CS 255 Assignment Module Seven– Project Two

Keyneisha Mcnealey

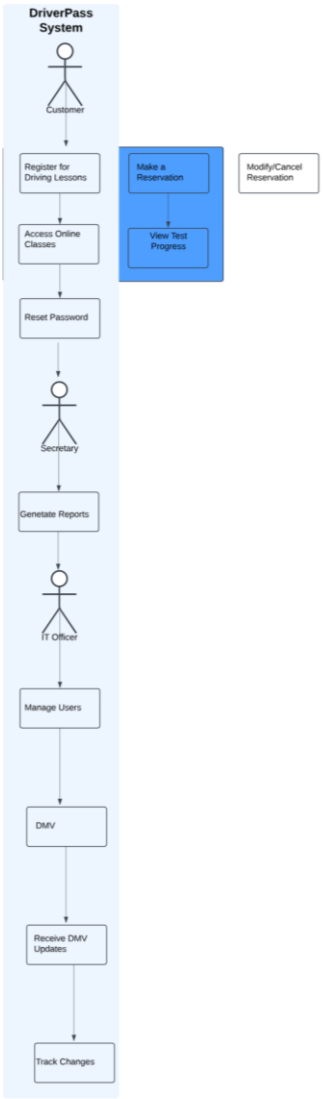
Denise Washington

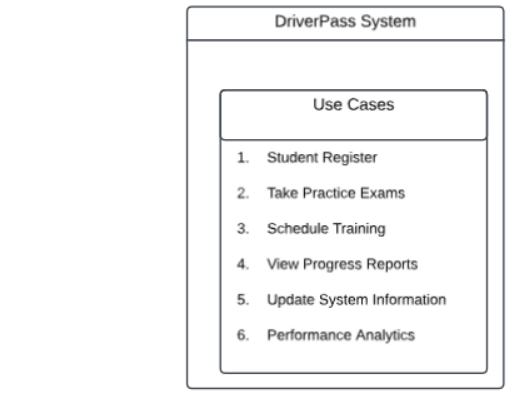
12/15/2024

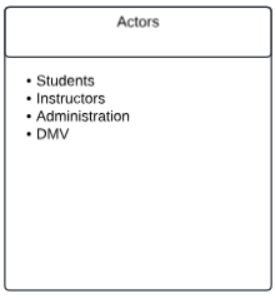
CS 255 System Design Document

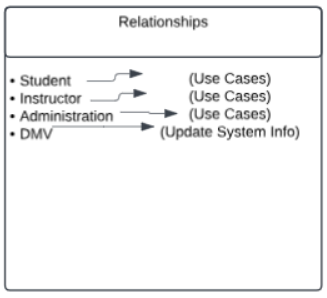
UML Diagrams

UML Use Case Diagram (Respective to UML Activity Diagrams)

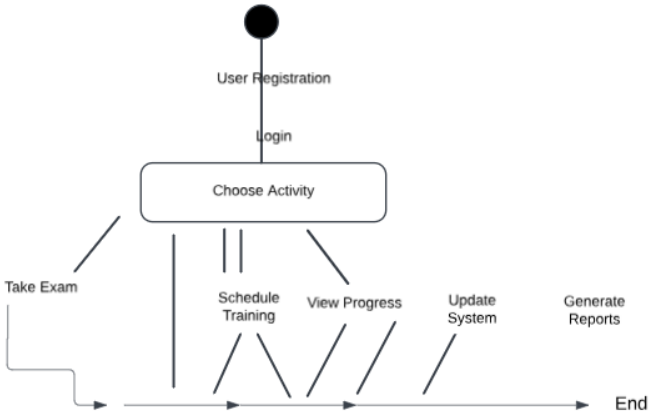


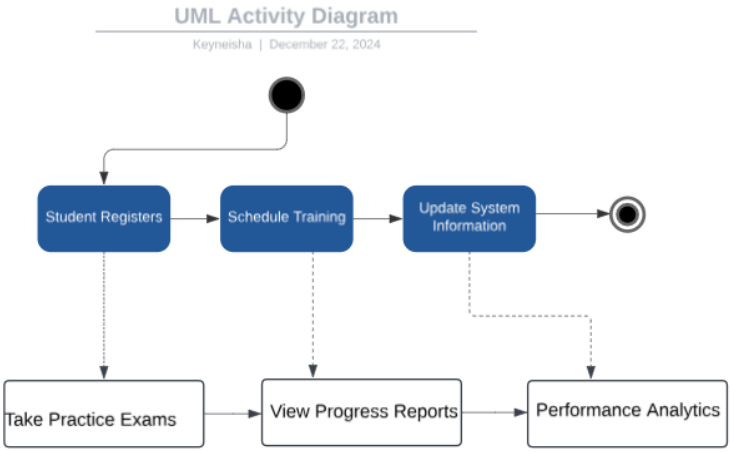




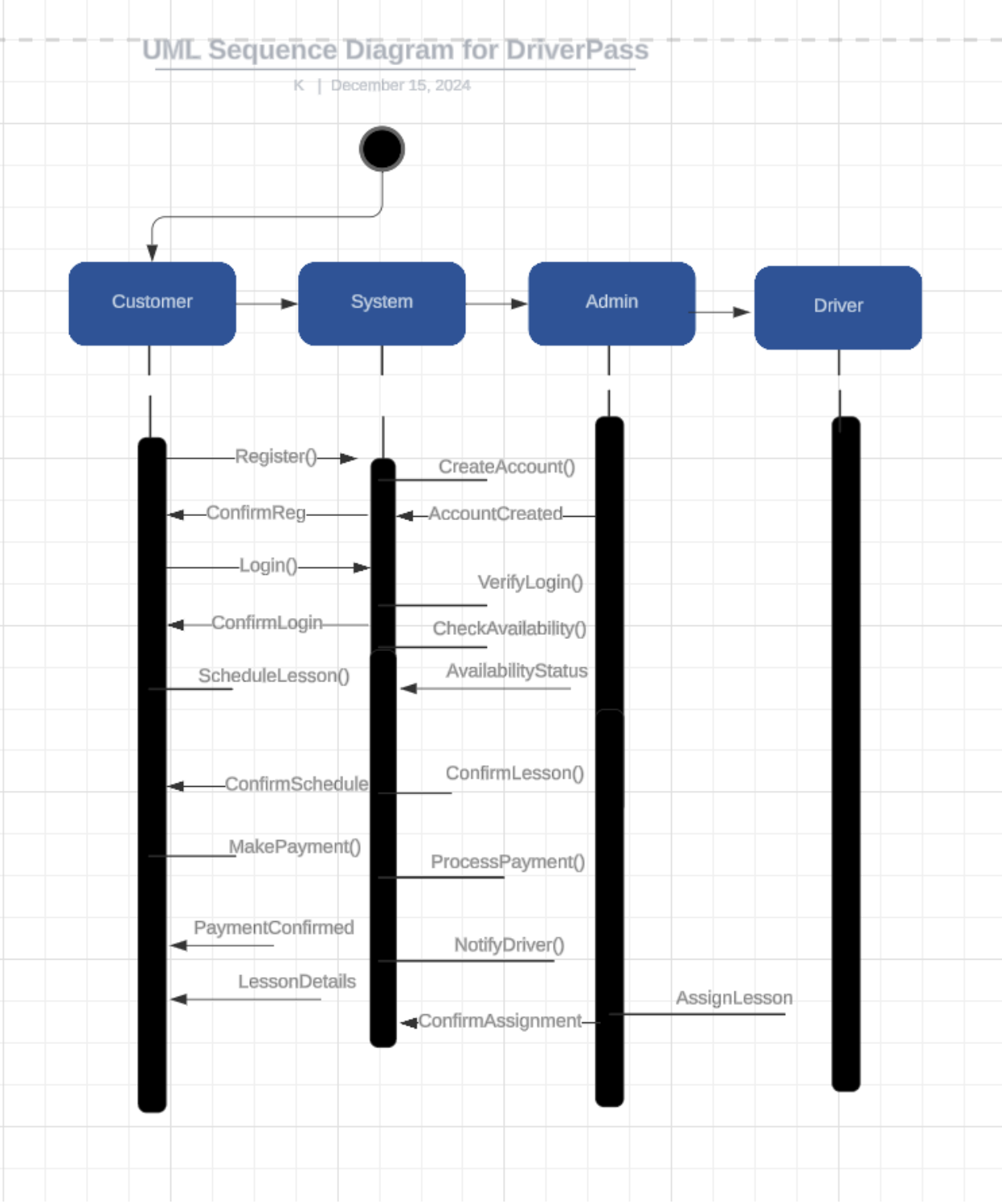


UML Activity Diagrams

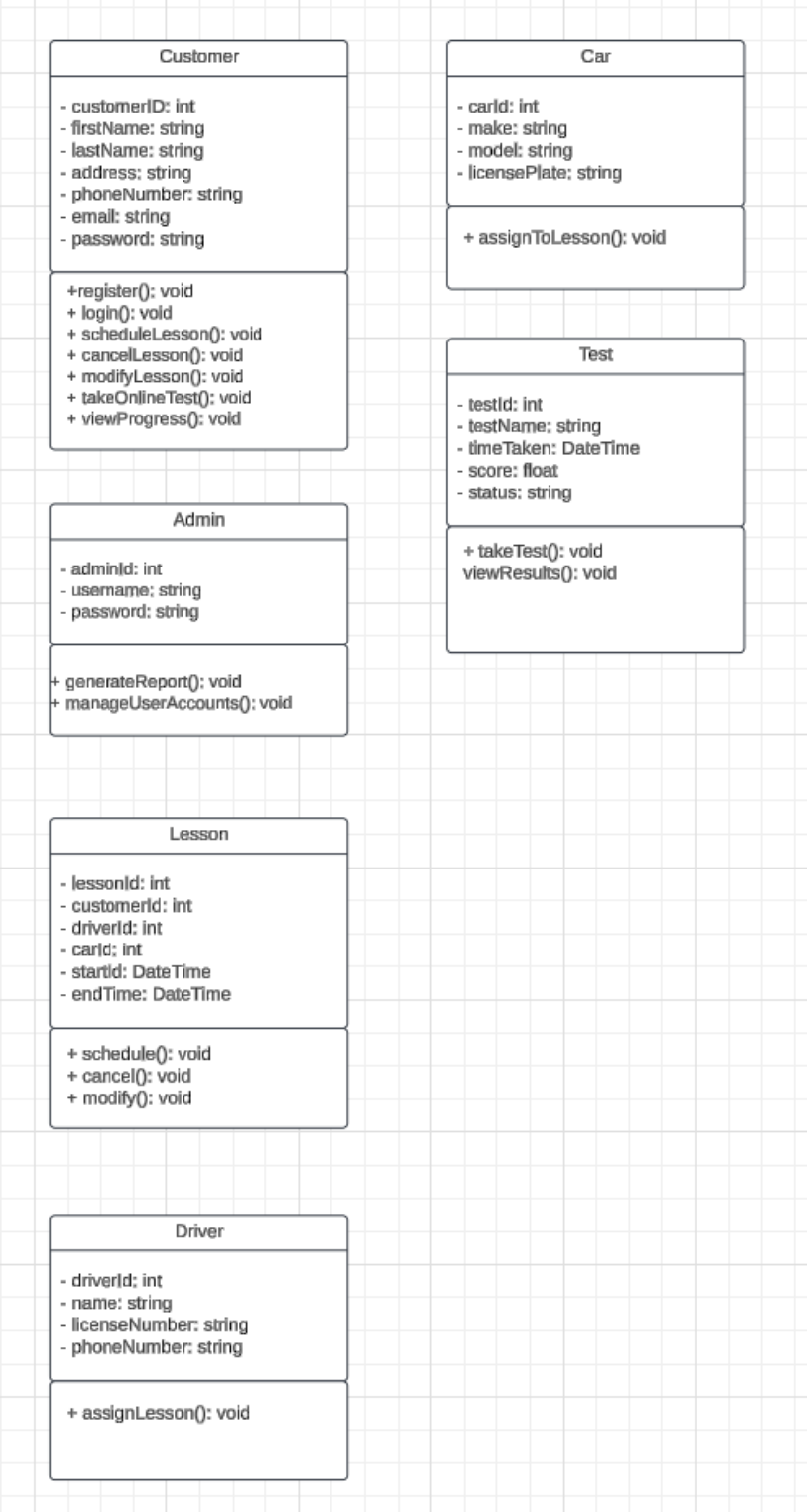




UML Sequence Diagram



UML Class Diagram



Technical Requirements

Hardware Requirements

Servers:

- High-performance web servers are essential for hosting the application and managing multiple concurrent users.
- Database servers must have sufficient storage capacity to manage user data, lesson schedules, test results, and other critical information.
- Backup servers are required for regular data backup and disaster recovery, ensuring data integrity and availability.

Network Infrastructure:

- High-speed internet connectivity is necessary to support real-time data exchange and communication between customers, drivers, and the system.
- Firewalls and network security devices are crucial for protecting the system from unauthorized access and cyber threats.

Client Devices:

- Computers, tablets, and smartphones used by customers, administrators, and drivers are essential for accessing the system.
- Peripheral devices such as printers are needed for generating physical reports and documents.

Software Requirements

Operating Systems:

- Compatibility with major operating systems such as Windows, macOS, and Linux is essential for server-side operations.
- Client-side compatibility with Windows, macOS, iOS, and Android ensures broad accessibility.

Web Server Software:

- Apache HTTP Server or Nginx should be used to serve the web application.
- Load balancers are necessary to distribute incoming traffic and ensure high availability and reliability.

Database Management System (DBMS):

- MySQL, PostgreSQL, or Oracle Database are suitable choices for managing and querying relational data.
- Database replication and clustering are required for scalability and fault tolerance.

Application Frameworks:

- A robust web application framework such as Django (Python), Spring Boot (Java), or ASP.NET Core (C#) is necessary for developing and maintaining the system.
- Front-end frameworks like React, Angular, or Vue.js are essential for building interactive and responsive user interfaces.

Security Software:

- SSL/TLS certificates are required for secure communication between clients and servers.
- Authentication and authorization libraries are needed to manage user roles and permissions.
- Regular security audits and vulnerability assessments are essential for identifying and mitigating risks.

Tools and Utilities

Development Tools:

- Integrated Development Environments (IDEs) like Visual Studio Code, PyCharm, or IntelliJ IDEA are necessary for efficient coding and debugging.
- Version control systems such as Git, and platforms like GitHub or GitLab, are essential for collaborative development and version management.

Monitoring and Logging:

- Monitoring tools like Prometheus and Grafana are required to track system performance and resource usage.
- Logging tools such as ELK Stack (Elasticsearch, Logstash, Kibana) are necessary for real-time log analysis and troubleshooting.

Continuous Integration and Deployment (CI/CD):

- Jenkins, CircleCI, or GitHub Actions are essential for automating the build, testing, and deployment processes.
- Containerization tools like Docker, and orchestration platforms like Kubernetes, are necessary for deploying and managing applications in a scalable and portable manner.

Infrastructure

Cloud Services:

- Cloud platforms like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP) are necessary to host and manage the application's infrastructure.
- Services such as Elastic Compute Cloud (EC2), Azure Virtual Machines, or Google Compute Engine provide scalable computing resources.
- Managed database services like Amazon RDS, Azure SQL Database, or Cloud SQL are essential for database management.

Storage Solutions:

- Cloud storage services like Amazon S3, Azure Blob Storage, or Google Cloud Storage are necessary for storing files, backups, and static assets.
- Persistent storage for database servers ensures data durability and high availability.

Backup and Recovery:

- Automated backup solutions are required for regular data backups and recovery.
- Disaster recovery plans and tools are essential to ensure business continuity in case of system failures or data loss.

Functional and Non-functional Requirements

Functional Requirements:

Training Materials: The system must provide access to online classes and practice tests to facilitate comprehensive driver education.

Reservation System: The system should enable customers to make, cancel, and modify reservations for driving lessons, ensuring flexibility and convenience.

Tracking and Reporting: The system must track user activities and generate comprehensive reports to monitor progress and performance.

Lesson Scheduling: The system should allow scheduling of driving lessons, including the selection of packages and matching with specific drivers and cars, to optimize resource management.

Password Management: The system should implement an automated password reset functionality to enhance user convenience and security.

Nonfunctional Requirements:

Performance Requirements: The system should operate efficiently in a web-based environment, offering fast data retrieval and updates, thereby ensuring a seamless user experience.

Platform Constraints: The system must be compatible with major operating systems such as Windows and Unix, and deployable on a cloud platform to ensure scalability and minimal maintenance overhead.

Accuracy and Precision: The system must ensure accurate data entry and retrieval, distinguishing between different users by unique identifiers, thereby maintaining data integrity.

Adaptability: The system should allow modifications such as adding, removing, or updating user information without requiring code changes, ensuring flexibility to accommodate future needs.

Security: The system must include strong user authentication mechanisms, secure data exchange protocols, and account lockout features after multiple failed login attempts to safeguard sensitive information.

Infrastructure:

These requirements encompass both functional and nonfunctional aspects, providing a robust framework for the system.

Cloud Services: Cloud platforms are essential for hosting and managing the system's infrastructure, ensuring scalability and reliability.

Storage Solutions: Cloud storage services are necessary for backups and data management, ensuring data durability and availability.

Backup and Recovery: Automated backup solutions and disaster recovery plans are crucial for ensuring business continuity in case of system failures or data loss.

Tools and Utilities:

These requirements support the development and operational aspects of the system.

Development Tools: Integrated Development Environments (IDEs) and version control systems are essential for efficient coding, debugging, and collaborative development.

Monitoring and Logging: Tools for performance monitoring and log analysis are necessary for tracking system performance and troubleshooting issues.

CI/CD: Continuous integration and deployment tools are essential for automating the build, testing, and deployment processes, ensuring a streamlined development workflow.

This detailed overview of the functional and nonfunctional requirements, along with infrastructure and tools, provides a comprehensive foundation for the DriverPass system, ensuring its efficiency, reliability, and scalability.