14:332:438 Software Engineering

Report II

Analysis, Design, and Implementation, Use Cases

Website:

https://github.com/sjlu/RU-Planner/wiki

Team Members:

Steven Lu

Nicholas Guida

Project: RUplanner

April 15, 2013

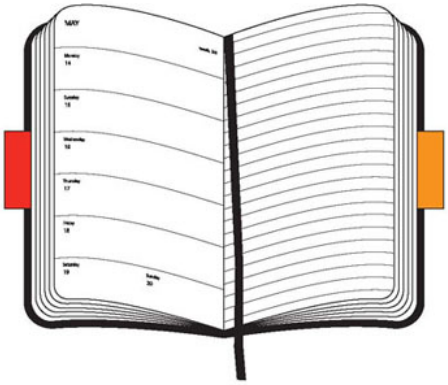


Table of Contents:

Scope of Report

For our second deliverable, our main goal was to get a functioning demo which incorporated the backend databases with the front end UI. This meant that we would, essentially, have everything working for one instance of our planner (in this case, the Computer Engineering major). We feel we have incorporated most important aspects into our report/demo, but can also add additional functionality if time permits.

Virtually all the use cases have been implemented, and assuming everything works fluidly with our demo, we will begin to incorporate all the other engineering disciplines. This should be somewhat trivial considering all of the hard background work will be in place, requiring nothing more than tedious data entry. We feel, however, that this report/demo was the most important step since it requires pulling everything together and creating something tangible. Our report should outline clearly how everything works together and what the expected behavior of our system is.

Use Cases

Implemented Use Cases for Deliverable II

Name: Create Planner (partially) **Number**: UC3
Description: Allows the user to create a planner for their major **Purpose**: To allow the user to create their planner and begin to plan out their academic caree

Name: Save Planner **Number**: UC4
Description: Allows the user to save their current planner **Purpose**: The user can save their existing work, which allows them to access it again at anytime using the same login.

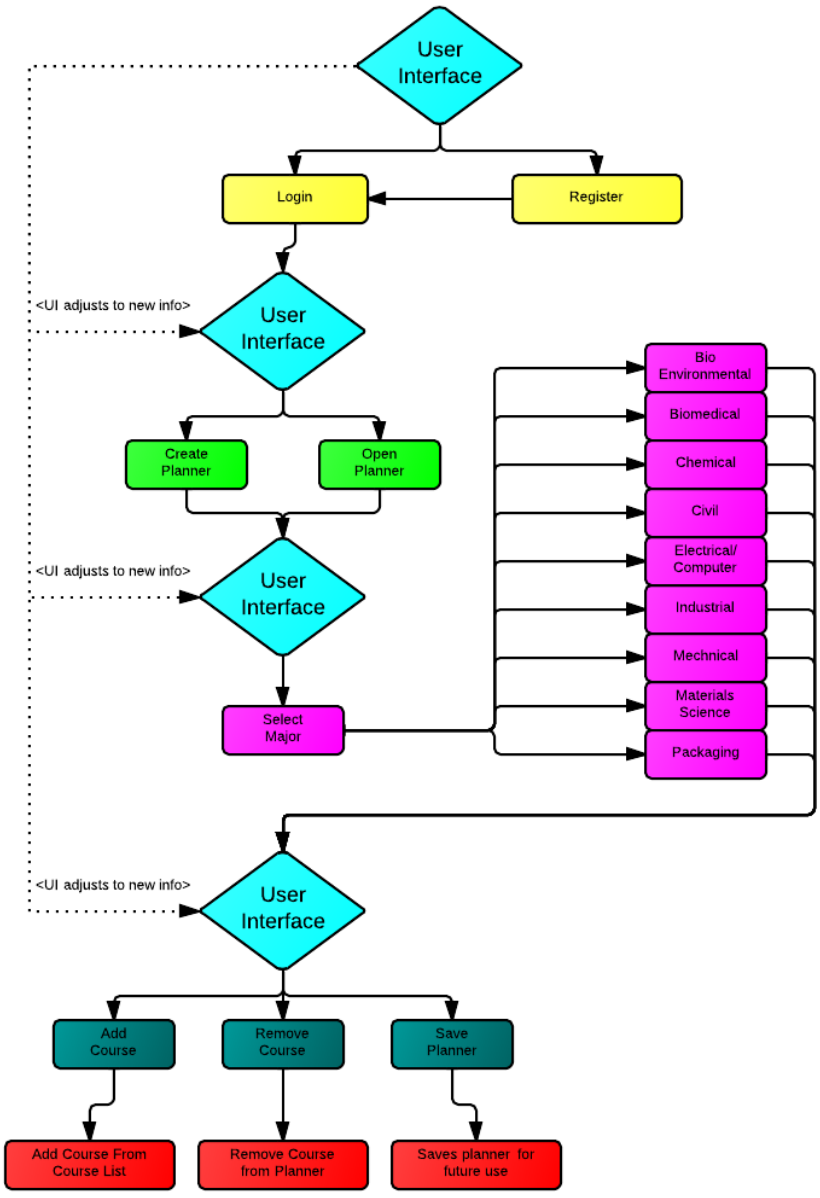
Name: Edit Planner **Number**: UC5
Description: Allows the user to edit/work on an existing planner **Purpose**: The user can access an already saved planner to edit it in order to add/drop courses and further figure out their schedule

Name: Delete Planner **Number**: UC6
Description: Allows the user to delete their planner **Purpose**: The user can completely erase their existing planner, allowing them to start from scratch or to work on another planner.

Name: Add Course **Number**: UC7
Description: Allows the user to add a course to their planner **Purpose**: The user can add courses to their planner in order to experiment with their course planner

Name: Delete Course **Number**: UC8
Description: Allows the user to delete a course in their planner **Purpose**: The user can delete a course from their planner in order to further experiment with their course planner.

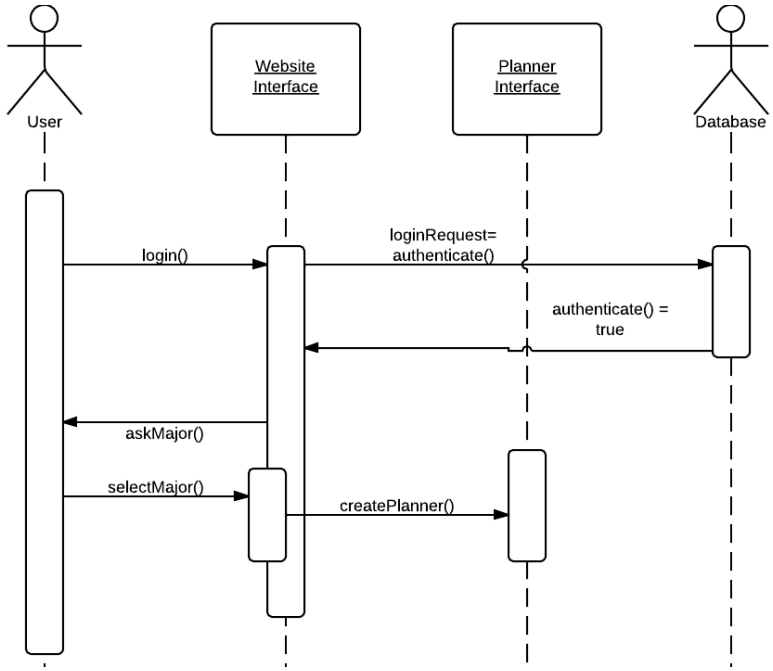
Conceptual Model



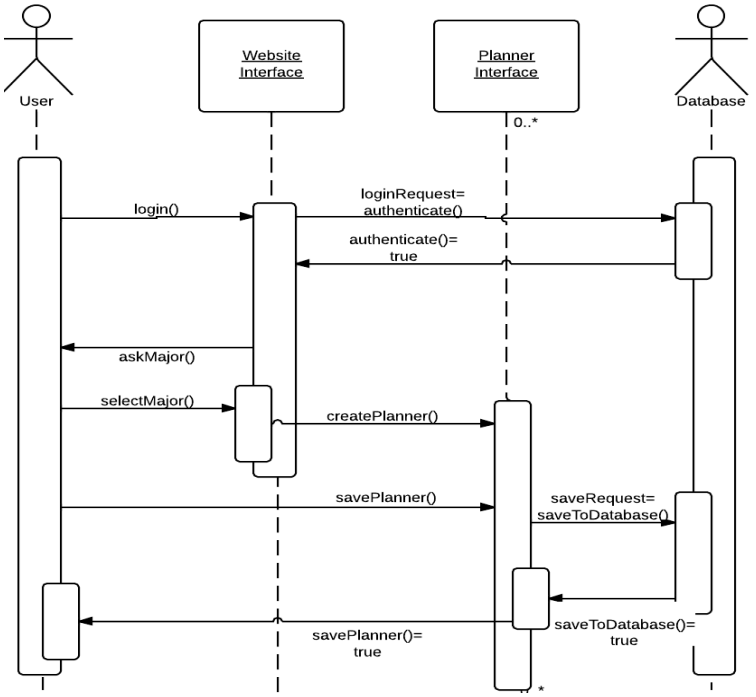
Our conceptual model has not changed from the previous report, and it still relates all of our use cases and systems together into a nice color coded model.

Sequence Diagrams

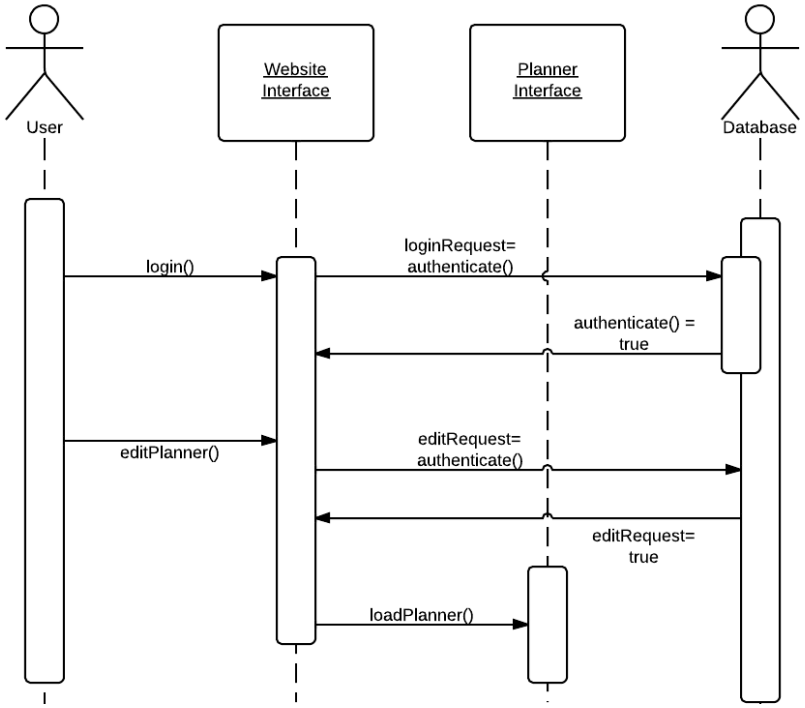
Create Planner



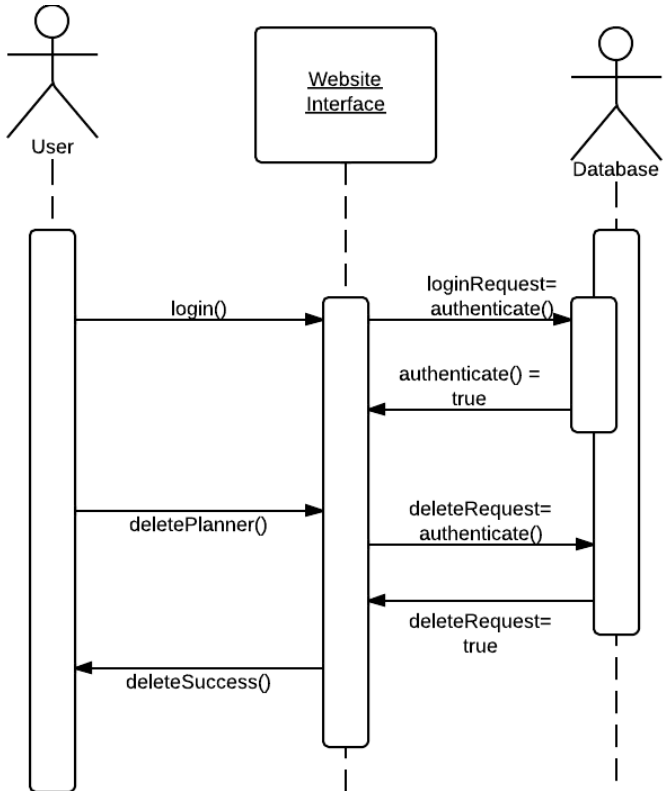
Save Planner



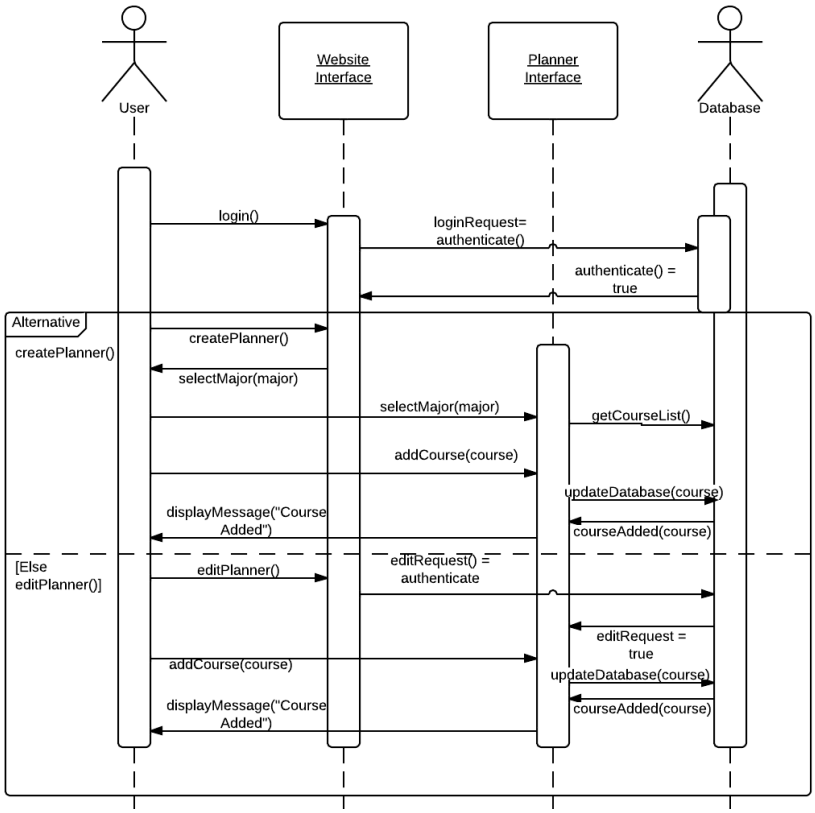
Edit Planner



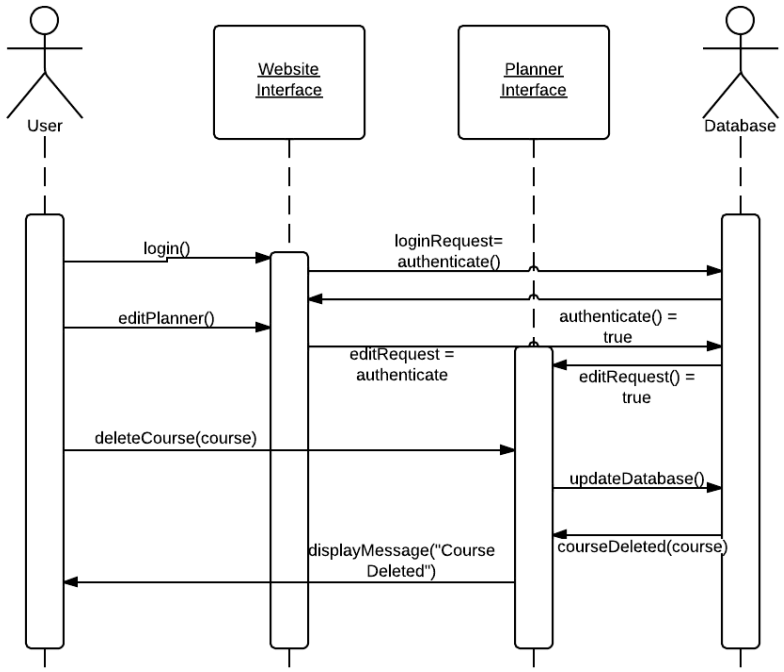
Delete Planner



Add Course



Delete Course



Document Contracts

Operation:	Create Planner
Cross References:	UC-3
PreCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering
PostCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering Student successfully selected his major and started a new planner

Operation:	Save Planner
Cross References:	UC-4
PreCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering Student has an open planner
PostCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering Student has a current open planner Student successfully saved his current open planner

Operation:	Edit Planner
Cross References:	UC-5
PreCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering Student has a previously saved planner
PostCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering Student has a previously saved planner Student successfully opened his previously saved planner

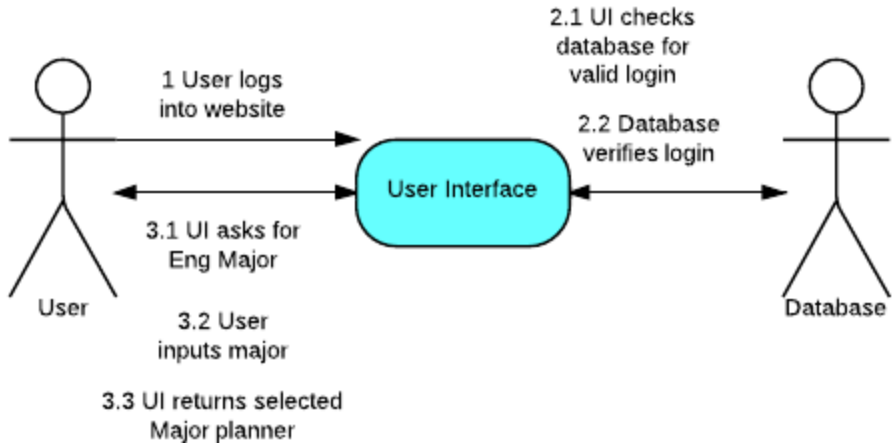
Operation:	Delete Planner
Cross References:	UC-6
PreCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering Student has a previously saved/open planner
PostCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering Student has a previously saved/open planner Student successfully deleted one of his planners

Operation:	Add Course
Cross References:	UC-7
PreCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering Student has a previously saved/open planner
PostCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering Student has a previously saved/open planner Student successfully added a course from the database

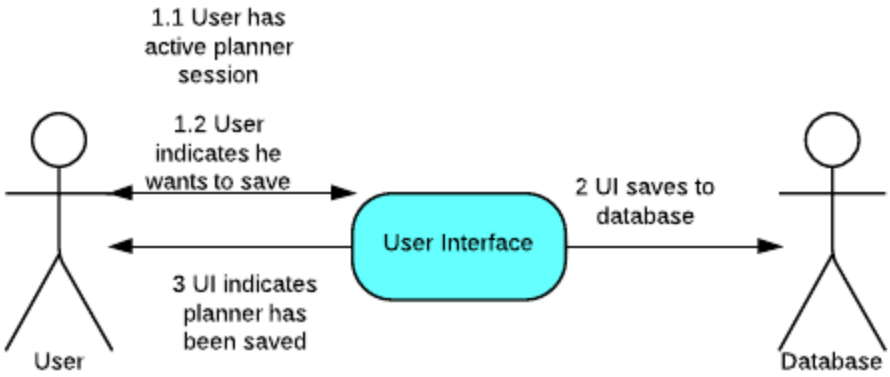
Operation:	Delete Course
Cross References:	UC-8
PreCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering Student has a previously saved/open planner Student has the course he wants to delete on his planner
PostCondition:	Student is at correct URL Student has a valid account Student is in the school of engineering Student has a previously saved/open planner Student has the course he wants to delete on his planner Student successfully deleted a course from his planner

Collaboration Diagrams

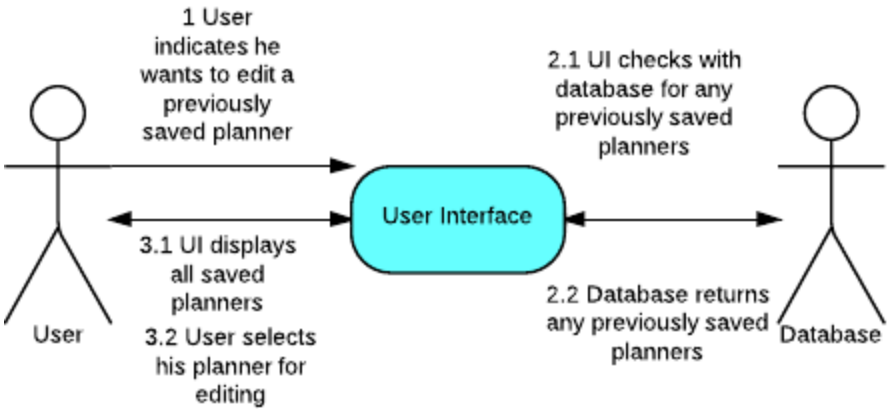
Create Planner



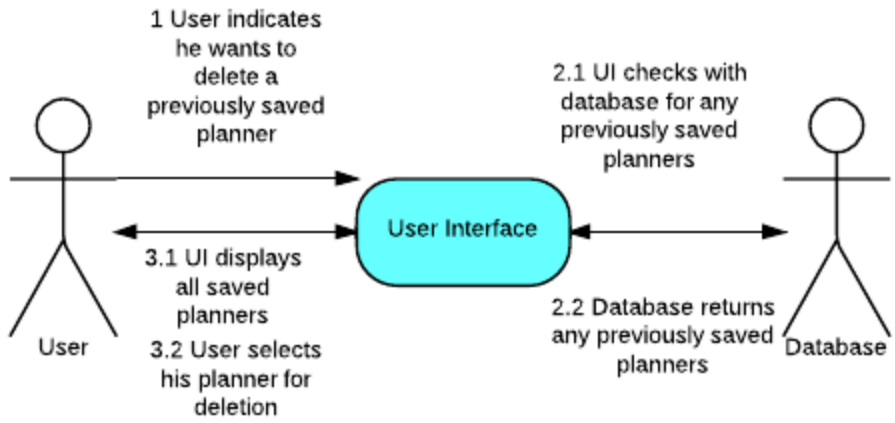
Save Planner



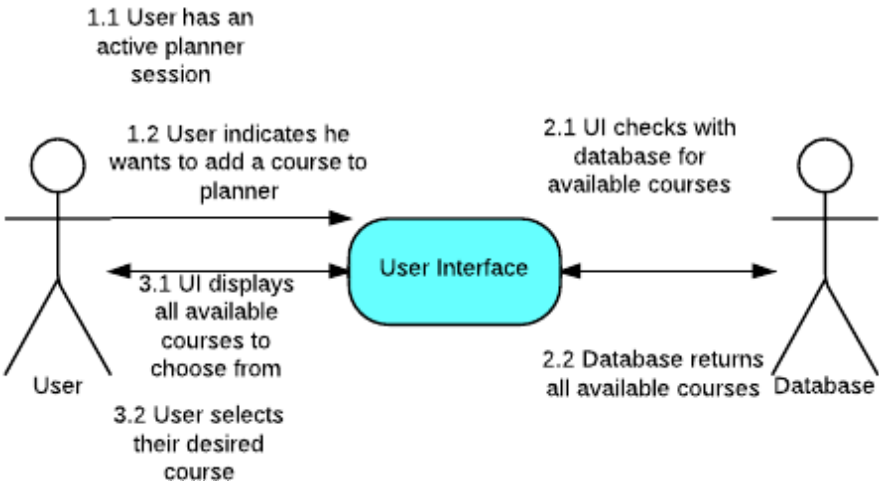
Edit Planner



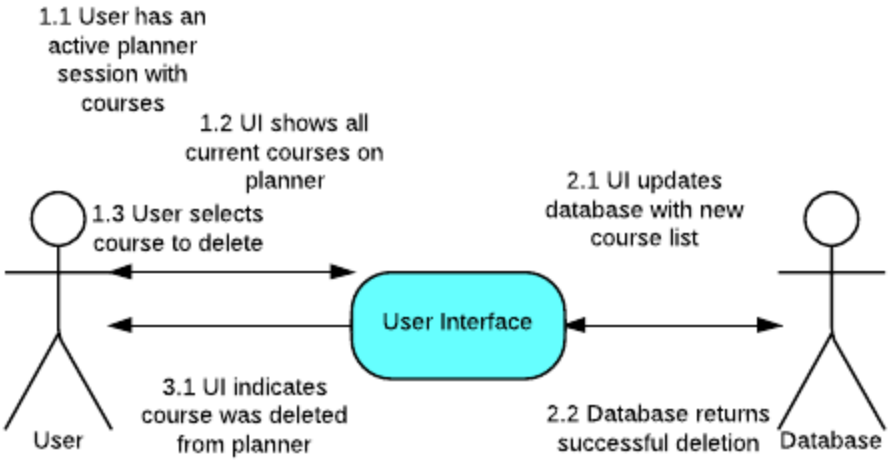
Delete Planner



Add Course



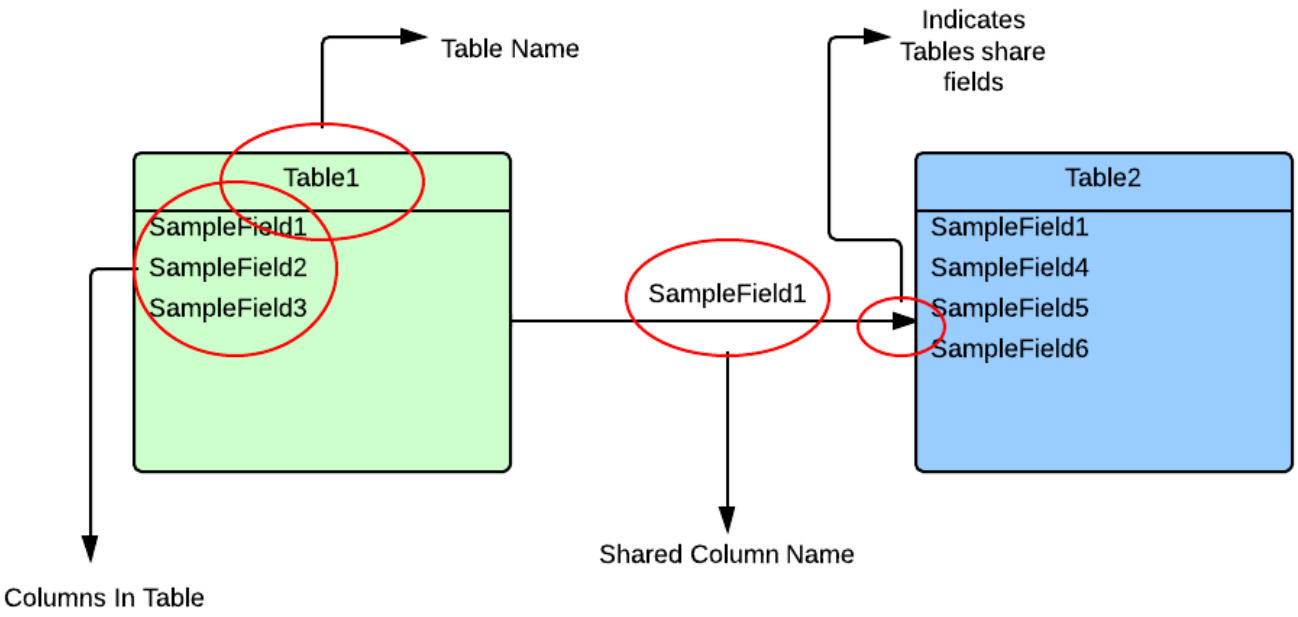
Delete Course



Class Diagram

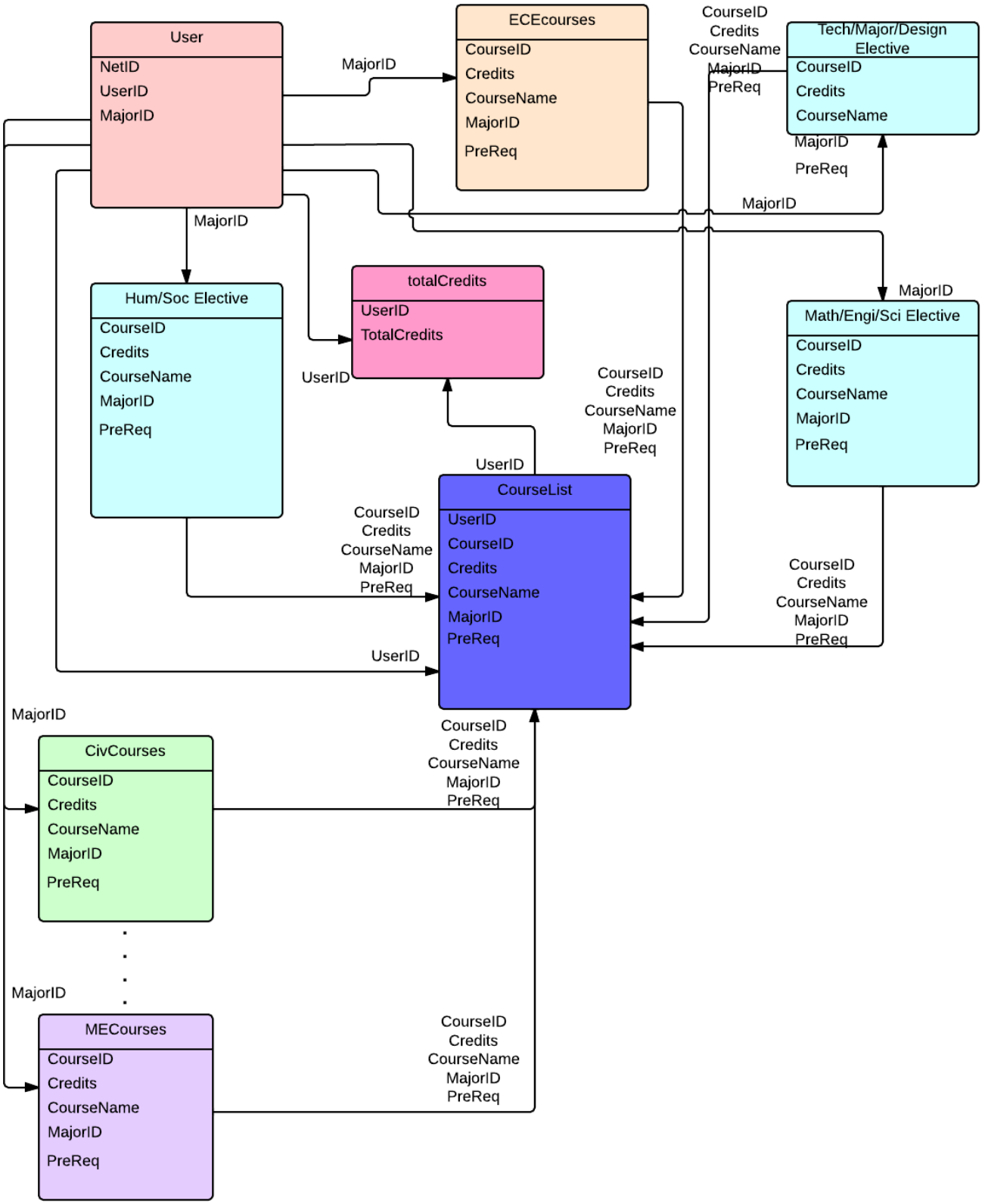
Because our capstone project is completely based on the web, we don't have any classes from OOP languages. However, since we had to create numerous databases, we'll include each type of database (e.g. we won't show each course database for every major, rather just one course database for one major so the idea is understood) and how they link together. Each database was programmed in mySQL and is subject to change throughout the project, however this is how it currently is.

Each Box represents a different table in our database, and each arrow represents how the tables communicate within the databases with the text next to the arrow showing the field which gets shared between tables. EX:



For report II, we have kept the same database structure and have only added further information to our databases. Our databases can be found at the same site as the rest of our code (next section).

Database Relationship Diagram



Code (Header Files)

All of our code (which is html/css/php/mysql) can be found on our github repository:

https://github.com/sjlu/RU-Planner

Modifications

We have modified our original deliverable chart, since we have added all available use cases (except for sharing) to our working demo. However, we do only have a functioning version of our project for Computer Engineering. The other majors will have to be held off until we get this working perfectly, at which case it should be trivial to add in all the other majors. For now, however, everything seems to be on track. Future modifications may still come out before our last report

UC#	Deliverable I (3/27/2013)	Deliverable II (4/15/2013)	Deliverable III (5/12/2013)
UC1	-	-	-
UC2	✓	✓	✓
UC3	x	✓	✓
UC4	x	✓	✓
UC5	x	✓	✓
UC6	x	✓	✓
UC7	x	✓	✓
UC8	x	✓	✓
UC9	x	x	✓
UC10	✓	✓	✓

Individual Contributions

Steven Lu:

- Web Page Design
- Web Page Development
- UI Development

Nicholas Guida:

- Created Report/Presentation
- Database Creator

Time Log

(Since last deliverable)

Name	Date	Time	Reason
Steven/Nick	4/01	2 hours	Meeting
Steven	4/03	4 hours	Website Coding
Steven	4/05	3 hours	Website Coding
Nick	4/05	3 hours	Database Creation
Steven	4/11	4 hours	Website Coding
Nick	4/11	4 hours	Database Input
Nick/Steven	4/13	2 hours	Meeting