



# **PROJECT SYNOPSIS**

## **ON**

# **LIBRARY MANAGEMENT**

**Submitted To:**

Mr. C. Radhakrishnan  
PGT Computer Science  
(083)

**Submitted By:**

**Name:** RITHIK.A  
**Class:** XII  
**Roll No:**  
**Subject:** Computer Science  
**Session:** 2024-2025

## ***Certificate***

This is to certify that the project entitles **LIBRARY MANAGEMENT** is a bona fide work done by **RITHIK. A** of **Class XII Computer Science**, Session 2024-25 in the partial fulfilment of CBSE AISSCE Examination 2024 and has been carried out under my direct supervision and guidance.

**Internal Examiner**

**External Examiner**

**Principal**

## ***Acknowledgement***

I would like to express my special thanks of gratitude to my teacher **Mr. C. Radhakrishnan** as well as our Principal **Ms. M. Porchelvi**, who gave me the golden opportunity to do this wonderful project on the topic **LIBRARY MANAGEMENT** which also helped me in doing a lot of research and I came to know about so many new things I am really thankful to them.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Last but not least, I would like to thank all those who had helped towards the completion of this project.

**RITHIK.A**

**Class – XII Computer Science**

<b>S. No</b>	<b>Contents</b>	<b>Page No</b>
1	Manual System	5
2	Proposed System	6
3	Introduction of the Project	8
4	Objectives and Scope of the Project	9
5	System Development Life Cycle	11
6	Hardware and Software Specifications	24
7	Working Environment	25
8	Functions and Modules used in Project	27
9	Source Code & Sample Output	28
10	Limitations and Future Scope	57
11	Bibliography	58

# 1. Manual System

## 1. **Introduction to Manual Library Management:**

- A manual library management system refers to the traditional, non-computerized approach to managing library operations.
- In this system, librarians and staff handle tasks manually, using paper-based records and registers.

## 2. **Characteristics of a Manual Library System:**

- **Record Books:** Librarians maintain handwritten registers for book transactions, member details, and other library-related activities.
- **Card Catalogs:** Physical card catalogs organize book information alphabetically or by subject.
- **Manual Borrowing and Returning:** Librarians manually issue books to members and record returns.
- **Late Fee Calculation:** Late fees for overdue books are calculated manually.
- **Inventory Management:** Librarians physically check shelves to verify book availability.
- **Member Registration:** New members are registered using paper forms.
- **Limited Accessibility:** Members need to visit the library in person for inquiries or updates.

## 3. **Advantages of Manual Systems:**

- **Simplicity:** Manual systems are straightforward and easy to understand.
- **Low Cost:** No need for expensive software or hardware.
- **Personal Interaction:** Librarians directly interact with members, providing personalized service.

## 4. **Challenges of Manual Systems:**

- **Time-Consuming:** Manual processes are slower due to handwritten records.
- **Error-Prone:** Mistakes can occur during data entry or calculations.
- **Limited Scalability:** Difficult to manage large libraries efficiently.
- **Lack of Real-Time Information:** Members cannot check book availability remotely.

## **2. Proposed System**

**1. Efficient Book Management:** Streamlining the process of book acquisition, cataloguing, and tracking to ensure an organized and easily accessible collection.

**2. User-Friendly Interface:** Developing an intuitive and user-friendly interface for library staff and patrons to facilitate easy navigation, quick retrieval of information, and seamless interaction with the system.

**3. Automation of Processes:** Automating routine library tasks such as book check-in and check-out, reservation management, and overdue notifications to improve operational efficiency and reduce manual workload.

**4. Inventory Management:** Implementing a robust inventory management system to monitor stock levels, identify popular titles, and facilitate timely reordering of books to maintain a well-stocked library.

**5. Enhanced Search and Retrieval:** Implementing an advanced search mechanism to allow users to quickly locate books, authors, or genres, promoting a more efficient and enjoyable library experience.

**6. User Account Management:** Providing features for patrons to create accounts, track their borrowing history, and manage personal preferences, fostering a personalized and user-centric library experience.

**7. Reporting and Analytics:** Incorporating reporting tools to generate insights into library usage, popular genres, and circulation trends, enabling informed decision-making for library administrators.

**8. Security and Access Control:** Implementing robust security measures to protect sensitive library data and incorporating access controls to ensure that only authorized personnel have access to specific functionalities.

**9. Integration with Other Systems:** Offering the flexibility for integration with other academic or administrative systems to create a cohesive and interconnected information ecosystem within the institution.

**10. Scalability:** Designing the system to be scalable, allowing for easy expansion and adaptation to the evolving needs of the library as it grows over time.

### **3. Introduction of the Project**

The **Library Management System** is a Python-based software project designed to efficiently manage library operations. It provides functionalities related to book management, member management, borrowing, and returning books.

Librarians can create and manage bibliographic records for library materials. Facilitates check-in, check-out, holds, and renewals. Online Public Access Catalog (OPAC): Allows users to search for materials and manage their accounts online. It Manages the process of acquiring new materials. Also Generates usage statistics and circulation data.

Librarians perform tasks such as adding new books, updating records, and managing user accounts. Users access the OPAC to search for materials, place holds, and view their borrowing history. The system centralizes operations, improving efficiency and user experience.

Efficiency: Streamlines library processes and reduces manual work. Accessibility: Provides easy access to resources through the online catalog. Accuracy: Maintains precise records of materials and user accounts.

#### **4. Objectives and Scope of the Project**

The prime objective of this project is to let the students apply the programming knowledge into a real-world situation/problem and exposed the students how programming skills helps in developing a good software.

Its added objectives include:

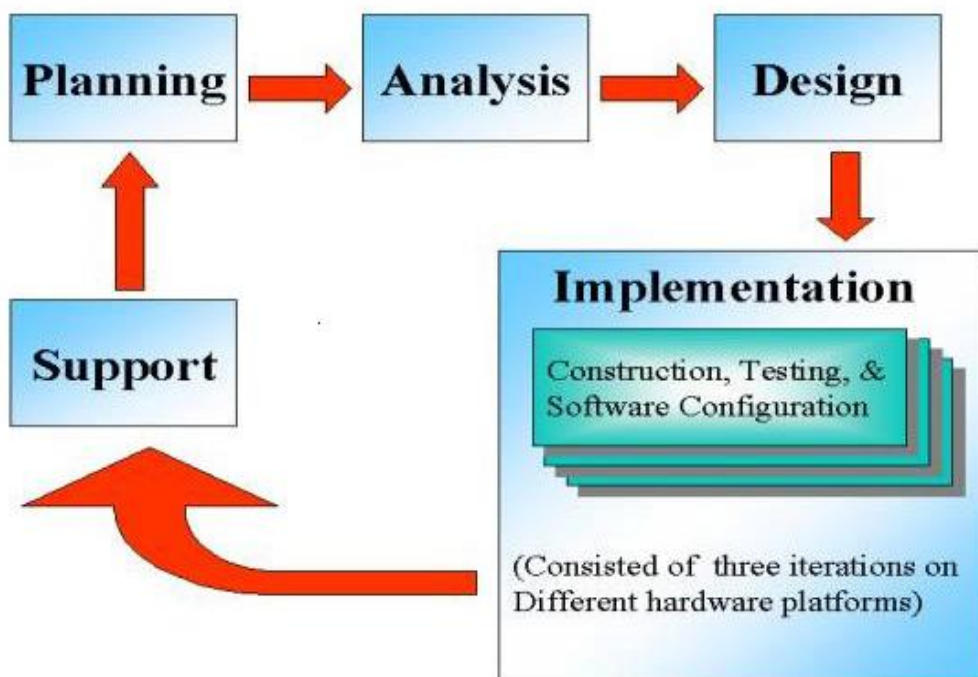
1. Write programs utilizing modern software tools.
2. Apply object-oriented programming principles effectively when developing small to medium sized projects.
3. Write effective procedural code to solve small to medium sized problems.
4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.
5. Students will demonstrate ability to conduct research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.



# System Development Life Cycle

SDLC is step by step procedure or systematic approach to develop software and it is followed within a software organization. It consists of various phases which describe how to design, develop, enhance and maintain particular software.

## SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)



The systems development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases. Segmenting projects allows managers to verify the successful completion of project phases before allocating resources to subsequent phases.

Software development projects typically include initiation, planning, design, development, testing, implementation, and maintenance phases. However, the phases may be divided differently depending on the organization involved.

For example, initial project activities might be designated as request, requirements-definition, and planning phases, or initiation, concept-development, and planning phases. End users of the system under development should be involved in reviewing the output of each phase to ensure the system is being built to deliver the needed functionality.

## PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE

### INITIATION PHASE

The Initiation Phase begins when a business sponsor identifies a need or an opportunity.

The purpose of the Initiation Phase is to:

- Identify and validate an opportunity to improve business accomplishments of the organization or a deficiency related to a business need.
- Identify significant assumptions and constraints on solutions to that need.
- Recommend the exploration of alternative concepts and methods to satisfy the need including questioning

the need for technology, i.e., will a change in the business process offer a solution?

- Assure executive business and executive technical sponsorship. The Sponsor designates a Project Manager and the business need is documented in a Concept Proposal. The Concept Proposal includes information about the business process and the relationship to the Agency/Organization.
- Infrastructure and the Strategic Plan. A successful Concept Proposal results in a Project Management Charter which outlines the authority of the project manager to begin project.

Careful oversight is required to ensure projects support strategic business objectives and resources are effectively implemented into an organization's enterprise architecture. The initiation phase begins when an opportunity to add, improve, or correct a system is identified and formally requested through the presentation of a business case. The business case should, at a minimum, describe a proposal's purpose, identify expected benefits, and explain how the proposed system supports one of the organization's business strategies. The business case should also identify alternative solutions and detail as many informational, functional, and network requirements as possible.

## SYSTEM CONCEPT DEVELOPMENT PHASE

The System Concept Development Phase begins after a business need or opportunity is validated by the Agency/Organization Program Leadership and the Agency/Organization CIO.

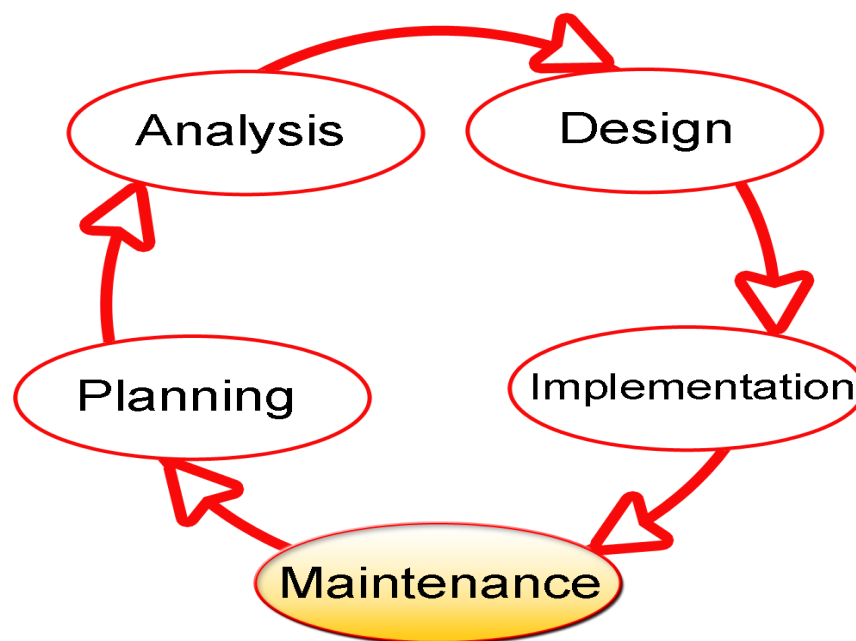
The purpose of the System Concept Development Phase is to:

- Determine the feasibility and appropriateness of the alternatives.
- Identify system interfaces.
- Identify basic functional and data requirements to satisfy the business need.
- Establish system boundaries; identify goals, objectives, critical success factors, and performance measures.
- Evaluate costs and benefits of alternative approaches to satisfy the basic functional requirements
- Assess project risks
- Identify and initiate risk mitigation actions, and Develop high-level technical architecture, process models, data models, and a concept of operations. This phase explores potential technical solutions within the context of the business need.
- It may include several trade-off decisions such as the decision to use COTS software products as opposed to developing custom software or reusing software

components, or the decision to use an incremental delivery versus a complete, onetime deployment.

- Construction of executable prototypes is encouraged to evaluate technology to support the business process. The System Boundary Document serves as an important reference document to support the Information Technology Project Request (ITPR) process.
- The ITPR must be approved by the State CIO before the project can move forward.

### PICTORIAL REPRESENTATION OF SDLC:



### PLANNING PHASE

The planning phase is the most critical step in

completing development, acquisition, and maintenance projects. Careful planning, particularly in the early stages of a project, is necessary to coordinate activities and manage project risks effectively. The depth and formality of project plans should be commensurate with the characteristics and risks of a given project. Project plans refine the information gathered during the initiation phase by further identifying the specific activities and resources required to complete a project.

A critical part of a project manager's job is to coordinate discussions between user, audit, security, design, development, and network personnel to identify and document as many functional, security, and network requirements as possible. During this phase, a plan is developed that documents the approach to be used and includes a discussion of methods, tools, tasks, resources, project schedules, and user input. Personnel assignments, costs, project schedule, and target dates are established.

A Project Management Plan is created with components related to acquisition planning, configuration management planning, quality assurance planning, concept of operations, system security, verification and validation, and systems engineering management planning.

## REQUIREMENTS ANALYSIS PHASE

This phase formally defines the detailed functional user requirements using high-level requirements identified in the Initiation, System Concept, and Planning phases. It also delineates the requirements in terms of data, system performance, security, and maintainability requirements for the system. The requirements are defined in this phase to a level of detail sufficient for systems design to proceed. They need to be measurable, testable, and relate to the business need or opportunity identified in the Initiation Phase. The requirements that will be used to determine acceptance of the system are captured in the Test and Evaluation Master Plan.

The purposes of this phase are to:

- Further define and refine the functional and data requirements and document them in the Requirements Document,
- Complete business process reengineering of the functions to be supported (i.e., verify what information drives the business process, what information is generated, who generates it, where does the information go, and who processes it),
- Develop detailed data and process models (system inputs, outputs, and the process.
- Develop the test and evaluation requirements that will be used to determine acceptable system performance.

## DESIGN PHASE

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase. Program designs are constructed in various ways. Using a top-down approach, designers first identify and link major program components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections.

Contemporary design techniques often use prototyping tools that build mock-up designs of items such as application screens, database layouts, and system architectures. End users, designers, developers, database managers, and network administrators should review and refine the prototyped designs in an iterative process until they agree on an acceptable design. Audit, security, and quality assurance personnel should be involved in the review and approval process. During this phase, the system is designed to satisfy the functional requirements identified in the previous phase. Since problems in the design phase could be very expensive to solve in the later stage of the



software development, a variety of elements are considered in the design to mitigate risk. These include:

- Identifying potential risks and defining mitigating design features.
- Performing a security risk assessment.
- Developing a conversion plan to migrate current data to the new system.
- Determining the operating environment.
- Defining major subsystems and their inputs and outputs.
- Allocating processes to resources.
- Preparing detailed logic specifications for each software module. The result is a draft System Design Document which captures the preliminary design for the system.
- Everything requiring user input or approval is documented and reviewed by the user. Once these documents have been approved by the Agency CIO and Business Sponsor, the final System Design Document is created to serve as the Critical/Detailed Design for the system.
- This document receives a rigorous review by Agency technical and functional representatives to ensure that it satisfies the business requirements.
- Concurrent with the development of the system design, the Agency Project Manager begins

development of the Implementation Plan, Operations and Maintenance Manual, and the Training Plan.

## DEVELOPMENT PHASE

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins. The procedures help ensure programmers clearly understand program designs and functional requirements. Programmers use various techniques to develop computer programs. The large transaction-oriented programs associated with financial institutions have traditionally been developed using procedural programming techniques. Procedural programming involves the line-by-line scripting of logical instructions that are combined to form a program. Effective completion of the previous stages is a key factor in the success of the Development phase. The Development phase consists of:

- Translating the detailed requirements and design into system components.
- Testing individual elements (units) for usability.
- Preparing for integration and testing of the IT system

## INTEGRATION AND TEST PHASE

- Subsystem integration, system, security, and user acceptance testing is conducted during the integration and test phase. The user, with those responsible for quality assurance, validates that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. OIT Security staff assess the system security and issue a security certification and accreditation prior to installation/implementation.

### Multiple levels of testing are performed, including:

- Testing at the development facility by the contractor and possibly supported by end users
- Testing as a deployed system with end users working together with contract personnel
- Operational testing by the end user alone performing all functions. Requirements are traced throughout testing, a final Independent Verification & Validation evaluation is performed and all documentation is reviewed and accepted prior to acceptance of the system.

## IMPLEMENTATION PHASE

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirements.

## OPERATIONS AND MAINTENANCE PHASE

The system operation is ongoing. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to the organization's needs. When modifications or changes are identified, the system may re-enter the planning phase.

## THE PURPOSE OF THIS PHASE IS TO:

- Operate, maintain, and enhance the system.
- Certify that the system can process sensitive information.
- Conduct periodic assessments of the system to ensure the functional requirements continue to be satisfied.
- Determine when the system needs to be modernized, replaced, or retired.

## Hardware & Software Specifications

### Hardware:

- Processor: Intel(R) Core (TM) i3-10100 CPU @  
3.60GHz 3.60 GHz
- Processor Speed: 533 MHz
- RAM : 4.00 GB
- System Type : 64-bit operating system
- Hard Disk Drive : 1 TB

### Software:

- Operating System : Windows 8.1
- IDE : IDLE Python
- Front End : Python 3.7
- Back End : MYSQL Server 5.0 or above

# **Working Environment**

## **What is Python?**

Python is a popular programming language. It was created by Guido van Rossum and released in 1991.

It is used to:

- Web development(server-side)
- Software development
- Data analysis
- System scripting

## **What python can do?**

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping or for production ready software development.

## **What is MySQL?**

MySQL is an open-source relational database management system. It is the most popular database system used with PHP. MySQL is developed, distributed, and supported by Oracle Corporation.

- The data in a MySQL database are stored in tables which consists of columns and rows.

- MySQL is a database system that runs on a server.
- MySQL is ideal for both small and large applications.
- MySQL is very fast, reliable, and easy to use database system. It used standard SQL.
- MySQL compiles on a number of platforms.

### **Interface Python with MySQL:**

MySQL python/connector is an interface for connecting to a MySQL database server from Python. It implements the Python Database API and is built on top of the MySQL.

The general workflow of a python program that interacts with a MySQL based database is as follows:

- Connect to the MySQL server.
- Create a new database
- Connect to the newly created or an existing database.
- Execute a SQL query and fetch results.
- Inform the database if any changes are made to a table.
- Close the connection to the MySQL server.

# **Functions and Modules used in Project**

## **1. MySQL Connector:**

This is used to connect MySQL databases with python.

## **2. Date time:**

Using date time module, current date and time will be displayed in the project.

## **3. Tabulate:**

Using tabulate module we can design the structure of table. Format of the table can be altered with this module.

## **4. Inflect:**

Using inflect module we can convert number into words.









