# **Client Requirement Specification (CRS)**

## **Project: Semantic Book Recommender with LLMs**

**Client:** NextRead Corp.

**Author:** [Your Name/Company Name]

**Document Version:** 1.0

**Date:** September 28, 2025

[Page 1]

## **Table of Contents**

1. Client Overview ..................................................................................... 3  
   1.1. Introduction to NextRead Corp ................................................... 3  
   1.2. Mission and Vision ...................................................................... 4  
   1.3. Market Position and Strategic Goals ......................................... 4
2. Problem Domain .................................................................................... 5  
   2.1. The Challenge of Content Discovery ........................................ 5  
   2.2. Limitations of Existing Recommendation Systems ................. 6  
   2.3. Business Opportunity .................................................................. 7
3. Functional Requirements ................................................................... 8  
   3.1. FR-1: Semantic Querying System ............................................ 8  
   3.2. FR-2: Recommendation Display .............................................. 9  
   3.3. FR-3: Genre-Based Filtering .................................................. 10  
   3.4. FR-4: Emotion-Based Filtering ............................................... 11  
   3.5. FR-5: User Interface Components ......................................... 12
4. Non-Functional Requirements ....................................................... 13  
   4.1. NFR-1: Performance ................................................................ 13  
   4.2. NFR-2: Usability ......................................................................... 14  
   4.3. NFR-3: Reliability and Availability ........................................ 15  
   4.4. NFR-4: Scalability ..................................................................... 16  
   4.5. NFR-5: Maintainability ............................................................. 16
5. Business Rules .................................................................................... 17  
   5.1. Data Eligibility and Processing Rules .................................... 17  
   5.2. Recommendation Logic Rules ............................................... 18  
   5.3. User Interface Default States ................................................... 18
6. Use Case Descriptions ...................................................................... 19  
   6.1. Use Case 1: Basic Semantic Search for a Book ................. 19  
   6.2. Use Case 2: Search Refined by Genre ................................ 20  
   6.3. Use Case 3: Search Based on Emotional Tone .................. 21
7. **Stakeholder List** ................................................................................ 22
8. **Glossary of Terms** ............................................................................ 23
9. **Document Approval** ......................................................................... 25

[Page 2]

## **1. Client Overview**

### **1.1. Introduction to NextRead Corp.**

NextRead Corp. is a forward-thinking technology startup founded in 2024 with the ambition to revolutionize the digital book discovery landscape. The company operates in the highly competitive online retail and content recommendation market, focusing exclusively on literature. While established players dominate the market through sheer volume and brand recognition, NextRead Corp. aims to differentiate itself by offering a uniquely intelligent and user-centric discovery experience.

The founding team comprises a blend of literary enthusiasts, data scientists, and machine learning engineers who share a common frustration with the current state of online book recommendations. They believe that the process of finding a new book should be as delightful and insightful as discussing literature with a well-read friend, a sentiment that current algorithmic approaches often fail to capture.

NextRead Corp. is currently in its seed stage, having secured funding to develop a minimum viable product (MVP) that showcases its core technological innovations. This Semantic Book Recommender project represents the cornerstone of that MVP, designed to demonstrate the tangible benefits of using advanced AI to connect readers with books they will truly love. The company's headquarters are located in Madurai, Tamil Nadu, reflecting its commitment to fostering tech talent in emerging hubs.

[Page 3]

### **1.2. Mission and Vision**

**Mission:** To empower readers to discover books that resonate with their intellectual and emotional needs through intelligent, intuitive, and personalized technology.

**Vision:** To become the most trusted platform for literary discovery, where technology and human curation converge to create a world where no reader feels lost in the vastness of the literary world.

### **1.3. Market Position and Strategic Goals**

NextRead Corp. positions itself as a boutique innovator in a market dominated by giants like Amazon, Goodreads, and Barnes & Noble. Its competitive strategy is not to compete on inventory size or price, but on the quality and relevance of its recommendations.

**Strategic Goals for the Next 18 Months:**

1. **Launch a Successful MVP:** Deploy the Semantic Book Recommender as a public-facing web application to attract an initial user base and gather valuable feedback.
2. **Demonstrate Technological Superiority:** Prove that an LLM-powered, semantic-first approach yields more satisfying and relevant recommendations than traditional keyword-based and collaborative filtering systems.
3. **Build a Community:** Foster a community of passionate readers who value quality recommendations and are willing to contribute to the platform's growth.
4. **Secure Series A Funding:** Leverage the success of the MVP to attract further investment for scaling the platform, expanding the dataset, and developing new features.

This project is the critical first step in achieving these goals. Its success will serve as the primary proof-of-concept for NextRead Corp.'s innovative approach to book discovery.

[Page 4]

## **2. Problem Domain**

### **2.1. The Challenge of Content Discovery**

The digital age has made more books accessible than at any point in history. While this is a tremendous benefit, it has also created a significant challenge: the "paradox of choice." Readers are often overwhelmed by the sheer volume of options, making it difficult to find books that align with their specific tastes and current interests.

The problem domain this project addresses is the gap between how readers *think* about books and how digital systems *allow* them to search for them.

* **Human-Centric Search:** A reader might think, "I want to read a fast-paced thriller with a strong female lead that makes me think," or "I'm in the mood for a heartwarming, light read about friendship." These queries are based on theme, pacing, character archetypes, and emotional response.
* **System-Centric Search:** Most existing platforms force users to translate these complex desires into a limited set of keywords and filters (e.g., Genre: "Thriller," Tag: "female protagonist"). This translation is often imperfect, leading to frustrating and irrelevant results. The system fails to capture the *semantic intent* and *emotional context* of the user's request.

This project directly tackles the challenge of bridging this gap by creating a system that can understand and process human-centric search queries.

[Page 5]

### **2.2. Limitations of Existing Recommendation Systems**

The current market is dominated by recommendation engines that, while commercially successful, suffer from several well-documented limitations that NextRead Corp. aims to overcome.

1. **Keyword Dependency:**
   * **Problem:** Systems are highly reliant on exact keyword matches in titles, descriptions, and user-generated tags. A search for "a story about redemption" will fail if the word "redemption" is not explicitly mentioned, even if the theme is central to the book's plot.
   * **Impact:** This leads to a narrow and often brittle search experience, penalizing books with subtle or artistically written descriptions.
2. **Lack of Emotional Intelligence:**
   * **Problem:** Current systems are "mood-blind." They cannot differentiate between a tragic romance and an uplifting one, or a humorous historical account and a serious academic one, as long as the keywords match.
   * **Impact:** Users cannot filter by the emotional experience they are seeking, which is a primary driver for reading choices. This leads to unsatisfying recommendations (e.g., being recommended a heartbreaking novel when looking for a cheerful escape).
3. **The "Cold Start" Problem:**
   * **Problem:** Collaborative filtering systems require a significant amount of user data to work effectively. They struggle to provide good recommendations for new users or for newly published books with no rating history.
   * **Impact:** New users may have a poor initial experience and abandon the platform. New books may struggle to find their audience.
4. **Popularity Bias:**
   * **Problem:** Many algorithms tend to over-recommend popular, best-selling books, creating a feedback loop that makes popular items even more popular.
   * **Impact:** This "rich-get-richer" effect stifles diversity and makes it difficult for users to discover hidden gems or niche authors.

[Page 6]

### **2.3. Business Opportunity**

By addressing these limitations, NextRead Corp. has a significant opportunity to capture a dedicated segment of the market: discerning readers who prioritize recommendation quality over quantity.

The business opportunity lies in providing a premium discovery experience that:

* **Reduces Friction:** Makes finding the perfect book faster, easier, and more enjoyable.
* **Increases User Trust:** Delivers consistently relevant and insightful recommendations, establishing NextRead Corp. as a trusted literary curator.
* **Drives Engagement:** Encourages users to return to the platform for all their book discovery needs, building a loyal user base.
* **Creates a Defensible Niche:** Builds a competitive advantage based on superior AI technology that is difficult for larger, more established players to replicate without significant architectural changes.

This Client Requirement Specification (CRS) outlines the features and functionalities necessary to build the first version of this innovative system and capitalize on this market opportunity.

[Page 7]

## **3. Functional Requirements**

This section details the functional requirements of the Semantic Book Recommender system. Each requirement specifies a behavior or function that the system must provide.

### **3.1. FR-1: Semantic Querying System**

ID: FR-1

Requirement: The system must allow users to search for books using natural language queries that describe themes, plots, concepts, or styles, in addition to simple keywords.

Description:

The core feature of the application is its ability to understand the semantic meaning behind a user's query. The system should not be limited to matching exact words but should identify books whose descriptions are conceptually related to the user's input.

**User Stories:**

* As a reader, I want to type "a coming-of-age story set during a war" so that I can find novels that match this specific theme and setting.
* As a reader, I want to search for "books that feel like a Wes Anderson movie" so that I can discover quirky, character-driven stories.
* As a reader, I want to enter "a philosophical novel about the meaning of life" to find deep, introspective literature.

**Acceptance Criteria:**

* FR-1.1: The system must provide a text input field for the user to enter their query.
* FR-1.2: The system backend must convert the user's text query into a high-dimensional vector embedding.
* FR-1.3: The system must perform a similarity search between the query embedding and the pre-computed embeddings of all book descriptions in the database.
* FR-1.4: The search must return a list of books ranked by their semantic similarity (cosine similarity) to the user's query.

[Page 8]

### **3.2. FR-2: Recommendation Display**

ID: FR-2

Requirement: The system must display the search results in a visually appealing and informative manner.

Description:

The presentation of the recommended books is crucial for the user experience. The results should be easy to browse, and each item should provide enough information for the user to make an initial judgment without needing to click through to another page.

**User Stories:**

* As a reader, I want to see the book covers of the recommendations so I can quickly scan them visually.
* As a reader, I want to see the title, author, and a short snippet of the description for each recommended book so I can understand why it was suggested.

**Acceptance Criteria:**

* FR-2.1: The recommendations must be displayed in a grid layout (gallery).
* FR-2.2: The system shall display a maximum of 16 recommendations per search query.
* FR-2.3: Each item in the grid must display the book's cover image. If a cover image is not available, a default placeholder image must be shown.
* FR-2.4: Each item in the grid must have a caption below the image containing:
  + The book's full title.
  + The book's author(s), formatted grammatically (e.g., "Author A and Author B").
  + A truncated version of the book's description, limited to the first 30 words followed by an ellipsis (...).
* FR-2.5: The gallery must be able to dynamically update with new results each time a new search is submitted, without requiring a full page reload.

[Page 9]

### **3.3. FR-3: Genre-Based Filtering**

ID: FR-3

Requirement: The system must allow users to filter the semantic search results by a simplified genre category.

Description:

To help users narrow down their search results, a high-level genre filter is required. This filter will be based on a zero-shot classification model that categorizes books into broad genres.

**User Stories:**

* As a reader, after searching for "stories about space travel," I want to filter the results to see only "Fiction" to exclude non-fiction books about astronomy.
* As a reader, when looking for "a biography of a famous scientist," I want to filter by "Non-fiction" to ensure I only get factual accounts.

**Acceptance Criteria:**

* FR-3.1: The user interface must include a dropdown menu labeled "Select a category:".
* FR-3.2: The dropdown menu must contain the following options: "All", "Fiction", "Non-fiction".
* FR-3.3: The default selection for the category filter must be "All".
* FR-3.4: When a user selects "Fiction" or "Non-fiction", the system must apply this filter *after* performing the initial semantic search, showing only the books from the initial result set that match the selected category.
* FR-3.5: The filtering must happen in real-time upon submitting a new search with a category selected.

[Page 10]

### **3.4. FR-4: Emotion-Based Filtering**

ID: FR-4

Requirement: The system must allow users to filter or re-rank search results based on the desired emotional tone of the book.

Description:

This feature is a key differentiator for the platform. It allows users to align their book discovery with their current mood. The system will use pre-computed emotion scores for each book's description to facilitate this.

**User Stories:**

* As a reader, I'm feeling down and want to search for "a funny story about family" and then select the "Happy" tone to find something uplifting.
* As a reader, I'm in the mood for a gripping thriller, so I'll search for "a detective story" and then select the "Suspenseful" tone to get the most intense options.
* As a reader, I want to explore profound themes, so I'll search for "a novel about loss" and select the "Sad" tone to find emotionally resonant books.

**Acceptance Criteria:**

* FR-4.1: The user interface must include a dropdown menu labeled "Select an emotional tone:".
* FR-4.2: The dropdown menu must contain the following options: "All", "Happy", "Surprising", "Angry", "Suspenseful", "Sad".
* FR-4.3: The default selection for the tone filter must be "All".
* FR-4.4: When a user selects a specific tone (e.g., "Happy"), the system must take the results from the semantic search (and category filter, if applied) and re-rank them based on the corresponding emotion score (e.g., 'joy' score) in descending order.
* FR-4.5: The re-ranking must be applied upon submitting a new search with an emotional tone selected.

[Page 11]

### **3.5. FR-5: User Interface Components**

ID: FR-5

Requirement: The system's user interface must be clean, intuitive, and responsive.

Description:

The overall design of the user interface should be simple and focused on the core task of book discovery. It should be easy for a non-technical user to understand and operate.

**User Stories:**

* As a first-time user, I want the interface to be self-explanatory so I can start searching for books immediately without needing instructions.
* As a user on a mobile device, I want the interface to adapt to my screen size so that all controls are usable and the results are clearly visible.

**Acceptance Criteria:**

* FR-5.1: The UI must be contained within a single page to avoid navigation complexity.
* FR-5.2: All input controls (text box, dropdowns, button) must be clearly labeled.
* FR-5.3: The layout must be responsive, adjusting gracefully to different screen sizes, from mobile phones to desktop monitors.
* FR-5.4: The application must present a clean, modern aesthetic (as defined by the "Glass" theme in Gradio).
* FR-5.5: There should be clear visual separation between the input controls section and the recommendation results section.

[Page 12]

## **4. Non-Functional Requirements**

This section defines the non-functional requirements (NFRs) that specify the quality attributes of the system, such as performance, usability, and scalability.

### **4.1. NFR-1: Performance**

ID: NFR-1

Requirement: The system must be highly responsive and provide a smooth user experience.

Description:

System performance is critical for user retention. Users expect search results to be returned quickly. Delays in processing can lead to frustration and abandonment of the application. This requirement sets the performance targets for the system's key operations.

**Metrics:**

* **NFR-1.1: Query Response Time:** The end-to-end time from when a user clicks the "Find recommendations" button to when the results are displayed in the gallery must be **less than 3 seconds** on average for a standard internet connection. This includes the time for the API call to the embedding model, the vector search, all filtering and ranking logic, and UI rendering.
* **NFR-1.2: Initial Page Load Time:** The time for the Gradio application to load initially in the user's browser must be **less** than **5 seconds**.
* **NFR-1.3: Backend Data Loading:** The one-time loading of the pandas DataFrame and the Chroma DB vector store into memory at application startup should complete within **10 seconds**. This is a one-time cost and does not affect the per-query user experience.
* **NFR-1.4: Concurrent Users (Initial Target):** The system should be able to handle at least **5 concurrent users** performing searches without significant degradation in response time for the MVP phase.

[Page 13]

### **4.2.** NFR-2: Usability

ID: NFR-2

Requirement: The application must be highly intuitive and easy to use for the target audience of general readers.

Description:

Usability focuses on the quality of the user's interaction with the system. A highly usable system is one that a new user can learn to operate quickly and effectively with minimal confusion or frustration.

**Metrics:**

* **NFR-2.1: Learnability:** A new user should be able to successfully perform a search with at least one filter applied within **60 seconds** of their first visit to the application, without any external instructions.
* **NFR-2.2: Simplicity:** The user interface must not contain more than five interactive components (1 text box, 2 dropdowns, 1 button, 1 gallery) on the main screen to avoid overwhelming the user.
* **NFR-2.3: Feedback:** The system should provide implicit feedback that a search is in progress. The Gradio framework's default loading animation on the output component is sufficient to meet this requirement.
* **NFR-2.4: Error Handling:** In the event of a backend error (e.g., API key is invalid), the system should fail gracefully and not crash. It should display an empty result set or a user-friendly error message in the output area.

[Page 14]

### **4.3. NFR-3: Reliability and Availability**

ID: NFR-3

Requirement: The system must be reliable during normal operation.

Description:

Reliability is the measure of the system's ability to perform its required functions under stated conditions for a specified period. For the MVP, the focus is on stability during typical usage.

**Metrics:**

* **NFR-3.1: Availability:** For the initial deployment phase, the application should target an uptime of **99%** during standard operating hours. This requirement is primarily dependent on the stability of the hosting platform and the OpenAI API.
* **NFR-3.2: Data Integrity:** The data loaded from the CSV files must be consistent and accurate. The recommendation logic must correctly join data between the vector search results and the main DataFrame using the isbn13 key, with no data mismatches.
* **NFR-3.3: Session Stability:** A user session should remain stable, and the application should not crash or become unresponsive during repeated searches and filter changes.

[Page 15]

### **4.4. NFR-4: Scalability**

ID: NFR-4

Requirement: The system architecture should be designed to accommodate future growth in data volume and user traffic.

Description:

While the MVP will operate on a limited dataset, the underlying architecture should not have inherent bottlenecks that would prevent future scaling.

**Metrics:**

* **NFR-4.1:** Data **Scalability:** The system design must be capable of handling a dataset of at least **100,000 books** without requiring a complete architectural redesign. This implies that the choice of vector database (Chroma DB) and data handling (pandas) should support this scale, even if performance targets would need to be revisited with a larger dataset.
* **NFR-4.2: Component Decoupling:** The logic for data retrieval, filtering, and presentation should be modular. This will allow for individual components (e.g., the vector database) to be scaled or replaced independently in the future (e.g., moving from a local Chroma DB instance to a cloud-hosted solution).

### **4.5. NFR-5: Maintainability**

ID: NFR-5

Requirement: The source code must be well-structured, commented, and easy for new developers to understand and modify.

Description:

Maintainability is crucial for the long-term health of the project. A maintainable codebase reduces the cost of fixing bugs and adding new features.

**Metrics:**

* **NFR-5.1: Code Readability:** The Python code must adhere to PEP 8 style guidelines. Function and variable names should be descriptive and self-explanatory.
* **NFR-5.2: Modularity:** The code should be organized into logical functions (e.g., retrieve\_semantic\_recommendations, recommend\_books). The Gradio UI definition should be separate from the core business logic.
* **NFR-5.3: Documentation:** All key functions must have docstrings explaining their purpose, parameters, and return values. Comments should be used to clarify any complex or non-obvious logic.
* **NFR-5.4: Dependency Management:** All external dependencies must be explicitly listed in the requirements.txt file to ensure a reproducible environment.

[Page 16]

## **5. Business Rules**

This section defines the specific business rules and constraints that govern the operation of the Semantic Book Recommender system. These rules are not functional requirements themselves but are the policies and logic that guide how the functional requirements are implemented.

### **5.1. Data Eligibility and Processing Rules**

ID: BR-1

Rule: Data Source Eligibility

A book from the raw dataset is only eligible for inclusion in the recommendation system if it has a non-empty and valid value for the following fields: description, published\_year, average\_rating, and num\_pages.

* **Rationale:** The description is essential for semantic embedding and sentiment analysis. The other fields are required for data integrity and potential future features.

ID: BR-2

Rule: Description Length for Processing

For a book to be included in the final dataset for embedding and sentiment analysis, its description must contain a minimum of 25 words.

* **Rationale:** Very short descriptions (e.g., "A novel.") do not contain enough semantic information to generate meaningful embeddings or accurate emotion scores, leading to poor recommendation quality.

ID: BR-3

Rule: Emotion Score Aggregation

The final emotion score for a book (e.g., 'joy', 'fear') is determined by taking the maximum score for that emotion from any single sentence within its description.

* **Rationale:** Book descriptions often cover multiple emotional tones. This rule ensures that the system captures the most intense emotional signal present in the text, which is likely to be a key marketing point and a strong indicator of the book's mood.

[Page 17]

### **5.2. Recommendation Logic Rules**

ID: BR-4

Rule: Search and Filter Precedence

The recommendation logic must be executed in a specific order:

1. **Semantic Search:** The initial set of candidates is always generated by the semantic similarity search against the user's query.
2. **Category Filter:** The genre filter ('Fiction'/'Non-fiction') is applied to the initial set of semantic results.
3. **Emotional Tone Re-ranking:** The emotional tone filter is applied last, re-ranking the already filtered list of books.

* **Rationale:** This ensures that the primary driver of the recommendation is always semantic relevance. The filters serve to refine this relevant set, not to generate a new set of recommendations independently.

ID: BR-5

Rule: Initial Result Set Size

The initial semantic search must retrieve a larger set of candidates (default: 50 books) from the vector database before any filters are applied.

* **Rationale:** Retrieving a larger initial set increases the likelihood that enough relevant books will remain after the category and tone filters are applied, preventing cases where the final result set is too small or empty.

ID: BR-6

Rule: Final Result Set Size

The application will display a maximum of 16 books to the user.

* **Rationale:** This prevents overwhelming the user with too many options and keeps the user interface clean and focused.

### **5.3. User Interface Default States**

ID: BR-7

Rule: Default Filter State

When the application is first loaded, both the "Category" and "Emotional Tone" dropdown filters must default to the "All" option.

* **Rationale:** This provides the broadest and most inclusive search experience for first-time users, who can then choose to apply filters as needed.

[Page 18]

## **6.** Use Case Descriptions

This section provides detailed descriptions of key user interactions with the system, outlining the actors, preconditions, flow of events, and postconditions for each use case.

### **6.1. Use Case 1: Basic Semantic Search for a Book**

Use Case ID: UC-01

Use Case Name: Perform a Basic Semantic Search

Actor: Reader (any user of the application)

Description: The reader wants to find books based on a theme or concept without applying any specific filters.

**Preconditions:**

* The application is loaded and running in the reader's browser.
* The data and vector database are loaded in the backend.

**Main Flow:**

1. The Reader enters a natural language query (e.g., "a novel about a family secret") into the "Please enter a description of a book:" textbox.
2. The Reader leaves the "Category" and "Emotional Tone" dropdowns at their default "All" setting.
3. The Reader clicks the "Find recommendations" button.
4. The System sends the query to the backend.
5. The System performs a semantic similarity search and retrieves the top-ranked books.
6. The System formats the results.
7. The System displays up to 16 recommended books in the gallery, showing their cover, title, author, and a short description.

**Postconditions:**

* The gallery is populated with books that are semantically relevant to the reader's query.

[Page 19]

### **6.2. Use Case 2: Search Refined by Genre**

Use Case ID: UC-02

Use Case Name: Refine a Search with a Genre Filter

Actor: Reader

Description: The reader wants to find books on a specific topic but only within a certain genre (e.g., Fiction).

**Preconditions:**

* The application is loaded and running.

**Main Flow:**

1. The Reader enters a query (e.g., "the history of computing") into the textbox.
2. The Reader clicks the "Select a category:" dropdown and selects "Non-fiction".
3. The Reader leaves the "Emotional Tone" dropdown at "All".
4. The Reader clicks the "Find recommendations" button.
5. The System performs the semantic search to get an initial list of relevant books.
6. The System filters this list, removing any books not categorized as "Non-fiction".
7. The System displays the top 16 remaining books in the gallery.

**Postconditions:**

* The gallery is populated only with non-fiction books that are semantically relevant to the history of computing.

[Page 20]

### **6.3. Use Case 3: Search Based on Emotional Tone**

Use Case ID: UC-03

Use Case Name: Find a Book Matching a Mood

Actor: Reader

Description: The reader wants to find a book that evokes a specific feeling or emotional response.

**Preconditions:**

* The application is loaded and running.

**Main Flow:**

1. The Reader enters a query (e.g., "a story about overcoming challenges") into the textbox.
2. The Reader leaves the "Category" dropdown at "All".
3. The Reader clicks the "Select an emotional tone:" dropdown and selects "Happy".
4. The Reader clicks the "Find recommendations" button.
5. The System performs the semantic search to get an initial list of relevant books.
6. The System re-ranks this list in descending order based on the 'joy' score associated with each book.
7. The System displays the top 16 re-ranked books in the gallery.

**Postconditions:**

* The gallery is populated with books relevant to the query, with the most uplifting and joyful stories appearing first.

[Page 21]

## **7. Stakeholder List**

This section identifies the key stakeholders involved in the Semantic Book Recommender project, outlining their roles and primary interests.

| **Stakeholder** | **Role / Title** | **Interest in the Project** |
| --- | --- | --- |
| **NextRead Corp. Executive Team** | CEO, CTO | **Primary Interest:** The overall success of the project as a proof-of-concept to secure future funding. Interested in the project meeting its strategic goals, timeline, and budget. |
| **Product Manager** | Head of Product | **Primary Interest:** Ensuring the final product meets the specified functional and non-functional requirements. Responsible for defining the feature set and ensuring it aligns with the business opportunity. |
| **ML Engineering Team** | Lead ML Engineer, Data Scientists | **Primary Interest:** The technical implementation of the system. Focused on model performance, data pipeline efficiency, code quality, and the successful integration of all ML components. |
| **UI/UX Designer** | (External Consultant) | **Primary Interest:** The usability and aesthetic appeal of the final Gradio application. Focused on ensuring the interface is intuitive, accessible, and provides a delightful user experience. |
| **Marketing Team** | Head of Marketing | **Primary Interest:** Using the deployed application as a key marketing asset to attract early adopters and showcase the company's innovative technology to potential investors and the media. |
| **End Users** | Readers, Book Enthusiasts | **Primary Interest:** A functional, fast, and effective tool for discovering books that they will enjoy reading. Their satisfaction is the ultimate measure of the project's success. |

[Page 22]

## **8. Glossary of Terms**

This glossary defines key technical terms used throughout this document to ensure a common understanding among all stakeholders.

| **Term** | **Definition** |
| --- | --- |
| **Client Requirement Specification (CRS)** | This document, which formally specifies the functional and non-functional requirements for the project from the client's perspective. |
| **Embedding (Vector Embedding)** | A numerical representation of a piece of text (a word, sentence, or document) in the form of a high-dimensional vector. The embedding captures the semantic meaning of the text. |
| **Semantic Search** | A search technique that determines the intent and contextual meaning of a user's query to return more relevant results than traditional keyword matching. It is powered by vector embeddings. |
| **Vector Database** | A specialized database designed to efficiently store, manage, and query high-dimensional vector embeddings. It uses algorithms like Approximate Nearest Neighbor (ANN) for fast similarity searches. |
| **Chroma DB** | An open-source vector database used in this project to store and search the book description embeddings. |
| **Large Language Model (LLM)** | A type of artificial intelligence model trained on vast amounts of text data to understand and generate human-like language. Examples include models from the GPT family and RoBERTa. |
| **Zero-Shot Classification** | An NLP technique where an LLM can classify text into predefined categories without having been explicitly trained on a labeled dataset for that specific task. |
| **Sentiment Analysis** | The process of computationally identifying and categorizing opinions or emotions expressed in a piece of text to determine whether the writer's attitude is positive, negative, or neutral, or to identify specific emotions like joy or fear. |
| **Gradio** | A Python library that allows for the rapid creation of customizable web-based user interfaces for machine learning models and data science workflows. |
| **API (Application Programming Interface)** | A set of rules and protocols that allows different software applications to communicate with each other. In this project, it refers to the OpenAI API used for generating embeddings. |
| **Minimum Viable Product (MVP)** | The version of a new product that allows a team to collect the maximum amount of validated learning about customers with the least effort. This project constitutes the core of the MVP for NextRead Corp. |

[Page 23]

[Page 24]

## **9. Document Approval**

This Client Requirement Specification document has been reviewed and agreed upon by the undersigned stakeholders. Any changes to these requirements must follow the project's formal change control process.

By signing below, the stakeholders acknowledge that they have read, understood, and approved the requirements detailed in this document.

**For NextRead Corp. (Client):**

**[Name], Product Manager**

**Date:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**[Name], Chief Technology Officer**

**Date:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**For [Your Company Name] (Development Team):**

**[Your Name], Project Lead**

\*\*Date