Test Plan

Date Submitted: 10/27/2021

Team Hobby:

Colin Creasman

Daniel Bribiesca

Team Lead: Jacob Delgado

Long Nguyen

Rifat Hasan

Document History

Date	Version	Changes
10/14/2021	0.1	Created Doc with Initial Outline
10/15/2021	0.2	Drafted Initial Test Scenario with Specific Test Case
10/15/2021	0.3	Removed Specific Test Cases; Added Test Scenarios
10/19/2021	0.4	Removed Redundant Testing; Added Timeline
10/23/2021	0.5	Added Specific Inputs for Test Cases
10/25/2021	0.6	Added Specific Execution Steps for Test Cases
10/27/2021	0.7	Submitted Final Draft

Table of Contents

1 - Introduction

1.1 - In Scope

The testing scope encompasses all of the system components that comprise all the various application features. Overall testing is split up between individual Test Scenarios of a granularity defined by each scenario's independently testable functionalities within their overall feature. The Test Scenarios will be used to verify and validate the all relevant business requirements for the application features:

- Account Creation
  - 3.1 - Account Registration
- Login
  - 3.2 - Login
  - 3.3 - Logout
- User Questionnaire
  - 3.4 - Questionnaire
- Search and Discovery
  - 3.5 - Search
  - 3.6 - Search Filter
  - 3.7 - Discovery Filter
- UI
  - 3.8 - UI Controls
- Notifications
  - 3.9 - Notifications
  - 3.10 - Notification Page
- Project Posts
  - 3.11 - Post Project
  - 3.12 - Upload File to Project
  - 3.13 - Remove File to Project
  - 3.14 - Save Project as Draft
  - 3.15 - Edit Project
  - 3.16 - Delete Project
  - 3.17 - Rate Project
- Project Comments
  - 3.18 - Post Comment
  - 3.19 - Rate Comment
  - 3.20 - Filter Comments
- User Recommendations
  - 3.21 - Project Recommendation
  - 3.22 - Tool Recommendation
- Project Collection

1.2 - Out of Scope

This document will not cover methods or features that have been left out of the business requirements document to concentrate base testing on in-scope functionalities. Non-functional requirements will be out of the scope along with other forms of testing such as unit, regression, penetration testing because of lack of time and client demands.

Example of non-functional functionalities that are out of the scope of this Test Plan include, but are not limited to:

- Client-side browser requirements (e.g. unsupported versions of Chrome)
- Client-side hardware requirements (e.g. unsupported CPU or RAM specs)
- Client-side OS requirements (e.g. unsupported Windows builds)

2 - Testing Strategy

2.1 - Methodology Overview

We will be using an **Agile** testing methodology

2.2 - Testing levels

A combination of integration and system (end-to-end) testing will be implemented in order to make the application run efficiently with proper error handling methods to validate that the features will run smoothly and optimize user experience. Unit testing will be used in general since each slice of the project will need to be validated through pass and fail scenarios. This combination of manual and automated testing will cover all the features specified in the business requirements document project scope. Estimated time will be allotted for each test case to develop a contingency plan.

- **Integration Level Testing:** Will be used for Test Scenarios that involve interaction between multiple system components in order to evaluate that the given Test Scenario is satisfying all its relevant business requirements
- **System (End-to-End) Level Testing:** Will be used when testing must be conducted on a on the complete and integrated system as a whole in order to evaluate that the given Test Scenario is satisfying its relevant business requirements
- **Unit Testing:** Will be used to test each individual functionality at the smallest class level. Unit testing for each functionality will be conducted during development of the feature so it is not explicitly shown in each Test Scenario description

3 - Test Scenarios

3.1 - Account Registration

Verification

When a user first arrives at the application homepage, the center of the page should display the UI component that allows them to either sign in or create a new account. If the user clicks the "New Account" button they should be directed to the Register page. This page shows empty text fields for Username and Password and a "Register" button underneath them.

Priority

High - Most of the application-specific features are only available to users with an account after logging in. Creating an account is a core component so many other features are dependent on it.

System (End-to-End) Level Testing

System-level testing is required because account creation will involve communication between multiple system components such as the public UI components where the user will input their credentials, the frontend data logic that will validate them, the backend business logic that will validate them, and the internal databases that will store them.

Pre-Conditions

- User is not signed into any account when registering for an account
- User does not register with an existing account

All of the following test cases must pass in order for the feature to be considered validated successfully

3.1.1 - Test Case 1: Valid Username for Registration using valid characters

Inputs

- Username: "johndoe123"
- Password: "Password!"

Test Execution Steps

- User goes to the Register page
- User enters "johndoe123" for username

- User enters "Password!" for password
- User clicks "Register"

Expected Outcome for Validation (all validations must occur)

- User is directed to the User Questionnaire Page
- System displays the "Account Created!" message at the top of the User Questionnaire Page

Estimated Execution Time - 1 Minute

3.1.2 - Test Case 2: Invalid Username for Registration using invalid characters (only 1 invalid character is needed for error to show)

Inputs

- Username: "+<>+"
- Password: "Password!"

Test Execution Steps

- User goes to the Register page
- User enters "+<>+" for username
- User enters "Password!" for password
- User clicks "Register"

Expected Outcome for Validation (all validations must occur)

- User is redirect back to the Register Page
- User is shown the error message "Error - Username cannot contain invalid characters"
- User is shown the "Try Again" button

Estimated Execution Time - 1 Minute

3.1.3 - Test Case 3: Invalid Password for Registration using password outside of constraints

Inputs

- Username: "johndoe123"
- Password: "aaa"

Test Execution Steps

- User goes to the Register page
- User enters "johndoe123" for username
- User enters "aaa" for password
- User clicks "Register"

Expected Outcome for Validation (all validations must occur)

- User is redirected back to the Register Page
- User is shown the error message:
  - "Error - Registration Failed. Password must meet the following requirements:
    - At least 1 special character
    - At least one uppercase letter
    - At least one 1 number
    - Character length of 8-16"
- User is shown the "Try Again" button

Estimated Execution Time - 1 Minute

3.1.4 - Timeline

Creating an account is a core component of the system, so it must undergo exhaustive testing before any of the other features that depend on it can be tested. This means that all testing for this functionality must be completed in the beginning of the development lifecycle.

Testing for the account creation feature will begin in **Sprint 5 (1/20/22 - 2/3/22)** and must be completed before the end of Sprint 5.

Total Estimated Execution & Validation Time for Feature - 3 Minutes

3.2 - Login

Verification

The functionality of the login feature is verified from the application homepage. Part of the application homepage should be centered around a large UI component. For login, it displays empty text fields for the user to input their username and password.

Priority

High - Aside from the public search and discovery page, all other application features are only available once a user is logged in, thus it is a core component of the

application. This means that the login functionality is of high priority as many other features' functionalities are dependent on it.

System (End-to-End) Level Testing

System-level testing will be required because logging in will involve end-to-end communication across various system components such as the third party API, the UI components of the login feature, backend business logic, and the internal database.

Pre-Conditions

- Users are not signed into any account when attempting to log into their account.
- The user has a Hobby Project Generator account.
- User is trying to login to an existing account and not register for a new one.
- User must be on login view to attempt logging in

All of the following test cases must pass in order for the feature to be considered validated successfully

3.2.1 - Test Case 1: Valid login using credentials found in database

Inputs

User submits correct credentials into username and password inputs.

- Username: "johndoe123"
- Password: "Password!"

Test Execution Steps

- User goes to the public application homepage
- User enters "johndoe123" for username
- User enters "Password!" for password
- User clicks "Log in"

Expected Outcome for Validation (all validations must occur)

- User is directed to the private application homepage
- User has access to all logged in capabilities
- Logged in user can see a recommended list of projects personalized to the user

Estimated Execution Time - 1 Minute

3.2.2 - Test Case 2: Invalid login using credentials not found in database

Inputs

User submits incorrect credentials into username and password inputs.

- Username: "john"
- Password: "1234"

Test Execution Steps

- User goes to the public application homepage
- User enters "john" for username
- User enters "1234" for password
- User clicks "Login"

Expected Outcome for Validation (all validations must occur)

- User is redirected back to the public homepage
- User is shown the error message "Login Failed - Incorrect Username and/or Password"
- User is shown the "Try Again" button

Estimated Execution Time - 1 Minute

3.2.3 - Timeline

Because the login functionality is a core component of the system, it must undergo exhaustive testing before any of the other features that depend on it can be tested. This means that all testing for this functionality must be completed as early in the development lifecycle as possible.

Since the login feature is dependent on the user having an account to log in, in the first place, testing for this functionality cannot begin until the account creation and user questionnaire features have been fully tested first. Testing for the login functionality will begin in **Sprint 6 (2/4/22 - 2/18/22)** and be completed before the end of Sprint 6.

Total Estimated Execution & Validation Time for Feature - 2 Minutes

3.3 - Logout

Verification

The functionality of the login feature is verified from the private application homepage. When a user is logged in, their homepage should show a "Logout" button at the bottom of it.

Priority

Low - The need to log out of an account is not as demanding as logging in. Most users would not need to use this feature unless they had multiple accounts, want to see the website as a non-registered user, or for personal reasons. Since users will not gain anything from logging out, this feature is not as essential as the other functional features.

Integration Level Testing

Once a user is logged in, their credentials do not need to be checked with the system's internal database again when they log out. Additionally, no data logic on the frontend is needed to validate the user input when logging out. This means that Logout only requires integration testing for the communication between the UI components and the backend to end the user's authentication period and redirect them back to the public homepage.

Pre-Conditions

- Users must be signed in in order to logout.
- User is done with their session and wishes to log out of their account.
- User must be on view with the sign out option

All of the following test cases must pass in order for the feature to be considered validated successfully

3.3.1 - Test Case 1: Valid Logout using "Logout" button

Inputs

- None

Test Execution Steps

- Logged in user goes to the private homepage
- Logged in user clicks "Logout"

Expected Outcome for Validation (all validations must occur)

- User is directed to the public application homepage
- User is shown a message displaying, "Logout was successful"

Estimated Execution Time - 1 Minute

3.3.2 - Test Case 2: Invalid Logout using "Logout" button

Inputs

- None

Test Execution Steps

- Logged in user goes to the private homepage
- Logged in user clicks "Logout"

Expected Outcome for Validation (all validations must occur)

- User is logged out of their account
- User is not redirected to the homepage
- "Error: Logout operation error" or no message is displayed to the user

Estimated Execution Time - 1 Minute

3.3.3 - Test Case 3: Invalid Logout using "Logout" button

Inputs

- None

Test Execution Steps

- Logged in user goes to the private homepage
- Logged in user clicks "Logout"

Expected Outcome for Validation (all validations must occur)

- User is shown to have been logged out of their account
- User is redirected to the homepage
- User still has access to their personal account with their saved data

Estimated Execution Time - 1 Minute

3.3.4 - Test Case 4: Invalid Logout using "Logout" button

Inputs

- None

Test Execution Steps

- Logged in user goes to the private homepage
- Logged in user clicks "Logout"

Expected Outcome for Validation (all validations must occur)

- User is logged out of their account

- User is redirected to the homepage as a non-signed in user after the requirement of 5 seconds

Estimated Execution Time - 1 Minute

3.3.5 - Timeline

Even though the logout feature is of low priority, it makes logistical sense to test it at the same time as the login feature. Testing will begin in **Sprint 6 (2/4/22 - 2/18/22)** and be completed before the end of Sprint 6

Total Estimated Execution & Validation Time - 4 Minutes

3.4 - Questionnaire

Verification

After clicking the UI component to register on the New Account page, the user should be directed to a page to fill out the questionnaire. The functionality of the questionnaire feature is verified when the user sees the four UI components for each part of the questionnaire displayed on this page. Each of the four UI components should display a series of options with checkboxes that the user can interact with.

Priority

High - Aside from the public search and discovery page, all other application features are only available after a user has created their account and completed the questionnaire, thus it is a **core component** of the application. This means that the questionnaire feature is of high priority as many other features' functionalities are dependent on it.

Integration Level Testing

Integration testing will be required for the user questionnaire because it involves communication between the UI components that fill out the questionnaire and the system's internal database that will store the information as the user fills it out.

Pre-Conditions

- User cannot be signed into an account to access the questionnaire
- Username and password (initial credentials) must be chosen prior to completing the questionnaire.

All of the following test cases must pass in order for the feature to be considered validated successfully

3.4.1 - Test Case 1: Valid selection using one item in each section

Inputs

- Skill Sets: "Carpentry"
- Skill Levels: "Intermediate"
- Skill Interests: "Blacksmith"
- Tools: "Phillips Screwdriver"

Test Execution Steps

- User goes to the Questionnaire page after creating their username and password
- User selects "Carpentry" in the Skill Sets section
- User selects "Intermediate" in the Skill Levels section
- User selects "Blacksmith" in the Skill Interests section
- User selects "Phillips Screwdriver" in the Tools section.
- User clicks "Save" at the bottom of the page.

Expected Outcome for Validation (all validations must occur)

- User is directed to the User Profile Page.
- User's selections are displayed on their Profile Page

Estimated Execution Time - 1 Minute

3.4.2 - Test Case 2: Invalid selection using one (or more) empty section(s)

Inputs

- Skill Sets: none
- Skill Levels: "Intermediate"
- Skill Interests: "Blacksmith"
- Tools: "Phillips Screwdriver"

Test Execution Steps

- User goes to the Questionnaire page after creating their username and password
- User does not select any items in the Skill Sets section
- User selects "Intermediate" in the Skill Levels section
- User selects "Blacksmith" in the Skill Interests section
- User selects "Phillips Screwdriver" in the Tools section.
- User clicks "Save" at the bottom of the page.

Expected Outcome for Validation (all validations must occur)

- User is redirected back to the User Questionnaire Page.
- User is shown the error message "Error - The highlighted section cannot be left blank"
    - The UI component for Skill Sets is highlighted in red.

Estimated Execution Time - 1 Minutes

3.4.3 - Timeline

Because the questionnaire functionality is a core component of the system, it must undergo exhaustive testing before any of the other features that depend on it can be tested. This means that all testing for this functionality must be completed in the beginning of the development lifecycle.

Testing for the questionnaire feature will begin in **Sprint 5 (1/20/22 - 2/3/22)** and must be completed before the end of Sprint 5.

Total Estimated Execution & Validation Time for User Questionnaire - 2 Minutes

3.5 - Search

Verification

Logged in users and non-logged in users can search for projects using the private and private discovery pages, respectively. In either case, the search functionality is verified when the search box is shown at the top of the discovery page. The UI components for Search should include both a fillable text field and a filter button underneath it.

Priority

High - This functionality has a high priority because it is part of the definition of Hobby Project Generator. Both users who have an account with our websites and those who don't should be able to find the project that they wish to do.

Integration Level testing

Integration testing is required because "Search" functionality will be interacting with the user UI in order to ensure that correct output is displayed with different users' input. In addition, it also finds and lists items from the database.

Pre-Conditions

- Must be on the homepage where the search function is located.

All of the following test cases must pass in order for the feature to be considered validated successfully

3.5.1 - Test Case 1: Valid search using less than 30 characters

Input:

- Search text: "Tool Bench"

Test Execution Steps:

- User goes to Discovery Page
- User enters "Tool Bench" in the Search text field
- User clicks "Search"

Expected Outcome for Validation (all validations must occur):

- Discovery page is updated with search results
- The first 10 projects matching the search "Tool Bench" are shown in the results feed

Estimated Execution Time - 5 minutes

3.5.2 - Test Case 2: Invalid search using more than 30 characters

Input:

- Search text: "This search is more than thirty characters" is entered

Test Execution Steps:

- User goes to Discovery Page
- User enters "This search is more than thirty characters" in the Search text field
- User clicks "Search"

Expected Outcome for Validation (all validations must occur):

- User is redirected to the same Discovery page
- User is shown an error message "Error: Search Failed - Search text cannot exceed 30 characters"

Estimated Execution Time - 5 minutes

3.5.3 - Test Case 3: Invalid search using illegal characters (1 input is needed in order for invalid search)

Input:

- Search text: "¤£§¦«"

Test Execution Steps:

- User goes to Discovery Page
- User enters "¤£§¦«" for the search text
- User clicks "Search"

Expected Outcome for Validation (all validations must occur):

- User is redirected to the same Discovery page
- User is shown the error message "Error: Search Failed - Search text cannot contain illegal characters"

Estimated Execution Time - 5 minutes

3.5.4 - Timeline

Because "Search" is a necessary feature in order to navigate many projects on our website for both users with an account and those without, it needs to be tested to perform to the way we expected when validating inputs and returning results from the database.

Testing for the "Search" feature will begin in **Sprint 7 (02/19/22 - 03/05/2022)** and must be completed before the end of **Sprint 7**

Total Estimated Execution & Validation Time for testing Search - 15 minutes

3.6 - Search Filter

Verification

The "Filters" feature must appear directly below the search box with different options (store in a drop-down menu) that can be applied to the result of the search such as:

- Skill sets
- Available hardware
- Star rating (with an option to select between lowest to highest and vice versa)

- Number of votes (with an option to select between lowest to highest and vice versa)
- Date Posted (with an option to select newest to oldest and vice versa)

Priority

Medium - This feature will provide quality of life and convenience when it comes to viewing the results set of projects with titles that match the user input. It will help the user organize the results in a way that is appropriate to their needs. This feature however is not project breaking, even without the filter options, the "Search" function should still behave normally.

Integration level testing

Integration testing is required because this feature will need to work with Search features to arrange the result sets in different order.

Pre-Conditions

- Must be on the homepage where search feature is located

All of the following test cases must pass in order for the feature to be considered validated successfully

3.6.1 - Test case 1: Valid Search Filter using one filter selection

Inputs:

- Search Text: "Work Bench"
- Selected Filter: "Highest Rating to Lowest Rating"

Test Execution steps:

- User goes to Discovery Page
- User enters "Work Bench" in the search box
- User clicks "Search"
- User clicks "Filter By"
- User selects the "Highest to Lowest Rating" filter

Expected Outcome for Validation (all validations must occur):

- The Discovery page is updated
- Users are shown the top 10 results for "WorkBench" with the highest rating in descending order.

Estimated Execution Time - 1 minute

3.6.5 Timeline

Because "Search Filter" is a feature that applies to the result sets of the "Search" feature, it will be tested during the same time frame as "Search".

Testing for the "Search Filter" feature will begin in **Sprint 7 (02/19/22 - 03/05/2022)** and must be completed before the end of Sprint 7

Total Estimated Execution & Validation Time for Feature - 1 Minute

3.7 - Discovery Filter

Verification

Both the public and private versions of the Discovery page should show a feed of featured projects underneath the search box. Above this feed should be the UI components for three Filters labeled "Time", "Highest Rated" , and "Most Rated".

Priority

Medium - This filter will help the users navigate the variety of projects that are available in our database to their preferences. This feature will influence what projects will be displayed on the "Discovery" page so it is important for the "Discovery" functionality.

Integration Level Testing

This requires integration level testing because it will work in accordance with the set of projects displayed by the discovery page.

Pre-Conditions

- Must be on the discovery page in order to access the discovery filter

All of the following test cases must pass in order for the feature to be considered validated successfully

3.7.1 - Test case 1: Valid Filter using Highest Rated for Discovery Filter

Selected Filters

- Highest Rated
- Weekly

Test Execution steps:

- User goes to Discovery page
- User selects the Highest Rated filter
- User selects the Weekly filter

Expected Outcome for Validation (all validations must occur):

- Discovery page updates
- Users are shown top 10 projects with the highest ratings from the current week.

Estimated Execution Time - 5 minutes

3.7.1 - Test case 2: Valid Filter using Highest Rated for Discovery Filter

Selected Filters

- Most Rated
- Monthly

Test Execution steps:

- User goes to Discovery page
- User selects the Most Rated filter
- User selects the Monthly filter

Expected Outcome for Validation (all validations must occur):

- Discovery page updates
- Users are shown the top 10 projects with the most ratings from the current week.

Estimated Execution Time - 5 minutes

3.7.2 - Timeline

Since the filter will directly affect what projects will be displayed on the Discovery page, it will be tested along with the development of the "Discovery" section on **Sprint 7 (02/19/22 - 03/05/2022)**

Total Estimated Execution & Validation Time for testing Discovery Filter- 10 minutes

3.8 - UI Controls

Verification

After clicking any UI component on the website, it should redirect to its corresponding page, feature, or functionality. For our website, users who do not own an account are not able to interact with the same amount of items as someone who is. An example would be within the search and discovery page, logged in users are able to view and interact with an extra section of the page.

Priority

High - The UI controls are extremely important because if they do not work consistently and fluidly, then it would deter users from wanting to stay and interact more with the website. Anything that could impact the growth of our product is considered high priority.

Integration Level Testing

Integration testing will be required only interacting with something that shouldn't cause an event. The rest of the UI will be end-to-end tested with each feature or component at the end of each page's development in order to ensure that it not only looks modern, but that it also functions correctly.

Pre-Conditions

- Some UI controls will be accessible to logged in users only.

All of the following test cases must pass in order for the feature to be considered validated successfully

3.8.1 - Test Case 1: The user interacts with a non-interactable section of the UI.

Inputs

- User clicks on whitespace.

Test Execution Steps

- User randomly clicks a non-interactable area within the UI.

Expected Outcome for Validation (all validations must occur)

- Nothing happens.

Estimated Execution Time - 1 second

3.8.2 - Timeline

Total Estimated Execution & Validation Time - 1 second

The UI is a part of a majority of the features, so we will not be able to ensure that nothing happens when clicking on whitespace until all pages are finished. This will be completed at the very end of the project as a last minute check and it will be completed by the end of **Sprint 11(4/20/22 - 5/1/22)**.

3.9 - Notification Alerts

Verification

The user can be on any page within our website and will receive a notification alert. The user can visually locate the alert on the notification icon.

Priority

Low - The notification alerts are nice to have, but the user would still have access to the notification page.

Integration Level Testing

Integration testing will be required only interacting with something that shouldn't cause an event. The rest of the UI will be end-to-end tested with each feature or component at the end of each page's development in order to ensure that it not only looks modern, but that it also functions correctly.

Pre-Conditions

- User must be logged into an existing account

All of the following test cases must pass in order for the feature to be considered validated successfully

3.9.1 - Test Case 1: The user clicks on the notification icon.

Inputs

- User clicks on the notification icon

Test Execution Steps

- The user is alerted of a new notification.
- The user clicks on a notification.

Expected Outcome for Validation (all validations must occur)

- The 5 most recent notifications are displayed to the user

Estimated Execution Time - 30 seconds

3.9.4 - Timeline

Total Estimated Execution & Validation Time - 30 seconds

The notification alerts are relatively low priority. This is planned to be completed by the end of **Sprint 7(2/19/22 - 3/5/22)**.

3.10 - Notification Page

Verification

The user can visit the notification page from anywhere on the website. The user will be able to view all of their past notifications within this page.

Priority

High - The notification page is our main location of getting users their customized recommendations. They receive both tool and project recommendations through this as well as notifications for activity on their project posts or comments. This ultimately drives more user interaction and is considered very important.

Integration Level Testing

Integration testing will be required only interacting with something that shouldn't cause an event. The rest of the UI will be end-to-end tested with each feature or component at the end of each page's development in order to ensure that it not only looks modern, but that it also functions correctly.

Pre-Conditions

- User must be logged into an existing account
- User must have notifications in order to navigate to the notifications page

All of the following test cases must pass in order for the feature to be considered validated successfully

3.10.1 - Test Case 1: The user clicks on "Notification Page".

Inputs

- User clicks on "Notification Page".

Test Execution Steps

- The user selects and interacts with the "Notification Page" icon.

Expected Outcome for Validation (all validations must occur)

- The user is redirected to the Notification Page.
- All notifications are displayed to the user.

Estimated Execution Time - 30 seconds

3.10.2 - Test Case 2: The user clicks on a notification within the notification page.

Inputs

- User selects and interacts with a notification.

Test Execution Steps

- The user selects and interacts with any notification within the Notification Page.

Expected Outcome for Validation (all validations must occur)

- The user is redirected to the source of the notification.

Estimated Execution Time - 30 seconds

3.10.3 - Test Case 3: The user receives a new notification while viewing the notification page.

Inputs

- User receives notification.

Test Execution Steps

- The user is viewing the notification page when a new notification is received.

Expected Outcome for Validation (all validations must occur)

- The user is prompted to refresh the page to view the new notification.

Estimated Execution Time - 30 seconds

3.10.4 - Test Case 4: The user selects the notification page without being logged into an account

Inputs

- User clicks on the "Notification Page."

Test Execution Steps

- The user selects and interacts with the "Notification Page" icon.

Expected Outcome for Validation (all validations must occur)

- Notification list will show nothing since the user is not logged in.
- The user is given an error message - "Error - Must be logged in for this feature"
- Send users to the sign up page.

Estimated Execution Time - 30 seconds

3.10.4 - Timeline

The notification page is a high priority component of the project as avid users will need this feature for regular use. This is why this feature will need to be completed by the end of **Sprint 7(2/19/22 - 3/5/22)**.

Total Estimated Execution & Validation Time - 2 minutes

3.11 - Post Project

Verification

Once a user has signed into their account, they will have access to the "User Projects" page. At the bottom of this page should be a button labeled "Post New Project" which will redirect the user to the "Post/Edit Project" page when it is clicked.

The functionality of the Post Project feature is verified if all the necessary UI components for creating and posting a project are visible and accessible to the user on this page. The first three UI components - "Edit Project Name", "Edit Short Description", and "Edit Details" - are editable text fields that should initially be empty for each new project post. Of these, only the "Edit Short Description" field can be left empty when posting a project, in which case the Short Description will be auto filled with the first 200 characters from the body of the post.

Underneath this should be two additional UI components showing lists of items from the Skill Levels and Skill Categories sections of the User Questionnaire. Each of these items should have a checkbox next to it that the user can click on to select as a tag for the project. Next is shown the UI component allowing the user to select tools as tags for the project. This component is made up of a search box and an "Add Tool" button adjacent to it. The user can use the search box to search the system database for their tool. Once found, the user can click the "Add Tool" button to add the queried tool to the list of tagged tools for the project. After being added, the tool should appear as an additional UI component with the same style as the tagged items in the other three sections.

Below the project description, a UI component containing thumbnail previews of any images/files uploaded to the post. For a new post, this component should be empty. The "Edit Project" and "Attach/Remove Files" buttons should also appear here.

The bottom of the page should show the "Save" and "Post" buttons allowing the user to post their finished project or save it as a draft. Finally, the bottom left of the page should show a "Back to Projects" button which allows the user to back out of the page.

Priority

High - Having the functionality for users to post their projects is an important feature as we will need it to shift from pooling our projects from other sites to only user posted

System (End-to-End) Level Testing

System-level testing is required because posting a project will involve communication between multiple system components such as the public UI components where the user will input various elements of the post, the frontend data logic that will validate them and the internal databases that stores users' project posts.

Pre-Conditions

- Must be logged into an existing account in order to use this feature
- User must have a project/file ready to upload as a project

All of the following test cases must pass in order for the feature to be considered validated successfully

3.11.1 - Test Case 1: Valid Post using valid project data and selection

Inputs

- Title: "Project Title for Test"
- Short Description: "Sample Short Description under 200 characters"
- Details: "This is a sample with less than 200 character for the Details section of the project post"
- Skill Categories: Woodworking

- Skill Levels: Beginner
- Tools: Miter Saw

Test Execution Steps

- User goes to the Post/Edit Project Page
- User enters "Project Title for Test" for Title
- User enters "Sample Short Description under 200 characters" for Short Description
- User enters "This is a sample with less than 200 character for the Details section of the project post" for Details
- User selects the "Beginner" tag in Skill Levels
- User selects the "Woodworking" tag in Skill Categories
- User selects the "Miter Saw" tag in Tools
- User clicks "Post"

Expected Outcome for Validation (all validations must occur)

- User is directed to the new Project Page for their project
- User is shown the message "Project Posted" at the top of the page
- User is shown a Project Post view of their project

Estimated Execution Time - 5 Minutes

3.11.1 - Test Case 2: Invalid Post using invalid data

Inputs

- Title: "This Project Title for Test is Over 30 Characters"
- Short Description: None
- Details: None
- Skill Categories: Woodworking
- Skill Levels: Beginner
- Tools: Miter Saw

Test Execution Steps

- User goes to the Post/Edit Project Page
- User enters "Project Title for Test" for Title
- User leaves Short Description empty
- User leaves Details empty
- User selects the "Woodworking" tag in Skill Categories
- User selects the "Beginner" tag in Skill Levels
- User selects the "Miter Saw" tag in Tools
- User clicks "Post"

Expected Outcome for Validation (all validations must occur)

- User is redirected to the Post/Edit Project Page
- User is shown the error message:
  - "Error - Post Failed. Project post must meet the following requirements:
    - Must include a title that is under 30 characters without any illegal characters or vulgar expressions
    - Must include project details under 10,000 characters
    - Must include at least one tag selected in each of the project tag sections

Estimated Execution Time - 5 Minutes

3.11.3 - Test Case 3: Invalid Post using invalid selection

Inputs

- Title: "Project Title for Test"
- Short Description: "Sample Short Description under 200 characters"
- Details: "This is a sample with less than 200 character for the Details section of the project post"
- Skill Categories: Woodworking
- Skill Levels: None
- Tools: None

Test Execution Steps

- User goes to the Post/Edit Project Page
- User enters "Project Title for Test" for Title
- User leaves Short Description empty
- User leaves Details empty
- User selects the "Woodworking" tag in Skill Categories
- User doesn't select any tag in Skill Levels
- User doesn't select any tag in Tools
- User clicks "Post"

Expected Outcome for Validation (all validations must occur)

- User is redirected to the Post/Edit Project Page
- User is shown the error message:
  - "Error - Post Failed. Project post must meet the following requirements:
    - Must include a title that is under 30 characters without any illegal characters or vulgar expressions
    - Must include project details under 10,000 characters

■ Must include at least one tag selected in each of the project tag sections

3.11.4 - Test Case 4: Project post from a user that is not logged in

Inputs

- Title: "Project Title for Test"
- Short Description: "Sample Short Description under 200 characters"
- Details: "This is a sample with less than 200 character for the Details section of the project post"
- Skill Categories: Woodworking
- Skill Levels: None
- Tools: None

Test Execution Steps

- User goes to the Post/Edit Project Page
- User enters "Project Title for Test" for Title
- User leaves Short Description empty
- User leaves Details empty
- User selects the "Woodworking" tag in Skill Categories
- User doesn't select any tag in Skill Levels
- User doesn't select any tag in Tools
- User clicks "Post"

Expected Outcome for Validation (all validations must occur)

- User is shown the error message - "Error - Must be logged in for this feature."
- Send users to the sign up page.

Estimated Execution Time - 5 Minutes

3.11.4 - Timeline

Total Estimated Execution & Validation Time for Feature - 20 Minutes

Being able to post a project is a core component of the application specific features, so it should be tested as early as possible in the development lifecycle. Its testing is dependent on the public features being developed and fully tested first, however, so testing will begin in **Sprint 6 (2/4/22 - 2/18/22**) and will be completed by the end of Spring 6

3.12 - Upload File to Project Post

Verification

After clicking on the Attach/Remove Files button on the Post/Edit Project page, the user should be redirected to the Attach/Remove Files page allowing them to upload and/or remove images and files. This page should show thumbnail previews of all the images/files that are currently uploaded along with a UI component allowing the user to check or uncheck each thumbnail. At the bottom of the page should be three UI components - buttons labeled "Remove Checked Files", "Attach Files", and "Done" - that the user can interact with.

Priority

Medium - Image and file uploads are an auxiliary functionality of the larger feature that allows users to post their projects, so there are no other features that depend on its functionality to work. Images and their corresponding thumbnails that appear with project previews are still an essential part of the application's visual appeal and accessibility, however, so this functionality is of medium priority.

Integration Testing

Integration testing will be required because uploading files to project posts will require communication between the UI components that the user interacts with and the data logic components that validate the uploads.

Pre-Conditions

- Must be logged into an existing account in order to use this feature
- Must have a file that can be added to the project post

All of the following test cases must pass in order for the feature to be considered validated successfully

3.12.1 - Test Case 1: Valid Upload using valid file types and sizes

Inputs

- File upload 1: A 'test.png' file under 5MB
- File upload 2: A 'test.pdf' file under 15MB

Test Execution Steps

- User goes to the Attach/Remove Files page
- User clicks "Attach Files"
- User attaches 'test.png' for File 1
- User attaches 'test.pdf' for File 2
- User clicks "Done" at the bottom of the page.

Expected Outcome for Validation (all validations must occur)

- User is directed back to the Post/Edit Project page
- User is shown thumbnail previews of File 1 and File 2 in the project post

Estimated Execution Time - 5 Minutes

3.12.2 - Test Case 2: Valid Upload using invalid file types and sizes

Inputs

- File 1: A 'test.png' file over 5 MB
- File 2: A 'test.pdf' file over 15MB
- File 3: A 'test.mp4' file over 30MB

Test Execution Steps

- User goes to the Attach/Remove Files page
- User clicks "Attach Files"
- User attaches 'test.png' for File 1
- User attaches 'test.pdf' for File 2
- User attaches 'test.mp4' for File 3
- User clicks "Done" at the bottom of the page.

Expected Outcome for Validation (all validations must occur)

- User is redirected back to the Attach/Remove File(s) page
- User is shown the error message:
  - "Error: Attach Failed - File uploads must follow the requirements:
    - Only uploads of type .PNG, .BMP, .JPEG, or PDF are accepted
    - Individual .PNG, .BMP, and .JPEG upload sizes cannot exceed 5 MB
    - Individual .PDF upload sizes cannot exceed 5 MB
    - Total upload size cannot exceed 30 MB"

Estimated Execution Time - 5 Minutes

3.12.3 - Timeline

Total Estimated Execution & Validation Time for Feature - 10 Minutes

Uploading files to a project post is an auxiliary functionality of the larger feature allowing users to upload their own project posts, so it cannot be tested until the Post/Delete Project feature has been fully tested. This means that the testing will begin after the development of the Post/Delete Project" feature in **Sprint 6 (2/4/22 - 2/18/22)** and must be completed before the end of Sprint 6.

3.13 - Remove File from Project Post

Verification

After clicking on the "Attach/Remove Files" button on the "Post/Edit Project" page, the user should be redirected to the "Attach/Remove Files" page allowing them to upload and/or remove images and files. This page should show thumbnail previews of all the images/files that are currently uploaded along with a UI component allowing the user to check or uncheck each thumbnail. At the bottom of the page should be three UI components - buttons labeled "Remove Checked Files", "Attach Files", and "Done" - that the user can interact with.

Priority

Low - Image and file uploads are an auxiliary functionality of the larger feature that allows users to post their projects, so there are no other features that depend on its functionality to work. Since the images and their corresponding thumbnails that appear with project previews are an essential part of the application's visual appeal and accessibility, the functionality for file uploads is of medium priority. Removing files from a post is less important, however, so it is of low priority.

Integration Testing

Integration testing will be required because uploading files to project posts will require communication between the UI components that the user interacts with and the data logic components that validate the uploads.

Pre-Conditions

- Must be logged into an existing account in order to use this feature
- Have to have a project post to remove a file from

All of the following test cases must pass in order for the feature to be considered validated successfully

3.13.1 - Test Case 1: Valid removal using selected files

Inputs

- File Selected: 'test.jpeg'

Test Execution Steps

- User goes to the Attach/Remove Files page
- User selects the file preview for 'test.jpeg'
- User clicks "Remove Files" at the bottom of the page

- User clicks "Done" at the bottom of the page.

Expected Outcome for Validation (all validations must occur)

- User is directed back to the Post/Edit Project page
- User is shown the project post without previews of the removed file

Estimated Execution Time - 5 Minutes

3.13.2 - Timeline

Total Estimated Execution & Validation Time for Feature - 5 Minutes

Uploading files to a project post is an auxiliary functionality of the larger feature allowing users to upload their own project posts, so it cannot be tested until the Post/Delete Project feature has been fully tested. This means that the testing will begin after the development of the Post/Delete Project" feature in **Sprint 6 (2/4/22 - 2/18/22)** and must be completed before the end of Sprint 6.

3.14 - Save Project as Draft

Verification

Once a user has signed into their account, they will have access to the "User Projects" page. At the bottom of this page should be a button labeled "Post New Project" which will redirect the user to the "Post/Edit Project" page when it is clicked.

The bottom of the page should show the "Save" and "Post" buttons allowing the user to post their finished project or save it as a draft. Finally, the bottom left of the page should show a "Back to Projects" button which allows the user to back out of the page.

Priority

Low - The ability to save a project post as a draft is not depended on by the functionality of any other features, so it is of low priority relative to the other functionalities of posting a project.

Integration Level Testing

Since saving a post as a draft does not require that the post be hosted live on the application's public homepage (until it is fully posted, that is), it does not require end-to-end level testing involving the public web server. It only requires integration level testing for the UI components with the user's data and the internal database

Pre-Conditions

- Must be logged into an existing account in order to use this feature
- Must have some material to save as draft

All of the following test cases must pass in order for the feature to be considered validated successfully

3.14.1 - Test Case 1: Valid Save using Draft with valid project data

Inputs

- Title: "Project Title for Test"
- Short Description: None
- Details: None
- Skill Categories: None
- Skill Levels: None
- Tools: None

Test Execution Steps

- User goes to the Post/Edit Project Page
- User enters "Project Title for Test" for Title
- User leaves Short Description empty
- User leaves Details empty
- User leaves Skill Categories empty
- User leaves Skill Levels empty
- User leaves Tools empty
- User clicks "Save"

Expected Outcome for Validation (all validations must occur)

- User is directed to the Project Page of their project
- User is shown the message "Project Saved as Draft" at the top of the page
- User is shown a Project Post view of their saved draft

Estimated Execution Time - 5 Minutes

3.14.2 - Test Case 2: Invalid Save using Draft with invalid project data

Inputs

- Title: None
- Short Description: None
- Details: None
- Skill Categories: None

- Skill Levels: None
- Tools: None

Test Execution Steps

- User goes to the Post/Edit Project Page
- User leaves Title empty
- User leaves Short Description empty
- User leaves Details empty
- User leaves Skill Categories empty
- User leaves Skill Levels empty
- User leaves Tools empty
- User clicks "Save"

Expected Outcome for Validation (all validations must occur)

- User is redirected back to the Post/Edit Project Page
- User is shown the error message: "Error: Save Failed - Cannot save draft without any data"

Estimated Execution Time - 5 Minutes

3.14.3 - Timeline

Total Estimated Execution & Validation Time for Feature - 10 Minutes

After initial developments of uploading,editing, and posting a user project, users will then have the ability to save their project after they have published their project. This will take place during **Sprint 7: (2/19/22 - 3/5/22)** right after the earlier modifications of posting a project because a project cannot be edited without those earlier functions in sprint 6.

3.15 - Edit Project

Verification

The user can edit a project that has already been posted and/or one that they have saved as a draft by clicking on the "Edit Project" button shown directly underneath the "Project Description and Details'' UI component on the "Post/Edit Project" page. After clicking on the button, the user is directed to the "Edit Project" page.

This page should display three UI components labeled "Edit Project Name", "Edit Short Description", and "Edit Details'' - each with an editable text field containing any text that is already written in that section of the project post. Underneath this should be another

UI component with four sections containing the chosen attributes from the Skills, Skill Levels, Skill Categories, and Tools categories. used in the user questionnaire) that are currently selected for this project. At the bottom of the page should be a final UI component for a button labeled "Save".

Priority

Medium - Users will need a way to alter their projects after the initial upload, making the Edit a Posted Project feature crucial to our application.

System (End-to-End) Level Testing

System-level testing is required because editing a project that has already been posted will involve communication between multiple system components such as the public UI components where the user will input various elements of the post, the frontend data logic that will validate them and the internal databases that stores users' project posts.

Pre-Conditions

- Must be logged into an existing account in order to use this feature
- Have a project where edits can be made

All of the following test cases must pass in order for the feature to be considered validated successfully

3.15.1 - Test Case 1: Valid Edit using Title with valid characters

Inputs

- Title edit: "Title with valid data"
- Short Description edit: None
- Details edit: None
- Skill Categories edit: None
- Skill Levels edit: None
- Tools edit: None

Test Execution Steps

- User goes to the Post/Edit Project Page of the project they want to edit
- User deletes any text currently in the Title
- User enters "Title with valid data" for Title
- User clicks "Save"

Expected Outcome for Validation (all validations must occur)

- User is directed to the new Project Page of their project
- User is shown their project with the new title "Title with valid data"

Estimated Execution Time - 5 Minutes

3.15.2 - Test Case 2: Invalid Edit using Title with invalid characters

Inputs

- Title edit: "Title with Illegal Characters +*%"
- Short Description edit: None
- Details edit: None
- Skill Categories edit: None
- Skill Levels edit: None
- Tools edit: None

Test Execution Steps

- User goes to the Post/Edit Project Page of the project they want to edit
- User deletes any text currently in the Title
- User enters "Title with Illegal Characters +*%" for Title edit
- User clicks "Save"

Expected Outcome for Validation (all validations must occur)

- User is redirected to the Post/Edit Project Page of their project
- User is shown the error message:
  - "Error - Edit Failed. Project post must meet the following requirements
    - Must include a title that is under 30 characters without any illegal characters or vulgar expressions
    - Must include project details under 10,000 characters
    - Must include at least one tag selected in each of the project tag sections"

Estimated Execution Time - 5 Minutes

3.15.2 - Timeline

Total Estimated Execution & Validation Time for Feature - 10 Minutes

After initial developments of uploading, posting, and saving a user project, users will then have the ability to edit their project after they have published their project. This will take place during **Sprint 7: (2/19/22 - 3/5/22)** right after the earlier modifications of posting a project because a project cannot be edited without those earlier functions in sprint 6.

3.16 - Delete Project

Verification

Verification occurs when, from the "Project" or "Project Draft" pages, the user clicks "Delete Project". A pop up will display the message "Are you sure you want to delete?" to the user allowing them to confirm or cancel the deletion.

Priority

Medium - This feature is necessary and important to our application as both the user and development team need a way to remove a project or project draft from our site

Integration Testing

Since deleting a project requires that it was previously hosted live on the application's public homepage (until it is fully posted, that is), it does not require end-to-end level testing involving the public web server. It only requires integration level testing for the UI components with the user's data and the internal database

Pre-Conditions

- Must be logged into an existing account in order to use this feature
- Have a project that user can delete

All of the following test cases must pass in order for the feature to be considered validated successfully

3.16.1 - Test Case 1: User deletes one a previously posted project

Test Execution steps:

- User logs in
- User goes to the Post/Edit Project Page
- User clicks "Delete"

Expected Outcome for Validation (all validations must occur):

- Post/Edit Project page is updated so that it no longer shows the deleted project data
- User is shown the message "Project Deleted" at the top of the page.

Estimated Execution Time - 5 minutes

3.16.2 - Timeline

The deletion feature will be implemented during **Sprint 6: (2/4/22 - 2/18/22)** where other similar features are based on projects and draft projects. The reasoning for this is so

that the implementation and testing will be done one after another since they are similar tasks.

Total Estimated Execution & Validation Time for Feature - 5 Minutes

3.17 - Rate Project

Verification

While on the "Project" page, verification occurs when the user is able to see the UI element for ratings, showing 5 stars that are grayed out if the user hasn't previously selected a rating or yellowing the correct number of stars to represent the user's previous rating.

Priority

Medium - The Rate a Project (or edit current rating) feature is not crucial to development of the application but will be used for sort filters and recommendations.

Integration Level Testing

Integration testing is required because of the communication between the private UI components the user interacts with to rate a project and the frontend data logic that validates their input.

Pre-Conditions

- Must be logged into an existing account in order to use this feature
- Must choose an existing project to rate

All of the following test cases must pass in order for the feature to be considered validated successfully

3.17.1 - Test Case 1: Valid project rating using the initial rating from a user

Input:

- Initial rating: none
- New rating: 3 stars

Test Execution Steps:

- User goes to project page
- User selects 3 of the 5 possible stars for project rating
- User refreshes page

40

Expected Outcome for Validation (all validations must occur):

- User is directed to the updated project page
- User is shown the project rating with 3 stars selected

Estimated Execution Time - 20 seconds

3.17.2 - Test Case 2: Invalid project rating using a new rating from the the user

Input:

- Initial rating: 3 stars
- New rating: 5 stars

Test Execution Steps:

- User goes to project page
- User selects all 5 of the 5 possible stars for project rating
- User refreshes page

Expected Outcome for Validation (all validations must occur):

- User is directed to the updated project page
- User is shown the project rating with 5 stars selected

Estimated Execution Time - 10 seconds

3.17.3 - Test Case 3: Valid Project rating using a new rating that is identical as the user's previous name

Input:

- Initial rating: 5 stars
- New rating: 5 stars

Test Execution Steps:

- User goes to project page
- User selects all 5 of the 5 possible stars for project rating
- User refreshes page

Expected Outcome for Validation (all validations must occur):

- User is redirected to initial project page
- User is shown the initial project rating with 5 stars selected

Estimated Execution Time - 5 seconds

3.17.4 - Test Case 4: Valid Project rating without being logged into an account

Input:

- Initial rating: 5 stars
- New rating: 5 stars

Test Execution Steps:

- User goes to project page
- User selects all 5 of the 5 possible stars for project rating
- User refreshes page

Expected Outcome for Validation (all validations must occur):

- The user is given an error message - "Error - Must be logged in for this feature"
- Send users to the sign up page.

Estimated Execution Time - 5 seconds

3.17.2 - Timeline

Total Estimated Execution & Validation Time - 40 seconds

Rating a project will take place during **Sprint 10: (4/5/22 - 4/19/22)**. This is a feature that is of medium priority and since none of the other features will build upon the rating of a project, we will test this at the end of the life cycle.

3.18 - Post Comment

Verification

On an individual project page, the user navigates to the "Comments'' section towards the bottom of the page. There, they should be able to view and have the option to post a comment of their own.

Priority

Medium - The comment section serves as a place for users to discuss amongst each other anything that they wished to. It does not directly affect the main goal of the "Hobby Project Generator" to recommend projects to users.

Integration Level Testing

Pre-Conditions

- Must be logged into an existing account in order to use this feature
- Navigate to the project user wants to comment on

All of the following test cases must pass in order for the feature to be considered validated successfully

Valid comments

3.18.1 - Test Case 1: Valid Project Comment using valid text

Input:

- Comment Text: "Great project, very informative."

Test Execution Steps:

- The user goes to the Comments page for the project
- The user enters "Great project, very informative" for Write New Comment
- The user clicks "Post Comment".

Expected Outcome for Validation (all validations must occur):

- User is shown the refreshed Comments page
- User is shown their comment "Great project, very informative" at the top of the updated Comments section

Estimated Execution Time - 1 minute

3.18.2 - Test Case 2: Valid Project Comment using valid file upload

Input:

- Comment Text: "Valid Comment"
- File upload 1: A "test.png" file under 5MB

Test Execution Steps:

- User goes to the Comments page of a project
- User enters "Valid Comment" for Write New Comment
- User clicks "Upload Image"
- User attaches the 'test.png' file
- User clicks "Post New Comment"

Expected Outcome for Validation (all validations must occur):

- User is shown the refreshed Comments page

- User is shown their comment with the text "Valid Comment" and thumbnail preview of the "test.png" file at the top of the updated Comments section

Estimated Execution Time - 1 Minute

3.18.3 - Test Case 3: Invalid Project Comment using invalid text

Input:

- Comment Text: "¤£§¦«"

Test Execution Steps:

- The user goes to the Comments page for the project
- User enters "¤£§¦«" for Write New Comment
- The user clicks "Post Comment".

Expected Outcome for Validation (all validations must occur)

- The user is shown the error message: "Error: Comment not Posted - Comments must be between 1 and 500 characters in length and cannot contain any illegal characters or vulgar expressions".

Estimated Execution Time - 5 seconds

3.18.4 - Test Case 4: Invalid Project Comment using invalid file upload

Input:

- Comment Text: "Valid Comment"
- File upload 1: A "test.mp4" file over 5 MB

Test Execution Steps:

- User goes to the Comments page of a project
- User enters "Valid Comment" for Write New Comment
- User clicks "Upload Image"
- User attaches the 'test.mp4' file
- User clicks "Post New Comment"

Expected Outcome for Validation (all validations must occur):

- User is redirected back to the Comments page
- User is shown the error message: "Error: Comment not Posted - Only .PNG, .JPEG, or .BMP files under 5 MB can be uploaded to a comment."

Estimated Execution Time - 20 seconds

3.18.5 - Test case 5: Multiple PNG, BMP, or JPEG files totaling less than 30 MB

Inputs:

- "Helloworld.png" - 4 MB
- "Hi.png" - 4 MB
- "Shark.png" - 4 MB

Test Execution steps:

- The user goes to the post a comment section.
- The user clicks on "upload image" while creating a comment.
- User select "Helloworld.png", "Hi.png", "Shark.png" from their files and clicks "upload".

Expected Outcome for Validation (all validations must occur):

- The files are successfully converted to .jpg and added to the comment.

Estimated Execution Time - 3 minutes

3.18.6 - Test case 6: Multiple PNG, BMP, or JPEG files totaling more than 30 MB

Inputs:

- "Helloworld.png" - 10 MB
- "Hi.png" - 15 MB
- "Shark.png" - 10 MB

Test Execution steps:

- The user goes to the post a comment section.
- The user clicks on "upload image" while creating a comment.
- User select "Helloworld.png", "Hi.png", "Shark.png" from their files and clicks "upload".

Expected Outcome for Validation (all validations must occur):

- The user is shown an error message on the same page: "Error: Total upload cannot exceed 30 MB".
- Return to the file selection screen

Estimated Execution Time - 3 minutes

3.17.7 - Test Case 7: Valid Project Comment using valid text without being logged into an account

Input:

- Comment Text: "Great project, very informative."

Test Execution Steps:

- The user goes to the Comments page for the project
- The user enters "Great project, very informative" for Write New Comment
- The user clicks "Post Comment".

Expected Outcome for Validation (all validations must occur):

- The user is given an error message - "Error - Must be logged in for this feature"
- Send users to the sign up page.

Estimated Execution Time - 1 minute

3.18.2 - Timeline

Total Estimated Execution & Validation Time - 8 minutes 45 seconds

The comment section can only be completed after the initial project page is done, it will have to be one of the later features. Comment section will be done by the end of **Sprint 9: (4/5/22 - 4/19/22)**

3.19 - Rate Comment

Verification

In order to rate a comment, the user must be registered and signed into their account and navigate to the comment section by clicking on a project page and pressing the UI component under description. Inside the comment section, users can choose to rate comments based on their individual opinion. The GUI will show the user that the rating of a comment has gone through by updating the number of upvotes/downvotes of the particular comment.

Priority

Low - Similar to filtering comments, this feature builds upon a core feature so getting comments finalized with functional error handling and proper testing would come before

allowing users to rate comments. In addition, this feature's purpose is to improve user experience and is another way for users to get quick feedback for projects they publish.

Integration Testing

The rating system would need to be tested so that user experience is not compromised by an invalid rating counter or UI glitches. Adequate error handling practices will be implemented to keep the user from encountering a broken counter or UI component.

Pre-Conditions

- Must be logged into an existing account in order to use this feature
- Find a comment to rate

All of the following test cases must pass in order for the feature to be considered validated successfully

3.19.1 - Test Case 1: Valid Comment Rating using a valid selection from logged in user

Input:

- Comment Rating: "Upvote" or "Downvote" selected
- Rating will be logged for future error handling and validation

Test Execution Steps:

- User signs in.
- User goes to the Comments page of a project.
- User clicks the upvote or downvote icon on the first comment in the list
- User refreshes page

Expected Outcome for Validation (all validations must occur):

- Rating counter increments or decrements by one depending on which option the user chose to rate the project comment.
- User is shown the highlighted upvote icon of their rated comment
- Rating will be logged for future error handling and validation

Estimated Execution Time - 30 seconds

3.19.2 - Test Case 2: Valid Comment Rating using a valid selection from a user that is not logged in

Input:

- Comment Rating: "Upvote" or "Downvote" selected

Test Execution Steps:

- User logs out.
- User goes to the Comments page of a project.
- User clicks the upvote or downvote icon on the first comment in the list
- User refreshes page

Expected Outcome for Validation (all validations must occur):

- User is redirected back to the project page
- User is shown the error message: "Error - Only logged in users are able to rate comments"
- Rating counter stays the same amount and there is no indication that a rating attempt has been made other than an error message.

Estimated Execution Time - 30 seconds

3.19.2 - Timeline

Total Estimated Execution & Validation Time for Feature - 1 Minute

Since the priority of rating a comment is not as demanding as the other more demanding features, testing this feature will not be a big concern for the entire project. The rating feature would come after the filtration system but are both similar in importance.

Testing the filter would occur in **Sprint 9: (3/21/22 - 4/4/22)** because the comment section precedes actually rating the comments and only editing the rating builds upon this feature. Comments will be implemented in the previous sprint so the rating system is not possible until the comment section is programmed, finalized, and fully-tested.

3.20 - Filter Comments

Verification

When registered users are signed in, they are able to comment on other project pages with their input, recommendations, and suggestions. When clicking on a project page, under the description there will be a UI component that allows you to look at comments for that particular project. From this comment page, filtration will occur on its own when the system detects inappropriate or discriminatory statements previously whitelisted into the website.

Priority

Low - The filtering of comments depends on many functions but no further features depend on filtering comments. Therefore, this will be an extra feature to optimize our user experience through preventing users commenting bigotry throughout the comment sections.

Integration Testing

Several test cases will be used to scope out the functionality of the filtering of comments so that users are not bypassing the filtering system through a bug or error.

Pre-Conditions

- None

All of the following test cases must pass in order for the feature to be considered validated successfully

3.20.1 - Test Case 1: Valid Comment Filter using the Highest to Lowest Filter

Input:

- Comment Filter: "Highest to Lowest" selected

Test Execution Steps:

- User goes to the Comments section of a project post
- User clicks the "Filter By" button
- User selects "Highest to Lowest" from the filter options

Expected Outcome for Validation (all validations must occur):

- User is shown the updated Comments page with the comments sorted by highest to lowest number of upvotes

Estimated Execution Time - 30 seconds

3.20.2 - Test Case 2: Valid Comment Filter using Lowest to Highest Filter

Input:

- Comment Filter: "Lowest to Highest" selected

Test Execution Steps:

- User goes to the Comments section of a project post
- User clicks the "Filter By" button
- User selects "Lowest to Highest" from the filter options

Expected Outcome for Validation (all validations must occur):

- User is shown the updated Comments page with the comments sorted by lowest to highest number of upvotes

Estimated Execution Time - 30 seconds

3.20.3 - Test Case 3: Valid Comment Filter using the Most Rated Filter

Input:

- Comment Filter: "Most Rated" selected

Test Execution Steps:

- User goes to the Comments section of a project post
- User clicks the "Filter By" button
- User selects "Most Rated from the filter options

Expected Outcome for Validation (all validations must occur):

- User is shown the updated Comments page with the comments sorted by highest total ratings first

Estimated Execution Time - 30 seconds

3.20.4 - Test Case 4: Valid Comment Filter using the Newest Filter

Input:

- Comment Filter: "Newest" selected

Test Execution Steps:

- User goes to the Comments section of a project post
- User clicks the "Filter By" button
- User selects "Newest" from the filter options

Expected Outcome for Validation (all validations must occur):

- User is shown the updated Comments page with the comments sorted by the newest comments first

Estimated Execution Time - 30 seconds

3.20.5 - Test Case 5: Valid Comment Filter using the Oldest Filter

Input:

- Comment Filter: "Oldest" selected

Test Execution Steps:

- User goes to the Comments section of a project post
- User clicks the "Filter By" button
- User selects "Oldest" from the filter options

Expected Outcome for Validation (all validations must occur):

- User is shown the updated Comments page with the comments sorted by the oldest comments first

Estimated Execution Time - 30 seconds

3.21.6 - Timeline

Total Estimated Execution & Validation Time for Feature - 2.5 Minutes

Since the priority of filtering comments is not as demanding as the other more demanding features, testing this feature will not be a big concern for the entire project. Testing the filter would occur in **Sprint 8 (3/6/22 - 3/20/22)** because the comment section precedes filtering and none of the other features builds upon the filtration of comments. Furthermore, comments will also be implemented in this sprint so the filtration system is not possible until the comment section is programmed and finalized.

3.21 - Project Recommendation

Verification

From the private home page, the user clicks the "Discovery" button to navigate to the Discovery page. On the Discovery page, the user's recommended projects should be shown on the as thumbnails previews populating the feed section of the page underneath the search bar.

Priority

High - Recommended projects is one of the main ways our users are able to receive a personalized experience when navigating throughout the website. For this reason, the priority to get this feature up and running is key for the purpose of the website.

System (End-to-End) Testing

Recommended projects must undergo end-to-end testing to ensure that the user's attributes and tags from their questionnaire inputs are being correctly matched to the corresponding attributes and tags of the projects that are recommended to them in their discovery feed.

Pre-Conditions

- Must be logged into an existing account in order to use this feature
- Questionnaire must be filled out for projects to be recommended

All of the following test cases must pass in order for the feature to be considered validated successfully

3.21.1 Test Case 1: Valid Recommendation using Project that matches User Questionnaire tags

Input (User's Tags from Questionnaire)

- Skill Sets: Blacksmith
- Skill Levels: Intermediate
- Skill Interests: Woodworking
- Tools: Circular Saw

Test Execution Steps

- User logs in to their account
- User goes to Discovery page
- User clicks on the first recommended project

Expected Outcome for Validation (all validations must occur)

- Project's Skill Categories section has the Woodworking and/or Blacksmith tag(s)
- Project's Skill Level section contains either the Intermediate or Beginner tag

- Project's Tools section contains the Circular Saw tag

Estimated Execution Time - 5 Minute

3.21.1 Test Case 2: Invalid Recommendation using Project that doesn't match User Questionnaire tags

Input (User's Tags from Questionnaire)

- Skill Sets: Blacksmith
- Skill Levels: Intermediate
- Skill Interests: Woodworking
- Tools: Circular Saw

Test Execution Steps

- User logs in to their account
- User goes to Discovery page
- User clicks on the first recommended project

Expected Outcome for Validation (all validations must occur)

- Project's Skill Categories section does not have the Blacksmith tag or Woodworking tag
- Project's Skill Level section has the Advanced tag
- Project's Tools section does not have the Circular Saw tag

Estimated Execution Time - 5 Minute

3.21.2 - Timeline

Total Estimated Execution & Validation Time for Feature - 10 Minutes

Although this feature has a high priority for the website, it is dependent on many other features such as the questionnaire and users with accounts. These features need to be implemented prior to the recommended projects section so that the projects can use the questionnaire to build a database where users can be fed a few recommended projects every week. For this reason, this feature will be worked on **Sprint 8: (3/6/22 - 3/20/22)** of the life cycle for the project.

3.22 - Receive a Tool Recommendation

Verification

From the notification page, the user is able to view their recommended tools based on their answers on the questionnaire and how many new projects the tool would open up for them.

Priority

High - Tool recommendations are a key component in allowing both our user's and our product to grow. These tool recommendations open possibilities for our users to take on a reasonable amount of new projects with just the addition of one new tool. With the growth of our user base and user projects, it opens future possibilities of monetizing these recommendations.

Integration Testing

Recommended projects can be tested through test cases to check the functionality of the section.

Pre-Conditions

- Must be logged into an existing account in order to use this feature
- Questionnaire must be filled out for projects to be recommended

All of the following test cases must pass in order for the feature to be considered validated successfully

3.22.1 - Test Case 1: Logged in users are able to view and interact with their recommended tools.

Inputs

- User action: Clicks on tool recommendation within notification page

Test Execution Steps

- User goes to the notification page
- User selects and interacts with the "Tool recommended for you" notification

Expected Outcome for Validation (all validations must occur)

- The user is redirected to the tool recommendation page where they are able to view an image of the tool, a description of the tool, and project recommendations with the tool.

Estimated Execution Time - 1 Minute

3.22.2 - Test Case 2: Logged in users are unable to view and interact with their recommended tools.

Inputs

- User action: Clicks on tool recommendation within notification page

Test Execution Steps

- User goes to the notification page
- User selects and interacts with the "Tool recommended for you" notification

Expected Outcome for Validation (all validations must occur)

- Nothing happens

Estimated Execution Time - 30 seconds

3.22.3 - Timeline

Total Estimated Execution & Validation Time - 1 Minute 30 seconds

Tool recommendations work hand in hand with project recommendations because both are of high priority and have similar function purposes. To make sure this component is implemented adequately, **Sprint 8: (3/6/22 - 3/20/22)** will contain this feature along with a few others so that this portion could be thoroughly functional and tested. This is the same sprint as project recommendations so that these similar features could be coupled together to be worked on the same sprint.

3.23 - Project Collection

Verification

Projects will populate the discovery page and will also show up in the database. Only 5 lines will be used to describe the project.

Priority

High - Project collection is a core component for our product because we need a large enough initial batch for our users to browse and interact with. A majority of the features are tied to using the collected projects.

Integration Testing

Recommended projects can be tested through test cases to check the functionality of the section.

Pre-Conditions

- Sites that projects are collected from must be up and running

All of the following test cases must pass in order for the feature to be considered validated successfully

3.23.1 - Test Case 1: Projects populate the database and are viewable through a query.

Inputs

- Text input: "select Titles from Projects"

Test Execution Steps

- Super user connects to the database
- Super user enters "select * from Projects"

Expected Outcome for Validation (all validations must occur)

- A table with the titles of all projects collected are displayed.

Estimated Execution Time - 1 Minute

3.23.2 - Test Case 2: Projects with full original contents populate the discovery page.

Inputs

- Data input: Projects from database

Test Execution Steps

- User goes to search and discovery page
- User selects and interacts with one of the projects within the page

Expected Outcome for Validation (all validations must occur)

- User is redirected to the project page and is able to view its full contents.

Estimated Execution Time - 1 Minute

3.23.3 - Test Case 3: Projects with partial contents populate the search and discovery page.

Inputs

- Data input: Projects from database

Test Execution Steps

- User goes to search and discovery page
- User selects and interacts with one of the projects within the page

Expected Outcome for Validation (all validations must occur)

- User is redirected to the project page, but unable to view full contents of the project.

Estimated Execution Time - 1 Minute

3.23.4 - Test Case 4: Collects the same project that is already in the database.

Inputs

- Text input: "select Titles, COUNT(Titles) from Projects GROUP BY Titles Having COUNT(Titles)>1"

Test Execution Steps

- Super user connects to the database
- Super user enters "select Titles, COUNT(Titles) from Projects GROUP BY Titles Having COUNT(Titles)>1"

Expected Outcome for Validation (all validations must occur)

- There should be no output, if an output is given then duplicates were found.

Estimated Execution Time - 1 Minute

3.23.5 - Test Case 5: Collects an outdated project into the database.

Inputs

- Text input: "select Titles from Projects where materials NOT IN Banned_Materials and materials IN Banned_Materials"

Test Execution Steps

- Super user connects to the database
- Super user enters "select Titles from Archive where materials NOT IN Banned_Materials and materials IN Banned_Materials"

Expected Outcome for Validation (all validations must occur)

- A blank table should be displayed to show that there are no projects with banned materials in the public view.

Estimated Execution Time - 1 Minute

3.23.6 - Timeline

Total Estimated Execution & Validation Time - 5 Minutes

Collecting and displaying projects from an external source is a core component of the project as users will need to have a handful of tangible projects to confirm our purpose for this website.

Since this is a key feature, it will need to be fully implemented and tested earlier on during **Sprint 6: (2/4/22 - 2/18/22)** where setting up and modifying the functionality of projects will occur.

3.24 - Archiving

Verification

Visually confirm projects containing outdated materials are no longer visible on the website anywhere.

Priority

Low - This is meant to be a record of previous projects that were once on the site. This is a low priority because we currently do not have a need for the outdated projects. In the future, they could serve a different purpose such as being useful for data analysis. They can also be recycled and updated by a staff member, while giving credit to the original creator.

Integration Testing

Outdated projects will be archived into a private folder in the database.

Pre-Conditions

- Network needs to have traffic in order to receive data to archive

All of the following test cases must pass in order for the feature to be considered validated successfully

3.24.1 - Test Case 1: Outdated project is archived.

Inputs

- Text input: "select Titles from Archive where materials IN Banned_Materials"

Test Execution Steps

- Super user connects to the database
- Super user enters "select Titles from Archive where materials IN Banned_Materials"

Expected Outcome for Validation (all validations must occur)

- A table should be displayed with all of the outdated projects that were once in the public view.

Estimated Execution Time - 1 Minute

3.24.2 - Test Case 2: Non outdated projects are archived.

Inputs

- Text input: "select Titles from Archive where materials NOT IN Banned_Materials and materials IN Banned_Materials"

Test Execution Steps

- Super user connects to the database
- Super user enters "select Titles from Archive where materials NOT IN Banned_Materials and materials IN Banned_Materials"

Expected Outcome for Validation (all validations must occur)

- A blank table should be displayed to show that there are no projects with banned materials in the public view.

Estimated Execution Time - 1 Minute

3.24.2 - Timeline

Total Estimated Execution & Validation Time - 2 Minutes

Archiving will have to take place during the testing of notifications, filters, and controls. Anything that needs to be archived will have to come before this component so that each piece can be thoroughly tested through the archive. This will happen during **Sprint 7: (2/19/22 - 3/5/22)** so that archiving can be used later on during the implementation of other features and functions.

3.25 - Logging

Verification

Visually confirm error logs are within the database.

High - Logging errors is extremely important because we need to have a record of previous failure points within the website in order to correct them if they arise again.

Integration Testing

In order to validate that logging works as intended, several test cases will be used to confirm that the errors were saved.

Pre-Conditions

- Network needs traffic (i.e. login attempts) in order to log data

All of the following test cases must pass in order for the feature to be considered validated successfully

3.25.1 - Test Case 1: An error is logged.

Inputs

- Text input: "Error: method "search" is undefined"

Test Execution Steps

- Super user connects to the database through the terminal
- Super user enters "cd Logs"
- Super user enters "cat errors.txt"

Expected Outcome for Validation (all validations must occur)

- The terminal will display "Error: method "search" is undefined" within the text file.

Estimated Execution Time - 1 Minute

3.25.2 - Test Case 2: An error is not logged.

Inputs

- Text input: "Error: method "search" is undefined"

Test Execution Steps

- Super user connects to the database through the terminal
- Super user enters "cd Logs"
- Super user enters "cat errors.txt"

Expected Outcome for Validation (all validations must occur)

- The terminal does not display "Error: method "search" is undefined" within the text file.

Estimated Execution Time - 1 Minute

3.25.3 - Timeline

Total Estimated Execution & Validation Time - 2 Minutes

Error logging is important and should be done as early as possible. Logging will be completed by the end of **Sprint 6: (2/4/22 - 2/18/22)**.

3.26 - Data Store

Verification

Data storage can be seen throughout the website, as long as there are projects populated the search and discovery page, then the storage is functioning as intended.

Priority

High - This is a core component to the website and without it, nothing can be saved. If users cannot have their work saved onto the website then there would be no point in using it as they would feel like they are wasting their time.

Integration Testing

There can be many tests, but these will be focused on what the average user can view.

Pre-Conditions

- Data must be available for the data store to cache data

All of the following test cases must pass in order for the feature to be considered validated successfully

3.26.1 - Test Case 1: Projects are stored.

Inputs

- Data input: Projects from database

Test Execution Steps

- User goes to search and discovery page

Expected Outcome for Validation (all validations must occur)

- The user is able to view projects within the page.

Estimated Execution Time - 1 Minute

3.26.2 - Test Case 2: Project ratings are stored.

Inputs

- Data input: Project ratings from database

Test Execution Steps

- User goes to search and discovery page
- User selects and interacts with one of the projects within the page
- User is redirected to the project page

Expected Outcome for Validation (all validations must occur)

- User is able to view the project rating within the project page

Estimated Execution Time - 1 Minute

3.26.3 - Test Case 3: Comments are stored.

Inputs

- Data input: Project ratings from database

Test Execution Steps

- User goes to search and discovery page
- User selects and interacts with one of the projects within the page
- User is redirected to the project page

Expected Outcome for Validation (all validations must occur)

- The user is able to view the comments at the bottom of the project page

Estimated Execution Time - 1 Minute

3.26.4 - Test Case 4: Comment ratings are stored.

Inputs

- Data input: Project ratings from database

Test Execution Steps

- User goes to search and discovery page
- User selects and interacts with one of the projects within the page
- User is redirected to the project page
- User navigates to comment section

Expected Outcome for Validation (all validations must occur)

- The user is able to view the comment ratings next to each comment. *Estimated*

Execution Time - 1 Minute

3.26.5 - Timeline

Total Estimated Execution & Validation Time - 4 Minutes

Data storage allows for the more involved features to work, so it must be completed before any features that involve storage begin development. Data storage will be complete by the end of **Sprint 6: (2/4/22 - 2/18/22)**.

3.27 - User Management

Verification

User management can be verified through account creation, deletion, updating user account, and enabling/disabling user account. A system administrator should be able to control these metrics both at a user level and bulk level and it should be shown through logging and archiving data.

Priority

High - This is a core component to the website and without it, the website will have no way of configuring users. The significance of having this configuration is for security and updating purposes. To enforce security within the website, administrators will have the ability to disable or delete an account if a situation ever called for it. If users could not be configured, no type of restrictions or deterrence would be in place to keep users from following website guidelines.

Integration Testing

Several user configurations will be performed and tested to see if this core requirement is working.

Pre-Conditions

- An authenticated user must have an active session in process
- User that is doing the user configurations must be on the UM configurations view
- User must be a system administrator with special privileges

All of the following test cases must pass in order for the feature to be considered validated successfully

3.26.1 - Test Case 1: User performs a single user management action

Inputs

- From user management page settings, the user chooses an option from a list of actions
- User chooses "single UM operation"

Test Execution Steps

- Page is reloaded to indicate request has gone through

Expected Outcome for Validation (all validations must occur)

- User receives a message from the website saying, "UM operation was successful" within 5 seconds of action

Estimated Execution Time - 1 Minute

3.26.2 - Test Case 2: User performs a bulk (<10k) user management action

Inputs

- From user management page settings, the user chooses an option from a list of actions
- User chooses "bulk UM operation"
- User inputs the amount

Test Execution Steps

- Page is reloaded to indicate request has gone through

Expected Outcome for Validation (all validations must occur)

- User receives a message from the website saying, "Bulk UM operation was successful" within 60 seconds after the action has gone through.

Estimated Execution Time - 2 Minute

3.26.3 - Test Case 3: User performs a bulk (10k) user management action

Inputs

- From user management page settings, the user chooses an option from a list of actions
- User chooses "bulk UM operation"
- User inputs the amount

Test Execution Steps

- Page is reloaded to indicate request has gone through

Expected Outcome for Validation (all validations must occur)

- User receives a message from the website saying, "Bulk UM operation was successful" within 60 seconds after the action has gone through.

Estimated Execution Time - 2 Minute

3.26.4 - Test Case 4: Single user management action takes longer than 5 seconds

Inputs

- From user management page settings, the user chooses an option from a list of actions
- User chooses "single UM operation"

Test Execution Steps

- Page is reloaded to indicate request has gone through

Expected Outcome for Validation (all validations must occur)

- User receives an error message from the website saying, "UM operation was unsuccessful -
    - Action took longer than 5 seconds
    - No error is displayed to the user
    - Most recent data is not captured by data store

Estimated Execution Time - 1 Minute

3.26.5 - Test Case 5: Bulk user management action takes longer than 60 seconds

Inputs

- From user management page settings, the user chooses an option from a list of actions
- User chooses "bulk UM operation"

Test Execution Steps

- Page is reloaded to indicate request has gone through

Expected Outcome for Validation (all validations must occur)

- User receives an error message from the website saying, "UM operation was unsuccessful -
    - Action took longer than 5 seconds
    - No error is displayed to the user
    - Most recent data is not captured by data store

Estimated Execution Time - 2 Minute

3.26.5 - Test Case 5: Bulk (10k) user management action takes longer than 60 seconds

Inputs

- From user management page settings, the user chooses an option from a list of actions
- User chooses "bulk UM operation"
- Enter amount of UM operations

Test Execution Steps

- Page is reloaded to indicate request has gone through

Expected Outcome for Validation (all validations must occur)

- User receives an error message from the website saying, "UM operation was unsuccessful -
    - Action took longer than 5 seconds
    - No error is displayed to the user
    - Most recent data is not captured by data store

Estimated Execution Time - 2 Minute

<u>*Estimated Total Test Plan Execution*</u> - 151 Minutes, 10 seconds