A project report on

E-HOUSE HELPING SOCIETY MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements

For the award of the Degree of

BACHELOR OF ENGINEERING

IN

ARTIFICIAL INTELLAGENCE

AND DATA SCIENCE

Submitted By

S.V.SAI POOJITHA

Under the Esteemed Guidance of

Mr. VENKATA PRASAD

Project Manager, Schemaxtech Pvt.Ltd.

VISAKHAPATNAM

ABSTRACT

This research developed a web based housing management system. The system is to manage the housing for senior staffs and to enhance easy application/update for accommodation. It also helps the housing unit to have easy access to data, increase its productivity, and save cost of production.

Nowadays the society management system is categorized by the number of houses. The people who lives in house they may be an owner of house or tenant of house. In this era, people are very busy with their daily routine work, so they do not have time to complain about small problem related to houses. We have developed the system for society member they can make complain from anywhere, anytime and we can solve the Complain as soon as possible. By this system the address can be easily detected of the members who post their complaints regarding their houses. The individuals who need to offer their homes or give it at lease, by utilizing this framework it is effortlessly conceivable by just login their points of interest. This webpage is basically utilized for owning a house or finding a leased house by simply checking this site. Further-more the user can also book society halls online using this application, check the status of the hall, the pricing and other required details. Generate invoice in a single click, Share snapshots of every society functions via gallery so that every members can access the snapshots.

The basic concept of this society system, we create global web based application in asp.net with c# language to manage Society with House and Owner of house detail. We provide platform to register society to our system, and there are many houses according to each society. Each house allocated to owner of house, all the owner are members of our project. After becoming a member they can login to give his house on rent or sell for all users. The people can easily find search the rent house or sell house in which society and get contact detail of owner of house by visiting our website. All members can make online complain related home to society management.

CONTENTS

1.Introduction

1.1 Motivation:

The motivation behind developing an E-Housing Helping Society management system is to streamline and improve the management and operations of housing societies or residential communities. Traditional manual methods of managing housing societies can be time-consuming, inefficient, and prone to errors. Implementing an electronic system offers various benefits and addresses several challenges faced by such societies , including efficiency , transparency , communication , Financial Management .

1.2 objective:

The main objective of the E-Housing Helping Society management system is to streamline and enhance the overall management and functioning of housing societies or residential communities.

The system aims to achieve the following main objectives:

- Efficient Management:
- Transparency and Accountability
- Improved Communication
- Financial Stability
- Enhanced Security

1.3 Purpose:

The purpose of the E-Housing Helping Society management system is to provide a comprehensive and integrated solution to manage all aspects of housing societies or residential communities.

Overall, the purpose of the E-Housing Helping Society management system is to create a cohesive and well-managed living environment, promote efficient operations, and enhance the overall living experience for residents within the housing society. It aims to bring transparency, convenience, and effectiveness to various aspects of housing society management.

3.SYSTEM ANALYSIS

- **EXISTING SYSTEM:**

Presently to maintain information about different aspects, the Owner is using manual process i.e., using houses and flats. Now the Admin requires a computerized environment where it is easy for storing information about user details, their name, addhar , phone , photo, username , password , and so on.

Using present manual process it is difficult in maintaining data and moreover it is time consuming. There is wastage of stationary and more human resources effort is required. When we have computerized environment it replaces all these problems.

3.2 PROPOSED SYSTEM:

The LIBRARY MANAGEMENT SYSTEM is a software application which avoids more manual hours in taking the book, that need to spend in record keeping and generating reports. Maintaining of user details is complex in manual system in terms of agreements, royalty and activities. This all have to be maintained in ledgers or books. Co-coordinators needs to verify each record for small information also.

- Easy search of book in the online library.
- Avoid the manual work.
- User need not go to the library for Issue any kind of book, he can renewal the book online.

4.SOFTWARE REQUIREMENT SPECIFICATION

4.1 HARDWARE REQUIREMENTS:

System	Intel(R) Core(TM) i5@ 1.60GHz 2.11 GHz
Hard Disk	40 GB.
Floppy Drive	1.44 Mb.
Monitor	15 VGA Colour.
Mouse	Lenovo.
Ram	512 Mb.

4.2 SOFTWARE REQUIREMENTS:

Operating system	Windows 11.
Coding Language	C#
IDE	Visual studio
Database	SQL SURVER

4.SOFTWARE DESCRIPTION

(Software Environment)

C#.NET Technology

C# is a programming language of .Net Framework.

C# is pronounced as "C-Sharp". It is an object-oriented programming language provided by Microsoft that runs on .Net Framework.

By the help of C# programming language, we can develop different types of secured and robust applications:

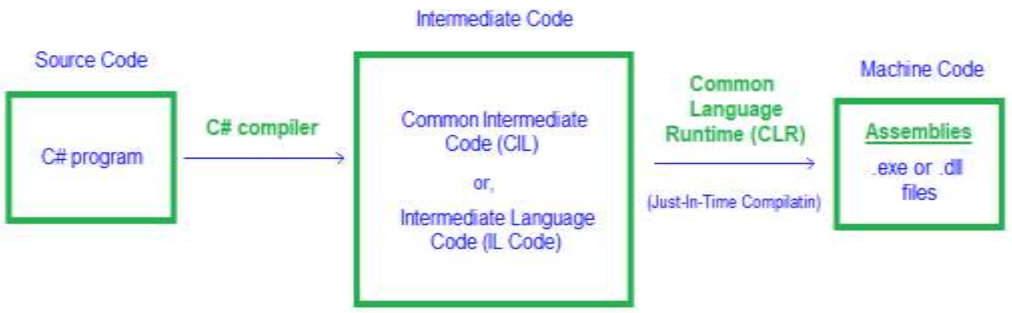
- Window applications
- Web applications
- Distributed applications
- Web service applications
- Database applications etc

Features:

1. Simple
2. Modern programming language
3. Object oriented
4. Type safe
5. Interoperability

6. Scalable and Updateable
7. Component oriented
8. Structured programming language
9. Rich Library
10. Fast speed

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The C# programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language the platform-independent codes interpreted by the interpreter on the c# platform. The interpreter parses and runs each byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.


Source Code
C# program
C# compiler
Intermediate Code
Common Intermediate Code (CIL)
or,
Intermediate Language Code (IL Code)
Common Language Runtime (CLR)
(Just-In-Time Compilatin)
Machine Code
Assemblies
.exe or .dll files

The C# platform:

C# is a programming language developed by Microsoft and is part of the .NET platform, which is a software framework designed for building and running applications on multiple platforms. The .NET platform provides a comprehensive set of libraries and tools that enable developers to create a wide range of applications, including desktop applications, web applications, mobile apps, cloud-based services, and more.

The key components of the .NET platform include:

Common Language Runtime (CLR): The CLR is the heart of the .NET platform. It is responsible for executing and managing .NET applications. When C# code is compiled, it is converted into an intermediate language (IL) code that is executed by the CLR. The CLR provides features like memory management, security, exception handling, and garbage collection.

Base Class Library (BCL): The BCL is a collection of reusable classes and functions that provide a wide range of functionality for common programming tasks. It includes classes for file handling, networking, data manipulation, security, user interface components, and more. The BCL allows C# developers to access a rich set of APIs without having to write low-level code.

Just-In-Time (JIT) Compiler: The JIT compiler is responsible for converting the IL code into machine code that can be executed by the target hardware. The JIT compilation process occurs at runtime, allowing the application to be optimized for the specific platform it is running on.

Common Language Specification (CLS): The CLS is a set of rules and guidelines that ensure that different .NET languages (such as C#, VB.NET, and F#) can interoperate seamlessly. By adhering to the CLS, developers can create libraries and components that can be used by other .NET languages.

What Can C# Technology Do?

C# technology, as a versatile and powerful programming language, offers a wide range of capabilities and applications. Some of the key things that C# technology can do include:

1. Building Desktop Applications: C# can be used to create Windows desktop applications with platforms like Windows Forms and WPF, enabling developers to design user-friendly interfaces and interact with the underlying operating system.

2. Developing Web Applications: C# is commonly used in web development with ASP.NET and ASP.NET Core, allowing developers to build dynamic and data-driven web applications, RESTful APIs, and web services.

3. Creating Mobile Apps: Through Xamarin, C# can be utilized to develop cross-platform mobile applications for iOS and Android, enabling code sharing and reducing development time.

4. Implementing Backend Services: C# is frequently used for server-side development to create backend services that handle business logic, data processing, and database interactions.

5. Building Games: C# is widely used in the gaming industry, particularly with Unity, a popular game engine. Unity allows developers to create 2D and 3D games for various platforms.

6. Working with Databases: C# can connect and interact with databases such as SQL Server, MySQL, and Oracle, enabling developers to manage data and perform database operations.

7. Implementing IoT Solutions: C# can be used in Internet of Things (IoT) projects to develop applications for embedded systems and IoT devices, facilitating communication with sensors and devices.

8. Developing Machine Learning Applications: C# can be employed for machine learning tasks with ML.NET, enabling developers to build and train machine learning models for data analysis and predictions.

9. Creating Cloud-Based Services: C# can be used to develop cloud-based services and applications through Microsoft Azure, providing scalability and reliability for cloud environments.

10. Automating Tasks: C# can be used for automation and scripting tasks, making it valuable for system administrators and IT professionals.

11. Integrating with IoT and Wearables: C# enables applications to connect and interact with IoT devices and wearables, facilitating data exchange and connectivity between devices.

Overall, C# technology offers a rich ecosystem and broad applicability, making it a popular choice for developers in various domains, including software development, web development, game development, IoT, machine learning, and cloud-based services. Its integration with the .NET framework and cross-platform capabilities contribute to its versatility and efficiency in modern software development.

How Will C#.NET Technology Change Our Life?

C#.NET technology has the potential to positively impact your life in various ways, especially if you are a software developer or someone who interacts with technology frequently. Here are some ways in which C#.NET technology can change your life:

1. Career Opportunities: Learning C#.NET opens up a wide range of career opportunities in the software development industry. With C#.NET skills, you can pursue roles in web development, mobile app development, game development, cloud computing, and more.

2. Software Development: As a software developer, C#.NET technology allows you to build robust and scalable applications for various platforms, including desktop, web, and mobile. It enables you to create innovative and feature-rich software solutions to solve real-world problems.

3. Cross-Platform Development: C#.NET technology is not limited to Windows. With the introduction of .NET Core, you can build cross-platform applications that run on Windows, macOS, and Linux, providing flexibility in application deployment.

4. Mobile App Development: If you're interested in mobile app development, C#.NET with Xamarin allows you to build native mobile apps for both iOS and Android platforms, sharing code between them, which saves time and effort.

5. Game Development: With Unity and C#, you can enter the world of game development. Creating 2D and 3D games becomes more accessible, and you can bring your creative ideas to life.

6. IoT and Embedded Systems: C# technology can be utilized in developing applications for Internet of Things (IoT) and embedded systems, enabling connectivity and control over various smart devices.

7. Machine Learning and Data Analysis: C#.NET has support for ML.NET, a machine learning framework, allowing you to explore the world of data science, predictive analytics, and artificial intelligence.

8. Automation and Scripting: C# can be used for automation and scripting tasks, simplifying repetitive tasks and increasing productivity in various domains.

9. Cloud Computing: C#.NET technology, with Microsoft Azure, provides the tools and services needed for cloud-based development, allowing you to create and deploy applications in the cloud.

10. Problem-Solving Skills: Learning C#.NET enhances your problem-solving and logical thinking abilities, which are valuable skills both in and outside the world of software development.

11. Collaboration and Contribution: C#.NET is a popular and well-supported language with a thriving community. By becoming part of this community, you can collaborate,

share knowledge, and contribute to open-source projects, making a positive impact on the tech community.

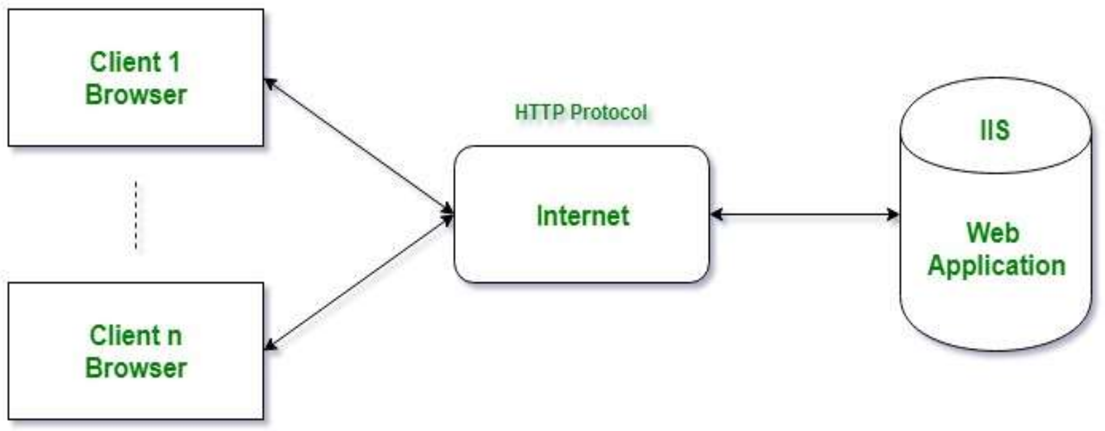
Overall, C#.NET technology can change your life by providing you with valuable skills, expanding your career options, and enabling you to create innovative solutions that can make a difference in various industries and domains. Whether you are a developer or someone interested in technology, C#.NET opens up new possibilities for growth and exploration in the ever-evolving world of software and technology.

ASP.NET

ASP.NET is a web application framework designed and developed by Microsoft. ASP.NET is open source and a subset of the .NET Framework and successor of the classic ASP(**A**ctive **S**erver **P**ages). With version 1.0 of the .NET Framework, it was first released in January 2002. So a question comes to mind that which technology we were using before the year 2002 for developing web applications and services? Answer is **Classic ASP**. So before .NET and ASP.NET there was *Classic ASP*.

ASP.NET is built on the CLR(Common Language Runtime) which allows the programmers to execute its code using any .NET language(C#, VB etc.). It is specially designed to work with HTTP and for web developers to create dynamic web pages, web applications, web sites, and web services as it provides a good integration of HTML, CSS, and JavaScript.

.NET Framework is used to create a variety of applications and services like Console, Web, and Windows, etc. But ASP.NET is only used to create web applications and web services. That’s why we termed ASP.NET as a subset of the .NET Framework.



What is web application:

A web application is an application installed only on the web server which is accessed by the users using a web browser like Microsoft Internet Explorer, Google Chrome, Mozilla FireFox, Apple Safari, etc. There are also some other technology like Java, PHP, Perl, Ruby on Rails, etc. which can be used to develop web applications. Web applications provide the cross-platform feature. The user needs only a web browser to access a web application. The web applications which are developed using the .NET framework or its subsets required to execute under the **Microsoft Internet Information Services(IIS)** on the server side. The work of IIS is to provide the web application's generated HTML code result to the client browser which initiates the request as shown in the below diagram.

- **Classic ASP**: It is the first server side scripting language developed by Microsoft.
- **ASP.NET**: It is web development framework and successor of Classic ASP. ASP.NET 4.6 is the latest version.
- **ASP.NET Core**: In November 2015, Microsoft released the 5.0 version of ASP.NET which get separated later and known as ASP.NET Core. Also, it is considered as an important redesign of ASP.NET with the feature of open-source and cross-platform. Before this version, ASP.NET is only considered as Windows-only version.
- **ASP.NET Web Forms**: These are the event-driven application model which are not considered a part of the new ASP.NET Core. These are used to provide the server-side events and controls to develop a web application.
- **ASP.NET MVC**: It is the Model-View-Controller application model which can be merged with the new ASP.NET Core. It is used to build dynamic websites as it provides fast development.

- **ASP.NET Web Pages**: These are the single page application which can be merged into ASP.NET Core.
- **ASP.NET API**: It is the Web Application Programming Interface(API).

Why ASP.NET:

- There are a lot of reasons which makes the ASP.NET popular among the developers. Some of the reasons are listed below:

- **Extending .NET Framework:** ASP.NET is a subset of .NET Framework as it extends the .NET Framework with some libraries and tools to develop web apps. The thing that it adds to the .NET Framework is *Libraries for common web patterns like MVC*, *Editor Extensions*, *the base framework to process the web requests*, and *web-page templating syntax like Razor*, etc.
- **Performance:** It is faster than the other web frameworks available in the market.
- **Backend Code:** With the help of ASP.NET you can write the backend code for data access and any logic in C#.
- **Dynamic Pages:** In ASP.NET, Razor provides the syntax for developing the dynamic web pages with the help of C# and HTML. ASP.NET can be integrated with JS(JavaScript) and it also includes the frameworks like React and Angular for the SPA(Single Page Application.)
- **Supporting different OS:** You can develop and execute ASP.NET apps on Windows, Linux, Docker, and MacOS. The Visual Studio provides the tools to build .NET apps different OS.

ADO.NET:

The .NET Framework includes its own data access technology i.e. **ADO.NET.** ADO.NET is the latest implementation of Microsoft's Universal Data Access strategy. ADO.NET consists of managed classes that allows .NET applications to connect to data sources such as Microsoft SQL Server, Microsoft Access, Oracle, XML, etc., execute commands and manage disconnected data..t Microsoft ADO.NET is the latest improvement after ADO.

Firstly, ADO.NET was introduced in the 10th version of the .NET framework, which helps in providing an extensive array of various features to handle data in different modes, such as connected mode and disconnected mode. In connected mode, we are dealing with live data and in disconnected mode, data is provided from the data store.

ADO.NET was primarily developed to address two ways to work with data that we are getting from data sources. The two ways are as follows :

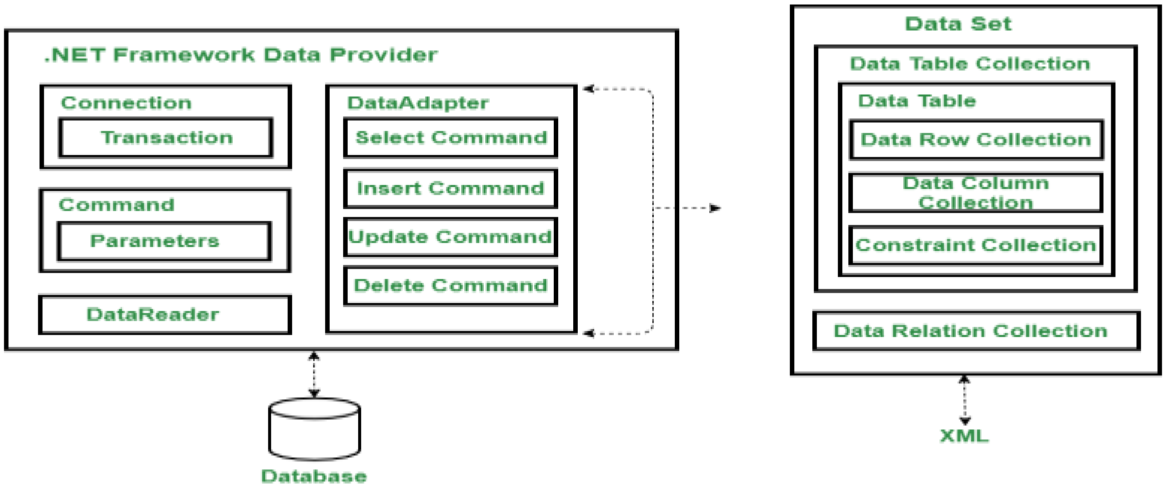
1. The first is to do with the user's need to access data once and to iterate through a collection of data in a single instance i.e caching the data in runtime memory.
2. The second way to work with data is in connected way which is we do not cache data. And we always go to database to retrieve it

Architecture of ADO.NET :

ADO.NET uses a multilayered architecture that revolves around a few key concepts as –

- asConnection
- Command
- DataSet objects

The ADO.NET architecture is a little bit different from the ADO, which can be shown from the following figure of the architecture of ADO.NET.



Features of ADO.NET :

The following are the features of ADO.NET –

- **Interoperability-**
  We know that XML documents are text-based formats. So, one can edit and edit XML documents using standard text-editing tools. ADO.NET uses XML in all data exchanges and for internal representation of data.
- **Maintainability** –
  ADO.NET is built around the idea of separation of data logic and user interface. It means that we can create our application in independent layers.
- **Programmability (Typed Programming)** –
  It is a programming style in which user words are used to construct statements or evaluate expressions. For example: If we want to select

the “Marks” column from “Kawal” from the “Student” table, the following is the way to do so:

- **Performance –**
  It uses disconnected data architecture which is easy to scale as it reduces the load on the database. Everything is handled on the client-side, so it improves performance.
- **Scalability –**
  It means meeting the needs of the growing number of clients, which degrading performance. As it uses disconnected data access, applications do not retain database lock connections for a longer time. Thus, it accommodates scalability by encouraging programmers to conserve limited resources and allow users to access data simultaneously.

6. System Design:

6.1 INTRODUCTION

Purpose and Scope

This section provides a brief description of the Systems Design Document’s purpose and scope.

Project Executive Summary

This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared. If appropriate, include the information discussed in the subsequent sections in the summary.

System Overview

This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable. Refer to the requirements trace ability matrix (RTM) in the Functional Requirements Document (FRD), to identify the allocation of the functional requirements into this design document.

Design Constraints

This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made by the project team in developing the system design.

Future Contingencies

This section describes any contingencies that might arise in the design of the system that may change the development direction. Possibilities include lack of interface agreements with outside agencies or unstable architectures at the time this document is produced. Address any possible workarounds or alternative plans.

Document Organization

This section describes the organization of the Systems Design Document.

Points of Contact

This section provides the organization code and title of the key points of contact (and alternates if appropriate) for the information system development effort. These points of contact should include the Project Manager, System Proponent, User Organization, Quality Assurance (QA) Manager, Security Manager, and Configuration Manager, as appropriate.

Project References

This section provides a bibliography of key project references and deliverables that have been produced before this point.

Glossary

Supply a glossary of all terms and abbreviations used in this document. If the glossary is several pages in length, it may be included as an appendix.

5.2 System Architecture:

6.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

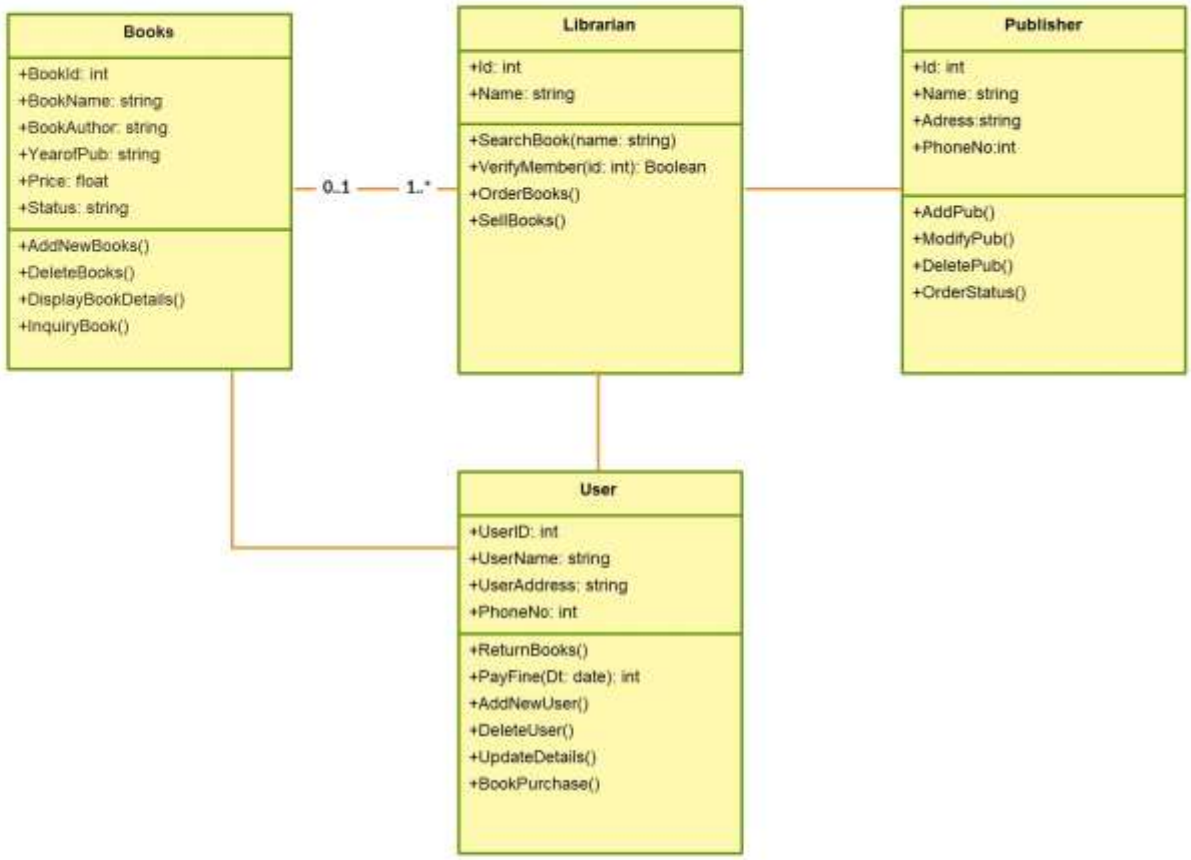
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

CLASS DIAGRAM:

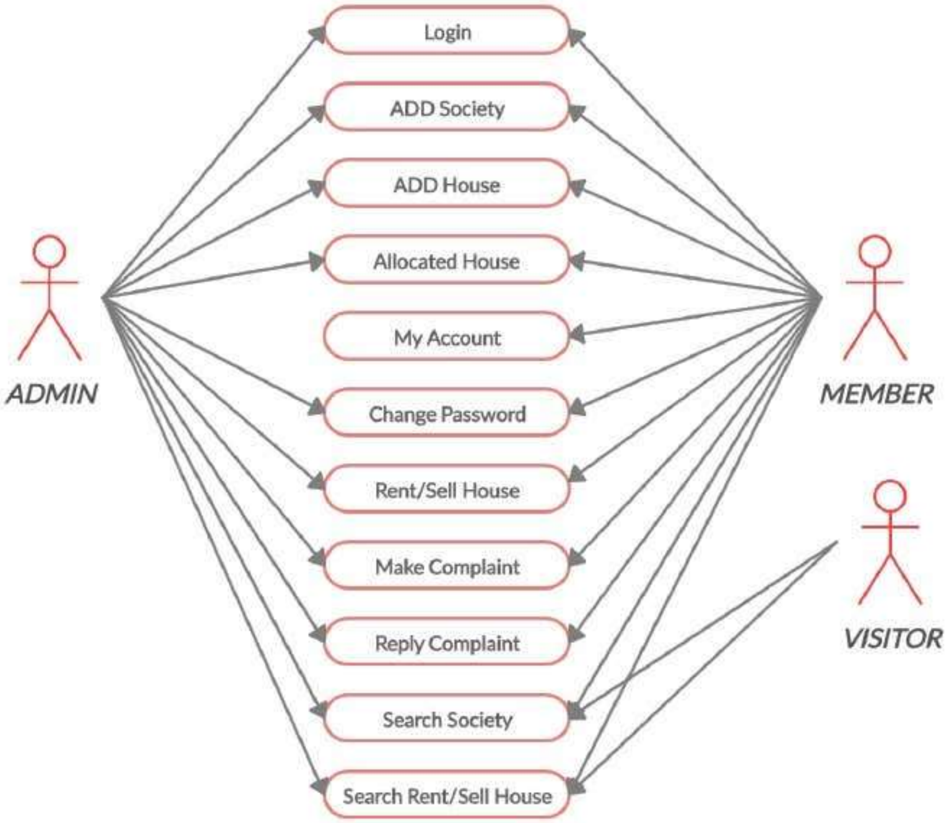
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



USE CASE DIAGRAM:

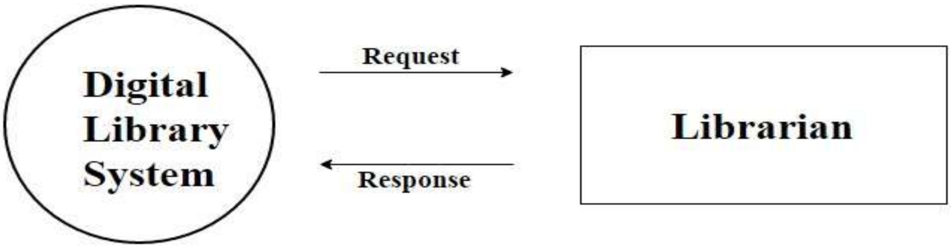
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Use Case Diagram



Sequence diagram :

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

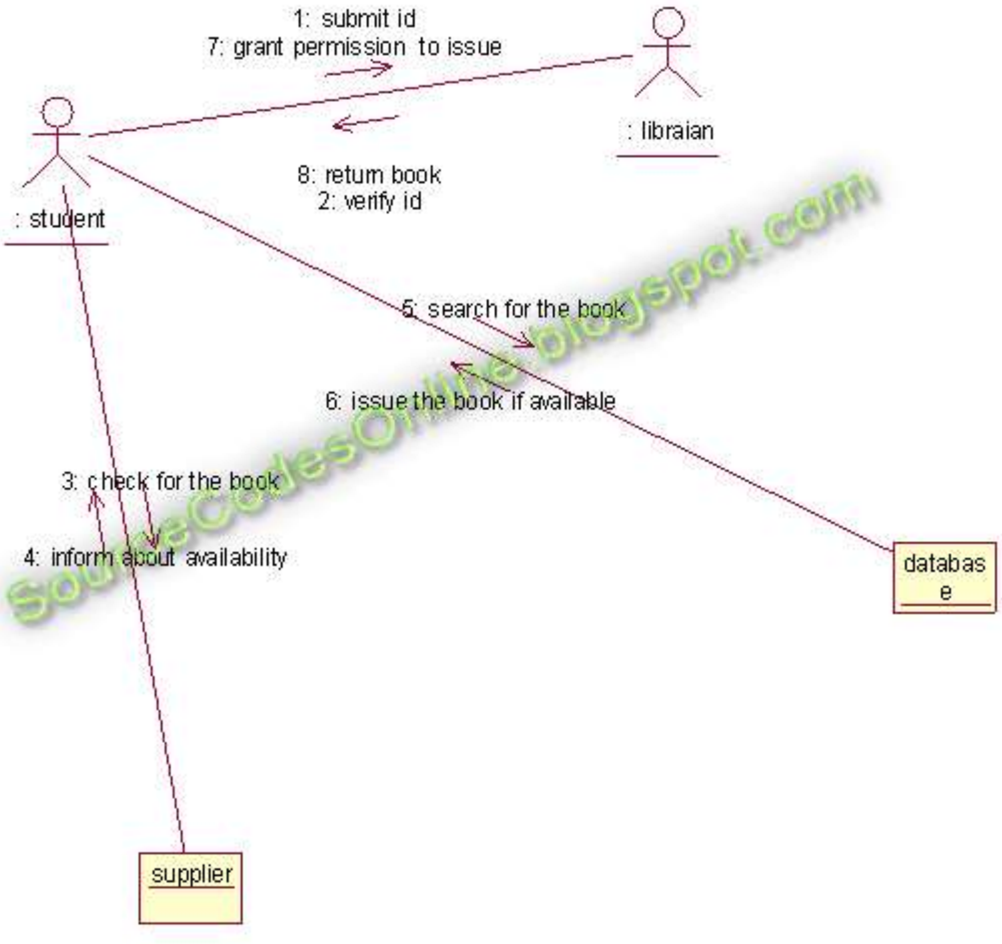


0 Level DFD

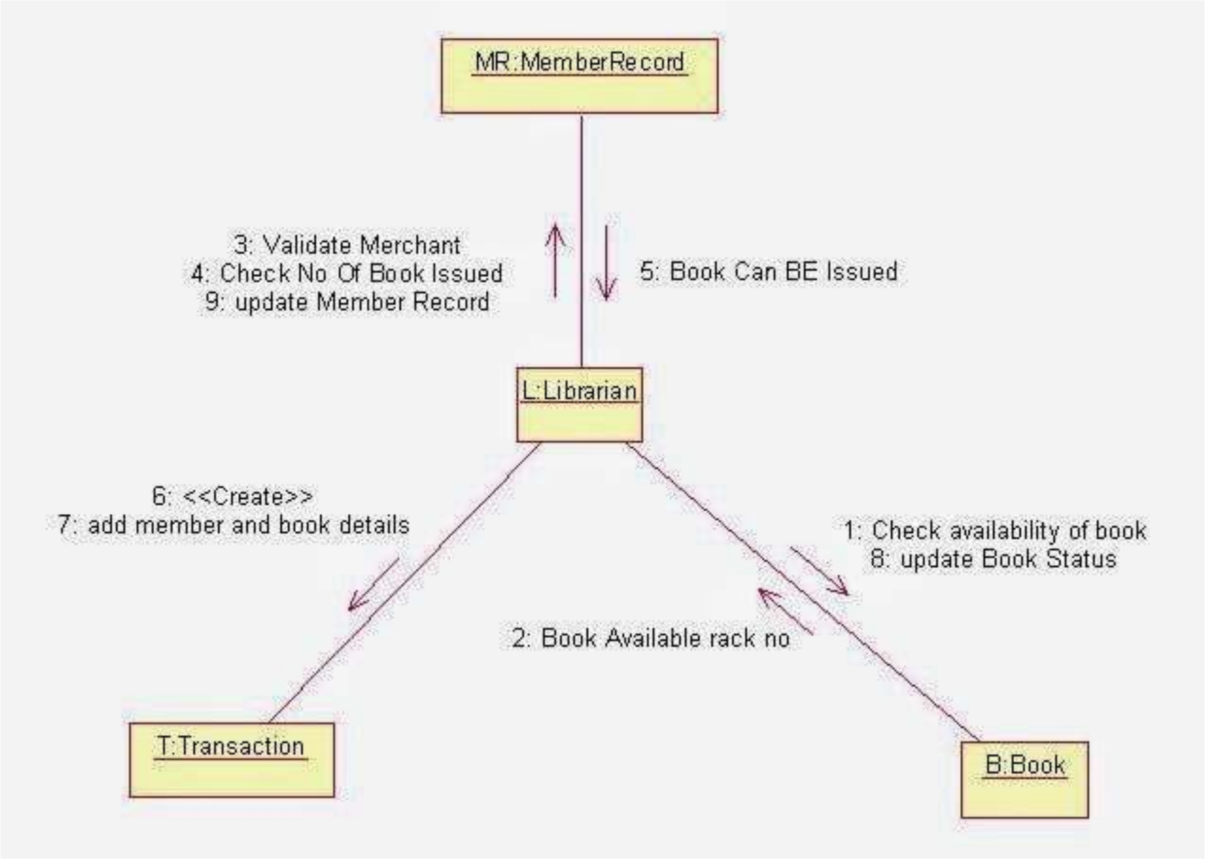
Collaboration Diagram :

collaboration diagram shows the object organization as shown below. Here in collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram.

The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization where as the collaboration diagram shows the object organization.

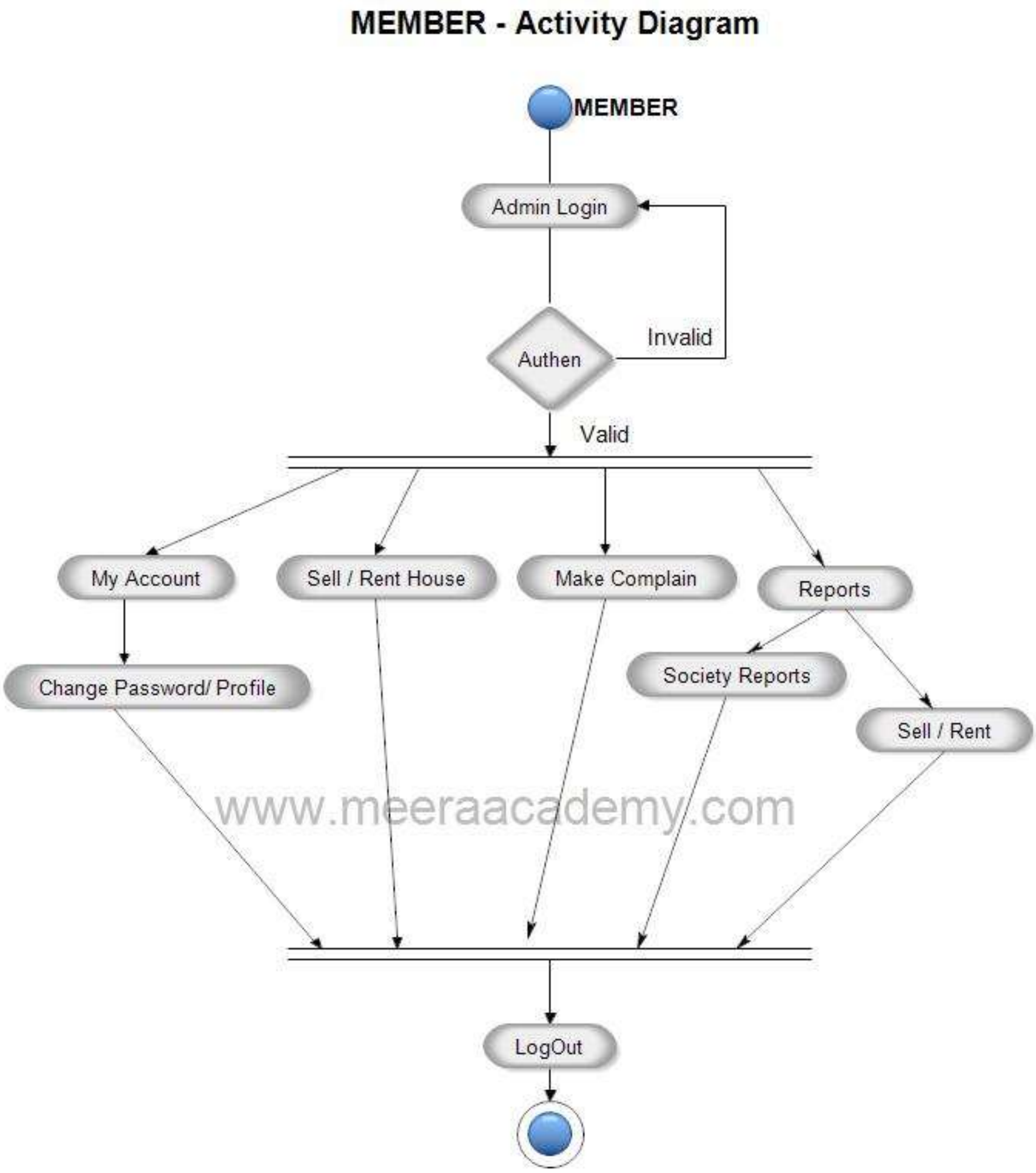


Collaberation diagram for member:



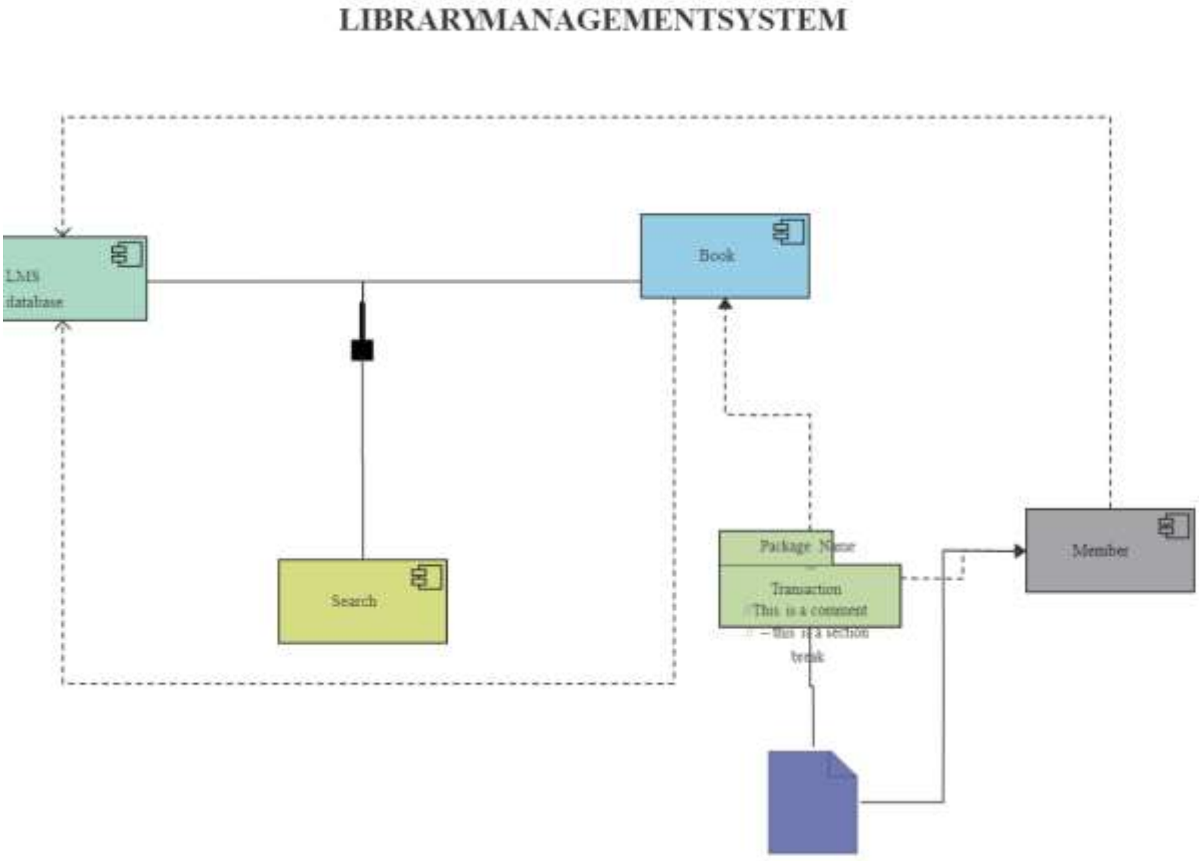
ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



Component Diagram:

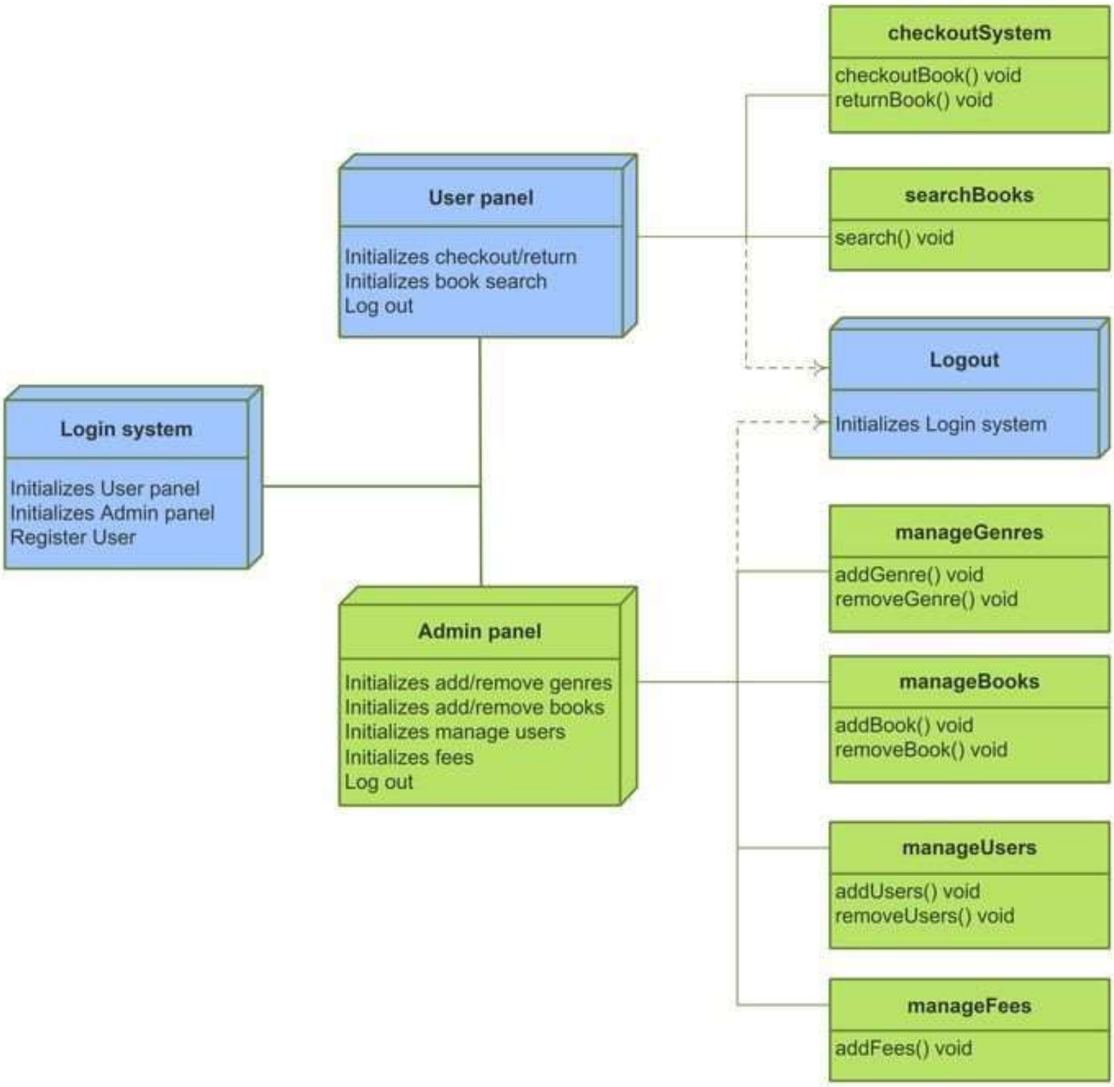
Component diagrams are different in terms of nature and behaviour. Component diagrams are used to model physical aspects of a system. Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.So from that point component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files etc.Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.



Deployment Diagram:

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

Deployment Diagram for Library Management System



6.4 SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ♦ ECONOMICAL FEASIBILITY
- ♦ TECHNICAL FEASIBILITY
- ♦ SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

7. Implementation Modules:-

7.1 Modules of the project:

SmartLib Library Management Software has a user friendly framework that consists of three modules.

The user classes can be divided into three categories..

1.Admin / Librarian

2.Staff

3.Student

7.2 Module Description:-

1) Administrator:

In a Computerized Library Management System the librarian manages the library activities in digital set-up. It negates the chances of losing and damage of paper works, documents and files. The transaction can be conducted with less time and more effect by the user. The books can be managed properly in the library with the system. Besides that, library management system also allows user to manage the publisher as well as lost book module. The library staff can handle the staff and view the report module

2)Staff

Library management system is all about organizing, managing the library and library-oriented tasks. It also involves maintaining the database of entering new books and the record of books that have been retrieved or issued, with their respective dates.

3)Student:

This system is designed for a user friendly environment so that student and staff of library can perform the various tasks easily and in an effective way. Through this part of the portal students can log in and have access to the library data.

7.3 Sample code (Implementation)

HOME.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Home.aspx.cs" Inherits="Home" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <style type="text/css">
    .style1
    {
        width: 100%;
    }
</style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <table class="style1">
    <tr>
        <td class="tblhead">
            Welcome to Digital Library System</td>
    </tr>
    <tr>
        <td>
            &nbsp;</td>
    </tr>
    <tr>
        <td style="text-align: center">
            <asp:Image ID="Image2" runat="server" ImageUrl="~/img/lin3.png" />
        </td>
    </tr>
</table>
</asp:Content>
```

Backend:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Home.aspx.cs" Inherits="Home" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <style type="text/css">
    .style1
    {
        width: 100%;
    }
</style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <table class="style1">
    <tr>
        <td class="tblhead">
            Welcome to Digital Library System</td>
    </tr>
    <tr>
        <td>
            &nbsp;</td>
    </tr>
    <tr>
        <td style="text-align: center">
            <asp:Image ID="Image2" runat="server" ImageUrl="~/img/lin3.png" />
        </td>
    </tr>
</table>
</asp:Content>
```

Default.aspx(home page):

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <link href="StyleSheet.css" rel="stylesheet" type="text/css" />
    <style type="text/css">
        .style1
        {
            width: 100px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
    <div id="head">

        <asp:Image ID="Image1" runat="server" ImageUrl="~/img/dlibrary500.png" />

    </div>
    <div id="main"><div id="img">
        <table class="style1">
            <tr>
                <td>
                    &nbsp;</td>
                <td>
        <asp:Image ID="Image2" runat="server" ImageUrl="~/img/lin3.png"
Height="314px"
```

```
                        style="margin-left: 94px" Width="561px" />
                </td>
            </tr>
        </table>
        </div>
    <div id="login">
        <table class="tbl">
            <tr>
                <td class="tblhead" colspan="2">
                    Login Area</td>
            </tr>
            <tr>
                <td>
                    &nbsp;</td>
                <td>
                    <asp:Label ID="lbl" runat="server" Font-Size="11px" 
ForeColor="Red"></asp:Label>
                </td>
            </tr>
            <tr>
                <td class="lbl">
                    UserName&nbsp; :</td>
                <td>
                    <asp:TextBox ID="txtuname" runat="server" 
CssClass="txt"></asp:TextBox>
                    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" 
runat="server" 
                        ControlToValidate="txtuname" ErrorMessage="!!!" 
ForeColor="Red" 
                        SetFocusOnError="True"></asp:RequiredFieldValidator>
                </td>
            </tr>
            <tr>
                <td class="lbl">
                    Password :
                </td>
                <td>
                    <asp:TextBox ID="txtupass" runat="server" CssClass="txt" 
TextMode="Password"></asp:TextBox>
                    <asp:RequiredFieldValidator ID="RequiredFieldValidator2" 
runat="server" 
                        ControlToValidate="txtupass" ErrorMessage="!!!" 
ForeColor="Red" 
                        SetFocusOnError="True"></asp:RequiredFieldValidator>
                </td>
            </tr>
            <tr>
                <td>
                    &nbsp;</td>
                <td>
                    <asp:RadioButton ID="rdolibrary" runat="server" 
Checked="True" 
                        ForeColor="Green" GroupName="a" Text="Librarian" />
&nbsp;<asp:RadioButton ID="rdosudent" runat="server" ForeColor="Green" 
GroupName="a" 
                        Text="Student" />
                </td>
            </tr>
            <tr>
                <td>
                    &nbsp;</td>
                <td>
```

```
                    <asp:Button ID="Button1" runat="server" CssClass="btn" 
Text="Login" 
                        Width="80px" Font-Size="10pt" onclick="Button1_Click" />
                </td>
            </tr>
        </table>
    </div>

    </div>
    </form>
</body>
</html>
```

Addmember.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class AddStudent : System.Web.UI.Page
{
    DS_BRANCH.BRANCH_SELECTDataTable BDT = new 
DS_BRANCH.BRANCH_SELECTDataTable();
    DS_BRANCHTableAdapters.BRANCH_SELECTTableAdapter BAdapter = new 
DS_BRANCHTableAdapters.BRANCH_SELECTTableAdapter();
    DS_STUDENT.STUDENT_SELECTDataTable SDT = new 
DS_STUDENT.STUDENT_SELECTDataTable();
    DS_STUDENTTableAdapters.STUDENT_SELECTTableAdapter SAdapter = new 
DS_STUDENTTableAdapters.STUDENT_SELECTTableAdapter();
    protected void Page_Load(object sender, EventArgs e)
    {
        lblmsg.Text = "";
        if (Page.IsPostBack == false)
        {
            BDT = BAdapter.SelectBranch();
            drpbranch.DataSource = BDT;
            drpbranch.DataTextField = "Branchname";
            drpbranch.DataValueField = "Branchid";
            drpbranch.DataBind();
            drpbranch.Items.Insert(0, "SELECT");
        }
    }
    protected void btnadd_Click(object sender, EventArgs e)
    {
        if (drpdd.SelectedIndex == 0 || drpmm.SelectedIndex == 0 || 
drpyyyy.SelectedIndex == 0)
        {
            lblmsg.Text = "Select Proper BithDate !!";
        }
        else
        {
            string gen="";
            if(rdomale.Checked==true)
            {
            gen="Male";
            }
            else
            {
            gen="Female";
            }
            if (FileUpload1.HasFile)
            {
                FileUpload1.SaveAs(Server.MapPath("~/img/") + 
FileUpload1.FileName);
                DateTime bdate = Convert.ToDateTime(drpdd.SelectedItem.Text + " " 
+ drpmm.SelectedItem.Text + " " + drpyyyy.SelectedItem.Text);
                SAdapter.Insert(txtsname.Text, drpbranch.SelectedItem.Text, 
txtmobile.Text, txtaddress.Text, txtcity.Text, txtpincode.Text, bdate, gen, 
txtemail.Text, txtpass.Text,"~/img/"+FileUpload1.FileName);
                lblmsg.Text = "Student Added Successfully !!";

            }
            else
            {
                DateTime bdate = Convert.ToDateTime(drpdd.SelectedItem.Text + " " 
+ drpmm.SelectedItem.Text + " " + drpyyyy.SelectedItem.Text);
                SAdapter.Insert(txtsname.Text, drpbranch.SelectedItem.Text, 
txtmobile.Text, txtaddress.Text, txtcity.Text, txtpincode.Text, bdate, gen, 
txtemail.Text, txtpass.Text, "~/img/std.png");
                lblmsg.Text = "Student Added Successfully !!";


            }
            txtsname.Text=""; drpbranch.SelectedIndex=0;
            txtmobile.Text="";
            txtaddress.Text=""; txtcity.Text="";
            txtpincode.Text="";
            txtemail.Text="";
             txtpass.Text="";
             drpdd.SelectedIndex = 0;
             drpmm.SelectedIndex = 0;
             drpyyyy.SelectedIndex = 0;


        }
    }
}
```

8. Testing & Validation

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

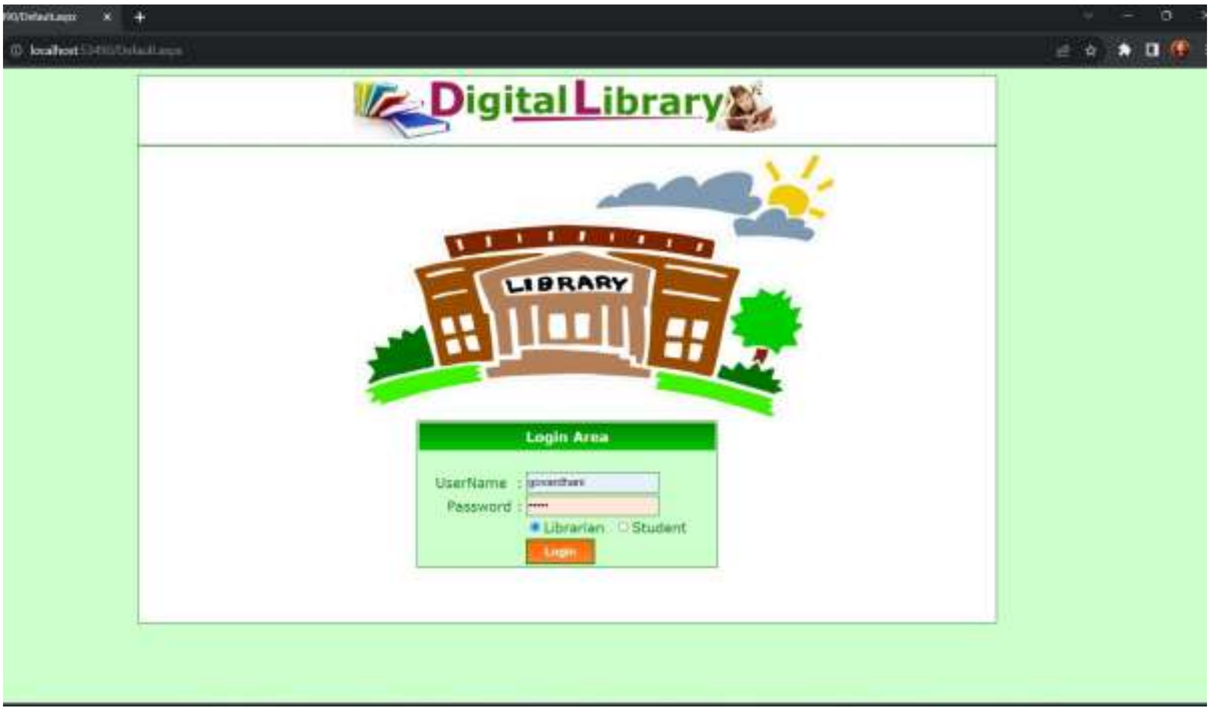
Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

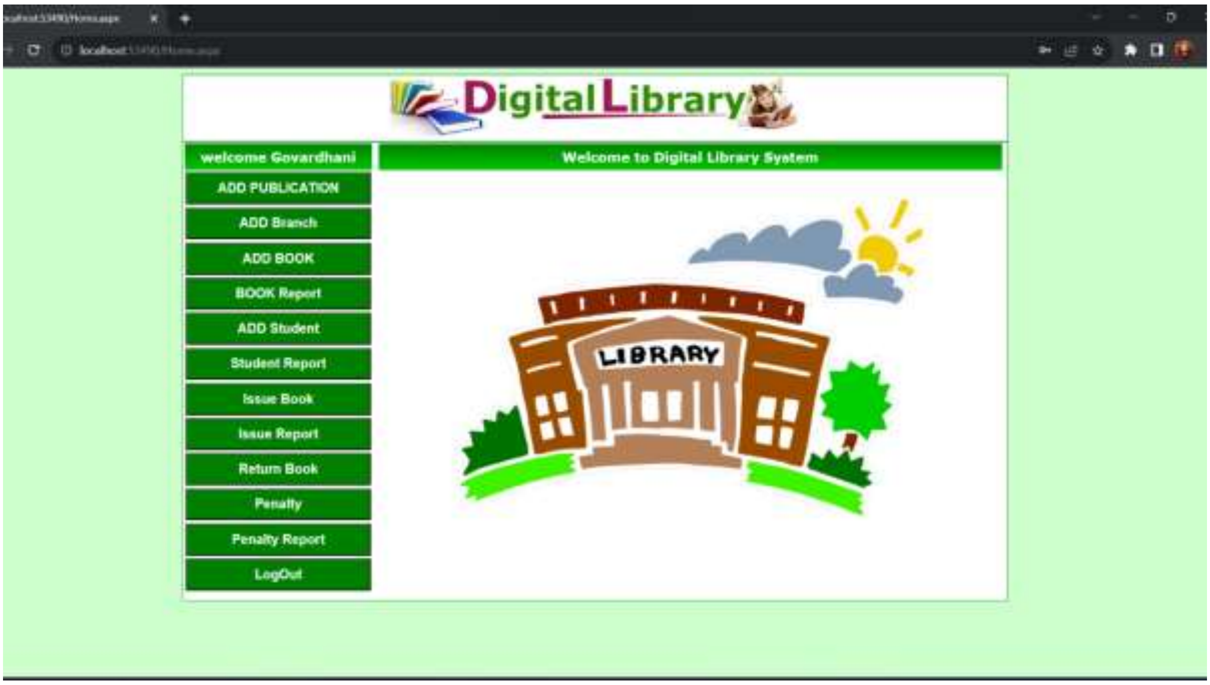
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

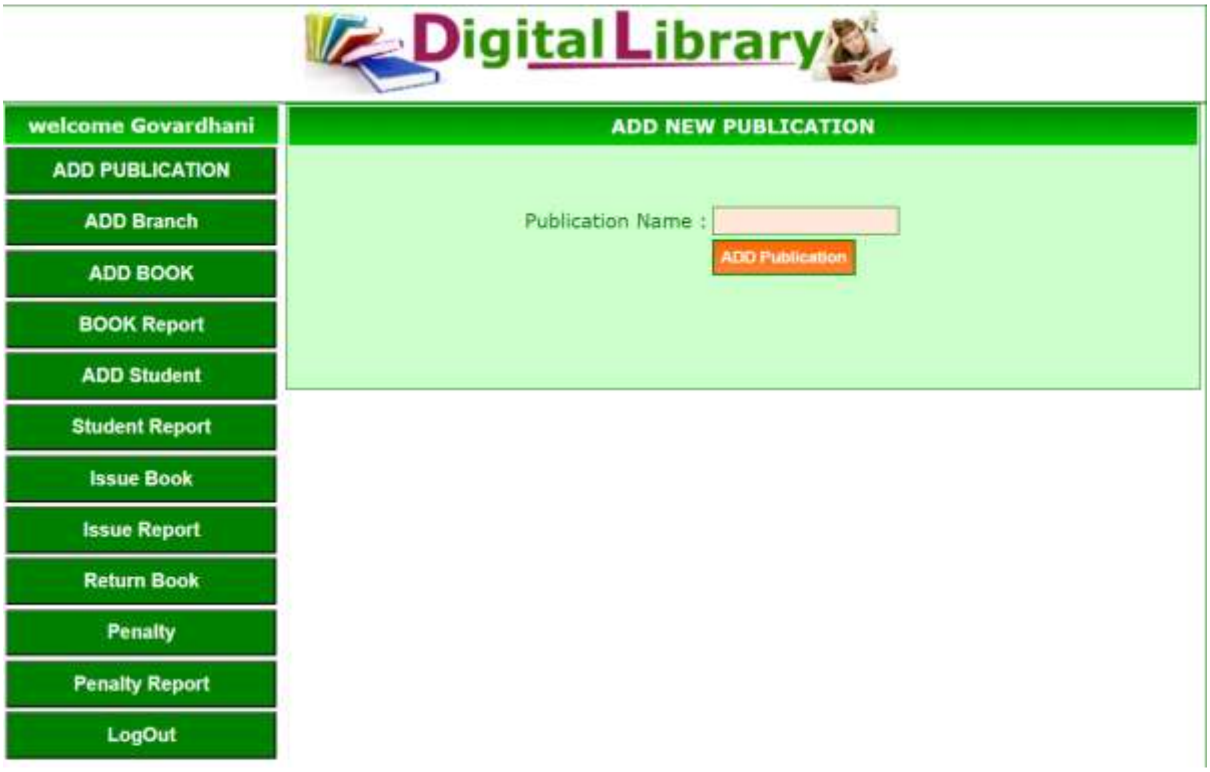
9.Output Screens:

HomePage:



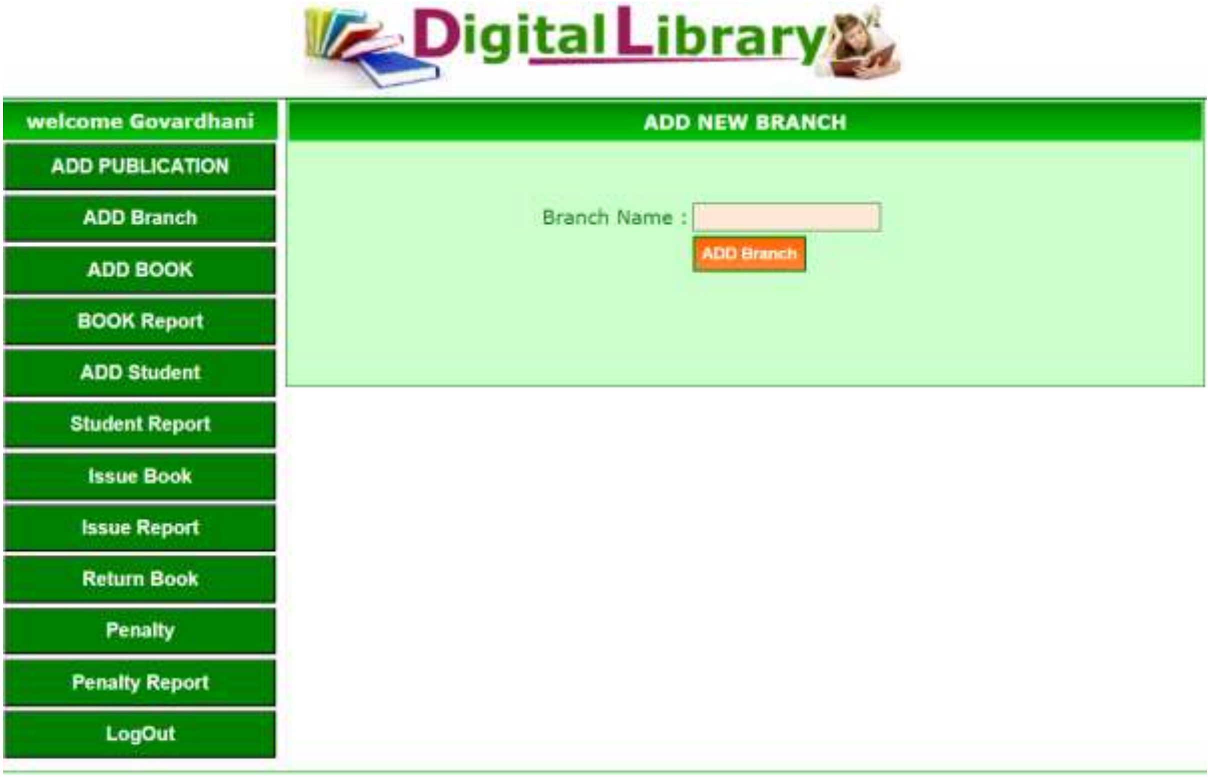
UserName : govardhani
Password : *****
● Librarian ○ Student





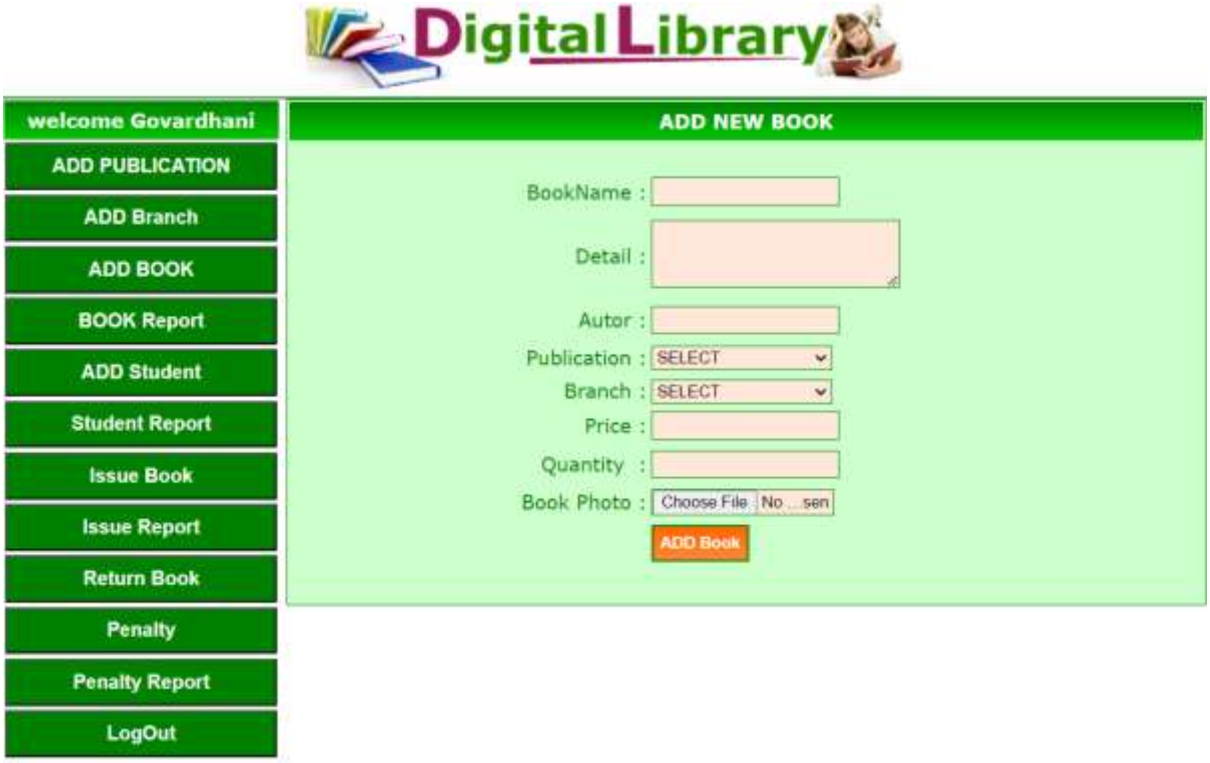
ADD NEW PUBLICATION

Publication Name :



ADD NEW BRANCH

Branch Name :



ADD NEW BOOK

BookName :

Detail :

Autor :

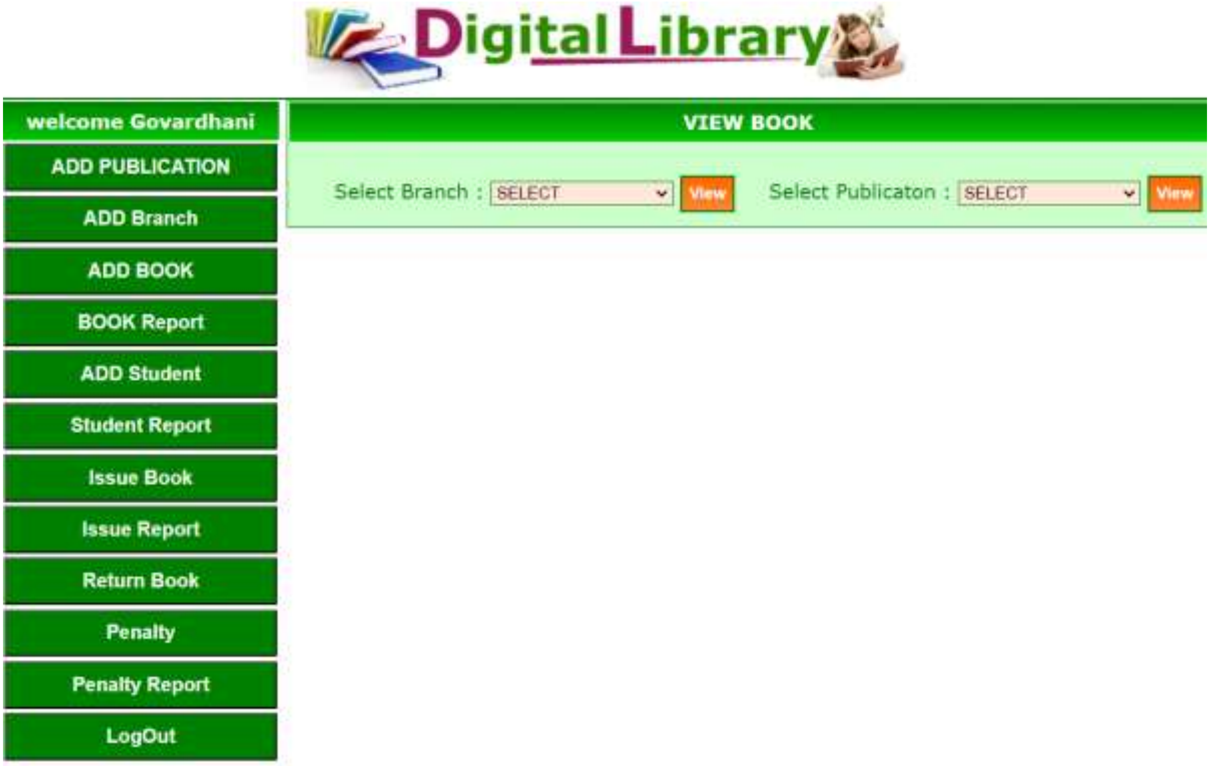
Publication : SELECT

Branch : SELECT

Price :

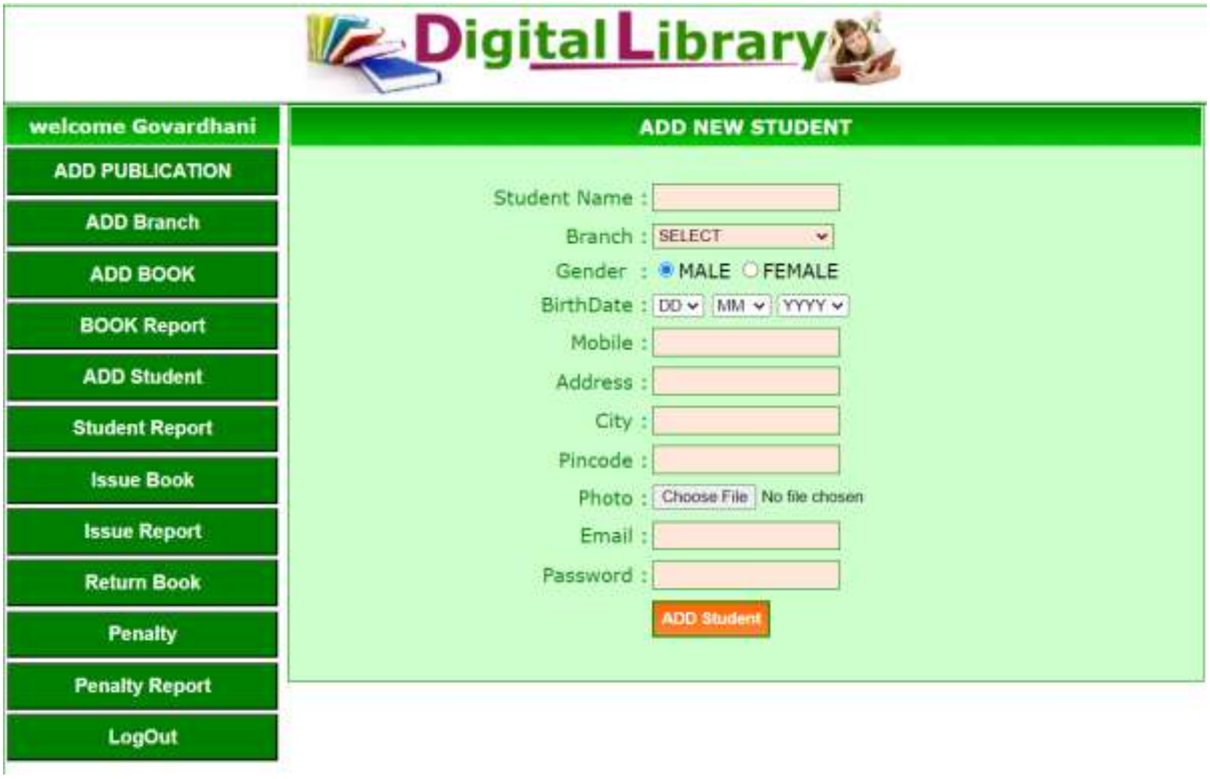
Quantity :

Book Photo : No ...sen



Select Branch : SELECT

Select Publicaton : SELECT



Student Name :

Branch : SELECT

Gender : ● MALE ○ FEMALE

BirthDate : DD MM YYYY

Mobile :

Address :

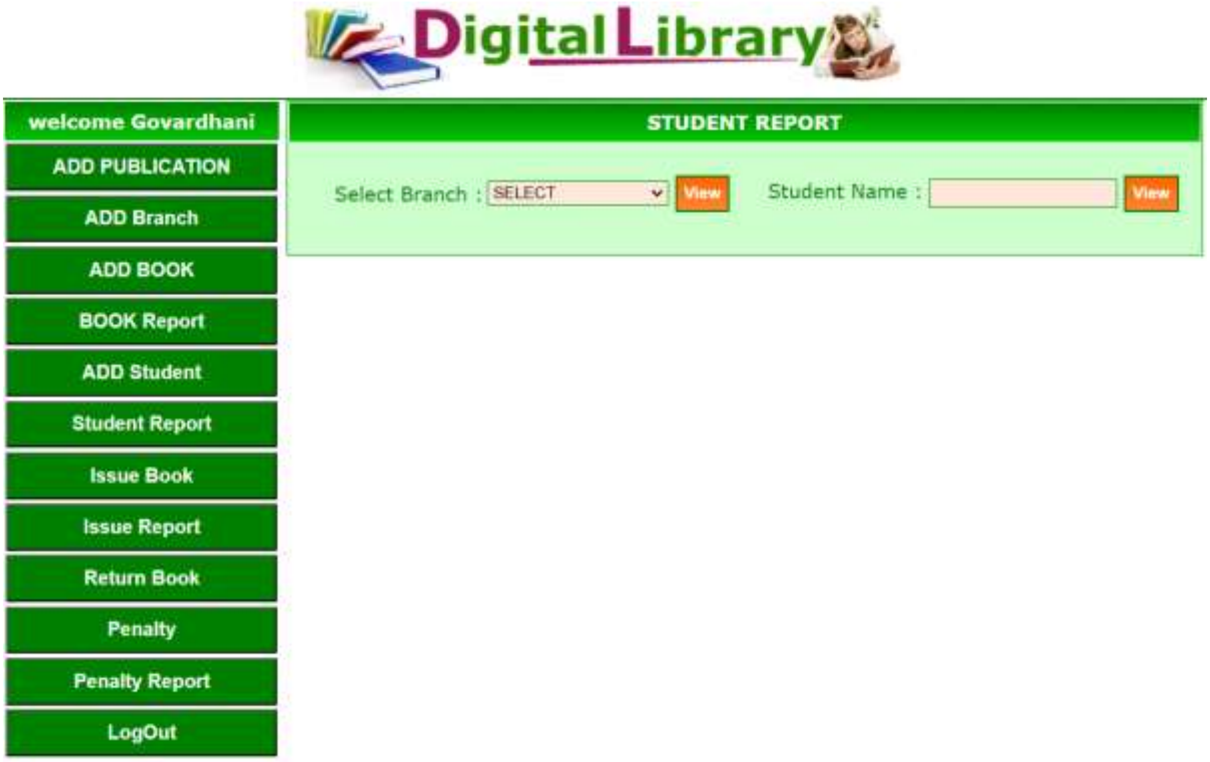
City :

Pincode :

Photo : No file chosen

Email :

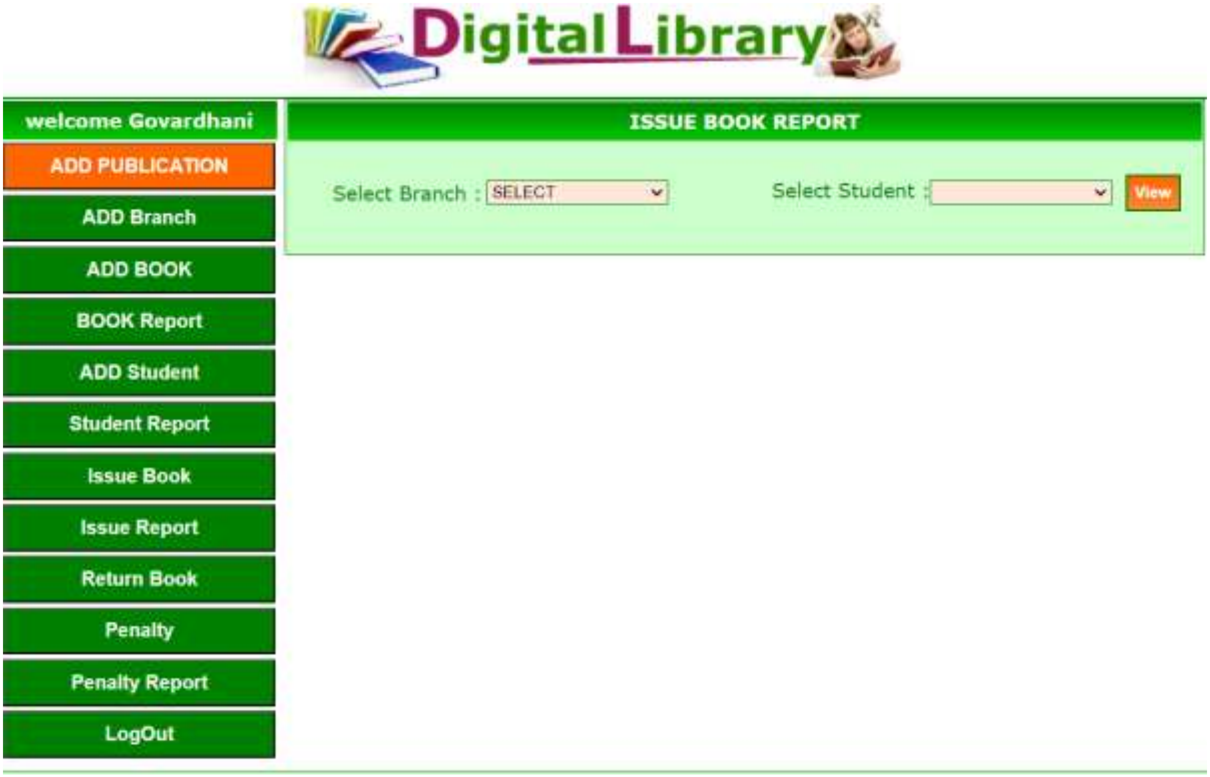
Password :



STUDENT REPORT

Select Branch : SELECT

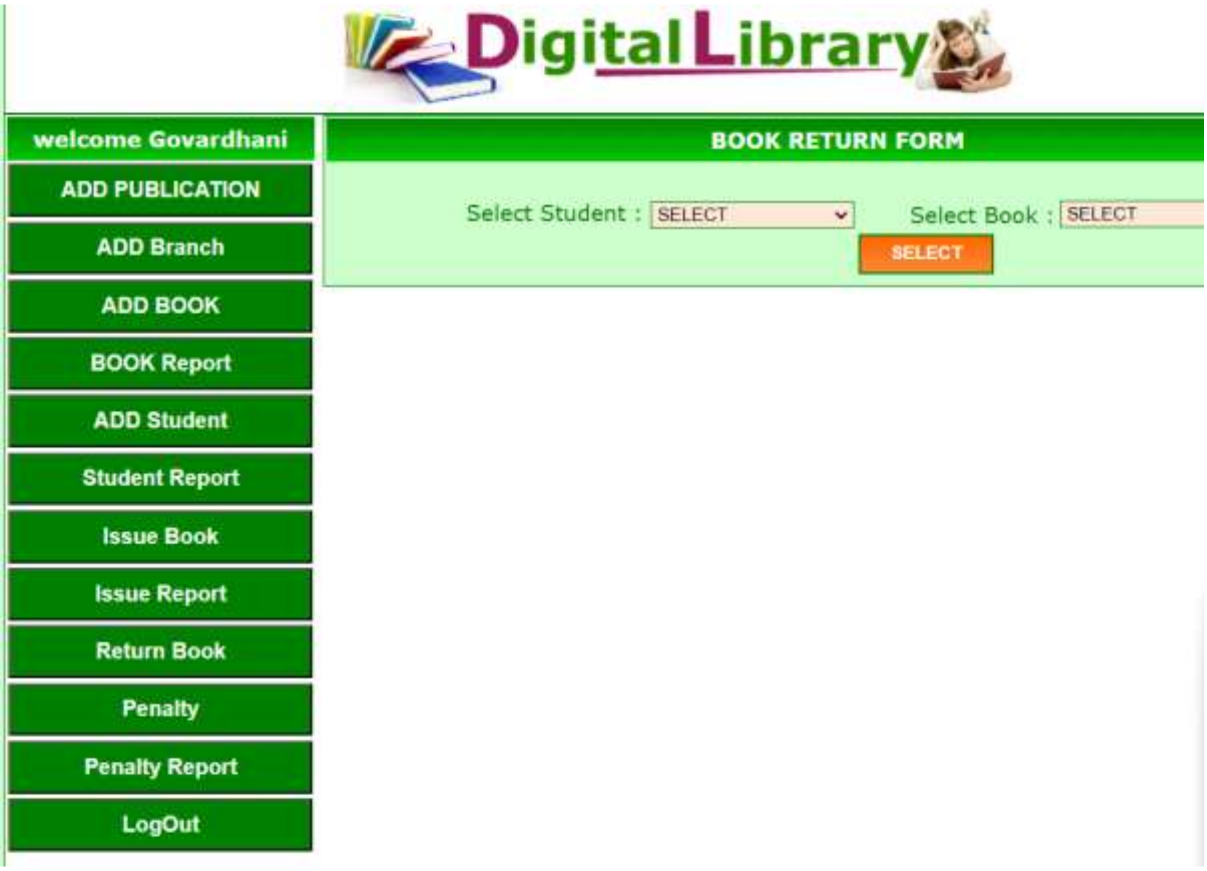
Student Name :



ISSUE BOOK REPORT

Select Branch : SELECT

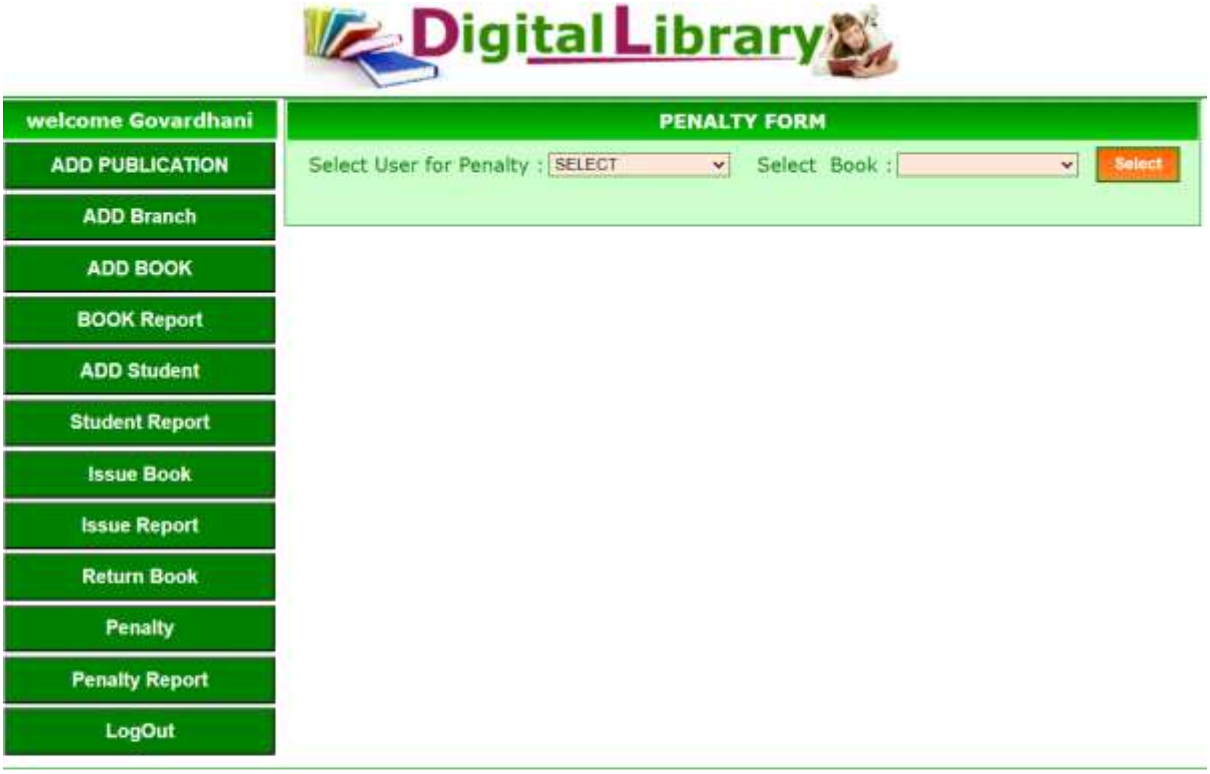
Select Student :



BOOK RETURN FORM

Select Student : SELECT

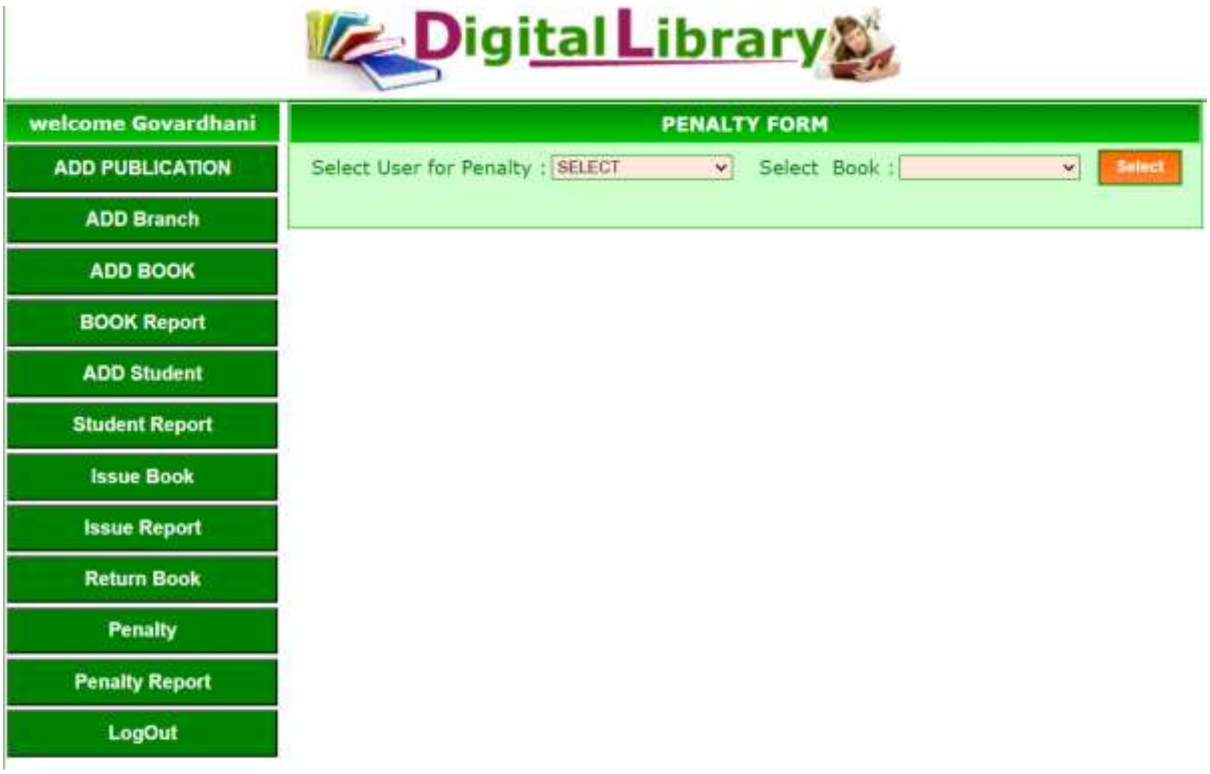
Select Book : SELECT



PENALTY FORM

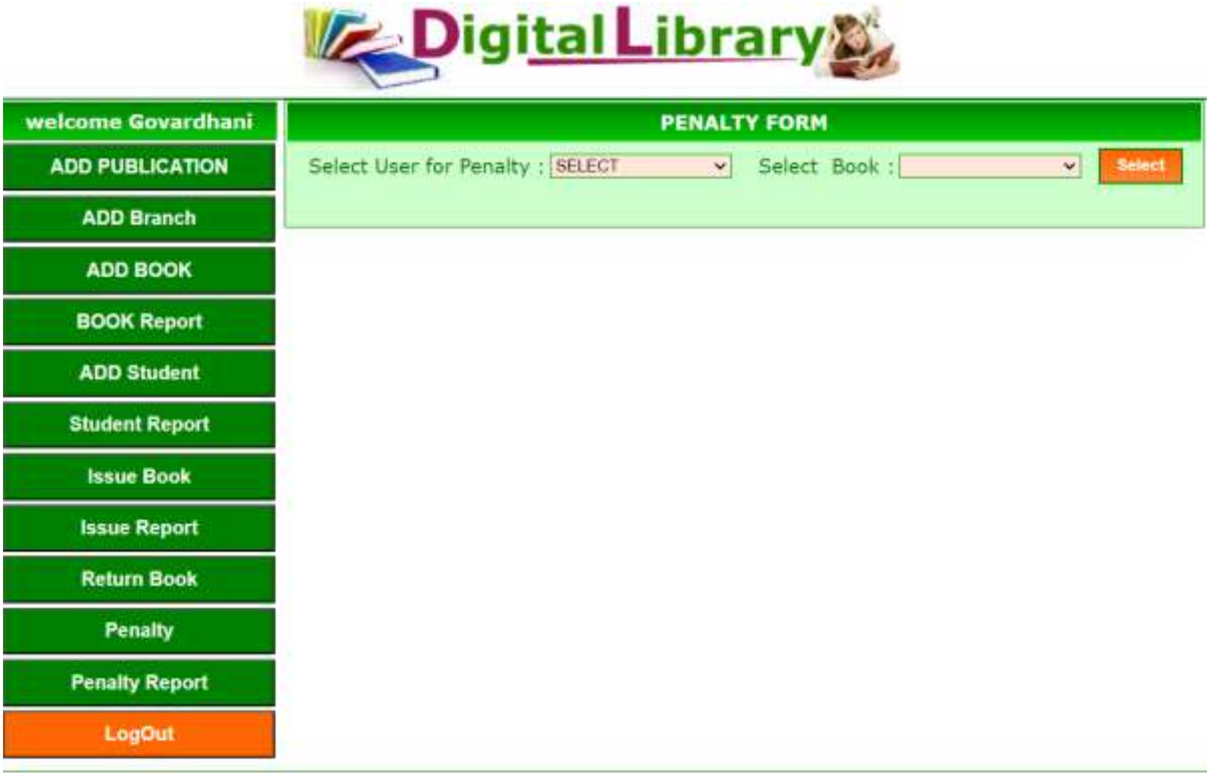
Select User for Penalty : SELECT

Select Book :



PENALTY FORM

Select User for Penalty : SELECT Select Book : Select



PENALTY FORM

Select User for Penalty : SELECT Select Book : Select

10)Conclusion:

The currently implemented Library Management System provides administrators and users powerful features to manage allocating books and performing some actions. This paper mainly focuses on how we can improve the traditional method of working of a library because the traditional method includes doing all the things in manual mode which is slow, less efficient, less secure, and difficult to manage. The solution to this is an online library management system which take care of all the work by automating and digitizing the whole process. Our application is based on Java and is linked to a relational database (sql). The frontend part has been coded using Java and its packages like awt and swing. The backend is supported and connected with database using java, its libraries and APIs. With the increase in the workload of the library, new features can be added to the existing application to make it relevant in the future as well.