Team:
Jonathan Schmitt
Joseph Vostrejs
Nathan Welch
Title:

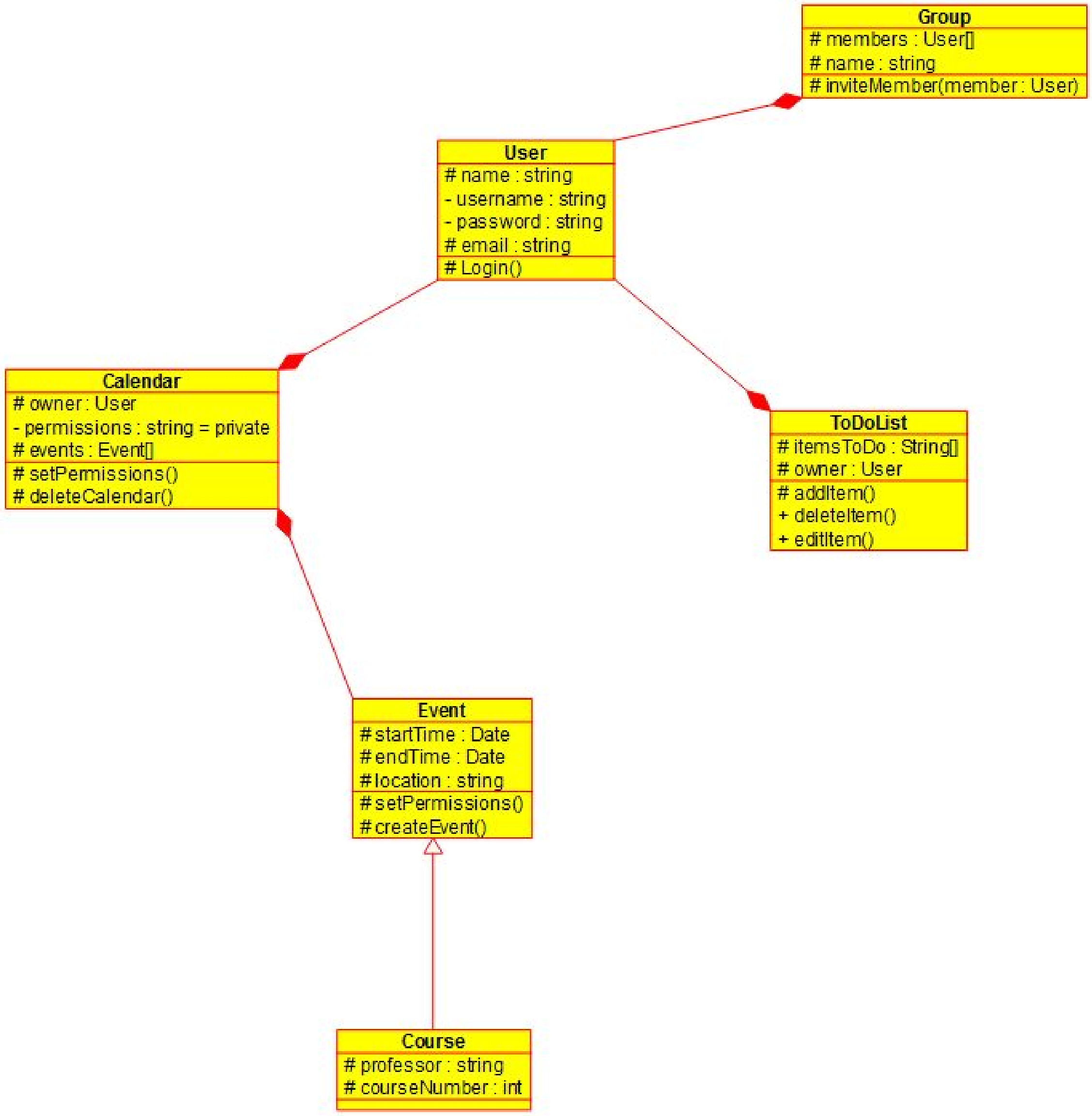
Entrak

Part 1 & 2

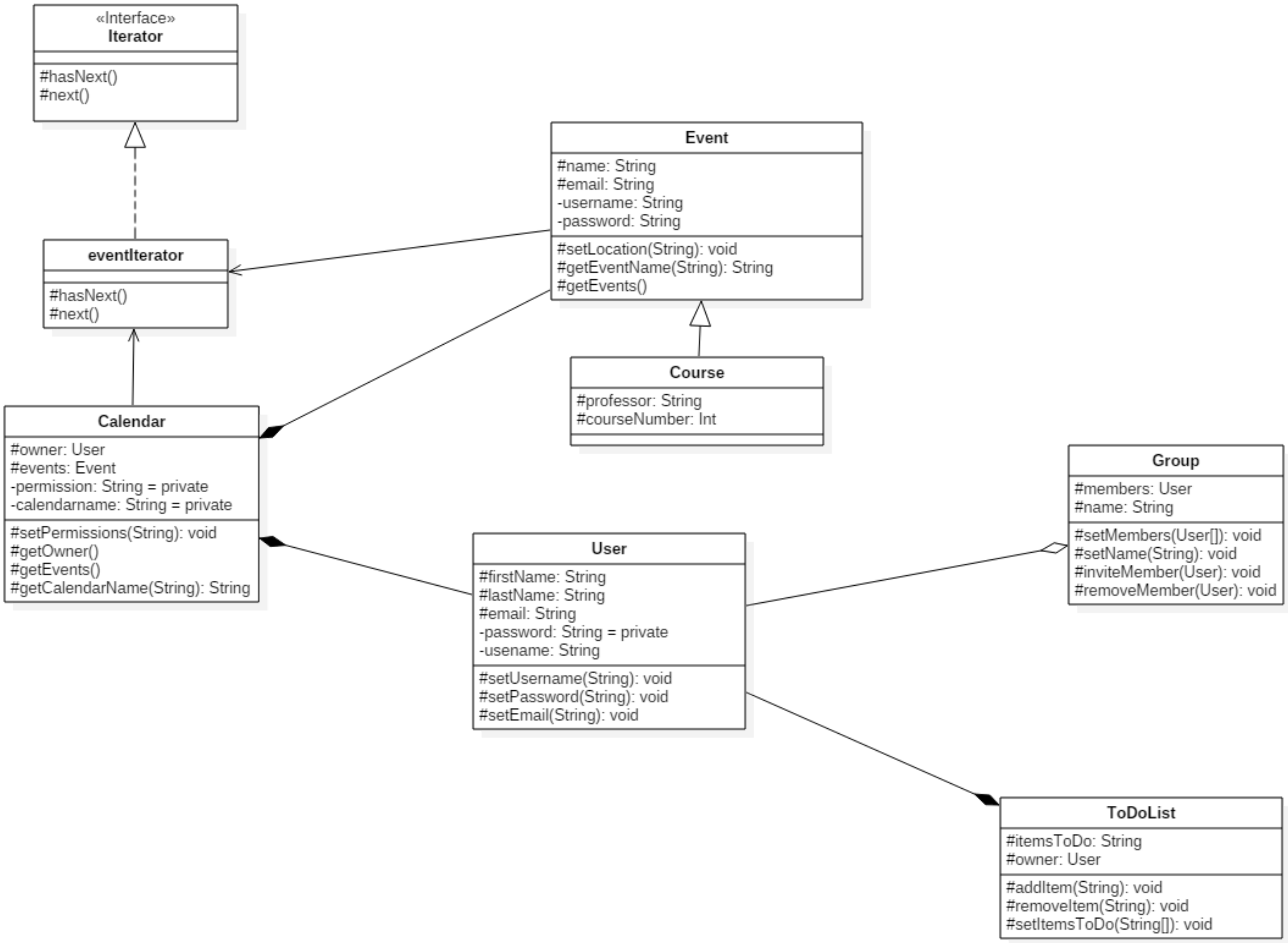
Features	Implemented?	Desc
BR-01	Y	Create Account
BR-02	N	Secure Password
UR-01	Y	Login and see schedule
UR-02	N	Edit event details
UR-03	N	Group events
UR-04	Y	New tasks in todolist
UR-05	N	Create a new group
UR-06	Y	Create a calendar
NFR-01	Y	See scheduler quickly at login
NFR-02	Y	Relevant information is first
NFR-03	-	Reliable and available

Part 3

Original Class Diagram:



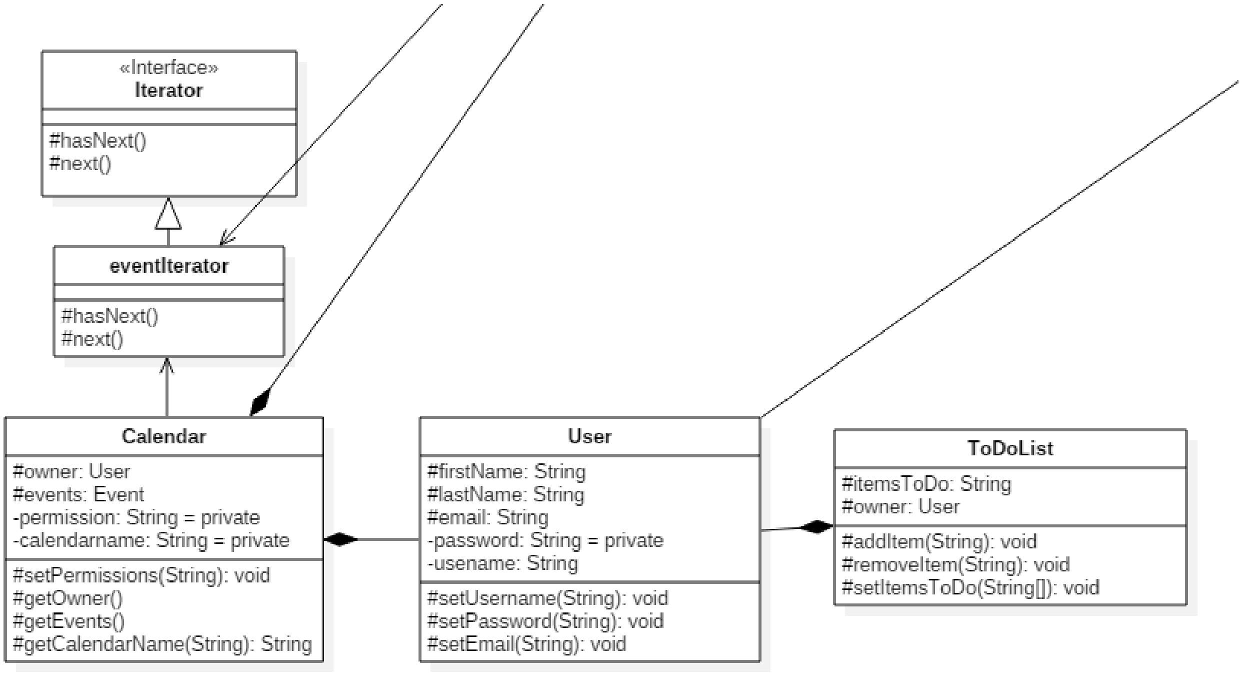
New Class Diagram:

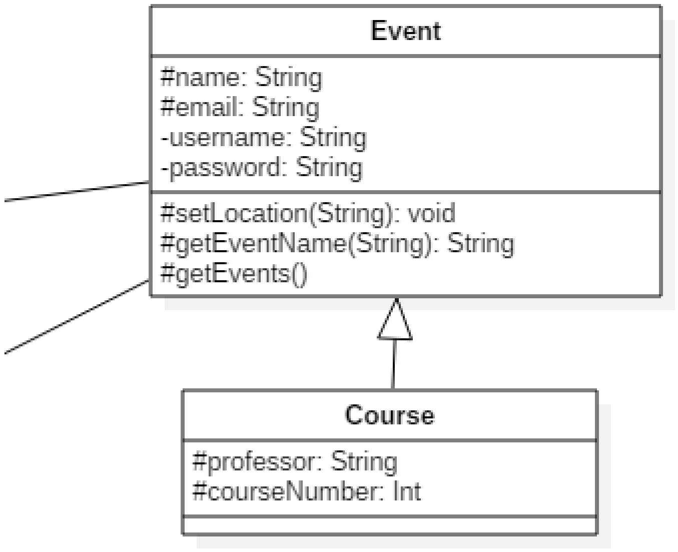


The Final version of our class diagram includes a lot more relevant methods and has updated formatting, in addition we’ve added an iterator interface. The chart has been cleaned up. Other than that having this design upfront helped us implement the classes for the project quickly, as we knew the requirements once we refactored once well. This allowed us to understand the design patterns we were using clearly, and the essential classes for our project to function.

Part 4:

We used both Iterator as well as Template design patterns. The Iterator design pattern was used through the main class to iterate through the todolist to access and view it. The Template design pattern was used as the Event superclass, allowing us to create multiple kinds of events, though as of current we only have one in the Course event type.





Part 5

This project has taught a lot about the usage of design patterns and planning design ahead. Having a class diagram early on was extremely useful, and we would definitely do it again in the future on other projects, even just a bare-bones one. Analysing and refactoring was also useful, as it allowed us to step back and re-evaluate our setup and code. This class has been useful in terms of learning many different design patterns and learning how to utilize them.