<University Name>
<Department Name>

Summer Practice Report
<Course Code>

Student Name:

Organization Name:

Address:

Start Date:

End Date:

Total Working Days:

Video Presentation Link (optional):

Contents

1 Introduction

This document contains information regarding my internship in Siren Bilisim on summer of 2021. The document will consist of information about the organization, project I have worked on and my role in the project. Although its development process isn't perfecly aligned with software development lifecycle, the details regarding the project, namely PGMaster, will be organized as following:

- **Analysis:** In this section, I will be explaining the effort put into understanding and determining the requirements for the software.
- **Design:** In this section, the design choices that were made in accordance with the requirements will be demonstrated.
- **Implementation:** The development of the software in somewhat intertwined with the prior phases. However, explanations regarding the implementation specifically will be included in this section.
- **Testing:** This software's implementation involves thorough testing of its different modules. In terms of the timeline, the testing is not necessarily performed *after* implementation. Information regarding testing will still be written in this section of the document for sake of sticking with the outline demanded by the department.

The internship is performed in Siren Bilisim under the supervision of Koray Kocamaz. During that time, I have been mainly involved in web development activities. Before the internship, I have had developed various applications that utilizes web technologies. During the internship, I had chance to learn more about frontend development on top of backend. The project mainly involves active participation of three developers. While I have performed my day-to-day development activities under supervision of Koray, the team member that I worked with the most was Derviş Mahmutoğlu from HAVELSAN. In that regard, I could partially claim that my internship project included the contribution of multiple organizations although HAVELSAN itself is not officially involved.

Although my involvement in PGMaster was not limited to the frontend, majority of the source code I have written would allow users interact with the underlying system. Due to that, I will be sharing extensive demonstration of the user interfaces.

On top of web development, the project revolves around databases, containers, and clusters. Domain knowledge about these entities have been also learned during this internship. I will be mentioning them in detail within relevant sections of the report.

2 Information about Project

In this chapter, I will be explaining technical details regarding the project. To provide better context, I shall be introducing the project first.

PGMaster is an admininstration tool that mainly intends to help database and system administrators. PGMaster provides an interface for the systems that utilize containers which are homes to databases. Using Docker containers, our system creates and manages clusters. These clusters offer capabilities like high availability and data integrity. With various configurations, databases can be set up with modes like master, standby and more. Intended users of the product is admins that manage systems with high amounts of transactions. Whether it is private or official organization, many admins perform high level database management operations using tools on command-line interface (CLI). PGMaster can be considered as an abstraction over such interfaces as it allows critical operations to be initiated, observed and completed using a graphical user interface (GUI).

PGMaster, although trivializing many tasks for admins, provides a CLI over the web application as well. This is achieved using WebSockets as well as various libraries which will be explained further in detail later on.

Without any due, I will get into the phases I have been through during my internship.

2.1 Analysis Phase

On my first day of the internship, after being given a computer and setting up my development environment, I have been introduced to Derviş. Prior to my involvement, he was one and only contributor of the project. It was determined that I was going to implement frontend application of the system. After discussing the high level requirements of the system, it was clear to be that library that I was going to use for the project, namely Reactjs, was suitable. The project didn't have any document prior to implementation.

Before my involvement, Derviş had already implemented some of the backend application modules which was using Java and Spring Boot framework. Being an experienced software developer (with 20+ years of experience), he was quite comfortable implementing Java and bash code. He also had extensive knowledge on PostgreSQL and database management systems in general. However, he needed a hand with the frontend application. There was only a JavaServer Pages (JSP) implementation with a form and some buttons written in HTML which were not sufficient for more sophisticated and dynamic nature of the system.

We have discussed the project briefly and I proceeded to learn Reactjs the following days. I will be telling about implementation related details that were discussed on initial meetings on following sections. After understanding the general requirements of the system I have spent time analyzing the following:

- Nature of the data to be processed.
- Design related constraints.
- Candidate modules and external interfaces.
- Documentation needs.
- Conventions to be followed during implementation.

I will try to explain considerations I have had for each of these items. Before jumping into these items, it seems necessary to talk about PGMaster modules and entities. The modules can be listed as follows:

- **PGManager:** Even if the name suggests something similar to PGAdmin, a web-based GUI tool that is commonly used to interact with PostgreSQL databases, it is actually the backend of PGMaster application. It is written in Java (using Spring Boot framework) and it has two main purpose: Performing CRUD operations on pgm database, the database that contains "meta" data regarding the application itself and manage operations that affect PGMaster entities. These entities will be explained below.
- **Grafana:** Grafana is a multi-platform open source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to data sources. Our web application's frontend is intended to be embedded into one of Grafana's custom panels. Within the context of PGMaster, this is its most relevant purpose. Other than that, various services and endpoints that works on the container of PGMaster provide the data Grafana shows. As understood, the application PGMaster is also containerized. The content of this container will also be mentioned later on the document.
- **WeTTY:** WeTTy (Web + TTY) is an npm module that makes it possible to have terminal access in browser over HTTP(S). As mentioned before, PGManager is intended to offer necessary tools that would help advanced users to perform various actions on CLI. We have decided that using a library for this purpose would be beneficial over implementing our own module. It basically utilizes WebSocket protocol to establish an SSH connection. In PGManager, we intend to offer access to PG hosts and containers using this library.
- **PG-Web:** This is the application I have implemented from the beginning using Typescript and Reactjs. It basically provides a GUI to PGManager users where they can perform various actions such as creating and initializing PGContainers, taking backups and restoring them, monitoring clusters and their nodes' status etc. PGMaster has over 100 such operations. While most of them needs only one parameter to be provided, some of them require more sophisticated data to be provided. For sake of simplicity, only some portions of these operations will be explained on the following sections of the document.

These four modules are the main "pillar stones" of a PGMaster ecosystem. Organizations that have systems with multiple databases which needs to deal with high volumes of transactions while maintaining high availability and also security and integrity of their data may unfortunately not have admins that are capable of performing operations that were mentioned above trivially. These modules intend to offer tools that are necessary to maintain, operate and administrate these systems whether they have advanced database administration skills or not.

I have mentioned that PGMaster have different entities. I will be listing what they are and what purpose do they serve below:

- **PGContainer:**

2.1.1 Data

The data that circulates on PGMaster could be divided into two:

- **Entity Data:**
- **Operation Results:**

2.1.2 Design Constraints

2.1.3 Modules and Interfaces

2.1.4 Documentation

Having had an introduction to software requirements specification (SRS) and software development lifecycle in general in our Software Engineering course, I usually try and understand the needs of any development activity prior to actual work. Aiming to follow the same trend here, I must admit that I was mildly disappointed, since although we didn't have a strict timeline to deliver the proof of concept (PoC) version of the application, the project was solely maintained and developed by one person. This should have -expectedly- obscured the need for any form of documentation.

I have actively communicated with my supervisor (Koray) and colleague (Derviş) on the subject and I was given permission (and responsibility) to write SRS for the project. I did not start writing the document right away, because it took me a while to fully grasp the overall needs and constraints of the system and the rest ofthe requirements naturally needs further discussion with stakeholders and the potential users. I still had chance to write some of the document by myself and it will be attached to the report. Final version of the SRS is expected to be completed after it is presented to The Ministry of Health.

2.1.5 Documentation

2.2 Design Phase

2.3 Implementation Phase

2.4 Testing Phase

3 Organization

3.1 Organization and Structure

3.2 Methodologies and Strategies Used in the Company

4 Conclusion

4.1 Appendices