

project ： Library management system

curriculum ：Data structure course design

Student's name: Huang Zigeng

Dong Wentao

Han Ruiyang

Shao Yingbo

He Pengyu

Student number: 2023311640

2023310770

2023310988

2023311023

2023311170

Faculty advisor: Lilia SFAXI

**abstract**

**I. Project description**

### As the library expands, the traditional manual management model has become inefficient and is unable to meet users 'needs for book searches, borrowing, and personalized recommendations. This project integrates knowledge from data structures courses to design a library management system that achieves digital management of books and users. By optimizing query performance using efficient data structures like B-trees, and by analyzing user borrowing behaviors to provide personalized recommendations, the system aims to enhance the library's service experience and deepen the understanding of data structures in practical applications.

**II.Project requirements analysis**

**1. Functional requirements:**

### User management: Support adding users, recording basic information (user ID, name), borrowing history and book type preference of users, and realizing the addition of borrowing records and dynamic update of preferences.

### Book management: support the addition and deletion of books, manage book information (book ID, title, type, etc.), and realize quick query of books through index.

### Borrowing and returning processing: record the borrowing behavior of users (update the status of books as unavailable, add the borrowing history of users, improve the preference of users for corresponding types) and return behavior (update the status of books as available).

### Book recommendation: Based on the user's historical borrowing preference, recommend books that meet the user's interest.

### GUI interface interaction: provide login interface, book list display, user operation entrance and other visual interactive interfaces, support file import and export functions.

**2. Program running requirements**

### Ease of use: the interface is simple and intuitive, the operation process is clear, suitable for different user groups.

### Efficiency: The book query speed is optimized through data structures such as B-tree index to ensure that the system can still run efficiently when the data volume is large.

### Stability: It has an exception handling mechanism, which can give hints and run normally when the user operation is wrong or the data is abnormal.

**III. item design**

**3.1 Module design**

### Recommendation service module: namely RecommendationService class, which is responsible for integrating book and user data, processing the impact of borrowing and returning behaviors on data (such as updating genre\_stats, user.preferences), and providing data support for recommendation function.

### GUI module: implemented based on tkinter and ttk, including login interface (\_show\_login\_screen method) and main interface (LibraryApp class). It implements file selection through filedialog and messagebox to realize message prompt, and is responsible for visual interaction between users and the system.

**Why GUI is needed:**

### Target users: librarians, general readers

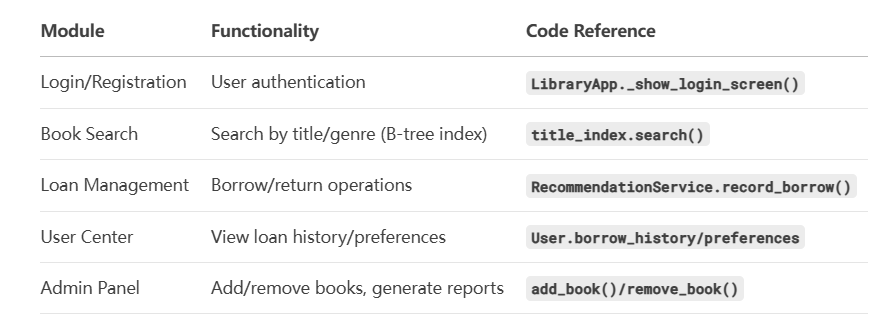
### Core requirements:

### Librarian: Efficient management of books/users/borrowing records

### Reader: Quick search for books, view personal borrowing history, get recommendations

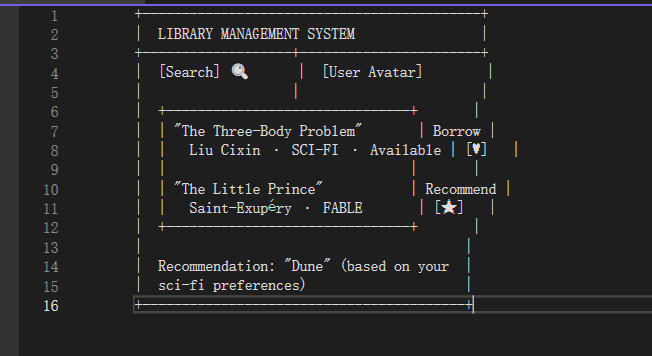
**What kind of GUI is needed:**

### 1. Core functional modules:



##### 2. GUI interface layout scheme:deepseek_mermaid_20250704_8d50ef

### 3. Prototype design diagram:



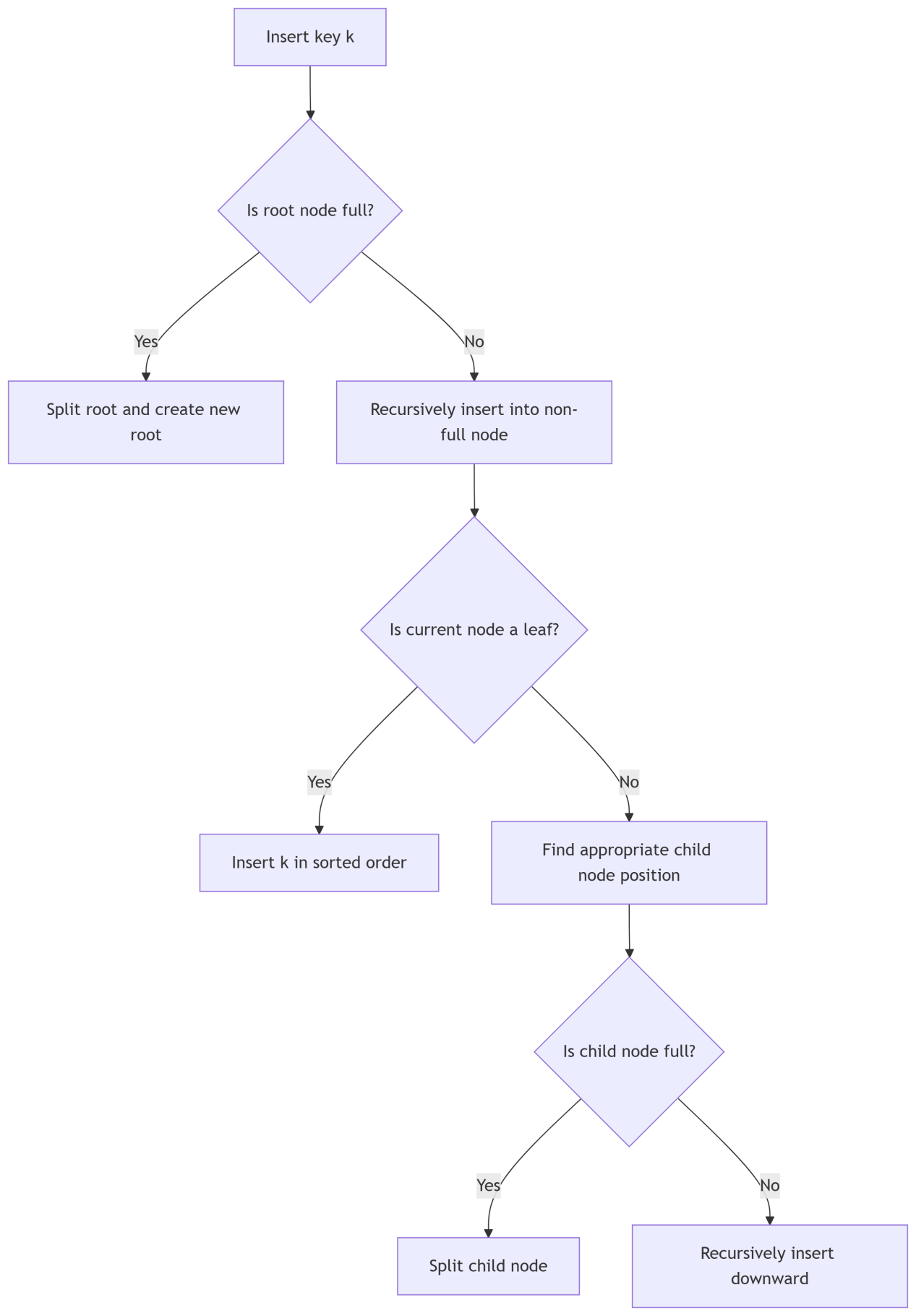
### Data management module: The User class and Book class (hidden in the code) are used to manage user and book data, and dictionaries (book\_data and user\_data) are used to store book and user information to realize fast access to data.

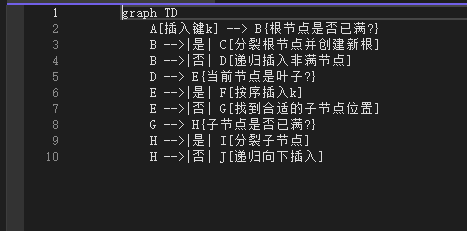
### File processing module: Based on csv and chardet libraries, it implements the import and export of book or user data. Chardet is used to detect file encoding to ensure correct reading of data.

**3.2 Data structure design**

### B tree: In RecommendationService, title\_index is a B tree (t=3), which is used to index book titles. Through the insertion and deletion operations of B tree, efficient queries of book titles can be realized, especially suitable for retrieval scenarios with large amounts of book data.

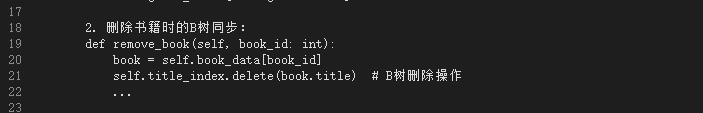
**B-Tree construction thinking diagram:**

****

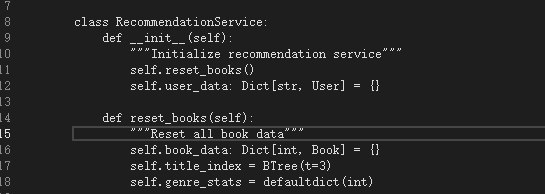


### 1. B tree update when inserting: books:

### 2. B-tree synchronization when deleting books:



### Dictionary (Dict): book\_data(int → Book) stores the mapping between book ID and book object, user\_data(str → User) stores the mapping between user ID and user object, genre\_stats(str → int) counts the number of books of each type, and uses the O (1) search complexity of dictionary to improve data access efficiency.



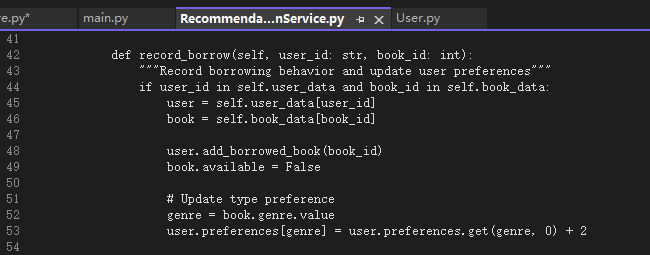
### (Note: B-tree (title\_index) is used to realize efficient indexing of book titles, and dictionary (book\_data, user\_data) is used to realize fast data access.)

### List (List): borrow\_history in User class is used to record the book ID borrowed by users. It adopts list implementation for sequential storage, which facilitates appending and traversing the history records.

### Enum: The Genre enum defines the book type (FICTION, ROMANCE, etc.) and unifies the representation of the book type to avoid confusion of the type value.

**3.3 Core algorithm design**

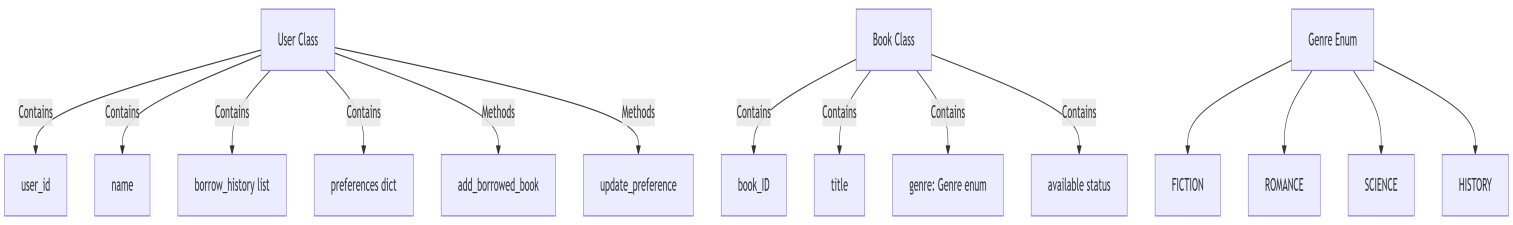
### Recommendation algorithm: Content recommendation based on user preferences. Record the preference score of users for book types (each borrowing corresponds to +2 points for each type) through the method record\_borrow, and prioritize the book types with high preference score when recommending, and select available books from them to recommend to users, so as to realize personalized recommendation.



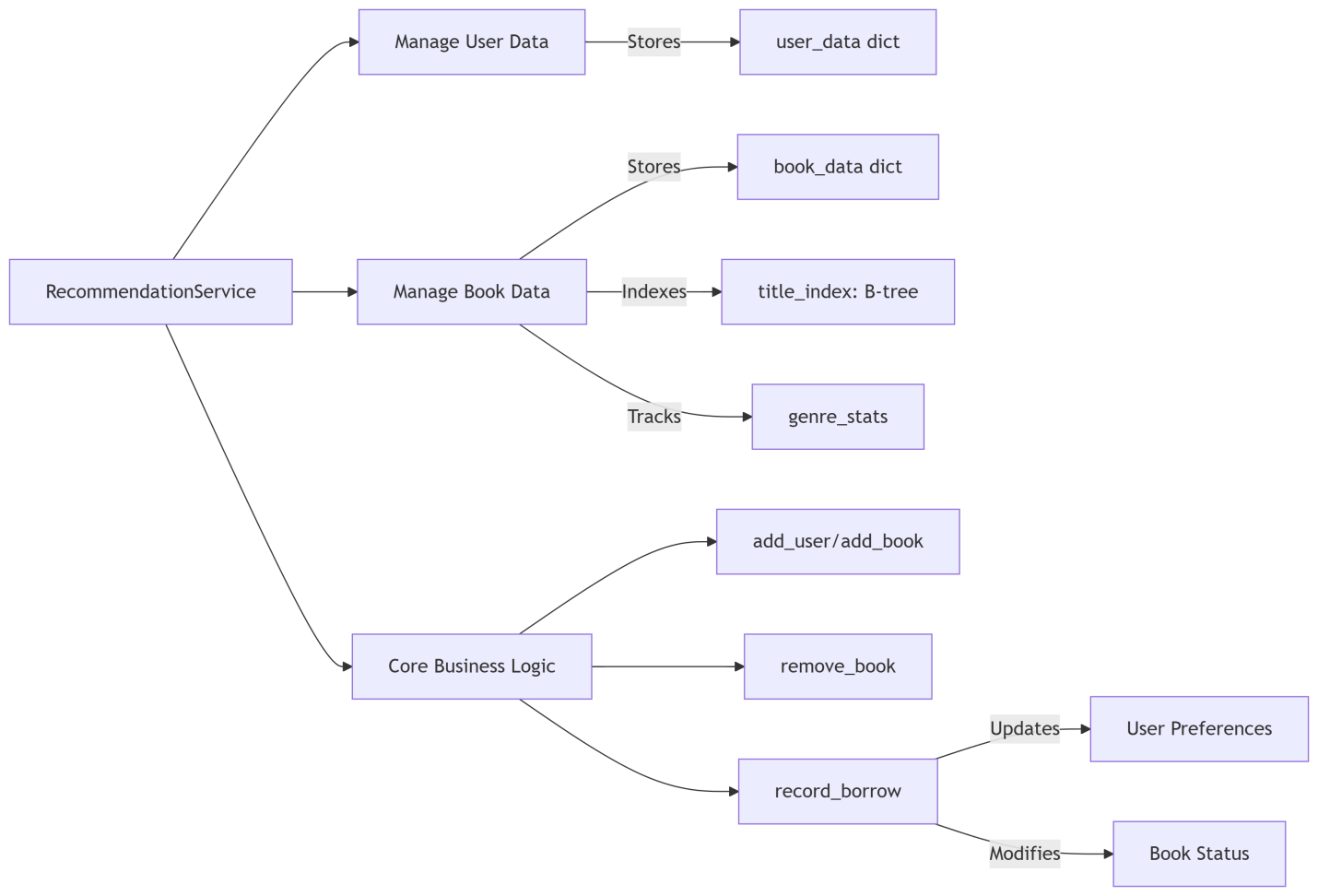
### (Note: User preferences are dynamically updated through borrowing behavior to provide a data basis for personalized recommendations.)

**3.4 Flowchart:**

### 1. Data layer (Models)

****

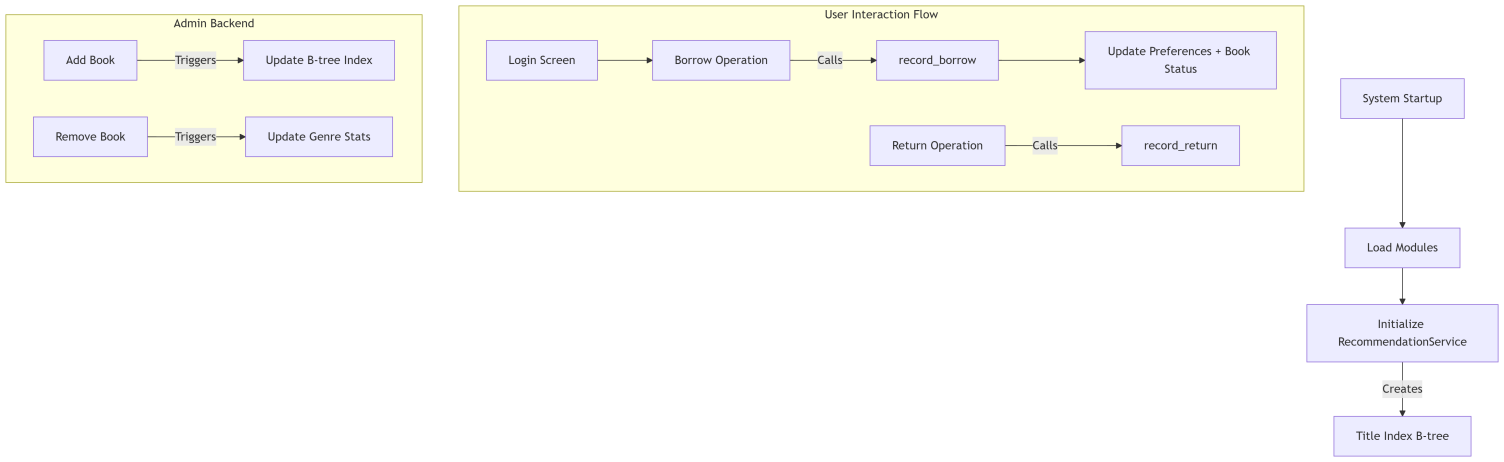
### 2. Business logic layer (Services)

****

### 3. presentation layer （GUI）

#### **deepseek_mermaid_20250704_6d6ee7**

### 4. System process

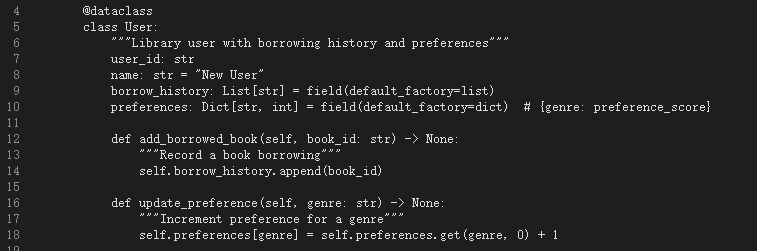


#### 5. Key data flows**deepseek_mermaid_20250704_d74bd2**

**IV. project implementation**

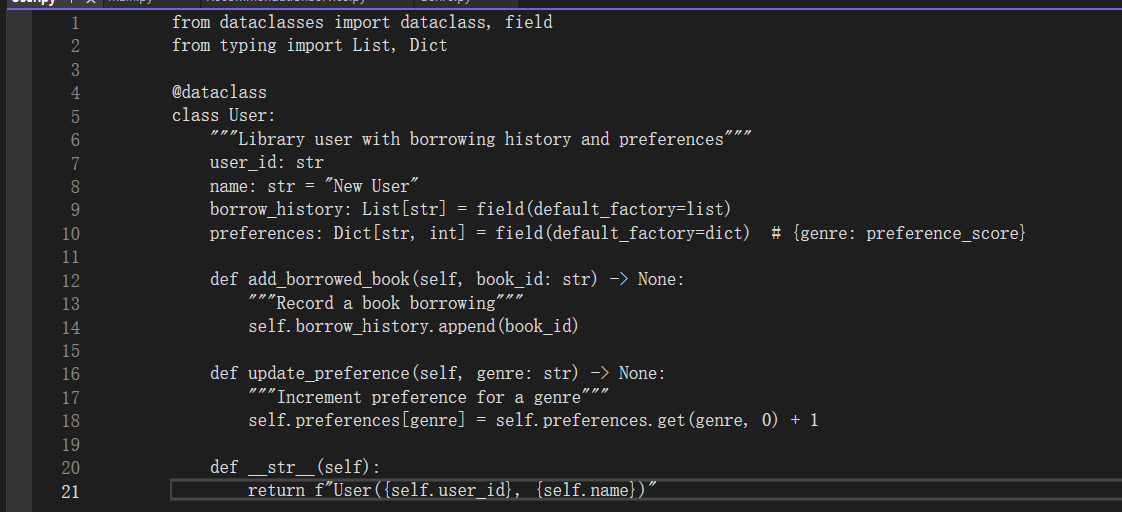
**4.1 Environment Building:**

### Determine the development environment to be Python, install the necessary libraries (tkinter is the standard library of Python, chardet needs to be installed additionally), and configure the project directory structure (including gui, services and other sub-module folders).

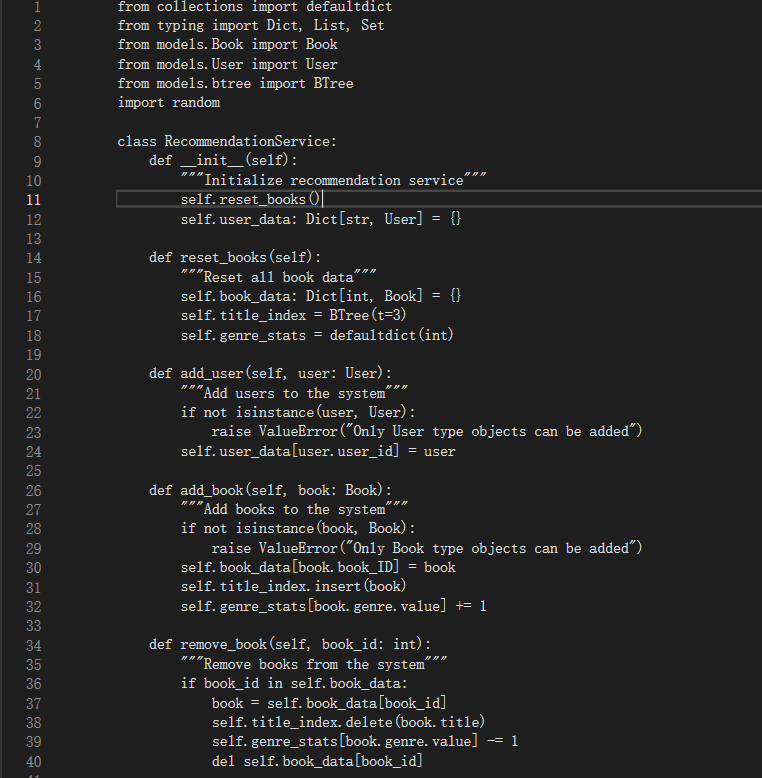
**4.2. Module** development:

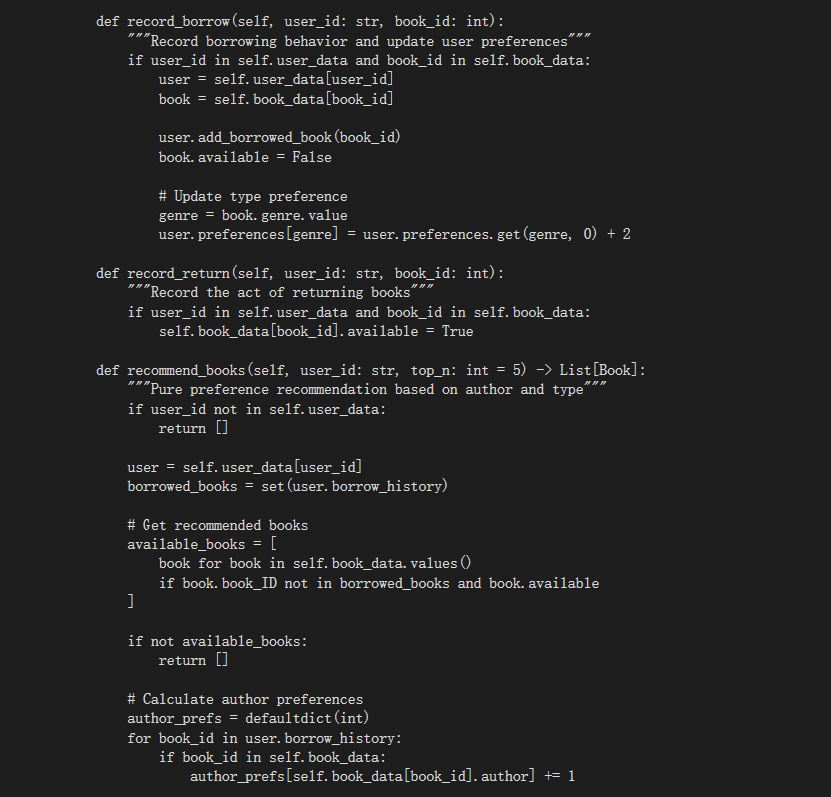
### (Note: Use dataclass to simplify the implementation of user classes, automatically generate initialization and string representation methods, and build data models)

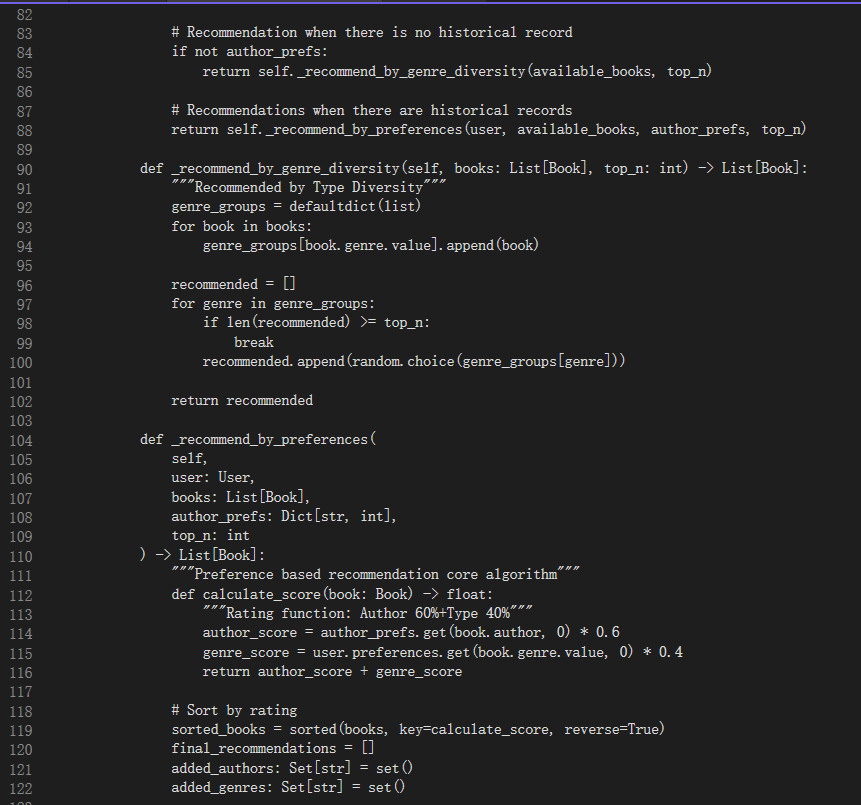
### First, implement the basic data class: define the Genre enumeration to determine the book type, and design the User class to encapsulate user data and operation methods (add\_borrowed\_book, update\_preference).

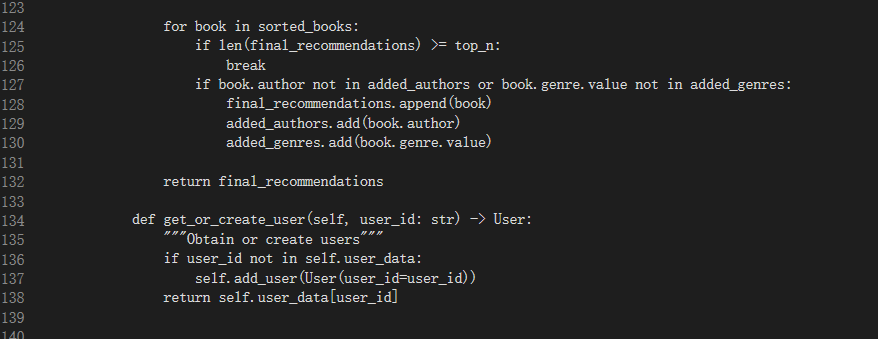


### Develop recommendation service module: implement RecommendationService class, complete the addition and deletion of books and users, borrowing and returning records logic, and integrate B tree as title index.

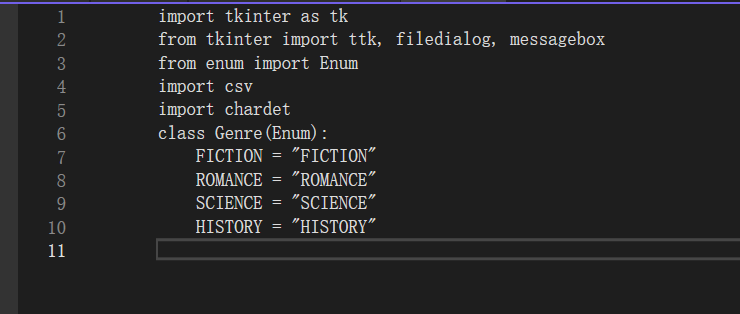




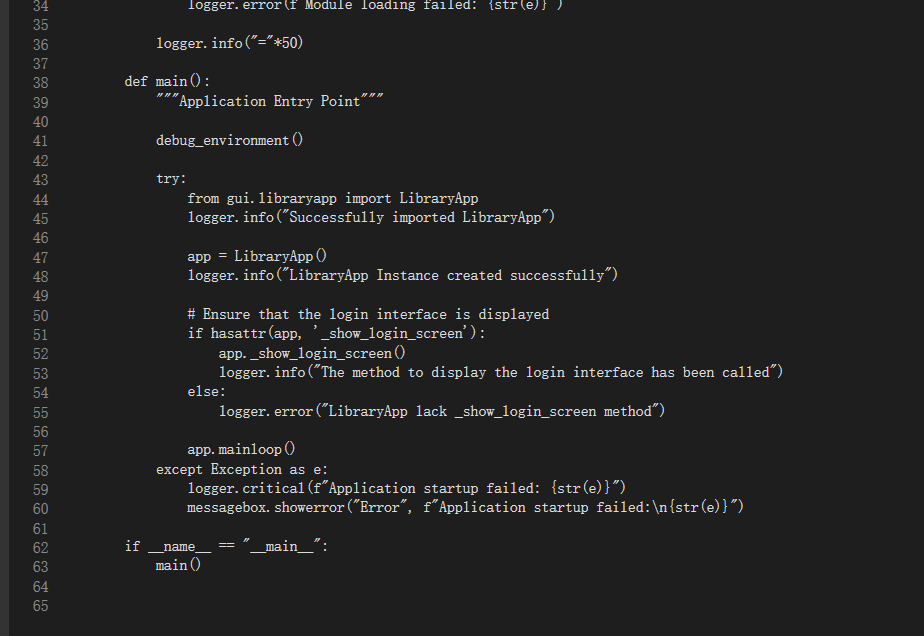
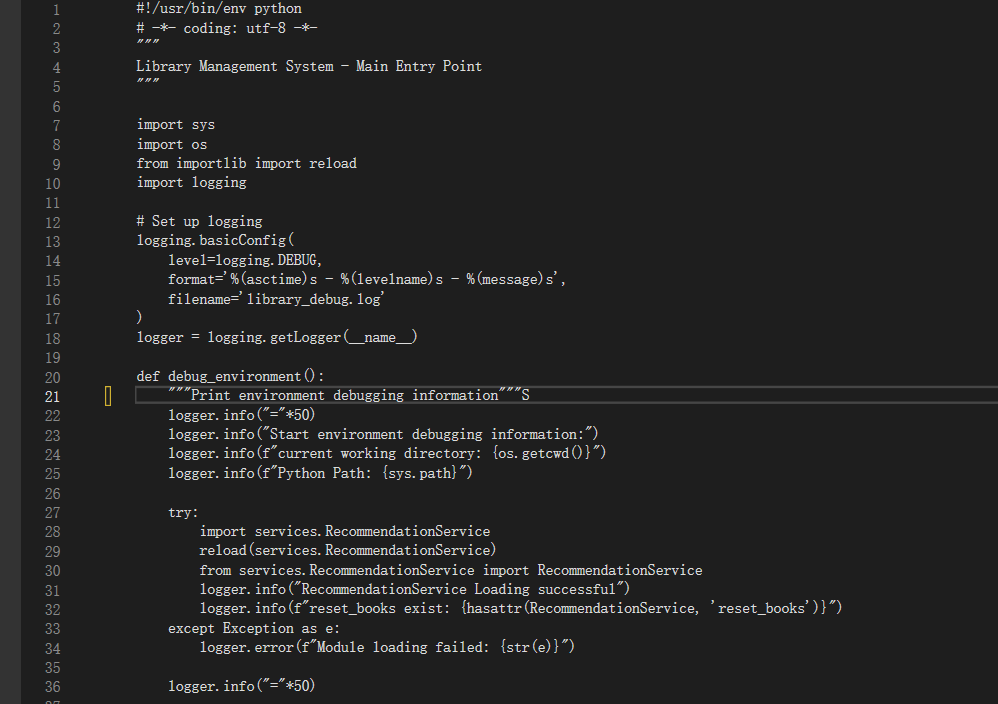




### Develop GUI module: design LibraryApp class based on tkinter, realize the layout of login interface and main interface, associate the functions of recommendation service module, and realize the visual response of user operation.



.



### Integrate file processing functions: Add a file selection button in the GUI, import and export data through the csv library, and use chardet to detect file encoding to ensure compatibility.

**4.3 Module integration:**

### The system startup logic is implemented in the main function, and the module loading status is detected through the debug\_environment function to ensure that the calls between modules are normal (such as the interaction between LibraryApp and RecommendationService).

**V.Project operation and commissioning**

**5.1 Operating environment**

### Python version: 3.x (compatible with libraries such as tkinter, enum)

### Depends on library: chardet (installed by pip install chardet)

### Operating system: Windows/macOS (supporting tkinter interface display)

**5.2 Debugging process**

### Module import problem: Record the Python path and current working directory through the debug\_environment function, troubleshoot the import errors of services, gui and other modules, and ensure that the directory structure meets the requirements of package import.

### B Tree operation exception: When adding or deleting books, if the title index error occurs, the key steps of B tree insertion and deletion are recorded through the log (library\_debug.log), and the correctness of B tree node splitting and merging logic is checked.

### GUI interface display problem: If the login interface is not displayed properly, check whether the method exists through hasattr (app, "\_show\_login\_screen") to ensure that the corresponding method in the GUI class is correctly defined.

### Data consistency problem: After borrowing or returning, check whether book.available status, user.borrow\_history and genre\_stats are updated synchronously, record key data changes through logs, and locate the cause of data inconsistency.

**VI. Summary**

**6.1 Daily work arrangement**

### Day 1-2: Demand analysis and module design, determine the data structure selection (B tree, dictionary, etc.) and the functional boundary of each module.

### Days 3-5: Core class development, including the implementation of User, Genre and RecommendationService, and the basic operations of data structures.

### Day 6-7: GUI module development and integration, realize the association between interface and business logic, add file processing function.

### Day 8-9: System debugging and optimization, troubleshooting through logs, optimizing the efficiency of B-tree query and interface response speed.

### Day 10: Project document sorting and summary, and complete the course design report as required.

**6.2 Internal division of labor in the group**

### Zhuang Zigeng: Responsible for core development such as GUI interface, implementing login and book management interfaces based on tkinter, and associating back-end functions. Responsible for the construction of B-tree and optimizing the query efficiency of B-tree and the response speed of the interface.

### Responsible for data structure design and implementation, including B-tree and dictionary applications,

### Han Ruiyang: Responsible for the development of RecommendationService module. Responsible for the implementation of User class enumeration and data management logic. Responsible for file processing and debugging. Responsible for the association between interface and business logic

### Dong Wentao: Responsible for designing the user and book data models. Implemented the Genre enumeration and data management logic. Developed the import/export of csv files, recorded debugging logs, and resolved exception issues. Organized and summarized project documents, and completed the course design report for our team as required.

### He Pengyu: Assist in the implementation of B-tree maintenance, verify the correctness of B-tree insertion/search, simulate the concurrent book borrowing scenario. Responsible for the integration of modules. Responsible for completing the ppt of the group defense.

### Shao Yingbo: Coordinate the project schedule, assist in optimizing the serialization of User/Book classes. Responsible for debugging and beautifying the tkinter interface. Responsible for following up the daily progress. Responsible for the defense task of this team.

**6.3 Tips (What everyone thinks)**

### Huang Zige: Through this design, I deeply realized the advantages of B-tree in large-scale data query. The combination of theoretical knowledge and practical application made me have a more thorough understanding of data structure. I deeply realized the efficient search performance of B-tree, its suitability for disk storage, self-balancing characteristics and other features.

### Han Ruiyang: When designing the User class, I used dataclass to simplify the code. At the same time, I realized that reasonable default value setting (such as field (default\_factory=list)) could avoid data sharing problems. Details determine system stability.

### Dong Wentao: By documenting debugging logs and resolving exceptions, I have developed the critical ability to pinpoint issues. Through implementing the Genre enumeration class and the User/Book data model, I gained a deeper understanding of how to use @dataclass and type annotations to construct robust data structures. I deeply appreciate that 'good data design is the skeleton of the system.'

### Shao Yingbo: tkinter interface development needs to take into account both beauty and function. When debugging the layout of the interface, I need to patiently adjust the attributes of components, which makes me understand the importance of user experience design, but also deeply realize the importance of teamwork.

### He Pengyu: File encoding problems often lead to data reading errors. The use of chardet library is very critical. Meanwhile, log recording plays a significant role in debugging and is an important tool for troubleshooting.

**6.4 Problems encountered and solutions**

### Question 1: When inserting book titles into B tree, the index is repeated. Solution: Add a judgment in the add\_book method to ensure that the book ID is unique and avoid repeated insertion of the same book to cause confusion in the index.

### Problem 2: The recommendation results do not change in real time after the user preference is updated. Solution: Check the record\_borrow method and find that the preference score calculation is wrong. Correct it to "user.preferences [genre] = user.preferences.get (genre,0) + 2" to ensure that the preference value is correctly accumulated.

### Problem 3: The book list does not display in the GUI interface. Solution: Through log troubleshooting, it is found that the data is not correctly loaded from book\_data. The data binding logic is corrected to ensure that the interface and the back-end data are synchronized.

### Question 4: Chinese characters are garbled when importing csv files. Solution: Use chardet to detect the file encoding (such as GBK), and specify the corresponding encoding when reading to solve the problem of garbled characters.

**catalogue**

I. Project description

2. Project demand analysis

3. Project design

If the algorithm is involved, write the relevant content of the algorithm here

Iv. Project implementation

V. Project operation and commissioning

VI. Summary

6.1 Daily work arrangement

6.2 Internal division of labor in the group

6.3 Tips (What everyone thinks)

6.4 Problems encountered and solutions

PS: Don't attach a lot of code

**Standard text format**

**The first level title font is Microsoft YaHei, small 4, bold**

**The subtitle font is Microsoft YaHei, size 5, bold**

The main font is Times New Roman and the size is 5.

The chart is in Times New Roman font and size 10.

The full text is 1.5 times longer