Drop Grade

Software Requirement Specifications

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

|  |  |
| --- | --- |
| Table of Contents |  |
| Introduction | 3 |
| Description | 4 |
| Functional Requirements | 5 |
| Class Diagram | 6 |
| Activity Diagrams | 7 |
| Use Case Diagrams | 16 |
|  |  |

**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

1. **Introduction**

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

* 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. **- 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.

* 1. **– 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. **– Actors**
* Students at The University of Alabama (User)
* Server (Database)

1. **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.

* 1. **– Storage**

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

* 1. **– Communication**

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

* 1. **- 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).

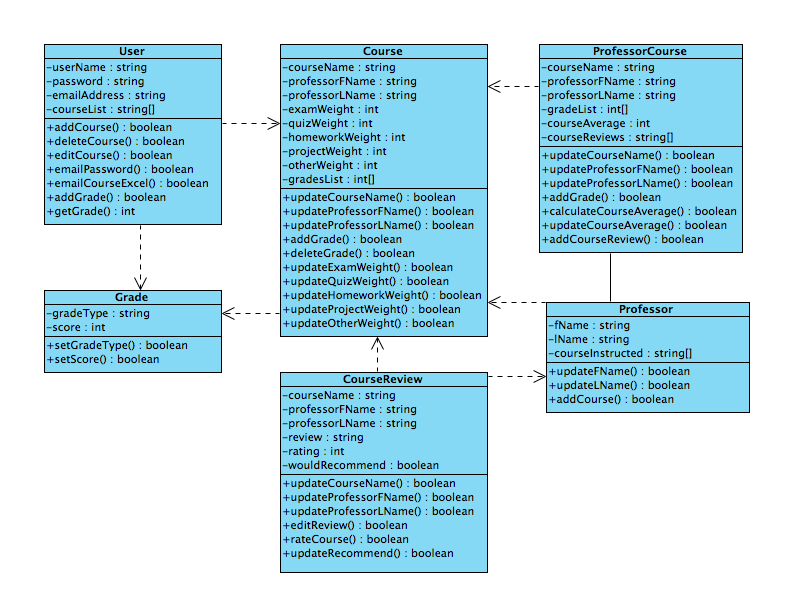
* 1. **- Assumptions**
* Grades will be given in a number format. User reported their grades honestly.
* User has internet access
  1. **- Stretch Goals**
* Reminder function that reminds user of a test or assignment.
* Have list of classes and their respective syllabi stored on server for easy download.
* Export course grade information to an excel spreadsheet

1. **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 |
| Submit Final Grade | Required |
| Manipulate Final Grade via the What-If-View | Required |
| Add 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 |

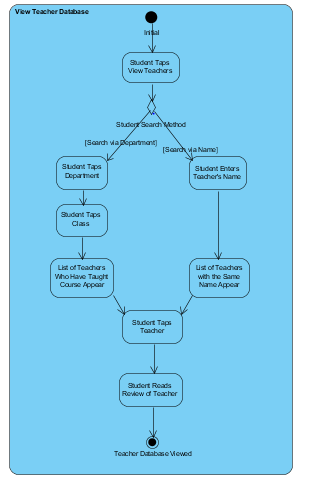
|  |  |
| --- | --- |
| Nonfunctional Requirement | Status |
| Handle multiple users accessing database | Required |
| Allow UA students to register | Required |

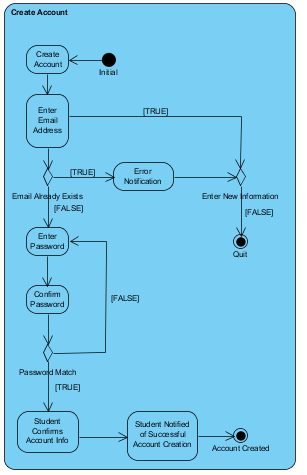
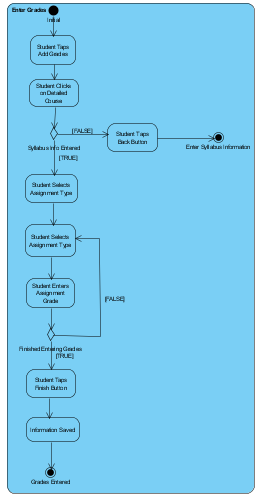
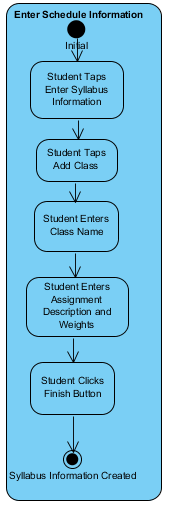
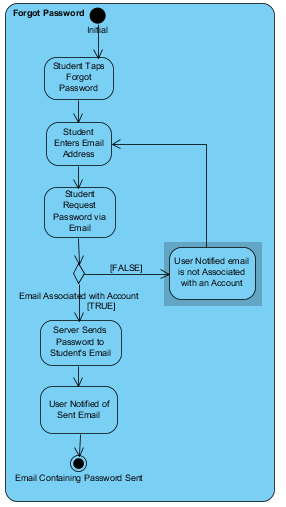
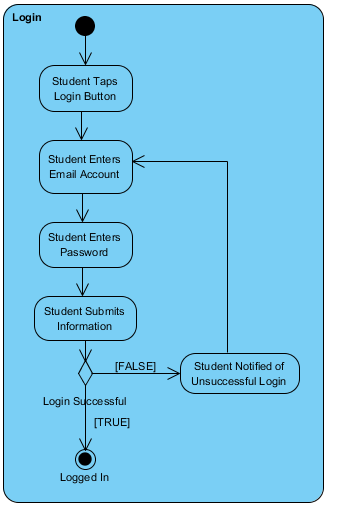
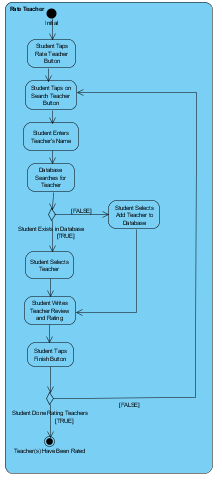
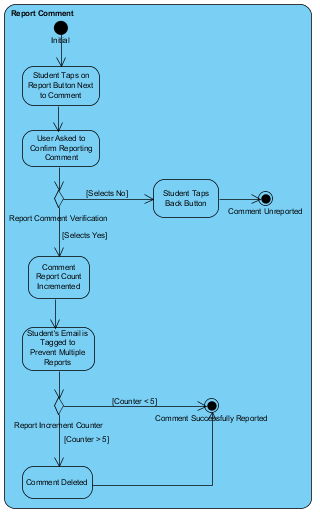
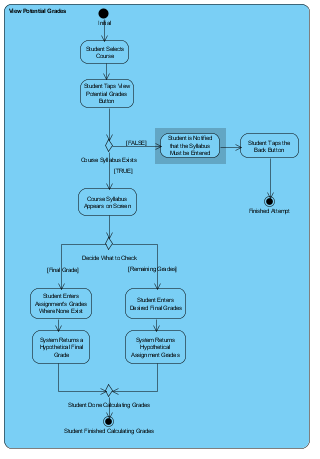
1. **Class Diagram**

****

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.

1. **Activity Diagrams**





1. **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.

