

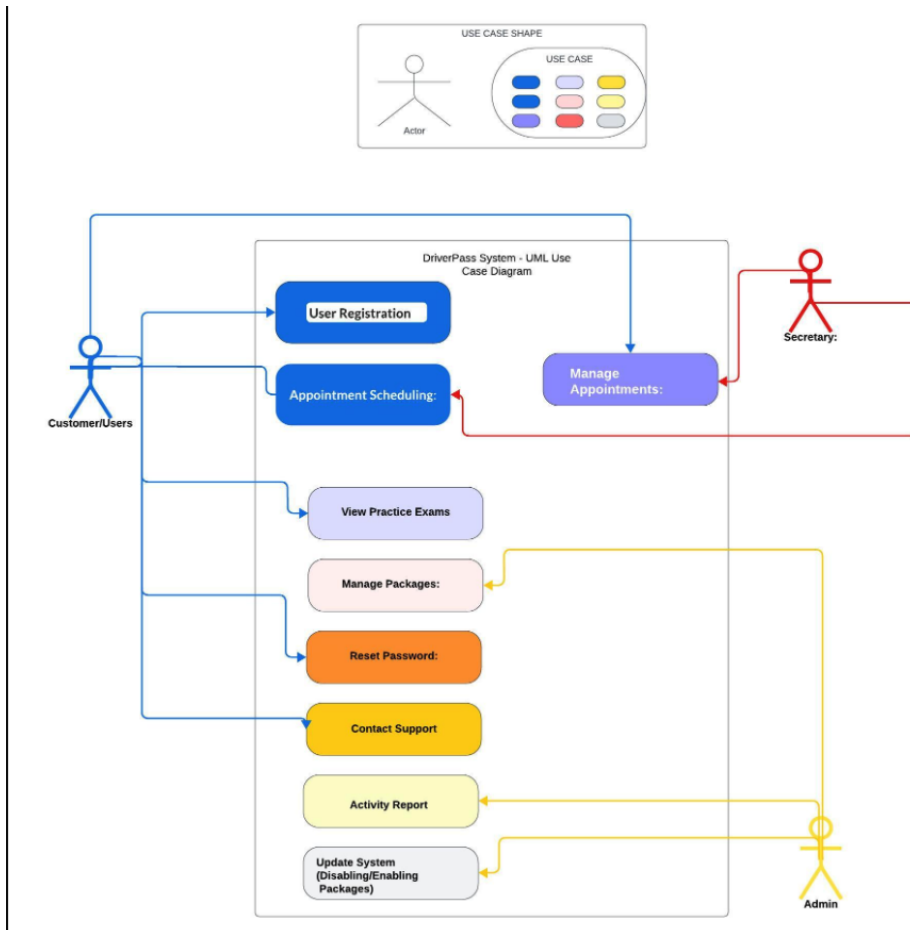
Jeury Santos

Prof. Federico Bermudez

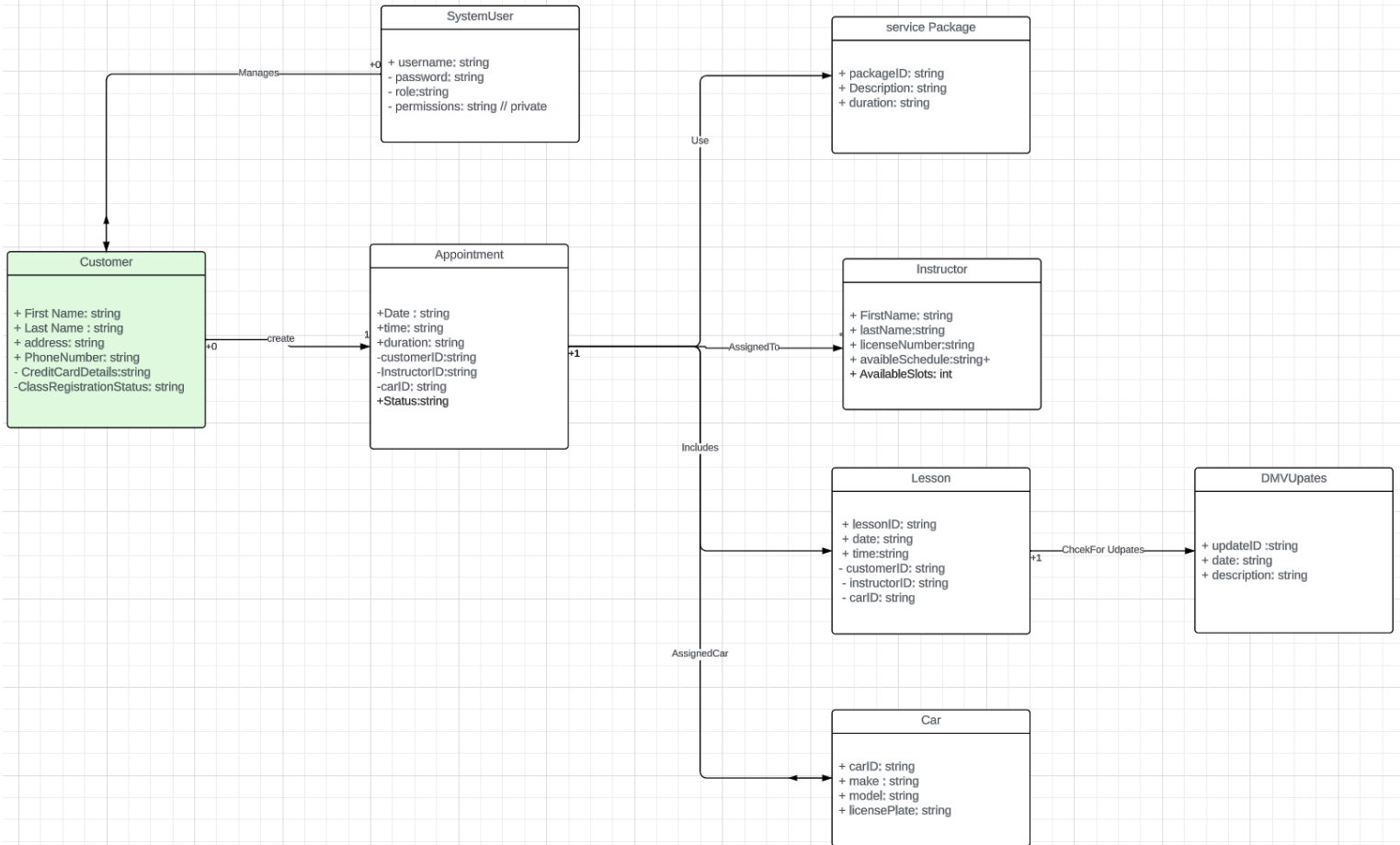
CS-255

Dec 10th , 2023

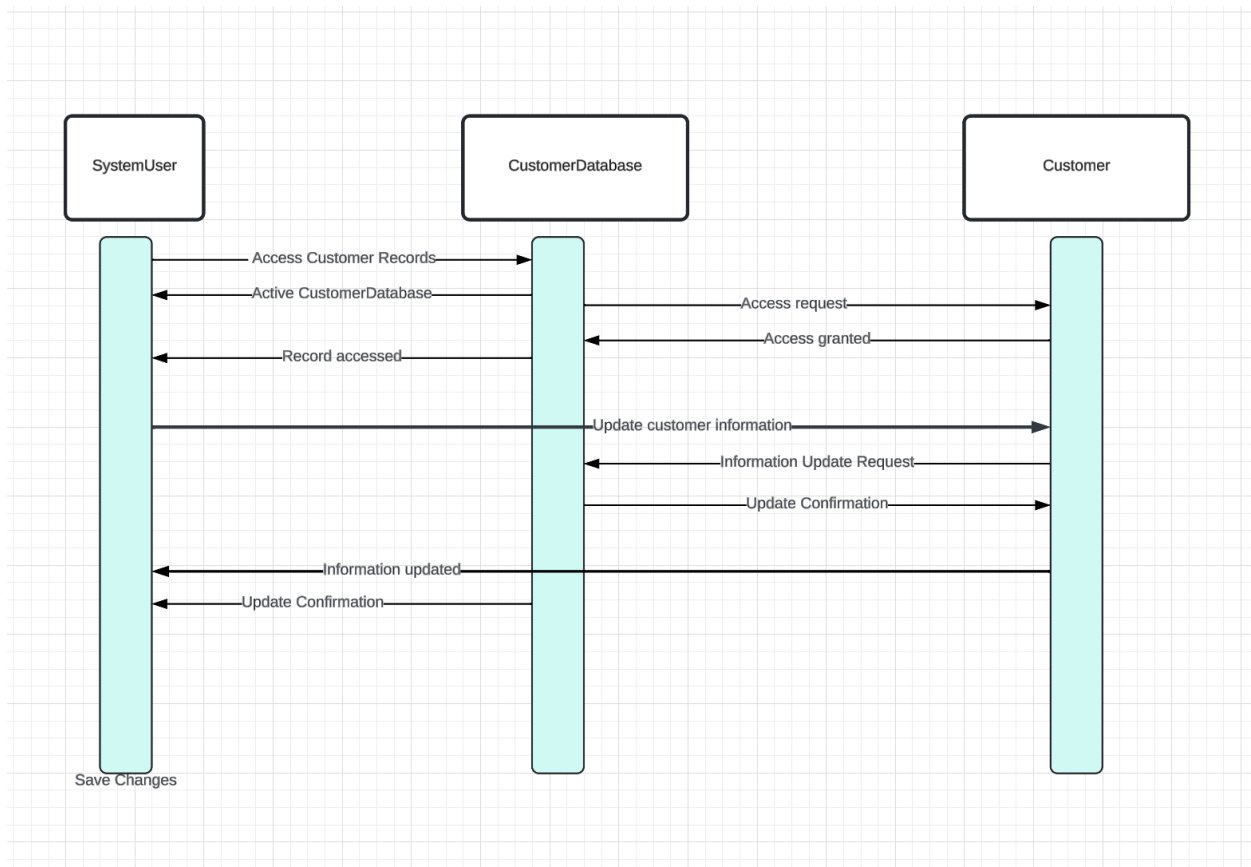
## UML Diagrams



## UML Use Case Diagram: Use Case 1: Register for a lesson - Activity Diagram



## UML sequence diagram: Use case :Manage Customer Information



Technical Requirements: Register for a lesson

### Hardware

- A server with at least 4 cores and 8 GB of RAM
- A database server (optional, if you are using a cloud database service)
- A network connection with sufficient bandwidth to handle the expected traffic load

## Software

- A web server (e.g., Apache, Nginx)
- A database (e.g., MySQL, PostgreSQL)
- A programming language and framework (e.g., Python/Django, Ruby on Rails, Java/Spring Boot)
- An operating system (e.g., Linux, Windows)

## Tools

- A version control system (e.g., Git, Mercurial)
- A build tool (e.g., Maven, Gradle)
- A testing framework (e.g., JUnit, RSpec, Cucumber)
- A deployment tool (e.g., Capistrano, Ansible)

## Infrastructure

- A web hosting provider (e.g., AWS, Azure, Google Cloud Platform)
- A database hosting provider (optional, if you are using a cloud database service)

## Additional Requirements

- The system should be able to handle concurrent users.
- The system should be secure against unauthorized access and data breaches.
- The system should be scalable to handle increased traffic load and additional features in the future.

## Recommendations

- I recommend using a cloud platform to host your system. This will provide you with the scalability and reliability you need without having to worry about managing your own infrastructure.
- I also recommend using a managed database service. This will save you the time and hassle of managing your own database server.
- Be sure to choose a programming language and framework that are well-suited for your needs. Consider factors such as performance, scalability, and ease of development.

- Use a version control system to track changes to your code and collaborate with other developers.
- Automate your build and deployment process using tools such as Maven, Gradle, Capistrano, and Ansible.
- Take steps to secure your system against unauthorized access and data breaches. This includes using strong passwords, encrypting sensitive data, and implementing firewalls and intrusion detection systems.
- Design your system for scalability. This means using a modular architecture and choosing technologies that can scale easily.