**AGU Data System: Project Proposal**

Projecto e Seminário - Spring Semester 2023/2024
Licenciatura em Engenharia Informática e de Computadores
March, 2024

| | | | |
|---|---|---|---|
| **Students:** | Carlos Pereira | n.º 48253 | e-mail: a48253@alunos.isel.pt |
| | André Matos | n.º 48331 | e-mail: a48331@alunos.isel.pt |
| | Rodrigo Correia | n.º 48335 | e-mail: a48335@alunos.isel.pt |

| | | |
|---|---|---|
| **Supervisors:** | Pedro Miguens | e-mail: pedro.miguens@isel.pt |
| | Ricardo Enguiça | e-mail: ricardo.roque@isel.pt |

## Introduction

Currently, gas supply is an essential service that should not be interrupted. However, when this supply is made through Autonomous Gas Units (AGU), it is necessary to forecast and anticipate consumption, guaranteeing minimum reserves in each unit.

Information on the levels of the AGU reservoirs is obtained from different recording platforms, some with data history, others without, allowing only instantaneous information.

Historical hourly information could be useful in future gas consumption forecasting algorithms, depending on various input parameters, such as gas consumption at the AGU, weather data in the AGU's geographical area, etc.

In order to provide a data system with a larger volume of data than the existing one, the aim is to develop and implement an infrastructure for collecting and storing input parameters, a dashboard for viewing gas consumption forecasts, and a delivery scheduling system.

This project will implement the infrastructure to support and visualize the forecasting algorithm, based on an architecture with a Back-End (database and Web API) and Front-End (Web App).

## Requirements

To achieve the project goals a set of requirements was defined, such requirements are either mandatory or optional.

**Mandatory Requirements:**

- **Dynamic Fetching Scheduler:** A Dynamic Fetching Scheduler that allows periodical retrieval of data from providers, for example, gas and weather information;
- **Database:** A database that persists data related to the application;
- **Web API:** A RESTful Web API that will access and manipulate the database content;
- **Web App:** A front-end web application that allows the viewing and manipulation of data;
- **Integration with prediction algorithm:** Integrate data from providers with the prediction algorithm.

**Optional Requirements:**

- **Routing:** A routing service to manage the fleet of deliveries to the AGU, can be made more complex by making sure that the deliveries are well balanced, with a similar amount of work and kilometers every week, taking into account available transport service.
- **Delivery Reports:** Generate optimal weekly cargo plans, based on the prediction algorithm and previous gas consumption.

## Proposed System Architecture

The architecture of the proposed system, is depicted in Figure 1, being composed by a Back-end component, that contains the database and micro-services that will be used to manage the Web API. In the Front-end component, we will have the part that represents the visualisation of the functionalities.

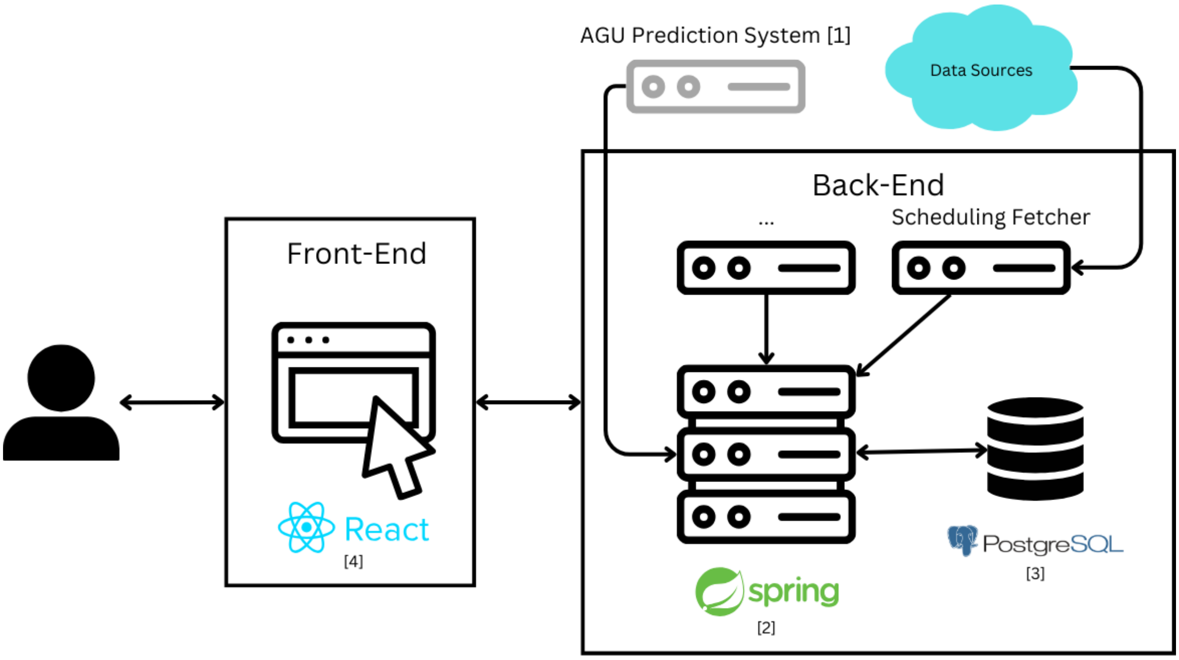


Figure 1: Proposed System Architecture

### General Description

The Web Application allows to see real live data gathered by the Web API, such data can be: alerts, dashboards or consumption predictions.

- **Scheduling Fetcher:** Micro-service that schedules a search for information from data sources. This information will later be used by other services with the prediction component for the best procedure to take.
- **Data Processing:** Micro-service that will fetch the saved data from the Scheduling Fetcher service, manipulating it using the prediction algorithm, that is, an algorithm that predicts the quantities that were present in the AGU to determine the best procedure for each AGU, and sending them to the Ideal Weekly Plan service.

- **Ideal Weekly Plan:** Micro-service that will already receive the data processed from previous services to then be used to carry out checks on levels and possible loads, that will later be displayed on the dashboard and in alerts.
- **Dashboards:** Multiple viewing dashboards, including a National, Regional, and AGU overview of the current and predicted levels of gas.
- **Alerts:** Alerts that require manual review will be shown based on important business rules, such as, if a certain AGU level is critical, a certain gas delivery to an AGU that was supposed to be made in the morning wasn't finished, a gas loading from an Liquefied Natural Gas (LNG) Terminal was supposed to be loaded but wasn't, amongst others that might be relevant.

The proposed system aims to establish a robust infrastructure for managing AGU-related data, enhancing forecasting capabilities, and optimising gas delivery processes. By integrating dynamic data fetching, a relational database, a RESTful Web API, and a user-friendly web application, the system provides a comprehensive solution for monitoring and managing gas supply. The inclusion of optional features like routing and delivery reports, further adds flexibility and optimization possibilities to the overall system.

## Planing

Figure 2 is the general planning for this project with the deliveries and tasks to perform, although the timeline for the tasks can slightly differ the deliveries are immutable.

## Task Flow

Firstly, we will start by analyzing the sketch made by the client (GALP), about what the dashboards will be, to understand and develop which fields are necessary to store in the database. Then, we will create the Web API micro-services that will use the database already created, and subsequently, pass the processed information to the Web App. In the Web App, with the sketch created at the beginning, we will replicate what was planned, improving the points that were changed throughout the project, creating an user-friendly Web App, with functionalities that encompass the functions created in the Web API.

Each section of the project, database, Web API and Web App, will be developed and supervised by all students in the group, however, to facilitate better organization and coordination between the group, each section will have only one student as the main supervisor.

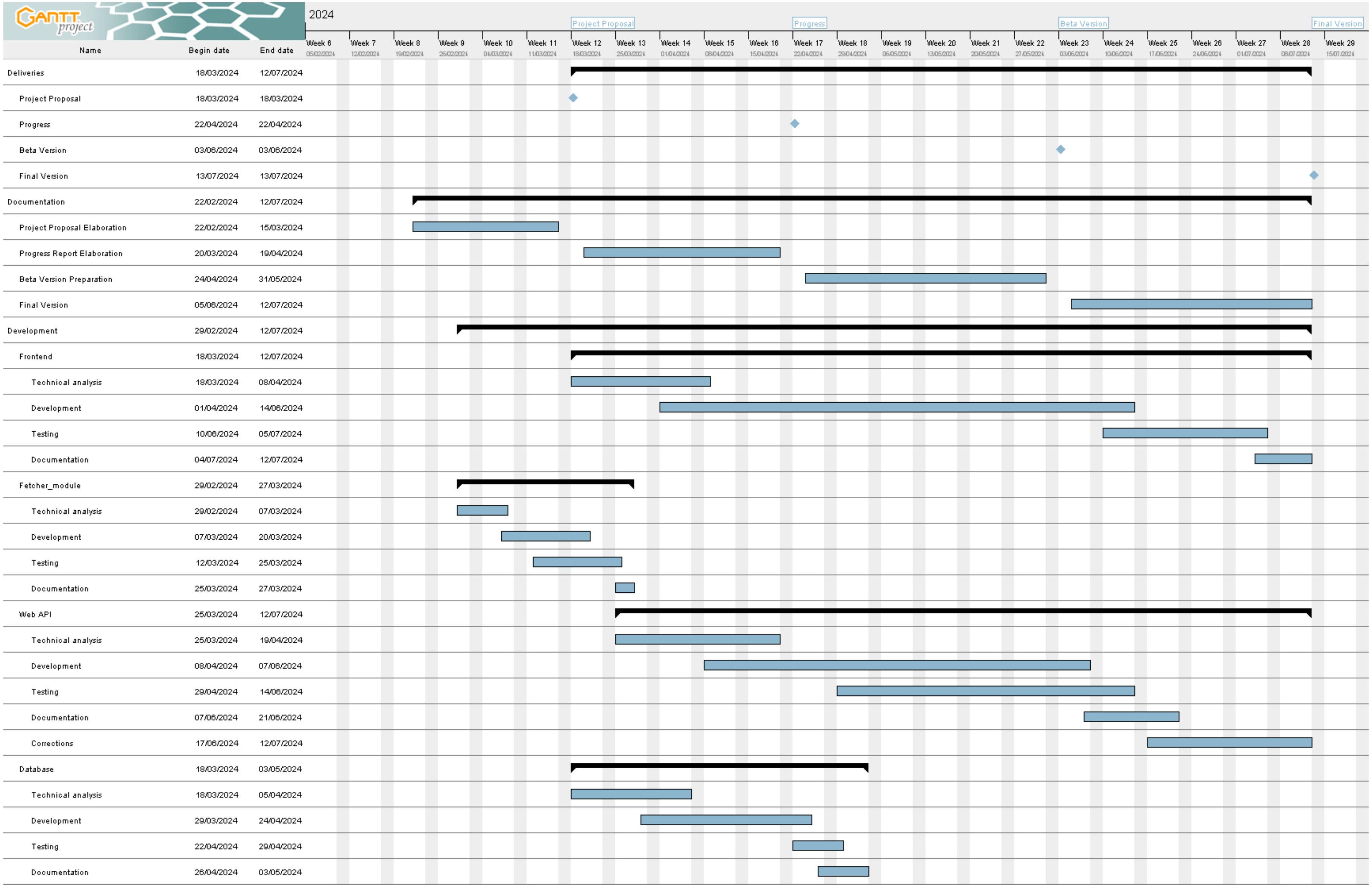
## References

[1] João Assunção, UAG Prediction System, 2023, Projecto e Seminário, Engenharia Informática e de Computadores.

[2] Spring , 2024. Online, accessed 28/02/2024
`https://spring.io/`

[3] PostgreSQL , 2024. Online, accessed 28/02/2024
`https://www.postgresql.org/`

[4] React , 2024. Online, accessed 11/03/2024
`https://react.dev/`



| Name | Begin date | End date |
|---|---|---|
| Deliveries | 18/03/2024 | 12/07/2024 |
| Project Proposal | 18/03/2024 | 18/03/2024 |
| Progress | 22/04/2024 | 22/04/2024 |
| Beta Version | 03/06/2024 | 03/06/2024 |
| Final Version | 13/07/2024 | 13/07/2024 |
| Documentation | 22/02/2024 | 12/07/2024 |
| Project Proposal Elaboration | 22/02/2024 | 15/03/2024 |
| Progress Report Elaboration | 20/03/2024 | 19/04/2024 |
| Beta Version Preparation | 24/04/2024 | 31/05/2024 |
| Final Version | 05/06/2024 | 12/07/2024 |
| Development | 29/02/2024 | 12/07/2024 |
| Frontend | 18/03/2024 | 12/07/2024 |
| Technical analysis | 18/03/2024 | 08/04/2024 |
| Development | 01/04/2024 | 14/06/2024 |
| Testing | 10/06/2024 | 05/07/2024 |
| Documentation | 04/07/2024 | 12/07/2024 |
| Fetcher_module | 29/02/2024 | 27/03/2024 |
| Technical analysis | 29/02/2024 | 07/03/2024 |
| Development | 07/03/2024 | 20/03/2024 |
| Testing | 12/03/2024 | 25/03/2024 |
| Documentation | 25/03/2024 | 27/03/2024 |
| Web API | 25/03/2024 | 12/07/2024 |
| Technical analysis | 25/03/2024 | 19/04/2024 |
| Development | 08/04/2024 | 07/06/2024 |
| Testing | 29/04/2024 | 14/06/2024 |
| Documentation | 07/06/2024 | 21/06/2024 |
| Corrections | 17/06/2024 | 12/07/2024 |
| Database | 18/03/2024 | 03/05/2024 |
| Technical analysis | 18/03/2024 | 05/04/2024 |
| Development | 29/03/2024 | 24/04/2024 |
| Testing | 22/04/2024 | 29/04/2024 |
| Documentation | 26/04/2024 | 03/05/2024 |

Figure 2: Project Timeline