

# StoryConnect

## Design Document

## Table of Contents

<b>Table of Contents.....</b>	<b>1</b>
<b>Executive Summary.....</b>	<b>4</b>
Introduction.....	4
Problem.....	4
Solution.....	4
Methodology.....	4
<b>Background.....</b>	<b>6</b>
Overview.....	6
Users.....	6
Similar Ideas.....	6
Front-End.....	7
Back-End.....	7
Assets and Engines.....	7
Software and Hardware Requirements.....	8
<b>Requirements Analysis.....</b>	<b>9</b>
System Architecture.....	9
Django Backend.....	9
SQL Database (Postgres).....	9
Flutter Front End.....	10
Personnel.....	11
Dallin Burton.....	11
David Harrington.....	11
Tanner VonWald-Villard.....	11
Novella Alvina.....	12
System Features.....	12
Rank 1: Infrastructure and Core Functionality.....	12
Rank 2: Planned Features.....	13
Rank 3: Features Advanced Individual Features.....	14

UI Portion.....	14
ML Portion.....	14
<b>Software Engineering Tools and Techniques.....</b>	<b>15</b>
Development Style.....	15
Tools.....	15
Code Reviews.	15
Feature Branches.....	15
Release Branches.....	16
Documentation.....	16
Team Communication.....	16
Team Meetings (Spring 2023).....	16
<b>Timeline.....</b>	<b>17</b>
Alpha Stage.....	17
Beta Stage.....	17
Production Stage.....	18
<b>Use Cases.....</b>	<b>19</b>
Reader Cases.....	19
Reader Use Case Number 1.....	19
Reader Use Case Number 2.....	19
Reader Use Case Number 3.....	20
Reader Use Case Number 4.....	21
Reader use Case Number 5.....	21
Reader Use Case Number 6.....	22
Reader Use Case Number 7.....	23
Reader Use Case Number 8.....	23
Reader Use Case Number 9.....	24
Reader Use Case Number 10.....	24
Reader Use Case Number 11.....	25
Writer / Editor Cases.....	27
Writer Use Case Number 1.....	27
Writer Use Case Number 2.....	27
Writer Use Case Number 3.....	28
Writer Use Case Number 4.....	29
Writer Use Case Number 5.....	29
Writer Use Case Number 6.....	30
Writer Use Case Number 7.....	31
Writer Use Case Number 8.....	31
Writer Use Case Number 9.....	32

Writer Use Case Number 10.....	33
Writer Use Case Number 11.....	33
Writer Use Case Number 12.....	34
Writer Use Case Number 13.....	34
<b>UI Sketches.....</b>	<b>36</b>
(Figure 1: Browsing Interface Sketch).....	36
(Figure 2: Reading Interface Sketch).....	37
(Figure 3: Annotation Sidebar Sketch).....	37
(Figure 4: Comment Sidebar Sketch).....	38
(Figure 5: My Page Sketch).....	38
(Figure 6: Review Highlighting/Annotating Interface).....	39
(Figure 7: Author Page Sketch).....	40
(Figure 8: Continuity Checker View Sketch).....	41
(Figure 9: Author Chapter Reviews View Sketch).....	41
(Figure 10: Proofread Edit View Sketch).....	42
(Figure 11: Wiki Sample Page View Sketch).....	42
(Figure 12: Character Sheet Broad View Sketch).....	43
(Figure 13: Character Sheet Specific View Sketch).....	43
(Figure 14: Book Writing View Sketch).....	44
(Figure 15: Demographics Overview View Sketch).....	45
(Figure 16: Demographics Engagement View Sketch).....	45
(Figure 17: Demographics Character View Sketch).....	46
(Figure 18: Demographics Reader View Sketch).....	47
(Figure 19: Demographics Feedback View Sketch).....	47
(Figure 20: Book Details View Sketch).....	48
(Figure 21: Book Copyright View Sketch).....	48
<b>Revisions.....</b>	<b>49</b>
Executive Summary.....	49
Background.....	49
Requirement Analysis.....	50
Software Engineering Tools and Techniques.....	50
Timeline.....	50
Use Cases.....	50
UI Sketches.....	50

# **Executive Summary**

## **Introduction**

StoryConnect's primary purpose is to provide a platform for connecting writers, readers, and publishers. We plan to do this by leveraging a range of different technologies to ensure users feel comfortable with any tools they decide to use.

## **Problem**

Current solutions are a disjointed experience. While some platforms cater to providing writers with helpful tools, they cannot help writers launch their works into the limelight. However, the platforms that provide a community element lack writing tools that help elevate the author's work to the next level and have an unappealing interface.

## **Solution**

StoryConnect will solve these issues by acting as a tool for readers and writers without cannibalizing the others' experiences. The plan is for the different user types to interact with and slowly improve one another. We plan on doing this with the goal in mind to help writers hone their craft and for readers to become invested in their books.

## **Methodology**

This platform is designed to connect two types of users: readers and writers. Readers will be able to easily find books they love in the platform's library through our use of suggestion algorithms. Readers will also be empowered to give feedback and comments for writers, creating a community of passionate readers. Publishers, a subtype of a reader, can additionally evaluate books and connect with authors through the platform, providing additional opportunities for writers.

The platform also plans to offer writers an opportunity to publish their books and receive ratings, reviews, and proposed edits from readers. Additionally, writers will have access to a suite of machine learning tools, including a 'Continuity Checker,' 'Road-Blocker,' and data-visualization tools, helping our writers be the best version of

themselves. While this document will discuss specifics of technologies later, we aim to help writers create books that readers will love.

# Background

## Overview

StoryConnect is a platform for enjoying, authoring, and sharing amateur fiction. We provide a host of writing tools that allow authors to map out, organize, and bring to life their stories. Beyond our suite of writing tools, StoryConnect will connect you to a vibrant community of fellow readers, writers, and publishers, with every user contributing to our platform's goal of connecting each other.

## Users

We are a platform that encompasses the entire book workflow. We plan to cater to those who love to get lost in another's world or create their own. By offering an extensive suite of professional writing tools, we encourage anyone to tell their story in the most streamlined way possible. Once a writer begins penning their story, we hope to give them a community of like-minded individuals who can engage with their work and help them along the way. Those same readers can also simply enjoy the new works or dive deeper with our annotation and feedback system. This system will allow readers to mark pages, highlight passages, and enjoy the story their way, allowing their contributions to be visible to the authors, as discussed previously.

## Similar Ideas

Since our goal is to bridge feature gaps, we have taken inspiration from several existing platforms: Wattpad and Chapterly. Wattpad allows readers to browse for books they like and provide feedback through annotated comments throughout the book, subsequently increasing engagement. However, Wattpad only provides limited writing tools, hampering the overall quality of the works produced. In our initial testing, we found that wattpad's user experience left a lot to be desired. Moving on to Chapterly, we encountered a stark contrast in its offerings compared to Wattpad. It offers a richer, more extensive writing toolbox to help create a more organized experience. Chapterly also provides features such as chapter sections, character sheets, roadmaps

organization, and annotated notes on the story that helps writers better keep track of their writing process and progress.

## Front-End

Our goal is to make StoryConnect available on both Mobile and Web platforms. Rather than developing separate front ends for each platform, which could be challenging and time-consuming, we'll use Dart & Flutter to create a single codebase that works across all platforms. Doing so will allow us to simplify the development process and allow Front-End developers to focus on features rather than struggling with platform idiosyncrasies.

## Back-End

Our Back-End will handle various tasks, such as handling REST API requests, managing Machine Learning Functions, and querying the Database. As per our requirements mentioned later, we will be using the Python/Django stacks to accomplish these tasks. Thanks to its wide range of data science libraries, Python is an excellent choice for handling machine learning tasks. Additionally, Django is a user-friendly framework for building web servers. As django provides greater tie-ins with PostgreSQL, we naturally decided to choose it for our DBMS. As for hosting our web application, we've opted to use Amazon Web Services. Lastly, we'll utilize Docker to quickly deploy and test our web server while minimizing installation-based hangups.

## Assets and Engines

Several of our functionalities will incorporate machine learning-based algorithm features, such as utilizing Natural Language Processing techniques, text classification, keyword extraction, and many more, in which we plan on making use of libraries from Python and integrating them into our program we built from scratch. In addition, one of our planned features is Road-UNBlocker which we plan to leverage tools similar to ChatGPT from their open-source API.

## Software and Hardware Requirements

Our application will run seamlessly on various devices, including iOS, Android, and Desktop platforms, via a web browser. Specifically, our app will run on platforms supporting these minimum requirements: Android devices with API 16 (Android 4.1) and iOS devices running iOS 11 and above. For web browsers, we will require users to be using Chrome 84 and above, Firefox 72.0 and above, Safari on El Capitan and above, and Edge 1.2.0 and above. For potentially more up-to-date information, visit:

<https://docs.flutter.dev/development/tools/sdk/release-notes/supported-platforms>

For our developers' toolchains, we will similarly be platform agnostic. Our team will use VS Code as our primary IDE, available on all major OSes, with accompanying installations of Python and Dart. Also abstracting away from any hardware requirements on the server side, we'll be using Docker on AWS. This decision will make it simple to deploy our back-end nearly all cloud service providers, so long as they support Docker itself.

# Requirements Analysis

## System Architecture

The system architecture consists of four main components: a SQL database (Postgres), a machine learning platform, a Django backend, and a Flutter front end. The SQL database will store and manage our data, while the machine learning platform will subsequently use that data (given user's consent) to generate models that can be used by the system. The Django backend will then provide Rest-APIs for our front end's use to interact with those tools, while the Flutter front end provides a user interface for users to interact with the system. The front end will be divided into two main parts: a reading platform and a writing platform. The reading platform will be available on both mobile and web, while the writing platform will be available only on the web.

## Django Backend

The Django backend will be a critical component of the system architecture, facilitating communication between the front end, database, and machine learning platform. To simplify the development and testing process, our backend will contain a containerized instance of Django, Nginx, LetsEncrypt bot, and Gunicorn. These containerized technologies will provide the foundation to serve secure HTTPS connections from our backend. It will also simplify the project by including its own data access layer, helping abstracts away error-prone database operations. To implement 3rd party sign-in options, we plan to use Firebase for user authentication and authorization instead of Django's native authentication platform.

## SQL Database (Postgres)

Our database schema design will be implemented using Django's data abstraction layer, implemented by a PostgreSQL instance, and hosted on AWS's managed database system. PostgreSQL's tools are better integrated with Django's database management system, saving development time. We plan to tailor our DB design to better integrate with our machine learning services. We will allow users also to save books offline on

mobile devices, which will be compatible with Flutter's SQLite integration. Machine Learning Platform

We will use several language model libraries, and APIs, to support our machine-learning features, such as a model capable of checking the continuity of authors' work. The text data will then be processed and tokenized using standard libraries like NLTK. We then plan on utilizing changelog hashes to effectively cache and use our model with new user-generated data again, after gaining writer's explicit consent. This will ensure that our model stays up-to-date in a time-efficient manner. Additionally, we plan to use OpenAI's GPT-API to construct an AI writing assistant that can aid users in generating short snippets of content when they are stuck.

## Flutter Front End

We will divide the front end into two main areas - the readers' and writers' views. Thanks to Flutter's cross-platform capabilities, our web and mobile platforms will share much of the same codebase. In the case of the reader view, it will be almost identical. Having said that, our writer's functionality will be constrained solely to the web version. Flutter will accommodate all these usages, and will simultaneously give us access to a comprehensive ecosystem of open-source resources with a variety of pre-built widgets, libraries, and other abstractions available.

# Personnel

## Dallin Burton

Dallin Burton is a seasoned software developer uniquely qualified to take on the critical role of leading the writing component of this project. With years of experience in the industry, Dallin has honed his skills in various complex projects, with a particular focus on Flutter. His expertise and hands-on experience with Flutter make him the ideal candidate to spearhead the interface's writing portion and contribute to the overall UI design. As an experienced full-stack developer, Dallin will also act as the team's utility player.

## David Harrington

David Harrington brings valuable experience to our project in both frontend and backend web application development, in addition to his work in database design. He has a strong interest in both frontend and database design, which makes him an ideal candidate for overseeing these aspects of the project. David will primarily be responsible for designing the reader's portion of the UI and contributing to the database design.

## Tanner VonWald-Villard

Tanner VonWald-Villiard is skilled in Python programming with considerable experience working with machine-learning libraries. He is particularly interested in natural language processing, and applying language models to solve real-world problems. Tanner's main responsibilities will involve backend engineering using Django, and assisting in implementing data analytics. Time permitting, Tanner also plans to implement various AI features.

## Novella Alvina

Novella Alvina has extensive experience in Python, focusing on machine learning and data mining, including data collection and cleaning. Her interest in working with data also extends to database management systems, making her a valuable asset for the database-related aspects of the project. Novella will be responsible for backend maintenance using Django, and for constructing user data analytics and maintaining the offline database. Additionally, she will assist in building a user goal-tracking system.

## System Features

### Rank 1: Infrastructure and Core Functionality

1. Database Architecture: The design of our database schema to compartmentalize data representation.
2. Client-Server Model Hosting - The use of Django and Flutter to separate computation appropriately to a device's capabilities.
3. Unit Test Suite: A set of comprehensive and thorough tests that will run on each version of our software.
4. Continuous Integration software: A CDI (Continuous Delivery/Integration) platform that will automate the building, testing, and deployment of our software via Jenkins.
5. Navigating between pages: Connecting all the main UI compliments together via a simple-to-use UI.
6. Writing Stories: Writing page for the user to create a book with all of the provided organization tools.
7. Reading stories: Users will be able to read published stories using the "Reading" interface.
8. Data Importing / Scrapping: Importing of public domain book data for test and demo purposes.

## Rank 2: Planned Features

1. User Registration: Users will be able to sign up using their emails to register themselves in the system.
2. User Account Management Page: Users will be able to change things about their profile, including if and how their author profile is displayed.
3. Book Browsing: Users can browse books by genre, tag and filters that will query the database.
4. Book Details: A page that will display various book details.
5. Highlighting / Annotation of books: Users will be able to highlight and annotate text, categorizing them and saving them for a later viewing.
6. Displaying Feedback in writing interface: A view for writers to see feedback regarding specific portions of their story.
7. Content Tagging / Flagging: Using machine learning, we will label and tag relevant attributes to the author's works.
8. Writing Data Analytics: We will use data-analysis to visually represent statistics revolving around predefined metrics.
9. Author Profile Page: A profile page that will contain descriptive information about an author for users to read including but not limited to: information about their books, a personal description, and an announcements page.
10. Story Navigation: Chapter navigation containing features associated with the relevant text.
11. Reader UI Customization: Readers will be able to switch themes (dark mode) and enable focus view which shows only the current page.
12. Story Covers: Authors will be able to create book covers for their work.
13. Copyright Selection Page: Users will be able to see copyright information for the book they are reading, as well as choose the copyright they wish to employ for their own books.

## Rank 3: Features Advanced Individual Features

### UI Portion

1. Continuity Checker (Dallin): This feature will display the results of using machine learning to scrape information from books and display whether or not there could be a potential conflict in writing.
2. Road Unblocker (Dallin and Tanner): This feature will use the GPT-API to suggest how to continue the story.
3. Character Sheets (Dallin): This feature will allow users to track their characters and attributes.
4. Broad Reviews (David): This feature will provide a place for a reader to leave reviews for chapters.
5. Community Engagement (David): This feature will support authors in interacting with their readers via formal data gathering techniques.
6. Offline Reading (David and Novella): This feature will allow users to download and read books offline.
7. Goal Tracking (Tanner and Novella): This feature will allow users to set and track writing goals, such as the number of words written per day, week, or month.

### ML Portion

8. Continuity Checker (Tanner): This feature will use machine learning to scrape information from books and display whether or not there could be a potential conflict in writing.
9. Character Sheets (Novella): This feature will use machine learning to allow users to gather data to visually track their characters and attributes.

# Software Engineering Tools and Techniques

## Development Style

Our team decided that Agile was best suited to the fast-paced environment of our class. This way, when we need to respond to feedback from the professor, or judges when demoing our prototype, we can implement suggestions more easily.

## Tools

IDE	Visual Studio Code
Main Libraries	Django Django REST-Framework Flutter Beamer (Flutter)
API Development Tool	Postman
Testing Plans and Tools	Flutter Unit Tests Python UnitTest
Bug Tracking	Gitlab
Versioning	Git Atlassian Sourcetree
Issue Tracking	Gitlab

## Code Reviews

### Feature Branches

We will be performing code reviews with our “Secondary Partner” every time before we push to our main branch from a feature branch.

## Release Branches

We will be performing code reviews with our team every time before we push a release branch to the main branch.

## Documentation

We will record all global information in our wiki pages on Gitlab and local updates on recent comments.

## Team Communication

We will communicate over Discord for all remote meetings. That is also where we will update each other and schedule future meetings. We will continue to meet in person for a large amount of our work and resort to remote meetings when convenient or necessary.

## Team Meetings (Spring 2023)

Day	Meeting Type	Location	Start Time	End Time
Monday	Team+Prof	In Person	10:45 AM	11:15 AM
Monday	Team	In Person	11:15 AM	11:45 AM
Wednesday	Team	In Person	10:00 AM	11:30 AM
Thursday	Team	Remote	7:00 PM	7:30 PM
Friday	Team	Remote	10:00 AM	11:30 AM

# Timeline

## Alpha Stage

	Dallin	David	Novella	Tanner
Week 1	TextEdit underflow	Users/Roles (Front End)	Make Unit Test Suites, Fix Data Importing	Fix CI/CD pipeline
Week 2	User Auth (Front end) / Profile Page (Author)	Book Creation Page	Data Query for Each Page	User Authentication
Week 3	Comment's UI	Book Covers (Front End)	Readers Book Browser	Firebase for book covers
Week 4	Comment's UI (Part 2)	Writer View UI Cleanup	Feedback testing	Data Scraping, Testing,

## Beta Stage

	Dallin	David	Novella	Tanner
Week 5	Major Writing View cleanup	Reader's Home View + Reader's Library U	Content Tagging	Highlighting and Annotation
Week 6	RoadUnblocker UI	Reader Chapter Feedback UI + Analytics UI	Content Tagging, Data Analytics	Comments and User Interaction
Week 7	RoadUnblocker UI (Part 2)	Analytics UI	Data Analytics	Reader backend functionality
Week 8	Character Sheet UI	Analytics UI part 2	Offline Database	Begin Machine Learning Tools

## Production Stage

	Dallin	David	Novella	Tanner
Week 9	Character Sheet (Part 2)	Reader's Annotation UI	Offline Database, Automatic Character Sheets	Continuity Checker
Week 10	Continuity Checker UI	Reader's Annotation UI (part 2) + Author Engagement UI	Automatic Character Sheets, Content Suggestion	Continuity Checker (contd) / Content Suggestion
Week 11	Continuity Checker UI (Part 2)	Offline Reading + Reader's Dark Mode	Goal Tracking	Content Suggestion
Week 12	Polishing and Testing	Offline Reading (part 2) + Polish and Test UIs	Polishing and Testing	Polishing and Testing

# Use Cases

## Reader Cases

Number	Reader Use Case Number 1
Title	<b>Browsing Interface</b>
Preparer	Tanner VonWald-Villiard
Actor/User	The Reader
User Story	As a reader, I want to find a new book to read that fits my specific tastes
Course of Events	<ol style="list-style-type: none"><li>1. Click the “Reading” tab</li><li>2. Select your chosen genre from the “Genre” tab on the right of the screen.</li><li>3. Use the “Filters” sidebar to further refine the selection</li><li>4. Browse the given selection of books by tags</li></ol>
Exceptions/Alternatives	- No Books were found for a specified search.
Related UI	Browsing Interface Sketch (Figure 1)

Number	Reader Use Case Number 2
Title	<b>Highlighting Text</b>
Preparer	Tanner VonWald-Villiard
Actor/User	The Reader
User Story	As a reader, I want to have the option to highlight passages with various colors, allowing me to organize my notes visually.

<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Select either a book you're currently reading from the "Library" banner under the "My Page" tab or a new book from the "Reading" tab</li> <li>2. Select the "Annotation" tab within the left sidebar</li> <li>3. Click "Add Highlighter" under the "Highlighters" banner in the "Annotation" tab</li> <li>4. Choose a highlighter color and name the highlighter (eg. quotes, sad, beautiful, etc.)</li> <li>5. For each line, select the desired highlighter and then click and drag over the desired text.</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- Max number of highlighter colors used</li> </ul>
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Reading Interface Sketch (Figure 2)</li> <li>2. Annotation Sidebar Sketch (Figure 3)</li> </ol>

<b>Number</b>	<b>Reader Use Case Number 3</b>
<b>Title</b>	<b>Leaving Notes</b>
<b>Preparer</b>	Tanner VonWald-Villiard
<b>Actor/User</b>	The Reader
<b>User Story</b>	As a reader, I enjoy marking pages and leaving notes on them to log my thoughts and feelings as I read.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Select either a book you're currently reading from the "Library" banner under the "My Page" tab or a new book from the "Reading" tab</li> <li>2. Select the "Annotation" tab within the left sidebar</li> <li>3. Select "Leave Note" under the "Annotation" tab</li> <li>4. Enter the note into the popup textbox</li> <li>5. Click "Done"</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- None</li> </ul>
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Reading Interface Sketch (Figure 2)</li> <li>2. Annotation Sidebar Sketch (Figure 3)</li> </ol>

<b>Number</b>	<b>Reader Use Case Number 4</b>
<b>Title</b>	<b>Viewing and Leaving Comments</b>
<b>Preparer</b>	Tanner VonWald-Villiard
<b>Actor/User</b>	Reader
<b>User Story</b>	As a reader interested in the platform's community, I want to view the comments on a particular page of my favorite book and leave one of my own.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Select the book you're currently reading from the "Library" banner under the "My Page" tab</li> <li>2. Select the "Comments" tab within the left sidebar. The sidebar will now display a list of comments for the user to read</li> <li>3. Select "Leave Comment" at the bottom of the left sidebar</li> <li>4. Enter the comment in the text popup and press "Done"</li> </ol>
<b>Exceptions/Alternatives</b>	- None
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Reading Interface Sketch (Figure 2)</li> <li>2. Comment Sidebar Sketch (Figure 4)</li> </ol>

<b>Number</b>	<b>Reader use Case Number 5</b>
<b>Title</b>	<b>Focus Mode</b>
<b>Preparer</b>	Dallin Burton
<b>Actor/User</b>	Reader

<b>User Story</b>	As a reader, I desire an immersive reading experience without distractions that may hinder my focus.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Open a book on the website</li> <li>2. Notice all of the highlights, ability to share, and different comments.</li> <li>3. Toggle a button that allows only the text to be shown, though allows me to still make highlights</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- Using the other features, such as community aspect (Use toggle for turning feature off)</li> </ul>
<b>Related UI</b>	1. My Page Sketch (Figure 5)

<b>Number</b>	<b>Reader Use Case Number 6</b>
<b>Title</b>	<b>Customize Reading Environment</b>
<b>Preparer</b>	Tanner VonWald-Villiard
<b>Actor/User</b>	Reader
<b>User Story</b>	As a reader, I want to customize my reading space to create a comfortable and personalized environment.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Select a book from the website</li> <li>2. Use the options bar to view customization options</li> <li>3. Tweak the options to customize options such as background color, font size, and line spacing</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- None</li> </ul>
<b>Related UI</b>	1. Reading Interface Sketch (Figure 2)

<b>Number</b>	<b>Reader Use Case Number 7</b>
<b>Title</b>	<b>Offline Access</b>
<b>Preparer</b>	Tanner VonWald-Villiard
<b>Actor/User</b>	Reader
<b>User Story</b>	As a reader, I want to access my books offline.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Select the desired book from the platform</li> <li>2. From the book's details page select "Save Offline"</li> <li>3. The book can now be selected from the "My Page" tab and be viewed offline</li> </ol>
<b>Exceptions/Alternatives</b>	- None
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Browsing Interface Sketch (Figure 1)</li> <li>2. Details Page Sketch</li> </ol>

<b>Number</b>	<b>Reader Use Case Number 8</b>
<b>Title</b>	<b>Mobile Viewing</b>
<b>Preparer</b>	Tanner VonWald-Villiard
<b>Actor/User</b>	Reader
<b>User Story</b>	As a reader, I prefer consuming content on my mobile device, as it allows me to read on the go and in various locations.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Access StoryConnect on your mobile device either through our app or your preferred web browser</li> <li>2. Sign in to your account</li> <li>3. When logged in on a mobile device, you will have access to the full "Reading" functionality of the platform</li> </ol>

<b>Exceptions/Alternatives</b>	- None
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Reading Interface Sketch (Figure 2)</li> <li>2. Browsing Interface Sketch (Figure 1)</li> </ol>

<b>Number</b>	<b>Reader Use Case Number 9</b>
<b>Title</b>	<b>Reviews</b>
<b>Preparer</b>	Dallin Burton
<b>Actor/User</b>	Reader
<b>User Story</b>	As a reader, I want to provide and read reviews of books
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Open a book's detail's page</li> <li>2. Navigate to the reviews section</li> <li>3. Either read reviews, or add a review that allows likert scale for different book metrics, as well as general textbox for a review</li> </ol>
<b>Exceptions/Alternatives</b>	- None
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Book Details page</li> <li>2. Review Interface (Figure 6)</li> </ol>

<b>Number</b>	<b>Reader Use Case Number 10</b>
<b>Title</b>	<b>Book Working-Relevancy</b>
<b>Preparer</b>	Dallin Burton
<b>Actor/User</b>	Robot / Reader / Admin
<b>User Story</b>	As a reader, I want to be able to access books that are either currently in development or already completed. I don't want to read abandoned books.

<b>Course of Events</b>	<p>(Robot)</p> <ol style="list-style-type: none"> <li>1. Get a list of all the books that haven't been marked as completed or haven't been worked on in a while</li> <li>2. Deprioritize the search results, particularly for browsing new books</li> </ol> <p>(User)</p> <ol style="list-style-type: none"> <li>1. Browse different books</li> <li>2. Notice the lack of abandoned books</li> <li>3. If a user finds an abandoned book, allow for the submission of a report</li> </ol> <p>(Admin)</p> <ol style="list-style-type: none"> <li>1. Review submissions for abandoned books</li> <li>2. Manually mark a book as abandoned</li> <li>3. Allow for writers to dispute, with metrics proving otherwise</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- Sometimes it is possible that writers can erroneously mark their book as done in order to not get deprioritized. Have a reporting system to allow admin settings for a book</li> </ul>
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Book Discovery</li> <li>2. Admin Reporting View</li> <li>3. Book Details</li> </ol>

<b>Number</b>	<b>Reader Use Case Number 11</b>
<b>Title</b>	<b>Author Interaction</b>
<b>Preparer</b>	Tanner VonWald-Villiard
<b>Actor/User</b>	Reader
<b>User Story</b>	As a reader, I want to engage with and interact with authors I follow outside their works.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Select any author on the platform to open their "Author Page"</li> </ol>

	<ol style="list-style-type: none"> <li>2. View the authors listed works, linked accounts, and posted updates.</li> <li>3. Comment on the author's posts or send them a personal message</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- None</li> </ul>
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Author Page Sketch (Figure 7)</li> </ol>

## Writer / Editor Cases

Number	Writer Use Case Number 1
Title	Check for Contradictions
Preparer	David Harrington
Actor/User	Writer
User Story	As a writer, I want to ensure that the portion of writing I have put hours of work into doesn't contradict earlier sections of my book.
Course of Events	<ol style="list-style-type: none"> <li>1. From the page the work is written on, navigate to click the Continuity Checking Option on the newest chapter.</li> <li>2. Use the Continuity Checker to search the work for possible contradictions, either in the chapter, the book or in the book's facts wiki.</li> <li>3. The checker finishes its work, returns any references to contradictory statements.</li> </ol>
Exceptions/Alternatives	<ul style="list-style-type: none"> <li>- I manually check my 'fact wiki' against what I have written for contradictions.</li> </ul>
Related UI	<ul style="list-style-type: none"> <li>- Continuity Checker View Sketch (Figure 8)</li> </ul>

Number	Writer Use Case Number 2
Title	Road Unblocker Use Case 1
Preparer	David Harrington
Actor/User	Writer
User Story	As a writer, I've written a significant portion of my story, and I have a specific idea for what I want to happen next, but I don't know how to write it, and I would like assistance with this step.

<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. A significant amount of work has been written, and a plan for what comes next is defined, but the way to connect those two points in the story is unclear.</li> <li>2. Use the Road unblocker to generate sample prompts to aid the reader in progressing through writer's block.</li> <li>3. The writer should decide how to use the generated results, ideally some part of a way forward is clear.</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- I persevere through my writer's block, or I don't, and I spend a lot of time being unproductive when I want to be writing.</li> </ul>
<b>Related UI</b>	<ul style="list-style-type: none"> <li>- None</li> </ul>

<b>Number</b>	<b>Writer Use Case Number 3</b>
<b>Title</b>	<b>Road Unblocker Case 2</b>
<b>Preparer</b>	David Harrington
<b>Actor/User</b>	Writer
<b>User Story</b>	As a writer, when I have written a large amount, but I have no idea where to go next with my story, I would like to bypass this writer's block.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Some work is written for the story, but the way forward is unclear.</li> <li>2. Rather than providing the road unblocker with a sample prompt, allow it to freestyle from the current point in the story.</li> <li>3. The unblocker will write prompts that continue from where the story left off.</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- See Case 2's Alternative</li> </ul>

<b>Related UI</b>	- None
-------------------	--------

<b>Number</b>	<b>Writer Use Case Number 4</b>
<b>Title</b>	<b>Viewing Chapter Feedback</b>
<b>Preparer</b>	David Harrington
<b>Actor/User</b>	The Writer
<b>User Story</b>	As a writer, I want to see feedback on my chapter from the general audience.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. The newest chapter of the book is uploaded.</li> <li>2. After some time has passed, navigate to the story's feedback section,</li> <li>3. Filter feedback by chapter and look for the newest.</li> <li>4. Read the review.</li> </ol>
<b>Exceptions/Alternatives</b>	- None
<b>Related UI</b>	1. Author Chapter Reviews View Sketch (Figure 9)

<b>Number</b>	<b>Writer Use Case Number 5</b>
<b>Title</b>	<b>View Detailed Feedback</b>
<b>Preparer</b>	David Harrington
<b>Actor/User</b>	The Writer
<b>User Story</b>	As a writer, I want to view detailed feedback for what I have written.
<b>Course of Events</b>	1. The newest chapter of the book is uploaded.

	<ol style="list-style-type: none"> <li>2. After some time has passed, navigate to the story's feedback section, select the chapter annotation view</li> <li>3. Browse through the annotated version of the latest chapter, read the comments and annotations.</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- Nobody has proofread my work, there is nothing to see</li> </ul>
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Proofread Edit View Sketch (Figure 10)</li> </ol>

<b>Number</b>	<b>Writer Use Case Number 6</b>
<b>Title</b>	<b>Reference Key Story World Facts</b>
<b>Preparer</b>	David Harrington
<b>Actor/User</b>	The Writer
<b>User Story</b>	As a writer, I would like a place where I can keep facts about various elements of my story, settings, characters, magic, timelines, etc. Something I can conveniently reference and perhaps update as I am writing or brainstorming.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Navigate to the story's wiki page.</li> <li>2. Enter the name of the setting and navigate to its relevant page.</li> <li>3. Optionally, navigate to places in the book describing the place, linked in the wiki, for exact context.</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- I have simply neglected to make a page regarding the setting I am returning to.</li> </ul>
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Wiki Sample Page Sketch (Figure 11)</li> </ol>

<b>Number</b>	<b>Writer Use Case Number 7</b>
<b>Title</b>	<b>Editing Character Sheets</b>
<b>Preparer</b>	Novella
<b>Actor/User</b>	The Writer
<b>User Story</b>	When writing a story, every character relating to the story has their own background and characteristics. As a writer, I would like to document information for each character.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Log in to user account</li> <li>2. Set the role to writer. This will open up a new tab in your window for writer's view</li> <li>3. Open the book detail section tab and view the character sheets from the right side tab under 'Character Sheets'</li> <li>4. Writer can view, edit and add character sheets</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- Background Characters that have an insignificant appearance likely won't have sheets.</li> </ul>
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Character Sheet Broad View Sketch (Figure 12)</li> <li>2. Character Sheet Specific View Sketch (Figure 13)</li> </ol>

<b>Number</b>	<b>Writer Use Case Number 8</b>
<b>Title</b>	<b>Annotate Chapter</b>
<b>Preparer</b>	Novella
<b>Actor/User</b>	The Writer / Editor(s)
<b>User Story</b>	On a certain part of a story or chapter or sentence, sometimes as a writer or editor, we want to be able to add some notes about it. Whether it is writing suggestions or our personal comments.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. Log in to user account</li> <li>2. Set the role to writer. This will open up a new tab in your window for writer's view</li> </ol>

	<ul style="list-style-type: none"> <li>3. The default section tab will set user's view directly to the writing section tab</li> <li>4. Between the writing space and the right tab(Outline), users can highlight the sentence or parts they want to comment or add suggestion and a little bubble will appear for them to input their comments or suggestions</li> </ul>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- None</li> </ul>
<b>Related UI</b>	<ul style="list-style-type: none"> <li>1. Book Writing View (Figure 14)</li> </ul>

Number	<b>Writer Use Case Number 9</b>
<b>Title</b>	<b>Enumerate the Chapter Outline</b>
<b>Preparer</b>	Novella
<b>Actor/User</b>	The Writer
<b>User Story</b>	As a writer, I already have a vision of what I would like to happen in a chapter. I would feel more comfortable to have it on sight as I'm writing and be able to mark it off once it's done, like a to-do list.
<b>Course of Events</b>	<ul style="list-style-type: none"> <li>1. Log in to user account</li> <li>2. Set the role to writer. This will open up a new tab in your window for the writer's view</li> <li>3. The default section tab will set user's view directly to the writing section tab</li> <li>4. The right tab named Outline, writers can make a list of scenes that they want to outline in this chapter. For each scene, it's provided with another sublist that will be marked with the status of the completion of the items, like a to-do list. The completion of that overall scene is shown in the status bar below the scene name.</li> </ul>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- A short story or piece of poetry that needs no outline.</li> </ul>
<b>Related UI</b>	<ul style="list-style-type: none"> <li>1. Book Writing View (Figure 14)</li> </ul>

<b>Number</b>	<b>Writer Use Case Number 10</b>
<b>Title</b>	<b>Check Demographics of my Book</b>
<b>Preparer</b>	Novella
<b>Actor/User</b>	The Writer / Editor(s)
<b>User Story</b>	After publishing a book, as a writer, I want to see and analyze the reception of the book.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. In your writer's view, set the tab under demographics</li> <li>2. On the left side, users can see which analysis they want to view, includes: overview, engagement, characters, readers and feedbacks</li> </ol>
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Demographics Overview View Sketch (Figure 15)</li> <li>2. Demographics Engagement View Sketch (Figure 16)</li> <li>3. Demographics Character View Sketch (Figure 17)</li> <li>4. Demographics Reader View Sketch (Figure 18)</li> <li>5. Demographics Feedback View Sketch (Figure 19)</li> </ol>

<b>Number</b>	<b>Writer Use Case Number 11</b>
<b>Title</b>	<b>Fill in Book Details</b>
<b>Preparer</b>	Novella
<b>Actor/User</b>	The Writer / Editor(s)
<b>User Story</b>	As a presentation and book properties, book details must be documented. Additionally, the many tags or genres of a story can be confusing and often missed. Content tagging would be very helpful to automatically detect popular tags relevant to the story and any detected sensitive topics useful for readers' warnings.

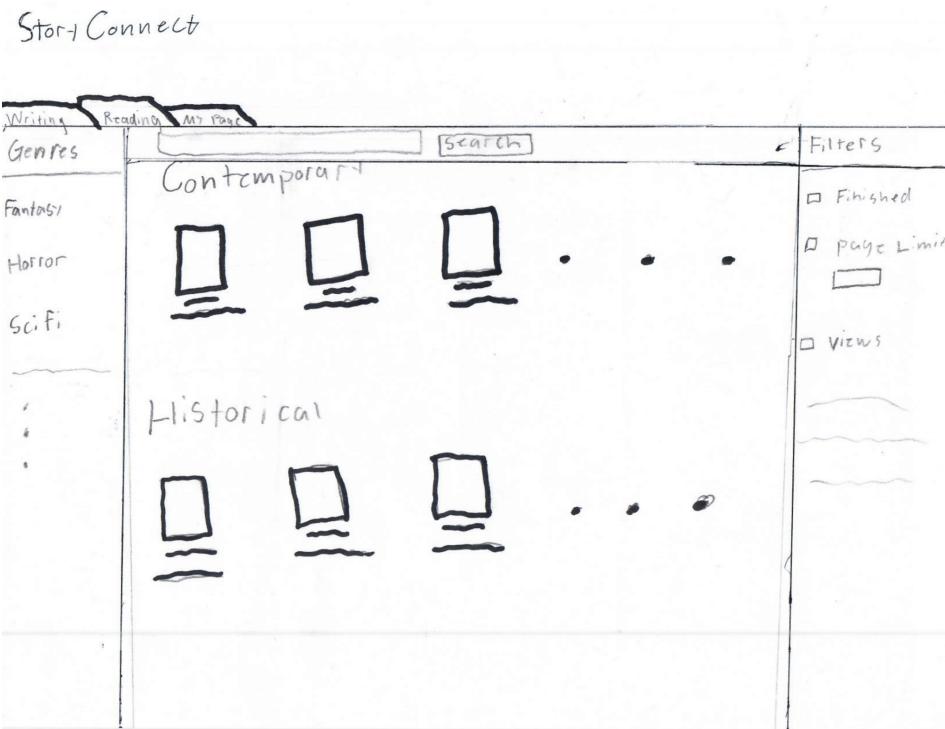
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. In your writer's view, set the tab under Book Details</li> <li>2. On the right side, make sure you are viewing Front Matter page that contains all the book details including book cover, title, target audience, language, completion status, mature status, synopsis and the content tagging(automation is available)</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- We could manually fill in the book's details and tags without using the automated process.</li> </ul>
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Book Details View Sketch (Figure 20)</li> </ol>

<b>Number</b>	<b>Writer Use Case Number 12</b>
<b>Title</b>	<b>Setup Copyright for the book</b>
<b>Preparer</b>	Novella
<b>Actor/User</b>	The Writer / Editor(s)
<b>User Story</b>	Whether I'm writing my own original story or translating someone else's or making fiction out of an existing 'universe', as a writer, I would like to have appropriate copyright of my book.
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. In your writer's view, set the tab under Book Details</li> <li>2. On the right side, ensure you are viewing the Copyright page that contains information about how copyright is defined in this platform and the selections of copyrights we offer.</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- None</li> </ul>
<b>Related UI</b>	<ol style="list-style-type: none"> <li>1. Copyright View Sketch (Figure 21)</li> </ol>

<b>Number</b>	<b>Writer Use Case Number 13</b>
---------------	----------------------------------

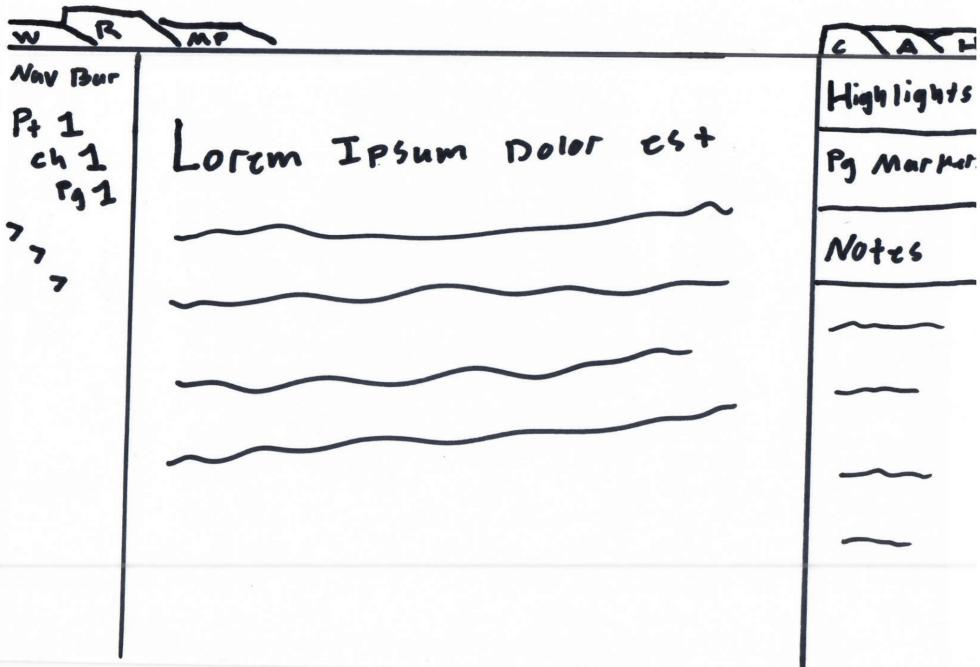
<b>Title</b>	<b>Creating a New Book</b>
<b>Preparer</b>	Tanner VonWald-Villiard
<b>Actor/User</b>	The Writer
<b>User Story</b>	As a writer, I wish to write a new story
<b>Course of Events</b>	<ol style="list-style-type: none"> <li>1. From the “My Page” tab (default page) select “New Book” under the “My Books” banner</li> <li>2. Enter the title of the book and press “Done”</li> <li>3. Use the writing interface to begin the new book</li> </ol>
<b>Exceptions/Alternatives</b>	<ul style="list-style-type: none"> <li>- None</li> </ul>
<b>Related UI</b>	<ul style="list-style-type: none"> <li>- None</li> </ul>

# UI Sketches

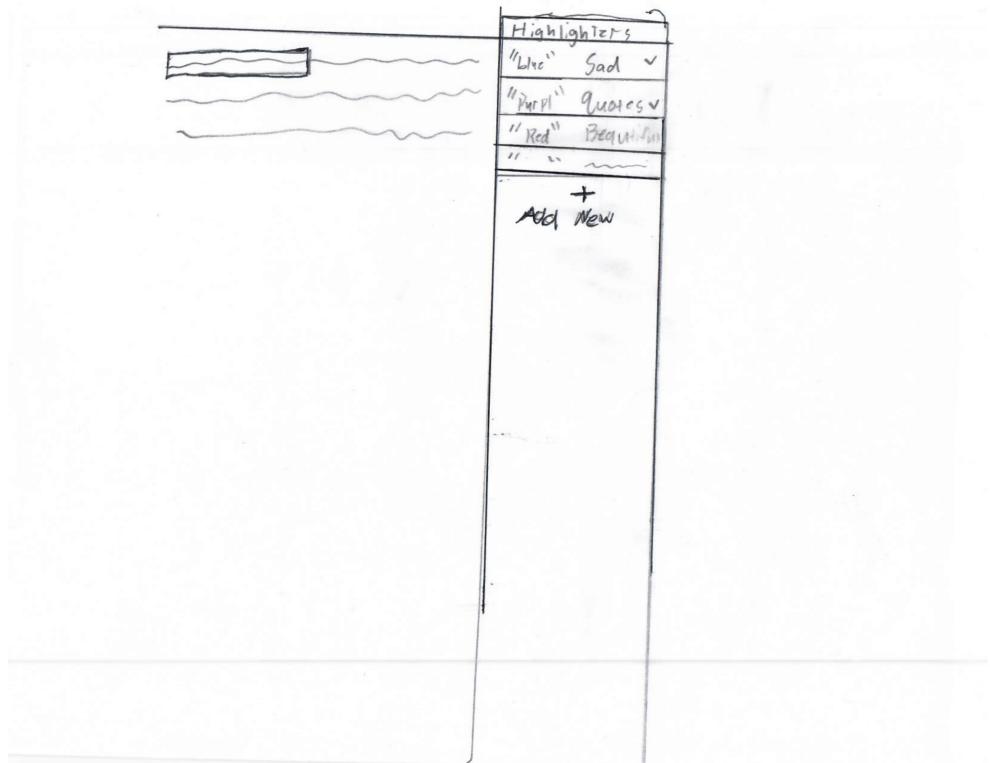


(Figure 1: Browsing Interface Sketch)

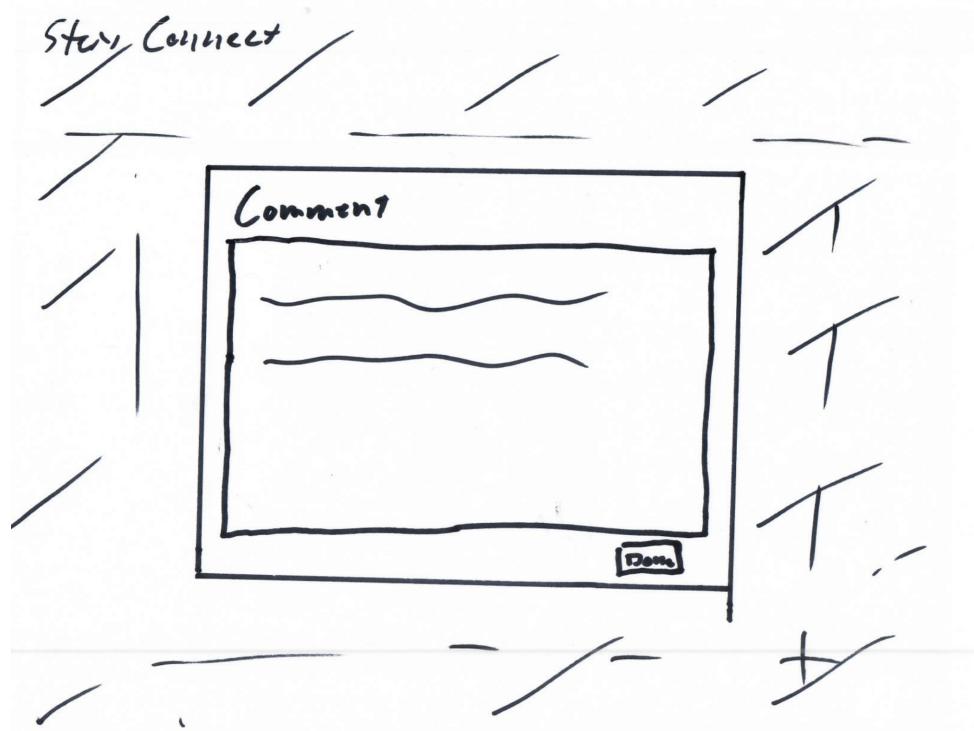
## Story Connect



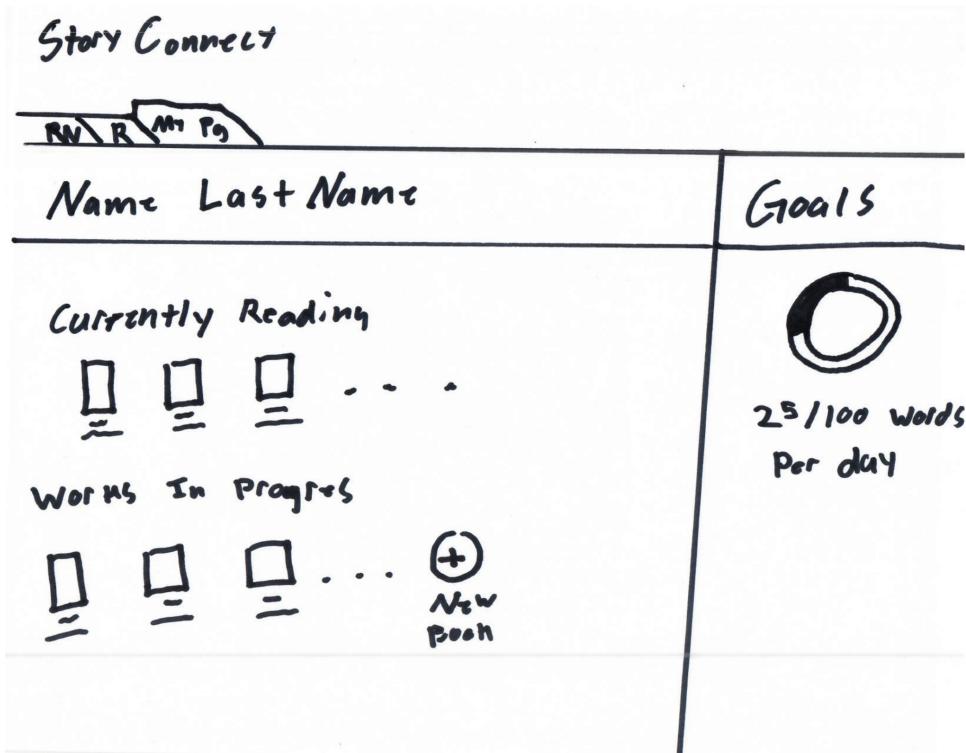
(Figure 2: Reading Interface Sketch)



(Figure 3: Annotation Sidebar Sketch)

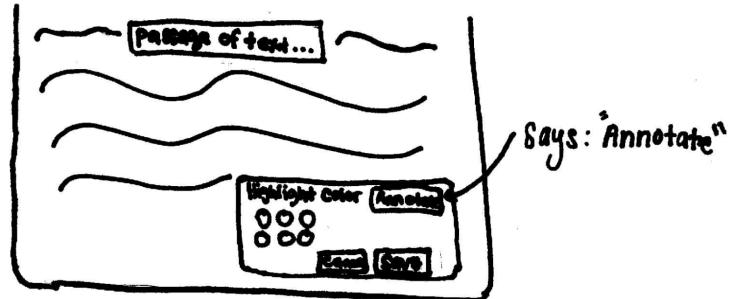


(Figure 4: Comment Sidebar Sketch)

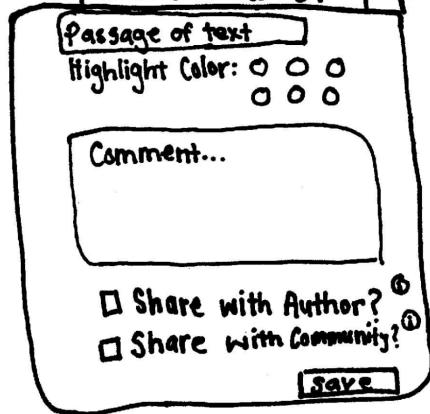


(Figure 5: My Page Sketch)

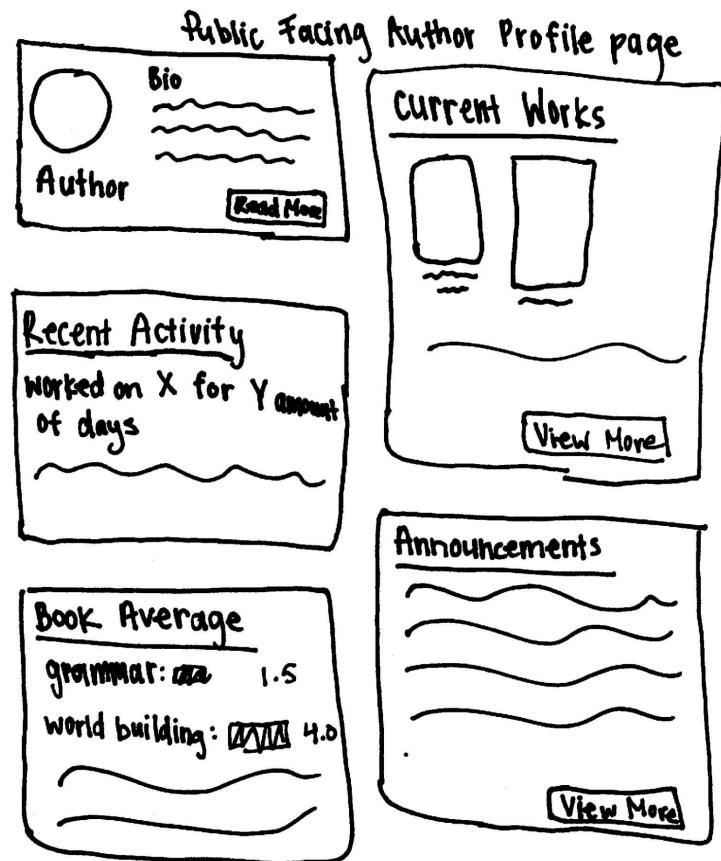
### Annotation / highlighting



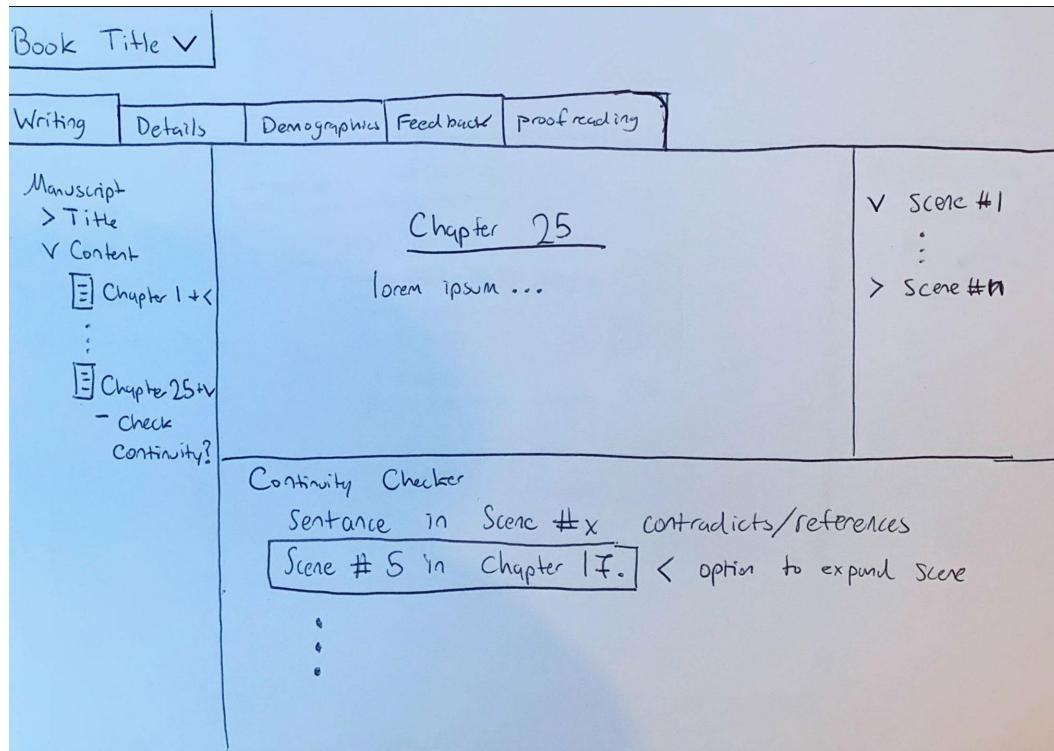
### Expanded View of popup



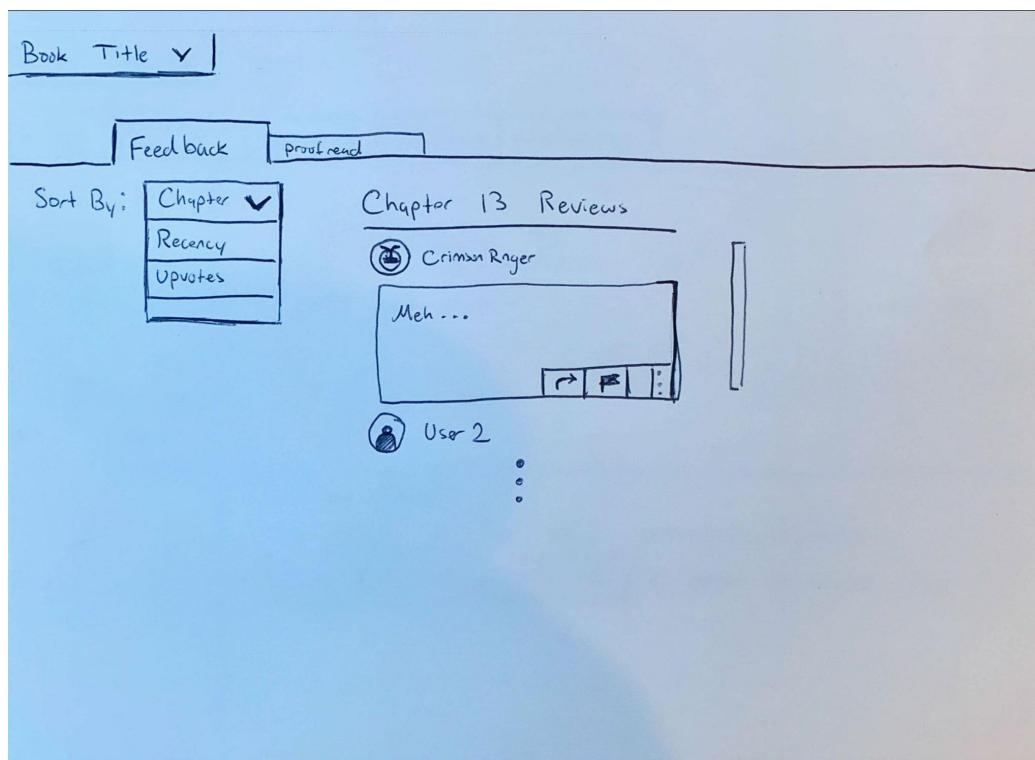
(Figure 6: Review Highlighting/Annotating Interface)



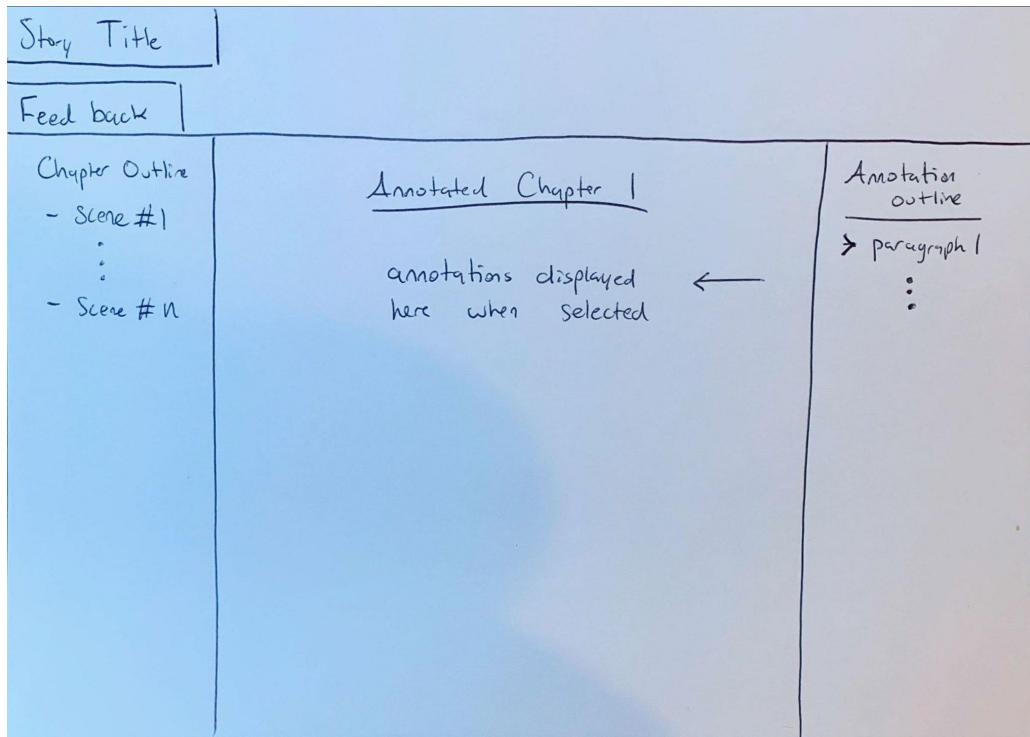
(Figure 7: Author Page Sketch)



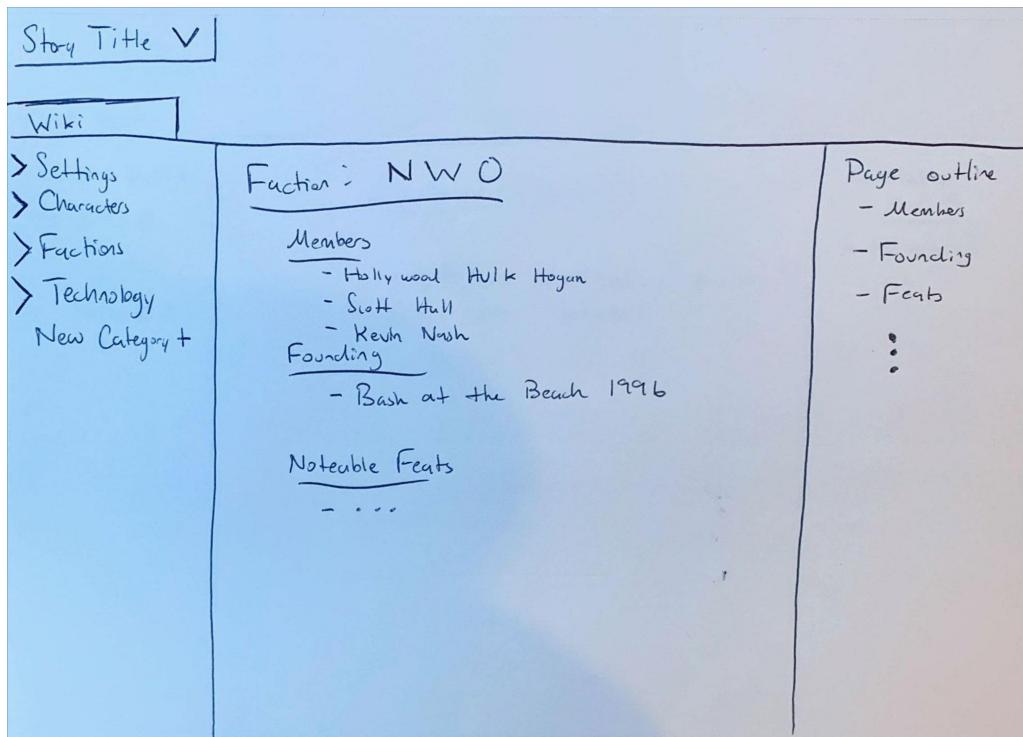
(Figure 8: Continuity Checker View Sketch)



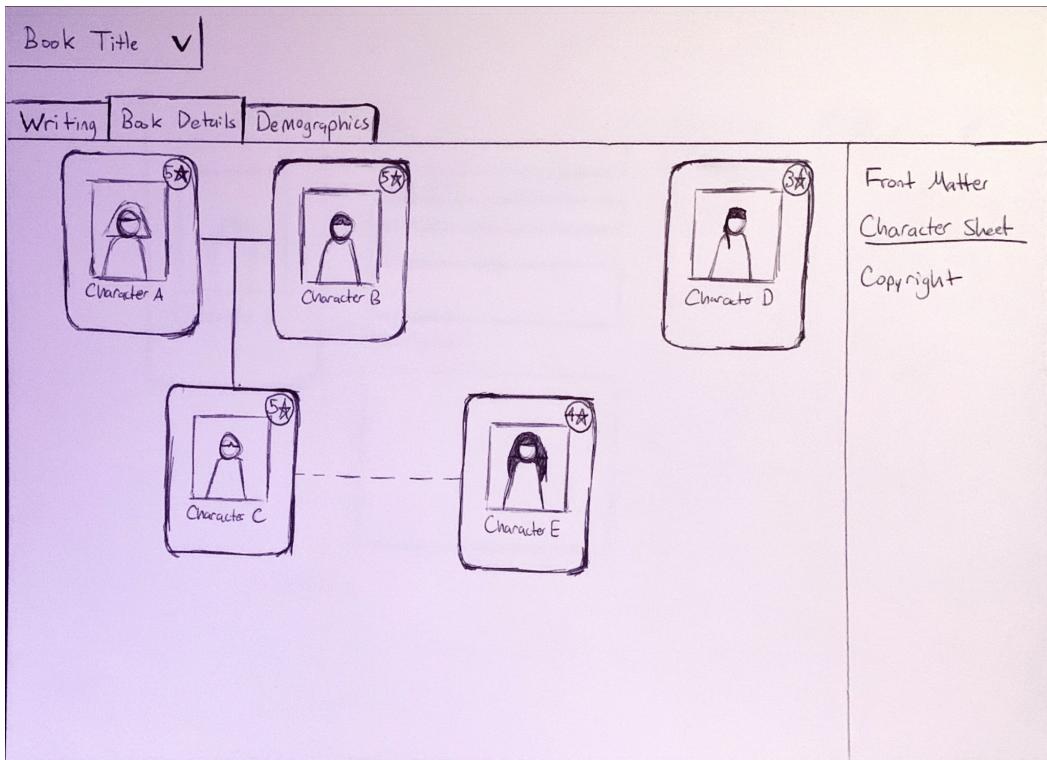
(Figure 9: Author Chapter Reviews View Sketch)



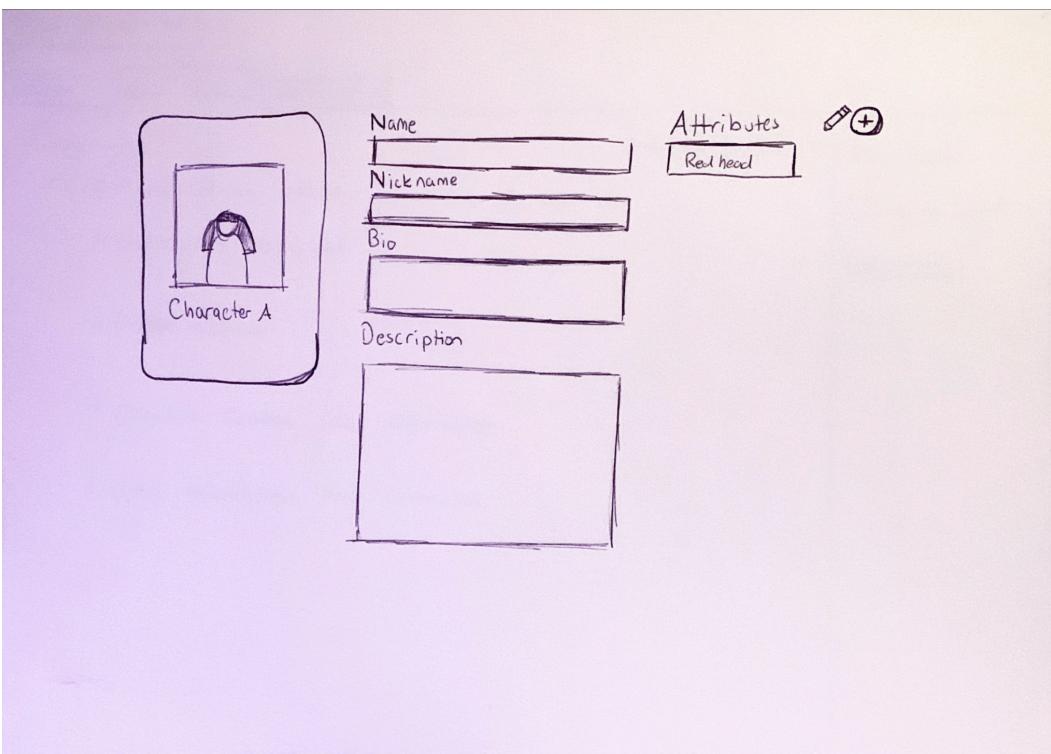
(Figure 10: Proofread Edit View Sketch)



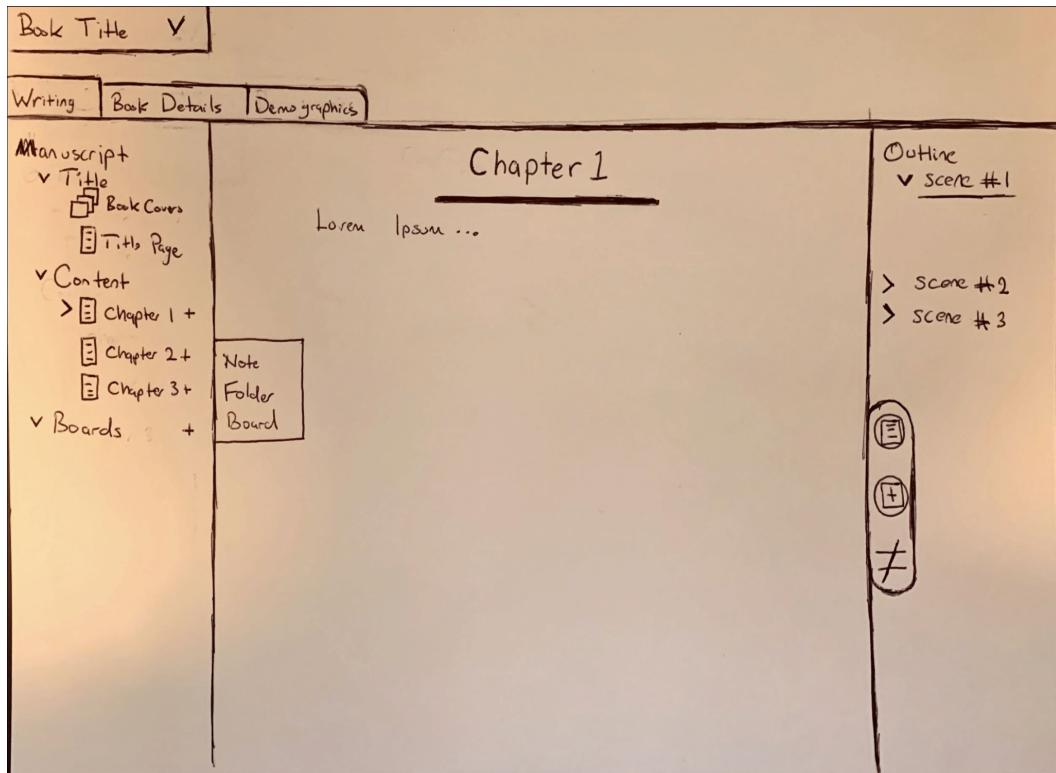
(Figure 11: Wiki Sample Page View Sketch)



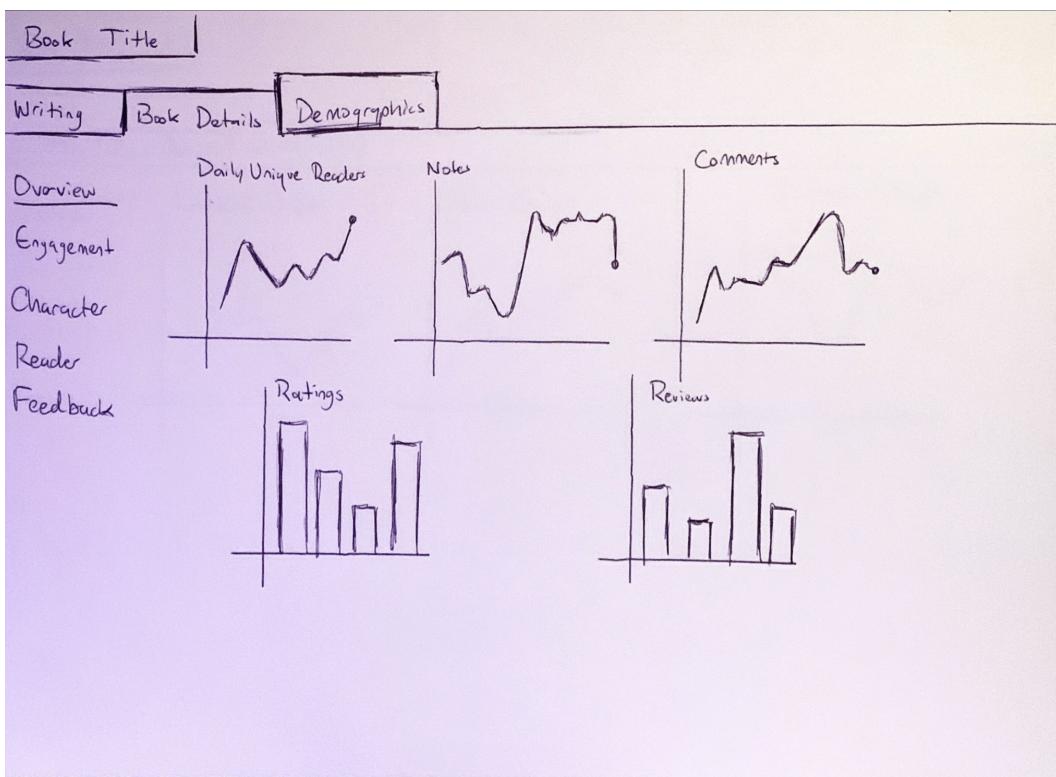
**(Figure 12: Character Sheet Broad View Sketch)**



**(Figure 13: Character Sheet Specific View Sketch)**



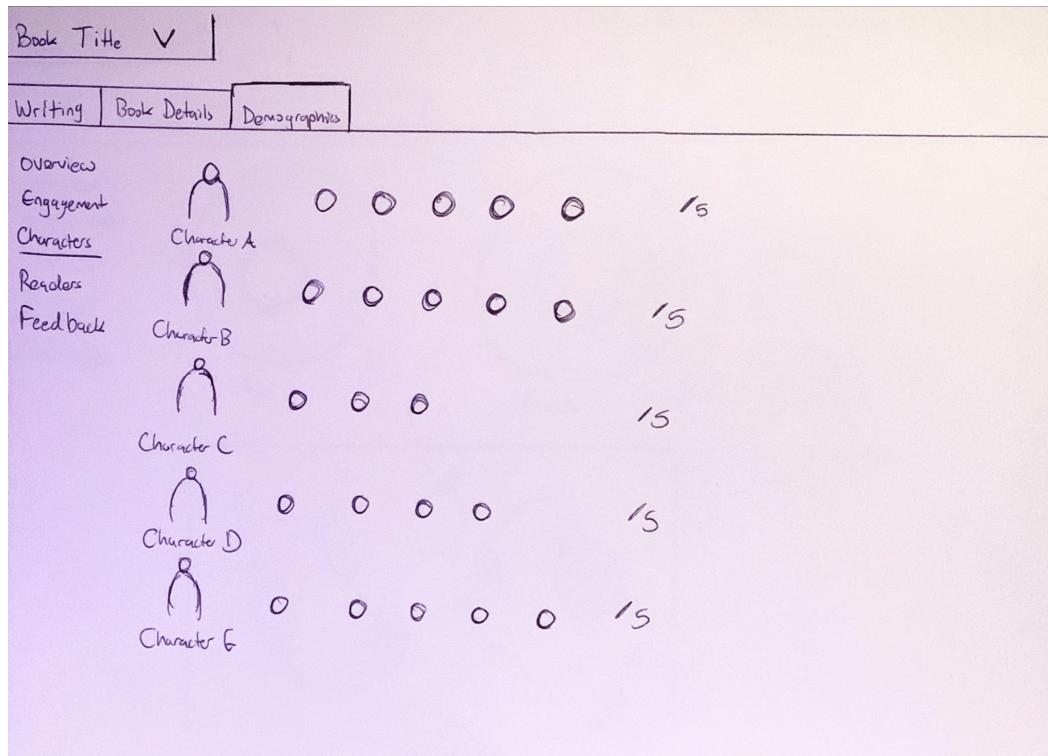
(Figure 14: Book Writing View Sketch)



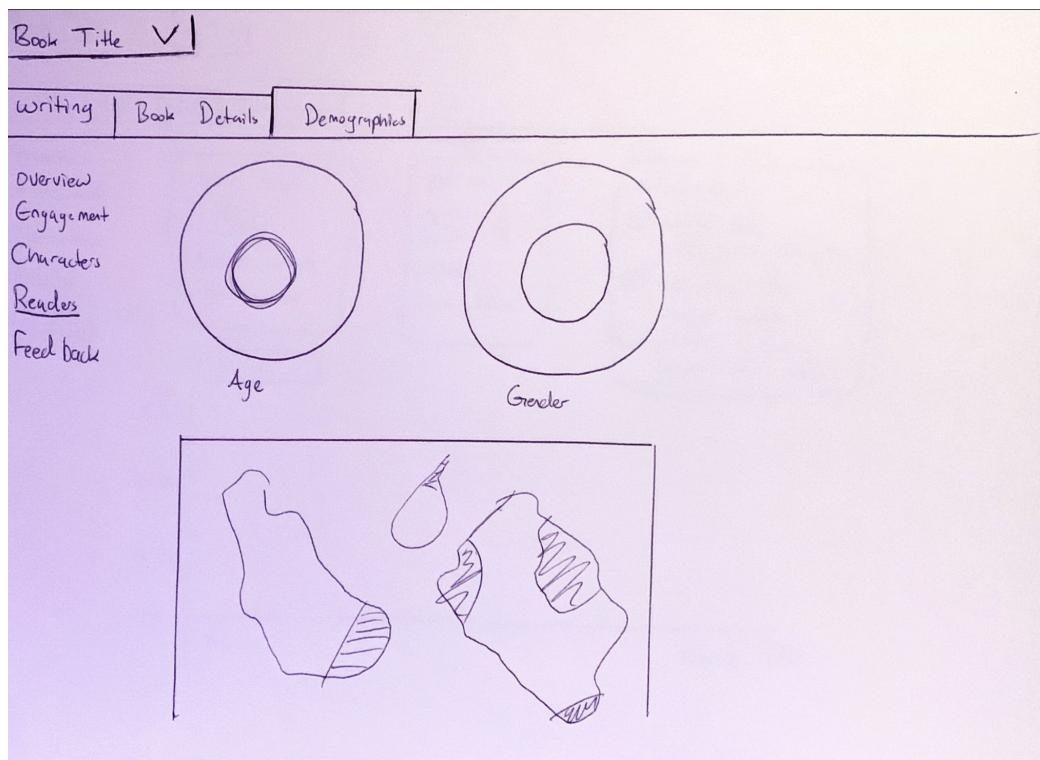
(Figure 15: Demographics Overview View Sketch)



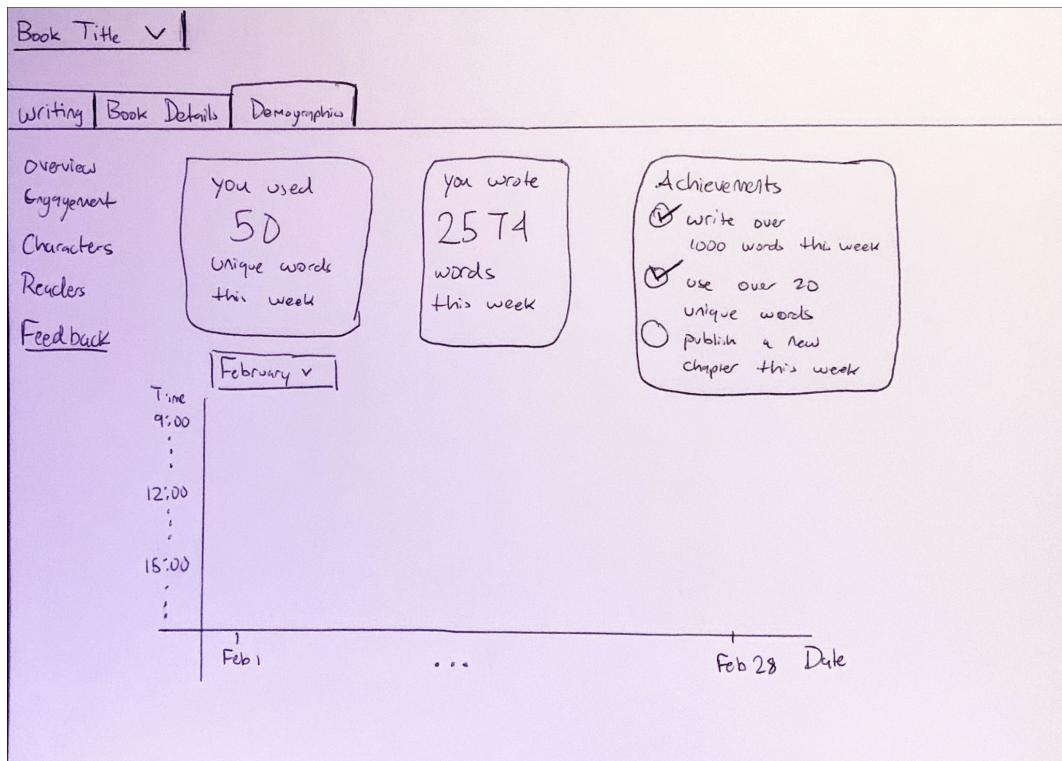
(Figure 16: Demographics Engagement View Sketch)



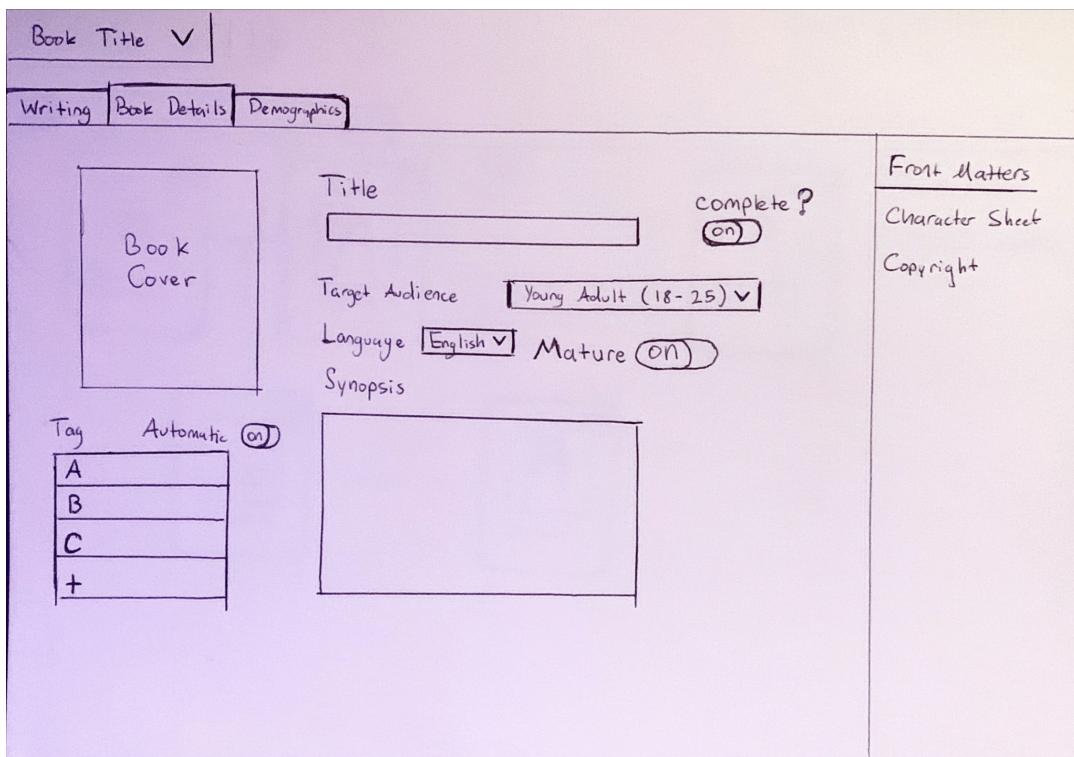
(Figure 17: Demographics Character View Sketch)



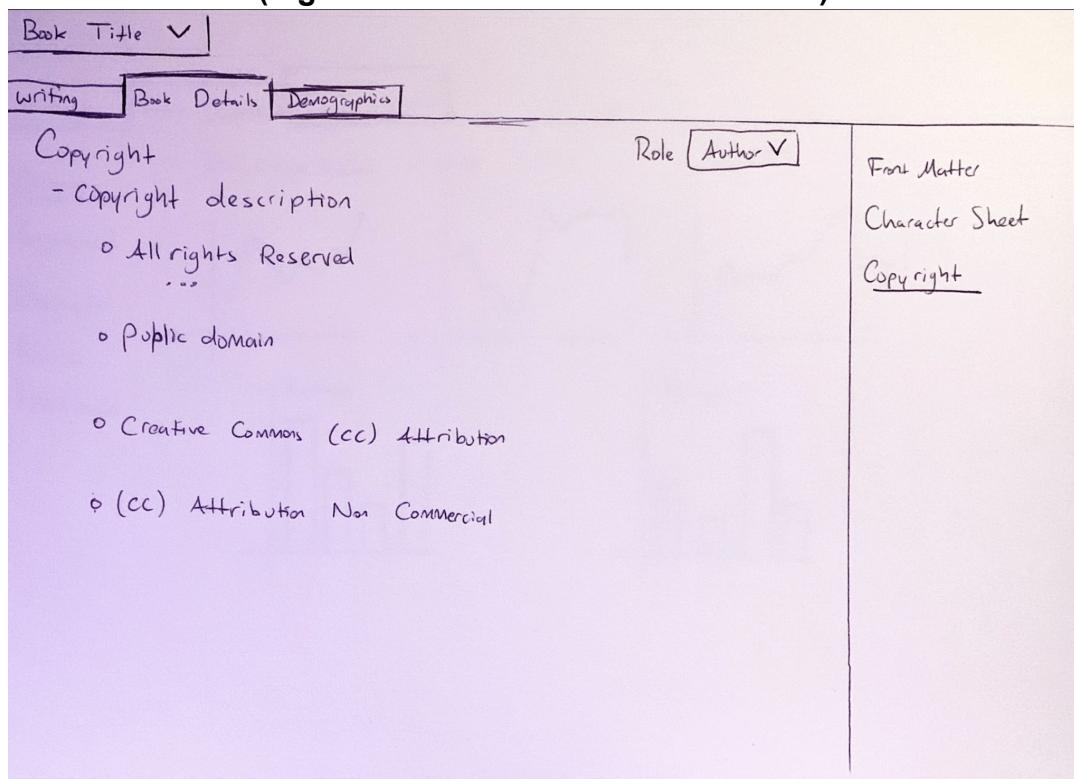
**(Figure 18: Demographics Reader View Sketch)**



**(Figure 19: Demographics Feedback View Sketch)**



**(Figure 20: Book Details View Sketch)**



**(Figure 21: Book Copyright View Sketch)**

# Revisions

## Executive Summary

- Revised language near end of executive summary in solution section.
- Condensed redundant language in introduction, clarified description and language describing our solution idea. Rewrote language describing the Reader User's role and Language describing what we use the AI API tools for in the writer's process.

## Background

- Changed from digital ocean to AWS
- Corrected typos and grammar
- Clarified and smoothed out the relationship between users, improved opening sentence of User's paragraph. Improved language describing the simplicity of the Reader role.
- Improved grammar/language describing findings of Wattpad's and Chapterly's User Experiences and Features.
- We improved the flow and cohesion of language describing the front end background.
- Specified language and altered meaning to include our Use of the Rest API
- Better described/justified our reasons for using DJANGO referring to later requirements in referring to use of Python/DJANGO library.
- Altered language describing reason for use of PostgresQL to make it read better.
- Added use of AWS Rather than digital ocean for hosting web app.
- Altered word choice for transition to back end conclusion.
- Altered language to better emphasize the fact that the browser settings aren't recommendations as suggested before, but requirements.
- Removed unnecessary language in software analysis, better emphasized necessity of Docker support for cloud service providers.

## Requirement Analysis

- Changed from digital ocean to AWS
- Revised small language errors
- Improved language describing system architecture, and added more descriptiveness to use of machine learning platforms and APIs. Added more reference to our use of REST APIs to backend description. Made front end requirements analysis less redundant.
- Further emphasized the dependency of Container Technologies in Django Requirements, removed redundant language.
- Removed redundant language in Postgres Analysis, polished and further emphasized user consent for data.

## Software Engineering Tools and Techniques

- No revisions made

## Timeline

- Recently added

## Use Cases

- Revised language in events section of use cases for writers to not be framed in first person.
- Revised redundant use case exception to refer to previous case's exception, as they are linked.

## UI Sketches

- No revisions made