

CS 30700 Group 10 (Synchronous): Sprint Retrospective 1

Group Members: Ryan Borzello, Kevin Wu, Saksham Jain, Shreyas Iyengar

1. What went well?

As a team that was very unsure about starting out with new technologies and new people, we felt that overall this sprint was a big success. We were able to pick up the technologies quickly and use them to help build a solid foundation for our project. We got all our user stories done, we helped each other out when we were stuck on something, we played to each other's strengths and there was great communication about struggles, successes, deadlines, and task management. We worked together quite well as a team, and did really well as a team. We hope to carry that same energy, dedication and camaraderie to the next sprint.

User Story 1

As a team leader, I would like to be able to create a workspace.

Tasks	Time	Assigned to
Create UI that allows a user to create a workspace	5 Hrs (Total)	Ryan/Shreyas
Create an algorithm to generate a unique ID for the database	2 Hrs (Total)	Saksham/Kevin
Interface with the database to store the workspace information	3 Hrs	Kevin
Connect the database with the UI to allow users to see whether the workspace was created or not(NOTE: still need a display to notify.)	3 Hrs (Total)	Ryan/Shreyas
Test to ensure that a workspace can be created and its information is stored in the database	2 Hrs (Total)	Ryan/Shreyas

Completed:

We have the main landing page that allows the user to create a workspace. There is an algorithm in place that allows us to generate a unique id for the database. The workspace information is stored within the database, and we have connected the UI with the database to allow users to see whether the workspace was created or not.

User Story 2

As a team leader, I would like to set a workspace's unique ID.

Tasks	Time	Assigned to
Build a UI text field to allow users to enter a unique ID and a button to set the ID	1 Hr (Total)	Ryan/Shreyas
Create algorithm to verify if a given ID has collisions	3 Hrs (Total)	Saksham/Kevin
Connect backend to database to modify a unique ID in the database	3 Hrs (Total)	Saksham/Kevin
Connect backend to UI to show whether the ID change succeeded	2 Hrs (Total)	Saksham/Kevin
Debug and test ID verification algorithm with unit tests	2 Hrs (Total)	Saksham/Kevin

Completed:

We have a text field in place to allow the users to enter a unique id and a button to set the ID.

We also used the UUID v4 to avoid ID collisions as well as generating a new ID. We have the connected backend to database to modify unique ids in the database, as well as connected the backend to UI to show whether the ID change succeeded.

User Story 3

As a team leader, I would like to be able to password protect my workspace during creation so that unauthorized users cannot access it.

Tasks	Time	Assigned to
Option to create a new password	1 Hr	Saksham
Prompt for password input (if exists)	1 Hr	Saksham
Allow access for correct password	3 Hrs	Kevin
Store passwords in the workspace database table in a secure manner	4 Hrs	Kevin
Send error message for incorrect password and reprompt	1 Hr	Saksham
Option to remove password	1 Hr	Saksham
Manually test that workspaces cannot be accessed without the correct password	2 Hrs	Saksham

Completed:

When a user creates a workspace, there is an option that allows them to password-protect their workspace. The given password after hashing is stored in the database. To access the database, a user inputs the password when joining a workspace. If the password is correct, they are redirected to the workspace page. Otherwise, they are shown an error message. Once logged into a workspace, they can optionally change or remove the password at any time.

User Story 4

As a team leader, I would like to optionally let others view my workspace without a password but have editing privileges be password-protected so I can share something for others to see but not edit.

Tasks	Time	Assigned to
Create separate view and edit modes	4 Hrs	Shreyas
Prompt to enter password if necessary	1.5 Hrs	Shreyas
Manually test that workspaces can be viewed but not edited without the correct password	2.5 Hrs	Shreyas

Completed:

There were separate view and edit modes available for each workspace based on whether the user entered the password for the workspace or not. The password was prompted whenever necessary and we tested it by making sure when a user entered a view only enabled workspace without a password, they were not in edit mode

User Story 5

As a team leader, I would like to have my workspace automatically delete itself after 24 hours.

Tasks	Time	Assigned to
Create a timing mechanism for seeing how long each workspace has been up for	4 Hrs	Saksham
Email user 1 hour before the workspace purges	2.5 Hrs	Saksham
Make sure that the workspace is deleted automatically after 24 hours.	2.5 Hrs	Saksham
Manually check what happens after the time limit expires (Change from 24 hours if necessary)	2 Hrs	Saksham

Completed:

The mechanism worked and the user could see how long the workspace had been up for, the user got emailed 1 hour before their workspace deleted, and the workspace deleted after 24 hours, which we were able to test by setting the purge time to 2 minutes.

User Story 6

As a team member, I would like to be able to access a workspace through its URL.

Tasks	Time	Assigned to
Create method that auto-generates a url based on workspace name	4 Hrs	Saksham/Kevin
Check if workspace URL is already taken	2 Hrs	Saksham/Kevin
Send proper error message if no workspace exists	2 Hrs	Saksham/Kevin
Test that workspace is accessible through its URL	3 Hrs	Saksham/Kevin

Completed:

A workspace's URL is generated based on the UUID assigned to the workspace. This prevents the URL from changing if the nickname is changed, resulting in users failing to find the workspace that was shared with them. If no workspace exists, then an error page is displayed saying that the given workspace does not exist.

User Story 7

As a team member, I would like to be able to access a workspace by entering its unique ID.

Tasks	Time	Assigned to
Build UI that prompts for workspace ID	2 Hrs	Ryan
Redirect user to respective workspace when ID is inputted	3 Hrs	Ryan
Send error message if no workspace exists	1 Hr	Ryan
Test that workspace is accessible by entering its unique ID	2 Hrs	Ryan

Completed:

There is a simple UI to enter the workspace ID you need, that redirects to the workspace when the correct id is added. When no workspace with name exists the box highlights red and an error message is displayed.

User Story 8

As a team member, I would like to be able to send an email out to invite people.

Tasks	Time	Assigned to
Create interface for users to input other users email addresses	2 Hrs	Ryan
Create a mechanism to send automatic emails	2 Hrs	Ryan
Put workspace URL/ID (and password if needed) within email	1 Hr	Ryan
Allow for additional comments to be put in email message	1 Hr	Ryan
Test manually to check if emails go through to users	2 Hrs	Ryan

Completed:

There is an email icon in the corner of all workspaces where an email invite can be sent. There are 2 text boxes, one for email and one for additional messages. After clicking the send button, the email is either sent or an error message is displayed.

User Story 9

As a user, I would like to be able to see who else is online in my workspace.

Tasks	Time	Assigned to
Create a users list in the workspace in the backend	4 Hrs	Kevin
Create a visual layer to show active/inactive users	2 Hrs	Kevin
Create a timestamp that activates after the user goes inactive to show duration of inactivity	2 Hrs	Kevin
Debug and test user list manually	3 Hrs	Kevin

Completed:

When a user connects to a workspace, they automatically connect to a WebSocket server that sends "user joined" messages to all other users connected to the workspace. When a user joins a message is received, the UI updates the list of current users. Finally, when a user goes inactive/returns from inactivity, their new state is broadcast to all other users and their UIs are updated.

User Story 10

As a user, I would like to be able to minimize, rearrange, and resize an app without affecting what other users connected to the workspace see so that I can customize my workspace layout.

Tasks	Time	Assigned to
Create a placeholder app for testing	2 Hrs (Total)	Ryan/Shreyas
Allow users to minimize an app	3 Hrs (Total)	Ryan/Shreyas
Allow users to un-minimize an app	1 Hr (Total)	Ryan/Shreyas
Allow users to drag and move an app	4 Hrs (Total)	Ryan/Shreyas
Allow users to resize an app	3 Hrs (Total)	Ryan/Shreyas
Test to check whether app is resizing, rearranging and minimizing	3 Hrs (Total)	Ryan/Shreyas

Completed:

The placeholder app has been created for testing under the test a workspace button on the website. We can minimize and un-minimize the app with a button click, drag and move the app by selecting the app and dragging with the mouse pointer, as well as resizing the app to grow or shrink the window by dragging. We ran some tests to see if it would work, and it seems to be working fine as intended.

User Story 11

As a user, I would like to be able to delete an app so that I can discard its contents.

Tasks	Time	Assigned to
Enable closing of tabs(App)	2 Hrs (Total)	Ryan/Shreyas
Discard all data in the workspace when app windows are closed.	1 Hr (Total)	Ryan/Shreyas
Make sure that other windows automatically resize to fill empty app space and fit the full window.	2 Hr (Total)	Ryan/Shreyas
Make sure that when the user reopens an app, it opens a fresh iteration of the app with the contents discarded.	1.5 Hr (Total)	Ryan/Shreyas
Make sure that the user can close the app in multiple ways (window, right click, keyboard shortcuts etc.)	2 Hrs (Total)	Ryan/Shreyas
Manually create and delete apps for testing	2 Hr (Total)	Ryan/Shreyas

Completed:

There exists a delete button that when clicked the App box goes away. When recreated all the data is gone. When a new app is created it is fresh every time. In addition to clicking on the delete button, right clicking also works.

User Story 12

As a user, I would like to be able to submit bug fixes and user issues to the developer.

Tasks	Time	Assigned to
Creating a way to access a bug report.	1 Hr	Shreyas
Finding or creating a tool that will allow users to create and submit the bug report.	3 Hrs	Saksham
Store the bug report in an easily accessible database or workspace, where each bug report is stated clearly in a particular format.	3 Hrs	Shreyas
Convert the bug reports into tasks that can be assigned to a team member to investigate and eventually fix.	2 Hrs	Shreyas
Manually send bug reports to test	2 Hrs	Shreyas/Saksham

Completed:

We were able to create a method for accessing the bug reports, and the user was able to send bug reports easily from the website. The bug reports were stored for us in a sorted manner to review. These bugs were then ready to be assigned to one of the developers (us) on a shared dev board. We were able to test it by manually sending in bug reports as users.

2 What did not go well?

Overall:

For multiple user stories we did not have accurate estimations for how long each task would take. This caused us to fall behind and rush the final week to finish the user stories. Also, our acceptance criteria were not really focused. We wanted to test things that probably should have been left for subsequent sprints.

For user story 11, we changed our approach from the interface being windowed, to the apps being free moving. Because of this change we were unable to have apps resize when one was deleted to fill up the missing space.

This was the only user story with an issue.

3 How should you improve?

Now that we have knowledge of how to use technologies we are implementing, we can provide better estimates for the length of time user stories will take. This was a large issue in sprint one and we need to improve on our estimates of how difficult tasks are.

In continuation of the previous point, one issue we did not really factor in was the time spent researching new API's or technologies. For any user stories in the next sprint that require new technologies, we should include ample time for research within the time estimate for completion.

Another thing to improve upon would be putting more effort in towards the beginning of the sprint rather than pushing tasks to the end. When we were first starting we slowly started learning and implementing parts of the project. This caused us to have a lot of tasks left for the last week of the sprint. We should manage the entire duration of the sprint better and work consistently through it.