**1 Introduction**

**1.1 - Background**

Nowadays, within a company, it is even more important to have an organized and cooperative team with knowledge of all the steps and goals that need to be worked on for the various projects they are currently participating in.

Each member of the team must keep track of high amounts of information, making it an easy task to overlook or forget issues that might bring problems to a project later on.

The information relative to a project can be obtained from different sources and it is all aggregated in one place, readily available to be displayed to a work team to better guide a certain project's development.

**1.2 - Relevancy**

The Lean Dashboard Project will be developed to help the company's workers keeping track of all the possible tasks for their projects, gathering all the information needed for the various activities from the many sources that are necessary, presenting it on an easy to read and reactive web application.

The project will center around the development of a responsive web application capable of running on a multitude of devices, ranging from smartphones to desktop computers to large screens such as TV’s. This web application will display to a work team of a company all the information regarding various projects being worked on.

The information being displayed will show the team what needs to be addressed in the project at hands, such as milestones, bugs, and current errors in the project.

**1.3 – Report Organization**

In this report, we will explain the early drafts of the implemented solution to the problem at hands, as well as provide a more in-depth analysis of the architecture and technology choices.

**2 Solution**

**2.1 – Functionalities**

The solution will be a responsive mobile-first web application that will provide features to its users such as:

* The creation of new projects by a back-office manager
* The creation of various Dashboards with the desired widgets
* The customization of a dashboard by selecting the widgets the user wants
* The management of projects by the back-office manager
* A view of various dashboards for each project the user is part of

**2.2 – Architecture**

**2.2.1 – Hardware Architecture**

Diagram

Description automatically generated

The system being developed is composed by the following modules: the ETL, responsible for obtaining and transforming the information from the various sources of information, the Lean Dashboard API, which will be used to gather all the information regarding the various platforms (such as Jira, Squash and Azure) in the form of widgets, and the application itself to be displayed in the various devices.

**2.2.2 – Software Architecture**

Architecture principles:

* ETL component is responsible for obtaining the information from the various sources and adapting it to widgets. It is a separate model from the application itself
* Lean Dashboard Server serves as a gatherer of information. Inside a Lean Dashboard project, a user can then create various Dashboards that inside have a variety of Widgets (that were created thanks to the ETL)

The back end was designed as following:

Diagram

Description automatically generated

Lean Dashboard Server:

* Designed as a Web API for the client to call it when it is developed.
* Accesses the Database to create and manage projects and dashboards.
* Accesses the ETL to create widgets with information and the scheduler to keep these updated.
* TRY TO WRITE SOME MORE

ETL:

* The main module Services gathers all the needed information and executes the needed functions to update and create widgets.
* Data module responsible for the access to the various data sources
* Transformer module responsible for the transformation of the data that it receives into widgets or smaller objects containing the needed information.
* TRY TO WRITE SOME MORE

**2.2.3 – Data Model**

The application requires a data model and a database to store all the required data.

The chosen database technology was Elastic Search since it provides a flexible no-SQL database. Since in this application there will be a great variety of widgets, all with different structures and data being stored, a NoSQL database such as a Elastic Search came with the advantage of allowing us to store said widgets all in the same index, even with said widgets being different from each other.

The data model would be organized by the following resources:

A picture containing text

Description automatically generated

**Lean Projects**

* Project objects have their own Id in the Elastic Search index.
* The property “Project Dashboard” represents an array of dashboards
* Only the owner (a manager) can edit the project
* The members are only there as viewers. They can access dashboards a view relevant information
* The main gatherer of information

**Dashboards**

* Dashboards are only present inside the array of dashboards in a project
* They serve as the collection of widgets
* A dashboard would be an entire page being displayed in the client (showing all the widgets present)
* The widgets array will contain the Id of the widgets inside said dashboard

**Widgets**

* Widgets are stored inside an elastic search index and each one has an unique Id
* Widgets are the main information being displayed
* They are the final result of the ETL Procedure
* Besides a few key points, the layout of a widget is arbitrary, because of the different types of data being displayed
* Will contain the necessary information to be displayed in the Lean Dashboard application, such as pie charts, data tables, gauge charts and graphs

**Widgets Structure**

Widgets can be a very abstract concept and somewhat difficult to grasp. Generally, we would go to the various APIs being used (we use the Jira, Squash and Azure ones right now), retrieve the desired information and present it in an application.

The Lean Dashboard application differs from some others in that aspect. The information