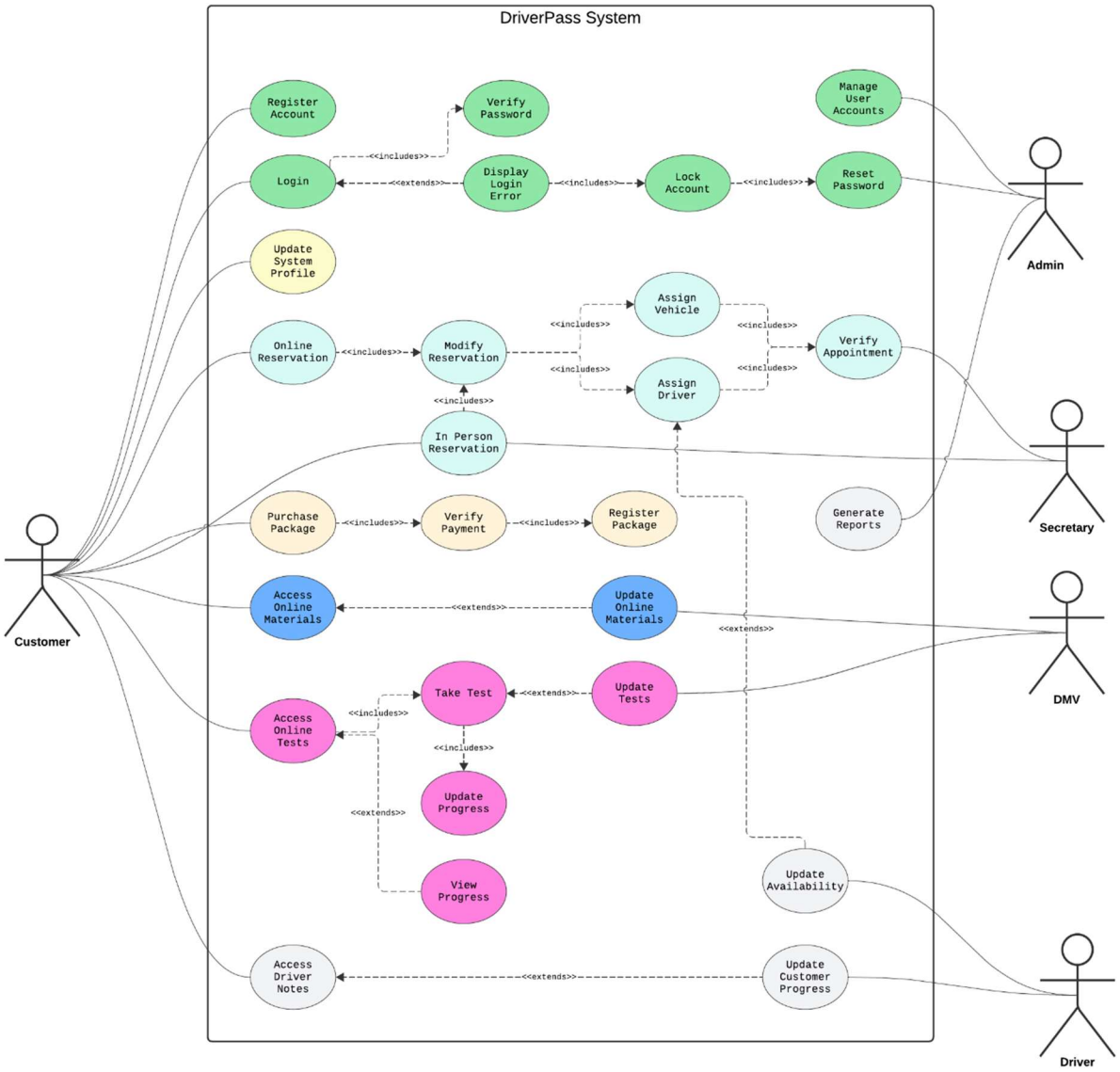
Southern New Hampshire University

CS 255 System Design Project Two

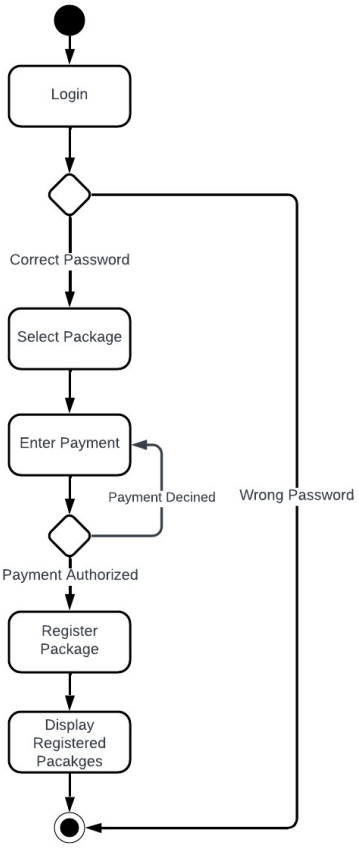
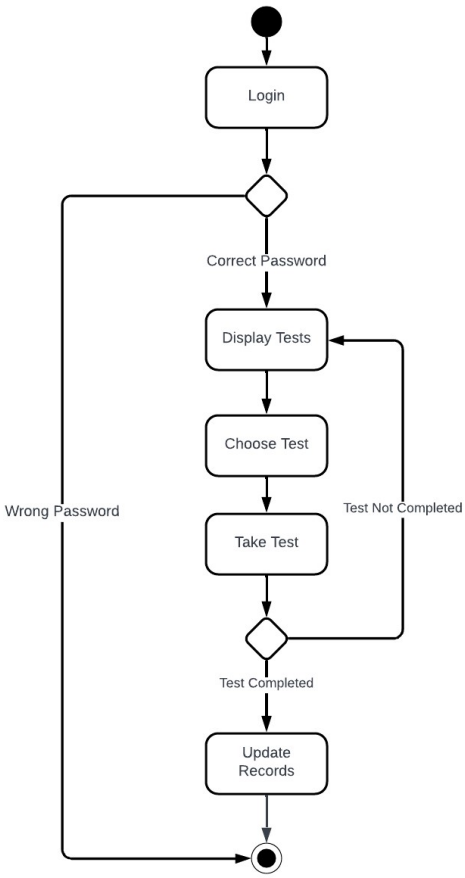
Grant McCord
CS-255
October 20, 2024

UML Diagrams

UML Use Case Diagram

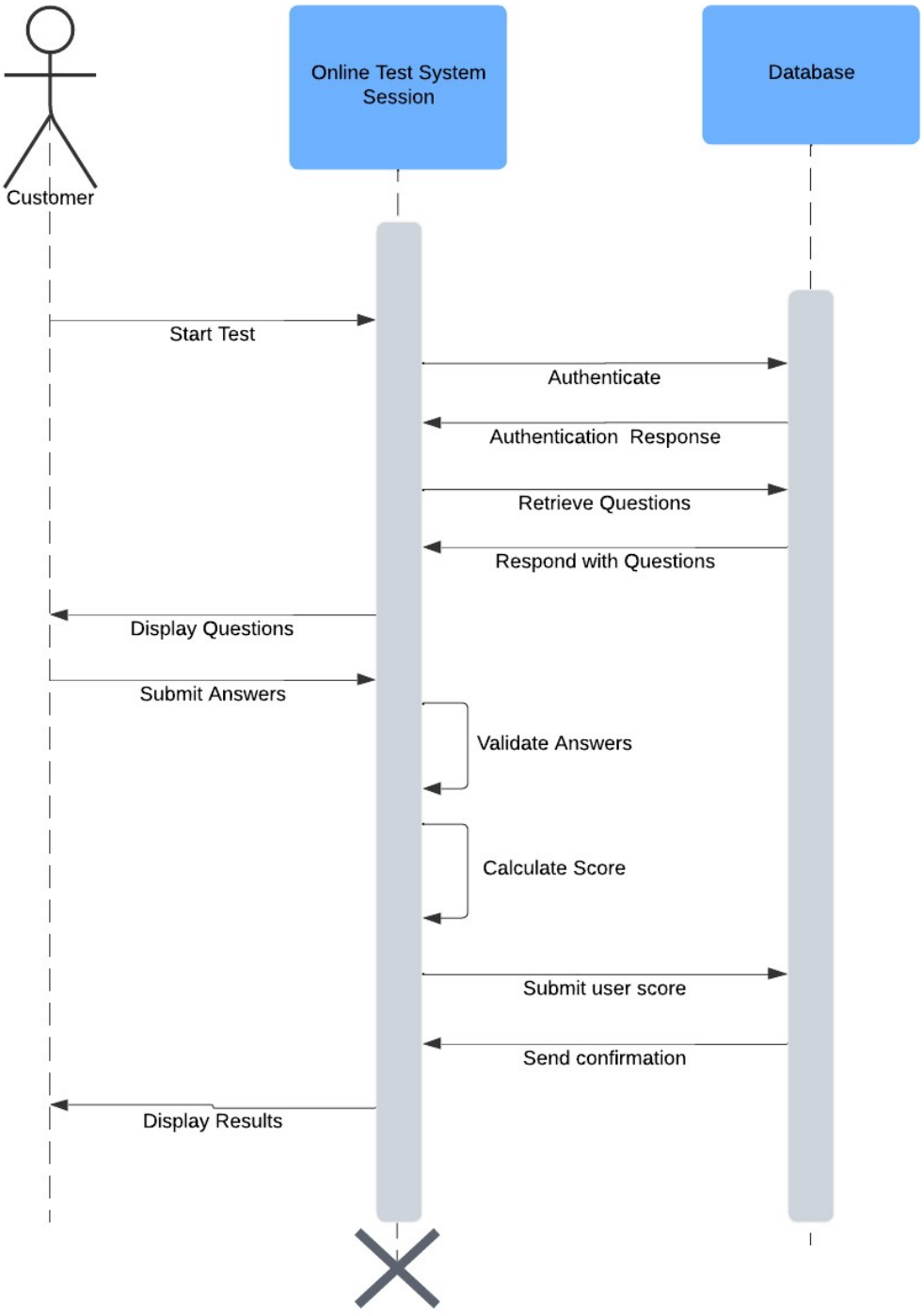


UML Activity Diagrams

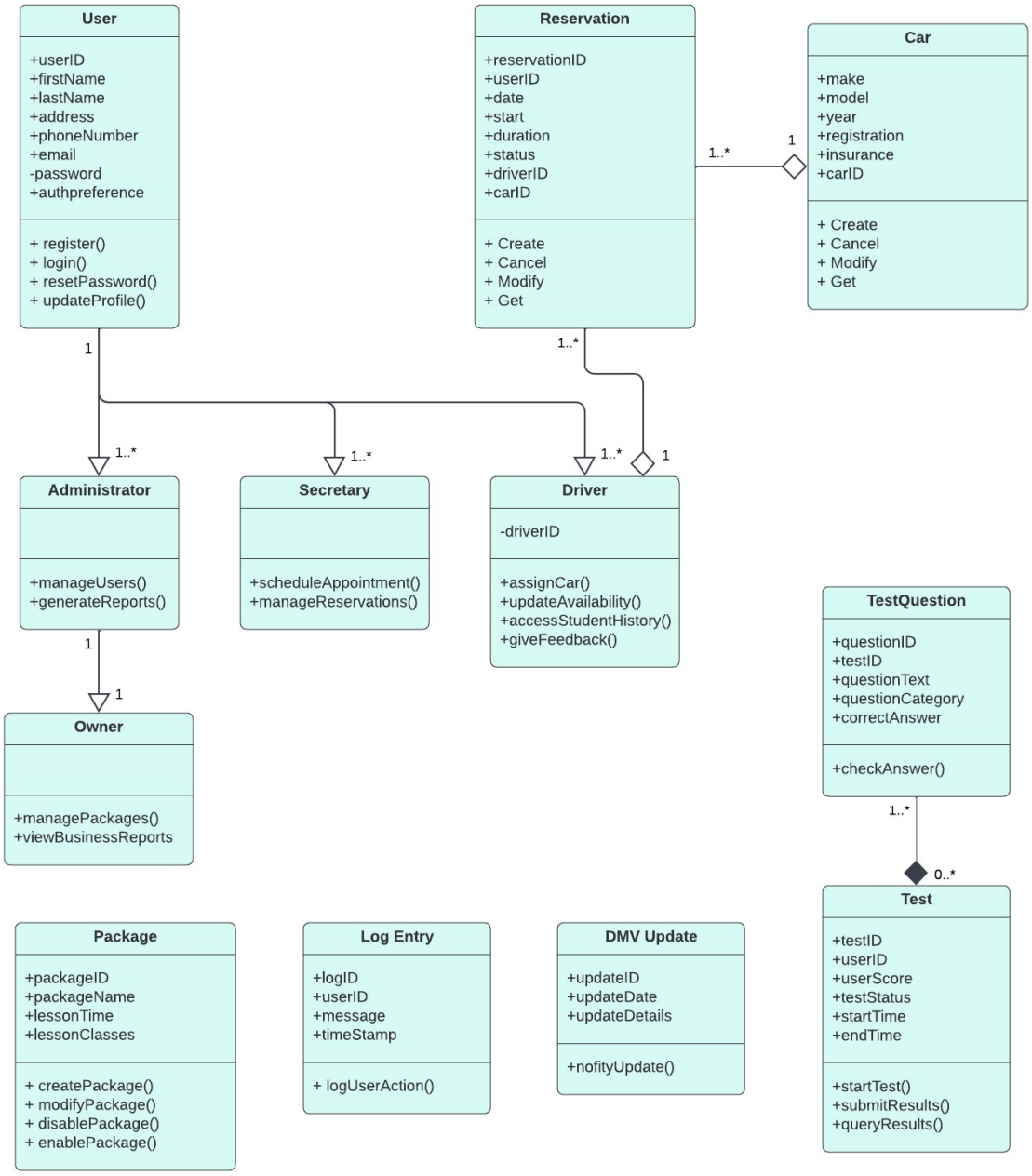
Purchase Package	User Taking an Online Test
	

UML Sequence Diagram

Customer taking an online test



UML Class Diagram



Southern New Hampshire University

Technical Requirements

1. Hardware Requirements

The DriverPass system will require robust server infrastructure to ensure high availability and performance. A cloud-based hosting solution is recommended to handle peak loads, especially during high-traffic periods, such as weekends or before DMV test dates. The system should be capable of scaling resources dynamically based on demand, accommodating up to 2,000 reservations per day.

2. Software Requirements

The system will operate on a web-based platform compatible with all major browsers, including Chrome, Firefox, Safari, and Edge. The backend will utilize a relational database management system (RDBMS) for efficient data storage and retrieval. Additionally, RESTful APIs will be essential for integrating third-party services, such as DMV notifications and payment processing systems.

3. Development Tools

To facilitate the development process, the team will use modern web development frameworks and libraries, such as React or Angular for the front end, and Node.js or Django for the backend. Version control tools like Git will be employed to manage code changes effectively. Testing tools should include automated testing frameworks to ensure functionality and security before deployment.

4. Infrastructure Requirements

The system will be hosted on a cloud platform, such as AWS or Azure, which provides reliable uptime and security features. The infrastructure must support load balancing to distribute traffic evenly and ensure seamless user experiences. Regular backups should be automated, with data recovery procedures in place to mitigate any potential data loss.

5. Security Measures

The system will implement robust security protocols, including HTTPS for secure communication between clients and servers. User authentication will require unique usernames and strong passwords, with options for two-factor authentication (2FA). Role-based access controls will ensure that users have the appropriate permissions, and the system will automatically lock accounts after multiple failed login attempts.

6. Compliance and Data Protection

The system will adhere to data protection regulations, ensuring user consent for data collection and the ability for users to access or delete their personal data. Integration with the DMV will require compliance with their data handling protocols. Regular security audits should be conducted to ensure ongoing compliance with applicable laws and standards.

7. Performance Requirements

The system must respond to user interactions within two seconds and load initial application interfaces in under three seconds. It should support at least 100 simultaneous users without performance degradation and ensure real-time data updates to maintain current information for users. Availability should be guaranteed 24/7 with minimal downtime.

8. User Interface Requirements

The user interface must be intuitive and responsive, allowing access via both mobile devices and desktop computers. It should feature search functionality, user management options, and easy navigation for scheduling lessons and taking tests. Feedback mechanisms must also be incorporated to facilitate communication between students and instructors.