Table of Contents

List of Figures

List of Tables

CHAPTER 1
INTRODUCTION

This chapter clarifies the project that the author worked on, including the background story and problem that motivate the start of the project. Moreover, the scopes of work, objectives, vision and mission of this project are discussed in this chapter.

1.1 Background

In the 19th century, the premodern era, oil and gas were mainly used as a source of lubricant, medical use, source of light, and energy source [1]. At present, oil and gas have been the center of modern industrial society and the major geopolitical objective and majority of energy sources for nations for centuries [2]. Both crude oil and natural gas are used as the source and material of extensive types of chemicals and materials, they are the majority and dominant source of food production, pesticides, and transportation fuels [1].

Several countries, including Indonesia, are greatly reliant on the use of oil and gas for the growth of the national economy as well as the industry [3]. It is one of Indonesia's economic pillars, contributing 21.09% to Indonesia's gross domestic product (GDP) from oil exports alone in 2004 according to the Organization of Petroleum Exporting Countries (OPEC) [2]. Pertamina, which is the state-owned oil-and-gas industries is involved in the government's oil and gas exploration and production activities as well as help improve the

Indonesian GDP by contributing to more than 60% of Indonesia's GDP [3]. Furthermore, they also provide more quality products to meet their consumer demands [3].

In order to help the oil and gas industry to be more productive, there are many existing dashboard-based software applications, such as, Virtual Data Room (VDR). VDR is an online archive of oil and gas resources [4]. A VDR can help the user to analyze their oil and gas production data. These applications include Lynx and INTviewer [5] [6]. INTviewer is a platform that offers users the ability to check their seismic data, geospatial integrity, and process their datasets, which could cost up to $4,000/person a year [5]. On the other hand, Lynx is a platform that offers access to geophysical and geographical information systems (GIS) while also offering the 2D and 3D seismic viewers that cost at least €250 a year for each user [6]. However, while this software is useful and benefits the oil and gas industries, the expenses needed to be able to use the software are expensive. According to the product owner, there are also some oil and gas companies in Indonesia who requested an application at a cheaper price and customized features. Therefore, the main goal of this project is to develop a VDR that has customized features at a lower cost using open-source libraries. In addition, the proposed VDR application will be developed as a website application as part of a requirement from the authors' customer, for example, PT. Geodwipa Teknika Nusantara (GTN) [7].

1.2 Scope

The scope of the problem, especially the solution that the author team thought of, and the scope of work and responsibility that the author had, and the task the author team had as a group are described in this chapter.

1.2.1 Scope of the Problem and Solution

The main problem of this project is the high cost of the software with preferable and useful features. To solve the problem, which was to develop a VDR website application with various features at a lower cost, the solution that the author discussed with their team was to make use of open-source libraries along with hand-picking essential features from the existing software stated previously to be used as well as developing custom features requested the author's customer.

The VDR website application's features are based on similar existing software applications, Schlumberger's ProSource Front Office [8]. The website application referenced the data visualization support of ProSource Front Office in the form of lines, bars, bubbles, and pie charts and their integration with ArcGIS Arcmap to perform their geospatial analysis [9]. Aside from visualizing, the website application will consist of the following features for the user, such as:

- upload and store the files,
- delete and download the files,
- viewing map,
- viewing data using a predictive model.

1.2.2 Scope of Work

The author's responsibility in this VDR website application project was the frontend of the website, the design and user interfaces (UI) / user experience (UX), and the business logic of the frontend development. In addition, the author is in charge of making one of

the visualizations, the charts, for the oil and gas. To ensure that the website application is running smoothly and as intended, testing is also required during the creation of the website application.

Meanwhile, the responsibility as a team for this project was to create the different kinds of visualizations and implement the various features and functions the author's team had made to the VDR website application. The members of the team had different responsibilities and importance to successfully finish this project as presented in Table 1.1.

Table 1.1 Scope of Work of the Team

Name	Role
Vicky Vanessa	Designing the UI/UX of the front end of the website application, visualizing the data of oil and gas, and doing the testing, along with being the Scrum master.
Elizabeth Chan	Design the frontend of the proposed VDR application that uses GIS, as well as the testing and act as Scrum team.
Kotrakona Harinatha Sreeya Reddy	Collecting and processing data, and using them to develop predictive models, visualizing them through diagrams such as charts while doing the data analytics and act as Scrum team.

1.3 Aim and Benefits

Apart from solving the problem stated in the previous section, the primary aim of developing this VDR website application is to help the oil and gas industry, through visualizing the volume of oil and gas over time along with the visualization of the reserve resources. In other words, helping the oil and gas industries to discover more profitable areas of resources. Additionally, this VDR website application can be used to obtain or derive latent information from abundant and meaningful production data. Moreover, this VDR website application can make a predictive model through the production data.

Furthermore, the author and the team wish to increase the local services in the oil and gas industries to aid the industries in attaining more profits and indirectly increase the national income. Some local companies would like to have their own VDR that match their needs. Aside from that, the author team wishes for the VDR website application to be known internationally. Apart from providing advantages for the oil and gas industries, the application is also useful to aid the engineers in comprehending sophisticated data and to gain a better understanding of how the data could be used.

Aside from the aim of the VDR website application in general, the author’s aim, based on Table 1.1, is to design the frontend of the VDR website application. In the frontend, the author aims to build and design the visualization of oil, gas, and water data, and the file management. Moreover, the author will perform the testing to ensure that the client side's features and functions will work as intended.

The customer requested for an application that could be run in multiple Operating System (OS) with minimal installation efforts. Thus, the author and team decided to develop a website application as it is an OS agnostic. Web browsers exist in any OS thus, the application will exist regardless of the OS. The application will also be cheaper than the existing software briefly described in Section 1.1 as it will apply Open Source Software (OSS) in the application.

1.4 Structure

This thesis consists of 5 chapters as defined further in this section

Chapter 1: Introduction

In chapter 1, the topic is introduced along with the description of the author's scope, objective, and aim for this project.

Chapter 2: Theoretical Foundation

In chapter 2, the project theories, why it is carried out in a certain way, how the project will be made, as well as what it is the author used to develop the project is described and explained.

Chapter 3: Problem Analysis

In chapter 3, the VDR website application will be compared to similar existing applications. Furthermore, the solution to the problem described previously will be discussed, especially the reason on why the author choose to use a particular tools or frameworks.

Chapter 4: Solution Design

In chapter 4, the features and functions will be explained, along with the flow of the system for the user. In addition, the design of the website application and planning on how the testing will be performed will be further discussed.

Chapter 5: Implementation

In chapter 5, how the system will be implemented and put together along with other features of the VDR website application will be explained.

Chapter 6: Conclusion

In chapter 6, the final chapter, will conclude at the end of the project.

CHAPTER 2
THEORETICAL FOUNDATION

This chapter discusses the theories that the author uses to develop and design this project. It highlights how the author and the team design the project together, what theory and concept they use to aid in building the project. Aside from that, this chapter also describes the reasoning behind the theory thoroughly throughout the chapter. Starting from the concept of VDR website application, OSS, Software Development Life Cycle (SDLC) to testing. Specifically on the approach, sources, limitations, and the frameworks of the frontend and the development that the author uses to build the project.

2.1 Virtual Data Room

Data room is a promising valuable asset of intelligence about oil and gas resources that are available for sale, undoubtedly used by the oil and gas companies who wish to distribute their share of assets [4]. Interested buyers' Mergers and Acquisitions (M&A) team would visit the data room to analyze and inspect the data placed and arranged by the seller, then decide whether or not the assets are worth purchasing [4]. Data rooms, with an abundant amount of confidential oil and gas information sources, have different forms, the Physical Data Room (PDR), VDR, or both combined [4]. Visiting a data room depends on the seller, whether or not they allow it. Most often than not, they will not allow a second look once they decided to close it [4]. Thus, both sellers and buyers must be conscientious assure everything is going well [4].

PDR is a closely monitored room to ensure the secrecy of the data placed by the seller or their representatives [4]. It has a lot of disadvantages as it is expensive, time-consuming, as well as an inconvenient geographical location [4]. VDR is built as an alternative to the cumbersome way of PDR. It also has replaced the majority of PDR [4]. VDR is a controlled-access website that allows loads of documents to be uploaded for a limitless amount of users to analyze documents [4]. VDR is available and accessible continuously non-stop and, in some circumstances, should be easy and quick to set up [4]. Any information on the VDR could be updated, modified, or deleted at any time [4].

VDR has a lot of benefits over the traditional PDR for the clients [4]. VDR offers information and documents at a faster speed, lower fees, as well as better efficiency [4]. Moreover, since the platform is online, it allows access no matter the current location and region of the clients [4]. Aside from that, by using VDR, clients do not have to go their way to send their teams to an inconvenient location of a PDR [4]. At times, it could be complicated for the clients to send the right team to the PDR, but the existence of VDR saves that complication [4]. Based on Schlumberger, there are three possible features of VDR which are:

- full series of petrotechnical solutions,
- generate and download reports as well as other documentation,
- available to access from anywhere around the world regardless of OS and devices [10].

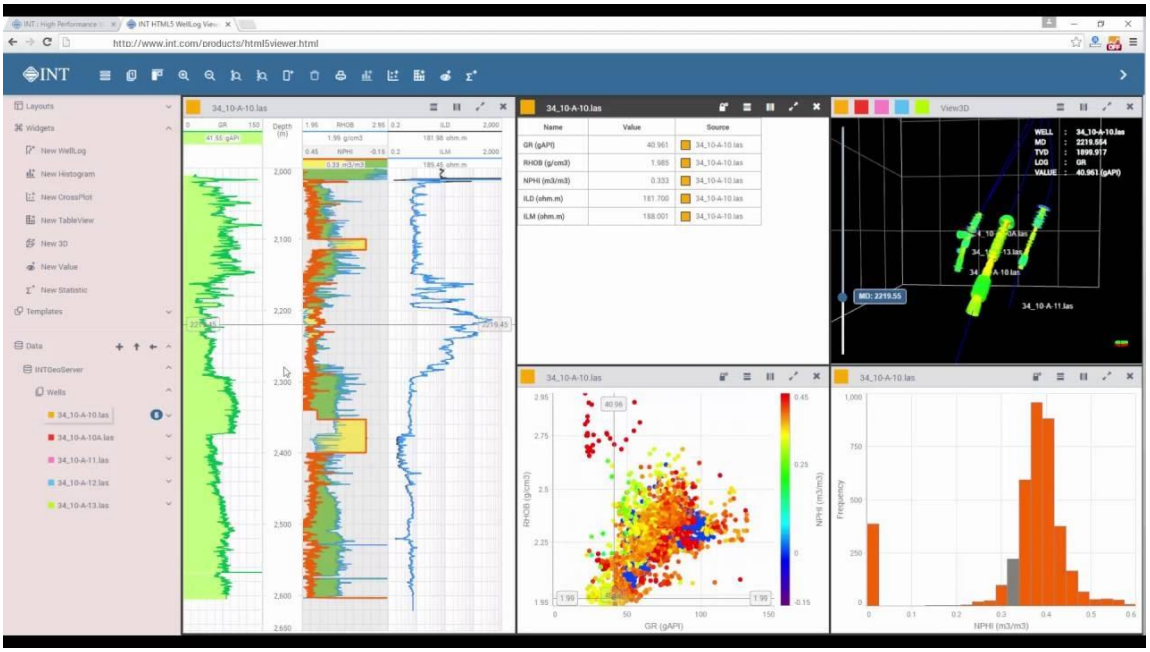

Figure 2.1 Dashboard in Virtual Data Room

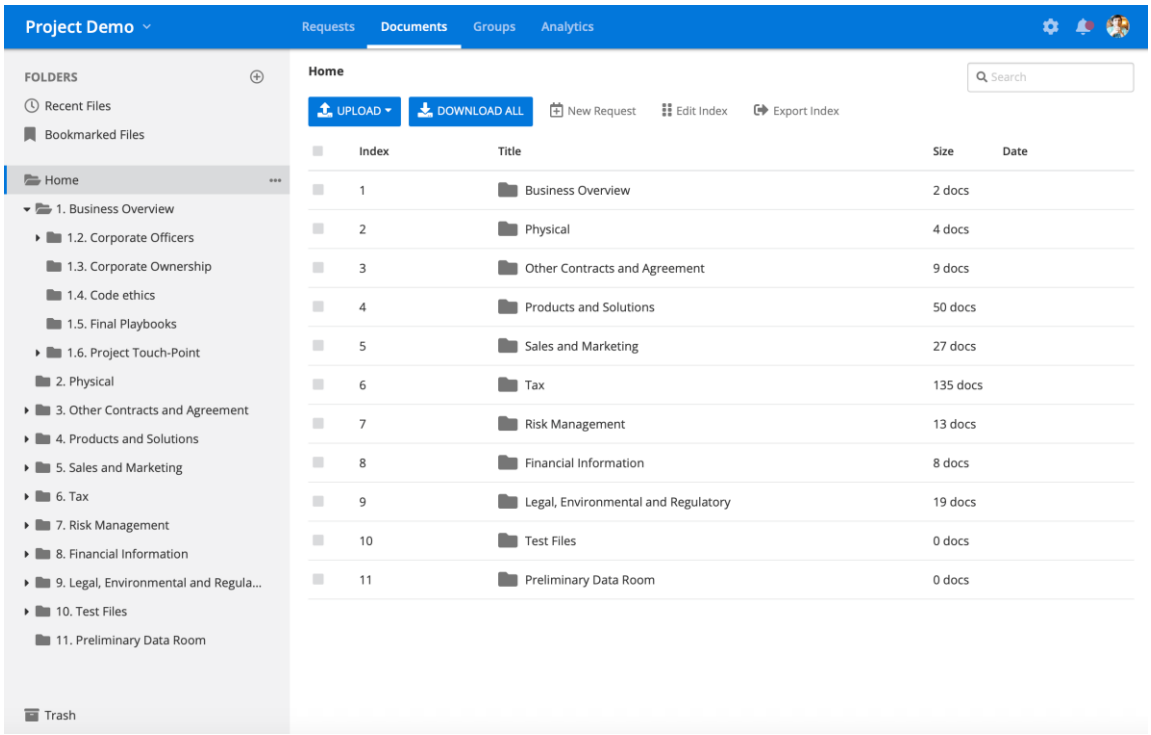

Figure 2.2 File Management in Virtual Data Room [11]

Figure 2.1 and Figure 2.2 show examples of some of the features available in a VDR. Figure 2.1 shows an example of a data visualization of oil and gas data, while Figure 2.2 shows the file management features in a VDR.

2.2 Website Application

VDR will be developed in the form of a website application. The website uses the World Wide Web (WWW), which was introduced to assist in achieving access to information from any source in a consistent and convenient method [12]. Introduced in the 1990s by the Conseil Européen pour la Recherche Nucléaire (CERN) in order to accommodate the enormous amount of data and documents required to be shared with other scientists [12]. The method adopted is by using HyperText Transfer Protocol (HTTP) which was designed to allow one computer to request data and documents from the server computer [12]. This helps users to access the documents through the computer [12]. WWW was perceived as a large archive of information that allows access to numerous amount of users [12]. Website application refers to the program that could return the page of data from the server to return its result to the user, some describe it as a combination of hypermedia and information system [12]. At present, website application is a powerful platform that supports individuals, SMEs, communities of users, and large corporate businesses [12]. In 1999, Fratenali stated the requirements of a website application which are:

- the need to handle both structured and non-structured data, referring to the database record and multimedia items, respectively,
- navigational interface in order to assist exploratory access,
- impressive graphical quality, and

- customizable and the dynamic adaptive content structure, navigation, and presentation style [12].

A website is a combination of connected and related website pages which occupied on a single server are available for a variety of categories and usages [12]. It continues to rapidly evolve while maintaining the basic concepts of WWW which are Uniform Resource Application (URL), HyperText Markup Language (HTML), and HTTP [12]. URL is a naming system that prescribes the way objects could be identified and located as well as searchable based on their attributes or names [12]. HTML is a document on the web in which the content can be displayed and linked to other documents to achieve enhanced visual presentation [12]. Meanwhile, HTTP is a communication of a request and reply protocol that contains the most used basic operations of GET, PUSH, PUT, DELETE which could be requested to the URL [12]. Website applications use a similar concept to that of software, they have a UI, may contain and manage a vast amount of data on the server, as well as the Model View Controller (MVC) design pattern [12]. In fact, several frameworks are developed in support of the design and development of the MVC website application [12].

Aside from that, research on modern website applications pointed out that there are several standard features in website applications which are:

- search,
- tagging,
- user participation, and
- UI and collaboration [12].

Searching and tagging have been a mechanism to assist users in finding the information that they need amongst the abundance amount of information on a website [12]. User participation is to create and design the structure of the data [12]. While UI and collaboration primarily exist in a messaging system, for instance, Google Mail and Skype, which would notify the user in real-time when someone in their contact is online [12].

In a website application, a three-tiered architecture model is widely used. It is an architecture with three different levels, as presented in Figure 2.3 [13]. The principle of this architecture model is that business logic is the vital core of the entire application system [13]. According to Figure 2.3, if the process is modified, the tiers would need to be changed correspondingly [13]. Similarly, if the data structure in the storage is changed, every tier would be changed to adjust to it [13]. As mentioned in Section 1.2, the frontend development is part of the scope. Thus, Section 2.2.1 will delve further into the presentation tier of this architecture.

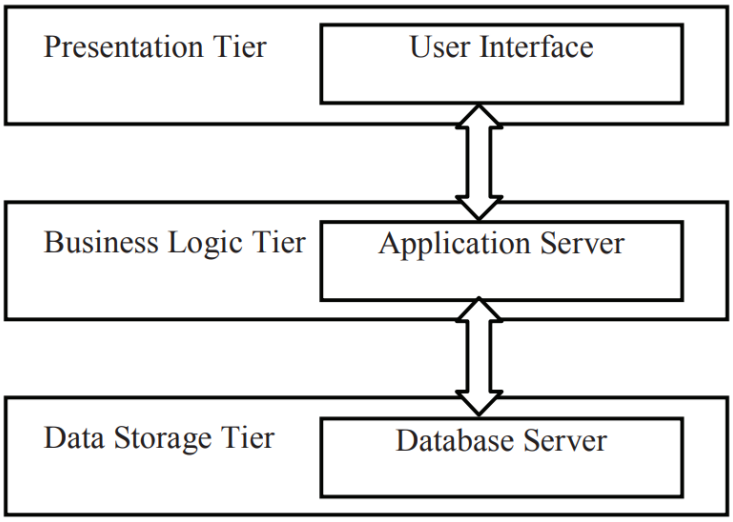

Figure 2.3 Three-tiered Architecture Model

2.2.1 Frontend Development

Website technology has expanded over the years along with the development of the internet, especially the HTML that leads the frontend development [14]. Frontend development has evolved simultaneously with the demand of business needs [15]. The traditional model which used JQuery was not enough to fully develop the website frontend that meets the interest of the current era of website development [15]. JQuery is an open-source JavaScript library to help the handling of HTML Document Object Model (DOM), ajax, and Cascading Style Sheet (CSS) [15]. Although it is still widely used in website development, more complex libraries that fit the business needs, for instance, Single Page Application (SPA) and Progressive Web Application (PWA) [15]. Developers frequently and mainly use frontend frameworks when developing a large and sophisticated website application, it allows them to reuse repetitive code, saving time in designing and developing the application [16]. There are numerous JavaScript frameworks and libraries for SPA, among those, the most popular frontend frameworks are Angular, React, and Vue [14] [17]. SPA is a website interface consisting of individual components that can be easily updated or replaced individually [16] [17]. Thus, preventing a reload of the page for every action the user made while loading dynamically [16] [17].

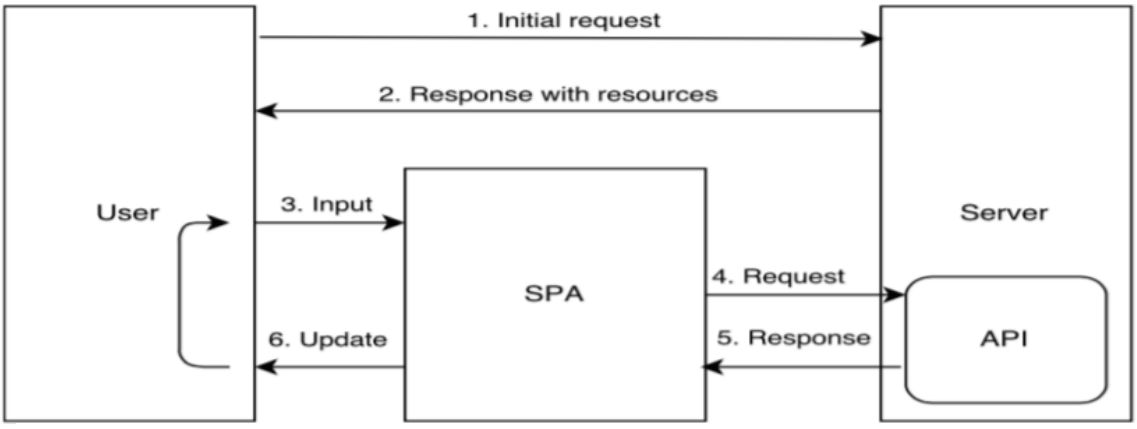

Figure 2.4 Communication between the user and a web browser with Single Page Application [16]

Figure 2.4 presents the communication between the user, SPA, and the backend server. The initial request is sent through HTTP by the user's web browser [16]. The request sent to the backend server is through JavaScript Object Notation (JSON), the server will send back JSON data in response [16]. As a result of the communication process described, SPA has a quick response/render time for the user [16]. Moreover, for the developer, it is advantageous for them as the presentation side and logic are separated, allowing them to be able to work and update on them separately [16]. Aside from that, the data transmission, sending, and receiving to and from the server are faster [16].

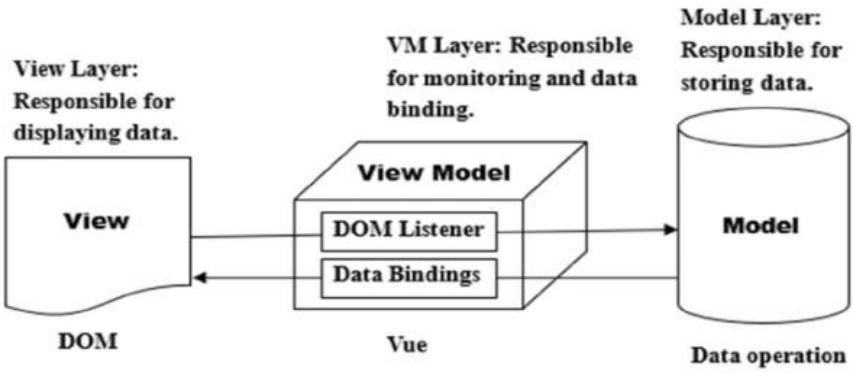

Figure 2.5 Model View View-Model Architecture Diagram of Vue [15]

As mentioned previously, Vue is one of the popular open-source JavaScript ES6 frontend frameworks [16]. Creating a SPA website with comprehensive and high-performance in developing an interactive interface [16]. Vue uses Model View View-Model (MVVM) design pattern that supports its data-driving and component-based development [15]. MVVM architecture is unique from the conventional operation of DOM nodes, as shown in Figure 2.5 [15]. Vue is more focused on the view-model aspect of MVVM, being bonded to both directions responsively [15]. The bindings made Vue respond

asynchronously to updates from the data of the user for the DOM to be synchronized with the updates [15] [18].

Vue is an excellent choice as a frontend framework for the reason that it has significant advantages over other frontend frameworks such as:

- greater efficiency, usability, user experience, fast and light,
- rendering broad modular models without additional time,
- great reactivity between HTML and JavaScript code, and
- DOM is modified accordingly to any updates from the user [17].

All these advantages are what makes it great as a SPA, furthermore, Vue is adaptable to the user, usable for users with poor connection, as well as great for computers with insufficient resources [17].

2.2.2 Human-Computer Interaction

Understanding Human-Computer Interaction (HCI) is essential to understanding the effectiveness of a UI. HCI is a term for the interaction between a human user and a computer system to ensure that the interaction between the two parties is effective, efficient, and satisfies the user [19]. HCI term emerged in 1980 and has been a successful technological and scientific undertaking by achieving an effective integration of software engineering with human factors or ergonomics of computing systems by using the concept of cognitive science [20] [21]. Human factors are science as well as a field of engineering related to human capabilities, limitations, and performance as well as system design [21]. Prioritizing the efficiency, comfort, and pleasantness for the human who uses the system

[21]. HCI is quite broad of a subject as it interests expertise from cognitive psychology, sociology, linguistics, cognitive science, as well as computer science [21].

In relation to computer science, HCIs connected with the Graphical User Interface (GUI) of software or an application [21]. GUI entered the mainstream in 1984 to replace the text-based Command Line Interfaces (CLI) to be more user-friendly and enable users to be able to control what and how they see [21] [22] Human Factors Engineering by Christopher D. Wickens in 1998 introduced 13 principles that have been perceived as an outstanding guide for designers in the industry [23]. It is further categorized into four different principle categories which are perceptual principles, mental model principles, principle-based on attention, as well as memory principles [23] which are separated as

The perceptual principles refer to:

- legible display,
- avoiding absolute judgment limits,
- top-down processing,
- redundancy gain, and
- avoiding similarities [23].

In this category of principles described that objects in the design should be clear to see and easy to read signals to the users should be according to the UX, signals that could be understood by presenting in an alternative form, as well as removing similar features [23].

Mental model principles are:

- pictorial realism,
- moving part [23].

This category explains that a character needs to represent what it seems, for instance, a higher bar in a bar graph indicates a higher value, following that, moving objects should need to reflect the change in value [23].

Principles-based on attention as:

- minimizing interaction cost,
- compatibility principle, and
- multiple resources [23].

In this category, the context means that regularly used components should be located efficiently to minimize the user's time and effort, the visual presentation should have consistency, along with using information that would help the user process signals easily such as using red colour to present a warning to the user [23].

And memory principles referring to:

- replacing memory with visual information,
- predictive aiding, and
- consistency [23].

This category describes that a good design should not have the user remember important information in their memory, using perceptual than cognitive to present the user's progress and usehe same consistency between versions [23].

2.2.3 The 8 Golden Rules

According to Ben Shneiderman, there are 8 golden rules that are relevant in the majority of interactive systems which are:

1. strive for consistency,
2. seek universal usability,

3. offer informative feedback,
4. design dialogs to yield closure,
5. prevent errors,
6. permit easy reversal of actions,
7. keep users in control, and
8. reduce short-term memory load [24].

To elaborate on the meaning of the 8 golden rules, striving for consistency refers to consistency in terms of the design of the interfaces, the whole system should have the same menu, colour, layouts, and fonts [24]. Seek universal usability means considering the different and diverse users that will use the system, the design of the system should analyze the target user, international variation, as well as the differences in technology [24]. Offer informative feedback for every action the user made in the system, there should be a response from the interface, a change in visual presentation in the object they interacted with would conveniently show the changes explicitly to the user [24]. Design dialogs to yield closure refer to a sequence or a set of processes and actions that have a beginning and an end as an indicator to organize and prepare for the following sequence of action [24]. Prevent errors indicate creating a design that would help prevent users from making errors in the interface and providing a simple step of recovery in the event that the users made an error [24]. Permit easy reversal action means users should have access to reverse their actions to offer them in exploring the available options [24]. Keep the user in control as users would want and hope to be the one in charge of the interface and for the interface to be able to respond to them, not wishing for any unexpected actions from the interface [24]. And reduce short-term memory load since there are limits to how much humans can process information in short-term memory [24]. The system of the design

should not require the user to memorize the information that would be needed in another display in the system [24]. These 8 rules could be further interpreted and modified in different environments, however, it provides a good start for designers [24].

2.2.4 Material Design

Material is an adaptable system consisting of guidelines, components, as well as tools to support a good practice of UI design made by Google [25]. Material components are available for Android, iOS, web, and Flutter to help developers develop their UI in accordance with the guidelines provided by Material [25] [26]. Material offers customizable UI components, enabling developers to build an appealing and functional UX [25] [26]. It is inspired by the real physical world and its material and textures, for instance, how light is reflected and cast their shadow, specifically using paper and ink as the medium [25]. Material components cover a range of the essentials of UI, for instance:

- display,
- navigation,
- actions,
- input, and
- communication [25].

Material Design in the web is available on SPA frameworks, such as Angular Material for Angular, MUI for React, as well as Vuetify for Vue.

Vuetify is a UI framework built for Vuejs, and unlike Material Design on other frameworks, Vuetify is designed from scratch to make it easy for learning [27]. It has active development as it is being patched frequently, seeking out issues in the community,

having more bug fixes, and got enhanced often [27]. Vuetify has a massive community to support developers and collaborate with them [27].

2.4 Open Source Software

VueJs framework and the libraries available for Vue is part of the OSS. OSS is a user-driven, self-organizing team of contributors formed through online interaction, who develops libraries and codes that are available under a license that could be reused, modified, improved by the communities [28] [29] [30]. Over the emanating development over the last decades, OSS has been becoming important as it is valuable to significant information technology (IT) companies [28]. For instance, Google created a platform to publicly host OSS projects for the community in 2005 named Google Code [28]. Figure 2.6 presents the difference between proprietary software with open source code and closed source code. While the corporate or individual could choose to publicize the code, according to Figure 2.6, software with open source could offer more flexibility and adaptability as it is in development by the community as it could be modified and fixed repeatedly [28].

Both users and developers are perchance licensed to use and modify the code and distribute the modification and improvements they made at their discretion [28]. However, according to Open Source Initiative (OSI) and Free Software Foundation (FSF), the free source of software does not entail that it is free of cost [28]. Instead, it implies the freedom that the users and developers could use the code or program for [28]:

- executing the program for one own's purpose,
- study how the program operates and modify it to one own's needs,

- redistributing copies of the original or the modified, improvised program, and
- improving the program and publicize it for the community.

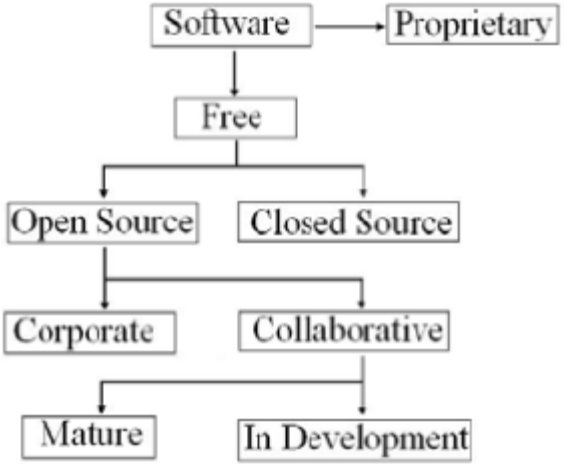

Figure 2.6 Software Taxonomy [28]

1. **Free redistribution:** the software to be available for distribution without payment.
2. **Source code:** the soft to be distributed with the source or well-publicized access to it.
3. **Derived works:** the license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
4. **Integrity of the author's source code:** distribution of "patch files" used to recreate derived works to be permitted.
5. **No discrimination against persons or groups:** the license must not discriminate against any person or group of persons.
6. **No discrimination against fields of endeavour:** for example, it may not restrict the program from being used in a business, or from being used for genetic research.
7. **Distribution of license:** the rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
8. **License must not be specific to a product:** license rights must not depend on the software being distributed with the other specific software.
9. **License must not restrict other software:** the license must not place restrictions on other software that is distributed along with the licensed software.
10. **License must be technology-neutral:** No provision of the license may be predicated on any individual technology or style of interface.

Figure 2.7 Open Source Delivery [28]

According to Open Source Definition (OSD), a document published by OSI, there are ten criteria to decide whether a license of software is open-source which is shown in Figure 2.7 [28] [31].

There are three critical benefits among the numerous benefits that could be found along with the emergence of OSS [29]. The first benefit is the fact that it reduces the cost for an IT company exponentially, the cost of development, and the innovation cost [29]. As it has the contribution from proficient developers in the community throughout the world [29]. Second of all, OSS can assist organizations, specifically small-medium enterprises (SMEs) and public institutes, to implement IT systems into their business process [29]. SMEs and public institutes may not be able to implement the IT systems due to the high cost of implementation and their limited budget [29]. Last but not the least, the information and knowledge from getting involved with the OSS project [29]. It is valuable to those who wish to increase their proficiency in programing language as they can learn through practice [29].

An abundant amount of individuals has involved themselves with OSS projects for their purpose and goal [29]. There are two key motivations as to why they are willingly contributing to the OSS projects, which are regarded as intrinsic motivation and extrinsic motivation [29]. Intrinsic motivation is driven by an individual interest or enjoyment of the project or task instead of relying on external pressure or rewards at the end of the project or task [29]. The research found that 54% of the individuals, especially young adults find enjoyment and satisfaction that they have during the programming and involvement in OSS projects [29] [32]. Their creativity and ideas that are implemented in

the OSS projects, conferred a great sense of accomplishment motivated them to contribute to it [29]. Contrary to intrinsic motivation, extrinsic motivation is driven by an outcome such as profits and reputation [29]. Strictly speaking, the motivation will disappear once those rewards are withdrawn [29]. Some large OSS projects and contributors have been analyzed that the contributions have been involved with the pursuit of rewards and reputations [29]. Aside from those, research also found that a sense of reciprocity and job opportunities are the other factors of extrinsic motivation [29]. Some individuals would aid the OSS community by responding to the question in the community for them to be able to obtain assistance from others when required in the future [29]. Moreover, some IT companies would like to recruit capable programmers from the OSS communities which encourages them to be involved in the OSS projects [29].

All software, including OSS, is subjected to copyright law [31]. The copyright holder is an individual or organization that has control over the work, they could give permission to other individuals to use, modify, and redistribute their software according to specific provisions through licensing [31]. OSS license is different than the general commercial license as it grants more access to the individual, allowing access to the source code and the right to modify it [31]. Copyleft, enforced by copyright law, is a term established by the free software community for a license condition to ensure that all versions of modification made to the source code can be copied, modified, and redistributed similar to how it is with the original [31]. There are frequently used open-source licenses for instance the Berkeley System Distribution (BSD) license, the Massachusetts Institute of Technology (MIT) license, and Apache. The BSD license is one of the least restricted as well as most acknowledged open-source licenses [31]. It allows redistribution of source

code, modified or not,s long as the work has its copyright notice regarding disclaimer and limitation on liability found in the license [31]. MIT license is quite identical with BSD license as it authorizes the reuse of open source code if the license includes proprietary software in their term [31]. Moreover, it also allows the use of the copyright holder's name for the promotion of the software [31]. Vuejs and vuetify are examples of an OSS project released under the MIT license [33]. Apache is similar to BSD and MIT licenses, as it grants the use of software without the obligation to redistribute the code of any version modification of the software [31]. Aside from that, it supports the clause of patent licensing and termination [31]. Understanding the difference between each licensing is essential in order to use OSS.

2.5 Software Development Life Cycle

The concept of SDLC could be used to develop a website application. Software Development Life Cycle which is commonly known as SDLC is a framework or a model that describes a continuous process of how the project started the deployment of the product [34]. Moreover, about planning on how to develop the product [34]. It is a cycle that will help arrange the activities to be executed during the process of development [34]. SDLC approach to complete the development is by using a step-by-step approach, and the main phases are:

- specification,
- design,
- validation, and
- evolution [34].

The specification phases focus on understanding and comprehending the problem and what is required by the customer or the user in the application [34]. Designing is involved in planning out a solution for the problem described in the specification and designing how the application will be executed and flow [34]. The validation phase is in which the application is evaluated and tested whether or not it meets customer's expectations [34]. Finally, evolution is the latter stage of the project focusing on how it will be maintained [34]. According to the SDLC methodologies and purposes, these phases are described and utilized differently. Throughout the cycle, it will enhance the understanding of the product, the problem, along with the solution for the problem, programming, as well as how to test and maintain the product that is developed [34].

Working without thorough planning and discussion will complicate the process of developing an application with a team [34]. In addition, without a framework to structure the stages of development, developers will find it challenging to develop a product that they do not have a sufficient understanding of [34]. Thus, SDLC is an advantageous cycle to have in development [34]. Aside from that, SDLC aids in developing a timeline for the project, the responsibilities of each team member, as well as to keep track of the progress made during the development [34].

There are various SDLC methodologies. A traditional SDLC methodology is most known as the waterfall model and the modern models such as iterative model, agile model, and V-shaped model. Methodology like the agile model is excellent in order to reduce the amount of error later in the deployment stage [34]. Due to the fact that revision could be made over and over again during the implementation stage [34]. Agile methodology is a

modern project management method in which the project is made and adjusted based on customer or a user requirements for the project and allows quick revision and modification to be made instantaneously [35]. The agility in completing the project is acquired by including only the process in the project [34]. It entails the exclusion of all types of processes and activities that are irrelevant and will consume more time than necessary for the project [34]. Each iteration of processes involved in the agile SDLC is as shown in Figure 2.8, it consists of small and manageable which could be completed within a couple of weeks [34].

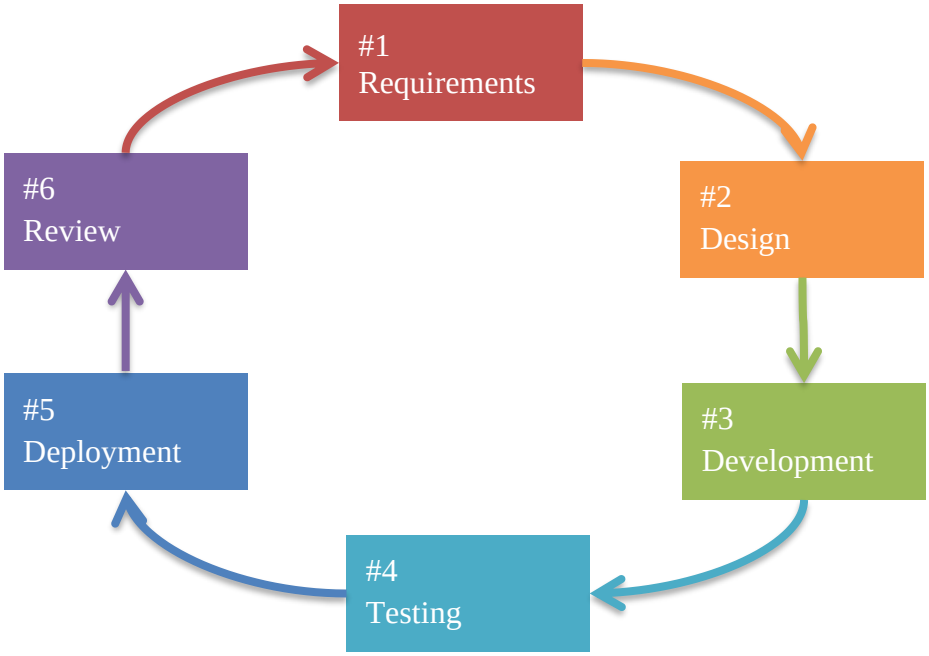

Figure 2.8 Software Development Life Cycle Agile Methodology

Each agile project regularly involves a customer or a stakeholder representative on the team [34]. At the end of each iteration, they will have to re-evaluate the progress and make some adjustments directly with the team if necessary [34] [36]. Thus, emphasizing the importance of enhancing their communication skill [34] [36]. Moreover, since more

requests to change the requirement may be made by the customer, agile relies on the implementation of the software rather than the documentation [34]. This methodology offers more efficiency and faster project completion time [34].

There are 12 principles behind agile software development according to the agile manifesto [37], which are:

- satisfy customers throughout the delivery of the software,
- accept any changes in requirements throughout the development process,
- frequently deliver working software in a short timescale,
- customer or stakeholder representatives and developers should work together throughout the project,
- provide developers the environment, support, and trust to help them finish their job,
- relay information to and from the development through face-to-face conversation,
- the main assessment of progress is a working software,
- promote sustainable development,
- enhancing agility by continuous attention to outstanding excellent and design,
- simplicity is essential,
- good architecture, requirements, and design comes from good self-organizing teams, and
- adjust and reflect periodically.

The main advantage of agile methodology as briefly described previously would be that the team would be able to reduce the overall development and building time of the entire project [34]. In addition, customers and representatives would be able to get updated often after each iteration, making feedback, revision, and adjustment would be able to get implemented as soon as possible [34].

In agile methodology, there exists a Scrum framework to help increase the agility and flexibility in developing a project [38]. It is developed as an additional framework for the purpose of becoming a development methodology as well as a management methodology for software projects [38]. In Scrum, there are three roles, a product owner (PO), Scrum Master (SM), as well as the team [38]. A PO usually consists of a project manager (PM) and a business analyst (BA) who is responsible for defining and registering the requirements and the specification of the project [38]. The team consists of developers, testers, and other relevant roles which will complete the backlog provided by the product owner [38]. Meanwhile, the Scrum master is the individual who will set the Scrum process in the project and make sure that everyone involved in the project implements the Scrum method provided [38]. The Scrum process is implemented as :

- determining the backlog,
- sprint planning,
- regular stand up meetings,
- sprint review,
- sprint retrospective [38].

The PO will be the one to prepare the backlog for the team to follow, which the SM will arrange the priority of the task [38]. The whole team needs to participate in a meeting to evaluate the backlog and assign the task or feature that the developers will need to finish by the end of the sprint for the next sprint [38]. To supervise the progress, there should be a meeting to determine what each member of the team has accomplished prior to the meeting [38]. At the end of every sprint, the task will be demonstrated to the customer or the stakeholder representative in order to show the features or progress that have been completed [38]. Thereafter, the whole team would discuss the problem they encountered while doing their task that would need to be discontinued in the next sprint [38]. By using the Scrum framework, the quality and progress of the project could be seen clearly and done more efficiently [38].

2.6 Design

Design defines the architecture of the system in the early phase of SDLC. The system structure is designed based on the requirement analysis [39]. The logic, architecture, physical design, and final design may be created in the design phase [39]. The design is used to sketch the overall picture of the system for the team [40]. To design the system, one should consider both the functional and non-functional requirements of the system [40]. Visualizing the design aid the team developing the system to picture the process of the system [40]. To visualize a system, diagrams is necessary [40].

There are various diagrams that could be used to visualize a system. One of them is Unified Modelling Language (UML). UML is used as a visual language to interpret and document a system as well as to present how the user could use the system and the behaviour [41]

[42]. There are several types of UML diagrams that developers could use such as class diagrams, activity diagrams, use case diagrams, and component diagrams [42]. The frequently used UML diagrams are:

- use case diagram,
- activity diagram,
- class diagram,
- sequence diagram,
- collaboration diagram,
- component diagram, and
- deployment diagram [41].

Each type of UML diagram has a different perspective for the customer and developer [41]. Use case diagram is one of the types of UML diagram used to aid the developers in illustrating the functionality of the system [41]. This includes the function requirements, the relationship between the ‘actors’ and the functions as well as the expected process of the system [41]. The actor is the individual who is interacting with the system [41]. This diagram shows a certain group of use cases with similar functionality.

Meanwhile, the logical design of a system could be represented by a flowchart [43]. Contrary to the use case diagram, a flowchart is a diagram that portrays a process, system, or computer algorithm [44]. This diagram is commonly used in a variety of fields, from documents, study plans, to complex processes in a comprehensible diagram [44]. A flowchart uses several shapes to define its functionalities and processes, such as

rectangles, diamonds, and ovals which are connected to one another [44]. Moreover, since flowchart can be used in a variety of forms, it is one of the most frequently used diagrams, used for both technical and non-technical fields [44]. The flowchart is used as a guide for developers to develop the system. [43]

2.7 Testing

As mentioned in Section 1.2, testing is part of the scope of the project. Testing is a phase to ensure delivery quality in the SDLC [45]. It is a significant process for quality assurance, finding a defect in the program before the program is delivered to the user or the customer [45]. Testing is planned in the early stages of SDLC, which is in the requirement gathering stage and continues to be refined as the program is being developed [46]. Not to mention, testing requires a lot of costs, in terms of finding a suitable testing case, the time it took to run those tests, and the time to fix the found errors [46]. The technique involved to evaluate the quality of the system could be categorized into 2, static and dynamic techniques [46]. The static technique is a technique that does not involve any code execution [46]. Instead, it is used to check any defects in the code, whether or not it adheres to the specifications specified on the requirement or the project documentation [46]. This technique could be used during the whole development process [46]. Contrary to the static technique, the dynamic technique involves the execution of the code and analyzing the responses and behavior to determine the existence of an error [46]. It obtains information about the program by observing the behaviour during the code execution, the methods could involve a simulation, time analysis, and prototyping [46].

There are several objectives for the testing aside from finding defects in the system, such as:

- acceptance testing,
- installation testing,
- alpha testing,
- beta testing,
- functional testing,
- non-functional testing,
- regression testing [46].

Acceptance testing confirms that the system follows the customer requirement, commonly tested by the customer themselves [46]. Installation testing validates that the system is running in the specified environment [46]. Alpha testing is tested by an internal user, which lets them explore the system themselves without any test plan involved [46]. While beta testing is similar to alpha testing, it requires an external user to test the system instead [46]. Functional testing confirms whether or not the functionality is working as specified in the requirement [46]. Meanwhile, non-functional testing tests the system's performance, reliability, functionality, and security [46]. And regression testing is a test to verify that any modification in the code does not affect other functionality that adheres to the requirements [46].

Throughout the process of SDLC, testing is utilized at different levels, which could involve a part or the whole system [46]. Based on the SDLC methodology used, the testing could be applied in different phases, each phase refers to specific requirements subjective

to different parts of the system [46]. Thus, no matter which SDLC methods are used, the team should be able to distinguish between unit, integration, system, and regression test [46]. The unit test refers to the test of a small piece of a program, usually created by a single programmer [46]. The purpose is to confirm that the feature and functionality in that unit satisfy the requirement, work, and are implemented as expected and intended [46]. Aside from that, it can also be applied to test for the interface and data structure [46]. Regarding integration testing, the purpose of this test is to find any errors that are present during the process of assimilating program components into one [46]. Although there might not be an error as an individual component, problems could arise when the components begin to interact with each other [46]. System testing tests whether a system is working properly as required when it is implemented into the intended environment [46]. Any errors revealed in this test could not be detected during a unit or integration testing, rather, it is more to the behaviour of the components when a user interacts with it [46]. Moreover, this testing also includes the test of performance, security, reliability, and recovery of the system [46]. As the interest and requirement for a significant program increase, so does the complexity of the website system, testing would be a major part of SDLC and quality assurance [47] [46].

CHAPTER 3
PROBLEM ANALYSIS

This chapter will delve into the competition of the website application briefly explained in Chapter 1. In addition, the author will also discuss how the website application is different from the existing application and the solution to the problem stated in Chapter 1. Moreover, this chapter will explain justifications for the tools that the author decided to use.

3.1 Existing Applications

As briefed in Chapter 1, there are three applications that the proposed VDR website applications are inspired from. These applications are Lynx, INTviewer, and ProSource Front Office. The following subsections will delve into the specifics of these applications.

3.1.1 EZLOG Well Log QC and Digitising

Lynx, or Lynx Information Systems, has been operating since 1989 in London and Houston, offering a variety of services for petroleum exploration industries worldwide [6]. Lynx is included due to its popularity in the petroleum industries. In EZLOG Well Log QC and Digitising, they display the raster well log image, the vector data, and the grid as depicted in Figure 3.1 [6]. Aside from the display, this software also offers the ability to edit and manipulate the curve of the data.

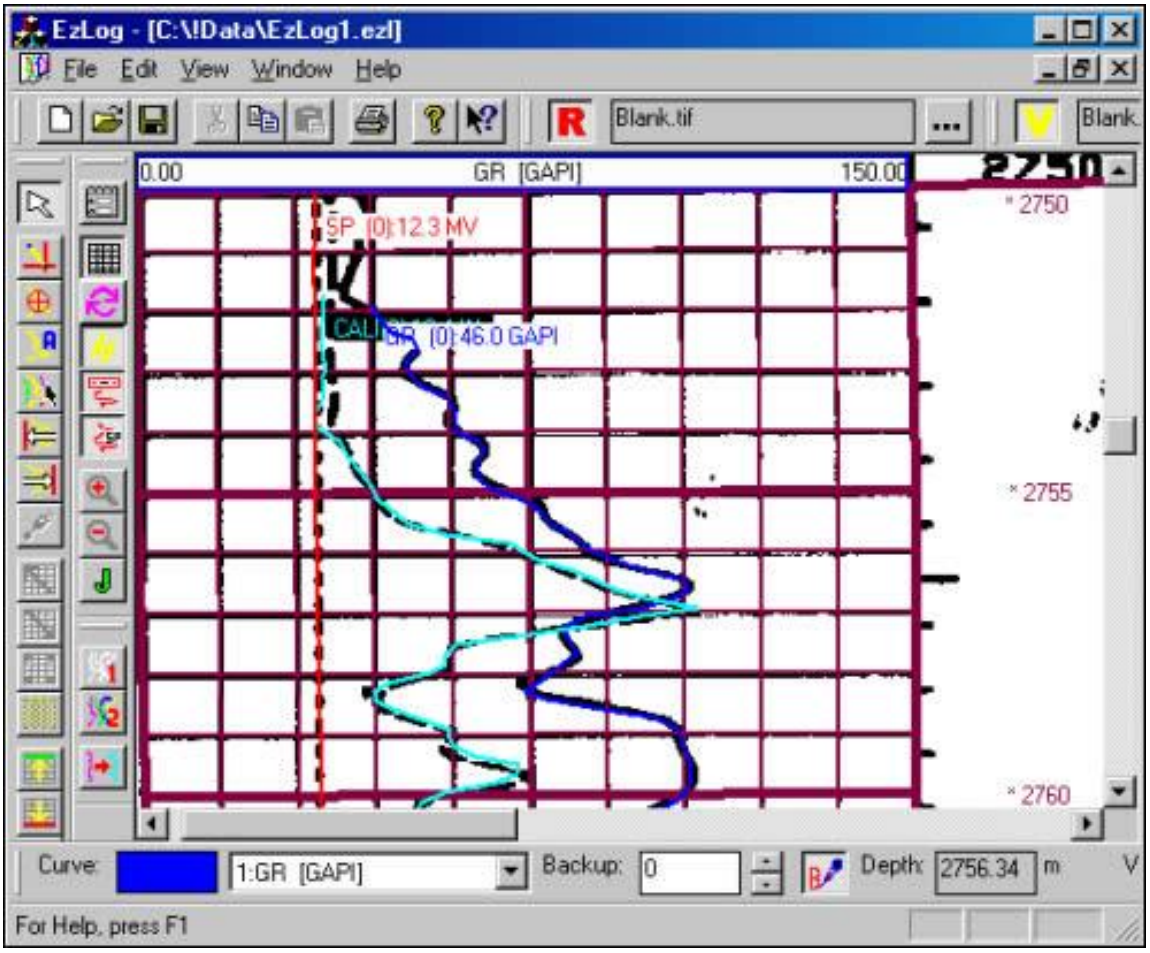

Figure 3.1 EZLOG Well Log QC and Digitising [6]

3.1.2 INTviewer

Interactive Network Technologies (INT) has been standing since 1989, creating products to visualize, monitor, as well as analyze the data from oil and gas companies [5]. INT have three products solutions which is flexible to both scientific and business industries [5]. Of these three products, the product where the proposed VDR website application might get their inspiration from is INTviewer [5]. INTviewer is a geoscience Quality Assurance (QA) / Quality Control (QC), allowing geoscientists to view seismic, well data, and analyze their dataset [5]. This application is designed to access an extensive dataset from

a desktop or cloud [5]. An example of this application can be seen in Figure 2.1. There are six main features that are available in INTviewer [5], which are:

- QC user seismic data,
- perform light processing,
- analyze user data,
- present user finding,
- customize user workflow, and
- automate user steps.

These features allow customization from the user, enabling user to use their own algorithm or automation Python scripts.

3.1.3 ProSource Front Office

Schlumberger provides the technology and services for the energy industry [48]. Schlumberger focuses on reservoir characterization, drilling, and production [48]. Schlumberger offers 41 products, the product that the VDR website application compares to is ProSource, which offers six different services [48]. Of the six services, the most similar is the ProSource Front Office, especially how they are a website application and visualize the data using charts as illustrated in Figure 3.2 [8]. The charts include pie charts, line charts, scatter plots, and bar charts [8]. They can also change the chart types, zoom in and out, and saves the charts to the user's local machine [8]. Aside from that, in ProSource Front Office provide the ability to [8]:

- map-based spatial filtering,
- export data, and
- detecting and handling data exceptions.

Figure 3.2 Schlumberger ProSource Office Data Visualization [8]

3.2 Contributions

To elaborate more on the solutions proposed in Section 1.2.1, the solution is to develop a VDR website application with a lower cost. To lower the cost of development, OSS will be used. As mentioned in Section 2.4, OSS could greatly help SMEs who could not afford to build an application for themselves. Furthermore, developing a website application means that the requirement for an OS is not liable for consideration as a web browser could exist in most OS, such as, Windows, Linux, MacOS, Android, and iOS. Users would be able to access the application through their mobile devices as well

The VDR website application also allows future customization. Some oil and gas companies request to have their own customized VDR website application, for instance, an integration with a showcase-like application, or handpicked features from the existing application.. Hence, the author's team decided to develop this website application to help the oil and gas industry in Indonesia. There are existing application as mentioned in Section 3.1. However, some features that these applications have are available are not necessary for them. Thus, in the future, this website application will be able to be customized according to their needs.

3.2.1 Difference with Existing Applications

Although the VDR website application shares some similarities and features with existing applications, there are features and concept that makes this VDR website application different. This description of the difference can be detailed in Table 3.1. The pricing provided is the rate charged for one employee, it is costly for oil and gas companies to cover the cost annually. The most common features between every applications is the data visualization, while the prediction model integration with the application is a new concept in a VDR. Most of the application have map integrated, however, not all have a showcase-like application in their map features. The EZLOG Well Log QC and Digitising and INTviewer only have their application available on certain OS, while the VDR website application and ProSource Front Office are available in the form of a website application. Moreover, since the development of the frontend and backend of the VDR website application are separate, any custom modifications for the feature in the backend will not heavily impact the frontend. Thus, future customization will be convenient to be implemented.

Table 3.1 Difference with Existing Application

Features	VDR Website Application	EZLOG Well Log QC and Digitising	INTviewer	ProSource Front Office
Pricing	Rp500,000,000/application	$400/user at the least	$333/user $4000/annual	Not Available for Public
Data Visualization	Available	Available	Available	Available
Prediction Model	Available	Not Available	Not Available	Not Available
Map	Available	Not Available	Available	Available
Showcase	Available	Not Available	Not Available	Not Available
Operating System	Website Application	Windows 10, 8, 7, Vista, XP	Windows, Linux	Website Application
Future Customization	Allowed	Not Allowed	Not Allowed	Not Allowed

3.2.2 Proposed Solution

There are a lot of open-source libraries and frameworks available to build the features that this project focused on. The website application will be made a SPA with VueJs as the framework. VueJs is an open-source JavaScript framework licensed by MIT. Moreover, as the author's scope in Table 1.1 is to visualize the data of oil and gas, the author uses, for example, ChartJs for the visualization. ChartJs is also an open-source JavaScript library with an MIT license. As for the UI libraries for the frontend development, the author decided to use Material Design by Google. Vuetify is a Vue UI library for Material

Design framework released under an MIT license. The latter subsections will further delve into the reason why the author choose the frameworks and libraries for the project.

3.2.2.1 Frontend Framework

Although there are other SPA frameworks such as Angular and React, the author decided to use VueJs instead. Aside from the advantages of Vue mentioned in Section 2.2.1, it is the most user-friendly and easier to understand for beginners who have just started to learn about frontend web development as depicted in Figure 3.3 [49] [50]. There are other additional advantages to using VueJs [49], such as:

- great CLI tool,
- Vuex for state management and Vue Router from the official Vue library,
- lightweight,
- easy to understand documentation.

Meanwhile, React and Angular have their own disadvantages. React needs to import libraries for state and model since it does not implement MVC architecture [17]. Moreover, Angular has various structures compared to Vue, which means it is tougher to learn for beginners [17]. It is also hard to adapt to Angular as there are always significant updates [17]. As not every member in the author's team had learned any frontend framework prior to this project, thus, the author decided to use Vue for being beginner-friendly [50].

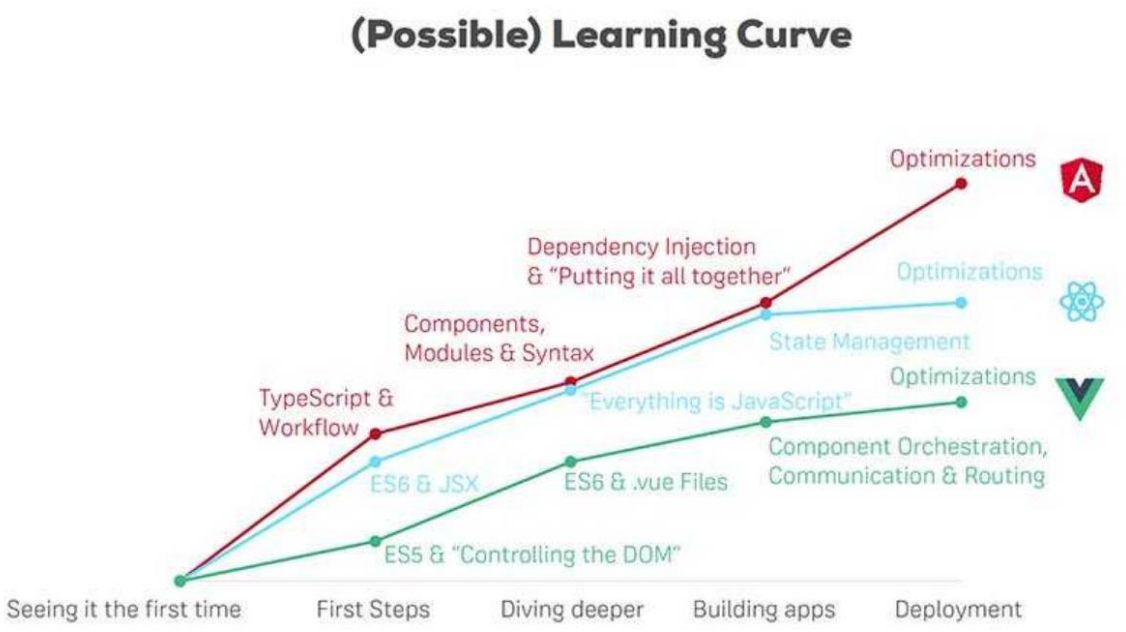

Figure 3.3 Frontend Framework learning Curve [50]

3.2.2.2 User Interface Libraries

Vue has a lot of in-built UI libraries, but the author decided to use Vuetify. As stated in Section 2.2.4, Vuetify uses the concept of Material Design from Google. It has advantages over the other Vue UI libraries, for instance, BootstrapVue, Buefy, Element UI, and Quasar [27]. These advantages can be depicted clearly in Figure 3.4. Compared to other UI libraries, it can be seen that Vuetify supports more features and supports. It is also easier as Vuetify has a large community to support the frontend developers. Moreover, rather than reinventing a component that already exist by using Vuetify, like pre-styled buttons, tables, and treeview, the author could focus more on the functions and features of the website application.

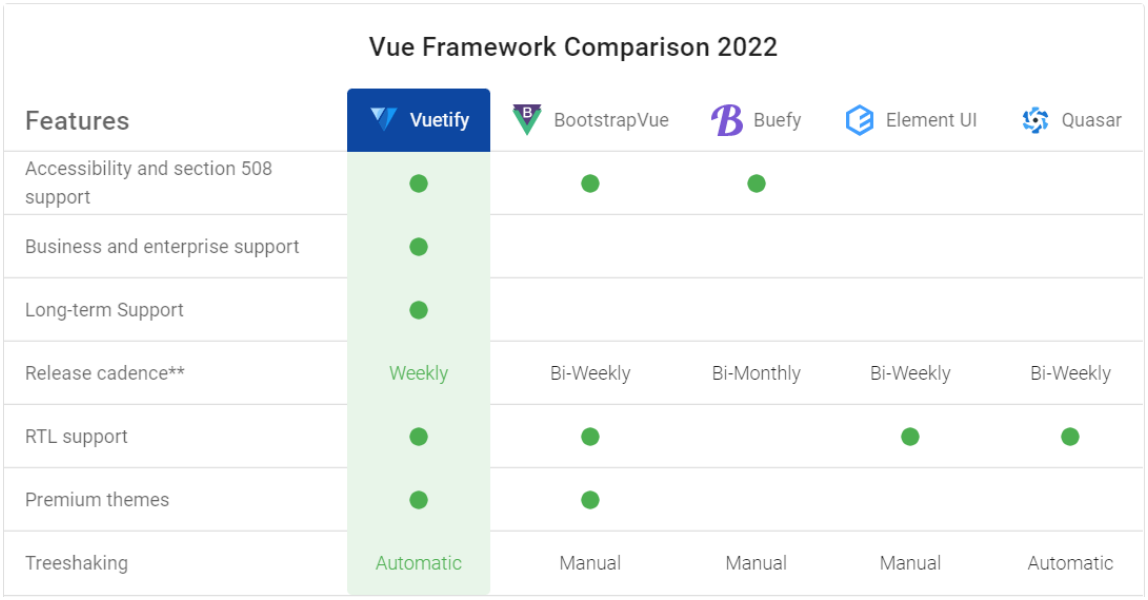

Figure 3.4 Vue User Interface Libraries Comparison [27]

3.2.2.3 Charts

ChartJs supports the various types of charts that the author is going to visualize the data with. For instance, bar charts, line charts, area charts, scatter charts, and bubble charts. There are also advanced tooltips, plugins, and animation options to help aid in the visualization process. There are more complete chart libraries like Google Charts, they support more chart types and supports. However, Google Charts does not support zoom and drag functions [51]. Users should be able to clearly see their data, and the zoom and drag functions are features that should be available for the users. Moreover, when compared to other chart libraries, ChartJs have more features as depicted in Figure 3.5 [52].

Feature	Chart.js	D3	HighCharts	Chartist
Completely Free	✓	✓		✓
Canvas	✓			
SVG		✓	✓	✓
Built-in Charts	✓		✓	✓
8+ Chart Types	✓	✓	✓	
Extendable to Custom Charts	✓	✓		
Supports Modern Browsers	✓	✓	✓	✓
Extensive Documentation	✓	✓	✓	✓
Open Source	✓	✓		✓

Figure 3.5 ChartJs Comparison with Other Charts [52]

CHAPTER 4
SOLUTION DESIGN

This chapter describes and presents the visualization of how the website application works. How it interacts with the user and the server with the use case diagram, the flow of the interaction of the user with the website application with the flowchart, the UI/UX design, as well as how the test phase are going to be conducted in this thesis. It focuses on visualizing the design of the website application at the initial phase of the SDLC, which is the design phase.

4.1 Use Case Diagram

Figure 4.1 represents the UML use case diagram, the main actors in this Figure are the premium client and the regular client, which falls under the same category of a client/user. Meanwhile, the secondary actors are the identity provider and the backend. Everything inside the rectangle container represents the functions or the process that the client/user can perform. In Figure 4.1, there are a total of 6 processes that the client/user can experience in the VDR website application. Every single process will go through the identity provider to confirm that the client/user has access to the process that they are accessing. At the same time, every process related to showing data and modifying it will be sent to the backend for the request.

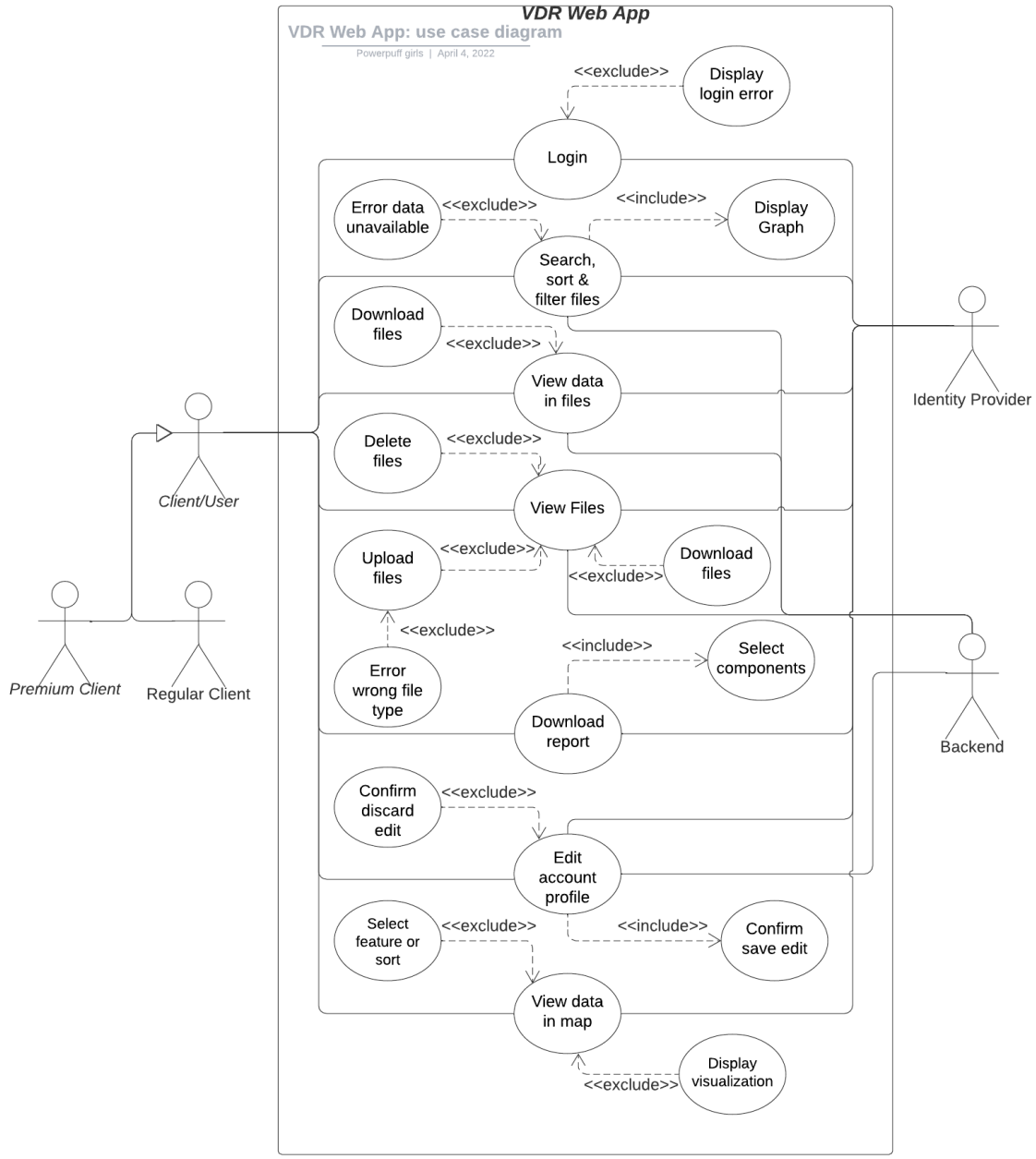

Figure 4.1 Use Case Diagram of a Client/User

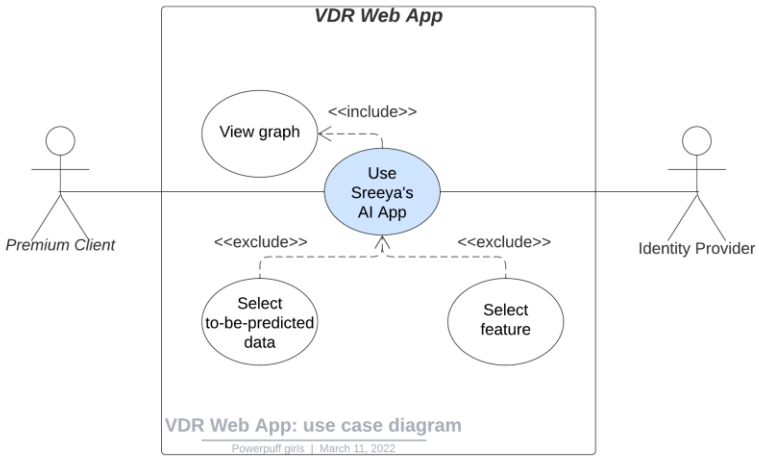

Figure 4.2 Use Case Diagram of a Premium Client

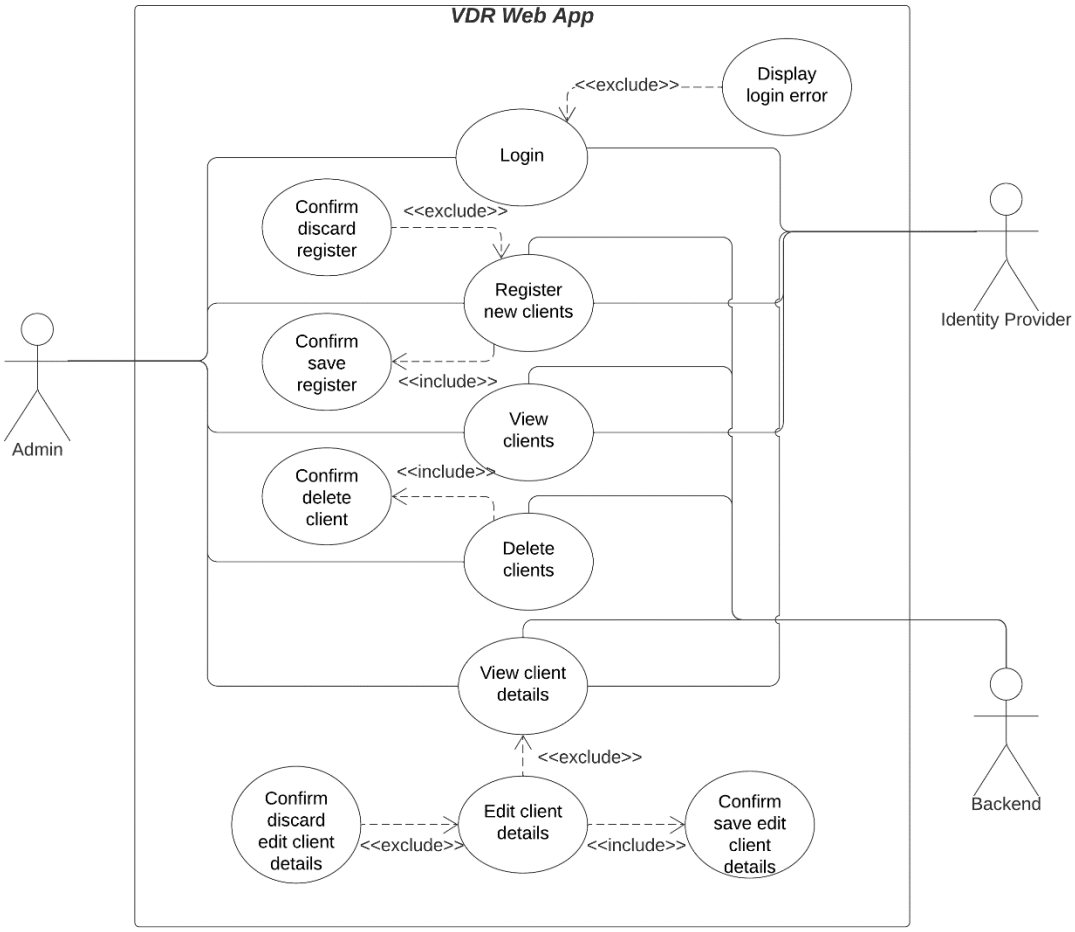

Figure 4.3 Use Case Diagram of an Admin

While Figure 4.1 shows every process that a client could access, Figure 4.2 present the process that is exclusive for a premium client. The process that is available to the premium client is accessing the predictive model that the author's team decided to develop and integrate into the VDR website application. Moreover, the process of the administrator in the website application can be seen in Figure 4.3. As administrators, they could access six processes which are all related to managing and modification of the client/user. As shown in Figure 4.1, there is no option for the user to register, a client/user would need to contact the administrator to get their accounts to be granted access to the website application.

4.2 Flowchart

Figure 4.4 presents the flowchart of a user, the process that the user could explore. Inside Figure 4.4, there are rectangles with a vertical line on the left and right sides of the rectangle. This shape signifies predefined processors which are presented in Figure A.1 and Figure A.2. Meanwhile, Figure 4.5 shows the administrator's flowchart, with their predefined processor defined in Figure A.3.

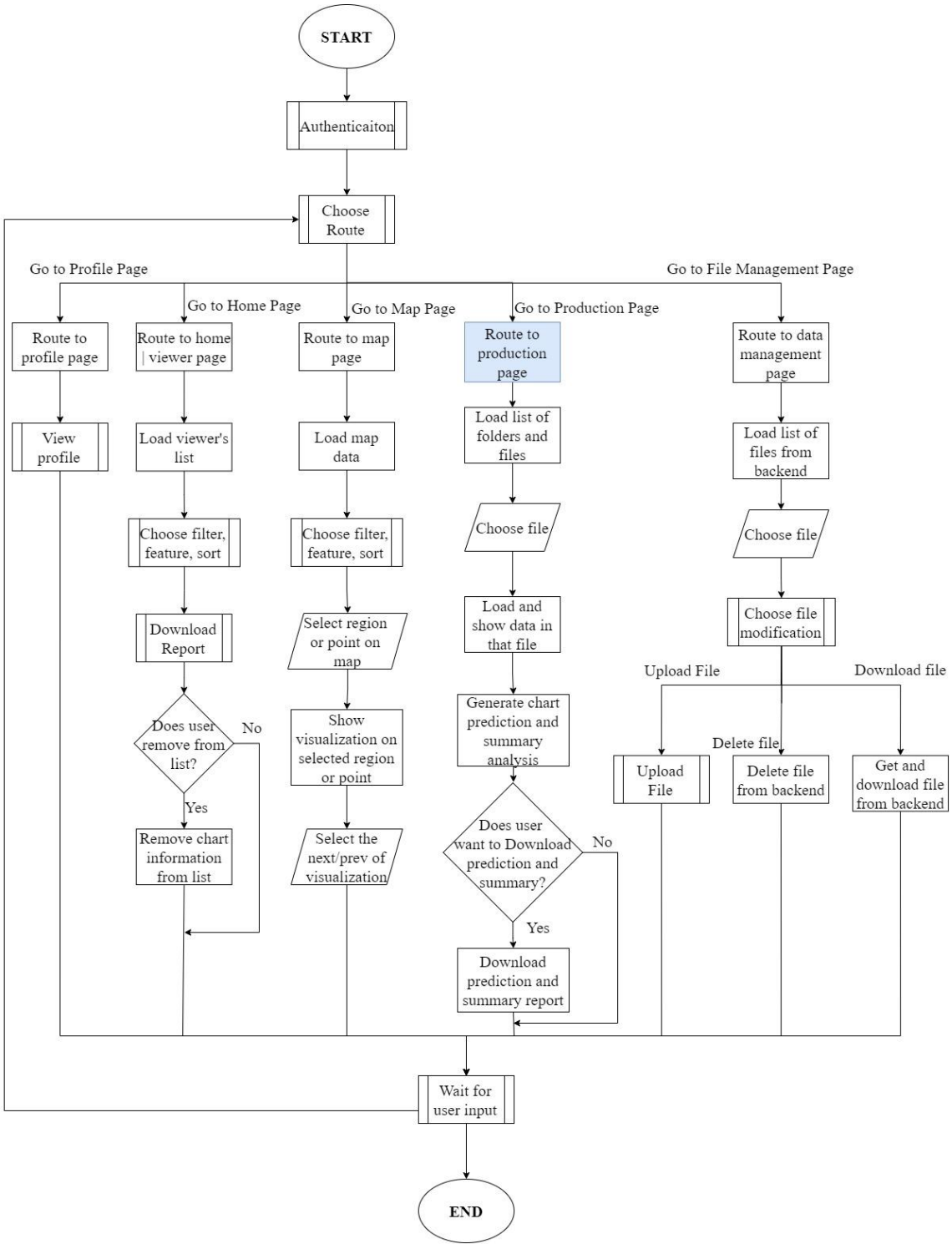

Figure 4.4 User Flowchart

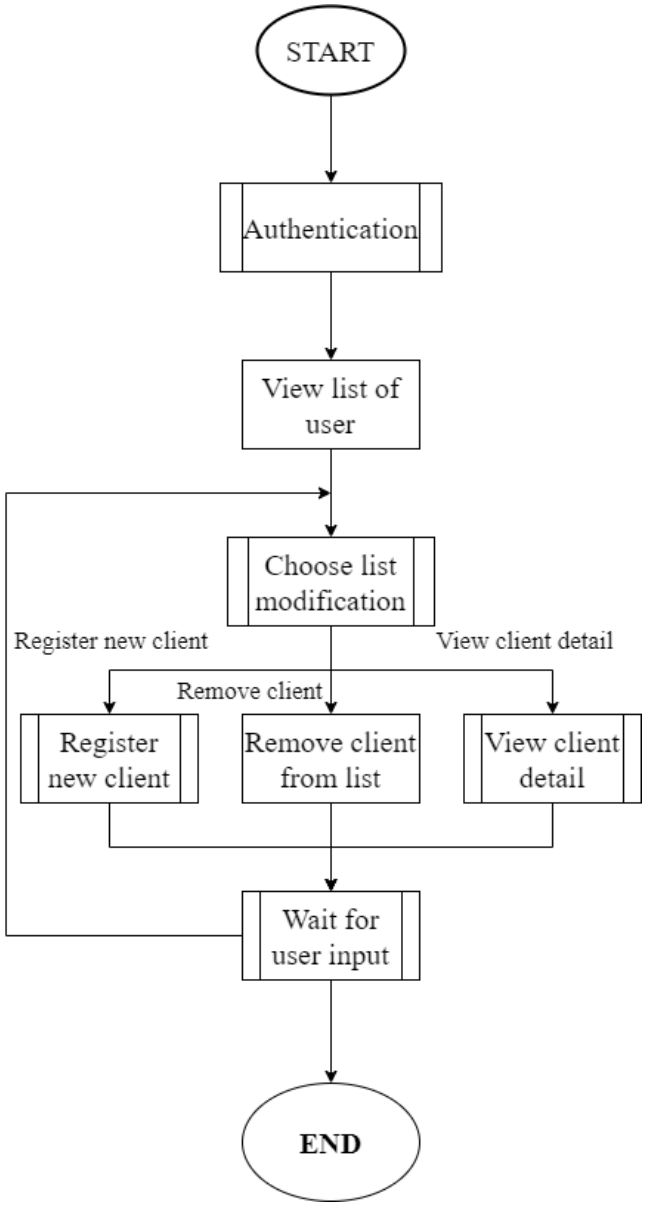

Figure 4.5 Administrator Flowchart

4.3 User Interface / User Experience Design

Using Figma, the home or the landing page as well as the file management page is as presented in Figure 4.6 and Figure 4.10, respectively. The login page and user/client profile of the website application is presented in Figure A.4. According to the design, the

map page as shown in Figure 4.7 is the page that will have the map GIS implemented, the overlay can be seen in Figure A.5. Furthermore, the production page as presented in Figure 4.8 is the page that will contain the result of the prediction model implemented in the website application. The pages for the administrator side are shown in Figure 4.10 and Figure 4.11, presenting how the administrator will view the clients and register new client. This frontend design is designed in accordance with the Material Design using Vuetify.

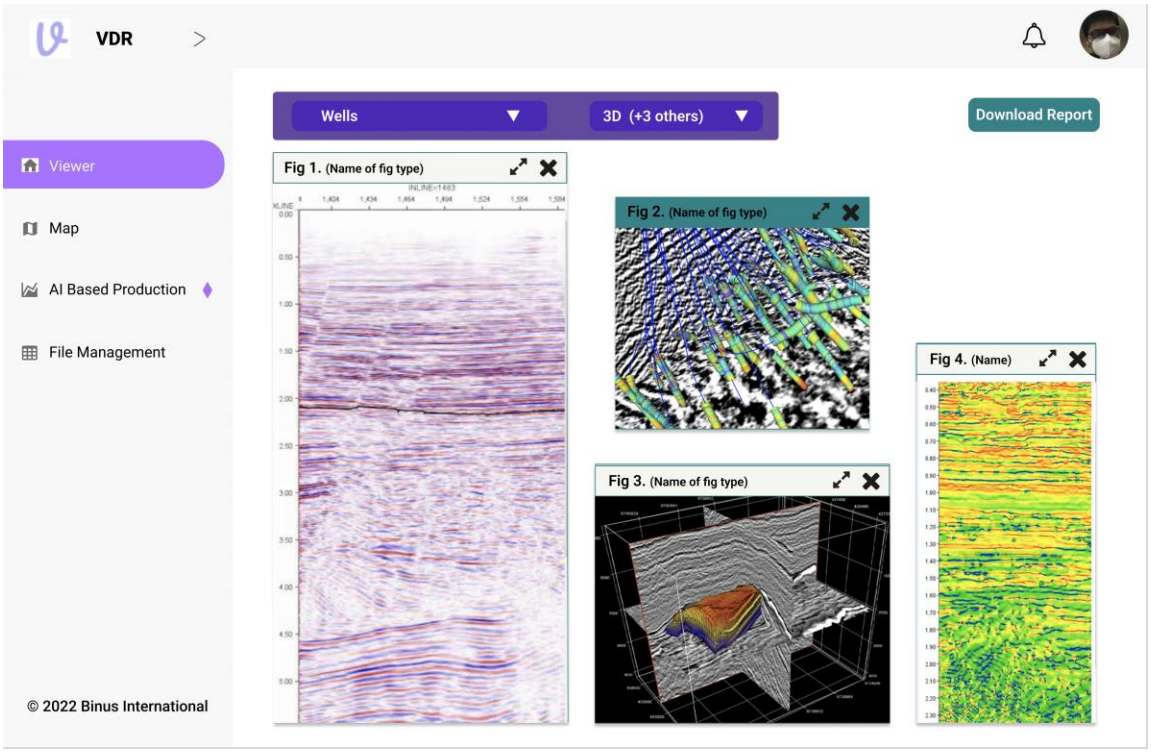

Figure 4.6 Home | Landing Page Design

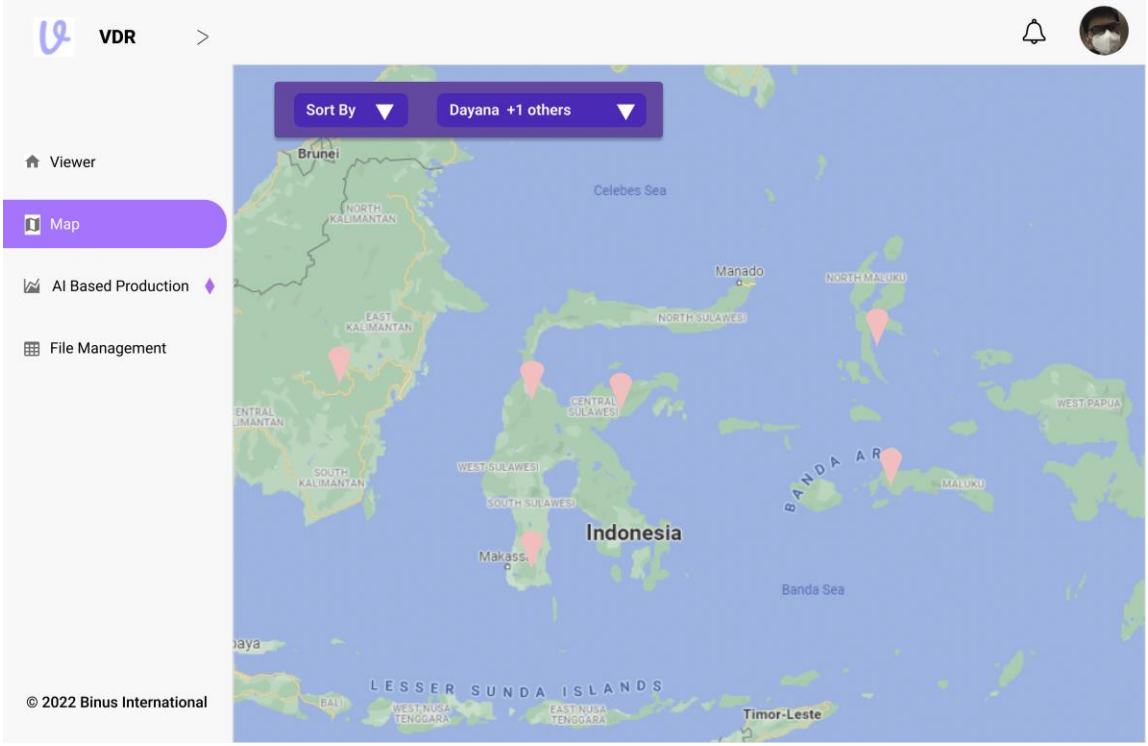

Figure 4.7 Map Page Design

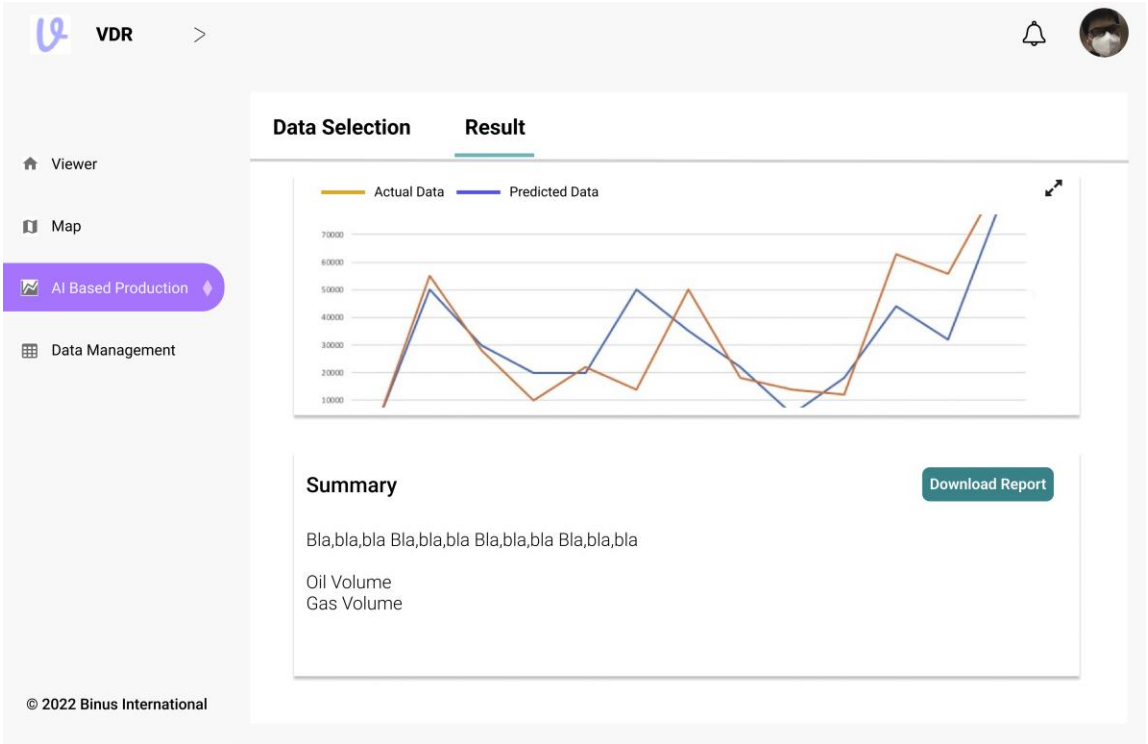

Figure 4.8 Production Page Design

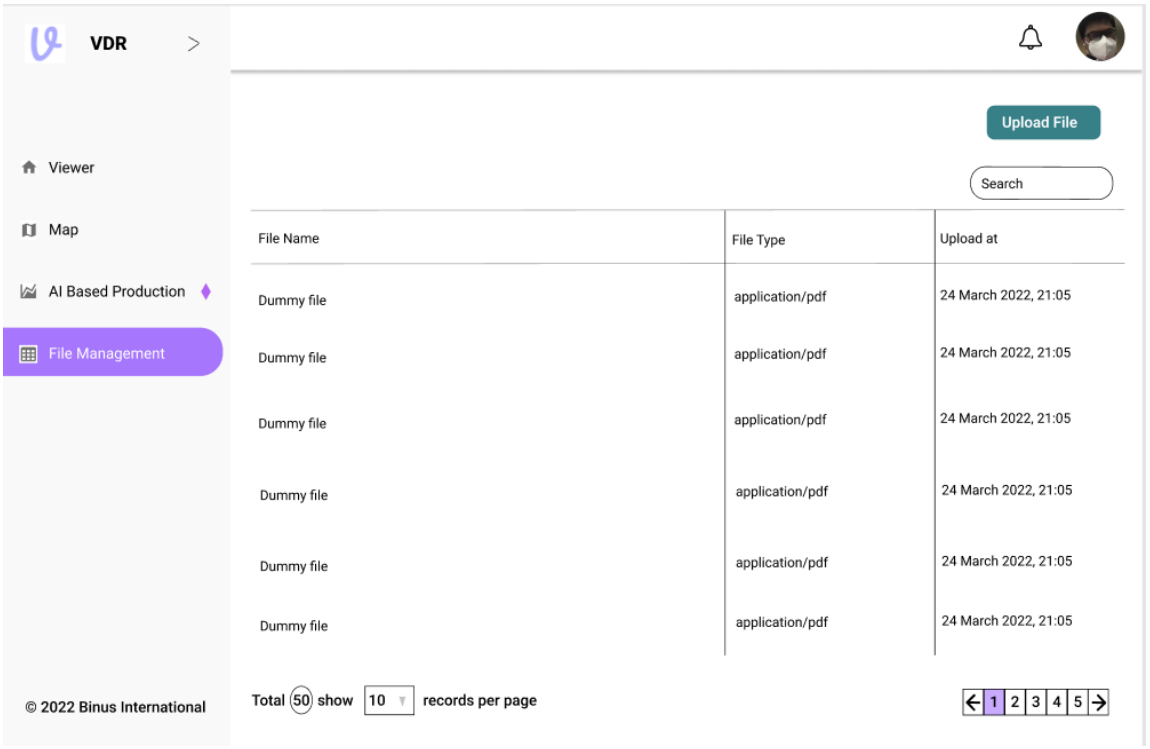

Figure 4.9 File Management Page Design

Figure 4.10 List of Clients Design

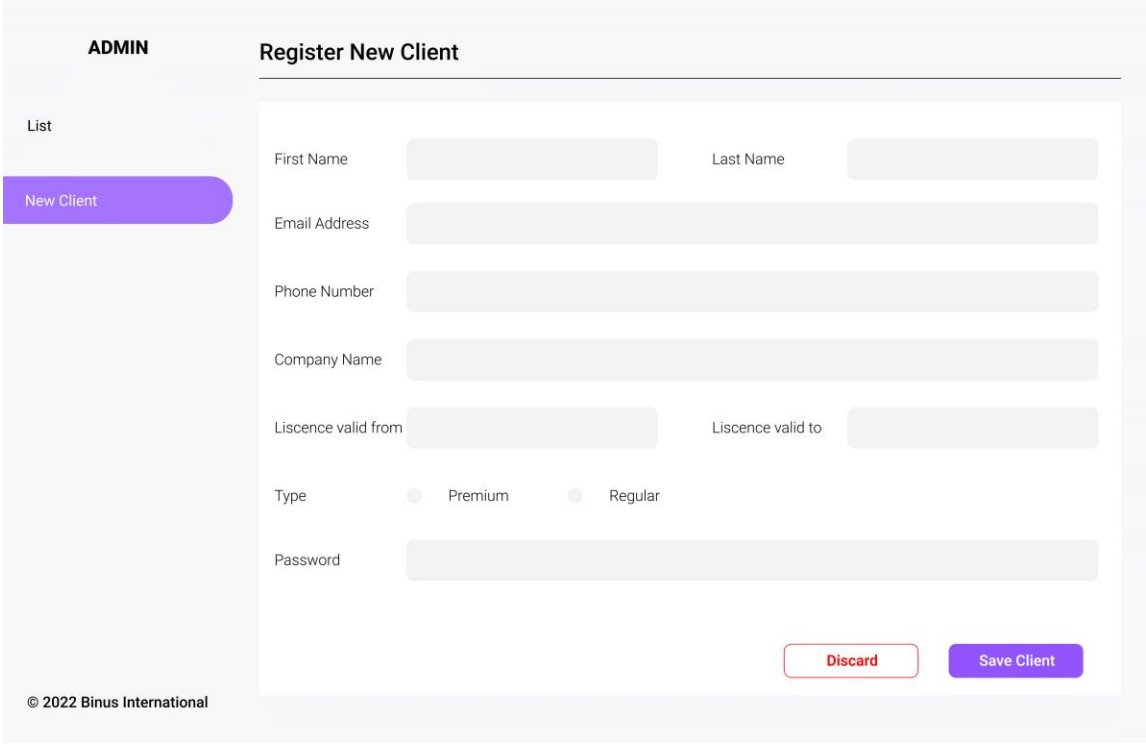

Figure 4.11 Registering New Client Design

4.4 Test Plan

The website frontend tests that the author planned to perform are unit testing and integration testing. As mentioned in Section 2.7, unit testing allows an isolation test for an individual set of codes while integration testing is a test for the assimilation of components [53]. Several unit testing frameworks are available to support developers' unit testing and integration testing, for instance, Jest and Mocha [53]. The author chooses Jest as the testing framework. It is part of the JavaScript test framework, which focuses on simplicity [53].

The author will conduct the test to evaluate the client's main features and function of the website application, which are depicted in Table 4.1.

Table 4.1 Testing Plan

Features	Requirements
Adding new file(s)	Check if file is included in the objects
Remove file(s)	Check if it still exist inside object
Download file(s)	Check if the file(s) is downloaded
Download report	Check if the report is downloaded
Edit profile account	Check if the data is saved and successfully updated
Delete client	Check if the client is still in the list
Edit client details	Check if the client detail is saved and successfully updated
Login	Check if account exist and their authority in the application
Register	Check if account exist and their authority in the application

CHAPTER 5
IMPLEMENTATION

This chapter will specifically elaborate and explain the solution on how the frontend website application would be built and connected. Specifically, on how the web application is operated through route and state, as well as the communication of the website application with the identity provider and the backend.

5.1 Routing

Vue route

5.2 State

vuex

5.3 Communication with Identity Provider

Protocol OpenID

5.4 Communication with Backend

use axios

REFERENCES

[1] M. S. Vassiliou, Historical Dictionary of the Petroleum Industry, Rowman & Littlefield, 2018.

[2] W. Faisol, S. Indriastuti and A. Trihartono, "Indonesia and OPEC: Why does Indonesia maintain its distance?," *IOP Conference Series: Earth and Environmental Science,* vol. 485, p. 012010, 2020.

[3] I. S. Chandranegara. and Z. A. Hoesein, "Policy concept and designs of oil and gas governance in Indonesia's oil companies," *International Journal of Energy Economics and Policy,* vol. 9(3), pp. 121-127, 2019.

[4] B. Harrison, "The Data Room," *Developments in Petroleum Science,* vol. 69, pp. 21-26, 1 October 2020.

[5] "Intviewer - Fast Geoscience Visualization, Analysis & QC," INT, 02 August 2021. [Online]. Available: https://www.int.com/products/intviewer/.

[6] "Lynx Information Systems," Licence Pricing - Lynx Information Systems, [Online]. Available: http://www.lynxinfo.co.uk/download-pricing.html.

[7] Geodwipa Teknika Nusantara, "ptgtn," GTN, [Online]. Available: https://ptgtn.com/.

[8] Schlumberger, ProSource Front Office User Guide, Texas, 2013.

[9] "ProSource front office," ProSource, [Online]. Available: https://www.software.slb.com/products/prosource/prosource-front-office#sectionFullWidthTable.

[10] "Secure, Remote Access to Field Datasets Enables Potential Investors to Complete Asset Evaluations," Schlumberger, [Online]. Available: https://www.slb.com/resource-library/case-study/dss/delfi-virtual-data-room-generic-asia-pacific-cs.

[11] M. Lewis, "The Importance of a Virtual Data Room Index and Folder Structure," Deal Room, [Online]. Available: https://dealroom.net/blog/the-importance-of-a-virtual-data-room-index-and-folder-structure.

[12] M. Jazayeri, "Some Trends in Web Application Development," *Future of Software Engineering (FOSE '07)*, pp. 199-213, 2007.

[13] J. Tie, J. Jin and X. Wang, "Study on application model of three-tiered architecture," in *Second International Conference on Mechanic Automation and Control Engineering*, Inner Mongolia, 2011.

[14] Y. Xing, J. Huang and Y. Lai, "Research and analysis of the front-end frameworks and libraries in E-Business Development," *Proceedings of the 2019 11th International Conference on Computer and Automation Engineering - ICCAE 2019*, 2019.

[15] N. Li and B. Zhang, "The research on Single Page Application Front-end development based on Vue," *Journal of Physics: Conference Series*, vol. 1883, 2021.

[16] N.-A. Sireteanu and D. Homocianu, "Front-end frameworks for development of spa and MPA web application," *SSRN Electronic Journal*, December 2021.

[17] V. Hutagikar and V. Hegde, "Analysis of Front-end Frameworks for Web Applications," *International Research Journal of Engineering and Technology (IRJET)*, vol. 07, no. 04, April 2020.

[18] "Getting started - vue.js," Vue.js, [Online]. Available: https://012.vuejs.org/guide/.

[19] R. Hartson and P. S. Pyala, "Introduction," in *The UX Book*, 2012, pp. 1-46.

[20] J. M. Caroll, "CHAPTER 1 - Introduction: Toward a Multidisciplinary Science of Human-Computer Interaction," in *HCI Models, Theories, and Frameworks*, 2003, pp. 1-9.

[21] MacKenzie and I. Scott, "Chapter 1 - Historical Context," in *Human-computer Interaction*, 2013, pp. 1-26.

[22] S. R. Venna, R. N. Gottumukkala and V. V. Raghavan, "Chapter 3 - Visual Analytic Decision-Making Environments for Large-Scale Time-Evolving Graphs," in *Handbook of Statistics*, vol. 35, 2016, pp. 81-115.

[23] J. Wang, "From self-efficacy to human-computer interaction design," *Journal of Physics: Conference Series*, vol. 1168, no. 3, February 2019.

[24] B. Shneiderman, "The Eight Golden Rules of Interface Design," University of Maryland, [Online]. Available: https://www.cs.umd.edu/users/ben/goldenrules.html.

[25] Google, "Material Design," [Online]. Available: https://material.io/.

[26] "Material Components," [Online]. Available: https://github.com/material-components/material-components.

[27] Vuetify, "Why Vuetify," [Online]. Available: https://vuetifyjs.com/en/introduction/why-vuetify/#why-vuetify3f.

[28] Z. Wei, "Research on the application of Open source software in Digital Library," *Procedia Engineering*, vol. 15, pp. 1662-1667, 2011.

[29] Q. Jiang., J. Qin. and L. Kang, "A literature review for Open Source Software Studies," *Lecture Notes in Computer Science*, pp. 699-707, 2015.

[30] M. Nelson, R. Sen and C. Subramaniam, "Understanding open source software: A research classification framework," *Communications of the Association for Information Systems*, vol. 17, pp. 266-287, February 2006.

[31] E. N. Hahn, "An Overview of Open-Source Software Licenses and the Value of Open-Source Software to Public Health Initiatives," *JOHNS HOPKINS APL TECHNICAL DIGEST*, vol. 32(4), 2014.

[32] J. Bitzer, W. Schrettl and P. J. H. Schröder, "Intrinsic Motivation in Open Source Software Development," *Journal of Comparative Economics*, vol. 35, no. 1, pp. 160-169, October 2006.

[33] VueJs, "Vue.js - The Progressive JavaScript Framework," [Online]. Available: https://vuejs.org/.

[34] G. Gurung, R. Shah and D. P. Jaiswal, "Software development life cycle models-A comparative study," *International Journal of Scientific Research in Computer Science, Engineering and Information*, pp. 30-37, 2020.

[35] N. S. Yadav, V. Goar and M. Kuri, "AGILE METHODOLOGY -A PERFECT SDLC MODEL WITH SOME IMPROVEMENTS," *Journal of Critical Reviews*, vol. 7, no. 19, pp. 2511-2514, August 2020 .

[36] R. Sherman, "Project Management," *Business Intelligence Guidebook*, pp. 449-492, 2015.

[37] "Manifesto for Agile Software Development," Agile Manifesto, 2001. [Online]. Available: https://agilemanifesto.org/.

[38] P. Adi, "Scrum Method Implementation in a Software Development Project Management," *International Journal of Advanced Computer Science and Applications,* vol. 6, no. 9, September 2015.

[39] S. Sylesh, "A Study of Software Development Life Cycle Process Models," *SSRN Electric Journal,* 10 June 2017.

[40] Lucid Chart, "How to Design Software Architecture," Lucid Chart, [Online]. Available: https://www.lucidchart.com/blog/how-to-design-software-architecture.

[41] Y. Waykar, "A Study of Importance of UML diagrams: With Special Reference to Very Large-sized Projects," in *International Conference on Reinventing Thinking beyond boundaries to Excel*, Faridabad, 2013.

[42] H. Koc, A. M. Erdoğan, Y. Barjakly and S. Peker, "UML Diagrams in Software Engineering Research: A Systematic Literature Review," *Proceedings,* vol. 74, no. 13, March 2021.

[43] M. A. M. Daril, R. Kassim, N. Afiqah S and M. I. A. , "A Development of Logical Design Flowchart for Computerized System of Problem Solving and Improvement Procedure," *International Journal of Advanced Trends in Computer Science and Engineering,* vol. 9, no. 1.1, pp. 400-407, March 2020.

[44] "What is a Flowchart," Lucid, [Online]. Available: https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial.

[45] S. Herbold, U. Bünting, J. Grabowski and S. Waack, "Deployable Capture/Replay Supported by Internal Messages," *Advances in Computers,* vol. 85, pp. 327-367, 2012.

[46] F. Lonetti and E. Marchetti, "Emerging Software Testing," *Advances in Computer,* vol. 108, pp. 91-143, 2018.

[47] C. Eaton, "Advances in Web Testing," *Advances in Computer*, vol. 75, pp. 281-306.

[48] Schlumberger, "Schlumberger," Schlumberger Limited, [Online]. Available: https://www.slb.com/.

[49] P. Pšenák and M. Tibensky, "The usage of Vue JS framework for web application creation," pp. 61-72, January 2020.

[50] B. Behera, "Comparison Between React, Angular and Vue," Medium, [Online]. Available: https://medium.com/pixance-studios/comparison-between-react-angular-and-vue-be488478cbfd.

[51] Fusion Charts, "Google Charts vs Chart.js," Fusion Charts, [Online]. Available: https://www.fusioncharts.com/javascript-charting-comparison/google-charts-vs-chartjs.

[52] ChartJs, "Comparison with Other Charting Libraries," ChartJs, [Online]. Available: https://www.chartjs.org/docs/2.9.4/notes/comparison.html.

[53] Vue.js, "Vue.js," Vue.js, [Online]. Available: https://v2.vuejs.org/v2/guide/testing.html.

[54] J. Manhas, "Comparative Study of Cross Browser Compatibility as Design Issue in Various Websites," *BVICAM's International Journal of Information Technology (BIJIT),* vol. 07, no. 1, pp. 815-820, January 2015.

[55] L. N. Sabaren, M. N. Mascheroni, C. L. Greiner and E. Irrazábal, "A Systematic Literature Review in Cross-browser Testing," *Journal of Computer Science and Technology*, vol. 18, no. 1, 2018.

[56] "Model–view–controller," wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller.

[57] B. Shneiderman, Designing the User Interface Strategies for Effective Human-Computer Interaction, 2016.

[58] E. Wong, "User Interface Design Guidelines: 10 Rules of Thumb," Interaction Design Foundation, [Online]. Available: https://www.interaction-design.org/literature/article/user-interface-design-guidelines-10-rules-of-thumb.

APPENDICES

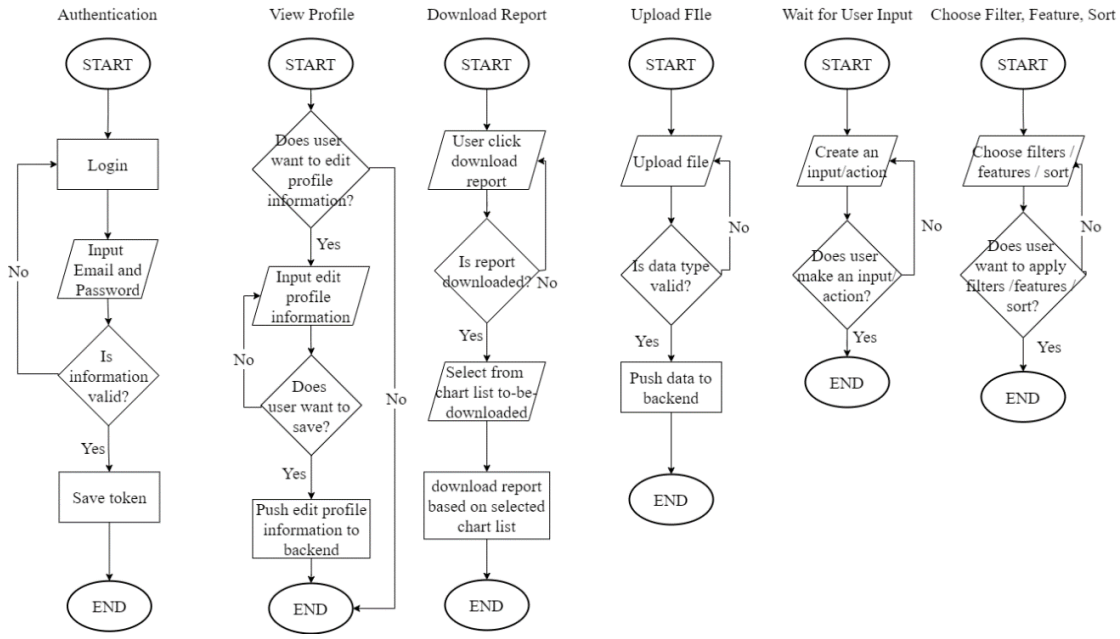

Figure A. 1 Predefined Processors part 1

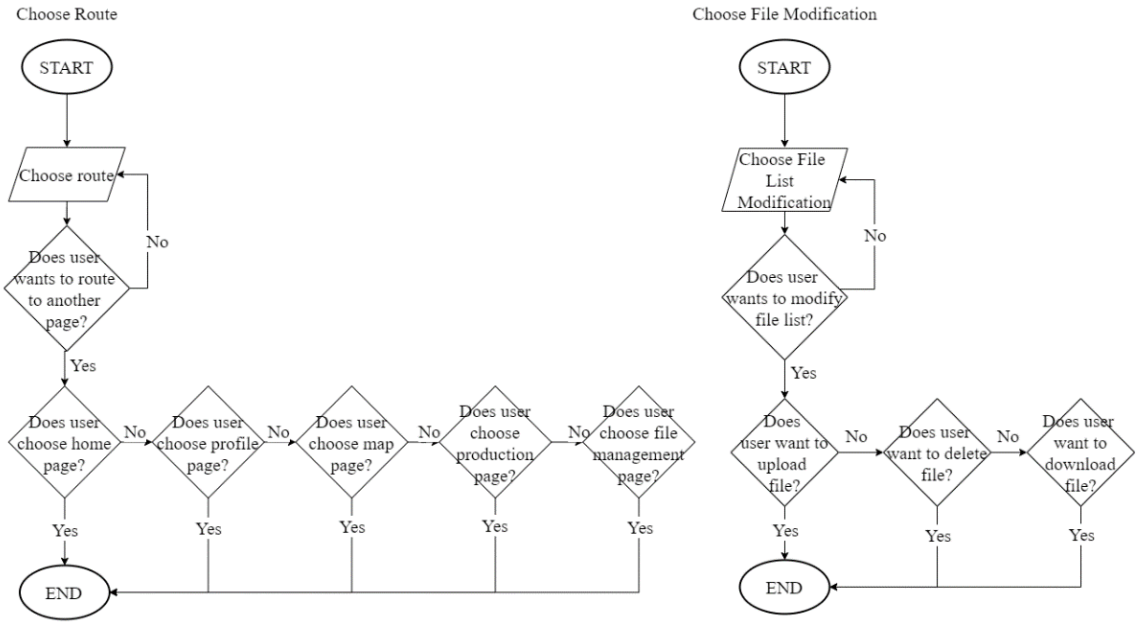

Figure A. 2 Predefined Processors part 2

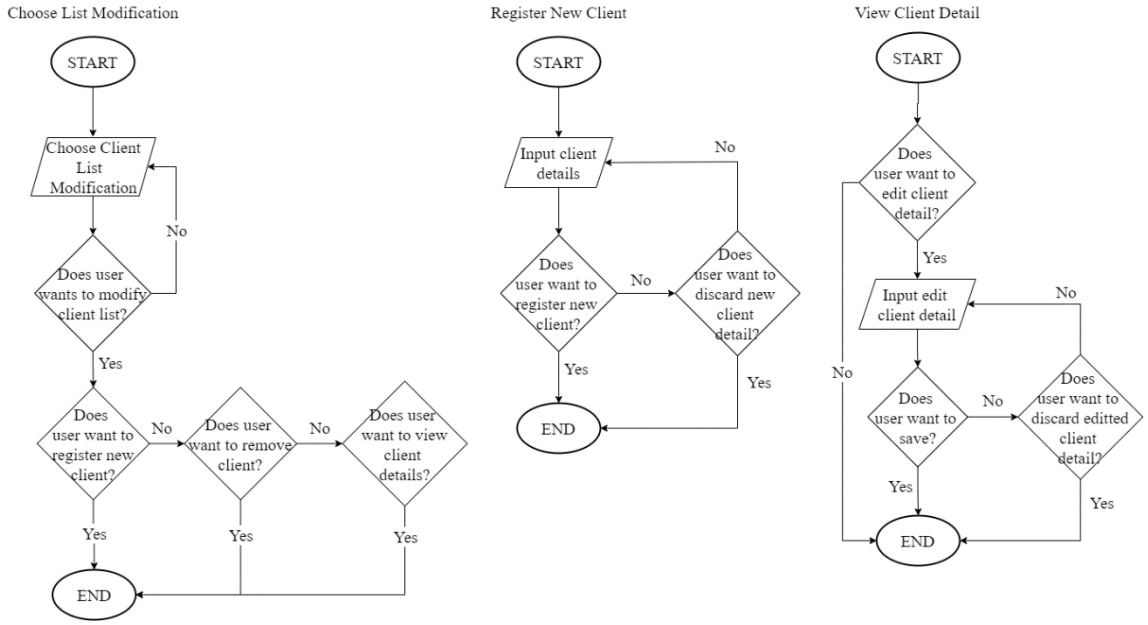

Figure A. 3 Predefined Processors part 3

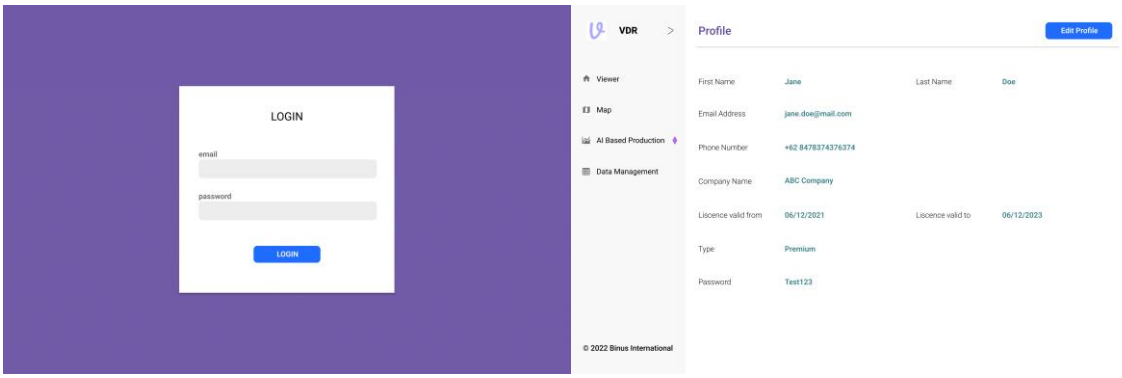

Figure A. 4 Design (a) Login Page (b) View Client Profile Detail

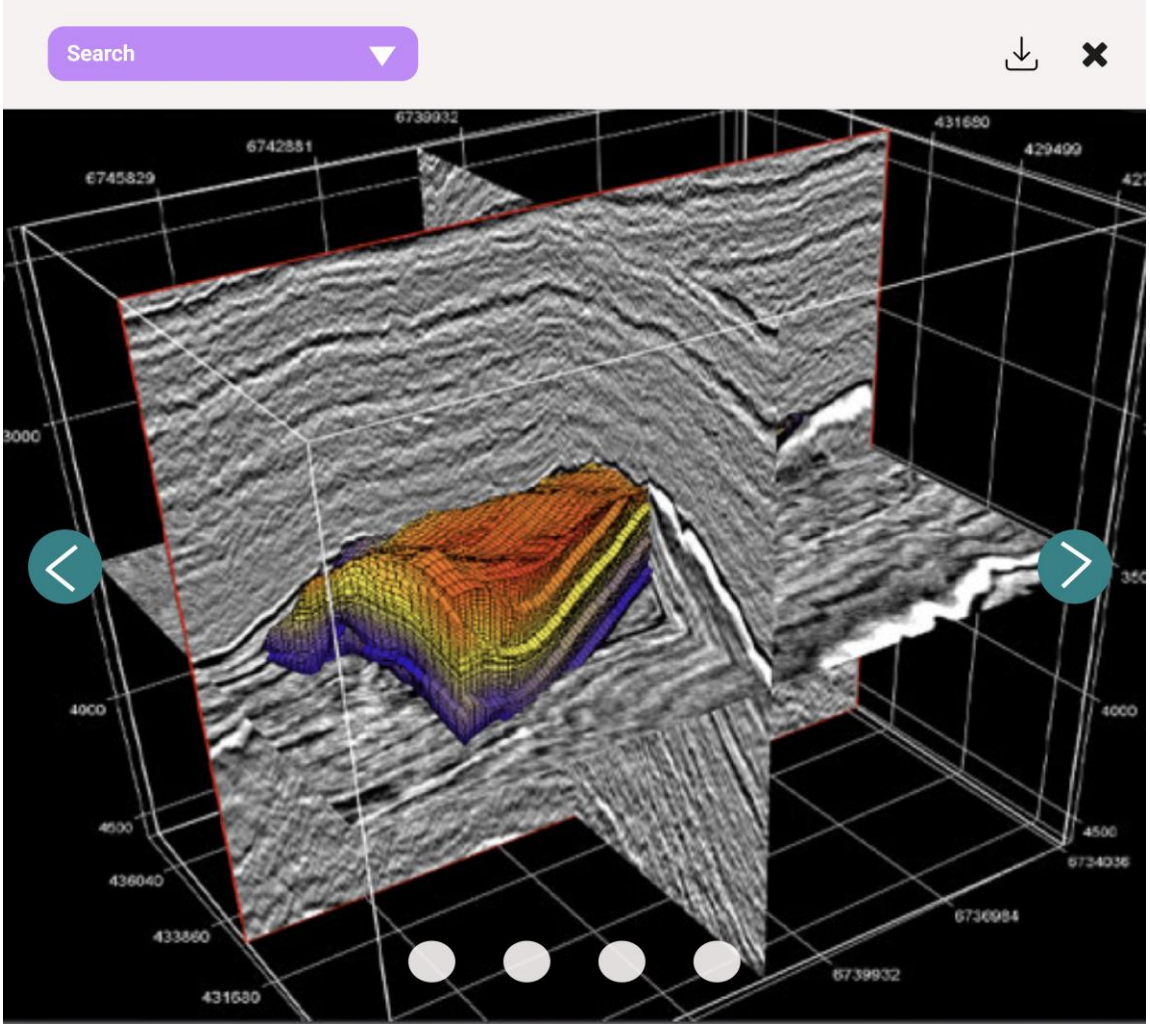

Figure A. 5 Final Design Map Overlay Detail

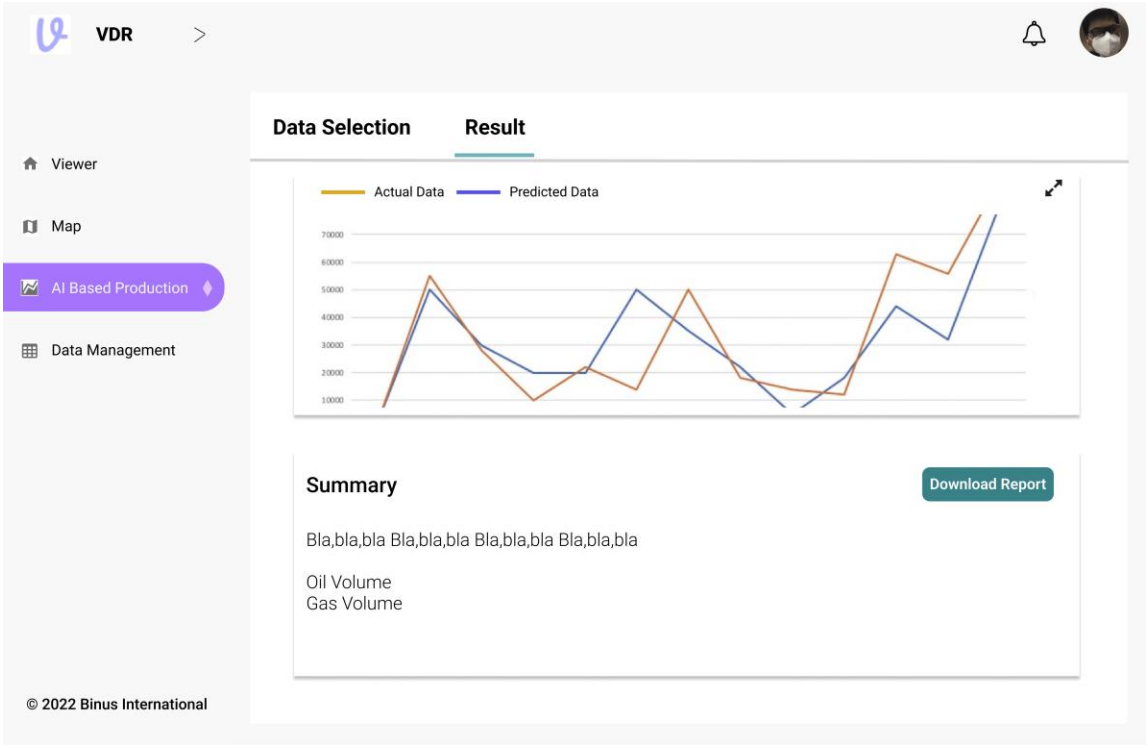

Figure A. 6 Final Design Production Page Result Page