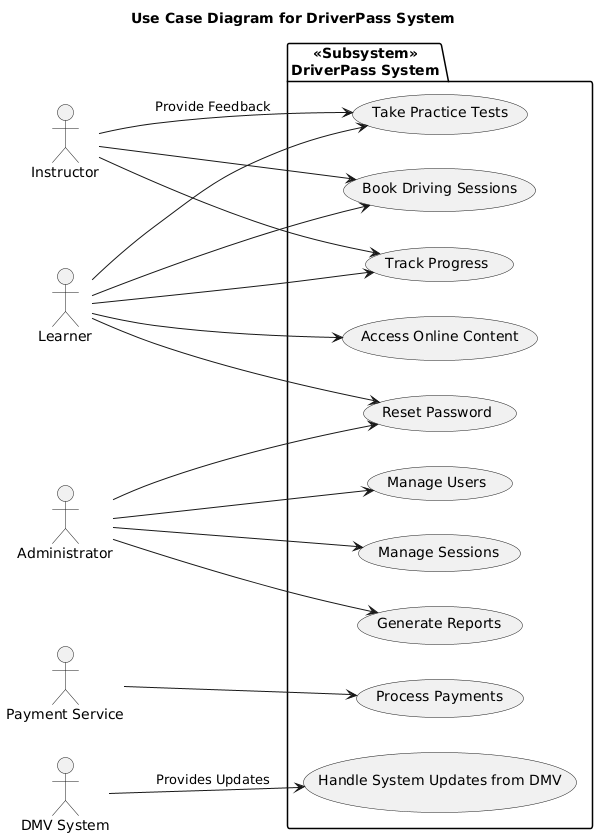
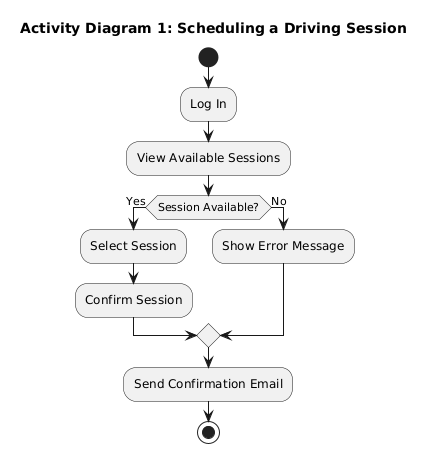
# CS 255 System Design Document

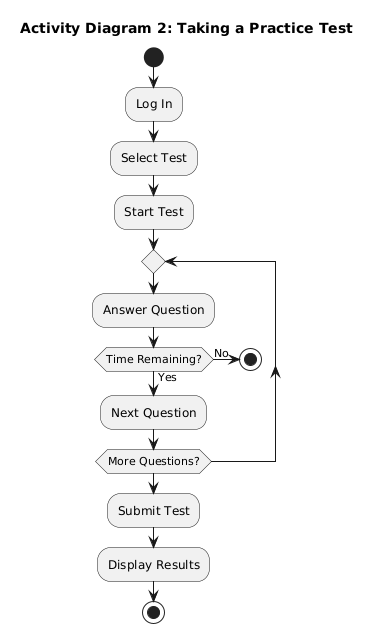
## UML Diagrams

### UML Use Case Diagram

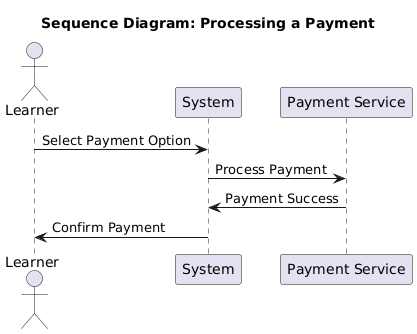


### UML Activity Diagrams

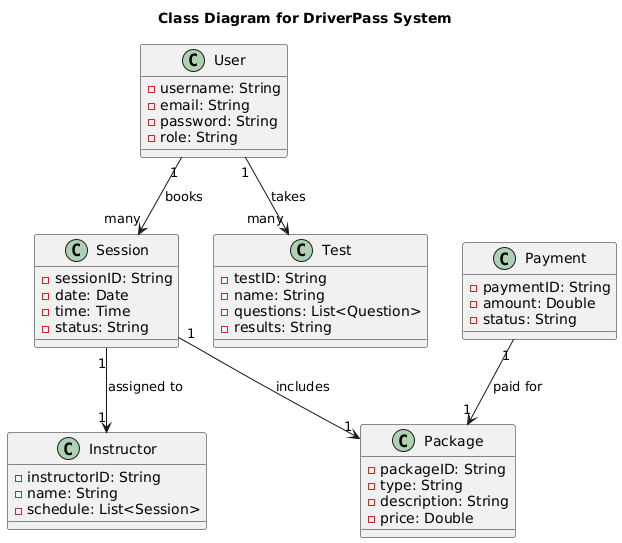




### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements for the DriverPass System

The **DriverPass** system aims to provide a seamless platform for learners preparing for their driving test, offering services ranging from booking driving lessons to taking practice tests. To ensure the system performs optimally and meets the specified needs in the **Business Requirements Document**, the following technical requirements have been outlined. These requirements cover the necessary **hardware**, **software**, **tools**, and **infrastructure** to support the system's functionality, scalability, security, and performance as the system evolves.

**1. Hardware Requirements**

The **DriverPass** system must be accessible on both **web** and **mobile platforms**, ensuring smooth user interaction across multiple devices. Below are the necessary hardware specifications for both the user and server environments.

**User Devices (End-User Hardware)**

**Mobile Devices**:

* + **Android** and **iOS** smartphones and tablets (screen sizes ranging from 4.7" to 10”).

**Desktop/Laptop Devices**:

* + Systems running **Windows**, **macOS**, or **Linux** with modern browsers (Chrome, Firefox, Safari, Edge).

**Administrative Access (IT Staff)**:

* + IT administrators and support staff will use desktops or laptops configured for system management, database interfaces, and administrative tools.

**Minimum Hardware Configuration**:

* + **Processor**: Intel Core i5 or equivalent
  + **RAM**: 8GB or more
  + **Storage**: 256GB SSD
  + **Graphics**: Integrated graphics supporting 1080p resolution
  + **Network**: High-speed internet access (100Mbps or higher)

**Server Hardware (For Hosting)**

The backend infrastructure will be hosted in the cloud for scalability and availability. A cloud-based solution such as **Amazon Web Services (AWS)**, **Google Cloud Platform (GCP)**, or **Microsoft Azure** will ensure flexibility and growth potential.

**Server Configuration**:

**CPU**: 4 vCPU or higher

**RAM**: 16GB or more

**Storage**: 1TB SSD or more (for database and educational content storage)

**Network**: 1Gbps internet connection for low latency and high throughput

**2. Software Requirements**

**Backend Software**

**Operating System**:

* + **Linux** (Ubuntu or CentOS) for stability, security, and cost-effectiveness.

**Database Management System (DBMS)**:

* + **Relational Database**: **PostgreSQL** or **MySQL** to handle structured data like user profiles, session details, and payment records.
  + **Non-relational Database** (for scalability): **MongoDB** for unstructured data like logs, feedback, and test results.

**Backend Framework**

* + **Node.js** or **Django (Python)** for building efficient APIs, user authentication, session management, and handling interactions with external systems such as the DMV for updates.

**Frontend Software**

**Web Application**:

* + **HTML5**, **CSS3**, and **JavaScript** for a responsive and dynamic web experience.
  + **React** or **Angular** for creating a fluid and responsive user interface.
  + **Bootstrap** or **Tailwind CSS** for responsive design, ensuring a smooth experience on various devices.

**Mobile Application**:

* + **React Native** or **Flutter** for cross-platform mobile app development (iOS and Android).
  + **Android Studio** and **Xcode** for platform-specific customizations.

**APIs**:

* + **RESTful APIs** for front-end and back-end communication, using **Express.js** (Node.js) or **Django REST Framework**.
  + **GraphQL** (if needed) for more flexible data fetching, especially for complex user dashboards or progress tracking.

**Security Software**

**SSL/TLS Encryption**: All data exchanges between the client and server will be encrypted using **SSL/TLS** to protect sensitive information like user credentials and payment details.

**OAuth 2.0 / JWT (JSON Web Tokens)**: For secure authentication and authorization, ensuring proper access control across different user roles.

**Two-Factor Authentication (2FA)**: A secondary layer of authentication will be added to prevent unauthorized logins, with authentication apps or email-based one-time passwords (OTP).

**3. Infrastructure Requirements**

**Cloud Hosting and Services**

**Cloud Infrastructure**: The system will be hosted on cloud platforms such as **AWS**, **Microsoft Azure**, or **Google Cloud Platform** for flexibility and scalability.

* + **Elastic Compute Cloud (EC2)** instances will provide scalable server capacity.
  + **Amazon RDS** or **Azure SQL Database** for database hosting.
  + **Cloud Storage** (AWS S3 or Azure Blob Storage) for educational content and user data.

**Content Delivery Network (CDN)**

**AWS CloudFront** or **Cloudflare CDN** will optimize the delivery of static content (e.g., images, videos, tutorials), ensuring fast load times and low latency globally.

**Load Balancer and Auto-scaling**

**Elastic Load Balancing (ELB)** and **auto-scaling** groups will ensure the system can scale to handle traffic surges without compromising performance.

**Backup and Recovery**

**Automated Backup Systems** using **AWS Backup** or **Azure Backup** will regularly back up user data and system configurations to ensure recovery in the event of system failure.

**Disaster Recovery**: The system will employ multi-region deployment and failover strategies to minimize downtime and ensure business continuity.

**4. Tools and Development Frameworks**

**Version Control**: **Git** (with **GitHub** or **GitLab**) will be used for source code management and collaborative development.

**Project Management Tools**:

* + **Jira** for issue and task tracking, ensuring efficient management of features, bugs, and progress.
  + **Confluence** or **Notion** for documenting system design, requirements, and project progress.

**CI/CD Pipeline**:

* + **Jenkins**, **GitLab CI**, or **GitHub Actions** will be used to automate testing, integration, and deployment, ensuring continuous delivery of new features and bug fixes.

**5. Network and Communication Infrastructure**

**API Communication**: The system will utilize **RESTful APIs** for efficient communication between the front-end and back-end. External integrations, such as updates from the DMV or payment services, will also use these APIs.

**WebSockets** will be employed for real-time communication, enabling live updates for features like lesson availability and user progress tracking.

**Secure Messaging**: For communication between learners, instructors, and administrators, **Socket.IO** or **Firebase Cloud Messaging (FCM)** will be integrated to facilitate real-time notifications and messages.

**6. Compliance and Regulatory Requirements**

**Data Privacy**

The system will comply with data privacy regulations such as **GDPR** (General Data Protection Regulation), **CCPA** (California Consumer Privacy Act), and any other local data privacy laws, including:

* **Encryption of sensitive data** (passwords, payment details).
* **User consent management** for data processing and analytics.
* Ensuring the **right to access** and **delete** personal data.

**Payment Compliance**

The integration with third-party payment services will comply with **PCI-DSS** standards to ensure secure handling of credit card transactions.

**Conclusion**

The **DriverPass** system is built with scalability, security, and high performance in mind. It requires a modern infrastructure, incorporating cloud hosting, secure APIs, and robust data management systems. The design ensures accessibility across web and mobile platforms, making it easy for learners, instructors, and administrators to interact with the system. By adhering to these technical requirements, **DriverPass** will provide a secure, efficient, and reliable solution for driving test preparation, with potential for future expansion as user needs grow.