**Collaborative Platform**

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

**Design Document**

Senior Project – CIS 4911 Section U01

4/23/2015

**Members:** **Instructor:**

Henry M. Romero Masoud Sadjadi

Steven S. Sanabria

Lorenzo Sanchez **Mentor:**

Anibal Sicilia Juan Caraballo

Nicholas Madariaga

Jonathan Sanchez

**Version 5**

Adrian Alfonso

Jorge Travieso

*This page was intentionally left blank.*

Copyright © Florida International University 2014. All Rights Reserved. No part of the Collaborative Platform Project or its documentation may be reproduced or modified without the express written consent of Florida International University.

# 

# Abstract

The design document begins with a reintroduction of the basis of the problem to be solved with the collaborative platform. Design methodology, definitions and acronyms are also covered, as well as a general overview of the document in its entirety. This is followed by describing the systems design, subsystem decomposition, hardware and software mapping, as well as an outline of the data management.

The third chapter within this document is aimed at a more detailed design. This includes static and dynamic models of class diagrams, sequence diagrams, state machine diagrams and many others. The chapter finishes with code specification. The fourth chapter focuses on the glossary; the fifth includes a list of appendices. From these appendices, complete use cases are found, as well as their diagrams, documented class interfaces and a list of meetings during this process. At the end of the document is a list of references used to complete this design document.

**Table of Contents**

Abstract

[1. Introduction](#h.2et92p0)

[1.1 Problem Definition](#h.tyjcwt)

[1.2 Design Methodology](#h.3dy6vkm)

[1.3 Definitions, Acronyms, and Abbreviations](#h.1t3h5sf)

[1.4 Overview of Document](#h.17dp8vu)

[2. System Design](#h.3rdcrjn)

[2.1 System Design Overview](#h.26in1rg)

[2.2 Subsystem Decomposition](#h.1ksv4uv)

[2.3 Hardware and Software Mapping](#h.44sinio)

[2.4 Persistent Data Management](#h.2jxsxqh)

[2.5 Security/Privacy](#h.3j2qqm3)

[3. Detailed Design](#h.1y810tw)

[3.1 Overview](#h.4i7ojhp)

[3.2 Static Model](#h.2xcytpi)

[3.3 Dynamic Model](#h.3as4poj)

3.4 Code Specification

[4. Glossary](#h.28h4qwu)

[5. Appendix](#h.nmf14n)

[5.1 Appendix A – Complete Use Cases Diagrams](#h.37m2jsg)

[5.2 Appendix B – Complete Use Cases](#h.2lwamvv)

5.3 Appendix C – Documented Class Interfaces

[5.4 Appendix D – Diary of Meetings](#h.3cqmetx)

[6. References](#h.2a997sx)

# Introduction

This first chapter will briefly state the current problem that needs to be solved and the design methodology of the project. It will also cover miscellaneous project specific terms, acronyms and abbreviations. Finally, the last section within this chapter will provide a general overview of what this document entails.

## 1.1 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.

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 amount 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.

## 1.2 Design 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.

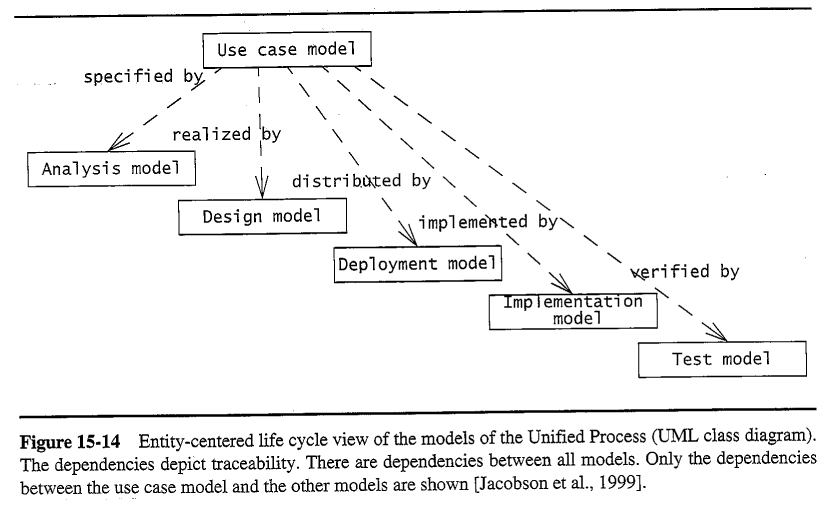


Figure 1.2. Unified Software Process Model

## 1.3 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.
* **Product owner**–The product owner represents the client and has enough domain knowledge to answer the questions the development team may have about the software product.
* **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 an object that is created when a mentor makes preferred selections during registration (e.g a domain mentor chooses domains which the admin then has to approve/reject)

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
* **SPW** - Senior Project Website
* **MM** - Mentor Module

Abbreviations:

* **N/A**

## 1.4 Overview of Document

In the next chapters we will be covering some aspects related to the design process. Chapter two covers the system design. Section 2.1 provides a high level overview with a description of the architecture design. Section 2.2 shows the subsystems decomposition, which includes a detailed description for each of the major subsystems. Section 2.3 provides the hardware and software mapping for each subsystem. Sections 2.4 and 2.5 contain the persistent data management that provides the identification of data that needs to be stored and any security and privacy requirements, respectively.

Chapter three will cover the detailed design. Section 3.1 starts with a brief overview of the behavior and structure of each subsystem. 3.2 contains diagrams of the static model with a detailed description of the structure for each subsystem. Section 3.3 offers the dynamic model that presents state machine diagram for the main control object in each subsystem. The final section concludes with the code specification which describes the class interfaces and constraints for the main control object in each system.

Chapter four contains the glossary with the terms used in the document. Chapter five includes the Appendix that contains the use case diagram, the complete uses cases description, documented interfaces, constraints and finally the diary of meetings. Lastly, chapter six contains the references used for the elaboration of the design document.

# 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.

## 2.1 System Design Overview

As a result of our understanding of the system, we determined that the system would be divided in three major components where the objects and classes have to be independent. By this, the team chose the three tier architecture model; this is because the system architecture allows the system to be divided into three independent layers, as shown in figure 2.1.a

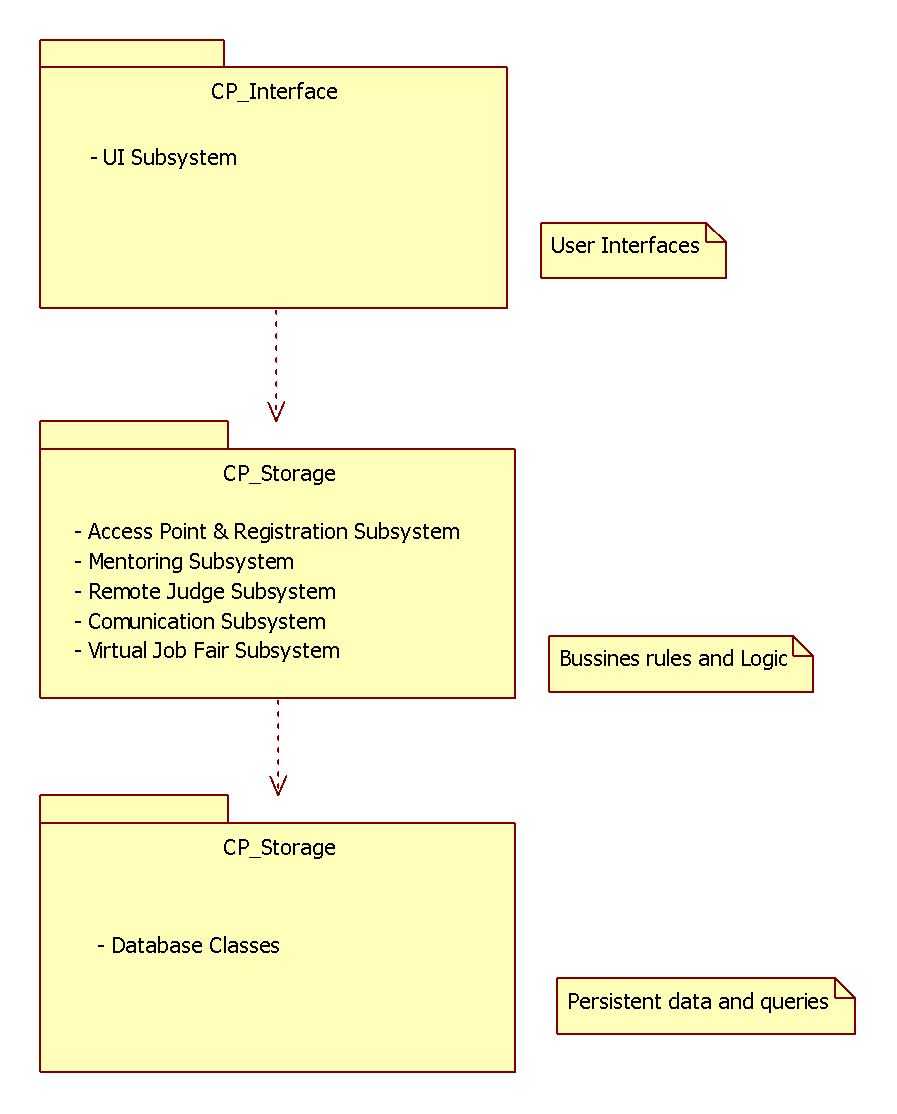


Figure 2.1.a Three Tier

In the figure the interface includes all user interfaces that serves as boundaries with the users. The application Logic will serve as an intermediary between the Interface and the Storage. Moreover, this layer ensures that all processing is performed correctly. The Storage contains all the classes needed to perform the queries of the system objects.

In addition, we are using 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.

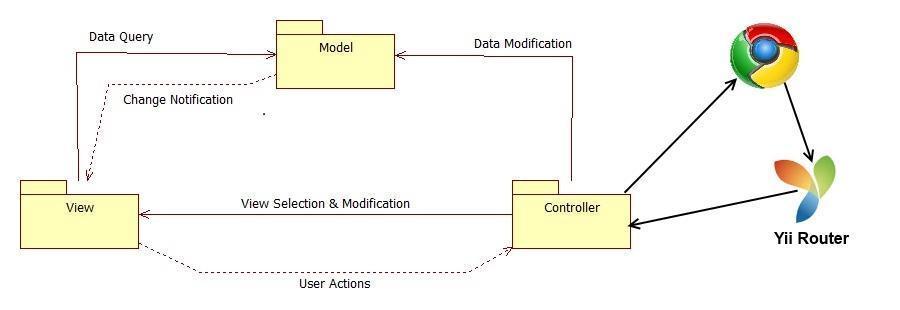


Figure 2.1.b Model View Controller

## 2.2 Subsystem Decomposition

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

1. **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 register User
  + Allow user Login
  + Allow user Logout
  + Allow reset password

1. **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 ability to retrieve information from the senior project website or from LinkedIn. It also covers the essential workflow of the ticketing system for issues created by mentees for the different types of mentors.

* Allow to retrieve project information from the SPW
* Allow to create a question for a specific ticket.
* Allow to add domain.
* Allow to retrieve skill-set/domain expertise from their Linkedln account
* Allow matching tickets with the most relevant domain mentor
* Allow to assign ticket to a domain mentor
* Allow to append comment to a ticket
* Allow to attach file to a ticket
* Allow Administrator to drop a ticket at his discretion
* Allow to close a ticket
* Allow to comment added notification
* Allow mentee to accept answer
* Allow mentee to decline answer

**Version 3**

* Allow a potential mentor to register for an account from a new entry point.
* Allow a Project Mentor to self-register and select projects they like or defer the decision to the system.
* Allow a Personal Mentor to self-register and select students they’d like or defer the decision to the system.
* Allow a Domain Mentor to self-register and select 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 review domains recommended by mentor
* Allow admin to view open invitations/re-invites
* Allow admin to customize the body of the message while sending an invite
* 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.

**Version 5**

* Allow admin to pull reports on mentors.
* Allow admin to pull reports on mentees.
* Allow admin to pull reports on Tickets.
* Allow admin to research on the utilization of tickets created.
* Allow admin to research on the utilization of tickets closed.
* Allow admin to research on the utilization of tickets duration (time opened to closed)
* Allow admin to research on the time that takes mentors to answer.
* Allow admin to research on the tickets that are still open (may or may not be answered).
* Allow admin to research on the tickets that are currently unanswered.
* Allow admin to get system suggestion on what kind of mentors to find (based on frequent mentee subdomains).

1. **Remote Judge Subsystem**

The remote judge subsystem will cover the fundamentals from the remote judge module. These functionalities are essential to the grading and viewing of projects done within the platform. The subsystem covers main requirements for the necessary two way communication found when giving feedback or grading an individual from certain projects.

* Allow judges to get rubric
* Allow judges to post grades
* Allow head professor to send invitation
* Allow show power point
* Allow judge and mentee to use video conference
* Allow judge and mentee to use the whiteboard

1. **Communication Subsystem**

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 to new Message Notification
  + Allow to send new message to a mentor
  + Allow to read message
  + Allow to delete Message
  + Allow to send ticket due notification

1. **Virtual Job Fair**

The virtual job fair system covers a large amount of functionality. The system plays an essential portion of the collaborative platform as its own module. The system allows a streamlined online interaction system for employers and employees/future employees alike. Document sharing and editing is important within the system as well as screen sharing for interview questions and overall viewpoint on work.

* Allow students and employers to register
* Allow students and employers to view respective profiles
* Allow students and employers to edit their basic profile information
* Allow students and employers to take part in a video interview
* Allow students to upload a resume and video resume
* Allow students and employers to upload an image for their profile
* Allow students to associate skills to their profile
* Allow students to add and delete education information
* Allow students to add and delete experience information
* Allow students to integrate with their LinkedIn account to provide education and experience information (security)
* Allow students to apply to open job postings and provide a cover letter
* Allow students to reply to an employer’s message
* Allow students to search for jobs based on skills
* Allow employers to post jobs
* Allow employers to close a job posting
* Allow employers to associate skills to a job posting
* Allow employers to search for students based on skills
* Allow employers to view student profiles
* Allow employers to send messages to students
* Allow employers to give students a “virtual handshake” to show interest in the student
* Allow an administrator to disable an account (security)
* Allow an administrator to close a job posting
* Allow an administrator to validate an employer registration (security)
* Require a username and password to log into the system (security)
* Encrypt the user password before storing into the database (security)
* Require login before viewing user profiles (security)
* Allow users to create a new shared document.
* Allow users to open a document.
* Allow users to share a document.
* Allow users to delete a document.
* Allow users to import a document.
* Allow users to export a document.
* Allow users to rename document.
* Allow users to save a shared document.
* Maintain access boundaries between non-collaborating temporary accounts (security).
* Allow FIU Computer Science Seniors to login using their FIU SCIS credentials
* Allow users to share their screens
* Allow users to view shared screens
* Allow users to stop sharing their screens
* Allow employers to send SMS to students
* Allow users to receive automatic email and SMS reminders
* Allow users to confirm their phone numbers
* Allow students and employers to start using the whiteboard functionality
* Allow students and employers to upload an image to share during an interview
* Allow students to view images uploaded by the other party in an interview
* Allow students and employers show or restore a whiteboard session
* Allow students and employers to select an image to upload to the server for sharing purposes
* Allow students and employers to draw using the whiteboard
* Allow students and employers to change the color of the drawing pencil tool
* Allow students and employers to type text into the whiteboard
* Allow students and employers to clear the drawings of the whiteboard
* Allow students and employers to partially erase drawings from the whiteboard

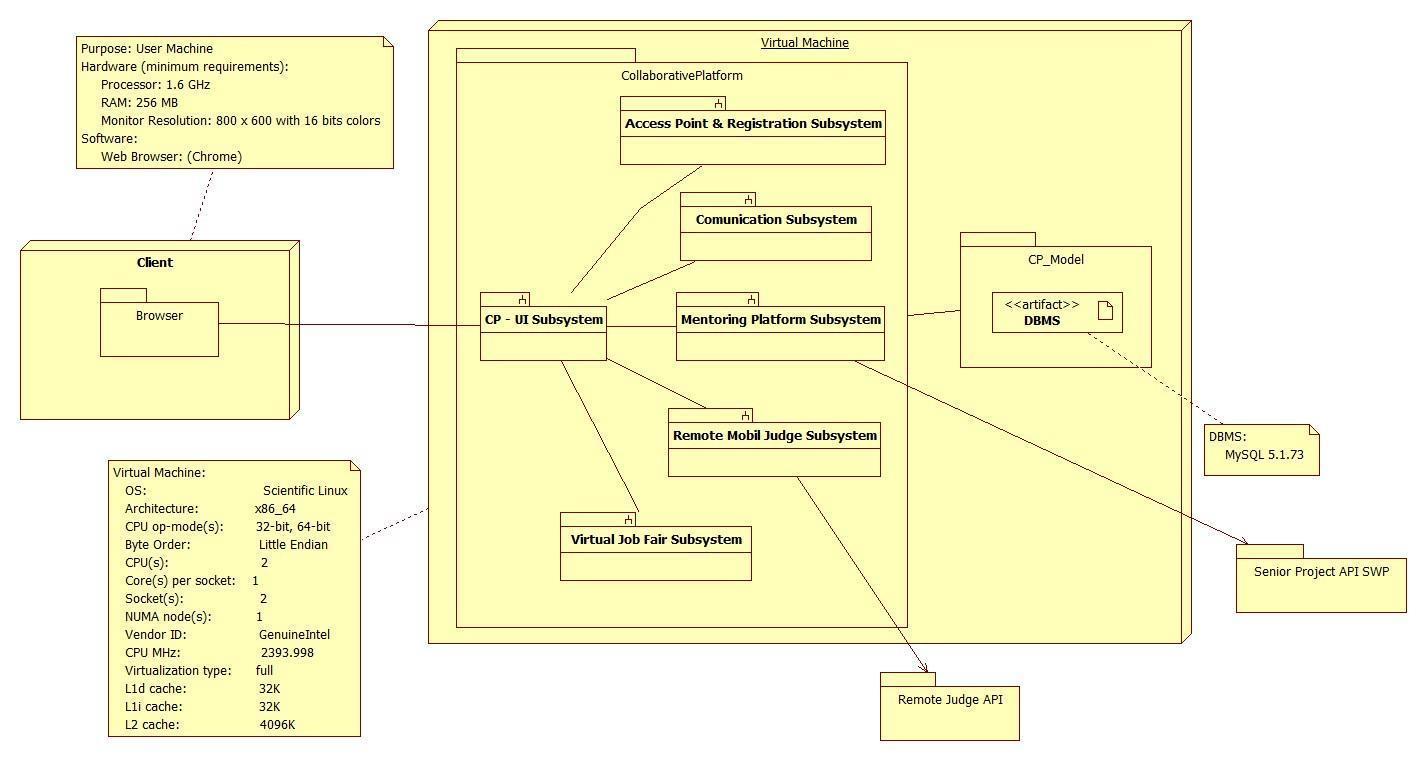
## 2.3 Hardware and Software Mapping

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 2.3: Deployment Diagram**

## 2.4 Persistent Data Management

**ER Model**

**Data Dictionary**

**cp\_attachment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| object\_id | int(11) | No |  |  |  |
| type | char(1) | No |  |  |  |
| file\_id | int(11) | No |  |  |  |
| inline | tinyint(1) | No | 0 |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | object\_id | 8 | A | No |  |
|  |  |  |  | file\_id | 8 | A | No |  |
|  |  |  |  | type | 8 | A | No |  |

**cp\_comment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| comment\_id | int(11) | No |  |  |  |
| ticket\_id | int(11) | No |  |  |  |
| user\_id | int(11) | No |  |  |  |
| added\_date | datetime | No |  |  |  |
| content | varchar(500) | No |  |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | comment\_id | 0 | A | No |  |
|  |  |  |  | ticket\_id | 0 | A | No |  |
|  |  |  |  | user\_id | 0 | A | No |  |
|  |  |  |  | added\_date | 0 | A | No |  |

**cp\_domain**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| domain\_id | int(11) | No |  |  |  |
| name | varchar(100) | No |  |  |  |
| description | varchar(500) | No |  |  |  |
| created\_date | datetime | No |  |  |  |
| updated\_date | datetime | No |  |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | domain\_id | 0 | A | No |  |

**cp\_file**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| id | int(11) | No |  |  |  |
| ft | char(1) | No | T |  |  |
| bk | char(1) | No | D |  |  |
| type | varchar(255) | No |  |  |  |
| size | bigint(20) | No | 0 |  |  |
| key | varchar(86) | No |  |  |  |
| signature | varchar(86) | No |  |  |  |
| name | varchar(255) | No |  |  |  |
| attrs | varchar(255) | Yes | *NULL* |  |  |
| created | datetime | No |  |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 3 | A | No |  |
| Ft | BTREE | No | No | ft | 3 | A | No |  |
| Key | BTREE | No | No | key | 3 | A | No |  |
| signature | BTREE | No | No | signature | 3 | A | No |  |

**cp\_message**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| message\_id | int(11) | No |  |  |  |
| to\_user\_id | int(11) | No |  |  |  |
| from\_user\_id | int(11) | No |  |  |  |
| subject | varchar(100) | No |  |  |  |
| message | varchar(1000) | No |  |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | message\_id | 0 | A | No |  |

**cp\_page**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| id | int(10) | No |  |  |  |
| isactive | tinyint(1) | No | 0 |  |  |
| type | enum('landing', 'offline', 'thank-you', 'other') | No | other |  |  |
| name | varchar(255) | No |  |  |  |
| body | text | No |  |  |  |
| lang | varchar(16) | No | en\_US |  |  |
| notes | text | Yes | *NULL* |  |  |
| created | datetime | No |  |  |  |
| updated | datetime | No |  |  |  |

**Indexes**

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

**cp\_session**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| session\_id | varchar(255) | No |  |  |  |
| session\_data | blob | Yes | *NULL* |  |  |
| session\_expire | datetime | Yes | *NULL* |  |  |
| session\_updated | datetime | Yes | *NULL* |  |  |
| user\_id | int(10) | No | 0 |  |  |
| user\_ip | varchar(64) | No |  |  |  |
| user\_agent | varchar(255) | No |  |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | session\_id | 3 | A | No |  |
| updated | BTREE | No | No | session\_updated | 3 | A | Yes |  |
| user\_id | BTREE | No | No | user\_id | 3 | A | No |  |

**cp\_team**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| team\_id | int(10) | No |  |  |  |
| lead\_id | int(10) | No | 0 |  |  |
| isenabled | tinyint(1) | No | 1 |  |  |
| noalerts | tinyint(1) | No | 0 |  |  |
| name | varchar(125) | No |  |  |  |
| notes | text | Yes | *NULL* |  |  |
| created | datetime | No |  |  |  |
| updated | datetime | No |  |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | team\_id | 2 | A | No |  |
| name | BTREE | Yes | No | name | 2 | A | No |  |
| isnabled | BTREE | No | No | isenabled | 2 | A | No |  |
| lead\_id | BTREE | No | No | lead\_id | 2 | A | No |  |

**cp\_team\_member**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| team\_id | int(10) | No | 0 |  |  |
| staff\_id | int(10) | No |  |  |  |
| updated | datetime | No |  |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | team\_id | 0 | A | No |  |
|  |  |  |  | staff\_id | 0 | A | No |  |

**cp\_ticket**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| ticket\_id | int(11) | No |  |  |  |
| number | varchar(20) | Yes | *NULL* |  |  |
| user\_id | int(11) | No | 0 |  |  |
| user\_email\_id | int(11) | No | 0 |  |  |
| topic\_id | int(10) | No | 0 |  |  |
| team\_id | int(10) | No | 0 |  |  |
| email\_id | int(11) | No | 0 |  |  |
| status | enum('open', 'closed') | No | open |  |  |
| isoverdue | tinyint(1) | No | 0 |  |  |
| isanswered | tinyint(1) | No | 0 |  |  |
| duedate | datetime | Yes | *NULL* |  |  |
| closed | datetime | Yes | *NULL* |  |  |
| lastcomment | datetime | Yes | *NULL* |  |  |
| lastanswer | datetime | Yes | *NULL* |  |  |
| created | datetime | No |  |  |  |
| updated | datetime | No |  |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | ticket\_id | 4 | A | No |  |
| status | BTREE | No | No | status | 2 | A | No |  |
| created | BTREE | No | No | created | 4 | A | No |  |
| closed | BTREE | No | No | closed | 4 | A | Yes |  |
| duedate | BTREE | No | No | duedate | 4 | A | Yes |  |
| topic\_id | BTREE | No | No | topic\_id | 4 | A | No |  |

**cp\_ticket\_attachment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| attach\_id | int(11) | No |  |  |  |
| ticket\_id | int(11) | No | 0 |  |  |
| file\_id | int(10) | No | 0 |  |  |
| ref\_id | int(11) | No | 0 |  |  |
| inline | tinyint(1) | No | 0 |  |  |
| created | datetime | No |  |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | attach\_id | 2 | A | No |  |
| ticket\_id | BTREE | No | No | ticket\_id | 2 | A | No |  |
| ref\_id | BTREE | No | No | ref\_id | 2 | A | No |  |
| file\_id | BTREE | No | No | file\_id | 2 | A | No |  |

**cp\_topic**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| topic\_id | int(11) | No |  |  |  |
| name | varchar(100) | No |  |  |  |
| description | varchar(500) | No |  |  |  |
| domain\_id | int(11) | No |  |  |  |
| created\_date | datetime | No |  |  |  |
| updated\_date | datetime | No |  |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | topic\_id | 0 | A | No |  |
|  |  |  |  | domain\_id | 0 | A | No |  |

**cp\_user**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| id | int(10) | No |  |  |  |
| default\_email\_id | int(10) | No |  |  |  |
| fname | varchar(128) | No |  |  |  |
| created | datetime | No |  |  |  |
| updated | datetime | No |  |  |  |
| lname | varchar(128) | No |  |  |  |
| phone | varchar(15) | No |  |  |  |

**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 1 | A | No |  |

**cp\_user\_email**

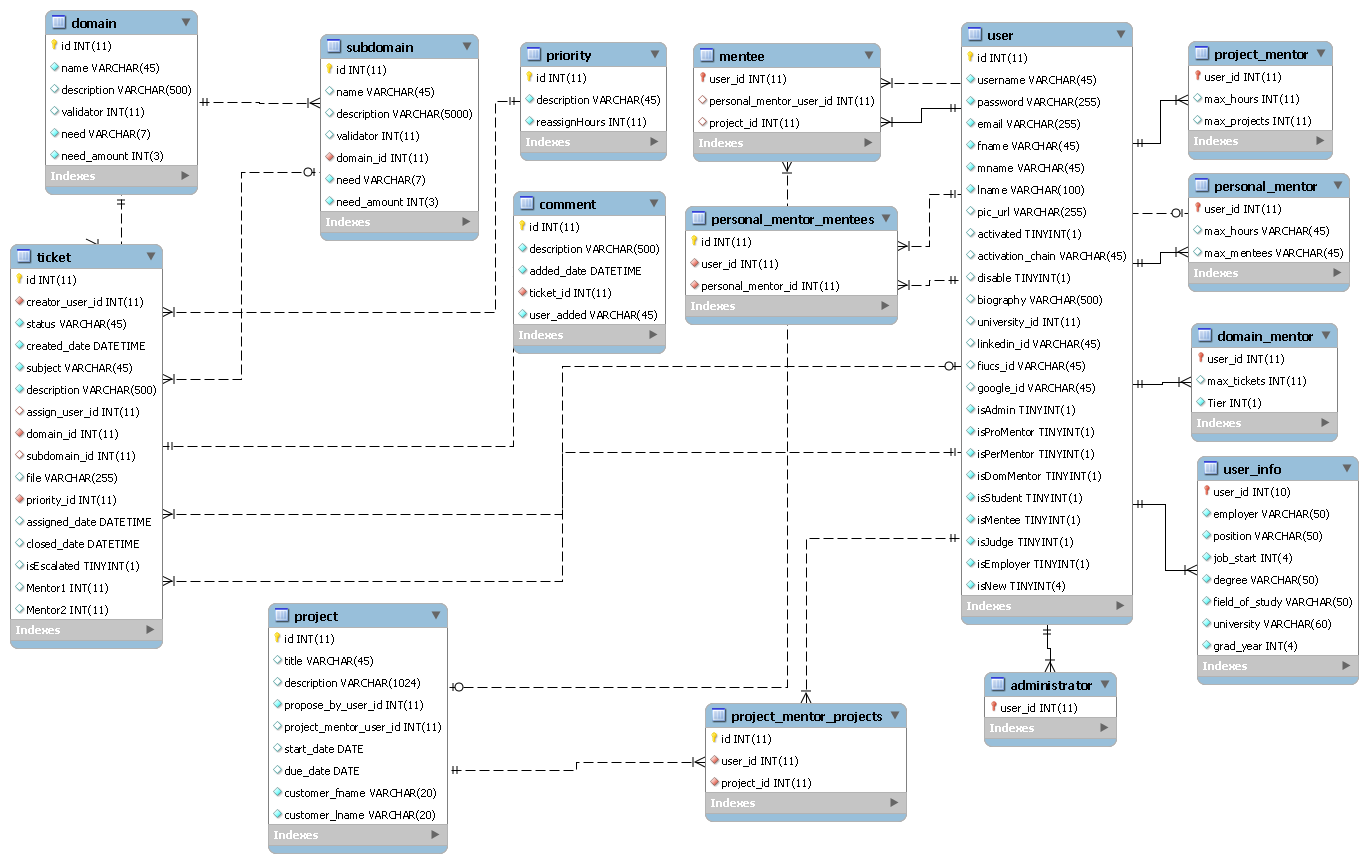
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** | **MIME** |
| id | int(10) | No |  |  |  |
| user\_id | int(10) | No |  |  |  |
| address | varchar(128) | No |  |  |  |

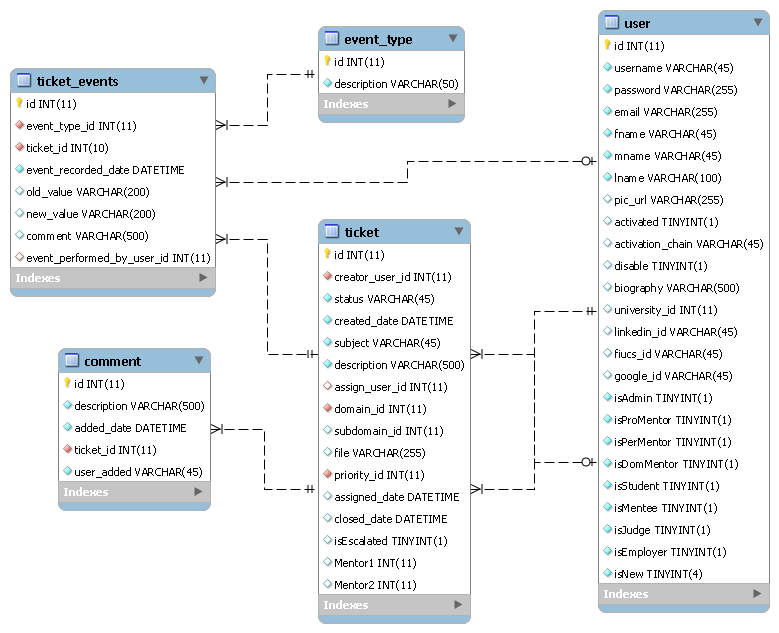
**Indexes**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** | **Comment** |
| PRIMARY | BTREE | Yes | No | id | 1 | A | No |  |
| address | BTREE | Yes | No | address | 1 | A | No |  |
| user\_email\_lookup | BTREE | No | No | user\_id | 1 | A | No |  |

Open new phpMyAdmin window

**Mentoring Module: Version 5**







## 2.5 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.

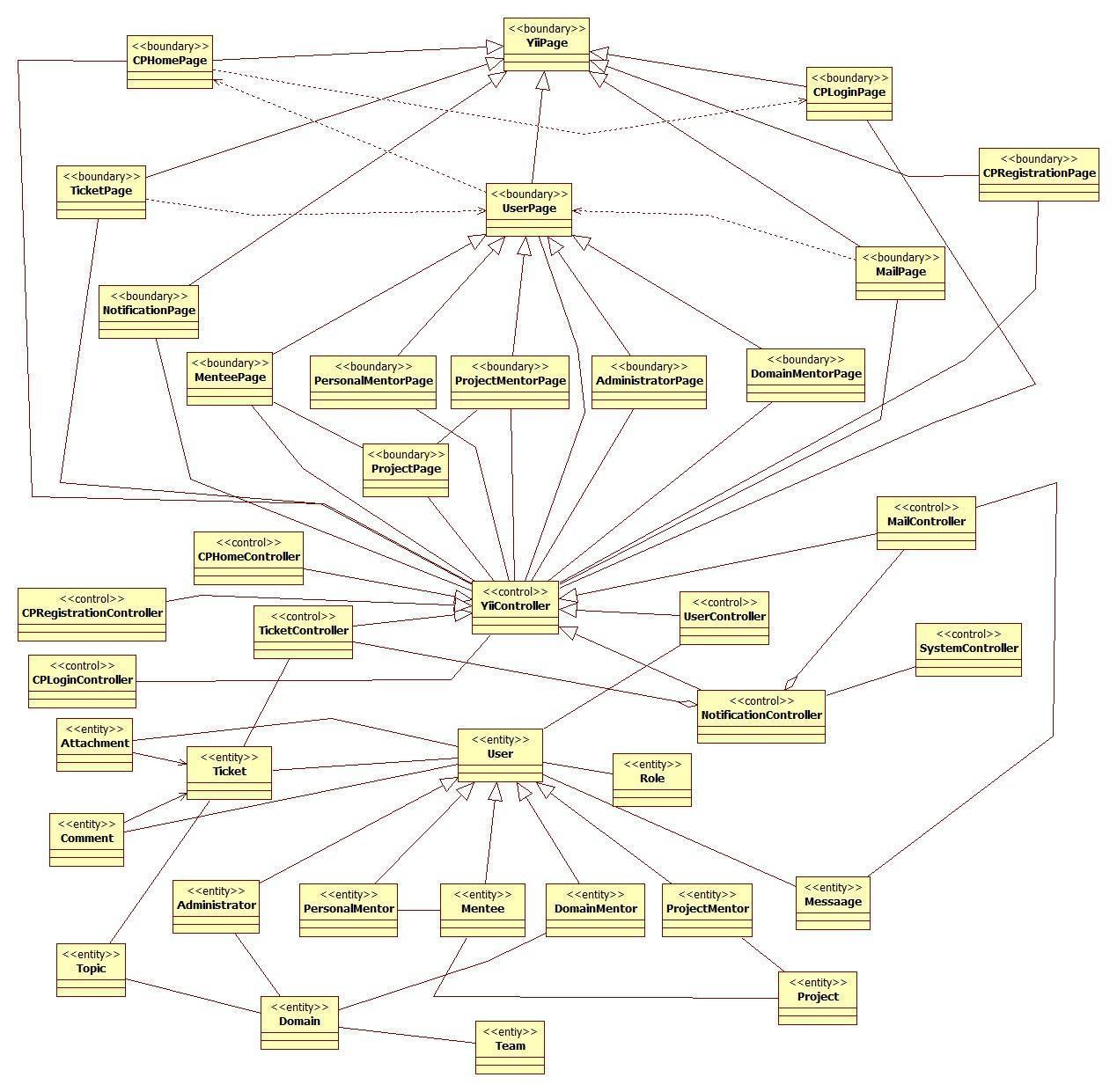
# Detailed Design

This section will describe in detail our proposed detailed design for the system by giving an overview of the four design patterns used. We will also describe the static and dynamic models associated with the system and some sequence diagrams for continuity. The chapter ends with code specification.

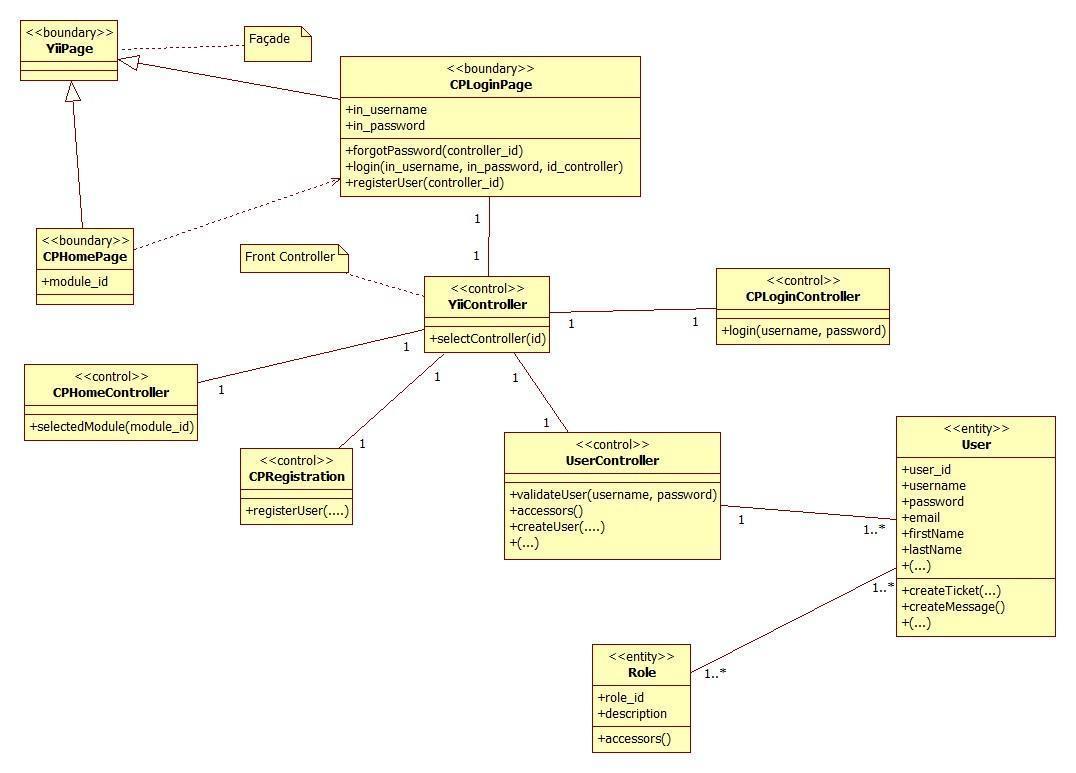
## 3.1 Overview

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.

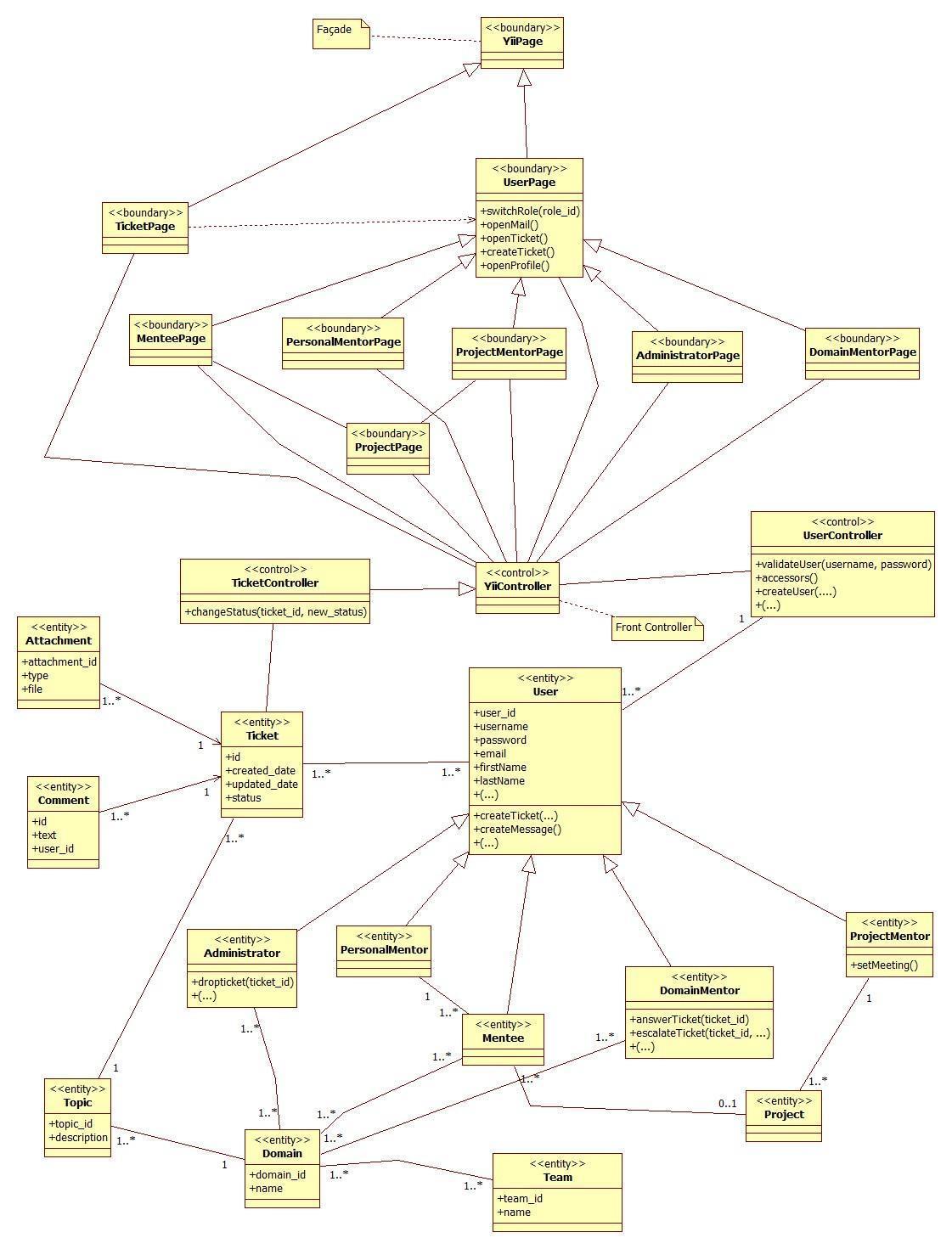
## 3.2 Static Model



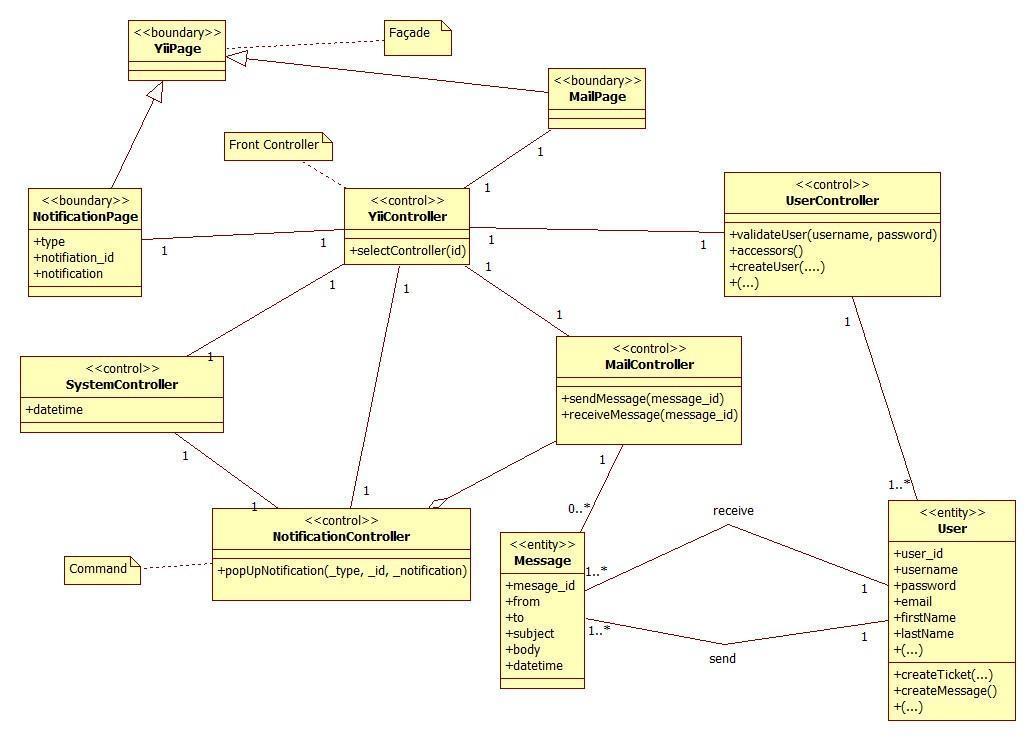
**Figure 3.2.a Minimal Class Diagram**



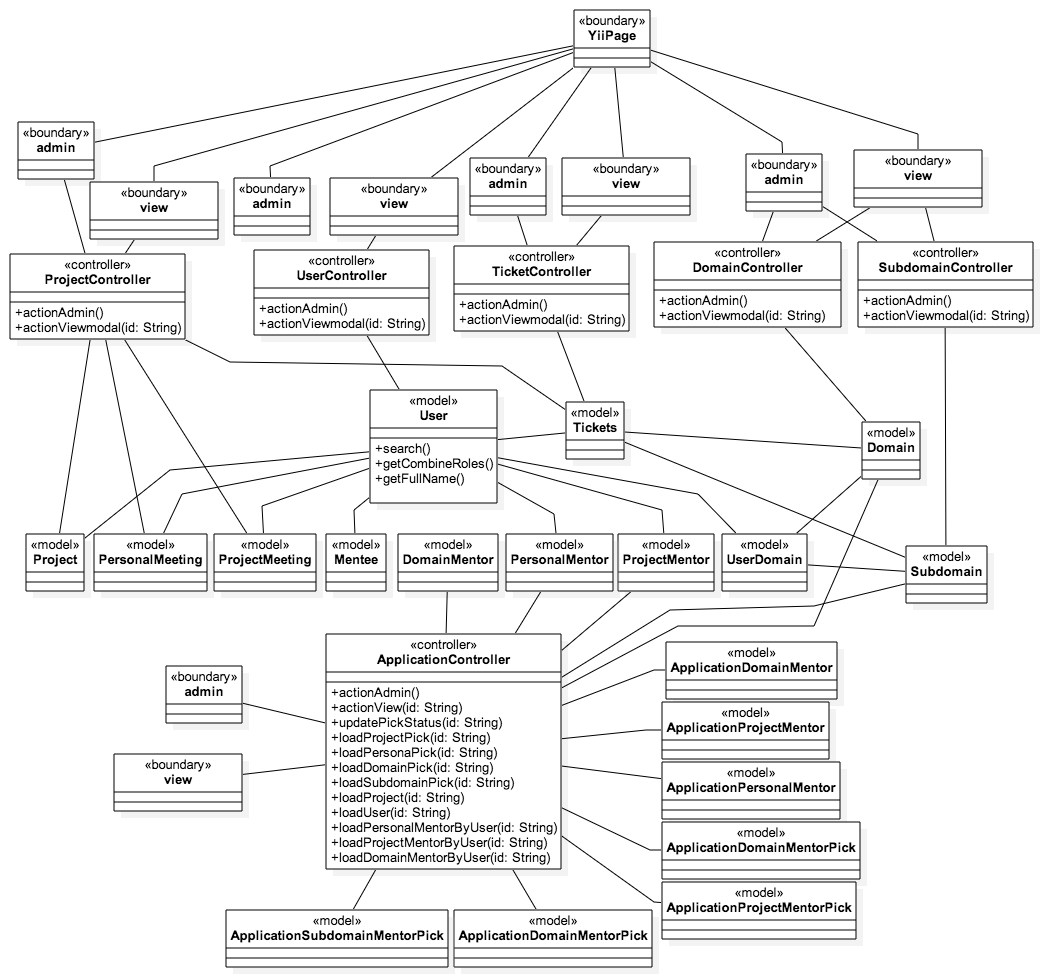
**Figure 3.2.b: Access Point & Registration Detailed Class Diagram**



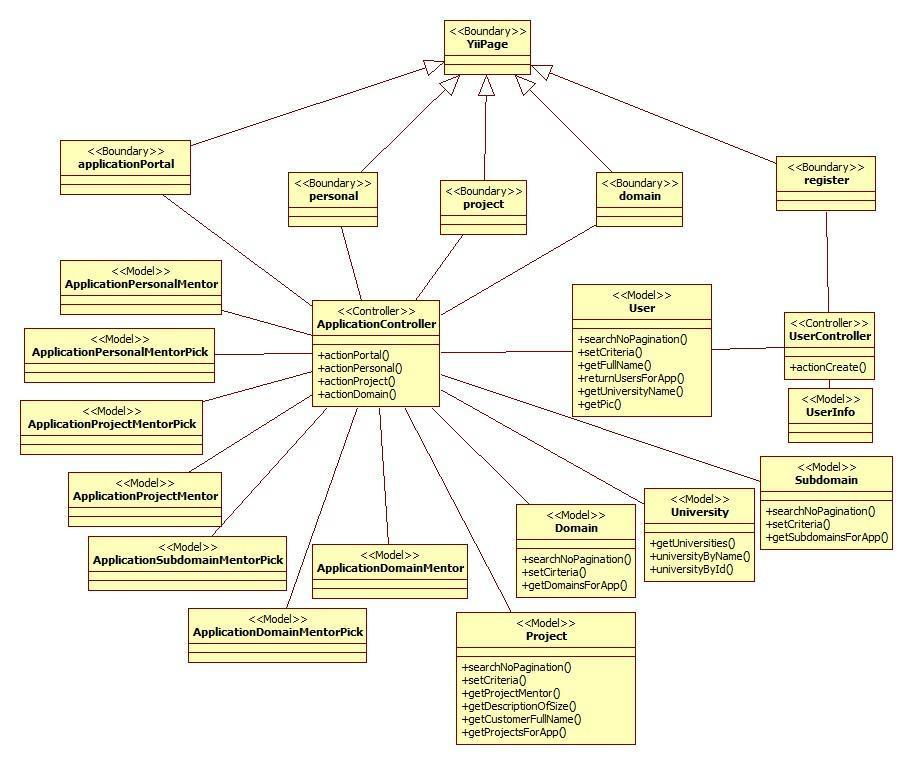
**Figure 3.2.c: Mentoring Subsystem Detailed Class Diagram**



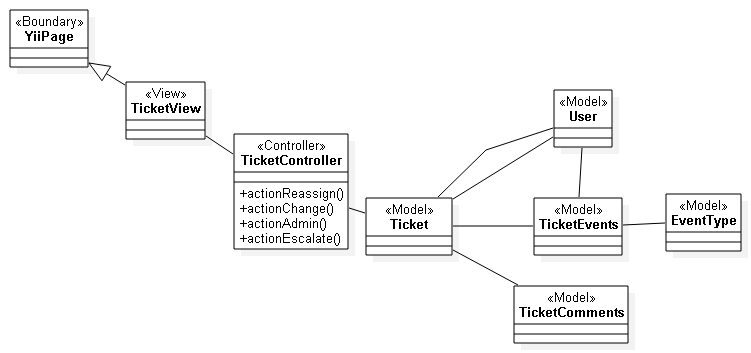
**Figure 3.2.d: Communication Subsystem Detailed Class Diagram**



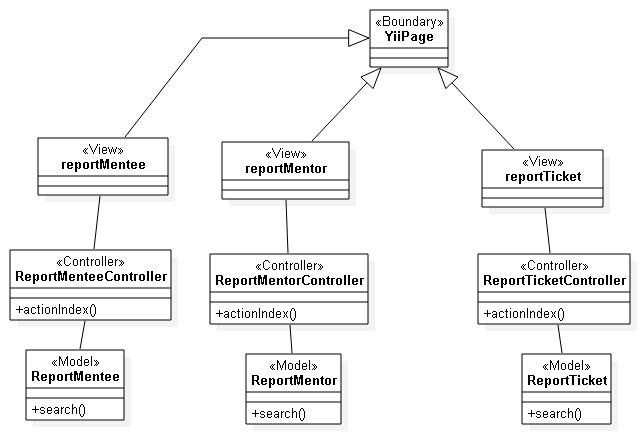
**Figure 3.2.e: Admin Management Detailed Class Diagram**



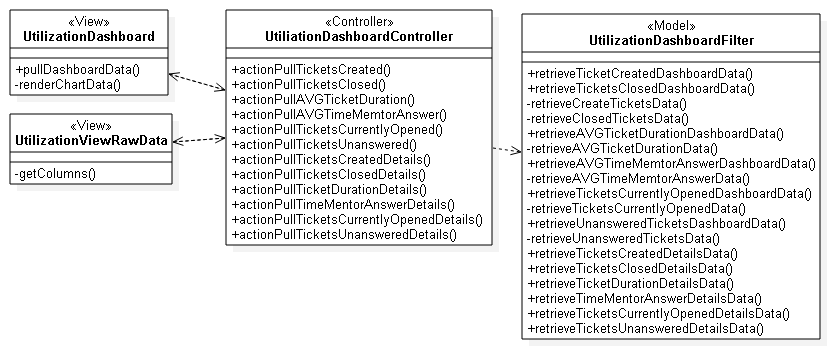
**Figure 3.2.f: Self-Registration Detailed Class Diagram**



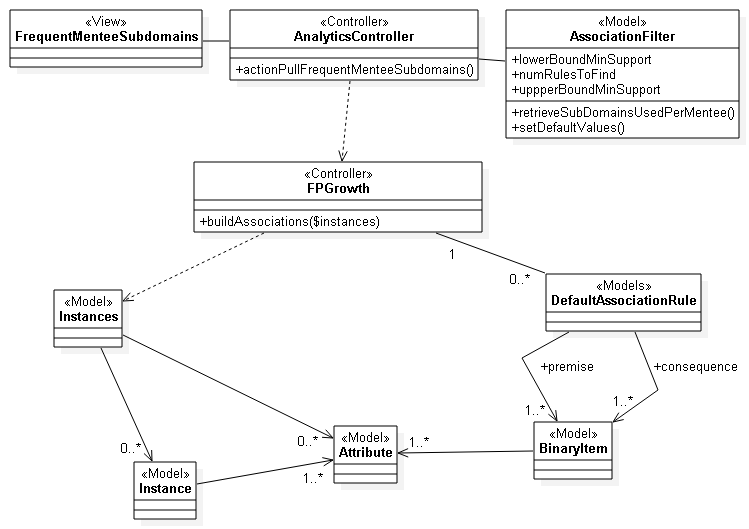
**Figure 3.2.g: Ticket events Detailed Class diagram**



**Figure 3.2.h: Admin reports detailed class diagram**



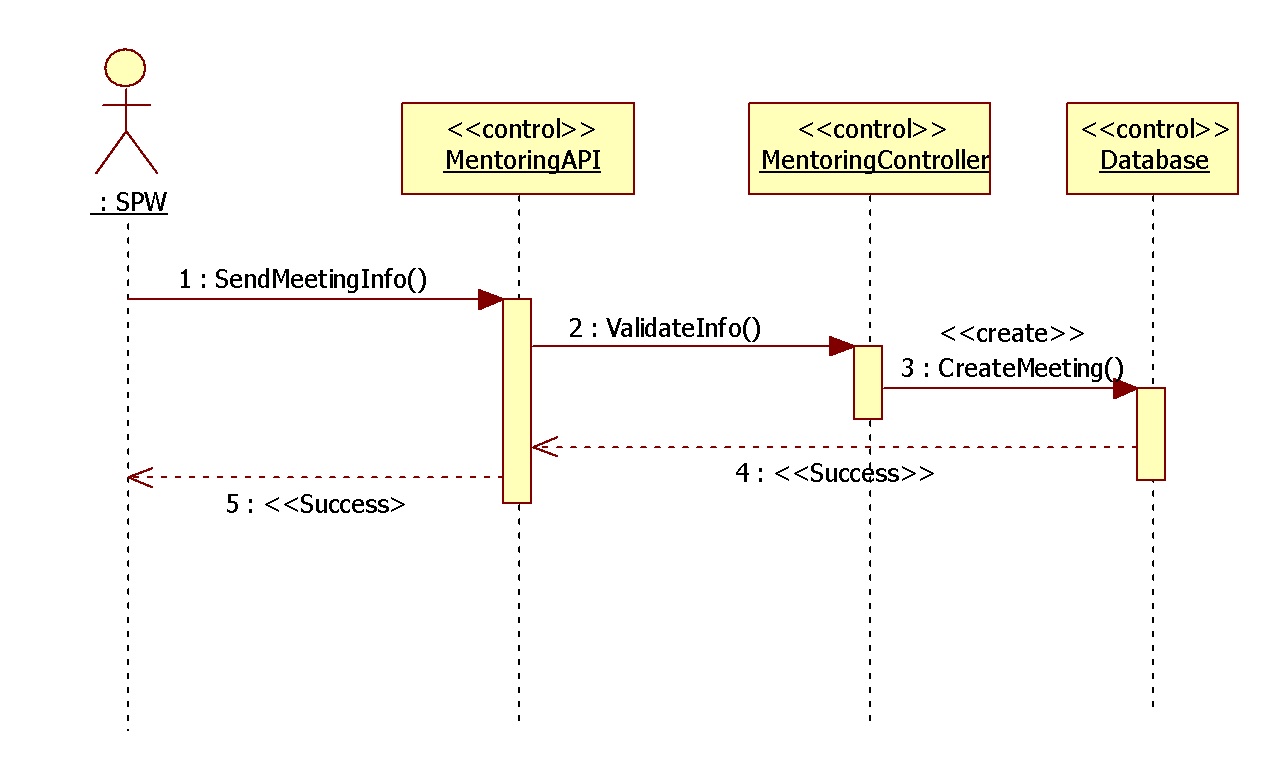
**Figure 3.2.i: Utilization dashboard detailed class diagram**



**Figure 3.2.j: Analytical reporting (Frequent sub-domain questions by mentee) detailed class diagram.**

## 3.3 Dynamic Model

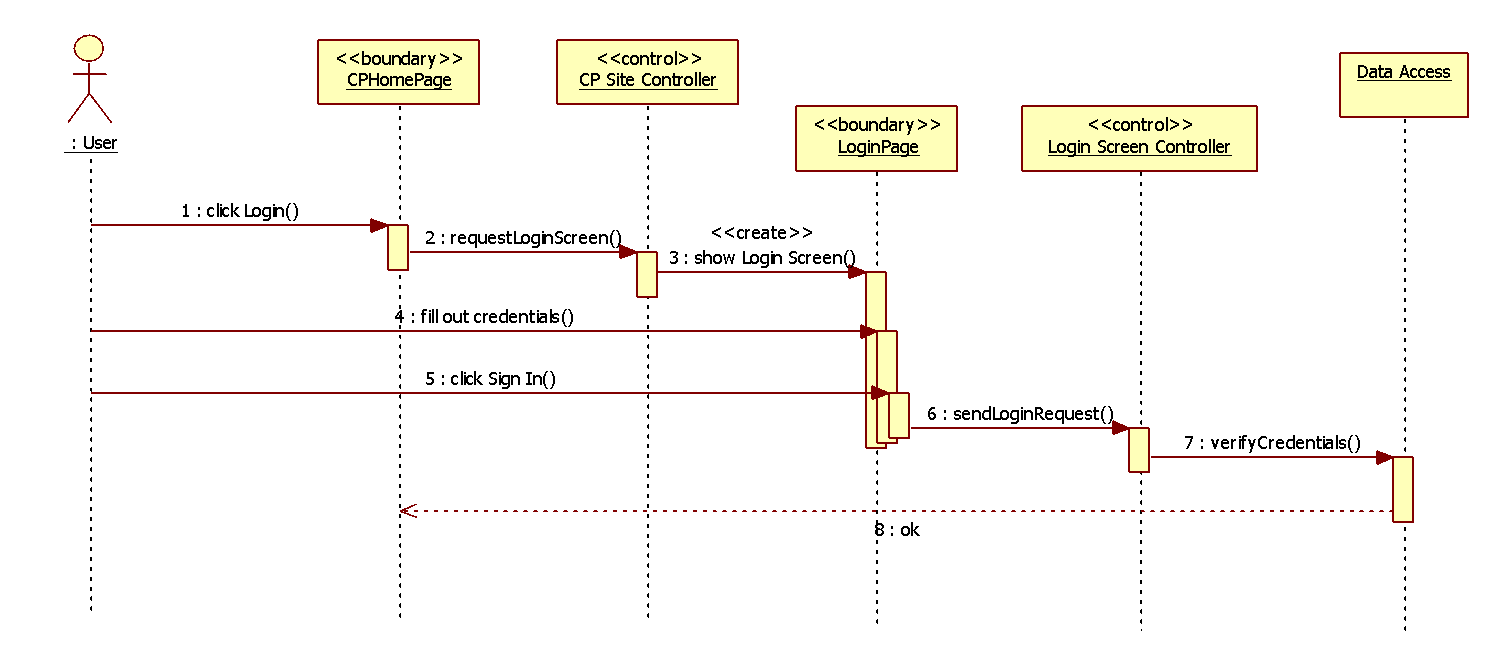
**CP001 - Setup a meeting from SPW**



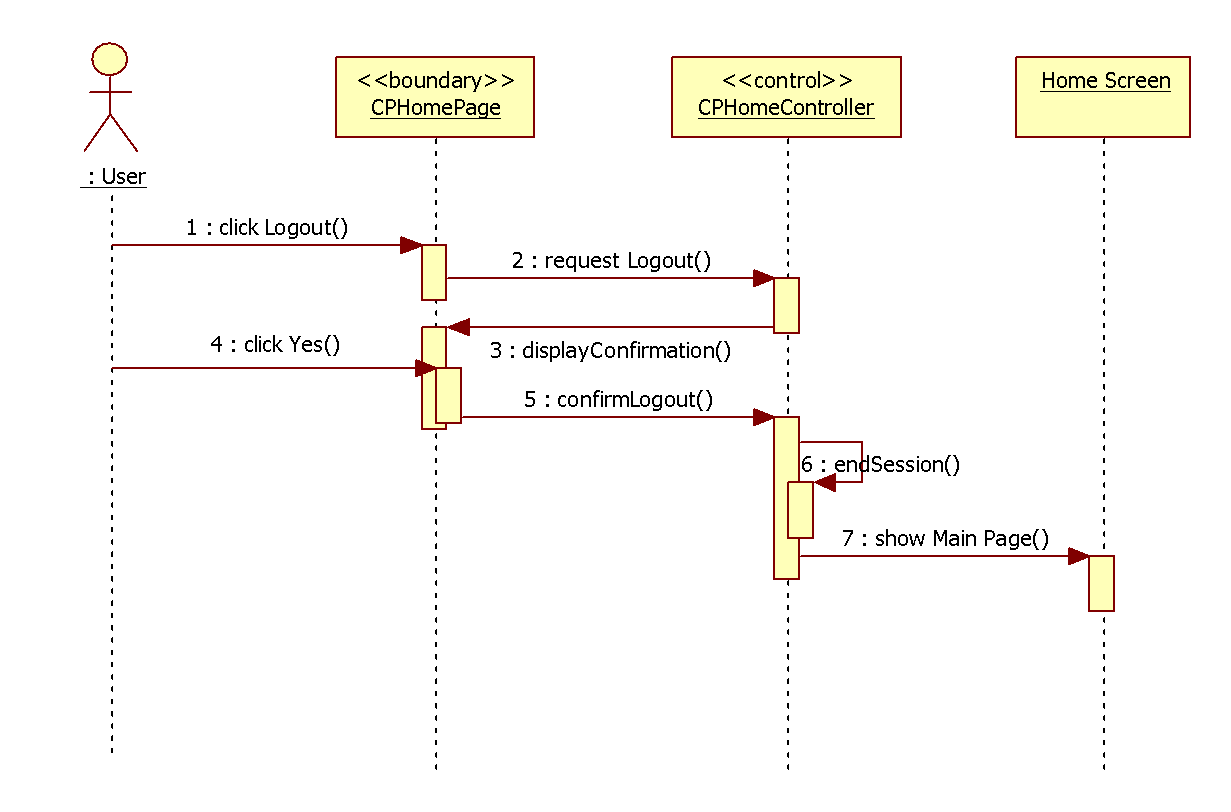
**CP005 – Register**



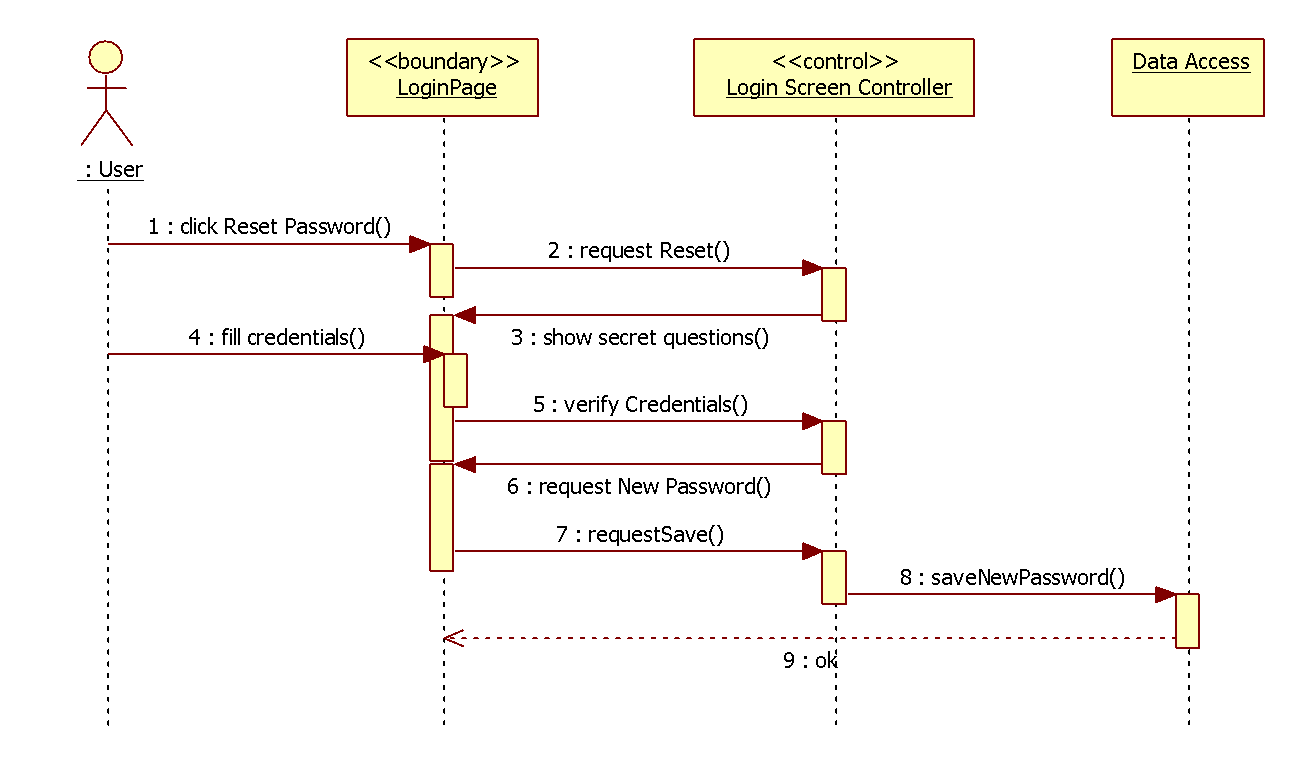
**CP-006 User Login**



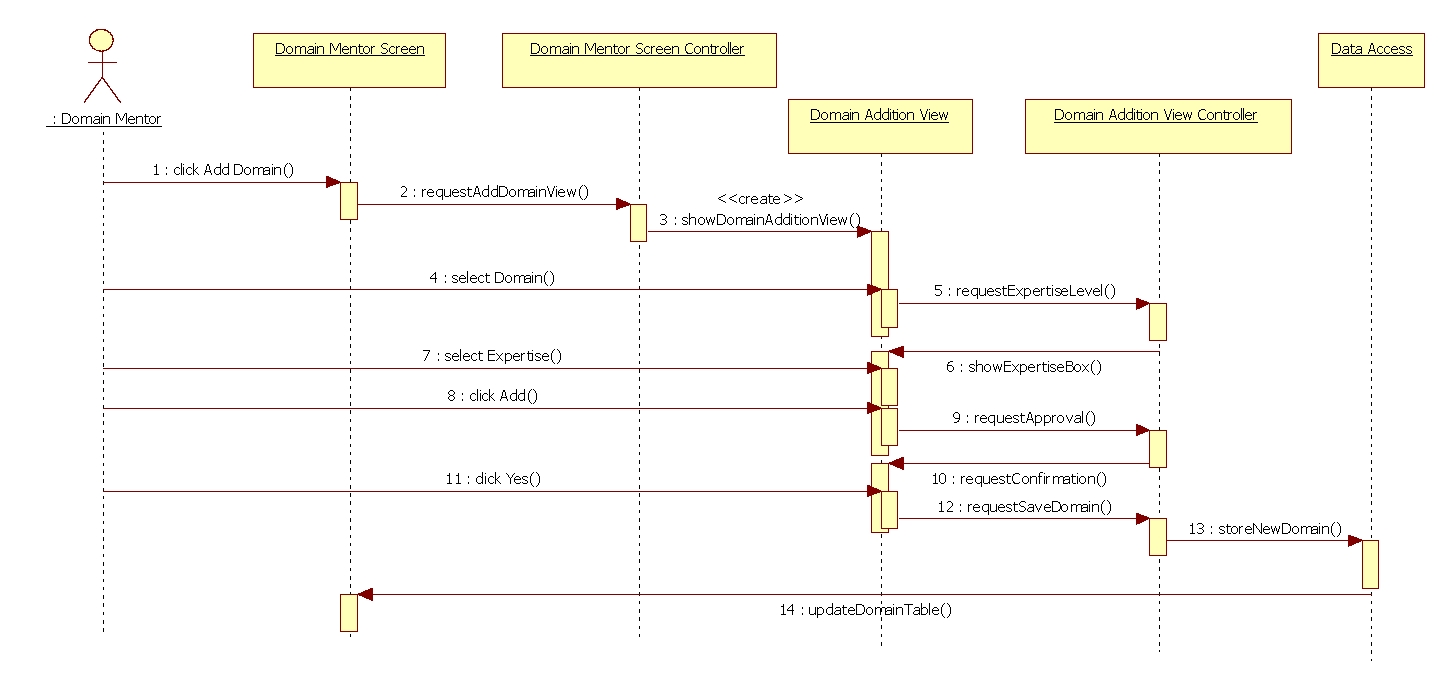
**CP-007 User Logout**



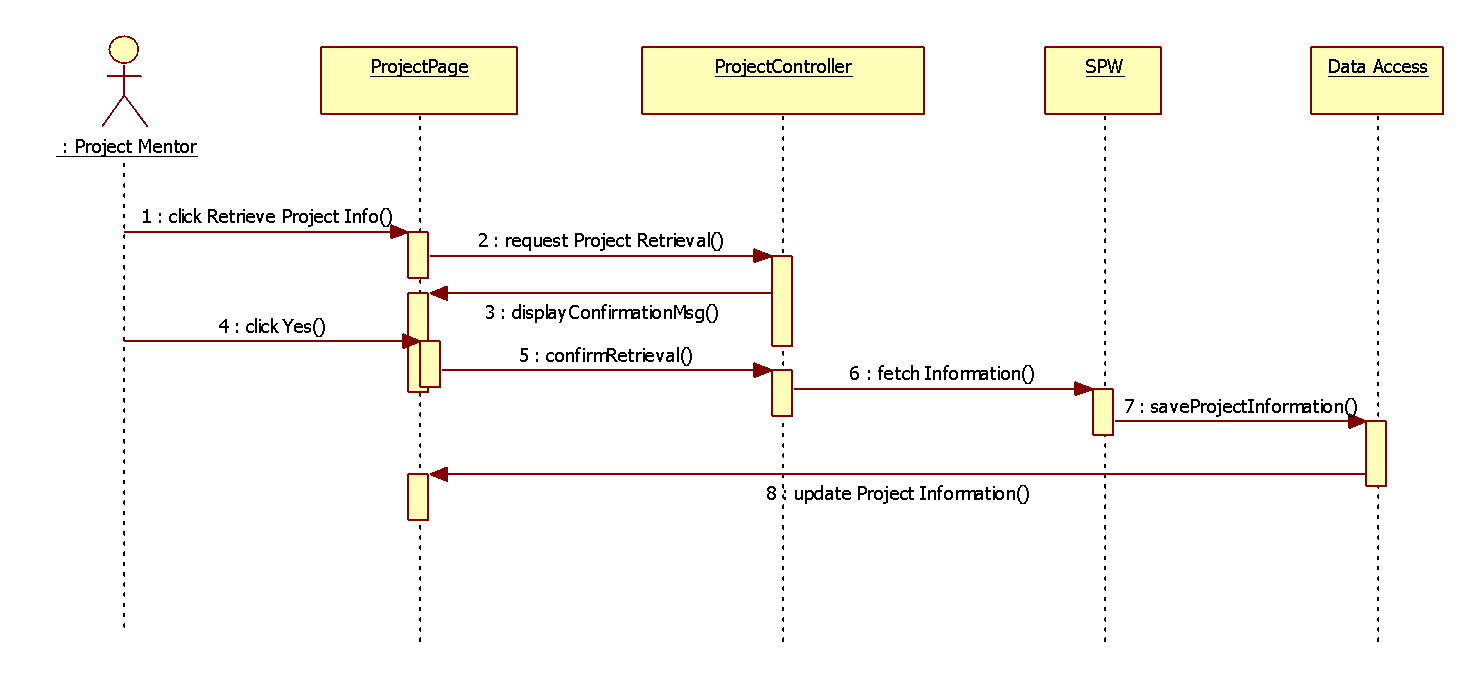
**CP-008 Reset Password**



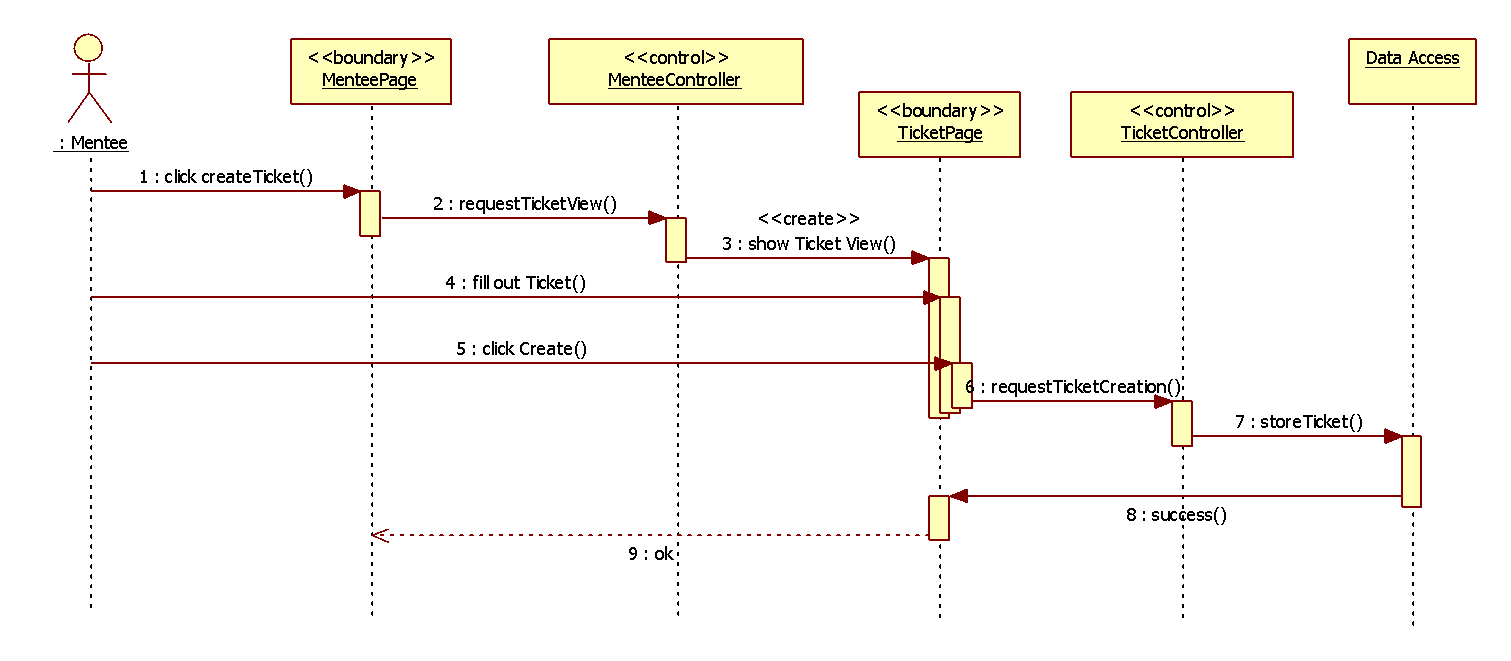
**CP-009 Add Domain**



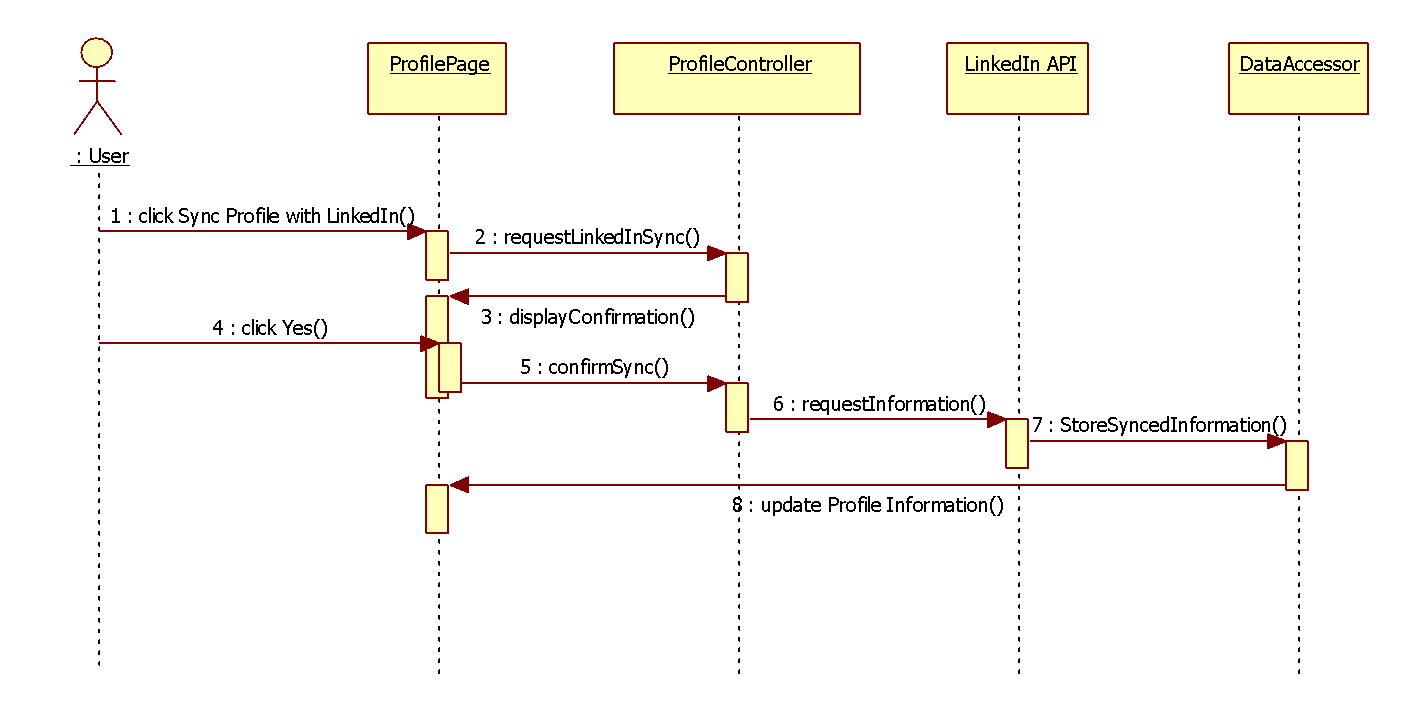
**CP-010 Retrieve project information from the SPW**



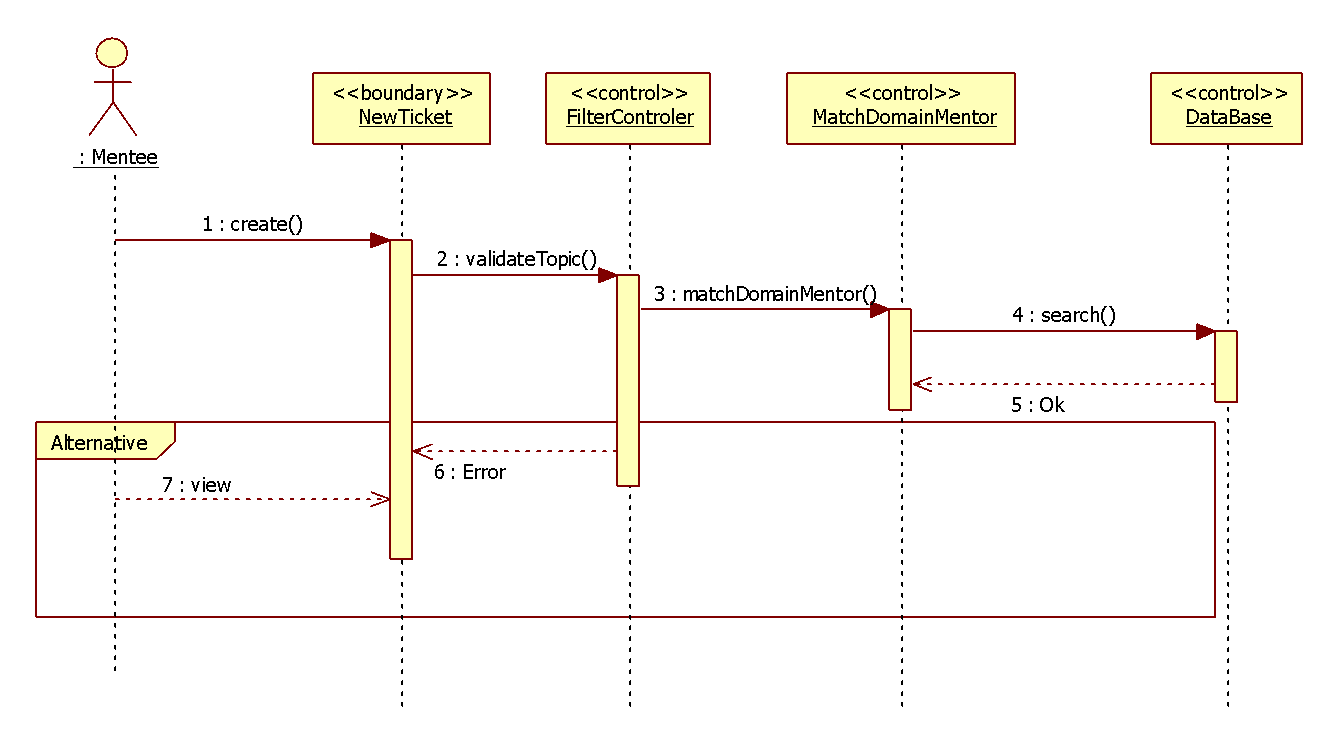
**CP-011 Create a question for a specific ticket.**



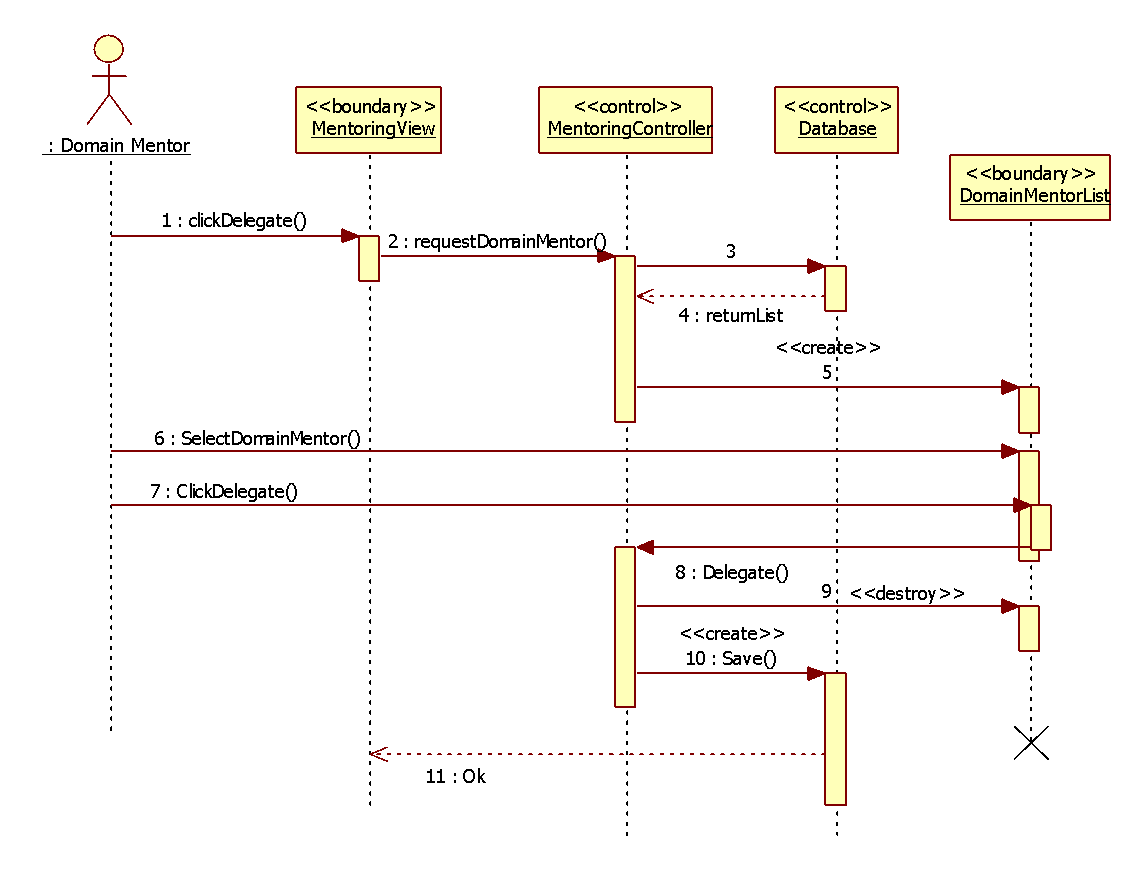
**CP-012 Retrieve skill-set/domain expertise from their LinkedIn account**



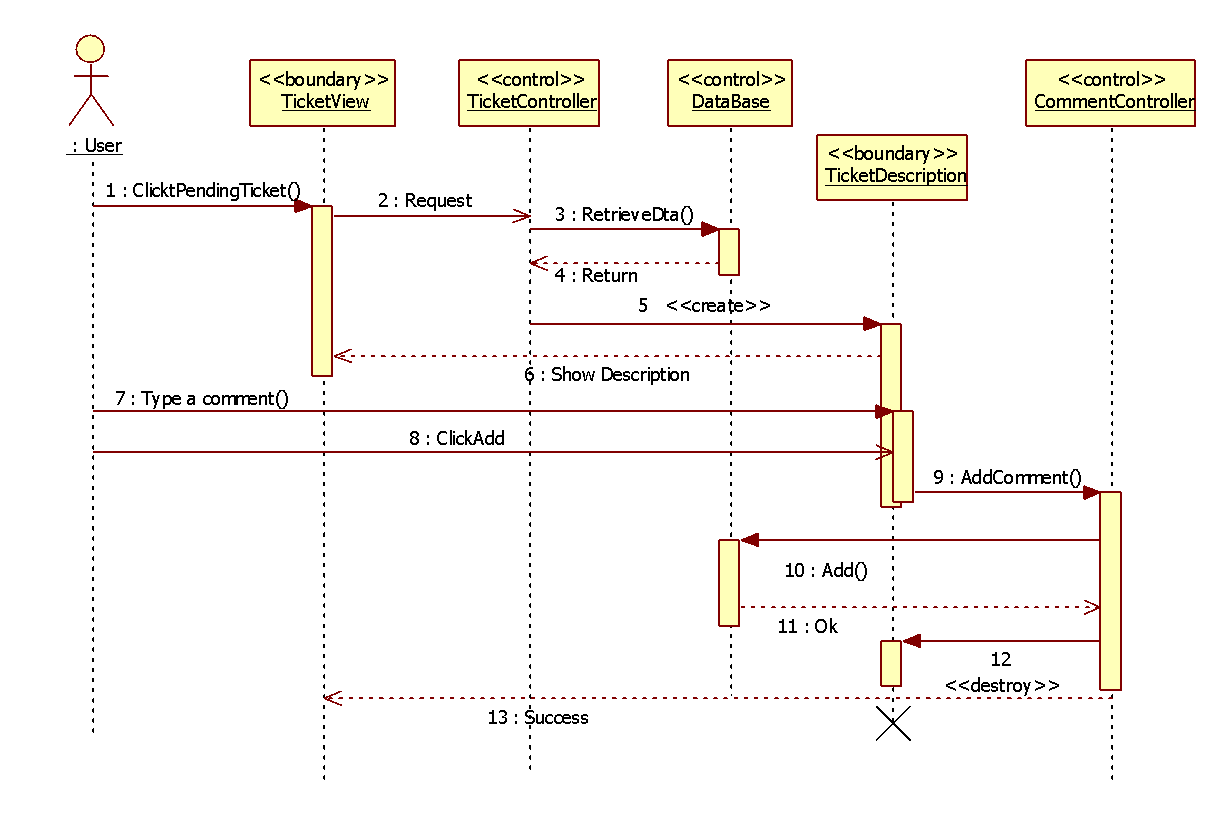
**CP-013 Matching tickets**



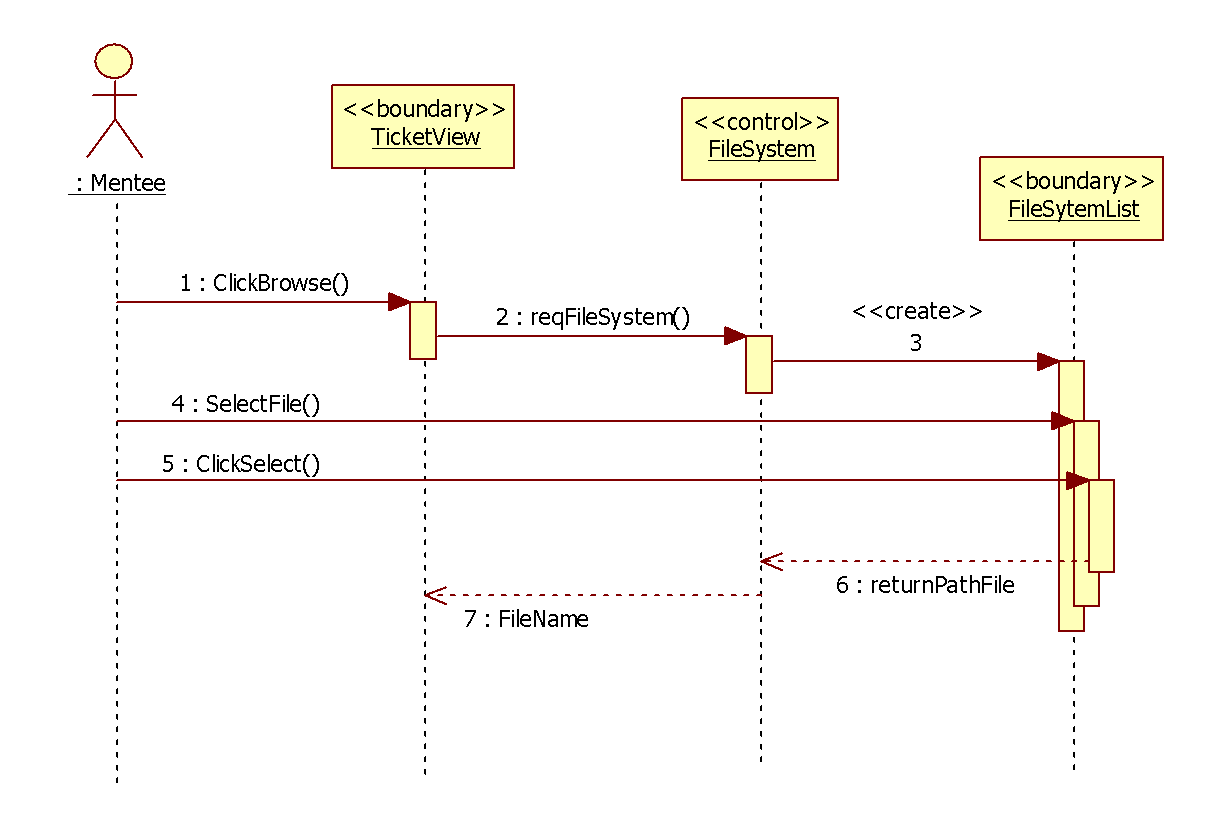
**CP-014 Assign ticket**



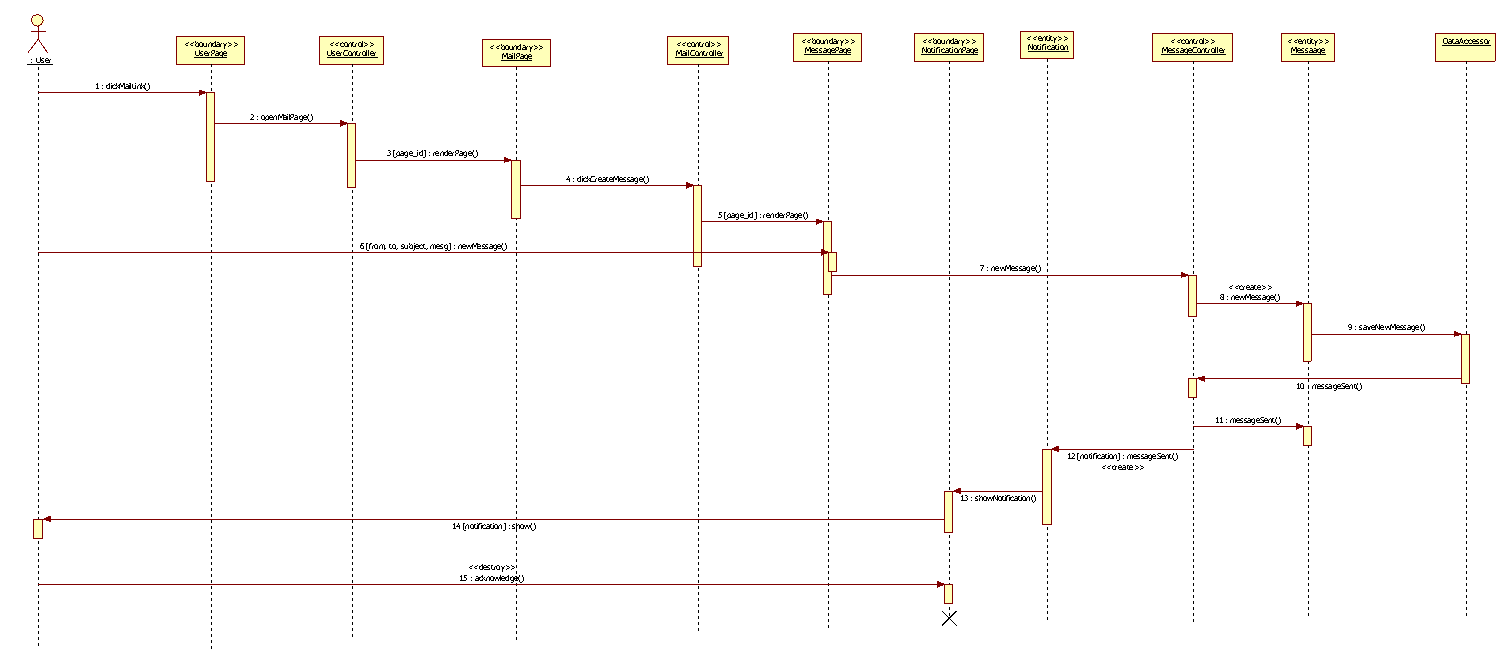
**CP-015 Append comment to a ticket**



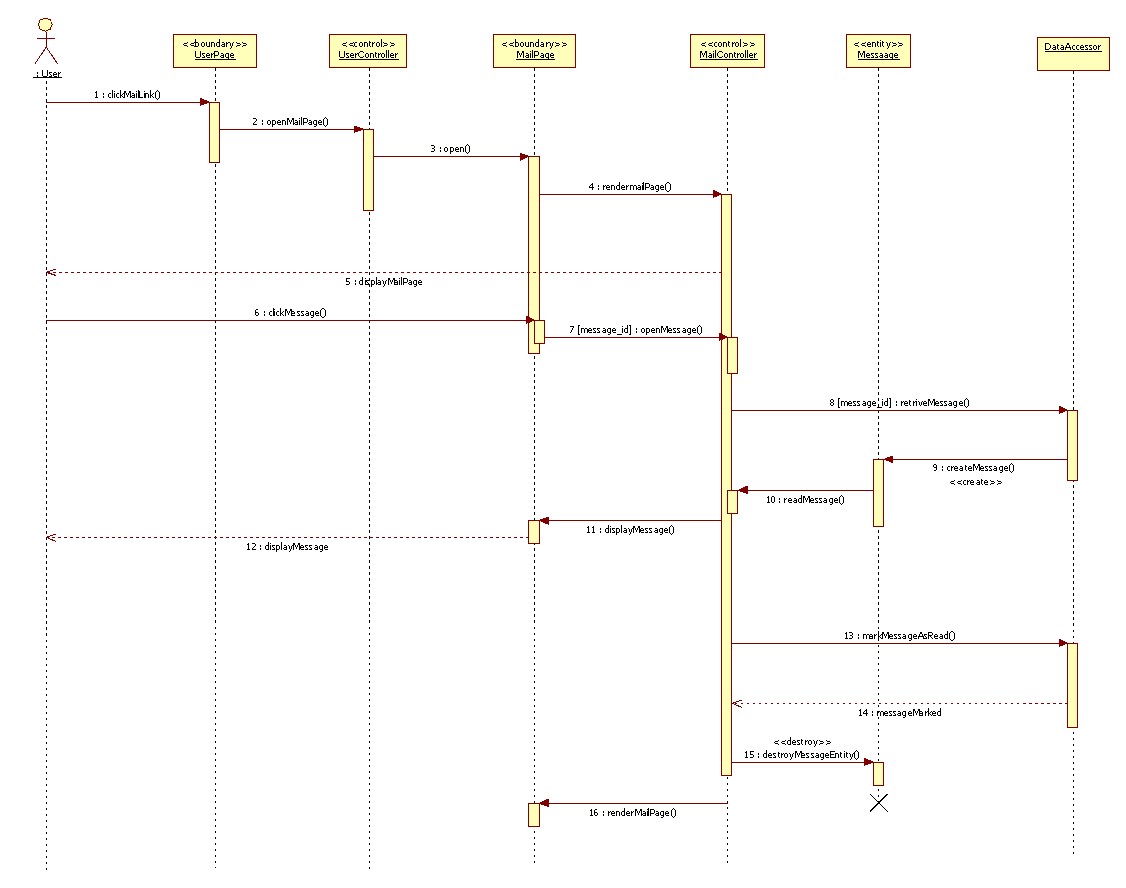
**CP-016 Attach file to a ticket**



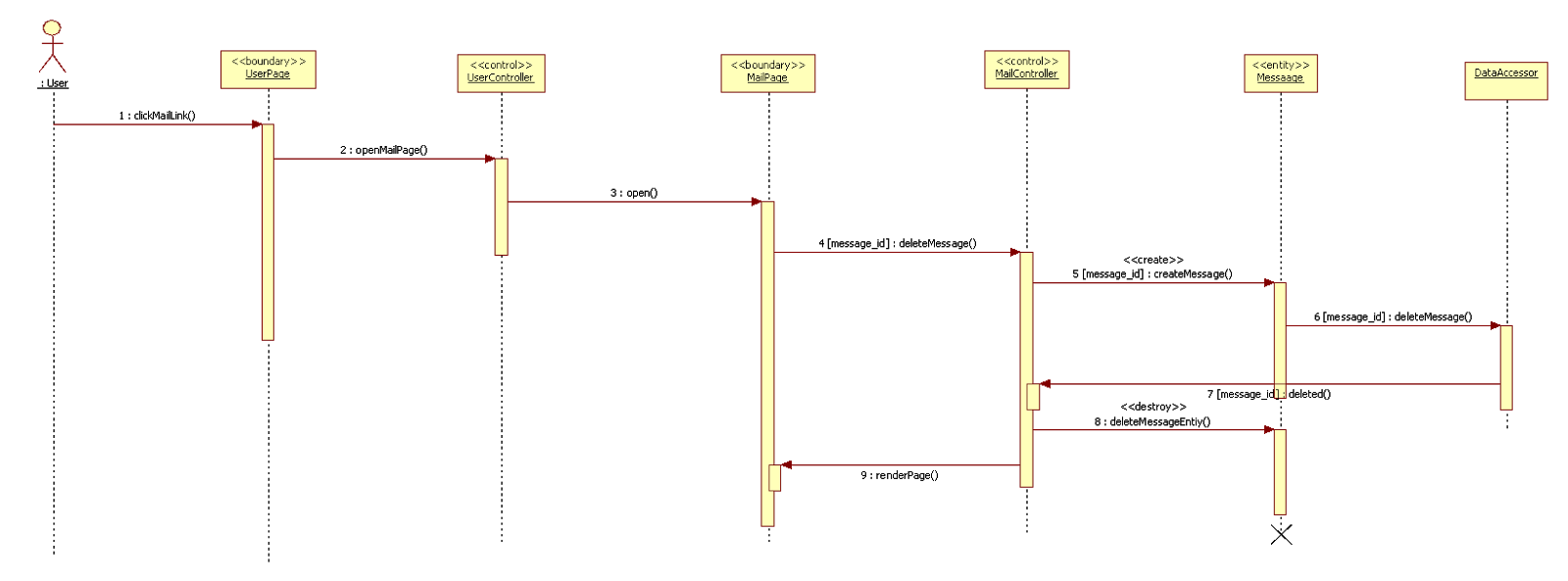
**CP-017 Send new message**



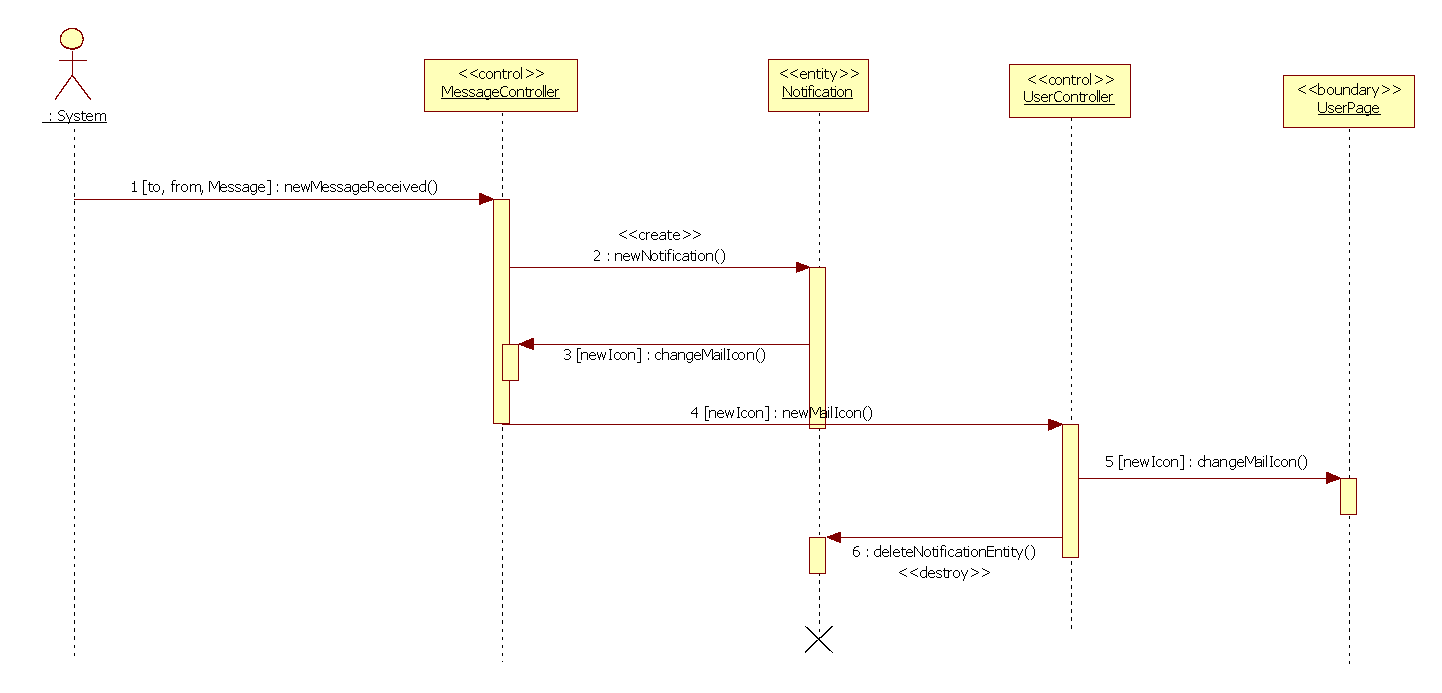
**CP-018 Read message**



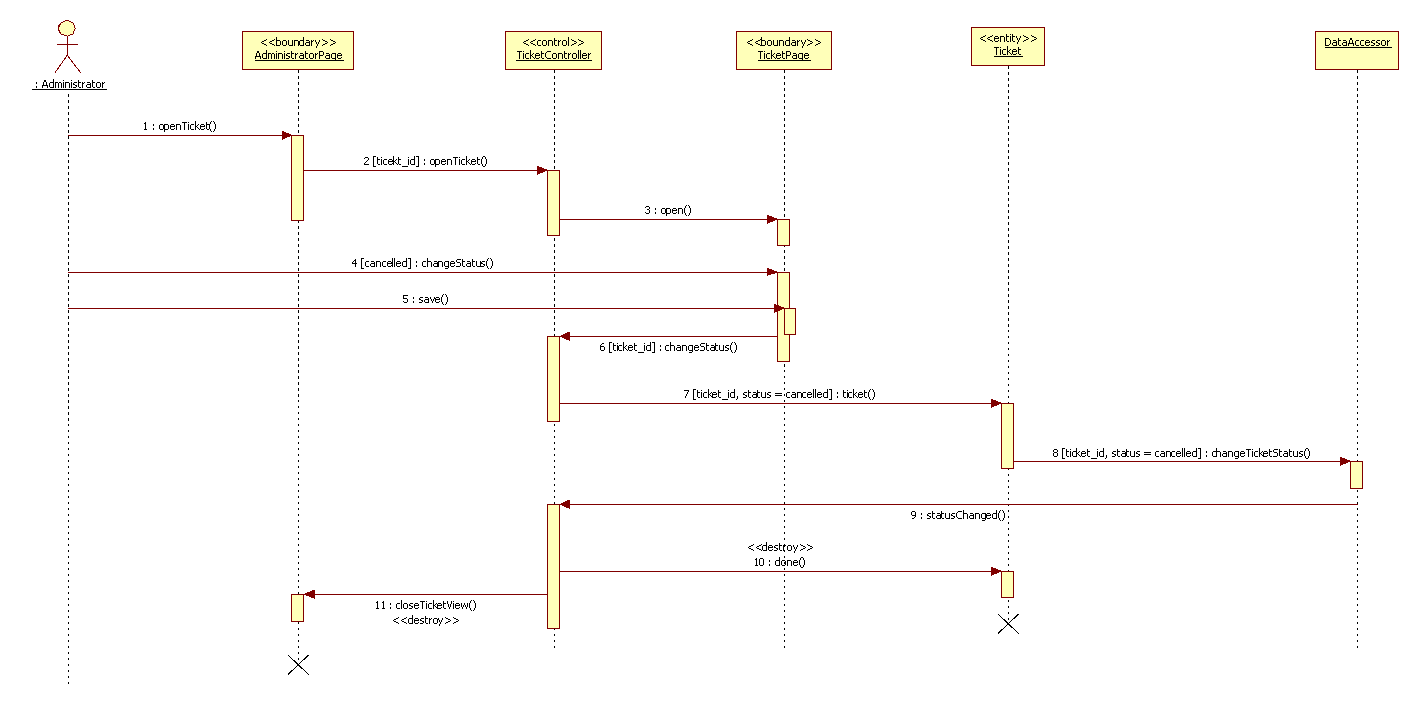
**CP-019 Delete Message**



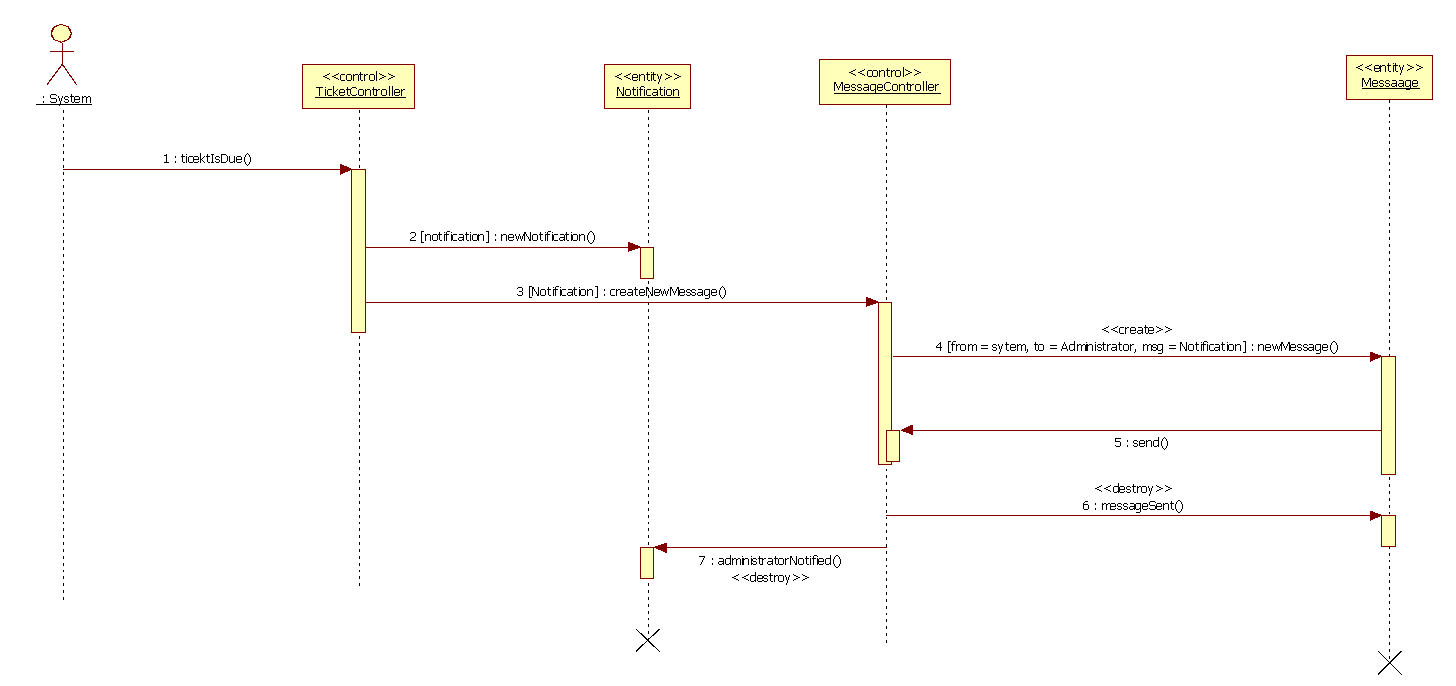
**CP-020 New Message Notification**



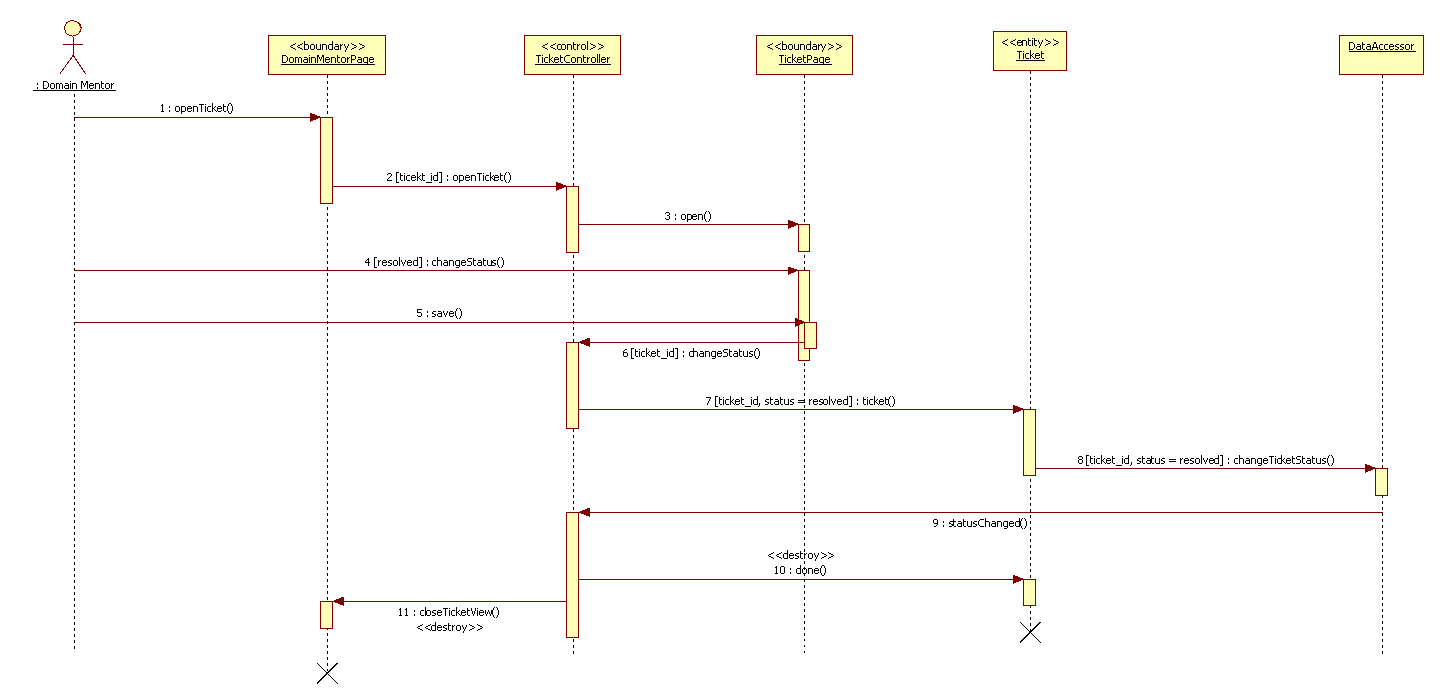
**CP-021 Drop a ticket**



**CP-022 Ticket Due Notification**



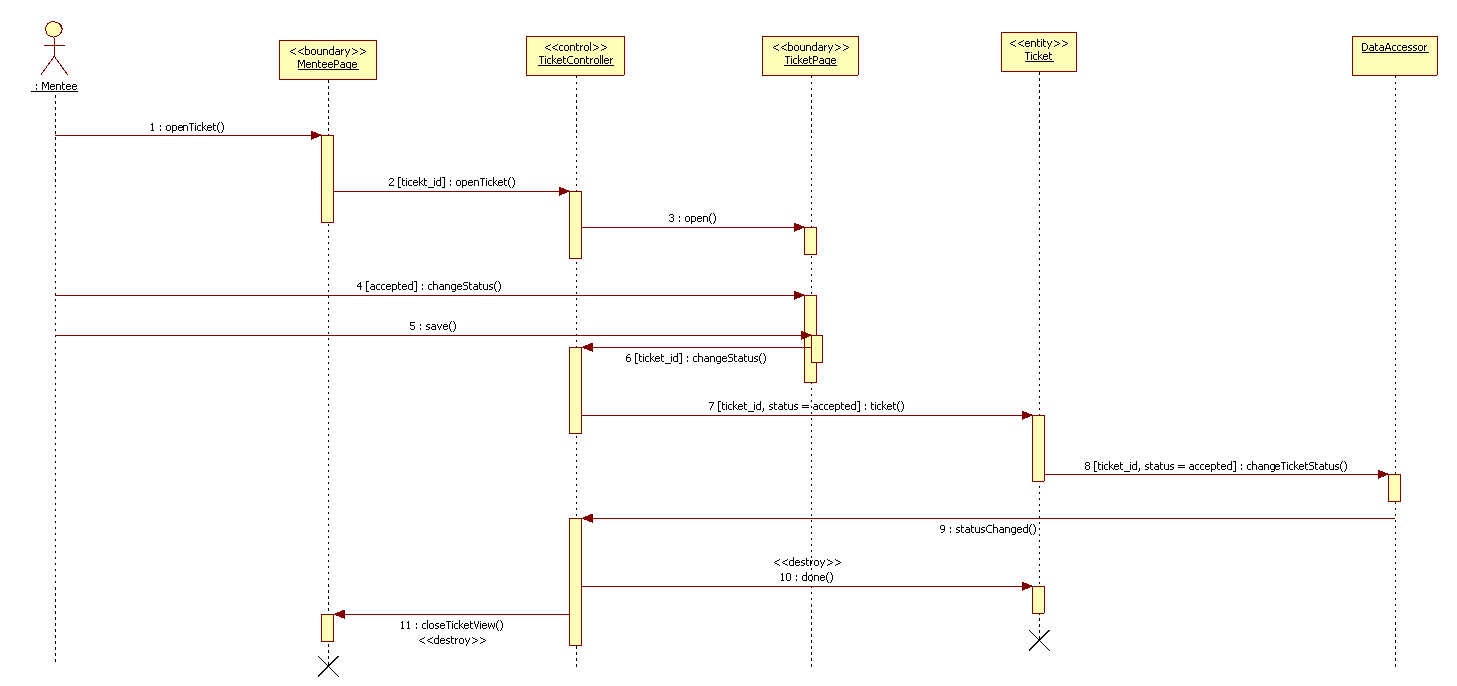
**CP-023 Close a ticket**



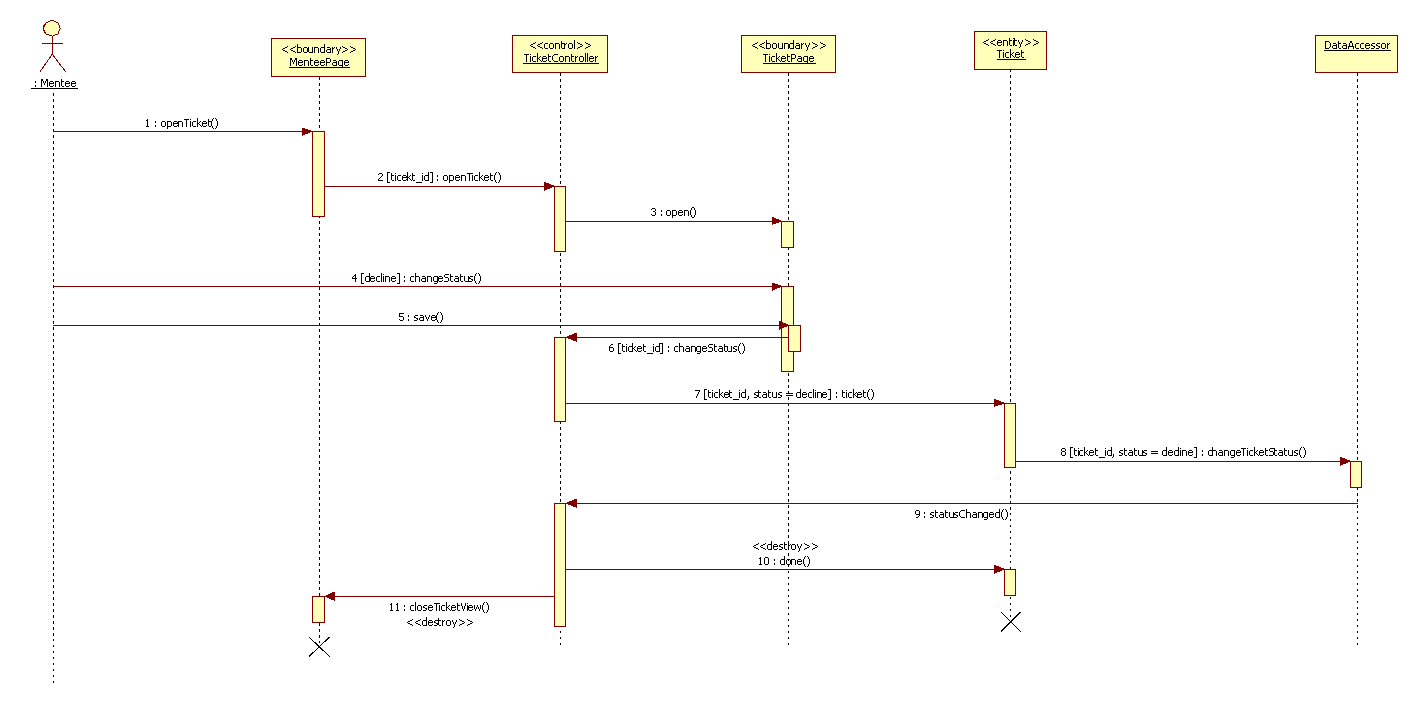
**CP-024 Comment Added Notification**



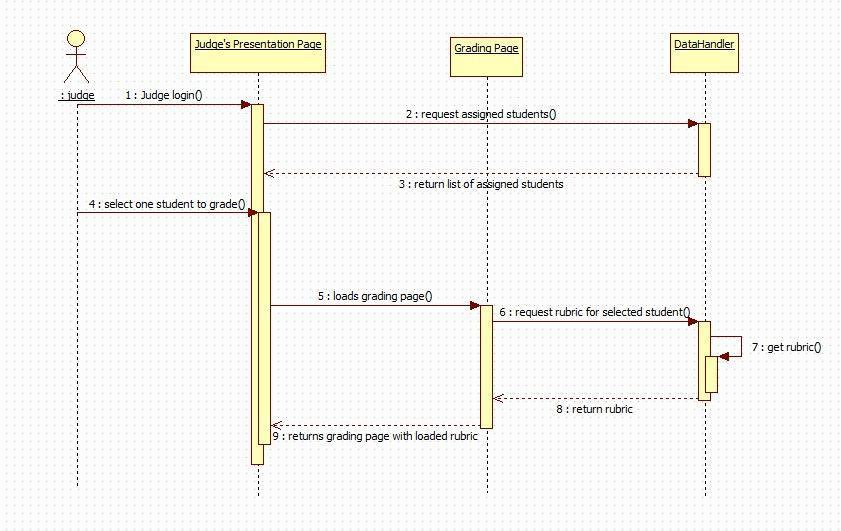
**CP-025 Accept Answer**



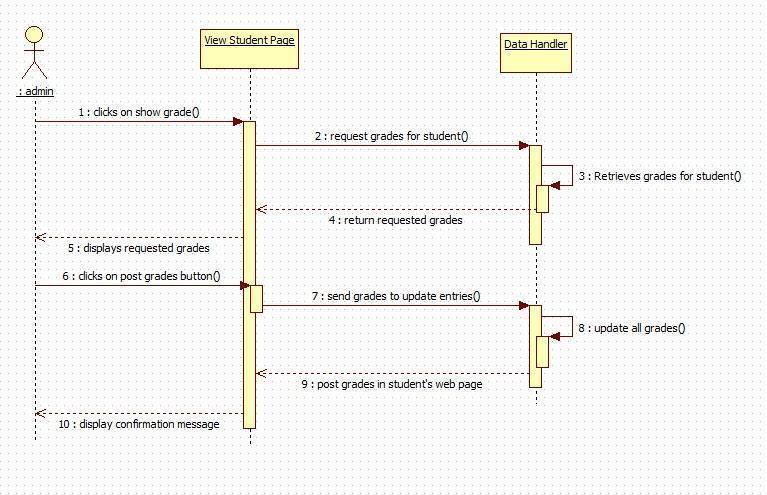
**CP-026 Decline Answer**



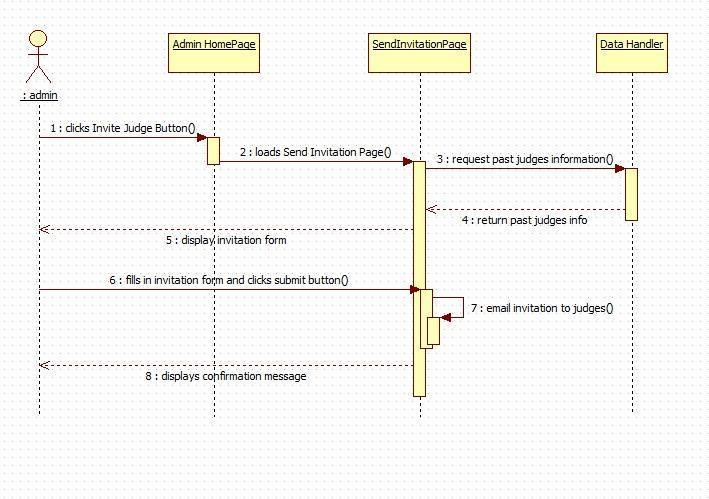
**CP-027 Get Rubric**



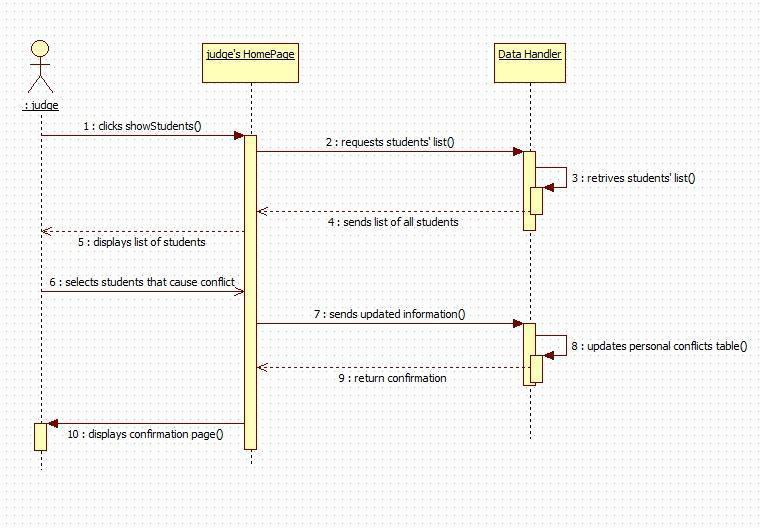
**CP-028 Post Grades**



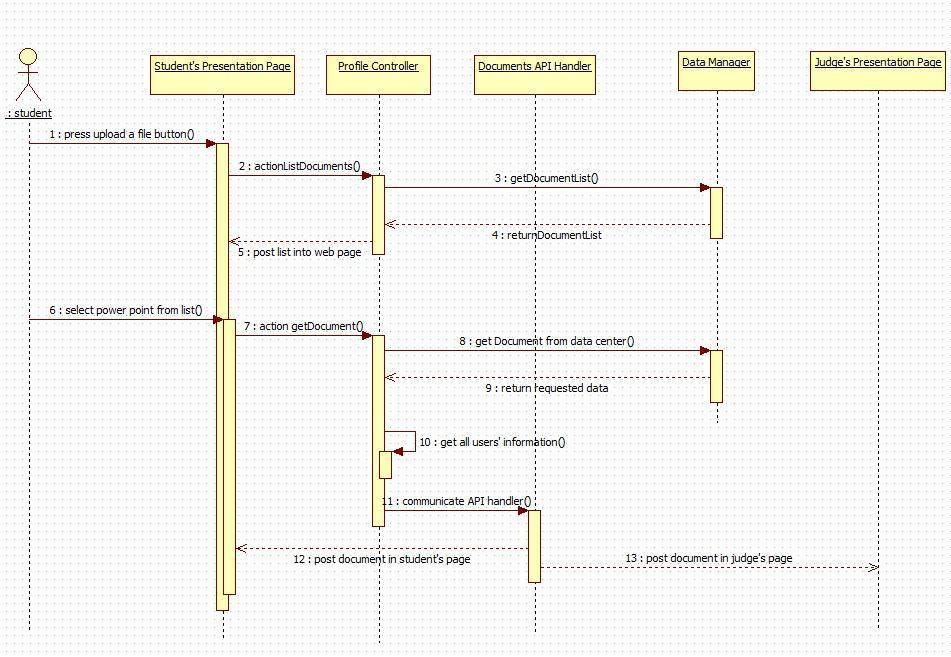
**CP-029 Send Invitation**



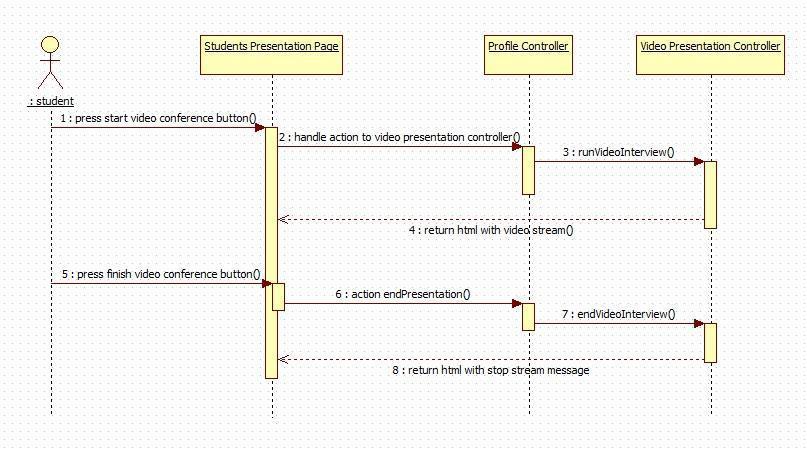
**CP-030 Set student preferences**



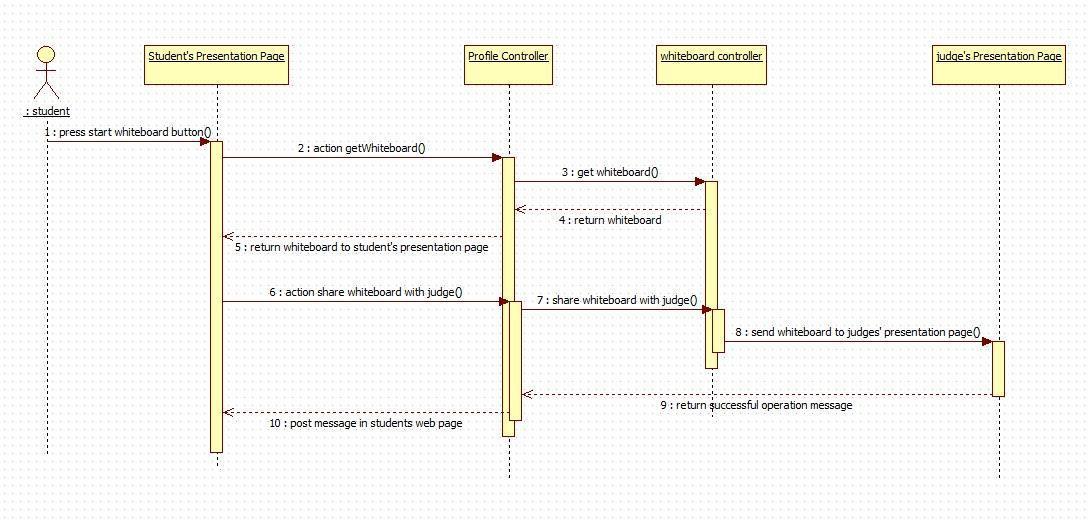
**CP-031 Show Power Point**



**CP-032 Use video conference**

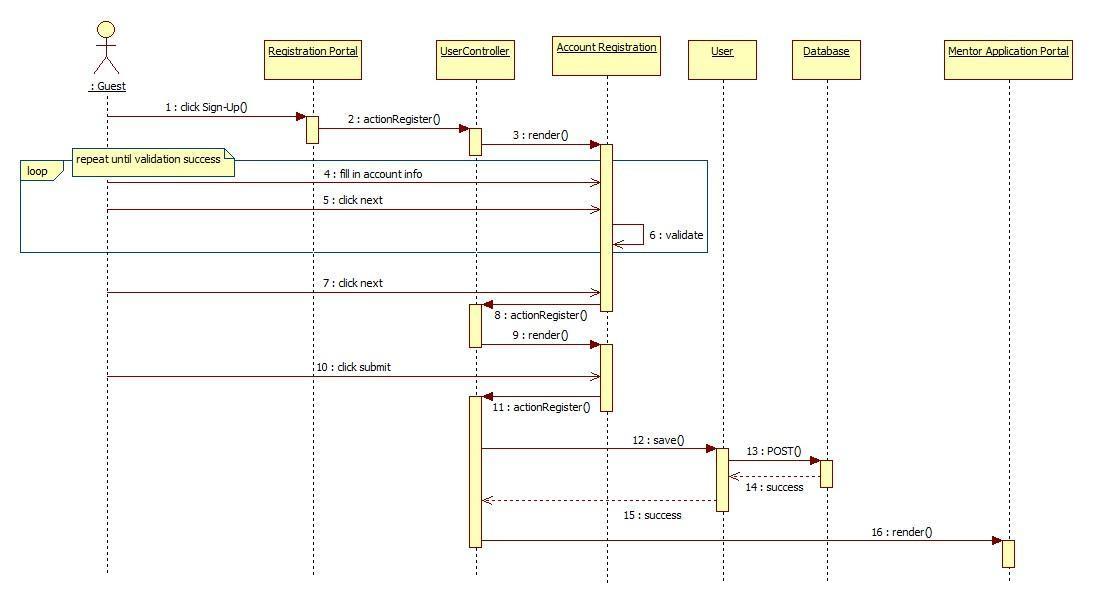


**CP-033 Use Whiteboard**

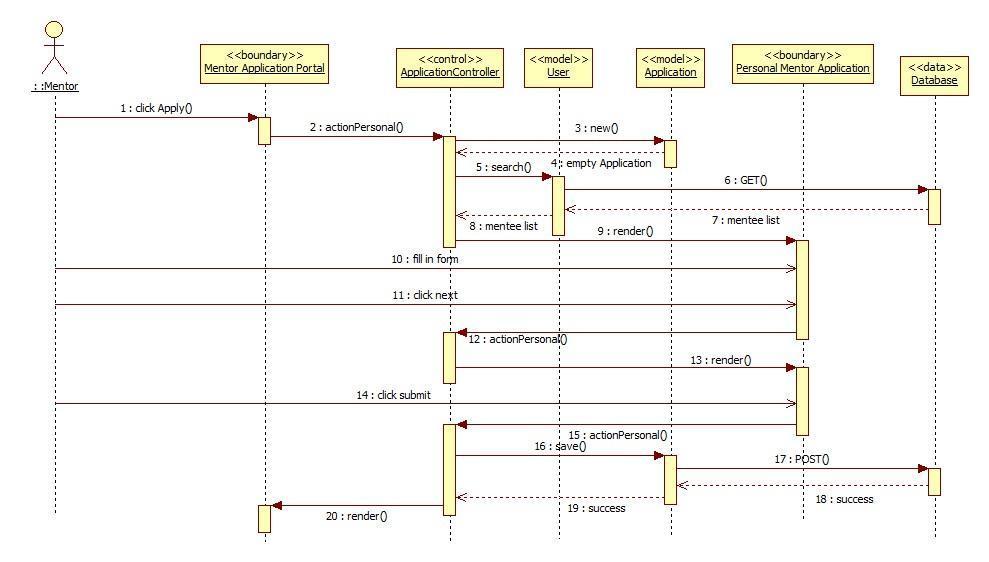


**Sequence Diagrams for Collaborative Platform Version 3**

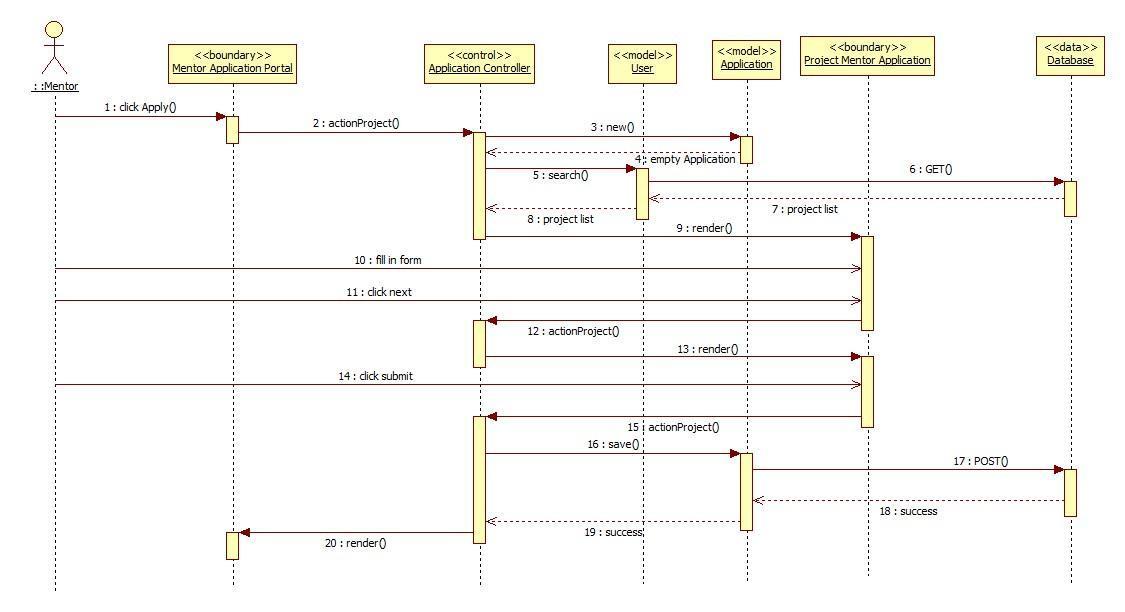
CPv3-001 Self Registration for Collaborative Platform



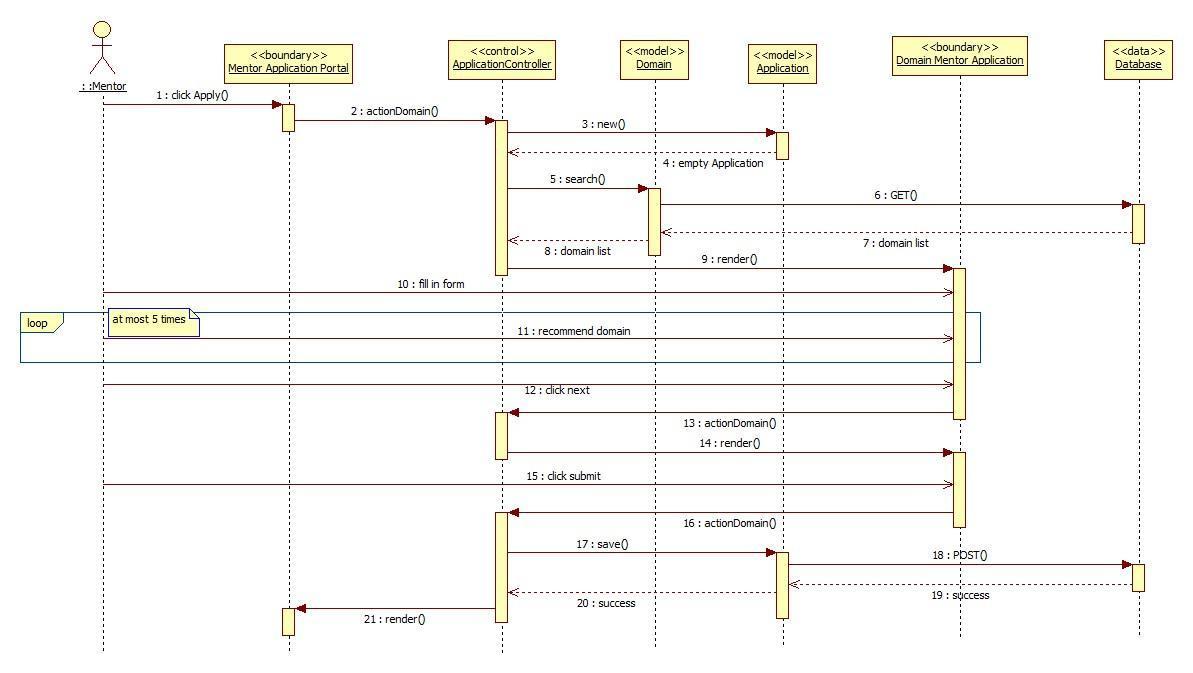
CPv3-002 Apply for Personal Mentorship



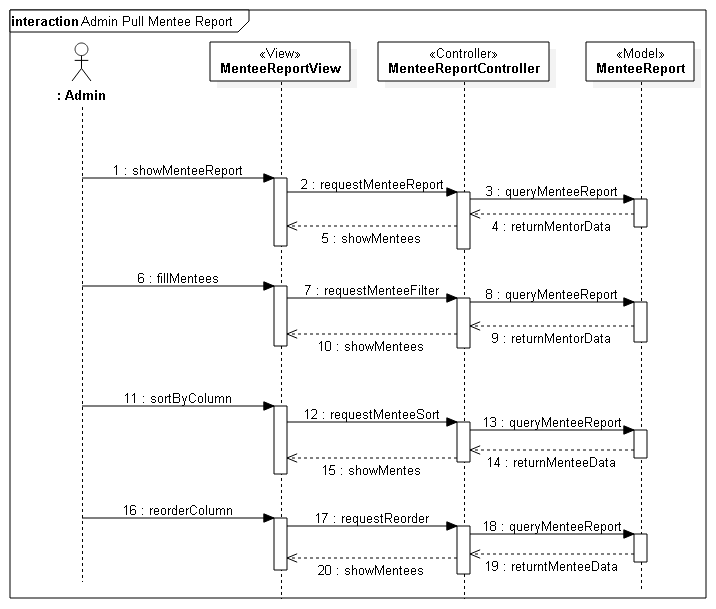
CPv3-004 Apply for Project Mentorship

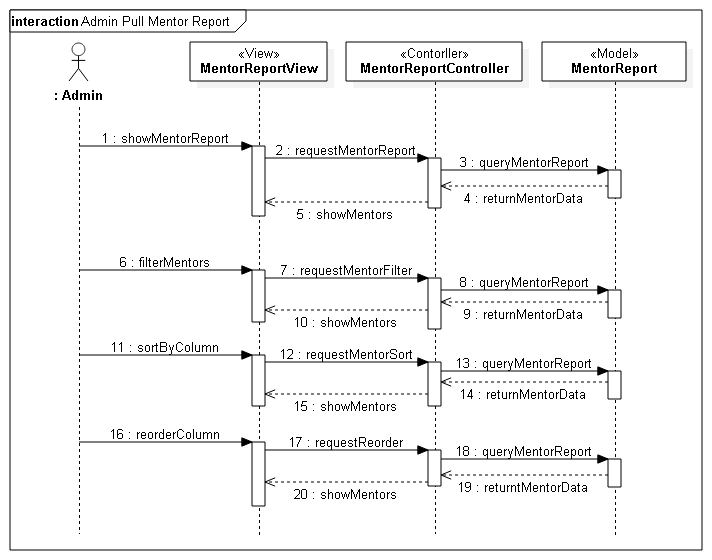


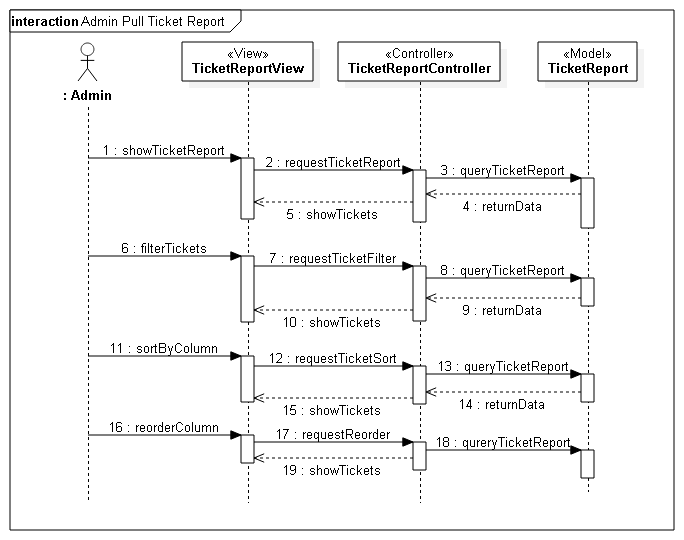
CPv3-006 Apply for Domain Mentorship

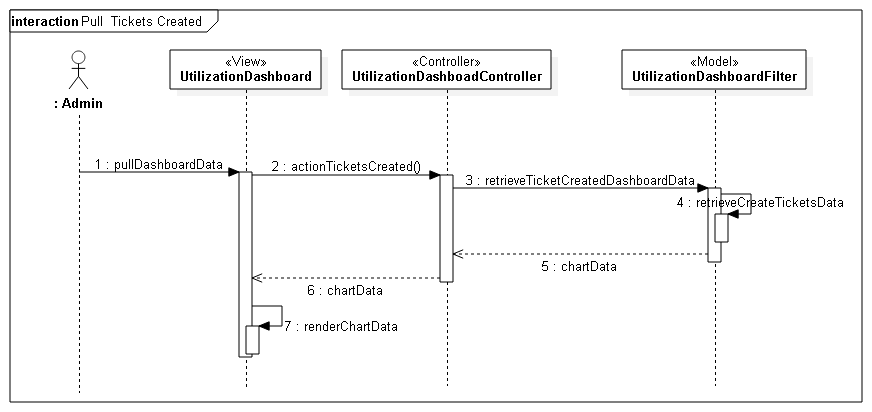


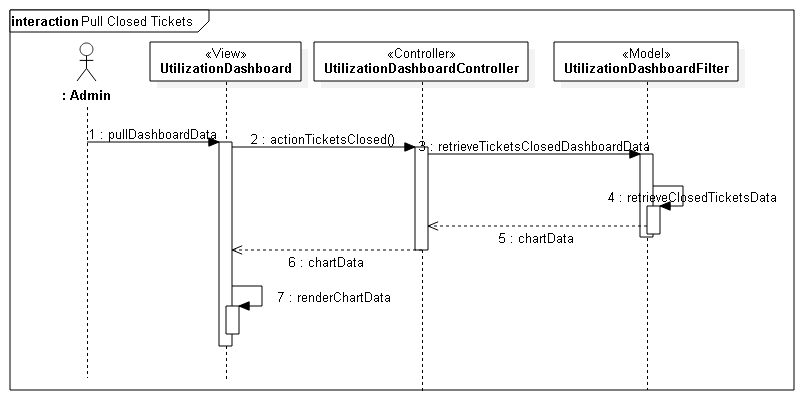
**Sequence Diagrams for Collaborative Platform Version 5**

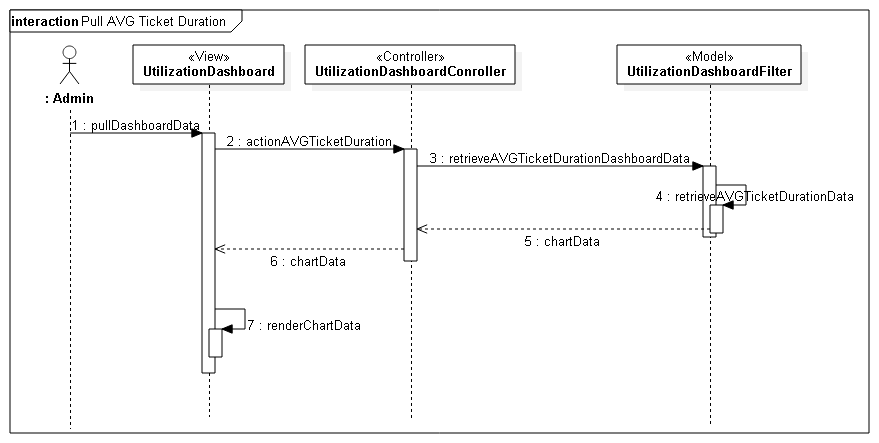


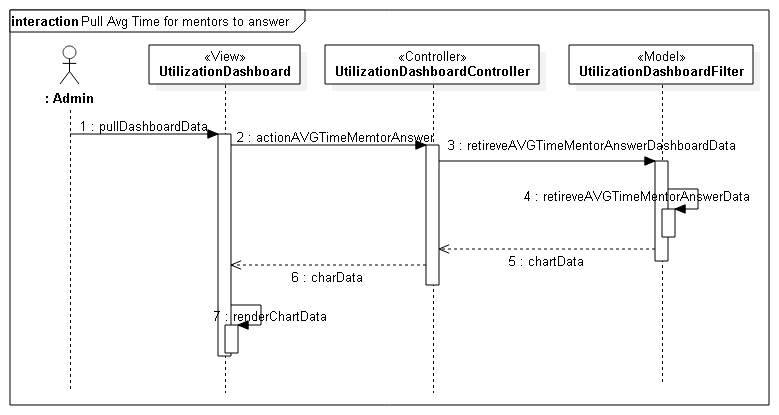


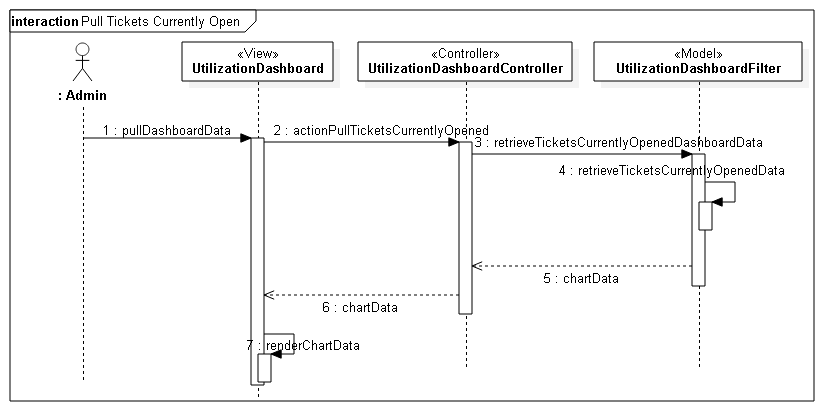


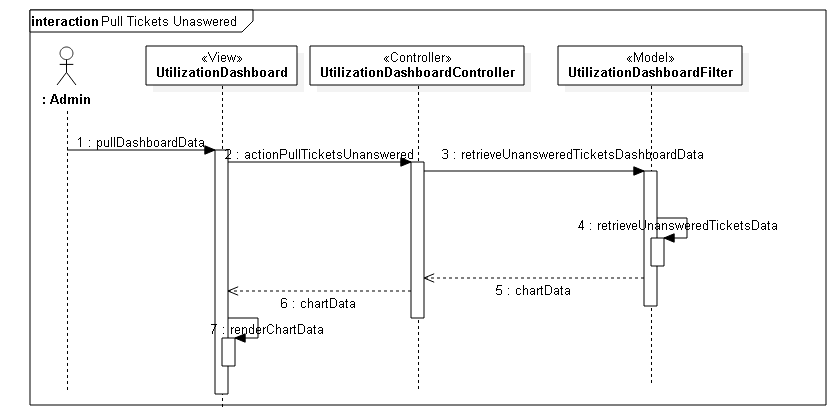


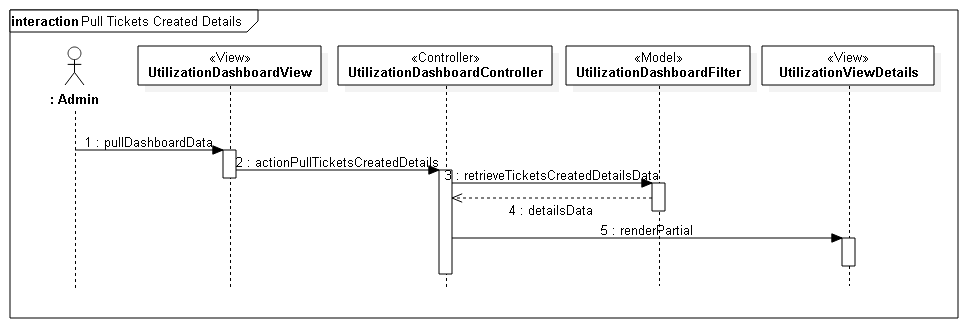


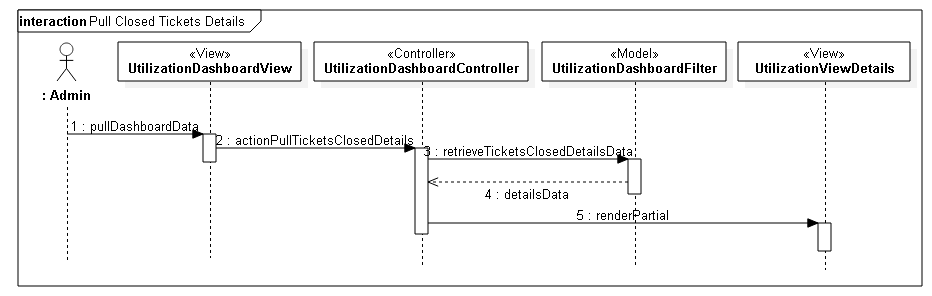


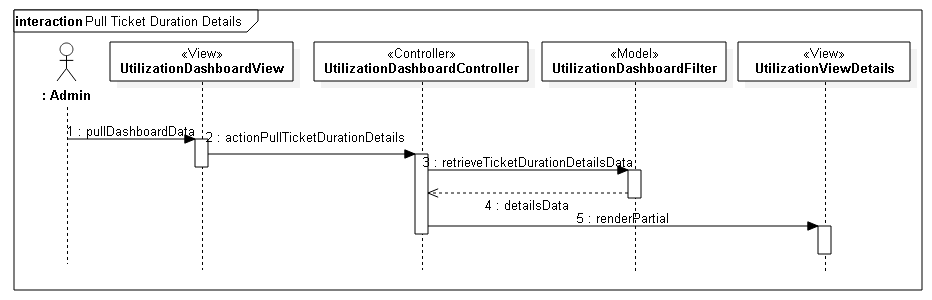




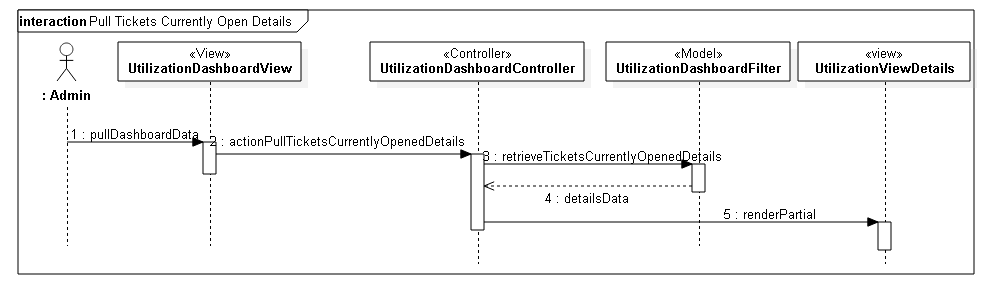


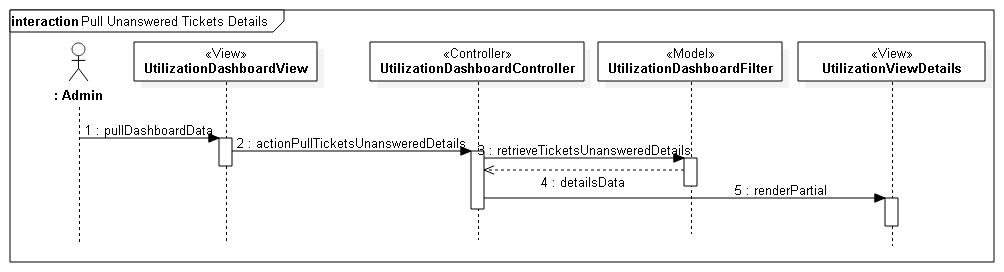


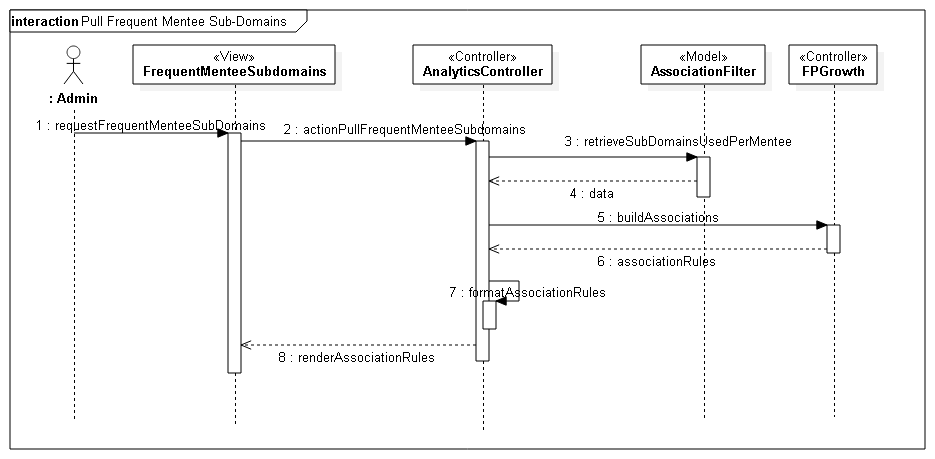


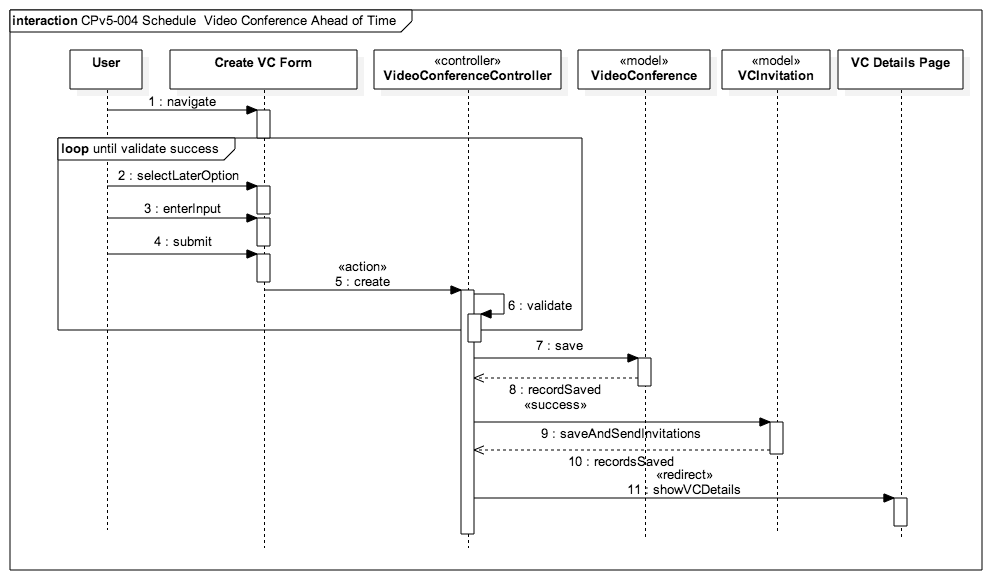


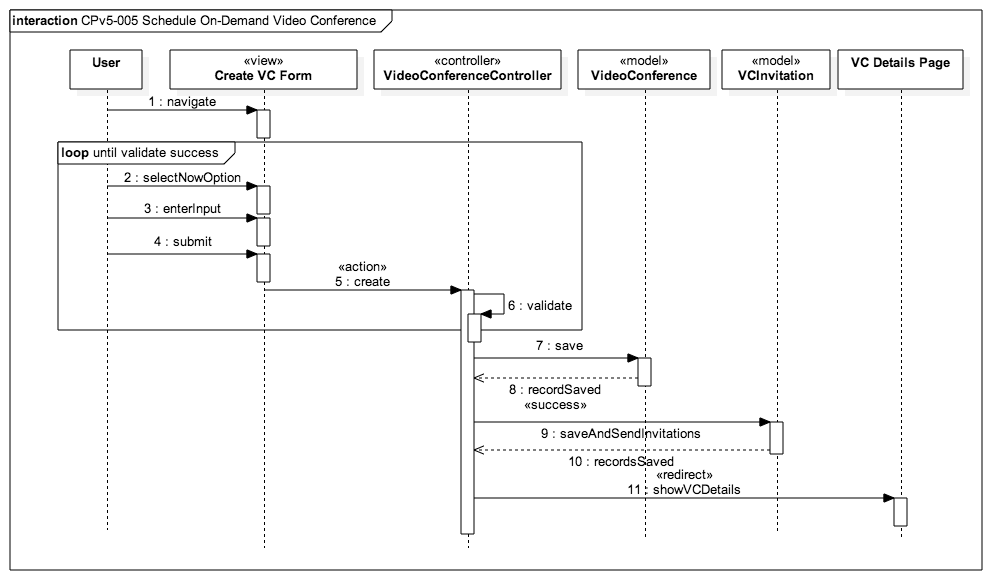


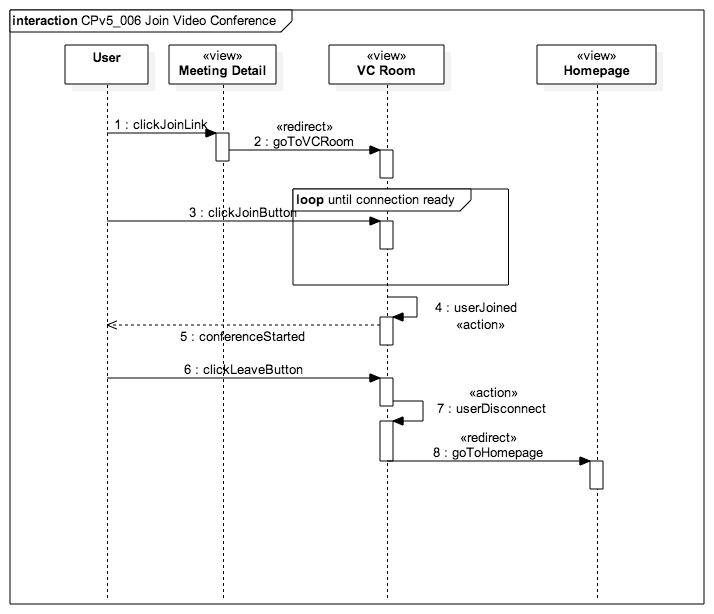


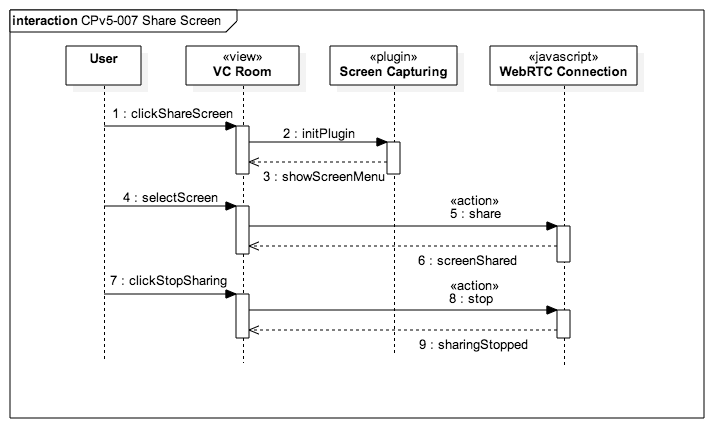




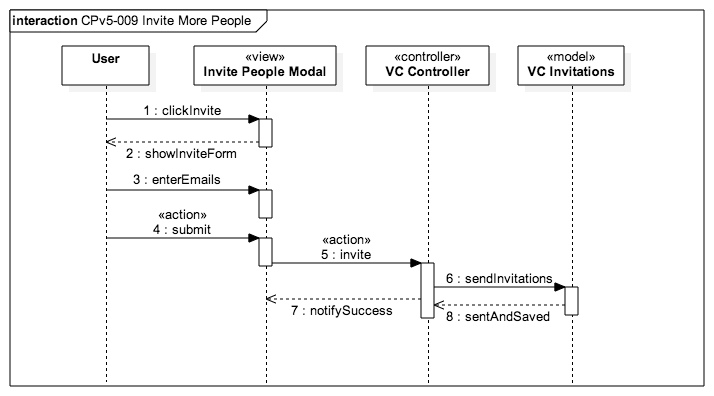


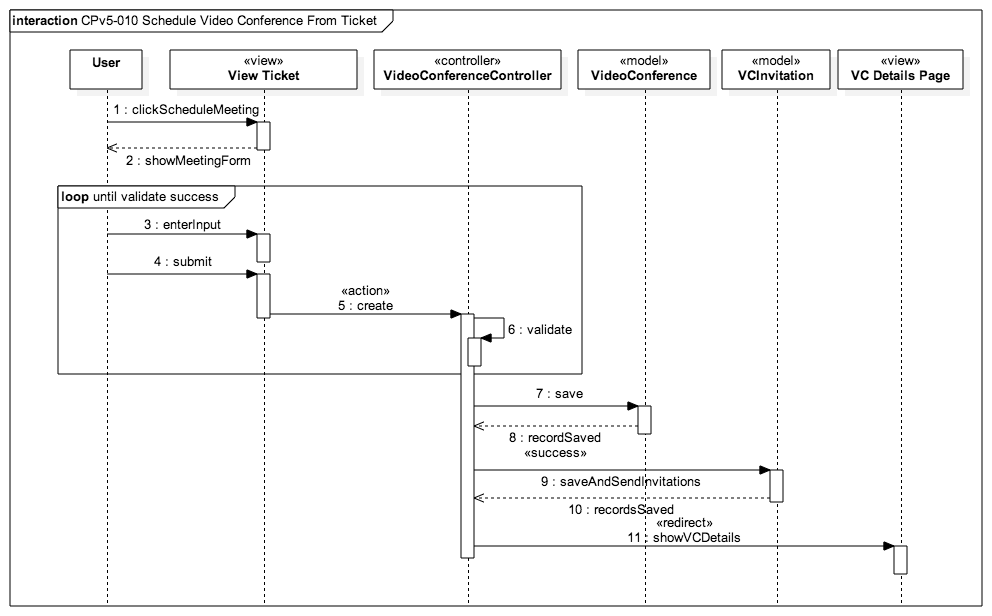
**

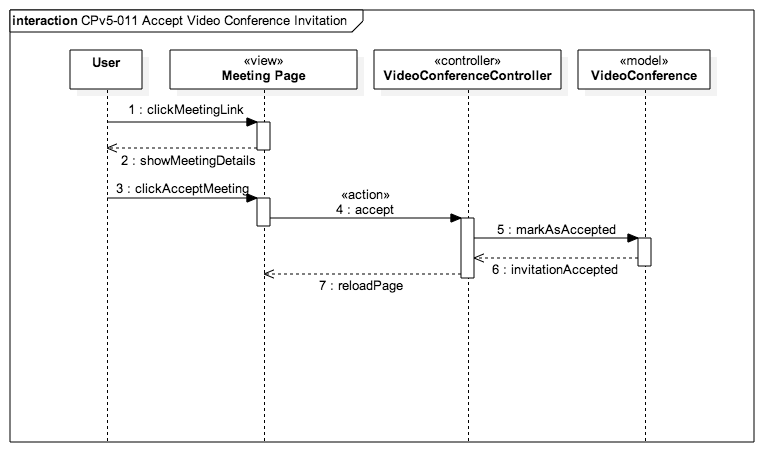


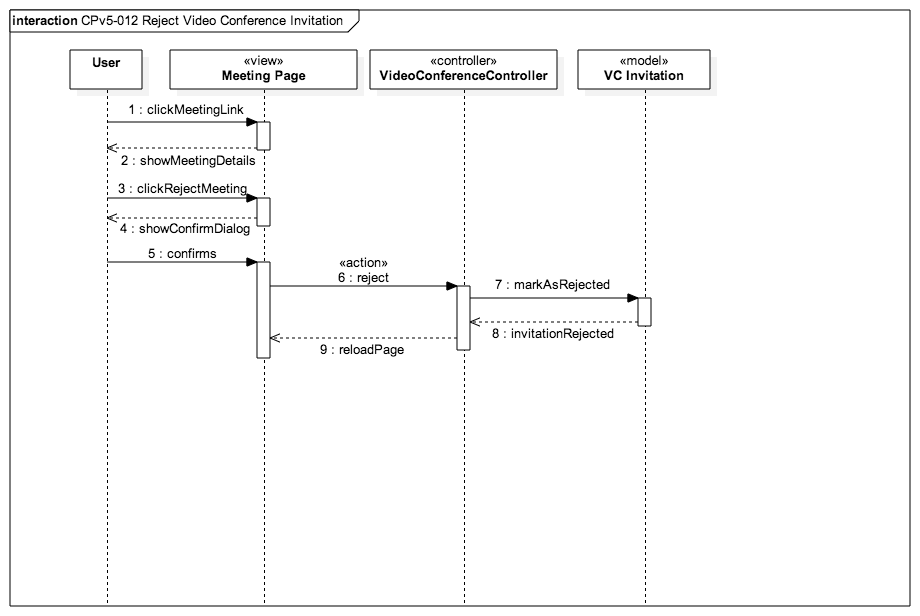
**

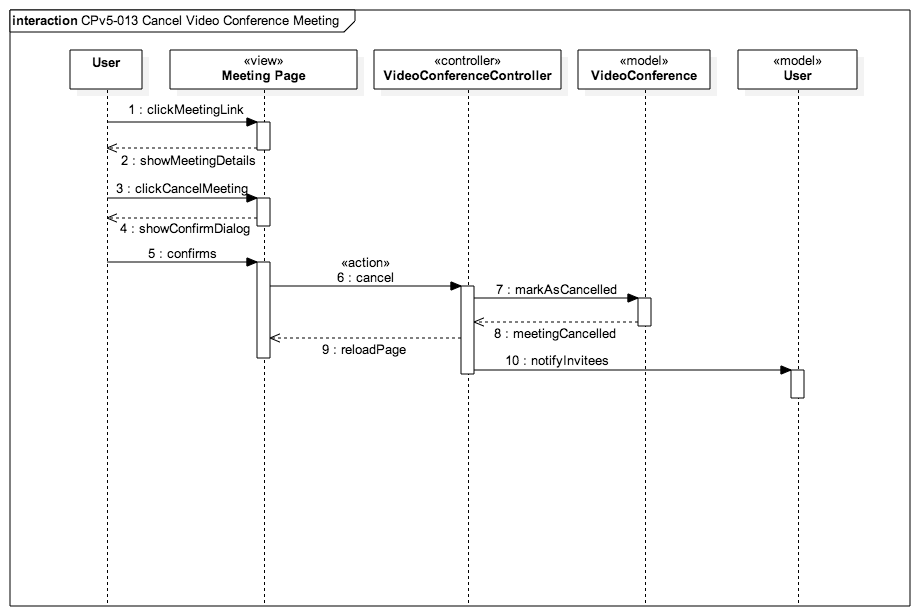












## 3.4 Code Specification

Refer to Appendix C for code interfaces.

# Glossary

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.

Acronyms:

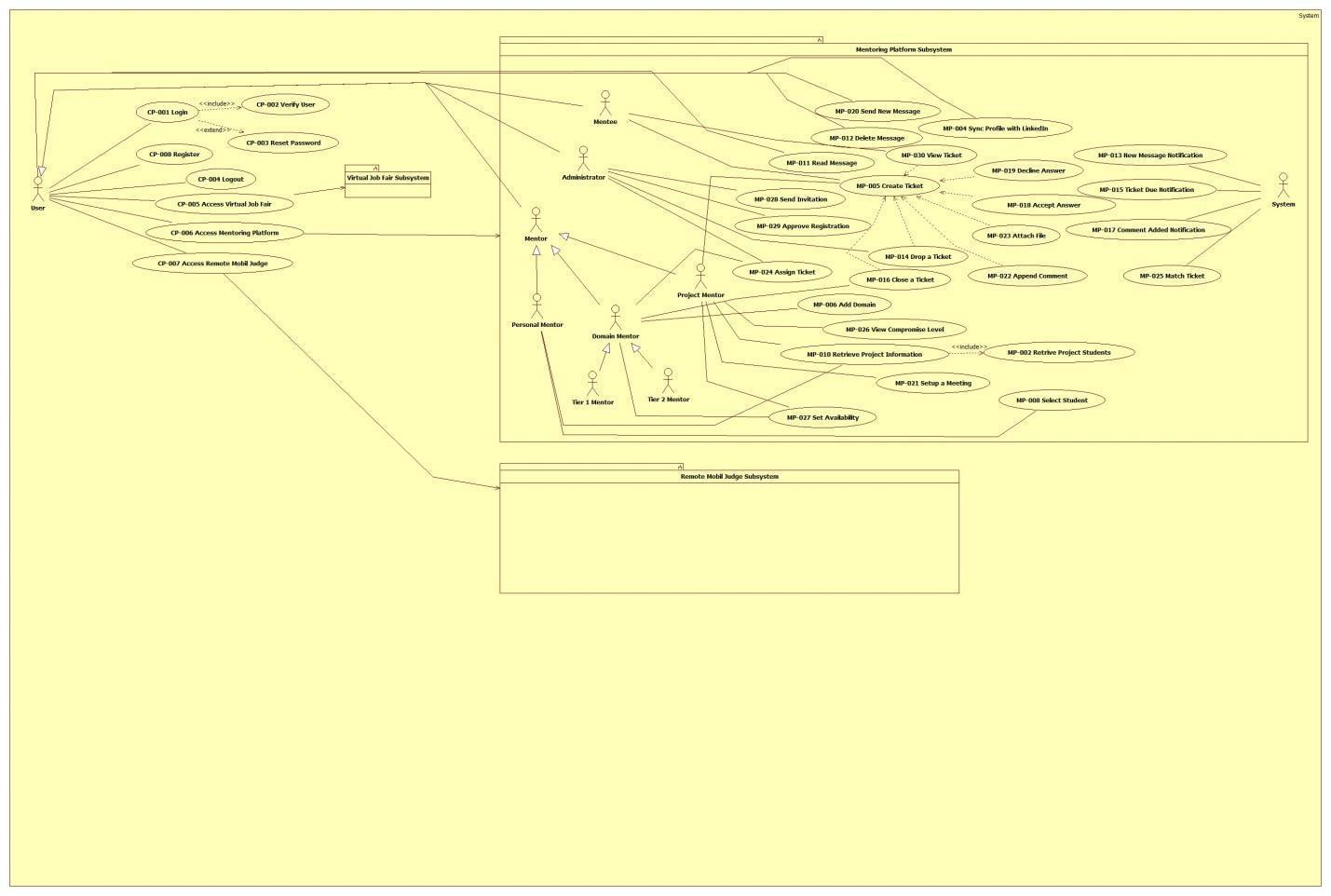
* **SCIS** – School of Computing & Information Sciences
* **FIU** – Florida International University
* **ECS** – Engineering & Computing Sciences (Building)
* **JCCL** – John C. Comfort Laboratory
* **VJF** – Virtual Job Fair
* **MJ** –Mobile Judge
* **WP** – Web Platform
* **T1**- Tier 1 Domain Mentor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

Abbreviations:

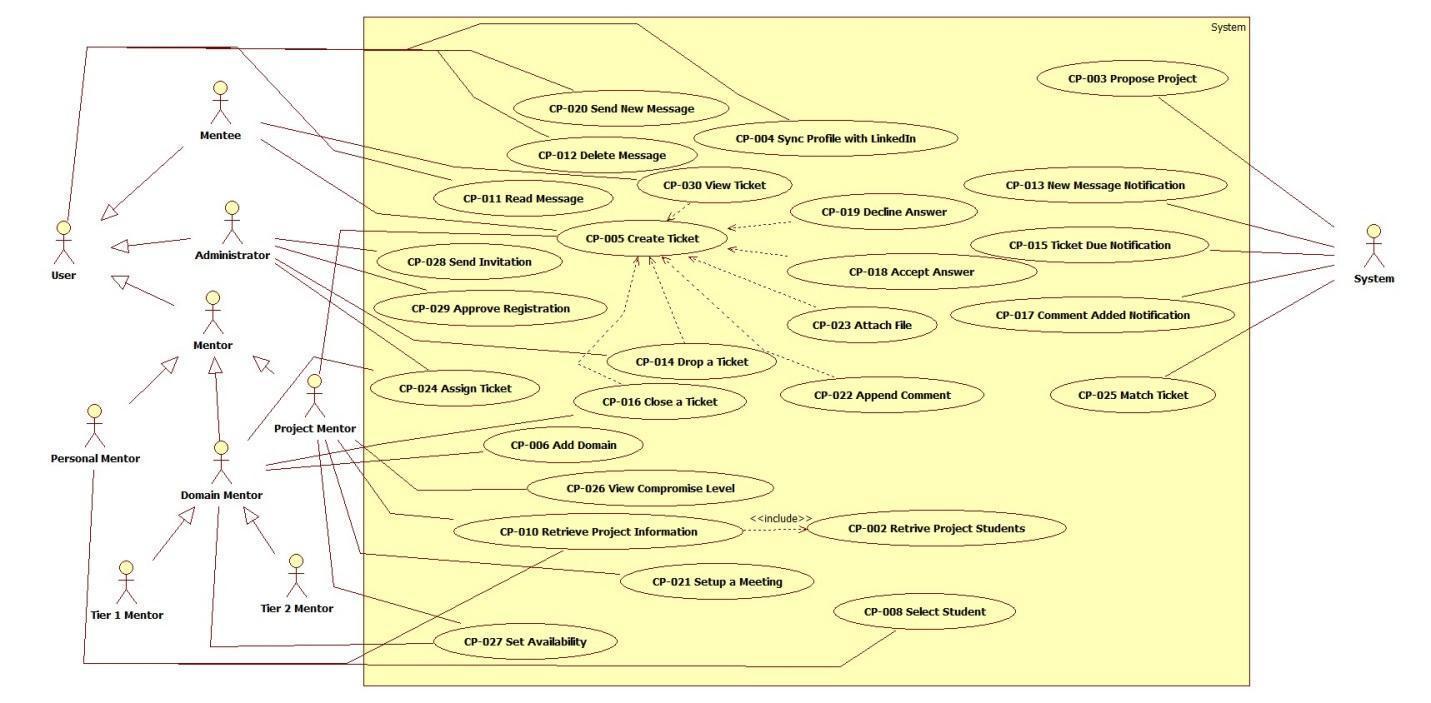
* **N/A**

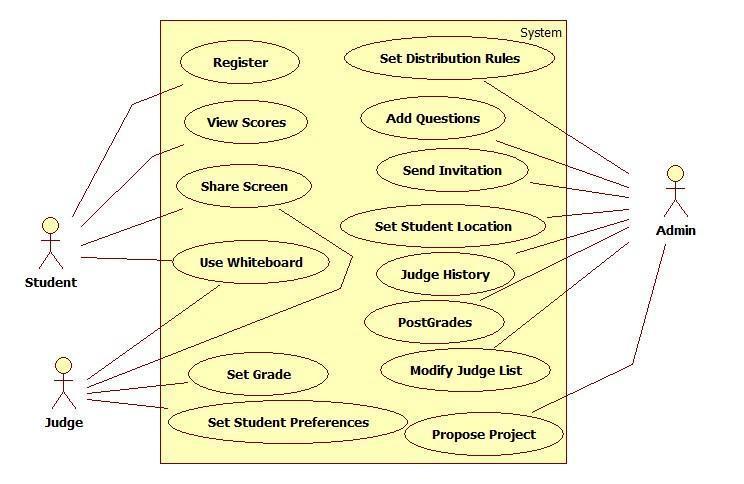
# Appendix

## 5.1 Appendix A – Complete Use Cases Diagrams

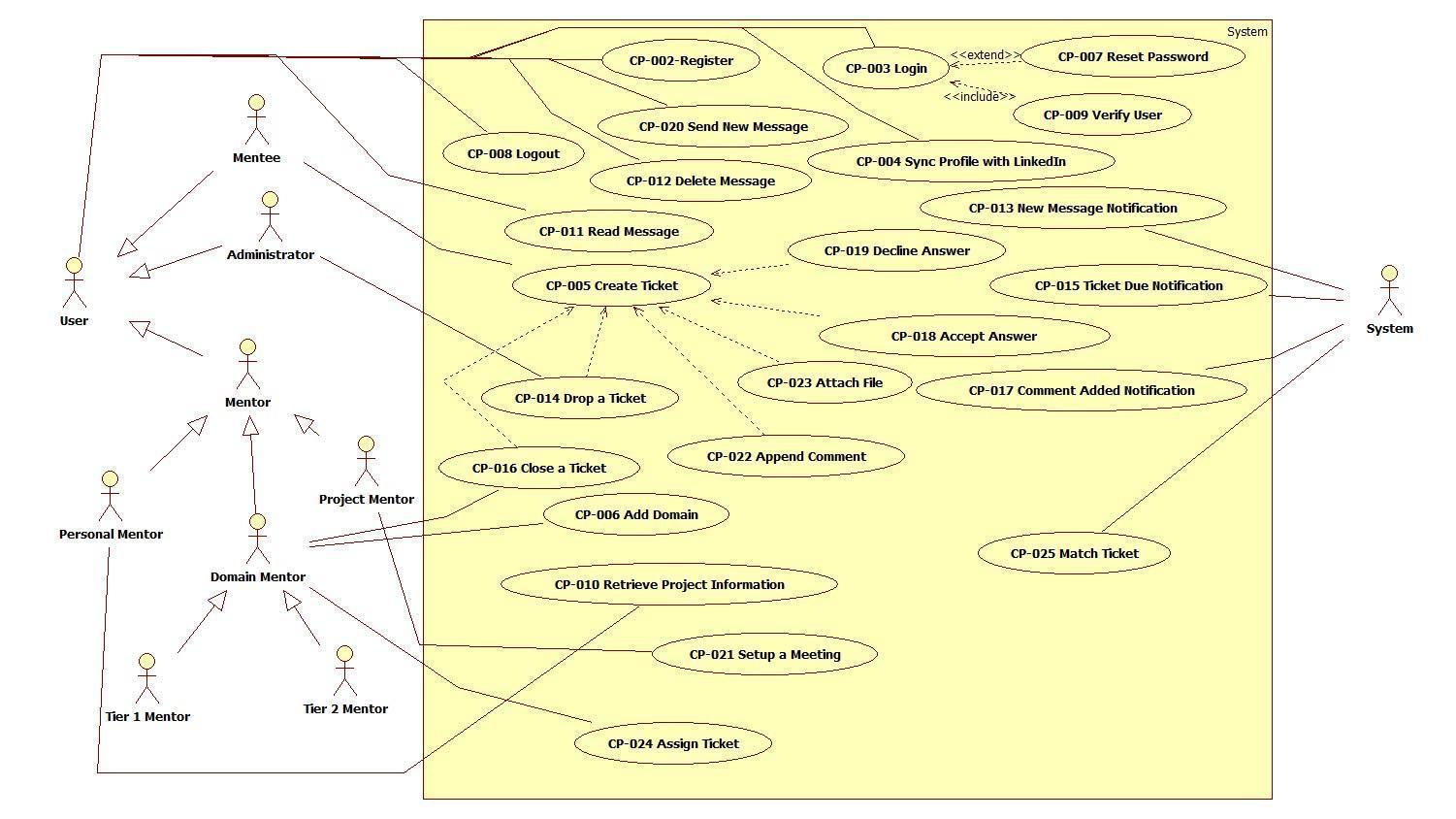


**Figure 2: Collaborative Platform**

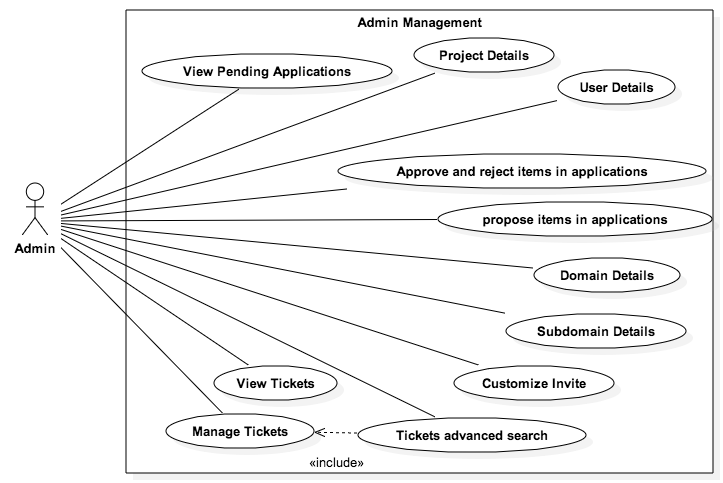


**Figure 3: Mentoring Module**

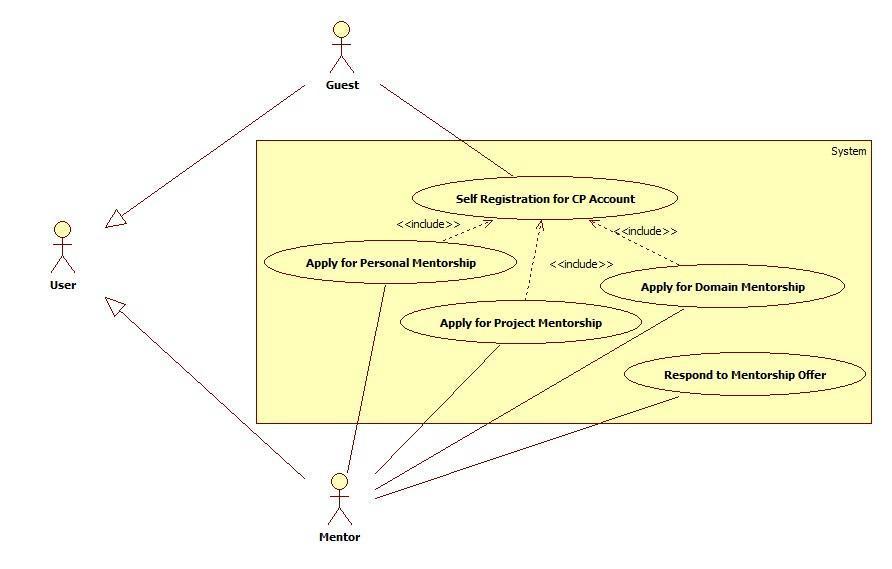
**Figure 4: Remote Judge Module**



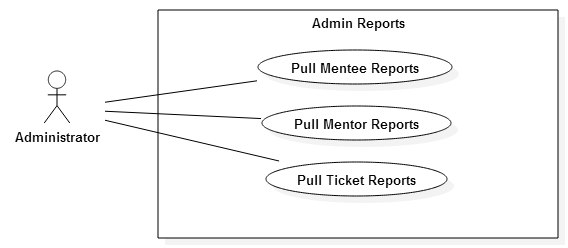
**Figure 5: Admin management**



**Figure 6: Mentor Application**

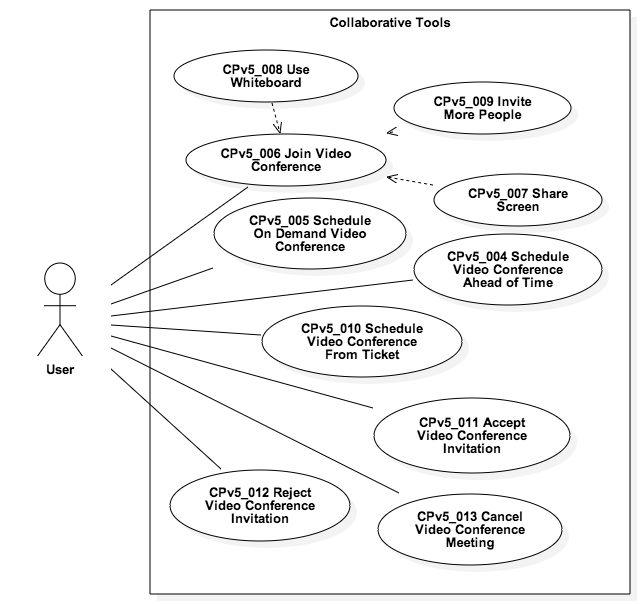


**Figure 7 Admin Reports**



**Figure 8 Utilization reporting and analytics**





## 5.2 Appendix B – Complete Use Cases

|  |  |
| --- | --- |
| **Use Case ID** | **CP-001 Set Up meeting from SPW** |
| **Description** | Allow Project mentor to set up a meeting from the SPW |
| **Actor** | SPW |
| **Pre-conditions** | 1. Project mentor request a meeting with their respective mentees in the SPW |
| **Steps** | 1. The use case begins when the system receives a set up a meeting request from the SPW. 2. The system validates that the information sent is enough to set up a meeting. 3. The system validates if the mentees exist in the system. 4. The system set up a meeting and sends a notification to the users about the new meeting. 5. The use case ends after the notification is send out. |
| **Post-conditions** | 1. The notification will appear In the appointments list for the project mentor and mentee |
| **Exceptions** | No Exceptions |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-002 Start using the white board** |
| **Description** | Allow Project mentor and mentee to start using the white board feature |
| **Actor** | Project mentor, mentees |
| **Pre-conditions** | 1. Projector mentor and mentee are successfully logged onto the collaborative platform. 2. Meeting has been schedule |
| **Steps** | 1. The use case begins when either project mentor or mentee click on the white board button. 2. The system calls the white board feature. 3. The use case finish when project mentor or mentee close the white board |
| **Post-conditions** | 1. User is able to share the white board |
| **Exceptions** | User tries to share the white board while is used by other. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-003 Participate in a video conference** |
| **Description** | Allow Project mentor and mentee to participate in a video conference |
| **Actor** | Project mentor, mentees |
| **Pre-conditions** | 1. Projector mentor and mentee are successfully logged onto the collaborative platform. 2. Meeting has been schedule |
| **Steps** | 1. The use case begins when either project mentor or mentee click on the video conference button. 2. The system calls the video conference feature. 3. The use case finish when project mentor or mentee close the video conference |
| **Post-conditions** | 1. The User will be able to perform video conference |
| **Exceptions** | No Exceptions |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-004 Share Files** |
| **Description** | Allow Project mentor and mentee share files |
| **Actor** | Project mentor, mentees |
| **Pre-conditions** | Projector mentor and mentee are logged into the project mentoring module.  Meeting has been schedule |
| **Steps** | 1. The use case begins when either project mentor or mentee click on the file share button. 2. The system will call the file system. 3. The user has to select from the file system the file that He/she wants to share. 4. The user click on the Ok button. 5. The use case ends after the file has been sent. |
| **Post-conditions** | 1. User share a file |
| **Exceptions** | User tries to share a file while other user is using the file sharing feature |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-005 Register User** |
| **Description** | Allow the user to register |
| **Actor** | User |
| **Pre-conditions** | User is at the registration screen |
| **Steps** | 1. The use case begins when the user clicks on the register button.  2. The system will call to open registration screen.  3. The user will enter the registration info required.  4. The user will click the confirm button.  5. The system will send a request to register.  6. The system will respond with a confirmation popup.  7. The user will confirm by clicking yes.  8. The use case ends when the registration is sent to the database. |
| **Post-conditions** | User is registered |
| **Exceptions** | User is already registered via email provided. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-006 User Login** |
| **Description** | Allow the user to login |
| **Actor** | User |
| **Pre-conditions** | User is a the home page screen |
| **Steps** | 1. The use case begins when the user clicks on the login button.  2. The system will to open the login screen.  3. The user will fill in their credentials  4. The user will then click the sign in button.  5. The system will send a request to verify credentials.  6. The user case ends when the user is logged in. |
| **Post-conditions** | User is logged into the system |
| **Exceptions** | User enters invalid inputs |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-007 User Logout** |
| **Description** | Allow the user to logout |
| **Actor** | User |
| **Pre-conditions** | User is logged in |
| **Steps** | 1. The use case begins when the user clicks on the logout button.  2. The system will send a request to logout and ask for confirmation.  3. The user will click yes  4. The system will end the session upon confirmation.  5. The system will redirect to the main page.  6. The use case ends when the user is logged out. |
| **Post-conditions** | User is logged out of the system. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-008 Reset Password** |
| **Description** | Allow the user to reset password |
| **Actor** | User |
| **Pre-conditions** | User is at login screen |
| **Steps** | 1. The use case begins when the user clicks on reset password button.  2. The system will send a request to reset  3. The user confirm security questions to their account  4. The system will verify credentials  5. The user will input a new password  6. The system will save the new password and send to the database  7. The use case ends when the user has a new password. |
| **Post-conditions** | User has new password stored in database |
| **Exceptions** | User does not verify security questions |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-009 Add Domain** |
| **Description** | Allow the domain mentor to add a new domain |
| **Actor** | Domain Mentor |
| **Pre-conditions** | Domain Mentor is logged in and at their home profile page |
| **Steps** | 1. The use case begins when the domain mentor clicks on the add domain button.  2. The system will request the domain addition pop up.  3. The domain mentor then selects a domain or types a new one.  4. The system responds with an expertise box.  5. The domain mentor selects the appropriate expertise level for the new domain.  6. The domain mentor then clicks the add button.  7. The system requests approval and shows a confirmation screen.  8. The domain mentor confirms and clicks yes.  9. The system requests the domain to be stored for approval  10. The use case ends when the domain is added to the users table. |
| **Post-conditions** | Domain is pending approval on users domain table. |
| **Exceptions** | Domain is already on list, domain does not get approved. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-010 Retrieve project information from the SPW** |
| **Description** | Allow project info to be retrieved from SPW |
| **Actor** | Project Mentor |
| **Pre-conditions** | Project Mentor is logged in and at Project View screen |
| **Steps** | 1. The use case begins when the project mentor clicks on the Retrieve project info button.  2. The system sends a request and asks for confirmation.  3. The project mentor confirms and clicks yes.  4. The system establishes a link and requests a fetch of information from SPW.  5. The project info is then saved into the database  6. The use case ends when the project info is updated and displayed on the mentors screen. |
| **Post-conditions** | Project info has been successfully retrieved from SPW |
| **Exceptions** | Project not on SPW |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-011 Create a question** |
| **Description** | Allow mentee to create a ticket |
| **Actor** | Mentee |
| **Pre-conditions** | Mentee is logged in and at the mentee screen. |
| **Steps** | 1. The use case begins when the mentee clicks on the create ticket button.  2. The system requests to show the ticket creation view.  3. The mentee fills out the information on the ticket.  4. The mentee hits the create button.  5. The system requests the ticket be stored in the database.  6. The use case ends when a success popup for the ticket creation is shown. |
| **Post-conditions** | Ticket is created and stored in the database. |
| **Exceptions** | N/A |
| **Use Case ID** | **CP-006 User Login** |
| **Description** | Allow the user to login |
| **Actor** | User |
| **Pre-conditions** | The user is at the login screen |
| **Steps** |  |
| **Post-conditions** | The user is logged into the system in their appropriate view. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-012 Retrieve skill-set/domain expertise from their Linkedln account** |
| **Description** | Allow users to sync their profile page with LinkedIn |
| **Actor** | User |
| **Pre-conditions** | User is on the second page of registration for their specific profile |
| **Steps** | 1. The use case begins when the user clicks on the sync profile with LinkedIn button.  2. The system then request a confirmation popup.  3. The user confirms and clicks yes.  4. The system requests information from the LinkedIn API.  5. The information is stored is then stored into the database.  6. The use case ends when the users profile is updated with their respective LinkedIn information. |
| **Post-conditions** | Users information has been retrieved from LinkedIn |
| **Exceptions** | User does not have LinkedIn, user inputs incorrect credentials |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-012 Matching Tickets** |
| **Description** | Allow matching the ticket with the most relevant domain mentor |
| **Actor** | System |
| **Pre-conditions** | 1. Mentee is successfully logged onto the collaborative platform. 2. Mentee has been clicked the button new ticket. |
| **Steps** | 1. The uses case begins when the mentee click the create button on the new ticket window. 2. The system shall verified if any field related to filter has been selected. 3. If any of the option was selected, then the system shall search in the database for most relevant domain mentor to answer the question. 4. The system shall create a notification box notifying if the operation was successfully complete 5. The use case finish when the system returns the result of the search |
| **Post-conditions** | 1. The ticket can be created and assign to the domain mentor |
| **Exceptions** | The mentee typed in other fields a topic, which any domain mentor is related to. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-014 Assign Ticket** |
| **Description** | Allow domain mentor to delegate the ticket to other domain mentor |
| **Actor** | Domain mentor |
| **Pre-conditions** | 1. Domain mentor is successfully logged onto the collaborative platform. 2. Ticket is assigned to a Domain mentor 3. The domain mentor opened the ticket assigned from the tickets queue |
| **Steps** | 1. The use case begins when the domain mentor clicks on the button delegate question. 2. The system shall to create a new box with a list of all domain mentors in the correspondent queue. 3. The domain mentor selects the domain mentor who will be delegate the question. 4. The domain mentor clicks on the Ok button to send the ticket. 5. The system shall create a box with the notification that the delegate process was successfully done. 6. The use case ends when the domain mentor clicks Ok in the notification box. |
| **Post-conditions** | 1. The ticket is removed from the domain mentors queue list 2. The system sends a notification to the domain mentor that a new ticket was assigned and needs his attention. |
| **Exceptions** | The domain mentor can click on the cancel button. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-015 Append comment to a ticket** |
| **Description** | Allow project mentor and mentees to append comments to a ticket |
| **Actor** | Project Mentor, Mentee |
| **Pre-conditions** | 1. Both users are successfully logged onto the collaborative platform system. 2. The ticket was already created. 3. The ticket is pending for answer. |
| **Steps** | 1. The use case begins when the project mentor or mentee click in the pending ticket in the ticket list on the domain mentoring module window. 2. The system shall create a box with the ticket description and a comment box. 3. The project mentor and mentee start typing into the comment box. 4. The Project or mentee click on the Add comment button. 5. The system shall to append the comment to the ticket in the DB. 6. The system shall notify that the append operation was successfully done. |
| **Post-conditions** | 1. Comment is append to a ticket an is ready to view by the Domain mentor. 2. The system shall notify the domain mentor that a comment was appended to a pending ticket that is on his queue. |
| **Exceptions** | The project mentor or mentee can click in the cancel button |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-016 Attach file to a ticket** |
| **Description** | The system shall allow attach file to a ticket |
| **Actor** | Mentee |
| **Pre-conditions** | 1. The mentee is successfully logged onto the collaborative platform system.3 2. The mentee has click on the new ticket button on the domain mentoring window. |
| **Steps** | 1. The use case begins when mentee clicks on the browse button on the new ticket window. 2. The system shall to create a box with the file system. 3. The mentee selects the desired file and clicks ok. 4. The path of the file is uploaded in the filed text field. |
| **Post-conditions** | 1. The file is attached to the new ticket |
| **Exceptions** | The mentees can clicks on the cancel button. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-017 Send new message** |
| **Description** | Allow user to send a message to another user |
| **Actor** | User |
| **Pre-conditions** | 1. The User is logged in |
| **Steps** | 1. The use case begin when the user click the message tab on his main page. 2. The system will change to the user mail view 3. The user will click the new message button 4. The system will pop up a new message form. 5. The user will click the To: button and a new search form will pop up for the user to select the receiver from the user list. 6. The user selects the receiver or receivers. 7. The user clicks the select button. 8. The search form close and the user gets bag to the new message form. 9. The user will type the message on the text box. 10. The user will click the send button. 11. The user will send the message to the receiver/s |
| **Post-conditions** | * 1. The messages gets send to the specified receiver/s |
| **Exceptions** | 1. The User cancels the messages before sending. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-018 Read message** |
| **Description** | Allow user to read a message from his/her queue |
| **Actor** | User |
| **Pre-conditions** | 1. The user is logged in. |
| **Steps** | 1. The use case begin when the user click the message tab on his main page. 2. The system will change to the user mail view 3. The user will select a messages from his queue. 4. The system will display the message on the bottom portion of the user mail view. |
| **Post-conditions** | 1. The message is marked as read on the database. 2. The font bold status is changed to false |
| **Exceptions** | 1. There are not messages on the user queue. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-019 Delete Message** |
| **Description** | Allow user to delete a message from his/her/queue |
| **Actor** | User |
| **Pre-conditions** | 1. The user is logged in |
| **Steps** | 1. The use case begin when the user click the message tab on his main page. 2. The system will change to the user mail view 3. The user will click the delete button on the right column of the message row. 4. The system will display a confirmation window asking for user confirmation. 5. The user will confirm the deletion. |
| **Post-conditions** | 1. The message is marked as deleted on the database. 2. The message is deleted from the mail view. |
| **Exceptions** | 1. There are not messages on the user queue. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-020 New Message Notification** |
| **Description** | Allow user to be notified upon message reception |
| **Actor** | System |
| **Pre-conditions** | 1. The user is logged in . 2. The user received a new message |
| **Steps** | 1. The use case begin when an user sends a messages. 2. The system will crate and Messages object. 3. The system will change the messages link on the receiver user side. |
| **Post-conditions** | * 1. The messages link changed from Message to Message\* |
| **Exceptions** | 1. No Exceptions will be accepted |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-021 Drop a ticket** |
| **Description** | Allow the administrator to drop (cancel) a ticket |
| **Actor** | Administrator |
| **Pre-conditions** | 1. The administrator is logged it. 2. The administrator already opened a ticket |
| **Steps** | 1. The use case begin when the administrator select a ticket that will be dropped. 2. The administrator changes the status to Drop or Cancel in the drop down list that identify the ticket status. 3. The administrator click the save button. |
| **Post-conditions** | * 1. The ticket will be marked as dropped or cancelled on the system DB |
| **Exceptions** | * 1. The administrator does not save the changes on the ticket. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-022 Ticket Due Notification** |
| **Description** | Alerts the system administrators to be notified when a ticket has being on a domain mentor queue for more than three calendar days without changing the status. |
| **Actor** | System |
| **Pre-conditions** | 1. The ticket has being on the domain mentor queue for more than 3 calendar days without status change. |
| **Steps** | 1. The uses case begins went the system trigger that handles the ticket due date is set to true. 2. The system will create a Notification object. 3. The system will send the notification to the administrators. |
| **Post-conditions** | 1. The administrator receive the system notification |
| **Exceptions** | 1. Non ticket triggers the due date. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-023 Close a ticket** |
| **Description** | Allow domain mentors to close a ticket as resolved. |
| **Actor** | Domain mentors |
| **Pre-conditions** | * 1. The above users had already logged in.   2. . The above users had opened a ticket. |
| **Steps** | 1. The above users will select resolved status from the status drop down list. 2. The above users will click the save button. 3. The ticket status will change resolved. |
| **Post-conditions** | 1. The ticket status is change on the Database to resolved 2. Users related to the ticket will receive notification on the status change |
| **Exceptions** | 1. The above users does not click the save button. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-024 Comment Added Notification** |
| **Description** | Notify users related to a ticket when comments are added to it by other users. |
| **Actor** | System |
| **Pre-conditions** | 1. User has added a comment to a ticket. |
| **Steps** | 1. The use case begin when the user that added a comment to a ticket hit the save button the ticket form. 2. The system will create a notification object. 3. The system will send this notification to the other user related to this ticket. Excluding the system administrators. |
| **Post-conditions** | 1. The user receive a notification message |
| **Exceptions** | 1. The user does not save the changes on the ticket. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-025 Accept Answer** |
| **Description** | Allow mentees to accept an answer to a ticket proposed by the domain mentor. |
| **Actor** | Mentee |
| **Pre-conditions** | 1. The mentee is logged in the platform 2. The domain mentor changed the ticket status to Answered |
| **Steps** | 1. The use case begin when the mentee open a ticket with status Answered 2. The mentee change the status of the ticket to answer accepted. 3. The mentee click the Save button |
| **Post-conditions** | 1. The ticket status will change to answer accepted |
| **Exceptions** | 1. The mentee did not change the status to Accepted. |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-026 Decline Answer** |
| **Description** | Allow mentees to decline an answer to a ticket proposed by the domain mentor. |
| **Actor** | Mentee |
| **Pre-conditions** | 1. The mentee is logged in the platform 2. The domain mentor changed the ticket status to Answered |
| **Steps** | 1. The use case begin when the mentee open a ticket with status Answered 2. The mentee will write a comment on a ticket 3. The mentee changes the status to pending. 4. The mentee click the Save button. |
| **Post-conditions** | 1. The ticket status will change to pending. 2. The system will notify the domain mentor. |
| **Exceptions** | 1. The mentee do not add a comment to the ticket that he/she will decline |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-027 view Score** |
| **Description** | Allow students to view their scores |
| **Actor** | Student |
| **Pre-conditions** | 1. The judges must have been assigned to students and they must have given grade to the students they were assigned. The grade must be posted by the admin. |
| **Steps** | 1. The student is logged in into the system. 2. The student grades must be posted already on the student’s home page. |
| **Post-conditions** | 1. The student is able to see the score. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-028 set Grade** |
| **Description** | Allow judge to grade an student during and after presentation |
| **Actor** | Judge |
| **Pre-conditions** | The judge must have assigned students. |
| **Steps** | 1. The judge is logged in into the system 2. The judge sees a list of all his/her assigned students who are ready to receive a grade. 3. The judge then, selects a student from the list and proceeds to grade him/her. 4. The judge completes and submits the grading form. 5. The judge receives a confirmation page of what he/she sent to the student. |
| **Post-conditions** | 1. The judge must have the student on their projects |
| **Exceptions** |  |
|  |  |
| **Use Case ID** | **CP-029**  **register Students** |
| **Description** |  |
| **Actor** | Student |
| **Pre-conditions** | The students must be enrolled into the CIS4911 and their emails set on the database. Also the student must have replied to registration email sent by the admin of the judges’ website. |
| **Steps** | 1. Student is on the registration page  2. The student selects and confirms the password.  3. The student receives the confirmation of registration. |
| **Post-conditions** | Student has been registered and is able to use the system. |
| **Exceptions** |  |
|  |  |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-030 setStudentPreference** |
| **Description** |  |
| **Actor** | Judge |
| **Pre-conditions** | The judge is already registered and is logged into his/her page |
| **Steps** | 1. The judge clicks the ‘Show Students’ button.  2. For each conflict student the judge selects.  3. The judge submits the selection. |
| **Post-conditions** | Any Conflict of student/judge will be on the database. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-031 Share Screen** |
| **Description** | Allow a user to share a screen |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | - User is logged in into the platform and is authenticated on the judges’ website  -  User is in the homepage  -  Presentation has been scheduled  -  Notification for the judging presentation is displayed for both involved parts  -  Other user involved in interview is not sharing screen |
| **Steps** | 1. User clicks on scheduled presentation notification  2. Used is redirected to the presentation page.  3. User clicks on share screen button |
| **Post-conditions** | User is able to share his/her screen and database is updated with required information |
| **Exceptions** | User tries to share screen while the other part is sharing in a different presentation |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-032 View Screen Share** |
| **Description** | User can star screen share session |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | -          User is logged in into the platform and is authenticated on the judges’ website  -          User is in the homepage  -          Presentation has been scheduled.  -          Notification for the presentation is displayed for both involved parties  -          The other user involved in the presentation is sharing a screen |
| **Steps** |  |
| **Post-conditions** | User is able to view screen shared by other by his/her counterpart |
| **Exceptions** | the other user is not sharing the a screen |
| **Use Case ID** | **CP-033 End Screen Sharing** |
| **Description** |  |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | -  User is logged in into the platform and is authenticated on the judges’ website  -  User is in the presentation portal  - User is sharing screen |
| **Steps** | - User clicks on end screen sharing |
| **Post-conditions** | -           User is not sharing screen  -          System information is updated to allow other party to share screen  -          Other party involved will not continue to see live feed from user screen |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-034 Create new document** |
| **Description** | Allows user to create a new document |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | -           User is logged in into the platform and is authenticated on the judges’ website  -          User is participating in a presentation |
| **Steps** | User clicks on Collaborative Editor          2. User clicks on create new document button          3. The new document editing session is started |
| **Post-conditions** | -          User is on the collaborative platform portal  -          A new document is being displayed |
| **Exceptions** | Connection error |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-035 Share active document** |
| **Description** | Allow users to invite another user to a shared document |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1. User is logged in 2. User is on the collaborative platform portal 3. User has at least one document to share |
| **Steps** | 1. User clicks on Collaborative editor 2. User clicks on share document button 3. Invitation is sent to other user 4. Other user receives notification 5. Other user joins the shared document session |
| **Post-conditions** | 1. User clicks on the judges portal 2. User is on the shared document session |
| **Exceptions** | 1. Connection error 2. Other user not available |
| **Use Case ID** | **CP-036 Delete shared document** |
| **Description** |  |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1. User is logged in 2. User is on the judges portal 3. User has at least one document to delete |
| **Steps** | 1.      User clicks on Collaborative Editor  2.      User clicks on the manage documents button  3.      User selects a document from documents list  4.      User clicks on the delete document button  5.      User is presented with a confirmation dialog  6.      User confirms deletion of file  7.      Document is deleted |
| **Post-conditions** | 1. User is on the collaborative platform portal |
| **Exceptions** | 1.      Connection Error  2.      User did not select a document to delete |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-037 Import Document** |
| **Description** | Allow a user to import a document |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.   User is logged in  2.   User is on the collaborative portal  3.   User has at least one document to import |
| **Steps** | 1.      User clicks on Collaborative Editor  2.      User clicks on import document button  3.      User chooses file to import and drags it over the import document area  4. The document is imported into the system |
| **Post-conditions** | 1. User is on the collaborative portal having either an interview, a session or a presentation |
| **Exceptions** | 1.      Connection Error  2.      File is not a valid document |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-038 Rename Document** |
| **Description** | Allow user to rename a document |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.    User is logged in  2.   User is on the collaborative portal having scheduled either a presentation, a session or an interview  3.   User has at least one document to rename |
| **Steps** | 1.      User clicks on Collaborative Editor  2.      User clicks on the manage documents button  3.      User clicks on document to rename  4.      User clicks on the rename document button  5.      Rename dialog appears  6.      User chooses new document name  7.      The document name is changed |
| **Post-conditions** | User is on the collaborative portal having either an interview, a session or a presentation |
| **Exceptions** | Connection error |
| **Use Case ID** | **CP-039 Save Document** |
| **Description** | Allow a user to save a document |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.    User is logged in  2.   User is on the collaborative portal having scheduled either a presentation, a session or an interview  3.   User has at least one document to save |
| **Steps** | 1.      User clicks on save document link  2.      The User is notified  3.      document is saved |
| **Post-conditions** | User is on the collaborative portal having either a presentation, a session or an interview |
| **Exceptions** | 1.      Connection Error  2.      There are no active documents to save |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-040 Open Document** |
| **Description** | Allows a user to open a document |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.    User is logged in  2.   User is on the collaborative portal having scheduled either a presentation, a session or an interview  3.   User has at least one document to open |
| **Steps** | 1.      User clicks on Collaborative Editor  2.      User clicks on manage documents  3.      User clicks on document to open  4.      User clicks on the open document button  5.      Document is loaded into the Editor |
| **Post-conditions** | User is on the collaborative portal having either a presentation, a session or an interview |
| **Exceptions** | 1.      Connection Error  2.      There are no active documents to open |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-041 Export Document** |
| **Description** | Allow users to export a document |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.    User is logged in  2.   User is on the collaborative portal having scheduled either a presentation, a session or an interview  3.   User has at least one document to export |
| **Steps** | 1.      User clicks on Collaborative Editor  2.      User clicks on the manage documents button  3.      User clicks on document to export  4.      User clicks on export document  5.      User receives the document as a file |
| **Post-conditions** | User is on the collaborative portal having either a presentation, a session or an interview |
| **Exceptions** | 1.      Connection Error  2.      There are no active documents to export |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-042 Upload Image** |
| **Description** | Upload a previously-selected image to the server during  an interview, presentation or a session |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1. A presentation, an interview, or a session is active 2. A whiteboard session has been started 3. An image has been selected for submission, according to the previous use case |
| **Steps** | 1.      The user shall click on the “Submit Drawing” button on top of the whiteboard  2.      The system shall upload the image  3.      The system shall let the user know whether the image was uploaded successfully by displaying a message below the “Choose File” button |
| **Post-conditions** | The file was successfully uploaded to the server and can be viewed by either user |
| **Exceptions** | 1. In step 2 of the even flow section, if the file was not one of the following types: “.gif”, “.jpeg”, “.jpg”, “.pjpeg”, “.x-png”, or “.png”, then the system will display an error message below the “Choose File” button asking the user to upload a different file |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-043 View uploaded image** |
| **Description** | view a previously-uploaded image specific to an interview, a session, or a presentation |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.  A presentation, an interview, or a session is active  2.  A whiteboard session has been started  3.  An image has been successfully submitted by either user and is ready to be seen |
| **Steps** | 1.   The user shall click on the “View Drawing” button, located below the “Choose File” button  2.   The system shall display the uploaded image below the whiteboard, ready for viewing |
| **Post-conditions** | The file was successfully uploaded to the server and can be viewed by either user |
| **Exceptions** | In step 2, if the image could not be retrieved for any reason, the system will display an error message below the “Choose File” button |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-044 Show or Restore Whiteboard** |
| **Description** | Places the whiteboard view in the shared presentation screen |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.  A interview session is active  2.  The whiteboard is not showing in the shared presentation screen. It is either blank or occupied by a different feature |
| **Steps** | 1. The user shall click in the blue “Whiteboard” button below the video chat |
| **Post-conditions** | The whiteboard is now showing in the shared screen for the user who clicked on the “Whiteboard” button |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-045 Select upload image** |
| **Description** | Select an image to upload to the server during a live interview to share it with the other user |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.      Either an interview, a presentation, or a session is active  2.      A whiteboard session has been started |
| **Steps** | 1.      User shall click on “Choose File” button on top of the whiteboard  2.      User shall browse for the image in his local drive and click on the “Open” button to upload it after clicking on the right file  3.      The system shall display the file name in the text box next to the “Choose File” button |
| **Post-conditions** | The file is ready for upload to the system |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-046 Draw with pencil** |
| **Description** | Allows actor to draw on the whiteboard using the pencil functionality |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.  A presentation, an interview, or a session is active  2.  A whiteboard session has already been started |
| **Steps** | 1.   The user shall click on the “Color” button on the left hand side of the whiteboard in the shared interview screen.  2.   The user shall select the color of the pencil from any of the following colors: red, green, blue, black, yellow, brown or purple, clicking on the button corresponding to the color he/she would like to use |
| **Post-conditions** | There will be a drawing on the whiteboard available for saving and viewing |
| **Exceptions** | In step 2, user clicks outside of the color palette, and the color is not changed |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-047 Type text into the whiteboard** |
| **Description** | Allows a user to type text into the whiteboard |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.  A presentation, an interview, or a session is active  2.  A whiteboard session has already been started |
| **Steps** | 1.      Use case begins when, on the whiteboard screen, the actor clicks on the “Pencil” option on the left-hand side  2.      From the sub-menu that pops up, the user shall click on the “Text” option  3.      The user shall click anywhere on the whiteboard screen where he/she wants the text to show  4.      The system shall display a window with a text box, and the user shall type the text that will appear  5.      Use case ends when the user presses “OK” after entering the text |
| **Post-conditions** | The text the user typed will appear on the screen where the user initially clicked before typing the text |
| **Exceptions** | 1.      In step 2, if the user clicks on the whiteboard when the sub-menu is showing, the current whiteboard feature selected will be used  2.      In step 4, if the user clicks “Cancel” instead of “OK”, the window will disappear, and no text will be shown |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-048 Clear contents of whiteboard** |
| **Description** | Allows users to clear the current contents of the whiteboard |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.  A presentation, an interview, or a session is active  2.  A whiteboard session has already been started |
| **Steps** | 1.      Use case begins when, on the whiteboard screen, the actor clicks on the “Menu” option on the left-hand side  2.       From the sub-menu that pops up, the user shall click on “Clear”  3.      The system shall display a pop-up message confirming if the user wants to continue  4.      Use case ends when user clicks “OK” on the pop-up |
| **Post-conditions** | The whiteboard screen is cleared |
| **Exceptions** | In step 4, if the user clicks on “Cancel”, the whiteboard contents will not be cleared, and the whiteboard screen will be shown again |

|  |  |
| --- | --- |
| **Use Case ID** | **CP-049 Erase from the whiteboard** |
| **Description** | Allowusers to clear to partially erase drawings |
| **Actor** | Student, Judge, Mentor, Mentee |
| **Pre-conditions** | 1.  Apresentation, an interview, or a session is active  2.  A whiteboard session has already been started |
| **Steps** | 1.   Use case begins when, on the whiteboard screen, the actor clicks on the “Pencil” option on the left-hand side  2.   From the sub-menu that pops up, the user shall click on “Eraser”  3.   The user shall click in the whiteboard location where he/she would like to start erasing  4.   The user shall drag the cursor to where he/she wants to erase from the whiteboard  5.   Use case ends when the user releases the click |
| **Post-conditions** | The drawing on the whiteboard screen is partially erased |
| **Exceptions** | In step 4, if the user drags the cursor beyond the boundaries of the share screen interview, the whiteboard contents will stop being erased and the use case will end |

**Version 3 Use Cases**

|  |  |
| --- | --- |
| **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** | **CPv3-014 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 |

**Version 5 Use Cases**

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-001 Pull Mentee Reports** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The administrator pulls mentee reports. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system. |
| **Steps** | 1. The use case begins when the administrator clicks on the menu for mentee reports. 2. The system is redirected to a page with the listing of mentees with information such as name, email, school, user status, project assignation, etc. 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      **\_\_\_\_** |
| **Alternative Actions** | * At step 2, the administrator can click a column of the report, and the system will render the report but sorted by the column just clicked. * At step 2, the administrator can use a filter below each column header and the system will render a new report filtered with the information just entered. * At step 2, the administrator drags and drop a column in the position of another column and the system will render the report with the new column order. |
| **Exceptions** | * NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-002 Pull Mentor Reports** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The administrator pulls mentor reports. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system. |
| **Steps** | 1. The use case begins when the administrator clicks on the menu for mentor reports. 2. The system is redirected to a page with the listing of mentors with information such as name, email, employers, user status, etc. 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●     **\_\_\_\_\_** |
| **Alternative Actions** | * At step 2, the administrator can click a column of the report, and the system will render the report but sorted by the column just clicked. * At step 2, the administrator can use a filter below each column header and the system will render a new report filtered with the information just entered. * At step 2, the administrator drags and drop a column in the position of another column and the system will render the report with the new column order. |
| **Exceptions** | * NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-003 Pull Ticket Reports** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The administrator pulls ticket reports. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system. |
| **Steps** | 1. The use case begins when the administrator clicks on the menu for ticket reports. 2. The system is redirected to a page with the listing of tickets with information such as Date Created, Status, Closed Date, etc. 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●     **\_\_\_\_\_** |
| **Alternative Actions** | * At step 2, the administrator can click a column of the report, and the system will render the report but sorted by the column just clicked. * At step 2, the administrator can use a filter below each column header and the system will render a new report filtered with the information just entered. * At step 2, the administrator drags and drop a column in the position of another column and the system will render the report with the new column order. |
| **Exceptions** | * NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-004 Schedule Video Conference Ahead of Time** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | As a user I want to schedule real-time video conferences ahead of time with anyone in the system so that we can collaborate and have a better planning. |
| **Actor** | Mentor/Mentee/Admin User |
| **Pre-conditions** | * The user is logged into the system. * The user has navigated to the video conferencing module |
| **Steps** | 1. The use case begins when the user clicks on the create videoconference link 2. The system shall redirect the user to the create videoconference page. 3. The system shall provide the user with a form for the videoconference creation. 4. The user selects the **later** option 5. The user shall enter the meeting **date**, **time**, **subject**, notes and invitee emails. 6. The use case ends when the user clicks the create button. |
| **Post-conditions** | * The number of video conference records has increased by one * An invitation is sent to each invitee. * The user is redirected to the meetings details page. |
| **Alternative Actions** | * After step 2, the user may opt not to enter the information and leave the page. |
| **Exceptions** | NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-005 Schedule On-Demand Video Conference** |
| **Update for** |  |
| **Description** | As a mentee/mentor I want to be able to create meetings on demand so that I have a fast way to set up meetings. |
| **Actor** | Mentor/Mentee/Admin User |
| **Pre-conditions** | * The user is logged into the system. * The user has navigated to the video conferencing module |
| **Steps** | 1. The use case begins when the user clicks on the create videoconference link 2. The system shall redirect the user to the create videoconference page. 3. The system shall provide the user with a form for the videoconference creation. 4. The user selects the **now** option 5. The user shall enter the **subject**, **notes** and **invitee emails.** 6. The use case ends when the user clicks the create button. |
| **Post-conditions** | * The system validates the entered input * The number of video conference records has increased by one * An invitation is sent to each invitee. * The user is redirected to the meeting details page. |
| **Alternative Actions** | * After step 2, the user may opt not to enter the information and leave the page. |
| **Exceptions** | NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-006 Join Video Conference** |
| **Update for** | \_\_\_\_\_ |
| **Description** | The user joins a video conference |
| **Actor** | Mentor/Mentee/Admin User |
| **Pre-conditions** | * The user is logged into the system * The user has navigated to the video conferencing module * The user has clicked on a meeting details link |
| **Steps** | 1. The use case begins when the user clicks on **join** link. 2. The user will be redirected to the videoconference room. 3. The user will wait for other participants to join the meeting. 4. The meeting will start when at least 2 participants join the meeting. 5. The use case ends when the user clicks the disconnect button. |
| **Post-conditions** | * The user will be redirected to the videoconferences page. |
| **Alternative Actions** | * At step 3, the user can leave the meeting by closing the page. * At step 3, no participants may join the meeting; therefore the meeting will not start. |
| **Exceptions** | * At step 3, the user might not allow the browser to access his/her camera. * At step 3, the user might not allow the browser to access his/her microphone. |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-007 Share Screen** |
| **Update For** |  |
| **Description** | User shares his/her screen during a videoconference |
| **Actor** | Mentor/Mentee/Admin |
| **Pre-conditions** | * The user is logged into the system. * The videoconference has started |
| **Steps** | 1. The use case begins when the user clicks the share screen button. 2. The Screen Capturing plugin shall load. 3. The Screen Capturing plugin shall prompt the user to select one of his/her screens 4. The user shall select the screen to share 5. The user screen will begin to be shared with the other participants. 6. The panel will show the screen to each participant. 7. The use case ends when the user clicks the stop sharing screen button. |
| **Post-conditions** |  |
| **Alternative Actions** | * At step 4, the user may not select a screen. * At step 5, the user may cancel the screen sharing so it is never shared. * At step 5, if other participant shares his/her screen, the current screen being shared will no longer be shared. |
| **Exceptions** | * At step 2, the plugin is not installed. * At step 2, the plugin fails to open. |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-008 Use Video Conference Whiteboard** |
| **Update For** |  |
| **Description** | * As a mentor/mentor I want to draw on a virtual whiteboard so that I can visually explain anything better during video conferences. |
| **Actor** | Mentor/Mentee/Admin |
| **Pre-conditions** | * The user is logged into the system. * The videoconference has started |
| **Steps** | 1. The use case begins when the user drags the mouse over the whiteboard panel as input. 2. The system shall illustrate the drawing pattern. 3. The system shall broadcast the changes through the connection. 4. The system shall render the drawing pattern on the other participants’ whiteboard. 5. The use case ends when the user clicks the clear whiteboard button. |
| **Post-conditions** |  |
| **Alternative Actions** | * At step 4, any of the other participants my click the clear whiteboard button and draw on the screen. * At step 1, the user may click the disconnect button. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-009 Invite More People** |
| **Update For** |  |
| **Description** | As a meeting participant I want to invite more people to the video conference while it is happening so that they are allowed to join the meeting. |
| **Actor** | Mentor/Mentee/Admin |
| **Pre-conditions** | * The user is logged into the system. * The videoconference has started |
| **Steps** | 1. The use case begins when the user clicks on the invite more people on the settings dropdown menu. 2. The system shall prompt a form for the invitations. 3. The user shall enter the invitee **email address**. 4. The use case ends when the user clicks the **submit** button |
| **Post-conditions** | 1. The system shall validate the input. 2. The number of invitations records has increased by one. 3. An invitation email is sent to the invitee. |
| **Alternative Actions** | * At step 2, the user may close the form. |
| **Exceptions** |  |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-010 Schedule Video Conference From Ticket** |
| **Update for** |  |
| **Description** | * As a mentor I want to able to schedule a videoconference from a ticket so that it can be easily scheduled. |
| **Actor** | Mentor |
| **Pre-conditions** | * The mentor is logged into the system. * The mentor has navigated clicked on a ticket on the homepage. |
| **Steps** | 1. The use case begins when the user clicks on the **schedule meeting** button. 2. The system shall present the user with a form for the videoconference creation, automatically filled with **subject** and **notes** fields. 3. The user shall enter the **date, time,** and any other **invitee emails.** 4. The use case ends when the user clicks the **schedule** button. |
| **Post-conditions** | * The system validates the entered input * The number of video conference records has increased by one * An invitation is sent to each invitee. * The user is redirected to the meeting details page. |
| **Alternative Actions** | * After step 2, the user may opt not to enter the information and leave the page. |
| **Exceptions** | NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-011 Accept Video Conference Invitation** |
| **Update for** |  |
| **Description** | * As an invitee I want to accept videoconference invitations so that they are quickly confirmed. |
| **Actor** | Any system user that has been invited to a videoconference. |
| **Pre-conditions** | * The user is logged into the system. * The user has been invited to join a videoconference. * The user has navigated to video conferencing module |
| **Steps** | 1. The use case begins when the user clicks on the **meeting link**. 2. The system shall present the user a panel with detailed information about the meeting. 3. The user shall click the **accept invitation** button. 4. The system shall prompt the user for confirmation. 5. The use case ends when the user confirm **yes**. |
| **Post-conditions** | * The invitation status is marked as “Accepted” |
| **Alternative Actions** | * After step 4, the user may opt not to confirm; therefore, the status would not change. |
| **Exceptions** | NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-012 Reject Video Conference Invitation** |
| **Update for** |  |
| **Description** | * As an invitee I want to reject videoconference invitations so that they can be rescheduled. |
| **Actor** | Any system user that has been invited to a videoconference. |
| **Pre-conditions** | * The user is logged into the system. * The user has been invited to join a videoconference. * The user has navigated to video conferencing module |
| **Steps** | 1. The use case begins when the user clicks on the **meeting link**. 2. The system shall present the user a panel with detailed information about the meeting. 3. The user shall click the **reject invitation** button. 4. The system shall prompt the user for confirmation. 5. The use case ends when the user confirm **yes**. |
| **Post-conditions** | * The invitation status is marked as “Rejected” |
| **Alternative Actions** | * After step 4, the user may opt not to confirm; therefore, the status would not change. |
| **Exceptions** | NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-013 Cancel Video Conference Meeting** |
| **Update for** |  |
| **Description** | * As a meeting moderator, I want to able to cancel meetings so that my invitees know the meeting has been cancelled. |
| **Actor** | A meeting moderator, i.e., a person who scheduled a videoconference. |
| **Pre-conditions** | * The user is logged into the system. * The user has scheduled a videoconference. * The user has navigated to video conferencing module |
| **Steps** | 1. The use case begins when the user clicks on the **meeting link**. 2. The system shall present the user a panel with detailed information about the meeting. 3. The user shall click the **cancel meeting** button. 4. The system shall prompt the user for confirmation. 5. The use case ends when the user confirm **yes**. |
| **Post-conditions** | * The meeting status is marked as “Cancelled” * The page is reloaded * The system will send a notification to the invitees. |
| **Alternative Actions** | * After step 4, the user may opt not to confirm; therefore, the status would not change. |
| **Exceptions** | NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-014 Pull amount of tickets created** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The administrator gets quantitative amount of tickets historically created. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with Tickets Created and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.). 2. The system refresh the report area, showing a chart where the Y axis represents the amount of tickets created and the X axis represent the second dimension (Field “By” selected in the step 1). 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      **\_\_\_\_** |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |
| **Use Case ID** | **CPv5-015 Pull tickets created details** |
| **Update for** | \_\_\_\_\_ |
| **Description** | The administrator gets details on the tickets historically created. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard. |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with Tickets Created and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.).The administrator selected the field “Report Format” with Details. 2. The system refresh the report area, showing a list with the tickets historically created. 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      \_\_\_\_ |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-016 Pull amount of tickets closed** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The administrator gets quantitative amount of tickets historically closed. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with Tickets Closed and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.). 2. The system refresh the report area, showing a chart where the Y axis represents the amount of tickets Closed and the X axis represent the second dimension (Field “By” selected in the step 1). 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      **\_\_\_\_** |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |
| **Use Case ID** | **CPv5-017 Pull tickets closed details** |
| **Update for** | \_\_\_\_\_ |
| **Description** | The administrator gets details on the tickets historically closed. |
| **Actor** | Administrator |
| **Pre-conditions** | ●   The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard. |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with Tickets Closed and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.).The administrator selected the field “Report Format” with Details. 2. The system refresh the report area, showing a list with the tickets historically closed. 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      \_\_\_\_ |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-018 Pull AVG ticket duration.** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The administrator gets quantitative amount on the average time (hours) that takes to answer a ticket. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with AVG Ticket Duration and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.). 2. The system refresh the report area, showing a chart where the Y axis represents the AVG hours for closing a ticket and the X axis represent the second dimension (Field “By” selected in the step 1). 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      **\_\_\_\_** |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |
| **Use Case ID** | **CPv5-019 Pull ticket duration details.** |
| **Update for** | \_\_\_\_\_ |
| **Description** | The administrator gets details on the tickets historically closed and the duration (in hours) of that ticket. |
| **Actor** | Administrator |
| **Pre-conditions** | ●   The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard. |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with AVG Ticket Duration and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.).The administrator selected the field “Report Format” with Details. 2. The system refresh the report area, showing a list with the tickets historically closed and the duration (in Hours). 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      \_\_\_\_ |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-020 Pull AVG Time Mentor to answer.** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The administrator gets quantitative amount on the average time (hours) that takes for a mentor to answer a ticket. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with AVG Time Mentor to answer and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.). 2. The system refresh the report area, showing a chart where the Y axis represents the AVG hours for answering a ticket and the X axis represent the second dimension (Field “By” selected in the step 1). 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      **\_\_\_\_** |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |
| **Use Case ID** | **CPv5-021 Pull Time Mentor to answer details.** |
| **Update for** | \_\_\_\_\_ |
| **Description** | The administrator gets details on the tickets that has been answered by mentor and how long it took (in hours) to answer that ticket. |
| **Actor** | Administrator |
| **Pre-conditions** | ●   The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard. |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with AVG Time Mentor to answer and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.).The administrator selected the field “Report Format” with Details. 2. The system refresh the report area, showing a list with the tickets historically answered and the time to answer (in Hours). 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      \_\_\_\_ |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-022 Pull amount of tickets currently open** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The administrator gets quantitative amount of tickets currently open. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with Tickets currently open and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.). 2. The system refresh the report area, showing a chart where the Y axis represents the amount of tickets currently open and the X axis represent the second dimension (Field “By” selected in the step 1). 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      **\_\_\_\_** |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |
| **Use Case ID** | **CPv5-023 Pull tickets currently open details** |
| **Update for** | \_\_\_\_\_ |
| **Description** | The administrator gets details on the tickets currently open. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard. |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with Tickets currently open and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.).The administrator selected the field “Report Format” with Details. 2. The system refresh the report area, showing a list with the tickets currently open. 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      \_\_\_\_ |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-024 Pull amount of tickets unanswered.** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The administrator gets quantitative amount of tickets currently unanswered. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with Tickets currently unanswered and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.). 2. The system refresh the report area, showing a chart where the Y axis represents the amount of tickets currently unanswered and the X axis represent the second dimension (Field “By” selected in the step 1). 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      **\_\_\_\_** |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |
| **Use Case ID** | **CPv5-025 Pull tickets currently unanswered details** |
| **Update for** | \_\_\_\_\_ |
| **Description** | The administrator gets details on the tickets currently unanswered. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system and opened the utilization dashboard via menu Reports->Utilization Dashboard. |
| **Steps** | 1. The use case begins when the administrator selects the required filed Report Type with Tickets currently unanswered and selects the required field “By” with the second dimension that he is looking for (Day, Year, Month, Project, Mentee, etc.).The administrator selected the field “Report Format” with Details. 2. The system refresh the report area, showing a list with the tickets currently unanswered. 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      \_\_\_\_ |
| **Alternative Actions** | * At step 2, the administrator can change the dimension to see a different view of the same report. * At step 2, the administrator can filter even more the data by using several filters: From, To, Domain (Aggregated), Mentee, etc. * At step 1 and 2, the administrator may selected another report; see all other utilization dashboard use cases. |
| **Exceptions** | * NONE |

|  |  |
| --- | --- |
| **Use Case ID** | **CPv5-026 Pull Frequent Mentee Sub-Domains** |
| **Update for** | **\_\_\_\_\_** |
| **Description** | The administrator pulls a list of the most frequently used sub-domains by mentees. |
| **Actor** | Administrator |
| **Pre-conditions** | ●     The administrator is logged into the system. |
| **Steps** | 1. The use case begins when the administrator clicks on the menu Repots, Submenu: Frequent Mentee Sub-Domain. 2. The system is redirected to a page with the listing of the top 10 frequent subdomains. That includes the combination of the most frequent sub-domains. 3. The use case ends when the administrator navigates to another page. |
| **Post-conditions** | ●      **\_\_\_\_** |
| **Alternative Actions** | * At step 2, the administrator can change the max number of rules (number of records to retrieve). Then click refresh and the system will adjust the listing with the max amount of records desired. * At step 2, the administrator can change the Upper and Lower bound min support (Option for data mining experts). After doing this and clicking refresh, the report will be refreshed according to the threshold stablished. |
| **Exceptions** | * NONE |

## 5.3 Appendix C – Documented Class Interfaces

**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

**Mentoring Module: Version 5 - Documented Class Interfaces**

Method Signatures for ReportMenteeController.php

**public function actionIndex()**

@precondition: The Administrator or ReportMenteeView has requested a search of mentees. The POST may contain instructions for sorting, filtering or displaying the report for first time.

@postcondition: The system renders the listing of mentees searched.

Method Signatures for the Model ReportMentee.php

**public function search()**

@precondition: The ReportMenteeController has requested a search of mentees. The parameter and sort instructions for the search are already set as attributes of the Model.

@postcondition: The system returns the Model with all the mentees that apply the to the filter conditions.

Method Signatures for ReportMentorController.php

**public function actionIndex()**

@precondition: The Administrator or ReportMentorView has requested a search of mentees. The POST may contain instructions for sorting, filtering or displaying the report for first time.

@postcondition: The system renders the listing of mentors searched.

Method Signatures for the Model ReportMentor.php

**public function search()**

@precondition: The ReportMentorController has requested a search of mentors. The parameter and sort instructions for the search are already set as attributes of the Model.

@postcondition: The system returns the Model with all the mentors that apply the to the filter conditions.

Method Signatures for ReportTicketController.php

**public function actionIndex()**

@precondition: The Administrator or ReportMentorTicket has requested a search of mentees. The POST may contain instructions for sorting, filtering or displaying the report for first time.

@postcondition: The system renders the listing of tickets searched.

Method Signatures for the Model ReportTicket.php

**public function search()**

@precondition: The ReportTicketController has requested a search of tickets. The parameter and sort instructions for the search are already set as attributes of the Model.

@postcondition: The system returns the Model with all the tickets that apply the to the filter conditions.

Method Signatures for UtilizationDashboardController.php

**public function actionPullTicketsCreated ()**

@precondition: The UtilizationDashboard view has requested the amount of tickets created. The POST variable will contain all the necessary parameters such as the second dimension and filters.

@postcondition: The system makes a response with a JSON string that can be feed to a Google chart.

**public function actionPullTicketsClosed ()**

@precondition: The UtilizationDashboard view has requested the amount of tickets closed. The POST variable will contain all the necessary parameters such as the second dimension and filters.

@postcondition: The system makes a response with a JSON string that can be feed to a Google chart.

**public function actionPullAVGTicketDuration ()**

@precondition: The UtilizationDashboard view has requested the average time for a ticket to be closed. The POST variable will contain all the necessary parameters such as the second dimension and filters.

@postcondition: The system makes a response with a JSON string that can be feed to a Google chart.

**public function actionPullAVGTimeMentorAnswer ()**

@precondition: The UtilizationDashboard view has requested the average time for a mentor to answer a ticket. The POST variable will contain all the necessary parameters such as the second dimension and filters.

@postcondition: The system makes a response with a JSON string that can be feed to a Google chart.

**public function actionPullTicketsCurrentlyOpened ()**

@precondition: The UtilizationDashboard view has requested the amount of tickets currently opend. The POST variable will contain all the necessary parameters such as the second dimension and filters.

@postcondition: The system makes a response with a JSON string that can be feed to a Google chart.

**public function actionPullTicketsCurrentlyOpened ()**

@precondition: The UtilizationDashboard view has requested the amount of tickets currently unanswered. The POST variable will contain all the necessary parameters such as the second dimension and filters.

@postcondition: The system makes a response with a JSON string that can be feed to a Google chart.

**public function actionPullTicketsCreatedDetails ()**

@precondition: The UtilizationDashboard view has requested the details of tickets historically created. The POST variable will contain all the necessary parameters such as the filters.

@postcondition: The system instruct the view to render the tickets that apply.

**public function actionPullTicketsClosedDetails ()**

@precondition: The UtilizationDashboard view has requested the details of tickets historically closed. The POST variable will contain all the necessary parameters such as the filters.

@postcondition: The system instruct the view to render the tickets that apply.

**public function actionPullTicketDurationDetails ()**

@precondition: The UtilizationDashboard view has requested the details of tickets historically closed with the time it took to close it. The POST variable will contain all the necessary parameters such as the filters.

@postcondition: The system instruct the view to render the tickets that apply.

**public function actionPullTimeMentorAnswerDetails ()**

@precondition: The UtilizationDashboard view has requested the details of tickets that have been answered by the mentor with the time it took to answer it. The POST variable will contain all the necessary parameters such as the filters.

@postcondition: The system instruct the view to render the tickets that apply.

**public function actionPullTicketsCurrentlyOpenedDetails ()**

@precondition: The UtilizationDashboard view has requested the details of tickets that are currently open. The POST variable will contain all the necessary parameters such as the filters.

@postcondition: The system instruct the view to render the tickets that apply.

**public function actionPullTicketsUnansweredDetails ()**

@precondition: The UtilizationDashboard view has requested the details of tickets that are currently uanswered. The POST variable will contain all the necessary parameters such as the filters.

@postcondition: The system instruct the view to render the tickets that apply.

Method Signatures for the model UtilizationDashboard.php

**public function retrieveTicketsCreatedDashboardData ()**

@precondition: The UtilizationDashboardController has requested the amount of tickets historically created. The model already contains attributes with all the parameters and the second dimension.

@postcondition: The system returns a JSON string that can be feed to a Google chart.

**public function retrieveTicketsClosedDashboardData ()**

@precondition: The UtilizationDashboardController has requested the amount of tickets historically closed. The model already contains attributes with all the parameters and the second dimension.

@postcondition: The system returns a JSON string that can be feed to a Google chart.

**public function retrieveAVGTicketDurationDashboardData ()**

@precondition: The UtilizationDashboardController has requested the average time that takes to tickets to be closed. The model already contains attributes with all the parameters and the second dimension.

@postcondition: The system returns a JSON string that can be feed to a Google chart.

**public function retrieveAVGTimeMentorAnswerDashboardData ()**

@precondition: The UtilizationDashboardController has requested the average time that takes a mentor to answer a ticket. The model already contains attributes with all the parameters and the second dimension.

@postcondition: The system returns a JSON string that can be feed to a Google chart.

**public function retrieveTicketsCurrentlyOpenedDashboardData()**

@precondition: The UtilizationDashboardController has requested the amount of tickets tha are currently open. The model already contains attributes with all the parameters and the second dimension.

@postcondition: The system returns a JSON string that can be feed to a Google chart.

**public function retrieveUnansweredTicketsDashboardData ()**

@precondition: The UtilizationDashboardController has requested the amount of tickets that are currently unanswerd. The model already contains attributes with all the parameters and the second dimension.

@postcondition: The system returns a JSON string that can be feed to a Google chart.

**public function retrieveTicketsCreatedDetailsData ()**

@precondition: The UtilizationDashboardController has requested the relation of historically created tickets. The model already contains attributes with all the parameters.

@postcondition: The system returns an array of the models filtered by the conditions.

**public function retrieveTicketsClosedDetailsData ()**

@precondition: The UtilizationDashboardController has requested the relation of historically closed tickets. The model already contains attributes with all the parameters.

@postcondition: The system returns an array of the models filtered by the conditions.

**public function retrieveTicketDurationDetailsData()**

@precondition: The UtilizationDashboardController has requested the relation of historically closed tickets and the time it took to close then. The model already contains attributes with all the parameters.

@postcondition: The system returns an array of the models filtered by the conditions.

**public function retrieveTimeMentorAnswerDetailsData ()**

@precondition: The UtilizationDashboardController has requested the relation of historically answered tickets and the time it took to close then. The model already contains attributes with all the parameters.

@postcondition: The system returns an array of the models filtered by the conditions.

**public function retrieveTicketsCurrentlyOpenedDetailsData ()**

@precondition: The UtilizationDashboardController has requested the relation of tickets that are currently open. The model already contains attributes with all the parameters.

@postcondition: The system returns an array of the models filtered by the conditions.

**public function retrieveTicketsUnansweredDetailsData ()**

@precondition: The UtilizationDashboardController has requested the relation of tickets currently unanswered. The model already contains attributes with all the parameters.

@postcondition: The system returns an array of the models filtered by the conditions.

Method Signatures for AnalyticsController.php

**public function actionPullFrequentMenteeSubdomains ()**

@precondition: The administrator has requested the relation of the most frequent mentee subdomains. The POST variable contains all the filters that for the report.

@postcondition: The system returns an array of the frequent sub-domains mined.

Method Signatures for the model AssociationFilter.php

**public function retrieveSubDomainsUsedPerMentee ()**

@precondition: The analytics controllers has requested the relation of the most frequent mentee subdomains.

@postcondition: The system returns an array with all the sub-domains per mentee.

Method Signatures for the component FPGrowth.php

**public function buildAssociations ($source)**

@precondition: The analytics controllers has requested the association mining of frequent itemsets. The parameter already have all the instances that will be mined.

@postcondition: The system returns an array with the rules mined.

## 5.4 Appendix D – Diary of Meetings

|  |  |
| --- | --- |
| **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 |

**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 |

**Mentoring Module: Version 5**

|  |  |
| --- | --- |
| **Date** | 01/19/2015 |
| **Location** | Conference Call |
| **Start** | 5:30 PM |
| **End** | 7:00 PM |
| **In Attendance** | Juan Caraballo, Jorge Travieso, Adrian Alfonso |
| **Late** | N/A |
| **Agenda** | Project Introduction |
| **Summary of Discussion** | An introduction to the project, the overall scope of work for this semester, and the expectations for us was discussed. |
| **Assigned Tasks** | * Get previous documentation and source code * Play with previous version of the application * Start collecting user stories * Identify bugs on previous versions |

|  |  |
| --- | --- |
| **Date** | 01/22/2015 |
| **Location** | Conference Call |
| **Start** | 5:30 PM |
| **End** | 7:00 PM |
| **In Attendance** | Juan Caraballo, Jorge Travieso, Adrian Alfonso, Robert Loredo |
| **Late** | N/A |
| **Agenda** | Capstone review of student storyboard |
| **Summary of Discussion** | User stories were reviewed |
| **Assigned Tasks** | * Create scenarios and mockups |

|  |  |
| --- | --- |
| **Date** | 01/23/2015 |
| **Location** | Conference Call |
| **Start** | 6:00 PM |
| **End** | 7:00 PM |
| **In Attendance** | Juan Caraballo, Jorge Travieso, Adrian Alfonso, Robert Loredo |
| **Late** | N/A |
| **Agenda** | Capstone review of student storyboard |
| **Summary of Discussion** | The carryover backlog was checked and the newly created user stories reviewed. The users stories were also redefined and expanded. |
| **Assigned Tasks** | * Expand user stories into scenarios to identify use cases, and create scenarios and UI mockups |

|  |  |
| --- | --- |
| **Date** | 01/27/2015 |
| **Location** | Conference Call |
| **Start** | 5:30 PM |
| **End** | 6:30 PM |
| **In Attendance** | Juan Caraballo, Jorge Travieso, Adrian Alfonso, Robert Loredo |
| **Late** | N/A |
| **Agenda** | Capstone review of student storyboard |
| **Summary of Discussion** | Reviewed new expanded users stories and UI mockups. Reviewed newly edited carryover backlog. |
| **Assigned Tasks** | * Split up scenarios into many user stories. * Assign tasks to stories. |

|  |  |
| --- | --- |
| **Date** | 01/29/2015 |
| **Location** | Conference Call |
| **Start** | 5:30 PM |
| **End** | 6:30 PM |
| **In Attendance** | Juan Caraballo, Jorge Travieso, Adrian Alfonso |
| **Late** | Jorge Travieso, Adrian Alfonso |
| **Agenda** | Capstone review of student storyboard |
| **Summary of Discussion** | Confirmed that we are going to be working on the reporting tool for now, and the analytics part would be postponed. The refined use cases were checked and confirmed. We were prepared to start sprint 1. |
| **Assigned Tasks** | * Assign use stories to sprint 1 and create tasks. |

|  |  |
| --- | --- |
| **Date** | 02/13/2015 |
| **Location** | Conference Call |
| **Start** | 5:30 PM |
| **End** | 7:30 PM |
| **In Attendance** | Juan Caraballo, Robert Loredo, Jorge Travieso, Adrian Alfonso |
| **Late** | N/A |
| **Agenda** | Sprint #1 Review |
| **Summary of Discussion** | * Reviewed the work done during Sprint #1. * Planned user stories were accepted. * Collected New Requirements * Planned Sprint #2 User Stories |
| **Assigned Tasks** | * Assign use stories to sprint 2 and create tasks. Meet with Robert the following week to brainstorm about the UX of the meetings and analytics module. |
| **Date** | 02/18/2015 |
| **Location** | Conference Call |
| **Start** | 5:30 PM |
| **End** | 6:30 PM |
| **In Attendance** | Robert Loredo, Jorge Travieso, Adrian Alfonso |
| **Late** | N/A |
| **Agenda** | Technical Specification |
| **Summary of Discussion** | * Brainstormed on how to improve the User Experience of the meetings views. * Defined what we want to do with the analytics platform. |
| **Assigned Tasks** | * Add new UX elements that we are missing in the current template such as popovers and instructions. * Research for analytics open source tools. |

|  |  |
| --- | --- |
| **Date** | 02/27/2015 |
| **Location** | Conference Call |
| **Start** | 5:30 PM |
| **End** | 6:30 PM |
| **In Attendance** | Juan Caraballo, Jorge Travieso, Adrian Alfonso |
| **Late** |  |
| **Agenda** | Sprint #2 Review |
| **Summary of Discussion** | * Reviewed the work done during Sprint #2. * Planned user stories were accepted. * Collected New Requirements * Planned Sprint #3 User Stories |
| **Assigned Tasks** | * Assign use stories to sprint 3 and create tasks. |

|  |  |
| --- | --- |
| **Date** | 03/13/2015 |
| **Location** | Conference Call |
| **Start** | 5:30 PM |
| **End** | 7:30 PM |
| **In Attendance** | Juan Caraballo, Jorge Travieso, Adrian Alfonso |
| **Late** |  |
| **Agenda** | Scenarios Discussion |
| **Summary of Discussion** | * We meet to discuss and redact the scenarios for the requirements, what is needed and how we are going to do it. |
| **Assigned Tasks** | * - Redact the scenarios discussed |

|  |  |
| --- | --- |
| **Date** | 03/20/2015 |
| **Location** | Conference Call |
| **Start** | 5:00 PM |
| **End** | 6:30 PM |
| **In Attendance** | Juan Caraballo, Jorge Travieso, Adrian Alfonso |
| **Agenda** | Sprint #3 Review |
| **Summary of Discussion** | * Reviewed the work done during Sprint #3. * Planned user stories were accepted. * Collected New Requirements * Planned Sprint #4 User Stories |
| **Assigned Tasks** | * Assign use stories to sprint 4 and create tasks. |

|  |  |
| --- | --- |
| **Date** | 04/03/2015 |
| **Location** | Conference Call |
| **Start** | 5:00 PM |
| **End** | 6:00 PM |
| **In Attendance** | Juan Caraballo, Jorge Travieso, Adrian Alfonso |
| **Late** |  |
| **Agenda** | Sprint #4 Review |
| **Summary of Discussion** | * Reviewed the work done during Sprint #4. * Planned user stories were accepted. * Collected New Requirements * Planned Sprint #5 User Stories |
| **Assigned Tasks** | * Assign use stories to sprint 5 and create tasks. |

# 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>