Software Requirement Specification



Team BEAM Solutions, Seneca College

Contents

[1. Document Information 4](#_Toc300126398)

[1.1 Document Authors 4](#_Toc300126399)

[1.1.1 Document Approvals 4](#_Toc300126400)

[1.2 Revision History 4](#_Toc300126401)

[1.3 Document Purpose 4](#_Toc300126402)

[2. Project Gemini Architecture 5](#_Toc300126403)

[2.2 Actors 5](#_Toc300126404)

[2.3 High Level System Review 5](#_Toc300126405)

[2.4 Design Trees 9](#_Toc300126406)

[3. Business Process 14](#_Toc300126408)

[3.1 Use Case Diagrams 14](#_Toc300126409)

[3.1.1 Business Use Case Diagram 14](#_Toc300126410)

[3.1.2 System Use Case Diagram 15](#_Toc300126412)

[3.2 Process Matrix: 16](#_Toc300126413)

[4. Design Documents 18](#_Toc300126414)

[4.1 Business Rules 18](#_Toc300126415)

[4.2. Use Case Specifications 20](#_Toc300126416)

[4.2.1 Create Team Account 20](#_Toc300126417)

[4.2.2 Match Teams/Projects 21](#_Toc300126418)

[4.2.3 View Project Status 23](#_Toc300126419)

[4.2.4 Manage Project Files 24](#_Toc300126420)

[4.2.5 Manage Site Accounts 25](#_Toc300126421)

[4.2.6 Change Project Status to Past 27](#_Toc300126422)

[4.2.7 Rank Projects 28](#_Toc300126423)

[4.2.8 Rank Teams 30](#_Toc300126424)

[4.2.9 Approve Project 31](#_Toc300126425)

[4.2.10 Login 33](#_Toc300126426)

[4.2.11 Proceed Projects to Appropriate Semester 34](#_Toc300126427)

[4.2.12 Review Comment 35](#_Toc300126428)

[4.2.13 Deactivate Accounts 36](#_Toc300126429)

[4.2.14 View Calendar 37](#_Toc300126430)

[4.2.15 View All Teams Page 38](#_Toc300126431)

[4.2.16 Manage Company Page 39](#_Toc300126432)

[4.2.17 Propose Project 41](#_Toc300126433)

[4.2.18 Leave Project Comment 44](#_Toc300126434)

[4.2.19 Create Company Account 45](#_Toc300126435)

[4.2.20 Publish Team Page 47](#_Toc300126436)

[4.2.21 Manage Milestones 48](#_Toc300126437)

[4.2.22 Change Project Status to Available 49](#_Toc300126438)

[4.2.23 Manage Team Page 50](#_Toc300126439)

[4.2.24 View All Projects 51](#_Toc300126440)

[4.3 Database Design 52](#_Toc300126441)

[4.3.1 Data Model 52](#_Toc300126442)

[52](#_Toc300126443)

[4.3.2 Data Dictionary 53](#_Toc300126444)

[4.4 Class Diagram 57](#_Toc300126445)

[4.5 User Interface Prototypes 58](#_Toc300126446)

[4.5.1 Admin Matching Page 58](#_Toc300126447)

[4.5.2 Home Page 59](#_Toc300126448)

[4.5.3 Sponsor Home Page 60](#_Toc300126449)

[4.5.4 Team Registration Page 61](#_Toc300126450)

[4.5.5 Team Page 62](#_Toc300126451)

[4.7 Test Cases 63](#_Toc300126452)

# 1. Document Information

1.1 Document Authors

| **Author’s Name** |
| --- |
| Anastasia Semionova |
| Matthew Schranz |
| Bryan Cohen |
| Edouard Davlatian |

* + 1. Document Approvals

| **Role** | **Approver Name** | **Signature** | **Date** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1.2 Revision History

| **Version** | **Date** | **Revision Notes** |
| --- | --- | --- |
| 0.1 | July 25 2011 | Initial draft |
|  |  |  |
|  |  |  |
|  |  |  |

1.3 Document Purpose

The purpose of this document is to provide a summarized description of the specifications for the Project Gemini application. This document is intended for client(s), user(s) and developer(s).

**2. Project Gemini Overview**

**2.1 Introduction/Goals**

Seneca is the largest college in Canada with 18,000 full time students, 100,000+ part time students and 2500 international students from over 80 countries. The college has 140+ 2 & 3 year diploma programs, 4 year bachelor degree programs and post graduate programs.

This project defines a web-based computer system to be used at Seneca’s School of Information and Communications Technology in the Faculty of Applied Science and Engineering Technology. Starting with the Seneca CPA program (Programming and Analysis), the goal is to enhance Seneca’s collaboration with industry partners in a way which promotes the students’ job readiness through practical, real-life design and project management experience. Industry partners are considered sponsor companies who benefit from a practicum and by having access to resources which reduce application backlogs while gaining exposure to potential hirable skills. There is no monetary cost to the sponsor company. This project is named “Gemini”

**2.2 Project Objectives**

Specific objectives include:

* Attract, log and track potential computer systems projects which can be utilized as a practicum for senior student project groups in Seneca’s CPA PRJ566 (hereafter referred to as PRJ566). The system could be opened to other Seneca practicum programs in the future.
* Streamline the legal and other administrative processes required for a customer to engage with Seneca and the student groups for PRJ566. This includes understanding and agreement to Intellectual Property conditions and other rights.
* Assist in the matching of student teams to possible industry projects.

**2.3 Stakeholders**

* Evan Weaver, Chair, School of Information and Communications Technology Faculty of Applied Science and Engineering Technology Seneca College of Applied Arts and Technology
* Mark Buchner and other Seneca professors associated with design practicum programs such as PRJ566
* Emile Ohan and other Seneca professors associated with the design implementation programs such as PRJ666
* IT Executives/Managers within “sponsor” companies who sign up for a practicum. This includes IT vendors, software developers, SMB or enterprise customers in any industry.
* Current and former Seneca students who participate in PRJ566 and have been asked to form teams.

**2.4 Business issues**

There are a number of existing business issues with the PRJ566 practicum process which this system seeks to resolve:

1. The scarcity of appropriate projects at the start of each semester when students cause significant delays which ultimately impede quality and cause extra sales work.
2. Procrastinated acceptance of legal and other administrative terms can cause projects to be needlessly delayed or even cancelled.
3. Customer satisfaction can be negatively affected because of lack of clear project status communications throughout the lifecycle of a project.
4. Lack of project information and obscurity of the matching process between potential projects and teams can cause project delays concerns and miscommunication.

**2.5 Solution outline**

The projects act as a solution to the above business problems by:

1. Representing a consistent, pervasive, secure and available portal which acts as publicity tool and a collection mechanism for project opportunities.
2. Including a workflow which forces potential sponsors to acknowledge legal terms and conditions before projects are registered.
3. Communicating project status in a secure and reliable way to sponsors.
4. Automate the matching process between student teams and sponsor projects.

**2.6 Project overview**

The system will streamline the process to connect a group of PRJ566 students to a potential Sponsor. It will act as the permanent host for the current semesters PRJ566 teams and act as an archive for past teams and projects. It will enable students to give descriptions about themselves and the team for interested Companies. Companies will be able to sign up for accounts to the site and provide information about their company and the project they want to be designed and implemented. Moreover, the Companies will be aware of their project status through their web page. Also, Teams and Companies will be able to rank each other based on their interest level and will be matched based on their preferences.

The system will have a Home Page where the description of the PRJ566 course will be displayed, and all the current projects and past projects can be viewed. From the Home Page a Public user (potential Sponsor) can register a project and later on, when an account is created by the administrator, can login to the Company Page. Also, from the Home Page, each team leader can login to their Team Page with credentials provided by site administrator. Moreover, the appointed site administrator (the professor who teaches the PRJ566 in the current semester) can login to the administration console to manage some site functionalities.

This function allows the Sponsor to view the common PRJ566 project life cycle on the weekly basis, allowing them to monitor the process. Also, the start and end dates for upcoming PRJ566 semesters are displayed, allowing Sponsors to plan their future projects’ timeline.

# 3. Project Gemini Architecture

**3.1 Technologies Used**

The following is the list of technologies that Project Gemini Application uses within its architecture:

* Eclipse and necessary plug-ins
* Java (jdk1.6. or later)
* Javascript and CSS
* MySQL
* GlassFish application server

3.2 Actors

**Public:** All users that do not have credentials. They can only view the randomly displayed past projects and can propose a new project, registering the company at the same time.

**Company Representatives:** All companies after the first approval of a proposed project will be provided with the credentials, and later on can login and use system for teams review, teams ranking, project status review, maintain project and maintain the page.

**Team Leaders:** Each team leader during the first weeks when class starts will be receiving the credentials that he/she can you to view all available projects, publish the team page, maintain team page, rank projects and update project status.

**Supervisor:** Each PRJ666 professor before the beginning of the semester will be provided with the credentials, and later when class starts he/she can login and use system to review all projects and change project status.

**Instructors:** Each PRJ566 professor before the beginning of a semester will be provided with the credentials, and later when class starts he/she can login and use the system to create accounts for teams and companies, approve proposed projects, initiate matching and notify everybody when it is time to do ranking.

**Site Administrator:** The appointed site administrator that is responsible for managing site accounts, review company comments and some responsibilities of a Superviser.

**Mailing System:** All parties are notified through e-mail

*\*Faculty: the other members of faculty that are interested in projects (not included in this PID)*

***NOTE:*** The following actors: *TeamLeader, Supervisor and Instructor*, have an **identifier** that helps determine their stream (PRJ566 or PRJ666), semester (winter (win), fall (fal) or summer (sum)) and year, and a **role** (a field in database) that defines the actor’s role (e.g. Supervisor, Instructor, TeamLeader).

3.3 System Major Functionalities

The system has the following major functions:

**Project Registration**

This function allows a company’s project to be pre-qualified as a project for PRJ566 students and if approved by the Instructor, the company's page is created where the company contact person can view and rank the teams. The legal agreement is accepted at the first step of the registration, to make a potential CompanyRep aware of any legal issues before he/she can start the registration process itself. A select number of mandatory fields must be filled in. The Instructor makes the final decision whether or not the particular project is suitable for PRJ566 needs. The CompanyRep has the ability to indicate certain project constraints such as timing (e.g. must be implemented by date), hardware (e.g. implemented on IBM iSeries), software (e.g. PHP), and architectural/other (e.g. transactional DB). When a project is complete, the CompanyRep has the opportunity to provide comments, which could be used by Seneca for publicity purposes and showcasing of prj projects. Password management functionality (reset password, forgot password etc) will be integrated.

**Team Page Creation**

This function allows a team leader to create the team page based on a template, after the instructor has created the account. The first time a team leader logs in the Create a Team Page appears allowing to upload any necessary pictures (e.g. team logo, team member’s pictures) and give short text descriptions (e.g. team description, team member short story). Spell check is integrated. When everything is uploaded and text is entered, the team leader publishes the page. From now on, when team leader logs in the Team Page appears skipping the creation step. From here all the offered projects can be viewed and ranked. The team page can be use to highlight certain constraints, such as timing (e.g. not available term 2), technology (e.g. limited to Apple), products (e.g .no Java), type (e.g. game).

**Project/Team Ranking**

This function allows Sponsors to rank teams for their project and Teams to rank projects according to their interests. A clever ranking and matching User Interface will be determined.

**Project-Team Matching**

This function is triggered by the Instructor. Moreover, it automates the matching process based on the rankings from the Teams and Sponsors. The result table can be edited by the Instructor. The function is integrated with an email system to automatically alert all affected parties (teams and companies).

The outline for the matching algorithm is provided in section 3.4.

**Calendar and Schedule**

This function allows the Sponsor to view the common PRJ566 project life cycle, allowing them to monitor the process. The PRJ team will have the ability to identify project milestones and keep manage the status. Also, the start and end dates for upcoming PRJ566 semesters are displayed, allowing Sponsors to plan their future projects’ timeline. A clever, easy-to-use UI will be designed.

**Series of Administration Consoles**

This function allows Instructor to approve company projects, create company account and company page (is generated automatically from a template upon account creation), create team accounts, finalize project-team matching process and change the status of a project. Also, the site administrator is able to manage Supervisor and Instructor accounts from his/her console. Moreover, the Supervisor administration console allows Instructor to manage project status.

**Authorization**

Only users with correct credential can login to the page. Every user once logged in is redirected to an appropriate page according to the user identifier/name. Each identifier will include the stream, section, semester and year, and also the specific role (student, company representative, professor or admin)

3.4. Matching pseudo-code

Following is the matching pseudo-code:

The company ranking **is a priority**. All company ranking and team ranking are stored in ArrayLists sorted by the ranking number from highest to lowest. First round: check the highest rank of every project and if there is no collision of interests (two companies rank the same team highest) match project with team and remove the project and team from ArrayList. In case there is a collision, check what project is ranked higher by that team and match team with that project. Continue until all projects are ranked.

Example:

P1, P2, P3 – projects

T1, T2, T3 - teams

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **P1** | **P2** | **P3** |  | **T1** | **T2** | **T3** |
| T1 | T3 | T1 |  | P1 | P2 | P3 |
| T2 | T1 | T2 |  | P3 | P3 | P2 |
| T3 | T2 | T3 |  | P2 | P1 | P1 |

Loop through all projects or till endLoop == false

Check what projects(companies) ranked highest

if (there is no collision) {

HashMap.put(key=P2, value=T3);

remove(P2);

remove(T3);

} else {

check what project this team (T1) ranked higher(by index);

HashMap.put(key=P1, value=T1);

remove(P1);

remove(T1);

}

if (only one team left){

HashMap.put(key = P3 (the last project ranked by team), value=T3)

endLoop = true;

}

End loop

End result should be:

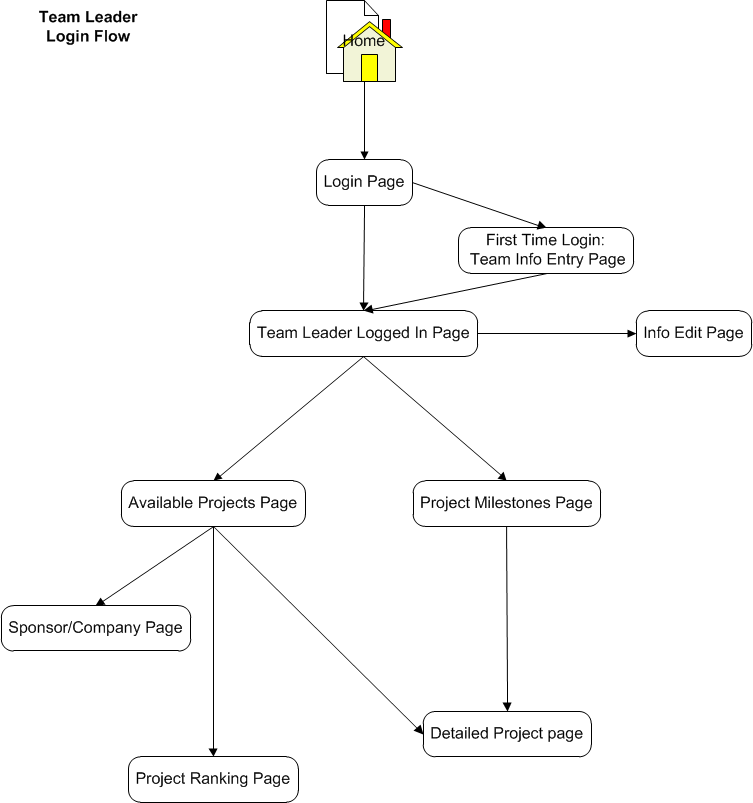
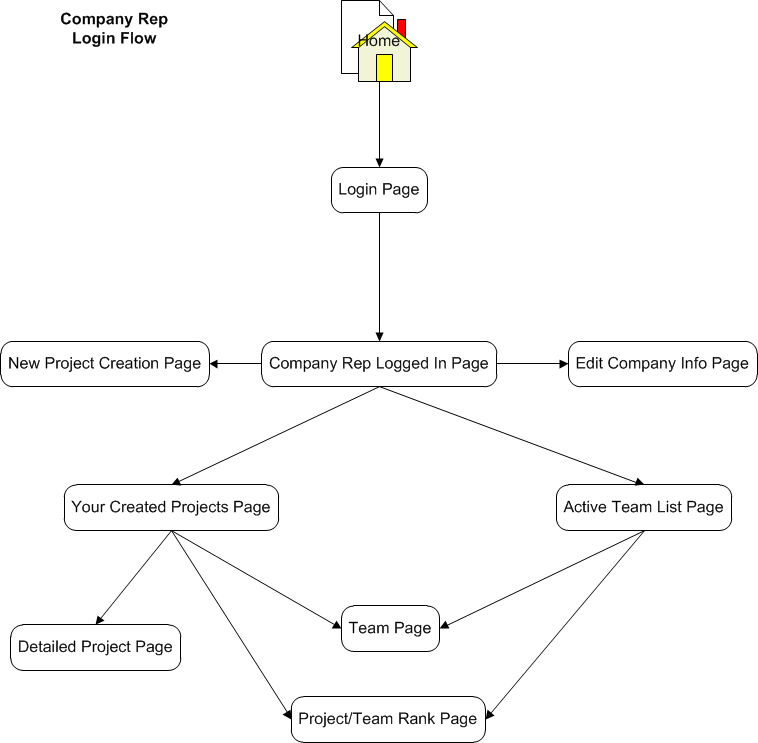
* P2 matched with T3
* P1 matched with T1
* P3 matched with T2

3.5 Design Trees

# 



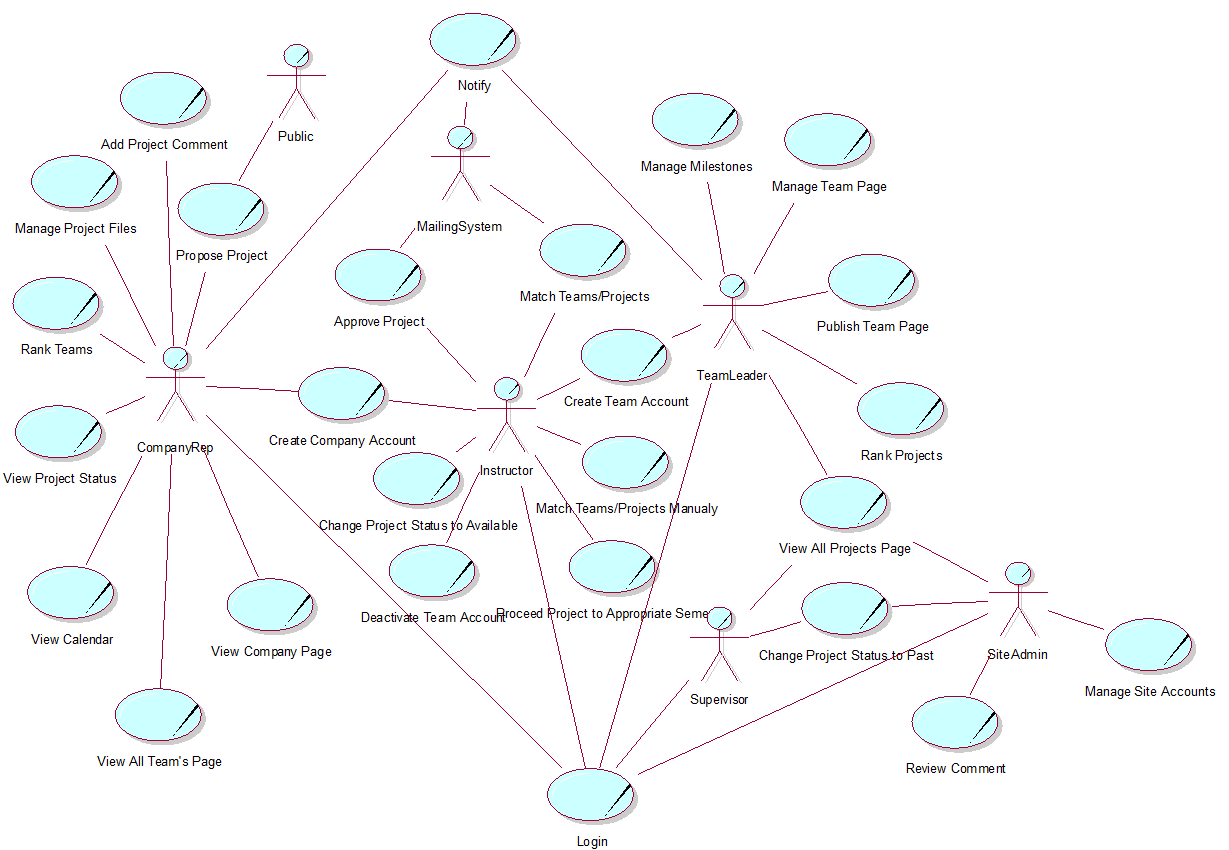


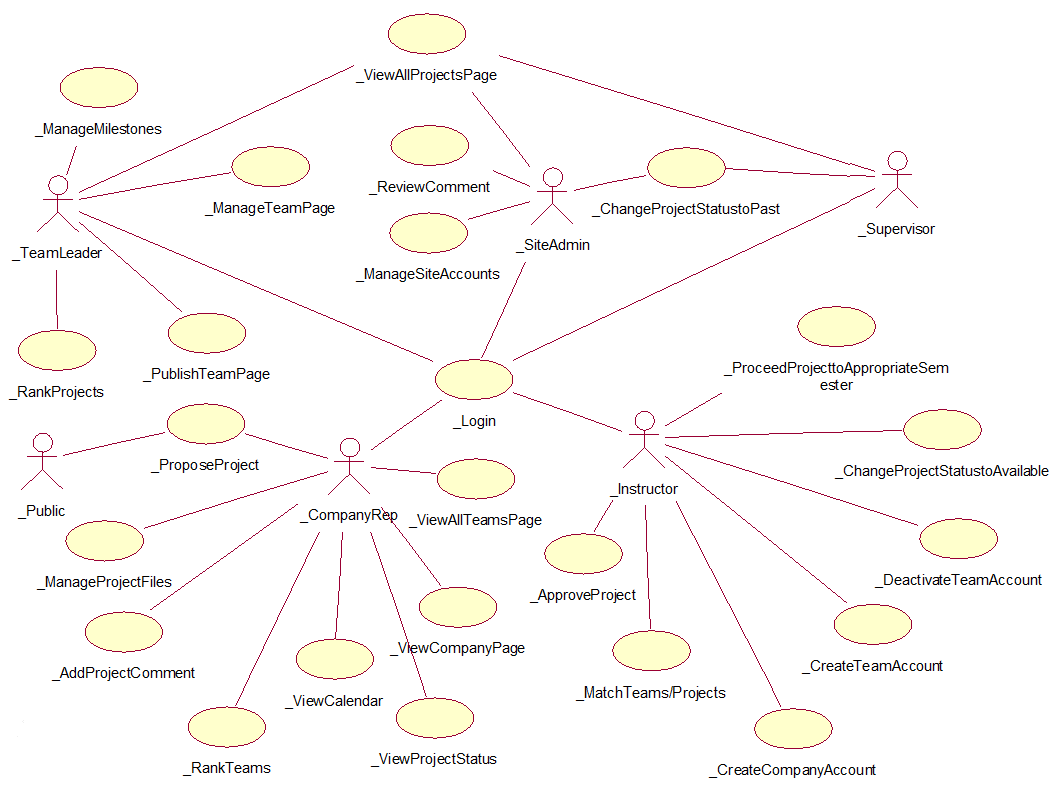


# 4. Business Process

4.1 Use Case Diagrams

4.1.1 Business Use Case Diagram

4.1.2 System Use Case Diagram



4.2 Process Matrix:

| **UC #** | **Business Process** | **Actor(s)** | **Notes/Process Options** | **Developer** |
| --- | --- | --- | --- | --- |
| 1 | Manage Project  Files | CompanyRep |  | Matthew |
| 2 | Rank Teams | CompanyRep |  | Anastasia |
| 3 | View Project Status | CompanyRep |  | Matthew |
| 4 | View Calendar | CompanyRep |  | Edouard |
| 5 | View All Teams Page | CompanyRep |  | Edouard |
| 6 | Manage Company Page | CompanyRep |  | Edouard |
| 7 | Propose Project | CompanyRep, Public |  | Edouard |
| 8 | Leave Project Comment | CompanyRep |  | Edouard |
| 9 | Create Company Account | Instructor |  | Edouard |
| 10 | Approve Project | Instructor |  | Anastasia |
| 11 | Create Team Account | Instructor |  | Matthew |
| 12 | Proceed Project to Appropriate Semester | Instructor |  | Anastasia |
| 13 | Match Teams/Projects | Instructor |  | Matthew |
| 14 | Change Project Status to Available | Instructor |  | Bryan |
| 15 | Deactivate Accounts | Instructor |  | Anastasia |
| 16 | View All Projects Page | Team Leader |  | Bryan |
| 17 | Manage Milestones | Team Leader |  | Bryan |
| 18 | Rank Projects | Team Leader |  | Anastasia |
| 19 | Publish Team Page | Team Leader |  | Bryan |
| 20 | Manage Team Page | Team Leader |  | Bryan |
| 21 | Change Project Status to Past | Supervisor,  Site Admin |  | Matthew |
| 22 | Manage Site Accounts | Site Admin |  | Matthew |
| 23 | Review Comment | Site Admin |  | Anastasia |
| 24 | Login | Instructor,  Team Leader,  CompanyRep,  Supervisor,  Site Admin |  | Anastasia |
| 25 | Match Team/Projects manualy | Instructor |  | Anastasia |

# 5. Design Documents

5.1 Business Rules

| **BR #** | **Business Rule** | **Use Case** |
| --- | --- | --- |
| 1 | Only projects approved by Instructor can become available for PRJ566 students. | UC10, UC07, UC14 |
| 2 | Company representative and Team Leader accounts can only be created by the Instructor. | UC11, UC09 |
| 3 | When particular semester is over, the Instructor must proceed the matched projects to appropriate semester | UC12 |
| 4 | All team and project ranking must be done more than once up to the matching initiation process. | UC18, UC02 |
| 5 | In case there are more projects available then teams, team leaders must rank the same amount of project as there are teams in the particular semester. | UC18 |
| 6 | Each team/project must be ranked with a different ranking number (sequentially). | UC18, UC02 |
| 7 | Instructor must be notified when a new project is registered. | UC10, UC07 |
| 8 | Before a company representative can register a new project and a company, he/she must acknowledge the Legal Agreement Contract. | UC07 |
| 9 | Company page must be created during the company account creation (generated from a template). | UC09 |
| 10 | Team Page must be created from a template. | UC19, UC05 |
| 11 | Team Leader must specify the team constraints when publishing the page. | UC19, UC05 |
| 12 | Instructor should be able to edit the matching results and all changes must be tracked. | UC13 |
| 13 | Company representative should be able to check project status. | UC03 |
| 14 | Team Leader should be able to update a milestone status. | UC03 |
| 15 | When teams are matched with the projects all parties (teams and companies) must be notified. | UC13 |
| 16 | Each matched project must have default milestones with the blank status. | UC17 |
| 17 | All created accounts must have a temporary password generated by the system, and it must be changed on the first log in. | UC11,UC16, UC19,UC21,  UC09 |
| 18 | Company representative should be able to leave comments upon project completion. | UC23, UC08 |
| 19 | Site administrator must be notified when a comment is left. | UC08 |
| 20 | Company comments can be viewed by site visitors only after appointed site administrator approval. | UC23, UC08 |
| 21 | Instructor should be able to notify all parties when it is time to do ranking. | UC18, UC02 |
| 22 | When a particular semester is over, the Instructor must deactivate the current team accounts | UC15 |
| 23 | Public should be able to view randomly displayed past projects on Site Home Page. |  |
| 24 | Team leader should be able to manage project milestones. | UC17 |
| 25 | Project information files can only be viewed by the company representative and the team matched to their project. | UC01 |
| 26 | Instructor and Supervisor accounts can only be managed by appointed site administrator. | UC21 |
| 27 | Only valid Seneca e-mail addresses are accepted for Instructor accounts. | UC21 |
| 28 | Team Leaders and Company Representatives should be able to do ranking more than once (overriding the old results). | UC02, UC18 |
| 29 | All Team member e-mails must be Seneca learn e-mails | UC19 |
| 30 | Supervisor should be able to view all the projects that match his/her identifier | UC16 |
| 31 | Supervisor and site administrator should be able to change the project status to Past | UC20 |
| 32 | Instructor account must be notified by the system when deactivated. | UC21 |
| 33 | Only appointed site administrator can manage site accounts (Instructor and Supervisor) | UC21 |
| 34 | If necessary, the appointed site administrator should be able to change Instructor and Supervisor name and e-mail before semester starts. | UC21 |
| 35 | The company ranking must be the priority during the matching process | UC13 |
| 36 | Team leader should be able to edit team’s page | UC20 |
| 37 | Only users with appropriate credentials can login and use the application | UC24 |
| 38 | Public and Company representative should be able to view the semester calendar with standard starting and ending dates and default project milestones | UC04 |
| 39 | Instructor should be able to match unmatched teams with projects manually | UC24 |
| 40 | Company Representative should be able to manage Company page | UC06 |

5.2. Use Case Specifications

5.2.1 Create Team Account

**Identifier: UC11**

**Brief Description:** This use case enables the actor, the administrator, to create a new team account.

**Section 1: Business Rule(s):**

BR#2: Company representative and Team Leader accounts can only be created by the Instructor.

BR#17: All created accounts must have a temporary password generated by the system, and it must be changed on the first log in.

**Section 2: All Scenarios (HD):**

**Scenario 1:** Create Team Account

**Preconditions:** Requestor is at the Create Team Account screen.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Enters required information and submits it. | Checks for class section, current semester and email.  Generates username with “prj566” prefix and then adds in the remaining based on current year, semester, section and number of team accounts already made for that section/year/semester combo. E.G. prj566fala05 (2011, Fall Semester, 4 other teams already registered).  Saves account information.  Requests acknowledgement. | currSemester, classSection, replyEmail |
| Acknowledges the message. | Sends email to provided email address notifying that account has been created with the credentials.  Actor is redirected to their home page. | email, userID, password |

**Successful Post Conditions:** Team account has been created and credentials are emailed.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Team** | userID, password, semester, section, teamId, email |

5.2.2 Match Teams/Projects

**Identifier: UC13**

**Brief Description:** This use case enables the actor, the Instructor, to match teams with projects and edit the results.

**Section 1: Business Rule(s):**

BR#12: Instructor should be able to edit the matching results and all changes must be tracked.

BR#15: When teams are matched with the projects all parties (teams and companies) must be notified.

BR#35: The company ranking must be the priority during the matching process

**Section 2: All Scenarios (HD):**

**Scenario 1:** Match teams/projects with no change.

**Preconditions:** System displays the Match Team/Projects screen.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Initiates the matching. | System matches teams and projects and displays the match results. | teamId, projectId, projectRank, teamRank |
| Accepts Matches. | Saves match information.  Requests Acknowledgment. | teamID, projectID, matchStatus |
| Acknowledges. | Sends notification emails to all teams and sponsors matched. | teamEmail, sponsorEmail, |

**Successful Post Conditions:** All matches are accepted and notifications are emailed.

**Scenario 2:** Match teams/projects with editing.

**Preconditions**: System displays the Match Team/Projects screen.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Initiates the matching. | System matches teams and projects and displays the match results. | teamId, projectId, projectRank, teamRank |
| Decides to edit the matches. | Displays all projects that have not been matched. | projectId, unrankedProjects |
| Selects a new project. | Saves the updated project match information. | projectId, teamID |
| Accepts Matches. | Saves match information.  Requests Acknowledgment. | teamID, projectID, matchStatus |
| Acknowledges. | Sends notification emails to all teams and sponsors matched. | teamEmail, sponsorEmail, |

**Successful Post Conditions:** All matches are accepted and notifications are emailed.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Team** | teamId, teamEmail |
| **Project** | projectId |
| **TeamRanking** | TeamId, projectID, projectRank |
| **ProjectRanking** | SponsorID, projectID, teamRank |

5.2.3 View Project Status

**Identifier: UC03**

**Brief Description:** This use case enables the actor, CompanyRep, to view the status of a submitted project.

**Section 1: Business Rule(s):**

BR#13: Company Representative should be able to check project status.

BR#14: Team Leader should be able to update a milestone status.

**Section 2: All Scenarios (HD):**

**Scenario 1: View Project Status**

**Preconditions: System displays the View Projects screen.**

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Chooses a project to view. | System displays the selected project milestones with status | projectID, projectName,  milestoneId,  milestone,  status |

**Successful Post Conditions:** Project status has been successfully displayed.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Project** | projectID, projectName |
| **Milestones** | milestoneId, milestone, status |

5.2.4 Manage Project Files

**Identifier: UC01**

**Brief Description:** This use case enables the actor, CompanyRep, to manage the files associated with a project.

**Section 1: Business Rule(s):**

BR#25: Project files can only be viewed by the company rep and the team matched to their project.

**Section 2: All Scenarios (HD):**

**Scenario 1:** Upload File

**Preconditions:** System is displaying the upload file screen.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Enters the required information and file and submits it. | Checks for file description, file name and uploaded file.  Saves file information.  Requests Acknowledgement. | fileDescription, fileName, fileID, projectID |
| Acknowledges. | Actor is redirected to the project's page. | projectID |

**Successful Post Conditions:** File has successfully been uploaded and added to the project.

**Scenario 2: Remove File**

**Preconditions: System is displaying the remove file screen.**

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Chooses the file and requests for removal. | Removes the file information.  Updates the project information.  Requests Acknowledgement. | fileID, projectID |
| Acknowledges. | Actor is redirected to the project's page. | projectID |

**Successful Post Conditions:** File has been successfully removed and project information is updated.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **ProjectFile** | fileDescription, fileName, uploadedFile, fileID |
| **Project** | projectID |

5.2.5 Manage Site Accounts

**Identifier: UC21**

**Brief Description:** This use case enables the actor, SiteAdmin, to add, remove and change the status of site accounts.

**Section 1: Business Rule(s):**

BR#17: All created accounts must have a temporary password generated by the system, and it must be change on first log in.

BR#26: Instructor and Supervisor accounts can only be managed by appointed site administrator.

BR#27: Only valid Seneca email addresses are accepted for Instructor accounts.

BR#32: Instructor account must be notified by the system when deactivated.

BR#33: Only appointed site administrator can manage site accounts.

BR#34: If necessary, the appointed site administrator should be able to change Instructor and Supervisor name and e-mail before semester starts.

**Section 2: All Scenarios (HD):**

**Scenario 1:** Create Account

**Preconditions:** System is displaying the Create Instructor Account screen.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Enters the name and email address. Chooses the account type along with other required information and submits. | The system validates the e-mail and saves the new account and notifies the actor that new account has been created.  Requests acknowledgement. | FirstName, lastName, email, systemsStream |
| Acknowledges the message. | Sends email to provided email address notifying that the account has been created with the credentials.  Actor is redirected to their home page. | email, userID, password |

**Successful Post Conditions:** A new account has successfully been created and credentials have been emailed.

**Scenario 2:** Change Account Status

**Preconditions:** System is displaying the Change Account Status screen.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Chooses the account and status and submits. | Updates account information to reflect new account status.  Requests acknowledgement. | accountID, accountStatus |
| Acknowledges the message. | Sends email to provided email address notifying that the account status has been changed.  Actor is redirected to their home page. | email |

**Successful Post Conditions:** An account's status has successfully been changed and notified.

**Scenario 3:** Remove Account

**Preconditions:** System is displaying the Remove Account screen.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Chooses the account and submits it for removal. | Updates account information to reflect new account status.  Requests acknowledgement. | accountID, accountStatus |
| Acknowledges the message. | Sends email to provided email address notifying that the account has been removed.  Removes the account information from the system.  Actor is redirected to their home page. | accountID, email |

**Successful Post Conditions:** An account has successfully been removed and notified.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Instructor** | systemsStream, instructorNum |
| **Supervisor** | systemsStream, supervisorNum |
| **User** | firstName, lastName, email, userIdentifier, userID, password, accountStatus, userRole |

5.2.6 Change Project Status to Past

**Identifier: UC20**

**Brief Description**: This use case enables the actors, Supervisor and SiteAdmin, to change the status of a completed project to past.

**Section 1: Business Rule(s):**

BR#31: Supervisor and Site Administrator should be able to change the project status to Past.

**Section 2: All Scenarios (HD):**

**Scenario 1:** Change Status to Past

**Preconditions:** System is displaying the Change Project Status screen.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Chooses a project and submits it for status change. | Updates the project information with new status.  Requests Acknowledgement. | projectID, projectStatus |
| Acknowledges the message. | Actor is redirected to their home page. |  |

**Successful Post Conditions:** Project status has successfully been changed to a past project.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Project** | ProjectID, projectStatus |

5.2.7 Rank Projects

**Identifier: UC18**

**Brief Description:** This use case enables the actor, the team leader, to rank projects.

**Section 1: Business Rule(s):**

BR#04: All team and project ranking must be done more than once up to the matching initiation process.

BR#05: In case there are more projects available then teams, team leaders must rank the same amount of project as there are teams in the particular semester

BR#06: Each team/project must be ranked with a different ranking number (sequentially).

BR#21: Instructor should be able to notify all parties when it is time to do ranking

BR#28: Team Leaders and Company Representatives should be able to do ranking more than once (overriding the old results).

**Section 2: All Scenarios (HD):**

**Scenario 1:** Rank Projects

**Preconditions**: The system is displaying a Team Home Page screen

Team Leader has been notified that it is time to rank

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| The actor selects the “rank projects” option | The system displays the rank project screen and checks if it is not the first time ranking and displays previous ranking also |  |
| The actor enters a ranking number for a project | The system validates the ranking number | projectID, rankingNumber |
| The actor repeats entering ranking until the appropriate number of projects are ranked | The system validates the ranking number and total number of ranked projects | projectID, rankingNumber |
| The actor submits the ranking | The system validates information, and asks for acknowledgement | projectID,  rankingNumber |
| The actor confirms the ranking | The system saves the ranking | teamId, projectId, rankingNumber |

**Successful Post conditions:** The projects are ranked and ranking is saved.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Teams** | teamId |
| **Company** | companyId |
| **TeamProjectRanking** | rankingNumber |

5.2.8 Rank Teams

**Identifier: UC02**

**Brief Description:** This use case enables the actor, the company representative, to rank all teams.

**Section 1: Business Rule(s):**

BR#04: All team and project ranking must be done more than once up to the matching initiation process.

BR#06: Each team/project must be ranked with a different ranking number (sequentially).

BR#21: Appointed site administrator should be able to notify all parties when it is time to do ranking

BR#28: Team Leaders and Company Representatives should be able to do ranking more than once (overriding the old results).

**Section 2: All Scenarios (HD):**

**Scenario 1:** Rank Teams

**Preconditions:** The system is displaying a Company Home Page screen

Company representative has been notified that it is time to rank

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| The actor selects the “rank teams” option | The system displays rank teams screen and checks if it is not the first time ranking and displays previous ranking also |  |
| The actor enters a ranking number for a team | The system validates the ranking number | teamID, rankingNumber |
| The actor repeats entering ranking until all teams are ranked | The system validates the ranking number | teamID, rankingNumber |
| The actor submits the ranking | The system validates information, and asks for acknowledgement | teamID,  rankingNumber |
| The actor confirms the ranking | The system saves the ranking | CompanyId, teamId, rankingNumber |

**Successful Post conditions:** The teams are ranked and ranking is saved.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Teams** | teamId |
| **CompanyTeamRanking** | rankingNumber |
| **Company** | companyId |

5.2.9 Approve Project

**Identifier: UC10**

**Brief Description:** This use case enables the actor, the Instructor, to approve a project for a registered and new company.

**Section 1: Business Rule(s):**

BR# 1: Only projects approved by Instructor can become available for PRJ566 students.

BR# 7: Instructor must be notified when a new project is proposed.

**Section 2: All Scenarios (HD):**

**Scenario 1:** Approve project for a registered company

**Preconditions:** The system is displaying a Instructor Home Page screen

An Instructor was notified that new project has been proposed

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| The actor selects the “view all proposed projects” option | The system displays all projects pending for approval |  |
| The actor selects a project | The system displays company information: company name, contact name, and phone number; and project information: project description, project constraints | companyId, companyName, companyContact,  companyPhoneNumber, projectId, projectDescription, projectConstraints |
| The actor approves the project | System changes the project status to approved and notifies the actor that a new project has been approved | companyId, companyName, companyContact,  companyPhoneNumber, projectId, projectDescription, projectConstraints,  projectStatus |
| The actor accepts the acknowledgment | The system checks the company status and redirects the actor to the Site Administrator Home Page, and sends e-mail to the company | companyStatus |

**Successful Post conditions**: The proposed project is approved

**Scenario 2:** Approve project for a new company

**Preconditions:** The system is displaying a Instructor Home Page screen

An Instructor was notified that new project has been proposed

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| The actor selects the “view all proposed projects” option | The system displays all projects pending for approval |  |
| The actor selects a project | The system displays company information: company name, contact name, and phone number; and project information: project description, project constraints | companyId, companyName, companyContact,  companyPhoneNumber, projectId, projectDescription, projectConstraints |
| The actor approves the project | System changes the project status to approved and notifies the actor that a new project has been approved | companyId, companyName, companyContact,  companyPhoneNumber, projectId, projectDescription, projectConstraints,  projectStatus |
| The actor accepts the acknowledgment | The system checks the company status and redirects the actor to the Create Company Account Page | companyStatus. companyId |

**Successful Post conditions:** The proposed project is approved

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Company** | companyId, companyName, companyContact, companyPhoneNumber, companyStatus |
| **Projects** | projected, projectDescription, projectConstraints, projectStatus |

5.2.10 Login

**Identifier: UC24**

**Brief Description:** This use case enables the actor, the site administrator/team leader/company representative, to login.

**Section 1: Business Rule(s):**

BR#37: Only users with appropriate credentials can login and use the application

**Section 2: All Scenarios (HD):**

**Scenario 1:** Login

**Preconditions:** The system is displaying a Home Page screen

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| The actor enters the credentials | The system checks the credentials entered, checks the actor’s role and identifier, and redirects user to appropriate home page | userId, password, userRole, identifier |

**Successful Post conditions:** The user is logged in

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **user** | userId, userPassword, userRole, identifier |

5.2.11 Proceed Projects to Appropriate Semester

**Identifier: UC12**

**Brief Description:** This use case enables the actor, the Instructor, to proceed matched projects to appropriate semester.

**Section 1: Business Rule(s):**

BR#3: When a particular PRJ566 semester is over, the Instructor must proceed the matched projects to appropriate semester

**Section 2: All Scenarios (HD):**

**Scenario 1:** Deactivate Accounts

**Preconditions:** The system is displaying a Instructor Home Page screen

A particular PRJ566 semester is over

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| The actor selects the “proceed projects” option | The system displays Proceed Projects screen with the list of all matched projects for the semester and list available semesters(fall, winter, summer) and year | projectIdentifier |
| The actor selects semester for a project that it needs to be proceeded to, and repeats it until all projects are proceeded, and submits | The system changes the project identifier (the semester part and course part), changes the project status to proceeded and notifies the actor of the changes | projectNum, projectIdentifier |
| The actor confirms the change | The system saves the changes and redirects the actor to the Instructor Home Page |  |

**Successful Post conditions:** The matched projects are proceeded to the appropriate semester.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Projects** | projectNum, projectIdentifier |

5.2.12 Review Comment

**Identifier: UC23**

**Brief Description:** This use case enables the actor, the site administrator, to review a comment.

**Section 1: Business Rule(s):**

BR#18: Company representative should be able to leave comments upon project completion.

BR#20: Company comments can be viewed by site visitors only after appointed site administrator approval.

**Section 2: All Scenarios (HD):**

**Scenario 1:** Review comment

**Preconditions:** The system is displaying a Site Administrator Home Page

A site administrator has been notified that a comment has been left

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| The actor selects the “view all pending comments” option | The system displays all project comments pending for approval | projectId, commentId, comment |
| The actor makes a selection whether to approve the project |  | projectId, commentId, comment |
| The actor repeats making selection for all pending comments and submits | System changes the project’s comment status to approved or not approved and notifies the actor that comments statuses have been processed | projectId, commentId,  commentStatus |
| The actor accepts the acknowledgment | The system saves the changes and redirects actor to the Site Administrator Home Page | projectId, commentId,  commentStatus |

**Successful Post conditions:** The project’s comment has been reviewed

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Projects** | projectId |
| **Comments** | commentId, commentStatus, comment |

5.2.13 Deactivate Accounts

**Identifier: UC15**

**Brief Description:** This use case enables the actor, the site administrator, to deactivate team accounts.

**Section 1: Business Rule(s):**

BR#22: When a particular PRJ566 semester is over, the Instructor must deactivate all current team accounts

**Section 2: All Scenarios (HD):**

**Scenario 1:** Deactivate Accounts

**Preconditions:** The system is displaying a Site Administrator Home Page screen

A particular PRJ566 semester is over

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| The actor selects the “deactivate team accounts” option | The system displays deactivate accounts screen with the list of all current teams |  |
| The actor selects all teams accounts that need to be deactivated and submits for deactivation | The system sets all selected accounts status to inactive and informs the actor what team accounts were deactivated | teamId, teamStatus |
| The actor confirms the massage | The system redirects the actor to the Site Administrator Home Page |  |

**Successful Post conditions:** The appropriate team accounts are deactivated.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Teams** | teamId, teamStatus |

5.2.14 View Calendar

**Identifier: UC04**

**Brief Description:** This use case enables the Company Rep to view the semester calendar.

**Section 1: Business Rule(s):**

BR#38: Company representative should be able to view the semester calendar with standard starting and ending dates and default project milestones

**Section 2: All Scenarios (HD):**

View Calendar

**Scenario 1:** View Calendar

**Preconditions:** Company Rep is logged in and is on their home page.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Selects the View Semester Calendar | Displays a new page that lists all the important dates for the current semester and default project milestones. | SemesterStartDate,  SemesterEndDate,  defaultMilestoneId,  defaultMilestones,  dueDate |

**Successful Post-Condition:** The calendar is displayed with all the events and their respective dates.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Semester** | SemesterID, SemesterStartDate, SemesterEndDate, |
| **Milestones** | defaultMilestoneId, defaultMilestone, dueDate |

5.2.15 View All Teams Page

**Identifier: UC05**

**Brief Description:** This use case enables the Company Rep to view the current semester’s available teams.

**Section 1: Business Rule(s):**

BR#10: Team Page must be created from a template.

BR#11: Team Leader must specify the team constraints when publishing the page.

**Section 2: All Scenarios (HD):**

**Scenario 1:** View Active Teams Page

**Preconditions:** The company rep must be logged in and on the company user home page.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Selects View Active Team List Page. | New Page is displayed with a list of all the active teams for the current semester. | TeamID,  TeamStatus,  TeamName,  TeamDesc |

**Successful Post-Condition:** A list of the active teams is displayed.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Team** | TeamID, TeamStatus, TeamName, TeamDesc |

5.2.16 Manage Company Page

**Identifier: UC06**

**Brief Description:** This use case enables the Company Rep to edit information on company page.

**Section 1: Business Rule(s):**

BR#40: Company Representative should be able to manage Company page

**Section 2: All Scenarios (HD):**

Edit Company Page

**Scenario 1:** Edit Company Page

**Preconditions:** Company Rep is logged in on their home page.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Selects Edit Company Info from menu. | A new page is displayed with an editable form for company info. | companyID |
| Edits required info and selects Save Changes. | Checks for required info. Saves the changes.  Displays an option to view the new page. | companyName,  companyLogo,  companyWeb,  companyDesc,  companyRepID,  companyRepEmail,  companyRepPhone |
|  |  |  |

**Successful Post-Condition:** Company page is edited.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Company** | companyID, companyName, campanyDesc, companyLogo, companyWeb, companyRep:CompanyRep |
| **CompanyRep** | companyRepID, companyRepName, companyRepEmail, companyRepPhone |

5.2.17 Propose Project

**Identifier: UC07**

**Brief Description**: This use case enables the actor, CompanyRep and Public, to propose a new project for approval by the administrator.

**Section 1: Business Rule(s):**

BR# 1: Only projects approved by appointed site administrator can became available for PRJ566 students.

BR# 7: Appointed site administrator must be notified when a new project is registered.

BR# 8: Before a company representative can register a company and a new project, he/she must acknowledge the Legal Agreement Contract.

**Section 2: All Scenarios (HD):**

Propose Project by CompanyRep

Propose Project by Public

**Scenario 1:** Propose Project by CompanyRep

**Preconditions:** User must be logged in with a company account.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Selects new project from menu | New page is displayed with a copy of the contract terms |  |
| Acknowledges the contract terms | Acceptance of terms is logged as well as the Company Rep ID, and new Project ID, and Date/Time. A form for user entry is displayed on the page. | companyRepID, projectID, termsAccepted, dateTimeAccepted |
| Enters the required project information and Submits for approval. | System checks for required fields: Project Name, Short Description (50 chars), Detailed description, if any files are to be uploaded and the Project Liaison Contact info. Data is saved along with acceptance of terms.  System then checks for optional fields like: Hardware requirements, Software Requirements, Constraints, Etc... Then saves this data.  An email is sent to admin informing them of new project application.  A notification is displayed to the user that the project has been submitted for approval. | projectName, projectShortDesc, projectDetailDesc, projectFileList, projectLiaisonName, projectLiaisonPhone, projectLiaisonEmail, projectHWReqs, projectSWReqs, projectConstraints, projectStatus = pending |
| Acknowledges notification | System is redirects user to their Sponsor homepage. |  |

**Successful Post conditions:** A new pending project is created.

**Scenario 2:** Propose Project by Public

**Preconditions:** User must have accepted the contract terms and all of the required company information has been filled out.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Enters the required information for the project and submits the application. | System checks for required project fields.  Data is saved along with acceptance of terms and the Company info.  An email is sent to Administrator informing them of new Sponsor and project application.  A notification is displayed to the user that their application has been submitted for approval. | projectID, projectName, projectShortDesc, projectDetailDesc, projectFileList, projectLiaisonName, projectLiaisonPhone, projectLiaisonEmail, projectStatus = pending companyRepID, companyRepName, companyRepPhone, companyRepEmail, companyID, companyName, companyWeb, companyLogo, companyStatus = pending, termsAccepted, dateTimeAccepted |
| Acknowledges notification | System redirects user to homepage. |  |

**Successful Post conditions:** A new application is created.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Company** | companyID, companyName, companyWeb, companyLogo, companyStatus, companyRep.CompanyRep, |
| **CompanyRep** | companyRepID, companyRepName, companyRepPhone, companyRepEmail, |
| **Project** | projectID, projectName, projectShortDesc, projectDetailDesc, projectFileList.UploadedFile, projectLiaison.ProjectLiaison, projectHWReqs, projectSWReqs, projectConstraints, projectStatus |
| **ProjectLiaison** | projectLiaisonID, projectLiaisonName, projectLiaisonPhone, projectLiaisonEmail |
| **Term Agreement** | termsID, projectID.Project, companyRepID.CompanyRep, termsAccepted, dateTimeAccepted |

5.2.18 Leave Project Comment

**Identifier: UC08**

**Brief Description:** This use case allows the company rep to add a comment to completed project.

**Section 1: Business Rule(s):**

BR#18: Company representative should be able to leave comments upon project completion.

BR#19: Site administrator must be notified when a comment is left.

BR#20: Company comments can be viewed by site visitors only after appointed site administrator approval.

**Section 2: All Scenarios (HD):**

Add Comment to Project

**Scenario 1:** Add Comment to project

**Preconditions:** Company Rep has to be logged in on their home page.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Selects view your projects from menu. | Page is displayed with a list of the company’s projects. | Set of Projects. |
| Selects a completed project. | Page is displayed with the completed project.  An option is added to the page to leave a comment. | projectStatus |
| Select the option to add a comment. | A new page is displayed with a form for text entry. |  |
| Enters comment and selects submit. | Saves the comment and a notification of the comment is sent to the associated course Admin/Instructor.  A notification is displayed that the comment has been added to the project. | projectComment, |
| Acknowledges the notification. | Detailed Project Page is displayed. |  |

**Successful Post-Condition:** A new pending comment is added to a completed project.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Project** | projectID, projectStatus, projectComment |

5.2.19 Create Company Account

**Identifier: UC09**

**Brief Description:** This use case enables the admin to create a new company/sponsor/rep account.

**Section 1: Business Rule(s):**

BR#2: Company Representative and Team Leader accounts can only be created by the appointed site administrator.

BR# 9: Company page must be created during the company account creation (generated from a template).

BR#17: All created accounts must have a temporary password generated by the system, and it must be change on first log in.

**Section 2: All Scenarios (HD):**

Create new company account

**Scenario 1:** Create new company account

**Preconditions:** An application must have been submitted by a sponsor/company rep. An email was sent to the admin informing them of the new application. Admin has logged into the website.

| **Actor** | **System** | **Data Used** |
| --- | --- | --- |
| Selects option to go to Pending Applications Page | New page is displayed with a list of pending applications. | List of pending applications |
| Selects an application from list. | Displays detailed information about the application and the proposed project. | All info pertaining to application and proposed project. |
| Selects Approve. | Changes status of application to active.  Changes status of project to active.  A new temporary password is generated depending on what the semester is.  A new company page is created.  An email notification is sent to the applicant that their account is now active with their temporary password and that their project has been approved.  Notification is displayed to user that a new account was created. | companyStatus = active  projectStatus = active  companyRepPassword |
| Acknowledges notification | System Redirects user to Pending applications page. |  |

**Successful Post conditions:** A new company account is created and associated project is set to active.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| Class | Attribute of class |
| Company | companyID, companyName, companyWeb, companyLogo, companyStatus, companyRep.CompanyRep, |
| CompanyRep | companyRepID, companyRepName, companyRepPhone, companyRepEmail, companyRepPassword |
| Project | projectID, projectName, projectShortDesc, projectDetailDesc, projectFileList.UploadedFile, projectLiaison.ProjectLiaison, projectHWReqs, projectSWReqs, projectConstraints, projectStatus |

5.2.20 Publish Team Page

**Identifier: UC19**

**Brief Description:** This use case enables the actor, a team leader, to initially create their team page for the PRJ566 semester.

**Section 1: Business Rule(s):**

BR#10: Team Page must be created from a template.

BR#11: Team Leader must specify the team constraints when publishing the page.

BR#17: All created accounts must have a temporary password generated by the system, and it must be change on first log in.

BR#29: All team member e-mails must be Seneca learn e-mails

**Section 2: All Scenarios (HD):**

**Scenario 1:** Create Team Page

**Preconditions:** A log-in screen is displayed

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| Logs in for the first time | Displays a form for changing a password |  |
| Enters in the old password and the new password twice | Validates the new password. Displays form for team information. | oldPassword, newPassword |
| Enters in name, description, an image, and an email for each team member as well as a team name and team logo | Validates the information and displays the newly created team page | memberName, memberImage, memberDesc,  memberEmail, teamName, teamLogo |

**Successful Post condition:** The team page is created and displayed with all the information.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Team** | password, name, logo |
| **Member** | name, image, desc, email |

5.2.21 Manage Milestones

**Identifier: UC17**

**Brief Description:** This use case enables the actor, a team leader, to create and edit the status of their milestones for their project.

**Section 1: Business Rule(s):**

BR#16: Each project must have milestones with the appropriate status.

BR#24: Team leader should be able to customize project milestones.

**Section 2: All Scenarios (HD):**

**Scenario 1:** Create Milestone

**Preconditions:** The Team Page is displayed

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| Selects manage milestones | Retrieve milestones and display on screen | milestone, date, status |
| Selects New Milestone | Retrieve and display form to enter in new milestones. |  |
| Enters in milestone name, date, and current status and submits | Validates the information and stores the milestone screen is redisplayed | milestone, date, status |

**Successful Post condition:** Milestone is created and the page is displayed with all updated information.

**Scenario 2:** Edit Milestone

**Preconditions:** The Team Page is displayed

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| Selects manage milestones | Retrieve milestones and display on screen | milestone, date, status |
| Select new status for a milestone and then submits | Stores new milestone status and redisplays milestone screen | milestone, date, status |

**Successful Post condition:** Milestone is edited and the page is displayed with all updated information.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| Milestone | name, date, status |

5.2.22 Change Project Status to Available

**Identifier: UC14**

**Brief Description:** This use case enables the actor, an Instructor, to change the status of a project.

**Section 1: Business Rule(s):**

BR#1: Only projects approved by Instructor can become available for PRJ566 students.

**Section 2: All Scenarios (HD):**

**Scenario 1:** Change Project Status

**Preconditions:** The Instructor's home page is displayed

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| Selects Change Project Status | Retrieves and displays a list of all approved projects | company, shortDesc, projectStatus |
| Selects a project | Retrieves the full description, name, and company of the project and displays | company, fullDesc |
| Changes project status to available | Saves the status of the project and emails the sponsor the update. Redisplays the list of all approved projects | sponsorEmail, projectStatus |

**Successful Post condition:** The project's status is updated and an e-mail has been sent to the sponsor to notify them of the change.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Sponsor** | email |
| **Project** | company, shortDesc, fullDesc, status |

5.2.23 Manage Team Page

**Identifier: UC20**

**Brief Description:** This use case enables the actor, a team leader, to manage their team page

**Section 1: Business Rule(s):**

BR#36:Team leader should be able to edit team’s page

**Section 2: All Scenarios (HD):**

**Scenario 1:** Edit Team Information

**Preconditions:** The Team Page is displayed

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| Selects Edit | Displays the template for creating a team page with all current information displayed | memberName, memberImage, memberDesc,  memberEmail, teamName, teamLogo, constraints |
| Updates any team information as necessary and submits | Saves the new information and redisplays the team page | memberName, memberImage, memberDesc,  memberEmail, teamName, teamLogo, constraints |

**Successful Post condition:** Team Information is updated and displayed.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Team** | name, logo, constraints |
| **Member** | name, image, desc, email |

5.2.24 View All Projects

**Identifier: UC16**

**Brief Description:** This use case enables the actor, a TeamLeader and Supervisor/Site Admin, to view any project.

**Section 1: Business Rule(s):**

BR#30: Supervisor should be able to view all the projects that match his/her identifier

**Section 2: All Scenarios (HD):**

**Scenario 1:** View a Project

**Preconditions:** The Team Page is displayed

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| Selects View Projects | Checks the actor identifier and retireves a list of appropriate current projects and their short descriptions and displays on screen | Identifier,  projectStatus, projectName, shortDesc, company |
| Select a project | Retrieve the full description for the project and displays on screen | fullDesc, projectName, company |

**Successful Post condition:** Actor is viewing a project.

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Project** | status, name, shortDesc, fullDesc, company |
| **User** | Identifier |

5.2.25 Match Team/Projects manually

**Identifier: UC25**

**Brief Description:** This use case enables the actor, an Instructor, to match team/projects manually.

**Section 1: Business Rule(s):**

BR#39: Instructor should be able to match unmatched teams with projects manually

**Section 2: All Scenarios (HD):**

**Scenario 1:** Match Team/Projects manually

**Preconditions:** The Instructor Home Page is displayed

|  |  |  |
| --- | --- | --- |
| **Actor** | **System** | **Data Used** |
| Selects “Match Team/Projects manually” | Displays the Match Team/Projects Manually page with the list of all unmatched projects and list of unmatched teams | teamId, status |
| Selects a team from the team list beside the project | Disables the selected project from the list | projectId |
| Repeats selecting teams for the projects until all projects are matched and submits | Displays the selected matching and asks for approval | teamId, projectId |
| Acknowledges | Saves the matching and notifies all matched teams and companies | companyEmail,  teamEmail,  projectId, teamId |

**Successful Post condition:** Team/Projects are matched manually

**Section 3: Classes/Attributes identified:**

|  |  |
| --- | --- |
| **Class** | **Attribute of class** |
| **Project** | projectId |
| **Team** | teamId, teamEmail, status |
| **Company** | companyEmail |

5.3 Database Design

5.3.1 Data Model



5.3.2 Data Dictionary

**Team**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Member Name** | **Description** | **Type** | **Sample Date** | **Default Value** | **Mandatory?** | **Unique?** | **PK** | **FK** |
| teamId | Team ID | int | 1 | Auto-increase | Y | Y | Y |  |
| teamIdentifier | Team idenftifier  (stream+ section + semester+year+  \_team#) | varchar(16) | prj566ASum2011\_1 |  | Y | Y |  |  |
| teamEmail | Group team e-mail | varchar(60) | asemionova1@learn.senecac.on.ca;mjschraz@learn.senecac.on.ca |  | Y |  |  |  |
| status | Active or inactive | boolean | true | true | Y |  |  |  |
| teamName | Team name | varchar(20) | BEAM Solutions |  | Y |  |  |  |
| teamLogo | Team logo image | blob | N/A |  |  |  |  |  |
| constraints | Team constraints | varchar(100) | Java, MySql |  | Y |  |  |  |
| projectID | Project the team was matched with | int | 1 | 0 |  |  |  | Y |
| userId | System user id | int | 1 |  | Y |  |  | Y |

**Project**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Member Name** | **Description** | **Type** | **Sample Date** | **Default Value** | **Mandatory?** | **Unique?** | **PK** | **FK** |
| projectId | Project ID | int | 1 | Auto-increase | Y | Y | Y |  |
| prjIdentifier | Project idenftifier  (stream+semester+year) | varchar(13) | prj566Sum2011 |  | Y |  |  |  |
| status | Pending, approved, available, proceeded, matched or past | char(2) | PE, AP, AV, PR, MA, or PA | PE | Y |  |  |  |
| prjName | Project name | varchar(20) | Gemini |  |  |  |  |  |
| description | Project description | varchar(500) | PRJ566 project registration |  | Y |  |  |  |
| prjConstraints | Constraints for the project | varchar(250) | Java, Oracle |  | Y |  |  |  |
| agreementDate | When legal agreement was reviewed | datetime | July19, 2011 | sysdate | Y |  |  |  |
| companyId | Company Id that registered project | int | 1 |  | Y |  |  | Y |
| teamId | Team the project was matched with | int | 1 |  |  |  |  | Y |
| InstructorId | Instructor ID | int | 1 |  | Y |  |  | Y |

**TeamMember**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Member Name** | **Description** | **Type** | **Sample Date** | **Default Value** | **Mandatory?** | **Unique?** | **PK** | **FK** |
| memberId | Team member ID | int | 1 | Auto-increase | Y | Y | Y |  |
| firstName | Member first name | varchar(15) | Anastasia |  |  |  |  |  |
| lastName | Member last name | varchar(15) | Semionova |  |  |  |  |  |
| email | Member e-mail | varchar(25) | asemionova1@learn.senecac.on.ca |  | Y |  |  |  |
| image | Member picture | blob | N/A |  |  |  |  |  |
| description | Member short story | varchar(250) | N/A |  |  |  |  |  |
| role | Is this member a team-leader? | boolean | false | false | Y |  |  |  |
| teamId | The member of what team | int | 1 |  | Y |  |  | Y |

**TeamProjectRanking**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Member Name** | **Description** | **Type** | **Sample Date** | **Default Value** | **Mandatory?** | **Unique?** | **PK** | **FK** |
| teamId | Team ID | int | 1 |  | Y | Y | Y | Y |
| projectId | Project Id | int | 1 |  | Y | Y | Y | Y |
| whoRanked | Who was doing ranking (team or company ) | char(1) | T or C |  | Y | Y | Y |  |
| ranking | RankinNumber | int | 2 |  | Y |  |  |  |

**ProjectFile**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Member Name** | **Description** | **Type** | **Sample Date** | **Default Value** | **Mandatory?** | **Unique?** | **PK** | **FK** |
| fileId | File ID | int | 1 | Auto-increase | Y | Y | Y |  |
| fileName | File name | varchar(16) | Constr.txt |  |  |  |  |  |
| fileDescription | File description | varchar(50) | File with constraints |  |  |  |  |  |
| ProjectId | What project it belongs to | int | 1 |  | Y |  |  | Y |

**Company**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Member Name** | **Description** | **Type** | **Sample Date** | **Default Value** | **Mandatory?** | **Unique?** | **PK** | **FK** |
| companyId | company ID | int | 1 | Auto-increase | Y | Y | Y |  |
| companyName | Company Name | varchar(25) | Seneca College |  | Y |  |  |  |
| companyPhone | Company phone number | varchar(12) | 416-156-1122 |  |  |  |  |  |
| status | Was company approved | boolean | false | N | Y |  |  |  |
| repID | Company representative ID | int | 1 |  |  |  |  | Y |

**Comment**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Member Name** | **Description** | **Type** | **Sample Date** | **Default Value** | **Mandatory?** | **Unique?** | **PK** | **FK** |
| commentId | Comment ID | int | 1 | Auto-increase | Y | Y | Y |  |
| status | Was comment approved? | boolean | false | false | Y |  |  |  |
| comment | Comment story | Varchar(250) | N/A |  | Y |  |  |  |
| projectId | What project the comment was left for | int | 1 |  | Y |  |  | Y |

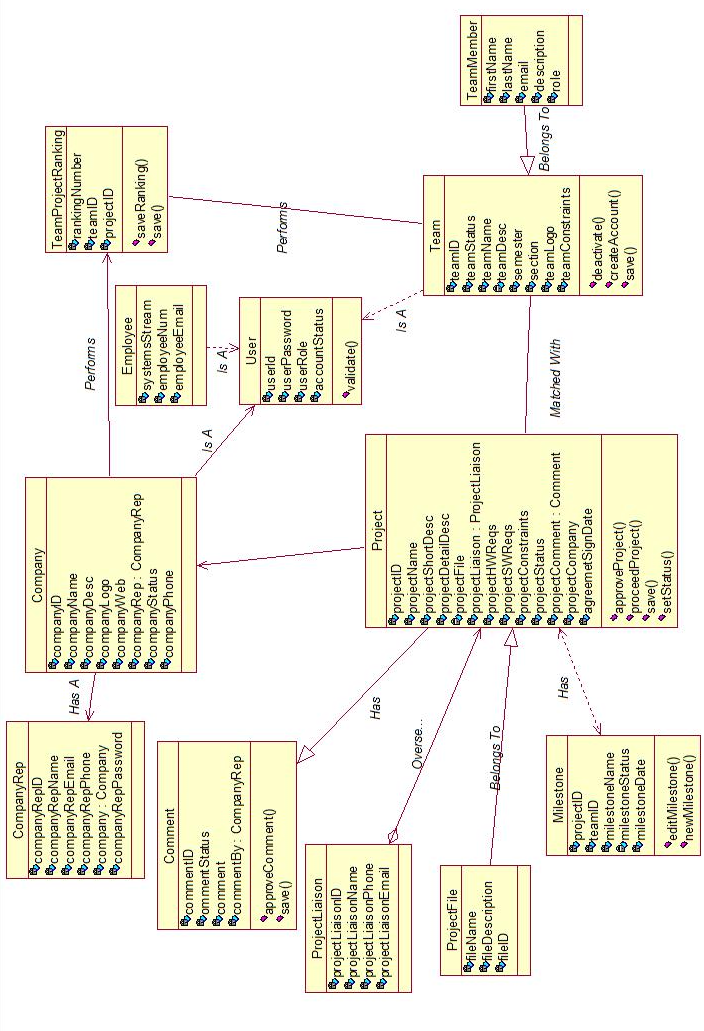
**Milestone**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Member Name** | **Description** | **Type** | **Sample Date** | **Default Value** | **Mandatory?** | **Unique?** | **PK** | **FK** |
| milestoneId | Milestone ID | int | 1 | Auto-increase | Y | Y | Y |  |
| milestone | Milestone description | varchar(125) | N/A |  | Y |  |  |  |
| projectId | What project this milestone belongs to | int | 1 |  | Y |  |  | Y |
| status | Milestone status: not started, in progress, complete | char(2) | NS, IP or CO | NS | Y |  |  |  |
| dueDate | When should be completed | datetime | July 19, 2011 | sysdate() | Y |  |  |  |

**User**

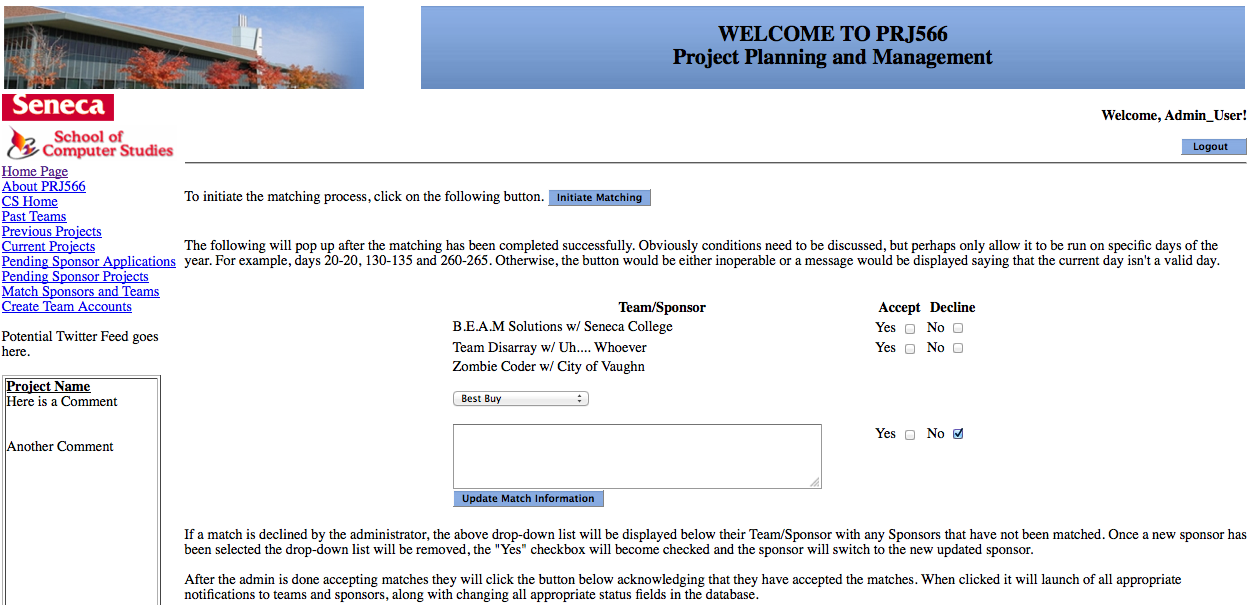
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Member Name** | **Description** | **Type** | **Sample Date** | **Default Value** | **Mandatory?** | **Unique?** | **PK** | **FK** |
| userId | System user ID | int | 1 | Auto-increase | Y | Y | Y |  |
| userFName | User first name | varchar(16) | Anastasia |  | Y |  |  |  |
| userLName | User last name | varchar(16) | Semionova |  | Y |  |  |  |
| userEmail | User e-mail | varchar(30) | asemionova1@learn.senecac.on.ca |  | Y |  |  |  |
| userIdentifier | Unique user identifier (stream+section+semester+year) | varchar(15) | PRJ566Sum2011 |  | Y |  |  |  |
| userRole | Is this user a supervisor, instructor, admin, team-lead, or company rep | char(2) | FA, IN, AD, TL or CR |  | Y |  |  |  |
| password | User password | varchar(40) | N/A | Sum2011 | Y |  |  |  |
| status | Is account active | boolean | true | true | Y |  |  |  |

5.4 Class Diagram

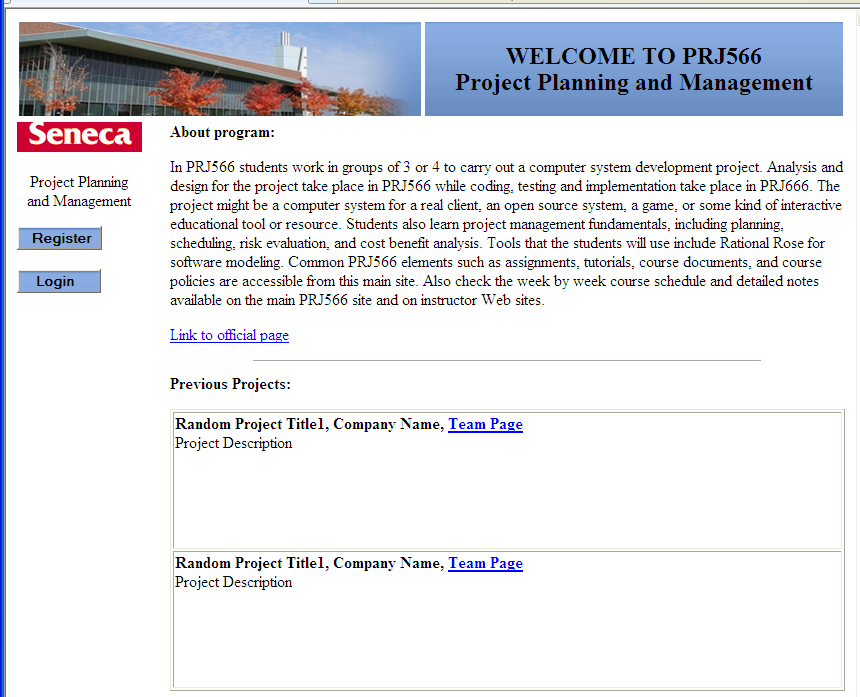


5.5 User Interface Prototypes

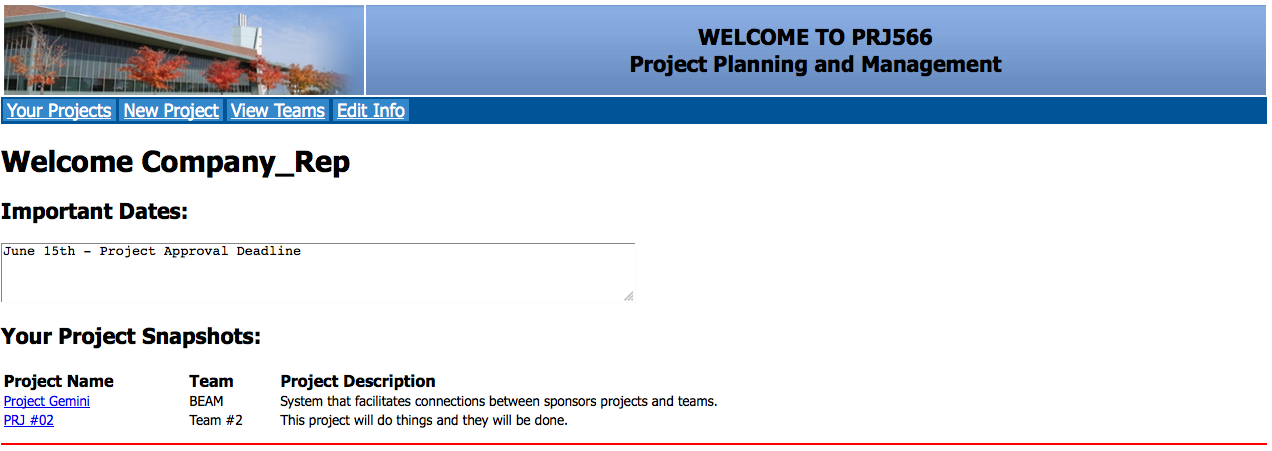
5.5.1 Admin Matching Page



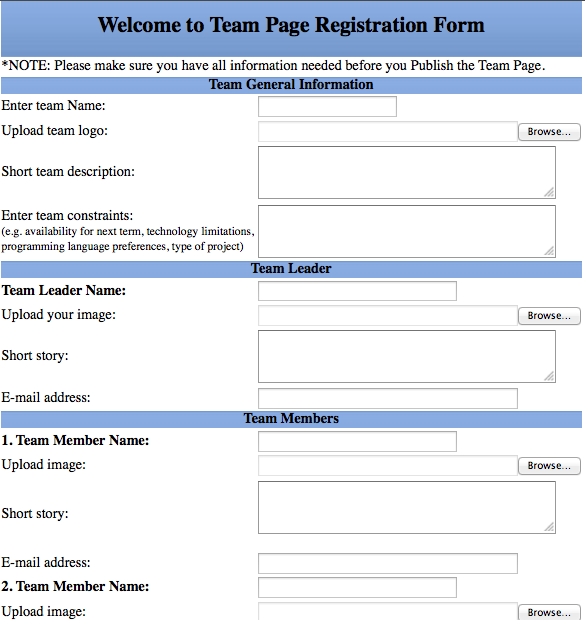
5.5.2 Home Page



5.5.3 Sponsor Home Page



5.5.4 Team Registration Page



5.5.5 Team Page



5.7 Test Cases

**Test Case: Proceed Projects to Appropriate Semester**

Step#

* + - 1. **Action:** Select “Proceed Projects” from the menu bar on the instructor’s home page

**Desired Result:** User is redirected to the “Proceed Projects” page

* + - 1. **Action:** Select the semester for each project from the list.

**Desired Result:** The appropriate semester is selected to all projects in the list

* + - 1. **Action:** Submit and confirm the action

**Desired Result:** All necessary projects are proceeded to the appropriate semester, and they are not displayed on the “View All Projects” page.

**Test Case: Login**

Step#

**Action:** Select “Login” on the Site Home Page

**Desired Result:** User is redirected to “Login” page

**Action:** Enter credentials.

**Desired Result:** User is redirected to the appropriate home page (site admin/ supervisor/instructor/company rep/team)

**Test Case: Approve Project**

Step#

1. **Action:** Select “View proposed projects” from the menu bar on the instructor’s home page

**Desired Result:** User is redirected to “View Proposed Projects Page” and the list of all proposed projects is displayed

1. **Action:** Select project to approve

**Desired Result:** The company and project information is displayed

3. **Action:** Submit and confirm the action

**Desired Result:** Project status is changed to approved, and if it is an unregistered company project, the user is redirected to the “Create Company Account Page”, or if it is a registered company, the user is redirected to the Instructor’s home page

**Test Case: Review Comment**

Step#

1. **Action:** Select “View Pending Comments” from the menu bar on the Site Administrator Home page

**Desired Result:** The user is redirected to the “View Pending Comments” page and the list of all pending comments is displayed

2. **Action:** Select the comments to be approved

**Desired Result:** All comments that are appropriate are selected

3. **Action:** Submit and confirm the action

**Desired Result:** The selected comments status is changed to approved and they should not appear in the pending comments list

**Test Case: Deactivate Accounts**

Step#

1. **Action:** Select ”Deactivate Team Accounts” from the menu bar on the Instructor’s Home Page

**Desired Result:** The user is redirected to “Deactivate Team Accounts” page where the list of all available teams for the current semester is displayed

2. **Action:** Select all teams that need to be deactivated

**Desired Result:** All teams that need to be deactivated are selected

3. **Action:** Submit and confirm the action

**Desired Result:** All selected teams statuses are changed to inactive and they should not appear on the “View All Teams” page.

**Test Case: Rank Projects**

Step#

1. **Action:** Select “Rank Projects” from the menu bar on the Team’s page

**Desired Result:** The user is redirected “Rank Projects” page where ranking for all available projects can be entered. If it is not the first time ranking, the previous ranking is also displayed

2. **Action:** Rank the appropriate amount of projects

**Desired Result:** Each entry is validated and the appropriate amount of projects is ranked

3. **Action:** Submit and confirm the action

**Desired Result:** The ranking is saved and user is redirected to the Team Home page

**Test Case: Rank Teams**

Step#

1. **Action:** Select “Rank Teams” from the menu bar on the Company Home page

**Desired Result:** The user is redirected to the “Rank Teams” page where the list of all available teams is displayed. If it is not the first time ranking, the previous ranking is also displayed

2. **Action:** Rank all teams.

**Desired Result:** Each entry is validated and only if all teams are ranked, the result can be submitted

3. **Action:** Submit and confirm the action

**Desired Result:** The ranking is saved and the user is redirected to the Company Home page

**Test Case: Match Teams/Projects**

Step#

1. **Action:** Select “Match Teams/Projects” from the menu bar of Instructor’s Home page.

**Desired Result:** The user is redirected to the “Match Teams/Projects” page

2. **Action:** Click on “Initiate Matching” button

**Desired Result:** Matching is dome for all projects and teams that have done ranking and results are displayed on the screen

3. **Action:** Select “Edit” on a match.

**Desired Result:** The matching is displayed in the editable form

4. **Action:** Adjust the matching

**Desired Result:** Matching is validated and the new matching results are displayed

5. **Action:** Click on “Finalize Matching”.

**Desired Result:** The matching results are saved and e-mails are send out to the instructor, all involved teams and companies

**Test Case: Change Project Status to Past**

Step#

1. **Action:** Select “Move Completed Projects to Archive” from the menu bar on the Supervisor Home page.

**Desired Result:** The list of all available projects for the current semester is displayed

2. **Action:** Select projects from list provided.

**Desired Result:** All projects needs to be archived are selected

3. **Action:** Click on “Submit Status Change”

**Desired Result:** The status of all selected projects is changed to past and all these projects should appear on the “View All Past Projects” page from the Site Home page

**Test Case: Create Team Account**

Step#

1. **Action:** Select “Create Team Account” from the menu bar on the Instructor’s Home page.

**Desired Result:** The user is redirected to “Create Team Account” page

2. **Action:** Enter class section, current semester and stream (PRJ566)

**Desired Result**: The entered information is validated

3. **Action:** Click on “Create New Account”

**Desired Result:** The new account credentials are displayed

**Test Case: View Project Status**

Step#

1. **Action:** Select a project form the list of all projects on the Company Home page.

**Desired Result:** The Project page is displayed with all applicable milestones and their statuses

**Test Case: Manage Project Files**

Step#

1. **Action:** Select a project from the all available projects list on the Company Home Page

**Desired Result:** The Project page is displayed

2. **Action:** Select “Upload File” from the menu bar

**Desired Result:** The upload file form appears

3. **Action:** Enter file description, file name and browse to the file to be uploaded, and submit

**Desired Result:** The user is redirected to the Project page and a link is displayed in the “Uploaded Files” section

**Test Case: Manage Site Accounts**

Step#

1. **Action:** Select “Change Account Status” from the menu bar on the Site Administrator Home page.

**Desired Result:** All Supervisor and Instructor accounts are displayed with their status in editable field

2. **Action:** Select the new status from the options provided and click “Change” button

**Desired Result:** The status is changed to the selected one

**Test Case: Change Project Status to Available**

Step#

1. **Action:** Select "View Approved Projects" from the menu bar on the Instructor’s Home page

**Desired Result:** The user is redirected to the All Approved Projects page where the list of all approved projects is displayed with the option to change the status

2. **Action:** Select a project

**Desired Result:** The Project information is displayed

3. **Action:** Change project status and submit

**Desired Result:** The project status is set to available and this project should appear on “View All Projects” page

**Test Case: Manage Milestones**

Step#

1. **Action:** Select "Manage Milestones" from the Team’s Home page

**Desired Result:** The user is redirected to the “Manage Milestones” page

2. **Action:** Select "Add Milestone"

**Desired Result:** The form to add new milestone appears

3. **Action:** Enter name, date, and description and submit

**Desired Result:** The new milestone is added to the project with the status N/A

**Test Case: Manage Team**

Step#

1. **Action:** Select "Edit" from the Team’s Home page

**Desired Result:** The team page is changed to the editable form

2. **Action:** Change all team information and submit

**Desired Result:** The new information is saved and the Team page is updated

**Test Case: Publish Team**

Step#

1. **Action:** Select "Log-In" from the Site Home Page

**Desired Result:** The user is redirected to the Login page

2. **Action:** Enter credentials

**Desired Result:** Credentials are validated and if valid, the change password form appears

3. **Action:** Change password and submit

**Desired Result: The Publish Team Page form appears**

4. **Action:** Enter in all team information and submit

**Desired Result:** The user is redirected to a Team Page where all entered information is displayed

**Test Case: View Project**

Step#

1. **Action:** Select a project form the list of all available projects on the Company Home page

**Desired Result:** The user is redirected to the Project page

**Test Case: View Calendar**

Step#

1. **Action:** Select “View Calendar” from the menu bar on the Company Home page

**Desired Result:** The user is redirected to the Calendar page

**Test Case: View All Teams Page**

Step#

1. **Action:** Select “View Active Teams” form the menu bar on the Company Home page

**Desired Result:** The user is redirected to All Teams page where a list of all available teams is displayed

**Test Case: Manage Company Page**

Step#

1. **Action:** Select “Edit Company Info” from the menu bar on the Company Home page

**Desired Result:** The user is redirected to the Edit Company Info page where all fields are displayed in editable mode  
2. **Action:** Enter new data (e.g. for Company Name, Logo, Web, Phone, and Description)

**Desired Result:** The entry is validated

3. **Action:** Submit the form

**Desired Result:** The user is redirected to the Company Home page and the company information is updated.

**Test Case: Propose New Project if registered**

Step#

1. **Action:** Select “New Project” from the menu bar on the Company Home page

**Desired Result:** The Legal Agreement form is displayed

2. **Action:** Accept the Legal Agreement

**Desired Result:** The propose a project form appears

3. **Action:** Enter data for Project name, description, constraints, files, project liaison contact information and submit

**Desired Result:** The entry is validated and saved in database as pending for approval. It should appear in “View All Proposed Projects” list from the Inspector Home Page

**Test Case: Propose New Project during Registration**

Step#

1. **Action:** Select “Register” from the menu bar on the Site Home page

**Desired Result:** The Legal Agreement form is displayed

2. **Action:** Accept the Legal Agreement

**Desired Result:** The propose a project form appears that also includes fields for the company information

3. **Action:** Enter all company information and project information (project name, description, constraints, files, project liaison contact information) and submit

**Desired Result:** The entry is validated and saved in database as pending for approval. It should appear in “View All Proposed Projects” list from the Inspector Home page

**Test Case: Leave Comment**

Step#

1. **Action:** Select “Leave the Comment” option beside the project from the all projects list

**Desired Result:** The “Leave the comment” form appears

2. **Action:** Enter the comment and submit

**Desired Result:** Comment is saved and is displayed under the project and Leave the comment option is no longer available. The submitted comment should appear in the all pending comments list from the Site Administrator Home page

**Test Case: Create Company Account**

Step#

1. **Action:** View company information and select the “Create Company Account” option

**Desired Result: The new company credentials are displayed**

2. **Action:** Acknowledge the new credentials

**Desired Result:** The user is redirected to the Instructor Home page and new credentials are e-mailed to the company and instructor. Company representative should be able to login with new credentials

# 6. Project Breakdown

This project will be implemented in one semester (4 months) by BEAM Solutions team in 2012 Winter semester.

#### Plan:

* Week 1 – 2: Create all necessary components for database (all team members)
* Week 3 – 5: Create Home Page (all team), and implement all functionalities for Team Page (Anastasia Semionova and Matthew Schranz) and Company Page (Bryan Cohen, Edouard Davlatian)
* Week 6– 9: Implement all functionalities for Instructor Page Page (Bryan Cohen, Edouard Davlatian), and Supervisor and Site Administrator pages (Anastasia Semionova and Matthew Schranz)
* Week 10 – 13: Testing and Performance Tuning (each team member test a page that was created by another team member)
* Week 14: Closure