**BOOK BANK MANAGEMENT SYSTEM**

**PROBLEM STATEMENT:**

A Book Bank lends books and magazines to members, who are registered in the system. Also,it handles the purchase of new titles for the Book Bank. Popular titles are brought in multiple copies. Old books and magazines are removed when they are out of date or poor in condition. A member can reserve a book or magazine that is not currently available in the book bank so that when it is returned or purchased by the book bank, that person is notified. The book bank can easily create, replace and delete information about the tiles, members, loans, and reservations from the system.

**Software Requirement Specification**

For

BOOKBANK MANAGEMENT SYSTEM

Version 1.0 approved

Prepared by:

1. N.HIRANYAAKSHITHA - 22BD5A6613
2. M.RISHITHA - 21BD1A6659
3. K.AKSHITHA REDDY - 21BD1A664W
4. M.HARSHITHA - 21BD1A665F

**Keshav Memorial Institute of Technology**

**X-XX-XXXX**.

**Table of Contents**

**1.Introduction**.........................................................................................................................

1.1Purpose........................................................................................................ 1.2 DocumentConventions................................................................................ 1.3 Intended Audience and Reading Suggestions........................................... 1.4 Product Scope............................................................................................ 1.5 References................................................................................................. **2.Overall Description**

2.1 Product Perspective…………………………………………………… 2.2 Product Functions…………………………………………………….. 2.3 Operating Environment………………………………………………. 2.4 User Characteristics………………………………………………….. 2.5 Design and Implementation Constraints…………………………… 2.6 User documentation…………………………………………………… 2.7 Assumptions and Dependencies…………………………………….

**3.Specific Requirements**

3.1Functional Requirements………………………………………………… 3.2 Non-Functional Requirements…………………………………………

**4. Appendix Diagram**

4.1 Use Case Diagram

4.2 Use Case Template

4.3 Sequence And Collaboration Diagram 4.4 Class Diagram

4.5 State Chart Diagram

4.6 Activity Diagram

4.7 Package Diagram

4.8 Component Diagram

4.9 Deployment Diagram

**BOOK BANK**

**1.INTRODUCTION:**

Book Bank is the interface between the students and the Librarian. It aims to improve the efficiency in the Issue of books or magazines and reduce the complexities involved in it to the maximum possible extent.

**1.1 PURPOSE:**

If the entire process of 'Issue of Books or Magazines' is done manually, it would take several months for the books or magazines to reach the applicant. Considering that the demand for Book banks is increasing every year, an Automated System becomes essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process. The system has been carefully verified and validated to satisfy.

**1.2 Document Convention:**

**Heading**:

Font-Size:16

Font-Style: Bold

Font: Times New Roman

**Subheading:**

Font-Size:14

Font-Style: Bold

Font: Times New Roman

**Content:**

Font-Size:12

Font: Times New Roman.

**1.3 Intended Audience and Reading Suggestions:**

The intended audience for the Book Bank system includes Book Bank management and staff, registered members, and administrators/IT support. Book Bank staff and management should refer to user manuals and documentation for operational guidance, covering tasks like adding titles, managing reservations, and handling loans. Members need user guides to navigate borrowing, reserving titles, and account management, alongside information on notification processes. Administrators and IT support personnel should grasp the system's technical aspects, including database management and security protocols, with access to troubleshooting resources. Training sessions, periodic updates, feedback mechanisms, and online support

should be implemented to ensure users' effective engagement and satisfaction with the system, enhancing the overall Book Bank experience.

**1.4 Product Scope:**

The System provides an online interface to the user where they can fill in their details and submit the necessary documents (maybe by scanning). The authority concerned with the issue of books can use this system to reduce his workload and process the application speedily.

**1.5 References:**

We took references from

www.reachbookbank.com

www.bookworldlibrary.com

**2.Overall Description**

**2.1 Product Perspective**

The Book Bank Management System is designed to operate as a standalone software solution, providing comprehensive tools and functionalities for efficiently managing the lending, inventory, and member interactions within a book bank. It serves as the primary interface between the Book Bank staff and members, offering a user-friendly environment for tasks such as cataloging, tracking, and loaning books and magazines. From a broader product perspective, this system can also be integrated with other library management systems or databases, facilitating seamless data exchange and resource sharing between different libraries or educational institutions. Additionally, the system can be further extended to include features like data analytics and reporting, allowing administrators to gain valuable insights into member preferences and inventory trends. This holistic perspective ensures that the Book Bank Management System not only meets immediate operational needs but also offers scalability and adaptability to cater to the evolving requirements of the book bank and its users.

**2.2 Product functions**

**2.2.1 Administrator**

• User Management: Administrators can create, modify, and deactivate user accounts, both for staff members responsible for system operation and book bank members. This ensures that access to the system is appropriately controlled and managed.

• Catalog Management: Administrators have the capability to add, update, or remove book and magazine titles from the catalog. This function includes the ability to edit details such as author names, publication dates, and categories to maintain an accurate and up-to-date inventory.

• Loan Management: This function allows administrators to oversee the loan process. They can track due dates, manage returns, and apply fines or penalties for overdue items. Administrators can also extend loan periods or process returns as needed.

• Reservation Management: Administrators are responsible for managing the reservation system. This includes maintaining reservation queues, sending notifications to members about available items, and allocating reserved items to members when they become available through returns or purchases.

• Inventory Maintenance: Administrators monitor the condition of items in the inventory. They can initiate the removal of outdated or damaged materials and facilitate the acquisition of new titles to keep the catalog current and in good condition.

• Reporting and Analytics: Administrators can generate reports and analyze data related to member activity, popular titles, and system performance. This data driven insight aids in decision-making and optimizing book bank operations.

• Security and Access Control: Administrators configure user access permissions and security settings to protect sensitive data and maintain the integrity of the system.

• System Configuration: Administrators have the ability to customize system settings, including loan durations, fine policies, and integration options with other library systems or databases, to suit specific operational needs.

• Backup and Recovery: The system includes backup and recovery features that administrators can use to safeguard data and ensure continuity in case of system failures or data loss.

• User Support: Administrators provide user support by addressing inquiries, troubleshooting issues, and ensuring a smooth experience for all system users, both staff and book bank members.

**2.2.2 Customers/Users**

• **User Registration**: Users, particularly book bank members, can register for an account in the system. This function typically includes providing personal information, contact details, and agreeing to terms and conditions.

• **Catalog Search:** Users can search for books and magazines within the catalog. They can perform searches based on titles, authors, categories, or keywords, making it easy to find items of interest.

• **Borrowing:** Members can borrow books and magazines from the book bank. They can view item availability, check out materials, and receive due date information.

• **Reservation:** Users have the option to reserve items that are currently unavailable. They can place reservations, receive notifications when reserved items become available, and pick up reserved materials.

• **Account Management:** Users can manage their accounts, including updating personal information, changing passwords, and viewing their borrowing history.

**2.3 Operating Environment**

**Operating System:** The system should be compatible with common operating systems, including:

• Windows Server (e.g., Windows Server 2016, 2019)

• Linux distributions (e.g., Ubuntu, CentOS, Red Hat)

• macOS for client-side applications

Web Server: If the system is web-based, it may require a web server such as:

• Apache HTTP Server

• Nginx

• Microsoft Internet Information Services (IIS)

**2.4 User Characteristics**

User characteristics play a crucial role in shaping the design and functionality of a Book Bank Management System. These characteristics encompass both administrators responsible for system operation and book bank members seeking to borrow materials. Administrators are typically individuals with various levels of technical expertise, including librarians and system administrators, who require robust tools for catalog management, user administration, and system configuration. They need efficient interfaces and advanced functionalities to ensure smooth system operation. On the other hand, book bank members vary in age, technological familiarity, and preferences. Hence, the system should offer an intuitive user experience, simplified search and borrowing processes, and accessible account management options. Recognizing these diverse user characteristics allows for the creation of a user-friendly system that caters to the needs of both administrators and members, ultimately enhancing the efficiency and effectiveness of the Book Bank Management System.

**2.5 Design and Implementation Constraints**

• Budget Constraints: Limited financial resources may constrain the development and maintenance of the system. This could impact the choice of technology stack, scalability options, and the extent of features that can be implemented.

• Technology Stack: Constraints related to existing technology infrastructure can influence the choice of programming languages, databases, and frameworks. Integration with legacy systems may also pose challenges.

• Regulatory Compliance: The system may need to adhere to specific regulations and standards, such as data privacy laws (e.g., GDPR), accessibility standards (e.g., WCAG), or library management guidelines. Compliance can impact design decisions and functionality.

• Data Security: Ensuring data security and privacy is paramount. Constraints related to security measures may affect the design and implementation, including encryption, authentication, and authorization mechanisms.

**2.6 Assumptions and Dependencies**

**Assumptions:**

• **User Internet Access:** It is assumed that both administrators and book bank members have access to the internet, as the system may be web-based.

• **User Device Compatibility:** Users are expected to have access to devices (e.g., computers, smartphones, tablets) that can run compatible web browsers or mobile apps for system access.

• **Data Accuracy**: It is assumed that data provided by publishers, including book and magazine information, is accurate and up-to-date.

• **User Authentication:** Users are assumed to have valid credentials for system access, and user authentication processes will be effective in verifying their identity.

• **Regulatory Compliance:** The system assumes that it will be designed and maintained in compliance with relevant data protection, accessibility, and library management regulations.

• **Adequate Resources**: Sufficient hardware resources, such as servers and network infrastructure, are assumed to be available to support the system's performance and reliability.

**Dependencies:**

• Database Management System: The system may depend on a specific database management system (e.g., MySQL, PostgreSQL) for data storage and retrieval.

• External Data Sources: The availability and reliability of external data sources, such as ISBN databases and online book catalogs, can impact the accuracy of catalog information.

• Third-Party APIs: If the system integrates with third-party services (e.g., payment gateways, email services), its functionality may depend on the availability and stability of these APIs.

• Library Network Integration: If the book bank is part of a larger library network, dependencies may exist on network-wide policies, standards, and protocols.

• User Training: Dependencies on user training resources, including the availability of training materials and staff for user assistance, may be required for system adoption.

• Server Hosting Services: For cloud-based systems, the availability and reliability of the chosen hosting service provider can be a critical dependency.

• Software Development Tools: The development process may depend on specific software development tools, frameworks, and libraries, which need to be available and up to date.

• Operating System Updates: Dependencies exist on the timely application of operating system updates and security patches to maintain system security and stability.

• Payment Processing Services: If the system involves payment processing, dependencies exist on payment gateway services for transaction processing.

• Data Backups and Recovery: The availability and effectiveness of data backup and recovery mechanisms are crucial dependencies for data protection and system continuity.

**3.Specific requirements**

**3.1 Functional Requirements:**

• User Registration: Users (both staff and book bank members) should be able to register for accounts, providing necessary information and agreeing to terms and conditions. • Catalog Management: Administrators should have the ability to add, edit, and remove

book and magazine titles from the catalog, including details like authors, publication dates, and categories.

• Loan Management: The system should track loans, due dates, and return statuses, allowing administrators to extend loan periods, process returns, and apply fines for overdue items.

• Reservation Management: Users should be able to reserve items, receive notifications when reserved items become available, and collect reserved materials promptly. • Inventory Maintenance: Administrators should monitor the condition of items, remove outdated or damaged materials, and facilitate the purchase of new titles to keep the catalog current.

• Reporting and Analytics: The system should provide reporting tools for administrators to generate reports and analyze data on member activity, popular titles, and system performance.

• Security and Access Control: User access permissions and security settings should be configurable by administrators to safeguard data and system integrity.

• System Configuration: Administrators should be able to customize system settings, including loan durations, fine policies, and integration options with other library systems or databases.

• Backup and Recovery: The system should implement regular data backups and recovery procedures to ensure data protection and system continuity.

• User Support: Users should have access to user support features, including inquiries, troubleshooting, and assistance with system usage.

**3.2 Non-functional Requirements:**

**3.2.1 Performance Requirements:**

• Responsiveness: The system should respond to user actions promptly, with a maximum acceptable response time of 2 seconds for most operations.

• Scalability: It should be able to handle a growing user base and increasing inventory size, with the capacity to support at least 1,000 concurrent users.

• Data Processing: The system should efficiently process and retrieve data, with a capacity to handle at least 10,000 catalog records.

**3.2.2 Safety Requirements:**

• Data Privacy: Personal and sensitive user data should be securely stored and protected in compliance with data privacy regulations.

• Redundancy: Implement redundancy and failover mechanisms to ensure data availability and system resilience.

**3.2.3 Security Requirements:**

• Authentication: Users must authenticate securely, with strong password policies and optional two-factor authentication for enhanced security.

• Authorization: Role-based access control should be in place to grant appropriate permissions to users based on their roles.

• Data Encryption: Sensitive data, including user credentials and personal information, should be encrypted during transmission and storage.

**3.2.4 Error Handling:**

• User-Friendly Error Messages: Provide clear and user-friendly error messages for both administrators and members to aid in issue resolution.

• Logging: Implement comprehensive error logging to track system issues and enable timely troubleshooting and resolution.

• These specific requirements are essential for designing and developing a reliable, secure, and user-friendly Book Bank Management System that meets the needs of administrators and book bank members while ensuring compliance with relevant regulations and standards.

**Software Development Life Cycle (SDLC)**

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

● SDLC is the acronym of Software Development Life Cycle.

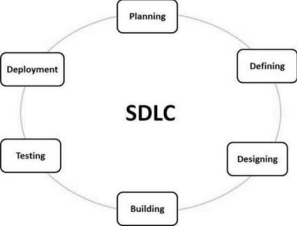
● It is also called the Software Development Process.

● SDLC is a framework defining tasks performed at each step in the software development process.

● ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

**What is SDLC?**

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.

**A typical Software Development Life Cycle consists of the following stages −**

**Stage 1: Planning and Requirement Analysis**

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

**Stage 2: Defining Requirements**

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

**Stage 3: Designing the Product Architecture**

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters such as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

**Stage 4: Building or Developing the Product**

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

**Stage 5: Testing the Product**

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

**Stage 6: Deployment in the Market and Maintenance**

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

**SDLC Models**

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred to as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

Following are the most important and popular SDLC models followed in the industry −

● Waterfall Model

● Iterative Model

● Spiral Model

● V-Model

● Big Bang Model

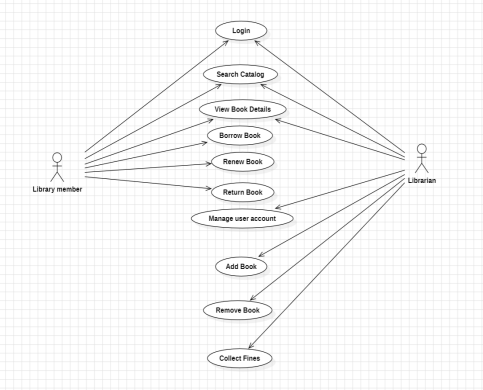
The foundational phases of the SDLC, such as requirements analysis, planning, system design, implementation, testing, deployment, and maintenance. These phases are commonly present in various SDLC models, including the Waterfall model, Agile methodologies (e.g., Scrum, Kanban), and DevOps practices.

**UML DIAGRAM:**

Unified Modeling Language (UML) is a general purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed.

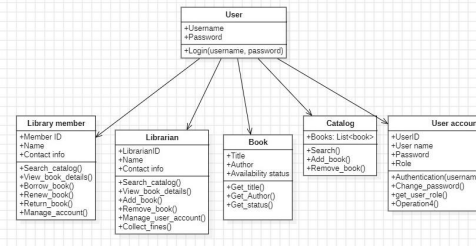
**USE-CASE:**

Use-Case is a list of actions or events. Steps typically define the interactions between a role and a system to achieve a goal. The use-case diagram consists of various functionality performed by actors like Student, Admin, book bank and DBA.



**CLASS DIAGRAM**:

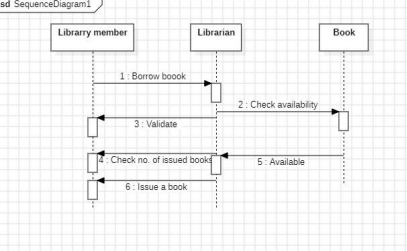
A class diagram in the unified modeling language is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations and the relationships among objects. The book bank system makes use of the following classes: student, book bank, admin and DBA.



**SEQUENCE DIAGRAM:**

A sequence diagram represents the sequence and interactions of a given use case or scenario. Sequence diagrams capture most of the information about the system. It is also represented in order by which they occur and have the object in the system send messages to one another. Here the sequence starts with interaction between student and book bank followed by a database. Once the book has been selected the next half of the sequence starts between the book bank and admin followed by a database.

**SEQUENCE DIAGRAM FOR BOOK ISSUE & RETURN:**

****

**COMPONENT DIAGRAM:**

A component diagram is a type of UML (Unified Modeling Language) diagram used in software engineering and system design to illustrate the high-level structure of a system or software application. It focuses on the organization and relationships among various components or modules within the system. Components are typically software elements that encapsulate and provide specific functionalities, and component diagrams help visualize how these components interact and collaborate to achieve the overall system's objectives.

Key elements of a component diagram include:

1. Components.

2. Interfaces.

3. Dependencies.

4. Relationships:

5. Notations.

