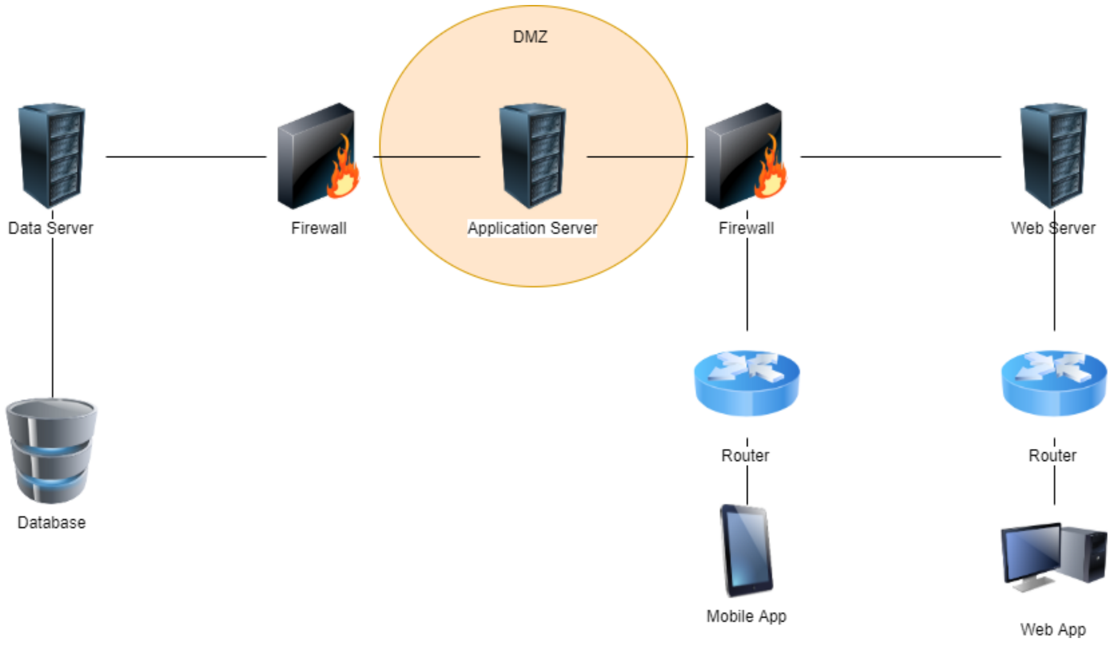
DMZ
Data Server
Firewall
Application Server
Firewall
Web Server
Database
Router
Router
Mobile App
Web App

# Design Document

Gregorio Galletti - Ibrahim El Shemy

A.A. 2019/2020 - Prof. Luciano Baresi

## Contents

# 1 Document Structure

1. `Introduction`: This section introduces the Design Document. It explains the Purpose, the Scope and the conventions of the document.

2. `Architectural Design`: This section describes the components used for the system and the relations between them, providing information about their deployment and how they works. It also specifies the architectural styles and the design patterns chosen to design the system.

3. `User Interface Design`: This section provides an overview on how the User Interface will look like. This section will be accurate enough to explain all our decisions about the design and the UI of the Mobile Application.

4. `Implementation, Integration and Testing`: This section contains the order of the system's subcomponents implementation, integration and testing.

# 2 Introduction

## 2.1 Purpose

This document represents the Design Document (DD) for SmartParking mobile application. The purpose of this document is to provide an overall guidance to the architecture of the software product and the interaction between all the components of the system to be developed, following the requirements and the goals that the software must satisfy.

## 2.2 Scope

SmartParking is a crowd-sourced application where users can view all the street parkings around them, together with detailed information. Users can also filter the parkings in several ways: searching for a specific address, a specific type, a maximum distance, etc... Moreover, users can pay the fee directly from the app, chosing the payment type and how much they want to stop.

## 2.3 Definition, Acronyms, Abbreviations

### 2.3.1 Definitions

- `User`: any client of the service, a person that logs in the system and uses it.

- `User Device`: any compatible device with the SmartParking application, mainly smartphones.

- `App`: abbreviation for the SmartParking Mobile Application.

### 2.3.2 Acronyms

- DD: Design Document.
- API: Application Programming Interface.
- GPS: Global Positioning System.

## 2.4 Revision History

- 3/04/2020: First Version of DD Document.

## 2.5 Reference Documents and Used Tools

**Reference Documents**

**Used Tools**

- *Github*: https://github.com/
- *TexMaker*: https://www.xm1math.net/texmaker/
- *Draw.io*: https://www.draw.io/
- *AdobeXD*: https://www.adobe.com/it/products/adobexd.html
- *LucidChart*: https://www.lucidchart.com/

# 3 Architectural Design

Our application is composed by two main components: a Front End, developed with the React Native framework, and a Back End that relies on Firebase, and in particular on Firebase Realtime Database.
These two components are highly connected: in fact, the mobile application has the role to show to the user all the data stored in the Database, and also to respond to the user's behaviour.
We can say that the mobile App represents the View of the system (in a Model-View-Controller like architecture), the Database represents the Model, and the entire Back End represent the Controller. That is, the App just shows the relevant data to the user, and provide the possibility to choose between some options to modify the shown data.

HIGH LEVEL DIAGRAM - app e db

### 3.1 Front End: Mobile Application

The Mobile Application is a React Native application composed by several screens, each one will be described in the User Interface Design Section. The choiche of React Native frameworks comes from the fact that, being SmartParking an application that will be mainly used "in real time", smartphones will be the target devices: it's more reasonable that a user is using his smartphone when driving (obviously respecting the street law code) rather than a tablet or a smartwatch. A **cross-platform** framework has been identified as the smartest choiche in this case, in order to reach the largest number of users.

This Application will obviously be a **multi-thread** application. This because we need to be able to handle all user actions and at the same time to keep the Google Map and the shown parkings updated, without having blocking instructions and so performing all the actions in parallel.

### 3.2 Back End: Firebase

The Firebase back end is where all the parkings, users and reservations data are stored. The back end is also encharged of authentication and user session management (a well implemented and tested functionality of Firebase).

### 3.3 Third Party interaction

- *Google*: Login
- *Facebook*: Login
- *Paypal*: Parking payments
- *Stripe*: Parking payments with credit cards
- *Redux*: Persistent storage of data
- *Google Maps APIs*: Directions, Distance matrix, Geocoding

## 4 User Interface Design

In this section we will describe, page by page, all the Screens of the Mobile Application and their functionalities, along with the decisions we made to provide a nice and pleasant User Experience.

### 4.1 Splash Screen

### 4.2 Intro Screen

### 4.3 Login Screen

### 4.4 Registration Screen

### 4.5 Home-Map Screen

### 4.6 List Screen

### 4.7 Filter Screen

### 4.8 Profile Screen

### 4.9 Vehicle Screen

### 4.10 Payment Screen

## 5 Implementation, Integration and test plan

We can split this section in three main phases:

### 5.1 Front End implementation

### 5.2 Back End implementation

### 5.3 Front End and Back End integration