DROP GRADE

SOFTWARE REQUIREMENT SPECIFICATIONS

Developed by: Alan Donham, Madison McHam, Shane Peters, and Nick Levert

Table of Contents

Change History

- 9/14/15 – Creation of document
- 9/16/15 – Add project Description – Shane Peters
- 9/16/15 – Add introduction – Alan Donham
- 9/16/15 – Add Functional and Nonfunctional Requirements – Madison McHam
- 9/16/15 – Add use case diagrams – Nick Levert
- 9/17/15 – Changed formatting to improve overall look – Alan Donham
- 9/17/15 – Added class diagram – Madison McHam
- 9/18/15 – Added activity diagrams – Shane Peters and Nick Levert
- 9/21/15 – Changed wording to improve readability - Group
- 10/1/15 – Added updated use case diagram - Group
- 10/1/15 – Updated to accommodate faults - Group

1. Introduction

This document will provide all the necessary information on the requirements necessary for development of the DropGrade application.

1.1 – Purpose

The purpose of this app is to provide the user not only with a way of tracking their course average in a class, but also with detailed information about professors and classes they are interested in registering for. Many apps offer features similar to this, however there does not exist one currently that is able to bundle these features we are offering into one sleek, easy to use, yet incredibly informative application like ours will.

1.2 - Goals and Challenges

One of the biggest challenges this application will face will be gathering and storing all the data collected in a way that will allow us to provide useful statistics to the user very quickly and in real time. This biggest challenge is also our number one goal because it is important that the user is able to acquire the data they want in near real time or they will likely not use the application we develop. Overall our goal is to provide the user with an application that will allow them to calculate their grade and view what if scenarios depending on the possible grades they can make on upcoming assignments and exams.

1.3 – Definitions

What-If-View – A view that will provide the user with tools to manipulate their final course average based on predicted future scores

Course-View – A view that will list all the courses a student is currently enrolled in

Professor-Stats – A view that will list statistics on professors from data that is collected

Course-Edit-View – A view that will allow user to input information regarding a course

GPS (Grade Prediction System) – The subsystem responsible for calculating course averages

GPA – A scale used to represent an average score that ranges from 0 to 4.3

Required - A requirement listed as required is seen as a requirement for completion

Stretch – A requirement listed as stretch will be completed should time allow

1.4 – Actors

- Students at The University of Alabama (User)

- Server (Database)

2. Description

2.1 – User Interface

The interface for the app will be a basic touchscreen interface similar to most apps on the market. The app will open onto a home screen that will have a list of classes that the user has entered. The information will be presented to and gathered from the user by a combination of table views, text boxes, drop down menus, and keypad entry.

2.2 – Storage

Information about course instructors is sent to and stored on a database.

2.3 – Communication

Requires cellular data or an Internet connection to send/receive data from a remote server.

2.4 - Application Functions

This application is intended to serve as a grade calculator and teacher-rating tool. Users will be asked to enter a class and it's syllabus information. As the users progress through the school term they will input their grades into the app and track their overall course grade. At the conclusion of the course the user will be asked to rate and give a description of their course instructor(s).

2.5 - Assumptions

- Grades will be given in a number format. User reported their grades honestly.
- User has internet access

2.6 - Stretch Goals

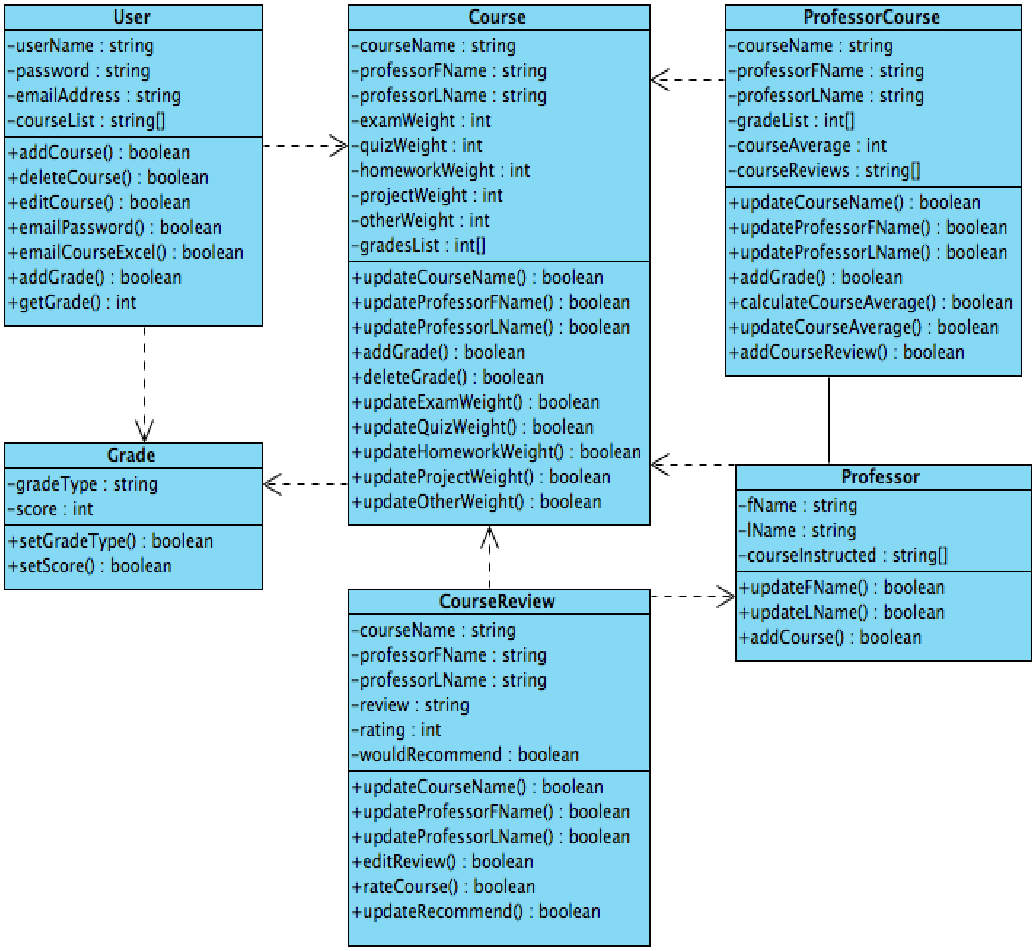
- Reminder function that reminds user of a test or assignment.
- Have list of classes and their respective syllabi stored on server for easy download.
- Export course grade information to an excel spreadsheet

3. Functional and Nonfunctional Requirements

Functional Requirement	Status
Create Account	Required
Log In	Required
Retrieve Password	Required
Add Course	Required
Add Course Info	Required
Enter Grades	Required
Edit Grade	Required
Submit Final Grade	Required
Manipulate Final Grade via the What-If-View	Required
Add Professor Review	Required
Delete Professor Review	Required
Edit Professor Review	Required
Export Data to an Excel Spreadsheet	Stretch
Send reminders to user when they have an upcoming Exam	Stretch
Store course information(including grading scale)	Stretch

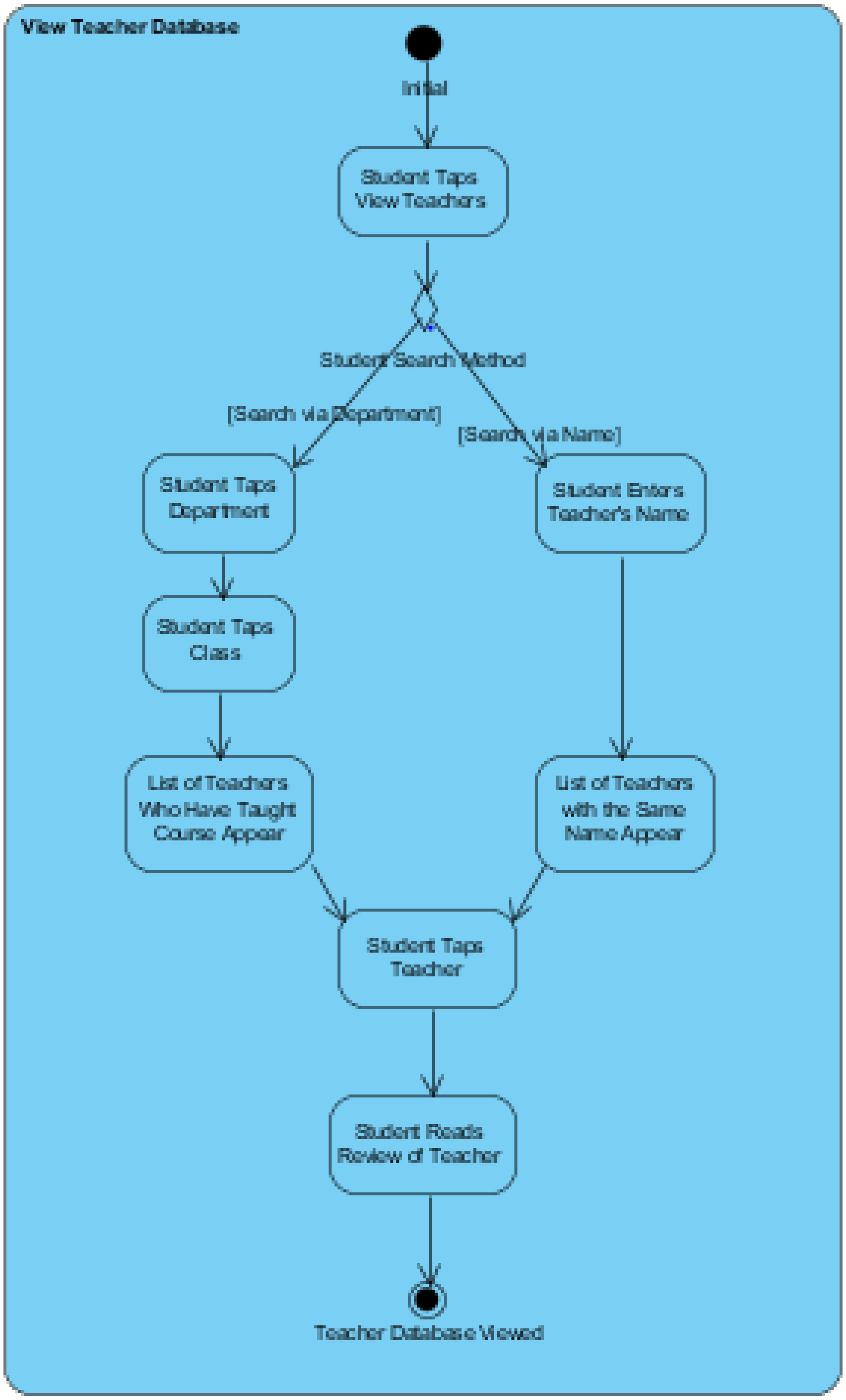
Nonfunctional Requirement	Status
Database that can handle multiple users	Required
App must not drain battery	Required

4. Class Diagram

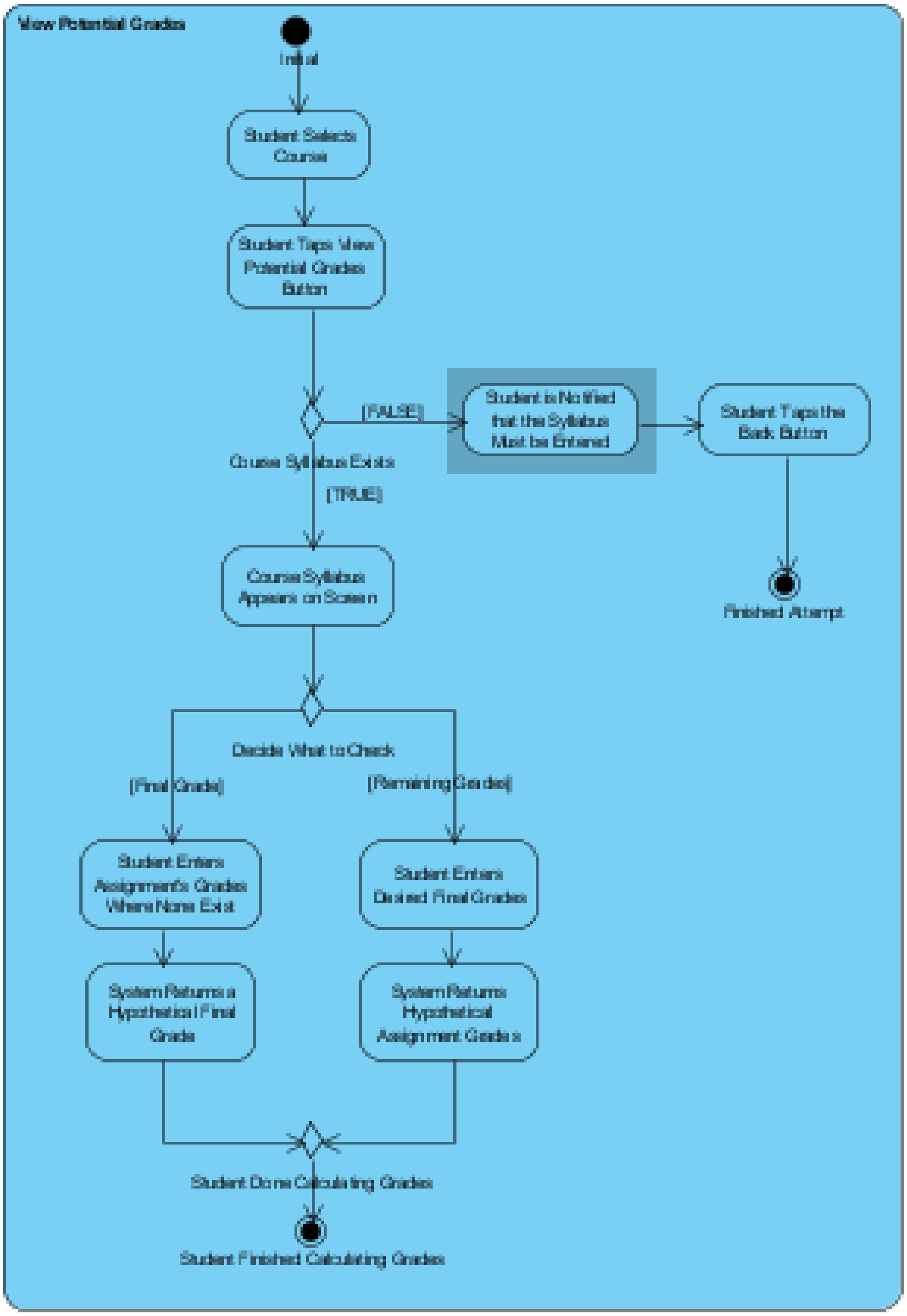


Our application will function in accordance with this diagram of our classes. As can be seen by looking at the diagram, everything is dependent on the User class which will store the necessary data to drive how our application talks to our database. The other classes listed above provide an organized way to store information, access it, and perform calculations on the information with ease.

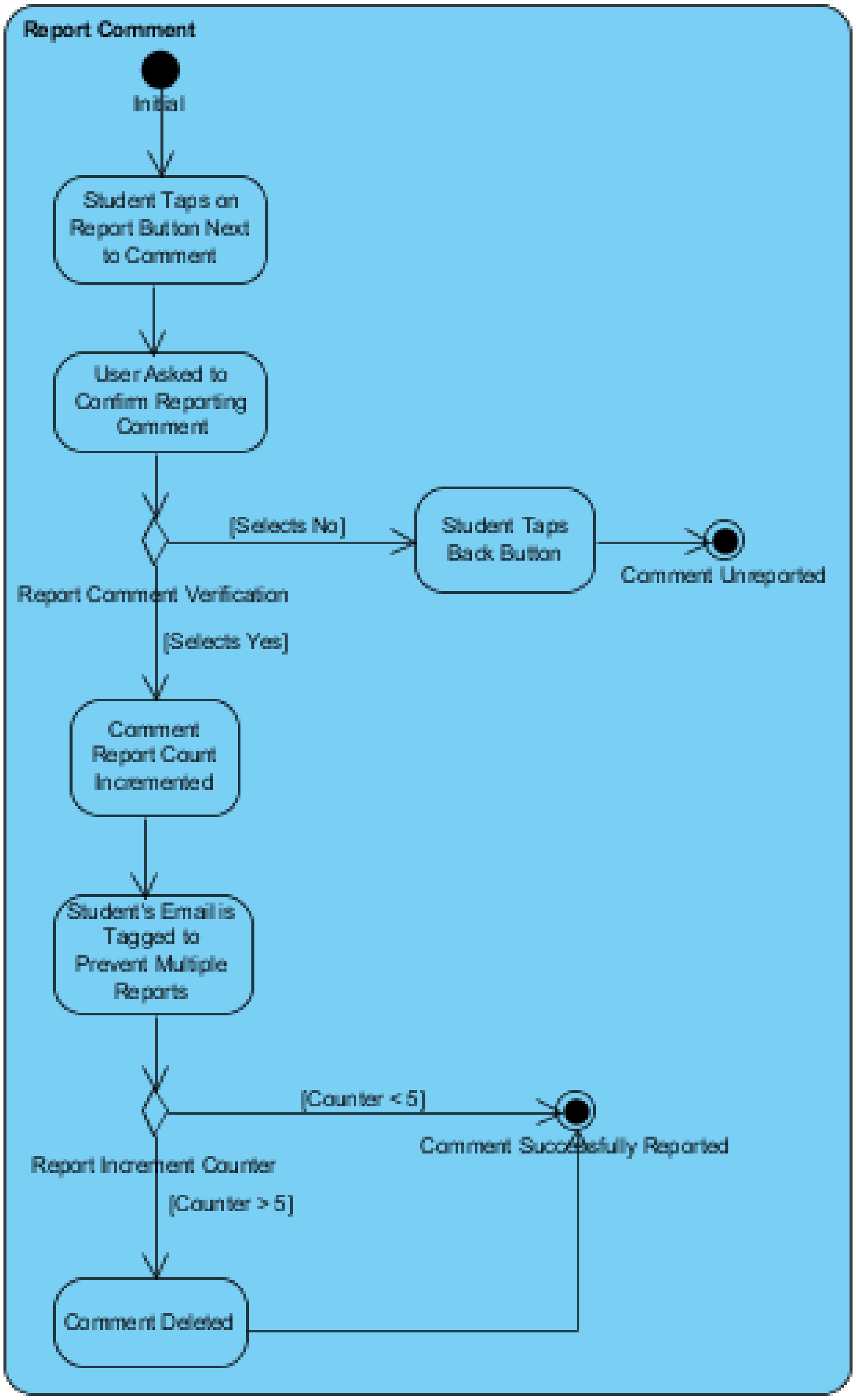
5. Activity Diagrams


View Teacher Database
Initial
Student Taps View Teachers
Student Search Method
[Search via Department]
[Search via Name]
Student Taps Department
Student Enters Teacher's Name
Student Taps Class
List of Teachers Who Have Taught Course Appear
List of Teachers with the Same Name Appear
Student Taps Teacher
Student Reads Review of Teacher
Teacher Database Viewed

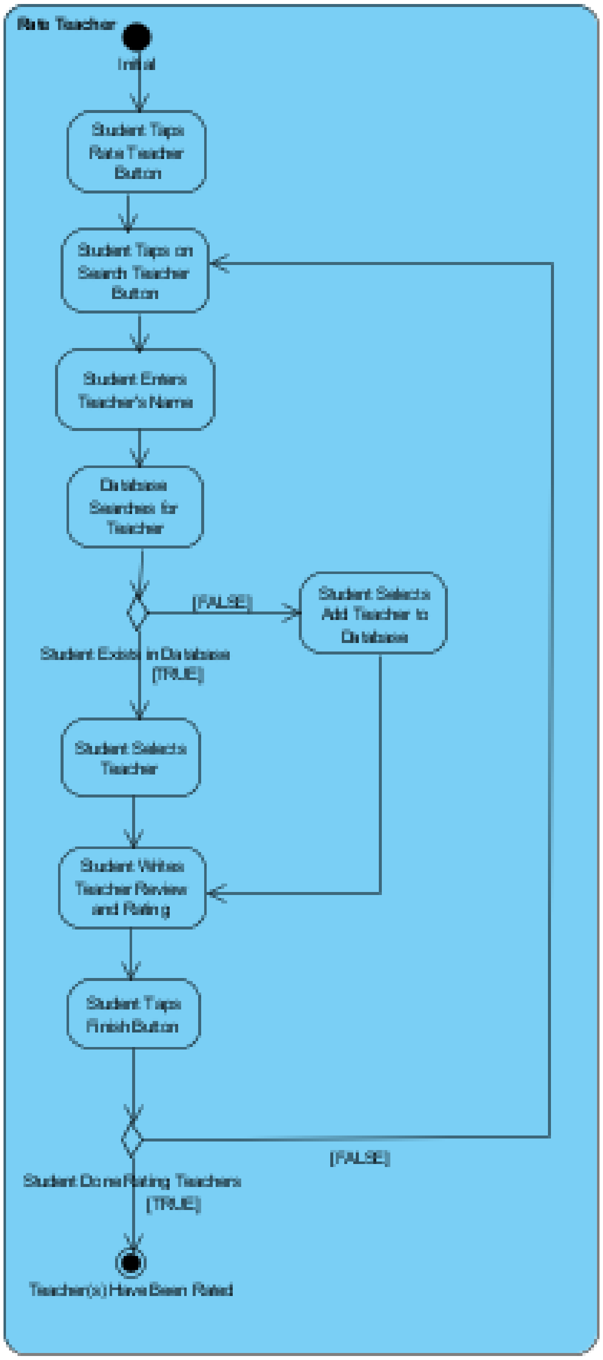
The user is able to access and read reviews about teachers.


View Potential Grades
Initial
Student Selects Course
Student Taps View Potential Grades Button
[FALSE]
Student is Notified that the Syllabus Must be Entered
Student Taps the Back Button
Finished Attempt
Course Syllabus Exists
[TRUE]
Course Syllabus Appears on Screen
Decide What to Check
[Final Grade]
[Remaining Grades]
Student Enters Assignment's Grades Where None Exist
Student Enters Desired Final Grades
System Returns a Hypothetical Final Grade
System Returns Hypothetical Assignment Grades
Student Done Calculating Grades
Student Finished Calculating Grades

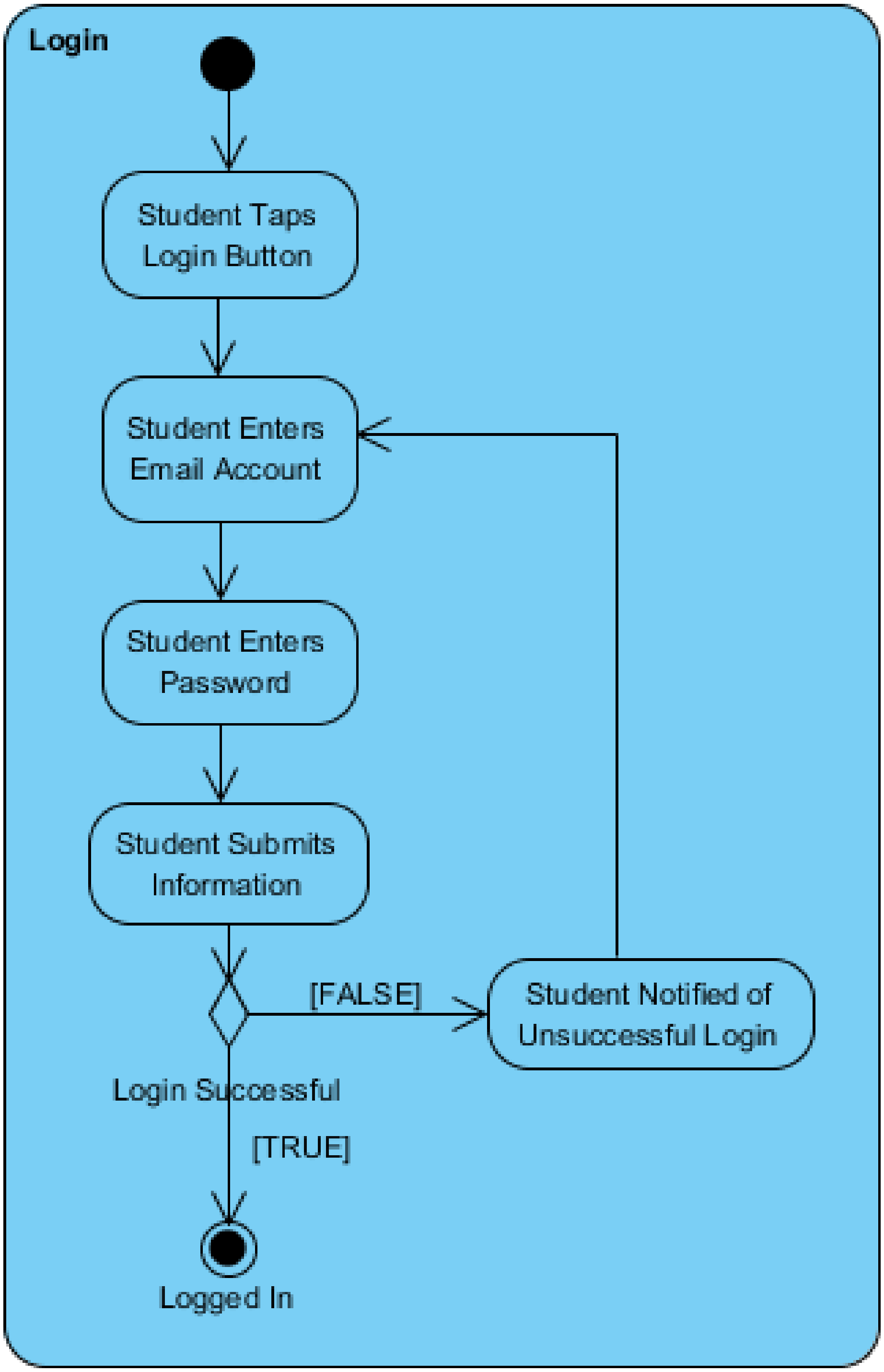
Allows student to see what grades they can potentially make based on his or her performance.


Report Comment
Initial
Student Taps on Report Button Next to Comment
User Asked to Confirm Reporting Comment
[Selects No]
Student Taps Back Button
Comment Unreported
Report Comment Verification
[Selects Yes]
Comment Report Count Incremented
Student's Email is Tagged to Prevent Multiple Reports
[Counter < 5]
Comment Successfully Reported
Report Increment Counter
[Counter > 5]
Comment Deleted

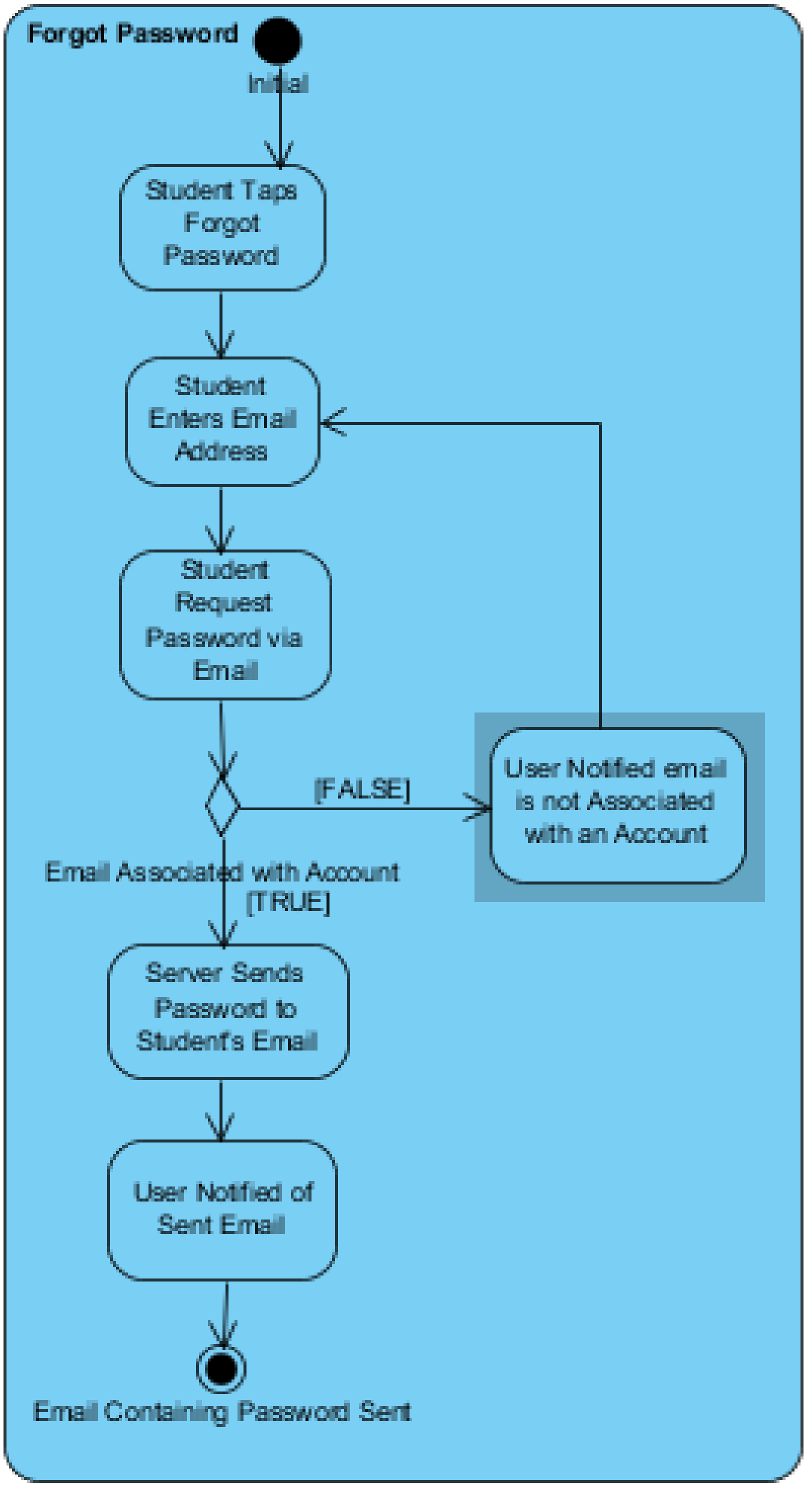
Allows user to report an incorrect or inappropriate review of a teacher.


Rate Teacher
Initial
Student Taps Rate Teacher Button
Student Taps on Search Teacher Button
Student Enters Teacher's Name
Database Searches for Teacher
[FALSE]
Student Selects Add Teacher to Database
Student Exists in Database
[TRUE]
Student Selects Teacher
Student Writes Teacher Review and Rating
Student Taps Finish Button
[FALSE]
Student Done Rating Teachers
[TRUE]
Teacher(s) Have Been Rated

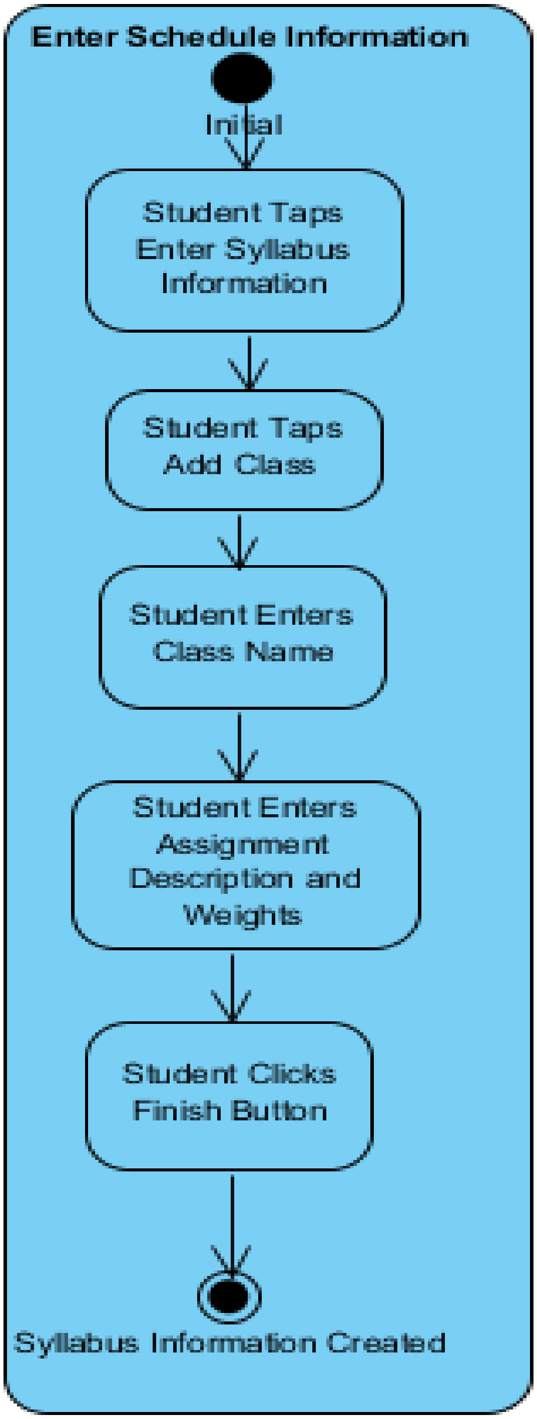
Allows user to rate their professor and add a review of them.


Login
Student Taps Login Button
Student Enters Email Account
Student Enters Password
Student Submits Information
[FALSE]
Student Notified of Unsuccessful Login
Login Successful
[TRUE]
Logged In

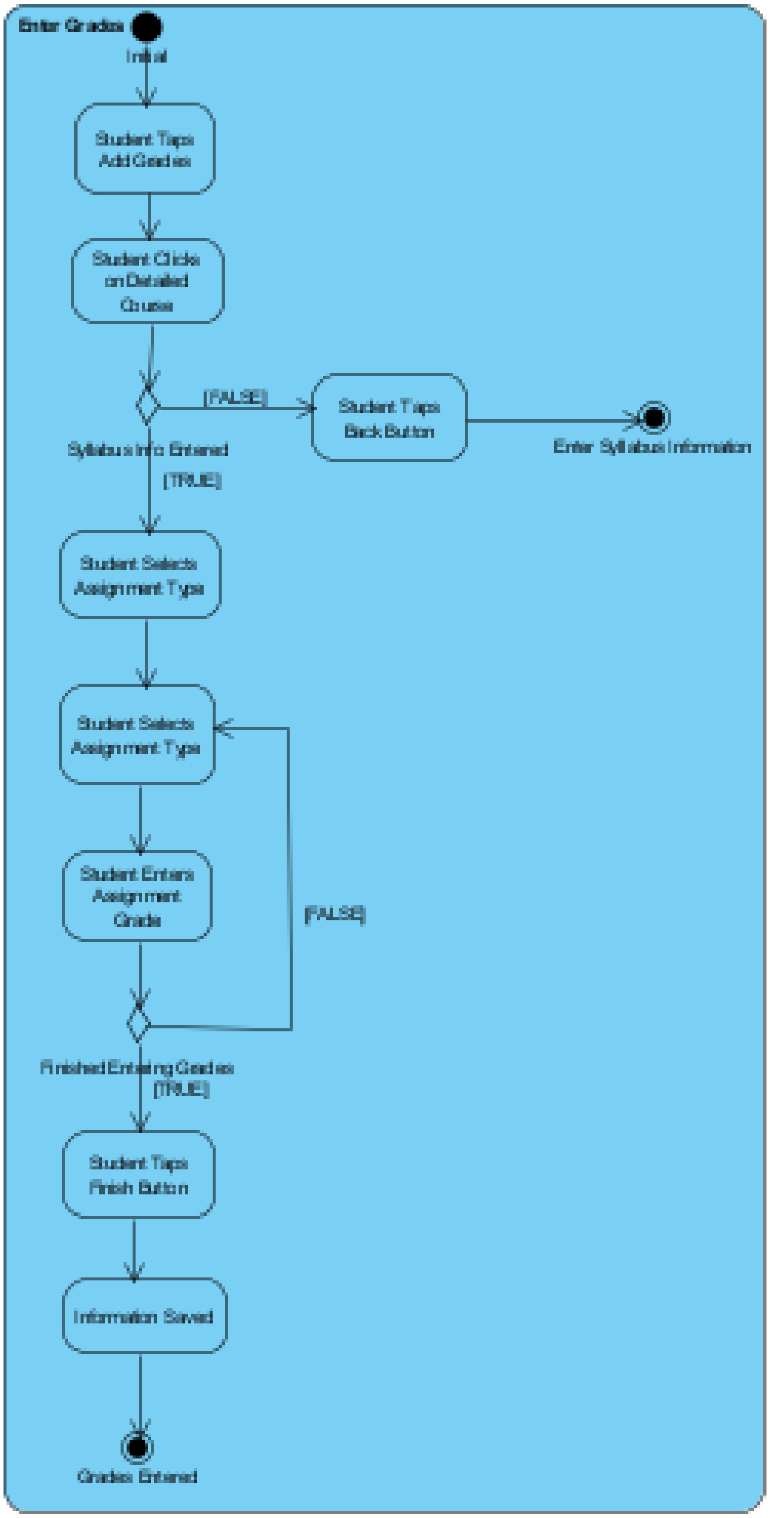
Allow user to login to app to access the app.


Forgot Password
Initial
Student Taps Forgot Password
Student Enters Email Address
Student Request Password via Email
[FALSE]
User Notified email is not Associated with an Account
Email Associated with Account
[TRUE]
Server Sends Password to Student's Email
User Notified of Sent Email
Email Containing Password Sent

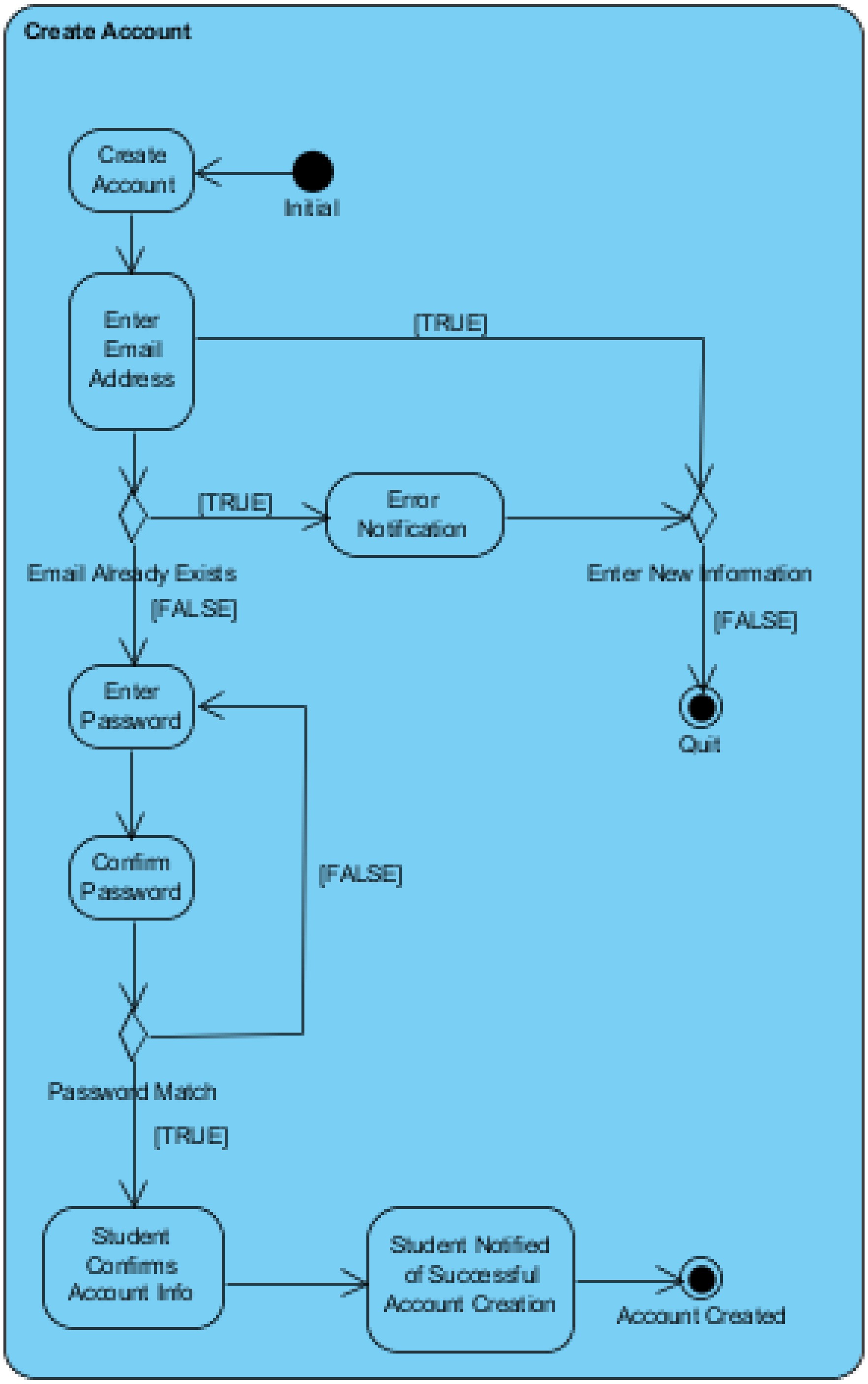
Allow user to reset their password if they have forgotten it.


Enter Schedule Information
Initial
Student Taps Enter Syllabus Information
Student Taps Add Class
Student Enters Class Name
Student Enters Assignment Description and Weights
Student Clicks Finish Button
Syllabus Information Created

Allows user to enter their classes into their schedule.


Enter Grades
Initial
Student Taps Add Grades
Student Clicks on Detailed Course
[FALSE]
Student Taps Back Button
Enter Syllabus Information
Syllabus Info Entered
[TRUE]
Student Selects Assignment Type
Student Selects Assignment Type
Student Enters Assignment Grade
[FALSE]
Finished Entering Grades
[TRUE]
Student Taps Finish Button
Information Saved
Grades Entered

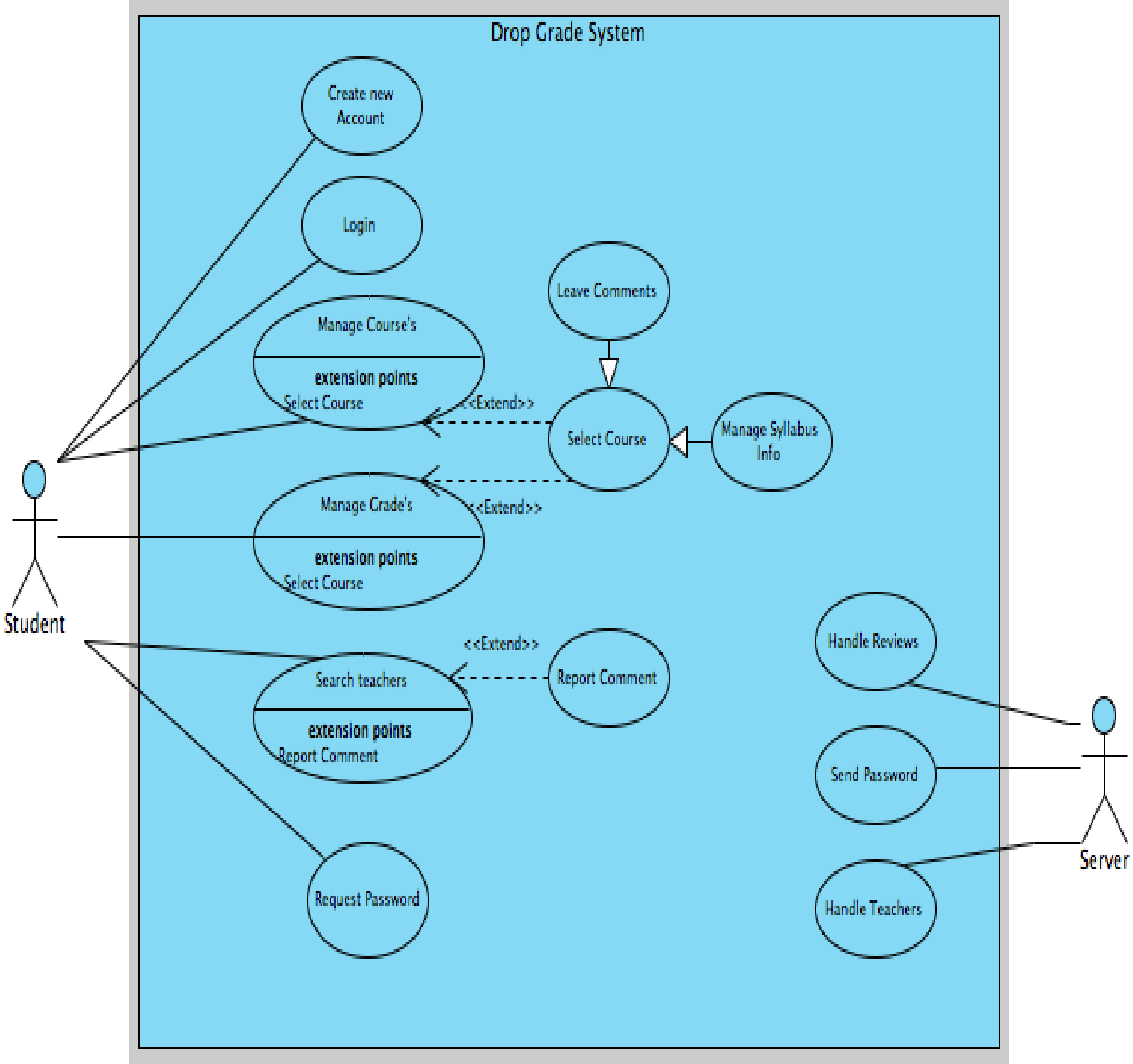
Allows user to enter grades for a class to calculate their grade.



Allows user to create account to begin using the app.

6. Use Case Diagrams

Below are the use cases that our application will be designed from. The design of our classes branched from these use cases. More use cases may be added in the future, however for now it has been decided that these use cases will be what we are focused on satisfying with our design. This will help to focus the team and enforce quality time management. If time allows more use cases may be added to describe the stretch goals listed above.


Drop Grade System
Create new Account
Login
Leave Comments
Manage Course's
extension points
Select Course
<<Extend>>
Select Course
Manage Syllabus Info
Manage Grade's
extension points
Select Course
<<Extend>>
Student
<<Extend>>
Search teachers
extension points
Report Comment
Report Comment
Handle Reviews
Send Password
Server
Request Password
Handle Teachers