

CS 255 System Design Document

Eric May

UML Use Case Diagram

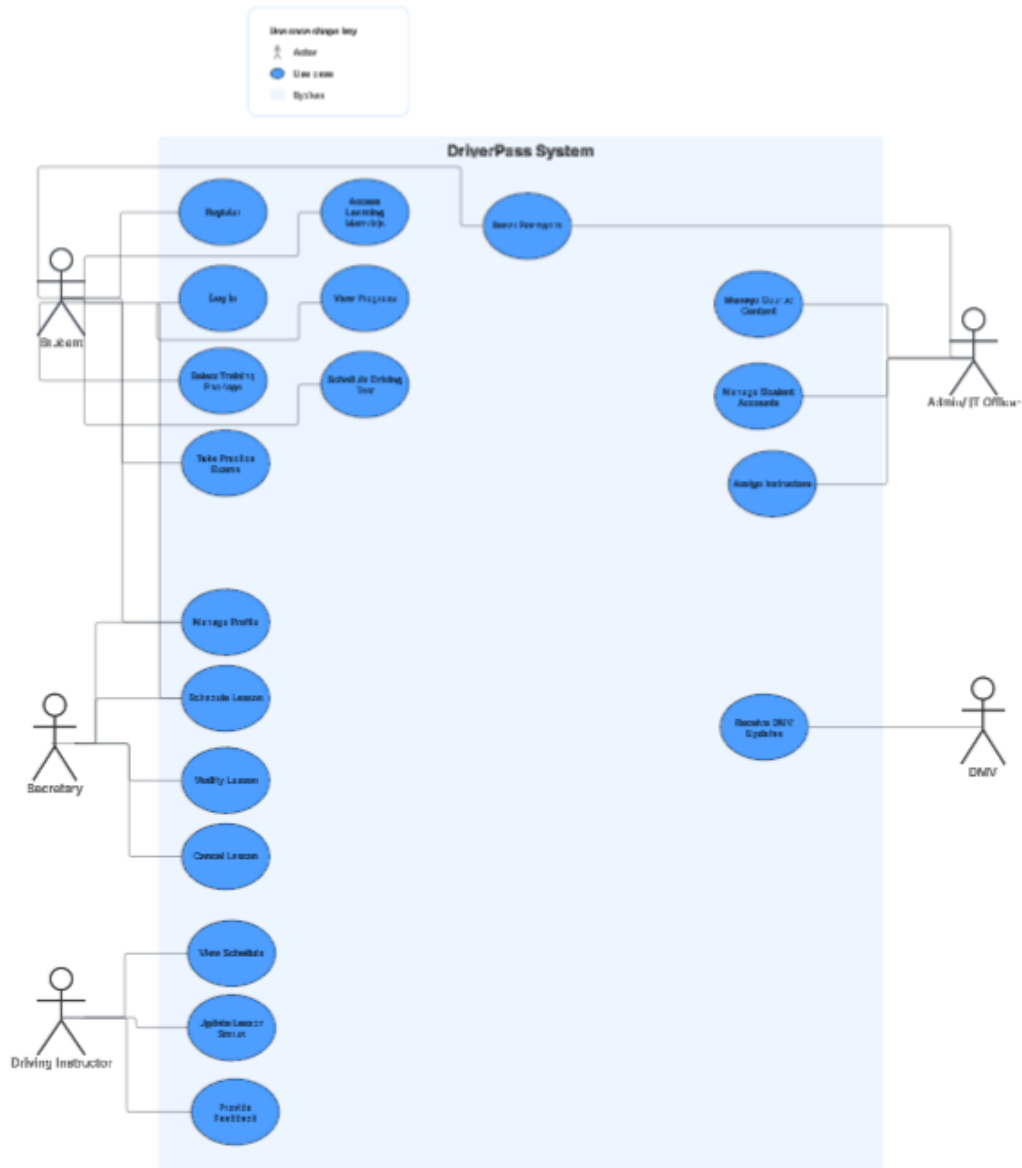


Figure 1. Overview of how students, the secretary, instructors, the IT admin, and the DMV interact with Driverpass to schedule lessons, take practice tests, and keep policies current.

UML Activity Diagrams

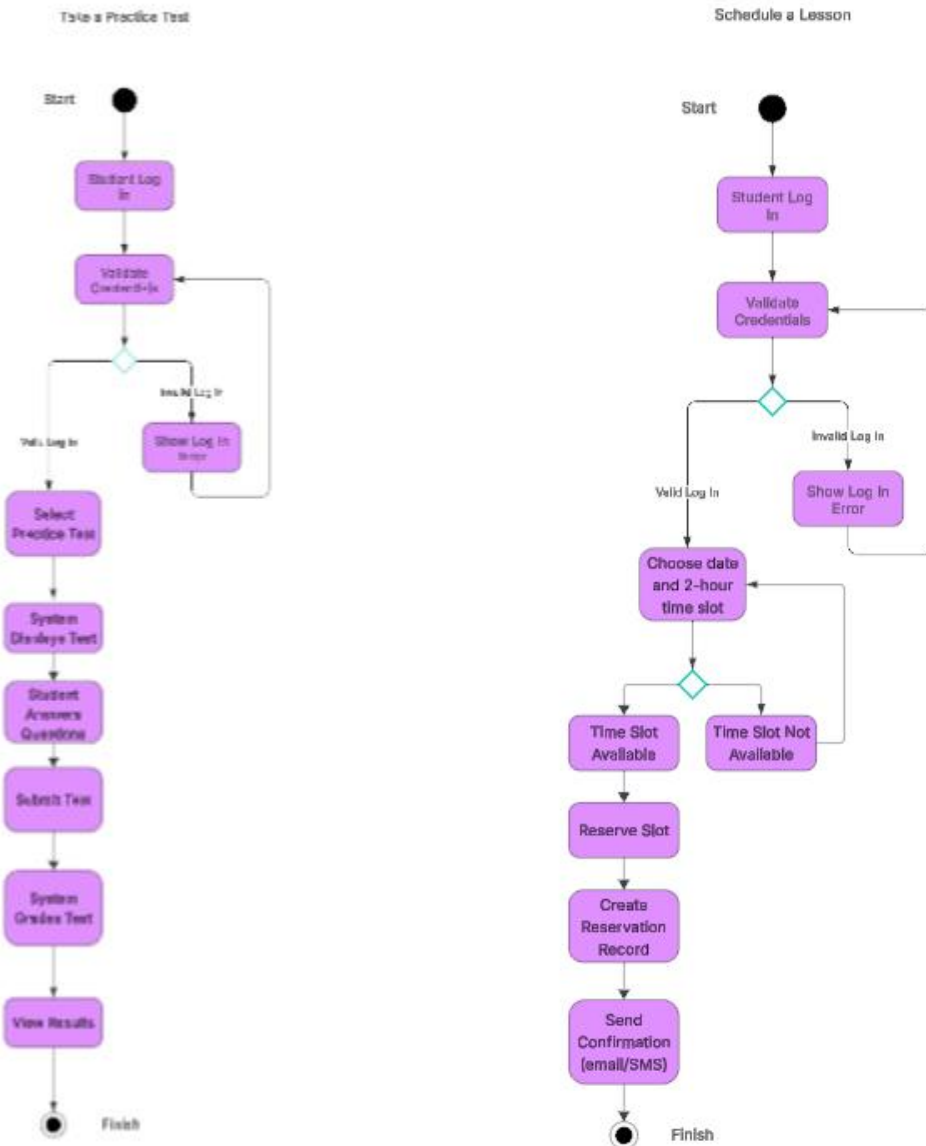


Figure 2. Steps a student follows to take an online practice test and receive instant results.

Figure 3. Steps a student follows to book a two-hour lesson, with handling for invalid logins and unavailable time slots.

UML Sequence Diagram

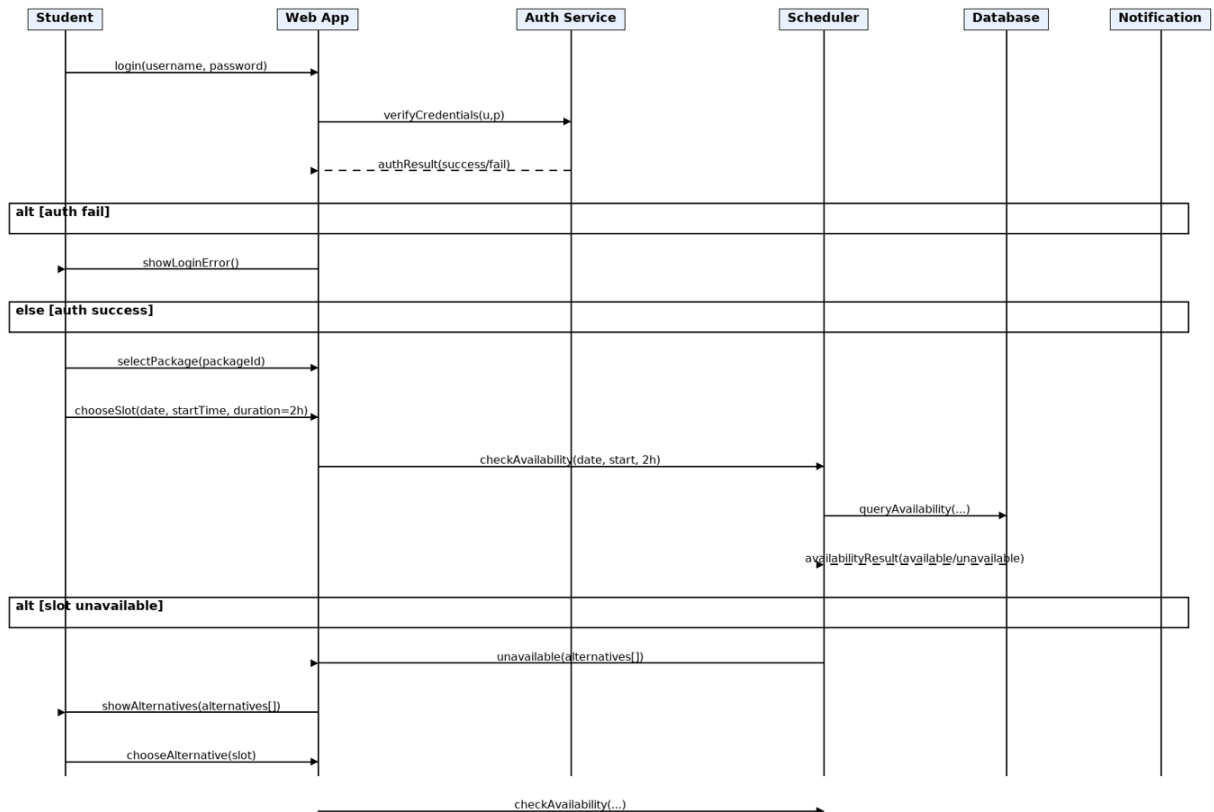


Figure 4. System interactions for scheduling a lesson: authentication, availability check, reservation, and confirmation.

UML Class Diagram

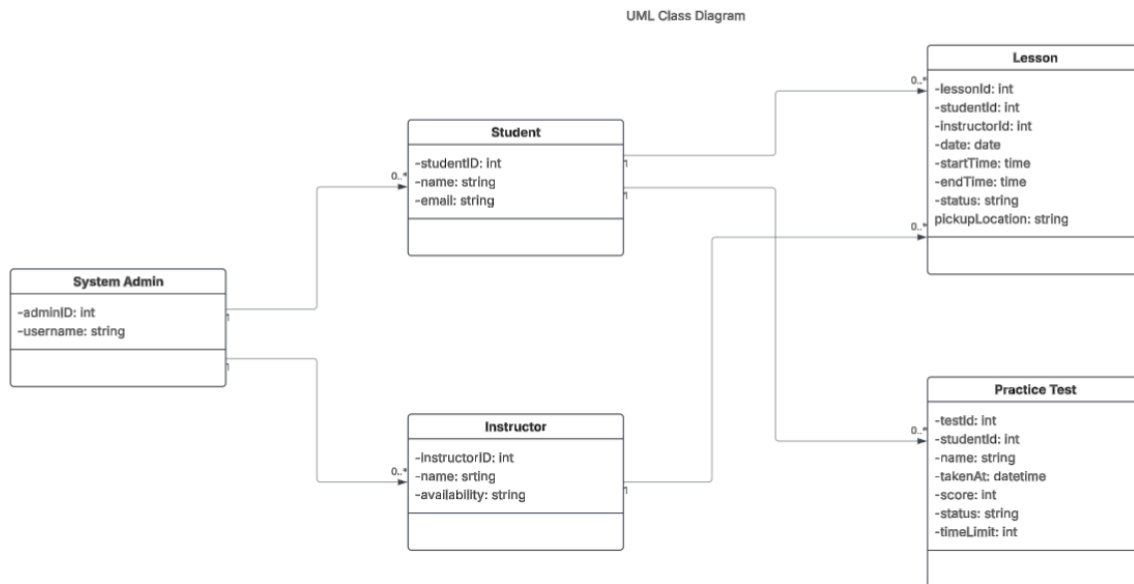


Figure 5. Core data model. (Student, Instructor, Lesson, Practice Test, System Admin) with relationships used for scheduling and tracking progress.

Technical Requirements

Functional Requirements

The DriverPass system shall provide an integrated web-based platform enabling students to access online practice driving tests, schedule on-the-road training sessions, and track their progress. Students will be able to create accounts, securely log in, and reset their passwords automatically if forgotten. The system shall allow customers to select from three training packages:

- **Package One:** Six hours of in-car training
- **Package Two:** Eight hours of in-car training plus an in-person DMV rules and policies lesson
- **Package Three:** Twelve hours of in-car training, in-person DMV rules and policies lesson, and full access to online courses and practice exams

Each driving session will be two hours, and students may schedule lessons online, by phone, or in person with the secretary. The system shall enable instructors to view and manage their schedules, log driver comments, and update student records. Administrators shall have the capability to manage users, modify account access, monitor system usage, and generate detailed activity reports showing who created, modified, or canceled reservations. The system shall connect to the DMV to receive automated updates on policy and testing changes, triggering internal notifications when updates are available. The system shall support modern web browsers (Chrome, Firefox) on desktop and mobile devices with responsive UI layouts to maintain usability across screen sizes.

Nonfunctional Requirements

The system shall be cloud-hosted to ensure high availability and accessibility from any device with an internet connection, supporting major browsers on both desktop and mobile platforms. Performance requirements include page loads of under two seconds under normal operating loads and system uptime of at least 99.9%. The back end will utilize a secure relational database, such as MySQL or PostgreSQL, to store structured data, and will employ TLS encryption for all data transmission. The system shall be updated quarterly to incorporate security patches and policy changes, with the ability to scale infrastructure to handle peak demand periods such as DMV testing seasons. The system design also supports horizontal scaling, allowing DriverPass to expand services to additional instructors, students, and DMV integrations without requiring major architectural changes.

Rationale:

A cloud-hosted architecture is selected to reduce the client's responsibility for maintenance and backups, aligning with DriverPass's request to minimize technical overhead. A relational database supports structured scheduling, package, and user data with enforced constraints to ensure accuracy. TLS encryption and password hashing address the security concerns expressed in the interview, while quarterly updates ensure compliance with evolving DMV standards. This design directly aligns with DriverPass's stated priorities of cloud-based access with minimal maintenance burden and timely incorporation of DMV rule updates, ensuring that both instructional content and scheduling remain current and reliable.

Security

- **User Authentication:** All users must log in with a unique username and password, with passwords stored using salted hashing (e.g., bcrypt).
- **Access Control:** Role-based access ensures students, instructors, admins, and secretaries only see and modify data relevant to their duties.
- **Encryption:** TLS 1.3 is required for all data in transit; sensitive fields (e.g., credit card numbers) are encrypted at rest using AES-256.

- **Brute Force Prevention:** After five consecutive failed logins, the account is temporarily locked for 15 minutes, and the attempt is logged; repeated lockouts trigger an admin alert. Admins can manually reset or unlock accounts after identity verification.
- **Audit Logging:** All user actions (create, modify, cancel reservations, update records, login attempts) are recorded with timestamps for reporting.
- **DMV Data Integrity:** Automated DMV updates are verified with digital signatures before integration.

System Limitations

While the system provides comprehensive training and scheduling capabilities, it relies on consistent internet connectivity for all transactions other than offline viewing of downloaded reports. Initial implementation will not include the ability for administrators to add or remove training packages without developer assistance, though they may disable packages. Hardware such as mobile devices and instructor computers is assumed to meet minimum browser requirements. Budgetary constraints limit the inclusion of advanced analytics and AI-based scheduling in the initial release, though these may be considered for future updates.

References

Lucid Software. (n.d.). *Lucidchart tutorial*. Lucidchart. <https://www.lucidchart.com/pages/uml-diagram>

National Institute of Standards and Technology. (2020). *Digital identity guidelines* (NIST Special Publication 800-63B). U.S. Department of Commerce. <https://doi.org/10.6028/NIST.SP.800-63b>

Open Web Application Security Project. (2021). *OWASP cheat sheet series: Password storage*. OWASP Foundation. https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

PostgreSQL Global Development Group. (n.d.). *PostgreSQL 15 documentation*.
<https://www.postgresql.org/docs/>

TechTarget. (2023). *Transport Layer Security (TLS)*. In *SearchSecurity*.
<https://www.techtarget.com/searchsecurity/definition/Transport-Layer-Security-TLS>