International Journal of Multiphysics
Volume 18, No. 3, 2024
ISSN: 1750-9548

Design and Research on Intelligent Library Guidance Software

Chunxiao Liu
Library, Heilongjiang Bayi Agricultural University, Daqing, 163319, China

Abstract

With the rapid development of artificial intelligence technology, libraries are increasingly leveraging new technologies to offer more convenient and efficient services. The design of intelligent library guidance software addresses this need by incorporating functionalities such as location-based services, book guidance, and data analysis. This software enhances user experience by providing personalized recommendations and streamlined access to library resources, thereby improving overall management efficiency. By employing RFID technology and Beidou positioning, the system can accurately guide users to their desired books and library facilities. Additionally, data mining techniques analyze user behavior and resource distribution, offering insights into reading habits and preferences. These insights are utilized to tailor recommendations and support decision-making in library procurement and resource management. The research and implementation of this intelligent system demonstrate its reliability, practicality, and innovation, paving the way for the intelligent evolution of library services.

Keywords: intelligence, Library, Guidance, Software, devise

1. Introduction

With the application of intelligent technology, people have also begun to pay attention to the intelligent management of libraries, especially the intelligent management of information resources assisted by intelligent technology.[1] The 2003 International Symposium on human-computer interaction and mobile devices and services introduced the concept of a smart library.[2]

The innovative function of the library access guide software is to use RFID technology and Beidou positioning technology to provide library access guidance services. This software can help readers quickly find the location of the books they need and the various facilities and services of the library. Another innovative function of the software is the use of data mining technology, through the analysis of library electronic resources, real-time analysis of the distribution of literature resources and readers' needs in the library, to discover the user's reading habits and preferences, so as to provide users with personalized recommendations and consulting services, enhance the reader's experience and satisfaction, and realize the efficient use and management of library resources. At the same time, it can also help libraries understand the reading needs of users, and provide reference for library procurement and resource construction.

The software offers all the knowledge and tools needed to construct and maintain a digital library, regardless of size or purpose. It is the perfect self-contained resource for individuals, agencies, and institutions wishing to put this powerful tool to work in their burgeoning information treasuries, thereby enhancing data management for libraries.[3]

My approach to software development has been significantly influenced by the production model of the Greenstone Digital Library Software. Greenstone, accessible at www.greenstone.org, offers a flexible method for organizing information and making it available on the internet or portable media such as CD-ROMs. Its primary

goal is to empower users, particularly those in universities, libraries, and other public service institutions, to create their own digital libraries.

As an open-source software, Greenstone is distributed under the terms of the GNU General Public License. This ensures that anyone can use, modify, and distribute the software freely, fostering a collaborative environment for software development and usage. The software was developed by the New Zealand Digital Library Project at the University of Waikato, in collaboration with UNESCO and the Human Info NGO, which helps in its development and distribution.

Libraries can offer immersive and interactive experiences, enhancing user engagement and access to information. The study discusses the potential benefits of our software, including personalized learning environments, which contribute to the intelligent management of library resources.[4]

By studying Greenstone's model, I've gained valuable insights into developing software that is both versatile and accessible. The collaborative nature of its development and the emphasis on flexibility and user empowerment have inspired me to adopt similar principles in my own software projects. This has allowed me to create solutions that are not only robust and user-friendly but also capable of being adapted and extended by others in the community.[5]

Technology Introduction:

1) RFID (Radio Frequency Identification) technology is a wireless communication technology, which is used to automatically identify target objects and obtain relevant data through radio signals, and carry out data transmission and identification through radio frequency.

Tag: Each tag has a unique electronic code that is attached to the object to identify the target object.

Reader: A reader is a device that can read (and sometimes write) tag information.

Antenna: Transmits RF signals between tags and readers.

Working process: When the tag enters the magnetic field of the reader, it will receive a radio frequency signal, and send the product information stored in the chip with the energy obtained by the induced current (passive tag or passive tag), or the tag will actively send a signal of a certain frequency (active tag or active tag). The reader reads and decodes the information and sends it to the central information system for data processing.

2) The Beidou Navigation System (BDS) is a global satellite navigation system independently developed by China, and its positioning principle is mainly based on radio ranging and three-dimensional coordinate calculation.

Working process: The user device receives signals from at least four satellites, and each signal contains the position information of the satellites and the time of transmission. The user device calculates the signal propagation time according to the received signal time and its own reception time, and then obtains the distance from the user device to each satellite. With the location information of the satellite and the distance of the user device to the satellite, the position of the user device can be calculated through three-dimensional coordinates.

3) Data mining technology is a method of extracting hidden, unknown, and valuable information and knowledge from large amounts of data. It is often used to analyze and discover patterns, trends, correlations, and predictions in data in order to provide useful information to decision-makers.

Workflow: Represent data as a pattern associated with a class or concept so that you can use the model to predict a class of objects with unknown class labels. Discover association rules, describe the relationship between items in a given data set, find out the model that describes and distinguish data classes or concepts, group data objects into multiple classes or clusters, and the objects in the same cluster have a high degree of similarity between them, while the objects in different clusters are quite different, and finally analyze the evolution of the class or cluster, describe the laws or trends of objects whose behavior changes over time, and model them. In addition, there are often data objects in the database that do not conform to the general model of the data, such data objects are called outliers, they are different or inconsistent with the rest of the data, and need to be specially analyzed.

This study had three objectives: recognizing the reasons for adopting the software for library automation, investigating the challenges librarians face while using it, and assessing satisfaction with its features. The findings underscore the need for continuous improvement of our software through practical application and feedback.[6]

2. Analysis of the Design Requirements of Intelligent Library Guidance Software

User requirements are the basis of the design of library admission guidance software, and they are the key factors for the successful implementation of the software. When designing a library access software, it's important to have a deep understanding of the user's needs.

2.1 Functional Requirements

Book Search. The software should provide the function of classifying the literature resources according to different classification standards to facilitate the user's search and browsing. The function should support a variety of classification methods, such as classification by subject, author classification, publication time, etc., and the classification results should be clear and easy for users to find. Users can search by entering keywords, author names, book titles and other keywords to quickly find the literature resources they need, and this function should also support fuzzy query, multi-condition filtering, sorting and other functions, so that users can find the books they need more conveniently.

Guidance function. The guidance function includes book guidance and location guidance, through RFID technology to obtain the positioning of the book or location, the use of Beidou satellite navigation to obtain the positioning of the reader, the reader manually enters the number of floors, and the software plans the route for the reader to find the book or place.

Book recommendations. The software can conduct data mining according to the user's usage history, search history, reading habits and other data, and recommend relevant literature resources to the user. This function has functions such as intelligent recommendation and personalized recommendation, which can provide users with more personalized recommendation services.

Resource downloads. The software should provide the function of downloading literature resources to the user's local device, so that the user can read and use it offline. The feature should support a variety of document formats, such as PDF, EPUB, CAJ, etc., and have a fast download speed, so that users can quickly obtain the information they need.

Resource sharing. The software should provide the function of sharing literature resources with others, so that users can share and communicate with others. The feature should support multiple sharing methods, such as email, social media, instant messaging, etc., and the sharing results should be clear and easy for users to share.

2.2 Performance Requirements

1) Responsiveness. For users, they want to be able to find the information they need quickly, so the software must be responsive enough. This means that the software must be able to respond quickly to a user's request and provide the appropriate results. When designing smart library guidance software, we should optimize the performance of the software as much as possible to improve the responsiveness.

2) System stability. For users, they want to be able to use a stable and reliable software without wasting time due to software crashes or bugs. Therefore, the software must have sufficient stability. When designing smart library guidance software, we should try to improve the stability of the software as much as possible to avoid errors.

2.3 Availability Requirements

Usability requirements are a very important aspect of software design, which involves many aspects such as ease of use, ease of learning, and ease of maintenance. In the design of smart library guidance software, we also need to consider these aspects to ensure that the software can provide users with a good user experience.

1) Ease of use. An easy-to-use software allows users to quickly get started and start using without much learning cost and time. In the design of the smart library guidance software, we can improve the ease of use of the software

through user-friendly interface design, clear operation prompts, reasonable menu structure and clear documentation.

2) Ease of learning. For most users, both the learning cost and the time cost of the software are very important factors. In the design of the smart library guidance software, we can reduce the user's learning cost and time cost by providing detailed instructions, video tutorials, online help, etc.

3) Easy maintenance. An easy-to-maintain software makes it easier for maintainers to fix problems in the software and ensures the stability and reliability of the software. In the design of intelligent library guidance software, we can improve the maintainability of the software through modular design, code standardization, version control, etc.

2.4 Compatibility Requirements

Compatibility requirements refer to the compatibility that the software should have, such as cross-platform, cross-browser, etc. These requirements are to ensure that the software works properly on different devices and operating systems, so that users can use the software conveniently.

1) Cross-platform compatibility

With the popularity of mobile devices, more and more people use mobile devices to access web resources, such as mobile phones, tablets, etc. Therefore, the smart library guidance software needs to support a variety of platforms, such as Windows, macOS, iOS, Android, etc. When designing, we need to ensure that the software can work properly on different platforms, including interface layout, function implementation, data storage, etc.

2) Cross-browser compatibility

Browsers are the main tools for people to access network resources, therefore, smart library guidance software needs to support a variety of browsers, such as Chrome, Firefox, Safari, etc. When designing, we need to make sure that the software will work well on different browsers.

2.5 Security Requirements

The software will involve users' personal information and library resource information, and the user's information, borrowing records, reservation records, catalog information and other private information stored in the software must be protected. Therefore, we need to take a series of security measures to ensure the security of data.

1) Access control. Restrict user access to the software, prevent unauthorized users from accessing certain information, and set different user roles and permissions, such as administrators, ordinary users, etc., and different user roles have different permissions.

2) Security audit. Security audits can record user behaviors and operations, so that administrators can detect and deal with security incidents in a timely manner. We can record user login, operation, access records and other information, and regularly analyze and audit this information, and set up a security incident response mechanism to detect and deal with security incidents in a timely manner to prevent security incidents from having a serious impact on the software.

3. Design of Intelligent Library Guidance Software System Architecture

3.1 System Module Division

The intelligent library guidance software is a complex project requiring the division into multiple modules to ensure functional integrity and project efficiency.

3.1.1 User Module

User Registration Module: Users create accounts and set personal information and passwords.

User Login Module: Users log in to access and manage their library resources.

Guidance Module: Users search for library resources by entering keywords and receiving guidance.

Recommendation Module: Recommends resources based on users' search and borrowing history using data mining.

User Borrowing Record Module: Users view their borrowing records, including details like due dates and fines.

3.1.2 Library Resource Module

Library Resource Management Module: Administrators manage resource information and properties.

Resource Query Module: Administrators view resource information such as name, author, and ISBN.

Borrowing Management Module: Administrators manage user borrowing records and set borrowing rules.

Data Analysis Module: Analyzes user data to inform library procurement and resource construction.

3.1.3 Positioning Guidance Module

Tag Conversion: Books or locations are tagged with RFID for identification.

Bibliographic Search: Readers search for books through various platforms.

Locate and Search: Users find books using a 3D navigation system integrated with Beidou positioning.

3.1.4 System Management Module

System Setting Module: Administrators configure system parameters.

Log Management Module: View system logs including login, error, and operation logs.

Data Backup Module: Backup system data and set backup schedules.

System Upgrade Module: Upgrade system versions and functions.

3.2 System Architecture Design

The system architecture design involves creating an overall framework where modules work together.

3.2.1 User Interface Layer

Interacts with users, displaying main and sub-interfaces for different functions like book borrowing, returning, and reservations.

3.2.2 Function Implementation Layer

Processes user requests and implements functions like guidance, recommendations, bibliographic retrieval, and borrowing management.

3.2.3 Data Access Layer

Interacts with the data storage layer to fetch and update information on books, readers, and borrowing records.[7]

3.2.4 Data Storage Layer

Stores data in relational, document, and cache databases for structured and unstructured data, and improves access speed.

4. Intelligent Library Guidance Software Interface Design

4.1 Interface design principles

Reasonably organize and arrange various graphics, buttons, text boxes, and menus on the interface to achieve beautiful, easy-to-use, and efficient purposes. In the design of smart library guidance software, the layout of the interface needs to take into account the functions of the software and the needs of users, such as placing commonly used functions in conspicuous positions. At the same time, you also need to consider the aesthetics of the interface, such as using clear fonts, concise layout, appropriate color matching, etc.

4.2 Interface interaction design

Interactive feedback is an important form of interaction between users and software. We need to design timely and accurate interactive feedback so that users can understand the status and results of the software in a timely manner. For example, when the user performs an operation, we can give prompt information in time and give timely feedback through sound, vibration, color, etc. to tell the user the specific content and results of the operation. We can also promptly provide prompt information when the user makes an operation error. Prompt information is given to help users correct errors.

5. Intelligent Library Guidance Software Implementation Technology Selection

5.1 Programming Languages

The intelligent library guidance software requires efficient and stable performance, making Java an ideal choice. Java's robust ecosystem offers numerous development tools and resources, along with a supportive community for technical assistance.

JavaFX for UI Development:

- Cross-Platform Consistency: Applications run smoothly on various platforms, including desktops, mobiles, and tablets.
- Scene Builder: Provides a drag-and-drop interface design tool integrated with IDEs like Eclipse and NetBeans.
- Swing Interoperability: Allows embedding Swing content within JavaFX applications.
- Rich UI Controls: Offers built-in controls for comprehensive application development.
- CSS-like Styles: Enhances application design with simple CSS knowledge.
- Canvas and Printing API: Provides APIs for immediate mode rendering and printing.
- Integrated Graphics Library: Supports 2D and 3D graphics with hardware acceleration.

5.2 Development Framework

Spring Boot is chosen for its ability to simplify the development process and ensure long-term stability and scalability.

- Simplified Configuration: Automates configuration with annotations and dependency management.
- Embedded Servers: Integrates Tomcat, Jetty, and Undertow for direct application deployment.
- Modular Design: Enhances code organization and reduces redundancy.
- Comprehensive Documentation: Offers extensive documentation and community support.
- Production-Ready Features: Includes performance metrics, application information, and health checks.

5.3 Database Technology

MySQL is selected for its efficiency, flexibility, and security in managing large volumes of document information.[**8**]

- Open Source: Offers community support and numerous third-party plugins without licensing costs.
- Performance: Utilizes B-tree disk tables and index compression for fast connections.
- Scalability: Supports distributed data storage and dynamic expansion without downtime.
- Security: Ensures data safety with encrypted transmission, access control, and real-time monitoring.
- Management Tools: Provides tools for monitoring, automated upgrades, and data migration.
- Java Integration: Supports standard SQL syntax and ODBC applications, with a secure permissions system.

6. Implementation of Guidance and Recommendation Functions

6.1 Insert Beidou Positioning Mobile Device

- Integrate Beidou SDK: Download and integrate the SDK into the project.
- Obtain Positioning Information: Use the SDK interface to get device location information.
- Parse Positioning Data: Extract relevant information such as longitude and latitude from the returned data.
- Display Positioning Information: Use map controls to mark coordinates on the map.
- Module Model ATGM336H-5N: Supports all GNSS systems for precise positioning.
- Parse NMEA-0183 Protocol Messages: Extract information like latitude, longitude, and speed.
- Authorization KEY: Obtain and integrate an independent authorization KEY from customer service.
- Additional Beidou Positioning Library: Place the library in the jniLibs directory.
- Set Satellite Positioning Signal Source: Configure the signal source through TencentLocationRequest.
- Floor Determination: Users manually input floor levels to match the map.

This streamlined content maintains essential information, making the technology selection and implementation steps concise and clear.

6.2 Locating books or places using RFID

1) Choose an RFID system: The RFID system consists of RFID tags and RFID readers.

2) Install RFID tags: Install RFID tags on the books or locations that need to be located.

3) Set up RFID readers: Install RFID readers in the areas that need to be located, and the locations of these readers need to be known.

4) Collect RFID signals: When the RFID reader reads the signal of the RFID tag, it will collect relevant information, such as signal strength, time, etc.

5) Positioning calculation: Positioning the tag through a reader with a known position can be divided into non-ranging methods and ranging methods.

6) Display positioning results: Display the positioning results in the application, which can be a map or list, etc.

6.3 Java inputs the flat map of each layer

TO be written using the JavaFX library

Create a map data structure: First, you need to create a data structure to store map information. This data structure can be a two-dimensional array, where each element represents a point on the map, and the values can represent attributes of the point (for example, whether it is an obstacle).

Enter map information: Then, you need to create a map for each floor and enter map information. It can be entered manually or read from a file or database.

Create a JavaFX application: Use JavaFX to create an application that will display a map. You need to create a Stage (window) and add a Scene (scene) inside it. In Scene, you can add various JavaFX components, such as Pane (Panel) or Group (Group), and draw maps in them.

Drawing maps: In a JavaFX application, use JavaFX's drawing API to draw maps. Use the Rectangle class to draw each point on the map. The color and size of the Rectangle can be set based on the properties of the points.

Add interaction: You can add event handlers to JavaFX components to respond to user actions. For example, you can add a mouse click event handler to display detailed information about a point on the map when the user clicks on it.

Here is a simple example showing how to create and display a simple map in JavaFX:

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.Pane;
import javafx.scene.shape.Rectangle;
import javafx.stage.Stage;
public class MapApp extends Application {
    private int[][] map;    // Map information
    @Override
    public void start(Stage primaryStage) {
        Pane pane = new Pane();
        for (int y = 0; y < map.length; y++) {
            for (int x = 0; x < map[y].length; x++) {
                Rectangle rect = new Rectangle(x * 10, y * 10, 10, 10);
                rect.setFill(map[y][x] == 1 ? Color.BLACK : Color.WHITE);
                pane.getChildren().add(rect);
            }
        }
        primaryStage.setScene(new Scene(pane));
        primaryStage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

6.4 Software analysis and planning route

To implement route planning between two points in Java, you can use graph algorithms. Dijkstra's algorithm can help find the shortest path between two points. If there are obstacles, consider the location of the obstacles as nodes in the graph, and then disconnect these nodes from other nodes so that the obstacles will be bypassed during path planning.

Here is a simple example using Dijkstra's algorithm

```
import java.util.*;
public class Dijkstra {
    public static void dijkstra(int[][] graph, int src) {
        int n = graph.length;
        int[] dist = new int[n];
        boolean[] visited = new boolean[n];
        Arrays.fill(dist, Integer.MAX_VALUE);
        dist[src] = 0;
        for (int i = 0; i < n - 1; i++) {
            int u = -1;
            for (int j = 0; j < n; j++) {
                if (!visited[j] && (u == -1 || dist[j] < dist[u])) {
                    u = j;
                }
            }
            visited[u] = true;
            for (int v = 0; v < n; v++) {
                if (graph[u][v] != 0 && !visited[v]) {
                    dist[v] = Math.min(dist[v], dist[u] + graph[u][v]);
                }
            }
        }
```

```
            for (int i = 0; i < n; i++) {
                System.out.println("Distance from " + src + " to " + i + " is " + dist[i]);
            }
        }
        public static void main(String[] args) {
            int[][] graph = {
                {0, 3, 5, 0, 0},
                {3, 0, 4, 6, 0},
                {5, 4, 0, 2, 2},
                {0, 6, 2, 0, 3},
                {0, 0, 2, 3, 0}
            };
            dijkstra(graph, 0);
        }
    }
```

In this code, graph is a two-dimensional array representing the distance between nodes in the graph, and src is the index of the starting node. The dijkstra function calculates the shortest distance from the starting node to all other nodes in the graph and prints the result.

If the user is on the same floor as the guidance book or location, the route will be planned to the destination. If not, the user will be guided to the more convenient stairs or elevator first, and the guidance will continue after reaching the same floor.

6.5 Conduct data mining and personalized recommendation of library resources

Data collection: First, the library's electronic resource data and user behavior data need to be collected. This may include metadata about the document (such as title, author, subject, etc.), as well as user behavioral data (such as search queries, browsing history, borrowing records, etc.).

Data Processing: The collected data then needs to be processed for analysis. Including data cleaning (such as removing duplicates, filling missing values, etc.), data transformation (such as converting text data into numerical data, etc.), and data integration (such as merging data from different sources, etc.).

Data Analysis: Use various data analysis methods to analyze the processed data. Use descriptive statistics methods to understand the distribution of literature resources, use association rule learning methods to discover users' reading habits and preferences, and use clustering methods to group users.

Recommendation generation: Based on the results of data analysis, you can generate personalized recommendations. Use collaborative filtering methods to generate recommendations. Collaborative filtering is a recommendation algorithm that analyzes the user's preferences, calculates the content that users who have the same preferences as the user like, calculates it in the background, and feeds the results back to the user.

Recommendation display: Finally, you need to display the generated recommendations to the user. Add a recommendation area to the user interface to display recommended literature and provide some operation options, such as viewing details, adding to the book list, etc. At the same time, the data is fed back to the backend to help the library understand users' reading needs and provide reference for library procurement and resource construction.

Feedback collection and model update: It is also necessary to collect user feedback on recommendations, such as click-through rates, like/dislike evaluations, etc., and update the recommendation model based on the feedback to improve the accuracy and satisfaction of recommendations.

7. Intelligent Library Guidance Software Testing And Maintenance

7.1 Test strategy

Reusable software components, typically integrated as libraries, are a central paradigm of modern software development. By incorporating a library into their software, developers trust in its quality and its correct and complete implementation. Since errors in a library affect all applications using it, there is a need for quality assurance tools such as automated testing that can be used by library and application developers to verify functionality.

Similarly, our software also requires continuous testing to ensure its functionality remains reliable. This ongoing testing process is crucial for maintaining the trust and confidence of our users. By employing automated testing and other quality assurance tools, we can proactively identify and address potential issues, ensuring that our software continues to meet high standards of performance and reliability.[9]

Testing strategy refers to the methods and strategies for testing software, aiming to discover and repair defects and problems in software. Testing strategy is a crucial part of the software testing process, which can ensure the quality and reliability of the software and improve user experience and satisfaction. The following are the testing tools selected based on the various parts of the test:

1) Functional testing tool: Selenium, which is an open source automated testing tool, mainly used for automated testing of web applications.

2) Performance testing tool: JMeter. JMeter is a Java-based stress testing tool designed to load test functional behavior and measure performance.

3) Security testing tool: Websecurify, which can help detect security vulnerabilities in web applications.

4) Automated testing tool: TestComplete. TestComplete is a powerful automated testing tool for creating, maintaining, executing and analyzing UI tests.

5) Test management tool: Jira is used to plan tests, manage the test process, generate test reports, etc.

6) Unit testing tool: JUnit is used to verify the functionality of individual software components.

7) Integration testing tool: Jenkins is used for testing at the component or system level to verify the interaction between different components.

7.2 Test environment setup

The first step in setting up a test environment is to determine the needs of the test environment. According to the software requirements specification, it is necessary to determine what functions and performances the test environment needs to have.

7.2.1 Hardware environment

The test environment needs to have sufficient hardware resources to ensure that the software can run normally in different hardware environments. It is necessary to ensure that the test environment has sufficient memory and CPU resources to ensure that the software does not freeze or crash during operation.

7.2.2 Software environment

The testing environment needs to have a suitable software environment to ensure that the software can run normally in different software environments. You need to install the software and tools required for testing, such as test management tools, automated testing tools, etc.

7.2.3 Network environment

The test environment needs to have a good network environment to ensure normal access to network resources during the test process. It is necessary to ensure that the test environment can normally access the library database, user interface, etc.

7.2.4 Data environment

The test environment needs to have sufficient data resources to ensure that data can be generated and processed normally during the test process. Sample data, test case data, etc. required for testing need to be prepared.

7.3 Test result analysis

Test result analysis is the process of analyzing and summarizing the results of software testing. This process can help the development team understand the performance and reliability of the software, thereby improving the software design and development process. The main purpose of test result analysis is to identify software defects and problems and provide solutions. The findings suggest that the sustainability of our software is influenced by multiple factors. Libraries must invest in resources such as funding, staff training, and community support to ensure the continued use and relevance of the software.[10]

1) Determine the type and number of defects: In the test result analysis, the development team should carefully analyze the test results to determine the type and number of defects in the software. Defect types can include functional defects, performance defects, security defects, etc. The number of defects should be classified and counted according to different defect types.

2) Determine the level of defects: Past research has proposed a reliability model for RFID software, considering the interdependence between fault detection and fault correction processes. This model enhances the stability and feasibility of our software by effectively estimating and forecasting its dependability, especially when compared with existing NHPP software reliability growth models (SRGMs).

In the analysis of test results, the development team should classify the defects into levels to determine the severity of the defects. Defect levels can include low, medium, and high. Leveling can help development teams better understand the impact and resolution of defects.

3) Determine the cause of the defect: During the test result analysis, the development team should carefully analyze the cause of the defect to determine the root cause of the defect. Causes of defects can include design defects, coding defects, testing defects, etc. Analyzing the causes can help the development team better understand the root cause of the defect and take appropriate measures to resolve the defect.

4) Determine solutions to defects: In the analysis of test results, the development team should propose corresponding solutions to solve software defects and problems. Solutions can include modifying code, fixing bugs, improving designs, etc. The solution should be chosen based on the severity and cause of the defect.

7.4 Software maintenance

Software maintenance refers to the later maintenance and updating of software after it is released to ensure the stability and reliability of the software while improving its performance and availability. As a common software application, intelligent library guidance software is also very important to maintain.

Security and privacy protection. Software security means that the software will not be subject to malicious attacks or intrusions by malware during use, thus ensuring the security of user data.

Maintainability and scalability. Maintainability refers to the time and cost required for software maintenance and updates in the future. In the design of intelligent library guidance software, it is necessary to adopt some maintainable design patterns and architectures, such as modular design, code reuse, etc., to reduce the difficulty and cost of software maintenance. At the same time, during the software maintenance process, the software also needs to be maintained and updated regularly to ensure the maintainability and scalability of the software.

8. Summary

8.1 Summary of research results

The research mainly discusses the requirements, design, implementation, testing and maintenance of intelligent intelligent library guidance software, and adds two innovative functions to it: using RFID technology and Beidou positioning technology to provide library entry guidance services. Use data mining technology to provide users

with personalized recommendations and consulting services, help libraries understand users' reading needs, and provide references for library procurement and resource construction.

After the above design and implementation, we have obtained an intelligent library guidance software that is powerful and meets user needs.

8.2 Future prospects

With the continuous advancement of science and technology, the development of library entry guidance software is also accelerating. In the future, the development direction and trend of intelligent library guidance software can be expected from the following aspects:

1) Intelligence

In the future, smart library guidance software will be more intelligent, able to automatically identify users' identities, preferences and needs, and provide more personalized services. The software can analyze the user's reading habits and interests, recommend relevant books and literature, and even provide users with a customized reading experience.

2) Socialization

In the future, smart library guidance software will pay more attention to socialization and enable interaction and communication between users. For example, the software can provide online discussion forums, social sharing and other functions. Users can share their reading thoughts and experiences in the discussion forum, and can also share reading materials with others through the social sharing function.

3) Integration

In the future, smart library guidance software will pay more attention to integration and can achieve integration with other libraries and academic resources. For example, the software can integrate data and resources from other libraries and academic resources to provide users with more comprehensive and rich information and services.

In the future, the development direction and trend of intelligent library guidance software will be more intelligent, social and integrated, and can provide users with more convenient, personalized and secure services.

Acknowledgements

This article is one of the research results of the Heilongjiang Provincial University Graphics Working Committee project "Research on Improving Strategies for Information Literacy Education in Colleges and Universities from an Ecological Perspective" (Project Number, 2021-047-B).

References

[1] Wu Z Q, Yang X X. A Preliminary Survey on Research and Practice of Smart Library in China. Library and information service, 2021(4): 20-27.

[2] Ojalat A R. Smart library: location-aware mobile library service. International symposiumon human computer interaction with mobile devices and services, 2003(5): 411-415.

[3] Witten I, Bainbridge D, Nichols D. How to Build a Digital Library. online, 2010.DOI:10.1016/C2009-0-19701-5.

[4] Qiao J. Accelerating the Application of the Metaverse: A Necessity for the Transformation from Digital to Smart Libraries. Frontiers in Educational Research, 2024,7(6):

[5] Bainbridge D, Witten I H .Greenstone digital library software: current research. 2004. DOI:10.1109/JCDL.2004.1336220.

[6] Mazhar I, Kabir M K, Arslan S .Use of software for automation of academic libraries in Sialkot. Information Discovery and Delivery, 2023, 51(4): 417-428.

[7] SUN Y, ZHOU G. The Construction of Resource Discovery System Platform Based on Microservices Architecture. Journal of Library Science of China, 2020(01): 114-124.

[8] Idocin F J ,Perez A J .Ex-Fuzzy: A library for symbolic explainable AI through fuzzy logic programming. Neurocomputing, 2024, 599128048-128048.

[9] Ahammad N, Bahry S D F, Husaini H. Sustainable library services with open-source library automation and digitisation software: A literature review. Business Information Review, 2024, 41(2): 59-68.

[10] Nicholas E U, Anthonia E N, Koushik G, et al. RFID library management software dependability through reliable fault-detection and fault correction procedures. Microsystem Technologies, 2024, 30(5): 647-659.