# CS 255 System Design Document Template

## UML Diagrams

### UML Use Case Diagram

*A diagram of a driver pass

AI-generated content may be incorrect.*

### UML Activity Diagrams

*A diagram of a diagram

AI-generated content may be incorrect.*

*A diagram of a software company

AI-generated content may be incorrect.*

### UML Sequence Diagram

*A diagram of a diagram

AI-generated content may be incorrect.*

### UML Class Diagram

*A diagram of a software company

AI-generated content may be incorrect.*

## Technical Requirements

**Hardware Requirements**

* Client Devices: Students, instructors, and staff will require access to a reasonably modern device (desktop, tablet, smartphone, or laptop) with a reliable internet connection to support live updates and interactive sessions. Client devices should also have a recommended 4 GB of RAM to support all features.
* Server: The system will be cloud-based. To meet demands, the virtual machine should run on 4-8 CPU cores, 16-32 GB of RAM, and SSD storage (e.g., 500 GB) configured for automatic scaling and high availability.
* Backup hardware: Cloud based storage to ensure data is not lost or destroyed.
* Firewall Implementation: Keeps users safe by defending against external attackers.
* Storage: Adequate storage for both database and the website

**Software Requirements**

* Operating Systems (OS): The system will be browser based and should support modern OS platforms including Windows 10+, MacOS 12+, Android 11+, and IOS 15+
* Web Browsers: Users should use the most updated version of their current web browser to fully support all features. Web browsers could include Google Chrome, Microsoft Edge, Safari, and Firefox.
* SQL Server: Deploying an SQL server for the database using the most recent version to store user accounts, lessons, scheduling, and progress tracking
* Security Software: Utilizing up-to-date security software to protect user data which could include SSL/TLS certificates for secure HTTPS connections
* Web Server: Implementing a webserver such as Apache.
* Back-end Application Framework: Created with Python (Django)
* Frontend software stack: HTML5, JavaScript to support a responsive interface for both desktop and mobile.

**Tools and Infrastructure**

* Cloud Hosted: The system will be launched via a cloud service that supports auto-scaling, load balancing, and geographic replication (Ahmad & Andras, 2019)
* Network: Communication between clients and servers must be encrypted via HTTP.
* Backup Recovery: Daily incremental backups followed by a weekly full backup to keep information from being lost.
* User Authentication: The system will use RBAC so each user (Student, Instructor, and Staff) can only see what is made visible to them via their role. RBAC is a strong model at simplifying access control. (Gillis & Rosencrance, 2024)

**Scalability**

* Cloud-based Architecture: Supports scaling to accommodate future increased foot traffic without an increase in investments. Scalability of cloud services has been empirically shown to benefit performance and cost-efficiency (Al-Said & Andras, 2019)
* Modular code design allows admins to add or disable lesson packages without the need to fully redeploy the system.

References  
  
Gillis, A. S., & Rosencrance, L. (2024). *What is role-based access control (RBAC)?* Search Security; TechTarget. https://www.techtarget.com/searchsecurity/definition/role-based-access-control-RBAC?

‌

Ahmad, A. A.-S., & Andras, P. (2019). Scalability analysis comparisons of cloud-based software services. *Journal of Cloud Computing*, *8*(1). https://doi.org/10.1186/s13677-019-0134-y

‌