MyGradeHint

Grade Prediction with Collaborative Filtering and Peer Textual Advice

Tyler Collins, Adrian McDevitt, Dillon Wilson, Kuba Wiszowaty

Spring 2020
CS 490-491 Dr. Young Park

Table of contents

What is MyGradeHint?

-MyGradeHint is a software system that is designed to help students better be able to predict their performance in a course

-MyGradeHint works by using a recommendation algorithm called collaborative filtering and peer given advice to give a student the grade they could expect to get in a course.

-MyGradeHint has been developed with students at Bradley University in mind, however will hopefully be expanded to other institutions down the line.

BRADLEY University

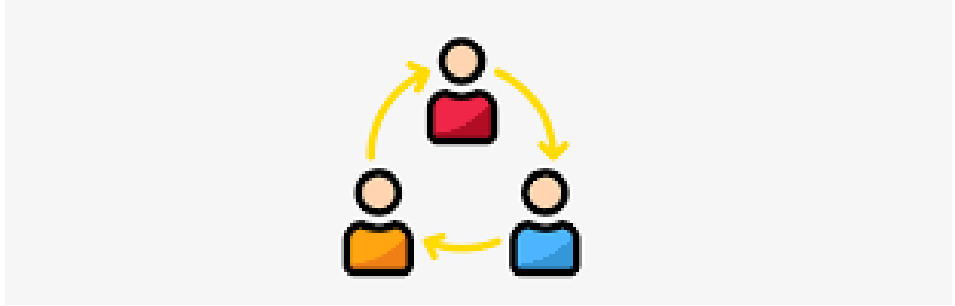
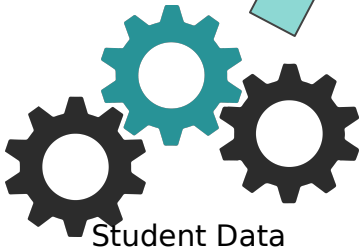


Image sources:
https://www.bradley.edu/
https://www.digitalvidya.com/blog/collaborative-filtering/

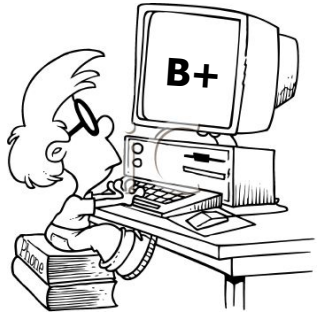
What is MyGradeHint?

Student
Enters Course
Information



Student Data
Processed

System Finds
Similar
Students

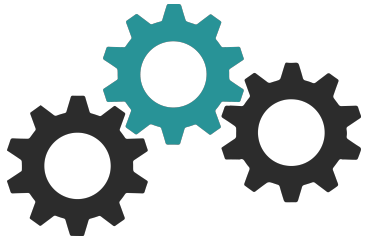


Student
Receives
Grade
Prediction

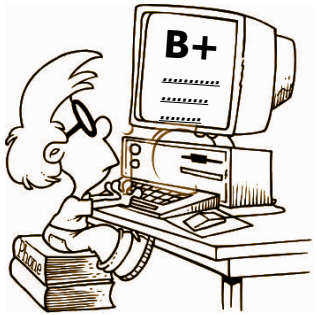
Image sources:
http://www.clipartguide.com/_pages/0511-1111-0813-0150.html
http://clipart-library.com/clip-art/gear-clipart-transparent-7.htm
http://clipart-library.com/large-group-cliparts.html

What is MyGradeHint's Peer Textual Advice System?

Student Enters Grade Received and advice for Peers



Student Data Processed & Stored



Another Student Receives Peer Textual Advice

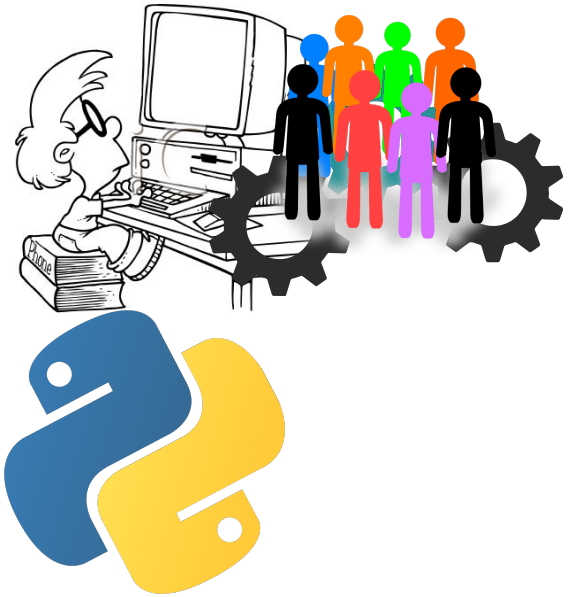
Image sources:
http://www.clipartguide.com/_pages/0511-1111-0813-0150.html
http://clipart-library.com/clip-art/gear-clipart-transparent-7.htm

Software Requirements

1. Project Scope

The users of myGradeHINT will be able to view predicted outcomes for a course that they are considering enrolling in. This system will either be made to work through a web based application like Webster, or some other tool.

The minimum viable product, MVP, will be to get grade prediction (before-grade and after-grade working) working by the end of the 2019 school year. Optional features like course-aware student similarity and sentiment analysis of textual self-reflection on courses will be incorporated if time permits.



Software Size Estimate & Risk Management

	Risk Description	Probability	Impact	Risk Mitigation, Monitoring & Management
1	Allocation of Resources	Low	Small	Time will be the one resource that will need to be allocated efficiently.
2	Less Time & Budget	Medium	Medium	
3	Database Problems	Low	Large	
4	Change in Requirements	Low	Medium	Requirements are well laid out, so changes are not likely.
5	Unexpected Technical Difficulties	Medium	Medium	Any difficulties should and will be resolved in a quick and timely manner
6	Number of Users	Medium	Medium	A sizable database will be required to store user grades for each course.
7	Inexperienced Staff	Low	Large	Most team members are experienced, but if needed, team members will be trained
8	Delivery Deadline	Medium	Large	
9	Browser Compatibility	Low	Small	

Name of phase	Iteration	Estimated Work Hours
Diagrams Phase	Iteration 1	24 Person Hours
Data Cleaning / Gathering	Iteration 2	50 Person Hours
Database Design	Iteration 2	8 Person Hours
Python Model Dev	Iteration 2	24 Person Hours
Front End Dev	Iteration 3	50 Person Hours
Presentation Prep	Iteration 4	8 Person Hours

Languages & Technologies

-MyGradeHint is written in a mixture of different programming languages and libraries within those languages.

Python: Python is a language that boasts one of the most complete machine learning/ AI library of packages of any language. Its Surprise! Package has many built in functions useful for collaborative filtering

HTML: HTML creates the structure of the look and some functions of the webpage.

JavaScript: JavaScript helps with file

Component	Language	Libraries
Prediction System	Python	Surprise! Pandas NumPy
Webpage Front-End	HTML, JavaScript	--
Webpage Back-End	Python, JavaScript	Django

Development Model (Iterative)

-The Iterative model has been selected for MyGradeHint.

The **1st Iteration** is the planning phase.

The **2nd Iteration** is the prototyping phase.

The **3rd Iteration** is the development phase.

The **4th Iteration** is the testing phase.

The **5th Iteration** is the concluding phase.

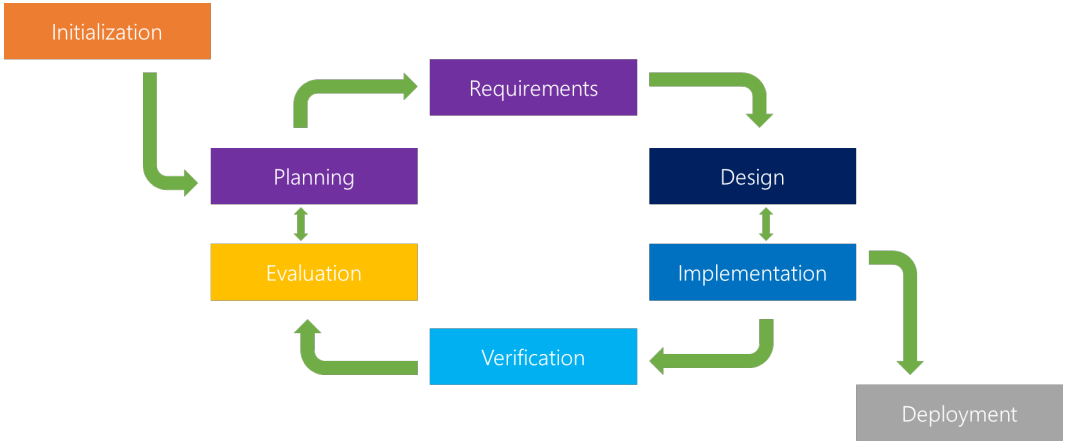


Image sources:
https://airbrake.io/blog/sdlc/iterative-model

1st Iteration

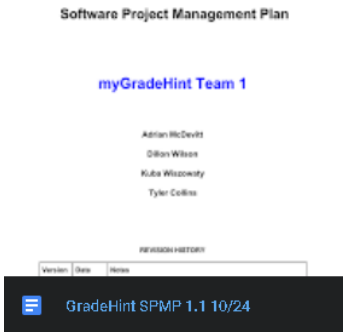
-During the 1st iteration, our team completed the planning stage of MyGradeHint.

-The Deliverables in this iteration were the Software Project Management Plan (SPMP) and the Software Requirements Specification (SRS).

-The SPMP contains software functions, management decisions, a schedule for the project as a whole and a set of estimates and risks.

-The SRS contains technical information

Link to SRS



Link to SPMP

2nd Iteration

-During the 2nd implementation, prototypes and rough versions of implementations were tested.

-We got our first experience with the Surprise! Library for collaborative filtering and with the Django python web framework.

-The deliverables were the Software Design Document (SDD) and the rough prototypes.

-The SDD provides design details regarding MyGradeHint, going over key

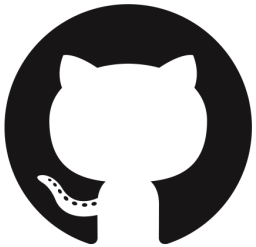
Link to SDD

3rd Iteration

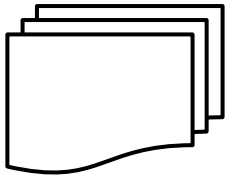
-During the 3rd Iteration, development was done on the main two parts of MyGradeHint, the webpage and the back end prediction system.

-The prediction system was made in Python with Surprise! And was able to accept student data and predict with relatively good accuracy early on in development.

-The Django webpage was able to handle logins, and student reports of grades early on in development as well.



Link to GitHub Repo



Link to Sample Student Data

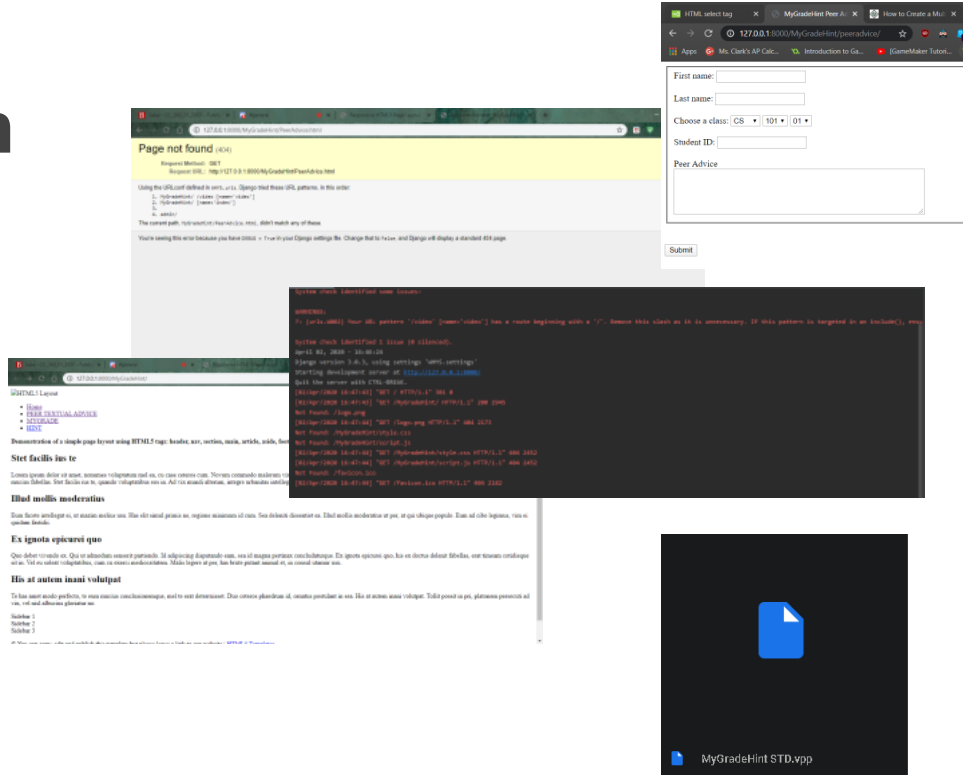
Image sources:
https://github.com/logos

4th Iteration

-During the 4th iteration, testing of the software began. We developed a series of testing conventions to make sure all aspects of the software had been pushed.

-Missing data points, strange inputs on the website, invalid credentials, and testing overall speed were addressed.

-The end product was more accurate, faster, and less fragile than before.

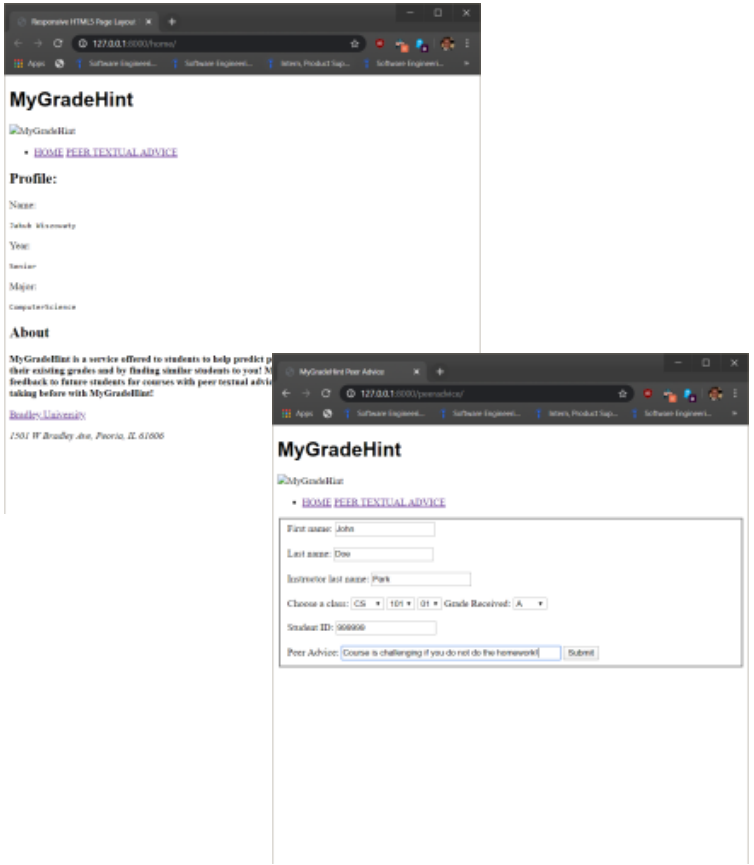


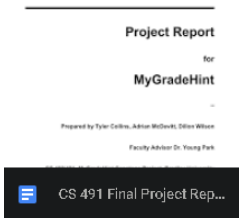
Link to STD

5th Iteration

-During the 5th iteration, we did final project clean up and assembled the final project report and presentation.

-During this iteration, conclusions were able to be drawn and future work was considered/planned. MyGradeHint accomplishes its current task, but as always in software there is room to improve for user satisfaction.





[Link to Final Report]

[Link to GitHub Repo]

Image sources:
https://github.com/logosl

Screenshots

All results were obtained from predicting 5 grades for a random student over 25 trials.

```
####                    ####
##    TESTING PORTION    ##
####                    ####

NUMBER_OF_TESTS = 10

getDatasetInfo()
df_matrix = pd.read_csv("C:\\Users\\Owner\\Desktop\\studentMatrix.csv")

accuracy_list = []
variance_list = []

tic = t.time()

for i in range(NUMBER_OF_TESTS):
    returnList = testCase()
    accuracy = returnList[0]
    accuracy_list.append(accuracy)
    n_neighbors = returnList[1]
    print(f"TEST {i}: \tAccuracy: {round(accuracy, 2)}% \n\t\t\tk_neighbors: {n_neighbors}")
    print()

accuracy_mean = statistics.mean(accuracy_list)

print(f"Average accuracy over {NUMBER_OF_TESTS} trials: {accuracy_mean}%")
print(f"Variance of accuracy from mean: {round(statistics.variance(accuracy_list), 2)}")
print(f"Standard Deviation from the mean: {round(statistics.stdev(accuracy_list), 2)}")
print(f"Time to complete: {str(round(t.time() - tic, 2))} seconds.")
```

```
Average accuracy over 25 trials: 83.56%
Variance of accuracy from mean: 45.09
Standard Deviation from the mean: 6.71
Time to complete: 417.76 seconds.
```

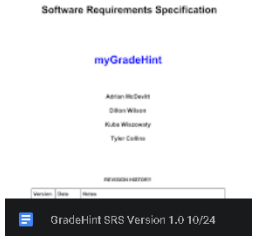
```
Average accuracy over 25 trials: 83.8%
Variance of accuracy from mean: 26.83
Standard Deviation from the mean: 5.18
Time to complete: 401.85 seconds.
```

```
Average accuracy over 25 trials: 82.8%
Variance of accuracy from mean: 27.83
Standard Deviation from the mean: 5.28
Time to complete: 409.25 seconds.
```

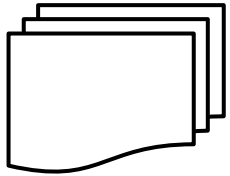
```
Average accuracy over 25 trials: 82.16%
Variance of accuracy from mean: 49.22
Standard Deviation from the mean: 7.02
Time to complete: 372.64 seconds.
```

Deliverables

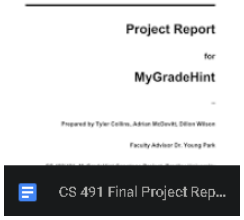
[Link to SDD]



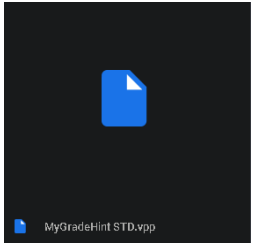
[Link to SRS]



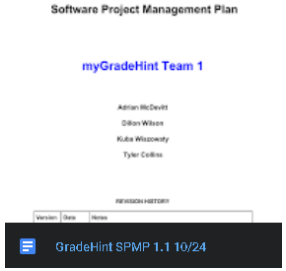
[Link to Sample Student Data]



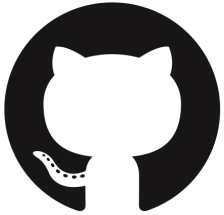
[Link to Final Report]



[Link to STD]



[Link to SPMP]



[Link to GitHub Repo]

*Final Report includes all iteration reports and written

Thank you!

Appendices

SPMP

1 Introduction

1.1 Project Scope

1.2 Major Software Functions

1.3 Performance/Behavior Issues

1.4 Management and Technical Constraints

1.5 Feasibility

2 Project Estimates

2.1 Historical Data

2.2.1 Estimate Technique: Analogy

2.2.2 Software Size Estimate (Effort Estimate)

3 Risk Management

SPMP (cont)

4. Project Schedule

4.1 Process Model, Framework Activities, and Task Set

4.2 Framework Activities

4.3 Task Set

4.4 Task Network

4.5 Timeline

5 Staff Organization

5.1 Model

5.2 Roles and Responsibilities

6. Technologies

6.1 Algorithmic Approach

6.2 Languages

6.3 Development Environments

6.4 Modelling Tools

6.5 Version Control

SPMP (Introduction)

- The goal of myGradeHINT: is to create a grade prediction system to help students better prepare for courses in higher educational institutions.
- A novel, personalized refined grade prediction and peer advice recommendation system
- can predict expected students grades before the course, and the actual grade in the course.
- Provides textual peer advice for the student to succeed in that course.

SPMP (Project Scope)

- Users will be able to view predicted outcomes for a course that they are considering enrolling in.
- This system will either be made to work through a web based application like Webster, or some other tool.
- The minimum viable product, MVP, will be to get grade prediction (before-grade and after-grade working) working by the end of the 2019 school year.
- Optional features like course-aware student similarity and sentiment analysis of textual self-reflection on courses will be incorporated if time permits.
- Completion of the project may require the creation of a database, a user interface to, and a prediction system in Python/R.

SPMP (Project Scope)(cont)

- Predictions will be refined based on the actual grade a student received in the class versus the predicted grade made before the class.
- Textual peer advice from other students that have taken the course will be available to the student to help them better prepare for the course.
- Software will be available to any student that has a valid login to myBU and has access to the internet and Webster.
- The prediction system should not be too complex
  - performance should not be a major concern.
- The data set that is being used will increase in size, affecting performance negatively.
- Grade predictions and textual advice will be in a user friendly format without causing any detriment to the function of Webster as it currently functions.
- Deadline of May 1, time will be valuable to the team.
  - Time should be managed appropriately

SPMP (Feasibility and Project Estimates)

- There are many grade prediction systems that already exist
- They do not predict student grades.
- Based on this information, the project is determined to be feasible.
- We estimate that the algorithm itself will be fairly small because the team has access to libraries in Python to do most of the complex math.
- Cleaning the data so that it will be useful will take some time.

SPMP (Project Estimates) (cont)

Name of phase	Iteration	Estimated Work Hours
Diagrams Phase	Iteration 1	24 Person Hours
Data Cleaning / Gathering	Iteration 2	50 Person Hours
Database Design	Iteration 2	8 Person Hours
Python Model Dev	Iteration 2	24 Person Hours
Front End Dev	Iteration 3	50 Person Hours
Presentation Prep	Iteration 4	8 Person Hours

SPMP (Risk Management)

	Risk Description	Probability	Impact	Risk Mitigation, Monitoring & Management
1	Allocation of Resources	Low	Small	Time will be the one resource that will need to be allocated efficiently.
2	Less Time & Budget	Medium	Medium	
3	Database Problems	Low	Large	
4	Change in Requirements	Low	Medium	Requirements are well laid out, so changes are not likely.
5	Unexpected Technical Difficulties	Medium	Medium	Any difficulties should and will be resolved in a quick and timely manner
6	Number of Users	Medium	Medium	A sizable database will be required to store user grades for each course.
7	Inexperienced Staff	Low	Large	Most team members are experienced, but if needed, team members will be trained
8	Delivery Deadline	Medium	Large	
9	Browser Compatibility	Low	Small	

SPMP (Project Schedule)

- Due to the short deadline of the project, the iterative model would be the best to use.
- The team will produce increments of the software, possibly with a limited set of functionality. The time will then be able to deliver functional increments of the software for review and comments.
- The first increment will take a sample data set...
- The project will follow the framework activities listed below:
  - Communication with client
  - Plan cost, scope, and constraints of the project
  - Design diagrams/analyze models
  - Develop and test code
  - Deploy application

SPMP (Timeline)

Week	Meeting?	Task	Leader	Deliverables
9/30-10/5 5	Weekly @ Lib	Calendar, Leaders, Work Hours Estimate, Software Tools, Version control, Algorithms,	Adrian	SPMP Draft 1
10/6-10/12 6	SPMP Presentation	SPMP	Adrian	SPMP Draft 2
10/13-10/19 7	Weekly @ Lib	Use Case Diagram, Data Flow	Adrian	SPMP Final Draft
10/20-10/26 8	1st Iteration Progress	1st Iteration Report	Adrian	SRS, SPMP Version 1.1
10/27-11-29	Weekly @ Lib	State Transition, GUI Hierarchical, GUI Functional	Adrian	Use Case Diagram, Python Toy Example
11/3-11/9 10	1st Iteration Presentation	1st Iteration Report - Python Prototype	Adrian	Database Design Plan, Previous Diagrams
11/10-11/16 11	Weekly @ Lib	Diagram team review	Adrian	Diagrams finished, review done
11/17-11/23 12	2nd Iteration Progress	2nd Iteration Report	Tyler	STD, GHD, GFD, UCD, DFD, and all other diagrams
11/24-11/30 13	THANKSGIVING	THANKSGIVING	Tyler	Data Collection Strategy
12/1-12/7 14	2nd Iteration	2nd Iteration Report	Tyler	First Batch of Data, ingest into python

SPMP (Timeline)(cont)

12/8-12/14 15	WINTER BREAK WINTER BREAK WINTER BREAK WINTER BREAK			Finished Data collection, Python Skeleton code, HTML skeleton code, Django setup
12/15-12/21 16				
12/22-12/28				
12/29-1/4				
1/5-1/11				
1/12-1/18				
1/19-1/25	Project Kickoff Meeting- Spring Semester	HTML, CSS, PHP Codeacademy	Kuba	Python model with partial functionality
1/26-2/1	Weekly @ lib	Dreamweaver work + Design	Kuba	Django functionality added to Python
2/2-2/8	3rd Iteration Progress Meeting	3rd Iteration Report	Kuba	Basic Webpage prototype complete
2/9-2/15	Weekly @ lib	HTML, CSS, PHP development	Kuba	Added features to Webpage, Python-webpage coupling begin
2/16-2/22	3rd Iteration Presentation to Clients	Website testing, UI confirmation	Kuba	Webpage and Python working together, user input
2/23-2/29	Weekly @ lib	Develop testing plan, recruit testers	Dillon	List of willing testers for system
3/1-3/7	4th Iteration Progress Meeting	4th Iteration Report	Dillon	Report templates for Testers, Data from initial tests
3/8-3/14	Weekly @ lib	Alpha Testing Period	Dillon	Fixes list from first round of testing
3/15-3/21	4th Iteration Presentation to clients	Beta Testing Period	Dillon	Fixes applied from first round of testing, list of fixes for second round
3/22-3/28	Weekly @ lib	Expo Planning	Dillon	Fixes for second round of testing, expo presentation outline
3/29-4/4	Expo Prep	Materials Planning	Dillon	Expo Presentation, Expo Trifold, Expo script

SPMP (Timeline)(cont)

4/5-4/11	EXPO WEEK	Expo Materials	--	Apply final fixes in separate version
4/12-4/18	Weekly @ lib	Prepare Final release	--	Final MyGradeHint release, and corresponding documentation
4/19-4/25	Final Iteration Progress Meeting		--	Final MyGradeHint Release
4/26-5/2			--	

SPMP (Staff Organization)

- Our team will be using a flat organizational model.
- There will be a team leader that works to keep the team focused on what needs to be worked on at the time,
- The professor is the only person with real authority over the project development
  - Also serving as the customer and will not be present at most team meetings.
- Our development team will not have a single team leader for the entire duration of the project.
  - Rather each team member will serve as team leader for a certain portion of time during the year-long project.
- Specific tasks may be assigned to a team member when convenient but each team member will not have a specific role during development.

SPMP (Technologies)

- For our predictive grading system, we will be using collaborative filtering (CF).
  - Collaborative filtering works by finding “neighbors” to the data we are looking at, essentially finding the most similar other data entries we have to the current entry, and comparing the outcomes of the previous data to what we are currently trying to find
- We will be using libraries in the Python language to implement CF
  - The Surprise library for Python contains built in functions for a variety of recommender systems and will be at the core of our system
- We will be using CSS and HTML for the front end design of the user interface, and then JavaScript and PHP for the backend of the webpage to help it couple with our Python model.

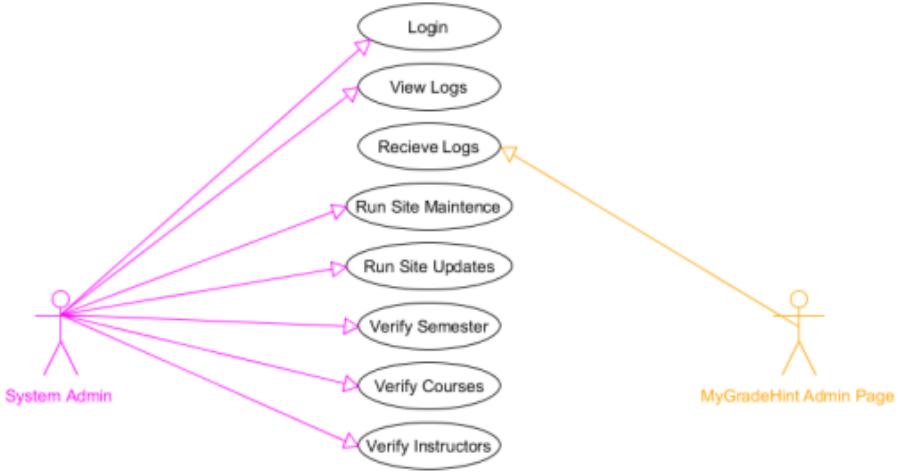
SPMP (Technologies) (cont)

- Within Python, we will be using SciKitLearn & Surprise for our model to help build a system using collaborative filtering.
- We will use Django to help give our Python model connection as a web application, and eventually couple it with the rest of our webpage development.
- For our Python development, we will be using the PyCharm IDE.
  - PyCharm allows us to easily import all the libraries we will need to work with and transform our data, along with the collaborative filtering functions.
- For our HTML, CSS, and PHP/JavaScript development, we will be using Adobe Dreamweaver.

SPMP (Technologies)(cont)

- To create our software diagrams we will use UMLet, Microsoft Visio, Visual Paradigm, SmartDraw, and IBM Rationale.
- We will use the above to create our Use Case, Class Object, Data Model, and all other necessary diagrams as the team sees fit.
- For version control and collaborative development, we will be using GitHub and more specifically, GitHub Desktop.
  - This will help us manage many different versions of our software and the many different languages our codebase.
- For other non code related items, we will be using a shared folder on Google Drive.
  - This will contain our project documentation and records of our meetings.

SRS



System Administrator and Admin Page

MyGradeHint Backend and Database

SRS (cont)

Students and MyGradeHint Website

SRS (cont)

5.3 Use Cases

UC 1. A student can use MyGradeHint to receive a prediction about what grade they might receive in a given course based on their previous performance.

UC 2. A student can use MyGradeHint to receive textual advice from other students about a specific class they are registered for, or considering registering for.

UC 3. The grade prediction system will be able to create a user profile for a student based on their previous performance in classes. This profile will be used for future grade predictions.

5.4 Use Case Descriptions

5.4.1 Student & MyGradeHint Webpage

Students will interact with the MyGradeHint system through a webpage. Students will log in using their MyBu account credentials and they will have access to their past and present grades as well as a course selector where they can look at a course in detail and see their predicted outcome for that course.

5.4.2 Grade Database & MyGradeHint Backend

Perform all prediction calculations and give students recommendations for classes based on a refined algorithm. Protect all student data from unauthorized access.

5.4.3 System Administrator & MyGradeHint Admin Page

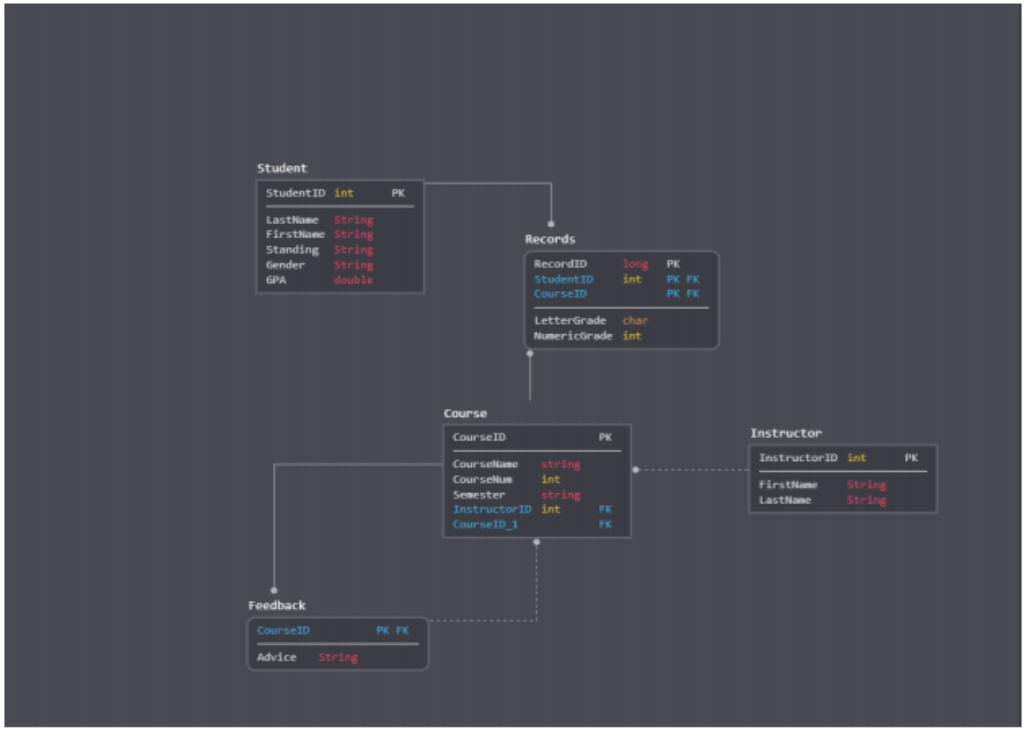
The System Administrator will get a view of all logs showing when data was put into the MyGradeHint system.

5.4.4 Advice Database & MyGradeHint Backend

The Advice Database will connect to the MyGradeHint Backend and provide relevant advice strings when students runs grade recommendations.

SRS (cont)

Data Model



7.0 Traceability Matrix

	UC 1.	UC 2.	UC 3.
FR 1.	X		
FR 2.		X	
FR 3.			X
FR 4.	X	X	X
FR 5.			X