Table of Contents

List of Figures

List of Tables

# CHAPTER 1
# INTRODUCTION

In this chapter, the background of this thesis is described, and the author & team's scope of doing the thesis is described. The objective, such as the aim and benefits, are also included in the chapter.

## 1.1 Background

Oil and gas were used for lamps and lubricants starting back in 346 AD [1]. The oil well drilled was done by China [1]. Then moving forward to the late 18$^{th}$ century and early 19th century, it was an era in which oil and gas industries that still dominate the world until today were established [1]. Since the mid-1950s, the oil demand has become the world's most important source of energy. Refined oil products are used for many things, such as supplying energy to the power industry, heating homes, providing fuel for vehicles and airplanes, and coating pills [2]. Aside from that, oil and gas are also economically essential as the oil and gas industries reduce unemployment as the industries provide numerous jobs. Oil and gas industries in America also support the country economically as the industries are projected to provide $1.6 trillion in federal and state tax revenue between 2012 and 2025 [3]. The tax revenue is then intended to support schools, hospitals, and public infrastructure across the country [3].

As for Indonesia, the well-known benefits of oil and gas are for vehicle fuels, LPG, and petrochemical products such as synthetic rubber and polyurethane materials [4]. These benefits increased oil and gas consumption from around 1180 thousand barrels a day in 2000 to 1750 thousand barrels a day in 2018 [5]. Unfortunately, these also lead to a decrease in oil and gas production that causes lesser income for the country [5]. However, the oil and gas industries had provided an enormous contribution to state revenue of around 216.9 trillion rupiahs in 2014, which

boosted the country economically [5]. According to BPS Statistics Indonesia, there were 20326 Indonesians that worked in the oil and gas industries in 2019 [6]. The unemployment number is lesser with these numbers, representing more economic growth.

With all the importance and benefits of oil and gas and the decrease in oil and gas production mentioned earlier, many software applications such as Virtual Data Room (VDR) now help the oil and gas industries. The VDR is an online data room in the form of a website that requires authorization to access it. VDR helps the industries to store and analyze data. It also attracts potential clients to invest in the industries by giving clients the platform to visualize the data. Other software applications may also help the industries to analyze which locations can produce more oil and gas. Software with a visualization tool can also help oil recovery [7]. For data visualizations, several well-known software services are Schlumberger's ProSource Front Office, Lynx, and INTViewer. However, both software is relatively pricey. On one side, Lynx provides visualization through seismic viewers, costing at least $400 per license per year [8]. At the same time, INTViewer, which provides visualization of seismic and geospatial data, can cost up to $60,000 per year [9]. Although the software is pricey, Pertamina still utilizes INT's web-based HTML5Viewer with their platform to visualize their data quickly and efficiently [10]. Some other oil and gas companies in Indonesia, according to the product owner, also frequently look for similar softwares with a cheaper price and customized features. Therefore, this inspired the author and team to develop a Virtual Data Room (VDR) application with custom features to the software mentioned before at a lower cost which became the primary goal of this thesis. In addition, the proposed VDR application will be developed as a web application as part of requirements from our customer, i.e., PT Geodwipa Teknika Nusantara [11].

The features include implementing a Geographic Information System (GIS) in the VDR. A Geographic Information System (GIS) is a system of software, hardware, and data that enables the exploitation, analysis, and display of data and information based on the location on the surface of the Earth that is connected to the data [11]. It helps to enhance the VDR application by providing a better understanding of mining locations through visualizations, determining which locations are better in terms of use, and better geographic data documentation for the industries and clients [11]. The knowledge of the data is richer after being analyzed using GIS [11]. Data is also easier to search, explore and display with a GIS map [11]. Therefore, the author will focus on implementing this matter.

## 1.2 Scope

The scope of the author's problem, the solution that the author thought of to solve the problem, and the scope of work that the author and the author team are responsible for are explained below.

### 1.2.1 Scope of the Problem and Solution

The main problem that leads to the creation of this thesis is the high cost of the existing software. The solution that the author and team thought of is to develop a modernized design website application using open-source frameworks and libraries to solve this problem. The website's features are chosen based on the author and team's decision on the vital features of previous software as well as developing custom features requested by our customer.

The software that the author and team referenced is Schlumberger's ProSource Front Office [12]. Therefore, the website application contains data visualization in pie charts, line charts, scatter plots, stackable charts of bar and area, and bubble charts similar to Schlumberger's ProSource Front

Office's features [12]. The author and team also attempt to implement other GIS integration features to conduct geospatial analysis.

### 1.2.2 Scope of Work

In this thesis, the author is responsible for the front end of one of the website application's features, including designing user interfaces (UI) / user experience (UX), developing the feature based on the design, and implementing user interaction. The feature that the author is responsible for is a visualization of production data utilized with a geographic information system (GIS). Inside the GIS of the VDR, the author is also responsible to develop a custom showcase. The showcase will display information regarding oil and gas based on the chosen locations for analysis that can be easily understand by non-technical users. By being responsible, the author is determined to use the frameworks that can provide new experiences of GIS that provide efficiency for the VDR users. The author also makes sure all the frameworks used have open-source licenses to fulfill the goal of developing cheaper software than existing. Once the author finishes developing this feature, the author is required to test the feature to ensure that all requirements are met.

Simultaneously, the author and team are responsible for assembling different kinds of visualizations, and their diverse functionalities merged in the end into a website application. Each team member had separate tasks that required them to successfully finish their tasks one by one to create a complete website. To complete the tasks, the author's team decided to implement an agile methodology as the Software Development Life Cycle (SDLC) in developing the tasks. The agile methodology is chosen due to certain benefits discussed in section 2.5. The tasks and roles in the SDLC for each member are:

*Table 1.1 Scope of Tasks*

| Member names | Tasks |
|---|---|
| Chan Elizabeth W | - Design and develop the front-end of the proposed VDR application that uses GIS<br>- Design and develop a custom showcase of data in GIS page that can easily understand by non-technical user<br>- Testing<br>- In the agile methodology, her role is as a scrum member. |
| Kotrakona Harinatha Sreeya Reddy | - Gathering and Processing Data<br>- Develop predictive models based on the data collected<br>- Visualizing data through diagrams, such as charts, as well as performing data analytics<br>- In the agile methodology, her role is as a scrum member. |
| Vicky Vanessa | - Design UI/UX of the front-end of the website application<br>- Visualizing the production data of oil and gas<br>- Testing<br>- In the agile methodology, her role is as a scrum master. |

## 1.3 Objective

The aim and benefits of this thesis are explained below.

### 1.3.1 Aim and Benefits

The major aim of developing this VDR website application is to help the oil and gas industries locate more profitable resources sites by visualizing the oil and gas volume data gathered throughout the time and visualizing the reserve resources. The visualization includes the GIS explained earlier. Additionally, the VDR website application aims to acquire explicit information from oil and gas production data which will be used to build a predictive model. This predictive model will help the oil and gas industries predict and reveal potential sites containing more oil and gas.

The VDR website application benefits oil and gas industries to personally store and visualize their data. The VDR also benefits the oil and gas industries to attract many domestic and international clients. Aside from providing benefits for the oil and gas industries, the application is also useful to aid engineers in comprehending complex data and understanding how the data can be used. With the application, the author and team also hope to increase the use of local services in the oil and gas industries that help the industries to gain more profits and indirectly increase the national income.

## 1.4 Structure

This thesis consists of five chapters that are shortly explained below.

### 1.4.1 Chapter 1: Introduction

Chapter 1 introduces the background, the scope, and the objective, including the aims and benefits of this thesis.

### 1.4.2 Chapter 2: Theoretical Foundations

Chapter 2 describes and explains the theoretical concept of the approaches in which the thesis is developing and the tools used by the author to develop the thesis.

### 1.4.3 Chapter 3: Problem Analysis

Chapter 3 discusses existing application and proposed solution along with the comparison.

### 1.4.4 Chapter 4: Solution Design

Chapter 4 further explains the user interaction with the system, the flow of the system, and interaction of the thesis features along with the design.

### 1.4.5 Chapter 5: Conclusion

Chapter 5 describes the final product and conclusion of the thesis

# CHAPTER 2
# THEORETICAL FOUNDATION

This chapter explains the theories behind the proposed application. The theories include the things that the author wants to develop, such as Virtual Data Room, Geographic Information System, the web application, and frontend development—then followed with the theories on open-source software as the author used this kind of software in developing the application. After that, theories on the SDLC of developing the application and testing the application are described.

## 2.1 Data Room (Virtual data room)

As one of the functionalities of this thesis is to store data and visualize it through a website application, the thesis implements the concept of a Virtual Data Room (VDR). The VDR is a kind of data room. By definition, a data room is a precious instrument when oil and gas companies want to sell their assets [13]. With data room, companies can attract more buyers by providing information for buyers to consider their product [13]. Aside from VDR, the other kinds of data rooms are Physical data rooms (PDR) and a combination of PDR and VDR [13].

The PDR is a room with high surveillance [13]. The room is filled with confidential documents that the team of potential buyers will examine thoroughly to find the financial secrets of potential investments [13]. As there are many documents, it is a time-consuming, expensive, and complex activity [13]. Time-consuming due to analyzing and sorting data that are useful [51]. Expensive due to the need to spend on high surveillance rooms and traveling if buyers' and sellers' origins are different [13]. It is complex due to the need to copy extensive important data and ship it back

to its origin [13]. It is also complex because the sellers need to keep the data up-to-date until the buyers have time to visit the sellers' PDR [13].

Due to the complexity and significant expense of PDR, it is largely replaced by VDR [13]. The VDR, also known as an online data room, is a website that requires authorization to access it [13]. With authorization such as unique passwords for buyers to access the VDR, it is as secure as PDR [13]. Aside from security, VDR also advances accessibility and saves cost [13]. For Sellers, it is easily accessible for quick data updates [13]. At the same time, potential buyers can easily access the website from any location and at any time [13]. Buyers also gain the advantage of saving money by not needing to travel [13]. Therefore, it can be concluded that VDR is better than PDR. Some of the examples of VDR are FirmRoom and Lynx's GeoViewer. FirmRoom provides a files and documents management page on their VDR website where users can upload, store, organize, and view the files and documents, as shown in Figure 2.1 [14]. While Lynx's GeoViewer provides visualization of the data stored on the website, which is chosen based on the data locations on the map, as shown in Figure 2.2 [15].

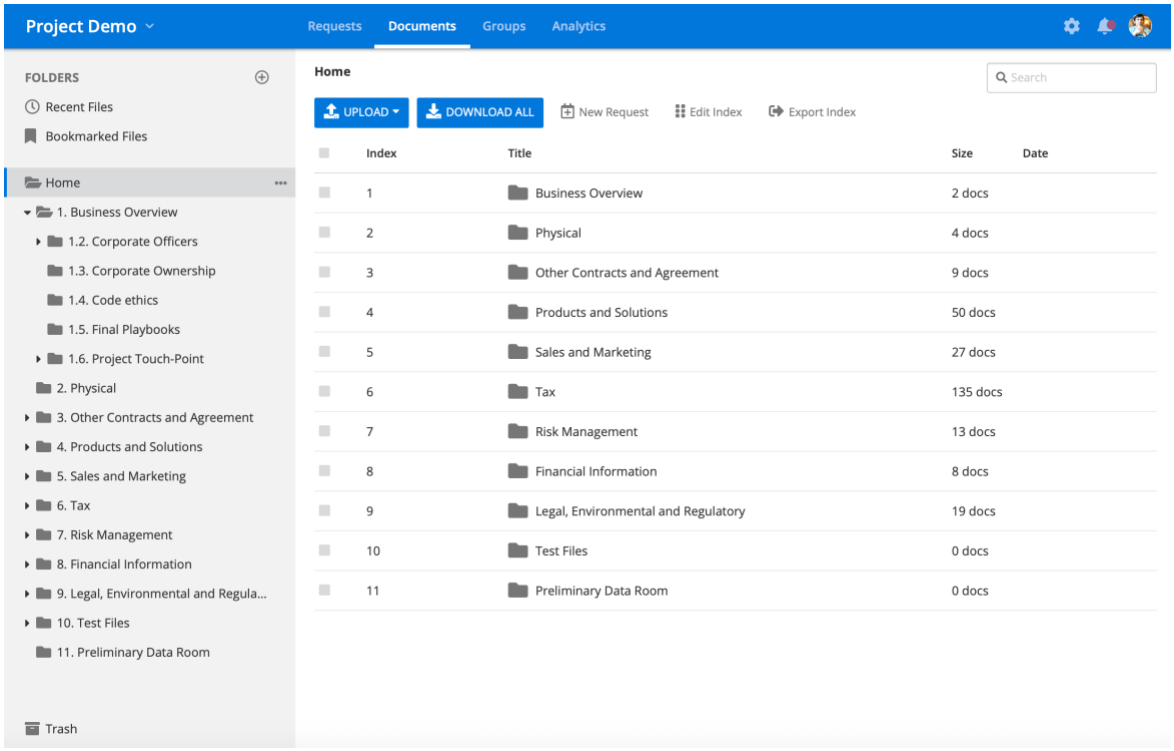


*Figure 2.1 FirmRoom's Data Room Index Demo [14]*

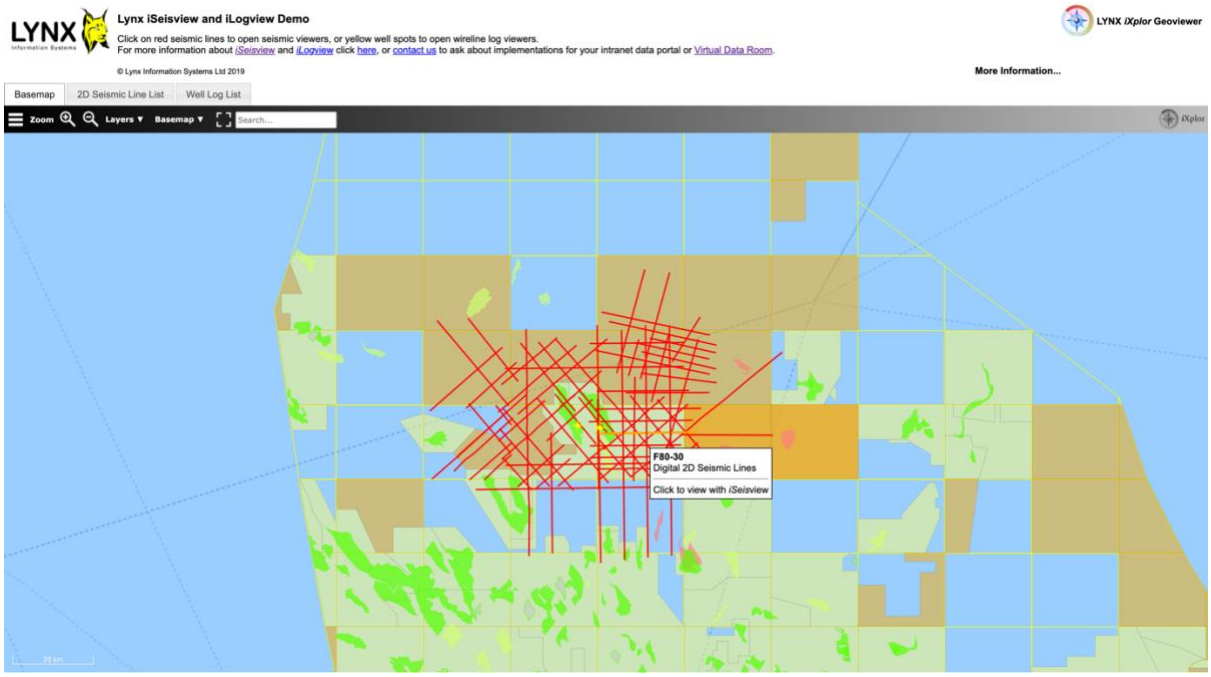


*Figure 2.2 Lynx's Geoviewer Demo with seismic viewers and well log viewers [16]*

## 2.2 Website application

The VDR and GIS are in the form of the website application. Therefore, it is vital to understand more about website applications and their development. The development of websites has also rapidly evolved since the first-time worldwide web was proposed by Tim Berners-Lee in 1989 [17]. The first purpose of the World wide web was to allow worldwide scientists to share information virtually. Aside from that, three essential technologies formed by Tim as the foundation of the web until nowadays are Hypertext Markup Language (HTML), Uniform Resource Identifier (URI), and Hypertext Transfer Protocol (HTTP) [18]. HTML is the formatting language for documents displayed in the web browser [18]. URI, also known as URL, is a unique naming system that identifies each resource on the web [18]. HTTP is the application layer protocol responsible for interacting users with web resources and transferring information between devices connected to the internet [18]. After Tim formed the foundation of the web, he made and deployed the first web page, which was simple and static without any pictures and primary fonts [17]. It was believed to be the only type of web page at that point in time. Fortunately, as time passes, many different design types of websites (collection of web pages) play an essential role in website development.

One of the common types is the static website which is similar to the first web page that Tim made, but it is less demanded as website technology and purposes have advanced [17]. The other common types are dynamic, responsive, content management system (CMS)-built, and eCommerce [17] [19]. A dynamic website is the upgrade of the static website where the web application's contents are editable, the website has many functionalities, and the user can interact with the website [17]. The only limitations of the dynamic are setting up the functionalities can be complicated, and the

website loads slower due to the compositions and elements present on each webpage [19]. This type of website is the active website that is available nowadays. The other type of website that is followed by many websites these days is responsive. It is famous as the responsive website can be accessed seamlessly from different devices, including mobile phones, and the website can adjust itself to fit the browser size [20]. A responsive website provides a fantastic experience for the user and increases productivity with this ability. However, there are still limitations such as browser compatibility issues when opening the website from the old browser version, slow load, and complications in developing the website considering perfect user experience [20]. Moving on to CMS-built websites, it is websites that allow non-programmers to add, modify and delete website pages along with their contents, such as texts and pictures [17]. A famous example of CMS-built websites is WordPress. The last common type of website is the eCommerce website that sells services and goods that customers can buy directly as the website is connected with a payment gateway [17].

In order to develop any types of websites mentioned above, developers need to understand the web application architectures. The web application architecture is the base of all components of the websites [21]. Developing the web application will be time-effective and cost-effective if the base is concrete [21]. The components of web application architecture are divided into two components which are structural (server) components and UI & UX (client) components [21]. The server component includes the web application's server and database [21]. At the same time, the client component provides for the layout of the elements of the websites, input controls, dashboards design, and many others [21]. These components are the general compositions of the system [21].

Aside from components of web application architecture, the layers of web app architecture are responsible for dividing the websites into layers to allow easy upgrade or modification of each layer independently [21]. The three layers of web app architectures are the presentation layer, business logic layer, and persistence layer [22]. The presentation layer is exposed to users using a browser and contains the user interface display and process components that allow communication with the system [22]. Next, the business logic layer, also known as the application layer, has the role of getting the user requests from the browser, processing the requests, and choosing the paths from where the data will be retrieved [22]. This layer encodes the functions of commutating the data and requests to the backend [22]. The last layer, the persistence layer, also known as the storage or data access layer, collects all the requested data and gives access to the application's data storage [22]. The persistence layer is linked to the business layer, so the logic recognizes which data storage has the requested information and enhances the process of retrieving it [22].

Some famous web application architecture patterns include client-server, Microservices, serverless, progressive web apps, and single-page applications [21].

- The client-server architecture pattern is the most common architecture where the client sends a request to a server, and as a response, the server will send data packets to the clients [21].
- The microservices architecture pattern is based on the concept of using an independent microserver to be responsible for a function in the system. As there can be many micro servers, each is connected with one other. This pattern is suitable for complex web applications as adding, maintaining, and testing the functionality of different parts of the system is straightforward [21].

- The progressive web apps (PWA) architecture pattern provides an advanced user interface for clients. These advanced features include running the web application offline, delivering push notifications for clients, and connecting with hardware's APIs [21].
- Single-page applications (SPA) architecture pattern allows clients to access all information from a single HTML page. With this pattern, the server loads are reduced significantly as only a component of the page area is reloaded based on clients' requests [21].

As there are many different architecture patterns, developers shall choose wisely. The website is guaranteed high performance, security, and trustworthiness with the right and excellent architecture pattern.

### 2.2.1 Front end development

Front-end development is the part of web development that concentrates on developing the presentation layer of web app architecture. It also involves converting the backend developers' code into a graphical interface and ensuring that the data displayed is reader-friendly and understandable [23]. In addition, front-end developers need to ensure that the websites work in various device types, OS, and different web browsers. Front-end developers need to consider that the website shall work on client applications, such as Chrome, Safari, Microsoft Edge, Internet Explorer, Mozilla Firefox, and Opera.

When people work as front-end developers, there are three technologies that they need to master. The three technologies are HTML, Cascading Style Sheets (CSS), and JavaScript [38]. As discussed before, HTML is the main component of a website that acts as a formatting language for documents to be displayed in the web browser. CSS is a set of instructions accountable for the

website's style, such as colors, animations, and layouts [23]. HTML and CSS are enough to create websites, but JavaScript is used to make the websites interactive. However, in the past, website performance is low. Therefore, Google presented the Chrome V8 JavaScript engine in 2008 that performs a lot faster than Internet Explorer [24]. JavaScript has excellent performance after the V8 engine was presented, similar to Java [24]. Therefore, the website can run rapidly. Because of the V8 JavaScript engine's high capabilities during that time, many JavaScript frameworks were formed afterward, which evolved the internet development [24]. Some JavaScript front-end frameworks available today are React Js and Vue Js.

React Js work by distributing webpage content as various components in the Document Object Model (DOM), and then the browser generates the component using JavaScript [24]. The advantage of React Js over traditional HTML websites is that React Js only re-render the part of the website that is updated rather than re-render all the parts of the website, which is time-consuming [24]. React Js can do so by comparing data bind with virtual DOM and displayed DOM. While Vue Js also uses DOM binding to update the content appearance of the website. The only difference between React Js and Vue Js is that React Js depend on functional programming principles where libraries manage state and communication between components [24]. In contrast, Vue Js provides built-in features and supporting libraries that make the development experience more seamless [24]. However, there are particular abilities that the Vue Js are more formidable than React Js, such as optimization efforts and familiar beginner-friendly templates [25]. Better optimization efforts as there is a function in the Vue component that is automatically implemented to know which components need to be re-render when state changes, while React Js require the developers to implement the function first or it will not be implemented [25]. Vue Js provide HTML-based templates that allow developers with experience in HTML to quickly adapt, while

React uses JavaScript and JSX (an XML-like syntax) that function in JavaScript [25]. Vue also allows developers to implement CSS in <style> component familiar with HTML component and can easily see with other components as all the components are present in a single file [25]. With these advantages, Vue Js is an excellent choice for developers, especially newbies.

### 2.2.2 Vue Js

Vue.js is a famous JavaScript open-source front-end framework invented by Evan You in 2014 [24]. The initial purpose of Vue being developed is to provide reactive data binding and UI components using an uncomplicated Application Programming Interface (API) [24]. Although a single-page application has limited functions and is hard to implement in commercial use, Vue can operate complex single-page applications, including state management, routing, and build tooling, with the support of third-party libraries and packages [24].

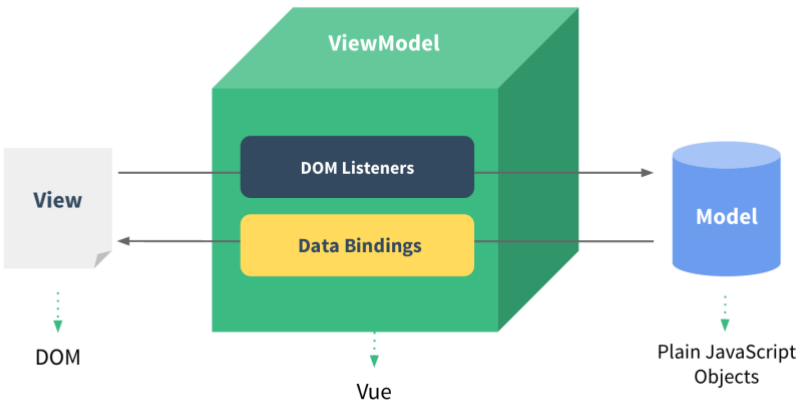


*Figure 2.3 Vue Js data-driven concept [26]*

The Figures above demonstrates that Vue.js retains three units to process data-driven, including View, View Model, and Model. The view unit is the displayed DOM that contains the website

content [24]. View Model unit contains Data Bindings and DOM listeners. It becomes a middleware responsible for the transmission between View and Model. Then, Vue uses the DOM listeners in the View Model unit to observe and update the data in the Model unit when users begin the data processing in the View unit [24]. After the data in the Model unit is updated, the DOM binding is used to update the display of the website content [24]. Therefore, it can be concluded that the one-way binding with DOM listeners in Vue has the same ability as two-way binding [24].

### 2.2.3 UI/UX Frameworks

In order to make the front end of the web application attract visitors, excellent UI/UX needs to be implemented. Therefore, this thesis's UI/UX frameworks are Materio, Material Design, Vuetify, and Material Design Icons. Materio Admin is an open-source, user-friendly, and customizable admin template built with Vue Js and Vuetify [27]. It was created by ThemeSelection and came out in two versions [27]. Both versions of Materio are utilized with Vuex, Vue Router, Webpack, and Material design icons [27]. The features of the free version include [27]:

- a vertical menu,
- a dark and light skin layout,
- a dashboard,
- basic pages,
- basic cards,
- basic tables,
- a chart library,

- basic documentation,
- regular support,
- manual customization.

While the features of the pro version include [27]:

- a collapsible vertical menu and horizontal menu,
- Three dark and light skin layouts: Default, Bordered & Semi-dark,
- Three choices of dashboards,
- 5 API ready applications,
- advance form elements, validation, and the form wizard,
- basic cards, advanced cards, and statistics cards,
- advanced tables,
- advanced charts, including two chart libraries,
- Two choices of authentication pages and various operable pages like login page, pricing, FAQ, knowledge base, and others,
- multiple navbar & menu options,
- 3D characters and Illustrations,
- detailed documentation,
- starter-kit,
- access control, including the function CRUD,
- quick search enables fast navigation between pages using hotkey support,
- internationalization support in components,
- priority support.

With only the features of the free version, developers can create attractive and exclusive single-page applications. Therefore, the author and team chose the free version as the project template. The other advantage is that the free version uses the MIT license, the most straightforward license, which is declared free-forever [27]. The free version is also available to be run in the latest browsers such as Chrome, Safari, Firefox, and Opera [27].

The Vuetify is used to build the Materio, and the Material design icons that are utilized in Materio are part of the material design. Google developed material design to create a united style for all webs and mobile apps [28]. In definition, material design is a flexible design of guidelines, components, and tools to support user interface design [28]. Many different material design frameworks and libraries offer components and styles that are directly usable [28]. One of the material design frameworks is Vuetify. At the same time, one of the material design libraries is Material design icons.

Vuetify is a UI framework. The purpose of Vuetify is to provide developers with tools that help them create engaging user experiences [29]. Unlike other frameworks, Vuetify is designed from the very beginning to be developer-friendly [29]. It is developed based on Material Design specifications, thoroughly prepared every component [29]. The features provided by Vuetify include Accessibility, Bidirectionality (LTR/RTL), Breakpoints, Global configuration, Icons, Internationalization, Layouts, Presets, Programmatic scrolling, SASS variables, Themes, Tree shaking, and Weekly code release on fixing issues found by the community [29]. Additionally, Vuetify provides 18 months of long-term support for each major release [29]. When these features are compared with other Vue UI frameworks, as shown in the Figure 2.4. below, Vuetify has the most thorough features, so it is recommended to use Vuetify.

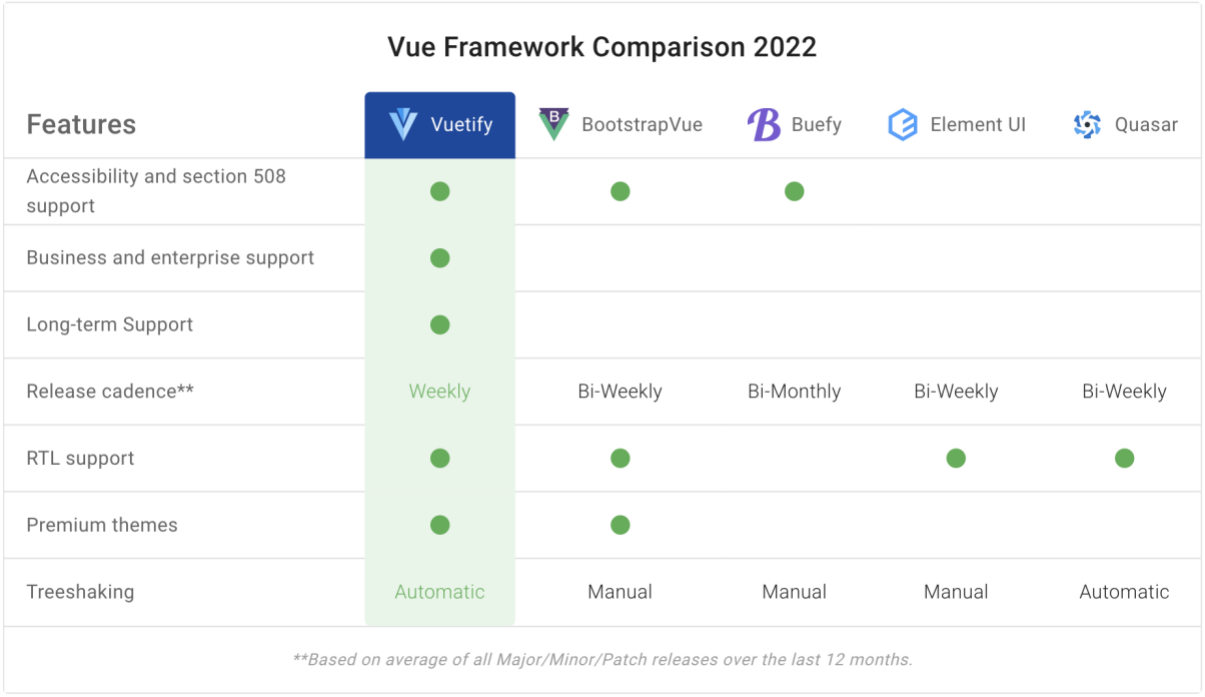


*Figure 2.4 Vue Framework Comparison 2022 [29]*

Material Design Icons is a library containing icon collections that can be downloaded in any format, color, and size according to the designers' or developers' projects' needs [30]. It can be utilized in various frameworks, including Vue and React [30]. It is maintained by Austin Andrews, and the general license of Material Design Icons is Pictogrammers Free License [30]. However, some icons are redistributed with Apache 2.0 license while others are redistributed with their respective license [30]. All fonts are also redistributed with Apache 2.0 license, and all non-font and non-icon files in the Material Design Icons GitHub apply the MIT license [30]. Material Design Icons can be downloaded using npm or from the website [30].

### 2.2.4 Human-computer interaction (HCI)

Talking about UI/UX, it is crucial to understand Human-computer interaction (HCI). HCI is a study about the innovation of computer technology and the interaction between humans and computers [31]. HCI is extensive and related to specific fields such as user experience design and user interface design [31]. HCI can be described as the ancestor of user experience design in numerous manners [31]. In the past, HCI specialists helped explore and invent the concepts of graphical user interfaces of windows, mouse pointing, menus, and icons. They also experimented and developed the idea of audio and video use in user interfaces, interactive tutorials, context-sensitive help, and hypertext links. When HCI specialists explore and develop, they need to [32]:

- comprehend the psychological, organizational, and social aspects of the interaction between humans and computers,
- explore and create processes for making the suitable HCI design,
- discover the practical interactions between single users or groups.

As HCI specialists need to comprehend the psychological, organizational, and social aspects of the interaction between humans and computers, it is also necessary to examine the effect of physical, social, and cognitive environments in making interface design [32]. When examining the physical environments, it is essential to consider designing a system that protects the user's well-being, such minimize device radiation should be the priority of the interface designer [32]. Interface designers also need to design the interface while considering the condition of the working space, such as enough room, efficiency, lighting, low distraction, and low pollution [32]

As for the social environment, it needs to be considered in designing interfaces because different social environments may require different computing frameworks [32]. For example, personal computing is mainly done in-house or office. While mobile computing is usually done outdoors. Another example is the design of machines like ATMs must conserve users' privacy and security by providing a shield around ATM's keypad and requiring users to input pins to do transactions [32].

In addition, the Cognitive Computing Environment examines the cognitive elements of people's interactions with computers [32]. Cognitive science-based topics encompass fundamental psychological concepts like learning and problem-solving in terms of abilities, methods, knowledge, and styles [32]. The cognitive elements that need to be considered by the designer when designing the interface are the users' age, users' disabilities, users' expectations, and users technical knowledge [32]. For example, the design of the interface needs to be clear, straightforward, and zero error for a government application. In comparison, the design of the interface needs to be more soothing and enjoyable for music applications.

With all the elements of interaction considered when designing an interface, HCI models that cover all the elements have been developed and utilized until now. Some HCI models are personal computing, mobile computing, large-scale computing, networked computing, collaborative environments, augmented reality, and virtual reality [32]. These models can be analyzed and determine the elements used for user interaction by using the who, what, where, why, and how (5W+H) procedure [32]. The procedure is broken down into three components [32]:

- What/How: This procedure step is used to learn the models' physical and virtual interface components. For example, I/O devices, windows, and others,
- Where/When: This step of the procedure is used to determine which physical environments (ex: office, portable, wearable systems) are suitable to use the models,
- Who/Why: This procedure step distinguishes the types of tasks the models can do.

After the procedure applies to each of the HCI models, the differences between the function of HCI models, the purpose of HCI models, and other elements of the HCI models are observable.

**2.3 GIS**

One of the examples of VDR mentioned in section 2.1 followed the concept of Geographic Information System (GIS) to visualize the data chosen based on the locations of the data on the map. GIS is a system of software, hardware, and data that enables the exploitation, analysis, and display of data and information based on the location on the surface of the Earth that is connected to the data, as mentioned in section 1.1 [11]. There are many other definitions of GIS established by different viewpoints. However, GIS can be understood by analyzing the three letters of the acronym GIS, which is explained below [11]:

- The letter 'G' represents Geographic which indicates an attraction to a terrestrial location of a particular substance on, beneath, or beyond the surface of the Earth.
- The letter 'I' represents Information that indicates the importance of data to be managed into helpful information that helps in decision-making.
- The letter 'S' represents a System that indicates the necessity for a team, computer hardware, and procedures (data gathering, processing, and visualization) to produce the information that helps in decision-making.

Up to this point in time, GIS has undergone many adaptations, which are [11]:

- Paper Mapping spatial analysis with Cholera Clusters where Dr. John Snow, a British physician, started mapping the locations of Cholera outbreaks, including the roads, surrounding buildings, and waterlines, in 1854. The mapping was done by putting geographic layers on a paper map. When he mapped these features, he distinguished that Cholera cases were commonly found along the waterlines. The map represents the significant relation between geography and public health safety with these findings.
- Before 1960, it was known as the Dark Ages of GIS due to computer mapping was not developed, so all mapping was done on paper. All the features are drawn on paper which can be complicated for cartographers
- 1960 to 1980s is known as the pioneer time of GIS due to the transformation of mapping from paper to computer.
    - In the 1960s, Roger Tomlinson, known as the father of GIS, collaborated with the Canadian government, where he originated, arranged, and led the Canadian Geographic System (CGIS) development. The CGIS was defined as the roots of Geographic Information Systems by many people. CGIS was unique because it adopted a layer approach system to map handling. The CGIS is used to measure which Canadian lands are suitable for planting and cropping based on data of soil, drainage, and climate characteristics. Over the years, CGIS has been modified and improved to keep pace with technology.

    - In 1970, the United States Census Bureau started to analyze/count data digitally using a format that implements the main principle of GIS known as GBF-DIME (Geographic Base File – Dual Independent Map Encoding). Other than the ability to input data to be stored, the format also supported error fixing and choropleth mapping. Using this format, the US Census Bureau began to digitize the number of borders, streets, and metropolitan areas. Data stored digitally was a huge step forward in the history of GIS.
    - During this period, the Ordnance Survey in the UK also started to develop a topographic map regularly, and it continues till now.
- From 1975 to 1990 is known as GIS software commercialization, where GIS software vendors increased. Esri, the largest GIS software company globally, started to provide GIS services. The other things that happened during this period were conferences and published work which also increased the acknowledgment of GIS.
- From 1990 to 2010, companies or individuals started implementing GIS for different purposes. Although in the beginning, companies still hesitate to implement GIS. The GIS began to be recognized by people as a valuable tool that helps in analyzing and decision-making. The software was also able to handle both vector and raster data throughout the time. The data collected from space could also be implemented in GIS as more satellites are being launched into orbit
- From 2010 till now, people have been introduced to open-source GIS. With GIS that leads open-source GIS, people who have computers are given the freedom to use GIS software. Therefore, GIS is becoming more popular, and more advanced open-source GIS is being made.

With all the transformations of GIS throughout time, the GIS always provides advantages such as [11]:

- better decision making on what the locations are useful for,
- improved communication as GIS maps and visualizations provides a better understanding of the locations,
- better geographic data documentation,
- the knowledge of the data is richer after being analyzed using GIS,
- questions of the analyst regarding the geographic data can be answered quickly by the GIS visualization,
- in a GIS environment, layers can represent different data,
- specialized maps or visualize tools can provide prediction results based on data,
- GIS eliminates the multiple redundant maps set,
- map data is easier to search, explore and display,
- cost savings resulting from greater efficiency.

Nowadays, GIS is mainly in the form of the web. The standard features for web-based GIS are shown in Figure 2.5 [33]. In the Figure 2.5, triangles represent special GIS features, crosses represent non-GIS features, and diamonds represent special web mapping features [33].


*Figure 2.5 Probable standard features of a web-based GIS grouped by functionality [33]*

### 2.3.1 Online Map Services

In the present day, most web users use maps and geospatial data on a day-to-day basis. Web maps are issued by several national mapping agencies [34]. Online map services are broadly used for navigation, locating desired places, travel arrangements, concurrent traffic details, education, and

many others. [34]. Online map services publicize maps with raster or vector data along with satellite images and provide features for interactivity [34]. The features include the ability to zoom in and out of the map, the ability to pan the map, retrieve information, the ability to navigate, and many others. [34]. Some online map services, such as Google Maps, can be used worldwide. In contrast, some online map services, such as Baidu Maps, can only be used in China [34]. As there is more demand for online map services from web users, the providers of map services are competing with each other to improve their services [34]. The criteria of online map services to be distinguished as worldwide popular are if they stand in high ranking, cover globally, and have many users worldwide [34]. According to November 2018 map website with the most traffic in the world provided by Similarweb (www.similarweb.com), the online map services that fulfill the criteria above include Google Maps, OpenStreetMap, Here maps, and Wikimapia [34]. The traffic data of the online map services are shown in Table 2.5.1.1 [34].

*Table 2.1 The traffic data* [34]

| Online Map Services | Monthly Visits in Millions (August 2018-October 2018 | Percentage of Visits from Country with Greatest Popularity | Year Founded | Main Traffic Source on Desktop | Rank in the Similarweb on map website with the most traffic | Global Rank | Platform Used by Most People |
|---|---|---|---|---|---|---|---|
| Google Maps | 108 | USA (30.04%) | 1998 | Direct | 1 | 691 | Mobile |
| OpenStreetMap | 10 | Germany (14.72%) | 2004 | Direct | 6 | 1819 | Mobile/Desktop |
| HERE Maps | 16 | USA (19.23%) | 1985 | Social | 8 | 2021 | Desktop |
| Wikimapia | 9 | Russia (36.76%) | 2006 | Search | 10 | 2692 | Mobile/Desktop |

For better information, a definition for each online map service above is defined below [34]:

- Google Maps: It is developed by Google. The features of Google Maps include maps, 360◦ streets views, satellite imagery, actual-time traffic situations, and a route guide for traveling

in various ways (foot, trains, and other ways). Most Android phones have Google Maps as the default map app.

- OpenStreetMap (OSM): It is an open-source platform developed to provide the world with a free customized map. It originates from affordable portable satellite navigation instruments created against expensive and restricted access to worldwide geographic information platforms. OSM is categorized as Volunteered Geographic Information (VGI). Furthermore, OSM also allows the creation of custom maps from the vector geographic data provided.

- HERE Maps: It is developed by HERE Technologies. The company was known to have government clients and famous clients such as BMW, Amazon.com, and Facebook. The products sell or are licensed to the clients, including content for mapping, navigation services, and location services for GIS.

- Wikimapia: It was created by Alexandre Koriakine and Evgeniy Saveliev. It works as a geographic encyclopedia project where all worldwide geographical entities are marked and described in an interactive map.

### 2.3.2 GIS Libraries

As online map services mentioned in section 2.2.1 increase in demand, the development of web GIS technologies is also trending [33]. More and more open-source or commercial web-based GIS frameworks and libraries are available [33]. The most widespread web-based GIS frameworks are

ESRI's ArcGIS and CARTO's CartoDB.js [33]. Following the nature of the web, web-based GIS software also consist of server-side and client-side components [33]. There are three types of architecture for the components: thin client architecture, thick client architecture, and medium client architecture [33]. Most web-based GIS frameworks used thick client architecture, which had a concept that the client-side is responsible for rendering the raw spatial data [33]. However, it also had a rich server-side component with geoprocessing functionality [33]. Other than frameworks, the famous JavaScript web-based GIS libraries are Leaflet, Cesium, NASA Web World Wind, and Open Layers [33].

The Leaflet is a lightweight library with a shallow learning curve (the measurement of understanding level when developers use the library) [33]. It has extensive third-party extensions [33]. This library is suitable for projects requiring a competent Web GIS where extensibility is not required [33]. The Leaflet is also beginner-friendly as it has detailed documentation and a good support team [33].

The Cesium is a library that allows the transformation of vector data accurately between global and polar projections [33]. It is suitable for a project expected to deliver a complex result while it starts to be a simple project [33]. This is due to the potential increase of the learning curve throughout development [33].

Cesium is a library with lesser coverage of GIS features [33]. It is categorized as a virtual globe [33]. It is suitable for a project expected to deliver a complex result while it starts to be a simple project [33]. This is due to the potential increase of the learning curve throughout the time of

development [33]. The developer may need to explore and learn a lot by themselves as Cesium only provide essential documentation [33].

NASA Web World Wind is a lightweight library alternative of Cesium that allows the transformation of vector data accurately between global and eight polar projections [33]. The 2 of 8 projections supported by NASA are also supported by Cesium [33]. It is also categorized as a virtual globe [33].

Open Layers libraries include Open Layers 2 and Open Layers 3. These libraries are suitable to support heritage browsers [33]. It is also suitable for applications with statistical abilities as the libraries can have the functionality of searching and filtering [33]. The GIS features of Open Layers 3 are more thorough rather than other libraries, and the documentation is quite complete, so the learning curve is intermediate [33].

### 2.4 Open-Source Software

With all the frameworks and libraries mentioned in previous sections, open-source software (OSS) is popular to use. Open-Source Software (OSS) is software that gives liberty for anyone to review, alter, and enrich the source code [35]. OSS has ruled the world as nowadays virtual product, service, or platform is powered by it [36]. Aside from using OSS, many companies have acknowledged the benefits of OSS and started to create platforms for the source code development of the OSS. Companies such as Google established Google Code in 2005, which supports OSS developers with fundamental tools and allows them to share their projects with the public [35]. Afterward, Microsoft released CodePlex, which allowed engineers and computer scientists to share their ideas

and OSS projects, although Microsoft was known to oppose Open Source [35]. In 2018, Microsoft also acquired GitHub, the most prominent open-source development platform to this time [36]. Since then, Microsoft has become the world's most significant open source contributor based on the number of employees actively contributing to open source projects on GitHub [37].

Whether companies provide their software as OSS, companies contribute to OSS, or developers use OSS, there are many benefits. The benefit for companies making their software as OSS is reducing their development cost and innovation cost of the software as their software can receive contributions from talented developers around the world [38]. While the benefits for companies that contribute to OSS are attracting excellent developers to join their companies without spending money on recruiters and attracting developers to use the paid upgraded version of OSS as they might feel unquestionable loyalty after using the free version for some time [39]. In addition, the benefits for developers in using OSS are the development can be cost-effective, and they are given freedom and flexibility to explore the OSS and produce new unique ideas [40]. OSS is also secure for use as developers participating in the OSS constantly review code, build improvements, and fix issues [40]. Aside from that, developers that participate in the OSS can increase their problem-solving skills by being exposed to different coding styles and new ways of fixing issues [39]. Developers also can increase their communication skills by communicating with other developers with different backgrounds that participate in the same OSS [39].

OSS is preferred over proprietary or closed source software with all the benefits explained above. The term proprietary software is used to define software that only the stakeholders of the software have the right to copy, inspect, and modify the software along with the source code [35]. To use

proprietary software, users must sign a license declared the first time they open the software to agree that they will avoid doing anything with the software that its authors have forbidden [35]. As users need to sign a license to use proprietary software, users also need to accept the terms of a license when they use the OSS. The terms often describe how people can use, study, modify and distribute the OSS [41]. As a requirement, the terms inside open-source licenses shall retain the criteria established by Open-Source Initiative [42]. The criteria, which are also known as Open-Source Definition, include [43]:

- free redistribution,
- source code of the software shall also be distributed and provide free access,
- the license shall allow modifications and work based on existing works. The license also needs to state the approval of the modified software to be distributed under the same conditions as the license of the original software,
- the integrity of the author's source code,
- the license shall not discriminate against any person or group,
- the license shall not discriminate the purpose the software is used,
- distribution of license,
- the license shall not be specific to a product,
- the license shall not restrain other software,
- the license shall be technology-neutral.

Based on the criteria above, it can be concluded that open-source licenses do not limit the user's purpose of using the OSS in general. However, there are open-source licenses that require anyone who redistributed a modified program of an OSS to attach the source code and preserve the same

rights as the original; these licenses are called “copyleft” [35]. An example of the same rights that need to be preserved by the redistributors is they are not allowed to charge licensing fees for the redistributed modified OSS. The well-known copyleft license is GNU’s General Public License (GPL). When developers use GPL features in their software, they need to obey the terms and conditions below [44]:

- on the software, developers are not authorized to claim patents or copyright. Furthermore, developers must include and show intact GPL notices, a copyright notice, a GPL copy, and a warranty disclaimer on the software,
- it is forbidden to alter the license or add extra terms and conditions,
- developers are bound by the reciprocal agreement that obligates them to release the software’s source code and provide others with the right to modify and redistribute.

Aside from that, software with a GPL license can be sold by the users whether the software is still original or already modified [44]. However, GPL gives buyers of the software the right to release the software to the public with price or not [44].

Other than GPL, which is categorized as copyleft, the famous open-source licenses that are not categorized as copyleft licenses are BSD licenses, MIT licenses, and Apache licenses. All three licenses are categorized as permissive software licenses, which acquire minimal restrictions on how the software can be used, altered, or redistributed either for open source or proprietary by the users [45]. Although they are categorized as one, they have their own terms requirements. When users use BSD licenses, they need to keep the copyright notice, the disclaimer, and the limitation on all the redistributed source codes and documentation supporting the redistribution [46]. When promoting the software with BSD licenses, users also need to include written permission from the contributors to write their names in the copyright notice [46]. While the MIT license is identical

to BSD, MIT allows the user to include the name of the contributors in the copyright notice for promotion purposes [46]. On the other hand, Apache is quite different as the license agreement is long to provide clear concepts on how the software can be used and a description of patent rights [47]. Almost all the open-source software that the author and team used has an MIT license and Apache license.

### 2.5 Software Development Life Cycle

In order to successfully develop the website application more structural and faster, a Software development life cycle (SDLC) is used. SDLC is a set of processes utilized to develop, design, and maintain software projects and ensure that the result of the projects corresponds with the user requirement [48]. The processes of the SDLC, also known as phases, which are necessary for the project's development to be completed include requirement analysis, design, implementation, testing, and deployment and maintenance/evolution [49].

- Requirement analysis [49]

  It is the first phase of SDLC where the development team needs to grasp the detail of the client's requirement, document the requirement, and determine ways to apply the requirement.

- Design [49]

  In this phase, the development team usually requires developers to be creative and utter a solution on how to develop the software and system based on the requirement specification. The risks, project limitations, development period, budget, technologies to be used, and project's subtasks distribution are discussed in this phase.

- Implementation [49]

  In this phase, the development team implements all the client's requirements. Project member such as front-end developers starts developing an interactive Graphical User Interface (GUI) that satisfy the client's requirements and connect the GUI with the APIs. At the same time, backend developers start coding the application logic, web services, and software APIs that front-end developers will use.

- Testing [49]

  It is the last phase before the software is deployed and delivered to the client. Testing is carried out to determine whether the software works expectedly based on the design decision in the earlier phase or not. With thorough testing, high-quality software can be delivered and avoid many bugs in production, which means fewer maintenance costs.

- Deployment and Maintenance/Evolution [49]

  This is the final phase of SDLC, where the software will be deployed according to client use. Then a maintenance team is usually available to monitor and report any post-production issue to the development team. The issue, if available, might require a hotfix (develop a solution and deploy in a short time) or can be fixed in the next version of the software based on the severity of the issue.

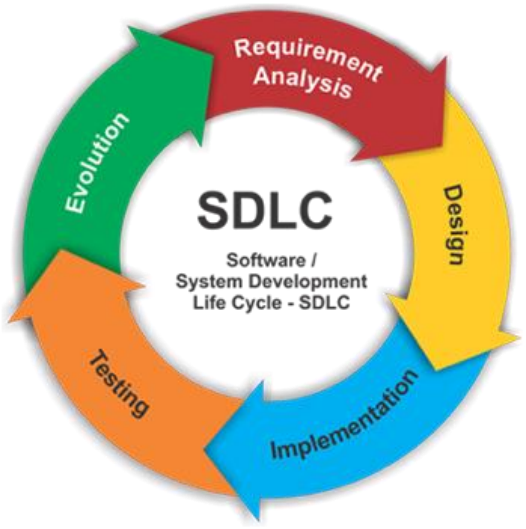


*Figure 2.6 SDLC Phases [50]*

Figure 2.1 shows the SDLC phases. The phases of SDLC can be undertaken slightly differently based on the SDLC models chosen to develop the software projects. Each model has different advantages and limitations when dealing with and considering certain conditions such as budget, project timeframe, and project requirements [51]. In addition, the SDLC models can be classified either as traditional models or agile models. A famous example of the traditional model is the waterfall model [52]. While some of the most popular agile models are extreme programming (XP) and Scrum.

In order to have a better understanding of the general difference between the traditional models and agile models, Table 2.2 is used to visualize the difference easily.

*Table 2.2 Difference between traditional models and agile models [52]*

| Traditional Models | Agile Models |
|---|---|
| Planning as a control mechanism | People's feedback as a control mechanism |
| In the early phase of the project, the requirements are exact. | In the early phase of the project, the requirements are still approximate. |
| Concentrating on documentation, less communication. | Communication is vital to develop the project in a short time. Documentation is negligible. |
| Follow detailed process | Limited process involved |
| Develop a complete project in a single cycle which takes a long duration | The iterative model allows development quickly in a short time |
| No scrum calls and stands up calls | Many scrum calls. Scrum calls are stand-up calls that happen recurrently to give updates on progress and receive feedback from clients |
| Understanding the project, developing, testing, and issue fixing takes a long time | Frequent contact with people involved in the project, iterative development, and accurate feedback have saved more time in each phase |
| Automation is not a usual approach | Continuous automated testing is recommended, and the use of it can assure better quality |

From the Table 2.2, it is clarified that agile models are better than traditional models. When teams or companies implement agile. they need to follow certain principles, which are [53]:

- customer is God, so the customers' requirements need to be prioritized through early and continuous development of the software,
- sudden changes are regular and need to be obeyed if it benefits the customer even though development is about to finish,
- update and deliver operating software frequently in various timespan but preferably short time span,

- developers and involved business individuals shall update each other's progress daily throughout the project,
- choose the trusted individuals that can be passionate about the project and companies to provide the support and environment that the individuals need,
- face-to-face communication is the best and most effective way of delivering information between individuals in a development team,
- published and useable software is the primary goal of the project,
- agile processes facilitate endurable development where all the stakeholders of the project can preserve a steady pace in doing their jobs,
- agility is improved if developers pay attention to technical superiority and designers pay attention to great design,
- simplicity is the best policy, which is essential to reduce unfinished work,
- a self-organizing team can gather outstanding requirements, produce good designs and architectures,
- during the daily meeting, each team member reflects on how to improve effectiveness, and then the member tries to adjust based on the things reflected.

By applying the principles, teams or companies can believe that the developed products will publish. Therefore, the author and team also decided to follow an agile model. The agile model chosen is Scrum.

Scrum is an agile model that implements iterative processes that result in significant project progress from each process [52]. This model is generally used to overcome ambiguous conditions

and tight deadlines. Most of the time, there are three roles when following the Scrum model: product owner, development team, and Scrum master [54]. The steps to be done when the scrum model is implemented are product backlog, daily Scrum, sprint, sprint planning meeting, sprint backlog, and sprint review meeting [54].

- Product backlog

  This is the first step where investors and project members gather all the business requirements and technical requests and list all the activities that will be developed during the project.

- Daily Scrum

  It is a quick daily meeting where project members give updates on the progress of the previous day's tasks, and it is also a time when the Scrum master assigns new tasks

- Sprint

  It is a period of one to four weeks where the development team start coding all the tasks listed when the Product Backlog

- Sprint planning meeting

  It is the meeting where the project members plan the things that need to be done in a sprint

- Sprint backlog

  It is a subgroup of Product Backlog where activities that need to be done during a sprint are listed down

- Sprint review meeting

  It is an after-sprint meeting where project members reflect on what was achieved, what does not go with the plan and the things that can be improved

From the steps, the advantages of Scrum are defined as productive communication among project members through the daily Scrum and constant feedback from product owners, which allows delivering the products accurately as expected [55]. The only disadvantage is Scrum is relative hard to master for people with no experience with Scrum [55].

### 2.6 Testing

Testing is the second last phase in SDLC which has an essential role in analyzing and ensuring the quality of a software product [56]. Testing is as complicated as developing the project because there are different analysis methods, different levels of testing, and different types of testing. As for this project, testing is part of the developers' scope, as mentioned in section 1.2.1. Therefore, it is vital to understand testing more. The two methods of analysis are static and dynamic analysis [56]. Static analysis is an analysis based on the details of documents such as product requirement documents, source code, design documents, and other documents [56]. Static analysis steps include complete code review after development, inspection, and algorithm research [56]. While dynamic analysis is an analysis that includes running the actual program with real input values and observing the performance and functionality [56].

Aside from analyzing, testing is essential to find defects [57]. Testing aims to find defects with different severities in a limited time and with lesser effort [57]. The testing measures include verification and validation to find defects and allow developers to fix the defects quickly [57]. Verification is to verify whether system acts are expected to pre-determined requirements and whether it is consistent [57]. Validation is to check whether the product build is already in accordance with what the user desired [57].

The activities that are accomplished during the testing phase of SDLC are [57]:

- test analysis,
- test design/plan,
- test execution.

With these activities accomplished, testing takes a lesser time and less cost.

In addition, the test execution can also be done at different levels. The levels are unit testing, integration testing, system testing, and acceptance testing [56]. Unit testing tests the specific units/components independently. Integration testing tests a combination of a few specific units/components together. This testing ensures that the units can function steadily when connected with each other. System testing tests all the software units altogether, and it is used to make sure that the software already works according to the pre-determined requirements. Acceptance testing tests the software from the user's perspective and ensures that the software can satisfy the users after being publicly published.

Aside from the levels, there are also types of testing: manual and automation testing [58]. Manual testing means manually executing the test cases [58]. It takes time, but testers do not need to understand any testing tools. Unlike automation testing, which uses certain tools to write scripts based on the test cases and runs the scripts of scenarios using some testing frameworks [58]. With automation, testing is faster and can avoid human error [58]. However, writing the scripts requires testers to have coding ability, and writing scripts can be complex when certain software functions have complicated flow or functionality [58].

# CHAPTER 3

PROBLEM ANALYSIS

This chapter will explore the features of existing applications and the proposed solution. Then follow with a comparison of the existing applications with the proposed solution. The existing applications and proposed solution, as mentioned in chapter 1, will be described further.

## 3.1 Existing Applications

This section further describes the features of the existing applications that integrated GIS. The existing applications, as mentioned in chapter 1, include Lynx's Virtual Data Room, INT's IVAAP, and Schlumberger's Prosource Front Office.

### 3.1.1 Lynx's Virtual Data Room

Lynx Information Systems has been operating since 1989 [59]. It offers geological & GIS services, software solutions, and data management for oil and gas industries worldwide [59]. They provide services that include VDR. Theirs VDR is designed to display oil and gas analysis datasets that are useful for farm-outs, data commercialism, and acquisition chances [15]. The VDR of Lynx is secure as only authorized users can access the data from anywhere using any web browser [15]. To be an authorized user, the user needs to sign confidential agreements issued by the issuing authority with the help of Lynx staff [15]. The user's activities are recorded and accessible to the issuing authority [15].

When authorized users log in to the VDR, they will see a map or list-based interface that displays certain data such as 2D seismic lines, 3D seismic, Wireline logs, and Well reports/seismic processing reports [15].

- 2D seismic lines displayed using Lynx *iSeis*view. The functionality includes the ability to select the display style, scale, and amplitude, with overlaid horizon/fault interpretations and intersecting lines/wells/3D surveys [15],
- 3D seismic is also displayed Lynx *iSeis*view using in the form of inline and crossline profiles with an interactive selection of display style, scale, and amplitude, with intersecting 2D lines/wells/other 3D surveys [15],
- Wireline logs displayed with Lynx *iLog*view in the form of interactive composite log display, select vertical scale, turn individual curves on and off, show formation tops and depth-correlated core images [15],
- Well reports, seismic processing reports in PDF documents [15].

These data need to pass quality control to be uploaded to the VDR. The user shall not be worried that the data is terrible as experienced Lynx staff will do the quality control and upload [15]. The Lynx VDR website is as shown in Figure 3.1.

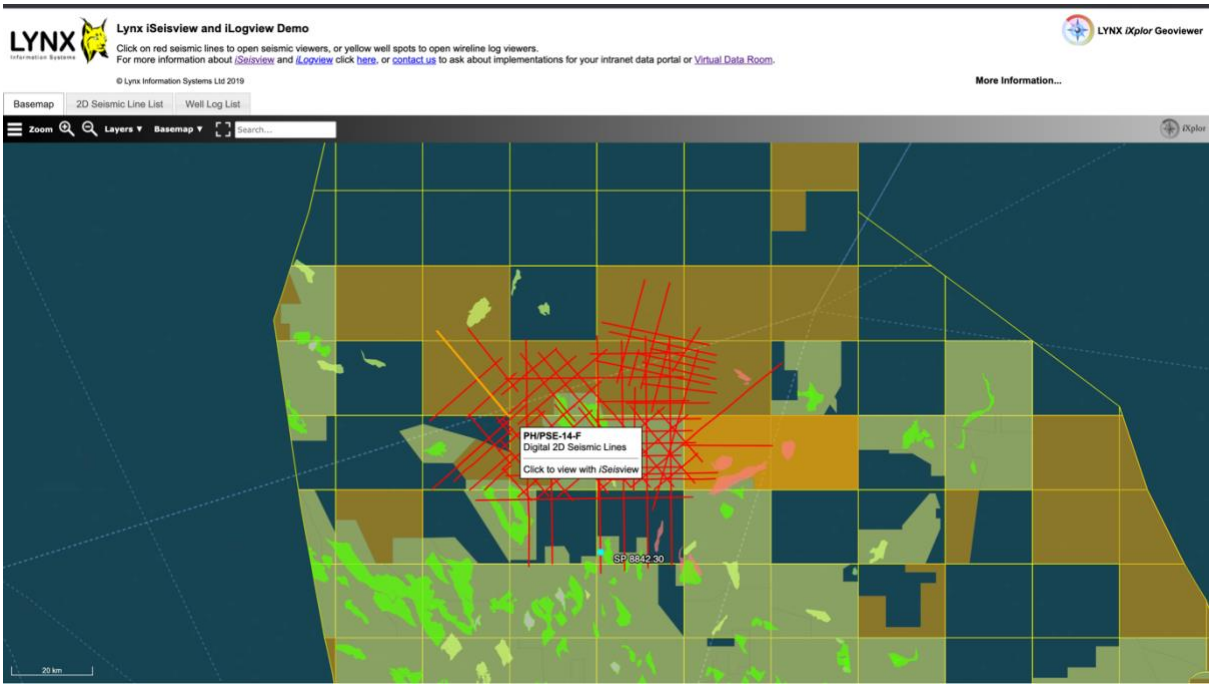


*Figure 3.1 Lynx VDR Demo [16]*

### 3.1.2 Interactive Network Technologies (INT) 's IVAAP

INT, similar to Lynx, has helped oil and gas industries visualize, monitor, and analyze their data since 1989. The product by INT includes INTViewer mentioned in previous sections. Other products are IVAAP and GeoToolkit [60]. One of the features of IVAAP has been integrated with GIS which allows users to visualize their subsurface data on a map along with the ability to filter data to be shown on the map [61]. The IVAAP also provides the ability to connect machine learning to the selected subsurface data [61]. IVAAP also allows easy connection to the source of the data with an API and SDK [61].

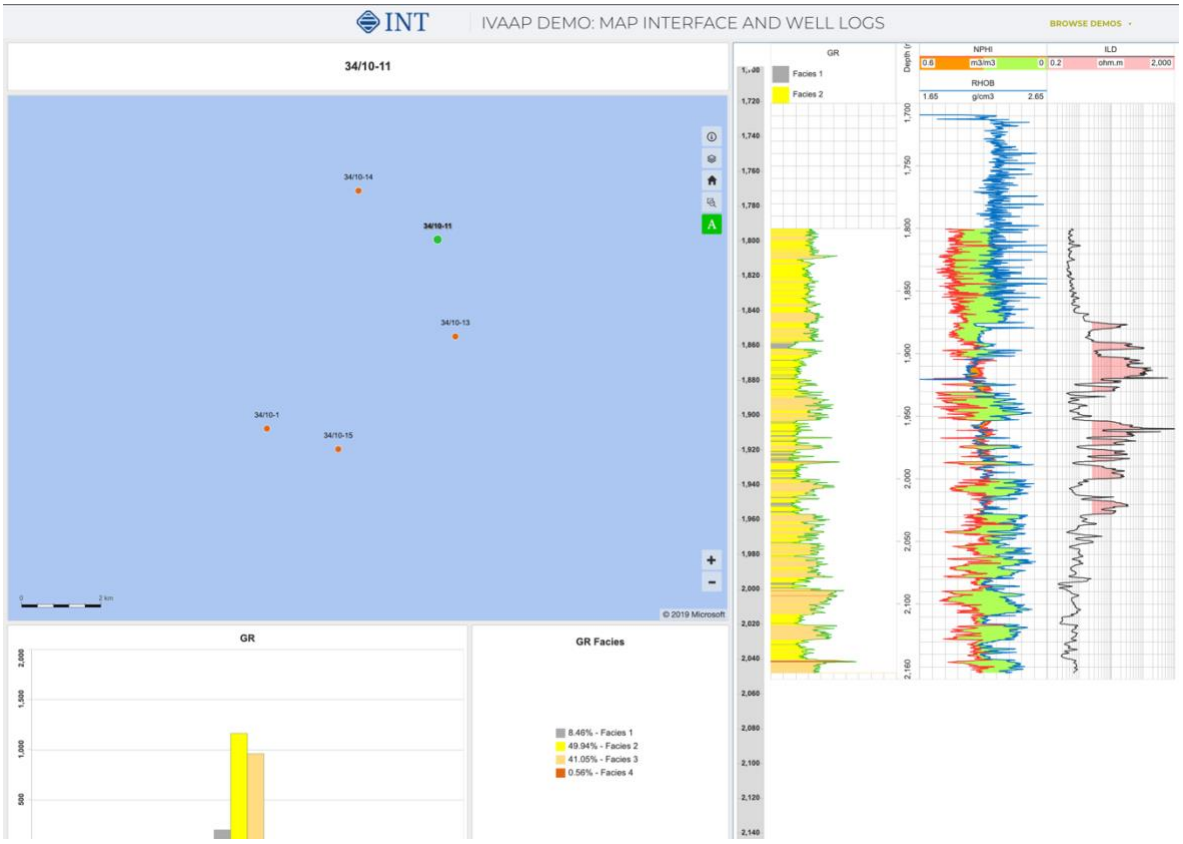


*Figure 3.2 IVAAP Demo on Map Interface and Well Logs [62]*

### 3.1.3 Schlumberger's ProSource Front Office

Schlumberger was built in 1926 with a vision to provide energy industries with richer and deeper data insights [63]. Schlumberger has provided many products throughout the time, including those mentioned earlier ProSource Front Office. The ProSource Front Office is compared with the author and team's VDR as it has an integrated GIS feature similar to the author's GIS. The GIS of ProSoruce Front Office includes visualizing oil and gas fields, wells, seismic lines, and seismic surveys in a location on a map [12]. Aside from that, the components on the map include map toolbox buttons, a map navigator, and a map right-click menu [12]. The users of ProSource Front Office can also pan, zoom in, and zoom out the map [12]. As shown in Figure 3.3, the map window contains several components represented by numbers.

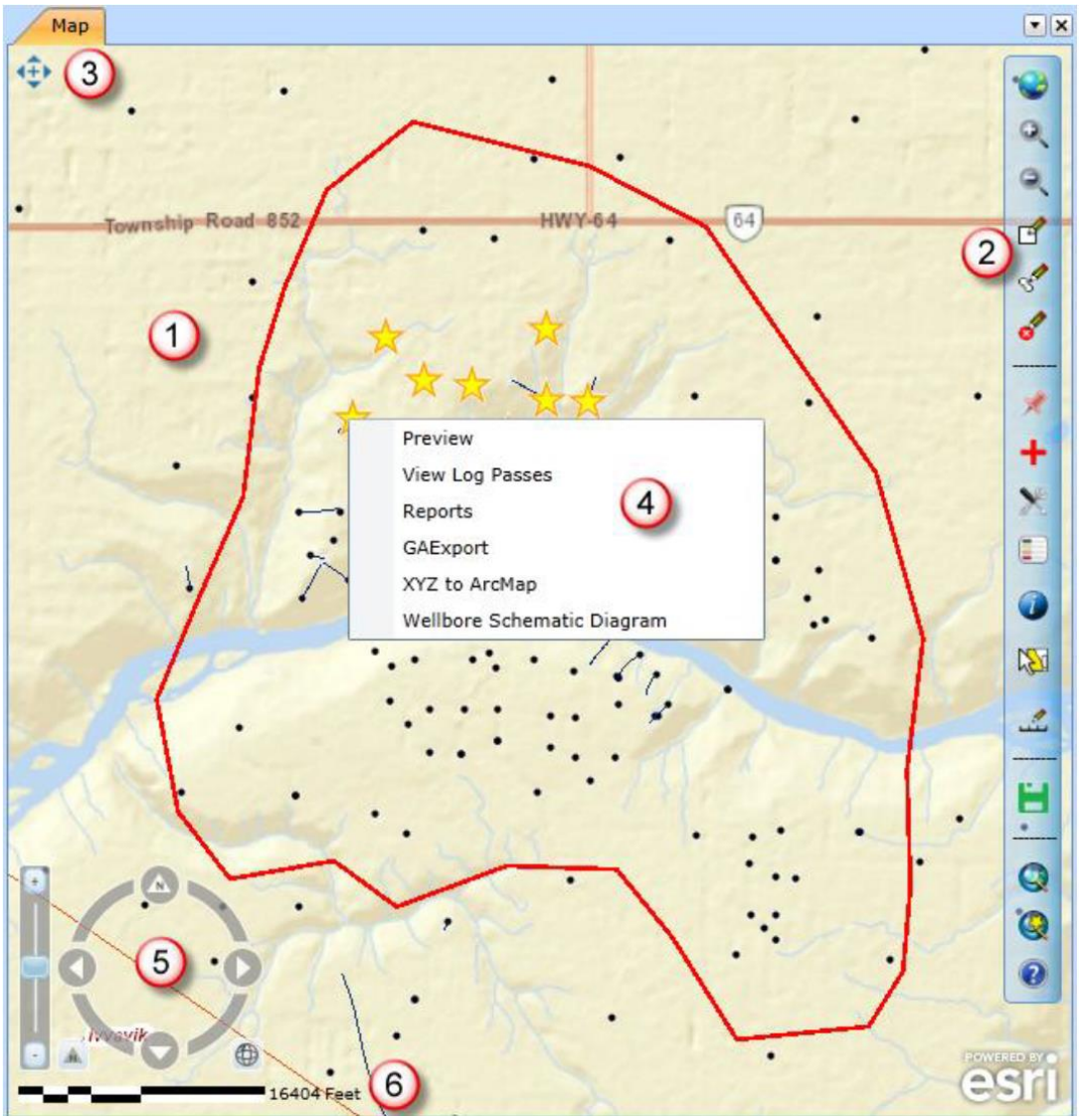


*Figure 3.3 Schlumberger ProSource Front Office's Map Window [12]*

Number one in the Figure 3.3, shows the graphical view of the fields, wells, seismic lines, seismic surveys, and leases [12]. Number two is the map toolbox buttons [12]. Each button has different functionality. For instance, a button is used to zoom in, another button is used to zoom out, another button is used to save the map display, and many others [12]. Number three is the open tool button which allows users to open the previously accessed toolbox map menu [12]. Number four represents the map right-click menu where users can choose an action based on the selected object

on the map [12]. Number five is the map navigator, which allows the user to move around the map and alter the scale [12]. Number six is the scale bar that shows the current scale [12].

### 3.2 Contribution

As the author and team decided to develop a cheaper VDR web application with GIS features than the existing applications mentioned above. The author and team came out with a solution to develop only the useful and demanded features by the oil and gas industries. The author and team use frameworks and libraries with open-source licenses to develop with no budget.

### 3.2.1 Comparison of existing applications with the author proposed solution

*Table 3.1 Comparison of existing application with the proposed solution*

| Features | Proposed Solution for the GS | Lynx | Interactive Network Technologies (INT) 's IVAAP | Schlumberger's ProSource Front Office |
|---|---|---|---|---|
| Cost | Cost-Efficient as developing it requires a lesser budget. Approximately Rp 500,000,000/application | $400/user/year | $1500/user/year | N/A |
| Filter the Map based on the oil and gas fields/wells | Available | Not Available | Available | Not Available |
| Connect to Machine Learning/API | Able | Not Able | Able | Not Able |
| Sort the Map based oil and gas volume | Available | Not Available | Not Available | Not Available |
| Overlay | Available | Available | Not Available | Not Available |
| A wide variety of data visualization/analysis can be chosen based on oil and gas location | Available | Not Available | Not Available | Not Available |

### 3.2.2 Proposed Solution

The solution mentioned in section 3.2 about using open-source frameworks and libraries to develop the features has left the author with many options in choosing the frameworks and libraries. However, the author decided to use Vue Js, Material Design, Vuetify, and Leaflet.

#### 3.2.2.1 Front End Framework Used

The author decided to use Vue Js due to certain advantages more than React Js. The advantages mentioned in section 2.2.1 include the optimization efforts and beginner-friendly templates Vue Js gave. As the author had previous experience with HTML, the author can quickly adapt to Vue templates. The author can also easily visualize all components in a single file, including CSS.

#### 3.2.2.2 UI/UX Framework Used

To support the front end, the author chooses UI/UX framework that support Vue Js, which is Vuetify that is supported with Material Design, as explained in section 2.2.3. Although many other Vue UI/UX frameworks exist, Vuetify has the most thorough features shown in Figure 2.4, which beat BootStrapVue, Buefy, Element UI, and Quasar. On another side, Material Design is chosen over other design systems due to some prestige, such as extensive documentation, providing flexibility & freedom for the designers to implement the design, and components provided assemble how it works in the real world but simplified [64]. Material Design is also chosen because it covers the trends in web designs, skeuomorphism, and flat design [65]. It mostly combines flat design elements with skeuomorphism's 3D touches, which means more interaction and great user experiences [65]. Therefore, the applications that implement Material Design will attract more users.

3.2.2.3 GIS Framework Used

As the author is responsible for integrating GIS to the VDR website application, the author has chosen Leaflet as the GIS library. Before the thesis, the author has no experience in implementing GIS. Therefore, Leaflet is chosen rather than another GIS library such as Open Layers because of Leaflet's superiority. The Leaflet superiority over Open Layers is shown in Table 3.2.

*Table 3.2 Leaflet vs OpenLayers [66]*

| Leaflet | OpenLayers |
|---|---|
| Shallow Learning Curve (Beginner-friendly) | Steep Learning Curve |
| More-structured documentation with a bunch of examples and tutorials | Extensive documentation but outdated |
| Higher community | Smaller community |
| Minimalist code when implementing | More code when implementing |

Moreover, the author used Vue2Leaflet to integrate Leaflet with Vue Js easily. Vue2Leaflet acted as a wrapper library for the Leaflet by providing vue-components that wrap almost all the Leaflet's functionalities [67]. With this, the author can just use vue-components to construct the map.

# CHAPTER 4

SOLUTION DESIGN

This chapter explains and displays the interaction between the website with the client and server and the flow of using the GIS. Then continue with the UI/UX design of the map components, and the function of the components is explained in this chapter. The scenario that will be tested, the route of the web application to the map page that the author is responsible for, and the functions to communicate between the website and backend are also explained.

## 4.1 Unified Modeling Language Diagram

For this thesis, the author and team use the Use Case Diagram, one of the Unified Modeling Language (UML) design techniques. The use case diagram is used to help developers to visualize the functionality of the system from the users' point of view [68]. The scenarios on how users interact with the system are illustrated [68]. The notations in the use case diagram are actor, use case, association, and system [68]. The actor represents the system's user, and it is illustrated with stickmen [68]. The use case represents a function of the system, and it is illustrated with ovals [68]. The association represents the relation between the actor and the use case, and it is illustrated with lines [68]. The system is illustrated with a rectangle [68].

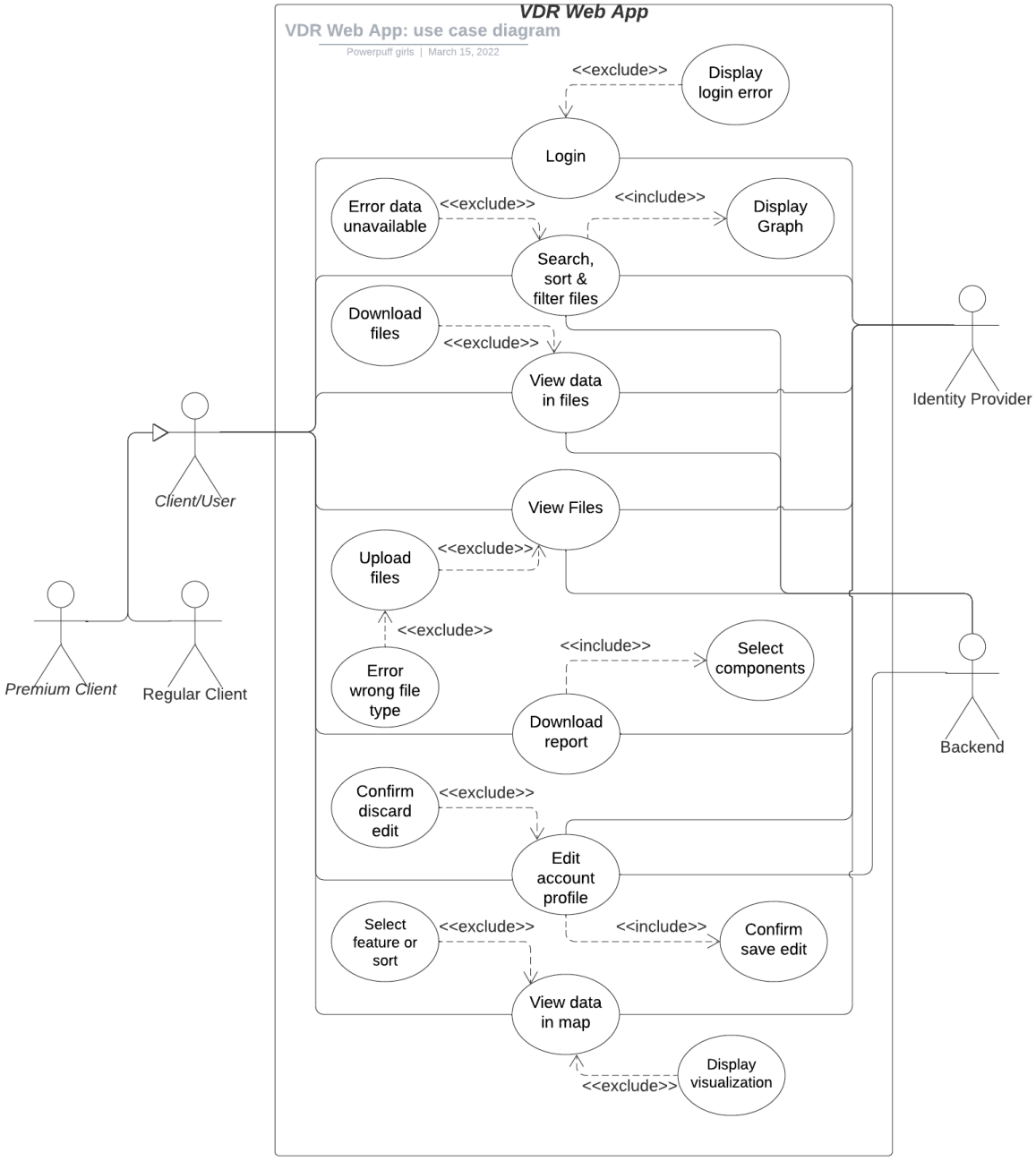


*Figure 4.1 Use case diagram of the Client*

Figure 4.1 shows the use case diagram for the thesis where the actor represents the premium or regular clients. The rectangle section represents the system, and inside it is the web application's functions that the client can do. There are seven functions, and the author is responsible for one of

the functions. Each function must surpass the identity provider to verify that the client has access to the function they want to access. Then each function also needs to request from the backend to get data to be displayed and communicate with the backend when updating the data. The function that the author is responsible for is the ‘View data in map’ with two optional use cases.

## 4.2 Flowchart

Aside from the use case diagram, the author uses a flowchart to illustrate the computer system, algorithm, and process [69]. With a flow chart, processes from various fields can be easily recorded, examined, arranged, recovered, and understood [69]. It is the most frequent diagram used by technical or non-technical fields [69]. Flowcharts use symbols such as ovals, trapeziums, rectangles, and diamonds to represent the type of steps and arrows to represent the direction of flow [69]. The ovals symbol defined the start and end of the flow. The trapeziums symbol defines the input or output. The rectangles symbol defined a process. The diamonds symbol defined decisions (yes or no).

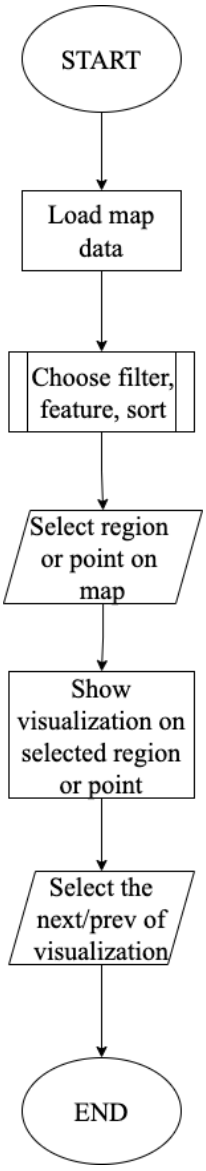


*Figure 4.2 Map Flowchart*

Figure 4.2 shows the flowchart of the process that the author is responsible for developing and the process that the clients/users can do. In the Figure 4.2, a rectangle with vertical lines on the right and left sides represents the predefined processor, shown in Figure 4.3.

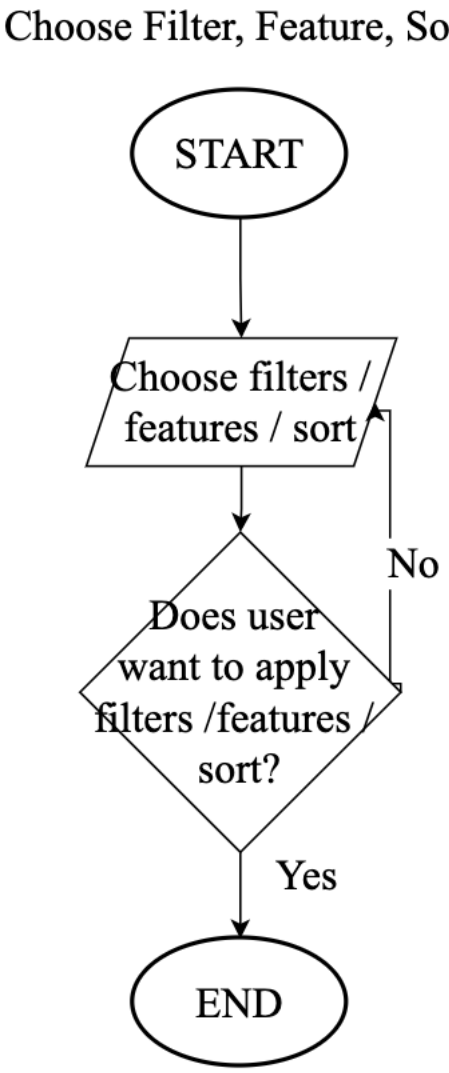


*Figure 4.3 Predefined Processor*

**4.3 User Interface / User Experience Design**

After certain meetings and discussions, the author and team have made the UI/UX design for the VDR website application. The design of the website application page that integrated the GIS is displayed in Figures 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, and 4.11.

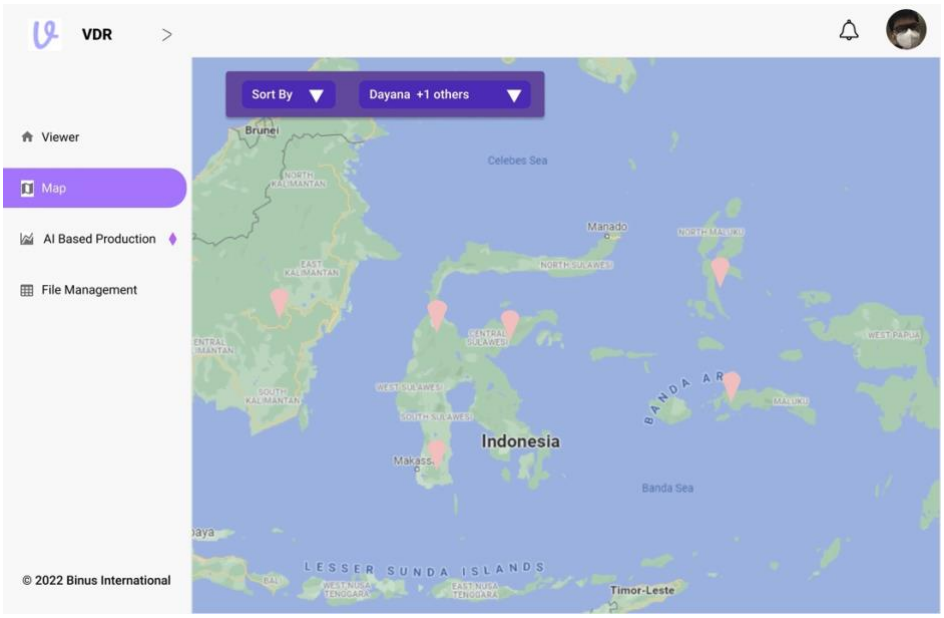


*Figure 4.4 Design of the Map Page*

*Figure 4.5 The Tooltip on Sort*

*Figure 4.6 The Tooltip on change when chosen during Filter*

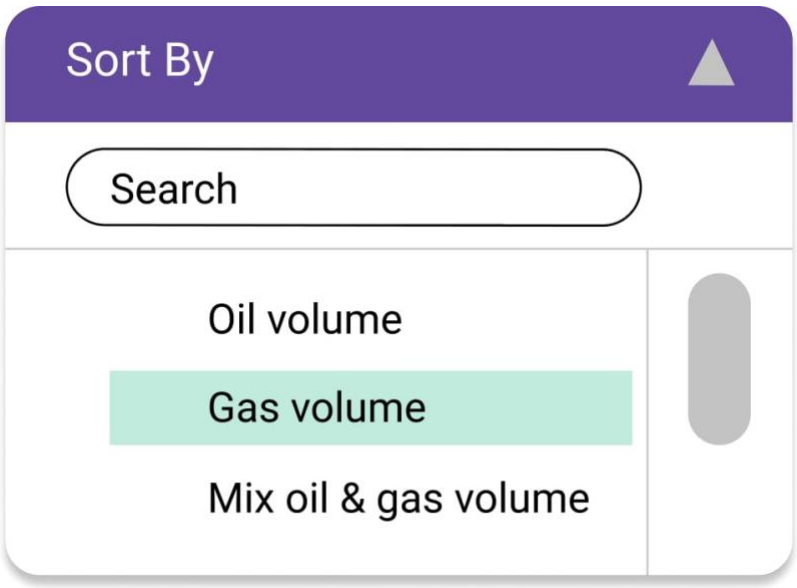


*Figure 4.7 Sort Dropdown in Map Page*

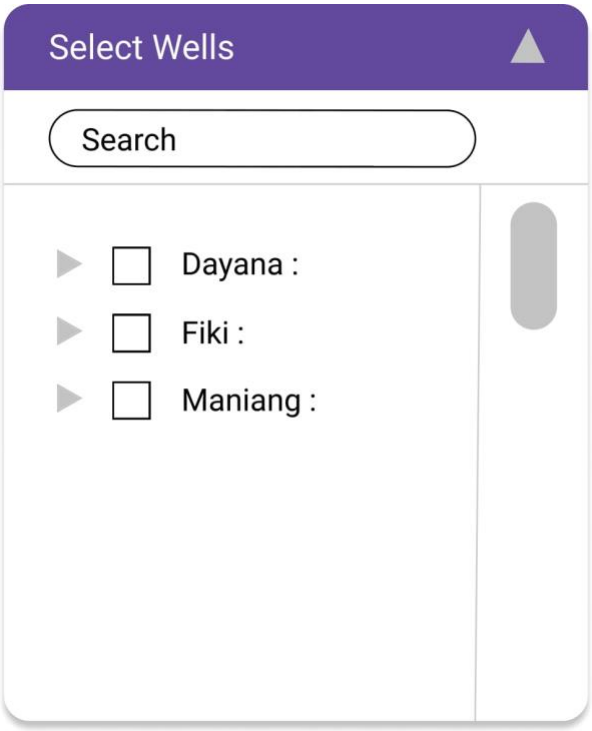


*Figure 4.8 Filter Dropdown in Map Page when Initial Open*

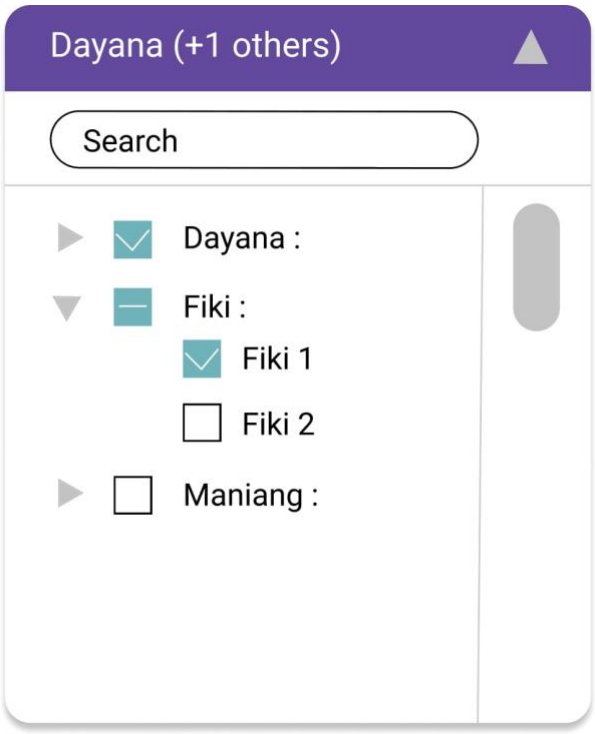


*Figure 4.9 Filter Dropdown in Map Page when Value Chosen*

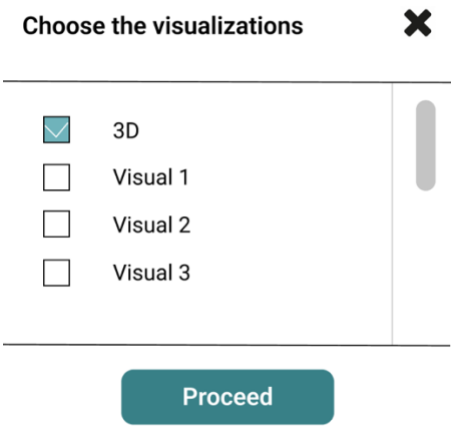


*Figure 4.10 Choose which Visualizations of the Data want to be Displayed*

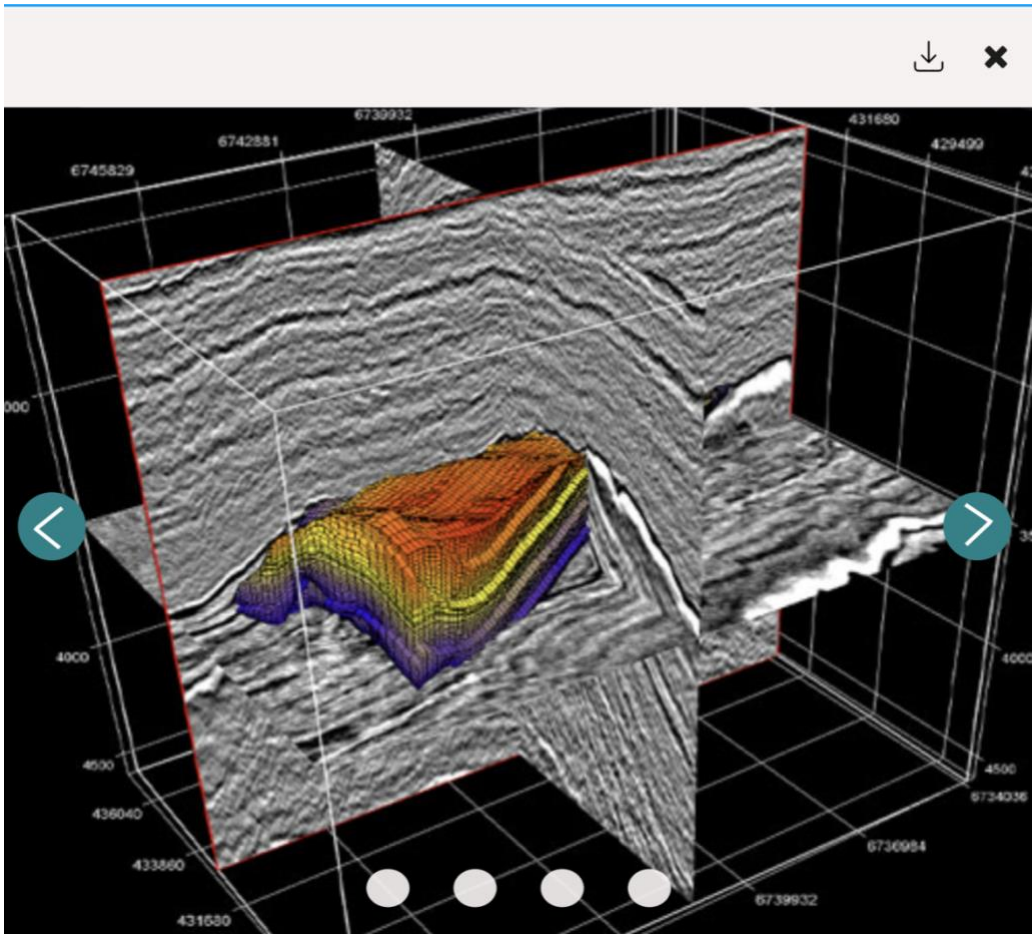


*Figure 4.11 Map Overlay with chosen visualizations*

### 4.4 Map Features

This section explains the features on the map page

#### 4.4.1 Sort the data of the map

The sort dropdown is made using Vuetify's autocompletes, where it provides a dropdown with search the value of the dropdown and other various abilities. The ability used for this feature is clearable, which allows users to clear the value of the sort when they do not want to implement it. The values that can be chosen are Oil Volume, Gas Volume, and Mix oil & gas volume, where the toggle of the data based on the volume chosen will be sorted. The color of the toggle will be dark

when the volume is high. The color of the toggle will be light when the volume is low. An example of the range of the colors is shown in Figure 4.2. Other than that, the sort dropdown can be moveable and resizable as it is put inside the VueDraggableResizable component.

### 4.4.2 Filter the data of the map

The filter dropdown uses Vuetify's text fields and tree view. The text field is used to search through the filter values. At the same time, the tree view allows the users to choose multiple values, including the data with parent and child structures. The filter values are the fields and wells of oil and gas drill locations. The toggle of the values chosen will resize to be bigger. The filter dropdown also used the VueDraggableResizable component to be moveable and resizable.

### 4.4.3 Hover and click the toggle

The toggle represents the location of the data. The toggle, when on hover, will change the image of the toggle. While off hover, the toggle will return to normal. When the toggle is clicked, the card overlay will show a checkbox where users can choose which visualizations they want to see, as shown in Figure 4.7.

### 4.4.4 Choose the visualizations of the data and display the image visualizations

Once users choose the visualizations and click the button in Figure 4.7, another card overlay will pop up and display the image visualization, as shown in Figure 4.8. The other function in Figure 4.8 is the download button where users can download the image visualization. The other common function of Figure 4.7 and Figure 4.8 is the close button to close each overlay.

## 4.5 Testing

One of the activities during the test phase in SDLC. During the test phase, there are three activities: test analysis, test plan, and test execution, as explained in chapter two. The test analysis is an activity that involves acquiring and making test objectives so that test cases can be settled. The author used Miro to make the test analysis in the form of a mind map. Each branch of the mind map shown in Figure 4.12 represents the scenario to be tested. Based on the scenario, the components that need to be tested include the map component, filter component, sort component, and card component of the map detail.

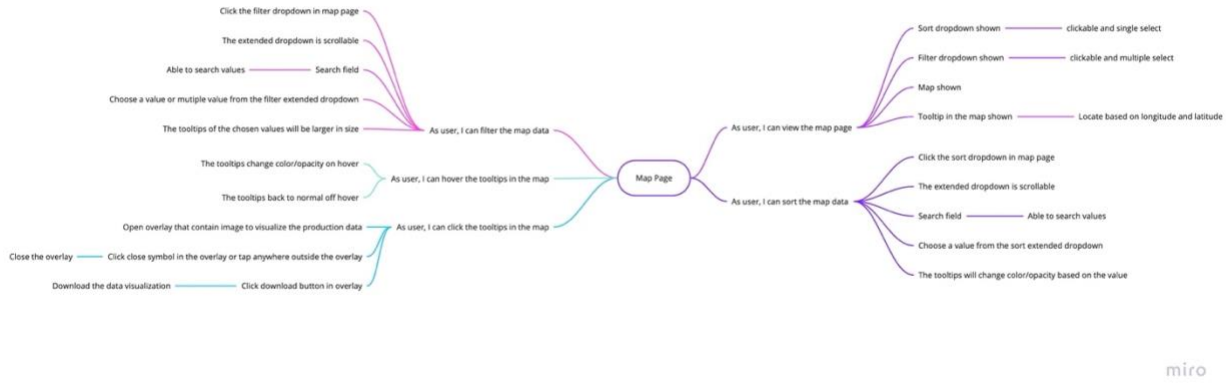


*Figure 4.12 Test Analysis Mind Map*

While in the test plan activity, the author and team used Jira to document the scenario wanted to be tested, which is defined earlier in the test analysis. Then during the test execution activity, the author prepared to do is unit testing and integration testing. Unit testing is suitable for Vue as the test script can be written straightforwardly for each component in isolation [70]. Developers can also quickly develop new components and write the test script for the component without affecting the other components' test script [70]. Some unit test frameworks support the developers' unit testings: Jest and Mocha [70]. The author chooses jest to test the components of the map page of the VDR web application due to its simplicity [70].

On the other hand, integration testing is required to test if all components will work expectedly as one. The integration test frameworks suitable for the Vue project are Vue Testing Library (@testing-library/vue) and Vue Test Utils [70]. The author chose Vue Testing Library as it is light and focused more on the components working expectedly without the need to test the detail of the components [70].

4.6 routing (Vue route)

4.7 communication with backend (Axios)

## REFERENCES


| | |
|---|---|
| [1] | U. Ali, "The history of the oil and gas industry from 347 AD to today," OFFSHORE TECHNOLOGY, 7 March 2019. [Online]. Available: https://www.offshore-technology.com/comment/history-oil-gas/#:~:text=The%20modern%20history%20of%20the,thicker%20oil%20suitable%20for%20lubrication. |
| [2] | "Why oil is important," UK Oil & Gas PLC, [Online]. Available: https://www.ukogplc.com/page.php?pID=74#:~:text=Oil%3A%20lifeblood%20of%20the%20industrialised,people%20all%20over%20the%20world. |
| [3] | D. Yergin, The New Map: Energy, Climate, and the Clash of Nations, New York: Penguin Group and U.S. Council on Economic Advisors, 2020. |
| [4] | P. Yasmin, "6 Pemanfaatan Minyak Bumi bagi Kehidupan, Apa Saja?," detikFinance, 09 February 2021. [Online]. Available: https://finance.detik.com/energi/d-5367484/6-pemanfaatan-minyak-bumi-bagi-kehidupan-apa-saja. |
| [5] | PwC Indonesia, "Oil and Gas in Indonesia," September 2019. [Online]. Available: https://www.pwc.com/id/en/energy-utilities-mining/assets/oil-and-gas/oil-gas-guide-2019.pdf. |
| [6] | BADAN PUSAT STATISTIK, "STATISTIK PERTAMBANGAN MINYAK DAN GAS BUMI 2014 - 2019," p. 12, 2020. |
| [7] | Y. Fan, K. Gao, J. Chen, W. Li and Y. Zhang, "Low-cost PMMA-based microfluidics for the visualization of enhanced oil recovery," *Oil & Gas Science and Technology – Revue d'IFP Energies nouvelles*, vol. 73, p. 26, 2018. |
| [8] | LYNX Information Systems, "Lynx Software - Licence Pricing," LYNX Information Systems, [Online]. Available: http://www.lynxinfo.co.uk/download-pricing.html. |
| [9] | INTViewer, "INTViewer Rapid Geoscience QA/QC," INTViewer, [Online]. Available: https://www.int.com/products/intviewer/. |
| [10] | INT, "Building an Upstream Data Management and Visualization Application," INT, [Online]. Available: https://www.int.com/customer_stories/pertamina/. |
| [11] | Geodwipa Teknika Nusantara, "Geodwipa Teknika Nusantara," [Online]. Available: https://ptgtn.com/. [Accessed 5 April 2022]. |
| [12] | E. Ali, "Geographic Information System (GIS): Definition, Development, Applications & Components," 2020. |
| [13] | Schlumberger, ProSource Front Office, Schlumberger, 2013. |
| [14] | B. Harrison, "The data room," *Developments in Petroleum Science*, vol. 69, pp. 21-26, 2020. |
| [15] | M. Lewis, "The Importance of a Virtual Data Room Index and Folder Structure," DealRoom, 2 March 2022. [Online]. Available: https://dealroom.net/blog/the-importance-of-a-virtual-data-room-index-and-folder-structure. [Accessed 26 March 2022]. |

| | |
|---|---|
| [16] | Lynx Information Systems, "Lynx Virtual Data Rooms," Lynx Information Systems, [Online]. Available: http://www.lynxinfo.co.uk/vdr.html. [Accessed 26 March 2022]. |
| [17] | "Lynx iSeisview and iLogview Demo," Lynx Information Systems, [Online]. Available: https://data.lynxinfo.co.uk/apps/iseisview/nethpublic#zoom=9&lat=7196871.98607&lon=449014.9111&layers=0B000TTTFTT. [Accessed 26 March 2022]. |
| [18] | C. Maxwell, "The Evolution of Website Development," terrostar, 5 November 2020. [Online]. Available: https://terrostar.com/the-evolution-of-website-development/. |
| [19] | " History of the Web," WORLD WIDE WEB FOUNDATION, [Online]. Available: https://webfoundation.org/about/vision/history-of-the-web/. |
| [20] | M. Storm, "6 Types of Web Design to Consider for Your Website," WebFX, 14 June 2021. [Online]. Available: https://www.webfx.com/blog/web-design/types-of-web-design/. |
| [21] | F. Almeida and J. Monteiro, "The role of responsive design in web development," vol. 14, no. 2, pp. 48-65, 2017. |
| [22] | M. Lozhko, "Web Application Architecture: Best Practices and Guides," LANARS, [Online]. Available: https://lanars.com/blog/web-application-architecture-101. [Accessed 28 March 2022]. |
| [23] | altexsoft, 25 July 2019. [Online]. Available: https://www.altexsoft.com/blog/engineering/web-application-architecture-how-the-web-works/. [Accessed 28 March 2022]. |
| [24] | "What is Frontend Development?," Techslang, [Online]. Available: https://www.techslang.com/definition/what-is-frontend-development/. |
| [25] | Y. K. Xing, J. P. Huang and Y. Y. Lai, "Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development," *ACM International Conference Proceeding Series*, pp. 68-72, 2019. |
| [26] | Vue.js, "Comparison with Other Frameworks," [Online]. Available: https://v2.vuejs.org/v2/guide/comparison.html?redirect=true. [Accessed 1 April 2022]. |
| [27] | "Overview," Vue.js, [Online]. Available: https://v1.vuejs.org/guide/overview.html. [Accessed 13 March 2022]. |
| [28] | "Materio – Free Vuetify Vuejs Admin Template," ThemeSelection, [Online]. Available: https://themeselection.com/products/materio-free-vuetify-vuejs-admin-template/. [Accessed 13 March 2022]. |
| [29] | MATERIAL DESIGN, [Online]. Available: https://material.io/. [Accessed 13 March 2022]. |
| [30] | J. Leider, "Why Vuetify?," Vuetify, [Online]. Available: https://vuetifyjs.com/en/introduction/why-vuetify/. [Accessed 13 March 2022]. |
| [31] | A. Andrews, "Material Design Icons," [Online]. Available: https://materialdesignicons.com/. [Accessed 13 March 2022]. |
| [32] | J. M. Carroll, "2. Human Computer Interaction - brief intro," in *The Encyclopedia of Human-Computer Interaction, 2nd Ed.*, pp. 21-62. |
| [33] | R. Valverde, Principles of Human Computer Interaction Design, Lambert Academic Publishing, 2011. |

| | |
|---|---|
| [34] | G. Farkas, "Applicability of open-source web mapping libraries for building massive Web GIS clients," *Journal of Geographical Systems*, vol. 19, no. 3, pp. 273-295, 2017. |
| [35] | A. Skopeliti and L. Stamou, "Online map services: Contemporary cartography or a new cartographic culture?," *ISPRS International Journal of Geo-Information*, vol. 8, no. 5, 2019. |
| [36] | "What is open source?," Opensource.com, [Online]. Available: https://opensource.com/resources/what-open-source. [Accessed 12 March 2022]. |
| [37] | M. Michels, "A Brief History of Open Source," 31 May 2021. [Online]. Available: https://maximilianmichels.com/2021/history-of-open-source/. [Accessed 11 March 2022]. |
| [38] | M. Asay, "Microsoft may be the world's largest open source contributor, but developers don't care–yet," TechRepublic, 4 November 2018. [Online]. Available: https://www.techrepublic.com/article/microsoft-may-be-the-worlds-largest-open-source-contributor-but-developers-dont-yet-care/. |
| [39] | Q. Jiang, J. Qin and L. Kang, "A Literature Review for Open Source Software Studies," vol. 9191, pp. 699-707, 2015. |
| [40] | A. Kubow, "A Business Case for Open Source – Why You Should Contribute to the Open Source Community," freeCodeCamp, 14 September 2021. [Online]. Available: https://www.freecodecamp.org/news/a-business-case-for-open-source/#:~:text=Contributing%20to%20Open%20Source%20Attracts,the%20upgraded%20or%20scaled%20versions. |
| [41] | Z. Staff, "THE BENEFITS OF OPEN SOURCE SOFTWARE," Zivtech, 12 July 2019. [Online]. Available: https://www.zivtech.com/blog/benefits-open-source-software. |
| [42] | S. Phipps, "License compliance is not a problem for open source users," Opensource.com, 25 October 2010. [Online]. Available: https://opensource.com/law/10/10/license-compliance-not-problem-open-source-users. [Accessed 11 March 2022]. |
| [43] | E. N. Hahn, "An Overview of Open-Source Software Licenses and the Value of Open-Source Software to Public Health Initiatives," vol. 32, no. 4, pp. 690-698, 2014. |
| [44] | R. J. Fakhlina and D. F. Saputra, "Development and Contribution of Open Source Software Communities for the Library Progress in Indonesia," vol. 5, no. 2, pp. 150-159, 2019. |
| [45] | R. SASS, "Top 10 GPL License Questions Answered," WhiteSource, 2 June 2021. [Online]. Available: https://www.whitesourcesoftware.com/resources/blog/top-10-gpl-license-questions-answered/. |
| [46] | "What is the MIT License? Top 10 questions answered," snyk, [Online]. Available: https://snyk.io/learn/what-is-mit-license/. |
| [47] | "What is the BSD License? Top 10 questions answered," snyk, [Online]. Available: https://snyk.io/learn/what-is-bsd-license/. |
| [48] | R. SASS, "Top 10 Apache License Questions Answered," WhiteSource, 6 May 2021. [Online]. Available: https://www.whitesourcesoftware.com/resources/blog/top-10-apache-license-questions-answered/. |
| [49] | T. J. Lehman and A. Sharma, "Software development as a service: Agile experiences," *2011 Annual SRII Global Conference*, pp. 749-758, 2011. |

| | |
|---|---|
| [50] | S. Barjtya, A. Sharma and U. Rani, "A detailed study of Software Development Life Cycle (SDLC) Models," vol. 6, no. 7, pp. 22097-22100, 2017. |
| [51] | J. Kwon, "What is Software Development Lifecycle (SDLC)?," 9 May 2019. [Online]. Available: https://joycekwon.medium.com/what-is-software-development-lifecycle-sdlc-523fd09340a6. |
| [52] | R. D. Amlani, "Advantages and Limitations of Different SDLC Models," *International Journal of Computer Applications & Information Technology*, vol. I, no. III, 2012. |
| [53] | S. Kumar and P. Dubey, "SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) ANALYTICAL COMPARISON AND SURVEY ON TRADITIONAL AND AGILE METHODOLOGY," vol. 2, no. 8, pp. 22-30, 2013. |
| [54] | "Principles behind the Agile Manifesto," Agile Manifesto, 2001. [Online]. Available: https://agilemanifesto.org/iso/en/principles.html. |
| [55] | B. V. d. M. Carvalho and C. H. Pereira, "Scrum agile product development method - literature review, analysis and classification," vol. 9, no. 1, pp. 39-49, 2011. |
| [56] | M. Sameen Mirza and S. Datta, "Strengths and Weakness of Traditional and Agile Processes --- A Systematic Review," vol. 14, no. 5, pp. 209-219, 2019. |
| [57] | N. Honest, "Role of Testing in Software Development Life Cycle," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 5, pp. 886-889, 2019. |
| [58] | T. Jindal, "Importance of Testing in SDLC," *International Journal of Engineering and Applied Computer Science (IJEACS)*, vol. 1, no. 2, pp. 54-56, 2016. |
| [59] | B. Kumari, N. Chauhan and H. H. Syed, "A COMPARISON BETWEEN MANUAL TESTING AND AUTOMATED TESTING," *Journal of Emerging Technologies and Innovative Research*, vol. 5, no. 12, pp. 323-331, 2018. |
| [60] | Lynx Information Systems, "About Lynx Information Systems," [Online]. Available: http://www.lynxinfo.co.uk/about.html . [Accessed 30 March 2022]. |
| [61] | Interactive Network Technologies, "INT Visualization Product Ecosystem," [Online]. Available: https://www.int.com/products/. [Accessed 30 March 2022]. |
| [62] | Interactive Network Technologies , "What Is IVAAP? [Overview]," YouTube, 25 July 2020. [Online]. Available: https://www.youtube.com/watch?v=UfSsktl7RaQ. [Accessed 31 March 2022]. |
| [63] | Interactive Network Technologies, "IVAAP DEMO: MAP INTERFACE AND WELL LOGS," [Online]. Available: https://www.int.com/ivaap-demos/map-interface-and-well-logs/. [Accessed 2 April 2022]. |
| [64] | Schlumberger, "Who We Are," [Online]. Available: https://www.slb.com/who-we-are. [Accessed 31 March 2022]. |
| [65] | C. Chapman, "Why Use Material Design? Weighing the Pros and Cons," [Online]. Available: https://www.toptal.com/designers/ui/why-use-material-design. [Accessed 1 April 2022]. |
| [66] | N. Babich, "Flat vs. Material vs. Skeuomorphic Design Examples," Adobe, 24 November 2020. [Online]. Available: https://xd.adobe.com/ideas/principles/web-design/flat-vs-material-skeuomorphic-examples/. [Accessed 1 April 2022]. |

| | |
|---|---|
| [67] | Geoapify, "Leaflet vs OpenLayers. What to choose?," 12 April 2019. [Online]. Available: https://www.geoapify.com/leaflet-vs-openlayers. [Accessed 2 April 2022]. |
| [68] | Vue Leaflet, "Introduction," [Online]. Available: https://vue2-leaflet.netlify.app/components/. [Accessed 2 April 2022]. |
| [69] | A. Suryana, "UNIFIED MODELING LANGUAGE," *Academia.* |
| [70] | Lucid, "What is a Flowchart," Lucid, [Online]. Available: https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial. [Accessed 15 March 2022]. |
| [71] | Vue.js, "Testing," Vue.js, [Online]. Available: https://v2.vuejs.org/v2/guide/testing.html. [Accessed 16 March 2022]. |
| [72] | G. Balinov, "Traditional Web Design vs. Growth-Driven Design," HYPE Digital Agency, [Online]. Available: https://agencyhype.com/blog/traditional-web-design-vs-growth-driven-design. |
| [73] | E. Wong, "Shneiderman's Eight Golden Rules Will Help You Design Better Interfaces," Interaction Design Foundation, 2021. [Online]. Available: https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces. |
| [74] | K. Siau, "An Analysis of Unified Modeling Language (UML) Graphical Constructs Based on BWW Ontology," *Journal of Database Management*, vol. 21, 2010. |