Database tables:

Books table (Author, Title, BookID, Genre)

IssuedBooks table (Borrower infor, book info)

Other:

Subsystem admin login:

Login extends JFrame{  
 admin login frame;

jPanel contentPane;

JTextFields pswdField, userNField;

Public static void main (String[] args) {  
 //launch the app

EventQueue(Later) new Runnable

Public void run() {

try{

frame is new admin login();

frame visible is true

} catch

Print stack

}

Public admin login(){

Create GUI for login screen

Set username to admin

Set password “pswd” to admin123

If both username and pswd match, allow logon

Else don’t allow logon and display error message

}

}

Subsystems addbook:

Public class addBooks{

Create addbooks btn;

Public static int save (Stringe bookID, String Title, String Author, String genre, int quantity){

Try{  
 connect to database;

Prepared statement to insert book information above into table;

Set values;

Update database;

Close connection;

} catch exception{}

Return status;

}

Return to menu;

}

Susbsystem remove books:

Public class remoceBooks{

Create removebooks btn;

Public static int save (Stringe bookID, String Title, String Author, String genre, int quantity){

Try{  
 connect to database;

Prepared statement to remove or delete a book using a primary key;

Set values;

Update database;

Close connection;

} catch exception{}

Return status;

}

Ret

Subsystem issue book:

Public class issueBook(){

Create issuebooks btn;

Public static Boolean checkBook(String bookID){

Boolean is false;

Try{

connect to database;

Prepared statement to select information from books table;

Execute query;

Update status;

Close connection;

} catch exception e{}

Return status;

}

Pubclis static int save(int bookID, int studentID, String studentName, String student contact){

Status is 0;

Try{

connect to database;

Prepared statement to insert information into issueBook table;

Execute query;

Update status;

Close connection;

} catch exception e{}

Return status;

Return to menu;

}

Update the quantity of books;

Subsystem View books

Public class viewBook(){

GUI for book display/ create view books btn;

Public static void main(String[]args)

Run this component

Try{

connect to database;

Prepared statement to select information from books table;

Execute query;

Display information in table format;

Update status;

Close connection;

} catch exception e{}

Return status;

return to menu

}

Subsystem View books

Public class viewIssuedBook(){

GUI for issued book display/ create view issued books button;

Public static void main(String[]args)

Run this component;

Try{

connect to database;

Prepared statement to select information from issuedBooks table;

Execute query;

Display information in table format;

Update status;

Close connection;

} catch exception e{}

Return status;

return to menu;

}

Subsystem return Book

Public class returnBook(){  
EventQueue for runnable component;

Public void run()

Try{

Run;

}catch( exception e){  
 }

Public returnBook(){  
 setup Gui configuration for this component

jLabel BookID;

jLbadel Student ID;

set up return book button;

try{

connect to database;

prepared statement to get quantity from books table;

update quantity from books/ issued books table;

set status to update;

close the connection;

return to menu;

}catch(exception e){}

}

Subsystem search books:

Public class searchBooks(){  
 setup Gui configuration for this component

jLabel BookID;

jLbadel other book information;

set up search book button;

try{

connect to database;

prepared statement to get some information from books table dependent on author, genre, title, etc;

display books with specified criteria;

set status to update;

close the connection;

}catch(exception e){}