# CS 255 System Design Document Dobbs

## UML Diagrams

### UML Use Case Diagram

*A screenshot of a computer program

Description automatically generated*

UML Activity Diagrams

*A diagram of a register for courses

Description automatically generatedA diagram of a test

Description automatically generated*

### UML Sequence Diagram

*A diagram of a course

Description automatically generated*

### UML Class Diagram

*A diagram of a class

Description automatically generated*

## Technical Requirements

Hardware Requirements

Servers:

* Web Server: To host the web application and handle user requests.
* Database Server: To store user data, course information, test results, and appointment details.

Network Equipment:

* Routers and Switches: To manage and direct data traffic between servers and users.
* Firewall: To secure the network and protect against unauthorized access.

User Devices:

* Computers, tablets, and smartphones: To allow users (students, instructors, admins, and secretaries) to access the system.

Software Requirements

Operating Systems:

* Linux or Windows Server: For hosting the web and database servers.

Web Server Software:

* Apache or Nginx: To serve the web application.

Database Management System (DBMS):

* MySQL, PostgreSQL, or MongoDB: To manage and store data efficiently.

Application Software:

* Web Application Framework: Such as Django (Python), Ruby on Rails (Ruby), or Spring Boot (Java) for developing the web application.
* Frontend Framework: Such as React, Angular, or Vue.js for building the user interface.

Development Tools:

* Integrated Development Environment (IDE): Such as Visual Studio Code, IntelliJ IDEA, or PyCharm for writing code.
* Version Control System: Git and platforms like GitHub or GitLab for version control and collaboration.

Security Software:

* SSL/TLS Certificates: For securing data transmission over the internet.
* Antivirus and Anti-malware: For protecting servers and user devices.

Tools and Infrastructure

Development Tools:

* Lucidchart: For creating UML diagrams and designing the system architecture.
* JIRA or Trello: For project management and tracking tasks.

Deployment Tools:

* Docker: For containerizing the application and ensuring consistency across different environments.
* Kubernetes: For orchestrating and managing containerized applications.
* CI/CD Pipeline: Tools like Jenkins or GitHub Actions for continuous integration and deployment.

Cloud Services:

* AWS, Azure, or Google Cloud: For scalable and reliable hosting solutions. Services such as EC2 for virtual servers, S3 for storage, RDS for managed databases, and CloudFront for content delivery.

Backup and Recovery:

* Regular data backups using cloud storage solutions or on-premises backup systems.
* Disaster recovery plan to ensure quick restoration of services in case of failures.

Monitoring and Maintenance:

* Monitoring Tools: Such as Prometheus, Grafana, or New Relic for real-time monitoring and alerting.
* Log Management: Using tools like ELK Stack (Elasticsearch, Logstash, Kibana) for centralized logging and analysis.

Security Infrastructure:

* Role-Based Access Control (RBAC): To manage permissions and access levels for different users (students, instructors, admins, secretaries).
* Encryption: For securing sensitive data both at rest and in transit.
* Regular Security Audits: To identify and mitigate vulnerabilities in the system.