CHAPTER 1
INTRODUCTION

This chapter familiarizes the project that the author chose to represent as thesis. In this chapter, the background of this project is described, the author and the author team scope of doing the project is described. The objective such as aim and benefits, and vision and mission is also included in the chapter.

1.1 Background

Oil and gas were used for lamps and lubricants starting back 346 AD [1]. The oil well drilled was done by China [1]. Then moving forward to the late 18th century and early 19th century, it was an era where oil and gas industries that still dominate the world till today were established [1]. Since the mid-1950s, the demand for oil has become the world's most important source of energy. The refined oil products are used for many things such as supplying energy to the power industry, heating homes, providing fuel for vehicles and aeroplanes, and coating of pills [2]. Aside from that, oil and gas are also economically important as the oil and gas industries reduce unemployment as they 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 be used to support the maintenance of 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, petrochemical products such as synthetic rubber, and polyurethane materials [4]. These benefits lead to an increase of oil and gas consumption from around 1180 thousand barrels a day in the year 2000 to 1750 thousand barrels a day in 2018 [5]. Unfortunately these also lead to the decrease of oil and gas production that cause lesser income for the country [5]. However the oil and gas industries had provided a large contribution to state revenue of around 216.9 trillion rupiah in 2014 which boosted the country economically [5]. According to BPS Statistics Indonesia, there are 20326 Indonesians that worked in the oil and gas industries in 2019 [6]. With these numbers, the number of unemployment is lesser which represents more economic growth.

With all the importance and benefits of oil and gas and the decrease of oil and gas production mentioned earlier, now there are many software applications that help the oil and gas industries. These software applications may help the industries to analyze which locations can produce more oil and gas. Software with a visualization tool can also help oil recovery [7]. Several well-known software that provides services for data visualization are Lynx and INTViewer. However, both software is relatively pricey. Lynx, which provides visualization through seismic viewers, costs at least $400 per license per year [8]. While INTViewer which provides visualization on seismic data and geospatial data can cost up to $60,000 per year [9]. Although the software is pricey, Pertamina still manages to utilize INT's web-based HTML5Viewer with their platform in order to visualize their data quickly and easily [10]. Therefore, this inspired the author team to develop a website application that has similar features to the software mentioned before at a lower cost which became the main goal of this project.

1.2 Scope

The scope of the problem that the author faced, 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 project is the high cost of the existing software. In order to solve this problem, the solution that the author and team thought of is to develop a modernized design website application using open source frameworks and libraries. The features of the website are chosen based on the author and team decision on which are the vital features of previous software.

The software that the author and team referenced to are Schlumberger's ProSource Front Office. Therefore, the website application contains data visualization in the form of pie charts, line charts, scatter plots, stackable charts of bar and area, and bubble charts similar to Schlumberger's ProSource Front Office's features [11]. The author and team also attempt to implement Schlumberger's ProSource Front Office's feature regarding the integration of ArcGIS Arcmap to conduct geospatial analysis.

1.2.2 Scope of Work

In this project, the author is responsible for the user interfaces (UI) / user experience (UX) of one of the features of the project, the design of the particular feature along with the user interaction. The feature that the author is responsible for is a visualization of production data utilized with geographic information system (GIS). Once the author finishes developing this feature, the author is required to test the feature to ensure that it functions smoothly.

Simultaneously, the author's team are responsible to assemble different kinds of visualizations along with its diverse functionalities that are merged in the end into a website application. Each member of the team had different tasks that required the member to successfully finish their tasks one by one in order to create a complete website.

1.3 Objective

The aim and benefits of this project along with the author's team vision and mission are explained below.

1.3.1 Aim and Benefits

The major aim of developing this website application is to help the oil and gas industries locate more profitable sites of resources by visualizing the data of oil, gas, and water volume gathered throughout the time and also visualizing the reserve resources. Additionally, the 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 by predicting and revealing potential sites that contain more oil and gas.

1.3.2 Vision and Mission

The vision of the author team is to increase the use of local service in the oil and gas industries that help the industries to gain more profits and indirectly increase the national income. Aside from that, the author team desires that the website application will be known internationally.

With the vision, the author team's mission is to develop a high-quality and affordable website application using open-source frameworks and libraries that include essential features that are important in the oil and gas industries. In another way, we eliminate the features that are ineffectual in order to lower the cost for the buyer. Aside from providing an advantage for the oil and gas industries, the application is also useful to aid engineers in comprehending complicated data and gaining a better understanding of how the data can be used.

1.4 Structure

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

1.4.1 Chapter 1

Chapter 1 introduces the background, the scope, the objective including the aims, vision, and mission of this project.

1.4.2 Chapter 2

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

1.4.3 Chapter 3

Chapter 3 contains the design of the project, the project's initial phase, and the project's development flow.

1.4.4 Chapter 4

Chapter 4 explains further the flow and interaction of the project features along with the design.

1.4.5 Chapter 5

Chapter 5 describes the final analysis

CHAPTER 2

THEORETICAL FOUNDATION

This chapter explains the theories on the process to develop and design this project. Furthermore, the approach, sources, and limitation of applying frameworks and methods on developing and designing this project are described.

2.1 Software Development Life Cycle

Software development life cycle (SDLC) is a set of processes which are utilized to develop, design, and maintenance of the software projects and ensure that the result of the projects corresponds with the user requirement [10]. The processes of the SDLC, also known as phases, which are necessary for the projects development to be completed includes :

- Requirement analysis

  It is the first phase of SDLC where the development team need to grasp the detail of client's requirement, document the requirement, and determine ways to apply the requirement. [11]

- Design

  In this phase, development team normally the developers are require to be creative and come out with 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.[11]

- Implementation

  In this phase, development team implement all the client's requirements. Project member such as front end developers start developing an interactive GUI that satisfy the client's requirements and connect the GUI with the APIs. While backend developers start coding the application logics, webservices and APIs of the software that will be used by front end developer [12].

- Testing

  It is the last phase before the software is deploy and deliver to client. Testing is carried out to determine whether the software works expectedly based on the design decided in earlier phase or not. With thorough testing, high-quality software can be deliver, and avoid many bugs in production which means fewer maintenance costs [13].

- Deployment and Maintenance/Evolution

  This is the final phase of SDLC where the software will be deploy 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 hot fix (develop solution and deploy in a short time) or can be fix in next version of the software based on the severity of the issue [11].

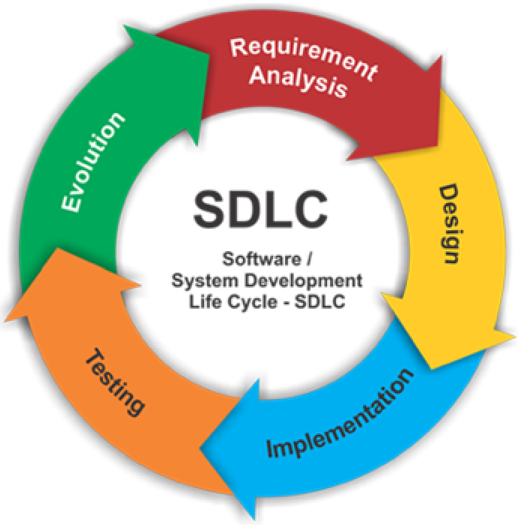


Fig 2.1 SDLC Phases [14]

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 [15]. In addition, the SDLC models are classified into traditional models or agile models. Some of the traditional models are the waterfall model, v model, iterative model, spiral model [11]. While some of the most popular agile models are extreme programming (XP), crystal methods, scrum, feature-driven development.

In order to have a better understanding of the general difference between the traditional models and agile models, the table is shown below:

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 important in order to develop the project in a short time. Documentation is less important.
Follow detailed process	Limited process involved
Develop a complete project in a single cycle which take long duration	Iterative model allow develop easily in a short time
No scrum calls and stand 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 each phase
Automation is not a usual approach	Continous automated testing is recommended and the use of it can assure better quality

Table 2.1. Difference between traditional models and agile models [16]

From the table, 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 [17]:

- Customer is God so the customer’requirements need to be prioritize through early and continuous development of the software
- Sudden changes are normal and need to be obey if it benefits the customer even though development is about to finish
- Update and deliver operating software frequently in various timespan but preferable short timespan
- Developers and involved business individuals shall updates each other’s progress daily throughout the project
- Choose the trusted individuals that can be passionate of the project and companies need to provide support and environment that the individuals need
- Face-to-face communication is best and effective way of delivering information between individuals in development team
- Published and useable software is the main goals of the project
- Agile processes facilitate endurable development where all the stakeholders of the project are able to preserve a steady pace in doing their jobs
- Agility is improve if developers paid attention to technical superiority and designers paid attention to great design
- Simplicity is the best policy which important to reduce unfinish work
- Self organizing team can gather great requirements, produce good designs, and architectures
- During daily meeting, each member of the teams reflects on how to improve effectivity and then the member try to adjust based on the things reflected.

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

Scrum is agile model that implement iterative processes that result in great progress of the project from each process [16]. This model is generally used to overcome ambiguous conditions and tight deadlines. Most of the time, there are three roles when following Scrum model which are product owner, development team, and scrum master [18]. The steps to be done when scrum model implement are [18]:

- 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 previous day's tasks and it is also time when scrum master assign new tasks

- Sprint
  It is a period of one to four weeks where development team implements (code) all the tasks listed when 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 achieved, what does not go with plan, and the things that can be improve

From the steps, the advantages of scrum are define as productive communication among project members through daily scrum, and constant feedbacks from product owners allow delivering the products accurately as expected [19]. The only disadvantage is scrum is relative hard to master for people with no experience of scrum [19].

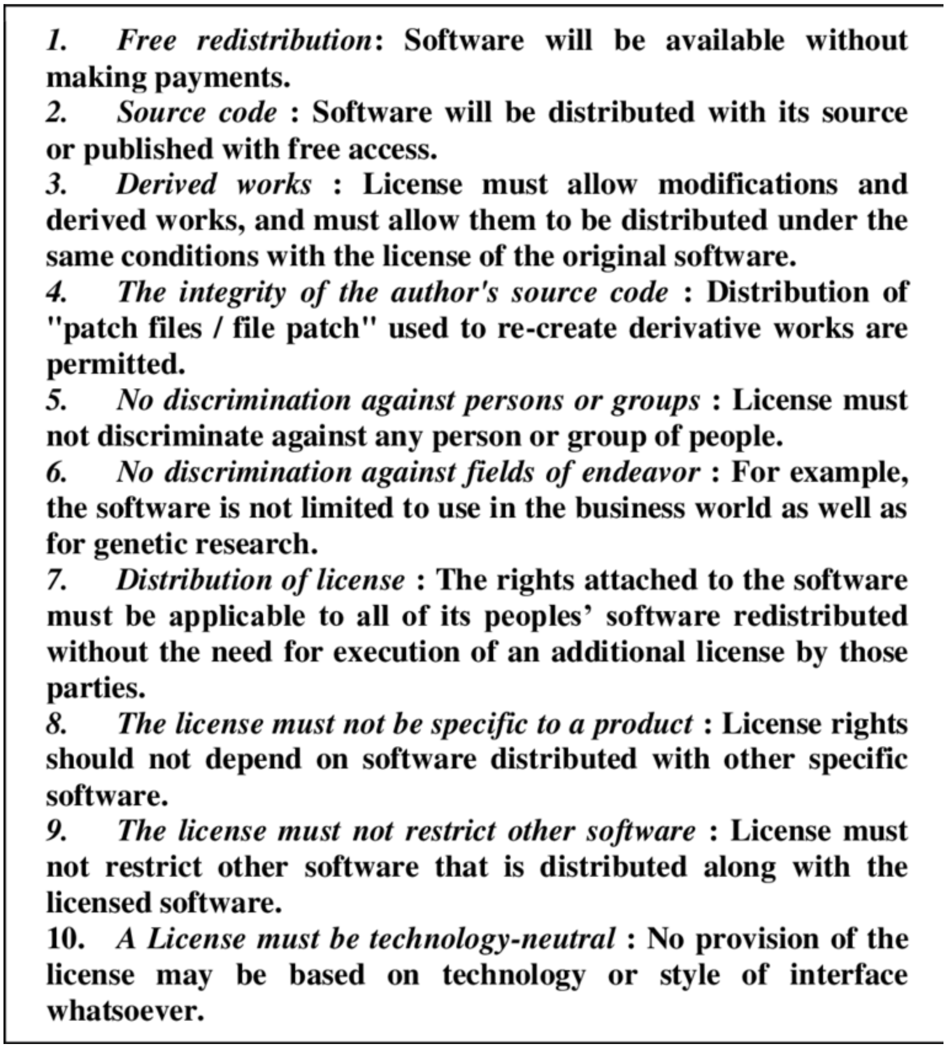
2.2 Open Source Software

Open Source Software (OSS) is software that gives liberty for anyone to review, alter, and enrich the source code [20]. OSS has ruled the world as nowadays virtual product, service, or platform is powered by it [21]. 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 that support OSS developers with fundamental tools and allow them to share their projects with the public [20]. Afterward, Microsoft released CodePlex which allow engineers and computer scientists to share their ideas and OSS projects although Microsoft was known to oppose Open Source [20]. In 2018, Microsoft also acquired GitHub the largest open-source development platform till this time [21]. Since then Microsoft has become the world's biggest open source contributor based on the number of employees actively contributing to open source projects on GitHub [22].

Whether companies provide their softwares as OSS, companies contribute to OSS, or developers use OSS, there are many benefits in doing it. The benefit for companies make their softwares 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 [23]. While the benefits for companies that contribute to OSS are attracting amazing developers to join their companies without the need to spend money on recruiters and attract developers to use the paid upgraded version of OSS as they might feel certain loyalty after using the free version for some time [24]. 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

[25]. OSS are also secure to use as developers participating in the OSS are constantly reviewing code, building improvements, and fixing issues [25]. 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 on fixing issues [24]. Developers also can increase their communication skills by communicating with other developers with different backgrounds that participate in the same OSS [24].

With all the benefits explained above, OSS is preferred over proprietary or closed source software. 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 [20]. In order to use proprietary software, users must sign a license declared the first time they open the software as an agreement that they will avoid doing anything with the software that the software's authors have forbidden [20]. 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 the way people can use, study, modify, and distribute the OSS [26]. As a requirement, the terms inside open-source licenses shall retain the criteria established by Open Source Initiative [27]. The criteria also known as Open Source Definition are shown in the figure below.



1. ***Free redistribution*****: Software will be available without making payments.**
2. ***Source code*** **: Software will be distributed with its source or published with free access.**
3. ***Derived works*** **: License must allow modifications and derived works, and must allow them to be distributed under the same conditions with the license of the original software.**
4. ***The integrity of the author's source code*** **: Distribution of "patch files / file patch" used to re-create derivative works are permitted.**
5. ***No discrimination against persons or groups*** **: License must not discriminate against any person or group of people.**
6. ***No discrimination against fields of endeavor*** **: For example, the software is not limited to use in the business world as well as for genetic research.**
7. ***Distribution of license*** **: The rights attached to the software must be applicable to all of its peoples' software redistributed without the need for execution of an additional license by those parties.**
8. ***The license must not be specific to a product*** **: License rights should not depend on software distributed with other specific software.**
9. ***The license must not restrict other software*** **: License must not restrict other software that is distributed along with the licensed software.**
10. ***A License must be technology-neutral*** **: No provision of the license may be based on technology or style of interface whatsoever.**

Fig 2.2 Open Source Definition [28]

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 with the original, these licenses are called "copyleft" [20]. 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 GPL. When developers used GPL features in their software, they need to obey the term and conditions below:

- 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 [29].

- It is forbidden to alter the license or add extra terms and conditions [29].
- Developers are bound with the reciprocal agreement that obligates them to release the source code of the software and provide others with the right to modify and redistribute [29].

Aside from that, software with a GPL license is allowed to be sold by the users whether the software is still original or already modified [29]. However, GPL gives buyers of the software the right to release the software to the public with price or not [29].

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 are licenses that acquire minimal restrictions on how the software can be used, altered, or redistributed either for open source or proprietary by the users [30]. Although they are categorized as one, they had their own terms requirements. When users use BSD licenses, they need to keep the copyright notice along with the disclaimer and the limitation on all the redistributed source codes and documentation that support the redistribution [31]. When promoting the software with BSD licenses, users also need written permission from the contributors to write their names in the copyright notice [31]. While MIT license is identical to BSD, MIT allows the user to include the name of the contributors in the copyright notice for promotion purposes [31]. In another hand, Apache is quite different as the license agreement is long to provide clear concepts on how the software can be used and description on patent rights [32].

2.3 Website application

Websites development has rapidly evolved since the first time worldwide web was proposed by Tim Berners-Lee in 1989 [33]. The first purpose of the World wide web was to allow scientists around the world to be able to share information virtually. Aside from that, there are three essential technologies that formed by Tim as the foundation of the web until nowadays which are HTML, URI, and HTTP [34]. HTML which is the acronym of HyperText Markup Language is the formatting language for documents to be displayed in the web browser [34]. URI which is the acronym of Uniform Resource Identifier and also known as URL is a unique naming system that identifies each resource on the web [34]. HTTP, the acronym of Hypertext Transfer Protocol is the application layer protocol that is responsible for the interaction of users with web resources and transferring information between devices connected with the internet [34]. 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 [33]. It was believed to be the only type of web page at that time. Fortunately, as time passes by, there are many different design types of websites (collection of web pages) that play important role in website development.

One of the common types is the static website which is familiar with the first web page that Tim made, but it is less demanded as website technology and purposes have advanced [33]. The other common types are dynamic, responsive, content management system (CMS)-built, and eCommerce [33] [35]. A dynamic website is the upgrade of the static website where the contents of the website are editable, the website have many functionalities and the user can interact with the website [33]. The only limitations of the dynamic are setting up the functionalities can be complicated and the website load slower due to the compositions and elements present in the each webpage [35]. This type of website is the active website that is available nowadays. The other type of website that followed by many websites these days is responsive. It is famous as responsive website can be access seamlessly from different type of devices including mobile phone and the website able to adjust itself to fit the browser size [36]. With this ability, responsive website provide amazing experience for the user and increase productivity. However, there are still limitations such as browser compatibility issues when opening the website from old browser version, slow load, and complications in developing the website considering perfect user experience [36]. 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 [33]. A famous example of CMS-built websites is WordPress. The last common type of website, eCommerce website which sells services and goods that can be bought directly by the customers as the website is connected with a payment gateway [33].

Aside from the various type of websites, the web design approaches which is also part of the website development has progressed significantly throughout time. With website technology and purposes have advanced as discussed before, each website needs to fulfill the needs of the user. Therefore the traditional website design approach is no longer sufficient as the ideas of this design only allow the website to be informational [33]. The developers of a company that implements the traditional design approach in creating a website also tend to go with their own assumption that they know what the web visitors demand without any data that support their assumption [33]. This means a high risk that the website does not attract customers as they did not conduct any research on customers' demand. If the website is really unattractive, the company need to redesign their website which means more money to spend and more time-consuming. After redesigning the website, the company may also face the same loss again as the redesign team will go with their own assumption again [33]. Then the cycle will repeat again indicating the possibility of more money loss and time-consuming.

With all the deficiencies explained above, the traditional design approach was overcome by a new modern approach in creating a website which is called Grown Driven Design (GDD) approach [37]. There are 2 phases in GDD approach that help a company to save their time, money, and resources on building the website [37]. The 2 phases are launchpad and continuous improvement [37]. In the first phase, it is important to dig more into the business goals [36]. Then list down all potential features and other things that support the project and list down the prioritized features that will make a significant impact on the company [36]. After that, the company shall start developing the website with prioritized features first and deploy it to the public [36]. Once the website start running, the company enters the second phase where the company shall measure the favorite existing features of the customers and determine the new features that can boost the website [37]. After choosing the new features to be implemented to the existing website, the second phase starts again as the company shall continue to measure the data again and develop new features. Through these phases, GDD approach also helps reduce the potential of failure, keep up with the trend, and help marketing and sale to make campaigns based on the data received from the website [37].

From all the design types of websites and web design approaches, companies and developers need to choose wisely on which design type and approach are suitable for the website that want to be build.

2.4. Front end development

Front end development is the part of web development that concentrates on the display of the web page. It also involves in converting the backend developers' code into a graphical interface and ensure that data displayed is reader-friendly and understandable [38]. In addition, front end developers which is responsible of front end development need to ensure that the websites work in various device type, OS, and different web browsers. Frontend developers need to put in more consideration that the website shall works on Chrome, Safari, Microsoft Edge, Internet Explorer, Mozilla Firefox, and Opera.

As front end developers, there are three technologies that they need to master. The three technologies are HTML, CSS, and Javascript [38]. HTML is main component of website that act as formatting language for documents to be displayed in the web browser as discussed before. CSS which is the acronym of Cascading Style Sheets that is a set of instructions that accountable for the website's style such as colors, animations, and layouts [38]. HTML and CSS are enough to create websites but in order to make the websites interactive, Javascript is used. However in the past, website performance is low. Therefore, google presented chrome V8 Javascript engine in 2008 that perform a lot more faster than Internet Explorer [39]. After V8 engine presented, JavaScript has great performance similar with Java [39]. Therefore, website can run fastly similar with traditional desktop software. Because of V8 JavaScript engine high capabilities during that time, many JavaScript frameworks are formed afterward and this evolve the internet development [39]. Some JavaScript front-end frameworks that are available until today are React Js, and Vue Js. React Js work by distributes webpage content as various components in the Document Object Model (DOM) and then the browser generate the component using JavaScript [39]. The advantage of React Js over traditional HTML website is the way React Js only re-render the part of the website that is updated rather than re-render all the part of the website which is time-consuming [39]. React Js able to do so by comparing data bind with virtual DOM and displayed DOM, the data contain the update of the part that will be re-render [39]. In addition, Vue.Js also use DOM binding to update content appearance of the website. The only difference between React Js and Vue Js is that React Js is dependant on functional programming principle where libraries are use to manage state and communication between components [39]. While Vue Js provide built-in features and supporting libraries that allow development experience more seamless [39]. Based on differences, developers can prefer one over the others.

2.4.1 Human-computer interaction (HCI)

Human-computer interaction (HCI) is a study about the innovation of computer technology and interaction between humans and computers [40]. HCI is extensive and related to certain fields such as user experience design and user interface design [40]. HCI can be described as the ancestor of user experience design in numerous manners [40]. 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 [41]:

- 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 the making of interface design [41]. When examining the physical environments, it is important to consider designing a system that protects the user's wellbeing such minimize device radiation should be the prioritize of interface designer [41]. Interface designers also need to design the interface while considering the condition of working space such as enough room, efficiency, lighting, low distraction, and low pollution [41].

As for the social environment, it needs to be considered in designing interfaces because different social environments may require different computing frameworks [41]. For example, personal computing is mostly done in-house or office. While mobile computing is usually done outdoors. Other 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 [41].

In addition, the Cognitive Computing Environment examines the cognitive elements of people's interactions with computers [41]. These are cognitive science-based topics that encompass fundamental psychological concepts like learning and problem-solving in terms of abilities, methods, knowledge, and styles [41]. 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 [41]. For example, the design of the interface needs to be clear, straightforward, and zero error for a government application. While the design of the interface needs to be more soothing and enjoyable for music applications.

With all the elements of interaction to be considered when designing an interface, HCI models that cover all the elements are developed and utilized until now. Some of the HCI models are personal computing, mobile computing, large-scale computing, networked computing, collaborative environments, augmented reality, and virtual reality [41]. These models can be analyzed and find out the elements used for user interaction by using the who, what, where, why, and how (5W+H) procedure [41]. The procedure is broken down into three components [41]:

- The What/How: This step of the procedure is used to learn the physical and virtual interface components used by the models. For example, I/O devices, windows, etc.
- 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 step of the procedure distinguishes the types of tasks can the models 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.

2.4.2 8 Golden Rules

The 8 Golden Rules are well-known user interface guidelines [42]. It was created by Ben Shneiderman who is an American computer scientist and professor at the University of Maryland Human-Computer Interaction Lab [42]. He first revealed his 8 golden rules in his popular book, "Designing the User Interface: Strategies for Effective Human-Computer Interaction". His 8 Golden Rules of interface design are as explained below [42]:

- Strike for consistency
  When there are alike situations or actions, it is best to design the elements (icons and colors) and the function (user flows) of the actions similarly. With identical designs for each action, users can easily familiarize themselves with how the product works. As users familiarize themselves with the product, the goals of the product are also achieved.

- Enable frequent users to use shortcuts
  Once the product is highly demanded, it is great for users of the product to quickly complete tasks. With the use of shortcuts, users can effortlessly and fastly navigate and utilize the user interface.

- Offer informative feedback
  As users have the right to know what they need to do and where they are at when using software, it is best that there are feedbacks for each action that the users did. A great example of applying this rule would be to inform users where they are in the process when filling out a multi-page questionnaire. While the example of bad feedback is an error message showing an error code that confuses users that are non-programmers.

- Design dialogue to yield closure
  Provide users with the result of their action rather than keeping the users guessing. For instance, it is the right thing to provide a “successful” message along with a digital receipt when users did an online purchase.

- Offer simple error handling
  It is great to design the systems to be as faultless as possible, but when inevitable errors happen, it is best to provide users with straightforward instructions on how to solve the problems as quickly as possible, For example, show error message near text fields that users forgot to fill in a form.

- Permit easy reversal of actions

  A product should be designed that users are able to reverse their actions in easy ways. For example, a back button in a system.

- Support internal locus of control

  Give users the control to do any actions with the system and convince the users that they can trust the system to do the things that they desire.

- Reduce short-term memory load

  Human concentration is limited to remembering five items at one time. Therefore, the project interfaces are better to be simple, deliver information as straightforward as possible, and provide hints that are easily recognized by users. For example, during exams, many people prefer multiple choice questions rather than short answer questions as multiple choice only requires them to recognize the answer. While short answers require people to recall from their memory on the answer to the questions.

Once all the rules are integrated with the design of the products, it is believed that the products will be successful. The best example of successful products that implement these rules is Apple's products [42]. One of the consistency of the products is represented by the Mac OS Menu Bar which has the same graphic elements design from the 1980s till now. Mac also provides various keyboard shortcuts which are aligned with the second rule. Then Mac desktop implemented the third rule by highlighting a file when the user clicks a file. After that, Mac follows the fourth rule when users install software from the browser where an informative dialog will appear and shows the progress of the installation. Then users can also go back to the previous step of installation when they input wrong details which follow the sixth rule. When the installation is finished and turns out that the software is not trusted by Mac, a human-readable error message will show which follows the fifth rule. However, users have the control to set their permission that they can still install the software even if it is not trusted by Mac which follows the seventh rule. For the last rule, apple's product that implements this rule is the iPhone. As humans can only remember

5 things in short-term memory, there are only 4 app icons displayed at the bottom of the iPhone screen.

2.4.3 Vue Js, Materio, Material Design, Vuetify, and Material Design Icons

Vue.js is a famous JavaScript open-source front-end framework invented by Evan You in 2014 [39]. The initial purpose of Vue being developed is to provide reactive data binding and UI components using an uncomplicated Application Programming Interface(API) [39]. Although a single-page application has limited functions and is hard to implement in commercial use, Vue is able to operate complex single-page applications including state management, routing, and build tooling with the support of third-party libraries and packages [39].

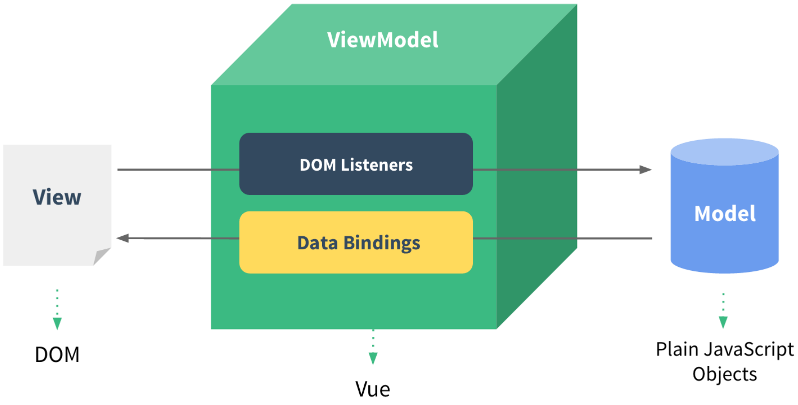


Fig 2.4.3.1 Vue Js data-driven concept [43]

Figure above demonstrates that Vue.js retains three units to process data-driven includes View, View Model, and Model. The view unit is the displayed DOM that contain the website content [39]. View Model unit contains Data Bindings and DOM listeners. It becomes a middleware that is responsible for the transmission between View and Model. The DOM listeners in View Model unit are use by Vue to observe and update the data in Model unit when users begin the data processing in View unit [39]. After the data in Model is updated, the DOM binding is use to update the display of the website content [39]. Therefore it can be conclude that the one-way binding with DOM listeners in Vue has same ability as two-way binding [39].

Materio Admin is an open-source, user-friendly, and customizable admin template built with Vue Js and Vuetify [44]. It is created by ThemeSelection and came out with two versions [44]. Both versions of Materio are utilized with Vuex, Vue Router, Webpack, and Material design icons [44]. The features of free version includes [44]:

- 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 pro version includes [44]:

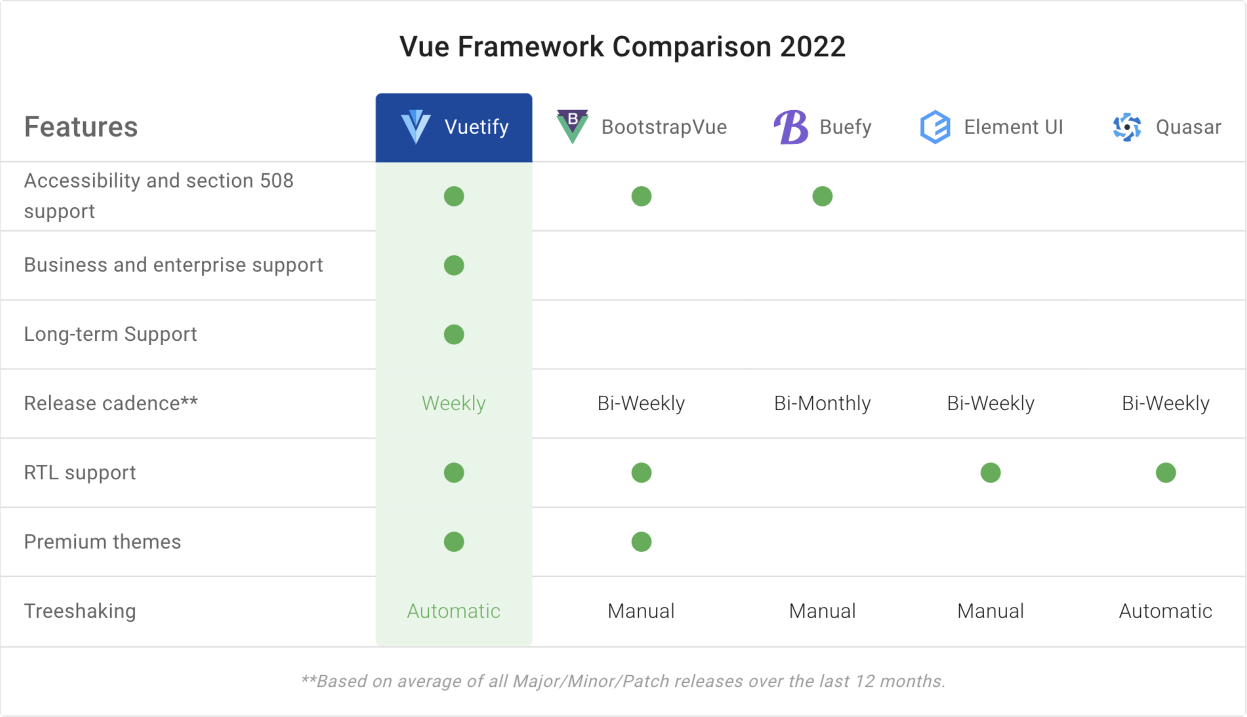
- A collapsable vertical menu and horizontal menu
- 3 dark and light skin layouts: Default, Bordered & Semi-dark
- 3 choice of dashboards
- 5 API ready application
- Advance form elements, validation, and form wizard
- Basic cards, Advanced cards, and Statistics cards
- Advanced tables
- Advanced charts including 2 chart libraries
- 2 choices of authentication pages and various operable pages like pricing, FAQ, knowledge base, etc

- Multiple Navbar & Menu Options
- 3D characters and Illustrations
- Detailed documentation
- Starter-kit
- Access Control including the function CRUD
- Quick Search enable fast navigate 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 the free version uses the MIT license which is the simplest license and it is declared as free forever [44]. The free version is also available to be run in the latest browsers such as Chrome, Safari, Firefox, and Opera [44].

The Vuetify used to build the Materio and Material design icons that are utilized in Materio are part of material design. Material design was developed by Google with the purpose to create a united style for all webs and mobile apps [45]. In definition, material design is a flexible design of guidelines, components, and tools to support user interface design [45]. There are many different material design frameworks and libraries that offer components and styles that are directly usable [45]. One of the material design frameworks is Vuetify. While one of the material design libraries is Material design icons.

Vuetify is a UI framework. The purpose of Vuetify is to provide tools for developers that help them in creating attractive user experiences [46]. Unlike other frameworks, Vuetify is designed from the very beginning to be developer-friendly [46]. It is developed based on Material Design specification with every component thoroughly prepared [46]. 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 [46]. Additionally, Vuetify also provides 18 months of long-term support for each major release [46]. When these features are compared with other Vue UI frameworks as shown in the figure below, Vuetify has the most complete features so it is recommended to use Vuetify.



Vue Framework Comparison 2022

Features	Vuetify	BootstrapVue	Buefy	Element UI	Quasar
Accessibility and section 508 support	●	●	●		
Business and enterprise support	●				
Long-term Support	●				
Release cadence**	Weekly	Bi-Weekly	Bi-Monthly	Bi-Weekly	Bi-Weekly
RTL support	●	●		●	●
Premium themes	●	●			
Treeshaking	Automatic	Manual	Manual	Manual	Automatic

***Based on average of all Major/Minor/Patch releases over the last 12 months.*

Fig 2.4.3.2 Vue Framework Comparison 2022 [46]

Material Design Icons is a library containing icon collections that can be downloaded in any format, color, and size according to the needs of the designers or developers' projects and it can be utilized in various frameworks including Vue and React [47]. It is maintained by Austin Andrews and the general license of Material Design Icons is Pictogrammers Free License [47]. However, some icons are redistributed with Apache 2.0 license while others are redistributed with their respective license [47]. All fonts are also redistributed with Apache 2.0 license and all non-font and non-icon files in the Material Design Icons GitHub applies the MIT license [47]. Material Design Icons can be downloaded using npm or from the website [47].

2.5. GIS

- 2.5.1 Map Provider
- 2.5.2 GIS Framework

2.6 Testing

- 2.6.1Unit
- 2.6.2 integration

REFERENCES

[1]https://www.offshore-technology.com/comment/history-oil-gas/#:~:text=The%20modern%20history%20of%20the,thicker%20oil%20suitable%20for%20lubrication.

[2]https://www.ukogplc.com/page.php?pID=74#:~:text=Oil%3A%20lifeblood%20of%20the%20industrialised,people%20all%20over%20the%20world.

[3]https://www.energy.gov/sites/prod/files/2020/10/f80/Economic%20Impact%20of%20Oil%20and%20Gas.pdf

[4]
https://finance.detik.com/energi/d-5367484/6-pemanfaatan-minyak-bumi-bagi-kehidupan-apa-saja

[5]https://www.pwc.com/id/en/energy-utilities-mining/assets/oil-and-gas/oil-gas-guide-2019.pdf

[6]
https://www.bps.go.id/publication/download.html?nrbvfeve=Nzg5YzQ1MDk0ODA3OGE5YTg1YmUxNzNk&xzmn=aHR0cHM6Ly93d3cuYnBzLmdvLmlkL3B1YmxpY2F0aW9uLzIwMjAvMTIvMjEvNzg5YzQ1MDk0ODA3OGE5YTg1YmUxNzNkL3N0YXRpc3Rpay1wZXJ0YW1iYW5nYW4tbWlueWFrLWRhbi1nYXMtYnVtaS0yMDE0LS0tMjAxOS5odG1s&twoadfnoarfeauf=MjAyMi0wMi0yNiAyMjowOToxOA%3D%3D

[7]
[Fan2018] Fan, Y., Gao, K., Chen, J., Li, W., & Zhang, Y. (2018). Low-cost PMMA-based microuidics for the visualization of enhanced oil recovery. Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles, 73, 26.

[8]http://www.lynxinfo.co.uk/download-pricing.html

[9]
[INT2021] INTViewer. Geoscience Analysis and QC, Simplied. (n.d.). Retrieved September 15, 2021, from https://www.int.com/products/intviewer/.

[10] Lehman, Tobin J., and Akhilesh Sharma. "Software development as a service: agile experiences."
SRII Global Conference (SRII), 2011 Annual. IEEE, 2011.

[11] http://www.ijecs.in/index.php/ijecs/article/view/2830/2616

[12] https://www.careerexplorer.com/careers/back-end-developer/

[13]
https://www.researchgate.net/publication/319716548_SOFTWARE_DEVELOPMENT_LIFE_CYCLE_SDLC_ANALYTICAL_COMPARISON_AND_SURVEY_ON_TRADITIONAL_AND_AGILE_METHODOLOGY

[14] https://addontraining.com/wp-content/uploads/2016/10/sdlc.png

[15] https://www.ijcait.com/IJCAIT/13/1334.pdf

[16]
https://www.researchgate.net/publication/319716548

[17] https://agilemanifesto.org/iso/en/principles.html

[18]https://www.researchgate.net/publication/230634624_Scrum_agile_product_development_method_-literature_review_analysis_and_classification

[19]http://www.jsoftware.us/vol14/382-MS07.pdf
[21] https://maximilianmichels.com/2021/history-of-open-source/
[20] https://opensource.com/resources/what-open-source
[22]
https://www.techrepublic.com/article/microsoft-may-be-the-worlds-largest-open-source-contributor-but-developers-dont-yet-care/

[23] https://rdcu.be/cIbxH

[24]
https://www.freecodecamp.org/news/a-business-case-for-open-source/#:~:text=Contributing%20to%20Open%20Source%20Attracts,the%20upgraded%20or%20scaled%20versions.

[25]https://www.zivtech.com/blog/benefits-open-source-software

[26] https://opensource.com/law/10/10/license-compliance-not-problem-open-source-users

[27] https://www.jhuapl.edu/Content/techdigest/pdf/V32-N04/32-04-Hahn.pdf

[28]
https://www.researchgate.net/figure/The-Open-Source-Definition-Source-Wei-2011_fig2_338173179

[29]https://www.whitesourcesoftware.com/resources/blog/top-10-gpl-license-questions-answered

[30] https://snyk.io/learn/what-is-mit-license/

[31]https://snyk.io/learn/what-is-bsd-license/

[32]https://www.whitesourcesoftware.com/resources/blog/top-10-apache-license-questions-answered/

[33]https://terrostar.com/the-evolution-of-website-development/

[34]https://webfoundation.org/about/vision/history-of-the-web/

[35] https://www.webfx.com/blog/web-design/types-of-web-design/

[36]
https://www.researchgate.net/publication/324131848_The_role_of_responsive_design_in_web_development

[37] https://agencyhype.com/blog/traditional-web-design-vs-growth-driven-design

[38] https://www.techslang.com/definition/what-is-frontend-development/

[39] https://www.researchgate.net/publication/332456776

[40]https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro

[41]
https://www.researchgate.net/publication/280689716_Principles_of_Human_Computer_Interaction_Design

[42]
https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces

[43] https://v1.vuejs.org/guide/overview.html
[44] https://themeselection.com/products/materio-free-vuetify-vuejs-admin-template/

[45] https://material.io/
[46] https://vuetifyjs.com/en/introduction/why-vuetify/

[47]https://materialdesignicons.com/