AY 2022/2023

Politecnico di Milano

DD: Design Document

Marcello De Salvo Riccardo Grossoni
Francesco Dubini

Professor
Elisabetta Di Nitto

Version 1.0
January 31, 2023

Contents

1 Introduction

1.1 Purpose

The objective of this document is the realization of a full technical description of the system presented in the RASD document. Here we will analyze both hardware and software architectures, focussing on the interaction between components that constitute the system. Additionally, we will also delve into the implementation, testing and integration process. This document will use technical language as it's aimed for programmers, but stakeholders are also invited to read as it can be useful to understand the characteristics of the development.

1.2 Definitions, acronyms, abbreviations

Acronyms

- **RASD**: Requirement Analysis and Specification Document
- **DD**: Design Document
- **ITD**: Implementation Document
- **API**: Application Programming Interface
- **DBMS**: Database Management System
- **DMZ**: Demilitarized Zone
- **OCPP**: Open Charge Point Protocol
- **UML**: Unified Modeling Language
- **GPS**: Global Positioning System
- **IT**: Information Technology
- **GUI**: Graphic User Interface
- **UI**: User Interface
- **HTTPS**:HyperText Transfer Protocol Security

- **HTML**: HyperText Markup Language
- **CSS**: Cascade Style Sheet
- **JS**: JavaScript

1.3 Revision history

- Version 1.0: first release

1.4 References

- Django Framework: `https://www.djangoproject.com/`
- REST Framework: `https://www.django-rest-framework.org/`
- Vue.js: `https://vuejs.org/`
- PostgreSQL: `https://www.postgresql.org/docs/14/index.html`
- Vercel: `https://vercel.com/docs`
-

2 Development

2.1 Implemented Functionalities

Qua ci mettiamo le cose implementate

2.2 Functionalities not implemented

Qua ci mettiamo le cose non implementate

2.3 Implemented requirements

Requirement implemenmtati

R1. Matching dei requirement. NON VEDO L'ORA!!!! ✓

R2. The system shall allow users to be identified by an email of their choosing. ✓

2.4 Design Choices

Bho la droga

2.5 Adopted Development Frameworks

Mvmm o link al DD

2.6 Programming languages

natural language per chatgpt, python per tutto il resto
Loro mettono pro e contro, imo evitabili ?

2.6.1 Django Framework

descrivilo

Django Middlewares

To manage the communication between backend and frontend, we mainly use the following Django middlewares:

- *SecurityMiddleware*
- *SessionMiddleware*
- *CsrfViewMiddleware*
- *AuthenticationMiddleware*

2.6.2 Django REST Framework

We decided to pair REST framework to our Django backend since REST allows even more security features. Boh il cringe di django rest framework

2.6.3 Vue.js

Vue fa vomitare odia le lettere maiuscole

2.7 API Integration

Ne abbiamo messe 0 lmao. Qua potremmo discutere il perchè era poco fattibile farle

2.7.1 Maps API

2.7.2 Car APi

Scuffed

2.8 DataBase

Abbiamo usato postgre wow

2.9 Vercel

Spiegazione di vercel come scelta, mettere i link per accedere

3 Source Code

3.1 Backend Structure

Spiegazione ella struttura del progetto. Direi di spiegare qua la cosa del doppio backend e magari di stare attenti alle varie env.
TBH la parte sotto si può lasciare cosi com'è, non è che sia una cosa che cambia molto. Il problema forse è il filtro anti plagio.

- **ITD**: da vedere
- **dream_backend**: da vedere
- **__init.py__**: it tells the Python interpreter that the directory is a Python package
- **settings.py**: main setting file for the Django project, used to configure all the applications and middleware, it also handles the database settings
- **urls.py**: URL declarations for the Django project, it contains all the endpoints that the website should have
- **wsgi.py**: entry-point for WSGI-compatible web servers to serve your project, it describes the way in which servers interact with the applications
- **asgi.py**: entry-point for ASGI-compatible web servers to serve your project, ASGI works similar to WSGI but comes with some additional functionality
- **migrations**: Django's way of propagating changes to the models into the database schema, when changes occur this folder is populated with the records of them
- **admin.py**: used for registering the Django models into the Django administration, it allows to display them in the Django admin panel
- **apps.py**: common configuration file for all Django apps, used to configure the attributes of the app

- **models.py**: it defines the structure of the database, it allows the user to create database tables for the app with proper relationships using Python classes. It tells about the actual design, relationships between the data sets and their attribute constraints
- **tests.py**: used to test the overall functionality of the app through unit tests
- **views.py**: provide an interface through which a user interacts with a Django website, it contains the business logic of the app
- **manage.py**: command-line utility for executing Django commands; these includes debugging, deploying and running

3.1.1 Apps

Lista delle varie app e funzionalità (opterei per metterla in una subsubsection)

3.2 Frontend Structure

Spiegare la struttura della repo del frontend, simile a quella di vercel

4 Testing

AHAHAAHAHAHA

4.1 Unit Testing

Segue una eterna lista di test che non abbiamo fatto

4.2 System Testing

Testing as a whole done throughout

4.3 Post-deployment Testing

Testing post deployment

5 Installation

As a web application we chose to deploy it on Heroku, so a fully running version of the software is available at:
`https://de-salvo-dubini-grossoni.vercel.app`.

Spiegare nel caso di seguire il readme

5.1 Requirements

dire che server python di una certa versione, abbastanza copiabile dal Loro

5.2 Installation

Riscrivere a roba già detta nel readme

5.2.1 EMSP backend

5.2.2 CPMS backend

5.2.3 Frontend

6 Effort Spent

Student	Time for implementation
Marcello De Salvo	20000h
Francesco Dubini	0 h (licenziato)
Riccardo Grossoni	3 KW/h

7 References

References

[1] MDN Web Docs Glossary: Definitions of Web-related terms -> MVC `https://developer.mozilla.org/en-US/docs/Glossary/MVC`