ASSIGNMENT 1 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 30: Application Development		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Nguyen Ngoc Duy	**Student ID**	BH00157
Class	IT0501	**Assessor name**	Nguyen Thanh Trieu
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	Duy

Grading grid

P1	P2	P3	M1	M2	D1

❒ **Summative Feedback:** ❒ **Resubmission Feedback:**		
Grade:	**Assessor Signature:**	**Date:**
Lecturer Signature:		

Contents

Table of Figures

A. Introduction

Concurrently with the rapid progress of technology. Due to its many benefits, which include high levels of interactivity and quick data storage and retrieval, web applications are becoming more and more popular. Combine many tasks to make features, administration, and monitoring easier. Increase brand impact, enhance customer satisfaction, and save maintenance expenses. FPT Company is still going in that direction. The intention is to offer an online platform that can manage the "Training" activities for the business's internal training initiative, ensuring that all FPT Corporation workers are immersed in a continuous learning environment. The system is utilized by the human resources division. Three roles are involved in this system: administrator, training staff, and trainer. The system's functions include subject assignment for the course, lecturer assignment for the course, lecturer management, course catalog management, topic management, course management, and student account management. topics, choose the course's instructors.

B. Content

I. Research the use of software development tools and techniques and identify any that have been selected for the development of this application

1. Design tool

a. UML definition:

Unified Modeling Language, or UML for short, is a standardized modeling language made up of an integrated collection of diagrams that was created to assist business and non-software developers as well as system and software developers in defining, visualizing, building, and documenting the artifacts of software systems. The UML is an assembly of top engineering techniques that have been successfully used to the modeling of complicated and sizable systems. Creating object-oriented software and the software development process both heavily rely on the UML. The UML primarily expresses software project design using graphical notations. Project teams may communicate more effectively, investigate alternative ideas, and validate the software's architectural design when they use the UML.

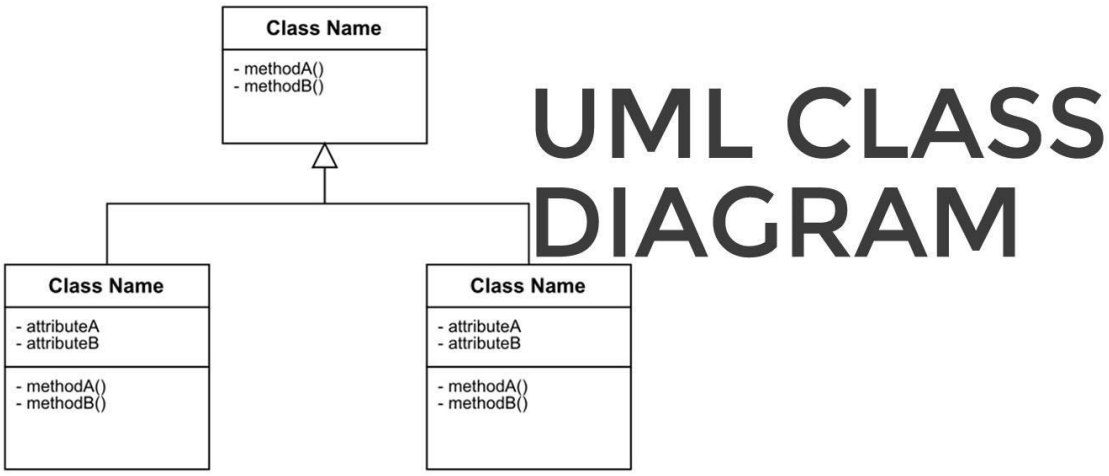

Figure 1: UML diagram

b. Characteristics of UML

The UML has the following features:

- It is a generalized modeling language.
- It is distinct from other programming languages like C++, Python, etc.
- It is interrelated to object-oriented analysis and design.
- It is used to visualize the workflow of the system.
- It is a pictorial language, used to generate powerful modeling artifacts.

c. Example using UML

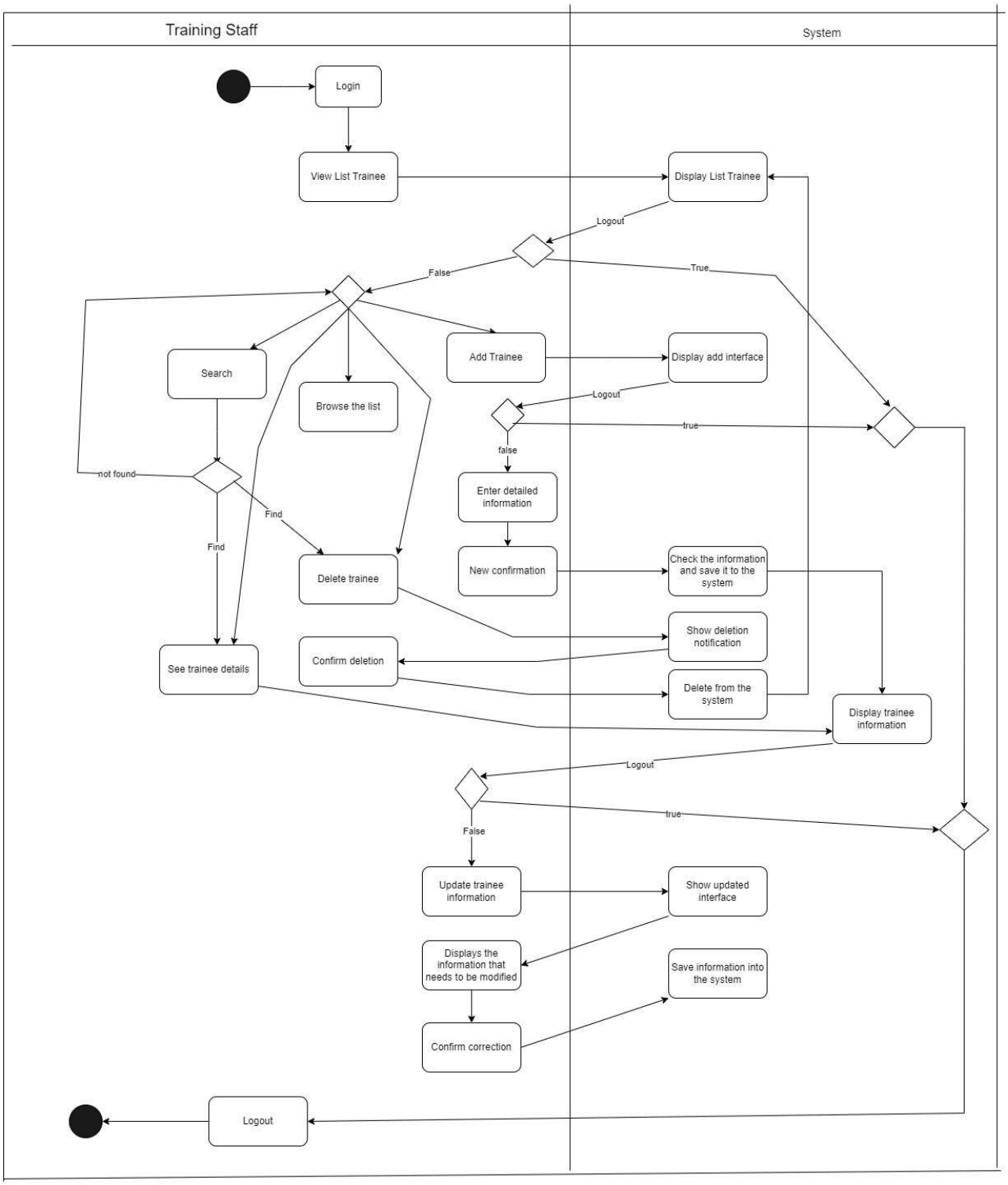

Figure 2: Example of UML

2. Chosen design tool

My recommendations for the FPT learning system will be based on a list of reputable cloud service providers. To address the topic of why select a specific service provider, I shall contrast it with other suppliers.

a. Draw.io

A proprietary program called Draw.io is used to create charts and diagrams. You can design a custom layout or use the software's automatic layout tool. They offer hundreds of visual elements and a wide variety of shapes to create a unique diagram or chart. The drag-and-drop functionality facilitates the creation of visually appealing charts and diagrams.

Figure 3: Draw.io

PROS:

- The first is rather obvious: it's free!
- It has an easy-to-use interface—which is user-friendly for beginners.
- It can be integrated with GitHub and GitLab repositories in addition to Google Drive, Dropbox, and OneDrive.

- You may quickly and easily construct high-quality diagrams by using a collection of premade forms.
- It lets you organize various forms.
- It allows room for collaboration.
- This app is hosted on the internet. It does not require downloading or installation on your phone or computer.

CONS:

- It is inferior to other online tools and continues to lack sophisticated functionality.
- It can occasionally be a little sluggish, especially while using a browser. may experience a few delays.
- These latencies and sporadic bugs may cause you to lose progress on part of your work and prevent it from being stored in real time.
- The diagrams might be difficult to export.
- The interface lacks support for cutting-edge design elements like operating in dark mode and has an antiquated, retro appearance.
- There are no keyboard shortcuts available to increase productivity.

b. Lucid chart

The recommended intelligent diagramming tool is Lucidchart, which enables teams to more quickly define complicated issues, come to consensus on concepts, and create the future. Signing up is free, and using it from anywhere is simple. Teams may operate asynchronously without losing time when they have access to real-time changes. Processes, systems, and organizational structure can be quickly mapped out and visualized to put everyone, literally, on the same page! After that, execute your ambitious strategies while maintaining attention on your corporate objectives. To maximize your tech stack, Lucidchart also interfaces with your preferred programs, like Microsoft, Atlassian, Google, and more.

Figure 4: Lucid Chart

PROS:

- **Ease of Use**: The user interface of Lucidchart is quite straightforward and pleasant, making it simple for both novice and expert users to build flowcharts and diagrams.
- **Collaboration**: Multiple users may collaborate in real-time while working on diagrams with Lucidchart. This functionality is very helpful for project teams.
- **Integration and Compatibility**: It works nicely with a variety of other widely-liked productivity programs, including Microsoft Office, Slack, Google Workspace, and others. It is simple to share and export Lucidchart files in a number of formats.
- **Wide Range of Templates**: By providing a vast collection of templates for various diagram types, Lucidchart helps users save time and effort when designing diagrams from scratch.
- **Cloud-Based and Accessible Anywhere**: Because Lucidchart is a cloud-based application, users may access their diagrams from any location with an internet connection. It works with a variety of devices.
- **Real-Time Updates**: Because it's a cloud-based solution, upgrades and enhancements happen automatically—manual installations are not necessary.
- **Security and Privacy**: To safeguard data, Lucidchart implements security protocols and provides choices for safe sharing and access management.

CONS:

- **Pricing Structure:** Some users may find Lucidchart's cost to be a drawback, particularly if they are on a tight budget or simply want the most basic diagramming tools.
- **Limited Free Version**: Because of its feature and storage limits, Lucidchart's free edition is not as ideal for users with sophisticated diagramming requirements.
- **Learning Curve for Advanced Features**: Despite Lucidchart's ease of use, considerable effort and training may be necessary to become proficient with its sophisticated capabilities and customization possibilities.
- **Internet Dependency**: Due to its cloud-based nature, Lucidchart is mostly dependent on internet access. When a user is offline, accessing and altering diagrams may provide difficulties.
- **Performance Issues**: Certain users have experienced sporadic latency or performance problems while collaborating with numerous people at once or working on intricate diagrams.
- **Limited Customization for Some Elements**: Despite Lucidchart's versatility, there could be restrictions on how certain diagram parts or shapes can be altered to suit certain design tastes.
- **Data Privacy Concerns**: Users may be concerned about data privacy and the storage of their schematics on external servers, as with any cloud-based application.

3. Development tools and techniques

A formal language is a collection of character sequences that may be used to generate different machine code outputs in software development. Programming languages are a subclass of computer languages that are used by programmers to build algorithms in the field of computer programming. In the parts that come next, we'll look at some popular programming languages.

a. C#

Microsoft created C#, a strongly-typed, compiled, object-oriented programming language. Since its introduction in the early 2000s, it has developed into a crucial part of the Microsoft.NET framework. C# is a robust, effective, and developer-friendly programming language that blends the ideas of C and C++ with elements from other languages like Java.

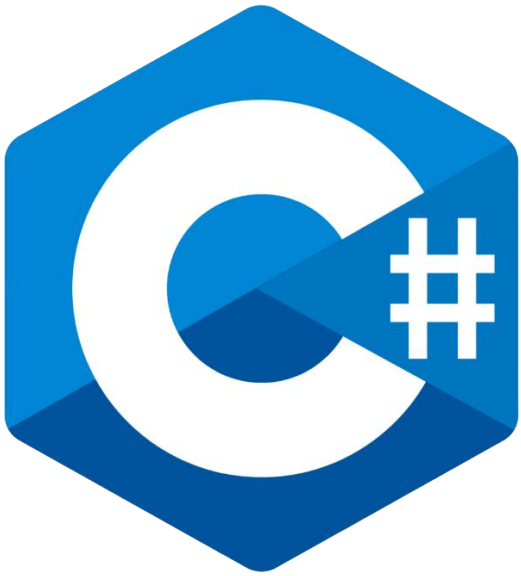

Figure 5: C#

PROS:

- **Powerful and Versatile:** C# is a potent programming language that can be used to create a variety of applications, including games, business software, mobile apps, and desktop and online applications.
- **Strongly Typed:** Because C# is a strongly typed language, it offers type safety and aids in identifying type-related mistakes during compilation, improving the robustness and dependability of the code.
- **Object-Oriented Programming (OOP) Paradigm:** The object-oriented programming paradigm, on which C# is built, encourages modular and well-organized code, code reuse, and simpler maintenance.
- **Rich Standard Library (.NET Framework):** The.NET framework, of which C# is a component, offers a sizable library of pre-built elements and functions, cutting down on development time and effort.
- **Language Interoperability:** Within the.NET environment, C# facilitates language interoperability, enabling developers to easily use components created in other.NET languages.
- **Asynchronous Programming:** The asynchronous programming feature in C# makes it simple to develop effective, non-blocking code, which is essential for applications that need to run concurrently.
- **Memory Management (Garbage Collection):** Automatic garbage collection makes memory management easier and lowers the chance of memory leaks in C#.
- **Developer-Friendly IDEs:** Robust Integrated Development Environments (IDEs) for C#, such as Visual Studio, boost productivity with built-in tools, IntelliSense, and code debugging.
- **Community and Ecosystem:** With a sizable and vibrant developer community, C# offers tools, resources, and open-source libraries to help programmers with their projects.
- **Cross-Platform Development:** Thanks to developments like.NET Core and.NET 5+, C# developers may create and run programs on a variety of operating systems, such as Windows, macOS, Linux, and others.

CONS:

- **Windows-Centric Focus:** C# has always been strongly linked to Windows and the Microsoft world. Despite efforts to enhance cross-platform compatibility, concerns of platform reliance may arise due to its roots.
- **Learning Curve:** The many capabilities and ideas of C# might present a challenging learning curve for novices or people who are not familiar with programming.
- **Limited Mobile Ecosystem:** Although Xamarin allows C# to be used for mobile app development, other languages and frameworks that are more popular in the mobile environment include Java/Kotlin for Android and Swift/Objective-C for iOS.
- **Performance Overhead:** Compared to lower-level languages, C# has high speed, although its controlled environment and automated garbage collection may cause a small performance penalty.
- **Dependency on .NET Framework:** Because of C#'s heavy reliance on the.NET framework, deploying programs may provide dependencies and versioning issues.

- **Less Low-Level Control:** In comparison to languages like C or C++, which may be required for certain performance optimizations or system-level programming, C# may provide less low-level control due to its high-level nature.
- **Proprietary IDE:** Even though Visual Studio is a strong IDE, some developers or organizations may be concerned about its proprietary nature and the need for a paid subscription in order to access all of its features.

b. JAVA

Java is an object-oriented, platform-neutral high-level programming language. It was created in 1995 at Sun Microsystems by Mike Sheridan and James Gosling, which was eventually purchased by Oracle Corporation. Java is a flexible language that is extensively used for producing a range of applications, including online, mobile, desktop, and business software. It was created to be straightforward, portable, safe, and robust.

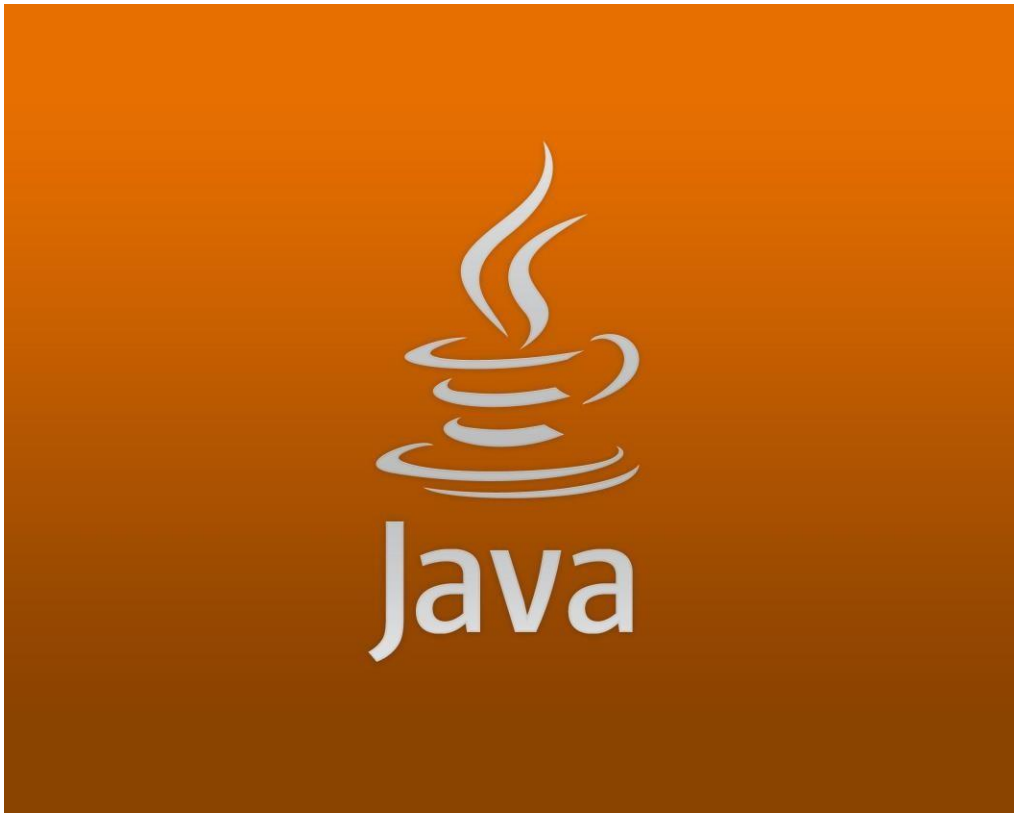

Figure 6: JAVA

PROS:

- **Versatility and Flexibility:** Java is a flexible programming language that has many uses, such as scientific applications, corporate software, mobile apps, web development, and more.
- **Platform Independence:** Platform independence is provided by Java's "write once, run anywhere" (WORA) concept, which enables Java programs to execute on any device that has a Java Virtual Machine (JVM).
- **Rich Ecosystem and Libraries:** Because of Java's extensive ecosystem and plenty of libraries, frameworks, and tools, development is quicker and more effective.
- **Strong Community Support:** There is a sizable and vibrant Java community that offers a wealth of tools, support, and frequent updates.
- **Object-Oriented Language:** The object-oriented programming (OOP) methodology used in Java encourages code modularity, reusability, and ease of maintenance.
- **Memory Management (Garbage Collection):** Java's automated garbage collection capability simplifies memory management for developers by lowering memory-related problems.
- **Multi-Threading Support:** Java has built-in threading functionality, which allows for concurrent execution and effective management of numerous jobs.
- **Security Features:** Java includes strong security features and built-in defenses against a range of security risks.
- **Scalability:** Java apps may readily grow to accommodate more users or work, which makes them appropriate for both small- and large-scale projects.
- **Consistency and Stability:** Java's mature development tools and well-defined specification provide uniformity and reliability in program development.

CONS:

- **Performance Overheads:** as it comes to very compute-intensive operations, Java may have performance overheads as compared to languages that are compiled to native code.
- **Learning Curve:** Java's vast feature set, intricate syntax, and OOP-related ideas make it more difficult for novices to master.
- **Memory Consumption:** Because Java runs on the JVM and automatically manages memory, Java programs may use more memory than those written in languages like C or C++.
- **Startup Time:** In some situations, the user experience may be negatively impacted by Java apps' slower starting times as compared to natively built languages.
- **Lack of Low-Level Features:** Because Java is a high-level language, it might not include all of the low-level functionality and direct memory manipulation capabilities found in C and C++.
- **Concurrency Challenges:** Even while Java allows for multi-threading, there may be difficult synchronization issues when controlling concurrent access to shared resources.

- **Limited Graphics and GUI:** Java's built-in GUI and visuals may not seem as strong or as advanced as those of other languages and frameworks.
- **Dependency on JVM:** Platform independence for Java depends on the JVM's accessibility and interoperability across a range of operating systems; modifications to the JVM's behavior may have an impact on applications.

4. Database server

a. MySQL server

Structured Query Language (SQL) is a tool used by MySQL, an open-source relational database management system (RDBMS), to manage and modify data in databases. It is renowned for being quick, dependable, scalable, and simple to use, making it one of the most popular database management systems in use worldwide. A prominent online development platform called the LAMP stack (Linux, Apache, MySQL, PHP/Python/Perl) includes MySQL, which is frequently used for web applications.

Figure 7: MySQL Server

PROS:

- **Open Source and Cost-Effective:** Since MySQL is an open-source database system, anybody can alter the system's source code to fit their own requirements. Because of this, it's an affordable option for companies and developers.
- **High Performance:** MySQL is renowned for its rapidity and great performance. It is appropriate for high-traffic websites and apps because it can efficiently manage a huge number of read and write operations.
- **Scalability:** Because of MySQL's exceptional scalability, organizations can manage increasing data volumes and user traffic without experiencing performance issues. It facilitates replication and clustering for both vertical and horizontal scalability.
- **Reliability and Durability:** Strong data integrity and reliability characteristics are offered by MySQL, such as transactional support, crash recovery techniques, and conformance with ACID (Atomicity, Consistency, Isolation, Durability) standards to guarantee consistent and dependable data.
- **Cross-Platform Compatibility:** Because MySQL is compatible with a wide range of operating systems—including Linux, Windows, macOS, and more—it is very adaptable and suitable in a variety of settings.
- **Community and Support:** There is a sizable and vibrant community of MySQL developers, administrators, and users that offer assistance, exchange expertise, and work to enhance the program. Furthermore, Oracle and other vendors give commercial support choices.
- **Security Features:** Strong security features including SSL support, user authentication, access control, and encryption are all provided by MySQL to guarantee data privacy and guard against illegal access.

CONS:

- **Complexity of Configuration:** Setting up MySQL may be difficult, particularly for people who don't have much expertise with database management. Setting up intricate settings and maximizing performance might call for expert understanding.
- **Lack of Some Features:** MySQL might not have all the sophisticated features and functions of some other database systems, however with newer releases and community additions, this difference has been closing.
- **Storage Engine Limitations:** Each MySQL storage engine has a unique set of advantages and disadvantages. It might be challenging for some users to select the right storage engine depending on their unique needs.
- **Limited Full-Text Searching:** Compared to some other databases, MySQL's full-text searching features are thought to be less sophisticated, therefore they might not be sufficient for customers who want more sophisticated full-text search capabilities.

- **Ownership and Licensing Concerns:** Oracle Corporation, the company that owns MySQL, has expressed worries about the software's future development, direction, and any changes to the license or maintenance arrangements.
- **Transaction Locking:** Performance problems can occasionally arise from the locking techniques MySQL uses for transactions, particularly in situations when there are a lot of concurrent transactions.
- **Suboptimal Performance for Complex Queries:** Although MySQL is typically quick and effective, if a large query with several joins and detailed subqueries is not optimized properly, performance may suffer.

b. SQL server

Large amounts of structured data may be managed and stored using SQL Server, a relational database management system (RDBMS) created by Microsoft. It offers a scalable and safe framework for organizing, storing, and retrieving data for a range of applications. SQL Server facilitates efficient creation, updating, and retrieval of data by utilizing the SQL (Structured Query Language) to communicate with the database.

Figure 8: SQL server

PROS:

- **Robust and Scalable:** Because of its scalability and resilience, SQL Server is a popular choice for managing big, complicated databases—even in enterprise-level applications.
- **Integration Services (SSIS):** Data management and processing are made simpler by SQL Server Integration Services (SSIS), which offers strong capabilities for data integration, data transformation, and data warehousing.
- **Reliable Performance:** Because SQL Server is performance-optimized, it offers capabilities like parallel processing, indexing, and query optimization to guarantee quick and effective data retrieval and query execution.
- **Comprehensive Business Intelligence (BI) Tools:** Businesses may efficiently analyze and display data with the help of SQL Server's suite of business intelligence (BI) tools, which include SQL Server Analysis Services (SSAS) and SQL Server Reporting Services (SSRS).
- **High Availability and Disaster Recovery:** AlwaysOn Availability Groups and database mirroring are two capabilities that SQL Server provides to provide high availability and disaster recovery, reducing downtime and data loss.
- **Security Features:** Strong security features are offered by SQL Server, such as role-based security, encryption, and auditing, to safeguard confidential information and adhere to security laws.
- **Ease of Management:** SQL Server Management Studio (SSMS) provides database managers with an easy-to-use interface that makes managing, monitoring, and maintaining databases simple.

CONS:

- **Cost:** For SQL Server, maintenance and licensing fees can be substantial, especially when compared to enterprise versions and other capabilities.
- **Resource Intensive:** Because SQL Server may be resource-intensive, particularly when it comes to memory and CPU utilization, it may need a substantial amount of hardware in order to operate at its best.
- **Vendor Lock-in:** Because SQL Server is a Microsoft proprietary software, there may be vendor lock-in and less freedom to select other options.
- **Complexity for Beginners:** Because of its many features, intricate setups, and wide range of capabilities, SQL Server may be difficult for novices to understand.
- **Limited Cross-Platform Support:** Even while SQL Server's cross-platform compatibility has improved, its Windows-only architecture has historically limited its deployment choices on other systems.

- **Scalability Challenges:** Despite the excellent scalability of SQL Server, scaling out (across numerous servers) might be more expensive and complicated than with alternative databases that are intended for horizontal scaling.
- **Version Compatibility:** When transferring databases or applications between SQL Server versions, compatibility problems may occur, necessitating meticulous preparation and testing.

5. Software models

a. Waterfall model

The first Process Model to be introduced was the Waterfall Model. Another name for it is a sequential-linear life cycle model. It is quite easy to use and comprehend. There is no phase overlap in a waterfall model; each step must be finished before the next can start. The first software development life cycle (SDLC) technique was the waterfall model.

The software development process is shown in the waterfall model as a sequential, linear flow. This implies that a development process phase may only start after the one before it is finished. The stages in this waterfall model don't cross over.

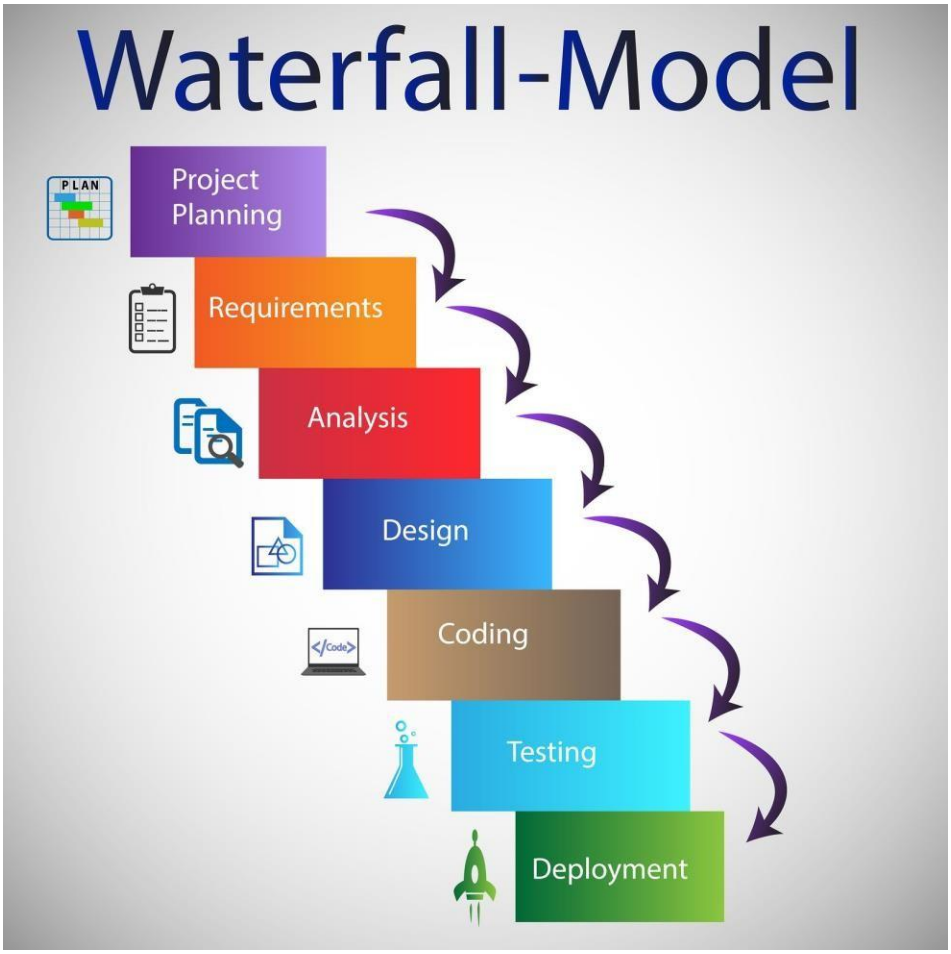

Figure 9: Waterfall Model

The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis:** During this stage, every potential need for the system that has to be constructed is gathered and recorded in a requirement specification document.
- **System Design:** This phase involves studying the need specifications from the previous phase and preparing the system design. This system design aids in determining the overall system architecture as well as the hardware and system requirements.
- **Implementation:** The system is first developed as units, or discrete programs, using inputs from the system design. These units are then merged in the subsequent phase. Unit testing is the process of developing and testing each unit to ensure it functions as intended.
- **Integration and Testing:** After each unit is tested, all of the units created during the implementation phase are combined into a single system. The complete system is checked for errors and malfunctions after integration.
- **Deployment of system:** After completing both functional and non-functional testing, the product is either launched into the market or deployed in the customer's environment.
- **Maintenance:** In the client setting, some problems might arise. Patches are published to address certain problems. Better versions of the product are also launched in an effort to improve it. To bring about these modifications in the client environment, maintenance is carried out.

b. Spiral model

The waterfall model's methodical, regulated elements are combined with the concept of iterative development in the spiral model. This spiral approach, which places a strong focus on risk analysis, combines the sequential linear development model, or the waterfall model, with the iterative development process model. With each round around the spiral, it permits incremental product launches or incremental improvement.

Figure 10: Spiral model

The sequential phases in Spiral model are:

- **Planning (Identification, Objective Setting):** The project's goals, options, and limitations are determined at this first stage. The project's objectives and restrictions are established in consultation with the main stakeholders, and an initial risk assessment is carried out.
- **Risk Analysis (Risk Assessment):** Potential risks and uncertainties are located and examined at this phase. Evaluating the likelihood and effect of hazards that have been identified is part of risk assessment. Additionally, methods for reducing and managing these risks are created.
- **Engineering (Development and Testing):** The real software is created during the development and testing phase using the requirements and analysis from earlier stages. Multiple iterations or sub-phases, such as design, implementation, integration, and testing, may be included.
- **Evaluation (Customer Evaluation, Review):** The client or end users examine the project's progress throughout this phase. After gathering feedback, any changes that should be made to the program or development process are determined. This stage aids in obtaining information on the functionality and performance of the product.

c. Agile model

The agile software development life cycle (SDLC) paradigm combines incremental and iterative process models, emphasizing process flexibility and customer satisfaction through quick delivery of functional software. Using agile methods, the product is built in tiny, incremental increments. Iterations of these builds are supplied. Usually, an iteration lasts between one and three weeks. In each iteration, cross-functional teams collaborate concurrently on many projects in areas including as

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing
- Acceptance Testing.

The client and other key stakeholders are shown a functional product at the conclusion of the iteration.

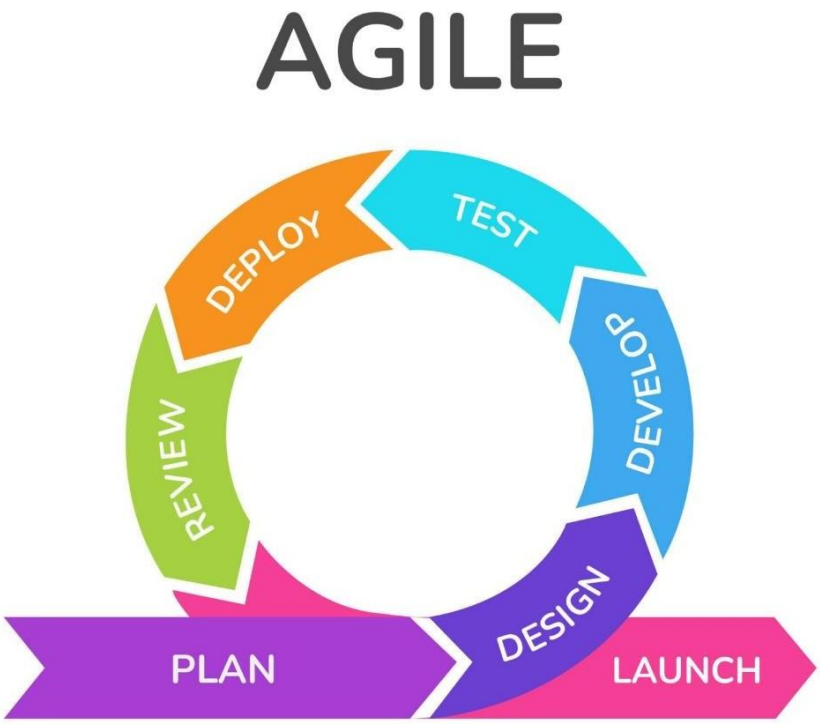

Figure 11: Agile Model

The sequential phases in Agile model are:

- **Planning**: Determine the project's needs and functions.
- **Design**: Describes the overall system or application architecture.
- **Development**: The specific tasks specified in the planning are started by the development team.
- **Testing and Review**: Test and quality control the functionality to make sure it functions as intended and does not lead to mistakes.
- **Deployment**: Release the finished product or feature to the public or market. Test the product and ensure that the final integration functions as planned.
- **User or Customer Feedback**: Get feedback from a consumer or end user by offering them a feature or product. This ensures that the products meet actual demands and aids in determining customer satisfaction.

C. Conclusion

After completing this report, I have a comprehensive grasp of the concept creation and implementation process. This problem identification section, which is based on requirements received from users and systems, indicates any risks associated with the successful execution of the application. I utilized an example or revised Software Requirements Specification (SRS) to do my assignment. I looked at the procedures and equipment utilized in software development. I've looked at a number of techniques and technologies in my spare time to see whether they would be useful in my circumstance. Score me on how effectively I explained how to select units based on characteristics including functionality, scalability, usability, and compliance with the project requirements, as well as how I examined different unit kinds. Using the tools we chose in the previous step, we now proceed to develop the strategy for our solution. This design diagram is based upon the specifications stated in the previous section. I make use of the chosen tools to provide a graphic depiction of the components, interactions, and architecture of our program. This diagram acts as a template for the development stage and provides a succinct overview of the flow structure and solution.

D. References

[1]. Sheldon, R. (2022) What is the C# object-oriented programming language? – TechTarget definition, WhatIs.com. Available at: https://www.techtarget.com/whatis/definition/C-Sharp (Accessed: 21 October 2023).

[2]. Fran (2023) What are HTML and CSS used for?: The Basics of Web Code, FutureLearn. Available at: https://www.futurelearn.com/info/blog/what-are-html-css-basics-of-coding (Accessed: 21 October 2023).

[3]. (No date) What is unified modeling language (UML)? Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/ (Accessed: 21 October 2023).

[4]. David Taylor and Taylor, D. (2023) Google cloud vs AWS: Difference between AWS and GCP, Guru99. Available at: https://www.guru99.com/google-cloud-vs-aws.html (Accessed: 21 October 2023).

[5]. Mleziva, M. (2023) 7 software development models in engineering you should know, Flexagon. Available at: https://flexagon.com/blog/7-software-development-models-you-should-know/ (Accessed: 21 October 2023).