Higher Nationals - Assignment 1

Student Name/ID	Nguyen Dinh Tam/GCD210186		
Unit Title	Unit 30: Application Development		
Assignment Number	1	**Assessor**	Pham Thanh Son
Submission Date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	

Assessor Feedback:

*Please note that constructive and useful feedback should allow students to understand:

a) Strengths of performance

b) Limitations of performance

c) Any improvements needed in future assessments

Feedback should be against the learning outcomes and assessment criteria to help students understand how these inform the process of judging the overall grade.

Feedback should give full guidance to the students on how they have met the learning outcomes and assessment criteria.

Grade:	**Assessor Signature:**	**Date:**

Resubmission Feedback:

*Please note resubmission feedback is focused only on the resubmitted work

Grade:	**Assessor Signature:**	**Date:**

Internal Verifier's Comments:

Signature & Date:

* Please note that grade decisions are provisional. They are only confirmed once internal and external moderation has taken place and grades decisions have been agreed at the assessment

Grading grid

P1	P2	P3	M1	M2	D1

TEAM MEMBERS

Full Name	Student Code
Nguyen Dinh Tam (Leader)	GCD210186
Hoang Viet	GCD210635
Dinh Tran Anh Khoa	GBD210500

FPT Education
TỔ CHỨC GIÁO DỤC FPT

UNIVERSITY OF GREENWICH
Alliance with FPT Education

Pearson | BTEC

APPLICATION DEVELOPMENT REPORT

ASSIGNMENT 1

Table Contents

List of Figures

A. Software requirements specification:

I. Introduction:

- In today's rapidly evolving job market, the process of job posting and hiring has become increasingly complex and time-consuming. Employers face challenges in finding the right candidates among a pool of applicants, while job seekers struggle to navigate through numerous job postings and effectively showcase their skills and qualifications. Recognizing these pain points, FPTJobMatch emerges as an innovative web-based platform designed to revolutionize the recruitment experience.

- This report provides a comprehensive analysis of FPTJobMatch, exploring its features and the underlying solution it offers to enhance the efficiency of the recruitment experience. By harnessing cutting-edge technology, FPTJobMatch streamlines the recruitment journey for employers and job seekers alike. The report delves into the initial requirements and the software design and development methodologies employed to create FPTJobMatch. Key transferable skills and competencies developed during its creation are highlighted, including producing clear and comprehensive Software Design Documents (SDD), utilizing Unified Modeling Language (UML) for diagram creation, prototyping to validate proposed solutions, and applying Agile/Scrum methodologies for iterative development.

- The report also emphasizes proficiency in version control systems, relevant programming languages, database design, and testing strategies. The utilization of development tools and Integrated Development Environments (IDEs) is discussed, along with the application of project management techniques for effective planning, monitoring, and control. With a user-friendly interface, advanced features, and a focus on efficiency, FPTJobMatch aims to revolutionize job postings, streamline application processing, and enhance candidate selection. Subsequent sections provide a detailed overview of the platform's comprehensive features and functionalities, demonstrating its potential to reshape the recruitment process.

1. Document purpose:

- The purpose of this Software Requirements Specification (SRS) document is to provide a clear and comprehensive outline of the software requirements for the FPTJobMatch platform. It identifies the scope of the product, including the covered subsystems, and serves as a reference guide for the design, development, and implementation of the system.

- The SRS document describes the functional and non-functional requirements of the platform, ensuring a common understanding among stakeholders, development teams, and other involved parties. It establishes the desired behavior and functionality of FPTJobMatch, facilitating effective communication, collaboration, and decision-making throughout the software development process. Additionally, the SRS document may serve as a future reference for enhancements and maintenance of the platform, ensuring its continued success as a robust and efficient job matching and recruitment solution.

2. Product scope:

- The scope of the FPTJobMatch platform covers a wide range of functionalities and features:

+ For employers, the FPTJobMatch platform offers numerous benefits, objectives, and goals. Employers can benefit from a streamlined and automated recruitment process, saving time and resources. The platform enables them to create and post job listings with all the necessary details, ensuring accurate and comprehensive job descriptions. By providing tools to manage incoming job applications, employers can efficiently review and respond to candidates, leading to a more efficient selection process. The ability to view detailed profiles of job seekers and utilize sorting and filtering options allows employers to identify the most suitable candidates for their job openings. The overall objective is to improve the quality of hires, reduce time-to-hire, and enhance the efficiency of the recruitment process.

+ For job seekers, the FPTJobMatch platform aims to provide a user-friendly and efficient job search experience. Job seekers can benefit from a wide range of job opportunities available on the platform, and they can search for relevant listings based on specific criteria such as job title, location, and industry. The ability to apply to job listings by submitting a self-introduction or attaching resumes and cover letters simplifies the application process. Confirmation notifications upon successful submission provide reassurance to job seekers. The objective is to enhance job seekers' visibility to potential employers, increase their chances of finding suitable employment, and streamline the application process.

+ For admins, the FPTJobMatch platform provides administrative features to manage user accounts effectively. Admins can create, suspend, and delete user accounts, ensuring the proper functioning of the platform. The availability of password reset functionality helps employers and job seekers regain access to their accounts. Admins also play a crucial role in reviewing and approving/rejecting new job categories requested by employers, ensuring the platform maintains relevant and appropriate job listings. The objective is to maintain the integrity and security of user accounts, facilitate smooth platform operation, and ensure the quality and relevance of job categories.

3. Intended audience and document overview:

- The intended audience of this Software Requirements Specification (SRS) document includes various stakeholders involved in the development and implementation of the FPTJobMatch platform. These stakeholders may include developers, project managers, marketing staff, users, testers, and documentation writers. Each reader type will find specific sections of the document relevant to their role and responsibilities.

- The SRS document is organized to provide a comprehensive understanding of the software requirements for FPTJobMatch. It begins with an overview section that describes the purpose, scope, and intended audience of the document. Following the overview, the document is divided into sections that cover different aspects of the platform, including user authentication and roles, employer features, job seeker features, admin features, and more. Each section outlines the specific requirements and functionalities related to the corresponding role or feature.

- To ensure effective comprehension, it is recommended to read the document in the following sequence:

+ Begin with the overview sections, including the purpose of the document and the product scope. This provides a high-level understanding of the goals, objectives, and benefits of the FPTJobMatch platform.

+ Developers and project managers should focus on the sections related to user authentication and roles, as well as the specific features and functionalities for employers, job seekers, and admins. These sections provide detailed requirements and specifications that will guide the development and implementation process.

+ Marketing staff and documentation writers can refer to the sections covering the benefits, objectives, and goals of the platform. These sections provide valuable insights for marketing materials and documentation creation, highlighting the advantages and value proposition of FPTJobMatch.

+ Users and testers should pay particular attention to the sections that outline the features and functionalities relevant to their roles. These sections provide a detailed understanding of how users will interact with the platform and the expected behavior of different system components.

II. Overall description:

1. Product overview:

- FPTJobMatch is a web-based software built on .NET platform. FPTJobMatch operates within the context of the job posting and hiring process. It serves as an intermediary between employers, job seekers, and the admin, providing a streamlined and user-friendly interface for efficient recruitment. FPTJobMatch enables employers to create and post job listings, manage applications, review candidate profiles, and request new job categories. Job seekers can search for job listings, apply with their resumes or self-introductions, and maintain their profiles. The admin plays a crucial role in managing user accounts, approving/rejecting new job categories, and ensuring the smooth operation of FPTJobMatch. The product aims to enhance the overall recruitment experience, saving time and effort for all parties involved.

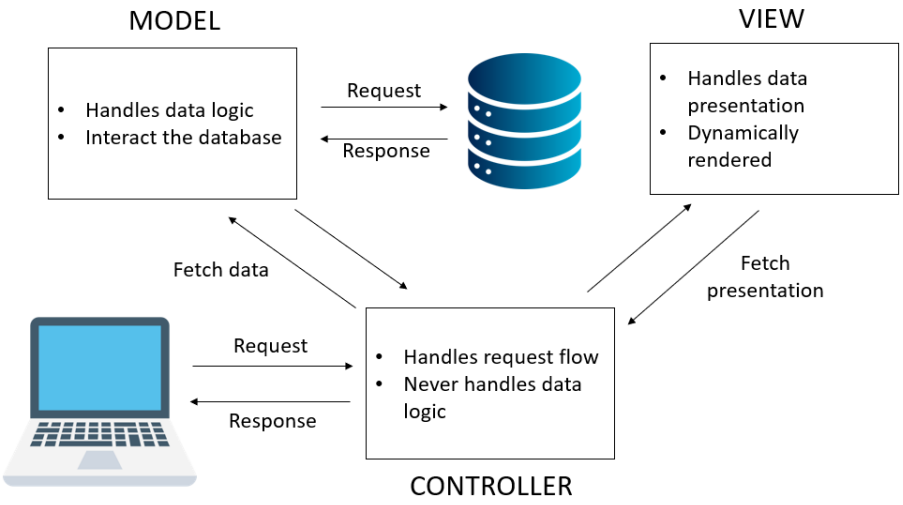

Figure 1: Overview

2. Product functionality:

- The major functions of the FPTJobMatch platform include:

+ **User Authentication**: The system provides a secure login mechanism for both employers and job seekers, ensuring that only authorized users can access their respective accounts.

+ **Role-based Access**: Users are categorized into two roles, Employers and Job Seekers, each with specific privileges and functionalities tailored to their needs.

+ **Job Posting**: Employers have the ability to create and post job listings, including details such as job title, description, qualifications, and application deadline.

+ **Job Search**: Job seekers can search for job listings based on various criteria such as job title, location, and industry, enabling them to find relevant opportunities.

+ **Application Submission**: Job seekers can apply to job listings by submitting a short self-introduction or attaching their resumes and cover letters, allowing employers to review their qualifications.

+ **Application Management**: Employers can view and manage incoming job applications, including tools for shortlisting, reviewing, and responding to candidates.

+ **Candidate Profiles**: Employers have access to detailed profiles of job seekers, enabling them to make informed decisions during the selection process.

+ **Sorting and Filtering**: Employers can utilize tools for sorting and filtering candidates based on relevant criteria, making it easier to identify the most suitable candidates for their job openings.

+ **User Account Management**: Admins have the ability to manage user accounts, including account creation, suspension, and deletion, ensuring the proper functioning of the platform.

+ **Job Category Management**: Admins review and approve/reject new job categories requested by employers, ensuring the quality and relevance of job listings.

III. Specific requirements:

1. Functional requirements:

1.1. Employer:

- As an employer, I want to be able to create a new account and log in securely, so that I can access personalized features and manage my job listings effectively.
- As an employer, I want to be able to create and post job listings with details such as job title, description, qualifications, and application deadline, so that I can attract potential candidates and provide comprehensive information about the job openings.
- As an employer, I want to be able to manage incoming job applications, including the ability to view, sort, filter, and respond to candidates, so that I can efficiently review and select the most suitable candidates for my job openings.
- As an employer, I want to be able to access detailed profiles of job seekers, including their resumes and contact information, so that I can make informed decisions during the selection process and easily contact potential candidates.
- As an employer, I want to receive notifications about new job applications and important updates, so that I can stay updated on the progress of my job listings and promptly respond to candidate applications.

- As an employer, I want to be able to edit and update job listings and their associated details, so that I can make necessary changes to reflect any modifications or updates to the job requirements or application process.
- As an employer, I want to be able to delete job listings that are no longer relevant or available, so that I can maintain an updated and relevant list of job openings.
- As an employer, I want to be able to search and filter job seekers' profiles based on specific criteria, so that I can easily find and identify the most suitable candidates for my job openings.

1.2. Job seeker:

- As a job seeker, I want to be able to create a new account and log in securely, so that I can access personalized features and manage my job search effectively.
- As a job seeker, I want to be able to search for job listings based on criteria such as job title, location, and industry, so that I can find relevant job opportunities.
- As a job seeker, I want to be able to view detailed job descriptions and requirements for each listing, so that I can assess the suitability of the job and make informed decisions about applying.
- As a job seeker, I want to be able to apply to job listings by submitting a self-introduction or attaching my resume and cover letter, so that employers can review my qualifications and consider me as a potential candidate.
- As a job seeker, I want to receive confirmation notifications upon successful application submission, so that I know that my application has been successfully received.
- As a job seeker, I want to be able to edit and update my profile, including personal and professional information, resumes, and contact details, so that I can keep my information up to date and showcase my skills and experience accurately.
- As a job seeker, I want to receive notifications about job opportunities matching my profile and application status updates, so that I can stay informed about relevant job openings and the progress of my applications.

1.3. Admin:

- As an admin, I want to be able to create, suspend, and delete user accounts, so that I can manage the user base and ensure the integrity of the platform.
- As an admin, I want to review and approve/reject new job categories requested by employers, so that I can maintain the quality and relevance of job listings.

- As an admin, I want to manage system settings, such as email notifications and platform configurations, so that I can customize the platform according to the needs and preferences of the users.
- As an admin, I want to access logs and reports for user activity and system performance, so that I can monitor the platform's usage and identify any issues or areas for improvement.
- As an admin, I want to reset passwords for employers and job seekers upon request, so that I can assist users in regaining access to their accounts.
- As an admin, I want to provide technical support and assistance to users, resolving any platform-related issues they may encounter, so that I can ensure a smooth user experience.
- As an admin, I want to monitor and ensure the security and integrity of the platform, implementing necessary measures to protect user data and prevent unauthorized access.
- As an admin, I want to keep the platform up-to-date with necessary software patches and updates, so that I can ensure the platform's stability, security, and compatibility with the latest technologies.

2. Use case diagram:

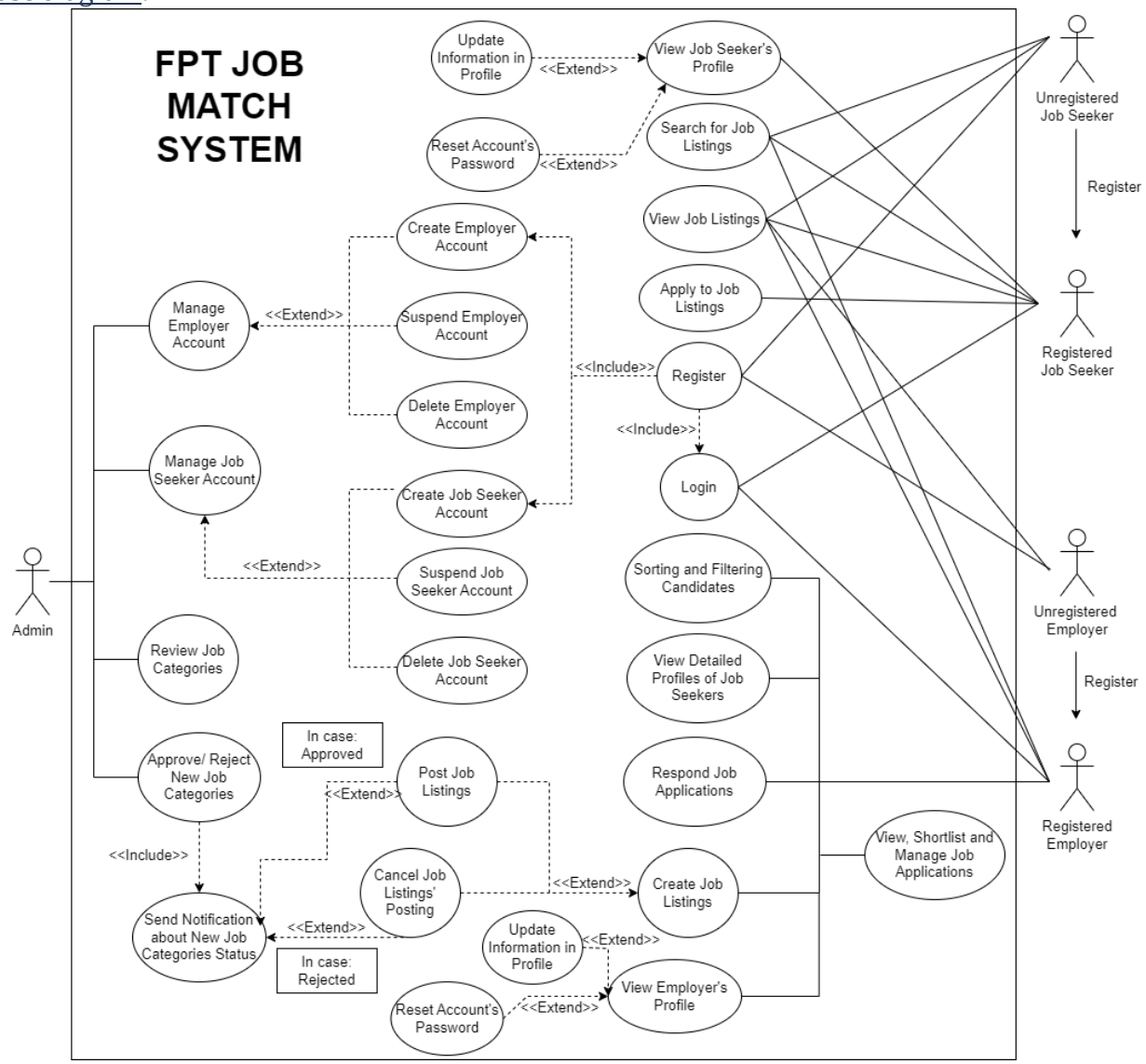

Figure 2: Use Case Diagram

3. Site map:

3.1. Home page:

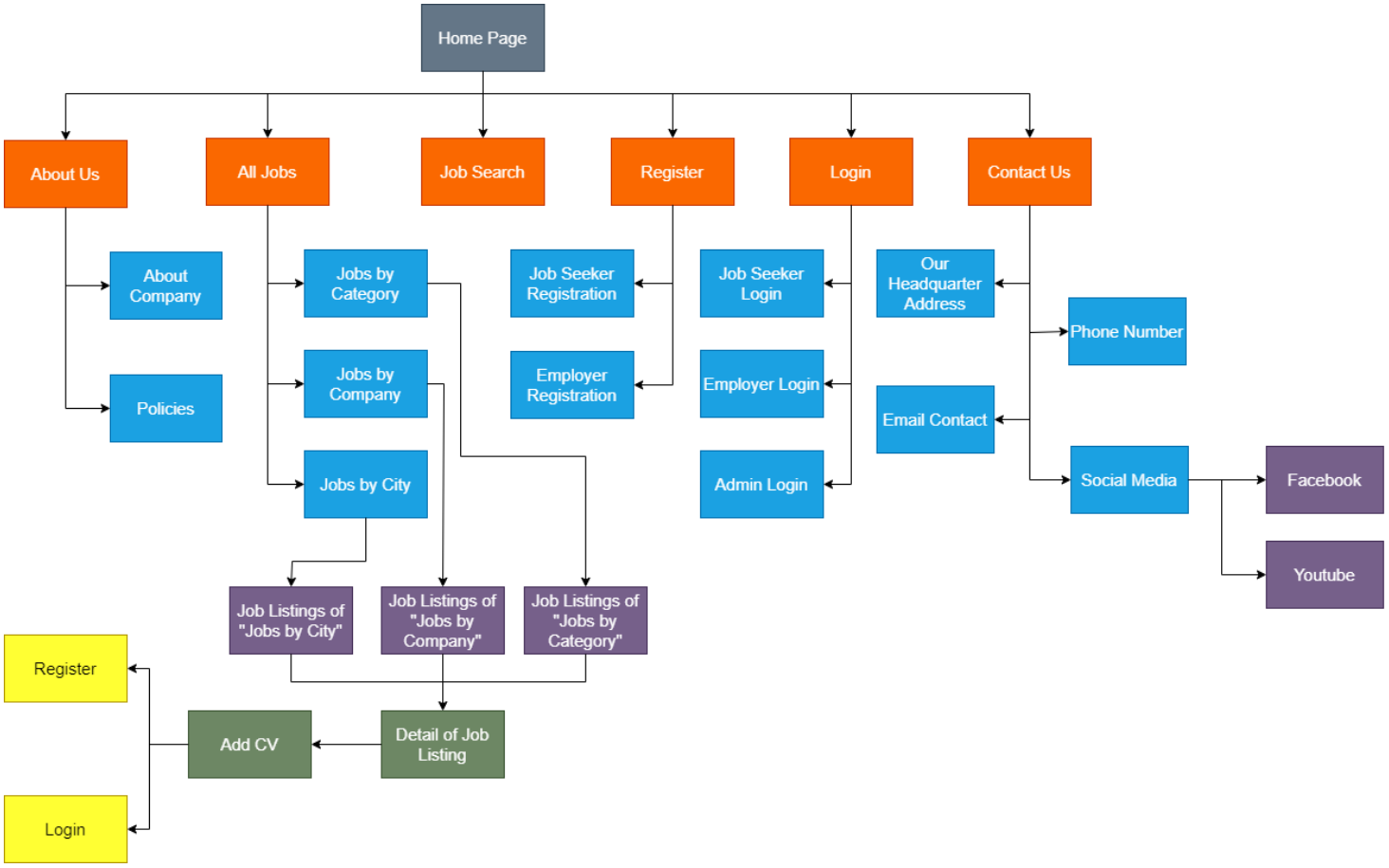

Figure 3: Home Page

3.2. Employer page:

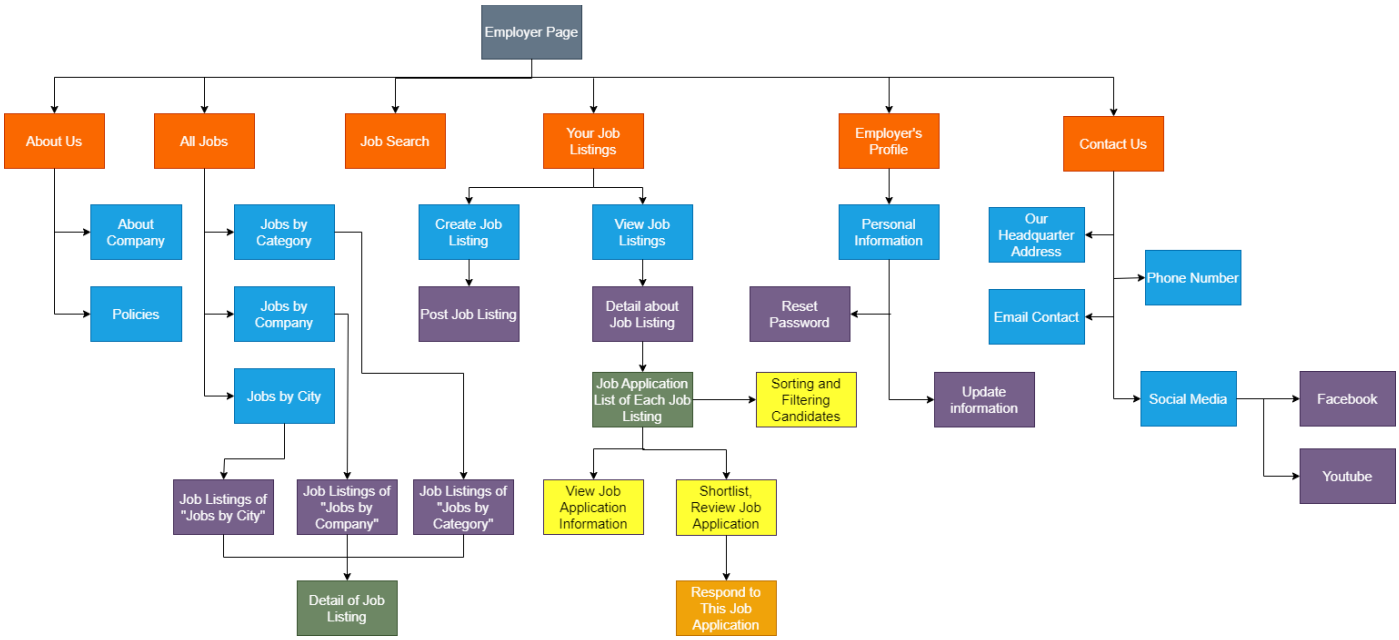

Figure 4: Employer Page

3.3. Job Seeker page:

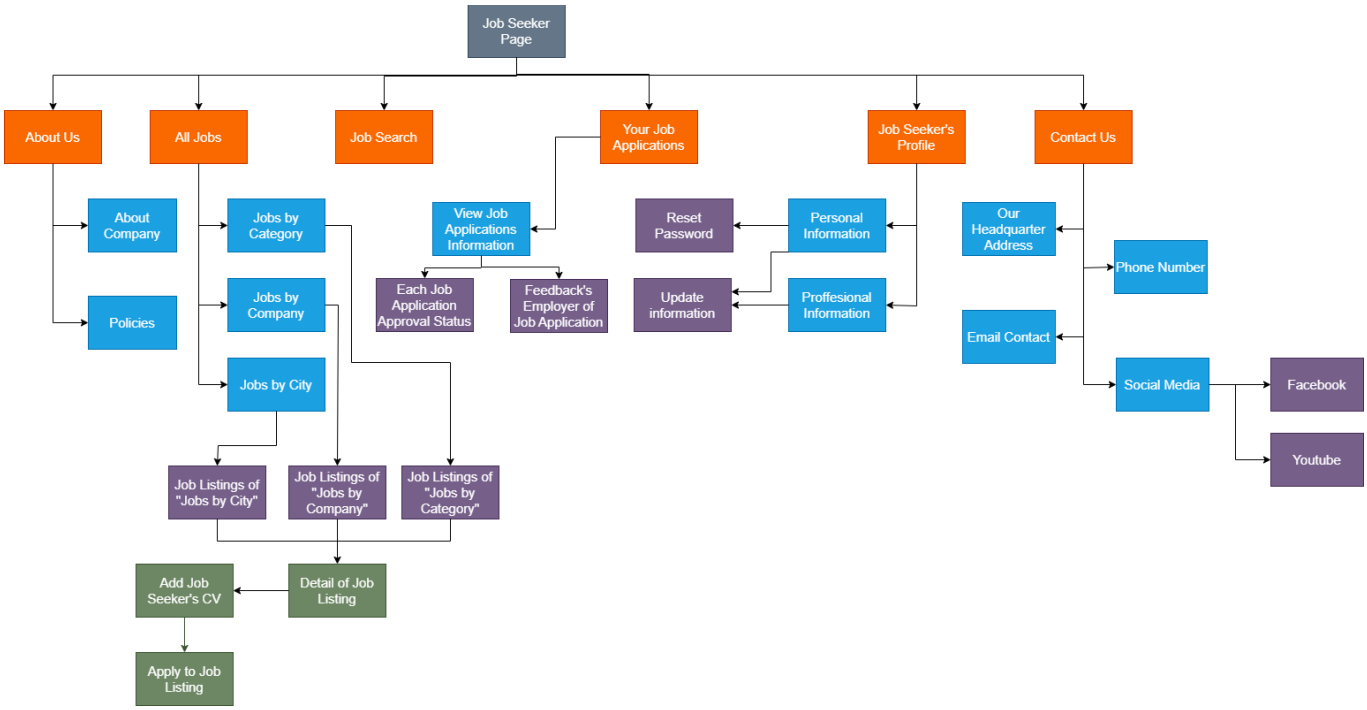

Figure 5: Job Seeker Page

3.4. Admin page:

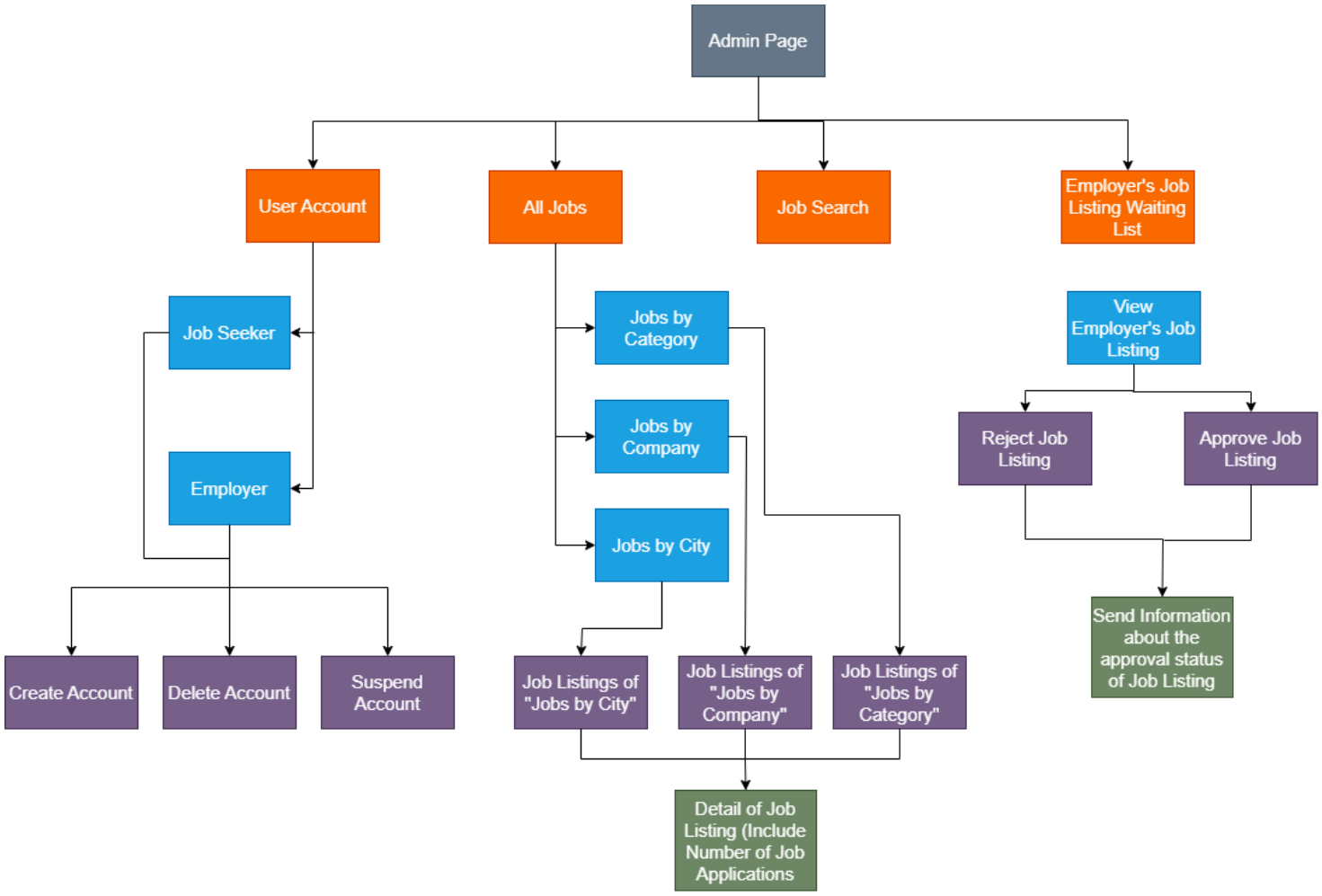

Figure 6: Admin Page

4. Risk assessment:

Type of Risk	Risk No.	Description	Impact	Frequence	Actions
Technical Risk	1	Integration challenges between the FPTJobMatch platform and external systems or APIs may arise, leading to delays or disruptions in functionality.	High	Medium	- Conduct thorough testing and verification of integration points during the development phase. - Establish clear communication channels with external system providers to address any integration issues promptly. - Implement fallback mechanisms or alternative solutions to mitigate the impact of integration failures.
	2	Security vulnerabilities in the platform may expose user data to unauthorized access or breaches.	High	Low	- Implement robust security measures, including encryption, access controls, and regular security audits. - Stay updated with the latest security patches and best practices to mitigate known vulnerabilities. - Conduct regular penetration testing and vulnerability assessments to identify and address potential security weaknesses.
Operational Risk	3	Server or network failures may result in temporary unavailability of the FPTJobMatch platform.	High	Medium	- Implement redundant server architecture and backup systems to ensure high availability. Regularly monitor server and network performance to identify and address potential issues proactively. - Establish disaster recovery plans and backup procedures to minimize downtime in case of system failures.

	4	Insufficient scalability of the platform may lead to performance degradation during peak usage periods.	Medium	Medium	- Perform load testing and capacity planning to identify potential scalability bottlenecks. Implement scalable infrastructure and optimize database queries and resource utilization. - Monitor system performance and scale resources as needed to accommodate increasing user demand.
Financial Risk	5	Insufficient funding or financial resources may hinder the development and maintenance of the FPTJobMatch platform.	Very High	Very Low	- Conduct a thorough financial analysis to estimate the project's costs and secure adequate funding. - Explore potential funding sources such as investors, grants, or partnerships. - Implement efficient financial management practices to optimize resource allocation and control expenses.
	6	Fluctuations in the economic landscape may affect the demand for job matching platforms and impact the platform's sustainability.	High	Medium	- Monitor economic trends and anticipate shifts in the job market. - Diversify revenue streams by offering additional services or exploring new markets. - Continuously analyze and adapt pricing strategies to remain competitive and attractive to users.
Legal and Compliance Risk	7	Non-compliance with data protection and privacy regulations may result in legal consequences and damage to the platform's reputation.	Very High	Low	- Ensure compliance with relevant data protection regulations such as GDPR or CCPA. - Implement privacy policies, consent mechanisms, and secure data storage practices. - Conduct regular audits to verify data protection compliance and promptly address any identified issues.

	8	Intellectual property infringement claims or disputes with third parties over content or technology used in the platform may lead to legal complications and financial liabilities.	High	Low	- Conduct thorough intellectual property research and ensure all content and technology used are properly licensed or owned. - Implement clear terms of service and user agreements to protect the platform's intellectual property rights. - Consult with legal professionals to navigate intellectual property matters and mitigate potential risks.
Market Risk	9	Low user adoption and limited engagement from employers or job seekers may affect the success of the platform	High	Very Low	- Conduct market research and user surveys to understand user needs and preferences. Implement user-friendly features and intuitive user interfaces to enhance user experience. - Develop effective marketing and promotional strategies to attract and engage employers and job seekers.
	10	Intense competition from other job matching platforms may impact the platform's market share.	Medium	Medium	- Conduct competitive analysis to identify unique selling points and differentiate the platform. Continuously monitor the market and adapt strategies to stay competitive. - Foster partnerships and collaborations with industry stakeholders to expand the platform's reach and offerings.

B. Evaluation report:

I. Design tools:

1. Tools for designing UML Diagram:

1.1. Draw.io:

- Draw.io is a freely available, open-source collaborative workspace designed for creating UML diagrams and various other diagrams such as wireframes and business charts. It offers predefined templates and can be accessed both as a software and an online tool.

- Widely utilized by enterprises, Draw.io is compatible with enterprise browsers and seamlessly integrates with Google Drive for automatic saving of work. Notable for its user-friendly interface, it is preferred for professional diagramming purposes.

- Established by Gaudenz Alder in 2000, Draw.io supports multiple file formats including PNG, JPEG, SVG, and PDF. It is entirely free to use and does not offer any paid services. Compatible with popular browsers like Chrome, Microsoft Edge, and Mozilla Firefox, Draw.io is supported across various operating systems including Windows, Linux, and macOS.

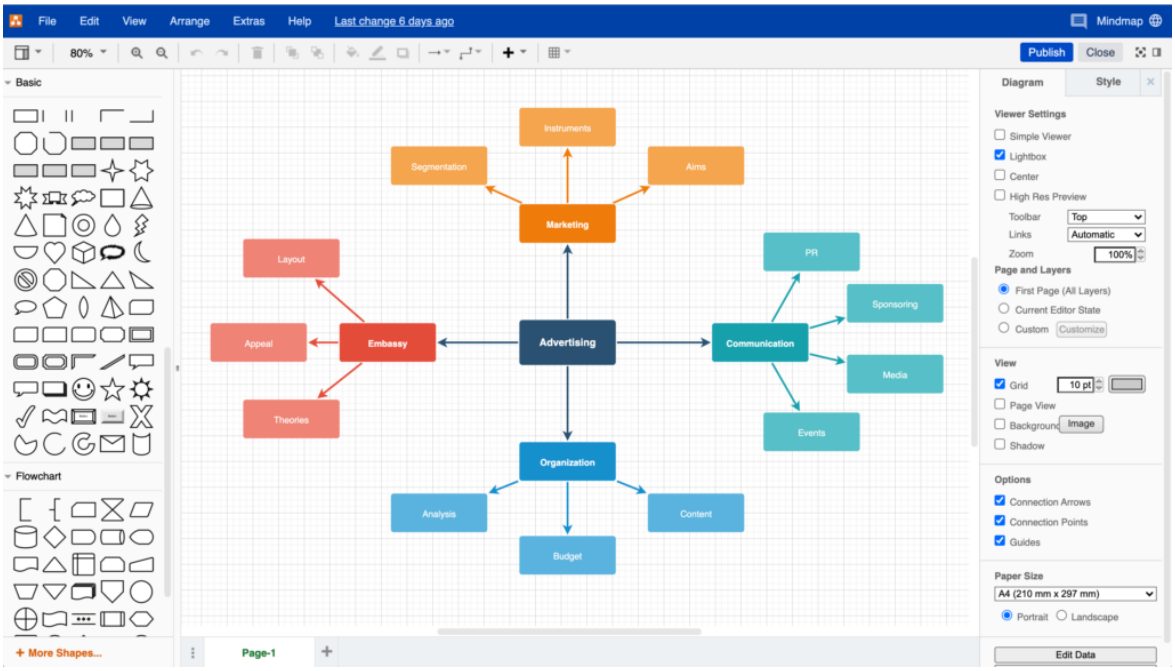

Figure 7: Draw.io

1.2. Lucidchart:

- Lucidchart is a platform tailored for drawing diagrams and charts, offering collaborative capabilities for users. Operating exclusively online, it functions as a subscription-based service. Users can access its features with a single sign-up, benefiting from its intuitive interface. With a comprehensive library of shapes, it facilitates the creation of various UML diagrams and provides essential templates for streamlined diagramming. Additionally, it aids business professionals in event planning, enhancing data visualization and project management endeavors.

- While Lucidchart offers a free trial option, it imposes limitations such as a restricted number of frames and shapes for trial users. Specifically, free users are limited to utilizing only 60 shapes per diagram. Notably, Lucidchart lacks support for enterprise browsers but serves as a versatile tool suitable for diverse project design and management needs.

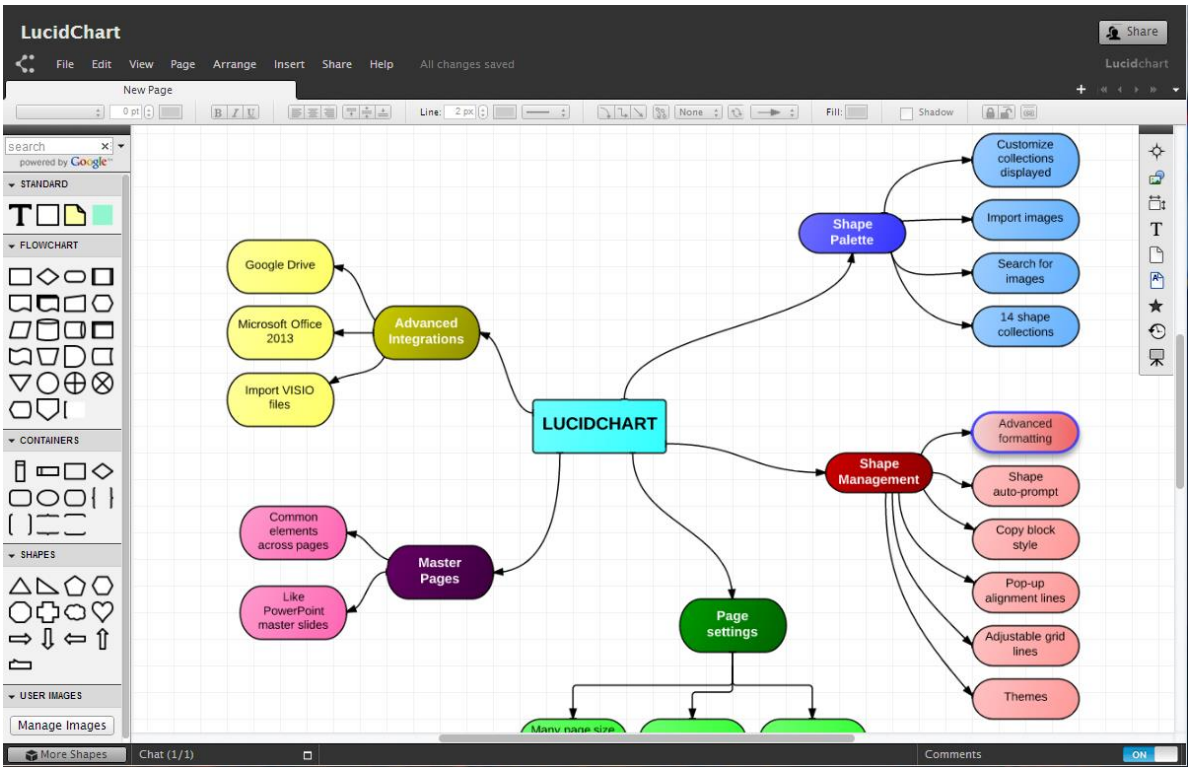

Figure 8: Lucidchart

1.3. Visual Paradigm:

- Visual Paradigm is a versatile diagramming tool utilized by businesses for planning and modeling purposes. Available in both online and software formats, users can access the online tool with a single sign-up. Featuring predefined layouts, Visual Paradigm operates on a subscription-based model with a 30-day free trial option.

- While primarily used for creating UML diagrams, its functionalities extend to diverse applications including designing business cards, brochures, book covers, and gift cards. Additionally, it serves as an image editing tool.

- Launched in 2002, Visual Paradigm offers various subscription categories such as Enterprise, Professional, Standard, and Modeler, each with distinct features and pricing.

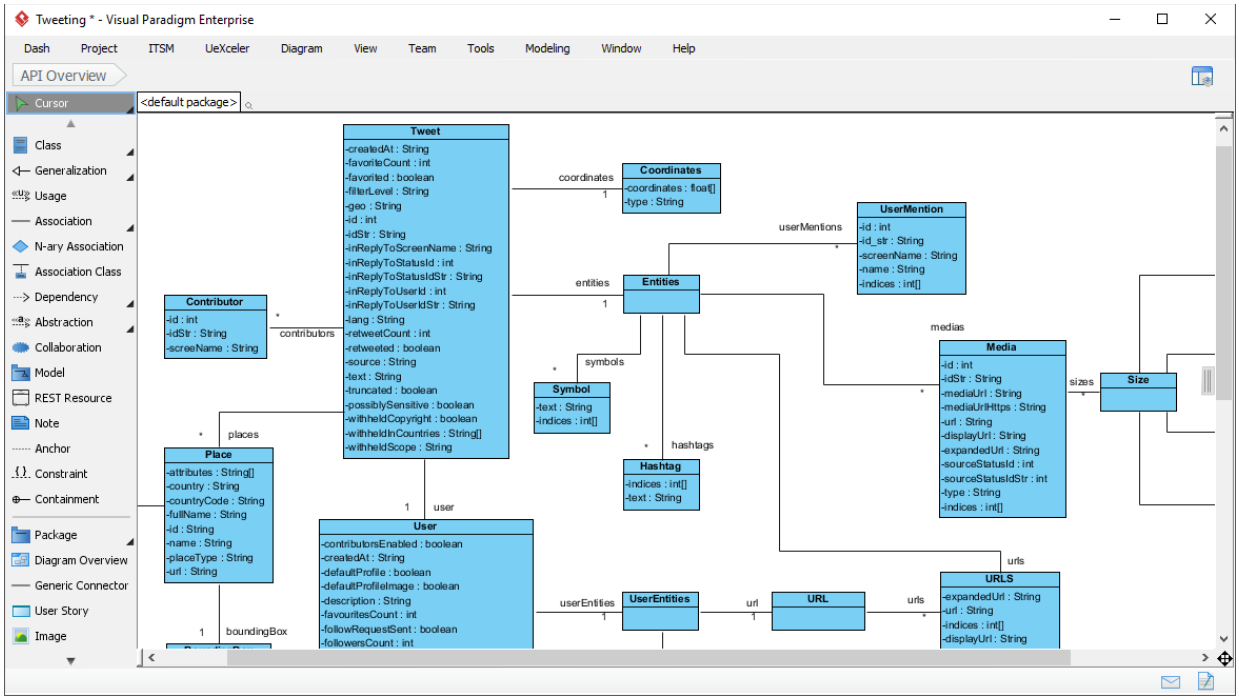

Figure 9: Visual Paradigm

2. Tools for designing User Interface:

2.1. Figma:

- Figma, a widely-used cloud-based UI/UX design platform, excels in crafting interactive prototypes, defining interactions, and implementing design transitions. It boasts a comprehensive set of features including content management, team libraries, audio communication, overlays, and version tracking. Notably, Figma facilitates AB user testing and provides a pen tool for creating arc designs.

- Additionally, Figma offers FigJam, an online whiteboard tailored for seamless collaboration. Users can incorporate notes, pin important files, monitor contributors, and collaborate with team members in real-time.

- Figma extends its functionality through integrations with various tools such as Axure, Bubble, Confluence, Flinto, Maze, Microsoft Teams, Pendo, Principle, ProtoPie, Sprig, and Zeplin, enhancing its versatility and compatibility within design workflows.

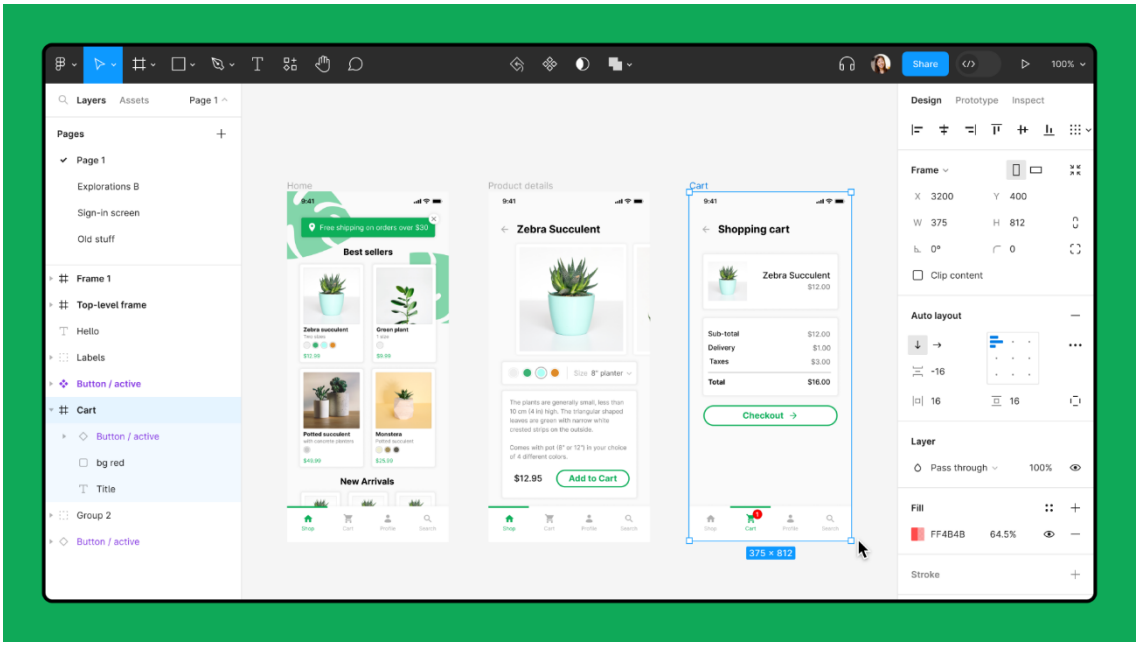

Figure 10: Figma

2.2. InVision:

- InVision stands out as an advanced UI/UX design platform tailored for software product development. It provides UX/UI designers with a unified dashboard for design creation, experimentation, testing, and collaboration. Key features encompass digital whiteboarding, prototype development, wireframing, and feedback management.

- Within the platform, users can style layers, animate drawings, and edit vectors, leveraging functionalities like editing timelines, instant playback, and mobile device mirroring. Additionally, InVision offers shared design libraries, version control with rollback capabilities, and streamlined push/pull changes.

- The platform seamlessly integrates with popular tools such as Confluence, Microsoft Teams, Trello, and Slack, enhancing its compatibility and facilitating efficient workflow integration for design teams.

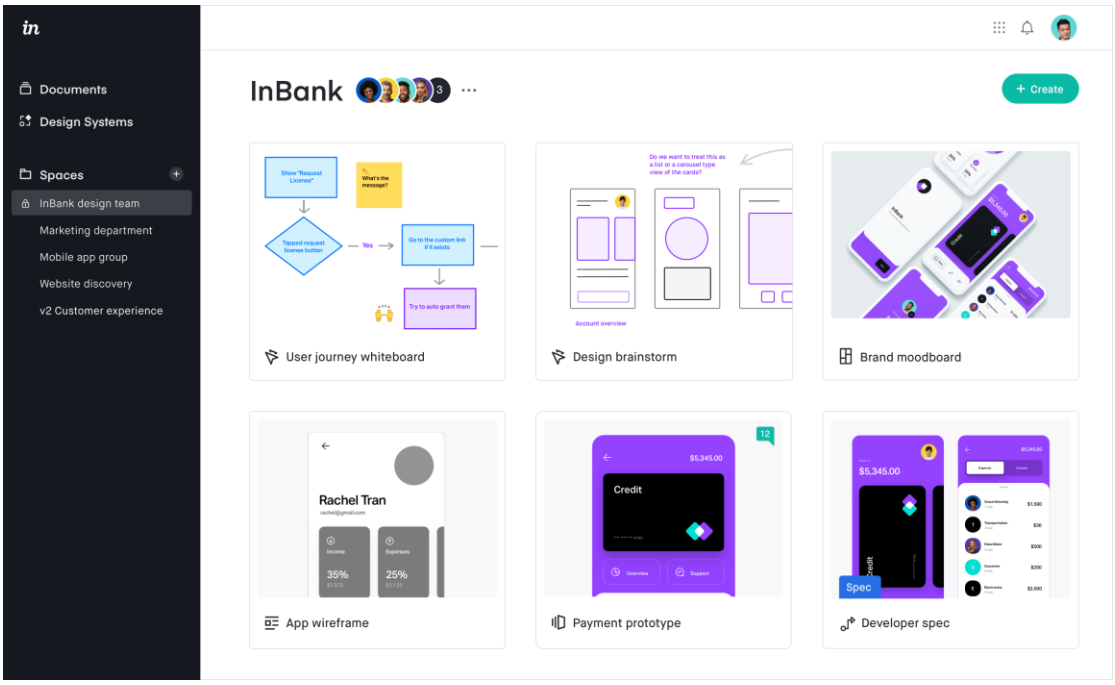

Figure 11: InVision

2.3. Mockplus:

- Mockplus offers a comprehensive solution empowering UX designers to swiftly craft sketches, wireframes, and high-fidelity prototypes. Its web-based interface boasts an intuitive drag-and-drop editor, simplifying the creation process, even for intricate designs.

- The collaborative feature stands out, enabling seamless teamwork among UX design team members for accelerated project development. Designers can effortlessly gather feedback from teammates, stakeholders, clients, and real users through a single link. Further enhancing collaboration, designs can be synced to the cloud for instant review, comments, and developer handoff.

- Mockplus supports the creation of user interface designs for various platforms, including iOS, Android, and the web.

- The platform integrates smoothly with a range of popular tools such as Adobe XD, Axure, Confluence, Figma, Jira, Lark, Photoshop, Sketch, and Slack, facilitating a cohesive workflow and enhanced productivity for design teams.

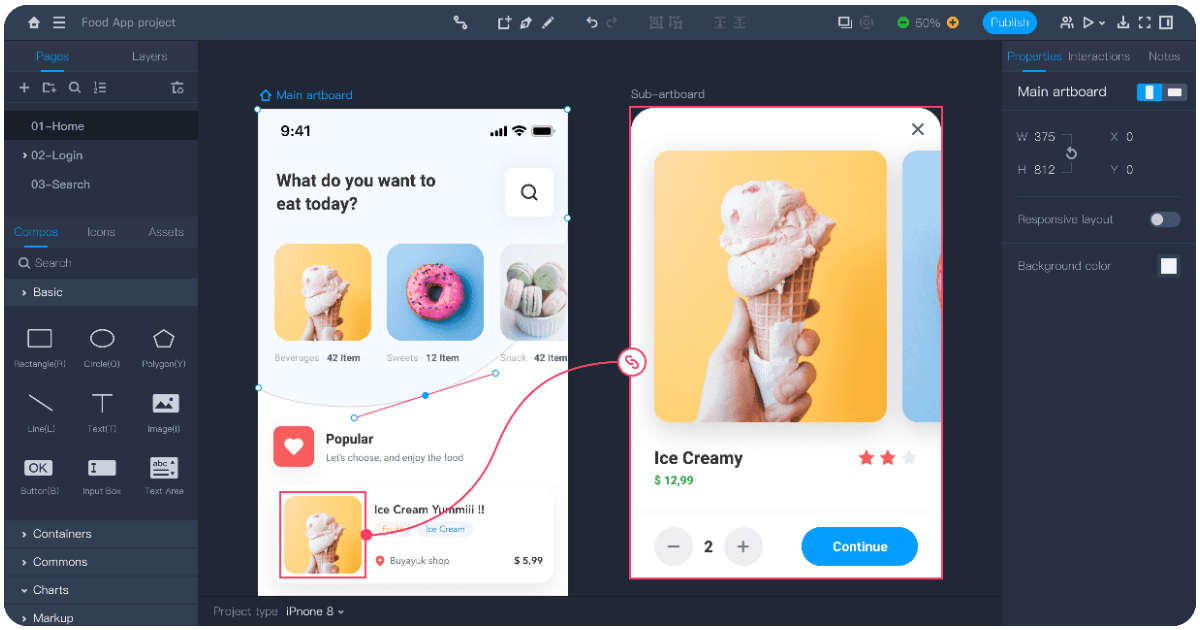

Figure 12: Mockplus

3. Conclude which design tool will be used for the development:

- For designing UML Diagram, I choose Draw.io. It is a versatile and user-friendly diagramming tool that offers predefined templates and can be accessed both as a software and an online tool. Draw.io is widely used by enterprises, compatible with enterprise browsers, and seamlessly integrates with Google Drive. With support for multiple file formats and its availability across various operating systems, Draw.io is a preferred choice for professionals in need of a powerful and intuitive tool for UML diagramming.

II. Front End technology:

1. Languages of Front End:

1.1. HTML:

- HTML (Hypertext Markup Language) is the standard markup language used for creating the structure and presenting content on the World Wide Web. It is the fundamental building block of web pages and is responsible for defining the structure and organization of the content within a web document.

- HTML uses a set of markup tags or elements to define different elements of a webpage such as headings, paragraphs, images, links, tables, forms, and more. These elements are enclosed within angle brackets (< >) and can include attributes that provide additional information about the elements.

- HTML is interpreted by web browsers to render the content and display it visually. It enables the browser to understand the hierarchy and meaning of the content, and how it should be displayed to the user. In combination with CSS (Cascading Style Sheets) and JavaScript, HTML forms the core technologies used to create visually appealing and interactive websites.

Figure 13: HTML

1.2. CSS:

- CSS (Cascading Style Sheets) is a stylesheet language used for describing the presentation and formatting of a document written in HTML or XML. It is a fundamental technology used in web development to control the visual appearance of web pages.

- CSS works by separating the content of a webpage from its presentation, allowing developers to apply styles and layouts to HTML elements. It provides a set of rules that define how elements should be displayed, including properties like color, font size, layout, spacing, and more.

- CSS is often used in conjunction with HTML and JavaScript to create dynamic and interactive web experiences. It allows web designers and developers to create visually appealing and user-friendly websites that are responsive and adaptable to different screen sizes and devices.

Figure 14: CSS

1.3. JavaScript:

- JavaScript is a high-level, interpreted programming language that is primarily used to add interactivity and dynamic behavior to web pages. It is one of the core technologies used in web development alongside HTML and CSS.

- JavaScript can be embedded directly into HTML code or included as an external script file. It provides a wide range of features and capabilities, allowing developers to manipulate the content and structure of a web page, handle user interactions, and communicate with the web server.

- Besides, JavaScript is an event-driven language, meaning that it can respond to events such as user actions (e.g., clicking a button, submitting a form), page loading, or timer events. It can also modify and update the HTML and CSS of a web page in real-time, enabling dynamic effects and animations.

Figure 15: JavaScript

1.4. CSS preprocessor:

1.4.1. Definition of CSS preprocessor:

- CSS Preprocessors are understood as languages for processing CSS. Its task is to logically process CSS code to be more similar to programming languages. Writing CSS preprocessors brings the following benefits:

- Saves time writing CSS.
- Easy to maintain and develop.
- Highly flexible and reusable.
- CSS files and code snippets are organized and arranged in a clear manner.

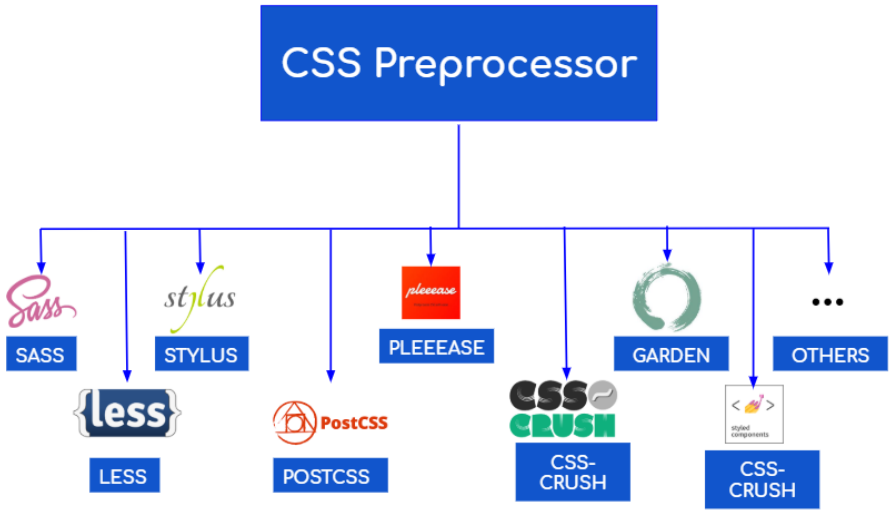

Figure 16: CSS Preprocessor

1.4.2. Some popular CSS preprocessors:

- **SCSS (Sassy CSS)**: SCSS is a superset of CSS syntax, meaning that any valid CSS code is also valid SCSS code. It uses the file extension ".scss" and is often referred to as "SCSS syntax." SCSS introduces additional features like variables (to store and reuse values), nesting (to nest CSS selectors inside one another), mixins (to define reusable styles), and more. It maintains a similar syntax to CSS, making it easy for developers familiar with CSS to transition to SCSS.

- **SASS (Syntactically Awesome Style Sheets)**: SASS is an older version of the SASS language and uses a different syntax compared to SCSS. It uses the file extension ".sass" and is often called "SASS syntax." SASS syntax is indentation-based and doesn't require the use of semicolons or curly braces. It also supports the same features as SCSS, such as variables, nesting, mixins, and functions. However, due to its different syntax, it can have a steeper learning curve for developers accustomed to CSS.

- **LESS (Leaner Style Sheets)**: LESS is another CSS preprocessor that extends CSS with additional features. It uses the file extension ".less." Similar to SCSS and SASS, LESS introduces features like variables, nesting, mixins, functions, and more. LESS syntax is similar to CSS but introduces its own set of rules and conventions. It provides a slightly different feature set compared to SCSS and SASS but serves the same purpose of enhancing CSS authoring.

2. JavaScript library/ framework:

2.1. React.js:

- React.js is a JavaScript framework developed by Facebook that simplifies the creation of interactive user interfaces. Its popularity can be attributed to several factors, including a large and active community, strong support from Facebook, a rich ecosystem, improved performance, and the concept of reusable components. React.js is particularly well-suited for building single-page applications (SPAs) and cross-platform applications, and it is also widely used in developing small business applications. Additionally, React.js serves as the foundation for React Native, a framework used for building mobile applications.

- One of the key aspects that sets React.js apart is its one-way data flow, also known as unidirectional data flow. This data flow model provides a more intuitive approach compared to bi-directional data binding, making it easier for developers to understand and manage the flow of data within their applications.

- Another highly acclaimed feature of React.js is hot reloading. This functionality allows developers to witness instant updates and view changes in real-time as they make modifications to their code. Hot reloading minimizes the need for manual page refreshing, thereby enhancing the development process and boosting productivity.

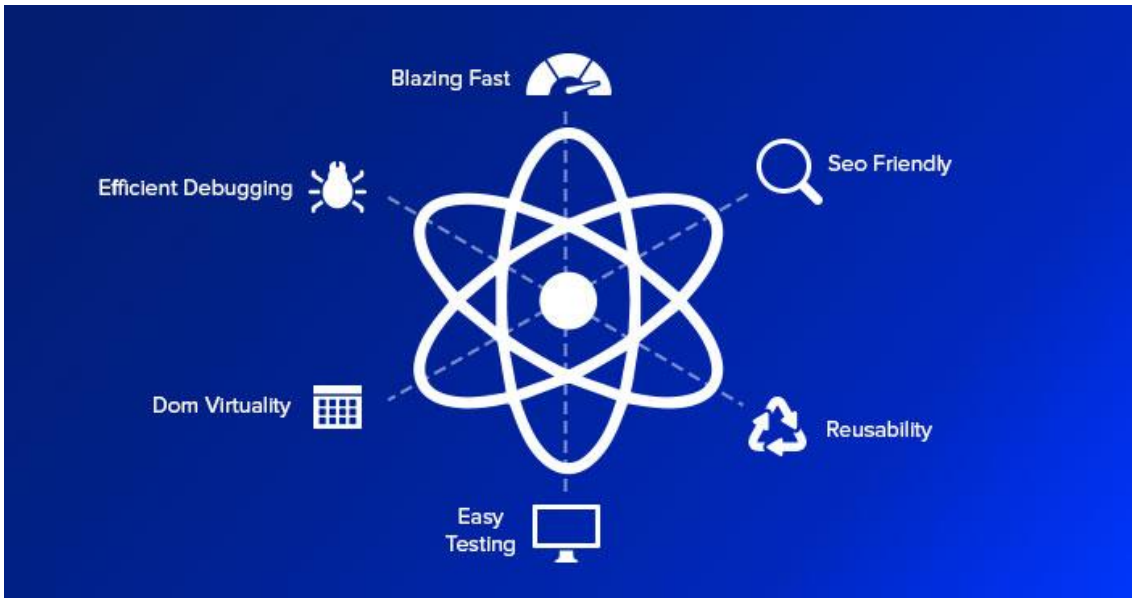

Figure 17: React.js

2.2. AngularJS:

- AngularJS, maintained by Google, is a framework designed to address common challenges in single-page application (SPA) development. It revolutionizes the use of HTML by leveraging its vocabulary to create dynamic web pages, as opposed to its previous limitation to static content.

- In SPA development, AngularJS enables the loading of content from the web server dynamically, eliminating the need for page reloading. This behavior aligns SPA with the experience of using mobile applications, providing a seamless and interactive user experience.

- AngularJS serves as an entry point for many JavaScript developers due to its widespread adoption and versatility. The framework's success can be attributed to its adaptability and rich feature set. It offers two-way data binding, dependency injection, a powerful command-line interface (CLI), directives for extending HTML, support for TypeScript, and follows the Model-View-Controller (MVC) architectural pattern, among many other features.

Figure 18: AngularJS

2.3. Vue.js:

- Vue.js is known as the "progressive" JavaScript framework due to its incremental adoption philosophy. The framework's core library focuses solely on the view layer, allowing new functionalities to be added gradually.

- Vue.js follows the Model-view-view model (MVVM) architectural pattern, which separates the graphical user interface (UI) or view from the application's business logic or model. The view model acts as a mediator for data synchronization.

- The framework's rapid growth can be attributed to its small size, comprehensive documentation, reactivity, reusability, TypeScript support, and short learning curve. Vue.js offers a rich collection of unique elements and is particularly popular in Asia. It excels at building modern, high-performance applications and is often favored for quick prototyping.

Figure 19: Vue.js

3. CSS framework:

3.1. Bulma:

- Bulma is an open-source CSS framework built on Flexbox, designed for responsiveness. It offers an extensive array of pre-designed features, reducing the need for manual CSS coding. The framework employs tiles for crafting Metro-style grids, facilitating smooth page layouts. Users can opt to import only the necessary elements, simplifying the setup process.

- With Bulma's freely available source code, users have the flexibility to extend its functionality according to their requirements. It advocates for a straightforward CSS-only approach, devoid of JavaScript components, and provides visually appealing default styles to kickstart projects.

- Notably, Bulma has gained significant traction within the Laravel community, contributing to its growing popularity.

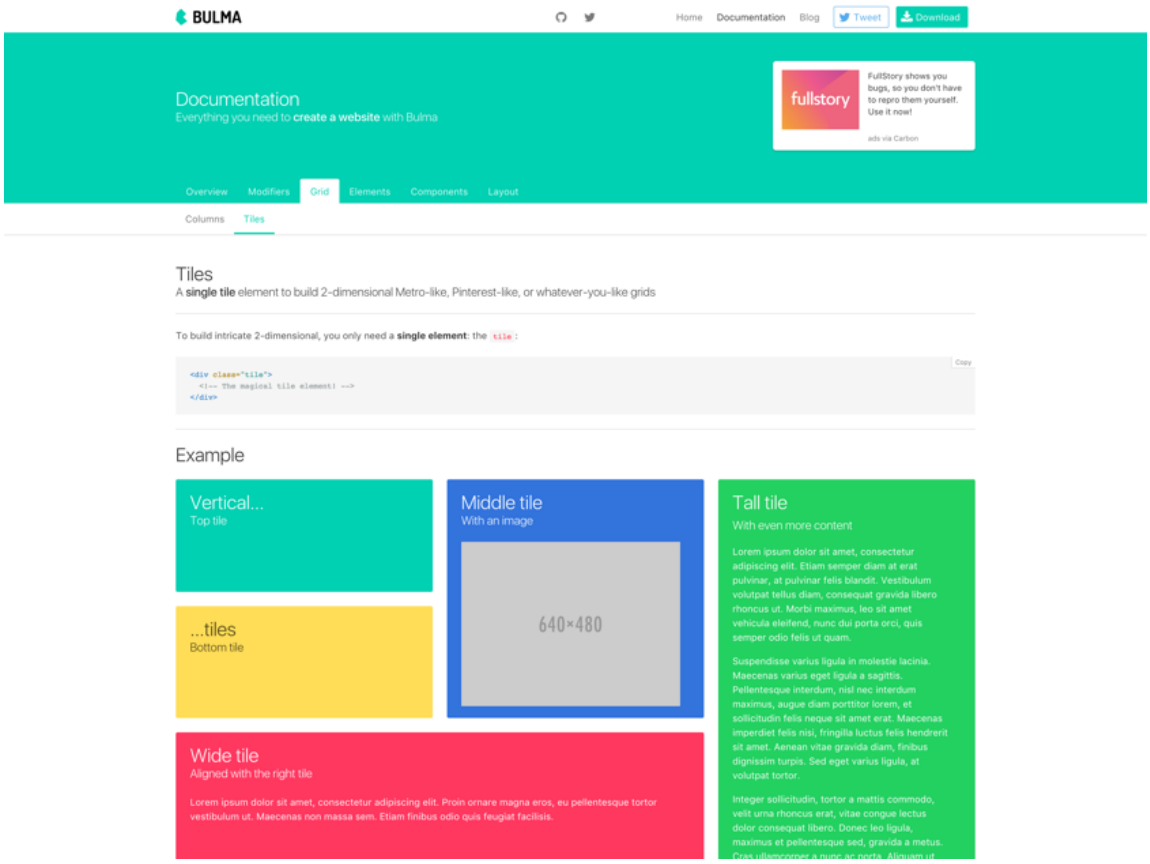

Figure 20: Bulma

3.2. Tailwind CSS:

- Tailwind CSS is described in its official documentation as a "utility-first CSS framework" that provides a comprehensive set of classes allowing users to craft custom UI designs directly within their markup. It simplifies the process of implementing inline styling, enabling users to create visually striking interfaces without the need to write traditional CSS code.

- As one of the most popular utility CSS libraries available, Tailwind CSS offers significant advantages for web design. Its creator, Adam Wathan, initially had to persuade developers of the superiority of utility-based CSS over semantic CSS. However, his efforts eventually gained traction, leading to widespread adoption of Tailwind CSS within the development community. Today, Tailwind CSS is frequently featured on lists of essential web development tools due to its efficiency and effectiveness in streamlining the styling process.

Figure 21: Tailwind CSS

3.3. Bootstrap:

- Bootstrap, created by Jacob Thornton and Mark Otto at Twitter, is an open-source framework featuring CSS and JavaScript templates for interface components, aiming to maintain consistency across internal tools.

- One of Bootstrap's significant contributions is its popularization of responsive design principles among web developers. It emphasized the importance of mobile-first approaches and provided tools for their seamless implementation. This was achieved through the introduction of a grid system, enabling developers to partition screens into columns for better layout control.

- Bootstrap simplifies the process of adapting websites for smaller screen sizes, eliminating the need for separate projects. Developers can integrate Bootstrap classes into their code, allowing the design to automatically adjust itself accordingly.

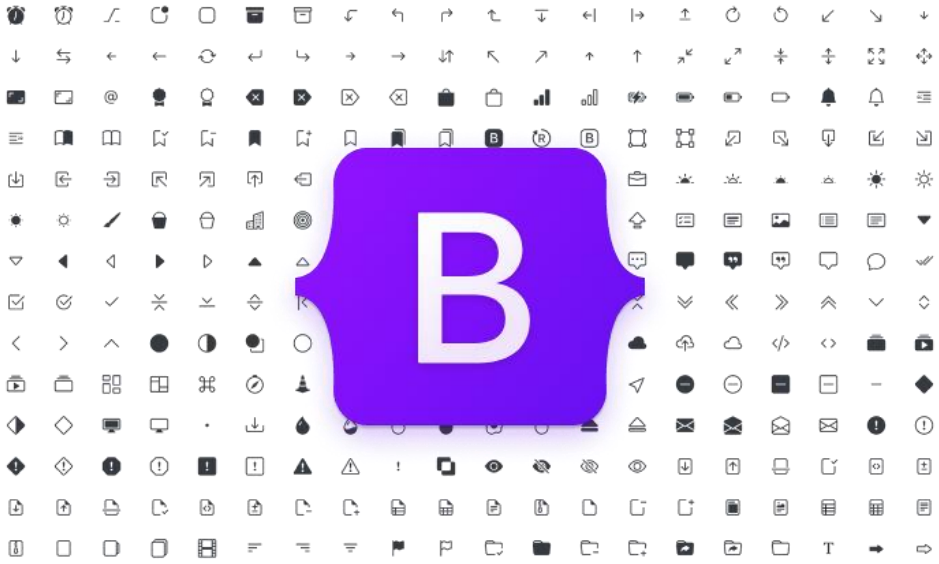

Figure 22: Bootstrap

4. Conclude which Front End technologies will be used for the development:

- For languages of Front End, I choose HTML, CSS, and JavaScript. HTML provides the structure and content of web pages, CSS is used for styling and formatting, and JavaScript adds interactivity and dynamic behavior.

- For the CSS framework, I choose Bootstrap. It offers a responsive grid system, pre-designed components, and utility classes that simplify the process of building responsive and mobile-first web applications. Bootstrap's emphasis on responsive design ensures a consistent user experience across devices. Its pre-styled components can be easily customized, saving development time. Additionally, Bootstrap has a large and active community, providing extensive documentation and support for developers.

III. Back End technology:

1. Languages of Back End:

1.1. C#:

- C# (C-sharp) is a powerful and versatile language in the realm of backend development, backed by Microsoft. It is renowned for its strong typing, security features, and seamless integration with Microsoft technologies. C# is an excellent choice for building Windows applications and robust web backends.

- When combined with the ASP.NET framework, C# becomes even more formidable in web development. ASP.NET provides a comprehensive set of tools for creating dynamic web applications with enhanced security measures. Its Model-View-Controller (MVC) architecture promotes the separation of concerns, allowing developers to build maintainable and scalable applications.

- The robust type system of C# and its integrated support for features like garbage collection contribute to the development of secure and memory-efficient applications. Furthermore, C# seamlessly integrates with the Microsoft ecosystem, enabling smooth collaboration with tools like Visual Studio and Azure. This integration simplifies deployment and management processes for developers.

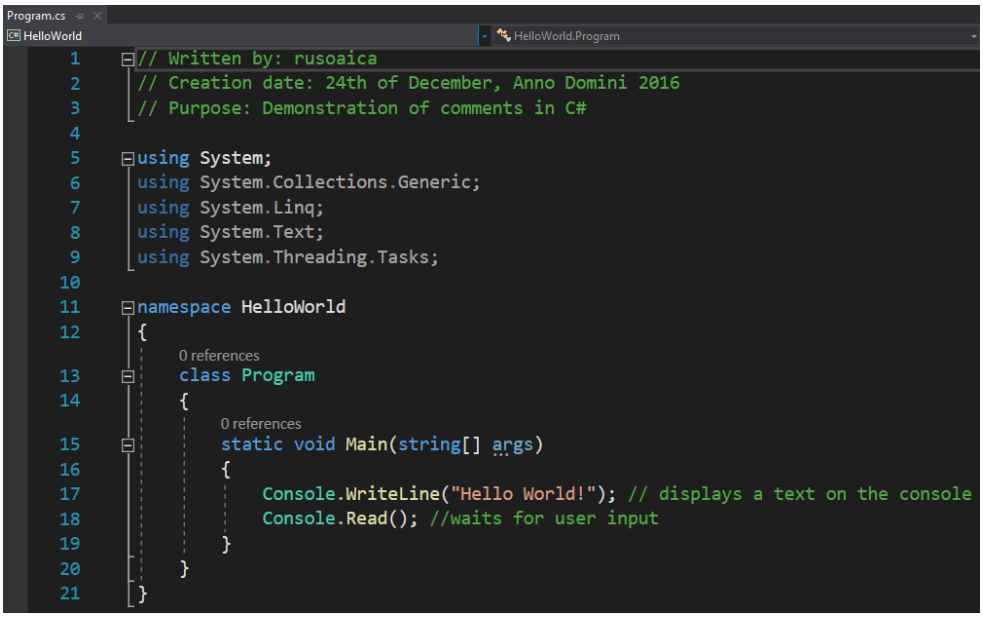

Figure 23: C#

1.2. Java:

- Java is highly regarded in the field of backend development for its platform independence and robustness. It is a popular choice, especially for enterprise-level applications and projects that require careful architectural planning. Java's mature ecosystem, strong typing, and scalability contribute to its reputation as a reliable backend language.

- One of Java's standout features is its ability to run on multiple platforms without requiring modifications. This is achieved through the "write once, run anywhere" principle, ensuring that Java applications can seamlessly function on different operating systems. This cross-platform compatibility enhances accessibility and mitigates compatibility issues.

Figure 24: Java

1.3. Python:

- Python is a high-level programming language known for its interpreted nature and object-oriented design, featuring dynamic semantics. Its built-in data structures and dynamic typing, along with dynamic binding, make it particularly appealing for Rapid Application Development and as a scripting language for integrating existing components. Python's clear and easy-to-understand syntax prioritizes readability, reducing the burden of program maintenance. Additionally, Python supports modules and packages, fostering code modularity and reuse. The Python interpreter and its extensive standard library are freely available in both source and binary forms for all major platforms, allowing for unrestricted distribution.

Figure 25: Python

2. Operating system:

2.1. Windows:

- Windows is a widely used operating system developed by Microsoft. It provides a graphical user interface (GUI) and supports a wide range of software applications. Windows is known for its user-friendly interface, extensive software compatibility, and widespread usage in personal computers, laptops, and servers.

Figure 26: Windows

2.2. MacOS:

- MacOS is the operating system developed by Apple Inc. for its line of Macintosh computers. It is known for its sleek design, seamless integration with Apple hardware and software, and focus on user experience. macOS offers a visually appealing interface, advanced productivity features, and a robust ecosystem of applications optimized for Apple devices.

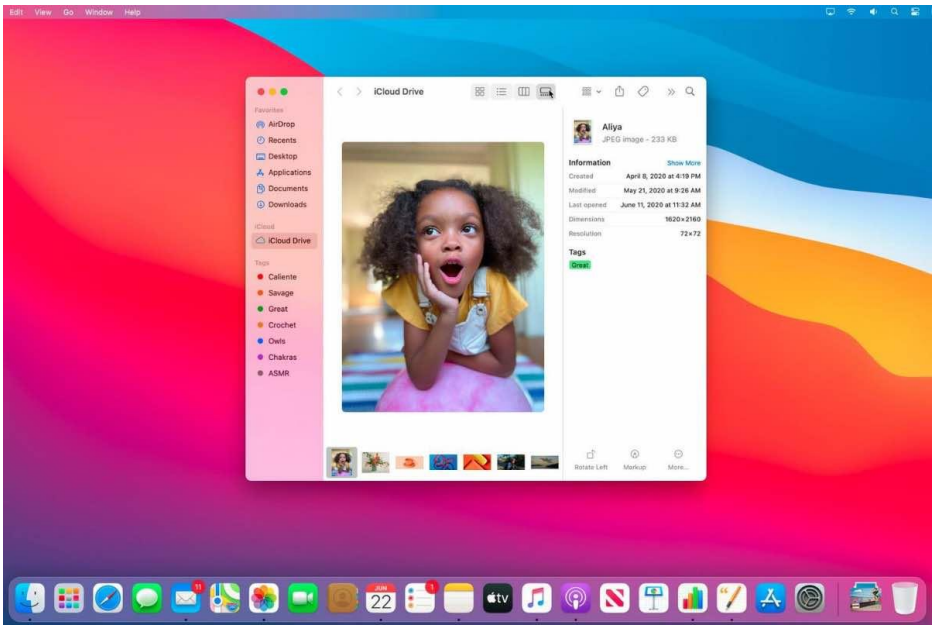

Figure 27: MacOS

2.3. Linux:

- Linux is an open-source operating system based on the Unix operating system. It is known for its stability, security, and flexibility. Linux is available in various distributions, or "distros," each offering its own set of features and package management systems. Linux is widely used in servers, embedded systems, and as the operating system for many devices, including smartphones and Internet of Things (IoT) devices. It is highly customizable, and its open-source nature allows for extensive community contributions and development. Linux distributions are often favored by developers, system administrators, and those seeking a free and open-source alternative to proprietary operating systems.

Figure 28: Linux

3. Web server:

3.1. Apache HTTP Server:

- Apache HTTP Server is an open-source web server that was initially developed by the Apache Group and first released on March 25, 1999. It is specifically designed for Unix, Linux, and Solaris platforms.

- Apache is currently the most widely used web server application globally. It is highly regarded for its strong security measures, speed, and reliability.

- Apache operates in a process-based architecture, creating new threads for each connection. It also includes features such as a built-in database for authentication and the ability to customize responses to errors and other issues. Multiple directory index directives have been thoroughly tested by both developers and users to ensure optimal performance.

Figure 29: Apache HTTP Server

3.2. Nginx:

- Nginx is a web server developed by Nginx Inc and was first released on October 4, 2004.

- Apart from its primary role as a web server, Nginx can also function as a reverse proxy server, allowing it to modify and redirect client requests to proxy servers.

- Additionally, Nginx is efficient in reverse caching and excels in serving static files like JS and CSS files. It enhances content and application quality while ensuring security.

- Being open-source, Nginx is known for its exceptional speed, lightweight nature, and high-performance capabilities in serving static files. Noteworthy companies such as IBM, Google, GitLab, DuckDuckGo, etc, rely on Nginx for their web server needs.

Figure 30: Nginx

3.3. IIS:

- Internet Information Services (IIS) is a versatile and widely-used web server developed by Microsoft. It is designed to run on Windows systems and is responsible for serving requested HTML pages or files.

- The primary function of an IIS web server is to accept requests from remote client computers and provide the appropriate response. This fundamental capability enables web servers to share and deliver information across various networks, including local area networks (LANs) like corporate intranets and wide area networks (WANs) such as the Internet.

- A web server powered by IIS can deliver information to users in various formats. It can serve static webpages created using HTML, facilitate file exchanges through downloads and uploads, and transmit documents in formats such as text files and image files. This flexibility allows web servers to cater to a wide range of user needs and deliver content efficiently.

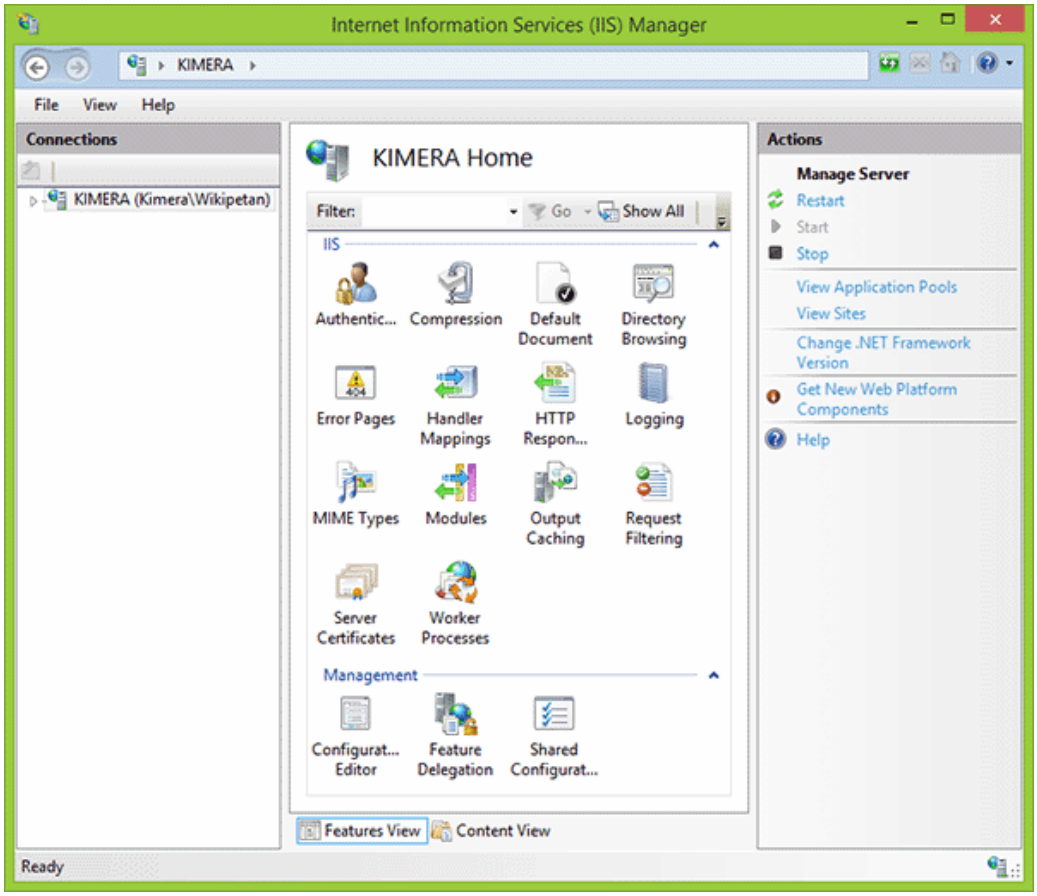

Figure 31: IIS

4. Database:

4.1. Microsoft SQL Server:

- Microsoft SQL Server is a widely used RDBMS developed by Microsoft. It provides a comprehensive set of tools and features for managing, querying, and securing databases. SQL Server offers robust transactional support, scalability, and integration with other Microsoft products and technologies. It is commonly used in enterprise-level applications and environments that require high performance and advanced database management capabilities.

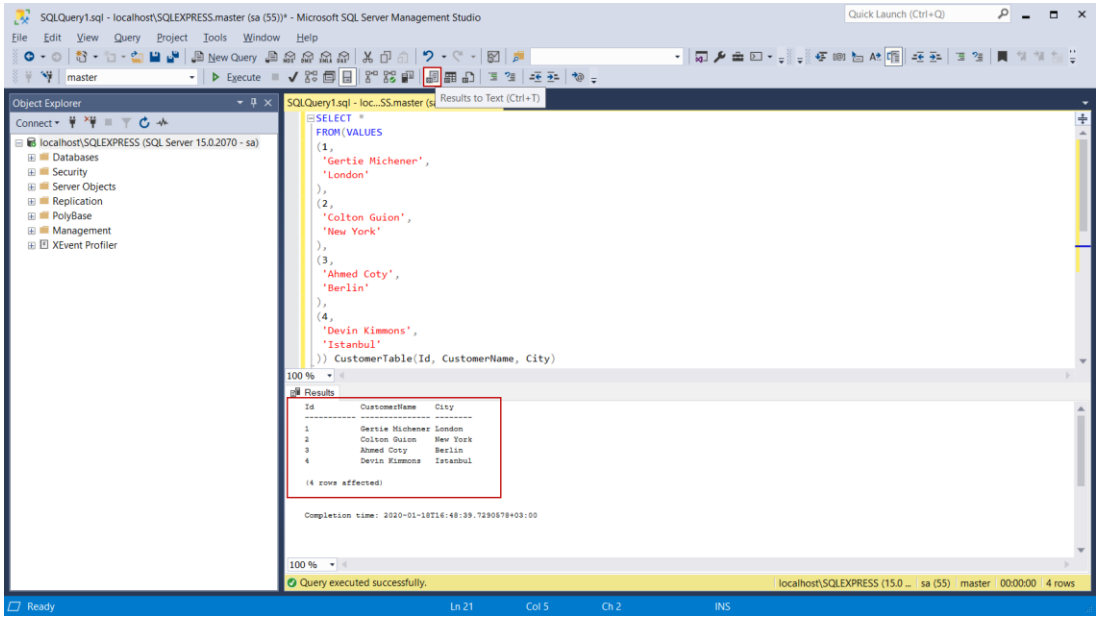

Figure 32: Microsoft SQL Server

4.2. MySQL:

- MySQL is an open-source RDBMS known for its speed, reliability, and ease of use. It is widely used in web applications and smaller-scale projects. MySQL supports multiple storage engines, allowing users to choose the one that best suits their specific requirements. It offers good performance, scalability, and compatibility with various operating systems and programming languages.

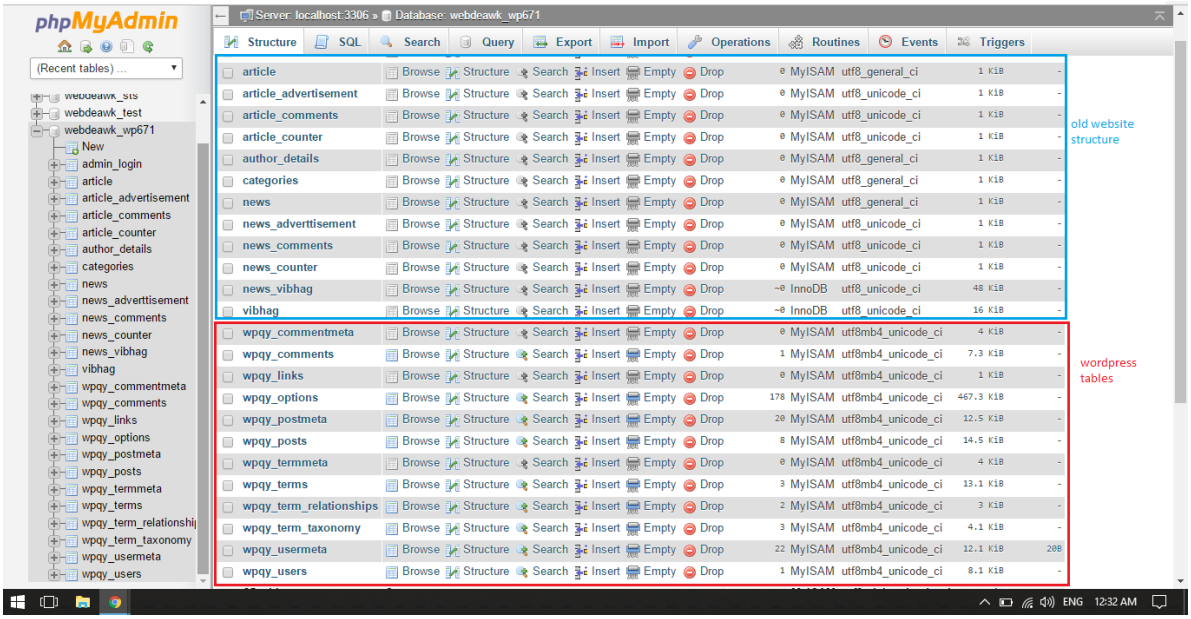

Figure 33: MySQL

4.3. PostgreSQL:

- PostgreSQL, also known as Postgres, is an open-source RDBMS known for its advanced features, extensibility, and robustness. It provides strong support for complex data types, advanced indexing, and transactions. PostgreSQL is highly customizable and offers a wide range of extensions and plugins, making it suitable for diverse application scenarios. It is often favored by developers who require advanced database features and flexibility.

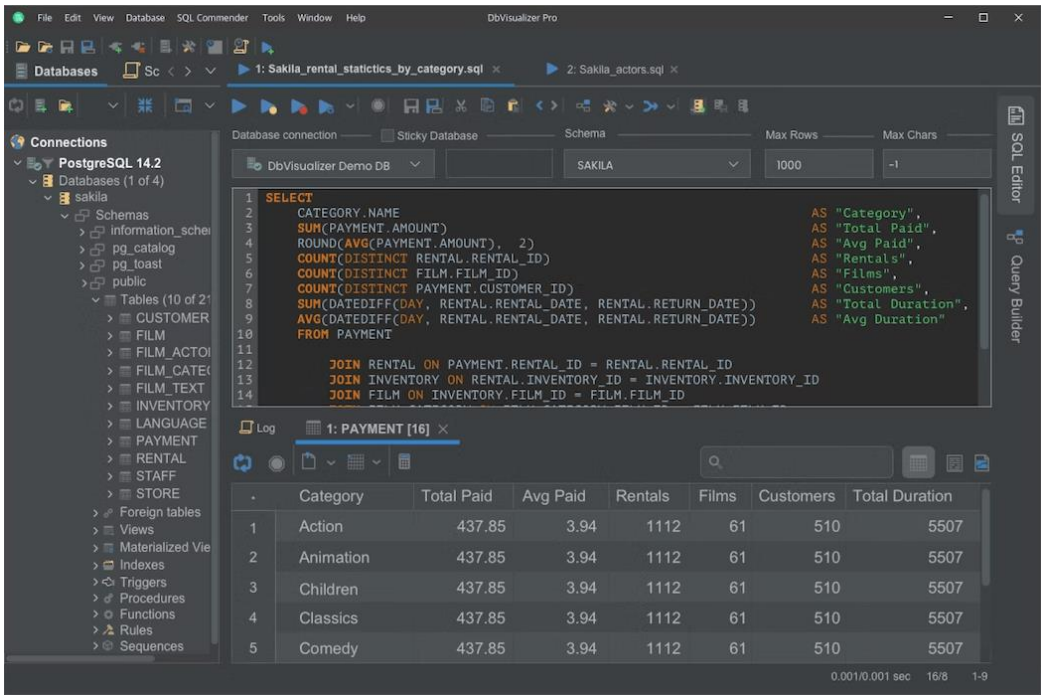

Figure 34: PostgreSQL

5. Hosting:

5.1. AWS:

- Amazon Web Services (AWS) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. AWS offers a broad set of global cloud-based products including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security, and enterprise applications.

Figure 35: AWS

5.2. Google Cloud:

- Google Cloud Platform, offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics, and machine learning.

Figure 36: Google Cloud

5.3. Azure:

- Azure, created by Microsoft, is a cloud computing service for building, testing, deploying, and managing applications and services through Microsoft-managed data centers. It provides software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) and supports many different programming languages, tools, and frameworks, including both Microsoft-specific and third-party software and systems.

Figure 37: Azure

6. Frameworks:
6.1. ASP.Net Core:

- ASP.NET Core is a high-performance, open-source framework for building modern, cloud-enabled, Internet-connected apps. ASP.NET Core is designed to allow runtime components, APIs, compilers, and languages to evolve quickly while providing a stable and supported platform to keep apps running. It supports development on Windows, macOS, and Linux.

Figure 38: ASP.Net Core

6.2. Laravel:

- Laravel is a free, open-source PHP framework intended for the development of web applications following the Model–View–Controller (MVC) architectural pattern. It provides a robust set of tools and resources for building modern PHP web applications and is known for its elegant syntax.

Figure 39: Laravel

6.3. Express.js:

- Express.js is the most popular web framework for Node.js. It's designed for building web applications and APIs and is known as the de facto standard server framework for Node.js. Express.js simplifies the server creation process in Node.js, allowing developers to focus more on writing the business logic of an application.

Figure 40: Express.js

7. Conclude which Back End technologies will be used for the development:

- For the language of the back end, I choose C#. C# is a versatile and powerful language supported by Microsoft, making it an excellent choice for backend development. When combined with the ASP.NET framework, C# becomes even more formidable in web development, offering enhanced security measures and a Model-View-Controller (MVC) architecture for building maintainable and scalable applications.

- As for the operating system, I choose Windows. Windows is a widely used operating system developed by Microsoft, known for its user-friendly interface and extensive software compatibility. Choosing Windows as the operating system for the back-end development aligns well with the use of C# and the ASP.NET framework, which are both developed and optimized for the Windows platform.

- Regarding the database, I choose Microsoft SQL Server. Microsoft SQL Server is a widely used Relational Database Management System (RDBMS) known for its robustness and comprehensive set of tools. It integrates well with the Microsoft ecosystem and offers advanced database management capabilities, making it a suitable choice for enterprise-level applications.

- Finally, for the framework, I choose the ASP.NET Core framework. ASP.NET Core is a web application framework provided by Microsoft that enables developers to build dynamic and scalable web applications. It integrates seamlessly with the C# language and offers features like authentication, routing, and caching, making it an excellent choice for developing robust backend systems.

IV. Tools for source control management:

1. GitHub:

- GitHub is a web-based platform designed for software developers, offering version control and collaboration functionalities. Founded in 2008, GitHub operates on a software as a service (SaaS) model and is built upon Git, an open-source code management system developed by Linus Torvalds to streamline software builds.

- Git serves as the backbone of GitHub, enabling developers to store project source code and monitor the complete evolution of changes made to that code. This fosters efficient collaboration among developers, facilitating the management of potentially conflicting alterations from various contributors.

- GitHub provides free access to public repositories, allowing developers to modify, customize, and enhance software hosted within them. However, it offers paid plans for private repositories, which include diverse features and benefits. Each repository, whether public or private, encompasses all files related to a project along with their revision history. Additionally, repositories can accommodate multiple collaborators, enhancing teamwork and enabling projects to be developed either openly or privately.

Figure 41: GitHub

2. GitLab:

- GitLab is a web-based Git repository offering both free open and private repositories, along with features such as issue tracking and wikis. Serving as a comprehensive DevOps platform, GitLab enables professionals to

seamlessly handle all project tasks - from planning and source code management to monitoring and security. Its collaborative environment empowers teams to collectively build superior software.

- By streamlining processes, GitLab aids in reducing product lifecycles and boosting productivity, thereby delivering enhanced value to customers. Unlike traditional setups requiring users to manage authorizations for individual tools, GitLab simplifies access control by setting permissions organization-wide, ensuring universal accessibility to all components.

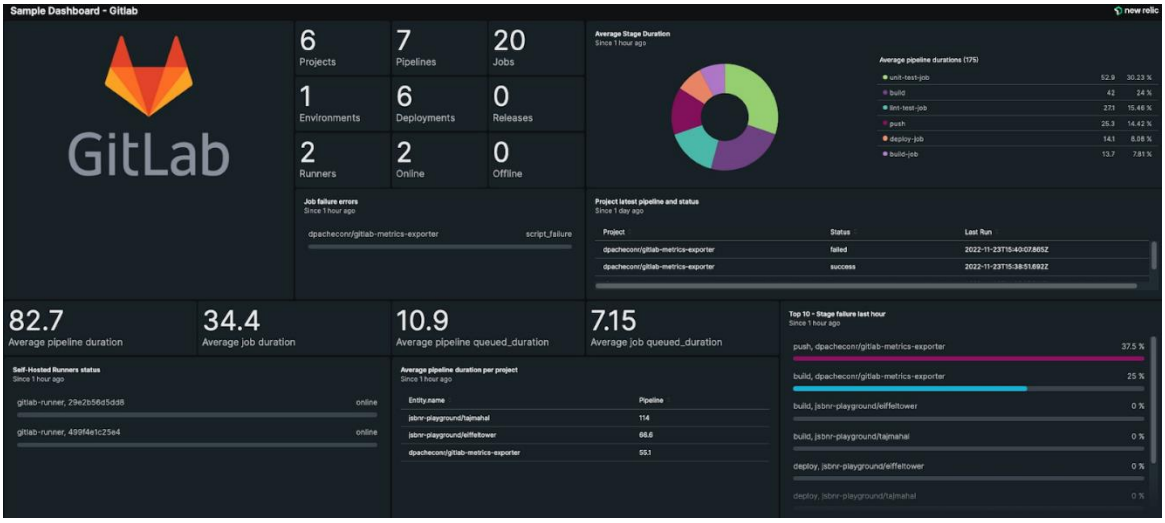

Figure 42: GitLab

3. BitBucket:

- BitBucket is a cloud-based platform designed to assist developers in storing, managing, tracking, and controlling changes to their code. Offering a cloud-based Git repository hosting service, BitBucket boasts a user-friendly interface that even beginner coders can easily navigate, whereas Git typically demands more technical expertise and command-line usage. Beyond basic code management, BitBucket extends its services to facilitate team collaboration, project creation, as well as testing and deployment of code.

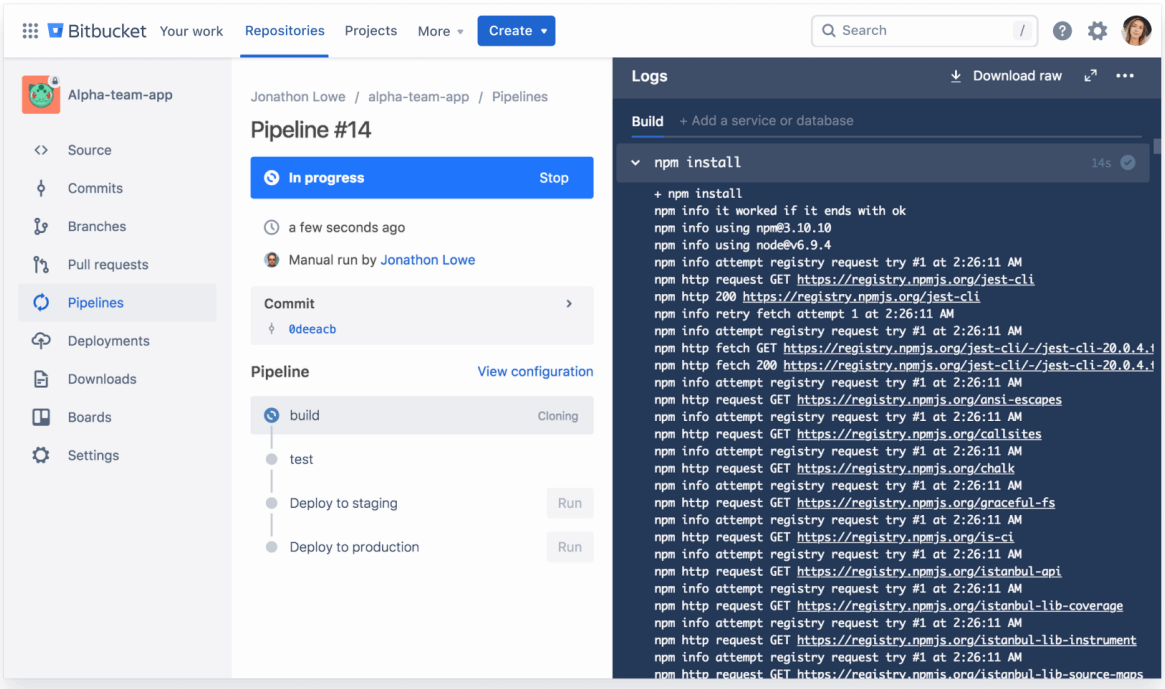

Figure 43: BitBucket

4. Conclude which tool will be used for the development:

- I decided to choose GitHub as the preferred hosting platform due to its robust and effective system. Its seamless integration with Git ensures proficient tracking and management of source code versions, enhancing overall efficiency.

V. Software Development Models:

1. Waterfall Model:

1.1. Definition of Waterfall Model:

- The Waterfall Model is the first Process Model introduced. It is very simple to understand and use. In the waterfall model, each stage must be completed before the next can begin, and there is no overlap in the stages.

- In addition, it is also known as the linear sequential life cycle model because it illustrates the software development process in a linear sequential flow. This shows that any stage in the development process begins only if the previous phase is complete and the stages do not overlap.

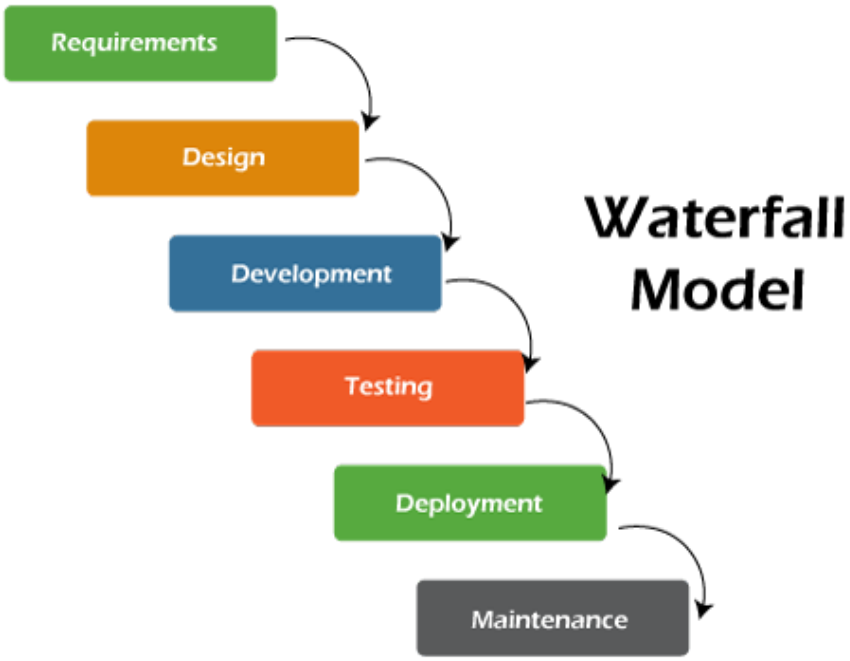

Figure 44: Waterfall Model

1.2. Stages in Waterfall Model:

- The stages of the Waterfall Model are performed sequentially as follows:

1. Requirement Gathering and Analysis: All possible system requirements to be developed are captured in this phase and documented in the requirements specification document.

2. System Design: The required specifications from "Requirement Gathering and Analysis" phase are studied in this phase and the system design is prepared. This system design helps to define system and hardware requirements and helps define the overall system architecture.

3. Implementation: With input from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each Unit is developed and tested for its functionality, known as Unit Testing.

4. Integration and Testing: All units developed during "Implementation" phase are integrated into a system after testing each unit. After integration, the whole system is checked for any errors and failures.

5. Deployment: After completing functional and non-functional testing, the product is deployed in a customer environment or to the market.

6. Maintenance: There are some problems occurring in the client environment. To fix those problems, patches are released. In addition to enhancing the product, several better versions are released. Maintenance is performed to accommodate these changes in the customer environment.

2. V-Model:

2.1. Definition of V-Model:

- The V-model is an SDLC model where the execution of processes takes place in a sequential manner in a V-shape. It is also known as the Verification and Validation model.

- The V-Model is an extension of the Waterfall Model and is based on the association of the test phase for each respective development phase. That means for every phase in the development cycle, there is a testing phase directly involved. This is a highly-disciplined model and the next phase only begins after the completion of the previous one.

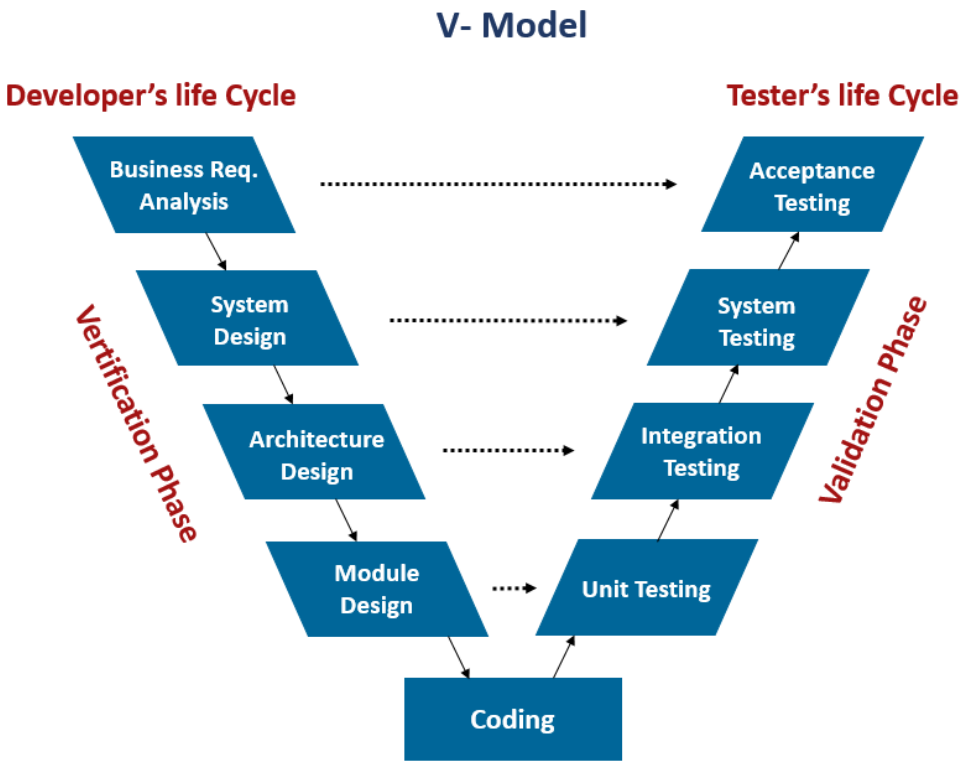

Figure 45: V-Model

2.2. Stages in V-Model:

- In V-Model, each testing phase is run in parallel with each corresponding development phase. So there are Verification phases on one side of the model and Validation phases on the other side. Coding phase links the two sides of the V-Model.

*** Verfication Phase** - The process of evaluating the product development stage to find out whether specific requirements are met.

1. Business Requirement Analysis:

+ This phase involves detailed communication with the customer to understand their exact expectations and requirements. This is a very important activity and needs to be managed well, as most clients are unsure of exactly what they need. Acceptance test design planning is done at this stage because business requirements can be used as input for acceptance testing.

2. System Design:

+ In this phase, system engineers analyze and interpret the business of the proposed system by studying the user requirements document.

3. Architecture Design:

+ Architectural specifications are understood and designed in this phase. The system design is further broken down into modules occupying different functions. This is also known as High Level Design (HLD).

+ Data transfer and communication between internal modules and with the outside world (other systems) are clearly understood and defined in this phase. With this information, integration tests can be designed and documented at this stage.

4. Module Design:

+ In this phase, the detailed internal design for all system modules is specified, known as Low Level Design (LLD). It is important that the design be compatible with other modules in the system architecture and other external systems.

+ Unit tests are an essential part of any development process and help eliminate bugs and errors at a very early stage as much as possible. These unit tests can be designed at this stage based on the internal modular designs.

*** Coding Phase:** After the design, the coding phase is started. Based on the requirements, a suitable programming language is decided. There are several guidelines and standards for coding. Before testing in the repository, the final build is optimized for better performance and the code will go through multiple code reviews to check the performance.

*** Validation Phase** - The process of evaluating software after the completion of the development phase to determine if the software meets customer expectations and requirements.

5. Unit Testing:

+ Unit tests designed in the module design phase are executed on the code during this validation phase. Unit testing is testing at the code level and helps to eliminate errors at an early stage, although it is not possible to detect all bugs with unit testing.

6. Intergration Testing:

+ This phase is closely associated with the architectural design phase. Integration testing is performed to check the coexistence and communication of internal modules in the system.

7. System Testing:

+ This phase is directly linked to the system design phase. System testing tests the entire system functionality and communication of the system being developed with external systems. Most software and hardware compatibility issues can be discovered during this system check.

8. Acceptance Testing:

+ This phase is linked to the business requirements analysis phase and involves testing the product in a user environment. Acceptance testing uncovers compatibility issues with other systems availabl e in the user environment. It also detects non-functional issues such as load and performance failures in real user environments.

3. Agile Model:

3.1. Definition of Agile Model:

- Agile Model is an approach that emphasizes iterative development. Instead of detailed long-term planning, Agile methods divide tasks into smaller iterations or parts.

- At the beginning of the development process, the project's scope and requirements are determined. The Agile approach involves clear and predefined plans for the number of iterations, their duration, and the scope of each iteration.

- In the Agile Model, each iteration is treated as a short time frame, typically lasting from one to four weeks. This division of the project into smaller parts serves several purposes. It helps minimize project risks and reduces the overall time required for project delivery.

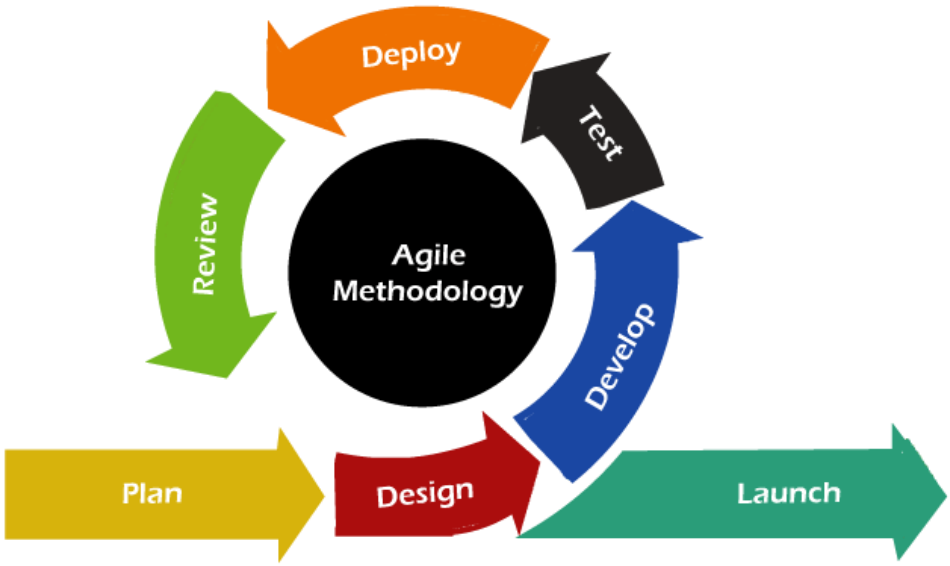

Figure 46: Agile Model

3.2. Stages in Agile Model:

- The Agile Model combines iterative and incremental approaches. It encompasses a series of steps designed to deliver software in a flexible and adaptive manner:

1. Requirement Gathering: The development team engages with the customer to gather project requirements, estimating the necessary time and effort. Technical and economic feasibility are assessed based on this information.

2. Designing Requirements: Using tools like user-flow diagrams and high-level UML diagrams, the team illustrates how new features integrate with existing software. This phase also involves wireframing and designing user interfaces.

3. Construction/ Iteration: Development team members begin working on the project with the goal of producing a functional product.

4. Testing/ Quality Assurance: Various testing phases are undertaken, including Unit Testing (for individual code blocks), Integration Testing (for combined software units), and System Testing (to ensure the software meets user requirements across scenarios).

5. Deployment: The completed project is deployed to end-users.

6. Feedback: This final step involves receiving feedback from customers, enabling the team to address any bugs or issues identified and enhance the product accordingly.

4. Spiral Model:

4.1. Definition of Spiral Model:

- Spiral Model is a risk-driven software development process model that combines the iterative development process model with elements of the Waterfall model. It is used by software engineers and is preferred for large, expensive and complex projects.

- In addition, it helps to apply software development elements of multiple process models to software project based on unique risk patterns ensuring efficient development process.

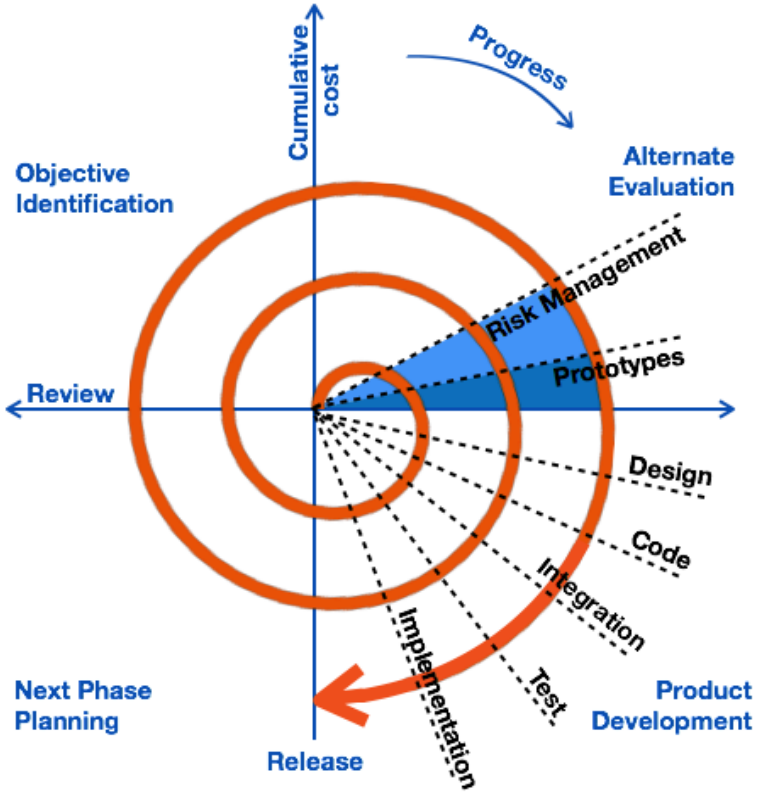

Figure 47: Spiral Model

4.2. Stages in Spiral Model:

- The spiral model has four stages. A software project that repeats these stages in iterations is called a spiral.

1. Identification:

- This phase begins with the basic spiral of gathering business requirements. In subsequent spirals as the product matures, defining system requirements, subsystem requirements, and unit requirements are all done during this phase.

- This phase also includes understanding the system requirements by means of ongoing communication between the customer and the system analyst. At the end of the cycle, the product is deployed in the defined market.

2. Design:

- The Design phase begins with conceptual design in the base spiral and involves architectural design, logical design of the modules, physical product design, and final design in the next spiral.

3. Construct:

- The Build phase refers to the actual production of the software product at every spiral. In the base vortex, when new product is devised and design is being developed, POC (Proof of Concept) is developed during this phase to get customer feedback.

- Then in subsequent spirals with greater clarity about the requirements and design details, a working model of the software called a build is created with a version number. These builds are sent to customers for feedback.

4. Evaluation and Risk Analysis:

- Risk analysis includes identifying, estimating, and monitoring technical feasibility and management risks, such as schedule slippage and cost overruns. After testing the build, at the end of the first iteration, the customer evaluates the software and gives feedback.

- Based on customer feedbacks, the software development process enters the next iteration and then follows a linear approach to implementing customer-suggested feedback. The spiral iterative process continues throughout the life of the software.

5. Conclude which SDLC Model will be used for the development:
- I decided to use Agile Model as the SDLC Model for our development project. gile is well-suited for handling evolving requirements and prioritizing customer feedback. It promotes iterative development, active customer collaboration, and a self-organizing team environment. By adopting Agile, we can effectively manage changes, incorporate customer input, and deliver a high-quality software solution that meets their needs.

C. Reference:

- GeeksforGeeks (2021) Top 7 UML Diagram Tools That You Can Consider. Available at: https://www.geeksforgeeks.org/top-7-uml-diagram-tools-that-you-can-consider/ (Accessed: 30 March 2024).
- Borowski, C. CXLead. (2024) 22 Best UI/UX Design Software Of 2024: Reviewed & Compared. Available at: https://thecxlead.com/tools/best-ui-ux-design-software/ (Accessed: 30 March 2024).
- GeeksforGeeks (2024) Web Technology. Available at: https://www.geeksforgeeks.org/web-technology/ (Accessed: 30 March 2024).
- TopDev (no date) SASS/SCSS là gì? Available at: https://topdev.vn/blog/sass-scss-la-gi/ (Accessed: 30 March 2024).

- Bose, S. BrowserStack. (2023) Top 5 CSS Frameworks for Developers and Designers. Available at: https://www.browserstack.com/guide/top-css-frameworks (Accessed: 30 March 2024).
- Raj, A. Almabetter. (2023) 7 Best Backend Languages for Web Development in 2024. Available at: https://www.almabetter.com/bytes/articles/top-backend-languages (Accessed: 30 March 2024).
- Bigelow, S. J. TechTarget. (2023) operating system (OS). Available at: https://www.techtarget.com/whatis/definition/operating-system-OS (Accessed: 30 March 2024).
- Digital Guide IONOS (2016) What is a web server? Available at: https://www.ionos.com/digitalguide/server/know-how/web-server-definition-background-software-tips/ (Accessed: 30 March 2024).
- Kirvan, P. , Bareny, N., and Gillis, A. S. TechTarget. (2024) Amazon Web Services (AWS). Available at: https://www.techtarget.com/searchaws/definition/Amazon-Web-Services (Accessed: 30 March 2024).
- Lutkevich, B. TechTarget. (2023) GitHub. Available at: https://www.techtarget.com/searchitoperations/definition/GitHub (Accessed: 30 March 2024).
- GeeksforGeeks (2022) Introduction to BitBucket. Available at: https://www.geeksforgeeks.org/introduction-to-bitbucket/ (Accessed: 30 March 2024).
- Kelley, K. Simplilearn. (2024) What is GitLab and How to Use It? Available at: https://www.simplilearn.com/tutorials/git-tutorial/what-is-gitlab (Accessed: 30 March 2024).
- Tutorialspoint (no date) SDLC - Quick Guide. Available at: https://www.tutorialspoint.com/sdlc/sdlc_quick_guide.htm (Accessed: 30 March 2024).
- GeeksforGeeks (2024) Agile Development Models – Software Engineering. Available at: https://www.geeksforgeeks.org/software-engineering-agile-development-models/ (Accessed: 30 March 2024).