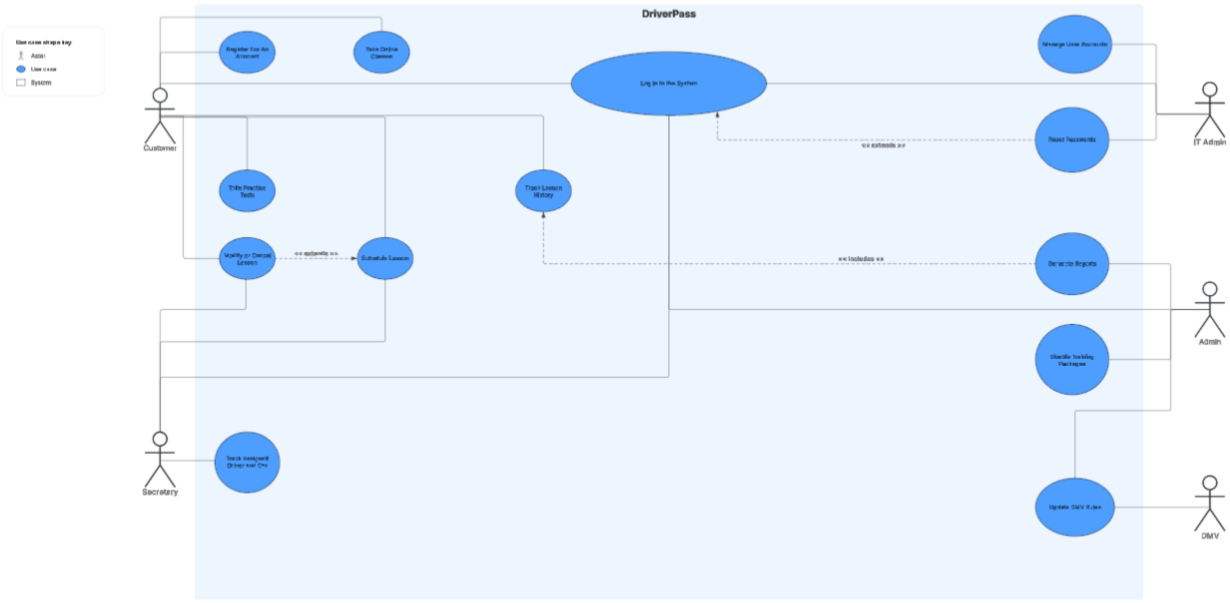
Southern
New Hampshire
University

CS 255 System Design Document Template

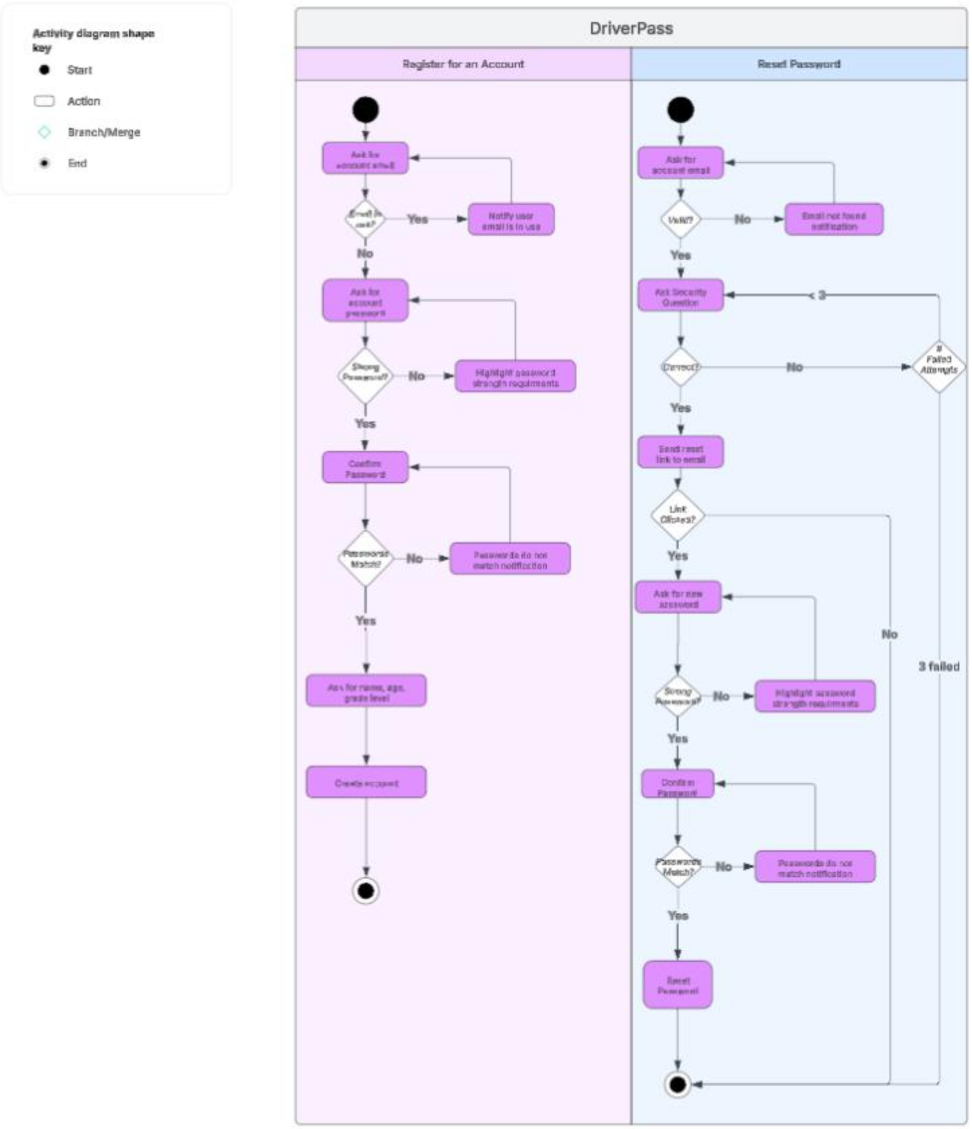
This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client's needs. There is no required length for the final document. Instead the goal is to complete each section based on what your client's needs are. Remove this note when you are finished, and replace all bracketed text with the relevant information.

UML Diagrams

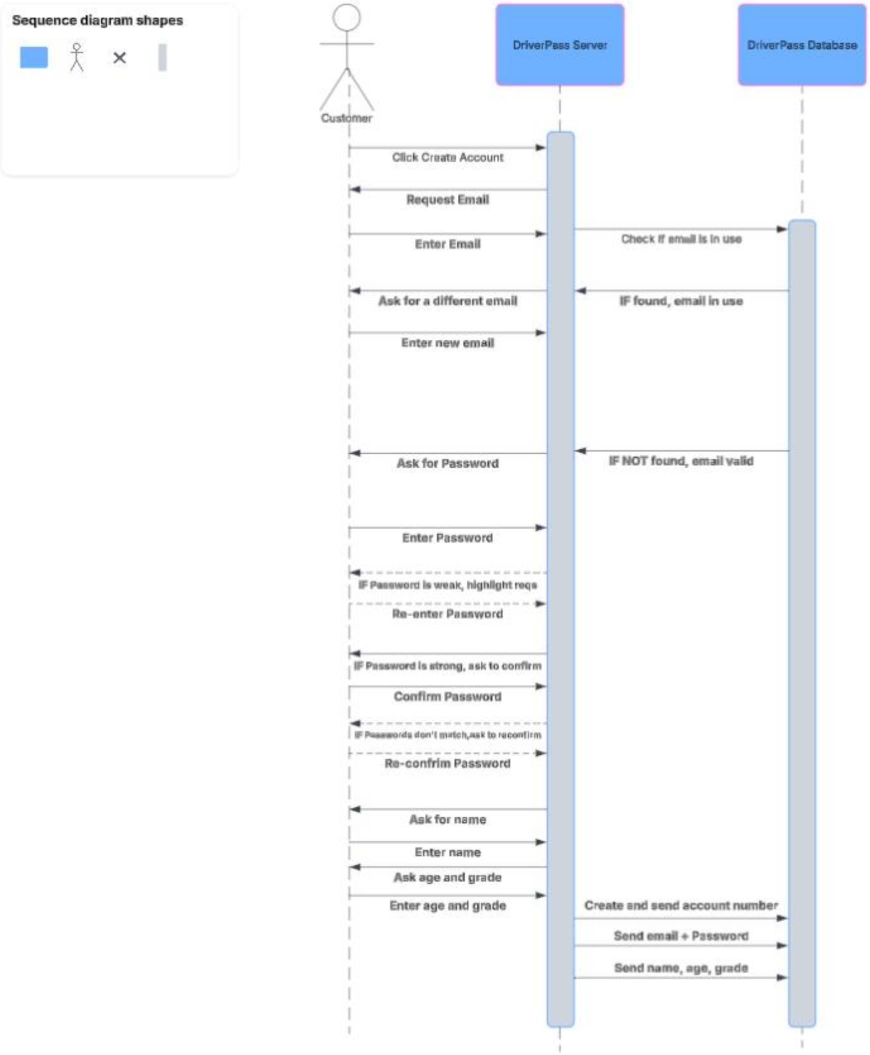
UML Use Case Diagram



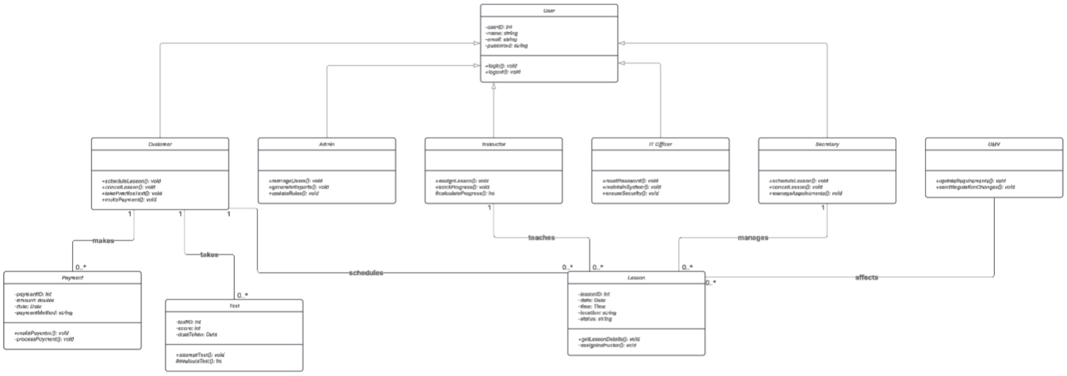
UML Activity Diagrams



UML Sequence Diagram



UML Class Diagram



Technical Requirements

The DriverPass system requires a combination of hardware, software, tools, and infrastructure to function efficiently, securely, and reliably. The system will be hosted on a cloud-based server infrastructure, utilizing providers such as AWS, Google Cloud, or Microsoft Azure to ensure scalability and availability. Client devices, including desktops, laptops, tablets, and smartphones, must be able to access the system without performance degradation. Secure storage solutions will be necessary for managing user data, lesson progress, and scheduling details, while a stable internet connection is required to support real-time interactions and API calls. Additionally, security measures such as hardware firewalls and intrusion detection systems will be implemented to protect sensitive user data from potential threats.

The software stack for the DriverPass system will consist of both front-end and back-end technologies. The system will be web-based, utilizing modern development frameworks such as React or Angular for an interactive and user-friendly interface. The backend will be built using scalable technologies such as Node.js, Python (Django/Flask), or Java (Spring Boot) to handle user authentication, scheduling, and financial transactions. A relational database management system, such as MySQL or PostgreSQL, will be used to store structured data securely and efficiently. Furthermore, the system will integrate with DMV APIs to provide real-time updates on driving test requirements and rule changes. To enhance security, user authentication will be managed through OAuth 2.0 or similar secure authentication services.

For development and maintenance, the system will utilize industry-standard tools and frameworks. Developers will work with IDEs such as Visual Studio Code, PyCharm, or IntelliJ IDEA for coding and debugging. Version control will be managed through Git repositories hosted on platforms like GitHub or GitLab, ensuring efficient code collaboration. Automated testing tools, including Selenium for UI testing and Postman for API validation, will be used to maintain system reliability. Monitoring tools such as AWS CloudWatch or Prometheus will track system performance, helping administrators detect and resolve issues in real-time. Additionally, a robust logging system will record user activities, including login attempts, scheduling changes, and payment transactions, supporting security auditing and system integrity.

Security and infrastructure are critical aspects of the DriverPass system. All data exchanges between the client and server will be encrypted using TLS protocols, while user data at rest will be secured using AES-256 encryption. Role-based access control will be implemented to ensure that different user roles, including students, instructors, and administrators, have the appropriate permissions. To prevent data loss, the system will employ automated backup solutions, ensuring quick recovery in the event of a failure or cyberattack. System updates will be deployed on a monthly basis, with critical security patches applied as necessary to maintain system stability and safeguard user information. These technical requirements ensure that DriverPass remains a secure, scalable, and efficient platform, meeting the needs of both users and administrators.