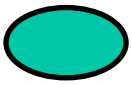
Southern New Hampshire University

CS 255 System Design Document Template

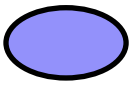
By: Dathan Pompa

UML Diagrams

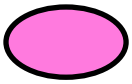
UML Use Case Diagram



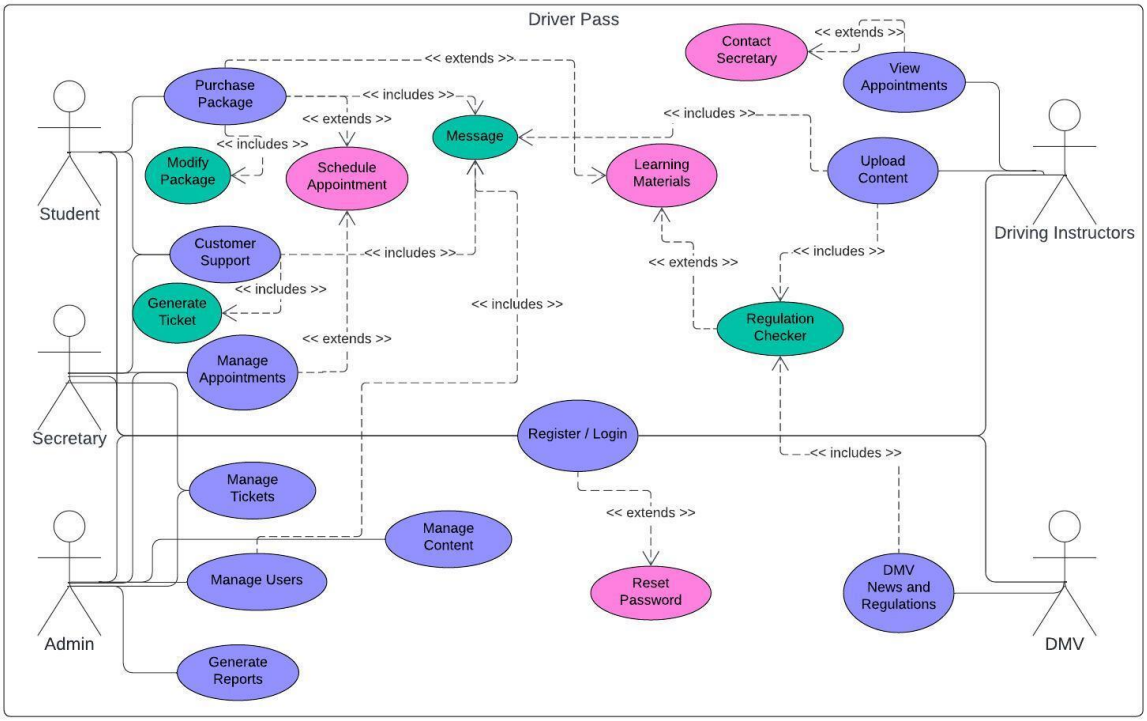
Use case accessed with includes



Use case



Use case accessed with extends

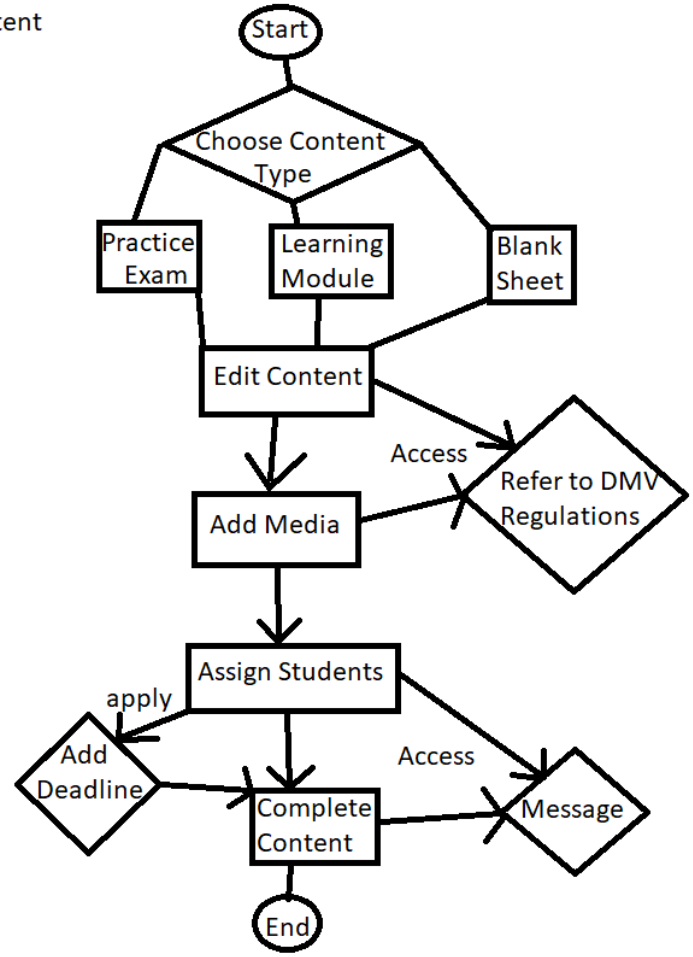


In the above use case diagram, we have 5 actors that help the system function properly. Our students are going to have the ability to purchase a package which will let them schedule an appointment for behind-the-wheel training. They can also access learning materials, message their driving instructor, and contact customer support. The secretary is going to manage whenever a student schedules a driving instructor, will assist the customer with questions they

may have, can create tickets if the help they need is going to need admin approval, and they can also manage the tickets that have been created. Administrators are going to have the ability to generate reports, can manage the users that are on the platform, manage the content that is currently available to the students, manage the tickets that have been created, and they will have the ability to manage the appointments if they choose to do so. The driving instructor will be able to upload content which can be verified by our content checker, viewing appointments for the month which they can also contact the secretary in case they need to change any appointments made, and they can message students if they have any questions or need extra help. Finally, the DMV actor will have access to the current DMV news and regulations which also gives them ability to access the regulation checker, this lets them verify any content made by the driving instructor is aligned with DMV rules.

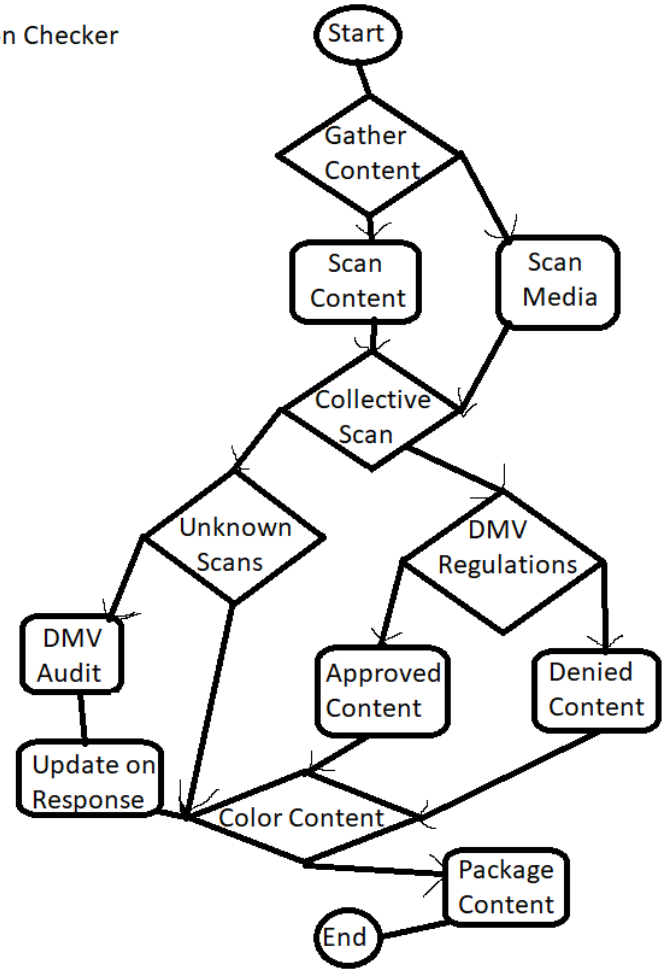
UML Activity Diagrams

Upload Content



The use case above shows how a user goes about the process of uploading content. They start by choosing what type of content they want to upload. After their choice, they have an option to edit and add media to their content. While doing this, they can refer to the DMV regulations so they can follow the guidelines easily. Once editing and adding media is done, the user will assign students to the content, after which they can add a deadline, message students or complete the content ready for it to be uploaded.

Regulation Checker

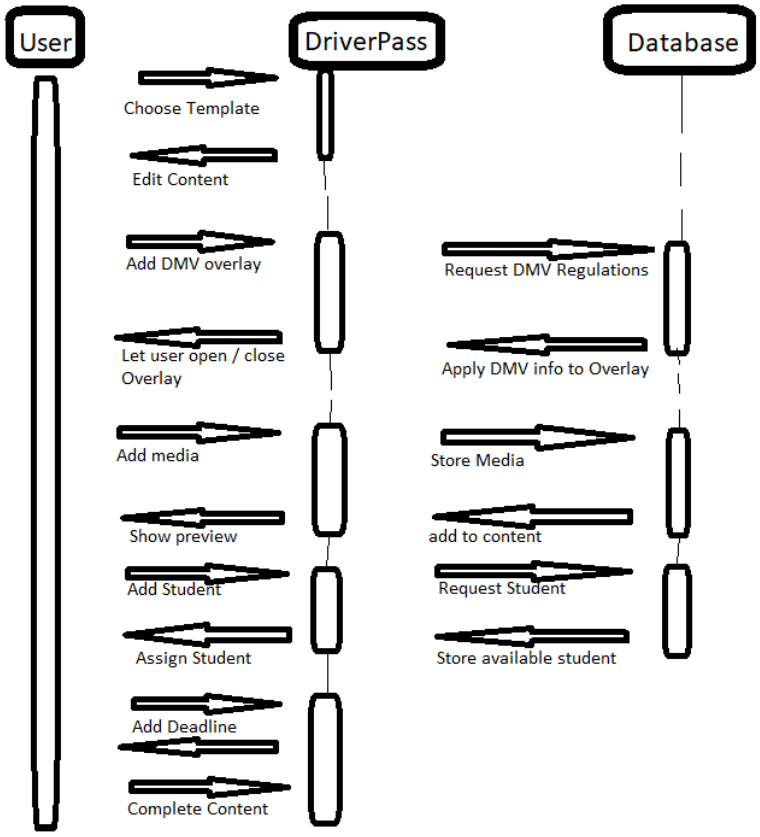


The use case for a regulation checker shows how DriverPass verifies all content made to ensure that it follows DMV rules and regulations. It starts with obtaining the content and doing a scan on its materials such as the content itself and any media. After it separates itself from the two, it will package it as a scan and send any unknown scans to the DMV to be audited. Once the collective scan has been made, it will be sent to the DMV regulation rules and verify the collective scan aligns with the DMV rules. After its scan, it will find its approved and denied content from the scan and highlight the content based on approved and denied content. It will also highlight any unknown content letting the user know it is waiting for the DMV to audit their

material. Once our program has color coordinated the highlighted content, it will package this with the content originally uploaded and complete its process.

UML Sequence Diagram

Upload Content

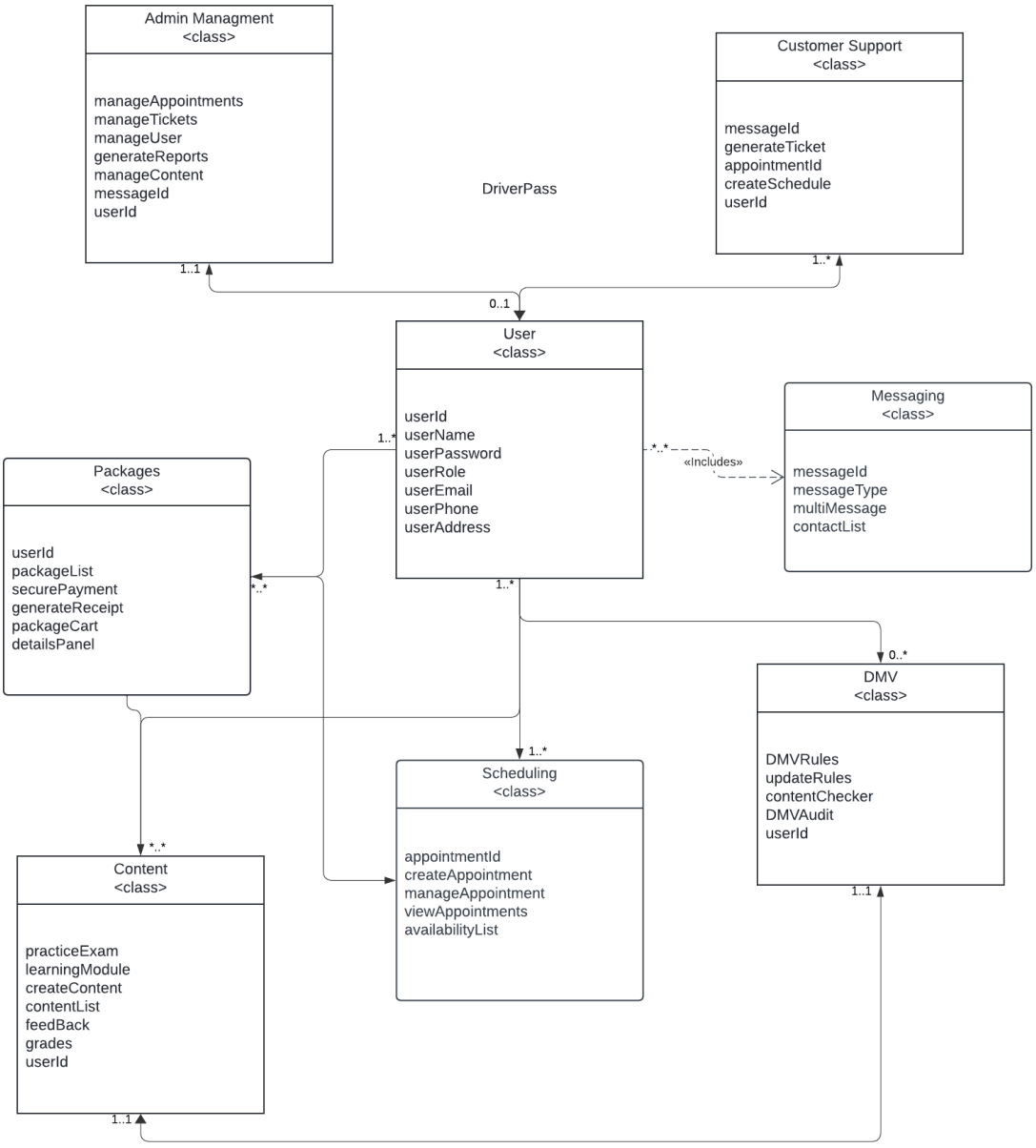


For my sequence diagram, it shows us which objects are getting interacted with on our Upload Media use case. To start, the user chooses a template, then they will edit their content. The user can add an overlay which sends a request to the database to pull the DMV regulations and send it back to the user so they can see the rules and regulations on the overlay while they are editing their content. The user can also add media if they wish to do so which sends the request to the DriverPass system, then it stores it in the database, and once it's stored, it will add it to the content they are working on and will display a preview of their media. After this, the user can add students' requests from the database and stores the available students on the

DriverPass system and lets the user assign the available students onto the content being created.

Once done, the user can add a deadline or complete their content.

UML Class Diagram



For my class diagram above, it shows how the different classes in my system will interact with one another. Think of classes as a way to define a certain action or number of actions that would be in constant use. The user class will store the information from the user and use that across the different classes in DriverPass. After user is made, certain classes are obtainable depending on the user's role, for a student the user can have access to the packages class which will let them purchase learning material and have access to schedule a behind the wheel driving lesson. The

behind the wheel driving lesson is obtainable from the user due to the scheduling class. The scheduling class is in charge of making appointments, checking for availability and managing current appointments. Each user has access to the Messaging class, which lets them send a message to another user or multiple messages, and you see the contact list of available users you can send one to. Students will have access to the driving instructors to get more help on their learning materials, driving instructors can message students and the secretary, the secretary has access to message students, teachers and administrators. The admin management class gives the admins the ability to manage users, tickets, content, and appointments, they can also generate reports. The customer support class will be for the secretary and for the student, this lets the user create a ticket based on what the need is, create a schedule for the driving instructors and lets them get appointment information. Finally, the DMV class will house the information that has the DMV rules and regulations, it lets them update the current rules, it includes the content checker for the driving instructors to utilize, and it lets the user have access to the DMV audit.

Technical Requirements

The DriverPass system that is proposed has a unique solution to giving students the additional help that will give them a better chance at passing the driver's test. A technical requirement for this system to operate is having a secure authentication and authorization method for maintaining security of information for our users. Another would be a real-time integration of the DMV, which is an external system, to allow our authentication checker for the content to be made to align with the DMV rules and regulations. One more technical requirement would be, the system should be able to handle many users interacting with the system at the same time, this ensures that students will not have their learning experience ruined when it comes to their practice exams or when they are trying to learn material.

Hardware

To solve these requirements DriverPass needs hardware, so they need to have a **server** to manage their data and host their application. DriverPass needs to have **workstations** for all of its employees, they will need equipment to make sure they have a **network**, so this includes having routers and switches connected to the server to allow for faster speeds. Lastly, having a **physical firewall device** connected to the routers will ensure that devices on the network have the most protection possible.

Software

That should be it for the hardware aspect, now let me talk about the different software needed. They will need a **server operating system** so that the server can function properly. I suggest Linux, they will need a **Database Management System** to manage the information being gathered and sent out of their program, I suggest MySQL. DriverPass will need **DMV Information software** that will be a third-party system, but it is needed to make sure DriverPass follows their rules and regulations. Finally a **Web Sever** will be needed, this will be required because we want the students to have a seamless interaction with learning or taking exams so a web server will ensure that their experience has little to no interruptions.

Tools

Let's talk about the different tools needed for DriverPass's system, for the developers to create and maintain the program they need an IDE named **Integrated Development Environment**. To let these developers collaborate with one another, they will need a **version control system**. Finally, they will need a **database design tool** which will help the developers or IT admins manage the database and design the structure of data being gathered and delivered.

Infrastructure

Finally, the last requirements for the system is going to include having a **network infrastructure** for the employees. The type they should have would be local so that their data would be safely stored on their own server. **Security** is going to be a big one as well, the processes they should include are user authentication and authorization, firewalls on their devices and network, data encryption, data masking, data backups, and data destruction. Following all these methods will help keep DriverPass safe from attackers. Finally for the DriverPass infrastructure, they need to have **internet connection** so that users can have the ability to generate reports from home, access data from outside their building and to allow students to freely enjoy the content they will be accessing.