Betterflye Content Management Team

Design Day

Betterflye Content Management Team

Sara Bailey

Jake Dibble

Dan Gonzalez

Daniel Spencer

Business Requirements

- BR1
  - We want to prevent users from posting inappropriate content.
  - The client wants the website to be appropriate for users of all ages. Acceptance testing revealed that people were worried about the lack of a content filter. Inappropriate content on the feed would deter future users and lower the reputation of the client.
- BR2
  - We want users to have the option to flag content.
  - No algorithm is perfect, we want users to have the option to manually flag content they find inappropriate. Flagging can be used as input to "train" the service. Flagging serves as a backup filter for content that aren't caught or is unable to be caught by the algorithm.
- BR3
  - We want admins to be able to control user content.
  - We want admins to control content so what is posted on the platform can be reflected in a positive light. This can be an additional backup if the service itself fails. Admins should have the ability to reply to appeals for blocked content to be unblocked.

Business Requirements

- BR4
  - We want admins to be alerted to content blocks.
  - This will keep admins and other staff in touch with the current state of the platform with alerts. It will allow admins to see what type of content is being posted or blocked more frequently. In addition, admins can see which users are causing most blocks, and take further action if needed.
- BR5
  - We want the content service to be able to scale with more users or different platforms.
  - Because the client is a startup, there are few users, but has potential to grow significantly, and the service needs to be able to accomodate this. The platform needs to be kept clean whether the user size is small or large. Content service should be able to transfer to any device or technology platform the client chooses in the future.

Use Cases

Actors

- Users
  - These are the people that use the platform. They are able to post photos and text. They are also able to flag other users' posts if they think it is inappropriate. Users can also appeal their flagged post to an admin so that it can be unblocked.
- Admins
  - These are the people that monitor the platform. They can do everything that a normal user can do and they can also accept or reject user appeals. Admins are also notified of any appeals and also notified of blocked posts that are caught by the filter.
- Filter System
  - This is the content management service that monitors the user's input for any inappropriate words or photos before it is posted. Admins will be notified of any blocked posts and will have the option to revert the block.

- UC1: Block inappropriate content
  - Actor: Filter system
  - meets BR1
  - This is an essential function of the content service. It ensures that the client's requirement to keep the service clean is met. Content should be blocked before it is posted to provide a good user experience.
  - Main flow:
    - The actor accepts user input.
    - The actor analyzes the input.
    - The content is posted. [A1: The content is blocked.]
  - Alternate Flows:
    - A1: The content is blocked.
      a. The admin of the page is notified.
      b. The user can request review of the content.
      c. The content is not posted.

- UC2: Appeal blocked content
  - actor: user
  - meets BR3
  - This function allows the admin to check the blocked content to make sure that it was not wrongly flagged. This allows users to have more control over the content. It allows the platform to overcome errors made the the content filtering system.
  - Main Flow:
    - The actor chooses to appeal their blocked submission.
    - The admin is notified of the appeal.
    - The admin views the content.
    - The admin accepts the content. [A1: The admin rejects the content.]
    - The user is notified and the submission is posted.
  - Alternate Flows:
    - A1: The admin rejects the content.
      a. The content is deleted.
      b. The user is notified.

UC3: Control blocked content

- Actor: Admin
- meets BR3 and BR4
- This allows admins to control what is on their own pages. This can also be used to identify recurring problematic users and can also be used to identify trends among the users and the content being posted on the platform. It is also necessary for admin to see blocked content in order to respond to a user appeal.
- Main Flow:
    - The actor views flagged or appealed posts.
    - The actor accepts the post. [A1: The actor rejects the post.]
    - The accepted material is posted publicly.
- Alternate Flows:
    - A1: The actor rejects the post.
        a. The content is deleted.
        b. The user is notified.

UC4: Flag a post

- Actor: user
- meets BR2
- This is a way to remove content that bypassed the filter. Users can flag content they think is inappropriate, which shall then be reviewed by the admin of the page. The content that is not caught by the filter can be used to update the filter so that it will catch that content in the future.
- Main Flow:
    a. The actor flags another user's post.
    b. The content is removed temporarily from public view.
    c. The admin is notified.
    d. The content is sent to the admin for review.

Requirements

Functional Requirements

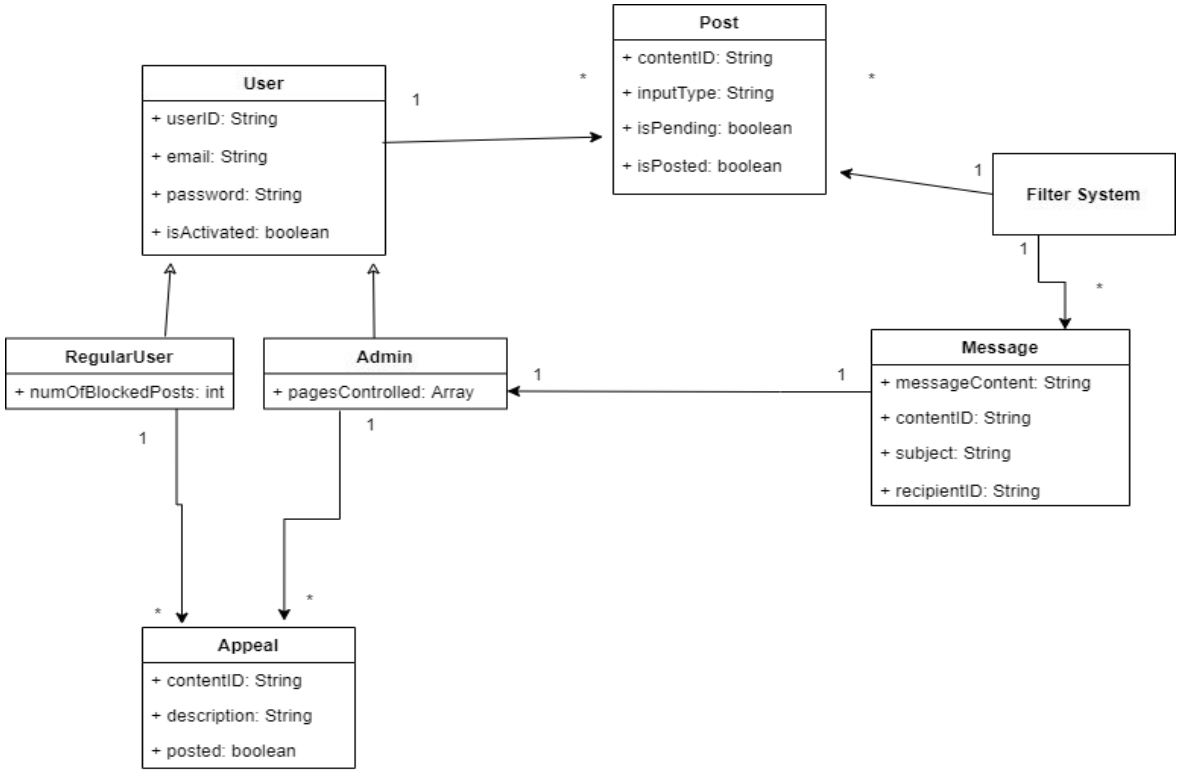
- FR1: If inappropriate content is drafted, the content filter prevents it from being posted.
  - Priority: High
  - BR1
- FR2: If a user's post is blocked, they can make an appeal to an admin.
  - Priority: Medium
  - BR3
- FR3: If inappropriate content bypasses this filter, then users can manually flag the content.
  - Priority: Medium
  - BR2
- FR4: Admins shall receive alerts when content is blocked.
  - Priority: Low
  - BR4

- FR5: If content is flagged by users, the content is temporarily removed from the platform until admin review.
  - Priority: Medium
  - BR2
- FR6: The system shall accept user input as text or photo file.
  - Priority: High
  - BR1
- FR7: The system shall evaluate the submitted content before it is posted.
  - Priority: High -BR1
- FR8: The admin shall be able to approve or reject content.
  - Priority: Medium
  - BR3
- FR9: The user shall receive an alert if their upload is blocked by the filter.
  - Priority: Low
  - BR1

Non-Functional Requirements

- NR1: The system shall be able to scale with new users.
  - Priority: High
  - BR5
- NR2: The content filter shall not cause lag for users.
  - Priority: High
  - BR1
- NR3: Records of users who have uploaded blocked content shall be stored.
  - Priority: Low
  - BR3, BR4
- NR4: The software shall be written using PHP.
  - Priority: High
  - BR5

Domain Model



Classes

- User: This is the parent class that supports all users of the Betterflye service. As a user, they can browse the website and upload content.
    - RegularUser: A child class of User. This is an average user with no special permissions.
    - Admin: A child class of User. Admins are users who run their own pages and have special permissions, such as the option to delete other users' posts. They also can approve appeals or flags.
- Appeal: This is created when a RegularUser has submitted content that the Content Filter blocked and requested a manual review by an Admin. The Appeal contains information about the blocked post and the user who submitted it and allows the admin to reject the appeal, permanently deleting the content, or to approve the appeal, allowing the content to be posted publicly.
- Post: This class contains the content that a user has created and wishes to be posted publicly. A Post interacts with the Content System, which decides whether to block or allow the post. If it is allowed, the content of the post is made public.
- Filter System: This class is the main goal of the project. The Filter System will be able to analyze Post content to determine if it contains inappropriate content. If it does, then the system will block the post from becoming public.
- Message: A Message is created when a Filter System blocks a post. This message contains information about the post and the user who created it, and it is sent to an admin.

Tech Stack

- HTML, JavaScript, CSS
  - This is the front end software that our client already uses, we may need this for implementing features
- PHP -> PhpStorm
  - https://www.jetbrains.com/phpstorm/
  - This is the back end software that our client uses and prefers for us to use
- Database -> MySQL
  - https://www.mysql.com/
  - We will use the database they already have set up for storing blocked content data.

Prototype

https://www.youtube.com/embed/wVwHhBjpoac

https://xd.adobe.com/view/dff24b8d-c90c-44ba-80aa-bbebded88cb2-8aa1/

First Iteration Features

Feature 1: If inappropriate content is drafted, the content filter prevents it from being posted.

Feature 2: The system shall accept text input from a user

Feature 3: The system shall evaluate the submitted content before it is posted

Mentor Feedback

- Business requirements, Use Cases, Tech Stack, Prototype, and First Iteration Features are all good
- Requirements: Michael stated that FR5 (flagged content by users) will need to change from automatic removal to an admin message
- Requirements: For NR4, Michael said we could use any tech stack we desire not just PHP (although since the system was built with it, we will continue using this).
- Domain Model: For user entity Michael mentioned we will need to take a look at their database when we get access to accurately name attributes in the model to the table columns.

Client Feedback - Use Cases

- Virginie noted that for Actors, we should have an actor called “Manager” as well. Manager will be an elevated User that manages an organization.
- She also suggested the option to ban users that frequently misuse the platform by posting inappropriate content.
- In addition to this, we discussed how the flagging feature can possibly be used maliciously and how to prevent this.

Client Feedback - Requirements

- Clarified that 'post' should refer to all open-ended input, such as user names, descriptions, comments, etc.
- In other words, ‘post’ simply refers to ‘content.’

Client Feedback - Tech Stack

- For our tech stack, we discussed the possibility of purchasing licenses for the software used like PhPStorm.
- We were also told that we can add to the tech stack and that we don’t necessarily have to use what is already being used.

Client Feedback - First Iteration Features

- We discussed the possibility of using an external API in the system. This API would possibly be used in the implementation of our filter.
- We also discussed adding a whitelist for usernames that may be flagged if they are legitimate.
- For example, if a user’s legal name contains a swear word, they should not be flagged.

Client Feedback Summary

- All of our design documents were approved.
- Business Requirements and Prototype documents were approved without any suggestions.

Tim’s Bakery - Best Donuts In the Indy Area??

- 6087 W Broadway, McCordsville, IN 46055
- Open Mon-Sat 5AM-11AM