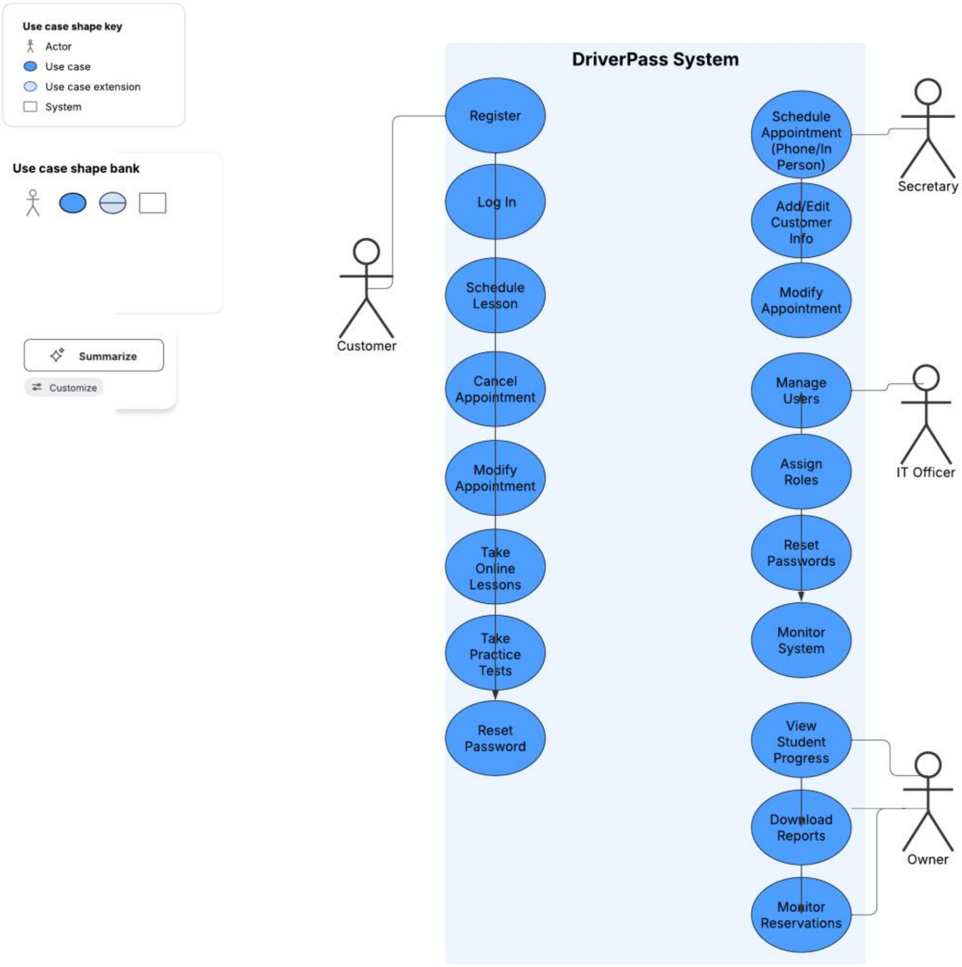
System Design Document: DriverPass

1. Use Case Diagram

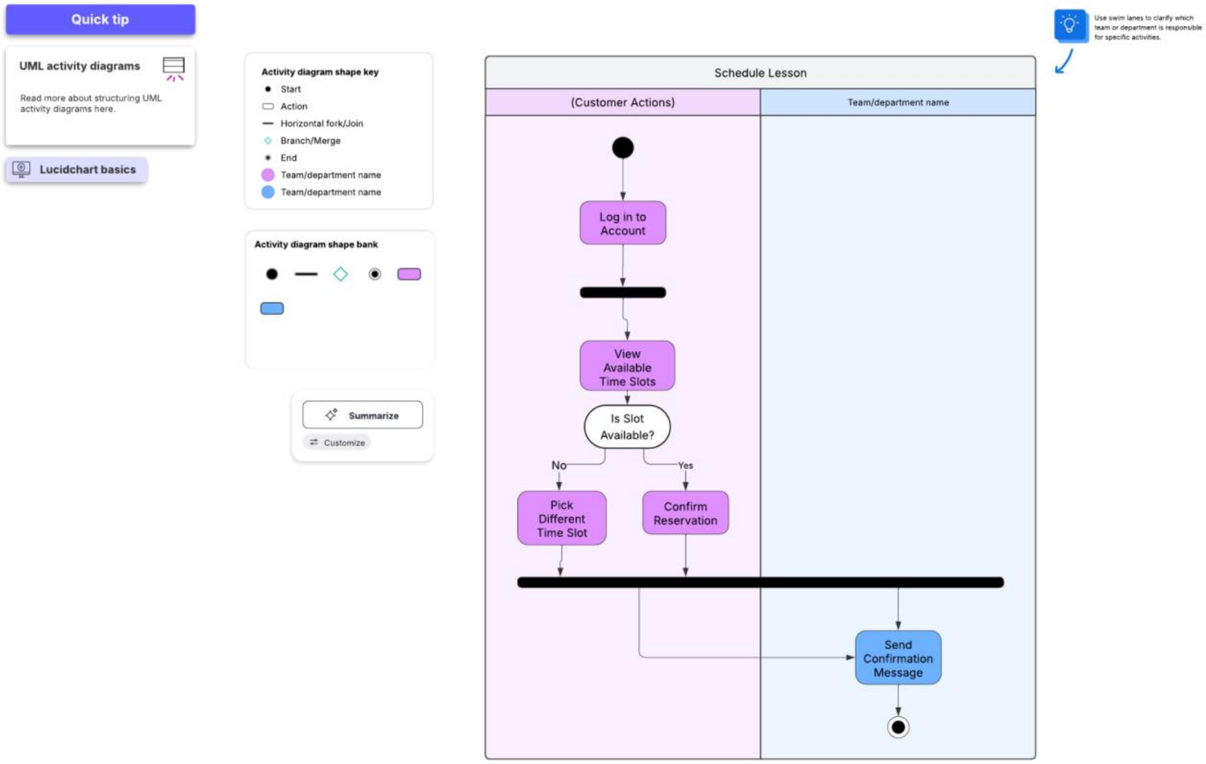
This use case diagram captures the core functionalities of the DriverPass system. Key actors include the Customer, Secretary, IT Officer, and Owner. Actions include registering, logging in, scheduling and canceling appointments, accessing online lessons and practice tests, and generating reports. The diagram supports all necessary interactions and roles described in the requirements.



2. Activity Diagrams

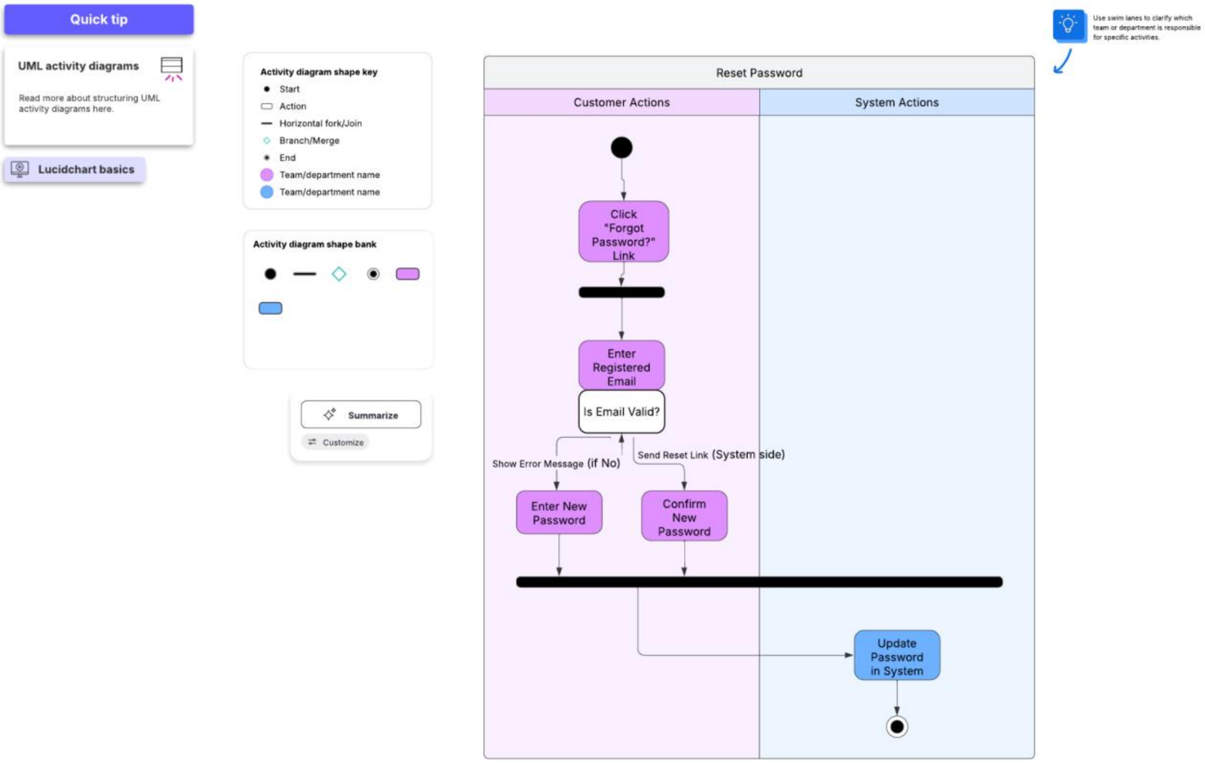
Activity Diagram 1: Schedule Lesson

This activity diagram outlines the process a customer follows to schedule a lesson. After logging in, the user views available slots, selects a package, chooses an instructor and vehicle, and confirms the appointment. If the selected time is unavailable, the user is prompted to try again.



Activity Diagram 2: Reset Password

This diagram illustrates the password reset workflow. Users request a password reset by entering their email. If valid, the system sends a reset link. After clicking the link, the user enters and confirms their new password.

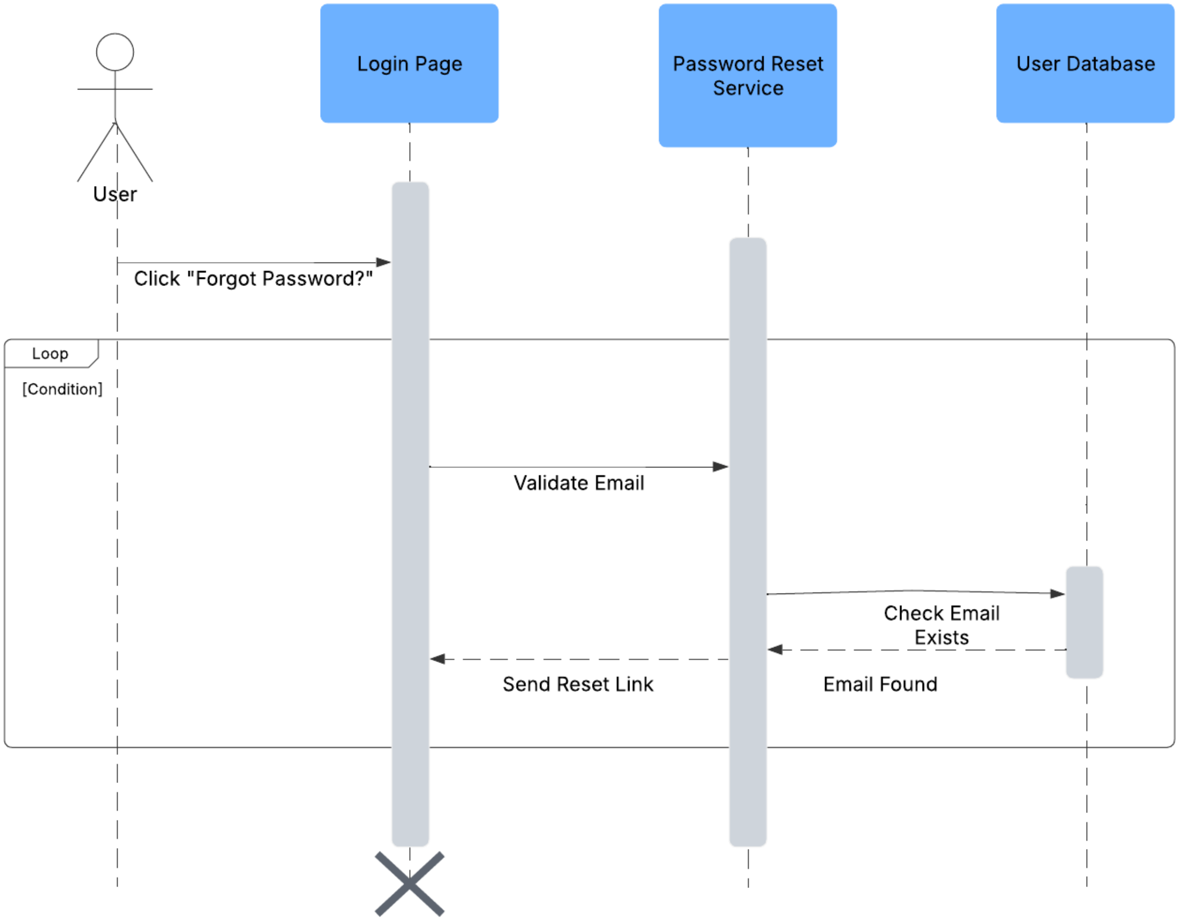


Read more about structuring UML activity diagrams here.

Use swim lanes to clarify which team or department is responsible for specific activities.

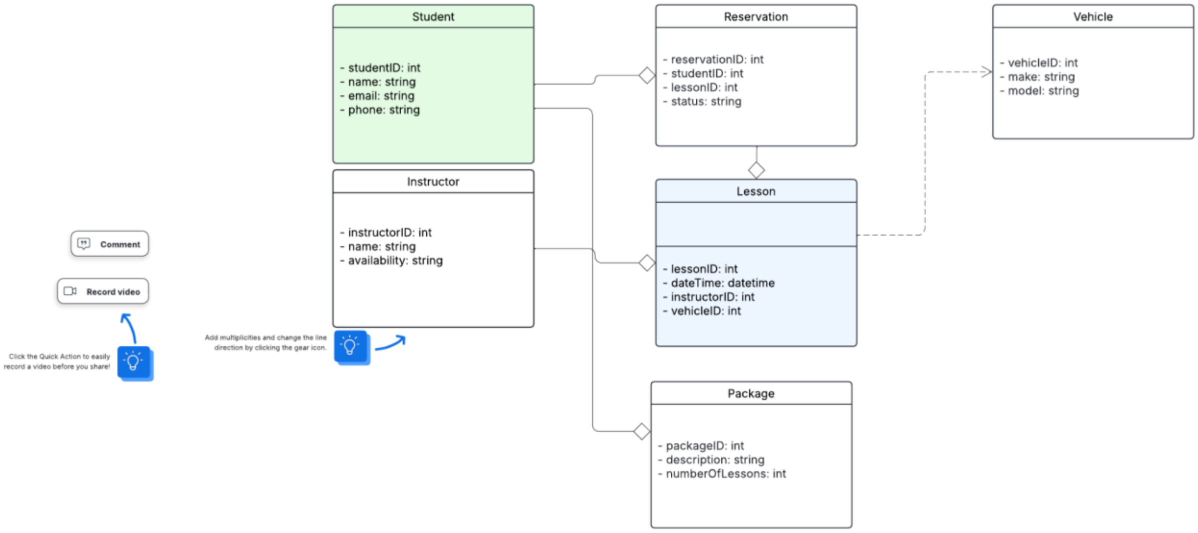
3. Sequence Diagram

The sequence diagram shows how system components interact during the password reset process. The user interacts with the login page, which communicates with the password reset service, the user database, and the email system. Each step aligns with the business rules and ensures security and user feedback.



4. Class Diagram

This UML class diagram reflects the data structure and associations in the DriverPass system. It includes classes such as Student, Lesson, Reservation, Instructor, Vehicle, and Package. Relationships represent how students are linked to packages and reservations, and how lessons connect with instructors and vehicles.



5. Technical Requirements

Hardware Requirements

- Cloud-hosted servers with at least 8 GB RAM and 4-core CPU
- Secure database hosting (e.g., AWS RDS, Google Cloud SQL)
- Reliable email delivery infrastructure

Software Requirements

- Frontend: HTML/CSS/JavaScript (mobile-friendly UI)
- Backend: Python (Flask) or Node.js
- Database: MySQL or PostgreSQL
- Authentication: OAuth 2.0 / JWT for secure logins
- Diagramming Tools: Lucidchart

Tools

- IDE: VSCode or PyCharm
- Source Control: Git/GitHub
- Email API: SendGrid or Amazon SES
- Issue Tracking: Trello or Jira

Infrastructure

- Runs on cloud platforms (AWS, Azure, GCP)
- SSL/TLS encryption for all communication
- Role-based access controls (admin, secretary, IT officer, customer)
- Regular cloud-based backups and security patches

System Limitations

- System currently only supports English
- New modules require developer involvement
- Password resets are handled only via the online system
- Internet access is required for all users