

anthologist

Team 36 - Product Backlog

Samuel Deutsch, Adil Bozai, Taimoor Javed, Hadeed Noshab, Sam Chippas

Problem Statement:

Writers often have to utilize multiple web services to effectively collaborate on stories. Anthologist attempts to bundle the features of these services together on one platform. This will enable creative writers to work together in creating stories block by block and allow readers to ultimately choose the outcome of their stories by majority vote.

Background Information:

Terminology:

- Block - A small portion of a Story created by a Writer, and voted upon by Readers to become a permanent part of the Story.
- Story - A new creative work created by an Author consisting of one Block and the criteria for how blocks should be posted and voted upon.
- Reader - Any user who has created an account on Anthologist. Writers, Authors, and Admins are also Readers.
- Writer - Any user that can add blocks to a story created by an author. They can also change aspects of the story, such as the title or tags.
- Author - The original poster of a Story, sets the criteria for posting and voting upon blocks. The author is also a Writer for his own Story.
- Admin - A special type of account capable of performing administrative tasks on Anthologist, such as deleting stories and accounts.

Audience:

In the past, creative writers and readers have had to adapt other more broadly focused platforms to fit their needs. Anthologist will attract creative writers with its focused collaborative features by enabling them to easily create and contribute to stories in the form of blocks. Anthologist will appeal to readers by allowing them to choose the path their favorite stories take by voting on individual blocks.

Similar Platforms:

There exist several platforms that writers and readers have utilized to share stories, such as: Reddit, Google Docs, Tumblr, and Facebook. These platforms all offer story sharing features, however each present them in a different way. Reddit, Facebook, and Tumblr primarily focus on being general purpose social media platforms, where users can create and share content. Similarly, Google Docs concentrates on being a general purpose word processor that users can use to collaborate while writing.

Limitations:

All of these platforms offer a plethora of useful features for creating works and sharing them; however, they fail in that their user scope is too broad. None of these platforms attempt to appeal specifically to creative writers and readers and in many cases may have some features that can complicate the collaborative process. Our goal with Anthologist is to create a site focused on providing a collaborative environment for the producers and consumers of creative writings. We want our platform to have as low of an entry ceiling as possible to ensure that they are able to easily find, read, or write stories that interest them.

Functional Requirements:

User Stories:

1. As a reader, I would like to register for an account on Anthologist.
2. As a reader, I would like to be able to login to my account.
3. As a reader, I would like to view my stories on my profile page.
4. As a reader, I would like to view my comments on my profile page.
5. As a reader, I would like to setup a profile picture.
6. As a reader, I would like to be able to report inappropriate profile pictures.
7. As a reader, I would like to receive notifications.
8. As a reader, I would like to reset my password.
9. As a reader, I would like to view stories.
10. As a reader, I would like to give positive and negative ratings to story blocks.
11. As a reader, I would like to bookmark stories that I am reading.
12. As a reader, I would like to see stories of the day.
13. As a reader, I would like to leave comments on blocks.
14. As a reader, I would like to sort stories by tags.
15. As a reader, I would like to sort stories by rating.
16. As a reader, I would like to sort by completed stories.
17. As a reader, I would like to view writer's ratings.
18. As a reader, I would like to view the most current blocks.

19. As a reader, I would like to be able to flag content as inappropriate or hateful.
20. As a reader, I would like to follow specific writer's contributions.
21. As a writer, I would like to see how many readers have viewed my story.
22. As a writer, I would like to be able to create a new story.
23. As a writer, I would like contribute to existing stories by creating blocks.
24. As a writer, I would like to edit my contributions.
25. As a writer, I would like to see my name above my contribution.
26. As a writer, I would like to see a list of my contributions.
27. As a writer, I would like to have a check-box to propose the ending block for the story.
28. As a writer, I would like to add a block as an end of story block.
29. As a writer, I would like to be able to delete my block during the editing phase
30. As an author, I would like the option to choose an edit a time limit.
31. As an author, I would like to set a minimum number of proposed blocks per voting period.
32. As an author, I would like to set a minimum story length by number of blocks.
33. As an author, I would like to cap the number of words per block.
34. As an author, I would like to mark a story as inappropriate.
35. As an author, I would like to choose which writers may work on my story.
36. As an admin, I would like to remove any stories.
37. As an admin, I would like to remove individual blocks.
38. As an admin, I would like to remove accounts from our site
39. As an admin, I would like to see writer statistics.

Non-Functional Requirements:

Architecture and Performance:

We will separate the project into the frontend and backend components, to divide up the work among our team. This will allow us to independently work on different aspects of the project while minimizing compatibility issues that may arise between them.

The backend will utilize a MySQL database in conjunction with a Express.js web server. Express.js is a versatile Node.js framework that will allow for an easy and lightweight web server implementation. Additionally, the Node.js backend will allow for easy communication between our web server and a MySQL database. The backend will also handle sending out emails for notification updates and email verification.

The front end will be a dynamic webpage created using React and JavaScript. React will enable us to easily create a visually appealing graphical user interface for our web service. Meanwhile, JavaScript will utilize Ajax requests to facilitate all dynamic data transfers between the backend and the front end, allowing users to dynamically fetch stories, post comments, and blocks without having to load a new page each time.

Security:

Security is paramount to ensuring a fair and collaborative environment, and the overall success of Anthologist. While anyone will be able to read stories on Anthologist, only those with an account will be able to vote on blocks and leave comments, as well as create new stories and blocks. To create an account, users will have to provide a valid email address and will have their passwords securely salted and hashed. All requests sent from the frontend will be sanitized and have their scopes verified before being executed. This is to ensure that no malicious intent, such as SQL injections or other attacks, will be able to affect the successful operation of Anthologist. Additionally, no requests will be executed that do not have a valid authentication token, ensuring every request can be associated with a specific user.

Usability:

The frontend graphical user interface should be both intuitive and to the point. The intuitive nature of the interface will allow new users to easily navigate our site and find stories that interest them. At the same time, experienced users will be able to utilize advanced search features to better refine the criteria for stories they are interested in. The to-the-point aspect of the interface will minimize clutter by only showing the user links and tools that are immediately useful to them on that specific portion of the web service. By adopting a strategy of minimalism, we hope to allow writers to compose new stories and blocks, and readers to vote on those blocks and post comments, all from the same interface.

Hosting and Deployment:

Once we are far enough along in development and have a functional front and back end, we can host our database and our website on a private web-server for testing and deployment. We will also likely obtain a domain for our website, so that it is publicly accessible to potential users.