**ASSIGNMENT 1 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | Unit 30: Application Development | | |
| **Submission date** | 06th/March/2024 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | PHAN HUU QUY | **Student ID** | BH00037 |
| **Class** | IT0502 | **Assessor name** |  |
| **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** | quy |

**Grading grid**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P1 | P2 | P3 | M1 | M2 | D1 |
|  |  |  |  |  |  |

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

Contents

[**A.** **Introduction** 4](#_Toc160662224)

[**B.** **Body** 4](#_Toc160662225)

[**I.** **Research the use of software development tools and techniques and identify any that have been selected for the development of this application.** 4](#_Toc160662226)

[**1)** **Integrated Development Environments (IDEs):** 4](#_Toc160662227)

[**2)** **ASP.NET** 6](#_Toc160662228)

[**3)** **C#** 7](#_Toc160662229)

[**4)** **SQL Server** 8](#_Toc160662230)

[**5)** **Design Tools** 10](#_Toc160662231)

[**5.1** **Diagrams.net** 10](#_Toc160662232)

[**5.2** **Use case diagram** 13](#_Toc160662233)

[**6)** **Version Control Systems** 14](#_Toc160662234)

[**C.** **Conclusion** 15](#_Toc160662235)

[Bibliography 16](#_Toc160662236)

[Figure 1 Visual studio 5](#_Toc160661732)

[Figure 2 C# 7](#_Toc160661733)

[Figure 3 SQL Server 9](#_Toc160661734)

[Figure 4 diagrams.net 11](#_Toc160661735)

[Figure 5 Database Diagram 11](#_Toc160661736)

[Figure 6 Use case diagram 13](#_Toc160661737)

1. **Introduction**

Embarking on the development journey of our application, we have strategically chosen a powerful arsenal of software development tools. Visual Studio Code anchors our IDE, offering a customizable workspace, Git integration, and cross-platform support. ASP.NET fuels our web applications, embracing the MVC pattern for robust development. C# stands as our primary language, emphasizing strong typing, object-oriented principles, and cross-platform versatility.

For database management, Microsoft SQL Server takes the lead, seamlessly integrating with .NET for efficient data processing and ensuring security compliance. Diagrams.net and UML diagrams streamline our design process, promoting collaboration and clarity.

GitHub, a Git-based hosting service, serves as our version control hub, facilitating collaboration through pull requests and code reviews. This curated toolset guarantees a streamlined and efficient development cycle, ensuring the delivery of a high-quality, feature-rich application.

1. **Body**
2. **Research the use of software development tools and techniques and identify any that have been selected for the development of this application.**
3. **Integrated Development Environments (IDEs):**

Integrated Development Environments (IDEs) play a crucial role in software development by providing a comprehensive workspace with multiple tools and supporting features. For this project, we intend to use the popular IDE, Visual Studio Code, developed by Microsoft.

Visual Studio Code is a free and open-source code editor developed by Microsoft. It supports most modern programming languages such as Java, Python, C#, C++, JavaScript, TypeScript, and many others. The main features include:

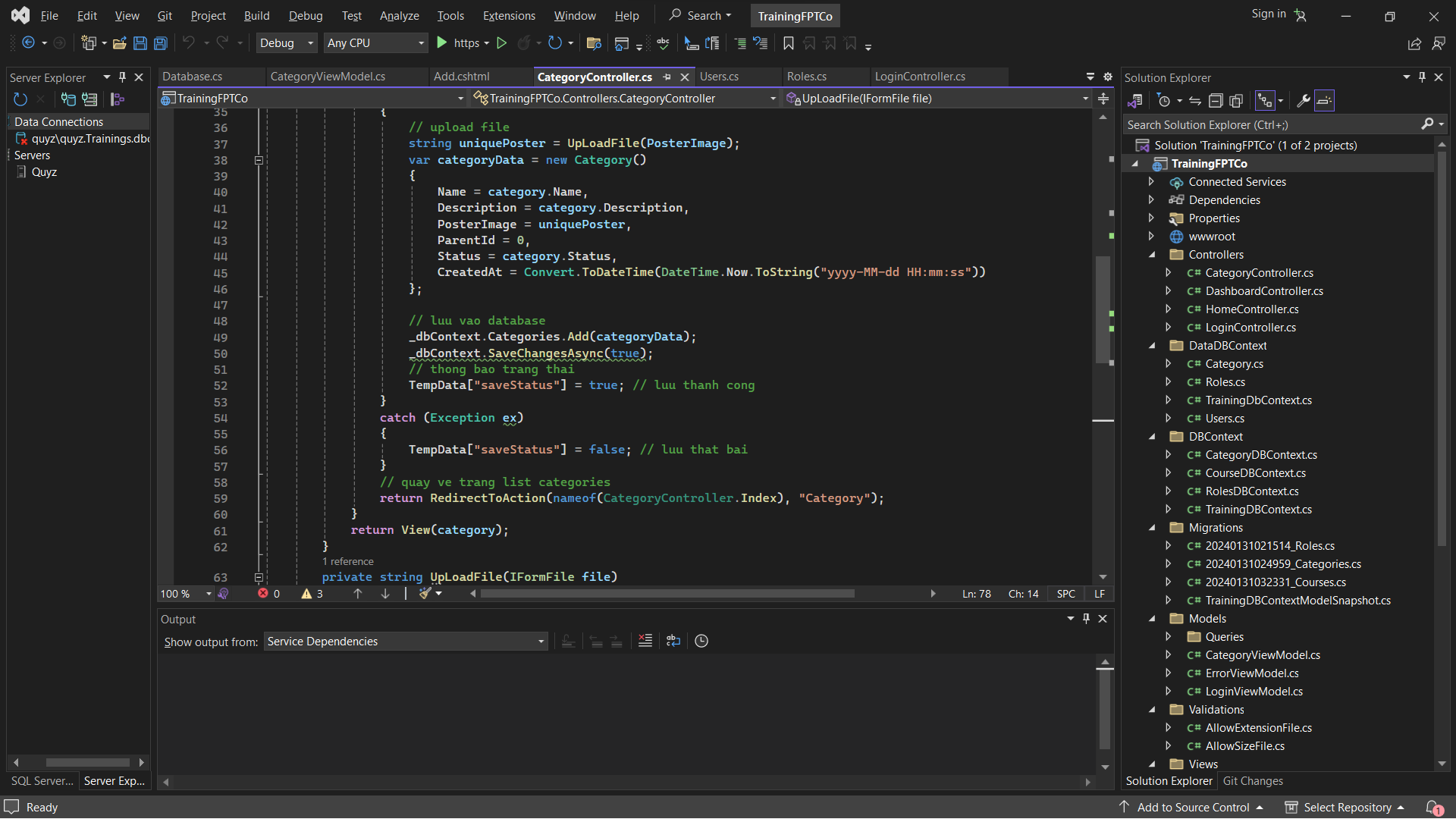


Figure 1 Visual studio

* Flexible and highly customizable interface: Visual Studio Code provides an intuitive and user-friendly interface, allowing developers to customize the layout, window arrangement, and keyboard shortcuts according to their preferences.
* Efficient programming tools: The IDE comes integrated with a compiler and debugger, supporting full programming concepts such as auto-complete, error highlighting, and syntax highlighting. It also supports static code analysis and type checking.
* Extensibility: Visual Studio Code has a rich ecosystem with thousands of extensions, allowing customization and enhancement of functionality for specific languages, tools, and frameworks.
* Git integration: The IDE is well-integrated with the distributed version control system Git, enabling developers to track changes, create branches, commit, and push code.
* Integrated Terminal panel: An integrated Terminal panel allows developers to execute command-line instructions and third-party tools directly within the IDE.
* Cross-platform support: Visual Studio Code is available for Windows, macOS, and Linux, ensuring a consistent development environment across multiple platforms.

By utilizing Visual Studio Code, our development team can leverage its powerful features and integrated workspace to write code, debug, test, and maintain the project efficiently. This will help increase productivity, streamline the development process, and ensure higher code quality.

Visual Studio Code's comprehensive toolset, extensibility, and seamless integration with various programming languages and technologies make it an excellent choice for our project's development environment. Its user-friendly interface and customization options will enhance the overall development experience for our team members. (Wikipedia, 2024)

1. **ASP.NET**

ASP.NET is a powerful and versatile web application framework developed by Microsoft, built on top of the .NET platform. It provides a comprehensive and feature-rich environment for building dynamic and scalable web applications and services using C# or Visual Basic programming languages.

One of the key strengths of ASP.NET is its support for the Model-View-Controller (MVC) architectural pattern, which promotes separation of concerns and facilitates the development of complex web applications. It also offers Web Forms, a more traditional approach for building event-driven web applications.

ASP.NET comes with built-in security features such as authentication, authorization, and data validation, ensuring that applications are secure from the ground up. It seamlessly integrates with Visual Studio, Microsoft's powerful Integrated Development Environment (IDE), providing developers with a rich set of tools and features for efficient coding, debugging, and deployment.

The framework promotes code reusability and extensibility through its support for user controls, custom controls, and integration with third-party components and libraries. With the introduction of .NET Core, ASP.NET applications can now run cross-platform, supporting Windows, macOS, and Linux operating systems.

ASP.NET applications are known for their scalability and high performance, leveraging features such as caching mechanisms, asynchronous processing, and optimized resource management to handle high traffic loads efficiently.

By leveraging ASP.NET for our web-based training management system, we can benefit from its robust architecture, rich feature set, and seamless integration with the .NET ecosystem, enabling us to develop a secure, scalable, and maintainable application. (Microsoft, n.d.)

1. **C#**

For this project, we have chosen C# as our primary programming language. C# is a modern, object-oriented language developed by Microsoft as part of the .NET framework. It offers several advantages that make it well-suited for developing robust and scalable applications, particularly for the web and enterprise domains.

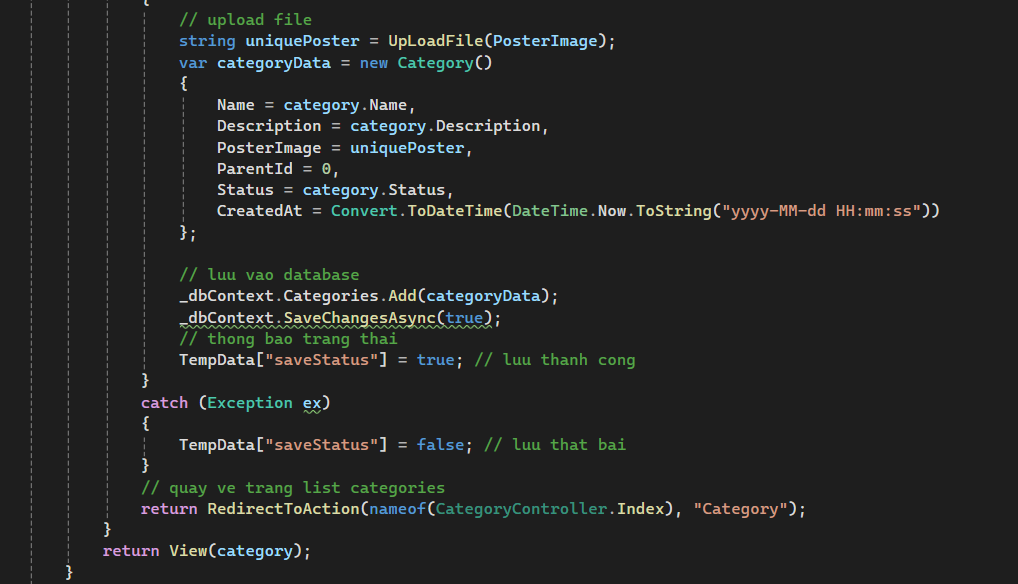


Figure 2 C#

* Strong Typing: C# is a strongly-typed language, which means that variables must be explicitly declared with their data types. This feature helps catch type-related errors during compile-time, improving code reliability and maintainability.
* Object-Oriented Programming (OOP): C# fully supports object-oriented programming principles such as encapsulation, inheritance, and polymorphism. This allows for better code organization, reusability, and modularity, which is essential for large-scale projects.
* Language Interoperability: C# seamlessly integrates with other .NET languages, enabling developers to leverage existing code libraries and components written in languages like Visual Basic or F#, promoting code reuse and collaboration.
* Robust Framework and Libraries: The .NET framework provides a vast collection of pre-built classes, libraries, and APIs for various tasks, such as database connectivity, networking, file I/O, and user interface development. This rich ecosystem accelerates development and promotes consistency across projects.
* Cross-Platform Capabilities: With the introduction of .NET Core, C# applications can now run on multiple platforms, including Windows, macOS, and Linux, making it a versatile choice for modern software development.
* Automatic Memory Management: C# incorporates automatic memory management through garbage collection, relieving developers from the burden of manual memory allocation and deallocation, reducing the likelihood of memory-related bugs.
* Advanced Language Features: C# continuously evolves, incorporating modern language features such as asynchronous programming, lambda expressions, LINQ (Language Integrated Query), and pattern matching, which enhance developer productivity and code readability.

By choosing C# as our primary programming language, we can leverage its powerful features, rich ecosystem, and cross-platform capabilities to develop a robust and scalable training management system. The language's strong typing, object-oriented nature, and extensive libraries will contribute to code quality, maintainability, and efficient development processes. (Wikipedia, 2024)

1. **SQL Server**

Regarding Database Management Systems (DBMS), tools like MySQL, SQL Server, PostgreSQL, or MongoDB are essential for efficiently storing and managing data within the application. For this project, we have chosen to utilize Microsoft SQL Server as our database management platform, with the following key reasons:

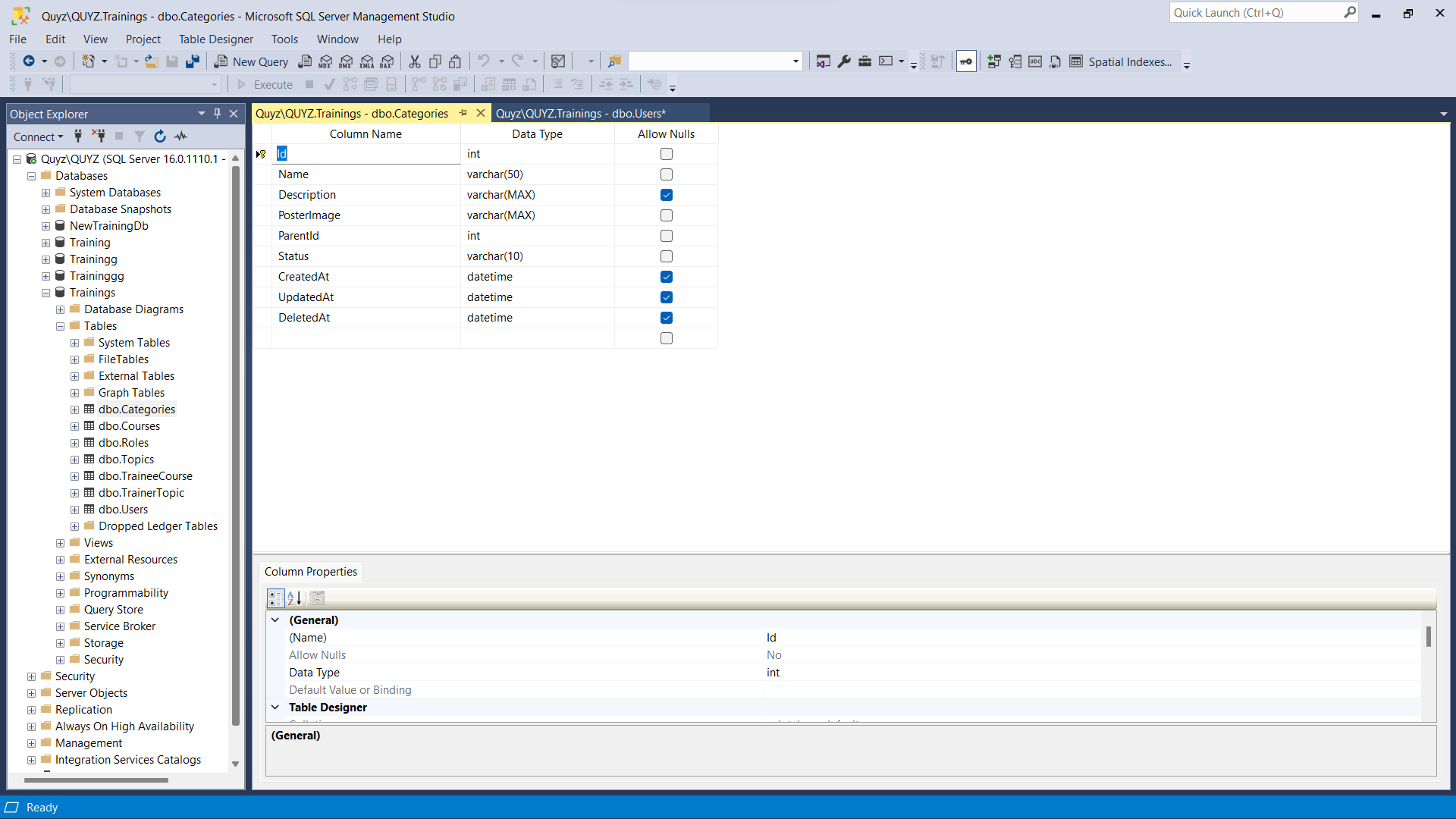


Figure 3 SQL Server

* **Tight Integration with .NET Framework**: SQL Server is developed by Microsoft and seamlessly integrates with the .NET Framework, the application development platform we have chosen to use. This simplifies the development process and enhances performance by allowing direct data access from .NET code.
* **Scalability and High Performance**: SQL Server is designed to efficiently handle complex data processing tasks and store large amounts of data. It offers high scalability, allowing our system to grow and expand according to future business requirements.
* **Robust Security Features**: SQL Server provides robust security features, including authentication, data encryption, access control, and auditing. This helps ensure data integrity, security, and compliance with regulations for sensitive data in our training management system.
* **Data Synchronization Support**: SQL Server offers tools and techniques for data synchronization, allowing us to easily integrate and synchronize data from various sources, if needed in the future.
* **Data Management and Analytics Tools**: SQL Server includes powerful data management, reporting, and data mining tools, enabling us to easily analyze and extract valuable insights from training data.
* **Rich Support and Resources**: With a large user community and abundant support resources, we can easily find support, guidance, and solutions for any SQL Server-related issues during development and deployment.

By selecting SQL Server as our database management platform, we ensure that our training management system will have the capability to store and process data efficiently, meeting the requirements for performance, security, and scalability in the future.

SQL Server's tight integration with the .NET ecosystem, robust feature set, and extensive support resources make it an ideal choice for managing the data needs of our application effectively and reliably. Its proven performance and scalability will allow our system to handle growing data volumes and user loads seamlessly. (Wikipedia, 2024)

1. **Design Tools**

UML, or Unified Modeling Language, is a standardized modeling language widely used in software engineering. It employs graphical notations to create visual representations of software-intensive systems, encompassing aspects such as structure, behavior, interactions, and architecture. The primary aim of UML is to enhance communication among software developers, analysts, designers, and stakeholders. By providing a shared language and notation, UML fosters understanding and collaboration across diverse stakeholders involved in the software development process.

1. **Diagrams.net**

Draw.io, or Diagrams.net, is a free and open-source online diagramming tool that allows users to create various types of diagrams, including flowcharts, organizational charts, process flows, UML models, entity-relationship diagrams (ERDs), and network diagrams. We have selected this tool for our project due to several compelling reasons:



Figure 4 diagrams.net

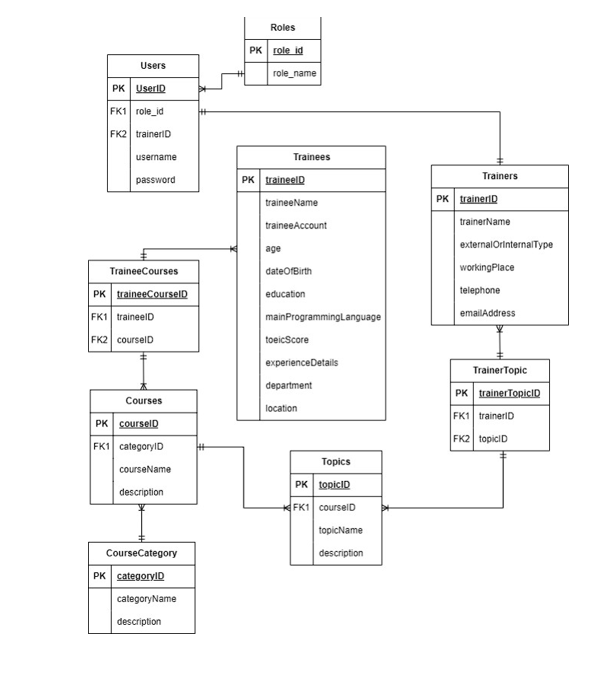


Figure 5 Database Diagram

* **Platform Independence and Accessibility**: Diagrams.net is a web-based application, making it accessible from any device with a compatible web browser. This cross-platform compatibility ensures that our team members can collaborate seamlessly, regardless of their operating systems or hardware configurations.
* **Intuitive and User-friendly Interface**: The tool's interface is designed to be intuitive and user-friendly, with a clean and modern layout. This reduces the learning curve for team members, allowing them to become proficient with the tool quickly and enhancing overall productivity.
* **Rich Diagramming Capabilities**: Diagrams.net offers a comprehensive set of features for creating various types of diagrams, including support for custom shapes, templates, and libraries. This versatility allows our team to effectively model and visualize different aspects of our software design, such as UML diagrams, database schemas, and architectural diagrams.
* **Integration and Collaboration**: Diagrams.net seamlessly integrates with popular cloud storage services like Google Drive and Dropbox, enabling efficient collaboration and real-time sharing of diagrams among team members. Additionally, it supports version control, allowing us to track changes and revert to previous versions if needed.
* **Export and Import Flexibility**: The tool supports a wide range of export and import formats, including PNG, SVG, JPEG, XML, and HTML. This flexibility ensures that our diagrams can be easily shared, integrated with other documentation, or imported into various development environments or presentation tools.
* **Open-source and Extensible**: As an open-source tool, Diagrams.net benefits from a vibrant community of contributors and developers. This community continuously enhances the tool's capabilities and provides support through forums and documentation. Additionally, the tool's extensible nature allows for the development of custom plugins and integrations to meet specific project requirements.

By incorporating Diagrams.net into our software development process, we can effectively visualize and communicate our design concepts, facilitate collaboration among team members, and ensure consistent and standardized documentation across the project. The tool's rich feature set, accessibility, and integration capabilities make it an invaluable asset for our design efforts, ultimately contributing to the successful delivery of our training management system. (Wikipedia, 2024)

1. **Use case diagram**

A use case diagram is a commonly employed visualization tool in UML (Unified Modeling Language) designed to showcase a system's functionality through the lens of its users. This diagram provides a visual representation of the interactions between users, referred to as actors, and the system being analyzed. Its main purpose is to capture and document system requirements by highlighting the different ways users engage with the system and by defining the system's scope.

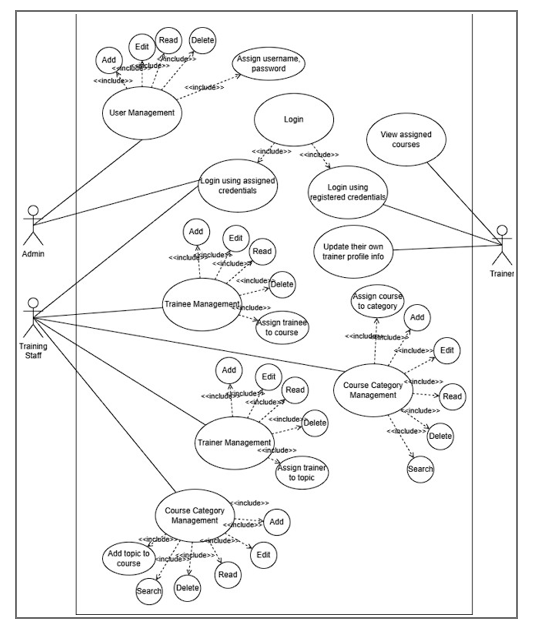
****

Figure 6 Use case diagram

In essence, a use case diagram serves as a blueprint for understanding how actors interact with the system to achieve specific goals. The actors, representing users or external entities, are depicted along with the various use cases, which represent specific functionalities or actions the system can perform. The connections, known as associations, between actors and use cases demonstrate the relationships and interactions within the system.

Overall, a use case diagram serves as a valuable communication tool during the early stages of software development, aiding in the identification and clarification of system requirements while offering a user-centric perspective on system functionality. (visual-paradigm, n.d.)

1. **Version Control Systems**

Version Control Systems like Git, SVN (Subversion), and Mercurial are essential tools that can help manage source code changes, facilitate collaboration among team members, and track a project's history over time. Of these options, GitHub was selected as the preferred version control system for this project.

GitHub is a Git-based hosting service that allows developers to manage and collaborate on projects from a central online platform. As an open source project, using GitHub provides transparency and allows anyone to view the source code repository. It streamlines the development workflow by enabling features like pull requests, issue tracking, wikis, and integrated code review functionality. Developers can fork repositories to propose and contribute changes without affecting the original project.

GitHub's large user base and extensive documentation make it easy for new contributors to get involved and learn how to use the version control system. The platform centralizes code hosting, issue tracking, code review, and documentation in one location. This consolidates workflow and facilitates better communication and coordination between team members. Developers can also integrate GitHub with continuous integration services for automated testing and deployment.

Additional benefits include unlimited private repositories for open source projects, two-factor authentication for enhanced security, and integration with various programming languages and tools through its API. GitHub is also freely available for open source and educational use, keeping costs low for students and academic projects. Its wide adoption in industry makes skills with GitHub valuable for future employment opportunities.

In summary, GitHub was selected as it provides a robust, full-featured version control system with collaboration tools and integration that streamline development. Its large ecosystem, free tier, and learnability make it suitable for managing the source code repository of this academic software project. GitHub's advantages over alternatives like GitLab or Bitbucket firmly establish it as the best choice. (HTG, 2016)

1. **Conclusion**

In conclusion, our meticulous selection of software development tools has laid the foundation for a cohesive and efficient development process. The synergy between Visual Studio Code, ASP.NET, C#, Microsoft SQL Server, Diagrams.net, UML diagrams, and GitHub ensures that our team operates seamlessly, from the initial stages of ideation to the final stages of version control and collaboration.

The chosen tools not only align with industry best practices but also address the specific needs of our project, emphasizing security, scalability, and code maintainability. Visual Studio Code's adaptability and integrated features enhance the development experience, while ASP.NET's robust framework provides a solid backbone for our web applications. C#'s versatility and object-oriented nature contribute to the development of scalable and maintainable code.

With Microsoft SQL Server, we guarantee efficient data management and security, crucial elements for a successful application. Diagrams.net and UML diagrams play a pivotal role in visualizing and communicating our design concepts, fostering collaboration and ensuring a shared understanding of the project's architecture.

GitHub's presence as our chosen version control system not only tracks the evolution of our codebase but also facilitates collaborative efforts through pull requests and integrated code reviews. This ensures a transparent and organized development lifecycle, enhancing the overall quality of our application.

In essence, our carefully curated toolkit reflects our commitment to delivering a high-quality and feature-rich training management system. Through the seamless integration of these tools, our development team is equipped to navigate the complexities of modern software development, ultimately leading to the successful realization of our project goals.

# Bibliography

HTG. (2016, September 21). *howtogeek*. Retrieved from What is github, and what is it used for?: https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/

*Microsoft*. (n.d.). Retrieved from What is ASP.NET?: .NET: https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet

*visual-paradigm*. (n.d.). Retrieved from What is use case diagram?: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/

*Wikipedia*. (2024, March 03). Retrieved from https://en.wikipedia.org/wiki/Visual\_Studio

*Wikipedia*. (2024, February 20). Retrieved from C sharp (programming language): https://en.wikipedia.org/wiki/C\_Sharp\_%28programming\_language%29#:~:text=C%23%20%28%2F%20%CB%8Csi%CB%90%20%CB%88%CA%83%C9%91%CB%90rp%20%2F%20see%20SHARP%29%20%5Bb%5D,%3A%E2%80%8A22%20object-oriented%20%28class%20-based%29%2C%20and%20component-oriented%20p

*Wikipedia*. (2024, February 26). Retrieved from Microsoft SQL server: https://en.wikipedia.org/wiki/Microsoft\_SQL\_Server

*Wikipedia*. (2024). Retrieved from Diagrams.net: https://en.wikipedia.org/wiki/Diagrams.net