VIT®

UNIVERSITY

(Estd. u/s 3 of UGC Act 1956)

VELLORE ■ CHENNAI

www.vit.ac.in

Library Database Management System

By

Pranshu Pranjal

16BIT0044

Under the guidance of

Prof Brijendra Singh

For

ITE1003: Database Management Systems

L25+26

Acknowledgment

In the accomplishment of this project completing successfully, many people have best owned upon me their blessings and the heart pledged support, this time I am utilizing to thank all the people who have been concerned with project.

I would like to thank Prof Brijendra Singh, whose valuable guidance has been the ones that helped me patch this project and make it full proof success. His suggestions and his instructions have served as the major contributor towards the completion of the project.

Then I would like to thank my friends who have helped me with their valuable suggestions and guidance that has been very helpful in various phases of completion of this project.

Introduction

A library management system is an enterprise resource planning system for a library, used to track items owned, orders made, bills paid, and patrons who have borrowed.

It usually comprises a relational database, software to interact with that database, and one graphical user interfaces. Most library management system separate software functions into discrete programs called modules, each of them integrated with a unified interface. Examples of modules might include:

- acquisitions (ordering, receiving, and invoicing materials)
- cataloguing (classifying and indexing materials)
- circulation (lending materials to patrons and receiving them back)
- serials (tracking magazine and newspaper holdings)

The Database Library System is intended to automate the library activities such as creating a new borrower, giving books to the borrowers, maintaining the details of all the item that were available in the books . This also helps the librarians by providing information such as total copies available each book, list of books that belong to a particular category (Short, Long Loan, Reference items, etc).

Each patron and item has a unique ID in the database that allows the library management system to track its activity.

The Library Database System that I will be designing will be for school level infrastructure and will be made using MySQL (back end) and Netbeans Java application (front end).

It will contain basic feature like borrowing and returning books, author and publisher details, administrator and employee access, book details and student details.

Entities

1) Author
2) Bill
3) Book
4) Employee
5) Login Info
6) Publisher
7) Security
8) Student

Detailed description of them:

Books

(This is the master table for all the books that are available in the Library)

a) Accession No (Primary Key)

- *This is unique ID given to every book. As there could be a large no. of books with same TITLE, this Accession no. will help us to distinguish between books of same title.*

b) Name of Book

- Provides the name of the book.

c) Author no (Foreign Key)

- *Gives details of author of that particular book.*

d) Co Author

- *Gives details of other authors who worked with main author to write the book. (if any.*

e) Publisher No (Foreign Key)

- *Gives details of publisher of that particular book.*

f) Publish year

- *Contains the year of publication.*

g) Pages

- *Contains the number of pages in the book.*

h) Genre (Foreign Key)

- *Links to the Genre table in Database.*

i) Subject

- *If it is a course reference book then this will help in categorising the books.*

j) Bill ID (Foreign Key)

- *Links to the bill table where purchasing information is stored.*

b) Number Unit

- *Number of same books available in Library*

c) Remark

- *Any specific comment like in repair or damaged*

Students

(This is the master table for all the students enrolled in the school)

a) Admission Number (Primary Key)

- *Unique number to identify each and every student in the school.*

b) Name

c) Gender

d) Class

e) Section

f) Father's Name

g) Mother's Name

h) Address

i) Mobile No

Borrowed Books

(This table provides basic functionality in a library)

a) Accession Number (Primary Key)

- *This is a particular unique number generated for each book for faster searching.*

b) Admission No (Foreign Key)

- *To identify the student who is borrowing.*

c) Issued Date

- *Gives the date of issue of book.*

d) Time Period

- *To be able to identify the date by which the book has to be returned.*

e) Return Date

- *Actual date on which book was returned.*

Authorized Personal (Employee)

(This is the master table for all the staff of the library)

a) Employee ID (Primary Key)

- The unique ID given to each staff member present in the Library.

b) Name

c) Email

d) Gender

e) Type

- *Tells whether the employee administrator or normal employee.*

f) Address

g) Mobile Number

h) Alternate Contact Number

Author

(This is the master table for all the authors whose books are present in library)

a) Author_ no (Primary Key)

- *Unique ID given to each Author*

b) First Name

c) Middle Name

d) Last Name

e) Pen Name

- *If any pseudo name was used to write books instead of real name*

Bill

(It contains the purchasing information of the book like its cost etc.)

a) Bill ID (Primary Key)
    a) *Unique number for each bill*
b) Accession No
    a) *G19PHY1,G19PHY2,G10MAT1 i.e. GenreNoSubjectBookNo*
c) Unit Cost
d) Discount
e) Seller
    a) *Book Store/ Retailer name from where book is purchased*
f) Total Cost
g) Date of purchase

Publisher

(This is the master table for all the publishers)

a) Publisher No (Primary Key)
    - *Unique ID given to each Publisher*
b) Publisher Name

Security

a) Employee ID (Primary Key)
b) User ID
    - *To use when logging in*

c) Password

d) Security Question (Foreign Key)

   - *If password is forgotten then can use security question and answer to log in.*

e) Security Answer

Security Question

a) Sq_id (Primary Key)

   - *Contains the unique number given to each question*

b) Question

Login Screen

a) User ID (Primary Key)

b) Punch IN

   - *Gives time of logging in*

Function Requirements

The Library DBMS which I will be making will be for use only by authorised users and not by customers. The Front end will start by showing login screen and then after proper authorization, it will take to Home Page.

The Front end will me made using Frames in Netbeans (Java Based) and interconnected using buttons. The back end will be made of MySQL and in turn be connected to the frontend using the connector module.

To go from one frame to another, menu driven concept would be utilised.

Employee constraint

The staff will be categorised into two category:

- Administrator
- Normal Employee

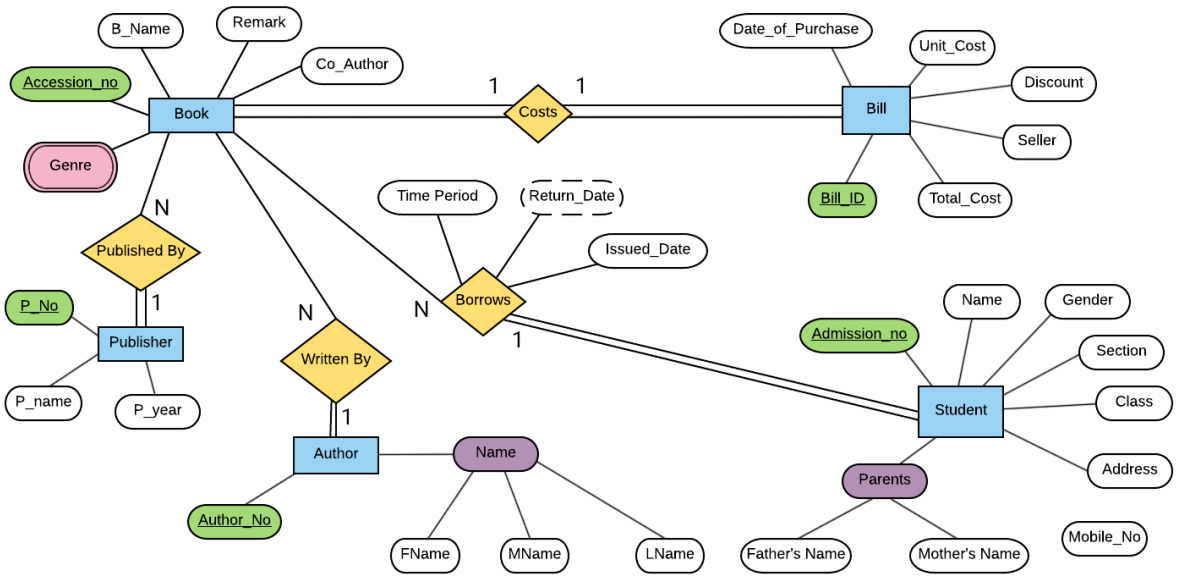
Administrator will have the power to add authors, publishers, new books (or delete them), modify student details whereas the normal employee can only provide facilities like borrowing or returning book, searching for a book in library etc.

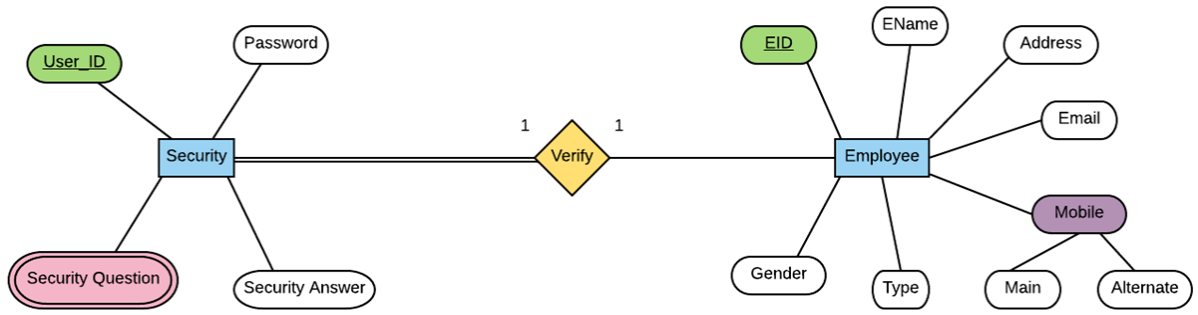
Constraints:

- Time period of issue will be in terms of weeks (7 days) only like 7,14,21,28.
- All search (author/ publisher/ book) frames will have reset buttons.
- Book search will be based on Accession Number or Book name only and should return the result as a table.

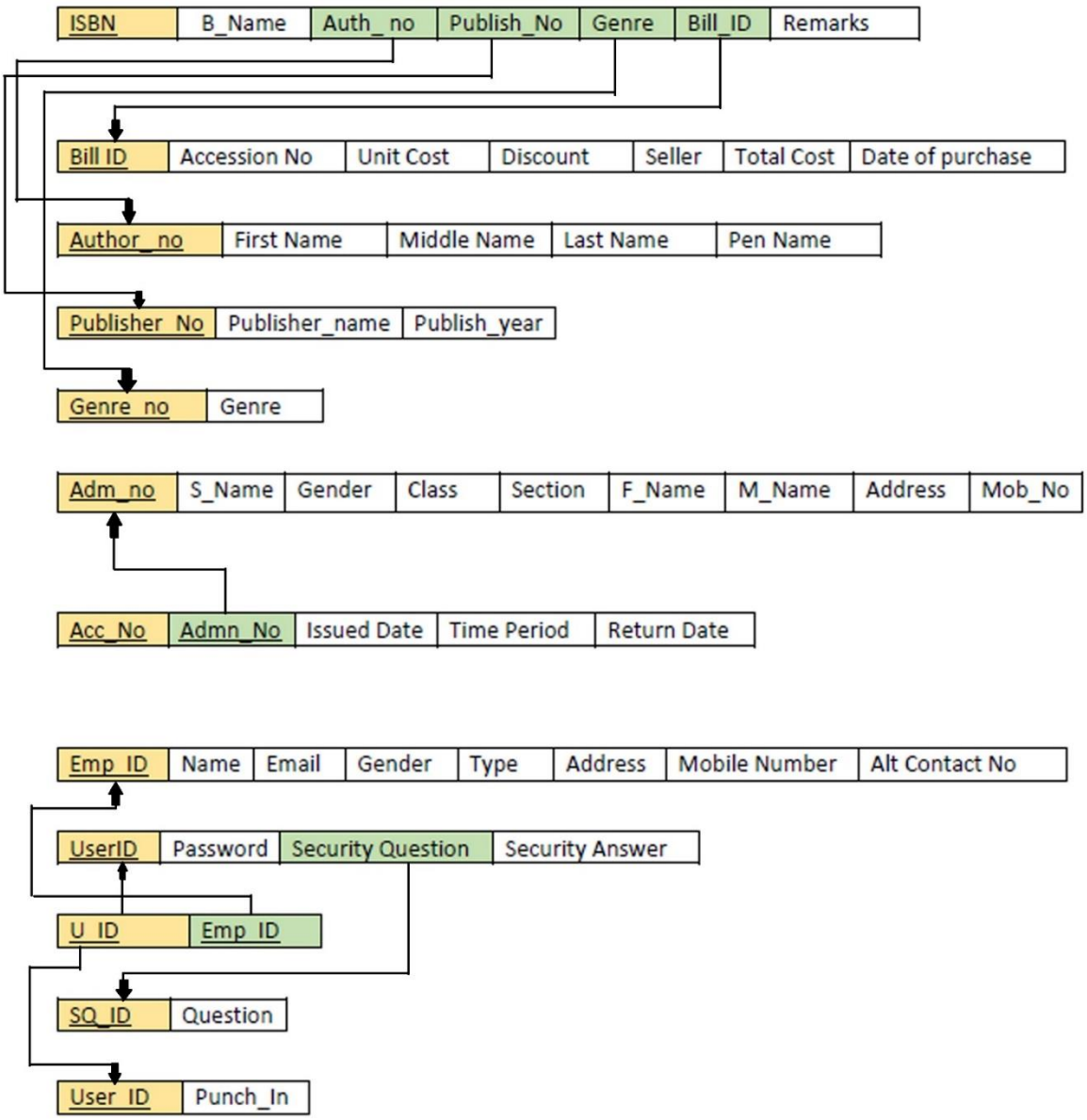
- Subject could be only entered if the genre is selected as “Text Book”.
- Search for student should be based on Name or Admission Number or class or section.
- Error should be reported if any exception occurs during any operation.
- Primary Key cannot be edited directly, it must be deleted first and then new one must be added.
- Book table should reference to author, publisher, bill, genre tables by their id using foreign keys.
- Entity relationship constraint must be maintained at all times.

ER schema


B_Name
Remark
Co_Author
Accession_no
Book
Genre
1
Costs
1
Date_of_Purchase
Unit_Cost
Discount
Seller
Bill
Bill_ID
Total_Cost
N
Published By
1
P_No
Publisher
P_name
P_year
Time Period
Return_Date
Issued_Date
Borrows
N
1
N
Written By
1
Author
Author_No
Name
FName
MName
LName
Admission_no
Name
Gender
Section
Class
Address
Student
Parents
Father's Name
Mother's Name
Mobile_No


User_ID
Password
Security
Security Question
Security Answer
1
Verify
1
EID
EName
Address
Email
Employee
Mobile
Gender
Type
Main
Alternate

Relation Schema



- For each regular (strong) entity type E in the ER schema, all the simple attributes of E are included.
- All the multi value attributes are broken into another table with primary key referencing to the foreign key in parent table.
- 1:N binary relation have been taken care of putting primary key of 1 part as foreign key of N part.
- 1:1 binary Relation have been taken care by putting primary key of partial relation as foreign key of total relation.

Functional Dependencies

FD1:

{ ISBN } → { B_Name, Auth_ no, Publh_No, Bill_No, Genre, Remarks }

FD2 :

{ Bill_ID } → {Accession No, Unit Cost, Discount, Seller, Total Cost, Date of purchase}

FD3:

{ Author_ no} → {First Name, Middle Name, Last Name, Pen Name }

FD4:

{ Publisher_no } → { Publisher_Name, Publishing_Year }

FD5:

{ Genre_no } → { Genre }

FD6:

{Adm_no} → { S_Name , Gender , Class, Section, F_Name, M_Name, Address, Mob_No }

FD7:

{ Acc_No , Admn_No } → { Issued Date, Time Period, Return Date }

FD8:

{ Emp_ID} → { Name, Email , Gender, Type, Address, Mobile_No, Alt Contact_No}

FD9:

{ UserID } → { Password, Security Question, Security Answer }

FD10:

{ SQ_ID } → { Question }

FD11:

{ User_ID } → { Punch In }

- All are in 1NF as they all are single attributes

  As multiple attributes were taken care by making a new Functional Dependency.
- All are in 2NF as they are no partial dependencies.
- All are in 3NF as they have no transitive dependencies.

  All possible transitive dependencies were already taken care of when making ER Schema.

Tables in MySQL

```
mysql> show tables;
+-------------------+
| Tables_in_project |
+-------------------+
| author            |
| bill              |
| book              |
| book_issued       |
| book_return       |
| employee          |
| genere            |
| login_info        |
| publisher         |
| security          |
| sques             |
| student           |
+-------------------+
12 rows in set (1.28 sec)
```

➢ Author Table

```
mysql> desc author;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| author_no   | int(5)      | NO   | PRI | NULL    |       |
| First_name  | varchar(50) | YES  |     | NULL    |       |
| middle_name | varchar(50) | YES  |     | NULL    |       |
| last_name   | varchar(50) | YES  |     | NULL    |       |
| Pen_name    | varchar(20) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.02 sec)
```

➢ Bill Table

```
mysql> desc bill;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| bill_id         | int(4)      | NO   | PRI | NULL    |       |
| Accesion_no     | int(5)      | YES  |     | NULL    |       |
| unit_cost       | int(5)      | YES  |     | NULL    |       |
| discount        | int(2)      | YES  |     | NULL    |       |
| Seller          | varchar(20) | YES  |     | NULL    |       |
| total_cost      | int(6)      | YES  |     | NULL    |       |
| Date_of_purchase | date       | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

➢ Book Table

```
mysql> desc book;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Accession_no | int(4)      | NO   | PRI | NULL    |       |
| Name_book    | varchar(70) | YES  |     | NULL    |       |
| Author_no    | int(5)      | YES  |     | NULL    |       |
| Co_Author    | varchar(60) | YES  |     | NULL    |       |
| Publisher_no | int(5)      | YES  |     | NULL    |       |
| publish_year | int(4)      | YES  |     | NULL    |       |
| Pages        | int(5)      | YES  |     | NULL    |       |
| Genre_no     | int(5)      | YES  |     | NULL    |       |
| Subject      | varchar(20) | YES  |     | NULL    |       |
| bill_id      | int(4)      | YES  |     | NULL    |       |
| Number_unit  | int(4)      | YES  |     | NULL    |       |
| remark       | varchar(90) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
12 rows in set (0.98 sec)
```

➢ Book Issued Table

```
mysql> desc book_issued;
+-----------------+---------+------+-----+---------+-------+
| Field           | Type    | Null | Key | Default | Extra |
+-----------------+---------+------+-----+---------+-------+
| Accesion_number | int(4)  | YES  |     | NULL    |       |
| Admission_no    | int(6)  | YES  |     | NULL    |       |
| issued_date     | date    | YES  |     | NULL    |       |
| time_period     | int(2)  | YES  |     | NULL    |       |
+-----------------+---------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

➢ Book Return Table

```
mysql> desc book_return;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| Accesion_number | int(4)      | NO   | PRI | NULL    |       |
| Name_book       | varchar(70) | YES  |     | NULL    |       |
| Admiission_no   | int(6)      | YES  |     | NULL    |       |
| return_date_    | date        | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
4 rows in set (0.02 sec)
```

➢ Employee table

```
mysql> desc employee;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| Employee_Id       | int(10)     | NO   | PRI | NULL    |       |
| Name              | varchar(20) | YES  |     | NULL    |       |
| Email             | varchar(30) | YES  |     | NULL    |       |
| Gender            | char(1)     | YES  |     | NULL    |       |
| type              | varchar(18) | YES  |     | NULL    |       |
| Address           | varchar(70) | YES  |     | NULL    |       |
| mobile_no         | varchar(12) | YES  |     | NULL    |       |
| alternate_mobileno| varchar(12) | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

➢ Genre Table

```
mysql> desc genere;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| genere_no | int(2)      | NO   | PRI | 0       |       |
| Genere    | varchar(20) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

➢ Publisher Table

```
mysql> desc publisher;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| Publisher_no   | int(5)      | NO   | PRI | NULL    |       |
| Publisher_name | varchar(20) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

➢ Security Table

```
mysql> desc security;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| Employee_Id       | int(10)     | NO   | PRI | NULL    |       |
| User_Id           | varchar(10) | YES  |     | NULL    |       |
| Password          | varchar(20) | YES  |     | NULL    |       |
| security_question | int(2)      | YES  |     | NULL    |       |
| security_answer   | varchar(34) | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
5 rows in set (0.03 sec)
```

➢ Security question

```
mysql> desc sques;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| sq_id    | int(2)      | NO   | PRI | NULL    |       |
| question | varchar(80) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

➢ STUDENT TABLE

```
mysql> desc student;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| admission_no | int(7)      | NO   | PRI | NULL    |       |
| Name         | varchar(48) | YES  |     | NULL    |       |
| class        | int(2)      | YES  |     | NULL    |       |
| section      | char(1)     | YES  |     | NULL    |       |
| gender       | char(1)     | YES  |     | NULL    |       |
| fathers_name | varchar(34) | YES  |     | NULL    |       |
| mothers_name | varchar(49) | YES  |     | NULL    |       |
| Address      | varchar(70) | YES  |     | NULL    |       |
| mobile_no    | varchar(12) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
9 rows in set (0.02 sec)
```

➢ LOGIN INFO TABLE

```
mysql> desc login_info;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| User_ID  | varchar(10) | YES  |     | NULL    |       |
| Punch_IN | varchar(20) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.03 sec)
```

NetBeans IDE

Menu

```
import java.awt.Toolkit;

import java.awt.event.WindowEvent;

public void close() {

        WindowEvent winClosingEvent = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);
Toolkit.getDefaultToolkit().getSystemEventQueue().postEven
t(winClosingEvent);

    }
```

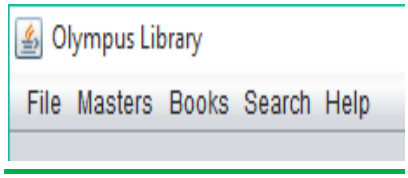
Menu Item Code:

```
close();

Home_Screen  s= new Home_Screen();

s.setVisible(true);
```

(The portion in italics was changed to corresponding jFrame name for each button)



Search Button

```
int clas=jComboBox1.getSelectedIndex();

String sectn=jComboBox2.getSelectedItem().toString();
```

```
DefaultTableModel model =(DefaultTableModel) jTable1.getModel();
int rows=model.getRowCount();
if (rows>0)
{
    for(int i=0;i<rows;i++)
   {
   model.removeRow(0);
   }
}
try
{
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","");
smt= c.createStatement();
String query= "select admission_no, name, class, section from student where class="+clas+" and section='"+sectn+"';";
rs=smt.executeQuery(query);
while(rs.next())
{
 model.addRow(new Object [] {
 rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),
```

```
 });
}
rs.close();
smt.close();
c.close();
}
catch(Exception e)
{
    System.out.println(e.getMessage());
}
    }
```

Reset Button

```
 jTextField1.setText("");
 jPasswordField1.setText("");
 jLabel8.setText("");
```

Forgot Password

```
 try
 {

 c=DriverManager.getConnection("jdbc:mysql://localhost:330
 6/project","root","");
    smt=c.createStatement();
```

```
    rs=smt.executeQuery("select question,security_answer
from security,sques where
User_ID="'+jTextField1.getText()+'" and
security_question=sq_id;");
    rs.next();
    String m=rs.getString(1);
    String n=rs.getString(2);
while(i<=3)
{
 String ans=JOptionPane.showInputDialog(this,m,"Security
Question",2);
  if(n.equals(ans))
  {
     close();
     Home_Screen s= new Home_Screen();
     s.setVisible(true);
     break;
  }
  else
  {
     JOptionPane.showMessageDialog(this,"Oops ! Wrong
Answer ! You have "+(3-i)+" more chance to try");
```

```
  }
 i++;
}
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(this,"Your User_Id doesn't seem to be right");
}
    Cursor cur1=new Cursor(Cursor.HAND_CURSOR);
        jLabel10.setCursor(cur1);
```

Update Button

```
String publisherno=jTextField1.getText();
String publishername=jTextField2.getText();
int res= JOptionPane.showConfirmDialog(null,"Want to update the record?");
if(res==JOptionPane.YES_OPTION)
    {
try
 {
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","");
```

```
stmt= c.createStatement();
String query="update publisher set publisher_name=
"'+publishername+"' where
publisher_no="'+publisherno+"';";
stmt.executeUpdate(query);
JOptionPane.showMessageDialog(null,"Record Updated!");
albus();
 }
catch (Exception e)
 {
JOptionPane.showMessageDialog(this,"Something went
haywire there, why don't you try again ?");
 }
    }
resetting();
    }
```

Delete Button

```
String publisherno=jTextField1.getText();
int res= JOptionPane.showConfirmDialog(null,"Want to
delete the record?");
if(res==JOptionPane.YES_OPTION)
    {
```

```
try
 {
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","");
stmt= c.createStatement();
String query="delete from publisher where publisher_no="+publisherno+";";
stmt.executeUpdate(query);
JOptionPane.showMessageDialog(null,"Record Deleted!");
albus();
 }
catch (Exception e)
 {
JOptionPane.showMessageDialog(this,"Something went haywire there, why don't you try again ?");
 }
    }
resetting();
  }
```

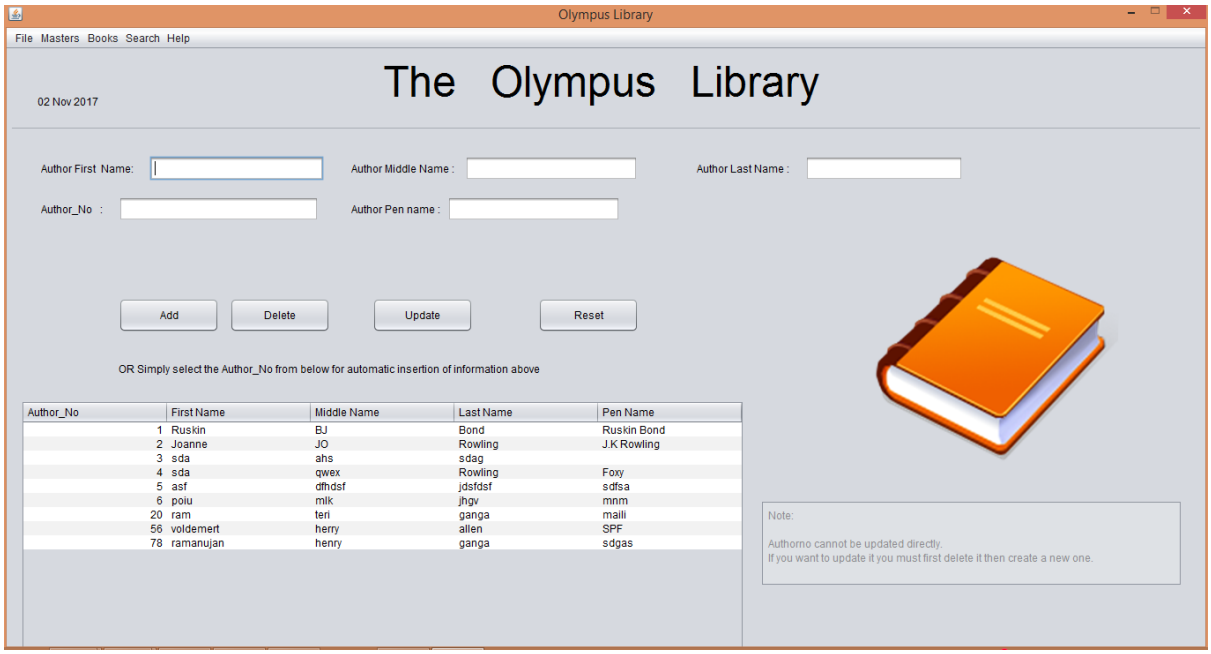
Add Button

```
String publishername=jTextField2.getText();
String publisherno=jTextField1.getText();
```

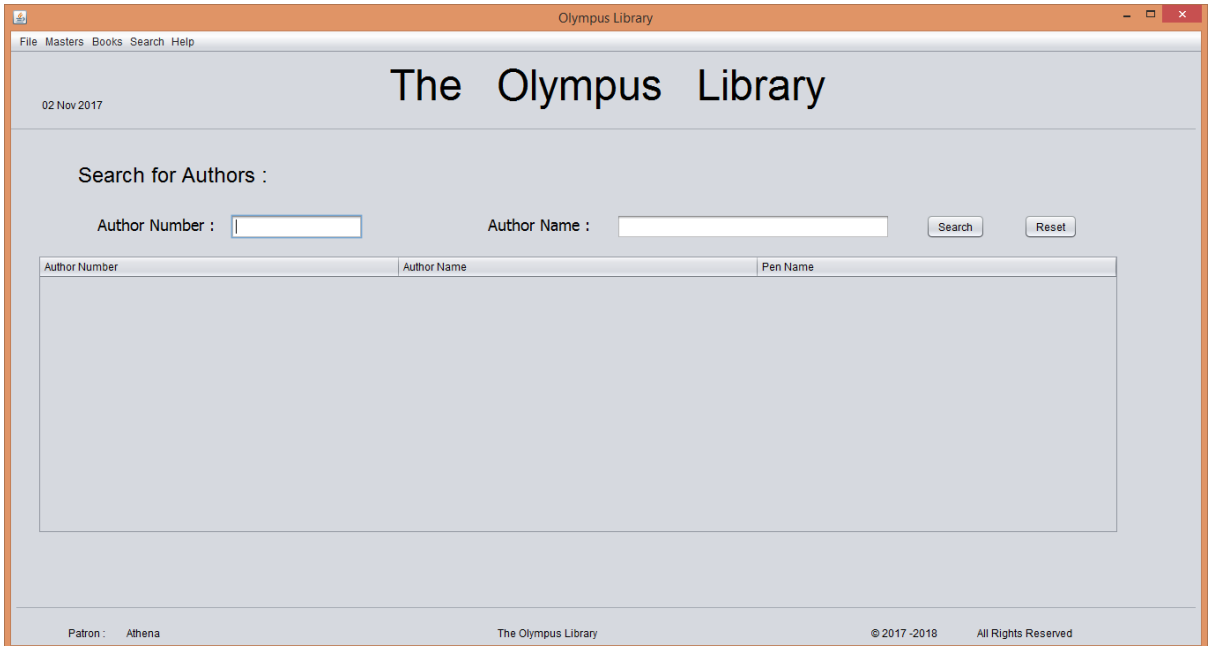
```
try
  {
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","");
stmt= c.createStatement();
String query="insert into publisher values('"+publisherno+"','"+publishername+"');";
stmt.executeUpdate(query);
JOptionPane.showMessageDialog(this,"Data successfully added. Click 'Ok' to continue!!!");
albus();
  }
catch (Exception e)
  {
JOptionPane.showMessageDialog(this,"Something went haywire there, why don't you try again ?");
  }
resetting();
  }
```

Screenshots of Front End

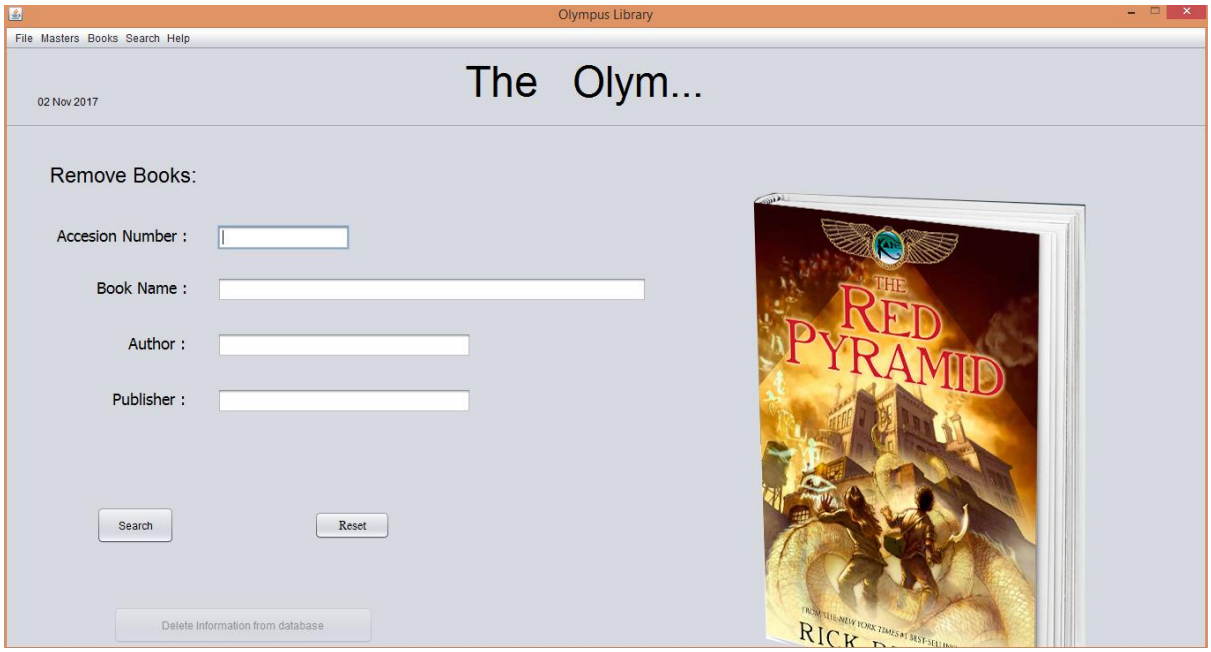
Author Master



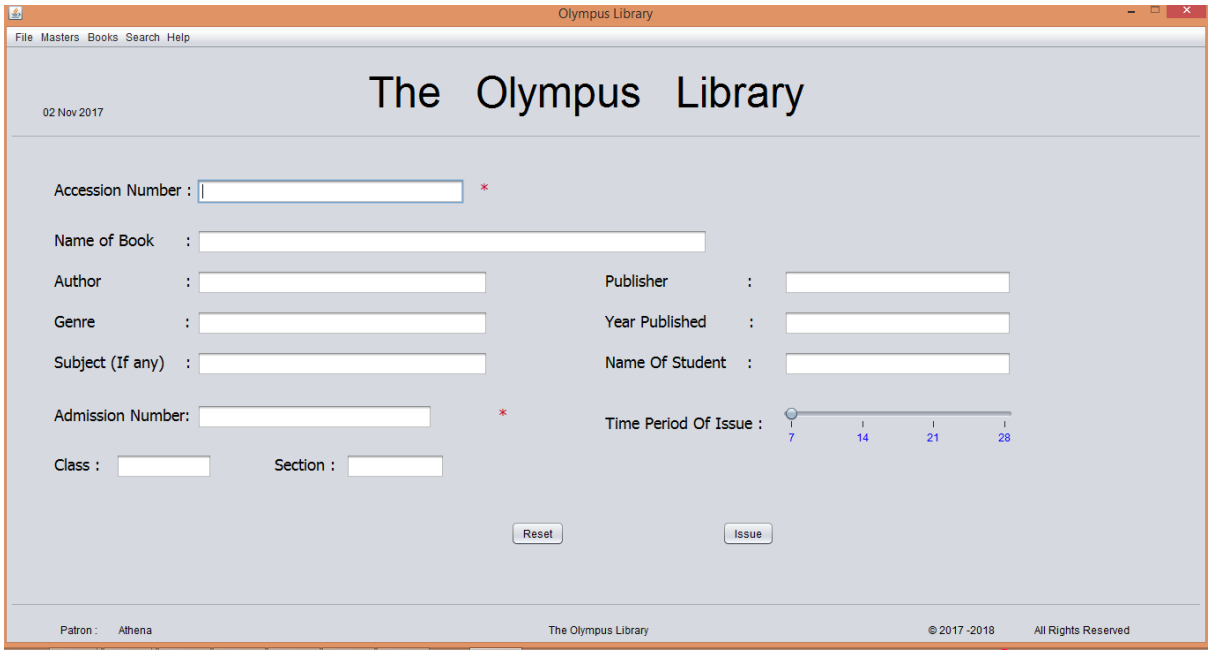
Author Search



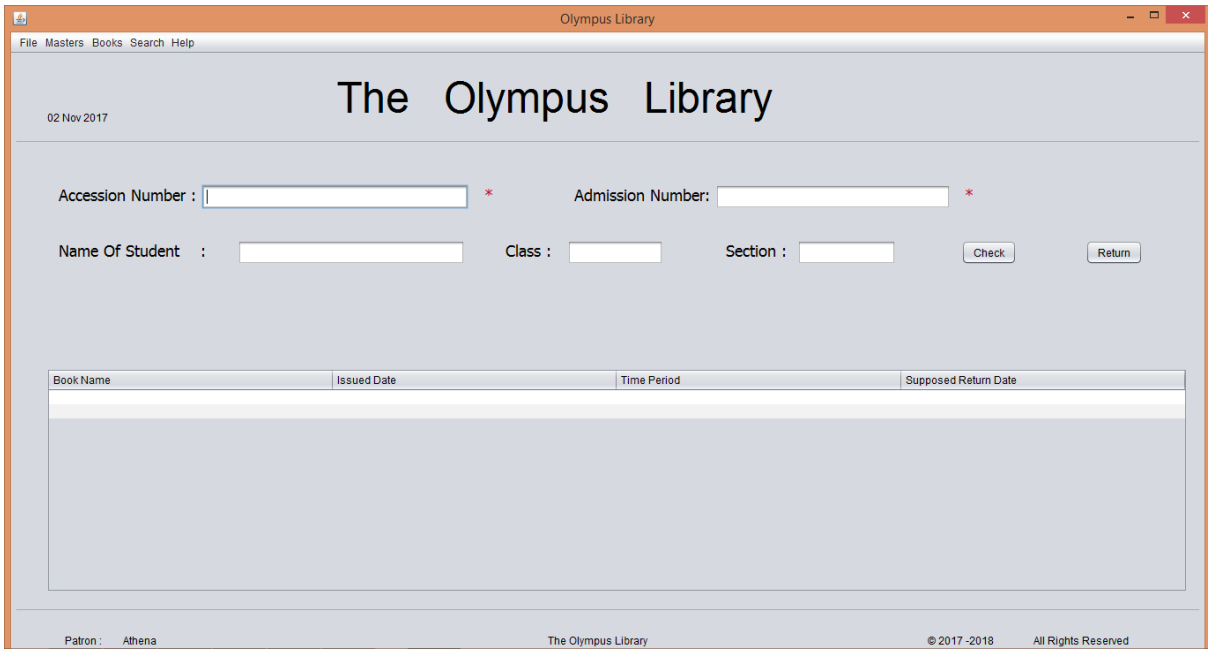
Book Remove



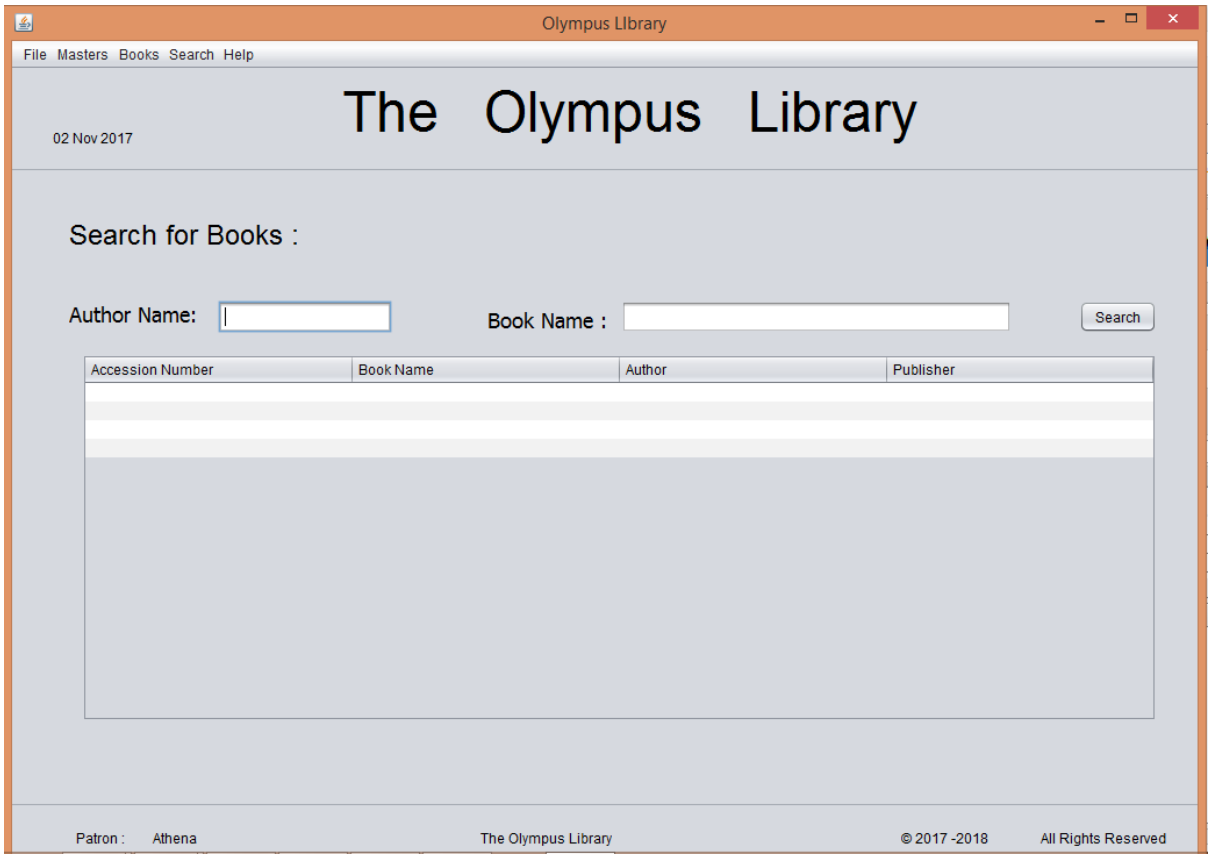
Book Issue



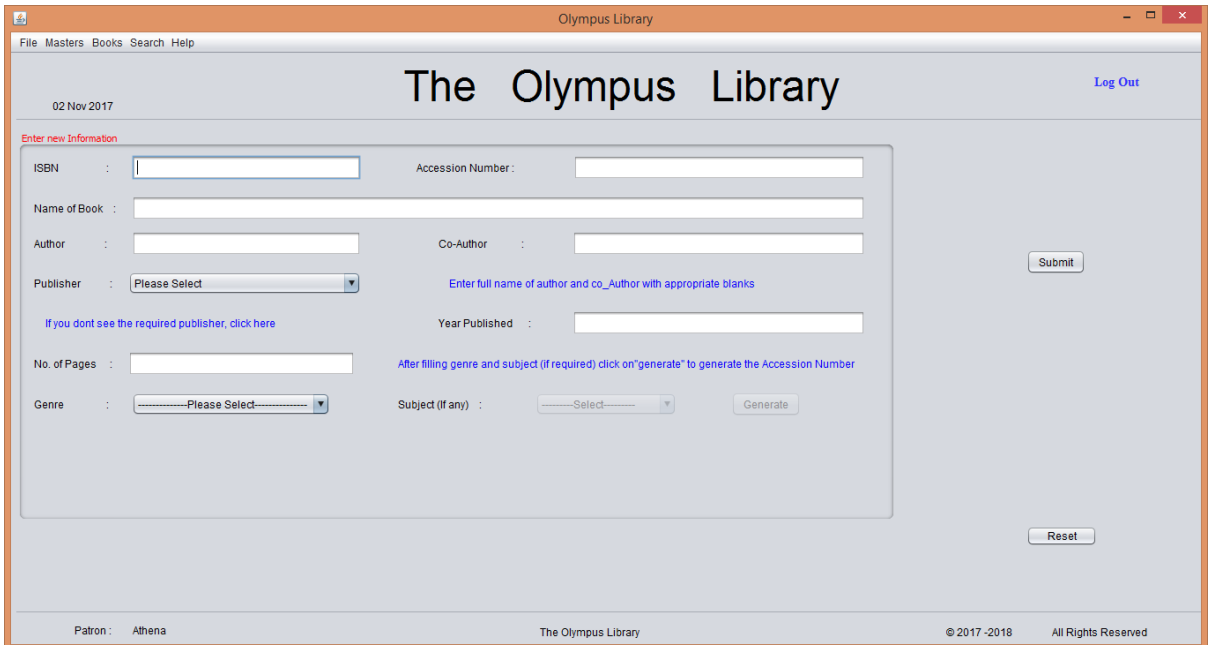
Book Return



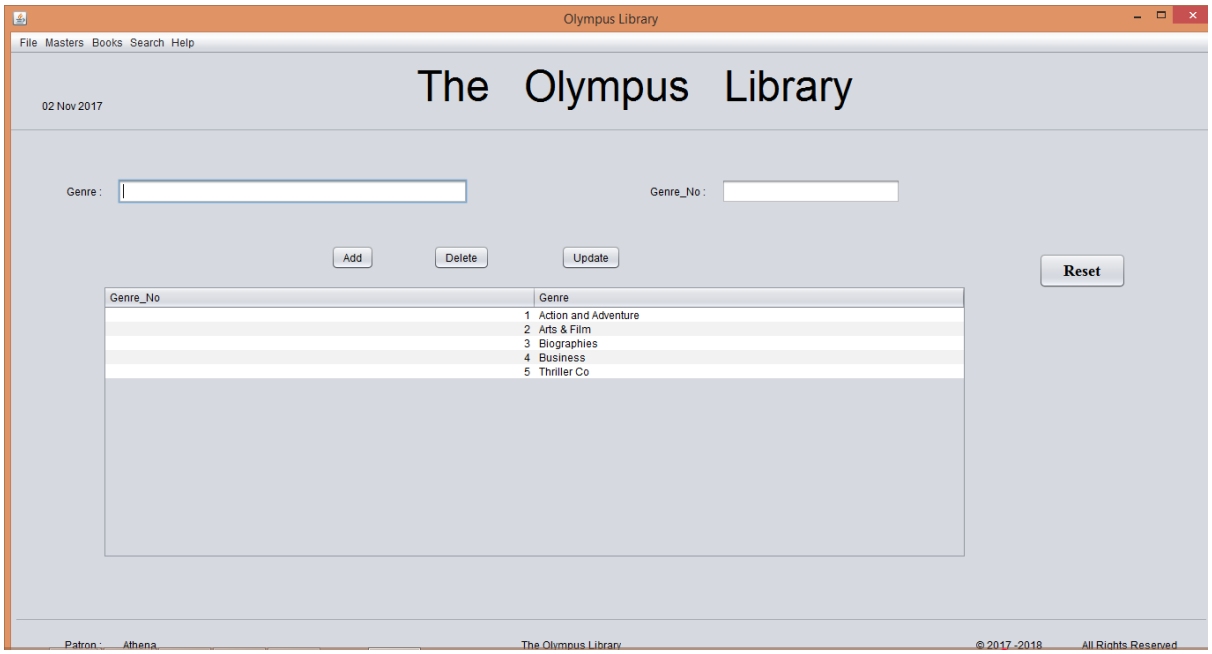
Book Search



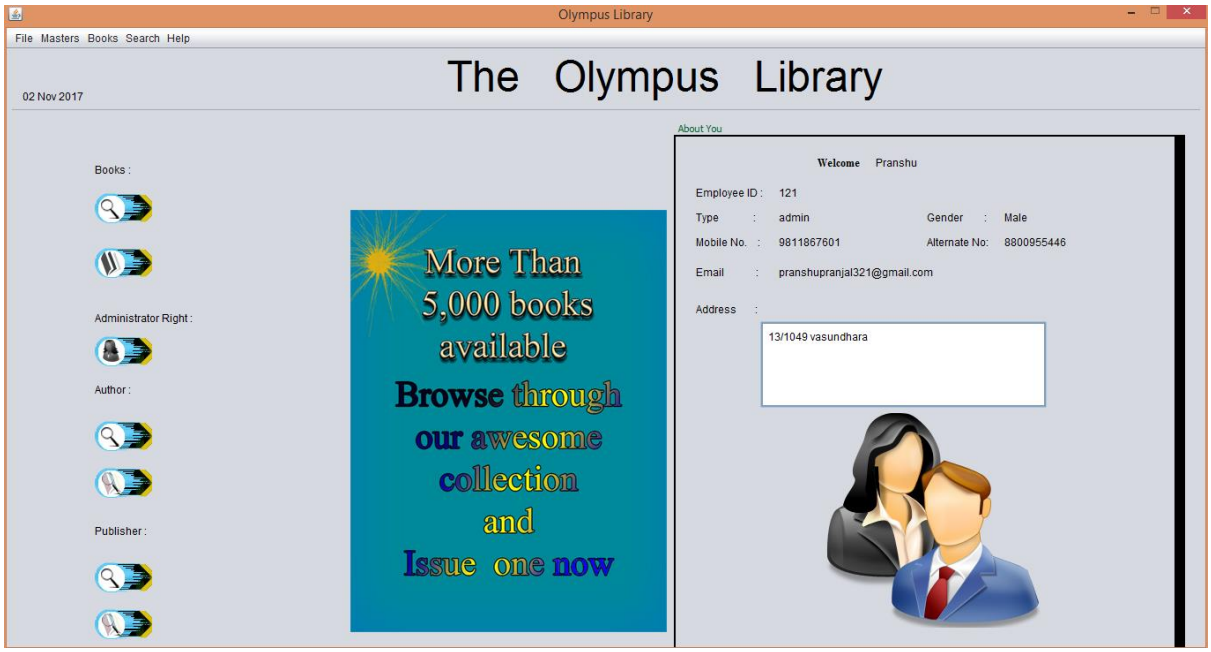
Book Add



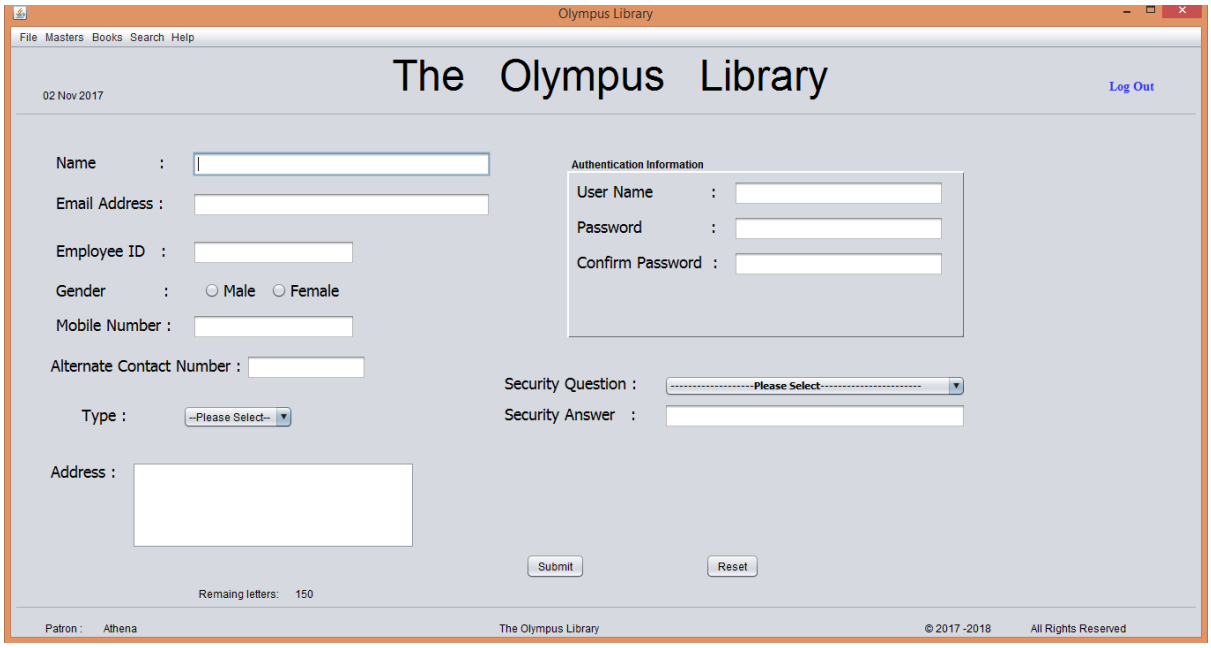
Genre Master



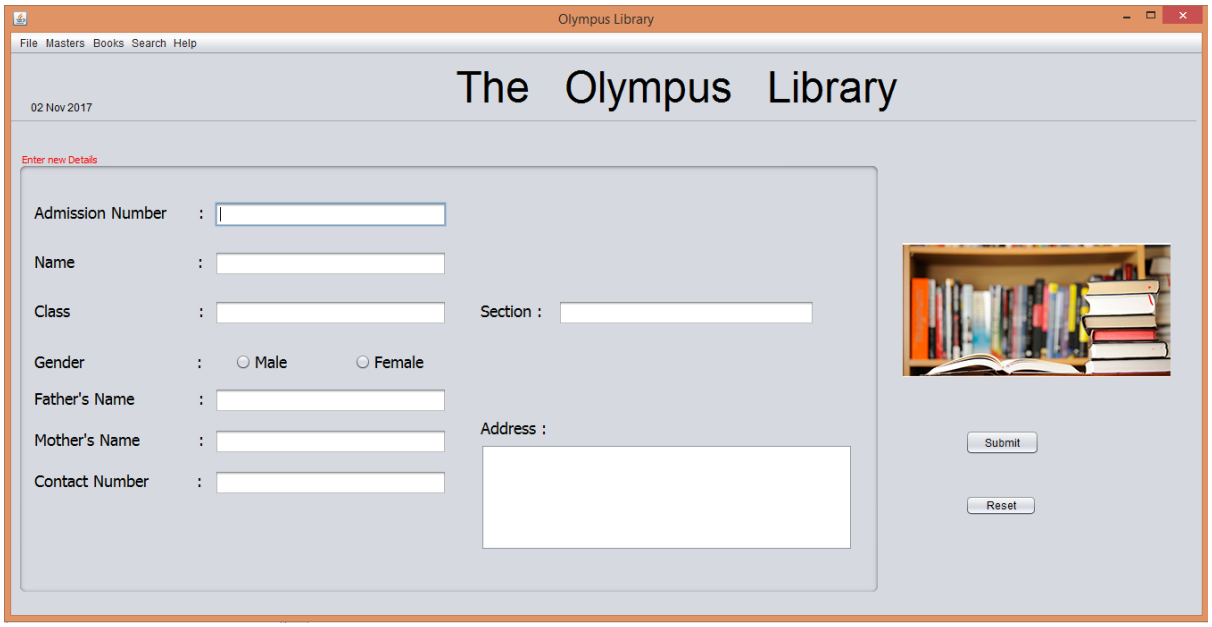
Home Screen



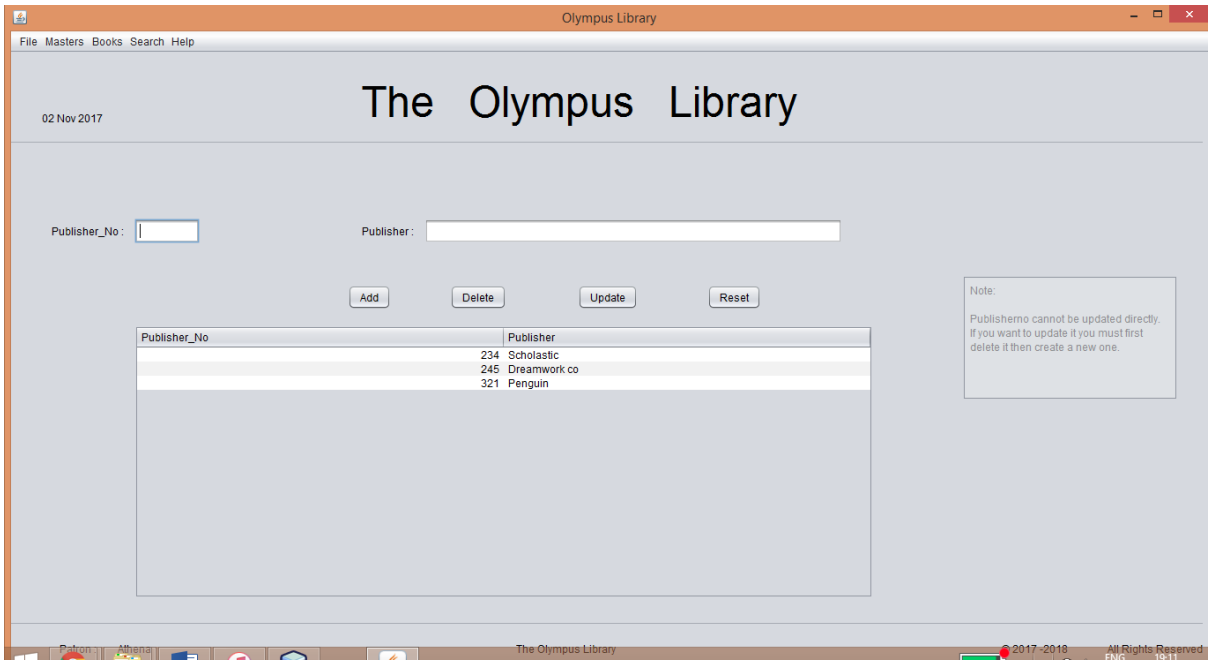
New Authorised Personnel



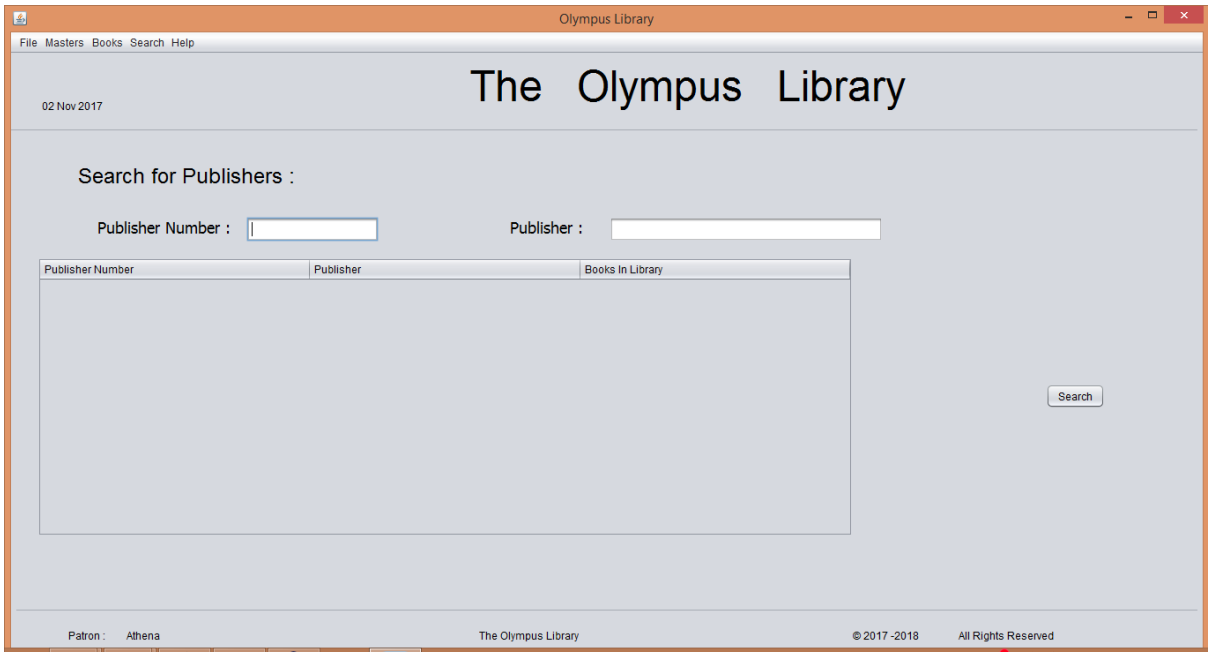
New Student Detail



Publisher Master

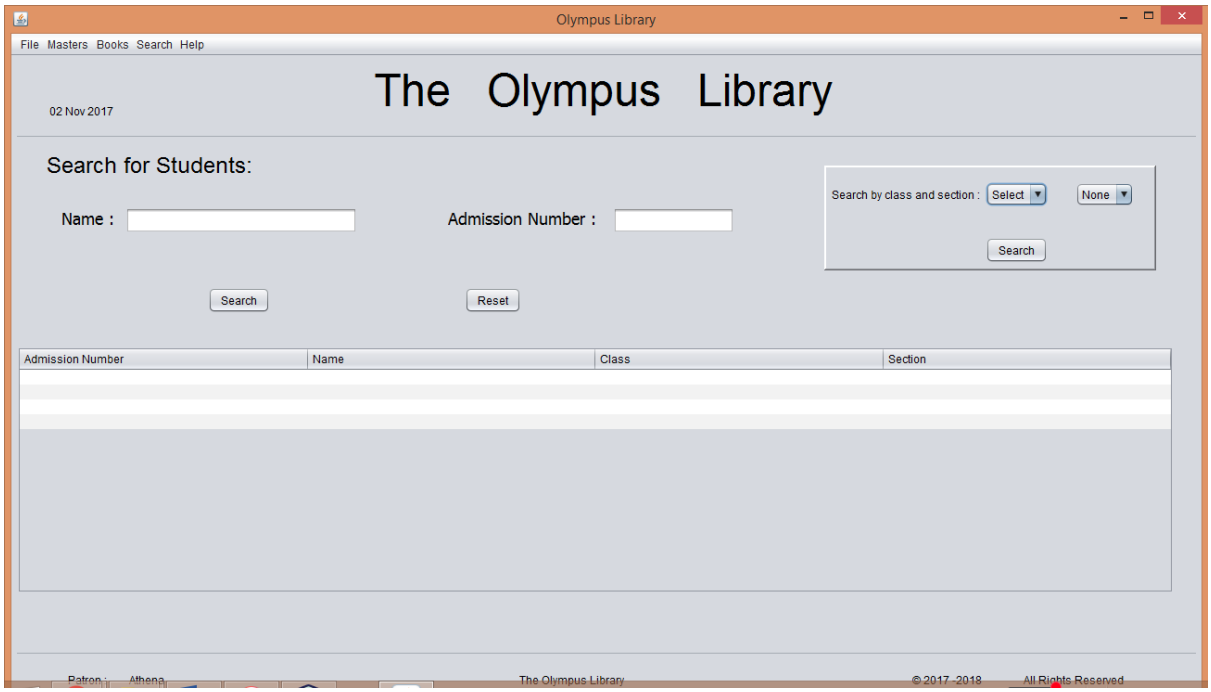


Publisher Search



Security Screen

Student Search



Conclusion

After completing the project on Library Management System I came to the conclusion that this Library DBMS is much more sophisticated and advantageous than a rudimentary physical library record book. *Library Management software* can be installed in a range of library organizations, ranging from academic libraries to joint use and public libraries. The *library management software* has advantages like being User Friendly, higher flexibility, high security, no data redundancy, efficient inventory and proper stock management, detailed automatically generated reports, and less filling errors. While doing this project I learned a lot of new concepts and tricks of Netbeans IDE and MySQL which really helped me a lot. We also came to realize that there is a potential of infinite development in this field of Informatics Practice and we soon hope to contribute to it and build better products in Information Technology.