CREATE DATABASE LIBRARY;

CREATE TABLE LIBRARY.BOOK(

Book\_Id CHAR(30) NOT NULL,

Title VARCHAR(30) NOT NULL,

PRIMARY KEY(Book\_Id)

);

CREATE TABLE LIBRARY.BOOK\_AUTHORS(

Book\_Id CHAR(30) NOT NULL,

Author\_name VARCHAR(30) NOT NULL,

PRIMARY KEY(Book\_Id, Author\_name),

FOREIGN KEY (Book\_Id) REFERENCES BOOK (Book\_Id)

);

CREATE TABLE LIBRARY.LIBRARY\_BRANCH(

Branch\_Id CHAR(20) NOT NULL,

Branch\_name VARCHAR(20) NOT NULL,

Address VARCHAR(100) NOT NULL,

PRIMARY KEY(Branch\_Id)

);

CREATE TABLE LIBRARY.BOOK\_COPIES(

Book\_Id CHAR(30) NOT NULL,

Branch\_Id CHAR(20) NOT NULL,

No\_Of\_Copies INTEGER(10) NOT NULL,

PRIMARY KEY(Book\_Id,Branch\_Id),

FOREIGN KEY (Book\_Id) REFERENCES BOOK (Book\_Id),

FOREIGN KEY (Branch\_Id) REFERENCES LIBRARY\_BRANCH (Branch\_Id)

);

CREATE TABLE LIBRARY.BORROWER(

Card\_no INTEGER(20) NOT NULL,

Fname VARCHAR(15) NOT NULL,

Lname VARCHAR(15) NOT NULL,

Address VARCHAR(100) NOT NULL,

Phone Varchar(20) NOT NULL,

PRIMARY KEY(Card\_no)

);

CREATE TABLE LIBRARY.BOOK\_LOANS(

Book\_Id CHAR(30),

Branch\_Id CHAR(20),

Card\_no INTEGER(20) NOT NULL,

Date\_out DATE NOT NULL,

Due\_date DATE NOT NULL,

Date\_in DATE NOT NULL,

PRIMARY KEY(Book\_Id, Branch\_Id, Card\_no),

FOREIGN KEY (Book\_Id) REFERENCES BOOK (Book\_Id),

FOREIGN KEY (Branch\_Id) REFERENCES LIBRARY\_BRANCH (Branch\_Id),

FOREIGN KEY (Card\_no) REFERENCES BORROWER (Card\_no)

);

* To delete all rows of table

delete from library.library\_branch

* To load data from a csv file

load data local infile 'C:/Users/Archit Garg.ARCHIT/Desktop/DB Project/Library\_data/library\_branch.csv'

into table library.library\_branch

fields terminated by '\t'

enclosed by ""

lines terminated by '\n';

* to load data from csv file skipping a column/entering data from particular columns only

load data local infile 'C:/book\_authors.csv' into table library\_branch fields terminated by '\t' enclosed by '"' lines terminated by '\n' (branch\_id,@dummy,address);

* to skip reading the first line of data from csv

load data local infile 'C:/Users/Archit Garg.ARCHIT/Desktop/DB Project/Library\_data/books\_authors.csv' into table book fields terminated by '\t' enclosed by '"' lines terminated by '\n' ignore 1 lines (book\_id, @dummy, title);

load data local infile 'C:/Users/Archit Garg.ARCHIT/Desktop/DB Project/Library\_data/books\_authors.csv' into table book\_authors fields terminated by '\t' enclosed by '"' lines terminated by '\n' ignore 1 lines (book\_id, author\_name,@dummy);

load data local infile 'C:/Users/Archit Garg.ARCHIT/Desktop/DB Project/Library\_data/library\_branch.csv' into table library\_branch fields terminated by '\t' enclosed by '"' lines terminated by '\n' ignore 1 lines (branch\_id, branch\_name, address);

load data local infile 'C:/Users/Archit Garg.ARCHIT/Desktop/DB Project/Library\_data/book\_copies.csv' into table book\_copies fields terminated by '\t' enclosed by '"' lines terminated by '\n' ignore 1 lines (book\_id, branch\_id, no\_of\_copies);

load data local infile 'C:/Users/Archit Garg.ARCHIT/Desktop/DB Project/Library\_data/borrowers.csv' into table borrower fields terminated by '\t' enclosed by '"' lines terminated by '\n' ignore 1 lines (card\_no,fname,lname,address,city,state,phone

);

* schema change

alter table borrower drop column phone;

alter table borrower add column city VARCHAR(20) add column state not null

alter table borrower add column phone VARCHAR(20);

load data local infile 'C:/Users/Archit Garg.ARCHIT/Desktop/DB Project/Library\_data/borrowers.csv' into table borrower fields terminated by '\t' enclosed by '"' lines terminated by '\n' ignore 1 lines (card\_no,fname,lname,address,city,state,phone);

update borrower set address=concat(address,' , ',city,' , ',state)

* merge two columns in one

UPDATE table\_name SET

full\_name = CONCAT(first\_name, ' ', last\_name);

* ALTER TABLE book\_copies DROP FOREIGN KEY book\_id
* Program to run a select \* query from java eclipse using jdbc

//STEP 1. Import required packages

**import** java.sql.\*;

**public** **class** MySQLTest {

// JDBC driver name and database URL

**static** **final** String *JDBC\_DRIVER* = "com.mysql.jdbc.Driver";

**static** **final** String *DB\_URL* = "jdbc:mysql://localhost/Library";

// Database credentials

**static** **final** String *USER* = "root";

**static** **final** String *PASS* = "";

**public** **static** **void** main(String[] args) {

Connection conn = **null**;

Statement stmt = **null**;

**try**{

//STEP 2: Register JDBC driver

Class.*forName*("com.mysql.jdbc.Driver");

//STEP 3: Open a connection

System.*out*.println("Connecting to database...");

conn = DriverManager.*getConnection*(*DB\_URL*,*USER*,*PASS*);

//STEP 4: Execute a query

System.*out*.println("Creating statement...");

stmt = conn.createStatement();

String sql;

sql = "SELECT \* FROM library.Book";

ResultSet rs = stmt.executeQuery(sql);

**while**(rs.next())

{

System.*out*.println(rs.getString(1)); //First Column

System.*out*.println(rs.getString(2)); //Second Column

}

}

**catch**(SQLException se){

//Handle errors for JDBC

se.printStackTrace();

}**catch**(Exception e){

//Handle errors for Class.forName

e.printStackTrace();

}

}

}

* USING CARDS(card Layout) IN GUI FOR BETTER LOOK ... We can use tabbed panes also

website : http://java.about.com/od/Layout\_Managers/ss/Cardlayout-Example-Program.htm

By the use of tabbed panes (logic)

**import** javax.swing.JFrame;

**public** **class** Calc **extends** JFrame {

**public** Calc() {

setTitle("Tabbed Pane");

JTabbedPane jtp = **new** JTabbedPane();

getContentPane().add(jtp);

JPanel jp1 = **new** JPanel();

JPanel jp2 = **new** JPanel();

JLabel label1 = **new** JLabel();

label1.setText("You are in area of Tab1");

JLabel label2 = **new** JLabel();

label2.setText("You are in area of Tab2");

jp1.add(label1);

jp2.add(label2);

jtp.addTab("", jp1);

jtp.addTab("", jp2);

}

**public** **static** **void** main(String[] args) {

Calc tp = **new** Calc();

tp.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

tp.setVisible(**true**);

}

}

CARD LAYOUT PROGRAM FOR LIB MGMT

//Imports are listed in full to show what's being used

//could just import javax.swing.\* and java.awt.\* etc..

**package** testPanel;

**import** java.awt.EventQueue;

**import** java.awt.BorderLayout;

**import** java.awt.CardLayout;

**import** java.awt.Color;

**import** javax.swing.BorderFactory;

**import** javax.swing.border.Border;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JComboBox;

**import** javax.swing.JButton;

**import** javax.swing.JLabel;

**import** javax.swing.SwingConstants;

**import** java.awt.Container;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** javax.swing.BoxLayout;

**import** com.jgoodies.forms.layout.FormLayout;

**import** com.jgoodies.forms.layout.ColumnSpec;

**import** com.jgoodies.forms.layout.RowSpec;

**import** java.awt.GridBagLayout;

**import** javax.swing.GroupLayout;

**import** javax.swing.GroupLayout.Alignment;

**import** javax.swing.JTextField;

**import** javax.swing.LayoutStyle.ComponentPlacement;

**public** **class** testPanel {

JFrame guiFrame;

CardLayout cards;

JPanel cardPanel;

JButton BookSearch;

JPanel firstCard,secondCard,thirdCard;

**private** JTextField textField;

**private** JTextField textField\_1;

**private** JTextField textField\_2;

**private** JTextField textField\_3;

**private** JTextField textField\_4;

**private** JTextField textField\_5;

**private** JTextField textField\_6;

**private** JTextField textField\_7;

**private** JTextField textField\_8;

**private** JTextField textField\_9;

**private** JTextField textField\_10;

**private** JTextField textField\_11;

**private** JTextField textField\_12;

**public** **static** **void** main(String[] args) {

//Use the event dispatch thread for Swing components

EventQueue.*invokeLater*(**new** Runnable()

{

@Override

**public** **void** run()

{

**new** testPanel();

}

});

}

**public** testPanel()

{

guiFrame = **new** JFrame();

//make sure the program exits when the frame closes

guiFrame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

guiFrame.setTitle("LIBRARY GUI");

guiFrame.setSize(469,380);

//This will center the JFrame in the middle of the screen

guiFrame.setLocationRelativeTo(**null**);

guiFrame.getContentPane().setLayout(**new** BorderLayout());

//creating a border to highlight the JPanel areas

Border outline = BorderFactory.*createLineBorder*(Color.*black*);

JPanel tabsPanel = **new** JPanel();

tabsPanel.setBorder(outline);

BookSearch = **new** JButton("BOOK SEARCH");

BookSearch.setActionCommand("BOOK SEARCH");

BookSearch.addActionListener(**new** ActionListener()

{

@Override

**public** **void** actionPerformed(ActionEvent event)

{

cards.show(cardPanel, "BOOK SEARCH");

}

});

tabsPanel.add(BookSearch);

JButton BookLoans = **new** JButton("BOOK LOANS");

BookLoans.setActionCommand("BOOK LOANS");

BookLoans.addActionListener(**new** ActionListener()

{

@Override

**public** **void** actionPerformed(ActionEvent event)

{

cards.show(cardPanel, "BOOK LOANS");

}

});

tabsPanel.add(BookLoans);

JButton BorrowerManagement = **new** JButton("BORROWER MANAGEMENT");

BorrowerManagement.setActionCommand("Switch Card");

BorrowerManagement.addActionListener(**new** ActionListener()

{

@Override

**public** **void** actionPerformed(ActionEvent event)

{

cards.show(cardPanel,"BORROW MANAGEMENT");

}

});

tabsPanel.add(BorrowerManagement);

guiFrame.getContentPane().add(tabsPanel,BorderLayout.*NORTH*);

cards = **new** CardLayout();

cardPanel = **new** JPanel();

cardPanel.setLayout(cards);

cards.show(cardPanel, "Book Search");

firstCard = **new** JPanel();

firstCard.setBackground(Color.*GREEN*);

secondCard = **new** JPanel();

secondCard.setBackground(Color.*BLUE*);

thirdCard = **new** JPanel();

secondCard.setBackground(Color.*BLUE*);

cardPanel.add(firstCard, "BOOK SEARCH");

JLabel lblBookId = **new** JLabel("BOOK ID :");

textField = **new** JTextField();

textField.setColumns(10);

JLabel lblTitle = **new** JLabel("TITLE :");

JLabel lblAuthorName = **new** JLabel("AUTHOR NAME :");

JButton btnSearch = **new** JButton("SEARCH");

JButton btnBack = **new** JButton("BACK");

textField\_1 = **new** JTextField();

textField\_1.setColumns(10);

textField\_2 = **new** JTextField();

textField\_2.setColumns(10);

GroupLayout gl\_firstCard = **new** GroupLayout(firstCard);

gl\_firstCard.setHorizontalGroup(

gl\_firstCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_firstCard.createSequentialGroup()

.addContainerGap()

.addGroup(gl\_firstCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_firstCard.createSequentialGroup()

.addComponent(btnSearch)

.addGap(43)

.addComponent(btnBack))

.addGroup(gl\_firstCard.createParallelGroup(Alignment.*TRAILING*, **false**)

.addGroup(gl\_firstCard.createSequentialGroup()

.addComponent(lblAuthorName)

.addPreferredGap(ComponentPlacement.*RELATED*, GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

.addComponent(textField\_2, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGroup(Alignment.*LEADING*, gl\_firstCard.createSequentialGroup()

.addGroup(gl\_firstCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(lblBookId)

.addComponent(lblTitle))

.addGap(77)

.addGroup(gl\_firstCard.createParallelGroup(Alignment.*TRAILING*)

.addComponent(textField, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(textField\_1, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)))))

.addContainerGap(231, Short.*MAX\_VALUE*))

);

gl\_firstCard.setVerticalGroup(

gl\_firstCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_firstCard.createSequentialGroup()

.addGap(19)

.addGroup(gl\_firstCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblBookId)

.addComponent(textField, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(27)

.addGroup(gl\_firstCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblTitle)

.addComponent(textField\_1, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(29)

.addGroup(gl\_firstCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblAuthorName)

.addComponent(textField\_2, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(26)

.addGroup(gl\_firstCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(btnSearch)

.addComponent(btnBack))

.addContainerGap(134, Short.*MAX\_VALUE*))

);

firstCard.setLayout(gl\_firstCard);

cardPanel.add(secondCard, "BOOK LOANS");

JLabel lblCheckingInBook = **new** JLabel("CHECKING IN BOOK");

JLabel lblCheckingOutBook = **new** JLabel("CHECKING OUT BOOK");

JLabel lblBookId\_1 = **new** JLabel("BOOK ID");

JLabel lblCardNo = **new** JLabel("CARD NO.");

JLabel lblDateIn = **new** JLabel("DATE IN");

textField\_3 = **new** JTextField();

textField\_3.setColumns(10);

textField\_4 = **new** JTextField();

textField\_4.setColumns(10);

textField\_5 = **new** JTextField();

textField\_5.setColumns(10);

JLabel lblBookId\_2 = **new** JLabel("BOOK ID");

JLabel lblCardNo\_1 = **new** JLabel("CARD NO.");

JLabel lblBranchNo = **new** JLabel("BRANCH NO.");

JLabel lblDateOut = **new** JLabel("DATE OUT");

JLabel lblDueDate = **new** JLabel("DUE DATE");

textField\_6 = **new** JTextField();

textField\_6.setColumns(10);

textField\_7 = **new** JTextField();

textField\_7.setColumns(10);

textField\_8 = **new** JTextField();

textField\_8.setColumns(10);

textField\_9 = **new** JTextField();

textField\_9.setColumns(10);

textField\_10 = **new** JTextField();

textField\_10.setColumns(10);

GroupLayout gl\_secondCard = **new** GroupLayout(secondCard);

gl\_secondCard.setHorizontalGroup(

gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_secondCard.createSequentialGroup()

.addGap(21)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_secondCard.createSequentialGroup()

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(lblBookId\_1)

.addComponent(lblCardNo)

.addComponent(lblDateIn))

.addGap(25)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(textField\_3, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(textField\_4, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(textField\_5, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)))

.addComponent(lblCheckingInBook))

.addGap(71)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(lblCheckingOutBook)

.addGroup(gl\_secondCard.createSequentialGroup()

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(lblBookId\_2)

.addComponent(lblDueDate)

.addComponent(lblDateOut)

.addComponent(lblBranchNo)

.addComponent(lblCardNo\_1))

.addGap(18)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(textField\_6, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(textField\_7, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(textField\_8, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(textField\_9, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(textField\_10, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))))

.addContainerGap(33, Short.*MAX\_VALUE*))

);

gl\_secondCard.setVerticalGroup(

gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_secondCard.createSequentialGroup()

.addGap(23)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblCheckingOutBook)

.addComponent(lblCheckingInBook))

.addGap(18)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblBookId\_1)

.addComponent(textField\_3, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(lblBookId\_2)

.addComponent(textField\_6, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(18)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblCardNo)

.addComponent(textField\_4, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(lblCardNo\_1)

.addComponent(textField\_7, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(18)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(lblDateIn)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(textField\_5, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(lblBranchNo)

.addComponent(textField\_8, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)))

.addGap(18)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblDateOut)

.addComponent(textField\_9, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(18)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblDueDate)

.addComponent(textField\_10, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addContainerGap(79, Short.*MAX\_VALUE*))

);

secondCard.setLayout(gl\_secondCard);

cardPanel.add(thirdCard, "BORROW MANAGEMENT");

JLabel lblNameOfBorrower = **new** JLabel("NAME OF BORROWER");

textField\_11 = **new** JTextField();

textField\_11.setColumns(10);

JLabel lblAddressOfBorrower = **new** JLabel("ADDRESS OF BORROWER");

textField\_12 = **new** JTextField();

textField\_12.setColumns(10);

GroupLayout gl\_thirdCard = **new** GroupLayout(thirdCard);

gl\_thirdCard.setHorizontalGroup(

gl\_thirdCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_thirdCard.createSequentialGroup()

.addGap(28)

.addGroup(gl\_thirdCard.createParallelGroup(Alignment.*LEADING*, **false**)

.addGroup(gl\_thirdCard.createSequentialGroup()

.addComponent(lblAddressOfBorrower)

.addGap(29)

.addComponent(textField\_12, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGroup(gl\_thirdCard.createSequentialGroup()

.addComponent(lblNameOfBorrower)

.addPreferredGap(ComponentPlacement.*RELATED*, GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

.addComponent(textField\_11, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)))

.addContainerGap(185, Short.*MAX\_VALUE*))

);

gl\_thirdCard.setVerticalGroup(

gl\_thirdCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_thirdCard.createSequentialGroup()

.addGap(36)

.addGroup(gl\_thirdCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblNameOfBorrower)

.addComponent(textField\_11, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(44)

.addGroup(gl\_thirdCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblAddressOfBorrower)

.addComponent(textField\_12, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addContainerGap(192, Short.*MAX\_VALUE*))

);

thirdCard.setLayout(gl\_thirdCard);

guiFrame.getContentPane().add(tabsPanel,BorderLayout.*NORTH*);

guiFrame.getContentPane().add(cardPanel,BorderLayout.*CENTER*);

guiFrame.setVisible(**true**);

}

//All the buttons are following the same pattern

//so create them all in one place.

}

* QUERIES

Book Copies

book\_id branch\_id no\_of\_copies

967697603 1 0

967697603 2 1

967697603 3 4

967697603 4 3

967697603 5 0

Library Branch

branch\_id branch\_name address

1 Oak Lawn 4100 Cedar Springs Road, 75219

2 Lakewood 6121 Worth Street, 75214

3 Grauwyler Park 2146 Gilford Street, 75235

4 Highland Hills 3624 Simpson Stuart Road, 75241

5 Audelia Road 10045 Audelia Road, 75238

Borrowers

card\_no fname lname address city state phone

9005 John Smith 731 Fondren Dallas TX (214) 328-2218

9006 Evan Wallis 134 Pelham Dallas TX (214) 789-9683

9007 Josh Zell 266 McGrady Plano TX (972) 718-7235

9008 Andy Vile 1967 Jordan Dallas TX (214) 374-5503

9009 Tom Brand 112 Third St Dallas TX (214) 897-9919

insert into BOOK\_LOANS

values('967697603', '1', 9005, NULL,NULL,NULL);

insert into BOOK\_LOANS

values('967697603', '2', 9006, NULL,NULL,NULL);

insert into BOOK\_LOANS

values('967697603', '3', 9007, NULL,NULL,NULL);

insert into BOOK\_LOANS

values('967697603', '4', 9008, NULL,NULL,NULL);

insert into BOOK\_LOANS

values('967697603', '5', 9009, NULL,NULL,NULL);

insert into BOOK\_LOANS

values ('30303370','2',9005, NULL,NULL,NULL);

insert into BOOK\_LOANS

values ('30303370','2',9005,NULL,NULL,NULL);

insert into BOOK\_LOANS

values ('60930497','5',9008, NULL,NULL,NULL);

Book loans

Bookid branch id card no date out due date date in

BOOK SEARCH AND AVAILIBILITY

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c .no\_of\_copies,

from book b,book\_authors b\_a,book\_copies b\_c

where b.book\_id = '967697603' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id;

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c.no\_of\_copies

from book b,book\_authors b\_a,book\_copies b\_c

where b.title = 'Seafood Lovers Almanac' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id;

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c.no\_of\_copies

from book b,book\_authors b\_a,book\_copies b\_c

where b\_a.author\_name = 'Nancy Parent' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id;

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c.no\_of\_copies

from book b,book\_authors b\_a,book\_copies b\_c

where b.book\_id = '967697603' and b.title = 'Seafood Lovers Almanac' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id;

or

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c.no\_of\_copies

from book b,book\_authors b\_a,book\_copies b\_c

where b.book\_id = '967697603' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id

union

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c.no\_of\_copies

from book b,book\_authors b\_a,book\_copies b\_c

where b.title = 'Seafood Lovers Almanac' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id;

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c.no\_of\_copies

from book b,book\_authors b\_a,book\_copies b\_c

where b.book\_id = '967697603' and b\_a.author\_name = 'Mercedes Lee' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id;

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c.no\_of\_copies

from book b,book\_authors b\_a,book\_copies b\_c

where b.book\_id = '967697603' and b.title = 'Seafood Lovers Almanac' and b\_a.author\_name = 'Mercedes Lee' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id;

select (b\_c.no\_of\_copies - count(b\_l.book\_id)) as available\_books

from book b, book\_authors b\_a, book\_copies b\_c,book\_loans b\_l

where b\_l.book\_id = b.book\_id and b\_c.book\_id = b.book\_id and b\_a.book\_id=b.book\_id and b\_l.branch\_id=b\_c.branch\_id and b\_l.book\_id='60930497' group by b\_c.branch\_id;

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c .no\_of\_copies,

(b\_c.no\_of\_copies - count(b\_l.book\_id)) as available\_books

from book b,book\_authors b\_a,book\_copies b\_c, book\_loans b\_l

where b.book\_id like'%3%' and b.title like '% %' and b\_a.author\_name like '% %' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id and b\_l.book\_id = b.book\_id and b\_l.branch\_id=b\_c.branch\_id group by b\_c.branch\_id, b\_c.book\_id

union

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c .no\_of\_copies, (b\_c.no\_of\_copies - count(b\_l.book\_id)) as available\_books

from book b,book\_authors b\_a,book\_copies b\_c,book\_loans b\_l

where b.book\_id like'% 15%' and b.title like '% %' and b\_a.author\_name like '% %' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id and b\_l.book\_id <> b.book\_id and b\_l.branch\_id<>b\_c.branch\_id group by b\_c.branch\_id, b\_c.book\_id;

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c .no\_of\_copies,

(b\_c.no\_of\_copies - count(b\_l.book\_id)) as available\_books

from book b,book\_authors b\_a,book\_copies b\_c, book\_loans b\_l

where b.title like '%Seafood Lovers Almanac%' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id and b\_l.book\_id = b.book\_id and b\_l.branch\_id=b\_c.branch\_id group by (b\_c.branch\_id and book\_id);

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c .no\_of\_copies,

(b\_c.no\_of\_copies - count(b\_l.book\_id)) as available\_books

from book b,book\_authors b\_a,book\_copies b\_c, book\_loans b\_l

where b\_a.author\_name like '%Mercedes Lee%' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id and b\_l.book\_id = b.book\_id and b\_l.branch\_id=b\_c.branch\_id group by (b\_c.branch\_id and book\_id);

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c .no\_of\_copies,

(b\_c.no\_of\_copies - count(b\_l.book\_id)) as available\_books

from book b,book\_authors b\_a,book\_copies b\_c, book\_loans b\_l

where b.book\_id = '967697603' and b.title = 'Seafood Lovers Almanac' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id and b\_l.book\_id = b.book\_id and b\_l.branch\_id=b\_c.branch\_id group by b\_c.branch\_id;

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c .no\_of\_copies,

(b\_c.no\_of\_copies - count(b\_l.book\_id)) as available\_books

from book b,book\_authors b\_a,book\_copies b\_c, book\_loans b\_l

where b.book\_id = '967697603' and b\_a.author\_name = 'Mercedes Lee' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id and b\_l.book\_id = b.book\_id and b\_l.branch\_id=b\_c.branch\_id group by b\_c.branch\_id;

select b.book\_id, b.title,b\_a. author\_name,b\_c. branch\_id,b\_c .no\_of\_copies,

(b\_c.no\_of\_copies - count(b\_l.book\_id)) as available\_books

from book b,book\_authors b\_a,book\_copies b\_c, book\_loans b\_l

where b.book\_id = '967697603' and b.title = 'Seafood Lovers Almanac' and b\_a.author\_name = 'Mercedes Lee' and b\_a.book\_id=b.book\_id and b\_c.book\_id = b.book\_id and b\_l.book\_id = b.book\_id and b\_l.branch\_id=b\_c.branch\_id group by b\_c.branch\_id;

BOOK LOANS

insert into book\_loans (book\_id,branch\_id, card\_no,curdate(), set duedate= curdate + interval 14 day)

select b\_l.book\_id, b\_l.branch\_id, bor.card\_no

from book b, library\_branch l\_b, book\_copies b\_c, borrower bor, book\_loans b\_l

where b\_c.book\_id = '201612585'and b\_c.branch\_id='4' and bor.card\_no=9002;

borrower management

insert into borrower (Fname,Lname,address = concat(address,",",city","state),concat("(",substring phone(0,3),substring phone(4,6),substring phone(6),")")

values('Archit','Garg','7825 Mccallum Blvd',' Dallas',' TX','(469-347-1427');

select count(\*) from borrower where Fname='Archit'and Lname='Garg' and Address='7825 Mccallum Blvd,Dallas,TX';

select exists (select fname, lname,address and count(\*) as cnt   
from borrower group by fname, lname , address  
having cnt > 1 order by cnt asc);

select exists (select fname, lname, address from borrower where fname,lname,address is not null)

//Imports are listed in full to show what's being used

**package** testPanel;

**import** java.awt.EventQueue;

**import** java.awt.BorderLayout;

**import** java.awt.CardLayout;

**import** java.awt.Color;

**import** javax.swing.BorderFactory;

**import** javax.swing.border.Border;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JComboBox;

**import** javax.swing.JButton;

**import** javax.swing.JLabel;

**import** javax.swing.SwingConstants;

**import** java.awt.Container;

**import** java.awt.event.ActionListener;

**import** java.awt.event.ActionEvent;

**import** javax.swing.BoxLayout;

**import** com.jgoodies.forms.layout.FormLayout;

**import** com.jgoodies.forms.layout.ColumnSpec;

**import** com.jgoodies.forms.layout.RowSpec;

**import** javax.swing.table.DefaultTableModel;

**import** java.awt.GridBagLayout;

**import** java.sql.\*;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTable;

**import** javax.swing.GroupLayout;

**import** javax.swing.GroupLayout.Alignment;

**import** javax.swing.JTextField;

**import** javax.swing.LayoutStyle.ComponentPlacement;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTextPane;

**import** javax.swing.ScrollPaneConstants;

**import** javax.swing.JScrollBar;

**import** javax.swing.JRadioButton;

**import** javax.swing.ButtonGroup;

**public** **class** testPanel {

JFrame guiFrame;

CardLayout cards;

JPanel cardPanel;

JButton BookSearch;

JPanel firstCard,secondCard,thirdCard;

**private** JTextField tbook\_id;

**private** JTextField t\_title;

**private** JTextField t\_author;

**private** JTextField textField\_3;

**private** JTextField textField\_4;

**private** JTextField textField\_6;

**private** JTextField textField\_7;

**private** JTextField textField\_8;

**private** JTextField textField\_11;

**private** JTextField textField\_12;

**private** **static** **int** *cno*=6;

**private** **static** String[] *cname*={"Book\_Id","Title","Author\_Names","Branch\_Id","No\_Of\_Copies","Available\_Copies"};

**private** **static** String[] *cname1*={"Loan\_Id","Book\_Id","Branch\_Id","card\_no","date\_out","due\_date","date\_in"};

JTextPane textPane1;

JTextPane textPane2;

JScrollPane scrollPane;

JScrollPane scrollPane\_0;

JScrollPane scrollPane\_1 ;

JTable table ;

JTable table1 ;

// JDBC driver name and database URL

**static** **final** String *JDBC\_DRIVER* = "com.mysql.jdbc.Driver";

**static** **final** String *DB\_URL* = "jdbc:mysql://localhost/Library";

// Database credentials

**static** **final** String *USER* = "root";

**static** **final** String *PASS* = "";

**private** JTextField textField;

**private** JTextField textField\_1;

**private** JTextField textField\_2;

**private** JTextField textField\_9;

**private** JTextField textField\_10;

**private** JTextField textField\_13;

**private** **final** ButtonGroup buttonGroup = **new** ButtonGroup();

**private** JTextField textField\_14;

**private** JTextField textField\_5;

**public** **static** **void** main(String[] args) {

//Use the event dispatch thread for Swing components

EventQueue.*invokeLater*(**new** Runnable()

{

@Override

**public** **void** run()

{

**new** testPanel();

}

});

}

**public** testPanel()

{

guiFrame = **new** JFrame();

//make sure the program exits when the frame closes

guiFrame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

guiFrame.setTitle("LIBRARY MANANGEMENT SYSTEM");

guiFrame.setSize(800,800);

//This will center the JFrame in the middle of the screen

guiFrame.setLocationRelativeTo(**null**);

guiFrame.getContentPane().setLayout(**new** BorderLayout());

//creating a border to highlight the JPanel areas

Border outline = BorderFactory.*createLineBorder*(Color.*black*);

JPanel tabsPanel = **new** JPanel();

tabsPanel.setBorder(outline);

BookSearch = **new** JButton("BOOK SEARCH");

BookSearch.setActionCommand("BOOK SEARCH");

BookSearch.addActionListener(**new** ActionListener()

{

@Override

**public** **void** actionPerformed(ActionEvent event)

{

cards.show(cardPanel, "BOOK SEARCH");

}

});

tabsPanel.add(BookSearch);

JButton BookLoans = **new** JButton("BOOK LOANS");

BookLoans.setActionCommand("BOOK LOANS");

BookLoans.addActionListener(**new** ActionListener()

{

@Override

**public** **void** actionPerformed(ActionEvent event)

{

cards.show(cardPanel, "BOOK LOANS");

}

});

tabsPanel.add(BookLoans);

JButton BorrowerManagement = **new** JButton("BORROWER MANAGEMENT");

BorrowerManagement.setActionCommand("Switch Card");

BorrowerManagement.addActionListener(**new** ActionListener()

{

@Override

**public** **void** actionPerformed(ActionEvent event)

{

cards.show(cardPanel,"BORROW MANAGEMENT");

}

});

tabsPanel.add(BorrowerManagement);

guiFrame.getContentPane().add(tabsPanel,BorderLayout.*NORTH*);

cards = **new** CardLayout();

cardPanel = **new** JPanel();

cardPanel.setLayout(cards);

cards.show(cardPanel, "Book Search");

firstCard = **new** JPanel();

firstCard.setBackground(Color.*ORANGE*);

secondCard = **new** JPanel();

secondCard.setBackground(Color.*ORANGE*);

thirdCard = **new** JPanel();

thirdCard.setBackground(Color.*ORANGE*);

cardPanel.add(firstCard, "BOOK SEARCH");

JLabel lblBookId = **new** JLabel("BOOK ID ");

lblBookId.setBounds(10, 22, 69, 14);

tbook\_id = **new** JTextField();

tbook\_id.setBounds(104, 19, 152, 20);

tbook\_id.setColumns(10);

JLabel lblTitle = **new** JLabel("TITLE ");

lblTitle.setBounds(383, 22, 54, 14);

JLabel lblAuthorName = **new** JLabel("AUTHOR NAME ");

lblAuthorName.setBounds(32, 156, 109, 14);

JButton btnSearch = **new** JButton("SEARCH");

btnSearch.setBounds(171, 318, 92, 23);

btnSearch.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

Connection conn = **null**;

PreparedStatement prepStmt = **null**;

ResultSet rs = **null**;

**try** {

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost/Library";

String connectionUser = "root";

String connectionPassword = "";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

String sqlStmt = "Select b.book\_id,b.title,ba.Author\_Name,bc.Branch\_id, bc.No\_of\_Copies,(bc.No\_Of\_Copies-(Count(bl.book\_id))) As Available\_Copies from book b, book\_authors ba, book\_copies bc, book\_loans bl where ba.author\_name like '%"+t\_author.getText()+"%' AND b.title like '%"+t\_title.getText()+"%' AND b.book\_id like '%"+tbook\_id.getText()+"%' AND b.book\_id=ba.book\_id AND b.book\_id=bc.book\_id AND b.book\_id=bl.book\_id AND bc.branch\_id=bl.branch\_id Group By bc.branch\_id, b.book\_id Union Select b.book\_id,b.title,ba.Author\_Name,bc.Branch\_id,bc.No\_of\_Copies,bc.No\_Of\_Copies As Available\_Copies from book b, book\_authors ba, book\_copies bc where ba.author\_name like '%"+t\_author.getText()+"%' AND b.title like '%"+t\_title.getText()+"%' AND b.book\_id like '%"+tbook\_id.getText()+"%' AND b.book\_id=ba.book\_id AND b.book\_id=bc.book\_id AND bc.branch\_id Not In (Select bl.branch\_id from book\_loans bl where bc.book\_id=bl.book\_id AND bc.branch\_id = bl.branch\_id) order by title,branch\_id;";

prepStmt = conn.prepareStatement(sqlStmt);

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt.toString());

rs = prepStmt.executeQuery();

DefaultTableModel tm=**new** DefaultTableModel();

**for**(**int** i=0;i<*cno*;i++)

{ tm.setColumnIdentifiers(*cname*); }

**int** count=0;

**while** (rs.next())

{ String[] rowData = **new** String[*cno*];

**for** (**int** i = 1; i <= *cno*; i++)

{ rowData[i - 1] = rs.getString(i);

} tm.addRow(rowData);

count++;

}

table.setModel(tm);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

JButton btnBack = **new** JButton("RESET");

btnBack.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e){

tbook\_id.setText("");

t\_title.setText("");

t\_author.setText("");

textField\_9.setText("");

textField\_10.setText("");

textField\_13.setText("");

//textfield.setText(null); //or use this

}

});

btnBack.setBounds(405, 318, 86, 23);

t\_title = **new** JTextField();

t\_title.setBounds(459, 19, 301, 20);

t\_title.setColumns(10);

t\_author = **new** JTextField();

t\_author.setBounds(151, 153, 109, 20);

t\_author.setColumns(10);

firstCard.setLayout(**null**);

firstCard.add(lblBookId);

firstCard.add(tbook\_id);

firstCard.add(lblTitle);

firstCard.add(t\_title);

firstCard.add(lblAuthorName);

firstCard.add(t\_author);

firstCard.add(btnSearch);

firstCard.add(btnBack);

JRadioButton rdbtnSearchBy = **new** JRadioButton("SEARCH BY ");

buttonGroup.add(rdbtnSearchBy);

rdbtnSearchBy.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

t\_author.setEditable(**true**);

}

});

rdbtnSearchBy.setBounds(10, 110, 109, 23);

firstCard.add(rdbtnSearchBy);

JRadioButton rdbtnNewRadioButton = **new** JRadioButton("SEARCH BY");

buttonGroup.add(rdbtnNewRadioButton);

rdbtnNewRadioButton.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

textField\_9.setEditable(**true**);

textField\_10.setEditable(**true**);

textField\_13.setEditable(**true**);

}

});

rdbtnNewRadioButton.setBounds(383, 110, 109, 23);

firstCard.add(rdbtnNewRadioButton);

JLabel lblFirstName = **new** JLabel("FIRST NAME");

lblFirstName.setBounds(405, 156, 69, 14);

firstCard.add(lblFirstName);

JLabel lblMiddleName = **new** JLabel("MIDDLE NAME");

lblMiddleName.setBounds(404, 191, 87, 14);

firstCard.add(lblMiddleName);

JLabel lblLastName = **new** JLabel("LAST NAME");

lblLastName.setBounds(405, 229, 69, 14);

firstCard.add(lblLastName);

textField\_9 = **new** JTextField();

textField\_9.setBounds(507, 153, 124, 20);

firstCard.add(textField\_9);

textField\_9.setColumns(10);

textField\_10 = **new** JTextField();

textField\_10.setBounds(507, 188, 124, 20);

firstCard.add(textField\_10);

textField\_10.setColumns(10);

textField\_13 = **new** JTextField();

textField\_13.setBounds(507, 226, 124, 20);

firstCard.add(textField\_13);

textField\_13.setColumns(10);

table = **new** JTable();

JScrollPane scrollPane = **new** JScrollPane(table);

scrollPane.setBounds(32, 373, 714, 314);

firstCard.add(scrollPane);

cardPanel.add(secondCard, "BOOK LOANS");

JLabel lblCheckingInBook = **new** JLabel("CHECK IN BOOK");

JLabel lblCheckingOutBook = **new** JLabel("CHECK OUT BOOK");

JLabel lblBookId\_1 = **new** JLabel("BOOK ID");

JLabel lblCardNo1 = **new** JLabel("CARD NO.");

textField\_3 = **new** JTextField();

textField\_3.setColumns(10);

textField\_4 = **new** JTextField();

textField\_4.setColumns(10);

JLabel lblBookId\_2 = **new** JLabel("BOOK ID");

JLabel lblBranchId2 = **new** JLabel("BRANCH ID");

JLabel lblCardNo2 = **new** JLabel("CARD NO.");

textField\_6 = **new** JTextField();

textField\_6.setColumns(10);

textField\_7 = **new** JTextField();

textField\_7.setColumns(10);

textField\_8 = **new** JTextField();

textField\_8.setColumns(10);

JButton btnReset = **new** JButton("RESET");

btnReset.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e){

textField\_3.setText("");

textField\_4.setText("");

textField\_5.setText("");

//textfield.setText(null); //or use this

}

});

JButton btnReset\_1 = **new** JButton("RESET");

btnReset\_1.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e){

textField\_6.setText("");

textField\_7.setText("");

textField\_8.setText("");

//textfield.setText(null); //or use this

}

});

JButton btnCheckIn = **new** JButton("SEARCH");

btnCheckIn.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

**try**{

Connection conn = **null**;

PreparedStatement prepStmt4 = **null**;

ResultSet rs4 = **null**;

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost/Library";

String connectionUser = "root";

String connectionPassword = "";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

String sqlStmt4 = "select b\_l.loan\_id, b\_l.book\_id, b\_l.branch\_id, b\_l.card\_no, b\_l.date\_out, b\_l.due\_date, b\_l.date\_in from book b,book\_loans b\_l, borrower bor where b\_l.book\_id=b.book\_id and b\_l.card\_no=bor.card\_no and bor.fname like '%"+textField\_14.getText()+"%' and bor.lname like '%"+textField\_5.getText()+"%' and b\_l.book\_id like '%"+textField\_3.getText()+"%' and b\_l.card\_no like '%"+textField\_4.getText()+"%' group by b\_l.book\_id,b\_l.card\_no,b\_l.branch\_id;";

prepStmt4 = conn.prepareStatement(sqlStmt4);

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt4.toString());

rs4 = prepStmt4.executeQuery();

DefaultTableModel tm=**new** DefaultTableModel();

**for**(**int** i=0;i<*cno*;i++)

{ tm.setColumnIdentifiers(*cname1*); }

**int** count=0;

**while** (rs4.next())

{ String[] rowData = **new** String[*cno*];

**for** (**int** i = 1; i <= *cno*; i++)

{ rowData[i - 1] = rs4.getString(i);

} tm.addRow(rowData);

count++;

}

table1.setModel(tm);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

JButton btnCheckOut = **new** JButton("CHECK OUT");

btnCheckOut.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

Connection conn = **null**;

PreparedStatement prepStmt5 = **null**;

PreparedStatement prepStmt6 = **null**;

PreparedStatement prepStmt7 = **null**;

PreparedStatement prepStmt8 = **null**;

PreparedStatement prepStmt9 = **null**;

PreparedStatement prepStmt10 = **null**;

ResultSet rs5,rs6,rs7,rs8,rs9,rs10= **null**;

**try** {

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost/Library";

String connectionUser = "root";

String connectionPassword = "";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

String sqlStmt5 = "insert into book\_loans (book\_id, branch\_id, card\_no, date\_out, due\_date, date\_in) values ('"+textField\_6.getText()+"','"+ textField\_8.getText()+"' ,"+ textField\_7.getText()+", curdate(), date\_add(curdate(),interval 14 day),null)";

String sqlStmt6 = "SELECT EXISTS(SELECT book\_id FROM book WHERE book\_id ='"+textField\_6.getText()+"') as count;";

String sqlStmt7 = "SELECT EXISTS(SELECT branch\_id FROM library\_branch WHERE branch\_id ='"+textField\_8.getText()+"') as count1;";

String sqlStmt8 = "select (b\_c.no\_of\_copies - count(b\_l.book\_id)) as available\_books from book\_loans b\_l, book\_copies b\_c, library\_branch l\_b, book b where b\_l.branch\_id=l\_b.branch\_id and b\_l.book\_id=b.book\_id and b\_c.book\_id=b.book\_id and b\_c.branch\_id=l\_b.branch\_id and b\_l.book\_id='"+textField\_6.getText()+"' and b\_l.branch\_id='"+textField\_8.getText()+"';";

String sqlStmt9 = "select count(card\_no) as maximum\_books from book\_loans where card\_no = "+textField\_7.getText()+";";

String sqlStmt10 = "SELECT EXISTS(SELECT card\_no FROM borrower WHERE card\_no ="+textField\_7.getText()+") as count2;";

prepStmt5 = conn.prepareStatement(sqlStmt5);

prepStmt6 = conn.prepareStatement(sqlStmt6);

prepStmt7 = conn.prepareStatement(sqlStmt7);

prepStmt8 = conn.prepareStatement(sqlStmt8);

prepStmt9 = conn.prepareStatement(sqlStmt9);

prepStmt10 = conn.prepareStatement(sqlStmt10);

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt5.toString());

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt6.toString());

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt7.toString());

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt8.toString());

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt9.toString());

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt10.toString());

rs6=prepStmt6.executeQuery();

rs7=prepStmt7.executeQuery();

rs8=prepStmt8.executeQuery();

rs9=prepStmt9.executeQuery();

rs10=prepStmt10.executeQuery();

rs6.next();

rs7.next();

rs8.next();

rs9.next();

rs10.next();

**if**(rs6.getInt("count")==0)

{ rs6.next();

JFrame error1 = **new** JFrame();

JOptionPane.*showMessageDialog*(error1, "The Book\_id you entered does not exist. Please enter again !!!");

}

**else** **if**(rs7.getInt("count1")==0)

{

rs7.next();

JFrame error2 = **new** JFrame();

JOptionPane.*showMessageDialog*(error2, "The Branch\_id you entered does not exist. Please enter again !!!");

}

**else** **if**(rs8.getInt("available\_books")<=0)

{

JFrame error3 = **new** JFrame();

JOptionPane.*showMessageDialog*(error3, " Book is not available in the branch . Sorry !!!");

}

**else** **if**(rs9.getInt("maximum\_books")>3)

{

JFrame error4 = **new** JFrame();

JOptionPane.*showMessageDialog*(error4, "You already have 3 books. Cannot issue more !!!");

}

**else** **if**(rs10.getInt("count2")==0)

{

JFrame error5 = **new** JFrame();

JOptionPane.*showMessageDialog*(error5, "Card No. does not exist. Please enter again !!!");

}

**else**

{

prepStmt5.executeUpdate();

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

table1 = **new** JTable();

// Enable row selection (default)

table1.setColumnSelectionAllowed(**false**);

table1.setRowSelectionAllowed(**true**);

scrollPane\_0 = **new** JScrollPane(table1);

JLabel lblFirstName\_1 = **new** JLabel("FIRST NAME");

textField\_14 = **new** JTextField();

textField\_14.setColumns(10);

JLabel label = **new** JLabel("");

JLabel lblLastName\_1 = **new** JLabel("LAST NAME");

textField\_5 = **new** JTextField();

textField\_5.setColumns(10);

JButton btnCheckIn\_1 = **new** JButton("CHECK IN");

btnCheckIn\_1.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**try**{

Connection conn = **null**;

PreparedStatement prepStmt4 = **null**;

PreparedStatement prepStmt5 = **null**;

ResultSet rs4,rs5 = **null**;

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost/Library";

String connectionUser = "root";

String connectionPassword = "";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

String sqlStmt4 = "select b\_l.loan\_id, b\_l.book\_id, b\_l.branch\_id, b\_l.card\_no, b\_l.date\_out, b\_l.due\_date, b\_l.date\_in from book b,book\_loans b\_l, borrower bor where b\_l.book\_id=b.book\_id and b\_l.card\_no=bor.card\_no and bor.fname like '%"+textField\_14.getText()+"%' and bor.lname like '%"+textField\_5.getText()+"%' and b\_l.book\_id like '%"+textField\_3.getText()+"%' and b\_l.card\_no like '%"+textField\_4.getText()+"%' group by b\_l.book\_id,b\_l.card\_no,b\_l.branch\_id;";

prepStmt4 = conn.prepareStatement(sqlStmt4);

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt4.toString());

rs4 = prepStmt4.executeQuery();

**int** sel\_rows[]= table1.getSelectedRows();

**for**(**int** j=0;j<sel\_rows.length;j++)

{

**int** rowid = sel\_rows[j];

String book\_id = table1.getModel().getValueAt(rowid,1).toString();

String branch\_id = table1.getModel().getValueAt(rowid,2).toString();

String card\_no = table1.getModel().getValueAt(rowid,3).toString();

System.*out*.println(book\_id);

System.*out*.println(branch\_id);

System.*out*.println(card\_no);

String sqlStmt5 = "update book\_loans set date\_in = curdate() where book\_id="+book\_id+" and branch\_id="+branch\_id+" and card\_no="+card\_no;

prepStmt5 = conn.prepareStatement(sqlStmt5);

prepStmt5.executeUpdate();

}

}

**catch** (Exception egdhncf) {

}

} });

GroupLayout gl\_secondCard = **new** GroupLayout(secondCard);

gl\_secondCard.setHorizontalGroup(

gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_secondCard.createSequentialGroup()

.addGap(21)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_secondCard.createSequentialGroup()

.addComponent(lblLastName\_1)

.addContainerGap())

.addGroup(gl\_secondCard.createSequentialGroup()

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*TRAILING*)

.addComponent(scrollPane\_0, Alignment.*LEADING*, GroupLayout.*DEFAULT\_SIZE*, 657, Short.*MAX\_VALUE*)

.addGroup(gl\_secondCard.createSequentialGroup()

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_secondCard.createSequentialGroup()

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_secondCard.createSequentialGroup()

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(lblBookId\_1)

.addComponent(lblCardNo1)

.addComponent(label))

.addGap(30)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(textField\_4, 105, 105, Short.*MAX\_VALUE*)

.addComponent(textField\_3, 105, 105, Short.*MAX\_VALUE*)

.addComponent(textField\_14, GroupLayout.*DEFAULT\_SIZE*, 105, Short.*MAX\_VALUE*)

.addComponent(textField\_5, GroupLayout.*DEFAULT\_SIZE*, 105, Short.*MAX\_VALUE*)))

.addComponent(lblCheckingInBook)

.addGroup(gl\_secondCard.createSequentialGroup()

.addComponent(btnReset)

.addGap(18)

.addComponent(btnCheckIn)

.addGap(18)

.addComponent(btnCheckIn\_1)))

.addGap(217))

.addGroup(gl\_secondCard.createSequentialGroup()

.addComponent(lblFirstName\_1, GroupLayout.*PREFERRED\_SIZE*, 73, GroupLayout.*PREFERRED\_SIZE*)

.addPreferredGap(ComponentPlacement.*RELATED*)))

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(lblCheckingOutBook)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*, **false**)

.addGroup(gl\_secondCard.createSequentialGroup()

.addComponent(btnReset\_1)

.addPreferredGap(ComponentPlacement.*RELATED*, GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)

.addComponent(btnCheckOut))

.addGroup(gl\_secondCard.createSequentialGroup()

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(lblBookId\_2)

.addComponent(lblBranchId2)

.addComponent(lblCardNo2))

.addGap(23)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*LEADING*, **false**)

.addComponent(textField\_6, GroupLayout.*DEFAULT\_SIZE*, 111, Short.*MAX\_VALUE*)

.addComponent(textField\_7)

.addComponent(textField\_8)))))))

.addGap(106))))

);

gl\_secondCard.setVerticalGroup(

gl\_secondCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_secondCard.createSequentialGroup()

.addGap(23)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblCheckingInBook)

.addComponent(lblCheckingOutBook))

.addGap(18)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblBookId\_1)

.addComponent(textField\_3, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(lblBookId\_2)

.addComponent(textField\_6, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(18)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblCardNo1)

.addComponent(textField\_4, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(lblCardNo2)

.addComponent(textField\_7, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(18)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(textField\_8, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addComponent(lblBranchId2)

.addComponent(label)

.addComponent(lblFirstName\_1)

.addComponent(textField\_14, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(18)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(lblLastName\_1)

.addComponent(textField\_5, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(26)

.addGroup(gl\_secondCard.createParallelGroup(Alignment.*BASELINE*)

.addComponent(btnReset)

.addComponent(btnReset\_1)

.addComponent(btnCheckOut)

.addComponent(btnCheckIn)

.addComponent(btnCheckIn\_1))

.addGap(34)

.addComponent(scrollPane\_0, GroupLayout.*DEFAULT\_SIZE*, 415, Short.*MAX\_VALUE*)

.addContainerGap())

);

secondCard.setLayout(gl\_secondCard);

cardPanel.add(thirdCard, "BORROW MANAGEMENT");

JLabel lblFname = **new** JLabel("FIRST NAME ");

textField\_11 = **new** JTextField();

textField\_11.setColumns(10);

JLabel lblLname = **new** JLabel("LAST NAME");

textField\_12 = **new** JTextField();

textField\_12.setColumns(10);

textField = **new** JTextField();

textField.setColumns(10);

textField\_1 = **new** JTextField();

textField\_1.setColumns(10);

JLabel lblAddress = **new** JLabel("ADDRESS");

JLabel lblCity = **new** JLabel("CITY ");

JLabel lblState = **new** JLabel("STATE");

textField\_2 = **new** JTextField();

textField\_2.setColumns(10);

JButton btnAddBorrower = **new** JButton("ADD BORROWER");

btnAddBorrower.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent arg0) {

Connection conn = **null**;

PreparedStatement prepStmt11 = **null**;

PreparedStatement prepStmt12 = **null**;

PreparedStatement prepStmt13= **null**;

PreparedStatement prepStmt14= **null**;

PreparedStatement prepStmt15= **null**;

ResultSet rs11,rs12,rs13,rs14,rs15 = **null**;

**try** {

Class.*forName*("com.mysql.jdbc.Driver").newInstance();

String connectionUrl = "jdbc:mysql://localhost/Library";

String connectionUser = "root";

String connectionPassword = "";

conn = DriverManager.*getConnection*(connectionUrl, connectionUser, connectionPassword);

String sqlStmt11 = "SELECT \* FROM LIBRARY.book ";

String sqlStmt12 = "SELECT \* FROM LIBRARY.book where title = ?;";

String sqlStmt13 = "SELECT \* FROM LIBRARY.book\_authors where author\_name = ?;";

String sqlStmt14 = "SELECT \* FROM LIBRARY.book where title = ?;";

String sqlStmt15 = "SELECT \* FROM LIBRARY.book where title = ?;";

prepStmt11 = conn.prepareStatement(sqlStmt11);

prepStmt12 = conn.prepareStatement(sqlStmt12);

prepStmt13 = conn.prepareStatement(sqlStmt13);

prepStmt14 = conn.prepareStatement(sqlStmt14);

prepStmt15 = conn.prepareStatement(sqlStmt15);

//prepStmt11.setString(1,tbook\_id.getText());

prepStmt12.setString(1, t\_title.getText());

prepStmt13.setString(1, t\_author.getText());

prepStmt14.setString(1, t\_author.getText());

prepStmt15.setString(1, t\_author.getText());

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt11.toString());

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt12.toString());

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt13.toString());

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt14.toString());

System.*out*.println("Prepared Statement after bind variables set:\n\t" + prepStmt15.toString());

rs11 = prepStmt11.executeQuery();

rs12 = prepStmt12.executeQuery();

rs13 = prepStmt13.executeQuery();

rs14 = prepStmt14.executeQuery();

rs15 = prepStmt15.executeQuery();

String ans="";

**while** (rs11.next()) {

ans=ans.concat("\n"+rs11.getString("book\_id")+"\t"+rs11.getString("title"));

}

textPane2.setText(ans);

**while** (rs12.next()) {

ans=ans.concat("\n"+rs12.getString("book\_id")+"\t"+rs12.getString("title"));

}

textPane2.setText(ans);

**while** (rs13.next()) {

ans=ans.concat("\n"+rs13.getString("book\_id")+"\t"+rs13.getString("author\_name"));

}

textPane2.setText(ans);

**while** (rs14.next()) {

ans=ans.concat("\n"+rs14.getString("book\_id")+"\t"+rs14.getString("title"));

}

textPane2.setText(ans);

**while** (rs15.next()) {

ans=ans.concat("\n"+rs15.getString("book\_id")+"\t"+rs15.getString("title"));

}

textPane2.setText(ans);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

textPane2 = **new** JTextPane();

scrollPane\_1 = **new** JScrollPane(textPane2);

scrollPane\_1.setVerticalScrollBarPolicy(ScrollPaneConstants.*VERTICAL\_SCROLLBAR\_ALWAYS*);

JButton btnReset\_2 = **new** JButton("RESET");

btnReset\_2.addActionListener(**new** ActionListener(){

**public** **void** actionPerformed(ActionEvent e){

textField.setText("");

textField\_1.setText("");

textField\_2.setText("");

textField\_11.setText("");

textField\_12.setText("");

textPane2.setText("");

//textfield.setText(null); //or use this

}

});

GroupLayout gl\_thirdCard = **new** GroupLayout(thirdCard);

gl\_thirdCard.setHorizontalGroup(

gl\_thirdCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_thirdCard.createSequentialGroup()

.addGap(28)

.addGroup(gl\_thirdCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(scrollPane\_1, GroupLayout.*PREFERRED\_SIZE*, 645, GroupLayout.*PREFERRED\_SIZE*)

.addGroup(gl\_thirdCard.createSequentialGroup()

.addComponent(lblFname, GroupLayout.*PREFERRED\_SIZE*, 75, GroupLayout.*PREFERRED\_SIZE*)

.addGap(32)

.addComponent(textField\_11, GroupLayout.*PREFERRED\_SIZE*, 195, GroupLayout.*PREFERRED\_SIZE*)

.addGap(90)

.addComponent(lblLname, GroupLayout.*PREFERRED\_SIZE*, 75, GroupLayout.*PREFERRED\_SIZE*)

.addGap(28)

.addComponent(textField\_12, GroupLayout.*PREFERRED\_SIZE*, 188, GroupLayout.*PREFERRED\_SIZE*))

.addGroup(gl\_thirdCard.createSequentialGroup()

.addComponent(lblAddress, GroupLayout.*PREFERRED\_SIZE*, 63, GroupLayout.*PREFERRED\_SIZE*)

.addGap(44)

.addComponent(textField, GroupLayout.*PREFERRED\_SIZE*, 301, GroupLayout.*PREFERRED\_SIZE*))

.addGroup(gl\_thirdCard.createSequentialGroup()

.addComponent(lblCity, GroupLayout.*PREFERRED\_SIZE*, 45, GroupLayout.*PREFERRED\_SIZE*)

.addGap(62)

.addComponent(textField\_1, GroupLayout.*PREFERRED\_SIZE*, 118, GroupLayout.*PREFERRED\_SIZE*))

.addGroup(gl\_thirdCard.createSequentialGroup()

.addComponent(lblState, GroupLayout.*PREFERRED\_SIZE*, 45, GroupLayout.*PREFERRED\_SIZE*)

.addGap(62)

.addComponent(textField\_2, GroupLayout.*PREFERRED\_SIZE*, 118, GroupLayout.*PREFERRED\_SIZE*))

.addGroup(gl\_thirdCard.createSequentialGroup()

.addComponent(btnAddBorrower, GroupLayout.*PREFERRED\_SIZE*, 148, GroupLayout.*PREFERRED\_SIZE*)

.addGap(62)

.addComponent(btnReset\_2, GroupLayout.*PREFERRED\_SIZE*, 92, GroupLayout.*PREFERRED\_SIZE*)))

.addGap(73))

);

gl\_thirdCard.setVerticalGroup(

gl\_thirdCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_thirdCard.createSequentialGroup()

.addGap(36)

.addGroup(gl\_thirdCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_thirdCard.createSequentialGroup()

.addGap(3)

.addComponent(lblFname))

.addComponent(textField\_11, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)

.addGroup(gl\_thirdCard.createSequentialGroup()

.addGap(3)

.addComponent(lblLname))

.addComponent(textField\_12, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(28)

.addGroup(gl\_thirdCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_thirdCard.createSequentialGroup()

.addGap(3)

.addComponent(lblAddress))

.addComponent(textField, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(30)

.addGroup(gl\_thirdCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_thirdCard.createSequentialGroup()

.addGap(3)

.addComponent(lblCity))

.addComponent(textField\_1, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(26)

.addGroup(gl\_thirdCard.createParallelGroup(Alignment.*LEADING*)

.addGroup(gl\_thirdCard.createSequentialGroup()

.addGap(3)

.addComponent(lblState))

.addComponent(textField\_2, GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*))

.addGap(39)

.addGroup(gl\_thirdCard.createParallelGroup(Alignment.*LEADING*)

.addComponent(btnAddBorrower)

.addComponent(btnReset\_2))

.addGap(30)

.addComponent(scrollPane\_1, GroupLayout.*PREFERRED\_SIZE*, 395, GroupLayout.*PREFERRED\_SIZE*)

.addContainerGap())

);

thirdCard.setLayout(gl\_thirdCard);

guiFrame.getContentPane().add(tabsPanel,BorderLayout.*NORTH*);

guiFrame.getContentPane().add(cardPanel,BorderLayout.*CENTER*);

guiFrame.setVisible(**true**);

}

}

INSERT INTO fines

SELECT b\_l.loan\_id, 0.25\*(datediff(curdate() ,b\_l.due\_date)),0

FROM book\_loans b\_l

WHERE (datediff(curdate(),b\_l.date\_out)) > 14 and date\_in is null

on duplicate key update fine\_amt=0.25\*(datediff(curdate(),b\_l.due\_date));

INSERT INTO fines SELECT b\_l.loan\_id, (0.25\*datediff(b\_l.date\_in ,b\_l.due\_date)),0

FROM book\_loans b\_l

WHERE (datediff(b\_l.date\_in,b\_l.date\_out))>14

on duplicate key update fine\_amt=0.25\*(datediff(b\_l.date\_in,b\_l.due\_date));

update fines

set paid=1

update fines f, book\_loans b\_l

set f.paid = 1

where b\_l.loan\_id = f.loan\_id and b\_l.date\_in is not null

select sum(f.fine\_amt)

from fines f, book\_loans b\_l

where f.loan\_id= b\_l.loan\_id

group by paid, b\_l.card\_no