**Project Two**

Nathalie Morales

CS-255: System Analysis and Design

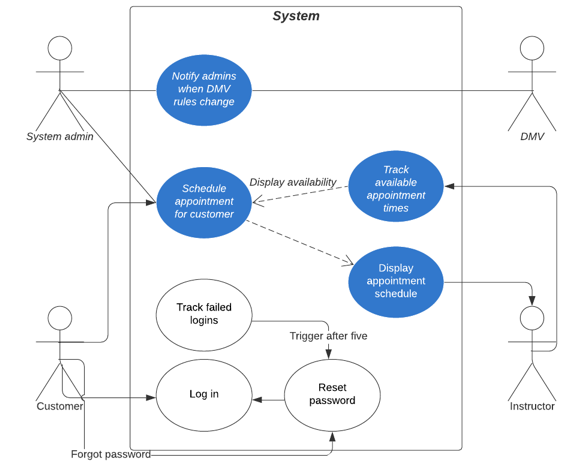
Southern New Hampshire University

04/26/2024

## UML Diagrams

### Use Case Diagram

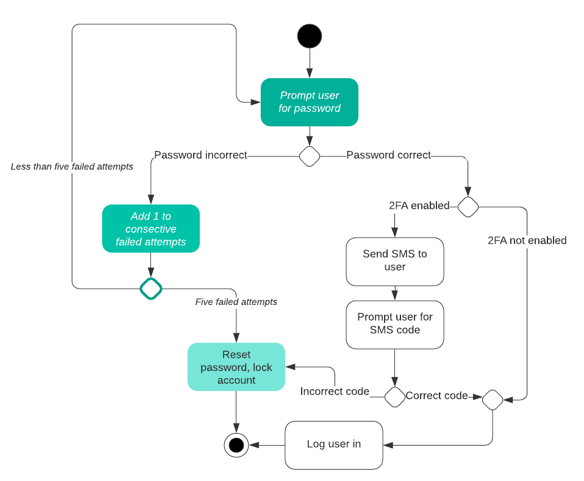
In the use case diagram, the process of interactions among actors (Boss, IT officer, secretary, students) and the system components will be modeled. It will involve cases like user registration, booking driving classes, attending virtual courses, and managing appointments.



### UML Activity Diagrams

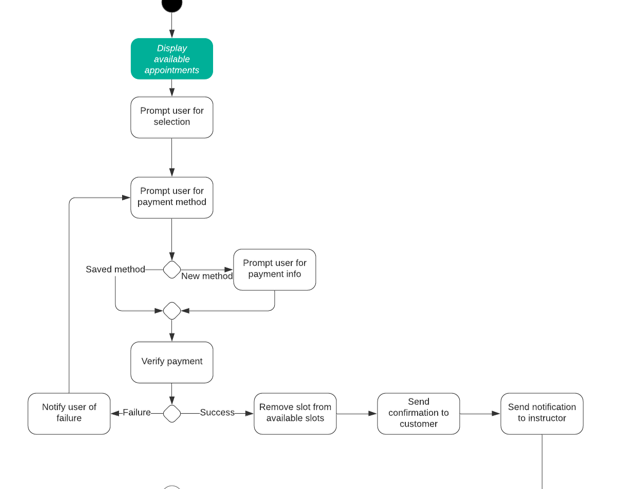
**Activity Diagram 1: Book**

These Diagrams will display the step-by-step process for booking a course. It will consist of type operations: choosing the course package, picking out the instructor, defining the date and time, and ensuring the booking has been confirmed.

**

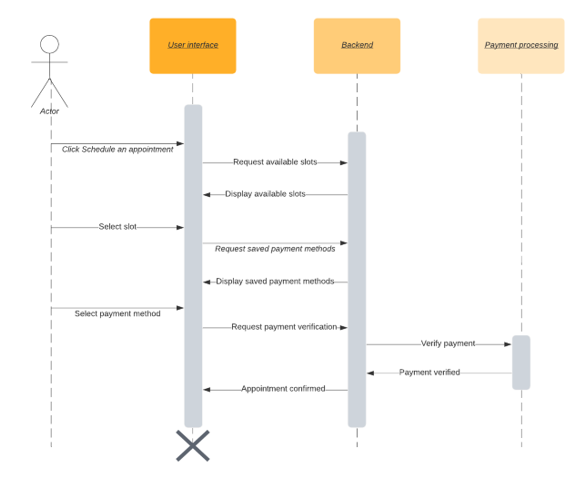
This flowchart will explain the sequence of the activities run when conducting synchronous or asynchronous classes using a virtual learning environment (Sundaramoorthy, 2022). In this paragraph, we will run through the activities like creating the course materials, putting the class on schedule, giving the lecture, and grading.

**Activity Diagram 2: Flow Chart**



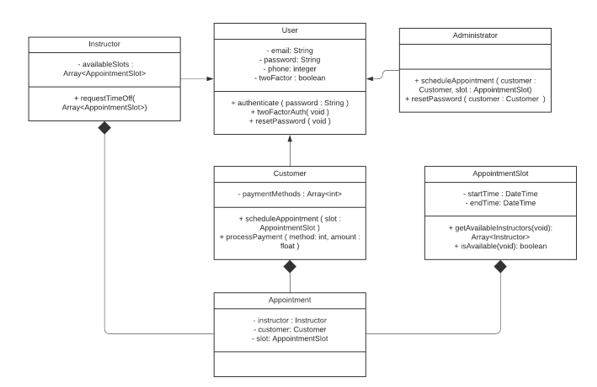
### Sequence Diagram

The sequence diagram will display the communications between the objects and components in a particular case, e.g., making a driving course reservation or attending a webinar. It will illustrate the messages passed between the objects to complete the operation.



### UML Class Diagram

The class diagram will represent the logically structured DriverPass system comprising classes, attributes, and relationships. It will consist of classes, such as User, Course, Appointment, Instructor, and Secretary, with their attributes and interconnections.

**

## Technical Requirements

**Hardware Requirements**

1. Standard computing hardware for server deployment.
2. Mobile devices (phones, tablets) and computers for client access.

**Software Requirements**

1. Web server software (e.g., Apache, Nginx).
2. Database management system (e.g., MySQL, PostgreSQL).
3. Programming languages (e.g., Java, Python) for backend development.
4. Frontend development frameworks (e.g., React, Angular) for the user interface.
5. Security software for data encryption and protection.

**Tools**

1. Integrated Development Environment (IDE) for software development.
2. Version control system (e.g., Git) for collaborative coding.
3. UML modeling tools for diagram creation and documentation.

**Infrastructure**

1. Cloud hosting platform (e.g., AWS, Azure) for scalability and reliability.
2. Internet connectivity for online access to the system.
3. Regular backups and disaster recovery mechanisms for data protection.

**CS 255 Business Requirements**

**System Components and Design**

**Purpose**

* The objective of this endeavor is to construct a tool in DriverPass that maximizes comprehension with the assistance of online practice tests and on-the-road driving practice in order to fulfill the requirements for passing the driving test.
* Client: DriverPass

Client's System Requirements:

* Opportunity to attend virtual lectures and learning aids.
* Instead of relying on weekend practices, we will offer personalized in-car training whenever possible.
* User visibility controls both the web world and the physical world.

**System Background**

DriverPass is focused on delivering comprehensive training programs to allow test takers to pass their driving tests, which is the core problem in the market today.

Components Needed:

* Learning online and exam sessions.
* Scheduling meetings for in-car teaching.
* The data could be accessed securely from any location

**Objectives and Goals**

Upon completion, the system should:

* Teach the theory through web-based classes and assess your progress with practice tests.
* Enable field learning based on the established schedule with users.
* Secure data both in context and content as well.
* Allow trainees to choose what they would like to be trained.

**Requirements**

**Nonfunctional Requirements**

**Performance Requirements**

* The system should be integrated into the web platforms.
* System speed: The immediate response time for a great user experience.
* Regular updates: Accordingly, the system should be reformed constantly to keep pace with facts

**Platform Constraints**

* Platforms: Web-based platform reachable from a phone, tablet, or laptop.
* Backend tools: The database is essential to the app`s features

**Accuracy and Precision**

* User distinction: Every user shall be identified by one.
* Input: The capability of data input is to be typed in the correct case memory type.
* Problem notification: The system must send the admin a quick call within a few minutes in case any error is found

**Adaptability**

* User changes: Drive people to web-based apps by allowing customers to change their personal information without coding.
* Platform updates: The system should follow the work processes of the associated platforms.
* IT admin access: System administration is in complete control by the administration team for efficient functioning

**Security**

* User login: A secure login procedure must be developed so that the user can authenticate their identity.
* Data exchange: Secure communication channels through SSL, TLS, or other encryptions.
* Brute force protection: Locking one's account after entering the wrong PIN for times.
* Password recovery: Granting password resets to users who forgot their passwords.

**Functional Requirements**

The system shall:

* Validate user names and passwords when users try to log in.
* Give people the possibility to book a driving course. Listen to the given audio and repeat the given sentence.
* Conduct virtual classes instead. Do a test to measure the progress.
* Log user activity reports and other statistics.
* Show driving course details in the to-do list, including time, instructor name, and car information.
* Follow up with cancellation and modification notices about the lessons.

**User Interface**

User Interface Needs

* Users: Boss, IT Officer, Secretary, Students.
* Interface Needs:
  + Boss: Harmonize reports and update invoices.
  + IT Officer: Manage systems, configure email notifications, and reset passwords.
  + Secretary: Schedule the dates and client info going in.
  + Students: Attend classes and see your placement and the past lessons under your profile information.
  + Interface Interaction: Web-based interface that would allow anyone to access it from any smart device easily

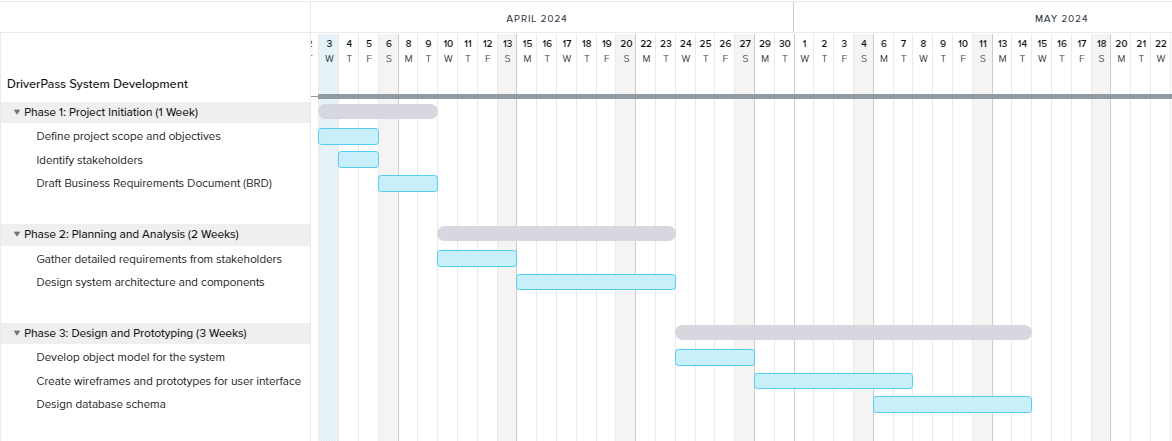
**Assumptions**

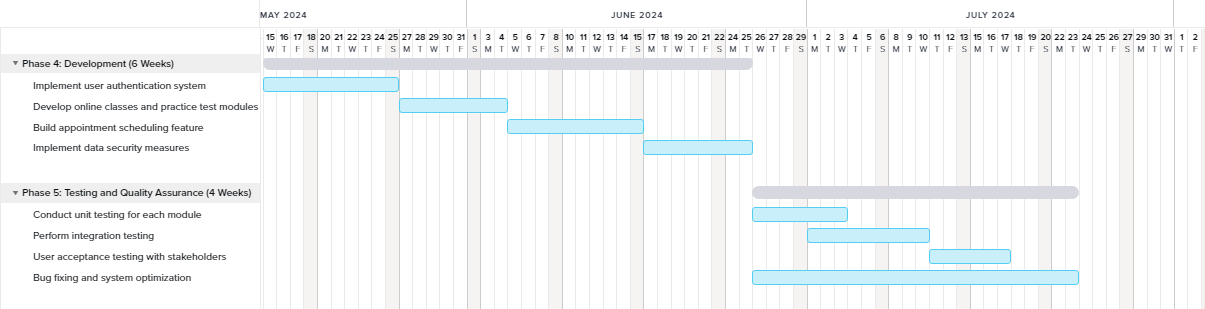
* End-users possess the rudimentary computer knowledge.
* Stability of internet connection in terms of online access.
* Regulation and Compliance with DMV Laws and Regulations.
* Either availability of a reasonable share of resources for the development of the systems

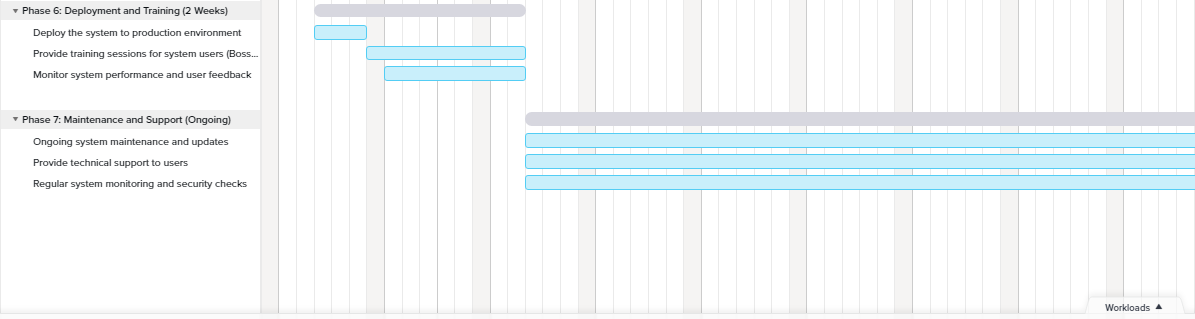
**Limitations**

* Developmental time for system development.
* Necessity of a division into chapters. In this phase, the line art designer shall manage the budget for the additional features.
* At the first launch, the product might not have all the functionalities.

**Gantt Chart**







**Reference**

Sundaramoorthy, S. (2022). UML diagramming: a case study approach. Auerbach Publications. <https://doi.org/10.1201/9781003287124>

**Appendix**

**Gantt Chart Description**

**Phase 1: Project Initiation (1 Week)**

* **Tasks:**
  + Define project scope and objectives
  + Identify stakeholders
  + Draft Business Requirements Document (BRD)
* **Duration:** 1 week
* **Start Date:** April 3, 2024
* **End Date:** April 9, 2024

**Phase 2: Planning and Analysis (2 Weeks)**

* **Tasks:**
  + Gather detailed requirements from stakeholders
  + Design system architecture and components
* **Duration:** 2 weeks
* **Start Date:** April 10, 2024
* **End Date:** April 23, 2024

**Phase 3: Design and Prototyping (3 Weeks)**

* **Tasks:**
  + Develop an object model for the system
  + Create wireframes and prototypes for the user interface
  + Design database schema
* **Duration:** 3 weeks
* **Start Date:** April 24, 2024
* **End Date:** May 14, 2024

**Phase 4: Development (6 Weeks)**

* **Tasks:**
  + Implement a user authentication system
  + Develop online classes and practice test modules
  + Build an appointment scheduling feature
  + Implement data security measures
* **Duration:** 6 weeks
* **Start Date:** May 15, 2024
* **End Date:** June 25, 2024

**Phase 5: Testing and Quality Assurance (4 Weeks)**

* **Tasks:**
  + Conduct unit testing for each module
  + Perform integration testing
  + User acceptance testing with stakeholders
  + Bug fixing and system optimization
* **Duration:** 4 weeks
* **Start Date:** June 26, 2024
* **End Date:** July 23, 2024

**Phase 6: Deployment and Training (2 Weeks)**

* **Tasks:**
  + Deploy the system to the production environment
  + Provide training sessions for system users (Boss, IT Officer, Secretary, Students)
  + Monitor system performance and user feedback
* **Duration:** 2 weeks
* **Start Date:** July 24, 2024
* **End Date:** August 6, 2024

**Phase 7: Maintenance and Support (Ongoing)**

* **Tasks:**
  + Ongoing system maintenance and updates
  + Provide technical support to users
  + Regular system monitoring and security checks
* **Duration:** Ongoing
* **Start Date:** (and ongoing)