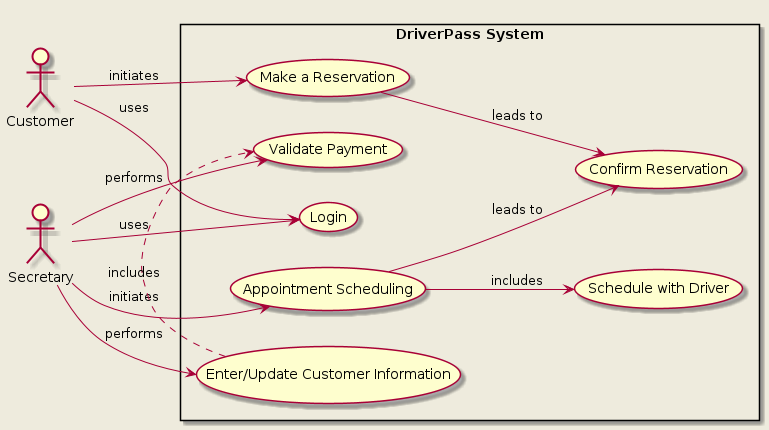
# CS 255 System Design Document Template

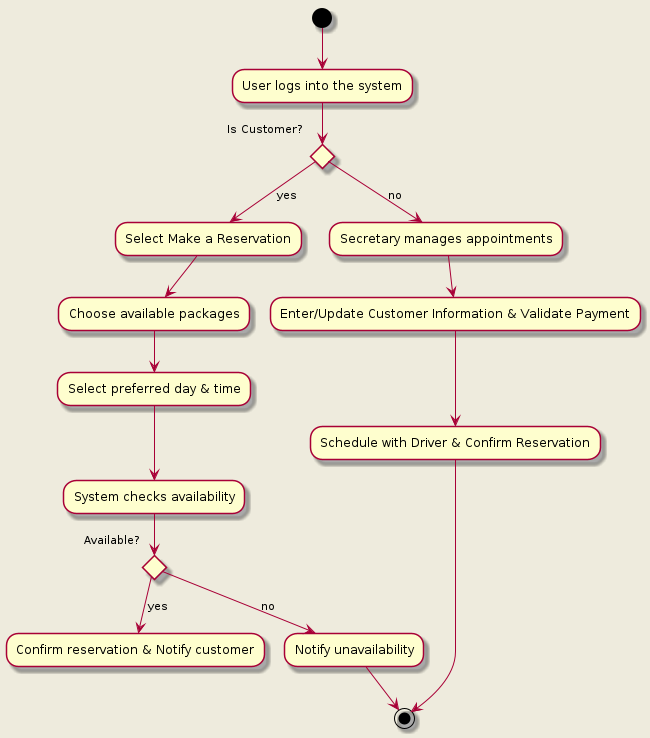
This template lays out all the different sections that you need to complete for Project Two. Each section has guidance to prompt your thinking. You will need to continually reference the interview transcript as you work to make sure that you are addressing your client’s needs. There is no required length 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

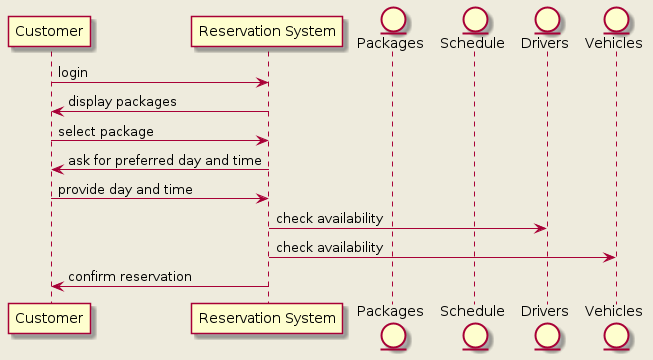
### UML Use Case Diagram



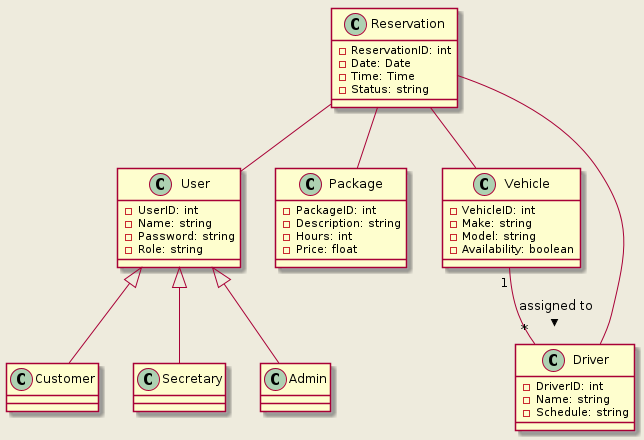
### UML Activity Diagrams



### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements

1. Hardware Requirements:

Servers: For hosting the web application, database servers, and possibly for caching solutions like Redis.

Workstations: Computers for the secretary and administration staff to manage appointments and other system requirements.

Networking Equipment: Routers, switches, and load balancers to manage network traffic.

Mobile Devices: Smartphones or tablets for customers to access the system remotely for reservation purposes.

2. Software Requirements:

Operating System: Server OS like Linux or Windows Server, and desktop OS like Windows or macOS for workstations.

Database Management System (DBMS): A DBMS like PostgreSQL, MySQL, or MongoDB to handle the reservation data, customer information, and other related data.

Web Server: Apache, Nginx, or similar to serve the web application.

Programming Languages: Languages such as JavaScript (with Node.js), Python, Ruby, or PHP for backend development, and HTML, CSS, JavaScript for frontend development.

Frameworks and Libraries: Frontend frameworks like React or Angular, and backend frameworks like Express.js, Django, or Ruby on Rails.

3. Tools:

Integrated Development Environment (IDE): Tools like Visual Studio Code, IntelliJ IDEA, or Eclipse for software development.

Version Control: Git and hosting services like GitHub or GitLab for version control and collaborative coding.

Project Management: Tools like Jira, Trello, or Asana for tracking tasks and managing the development process.

4. Infrastructure Requirements:

Cloud Services: AWS, Google Cloud, or Azure for cloud hosting, storage, and other services like load balancing and auto-scaling.

Content Delivery Network (CDN): A CDN like Akamai or Cloudflare to serve static content efficiently.

Security: Firewalls, SSL certificates for secure communication, and tools for regular security audits and compliance checks.

Backup and Recovery: Solutions for data backup and disaster recovery.

5. Additional Technical Specifications:

APIs: Development of RESTful or GraphQL APIs for interaction between the frontend and backend systems.

Payment Gateway Integration: For processing payments, integration with services like Stripe, PayPal, or Square.

Authentication: A system for managing user roles and permissions, and for securely handling login operations, potentially using OAuth or JWT.

Email Service: For sending out reservation confirmations and notifications, integration with an email service provider like SendGrid or Amazon SES.

6. Non-Functional Requirements:

Scalability: The system should be scalable to handle growing numbers of users and data.

Performance: Optimization for fast response times and efficient processing of requests.

Reliability: High availability and fault tolerance to ensure the system remains operational with minimal downtime.

Security: Secure storage and transmission of data, regular security updates, and adherence to best practices in data protection.