

GO!

Get Organized!

Produced By:

Seth Bedford, Katerin Flores, Sushith Ghanta
Shrinath Rao, Christopher tePoele

Final Project Draft Description

GO!

Our inspiration behind GO! (Get Organized) was the lack of availability of a single, unified organization app geared towards students. While many tools already exist to accomplish individual aspects of our platform, there is no one tool that caters to every potential need of the student.

It is for these reasons that we have decided to build a cohesive, all-in-one platform that provides all the features that a success-driven student would need. We anticipate our design to be used by students that want to be conscious about their academic performance and are looking to stay on top of their studies.

GO! Contains the following features:

- Study tracker
 - Used to plan, log, and track overall studying time goals for each course
- Grade calculator
 - Students will input their grades throughout the semester
 - The app will use user-given weights to calculate the grades
 - The grade calculator will provide real-time feedback about the scores a student will need on upcoming project to achieve a desired grade
- Calendar
 - Will import assignments from the assignments tracker, in addition to any extracurricular activities
 - Will also be used to track absences from class, which can be useful in classes that have certain attendance requirements for grades
- Assignments Tracker
 - Students will input the due dates of all assignments for their course to keep track of due dates across platforms

Sounds like a great support for students.

Member	Final Deliverable 1	Final Deliverable 2	
Seth Bedford	Class diagram and software process model	Project scheduling	Conclusion
Katerin Flores	Software requirements and address feedback	Test planning	
Sushith Ghanta	Sequence diagrams	Comparison of work with similar designs	
Shrinath Rao	Use case diagrams	Cost/effort/pricing	
Christopher tePoele	Architectural design	Cost/effort/pricing	

Fair

Pls make sure to include extra feature(s) to your design so as to make it uniquely different. Also, include a comprehensive comparison w/ similar applications in your final report.

Address Feedback:

To make our application more unique we decided to add file uploading features. When the user adds assignments they need to complete, they will be able to do more than specify the class and due date. The user will be able to upload the assignment specifications and any relevant sources or notes they will need to complete the assignment. This will allow the student to have all the necessary materials for each assignment accessible in one place and at their disposal regardless of their location or computer. The students will also be able to upload the syllabus to each individual class to have access to a convenient reference when needed.

Setting Up a GitHub Repository:

Team Members:

Seth Bedford - sethbedford
Shrinath Rao - shrinathrao97
Sushith Ghanta - sghanta
Katerin Flores - ksf-20
Christopher tePoele - CtePoele

Task 1.3 Completed By:

Katerin Flores

Task 1.4 Completed By:

Seth Bedford

Task 1.5 Completed By:

Shrinath Rao

Team Repository URL:

<https://github.com/ksf-20/3354-GO>

Task Delegation:

Member	Final Deliverable 1
Seth Bedford	Class diagram and software process model
Katerin Flores	Software requirements and address feedback
Sushith Ghanta	Sequence diagrams
Shrinath Rao	Use case diagrams
Christopher tePoele	Architectural design

Employed Software Process Model:

Our project will employ the spiral model due its evolutionary approach. GO! is a very modular application – there are several separate feature sets within the one application. By using the spiral model, we can develop GO! modularly and prototype each feature along the way. This stepwise approach will allow us to assess risks relevant to each function of the project at each stage. This model also allows for changes to be made throughout the process. If we decide that another feature needs to be added or an existing feature needs to be modified or removed altogether, then the spiral model will provide us with that flexibility.

Software Requirements:

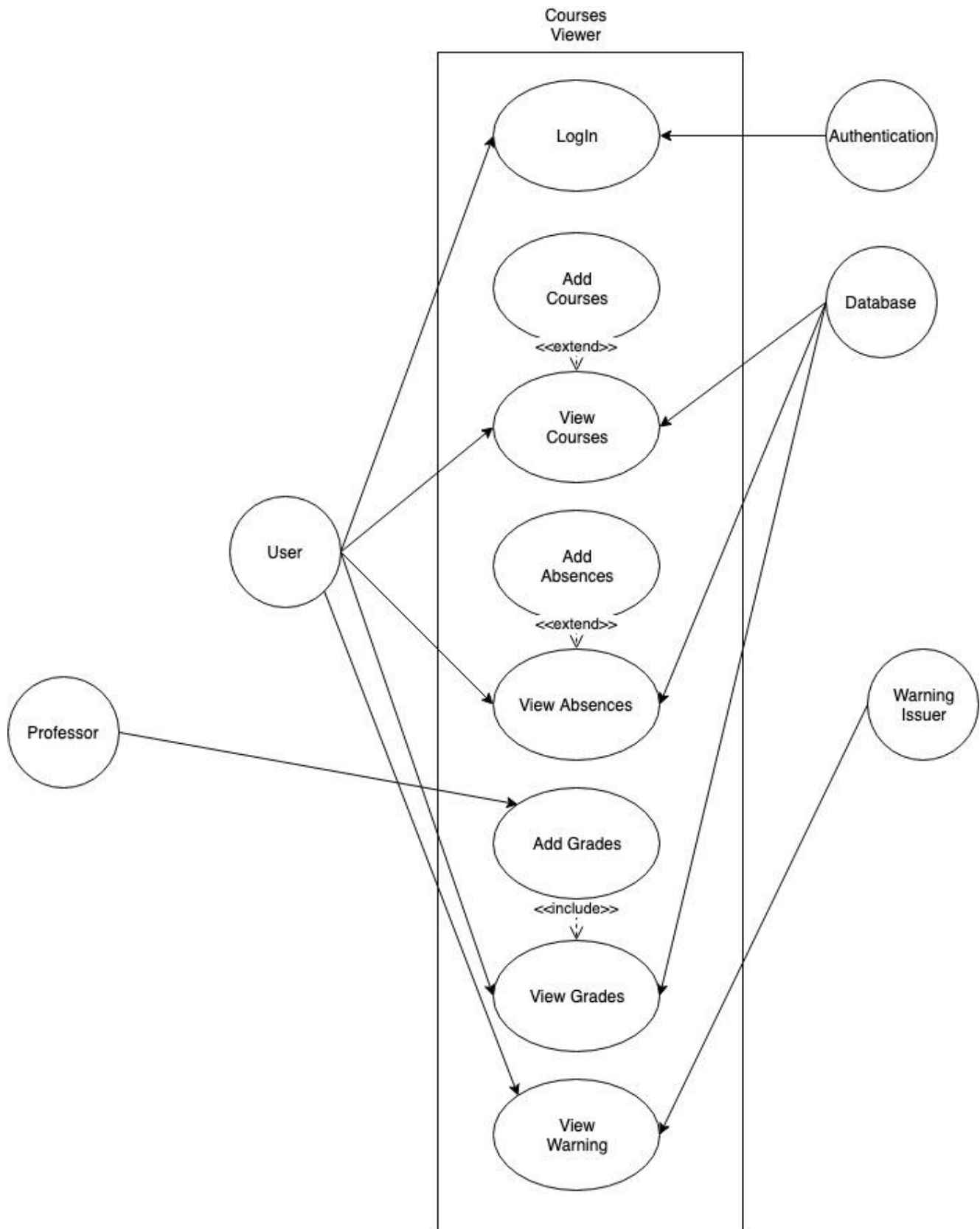
Functional Requirements:

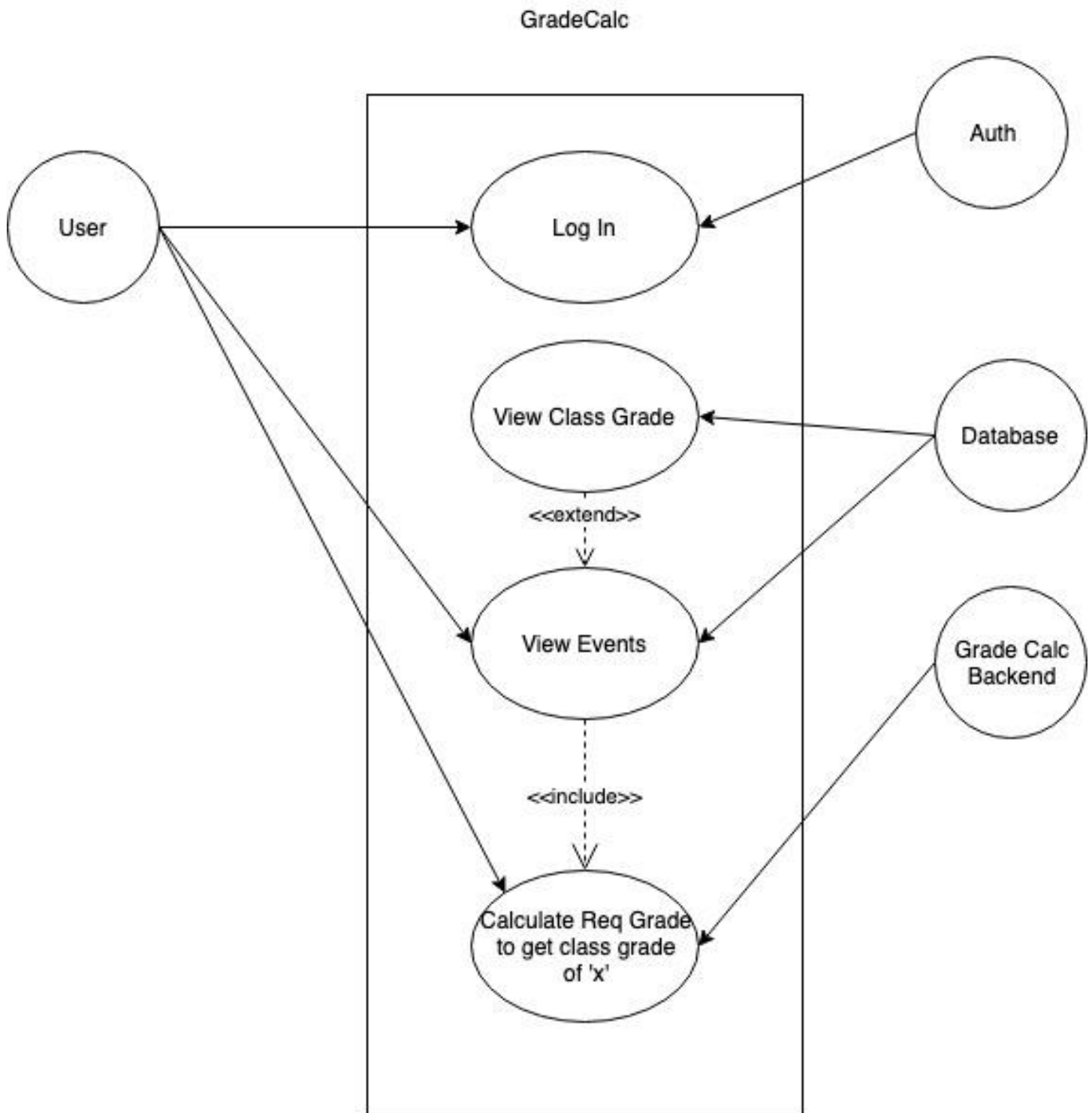
1. The user should be able to add class courses.
2. The user should be able to log absences and request receipt of past absences.
3. The user should be able to request required scores for desired grade.
4. The user should be able to add assignments and attach related files.
5. The user should be able to log their study hours.
6. The system should generate a weekly overview of assignments due, studying progress and upcoming events.
7. The system should issue reminders of assignments and exams that will be due in 24 hours.
8. The system should issue warnings when absences will exceed class regulations.

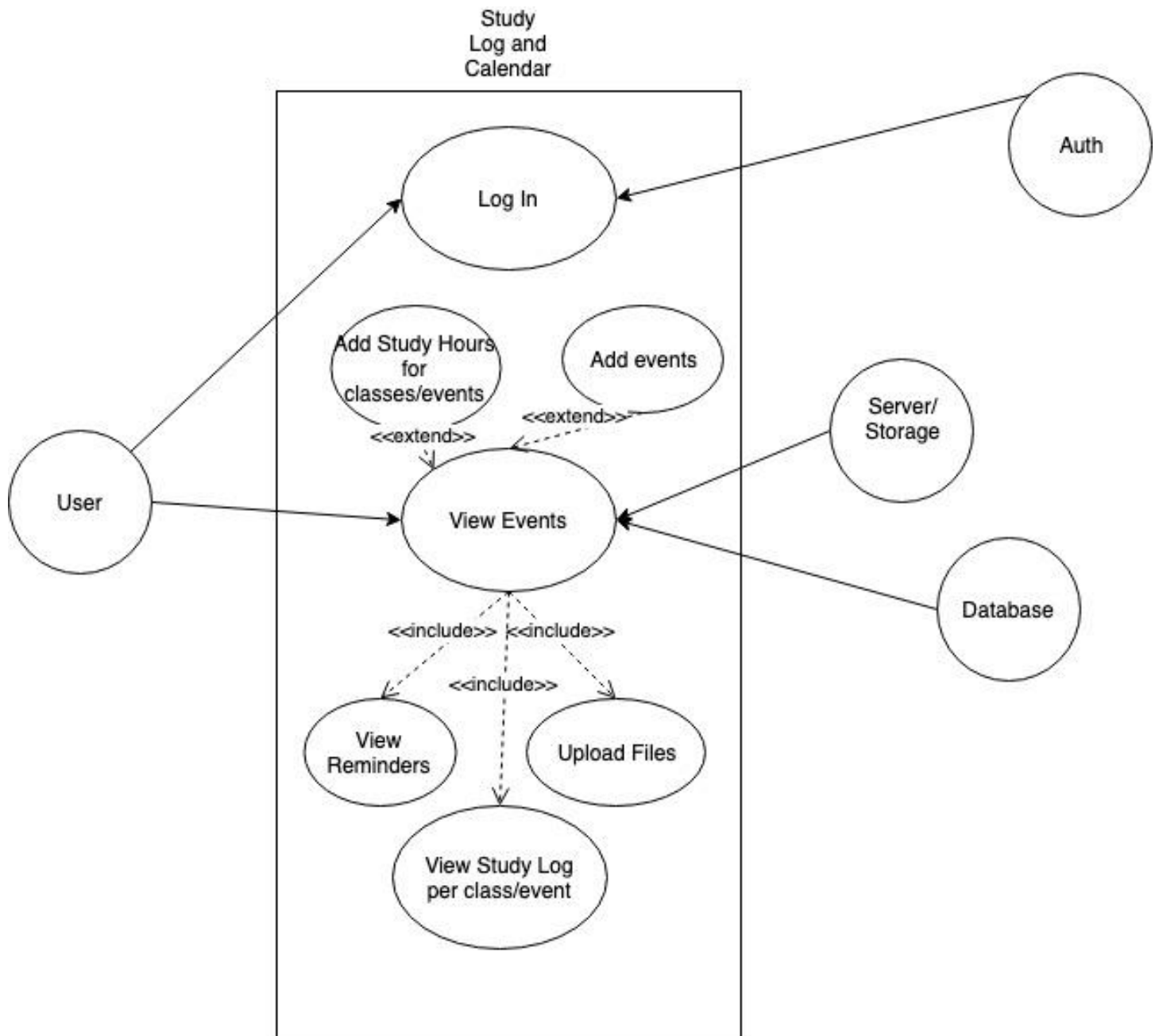
Non-functional Requirements:

1. (Product requirement) The GO! application should consistently update in real time the progress of the student, assignments due and activities that will occur within a 7-day period beginning on the present day.
2. (Organizational requirement) Users of the GO! application are required to sign-up with a school email to verify they qualify for free use of our service.
3. (External Requirements) The user's email, class assignments, grades and any other information provided by the user needs to remain confidential to protect the user's privacy.

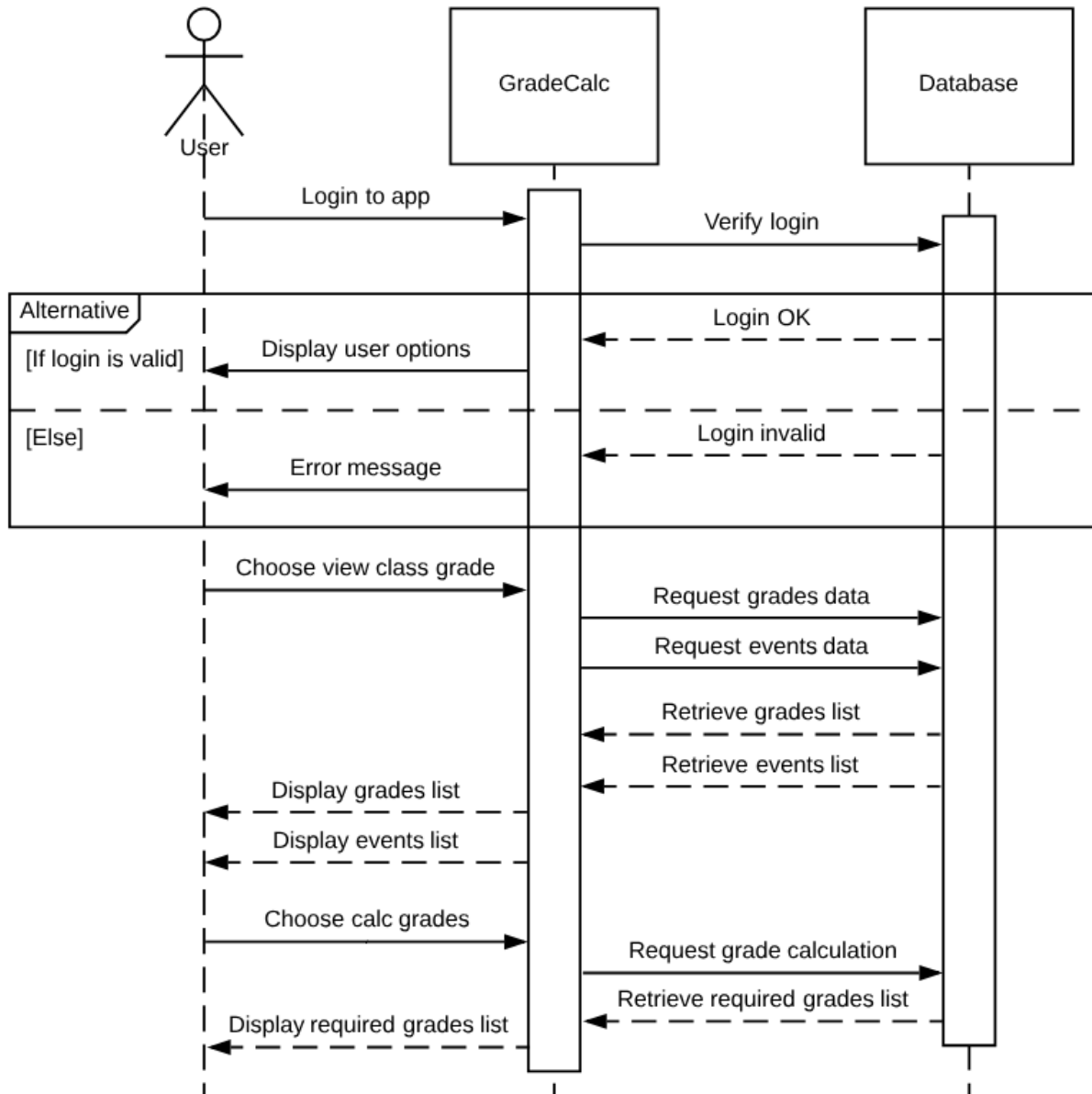
Case Diagrams:

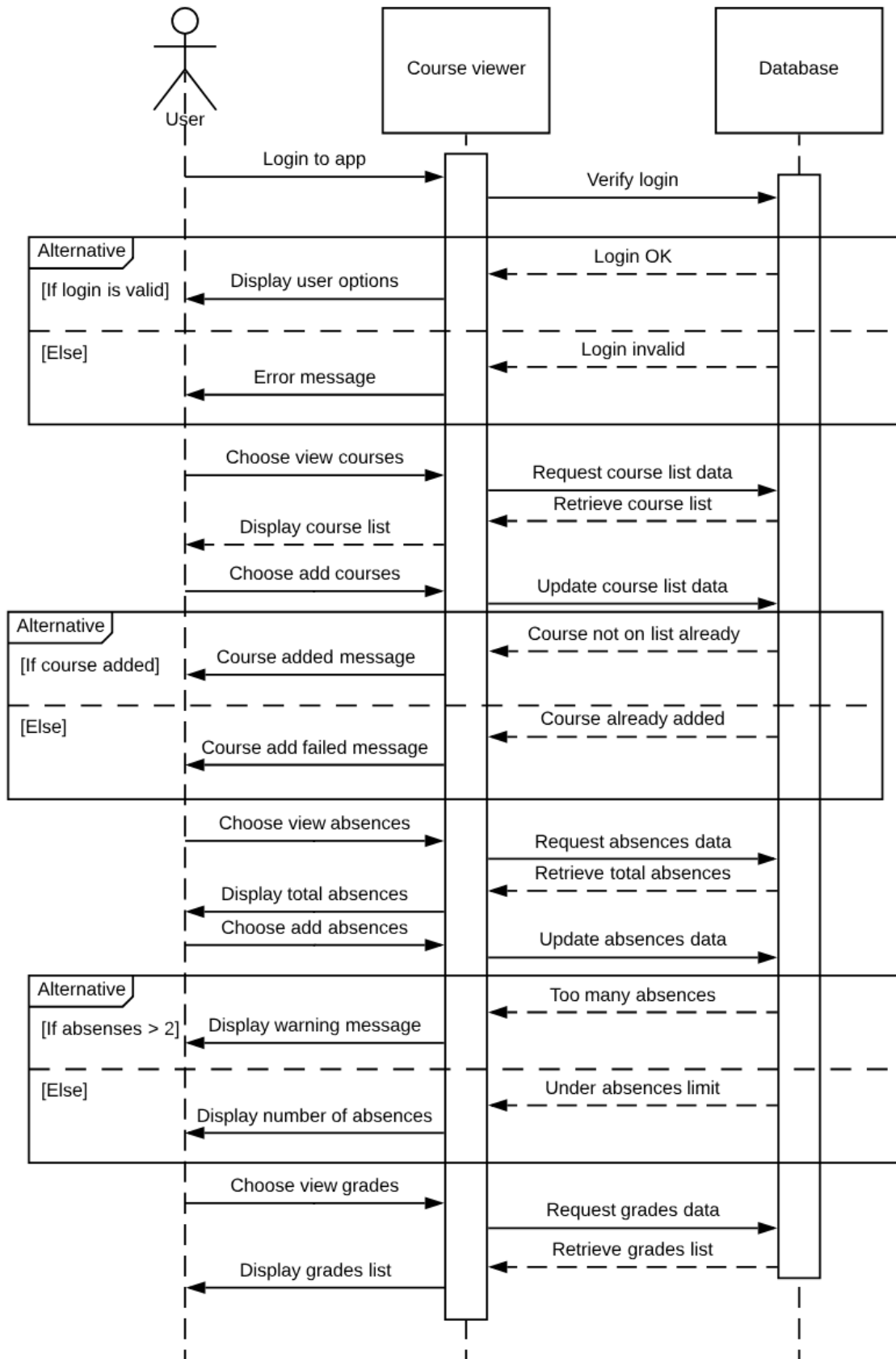


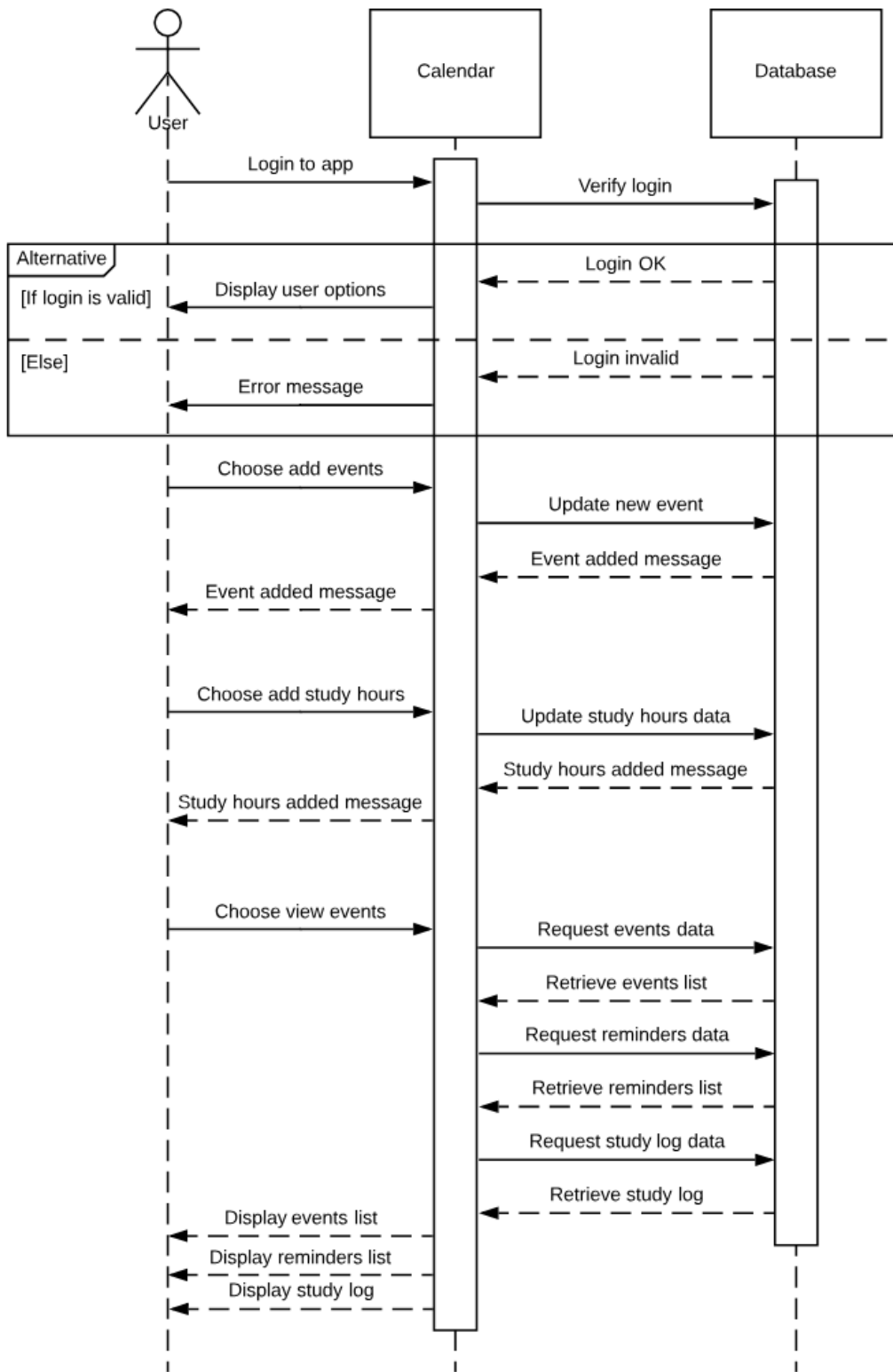




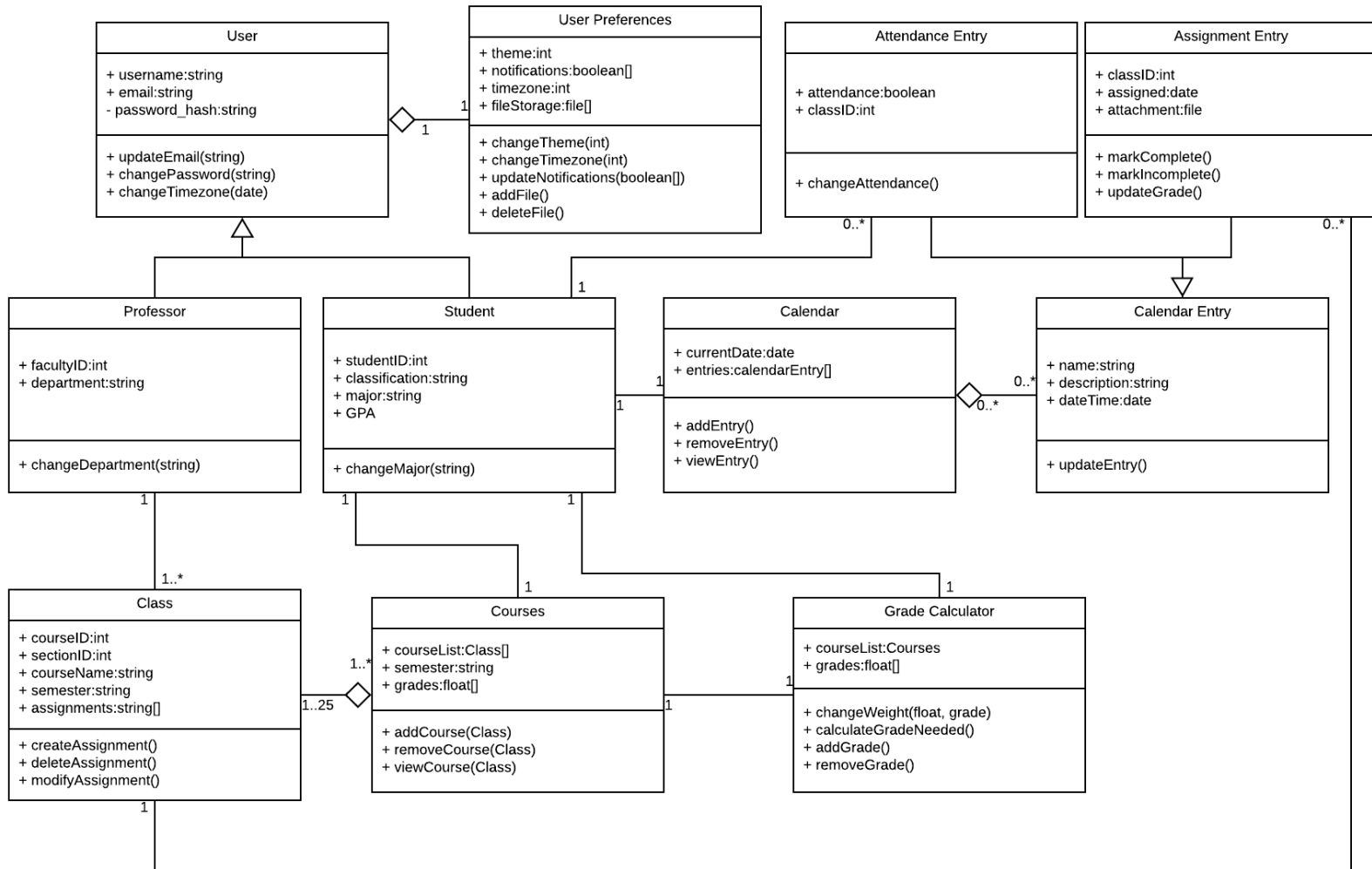
Sequence Diagrams:







Class Diagram:



Architectural Design – MVC:

