Florida International University
School of Computing and Information Sciences

CIS 4911 - Senior Capstone Project
Software Engineering Focus

Final Deliverable

Project Title:
FIU GPA Tracker and Forecaster 2.0

Team Members:
Lizette Mendoza
Alex Sanchez

Product Owner(s):
Johann Henao

Instructor:
Masoud Sadjadi

Abstract

The FIU GPA Tracker and Forecaster is a website that assists students keep up with their academics. It does this by providing detailed information about how they are doing in their classes and what they need to earn to achieve their goal GPA by the of graduation. This information is broken down into manageable semester-long goals. In addition, a small Android app has been created that allows students to view and enter their semester grades. This document covers how these features have been improved in the 2.0 version of this system. Details of the makeup of the system and the functionality will be described in detail.

Table of Contents

INTRODUCTION
- CURRENT SYSTEM
- PURPOSE OF NEW SYSTEM

USER STORIES
- IMPLEMENTED USER STORIES
- PENDING USER STORIES

PROJECT PLAN
- HARDWARE AND SOFTWARE RESOURCES
- CODING NAMING CONVENTIONS AND STANDARDS
- SPRINTS PLAN
  - *Sprint 1*
  - *Sprint 2*
  - *Sprint 3*
  - *Sprint 4*
  - *Sprint 5*
  - *Sprint 6*
  - *Sprint 7*

SYSTEM DESIGN
- ARCHITECTURAL PATTERNS
- SYSTEM AND SUBSYSTEM DECOMPOSITION
- DEPLOYMENT DIAGRAM
- DESIGN PATTERNS

SYSTEM VALIDATION

GLOSSARY

APPENDIX
- APPENDIX A - UML DIAGRAMS
  - *Static UML Diagrams*
  - *Dynamic UML Diagrams*
- APPENDIX B - USER INTERFACE DESIGN
- APPENDIX C - SPRINT REVIEW REPORTS
- APPENDIX D - SPRINT RETROSPECTIVE REPORTS

REFERENCES

INTRODUCTION

The FIU GPA Forecaster and Tracker is a website that helps Florida International University students achieve their desired GPA before graduating. The main goal is to automate tedious tasks and get users motivated to do better, both for themselves or for the good of the university. The system will allow the user to define how well they want to do in certain courses and takes their opinion into account to advise the student what grade they should get by means of forecast reports. This opening section will begin with a discussion of the current system and a description of the new system.

The remaining document will continue as follows: all the user stories accomplished as well as pending stories are featured. Following that is a breakdown of how the work was accomplished and the resources required to achieve this. Details about the system's design is discussed in the next section. All the tests conducted on the system is described in detail, followed by a section on the glossary of terms used throughout the document. Lastly, the appendix contains diagrams, screenshots and reports.

Current System

The FIU GPA Tracker and Forecaster 1.0 was created to aid students in reaching their desired GPA goal by the time of graduation. Currently, it does so by allowing students to import their PantherSoft GPA Audit to upload all their courses, those completed and those remaining. Students are then able to monitor the progress of their GPA over the course of their semesters at FIU. For the courses in-progress and remaining, students can select a weight and relevance value for each course to select the difficult level and the level of importance the student feels the class has. Students can also log their grades into the system for their current courses. They can then monitor the progress for the semester.

Purpose of New System

The purpose of the 2.0 version is to take things one step further by allowing students to see what they need to attain their GPA goal based on the information they input into the system. This will be done by means of forecast reports. The reports will show the student what averages/grades they need to maintain in each of their remaining courses based on the weight and relevance selected. It will also display a graph for the student to visually see how they're doing with their goal and what their projected outcome might be based on their performance thus far.

The admin user will also be given more privileges to better handle the curriculum information and student data. The system will be updated to perform faster and have a newer design to be more appealing to the users. Also, error logging features will be improved to increase the system's maintenance.

USER STORIES

This section will include the user stories of utmost importance to the product owner. However, due to time constraints, not all the stories were implemented. The stories under the first subheading were completed throughout the seven Sprints. The next subheading includes stories that will hopefully be tackled in the near future.

Implemented User Stories

User Story #803 Complete JavaScript Tutorial

As a web developer, I would like to complete the JavaScript tutorial on codeacademy.com to incorporate in the project.

User Story #804 Complete PHP / Python Tutorials

As a web developer, I would like to complete the PHP and Python tutorials on codeacademy.com to contribute this skill in the project where applicable.

User Story #805 Setup Local Environment

As a web developer, I want everything set up my local environment to develop the project on my local machine.

User Story #828 Add Semester Forecast Functionality

As a student, I would like view what grades I need for the courses in-progress this semester in order to achieve my Goal GPA in the remaining semesters.

User Story #823 (Teamwork) Create Error Logging Interface

As a web developer, I want to be able to log any errors or important information to a text file to be able to review at a later time.

User Story #811 Troubleshoot GPA Audit Import

As a web developer, I would like to fix any bugs related to the GPA Audit import so all courses show up properly in GPA dashboard.

User Story #809 Create Formatted Semester Forecast Report

As a student, I would like to generate a formatted forecast report to allow me to view a printed report of what I need to achieve for the current semester.

User Story #827 Create Error Log System

As a web developer, I would like the error log interface to have concurrency so that the user doesn’t experience delays when the system is writing to the log.

User Story #846 DEFECT: Fix Data Points and Calculations
As a student, I would like to view a graph with all courses displayed in my Semester Forecast Report so I can see how I progress over the course of the semester.

User Story #852 DEFECT: Fix Graph in Semester Dashboard
As a student, I would like to view a graph with all courses displayed in the Semester Dashboard so I can see how I progress over the course of the current semester. (Defect from work done in CIS4911 in Fall 2015 semester)

User Story #826 Instantiate Curriculum Requirements
As a student I would like to see the courses I am required to take so I can plan for them and generate my report.

User Story #857 Migrate Semester Forecast Report to Object Oriented Design
As a web developer, I would like the code for the Semester Forecast Report that was designed last semester (Fall 2015) be migrated to Object Oriented design.

User Story #873 Map Admin Curriculum Import
As a system admin I would like to import the curriculum requirements for a major program so that we can generate forecast reports.

User Story #819 Generate GPA Forecast Report
As a student, I would like to generate a report that will allow me to break down my GPA goal over remaining semesters to understand what I must do to achieve my GPA goal by graduation.

User Story #867 DEFECT: Fix GPA Dashboard Graph
As a student, I would like to view a graph with my GPA history since starting at FIU displayed in the GPA Dashboard so I can see how I progress over the course of the semesters. (Defect from work done in CIS4911 in Fall 2015 semester)

User Story #891 Fix Import Curriculum
As a system administrator, I would like to automate the process of importing the curriculum requirements so that it can be quickly done for many curriculums.

User Story #887 Migrate Semester Dashboard Code to OO
As a web developer, I would like the code for the Semester dashboard to be migrated to Object Oriented design.

User Story #847 Migrate Error Log Feature to Object Oriented Design
As a web developer, I would like the code for the Error Log to be migrated to Object Oriented design.

User Story #866 DEFECT Fix JSON Errors on Sem Dash
As a student I would like to be able to add assessments and grades for courses that are in progress. (DEFECT from Fall 2015)

User Story #820 Understand GPA Dashboard Data
As a student I would like to see the courses I have taken, courses I need to take, and my current GPA so that I can check for discrepancies and plan what courses I will take.

User Story #812 Improve Semester Dashboard Look-And-Feel
As a student, I would like to see a more updated look on the Semester Dashboard and for it to run more efficiently.

User Story #895 Update Student's Grades
As a system admin, I would like to add, delete, and update a student's grades to correctly represent their course history.

User Story #899 Update Grad Programs
As a system admin, I would like to add, update, and delete graduate program information to keep it current with FIU.

User Story #893 Improve Breakdown Look-and-Feel
As a student, I would like to see a more updated look on the Assessment Breakdown and for it to run more efficiently.

User Story #886 Project Future Data for Semester Forecast Reports
As a student, I would like a graph that will project what the remaining of my semester may look like the in the current course(s) I am enrolled in to have an idea of the end result.

User Story #885 Project Future Data for GPA Forecast Reports
As a student, I would like to have a graph that will project what my remaining semesters may look like to have a better, visual idea of the end return.

User Story #874 Migrate GPA Dashboard to AngularJS
As a student, I would like the GPA dashboard to have an updated look so that it is easier to use.

User Story #909 Update Weights and Relevance
As a student, I would like to change the weights and relevance of courses so that I can generate a forecast report.

Pending User Stories

User Story #810 Complete Android Application
As a student, I would like to access the GPA tracker as an app because it would be easier on a mobile device than a web page.

User Story #879 Create iPhone/iPad app with same features as android app
As a student, I would like to have the same functionalities as the Android app on an iPhone/iPad.

User Story #880 Make a mobile friendly html site with same functions as Android app.
As a student, I would like the GPA Tracker to be mobile friendly on any mobile device and have the same features as the Android app.

User Story #868 Semester Forecast Report Improvements
As a student, I would like to see a report that will allow me to keep close track of my graduation goal and notify be of my progress so I can achieve my graduation goal.

User Story #881 Complete Help Feature
By using queues, producer, consumer, semaphore techniques already proven in the logging solution, identify rest of potential concurrency issues and resolve them user the aforementioned techniques.

User Story #877 Identify data update concurrency scenarios and resolve them
Per class, add a feature where students enrolled in that class can have a blog, share files, and do text messaging. As an example we can look at piazza.com

User Story #894 Add Profile Widget
As a student, I would like a profile widget so I can easily change settings regarding my user profile.

User Story #875 Finalize Admin interface
As a system administrator, I would like to change the curriculum requirements so that the modeled in the system is up to date with FIU.

User Story #892 Create Pie Chart for Both Forecast Reports

As a student, I would like to see a graph that will allow me to visualize the how many of each grade I need for the courses mentioned to know what grades are required of me to reach my GPA Graduation Goal.

User Story #883 Implement email feature.
As a student, I would like an email feature so I can email my forecast reports, submit questions, etc.

User Story #884 Implement to-do calendar feature for Semester Dashboard
The students should be able to add to do tasks for reminders on important class events such as upcoming tests, homework, quizzes etc.… There will be a to-do list calendar per class with simple functionality such as task title, description, notes, and reminder data/time. Then at the Semester dashboard landing page level the to-do list is the aggregation of all to-do tasks of all courses + at this level the student can add general tasks which may or may not be directly linked to a particular course.

User Story #876 Improve Website Security and proper sessioning
As a student, I would like the website to have better security so that my personal student data and other information can remain safe.

PROJECT PLAN

To work efficiently, we utilized the scrum development method. Each sprint in scrum lasted two weeks in which the assigned user stories were to be completed. Any remaining user stories were either assigned for the following sprint or returned to the product backlog. Below, we describe what recourses the system requires to function, the coding standards we implemented when programming, and the breakdown of our sprints.

Hardware and Software Resources

The following resources were used in the system:

Software

- Oracle VM VirtualBox Manager Version 5.0.14: Used to run Ubuntu 64-bit. Ubuntu was selected as the development environment because Linux platforms are well known to be very stable.
- Front-end languages: HTML, JavaScript, CSS, jQuery. These languages were used to build the web page layout.
- Back-end languages: PHP, SQL, Python. PHP is ideal because it has great support for session variables, post variables, and it interacts with mySQL excellently. Python was chosen because it can use the excellent and easy to use PyPDF parser for the GPA Audit.
- Jetbrains PHPStorm IDE Version: 10.0.3: This platform helped in developing the code for the system. Although focused on PHP, the IDE was useful in coding for all languages.
- PhpMyAdmin Version 4.4.13.1: Used to handle data in the database in an easier manner.
- AngularJS 1.0 Framework: Added dynamic features to HTML. It is also easier to read and quicker to develop.
- Apache 2.4.18: Apache was used as the web server due to its ease of use and stability.

Hardware

The following systems were used in developing the GPA Tracker:

Toshiba Satellite M645-S4118X

- Processor Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz 2.30 GHz
- System Memory: 8.00GB
- Operating System: Windows 10 Pro
- Hard Disk Size: 449GB

Microsoft Surface Book

- Processor Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz
- System Memory: 8.00GB
- Operating System: Windows 10 Pro
- Hard Disk Size: 128GB

Coding Naming Conventions and Standards

1. Camel naming convention. Functions and tables as: Word1Word2, fields and variables as word1Word2.
2. Indentation 3 spaces per indentation

3. Function names should be in the form of actions such as: GetName, ExportData, and ImportData.
4. Line length should not be too long so that it goes out of the screen range, continue in the next line if this is the case.
5. Functions should focus on one activity only and try to keep functions length to no more that it can be read completely in one screen.
6. Add comments on things that are NOT obvious, especially if it is something complex.
7. Refactor soon and refactor often for improving code readability.

Sprints Plan

For each sprint, the list the user stories selected for implementation are laid out in descending order of priority.

Sprint 1

(01/19/2016 - 01/29/2016)

User Story #805 Setup Local Environment

Acceptance Criteria:

- Install all required applications.
- GPA Tracker 2.0 ready for development.

User Story #803 Complete JavaScript Tutorial

Acceptance Criteria:

- Finish tutorial on codeacademy.com

User Story#804 Complete PHP / Python Tutorials

Acceptance Criteria:

- Complete PHP tutorial
- Complete Python tutorial

User Story #807 Review PHP Tutorial

Acceptance Criteria:

- Review tutorial on codeacademy.com

Sprint 2

(02/01/2016 - 02/12/2016)

User Story #828 Add Semester Forecast Functionality

Tasks:

- 830 Testing - Semester Forecast Functionality
- 818 Documentation for GPA Forecast Report

- 817 Develop Forecast Functionality
- 816 Algorithm / Pseudo Code for Forecast Report
- 815 Forecast Report Mock-Up

Acceptance Criteria:

- Student will be notified if GPA Goal can be achieved by graduation or not.
- Student will input weight and relevance for classes currently In-Progress (if not already set).
- Report will display on the screen to notify student how to achieve GPA Goal, if attainable.

Modeling:
Please refer to Figure A-3 in Appendix A for the sequence diagram that was created. Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #823 (Teamwork) Create Error Logging Interface

Tasks:

- 832 Research Standards for Proper Error Logging
- 831 Research Global Functions PHP
- 825 Develop Error Log
- 824 Error Logging Algorithm

Acceptance Criteria:

- Code should log errors, warnings, info, and debugs to file.
- Settings file with error ids and user messages.
- Error messages can be logged from anywhere.
- Will write directly to a file.
- Can be accessed by all functions within the PHP files.

Modeling:
Please refer to Figure A-5 in Appendix A for the sequence diagram that was created. Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #811 Troubleshoot GPA Audit Import

Tasks:

- 822 Implement algorithm
- 821 Develop new algorithm
- 814 Understand existing code

Acceptance Criteria:

- In progress courses are shown in the semester dashboard.
- Courses where student did not meet the min requirements are imported.
- Courses that are not part of the major curriculum, but affects GPA are imported.

Modeling:
Please refer to Figure A-12 in Appendix A for the use case diagram.

Sprint 3

(02/15/2016 - 02/26/2016)

User Story #809 Create Formatted Semester Forecast Report

Tasks:

- 845 Test Semester Forecast Report
- 844 Add Assessments Graph
- 837 HTML Page
- 836 "Generate Forecast" Button

Acceptance Criteria:

- Generate Forecast Report from the Semester Dashboard.
- A new HTML page will be open with a printed report.
- Student should see a table of classes, weights, relevance, minimum grade requirements, secure grade requirements, and estimated study time recommendations.
- Include Semester Assessments graph.

Modeling:

Please refer to Figure A-3 in Appendix A for the sequence diagram that was modified. Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #827 Create Error Log System

Tasks:

- 848 Research PHP Threads
- 843 Modify Development Environment
- 842 Research RabbitMQ
- 841 Research PHP Message Queues

Acceptance Criteria:

- Pages should not crash when waiting to write to the log.
- Interface should be able to handle multiple requests.

Modeling:

Please refer to Figure A-7 in Appendix A for the sequence diagram that was created. Please refer to Figure A-12 in Appendix A for the use case diagram.

Sprint 4

(02/29/2016 - 03/11/2016)

User Story #846 DEFECT: Fix Data Points and Calculations

Tasks:

- 865 Migrate Semester Report to OO
- 854 Add Legend
- 851 Test Graph
- 850 Fix Graph

Acceptance Criteria:

- Graph matches course assessment grades.
- Display legend for graph (for each course).
- Display date ranges along x-axis.

Modeling:
Please refer to Figure A-3 in Appendix A for the sequence diagram that was modified.

User Story #852 DEFECT: Fix Graph in Semester Dashboard

Tasks:

- 863 Test Graph
- 855 Add Legend to Existing Graph
- 853 Fix Graph

Acceptance Criteria:

- Data points begin from the start of the semester.
- Data points do not extend past the end of the current semester.
- Data points are calculated accurately.
- Graph is aesthetically pleasing.

Modeling:
Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #826 Instantiate Curriculum Requirements

Tasks:

- 859 Install PHPUnit
- 858 Create Database Connector
- 849 Migrate Code to OO Design

Acceptance Criteria:

- Needed courses will be displayed in courses needed table on GPA dashboard.
- Courses in database will match GPA audit report.

Modeling:
Please refer to Figure A-6 in Appendix A for the sequence diagram that was modified. Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #857 Migrate Semester Forecast Report to Object Oriented Design

Tasks:

- 864 Test New OO Controllers
- 862 Update UMLs to new OO Design
- 861 Create SemesterForecastController
- 860 Create SemesterDashboardController

Acceptance Criteria:

- Create PHP file for SemesterDashboardController
- Create PHP file for SemesterForecastCastController

Modeling:
Please refer to Figure A-3 in Appendix A for the sequence diagram that was modified. Please refer to Figure A-12 in Appendix A for the use case diagram.

Sprint 5

(03/21/2016 - 04/01/2016)

User Story #873 Map Admin Curriculum Import

Tasks:

- 889 Map a Curriculum
- 888 Migrate to OO Design

Acceptance Criteria:

- Curriculum requirements XML file can be imported successfully.
- Different Program requirements are proven to import as expected. (Note that the xml data needs to be entered manually. Choose three below, most complicated ones. We need to get a few examples of GPA Audit reports to test properly.
    - Biology Major
    - Nursing
    - Business
    - Mechanical Engineering
    - Law
    - Computer Engineering
- XML file interface requirements is properly documented as well as data flow.

Modeling:
Please refer to Figure A-8 in Appendix A for the sequence diagram that was created. Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #819 Generate GPA Forecast Report

Tasks:

- 871 Tested GPA Forecast Report
- 870 Develop GPA Forecast Report
- 869 Pseudo Algorithm for GPA Report

Acceptance Criteria:

- Generate GPA Forecast Report from the GPA Dashboard.
- A new HTML page will be open with a printed report.

- Student should see a table of classes, weights, relevance, minimum grade requirements, secure grade requirements, and estimated study time recommendations for all remaining courses in the program.
- Include GPA History graph.

Modeling:
Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #867 DEFECT: Fix GPA Dashboard Graph
Tasks:
- Fix data points algorithm
- Fix look of the dashboard

Acceptance Criteria:
- Data points begin from the first semester completed.
- Data points do not extend past the previous completed semester.
- Data points are calculated accurately.
- Graph is aesthetically pleasing.

Modeling:
Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #887 Migrate Semester Dashboard Code to OO
Tasks:
- Finish migrating Dashboard
- Update UMLs
- Test new OO functions

Acceptance Criteria:
- More function in SemesterDashboardController

Modeling:
Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #847 Migrate Error Log Feature to Object Oriented Design
Tasks:
- Create initialization file

Acceptance Criteria:
- Fix issues with consumer not starting.
- Worker thread should start the consumer.

Modeling:
Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #891 Fix Import Curriculum

Tasks:

- Create new XML format for curriculums
- Testing using Nursing Curriculum

Acceptance Criteria:

- Courses in XML file should be saved
- Buckets in XML file should be saved
- Major name should be saved
- Major min GPA should be saved

Modeling:

Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #866 DEFECT Fix JSON Errors on Sem Dash

Tasks:

- Update database calls to OO

Acceptance Criteria:

- As a student I would like to be able to add assessments and grades for courses that are in progress.
- DEFECT from Fall 2015

Modeling:

Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #820 Understand GPA Dashboard Data

Tasks:

- 890 Migrate to OO design
- 835 UML Documentation for Data tables
- 834 Flowchart for Data tables
- 833 Pseudo Algorithm for Data tables

Acceptance Criteria:

- Map of how the courses taken and required are loaded into the GPA Dashboard data tables (solution / data flow)
- Correct GPA is displayed to the user.
- Migrate existing code to OO design.

Modeling:

Please refer to Figure A-4 in Appendix A for the sequence diagram that was created. Please refer to Figure A-12 in Appendix A for the use case diagram.

Sprint 6

(04/04/2016 - 04/15/2016)

User Story #812 Improve Semester Dashboard Look-And-Feel

Tasks:

- 904 Implement Angular Data Tables and Graph
- 903 Tutorials on AngularJS
- 902 Testing Semester Dashboard
- 901 Create UMLs
- 900 Research and Install Angular Data Tables/Charts

Acceptance Criteria:

- Improve layout of the Semester Dashboard
- Use Angular Material as the framework
- Implement new Angular Data Tables
- Implement new Angular Chart

Modeling:

Please refer to Figure A-12 in Appendix A for the use case diagram. Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #895 Update Student's Grades

Tasks:

- 898 Implement Material Design
- 897 Create Mock-ups
- 896 Learn AngularJS

Acceptance Criteria:

- Use angular material design.
- Display information on material design data table.
- Save information to repository.

Modeling:

Please refer to Figure A-10 in Appendix A for the sequence diagram that was created. Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #899 Update Grad Programs

Tasks:

- Research angular material data tables
- Research angular controllers and services

Acceptance Criteria:

- Be able to update the program name and required GPA.
- Information will be displayed on material data table.

- Information will be saved in repository.

Modeling:
Please refer to Figure A-12 in Appendix A for the use case diagram.

Sprint 7

(04/18/2016 - 04/29/2016)

User Story #893 Improve Breakdown Look-and-Feel

Tasks:

- 908 Update Look of Breakdown (CSS)
- 907 Testing Assessment Breakdown
- 906 Migrate Front-End to AngularJS
- 905 Breakdown UMLS

Acceptance Criteria:

- Use AngularJS framework
- Implement Angular data tables
- Implement Angular charts
- Implement Angular tabs

Modeling:
Please refer to Figure A-9 in Appendix A for the sequence diagram that was created. Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #886 Project Future Data for Semester Forecast Reports

Tasks:

- 912 Research Projection WMA
- 911 Implement Projection
- 910 Semester Projection Algorithm

Acceptance Criteria:

- A graph at the bottom of the Semester Forecast Report.
- Graph will show the estimated final grade of the class based on grades provided.
- The “Moving Average” forecasting method will be applied to generate data.

Modeling:
Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #885 Project Future Data for GPA Forecast Reports

Tasks:

- 914 Implement GPA Projection
- 913 GPA Projection Algorithm

Acceptance Criteria:

- A line graph will display what the remaining semesters may average out to be.
- The “Moving Average” forecast method will be implemented to generate the data values.

Modeling:
Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #874 Migrate GPA Dashboard to AngularJS
Tasks:

- 917 Store GPA from Audit Report
- 916 Research Autocomplete

Acceptance Criteria:

- Fixes:
    - GPA AVG needs to match GPA Audit report.
    - If a bucket requirement has been completely fulfilled, then it should not be displayed in the Required data table.
    - Correct any other obvious bugs/errors.
- Improvements:
    - Data table headers should be fixed (frozen) as you scroll down.
    - Data tables need to look more native, they look kind of alien. Scroll bar seems to not belong, it looks like the frame is the wrong one for the data table.
    - Feature to expand all buckets/sub buckets.
    - Feature to collapse all buckets.
    - When a bucket is expanded then a minus (–) sign should be shown instead of (+)
    - Target Graduate program should be a searchable field and also if possible drop down list.
    - Fully test that the Audi Import is properly reflected on the data tables.
    - Improve Modify weight and relevance. Ideally it would be nice if we could provide a horizontal slide bar to change the weight and relevance values: i.e.: weight: ----|- relevance --|---
    - If a graduate program is not selected then the student should be able to enter a desired target GPA manually. Ideally this can be controlled by a horizontal slide bar with possible values from 2 to 4, as well as a manual entry with allowed possible values from 2.0 to 4.0 (Let’s check if 2.0 is the minimum GPA allowed by FIU)

Modeling:
Please refer to Figure A-12 in Appendix A for the use case diagram.

User Story #909 Update Weights and Relevance
Tasks:

- 915 Research Input Sliders

Acceptance Criteria:

- Use angular material design.
- Weight and relevance should be set with a slider bar.
- Sliders should show when the course is selected on data table.

Modeling:

Please refer to Figure A-11 in Appendix A for the sequence diagram that was created. Please refer to Figure A-12 in Appendix A for the use case diagram.

SYSTEM DESIGN

This section provides high level details of the design of the system. First, we discuss the architecture patterns used for the system, followed by the decomposition of the systems into smaller subsystems. The deployment diagram will discuss which subsystems will reside on each hardware component and show how the different pieces are connected. This section ends by describing the design patterns of the system.

Architectural Patterns

For this project we used the Model-View-Controller (MVC) architecture. We chose this model because AngularJS is based on MVC. This makes it a natural choice to use when developing web applications.

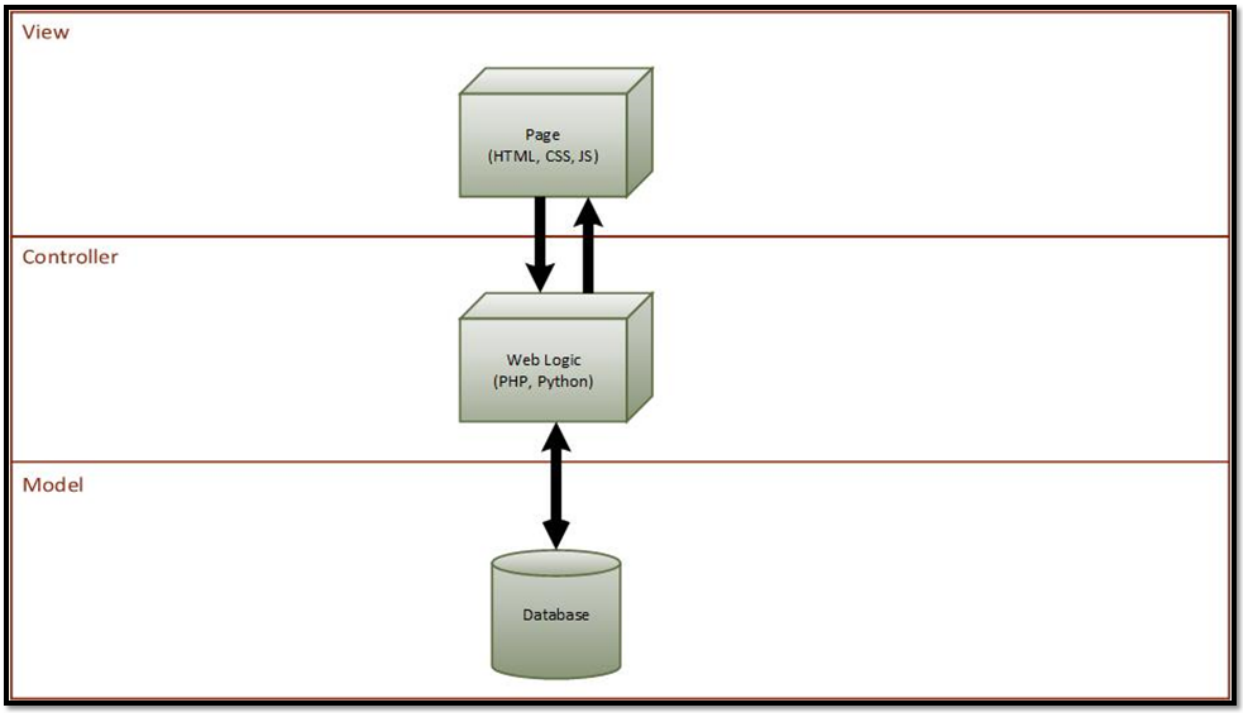


Figure 1.1 Model-View-Controller Architecture

System and Subsystem Decomposition

In the figure below, the main subsystems are the client-side applications, backend web services, and the data subsystem. The data subsystem allows the synchronization of the two client-side applications.

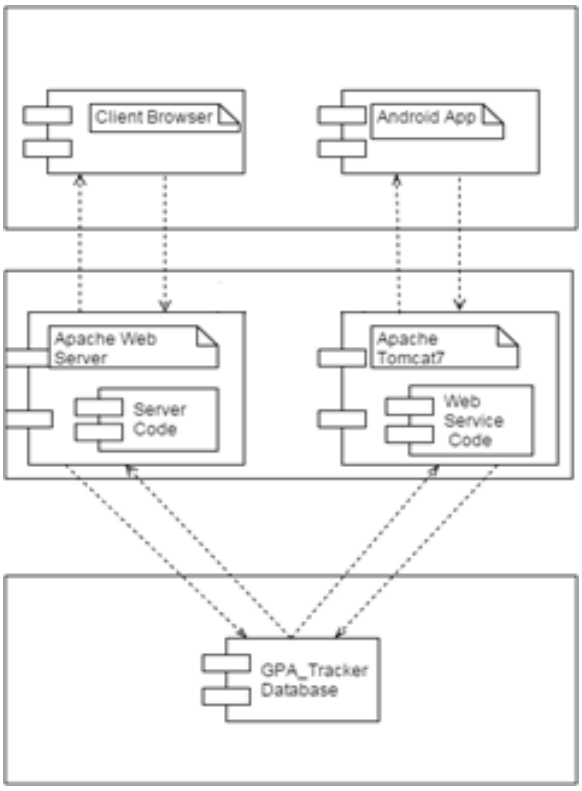


Figure 1.2 Subsystem Decomposition Diagram

Deployment Diagram

This project uses a LAMP stack on the server-side. The server-client communication occurs through HTTP. On the server machine the web server subsystem can communicate with the data subsystem using TCP/IP.

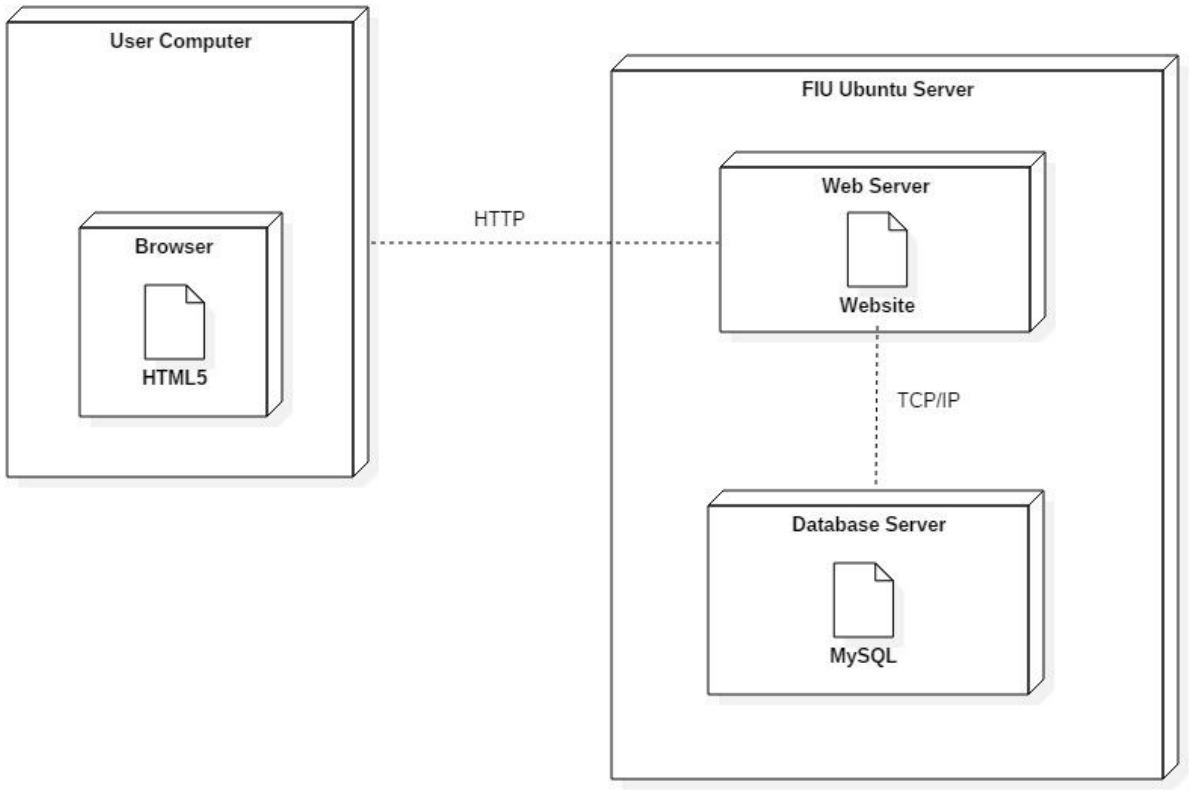


Figure 1.3 Deployment Diagram

Design Patterns

This site uses server-client pattern to frequently communicate asynchronously. The server can receive connections from different clients and create multiple processes to handle them.

SYSTEM VALIDATION

Systems testing and unit testing was conducted on the user stories below. For systems testing, we created sunny day and rainy day scenarios. Unit tests were done to make sure that as new features are implemented and older features are modified, the values of the functions will remain the same. Extreme values were used to make sure a variety of scenarios were covered.

User Story #809 Formatted Semester Forecast Report
System Tests

- Test001 (Sunny Day) - A Semester Forecast Report is generated (including a data and graph) when a Student meets all the requirements (registered student, imported courses, attainable GPA goal, input assessment grades).
- Test002 (Rainy Day) - A Semester Forecast Report should not generate if the Student does not meet at least one of the requirements.

User Story #811 Troubleshoot GPA Audit Import
System Tests

- Test003 (Sunny Day) - Student is able to import their PantherSoft GPA Audit when selecting the correct PDF file.
- Test004 (Rainy Day) - Student is not able to import an incorrect PDF file in the GPA Audit Import.

User Story #812 Improve Semester Dashboard Look-And-Feel
System Tests

- Test005 (Sunny Day) - Student is able to see a data table and graph on the Semester Dashboard when they are registered, imported their courses, and added grades for their courses in-progress.
- Test006 (Rainy Day) - Student is not able to import an incorrect PDF file in the GPA Audit Import.

Unit Tests

- GetGraphData001 - Student's credentials are passed to return the graph data related to their courses.
    - Pass.
- CourseLegend001 - Student's credentials are passed to return the courses for the graph's legend
    - Pass.

User Story #819 Generate GPA Forecast Report
System Tests

- Test007 (Sunny Day) - Students are able to generate a GPA Forecast Report in the GPA Dashboard if they are a registered user with an imported PantherSoft GPA Audit and have selected the weight and relevance values for all in-progress and remaining courses.

- Test008 (Rainy Day) - Students cannot generate a GPA Forecast Report in the GPA Dashboard if they have not imported their PantherSoft GPA Audit.

Unit Tests
- CheckWeightAndRelevance001 - Return all weights and relevance values for courses that are in-progress or incomplete for the Student.
    - Pass.
- GetGraphData002 - GPA is generate for each semester completed at FIU for the Student who import their PantherSoft GPA Audit.
    - Pass.
- GetGraphData003 - No GPA data is returned for a new user who has not imported a PantherSoft GPA Audit.
    - Pass.

User Story #820 Understand GPA Dashboard Data
System Tests
- Test009 (Sunny Day) - Tables and graph are displayed to the Student when all requirements are met.
- Test010 (Rainy Day) - No tables or graphs are displayed for a new Student who has not met all the requirements.

Unit Tests
- GetBuckets001 - Returns list of buckets and courses for the Student.
    - Pass.
- GetProgramInfo001 - Returns correct GPA information.
    - Pass.
- GetBuckets002 - Returns a null value for a new student who has not import a PantherSoft GPA Audit.
    - Pass.
- GetProgramInfo002 - Returns a null value for a new student who has not import a PantherSoft GPA Audit.
    - Pass.

User Story #823 (Teamwork) Create Error Logging Interface
System Tests
- Test011 (Sunny Day) - The error log gets written to when a GPA Audit file is imported into the system.
- Test012 (Rainy Day) - The error log does not get written to when an incorrect file is attempted to be loaded in the “import” for the GPA Audit.

User Story #826 Instantiate Curriculum Requirements
System Tests
- Test013 (Sunny Day) - Student sees the classes that he still has to take on the GPA Dashboard when a GPA Audit is imported.
- Test014 (Rainy Day) - Student will not see the classes that he still has to take on the GPA Dashboard when an incorrect file is imported.

Unit Test

- ImportAudit001 - Ensure the correct courses are instantiated when a student does an import for the first time. List of courses student has taken and needs to take is returned.
    - Pass.
- ImportAudit002 - Ensure the correct courses are updated when a student does an import. When a list of updated courses is passed, an updated list of courses student has taken and needs to take is returned.
    - Pass.

User Story #827 Create Error Log System

System Tests

- Test015 (Sunny Day) - The error log interface will be accessed and printed to when a GPA Audit PDF is imported.
- Test016 (Rainy Day) - Test if the error log will print when any parameters are passed. Log file will have a new entry with arbitrary input.

User Story #828 Add Semester Forecast Functionality

System Tests

- Test017 (Sunny Day) - Students are able to generate a semester forecast report (popup) when classes for been imported and relevance and weight values have already been set for the classes in-progress for the current semester.
- Test018 (Rainy Day) - Students cannot generate a report unless classes have already been imported into the system. When the student attempts to generate the semester forecast, student will be prompted with an error message warning them that "Classes Have NOT Been Selected for Forecast to be Generated."
- Test019 (Sunny Day) - Students are able to generate a semester forecast report when classes for been imported and not all weight and relevance values have been set for the classes in-progress for the current semester prior generating the report. When the student attempts to generate the semester forecast, student will be prompted to enter an appropriate value for weight and relevance for MAD3512. The Semester Forecast then will display with the new values.

User Story #846 DEFECT: Fix Data Points and Calculations

System Tests

- Test020 (Sunny Day) - Student is able to view the graph generated at the bottom of the semester forecast report when classes for been imported and relevance and weight values have already been set for the classes in-progress for the current semester. The Semester GPA Forecast Report will open in a new tab with the Graph for "Grade Trends for Current Courses" at the bottom of the report.
- Test021 (Rainy Day) - Student cannot view a graph in the Semester GPA Forecast Report unless classes have already been imported into the system.

Unit Test

- GPAGoal001 - When a student has input a GPA Goal, the function will retrieve the value from the database
    - Pass.
- GradesAndCredits001 - Test that all the grades for courses completed by the student are

returned along with the classes' credit hour.
    - Pass.

User Story #852 DEFECT: Fix Graph in Semester Dashboard
System Tests
- Test022 (Sunny Day) - Student is able to view the graph generated at the bottom of the semester forecast report when classes for been imported and relevance and weight values have already been set for the classes in-progress for the current semester.
- Test023 (Rainy Day) - Student cannot view a graph in the Semester GPA Forecast Report unless classes have already been imported into the system.

Unit Tests
- GetGraphData003 - To test all the data points to be plotted for the Grade Trends for Current Courses Graph are returned along with the dates associated with the point.
    - Pass.
- CourseLegend002 - To ensure that the current semester courses with Assessments values in the database are returned for the graph.
    - Pass.

User Story #857 Migrate Semester Forecast Report to Object Oriented Design
System Tests
- Test024 (Sunny Day) - Student is able to view the graph generated at the bottom of the semester forecast report when classes for been imported and relevance and weight values have already been set for the classes in-progress for the current semester. The Student has also added grades for assessments.
- Test025 (Rainy Day) - Student cannot view a graph in the Semester GPA Forecast Report unless classes have already been imported into the system.

Unit Tests
- TakenAndRemaining001 - To calculate and return the amount of credits taken by the student and the credits remaining in his/her field of study.
    - Pass.
- GradesAndCredits002 - To test that all the grades for courses completed by the student are returned along with the classes' credit hour.
    - Pass.
- CurrentCourses001 - To test that only the courses from the current semester taken by the student are returned.
    - Pass.

User Story #873 Map Admin Curriculum Import
System Tests
- Test026 (Sunny Day) - Ensure that the admin can import a curriculum. Nursing curriculum XML file is imported. Success message appears and curriculum will be stored in database.

- Test027 (Rainy Day) - If the admin selects an incorrect file to import, an error message appears.

Unit Tests
- ImportReq001 - Ensure that all the information from the XML file is imported. List of buckets and courses that were stored in the system are returned.
    - Pass.

User Story #893 Improve Breakdown Look-And-Feel
System Tests
- Test028 (Sunny Day) - Ensuring the new tabs, data tables and graph will display when a registered Student (that meets all requirements) reaches the Assessment Breakdown.
- Test029 (Rainy Day) - Ensuring the new tabs, data table will not display any courses and the graph will not display below when the Student's courses are not imported and assessments for courses have not been added respectively.

Unit Tests
- PlotPoints001 - To ensure that the graph will display when classes are imported and assessment grades have been entered.
    - Pass.
- GetGrades001 - To ensure that grade are returned for assessments according to the course.
    - Pass.

User Story #895 Update Student's Grades
System Tests
- Test030 (Sunny Day) - Ensure that the admin can change a grade. When a grade is changed, a success message appears.
- Test031 (Rainy Day) - When the admin selects a student with no courses, a blank data tables appears.

Unit Tests
- GetGrades001 - Ensure that all student grades are returned to be displayed. Returns a list of courses and their grades.
    - Pass.

User Story #909 Update Weights and Relevance
System Tests
- Test032 (Sunny Day) - Ensure that student can change weight and relevance. When values are changed, new values are displayed.
- Test033 (Rainy Day) - Ensure that student can back out and not save. When Student selects course and clicks cancel or clicks away from popup, displays the old values.

Unit Tests
- Save001 - Ensure that weight and relevance is being saved in repository. Values are saved with the new value when changed on the slider. Return "success".

- Pass.

GLOSSARY

Some key terms used throughout the document are defined below:

Bucket: A category of that needs to be fulfilled in major requirements. Examples include CS Science Requirements for Computer Science and Art for the UCC requirements.
Category: A term synonymous with Bucket. It's what determines the separation of courses and the requirements they fulfill.
FIU: Short for Florida International University.
Forecast Report: This is an overview of the student's progress that comes with a message to let the student know what grades are required to achieve the student's GPA goal. This is done on the GPA Dashboard for all remaining courses and on the Semester Dashboard for specific to the current semester.
GPA: Short for Grade Point Average. A calculated average of the letter grades earned in all classes based on credit hours in a scale from 0.0 to 4.0 (or 5.0 in other schools).
GPA Dashboard: The dashboard that takes all the courses the user has taken and remains and displays the progress of their GPA over time at FIU.
PantherSoft GPA Audit: The report provided by PantherSoft that has information about what grades a student has, what courses the student is taking, what courses remain, and what their major is.
Relevance: Refers to the student's opinion of the class's importance to the student personally. This value is used in the forecast reports.
Scrum: An agile software development framework for managing complex development projects.
Semester Dashboard: The dashboard that tracks the grades of a student throughout the semester.
Sprint: A time period (of two weeks in the project) where the development team is committed to completing a number of items.
Sprint Retrospective: A meeting that reflects on the last sprint, and agrees in improvements that can be made in the future.
User Story: A tool used in Agile software development that describes the feature to be implemented. The story's description includes the who, the what, and the why.
Weight: Refers to the student's opinion of the class difficulty level of a particular course. This value is used in the forecast reports.

APPENDIX

Appendix A - UML Diagrams

Static UML Diagrams

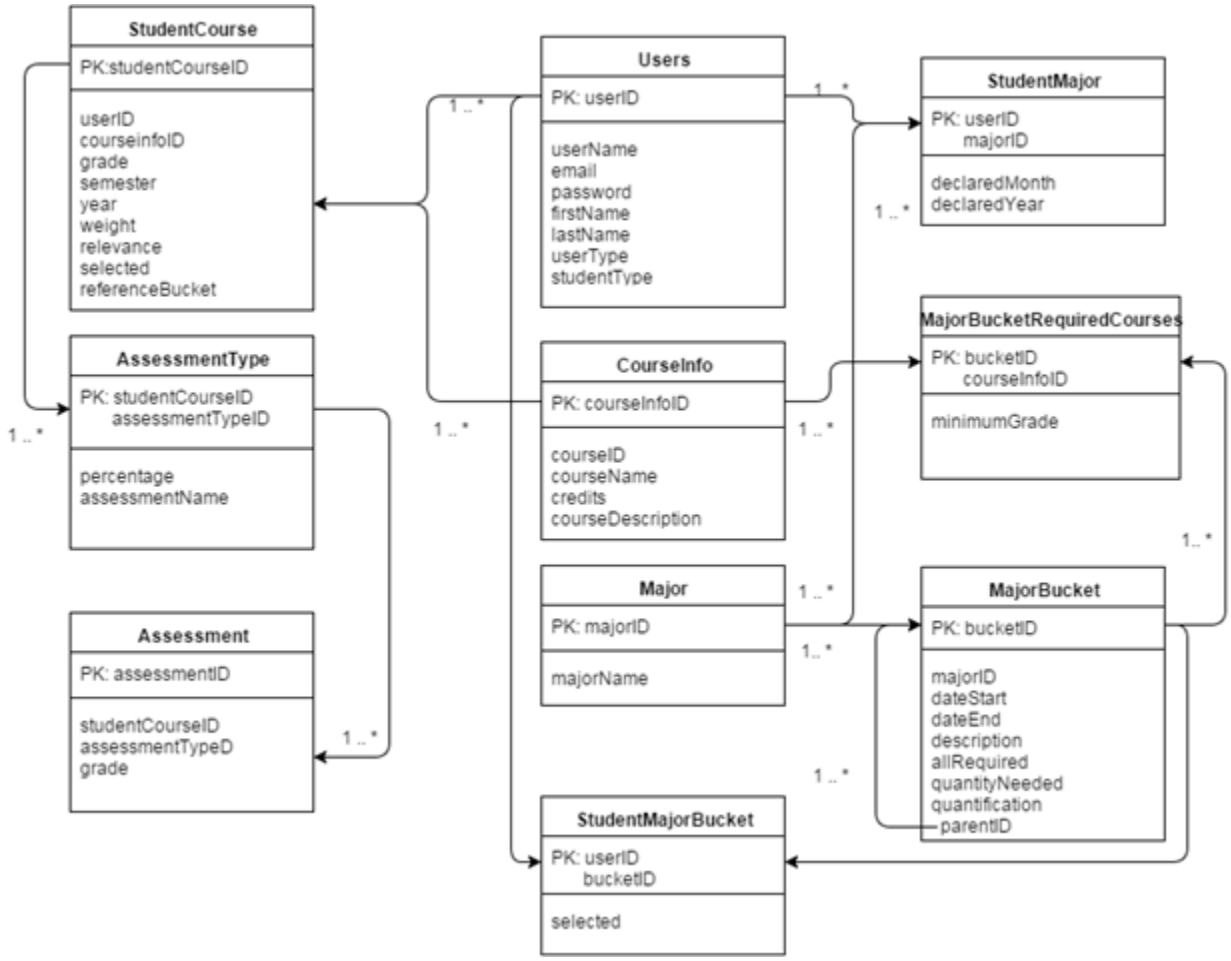


Figure A-1 - Persistent Data Design Diagram

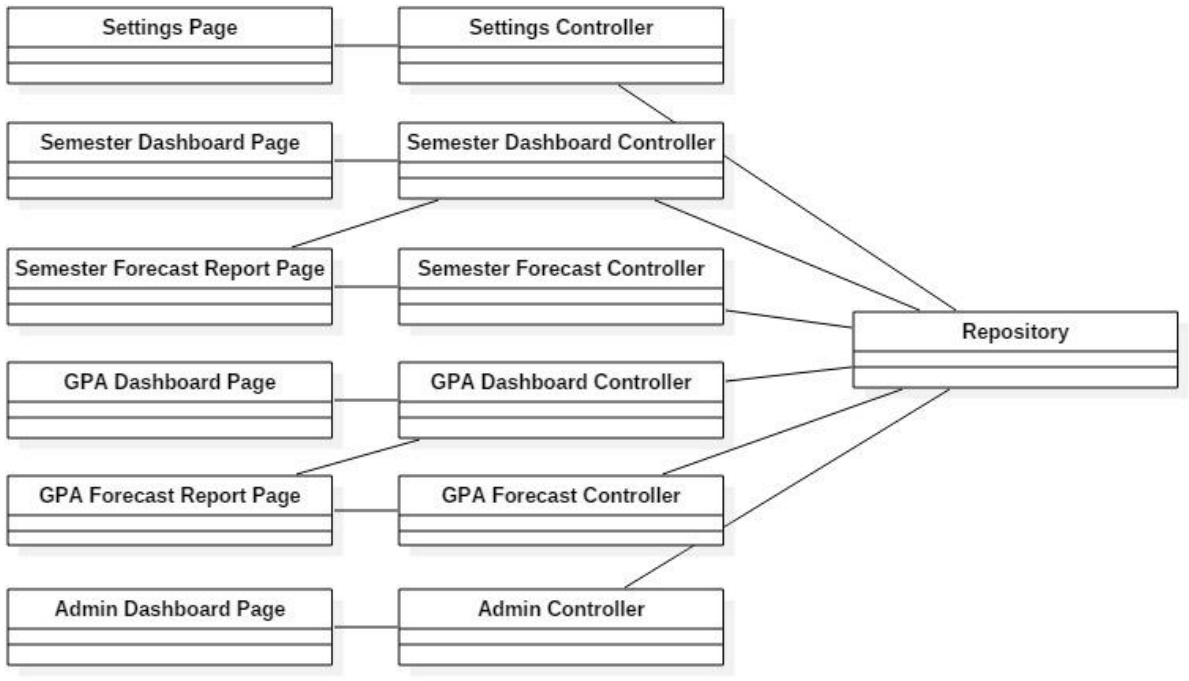


Figure A-2 - Minimal Class Diagram

Dynamic UML Diagrams

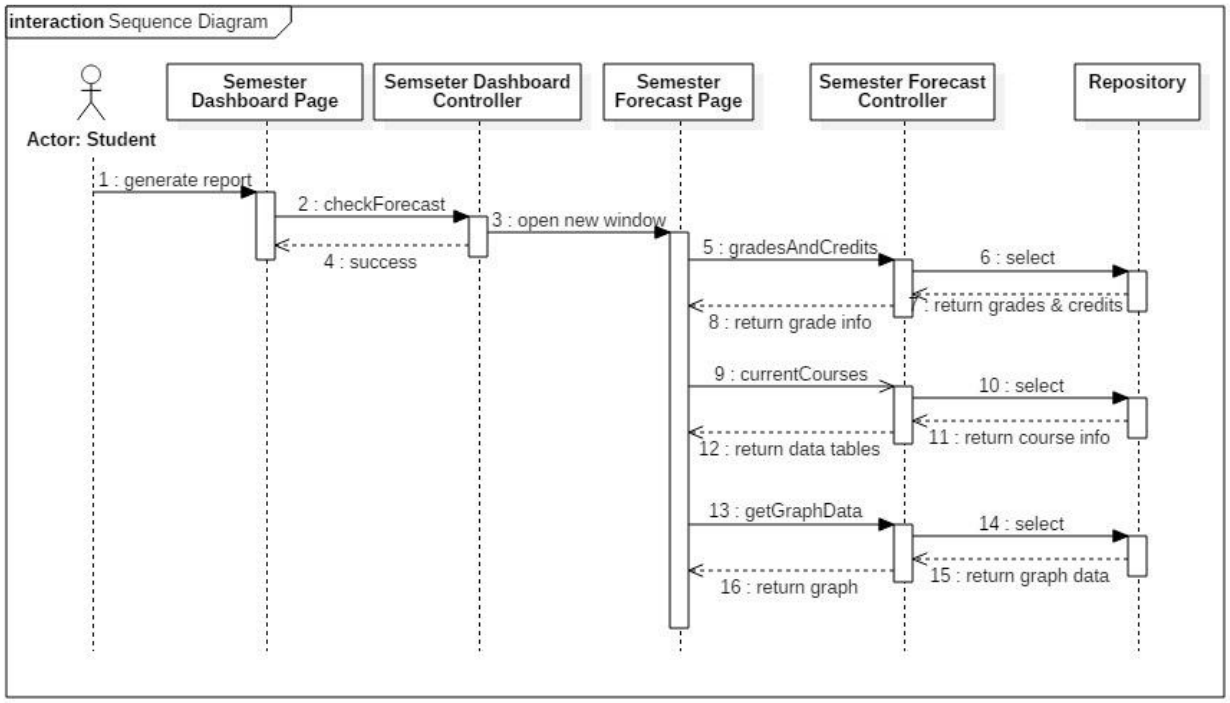


Figure A-3 - Generating Semester Forecast Report

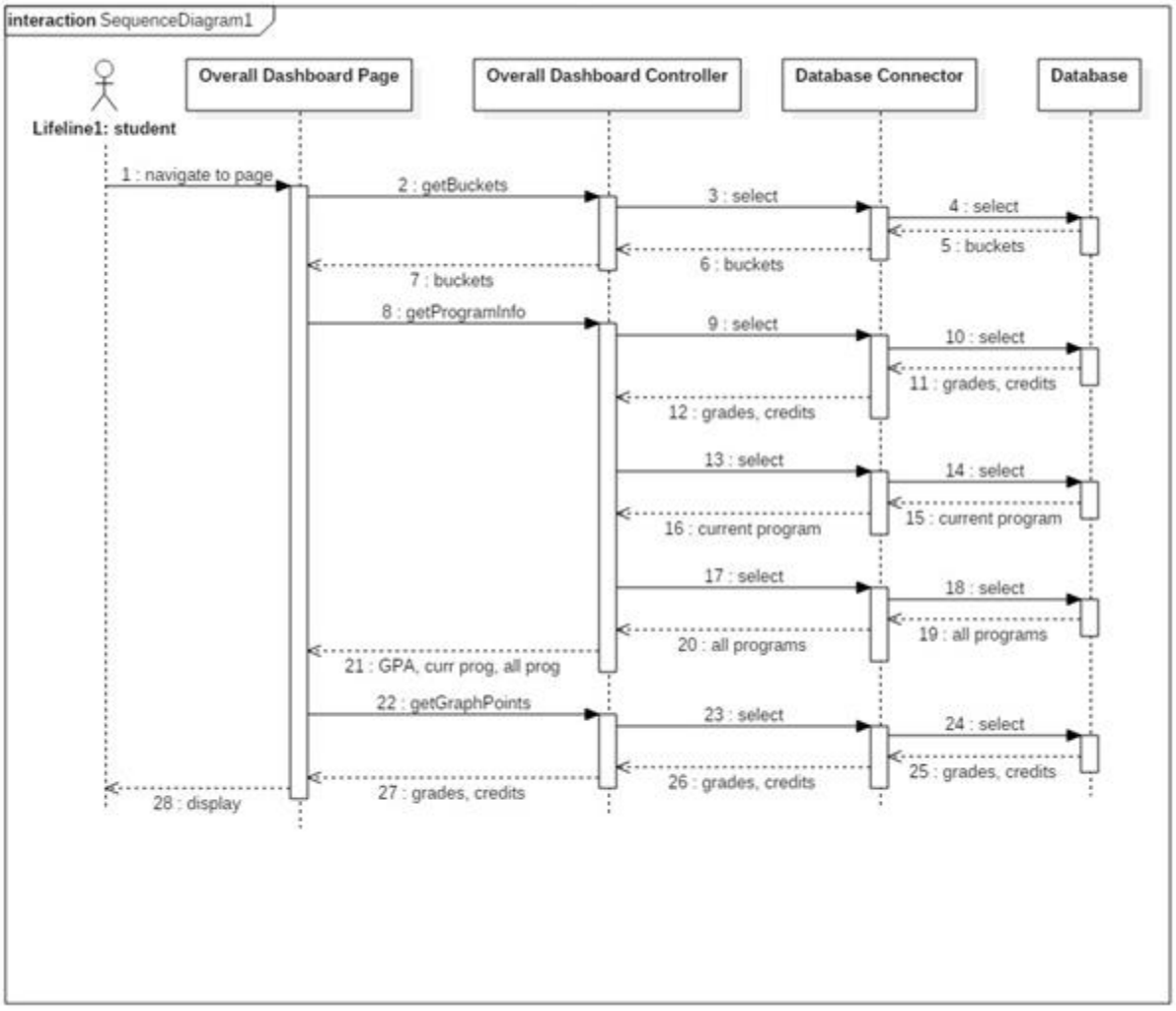


Figure A-4 - GPA Dashboard Information

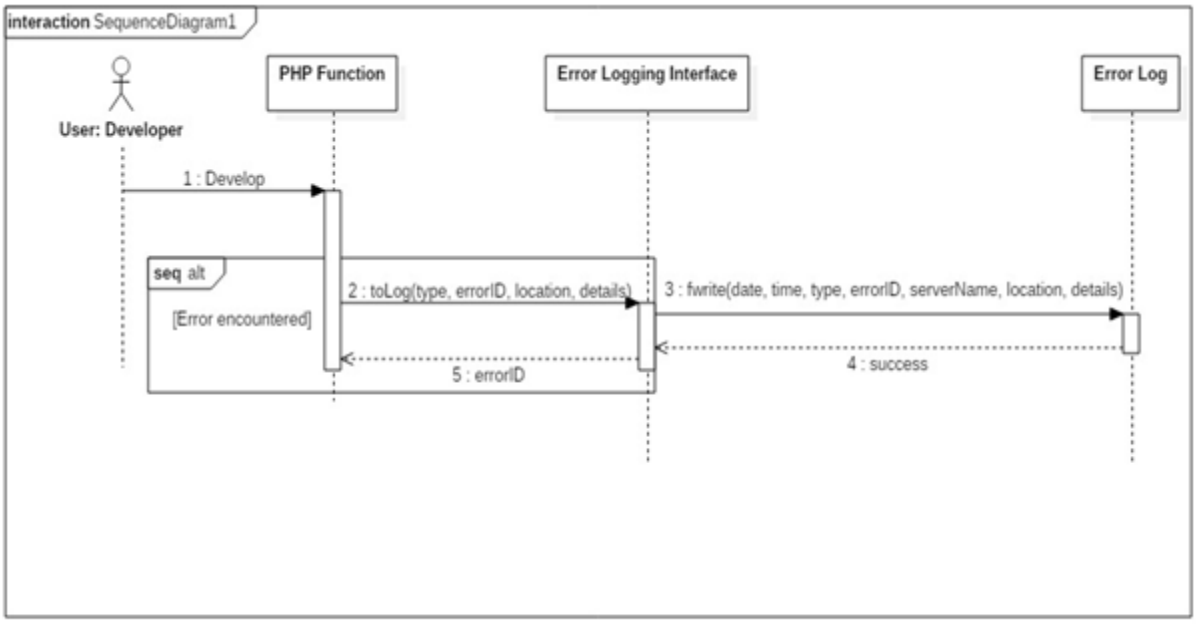


Figure A-5 - Writing to the Error Log

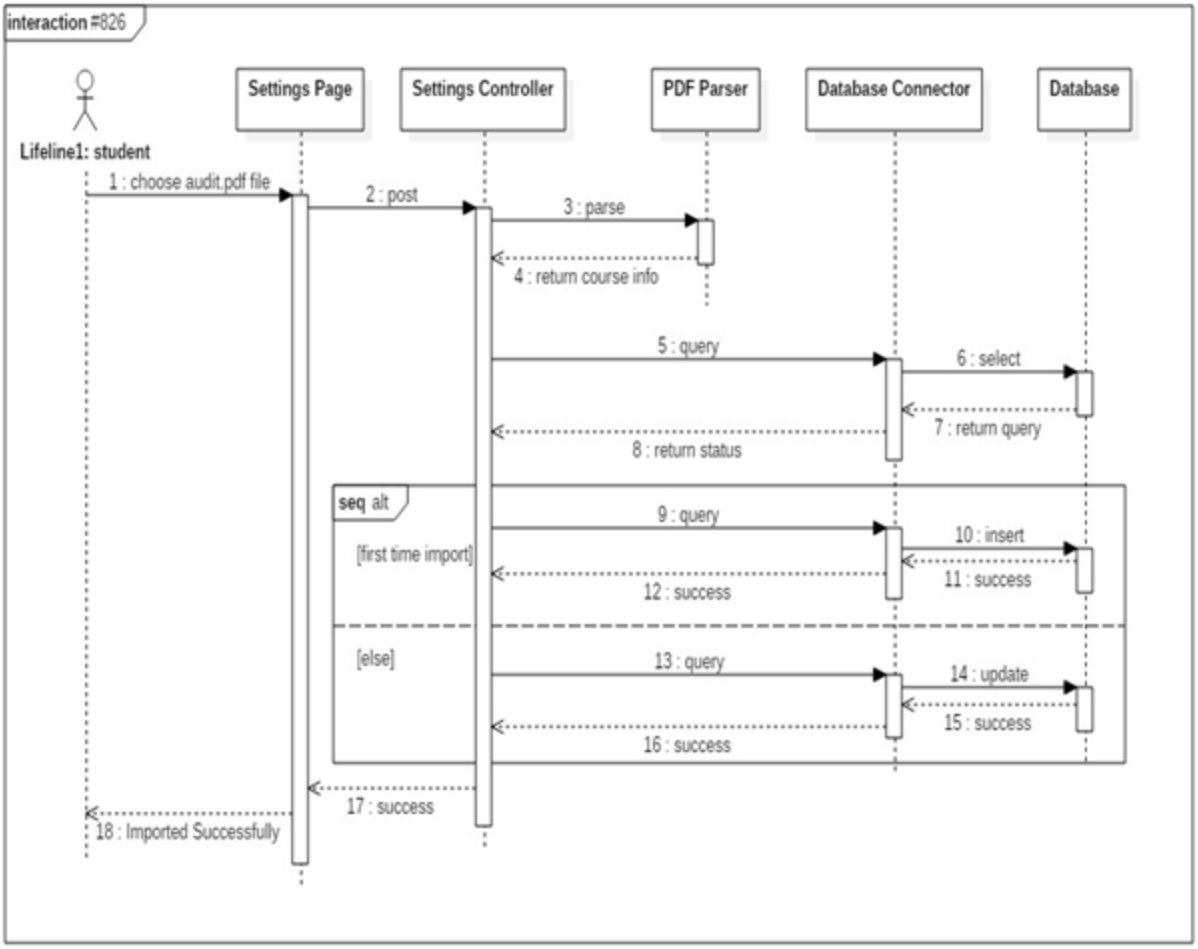


Figure A-6 - Import GPA Audit

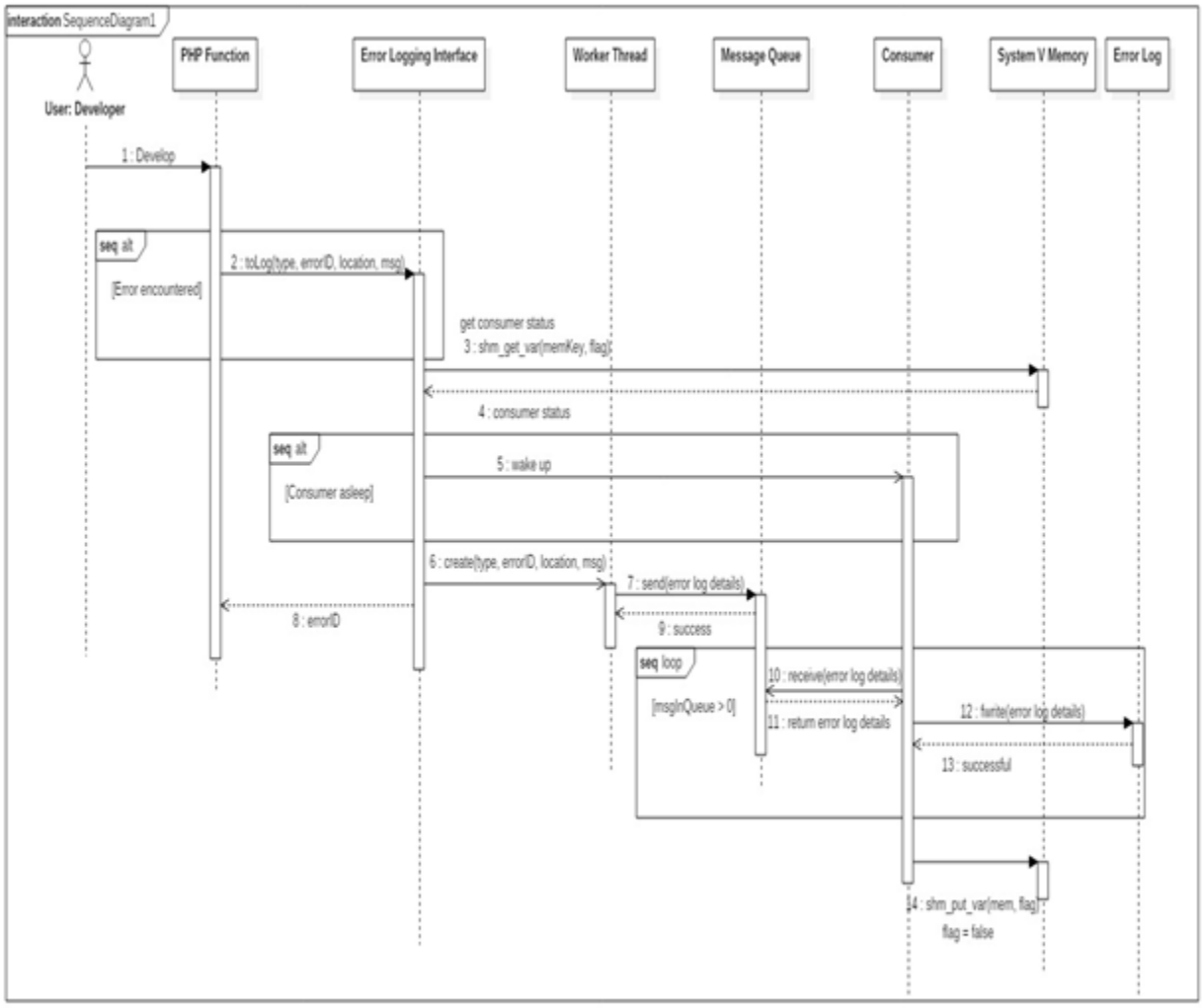


Figure A-7 - Concurrent Error Logging Interface

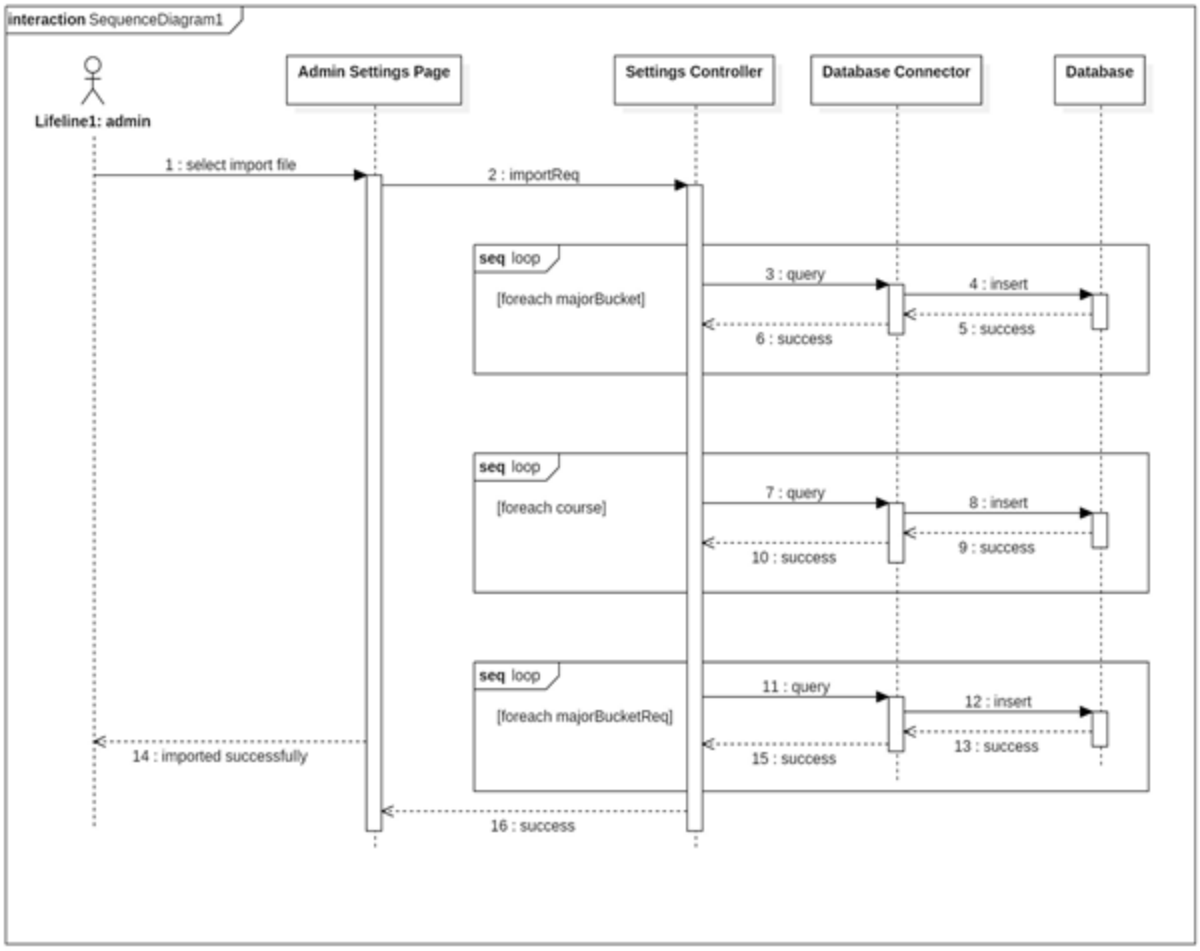


Figure A-8 - Admin Curriculum Import

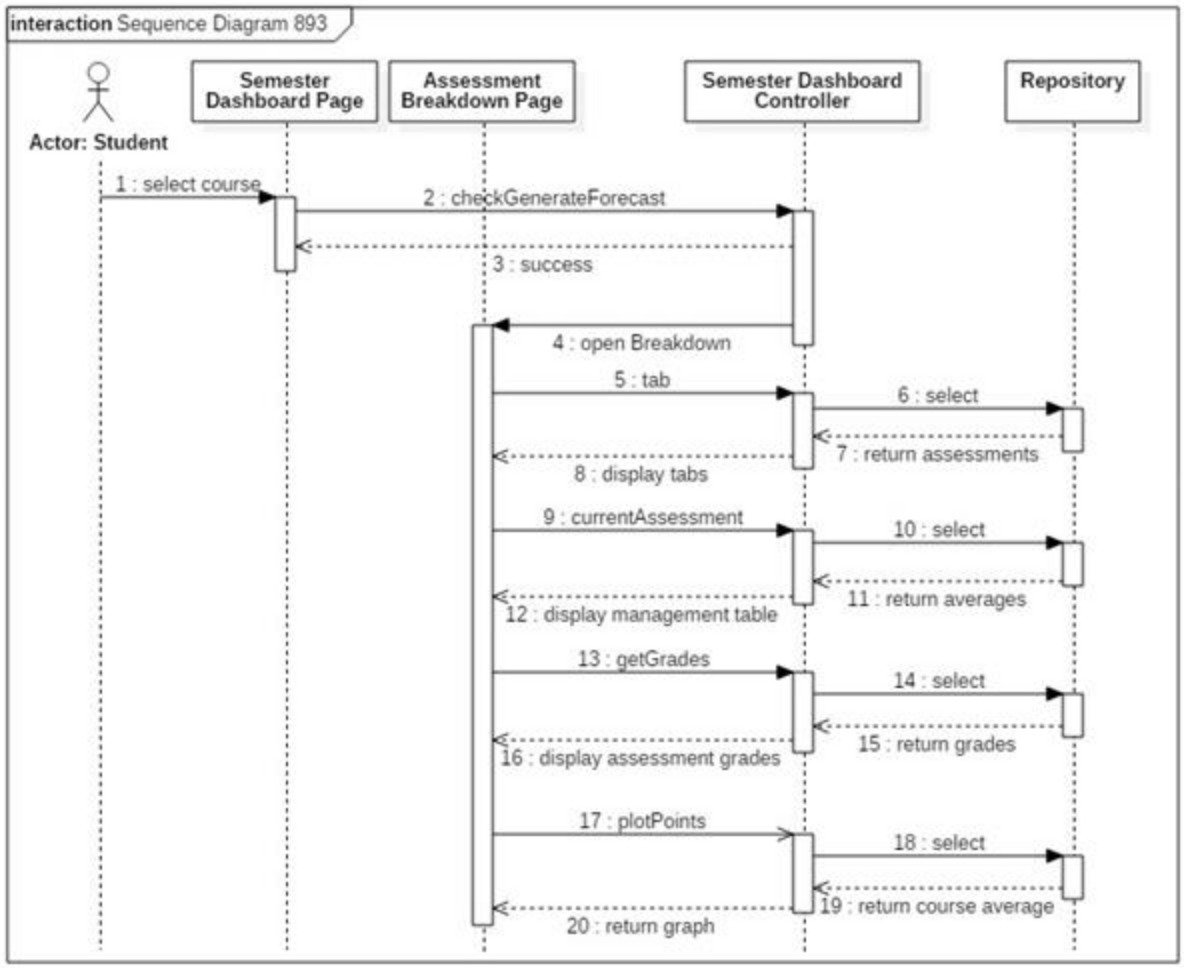


Figure A-9 - Assessment Breakdown Displayed

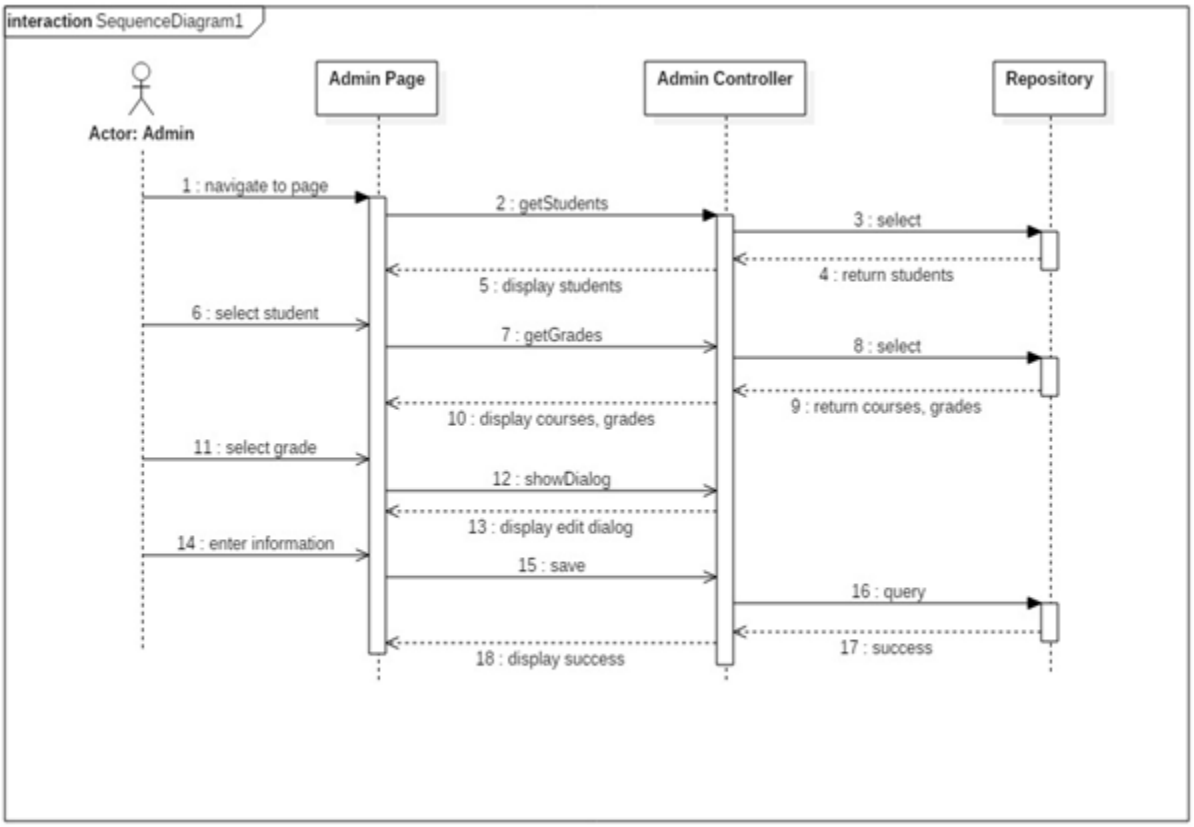


Figure A-10 - Admin Updating Student Grades

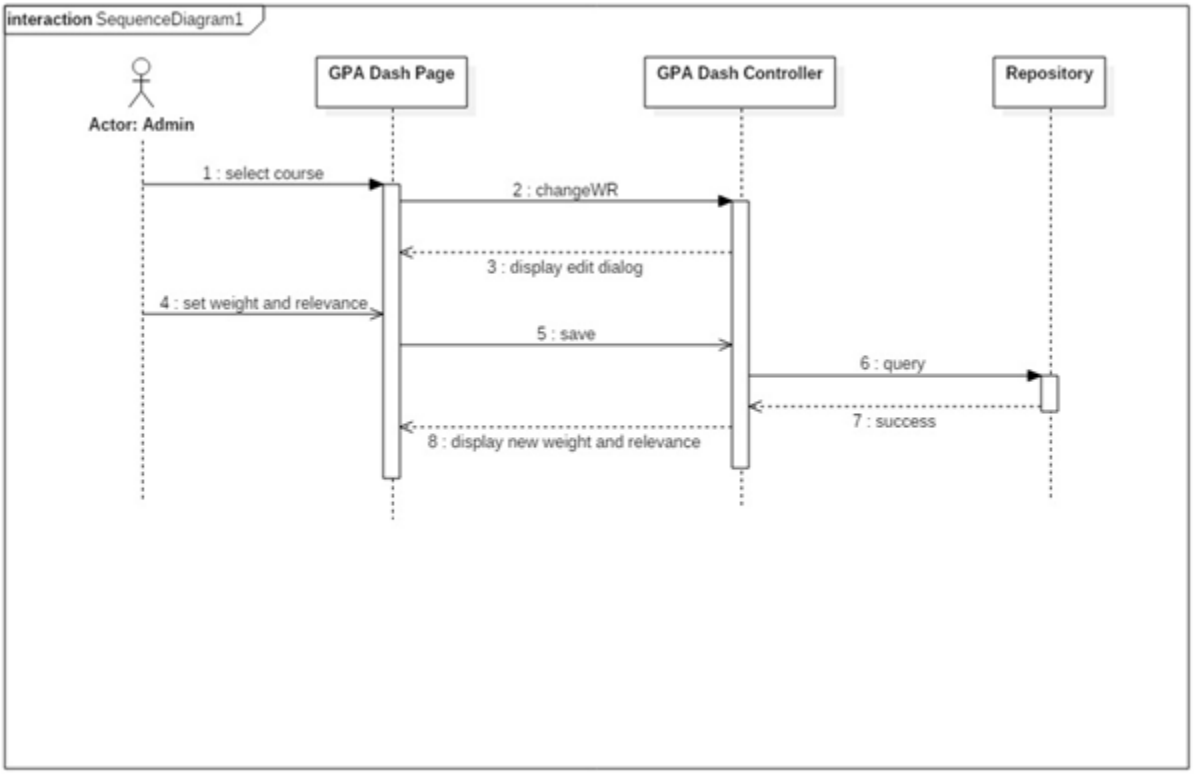


Figure A-11 - Updating Weight and Relevance

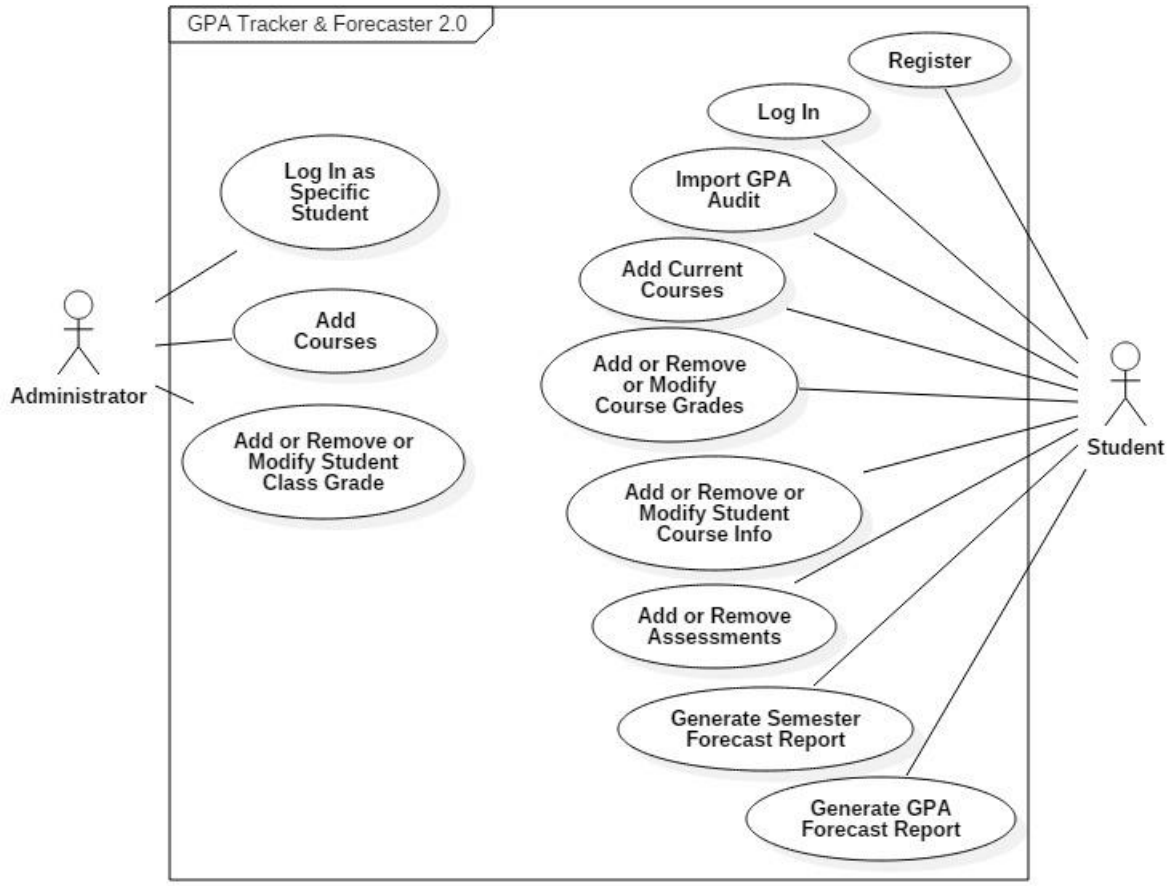


Figure A-12 Use Case Diagram

Appendix B - User Interface Design

Figure B-1 Home Page

Figure B-2 Registration

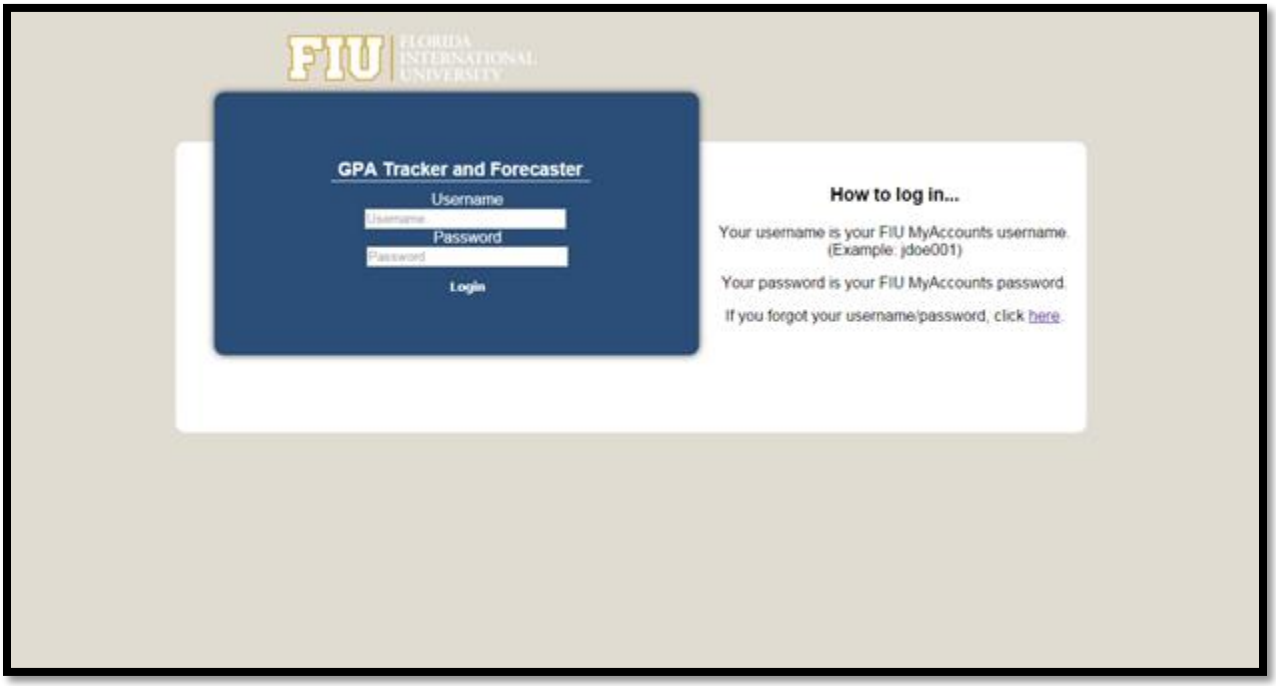


Figure B-3 Log In

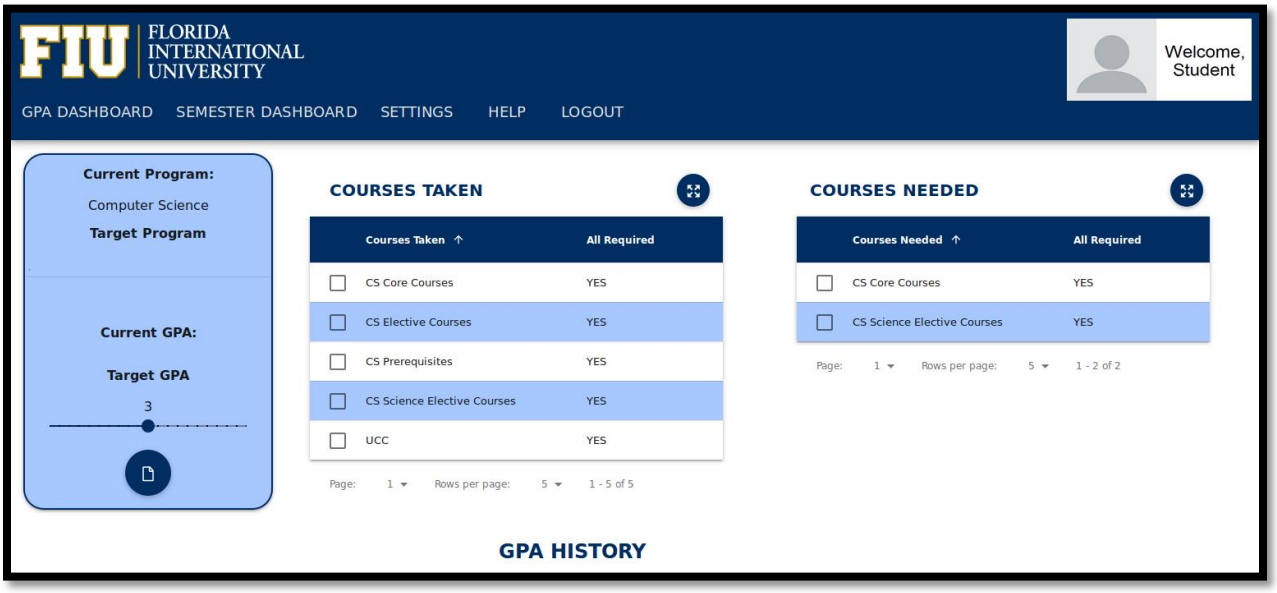


Figure B-4 GPA Dashboard Data Tables

Figure B-5 GPA Dashboard Graph

OVERALL GPA FORECAST REPORT

Current GPA: 2.96
Graduation Goal GPA: 3.00
Credits Remaining: 24

In order to your Graduation Goal GPA of 3.00, the following forecast has been generated according to the weight and relevance you provided for all the remaining courses:

Class	Weight	Relevance	Min. Grade Required	Secure GPA Path	Estimated Study Time(Hrs/Week)*
CIS4911	3	3	A-	A	9
COP4555	2	2	B+	A-	4
COP4610	2	2	B	B+	4
MAD3512	1	2	B	B+	2
COM3417	3	2	B	B+	6

**While only a recommendation, we highly recommend students to consider their circumstances and select an appropriate schedule based on their workload.*

GPA HISTORY

Figure B-6 GPA Forecast Report Data Table

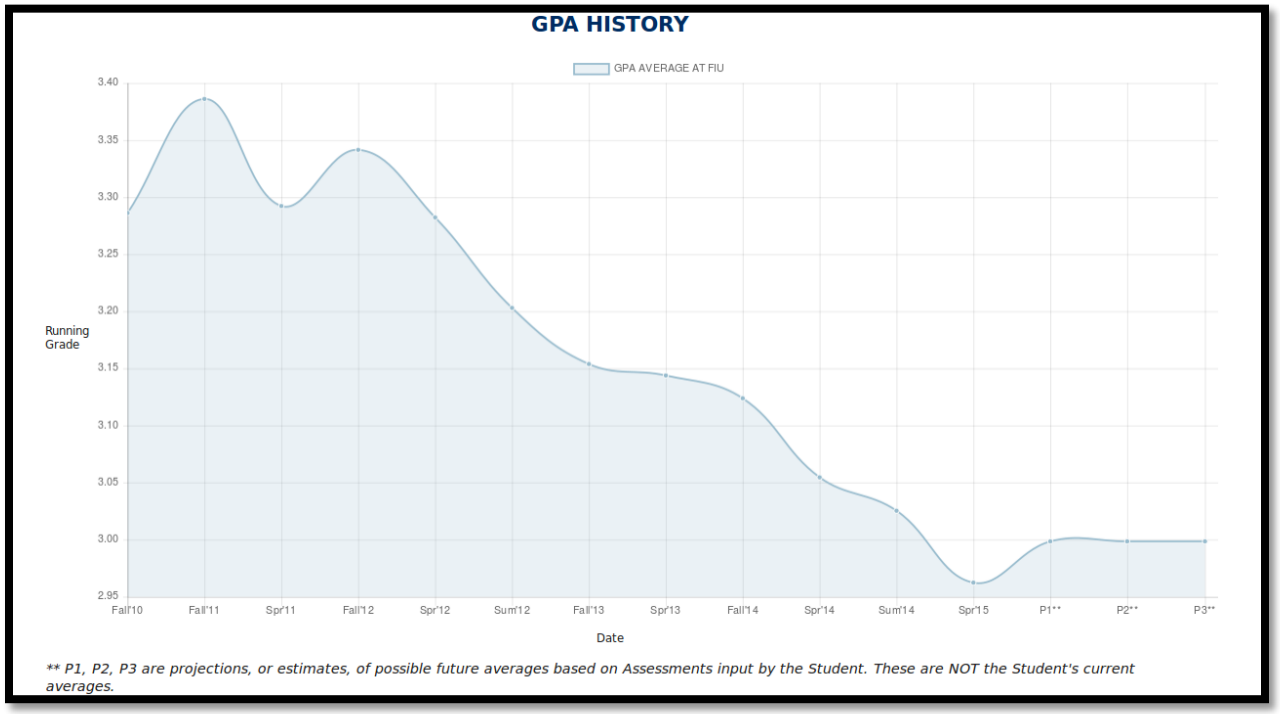


Figure B-7 GPA Forecast Report Graph + Projection

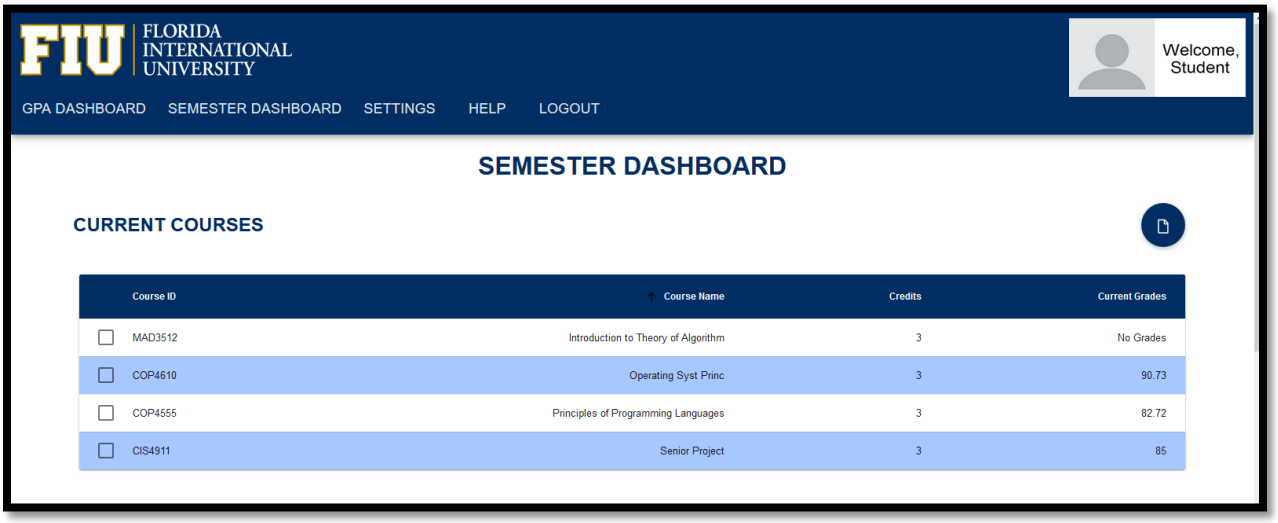


Figure B-8 Semester Dashboard Data Table

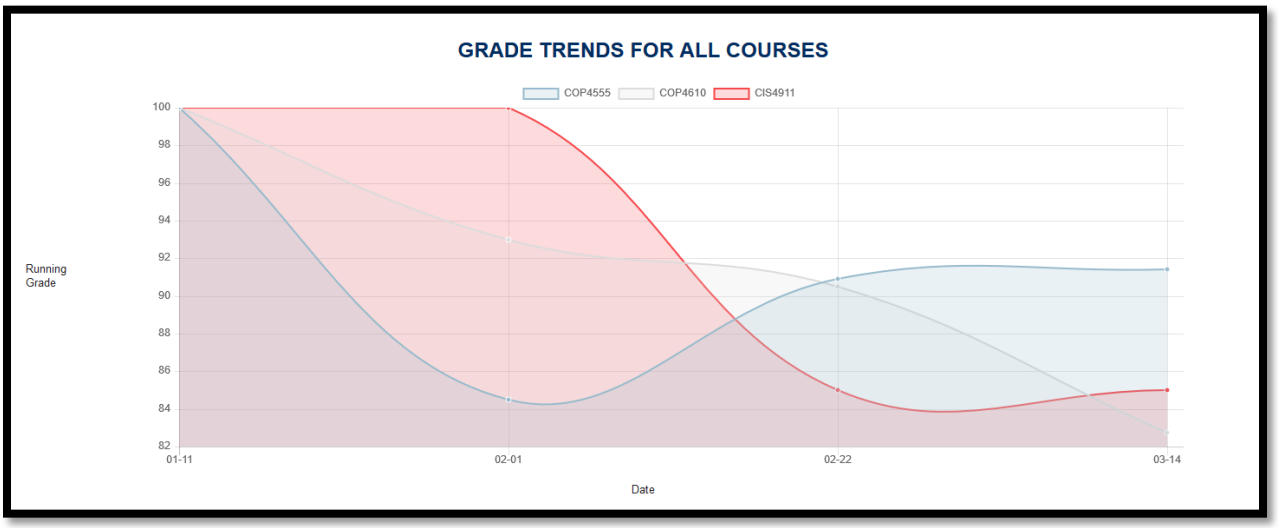
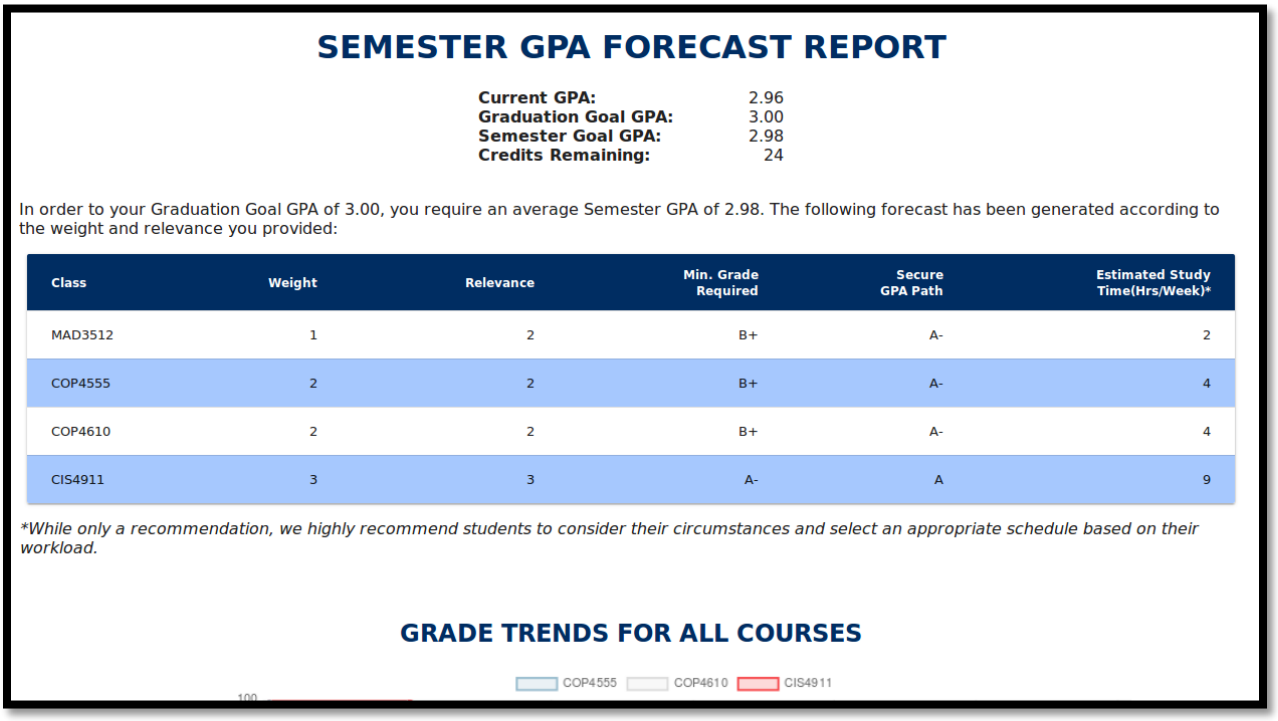


Figure B-9 Semester Forecast Graph



SEMESTER GPA FORECAST REPORT

Current GPA: 2.96
Graduation Goal GPA: 3.00
Semester Goal GPA: 2.98
Credits Remaining: 24

In order to your Graduation Goal GPA of 3.00, you require an average Semester GPA of 2.98. The following forecast has been generated according to the weight and relevance you provided:

Class	Weight	Relevance	Min. Grade Required	Secure GPA Path	Estimated Study Time(Hrs/Week)*
MAD3512	1	2	B+	A-	2
COP4555	2	2	B+	A-	4
COP4610	2	2	B+	A-	4
CIS4911	3	3	A-	A	9

**While only a recommendation, we highly recommend students to consider their circumstances and select an appropriate schedule based on their workload.*

GRADE TRENDS FOR ALL COURSES

COP4555 COP4610 CIS4911

100

Figure B-10 Semester Forecast Report Data Table

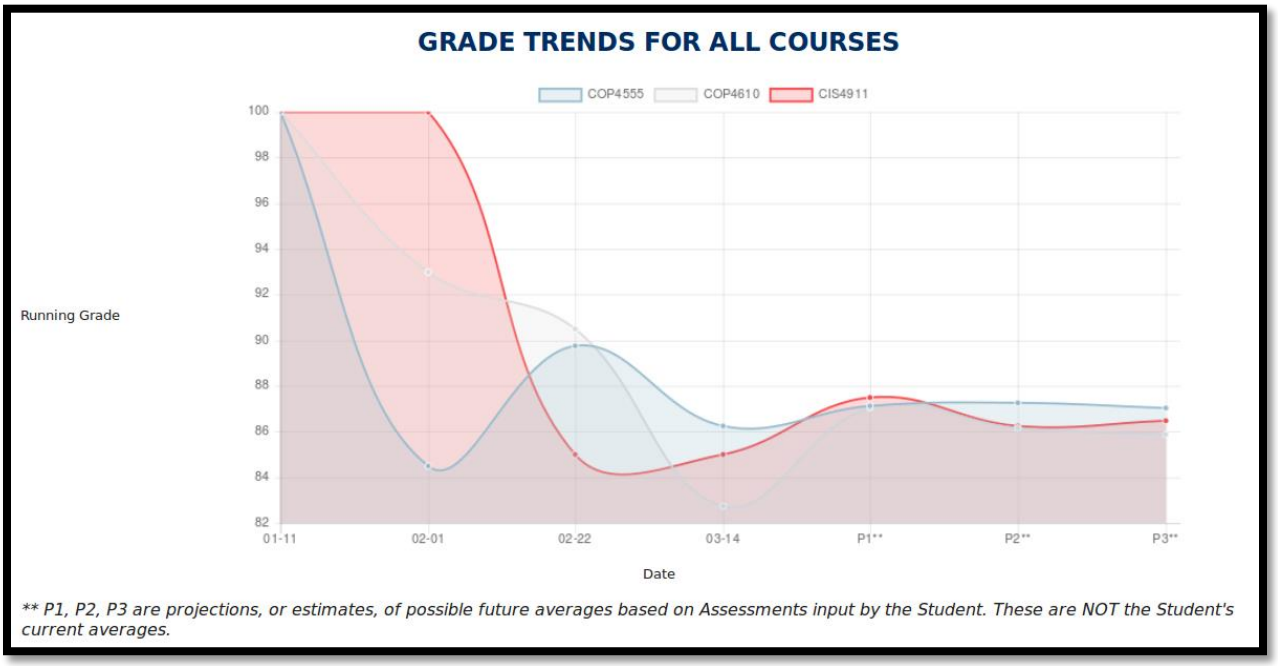


Figure B-11 Semester Forecast Report Graph

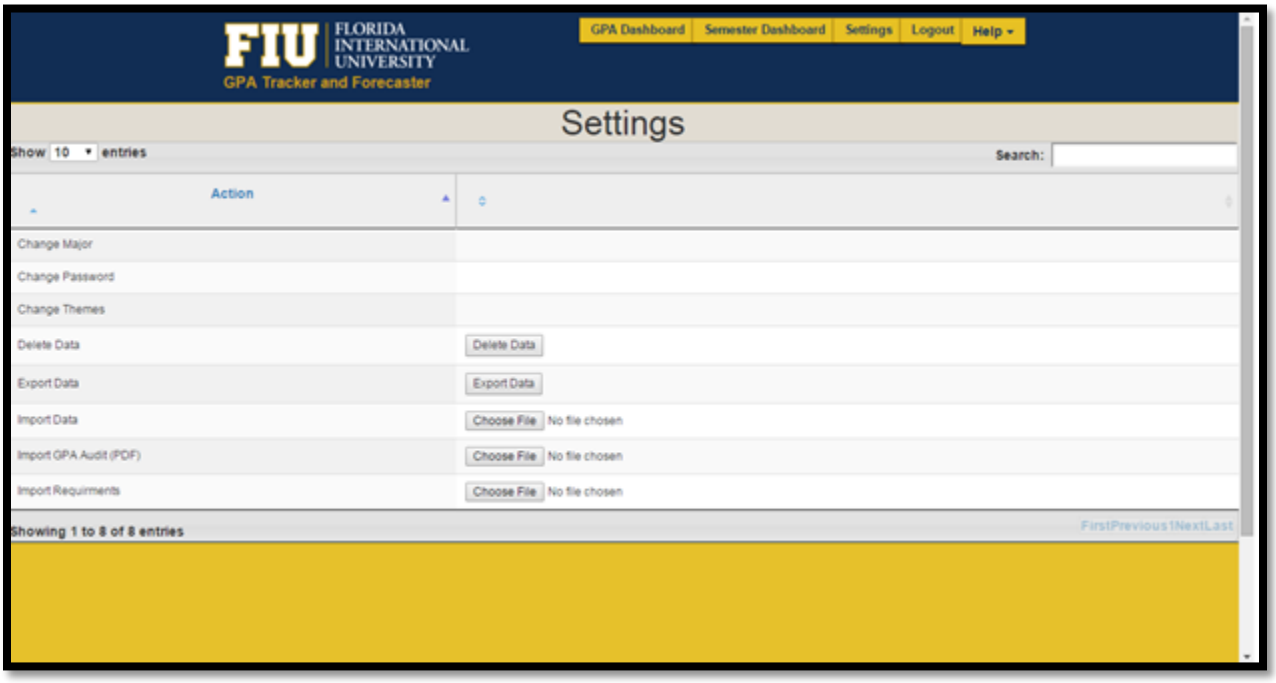


Figure B-12 Settings

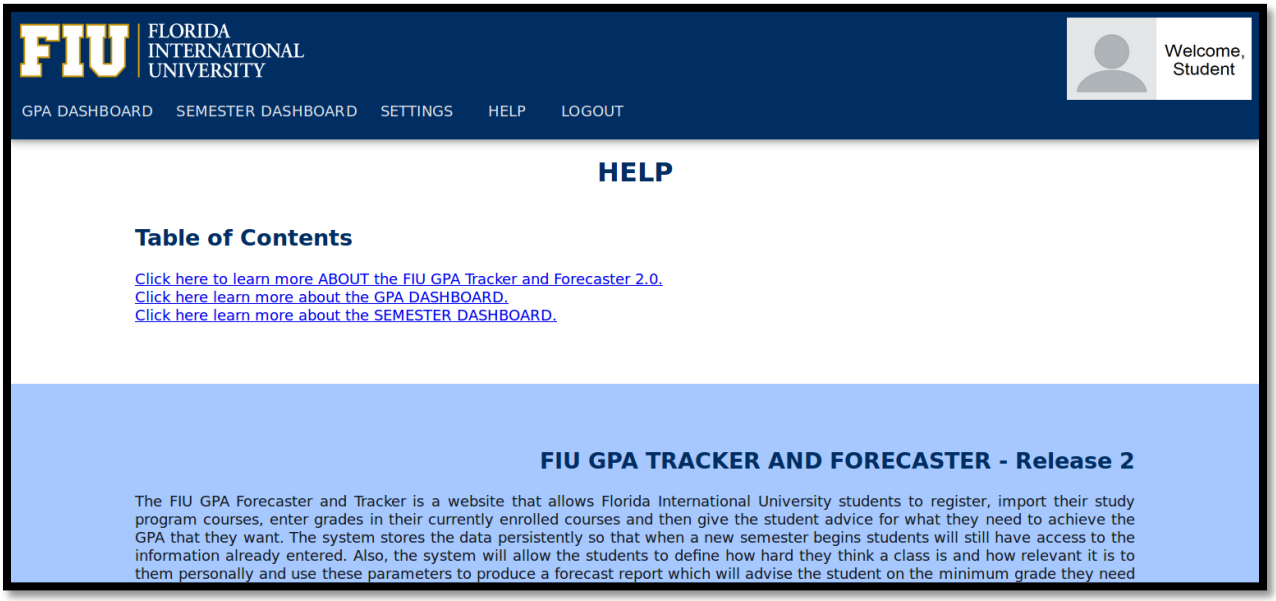


Figure B-13 Help / About

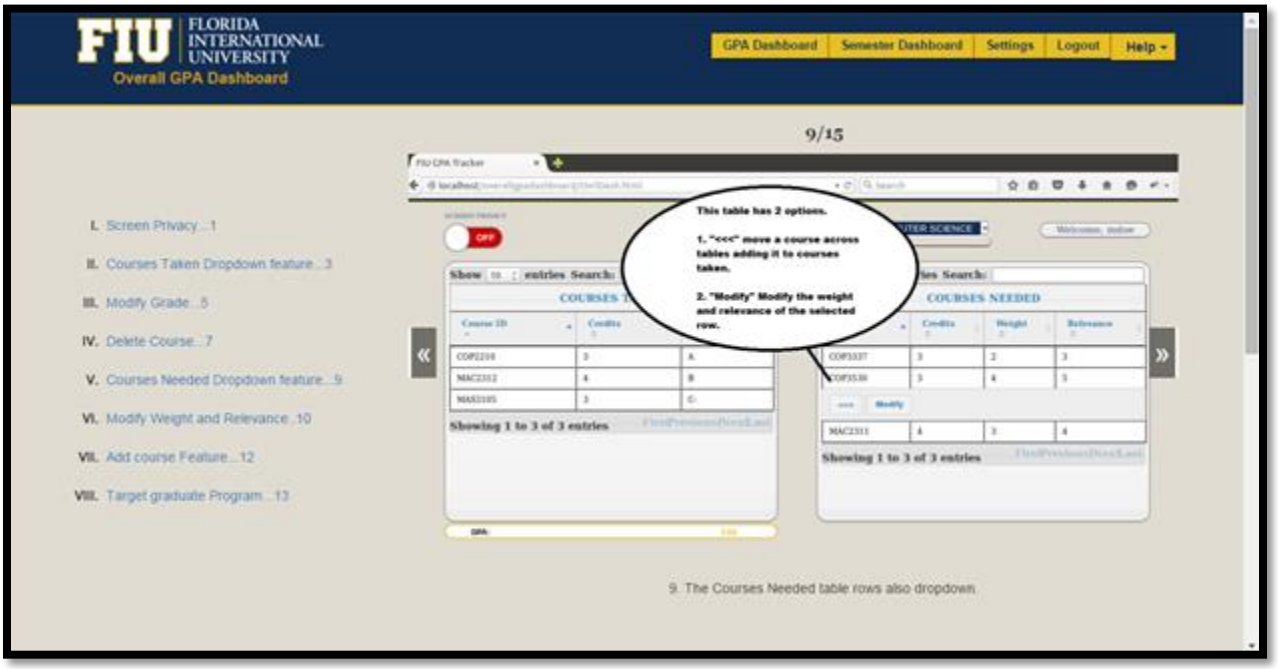


Figure B-14 GPA Dashboard "How-To"

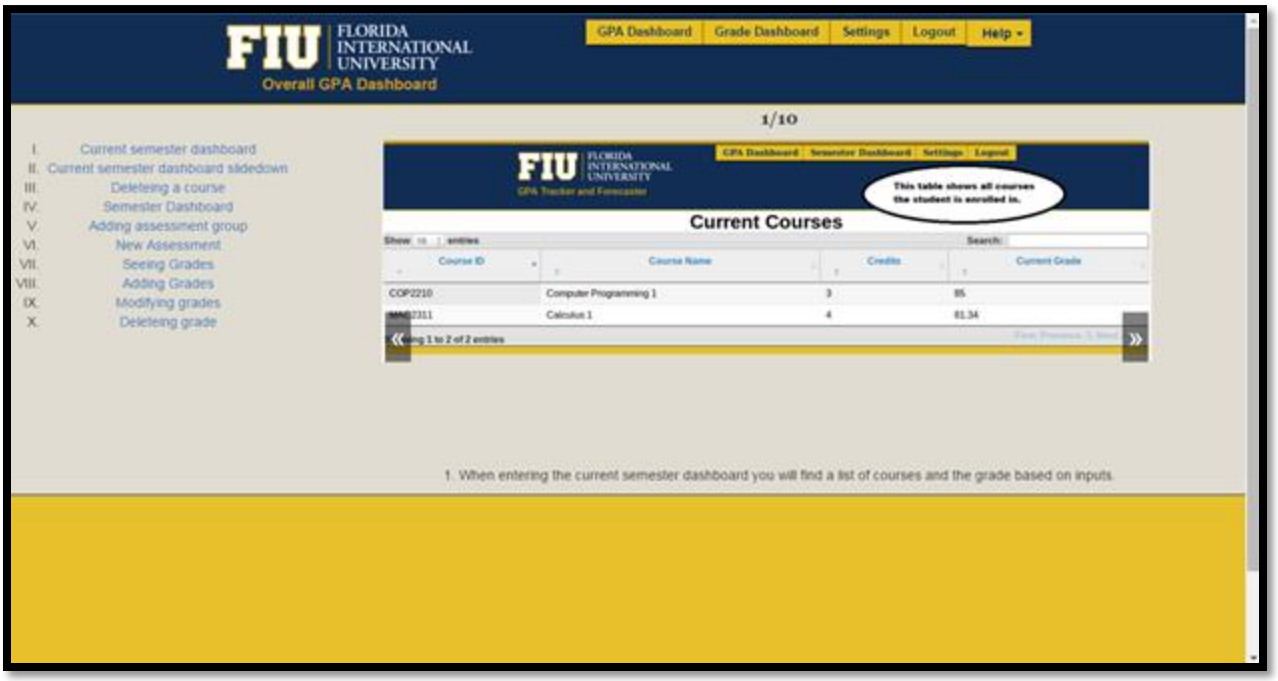


Figure B-15 Semester Dashboard "How-To"

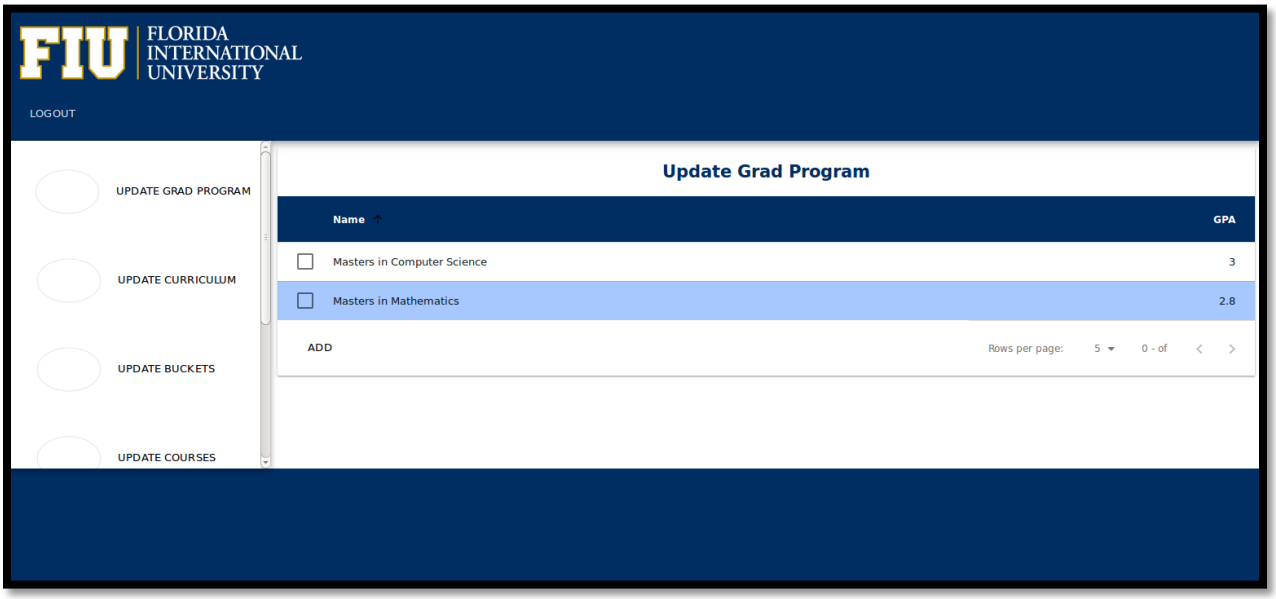


Figure B-16 Admin Page

Appendix C - Sprint Review Reports

At the end of every sprint, a review meeting was held between the developers and the product owner. This section in the appendix notes the remarks from the review meetings.

Sprint 1 Review

Date: 01/29/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Start time: 8:20PM
End time: 8:30PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #803 Complete JavaScript Tutorial
- User Story #804 Complete PHP / Python Tutorial
- User Story #805 Setup local development Environment
- User Story #807 Review PHP Tutorial

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story #809 Semester Forecast Report
    - Reason for rejection:
        - Not implemented yet
- User Story #811 Troubleshooting GPA Audit Import
    - Reason for rejection:
        - Not fully implemented

Sprint 2 Review

Date: 02/12/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Start time: 8:10PM
End time: 8:22PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #811 Error Logging
- User Story #823 Error Logging
- User Story #828 Semester Forecast Functionality

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story #819 Overall GPA Dashboard Forecast
    - Reason for rejection:
        - Not started yet
- User Story #820 Understand GPA Dashboard Data
    - Reason for rejection:
        - Not started yet

Sprint 3 Review

Date: 02/26/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Start time: 8:10PM
End time: 8:25PM

Discussed Topics:
After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #809 Formatted Semester Forecast Report
- User Story #827 Add Concurrency to Error Log

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story #819 Overall GPA Dashboard Forecast Report
    - Reason for rejection:
        - Not started
- User Story #820 Understand GPA Dashboard Data
    - Reason for rejection:
        - Not started
- User Story #826 Instantiate Curriculum Requirements
    - Reason for rejection:
        - Not started
- User Story #812 Improve Semester Dashboard Look and Feel
    - Reason for rejection:
        - Not started

Sprint 4 Review

Date: 03/11/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Discussed Topics:

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #826 Instantiate Curriculum Requirements
- User Story #846 DEFECT: Fix Data Points and Calculations
- User Story #852 DEFECT: Fix Graph in Semester Dashboard
- User Story #857 Migrate Semester Forecast Report to Object Oriented Design

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

- User Story #812 Improve Semester Dashboard Look and Feel
    - Reason for rejection:
        - Not started
- User Story #847 Migrate Error Log Feature to Object Oriented Design
    - Reason for rejection:
        - Not started
- User Story #820 Understand GPA Dashboard Data
    - Reason for rejection:
        - Not started
- User Story #819 GPA Forecast Report
    - Reason for rejection:
        - Not started

Sprint 5 Review

Date: 04/01/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Discussed Topics:

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #820 Understand GPA Dashboard Data
- User Story #819 Generate GPA Forecast Report
- User Story #867 DEFECT: Fix GPA Dashboard Graph
- User Story #887 Migrate Semester Dashboard Code to OO
- User Story #847 Migrate Error Log Feature to Object Oriented
- User Story #866 DEFECT: Fix JSON Errors on Sem. Dash.
- User Story #873 Map Admin Curriculum Import

Sprint 6 Review

Date: 04/15/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Discussed Topics:

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #812 Improve Semester Dashboard Look-and-Feel
- User Story #895 Update Student's Grades
- User Story #899 Update Grad Programs

Sprint 7 Review

Date: 04/29/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Discussed Topics:

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User Story #874 Migrate GPA Dashboard to AngularJS
- User Story #893 Improve Breakdown Look-and-Feel
- User Story #886 Project Future Data for Semester Forecast Reports
- User Story #885 Project Future Data for GPA Forecast Reports
- User Story #868 Semester Forecast Report Improvements

Appendix D - Sprint Retrospective Reports

At the end of every sprint, a review meeting was held between the developers and the product owner. This section in the appendix notes the remarks from the retrospective meetings.

Sprint 1 Retrospective

Date: 01/29/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Start time: 8:30PM
End time: 8:40PM

What went wrong?

- Did we do a good job estimating our team's velocity?
    - N/A
- Did we do a good job estimating the points (time required) for each user story?
    - N/A
- Did each team member work as scheduled?
    - Yes
- Disengagement / not communicating at first in Slack

What went right?

- Better communication in Slack (#shknowledge channel)
- Understand flow of website
- Good understanding of web development
- Able to navigate through code
- Acquainted with database design

How to address the issues in the next sprint?

- How to improve the process?
    - A lot more communication and sharing of knowledge
- How to improve the product?
    - N/A

Sprint 2 Retrospective

Date: 02/12/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Start time: 8:22PM
End time: 8:42PM

What went wrong?

- Did we do a good job estimating our team's velocity?
  - N/A
- Did we do a good job estimating the points (time required) for each user story?
  - N/A
- Did each team member work as scheduled?
  - Yes
- Time management needs to improve
- Clear, early identification of roadblock, obstacles should be addressed immediately
- Follow up on Mingle murmurs

What went right?

- Teamwork working well
- UML diagrams
- Understanding of database design
- Understand whatif report
- Story documentation

How to address the issues in the next sprint?

- How to improve the process?
  - If there is unnecessary code from previous semester, remove and continue
  - Reach out to each if stuck on something
  - Spread out time management properly
  - Complete documentation before implementation
  - Research
  - Complete user stories for the second Wednesday
- How to improve the product?
  - N/A

Sprint 3 Retrospective

Date: 02/26/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Start time: 8:25PM
End time: 8:40PM

What went wrong?

- Lizette:
    - Stick to user story
- Alex:
    - N/A

What went right?

- Even though stories were complex, group still pulled through
- Pseudo code algorithm is very good - helps a lot
- Alex:
    - scrum presentations improved
    - solution was really good
- Lizette:
    - pulled through on report

How to address the issues in the next sprint?

- How to improve the process?
    - get to algorithm / pseudo code ASAP, beginning of Sprint
- How to improve the product?
    - N/A

Discussed Topics:

- Did we do a good job estimating our team's velocity?
    - No
- Did we do a good job estimating the points (time required) for each user story?
    - No
- Did each team member work as scheduled?
    - Yes

Sprint 4 Retrospective

Date: 03/11/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez

What went wrong?

- Lizette:
  - ...
- Alex:
  - …
- Refactor with proper logging
- Produce pseudo algorithms quicker

What went right?

- Alex:
  - Doing really well
  - Instantiation completed
- Lizette:
  - Was falling behind but then came up to speed
  - Semester Forecast Report completed
- Error logging is coming along

How to address the issues in the next sprint?

- How to improve the process?
  - Pick up on the pace
  - Produce algorithms quicker
- How to improve the product?
  - N/A

Discussed Topics:

- Did we do a good job estimating our team's velocity?
  - No
- Did we do a good job estimating the points (time required) for each user story?
  - No
- Did each team member work as scheduled?
  - Yes

Sprint 5 Retrospective

Date: 04/01/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Start time: 9:35PM
End time: 9:45PM

What went wrong?

- Lizette:
  - Misunderstanding / communications
  - Catch errors gracefully
- Alex:
  - Didn’t complete all stories expected to complete

What went right?

- Alex:
  - Got to pick up the pace on user stories
- Lizette:
  - Pace has been picking up since last sprint
  - New ideas to add to site
  - More engaged and showing initiative

How to address the issues in the next sprint?

- How to improve the process?
  - Show faster results
  - Better use of resources
- How to improve the product?
  - N/A

Discussed Topics:

- Did we do a good job estimating our team's velocity?
  - No
- Did we do a good job estimating the points (time required) for each user story?
  - No
- Did each team member work as scheduled?
  - Yes

Sprint 6 Retrospective

Date: 04/15/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Start time: 8:00PM
End time: 8:20PM

What went wrong?

- Lizette:
  - Did not calculate well velocity of User Story
- Alex:
  - Didn't project velocity
- Better projection of how long it will take to accomplish a task

What went right?

- Alex:
  - Completed user stories
- Lizette:
  - Good understanding of new tools
  - Nice updates to look-and-feel

How to address the issues in the next sprint?

- How to improve the process?
  - Communicate how user stories are coming along and if will be completed on time
- How to improve the product?
  - N/A

Discussed Topics:

- Did we do a good job estimating our team's velocity?
  - No
- Did we do a good job estimating the points (time required) for each user story?
  - No
- Did each team member work as scheduled?
  - Yes

Sprint 7 Retrospective

Date: 04/29/2016
Attendees: Johann Henao, Lizette Mendoza, Alex Sanchez
Start time: 9:40PM
End time: 9:50PM

What went wrong?

- Lizette:
  - Ran into a number of hurdles
- Alex:
  - Difficulty with layout design

What went right?

- Alex:
  - Completed User Stories
  - New design updated for GPA Dashboard
- Lizette:
  - Made a lot of updates to website - migration to AngularJS
  - Completed 3 User Stories

How to address the issues in the next sprint?

- How to improve the process?
  - Communicate how user stories are coming along and if will be completed on time
- How to improve the product?
  - N/A

Discussed Topics:

- Did we do a good job estimating our team's velocity?
  - No
- Did we do a good job estimating the points (time required) for each user story?
  - No
- Did each team member work as scheduled?
  - Yes

REFERENCES