Technical Report - **Product specification**

# SolarLink

| | |
|---|---|
| Course: | IES - Introdução à Engenharia de Software |
| Date: | Aveiro, <date of submission> |
| Students: | 102993: Pedro Matos<br>103252: André de Sousa Fernandes<br>107634: Rodrigo Martins Graça<br>104092: Simão Moreno Antunes |
| Project abstract: | Is a web based application that offers the means to manage a solar grid system and its internal logic. |

Table of contents:

## Índice

Introduction ..... 2
Product concept ..... 2
Vision statement ..... 2
Personas ..... 3
Main Scenarios ..... 4
User Stories ..... 4
Architecture notebook ..... 5
Key requirements and constraints ..... 5
Architectural view ..... 6
Module interactions ..... 7

# Introduction

This project was made in the context of the Introduction to Software Engineering course. In it, we propose, conceptualize, and implement a solution for a Household Solar grid management system, or as we call it, SolarLink.

This document serves to describe the product requirements and project structure, within the scope of the course. Personas, use cases and main scenarios are explored in the following chapters. The chosen architecture is also described.

Each member of our team has a defined role, in order to split the workload in an efficient manner:

- Team Manager - André
- Product Owner - Pedro
- Architect - Simão
- DevOps - Rodrigo

# Product concept

## Vision statement

SolarLink, was developed with the intention to provide a tracking and management service for a household solar grid system.

This system was designed to be administered by a grid supervisor (super-user) and used by regular people with solar panels on their homes (regular users).

It will provide real-time data about the consumption and production of solar energy as well as historical data and statistics. It will also serve as a management hub for the grid supervisor, where he can manage new users and old ones as well as manage extreme events such as the malfunctioning of the solar panels in a specific home.

With this system in place, we can easily supply regular households with the information they need to manage their energy consumption, and the means for the grid supervisor to handle all his workload.

## Personas

Daniel Rodrigues is a 30 year old male constructor, lives in Aveiro with his wife, two daughters and his son. He likes to spend time with his family and co-workers, likes to watch football matches, is a supporter of S.L Benfica and on Sunday likes the occasional barbeque.

Due to the current rise in living costs he decided to purchase some solar panels to save on the electricity bill, but a few months have gone by and although he is spending less than before, he believes that he can save more by tracking the energy production and consumption of the house and develop a more efficient way of spending energy during the day and night.

Vitorino Machado is the grid supervisor. He is 45 years old and still single. Lives in Leira but also has a home in Alentejo where he likes to spend the weekends.

He likes to hang out with his friends and go out at night, he is very present in social media. He is a hardworking man and very focused and confident.

In his work as the grid supervisor, he is the one that makes sure that everything goes as smoothly as possible.

## Main Scenarios

- Daniel heard about our service and wants to register, after he fills the form with the necessary information he waits for the acceptance of his profile into the service and makes the appointment for the sensor installment. After the installment, he can track real-time production and consumption of energy, historical data and statistics as well as report malfunction of his system.

- Vitorino starts to work at 9h00 and logs into his account. He starts by reviewing the new applications for the service and confirms them as well as the appointments of the installations. In mid-afternoon he receives an alert that a house is not producing energy when it should, so he contacts the client through the app and arranges a possible appointment to solve the issue.

## User Stories

1. As a user, I want to see how much energy I'm consuming and producing in real time, as well as historical data of production and consumption of energy. I also want to be able to report when something is not working right.

2. As the grid supervisor, I want to have access to all the available data, to accept new users, mark appointments, receive reports from the users and receive alerts of extreme events.

# Architecture notebook

## Key requirements and constraints

- The users should be able to have access to the WebApp on any platform requiring only that they provide the right credentials for that.
- The users must be able to change their personal account information.
- The grid supervisor account parameters should be name, email, phone number, home address, username, and password.
- The user account parameters should be name, email, phone number, address, username, and password.
- The grid supervisor must be the only one able to manage all the accounts.
- A user can see how much energy it is consuming and producing in real-time.
- A user can consult his historical data.
- A user can report malfunction or ask for assistance.
- The grid supervisor has access to the grid consumption and production in real time as well as historical data.
- The grid supervisor receives alerts of extreme events.
- A user receives alerts of extreme events.
- The grid supervisor is the only one able to approve appointments.

## Architectural view

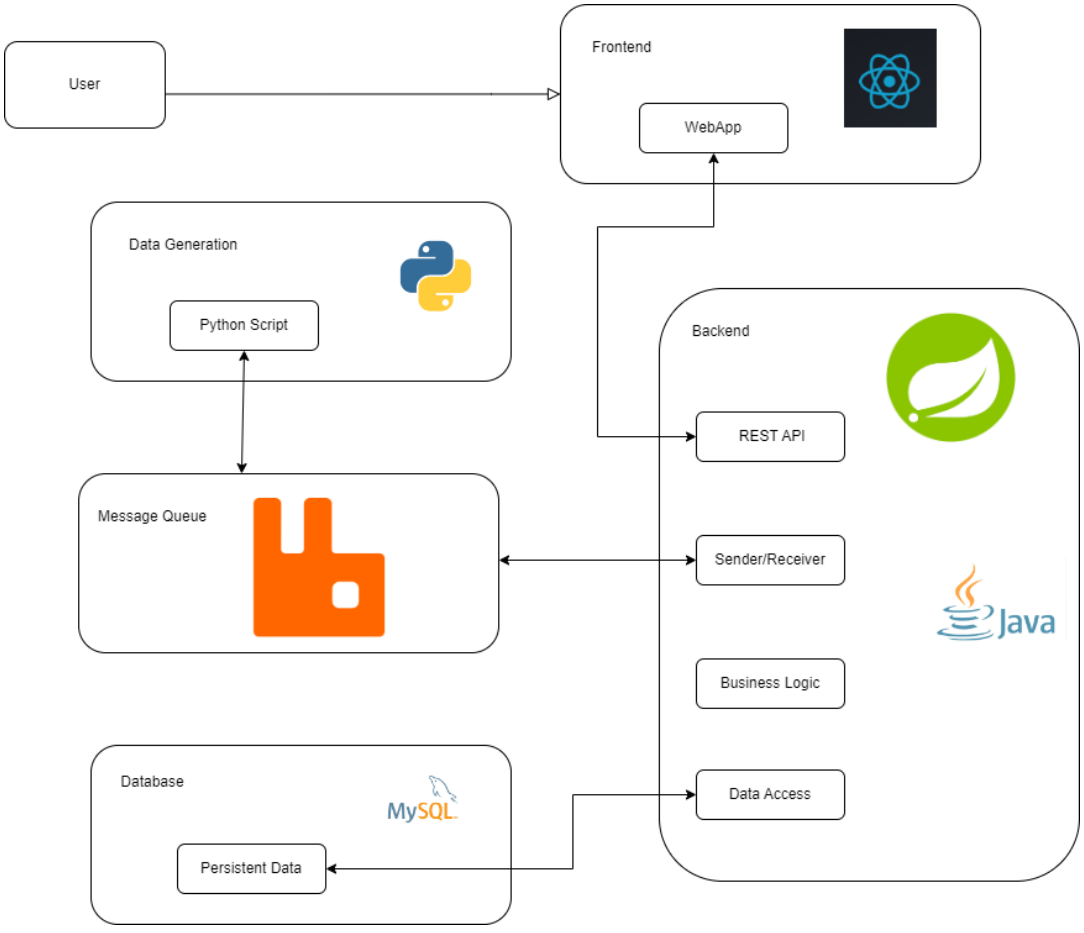
The project architecture will be split in 4 main parts:

- The data generation layer, which is responsible for generating data about the application context and will send it through message queues to the backend for processing. The data generation will be simulated using python scripts and the messages queues will be implemented using RabbitMQ;

- The database, which will store all our persistent data. The database will be implemented in MySQL, using a relation-type database;

- The backend layer, responsible for handling all internal logic. It fetches and stores from the database, receives data from message queues, processes it and displays it in the frontend, through websockets. It will be implemented using Spring Boot;

- The data presentation layer, or frontend, which will communicate through the API-part of the backend - to fetch the data it should present. It will be implemented using React.



## Module interactions

The interactions between modules respect each module's responsibility and role in the app.

For example, for the login of the user, the frontend posts the login credentials to the API. The backend then verifies the credentials, using the information stored in the database.

Assuming the login went as expected (valid credentials), the web App will then display the respective dashboard, with the respective data that the user is allowed to have access to.

The data generation layer generates data according to the home solar production and consumption context, which is passed to the backend for processing and storage.

Most of the module interactions can be summed up with the following sequence diagram: