which were explained in RASD.

- [G1] The clients can get information whether they can enter the city center or not through web/mobile application or large displays.

    1. The Router
    2. The CO_2Calculator
    3. The NotificationHelper

- [G2] The clients can get information about the parking areas if they are available or not through web/mobile application.

    1. The Router
    2. The CO_2Calculator
    3. The ParkController
    4. The NotificationHelper

- [G3] The clients can get alternative routes in the city of the city center entrance is limited.

    1. The Router
    2. The CO_2Calculator
    3. The CarController
    4. The LightController

REFERENCES

Used Tools

The tools which were used to create the DD are :

- www.draw.io : To create diagrams
- Github : To upload the documents
- Lyx : To create the documents
- Microsoft Word : To create the documents

WORKING HOURS

Hasancan Sayılan

- 25/05/2017 : 2h
- 26/05/2017 : 5h
- 29/05/2017 : 30m
- 30/05/2017 : 30m
- 31/05/2017 : 4h
- 01/06/2017 : 1h
- 02/06/2017 : 6h