Project Report

for

MyGradeHint

--

Prepared by Tyler Collins, Adrian McDevitt, Dillon Wilson

Faculty Advisor Dr. Young Park

CS 490/491, MyGradeHint Capstone Project, Bradley University

5/1/2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
1st Draft	4/25/2020	Initial Draft	1
2nd Draft	4/27/2020	Added Diagrams, Organization	2
3rd Draft	4/29/2020	Spell Check, Organization, Conclusion	3

1. Introduction

1.1 Purpose

This document contains all documentation produced during the development of MyGradeHint. MyGradeHint is a piece of software to be used at an academic institution to

1.2 Intended Audience and Reading Suggestions

This document is intended for use alongside the development and consulting process. The customer as well as the developers should be familiar with this document. Reading is recommended as a team before steps during the project or testing.

2. Software Project Management Plan

2.1 Introduction

The goal of myGradeHINT is to create a grade prediction system to help students better prepare for courses in higher educational institutions. myGradeHINT is a novel, personalized refined grade prediction and peer advice recommendation system that will be able to predict a student's expected grades before the course, and the actual grade in the course. Along with this it will provide textual peer advice for the student to succeed in that course.

1.1. 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.

Completion of the project may require the creation of a database to store significant data to making predictions, a user interface to present the predictions to the user, and a prediction system in a programming language like Python/R.

Major Software Functions;

The user will be able to view grade predictions for courses they are potentially interested in enrolling in for the next semester. Predictions will be refined based on the actual grade a student received in the class versus the predicted grade made before the class. Alongside the pure grade predictions, 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.

This software system will be available to any student that has a valid login to myBU and has access to the internet and Webster.

Performance/Behavior Issues;

The prediction system should not be too complex so performance should not be a major concern. Depending on the user base, the data set that is being used will increase in size, affecting performance negatively. The user should be able to view their grade predictions and textual advice in a user friendly format without causing any detriment to the function of Webster as it currently functions.

Management and Technical Constraints;

Since the project has a deadline of May 1, time will be valuable to Team 1. Therefore, time should be managed appropriately in order to ship the project out on time and in working condition.

1.2. Feasibility

There are many grade prediction systems that already exist, there are also plenty of prediction systems that operate on the same principles but do not predict student grades. There is an abundance of research on how these systems function but there are currently no public grade prediction systems. Based on this information, the project is determined to be feasible.

Project Estimates

2.1. Estimate Techniques

2.1.1. Software Size Estimate (Effort Estimate)

Because of the ability for the team to use libraries in Python to do most of the complex math for the prediction system, we estimate that the algorithm itself will be fairly small. There will need to be a decent amount of work to clean the data so that it will be useful.

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

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	

Project Schedule

Process Model, Framework Activities, and Task Set

Due to the short deadline of the project, the most appropriate model for this project would be the iterative model. By using this model, Team 1 will produce increments of the software, possibly with a limited set of functionality. Team 1 will then be able to deliver functional increments of

the software for review and comments. The first increment will take a sample data set...

Framework Activities

The project will follow the framework activities listed below:

1. Communication with client
2. Plan cost, scope, and constraints of the project
3. Design diagrams/analyze models
4. Develop and test code
5. Deploy application

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
10/27-11-2	Weekly @	State Transition,	Adrian	Use Case Diagram,

9	Lib	GUI Hierarchical, GUI Functional		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 Presentation	2nd Iteration Report	Tyler	First Batch of Data, ingest into python and model
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	3rd Iteration Report	Kuba	Basic Webpage prototype complete

	Progress Meeting			
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**
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**		**--**	**Final MyGradeHint Release**

	Meeting			
4/26-5/2			**--**	

Staff Organization

Model
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, but there is no hierarchy in authority between team members. The professor is the only person with real authority over the project development but they are serving as the customer and will not be present at most team meetings.

Roles and Responsibilities
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.

7 Technologies

7.1 Algorithmic Approach

For our predictive grading system, we will be using collaborative filtering. 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.

7.2 Languages

To implement our system using collaborative filtering we will be using libraries in the Python language. The Surprise library for Python contains built in

functions for a variety of recommender systems and will be at the core of our system.

To implement the webpage part of our application, 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.

Within Python, we will be using k Nearest Neighbor 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.

7.3 Development Environments

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. PyCharm also allows us to couple directly to GitHub.

For our HTML, CSS, and PHP/JavaScript development, we will be using Adobe Dreamweaver. Dreamweaver has a large pool of online resources that will allow the team to learn it quickly and its large number of third party packages will help us customize the webpage for the best functionality.

7.4 Modelling Tools

To create our software diagrams, we will use a variety of tools for each different kind of diagram. 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.

7.5 Version Control

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.

3. Iteration 1

Deliverables Completed: SPMP, SRS

Software Project Management Plan

Overview of the system as a whole, goes into detail about all of the most important aspects of MyGradeHint. This Software Project Management Plan (SPMP) includes information about the project scope, the major software functions, performance and behavioral issues, management and technical constraints, and the feasibility of the project,

The SPMP includes estimates pertaining to the size of the final software and compares our system to some historical examples to get a better idea of the full scope. Following this is the risk management plan which outlines the biggest problems that could occur during development and how to prevent these risks from significantly impacting the project. A detailed timeline of all tasks and iterations from the beginning of the project until the final presentation are also outlined here. The technologies used for MyGradeHint are detailed as well as what environments will be used.

Software Requirements Specification

The Software Requirements Specification (SRS) includes detailed requirements about the exact functions and subsystems of the final product. Functional and Non-Functional requirements are laid out in this document, explaining features that must be part of the final system.

The SRS contains diagrams, including the Use Case Diagram, and ERD. The Use Case Diagram outlines how different actors will interact with MyGradeHint, including students and administrative staff. Use Cases are detailed in the section below the diagram. The data model was built in SQLDBM and shows the way we will design the database to store the grades, peer advice, and other attributes needing to be stored.

Both of these documents sum up iteration 1, and they together create our project plan and design. Iteration 2 will focus on prototypes and more detailed diagramming, including specific functions and technologies.

4. Iteration 2

Deliverables Completed: SDD, Prototypes

Software Design Document

The SDD (Software Design Document) provides design details regarding MyGradeHint, going over the key features and framework of the final MyGradeHint system. Discussed were elements such as architectural design, interface design, and database design. The architectural design details the hardware and software used in the creation of MyGradeHint, including platforms it will be available on and hardware needed to host and maintain the system. It also outlines the different layers for users, from students to administration. The interface design features functional views of the GUI of MyGradeHint. These views are rough prototypes that will be replicated later on in Django using custom CSS and HTML and included tools in Django.

Prototypes

The two prototypes we prepared in Iteration 2 were one Collaborative Filtering prototype in python and one Django prototype also in python. We made them in pycharm and Jupyter notebook.

The Django prototype was a simple proof of concept, and followed an online guide provided by the jetbrains Pycharm professional IDE for Django. Following the guide, Adrian and Kuba were able to prepare a basic web server with a survey system and a prompt for students to enter feedback and have the feedback stored in a SQLLite database.

The Collaborative Filtering prototype was implemented by Dillon and Tyler using Surprise! Libraries and a few helping libraries such as pandas and numpy. This prototype used the movie ratings dataset found online and was able to create basic

recommendations using this dataset. Applying a grades dataset to it would be simple and it provides a nice framework for the team for Iteration 3 in terms of accuracy tuning and data processing.

5. Iteration 3

Deliverables Completed: Django Webapp, Python Collaborative Filtering Prediction App

In the 3rd Iteration, the team worked on getting the prediction system developed to a point where it can predict a grade for a given course with our dataset. Work also went into getting the Django web framework more developed in an attempt to prepare to integrate the two main components of the system.

Heading into the 4th iteration, we will be implementing a more sophisticated prediction system using Surprise that will be able to consider the course work that has only happened previously to the date we are looking at. We will also be starting preparations for our presentation at the Bradley Expo in April.

6. Iteration 4

Deliverables: STD (Acceptance Testing)

For the 4th Iteration, the team worked on testing and improving the functionality of the algorithm and website. While a lot of testing was done by the team members, we also shared the website with peers who pointed out areas where critical errors occurred. We tested the Reccomender system with different sized datasets in order to better optimize its speed. We are also adding to the website with HTML for both functionality and appearance.

Our goal is to complete the website and integrate the Reccomender system into it such that we have a fully-functional platform.

7. Iteration 5

Deliverables Completed: Capstone Project Final Report, MyGradeHint Webapp and

Grade Prediction App, MyGradeHint Final Presentation

Capstone Project Final Report

This is the final report for the capstone senior project, and it contains all the written and visual deliverables from the course of the project, as well as a conclusion and recommendations for future development.

The specific pieces of the report are the Client Business Needs, SPMP, Iteration 1-5 reports, Conclusion, Future Work, and an appendix that includes diagrams from the project.

MyGradeHint Webapp

The MyGradeHint Webapp is built on the django python framework. It currently is a webpage with a login page and an area for a user to provide and include peer advice in their feedback and can export this data into a form that can be added to our database for use in grade predictions.

MyGradeHint Prediction App

The MyGradeHint prediction app is a python based application using the surprise libraries to predict student grades using collaborative filtering. It currently can look at our existing database and from that be able to draw predictions for any one student. A live demo of this app and the webapp are to be shown during the presentation at the end of iteration 5.

MyGradeHint Final Presentation

The final presentation goes through the deliverables and the contents of the final report in a powerpoint format. It was made to be within the 30 minute time and covers

everything done by the team during the course of the semester. It is split into a part that covers documentation and requirements, and a part that includes screenshots from the program at work.

8. Software Requirements Specification

8.1 Introduction

For this project, the MyGradeHint Team will be serving as the developers in a startup company; Professor Park will be serving as the client and main stakeholder. After the software system has been created, it will likely we tested first with students in the CS department. In the future, it is intended that a wide range of students will use this system to help them plan their sequence of college courses.

8.2 Functional Requirements

FR 1. The system shall be able to make a refined, grade prediction about how a student will do in a given class before and after taking it based on their performance in previous, similar courses.

FR 2. The system shall be able to record textual self-advice from a student about how a course went. This personalized advice will be used to help other students better prepare for this course by addressing their specific needs and requirements for success.

FR 3. The system shall use a denormalized database used to store a large amount of raw data on students. This data will be what our prediction model is built from.

FR 4. The system shall have a system implemented to verify users credentials with a login function and logout function. This user profile will keep records of academic performance which will be used to help make future predictions by comparing new courses with those that data already exists on.

8.3 Non-Functional Requirements

NFR 1. The system shall function on all modern browsers; Google Chrome, Mozilla Firefox, Safari and Microsoft Edge.

NFR 2. The system shall be understandable for users with and without background knowledge in Computer Science or Statistics. Information should be presented in a way that is non-ambiguous.

NFR 3. The system shall be available for use whenever normal Bradley web services are available. Planned outages should be communicated to users ahead of time.

NFR 4. Performance shall be a concern but will likely not be a major issue given the size of our dataset and the complexity of the calculations. Results will be affected by the speed of an internet connection in certain situations as well.

8.4 Other Constraints

8.5 Use Case Model

8.5.1 Actors

a. Students: Any student using the program
b. Grade Database: Contains the grades of the courses the student has entered
c. System Administrator: An individual that maintains the program and the databases
d. Advice Database: Contains various pieces of curated advice for the student
e. Course Database: A database that contains all courses in the CS/CIS depatment

8.5.2 Use Case Diagrams

Students and System Admin

8.5.3 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 personalized textual advice from other students about a specific class they are registered for, or considering registering for.

UC 3. An admin can add or remove a course from the database if the course catalog changes in between semesters.

UC 4. An admin can make changes to the Instructor table to make changes or updates as needed.

UC 5. An admin can make any other changes to the database as a whole as needed.

UC 6. 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.

8.5.4 Use Case Descriptions

8.5.4.1 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.

8.5.4.2 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.

8.5.4.3 System Administrator & MyGradeHint Admin Page

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

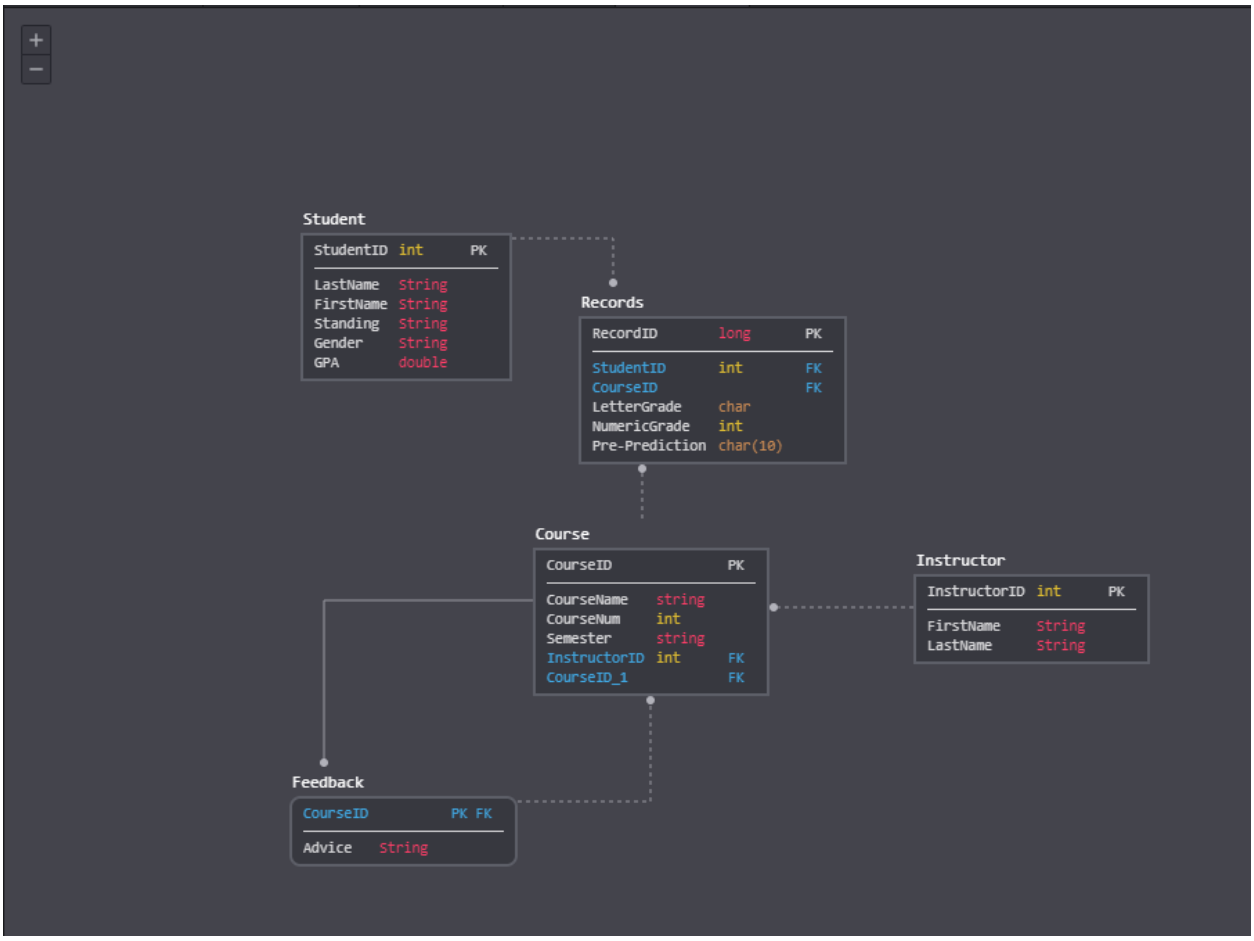
8.5.4.4 Advice Database & MyGradeHint Backend

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

8.5.4.5 Course Database & MyGradehint Backend

The Course Database will be regularly updated each semester and loaded into MyGradeHint to provide students with accurate data regarding available and ongoing courses. This will be a scheduled upload approved by the System Administrator.

8.6 Data Model



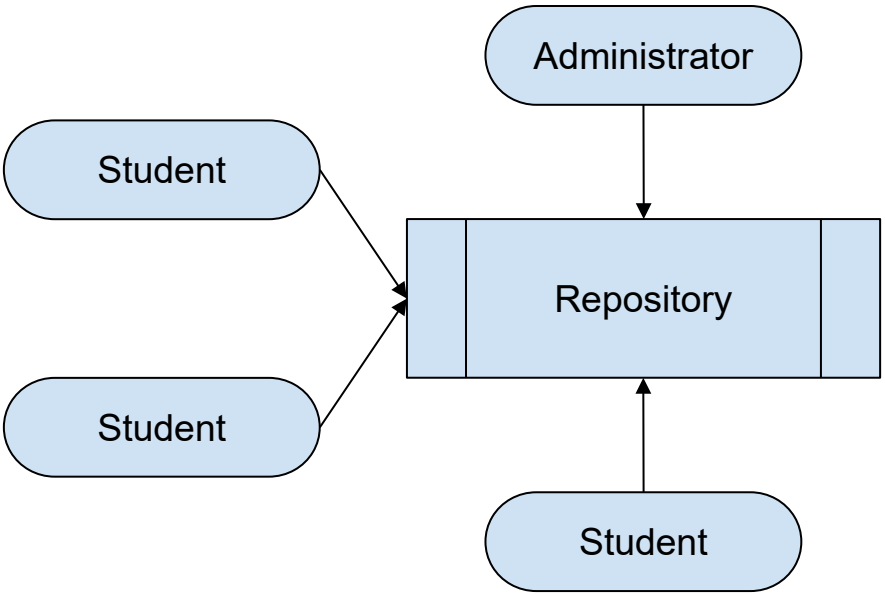
9. Software Design Document

9.1 Architectural Design

9.1.1 Hardware Architecture:

Our software system will use a Data-Centered architecture design. With a large data store at the center housing records, references and other

information and clients accessing the data stored within. More specifically, we will use a repository style. The data store will be passive and requests for changes will be done by the user. For example, an administrator can add a record to the course table. Only certain users will have the ability to make changes to the data, and others will have read-only access.



9.1.2 Software Architecture:

Our software system will be a web based application. With users having access to certain functions based on their permissions and a database that will return information to them. A navigation system will be used to get a user to their desired page easily. Our system will be connected with a web server that will query the database server in order to get the information the user requires.

Student Access Layer:
Administrator Access Layer:

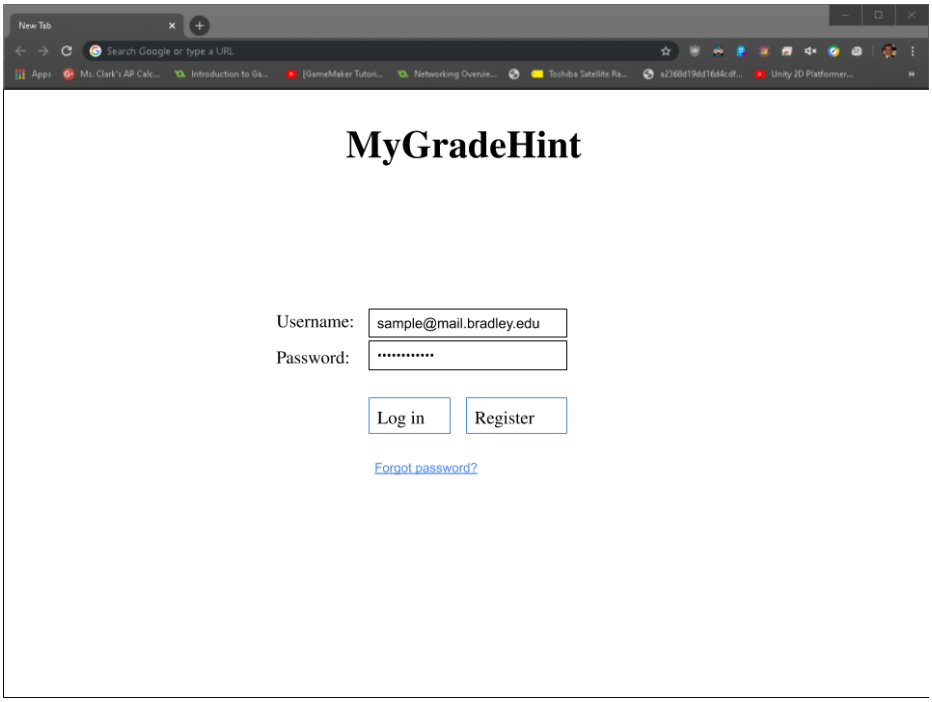
Database Layer:

9.2 Interface Design

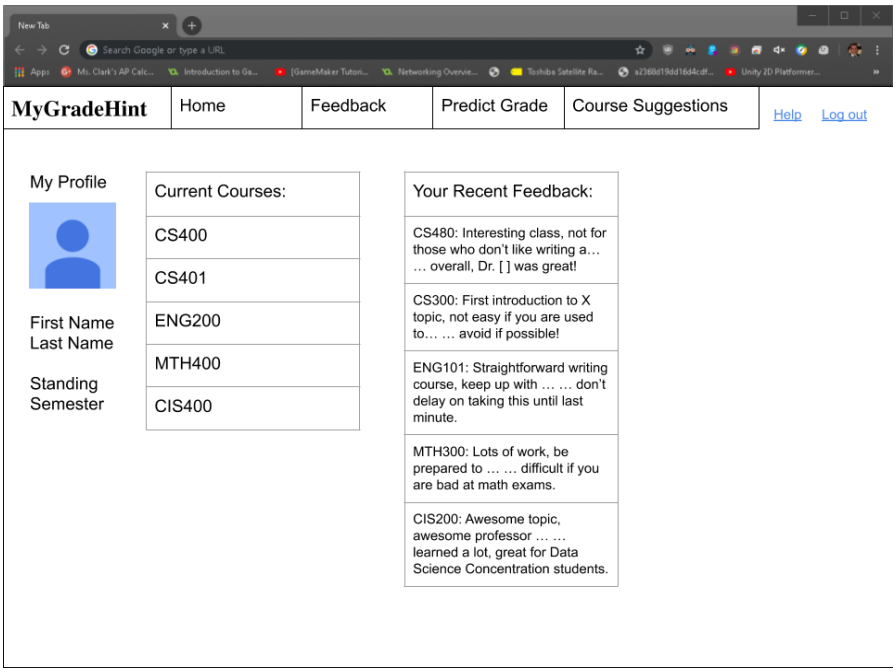
9.2.1 User Interface Specifications (STD, UI Design Mockup)

MyGradeHint's webpage interface is discussed in more detail in this section, which includes rationale and prototype views of what the User Interface may look like.

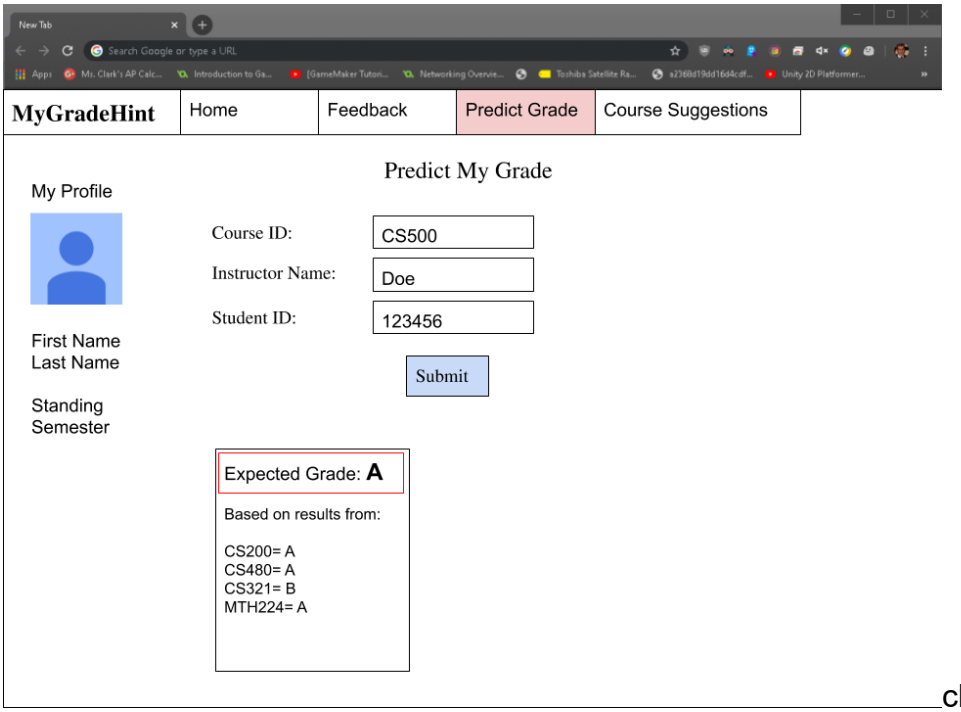
9.2.1.1 Login Screen



9.2.1.2 Home Page



9.2.1.3 Predict Grade Page



chuic

10. Conclusion

10.1 Outcomes

MyGradeHint works with a prediction application and a webapp that allows new data to be put into our database. The two apps work alongside each other to give a student a prediction on their performance in a class they are thinking of taking. This is the goal we set out to accomplish at the beginning of the project and we have been able to get both parts working.

Over each iteration, the project grew more complete and the iterative model suited the development very nicely. From following our documentation in planning to the final product, each iteration provided a new level of depth to the project.

The final product is able to predict student grades at above an 80% accuracy with our fairly small dataset. The website is able to be locally hosted, but could be easily set up on a service such as Heroku for final deployment to students at Bradley. The webpage was also able to handle and store student information and export the students id, name, grade, instructor, and their peer advice, or feedback to give to other students regarding the course and whether or not they should take it. From a functional standpoint, MyGradeHint has accomplished what we set out to do.

10.2 Future Work

The current version of MyGradeHint has an accuracy a little lower than we might like when it comes to user confidence in the software. Future work might include finding larger student datasets or sourcing students in surveys to get more data to obtain better predictions. With collaborative filtering, more data points usually leads to better predictions as there are more students that can be seen as “similar” to the current student.

The webpage currently exists in a barebones look, even though it contains the functionality it needs. Students may be hesitant to use a site that looks very bare. Adding a CSS style element, or images and logos for MyGradeHint and perhaps logos and images from Bradley would help make the webpage a little more appealing.

Connecting to Bradley’s grades database instead of sample data would be the best way of testing MyGradeHint, and it would allow MyGradeHint to allow students to use their Bradley ID and password to log in, similar to most of the other services Bradley offers online. Currently, the student database is local to the machine hosting the website and transferring all the credentials would be a big upside.