FINAL PROJECT

Library Management System

ONL00080489 K.Ravindu Madhushankha
DISE | Tangalle Branch

Task 01

a.

The first step is **Planning and requirement analysis**. The requirements are given in the scenario. An online host was bought to store the database. Because it is an easy way to maintain all the tables in our database. Hence an internet connection needs to run this application. The software name was named as **Library**.
All the company details related to this project should be collected. A computer which supports this software and an IDE are additional needs.
I divided Library Management system to two parts. They are **Library** and **Library Manager**. In the Library, search books, borrow books and update own info of members can be done. In the Library Manager, maintain books, maintain member, maintain report… etc. can be done.

The requirements in this software for members

- Sign in
- Sign up
- Change their own details
- Borrow books
- Search books
- Show availability of books
- Show notifications
- Call & email

The requirements in this software for admins

- Sign in
- Change their own details
- Make admins to maintain the system
- Add books
- Add members
- Add notices
- Remove books
- Remove members
- Remove notices
- Borrow books
- Search books
- Search members
- Update returned book details
- Show availability of books
- Show notifications
- Call & email
- Maintain all the data of tables like members, books, admins, notifications, reports…etc.

Finally this project was **analyzed** as following below.

- The software was named as Library.
- Students, lecturers and people who need to borrow books are used this software.
- This software is used to borrow books and maintain all the data of tables in the database without going to the library branch.
- This software is worked from an online database. So an internet connection should be needed.
- JAVA platform is needed to create this software.

The next step is **Feasibility Study**. In this phase, these feasibilities are checked.

- We can complete the project within the budget. Because the cost goes to only for online hosting.
- We can handle this project from online Database Management System.
- We can create operations which is expected by the client.
- The current computer system supports the software.
- The project can be completed within the given schedule.

The next step is **Design the Software**. In this phase, flow charts, ER diagrams, use case diagrams, class diagrams are drawn to design the software before developing the system. This designing phase serves as that for the next phase of the model. App.gleek.io and starUML can be used to draw these diagram.

Once the design phase is over, the next phase is **Develop the project**. Coding is being done in this step. In this case JAVA programming language is c hosen for writing to code. The netbeans IDE is chosen as the developing environment.

Once the developing phase is over, Next step is **testing**. In this testing phase, we may find some bugs/defects we must fix the bug and run the code again for a re-test. Testing process continues until the software is bug-free and stable.

Once testing phase is over then the final **Deployment** process starts. Then the final software is released and checked for deployment issues if any.

Once students start using the developed system, **Maintenance** Process starts; following activities occur.

- Bug fixing – bugs are reported because of some scenarios which are not tested at all.
- Upgrade – Upgrading the application to the newer versions of the software.
- Enhancement – Adding some new features into the existing system.

b.

Waterfall Methodology was selected as the recommended model for this project.

Waterfall Methodology compared with Agile Model

- Waterfall has a fixed timeline. The idea is that the start and finish of the project are already mapped out from the beginning.
- Waterfall is not as flexible as Agile because each phase needs to be fully completed before moving on to the next phase.
- The budget for projects using waterfall methodology is generally fixed. Because the project is determined from start to finish, there is less room to change the budget mid-project.

Waterfall Methodology compared with Spiral Model

- Project size: The waterfall model is best for small projects in which the goals are clear, while the spiral model is better for larger projects of greater complexity.
- Requirement gathering: In the waterfall model, the developers gather and determine project requirements once, at the beginning of the process, and build their project tasks on these determinations. In the spiral model, the developers gather requirements at the start of each iteration.
- Risk identification and testing frequency: Similarly, the waterfall model can identify risks only toward the end of development during the testing stage, whereas the spiral model integrates risk identification and testing in each iteration.

c.

A **software** is a collection of instructions that enable the user to interact with a computer, its hardware or perform tasks.

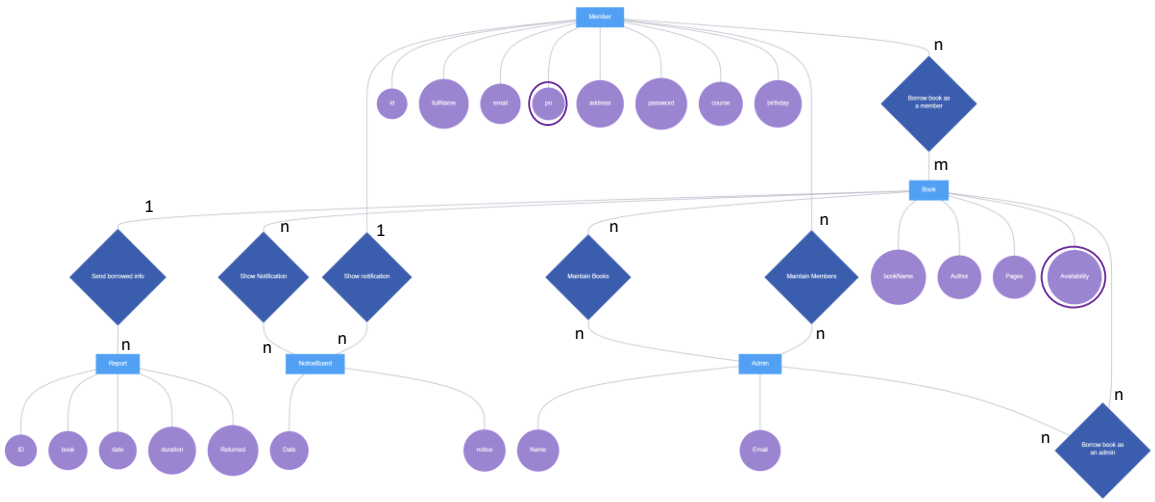
This software can be grouped as an **Application Software**.

This product for an individual client. Individual client may be a company or group of persons. This product mostly has a distinct need in the market only for a limited time and is for the specialized business needs. So this software can be grouped as a **Bespoke Software**.

Task 02

a.

ER Diagram



b.

Normalization

ID	Name
2022MAT0045	K.R.Madhushankha
2022TAN0203	H.G.Sumudu
2022GAL0003	M.J.Aariyadasa

BookIndexNumber	Name
1	Pariganaka - 2022 February
2	Madoldoowa
3	JAVA Programming II

Date	Notice
1/4/2022	2022MAT0045 has borrowed Pariganaka - 2022 February for 1 month.
1/29/2022	2022TAN0203 has borrowed Madoldoowa for 2 months.
3/18/2022	2022GAL0003 has borrowed JAVA Programming II for 1 month.

ID	BookIndexNumber	Date	Duration	Returned
2022MAT0045	1	1/4/2022	for 1 month	Yes
2022TAN0203	2	1/29/2022	for 2 month	Yes
2022GAL0003	3	3/18/2022	for I month	No

c.

- I decided to make admins from member table. When we make an admin from the member table, admin goes to Admin table. So we need two tables above the database. These tables are Admin and Member.
- And we need a Book table for listing boos information.
- The Notice table was made to inform update of members and update of books.
- There is a one more table named as Report to checked borrowed and returned books information.
- IndexNumber was selected as the primary key. Because it is easier to maintain sql quires.
- Finally **Member, Admin, Book, Notice, Report** are the main tables in our database.

	Table	Action	Rows	Type	Collation	Size	Overhead
☐	admin	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 KiB	-
☐	book	Browse Structure Search Insert Empty Drop	12	InnoDB	latin1_swedish_ci	16 KiB	-
☐	member	Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	16 KiB	-
☐	notice	Browse Structure Search Insert Empty Drop	22	InnoDB	latin1_swedish_ci	16 KiB	-
☐	Report	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 KiB	-
	5 tables	Sum	53	InnoDB	latin1_swedish_ci	80 KiB	0 B

Book table

```
SELECT * FROM `book`
```

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	Name	Type	Author	Pages	Availability	IndexNumber
☐ Edit Copy Delete	Hathpana	Story(Sinhala)	Munudasa Kumarathunga	61	6	8
☐ Edit Copy Delete	Infomation Technology I	Book(English)	ESOFT	120	5	12
☐ Edit Copy Delete	English Master I	Book(English)	Vijaya	32	26	14
☐ Edit Copy Delete	English Master II	Book(English)	Vijaya	32	29	15
☐ Edit Copy Delete	English Master III	Book(English)	Vijaya	32	13	16
☐ Edit Copy Delete	English Master IV	Book(English)	Vijaya	32	15	17
☐ Edit Copy Delete	C# Programming	Book(Sinhala)	ESOFT	48	10	26
☐ Edit Copy Delete	Python Programming	Book(Sinhala)	ESOFT	68	10	27
☐ Edit Copy Delete	Madoldoowa	Novel(Sinhala)	Martin Wickramasinge	441	4	36
☐ Edit Copy Delete	Infomation Technology II	Book(English)	ESOFT	96	4	39
☐ Edit Copy Delete	JAVA Programming	Book(Sinhala)	ESOFT	54	10	42
☐ Edit Copy Delete	Pariganaka 2022-02	Magazine(Sinhala)	Vijaya	40	2	43

Check all *With selected:* Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

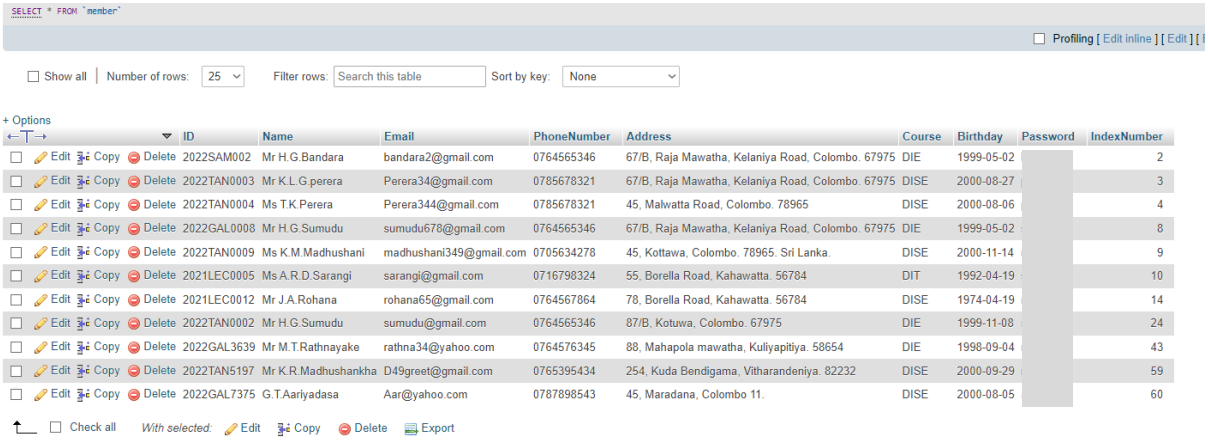
Admin Table

+ Options

	Name	Email	Password	IndexNumber
Edit Copy Delete	Mr H.G.Sumudu	sumudu678@gmail.com	sumudu8	8
Edit Copy Delete	Ms A.R.D.Sarangi	sarangi@gmail.com	sarangi	10
Edit Copy Delete	Mr J.A.Rohana	rohana65@gmail.com	rohana65	14
Edit Copy Delete	Mr K.R.Madhushankha	D49greet@gmail.com	ravii#00	59

.

Member Table



Notice Table

+ Options

	Date	Notice	IndexNumber
Edit Copy Delete	2022-01-07	New books were added!	1
Edit Copy Delete	2022-01-08	2022GAL002, Please return your book before 2022-01...	2
Edit Copy Delete	2022-01-09	English Master II is available now.	4
Edit Copy Delete		English Master III is avalable now.	5
Edit Copy Delete	2022-01-10	2021LEC005 has borrowed C# Programming for 1 month...	19
Edit Copy Delete	2022-01-12	New book is added!	25
Edit Copy Delete	2022-01-13	2021LEC005 has borrowed JAVA Programming for 1 mon...	33
Edit Copy Delete	2022-01-13	2022SAM002 has borrowed C# Programming for 2 month...	34
Edit Copy Delete	2022-01-14	2022SAM002 has requested to remove account!	35
Edit Copy Delete	2022-01-17	2022GAL3639 has borrowed English Master IV for 2 w...	48
Edit Copy Delete	2022-01-22	2021LEC0005 has borrowed Infomation Technology I f...	59
Edit Copy Delete	2022-01-28	2022GAL3639, Return your book(English Master IV) b...	62
Edit Copy Delete	2022-03-06	A new book is added!	63
Edit Copy Delete	2022-03-17	A new book is added!	65
Edit Copy Delete	2022-03-17	2022GAL7375 has borrowed Python Programming for 1 ...	66
Edit Copy Delete	2022-03-26	2021LEC0005, Your issued time is over! Please retu...	67
Edit Copy Delete	2022-03-26	2022TAN5197 has borrowed C# Programming for 1 mont...	69
Edit Copy Delete	2022-03-26	2022TAN5197 has borrowed JAVA Programming for 1 mo...	70
Edit Copy Delete	2022-03-28	A new book is added!	78
Edit Copy Delete	2022-03-29	2022TAN5197 has borrowed English Master I for 1 mo...	85
Edit Copy Delete	2022-03-29	2021LEC0005 has borrowed Python Programming for 2 ...	86
Edit Copy Delete	2022-03-29	2021LEC0012 has borrowed JAVA Programming for 3 mo...	87

Report Table

```
SELECT * FROM `Report`
```

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	IndexNumber	ID	Book	Date	Duration	Returned
Edit Copy Delete	3	2022TAN5197	Infomation Technology I	2022-03-29	for 1 month	No
Edit Copy Delete	4	2022TAN5197	English Master I	2022-03-29	for 1 month	No
Edit Copy Delete	5	2021LEC0005	Python Programming	2022-03-29	for 2 months	No
Edit Copy Delete	6	2021LEC0012	JAVA Programming	2022-03-29	for 3 months	No

d.

This code was used to create my database.

CREATE DATABASE sql6464172;

These codes were used to create tables.

CREATE TABLE Member(IndexNumber int, Name varchar(255), Email varchar(255), Address varchar(255), PhoneNumber varchar(255), Birthday varchar(255), Course varchar(255), Password varchar(255));

CREATE TABLE Admin(IndexNumber int, Name varchar(255), Email varchar(255), Password varchar(255));

CREATE TABLE Book(IndexNumber int, Name varchar(255), Author varchar(255), pages varchar(255), Availabitily varchar(255));

CREATE TABLE Notice(IndexNumber int, Date varchar(255), Notice varchar(255));

CREATE TABLE Report(IndexNumber int, ID varchar(255), Book varchar(255), Date varchar(255), Duration varchar(255), Returned varchar(255));

These codes were used to maintain above details in IDE.

```
void book(){
    try {
        String sql="SELECT Name, Type, Author, Pages, Availability, IndexNumber FROM book ORDER BY IndexNumber DESC";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        jTable1.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        NewJFrame4 cf=new NewJFrame4();
        //cf.setVisible(true);
    }
}
```

```
public void bookSearch(){
    String srch=jTextField1.getText();
    try {
        String sql="SELECT * FROM book WHERE Name LIKE '%"+srch+"%' OR Type LIKE '%"+srch+"%' OR Author LIKE '%"+srch+"%' OR availability LIKE '%"+srch+"%'";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        jTable1.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        //JOptionPane.showMessageDialog(null, e);
    }
}
```

```
 void addBookNotification(){
     String name=jTextField2.getText();
     String sql, date, notice;
     LocalDate todaysDate = LocalDate.now();

     notice="A new book is added!";

     try {
         sql="INSERT INTO notice(Date, Notice) VALUES ('"+todaysDate+"', '"+notice+"')";

         pst=conn.prepareStatement(sql);
         pst.execute();
         noticeBoard();
         jLabel8.setText("New book is added!");
         greenColor(jPanel8);

     } catch (Exception e) { //JOptionPane.showMessageDialog(null, e);
     }

 }
```

```
 void accountRemoveNotification(){
     int r=jTable2.getSelectedRow();
     String name=jTable2.getValueAt(r, 0).toString();
     String id=jLabel16.getText();
     String sql, date, notice;
     LocalDate todaysDate = LocalDate.now();
     notice=id+" has requested to remove account!";

     try {
         sql="INSERT INTO notice(Date, Notice) VALUES ('"+todaysDate+"', '"+notice+"')";

         pst=conn.prepareStatement(sql);
         pst.execute();
         noticeBoard();
         jLabel8.setText("Your account is being removed");
         redColor(jPanel8);

     } catch (Exception e) { //JOptionPane.showMessageDialog(null, e);
     }
 }
```

```
void bookUpdate(){
    String name, sql, author,type, availability, pages, index;
    name=jTextField2.getText();
    type=jTextField5.getText();
    author=jTextField3.getText();
    pages=jTextField4.getText();
    availability=jTextField6.getText();
    index=jLabel35.getText();

    try {
        sql="UPDATE book SET Name='"+name+"', Type='"+type+"', Author='"+author+"', Pages='"+pages+"', Availability='"+availability+"' WHERE IndexNumber='"+index+"' ";

        pst=conn.prepareStatement(sql);
        pst.execute();
        book();
        jTextField2.setText("");
        jTextField5.setText("");
        jTextField3.setText("");
        jTextField4.setText("");
        jTextField6.setText("");

        //JOptionPane.showMessageDialog(null, "changed!");
       jLabel8.setText("Information changed!");
       orangeColor(jPanel8);
    } catch (Exception e) { //JOptionPane.showMessageDialog(null, e);
    }
}
```

```
void bookRemove(){
    String sql, id, name;
    name=jTextField2.getText();

    int r=jTable1.getSelectedRow();
    String index=jTable1.getValueAt(r, 5).toString();

    try {
        sql="DELETE FROM book WHERE IndexNumber='"+index+"'";
        pst=conn.prepareStatement(sql);
        pst.execute();
        book();

        jTextField2.setText("");
        jTextField5.setText("");
        jTextField3.setText("");
        jTextField4.setText("");
        jTextField6.setText("");

        jLabel8.setText("Removed!");
        redColor(jPanel8);
    } catch (Exception e) {
    }
}
```

```
void member(){
    try {
        String sql="SELECT ID, Name, Email, PhoneNumber, Address, Course, Birthday, IndexNumber FROM member ORDER BY IndexNumber DESC";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        jTable2.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        NewJFrame4 cf=new NewJFrame4();
        //cf.setVisible(true);
    }
}
```

```
public void addMember(){
    String id, name, sql, address, birthday, password, course, pn, email;
    id=jTextField18.getText();
    name=jTextField21.getText();
    address=jTextField20.getText();
    birthday=jTextField24.getText();
    password=jTextField25.getText();
    course=jTextField23.getText();
    pn=jTextField22.getText();
    email=jTextField19.getText();

    try {
        sql="INSERT INTO member(Name, Address, Email, Password, PhoneNumber, Course, ID, Birthday) VALUES "
                + "('"+name+"', '"+address+"', '"+email+"', '"+password+"', '"+pn+"', '"+course+"', '"+id+"', '"+birthday+"')";

        pst=conn.prepareStatement(sql);
        pst.execute();
       // member();
       member();
        jTextField19.setText("");
        jTextField15.setText("");
        jTextField20.setText("");
        jTextField21.setText("");
        jTextField25.setText("");
        jTextField22.setText("");
        jTextField23.setText("");
        jTextField24.setText("");

        jLabel8.setText("New member is added!");
        greenColor(jPanel8);

    } catch (Exception e) { //JOptionPane.showMessageDialog(null, e);
    }
}
```

```
public void memberSearch(){
    String srch=jTextField7.getText();
    try {
        String sql="SELECT * FROM member "
                + "WHERE Name LIKE '%"+srch+"%' OR Address LIKE '%"+srch+"%' OR Email LIKE '%"+srch+"%' OR Course LIKE "
                + "'%"+srch+"%' OR Birthday LIKE '%"+srch+"%' OR IndexNumber LIKE '%"+srch+"%' OR PhoneNumber LIKE '%"+srch+"%' ORDER BY IndexNumber DESC     ";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        jTable2.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        //JOptionPane.showMessageDialog(null, e);
    }
}
```

```
public void memberTable5(){

    int r=jTable2.getSelectedRow();
    String index=jTable2.getValueAt(r, 7).toString();

    try {
        String sql="SELECT * FROM member WHERE IndexNumber='"+index+"'   ";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        jTable5.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        NewJFrame4 cf=new NewJFrame4();
       //cf.setVisible(true);
    }
}
```

```
void memberForAdmin(){
    try {
        String sql="SELECT ID, Name, Email, PhoneNumber, Address, Course, Birthday, IndexNumber FROM member ORDER BY IndexNumber DESC";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        jTable2.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        NewJFrame4 cf=new NewJFrame4();
       cf.setVisible(true);
    }
 }
```

```
void memberUpdate(){
    String id, name, sql, address, birthday, course, pn, email, index;
    id=jTextField8.getText();
    name=jTextField11.getText();
    address=jTextField10.getText();
    birthday=jTextField17.getText();
    //password=jTextField18.getText();
    course=jTextField16.getText();
    pn=jTextField12.getText();
    email=jTextField9.getText();
    index=jLabel38.getText();

    try {
        sql="UPDATE member SET ID='"+id+"', Name='"+name+"', Address='"+address+"', Birthday='"+birthday+"',"
                + " Course='"+course+"', PhoneNumber='"+pn+"', Email='"+email+"' WHERE IndexNumber='"+index+"'";

        pst=conn.prepareStatement(sql);
        pst.execute();
        member();
        adminUpdate();
        admin();
        jTextField8.setText("");
        jTextField11.setText("");
        jTextField10.setText("");
        jTextField17.setText("");
        //jTextField18.setText("");
        jTextField16.setText("");
        jTextField12.setText("");
        jTextField9.setText("");

        //JOptionPane.showMessageDialog(null, "changed!");
       jLabel8.setText("Information changed!");
       orangeColor(jPanel8);
    } catch (Exception e) { //JOptionPane.showMessageDialog(null, e);
    }

}
```

```
void memberRemove(){
    String sql, id, name;
    //index=jLabel38.getText();
    id=jTextField8.getText();
    name=jTextField11.getText();

    int r=jTable2.getSelectedRow();
    String index=jTable2.getValueAt(r, 7).toString();

    try {
        sql="DELETE FROM member WHERE IndexNumber='"+index+"'";
        pst=conn.prepareStatement(sql);
        pst.execute();
        member();

        jLabel8.setText("Removed!");
        redColor(jPanel8);

    } catch (Exception e) {//JOptionPane.showMessageDialog(null, e);
    }

}
```

```
void memberToAdmin(){
    String sql;

    int r=0; //jTable5.getSelectedRow();
    String id=jTable5.getValueAt(r, 0).toString();
    String name=jTable5.getValueAt(r, 1).toString();
    String email=jTable5.getValueAt(r, 2).toString();
    String pn=jTable5.getValueAt(r, 3).toString();
    String address=jTable5.getValueAt(r, 4).toString();
    String course=jTable5.getValueAt(r, 5).toString();
    String birthday=jTable5.getValueAt(r, 6).toString();
    String password=jTable5.getValueAt(r, 7).toString();
    String index=jTable5.getValueAt(r, 8).toString();

    try {
        sql="INSERT INTO admin(Name, Email, Password, IndexNumber) VALUES ('"+name+"', '"+email+"', '"+password+"', '"+index+"') ";

        pst=conn.prepareStatement(sql);
        pst.execute();
        admin();
       member();

        jLabel8.setText(id+ " " + name + " is an admin now!");
        greenColor(jPanel8);

    } catch (Exception e) { JOptionPane.showMessageDialog(null, "Already added!");
    }

}
```

```
void admin(){
    try {
        String sql="SELECT Name, Email, IndexNumber FROM admin ORDER BY IndexNumber DESC";
        //String sql="SELECT * FROM admin ORDER BY IndexNumber DESC";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        jTable4.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
       // JOptionPane.showMessageDialog(null, "Admin Error!");
    }
}
```

```
void adminUpdate(){
   String name, sql, email, index;
   name=jTextField11.getText();
   //password=jTextField18.getText();
   email=jTextField9.getText();
   index=jLabel38.getText();

   try {
       sql="UPDATE admin SET Name='"+name+"', Email='"+email+"' WHERE IndexNumber='"+index+"'";

       pst=conn.prepareStatement(sql);
       pst.execute();
       admin();
       jLabel8.setText("Information changed!");
       orangeColor(jPanel8);
       //JOptionPane.showMessageDialog(null, "changed!");
   } catch (Exception e) { JOptionPane.showMessageDialog(null, e);
   }

}
```

```
void adminRemove(){
    String sql;
    int r=jTable4.getSelectedRow();
    String index=jTable4.getValueAt(r, 2).toString();
    //String id=jTable2.getValueAt(r, 0).toString();
    //String name=jTable4.getValueAt(r, 0).toString();

    sql="DELETE FROM admin WHERE IndexNumber='"+index+"'";
    try {
        pst=conn.prepareStatement(sql);
        pst.execute();
        admin();
        jLabel8.setText("Removed from admin list!");
        redColor(jPanel8);
    } catch (Exception e) {JOptionPane.showMessageDialog(null, e);
    }
}
```

```
void noticeBoard(){
    try {
        String sql="SELECT * FROM notice ORDER BY IndexNumber DESC";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        jTable3.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        NewJFrame4 cf=new NewJFrame4();
        cf.setVisible(true);
    }
}
```

```
void noticeRemove(){
    int r=jTable3.getSelectedRow();
    String index=jTable3.getValueAt(r, 2).toString();

    try {
        String sql="DELETE FROM notice WHERE IndexNumber='"+index+"'";
        pst=conn.prepareStatement(sql);
        pst.execute();
        noticeBoard();

        jLabel8.setText("Removed!");
        redColor(jPanel8);
    } catch (Exception e) {
    }
}
```

```
public void report(){
    String srch=jTextField1.getText();
    try {
        String sql="SELECT ID, Book, Date, Duration, Returned, IndexNumber FROM Report ORDER BY IndexNumber DESC";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        jTable6.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        //JOptionPane.showMessageDialog(null, e);
    }
}
```

```
public void updateReport(){
    int r=jTable6.getSelectedRow();
    String index=jTable6.getValueAt(r, 5).toString();
    String complete = jComboBox1.getSelectedItem().toString();

    try {
        String sql="UPDATE Report SET Returned='"+complete+"' WHERE IndexNumber='"+index+"'";

        pst=conn.prepareStatement(sql);
        pst.execute();
        report();
        jLabel8.setText("Information changed!");
        orangeColor(jPanel8);
        //JOptionPane.showMessageDialog(null, "changed!");
    } catch (Exception e) { //JOptionPane.showMessageDialog(null, e);
    }
}
```

Task 03

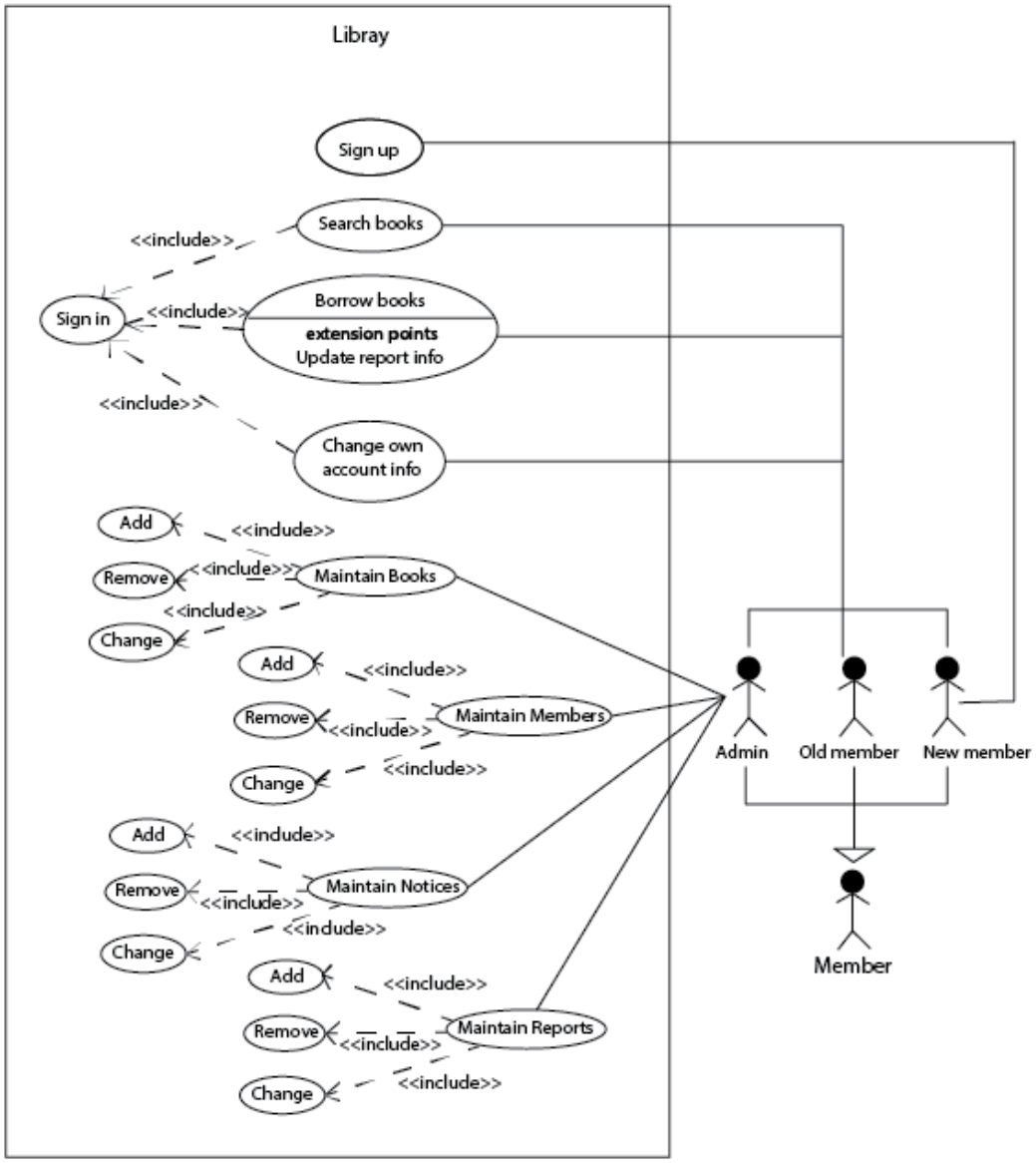
a.

Unified Modeling Language is a general-purpose modeling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite like blueprints used in other field of engineering.

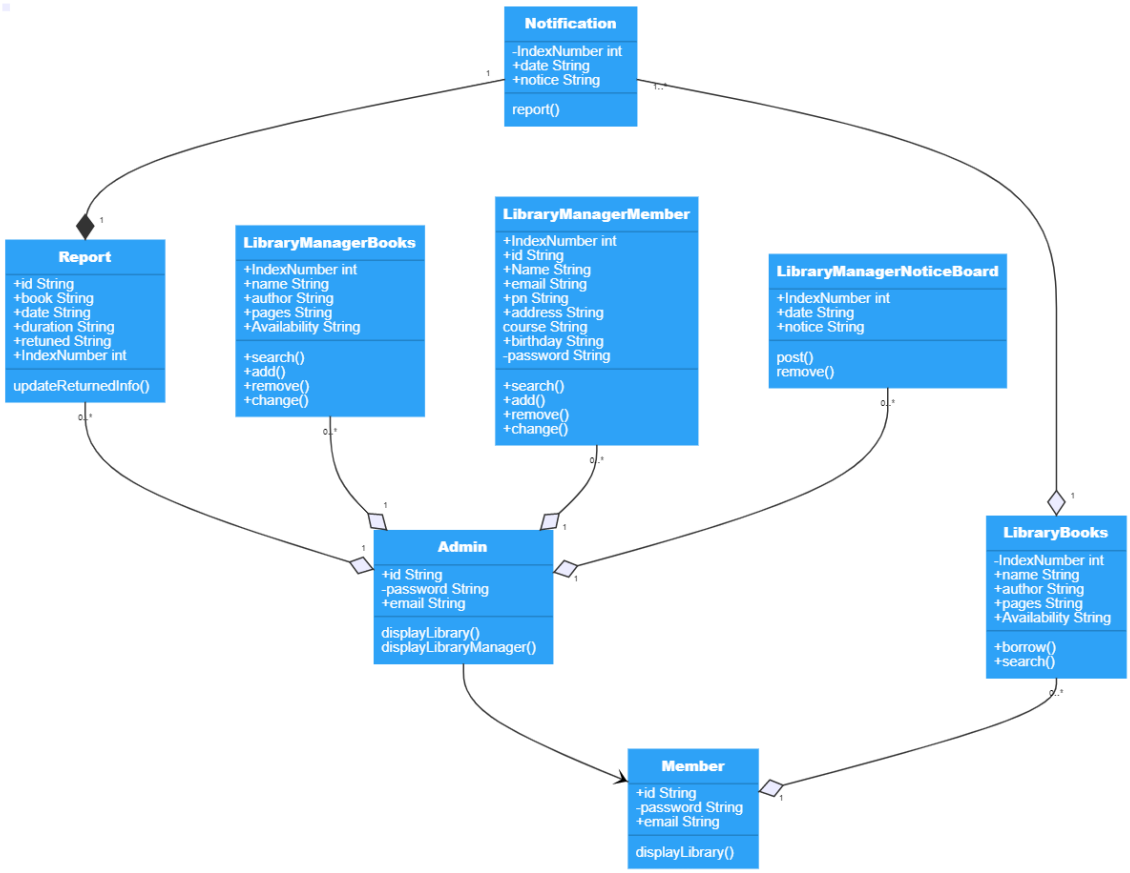
UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modeling, design and analysis.

b.

Use case diagram



Class diagram



c.

Use case diagram

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Class diagram

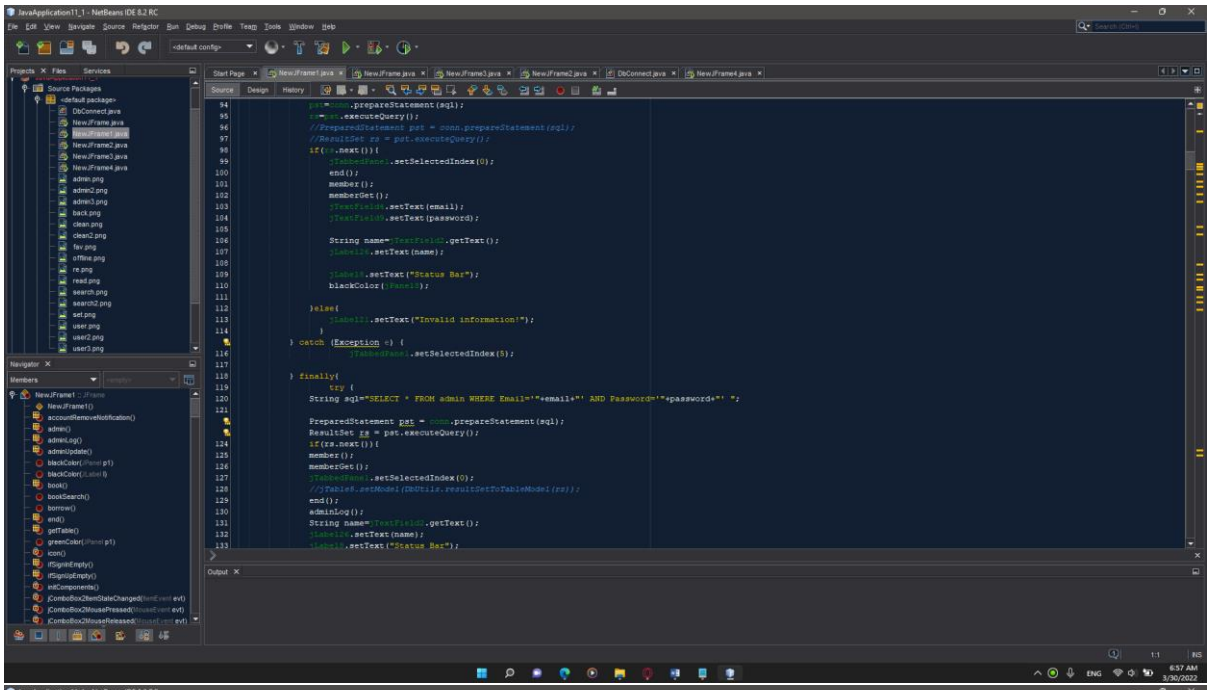
- Visibility
- Scope
- Abstract, Root, Leaf and Polymorphic elements

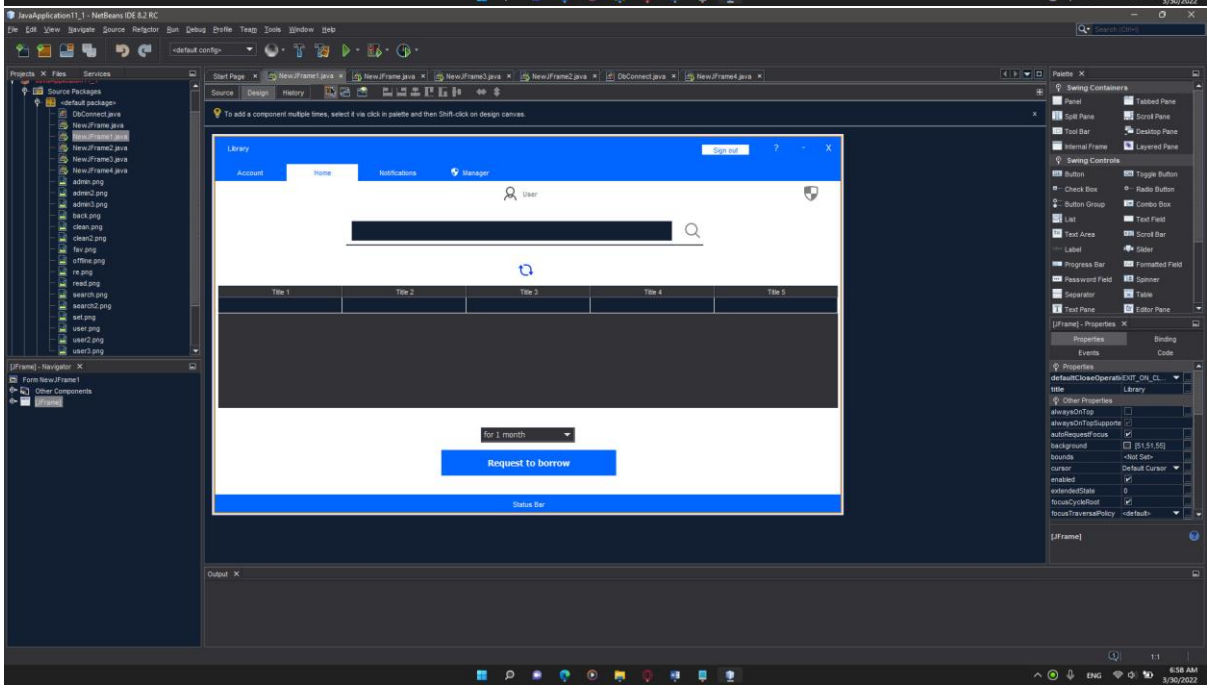
- Multiplicity
- Attributes
- Operations

Task 04

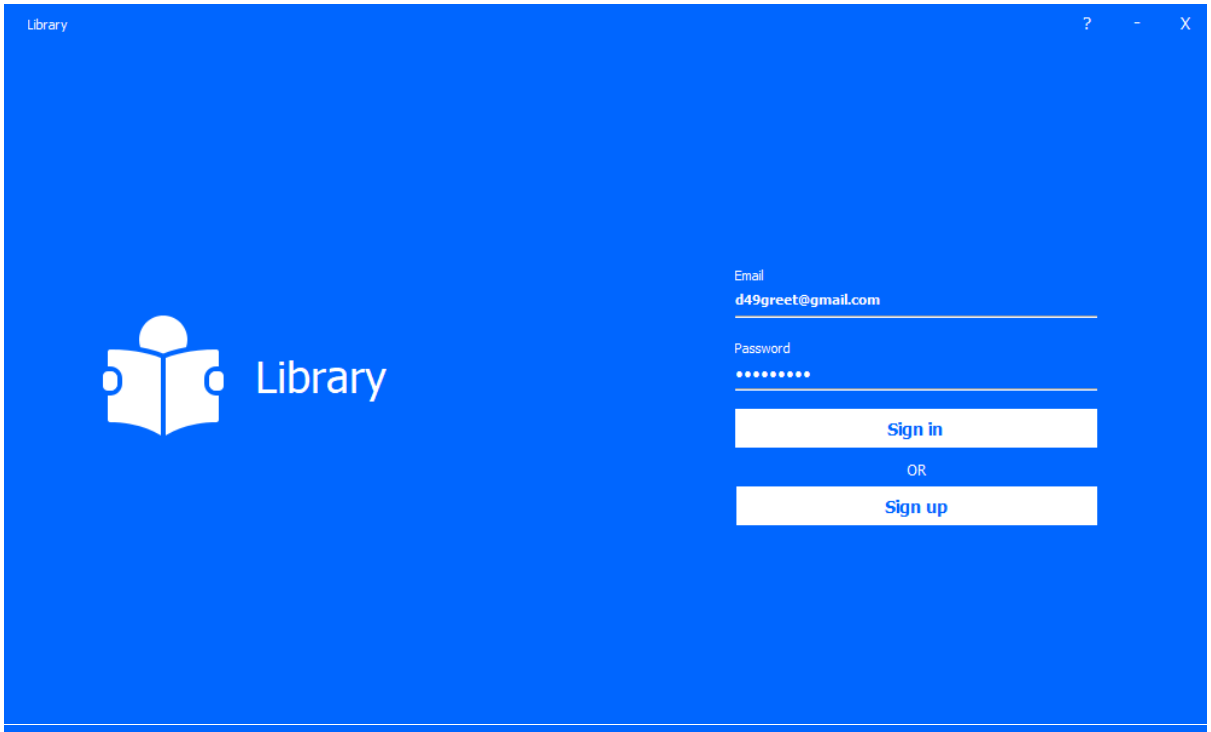
a.

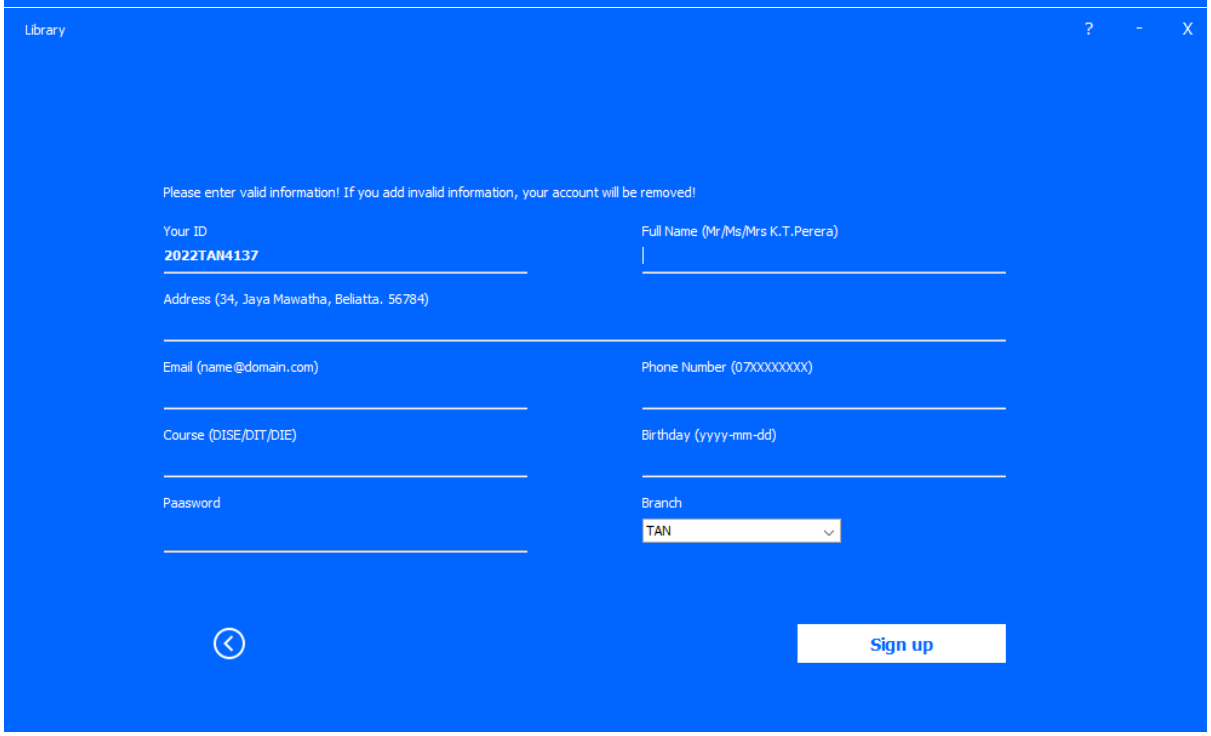
NetBeans IDE

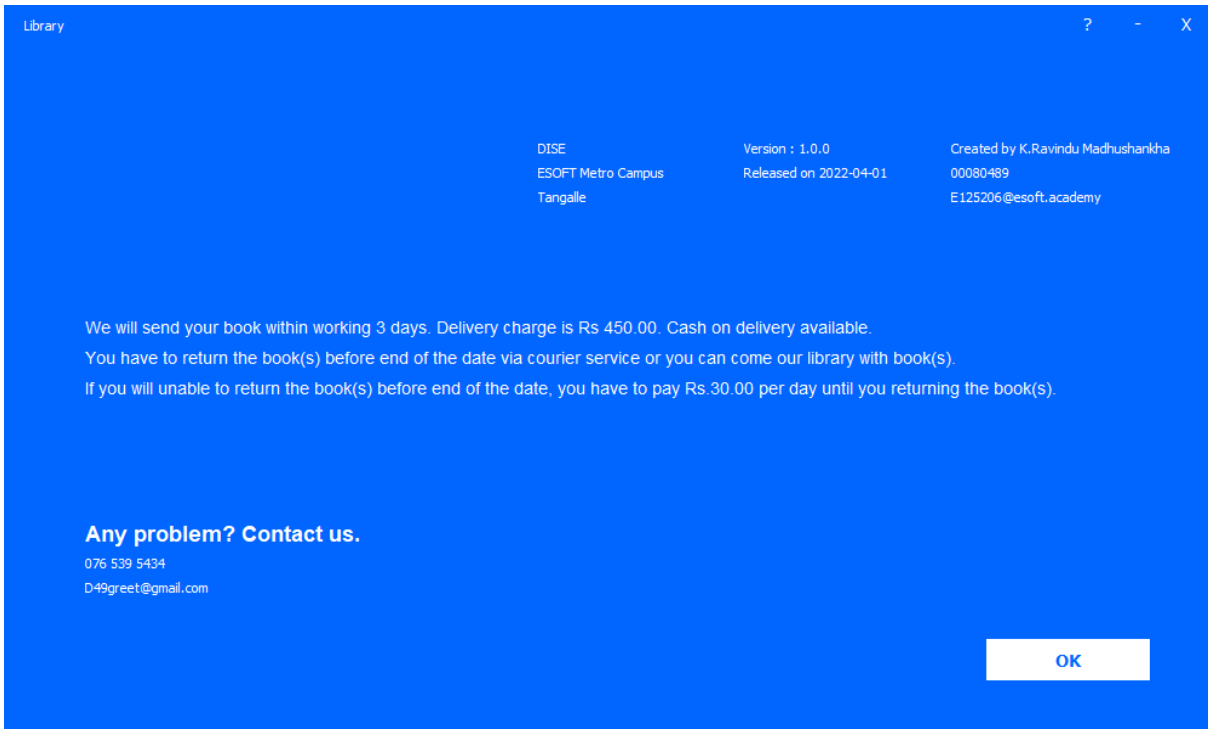




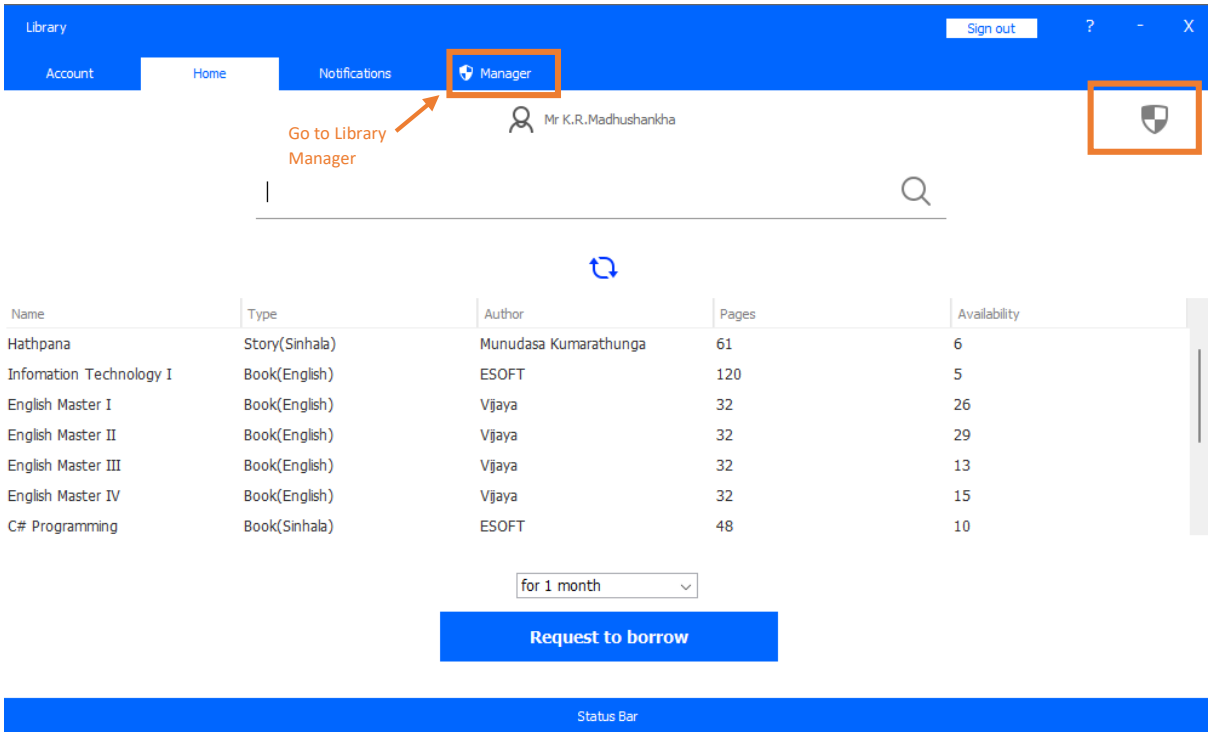
JAVA Frames

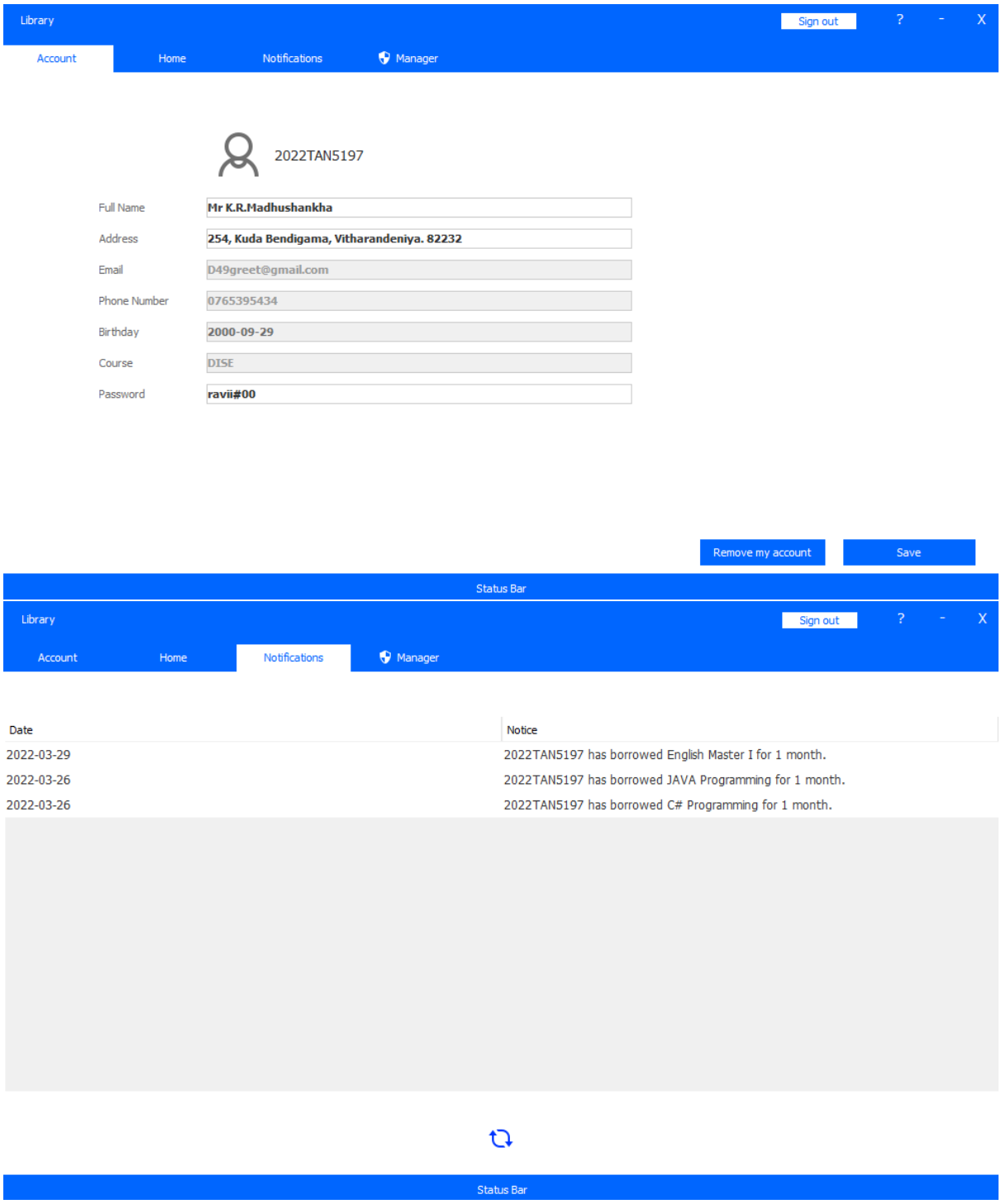

Library
Library
Email
d49greet@gmail.com
Password
Sign in
OR
Sign up


Library
Please enter valid information! If you add invalid information, your account will be removed!
Your ID
2022TAN4137
Full Name (Mr/Ms/Mrs K.T.Perera)
Address (34, Jaya Mawatha, Beliatta. 56784)
Email (name@domain.com)
Phone Number (07XXXXXXXX)
Course (DISE/DIT/DIE)
Birthday (yyyy-mm-dd)
Paasword
Branch
TAN
Sign up



Only admins can see these labels.




Library
Sign out
?
-
X
Account
Home
Notifications
Manager
2022TAN5197
Full Name
Mr K.R.Madhushankha
Address
254, Kuda Bendigama, Vitharandeniya. 82232
Email
D49greet@gmail.com
Phone Number
0765395434
Birthday
2000-09-29
Course
DISE
Password
ravii#00
Remove my account
Save
Status Bar
Library
Sign out
?
-
X
Account
Home
Notifications
Manager
Date
Notice
2022-03-29
2022TAN5197 has borrowed English Master I for 1 month.
2022-03-26
2022TAN5197 has borrowed JAVA Programming for 1 month.
2022-03-26
2022TAN5197 has borrowed C# Programming for 1 month.
Status Bar

Library Manager

Members | Books | Admin

Name	Type	Author	Pages	Availability	IndexNumber
Pariganaka 2022-02	Magazine(Sinhala)	Vijaya	40	2	43
JAVA Programming	Book(Sinhala)	ESOFT	54	10	42
Infomation Technology II	Book(English)	ESOFT	96	4	39
Madoldoowa	Novel(Sinhala)	Martin Wickramasinge	441	4	36
Python Programming	Book(Sinhala)	ESOFT	68	10	27
C# Programming	Book(Sinhala)	ESOFT	48	10	26
English Master IV	Book(English)	Vijaya	32	15	17

Book Name: **Python Programming**
Type: **Book(Sinhala)**
Author: **ESOFT**
Availability: **10**
Pages: **68**

Clean all the Text fields

Remove | Save | Add

Status Bar

Library Manager

Members | Books | Admin

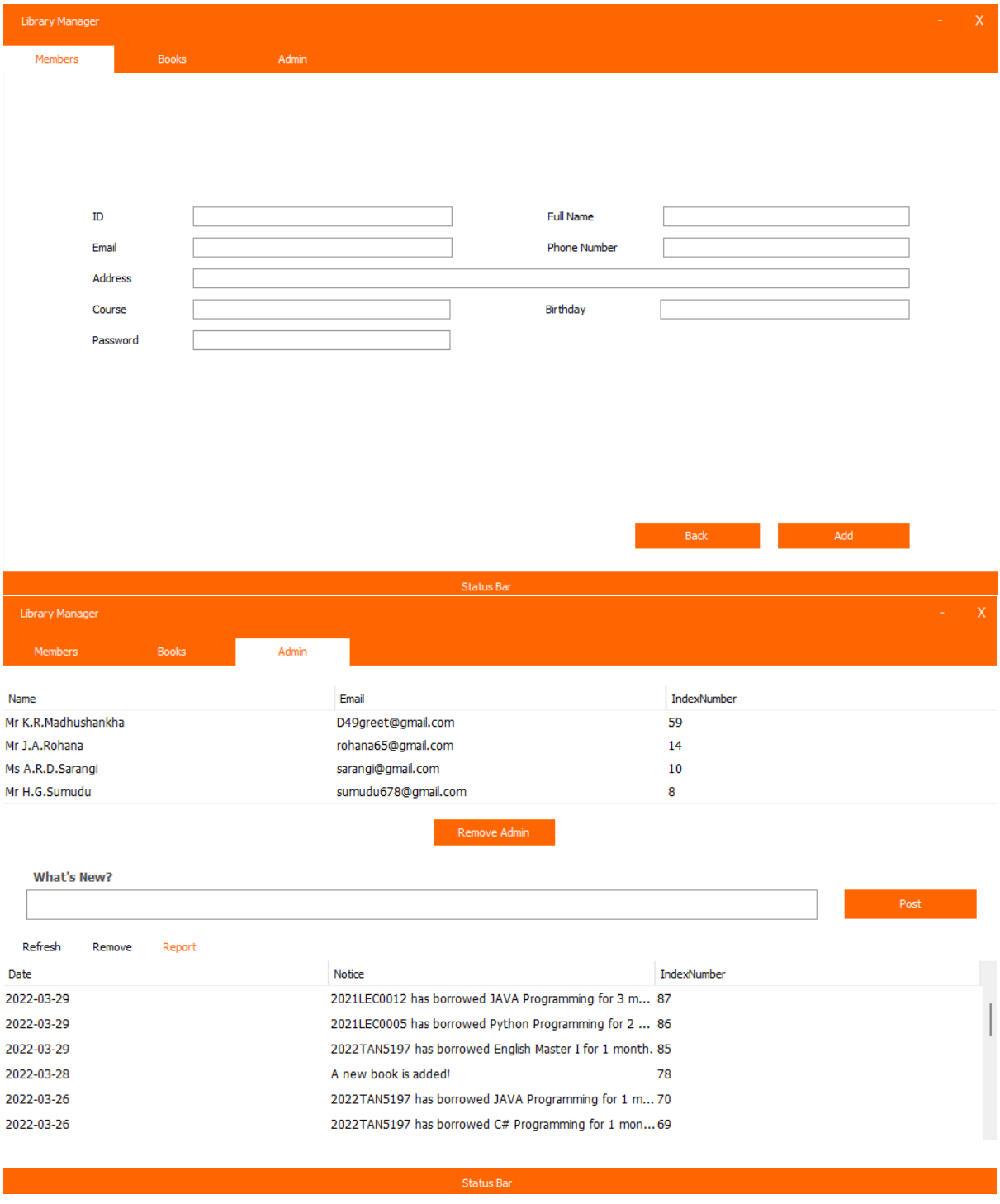
ID	Name	Email	PhoneNumber	Address	Course	Birthday	IndexNumber
2022GAL7375	G.T.Aariyadasa	Aar@yahoo.com	0787898543	45, Maradana, Colo...	DISE	2000-08-05	60
2022TAN5197	Mr K.R.Madhushankha	D49greet@gmail.com	0765395434	254, Kuda Bendiga...	DISE	2000-09-29	59
2022GAL3639	Mr M.T.Rathnayake	rathna34@yahoo.com	0764576345	88, Mahapola mawa...	DIE	1998-09-04	43
2022TAN0002	Mr H.G.Sumudu	sumudu@gmail.com	0764565346	87/B, Kotuwa, Colo...	DIE	1999-11-08	24
2021LEC0012	Mr J.A.Rohana	rohana65@gmail.com	0764567864	78, Borella Road, K...	DISE	1974-04-19	14
2021LEC0005	Ms A.R.D.Sarangi	sarangi@gmail.com	0716798324	55, Borella Road, K...	DIT	1992-04-19	10
2022TAN0009	Ms K.M.Madhushani	madhushani349@g...	0705634278	45, Kottawa, Colo...	DISE	2000-11-14	9

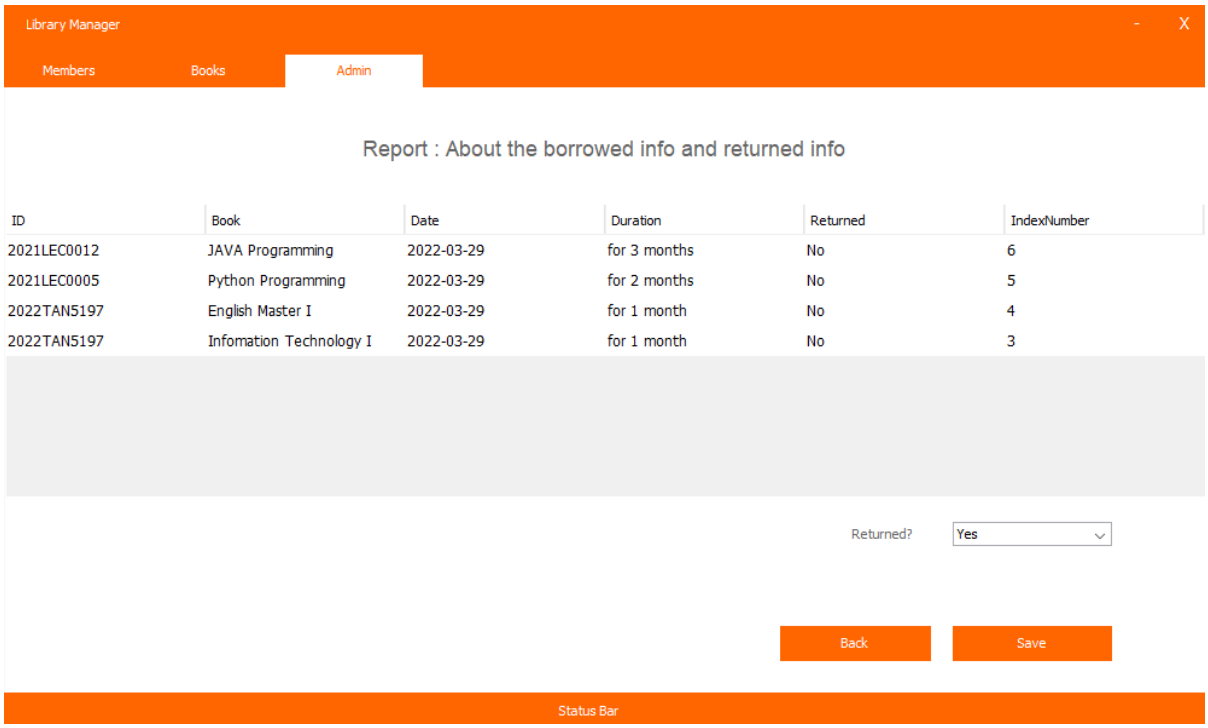
ID: **2021LEC0005**
Full Name: **Ms A.R.D.Sarangi**
Email: **sarangi@gmail.com**
Phone Number: **0716798324**
Address: **55, Borella Road, Kahawatta. 56784**
Course: **DIT**
Bithday: **1992-04-19**

Make as Admin | Remove | Save | Add a member

Status Bar

Add a new member form in member tab.





b.

Facilities in NetBeans IDE that was important while the development process.

- Project management
- Visual debugger
- Static analysis tools
- Code converters
- NetBeans Profiler
- Batch code analyzers
- Cross-platform support
- Multiple language support

Advantages of using NetBeans IDE.

- Supports multiple languages which help in coding for developers.
- Support cross-platform like Mac, Ubuntu, Windows etc.
- Rich set of community provided plugins.
- Easy & efficient project management.
- Bug free formatted code.

c.

Testing

- The project software was analyzed.
- Define scope of testing, identify testing type, document risk & issues and create test logistics were done.
- Functionality, performance, GUI… etc. were listed for testing the system and they were done.
- In resource planning, human resource and system resource were detailed for testing and they were tested.
- Test environment was done.

After that **Unit testing** method was chosen to test my project.

White box test plan

- Testing types – Following testing types will be conducted to carry out the activities of white box test plan.
- Code coverage – Source code of the developed library management system will be run and al the code lines are verified to ensure that there are not errors present in the developed code.
- Segment coverage – Segments of code are executed in this type of testing in order to ensure that code's performance is adequate.
- Branch coverage – Every branch of code is executed in this type of testing.
- Compound condition coverage – Following coverage of conditions shall be done:
- Display home screen of the proposed LMS only to the authenticated users who enter valid login credentials or else display an appropriate error message.
- Provide loan data of the student on entering correct member ID or else display an appropriate error message.
- Allows members to register in the system if the entered details generate unique ID or else ask members to enter new details.
- Data Flow Testing – In this testing, validation of all possible values in all the fields of the system need to be validated.
- Loop testing – All loops are tested i.e. single, concatenated and nested.

Black Box Test Plan

- Testing Types - Following testing types will be conducted to carry out the activities of black box test plan.
- Security testing – This type of testing ensures that system satisfies the integration of security tools and controls in the proposed system.
- Integration testing – This type of testing integrates modules and test the performance of these integrated modules as a whole.

- System testing – This type of testing includes testing of functional aspects of the proposed application. All the modules and requirements of the system are required to be tested.
- Regression testing – Testing team will log various types of bugs in the tool. The development team shall resolve those bugs. Then testing team again tests those resolved bugs to ensure that defects are correctly so that it does not beak code of any other part of the developed system.

The **Test case** was created. Then checked cases which are following below and fixed errors,

- Sign in
- Sign up
- Change their own details
- Make admins to maintain the system
- Add books
- Add members
- Add notices
- Remove books
- Remove members
- Remove notices
- Borrow books
- Search books
- Search members
- Update returned book details
- Show availability of books
- Show notifications
- And all the data of tables like members, books, admins, notifications, reports…etc.

Importance of the testing

- Increase the quality
- Reduce risks
- Cost Effective
- Gain customer confidence

Finally, the significance of software testing cannot be overstated. Because it increases consistency and performance, software testing is an important part of the software development process. The fundamental advantage of testing is that it identifies and eliminates errors. Testing, on the other hand, aids developers and testers in comparing real and predicted outcomes to improve quality. If software is developed without being thoroughly tested, it may be ineffective or even dangerous to clients. As a result, a tester should wear a special helmet that safeguards the software's reliability and makes it safe to use in real-world settings.

Task 05

a.

The difficulties that I faced during the development process

- That was an individual project. All the documents and coding & designing were done myself.
- Scope, definition, tasks, schedule, cost, quality, organization, staff, communications, risk, procurement was planned and executed. Then they were managed.
- Time duration was 4 months.
- The cost was gone to online hosting server and Ethernet connection. Then I have to pay again for hosting year by year.
- There were some risks. They were managed in risk management task.
- Human resource were collected and checked.
- Procurement was done.
- Stakeholder was me and then all the activities have been checked.

b.

Web design techniques for my Library system

- An online host should be bought for maintained all the data of tables.
- It is should be a mobile responsive site.
- The images should be responsive images.
- Minimize flash and animations
- Use CSS media queries
- Use Javascript media queries
- Fast hosting sever should be used.
- Data tables should be mobile responsive.
- Add an online checkout session.

c.

Future development suggestions

- I decided to create a website for this Library Management system project.
- UI/UX design should be increased. The design can be increased by using FIGMA.
- Short professional domain name and fast hosting server should be bought.
- Books, Magazines, Newspapers will be added to read in future as PDF files. And those PDF files can be downloaded also.
- All the updated setup files with new feature could be downloaded from internet.
- Feedback session will be added.
- Contact session will be added without calling and emailing manually.

ONL00080489 K.Ravindu Madhushankha
DISE | Tangalle Branch