**Collaborative Platform**

**Virtual Job Fair, Remote Judge and Mentoring Modules**

**Final Document**

Senior Project – CIS 4911 Section U01

4/23/2015

**Members:**

**Spring 2014**  **Instructor:**

Henry M. Romero Masoud Sadjadi

Steven S. Sanabria

Lorenzo Sanchez **Mentor:**

Juan Caraballo

**Summer 2014**

Ramon Gomez

Jonathan Santiago

**Fall 2014**

Nicholas Madariaga

Jonathan Sanchez

Spring 2015

Adrian Alfonso

Jorge Travieso

*This page was intentionally left blank.*

Copyright © Florida International University 2013. All Rights Reserved. No part of the Virtual Job Fair Project or documentation may be reproduced or modified without the express consent of Florida International University. Redistribution or commercial use is strictly prohibited.

# EXECUTIVE SUMMARY

The Final Deliverable of the Collaborative Platform is meant to provide the reader with all the necessary development information regarding the project. The Collaborative Platform is a website that connects students, employees, professors, and anyone else with a desire to ask questions and learn. The website provides the availability to quickly pose a question, with the system sending automated alerts, allowing for an immediate response. The task of the Collaborative Platform is to add a mentoring module in which there will be three types of mentors that can actively interact with the community of mentees. This will bridge the gap of slow and unreliable information often found online or through waiting at professor’s office hours. Within this document, detailed system design, object design and overall implementation is analyzed and described.

**Table of Contents**

**EXECUTIVE SUMMARY**

1. Introduction

1.1 Problem definition.

1.2 Scope of system.

1.3 Overall development methodology

1.4 Definitions, acronyms, and abbreviations

1.5 Overview of document

2. Feasibility Study

2.1 Description of current system. Identify limitations and constraints

2.2 Description of alternative solutions considered.

2.3 Recommendation with explanation of why the solution was selected.

3. Project Plan

3.1 Project Organization

3.1.1 Project Personnel

3.1.1 Project Personnel Organization

3.1.2 Hardware and Software Resources

3.2 Identification of Tasks, Milestones and Deliverables

3.3 Cost of the Project

4. System Requirements

4.1 Functional and Nonfunctional Requirements

4.2 Requirements Analysis

4.2.1 Scenarios

4.2.2 Use Case Model

4.2.2 Static model

4.2.3 Dynamic model

5. System Design

5.1 Overview

5.2 Subsystem Decomposition

5.3 Hardware and Software Mapping

5.4 Persistent Data Management

5.4.1 Data Dictionary

5.5 Security/Privacy

6. Detailed Design

6.1 Overview

6.2 Static model.

6.3   Dynamic model

6.4 Code Specification

7. System Validation

7.1 System Tests

7.2 Subsystem Test

7.3 Evaluation of Tests

8. Glossary

9. Appendix

9.1 Appendix A - Project schedule (Gantt chart or PERT chart).

9.2 Appendix B - All use cases with nonfunctional requirements

9.3 Appendix C – User Interface Design

9.4 Appendix D – Analysis Model

9.5 Appendix E – Design Model

9.6 Appendix F - Documented Class interfaces and constraints

9.7 Appendix G – Documented code for test drivers and stubs

9.8 Appendix H – Diary of meetings and tasks for the entire semester

4.3 Appendix C – Cost Matrix

4.4 Appendix D – Diary of Meetings

10. References

# Introduction

      The introductory chapter provides a brief overview of what the Collaborative Platform entails. First, the problem is stated: helping reduce time constraints between student and professor interaction by allowing fully remote questioning and with time, meetings. The second section covers the ideal scope of the system, followed by the process model used in developing the Collaborative Platform. Definitions, acronyms, and abbreviations follow, with the last section covering a more general overview on what is to be expected for the entirety of this document and project.

## Problem definition.

In the world today, for prospects of different companies, a common issue arises that could prevent them from getting a chance. From startups to large corporations, money can always be an issue. This can play a pivotal part in the hiring of future employees. The chance to be flown out of town to do an onsite visit can never happen if the budget doesn’t suffice for said company. With the Virtual Job Fair, the idea was to allow an interface for interviewers to communicate with potential new hires. At the moment, the virtual job fair is in need of some revamping. The plan will be to recast it within a collaborative platform along with the remote judge and mentor modules.

Professionals all over the world currently working in industry know the value of time. The senior project demonstrations normally bring many of these professionals to FIU to judge and grade students work. Many, however, due to time constraints or other simple logistics, cannot make their way onto FIUs campus to judge face to face. The solution to this problem could greatly benefit future students to get the most elaborate and complete reviews for their work. As it stands, for this senior project course, judges have to physically come to FIU in order to participate in the judging aspect for each student. We hope to integrate the current mobile judge application into the collaborative website, facilitating feedback retrieval from professionals and guests that may not be able to get to FIU.

Lastly, a key integration that is hoped to be built for the platform is a mentoring module. One of the key drawbacks of forum based assistantship/tutorials is a severe lack of responsiveness and/or initiative. One of the more urgent problems is the sometimes sluggish back and forth messaging with assigned mentors. The optimistic solution to this problem will be a platform that has a forum in which each user can interact in a plethora of ways; whether through video live chat, white board brainstorming, file sharing or simple messaging.

From the mentors’ perspective, the problem lies with time constraints in being able to effectively give back feedback or answer questions to mentees. They are also faced with the challenge of distance, in some cases they may not be able to meet face to face with their assigned mentees. The mentoring module will propose a solution that allows mentees to ask questions at their will, and have a sufficient amount of mentors that can answer these questions. This will facilitate the interaction between mentors and mentees via the internet – allowing the mentors to remotely answer questions without having to be there in person.

## Scope of system.

The system will allow for an integration of three modules into one single collaborative platform. The system will allow users to register and log into the site and pick their main functionality. In some cases users can have various roles and thus will have multiple views available within the site. Each user will have a profile page and depending on their role, they will have a more detailed interaction for their account(s). For example, within the mentoring module there are three mentors: domain, project and personal. All types of mentors will be allowed to sync their profile with LinkedIn in order to upload a picture and provide a short bio, as well as any other related information.

The project mentor can see all of the projects they are mentoring and will have the ability to screen share, use a whiteboard, and other functionality that will be taken from the other two modules and be integrated within the mentoring module. They also have the ability to grab the project information directly from the senior project website. The domain mentors are mentors of specific skills. They will have the ability to add new domains and rate themselves accordingly. These additions will be approved or denied by the administrator before being updated on their profiles. They will have the functionality to see all tickets assigned, whether answered, dropped, or pending. From here, they can answer the questions, append comments, and in some cases they can delegate it to all other tier 1 mentors of that domain. They can also communicate within the internal mail service to other mentors.

The system will match the tickets created by mentees to the most appropriate domain mentors – those with the most expertise in that specific skill. If the system notices the inactivity on one of the ticket and notices that the last comment added to the ticket was not made by the assigned mentor, it proceeds to rank it to another mentor in the same tier. If this situation happens two more consecutives times for the same ticket, the system will have automatically notified the administrator for immediate resolution. Besides being able to append comments to tickets, the mentees will be able to attach files. Once these tickets have been answered, the mentees will be allowed to close the ticket or reject the answer provided. Another module of the platform is the remote judge. This will be an integration of the mobile app into a remote desktop available interface.

Within the remote judge the system will allow judges to properly judge a senior project student. This will be facilitated by allowing access to the students resume, a grading rubric, and a virtual whiteboard to express ideas or questions. The system will also have the ability to share screens between the judge and the student.

The system will allow self-registration for users and provide self-serve mentorship applications. These applications will provide options for manually selecting preferred choices and the option to defer selection to the system. The system will also allow for the recommendation of new domains. Submitted applications will be vetted by an admin. The admin will be able to approve/deny what he chooses and also send counter-offers back to the mentor.

The system will allow reporting mentees, mentors and tickets. Those reports will be available to the administrator in order to gather as much insight of the data as possible. Additionally the system will provide the administrator with a way of making a quantitative analysis of the utilization of the system via a dashboard. The utilization dashboard will include the following reports: Amount of tickets created, Amount of tickets closed, Average duration of a ticket, Average time a mentor takes to answer a ticket, Amount of tickets currently open, Amount of tickets that are currently closed. Also, as part of the analysis, the system will be able to analyze and pull the most frequent type of questions that mentees may ask.

In order to facilitate the interaction mentor-mentee, the system will support video conferences where the mentor and mentee can meet face to face and collaborate with tools such as screen sharing, whiteboard and chat. Those video conferences can be scheduled to take place instantly or in the future and it may include several participants.

## 1.3 Overall development methodology

We have used the unified software process model as the basis for our planning and organization during the design of our software. The model is largely based on the construction of the specific use cases to help formulate and drive the design process. After formulating a set of uses cases, the team can then take those specific use cases and apply them to different UML diagrams to flesh out the design process. For example, class diagrams can show the relationships between different classes within subsystems and how specific subsystems can relate to one another. Sequence diagrams help plan out specific paths of method calls before the team actually begins to code. Finally having carefully specified requirements document allows the team to stay focused and on task as to the purpose of the design.

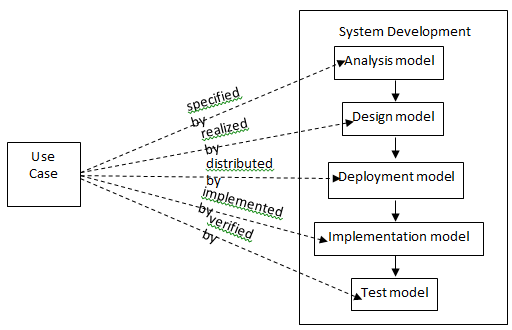


Figure 1.3. Unified Software Process Model

## 1.4 Definitions, acronyms, and abbreviations

Definitions:

* **User** – A generalization given to all user groups.
* **Student** – Individual who is currently enrolled in the senior project class.
* **Project** **Mentor** – An individual who is assigned to overlook one or many senior projects.
* **Domain** **Mentor**– An individual with particular knowledge in a certain skill or language(2 Tiers)
* **Personal Mentor** –An individual that mentors only a single person – not limited to career advice.
* **Mentee** - An individual that may request help or ask a question.
* **Administrator** –An individual assigned with administering the web portal.
* **Interviewer** – An individual from industry that can remotely interview potential employees.
* **Interviewee** – An individual and future prospect that may engage in the virtual job fair portal.
* Application - An application is basically the form that is used by a mentor to designate what selections they would like to be considered form. The admin uses the application to approve/reject those selections.

Acronyms:

* **SCIS** – School of Computing & Information Sciences
* **FIU** – Florida International University
* **ECS** – Engineering & Computing Sciences (Building)
* **JCCL** – John C. Comfort Laboratory
* **VJF** – Virtual Job Fair
* **MJ** –Mobile Judge
* **WP** – Web Platform
* **T1**- Tier 1 Domain Mentor
* **T2**- Tier 2 Domain Mentor
* **PM**- Project Mentor
* **DBMS** – Database Management System

Abbreviations:

* **N/A**

## 1.5 Overview of document

This final document is divided into several sections, each defining a major purpose in the software lifecycle. Following this section, in section 2, the feasibility study is found. Here sections cover a general overview on possible solutions, description of the current system and recommendations for implementation. Chapter 3 covers the main project plan which outlines the requirements, our deadline for milestones, cost, and general breakdown of responsibilities within the system construction. Chapter 4 goes over the general system requirements, as well as the use case model, and important scenarios to correspond with our functional requirements.

Chapter 5 covers the system design and subsystem decomposition. This includes the hardware and software mapping, as well as the general database design and representation for persistent data. Chapter 6 breaks down the system into higher levels of design; mainly into our general subsystems. Here, each of our subsystems are detailed and described. Chapter 7 covers our system validation to ensure that our integrated work and individual subsystems work independently, as well as together. System tests ensure that our functionality is kept without compromising requirements. Chapter 8 is a simple overview of our glossary of terms to become familiar with within this document. Chapter 9 covers the main appendices in which many of our diagrams can be found. Chapter 10 is our references section for when we draw information from outside sources to be cited.

# Feasibility Study

The Feasibility Study, known as chapter 2, covers the general ideas of this collaborative platform from a systematic view. They are covered in detail, beginning with the description of the systems, their limitations and constraints from previous versions, and what will be brought into play with the new revisions with the web platform. This also includes their integration plan, purpose, and ideal improvements for each feature that may be ascertained by the end of this project. Lastly, in this chapter you will find alternative solutions to currently proposed problems.

## 2.1 Description of current system. Identify limitations and constraints

The Virtual Job fair provides the interaction between students and employers. The students are able to upload their resumes and search for possible companies where they can apply for a job. Employers can look up possible candidates for a specific position. Also, employers can set up virtual interviews using the video chat and whiteboard features. There is no limitation and constraints on the current system, except for some features that need to be readjusted.

Referring to remote judge, what we have now is an implementation of a grading system that works only in mobile applications. The idea is, at the end of the spring project, to have this system fully integrated into the collaborative platform. As of now, judges are unable to grade students remotely and in order to watch a student present their project they have to be in that exact location. There is also a limited amount of information in the mobile judge application for the judge to obtain information on the student being evaluated.

The Collaborative Platform: Mentoring Module currently allows students, known as mentees, to interact with mentors who can be Personal Mentors, Project Mentors, or Domain Mentors. This interaction is through tickets which are entered into the system by students. These tickets are initially sent to tier 1 domain mentors. If the ticket is not handled within the time limit the system will automatically reassign the ticket to another tier 1 domain mentor. A tier 1 domain mentor with a pending ticket has the option to defer the ticket to a specific tier 1 or tier 2 mentor or defer the ticket to the system for automatic reassignment. The system allows Mentors to see what tickets belong to them depending on the role that they have in the system (personal, project, domain). The limitation that the platform currently has is that there is no self-registration for mentors. The admin has to manually create an account to add every mentor to the system. The admin dashboard currently has no way to infer information on the usage of the system. The admin also does not have a way to go through applications and approve/reject mentors.

## 2.2 Description of alternative solutions considered.

**Web Application Framework**

**1.** CakePHP, is an open source web application framework that was designed following the concepts of Ruby on Rails. It became public in 2006.

**2.** Yii, is a component-based framework which uses MVC software architecture in an object oriented environment.

**3.** Codeigniter, is an open source rapid development web application framework in building dynamic web sites with PHP.

**Data Base Management System**

**4.** PostgreSQL, is an open-source Object-Relational DBMS with emphasis on extensibility and standards-compliance.

**5.** Microsoft SQL Server 2012, is a relational DBMS developed by Microsoft.

**6.** MySQL it is the second most widely used open-source relational DBMS.

**Ticket System - Workflow**

**9.** Osticket, an open source ticket system that can route inquires, created online or through forms into a web platform. This allows easy creation of tickets and a base to route mentors and mentees. It is written in PHP and allows connectivity with MySQL.

**10.** eTicketSupport, a PHP based open source support ticket service that also allows a simple helpdesk type platform. This would facilitate the interaction of tickets created on the platform, while maintaining continuity of languages.

**11.** Write the flowTicket from scratch. These will be the last choice if any of the other fits with the client requirements.

## 2.3 Recommendation with explanation of why the solution was selected.

What we recommend to implement in our Collaborative Platform system is a consequence of all the comparison charts from the previous section. The selection of any software to conduct our work follows the higher rating established among similar software after analyzing the pros and cons.

**Framework implementation**:  In order to continue with the same workflow of the two previous virtual job fair projects and in order to create continuity in previously used technologies we recommend to keep using Yii Framework because of its MVC architecture model and it will adapt perfectly to the mentoring and remote judge modules. Nonetheless, because of the better rating of CodeIgniter the mentoring platform will be developed on this framework.

**Database development:** Given its elevated rating after analysis and because it has been used without any relevant issues on previous projects of Job Fair, Remote Judge and Senior Website, MySQL will be our election for the database design in the elaboration of our mentoring platform. This decision will also help to eliminate any compatibility issues that could arise during the process.

**Implement the Flow Ticket from Scratch**: At first we considered the OS ticket as one of our first alternative of solutions, but after get the requirements and posterior analysis of them, we decided to implement the flow ticket from the scratch because none of the open source flow will allow us to implement the flow ticket as per client request.

# Project Plan

This section describes the project plan which includes the project organization, in the software development process. The section 3.1.1 offers the project personal organization and the section 3.1.2 includes the hardware and software resources. Lastly, section 3.2 includes the identification of task, milestone and deliverables with a specific work breakdown and cost estimates.

## 3.1 Project Organization

For this project, each of three members will be in charge of adding at least two new pieces of functionality to the system.

Henry D. Muniz Romero will be responsible for developing an Internal Mail Service to use within the platform. In addition, will be responsible for the system notifications that keep the user in sync with what is happening in the platform. Finally, he will develop an administration menu for the System Administrator to manage users, invitations, domains, subdomains, and projects.

Lorenzo A. Sanchez My will be responsible to implement some functionalities of the mentoring subsystem such as allows users to create tickets, upload files for a ticket, append comments to a ticket, allow a project mentor to set up meeting with their respective mentees, allow administrator to monitor the upcoming meetings between project mentors and mentees. Also, write an algorithm that allows the system auto assign a ticket to the most appropriate domain mentor expert in the field based on domain specified in the ticket and mentor availability.

Steven S. Sanabria will be responsible for the user access point and registration subsystem. This includes the log in, log out and registration into the collaborative platform site. In addition, he is responsible for the user profile management; this includes allowing users to set up their availability, which will be vital in determining their allowable involvement. Domain mentors will be able to add domains and rate themselves for each skillset. Project mentors and Personal mentors will be able to select projects and mentees, respectively. Lastly, administrators will also be allowed to edit each user’s profile – as desired.

Ramon Gomez will be responsible to implement some functionalities of the mentoring subsystem such as allow domain mentors in tier 1 to escalate tickets when the question is beyond of what a tier 1 domain mentor should know. Also add priorities in the tickets. In addition, he is responsible to allow the system to automatically reassign the tickets when they are not answered based in its priority. Also, Ramon is responsible to allow mentee to assign tickets to his/her personal mentor and allow the personal mentors to make comment in tickets created by his/her mentees. Also Ramon is going to add new notifications to the system, for example new notifications are going to be sending when the ticket is closed, escalated or automatically reassigned.

Jonathan Santiago is responsible to integrate CP with SPW. He will be bringing projects and mentees (triggered automatically and manually) from SPW to CP and relate them accordingly. Also he will give the mentees the ability to login to CP using their FIU Google credentials just like they do in SPW. In addition Jonathan will improve the registration process by allowing the admin to register all types of types: project, domain and personal. The admin will also have the ability to edit the roles of any user. Jonathan will also be involved in improving the user interface in multiple sections of the system.

Nicholas Madariaga is responsible for developing the self-serve account registration and Mentorship Applications. A new access point will be provided for potential Mentors. The new registration form will store more the work experience and education of the Mentor. The mentorship applications will involve the different steps and layouts for each mentor role that is defined: project, personal, and domain expert. The Project Mentor Application provides detailed information on projects and the ability to select what projects the mentor is interested in or have them automatically assigned by the system. The Personal Mentor Application involves basic information and the ability to select what students he does wish to mentor and provides the option to defer selection to the system. The Domain Mentor Application will provide details of each domain and the ability to select what domains and subdomains they’re more knowledgeable in along with the ability to rate their proficiency. Client side validation and user verification will be provided for user registration and all Mentor Applications. Lastly, he will implement a way to handle any mentorship proposals sent by an admin. He will also be involved with fixing miscellaneous bugs that are found on the module. Throughout development User Experience is a primary focus. He will also have the role of being the tester and project manager for Jonathan Sanchez.

Jonathan Sanchez is responsible for an overhaul of the admin dashboard and new features - Users, Domain, Projects, Tickets, Invitation, and Application pages. The Users page will just get a look overhaul and an advanced search feature. Users will be clickable where a popup modal will show details relating to the user. The Domain page will be merged into one single page with a domain subsection and a subdomain subsection. Domains will be clickable where a popup modal will show details relating to the domain. The Projects page will have the same look as well and not much will be done beyond look update. Projects will be clickable where a popup modal will show details of project. The invites page will show any open invitations and clicking on an invite will show details as well as re-invites. There will be an addition to the “Manage” drop down; we will be adding “Tickets” and “Applications”. The tickets page will have a list of all tickets and an advanced search function. The Applications page will show any pending applications requiring the action of an admin. The visual aspects of these updates will primarily be focused on improving the User Experience and making things more intuitive. He will also have the role of being the tester and project manager for Nicholas Madariaga.

### 3.1.1 Project Personnel

This section will explain individual roles and assignment of responsibilities among team members. It will also give a detailed explanation for software and hardware components required to carry out the development of the system.

### 3.1.1 Project Personnel Organization

This section describes the assignment of roles during the project

|  |  |  |
| --- | --- | --- |
| **Name** | **Roles** | **Assigned task based on functional requirements** |
| Lorenzo Sanchez | Developer / Team Leader | **Mentoring Subsystem.**   * Auto assign the most appropriate domain mentor to a ticket. * Allow administrator to reassign the ticket to other domain mentor. * Allow project mentor to retrieve the description of the projects assigned. * Allow project mentor to retrieve all the mentees for each project. * Allow project mentor to set up meetings with his/her mentees. * Allow project mentor to retrieve all the upcoming meetings. * Allow project mentor to assign ticket to other project mentor. * Allow project mentor to see the ticket created by their mentees. * Allow mentees to assign tickets to his/her project mentor. * Allow user create a ticket. * Allow user to select a specific domain for a new ticket. * Allow user to select a specific sub domain. * Allow user to upload file to a ticket. * Allow user to download file from a ticket. * Allow user to append comments to a ticket. * Allow user to retrieve all the details of the ticket of the ticket that he/she created or was assigned. * Allow user to close a ticket. * Allow user to reject a ticket. |
| Henry D. Muniz Romero | Developer / DB Admin | **Communication & Notification Subsystem**   * Allow users to send a message to another user. * Allow users to read a message. * Allow users to delete a message. * Send ticket due email notification. * Send new message email notification. * Send password change email notification. * Send profile change email notification. * Send ticket assigned email notification. * Send comment added to a ticket email notification.   **Mentoring Subsystem.**   * Allow System Administrators (SA) to create new administrators. * Allow SA to change user profile information and scope. * Allow SA to manage Domains and Sub-Domains. * Allow SA to disable and enable users. * Allow SA to send invitations * Allow SA to manage projects |
| Steven Sanabria | Developer / Web Master | **Registration and Access Point Subsystem**   * Allow the user to register. * Allow the user to log in. * Allow the user to log out. * Allow the user to retrieve forgotten password.   **Mentoring Subsystem**   * Allow users to edit their availability * Allow users to edit their photo * Allow project mentor to select desired projects * Allow personal mentor to select desired personal mentees * Allow domain mentors to add domain and expertise ratings * Allow administrators to view and edit user profiles |
| Ramon |  | **Mentoring Subsystem**   * Allow mentees to assign tickets to his/her personal mentor. * Allow personal mentor to see the ticket created by their mentees. * Allow personal mentor to comment the ticket created by their mentees. * Allow Domain mentor in tier 1 to escalate the ticket to a Domain Mentor in tier 2 * Allow user to select a specific priority for a new ticket. * Allow administrator to change the priority waiting time. * Allow the system to automatically reassign the tickets.   **Communication & Notification Subsystem**   * Send ticket closed email notification. * Send ticket not available mentor email notification in the Automatic Reassign. * Send ticket reassigned email notification to admin when the ticket was automatically reassigned three times. * Send ticket escalated email notification. |
| Jonathan Santiago |  | **Integration with SPW**   * Allow admin to import data from SPW manually * Allow the system to import data(students/projects) from SPW automatically * Allow senior project students to login with their FIU Google login credentials * Allow system to update reassign projects based on changes made in SPW * Allow system to update reassign mentors based on changes made in SPW   **Registration**   * Allow the admin to register mentors (project/domain/personal) * Allow admin to edit mentors roles * Allow mentors to login with credentials provided by admin |
| Nicholas Madariaga | Developer/  Tester/  Project Manager | **Self Serve Mentor Registration**   * Provide a separate access point for users without accounts * Allow users to register for an account themselves * Allow users to provide information about education and work experience * Allow users to apply for a Personal Mentorship * Allow users to apply for a Project Mentorship * Provide detailed information about mentees, projects and domains within their respective applications * Allow users to select projects and mentees manually or have the system select for them based on preferences * Allow users to apply for a Domain Mentorship * Allow users to set their proficiency within a Domain * Allow users to recommend new domains * Provide user verification for all appications * Provide client side validation for all forms   **Mentor Dashboard**   * Allow mentors to apply for new mentorships from and all-in-one Mentor Application portal * Allow mentors to begin their mentorships whenever approved by an admin * Allow mentors to look over counter-offers made by an admin * Allow mentors to accept or deny offers made by an admin |
| Jonathan Sanchez | Developer/  Tester/  Project Manager | **Admin Dashboard**   * Allow admin to view pending applications * Allow admin to invite Mentors through email * Allow admin to view open invitations/re-invites * Allow admin to send a re-invite * Allow admin to gain insight on site usage through admin dashboard * Allow admin to gain insight on a particular project * Allow admin to gain insight on a particular user * Allow admin to gain insight on a particular domain * Allow admin to gain insight on a particular subdomain * Allow admin to view tickets * Allow admin to manage tickets * Allow admin to use an advanced search for tickets. * Allow admin to customize the body of the message while sending an invite   **Admin Approval**   * Allow admin to approve/reject Projects in a Project Mentor Application * Allow admin to propose Projects in a Project Mentor Application * Allow admin to approve/reject Mentees in a Personal Mentor Application * Allow admin to propose Mentees in a Personal Mentor Application * Allow admin to approve/reject Domains in a Domain Mentor Application * Allow admin to propose Domains in a Domain Mentor Application |

### 3.1.2 Hardware and Software Resources

**Hardware Requirements:**

* 6 personal computers (Maverick OS, Windows 7 and Ubuntu 10.1)
* 1 Subversion Server (located at FIU)

**Software Requirements:**

* Apache v.2.2.15
* MySQL v5.0
* PHP v5.3.3
* phpMyAdmin
* phpMyStorm
* Netbeans
* Yii Framework
* Linux
* Google Chrome 30.8
* StarUML
* GitLab

## 

## 

## 3.2 Identification of Tasks, Milestones and Deliverables

Outlined below are the milestones required for the development of this web platform.

**Version 1.0**

|  |  |  |
| --- | --- | --- |
| **Checkpoint** | **Description** | **Date** |
| **Milestone 1** | Feasibility and Project Plan | 2/3/2014 |
| **Milestone 2** | Requirements Document | 2/17/2014 |
| **Milestone 3** | Design Document | 3/3/2014 |
| **Milestone 4** | Testing Requirements | 3/31/2014 |
|  | **Implementation & Unit Testing** | 3/17/2014 |
|  | **Integration & System Testing** | 3/31/2014 |
| **Milestone 5** | Poster Due Date | 4/14-25/2014 |
|  | **Group Presentation** | 4/21/2014 |
|  | **Final Showcase** | 4/25/2014 |

**Version 2.0**

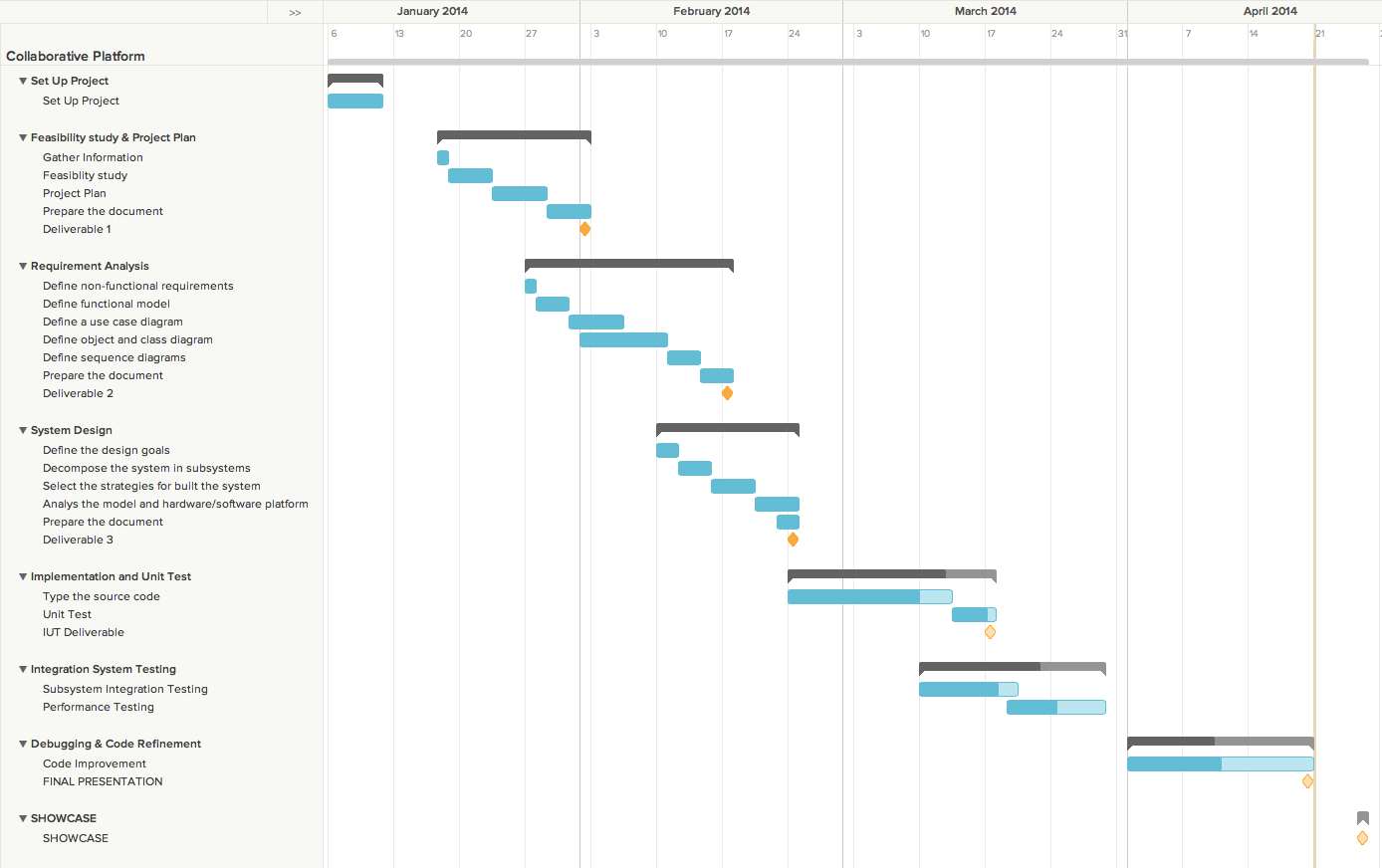
|  |  |  |
| --- | --- | --- |
| **Checkpoint** | **Description** | **Date** |
| **Milestone 1** | Feasibility and Project Plan | 7/7/2014 |
| **Milestone 2** | Requirements Document | 7/10/2014 |
| **Milestone 3** | Design Document | 7/12/2014 |
| **Milestone 4** | Testing Requirements | 7/13/2014 |
|  | **Implementation & Unit Testing** | 7/13/2014 |
|  | **Integration & System Testing** | 7/13/2014 |
| **Milestone 5** | Poster Due Date | 7/18/2014 |
|  | **Group Presentation** | 7/25/2014 |
|  | **Final Showcase** | 7/25/2014 |

**Version 3.0**

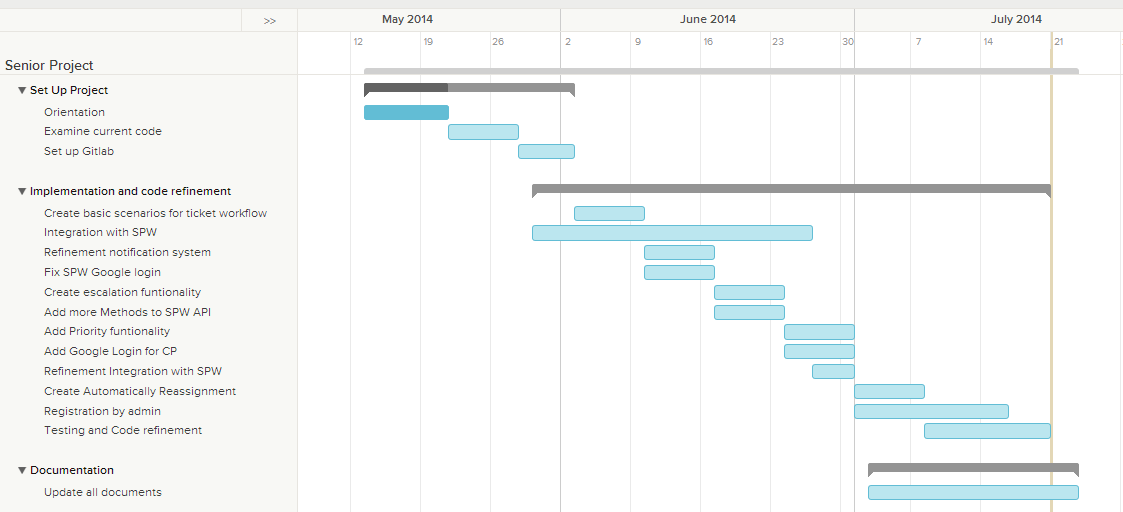
|  |  |  |
| --- | --- | --- |
| **Checkpoint** | **Description** | **Date** |
| **Milestone 1** | Feasibility and Project Plan | 12/10/2014 |
| **Milestone 2** | Requirements Document | 12/10/2014 |
| **Milestone 3** | Design Document | 12/10/2014 |
| **Milestone 4** | Testing Requirements | 12/10/2014 |
|  | **Implementation & Unit Testing** | 12/10/2014 |
|  | **Integration & System Testing** | 12/10/2014 |
| **Milestone 5** | Poster Due Date | 12/08/2014 |
|  | **Group Presentation** | 12/12/2014 |
|  | **Final Showcase** | 12/12/2014 |

The following GANTT chart contains the project schedule for the semester:

**Spring 2014**



**Summer 2014**



**Fall 2014**

## 

## 

## 3.3 Cost of the Project

The following feasibility matrix represents an estimate of the items and labor required for the project.

|  |  |
| --- | --- |
| **Feasibility Criteria** | **Description** |
| Operational feasibility | The proposed system will solve all the issues encountered during the system requirements process. So, the new system is operationally feasible. |
| Technical feasibility | The collaborative platform is a system that it will be developed for the computer science department. So, all the resources needed to develop the system are available for the team. |
| Schedule feasibility | The project is very well spread out according to the academic calendar of spring 2014. Team members and our mentor agree that the amounts of the requirements are enough to be completed within the given timeframe. |
| Economic feasibility | The team will not spend money to develop the project, because it is not required to pay for software (since it is open source) and hardware (supplied through personal computer or FIU Lab). |

|  |  |
| --- | --- |
| **Item** | **Item Cost** |
| Hardware (Personal Laptops) | $ 0.00 |
| Hardware (FIU Lab) | $ 0.00 |
| Software (Open Source) | $ 0.00 |
| Total Cost | $ 0.00 |

# System Requirements

The proposed system is Virtual Job Fair 2.0, which would expand upon the already-implemented Virtual Job Fair. The main idea behind Virtual Job Fair 2.0 is to add new dynamic functionality to the interview process in order to make it more dynamic. This chapter will introduce the functional and non-functional requirements of the system and the requirements analysis phase of the system.

## 4.1 Functional and Nonfunctional Requirements

**Mentoring Subsystem**

* Auto assign the most appropriate domain mentor to a ticket.
* Allow administrator to reassign the ticket to other domain mentor.
* Allow project mentor to retrieve the description of the projects assigned.
* Allow project mentor to retrieve all the mentees for each project.
* Allow project mentor to set up meetings with his/her mentees.
* Allow project mentor to retrieve all the upcoming meetings.
* Allow project mentor to assign ticket to other project mentor.
* Allow project mentor to see the ticket created by their mentees.
* Allow mentees to assign tickets to his/her project mentor.
* Allow user create a ticket.
* Allow user to select a specific domain for a new ticket.
* Allow user to select a specific sub domain.
* Allow user to upload file to a ticket.
* Allow user to download file from a ticket.
* Allow user to append comments to a ticket.
* Allow user to retrieve all the details of the ticket of the ticket that he/she created or was assigned.
* Allow user to close a ticket.
* Allow user to reject a ticket.
* Allow System Administrators (SA) to create new administrators.
* Allow SA to change user profile information and scope.
* Allow SA to manage Domains and Sub-Domains.
* Allow SA to disable and enable users.
* Allow SA to send invitations
* Allow SA to manage projects
* Allow user to edit their availability
* Allow user to edit their photo
* Allow project mentor to select desired projects
* Allow personal mentor to select desired personal mentees
* Allow domain mentors to add domain and expertise ratings
* Allow administrators to view and edit user profiles
* Allow mentees to assign tickets to his/her personal mentor.
* Allow personal mentor to see the ticket created by their mentees.
* Allow personal mentor to comment the ticket created by their mentees.
* Allow Domain mentor in tier 1 to escalate the ticket to a Domain Mentor in tier 2
* Allow user to select a specific priority for a new ticket.
* Allow administrator to change the priority waiting time.
* Allow the system to automatically reassign the tickets.

**Registration and Access Point Subsystems**

* Allow the user to register.
* Allow the user to log in.
* Allow the user to log out.
* Allow the user to retrieve forgotten password.

**Communication and Notification Subsystem**

* Allow users to send a message to another user.
* Allow users to read a message.
* Allow users to delete a message.
* Send ticket due email notification.
* Send new message email notification.
* Send password change email notification.
* Send profile change email notification.
* Send ticket assigned email notification.
* Send comment added to a ticket email notification.
* Send ticket closed email notification.
* Send ticket not available mentor email notification in the Automatic Reassign.
* Send ticket reassigned email notification to admin when the ticket was automatically reassigned three times.
* Send ticket escalated email notification.

**Integration with SPW**

* Allow admin to import data from SPW manually
* Allow the system to import data(students/projects) from SPW automatically
* Allow senior project students to login with their FIU Google login credentials
* Allow system to update reassign projects based on changes made in SPW
* Allow system to update reassign mentors based on changes made in SPW

**Registration**

* Allow the admin to register mentors (project/domain/personal)
* Allow admin to edit mentors roles
* Allow mentors to login with credentials provided by admin

**Mentor Module: Version 3**

The system shall...

* Allow a potential mentor to register for a mentor account from a new landing page.
* Allow a Project Mentor to apply by selecting projects they like or deferring the decision to the system.
* Allow a Personal Mentor to apply by selecting students they like or deferring the decision to the system.
* Allow a Domain Mentor to apply by selecting topics they’re proficient in.
* Allow a Domain Mentor to suggest new domains.
* Allow Project Mentor to approve/reject projects suggested by admin.
* Allow Domain Mentor to approve/reject topics suggested by admin.
* Allow Personal Mentor to approve/reject mentee suggested by admin.
* Allow admin to view pending applications
* Allow admin to approve/reject Projects in a Project Mentor Application
* Allow admin to propose Projects in a Project Mentor Application
* Allow admin to approve/reject Mentees in a Personal Mentor Application
* Allow admin to propose Mentees in a Personal Mentor Application
* Allow admin to approve/reject Domains in a Domain Mentor Application
* Allow admin to propose Domains in a Domain Mentor Application
* Allow admin to view open invitations/re-invites
* Allow admin to customize the body of the message while sending an invite
* Allow admin to gain insight on a particular project
* Allow admin to gain insight on a particular user
* Allow admin to gain insight on a particular domain
* Allow admin to gain insight on a particular subdomain
* Allow admin to view tickets
* Allow admin to manage tickets
* Allow admin to use an advanced search for tickets.

## 4.2 Requirements Analysis

Analysis models – contains the complete functional specification and is mainly for the designers and programmers. This section describes the diagrams in the Appendices B - D and validates the models against the use cases.

### 4.2.1 Scenarios

**Registration:**

Mark is a computer science professor at Florida International University. He specializes in teaching data structures and algorithm techniques courses in Java. Every day Mark gets questions from various computer science students in the senior project class, specifically related to Java and algorithms. After responding to some students via email and office hours he realizes that he needs to find a faster and more efficient way to reply. The next day he decides to stop by the office of the professor for the senior project course, Dr. Masoud Sadjadi. He talks to Masoud and asks if there is a simpler way to address the questions to his students. Masoud sends an email to Mark so that he can register within the collaborative platform. Mark clicks on the link through his email and goes to the registration page on the website.

Mark fills in his information, email address, password and other necessary credentials. Below he sees a section asking him to request a role within the website. The roles he sees are employer, interviewee, and administrator, as well as project, domain, and personal mentors among others. While hovering over each role he sees a short description of expected contribution and responsibilities. Remembering his discussion with Masoud regarding possible roles, he clicks to request the position of a domain mentor. From here Mark is requested to fill out his potential profile.

**Integration with SPW**

**Automatic Sync CP with SPW**

John is a student registered in Senior Project class. In the first day of class Professor Masoud Sadjadi advises students to choose a project using the Senior Project Website. John is able to login to SPW using FIU Google credentials (he is in the flat file used by SPW for validation) and chooses a project. The same day at midnight an automated routine brings students and projects (both associated with each other) from Senior Project Website to Collaborative Platform. The next day Masoud Sadjadi advises students that they can start using Collaborative Platform to communicate with their project mentors and also they’ll be able to post general questions for domain mentors to answer. John goes to the Collaborative Platform website and is able to login using his FIU Google credentials. John is now able to use Collaborative Platform.

**Changes in SPW reflect in CP**

For some reason John has changed his project in SPW He logs in to CP and he sees he is still associated with his old project. The same day at midnight a routine brings new students and projects as well as updates from SPW to CP. The next day John logs in to CP using his FIU Google credentials and he goes to hi profile and sees he now associated with his new project and consequently with his new project mentor.

**Manual Sync CP with SPW**

For some reason the automated routine that syncs CP with SPW failed and the students need to start using the platform immediately. Professor Sadjadi logs in using his administrator credentials. On his home screen he clicks on a button labeled “Sync with SPW” which trigger the routine that migrates data from SPW to CP. Now the students are able to login to CP and they are associated with the projects they chose in SPW.

**Registration**

Mark is a computer science professor at Florida International University. He specializes in teaching data structures and algorithm techniques courses in Java. He has also been involved in multiple research projects. Masoud talks to him about his senior project class and asks him if he wants to be a project/domain/personal mentor and Mark agrees. Masoud also says to Mark he will be able to easily communicate with his mentees and other students using the mentoring subsystem of Collaborative platform. Masoud asks Mark for his email and tells him soon he will be getting an email with his credentials to login.

Masoud goes ahead and logins to the website and goes to the “Add user” screen. There he fills out Mark’s information like first name, last name,username and email and selects the roles he wants Mark to be When Masoud clicks next he is taken to a wizard where he will be able to choose how he wants Mark to participate in the system.There will be a screen for each role:

* On the project mentor screen he will have option to select the projects and max ticket for the new user.
* On the domain mentor screen he will be able to choose the domain/subdomain, rating and tier for the new user.
* On the personal mentor screen he will be able to choose the mentees and max hours for the new user.

Once Masoud reaches the last screen he will be able to submit the form and complete the registration. This will trigger an email to the Mark with his credentials to login.

**Mentoring Module, Domain Mentor:**

**-**Continuation from Registration-

Mark is then greeted by a new screen where he sees the ability to add a picture of himself, give a detailed biography on previous jobs and also create a list of domains. Next to the blank picture finds and clicks on the “Sync with LinkedIn” button so that he can more easily fill out the necessary information. He completes the sync and finds his photo and job history on his potential profile page filled out. On the right he sees a table with a long list of suggested domains based on his skills from LinkedIn. From this list he is requested to specify a condensed list of domains that he wishes to mentor for. From here he gives each domain an expertise level ranging from one to ten, ten being at an expert level. He sees a short description for expertise level; 1-2 poor, 5-6 average, and 9-10 outstanding knowledge.

Next to the table he fills out an availability section detailing how many maximum issues he wishes to respond to on a monthly basis, the lower limit being 1. After setting up the page he clicks to send it for approval from the administrator.

-Assume to be approved-

Mark is a newly registered domain mentor as approved by the administrator. Based on the provided work experience and qualifications, Masoud, the administrator, has selected Mark to be a Tier 2 mentor. Mark receives this acceptance email requesting further action via email. He clicks on the email and is directed to his profile page where he can accept or decline the approval supplied by the administrator. Mark, happy with the approval of listed domains and ranking in the tier 2 mentor table, accepts the approval and completes the full registration process into the mentoring module.

-Assume issue has been assigned-

Mark checks his email one Saturday afternoon and finds an email from the collaborative platform. In it he sees that he has been assigned two issues. He clicks on the link and it directs him to his homepage within the mentoring module. Here he sees his queue with the two outstanding issues, one for Java another for C++. He clicks on the java issue, a question that has been pushed up by a tier 1 mentor, John. Mark, being that he has over 20 years of experience with java, easily answers the issue and sends it back to John.

Once back on the homepage, Mark glances at his issue queue and sees the java question he just answered has been grayed out and says “answered”. He then clicks on the “pending” C question to see if he can answer it. Mark finds the question very ambiguous and thus cannot fully propose a resolution. He chooses to append a comment on the issue to see if he can clarify the question proposed. Michael, the originator of the question, clarifies the question to Mark and sends it back. Mark sees it and unfortunately still cannot fully give an answer. He decides to open the issue and reroute it to one of his colleagues, Jason, who he feels may be more capable of answering the C issue. Mark could have also messaged Jason for help, but as Mark was already a little overloaded with work, decided to simply reroute the ticket to Jason as it was his expertise. (In which case messages could be accessed via the message tab)

-Assume another issue has been assigned-

-Assume request for new domain-

Clarke is a professor originally from UCLA in California. Clarke is also a registered domain mentor who has previous experience teaching in Ruby on Rails (RoR). Clarke has been approached a few times by different students requesting for some help with RoR. He decides that it would be beneficial to add it to his list of known domains for any future questions asked by potential and current mentees. Clarke goes to his profile page and finds the table with domains and expertise levels. He clicks to add domain and writes in the drop down, “Ruby on Rails”. He selects an expertise level of 7 as he has about 6 years of industry knowledge as well as educational research experience with the language. Once done, he clicks to add and confirms to send it to the administrator to be approved.

-Domain Mentor Tier 1 has similar functionality with the exception of being able to escalate an issue as described to a Tier 2 mentor. –

**More Scenarios:**

**Scenario 1**

1a. The system automatically assigned Paul’s ticket to a domain mentor in tier 1 of such domain. Then, the system sent a notification email to the assigned mentor. Jonathan the domain mentor, to whom the ticket was assigned, for whatever reason he is unsure of what to do. He decides to reassign the ticket to the administrator. He clicked in the reassign button and a window appears to write a comment, where he can explain the reason for reassigning the ticket. The comment is created and the system reassigns the ticket to the administrator. The system sends three notification emails. One notification is for the administrator. Another email is sent to the mentee, to let him know about the changes made. And finally to Jonathan, the domain mentor, who reassigned the ticket, to let him know that the ticket is now out of his queue. Then, the administrator is going to decide what to do next with the ticket.

1b. Alternately, Jonathan if he is unsure of how to best respond or can’t in a reasonable time, he can just pass on the ticket so the system can reassign it to the next capable person on the queue if there is one and if not it will go to the administrator. The system is going to send three notification emails. One notification is for the person to whom the ticket was reassigned by the system. Another email is sent to the mentee, to let him know about the changes made. And finally to Jonathan, the domain mentor, who reassigned the ticket, to let him know that the ticket is now out of his queue.

**Scenario 2**

The system automatically assigned Paul’s ticket to a domain mentor in tier 1 of such domain. Then, the system sent a notification email to the assigned mentor. Jonathan the domain mentor, to whom the ticket was assigned, decides to manually reassign the question to Mark. Jonathan took that step because he feels that Mark is more capable of answering that issue. He clicks on the reassign button, a new screen opens and he selects Mark’s name from the combo box. He creates a comment and clicks OK. Then, the comment is created and the system reassigns the ticket to Mark. Also, the system is going to send three notification emails. One notification is for Mark, the new mentor to whom the ticket was reassigned. Another email is sent to the mentee, to let him know about the changes made. And finally to Jonathan, the domain mentor, who reassigned the ticket, to let him know that the ticket is now out of his queue.

**Scenario 3**

The system automatically assigned Paul’s ticket to a domain mentor in tier 1 of such domain. Then, the system sent a notification email to the assigned mentor. Jonathan the domain mentor, to whom the ticket was assigned, chooses the ticket from his queue and answers Paul’s question. At this time, the system creates a notification for the mentee, to let him know that a comment was added to the ticket. Paul, the mentee, is not satisfied with the answer given by Jonathan, and creates a comment to the ticket clarifying the question. Jonathan, the mentor, creates another comment in response to Paul’s comment. Again, the system creates a notification for the mentee, to let him know that a new comment was added to the ticket. Finally, the mentee, still unsatisfied with the comment received, decided to reject the ticket. He clicked in the reject button and a window appears to write a comment, where he can explain the reason for rejecting the ticket. The comment is created and now, the ticket can be reassigned only by the administrator. The system sent two notification emails. One is for the mentor, to whom the ticket was previously assigned, to notify him about the rejection of the ticket. And, another email is for the administrator, to notify him that the ticket was rejected. Then, the administrator is going to decide what to do next with the ticket.

**Scenario 4**

The system automatically assigned Paul’s ticket to a domain mentor in tier 1 of such domain. Then, the system sent a notification email to the assigned mentor. Jonathan the domain mentor, to whom the ticket was assigned, chooses the ticket from his queue and answers Paul’s question. Consequently, the system creates a notification for the mentee, to let him know that a comment was added to the ticket. Paul was satisfied with the response received and decided to manually change the ticket status to close. A notification is sent to the mentor letting him know that the ticket was closed.

**Scenario 5**

The ticket created by Paul, was assigned to a domain mentor in tier 1 of such domain. Then, the system sent a notification email to the assigned mentor. Fifteen minutes later, the mentee found the answer in his programming book by himself. He returned to the Collaborative Platform website and decided to close manually the ticket without been answered. . A notification is sent to the mentor letting him know that the ticket was closed.

**Scenario 6**

The system automatically assigned Paul’s ticket to a domain mentor in tier 1 of such domain. Then, the system sent a notification email to the assigned mentor. Jonathan is the domain mentor, to whom the ticket was assigned. Robert is Paul’s personal (or could be a project mentor) mentor. For that reason, he also is able to see Paul’s ticket (not on his queue… his queue only has tickets that are assigned to him not that he can view). Robert knows further info that may be helpful to assist in answering the question, then, he chose Paul’s ticket and created a comment for the ticket. Consequently, the system creates two notifications. One is for the mentee, to let them know that a comment was added to the ticket. And, another is for the domain mentor, to inform him that a comment was added to his assigned ticket. Jonathan, who is working the issue that is on his queue now has additional information to help him respond to Paul.

**Scenario 7 (escalation)**

The system automatically assigned Paul’s ticket to a domain mentor in tier 1 of such domain. Then, the system sent a notification email to the assigned mentor. Jonathan the domain mentor, to whom the ticket was assigned, decides to manually escalate the ticket to a mentor in tier 2. Jonathan took that step because the question is beyond of what a tier 1 should know. He clicks on the escalate button, and a new screen opens and he clicks in the ok button to proceed with the escalation. The system creates a new ticket with all the comments that already exist in the original ticket. Jonathan the domain mentor in tier 1 appears as the creator of the new ticket and the system assigns the ticket to mentor in tier 2 of such domain and the original ticket and this new ticket are cross linked by comment created by the system. Then, the system sent a notification email to the assigned mentor in tier 2.

After this point, the relation between a tier 1 and a tier 2 mentor, it is going to be similar to the relation between a mentee and a tier 1 mentor. Therefore, the basic scenarios are going to apply to the ticket created by the mentor in tier 1 and assigned to a mentor in tier 2.

The mentor in tier 2 can see all the comments made on that ticket. After analyzed them all, he answered the ticket. The system creates a notification for the domain mentor in tier 1, to let him know that a comment was added to the ticket. The domain mentor in tier 1 sees the solution, and answers the ticket created by the mentee based in the solution given by the mentor in tier 2. The system creates a notification for the mentee, to let him know that a comment was added to the ticket. Paul, the mentee, was satisfied with the response received and decided to manually change the ticket status to close. The domain mentor in tier 1 sees that the mentee closed the original ticket and he received a notification about this action. He manually closed his ticket and the mentor in tier 2 also received a notification letting him know that the ticket was closed.

**Mentoring Module, Project/Personal Mentor:**

**-**Assume to be past first page of registration-

Bryan is a software development manager at UIF Technologies. He has a few side projects going besides work and wishes to be more involved with students at the local university. He is contacted by Masoud to join as a potential project mentor within the platform. Bryan is linked by email from Masoud and passes the first page of registration. Once entering basic credentials and password he is greeted with a page displaying all the projects drawn from the senior project website. Here he can see the project info; students signed up, skills related to the project, and client. He also sees an availability section asking him to detail his weekly availability as well as the desired number of projects. On the project selection he sees two tables. One asks him to rate the projects he’s interested in; 1-5, 1 being low and 5 being high. The other simply asks him to click whichever projects he does NOT want to mentor. He decides to simply click that he does not want to do mobile applications and will mentor 2 projects this semester.

-Assume to be approved-

Bryan is a newly approved and registered project mentor within the collaborative platform. He has been assigned 2 projects, one cloud computing and another for the senior project website. He has communicated a few times with his mentees and notices he has an email with a pending issue. Bryan clicks the issue via email and is brought back to his home page with a question from a mentee asking “what is the difference between public and private cloud systems?” Bryan finds this question a little vague and chooses to add a comment to the ticket, “which type of public and which type of private cloud systems?”

John, the mentee that originated the issue, clarifies the question and then Bryan quickly sends a response back. Bryan notices that he also had the ability to drop the question if he so deemed it unnecessary. Bryan also notices on his homepage an access point to see the work submitted in forms of deliverables from the project he is mentoring.

-Assume to have deliverable submitted-

Bryan receives an email from the platform alerting him that one of the projects he is mentoring has submitted a deliverable. Upon reading the deliverable and closely analyzing much of the information, he realizes that he needs to set up a meeting with the projects student contributors (mentees). Bryan goes to his homepage and selects the meetings tab. From here he requests a virtual meeting and selects a date and time for the students in the cloud computing project.

-Assume to be in a meeting-

Bryan is in a meeting with his project mentees. He chooses to begin with a quick video chat to ask if there are any questions. Once having jotted down a few notes he chooses to go down the list one by one. Bryan then uses the virtual white board to tell his mentees a specific pseudo code for an algorithm they need more information on. After writing it on the whiteboard, they are still left a little confused. Bryan, choosing to be nice, uses the ability to share screens as he writes down a few samples of code for the algorithm. Bryan then receives a question regarding a previous ticket that was assigned. He goes to the tickets tab and clicks on resolved, finds the “Systems” issue and quickly brushes up the teams understanding. Once the idea was fully grasped from his mentees, he asks them to submit a sample via the file sharing functionality within the module.

-Personal Mentor has similar functionality with the exception of the meeting rooms and selection of students as opposed to projects-

**More scenarios:**

**Scenario 1**

1a. Paul assigns his ticket to Jack (personal or project mentor). Then, the system sent a notification email to the assigned mentor. Jack the project or personal mentor, to whom the ticket was assigned, for whatever reason he is unsure of what to do. He decides to reassign the ticket to the administrator. He clicked in the reassign button and a window appears to write a comment, where he can explain the reason for reassigning the ticket. The comment is created and the system reassigns the ticket to the administrator. The system sends three notification emails. One notification is for the administrator. Another email is sent to the mentee, to let him know about the changes made. And also, to Jack the personal or project mentor, who reassigned the ticket, to let him know that the ticket is now out of his queue. Then, the administrator is going to decide what to do next with the ticket.

1b. Alternately, Jack if he is unsure of how to best respond or can’t in a reasonable time, he can just pass on the ticket so the system can reassign it to the next capable person on the queue if there is one and if not it will go to the administrator. . The system is going to send three notification emails. One notification is for the person to whom the ticket was reassigned by the system. Another email is sent to the mentee, to let him know about the changes made. And finally, to Jack the personal or project mentor, who reassigned the ticket, to let him know that the ticket is now out of his queue.

**Scenario 2**

Paul assigns his ticket to Jack (personal or project mentor). Then, the system sent a notification email to the assigned mentor. Jack the project or personal mentor, to whom the ticket was assigned, decides to manually reassign the question to Mark. Jack took that step because he feels that Mark is more capable of answering that issue. He clicks on the reassign button, a new screen opens and he selects Mark’s name from the combo box. He creates a comment and clicks OK. Then, the comment is created and the system reassigns the ticket to Mark. Also, the system is going to send three notification emails. One notification is for Mark, the new mentor to whom the ticket was reassigned. Another email is sent to the mentee, to let him know about the changes made. And finally, to Jack the project or personal mentor, who reassigned the ticket, to let him know that the ticket is now out of his queue.

**Scenario 3**

Paul assigns his ticket to Jack (personal or project mentor). Then, the system sent a notification email to the assigned mentor. Jack the project or personal mentor, to whom the ticket was assigned, chooses the ticket from his queue and answers Paul’s question. Consequently, the system creates a notification for the mentee, to let him know that a comment was added to the ticket. Paul, the mentee, is not satisfied with the answer given by Jack, and creates a comment to the ticket clarifying the question. Jack, the mentor, creates another comment in response to Paul’s comment. Again, the system creates another notification for the mentee, to let him know that another comment was added to the ticket by Jack. Finally, the mentee, still unsatisfied with the comment received, decided to reject the ticket. He clicked in the reject button and a window appears to write a comment, where he can explain the reason for rejecting the ticket. The comment is created and now, the ticket can be reassigned only by the administrator. The system sent two notification emails. One is for the mentor, to whom the ticket was previously assigned, to notify him about the rejection of the ticket. And, another email is for the administrator, to notify him that the ticket was rejected. Then, the administrator is going to decide what to do next with the ticket.

**Scenario 4**

Paul assigns his ticket to Jack (personal or project mentor). Then, the system sent a notification email to the assigned mentor. Jack the project or personal mentor, to whom the ticket was assigned, chooses the ticket from his queue and answers Paul’s question. Consequently, the system creates a notification for the mentee, to let him know that a comment was added to the ticket. Paul was satisfied with the response received and decided to manually change the ticket status to close. A notification is sent to the mentor letting him know that the ticket was closed.

**Scenario 5**

The ticket created by Paul, was assigned to a personal or project mentor. Then, the system sent a notification email to the assigned mentor. Fifteen minutes later, the mentee found the answer in his programming book by himself. He returned to the Collaborative Platform website and decided to close manually the ticket without been answered. A notification is sent to the mentor letting him know that the ticket was closed.

**Scenario 6**

Paul assigns his ticket to Bob (project mentor). Then, the system sent a notification email to the assigned mentor. Robert is Paul’s personal mentor. For that reason, he also is able to see Paul’s ticket (not on his queue… his queue only has tickets that are assigned to him not that he can view). Robert knows further info that may be helpful to assist in answering the question, then, he chose Paul’s ticket and created a comment for the ticket. Consequently, the system creates two notifications. One is for the mentee, to let them know that a comment was added to the ticket. And, another is for the project mentor, to inform him that a comment was added to his assigned ticket. Bob, who is working the issue that is on his queue now has additional information to help him respond to Paul.

**Mentoring Module, Mentee:**

-Assume to be registered-

John is a student within the senior project course at FIU. He is doing some work with Java and has a few questions regarding the languages functionality. He logs into the platform and is greeted by his homepage. Here he chooses the option to create a new ticket. From the pop up menu he create the title with “Java – multi threading” and specifies the filter for Java. In addition to the details within the ticket, he decides to include a short clip of the source code he is working with. John attaches the source code file and submits the ticket into the system.

-Assume to have ticket answered-

John receives an email from the collaborative platform. Within the email he reads that a ticket has been answered. Delighted to find the answer to an issue he had asked, John clicks the link to take him to his homepage. Here he sees his queue of issues with the Java question marked as answered. John opens the ticket to see the resolution and is indeed pleased with the solution. He then accepts the answer and closes the issue ticket.

**Administrator:**

-Assume to have pending registrations-

Masoud is the head professor for the computer science senior project course. He also works as the administrator for the collaborative platform, given their close interaction. Masoud checks his email to see that he has a few pending registrations for new members within the platform. The first he opens is Mark, a well-known and brilliant professor with a specific expertise in algorithms and Java. Upon opening his profile page, Masoud glances over his credentials and approves Mark to be a Tier 2 Domain Mentor.

The second on his list of registrations is Luke. Luke is a young professional within the software engineering field. Masoud had previously assigned Luke to be a Tier 1 Domain mentor with specializations in C++, C#, and Operating Systems. Luke, however, disapproved with Masoud and felt that he should have been assigned to be a Tier 2 mentor. Luke opened negotiations with Masoud and has since sent him the most recent registration to be approved. Masoud opens the registration to see Luke’s new credentials specifying a more detailed background within the domains and approves his request to be bumped up into Tier 2.

-Assume to have multiple approved profiles-

Masoud logs in one day and decides to take a look over the registered users within the collaborative platform. Once logged in, he clicks on the profiles tab and is greeted with a list of registered users. For each user that Masoud clicks on, via a given list, he is allowed to see what roles they have signed up for, their domain, projects, personal mentees, and other related information per each specific user. From here, if the administrator finds it necessary, he can edit the information in any way he wants, or simply delete the user if they are no longer active or wish to leave.

-Assume pending domain approval -

Masoud receives an email from the collaborative platform system informing him there has been a request to add a new domain to the database. He opens the email and finds that the request is from Clarke. Clarke is a professor originally from UCLA in California who has previous experience teaching in Ruby on Rails (RoR). Masoud knowing the importance of the now widely used scripting language decides that he feels comfortable with Clarke’s expertise level in the language and approves it to be added for future mentors to list as well.

-Superpower View-

When Masoud logs into the system he has the ability to view anything he wishes. Masoud decides to take a look at the entire ticket queue and see the statuses on each assigned ticket for completion. On one ticket for PHP, he realizes that the ticket has gone stale without an answer for two days. Although the assigned mentor had replied with a comment, they never posted an actual solution. To expedite the resolution, Masoud reassigns the ticket to another mentor so that it can be answered. When doing so, he also has the ability to upload files, append comments, or even close, accept and cancel the issue. Afterwards, he decides to then take a look at the project mentors and any current or future meetings. Masoud clicks on the meetings tab and is greeted with a new page showing him the complete list of meetings that have been scheduled.

**Scenario**

Administrators are the only users that can manage the ticket‘s priorities. When the administrator logs into the system, he has the ability to view or update the priorities of the tickets by clicking on the “Priorities” button. When he clicks in the “Priorities” button, the “Manage Priorities” screen opens. From this screen, he can select any of the existing priorities and update the number of hours, for which the system should be waiting for a mentor’s answer, before automatically reassign a ticket with that particular priority.

**Mentoring Module: Version 3 Scenarios**

**Mentor Account Registration**

* Scenario 1: Email Invitation

Dr. Steven Bruhl logs into his email and finds an email from the Collaborative Platform. He opens it and sees that it is an invitation to be a Mentor for the Mentoring Module sent by his good friend Masoud. After reading the description Dr. Bruhl decides to click the link and check it out. His browser then loads up the Collaborative Platform landing page. He reads the descriptions of the Collaborative Platform, the Mentoring Module and the expectations of a mentor. He loves the idea and immediately clicks sign-up.

He is taken to the account registration page. Dr. Bruhl proceeds to fill out his desired account info including his email, username and password. He then proceeds to fill out his current employment information and then the details of his highest degree. He clicks the Next button and a verification modal appears. He overlooks his information and realizes he misspelled his name. He then clicks the back button to close the modal. He fixes the spelling on his name and clicks Next again. he verifies the updated information and then clicks the Submit button. He is then immediately logged in and taken to the Mentor Application Portal.

* Scenario 2: No Invite

Prof. Benny Hannah has been hearing more and more about the famous Collaborative Platform. He decides to go to the website and see it for himself. On the login page he sees a link labelled Register. He clicks it and is taken to the Collaborative Platform landing page. he reads a bit more about the Platform and decides to sign up. He clicks the button and is taken to the account registration page.

Prof. Hannah then quickly fills out his account, work and education information. He clicks the Next button and triggers some errors. He sees that he has not entered in a starting year for his latest job and his password confirmation does not match the original. He enters in the starting data and retypes in his password and password confirmation. He clicks next and the verification modal is loaded. He looks over his information and is satisfied. He clicks submit and finds himself logged into his new account and at the Mentor Application Portal.

* Scenario 3: Cancellation

Kenny Powers, Senior Mobile App Developer from Company X, has received an email invitation from the Collaborative Platform. He clicks the link and finds himself on the landing page. He is pressed for time and quickly skims over the descriptions. He clicks Sign Up and is taken to the account registration page. He looks at his clock and sees that he does not have enough time to fill out this form. He clicks the Cancel button and is taken back to the log in page. He then closes his browser and gets back to work knowing he can always use his email invitation to come back.

**Personal Mentor Application**

* Scenario 1: Full Application

Dr. Steve Bruhl just finished creating his account and is at the Mentor Application Portal. He reads the description for a Personal Mentor. He has always wanted a fresh student to mentor so he clicks on the Personal Mentor portrait and is redirected to the Personal Mentor Application page.

Dr. Bruhl reads the instructions and sees he can choose students from the list provided. He hovers over the information icon above the list and reads how it works. Now with an understanding of how the list works he filters the list by university by clicking the appropriate header. He scrolls down to the section of FIU students and sees that one of his old students Timmy Turner is there on the list. He hovers over his name to make sure its him. Timmy uploaded a picture of himself so Dr. Bruhl recognized him right away. He clicks his name and Timmy is moved to the column of selections.

Dr. Bruhl does not recognizing any of the other students but still wants another Mentee. He goes to the system pick column and sets the criteria for one more student from FIU. Satisfied, Steven clicks the Next button and a verification modal appears. He verifies his information and then clicks Submit. He is redirected back to the Mentor Application Portal. He sees on the Personal Mentor Application column that his application is pending.

* Scenario 2: Partial Application

Prof. Benny Hannah is browsing the Mentor Application Portal. He decides to apply for personal mentorship by clicking the apply button. he sees the list of Mentees but does not recognize any of them. He instead decides to have the system pick a Mentee for him from FIU. He clicks next and a warning pops up saying that he must select at least one student from the list or have at least 1 student assigned to him. He sees that he forgot to fill in the system pick amount field. He enters the number 1 and then clicks next again. A verification modal appears and Prof. Hannah reads over the summary and clicks Submit. He is redirected back to the Mentor Application Portal. He sees on the Personal Mentor Application column that his application is pending acceptance.

* Scenario 3: Cancelled Application

Kenny Powers finally had time to finish registering for an account. He finds himself at the Mentor Application Portal and clicks Apply on the first button he sees. He is taken to the Personal Mentor Application page. He does not recognize any of the students on the list nor is he pleased with any of the Universities with participating Mentees. Being unsatisfied with the options presented grumpy Mr. Powers clicks the Cancel button and is taken back to the Mentor Application Portal.

**Project Mentor Application**

* Scenario 1: Full Application

Dr. Steve Bruhl just finished applying for some Personal Mentorships and can barely contain his excitement. He then reads the description of project mentorships and really likes the idea of being able to act as a mentor for a whole project. The clicks the Project Mentor portrait and is redirected to the Project Mentor Application page.

Dr. Bruhl reads the instructions and sees he can choose projects from the list provided. He hovers over the information icon above the list and reads how it works. Now with an understanding of how the list works he begins to read the description of every project on the list! He sees that one of the projects has his favorite student Timmy Turner as a mentee. He clicks on the project, “Collaborative Platform: Version 4” and it is moved to the column of selections.

Dr. Bruhl, as excited as he may be, wants to dedicate his time to only one project so he leaves the system pick amount at 0. Then he clicks the next button, looks over the summary and changes his mind. He immediately clicks the back button and sets the system pick amount to 5. He could never pass on an opportunity like this! He clicks next again, goes over the summary and clicks Submit. He finds himself back at the Mentor Application Portal. He sees on the Project Mentor Application column that his application is pending acceptance.

* Scenario 2: Partial Application

Prof. Benny Hannah is browsing the Mentor Application Portal. He decides to apply for project mentorship by clicking the corresponding apply button. He sees the list of Projects and reads the descriptions of a few. The “Collaborative Platform: Version 4” catches his eye and he selects it. Satisfied with his selection he clicks Next. A verification modal appears and Prof. Hannah reads over the summary and clicks Submit. He is redirected back to the Mentor Application Portal. He sees on the Project Mentor Application column that his application is pending acceptance.

* Scenario 3: Cancelled Application

Kenny Powers, displeased with his last attempt at applying for mentorship and running short on time as usual, quickly selects the next application.He is taken to the Project Mentor Application page. He starts to read over some of the project descriptions but he doesn’t really have the time to read them. He decides instead that he would like to have the system pick a project for him. He clicks Next and a warning appears telling him to either select a project from the list or elect to have the system choose one for him. Upset that a machine was telling him what to do Kenny furiously clicks the Cancel button and is redirected to the Mentor Application Portal.

**Domain Mentor Application**

* Scenario 1: Full Application

Dr. Steve Bruhl has applied for two different mentorships and is on Cloud 9. He sees that there is one type of mentorship left. He then reads the description of domain mentorships and then clicks the Domain Mentor portrait. He is redirected to the Domain Mentor Application page.

Dr. Bruhl reads the instructions and sees that in order to pick a domain he must set his proficiency on the desired Domain. Being an expert on Programming Languages, Dr. Bruhl sets his proficiency for the Programming Languages Domain to 10. He clicks on the Programming Languages link and the table expands to show its Subdomains. Dr. Bruhl is well versed in many programming languages and sets his proficiency for each.

He then scrolls to the bottom and fills in the field for tickets per month. He confidently puts down that he wants 50 tickets per month. He could answer these tickets with one hand tied behind his back, he is a doctor after all. Then Dr. Bruhl clicks the next button, verifies the submitted information and clicks Submit. He is redirected to the Mentor Application Portal and sees that he has an application of every type pending. He clicks the Done button and is redirected to the homepage.

* Scenario 2: Partial Application

Prof. Benny Hannah is browsing the Mentor Application Portal. He decides to apply for domain mentorship by clicking the corresponding apply button. He sees the list of Domains and reads the descriptions of a few. A few catch his eye and he expands them to see what Subdomains they offer. Prof. Hannah hasn’t been coding as often as he used to se he elects to only apply for the general Programming Languages Domain and puts a proficiency of 7. He also feels he has the skills needed to answer general Biology questions as well and sets his proficiency to 5 on that one. Satisfied he then clicks the Next button. A verification modal appears and Prof. Hannah reads over the summary and clicks Submit. He is redirected back to the Mentor Application Portal. He sees on the Project Mentor Application column that his application is pending acceptance.

* Scenario 3: Cancelled Application

Kenny Powers, now livid from his failed attempts to apply, quickly selects the next application. He is taken to the Domain Mentor Application page. He starts to read over some of the Domain descriptions and realizes that he has no time at all for this. He quickly clicks the Cancel button and is redirected to the Mentor Application Portal.

**Manage Users:**

Scenario 1: Perform search on Users to find all Mentors.

Admin, Martin Lawrence, decides to see the list of all users who have the following roles: domain mentor, personal mentor, and project mentor. The admin selects the appropriate checkboxes based on the defined criteria explained before. Martin then selects the search button and the table is regnerated with the requested information. Martin now can see the information needed and can proceed to click on details of the user to see any additional details.

Scenario2: Find Mentee “Jonathan Sanchez” to see how the student is using the system.

Admin, Martin Lawrence decides to see how the mentee “ Jonathan Sanchez “ is using the system. He will proceed to the Manage->Users page and select “Mentee” from the basic search function and clicks search. The admin can now see users that have the mentee roles. Martin would now like to search in the provided Name field. Martin types “Jona” and hits the return key. The table is regenerated with the requested data and find the User he was look for. Martin then clicks on the eyeball which will cause a Modal to open. Then admin can now see how “Jonathan” is using the system by seeing how many tickets have been opened, how many tickets have been closed, who their personal mentors are, who their project mentors are, what project they’re assigned to, what meetings they have had as well as any future meetings. Martin Lawrence then closes the modal by clicking outside of the modal instead of the provided “X” - close button.

**Manage Projects:**

Scenario 1: Find Mentor Module v3 and gain insight on system usage.

Admin, Martin Lawrence, continues to dig into the system data to gain information. Now he is looking for how a particular group assigned to project: Mentor Module. He clicks on Manage->Projects and used the provided search function on ‘Title’. He enters the name and finds the needed project. Martin proceeds to click on the eyeball to get more details. The system provides a modal with the insight he is looking for. The modal provides information such as, what mentors are assigned to the project, what mentees are working on the projects (as well as the ability to click on a user to see their individual profile page), and what meetings are scheduled with the project members, and any tickets that have originated from users within the project. Martin sees that the members of the project have had significant amount of meetings and is satisfied with the usage of the system. He then decides to give everyone on the team an A on project/mentor communication.

**Manage Applications**

Scenario 1: View Applications

Admin, Martin Lawrence, has no idea how many applications are pending and would like to see them to perform his duties. Martin then selects Manage->application from the dropdown and is redirected to a view where he can see a list of all pending applications. Martin sees that there are only 3 applications and that their submission dates are from 2 hours ago so decides to postpone the approval because he has other urgent matters to attend to.

Scenario 2: Accept / Reject Roles

Admin, Martin Lawrence, decides to go into the application submitted by Mentor, Michael Faterno. The admin opens his applications through Scenario 1 and clicks on the eyeball to view the details of the application. At this point the admin sees that Mr. Faterno has submitted applications for all three roles and proceeds to evaluate all selections. Martin approves Michael Faterno all requested items because of his significant experience and exposure in the industry. Since all items have been agreed to the Mentor submits the applications and the system will then check to see if any items are pending approvals on the mentor side (this only occurs if the admin proposes new items). The application is the marked as closed and Martin calls Michael up to give him the good news.

Scenario 2 : Cancel Application process

Admin, Martin Lawrence decides to go into the application of Mentor, Charles Xavier to review items but then suddenly receives an urgent call. The mentor can simply select the cancel button at the bottom of the form or press the back button in the browse. Martin can now go handle the urgent matter reported by his wife, Jennifer Lawrence.

### 

### 4.2.2 Use Case Model

The use case diagram provides a graphical picture about the functionalities of the Collaborative Platform and its relation with the actors, for our purpose we are showing the two new modules: Mentoring and Remote Judge. The Virtual Job Fair was not included in this document but you can find the link to it, in the Reference section. (Refer to Appendix B)

### 4.2.2 Static model

Refer to Appendix C

The Static Model in appendix C provides developers with a specification of all the classes that should be implemented for the system. It does not differentiate between the different subsystems; it simply contains all of the necessary classes that were identified during the requirements analysis, i.e. the collection of all the classes from the different subsystems. Conventional UML notation was used to make it easier to differentiate between boundary, controller, and model classes, as per-required for the MVC architectural pattern.

### 4.2.3 Dynamic model

Refer to Appendix D

The Dynamic model in appendix D contains the sequence diagrams extracted from analyzing the specific requirements of each of the Use Cases presented in Appendix A. It gives the team of developers a more specific view of the interaction between the user and the system for each specific piece of functionality from the Use Case Model.

By referring to this diagram, developers can see the main artifacts involved in the flow of information for each use case, and how they are associated with each other: boundary objects as the intermediaries between user requests and controllers, controllers accessing model objects and passing along the returned data to the boundary objects, etc.

Conventional UML notation was used to produce the sequence diagrams.

# 5. System Design

This section will describe in detail our system architectures for the Collaborative Platform giving an overview of the primary and the secondary architecture patterns. In addition, it will provide a detailed subsystem decomposition description for each of the major subsystems. Also, the hardware and software mapping are defined, with data structures that the system will need to perform the operations required by the client. Finally, some aspects of security and privacy are covered as they are needed for keeping the integrity of the data stored in the database.

## 5.1 Overview

As a result of our understanding of the system, we determined to use the Model View Controller (MVC) architecture, because this architecture will benefit the team at the time of implementation, because we can effectively divide the work, so any change in one of the classes does not need a large alteration, if any, to other classes.

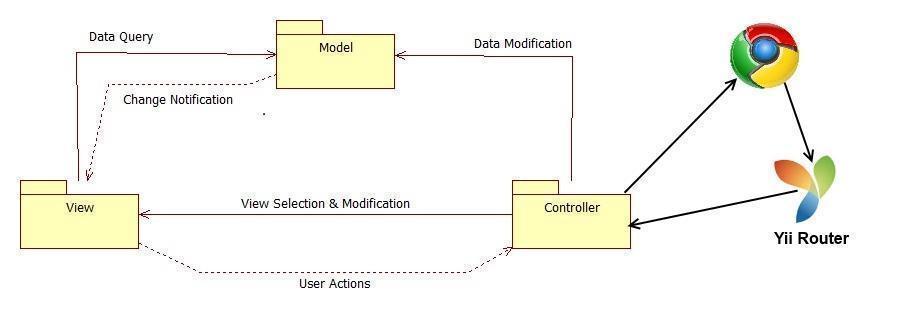


Figure 2.1.b Model View Controller

## 

## 

## 5.2 Subsystem Decomposition

Below we present a description of each of the major subsystems and their respective functional requirements.

**Registration and Access Point Subsystem**

The registration and access point subsystem will be the main entry point for our platform. This will facilitate the process across the user database. Since there will be one login for each individual module, the access point is simplified into one subsystem. This subsystem will also cover the creation of new users and the retrieval of passwords for current users.

* Allow the user to log in.
* Allow the user to log out.
* Allow the user to retrieve forgotten password.

**Mentoring Subsystem**

The mentoring subsystem will cover the essentials of the mentoring module. Specifics to the mentoring module are separated into this subsystem. These functionalities include the essential workflow of the ticketing system for issues created by mentees for the different types of mentors.

* Auto assigns the most appropriate domain mentor to a ticket.
* Allow administrator to reassign the ticket to other domain mentor.
* Allow project mentor to retrieve the description of the projects assigned.
* Allow project mentor to retrieve all the mentees for each project.
* Allow project mentor to set up meetings with his/her mentees.
* Allow project mentor to retrieve all the upcoming meetings.
* Allow project mentor to assign ticket to other project mentor.
* Allow project mentor to see the ticket created by their mentees.
* Allow mentees to assign tickets to his/her project mentor.
* Allow user create a ticket.
* Allow user to select a specific domain for a new ticket.
* Allow user to select a specific sub domain.
* Allow user to upload file to a ticket.
* Allow user to download file from a ticket.
* Allow user to append comments to a ticket.
* Allow user to retrieve all the details of the ticket of the ticket that he/she created or was assigned.
* Allow user to close a ticket.
* Allow user to reject a ticket.
* Allow System Administrators (SA) to create new administrators.
* Allow SA to change user profile information and scope.
* Allow SA to manage Domains and Sub-Domains.
* Allow SA to disable and enable users.
* Allow SA to send invitations
* Allow SA to manage projects
* Allow users to edit their availability
* Allow users to edit their photo
* Allow project mentor to select desired projects
* Allow personal mentor to select desired personal mentees
* Allow domain mentors to add domain and expertise ratings
* Allow administrators to view and edit user profiles
* Allow mentees to assign tickets to his/her personal mentor.
* Allow personal mentor to see the ticket created by their mentees.
* Allow personal mentor to comment the ticket created by their mentees.
* Allow Domain mentor in tier 1 to escalate the ticket to a Domain Mentor in tier 2
* Allow user to select a specific priority for a new ticket.
* Allow administrator to change the priority waiting time.
* Allow the system to automatically reassign the tickets.

**Communication and Notification Subsystems**

The communication subsystem is a key subsystem that provides functionality derived from the mentoring module. This communication will be vital to the ticket system as users may frequently send time-sensitive information regarding projects or questions made by mentees.

* Allow users to send a message to another user.
* Allow users to read a message.
* Allow users to delete a message.
* Send ticket due email notification.
* Send new message email notification.
* Send password change email notification.
* Send profile change email notification.
* Send ticket assigned email notification.
* Send comment added to a ticket email notification.
* Send ticket closed email notification.
* Send ticket not available mentor email notification in the Automatic Reassign.
* Send ticket reassigned email notification to admin when the ticket was automatically reassigned three times.
* Send ticket escalated email notification.

**Integration with SPW**

* Allow admin to import data from SPW manually
* Allow the system to import data(students/projects) from SPW automatically
* Allow senior project students to login with their FIU Google login credentials
* Allow system to update reassign projects based on changes made in SPW
* Allow system to update reassign mentors based on changes made in SPW

**Registration by admin**

* Allow the admin to register mentors (project/domain/personal)
* Allow admin to edit mentors roles
* Allow mentors to login with credentials provided by admin

**Mentor Module: Version 3**

The system shall…

**Self-serve Mentor Registration**

* Allow a potential mentor to register for a mentor account from a new landing page.
* Allow mentors to apply for new mentorships from and all-in-one Mentor Application portal
* Allow mentors to begin their mentorships whenever approved by an admin
* Allow a Project Mentor to apply by selecting projects they like or deferring the decision to the system.
* Allow a Personal Mentor to apply by selecting students they like or deferring the decision to the system.
* Allow a Domain Mentor to apply by selecting topics they’re proficient in.
* Allow a Domain Mentor to suggest new domains.
* Allow Project Mentor to approve/reject projects suggested by admin.
* Allow Domain Mentor to approve/reject topics suggested by admin.
* Allow Personal Mentor to approve/reject mentee suggested by admin.
* Provide user verification for all appications
* Provide client side validation for all forms

**Admin Dashboard**

* Allow admin to view pending applications
* Allow admin to approve/reject Projects in a Project Mentor Application
* Allow admin to propose Projects in a Project Mentor Application
* Allow admin to approve/reject Mentees in a Personal Mentor Application
* Allow admin to propose Mentees in a Personal Mentor Application
* Allow admin to approve/reject Domains in a Domain Mentor Application
* Allow admin to propose Domains in a Domain Mentor Application
* Allow admin to view open invitations/re-invites
* Allow admin to customize the body of the message while sending an invite
* Allow admin to gain insight on site usage through admin dashboard
* Allow admin to gain insight on a particular project
* Allow admin to gain insight on a particular user
* Allow admin to gain insight on a particular domain
* Allow admin to gain insight on a particular subdomain
* Allow admin to view tickets
* Allow admin to manage tickets
* Allow admin to use an advanced search for tickets.

## 

## 

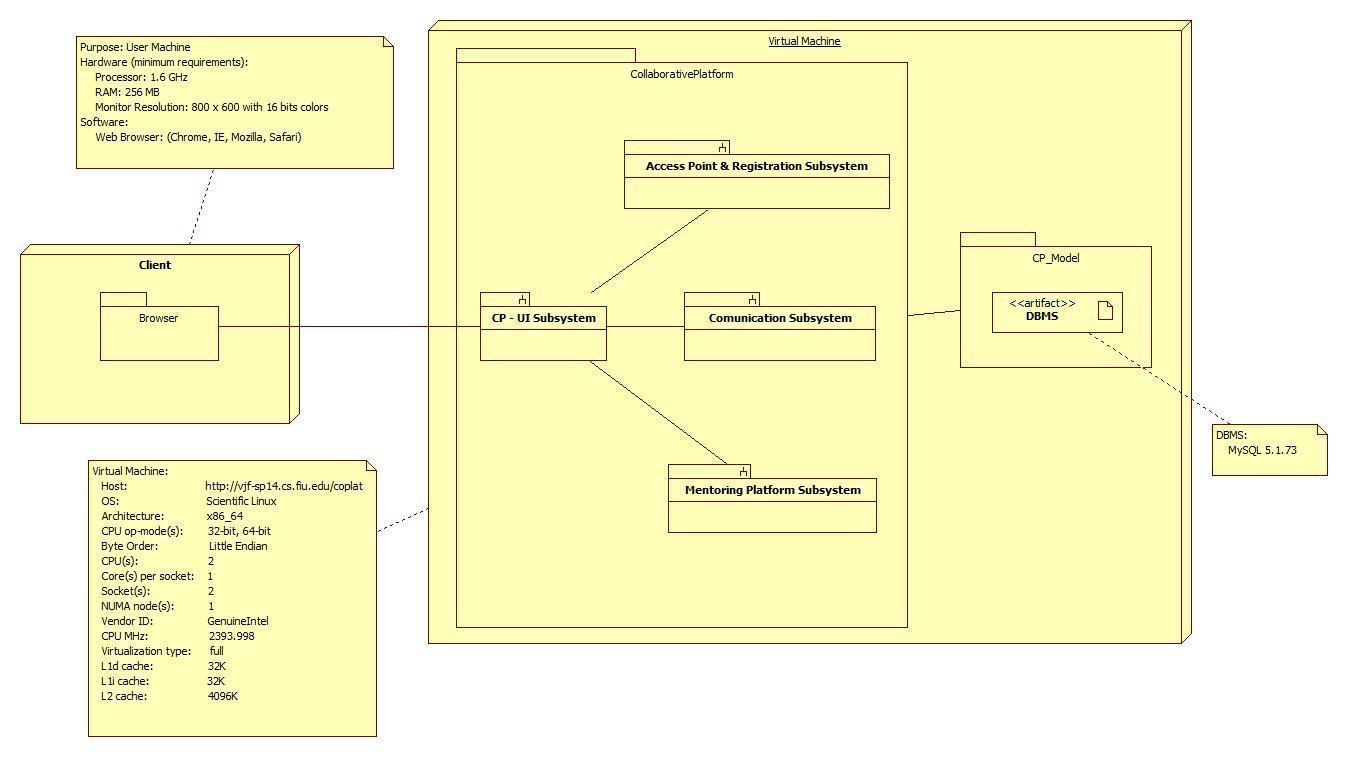
## 5.3 Hardware and Software Mapping

The collaborative platform system requires the following specifications:

|  |  |
| --- | --- |
| OS | Scientific Linux |
| Architecture | x86\_64 |
| CPU op-mode(s) | 32-bit, 64-bit |
| Core(s) per socket | 1 |
| Socket(s) | 2 |
| NUMA node(s) | 1 |
| CPU MHz | 2393.998 |
| Virtualization type | Full |
| L1d cache | 32k |
| L1i cache | 32k |
| L2 cache | 4096k |

The minimum hardware and software requirements for the user to access the collaborative system is as follows:

|  |  |
| --- | --- |
| OS | Web Browser (Chrome) |
| Processor | 1.6 GHz |
| RAM | 256 MB |
| Monitor Resolution | 800 x 600 with 16 buts color |

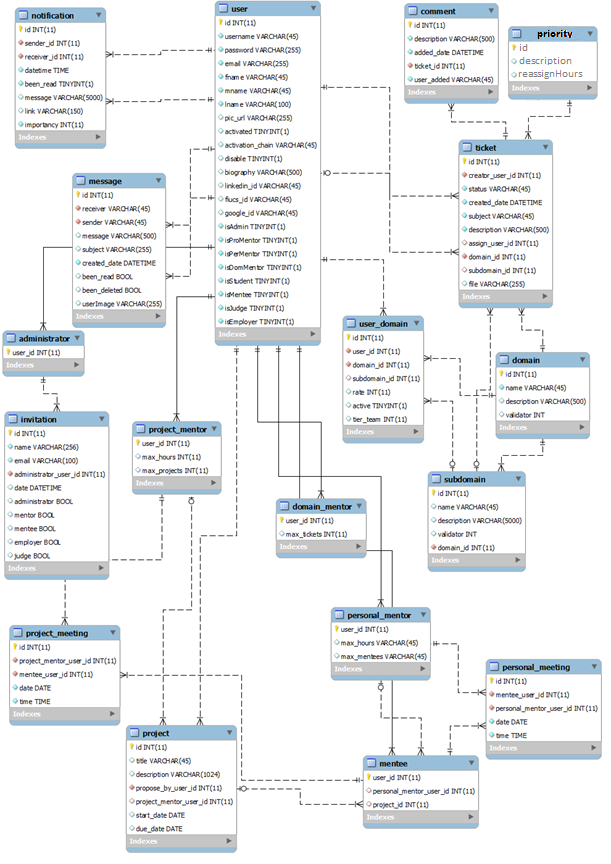


**Figure 5.3 Deployment Diagram**

The deployment diagram shown above represents the hardware and software mapping in the Collaborative Platform system.  The main components of the system are the Apache and MySQL environment hosted on a Linux operating system.  The Yii framework environment is using apache to execute, and contains our various artifacts used in development (Model, View, and Controller).  The models are mapped to tables in the MySQL environment set up on the same machine.  The browser on the client’s machine communicates with the server using HTTP.

## 5.4 Persistent Data Management

The previous group completed an ER diagram. Since we are adding new tables and columns into existing tables and database, respectively, below is the updated version of the ER diagram of the original group, which includes our new data, which is identified by a yellow background and dark red text.



### 5.4.1 Data Dictionary

#### administrator

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| user\_id | int(11) | No |  | user -> id |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | user\_id | 2 | A | No |  |
| fk\_administrator\_user1\_idx | BTREE | No | No | user\_id | 2 | A | No |  |

#### comment

Table comments:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |  |
| description | varchar(500) | No |  |  |  |  |
| added\_date | datetime | No |  |  |  |  |
| ticket\_id | int(11) | No |  | ticket -> id |  |  |
| user\_added | varchar(45) | No |  |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 0 | A | No |  |
| fk\_comment\_ticket1\_idx | BTREE | No | No | ticket\_id | 0 | A | No |  |

#### domain

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |
| name | varchar(45) | No |  |  |  |
| description | varchar(500) | Yes | *NULL* |  |  |
| validator | int(11) | Yes | 5 | Integer that validates the domain Tier Level. |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 6 | A | No |  |

#### domain\_mentor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| user\_id | int(11) | No |  | user -> id |  |  |
| max\_tickets | int(11) | Yes | *NULL* |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | user\_id | 7 | A | No |  |
| fk\_domain\_mentor\_user1\_idx | BTREE | No | No | user\_id | 7 | A | No |  |

#### invitation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |  |
| email | varchar(100) | No |  |  |  |  |
| administrator\_user\_id | int(11) | No |  | administrator -> user\_id |  |  |
| date | datetime | Yes | *NULL* |  |  |  |
| administrator | tinyint(1) | Yes | *NULL* |  |  |  |
| mentor | tinyint(1) | Yes | *NULL* |  |  |  |
| mentee | tinyint(1) | Yes | *NULL* |  |  |  |
| employer | tinyint(1) | Yes | *NULL* |  |  |  |
| judge | tinyint(1) | Yes | *NULL* |  |  |  |
| name | varchar(20) | No |  |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 0 | A | No |  |
| fk\_invitation\_administrator1\_idx | BTREE | No | No | administrator\_user\_id | 0 | A | No |  |

#### mentee

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| user\_id | int(11) | No |  | user -> id |  |  |
| personal\_mentor\_user\_id | int(11) | Yes | *NULL* | personal\_mentor -> user\_id |  |  |
| project\_id | int(11) | Yes | *NULL* | project -> id |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | user\_id | 1 | A | No |  |
| fk\_mentee\_personal\_mentor1\_idx | BTREE | No | No | personal\_mentor\_user\_id | 1 | A | Yes |  |
| fk\_mentee\_project1\_idx | BTREE | No | No | project\_id | 0 | A | Yes |  |

#### message

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| id | int(11) | No |  |  | Message ID |  |
| receiver | varchar(45) | No |  | user -> username | Receiver username |  |
| sender | varchar(45) | No |  | user -> username | Sender username |  |
| message | varchar(500) | Yes | *NULL* |  | Message Body |  |
| subject | varchar(255) | Yes | *NULL* |  | Message Subject |  |
| created\_date | datetime | No |  |  | Message Creation Date |  |
| been\_read | tinyint(1) | Yes | 0 |  | 0: NO 1: YES |  |
| been\_deleted | tinyint(1) | Yes | 0 |  | 0: NO 1: YES |  |
| userImage | varchar(255) | Yes | *NULL* |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 0 | A | No |  |
| fk\_message\_user1\_idx | BTREE | No | No | receiver | 0 | A | No |  |
| fk\_message\_user2\_idx | BTREE | No | No | sender | 0 | A | No |  |

#### notification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |
| sender\_id | int(11) | No |  |  |  |
| receiver\_id | int(11) | No |  |  |  |
| datetime | time | No |  |  |  |
| been\_read | tinyint(1) | No | 0 |  |  |
| message | varchar(5000) | Yes | *NULL* |  |  |
| link | varchar(150) | Yes | *NULL* |  |  |
| importancy | int(11) | No | 0 |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 6 | A | No |  |

#### personal\_meeting

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |  |
| mentee\_user\_id | int(11) | No |  | mentee -> user\_id |  |  |
| personal\_mentor\_user\_id | int(11) | No |  | personal\_mentor -> user\_id |  |  |
| date | date | No |  |  |  |  |
| time | time | No |  |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 0 | A | No |  |
| fk\_personal\_meeting\_mentee1\_idx | BTREE | No | No | mentee\_user\_id | 0 | A | No |  |
| fk\_personal\_meeting\_personal\_mentor1\_idx | BTREE | No | No | personal\_mentor\_user\_id | 0 | A | No |  |

#### personal\_mentor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| user\_id | int(11) | No |  | user -> id |  |  |
| max\_hours | varchar(45) | Yes | *NULL* |  |  |  |
| max\_mentees | varchar(45) | Yes | *NULL* |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | user\_id | 5 | A | No |  |
| fk\_personal\_mentor\_user1\_idx | BTREE | No | No | user\_id | 5 | A | No |  |

#### priority

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |
| description | varchar(45) | No |  |  |  |
| reassignHours | int(11) | No |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 3 | A | No |  |

#### project

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |
| title | varchar(45) | Yes | *NULL* |  |  |
| description | varchar(1024) | Yes | *NULL* |  |  |
| propose\_by\_user\_id | int(11) | No |  | Propose By |  |
| project\_mentor\_user\_id | int(11) | Yes | *NULL* |  |  |
| start\_date | date | Yes | *NULL* |  |  |
| due\_date | date | Yes | *NULL* |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 3 | A | No |  |
| fk\_project\_user1\_idx | BTREE | No | No | propose\_by\_user\_id | 3 | A | No |  |
| fk\_project\_project\_mentor1\_idx | BTREE | No | No | project\_mentor\_user\_id | 3 | A | Yes |  |

#### project\_meeting

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |  |
| project\_mentor\_user\_id | int(11) | No |  | project\_mentor -> user\_id |  |  |
| mentee\_user\_id | int(11) | No |  | mentee -> user\_id |  |  |
| date | date | No |  |  |  |  |
| time | time | No |  |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 0 | A | No |  |
| fk\_project\_meeting\_project\_mentor1\_idx | BTREE | No | No | project\_mentor\_user\_id | 0 | A | No |  |
| fk\_project\_meeting\_mentee1\_idx | BTREE | No | No | mentee\_user\_id | 0 | A | No |  |

#### project\_mentor

Table comments:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| user\_id | int(11) | No |  | user -> id |  |  |
| max\_hours | int(11) | Yes | *NULL* |  |  |  |
| max\_projects | int(11) | Yes | *NULL* |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | user\_id | 7 | A | No |  |
| fk\_project\_mentor\_user1\_idx | BTREE | No | No | user\_id | 7 | A | No |  |

#### subdomain

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |  |
| name | varchar(45) | Yes | *NULL* |  |  |  |
| description | varchar(5000) | Yes | *NULL* |  |  |  |
| validator | int(11) | Yes | 5 |  | Integer that validates the domain Tier Level. |  |
| domain\_id | int(11) | No |  | domain -> id |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 16 | A | No |  |
| fk\_subdomain\_domain1\_idx | BTREE | No | No | domain\_id | 16 | A | No |  |

#### ticket

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |  |
| creator\_user\_id | int(11) | No |  | user -> id |  |  |
| status | varchar(45) | No |  |  |  |  |
| created\_date | datetime | No |  |  |  |  |
| subject | varchar(45) | No |  |  |  |  |
| description | varchar(500) | No |  |  |  |  |
| assign\_user\_id | int(11) | Yes | *NULL* | user -> id |  |  |
| domain\_id | int(11) | No |  | domain -> id |  |  |
| subdomain\_id | int(11) | Yes | *NULL* | subdomain -> id |  |  |
| file | varchar(255) | Yes | *NULL* |  |  |  |
| priority\_id | int(11) | No |  | priority -> id |  |  |
| assigned\_date | datetime | Yes | *NULL* |  |  |  |
| isEscalated | tinyint(1) | Yes | *NULL* |  |  |  |
| Mentor1 | int(11) | Yes | *NULL* |  |  |  |
| Mentor2 | int(11) | Yes | *NULL* |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 0 | A | No |  |
| fk\_ticket\_user2\_idx | BTREE | No | No | creator\_user\_id | 0 | A | No |  |
| fk\_ticket\_user1\_idx | BTREE | No | No | assign\_user\_id | 0 | A | Yes |  |
| fk\_ticket\_domain1\_idx | BTREE | No | No | domain\_id | 0 | A | No |  |
| fk\_ticket\_subdomain1\_idx | BTREE | No | No | subdomain\_id | 0 | A | Yes |  |
| priority\_id | BTREE | No | No | priority\_id | 0 | A | No |  |

#### user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| id | int(11) | No |  | User ID |  |
| username | varchar(45) | No |  | User Name |  |
| password | varchar(255) | No |  | User Password |  |
| email | varchar(255) | No |  | User Email |  |
| fname | varchar(45) | No |  | User First Name |  |
| mname | varchar(45) | No |  | User Middle Name |  |
| lname | varchar(100) | No |  | User Last Name |  |
| pic\_url | varchar(255) | Yes | *NULL* | User Picture Location |  |
| activated | tinyint(1) | Yes | 0 | 1: Yes 0: No |  |
| activation\_chain | varchar(45) | Yes | *NULL* | Activation String |  |
| disable | tinyint(1) | Yes | 0 | Profile Status 1: Enable 0: Disable |  |
| biography | varchar(500) | Yes | *NULL* |  |  |
| linkedin\_id | varchar(45) | Yes | *NULL* |  |  |
| fiucs\_id | varchar(45) | Yes | *NULL* | FIU CS ID |  |
| google\_id | varchar(45) | Yes | *NULL* | Google ID |  |
| isAdmin | tinyint(1) | No | 0 | The user is administrator |  |
| isProMentor | tinyint(1) | No | 0 | The user is a Project Mentor |  |
| isPerMentor | tinyint(1) | No | 0 | The user is a Personal Mentor |  |
| isDomMentor | tinyint(1) | No | 0 | The user is a Domain Mentor |  |
| isStudent | tinyint(1) | No | 0 | The use is student |  |
| isMentee | tinyint(1) | No | 0 | The user is a Mentee |  |
| isJudge | tinyint(1) | No | 0 | The user is a Judge |  |
| isEmployer | tinyint(1) | No | 0 | The user is an Employeer |  |
| isNew | tinyint(4) | No |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 16 | A | No |  |
| username | BTREE | Yes | No | username | 16 | A | No |  |
|  |  |  |  | email | 16 | A | No |  |

#### user\_domain

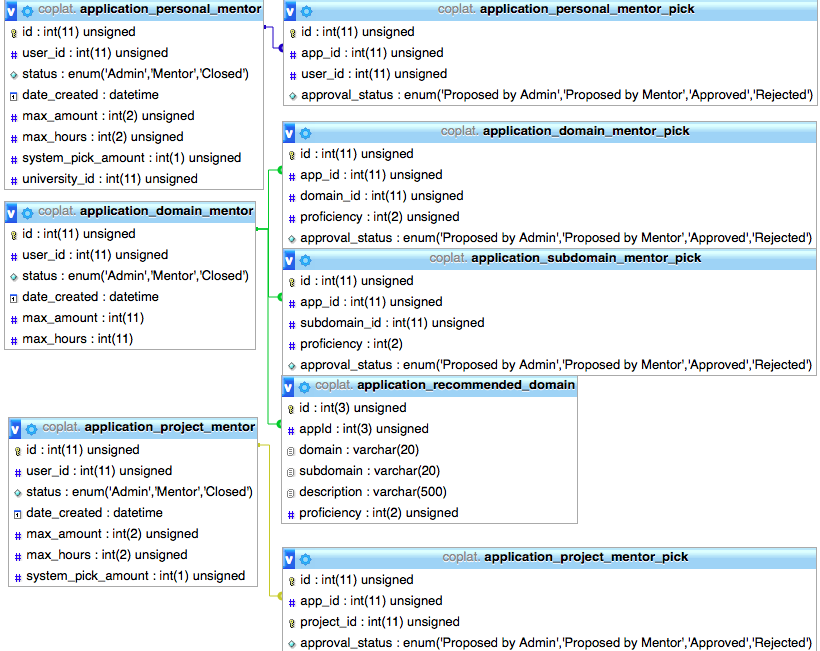
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Links to** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |  |
| user\_id | int(11) | No |  | user -> id |  |  |
| domain\_id | int(11) | No |  | domain -> id |  |  |
| subdomain\_id | int(11) | Yes | *NULL* | subdomain -> id |  |  |
| rate | int(11) | Yes | *NULL* |  |  |  |
| active | tinyint(1) | Yes | *NULL* |  |  |  |
| tier\_team | int(11) | Yes | *NULL* |  |  |  |

##### Indexes

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 7 | A | No |  |
| fk\_domain\_has\_user\_user1\_idx | BTREE | No | No | user\_id | 7 | A | No |  |
| fk\_domain\_has\_user\_domain1\_idx | BTREE | No | No | domain\_id | 1 | A | No |  |
| fk\_user\_domain\_subdomain1\_idx | BTREE | No | No | subdomain\_id | 2 | A | Yes |  |

Open new phpMyAdmin window

**Mentoring Module: Version 3**



## 

## 

## 5.5 Security/Privacy

**Yii access control rules**

The Yii framework provides access control with respect to any controller being used.  This access control will reject a subset of users (not logged, students, employers, etc…) from performing certain actions.  For example, users that are not logged in will not have access to profile pages.

**Security Features**

User password will be hashed in the database. Upon registration into the system, passwords entered will be hashed right away and will not be saved anywhere on the system.  Upon login, the password entered again will be hashed and the hashed data will be used to query the database.

**Cross-site Scripting Prevention**

The Yii framework takes measures against common web exploitations such as cross-site scripting or MySQL injection.  Using Yii, we can be rest assured that such things should not occur.

**Secure registration process**

The registration process is not as simple as most sites, especially for employers.  Administrators will have to verify employers after they register to ensure they are actual employers to ensure the integrity of the system.  Only then will they be able to post jobs and interact with students.

# Detailed Design

This section breaks down the design of the system into a higher level of detail, specifying exactly how each of the subsystems is broken down into associated classes, along with their attributes and inheritance hierarchy. It also lays out the dynamic model of the system by a series of sequence diagrams, and shows the method signatures of the main classes of each subsystem.

## 6.1 Overview

For each one of the subsystems presented in section 5.2, the Active Record pattern is the most evidently used. Each one of the model classes in each subsystem is an Active Record, i.e., they inherit from the Active Record Base class. This provides a very simple interface to the persistent data of the system, since each attribute of a class is mapped to a column of a row in a database table. The Active Records also come packed with a lot of useful functionality provided by the Yii Framework.

Refer to section 5.2 for a description of each subsystem

## 6.2 Static model.

Refer to appendix E for class diagrams

**Registration and Access Point Subsystems**

The registration and access point subsystem will be the main entry point for our platform. This will facilitate the process across the user database. Since there will be one login for each individual module, the access point is simplified into one subsystem. This subsystem will also cover the creation of new users and the retrieval of passwords for current users.

**Mentoring Subsystem**

The mentoring subsystem will cover the essentials of the mentoring module. Specifics to the mentoring module are separated into this subsystem. These functionalities include the essential workflow of the ticketing system for issues created by mentees for the different types of mentors.

**Communication and Notification Subsystem**

The mail subsystem structure relies heavily on Ajax to transfer JSON data between the client and the server. This type of interaction between the user and the system was implemented in most of the messaging use cases and results in a dynamic user experience (desktop-like), but it is a bit more difficult to implement than a pure Restful implementation for instance. Given the allotted time, it was decided that it could be implemented this way for the purposes of improving the user experience. The tradeoff was implementation time and implementation simplicity.

The notification subsystem acts as the mailman for the whole system. It is interconnected with all other subsystems to receive information and propagate that information through the system to the respective users. The notification subsystem is connected to the main controllers of all other subsystem. This connectivity allows it to keep track of all interactions between all the other subsystems and stay up to date with what is going on.

**Mentoring Module: Version 3**

**Self-serve Mentor Application**

The Mentor Application subsystem relied on the built in forms provided by Yii for the transfer of JSON objects between the client and the server. These forms do this through HTTP POST and GET requests. In order to stay faithful to the requirements the views for the application required a significant amount of customization. The widgets provided by Yii and Yii extensions were very rigid and did not allow for the amount of customization required. Instead custom widgets were created using JavaScript and jQuery to manipulate the DOM objects as was originally intended. All form validation is done client side using JavaScript and jQuery to check all the fields for valid information. The result proved to require more work than using a built in widget but provided a faithful realization of the requirements.

The Mentor Application subsystem creates an Application object that will be passed back and forth between a Mentor and an Admin. These Applications were designed for the purpose of being used by both types of users. Any selections made on Applications, such as selected Mentees for a Personal Mentorship, are stored in the corresponding Pick table. Yii models are not set up to handle multiple entries per row. These Pick tables act as a relational table that maps Applications to specific User, Project or Domain IDs.

## 

## 

## 6.3   Dynamic model

Refer to appendix E for sequence diagrams

**Access Point and Registration Subsystems**

The registration and access point subsystem will be the main entry point for our platform. This will facilitate the process across the user database. Since there will be one login for each individual module, the access point is simplified into one subsystem. This subsystem will also cover the creation of new users and the retrieval of passwords for current users.

**Mentoring Subsystem**

The mentoring subsystem will cover the essentials of the mentoring module. Specifics to the mentoring module are separated into this subsystem. These functionalities include the essential workflow of the ticketing system for issues created by users for the different types of mentors.

**Communication and Notification Subsystems**

The notification subsystem relies on the notifications table to function. Methods are created within model classes (mostly User) to handle events such as registration, message creation, and ticket flow. These methods create notifications and if necessary send out emails to users about the event. The subsystem also relies on the mail function which is provided by a library in Yii. Read notification is handled when a notification link is clicked. The link contains a parameter containing the notification id and upon page completion, the parameter will be picked up and the notification entry in the database will be marked as read. Delete notification simply deletes a row from the database.

For the messaging subsystem, there’s a main message table on the backend which holds information regarding a message along with foreign keys to the receiver and sender of the message. When a user selects a message on the client side, or clicks on the list of ‘sent’ messages for instance, an AJAX call gets executed and handled by a controller method who accesses the corresponding data in the model, transforms the data into JSON, and sends it back to the JavaScript method in a callback function.

Once the data is received by the client, it is processed by a number of JQUERY methods who extensively manipulate the DOM of the page to present this data while destroying previously requested content. This pattern of execution was used in most of the messaging use cases and it is necessary for dealing with the Ajax calls.

**Mentoring Module: Version 3**

**Self-serve Mentor Application**

Mentor Applications use the existing MVC architecture to fetch the information necessary for displaying a full application. Being routed to a specific HTML address is what triggers the corresponding action in a Controller. Based on the action triggered the Controller will fetch only the information it needs to display from any Models with the required data. This creates unique JSON objects for each type of Application. The Controller then renders the necessary View and passes along the JSON objects with it. Once an Application is submitted the page is refreshed triggering the same action once again. This time the Controller catches the POST request, creates/updates the necessary Models, and redirects the User to the appropriate page.

**Admin Management Tools**

The admin management pages that are provided utilizes the Yii Framework’s established MVC architecture to provide all the views. The controller is only activated upon the receiving a call from the Yii Router. The yii Router is activated only when the page is requested by the browser. Once the page is loaded and rendered any submissions using the activeForm class will again, call the controller but in a submit instance you would just need an if statement to catch if it is a POST. When you catch this you can then perform any required operations on the data that was captured.

## 6.4 Code Specification

Refer to Appendix F for code interfaces

# 

# 

# 7 System Validation

The system validation chapter introduces testing to make sure that the project works as was intended. For this specific phase of the project, we performed system testing and subsystem testing, which involves testing using the GUI and then going deep into code to go ahead and test it. Due to the nature of the Yii framework, it was especially troublesome to perform test on controllers. After much research we were able to overcome some of the limitation imposed by the framework in order to test the subsystem. For subsystem testing we used PHPUnit which was invaluable for our subsystem validation. In the following sections we have include the drivers and script for both system and subsystem.

## 7.1 System Tests

|  |  |
| --- | --- |
| Test Case ID | MP-TC01 |
| **Purpose** | Test successful login functionality |
| **Test Setup** | Username and password exist in the database that will enable the user to login. |
| **Input** | User navigates to Collaborative Platform site and clicks Login.  User enters username and password. |
| **Expected Output** | User is redirected to Home page |
| **Actual output** | User is redirected to Home page |

|  |  |
| --- | --- |
| Test Case ID | MP-TC02 |
| **Purpose** | Test unsuccessful login functionality |
| **Test Setup** | Username and password combination that user will input does not exist in the database |
| **Input** | User navigates to Collaborative Platform site and clicks Login.  User enters username and password that does not exist. |
| **Expected Output** | The system displays “Incorrect username or password.” |
| **Actual output** | The system displays “Incorrect username or password.” |

|  |  |
| --- | --- |
| Test Case ID | MP-TC03 |
| **Purpose** | Test logout |
| **Test Setup** | User is logged in the system |
| **Input** | User clicks “Logout” |
| **Expected Output** | User is redirected to Main Page |
| **Actual output** | User is redirected to Main Page |

|  |  |
| --- | --- |
| Test Case ID | MP-TC04 |
| **Purpose** | Test user edit profile |
| **Test Setup** | User is logged in as Project Mentor and located at profile page |
| **Input** | User clicks ‘Edit’ and Enters change the number of projects from ‘1’ to ‘2’. |
| **Expected Output** | Redirect to profile view with changes on the number of projects in the Project Mentor section |
| **Actual output** | Redirect to profile view with changes on the number of projects in the Project Mentor section |

|  |  |
| --- | --- |
| Test Case ID | MP-TC05 |
| **Purpose** | Test domain mentor add domain |
| **Test Setup** | User is logged in as domain mentor and he is in the profile page |
| **Input** | User enters required information and information is valid |
| **Expected Output** | Redirect to profile view with new domain in the Domain Mentor section |
| **Actual output** | Redirect to profile view with new domain in the Domain Mentor section |

|  |  |
| --- | --- |
| Test Case ID | MP-TC06 |
| **Purpose** | Test student change password successfully |
| **Test Setup** | User is logged in as student and has clicked ‘change password’ link |
| **Input** | User enters old password, new password, and new password confirmation correctly |
| **Expected Output** | User is redirected to profile page |
| **Actual output** | User is redirected to profile page |

|  |  |
| --- | --- |
| Test Case ID | MP-TC07 |
| **Purpose** | Test user change password unsuccessfully |
| **Test Setup** | User is logged in as user and has clicked ‘change password’ link |
| **Input** | User enters old password (incorrectly), new password, and new password confirmation |
| **Expected Output** | Page displays ‘Old password was incorrect’ |
| **Actual output** | Page displays ‘Old password was incorrect’ |

|  |  |
| --- | --- |
| Test Case ID | MP-TC16 |
| **Purpose** | Test domain mentor tier 1 escalate a ticket successfully |
| **Test Setup** | Domain mentor is logged and located at ticket page. |
| **Input** | Domain mentor click ‘escalate’ button |
| **Expected Output** | Page displays dashboard page where a new ticket is going to appear, The creator of the ticket is the mentor in tier 1 and is assigned to a mentor in tier 2. The new ticket has all the history from the old ticket and a new comment automatically created by the system linking the original ticket with the new one. |
| **Actual output** | Page displays dashboard page where a new ticket is going to appear, The creator of the ticket is the mentor in tier 1 and is assigned to a mentor in tier 2. The new ticket has all the history from the old ticket and a new comment automatically created by the system linking the original ticket with the new one. |

|  |  |
| --- | --- |
| Test Case ID | MP-TC17 |
| **Purpose** | This test verifies if data (students and projects) is brought automatically and correctly from SPW |
| **Test Setup** | Remove all SP students (mentees) and projects from database |
| **Input** | The automated job runs at midnight |
| **Expected Output** | Manage user screen shows new mentees  Manage projects screen show new projects |
| **Actual output** | Manage user screen shows new mentees  Manage projects screen show new projects |

|  |  |
| --- | --- |
| Test Case ID | MP-TC18 |
| **Purpose** | This test verifies if data (students and projects) is brought manually and correctly from SPW |
| **Test Setup** | Remove all SP students (mentees) and projects from database |
| **Input** | The admin clicks on “Sync with SPW” |
| **Expected Output** | Manage user screen shows new mentees  Manage projects screen show new projects |
| **Actual output** | Manage user screen shows new mentees  Manage projects screen show new projects |

|  |  |
| --- | --- |
| Test Case ID | MP-TC20 |
| **Purpose** | Test admin edit priority |
| **Test Setup** | User “admin” is logged in as administrator and located in the priority page |
| **Input** | 1. User clicks ‘Edit’ and change the number of hours in priority low from ‘24’ to ‘48’. 2. Admin clicks in save button. |
| **Expected Output** | Redirect to priority page and time interval for low priority is 48 |
| **Actual output** | Redirect to priority page and time interval for low priority is 48 |

## 

## 

## 7.2 Subsystem Test

Due to time constraints for the project, subsystem testing was omitted. However, the nature of the testing tool, Selenium, enabled us to test the subsystems while performing system testing. We present the test cases in the next section and provide the test code in Appendix G.

|  |  |
| --- | --- |
| Test Case ID | MP-TC08 |
| **Purpose** | This test verifies that the system correctly autocompletes the given input |
| **Test Setup** | Create a user with username “test” |
| **Input** | $\_GET ['term'] = 'Lor'; |
| **Expected Output** | “Lorenzo Sanchez” |
| **Actual output** | “Lorenzo Sanchez” |

|  |  |
| --- | --- |
| Test Case ID | MP-TC09 |
| **Purpose** | This test verifies that the system correctly autocompletes the given input |
| **Test Setup** | Create a user with username “test” |
| **Input** | $\_GET ['term'] = 'xxxxxxxx'; |
| **Expected Output** | “” |
| **Actual output** | “” |

|  |  |
| --- | --- |
| Test Case ID | MP-TC10 |
| **Purpose** | This test verifies that the system correctly checks for valid screenName |
| **Test Setup** | No user in database with username “John” |
| **Input** | $\_POST ['username'] = 'john'; $\_POST ['Body'] ['Message'] = "hello"; |
| **Expected Output** | "User does not exist. <br />" |
| **Actual output** | "User does not exist. <br />" |

|  |  |
| --- | --- |
| Test Case ID | MP-TC11 |
| **Purpose** | This test verifies that the system correctly validates message values |
| **Test Setup** | Create a user with username “Peter”  Set user as not validated |
| **Input** | $\_POST ['username'] = 'Peter' ;$\_POST ['Body'] ['Message'] = ""; |
| **Expected Output** | "Please enter a message to be sent. No blank messages allowed. <br /> " |
| **Actual output** | "Please enter a message to be sent. No blank messages allowed. <br /> " |

|  |  |
| --- | --- |
| Test Case ID | MP-TC12 |
| **Purpose** | This test verifies that users can successfully login using valid credentials |
| **Test Setup** | set user |
| **Input** | Username: admin password: admin |
| **Expected Output** | User->admin > 5 |
| **Actual output** | User->admin > 5 |

|  |  |
| --- | --- |
| Test Case ID | MP-TC13 |
| **Purpose** | This test verifies that register do not have a previous account in the system |
| **Test Setup** | set valid user with an account |
| **Input** | $\_POST ['username'] = 'hmuni006';  $\_POST ['email'] = "hmuni006@fiu.edu"; |
| **Expected Output** | User email is already linked with another account. |
| **Actual output** | User email is already linked with another account. |

|  |  |
| --- | --- |
| Test Case ID | MP-TC15 |
| **Purpose** | This test verifies that the system does not allow empty To: section on the Messages |
| **Test Setup** | User is logged in as an administrator  User write a messages to user ‘Henry Muniz’ |
| **Input** | To: “”  Message: “Hello Henry” |
| **Expected Output** | "To: field is in the wrong format” |
| **Actual output** | "To: field is in the wrong format” |

|  |  |
| --- | --- |
| Test Case ID | MP-TC19 |
| **Purpose** | This test verifies if the admin is able to register mentors |
| **Test Setup** | The admin is on the “Add user” screen |
| **Input** | Admin inputs information needed to register a mentors |
| **Expected Output** | New user receives email with credentials  Users appear in the list of users |
| **Actual output** | New user receives email with credentials  Users appear in the list of users |

**Mentor Module: Version 3**

Version 3 focused solely on subsystem development and improvement. Thus, only subsystem tests were done.

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC01 |
| **Purpose** | This test verifies if the invitation email routes Users to the correct page |
| **Test Setup** | * The Admin has sent the User an invitation email * The User is logged into his email account * The User has the email open |
| **Input** | The user clicks on the provided link |
| **Expected Output** | The Collaborative Platform landing page is loaded on the browser |
| **Actual output** | The Collaborative Platform landing page is loaded on the browser |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC02 |
| **Purpose** | This test verifies if a Mentor account is created when a User fills out the registration form correctly |
| **Test Setup** | * The User has correctly filled out the registration form without errors * The User clicks submit on the verification modal |
| **Input** | $\_POST([‘User’]) && $\_POST([‘UserInfo’]) |
| **Expected Output** | The new Mentor account is created, the Mentor is logged in, and the Mentor Application Portal is loaded |
| **Actual output** | The new Mentor account is created, the Mentor is logged in, and the Mentor Application Portal is loaded |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC03 |
| **Purpose** | This test verifies if the system correctly recognizes mismatched passwords on the Account Registration form |
| **Test Setup** | * The User has entered two different passwords for the Password and Re-Password fields * The User clicks next |
| **Input** | Password:”linguini”, Re-Password: “spaghetti” |
| **Expected Output** | Error: “Passwords do not match” |
| **Actual output** | Error: ”Passwords do not match” |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC04 |
| **Purpose** | This test verifies if the system correctly catches a duplicate username on the Account Registration form |
| **Test Setup** | * A User with the username “demo” exists in the system * The new User has entered “demo” for his username * The User clicks submit on the verification modal |
| **Input** | username:”demo” |
| **Expected Output** | The system does not create the account. The system reloads the page.  Error: “The username or email entered has already been taken” |
| **Actual output** | The system does not create the account. The system reloads the page.  Error: “” |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC05 |
| **Purpose** | This test verifies if a fully completed Personal Mentor Application is correctly saved by the system |
| **Test Setup** | * The Mentor is on the Personal Mentor Application page * The Mentor has selected 2 students from the list and has elected to have 1 student from FIU assigned to him * The User clicks submit on the verification modal |
| **Input** | hiddeninput:”11,9”  system\_pick\_amount: 1  Preferred University: “Florida International University” |
| **Expected Output** | The system creates the application. The corresponding application picks table has 3 new entries with the User IDs 11, 9 and one random User ID from an FIU student all mapped to the new application ID. The Mentor is redirected to the Mentor Application Portal and the Personal Mentor Application reads “Pending...” |
| **Actual output** | The system creates the application. The corresponding application picks table has 3 new entries with the User IDs 11, 9 and one random User ID from an FIU student all mapped to the new application ID. The Mentor is redirected to the Mentor Application Portal and the Personal Mentor Application reads “Pending...” |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC06 |
| **Purpose** | This test verifies if the system correctly catches an empty Personal Mentor Application Form |
| **Test Setup** | * The Mentor is on the Personal Mentor Application page * The Mentor has not selected any Mentee from the list and has elected to have 0 Mentees assigned by the system * The Mentor clicks Next |
| **Input** | hiddeninput:””, system\_pick\_amount = 0 |
| **Expected Output** | Error: “Must select a Mentee from the list or have the system pick one for you” |
| **Actual output** | The system does not create the account. The system reloads the page.  Error: “Must select a Mentee from the list or have the system pick one for you” |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC07 |
| **Purpose** | This test verifies if the system correctly handles a Project Mentor Form with only system picks |
| **Test Setup** | * The Mentor is on the Project Mentor Application page * The Mentor has not selected any Mentee from the list * The Mentor has elected to have 1 Mentee assigned by the system * The Mentor clicks Next |
| **Input** | hiddeninput:””, system\_pick\_amount = 1 |
| **Expected Output** | The verification modal is loaded.  Message: “You have selected NO projects. The system will pick 1 project for you” |
| **Actual output** | The verification modal is loaded.  Message: “You have selected NO projects. The system will pick 1 project for you” |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC08 |
| **Purpose** | This test verifies if a fully completed Domain Mentor Application is correctly saved by the system |
| **Test Setup** | * The Mentor is on the Domain Mentor Application page * The Mentor has set the proficiency for 2 domains from the list and 1 subdomain from the list * The Mentor clicks submit on the verification modal |
| **Input** | hiddendom:”8:7,10:7”  hiddensub:”1:8”  max\_tickets: “10” |
| **Expected Output** | The system creates the application and has a value of 10 for max\_amount. The domain application picks table has 2 new entries with the User IDs 18 and 7 with the proficiencies set to 7 and 7. The subdomain application picks table has 1 new entry with the User ID 1 with the proficiency set to 8 . The Mentor is redirected to the Mentor Application Portal and the Domain Mentor Application reads “Pending...” |
| **Actual output** | The system creates the application and has a value of 10 for max\_amount. The domain application picks table has 2 new entries with the User IDs 18 and 7 with the proficiencies set to 7 and 7. The subdomain application picks table has 1 new entry with the User ID 1 with the proficiency set to 8 . The Mentor is redirected to the Mentor Application Portal and the Domain Mentor Application reads “Pending...” |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC09 |
| **Purpose** | This test verifies if clicking an application labeled pending does not load an application page |
| **Test Setup** | * The Mentor is on the Mentor Application Portal page * The Mentor has a pending Personal Mentor Application * The Mentor clicks on the Personal Mentor image |
| **Input** | -- |
| **Expected Output** | No page is loaded, the Mentor is still at the Mentor Application Portal |
| **Actual output** | No page is loaded, the Mentor is still at the Mentor Application Portal |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC10 |
| **Purpose** | This test verifies that the data being pulled into the User Modal is consistent with the database. |
| **Test Setup** | * The mentor “Demo Demo” has all three roles (Project, Domain, Personal). The database has been set so that Demo is mentoring the following Project : Collaborative Platform and Senior Project, Domain: Programming Languages (proficiency 5) and Programming Languages, C (proficiency 4) Personal Mentor: Nicholas Madariaga and Jonathan Sanchez * Admin is on the manage->users page. |
| **Input** | Click on the view details button next to “Demo Demo” |
| **Expected Output** | The roles for Project List: Collaborative PLatform and Senior Project  Domain List: Programming Languages - Proficiency 5, Programming Languages: C - Proficiency 4  Personal List : Nicholas Maragia and Jonathan Sanchez |
| **Actual output** | The roles for Project List: Collaborative PLatform and Senior Project  Domain List: Programming Languages - Proficiency 5, Programming Languages: C - Proficiency 4  Personal List : Nicholas Maragia and Jonathan Sanchez |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC11 |
| **Purpose** | This test verifies that the project modal is displaying the correct data. |
| **Test Setup** | * The coplat database is set to have collaborative platform mentor module project with the following project mentor “Masoud Sadjadi” and 3 tickets with default data on all columns except they all originate from “Jonathan Sanchez” who is a member of the project |
| **Input** | Click on the eyeball next to the project name of “Collaborative Platform: Mentor Module” |
| **Expected Output** | The modal is rendered and the data within the modal contains that “Jonathan Sanchez “ is a member of the project, and that there are 3 tickets currency pending. |
| **Actual output** | The modal is rendered and the data within the modal contains that “Jonathan Sanchez “ is a member of the project, and that there are 3 tickets currency pending. |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC12 |
| **Purpose** | This test verifies that all items that have been approved were updated to their correct status and have been made live into system. The parent application is updated to show “closed” |
| **Test Setup** | * The mentor “Demo Demo” has an application for all three roles. * Project Applications : Collaborative Platform : Mentor Module * Personal Mentor: Jonathan Sanchez * Domain Mentor: “Programming Languages : C” * Admin is in applications page |
| **Input** | The admin selects accept on the 3 items and then the submit button. |
| **Expected Output** | The status for the pick (3 items) is changed to Approved and an entry is made into the personal\_mento\_mentees table for Mentor Demo and User Jonathan Sanchez The user\_info table contains an entry for Programming languages and C as the subdomain. The project mentor contains an entry in project\_mentor\_mentors for collaborative platform and has Demo Demo as one of the listed. The parent application is also set to closed because there are no pending items. |
| **Actual output** | The status for the pick (3 items) is changed to Approved and an entry is made into the personal\_mento\_mentees table for Mentor Demo and User Jonathan Sanchez The user\_info table contains an entry for Programming languages and C as the subdomain. The project mentor contains an entry in project\_mentor\_mentors for collaborative platform and has Demo Demo as one of the listed. The parent application is also set to closed because there are no pending items. |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC13 |
| **Purpose** | This test verifies that the user has been updated to reflect that they now have a project to mentor and thus have a 1 in their flag for isProMentor from the user table.. |
| **Test Setup** | * The mentor “Demo Demo” has an application for project mentor * Project Applications : Collaborative Platform : Mentor Module * Admin is in applications page |
| **Input** | The admin selects accept on the entry specified above and then submit. |
| **Expected Output** | The isProMentor flag has been set to 1 in the user table of the database. |
| **Actual output** | The isProMentor flag is still 0 in the user table of the database. |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC14 |
| **Purpose** | This test verifies that the picks in an applications have been updated correctly when rejected. |
| **Test Setup** | * The mentor “Demo Demo” has an application for project mentor * Project Applications : Collaborative Platform : Mentor Module * Domain Mentor: “Programming Languages : C” * Admin is in applications page |
| **Input** | The admin selects “reject” on all the options and clicks “submit” |
| **Expected Output** | The picks have been updated to “Rejected” and the parent application has been updated to “Closed” |
| **Actual output** | The picks have been updated to “Rejected” and the parent application has been updated to “Closed” |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC15 |
| **Purpose** | This test verifies that the picks in an applications have been updated correctly when rejected. |
| **Test Setup** | * The mentor “Demo Demo” has an application for project mentor * Project Applications : Collaborative Platform : Mentor Module * Domain Mentor: “Programming Languages : C” * Admin is in applications page |
| **Input** | The admin selects “reject” on all the options and clicks “submit” |
| **Expected Output** | The picks have been updated to “Rejected” and the parent application has been updated to “Closed” |
| **Actual output** | The picks have been updated to “Rejected” and the parent application has been updated to “Closed” |

|  |  |
| --- | --- |
| Test Case ID | CPv3-TC16 |
| **Purpose** | This test verifies that the customized invite has been emailed to the invitee and the user is added to the invitation table |
| **Test Setup** | The admin has entered the invite page and selected create new invite  admin entered Jonathan Sanchez for named and [jsanc090@fiu.edu](mailto:jsanc090@fiu.edu) for email and selects the mentor. Then the admin adds “COOL” to the displayed template. |
| **Input** | The admin clicks create new invite.  enters jonathan sanchez for name  enters [jsanc090@fiu.edu](mailto:jsanc090@fiu.edu) for email and selects NEXT |
| **Expected Output** | Hello  The Collaborative Platform system administrator, Masoud Sadjadi, through this email would like to invite you to participate on it as:  Mentor  Domain Mentor: Provide solutions using his/her expertise in specific domains to questions within the platform.  Project Mentor: Guide the project development through the semester.  Personal Mentor: Provide assistance and guidance to his/her mentees.  Click here to access the platform. COOL |
| **Actual output** | Hello  The Collaborative Platform system administrator, Masoud Sadjadi, through this email would like to invite you to participate on it as:  Mentor  Domain Mentor: Provide solutions using his/her expertise in specific domains to questions within the platform.  Project Mentor: Guide the project development through the semester.  Personal Mentor: Provide assistance and guidance to his/her mentees.  Click here to access the platform. COOL |

## 7.3 Evaluation of Tests

**System**

|  |  |
| --- | --- |
| Identification | Result |
| MP-TC01 | PASS |
| MP-TC02 | PASS |
| MP-TC03 | PASS |
| MP-TC04 | PASS |
| MP-TC05 | PASS |
| MP-TC06 | PASS |
| MP-TC07 | PASS |
| MP-TC16 | PASS |
| MP-TC17 | PASS |
| MP-TC18 | PASS |
| MP-TC20 | PASS |

**Subsystem**

|  |  |
| --- | --- |
| Identification | Result |
| MP-TC08 | PASS |
| MP-TC09 | PASS |
| MP-TC10 | PASS |
| MP-TC11 | PASS |
| MP-TC12 | PASS |
| MP-TC13 | PASS |
| MP-TC15 | PASS |
| MP-TC20 | PASS |

# 

# 

# Mentoring Module: Version 3 - Test Case Evaluation

|  |  |
| --- | --- |
| Identification | Result |
| CPv3-TC01 | PASS |
| CPv3-TC02 | PASS |
| CPv3-TC03 | PASS |
| CPv3-TC04 | FAIL |
| CPv3-TC05 | PASS |
| CPv3-TC06 | PASS |
| CPv3-TC07 | PASS |
| CPv3-TC08 | PASS |
| CPv3-TC09 | PASS |
| CPv3-TC10 | PASS |
| CPv3-TC11 | PASS |
| CPv3-TC12 | PASS |
| CPv3-TC13 | FAIL |
| CPv3-TC14 | PASS |
| CPv3-TC15 | PASS |
| CPv3-TC16 | PASS |

# 

# 

# 8. Glossary

Definitions:

* **User** – A generalization given to all user groups.
* **Student** – Individual who is currently enrolled in the senior project class.
* **Mentorship** - is a personal developmental relationship in which a more experienced or more knowledgeable person helps to guide a less experienced or less knowledgeable person.
* **Mentor** - An Individual with a Collaborative Platform account seeking Mentorship.
* **Project** **Mentor** – An individual who is assigned to overlook one or many senior projects.
* **Domain** **Mentor**– An individual with particular knowledge in a certain skill or language(2 Tiers)
* **Personal Mentor** –An individual that mentors only a single person – not limited to career advice.
* **Mentee** - An individual that may request help or ask a question.
* **Administrator** –An individual assigned with administering the web portal.
* **Interviewer** – An individual from industry that can remotely interview potential employees.
* **Interviewee** – An individual and future prospect that may engage in the virtual job fair portal.
* **Application** - A form containing information provided by a Mentor seeking a Mentorship.
* **System Pick** - An automated selection done by the system.
* **Project** - A collaborative enterprise composed of students with the purpose of achieving a goal. Currently only refers to projects from the Senior Project Website.

Acronyms:

* **SCIS** – School of Computing & Information Sciences
* **FIU** – Florida International University
* **ECS** – Engineering & Computing Sciences (Building)
* **JCCL** – John C. Comfort Laboratory
* **VJF** – Virtual Job Fair
* **MJ** –Mobile Judge
* **WP** – Web Platform
* **T1**- Tier 1 Domain Mentor
* **T2**- Tier 2 Domain Mentor
* **PM**- Project Mentor
* **DBMS** – Database Management System

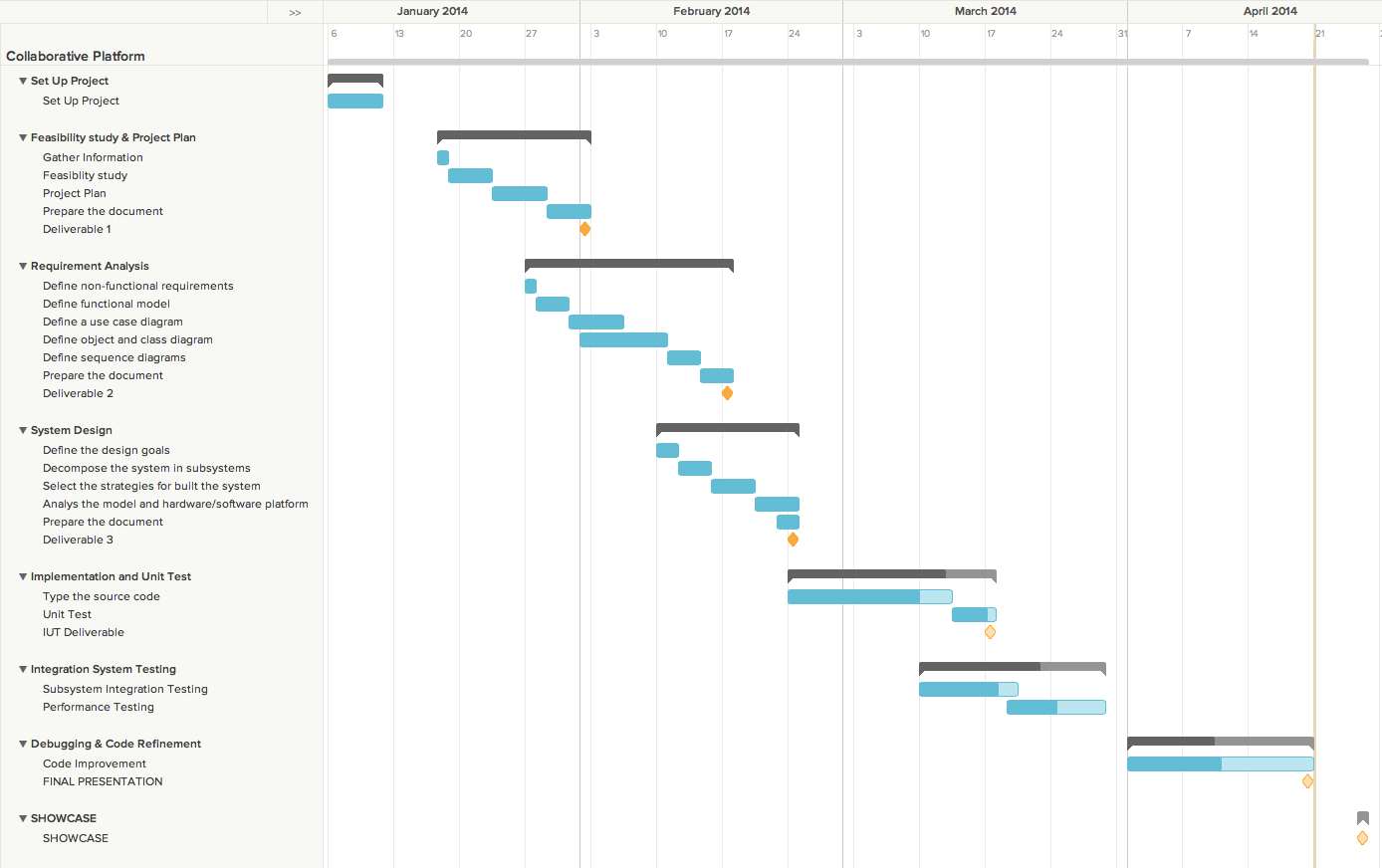
Abbreviations:

* **N/A**

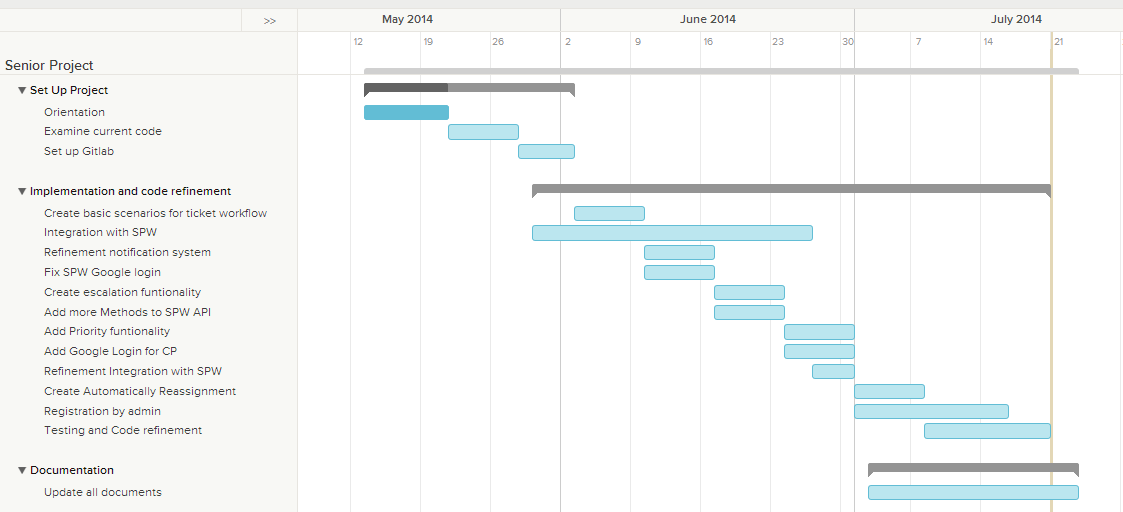
# 9. Appendix

## 9.1 Appendix A - Project schedule (Gantt chart or PERT chart).

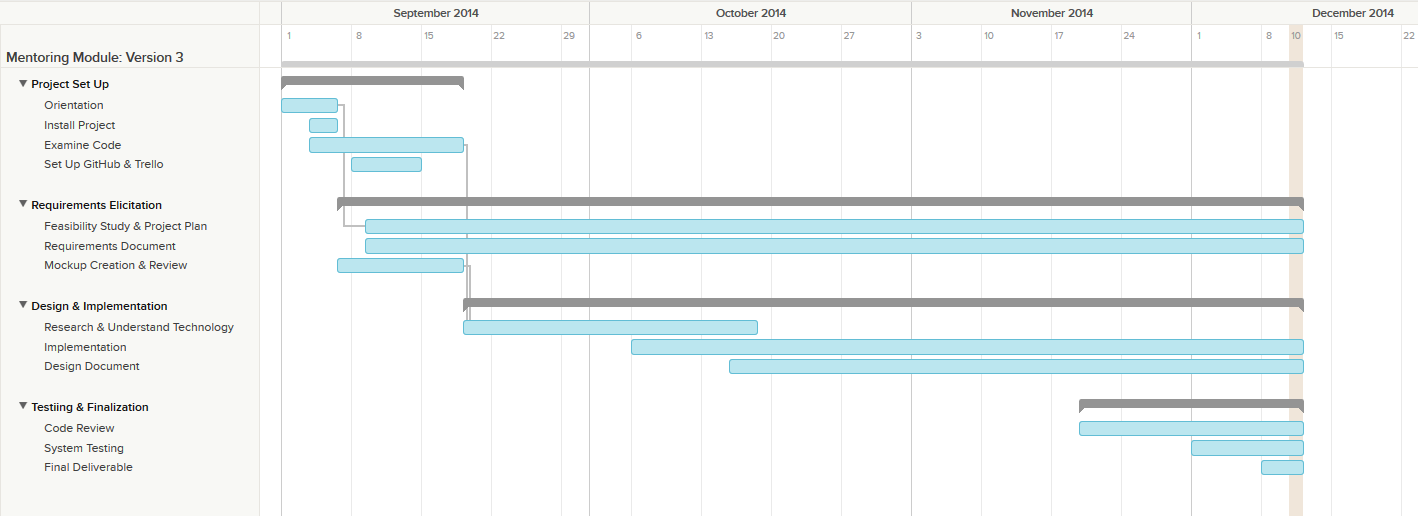
**Spring 2014**



**Summer 2014**



**Fall 2014**



## 

## 

## 9.2 Appendix B - All use cases with nonfunctional requirements

|  |  |
| --- | --- |
| Use Case ID | MP - Login |
| Description | Allow User to Login |
| Actor | User |
| Pre-conditions | * User is at the main login page |
| Steps | 1. Use case begins when the user enters their credentials. 2. User clicks the login button 3. Use case ends when the user is authenticated and logged in. |
| Post-conditions | The system checks for the user in the database, and then checks credentials |

|  |  |
| --- | --- |
| Use Case ID | MP – Logout |
| Description | Allow User to Logout |
| Actor | User |
| Pre-conditions | * User is logged into the system |
| Steps | 1. Use case begins when the user clicks on the logout button 2. System logs out the user 3. Use case ends when the user is redirected to the homepage. |
| Post-conditions | System refreshes permissions to no one actively online |

|  |  |
| --- | --- |
| Use Case ID | MP - Register |
| Description | Allow User to Register |
| Actor | User |
| Pre-conditions | * User is at the home page |
| Steps | 1. Use case begins when the user clicks on register 2. User enters registration info 3. User clicks on register 4. System fills database with new user 5. Use case ends when the user is redirected to waiting for authorization page |
| Post-conditions | The system sends a validation email to the SA for approval |

|  |  |
| --- | --- |
| Use Case ID | MP – Add Domain |
| Description | Allow Domain User to add domain |
| Actor | Domain Mentor |
| Pre-conditions | * Domain Mentor is at their editProfile |
| Steps | 1. Use case begins when the mentor chooses an available domain. 2. Mentor drags down and chooses an expertise level for the domain. 3. Use case ends when the mentor has the newly added domain on their table. |
| Post-conditions | The system checks the database and adds it to their respective table. |

|  |  |
| --- | --- |
| Use Case ID | MP – Forgot Password |
| Description | Allow User to retrieve forgotten password |
| Actor | User |
| Pre-conditions | * User is at the main login page |
| Steps | 1. Use case begins when the user clicks on forgot password 2. User enters their email address used to register 3. System sends the user a link with their new password 4. Use case ends when the user is given their password. |
| Post-conditions | The system checks and retrieves the password for the user. |

|  |  |
| --- | --- |
| Use Case ID | MP – Edit Bio |
| Description | Allow user to edit their biography |
| Actor | User |
| Pre-conditions | * User is at their profile page |
| Steps | 1. Use case begins when the user clicks on the edit button. 2. User types in new text into their biography 3. Use case ends when the user clicks save. |
| Post-conditions | The users profile will be refreshed with their new biography. |

|  |  |
| --- | --- |
| Use Case ID | MP – Edit Availabilty |
| Description | Allow mentors to change their availabilities |
| Actor | Mentors |
| Pre-conditions | * Mentors are at their profile page |
| Steps | 1. Use case begins when the mentor clicks on edit. 2. Mentor changes their availabilities for the respective hours and project or tickets. 3. Use case ends when the mentor clicks on save. |
| Post-conditions | The users profile will be refreshed with their new availability. |

|  |  |
| --- | --- |
| Use Case ID | MP- Send Message |
| Description | Send a message to a user |
| Actor | User |
| Pre-conditions | * User is in the compose message page |
| Steps | 1. User populates the ‘To’ field 2. User populates the ‘Subject’ field 3. User types in the message in the text area |
| Post-conditions | The system sends the message. The message appears in the inbox of the target user |
| Exceptions | * Inexistent username selected as the receiver of the message * Wrong username format typed in the ‘To’ field |

|  |  |
| --- | --- |
| Use Case ID | MP- Get Inbox Messages |
| Description | User requests to see all the received messages |
| Actor | User |
| Pre-conditions | * The user is logged in |
| Steps | 1. The User navigates to the message page |
| Post-conditions | User is shown with a list of all the trashed messages in the  format: <Sender> <Subject> |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | MP- Get Sent Messages |
| Description | User requests to see all the messages he/she has sent |
| Actor | User |
| Pre-conditions | * The user is logged in |
| Steps | User selects the “Sent” messages from the messages page |
| Post-conditions | User is shown with a list of all the trashed messages in the  format: <Receiver> <Subject> |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | MP- Get Trashed Messages |
| Description | User requests to see all the trashed messages |
| Actor | User |
| Pre-conditions | * The user is logged in |
| Steps | User selects the ‘Trash” messages from the messages page |
| Post-conditions | User is shown with a list of all the trashed messages in the  format: <Sender/Receiver> <Subject> |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | MP-Replay to Message |
| Description | Reply to a message from some user |
| Actor | User |
| Pre-conditions | * User has selected a message to be seen |
| Steps | 1. User clicks on the Reply button that appears when reading a message |
| Post-conditions | The System redirects the user to the compose a message page, and the original message the user had selected appears in the text area in the format:  On <Date> <User> wrote:  <message> |
| Exceptions | None |

|  |  |
| --- | --- |
| Use Case ID | MP-Delete Messages |
| Description | User checks all the messages he/she wants to send to the trash |
| Actor | User |
| Pre-conditions | * User is logged in |
| Steps | 1. User checks the messages to be sent to the trash 2. User clicks on the trash icon |
| Post-conditions | The selected messages are sent to the trash |
| Exceptions | User does not select any messages before clicking on the trash icon. The System invokes an alert message |

|  |  |
| --- | --- |
| Use Case ID | MP- Send Meeting Notification |
| Description | Send email to mentee |
| Actor | System |
| Pre-conditions | * A Project Mentor or Personal Mentor has schedule a meeting |
| Steps | 1. System generate email 2. System send email to mentee |
| Post-conditions | The mentee will receive an email with the meeting information (date and time) |
| Exceptions | * The server mail service is down. |

|  |  |
| --- | --- |
| Use Case ID | MP- Send Ticket Reassigned Notification |
| Description | Send email to users related to the ticket |
| Actor | System |
| Pre-conditions | * A Mentee, Domain Mentor or the Administrator reassigned the ticket |
| Steps | 1. System generates email 2. System send email to all the users related to the ticket excluding the one that performed the re-assignation |
| Post-conditions | The users will receive an email with the re-assignation reason |
| Exceptions | * The server mail service is down. |

|  |  |
| --- | --- |
| Use Case ID | MP- Send Ticket Assigned Notification |
| Description | Send email to user to whom the ticket was assigned |
| Actor | System |
| Pre-conditions | * The User create a ticket * Ticket gets assign to an specific Domain Mentor (the user) or out-assign to the first available Domain Mentor |
| Steps | 1. System generates email 2. System send email to the Domain Mentor to whom the ticket was assigned |
| Post-conditions | The users will receive a notification email with the ticket information(id and subject) |
| Exceptions | * The server mail service is down. |

|  |  |
| --- | --- |
| Use Case ID | MP- Send Comment Added Notification |
| Description | Send email to users related to the ticket |
| Actor | System |
| Pre-conditions | * The User add a comment to the ticket |
| Steps | 1. System generates email 2. System send email to the users associated to the ticket excluding the comment originator. |
| Post-conditions | The users will receive an email with comment content |
| Exceptions | * The server mail service is down. |

|  |  |
| --- | --- |
| Use Case ID | MP- Send Invitation Notification |
| Description | Send email to the user that was invited to join the platform |
| Actor | System |
| Pre-conditions | * The System Administrator created an Invitation |
| Steps | 1. System generates email 2. System sends email to the email specified by the administrator on the Invitation. |
| Post-conditions | The users will receive an email with description of all the available roles in the Collaborative Platform |
| Exceptions | * The server mail service is down. |

|  |  |
| --- | --- |
| Use Case ID | MP-Automatically assign ticket |
| Description | The system will assign the most expert for the domain specified by the user. |
| Actor | System |
| Pre-conditions | * The User has been specified the domain and/or subdomain. * The user did not specify the mentor field in the ticket form. * The user has been clicked the submit button |
| Steps | 1. The System validates the inputs 2. The system verified if the mentor was selected 3. The System verified if domain and subdomain where specified. 4. The System search for the most expert domain mentor. 5. The use case finish when the system returns the domain mentor |
| Post-conditions | 1. The Ticket can be created along with the domain mentor who was assigned to. 2. The system send a notification to the assigned domain mentor |
| Exceptions | 1. No domain mentors available to take care the ticket |

|  |  |
| --- | --- |
| Use Case ID | MP-Reassign ticket |
| Description | The administrator is able to reassign a ticket, when this was rejected by its creator. |
| Actor | Administrator |
| Pre-conditions | * The Ticket creator has been change the status of the ticket to reject. * The ticket is in the administrator open queue. |
| Steps | 1. The administrator logs in to the collaborative platform 2. The administrator click on the ticket which status is reject in order to see the details. 3. The administrator click on the change status button. 4. In the form change status form, the administrator selects the desire domain mentor and send a message to this domain mentor. 5. The use case finish when the administrator click on Ok in the warning that pop ups. Alerting if he really wants to change the status of the ticket. |
| Post-conditions | 1. The ticket status is change to pending. 2. The ticket is reassign to a new domain mentor. 3. A notification is send to the creator and the new domain mentor, as well to the old domain mentor. |
| Exceptions | 1. Could be an error while the action is trying to access the DB. |

|  |  |
| --- | --- |
| Use Case ID | MP-Retrieve the project information |
| Description | Project Mentor will be able to see the description and the mentees enrolled in the project that has been assigned. |
| Actor | Project Mentor |
| Pre-conditions | * The Project Mentor has been assigned a project * The Project has been logged in to the system |
| Steps | 1. The use case begin when the project mentor clicks on the Project Mentor Button. 2. The system creates a new Project Meeting view with the projects where the Project mentor is assigned. 3. The Administrator click on the more popover to see the description and the mentees member of the project 4. The use case finish when the project mentor close the view |
| Post-conditions | 1. The information is displayed allowing the project mentor to see the details |
| Exceptions | 1. The Project mentor was not assigned to any project. |

|  |  |
| --- | --- |
| Use Case ID | MP-Set up meetings |
| Description | Project mentors will be able to set up meetings with his/her mentees. |
| Actor | Project Mentor |
| Pre-conditions | * The Project Mentor has been assigned a project * The Project has been logged in to the system * The Project mentor has clicked on the project mentor button |
| Steps | 1. The use case begins when the project mentor clicks on the set up meetings button. 2. The system provides a form with a dropdown list with all the mentees assigned to him, the calendar and time. 3. The project mentor selects the mentee, select the day from the calendar and the hour. 4. The system ask if He is sure to create a meeting. 5. The use case finish when the project mentor clicks on the Ok button. |
| Post-conditions | 1. The system creates a meeting. 2. The system notifies the mentee that a meeting has been set up. |
| Exceptions | 1. The Project mentor do not have any mentee. |

|  |  |
| --- | --- |
| Use Case ID | MP-Retrieve all the upcoming meetings |
| Description | The administrator will be able to monitor all the meetings set up by the project mentors. |
| Actor | Administrator |
| Pre-conditions | * Meetings has been set up by the Project mentors. * The administrator has been logged in to the system |
| Steps | 1. The use case begins when the administrator clicks on the manage project mentor button. 2. The system provides a view with the list of all the meetings set up in the system. 3. The use case finish when the administrator close the view |
| Post-conditions | 1. Administrator is able to see all the meetings created in the system. |
| Exceptions | 1. No meetings has been created. |

|  |  |
| --- | --- |
| Use Case ID | MP-Assign ticket to other project mentor. |
| Description | The Project mentor is able to assign a ticket directly to another project mentor |
| Actor | Project Mentor |
| Pre-conditions | * Project Mentor has logged into the system * Project Mentor has click in the New Ticket button. |
| Steps | 1. The use case begins when the mentee click on the submit button 2. The System validates the inputs 3. The system verified if the project mentor was selected 4. The System search for the project mentor information 5. The use case finish when the system returns the project mentor |
| Post-conditions | 1. The Ticket can be created along with the project mentor 2. The system send a notification to the assigned project mentor |
| Exceptions | 1. No project mentor available to take care the ticket |

|  |  |
| --- | --- |
| Use Case ID | MP-Retrieve the tickets created by their mentees. |
| Description | Project Mentor will be able to see the ticket created by their mentees and are assign to other domain mentor |
| Actor | Project Mentor |
| Pre-conditions | * The Project Mentor has been assigned a project * The Project has been logged in to the system |
| Steps | 1. The use case begins when the project mentor clicks on the Project Mentor Button. 2. The system creates a new Project Meeting view with the list of tickets created by his/her mentees 3. The project mentor clicks on the ticket to see the details. 4. The use case finish when the project mentor close the view |
| Post-conditions | 1. The information is displayed allowing the project mentor to see the details |
| Exceptions | 1. The Project mentor was not assigned to any project. |

|  |  |
| --- | --- |
| Use Case ID | MP-Assign tickets to his/her project mentor. |
| Description | The mentee is able to assign a ticket directly to his/her project mentor |
| Actor | Mentee |
| Pre-conditions | * Mentee has logged into the system * Mentee has click in the New Ticket button. |
| Steps | 1. The use case begins when the mentee click on the submit button 2. The System validates the inputs 3. The system verified if the project mentor was selected 4. The System search for the project mentor information 5. The use case finish when the system returns the project mentor |
| Post-conditions | 1. The Ticket can be created along with the project mentor 2. The system send a notification to the assigned project mentor |
| Exceptions | 1. No project mentor available to take care the ticket |

|  |  |
| --- | --- |
| Use Case ID | MP-Create a ticket. |
| Description | The User is able to create a ticket in the collaborative platform |
| Actor | User |
| Pre-conditions | * User has logged into the system * User has click in the New Ticket button. |
| Steps | 1. The use case begins when the user click on the submit button 2. The System validates the inputs 3. The system verified if the project mentor was selected 4. The System search for the project mentor information. 5. The system save the current date. 6. The use case finish if the ticket was created successfully in the system. |
| Post-conditions | 1. A new ticket is created in the collaborative platform. |
| Exceptions | 1. Failed to access the DB can be occur while the ticket is saving. |

|  |  |
| --- | --- |
| Use Case ID | MP-Select a specific domain for a new ticket. |
| Description | The User is able to select a domain for a new ticket in the collaborative platform |
| Actor | User |
| Pre-conditions | * User has logged into the system * User has click in the New Ticket button. |
| Steps | 1. The use case begins when the user click on the dropdown list Domain. 2. The system provides a list of domains available. 3. The use case finish when the user selects the desire domain |
| Post-conditions | 1. A domain is specified for the new ticket |
| Exceptions | 1. No Domain available. |

|  |  |
| --- | --- |
| Use Case ID | MP-Select a specific sub domain. |
| Description | The User is able to select a sub domain for a new ticket in the collaborative platform |
| Actor | User |
| Pre-conditions | * User has logged into the system * User has click in the New Ticket button. * User has been specified a Domain. |
| Steps | 1. The use case begins when the user click on the dropdown list sub Domain. 2. The system provides a list of sub domains available according to the domain selected. 3. The use case finish when the user selects the desire sub domain |
| Post-conditions | 1. A sub domain is specified for the new ticket |
| Exceptions | 1. No sub domain available. |

|  |  |
| --- | --- |
| Use Case ID | MP-Upload file to a ticket. |
| Description | The User can upload to a ticket, in order to clarify his question. |
| Actor | User |
| Pre-conditions | * User has logged into the system * User has click in the New Ticket button. |
| Steps | 1. The use case begins when the user clicks the Choose File button. 2. The system provides the system root. 3. The user selects the desire file to upload. 4. The use case finish when the user clicks on the button Open |
| Post-conditions | 1. The upload file is ready to be saving along with the ticket. |
| Exceptions | 1. An error can occur while the system is uploading the file. |

|  |  |
| --- | --- |
| Use Case ID | MP-Download file from a ticket. |
| Description | The User can download a file from the ticket. |
| Actor | User |
| Pre-conditions | * User has logged into the system * User has click in the New Ticket button. * A file has been uploaded to a ticket. |
| Steps | 1. The use case begins when the user clicks the Click here to download the file link. 2. The server start downloading the file into the client computer. 3. The use case finish when the file ends of downloading. |
| Post-conditions | 1. The file is on the client computer ready to be open. |
| Exceptions | 1. An error can occur while the system is downloading the file. |

|  |  |
| --- | --- |
| Use Case ID | MP-Append comments to a ticket. |
| Description | The user can append comment to a ticket while the ticket is pending. |
| Actor | User |
| Pre-conditions | * User has logged into the system * A ticket is pending. * The user has click on the ticket to view the details. |
| Step | The use case begins when the user clicks on the add comment button on the detail view  The system provides a form with the comment field.  The user type the comment.  The use case finish when the user clicks on the append comment. |
| Post-conditions | 1. A comment is added to a ticket. 2. The system sends notification that a comment has been added to a ticket. |
| Exceptions | 1. An error can occur while the system is saving the comment. |

|  |  |
| --- | --- |
| Use Case ID | MP-Retrieve the ticket details |
| Description | The user can view all the details of the ticket. |
| Actor | User |
| Pre-conditions | * User has logged into the system |
| Steps | 1. The use case begins when the user clicks on a specific ticket 2. The system provides a detailed view of the ticket. 3. The use case finish when the user close the view |
| Post-conditions | 1. User is able to see all the ticket details. |
| Exceptions | 1. An error can occur while the system is loading the details of the ticket. |

|  |  |
| --- | --- |
| Use Case ID | MP-Close a Ticket |
| Description | The user can close a ticket when he/she feels that the comments provides byt the mentor have answer the question. |
| Actor | User |
| Pre-conditions | * User has logged into the system. * The User has to be the creator of the ticket or administrator. * The user has click on the ticket to view the details |
| Steps | 1. The use case begins when the user clicks on the change status button on the detail view ticket. 2. The system provides a dropdown list with the status, also a text field to leave a comment. 3. The use case finish when the user clicks on the Ok button |
| Post-conditions | 1. The ticket is close 2. The ticket is set on read only. 3. A notification is sent to the assigned mentor lets him know that the ticket was closed. |
| Exceptions | 1. An error can occur while the system is changing the status |

|  |  |
| --- | --- |
| Use Case ID | MP-Reject a Ticket |
| Description | The user can reject a ticket, that means the user ticket creator is not satisfy with the comments received. So He/She change the status to reject in order to notify the admin. |
| Actor | User |
| Pre-conditions | * User has logged into the system. * The User has to be the creator of the ticket or administrator. * The user has click on the ticket to view the details |
| Steps | 1. The use case begins when the user clicks on the change status button on the detail view ticket. 2. The system provides a dropdown list with the status, also a text field to leave a comment. 3. The use case finish when the user clicks on the Ok button |
| Post-conditions | 1. The ticket is reject 2. A notification is send to the administrator. 3. The ticket can be reassign only by the administrator |
| Exceptions | 1. An error can occur while the system is changing the status |

|  |  |
| --- | --- |
| Use Case ID | MP-system automatically Reassign ticket |
| Description | The System is able to reassign a ticket |
| Actor | System |
| Pre-conditions | * The Ticket status of the ticket is pending. * The mentee was waiting for an answer from his/her assigned mentor for a determined period of time. |
| Steps | 1. The server run the automated job from cron witch runs the function that automatically reassign the ticket. |
| Post-conditions | 1. The ticket is reassigned to a new domain mentor in the same tier than the previous one. 2. A notification is send to the creator and the new domain mentor, as well to the old domain mentor. 3. A new comment created by system is added to the ticket to keep the history. 4. If the ticket is automatically reassigned by the system 3 times the system send a notification to the administrator. |
| Exceptions | No any available mentor |

|  |  |
| --- | --- |
| Use Case ID | MP- Send Ticket closed Notification |
| Description | Send email to assigned mentor to whom the ticket was assigned |
| Actor | System |
| Pre-conditions | * The User close a ticket |
| Steps | 1. System generates email 2. System send email to the Mentor to whom the ticket was assigned |
| Post-condition | The assigned mentor will receive a notification email with the ticket information(id and subject) |
| Exceptions | * The server mail service is down. |

|  |  |
| --- | --- |
| Use Case ID | MP-Retrieve to personal mentors the tickets created by their mentees. |
| Description | Personal Mentor will be able to see the ticket created by their mentees and are assign to other domain mentor |
| Actor | Personal Mentor |
| Pre-conditions | * The Personal mentor has been logged in to the system |
| Steps | 1. The use case begins when the personal mentor clicks on the Personal Mentor Button. 2. The system creates a new view with the list of tickets created by his/her mentees 3. The personal mentor clicks on the ticket to see the details. 4. The use case finish when the personal mentor close the view |
| Post-conditions | 1. The information is displayed allowing the personal mentor to see the details |
| Exceptions | - |

|  |  |
| --- | --- |
| Use Case ID | MP-Assign tickets to his/her personal mentor. |
| Description | The mentee is able to assign a ticket directly to his/her personal mentor |
| Actor | Mentee |
| Pre-conditions | * Mentee has logged into the system * Mentee has click in the New Ticket button. |
| Steps | 1. The use case begins when the mentee click on the submit button 2. The System validates the inputs 3. The system verified if the personal mentor was selected 4. The System search for the personalmentor information 5. The use case finish when the system returns the personal mentor |
| Post-conditions | 1. The Ticket can be created along with the personal mentor 2. The system send a notification to the assigned personal mentor |
| Exceptions | - |

|  |  |
| --- | --- |
| Use Case ID | MP – Edit Priority |
| Description | Allow Admin to update priorities |
| Actor | Admin |
| Pre-conditions | * Admin is in his home page (Dashboard) |
| Steps | 1. Use case begins when the admin clicks on Priorities button. 2. Admin sets new time interval for ticket reassignment. 3. Use case ends when the admin clicks on save. |
| Post-conditions | The new time interval will be updated in the database. |

|  |  |
| --- | --- |
| Use Case ID | MP- Escalate Ticket |
| Description | The domain mentor in tier 1 is able to escalate a ticket. |
| Actor | Domain mentor tier 1 |
| Pre-conditions | * Domain mentor has logged into the system * A ticket is pending. * The domain mentor has click on the ticket to view the details. |
| Steps | 1. The use case begins when the mentor clicks on the escalation button 2. The system creates a new ticket with all the comments that already exist in the original ticket and assigns the new ticket to mentor in tier 2. 3. A comment created by the system is added to the original ticket and to the new one linking both tickets**.** |
| Post-conditions | 1. A new Ticket is created and assigned to a domain mentor tier 2 2. The system send a notification to the assigned domain mentor tier 2 |
| Exceptions | -If you are not a domain mentor in tier 1 |

|  |  |
| --- | --- |
| Use Case ID | MP-Import data from SPW automatically |
| Description | At midnight data from SPW should be imported to CP automatically |
| Actor | System |
| Pre-conditions | * N/A |
| Steps | 1. The use case begins when the cron job is triggered at midnight 2. The system request the data through the SPW API 3. The SPW API retrieves the data |
| Post-conditions | 4. The database is populated |
| Exceptions | 1. The SPW API could be down causing an exception |

|  |  |
| --- | --- |
| Use Case ID | MP-Allow Mentees to login suing their FIU Google credentials |
| Description | Students imported from SPW (mentees) should be able to login using their FIU Google credentials |
| Actor | Mentee |
| Pre-conditions | * The mentee should be in the login page |
| Steps | 1. The use case begins when the mentee clicks on the “FIU Login” button located in the login page 2. The system takes the mentee to the Google authentication page 3. The mentee enters his/her FIU email credentials 4. Google verifies the credentials are correct |
| Post-conditions | 5. The mentee is taken to his home page (dashboard) |
| Exceptions | 1. The Google API could be down causing an exception |

|  |  |
| --- | --- |
| Use Case ID | MP-Import data from SPW manually |
| Description | The admin should be able to import data from SPW manually |
| Actor | Admin |
| Pre-conditions | * The admin is in his home page(dashboard) |
| Steps | 1. The use case begins when the admin clicks the “Sync with SPW” button located in his dashbaord 2. The system request the data through the SPW API 3. The SPW API retrieves the data |
| Post-conditions | 4. The database is populated |
| Exceptions | 1. The SPW API could be down causing an exception |

|  |  |
| --- | --- |
| Use Case ID | MP-Allow the admin to register mentors (project/domain/personal) |
| Description | The admin should be able to register mentors |
| Actor | Admin |
| Pre-conditions | * The admin is in his home page (dashboard) |
| screenSteps | 1. The use case begins when the admin goes to “Manage users ” and click on “Add user” 2. The admin inputs the basic information for the new user and clicks next. 3. The admin is taken to a wizard where he sets the roles 4. The admin clicks submit |
| Post-conditions | 5. The new user is added to users list  6. The system sends an email to the new user with his/her credentials |
| Exceptions | 7. The user might already exists, but that’s taken care of in the first screen |

**Use Cases for Collaborative Platform Version 3**

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-001 Self Registration for CP Account** |
| **Update for** | **CP-005 Register User** |
| **Description** | A User registers with the Collaborative Platform to create an account. |
| **Actor** | User |
| **Pre-conditions** | * User has received an email invitation from the Collaborative Platform. |
| **Steps** | 1. The use case begins when the User clicks the link provided and is redirected to the new\_user\_landing\_page 2. The User then clicks the Sign-Up button and is redirected to the registration\_page 3. The User then fills in desired account information. 4. The User then fills in desired personal information 5. The User then clicks the next button and the registration\_verification\_modal is loaded. 6. The User then verifies that the data entered was correct 7. The use case ends when the User then clicks the Submit button and is redirected to the mentor\_application\_portal\_page |
| **Post-conditions** | * A new User is added to the database * The new User is logged in to the system |
| **Alternative Actions** | 1. At step 2 the User could click the cancel button and be redirected back to the new\_user\_landing\_page 2. At step 6 the user could click the back button and close the modal to view the registration\_page to make changes to his information |
| **Exceptions** | * The User does not fill in a required field * The User attempts to register with an email address that is associated with an existing account * The User attempts to register with an username that is associated with an existing account |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-002 Apply for Personal Mentorship** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | A Mentor applies for a Personal Mentorship. |
| **Actor** | Mentor |
| **Pre-conditions** | * Mentor is logged into the Collaborative Platform * Mentor is at the mentor\_application\_portal\_page |
| **Steps** | 1. The use case begins when the Mentor clicks the Personal Mentor portrait and is redirected to the personal\_mentor\_application\_page 2. The Mentor then selects desired Mentees from the list provided 3. The Mentor then selects desired System Pick criteria 4. The Mentor then clicks the next button and is the personal\_mentor\_application\_verification\_modal is loaded 5. The Mentor then verifies that the data entered is correct 6. The use case ends when the Mentor clicks the submit button and is redirected back to the mentor\_application\_portal\_page |
| **Post-conditions** | * A new Application is added to the database * The user and system picks are added to the database |
| **Alternative Actions** | 1. At step 1 the Mentor may click the apply button located beneath the Personal Mentor description 2. At step 2 the Mentor may hover over a Mentee from the list to see more information about them 3. At step 2 the Mentor may click a column header to filter by Name or University 4. At step 4 the Mentor could click the cancel button and be redirected back to the mentor\_application\_portal\_page 5. At step 6 the Mentor could click the back button and be redirected back to the personal\_mentor\_application\_page |
| **Exceptions** | * The Mentor submits an empty application * The Mentor has selected too many Mentees * The Mentor selects no Mentees * The Mentor selects no system picks * The Mentor enters an invalid number of system picks |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-003 Search for User(s)** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | Admin wants to search for specific user(s) based on some predefined criteria. |
| **Actor** | Admin |
| **Pre-conditions** | * User is logged in as admin * The Admin is on the manage\_users\_page * The page defaults to selections on   + Filter Type: Project Mentor, Personal Mentor, Domain Mentor, Student   + Filter Status: Active |
| **Steps** | 1. The use case begins when the Admin selects from the different Filter Types as query requires (ProjMentor, PersMentor, DomMentor, Student). 2. The Admin then selects to filter Status (active and/or disabled) 3. The use case ends when the filter results are displayed on the page. |
| **Post-conditions** | * The filtered results have been rendered on the table. |
| **Alternative Actions** | * None |
| **Exceptions** | * None |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-004 Apply for Project Mentorship** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | A Mentor applies for a Project Mentorship. |
| **Actor** | Mentor |
| **Pre-conditions** | * Mentor is logged into the Collaborative Platform * Mentor is at the mentor\_application\_portal\_page |
| **Steps** | 1. The use case begins when the Mentor clicks the Project Mentor portrait and is redirected to the project\_mentor\_application\_page 2. The Mentor then selects desired Project from the list provided 3. The Mentor then selects desired System Pick criteria 4. The Mentor then clicks the next button and the project\_mentor\_application\_verification\_modal is loaded 5. The Mentor then verifies that the data entered is correct 6. The use case ends when the Mentor clicks the submit button and is redirected back to the mentor\_application\_portal\_page |
| **Post-conditions** | * A new Application is added to the database * The user and system picks are added to the database |
| **Alternative Actions** | 1. At step 1 the Mentor may click the apply button located beneath the Project Mentor description 2. At step 2 the Mentor may hover over a Project from the list to see more information 3. At step 4 the Mentor could click the cancel button and be redirected back to the mentor\_application\_portal\_page 4. At step 6 the Mentor could click the back button and be redirected back to the project\_mentor\_application\_page |
| **Exceptions** | * The Mentor submits an empty application * The Mentor has selected too many Projects * The Mentor selects no Projects * The Mentor selects no system picks * The Mentor enters an invalid number of system picks |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-005 Allow admin to view pending application** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | Admin wants to see what applications are pending admin’s approval/review. |
| **Actor** | Admin |
| **Pre-conditions** | * Actor is logged in as admin * There are active applications requiring admins approval |
| **Steps** | 1. The use case beings when the admin clicks on the Manage\_Applications link from the dropdown. 2. The system then queries the database to check for outstanding applications and displays the result 3. The admin now can click on any application to take action. 4. The use case ends when the admin clicks on an application |
| **Post-conditions** | * The results have been rendered on the table. |
| **Alternative Actions** | 1. none |
| **Exceptions** | * The Admin does not have any pending applications. |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-006 Apply for Domain Mentorship** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | A Mentor applies for a Domain Mentorship. |
| **Actor** | Mentor |
| **Pre-conditions** | * Mentor is logged into the Collaborative Platform * Mentor is at the mentor\_application\_portal\_page |
| **Steps** | 1. The use case begins when the Mentor clicks the Domain Mentor portrait and is redirected to the Domain\_mentor\_application\_page 2. The Mentor then enters his proficiency for desired Domains 3. The Mentor then clicks the recommend new domain button 4. The Mentor then fills in the information for the recommended domain 5. The Mentor then enters how many tickets he desires per month 6. The Mentor then clicks the next button and the domain\_mentor\_application\_verification\_modal is loaded 7. The Mentor then verifies that the data entered is correct 8. The use case ends when the Mentor clicks the submit button and is redirected back to the mentor\_application\_portal\_page |
| **Post-conditions** | * A new Application is added to the database * Domain and subdomain picks are added to the database |
| **Alternative Actions** | 1. At step 1 the Mentor may click the apply button located beneath the Domain Mentor description 2. At step 2 the Mentor may hover over a Domain from the list to activate the domain\_info\_modal to see more information 3. At step 2 the Mentor may expand or collapse domains from the list to view subdomains 4. At step 5 the Mentor could recommend another domain 5. At step 6 or 8 the Mentor could click the cancel button and be redirected back to the mentor\_application\_portal\_page 6. At step 8 the Mentor could click the back button and be redirected back to the domain\_mentor\_application\_page |
| **Exceptions** | * The Mentor submits an empty application * The Mentor has selected too many Domains * The Mentor enters invalid data * The Mentor tries to recommend a domain without filling in the required fields for the previous recommended domain |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-007 Allow admin to gain insight on a particular user** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin wants to gain insight on how a particular user is utilizing the system. |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_users\_page |
| **Steps** | 1. The use case begins when the admin clicks on the ‘details’ button depicted with an eyeball icon. 2. The system will open a modal with the following information:    1. Project Mentor User: what projects are being mentored    2. Personal Mentor User: what menttes are being mentored    3. Domain Mentor: What domains/subdomains are being mentored. Changes are allowed directly to the data.    4. Mentee: What mentors they have and what project they belong to.    5. Tickets belonging to user (created or assigned to)    6. Meetings relating to user 3. The use case ends when the admin clicks on the exit icon on the modal. |
| **Post-conditions** | * The modal is closed and the view that the admin had before the modal is restored. |
| **Alternative Actions** | 1. Step 1: The admin right clicks the ‘eyeball’ and selects open in new tab. This will skip the modal and render the page in a new tab. |
| **Exceptions** | * The user has no roles. * The user has no tickets * the user is not a project mentor * the user is not a personal mentor * the user is not a domain mentor * the user is not a mentee * the user has no meetings |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-008 Add a Domain/Sub-Domain** |
| **Update for** | **Add a domain** |
| **Description** | The admin wants to add a domain/sub-domain to the system. |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_domain\_page |
| **Steps** | 1. The use case begins when the admin clicks on “Add New Domain” 2. The system will then render a new page for input form the admin. 3. The admin enters the appropriate data (Domain name, description, need, amount, proficiency cutoff). 4. The admin then clicks on submit 5. The use case ends when the domain has been added to the database. |
| **Post-conditions** | * The database has been updated with new domain. |
| **Alternative Actions** | 1. Step 1 - The admin clicks on “Add new subdomain” 2. Step 4 - The admin failed to fill in one of the required fields at which time validation will prevent submission. |
| **Exceptions** | * None |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-009 Allow admin to gain insight on a particular domain** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin wants to see the status of a particular domain and gain insight. |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_domain\_page |
| **Steps** | 1. The use case begins when the admin clicks on the eyeball for a particular domain 2. A pop-up Modal will appear with the Details of the domain. (Name, Members Needed, Actual, Need, Description, Proficiency Cutoff, a list of tickets that have the domain listed, a list of all mentors currently mentoring the domain) 3. The use case ends when the admin clicks on “Close”. |
| **Post-conditions** | * The modal closes and the domain\_page view is restored. |
| **Alternative Actions** | 1. Step 2 - The admin can right click the eyeball and selects open in new tab. The system will generate a new page with same information instead of a modal. |
| **Exceptions** | * The domain has not tickets relating to it * The domain has no mentors for it. |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-010 Allow admin to gain insight on a particular subdomain** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin wants to see the status of a particular subdomain and gain insight. |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_domain\_page |
| **Steps** | 1. The use case begins when the admin clicks on the eyeball for a particular subdomain 2. A pop-up Modal will appear with the Details of the domain. (Name, Members Needed, Actual, Need, Description, Proficiency Cutoff, a list of tickets that have the subdomain listed, a list of all mentors currently mentoring the subdomain) 3. The use case ends when the admin clicks on “Close”. |
| **Post-conditions** | * The modal closes and the domain\_page view is restored. |
| **Alternative Actions** | 1. Step 2 - The admin can right click the eyeball and selects open in new tab. The system will generate a new page with same information instead of a modal. |
| **Exceptions** | * The subdomain has not tickets relating to it * The subdomain has no mentors for it. * The parent domain has been deleted, null would appear for domain. |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-011 Allow admin to view tickets** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin would like to access and view all tickets in the system |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on homepage |
| **Steps** | 1. The use case begins when the mentor clicks on the dropdown menu and selects “Manage” -> “Tickets” 2. The system will then render a page with a sortable/searchable list of all tickets in the system. 3. THe use case ends after the page is rendered completely. |
| **Post-conditions** | * None |
| **Alternative Actions** | 1. None |
| **Exceptions** | * There are no tickets in the system. |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-012 Allow admin to manage tickets** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin wants to delete, view or update a ticket that is currently in the system. |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_tickets\_page |
| **Steps** | 1. The use case begins when the admin identifies that ticket in question 2. The admin selects the trash icon to delete the entry. 3. The use case ends when the ticket is deleted from the system |
| **Post-conditions** | * The ticket has been removed from the database and the table has been refreshed. |
| **Alternative Actions** | 1. At step 2 the admin can select to view the ticket, which would open a modal with details regarding ticket.    1. The admin selects the eyeball icon    2. The modal is rendered 2. At step 2, the admin can select to update the ticket    1. The admin selects the pencil (update) icon.    2. The update page is rendered. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-013 Allow admin to use an advanced search for tickets** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin is curious and would like to search through tickets based on some parameters. |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_tickets\_page |
| **Steps** | 1. The use case begins when the admin clicks on the “advanced search” link 2. A new section will appear allowing the admin to input specific parameters relating to tickets (e.g. date create, date closed, assigned to, created by - all elements in the ticket can be filtered and searched) 3. The admin then clicks on submit button 4. The use case ends when the results are displayed in the gridview. |
| **Post-conditions** | * The results are displayed in the gridview |
| **Alternative Actions** | None |
| **Exceptions** | * There are no tickets. |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-013 Allow admin to use an advanced search for tickets** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin is curious and would like to search through tickets based on some parameters. |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_tickets\_page |
| **Steps** | 1. The use case begins when the admin clicks on the “advanced search” link 2. A new section will appear allowing the admin to input specific parameters relating to tickets (e.g. date create, date closed, assigned to, created by - all elements in the ticket can be filtered and searched) 3. The admin then clicks on submit button 4. The use case ends when the results are displayed in the gridview. |
| **Post-conditions** | * The results are displayed in the gridview |
| **Alternative Actions** | None |
| **Exceptions** | * There are no tickets. |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-014 Allow admin to approve/reject items in an application** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin is reviewing an application (personal, project, or domain) |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_applications\_page |
| **Steps** | 1. The use case begins when the admin clicks on one of the availilbe application. 2. The admin then is presented with all available applications relating to the user. 3. The admin now has the ability to accept/reject all items that have a status of “proposed by system” and “proposed by mentor” 4. There is a counter to keep track that all items have been selected. Once all items have an appropriate choice the submit button is eligible for click. 5. The use case ends when the admin has selected the submit button at the bottom of the page. |
| **Post-conditions** | * All items that have been rejected will get updated to status “Rejected” * All items that have been accepted will get updated to status “ Accepted” and will also create an entry into the appropriate table to make the selection live. * The parent of the pick application has been update to “Mentor” or will change to “Closed” if there are no pending items. |
| **Alternative Actions** | In step 3, the admin has the ability to click cancel and return to the view of applications. |
| **Exceptions** | * There are no picks availible for selection yet the parent application has a status of “Admin” |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-015 Allow admin to propose items in an application** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin is reviewing an application (personal, project, or domain) |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_applications\_page |
| **Steps** | 1. The use case begins when the admin wants to propose new items in each application. 2. The admin selects a mentee from the dropdown list in the personal mentor application section 3. The admin clicks submit and the entry gets submitted. 4. The admin then selects a project in the project mentor application section 5. The admin clicks submit and the entry gets submitted 6. The admin then selects a domain the domain application mentor section and enters a proficiency. 7. The admin clicks submit and the entry gets submitted 8. The use case ends when the admin has no more proposals. |
| **Post-conditions** | * The personal mentor proposal creates an entry with selected mentee and status of “Proposed by Admin” * The project mentor proposal creates an entry with selected project and status of “Proposed by Admin” * The domain mentor proposal creates an entry with selected domain and status of “Proposed by Admin” |
| **Alternative Actions** | * In step 4 the admin can end the use case by not submitting anything else. * In Step 6 the admin can end the use case by not submitting anything else. |
| **Exceptions** | * None |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-016 Respond to Mentorship Offer** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | A mentor is offered a mentorship by an admin |
| **Actor** | Mentor |
| **Pre-conditions** | * The user is logged on as mentor * The admin proposed an offer to the mentor * The user is on the mentor\_home\_page |
| **Steps** | 1. The use case begins when the mentor clicks on an offer and is redirected to the mentorship\_offer\_page 2. The mentor clicks accept for each offer 3. The mentor clicks submit 4. The use case ends when the mentor is redirected to the mentor\_home\_page |
| **Post-conditions** | * The offered mentorships are added to the database * Accepted offers go immediatly go live |
| **Alternative Actions** | 1. At step 2 the mentor could decide to reject an offer |
| **Exceptions** | * Mentor attempts to submit without either accepting or rejecting each offer |

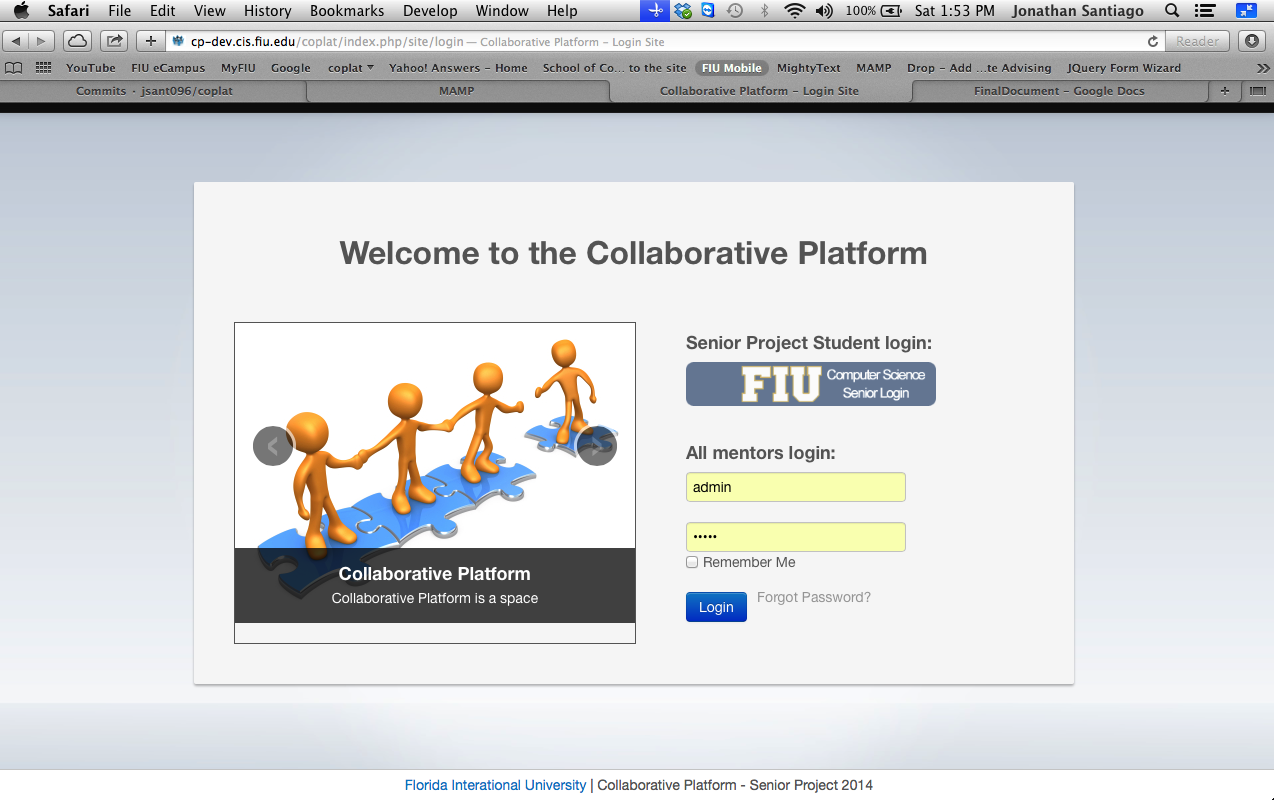
|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-017 Allow admin to view open invitations/re-invites** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin would like to see how many re-invitations have been sent. |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_invitations\_page |
| **Steps** | 1. The use case begins when the admin selects an invitation. 2. The system will render a modal with re-invitation details. 3. The use case ends when the admin closes the modal |
| **Post-conditions** | * None |
| **Alternative Actions** | * None |
| **Exceptions** | * None |

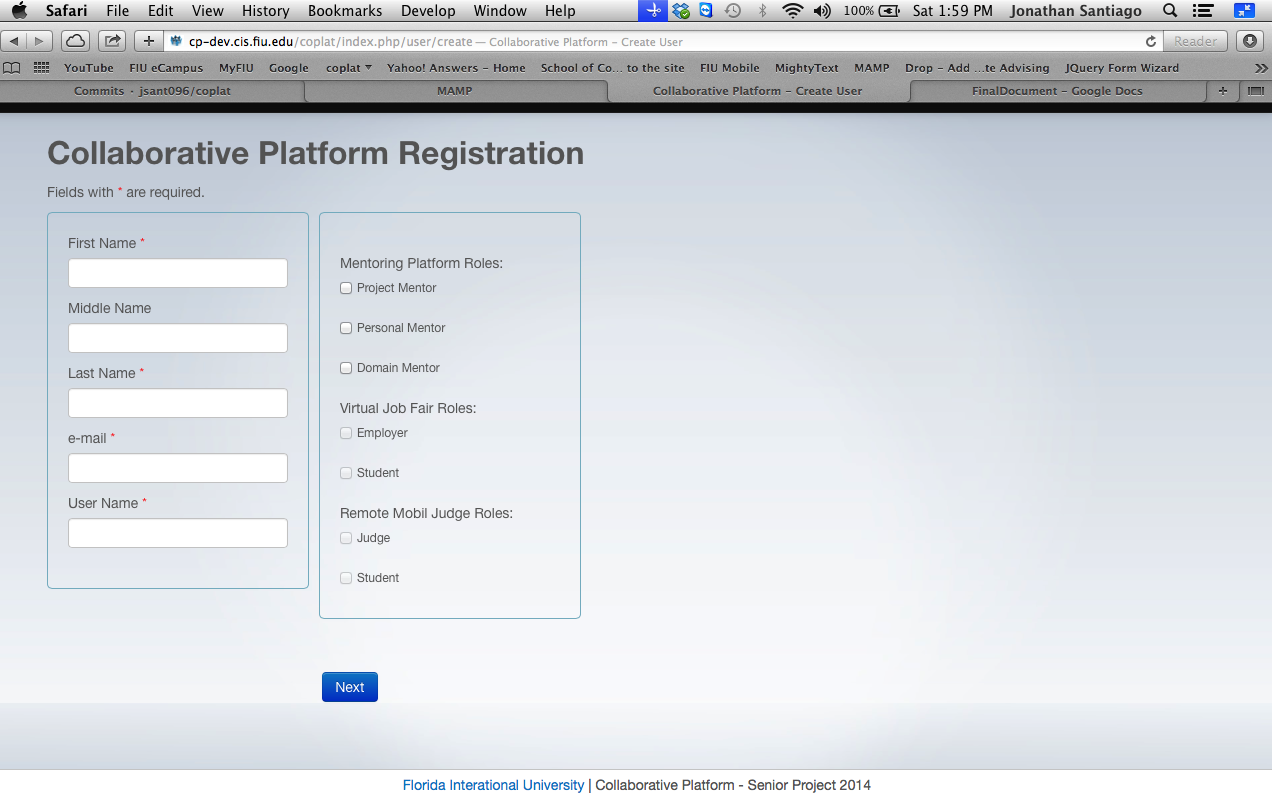
|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-018 Allow admin to send a custom invite** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin woud liek to send an invite but would like to edit the default template before email is sent out to potential mentor |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_invitations\_page |
| **Steps** | 1. The use case begins when the admin create new invitation 2. The system is redirected to a page to allow the admin to input name, email, and what roles to email the mentor about. 3. The admin clicks next, which will take the admin to a page where they can review the template 4. The admin can review and make changes. 5. The use case ends when the admin clicks on submit. |
| **Post-conditions** | * On step 5, on submit, the system will send out the email with the requested modifications. * An entry is made into database for the invite. |
| **Alternative Actions** | * Step 3, the admin can click cancel instead of next. |
| **Exceptions** | * None |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv3-019 Allow admin to gain insight on a particular project** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The admin would like to gain insight on a particular project |
| **Actor** | Admin |
| **Pre-conditions** | * The user is logged on as admin * The admin is on the manage\_projects\_page |
| **Steps** | 1. The use case begins when the admin selects a particular project 2. the system will render a modal with details regarding the project (e.g who is on the project, who is mentoring the project, and what tickets relate, as well as any meetings that have been scheduled with a project mentor) |
| **Post-conditions** | * None |
| **Alternative Actions** | * None |
| **Exceptions** | * None |

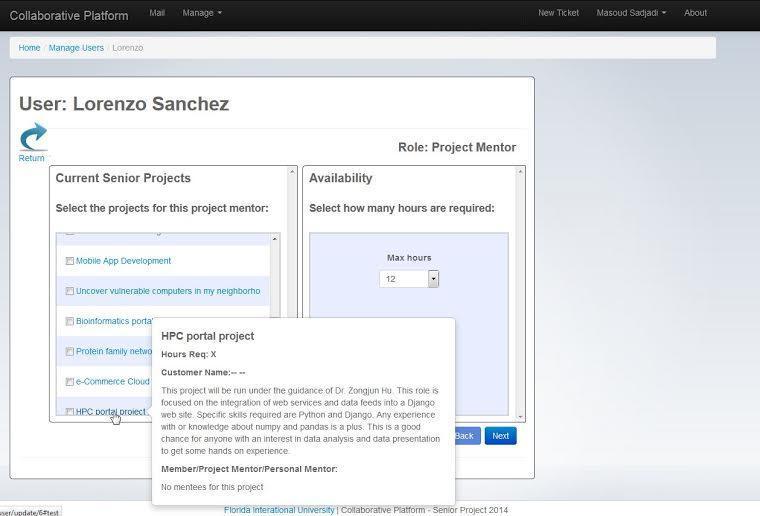
## 9.3 Appendix C – User Interface Design

**Access and Registration**

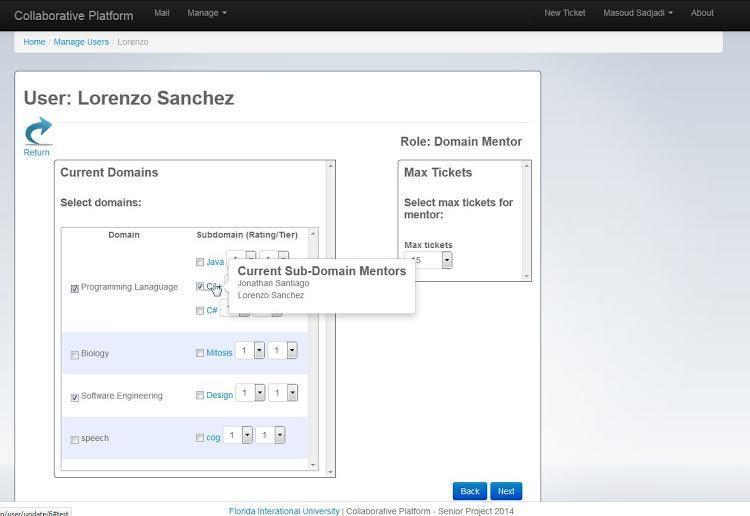
**Figure 1 Login Page**



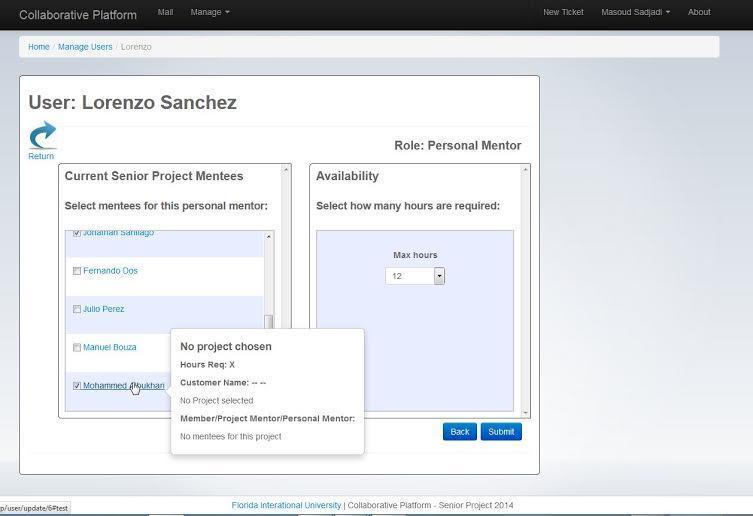
**Figure 2 Registration Page**

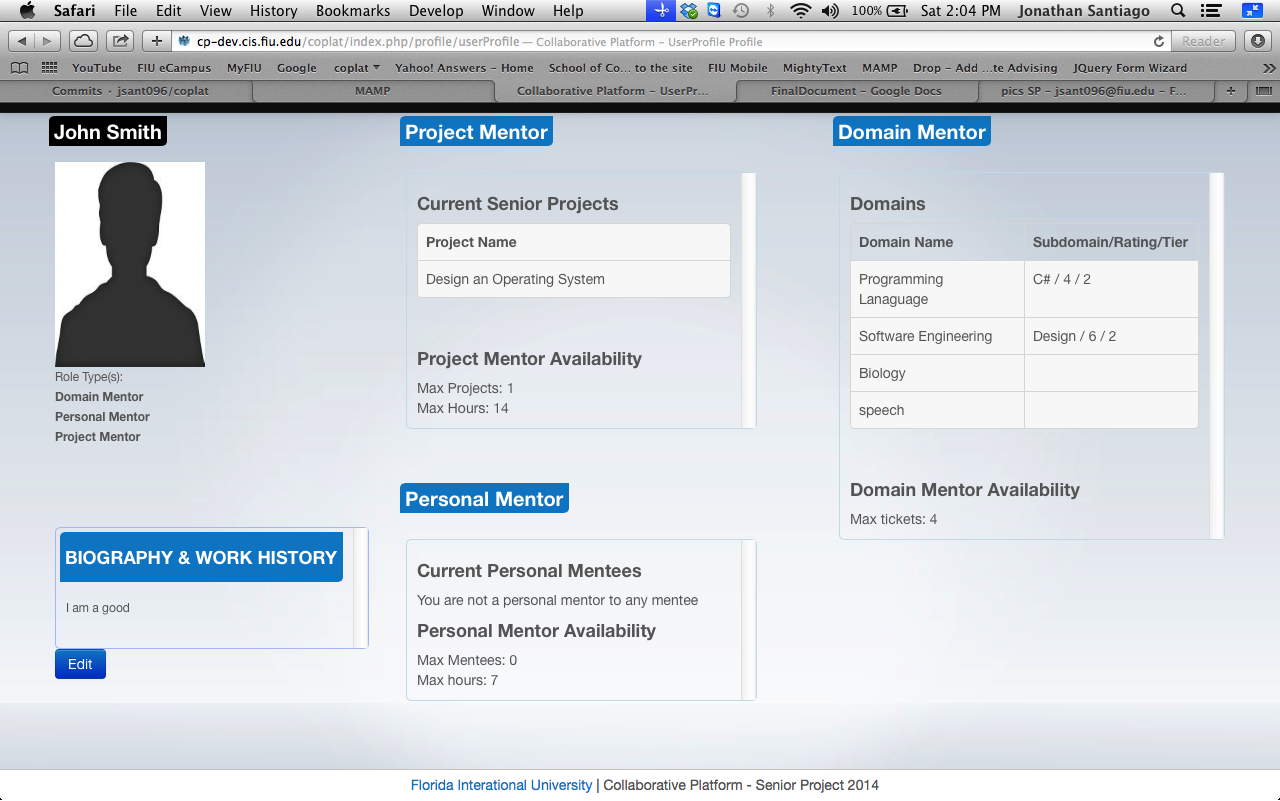


**Figure 2.1 Registration Page**



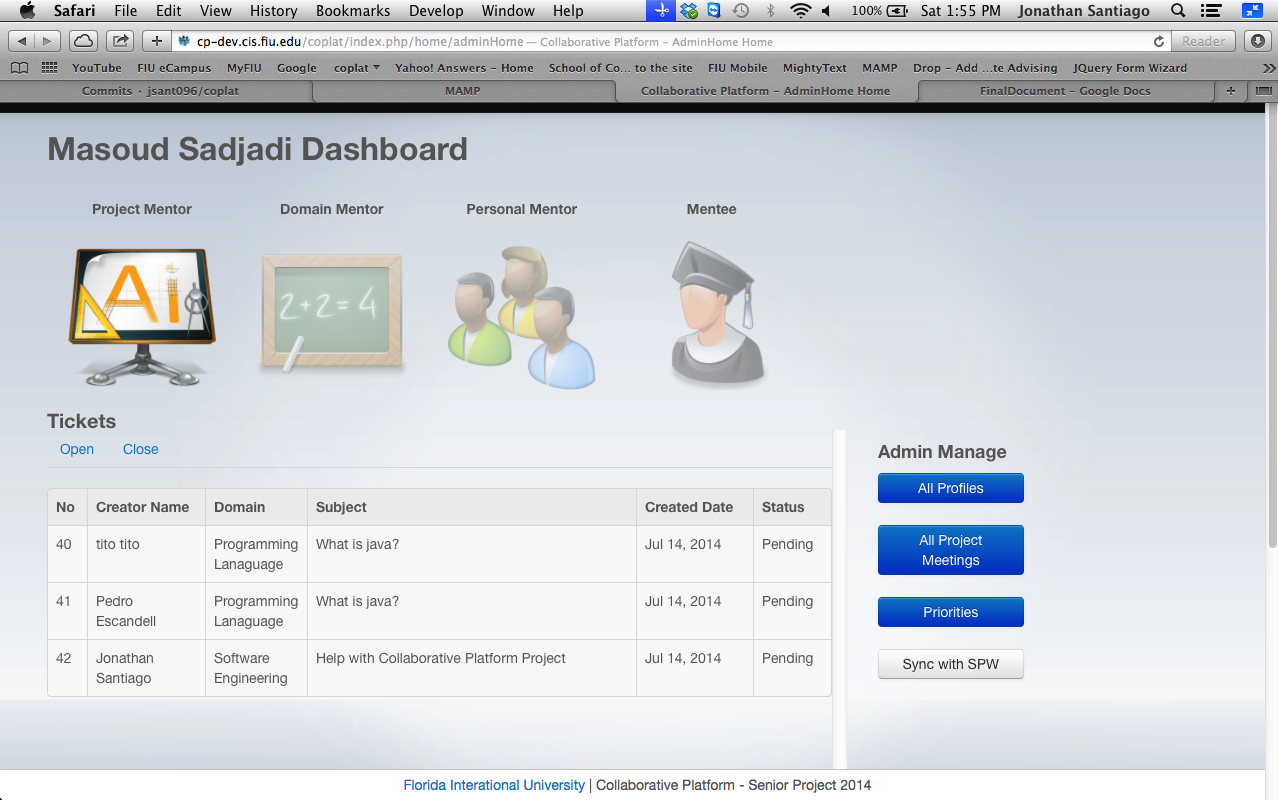
**Figure 2.2 Registration Page**

**Figure 2.3 Registration**

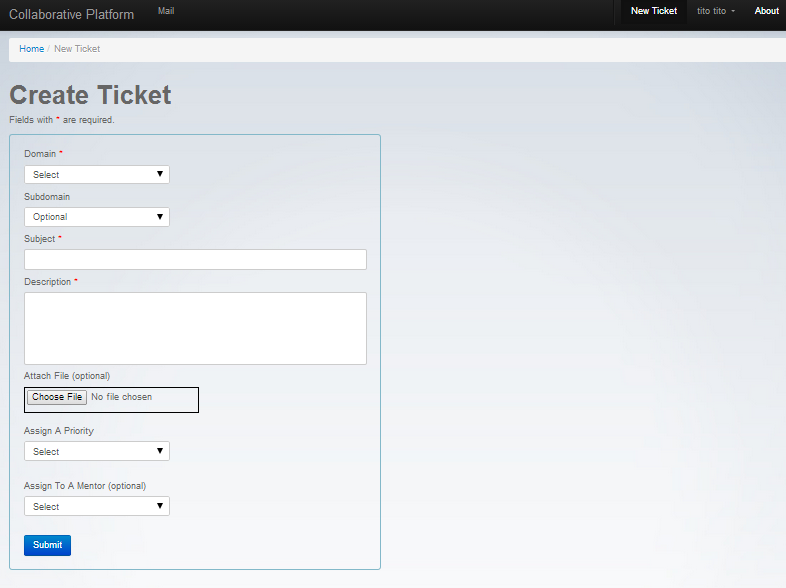


**Figure 3 Profile View**

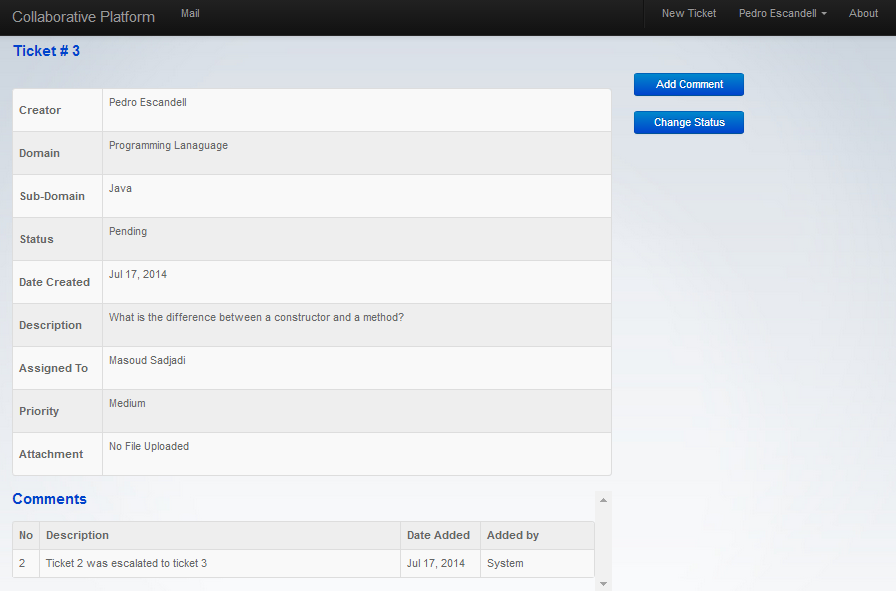
**Mentoring**



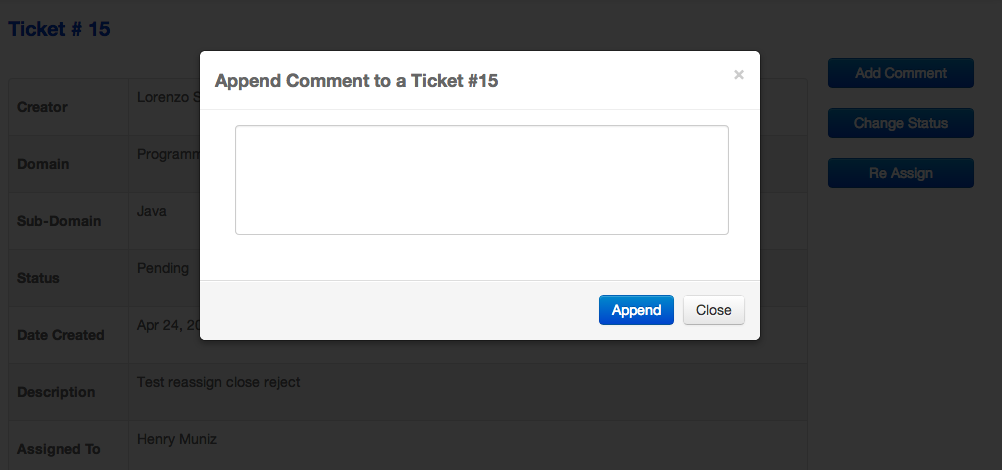
**Figure 4 userHome View**



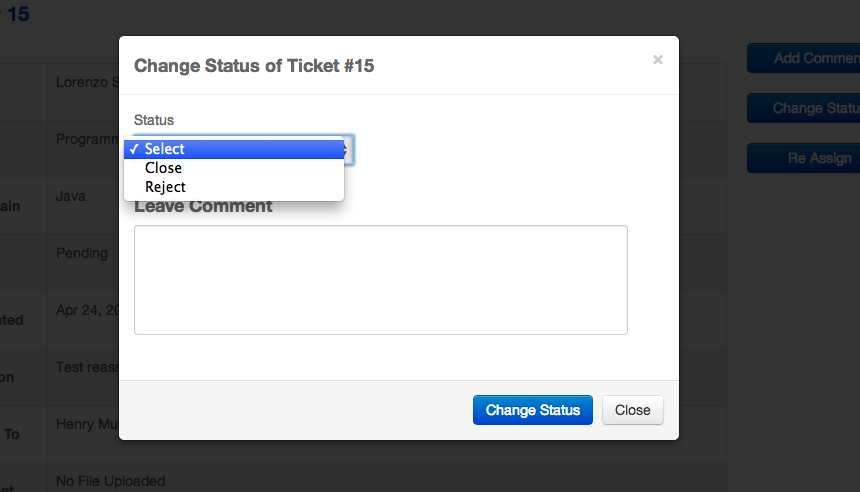
**Figure 5 New Ticket Page**



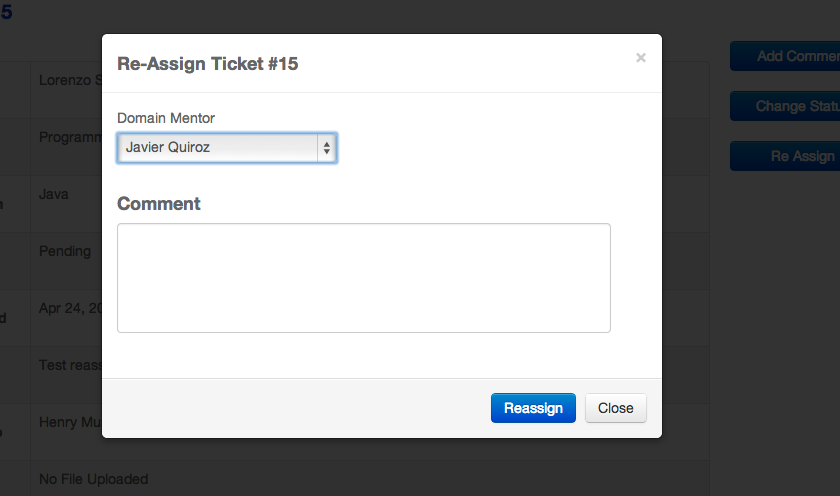
**Figure 6 Ticket Detail Page**



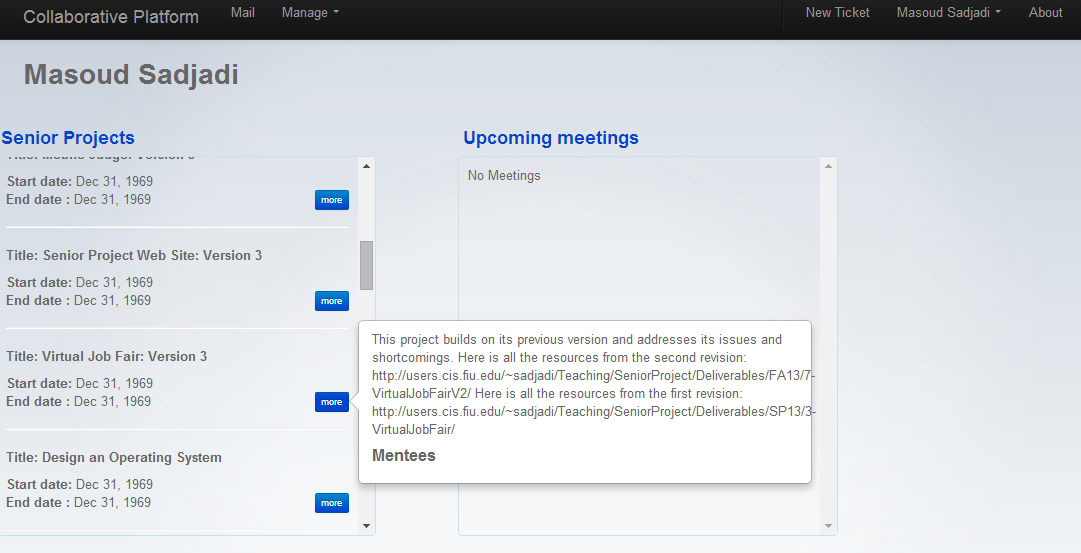
**Figure 7 Append Comment Page**



**Figure 8 Change Status Page**

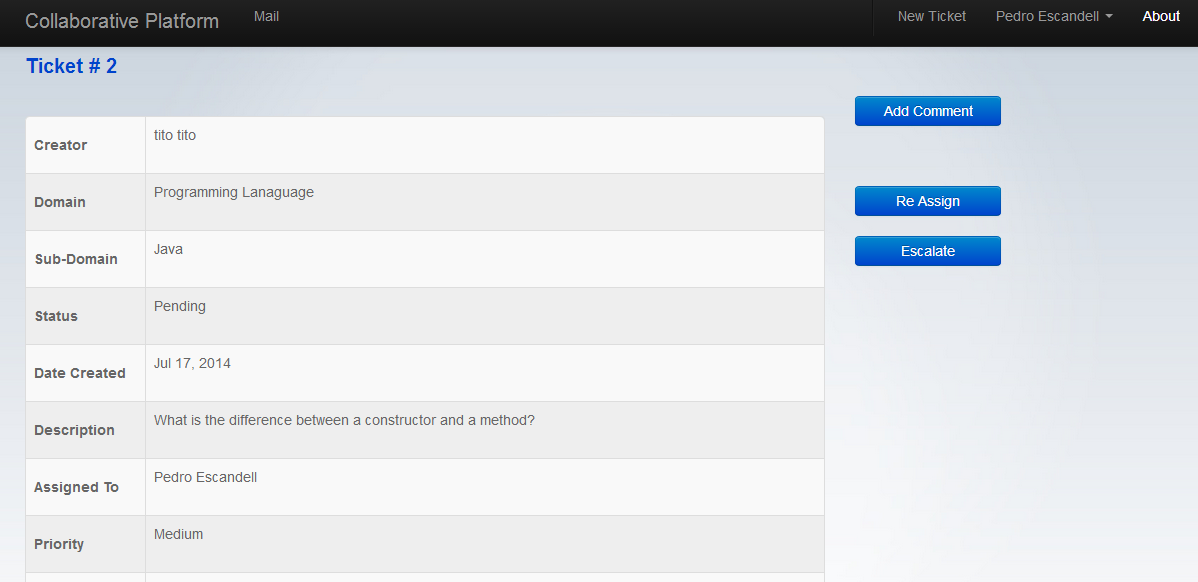


**Figure 9 Reassign Ticket Page**

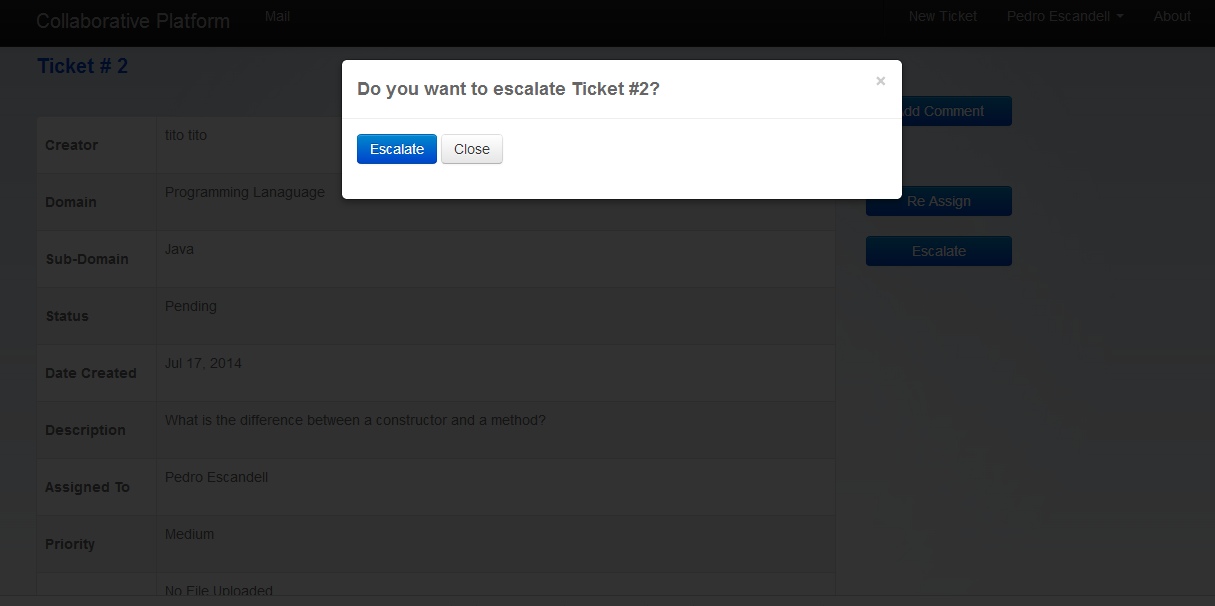


**Figure 10 Project Mentor Page**

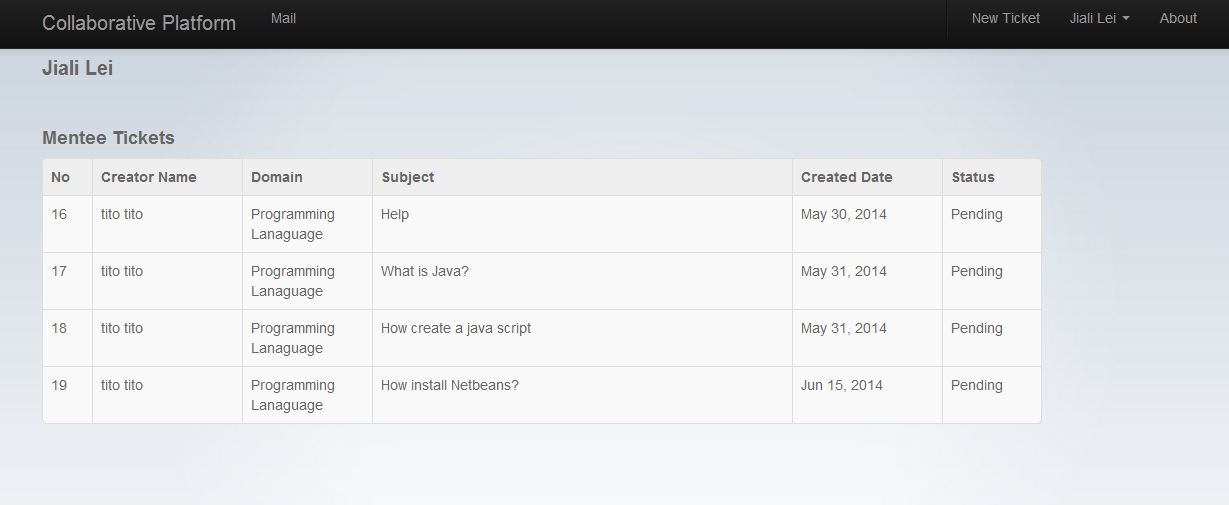
 **Figure 11 Manage Priorities Page**



**Figure 12 Escalation Page**

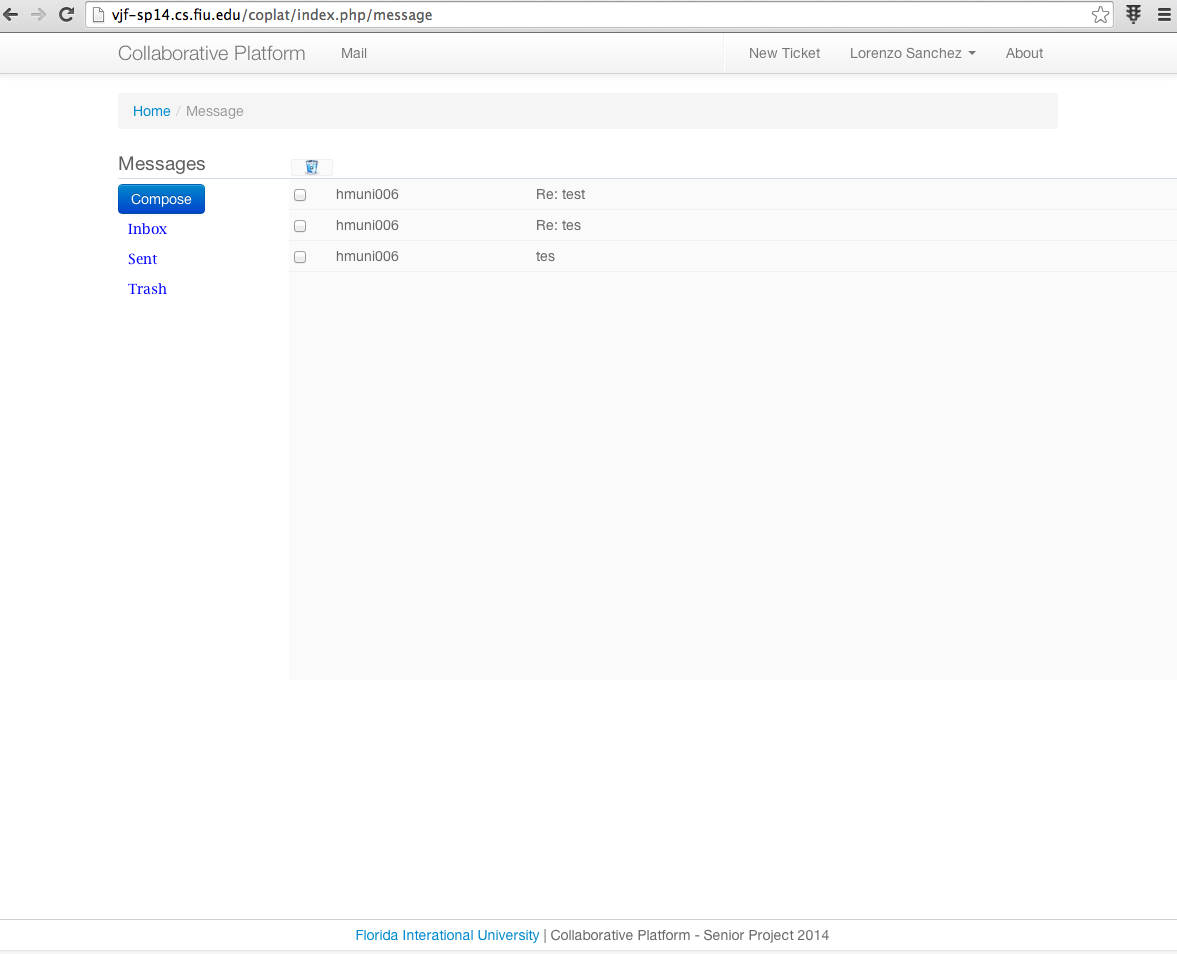


**Figure 12.1 Escalate ticket Page**

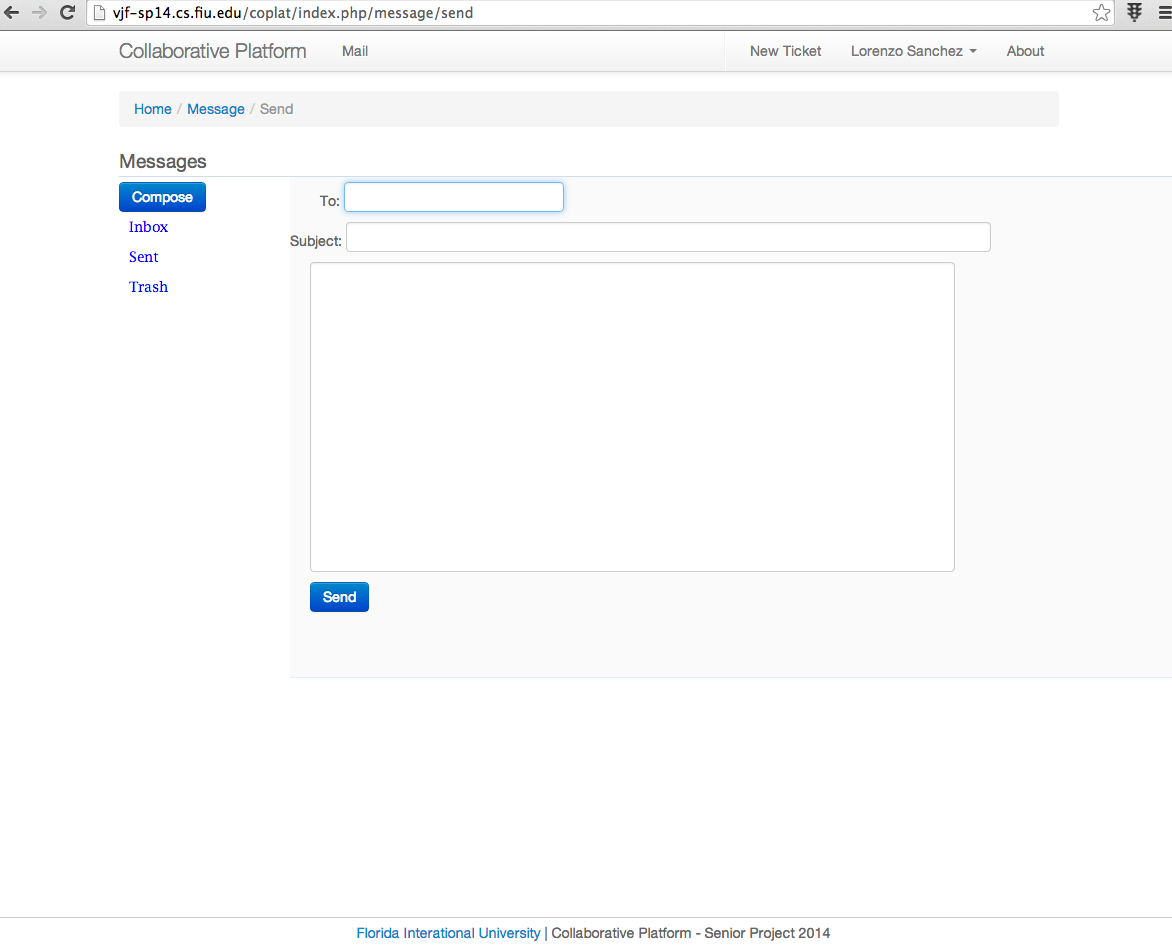


**Figure 13 Personal Mentor Page**

**Communication Subsystem**



**Figure 14 Mail Page**

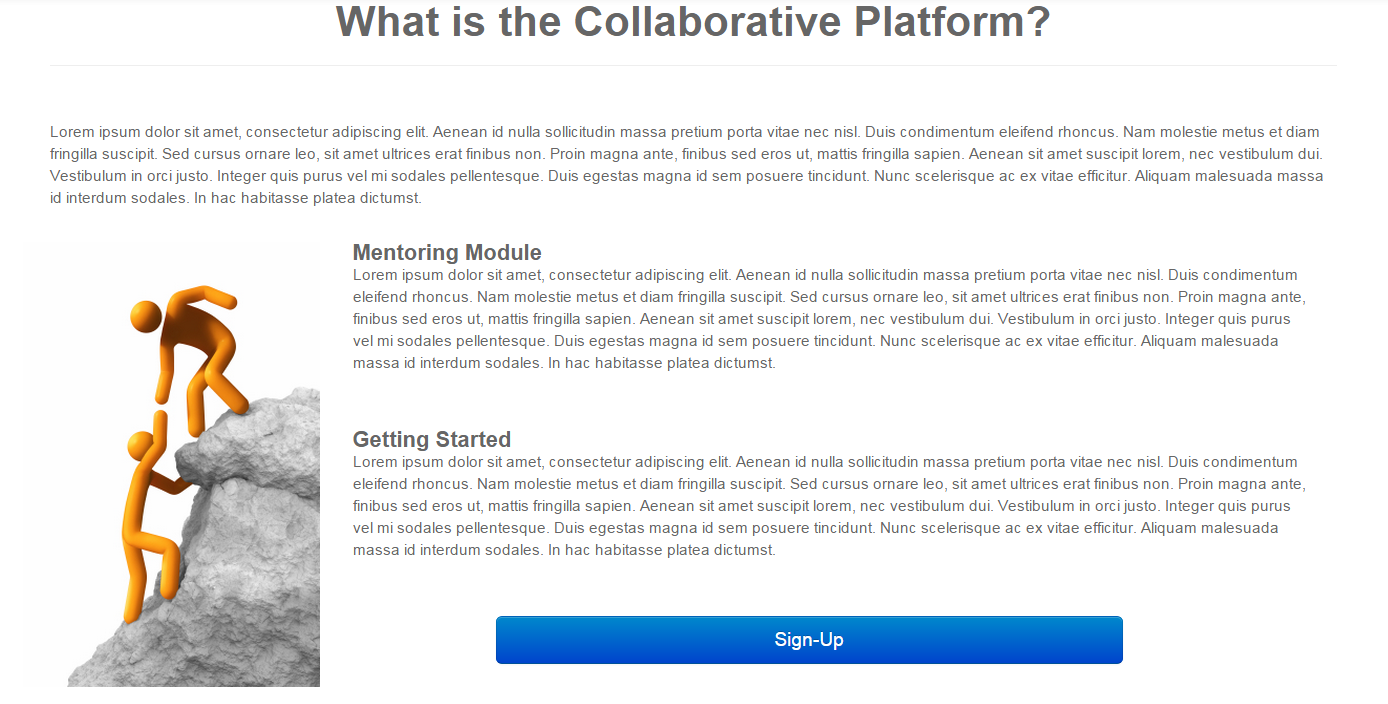
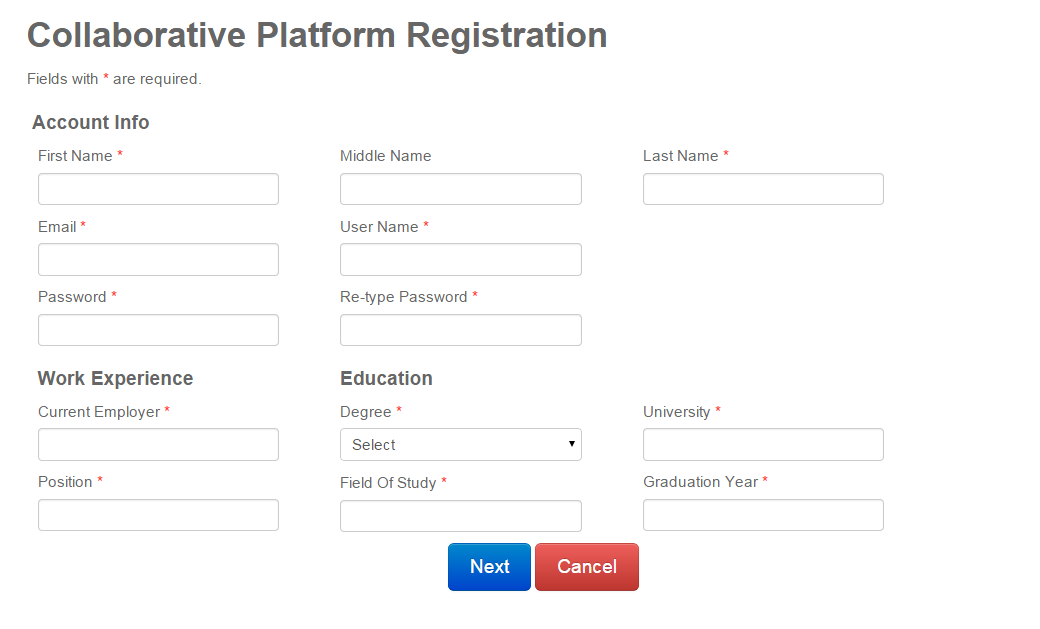


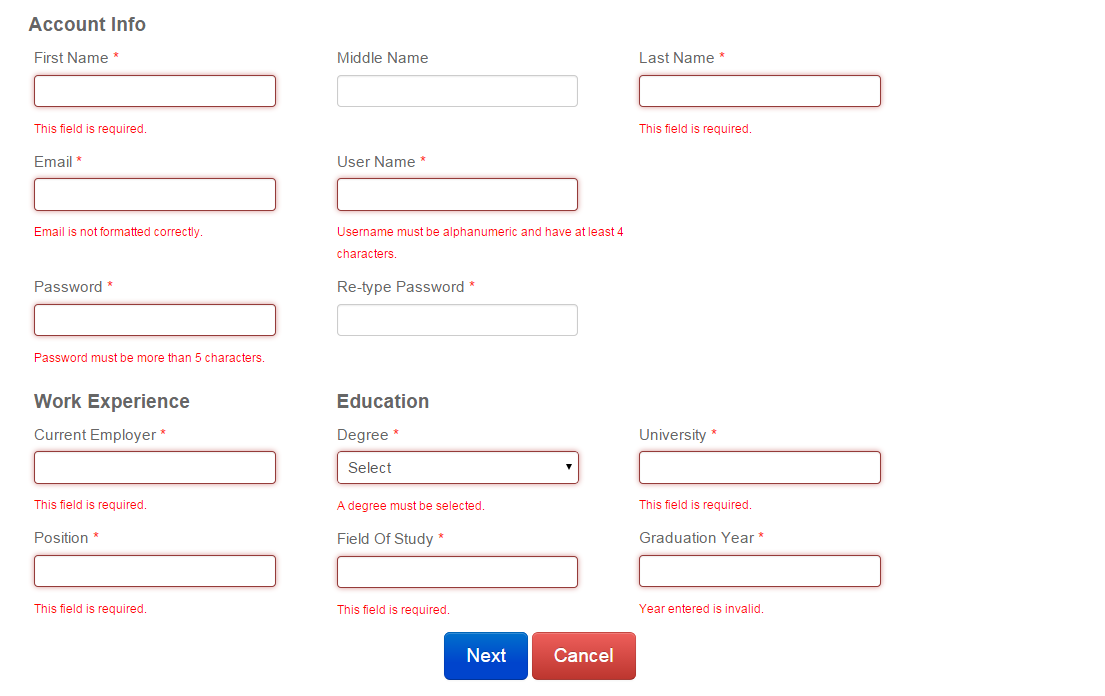
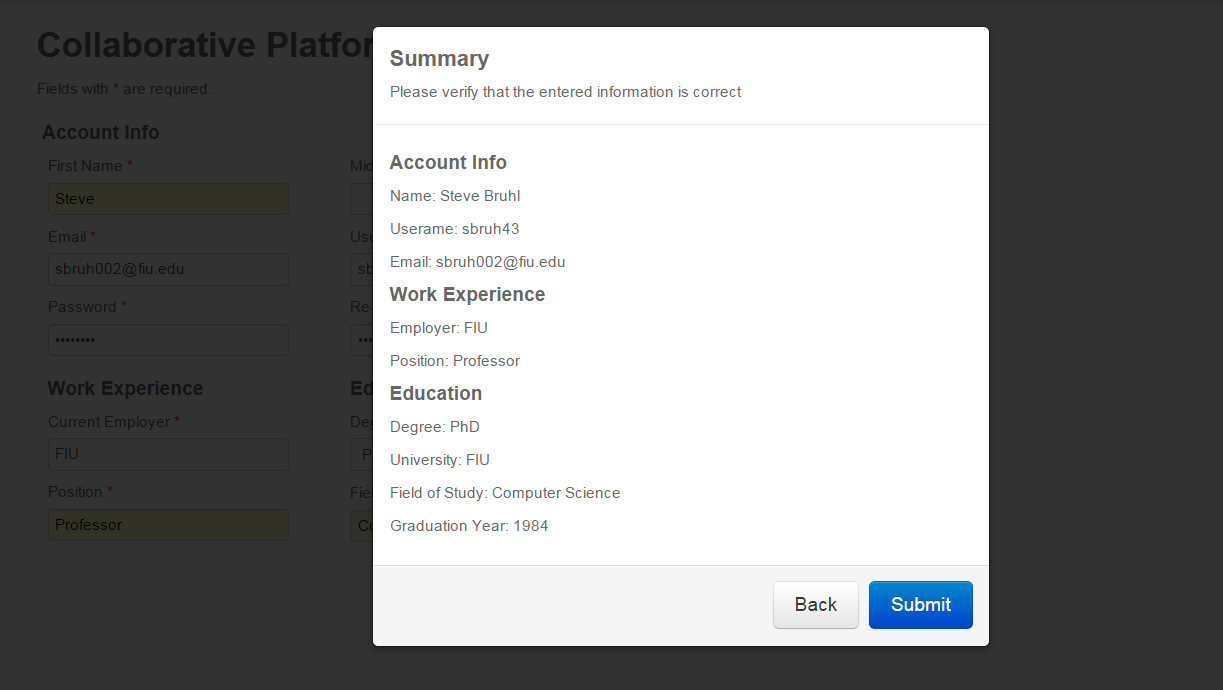
**Figure 15 Compose Email Page**

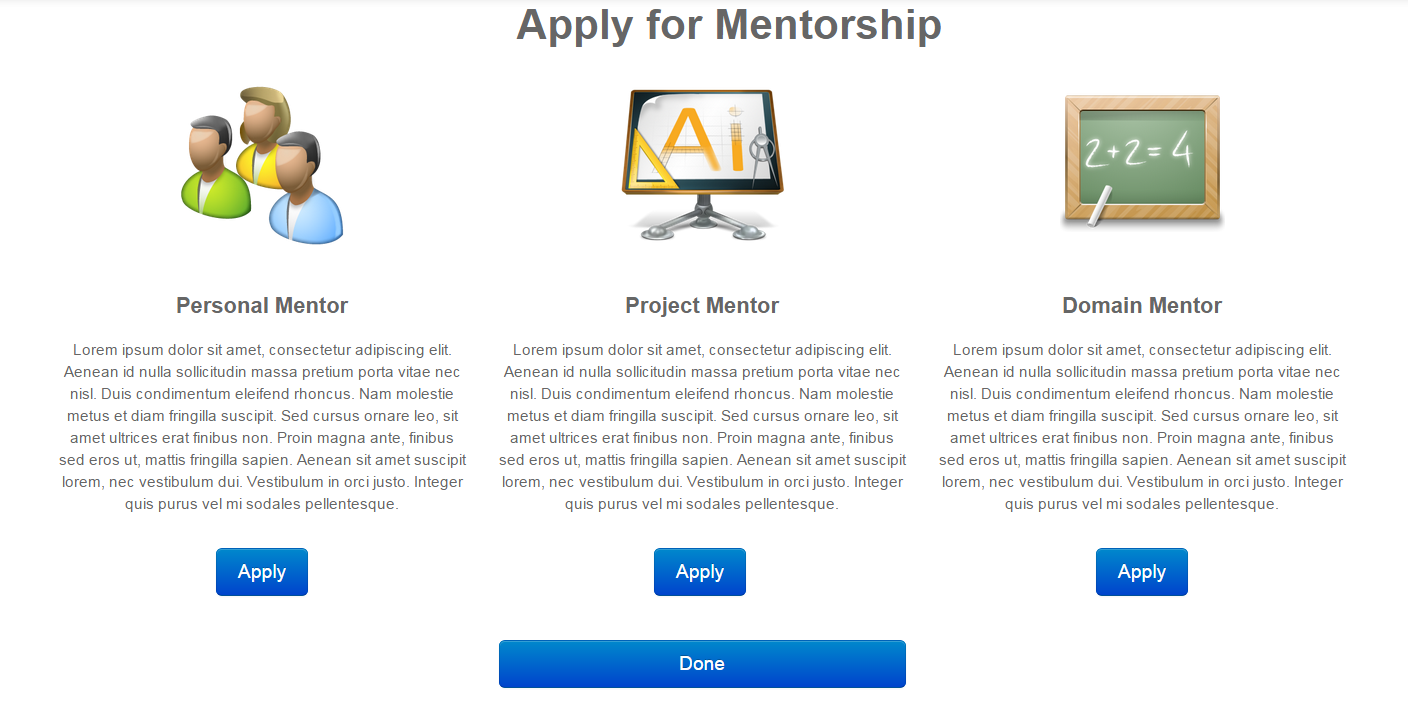
**UI Design for Collaborative Platform Version 3**

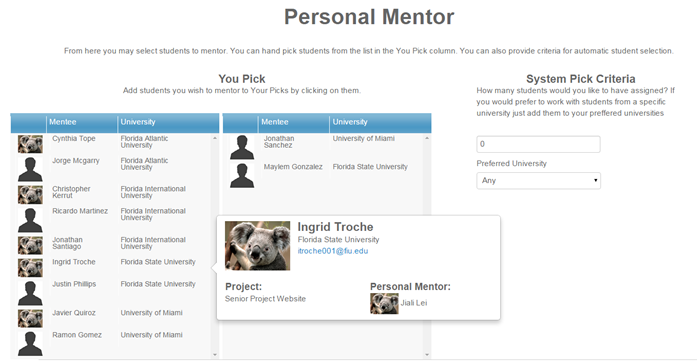
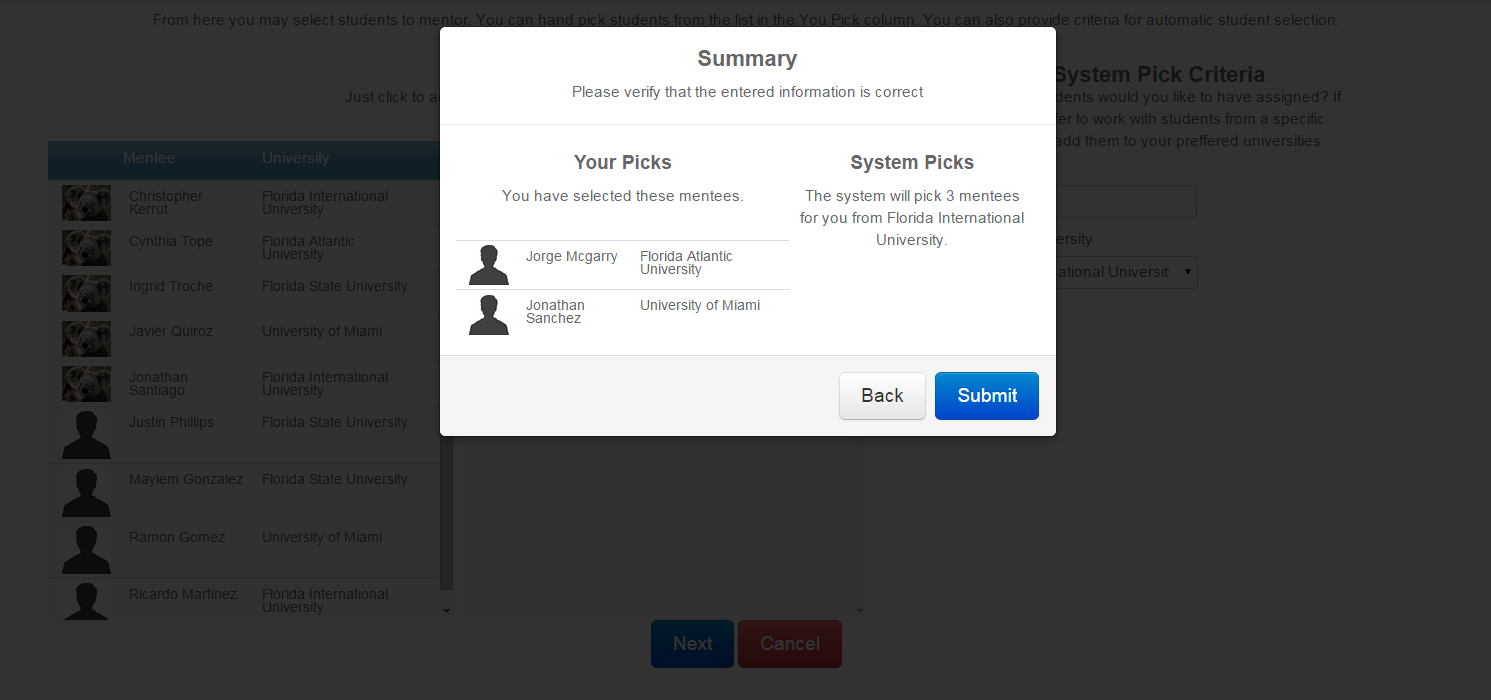
**Mentor Applications**

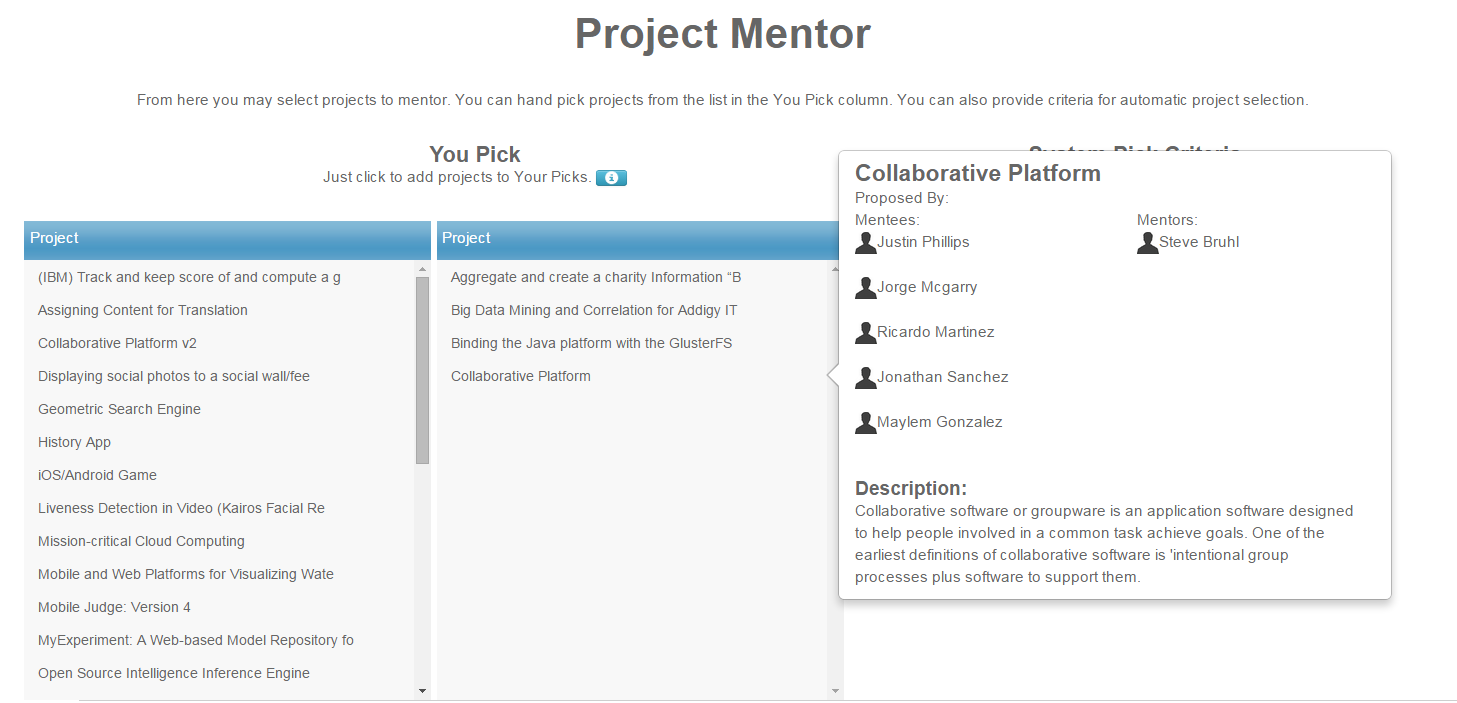
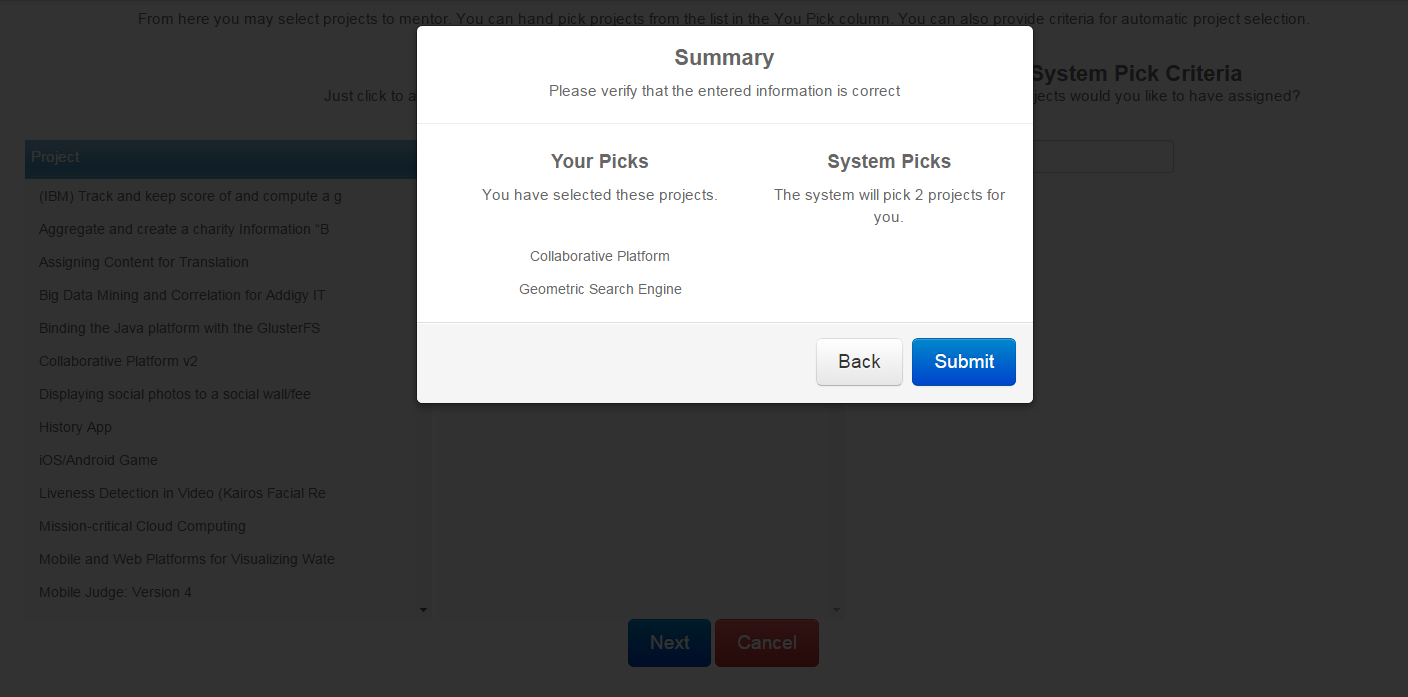
**Figure M1 - New Mentor Landing Page**

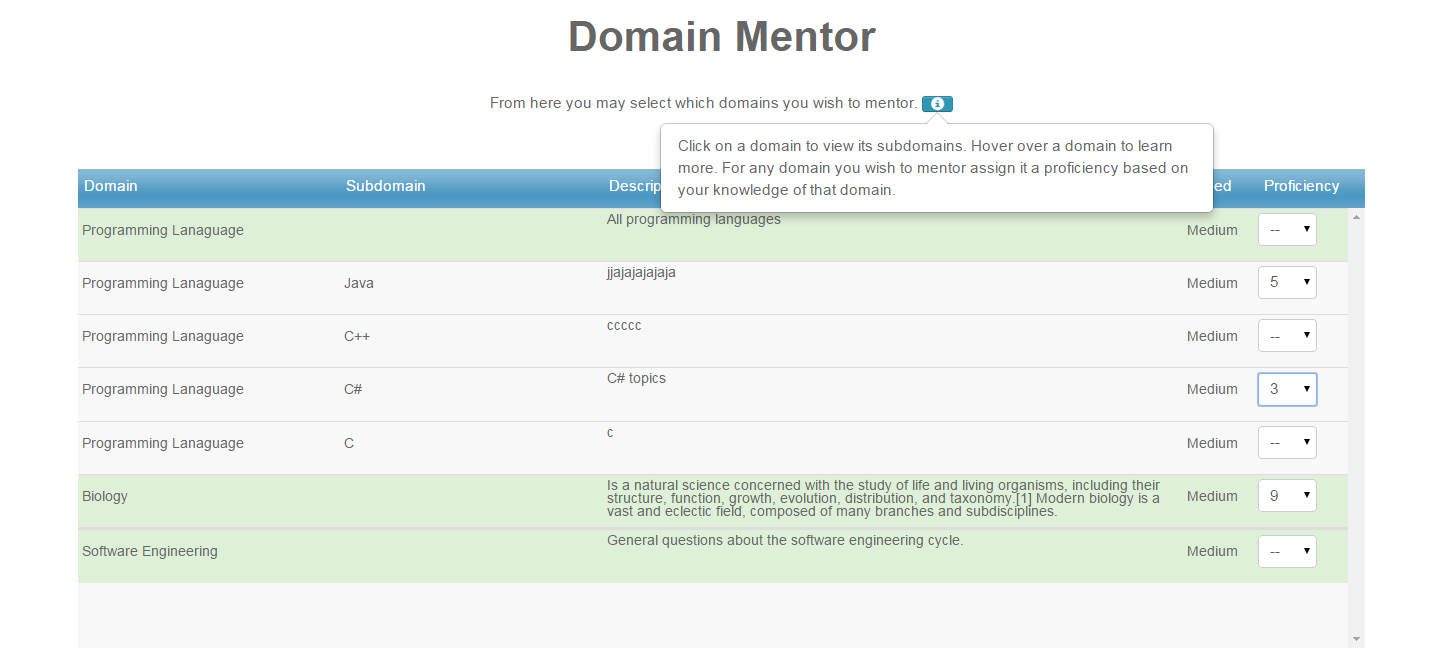
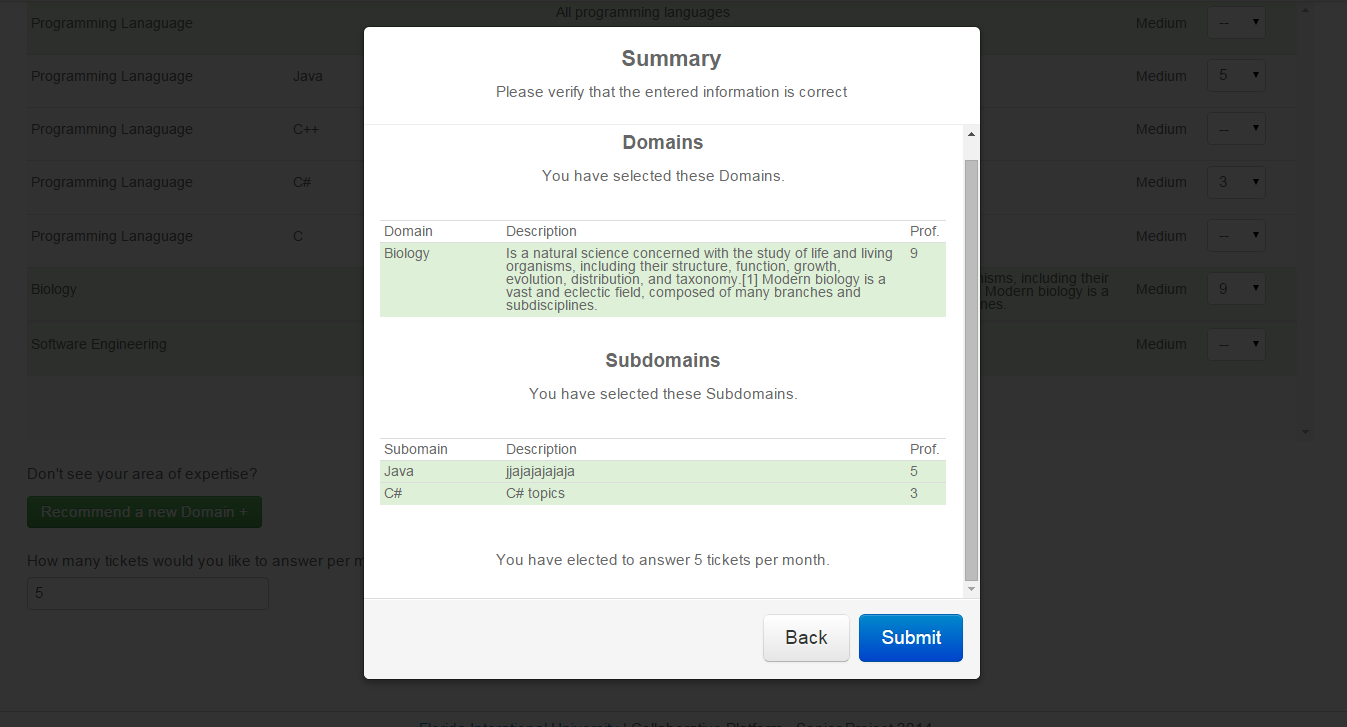
**Figure M2 - Account Registration**

**Figure M3 - Account Registration Errors****Figure M4 - Account Registration Verification**

**Figure M5 - Mentor Application Portal**

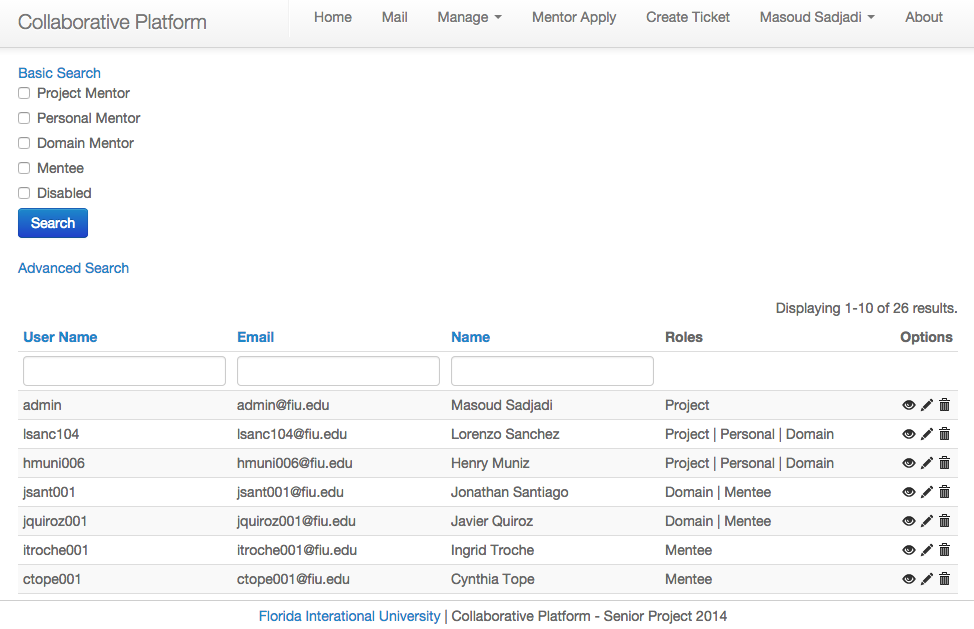
**Figure M6 - Personal Mentor Application****Figure M7 - Personal Mentor Application Verification**

**Figure M8 - Project Mentor Application****Figure M9 - Project Mentor Application Verification**

**Figure M10 - Domain Mentor Application****Figure M11 - Domain Mentor Application Verification**

**Admin Management**

**Figure A1 - Manage Users**



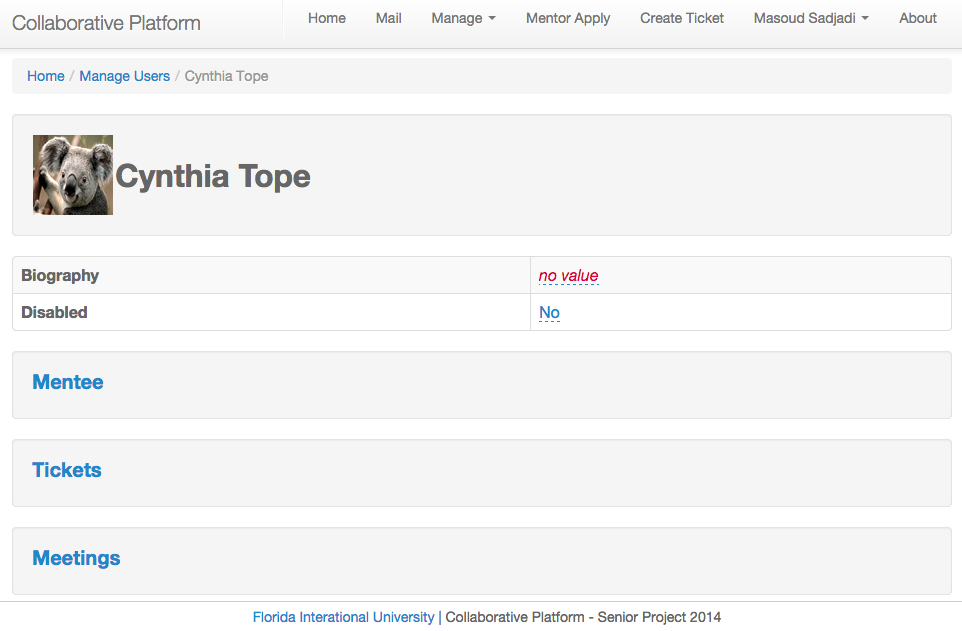
**Figure A2 - User Details**

Figure A3 - Project Details

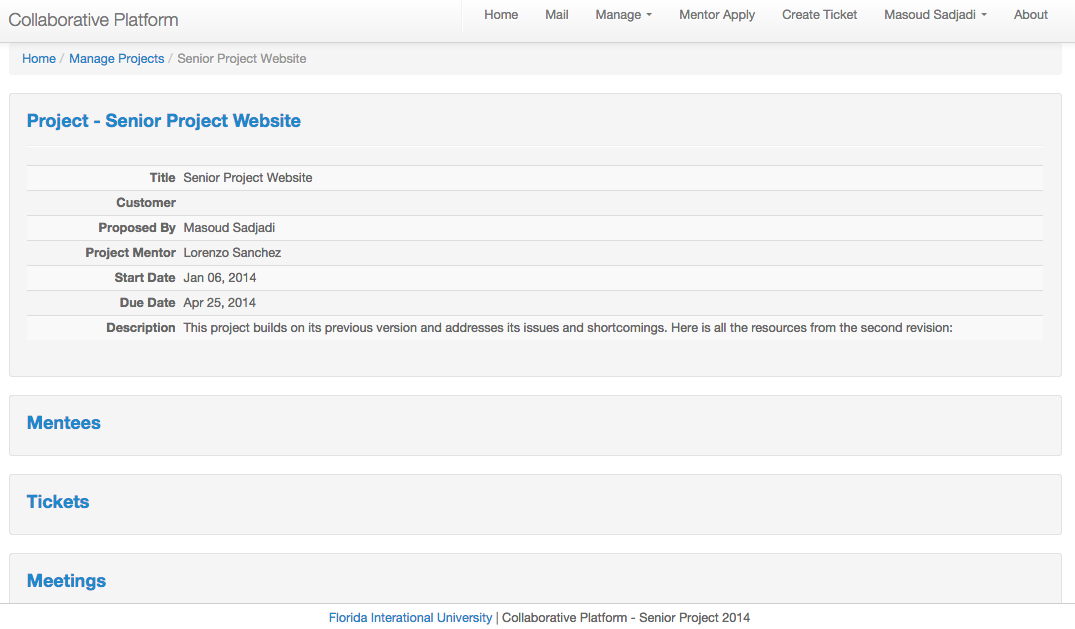


Figure A4 - View Domains/Subdomains

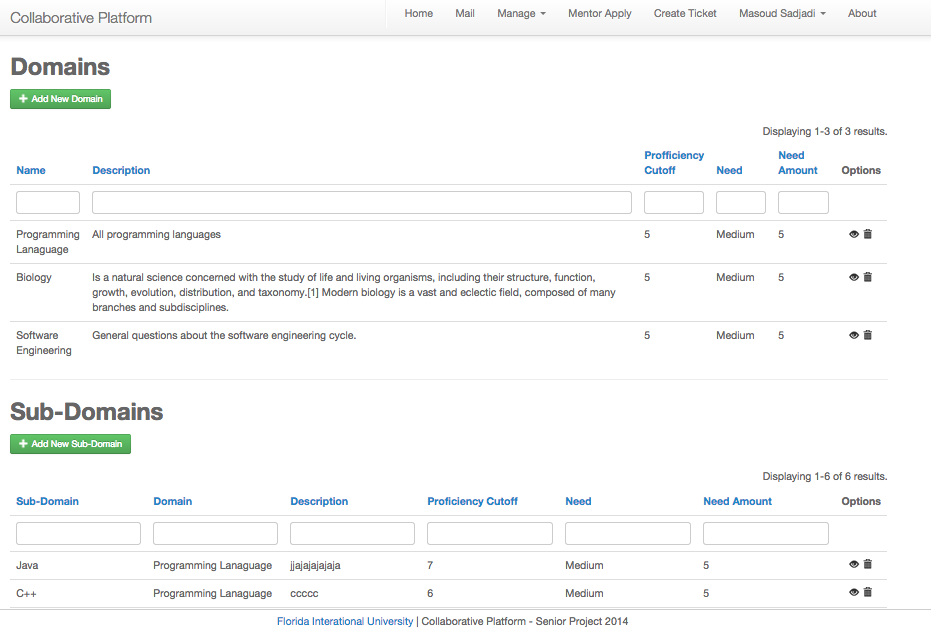


Figure A5 - Domain Details

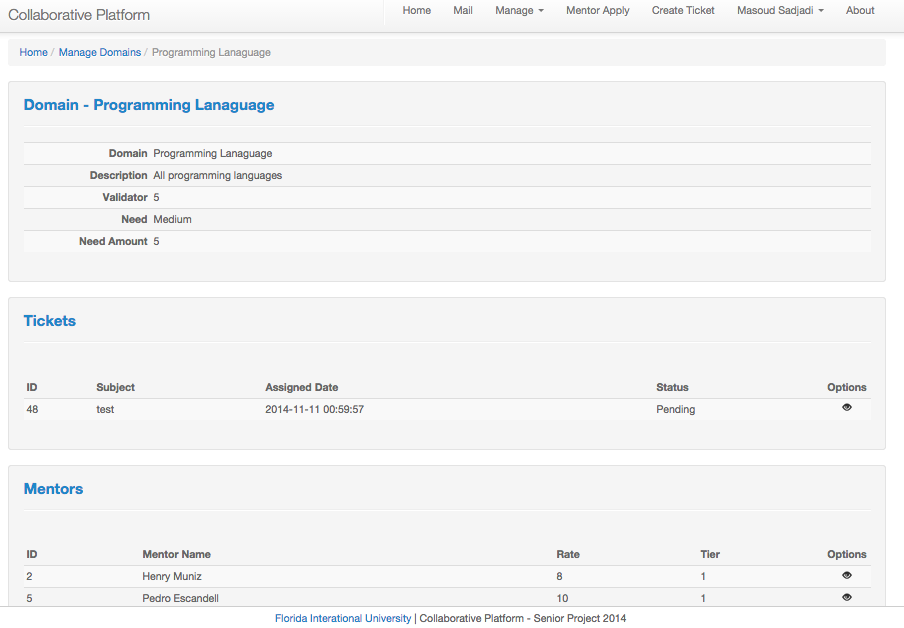


Figure A6 - Manage Tickets

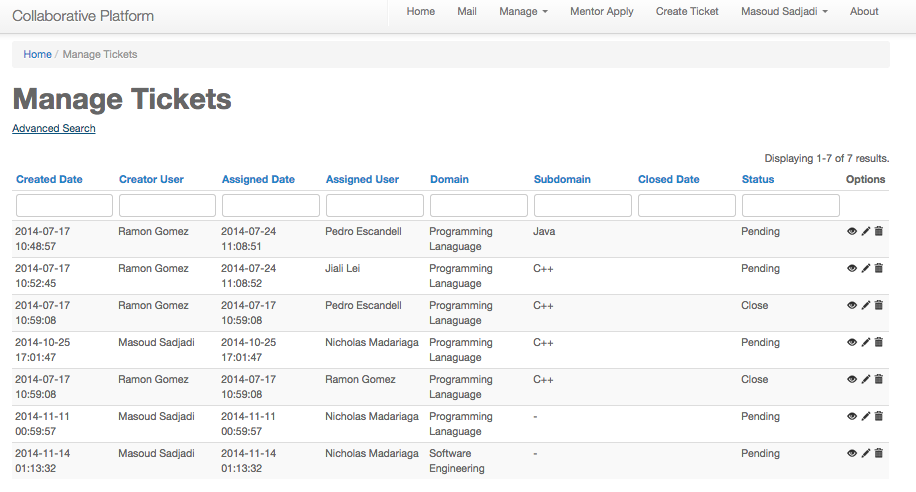


Figure A7 - Manage Tickets (Advanced Search)



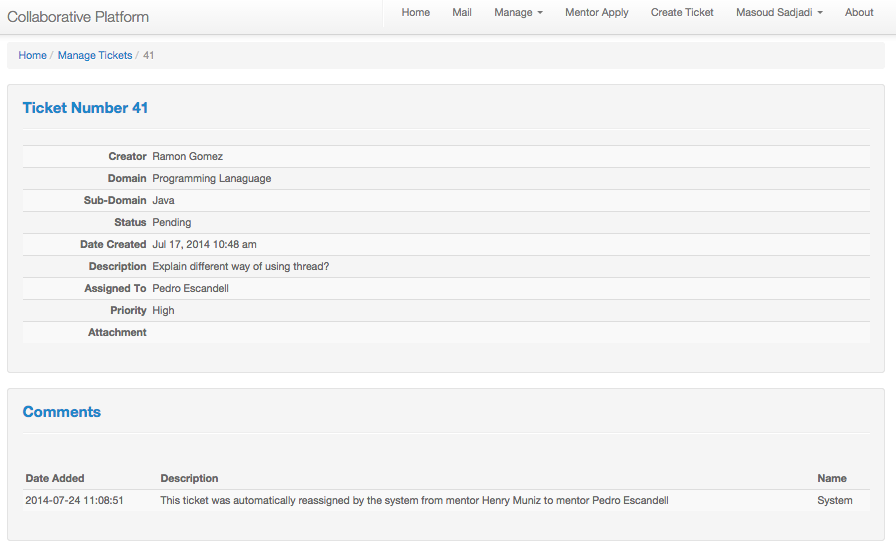
Figure A8 - View Ticket Details 

Figure A9 - Invite customization

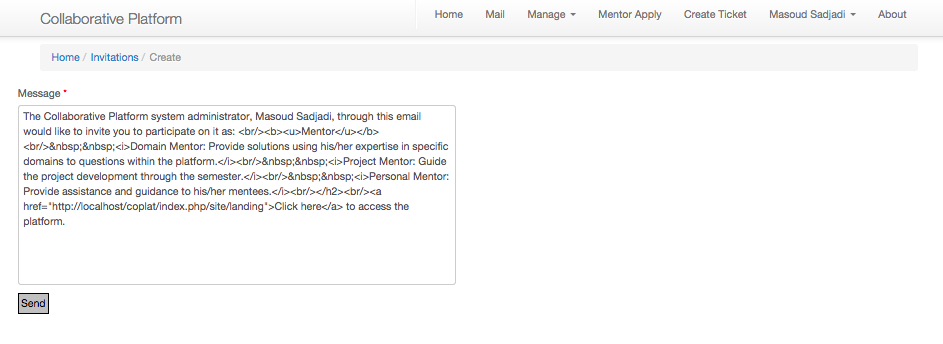


Figure A10 - Invite Details

## Screen Shot 2014-12-10 at 8.19.02 PM.png

Figure A11 - Manage Applications

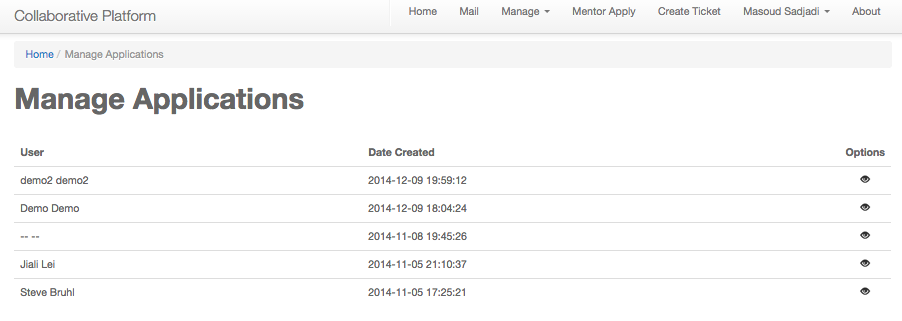


Figure A12 - Application Detail View

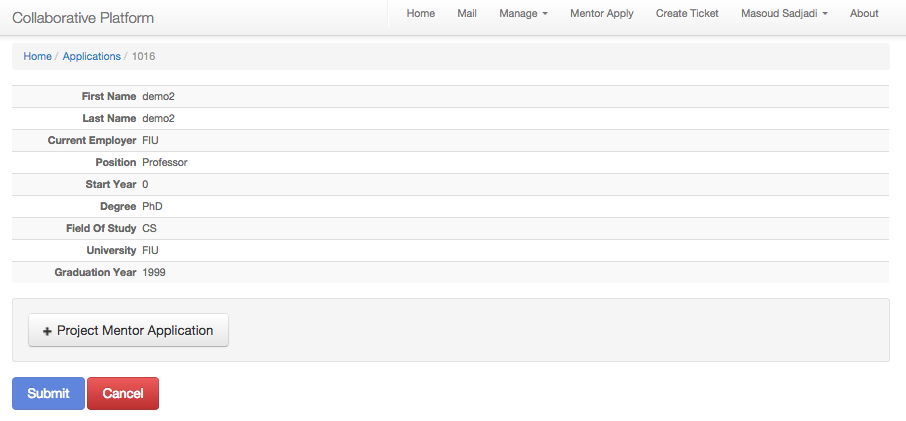
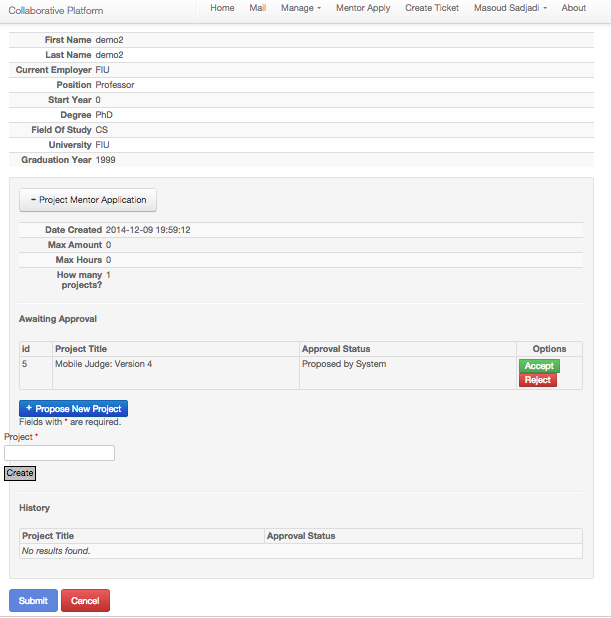


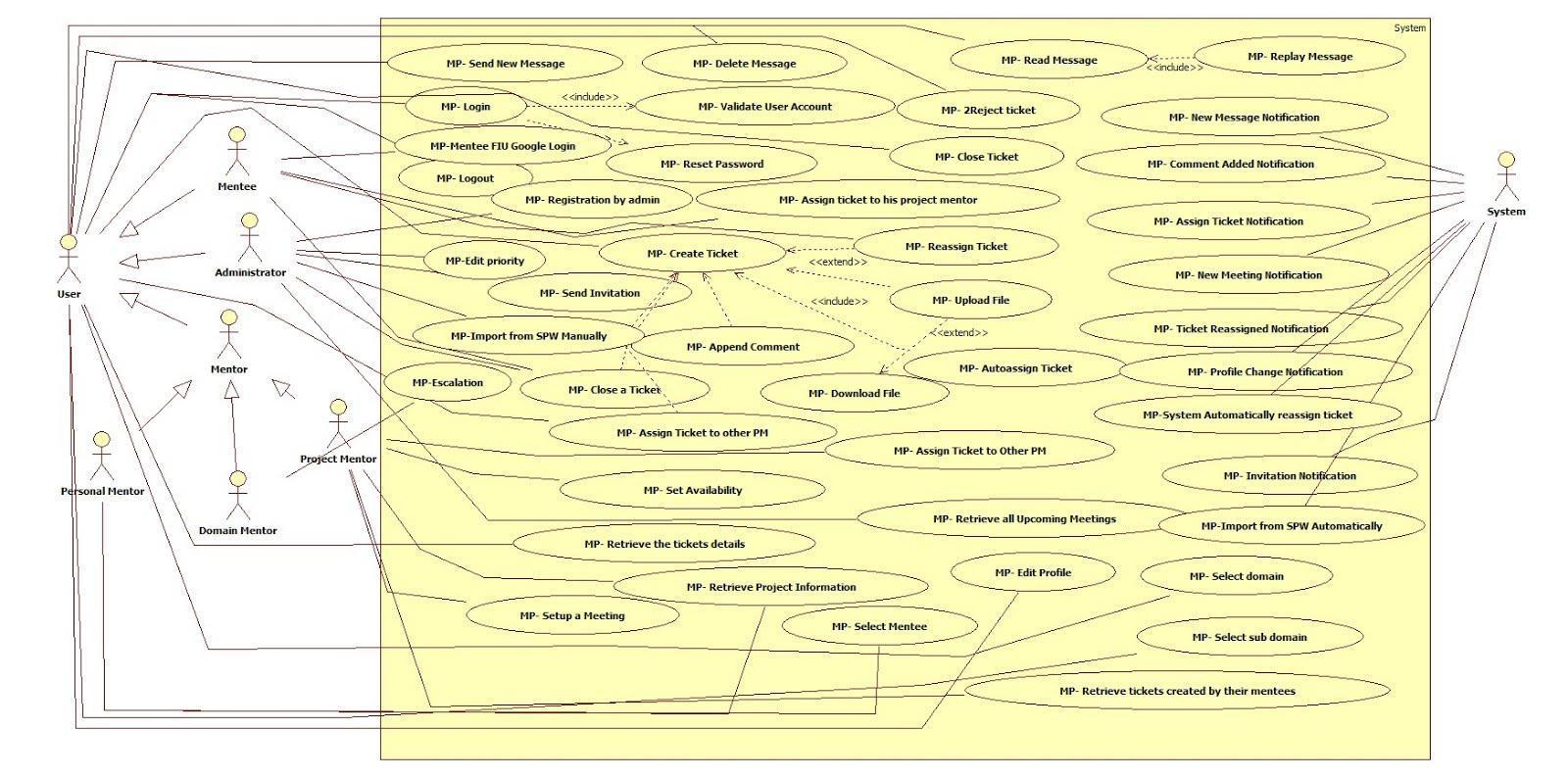
Figure A13 - Application (Accept/Reject)



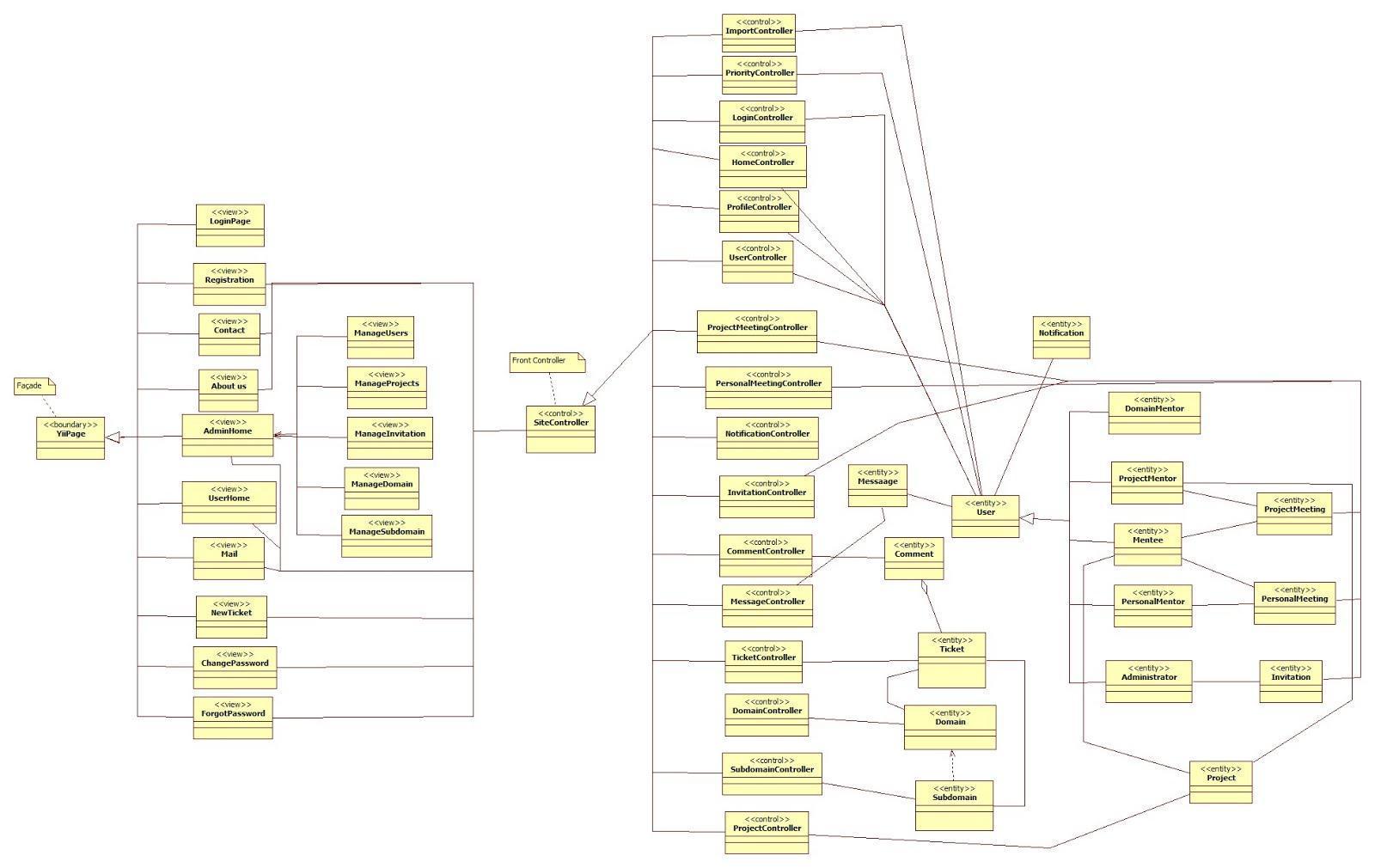
## 

## 

## 9.4 Appendix D – Analysis Model



**Figure 9.4.1 Use Case Model**



**Figure 9.4.2 Minimal Class Diagram**

**Mentoring Module: Version 3**

## CPv3-UDC-01: Mentor Application

## 

## 

## 

## CPv3-UDC-02: Admin Dashboard

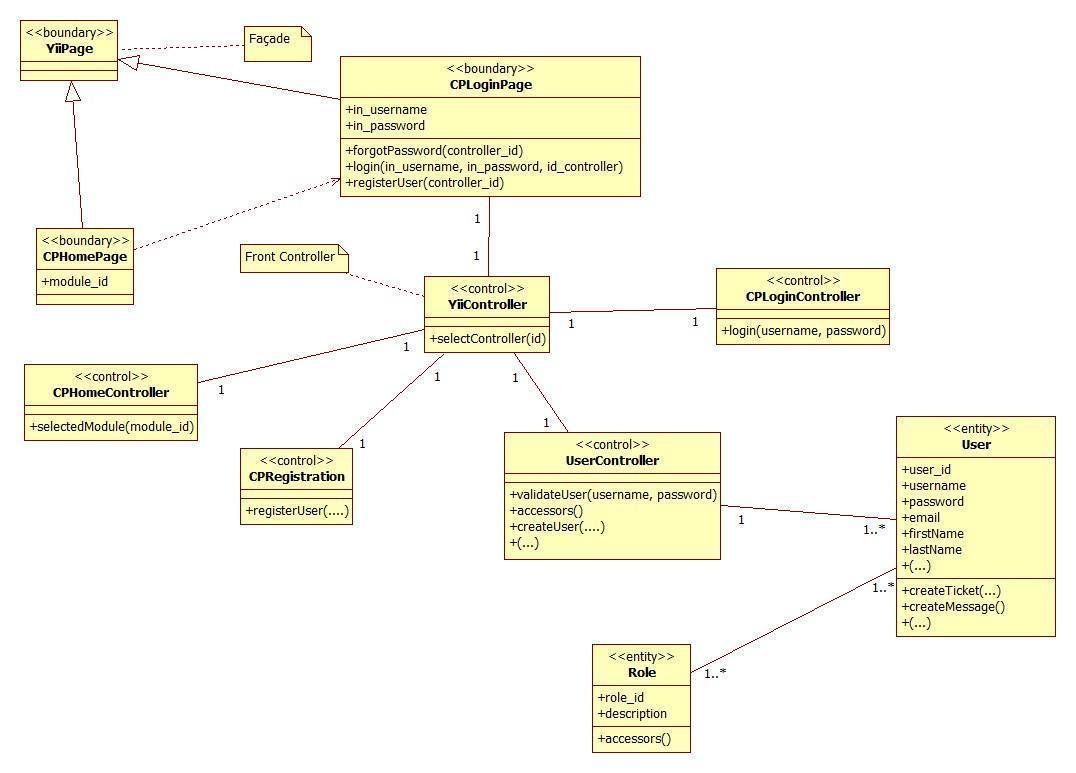
## usecasediagram.png

## 

## 

## 

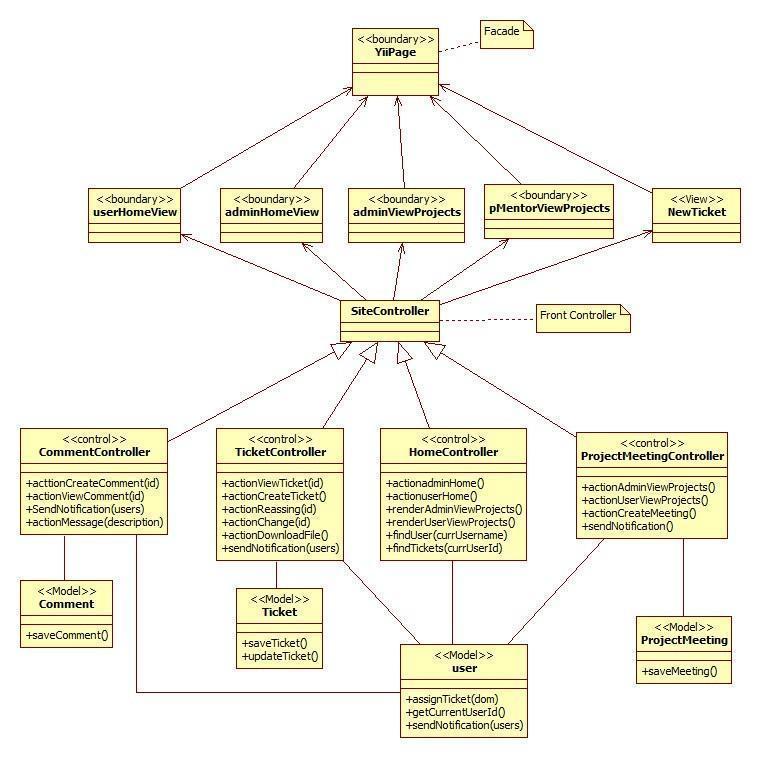
## 9.5 Appendix E – Design Model



**Figure 9.5.1 Access Point & Registration Detailed Class Diagram**



**Figure 9.5.2 Communication and Notification Subsystem Detailed Class Diagram**



**Figure 9.5.3 Mentoring Subsystem Detailed Class Diagram**

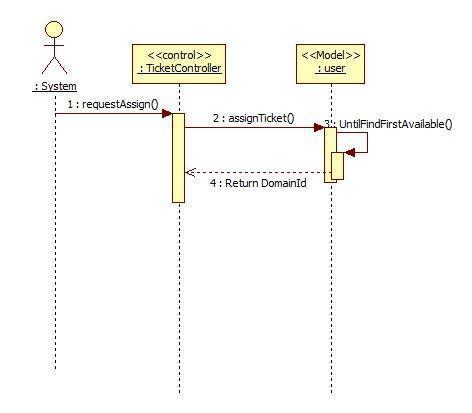
**Mentoring Module: Version 3**

**Figure 9.5.4: Admin Management Detailed Class Diagram**

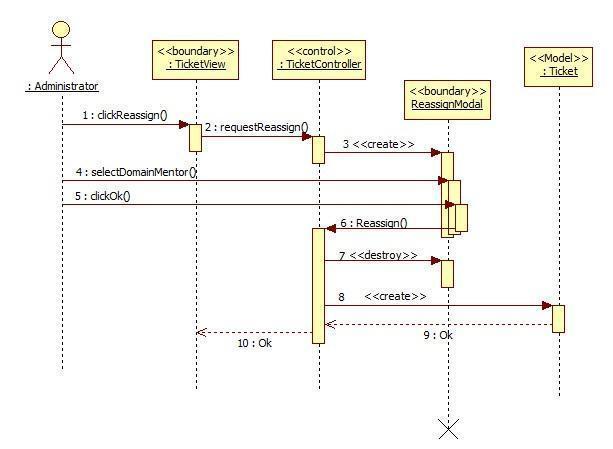
## Main.png

**Figure 9.5.5: Self-Registration Detailed Class Diagram**

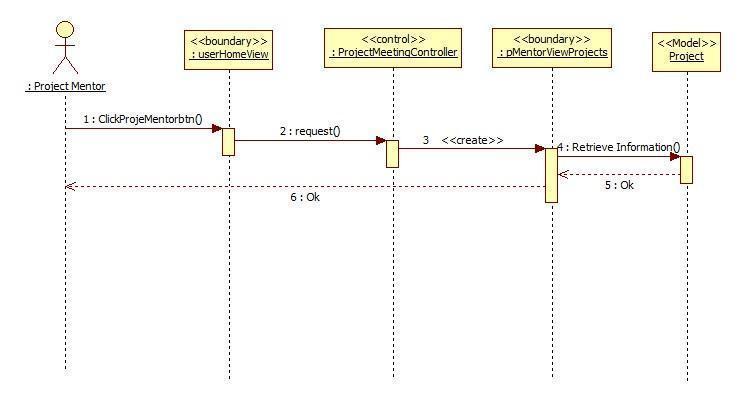
## 



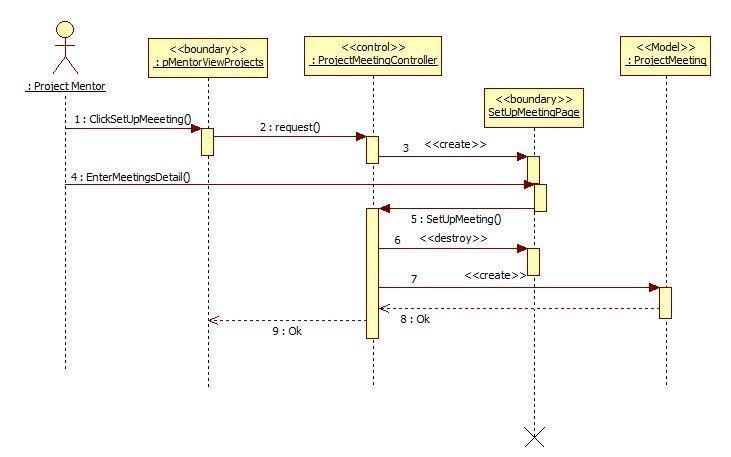
**Figure 9.5.4 MP-Auto Assign Ticket**



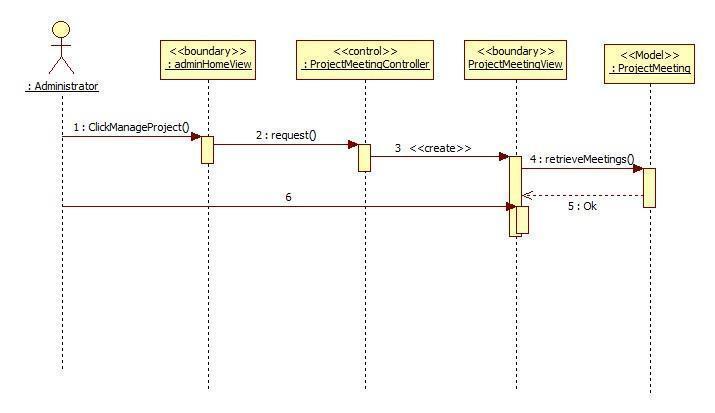
**Figure 9.5.5 MP-Reassign Ticket**



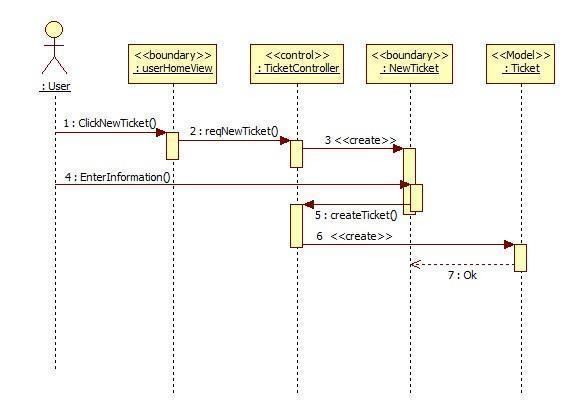
**Figure 9.5.6 MP-Retrieve the project information**



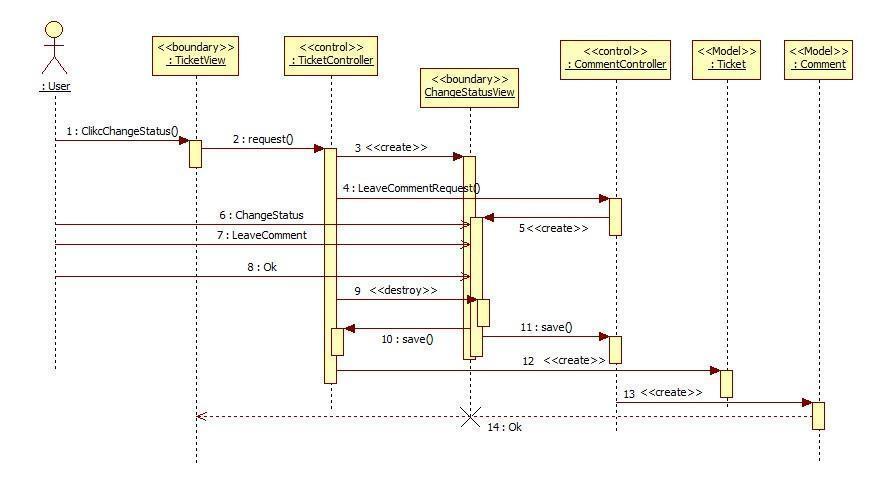
**Figure 9.5.7 MP-Setup Meetings**



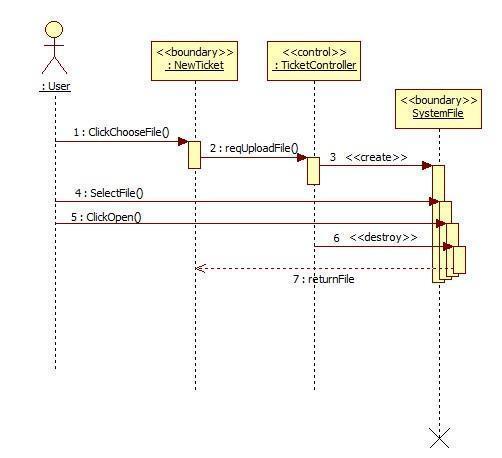
**Figure 9.5.8 MP-Retrieve all the upcoming meetings**



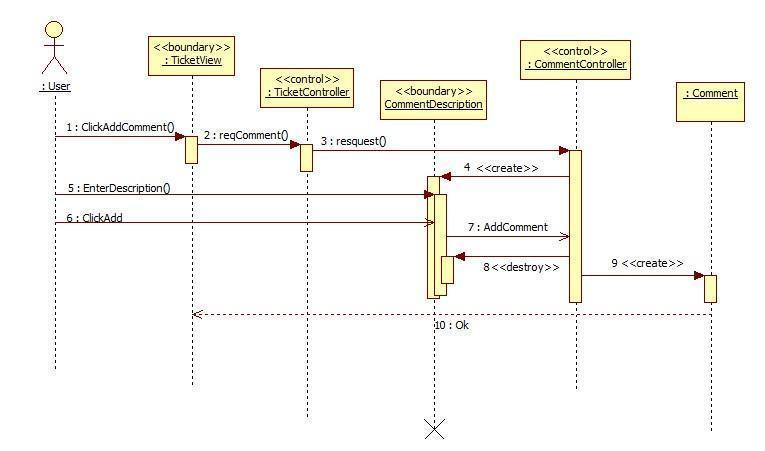
**Figure 9.5.9 MP-Reassign Ticket**



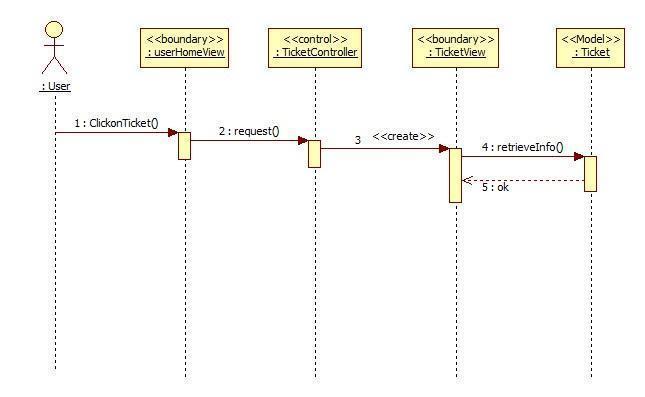
**Figure 9.5.10 MP-Change Status**



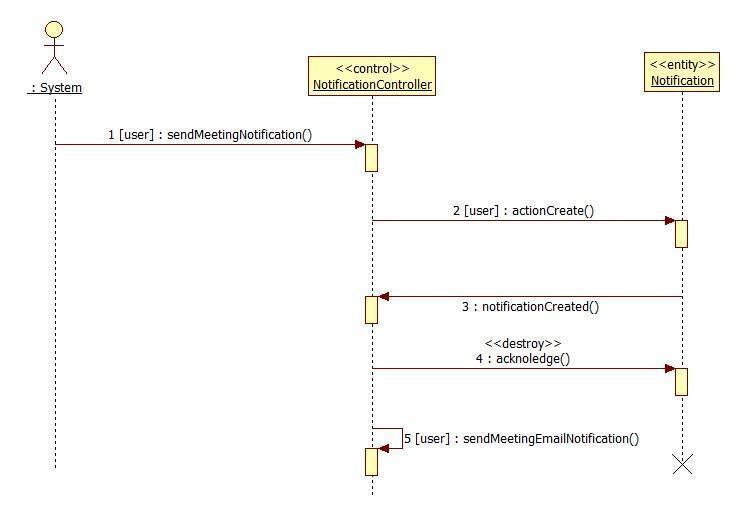
**Figure 9.5.11 MP-Upload Fie**



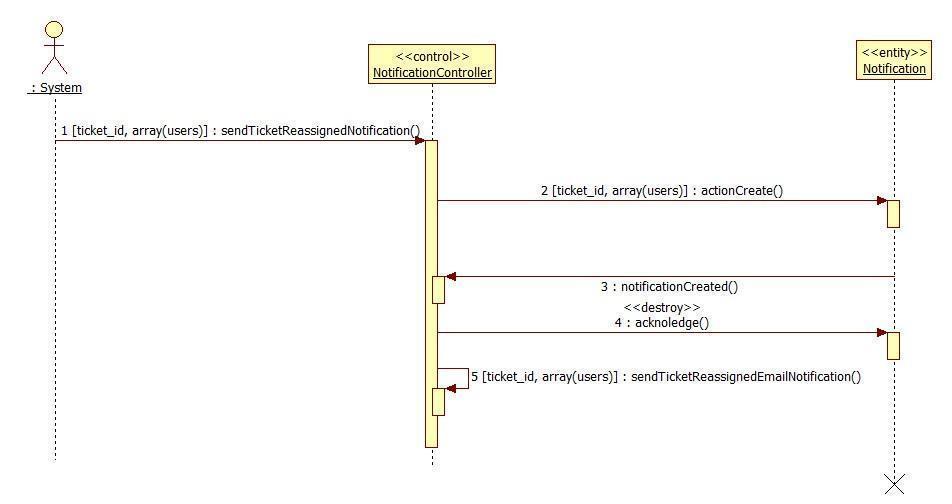
**Figure 9.5.12 MP-Append Comment**



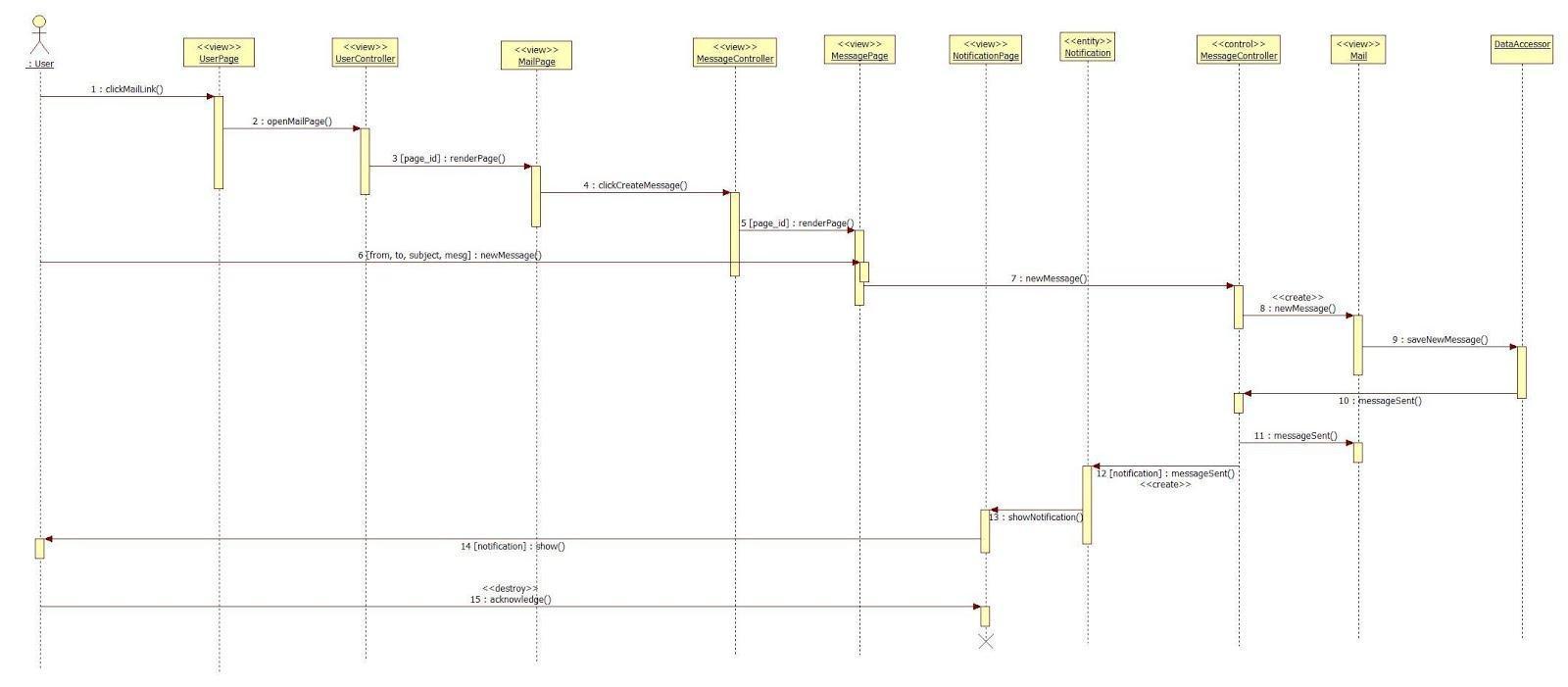
**Figure 9.5.13 MP-Retrieve the ticket detail**



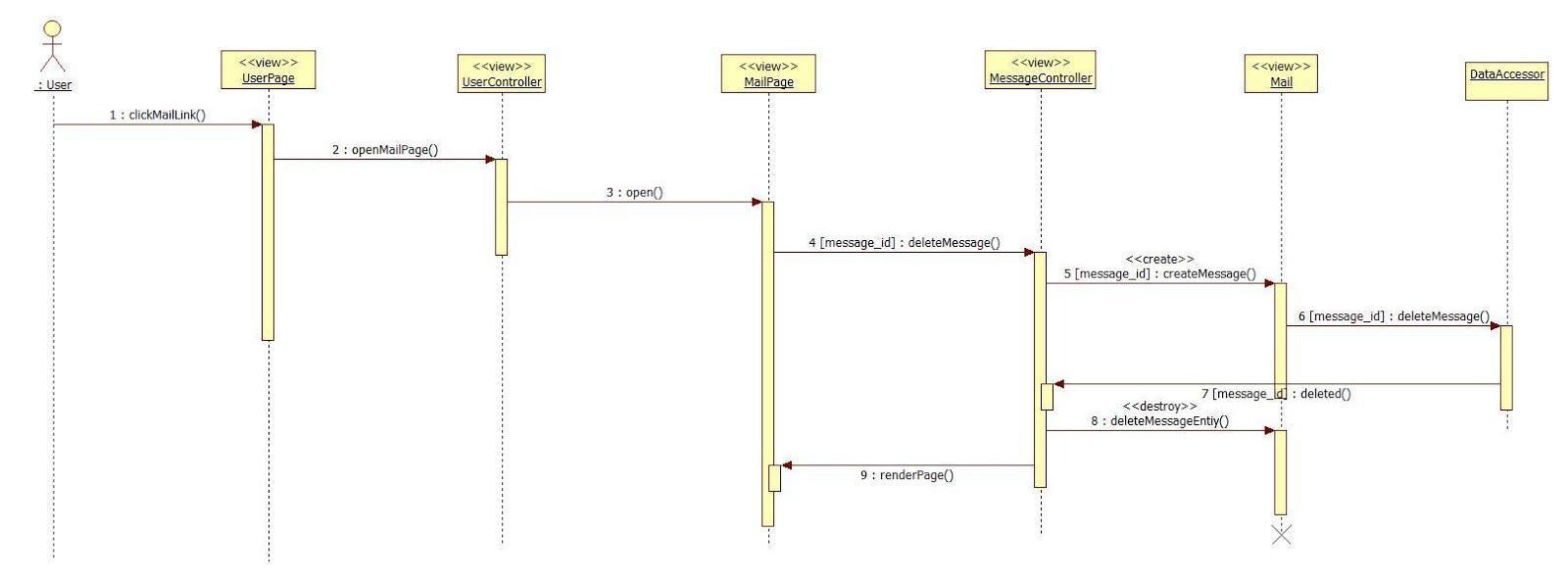
**Figure 9.5.14 New Meeting Notification**



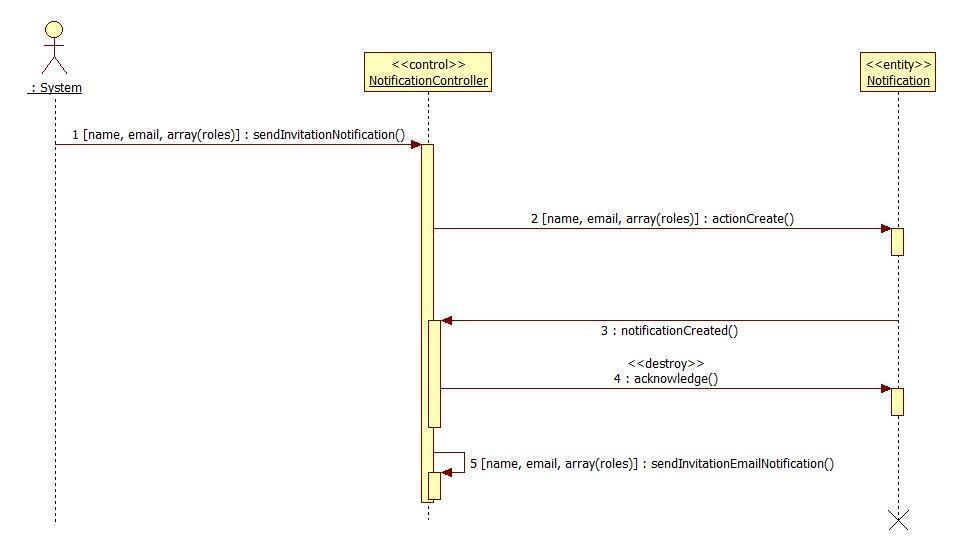
**Figure 9.5.15 Ticket Reassigned Notification**



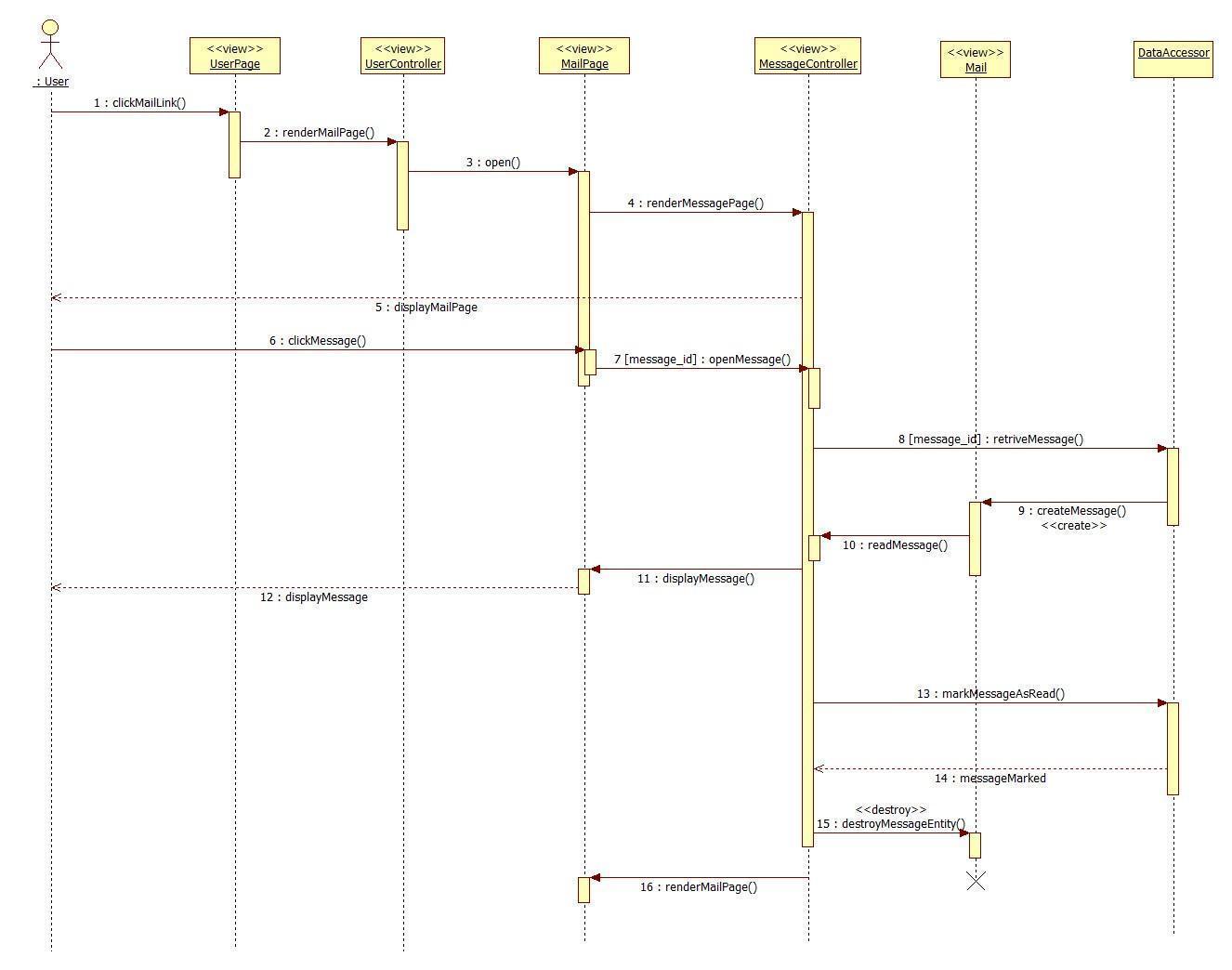
**Figure 9.5.16 Send New Message**



**Figure 9.5.17 Delete Message**



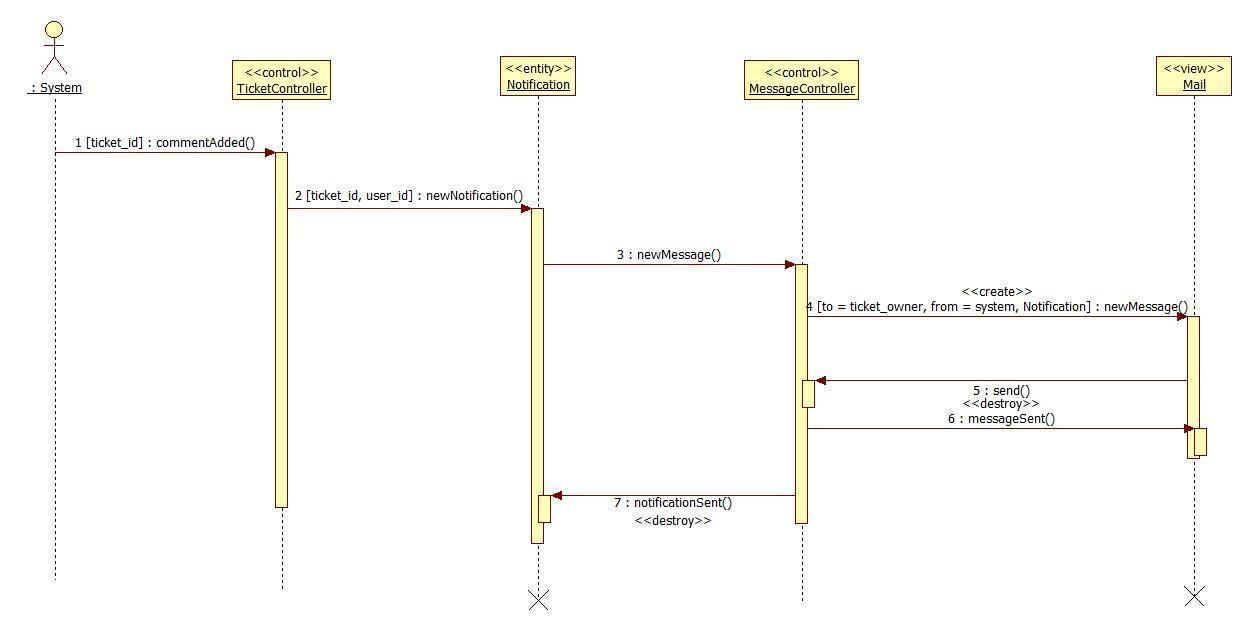
**Figure 9.5.18 Invitation Notification**



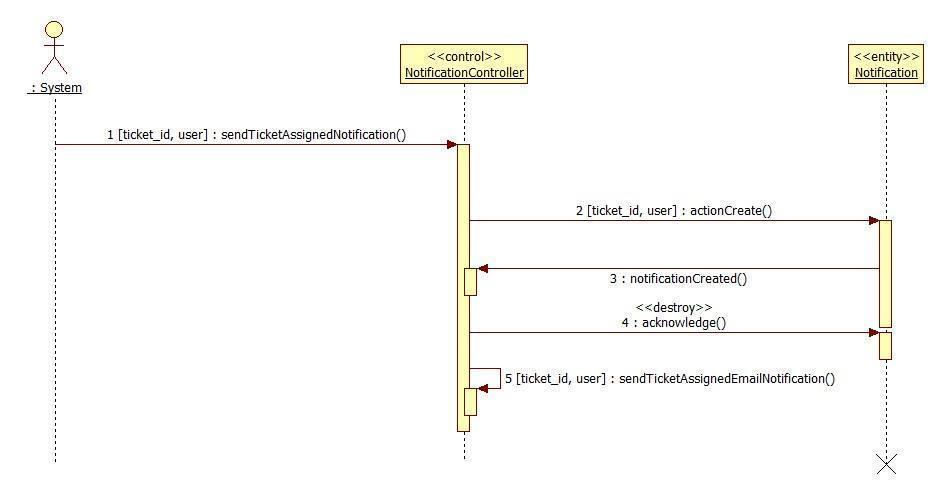
**Figure 9.5.19 Read Message**



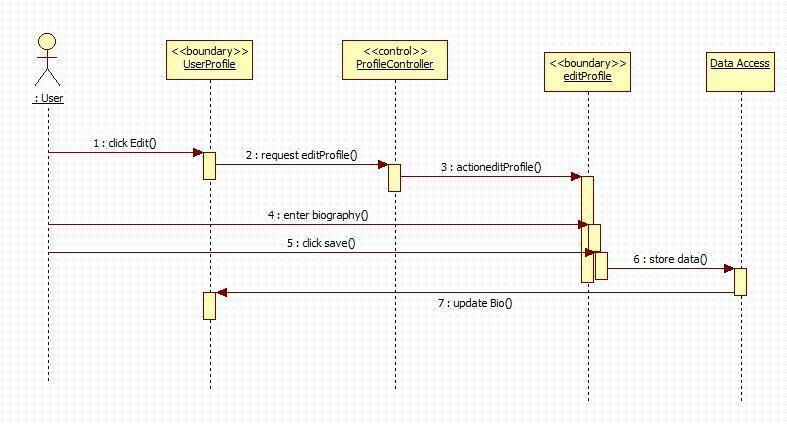
**Figure 9.5.20 New Message Notification**



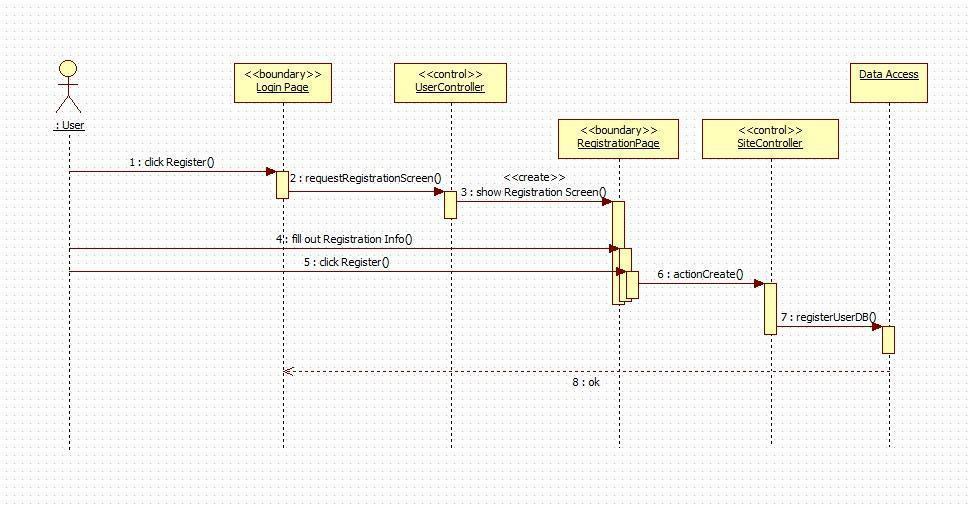
**Figure 9.5.21 Comment Added Notification**



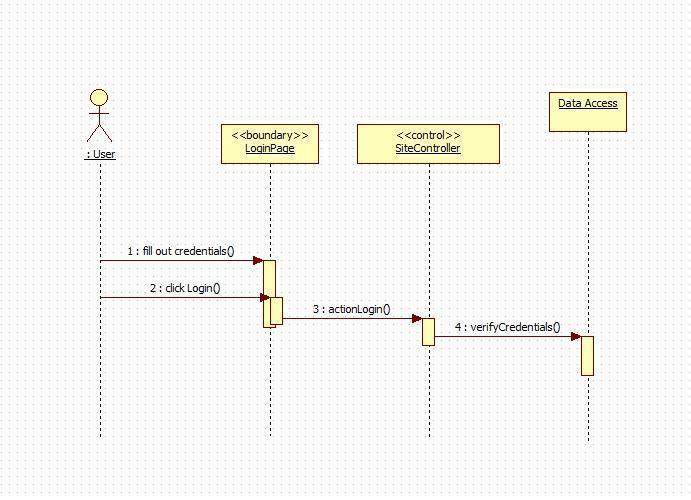
**Figure 9.5.22 Ticket Assigned Notification**



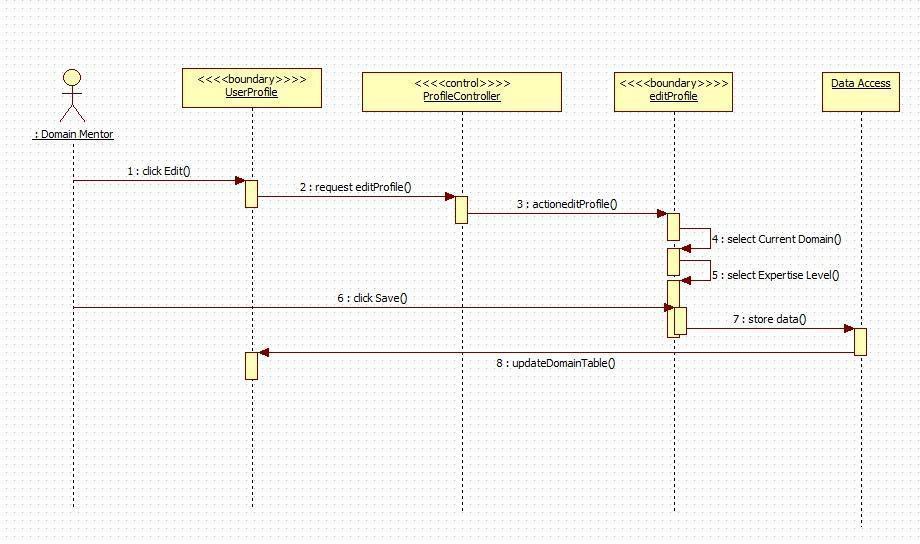
**Figure 9.5.23 Edit Biography**



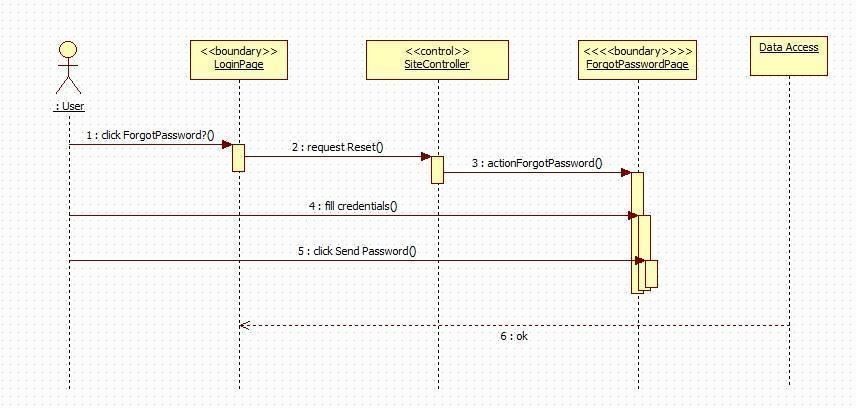
**Figure 9.5.24 Register**



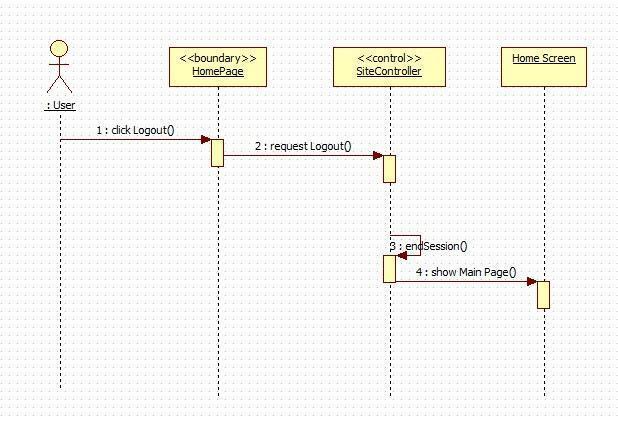
**Figure 9.5.25 Login**



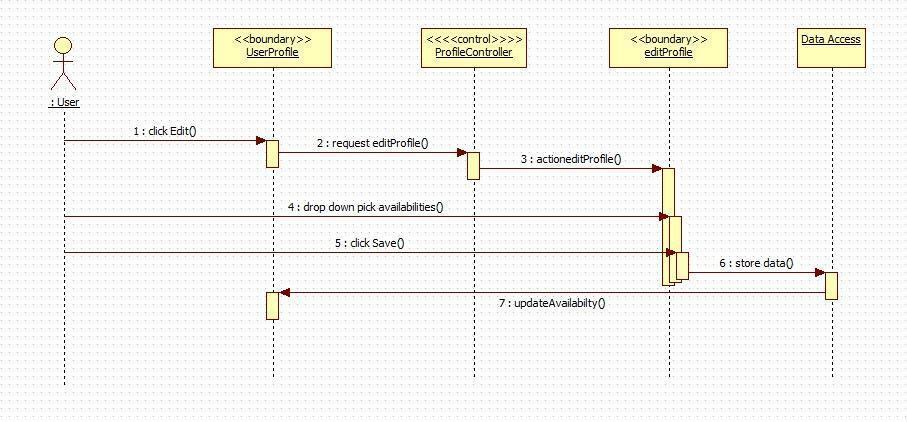
**Figure 9.5.26 Add Domain**



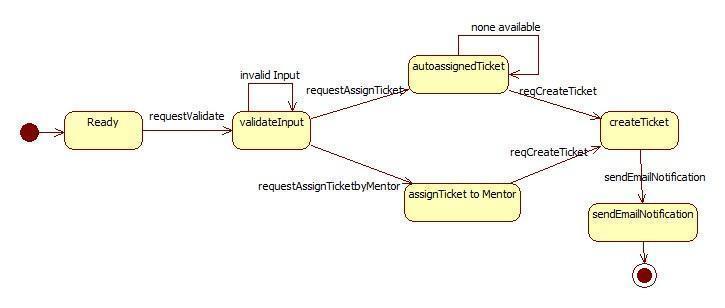
**Figure 9.5.27 Forgot Password**



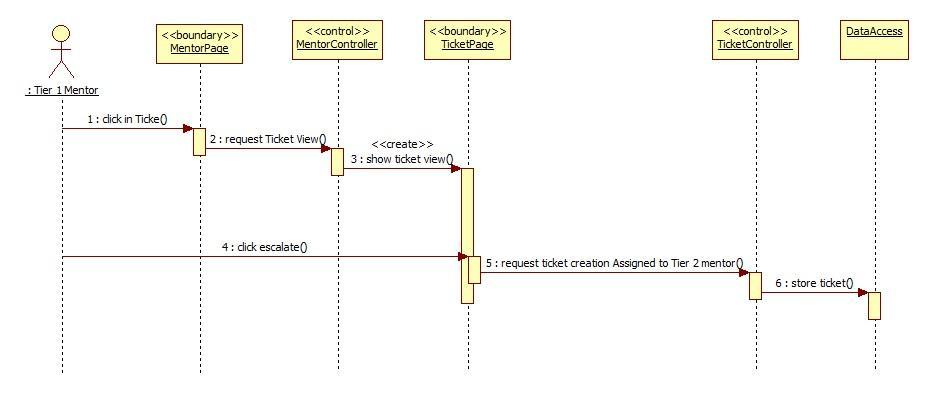
**Figure 9.5.28 Logout**



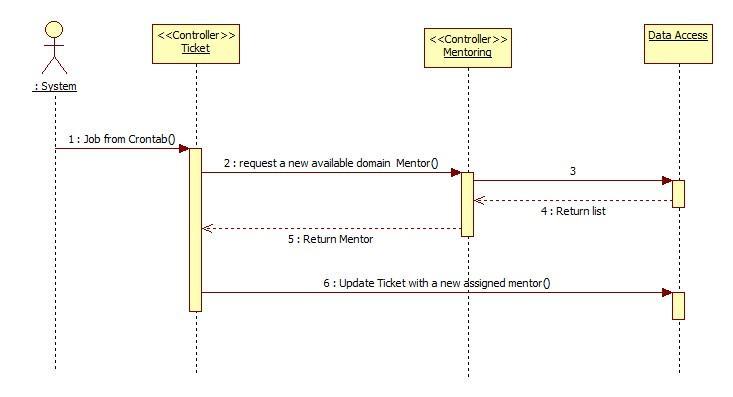
**Figure 9.5.28 Edit Availability**



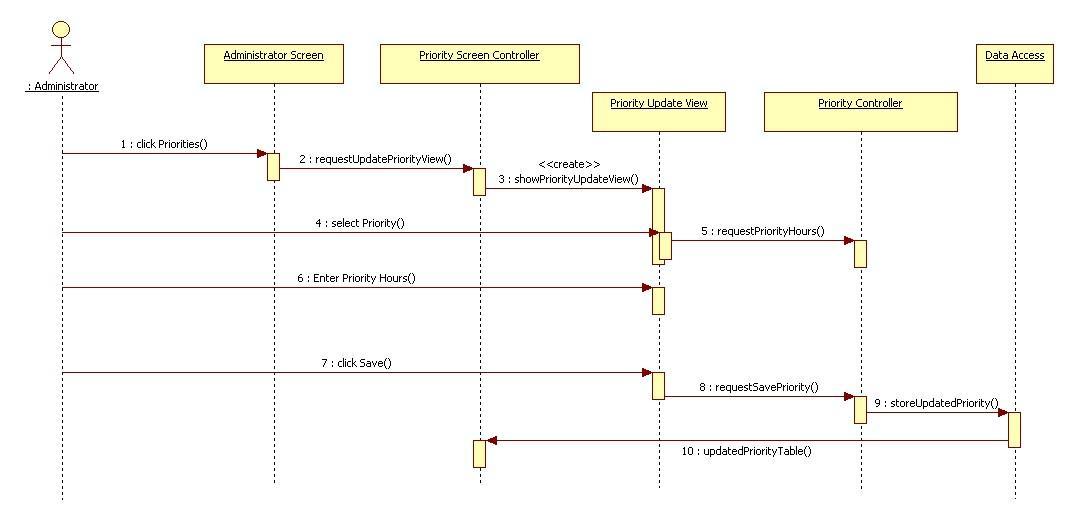
**Figure 9.5.23 Create Ticket**



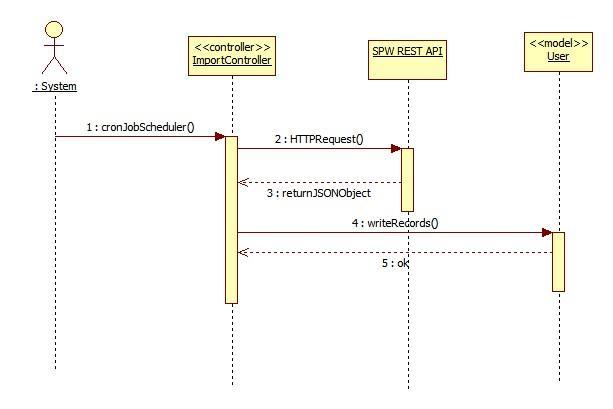
**Figure 9.5.24 MP- Escalate Ticket**



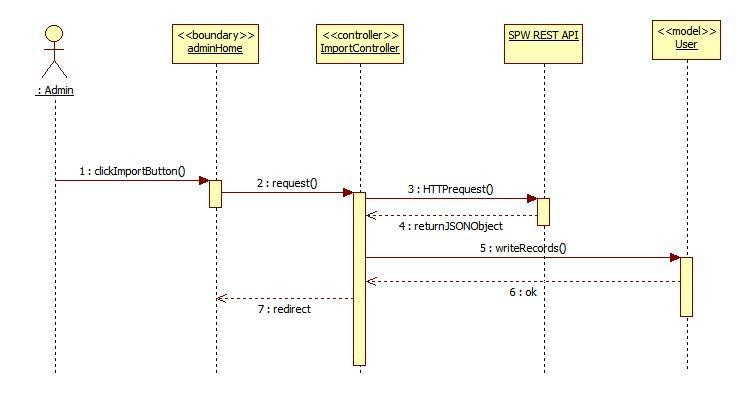
**Figure 9.5.25 MP-Automatically reassign ticket**



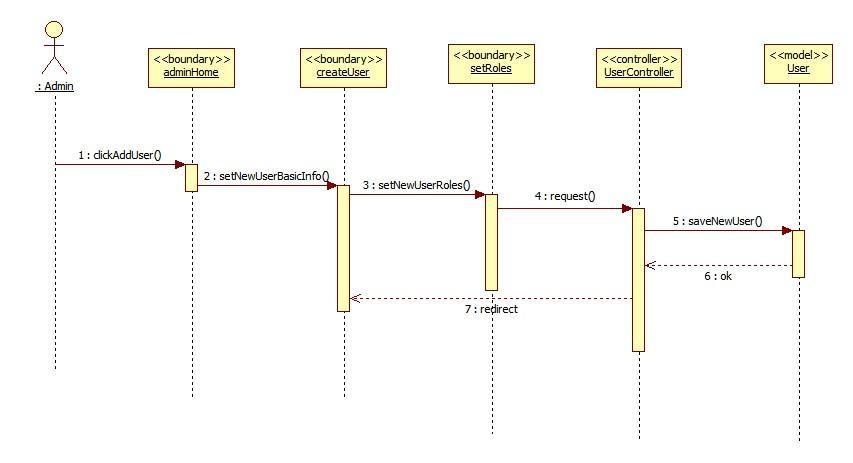
**Figure 9.5.26 MP – Edit Priority**



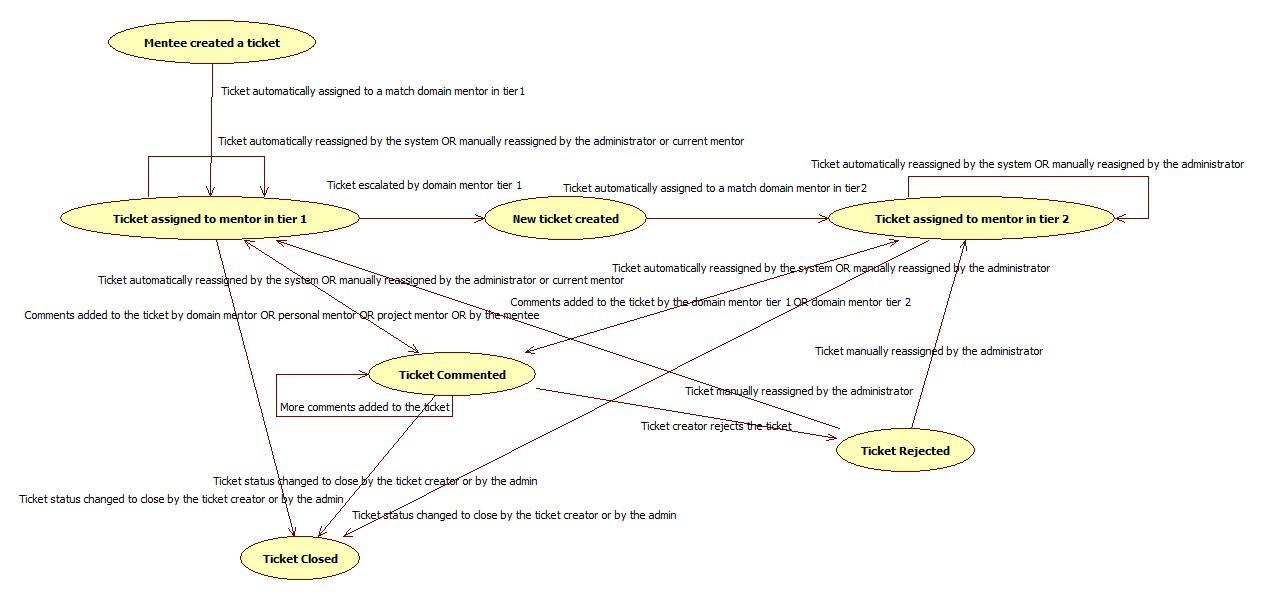
**Figure 9.5.27** **MP-Import data from SPW automatically**



**Figure 9.5.28** **MP-Import data from SPW Manually**



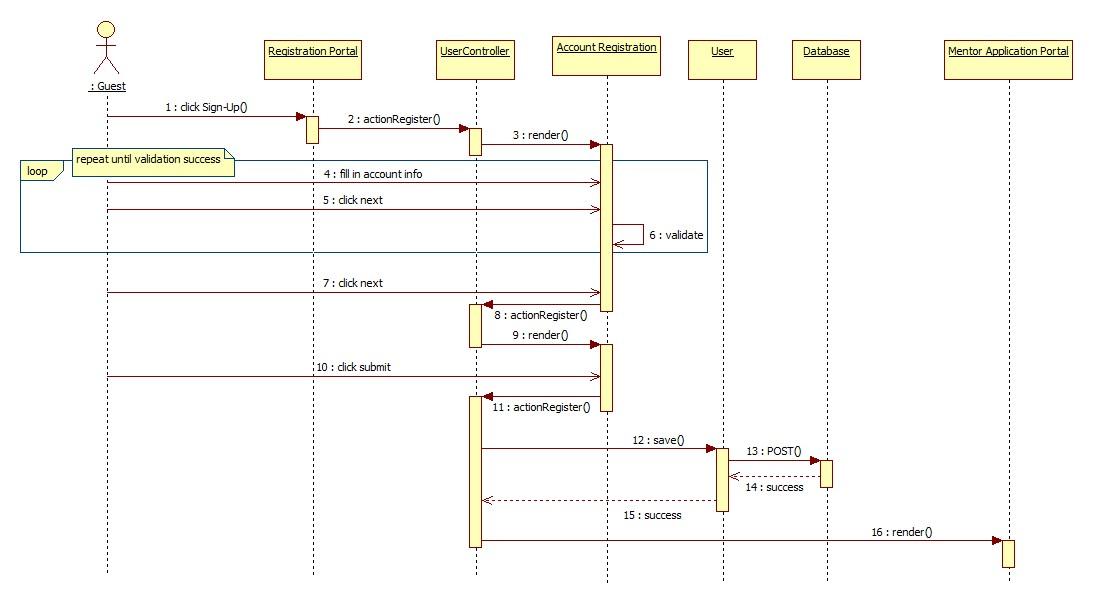
**Figure 9.5.29** **MP-Allow the admin to register mentors**

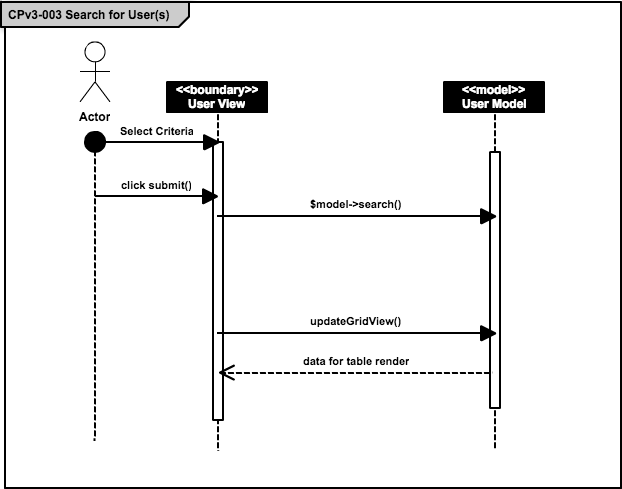
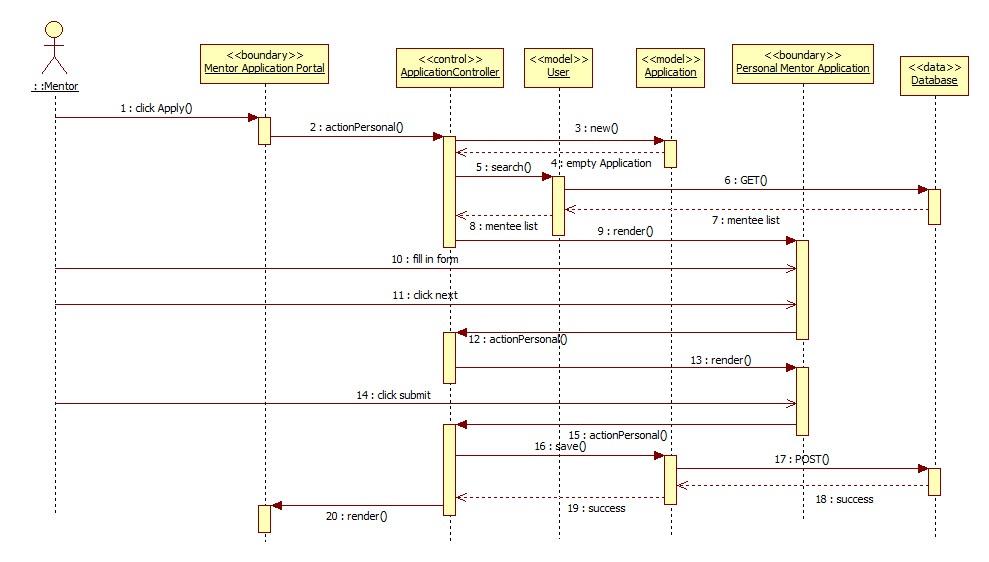


**Figure 9.5.30 Workflow (Ticket)**

**Mentoring Module: Version 3 - Sequence Diagrams**

**CPv3-001 Self Registration for Collaborative Platform**



**CPv3-002 Apply for Personal Mentorship**

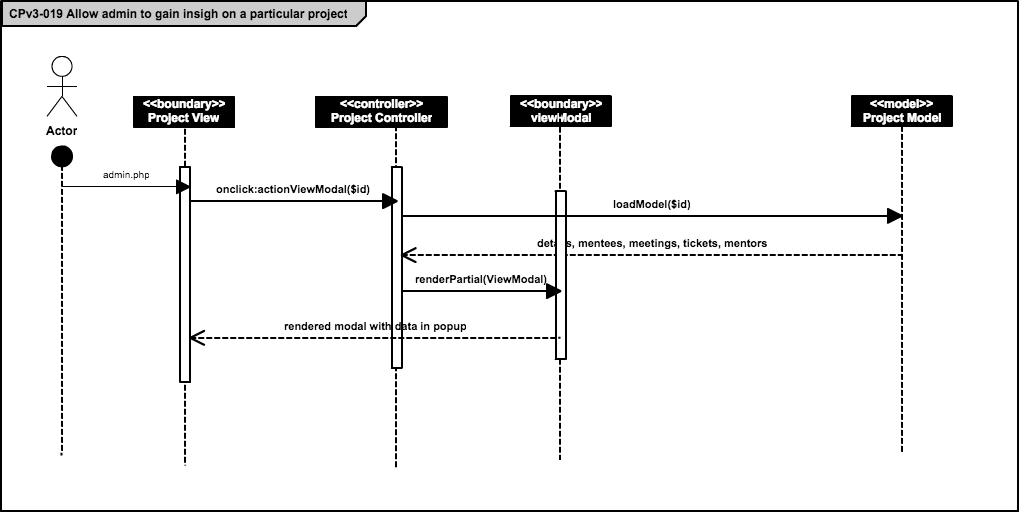
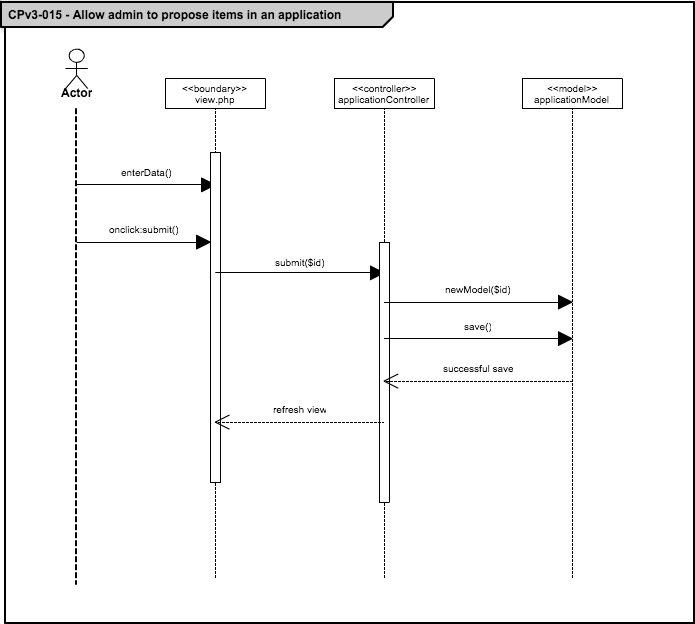
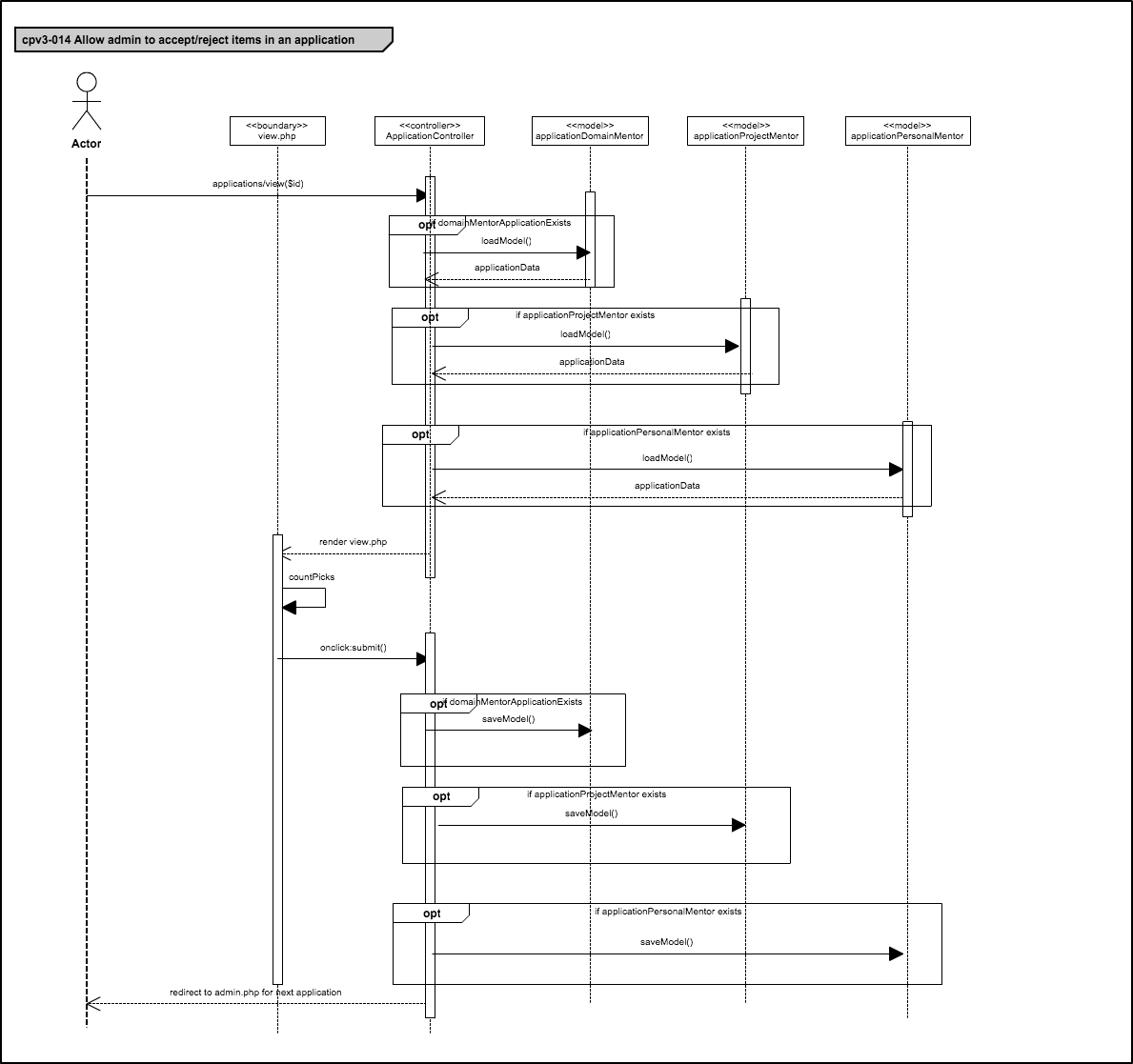
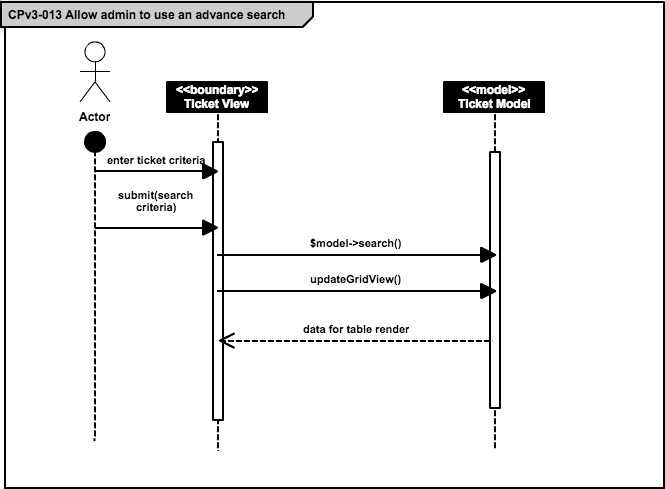
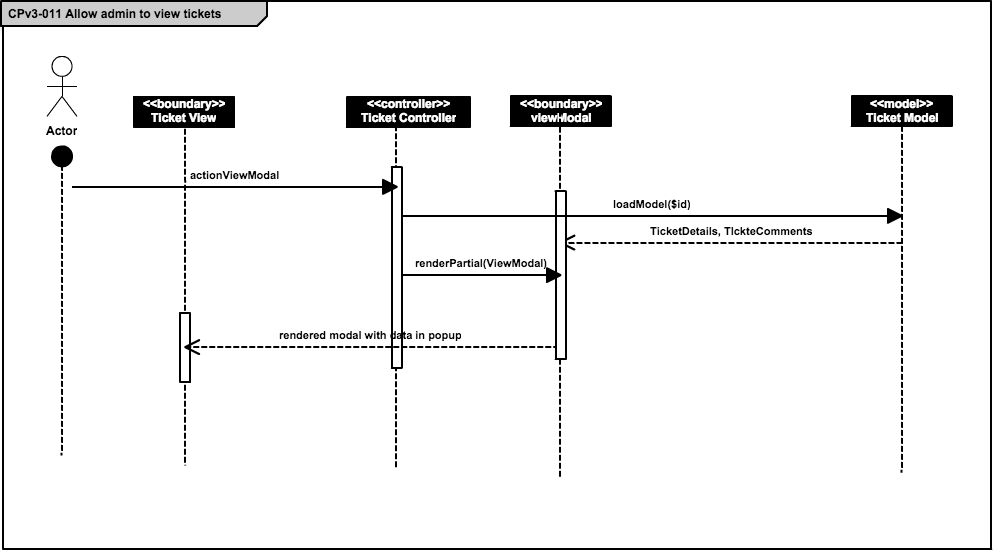
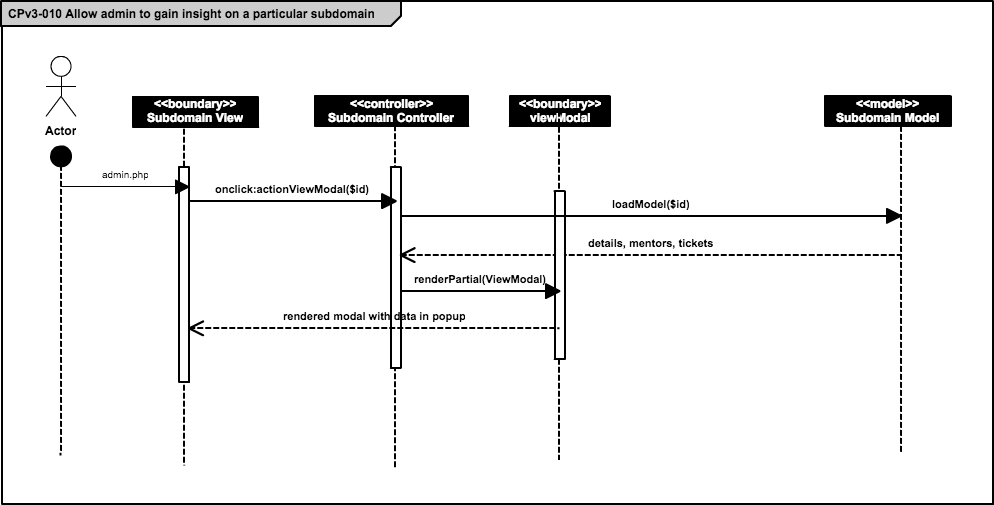
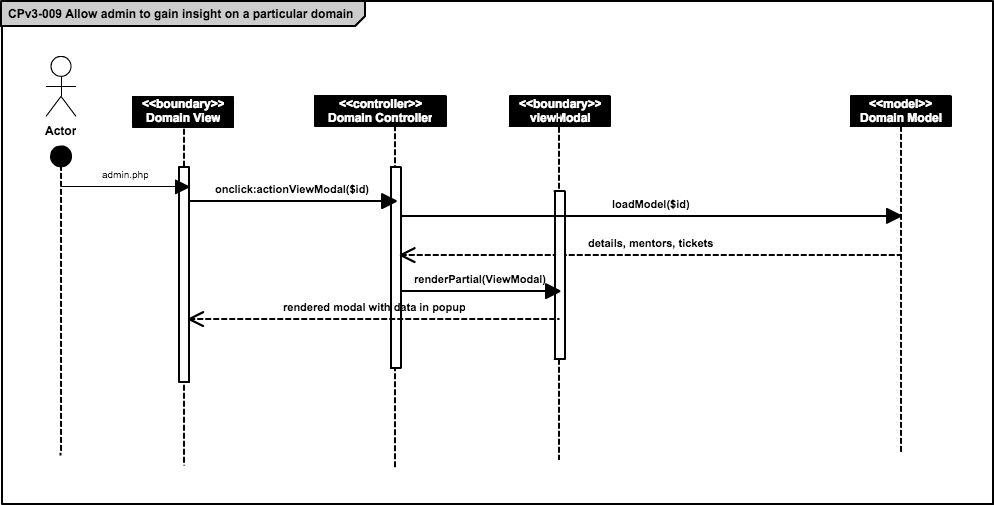
## CPv3-004 Apply for Project Mentorship

## 

## cpv3-005.png

## CPv3-006 Apply for Domain Mentorship

## Cpv3-007.png



## 

## 

## 9.6 Appendix F - Documented Class interfaces and constraints

**Registration and Access Point Subsystem**

Method Signatures SiteController.php

**public function actionLogin()**

@precondition: Valid input has been entered

@postcondition: User is created in the DB.

**public function actionLogout()**

@precondition: User is logged in

@postcondition: User is logged out of the system

**public function actionForgotPassword()**

@precondition: User is on screen and enters valid email

@postcondition: Email sent with password

Method Signatures UserController.php

**public function actionCreate()**

@precondition Valid inputs have been input

@postcondition Email sent with verification for new user

**Communication and Notification Subsystems**

Method Signatures MessageController.php

**public function actionIndex($target = null)**

@precondition: a messages has being selected

@post condition: message is displayed

**public function actionSend($username = null, $reply=null, $selfReply = null)**

@precondition: To field and Subject has being entered

@post condition: message is send

**public function actionGetInbox()**

@precondition: inbox link has being selected

@post condition: if there are inbox messages they are displayed

**public function actionGetMessage($id)**

@precondition: message id is valid

@post condition: message is diplayed

**public function actionGetSent()**

@precondition: sent link has being selected

@post condition: sent messages are displayed

**public function actionGetTrash()**

@precondition: trash link has being selected

@post condition: trash messages are displayed

**public function actionSetAsRead($id)**

@precondition: message has being opened

@post condition: message label change from bold to normal

**public function actionSentToTrash()**

@precondition: message has being deleted

@post condition: message is sent to trash

**public function actionDeleteMessage()**

@precondition: message has being selected

@post condition: message is sent to trash

**public function actionDeleteMessages()**

@precondition: messages has being selected

@post condition: messages are sent to trash

**public function actionAutoComplete()**

@precondition: text is entered in the auto complete field

@post condition: matching data is dsiplayed

**private function getReceivers($string)**

@precondition: require parameters is of the right data type

@post condition: array of receivers is return

Method Signatures User.php

**public static function sendVerificationEmail($userfullName, $user\_email, $adminfullName, $admin\_email)**

@precondition: user register on the system

@post condition: verification email is sent to all the administrators

**public static function sendRejectionAlertToAdmin($ticket\_id, $userfullName, $user\_email, $adminfullName, $admin\_email)**

@precondition: ticket has being rejected by not admin user

@post condition: notification is sent to all the system administrators

**public static function sendEmailPasswordChanged($user\_id)**

@precondition: user has changed his/her password

@post condition: notification is sent to the user

**public static function sendEmailWithNewPassword($username, $password)**

@precondition: forgot password action was executed

@post condition: notification with new password is sent

**public static function sendNewAdministratorEmailNotification($receiver\_email, $password)**

@precondition: new administrator has being created on the system

@post condition: notification is sent to the new administrator

**public static function sendTicketAssignedEmailNotification($creator\_id, $assign\_id, $ticket\_domain)**

@precondition: ticket has being assigned

@post condition: notification is sent to the user to whom the ticket was assigned

**public static function sendTicketCommentedEmailNotification($ticket\_id)**

@precondition: comment has being added to a ticket

@post condition: notification sent to all parties involve

**public static function sendInvitationEmail($invitation)**

@precondition: invitation has being created and send

@post condition: email is sent to invitee

**public static function addNewMessageNotification($sender, $receiver, $link, $level)**

@precondition: message has being sent

@post condition: receiver is notified

**public static function sendTicketStatusCommentedEmailNotification($ticket\_id, $description, $done\_by)**

@precondition: comment has being added to a ticket

@post condition: alert is sent to the user related to the ticket

**public static function sendStatusReassignedEmailNotificationToOldMentor($ticket\_id, $prev\_mentor, $done\_by)**

@precondition: ticket has being reassigned

@post condition: email is sent out to the parties involved

**public static function sendMeetingNotification($project\_mentor\_user\_id, $mentee\_user\_id, $date, $time)**

@precondition: meeting has been scheduled

@post condition: email is sent out to the participants

**Mentoring Subsystem**

Method Signatures TicketController.php

**public function actionCreate()**

@precondition: inputs has been entered as required

@postcondition: create a Ticket in the mentoring system

**public function actionView()**

@precondition: Tickets are already created in the system

@postcondition: retrieve all the details ticket by specific condition

**public function actionReassign ()**

@precondition: ticket has been assign to a user

@postcondition: a new mentor is assign to a ticket

**public function actionChange()**

@precondition: The user has been selected either close or reject as status of the ticket

@postcondition: The status of the ticket is changed

**public function actionUploadFile()**

@precondition: The user has selected attach file to a ticket option

@postcondition: an image will have been uploaded to the server

**public function actionDownloadFile ()**

@precondition: A file has been uploaded to a specific ticket

@postcondition: the file is downloaded to the client computer.

**public function actionUploadFile()**

@precondition: The user has selected attach file to a ticket option

@postcondition: an image will have been uploaded to the server

Method Signatures User.php

**public function assignTicket()**

@precondition: create ticket request has been submitted. Either with specific user or not

@postcondition: The system assign user to a ticket according to the specifications.

Method Signatures CommentController.php

**public function actionCreate()**

@precondition: the user type a comment to a ticket

@postcondition: a comment is saved in the database

**public function actionMessage()**

@precondition: The user has change the status of a ticket and leave a message

@postcondition: the message is sent and added to a ticket

**public function actionView()**

@precondition: the user chose a specific ticket

@postcondition: all the comments related to the ticket has been render to the view

Method Signatures ProjectMeetingController.php

**public function actionCreate()**

@precondition: the project mentor has request a meeting with his/her mentee

@postcondition: the meeting is saved in the server.

**public function actionadminViewMeetings()**

@precondition: the administrator has request to view all the meetings in the system

@postcondition: the administrador overview all the meetings created by the projects mentors.

**public function actionpMentorViewMeetings()**

@precondition: the project mentor clicks in the button project mentor

@postcondition: the system render all the meetings associated to the project mentor.

**public function actionpMenteeViewMeetings()**

@precondition: the mentee clicks in the button project mentor

@postcondition: the system render all the meetings associated to the mentee.

**Mentoring Module: Version 3 - Documented Class Interfaces**

Method Signatures for ApplicationController.php

**public function actionPortal()**

@precondition: the Mentor goes to the Mentor Application Portal

@postcondition: the system renders the Mentor Application Portal

**public function actionPersonal()**

@precondition: the Mentor clicks the Apply for Personal Mentorship button

@postcondition:  
 @event(Initial load): the system renders the Personal Mentor Application

@event(submit): the system saves Personal application details

**public function actionProject()**

@precondition: the Mentor clicks the Apply for Project Mentorship button

@postcondition:  
 @event(Initial load): the system renders the Project Mentor Application

@event(submit): the system saves Project application details

**public function actionDomain()**

@precondition: the Mentor clicks the Apply for Domain Mentorship button

@postcondition:  
 @event(Initial load): the system renders the Domain Mentor Application

@event(submit): the system saves Domain application details

Method Signatures for User.php

**public function getFullName()**

@precondition: a Controller has requested for a User’s full name

@postcondition: the system returns the User’s full name to the Controller

**public returnUsersForApp($dataprovider)**

@precondition: the Personal Mentor Application is being prepared

@postcondition: the system returns a JSON object with all required User details

**public function getFullName()**

@precondition: a Controller has requested for a User’s full name

@postcondition: the system returns the User’s full name to the Controller

**public function getUniversityName()**

@precondition: a Controller has requested for the name of a User’s university

@postcondition: the system returns the Universit name to the Controller

**public function getPic()**

@precondition: a Controller has requested for a User’s avatar

@postcondition: the system returns the User’s avatar to the Controller

Method Signatures for UserController.php

**public function actionCreate()**

@precondition: the Mentor clicks the Sign Up button for account registration

@postcondition:  
 @event(Initial load): the system renders the Account Registration form

@event(submit): the system saves User’s account details

Method Signatures for Project.php

**public function getProjectMentor()**

@precondition: a Controller has requested for the User that mentors this Project

@postcondition: the system returns the User to the Controller

**public getProjectsForApp($dataprovider)**

@precondition: the Project Mentor Application is being prepared

@postcondition: the system returns a JSON object with all required Project details

**public function getDscriptionOfSize($size)**

@precondition: a Controller has requested the project description of a specific size

@postcondition: the system returns the Description trimmed to the size

**public function getCustomerFullName()**

@precondition: a Controller has requested for the full name of the Project customer

@postcondition: the system returns the customers full name to the Controller

Method Signatures for Domain.php

**public getDomainsForApp($dataprovider)**

@precondition: the Domain Mentor Application is being prepared

@postcondition: the system returns a JSON object with all required Domain details

Method Signatures for Subdomain.php

**public getSubdomainsForApp($dataprovider)**

@precondition: the Domain Mentor Application is being prepared

@postcondition: the system returns a JSON object with all required Subdomain details

Method Signatures for University.php

**public getUniversities($dataprovider)**

@precondition: a Controller has requested for a list of all Universities

@postcondition: the system returns a list of JSON objects with University details

**public function universityByName($name)**

@precondition: a Controller has requested for a Univeristy object but only has a name

@postcondition: the system returns the University object to the Controller

**public function universityById($id)**

@precondition: a Controller has requested for a Univeristy object but only has an id

@postcondition: the system returns the University object to the Controller

## 9.7 Appendix G – Documented code for test drivers and stubs

Following are two codes test cases for the system. The complete testing code can be found in the testing folder along with the software solution.

// MP-TC39

// Test schedule Meeting

@Test

**public** **void** testScheduleMeeting() {

**try** {

// go to login page

driver.get("http://vjf-sp14.cs.fiu.edu/coplat/index.php/site/login");

delay.Del(2000);

driver.findElement(By.*id*("LoginForm\_username")).sendKeys("lsanc104");

driver.findElement(By.*id*("LoginForm\_password")).sendKeys("lsanc104");

delay.Del(3000);

driver.findElement(By.*className*("btn")).click();

delay.Del(2000);

// go to user

driver.get("http://vjf-sp14.cs.fiu.edu/coplat/index.php/home/adminHome");

delay.Del(2000);

// click Set-Up Meeting

driver.findElement(By.*xpath*("//\*[@id='contactlinks']/div/a[1]")).click();

delay.Del(2000);

// enter date

driver.findElement(By.*xpath*("//\*[@id='Meeting\_date']")).click();

driver.findElement(By.*xpath*("//\*[@id=Meeting\_date']")).sendKeys("time");

delay.Del(2000);

List <WebElement> dates = driver.findElements(By.*xpath*("//\*[@class='ui-datepicker-calendar']//\*[@href='#']"));

delay.Del(1000);

dates.get(29).click();

delay.Del(1000);

driver.findElement(By.*id*("Meeting\_time")).sendKeys("3:28am");

delay.Del(2000);

driver.findElement(By.*xpath*("//\*[@class='modal-footer']/button")).click();

*assertTrue*(driver.getCurrentUrl().contains("lsanc104@fiu.edu"));

//driver.findElement(By.xpath("//\*[@id='yw0']/li[5]/a")).click(); // logout

} **catch** (Exception e) {

e.printStackTrace();

}

}

// MP-TC40

// login succesfully

@Test

**public** **void** testLogin() {

**try** {

// go to login page

driver.get("http://vjf-sp14.cs.fiu.edu/coplat/index.php/site/login");

delay.Del(2000);

driver.findElement(By.*xpath*("//\*[@id='altlogin']/a[3]")).click();

delay.Del(3000);

// enter login and click login

driver.findElement(By.*xpath*("//\*[@name='session\_key']")).sendKeys("lsanc104");

driver.findElement(By.*xpath*("//\*[@name='session\_password']")).sendKeys("xxxxxxxx");

driver.findElement(By.*xpath*("//\*[@name='authorize']")).click();

delay.Del(2000);

// user is redirected to userHome page

*assertTrue*(driver.getCurrentUrl().contains("home/userHome"));

driver.findElement(By.*xpath*("//\*[@id='yw0']/li[5]/a")).click(); // logout

} **catch** (Exception e) {

e.printStackTrace();

}}

**Mentoring Module: Version 3**

Due to time constraints and constantly added requirements we were unable to perform any Unit testing using test drivers or stubs. All testing was done by reenacting scenarios and recording the results. We acknowledge that this is by no means ideal but it was the only way we could keep up with our clients expectations of a “finished” product.

## 9.8 Appendix H – Diary of meetings and tasks for the entire semester

|  |  |
| --- | --- |
| **Feasibility Criteria** | **Description** |
| Operational feasibility | The proposed system will solve all the issues encountered during the system requirements process. So, the new system is operationally feasible. |
| Technical feasibility | The collaborative platform is a system that it will be developed for the computer science department. So, all the resources needed to develop the system are available for the team. |
| Schedule feasibility | The project is very well spread out according to the academic calendar of spring 2014. Team members and our mentor agree that the amounts of the requirements are enough to be completed within the given timeframe. |
| Economic feasibility | The team will not spend money to develop the project, because it is not required to pay for software (since it is open source) and hardware (supplied through personal computer or FIU Lab). |

## 4.3 Appendix C – Cost Matrix

|  |  |
| --- | --- |
| **Item** | **Item Cost** |
| Hardware (Personal Laptops) | $ 0.00 |
| Hardware (FIU Lab) | $ 0.00 |
| Software (Open Source) | $ 0.00 |
| Total Cost | $ 0.00 |

## 4.4 Appendix D – Diary of Meetings

|  |  |
| --- | --- |
| **Date:** | 1/17/14 |
| **Duration:** | 1 hour and a half |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria  Prof. Caraballo |
| **Summary of discussion:** | Discussing the requirements and features of the system purpose |

|  |  |
| --- | --- |
| **Date:** | 1/18/14 |
| **Duration:** | 3 hours |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria |
| **Summary of discussion:** | Discuss features and brainstorm the problem and possible solutions. |

|  |  |
| --- | --- |
| **Date:** | 1/22/14 |
| **Duration:** | 2 hours |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria, Juan Caraballo |
| **Summary of discussion:** | Review the first draft. Discussing the project problem and system solution. |

|  |  |
| --- | --- |
| **Date:** | 1/23/14 |
| **Duration:** | 2 hours |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria |
| **Summary of discussion:** | Work on the feasibility report based on the feedback given by Juan. |

|  |  |
| --- | --- |
| **Date:** | 1/24/14 |
| **Duration:** | 1 hours |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria, Juan Caraballo |
| **Summary of discussion:** | Prof. Caraballo provided feedback on our first feasibility & project plan draft |

|  |  |
| --- | --- |
| **Date:** | 1/25/14 |
| **Duration:** | 4 hours |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria |
| **Summary of discussion:** | Work on the final feasibility & project plan document.  Assign specific functionalities of the propose system to each team member.  Work on the first presentation. |

|  |  |
| --- | --- |
| **Date:** | 1/29/14 |
| **Duration:** | 1 hour |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Juan Caraballo |
| **Summary of discussion:** | Redefined functionality in order to create breakdown of team assignments. |

|  |  |
| --- | --- |
| **Date:** | 1/30/14 |
| **Duration:** | 1 hour |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria, Juan Caraballo |
| **Summary of discussion:** | Break down workload on individual basis; discuss improvements for presentations on Monday. |

|  |  |
| --- | --- |
| **Date:** | 2/1/2014 |
| **Duration:** | 4 hours |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria |
| **Summary of discussion:** | Finalize FSPP, begin work on use cases for RD.  Re-assign specific functionalities of the propose system to each team member.  Finalize slides for presentation. |

|  |  |
| --- | --- |
| **Date:** | 2/19/14 |
| **Duration:** | 1 hours (Conference Call) |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria, Prof Juan Caraballo |
| **Summary of discussion:** | Agenda:   * Feedback on scenarios and Mockups |

|  |  |
| --- | --- |
| **Date:** | 02/21/14 |
| **Duration:** | 1 hour (Conference Call) |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria  Prof. Caraballo. |
| **Summary of discussion:** | Agenda:   * More Feedback on scenarios and Mock Ups. |

|  |  |
| --- | --- |
| **Date:** | 02/22/14 |
| **Duration:** | 4 hour |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria. |
| **Summary of discussion:** | Agenda:   * Work on Mock Ups, Scenarios, Functional Requirements. * Minimal Class Diagram. * ER Model |

|  |  |
| --- | --- |
| **Date:** | 02/24/14 |
| **Duration:** | 3 hours |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria, Prof. Juan Caraballo |
| **Summary of discussion:** | Agenda:   * Feedback on Scenarios and mockups |

|  |  |
| --- | --- |
| **Date:** | 02/27/14 |
| **Duration:** | 1 hours (Conference Call) |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia |
| **Summary of discussion:** | Agenda:   * More Feedback on Mock Ups * Feedback on Functional Requirements |

|  |  |
| --- | --- |
| **Date:** | 03/01/2014 |
| **Duration:** | 6 hours |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria. |
| **Summary of discussion:** | Agenda:   * Detailed Design * Work on design document and UML diagrams |

|  |  |
| --- | --- |
| **Date:** | 03/03/2014 |
| **Duration:** | 1 hour |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria, Prof. Caraballo |
| **Summary of discussion:** | Agenda:   * More Feedback on Mock Ups |

|  |  |
| --- | --- |
| **Date:** | 2/19/14 |
| **Duration:** | 1 hours (Conference Call) |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria, Prof Juan Caraballo |
| **Summary of discussion:** | Agenda:   * Feedback on scenarios and Mockups |

|  |  |
| --- | --- |
| **Date:** | 02/21/14 |
| **Duration:** | 1 hour (Conference Call) |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria  Prof. Caraballo. |
| **Summary of discussion:** | Agenda:   * More Feedback on scenarios and Mock Ups. |

|  |  |
| --- | --- |
| **Date:** | 02/22/14 |
| **Duration:** | 4 hour |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria. |
| **Summary of discussion:** | Agenda:   * Work on Mock Ups, Scenarios, Functional Requirements. * Minimal Class Diagram. * ER Model |

|  |  |
| --- | --- |
| **Date:** | 02/24/14 |
| **Duration:** | 3 hours |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria, Prof. Juan Caraballo |
| **Summary of discussion:** | Agenda:   * Feedback on Scenarios and mockups |

|  |  |
| --- | --- |
| **Date:** | 02/27/14 |
| **Duration:** | 1 hours (Conference Call) |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia |
| **Summary of discussion:** | Agenda:   * More Feedback on Mock Ups * Feedback on Functional Requirements |

|  |  |
| --- | --- |
| **Date:** | 03/01/2014 |
| **Duration:** | 6 hours |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria. |
| **Summary of discussion:** | Agenda:   * Detailed Design * Work on design document and UML diagrams |

|  |  |
| --- | --- |
| **Date:** | 03/03/2014 |
| **Duration:** | 1 hour |
| **Participants:** | Lorenzo Sanchez, Anibal Sicilia, Henry Muniz, Steven Sanabria, Prof. Caraballo |
| **Summary of discussion:** | Agenda:   * More Feedback on Mock Ups |

|  |  |
| --- | --- |
| **Date:** | 03/05/2014 |
| **Duration:** | 3 hour |
| **Participants:** | Lorenzo Sanchez, Henry Muniz, Steven Sanabria |
| **Summary of discussion:** | Agenda:   * Work on the implementation |

|  |  |
| --- | --- |
| **Date:** | 03/12/2014 |
| **Duration:** | 3 hour |
| **Participants:** | Lorenzo Sanchez, Henry Muniz, Steven Sanabria |
| **Summary of discussion:** | Agenda:   * Work on the implementation * Test the subsystem |

|  |  |
| --- | --- |
| **Date:** | 03/19/2014 |
| **Duration:** | 3 hour |
| **Participants:** | Lorenzo Sanchez, Henry Muniz, Steven Sanabria |
| **Summary of discussion:** | Agenda:   * Work on the implementation * Test the subsystem * Unit Test |

|  |  |
| --- | --- |
| **Date:** | 03/20/2014 |
| **Duration:** | 3 hour |
| **Participants:** | Lorenzo Sanchez, Henry Muniz, Steven Sanabria |
| **Summary of discussion:** | Agenda:   * Work on the implementation * Test the subsystem * Unit Test |

**Summer 2014**

|  |  |
| --- | --- |
| **Date:** | 5/15/14 2:40 pm |
| **Duration:** | 1 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez, Mohammed HAlbukhari, Jesus Jordan.  Prof. Caraballo |
| **Summary of discussion:** | To review the documentation as team. |

|  |  |
| --- | --- |
| **Date:** | 5/15/14 8:00 pm |
| **Duration:** | 1 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez, Mohammed HAlbukhari, Jesus Jordan.  Prof. Caraballo |
| **Summary of discussion:** | To review the system requirements. |

|  |  |
| --- | --- |
| **Date:** | 5/17/14 |
| **Duration:** | 12 hour |
| **Participants:** | Jonathan Santiago,Ramon Gomez, Mohammed HAlbukhari |
| **Summary of discussion:** | Set up the system |

|  |  |
| --- | --- |
| **Date:** | 5/15/14 8:00 pm |
| **Duration:** | 1 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez, Mohammed HAlbukhari, Jesus Jordan.  Prof. Caraballo |
| **Summary of discussion:** | To review the system requirements. |

|  |  |
| --- | --- |
| **Date:** | 5/20/14 |
| **Duration:** | 2 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez, Mohammed HAlbukhari |
| **Summary of discussion:** | Discuss the requirement within the members of the team |

|  |  |
| --- | --- |
| **Date:** | 5/22/14 |
| **Duration:** | 2 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez, Mohammed HAlbukhari, Jesus Jordan.  Prof.Sadjadi |
| **Summary of discussion:** | To clarify the system requirements. |

|  |  |
| --- | --- |
| **Date:** | 5/30/14 |
| **Duration:** | 1 hour and a half |
| **Participants:** | Ramon Gomez, Jesus Jordan.  Prof. Caraballo |
| **Summary of discussion:** | Basic scenarios in ticket workflow. |

|  |  |
| --- | --- |
| **Date:** | 5/31/14 |
| **Duration:** | 12 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez. |
| **Summary of discussion:** | To Set GitLab. |

|  |  |
| --- | --- |
| **Date:** | 6/07/14 8:00 pm |
| **Duration:** | 4 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez. |
| **Summary of discussion:** | To review the scheduling mechanism. |

|  |  |
| --- | --- |
| **Date:** | 6/16/14 8:00 pm |
| **Duration:** | 1 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez.  Prof. Caraballo |
| **Summary of discussion:** | Ticket escalation process. |

|  |  |
| --- | --- |
| **Date:** | 6/24/14 8:00 pm |
| **Duration:** | 1 hour and a half |
| **Participants:** | Jonathan Santiago.  Prof. Caraballo |
| **Summary of discussion:** | To review the registration scenarios. |

|  |  |
| --- | --- |
| **Date:** | 7/5/14 |
| **Duration:** | 6 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez. |
| **Summary of discussion:** | Documentation |

|  |  |
| --- | --- |
| **Date:** | 7/8/14 8:00 pm |
| **Duration:** | 1 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez.  Prof. Caraballo |
| **Summary of discussion:** | To discuss user interface. |

|  |  |
| --- | --- |
| **Date:** | 7/12/14 |
| **Duration:** | 6 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez. |
| **Summary of discussion:** | Documentation |

|  |  |
| --- | --- |
| **Date:** | 7/19/14 |
| **Duration:** | 8 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez. |
| **Summary of discussion:** | Documentation |

|  |  |
| --- | --- |
| **Date:** | 7/20/14 |
| **Duration:** | 8 hour and a half |
| **Participants:** | Jonathan Santiago,Ramon Gomez. |
| **Summary of discussion:** | Documentation |

**Mentoring Module: Version 3**

|  |  |
| --- | --- |
| **Date** | 09/03/2014 |
| **Location** | Telephone Conference |
| **Start** | 3:00 PM |
| **End** | 4:30 PM |
| **In Attendance** | Juan Caraballo, Nicholas Madariaga, Jonathan Sanchez |
| **Late** | N/A |
| **Agenda** | Get background information about the Collaborative Platform and gather specifications and requirements for the next iteration of the site. |
| **Summary of Discussion** | Discussed purpose, objective, goals, and future of the site as well as what specific requirements Juan is looking to get done for the next iteration. Juan advised on best ways to communicate with him and gave tips on the project itself. Primary goals for site at this time is to complete the Registration Process for a Mentor and once that is done we can focus on other areas like makeover of the site, e.g. Admin dashboard, email invite by admin to mentor, tweaking registration process for added function. These other features will be completed as time allows. |
| **Assigned Tasks** | Nick and Jonathan:   * Verify GitHub accounts * Create Google Drive for Docs * Create Trello account * Get the local version of the Collaborative Platform working |

|  |  |
| --- | --- |
| **Date** | 09/07/2014 |
| **Location** | Jonathan’s House |
| **Start** | 12:00 PM |
| **End** | 5:30 PM |
| **In Attendance** | Nicholas Madariaga, Jonathan Sanchez |
| **Late** | N/A |
| **Agenda** | Brainstorm the look of the site and come up with mockups. |
| **Summary of Discussion** | next meeting: 09/08 @ 9AM |
| **Assigned Tasks** | Nick and Jonathan:   * setup git on local system * type notes |

|  |  |
| --- | --- |
| **Date** | 09/08/2014 |
| **Location** | ECS 237 |
| **Start** | 9:00 AM |
| **End** | 3:00 PM |
| **In Attendance** | Nicholas Madariaga, Jonathan Sanchez |
| **Late** | N/A |
| **Agenda** | Begin Feasibility and Requirements Documents to get a rough draft version. |
| **Summary of Discussion** | Go over mockups for features we are implementing.  Rough Draft - Feasibility document  Rought Draft - Requirements document |
| **Assigned Tasks** | * update trello with mockups * update trello with rough draft of docs |

|  |  |
| --- | --- |
| **Date** | 09/10/2014 |
| **Location** | Telephone Conference |
| **Start** | 12:00 PM |
| **End** | 1:45 PM |
| **In Attendance** | Juan Caraballo, Nicholas Madariaga, Jonathan Sanchez |
| **Late** | N/A |
| **Agenda** | Get juan to sign up to trello to review mockups |
| **Summary of Discussion** | Overview of admin dashboard and suggested features:   * insight on domain knowledge and how many mentors are assigned * insight on tickets (pending, avg ticket response.. etc) * insight on specific mentor stats (assigned tickets, longest response, total mentors and breakdown) * insight on project or student (project meeting, breakdown of pending tickets by project name) * pending mentor approvals * active mentors in current month vs last month   Overview of Mentor Registration and suggested features:   * Mentor registration is all about workflow   1. Mentor receives invitation   2. Mentor creates account with coplat   3. Mentor applies for mentorships of his choice   4. The application is sent out for review   5. Admin approves application   6. Mentor confirms or denies admins changes * Give option to sign up via LinkedIn * Admins should see all applications from any one mentor at once   The mockups have no context for juan so he gave us examples on how to do that for each feature we’re working with.  Mentor Dashboard   * tabled for implementation after self-registration, admin management panel, and back and forth of approvals of roles between admin and mentor   Mentor Application Response from Admin.   * what will the mockup look like for mentor once applciation returns from admin |
| **Assigned Tasks** | * Create scenarios for mockups so they can have context and be ready to present that for next meeting - tomorrow 8:30AM |

|  |  |
| --- | --- |
| **Date** | 09/11/2014 |
| **Location** | Video Conference via Google Hangouts |
| **Start** | 8:30 AM |
| **End** | 11:20 AM |
| **In Attendance** | Juan Caraballo, Nicholas Madariaga, Jonathan Sanchez |
| **Late** | N/A |
| **Agenda** | Go over mockups alongside workflow |
| **Summary of Discussion** | The plan is to have two seperate entry points for logging in:   1. For users with existing accounts 2. For new users   Review of Mentor Registration mockups:   * Account Registration   + From here let users know how long their applications could potentially take * Mentor Application Hub   + Mixed with account registration or not? * Personal Mentor Application   + Needs re-arranging and add name hover * Project Mentor Application   + Add criteria to automation   + Make similar to personal mentor * Domain Mentor Application   + Display Domains in need of help   + Add how many questions per month * All Applications need some way of cancelling/going back/redo * Mockup for deny/approval email   Review of Admin dashboard mockups: |
| **Assigned Tasks** | Update and add mockups based on what was discussed during this meeting so we can run through them again during the next meeting. |

|  |  |
| --- | --- |
| **Date** | 09/12/2014 |
| **Location** | Video Conference via Google Hangouts |
| **Start** | 10:00 AM |
| **End** | 11:45 AM |
| **In Attendance** | Juan Caraballo, Nicholas Madariaga, Jonathan Sanchez |
| **Late** | N/A |
| **Agenda** | Go over mockups alongside workflow round 2 |
| **Summary of Discussion** |  |
| **Assigned Tasks** |  |

|  |  |
| --- | --- |
| **Date** | 09/14/2014 |
| **Location** | Jonathan’s House |
| **Start** | 1:30 PM |
| **End** | 3:30 PM |
| **In Attendance** | Nicholas Madariaga, Jonathan Sanchez |
| **Late** | N/A |
| **Agenda** | Dissect site, mockups, update trello. |
| **Summary of Discussion** | Continue to dissect site and try to understand the flow of information through the pages. |
| **Assigned Tasks** |  |

|  |  |
| --- | --- |
| **Date** | 09/16/2014 |
| **Location** | Video Conference via Google Hangouts |
| **Start** | 11:00 AM |
| **End** | 12:15 PM |
| **In Attendance** | Juan Caraballo, Nicholas Madariaga, Jonathan Sanchez |
| **Late** | N/A |
| **Agenda** | Review mockup revisions |
| **Summary of Discussion** | Additional changes suggested for mockups. Meeting tomorrow to review those changes as well. |
| **Assigned Tasks** | Revise Mockups |

|  |  |
| --- | --- |
| **Date** | 09/17/2014 |
| **Location** | Video Conference via Google Hangouts |
| **Start** | 3:30 PM |
| **End** | 4:30 PM |
| **In Attendance** | Juan Caraballo, Nicholas Madariaga, Jonathan Sanchez |
| **Late** | N/A |
| **Agenda** | Review mockups |
| **Summary of Discussion** | Mockups looked good but minor changes requested. Upload to trello in a way that the reader can see transitions and follow along. Get together and come up with final universal approval panel. |
| **Assigned Tasks** | Make the minor changes and get (Nick and Jonathan) get together to sort out the details for the universal approval panel. Meeting set for Thursday @ 12 PM. |

|  |  |
| --- | --- |
| **Date** | 09/18/2014 |
| **Location** | Video Conference via Google Hangouts |
| **Start** | 12:00 PM |
| **End** | 1:30 PM |
| **In Attendance** | Nicholas Madariaga, Jonathan Sanchez |
| **Late** | N/A |
| **Agenda** | Sort out the GUI for the universal approval panel |
| **Summary of Discussion** | We combined both mockups to create one universal approval panel. We also setup cp-dev.cs.fiu.edu and cp.cs.fiu.edu with the project and git. |
| **Assigned Tasks** | Upload mockups onto trello for mentor approval. Work on getting more knowledge on the technology used onthe website (Yii in particular). |

|  |  |
| --- | --- |
| Date | 09/22/2014 |
| Location | ECS 237 |
| Start | 10:00 AM |
| End | 3:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Prepare presentation material, review introduction of project, continue to work on features. |
| Summary of Discussion | Nick managed to get Register Page up. Jonathan made some changes to the pages and links added buttons on pages. We continued to mess around with Bootstrap and PHP. |
| Assigned Tasks | Continue to do tutorials for Yii and PHP. |

|  |  |
| --- | --- |
| Date | 9/28/14 |
| Location | Nick’s House |
| Start | 1PM |
| End | 7PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Determine what we need to begin implementing our mockups |
| Summary of Discussion | We needed to make a few changes to the database:Added a tier column to the domain\_mentor tableCreated several new tables to store mentor application dataUpdated our local db with new db and cp-dev.cs.fiu.eduWe worked on our git repository:We added a develop branch to keep all changes we have both made and approved.We then made two feature branches to store our individual progress as we goWe finally found a way to ignore files in a way that workedWe created a new model, controller and some views for mentor applications |
| Assigned Tasks | Jonathan - Improve the layout of existing admin viewsNick - Get self serve account creation working |

|  |  |
| --- | --- |
| Date | 09/29/14 |
| Location | ECS237 |
| Start | 9AM |
| End | 2PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Work on Documentation |
| Summary of Discussion | Made revisions to all three documents. |
| Assigned Tasks | Continue working on Use Cases and Scearios |

|  |  |
| --- | --- |
| Date | 10/05/14 |
| Location | Google Hangounts |
| Start | 12PM |
| End | 1PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Discuss implementation action plan |
| Summary of Discussion | We discussed what we had worked on up until before the meeting.We noticed that model relations could be automated through gii.We discussed some problems we were running into. |
| Assigned Tasks | Work on our assigned tasks and reconvene at 8PM to give an update on our progress |

|  |  |
| --- | --- |
| Date | 10/05/14 |
| Location | Google Hangounts |
| Start | 8PM |
| End | 9PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Discuss todays progress |
| Summary of Discussion | We had not accomplished as much as we wanted due to unforeseen complications. We shared any discoveries we made about how the current system actually works and which parts of them can be reused. |
| Assigned Tasks | Continue working on our assigned tasks based on the new information we gathered about how our solution works.Reconvene tomorrow morning for another update and to prepare for our presentation. |

|  |  |
| --- | --- |
| Date | 10/06/14 |
| Location | ECS237 |
| Start | 10:30 AM |
| End | 2:45 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Discuss progress and yii framework and also prepare for todays presentation. |
| Summary of Discussion | Prepare for presentation and continue working on our individual features. We also merged our feature branch into the developer branch and had the dev server pull the features we’ve implemented so far. |
| Assigned Tasks | Continue to work on documentation and features. (Nick - Self Registration initial page), (jonathan - Modal for detailed view).Tomorrow Meeting to advance on individual features - 3:00 PM - 6:00 PM. |

|  |  |
| --- | --- |
| Date | 10/07/14 |
| Location | ECS237 |
| Start | 3:00 PM |
| End | 6:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Discuss progress, Yii research and Coding session. |
| Summary of Discussion | Still struggling with Yii framework. Learned some more about Yii but still not enough to get things working as we like. |
| Assigned Tasks | Keep studying how Yii works and try to code. |

|  |  |
| --- | --- |
| Date | 10/10/14 |
| Location | Jonathans House |
| Start | 12:00 PM |
| End | 4:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Discuss progress, Yii research and Coding session. |
| Summary of Discussion | Gained a better understanding of how Yii handles interactions between its parts. Still struggling with our code, specifically formatting. |
| Assigned Tasks | Keep studying and trying to get things working independently. Try to find Yiisus. |

|  |  |
| --- | --- |
| Date | 10/13/14 |
| Location | ECS237 |
| Start | 1:00 PM |
| End | 3:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Discuss progress, Yii research and Coding session. |
| Summary of Discussion | Discussed our findings. Some progress made but still nothing substantial. Begin panicking. |
| Assigned Tasks | Continue researching and trying to get our functions working. Never give up. |

|  |  |
| --- | --- |
| Date | 10/16/14 |
| Location | ECS237 |
| Start | 8:00 PM |
| End | 11:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Coding Session |
| Summary of Discussion | Major breakthroughs made on both our sections! User registration w/ personal info functional. Tickets advanced search functional. |
| Assigned Tasks | Continue to work on respective features. (Nick - Mentor Application Portal), (jonathan - Modal for detailed view).Plan to reconvene on Sunday 10/19/14 |

|  |  |
| --- | --- |
| Date | 10/19/14 |
| Location | ECS237 |
| Start | 12:30 PM |
| End | 6:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Coding Session |
| Summary of Discussion | Finalized our main features for this cycle. Implemented Mentor Application Portal. Started working on Personal Mentor App. Modals for admin view made functional. |
| Assigned Tasks | Continue to work on respective features. (Nicholas - Personal Mentor App) (Jonathan - Improve look of modals) |

|  |  |
| --- | --- |
| Date | 10/20/14 |
| Location | ECS237 |
| Start | 11:00 AM |
| End | 3:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Discuss progress and yii framework and also prepare for today’s presentation. |
| Summary of Discussion | Last minute touch ups made to our code. Merged our branches and pulled to the demo site. Work on some sequence diagrams. |
| Assigned Tasks | Continue to work on respective features. Finalize Feasibility Document |

|  |  |
| --- | --- |
| Date | 10/27/14 |
| Location | ECS237 |
| Start | 10:00 AM |
| End | 3:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Coding Session & Documentation |
| Summary of Discussion | Worked on respective features |
| Assigned Tasks | None |

|  |  |
| --- | --- |
| Date | 11/03/14 |
| Location | ECS237 |
| Start | 10:00 AM |
| End | 3:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Coding Session & Prepare for Presentation |
| Summary of Discussion | Worked on respective features |
| Assigned Tasks | None |

|  |  |
| --- | --- |
| Date | 11/10/14 |
| Location | ECS237 |
| Start | 12:00 PM |
| End | 3:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Coding Session & Documentation |
| Summary of Discussion | Worked on respective features |
| Assigned Tasks | None |

|  |  |
| --- | --- |
| Date | 11/17/14 |
| Location | ECS237 |
| Start | 11:00 AM |
| End | 3:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Coding Session |
| Summary of Discussion | Worked on respective features |
| Assigned Tasks | None |

|  |  |
| --- | --- |
| Date | 11/19/14 |
| Location | Google Plus |
| Start | 12:00 PM |
| End | 3:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez, Juan Caraballo |
| Late | N/A |
| Agenda | Update client on progress |
| Summary of Discussion | Recognized progress is being made, but also recognized we are falling behind |
| Assigned Tasks | Jonathan - Finish Admin Application View, Account Invites.Nick - Finish the Domain Mentor Application, More info on Hover |

|  |  |
| --- | --- |
| Date | 11/27/14 |
| Location | ECS237 |
| Start | 12:00 PM |
| End | 3:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Coding Session |
| Summary of Discussion | Hammered out more Coplat functionality |
| Assigned Tasks | None |

|  |  |
| --- | --- |
| Date | 12/08/14 |
| Location | ECS237 |
| Start | 11:00 AM |
| End | 5:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez |
| Late | N/A |
| Agenda | Senior Project Posters |
| Summary of Discussion | Work on posters. Collaborate on verbage. |
| Assigned Tasks | None |

|  |  |
| --- | --- |
| Date | 12/09/14 |
| Location | Google Plus |
| Start | 11:00 AM |
| End | 2:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez, Juan Caraballo |
| Late | N/A |
| Agenda | Do a full run through the registration process to look for bugs and recognize potential improvements |
| Summary of Discussion | Lots of progress since last time but some key functionality was missing |
| Assigned Tasks | Jonathan - Email Invites, Admin Proposals.Nick - Allow for multiple projects per mentor, Instruction popovers,Cosmetic improvements |

|  |  |
| --- | --- |
| Date | 12/10/14 |
| Location | Google Plus |
| Start | 11:00 AM |
| End | 2:00 PM |
| In Attendance | Nicholas Madariaga, Jonathan Sanchez, Juan Caraballo |
| Late | N/A |
| Agenda | Do another full run through the registration process to look for bugs and recognize potential improvements. |
| Summary of Discussion | Almost done missing one key feature |
| Assigned Tasks | Jonathan - App approval.Nick - Recommend Domain, Cosmetic improvements |

## 

## 10 References

**Mobile Judge V1**

<http://users.cis.fiu.edu/~sadjadi/Teaching/SeniorProject/Deliverables/SP13/7-MobileJudge/>

**Mobile Judge V2**

<http://users.cis.fiu.edu/~sadjadi/Teaching/SeniorProject/Deliverables/FA13/4-MobileJudgeV2/>

**Virtual Job Fair V1**

<http://users.cis.fiu.edu/~sadjadi/Teaching/SeniorProject/Deliverables/SP13/3-VirtualJobFair/>

**Virtual Job Fair V2**

<http://users.cis.fiu.edu/~sadjadi/Teaching/SeniorProject/Deliverables/FA13/7-VirtualJobFairV2/>

**Mentor Module V1**

<http://users.cis.fiu.edu/~sadjadi/Teaching/SeniorProject/Deliverables/SP14/02-CollaborativePlatform.rar>

**Mentor Module V2**

<http://users.cis.fiu.edu/~sadjadi/Teaching/SeniorProject/Deliverables/SU14/02-CollaborativePlatformV2.zip>