

Repota

A Service Report Application

John Shields

February 2021



Department of Computer Science and Applied Physics Galway-Mayo Institute of Technology

B.S.c. (Hons) in Software Development

Applied Project and Minor Dissertation

Advised by: Andrew Beatty

Contents

1 Introduction 1

1.1 GitHub Repository URLs 1

1.2 Project Introduction 1

2 Methodology 3

2.1 Project Scope & Goal 3

3 Technology Review 3

4 System Design 3

5 System Evaluation 3

6 Conclusion 3

List of Figures

1 Application Icon 2

1 Introduction

1.1 GitHub Repository URLs

- Project Dissertation: <https://github.com/johnshields/AP-MD-FYP/tree/main/dissertation>
- Repota Application: <https://github.com/johnshields/Repota-App/tree/main/Repota>
- Horton Back-end Server: <https://github.com/johnshields/Repota-App/tree/main/horton>
- Repota's Database: <https://github.com/johnshields/Repota-App/tree/main/database>

1.2 Project Introduction

This project aimed to develop a service report application titled 'Repota' that relied on modern-day frameworks and technologies. The project focused on using these frameworks and technologies to build an app designed for automobile technicians to fill out service reports for auto dealerships and rental companies. The reports are done in a step by step process. Technicians can not only create their reports. They can view, edit, delete and export them to a PDF. These reports are usually done on paper or excel. Some similar apps have a similar idea, but it is almost as if they are trying to do too much for what they are supposed to be. With that said, there is no app that does just this service. This app is a necessity for technicians for their day-to-day work.

This project aimed to create a sophisticated front-end framework to a quick back-end that would be connected to a stable and strong database. These technologies choices that were researched and implemented are as follows: Angular and Ionic for the front-end, Golang for the back-end and MySQL for the database. The back-end and database are hosted from a Linux Ubuntu Server. Before this project, Golang (Go) and a Linux Ubuntu Server were utterly new. Go's first version being released in 2012, meaning it is a relatively new language.

- write about the final product and how it came to be...

Figure 1: Application Icon



2 Methodology

2.1 Project Scope & Goal

The project scope had to have something along the lines of a system with several functioning parts that worked with as many new technologies as possible. Original ideas for this project were initially the likes of Desktop Application Development and Game Development. These possible routes for the project were researched with the focus of on User Interface and experience. One of these ideas was a desktop application for people to send curriculum vitae (CV) to companies that their CV related to. On the app the user would upload their CV. That CV would be then sent through a system that would read it, pick out the keywords, and send it to companies looking for future employees in that field. At first, this was a great concept, but constructing an app for with was an extensive scope as it would have to have either a vast database or some API to get/pull all these companies from different sources.

Another initial idea was to develop a fully-fledged Open World Game in Unity. This idea was drastically different from the others. Old noir films and parallel universes inspired the game. Making a mix of these two genres was quite exciting and meant that this open-world would have to be huge. This idea was a favorite. Unfortunately, it had to be forfeited as this project was to be done alone. This did not seem to be achievable as many more people (a full team) would be needed to develop a game of this size. If one person were to develop a game of this size, they would need a lot more time than is allocated for the project, especially balancing it with other modules.

The chosen idea was settled to be a Service Report Web Application. This app was originally focused on being just a desktop app, but it was believed that this app would be very suitable to work on a tablet, so a mobile phone. So, therefore, the Ionic and Angular framework was chosen for the front-end. The app needed to be trusted with work-related data. Therefore, the desktop app focus was not ideal as the app will not ask for permission to access data associated with privacy settings.

This app would need a quick back-end, so many technologies such as Node.js, Python, Java and Golang were heavily looked into. The back-end was narrowed down to use Google's language Golang (Go). This language is relatively new compared to other languages and had not been covered in the course modules. This means there are not many libraries, meaning that there are only a number of ways to go about writing the code. The uniqueness of Go's compiler is that it is written in Go. Obviously, the very first compiler was not written in Go it was written in C. Go is capable of processing big files in seconds. It has been tested that Go is able to extract logs from a log file in around 25 seconds. This seemed to be the ideal back-end. Therefore it was chosen.

A reliable and stable database was essential as this project purpose was to be used in a working environment meaning it would have to store a large quantity of data that needed to be accessed quickly and be able to add and update the data freely. MySQL was chosen for the database as it is very known for these mentioned qualities.

3 Technology Review

4 System Design

5 System Evaluation

6 Conclusion