Florida International University
School of Computing and Information Sciences

CIS 4911 - Senior Capstone Project
Software Engineering Focus

Final Deliverable

FIU GPA Forecaster and Tracker
Team # 5

Team Members
Juan Machado
Ike Ukachi-Lois

Product Owner: Johann Henao
Instructor: Masoud Sadjadi

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained

within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

Table of Contents

Abstract

The FIU GPA Tracker and Forecaster is a website that assists students with keeping 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 get their goal GPA. In addition a small Android app has been created that allows students to view and enter their semester grades. The system follows the three tiered architecture that is often implemented in web based solutions.

INTRODUCTION

The FIU GPA Forecaster and Tracker is a website that allows for Florida International University students to register, input their courses, and the grades in their current courses and then give the student advice for what they need to achieve the GPA that they want. The main goal is to automate tedious tasks and get users motivated to do better. The system will store the data persistently so that when a new semester begins they do not need to submit their grades again. The system will allow the user to define how hard they think a class is and how relevant it is to them personally and use these parameters to advise the student on the grade they should get.

This document will begin with a discussion of the current system and a description of the new system that will show why the new system is beneficial. That will be followed by an overview of what was done by showing all the user stories that were implemented as well as those that were not. That will be followed by a discussion of how the project was managed including what software and hardware was used as well as how our time was scheduled. Following that will be a discussion of the system architecture and what design patterns were used. Finally, there will be a discussion of how system validation.

Current System

While the GPA for a student is calculated automatically by Panthersoft, if a student has a desired GPA and they want to know what grades they need they have to do that calculation manually. To do that they need to look at all their grades, add them up with their current classes and what they expect to get. While this system might be fine when the student has only taken few classes, it becomes a very tedious to manually add up all the grades when the student has taken dozens of courses. Besides the fact that it is very tedious it is prone to error, if the student’s finger slips and they hit a 4 instead of a 3 on the calculator the entire calculation becomes invalid.

Moodle is an excellent system in place that can help students track their semester grades. However, professors are not required to make use of Moodle, and even if they do they may be slow to updating grades. If the professor does not use Moodle at all then the student must keep track of their grades manually, this can be especially difficult if the student is not allowed to keep any graded assignment.

Purpose of New System

The purpose of the new system is to show student how they are doing, show them what they need to get their dream GPA, and motivate them. We do this by automating tedious tasks such as calculating grades and GPA and giving them a visible model of how their doing. By persistently storing the data, at the start of every new semester the student only needs to input the new courses they will be taking this semester. With that and the user's desired GPA the system can then advise the student on what they need to achieve their goals. Furthermore the system will help track their academics as the semester continues if the user inputs their grades as they get them. This feature was included to keep students motivated throughout the whole year. If they do the system can say what their current average is and what they need to get on certain assessments to get the grade they want.

USER STORIES

This section will provide an overview of each of the user stories that were planned for this semester. This includes the ones that were successfully implemented as well as some that were uncompleted. The ones that were completed had a higher priority than the ones that were unfinished. The uncompleted list can be seen as a wish list of sorts, they would be nice to have but only if time permits.

Implemented User Stories

User Story # 669 - Design front page with html/css
Should mimic FIUs front page

User Story # 666 - Complete HTML/CSS tutorial at codecademy
Neither of us have experience with web programming so we have to learn before starting serious work.

User Story # 667 - Complete php tutorial at codecademy
Since neither of us have experience with web programming we need to learn before starting serious production.

User Story # 668 - Design Database
Create ER Diagram. Design tables to be used in the database.

User Story # 670 - Create login page/register page with html/css
Register page should have entry fields for email, username, first name, major, password, confirm password, and date declared. Login page should have username and password field. Also a forgot password link. Make sure that login fields are not vulnerable to SQL injection.

User Story # 679 -Design Login Page
Login page should have a look and feel like that of FIU’s blackboard login. Which includes a designated input box for the user to sign in with their username and password

User Story # 680 - Setup development environment
As a developer I would like to set up the development environment to code in the necessary languages.

User Story # 681 - Login Functionality
As a developer I would like to allow users in the database to be able to login with their username and corresponding password. Login will eventually lead the user to the Overall GPA dashboard, allowing them to view or import their Grades prior to the current semester.

User Story # 682 - Complete JQuery Tutorial
As a developer I would like to learn jQuery so I can create more dynamic pages.

User Story # 683 - Design Overall GPA Dashboard
Create a design for the dashboard that is streamlined and consistent with the design of the Current semester dashboard. The system shall allow students to enter list of courses previously taken with their corresponding letter grade along with future courses to be taken. A List of courses should come from a possible file, imported to the database and then presented to the user as a read only select table. The Dashboard provides the user with an easy to navigate interface that is similar to the mock up page created with balsamiq.

User Story # 698 - Website Mockup Pages
As a web developer I would like to create a mockup of the website that shows what the flow of admin and student users will look like from page to page. The mockups provide a clear visual description of the flow of the site

User Story # 705 - Allow administrator to manually add a course to the CourseData table
As a web developer I want to provide functionality that allows an administrator to insert a course into CourseData and indicate what major it is necessary for.

User Story #706 - Overall Dashboard and related pages Data Tables
As a web developer I plan to create data tables for the overall GPA Dashboard which will be linked to the database through the tables “Overall Grades”, “Courses Needed”, as well as the dropdown box for target graduate program. The data tables will have a “row” slide down functionality which will display the buttons required for add, modify, delete data as required.

The add and modify features will provide the user a slick and intuitive form to add/update information.

User Story#709 - Add central page to semester dashboard
As a Web Developer I would like to create a page that acts as a central page when entering the current semester dashboard.

User Story #710 - Current Semester dashboard and related pages Data Tables
As a web developer I plan to create data tables for the student “current Semester” Dashboard and related pages which will be linked to the database through the jquery ajax interactions. The data tables will have a “row” slide down functionality which will display the buttons required for add, modify, delete data as required. The add and modify features will provide the user a slick and intuitive form to add/update information.

User Story # 708 - Create script/php that detects if user is logged in and changes Navigation Bar
Detects if a user is logged in and if they are replace the register/login/about navigation bar with links to the dashboards and logout.

User Story #728 - Setup referential integrity constrains on Overall GPA dashboard tables
As a web developer, I would like to include referential constraints on the database tables such that any changes that occur in one table will affect the necessary associated tables. This constraint will keep the datatables on the Overall GPA dashboard relevant.

User Story #729- SETUP REFERENTIAL INTEGRITY CONSTRAINS ON CURRENT SEMESTER COURSES DASHBOARD TABLES
As a web developer I would like to set up referential constraints on the database so that we don’t have stale data if a file on a different database that is related is removed. Instead after the constraints are defined, it should cascade on delete to remove the stale data on other tables.

User Story #730 Student data import/export/delete functionality
As a web developer I would like to provide for each student the option to export their data on their personal computer as an XML file as backup. Furthermore, they should have the option to recover their data with that XML backup but they can only import to rows with their own username in it.

User Story #731 Delete/Modify Function For the Master Course List
As a web developer, I would like to give administrators, the ability to Modify a course that is on the master list. This would include the ability to change the course name and or the number of credits for a specified course.
In addition, the ability for deleting a course must be developed. If there is a need for an administrator to delete a course from the database, they must be able to have that functionality available

User Story #732 - Synchronize Databases
As web developers, we would like to have the database structure, as well as some of its contents such as users and courses, synchronized so that we have the ability to seamlessly combine our code when transitioning to the FIU development environment.

User Story #733 - Current Semester GUI improvements Set 1
On Current Courses page:

1. Improve overall Fonts
2. Fix “First Previous Next Last” overlapping buttons.
3. Make page more appealing, (colors, add a help feature, get creative here...)
4. Buttons need to be the same size (with and height) and there should be a small space between them.
5. Change “Add/Change Breakdown” to “Assessment Breakdown”
6. Add the current grade avg for each course listed as a row field value.
7. Change “Remove” to “Remove Course

User Story #734 - Current Semester GUI improvements Set 2

1. On Assessments Page:
    1. Change “Add Assessment” to “Assessment Management”
    2. Remove “Assessment Type, Percentage” fields, instead when “Add Assessment” button is clicked it should display a slick looking input form.
    3. Show list of assessments in a data table with a current grade avg for each assessment and add a remove, modify button on slide down.
2. On Assessment Tabs:
    1. Change “Remove bucket” to “Remove Assessment”
    2. Remove add grade field and instead present the user with a slick input form when “add grade” button is clicked.

3. Choose a nice location in this form to show the current grade avg for the assessment.

User Story #735 - Design Program Curriculum Database Structure
Elaborate an Object diagram of all the tables and referential constrains necessary to model program curriculum requirements for any FIU Major. This will be the primary data source that will be used to come up with the data view that will be presented to the students in the GPA Overall Dashboard and Current Semester Dashboard.

User Story #736 - Overall GPA Dashboard improvements/adjustments

1. Screen Privacy functionality: When turned on All rows should say “Screen Privacy”. Double Check that html source code does not contain the grades.
2. Look and feel of data table: give me three options. The data tables have good functionality but look a bit alien on the edges. Look for options to improve look and feel.
3. Improve size of fonts and position of text. Make it more appealing, interesting. Get creative here.
4. Movement of courses from Courses Required to Courses needed should be more intuitive with selection of row and pressing of an arrow between the two tables. Arrow should have a slick look and feel that fits well with the theme of the page.
5. Make size of data table larger, so that it can accommodate at least 5 rows + one slide down. So no expansion of datatable should occur with one slide down.
6. Improve “TARGET GRADUATE PROGRAM” Drop down list. Make font and box much more appealing. Look for ideas on the web. Same for Required GPA and GPA.
7. Improve “WELCOME STUDENT” make it more appealing.
8. Improve modify Grade input form. The input field and the label should be on at the same level not one below the other. Leave a sufficient space for between this and the submit button.
9. Current GPA should be a nice cool, modern green color if GPA target is met, Yellow if not met, and white if not met.
10. Think of what else we can add to this dashboard to make it really interesting but not to cluttered either.

User Story #737 - Defect: Overall GPA Dashboard HTML elements out of alignment with page zoom
As a developer, I would like to fix the issue with the positioning of html elements when the screen resolution is adjusted.

User Story #757 - Leverage "what if" feature to automate curriculum data import.
As a web developer I would like to automate the process of importing grades and curriculum requirements so to make inserting data more practical. This can be done by using Panthersoft's "what if" feature which shows both what the student has taken and what they need to take.

User Story #767 - Help Feature for GPA Dashboard
As a developer, I would like to provide the user with information to aid them when navigating the features available to them on each dashboard. The information will be displayed in a slideshow and presented as a tutorial that explains all functions.

User Story #768 -Help Feature for Semester Dashboard.
As a web developer I would like to add two extra pages to the help feature of the semester dashboard to match the look and feel of the overall dashboard help feature.

User Story #769 -Refactor code of GPA dashboard to work with new DB design.
With the new database design, the information for the courses each student needs to take will be retrieved differently. In addition, the weight and relevance must be obtained differently also. Specifically, the course a student requires, will have to be obtained through the major they have selected. Through the major, the specific buckets related to that major will be selected and displayed appropriately on the GPA Dashboard.

User Story #770 - Refactor code of Semester dashboard to work with new DB design.
As a web developer I want the website to be able to work with the newly redesigned database so that the website can benefit from the improvements.

User Story #782 - Manual entry of curriculum requirements
As a web developer I would like to manually create two xml files that represent the curriculum requirements of the Computer Science and electrical engineering as well as the back end code that takes it and uploads it to the database.

User Story #783 - Simulated Run of data Import
As web developers, we would like to simulate the process of an admin importing a major's curriculum.
After a major has been setup within the database (After the admin import) the student will be able to import their GPA audit report which will populate the database with information related to their uploaded file.

User Story # 784 - Setup Web Service Development Environment POC
As Product Owner/Mentor I would like to setup a fully functional development environment to implement and deploy two web services. One webservice will be a simple HelloWorld and the other will perform an interaction with the GPA_Tracker and fetch the list of users. The development environment chosen will be running Tomcat,Axis2 and Eclipse.

User Story # 696 Login Security
As a developer I would like to provide users with the ability to securely log in without the fear of compromising their password. This includes regulation of sessions with session ID's and hashed and salted passwords. SQL injection must also be implemented.
The login security will add to the User friendliness of the website allowing students to feel comfortable about their session.

User Story # 785 - Setup Android Development environment and Create an Android App (POC) that consumes the Tomcat Web services from Story 784.
As a Product Owner/Mentor, I would like to come up with a proof of concept Android Application that consumes the Webservices implemented for story 784. First I will need to setup the development environment. The IDE of choice will be Android Studio.

User Story # 788 - Semester Dashboard GPA Trend Graph
As a web developer I would like to show a graph that shows how the grades of the student changes as the semester goes on.

Pending User Stories

User Story # 691 - Administrator data import/export Functionality
As a web developer I would like to give the administrator the ability to import, export, and delete the data which includes student roster, curriculum requirements and course taken information, current semester break down.
In addition to the data, the option to import export and delete schema should also be a feature for the administrator. This encompasses the structure of the database which includes all tables.

User Story#692-EPIC: Implement security of the Dashboard: Enrolled Courses Performance Management
As a user of the website I would like the website to be secure so that hacker cannot have access to my personal data.

User Story#693-EPIC: Overall GPA outlook dashboard
This dashboard is a center of focus of the site that will provide the student an intuitive overview of his GPA standing in relation to his goals and postgraduate academic interests.

User Story#695-EPIC: Implement security of the Dashboard: Enrolled Courses Performance Management
As a student I would like to input the classes I am currently taking along with grade breakdown (buckets) so that the system can tell provide me with forecast suggestions that tell me what I need to get on certain assignments.

User Story#710-Current Semester dashboard and related pages Data Tables
As a web developer I plan to create data tables for the student “current Semester” Dashboard and related pages which will be linked to the database through the jquery ajax interactions. The datatables will have a “row” slide down functionality which will display the buttons required for add, modify, delete data as required. The add and modify features will provide the user a slick and intuitive form to add/update information.

User Story #735 - Design Program Curriculum Database Structure (W/ associated tasks)

Elaborate an Object diagram of all the tables and referential constrains necessary to model program curriculum requirements for any FIU Major. This will be the primary data source that will be used to come up with the data view that will be presented to the students in the GPA Overall Dashboard and Current Semester Dashboard.

Associated Tasks -

Task 758 - Redo code for the Courses Needed data retrieval

With the new database design, the information for the courses each student need to take will be retrieved differently. In addition, the weight and relevance must be obtained differently also. Specifically, the course a student requires, will have to be obtained through the major they have selected. Through the major, the specific buckets related to that major will be selected and displayed appropriately on the GPA Dashboard. The Courses Needed Table display will have to change in order to accommodate the introduction of concentrations and electives.

Task 759 - Redo Courses Needed Design to accommodate for the major courses buckets

The Courses a student needs to take for a specific major, is split up even further into sub categories we call buckets. The buckets will be distinct groups of courses such as UCC requirements, Core classes and electives. The table on the GPA Dashboard must be able to display all possible buckets in an intuitive way to allow the student to easily interact with the data.

PROJECT PLAN

For this project the scrum development method was used. In scrum, development is broken up into sprints. For this project, each sprint was a total of two weeks long. As such stories were assigned for each sprint and the goal was to finish them within a sprint. The nature of the two week sprint can be seen in the Gantt chart in the Appendix E. We had the following roles during the project:

- Juan Machado: Developer, Scrum Master.
- Ike Ukachi-Lois: Developer
- Johann Henao: Product Owner

Hardware and Software Resources

In terms of software, all development was done on an Ubuntu 14.04. Ubuntu was chosen as the development environment because Linux platforms are well known to be very stable. For client side code we used HTML, CSS, Javascript, and JQuery. HTML and CSS are the standard for creating web page layout. Javascript and JQuery are also used extensively and supported by every major browser. PHP was used for most server side code. PHP is ideal because it has great support for session variables, post variables, and it interacts with mySQL excellently. Python was used for the sole purpose of parsing the Grade Audit. Python was chosen because it can use the excellent and easy to use PyPDF parser. Apache was used as the web server due to its ease of use and stability.

The Android app was developed in Android Studio using Java. Java is the standard for coding Android application. The web services were created using Java in the Eclipse IDE. We used the Eclipse IDE because it is one of the most popular Java IDE which can facilities exporting the code as a WAR file. Java was used because everyone on the team is familiar with Java.

Coding Naming Conventions and Standards

The Database Uses camel notation to describe Table names and fields within them.

- Table student_course would be written as StudentCourse
- field course_info_id would be written as courseInfoID (acronyms and all other initializations are to be written in all uppercase).

1. **Camel naming convention.**
   - Functions and table as: Word1Word2.
   - Fields and variables as word1Word2.

2. **Indentation.**
   - 3 spaces per indentation

3. **Function names should be in the form of actions.**
   - Ex. GetName, ExportData, ImportData.

4. **Line Length.**
   - The length of a line should not exceed the width of the screen and should returns to the next line if it happens to be the case.

5. **Function Length.**
   - If possible, a function should focus on one activity and able to be read completely on a single page.

6. **Comments.**
   - Add comments on things that are NOT obvious, especially if it is complex.

7. **Refactor.**
   - Refactor soon and refactor often for improving code readability.

Sprints Plan

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

Sprint 1

(08/31/2015 - 09/11/2015)

User Story # - 676 Design Landing Page with HTML/CSS

User Story # - 668 Design Database

Related tasks

- 672 Create Student_info table

User Story # - 660 Setup development environment

User Story # - 670 Create login page and register page

Modeling

Sequence Diagrams: User Login and Register

Sprint 2

(08/31/2015 - 09/11/2015)

User Story # - 681 Basic Functionality of Login

Acceptance Criteria:

1. The registered user is able to go log in through the login page with their username and password.

User Story # - 683 Design dashboard for overall gpa

Acceptance Criteria:

1. The Dashboard provides the user with an easy to navigate interface that is similar to the mock up page created with balsamiq

User Story # - 705 Allow administrator to manually add a course to the course_data table

Acceptance Criteria:

1. There must be an html/css page that looks reasonably nice.
2. There must be a form where the administrator can add course ID, Course Name, and number of credits
3. After the user submits the form the course is added.

User Story # - 671 Create back end code to take data from data table and store it in the database

Tasks

- 699 Create/HTML/CSS page for user to input courses enrolled
- 700 Create Data table in the insert classes page

User Story # - 708 create script/php that changes navigation bar

Modeling

Sequence Diagrams: User Login register, and add course

Sprint 3

(09/28/2015 - 10/09/2015)

User Story # - 706 Overall Dashboard and related datatables

Acceptance Criteria:

1. The overall grades table will contain data for the course name, the credits and grade received
2. The courses needed table will contain data for the course name, the credits and relevance of the course (decided by the user)
3. The Drop down box will contain data for the name of the graduate program and the gpa required to enter the program.

4. Datatable rows provide a slide down feature and display buttons for add/modify and delete.
5. Modify and Add functions provide a slick/intuitive form that allows the data input.
6. Form data is validated using up to date standard techniques.

Related Tasks:

- 727 Student roster page
- 716 Screen privacy
- 715 GPA Calculation
- 714 Modify Course
- 713 Datatable Delete Course
- 712 Datatable Add course
- 711 Datatable display

User Story # - 710 Current semester dashboard and data tables

Acceptance Criteria:

1. The student "current Semester" dashboard and related pages show the datatables with a slick look and feel.
2. All datatables on the student "current Semester" dashboard and related pages display proper student information as per Functional requirements specifications and use case specifications.
3. Datatable rows provide a slide down feature and display buttons for add/modify and delete.
4. Modify and Add functions provide a slick/intuitive form that allows the data input.
5. Form data is validated using up to date standard techniques.

User Story # - 709 Add central page to semester dashboard

Acceptance Criteria:

1. The page should show all the classes that the student is currently enrolled in.
2. The page should have link to all additional features provided in the current semester dashboard.
3. The page should be good looking and intuitive to use.

Modeling

Sequence Diagrams: Add course, Delete Course, Modify Course, Add Course Grade

Sprint 4

(10/12/2015 - 10/23/2015)

User Story # - 732 Synchronize Databases

Acceptance Criteria:

1. The team developers should be able to have code that works on any environment using the same database structure.
2. Any files that query the database through MySQL queries should not need altering in order to execute successfully.

User Story # - 733 Current Semester GUI Improvements Set 1

Acceptance Criteria:

1. Each directive listed above that has an objective end is met.
2. The product owner is satisfied with the look and feel of page as a whole.

User Story # - 733 Current Semester GUI Improvements Set 2

Acceptance Criteria:

1. Every improvement above that has an objective requirement is met.
2. The product owner is pleased with the look and feel of the page.

User Story # - 736 Overall GPA Dashboard Improvements

Acceptance Criteria:

1. The functions for add modify and delete work flawlessly even after the implementation of the arrow button to add courses taken.
2. The aesthetics are more appealing and make the site feel more personable. (Fonts, borders and positioning)

Related Tasks

- 756 Help feature
- 755 Data Table Fixed Sizing
- 741 Arrow button
- 740 Screen Privacy Functionality

Modeling

Sequence Diagrams: Add Course, Modify Course, Delete Course

Sprint 5

(10/26/2015 - 11/6/2015)

User Story # - 729 Setup referential constraints on the semester Courses Dashboard

Acceptance Criteria:

1. An object diagram is created that describes the database. The diagram should go through revisions as everyone gives their opinion on the structure of the database.
2. Using the innoDB engine the constraints that were discussed using the diagrams should be implemented.

Related tasks

- 766 Discuss with team the best way to show curriculum requirements

User Story # - 728 Setup referential constraints on the GPA Dashboard tables

Acceptance Criteria:

1. The integrity constraints should allow the data displayed on the Overall GPA Dashboard tables, to be synchronized and properly up to date with the database information
2. Creation of an object diagram will describes the relationships between tables in the database. The diagram should go through revisions as everyone gives their opinion on the structure of the database.
3. Using the innoDB engine the constraints that were discussed using the diagrams should be implemented.

User Story # - 731 Delete/Modify Function For Master Course List

Acceptance Criteria:

1. The admins are able to modify a course ID, name and or credits for a course and have the changes reflect the in the database.
2. After a course has been deleted, the changes will be reflected in the database and each student and major associated with the course will also see the change.

User Story # - 767 Help Feature for GPA Dashboard

Acceptance Criteria:

1. The User has the ability to easily look up how to utilize any function offered an the dashboard.
2. The User has access to descriptions of the dashboard as well as an about page that describes the purpose of the FIU GPA Tracker and Forecaster website.

Modeling

Sequence Diagrams: Add Current Course, modify course, delete course

Sprint 6

(11/9/2015 - 11/20/2015)

User Story # - 757 Leverage “what if” feature to automate curriculum data import.

Acceptance Criteria:

1. The system can take as input an html file that contains the contents of what a student has taken and what they need and inputs it into the database.
2. For an administrator they should be able input the html file of a what if html page and have the system automatically import the requirements for the major that the what if report was about.
3. A help page is available for guiding the user in what to do.

Related tasks:

- 777 Create Python script to parse PDF grade audit
- 765 Create back end code to parse file output from python script
- 764 Create GUI for user to upload what if html page
- 763 Create Python script to parse html of what if page

User Story # - 735 Design Program Curriculum Database Structure

Acceptance Criteria:

1. Using a combination of referential constraints and innoDB storage engine, a database structure that is well connected to associated tables is expected.
2. Factors such as major, the start and end date of the major, specific student exceptions will all be taken into consideration when the structure is implemented.
3. Must take into account electives, concentrations and the possibility of different majors having a different breakdown of the needed electives and concentration.

Related Tasks

- 759 Redesign Courses Needed to accommodate for the Major Courses Buckets.
- 758 Redo code for the Courses Needed data retrieval

User Story # - 770 Refactor code of Semester dashboard to work with the new DB design

Acceptance Criteria:

1. In every single PHP every SQL query needs to be refactored so that it access the tables from the new database.

Related Tasks:

- 770 Refactor each PHP file

User Story # - 769 Refactor code of GPA dashboard to work with new DB design

Acceptance Criteria:

1. The same Functionality should be working the same.
2. Any functionality such as Screen Privacy and GPA calculation should be adjusted to fit the new design

Related Tasks

- 781 Implementation of handling categories with child buckets
- 776 Addition of unknown courses (Natural Science Issue)
- 775 Selection of electives for forecast report

User Story # - 769 Manual entry of curriculum requirements

Acceptance Criteria:

1. The XML files should have all the information that is pertinent to the CS and EE majors respectively.
2. The back end code should accept and be able to input all the information into the database, replacing any identical entries.

User Story # - 730 Student Import/export/delete functionality

Acceptance Criteria:

1. The user should be able to make a backup of all information pertinent to them as an sql file and download it to their personal workstation.
2. The user can upload data as an sql file.
3. The user can delete their personal course information.
4. All data interaction is guaranteed to be in the context of the logged in user only. The system will have strict check measures so that import/export/delete of data for other students is not accessible by any means by other users

Related Tasks:

- 780 Add delete feature
- 762 Create back end code to parse xml sql file
- 761 Add GUI features to allow user to upload file
- 760 Learn how to use PHP XML parser

User Story # - 768 Help feature for Semester Dashboard

Acceptance Criteria:

1. There should be a page that links to the current help feature as well as an about feature.
2. Another page should be added roughly describing what the semester dashboard does.

Related Tasks:

- 778 Change help feature to resemble Overall GPA dashboard.

Modeling

Sequence Diagrams: Add Student Course, Modify Student Course, Delete Student Course, Delete Data, Import Grades, Import Data, Export Data, Modify Course Grades

Object Diagram

Sprint 7

(11/23/2015 - 12/04/2015)

User Story # - 771 Dynamically represent Student's Curriculum requirements in GPA Dashboard

Acceptance Criteria:

1. The interface will represent the curriculum requirements of a student's major.
2. The buckets listed on a specific table, whether the courses needed or the courses taken) will correspond with the bucket of the same name on the other table

User Story # - 738 Overall Dashboard Forecast Functionality/ Report

Acceptance Criteria:

1. The user has selected/met all requirements needed to generate the forecast of the remaining courses needed.
2. The weight and relevance has been adjusted accordingly (No zero's entered).

User Story # - 784 setup web Service Development Environment POC

Acceptance Criteria:

1. Development environment is fully operational with Eclipse as the IDE.
2. Tomcat server is operational.
3. Webservices are deployed to Tomcat and available over the internet.

User Story # - 785 Setup Android Development environment and Create an Android App (POC) that consumes the Tomcat Web services from Story 784

Acceptance Criteria:

1. Development environment is fully operational using Android Studio.
2. An Android application is implemented that consumes the HelloWorld and Database interaction Webservices from Story 784.

User Story # - 786 Setup development environment for Web Services and Android Development

Acceptance Criteria:

1. Development Environment with Tomcat, Axis2, Eclipse and Android Studio is fully operational.

Related tasks:

- 789 Download and test development setup for android and web services

User Story # - 787 Setup development environment and create WebServices and Android Development

Acceptance Criteria:

1. Development Environment with Tomcat, Axis2, Eclipse and Android Studio is fully operational.
2. The user has properly downloaded the app, and has registered to FIU GPA tracker and forecaster website.
3. WebServices provide services for the app users

User Story # - 788 Semester Dashboard GPA Trend Graph

Acceptance Criteria:

1. There should be an overall semester dashboard graph that tracks how grades have changed over time for all courses.

2. There should be individual graphs for each course when going into assessment breakdown.

Related Tasks:

- 791 Create JS code that takes plot points and graphs them
- 790 Create back end code to allow compute plot points for graph

User Story # - 777 Create an Android Mobile App to allow students to interact with Current Semester Grades

Acceptance Criteria:

1. The interface should look relatively nice.
2. The app should be able to connect to the web services created by Ike to get student information.
3. The user should be able to add and remove grades.

Related tasks

- 793 Create an android app that can connect to web services to alter current semester grades.

User Story # - 696 Login Security

Acceptance Criteria:

1. Passwords hashed and salted
2. SQL injection must be utilized

3. The user must be securely logged into the correct session.

User Story # - 739 Overall GPA performance chart

Acceptance Criteria:

1. The GPA history chart will display a student's GPA for each prior semester through a line graph. The data goes as far back as their initial enrollment to FIU.
2. The student will have the capability to hide this information like the grades and calculated GPA.

User Story # - 783 Simulated Run of data Import (Juan is joint owner)

Acceptance Criteria:

1. Admin imports the Major Curriculum requirements from (simulated) parsed xml file.
2. A student registers and imports their GPA Audit report. The import will instantiate the StudentCourses Table (only the all required ones) according to the Major imported from the GPA Audit pdf Report. Grades are updated accordingly and courses added (the electives and natural sciences etc.).

Related Tasks:

- 792 Extend function of PDF parser to allow it to pick up major buckets

Modeling

Sequence Diagrams: Modify Student Course, Delete Student Course, Delete Data, ImportGrades, Import Data, Export Data, Modify Course Grades, Add Course, GPA History Graph

Object Diagram

SYSTEM DESIGN

The following section will provide high level details about how the system was designed. It will begin by discussing the architectural pattern used by the system. Then how the system was broken down into different subsystems will be discussed. The different use cases that were implemented will then be shown in the use case diagram. The hardware and software requirements will be shown with the deployment diagram. Finally, the different system patterns will be shown. Each of these subsections will be provided with diagrams and explanations as to why these methods were shown.

Architectural Patterns

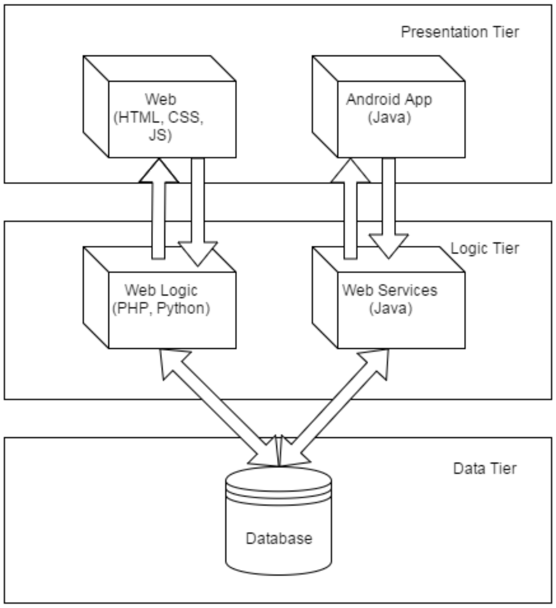


Figure # 1

For this project we chose to use a three-tier architecture. The reason why is because it lends itself naturally to the LAMP development environment we set up. The HTML and Javascript are client side and focus primarily with communicating with the server and displaying the logic that it receives for itself. The PHP scripts on the server side naturally lend themselves to the logic tier as they perform the bulk of the logic. As it happens, this system also worked well with the Android application as well. The Android app communicates easily with the Web Services.

Besides the fact that web development naturally lends itself well to this system architecture, it makes sense, security wise to create a barrier between the presentation tier and the data tier. For example an MVC architectural pattern would give the view subsystem access to the model subsystem.

System and Subsystem Decomposition

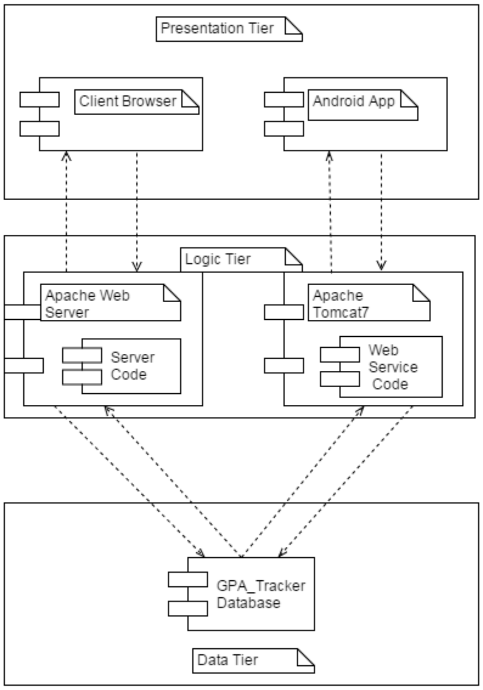


Figure #2

As can be seen in the figure above the main subsystems are the tiers themselves. The subsystem with the most dependencies is the data tier. The data tier is the main form of synchronization between the web solution and the Android solution. If a change in grades is made in the Andoid solution it will be visible in the website because of this and vice versa. The presentation tier has dependencies towards the logic tier because it does most of the computation for data the student enters and it acts as a middleman between it and the data tier.

Use Case Model

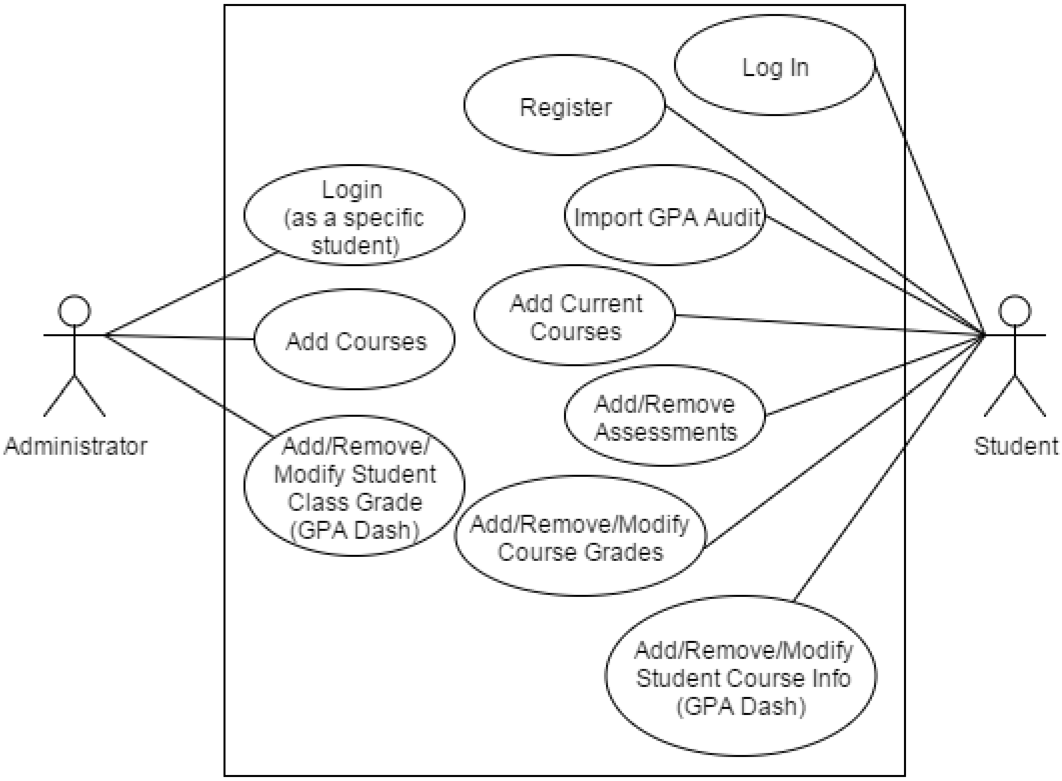


Figure # 3

The above diagram shows the use case models with every use case that has been implemented. There are two actors that interact with our system, an administrator and a student. An administrator can do anything a student can as well as the special use cases shown above. The student's use cases mostly relate to them managing the view of their grades in the Overall GPA and Semester dashboard along with some settings features.

Deployment Diagram

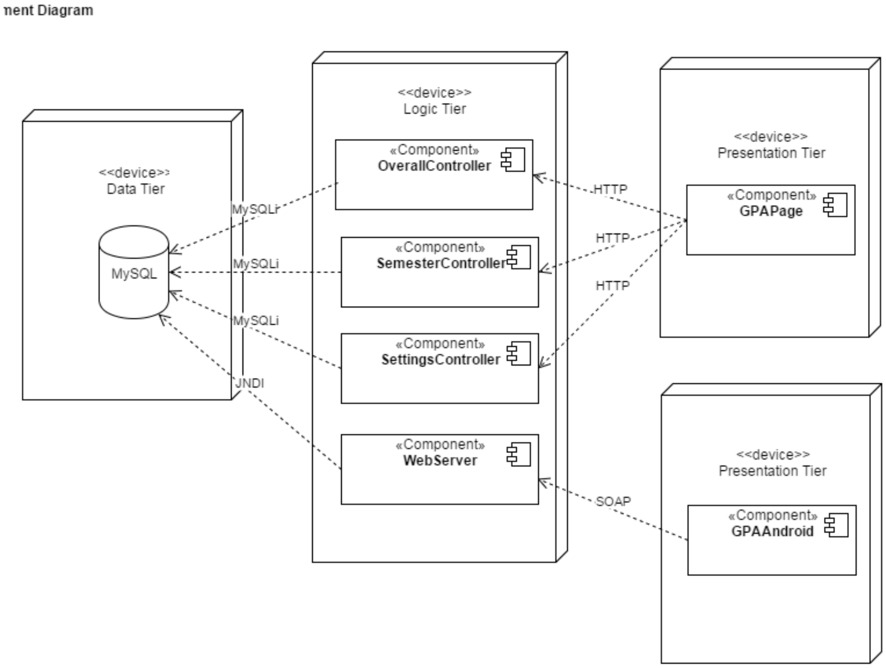


Figure #4

As stated previously, we are using the three tier architecture. This can clearly be seen in the deployment diagram above. The presentation tier is divided into two nodes. The first one represents users who are connecting using the Android application. This connects to the web services using SOAP. The second presentation node represents users who are connecting to the site using a browser, naturally this connection is made using HTTP. The logic tier is all hosted on the same development server. It contains the OverallController, the SemesterController, the SettingsController, and the WebServer. These the WebService component connects to the Data tier with a JNDI connector. The other components connect to the data tier with a mySQLi connection. The data tier is also hosted by the development server.

Design Patterns

On top of the three tier architecture, this web solution also takes advantage of the server-client pattern. Frequent ajax calls are made from the user's browser, which acts as the client, to the server. The server can receive multiple requests to process from the client. In this way the server acts like an active object with several multithreaded requests.

SYSTEM VALIDATION

System testing was done by creating a set of test cases for many user stories that involved having both rainy day and sunny day cases. Many user stories do not lend themselves well to test, however, such as the story for the team to learn web development. Besides selecting both sunny and rainy day scenarios, we tested for extreme values that could cause trouble.

User Story # 670 - Create login page/register page

System Tests

- Test ID - 001
- Purpose: To test if a regular user can successfully create an account.
- Test Setup: The website is open in the browser.
- Input: Email=bondJ@fiu.edu, Username=007, Password=12345678, Confirm Password=12345678, First Name=James, Last Name: Bonds.
- Expected Output: Congratulations page comes up, user is added to the database.

System Tests

- Test ID - 002
- Purpose: To test if the system will block a user with the same username.
- Test Setup: The website is open in the browser.
- Input: Email=bondWannabe@fiu.edu, Username=007, Password=12345678, Confirm Password=12345678, First Name=James, Last Name: Bonds.
- Expected Output: Error page shows up, nothing is added to the Database.

System Tests

- Test ID - 003
- Purpose: To test if the system will block a user with the same email.
- Test Setup: The website is open in the browser.
- Input: Email=bondJ@fiu.edu, Username=008, Password=12345678, Confirm Password=12345678, First Name=James, Last Name: Bonds.
- Expected Output: Error page shows up, nothing is added to the Database.

System Tests

- Test ID - 004
- Purpose: To test if user can login with the account they created.
- Test Setup: The person has registered with the site.
- Input: Username=007, Password=12345678
- Expected Output: The user is directed to the Overall Dashboard and the session variables are set.

System Tests

- Test ID - 005
- Purpose: To test if login is blocked when an incorrect username is input.
- Test Setup: The person has registered with the site.
- Input: Username=070, Password=12345678
- Expected Output: An error message appears indicating that login failed.

System Tests

- Test ID - 006
- Purpose: To test if login is blocked when a bad password is entered.
- Test Setup: The person has registered with the site.
- Input: Username=007, Password=1234
- Expected Output: An error message appears indicating that login failed.

User Story # 705 - Allow administrator to manually add a course to the CourseData table

System Tests

- Test ID - 007
- Purpose: To if admin can insert a course into the CourseInfo table.
- Test Setup: An admin is logged in.
- Input: CourseID=COP1000, Course Name=Introduction to PHP, Credits=3
- Expected Output: A success message is shown, the course is inserted into the DB.
-

System Tests

- Test ID - 008
- Purpose: To if system will block admin from inserting course where a non-numeric value is placed for number of credits.

- Test Setup: An admin is logged in.
- Input: CourseID=COP1000, Course Name=Introduction to PHP, Credits=three
- Expected Output: An error message is shown, nothing is added to the database.

System Tests

- Test ID - 009
- Purpose: To if system will block admin from inserting course where a negative value is placed for number of credits.
- Test Setup: An admin is logged in.
- Input: CourseID=COP1000, Course Name=Introduction to PHP, Credits=-1
- Expected Output: An error message is shown, nothing is added to the database.

User Story #706 - Overall Dashboard and related pages Data Tables

System Tests

- Test ID - 010
- Purpose: Testing if the modify grade button works.
- Test Setup: A user is logged in and in the overall dashboard.
- Input: The user types in a grade of B- for the new Grade.
- Expected Output: The row in StudentCourse with that course will have the grade column modified to B-, that change will be reflected in the dashboard.

System Tests

- Test ID - 011
- Purpose: Testing if system will block a user changing grade to an illegal value.
- Test Setup: A user is logged in and in the overall dashboard.
- Input: The user types in a grade of Q+ for the new Grade.
- Expected Output: An error message will appear and the database will be unaltered.

System Tests

- Test ID - 012
- Purpose: Test if deleteCourse does change a course to have a grade of ‘ND’ .
- Test Setup: A user is logged in and in the overall dashboard.
- Input: The user hits the arrow key from the courses taken table.
- Expected Output: The course will be change to have a grade of “ND”.

System Tests

- Test ID - 013
- Purpose: Test if addCourse method inputs the user's record correctly
- Test Setup: A user is logged in and in the overall dashboard.
- Input: The user hits the arrow key from the courses needed table.
- Expected Output: The course will be change to have a grade of "IP".

User Story#709 - Add central page to semester dashboard

System Tests

- Test ID - 014
- Purpose: Testing if the remove course functionality works
- Test Setup: A user is logged in and in the current courses page.
- Input: User clicks the row for the slide down, then hits delete.
- Expected Output: Course is removed from datatable and from StudentCourseTable in the database.

System Tests

- Test ID - 015
- Purpose: Testing if the assessment breakdown button works.
- Test Setup: A user is logged in and in the current courses page.
- Input: User clicks the row for the slide down, then hits assessment breakdown.
- Expected Output: User is directed to course assessment page for the course they clicked.

User Story #710 - Current Semester dashboard and related pages Data Tables

System Tests

- Test ID - 016
- Purpose: Testing if the add assessment breakdown button works.
- Test Setup: A user is logged in and in an empty course breakdown page.
- Input: Hits the add assessment button and types in "homework" for name and 30 for percentage.
- Expected Output: A new tab is created with the homework name. The homework assessment type is added to the database.

System Tests

- Test ID - 017
- Purpose: Testing if the system blocks adding two assessment types of the same value.
- Test Setup: A user is logged in and in a course breakdown page with a homework assessment.
- Input: Hits the add assessment button and types in “homework” for name and 30 for percentage.
- Expected Output: An error pops up and nothing is added to the database.

System Tests

- Test ID - 018
- Purpose: Testing if the remove assessment button works.
- Test Setup: A user is logged in and in a course breakdown page with at least on assessment type.
- Input: The user goes to an Assessment type tab and hits remove assessment.
- Expected Output: The tab is removed and the assessment type is removed from the database.

System Tests

- Test ID - 019
- Purpose: Testing if the add grade button works.
- Test Setup: A user is logged in and in a course breakdown page with at least on assessment type.
- Input: The user goes to an Assessment type tab and hits the add grade button and puts in a 85 for the grade.
- Expected Output: The datatable is updated with the grade and it is inserted into the Assessment table.

System Tests

- Test ID - 020
- Purpose: Testing if the system will prevent the user from adding an illegal grade.
- Test Setup: A user is logged in and in a course breakdown page with at least on assessment type.

- Input: The user goes to an Assessment type tab and hits the add grade button puts in a grade of “hello”.
- Expected Output: An error message appears and nothing is updated.

System Tests

- Test ID - 021
- Purpose: Testing if the delete grade button works.
- Test Setup: A user is logged in and in a course breakdown page with at least on assessment type.
- Input: The user goes to an Assessment type tab and hits the the grade to open the slide down, they then hit the delete grade button and a confirm dialog comes out, they hit ok for it.
- Expected Output: The grade is removed from the datatable and the database.

System Tests

- Test ID - 022
- Purpose: Testing if the modify button works.
- Test Setup: A user is logged in and in a course breakdown page with at least on assessment type.
- Input: The user goes to an Assessment type tab and hits the the grade to open the slide down, they then hit the modify grade button. Type in “Hello” for new grade.
- Expected Output: An error message pops up and nothing is changed.

System Tests

- Test ID - 023
- Purpose: Testing if the system prevents illegal values from being entered via the modify button.
- Test Setup: A user is logged in and in a course breakdown page with at least on assessment type.
- Input: The user goes to an Assessment type tab and hits the the grade to open the slide down, they then hit the modify grade button. Type in 76 for new grade.
- Expected Output: The row is updated in the datatable and in the column.

User Story #730 Student data import/export/delete functionality

System Tests

- Test ID - 024
- Purpose: Testing if the delete data functionality works.
- Test Setup: A user is logged in and in the settings page.
- Input: The user hits the delete data button.
- Expected Output: All information related to the student's courses will be removed.

System Tests

- Test ID - 025
- Purpose: Testing if the export data functionality works.
- Test Setup: A user is logged in and in the settings page.
- Input: The user hits the export data button.
- Expected Output: An xml file is downloaded with all the student's information.

System Tests

- Test ID - 026
- Purpose: Testing if the import data functionality works.
- Test Setup: A user is logged in, in the settings page, and just deleted all of their courses.
- Input: The user hits the import data button and enters the xml file they downloaded.
- Expected Output: All their courses will be restored.

User Story #757 - Leverage "what if" feature to automate curriculum data import.

System Tests

- Test ID - 027
- Purpose: Testing if the PDF import works correctly.
- Test Setup: The user is in the settings page.
- Input: The user hits import and selects their grade audit.
- Expected Output: All the user's grades, current courses, needed courses, and majors will be uploaded.

System Tests

- Test ID - 028
- Purpose: Testing if the system will reject an incorrect file.
- Test Setup: The user is in the settings page.
- Input: The user selects a random file.
- Expected Output: Nothing will be imported and an error message should appear.

User Story # 785 - Setup Android Development environment and Create an Android App (POC) that consumes the Tomcat Web services from Story 784.

System Tests

- Test ID - 029
- Purpose: Testing if the android app can log in correctly.
- Test Setup: The user has an account and the Android app open.
- Input: The user enters their username and password correctly.
- Expected Output: The user will be logged in.

System Tests

- Test ID - 030
- Purpose: Testing if the android app prevents login with bad password.
- Test Setup: The user has an account and the Android app open.
- Input: The user enters their username correctly but messes up the password.
- Expected Output: An error message will appear indicating that the username and password are incorrect.

System Tests

- Test ID - 031
- Purpose: Testing if user can remove a grade
- Test Setup: The user has an account and the Android app open.
- Input: The user hits the grade to be deleted and confirms.
- Expected Output: The grade is removed from the listview and from the Database.

System Tests

- Test ID - 032
- Purpose: Testing if user can add a grade
- Test Setup: The user has an account and the Android app open.

- Input: The user adds a grade.
- Expected Output: Listview from the previous page is updated and the grade is added to the database.

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
- **GPA**: Stands for grade point average. It takes the average of all the grades the user has taken and represents it as a number.
- **Semester Dashboard**: The dashboard that tracks the grades of a student throughout the semester.
- **GPA Grade Audit**: The report provided by PantherSoft that has information about what grades a student has, what grades s/he is taking, what courses s/he needs to take, and what is their major.
- **Forecast**: This is an overview of the student's progress that comes with a message to let the student know what GPA to maintain in order to achieve their goal GPA.
- **Overall GPA Dashboard**: The dashboard that takes all the courses the user has taken and gives advice about what is needed to achieve a certain GPA.

APPENDIX

Appendix A - UML Diagrams

Static UML Diagrams

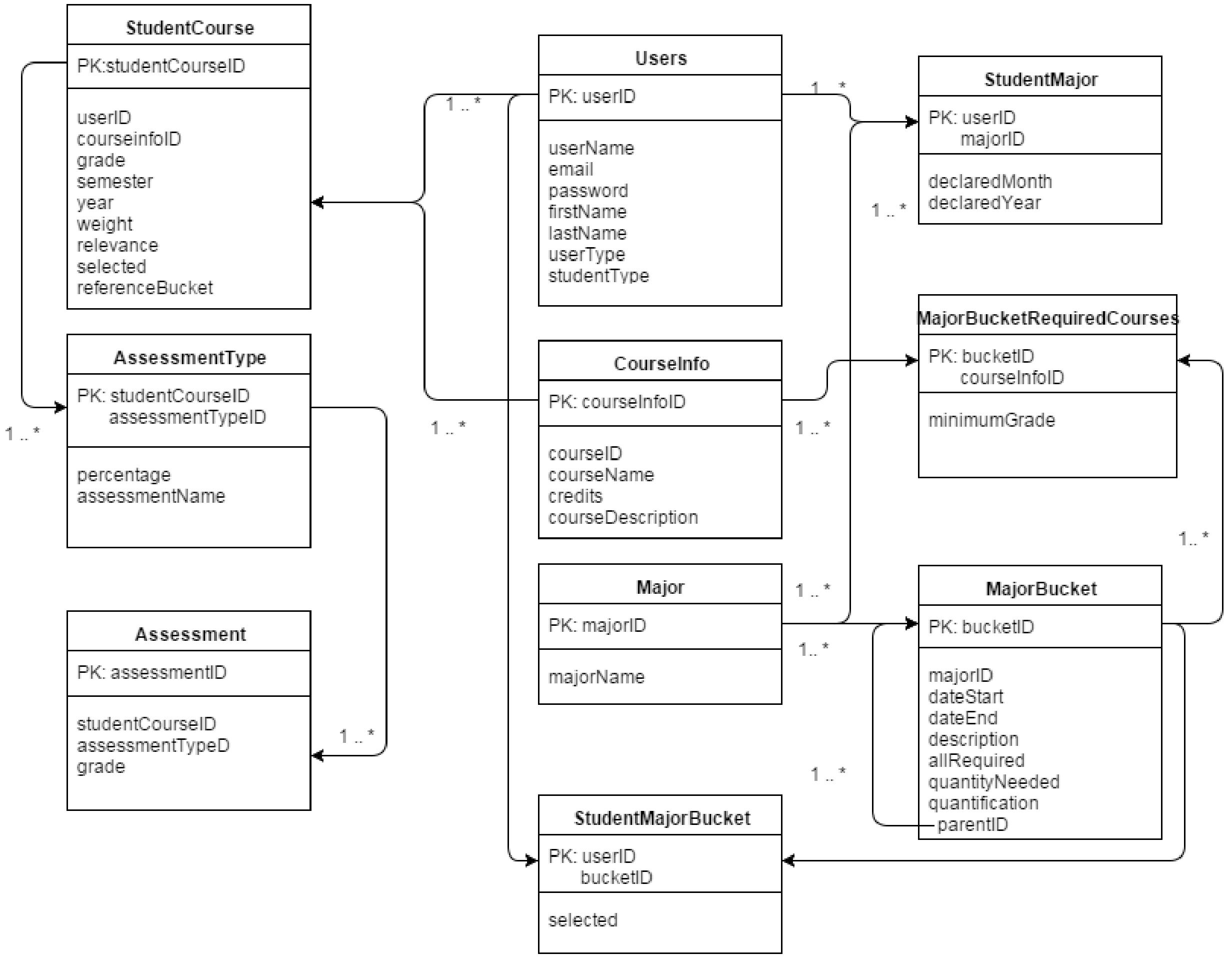


Figure #5 - Object Diagram

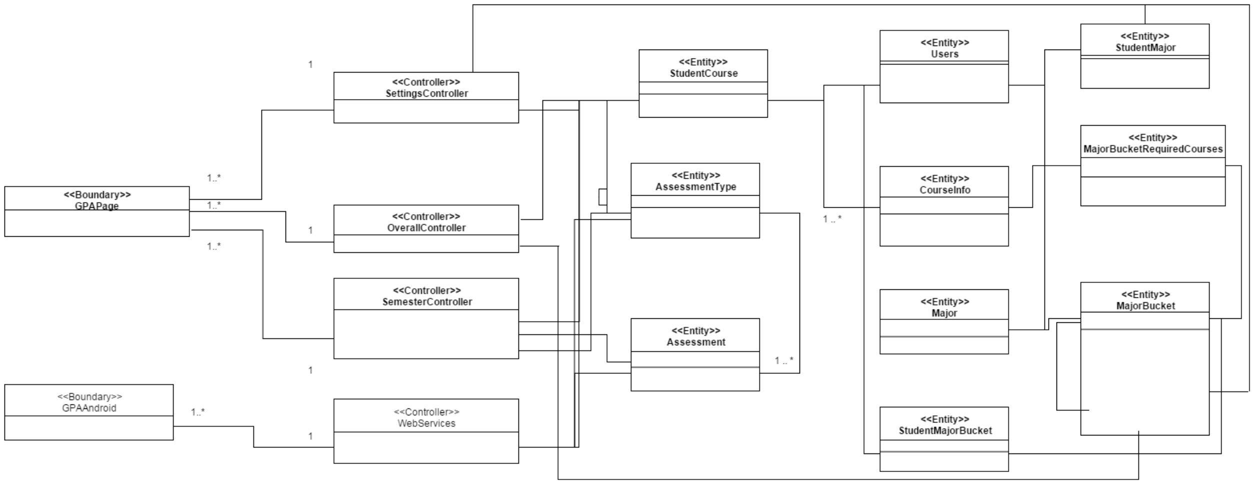


Figure #6 Minimal Class Diagram

Dynamic UML Diagrams

Sequence Diagrams

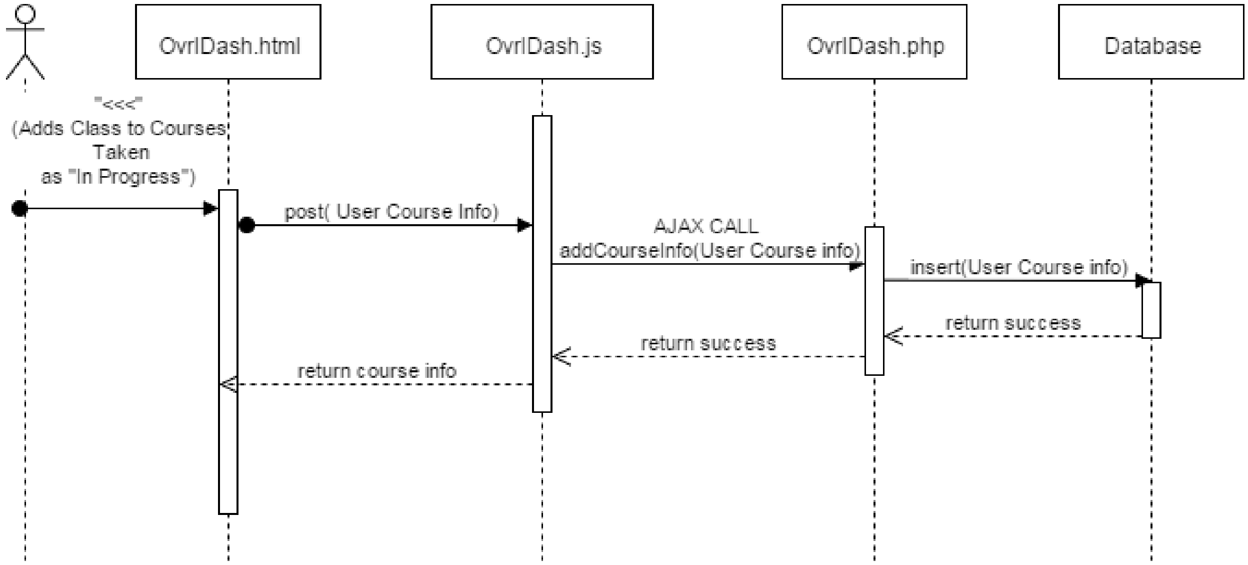


Figure #7 Add Student Course

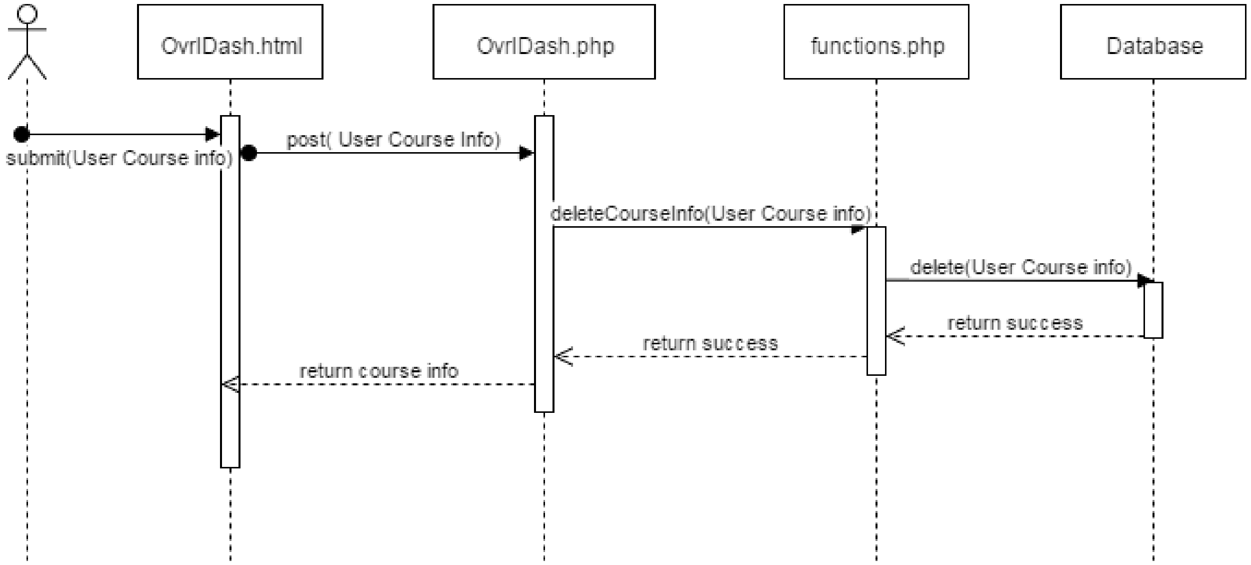


Figure #8 Delete Student Course

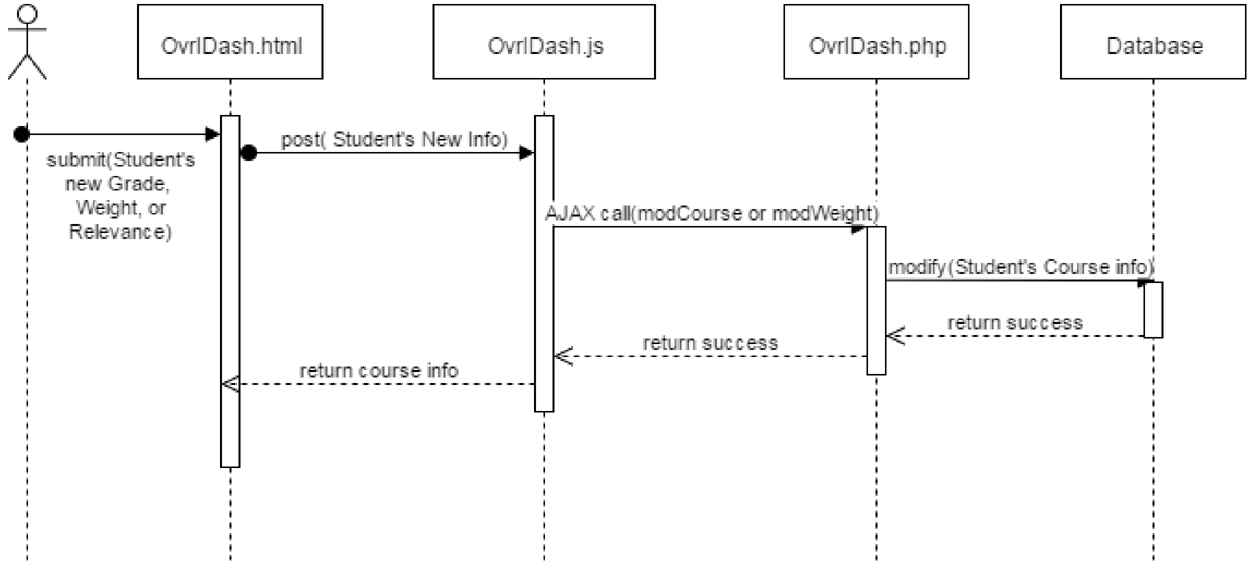

OvrlDash.html
OvrlDash.js
OvrlDash.php
Database
submit(Student's new Grade, Weight, or Relevance)
post(Student's New Info)
AJAX call(modCourse or modWeight)
modify(Student's Course info)
return success
return success
return course info

Figure #9 Modify Student Course

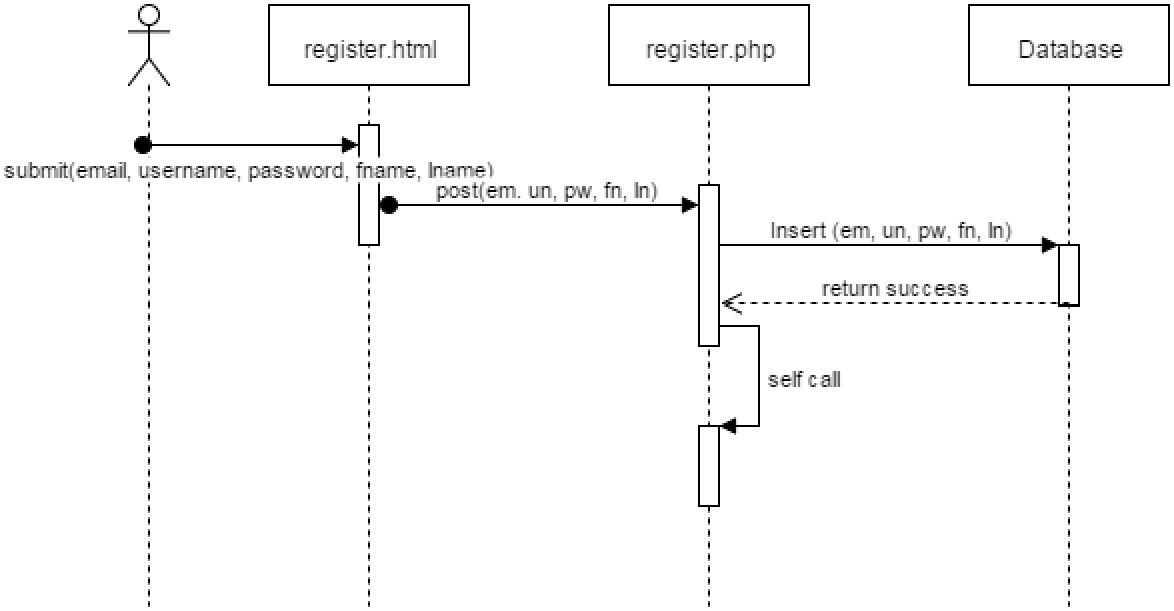


Figure #10 Register

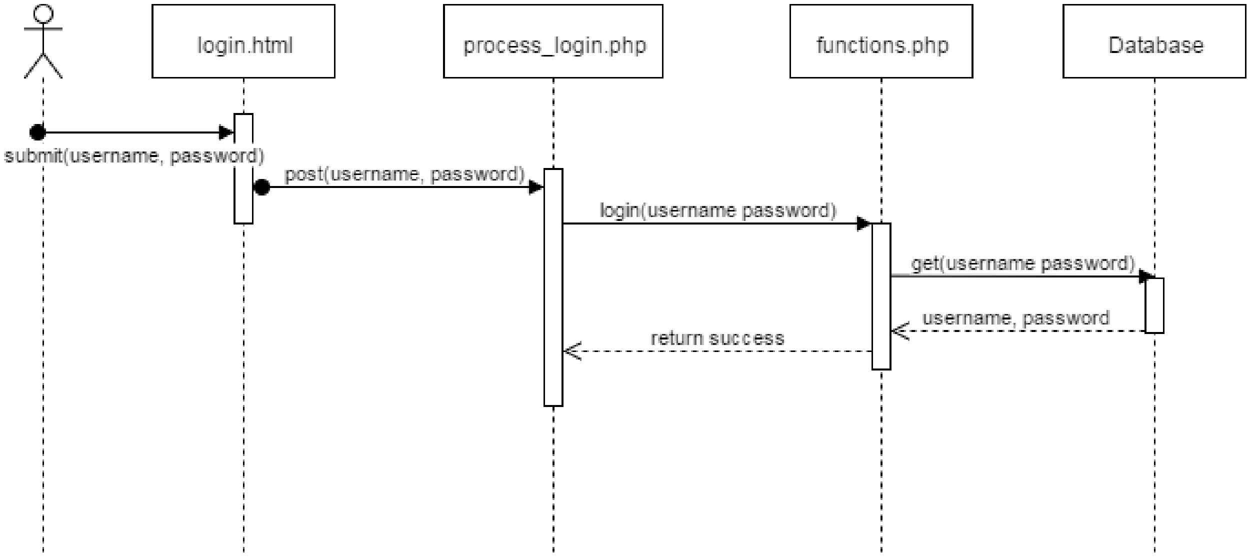


Figure #11 User Login

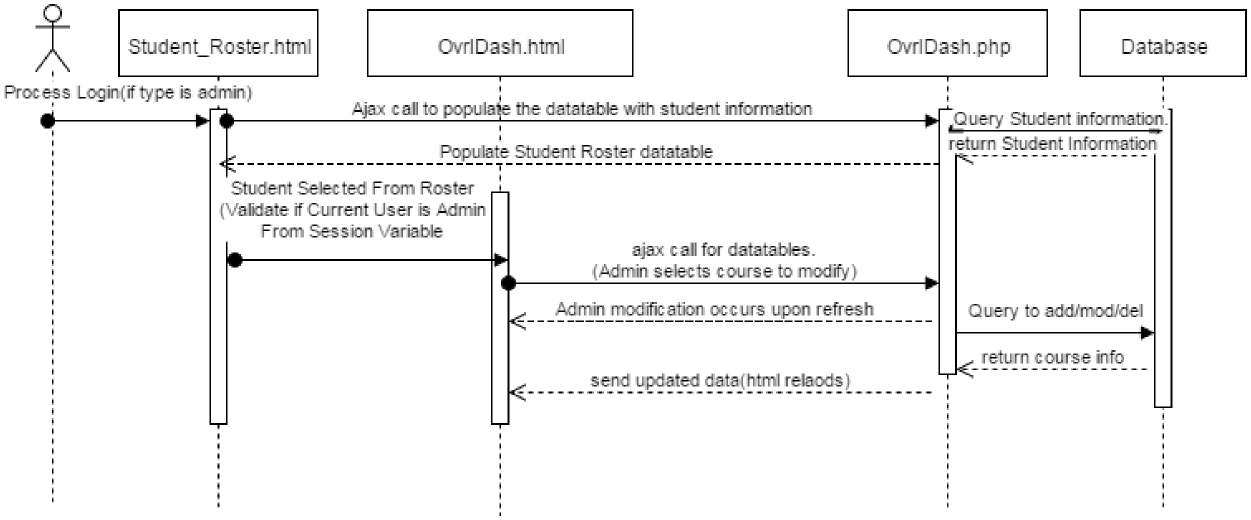

Student_Roster.html
OvrlDash.html
OvrlDash.php
Database
Process Login(if type is admin)
Ajax call to populate the datatable with student information
Query Student information.
return Student Information
Populate Student Roster datatable
Student Selected From Roster
(Validate if Current User is Admin
From Session Variable
ajax call for datatables.
(Admin selects course to modify)
Admin modification occurs upon refresh
Query to add/mod/del
return course info
send updated data(html relaods)

Figure #12 Admin User Login

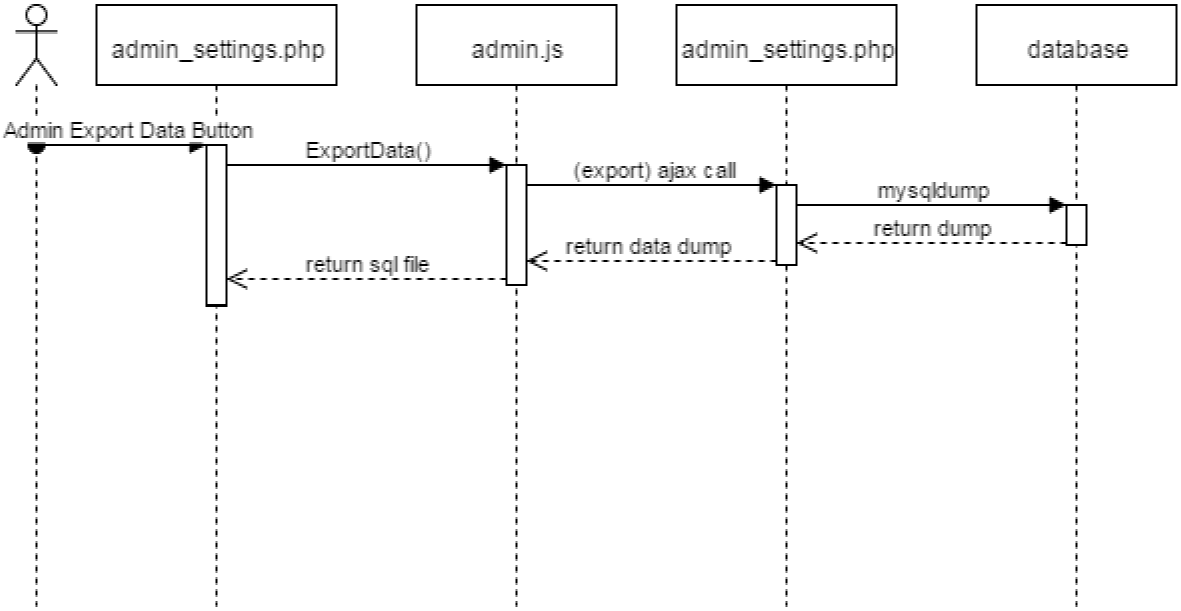


Figure # 13 Admin Export Data

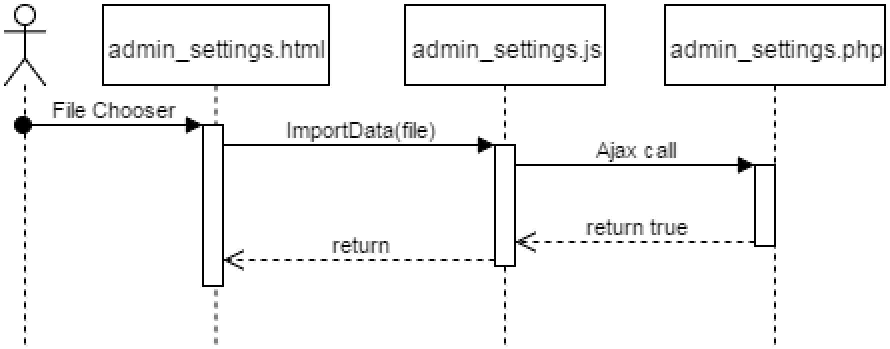


Figure #14 Admin Import Data

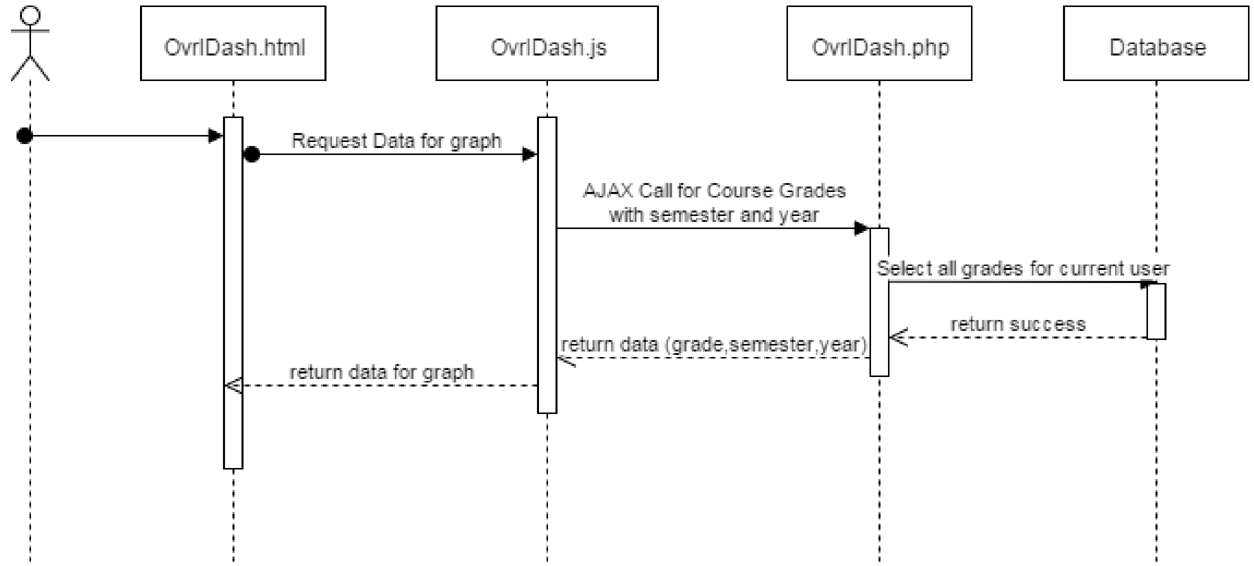

OvrlDash.html
OvrlDash.js
OvrlDash.php
Database
Request Data for graph
AJAX Call for Course Grades with semester and year
Select all grades for current user
return success
return data (grade,semester,year)
return data for graph

Figure #15 GPA Dashboard Graph

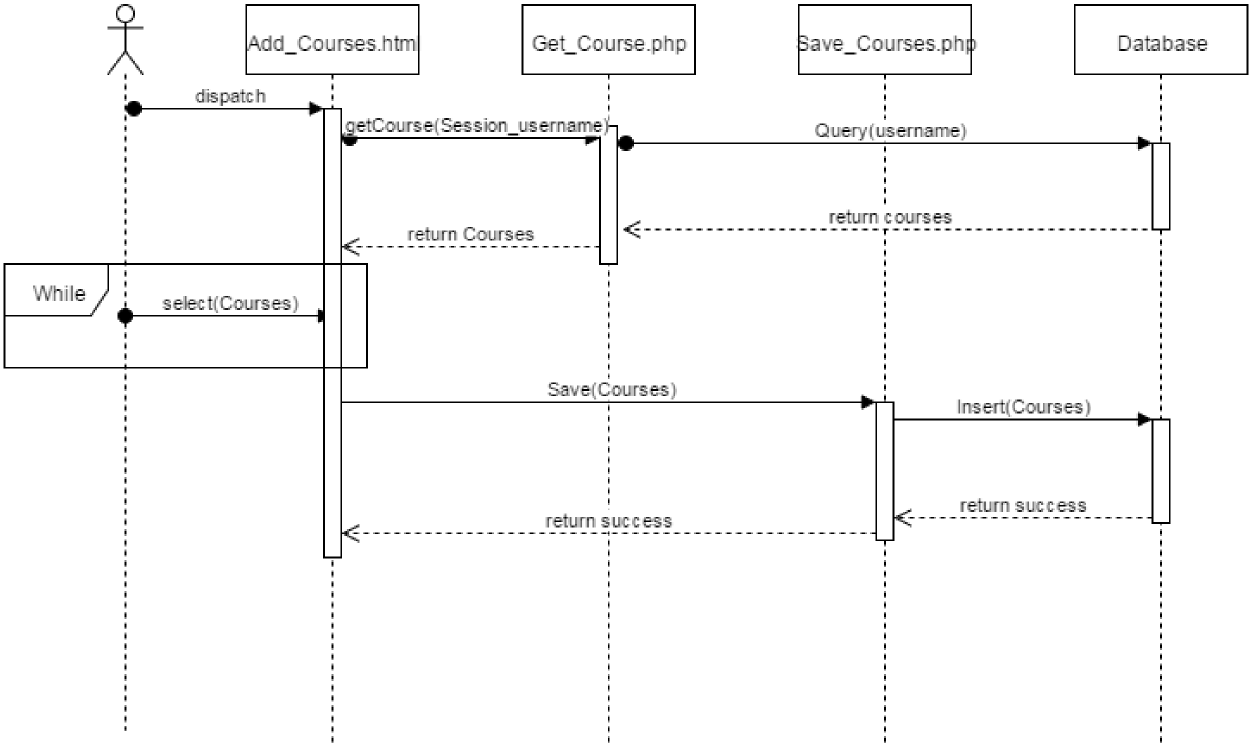


Figure #16 Add Current Courses

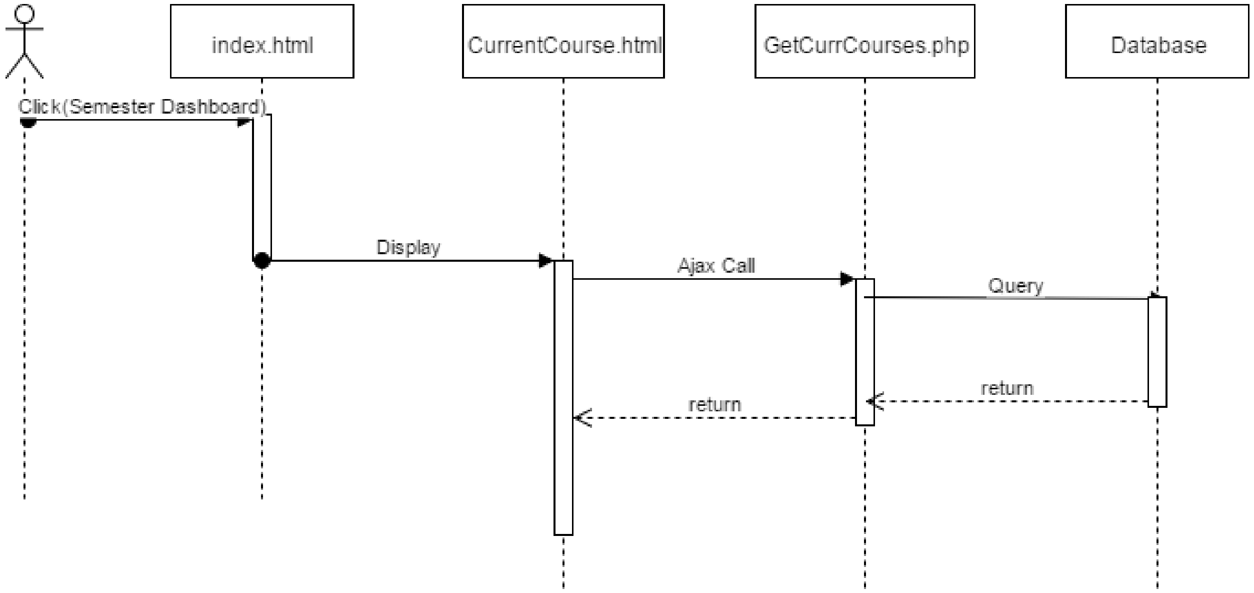


Figure #17 Get Current Courses

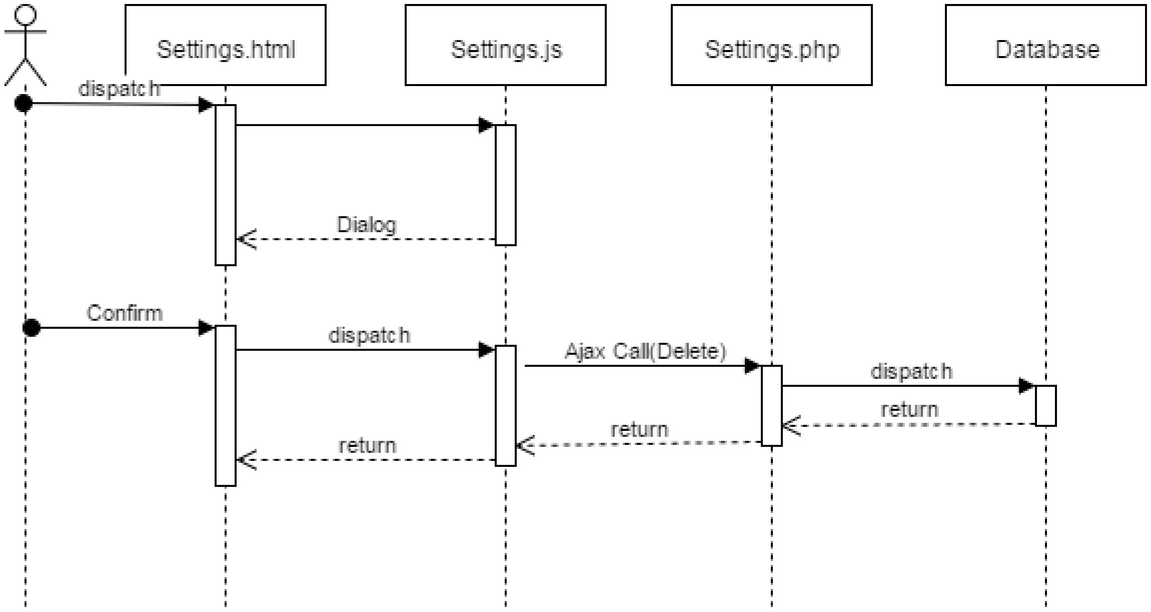


Figure #18 Student Delete Data

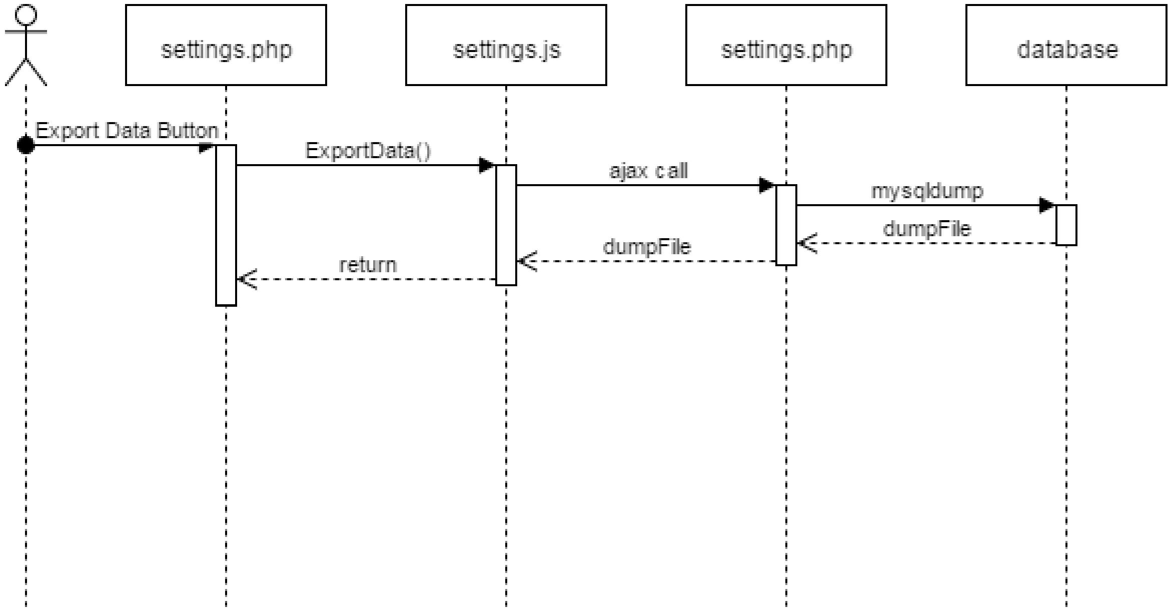


Figure #19 Student Export Data

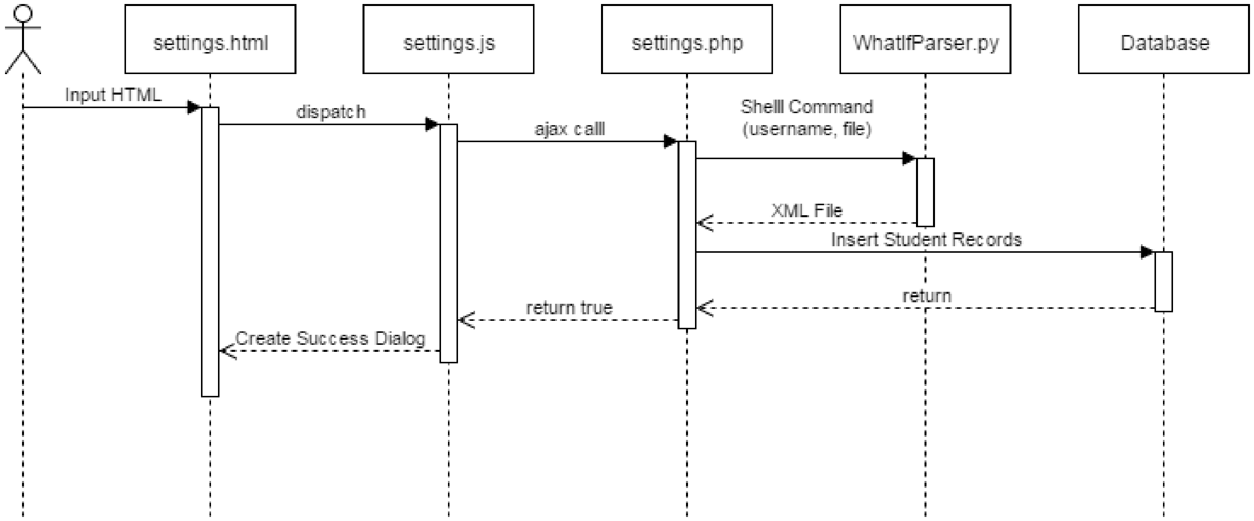


Figure #20 Import Grades from What If

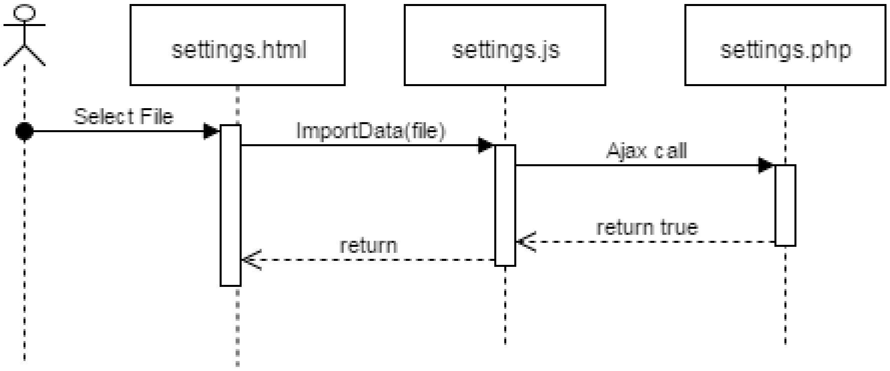


Figure #21 Import Student Data

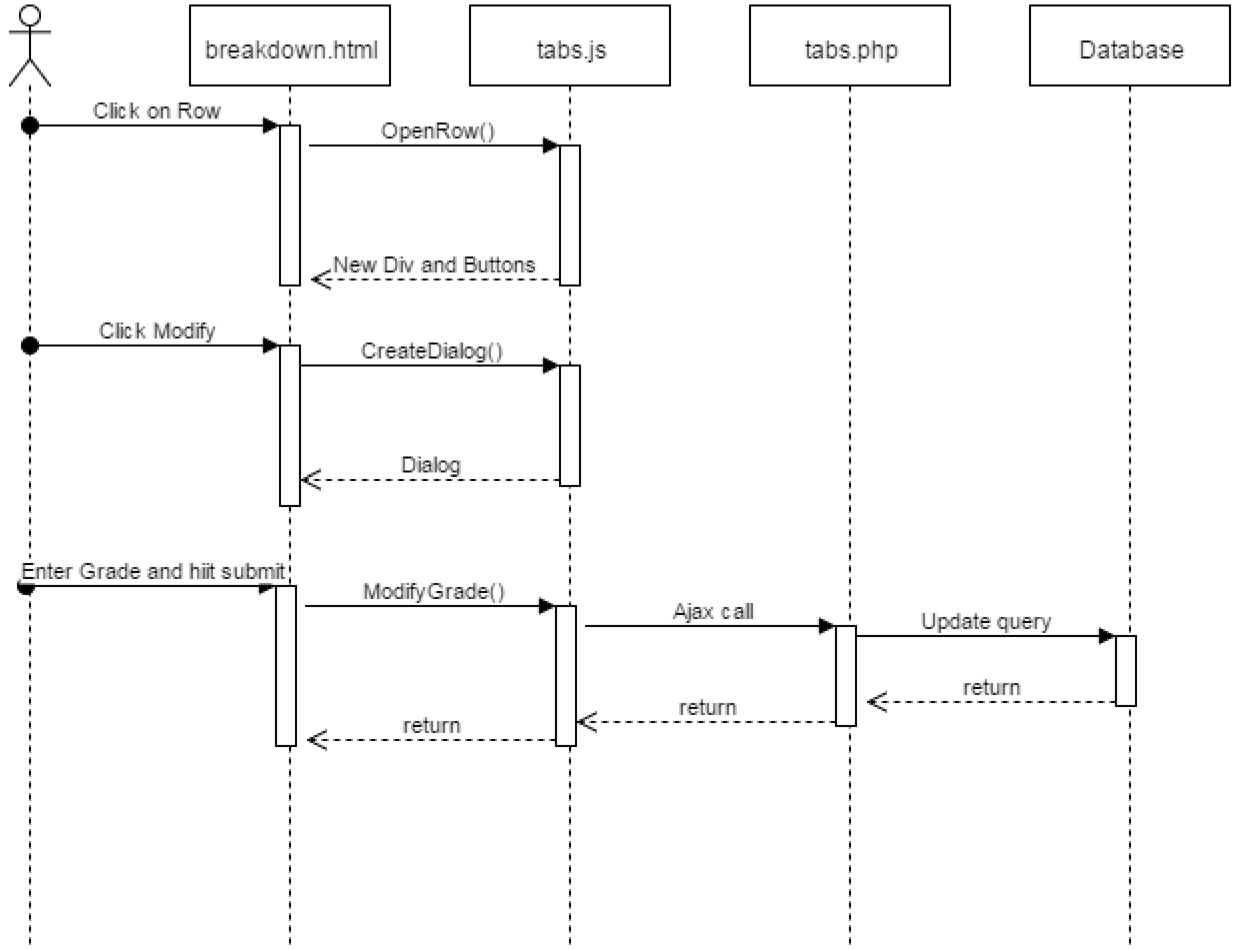


Figure #22 Modify Course Grades

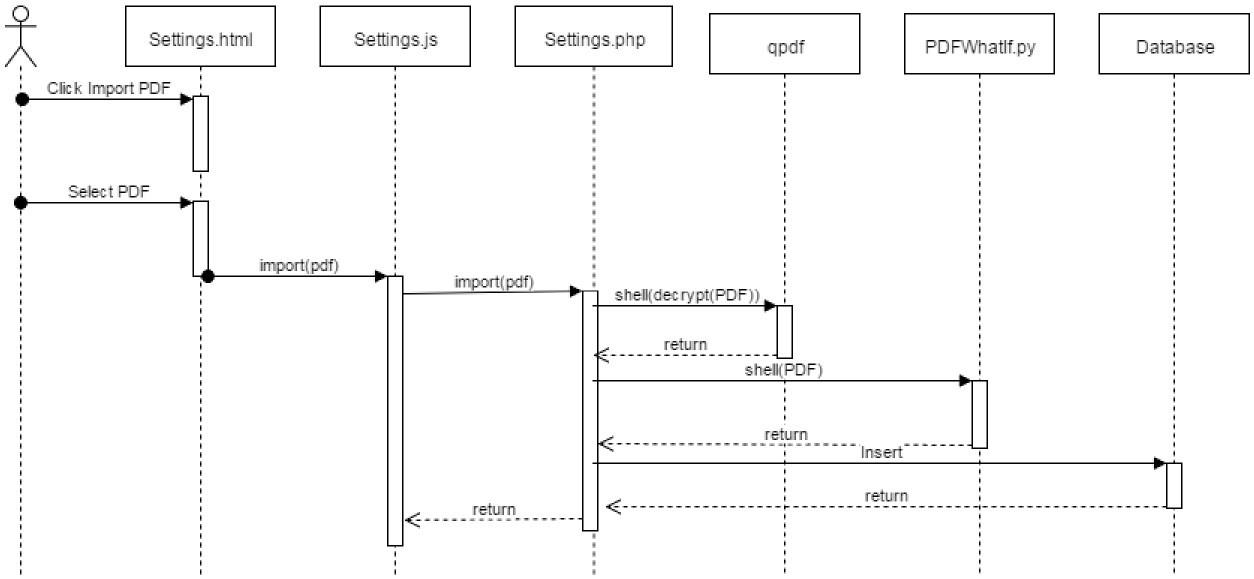


Figure #23 PDF Parser

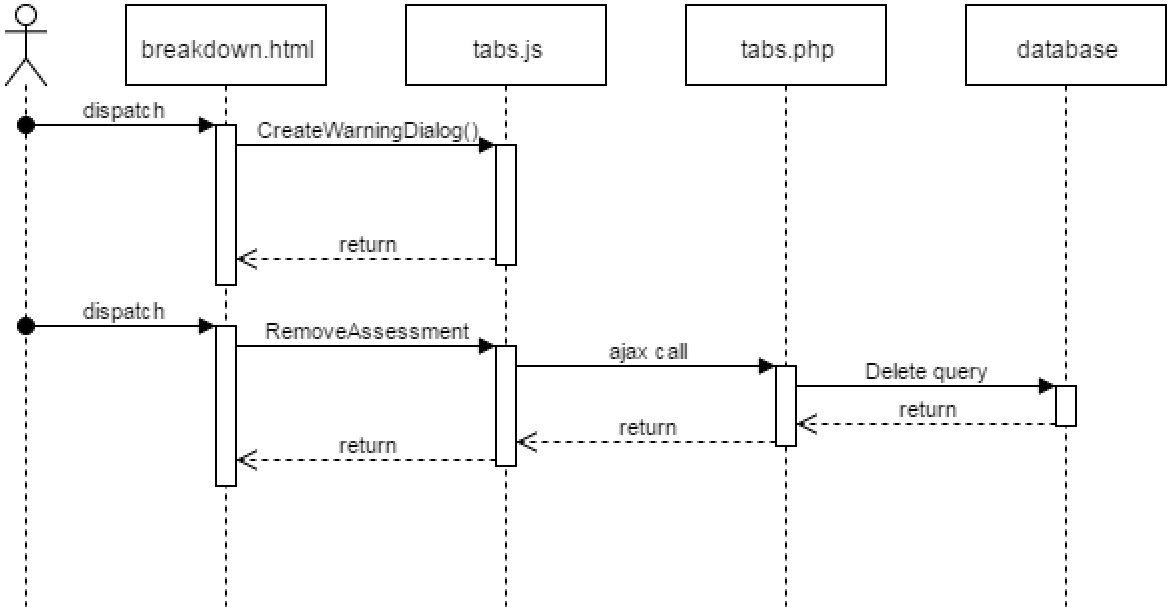


Figure #22 Remove Assessment

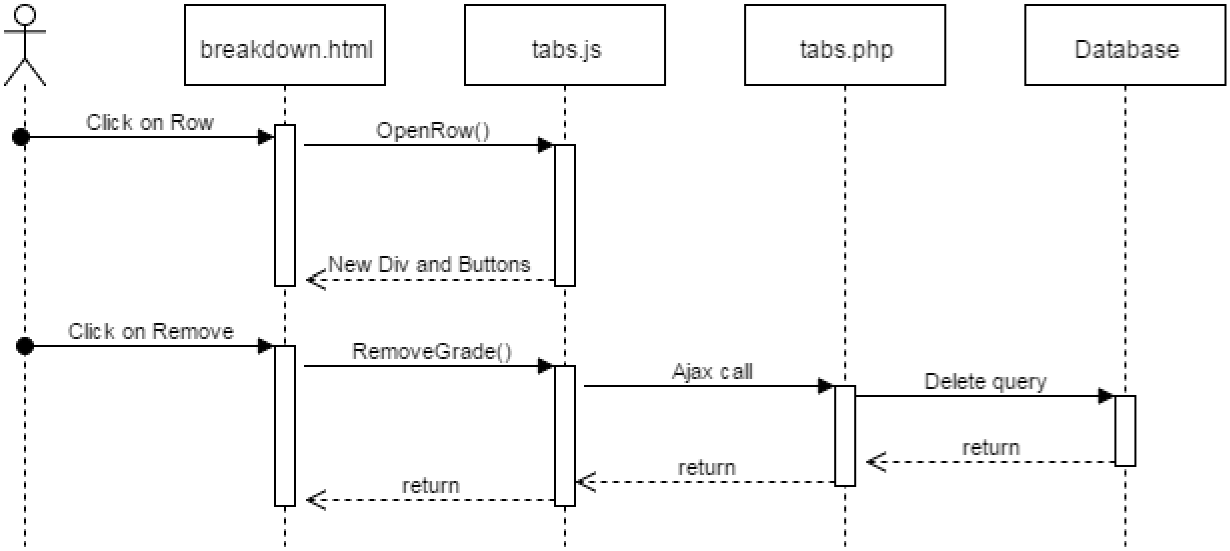


Figure #23 Remove Course Grade

State Diagrams

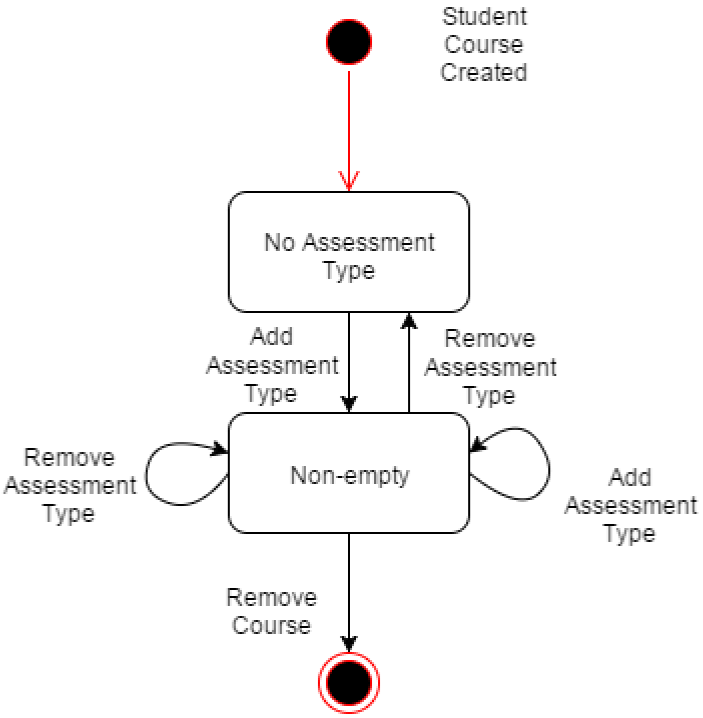
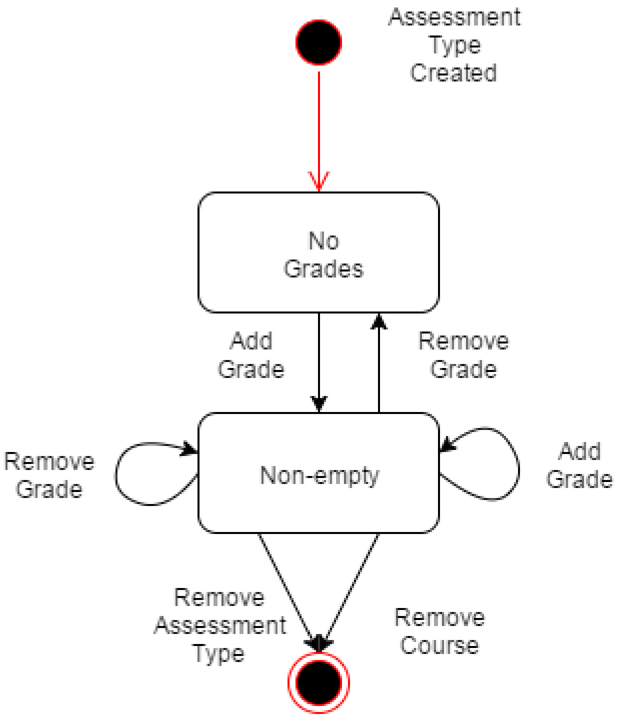
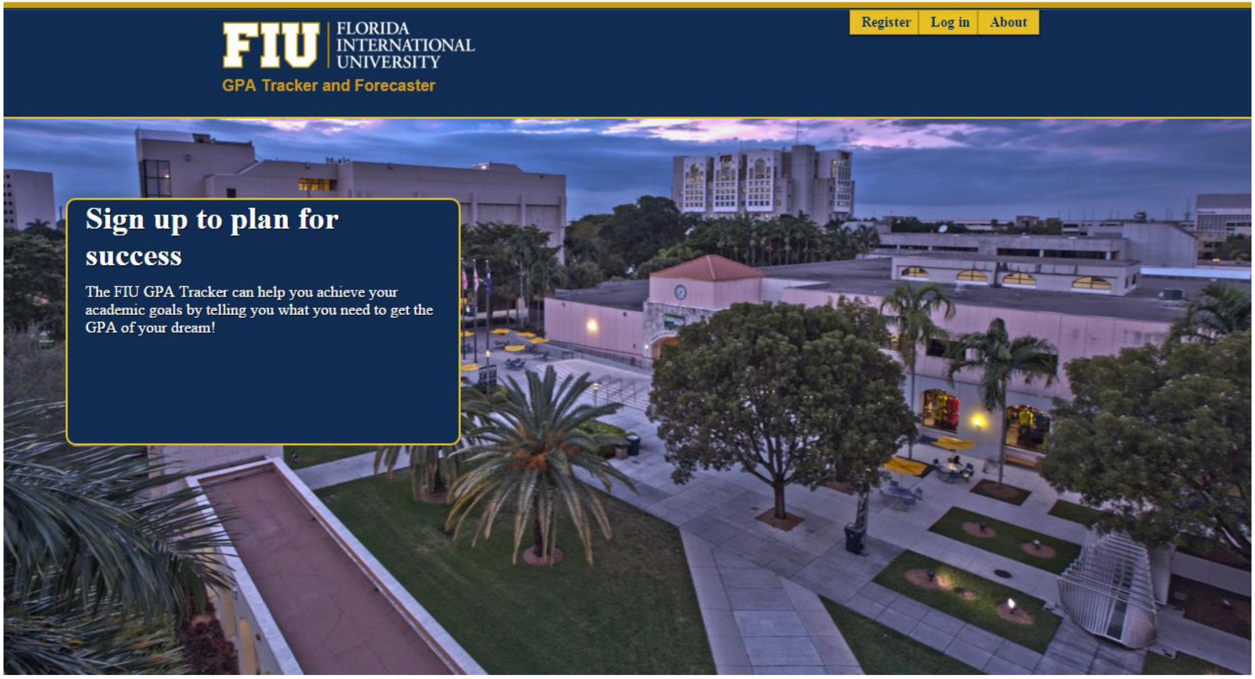


Figure #24 - Student Course



Figure#25 Assessment Type

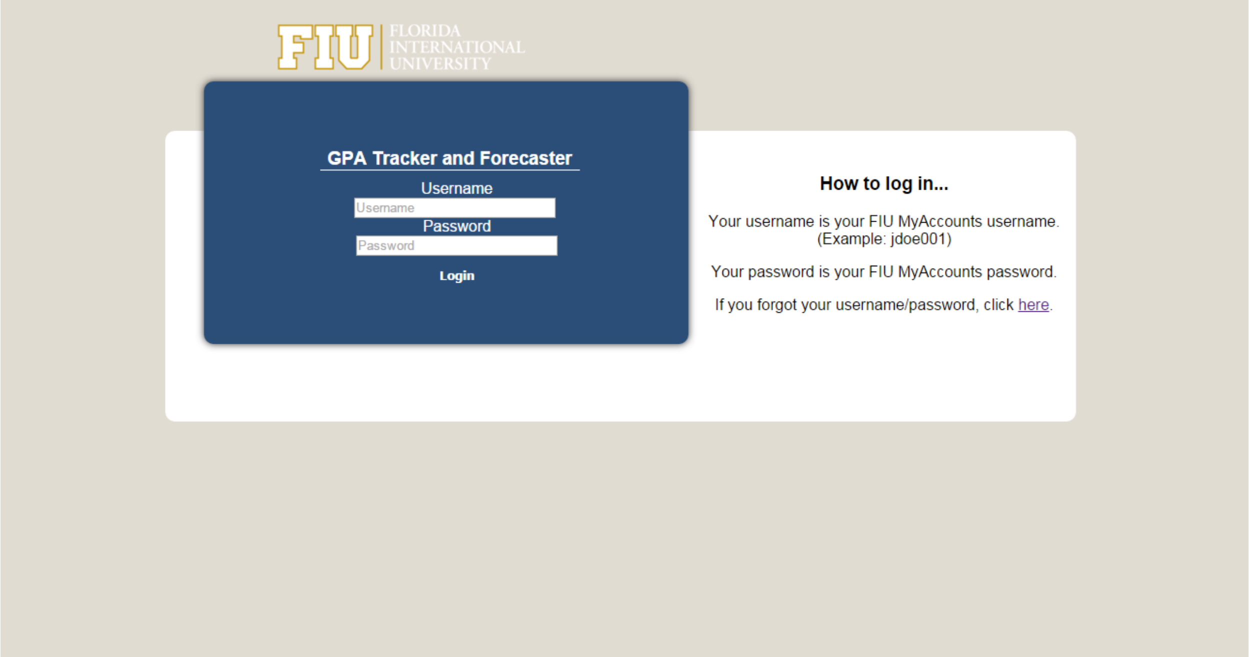
Appendix B - User Interface Design


Register
Log in
About
FIU FLORIDA INTERNATIONAL UNIVERSITY
GPA Tracker and Forecaster

Sign up to plan for success

The FIU GPA Tracker can help you achieve your academic goals by telling you what you need to get the GPA of your dream!

Figure #26 - Landing Page


FIU FLORIDA INTERNATIONAL UNIVERSITY
GPA Tracker and Forecaster
Username
Username
Password
Password
Login

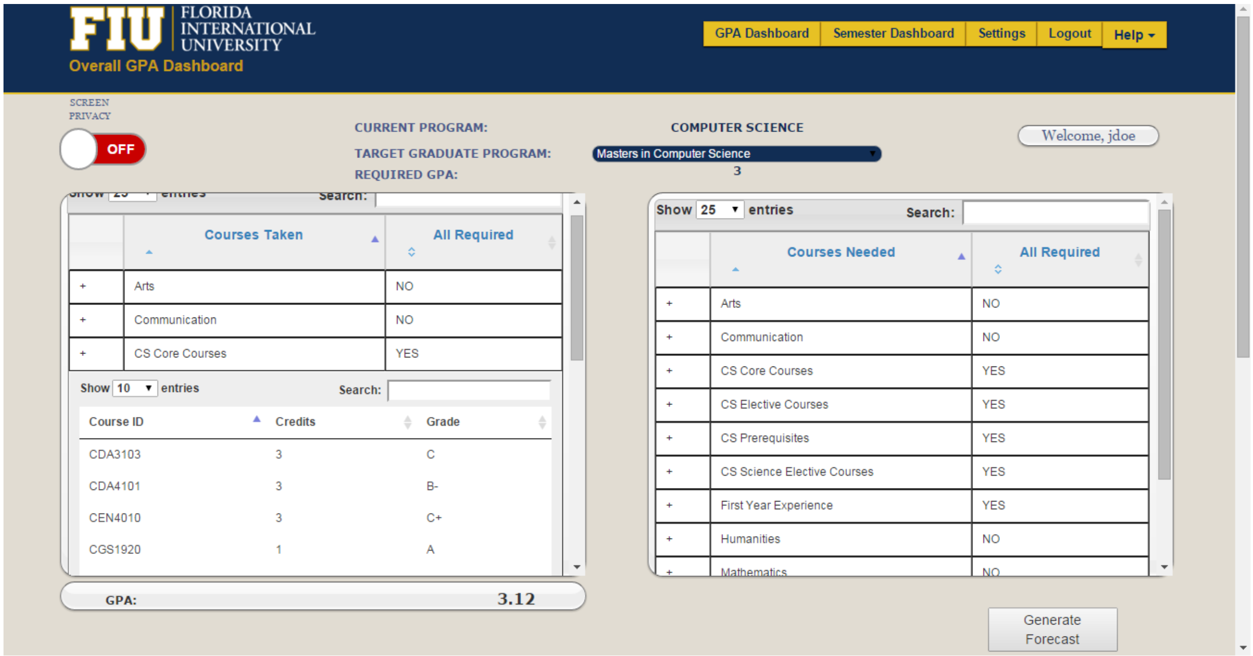
How to log in...

Your username is your FIU MyAccounts username. (Example: jdoe001)

Your password is your FIU MyAccounts password.

If you forgot your username/password, click here.

Figure #27 - Login Page



CURRENT PROGRAM: COMPUTER SCIENCE

TARGET GRADUATE PROGRAM: Masters in Computer Science

REQUIRED GPA: 3

	Courses Taken	All Required
+	Arts	NO
+	Communication	NO
+	CS Core Courses	YES

Show 10 entries Search:

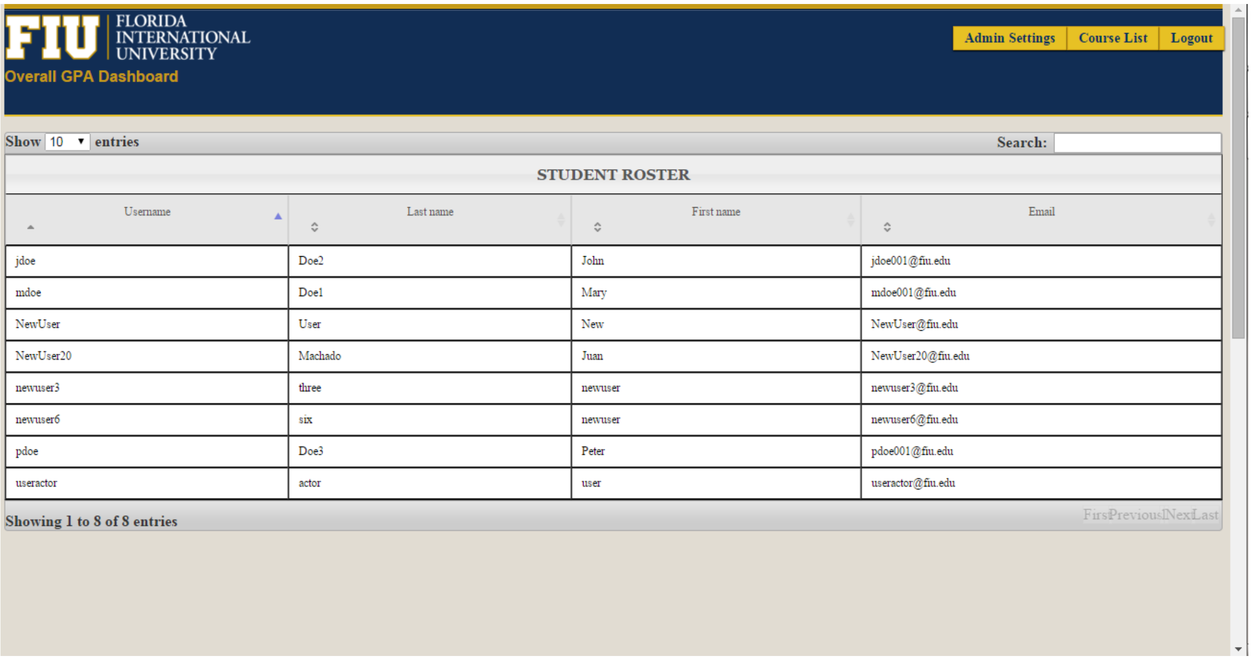
Course ID	Credits	Grade
CDA3103	3	C
CDA4101	3	B-
CEN4010	3	C+
CGS1920	1	A

GPA: 3.12

Show 25 entries Search:

	Courses Needed	All Required
+	Arts	NO
+	Communication	NO
+	CS Core Courses	YES
+	CS Elective Courses	YES
+	CS Prerequisites	YES
+	CS Science Elective Courses	YES
+	First Year Experience	YES
+	Humanities	NO
+	Mathematics	NO

Figure #28 - GPA Dashboard

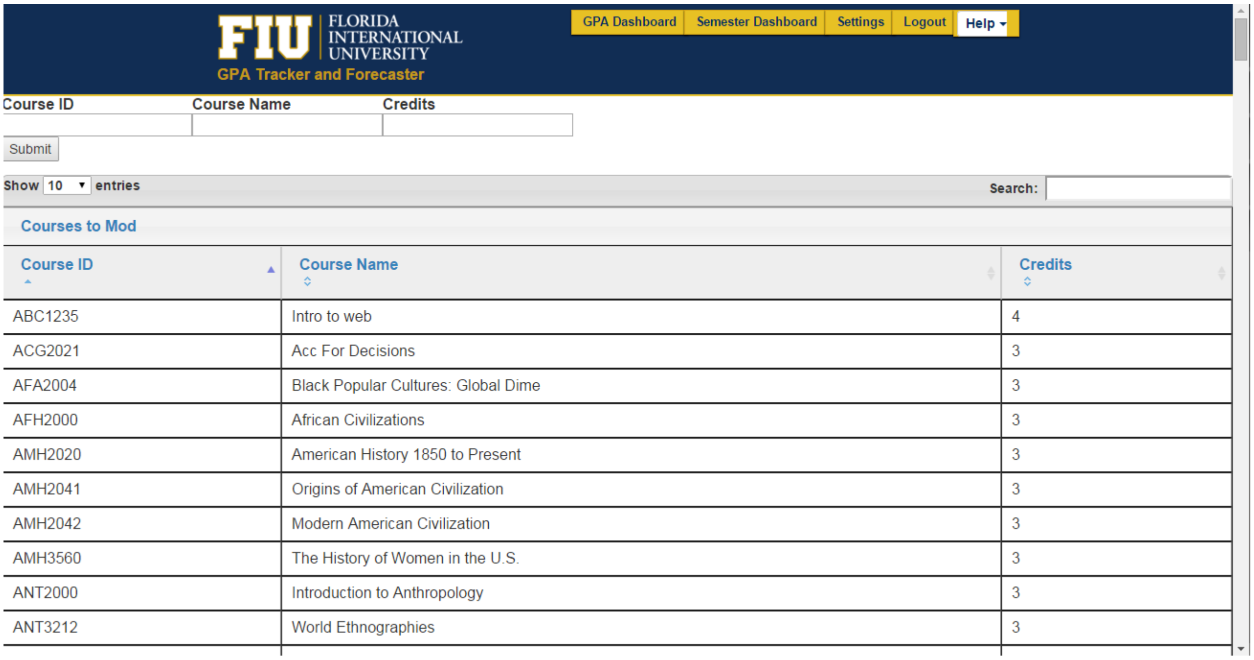


STUDENT ROSTER

Username	Last name	First name	Email
jdoe	Doe2	John	jdoe001@fiu.edu
mdoe	Doe1	Mary	mdoe001@fiu.edu
NewUser	User	New	NewUser@fiu.edu
NewUser20	Machado	Juan	NewUser20@fiu.edu
newuser3	three	newuser	newuser3@fiu.edu
newuser6	six	newuser	newuser6@fiu.edu
pdoe	Doe3	Peter	pdoe001@fiu.edu
useractor	actor	user	useractor@fiu.edu

Showing 1 to 8 of 8 entries

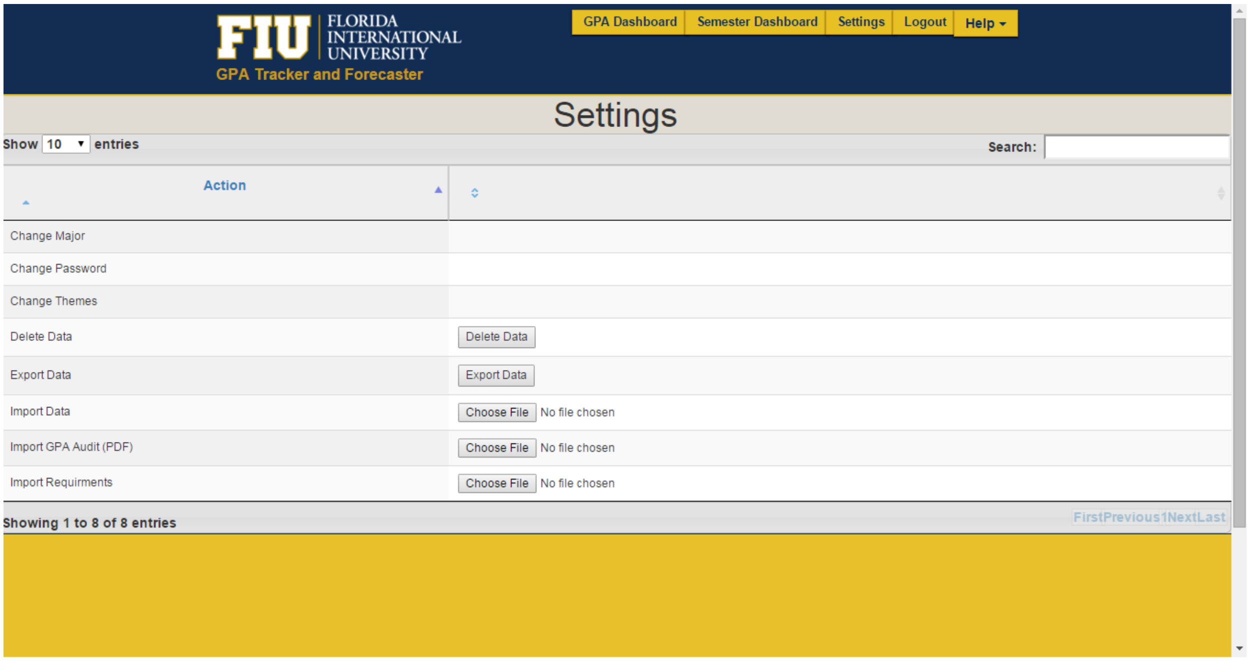
Figure #29 - Student Roster (Admin)



Courses to Mod

Course ID	Course Name	Credits
ABC1235	Intro to web	4
ACG2021	Acc For Decisions	3
AFA2004	Black Popular Cultures: Global Dime	3
AFH2000	African Civilizations	3
AMH2020	American History 1850 to Present	3
AMH2041	Origins of American Civilization	3
AMH2042	Modern American Civilization	3
AMH3560	The History of Women in the U.S.	3
ANT2000	Introduction to Anthropology	3
ANT3212	World Ethnographies	3

Figure #30 Master Course List (Admin)



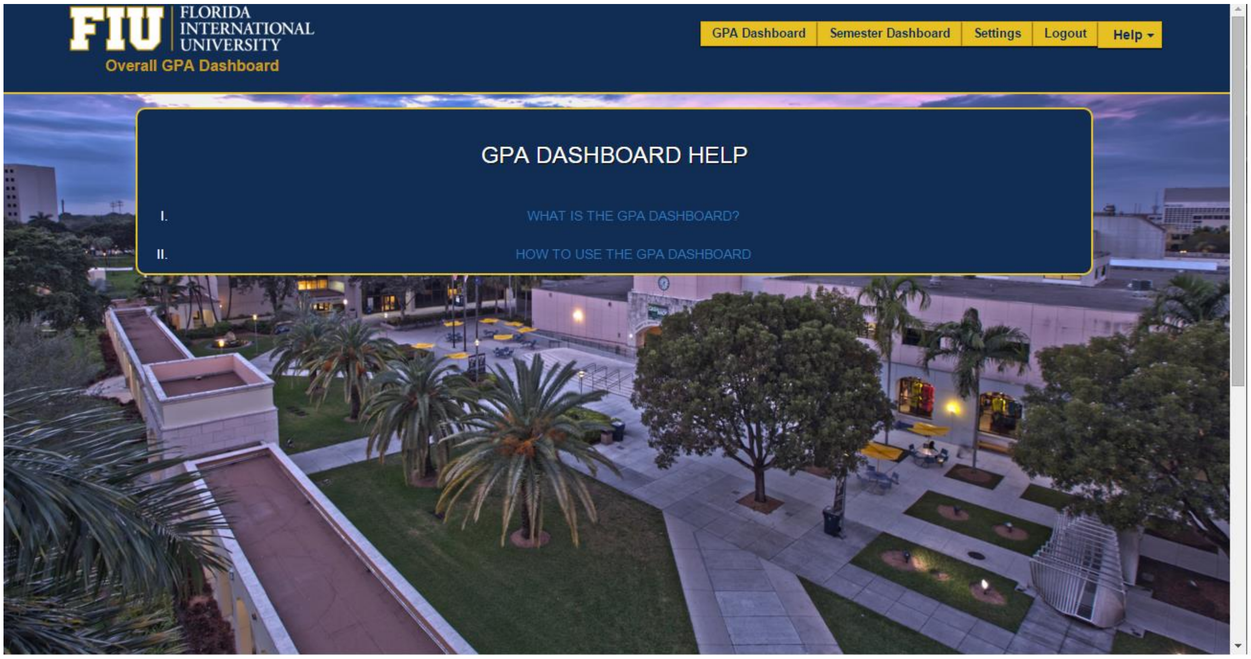
Settings

Show 10 entries

Action	
Change Major	
Change Password	
Change Themes	
Delete Data	Delete Data
Export Data	Export Data
Import Data	Choose File No file chosen
Import GPA Audit (PDF)	Choose File No file chosen
Import Requirments	Choose File No file chosen

Showing 1 to 8 of 8 entries

Figure #31 - Settings

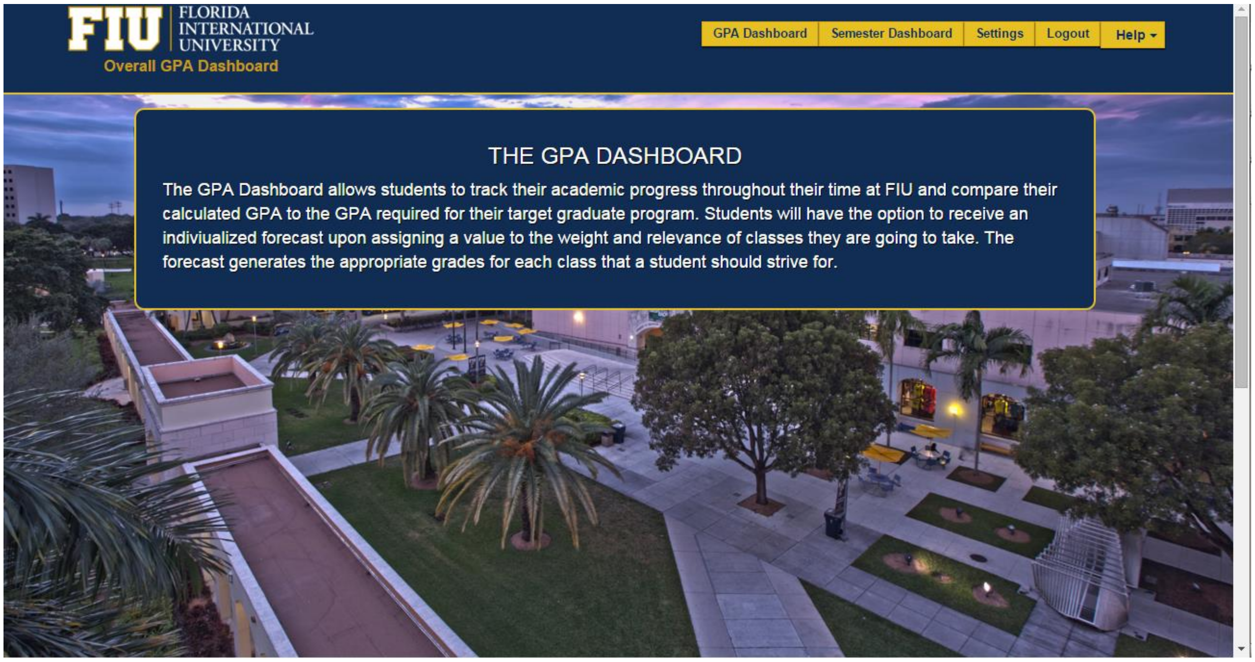


GPA DASHBOARD HELP

I. WHAT IS THE GPA DASHBOARD?

II. HOW TO USE THE GPA DASHBOARD

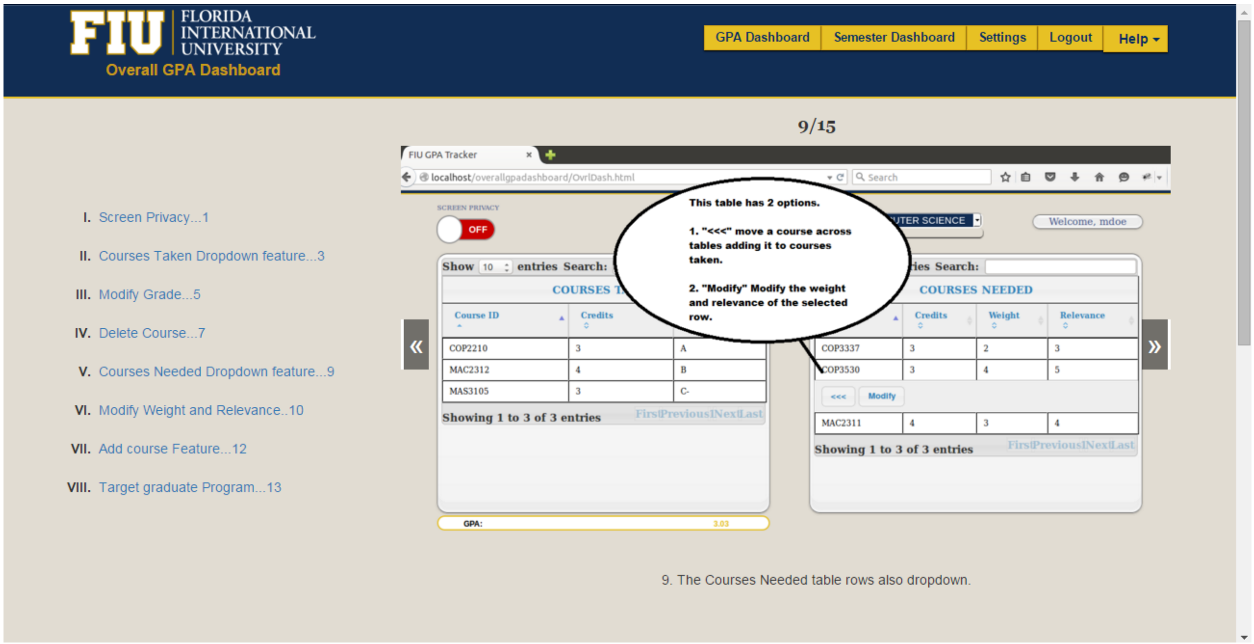
Figure #32 - GPA Dashboard Help



THE GPA DASHBOARD

The GPA Dashboard allows students to track their academic progress throughout their time at FIU and compare their calculated GPA to the GPA required for their target graduate program. Students will have the option to receive an indiviualized forecast upon assigning a value to the weight and relevance of classes they are going to take. The forecast generates the appropriate grades for each class that a student should strive for.

Figure #33 - GPA Dashboard Description



9/15

This table has 2 options.

1. "<<<" move a course across tables adding it to courses taken.

2. "Modify" Modify the weight and relevance of the selected row.

Show 10 entries Search:

COURSES T

Course ID	Credits	
COP2210	3	A
MAC2312	4	B
MAS3105	3	C-

Showing 1 to 3 of 3 entries FirstPreviousNextLast

GPA: 3.03

ies Search:

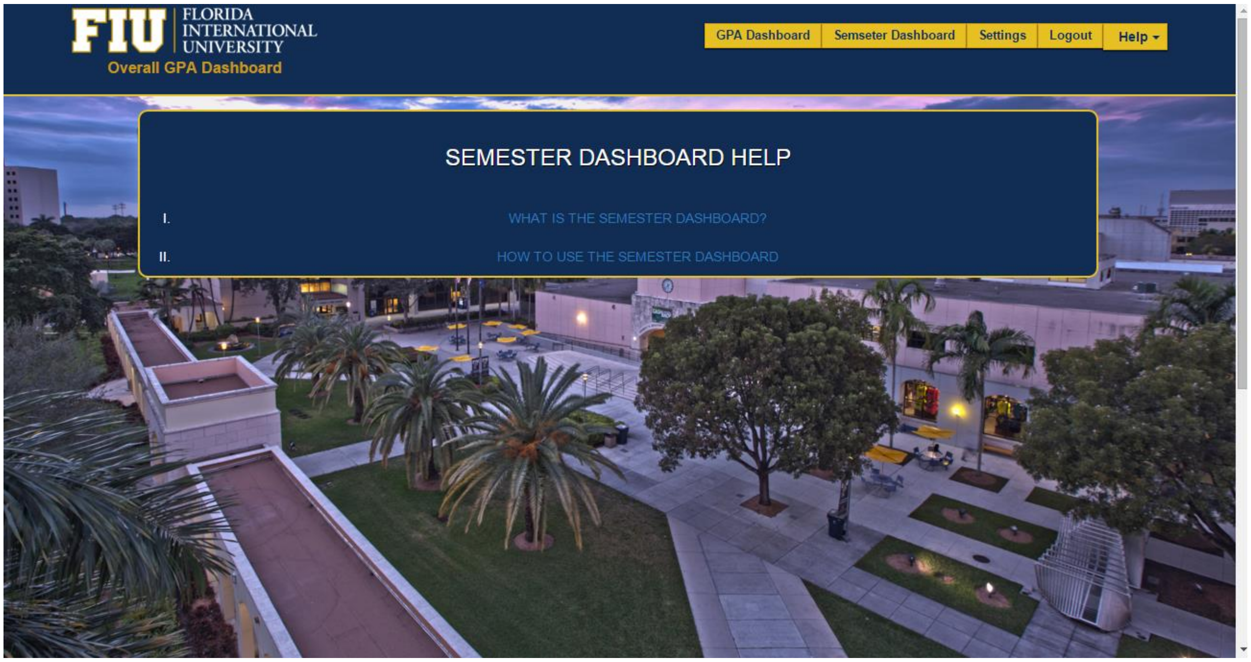
COURSES NEEDED

	Credits	Weight	Relevance
COP3337	3	2	3
COP3530	3	4	5
<<< Modify			
MAC2311	4	3	4

Showing 1 to 3 of 3 entries FirstPreviousNextLast

9. The Courses Needed table rows also dropdown.

Figure #34 - GPA Dashboard How to

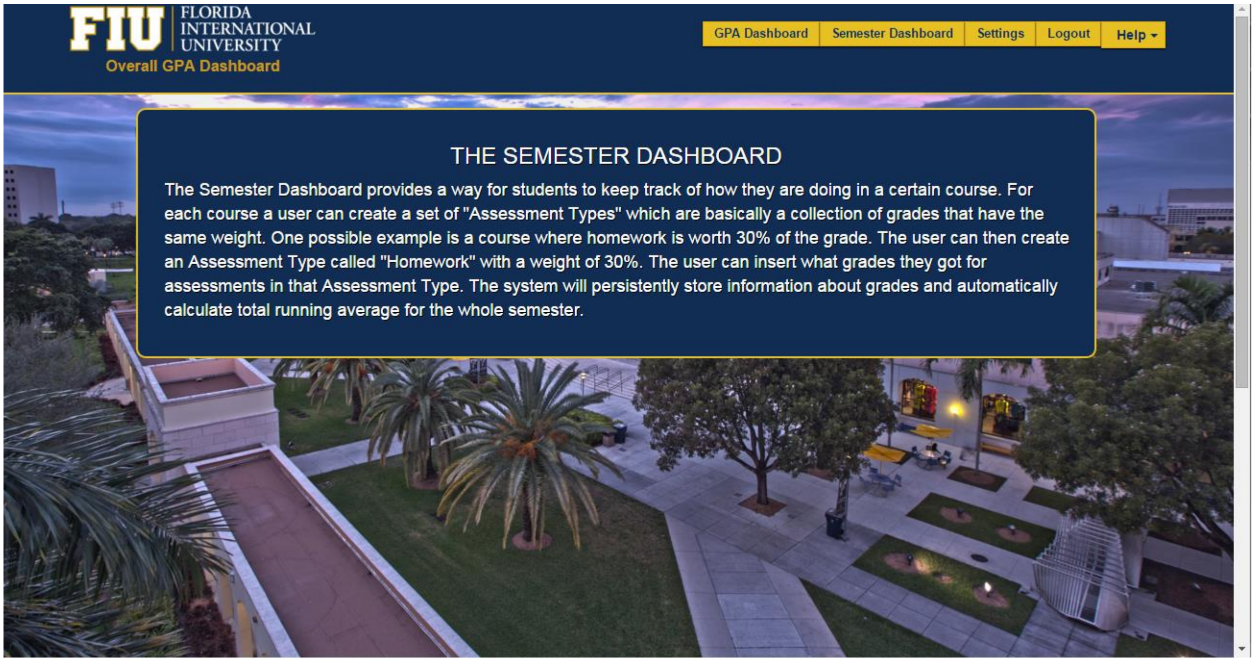


SEMESTER DASHBOARD HELP

I. WHAT IS THE SEMESTER DASHBOARD?

II. HOW TO USE THE SEMESTER DASHBOARD

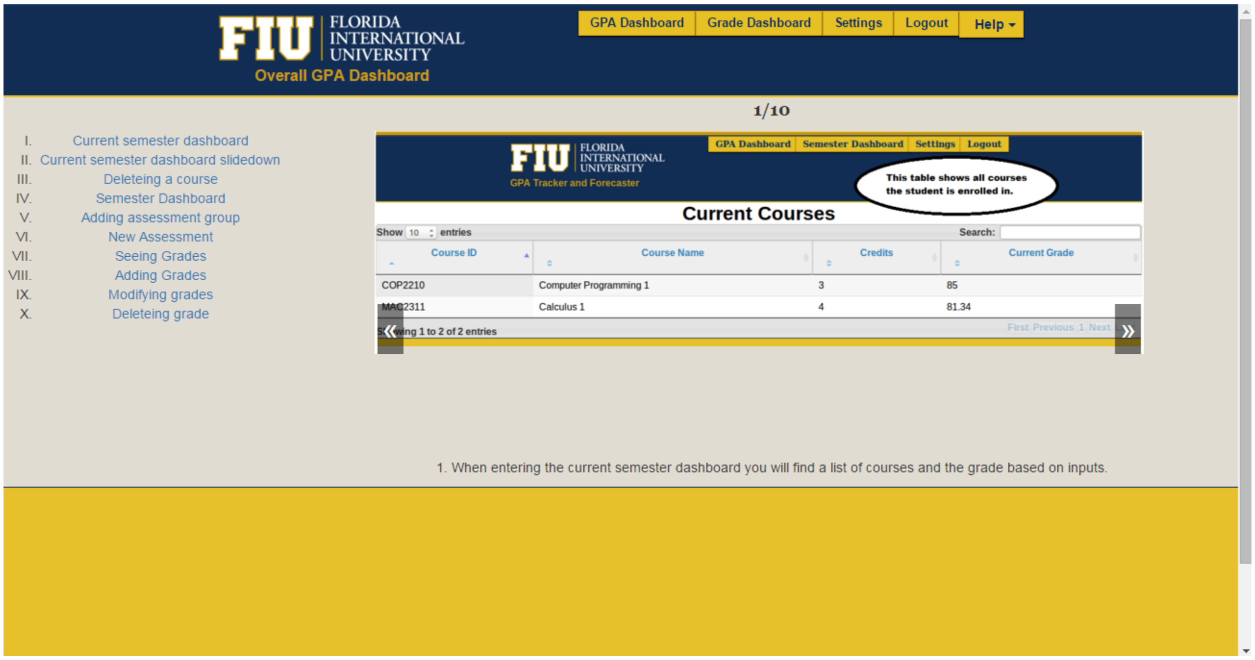
Figure #35 - Semester Dashboard Help



THE SEMESTER DASHBOARD

The Semester Dashboard provides a way for students to keep track of how they are doing in a certain course. For each course a user can create a set of "Assessment Types" which are basically a collection of grades that have the same weight. One possible example is a course where homework is worth 30% of the grade. The user can then create an Assessment Type called "Homework" with a weight of 30%. The user can insert what grades they got for assessments in that Assessment Type. The system will persistently store information about grades and automatically calculate total running average for the whole semester.

Figure #36 - Semester Dashboard Description



I. Current semester dashboard
II. Current semester dashboard slidedown
III. Deleteing a course
IV. Semester Dashboard
V. Adding assessment group
VI. New Assessment
VII. Seeing Grades
VIII. Adding Grades
IX. Modifying grades
X. Deleteing grade

This table shows all courses the student is enrolled in.

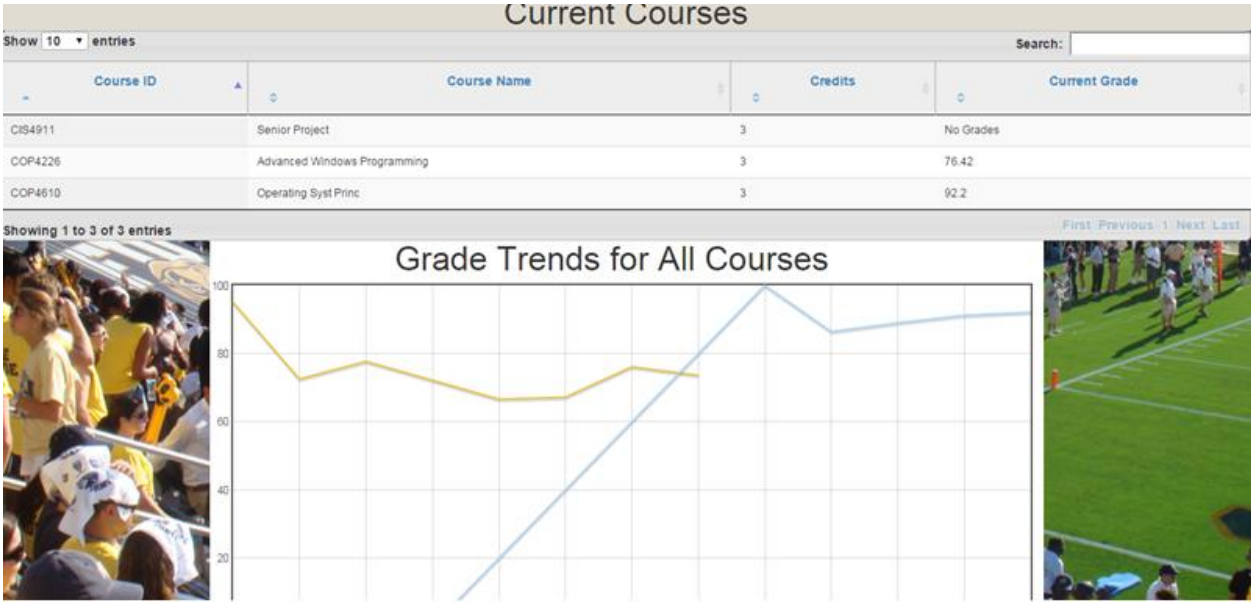
Current Courses

Course ID	Course Name	Credits	Current Grade
COP2210	Computer Programming 1	3	85
MAC2311	Calculus 1	4	81.34

Showing 1 to 2 of 2 entries

1. When entering the current semester dashboard you will find a list of courses and the grade based on inputs.

Figure#37 - Semester Dashboard How to.



Current Courses

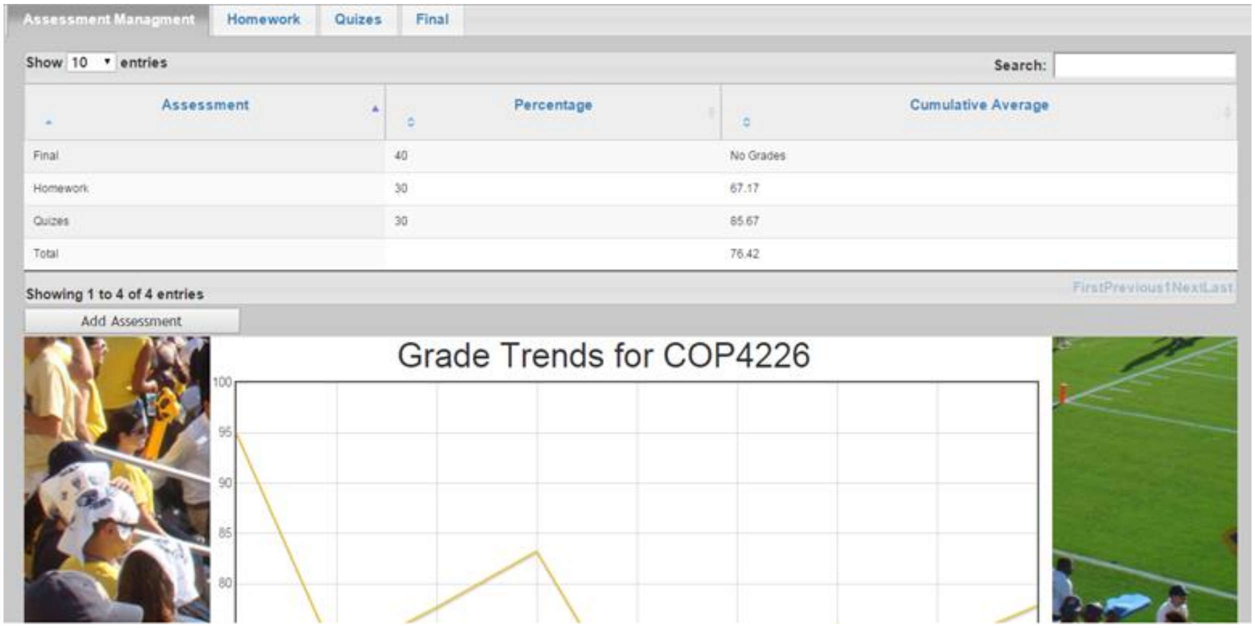
Show 10 entries

Search:

Course ID	Course Name	Credits	Current Grade
CIS4911	Senior Project	3	No Grades
COP4226	Advanced Windows Programming	3	76.42
COP4610	Operating Syst Princ	3	92.2

Showing 1 to 3 of 3 entries

Figure #38 - Semester Dashboard

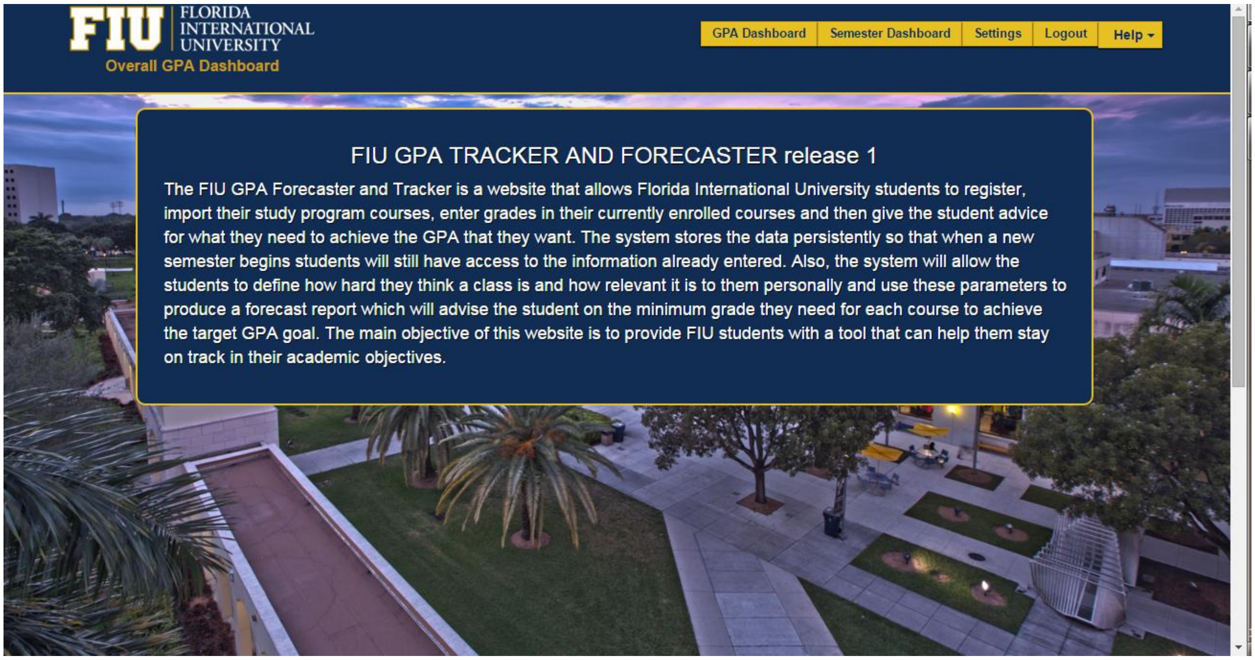


Show 10 entries Search:

Assessment	Percentage	Cumulative Average
Final	40	No Grades
Homework	30	67.17
Quizes	30	85.67
Total		76.42

Showing 1 to 4 of 4 entries

Figure #39 - Semester Dashboard Assessment Breakdown



FIU GPA TRACKER AND FORECASTER release 1

The FIU GPA Forecaster and Tracker is a website that allows Florida International University students to register, import their study program courses, enter grades in their currently enrolled courses and then give the student advice for what they need to achieve the GPA that they want. The system stores the data persistently so that when a new semester begins students will still have access to the information already entered. Also, the system will allow the students to define how hard they think a class is and how relevant it is to them personally and use these parameters to produce a forecast report which will advise the student on the minimum grade they need for each course to achieve the target GPA goal. The main objective of this website is to provide FIU students with a tool that can help them stay on track in their academic objectives.

Figure #40 - About

Appendix C - Sprint Review Reports

Sprint 1 Report

Date: 9/11/2015
Attendees: Johann Henao, Ike Ukachi- Lois, Juan Machado
Discussed Topics:
Considering the initial goals, cost estimates, and acceptance criteria, briefly explain what was achieved and what was not achieved in this sprint. Specify the reasons for not being able to finish all the work that was initially planned for this sprint. Specify if the product backlog was modified as a result of this meeting and if so, how.

In Sprint 1 the initial goals were to setup the development environment and to acquire basic web development skills to allow the team to be comfortable with the forthcoming project. A landing page as well as the database design and setup were also goals for sprint 1. A number of tutorials were completed on Codecademy for languages including html/CSS, jQuery, JavaScript and some PHP. Upon completion, the team was able to create the landing page for the FIU GPA Tracker and Forecaster website. In addition, a register page was created along with functionality so the data would be inputted into the database. The login page was also created however the functionality wasn't completed due to inexperience with handling PHP sessions.

Sprint 2 Report

Date: 9/25/2015
Attendees: Johann Henao, Ike Ukachi- Lois, Juan Machado
Discussed Topics:

In Sprint 2, the initial goals, were to complete any unfinished stories from the first sprint and to begin creating the two main dashboards for the website. additional tutorials were taken to enhance the teams capability when it comes to web development. Also mockups for the website were to be completed in order to understand the overall flow of the entire website. The production of content for the website was initially slow due to confusion with AJAX calls. The mockups and tutorials were able to be completed along with both dashboards without complete functionality.

Sprint 3 Report

Date: 10/9/2015
Attendees: Juan Machado, Ike Ukachi- Lois, Johann Henao
Discussed Topics:
In Sprint 3, the initial goals were to finish any incomplete stories from sprint 2 which was to complete functionality for both dashboards. Many tasks assign to each dashboard were completed but ultimately the Dashboards were still not fully functional. Tasks such as the ability to add, delete and modify information on the overall dashboard were completed, although the differentiation of admin and user was left unfinished. As for the current dashboard, the grading breakdown buckets still needed to be completed. The hurdles for the sprint included the lack of comprehension of what each function was supposed to do.

Sprint 4 Report

Date: 10/23/2015
Attendees: Juan Machado, Ike Ukachi- Lois, Johann Henao
Discussed Topics: As per the professor, there is a needs to be more focus on automation. There should be more progress on refinement and small details.

In Sprint 4, the initial goals were to finish any leftover stories from sprint 2 which was to complete the flaws left in both dashboards. Many tasks assign to each dashboard were completed but ultimately the Dashboards help feature was not fully functional. Tasks such as the ability to import, delete and export information from the students' perspective were worked on. As for the semester dashboard, the main functionality was done, but some bugs needed to be dealt with. The hurdles for the sprint were the same as the last and included the lack of comprehension of how certain functions were to be implemented.

Expectations for remaining sprints: expects refinement, pinpoint pitfalls. Exceed web design expectations. Show initiative. Work on documentation first before any implementation.

Sprint 5 Report

Date: 11/6/2015
Attendees: Juan Machado, Ike Ukachi- Lois, Johann Henao
Discussed Topics: In Sprint 5, the initial goals were to finalize the design for the database using the program curriculum as a guideline. We did accomplish a major goal in the creation and setup of referential constraints that allows the design to become much more sophisticated. Changes were also made to the naming conventions used, as this will improve readability for future developers.
In addition we were able to setup a help feature that will help guide the user when attempting to use the features offered on the Website.

Sprint 6 Report
expectations for sprint 7: work together, scrum(put in time), increase amount of results(velocity/efficiency)

Date: 11/20/2015
Attendees: Juan Machado, Ike Ukachi- Lois, Johann Henao
Discussed Topics: In Sprint 6, the initial goals were to refactor after the drastic change in the database design successfully leverage the what if report. this was to be done by parsing a pdf to obtain the student's grades on the transcript and have them stored in the database. this was a task that proved to be vital for the team's progression, as manual input of a student's grade is impractical.
The professor provided us with a possible solution by mentioning of simplifying the process, by asking the user for a link, but this proved to be something that bordered on privacy, so we decided not to go that route.
In this sprint, we received note that the speed of production was not where it was supposed to be and that we should start the next sprint with a different attitude.
Overall it was a successful sprint, with major accomplishments.

Sprint 7 Report

Date: 12/4/2015
Attendees: Juan Machado, Ike Ukachi- Lois, Johann Henao
Discussed Topics: An individual assessment and group assessment was given at the end of this sprint. Juan, showed great solutions in the features such as the pdf parser and also with the ideas implemented in the semester dashboard. He was able to have a good comeback in last sprint where he was slowed down by a few hurdles. He also showed better communication and followed up with tasks given to him. Ike showed great solutions with recursive child buckets and a help feature to guide uncertain users. he also showed better communication and followed up in the same way as Juan did, by responding to the task at hand. Both showed great responsibility in their roles, and showed tremendous improvement in technical and software engineering skills.

Appendix D - Sprint Retrospective Reports

Sprint 1 Retrospective

Date: 9/11/2015
Attendees: Johann Henao, Ike Ukachi- Lois, Juan Machado

Discussed Topics:
Explain what went well and not so well in this sprint. Include any agreed-upon actions to mitigate any issues for the next sprint.

In this sprint, we completed tutorials necessary for web development as well as the setup of the development environment. The landing page, database and basic functionality for login and registration was expected and mostly completed. upon completion of the sprint, there were no real issues this initial sprint, as we were merely learning the basics to build a solid foundation going forward.

Date: 9/11/2015
Attendees: All
Start time: 12:00PM
End time: 12:45PM
Minute Taker: All

Juan Machado:

- What is planned to be done to start the next sprint?
    - Split Epic into several User Stories

Ike Ukachi-Lois:

- What is planned to be done to start the next sprint?
    - Split Epic into several User Stories for sprint 2
    - Finish log in functionality
    - Continue Javascript tutorials

Sprint 2 Retrospective

Date: 9/25/2015
Attendees: Johann Henao, Ike Ukachi- Lois, Juan Machado

Discussed Topics:
Explain what went well and not so well in this sprint. Include any agreed-upon actions to mitigate any issues for the next sprint.

In sprint two, we were to come up with mockups for the website's pages and to begin coding the html for each page. Since we had been utilizing web development languages for a few weeks at this point, more was expected from us. We were able to create pages for potential users, with tables for the insertion data. The datatables proved to be a small hurdle and to work around this issue, the product owner Johann offered his help to resolve the problem.

Date: 9/25/2015
Attendees: All
Start time: 1:35PM
End time: 1:56PM
Minute Taker: All

Juan Machado:

- What is planned to be done to start the next sprint?
    - Add central Page to semester dashboard
    -

Ike Ukachi-Lois:

- What is planned to be done to start the next sprint?
    - Complete data tables for GPA Dashboard (US #706)
    -

Sprint 3 Retrospective

Date: 10/9/2015
Attendees: Johann Henao, Ike Ukachi- Lois, Juan Machado
Discussed Topics: Unfortunately, in sprint three we began with a slow start. the amount of productivity, did not meet expectations. The tasks were to create a dashboard for the students overall GPA as well as their currently enrolled classes. These are what represent the software solution for the project, and were vital its progression. In addition the daily meeting minutes were not specific enough. Lack of communication between team was a big concern.

Date: 10/9/2015
Attendees: All
Start time: 4:18PM

End time: 4:40PM
Minute Taker: All

Juan Machado:

- What is planned to be done to start the next sprint?
  - Focus on proper Documentation on Meeting Minutes.
  - Code to dynamically add data table to each tab
  - Ability Remove Each assessment/Tab
  - Click on datatable to remove grades and modify them (On the assessments Details

Ike Ukachi-Lois:

- What was done since the last scrum meeting?
  - Implemented GPA calculations for Overall GPA table
- What is planned to be done to start the next sprint?
  - Focus on proper Documentation for each User stories

Sprint 4 Retrospective

Date: 10/23/2015
Attendees: Johann Henao, Ike Ukachi- Lois, Juan Machado
Discussed Topics: In this sprint we saw improvements in the technical aspects. Our progress however was impacted due to overflow of work from sprint 3. The focus moving ahead was to aim for a faster rate of production while keeping the same quality and integrity of the system.

Date: 10/23/2015
Attendees: All
Start time: 1:30PM
End time: 1:55PM
Minute Taker: All

Juan Machado:

- What is planned to be done to start the next sprint?
  - Test all features (smoke test)
  - Prepared For Sprint review with professor
  - Worked on use case model documentation

Ike Ukachi-Lois:

- What is planned to be done to start the next sprint?
  - Work on Final Documentation (Use Case Models)
  - What is planned to be done to start the next sprint?
  - Finish Help feature
  - Modify/Delete Courses from master Course List.
  - Export Data to XML format (Admin)

Sprint 5 Retrospective

Date: 11/6/2015
Attendees: Johann Henao, Ike Ukachi- Lois, Juan Machado

Date: 11/6/2015
Attendees: All
Start time: 2:00PM
End time: 2:53PM
Minute Taker: All

Juan Machado:

- What is planned to be done to start the next sprint?
  - Finish the final database design
  - Refactor code to work with the new Database (SQL query)

Ike Ukachi-Lois:

- What is planned to be done to start the next sprint?
    - Worked on database design
    - Add Extra details to Mingle Stories
    - Finished final database design.
    - Refactored code for Overall GPA Dashboard and login.

Discussed Topics:

Sprint 6 Retrospective

Date: 11/20/2015
Attendees: Johann Henao, Ike Ukachi- Lois, Juan Machado
Discussed Topics:

Date: 11/20/2015
Attendees: All
Start time: 2:00PM
End time: 2:45PM
Minute Taker: All

Juan Machado:

- What is planned to be done to start the next sprint?
    - Working on Administrator Import
    - Completed manually creating the XML file that represents the Computer Science Major. And worked on functionality
    - From the xml, the courses are able to be input to the database, as well as buckets without parentID's.
    - The input of major Course buckets
    - Major Buckets That have with parentID

Ike Ukachi-Lois:

- What is planned to be done to start the next sprint?
    - Worked on the functionality of adding an unlisted course using the new column Student Course table.
    - Worked on visually displaying the child buckets.
    - Finish the adding of unlisted courses.
    - Work on displaying child buckets.
    -

Sprint 7 Retrospective

Date: 12/4/2015
Attendees: Johann Henao, Ike Ukachi- Lois, Juan Machado
Discussed Topics:

Date: 12/4/2015
Attendees: All
Start time: 2:00PM
End time: 2:43PM
Minute Taker: All

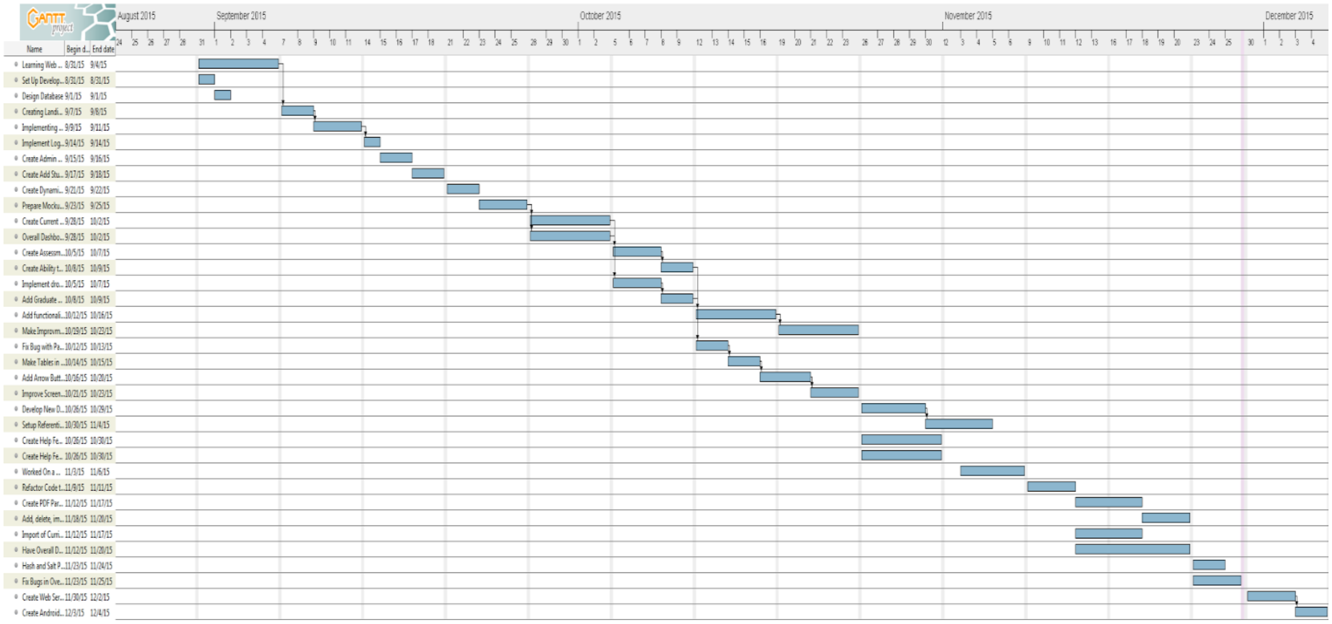
Juan Machado:

- What is planned to be done to start the next sprint?
    - Setting up web application in a separate thread
    - have a space between the list of courses and title.
    - on the grade for assessment put identifies for the course
    - have a plan B adroid project (points to jhenaoweb)
    - add an about page for app that displays a message
    - Add title to graph. make it less lonely.
    - Add password hashing solution for android
    - APP LOGOS

Ike Ukachi-Lois:

- What is planned to be done to start the next sprint?
    - Add error message to incorrect log in.
    - Admin help Feature
    - GPA History graph
    - Forecast Report
    - Logout

Appendix E - Gantt Chart



Name	Begin d...	End date
Learning Web ...	8/31/15	9/4/15
Set Up Develop...	8/31/15	8/31/15
Design Database	9/1/15	9/1/15
Creating Landi...	9/7/15	9/8/15
Implementing ...	9/9/15	9/11/15
Implement Log...	9/14/15	9/14/15
Create Admin ...	9/15/15	9/16/15
Create Add Stu...	9/17/15	9/18/15
Create Dynami...	9/21/15	9/22/15
Prepare Mocku...	9/23/15	9/25/15
Create Current ...	9/28/15	10/2/15
Overall Dashbo...	9/28/15	10/2/15
Create Assessm...	10/5/15	10/7/15
Create Ability t...	10/8/15	10/9/15
Implement dro...	10/5/15	10/7/15
Add Graduate ...	10/8/15	10/9/15
Add functionali...	10/12/15	10/16/15
Make Improvm...	10/19/15	10/23/15
Fix Bug with Pa...	10/12/15	10/13/15
Make Tables in ...	10/14/15	10/15/15
Add Arrow Butt...	10/16/15	10/20/15
Improve Screen...	10/21/15	10/23/15
Develop New D...	10/26/15	10/29/15
Setup Referenti...	10/30/15	11/4/15
Create Help Fe...	10/26/15	10/30/15
Create Help Fe...	10/26/15	10/30/15
Worked On a ...	11/3/15	11/6/15
Refactor Code t...	11/9/15	11/11/15
Create PDF Par...	11/12/15	11/17/15
Add, delete, im...	11/18/15	11/20/15
Import of Curri...	11/12/15	11/17/15
Have Overall D...	11/12/15	11/20/15
Hash and Salt P...	11/23/15	11/24/15
Fix Bugs in Ove...	11/23/15	11/25/15
Create Web Ser...	11/30/15	12/2/15
Create Android...	12/3/15	12/4/15

REFERENCES