Library Book Issue System Portal

SOFTWARE ENGINEERING PROJECT REPORT

[Submitted in partial fulfilment]

As a part of the curriculum of

B.Sc. (H) COMPUTER SCIENCE

Submitted by:

Shreya Chauhan (21/5064)
Priya Bidhuri (21/5076)
Vaishnavi Arora (21/5078)
Manvi Tyagi (21/5085)
Bhavya Tyagi (21/5076)

B.Sc. (H) COMPUTER SCIENCE
Shyama Prasad Mukherjee College, West Punjabi Bagh, Delhi – 110026

ACKNOWLEDGEMENT

It gives me immense pleasure to present you this Library Book Issue System. We were lucky to get enormous support from extremely talented people, who deserve our great gratitude.

Firstly, we would like to thank our teacher and guide, Mr. Lavkush Gupta Sir who gave his valuable suggestions and ideas whenever we needed them. Also, he encouraged us to work on this project tirelessly by giving us numerous consultations.

We are also grateful to our parents for their constant support, guidance and providing us the necessary resources for the project.

Lastly, we would like to thank our classmates for their valuable suggestions for the betterment of this project and everybody who has helped us directly or indirectly in completion of this project.

We are immensely grateful to everyone involved in this project a without their inspiration and valuable suggestion it would not have been possible to develop the project within the prescribed time.

With sincere thanks,

Shreya Chauhan (21/5064)
Priya Bidhuri (21/5076)
Vaishnavi Arora (21/5078)
Manvi Tyagi (21/5085)
Bhavya Tyagi (21/5076)

CERTIFICATE

This is to certify that the project entitled, "Library Book Issue System", has been submitted by Shreya Chauhan, Priya bidhuri, Vaishnavi Arora, Manvi Tyagi and Bhavya Tyagi in partial fulfillment of the requirements of Bachelor of Computer Science (Hons.) embodies the work done by them during, semester IV of their course under the supervision of Mr. Lavkush Gupta, Assistant Professor, Department of Computer Science, Shyama Prasad Mukherjee College for Women, University of Delhi.

Mr. Lavkush Gupta
(Project Guide)

Table of contents

PROBLEM STATEMENT

It is a College Library Portal only accessible by college administration developed to facilitate the library book issue system to manage the information of book that has been issued, misplaced, not returned by the student, and available to issue. It also manages fine levied on the student who has not returned the book on time.

It aims in providing an easy, quick way to manage the college library book system. This portal will allow the administrator to see the information of books issued, misplaced, and fine on the student. The portal provides an option to keep a record of which student issued which book with the help of a book id. The administrator can log in and use the portal to store student info on who has issued the book and to keep a record of who has not returned the book.

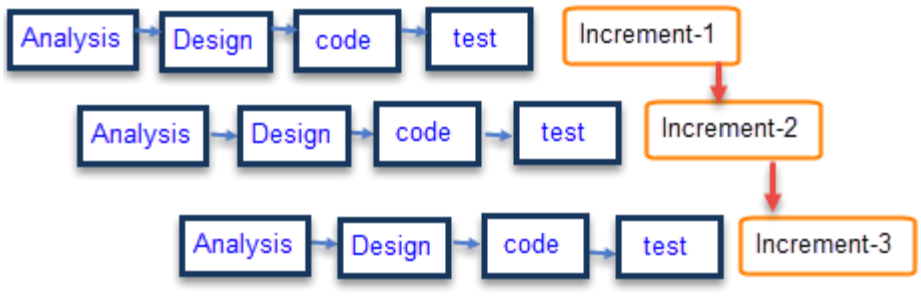
Admin will be responsible for maintaining the portal and will also have access to the whole system.

PROCESS MODEL

Model best suited for this project is Incremental Process model. We have used the incremental model as it combines elements of linear and parallel process flows. It generates working software quickly and early during the software lifecycle. This model is more flexible and less costly to change scope and requirements. It is easier to test and debug during a smaller iteration. In this model, the customers can respond to each built. Also, functionality can be refined and expanded in the later stages in the later software releases. The user can visualize the software before the completion of the entire project in order to evaluate and provide feedback. We are using this model as requirements are completely understood, however, small changes can be incorporated.

Advantages of Incremental Model

- **Errors are easy to be recognized.**
- **Easier to test and debug**
- **More flexible.**
- **Simple to manage risk because it handled during its iteration.**
- **The Client gets important functionality early.**

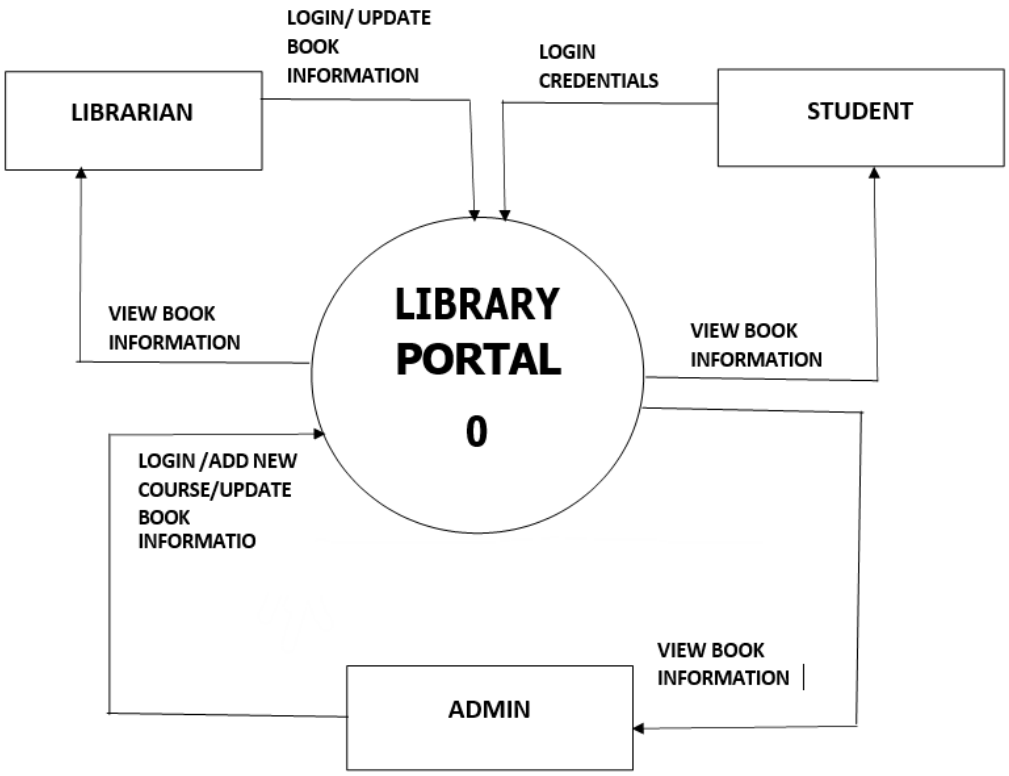


Incremental Model

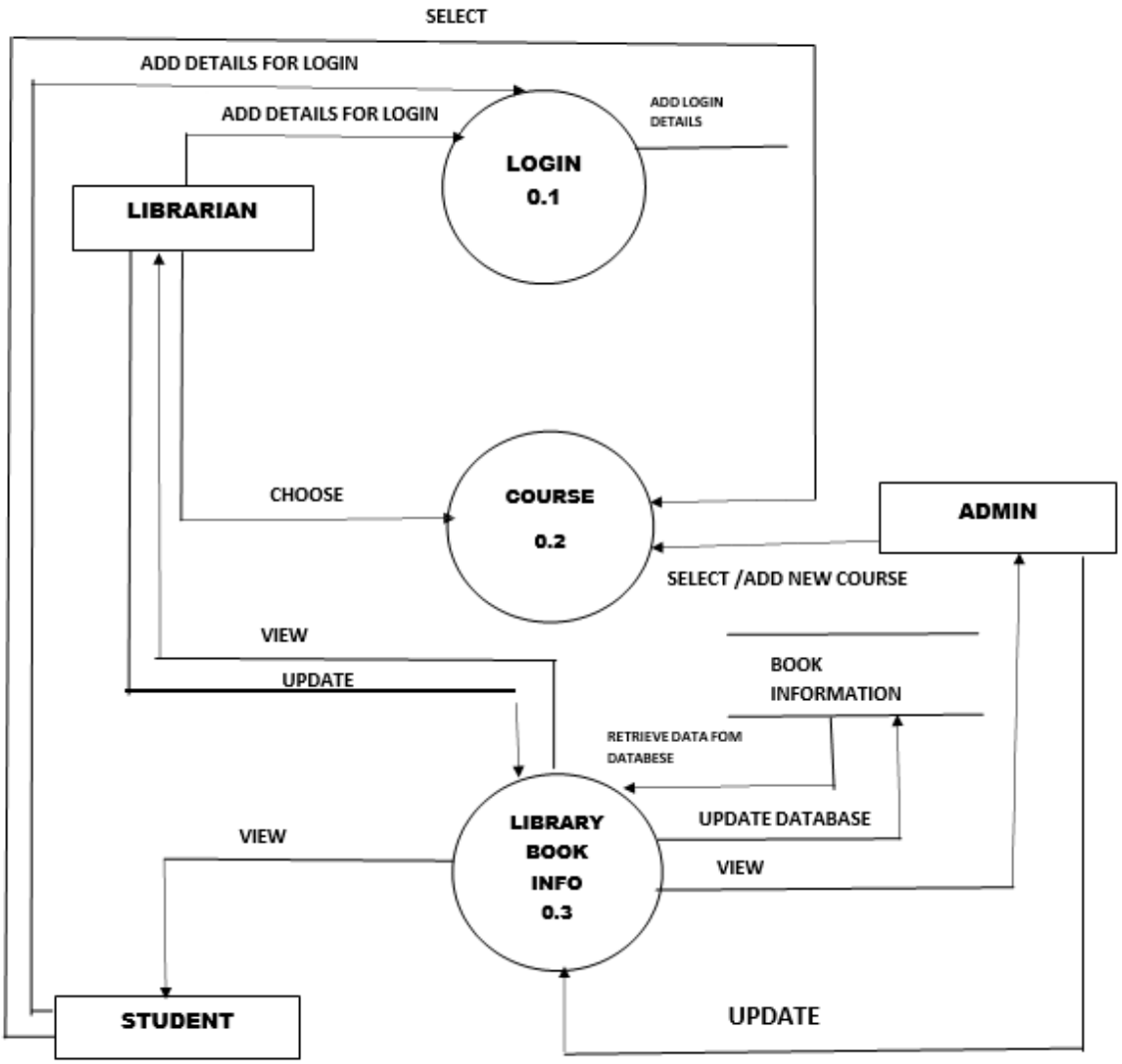
REQUIREMENT ANALYSIS

DFD (DATA FLOW DIAGRAMS)

Context level DFD

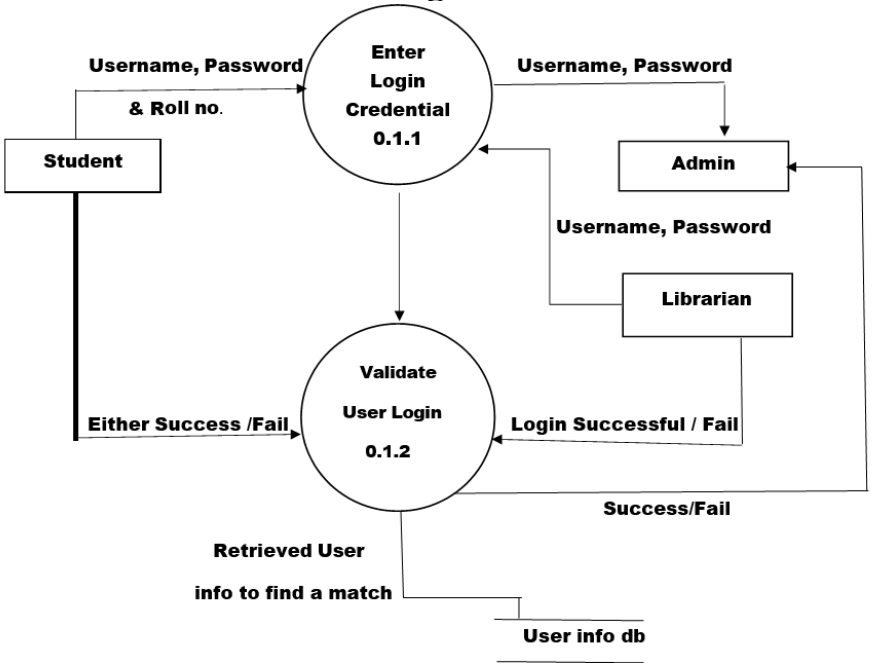


Level 1 DFD

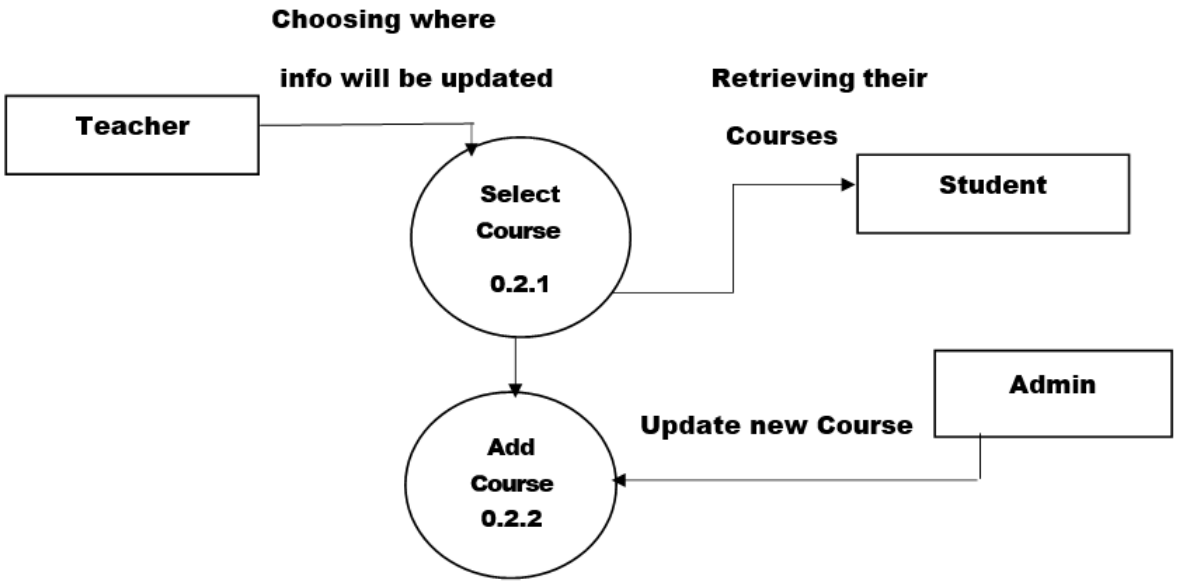

SELECT
ADD DETAILS FOR LOGIN
ADD DETAILS FOR LOGIN
ADD LOGIN DETAILS
LOGIN 0.1
LIBRARIAN
CHOOSE
COURSE 0.2
ADMIN
SELECT /ADD NEW COURSE
VIEW
UPDATE
BOOK INFORMATION
RETRIEVE DATA FOM DATABESE
UPDATE DATABASE
VIEW
VIEW
LIBRARY BOOK INFO 0.3
STUDENT
UPDATE

Level 2 DFD

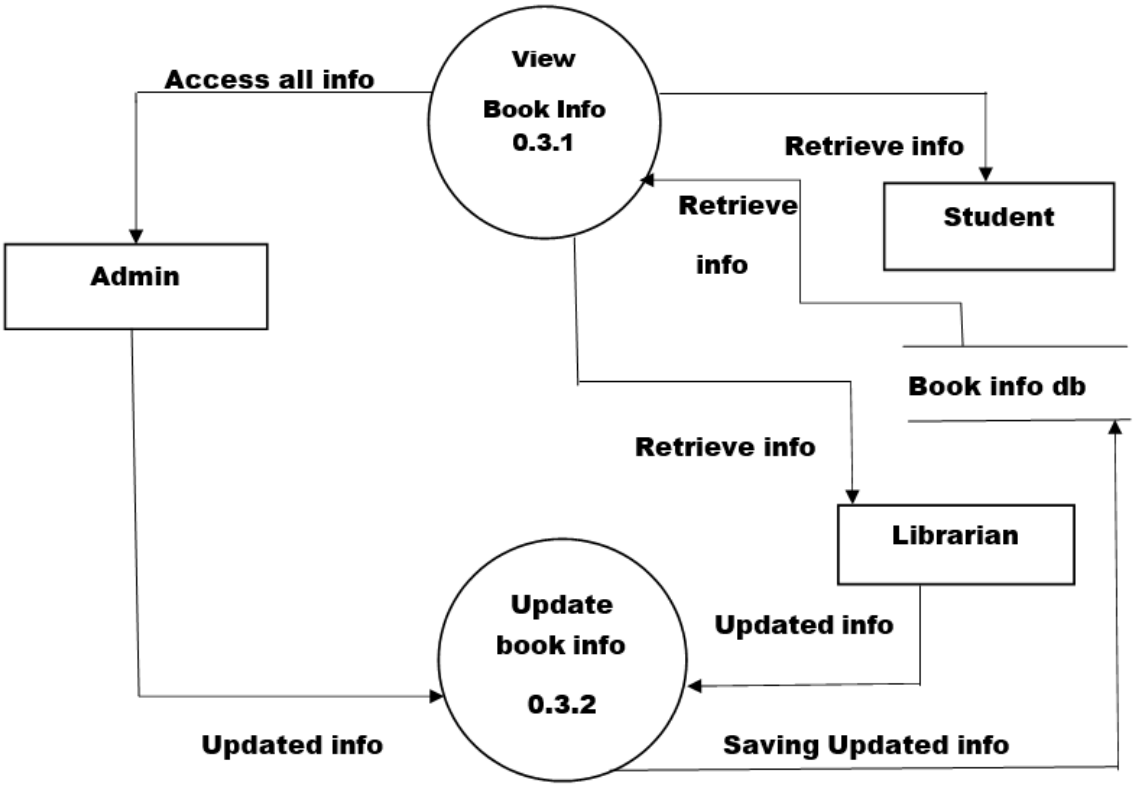
Login –



Select /Updating Course –



Book info {Book issued, book available, book returned} –



DATA DICTIONARY

Legal character: [a-z|A-Z]

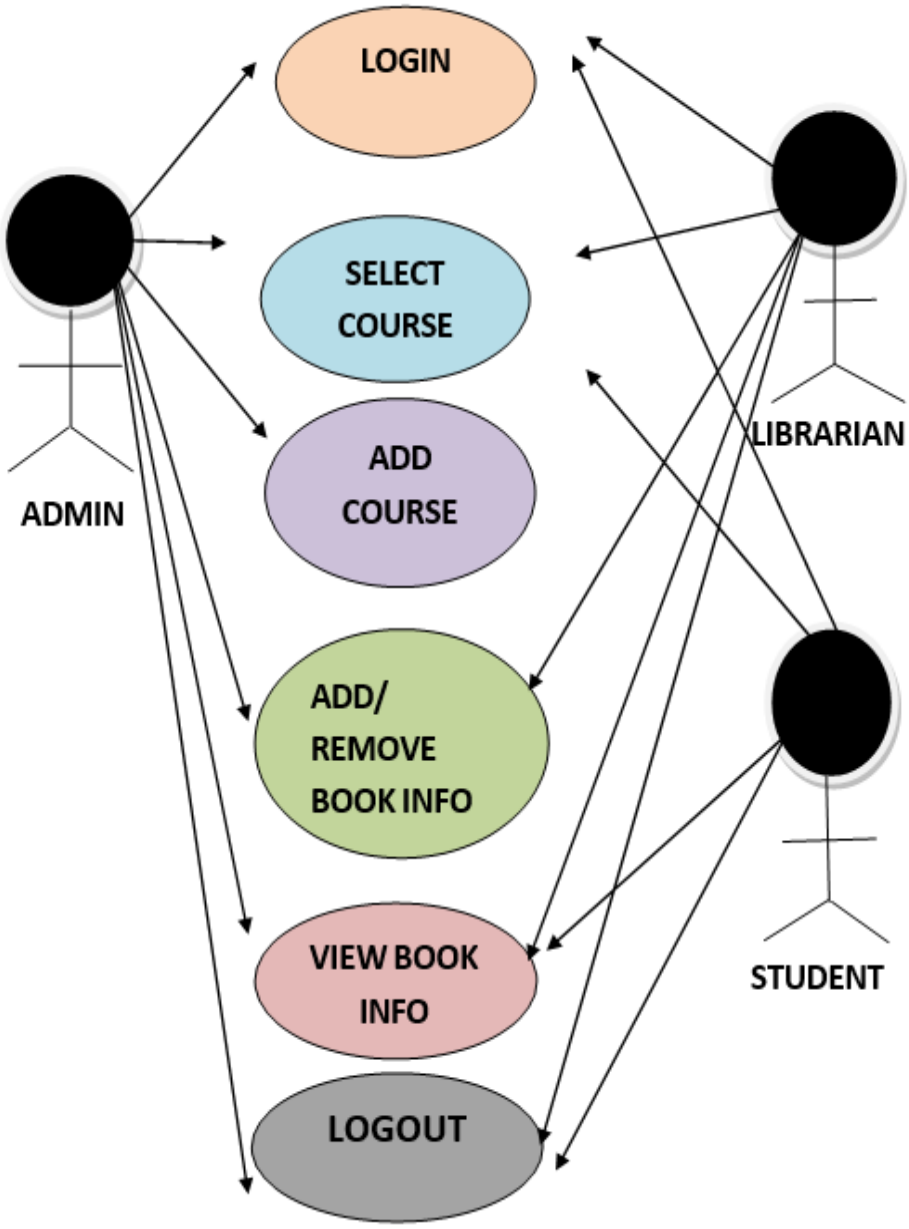
Digit: [0-9]

Special character: [@|$|#|+|-|.]

1	**Username**	**{Legal character} ***
2	**Password**	**{Legal character + Digit + Special character} ***
3	**BId**	**{Digit + Digit + Digit} ***
4	**Book Name**	**{Legal character} ***
5	**Book Writer**	**{Legal character} ***
6	**Student roll number**	**{Digit + Digit + Special character + Digit + Digit + Digit + Digit} ***
7	**Issued date**	**{Digit + Digit + Digit} ***
8	**Return date**	**{Digit + Digit + Digit} ***

USE CASE

Use Case Diagram –



Use Case Descriptions –

1. **Login:**

Brief description: A way to use portal is user id & password.

Actors: Admin, Librarian and Student.

Flow of events:

- Basic Flow- System verifies the user id and password enter by the actor. After successful verification actor is logged into the system.
- Alternative Flow- Invalid user id and password. The system will display ERROR.

Special Requirement: User name and password must be known to access portal.

Pre-Condition: Actor must have login credentials.

Post Condition: If login details are correct, actor can log into system. If not, the system will remain same.

2. **Select Course:**

Brief description: A way to select course whose subject book you want to issue.

Actors: Admin, Librarian & Student.

Flow of events:

- Basic Flow- Selected course exist.
- Alternative Flow- Course which you want to select doesn't exist.

Special Requirement: None

Pre-Condition: Course must exist.

Post Condition: None.

3. **Add Course:**

Brief description: Only admin can add new course if required.
Actors: Admin.
Flow of events:

- Basic Flow- None.
- Alternative Flow- None.

Special Requirement: None
Pre-Condition: None
Post Condition: None.

4. **Add / Remove book info:**

Brief description: Only admin or librarian can add or remove book info(which includes which book is available, already issued & not returned on time).
Actors: Admin, Librarian.

Flow of events:

- Basic Flow- Book info (like book id).
- Alternative Flow- Incorrect book id.

Special Requirement: None
Pre-Condition: Book info must be known.
Post Condition: None.

5. **View Book info:**

Brief description: Student, Librarian or Admin can view the book information any time.
Actors: Admin, Librarian & Student.

Flow of events:

- Basic Flow- Book info exist.
- Alternative Flow- Book info doesn't exist.

Special Requirement: None
Pre-Condition: None
Post Condition: None.

6. **Log out:**

Brief description: Admin, Librarian or Student can log out any time.
Actors: Admin, Librarian & Student.

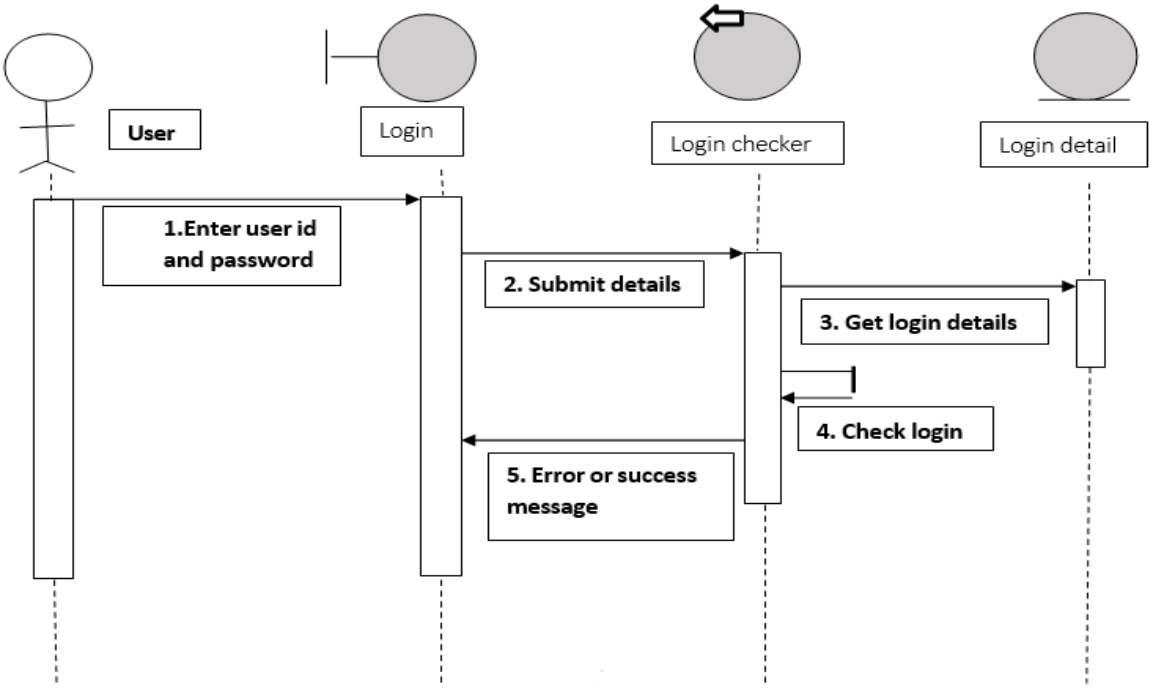
Flow of events:

- Basic Flow- None.
- Alternative Flow- None.

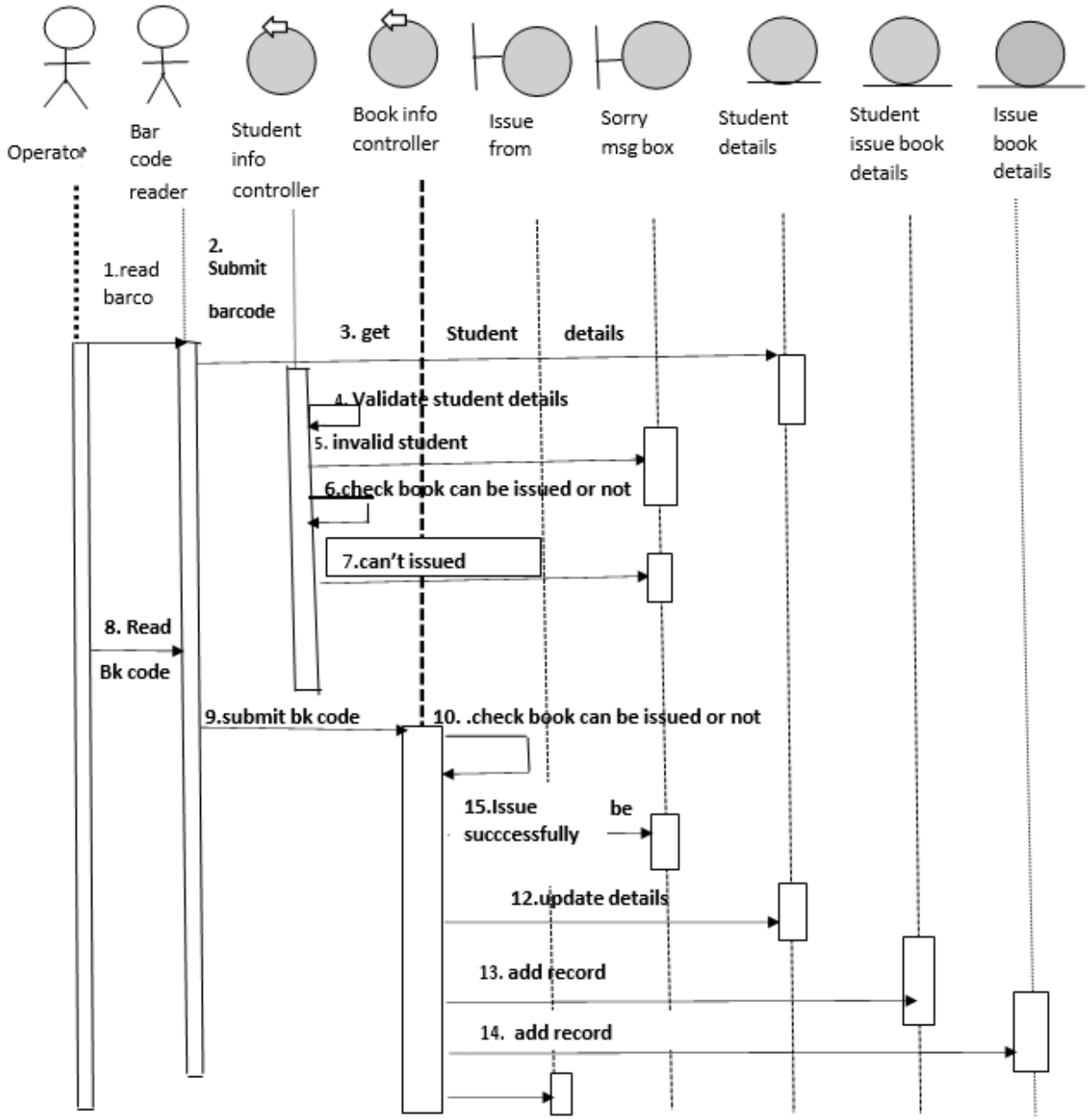
Special Requirement: None.
Pre-Condition: User may log out after accessing system.
Post Condition: None.

SEQUENCE DIAGRAM

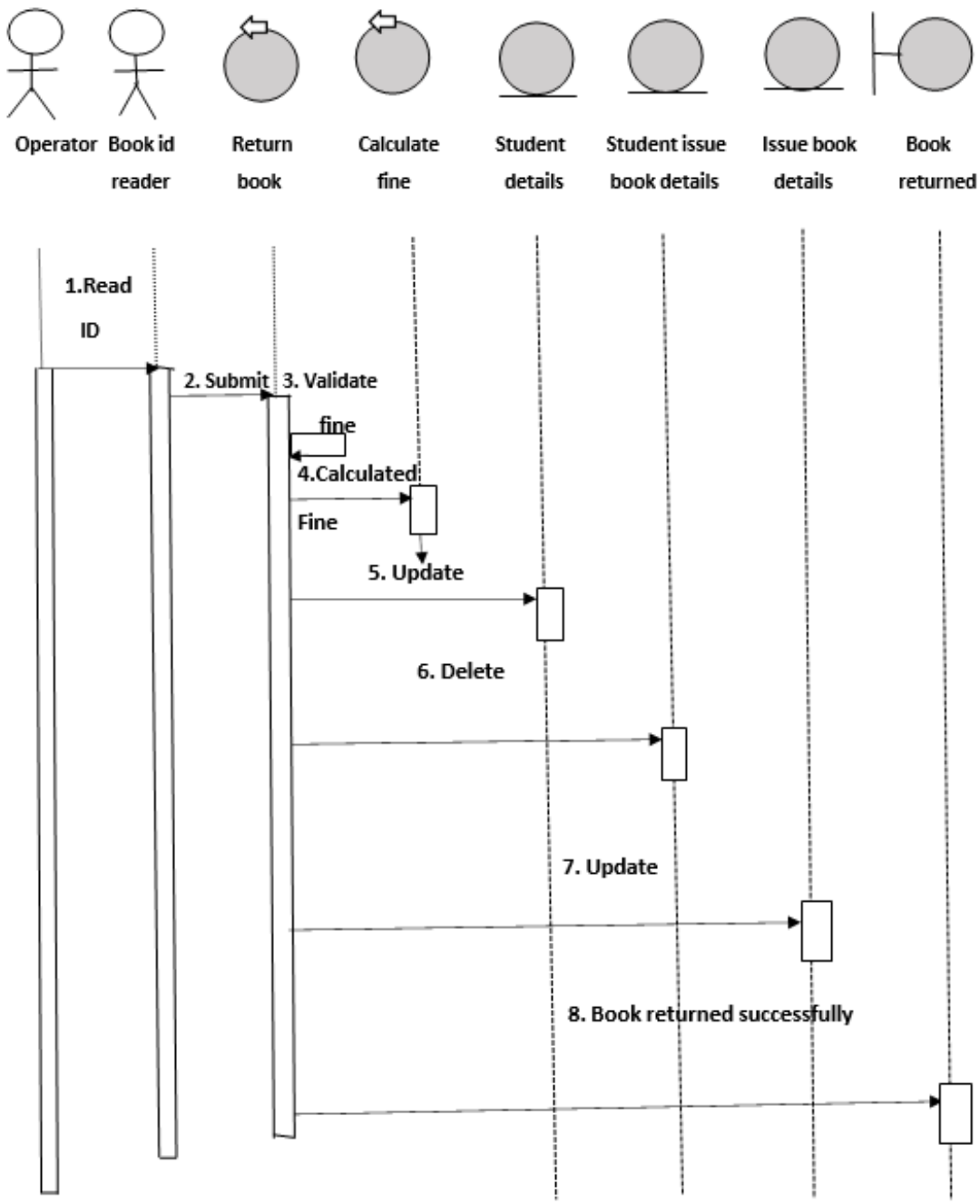
Login –



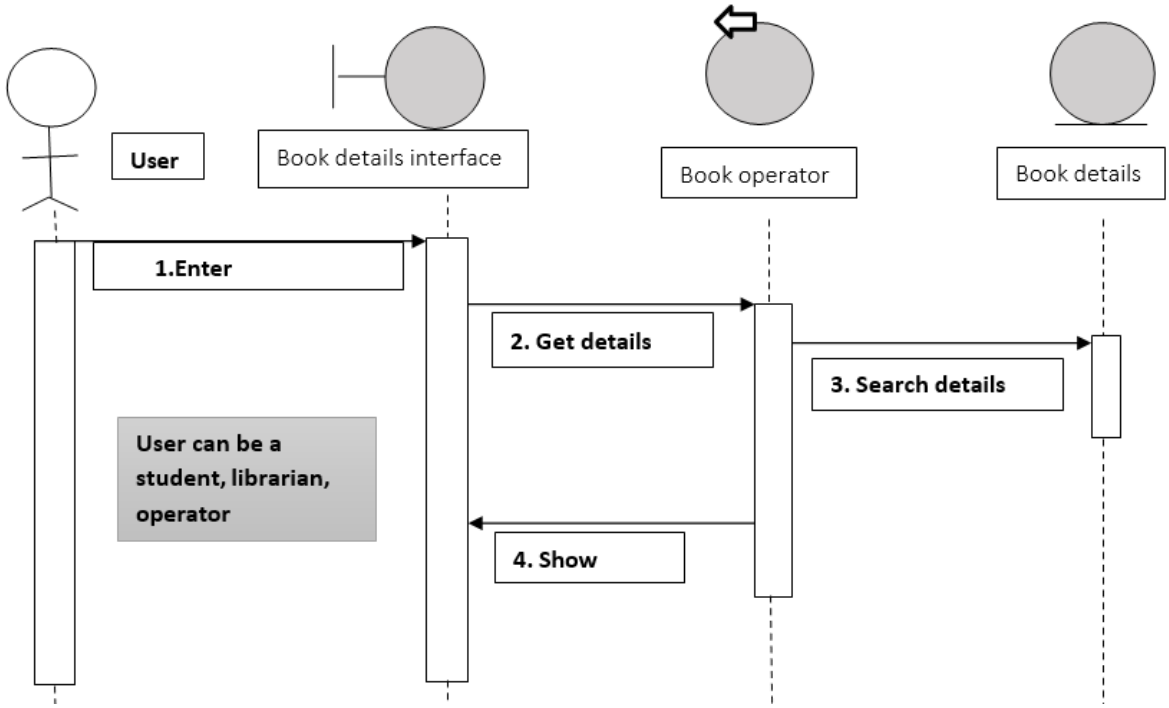
Issue book –


Operator
Bar code reader
Student info controller
Book info controller
Issue from
Sorry msg box
Student details
Student issue book details
Issue book details
1.read barco
2. Submit barcode
3. get Student details
4. Validate student details
5. invalid student
6.check book can be issued or not
7.can't issued
8. Read Bk code
9.submit bk code
10. .check book can be issued or not
15.Issue be succcessfully
12.update details
13. add record
14. add record

Return book –


Operator
Book id reader
Return book
Calculate fine
Student details
Student issue book details
Issue book details
Book returned
1.Read ID
2. Submit
3. Validate
fine
4.Calculated Fine
5. Update
6. Delete
7. Update
8. Book returned successfully

Search/Query about book –



SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

1. INTRODUCTION

This section provides the scope, description and everything present in this document including the purpose of the project.

1.1 PURPOSE

The purpose of this project is to build an online system to facilitate the college library book issue system that includes: - managing the information of the books issued, returned, misplaced, and books available to be issued.

The primary aim is to provide an easy, quick way to manage the college library book system.

1.2 SCOPE

Library Management system is basically updating the manual library system into an internet- based application so that users can know the details of their accounts, availability of books and maximum limit of borrowing.

According to the customer requirements the software to be developed will consist of 2 databases:-

- Item's database (books, journals, magazines, newspapers, diploma thesis, etc)

- Student's database

LMS will provide all necessary services for databases such as creating, deleting, updating and searching information.

Also, users can provide feedback for adding some new books to the library.

1.3 DEFINITIONS, ACRONYMS/ABBREVIATIONS

User: The people who interact directly with the project.

Here the users are:

Administrator: The one who manages, deletes and updates the records of the database.

Students who wants to issue the book

- JAVA- platform independency
- SQL- Structured Query Language
- HTML- Hyper Text Markup Language
- CSS- Cascading Style Sheets
- PHP- Hyper Text Pre-Processor
- SRS- Software Requirements Specification
- GUI- Graphic User Interface
- LMS- Library Management System

1.4 REFERENCES

https://www.geeksforgeeks.org/how-to-write-a-good-srs-for-your-project/

https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

1.5 OVERVIEW

The implementation of the Library Management System starts with entering and updating master records like book details, library information. Any further

transaction like book issue, book return will automatically update the current books.

2. OVERALL DESCRIPTION

This section will give the overview of the whole system. This section basically describes the software's principal functions. The system will be explained in context to show how the system interacts with other systems and description of this section would cover the system interfaces, hardware and network interfaces, system memory demands and site specific requirements. It will also describe the type of stakeholders that will use the system and what type of functionalities are available to them. At last, the description with limitations that affect the software's development will be presented. These constraints might include hardware requirements, reliability plus security elements, auditing requirements etc.

2.1 PRODUCT PERSPECTIVE

Library book issue system will take care of the current book detail at any point of time. The book issue will update the book details automatically so that user will get the update current book details.

Actors : Admin, Librarian & Student.

- The computerization of the library book issue system will reduce a lot of paperwork and hence the load on the library staff.
- It would be easy to search about a specific book according to the selected course.
- The system provides for User id validation(user id & password). Hence unauthorized access is prevented.

2.1.2 SYSTEM INTERFACES

The software will work over the latest version of chrome or Firefox browser on Windows, Linux and Mc.

2.2.2 USER INTERFACES

The software provides graphical interface for the user and the admin can operate on the system , performing the required tasks such as login , select course , add course, add/remove book info , view book info , logout.

- Allow user(student) to login, select course, view book info and logout.

2.1.3 HARDWARE INTERFACES

- Operating System: Window
- Hard disk: 40 GB
- RAM : 256 MB or more
- Processor: Pentium (R) Dual Core CPU

2.1.4 SOFTWARE INTERFACES

If the software is on server so it requires the support of scripting languages like Html, JavaScript and PHP 7.

The version of system's software will be compatible to the application to make it accessible .

2.1.5 COMMUNICATION INTERFACES

Window

2.1.6 MEMORY CONSTRAINS

The system is expected to have a memory of 256 MB and disk space of 500 MB. But it is recommended that the system has memory capacity of 1 GB and disk space of 1 GB.

2.1.7 OPERATIONS

The base operations of the Library Book issue system are described as follow:

- The staff member can login into the system with his/her username and password.
- The staff member can then select course , add course, add/remove book info , view book info , logout.
- The staff member can issue books when requesting by the student and then update the book records too.

- The staff member will update the book issue and return records into the Library Book issue system.
- The student can select course and view book info.

2.1.8 SITE ADAPTATION REQUIREMENTS

The system will require an application server for the runtime components and a database for storage. The system will run on select popular application servers and use select popular database for data storage.

2.2 PRODUCT FUNCTIONS

2.2.1 ADMINISTRATOR

1)Admin should be able to insert modify and delete books.

2) Can accept or reject user according to the information that is store in database.

3)Increase the period for borrowing a book for specific type or group of users.

4)Can get the information(status report)of any member who has borrowed a book .

5)Add and edit book categories and arrange books by categories.

6) Add and edit authors and publishers information.

7) Can record books returned by the users.

2.2.2 USERS(STUDENTS)

1) The student should be provided with the updated information about the books catalog.

2) Students are given a provision to check their information.

3)The student may suggest a book to be brought to the library book collection.

4) Student can check fine levied on them.

2.3 USER CHARACTERISTICS

We have 2 levels of users

1) User module:- In the user module, user will check the availability of the book.
   -book available
   -book issue
   -book not return
   2) Administration module:-The following are the sub module in the administration module.
   - register user
   -enter book details
   -book issued
   -add book
   - delete book
   -add course

2.4 CONSTRAINTS

1) Any update regarding the book from the library is to be recorded to have update and correct values.

2) The information of all the users , books and the libraries must be stored in a database that is accessible by the website.

3) The online library system is running 24 hours a day .

4) User may access LMS from any computer that has Internet browsing capabilities and an Internet connection .

5)User must have their correct usernames and passwords to enter into the online account and do actions.

2.5 ASSUMPTIONS AND DEPENDENCIES

The success of the system depends on

1) Existence of an internet service to all the people .
2) Are librarians and users comfortable with the computer and have enough ability to work with the product ?
3) Website interface must be friendly and easy to use.
4) The search mechanism should be simple and fast.

3 SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACES

3.1.1 SIGN UP

- **Input:** detail about the user (librarian and students) as mentioned in the description.
- **Output:** confirmation of the registration status.
- **Processing:** all the details will be checked and if any error are found then an error will display else confirm the registration.

3.1.2 LOGIN

- **Input:** enter the id and password created.
- **Output:** user will be able to use the feature of the software.

3.1.3 MANAGE BOOK BY STUDENT

BOOK ISSUED

- **Description:** list of books will be displaced along with date of issue

SEARCH

- **Input:** enter the book id of the books to be issued.
- **Output:** list of the books related to the keyword.

ISSUES BOOK

- **State:** search the book student want to issues.
- **Input:** click the book student wants.
- **Output:** conformation for the book issue and apology for failure in issue.

RENEW BOOK

- **State:** book is issued and is about to reach the date of return.
- **Input:** select the book to be renewed.
- **Output:** conformation message

RETURN

- **Input:** return the book to the library.
- **Output:** issued list will be updated.

3.2 FUNCTIONAL REQUIREMENTS

- The library management system allows searching of books by author, title or keywords
- The library management system should allow librarian to add, delete and modify books in database and check the availability of the books
- The library management system allow student to register by fill required details.
- The library management system allow student to login by using college id and password after signup.
- Students can search for book.

3.3 PERFORMANCE REQUIREMENTS

- The performance of the system should be faster and accurate.
- Server will be working 24x7.
- The system shall handle the expected and non-expected error. It should have inbuilt error testing to identify invalid id and password.
- Throughput: the system should be able to handle large amount of data. Thus it should accommodate high number of books and users without any fault.

3.4 LOGICAL DATABASE REQUIREMENTS

The library management system software database includes:

- Login table: it consists of details of id and passwords
- Book available table: it consists of details of book id, book name and book writer.
- Book issued table: it consists of details of book id , student id, issued date and return date
- Book not return table: it consists of details of book id, student id and fine.

3.5 DESIGN CONSTRAINTS

Software language used: in our project we used HTML, CSS and PHP.
Database design: in our database design, we used MYSQL.

3.6 STANDARD COMPLIANCE

Report format: All the reports produced for this project are in compliance with the standard templates in accordance with the standard guidelines and policy.

Naming Conventions: All the documents are named using the standard naming convention

3.7 SOFTWARE SYSTEM ATTRIBUTES [NONFUNCTIONAL REQUIREMENTS]

Performance Requirements

- the system shall accommodate high number of books and users without any fault.
- responses to view information shall take no longer than 5 seconds to appear on the screen.

Safety Requirements

- system use shall not cause any harm to human users.

Security Requirements

- system will use secured database
- student can just read information but they cannot edit or modify anything except their personal and some other information.
- system will have different types of users and every user has access constraints.

Error Handling

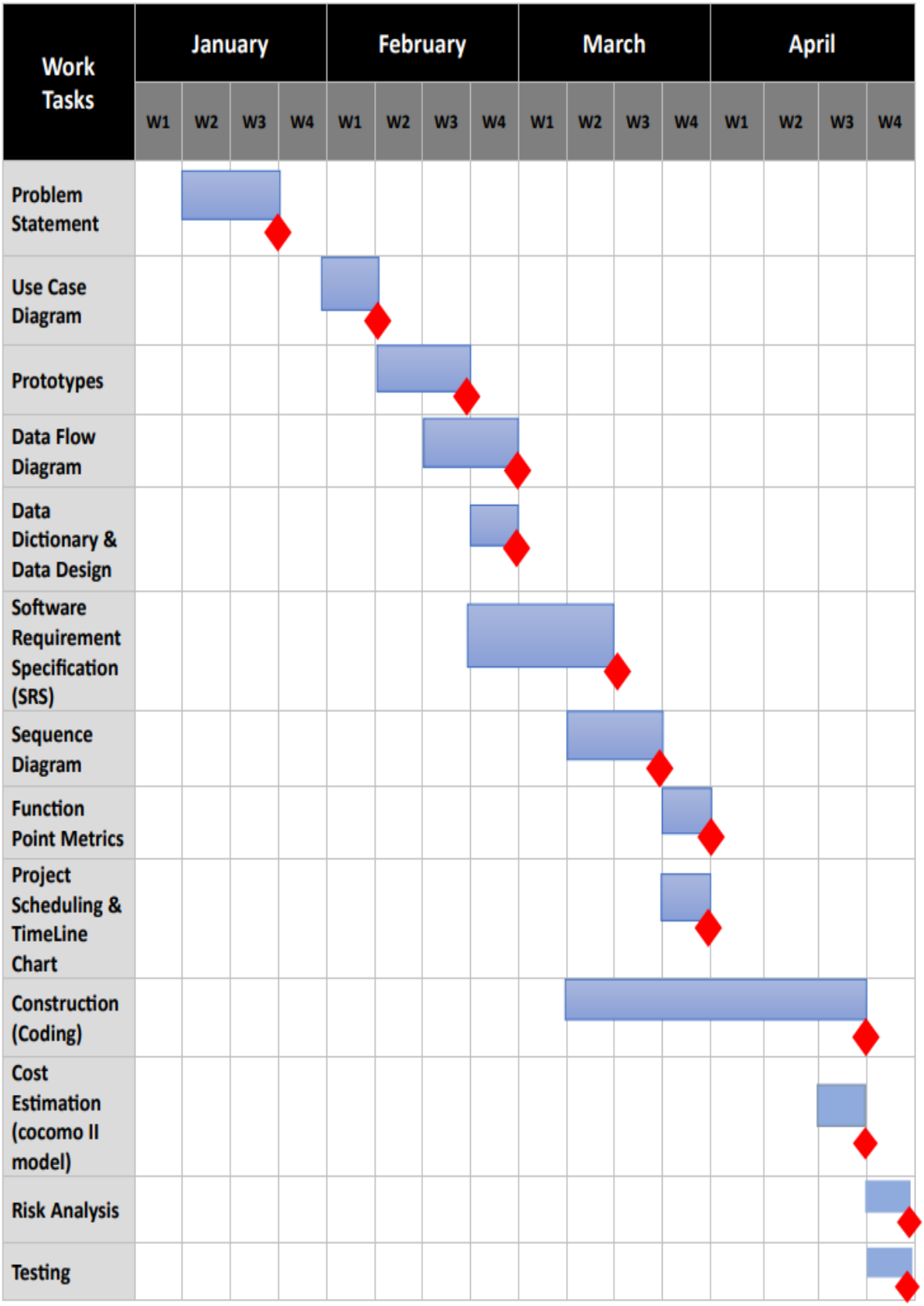
- software shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period.

PROJECT PLANNING

PROJECT SCHEDULING

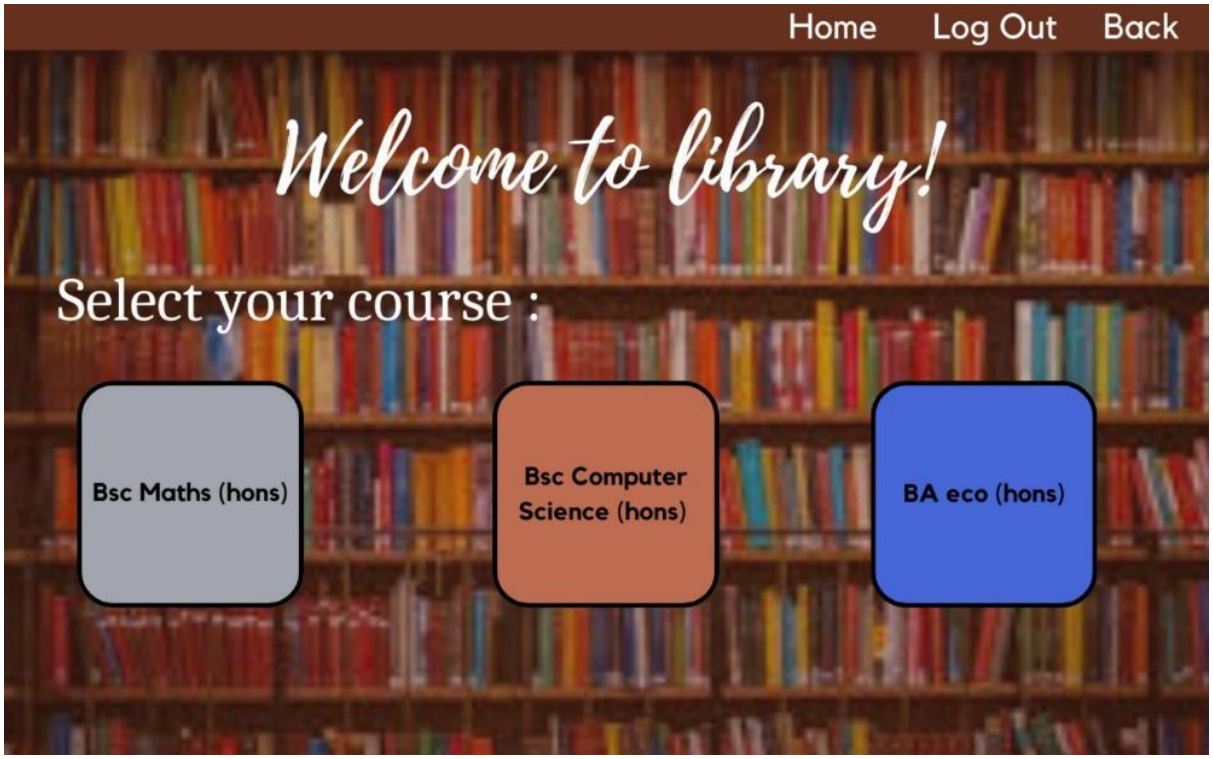
	Planned Start	Actual Start	Planned Complete	Actual Complete	Assigned Person(s)	Effort Allocated
Problem Statement	January, w2	January, w2	January, w3	January, w3	Vaishnavi Arora Shreya Chauhan	2 person per week
Use Case Diagram	February, w1	February, w1	February, w2	February, w1	Manvi Tyagi Bhavya Tyagi	2 person per week
Prototypes	February, w2	February, w2	February, w3	February, w3	Priya Bidhuri Vaishnavi Arora Shreya Chauhan Manvi Tyagi Bhavya Tyagi	5 person per week
Data Flow Diagram	February, w3	February, w3	February, w4	February, w4	Vaishnavi Arora Shreya Chauhan Manvi Tyagi	3 person per week
Data Dictionary & Data Design	February, w4	February, w4	February, w4	February, w4	Priya Bidhuri Bhavya Tyagi	2 person per week
Software Requirements Specification (SRS)	February, w4	February, w4	March, w2	March, w2	Priya Bidhuri Vaishnavi Arora Shreya Chauhan Manvi Tyagi Bhavya Tyagi	5 person per week
Sequence Diagram	March, w2	March, w2	March, w3	March, w3	Priya Bidhuri Vaishnavi Arora Shreya Chauhan Manvi Tyagi Bhavya Tyagi	5 person per week
Function Point Metrics	March, w4	March, w4	March, w4	March, w4	Priya Bidhuri	1 person per week
Project Scheduling & TimeLine Chart	March, w4	March, w4	March, w4	March, w4	Vaishnavi Arora	1 person per week
Construction (Coding)	March, w2	March, w2	April, w3	April, w3	Priya Bidhuri Vaishnavi Arora Shreya Chauhan Manvi Tyagi Bhavya Tyagi	5 person per week
Cost Estimation (cocomo II model)	April, w3	April, w3	April, w3	April, w3	Vaishnavi Arora Bhavya Tyagi	2 person per week
Risk Analysis	April, w4	April, w4	April, w4	April, w4	Shreya Chauhan Priya Bidhuri	2 person per week
Testing	April, w4	April, w4	April, w4	April, w4	Vaishnavi Arora Manvi Tyagi	2 person per week

TIMELINE CHART

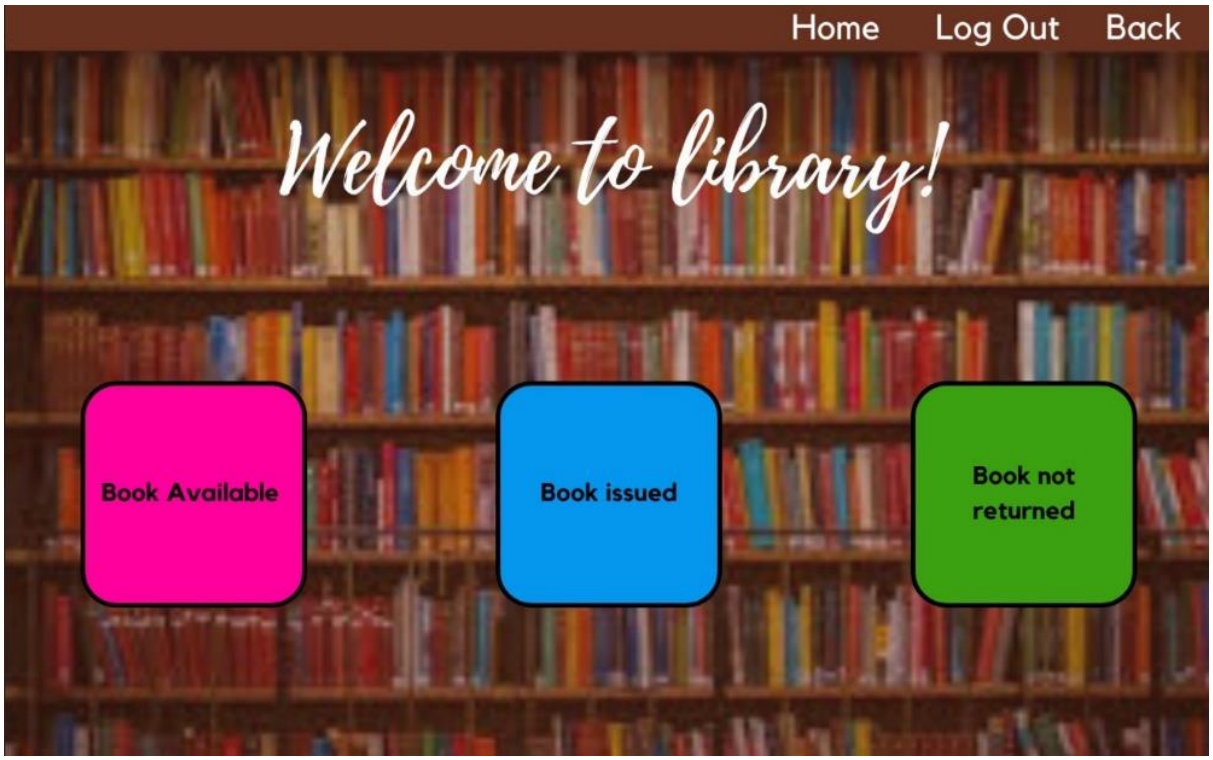


FUNCTION POINT METRICS

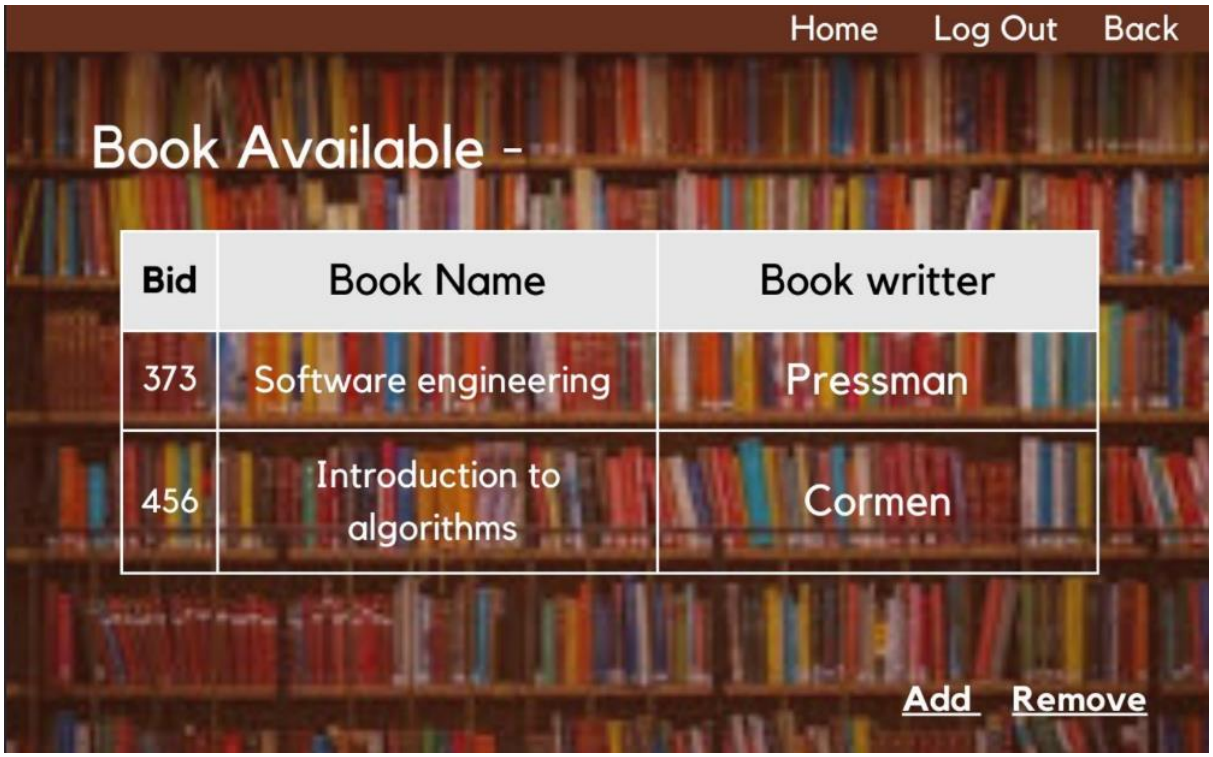
Number of External Inputs: 2
Number of External Outputs: 1
Number of External Inquiries: 0
Number of Internal Logical Files: 1
Number of External Interfaces: 0



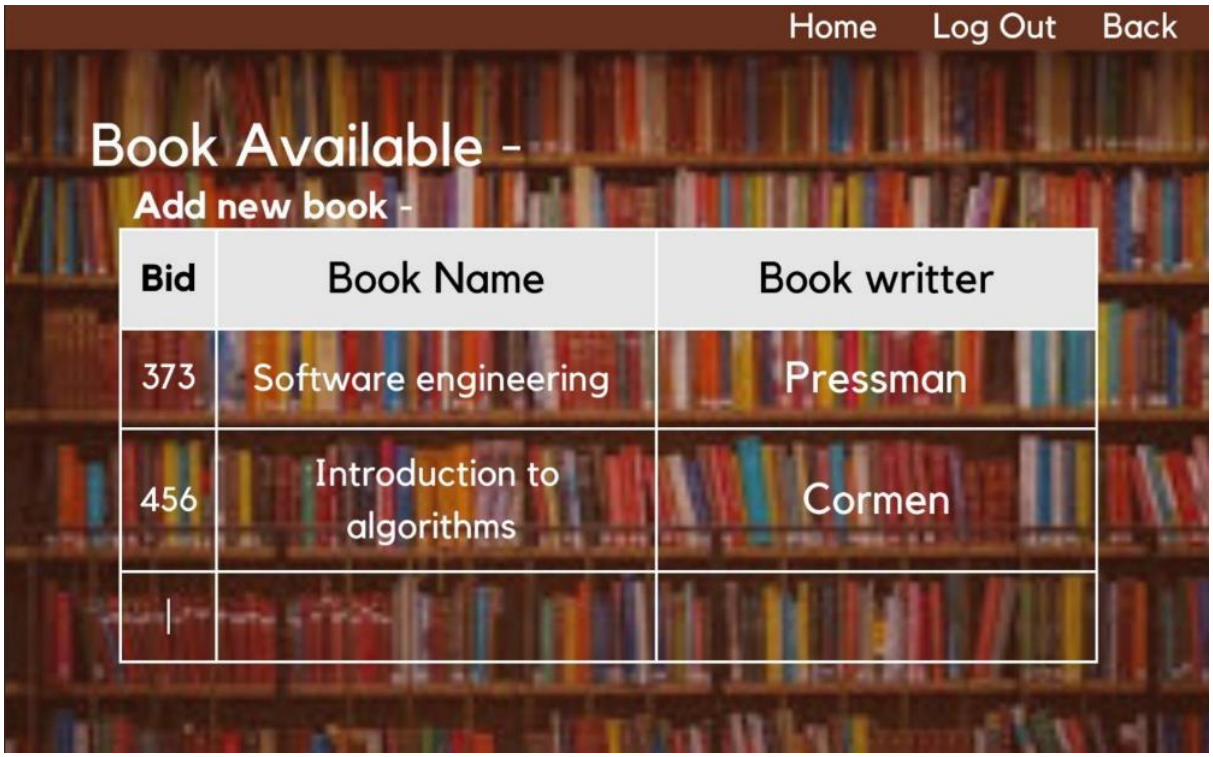
Number of External Inputs: 0
Number of External Outputs: 1
Number of External Inquiries: 0
Number of Internal Logical Files: 0
Number of External Interfaces: 0



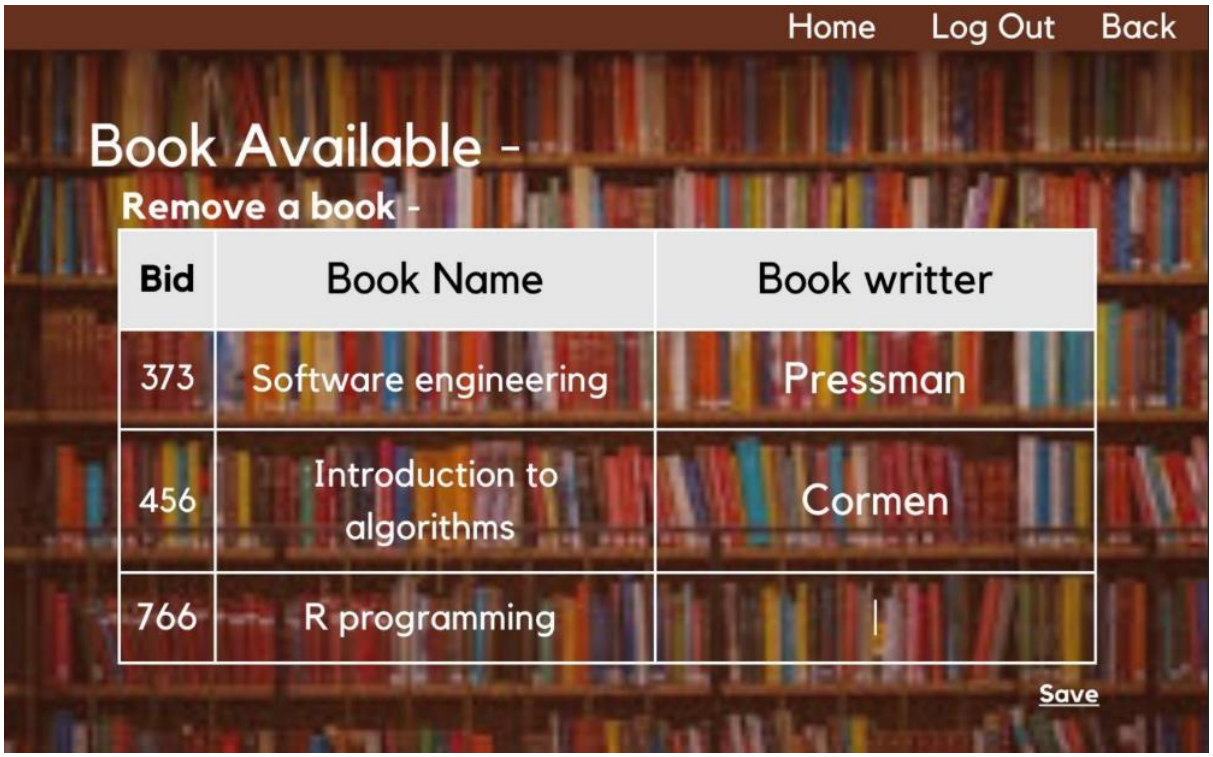
Number of External Inputs: 0
Number of External Outputs: 1
Number of External Inquiries: 1
Number of Internal Logical Files: 0
Number of External Interfaces: 0



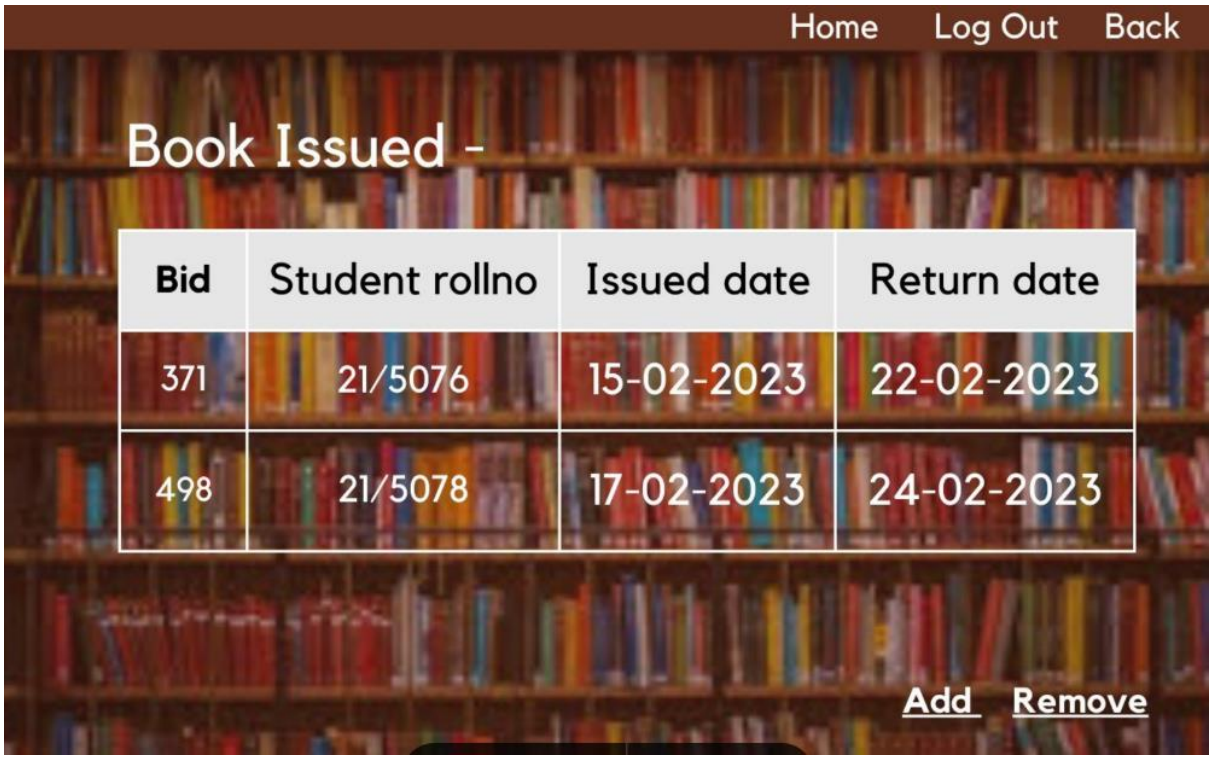
Number of External Inputs: 0
Number of External Outputs: 1
Number of External Inquiries: 0
Number of Internal Logical Files: 0
Number of External Interfaces: 0



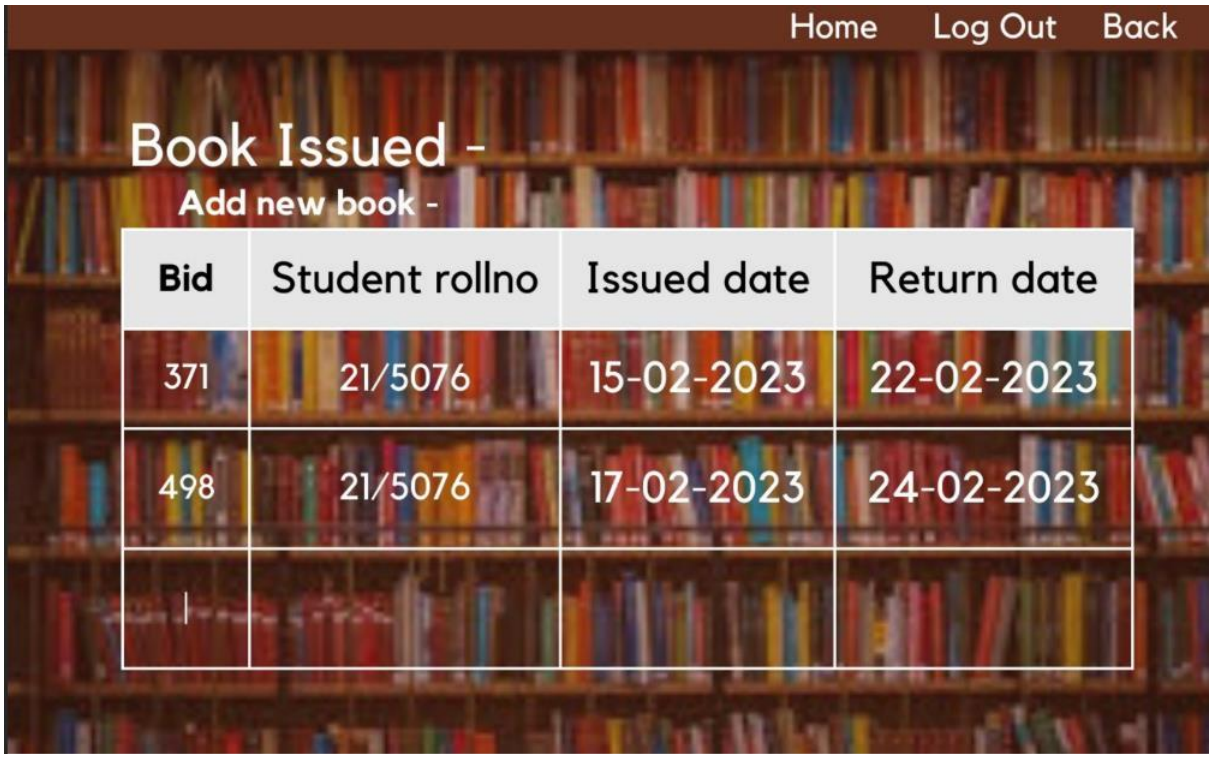
Number of External Inputs: 3
Number of External Outputs: 1
Number of External Inquiries: 0
Number of Internal Logical Files: 1
Number of External Interfaces: 0



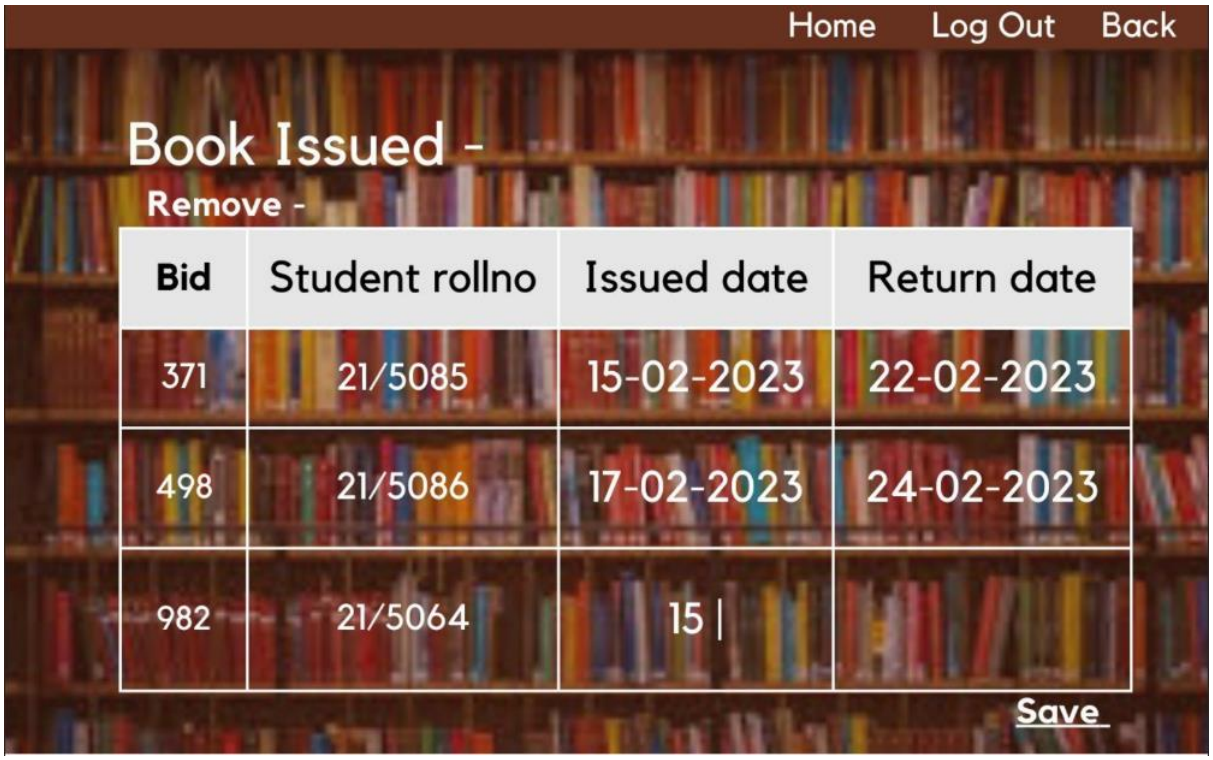
Number of External Inputs: 0
Number of External Outputs: 1
Number of External Inquiries: 0
Number of Internal Logical Files: 1
Number of External Interfaces: 0



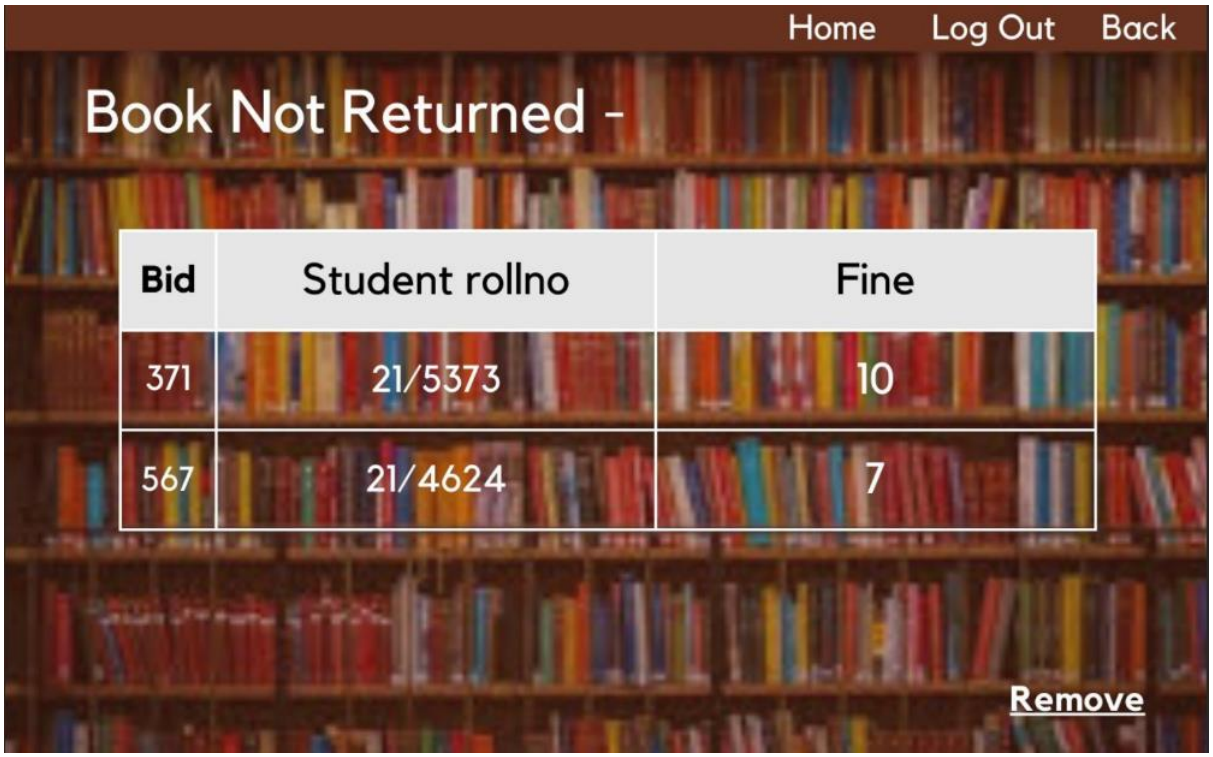
Number of External Inputs: 0
Number of External Outputs: 1
Number of External Inquiries: 0
Number of Internal Logical Files: 1
Number of External Interfaces: 0



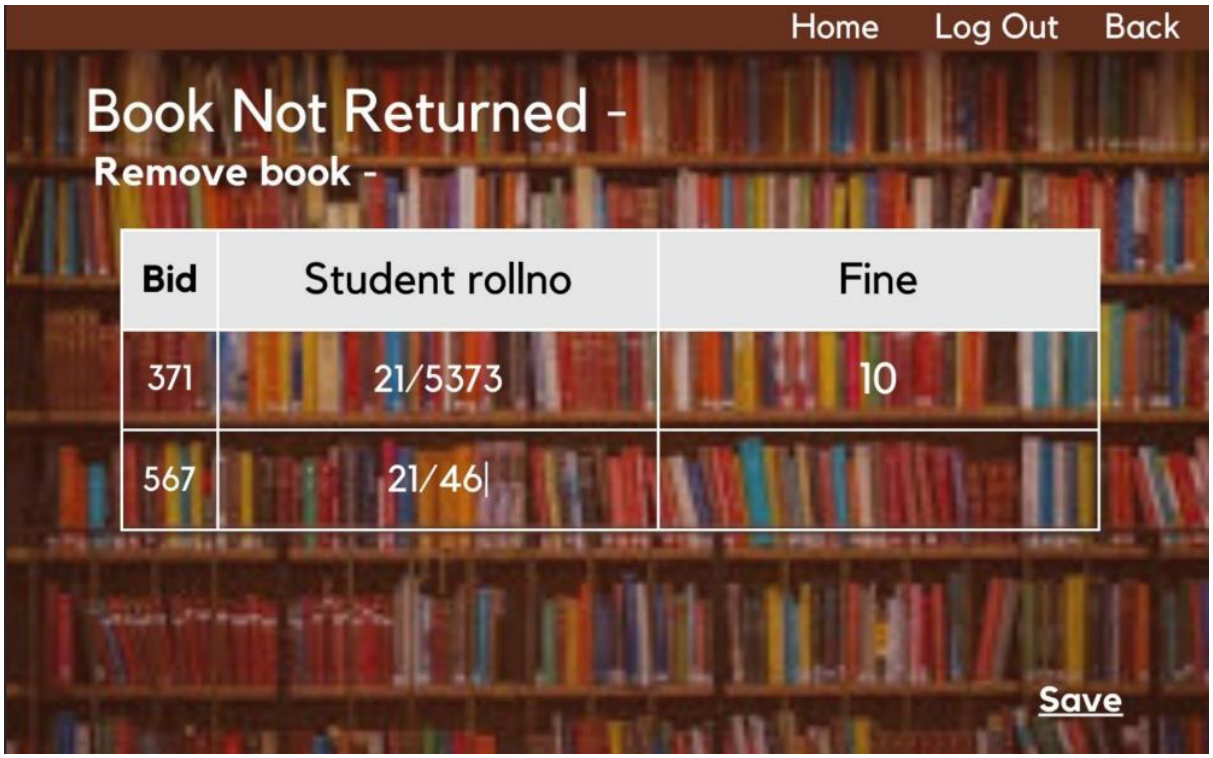
Number of External Inputs: 4
Number of External Outputs: 1
Number of External Inquiries: 0
Number of Internal Logical Files: 1
Number of External Interfaces: 0



Number of External Inputs: 0
Number of External Outputs: 1
Number of External Inquiries: 0
Number of Internal Logical Files: 1
Number of External Interfaces: 0



Number of External Inputs: 0
Number of External Outputs: 1
Number of External Inquiries: 0
Number of Internal Logical Files: 1
Number of External Interfaces: 0



Number of External Inputs: 0
Number of External Outputs: 1
Number of External Inquiries: 0
Number of Internal Logical Files: 1
Number of External Interfaces: 0

Effort Estimation: Fp- Based Computing

- **Function Point Metric is an example of Product metrics for Analysis Model.**
- **It is used as a means for measuring the functionality delivered by a system and also examines requirement/ analysis model for predicting size of resultant system.**
- **Using historical data, Function Point metric can be used to:-**

i. Estimate the effort or cost required to design, code or test the software.

ii. To predict number of errors that will be encountered during testing.

iii. Forecast number of components or number of projected source links in implemented system. Function points are derived using empirical relationship based on countable (direct) measures of software's information domain and quantitative assessment of software complexity.

- **Software Information Domain Values consists of number of:-**

i. External Inputs (EI)

ii. External Outputs (EO)

iii. External Inquiries (EI)

iv. Internal Logical Files (ILF)

v. External Interface Files (EIF)

To compute function points (FP), the following relationship is used:

$$FP = \text{Count total} \times [0.65 + 0.01 \times \sum (Fi)],$$

where Count total= Sum of all Function Point entries

Calculation of Value Adjustment Factors (VAF) is based on the responses of the following questions:

1	Does the system require reliable backup and recovery?	3
2	Are specialized data communications required to transfer the information to and from the application?	3
3	Are there distributed processing functions?	3
4	Is performance critical?	3
5	Will the system work in an existing heavily utilized operational environment?	3
6	Does the system require online data entry?	3
7	Does the online data entry require input transaction to be built over multiple screens or operations?	3
8	Are Internal Logical Files updated online?	3
9	Are input-output queries or files complex?	3
10	Is the internal processing complex?	3
11	Is the code designed to be reusable?	3
12	Are conversion and installation included in design?	3
13	Is the system designed for multiple installations in multiple organizations?	3
14	Is the application designed to facilitate changes	3

	and ease of use by the user?	
	COUNT TOTAL (Σ Fi)	42

Table : Value Adjustment Factors (VAF)

The count total is the sum of all FP entries obtained from the following table:

Information Domain Value	Count	Weighting factor (Average)	Count* (Average)
External Inputs (EIs)	9	4	36
External Outputs (EOs)	11	5	55
External Inquiries (EQs)	1	4	4
Internal Logical Files (ILFs)	8	10	80
External Interface Files (EIFs)	0	7	0

Total count = 175

Function Point = Total count * [0.65 + (0.01 *∑fi)]
= 175 * [0.65 + (0.01 * 3 * 14)]
= 187.25 (W)

Let
Average Productivity = X = 4 FP/pm
Labour rate = Y =Rs 8000 person per month
So,
Cost per FP = Z = Y/X = Rs 8000/4 = Rs 2000(approx.)
Total estimated project cost = W*Z = Rs 374500 (approx.)

COST ESTIMATION: COCOMO II MODEL

The Constructive Cost Model (COCOMO II) is a more comprehensive estimation model. It is the model that allows one to estimate the cost, effort and schedule when planning a new software development activity.
COCOMO II is actually a hierarchy of estimation models that address different "stages" of the software process.

1. **Stage-I:**
   It supports estimation of prototyping. For this it uses **Application Composition Estimation Model**. This model is used for the prototyping stage of application generator and system integration.
2. **Stage-II:**
   It supports estimation in the early design stage of the project, when we less know about it. For this it uses **Early Design Estimation Model**. This model is used in early design stage of application generators, infrastructure, and system integration.
3. **Stage-III:**
   It supports estimation in the post architecture stage of a project. For this it uses **Post Architecture Estimation Model**. This model is used after the completion of the detailed architecture of application generator, infrastructure, and system integration.

The COCOMO II models require sizing information. Three different sizing options are available as part of the model hierarchy:

1. **Object points**
2. **Function points**
3. **Lines of source code.**

The object point is an indirect software measure that is

computed using counts of the number of

(i) **screens (at the user interface)**

(ii) **Reports**

(iii) **Components likely to be required to build the application**

Each object instance (e.g., a screen or report) is classified into one of three complexity levels (i.e., simple, medium, or difficult). Once complexity is determined, the number of screens, reports, and components are weighted according to the table.

OBJECT TYPE	COMPLEXITY WEIGHT		
	SIMPLE	MEDIUM	DIFFICULT
SCREEN	1	2	3
REPORT	2	5	8
3GL COMPONENT			10

Figure : Complexity Weighing For Object Points

The object point count is then determined by multiplying the original number of object instances by the weighting factor in the figure and summing to obtain a total object point count.

When component-based development or general software reuse is to be applied, the percent of reuse (%reuse) is estimated and the object point count is adjusted:

NOP = (object points) x [(100 - %reuse)/100] where

NOP is defined as new object points.

To derive an estimate of effort based on the computed NOP value, a "productivity rate" must be derived: -

PROD = NOP/person-month
Productivity rate for different levels of developer experience and development environment maturity: -

Developer's experience	Very low	Low	Nominal	High	Very high
Environment maturity	Very low	Low	Nominal	High	Very high
PROD	4	7	13	25	50

Figure : Productivity Rate For Object Points

Once the productivity rate has been determined, an estimate of project effort is computed using Estimated effort = *NOP*/PROD

COCOMO ESTIMATION FOR OUR PROJECT

Number of screens = 11
Number of reports = 1
Number of 3GL components used = 0
In our project there are simple screens and reports.
So, object points = 11 x 1 + 1 x 2
= 13
Since we are not reusing any of the components in our project, the % reuse is zero here.
NOP = 13 x [(100-0)/100] = 13
PROD = 4

Estimated effort = NOP/PROD =13/4 =3.25 person-month (3 person-month)

RISK ANALYSIS

Risk analysis is the process of identifying the risks or uncertainties in the applications or software.

Steps involved in Risk analysis are:-
Risk identification i.e. identify anything that may cause harm. Decide who may be harmed and how.
Assess the risks and make record of the findings
Development of plan to manage these risks.

Risk Identification

Risk identification is a systematic attempt to specify threats to the project plan (estimates, schedule, resource loading, etc.).

There are two distinct types of risks :-
Generic risks are a potential threat to every software project.
Product-specific risks are the risks that are specific for that project.

The checklist can be used for risk identification in the following generic subcategories:
- **Product size (PS)**—Risks associated with the overall size of the software to be built or modified.
- **Business impact (BU)**—Risks associated with constraints imposed by management or the marketplace.
- **Stakeholder characteristics (CU)**—Risks associated with the sophistication of the stakeholders and the developer's ability to communicate with stakeholders in a timely manner.
- **Process definition (PD)**—Risks associated with the degree to which the software process has been defined and is followed by the development organization.

• **Development environment (DE)**—Risks associated with the availability and quality of the tools to be used to build the product.
• **Technology to be built (TE)**—Risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system.
• **Staff size and experience (ST)**—Risks associated with the overall technical and project experience of the software engineers who will do the work.

Assessing Overall Project Risk

1. **Have top software and customer managers formally committed to support the project?** Yes
2. **Are end users enthusiastically committed to the project and the system/ product to be built?** No
3. **Are requirements fully understood by the software engineering team and its customers?** Yes
4. **Have customers been involved fully in the definition of requirements?** Yes
5. **Do end users have realistic expectations?** Yes
6. **Is the project scope stable?** Yes
7. **Does the software engineering team have the right mix of skills?** No
8. **Are project requirements stable?** No
9. **Does the project team have experience with the technology to be implemented?** No
10. **Is the number of people on the project team adequate to do the job?** Yes
11. **Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?** Yes

RISK TABLE

RISKS	CATEGORY	PROBABILITY	IMPACT
Size estimate may be significantly low	PS	30%	2
Large no. of user than planned	PS	10%	2
Less reuse than planned	PS	30%	3
End user may resist the system	BU	30%	2
Delivery decline will be tightened	BU	50%	2
Funding will be lost	CU	20%	1
Customer will change requirements	PS	60%	2
Technology will not meet expectation	TE	5%	1
Lack of training on tools	DE	70%	3
Staff inexperience	ST	30%	3
Staff turnover will be high	ST	5%	2

Impact Values:

1- Catastrophic
2- Critical
3- Marginal
4- Negligible

RISK CONTROL

RISKS	RMMM Plan
Size estimate may be significantly low	Collect more historical data to get accurate estimates.
Large no. of user than planned	Set a deadline for proposing expected no. of users .
Less reuse than planned	Collect reviews to improve the

	user's requirements & hire a experts for that.
End user may resist the system	**Identify the reason for resistance , talk to end users and try to understand the issue and also collect feedback to ensure their needs.**
Delivery decline will be tightened	**Continuously trace the timeline chart & hire experts to meet the deadline.**
Funding will be lost	**Communicate with stakeholders and reassess the budget. Explore alternative funding sources such as loans , partnerships or grants. Consider cost saving measures and prioritize tasks.**
Customer will change requirements	**Set a deadline for proposing changes after which changes proposed would be chargeable and conduct reviews to understand the requests.**
Technology will not meet expectation	**Develop a plan for implementing the new technology such as timeline , budget & a strategy for migrating data from the old technology to new one.**
Lack of training on tools	**Allocate the available tools to the team based on their skillset and prefer tools that the staff is experienced with.**
Staff inexperience	**Appoint experienced staff & train the existing staff.**
Staff turnover will be high	**Communicate with them . Conduct knowledge transfer session. This may involve documentation , training sessions or shadowing opportunities .**

DESIGN

DATA DESIGN

User Login Table -

Field name	Type	Specifications	Constraint	Unique	Description	Example
User name	Alphanumeric	6 alphanumeric characters	Not Null, Primary Key	Yes	User Id of User	Library123
Password	Alphanumeric	12 alphanumeric characters	Not Null	Yes	User's password must be of 8 characters	xyz@1234

Book Available Table -

Field name	Type	Specifications	Constraint	Unique	Description	Example
Bid	Integer	3 digits	Not Null, Primary Key	Yes	Book id of book	373
Book name	String	100 alphanumeric characters	Not Null	No	Name of book	Software engineering
Book writter	String	100 alphanumeric characters	Not Null	No	Writter of book	Pressman

Book Issued Table -

Field name	Type	Specifications	Constraint	Unique	Description	Example
Bid	Integer	3 digits	Not Null, Primary Key	Yes	Book id of book	373
Student roll no.	Integer	**/****	Not Null	No	One student can issue one or more books so not unique	21/5078
Issued date	Date	dd-mm-yyyy	Not Null	No	It provides when student issued book	15-02-2023

Return date	Date	dd-mm-yyyy	Not Null	No	It provides when student have to return book	22-02-2023

Book Not Returned Table -

Field name	Type	Specifications	Constraint	Unique	Description	Example
Bid	Integer	3 digits	Not Null, Primary Key	Yes	Book id of book	373
Student Roll no.	Integer	**/****	Not Null	No	One student can issue one or more books so not unique	21/5076
Fine	Integer	2 digits	Not Null	No	Fine levied on student for not returning on time	36

COMPONENT LEVEL DESIGN

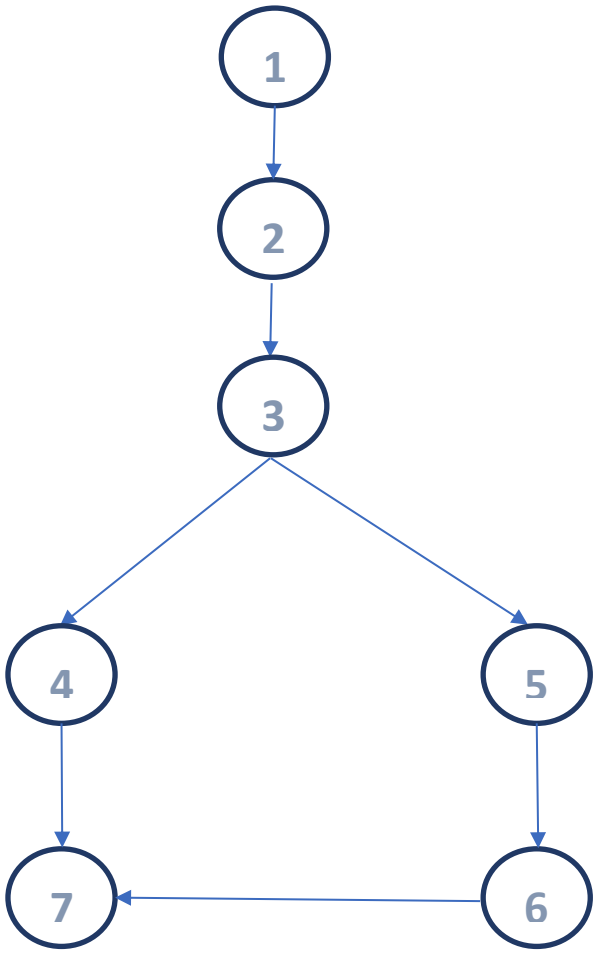
https://drive.google.com/drive/folders/17gs10x9ntpD50HHzSU_aWia2f9xWJL8X?usp=share_link

TESTING

Performing the White Box testing for Login & Book available add or remove modules:-

LOGIN MODULE -

1. Start
2. Taking input (username & password)
3. If username & password valid and matched
4. Successful login
5. Else
6. prompt user to enter details again
7. End



CONTROL FLOW DIAGRAM

Cyclomatic complexity of resultant graph

Method 1

- cc = Total number of regions = 2

Method 2

- cc = Number of edges – Number of nodes + 2*components

=> e – n + 2p

=> 7 – 7 + 2(1) = 2

Method 3

- cc = Number of predicate nodes + 1

=> 1 + 1 = 2

Independent basic patterns from the graph

PATH 1: 1->2->3->4->7

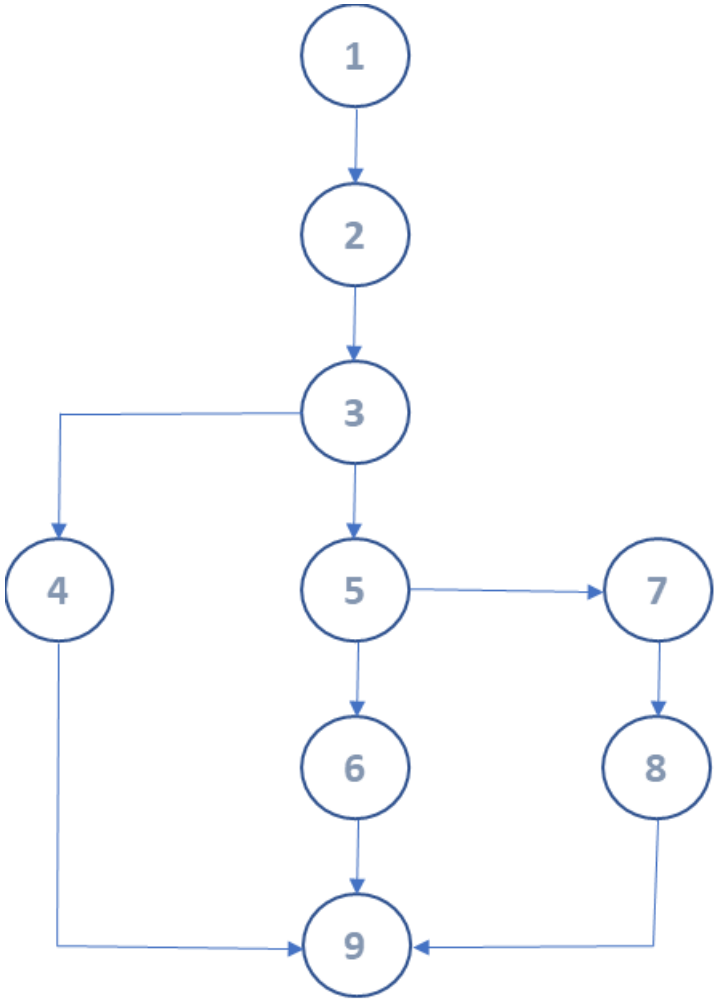
PATH 2: 1->2->3->5->6->7

Generate test cases from basic path set

1. User entered correct username & password and successfully logged in.
2. User entered wrong details & need to login again.

BOOK AVAILABLE MODULE

1. Start
2. Taking input (admin clicking on add or remove)
3. If input is add
4. enter details (book id, email & writer) and add
5. Else if input is remove
6. click on remove and save
7. Else
8. view book available
9. End



CONTROL FLOW DIAGRAM

Cyclomatic complexity of resultant graph

Method 1

- cc = Total number of regions = 3

Method 2

- cc = Number of edges – Number of nodes + 2*components

=> e – n + 2p

=> 10 – 9 + 2(1) = 3

Method 3

- cc => Number of predicate nodes + 1

=> 2 + 1 = 3

Independent basic patterns from the graph

PATH 1: 1->2->3->4->9
PATH 2: 1->2->3->5->6->9
PATH 3: 1->2->3->5->7->8->9

Generate test cases from basic path set

1. If user click on add, user can add information of book available.
2. If user click on remove, user can remove information of book not available now.
3. Else user can view the books available.

REFERENCES

1. Pressman, R. S., & Maxim, B. R., “Software Engineering: A Practitioner’s Approach”, 8th edition, (2015). McGraw-Hill.
2. Aggarwal, K. K., & Singh, Y., “Software Engineering”, 3rd edition, (2007), New Age International Publishers.
3. https://www.geeksforgeeks.org/software-engineering/
4. https://www.ieee.org/
5. https://www.javatpoint.com/software-engineering
6. https://en.wikipedia.org/wiki/Software_engineering