

Feature Name: Project Showcase
Developer: Garret Tsumaki
Date Developer Submitted: 4/16/2023
Reviewer: Kevin Dinh
Date Review Completed: 4/16/2023

Major Positives and Negatives:

Top Qualities of the design:

- The wire frame of the feature is clear and concise. The document clearly establishes how the view should look, what AJAX calls need to be done, and how each case is handled as a failure or success.
- The design document goes into great lengths to establish what is to be expected from the feature. Examples of this can be seen in what each type of user is able to do within the feature, restrictions based on files or text, and the kinds of AJAX calls to be expected on pages 15-17.

Top Flaws of the design:

- Sequence diagrams for Create Project Showcase Draft, Edit Drafted Project Show, and Remove Project Showcase Draft are duplicates of Like Project Showcase. There are no signs of how these components of the feature should run from front end to back end. Only the wire frame can be used as a reference to see how these components may behave.
- Organization of design documents is questionable. Although the information given is informative and allows for an abundance of information to be gathered, the overall structure of how the document is laid out may cause confusion.
- Content surrounding High Level Design does not seem relevant to its section. Specifications on Permissions of different types of users, types of restrictions, and Showcase ID assignment, may be better included in a different section relating to Business Rules or a summary of them. Endpoint format appears too granular to belong in the High Level Design Section and may serve better in a Low Level Design Section.
- There are no High Level Diagrams within the High Level Design section. Without any diagrams showing use cases in a less granular level, and with the reason above, there is no real reason for there to be a High Level section.

Unmet Requirements:

- Edit and Delete **Published** Showcase View is not accounted for. If it is meant to be overlapping with Create_Edit_Publish_DeleteShowcase, it is unclear.
- Missing Sequence diagrams for Create Project Showcase Draft, Edit Drafted Project Show, Remove Project Showcase Draft due to duplication

Design Document Recommendations:

- Subsections for Permissions, DefaultUser, VerifiedUser, and Admin user might serve better outside of High Level Design
- File Restrictions, text restrictions, and showcase ID assignment should fall under Business rules or a summary of business rules as it focuses on the requirements needed to be fulfilled related to those aspects of the feature
- Endpoint Format is better placed within Low Level Design due to its granularity and specificity.
- Remove High Level Design Section unless content appropriate for it was intended.
- Low Level Design
 - There are no data types to define arguments in functions
 - Expected return data is summarized. For consistency across all other documents, payload should be specified to reflect what the intended result should be.
 - Separation of success and failure cases are recommended. Most failures are summarized within the sequence diagram and do not show possible points of failure. For consistency across all other documents, the inclusion of success and failure cases may improve clarity of expectations of readers across all features.
- Preview Drafted Project Showcase
 - Misspelling on Showcase Preview View
- Admin interactions
 - Publish Draft Project Showcase and Remove Published Project Showcase
 - Not entirely sure why authentication of Admin User needs to occur for publishing Drafted Project Showcase. Is not listed in requirements involvement of outside entity reviewing showcase.
 - I understand why admin might need to be involved with removal of publish project showcase, do not understand involvement with publish draft
 - Publish user drafts might be invasive
 - Post, Edit, and Remove Project Showcase Comment
 - Not entirely sure why admin needs to be checked during post or edit project showcase comment
 - For removing, may be related to if a comment is reported for being inappropriate.
- Edit Published Project Showcase
 - Need to account for saving changes
 - "Project showcase is viewable by all users

- “Project showcase reflects edited fields”
- Reorder Attached files in Project Showcase
 - Is the reorder intended to be permanent changes? If so, it will require some way to save the new order. If not, the use case does not seem useful enough to keep.
- Vote Project Showcase Comment
 - Separating the three scenarios into their own diagrams may provide better clarity of each scenario.
 - Could have comment, when created, have a rating of zero, thus eliminating the case of having a null payload. This can simplify the complexity of this specific use case.
- Is there any way to see the reports either from the reported user in question or by administration?

Test Recommendations:

Business rule: A project showcase must include 1 word title, 50 word description, and 1 file upload for minimal publishing requirements

- Test to see if the Project Showcase Draft completes when one and/or multiple of these conditions are unfulfilled.
 - If an expected failed Project Showcase successfully publishes, it creates problems of quality for users in question.
 - This should be an E2E test since the majority of the interactions are done in the front end, but can be done in the back end as well.

Publish/Delete Draft Listing as Admin User

- Test to see if a Project Showcase Draft can be published by an Admin User that is not the owner of the draft in question.
 - If an outside source other than the author of the draft is able to publish or delete them without the author's permission, it can potentially be used for malicious intent. It should also be out of the hands of anyone but the author to make these actions.

Edit Project Showcase Comment as Admin User

- Test to see if a comment made by another user apart from an Admin user is able to edit and save edits.
 - Similarly to the test case above, this can potentially be invasive to a user's privacy and their usage of Hubba as a service. This can also be abused to make certain showcases appear better than they are and vice versa

Backend validation for Get{Feature} methods

- Test your validation function using incorrect DTO objects with DB.null inputs.
 - This is low priority since it is only checking that the functions created by fellow developers on the team work properly. In an ideal scenario there would be no issue.