

Florida International University
School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title:
FIU GPA Tracker and Forecaster 3.0

Team Members:
Camila Perez

Product Owner(s):
Johann Henao

Mentor(s):
Johann Henao
Mohsen Taheri

Instructor:
Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 *Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

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 time 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 3.0 version of this system. Aspects of the makeup of the system and the functionality will be described in detail.

Table of Contents

INTRODUCTION	
CURRENT SYSTEM	5
PURPOSE OF NEW SYSTEM	5
USER STORIES	
IMPLEMENTED USER STORIES	6
PENDING USER STORIES	7
PROJECT PLAN	
HARDWARE AND SOFTWARE RESOURCES	9
SPRINTS PLAN	10
<i>Sprint 1</i>	10
<i>Sprint 2</i>	10
<i>Sprint 3</i>	10
<i>Sprint 4</i>	11
<i>Sprint 5</i>	12
<i>Sprint 6</i>	12
<i>Sprint 7</i>	13
SYSTEM DESIGN	
ARCHITECTURAL PATTERNS	14
SYSTEM AND SUBSYSTEM DECOMPOSITION	15
DEPLOYMENT DIAGRAM	16
DESIGN PATTERNS	16
SYSTEM VALIDATION	17
GLOSSARY	23
APPENDIX	24
APPENDIX A - UML DIAGRAMS	24
<i>Static UML Diagrams</i>	24
<i>Dynamic UML Diagrams</i>	25
APPENDIX B - USER INTERFACE DESIGN	29
APPENDIX C - SPRINT REVIEW REPORTS	36
APPENDIX D - USER MANUALS, INSTALLATION/MAINTENANCE DOCUMENT, SHORTCOMINGS/WISHLIST DOCUMENT AND OTHER DOCUMENTS	38
REFERENCES	43

INTRODUCTION

The FIU GPA Tracker and Forecaster 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 2.0 allows students to see what they need to attain their GPA goal based on the information they input into the system. Currently, this is be done by means of forecast reports. The reports show the student what averages/grades they need to maintain in each of their remaining courses based on the weight and relevance selected. It also displays 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.

In addition, the admin user was given more privileges to better handle the curriculum information and student data. Also, the system partially has a modern look and feel that is more appealing to the users. Lastly, the error logging feature was improved to increase the system's maintenance.

Purpose of New System

The purpose of the FIU GPA Tracker and Forecaster 3.0 is to enhance the set of student and admin functionalities, as well as update the physical design of the website. The system will allow students to export their data to their PC as an XML file, completely delete their data from the website, and import an XML backup of their data. Also, the website's look and feel will be updated so that it is more appealing to the users.

In addition, admin users will be able to automatically generate an XML file containing the description of a new curriculum they which to import to the system. Last, the error logging feature and the security of the website will be improved.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the FIU GPA Tracker and Forecaster 3.0 project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

User Story #927 Do full regression test of GPA Dashboard and fix bugs encountered.

As a developer, I would like to perform a full regression test of the GPA dashboard and fix bugs encountered so that students can properly use this page.

User Story #930 Migrate Settings page to AngularJS.

As a developer, I would like to migrate the Settings page to AngularJS so that the page is consistent with the rest of the website.

User Story #943 Fix Import Data functionality.

As a student, I would like to use the import functionality so that I am able to recover data with an XML backup that I previously downloaded using the export functionality.

User Story #939 Understand and document PHP unit tests for GPA Dashboard.

As a developer, I want to understand and document the PHP Unit Tests for the GPA Dashboard so that continued testing is allowed as new features are added to the website.

User Story #928 Fix and fully test error log feature.

As a developer, I would like to fully test the FIU GPA Tracker log feature and fix any bugs encountered along the way. Also I would like to refactor the code.

User Story #931 Create test drivers for Major Programs import and GPA import.

As a developer, I want to use test drivers for the GPA Audit Import and the Program Curriculum Import so that I can test the website with different scenarios.

User Story #876 Improve website security.

As a developer, I would like to ensure the FIU GPA Tracker and Forecaster website has the minimum security standard configuration to prevent hackers from exploiting any technology weakness.

User Story #956 Defect: Fix issues about basic functionality of test drivers.

As a developer, I want to fix issues pertaining the basic functionality of the Curriculum Test Drivers so that I can properly create different curriculums.

User Story #960 Integrate Test Driver XML files with Curriculum and GPA Audit import.

As an administrator, I would like to be able to import the curriculum and GPA Audit XML output from the Test Drivers so that I can automatically enable new curriculums and create new test student data.

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 #929 Do full regression test of Semester Dashboard and fix bugs.

As a developer I would like to perform a full regression test of the Semester Dashboard and fix bugs encountered. This includes updating existing documentation, UML diagrams, object diagram as well as refactor the code and eliminate unnecessary/unused code.

User Story #953 Fix forgiveness policy issue on GPA Graph.

As a student, I want the GPA graph to take into account the FIU forgiveness policy so that I can see my correct GPA history.

PROJECT PLAN

The scrum development method was used for this project. In scrum, development is broken down into sprints. For this project, each sprint 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. We had the following roles during the project:

- Camila Perez: Developer, Scrum Master
- Johann Henao: Product Owner

Hardware and Software Resources

The following hardware and software resources were used in this project:

Software:

- Oracle VM VirtualBox Manager Version 5.1.4: 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:

- HP – Envy x360
 - 7th Gen Intel® Core™ i5-7200U mobile processor
 - System Memory: 12GB
 - Operating System: Windows 10
 - Hard Disk Size: 1TB

Sprints Plan

Sprint 1

(8/29/2016 – 9/9/2016)

(Complete JavaScript, PHP, Python, AngularJS tutorials. Set up local environment)

Sprint 2

(9/12/2016 – 9/23/2016)

USER STORY #927 DO FULL REGRESSION TEST OF GPA DASHBOARD AND FIX BUGS ENCOUNTERED.

Tasks:

- 933 Refactor GPA Dashboard Code.
- 934 Fix next/previous buttons.
- 935 Complete regression test for GPA Dashboard.

Acceptance Criteria:

- GPA Dashboard features from previous semester work as expected with different test cases.
- Bugs are eliminated and errors are handled gracefully.
- Code has been refactored and unneeded code removed.

Modeling:

Please refer to Figure A-3 in Appendix A for the use case diagram.

Sprint 3

(9/26/2016 – 10/7/2016)

USER STORY #930 MIGRATE SETTINGS PAGE TO ANGULARJS.

Tasks:

- 936 Migrate Settings navigation bar to AngularJS.
- 937 Migrate Import Audit button to AngularJS.
- 938 Update Settings page's look and feel.
- 940 Migrate Import Data button to AngularJS.
- 941 Migrate Export Data button to AngularJS.
- 942 Migrate Delete Data button to AngularJS.

Acceptance Criteria:

- Page look and feel is consistent with the rest of the website.
- Each button works as expected.

Modeling:

Please refer to Figure A-3 in Appendix A for the use case diagram.

Sprint 4

(10/10/2016 – 10/21/2016)

USER STORY #943 FIX IMPORT DATA FUNCTIONALITY.

Tasks:

- 944 Fix Import Data Functionality.
- 945 Fix Delete Data Functionality.
- 946 Test import/export/delete functionalities.

Acceptance Criteria:

- The user is able to upload data as an XML file.
- The data contained in the XML file that is to be uploaded is GPA, major, declared date, and for each course, grade, weight, relevance, course id, semester, and year.
- The uploaded data should only alter the data of the logged in user.

Modeling:

Please refer to Figure A-4 in Appendix A for the sequence diagram that was created. Please refer to Figure A-3 in Appendix A for the use case diagram.

USER STORY #939 UNDERSTAND AND DOCUMENT PHP UNIT TESTS FOR GPA DASHBOARD.

Acceptance Criteria:

- Fully understand how PHP Unit Tests work.
- Document PHP Unit Tests for GPA Dashboard.

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 #928 FIX AND FULLY TEST LOG FEATURE.

Acceptance Criteria:

- Log feature works with Debug, Info, Warning, Error levels.
- UML diagrams and documentation are updated to reflect any fixes/adjustments performed.
- Code is refactored and documentation is updated with any fixes/adjustments performed.

Sprint 5
(10/24/2016 – 11/4/2016)

USER STORY #931 CREATE TEST DRIVERS FOR MAJOR PROGRAMS IMPORT AND GPA IMPORT.

Tasks:

- 947 Create UMLs and pseudo code.
- 948 Create user interface.
- 951 Develop admin test driver.
- 952 Develop student test driver.

Acceptance Criteria:

- There should be a user interface that allows the user to create data scenarios (one interface for GPA Audit Import and one for Program Curriculum Import).
- Test drivers should generate data scenarios as xml files.
- Test drivers should import the data scenarios to the database.
- Test drivers should validate that the import is successful.

Modeling:

Please refer to Figure A-5 in Appendix A for the sequence diagram that was created. Please refer to Figure A-7 in Appendix A for the use case diagram.

USER STORY #876 IMPROVE WEBSITE SECURITY.

Tasks:

- 949 Research and Implement Linux OS minimum security recommendations.
- 950 Research and implement Apache recommended Security configurations.
- 954 Research and implement PHP minimum security recommendations.
- 955 Perform Penetration Test to Identify vulnerabilities.

Acceptance Criteria:

- OS and file system has the minimum standard security configuration.
- Apache has the minimum standard security configuration.
- PHP has the minimum standard security configuration.
- MySQL has the minimum standard security configuration.

Sprint 6
(11/7/2016 – 11/18/2016)

USER STORY #956 DEFECT: FIX ISSUES ABOUT BASIC FUNCTIONALITY OF TEST DRIVERS.

Tasks:

- 957 Fix infinite child addition functionality.
- 958 Fix issues pertaining course addition.
- 959 Test the test drivers.

Acceptance Criteria:

- The user should be able to add an infinite number of children in the Curriculum Test Driver.
- The user should be able to properly add courses in the Curriculum Test Driver.

Modeling:

Please refer to Figure A-6 in Appendix A for the sequence diagram that was created. Please refer to Figure A-7 in Appendix A for the use case diagram.

USER STORY #960 INTEGRATE TEST DRIVER XML FILES WITH CURRICULUM AND GPA AUDIT IMPORT.

Tasks:

- 961 Fix admin import feature.
- 962 Add import functionality to Audit test driver.

Acceptance Criteria:

- Admin import feature should be able to import the test driver Curriculum XML.
- Python parser should produce an xml file for the GPA Audit import.
- A button should be created to be able to import any GPA Audit xml test driver output file, the user should be prompted to select a file from the local computer.

Modeling:

Please refer to Figure A-5 in Appendix A for the sequence diagram that was created. Please refer to Figure A-8 in Appendix A for the use case diagram.

Sprint 7

(11/21/2016 – 12/2/2016)

(Work on poster, final presentation, final deliverable, videos)

SYSTEM DESIGN

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

Architectural Patterns

The MVC pattern was used as it allows the separation of the backend logic from the front end. Also, AngularJS is based on MVC. This makes it a natural choice to use when developing web applications.

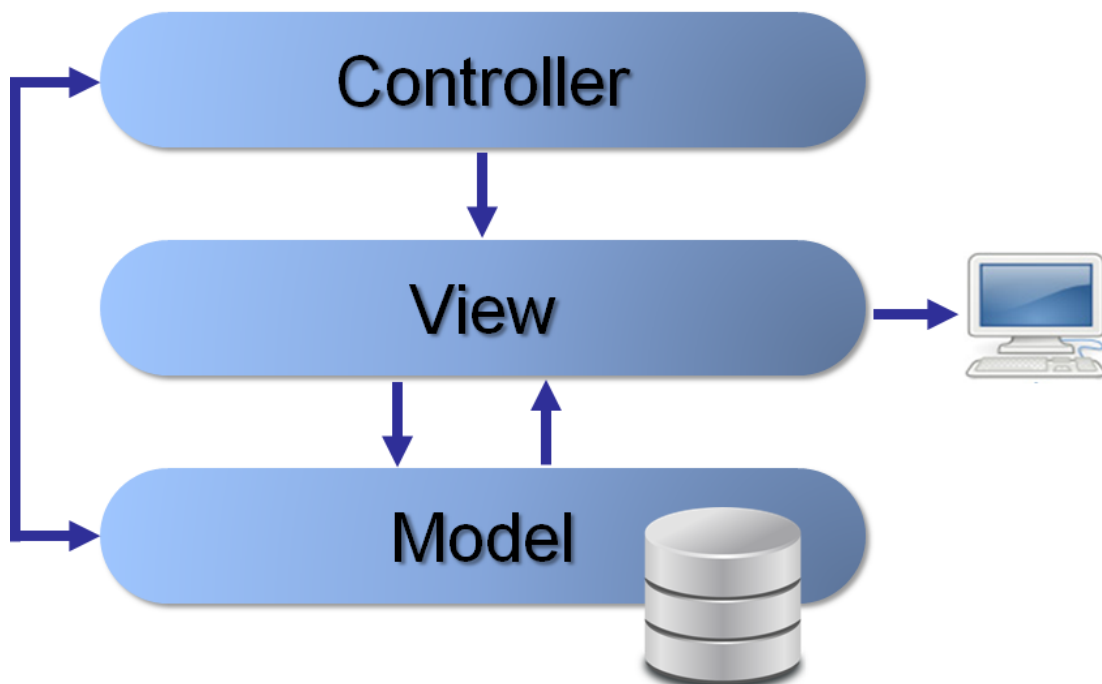


Figure 1.1 Model-View-Controller Architecture

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.

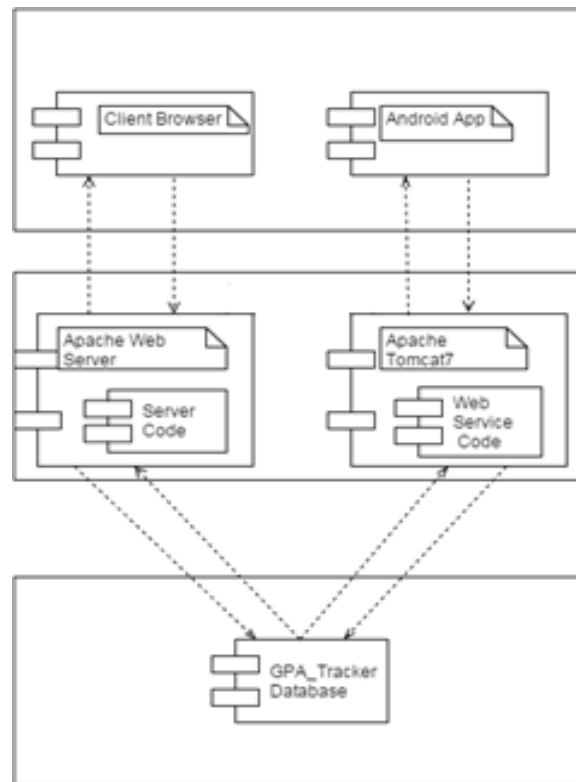


Figure 1.2 Subsystem Decomposition Diagram

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.

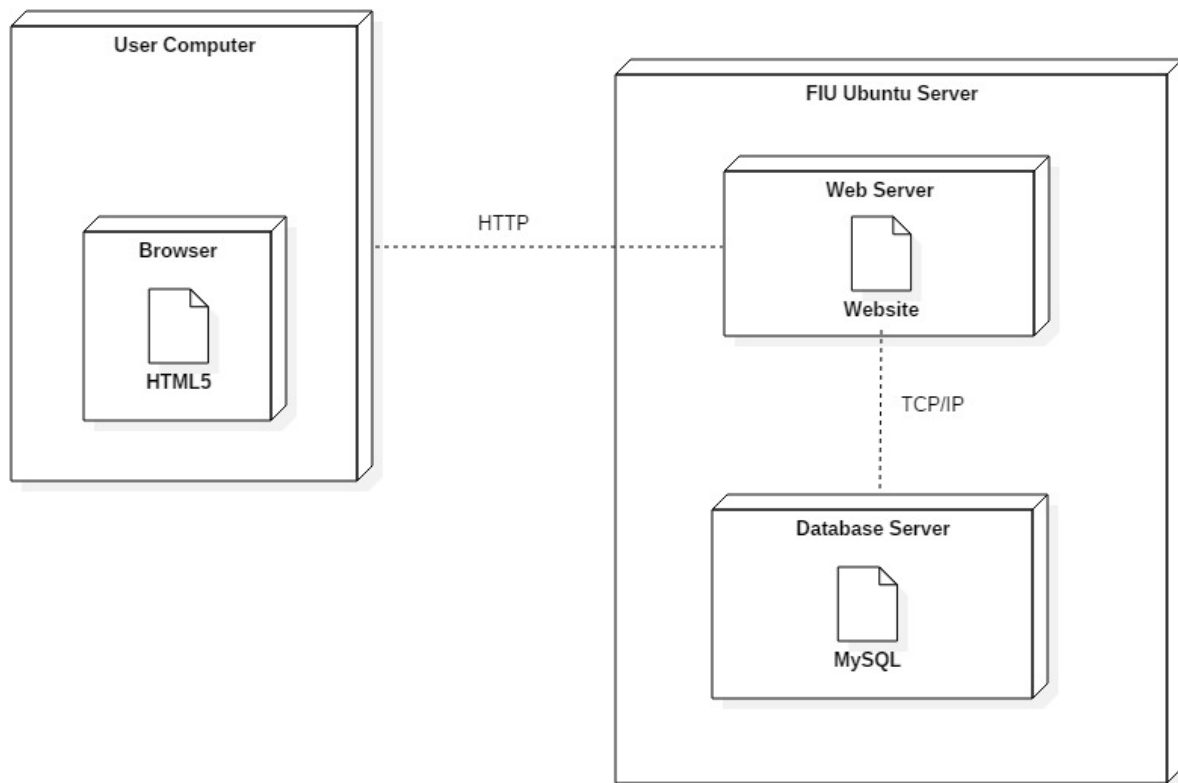


Figure 1.3 Deployment Diagram

Design Patterns

On top of the MVC architecture, this web solution also takes advantage of the client-server pattern. Frequent ajax calls are made from the user's browser, which acts as the client, to the server. The server can receive connections from different clients and create multiple processes to handle them.

SYSTEM VALIDATION

System testing and unit testing were conducted on the user stories below. Unit tests were performed on the system using PHPUnit. This allowed the automation of tests for the continued testing as new features were added to the website. System testing was conducted using sunny and rainy day test cases.

USER STORY #927 DO FULL REGRESSION TEST OF GPA DASHBOARD AND FIX BUGS ENCOUNTERED.

System Tests:

Test Case	Sunny Day
Purpose	To test that the courses lists are displayed properly empty when a user has not imported his/her Audit.
Precondition	- Website running. - GPA Dashboard displayed.
Input Values	- PDF Audit not imported.
Expected Output	Courses lists will be displayed as empty and the list label will be “0 – 0 of 0.”

Unit Tests:

USER STORY #930 MIGRATE SETTINGS PAGE TO ANGULARJS.

System Tests:

Test Case	Sunny Day
Purpose	To test that the “Import Panther Degree (PDF)” button properly uploads the file.
Precondition	- Website running. - Settings page displayed.
Input Values	- The user clicks on the “Import Panther Degree (PDF)” button. - The user selects the PDF file to be uploaded from the file uploader window.
Expected Output	- The file uploads correctly and fields on GPA Dashboard and Semester Dashboard are filled appropriately. - A message saying “PDF uploaded successfully” is displayed.

USER STORY #943 FIX IMPORT DATA FUNCTIONALITY.

System Tests:

Test Case	Sunny Day
Purpose	To test that the “Import Data” button properly imports the data.
Precondition	- Website running. - Settings page displayed.
Input Values	- The user clicks on the “Import Data” button. - The user selects the XML file to be imported from the file uploader window.
Expected Output	Message “Failed loading XML” displayed.

Test Case	Rainy Day
Purpose	To test that a student cannot import an XML file that was previously exported by other student.
Precondition	- Website running. - Settings page displayed.
Input Values	- The user clicks on the “Import Data” button. - The user selects the XML file to be imported from the file uploader window.
Expected Output	Data for this user is updated correctly.

Unit Tests:

- test04 - Ensure that a user's courses are deleted from the database when the user deletes data
 - Pass
- test05 - Ensure that the correct user's course info is pushed into the database for a user that imports data
 - Pass

USER STORY #939 UNDERSTAND AND DOCUMENT PHP UNIT TESTS FOR GPA DASHBOARD.

Unit Tests:

- testGetCurrProgram001_UT004 - Ensure that the correct program is returned for a user who imported the Audit
 - Pass
- testGetGPA001_UT005 - Ensure that the correct GPA is returned for a user who imported the Audit
 - Pass
- testCheckWeightAndRelevance001_UT001 - Ensure that all the in-progress and incomplete classes along with their weight and relevance values are returned for a user with imported Panther Audit
 - Pass

USER STORY #928 FIX AND FULLY TEST LOG FEATURE.

Unit Tests:

- testErrorLevel_EntryExists - Error should only Write Error log entries and nothing else
 - Pass
- testWarnigLevel_EntryExists - Warning should write Warning log entries and Error log entries
 - Pass
- testInfoLevel_EntryExists - Info should write: Info, Warning and Error
 - Pass
- testDebugLevel_EntryExists - Debug should write: Debug, Info, Warning and Error
 - Pass

USER STORY #931 CREATE TEST DRIVERS FOR MAJOR PROGRAMS IMPORT AND GPA IMPORT.

System Tests:

Test Case	Sunny Day
Purpose	Ensure base data is submitted correctly.
Precondition	Curriculum or Audit Test Driver running.
Input Values	User submits base data.
Expected Output	XML file shows base data correctly.

Test Case	Sunny Day
Purpose	Ensure rows are added to the tables as user inputs elements (buckets, courses) to each table.
Precondition	Curriculum or Audit Test Driver running.
Input Values	User adds elements to tables.
Expected Output	Rows are dynamically added to corresponding table.

USER STORY #956 DEFECT: FIX ISSUES ABOUT BASIC FUNCTIONALITY OF TEST DRIVERS.

System Tests:

Test Case	Sunny Day
Purpose	Ensure many levels of child buckets can be added to data table.
Precondition	Curriculum Test Driver running.
Input Values	User adds many levels of child buckets.
Expected Output	Child buckets are correctly displayed in table, each inside its parent.

Test Case	Sunny Day
Purpose	Ensure correct data is displayed as user collapses and then expands row.
Precondition	Curriculum Test Driver running. Data has been input to data tables.
Input Values	User collapses row. User expands same row.
Expected Output	Correct data (child buckets/courses) is displayed under expanded row.

Test Case	Sunny Day
Purpose	Ensure child buckets and courses are placed correctly in XML file.
Precondition	Curriculum Test Driver running. Data has been input to data tables.
Input Values	User generates an XML file from input data.
Expected Output	Child buckets are correctly displayed in XML file, each child bucket or course is inside its parent.

Test Case	Rainy Day
Purpose	Ensure system won't let the user leave empty fields in form.
Precondition	Curriculum Test Driver running.
Input Values	User leaves empty field while inputting program, bucket, or course data. User clicks submit button.
Expected Output	Message "Empty Fields" displayed

Test Case	Rainy Day
Purpose	Ensure system won't let the user proceed with the data entry unless the base program data has been submitted first.
Precondition	Curriculum Test Driver running. Program Data not submitted.
Input Values	User inputs bucket or course data. User clicks submit button.
Expected Output	Message "Submit program data first" displayed

USER STORY #960 INTEGRATE TEST DRIVER XML FILES WITH CURRICULUM AND GPA AUDIT
System Tests:

Test Case	Sunny Day
Purpose	Ensure curriculum data is correctly imported to database after admin import.
Precondition	Admin logged in to website.
Input Values	Admin selects xml file to import.
Expected Output	Database is updated with correct data.

Test Case	Sunny Day
Purpose	Ensure new student is registered after Audit xml file has been imported.
Precondition	Audit Test Driver running.
Input Values	Developer imports Audit xml.
Expected Output	New student registration data is correctly imported to database.

Test Case	Rainy Day
Purpose	Ensure system doesn't crash if audit xml file contains a course that is not currently stored in the database.
Precondition	Audit Test Driver running.
Input Values	Developer imports Audit xml with a course that is not currently stored in the database.
Expected Output	New course is first inserted, then assigned to student.

GLOSSARY

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:

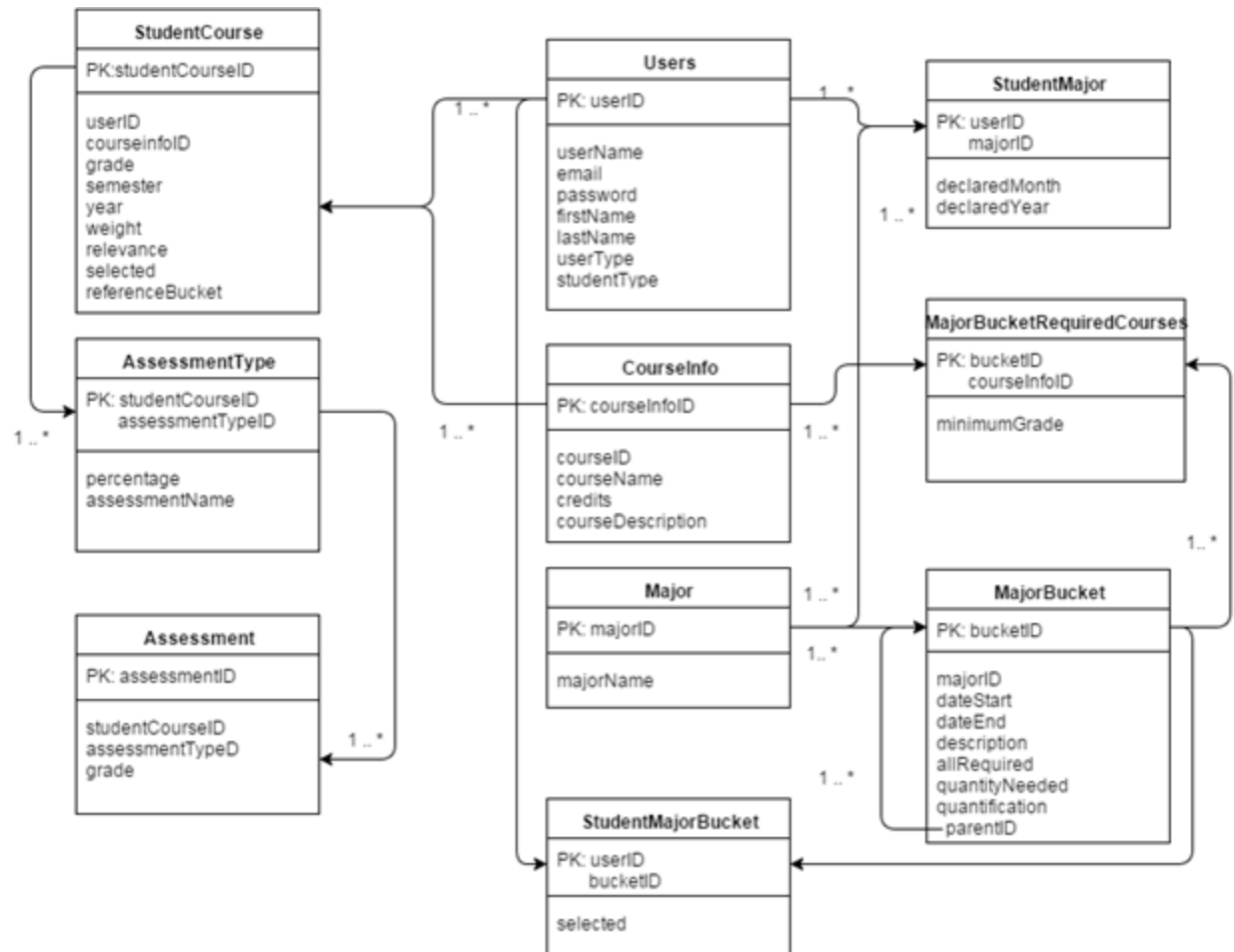


Figure A-1 - Persistent Data Design Diagram

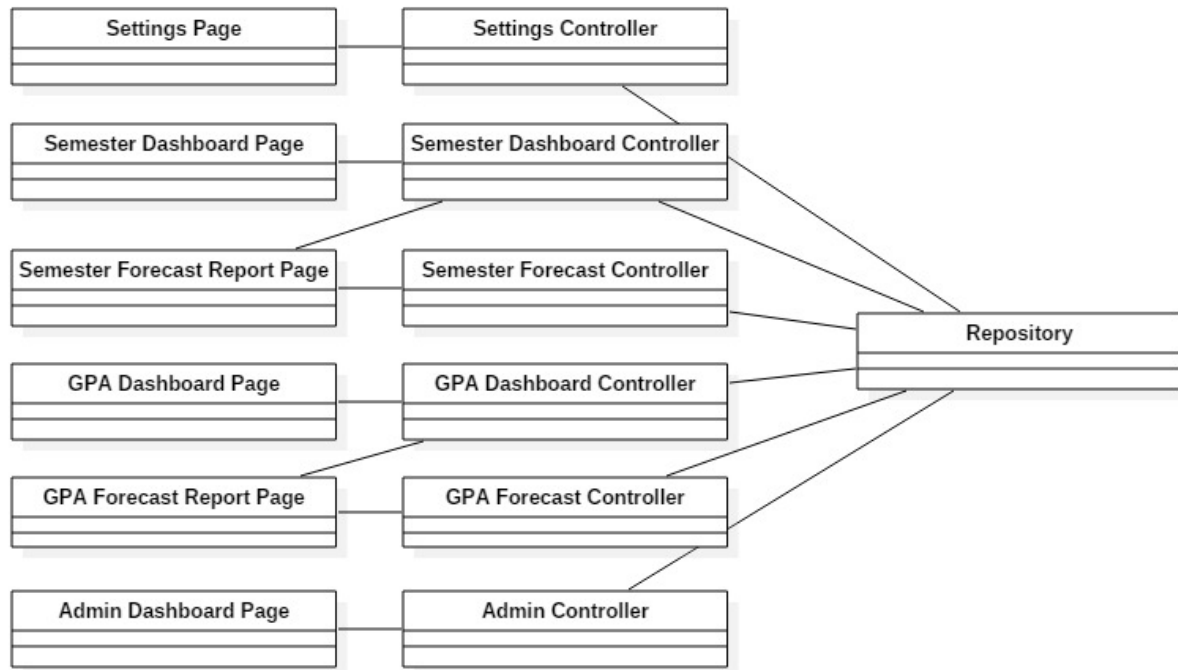


Figure A-2 - Minimal Class Diagram

Dynamic UML Diagrams:

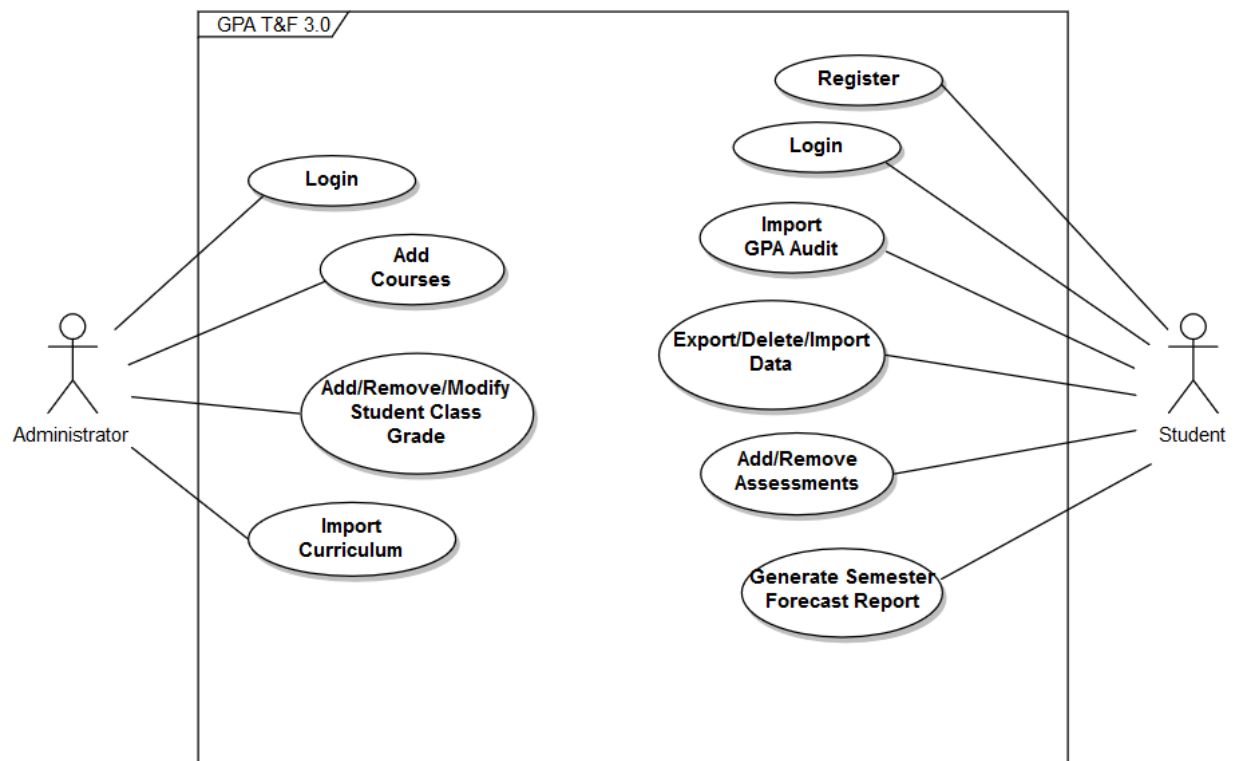
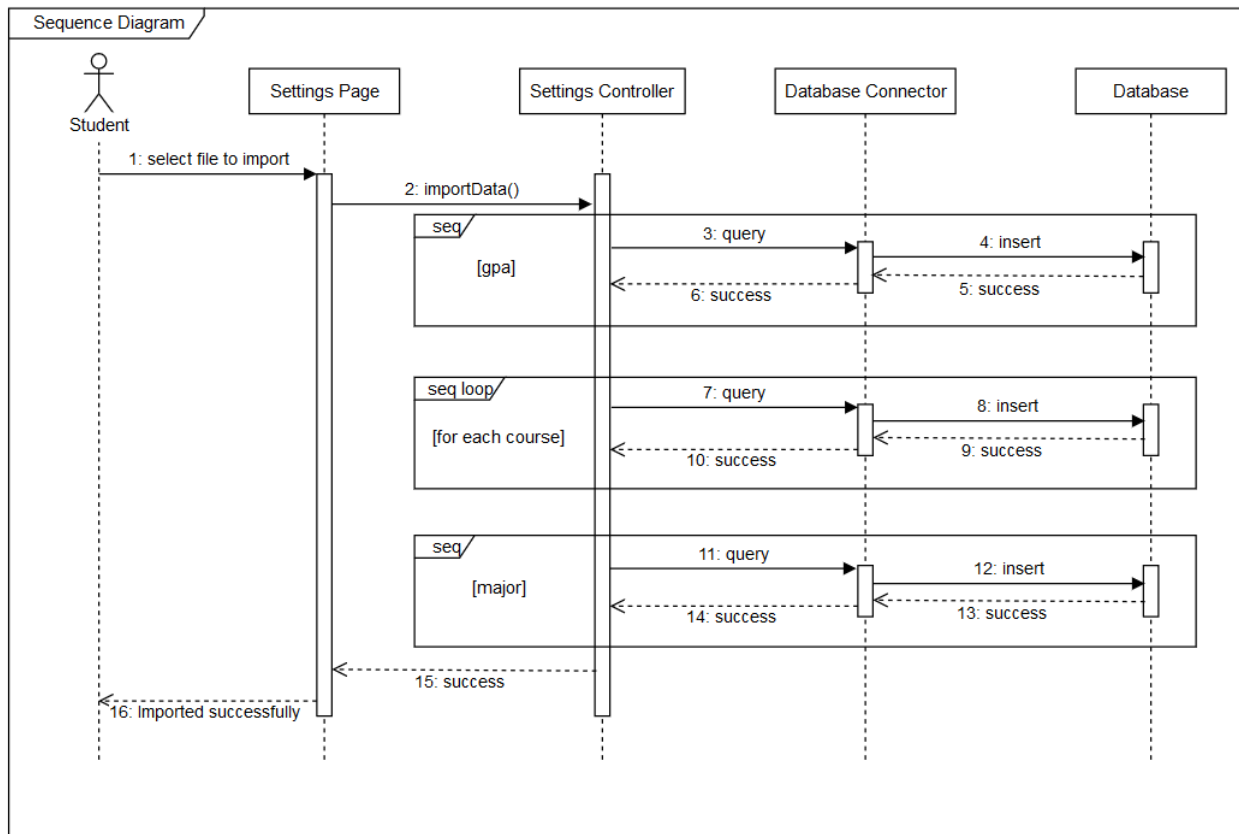


Figure A-3 Use Case Diagram

**Figure A-4** Import Data

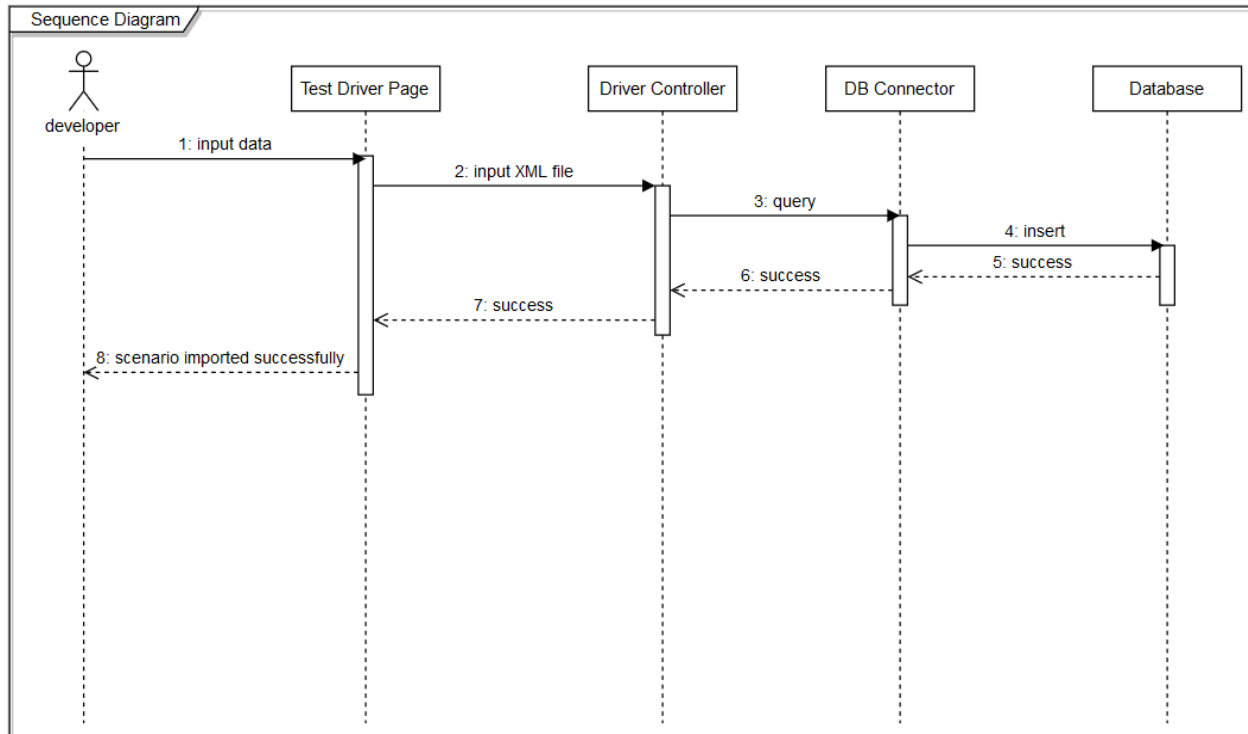


Figure A-5 Test Driver

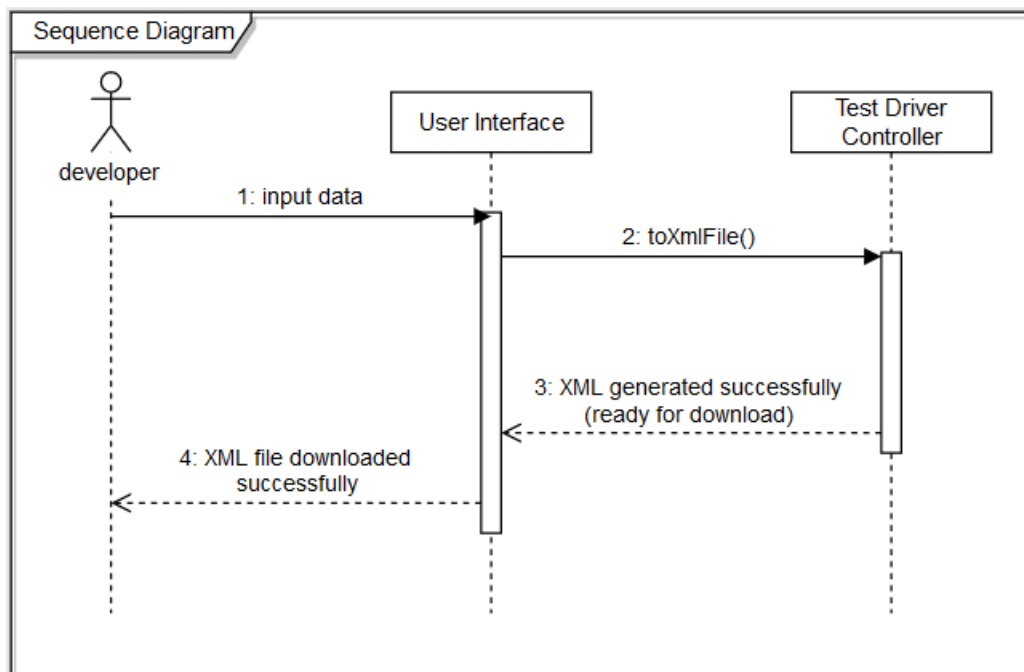


Figure A-6 Test Driver

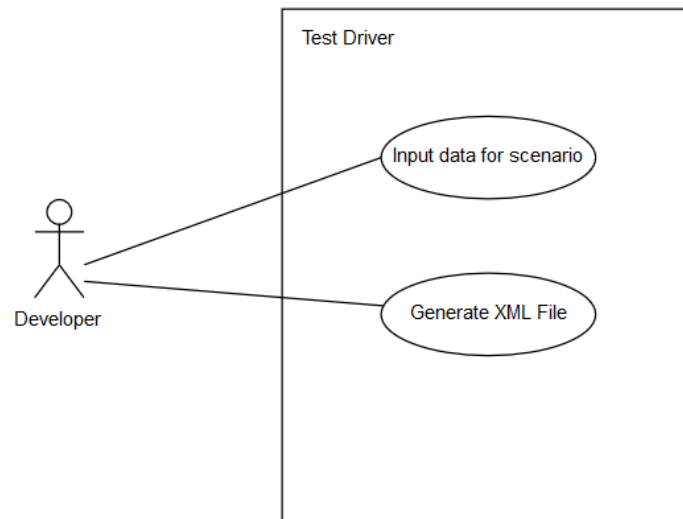


Figure A-7 Use Case Diagram for Test Drivers

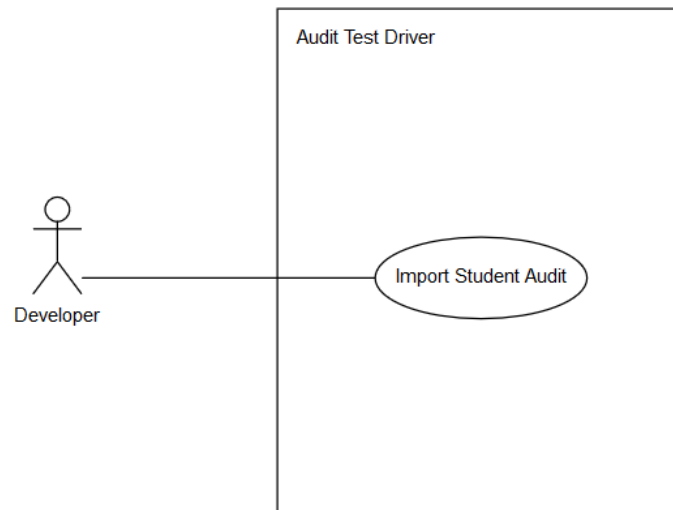


Figure A-8 Use Case Diagram for Audit Test Drivers

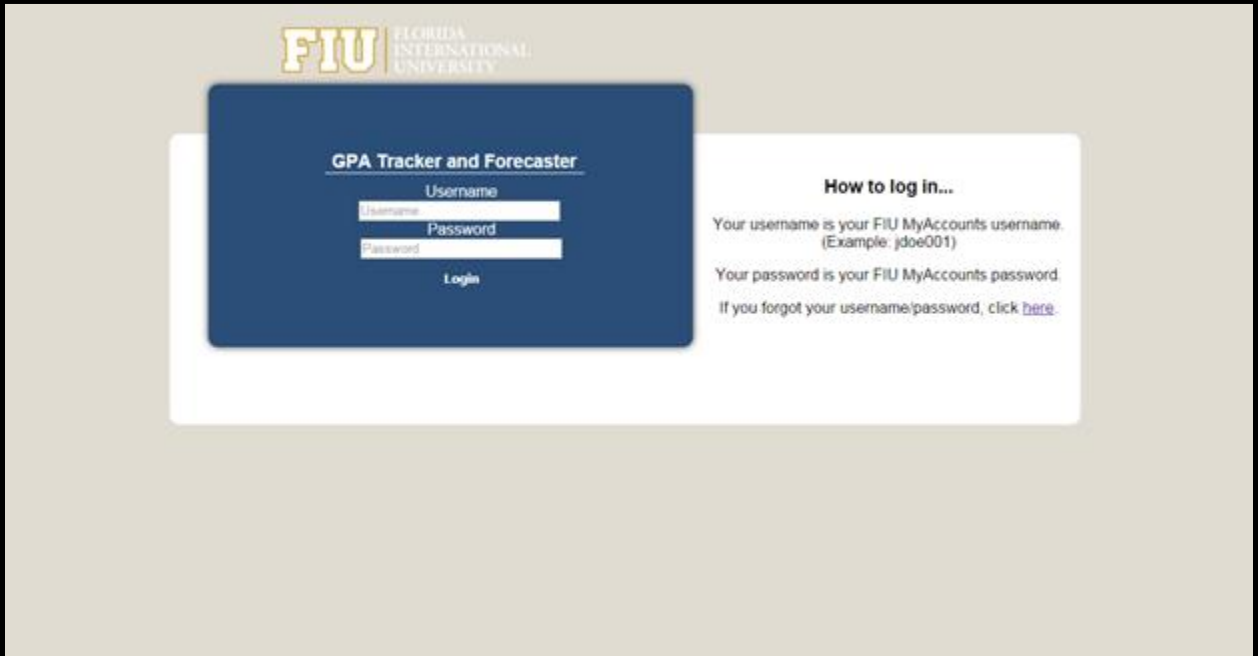
Appendix B - User Interface Design



Figure B-1 Home Page

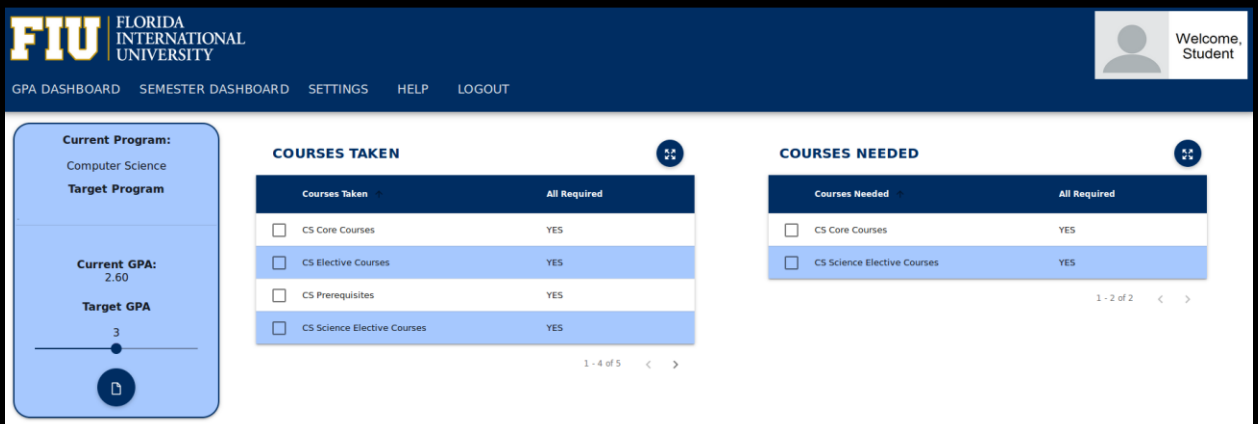


Figure B-2 Registration



The login screen features the FIU logo at the top left. The main content area is divided into two sections. On the left, a dark blue box titled "GPA Tracker and Forecaster" contains a login form with fields for "Username" and "Password", and a "Login" button. On the right, a white box titled "How to log in..." provides instructions: "Your username is your FIU MyAccounts username. (Example: jdoe001)", "Your password is your FIU MyAccounts password.", and "If you forgot your username/password, click [here](#)."

Figure B-3 Log In



The dashboard displays the FIU logo and navigation links: GPA DASHBOARD, SEMESTER DASHBOARD, SETTINGS, HELP, and LOGOUT. A user profile section on the right shows a placeholder icon and the text "Welcome, Student". The main content area is divided into three sections:

- Current Program:** Computer Science
- Target Program:** (empty)
- Current GPA:** 2.60
- Target GPA:** 3.0 (indicated by a slider)

Below these are two tables:

COURSES TAKEN	
Courses Taken	All Required
<input type="checkbox"/> CS Core Courses	YES
<input type="checkbox"/> CS Elective Courses	YES
<input type="checkbox"/> CS Prerequisites	YES
<input type="checkbox"/> CS Science Elective Courses	YES

1 - 4 of 5

COURSES NEEDED	
Courses Needed	All Required
<input type="checkbox"/> CS Core Courses	YES
<input type="checkbox"/> CS Science Elective Courses	YES

1 - 2 of 2

Figure B-4 GPA Dashboard



Figure B-5 GPA Dashboard Graph

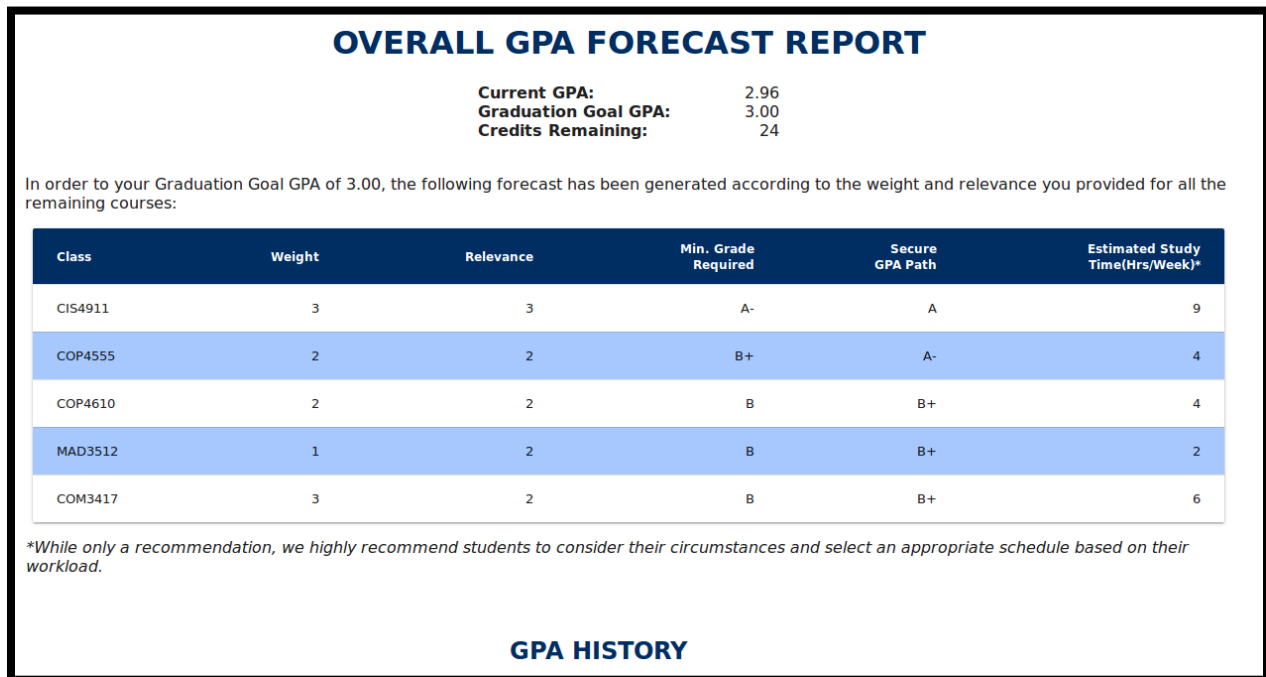


Figure B-6 GPA Forecast Report Data Table

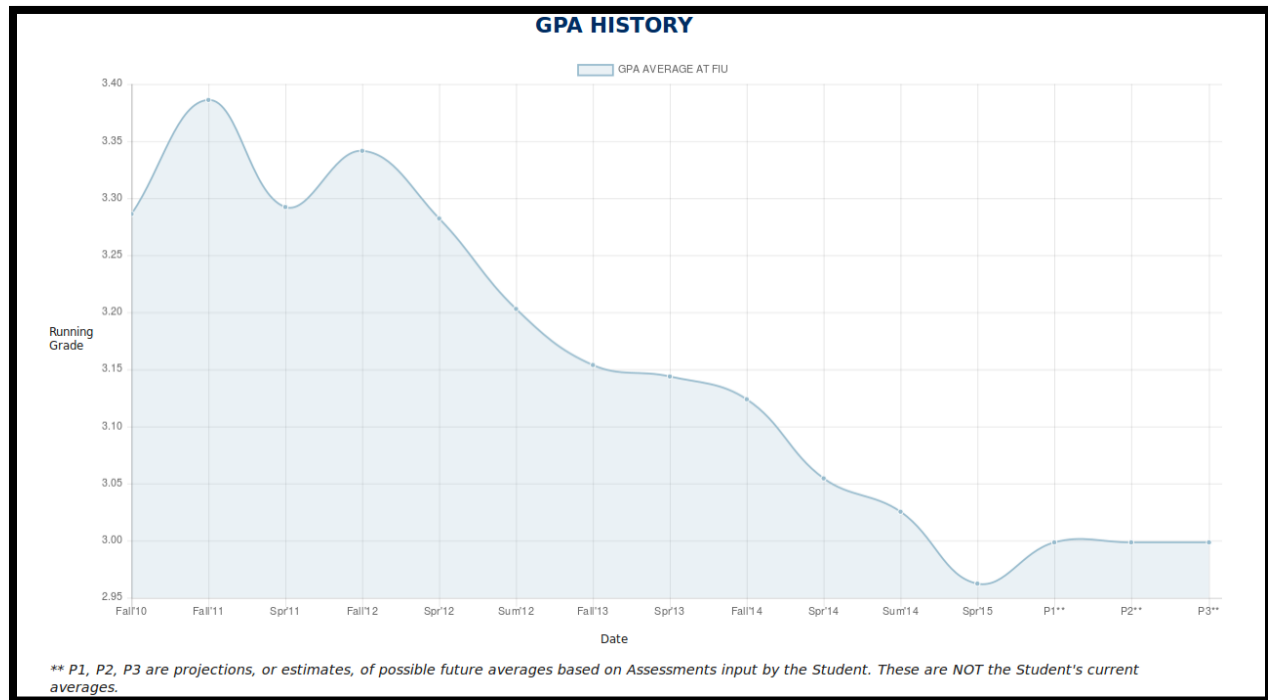


Figure B-7 GPA Forecast Report Graph + Projection

**FLORIDA
INTERNATIONAL
UNIVERSITY**

Welcome,
Student

[GPA DASHBOARD](#)
[SEMESTER DASHBOARD](#)
[SETTINGS](#)
[HELP](#)
[LOGOUT](#)

SEMESTER DASHBOARD

CURRENT COURSES

Course ID	Course Name	Credits	Current Grades
<input type="checkbox"/> MAD3512	Introduction to Theory of Algorithm	3	No Grades
<input type="checkbox"/> COP4610	Operating Syst Princ	3	90.73
<input type="checkbox"/> COP4555	Principles of Programming Languages	3	82.72
<input type="checkbox"/> CIS4911	Senior Project	3	85

Figure B-8 Semester Dashboard

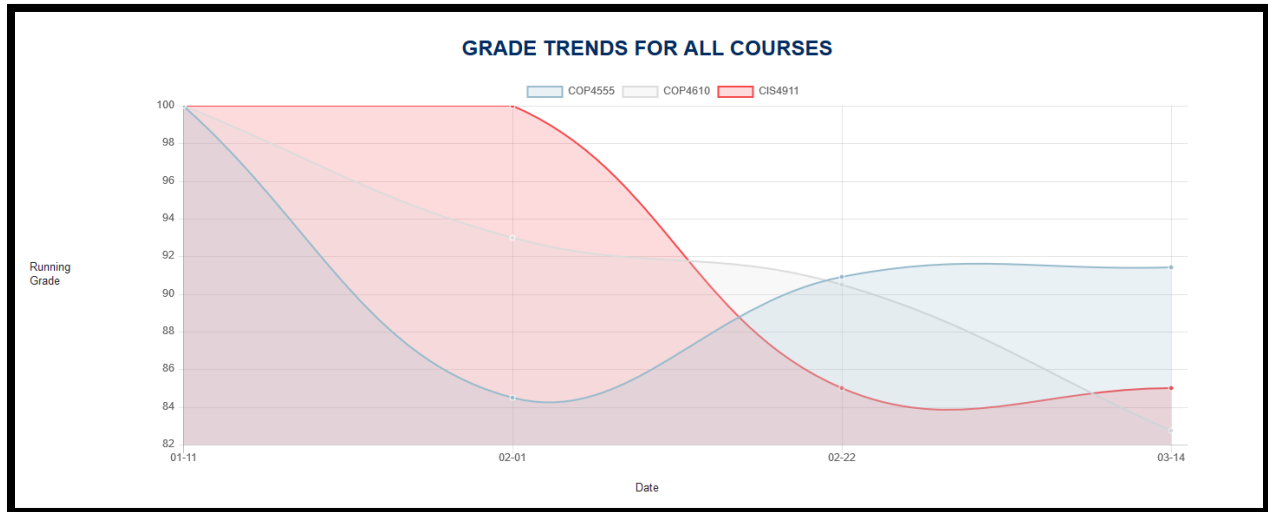


Figure B-9 Semester Forecast Graph

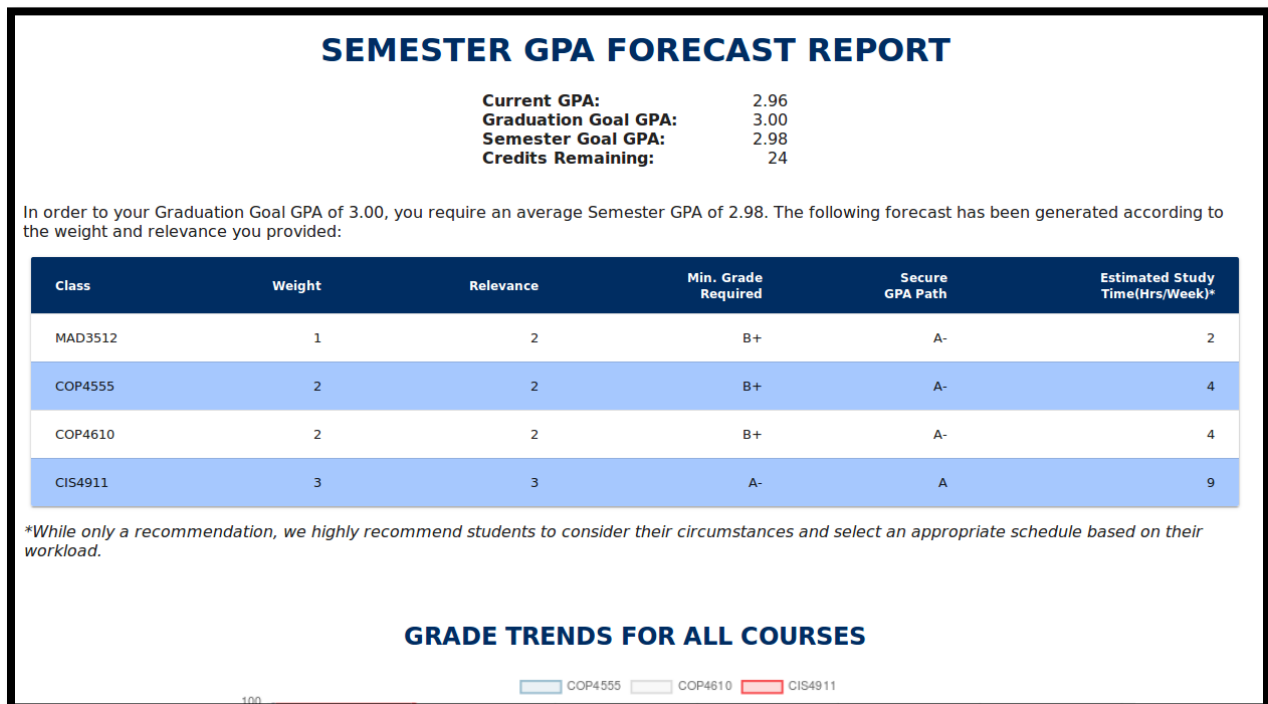


Figure B-10 Semester Forecast Report Data Table

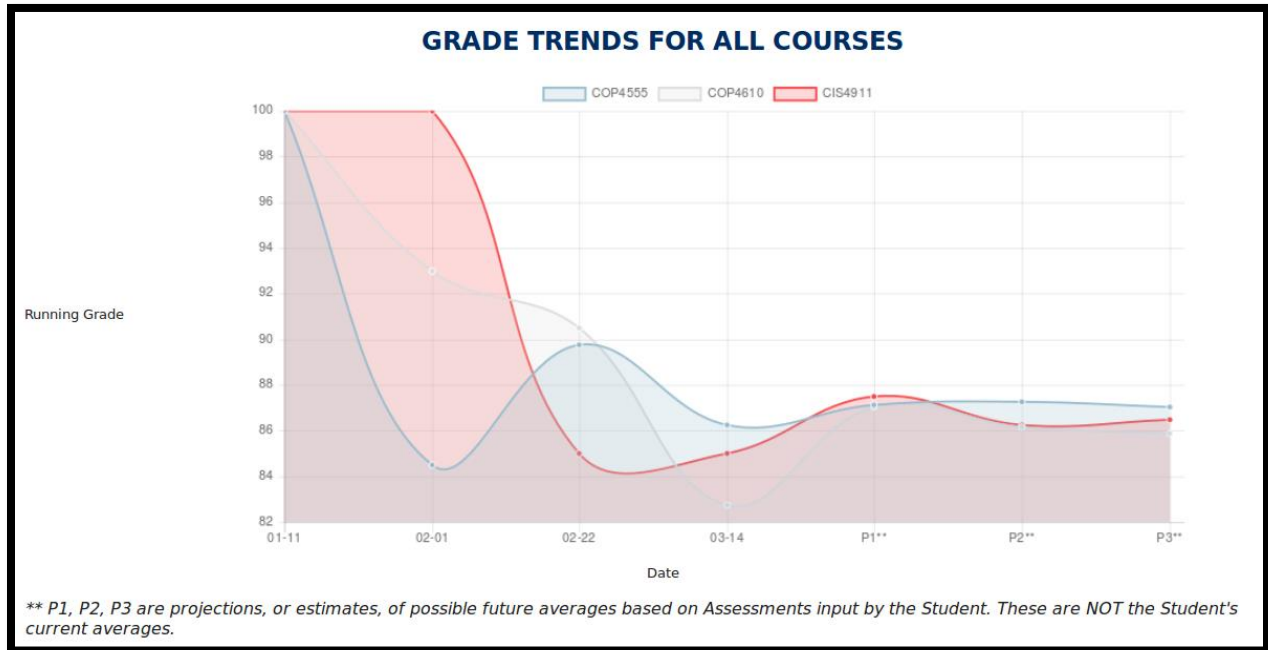


Figure B-11 Semester Forecast Report Graph

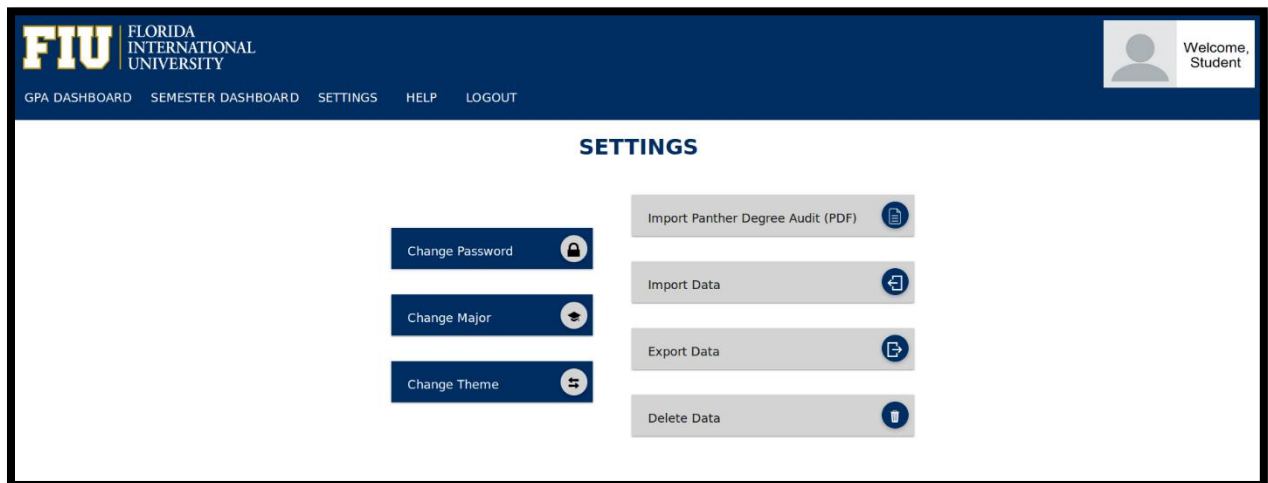


Figure B-12 Settings

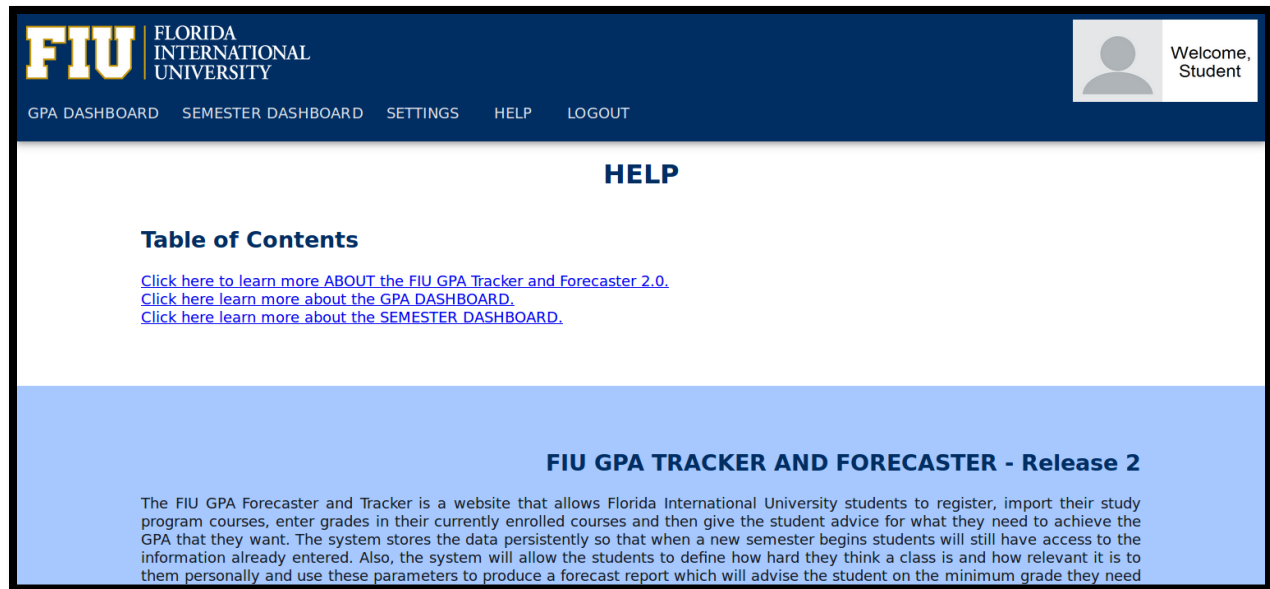


Figure B-13 Help / About

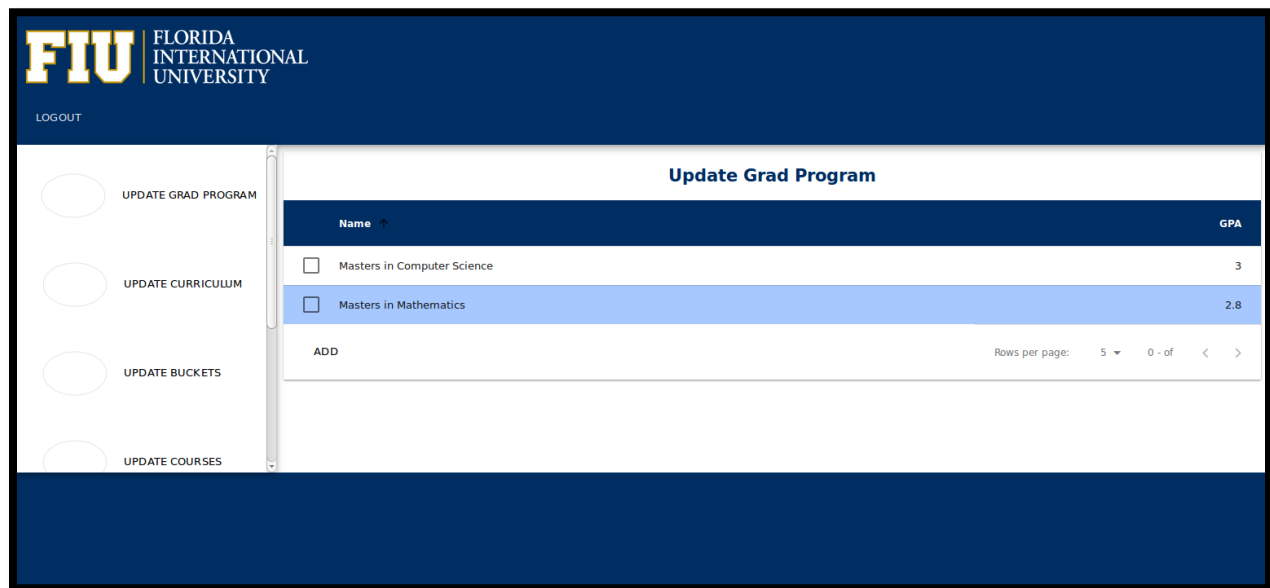


Figure B-14 Admin Page

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.

Date: 9/23/2016

Attendees: Camila Perez, Sebastian Buccollini, Johann Henao

Start time: 8:40 PM

End time: 9:00 PM

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

- User Story #927 Do full regression test of GPA Dashboard and fix bugs encountered

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 #929 Do full regression test of Semester Dashboard and fix bugs
- User Story #928 Fix and fully test log feature

Date: 10/10/2016

Attendees: Camila Perez, Johann Henao

Start time: 8:40 PM

End time: 9:30 PM

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

- User Story #930 Migrate Settings page to AngularJS

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 User Story #929 Do full regression test of Semester Dashboard and fix bugs

Date: 10/25/2016

Attendees: Camila Perez, Johann Henao

Start time: 8:40 PM

End time: 9:30 PM

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

- User Story #943 Fix Import Data functionality
- User Story #939 Understand and Document PHP Unit Tests for GPA Dashboard

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 #931 Create test drivers for Major Programs Import and GPA Import
- User Story #929 Do full regression test of Semester Dashboard and fix bugs

Date: 11/8/2016

Attendees: Camila Perez, Johann Henao

Start time: 8:40 PM

End time: 9:30 PM

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

- User Story #931 Create test drivers for Major Programs Import and GPA Import

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 #929 Do full regression test of Semester Dashboard and fix bugs

Date: 11/22/2016

Attendees: Camila Perez, Johann Henao

Start time: 8:40 PM

End time: 9:30 PM

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

- User Story #956 Defect: Fix issues about basic functionality of test drivers.
- User Story #960 Integrate test drivers XML files with Curriculum and GPA Audit import.

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document

User Manual:

1) Registration: To register for the FIU GPA Tracker and Forecaster simply click on the "Register" navigation link in the landing page. There it will ask you for your email and name and ask you to set your username and password. Password must be at least eight characters long.

2) Login: Login is just as simple click the login navigation button and enter your username and password.

3) Import PDF Grade Report: The first step any user should take to facilitate their experience is importing their Grade report. This PDF can be downloaded from PantherSoft under the "Panther Degree Audit" by hitting "Download PDF". After downloading the GPA go to settings, click the button next to the "import GPA Audit (PDF)" label, and select the Grade Report PDF. Just like that, all of your grades and major will be imported.

4) The overall GPA dashboard will tell you what your current GPA is. You can "Unenroll" in classes there (mark courses as not currently taking) by hitting the right arrow button on the courses taken table. Conversely, you can "enroll" in a course you are currently taking. If for whatever reason there is a discrepancy between the grade you have and the grade shown, it can be modified. More importantly, the Overall GPA Dashboard can provide a forecast report which tells you what you need to earn to get into your desired graduate program.

5) The Semester Dashboard is the perfect place to keep track of all the courses you are currently taking. For each course you are taking you can describe the grade breakdown of your class (i.e. Homework - 30%, Quizzes - 20%, etc.) and when you enter grades for each of these categories your grade will instantly be calculated.

Installation Document:***Setup**

```
sudo apt-get update
sudo apt-get install virtualbox-guest-dkms -y
```

***Enable repositories**

```
sudo add-apt-repository "deb http://archive.ubuntu.com/ubuntu $(lsb_release -sc) main universe
restricted multiverse"
sudo apt-get build-dep apache2
sudo apt-get install libxml2-dev
sudo apt-get install autoconf
```

***Download and install MySQL**

```
sudo apt-get install mysql-server
pass: sqliscool
```

***Create database**

```
*login to MySql → mysql -u root -p
CREATE DATABASE GPA_Tracker;
```

***Create users**

```
CREATE USER 'sec_user'@'localhost' IDENTIFIED BY 'Uzg82t=u%#bNgPJw';
GRANT CREATE, DROP, DELETE, INSERT, SELECT, UPDATE ON GPA_Tracker.* TO
'sec_user'@'localhost';
CREATE USER 'admin'@'localhost' IDENTIFIED BY '12345678';
GRANT ALL ON GPA_Tracker.* TO 'admin'@'localhost';
```

*** Install git**

```
sudo apt-get install git
```

*** Setup git**

```
git config --global user.name "Johann Henao"
git config --global user.email "jhena005@fiu.edu"
```

*** Clone github repository***** Create directory**

```
mkdir /home/sproject
cd /home/sproject
git clone https://github.com/FIU-SCIS-Senior-Projects/GPA2.git
cd GPA2
git checkout development
```

*** Import database - located in /home/sproject/GPA2/Code/Database**

the unzip password is: sqliscool

to import the db:

```
mysql -u root -p GPA_Tracker < DB.sql
```

Download to user's home folder and Install apache

```
sudo apt-get update
sudo apt-get install libpcre3 libpcre3-dev
```

```
cd /home/userfolder/Downloads
wget http://archive.apache.org/dist/httpd/httpd-2.4.18.tar.gz
unzip: $ tar -xzf httpd-2.4.18.tar.gz
* Install apr, apr-util follow instructions at: http://httpd.apache.org/docs/2.4/install.html
cd into directory
sudo ./configure
sudo make
sudo make install
```

```
* From /home/userfolder/Downloads Download pthread library
wget http://pecl.php.net/get/pthreads-2.0.10.tgz
unzip: $ tar -xzf pthreads-2.0.10.tgz
```

```
* From /home/userfolder/Downloads Download and install php
wget -O php-5.6.18.tar.gz http://php.net/get/php-5.6.18.tar.gz/from/this/mirror
unzip: $ tar -xzf php-5.6.18.tar.gz
mv pthreads-2.0.10 php-5.6.18/ext/pthreads (from downloads folder)
cd into php-5.6.18
rm -rf aclocal.m4
rm -rf autom4te.cache/
./buildconf --force
./configure --help | grep pthreads
```

Ensure it shows enable-pthreads

```
./configure --with-apxs2=/usr/local/apache2/bin/apxs --with-mysql --with-mysqli --with-json --
enable-mbstring --enable-soap --enable-maintainer-zts --enable-pthreads --enable-sysvsem --
enable-sysvshm --enable-sysvmsg
make
sudo make install
```

Configure Apache

```
Edit /usr/local/apache2/conf/httpd.conf
<FilesMatch \.php$>
    SetHandler application/x-httpd-php
</FilesMatch>
ServerName localhost
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
DocumentRoot "/home/sproject/GPA2/Code/WebSite/fiugpatf"
<Directory "/home/sproject/GPA2/Code/WebSite/fiugpatf">
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Require all granted
</Directory>
<DirectoryMatch "/home/sproject/GPA2/Code/WebSite/fiugpatf">
php_admin_value auto_prepend_file
"/home/sproject/GPA2/Code/WebSite/fiugpatf/common_files/toLog.php"
```


</DirectoryMatch>

* Configure php

Copy php.ini-development from php-5.8.18 to /usr/local/lib as php.ini

Edit php.ini

date.timezone = 'America/New_York'

change: session.cache_limiter = public

uncomment: session.save_path = "/tmp"

uncomment: always_populate_raw_post_data = -1

* Install Python

sudo apt-get install python-pip python-dev build-essential

sudo pip install --upgrade pip

sudo pip install --upgrade virtualenv

sudo pip install PyPDF2

Wishlist Document:

- Mobile App
 - Finish Android app and start IOS app.
 - Make it easier for students to insert grades and to check current grades.
- Update Help page look and feel.

REFERENCES

...