# DriverPass System Design Document

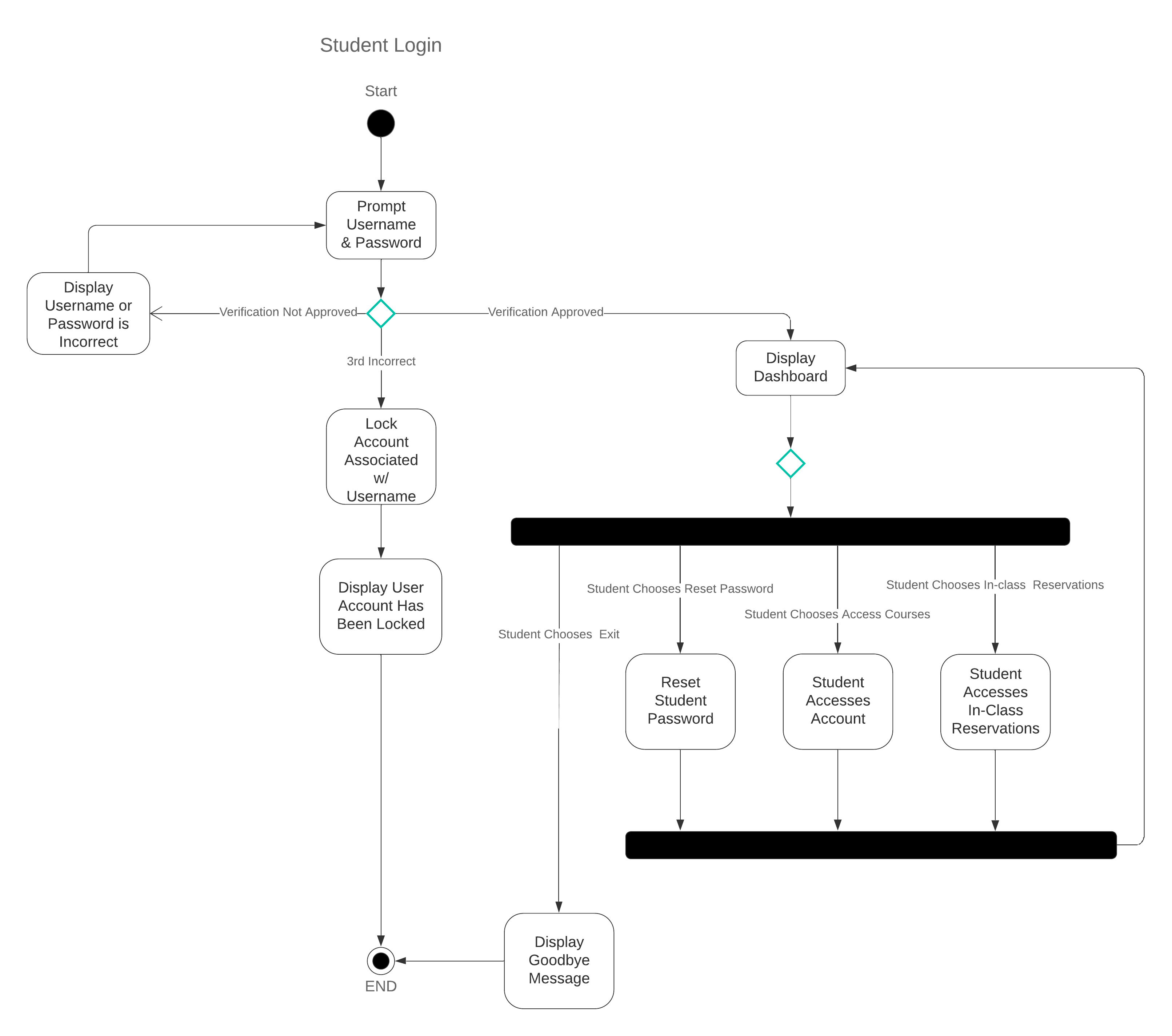
## UML Diagrams

### UML Use Case Diagram

A diagram of a diagram

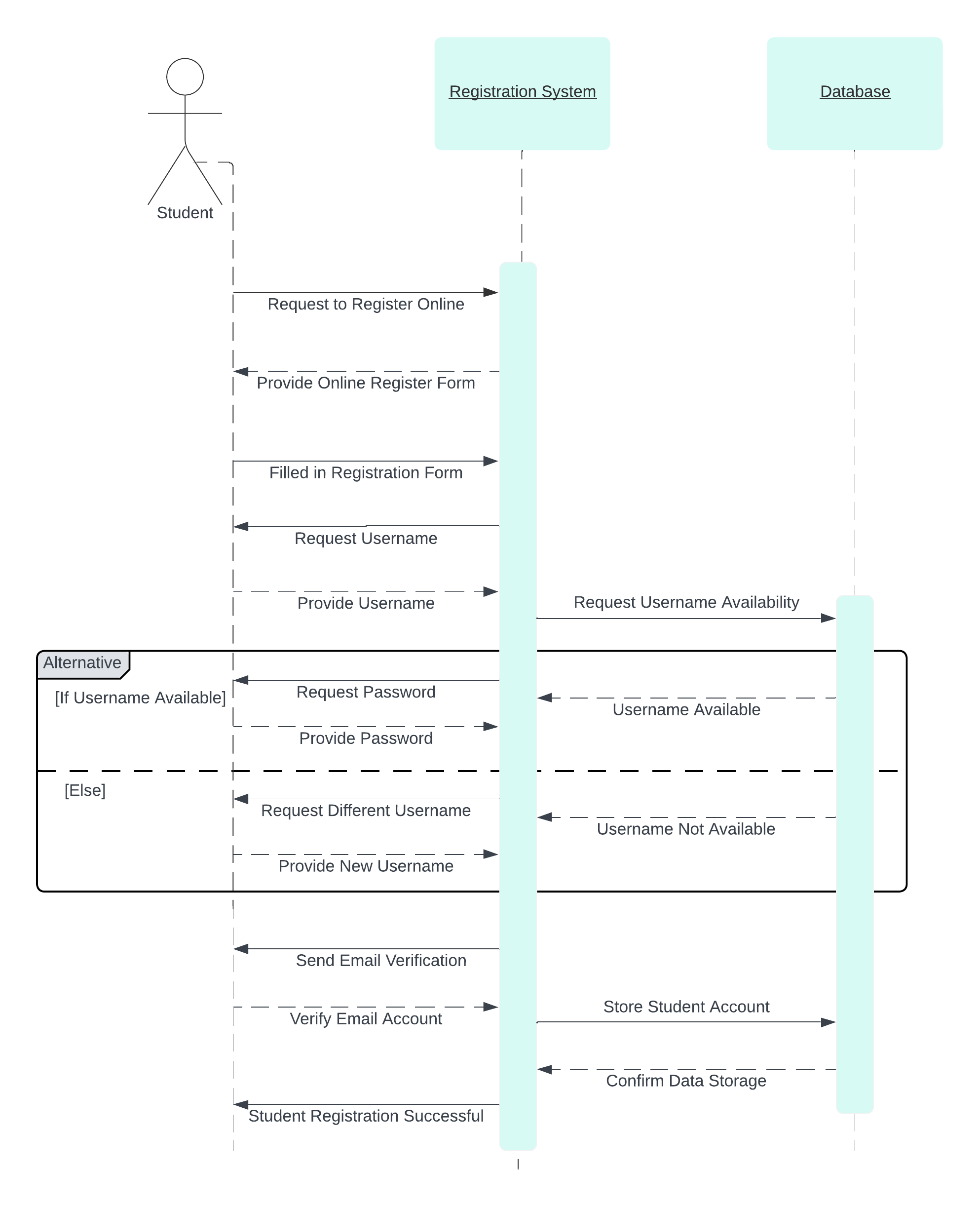
Description automatically generated

### UML Activity Diagrams

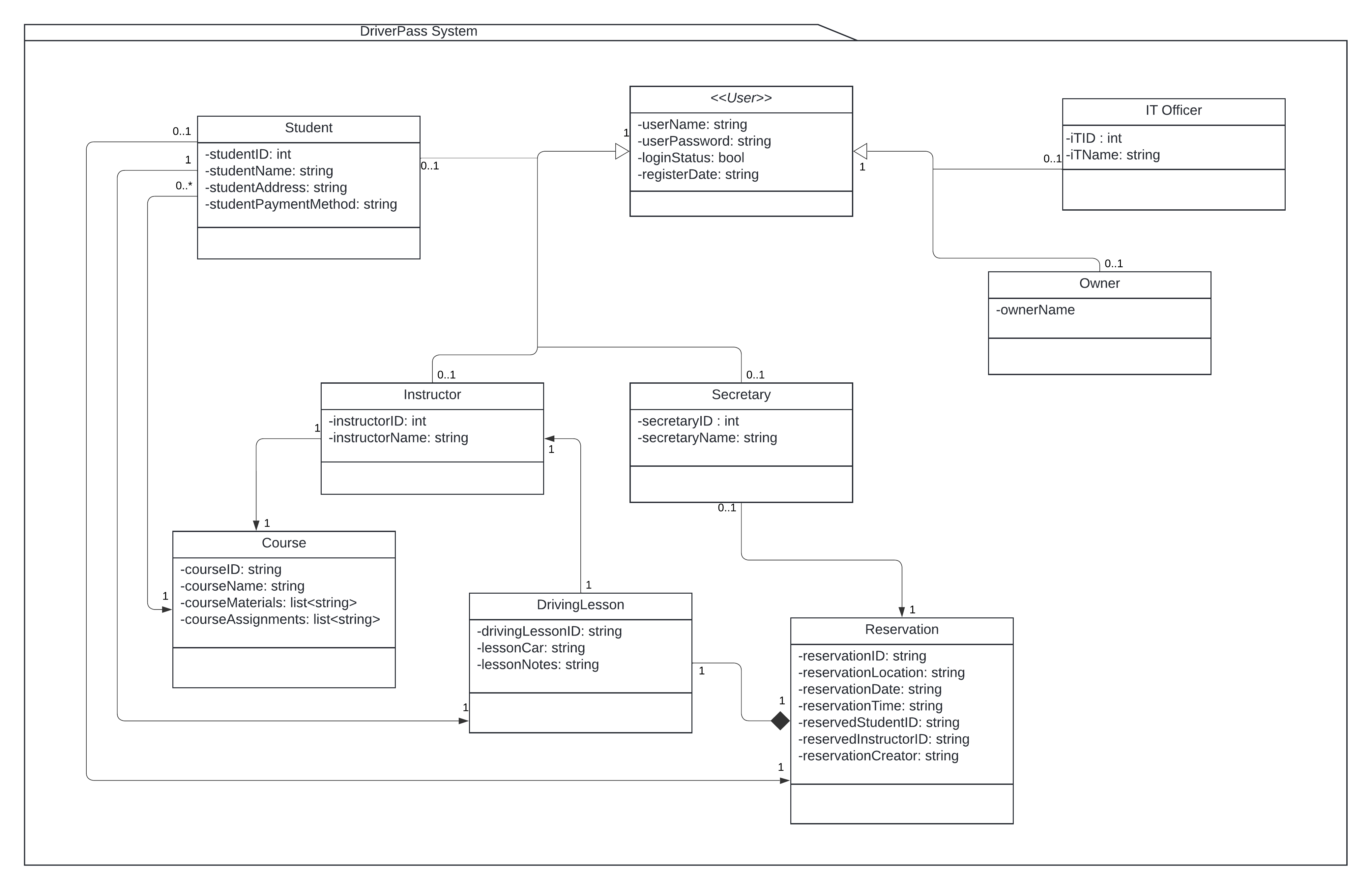


### A diagram of a company Description automatically generated

### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements

**Hardware Requirements:**

1. **Server Infrastructure:**
   * High-performance cloud based servers to host the web-based application and database.
   * Adequate storage capacity to store user data, reservations, and system logs.
   * Redundancy and failover mechanisms to ensure system reliability.
2. **Client Devices:**
   * Compatibility with various devices, including computers and mobile devices.
   * Responsive design to cater to different screen sizes and resolutions.
   * Recommendations for minimum hardware specifications for optimal user experience.
3. **Network Infrastructure:**
   * Stable and high-speed internet connection for seamless data exchange.
   * Implement security measures to protect against network vulnerabilities.

**Software Requirements:**

1. **Web-based Application:**
   * Development using suitable web frameworks (Node.js, React.js, Angular ).
   * Compatibility with major web browsers (Chrome, Firefox, Safari, Edge).
   * Integration of HTTPS protocol for secure data transmission.
2. **Database Management System (DBMS):**
   * Integration with a robust cloud-based DBMS for efficient data storage and retrieval.
   * Regular backups and data consistency checks to maintain database integrity.
3. **Cloud Services:**
   * Deployment on a cloud-based platform (e.g., AWS, Google Cloud) for scalability and reliability.
   * Integration with cloud services for storage, authentication, and notification functionalities.
4. **Development Tools:**
   * Version control tools (e.g., Git) for collaborative development and code management.
   * Integrated Development Environment (IDE) suitable for the chosen programming language.
5. **Security Software:**
   * Implementation of encryption protocols (SSL/TLS) for secure data transfer.
   * Regular security audits and vulnerability assessments to identify and address potential threats.
   * Security software to protect against common web vulnerabilities (e.g., firewalls, intrusion detection systems).

**Infrastructure Requirements:**

1. **Scalability:**
   * Design architecture to handle an increased user load during peak hours.
   * Implement load balancing for efficient distribution of user requests.
2. **Monitoring and Logging:**
   * Implementation of logging mechanisms for tracking user activities, errors, and system performance.
   * Integration with monitoring tools to detect and address issues in real-time.
3. **Backup and Recovery:**
   * Regular automated backups of the system data to prevent data loss.
   * Recovery mechanisms to restore the system to a stable state in case of failures.

**Development Methodologies:**

1. **Agile Development:**
   * Adoption of agile development methodologies for iterative and incremental system development.
   * Regular sprint cycles with defined goals and deliverables.

**Collaboration Tools:**

1. **Communication Platforms:**
   * Use of collaboration tools (e.g., Slack, Microsoft Teams) for effective communication among team members and stakeholders.
   * Regular status meetings and updates to ensure alignment with project goals.

**Documentation:**

1. **Technical Documentation:**
   * Comprehensive documentation for codebase, APIs, and system architecture.
   * User manuals and guides for system administrators and end-users.