****

***Team 14:*** *Jeremy Coney, Trey Kline, Jonathan Du, Daniel Ostrowski, Alex van de Sandt*

**Sprint Overview**

During this sprint we will be focusing on more in depth features, as well as polishing existing features. We finished the basic framework and outline in the last sprint that will allow us to go more in depth with our software in this sprint. We will be adding more interaction between different users like implementing the abilities to add other people to projects and assigning people tasks. We will also be starting the notification system, which will require setting up emails as notifications. Along with the notification system, we will need to be able to update all users’ displays when a peer updates information related to the user.

**Scrum Master:** Trey Kline

**Meeting Plan:** Tuesdays 6:30pm-7:30pm and Thursdays 3:30pm-4:30pm

**Risks and Challenges:** For this sprint, the challenge will be the more complex, interrelated interactions we will have to manage in our software. The first sprint consisted mainly of accepting information, storing it, and then displaying it back. Now, we will need to make updates to views across all concurrent users, and validate how a user will attempt to interact with another user. For example, when adding another user to a project the person adding other users must be the project’s manager. This will add much more complexity to our Javascript, which was already a challenge for us in the first sprint.

**Current Sprint Detail**

**User Story #7**

As a user, I want to create a project.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI to add other users to a project | 2.5 | Jeremy |
| 2 | Set up the connection between UI and the controller | 3 | Jeremy |
| 3 | Write and execute unit tests for controller and backend | 1.5 | Jeremy |

**Acceptance Criteria**

* Given that the user is logged in, when the user attempts to create a new project, then the user will be taken to the project creation page.
* Given that the UI is implemented correctly, when the user attempts to enter invalid information, then the UI will display the proper error messages.
* Given that all the information provided is valid, when the user is attempting to create a new project, then the application will create a new project.
* Given the UI has been correctly created, when the user requests to create a new project, then the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the UI has sent a project creation request, then the controller will send the correct queries to the database.

**User Story #9**

As a user, I want to create tasks.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for creating tasks for a certain project | 2.5 | Jeremy |
| 2 | Set up the connection between UI and the controller | 4 | Jeremy |
| 3 | Write and execute unit tests for controller and backend | 1.5 | Jeremy |

**Acceptance Criteria**

* Given that the user is logged in, when the user attempts to create a new task, the user will be taken to a task creation page
* Given that the user is creating a task, when a user clicks the confirm button, the UI will notify the user if any required fields are missing.
* Given the user has filled in all required fields, when the user clicks the confirm button, the application will create a new task.
* Given the UI has been correctly created, when the user requests to create a new task, then the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the UI has sent a task creation request, then the controller will send the correct queries to the database.

**User Story #10**

As a user, I want to modify tasks.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for modifying tasks | 3 | Jeremy |
| 2 | Set up the connection between UI and the controller. | 4 | Jeremy |
| 3 | Write and execute unit tests for controller and backend | 1.5 | Jeremy |

**Acceptance Criteria**

* Given that the user is logged in, when the user attempts to modify an existing task, the user will be taken to the task editing page.
* Given the UI has been correctly created, when the user requests to modify an existing task, then the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the UI has sent a task modification request, then the controller will send the correct queries to the database.

**User Story #11**

As a user, I want to delete tasks.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for deleting tasks | 3 | Jeremy |
| 2 | Set up the connection between UI and the controller. | 4 | Jeremy |
| 3 | Write and execute unit tests for controller and backend | 1.5 | Jeremy |

**Acceptance Criteria**

* Given the user has appropriate permissions for the task, when the user clicks delete, the user will be shown a confirmation dialog.
* Given the application has been set up correctly, when the user confirms the deletion, the application will delete the task.
* Given the UI has been correctly created, when the user requests to delete a new task, then the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the UI has sent a task deletion request, then the controller will send the correct queries to the database.

**User Story #12**

As a user, I want to submit daily progress reports.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for daily progress reports | 4 | Alex |
| 2 | Create unit tests for viewing progress reports | 2 | Alex |

**Acceptance Criteria**

* Given that the user is logged in, when the user attempts to submit a daily progress report, the manager will be taken to the ‘add report’ page.
* Given that the user is adding a report, when the user clicks the confirm button, the UI will notify the user if any required fields are missing.
* Given the user has filled in all required fields, when the user clicks the confirm button, the application will add a daily progress report.
* Given the UI has been correctly created, when the user requests to submit a report, the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the UI has sent a submit report request, then the controller will send the correct queries to the database.

**User Story #13**

As a user, I want to be able to view other teammates daily progress reports.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for viewing progress reports | 4 | Alex |
| 2 | Create unit tests for viewing progress reports | 2 | Alex |

**Acceptance Criteria**

* Given that the user is assigned to a project and is logged in and given that the UI has been correctly created, when the user clicks a report summary on a project’s dashboard, the UI will send the appropriate request to the controller.
* Given that the UI has been written correctly, when the UI has sent a request for a report’s full body, then the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the controller receives a request for the full body of a report, the controller will make the appropriate query to the database.
* Given that the controller and database query have been written correctly, the database will yield the full report body, which the controller will forward to the UI.
* Given that the UI is correctly written, the UI will display the full report body in a page by itself.

**User Story #14**

As a user, I want to assign tasks to myself.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI to assign tasks | 3.5 | Jon |
| 2 | Set up the connection between UI and the controller. | 4 | Jon |
| 3 | Create and execute unit tests for the frontend and the backend | 2.5 | Jon |

**Acceptance Criteria**

* Given that the task exists, the user is assigned to the project, and the user is logged in, when the user clicks the volunteer button, the application will assign the user to that task.
* Given the UI has been correctly created, when the user requests to assign an existing task to themself, then the UI will invoke the proper controller.
* Given that the controller is implemented correctly, when the controller receives the self-assignment request from the UI, the controller will delegate the request to the appropriate database query.
* Given that the controller and query are written correctly, the database will be updated to reflect the assignment, and the controller will receive indication that the query succeeded or failed
* Given that the controller is written correctly, the controller will receive the indication of success or failure from the query and notify the UI of the result.
* Given that the UI is implemented correctly, the UI will signal to the user whether the self-assignment succeeded or failed.

**User Story #15**

As a user, I want to view tasks I’ve created, as well as tasks assigned to me.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for viewing tasks | 3.5 | Jon |
| 2 | Set up the connection between UI and the controller. | 4 | Jon |
| 3 | Create and execute unit tests for the frontend and the backend | 2.5 | Jon |

**Acceptance Criteria**

* Given the UI has been correctly created, and that the user is logged in, when the user requests to view their tasks, the UI will invoke the proper controller.
* Given that the controller is implemented correctly, when the controller receives a request from the UI for all of the user’s tasks, the controller will delegate the request to the appropriate database query.
* Given that the query is written correctly, the query will yield a list of all tasks that are assigned to the user along with all tasks that the user has created.
* Given that the controller is implemented properly, the controller will forward the list of results from the query or an indication of failure to the UI.
* Given that the UI is written correctly, it will then display all of the user’s tasks or notify the user that the request failed, as appropriate.

**User Story #17**

As a user, I would like to make a public comment associated with a task I am working on

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for adding comments to a task | 3.5 | Jon |
| 2 | Set up the connection between UI and the controller. | 4 | Jon |
| 3 | Set up the controller and backend API | 2 | Daniel |
| 4 | Create unit tests. | 2.5 | Jon |

**Acceptance Criteria**

* Given the UI has been correctly created, and that the user is logged in, when the user requests to make a public comment, the UI will invoke the proper controller.
* Given that the controller is implemented correctly, when the controller receives a request from the UI for all of the public comments associated with a task, the controller will delegate the request to the appropriate database query.
* Given that the query is written correctly, the query will yield a list of all comments that correspond to the user.
* Given that the controller is implemented properly, the controller will forward the list of results from the query or an indication of failure to the UI.
* Given that the UI is written correctly, it will then display all of the user’s comments or notify the user that the request failed, as appropriate.

**User Story #18**

As a user, I would like to see all comments on a task.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for viewing comments on a task | 2.5 | Trey |
| 2 | Set up the connection between UI and the controller | 2.5 | Trey |
| 3 | Set up the controller and backend API | 2 | Daniel |
| 4 | Write and execute unit tests for controller and backend | 2 | Trey |

**Acceptance Criteria**

* Given that the UI is implemented correctly, when the user selects a task, the UI should invoke the proper controller
* Given the controller is implemented correctly, when the controller is invoked, it should request all comments associated with a task.
* Given that the query is written correctly, the query will yield a list of all comments that correspond to the task.
* Given that the UI is written correctly, it will then display all of the comments for the task or notify the user that the request failed, as appropriate.

**User Story #19**

As a user, I would like to be notified when a task assigned to me with a deadline is almost due.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 2 | Add section of backend to detect when deadlines are near and dispatch emails accordingly | 5 | Daniel |

**Acceptance Criteria**

* Given that the controller is implemented correctly, when it is a set time before the task is due, all users assigned to the task shall receive an email notification.
* Given that the notification controller and backend has been written correctly, when the task has already been completed, then the notification will be suppressed.
* Given that the UI has been implemented correctly, when the user gets a notification email, then the email will contain a list of the tasks assigned, the tasks description, and the deadline.
* Given that the model has been correctly written, when the user gets a notification email, then the email will contain a link to the tasks page.

**User Story #20**

As a user I would like to have the ability to modify and save settings.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for modifying and saving setting s | 2.5 | Trey |
| 2 | Set up the connection between UI and the controller | 2.5 | Trey |
| 3 | Set up controller and backend API | 2 | Daniel |
| 4 | Write and execute unit tests for controller and backend | 2 | Trey |

**Acceptance Criteria**

* Given that the user is logged in, when the user attempts to modify settings, the user will be taken to a edit settings page.
* Given that the user is editing their settings, when a user clicks the confirm button, the UI will notify the user if any required fields are missing.
* Given the user has filled in all required fields, when the user clicks the confirm button, the application will save the changed settings.
* Given the UI has been correctly created, when the user requests to edit their settings, the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the UI has sent a modify settings request, then the controller will send the correct queries to the database.

**User Story #23**

As a project manager, I want to invite others to my project.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for inviting others to my project | 2.5 | Trey |
| 2 | Set up the connection between UI and the controller | 2.5 | Trey |
| 3 | Set up controller and backend API | 3 | Daniel |
| 4 | Write and execute unit tests for controller and backend | 2 | Trey |

**Acceptance Criteria**

* Given that the manager is logged in, when the manager attempts to invite others to a project, the manager will be taken to a project invitation page.
* Given that the manager is inviting others, when the manager clicks the confirm button, the UI will notify the manager if any required fields are missing.
* Given the manager has filled in all required fields, when the manager clicks the confirm button, the application will save the invitations.
* Given the UI has been correctly created, when the manager invites others to a project, the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the UI has sent a project invitation, then the controller will send the correct queries to the database.

**User Story #24**

As a project manager, I want to assign tasks to my team members.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for assigning tasks to my team members | 2.5 | Trey |
| 2 | Set up the connection between UI and the controller | 2.5 | Trey |
| 3 | Adjust controller and backend API | 2 | Daniel |
| 4 | Write and execute unit tests for controller and backend | 2 | Trey |

**Acceptance Criteria**

* Given the UI is implemented correctly, when the manager edits a task, he should see a list of users currently on the project.
* Given the UI is implemented correctly, when the manager selects one or more users, the UI should send a request to the controller.
* Given the controller is implemented correctly, when it receives a request, it should update the task and assigned users in the database to reflect the assignment.
* Given the controller is properly written, if the user making the request is not the manager of the project, an error message will be returned instead, as a security measure.

**User Story #25**

As a project manager, I want to be able to view the team’s daily progress reports.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 3 | Adjust controller and backend API | 2 | Daniel |

**Acceptance Criteria**

* Given the UI is correctly implemented, project managers will have a button per project they manage linking to a reports page.
* Given the UI is correctly implemented, the reports page will display all reports for a project, grouped by user.
* Given the controller is correctly implemented, the controller will return the list of reports already in the correctly order.
* Given the controller is properly written, the controller will send an error message instead if the user is not the project manager for the requested project as a security measure.

**User Story #26**

As a project manager, I would like to be notified when a task has been finished

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 2 | Set up controller and backend API | 2 | Daniel |

**Acceptance Criteria**

* Given email notifications are implemented and configured correctly, when a user marks a task as completed, the project manager should receive an email notification.
* Given the controller is implemented properly, notifications will be suppressed based on an application-level setting.
* Given the controller is implemented correctly, the email will be sent directly to the project manager, with the tasks assigned users carbon-copied.
* Given the controller is written properly, the email will contain the description of the task and a link to the task’s page.

**User Story #27**

As a project manager, I would like to create tasks for users.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for creating tasks for users | 2.5 | Trey |
| 2 | Set up the connection between UI and the controller | 2.5 | Trey |
| 3 | Adjust controller and backend API | 2 | Daniel |
| 4 | Create unit tests. | 2 | Trey |

**Acceptance Criteria**

* Given that the manager is logged in, when the manager attempts to create tasks for users, the manager will be taken to a create tasks page.
* Given that the manager is creating tasks, when the manager clicks the confirm button, the UI will notify the manager if any required fields are missing.
* Given the manager has filled in all required fields, when the manager clicks the confirm button, the application will create the task.
* Given the UI has been correctly created, when the manager requests to create a task, the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the UI has sent a create task request, then the controller will record the new task and update affected records.

**User Story #28**

As a project manager, I would like to add skills required for tasks.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 3 | Add controller and backend API | 3 | Daniel |

**Acceptance Criteria**

* Given that the manager is logged in, when the manager attempts to add skills required for tasks, the manager will be taken to an add skills page.
* Given that the manager is adding skills, when the manager clicks the confirm button, the UI will notify the manager if any required fields are missing.
* Given the manager has filled in all required fields, when the manager clicks the confirm button, the application will add skills required for the task.
* Given the UI has been correctly created, when the manager requests to add a skill, the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the UI has sent an add skills request, then the controller will update the task in the database with the newly-associated skills.
* Given the controller is properly written, an error message will be sent if the user adding skills to the task is not the task creator or the project manager, as a security measure.

**User Story #30**

As a project manager, I would like to set deadlines for tasks.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for setting deadlines on tasks | 4 | Alex |
| 2 | Set up the connection between UI and the controller | 4 | Alex |
| 3 | Add controller and backend API | 2 | Daniel |
| 4 | Create Unit Tests | 4 | Alex |

**Acceptance Criteria**

* Given that the manager is logged in, when the manager attempts to set deadlines required for tasks, the manager will be taken to a deadlines page.
* Given that the manager is setting deadlines, when the manager clicks the confirm button, the UI will notify the manager if any required fields are missing.
* Given that the manager has filled in all required fields, when the manager clicks the confirm button, the application will set deadlines required for the task.
* Given the UI has been correctly created, when the manager requests to set a deadline, the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the UI has sent a set deadline request, then the controller will send the correct queries to the database.
* Given that the controller has been written correctly, the controller will send a failure message to the UI if an attempt is made to set a deadline to a past date.

**User Story #32**

As a project manager, I would like to be able to push a custom one-time notification out to users assigned to a particular task

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Time | Owner |
| 1 | Set up the UI for sending a notification to users | 4 | Alex |
| 2 | Set up the connection between UI and the controller | 4 | Alex |
| 3 | Set up the email notifications to be sent to users | 3 | Daniel |

**Acceptance Criteria**

* Given that the manager is logged in, when the manager attempts to push a notification to users for a task, the manager will be taken to a notification creation page.
* Given that the manager is adding a notification, when the manager clicks the confirm button, the UI will notify the manager if any required fields are missing.
* Given the manager has filled in all required fields, when the manager clicks the confirm button, the application will push a notification required for the task.
* Given the UI has been correctly created, when the manager requests to push a notification, the UI will invoke the proper controller.
* Given that the controller has been written correctly, when the UI has sent an add skills request, then the controller will send the correct queries to the database.
* Given that the controller has been written correctly, when the UI makes the request to send the notification, the backend will send emails to the relevant users.

**Remaining Backlog (33/52 Stories Complete)**

1. ~~As a user, I want to create an account~~
2. ~~As a user, I want to delete my account.~~
3. ~~As a user, I want to log in.~~
4. ~~As a user, I want to be able to view my profile information.~~
5. ~~As a user, I want to be able to edit my profile information.~~
6. ~~As a user, I want to specify my skills and experiences.~~
7. ~~As a user, I want to create a project.~~
8. ~~As a user, I want to join someone else’s project.~~
9. ~~As a user, I want to create tasks.~~
10. ~~As a user, I want to modify tasks.~~
11. ~~As a user, I want to delete tasks.~~
12. ~~As a user, I want to submit daily progress reports.~~
13. ~~As a user, I want to be able to view other teammates daily progress reports.~~
14. ~~As a user, I want to assign tasks to myself.~~
15. ~~As a user, I want to view tasks I’ve created, as well as tasks assigned to me.~~
16. As a user, I would like to notify the project manager when a task has been completed
17. ~~As a user, I would like to make a public comment associated with a task I am working on~~
18. ~~As a user, I would like to see all comments on a task~~
19. ~~As a user, I would like to be notified when a task assigned to me with a deadline is almost due.~~
20. As a user I would like to have the ability to modify and save settings.
21. ~~If time allows, as a user, I want to be able to recover my username by email if forgotten.~~
22. ~~If time allows, as a user, I would like to be able to reset my password if forgotten.~~
23. ~~As a project manager, I want to invite others to my project.~~
24. ~~As a project manager, I want to assign tasks to my team members.~~
25. ~~As a project manager, I want to be able to view the team’s daily progress reports.~~
26. ~~As a project manager, I would like to be notified when a task has been finished~~
27. ~~As a project manager, I would like to create tasks for users~~
28. ~~As a project manager, I would like to add skills required for tasks.~~
29. As a project manager, I would like to edit the skills required for tasks.
30. ~~As a project manager, I would like to set deadlines for tasks.~~
31. As a project manager, I would like to be notified when a task is overdue.
32. ~~As a project manager, I would like to be able to push a custom one-time notification out to users assigned to a particular task~~
33. As a project manager, I would like to be able to push a custom one-time notification out to users assigned to a particular project
34. If time allows, as a project manager, I would like to be able to track the time that each employee works on the project and its parts.
35. If time allows, as a project manager, I would like to request progress updates from team members.
36. If time allows, as a system administrator, I would like to be able to run a script to install and run the server-side software components
37. If time allows, as a system administrator, I would like to be able to run a script to create user accounts in a batch
38. If time allows, as a system administrator, I would like to have logs detailing server-side software component usage
39. If time allows, as a software developer, I would like an API, so I could send additional custom notifications programmatically.
40. As developers, we want the user interface to be simplistic and display all necessary information.
41. As developers, we want the application to work across all web browsers.
42. As developers, we want the data retrieval queries to take less than 300ms.
43. ~~As developers, we want to ensure information is protected and secured.~~
44. As developers, we want to be able to handle a load of at least 15 concurrent users.
45. As developers, we want data changes to become viewable by other users within 10 seconds of being made.
46. ~~As developers, we want to see the total number of accounts present.~~
47. ~~As developers, we want to see the number of total skills added to a project.~~
48. ~~As developers, we want the user to have access to a settings mode.~~
49. If time allows, as developers, we want the user interface to be visually appealing
50. If time allows, as a user, I would like to have access to a dark mode for appearance.
51. If time allows, as a user, I would like to be able to change information colors as desired.
52. If time allows, as a sure, I would like to have customizable notification sounds.