LIBRARY MANAGEMENT SYSTEM

SOFTWARE ENGINEERING

PROJECT REPORT

B.Sc (Hons.) Computer Science

ABSTRACT

Library management system is a project, which aims in developing a computerized system to maintain all the daily work of library. This project has many features that are generally not available in normal library management systems like facility of user login and a facility to teacher's login. It also has a facility of admin login through which the admin can monitor the whole system. It also includes feature where student after logging in their accounts can see list of books issued and its issue date and return date. The librarian after logging into his account i.e. admin account can generate various reports such as student report, issue report, teacher report and book report .Overall this project is developed to help the students as well as staff of library to maintain the library in the best way possible and also reduce the human efforts. Hence, while creating the technology more emphasis was on the use of innovative and nontraditional methods. This project also focuses on developing a simple user interface and providing users with the best possible comfort in Library Management.

INDEX

CHAPTER1

SRS (Software Requirement Specification)

CHAPTER2

Estimation and Scheduling

CHAPTER3

Architectural Design

CHAPTER4

Risk Management

CHAPTER5

Implementation

CHAPTER6

Testing

CHAPTER7

User Manual

CHAPTER8

Conclusion

CHAPTER9

References

Chapter-1

SRS (Software Requirement Specification)

1. Introduction

1.1 Purpose

Library management system is a project, which aims in developing a computerized system to maintain all the daily work of library. This project has many features that are generally not available in normal library management systems like facility of user login and a facility of teacher's login. It also has a facility of admin login through which the admin can monitor the whole system. It has also a facility where students after logging in can see list of books issued and number of books available. The librarian has access to add new books, check for the fine and various functionalities.

Overall, the project of ours is developed to help the students as well as staff of library to maintain the library in the best way possible and reduce the human efforts.

1.2 Document Conventions

The conventions that are used in this project are listed below:

- Headings are marked in Bold and are Underlined.
- The subheads are Bold.
- The sub-sub headers are defined with Bullets and Numbering function.
- The Code for this system is written in Consolas Font.
- The Entire document uses Times New Roman as the default font.

1.3 Intended Audience and Reading Suggestions

The intended audience and the main users for this software include the students/ users, teachers and other faculty members and the library staff. Every audience that tends to use the software is exposed to common as well as unique functionalities. The librarian has the main role in using and administering the software.

- To implement the software, the users must have a little hand on touch with current trend of the technology.
- No specialization of any subject/topic is required to use the software.
- The approval and allowance to users to use the software will be the sole responsibility of librarian.
- The students must be given the user manual before handing over the right to use.

1.4 Product Scope

The document is intended to serve several groups of audiences:

· First, it is anticipated that the SRS will be used by the application designers. Designers will use the information recorded here as the basis for creating the application's design.

· Second, the client for the project, the library manager in our case, is expected to review this document. The SRS will serve to establish a basis for agreement between the client and development team about the functionality to be provided by the application.

· Third, the application maintainers will review the document to clarity their understanding of what the application does.

2. Overall Description

2.1 Product Perspective

The project has been taken into consideration due to the technological development in every field of application. This project is a new innovation in management of the library systems. This project has been developed from scratch and has every required feature necessary in a library. The project brings the real world entity, library closer to the technology. The inspiration came from an online system that tends to reserve the flight tickets. This project has been built on the same fundamental idea with a difference that the book can be reserved as well as issued. The project tends to target the intended audiences namely students, teachers and library staff members.

2.2 Product Functions

1. NORMAL USER

1.1 USER LOGIN

Description of feature

This feature used by the user to login into system. They are required to enter user id and password before they are allowed to enter the system. The user id and password will be verified and if invalid id is there user is allowed to not enter the system.

Functional requirements

-user id is provided when they register

-The system must only allow user with valid id and password to enter the system

-The system performs authorization process which decides what user level can access to.

-The user must be able to logout after they finished using system.

1.2 REGISTER NEW USER

Description of feature

This feature can be performed by all users to register new user to create account.

Functional requirements

-System must be able to verify information

-System must be able to delete information if information is wrong

1.3 REGISTER NEW BOOK

Description of feature

This feature allows adding new books to the library

Functional requirements

-System must be able to verify information

-System must be able to enter number of copies into table.

- System must be able to not allow two books having same book id.

1.4 SEARCH BOOK

Description of feature

We can search book based on book id, book name or by author name.

Functional requirements

- System must be able to search the database based on select search type

- System must be able to filter book based on keyword entered
- System must be able to show the filtered book in table view

1.5 ISSUE BOOKS AND RETURN BOOKS

Description of feature

This feature allows issuing and returning books and also viewing reports of book issued.

Functional requirements

-System must be able to enter issue information in database.

-System must be able to update number of books.

- System must be able to search if book is available or not before issuing books

-System should be able to enter issue and return date information

2.3 User Classes and Characteristics

Library Staff User Class

- The members of this class will have the access to modify/ add/ remove and perform all the important functions to manage the LMS. They are assigned the administrators and no other entity apart from Librarian will be given access to work and use the entire system software.

Teacher User Class

- The members of this class will be allowed to use the features designated for a general user using the LMS. The teacher's class will have a special privilege to reserve the book if it is not available currently. This class neither will have any access to modify the database nor will be given any administrative powers.

General User/ Student Class

- The member of the class will be given very general role that includes all the necessary library features including the searching, issuing but has no privilege to reserve the book. The class has the most basic yet useful functionalities as per the class.

Developer Class

- This class member has the access to entire system. This class comes into action when there is any technical difficulty with the system and the user need the expertise to resolve the issue. Being developer of the system they are allowed to manipulate and modify the entire system as per the need.

2.4 Operating Environment

Following if the requirement to run the software:

Processor	Intel Core Processor or better performance
Operating System	Windows Vista ,Windows7 or 10, Ubuntu
Memory	1GB RAM or more
Hard Disk Space	Minimum 3 GB for database usage for future
Database	My SQL

2.5 Design and Implementation Constraints

This system is Web based. There will be a need to provide PC Server hardware connected to the internet.

LMS can potentially have more than hundreds of users. It is unrealistic to provide training for everyone. Therefore, the system should be designed for easy to use, providing help instructions, and appropriate error messages for invalid user inputs.

Security is important to library operation. Library user is allowed to use the Library Management System only for searching book records. User should never be able to break into the system and to perform any modification.

Reliability is vital to library operation. The LMS should not have any unscheduled down time during library operation hours. Any down time in operation hours has significant impact to the operation and cause inconvenience to everyone in library.

2.6 User Documentation

A video will be provided on how to use and initialize the product according to the need and requirement of the Library.
A tutorial will be provided on how to perform all the operation and management of the databases.
A user manual and training session for the administrator will be organized after the successful delivery of the system.

2.7 Assumptions and Dependencies

The following is a list of assumptions and dependencies that would affect the software requirements if they turned out to be false:

Users have basic understanding to PC and Windows and internet.

There is a method to convert all book records and library user records from the existing system into the Library Management System.

3. External Interface Requirements

3.1 User Interfaces

- The software provides good graphical interface for the user and the administrator can operate on the system, performing the required task such as create, update, viewing the details of the book.
- It allows user to view quick reports like Book Issued/Returned in between particular time.
- It provides stock verification and search facility based on different criteria.
- The user interface must be customizable by the administrator
- All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined
- The design should be simple and all the different interfaces should follow a standard

- The user interface should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module.

  Login Interface:-
  In case the user is not yet registered, he can enter the details and register to create his account. Once his account is created he can 'Login' which asks the user to type his username and password. If the user entered either his username or password incorrectly then an error message appears.

  Search:-
  The member or librarian can enter the type of book he is looking for and the title he is interested in and then he can search for the required book by entering the book name. Categories
  View:-
  Categories view shows the categories of books available and provides ability to the librarian to add/edit or delete category from the list.

  Librarian's Control Panel:-
  This control panel will allow librarian to add/remove users; add, edit, or remove a resource and manage lending options.

3.2 Hardware Interfaces

The existing Local Area Network (LAN) will be used for collecting data from the users and also, for updating the Library Catalogue.

3.3 Interfaces

Database:
- SQL Server.

Application:
- ASP (Active Server Pages)

Web Server:
- IIS (Internet Information Services (IIS) is a powerful Web server that provides a highly reliable, manageable, and scalable Web application infrastructure

3.4 Communications Interfaces

The Customer must connect to the Internet to access the Website:

- Dialup Modem of 52 kbps
- Broadband Internet
- Dialup or Broadband Connection with a Internet Provider.

4. System Features

4.1 LMS Features and Security

The users of the system should be provided the surety that their account is secure. This is possible by providing:-

- User authentication and validation of members using their unique member ID
- Proper monitoring by the administrator which includes updating account status, showing a popup if the member attempts to issue number of books that exceed the limit provided by the library policy, assigning fine to members who skip the date of return
- Proper accountability which includes not allowing a member to see other member's account. Only administrator will see and manage all member accounts

4.1.1Description and Priority

Proposed Database is intended to store, retrieve, update, and manipulate information related to university/school which includes:

- Books availability

-Staff information

-Student details

- My Account

-Calculation of fines

4.1.2 Stimulus/Response Sequences

Responses for Administrator:

- The administrator can Login and Logout.
- When the Administrator Logs into the Library system, the system will check for validity of login.
- If the Login and password are valid, the response to this action is the administrator will be
- Software Requirements Specification for library management system able to modify, view, add, deleting and all other functions that can be performed on the database.

4.1.3 Functional Requirements

1 User Interface

The user interface requirements are concerned with the user interface and how information is presented to the user.

· Usability

Interfaces are a critical class of components within the DML that will provide the means by which users interact with the system. As such, all interfaces should provide easy access to help as well as clearly indicate the current state of the user's transaction when the user isn't idle. Transaction and error status MUST be displayed within each interface component. Cut and paste of text within interfaces and into and out of the interfaces MUST be supported.

· Administrative

Administrative interfaces will assist Library Staff in building/maintaining collections and controlling access to them. Because of the complexity of the data model, Library Staff will need to be able to edit multiple records simultaneously and create links between them. Administrative MUST be able to have multiple records open for editing Administrator MUST be able to create links (references) between records without needing to type in record identifiers. Additionally data represented

in the administrative interface may be in a different state than that stored in the repository. For example, after a record has been edited, but before it has been "saved" into the repository two versions of the record exist. The interface should clearly indicate the state of the locally edited record relative to the version stored in the repository. All editors MUST clearly indicate the state of the edited record (new, saved, and modified/not yet saved).

2 Library user account manage system

SRS-001: The system shall display the user account information including user ID, last and first name, and user position, privilege.

SRS-002: The system shall use a graphic user interface which allows librarians to choice actions including removing, changing and adding user account and account information.

2.1 Logging

Within the system, logging will be used to provide a trail of transactions that have taken place. This might either be for developer debugging purposes, administrative checks on usage, or research on the usability of interfaces.

SRS-003: Transaction logs MUST be kept for each service provided.

SRS-004: Sufficiently detailed client session logs MUST be generated to support analysis of user activities. Security and Privacy

SRS-005: The user's password MUST never be exposed to compromise.

SRS-006: User session logs stored for usability and other research MUST be anonymous.

3 Search book record

The system shall display a list of books which are matching the search criteria sorted by book titles including:

· Category

· Title

· Author

When required by users, the system shall display the information about a particular-book including:

· Category

· Title

· Publisher

-The brief description of the book (if any stored in database) the location

4 Data Entry

The data entry requirements are concerned with how data is entered and validated.

SRS-007: The system shall allow a user to enter his/her data via a keyboard

SRS-008: The system shall allow a user to enter his/her data via choose an item via a mouse.

SRS-009: Whenever the "date" data is needed, it shall be entered only by choose date from a online calendar.

SRS-010: The system shall allow the user to enter the library card number by typing

SRS-011: The system shall allow the user to enter book issuing, recalling data as frequently as required.

5 Search book record

SRS-015: The system shall allow the user typing in search criteria including book title, key word in title, subject, category.

SRS-016: The system shall allow the user choose language option which the searched book is used including English, Arabic

SRS-017: If the search results are a list of books, the system shall allow the user to choose any one of them to see the details.

6 Update book database

SRS-018: The system shall allow the user to add or change the record information including:

-Category

-Title

- Publisher

- Brief description of the book

- Location in library

7 Report Generation

The report generation requirements are concerned with the report generation capabilities of the Library system.

SRS-01: The system shall have a report feature that will allow the user to generate a report showing the information of a particular patron.

SRS-02: The system shall have a report feature that will allow the user to generate a report showing the information of book purchase information in a period including the book titles, category, the author, the publisher, the price. It also shall give

statistic data about the total number of books purchased, the money paid by category.

SRS-03: The system shall be generating those reports to the display, a file or a printer which is linked to the system.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

SRS-04: The system shall be recovered within 10 minutes if it is down.

SRS-05: The system shall be recovered without intervention at user terminal if it is down.

SRS-06: The system shall show appropriate messages at terminal when system is down.

SRS-07: The system shall have 99% reliability during library operating hours.

SRS-08: Scheduled down time after library operating hours shall not be more than 1 hour per day.

SRS-09: The system shall generate error messages when the user attempts to enter invalid data.

5.2 Safety Requirements

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

5.3 Security Requirements

SRS-10: The account management system shall only be used by managers or users with defined privileges.

SRS-11: The Patron information report shall be generated by users who have librarian account.

SRS-12: The book purchase report shall only be generated by managers or users with defined privileges.

SRS-7: Database update data shall be committed to the database only after the managers have approved.

5.4 Software Quality Attributes

Our software has many quality attribute that are given below-

Adaptability

This software is adaptable by any organization/ Library.

Availability-

The availability of the software is easy and for everyone.

Correctness-

The results of the function are pure and accurate.

Flexibility-

The operation may be flexible and reports can be presented in many ways.

Maintainability-

After the deployment of the project if any error occurs then it can be easily maintain by the software developer.

Portability-

The software can be deployed at any machine.

Reliability-

The performance of the software is better which will increase the reliability of the software.

Reusability-

The data and record that are saved in the database can be reused if needed.

Robustness-

If there is any error in any window or module then it does not affect the remaining part of the software.

Testability-

The software will be tested at every step including: Alpha Testing, Beta Testing, and Acceptance Testing

Usability-

To perform any operations and to understand the functioning of software is very easy.

Productivity-

This software will produce every desired result with accurately.

Timelines-

The time limit is very important. It will save much time and provide fast accessing.

Cost effective-

This software is less in cost and bearable by any organization.

5.5 Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data. This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

6. Other Requirements

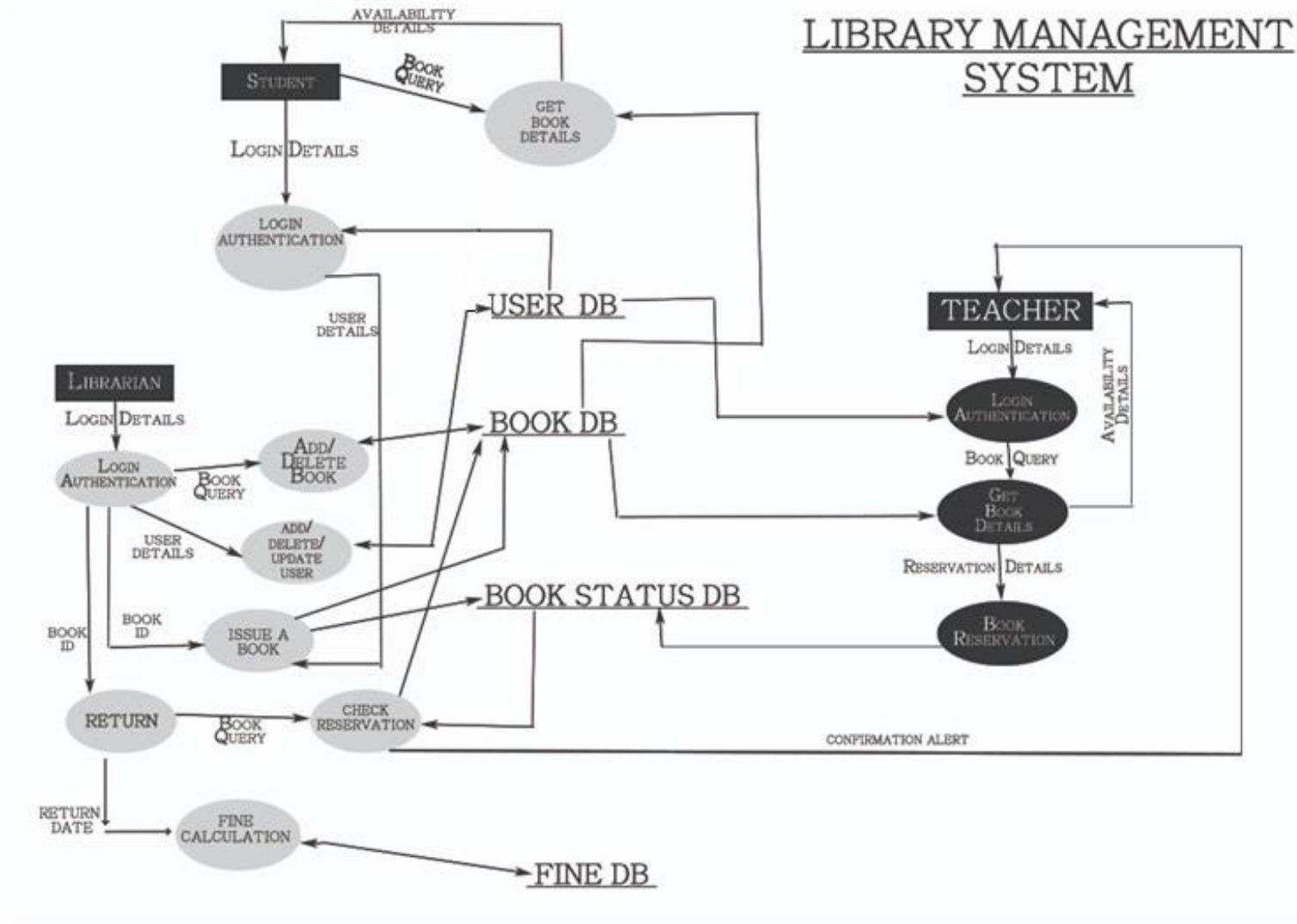
There are different categories of users namely teaching staff, Librarian, Admin, students etc. Depending upon the category of user the access rights are decided. It means if the user is an administrator then he can be able to modify the data, delete, append etc. All other users except the Librarian only have the rights to retrieve the information about database. Similarly there will be different categories of books available. According to the categories of books their relevant data should be displayed. The categories and the data related to each category should be coded in the particular format.

Appendix A: Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

- Administrator: A login id representing a user with user administration privileges to the software
- User: A general login id assigned to most users
- Client: Intended users for the software
- SQL: Structured Query Language; used to retrieve information from a database
- SQL Server: A server used to store data in an organized format ¬ Layer: Represents a section of the project
- User Interface Layer: The section of the assignment referring to what the user interacts with directly
- Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
- Data Storage Layer: The section of the assignment referring to where all data is recorded
- Use Case: A broad level diagram of the project showing a basic overview
- Interface: Something used to communicate across different mediums
- Unique Key: Used to differentiate entries in a database

Data Flow Diagram



Data Dictionary:

1. Book Query: BookName + BookID + BookAuthor + Department
2. Login Details: LoginID + Password
3. User Details: LoginID + Username + RollNo. + Course + Class

4. BookID: BookID
5. Return Date: Date
6. Confirmation Alert: BookID + BookName + Confirmation (Bool)
7. Reservation Details: BookID + BookName + TeacherID
8. Availability Details: BookLocation + No. of Books

Chapter 2

Estimation and scheduling

System Design:

Book table for book records

Field	Data type	Default	Key	Extra
Code	INT(11)	Not Null	Primary	Auto increment
Bookname	VARCHAR(255)	Null		
Author	VARCHAR(255)	Null		
Publication	VARCHAR(255)	Null		
Subject	VARCHAR(255)	Null		
No of copies	INT(10)	Null		

Student table for student information

Field	Data type	Default	Key	Extra
libid	INT(11)	NOT NULL	Primary key	Auto increment
regno	INT(10)	NULL		
branch	VARCHAR(255)	NULL		
section	VARCHAR(255)	NULL		
semester	VARCHAR(255)	NULL		
section	VARCHAR(2)	NULL		
yearofadm	INT(5)	NULL		

Teachers table for teacher's information

Field	Data Type	Default	Key	Extra
Tid	INT(11)	NOT NULL	Primary key	Auto increment
Name	VARCHAR(255)	NULL		
Designation	VARCHAR(255)	NULL		
Branch	VARCHAR(255)	NULL		
Contactno	INT(13)	NULL		
Lectures	LONG BLOB	NULL		

Table to keep record of issued books

Field	Data Type	Default	Key	Extra
bookid	INT(11)	NOT NULL	Foreign key	References book
stuid	INT(11)	NOT NULL	Foreign key	References Student
issuedate	DATE	NULL		
returndate	DATE	NULL		

Student login table

Field	Data type	Default	Key	Extra
logid	INT(11)	NOT NULL	Foreign key	References Student
Username	VARCHAR(255)	NULL		
Password	VARCHAR(255)	NULL		
numbooks	INT(1)	NULL		

Teachers login table

Field	Data Type	Default	Key	Extra
Loginid	INT(11)	NOT NULL	Foreign key	References teacher
Username	VARCHAR(255)	NULL		
Password	VARCHAR(255)	NULL		

Size estimation

Assumptions are stated below:

- We have categorized external inputs on subfields
  If < 3-Simple

  <4-Medium

  >=4-Complex

- For external output
  If 1 detail -Simple

  2 details -Medium

  >2 details -Complex

- We take all logical fields as well as external enquiry as simple.

External Input

- Login credential=>userid, password (simple)
- Book query=>department, name, author (complex)
- librarian login=>admin account, password, captcha(medium)
- Reservation details=>book name, department, userid (medium)
- Book database queries=>(complex)
- User database queries=>(medium)

External Output

- Reservation details=>Book detail + available date (medium)
- Book availability details=>location+availability status (simple)
- Fine calculation=>fine to be paid (simple)

Logical Internal Files (simple)

- User login
- Book details
- Reservation check
- Manage books
- Manage users
- Reserve book
- Admin login

- Issue book
- Return book

External Interface Files

- No files.

External Enquiries

- Book details
- Books under maintenance
- Books issued
- New Book request
- Books not returned on time

	Simple	**Medium**	**Complex**
External I/P	1	3	2
External O/P	2	1	0
Logical Interface	9	0	0
External Interface	0	0	0
External Enquiry	5	0	0

UFP

=1X3+3X4+2X6+2X4+1X5+9X7+5X3

=3+12+12+8+5+63+15

=118

CAF

=0.65+0.01X14X3

=1.07

FP=UFPXCAF

=118X1.07

=126.26

=127(approx)

Cost Estimation

Assumptions are stated below:

- We assume nominal developer experience
    - Screens-11
    - Reports-4
    - 3GL-0
- Abbreviations
    - NOP=New Object Point
    - PROD=Productivity

Screens (Medium)
- Login Screen (Student, Teacher)
- Login Screen (Librarian)
- Book update
- User update
- Issue Book
- Return Book
- Reservation Check
- Fine Calculations
- Book Details (Student)
- Book Details (Teacher)
- Reserving Book

Reports (Medium)
- Availability Details for the Book
- Fine Calculated
- Details of Issues Books
- Reservation Confirmation

Object Point

=11X2+4X5

=22+20

=42

NOP=Object Point X(1-reuse)

=42X(1-0)

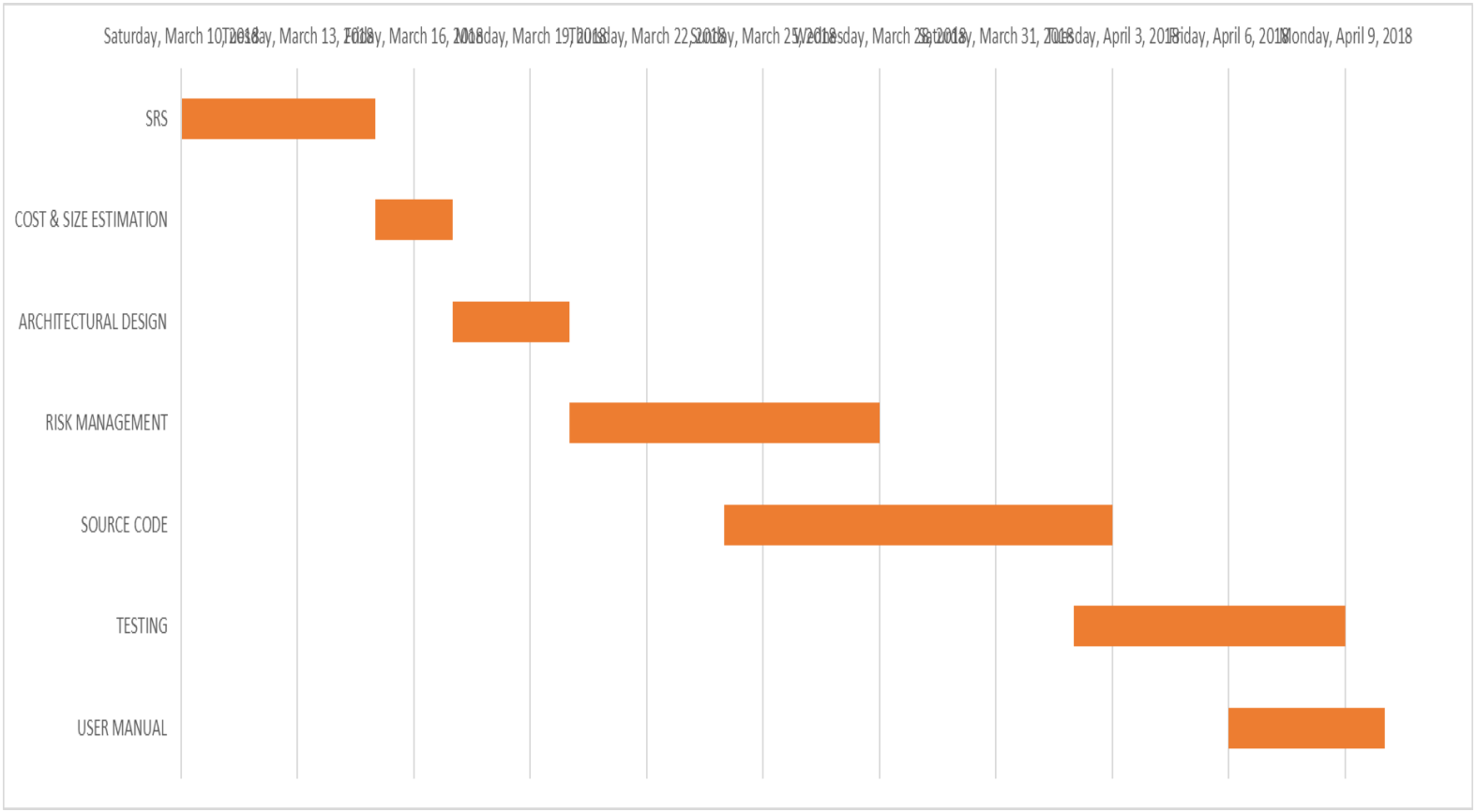
=42

Efforts=NOP/PROD

=42/13=3.23

Gantt chart

A Gantt chart is a graphical depiction of a project schedule. The following is the gantt chart in regard to our project (LMS):

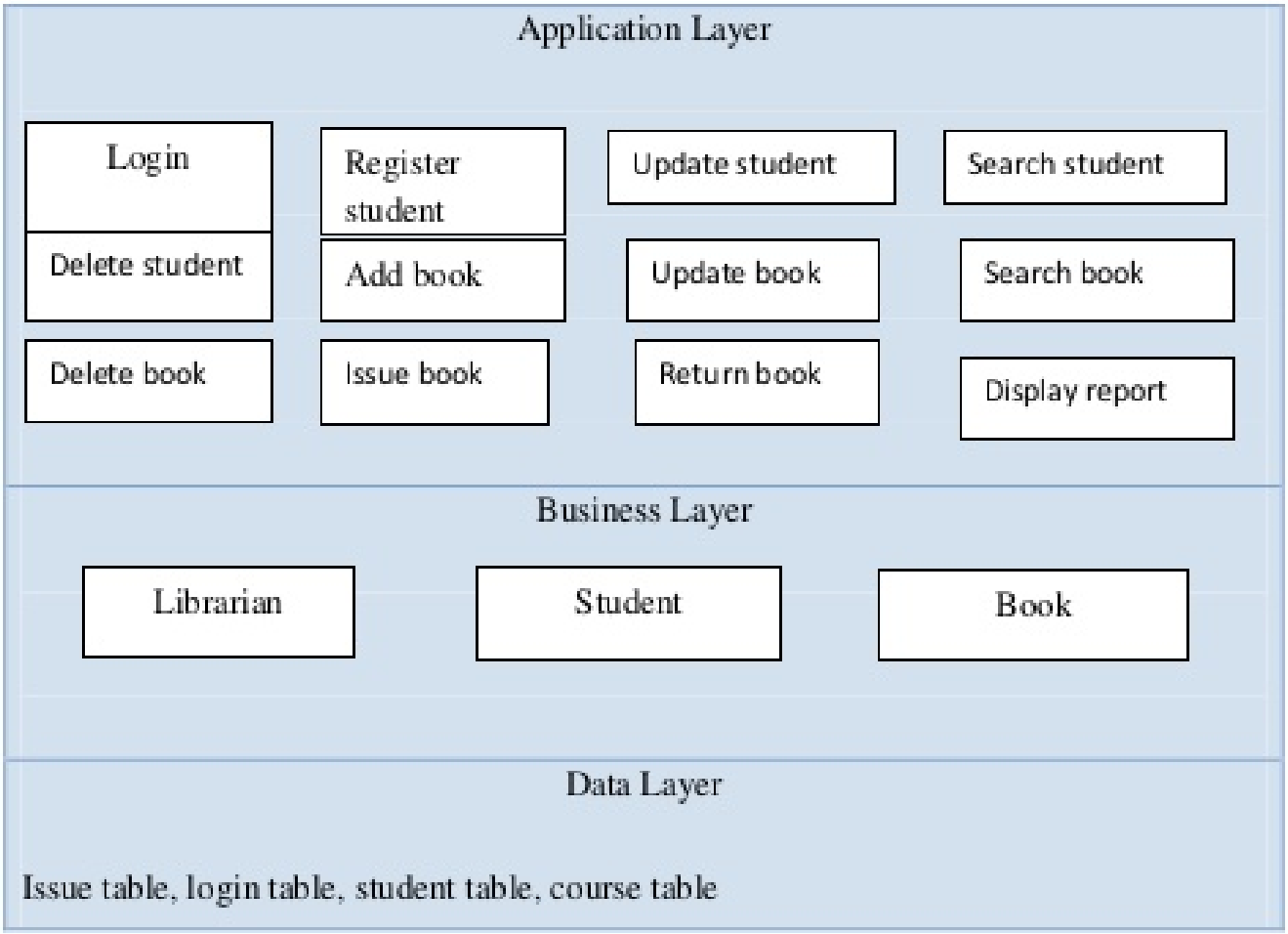


Chapter 3

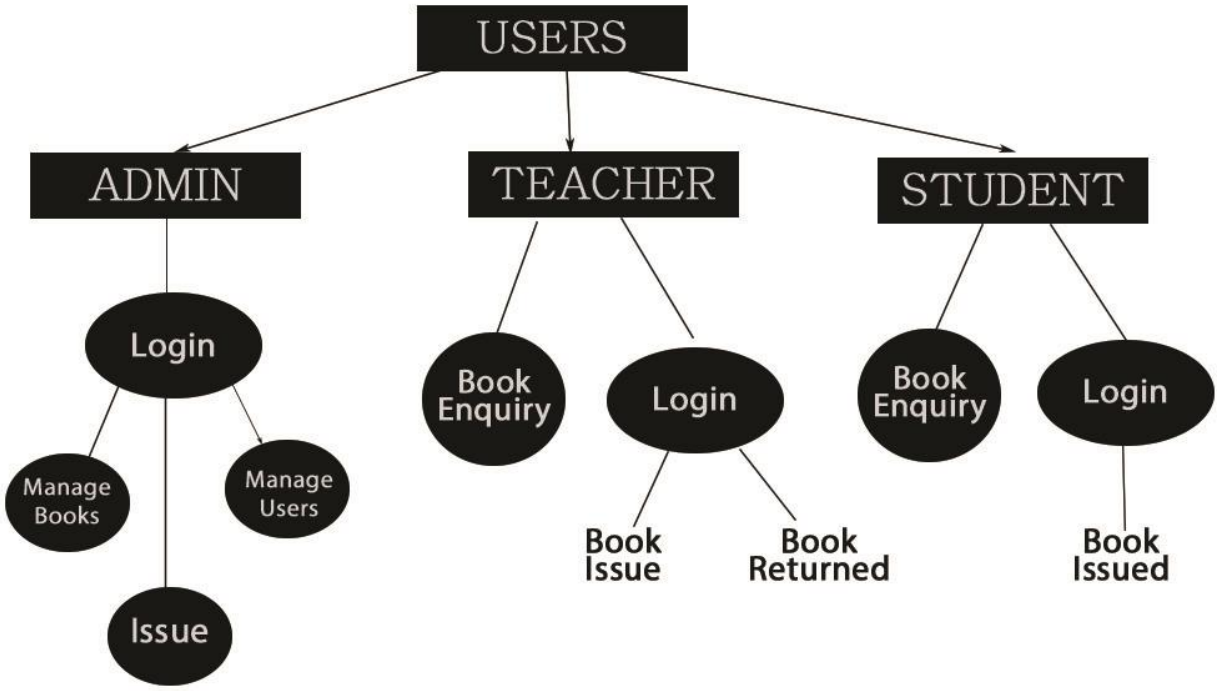
Architectural design

Architectural Designing is the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.

Layered Architecture:



Architectural Design:



Chapter 4

Risk management

Purpose of the Risk Management

A risk is an event or condition that, if it occurs, could have a positive or negative effect on a project's objectives. Risk Management is the process of identifying, assessing, responding to, monitoring, and reporting risks. This Risk Management Plan define show risks associated with the Library Management System project will be identified, analyzed, and managed. The outline show risk management activities will be performed, recorded, and monitored throughout the life cycle of the project and provides templates and practices for recording and prioritizing risks.

The Risk Management Plan is created by the project manager in the Planning Phase and is monitored and updated throughout the project.

Risk Management Procedure

1. PROCESS

The project manager working with the project team will ensure that risks are actively identified, analyzed, and managed throughout the life of the project. Risks will be identified as early as possible in the project so as to minimize their impact. The steps for accomplishing this are outlined in the following sections.

2. RISK IDENTIFICATION

Risk identification will involve the project team members and will include an evaluation of environmental factors, organizational culture and the project management plan including the project scope. Careful attention will be given to the project deliverables, assumptions, constraints, cost/effort estimates, resource plan, and other key project documents.

3. RISK ESTIMATION

Risk projection, also called risk estimation, attempts to rate each risk in two ways:

(1) The likelihood or probability that the risk is real.

(2) The consequences of the problems associated with the risk, should it occur. You work along with other managers and technical staff to perform four risk projection steps:

1. Establish a scale that reflects the perceived likelihood of a risk.
2. Identify the consequences of the risk.
3. Estimate the impact of the risk on the project and the product.
4. Assess the overall accuracy of the risk projection so that there will be no misunderstandings.

Risks	**Probability**	**impact**
Database crash	20%	1
Software Hacked	10%	2
Staff turnover will be high	40%	3
Delivery deadline will be tightened	50%	2

Impact values:

1—catastrophic
2—critical
3—marginal
4—negligible

4. THE RMMM PLAN

A separate risk mitigation, monitoring, management plan for each major risk will be assigned to a project team member for monitoring purposes to ensure that the risk will not "fall through the cracks". The RMMM plan for each major risk would be stated by a RIS (Risk Information Sheet).

Major Risks:

1.Database Crash
2.Software Hacking

RISK INFORMATION SHEET			
RiskID:ROP-01	Date:3/15/1	Prob:20%	Impact :Catastrophic
Description: Database Crash means that our database stops responding to commands from users and is unable to fetch data for the same.			
Refinement/Context: The server of the Database provider is not working properly. The database location is ambiguous. No. of users exceeds the limit.			
Mitigation/Monitoring: To avoid such situations, Backup of data is taken every day. We can make a simple routine to back up our data at a particular time in the morning and in the evening as well.			
Management: As soon as the Database Crashes we can restore data from recent backup and provide our users the same experience again.			
Current Status: Not yet occurred.			
Assigned:			

RISK INFORMATION SHEET			
RiskID:ROP-02	Date:3/20/18	Prob:10%	Impact :Catastrophic
Description: Software Hacked means some hacker get unauthorized access of our data.			
Refinement/Context: The security company that we hired was unable to provide quality services.			
Mitigation/Monitoring: To enhance the security of our product , we can hire some internet security companies to secure our software product from malicious attacks.			
Management: We have to hire ethical hackers who could help us to deal with the problem.			
Current Status: Not yet occurred.			
Assigned;			

Chapter 5

Implementation and Snapshots

Users login

```
<?xml
version="1.0"
encoding="utf-
8"?>
                <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
                    xmlns:app="http://schemas.android.com/apk/res-auto"
                    xmlns:tools="http://schemas.android.com/tools"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:layout_margin="8dp"
                    tools:context="com.example.manav.e_library.LoginActivity">

                    <TextView
                        android:id="@+id/textView1"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="Library Management System"
                        android:layout_centerInParent="true"
                        android:layout_centerHorizontal="true"
                        android:textSize="24sp"/>
                    <LinearLayout
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:layout_alignParentBottom="true"
                        android:layout_below="@+id/textView1">

                        <Button
                            android:id="@+id/buttonStudent"
                            android:layout_width="0dp"
                            android:layout_weight="1"
                            android:layout_height="wrap_content"
                            android:layout_gravity="bottom"
                            android:text="Student"/>

                        <Button
                            android:id="@+id/buttonTeacher"
```

```
                android:layout_width="0dp"
                android:layout_weight="1"
                android:layout_height="wrap_content"
                android:layout_gravity="bottom"
                android:text="Teacher"/>

        </LinearLayout>

    </RelativeLayout>
```

Admin login

```
<?xml
version="1.0"
encoding="utf-
8"?>
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="8dp"
        android:orientation="vertical"
        tools:context="com.example.manav.e_library.AdminLogin">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Admin Login"
            android:textAlignment="center"
            android:textSize="32sp"
            android:textStyle="bold"
            />
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="username"
            android:layout_marginTop="36dp"
            android:inputType="text"/>
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="password"
            android:layout_marginTop="36dp"
```

```
            android:inputType="textPassword"/>
        <Button
            android:id="@+id/buttonAdminLogin"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="36dp"
            android:text="login"/>
    </LinearLayout>
```

List of Books

```
<?xml
version="1.0"
encoding="utf-
8"?>
    <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/drawer_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <ListView
            android:id="@+id/bookListView"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            />
    </RelativeLayout>
```

Student login

```
<?xml
version="1.0"
encoding="utf-
8"?>
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="8dp"
        android:orientation="vertical"
        tools:context="com.example.manav.e_library.StudentLogin">
```

```
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Student Login"
            android:textAlignment="center"
            android:textSize="32sp"
            android:textStyle="bold"
            />
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="username"
            android:layout_marginTop="36dp"
            android:inputType="text"/>
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="password"
            android:layout_marginTop="36dp"
            android:inputType="textPassword"/>
        <Button
            android:id="@+id/buttonStudentLogin"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="36dp"
            android:text="login"/>


    </LinearLayout
```

Teachers login

```
<?xml
version="1.0"
encoding="utf-
8"?>
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="8dp"
        android:orientation="vertical"
        tools:context="com.example.manav.e_library.TeacherLogin">
```

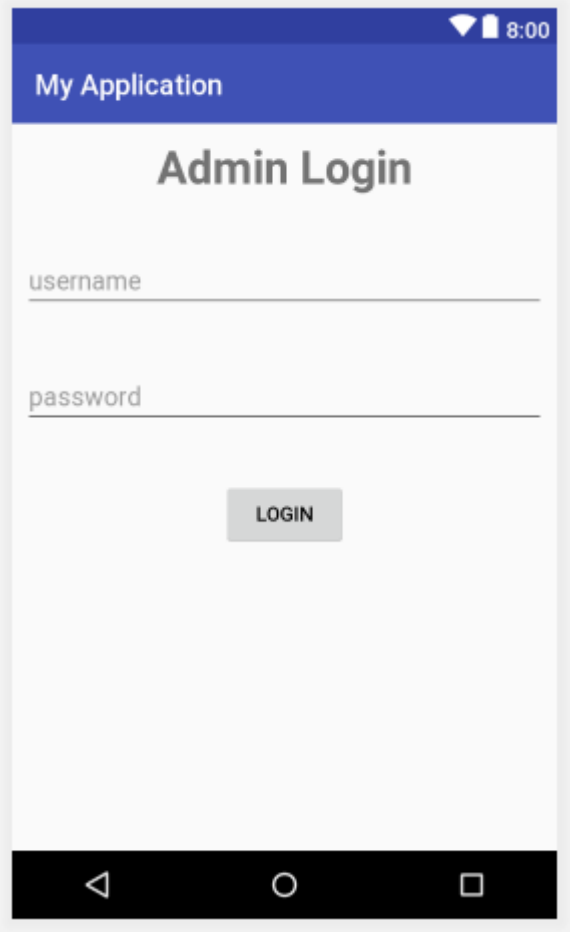
```
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Teacher Login"
            android:textAlignment="center"
            android:textSize="32sp"
            android:textStyle="bold"
            />
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="username"
            android:layout_marginTop="36dp"
            android:inputType="text"/>
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="password"
            android:layout_marginTop="36dp"
            android:inputType="textPassword"/>
        <Button
            android:id="@+id/buttonTeacherLogin"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="36dp"
            android:text="login"/>

    </LinearLayout>
```

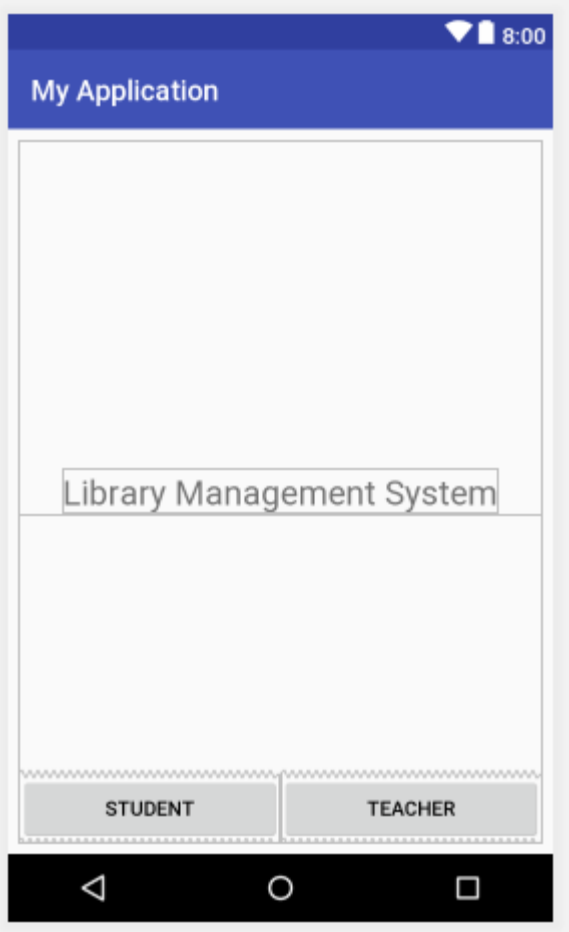
Book List item

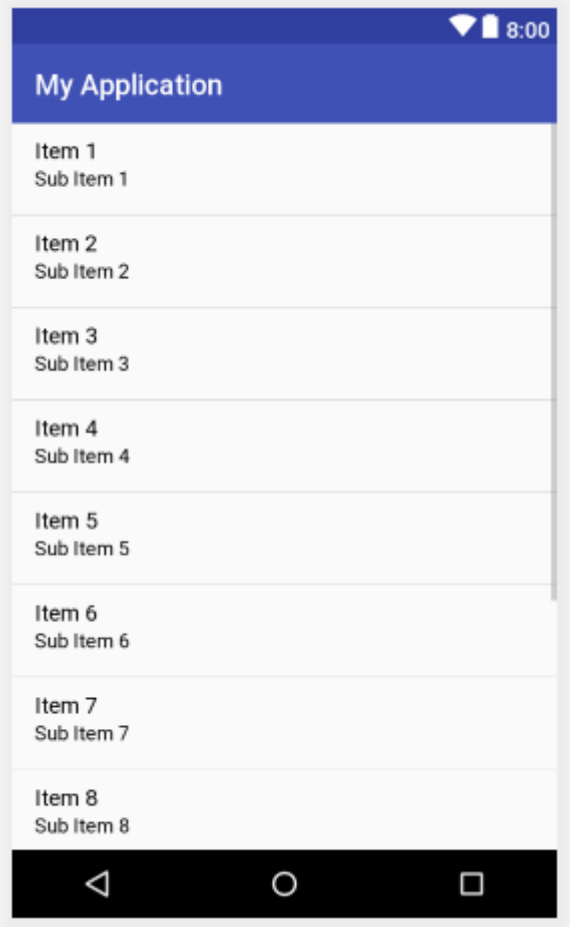
```
<?xml
version="1.0"
encoding="utf-
8"?>
    <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="8dp">

        <ImageView
```

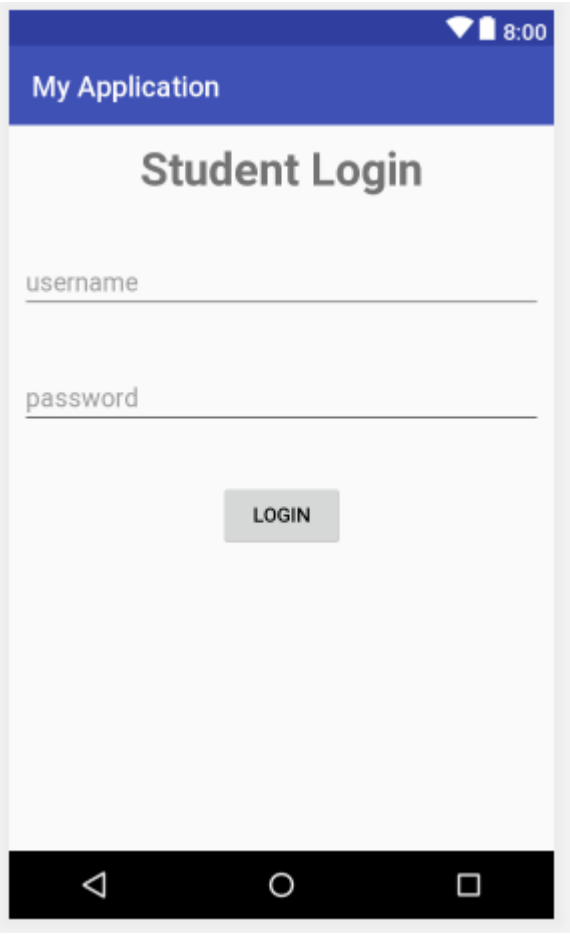
```
        android:id="@+id/bookThumbImage"
        android:layout_width="72dp"
        android:layout_height="72dp"
        android:src="@mipmap/ic_launcher"
        />

    <TextView
        android:id="@+id/nameTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        tools:text="Book Name"
        android:paddingLeft="4dp"
        android:layout_toRightOf="@id/bookThumbImage"
        android:layout_toEndOf="@id/bookThumbImage"
        android:textSize="24sp"/>
    <TextView
        android:id="@+id/authorTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="4dp"
        android:layout_below="@id/nameTextView"
        android:paddingTop="12dp"
        android:textSize="18sp"
        tools:text="Author"
        android:layout_toRightOf="@id/bookThumbImage"
        android:layout_toEndOf="@id/bookThumbImage"/>
    <TextView
        android:id="@+id/priceTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="Price:900"
        android:layout_alignBaseline="@+id/authorTextView"
        android:layout_alignBottom="@+id/authorTextView"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

</RelativeLayout>
```

Snapshots of Module

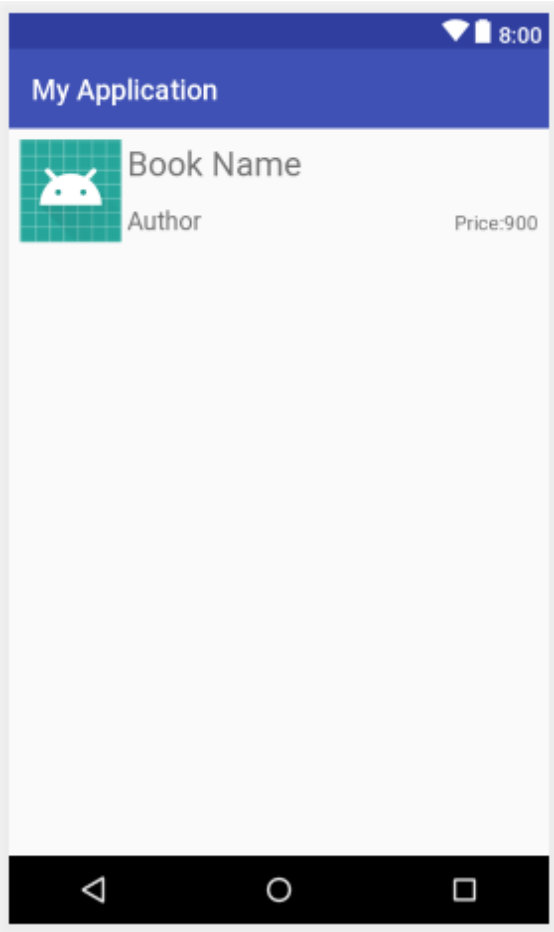

8:00
My Application
Admin Login
username
password
LOGIN


8:00
My Application
Library Management System
STUDENT
TEACHER


8:00
My Application
Item 1
Sub Item 1
Item 2
Sub Item 2
Item 3
Sub Item 3
Item 4
Sub Item 4
Item 5
Sub Item 5
Item 6
Sub Item 6
Item 7
Sub Item 7
Item 8
Sub Item 8


8:00
My Application
Student Login
username
password
LOGIN


8:00
My Application
Teacher Login
username
password
LOGIN


8:00
My Application
Book Name
Author
Price:900

Java files

Adapter

```
package
com.example.manav.e_library
.Adapters;

import android.app.Activity;
import android.content.ContentResolver;
import android.content.Context;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;
import java.util.ArrayList;
import com.example.manav.e_library.R;
import com.example.manav.e_library.Utility.Book;
/**
 * Created by manav on 4/1/18.
 */
public class BookAdapter extends ArrayAdapter<Book> {

    public BookAdapter(Context context, int resource,
ArrayList<Book> objects) {
        super(context,resource,objects);
    }

    @NonNull
    @Override
    public View getView(int position, @Nullable View
convertView, @NonNull ViewGroup parent) {
        if(convertView == null){
            convertView =
((Activity)getContext()).getLayoutInflater().inflate(R.layout.li
st_item,parent,false);
        }
        TextView name = (TextView)
convertView.findViewById(R.id.nameTextView);
        Book book = getItem(position);
        name.setText(""+book.getName());
```

```
                return convertView;
            }
        }
```

Book

```
package
com.example.manav.e_library.Utility;

/**
 * Created by manav on 4/1/18.
 */

public class Book {
    private String name;
    private String author;
    private String category;
    private String Department;
    private double price;

    public Book() {
    }

    public Book(String name, String author, String
category, String department, double price) {
        this.name = name;
        this.author = author;
        this.category = category;
        Department = department;
        this.price = price;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAuthor() {
        return author;
    }
```

```
    public void setAuthor(String author) {
        this.author = author;
    }

    public String getCategory() {
        return category;
    }

    public void setCategory(String category) {
        this.category = category;
    }

    public String getDepartment() {
        return Department;
    }

    public void setDepartment(String department) {
        Department = department;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }
}
```

Login Users Type

```
package
com.example.manav.e_libra
ry;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
```

```
public class LoginActivity extends AppCompatActivity {

    private Button mStudentLogin,mTeacherLogin;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        mStudentLogin = (Button) findViewById(R.id.buttonStudent);
        mTeacherLogin = (Button) findViewById(R.id.buttonTeacher);
        mStudentLogin.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new
Intent(getApplicationContext(),StudentLogin.class);
                startActivity(i);
                finish();
            }
        });
        mTeacherLogin.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new
Intent(getApplicationContext(),TeacherLogin.class);
                startActivity(i);
                finish();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if
it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
```

```
        // automatically handle clicks on the Home/Up button, so
long
        // as you specify a parent activity in
AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        Intent intent;
        switch(id){
            case R.id.action_login:

Toast.makeText(getApplicationContext(),"Admin",Toast.LENGTH_SHORT)
.show();
                return true;
            default:
                return true;
        }

    }


}
```

Student login

```
package
com.example.manav.e_library;

   import android.content.Intent;
   import android.support.v7.app.AppCompatActivity;
   import android.os.Bundle;
   import android.view.View;
   import android.widget.Button;

   public class StudentLogin extends AppCompatActivity {
       private Button mLogin;
       @Override
       protected void onCreate(Bundle savedInstanceState) {
           super.onCreate(savedInstanceState);
           setContentView(R.layout.activity_student_login);
           mLogin = (Button)
   findViewById(R.id.buttonStudentLogin);
           mLogin.setOnClickListener(new View.OnClickListener() {
               @Override
```

```
            public void onClick(View view) {
                Intent i = new
Intent(getApplicationContext(),MainActivity.class);
                startActivity(i);
                finish();
            }
        });
    }
}
```

Teacher login

```
package
com.example.manav.e_library;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class TeacherLogin extends AppCompatActivity {
    private Button mLogin;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_teacher_login);
        mLogin = (Button)
findViewById(R.id.buttonTeacherLogin);
        mLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new
Intent(getApplicationContext(),MainActivity.class);
                startActivity(i);
                finish();
            }
        });
    }
}
```

Chapter 6

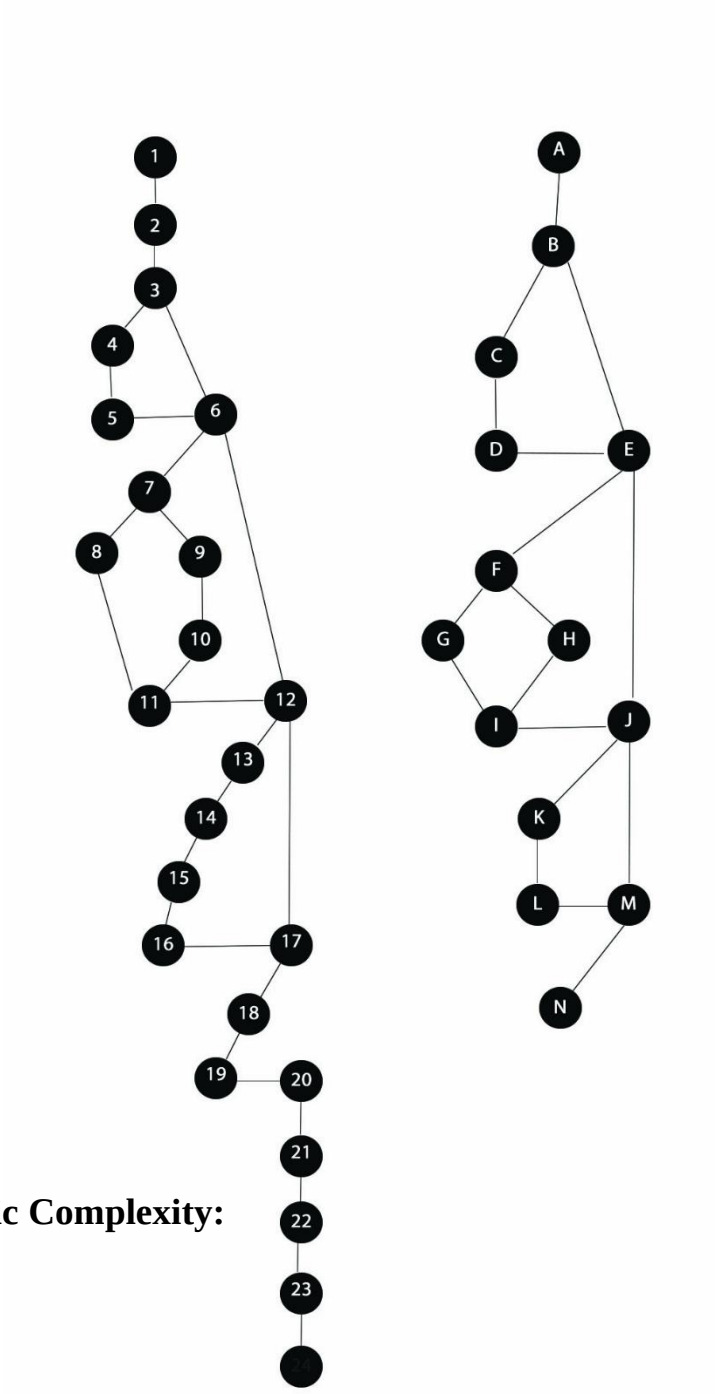
Testing

Fine Calculation Module

```
int calculateFine(int *actualDate, int *returnDate)
   {

            int fine=0;
      // Checking various conditions

      if (actualDate[0] <= returnDate[0] && actualDate[1] == returnDate[1] && actualDate[2] == returnDate[2])
         {
            fine = 0;

         }
      else if(actualDate[0] <=returnDate[0] && actualDate[1] > returnDate[1] && actualDate[2] == returnDate[2])
         {
             if (actualDate[1]-returnDate[1]>1 &&  actualDate[0] ==returnDate[0])
                        fine=500*  (actualDate[1]- returnDate[1] );
            else
            {
                        fine = 15*(30-(actualDate[0]-returnDate[0]));

            }

         }

      else if(actualDate[0] > returnDate[0] && actualDate[1] == returnDate[1] && actualDate[2] == returnDate[2])
         {
            late = actualDate[0] - returnDate[0];
            fine = 15*late;

         }
      else if(actualDate[1] > returnDate[1] && actualDate[2] == returnDate[2])
         {
            late = actualDate[1] - returnDate[1];
            fine = 500*late;

         }
      else
         {
            fine = 10000;

         }



            return fine;
   }
```

Basis Path Testing:

Program Flow Diagram



Cyclomatic Complexity:

=e-n+2

n = 14, e = 17

C.C = 17 – 14 + 2 = 5

Independent Paths:

1. A -> B -> C -> D -> E -> J -> M -> N
2. A -> B -> E -> F -> G -> I -> J -> M -> N
3. A -> B -> E -> F -> H -> I -> J -> M -> N
4. A -> B-> E -> J -> K -> L -> M -> N
5. A -> B -> E -> J -> M -> N

Test Cases:

Path No.	RETURN DATE	ACTUAL DATE	Observed Output (FINE)
1.	12-MAR-2017	12-MAR-2017	0
2.	12-FEB-2017	12-MAR-2017	500
3.	12-FEB-2017	10-MAR-2017	420
4.	16-FEB-2018	17-MAR-2018	500
5.	17-FEB-2017	18-MAR-2018	10000

Functional Testing:

Inputs						Expected Output	Observed Output
Return Date[0]	**Return date[1]**	**Return date[2]**	**Actual date[0]**	**Actual date[1]**	**Actual date[2]**		
1	1	2012	1	1	2012	0	0
1	1	2012	2	1	2012	15	15
22	1	2012	26	2	2012	500	500
1	1	2012	2	3	2013	10000	10000

Input Equivalence Class:

I1= {Actualdate, As per the valid date range in calendar}

I2= {Returndate, As per the valid date range in calendar}

Output Equivalence Class:

O1= {0, Book returned before or on the return date}

O2= {15, per day fine until a month from return date}

O3= {500, per day fine until a year from the return date}

O4= {10000, If book is returned after an year}

Chapter 7

User manual

The user must read this manual before working and handling the software:

Patron – Login

No.	Input & Procedures
1.	To access system as Librarian Administrator: - Login as: [Username] - Password as: [Username]
2.	To access system as Teacher: - Login as: [Teacher] - Password as: {Assigned by the Administrator}
3.	To access system as Student: - Login as: [StudentName] - Password as: {Assigned by the administrator}

Patron – Operations

No.	Input and Procedure
1.	To search an item: - Click Search - Enter the name of book - Enter the edition - Enter the author - Click Search
2.	To issue a book: - Click Search - Check if the book is available - Click issue the book - Collect it from the Administrator/ Librarian
3.	To return a book: - View the returning date

- Visit Administrator/ Librarian
- Return the issued book
- Confirm regarding dues
- Book status updated

4. To pay the fine:
   - Visit the administrator
   - Get the return date
   - Administrator will generate fine
   - Return the book
   - Book status updated
   - Fine/ Dues Cleared Status updated

5. To add/ remove the user:
   - Visit the Administrator
   - Provide your personal details and ID Card
   - Account will be generated
   - UserID and Password will be generated
   - User Status Updated
   - Access to the System to be granted by the Administrator.

6. To reserve the book (Teachers)
   - Search the desired book
   - Look for the availability
   - Click reserve if not available
   - Status Updated for reservation
   - Date generated for issue
   - Visit the administrator for issue and status update.

Chapter 8

Conclusion

Library Management System allows the user to store the book details and the customer details. This software package allows storing the details of all the data related to library. The system is strong enough to withstand regressive yearly operations under conditions where the database is maintained and cleared over a certain time of span. The implementation of the system in the organization will considerably reduce data entry, time and also provide readily calculated reports. Thus we conclude that the project aims in digitalizing the Library and making it more interactive more user friendly and project which fulfills each users need in the best way possible.

Chapter 9

References

This project uses information and technology from various references stated below:

- R.S. Pressman, Software Engineering: A Practitioner's Approach (7th Edition) McGrawHill, 2009.
- P. Jalote, An Integrated Approach to Software Engineering (2nd Edition), Narosa Publishing House, 2003.
- K.K. Aggarwal and Y. Singh, Software Engineering (revised 2nd Edition), New Age International Publishers, 2008.
- I. Sommerville, Software Engineering (8th edition), Addison Wesle, 2006. 5. D. Bell, Software Engineering for Students (4th Edition), Addison-Wesley, 2005. 6. R. Mall, Fundamentals of Software Engineering (2nd Edition), Prentice-Hall of India, 2004.
- A lot of information has been collected from numerous websites including,
    - www.google.co.in
    - www.wikipedia.org
    - www.stackoverflow.com
    - www.tutorialspoint.com