**Distributed Systems**

Assignment 1

**ONLINE ENERGY UTILITY PLATFORM**

****

**Student:** Filip Denisa-Mariana

**Group:** 30441

**Semigroup:** 2

**Faculty of Automation and Computer Science, TUCN**

**2022-2023**

Table of Contents

[1. Conceptual Architecture of the Online Platform 3](#_Toc118756590)

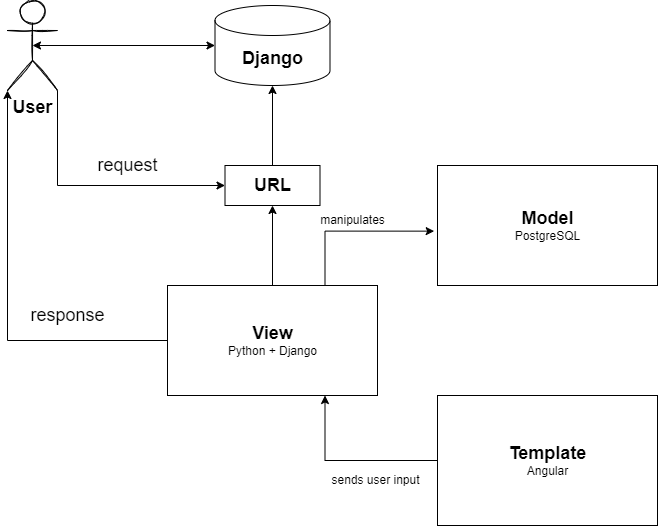
[2. DataBase Design 4](#_Toc118756591)

[3. UML Deployment Diagram 5](#_Toc118756592)

[4. Build and Execution Considerations 5](#_Toc118756593)

# Conceptual Architecture of the Online Platform

The conceptual architecture used is MVT (Model View Template), employed by Django.

* **Model** – manages the data and is represented by a database (PostgreSQL). One model corresponds to a database table.
* **View** – receives HTTP requests and sends HTTP responses. A view interacts with a model and template to complete a response.
* **Template** – the front-end layer and the dynamic HTML component, represented by Angular.

# DataBase Design

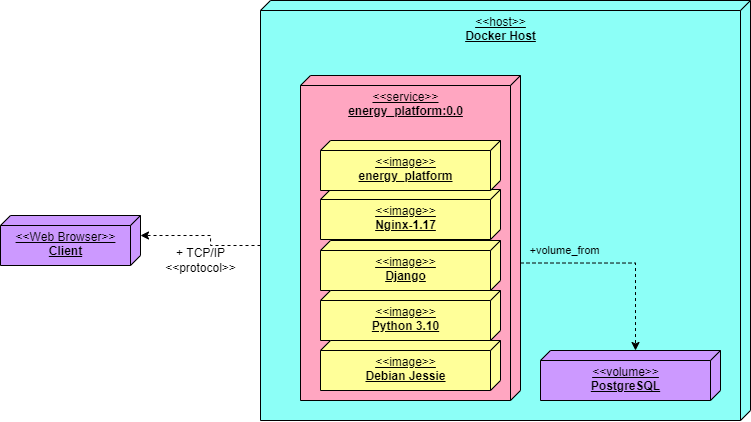
Only four tables were needed to satisfy the requirements of the assignment.

Table

Description automatically generated

# UML Deployment Diagram

**Back-end**: Python + Django

**Front-end**: Angular  
**DataBase**: PostgreSQL

# Build and Execution Considerations

For Docker deployment, a **docker-compose.yml** file is provided for front-end, back-end and database containers. The **create\_private\_network.bat** file is used to create a Docker network to bind the containers together.

To start the deploy, execute the create\_private\_network.bat file by double clicking on it, and then run:

docker-compose up --build

Three Docker images will be created:

* postgres
* assignment1-angular-integration
* assignment1-django-integration

And three Docker containers, on top of the Docker images:

* postgres-integration
* angular-integration
* django-integration

The backend can be accessed at <http://localhost:8000/api>, and the front-end at <http://localhost:4200>