ShelfMate

Database Design & Chatbot Architecture

Capstone Project
2024/2025

Data Sources

Database Design

User Stories for Features

Architecture Diagrams

Adriana Pinto | 20221921
Inês Mendes | 20211624
Maria Teresa Silva | 20221821
Miguel Nascimento | 20221876
Marta Alves | 20221890
David Duarte | 20221899

INDEX

Data Sources and Database Design

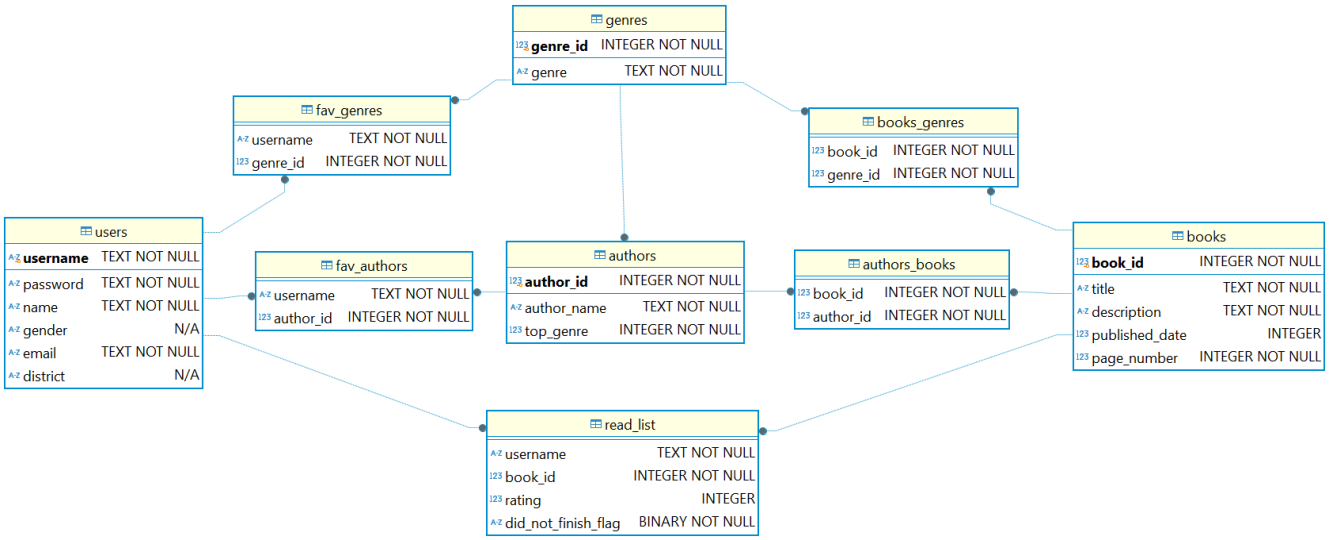
1. Data Sources

- Amazon Books Review/ Books Data
  Provided data for *books, books_genres, genres, authors_books, authors* tables.
  [https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews/data?select=books_data.cs]

- Chatbot users, generated artificially using gpt-3.5-turbo (OpenAI API).
- Users' favorite genres and authors, given the available genres in the database, are generated artificially using gpt-3.5-turbo (OpenAI API).
- Users' read lists, given the available books and users, generated artificially using gpt-3.5-turbo (OpenAI API).
- Books' number of pages generated artificially using a normal distribution, after failed attempts using OpenAI API.
- company_pdf PDF file, humanly written with information about Shelfmate company.
- features_pdf PDF file, humanly written with information about the chatbot features.
- libraries_per_district PDF file, generated artificially using gpt-4o-mini and assembled with FPDF.

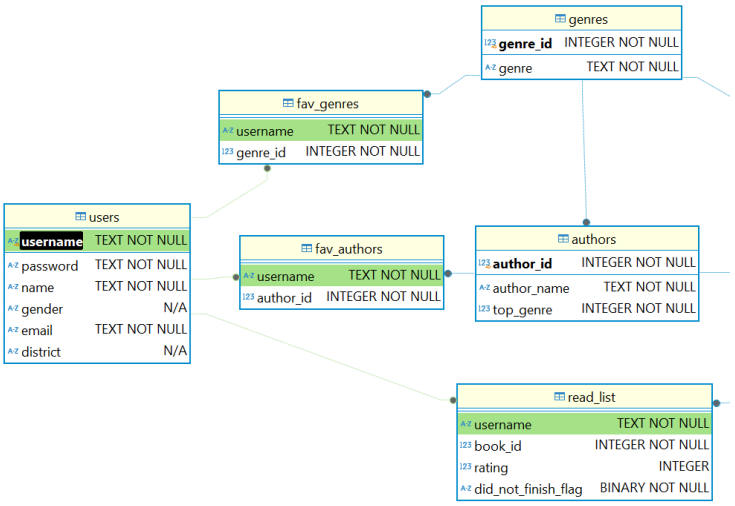
2. Database Design

The database schema for our chatbot consists of 9 tables and was created automatically with DBeaver by importing the .db file obtained from the sqlite script.



Primary keys are identified in bold and foreign keys were defined on the script and are as follows:

fav_genres(username), favorite_authors(username) and read_list(username) reference users(username).
fav_genres(genre_id), authors(top_genre), books_genres(genre_id) reference genres(genre_id).
fav_authors(author_id), authors_books(author_id) reference authors(author_id).
books_genres(book_id), authors_books(book_id), read_list(book_id) reference books(book_id).



Example of foreign keys of username in DBeaver.

Relationships
By using junction/bridge tables to solve cases with books with multiple authors or genres, or allowing users to have more than one favorite author or genre, every relationship in this schema is a One-to-Many relationship, instead of having Many-to-Many relationships.

Tables

Users: The users table is essential for a personalized experience, secure authentication, and understanding user preferences. This includes storing details like username, password, gender, email, and district, which helps tailor content and recommendations.

Genres: The genres table is a reference table that matches an id to all available book genres, serving as a baseline to other 3 tables. It plays a key role in content organization and search functionality.

Books: The books table is at the core of the database, storing information like title, description, publication date, and page number (the latter generated artificially). It facilitates efficient book discovery and enables analysis of the book catalog.

Authors: This table contains information about authors, including their names and their primary genre (extracted from the books dataset), supporting better content categorization and author-specific recommendations for users.

Favorite Authors: The favorite authors table links users to their preferred authors, helping track user interests and enabling personalized content suggestions based on author preferences.

Favorite Genres: This table associates users with their preferred genres, allowing the system to recommend books and authors that match their interests and enhance their overall experience.

Books_Genres: Books are linked to their respective genres through this table (supporting multi-genre books), ensuring accurate categorization and making it easier to search for books within a specific genre.

Authors_Books: This table connects authors with the books they have written (supporting multi-author books), helping track bibliographies and supporting author-specific searches and recommendations.

Read List: The read list tracks the books a user has interacted with, storing data on user ratings and whether a book was finished. This enables the system to improve recommendation algorithms and understand user behavior.

Chatbot Architecture

1. User stories for each feature

1.1. Update profile information
“As a user,
I want to update my profile information through the chatbot (username, password, name, gender, email, district),
So that I receive personalized recommendations.”

1.2. Insert new favorite author or genre
“As a user,
I want to add new favorite authors or genres to my profile,
So that the chatbot can refine recommendations to suit my evolving tastes.”

1.3. Add a book to my read list
“As a user,
I want to add books to my “Read List” and rate them via the chatbot,
So that I can keep track of what I’ve read and personalize my suggestions.”

1.4. Suggest new authors given my favorite books, genres or authors
“As a user,
I want the chatbot to suggest new authors based on my favorite books, genres, or authors,
So that I can discover new reading material aligned with my tastes.”

1.5. Suggest new authors given a book, author or gender given by user input
“As a user,
I want to receive author suggestions based on a book, author, or genre I provide to the chatbot,
So that I can explore similar writers to the ones I enjoy or want to explore.”

1.6. Suggest new books given my favorite books, genres or authors
“As a user,
I want the chatbot to recommend new books based on my favorite books, genres, or authors,
So that I can discover more books tailored to my interests.”

1.7. Suggest new books given a book, author or gender given by user input
“As a user,
I want to receive book suggestions based on a book, author, or genre I provide to the chatbot,
So that I can quickly find books related to my current interests.”

1.8. Suggest new books given a user input trope
“As a user,
I want the chatbot to suggest books based on a specific trope I mention (e.g., enemies-to-lovers, found family),
So that I can find books with themes or storylines I enjoy.”

1.9. Browse available genres, books of a specific genre and authors with a predominant genre
“As a user,
I want to browse available genres, view books of a specific genre, and discover authors with a predominant genre via the chatbot,
So that I can explore books and authors that match my interests and preferences.”

1.10. Create a monthly or annual plan for readings
“As a user,
I want to create a monthly or yearly reading plan based on my favorite books, genres, or authors or specific inputs,
So that I can organize my reading goals and stay motivated.”

1.11. Ask about recommended bookstores per district
“As a user,
I want to ask questions about recommended bookstores per district
So that I can find the best libraries in each district.”

1.12. Ask about chatbot features
“As a user,
I want to ask questions about the features of the chatbot
I can understand its capabilities and how it can assist me effectively.”

1.13. Ask about company information
“As a user,
I want to ask questions about the company,
So that I can learn about its mission, history, values and vision.”

2. Architecture diagrams

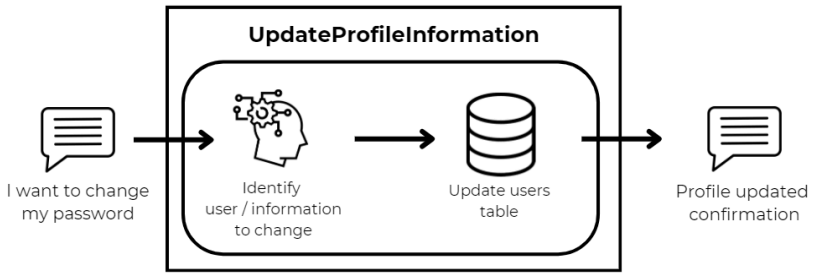
Note 1: When *"compute similarity between books"* is mentioned, it means that the descriptions of those books are computationally compared in order to assess their degree of similarity.

Note 2: The diagrams do not represent some checks that will be addressed to prevent some input overload from the user (inputting 50 books to suggest a new book, per example) that can increase exponentially the amount of computation power and time needed. Besides that, more prompt injection risks will be taken into consideration when building the chatbot based on these diagrams, such as defining forbidden words in the user input like "DELETE", that can undermine the chatbot database.

2.1. Update Profile Information

Goal: Allow users to update their profile information (password, name, gender, email, district) through the chatbot to receive personalized recommendations.

Implementation: Implement a process where the chatbot retrieves the user's current profile information, allows the user to update details, and then stores the updated information in the database. Return a confirmation message in string format.

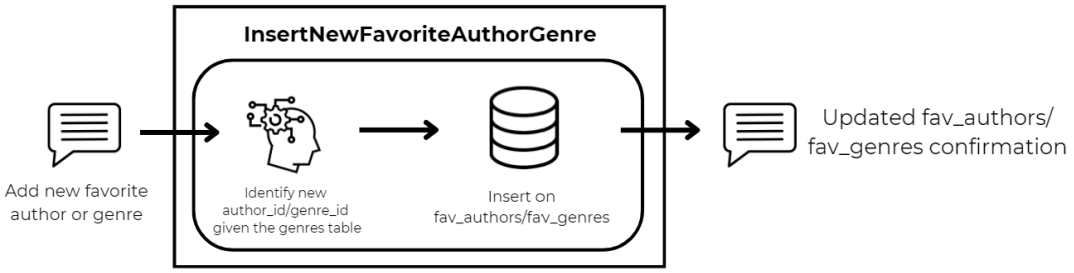


Note: If the user wants to alter its username, we should first verify if that username already exists or not, so they can proceed with the alteration.

2.2. Insert new favorite author or genre

Goal: Expand the user's profile by adding new favorite authors or genres for improved personalization of recommendations.

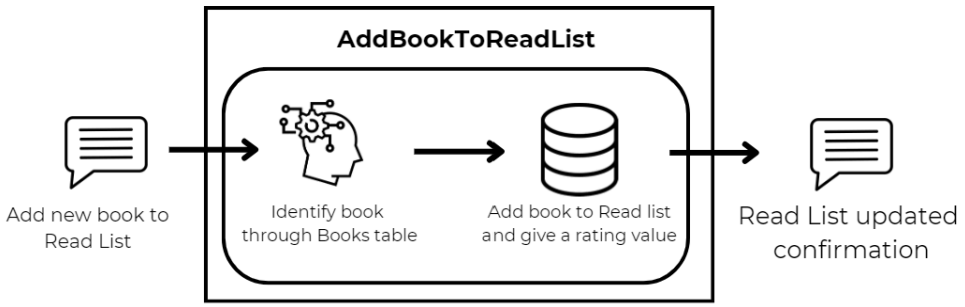
Implementation: Provide an option in the chatbot for users to input additional authors or genres to their favorite lists. Update the user's profile in the database and use this information to refine future recommendations.



2.3. Add a book to my read list

Goal: Enable users to mark books as “read” and optionally rate them to track progress and refine suggestions.

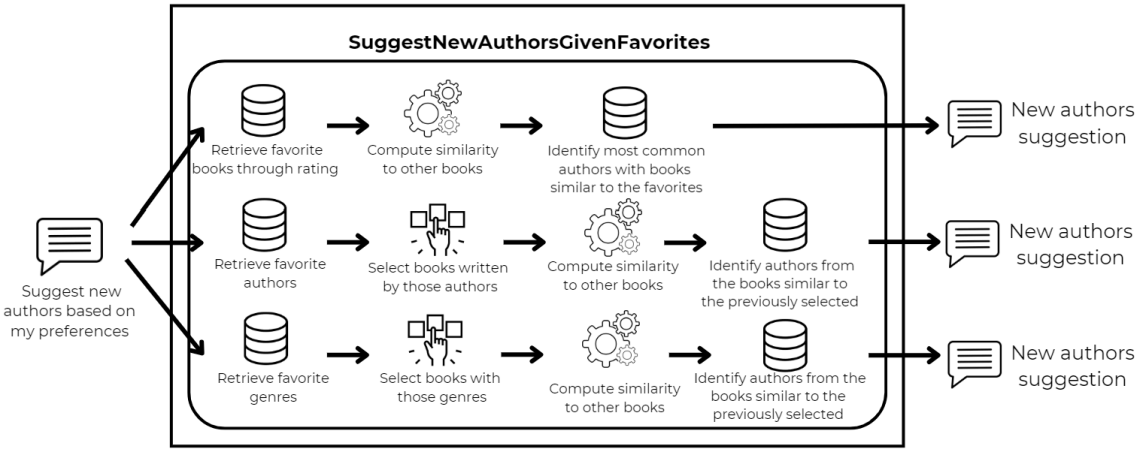
Implementation: Implement a feature where users can add books to their "Read List" via a chatbot command and allow optional star rating. Store this data in the database and adjust recommendations accordingly.



2.4. Suggest new authors given my favorite books, genres or authors

Goal: Suggest new authors based on the user’s favorite books, genres, or authors to improve users' discovery.

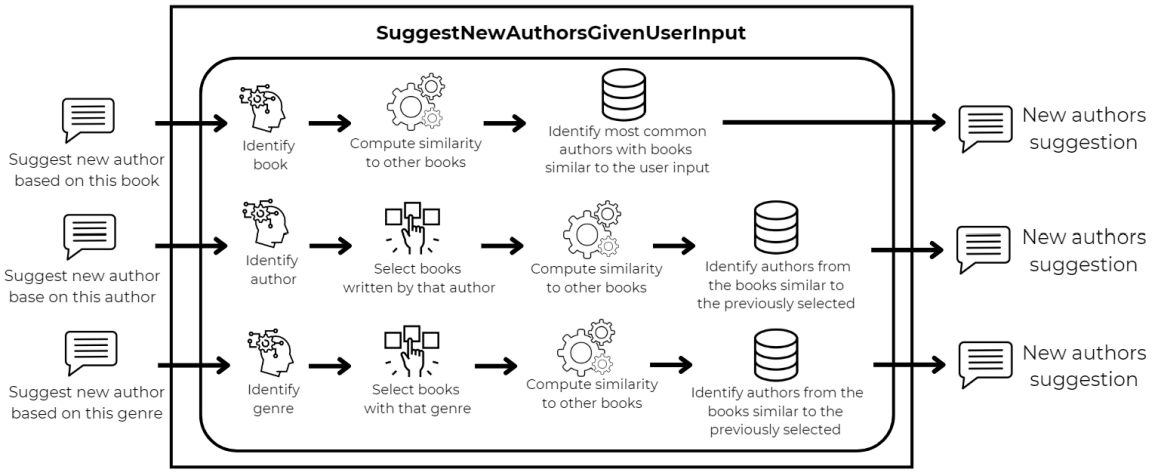
Implementation: We will implement a chain that retrieves new authors from the database based on the user's favorite books, genres, or authors and presents the suggestions in a list format.



2.5. Suggest new authors given a book, author or gender given by user input

Goal: Give author suggestions based on a book, author, or genre the user provides, so they can explore related writers.

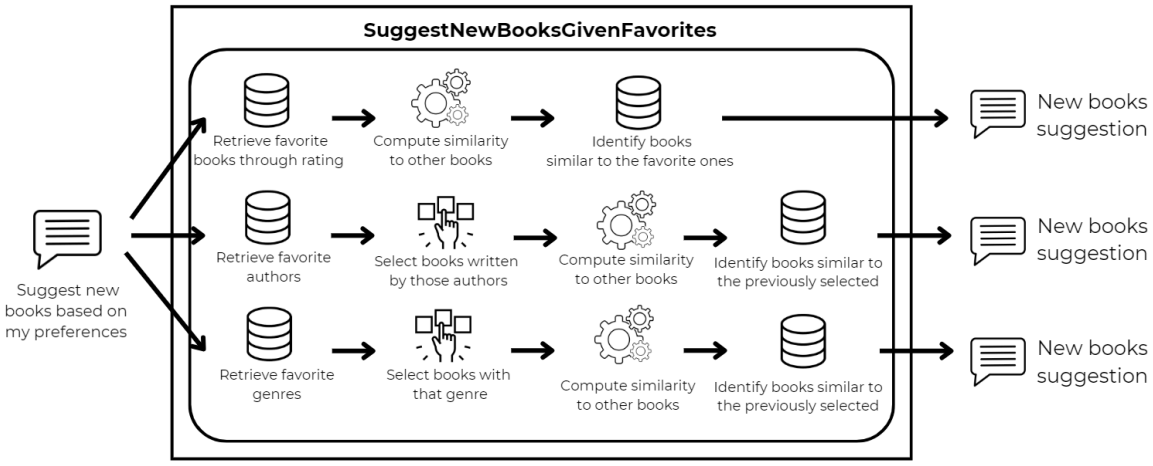
Implementation: We will implement a chain that retrieves new authors from the database based on the user's input (book, genre or author) and presents the suggestions in a list format.



2.6. Suggest new books given my favorite books, genres or authors

Goal: Provide personalized book recommendations tailored to the user's favorite books, genres, or authors.

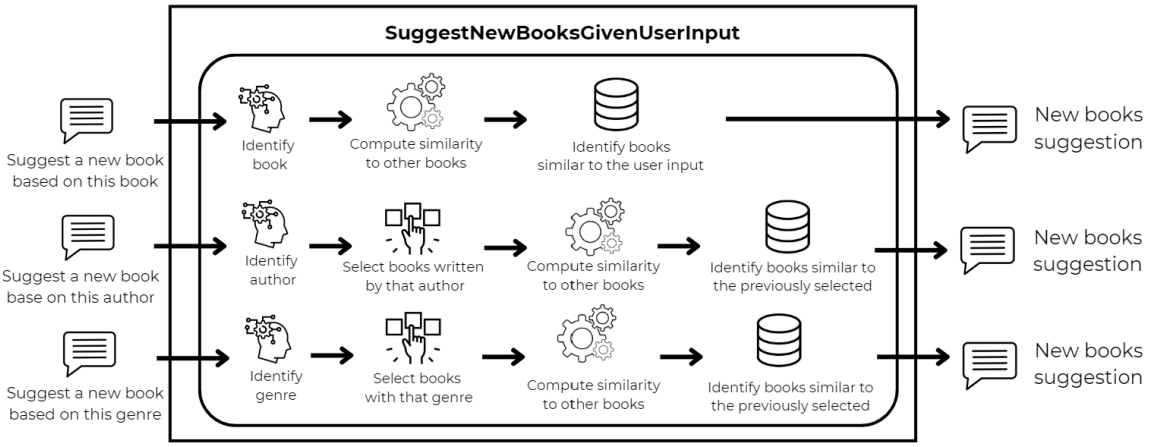
Implementation: Build a recommendation engine that analyzes the user's favorite books, genres, or authors and selects books with similar attributes. Presents the suggestions in a list format with the book title, author, and a brief description.



2.7. Suggest new books given a book, author or genre given by user input

Goal: Enable users to find new books by providing the chatbot with a book, author, or genre they enjoy.

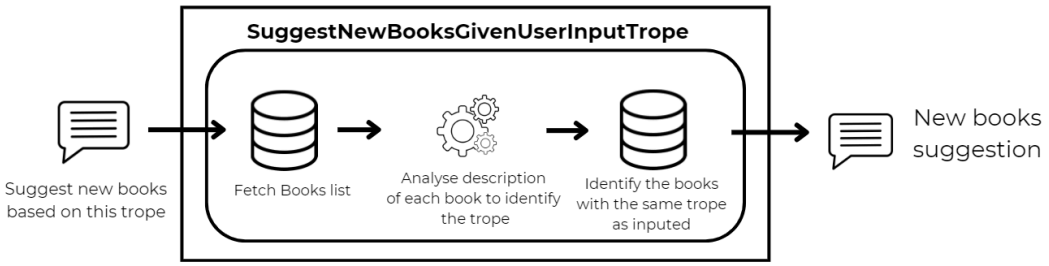
Implementation: Implement a chain that retrieves new books from the database based on the user's input (book, genre or author) and presents the suggestions in a list format with the book title, author, and a brief description.



2.8. Suggest new books given a user input trope

Goal: Allow users to find books that include specific tropes or themes (e.g., enemies-to-lovers, found family).

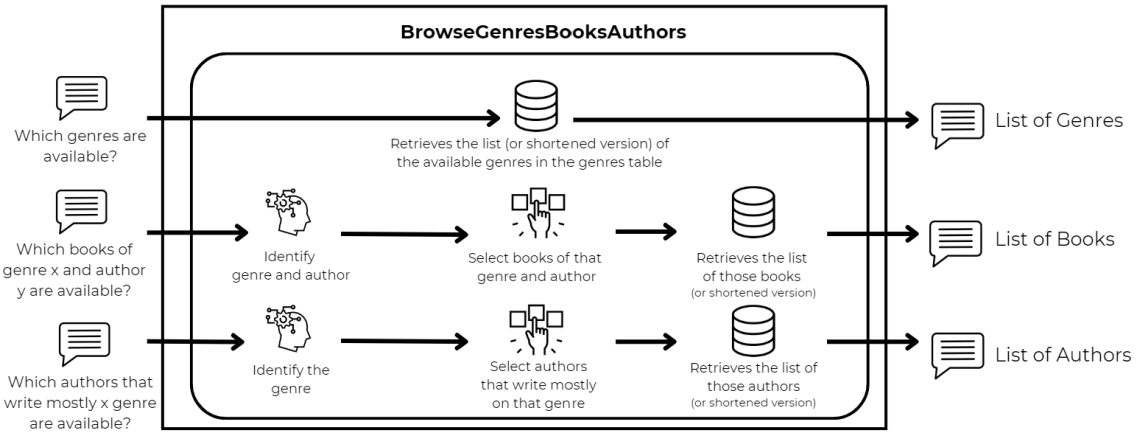
Implementation: Implementation of a chain that retrieves new books based on the trope the user imputed. This is made by analyzing the book's descriptions and identification of the tropes. Provide a list of the final identified books.



2.9. Browse available genres, books of a specific genre and authors with certain predominant genre

Goal: Allow users to browse available genres, books within specific genres, and authors associated with a predominant genre.

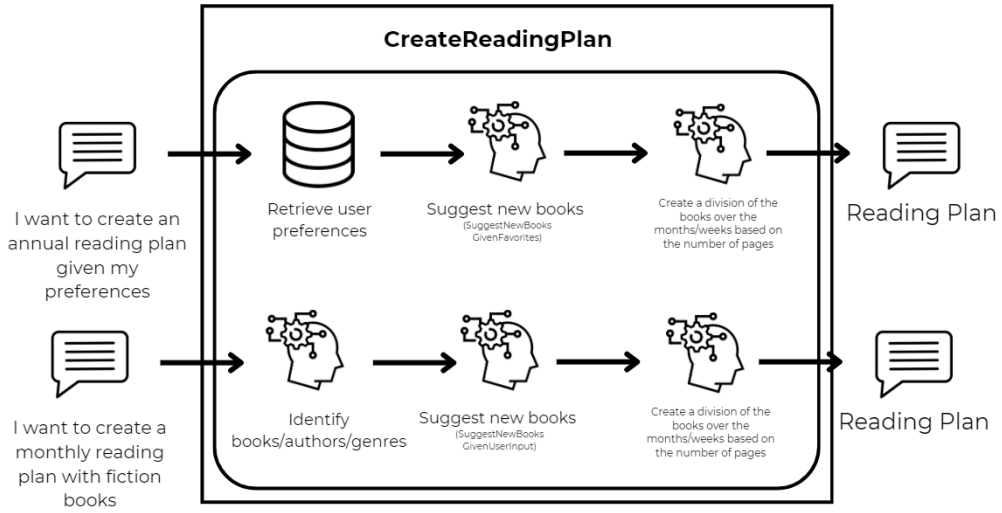
Implementation: Create a database query feature that lists all available genres in the database. When a genre is selected, display books and associated authors to that genre.



2.10. Create a monthly or annual plan for readings

Goal: Allows users to create personalized reading plans for a month or year based on specific inputs or preferences.

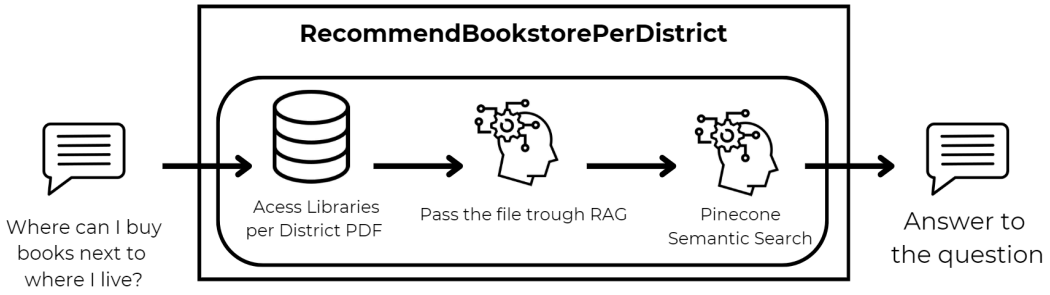
Implementation: Build a system that generates a reading plan based on user-provided inputs like favorite books, genres, or authors, based on the number of pages of each book. Display a timeline with recommended books for each month. Allow users to adjust the plan.



2.11. Ask about recommended bookstore per district

Goal: Recommend the bookstores per district according to the localization of the user.

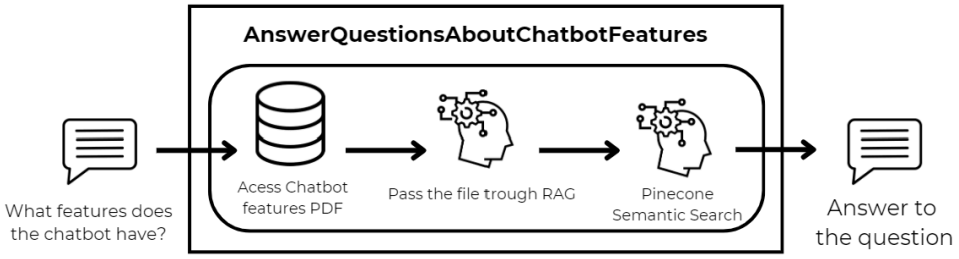
Implementation: Allow the user to ask about bookstores nearby and the is able to recommend book stores according to the district of the user saved in the user profile. This is possible due to the PDF with these generated data and the use of RAG.



2.12. Ask information about the chatbot features

Goal: Provide users with insights of the chatbot features from the PDF file without requiring them to read the entire document.

Implementation: Allow users to ask targeted questions, and use keyword searching to locate and return relevant sections of the PDF document in chatbot responses through RAG.



2.13. Ask about company information

Goal: Enable users to ask the chatbot about company-related information like mission, history, values, and vision.

Implementation: Allow users to ask targeted questions about the company, and use keyword searching to locate and return relevant sections of the PDF document through RAG.