The Student Companion

This project will bring students a convenient way to keep track of all their course's information, performance, and exam time tables. A web application will be developed to allow students to register with their email, and login to have access to their information. This system will offer students a way to organize and enter all their data that has to do with their courses. This will help organize and manage a student's information on their courses during their time in University.

The Student Companion will focus on mainly organizing academic activity such as time tables, storing exam dates, and keeping any notes or comments one may get after an assignment. Once the semester is finished they can enter their grade for a particular course so that a term GPA will be calculated to keep track of their progression. The data they create will be stored in a database so that they may use many devices. For our design students must register with an email to create an account. Once their account is confirmed they will be able to login, and access the information they have anywhere.

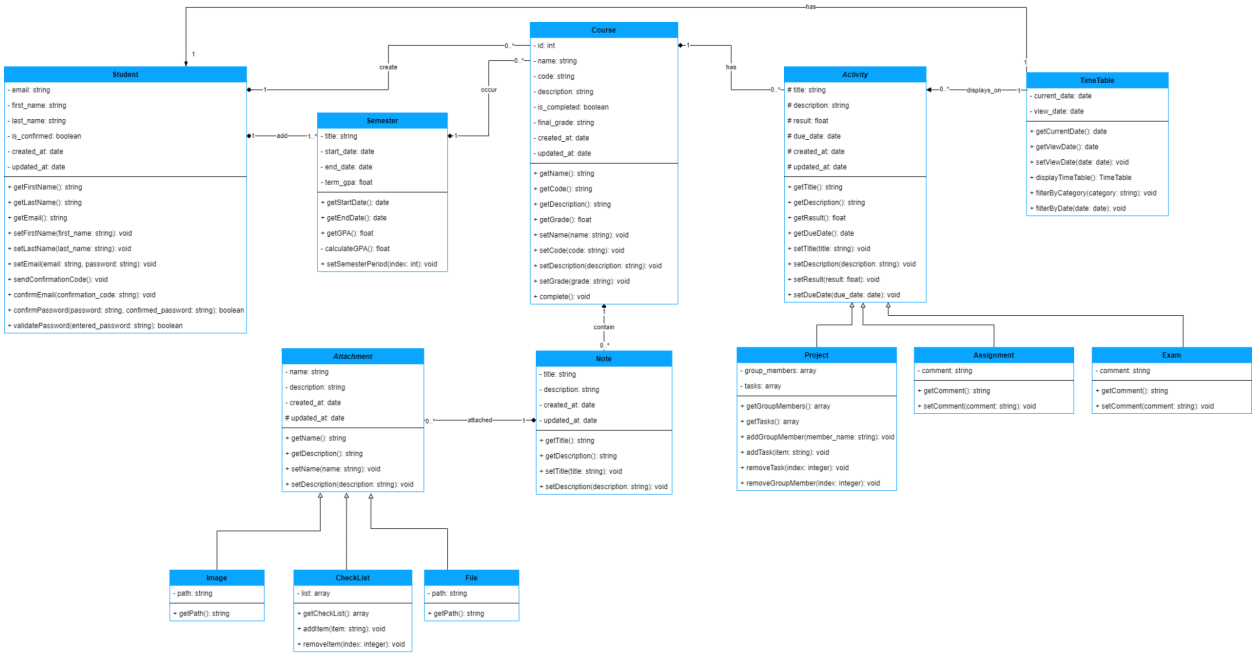
To build this web application we would need to use an application framework and a database. All the information created by students will be stored in a database for continuous use. This information will be created, read, updated and deleted, with the aid of the user interface which will be facilitated by a web application. This web application will make it possible to access it from anywhere needed. Version control such as GitHub will also be useful to keep track of progress and backup the code. For a framework we will use ASP.net for its ability to create Web Applications, and strong use of OOP concepts. MS Access will be used for the database also to keep inline with the stack so that the entire project is synchronized and compatible without too many issues.

Student Companion should be available through the web to allow students to have access on any device. Once they have a browser, Student Companion will be accessible on mobile, tablets ,and desktop browsers. This is important so that the many types of devices students may have access to can utilize this solution. A native mobile app, or desktop application may be too limiting, as not all students may have compatible laptops, or not all may have smart mobile devices, however, students may most likely have access to either or, and a browser.

UML

Class Diagram

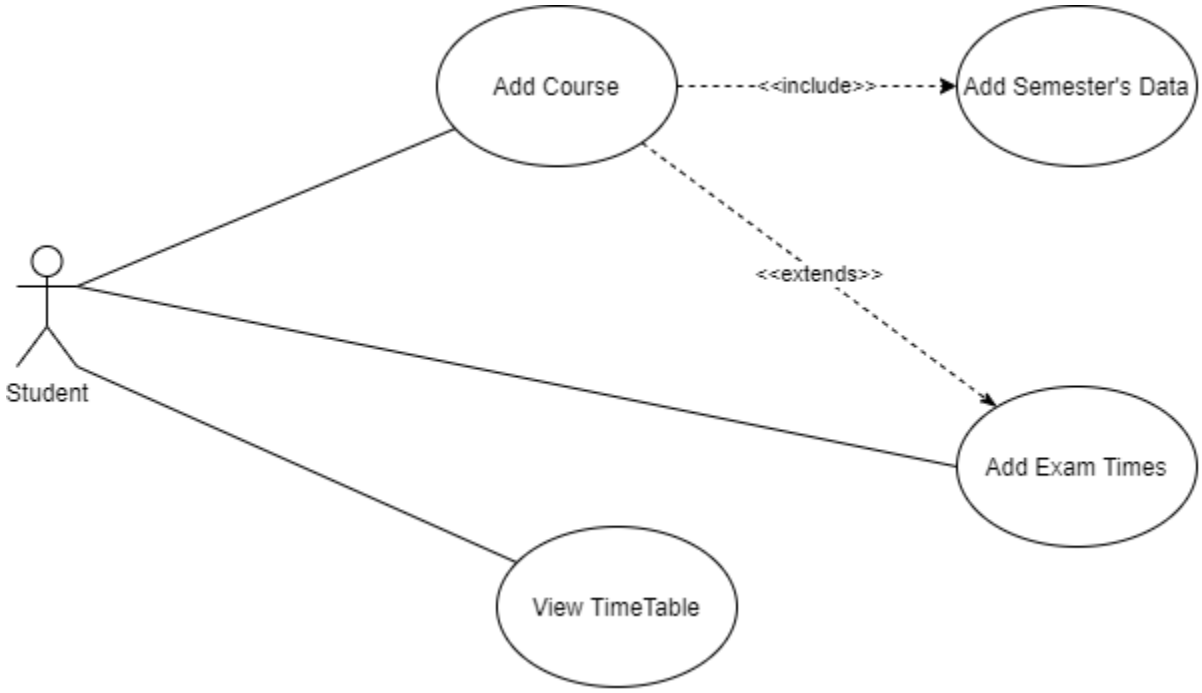
Shows the relationships between all classes within the application.



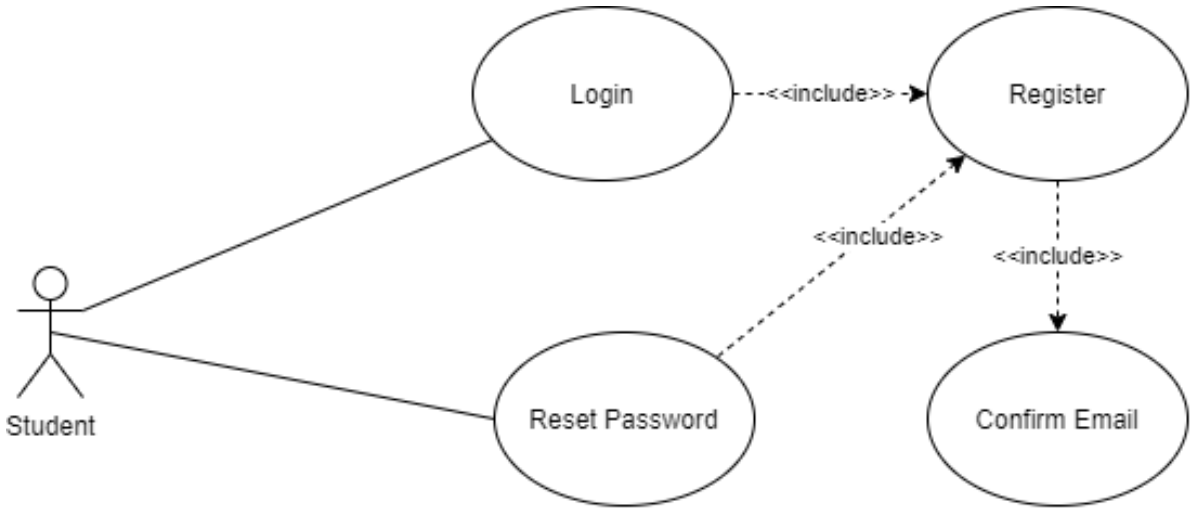
Use Case

Shows actions to be performed by actor

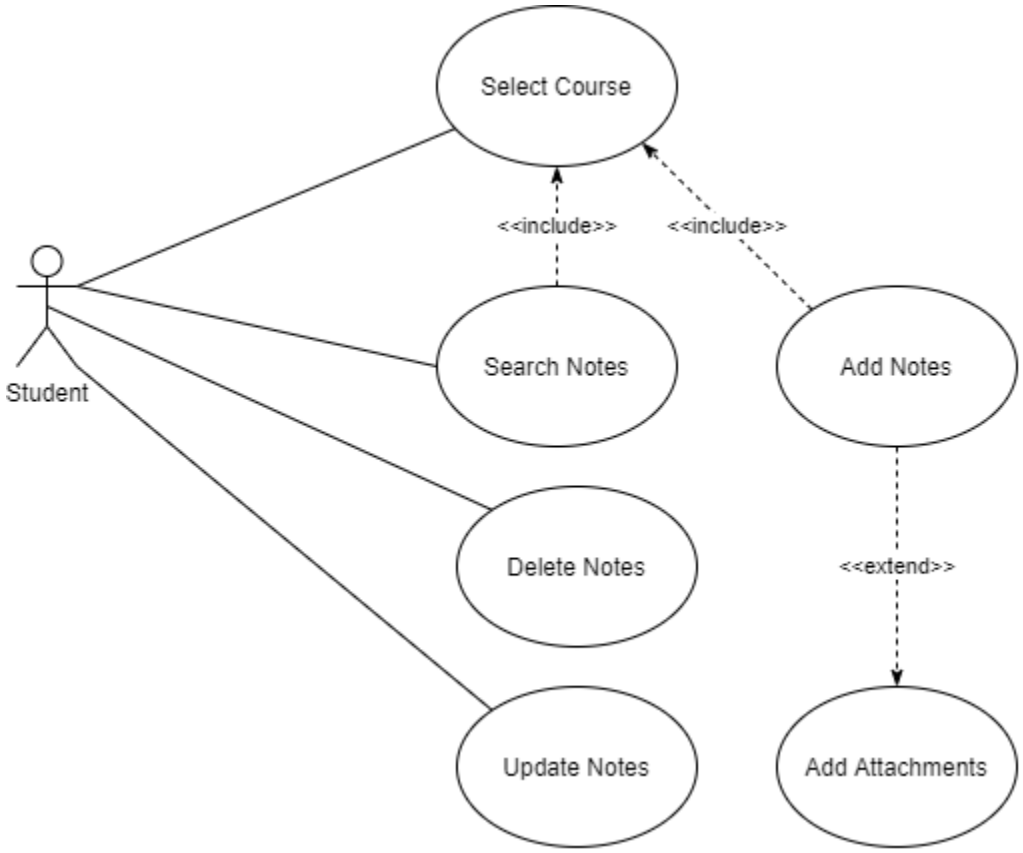
Adding an exam date to the timetable should include adding a course which includes adding semester data. The student must enter which semester they are in beforing adding courses to it. After that they can add dates to their specific course.



Students must login to enter the Student Companion. They need to do this by registering and confirming their email.

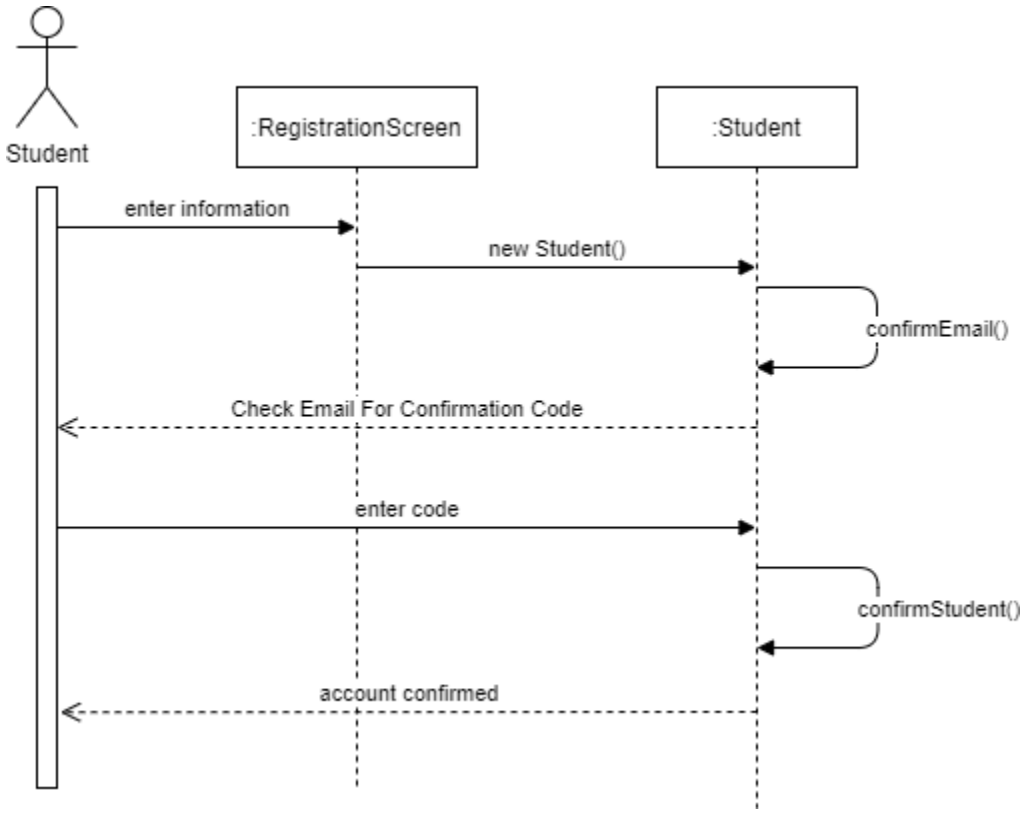


Adding notes to a course must require a user to select it then add notes. These notes can include attachments.

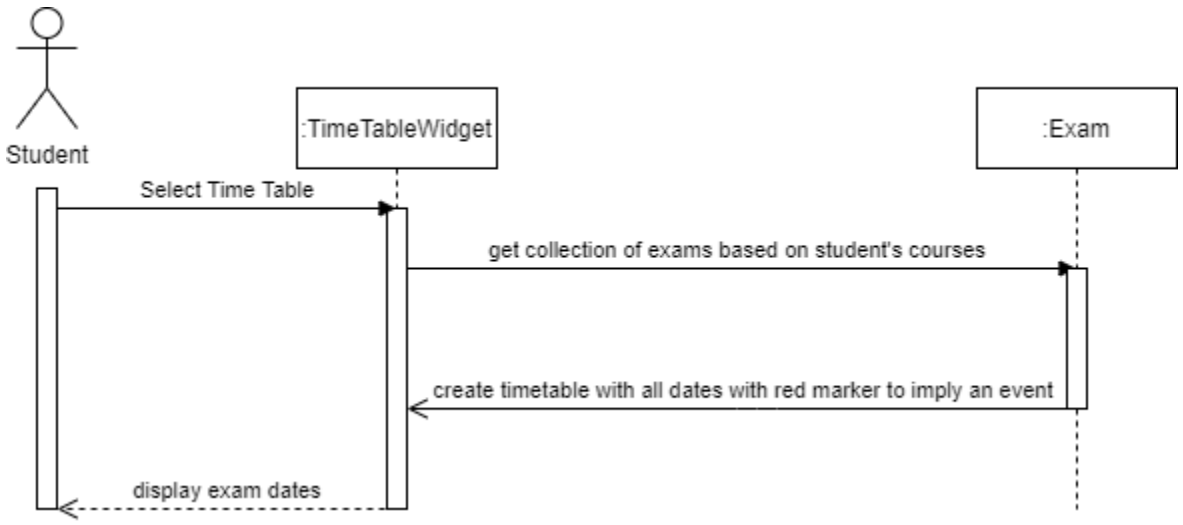


Sequence Diagrams

This shows the flow of registering a student to the system.



Viewing a timetable involves fetching a list and creating objects from the Exam class. After this will be displayed on the timetable widget/window.



Adding notes will require a user to select which course they want to add a note for first.