Discussion Board Requirements and Design Proposal

3/22/2023

Enrique Munoz

Objective

The wiki does not contain interaction among community members at the moment, but a discussion board implementation will allow users to interact with one another on a specific page of the wiki. Since the site is about music artists and groups, individuals could share their opinion about these groups, share their favorite songs, or recommend similar types of artists to other people.

Use Case

UC1. User writes a discussion post

Steps:	1. User enters a specific page 2. User enters the discussion page of the previous page 3. User uploads their comment 4. Post is saved specific page section of GCS Bucket
Hazards:	1. User's comment uses vulgar language 2. Post is stored in incorrect GCS Bucket 3. Limit of discussion posts to prevent system from overflow

UC2. User deletes their own discussion post

Steps:	1. User enters specific page 2. User enters discussion board section of that page 3. User searches discussion board for their post 4. User deletes their post 5. Post is deleted from the bucket
Hazards:	1. User can delete any other user's posts 2. Comment is not deleted from corresponding bucket

Requirements

R1. Backend checks post before posting

For UC/R:	UC1
Type:	Non-functional, interface
Rationale:	Individuals may post vulgar posts that don't pertain to the content's material.
Criterion:	Posts with offensive language are not allowed to be posted, tested by unit test
Priority:	1
Effort:	4h

R2. Specific posts can only be deleted by the original poster

For UC/R:	UC2
Type:	Functional
Rationale:	Users should only be allowed to delete their own posts.
Criterion:	Discussion post is deleted from the bucket and removed from the page, verified by Integration Test
Priority:	1
Effort:	6h

R3. Limit discussion posts on specific page from individual user

For UC/R:	UC1
Type:	Non-functional, Operating
Rationale:	Users may abuse the discussion posts and post many times without any substance to their post, increasing the database's capacity.
Criterion:	Users can only post up to 100 discussion posts on specific pages, tested using Integration Tests
Priority:	1
Effort:	4h

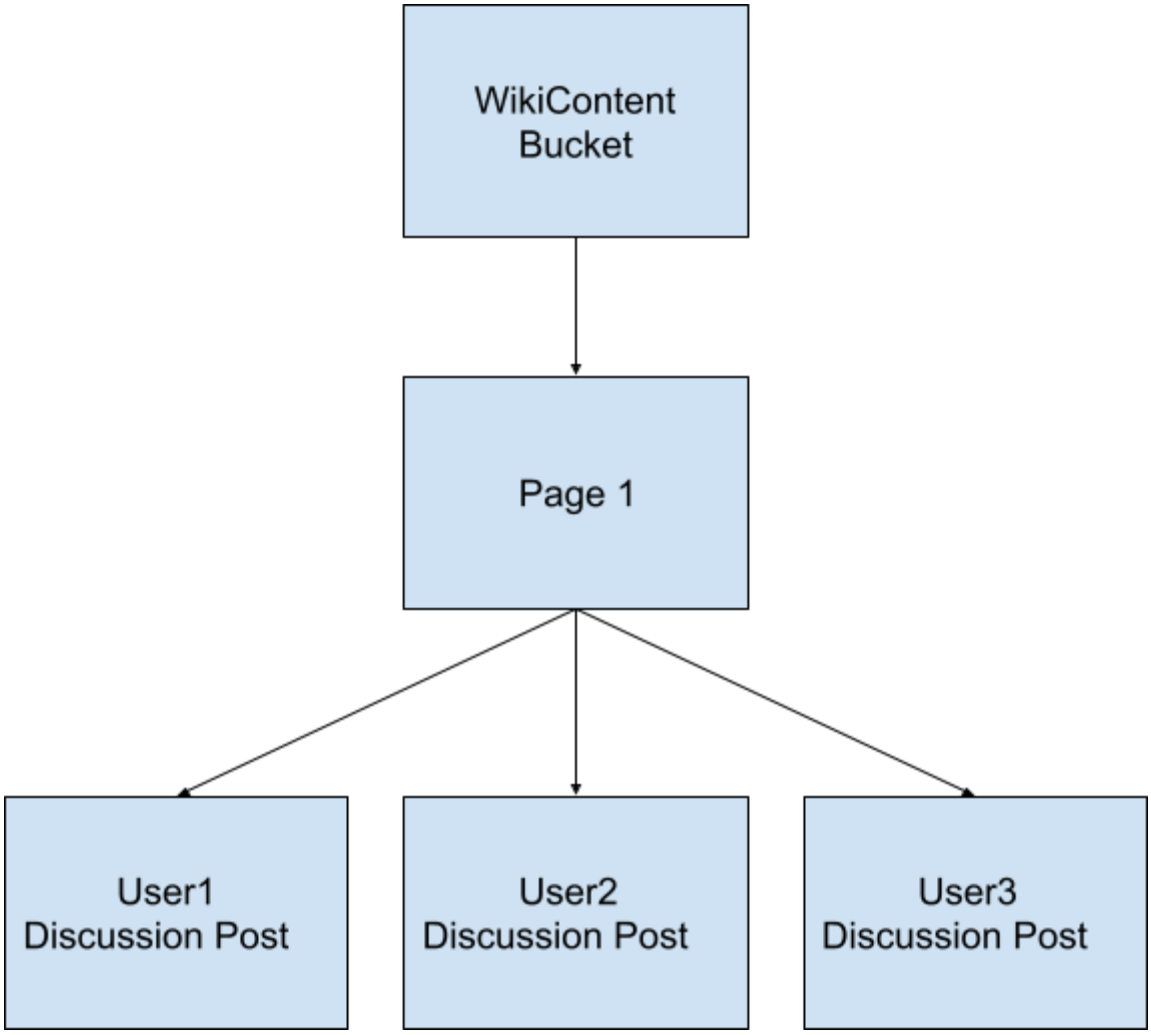
R4. Have a character limit on discussion post

For UC/R:	UC1
Type:	Non-functional, Operating
Rationale:	Users may commit nefarious acts by typing a bunch of characters which would slow the system down.
Criterion:	Users can only post discussion posts with 150 characters max, tested using unit tests.
Priority:	1
Effort:	4h

Detailed Design

This design shows the implementation of the discussion post in the GCS bucket. We would have to access the wiki-content-bucket and have access to a specific page. That page should include a folder that contains all of the possible posts that users have uploaded. If a user deletes a post, the backend should be able to look for the specific post in that bucket and remove it from the wiki-content-bucket page's folder. Meaning that the post would not be included in the database anymore therefore not being displayed on the website.

Figure 1: Discussion Folder Inside of Each Page



R2

This design shows the implementation of a process diagram in the user interaction in the discussion posts. The individual has to view the page in order to see the discussion posts that the specific page contains. After they are allowed the option of uploading or deleting a post. If they choose to delete a post, they must have a post uploaded which would remove it from the wiki-content-bucket. If the user has no posts to delete, they should be returned back to the discussion posts section. The other option is to upload a discussion post. This part has no requirements and instead uploads the post to the specified page in the wiki-content-bucket.

Figure 2: Process Diagram

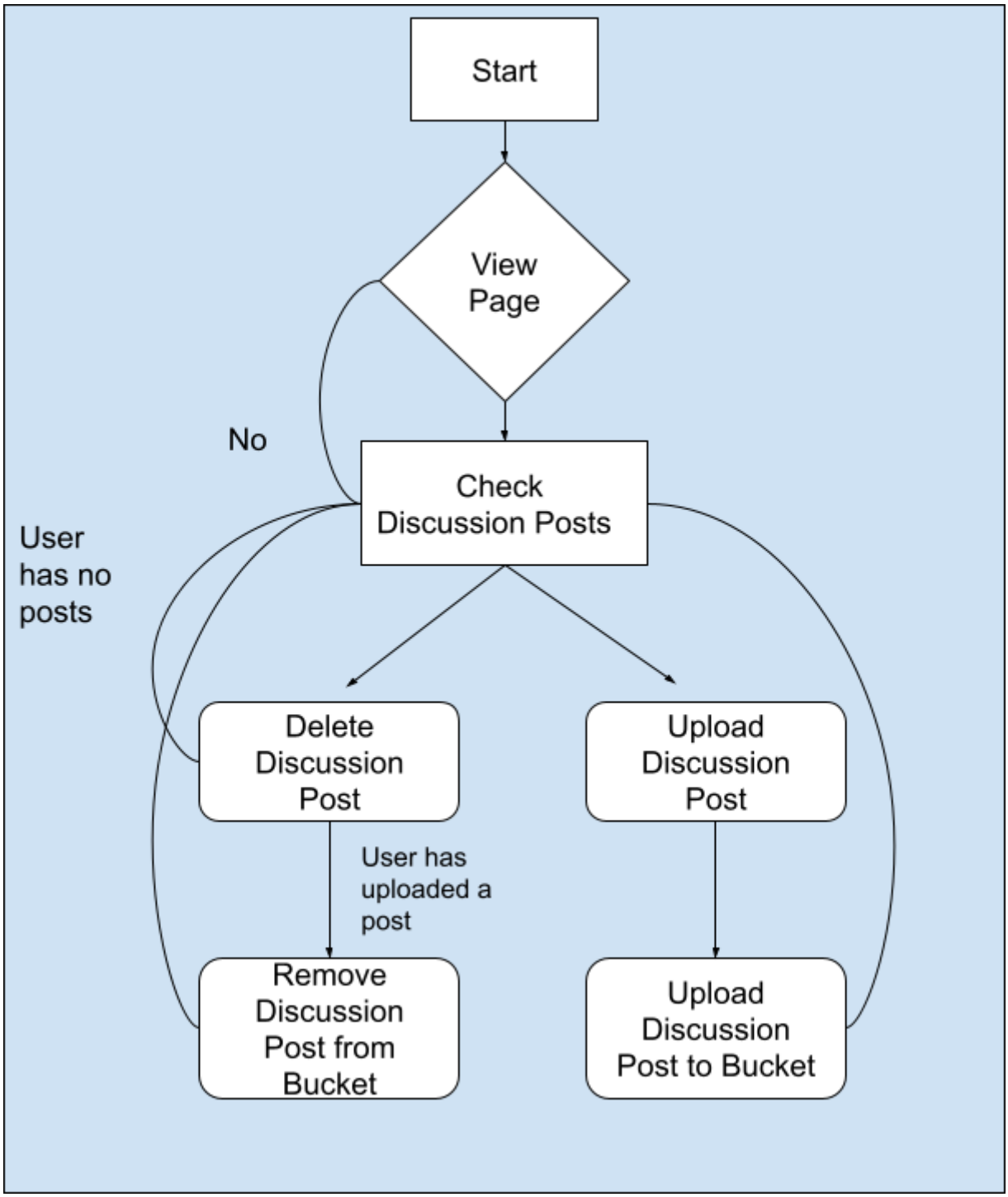
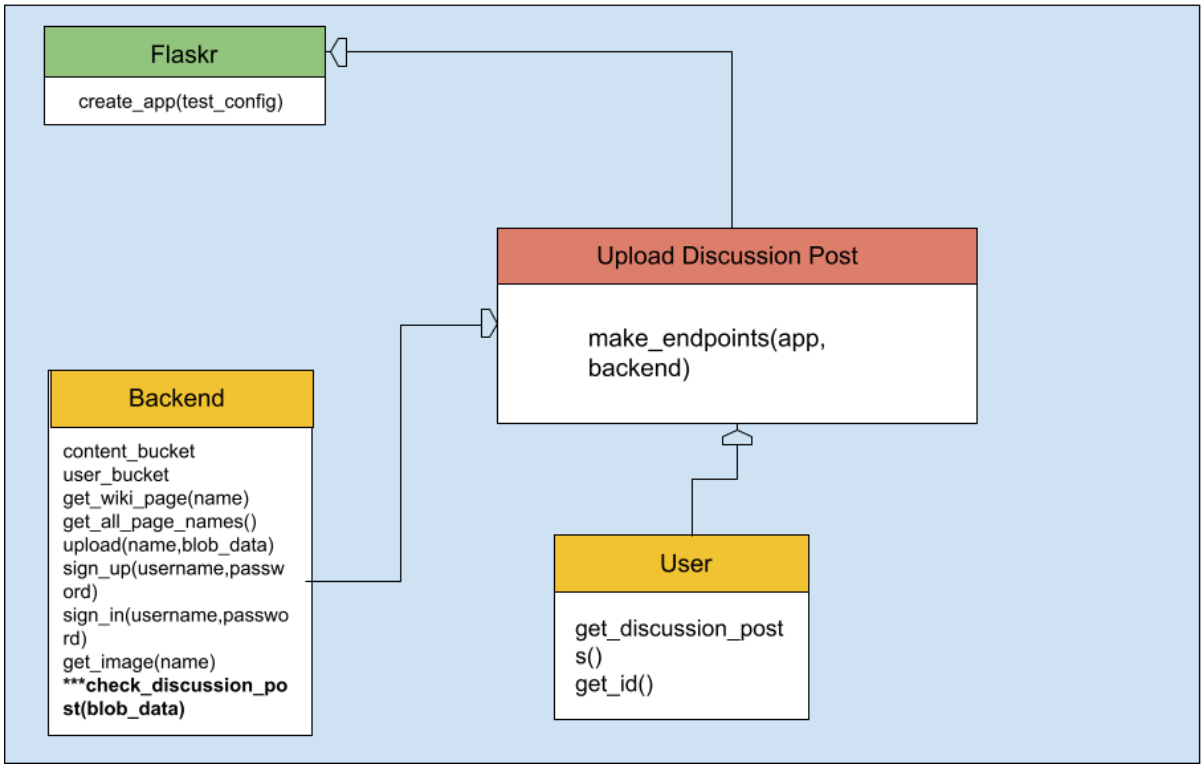


Figure 3: Class Diagram

R1, R3, R4

This design shows the implementation of the discussion post in terms of a class diagram, seeing how the code would interact with the idea of uploading a discussion post. In the backend, I would include a new method called check_discussion_post, which could take in account the character limit, profanity filter, and check the user uploading to see how much discussion posts the user had already uploaded in that current page. Meaning that we could check all of these conditions under one function.



Alternatives

1. Create a Separate Bucket for Discussion Posts

Instead of keeping the discussion post in the wiki-content bucket, we could implement it in its own bucket and characterize it by the specific page it was posted on.

- **Benefit:** Easier to implement than having to set up a dictionary-type structure where the page would be the key and map to the discussion posts uploaded.

- **Blocker:** The only problem would be if we are required to only have two buckets in the project and don't have the ability to create new buckets.
- **Cost:** We would have to create another bucket to handle the uploaded discussion posts

2. Create a discussion post section

Instead of showing the discussion posts at the bottom of the page, we would instead implement it in another section which would be the discussion post section which could be accessed from the page.

- **Benefit:** Easier to implement since we could route to this section instead of having to implement every post in the page.
- **Blocker:** <Blocker if applicable>
- **Cost:** <Cost if applicable>

Rubric

Requirement	**Grade**	**Points**	**Rationale**
Checkpoint		5	
Objective		5	
Use Cases		10	
Requirements		20	
Detailed Design		20	
Technical Diagram		10	
Alternatives		10	
Can this be implemented?		10	
Total		**100**	