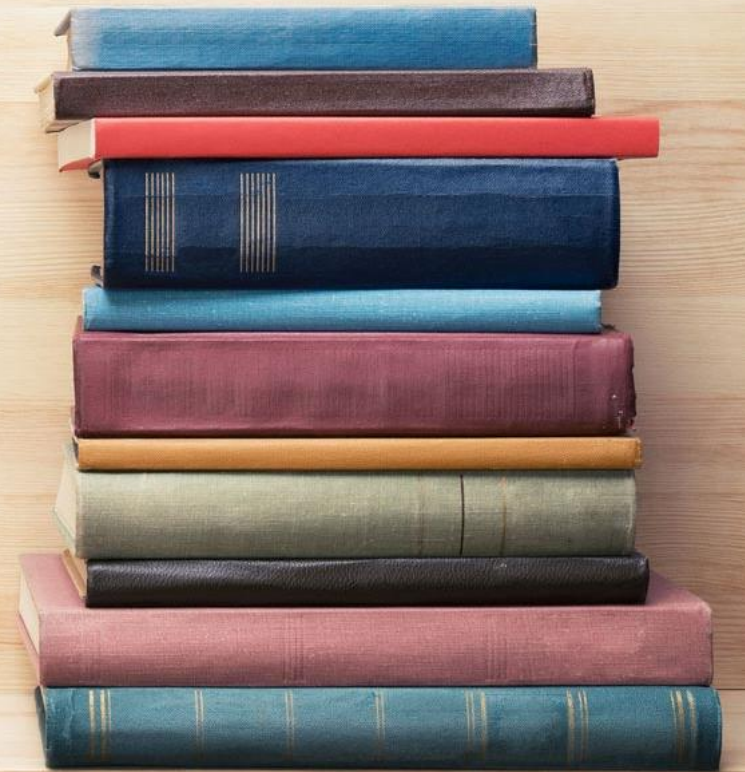


# Borrow And Finding book System



# บทคัดย่อ

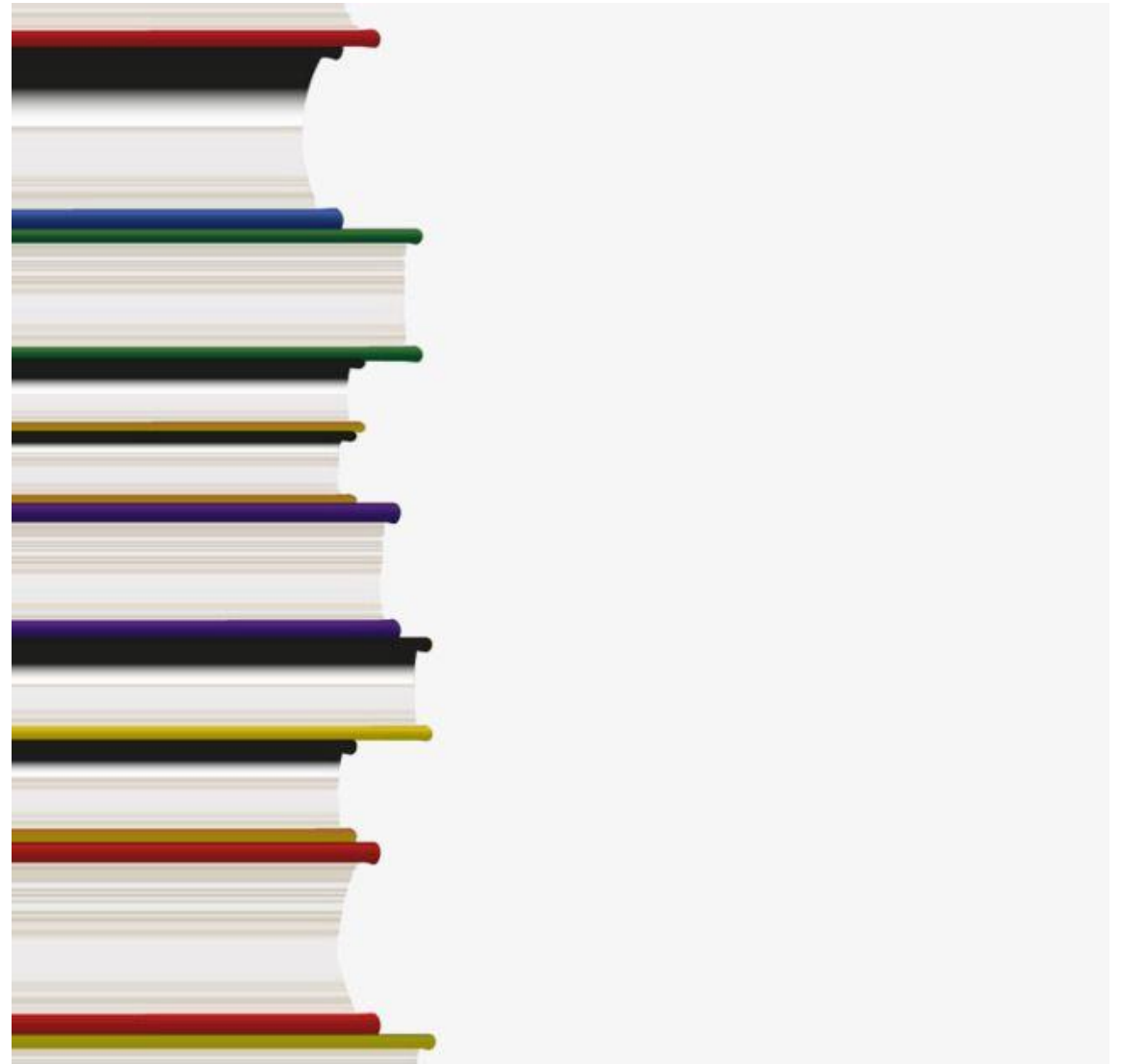
ชื่อโครงการ: Borrow&Finding Book System

โปรแกรม Library System นี้มีจุดประสงค์ทำขึ้นเพื่อเป็นโปรแกรมเพื่อการใช้งานภายในห้องสมุด เกี่ยวกับการยืม-คืน และเช็คสถานะของหนังสือเป็นต้น โดยตัวโปรแกรมจะมีการ login เข้าสู่ระบบ ในฝั่งของ Admin และ User เพื่อแบ่งการอนุญาตการเข้าถึงของระบบ โดยฝั่ง User จะสามารถค้นหาหนังสือ เช็ครายการที่ยืมและประวัติในการยืม-คืนหนังสือ ส่วนในฝั่งของ Admin สามารถค้นหาหนังสือ ทำรายการยืม-คืน เช็คประวัติการยืม-คืนหนังสือ และลงทะเบียนหนังสือหรือลบหนังสือนั้นออกจากระบบ โดยระบบการจัดการทั้งหมดจะอยู่ในฝั่งของ Admin

# วัตถุประสงค์ (objective)

---

- เพื่อศึกษารูปแบบการทำ GUI
- เพื่อศึกษาการสร้างฐานข้อมูล
- เพื่อศึกษาวิธีการเชื่อมต่อฐานข้อมูลจาก GUI
- เพื่อฝึกการทำงานร่วมกัน
- เพื่อศึกษาการแสดงผลข้อมูลโดยดึงข้อมูลจากฐานข้อมูล(localhost)



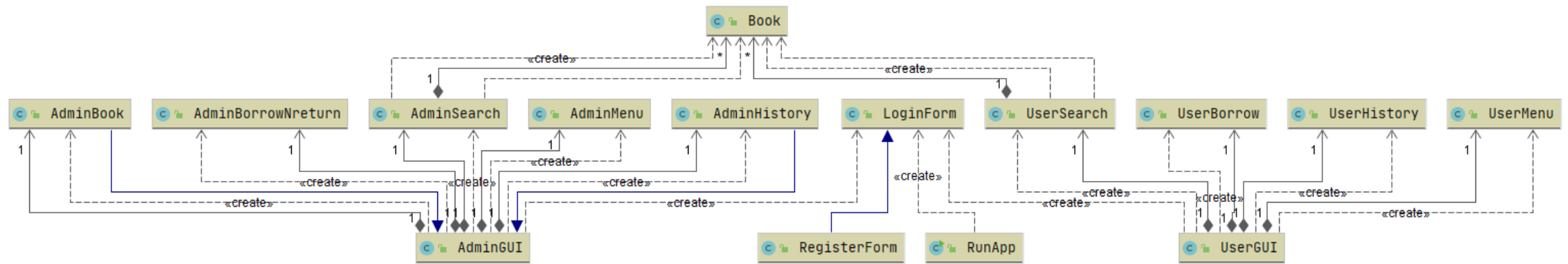
# ประโยชน์ที่ได้รับ

---

- ผู้ใช้สามารถค้นหาหนังสือที่มีภายในห้องสมุดพร้อมกับเช็คสถานะของหนังสือเล่มนั้น
- ผู้ใช้สามารถเช็คประวัติการยืม-คืนหนังสือและรายการที่ของหนังสือที่กำลังยืมได้ด้วยตนเอง
- ผู้ดูแลสามารถจัดการหนังสือด้วยระบบต่างๆ อย่างเข้าใจง่าย
- มีฐานข้อมูลที่จัดการได้ง่าย



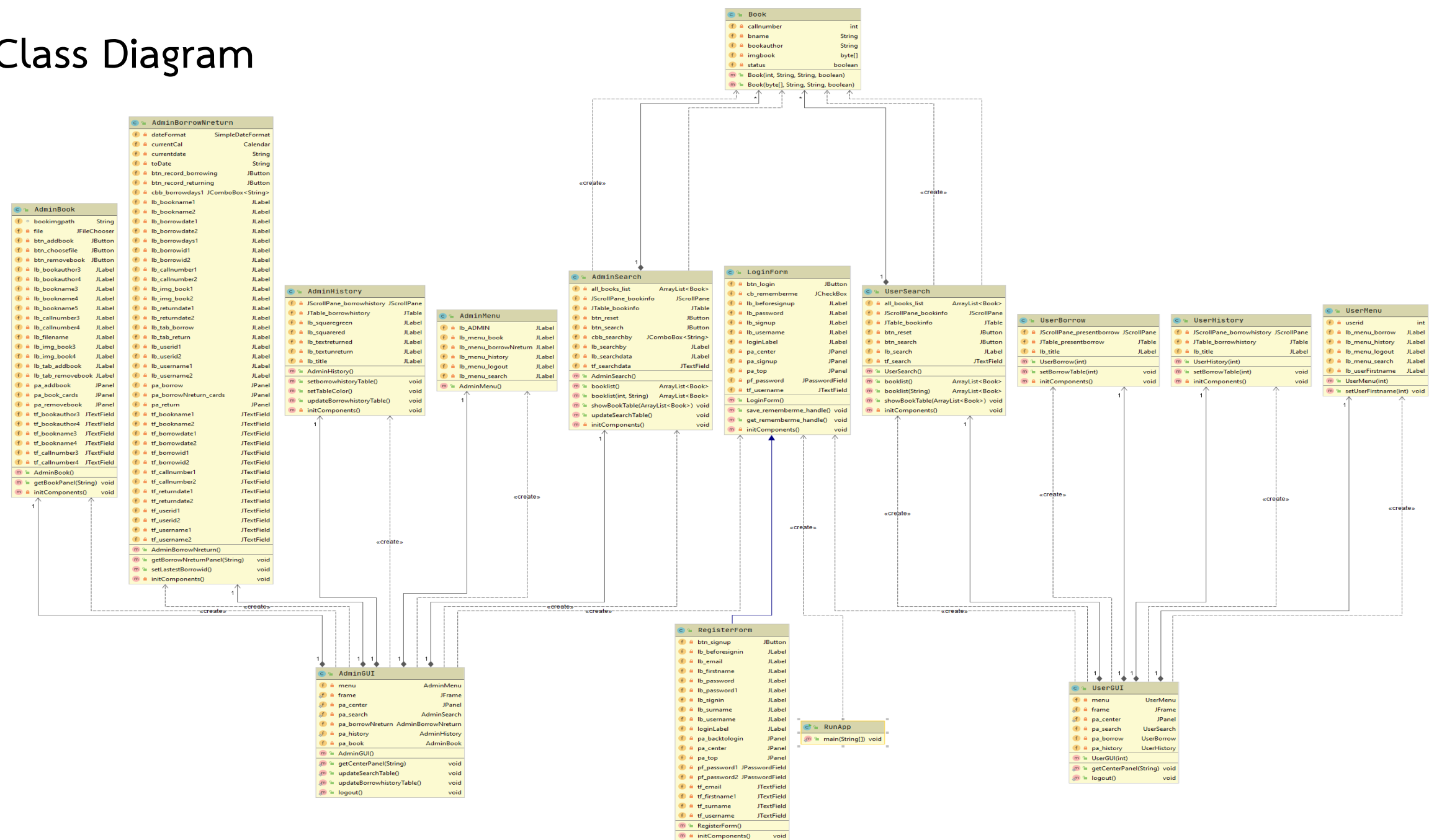
# Class Diagram



Connect

Powered by yFiles

# Class Diagram







## โค้ดและคำอธิบายโค้ด

### Class runApp

```
public class runApp {  
    public static void main(String[] args) {  
        try {  
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {  
                if ("Nimbus".equals(info.getName())) {  
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());  
                    break;  
                }  
            }  
        } catch (Exception ex) {  
            System.out.println(ex);  
        }  
        new LoginForm();  
    }  
}
```

Class นี้จะ set Look and Feel เป็น Nimbus และสร้าง LoginForm ขึ้นมา



# โค้ดและคำอธิบายโค้ด

## Class Connect

```
public class Connect {  
  
    public static Connection connectDB() {  
        String db_name = "mdb";  
        String user = "root";  
        String pass = "";  
        String hostName = "localhost";  
        String driverName = "com.mysql.jdbc.Driver";  
        try {  
            Class.forName(driverName);  
            String url = "jdbc:mysql://" + hostName + "/" + db_name + "?useUnicode=true&characterEncoding=utf-8";  
            Connection con = DriverManager.getConnection(url, user, pass);  
            System.out.println("Successfully connected to the database.");  
            return con;  
        } catch (Exception e) {  
            System.out.println("Failed to connect to the database.");  
        }  
        return null;  
    }  
}
```

- Class ที่ใช้ในการเชื่อมฐานข้อมูล โดยระบุ hostname (localhost) ของเรา ที่ใช้เก็บฐานข้อมูล (mysql) โดยทุกๆ การใช้ฐานข้อมูลภายในโปรแกรมจะต้องมีการเรียก Static method ของ Class นี้เพื่อเชื่อมต่อฐานข้อมูลและเรียกใช้ข้อมูลเพื่อแสดงผล

The screenshot shows the phpMyAdmin interface. On the left, the database 'mdb' is selected, showing its structure with tables 'bookinfo', 'borrowhistory', and 'userinfo'. On the right, the 'Structure' tab is active, displaying a table with columns: Table, Action, Rows, Type, Collation, Size, and Overhead. The table lists the three tables in the database.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> bookinfo	★ Browse Structure Search Insert Empty Drop	100	InnoDB	utf8_unicode_ci	9.5 MiB	-
<input type="checkbox"/> borrowhistory	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_unicode_ci	16.0 KiB	-
<input type="checkbox"/> userinfo	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_unicode_ci	16.0 KiB	-
3 tables	Sum	102	InnoDB	utf8_unicode_ci	9.5 MiB	0 B





โค้ดและคำอธิบายโค้ด

## Class LoginForm

```
private void lb_signupMouseClicked(java.awt.event.MouseEvent evt) {  
    new RegisterForm();  
    this.dispose();  
}
```

```
private void formWindowClosing(java.awt.event.WindowEvent evt) {  
    save_rememberme_handle();  
}
```

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {  
    get_rememberme_handle();  
}
```

- ไปยังหน้า Register เมื่อคลิกที่ Sign up here!
- มีการจัดการ RememberMe เมื่อหน้าต่างนี้ถูกเปิด/ปิด

```
public void save_rememberme_handle(){  
    try (FileOutputStream fout = new FileOutputStream("rememberMe.dat");  
        DataOutputStream dout = new DataOutputStream(fout);) {  
        if (cb_rememberme.isSelected()) {  
            dout.writeUTF("remember=" + tf_username.getText() + "=" + pf_password.getText());  
        } else {  
            dout.writeUTF("");  
        }  
    } catch (IOException ex) {  
        ex.printStackTrace();  
    }  
}
```

```
public void get_rememberme_handle(){  
    File f = new File("rememberMe.dat");  
    if (f.exists()) {  
        try (FileInputStream fin = new FileInputStream(f);  
            DataInputStream din = new DataInputStream(fin);) {  
            String[] text_remember = din.readUTF().split("=");  
            if (text_remember[0].equals("remember"))  
                cb_rememberme.setSelected(true);  
  
            tf_username.setText(text_remember[1]);  
            pf_password.setText(text_remember[2]);  
  
            System.out.println("getRememberMe: username=" + text_remember[1] + ", password=" + text_remember[2]);  
        } catch (IOException ex) {  
            ex.printStackTrace();  
        } catch (ArrayIndexOutOfBoundsException ex) {  
            System.out.println("No remembered username or password");  
        }  
    }  
}
```



# ฐานข้อมูล

## SQL

```
"Select * From userinfo where user=? and pass=?";
```

- เชื่อมต่อกับฐานข้อมูลและดึงข้อมูลผู้ใช้ในฐานข้อมูล(userinfo) เพื่อเทียบค่า user & pass ว่ามีในฐานข้อมูลไหม ถ้ามีก็จำเข้าระบบฝั่ง User

## ตัวอย่าง

```
"INSERT INTO `userinfo`(`fname`, `sname`, `email`, `user`, `pass`) VALUES (?, ?, ?, ?, ?)";
```

- เพิ่มข้อมูลที่อยู่ในฐานข้อมูล ตามลักษณะของฐานข้อมูลที่สร้าง ในรูปตัวอย่างคือ การ Register เพิ่มข้อมูลผู้ใช้งานฐานข้อมูล

```
String update = "UPDATE `bookinfo` SET `status`= 0 WHERE callnumber=?";
```

- ทำการ เปลี่ยนแปลงค่า Status ที่อยู่ใน table bookinfo โดยเช็ค callnumber เพื่อเปลี่ยน ค่า Status แลวนั้น



ฐานข้อมูล

SQL

ตัวอย่าง

```
"DELETE FROM `bookinfo` WHERE callnumber=?";
```

- เชื่อมต่อกับฐานข้อมูล(bookinfo) และเช็ค callnumber แล้วทำการลบออกจาก table ของ bookinfo

```
String query = "SELECT * FROM borrowhistory "  
+ "INNER JOIN bookinfo ON borrowhistory.callnumber = bookinfo.callnumber "  
+ "INNER JOIN uesrinfo ON borrowhistory.userid = uesrinfo.userid "  
+ "WHERE callnumber=? AND returned=0";
```

- การเชื่อม table ใน ฐานข้อมูลเข้าด้วยกันโดยการกำหนด Key เพื่อเทียบกันในการเรียก



โค้ดและคำอธิบายโค้ด

## Class LoginForm

Username : admin

Password : \*\*\*\*\*

☒ Remember me

Login

Don't have an account? [Sign up here!](#)

เมื่อคลิก **Login** จะตรวจสอบว่าเป็น Admin หรือ User

ถ้าพบว่าเป็น Admin -> สร้าง AdminGUI

ถ้าพบว่าเป็น User (เช็คโดยใช้ข้อมูลใน Database) -> สร้าง UserGUI

พร้อมกับใส่ argument userid

```
private void btn_loginActionPerformed(java.awt.event.ActionEvent evt) {  
    String username = tf_username.getText().trim();  
    String password = pf_password.getText().trim();  
    if (username.equals(username_admin) && password.equals(password_admin)) {  
        save_rememberme_handle();  
        System.out.println("Welcome Admin");  
        new AdminGUI();  
        this.dispose();  
    } else {  
        try {  
            Connection con1 = Connect.connectDB();  
            String sql = "Select * From userinfo where user=? and pass=?";  
            PreparedStatement pst = con1.prepareStatement(sql);  
            pst.setString(1, tf_username.getText());  
            pst.setString(2, pf_password.getText());  
            ResultSet rs = pst.executeQuery();  
            if (rs.next()) {  
                save_rememberme_handle();  
                System.out.println(rs.getInt("userid"));  
                new UserGUI(rs.getInt("userid"));  
                this.dispose();  
            } else {  
                JOptionPane.showMessageDialog(null, "Invalid user...");  
                tf_username.setText("");  
                pf_password.setText("");  
            }  
        } catch (Exception ex) {  
            JOptionPane.showMessageDialog(null, ex);  
        }  
    }  
}
```

Admin

User



# โค้ดและคำอธิบายโค้ด

## Class RegisterForm

Library System

### Account Register

Firstname :

Surname :

Email :

Username :

Password :

Confirm Password :

Already have an account?

- เมื่อกรอกข้อมูลและกดปุ่ม Sign up จะนำข้อมูลใน textfield ต่างๆ ไป insert ลงใน Database
- ไปยังหน้า Login เมื่อคลิกที่ Sign in

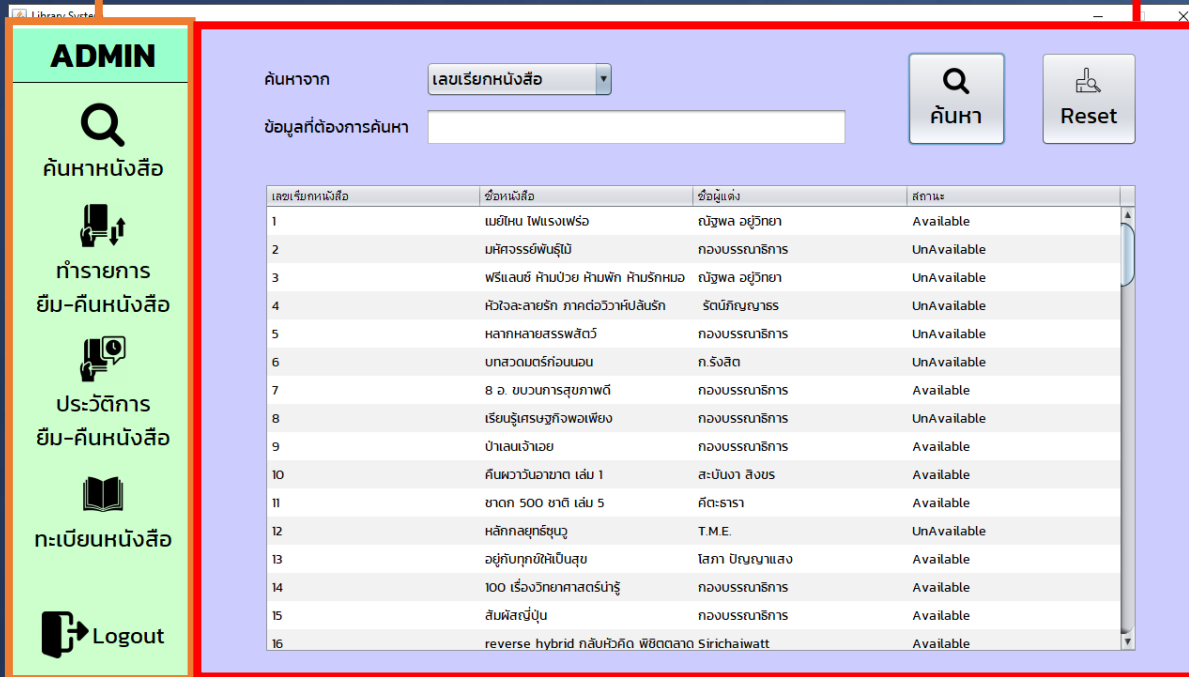
```
private void btn_signupActionPerformed(java.awt.event.ActionEvent evt) {  
    if (pf_password1.getText().equals(pf_password2.getText())) {  
        try {  
            Connection con1 = Connect.connectDB();  
            String query = "INSERT INTO `userinfo` (`fname`, `sname`, `email`, `user`, `pass`) VALUES (?, ?, ?, ?, ?)";  
            PreparedStatement pst = con1.prepareStatement(query);  
            pst.setString(1, tf_firstname1.getText());  
            pst.setString(2, tf_surname.getText());  
            pst.setString(3, tf_email.getText());  
            pst.setString(4, tf_username.getText());  
            pst.setString(5, pf_password1.getText());  
            pst.executeUpdate();  
            JOptionPane.showMessageDialog(null, "Success");  
        } catch (Exception ex) {  
            JOptionPane.showMessageDialog(null, ex);  
        }  
    } else {  
        JOptionPane.showMessageDialog(null, "Password and confirm password do not match.");  
    }  
}
```

```
private void lb_signinMouseClicked(java.awt.event.MouseEvent evt) {  
    new LoginForm();  
    this.dispose();  
}
```



# โค้ดและคำอธิบายโค้ด

## Class AdminGUI



```

public class AdminGUI {
    private AdminMenu menu;
    private static JFrame frame;
    private static JPanel pa_center;
    private static AdminSearch pa_search;
    private AdminBorrowNreturn pa_borrowNreturn;
    private static AdminHistory pa_history;
    private AdminBook pa_book;

    public AdminGUI() {
        frame = new JFrame("Library System");
        menu = new AdminMenu();
        menu.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 1, Color.black));

        pa_center = new JPanel();
        pa_search = new AdminSearch();
        pa_borrowNreturn = new AdminBorrowNreturn();
        pa_history = new AdminHistory();
        pa_book = new AdminBook();

        pa_center.setLayout(new CardLayout());
        pa_center.add(pa_search, "panel_search");
        pa_center.add(pa_borrowNreturn, "panel_borrowNreturn");
        pa_center.add(pa_history, "panel_history");
        pa_center.add(pa_book, "panel_book");

        frame.add(menu, BorderLayout.WEST);
        frame.add(pa_center, BorderLayout.CENTER);

        frame.setSize(1280, 720);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

```

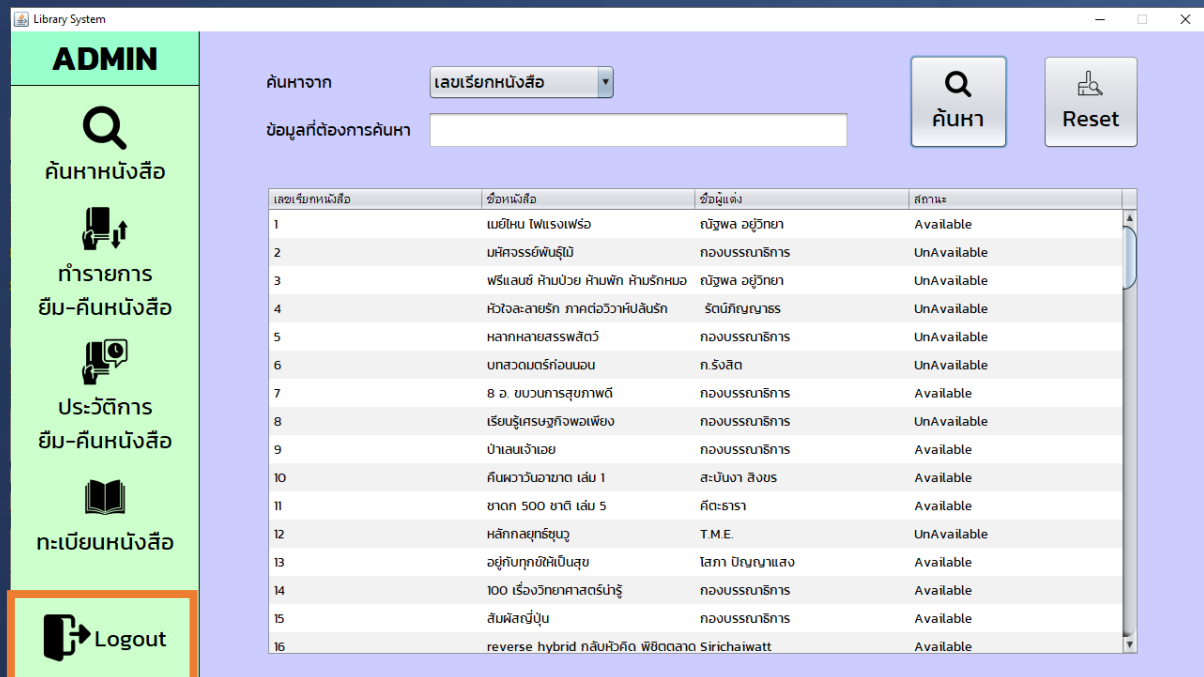
เป็น Class ที่นำแต่ละ Panel มารวมกัน (รวมไว้ใน pa\_center โดยใช้ CardLayout) และจัดวาง GUI ให้ออกมาดังภาพ





# โค้ดและคำอธิบายโค้ด

## Class AdminGUI



```
public static void logout() {
    frame.dispose();
    new LoginForm();
}
```

Method นี้จะถูกเรียกใช้เมื่อกดปุ่ม Logout

```
public static void getCenterPanel(String panel_name){
    CardLayout card = (CardLayout)(pa_center.getLayout());
    card.show(pa_center, panel_name);
    System.out.println("Change to " + panel_name);
}
```

Method นี้จะถูกเรียกใช้เมื่อกดเมนูต่างๆ

```
public static void updateSearchTable() {
    pa_search.updateSearchTable();
    System.out.println("Update Search Table");
}

public static void updateBorrowhistoryTable() {
    pa_history.updateBorrowhistoryTable();
    System.out.println("Update Borrowhistory Table");
}
```

2 Method นี้จะถูกเรียกใช้เมื่อ  
ทำรายการยืม-คืนหนังสือ หรือ เพิ่ม/ลบข้อมูลหนังสือ  
โดยจะไปเรียกใช้ method ของ class ที่มี Table นั้น



# โค้ดและคำอธิบายโค้ด

## Class AdminMenu

### ADMIN



ค้นหาหนังสือ



ทำรายการ  
ยืม-คืนหนังสือ



ประวัติการ  
ยืม-คืนหนังสือ



ทะเบียนหนังสือ



Logout

```

public class AdminMenu extends javax.swing.JPanel {

    public AdminMenu() {
        initComponents();

        ImageIcon icon_search = new ImageIcon(new ImageIcon(getClass().getResource("../image/icon_search.png")).getImage().getScaledInstance(50, 50, Image.SCALE_SMOOTH));
        ImageIcon icon_borrowNreturn = new ImageIcon(new ImageIcon(getClass().getResource("../image/icon_borrow.png")).getImage().getScaledInstance(50, 50, Image.SCALE_SMOOTH));
        ImageIcon icon_borrowhistory = new ImageIcon(new ImageIcon(getClass().getResource("../image/icon_borrowhistory.png")).getImage().getScaledInstance(50, 50, Image.SCALE_SMOOTH));
        ImageIcon icon_book = new ImageIcon(new ImageIcon(getClass().getResource("../image/icon_book.png")).getImage().getScaledInstance(50, 50, Image.SCALE_SMOOTH));
        ImageIcon icon_logout = new ImageIcon(new ImageIcon(getClass().getResource("../image/icon_logout.png")).getImage().getScaledInstance(50, 50, Image.SCALE_SMOOTH));

        lb_menu_search.setIcon(icon_search);
        lb_menu_borrowNreturn.setIcon(icon_borrowNreturn);
        lb_menu_history.setIcon(icon_borrowhistory);
        lb_menu_book.setIcon(icon_book);
        lb_menu_logout.setIcon(icon_logout);
    }
}

```

สร้าง ImageIcon ขนาด 50 x 50 และนำไป Set ใน Label ของ Menu ต่างๆ

```

public void lb_menu_mouseentered(MouseEvent evt){
    ((JLabel) evt.getSource()).setForeground(Color.red);
}

public void lb_menu_mouseexited(MouseEvent evt){
    ((JLabel) evt.getSource()).setForeground(Color.black);
}

```

เมื่อเกิด MouseEvent จะส่งมาจัดการที่ methods นี้

```

private void lb_menu_searchMouseClicked(java.awt.event.MouseEvent evt) {
    AdminGUI.getCenterPanel("panel_search");
}

private void lb_menu_borrowNreturnMouseClicked(java.awt.event.MouseEvent evt) {
    AdminGUI.getCenterPanel("panel_borrowNreturn");
}

private void lb_menu_historyMouseClicked(java.awt.event.MouseEvent evt) {
    AdminGUI.getCenterPanel("panel_history");
}

```

เมื่อกดเมนูจะเรียก method ในการสลับ panel จาก class AdminGUI



# โค้ดและคำอธิบายโค้ด

## Class AdminSearch

ค้นหาจาก

ข้อมูลที่ต้องการค้นหา

เลขเรียกหนังสือ	ชื่อหนังสือ	ชื่อผู้แต่ง	สถานะ
1	เมย์ไหน ไพล่องเพร่อ	ณัฐพล อยู่วิทยา	Available
2	มหัศจรรย์พันธุ์ไม้	ทองบรรณารักษ์	UnAvailable
3	ฟรีแลนซ์ หันป่วย หันพัก หันรักหมอ	ณัฐพล อยู่วิทยา	UnAvailable
4	หัวใจละลายรัก ภาคต่อวิวาห์สิ้นรัก	รัตติกาลญารส	UnAvailable
5	หลากหลายสรรพสัตว์	ทองบรรณารักษ์	UnAvailable
6	บทสวนนตรก่อนนอน	ก. รังสิต	UnAvailable
7	8 อ. ขบวนการสุขภาพดี	ทองบรรณารักษ์	Available
8	เรียนรู้เศรษฐกิจพอเพียง	ทองบรรณารักษ์	UnAvailable
9	ป่าเลนเจ้าเฉย	ทองบรรณารักษ์	Available
10	คืนทวารีนายาต เล่ม 1	สะบิงงา สิงขร	Available
11	ซากด 500 ซาก เล่ม 5	ศิดะธรา	Available
12	หลักกลยุทธ์บุญ	T.M.E.	UnAvailable
13	อยู่กับทุกข์ให้เป็นสุข	โสภา ปิณฑาแสง	Available
14	100 เรื่องวิทยาศาสตร์น่ารู้	ทองบรรณารักษ์	Available
15	สัมผัสญี่ปุ่น	ทองบรรณารักษ์	Available
16	reverse hybrid กลืนหัวใจคิด พิชิตดงลา Sirichaiwatt		Available

```
public ArrayList<Book> booklist(){
    ArrayList<Book> booksList = new ArrayList<Book>();
    try {
        Connection con = Connect.connectDB();
        String querybook = "SELECT * FROM bookinfo";
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(querybook);
        Book book;
        while (rs.next()){
            book = new Book(rs.getInt("callnumber"), rs.getString("bname"), rs.getString("bookauthor"), rs.getBoolean("status"));
            booksList.add(book);
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
    return booksList;
}
```

return ArrayList ที่ประกอบด้วยหนังสือทั้งหมด

```
public ArrayList<Book> booklist(int searchindex, String searchText){
    ArrayList<Book> booksList = new ArrayList<Book>();
    try {
        Connection con = Connect.connectDB();
        String searchby="";
        switch (searchindex){
            case 0: searchby = "callnumber"; break;
            case 1: searchby = "bname"; break;
            case 2: searchby = "bookauthor"; break;
        }
        String querybook;
        if (searchby.equals("callnumber"))
            querybook = "SELECT * FROM bookinfo WHERE " + searchby + " LIKE '" + searchText + "'";
        else
            querybook = "SELECT * FROM bookinfo WHERE " + searchby + " LIKE '%" + searchText + "%'";
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(querybook);
        Book book;
        while (rs.next()){
            book = new Book(rs.getInt("callnumber"), rs.getString("bname"), rs.getString("bookauthor"), rs.getBoolean("status"));
            booksList.add(book);
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
    return booksList;
}
```

return ArrayList ที่ประกอบด้วยหนังสือที่ตรงกับ คำค้นหา



# โค้ดและคำอธิบายโค้ด

## Class AdminSearch

ค้นหาจาก

ข้อมูลที่ต้องการค้นหา

เลขเรียกหนังสือ	ชื่อหนังสือ	ชื่อผู้แต่ง	สถานะ
1	เมย์ไหน โพลแรงเปรี้ยว	ณัฐพล อยู่วิทยา	Available
2	มหัศจรรย์พันธุ์ไม้	กองบรรณาธิการ	Unavailable
3	ฟรีแลนซ์ หันป่วย หันพัก หันรักหมอ	ณัฐพล อยู่วิทยา	Unavailable
4	หัวใจละลายรัก ภาคต่อว่าฉันลืมรัก	รัตนกัญญา	Unavailable
5	หลากหลายสรรพสัตว์	กองบรรณาธิการ	Unavailable
6	บทสวนนตรีก่อนนอน	ก. รังสิต	Unavailable
7	8 อ. ขบวนการสุขภาพดี	กองบรรณาธิการ	Available
8	เรียนรู้เศรษฐกิจพอเพียง	กองบรรณาธิการ	Unavailable
9	ป่าเลนเจ้าเฉย	กองบรรณาธิการ	Available
10	คืนพวกรัณษายาต เล่ม 1	ละปิงงา สิงขร	Available
11	ซาก 500 ซาก เล่ม 5	หัตถะธรา	Available
12	หลักกลยุทธ์	T.M.E.	Unavailable
13	อยู่กับทุกข์ให้เป็นสุข	โสภณ ปิณฑาแสง	Available
14	100 เรื่องวิทยาศาสตร์น่ารู้	กองบรรณาธิการ	Available
15	สัมผัสญี่ปุ่น	กองบรรณาธิการ	Available
16	reverse hybrid กลับหัวคิด พิชิตอนาคต Sirichaiwatt		Available

```
public void showBookTable(ArrayList<Book> list) {
    DefaultTableModel model = (DefaultTableModel) jTable_bookinfo.getModel();
    Object[] row = new Object[4];
    model.setRowCount(0);
    for (int i = 0; i < list.size(); i++) {
        row[0] = list.get(i).getCallnumber() + "";
        row[1] = list.get(i).getBname();
        row[2] = list.get(i).getBookauthor();
        if (list.get(i).getStatus() == true) {
            row[3] = "Available";
        } else {
            row[3] = "Unavailable";
        }
        model.addRow(row);
    }
}
```

นำข้อมูลใน ArrayList มาแสดงใน Table

```
private void btn_searchActionPerformed(java.awt.event.ActionEvent evt) {
    showBookTable(booklist(cbb_searchby.getSelectedIndex(), tf_searchdata.getText()));
}

private void btn_resetActionPerformed(java.awt.event.ActionEvent evt) {
    showBookTable(all_books_list);
}
```

- เมื่อกดปุ่ม Search จะแสดง Table จาก ArrayList ที่มีเฉพาะหนังสือที่ตรงกัน
- เมื่อกดปุ่ม Reset จะแสดง Table จาก ArrayList ที่มีหนังสือทั้งหมด



# โค้ดและคำอธิบายโค้ด

## Class AdminBorrowNreturn

Panel: borrow

ยืมหนังสือ

คืนหนังสือ

รหัสการยืม : 002

รหัสเรียกหนังสือ : 56

ชื่อหนังสือ : Effective Java

รหัสผู้ยืม : 1

ชื่อผู้ยืม : Kanasin Amornkittisarn

จำนวนวันที่ยืม : 7 วัน

วันที่ยืม : 08/12/2563

วันที่คืน : 15/12/2563

บันทึกการยืมหนังสือ



```
private void tf_calnumber1KeyReleased(java.awt.event.KeyEvent evt) {  
    if (!tf_calnumber1.getText().isEmpty()) {  
        try {  
            Connection con = Connect.connectDB();  
            String querybook = "SELECT * FROM bookinfo WHERE calnumber = " + tf_calnumber1.getText() + " AND status = 1";  
            Statement st = con.createStatement();  
            ResultSet rs = st.executeQuery(querybook);  
            if (rs.next()) {  
                tf_bookname1.setText(rs.getString("bname"));  
                byte[] img = rs.getBytes("imgbook");  
                ImageIcon image = new ImageIcon(img);  
                Image im = image.getImage();  
                Image myImg = im.getScaledInstance(lb_img_book1.getWidth(), lb_img_book1.getHeight(), Image.SCALE_SMOOTH);  
                ImageIcon newImage = new ImageIcon(myImg);  
                lb_img_book1.setIcon(newImage);  
            } else {  
                tf_bookname1.setText("ไม่พบ");  
                ImageIcon img_questionbook = new ImageIcon(new ImageIcon(getClass().getResource("../image/questionbook.jpg")).getImage().getScaledInstance(300, 400, Image.SCALE_SMOOTH));  
                lb_img_book1.setIcon(img_questionbook);  
            }  
        } catch (SQLException ex) {  
            JOptionPane.showMessageDialog(null, ex);  
        }  
    } else {  
        tf_bookname1.setText("ไม่พบ");  
        ImageIcon img_questionbook = new ImageIcon(new ImageIcon(getClass().getResource("../image/questionbook.jpg")).getImage().getScaledInstance(300, 400, Image.SCALE_SMOOTH));  
        lb_img_book1.setIcon(img_questionbook);  
    }  
}
```

ค้นหาหนังสือเรียกหนังสือใน Database แล้วนำมาแสดง ชื่อหนังสือและรูปหนังสือ

```
private void tf_userid1KeyReleased(java.awt.event.KeyEvent evt) {  
    if (!tf_userid1.getText().isEmpty()) {  
        try {  
            Connection con = Connect.connectDB();  
            String querybook = "SELECT * FROM userinfo WHERE userid = " + tf_userid1.getText();  
            Statement st = con.createStatement();  
            ResultSet rs = st.executeQuery(querybook);  
            if (rs.next()) {  
                tf_username1.setText(rs.getString("fname") + " " + rs.getString("sname"));  
            } else {  
                tf_username1.setText("ไม่พบ");  
            }  
        } catch (SQLException ex) {  
            JOptionPane.showMessageDialog(null, ex);  
        }  
    } else {  
        tf_username1.setText("ไม่พบ");  
    }  
}
```

ค้นหารหัสผู้ใช้ใน Database แล้วนำมาแสดง ชื่อผู้ยืม





# โค้ดและคำอธิบายโค้ด

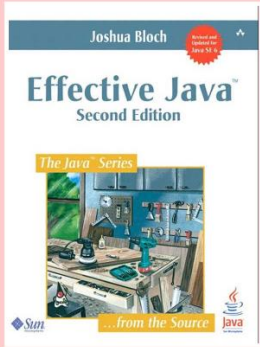
## Class AdminBorrowNreturn

Panel: borrow

ยืมหนังสือ

คืนหนังสือ

รหัสการยืม : 002  
รหัสเรียกหนังสือ : 56  
ชื่อหนังสือ : Effective Java  
รหัสผู้ยืม : 1  
ชื่อผู้ยืม : Kanasin Amornkittisarn  
จำนวนวันที่ยืม : 7 วัน  
วันที่ยืม : 08/12/2563  
วันที่คืน : 15/12/2563



บันทึกการยืมหนังสือ

```
private void btn_record_borrowingActionPerformed(java.awt.event.ActionEvent evt) {  
  
    if (!tf_callnumber1.getText().isEmpty() && !tf_userid1.getText().isEmpty() && cbb_borrowdays1.getSelectedIndex() != 0 &&  
        !tf_bookname1.getText().equals("ไม่พบ")){  
        try {  
            Connection con = Connect.connectDB();  
            String insert = "INSERT INTO borrowhistory(callnumber, userid, borrowdate, returndate) VALUES (?, ?, ?, ?)";  
            String update = "UPDATE `bookinfo` SET `status` = 0 WHERE callnumber=?";  
            PreparedStatement pst = con.prepareStatement(insert);  
            pst.setString(1, tf_callnumber1.getText());  
            pst.setString(2, tf_userid1.getText());  
            pst.setString(3, tf_borrowdate1.getText());  
            pst.setString(4, tf_returndate1.getText());  
            pst.executeUpdate();  
            pst = con.prepareStatement(update);  
            pst.setString(1, tf_callnumber1.getText());  
            pst.executeUpdate();  
  
            AdminGUI.updateSearchTable();  
            AdminGUI.updateBorrowhistoryTable();  
        } catch (SQLException ex) {  
            Logger.getLogger(AdminGUI.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    }  
}
```

เมื่อกรอกข้อมูลครบถ้วนและกดบันทึกการยืมหนังสือ จะเพิ่มข้อมูลการยืมหนังสือลงใน Database จากนั้นจะสั่งให้ AdminGUI ทำการ update ตารางในหน้า AdminSearch และ AdminHistory





# โค้ดและคำอธิบายโค้ด

## Class AdminBorrowNreturn

### Panel: Return

ยืมหนังสือ

คืนหนังสือ

รหัสการยืม : 17  
รหัสเรียกหนังสือ : 56  
ชื่อหนังสือ : Effective Java  
รหัสผู้ยืม : 1  
ชื่อผู้ยืม : Kanasin Amornkittisarn  
วันที่ยืม : 08/12/2563  
วันที่คืน : 22/12/2563

บันทึกการคืนหนังสือ

```
private void tf_borrowid2KeyReleased(java.awt.event.KeyEvent evt) {
    try {
        Connection con = Connect.connectDB();
        String query = "SELECT * FROM borrowhistory "
            + "INNER JOIN bookinfo ON borrowhistory.callnumber = bookinfo.callnumber "
            + "INNER JOIN userinfo ON borrowhistory.userid = userinfo.userid "
            + "WHERE borrowid=? AND returned=0";
        PreparedStatement pst = con.prepareStatement(query);
        pst.setString(1, tf_borrowid2.getText());
        ResultSet rs = pst.executeQuery();
    }
}
```

```
private void tf_callnumber2KeyReleased(java.awt.event.KeyEvent evt) {
    try {
        Connection con = Connect.connectDB();
        String query = "SELECT * FROM borrowhistory "
            + "INNER JOIN bookinfo ON borrowhistory.callnumber = bookinfo.callnumber "
            + "INNER JOIN userinfo ON borrowhistory.userid = userinfo.userid "
            + "WHERE borrowhistory.callnumber=? AND returned=0";
        PreparedStatement pst = con.prepareStatement(query);
        pst.setString(1, tf_callnumber2.getText());
        ResultSet rs = pst.executeQuery();
    }
}
```

เมื่อกรอก รหัสการยืม หรือ รหัสเรียกหนังสือ จะดึงข้อมูลใน Database ที่มีการ JOIN แต่ละ Table เข้าด้วยกัน เพื่อนำมาแสดงข้อมูลต่างๆ ที่มาจากแต่ละ Table

```
private void btn_record_returningActionPerformed(java.awt.event.ActionEvent evt) {
    if (!tf_callnumber2.getText().isEmpty() && !tf_borrowid2.getText().isEmpty()) {
        try {
            Connection con = Connect.connectDB();
            String updateborrowhistory = "UPDATE `borrowhistory` SET `returned` = 1 WHERE borrowid=?";
            String updatebookinfo = "UPDATE `bookinfo` SET `status` = 1 WHERE callnumber=?";
            PreparedStatement pst = con.prepareStatement(updateborrowhistory);
            pst.setString(1, tf_borrowid2.getText());
            pst.executeUpdate();
            pst = con.prepareStatement(updatebookinfo);
            pst.setString(1, tf_callnumber2.getText());
            pst.executeUpdate();

            AdminGUI.updateSearchTable();
            AdminGUI.updateBorrowhistoryTable();
        }
    }
}
```

เมื่อพบข้อมูลการยืมหนังสือ และ กดบันทึกการคืนหนังสือ  
จะอัปเดต สถานะการยืม และ สถานะของหนังสือ ใน Database  
จากนั้นจะสั่ง AdminGUI ให้ Update JTable



# โค้ดและคำอธิบายโค้ด

## Class AdminBorrowhistory

### ประวัติการยืม-คืนหนังสือ

คืนแล้ว  ยังไม่คืน

รายการยืม	รายการหนังสือ	ชื่อหนังสือ	ชื่อผู้ยืม	วันที่ยืม	วันที่คืน	สถานะการยืม
1	1	เมย์ทูน ไฟแรงพรี	Kanasin Amornkittisarn	06/12/2563	16/12/2563	คืนแล้ว
17	56	Effective Java	Kanasin Amornkittisarn	08/12/2563	22/12/2563	ยังไม่คืน
18	80	The Hardware Hacker	Kanasin Amornkittisarn	08/12/2563	15/12/2563	ยังไม่คืน

```
public void setborrowhistoryTable(){
    try {
        Connection con = Connect.connectDB();
        String queryhistory
            = "SELECT * FROM borrowhistory "
            + "INNER JOIN bookinfo ON borrowhistory.callnumber = bookinfo.callnumber "
            + "INNER JOIN userinfo ON borrowhistory.userid = userinfo.userid";
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(queryhistory);
        DefaultTableModel model = (DefaultTableModel) jTable_borrowhistory.getModel();
        Object[] row = new Object[7];
        model.setRowCount(0);
        while (rs.next()){
            row[0] = rs.getString("borrowid");
            row[1] = rs.getString("callnumber");
            row[2] = rs.getString("bname");
            row[3] = rs.getString("fname") + " " + rs.getString("sname");
            row[4] = rs.getString("borrowdate");
            row[5] = rs.getString("returndate");
            if (rs.getBoolean("returned") == false)
                row[6] = "ยังไม่คืน";
            else
                row[6] = "คืนแล้ว";
            model.addRow(row);
        }
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, ex);
    }
}
```

แสดง Table โดยดึงข้อมูลจากใน Database

ตั้งค่าสีพื้นหลังของตารางในแต่ละแถว โดยสร้าง Class ที่กำหนดการ render ของ cell ในแต่ละแถวขึ้นมาใหม่ โดยสีพื้นหลังที่แสดง จะพิจารณาจากค่าในคอลัมน์

สถานะการยืม

```
public void setTableColor() {
    jTable_borrowhistory.setDefaultRenderer(Object.class, new DefaultTableCellRenderer() {
        public Component getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int col) {
            super.getTableCellRendererComponent(table, value, isSelected, hasFocus, row, col);
            String status = (String) table.getModel().getValueAt(row, 6);
            if (status.equals("ยังไม่คืน")) {
                setBackground(new Color(255, 153, 153));
                setForeground(table.getForeground());
            } else {
                setBackground(new Color(153, 255, 153));
                setForeground(table.getForeground());
            }
            return this;
        }
    });
}
```



# โค้ดและคำอธิบายโค้ด

## Class AdminBook

Panel: Addbook

เพิ่มข้อมูลหนังสือ

ลบข้อมูลหนังสือ

รหัสเรียกหนังสือ : 100

ชื่อหนังสือ : godzilla awakening

ชื่อผู้แต่ง : เกิร์ก ธิมเรชชาติเน้

รูปปกหนังสือ :

bookinfo-imgbook.bin



```
private void btn_addbookActionPerformed(java.awt.event.ActionEvent evt) {  
    try{  
        Connection con = Connect.connectDB();  
        String query = "INSERT INTO bookinfo(callnumber, bname, bookauthor, imgbook, status) VALUES (?, ?, ?, ?, ?)";  
        PreparedStatement pst = con.prepareStatement(query);  
        InputStream is = new FileInputStream(new File(bookimgpath));  
        pst.setString(1, tf_callnumber3.getText());  
        pst.setString(2, tf_bookname3.getText());  
        pst.setString(3, tf_bookauthor3.getText());  
        pst.setBlob(4, is);  
        pst.setBoolean(5, true);  
        pst.executeUpdate();  
  
        AdminGUI.updateSearchTable();  
        JOptionPane.showMessageDialog(null, "เพิ่มข้อมูลหนังสือสำเร็จ");  
    }catch(Exception ex){  
        JOptionPane.showMessageDialog(null, ex);  
    }  
}
```

เมื่อกรอกข้อมูลหนังสือพร้อมแนบไฟล์รูปภาพ และ กดปุ่มเพิ่มข้อมูลหนังสือ  
จะ Insert ข้อมูลหนังสือลงใน Database  
และสั่งให้ AdminGUI อัปเดต jTable ใน หน้าค้นหาหนังสือ



# โค้ดและคำอธิบายโค้ด

## Class AdminBook

### Panel: Removebook

เพิ่มข้อมูลหนังสือ

ลบข้อมูลหนังสือ

รหัสเรียกหนังสือ : 23

ชื่อหนังสือ : 5นธิ The Da Vinci Code

ชื่อผู้แต่ง : แดน บราวน์



ลบข้อมูลหนังสือ

```
private void tf_callnumber4KeyReleased(java.awt.event.KeyEvent evt) {  
    try{  
        Connection con = Connect.connectDB();  
        String query = "SELECT * FROM `bookinfo` WHERE callnumber=?";  
        PreparedStatement pst = con.prepareStatement(query);  
        pst.setString(1, tf_callnumber4.getText());  
        ResultSet rs = pst.executeQuery();  
    }  
}
```

เมื่อกรอกรหัสเรียกหนังสือ จะดึงข้อมูลและ  
รูปภาพหนังสือ ใน Database มาแสดง

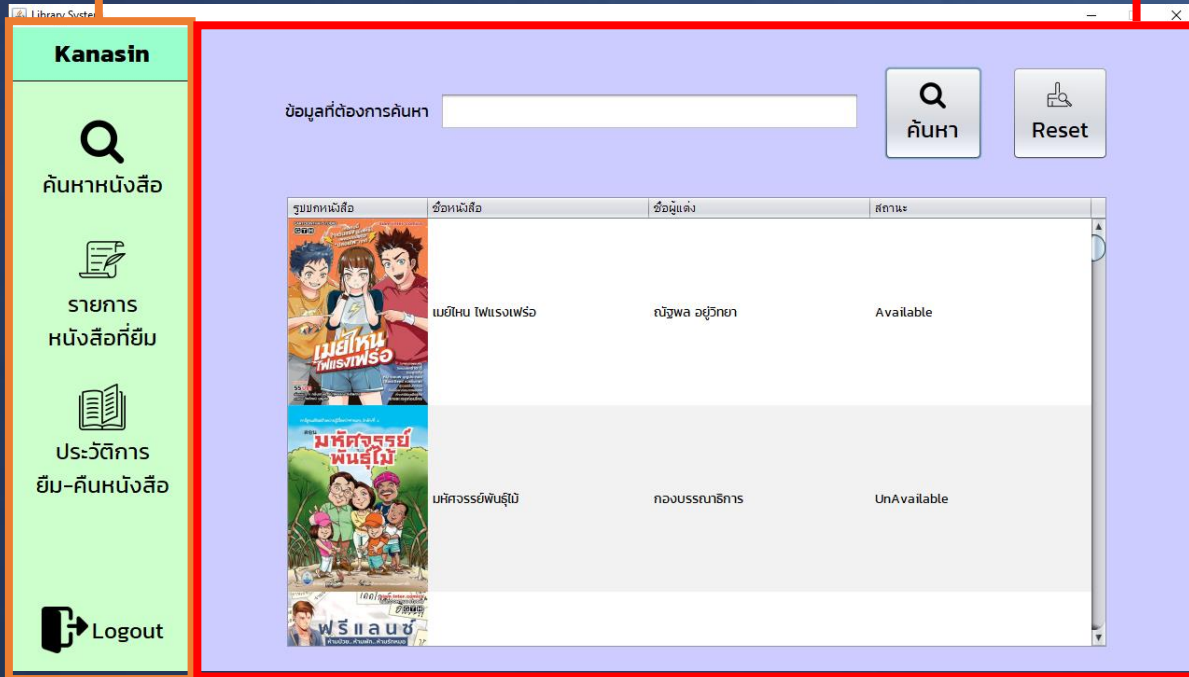
```
private void btn_removebookActionPerformed(java.awt.event.ActionEvent evt) {  
    try{  
        Connection con = Connect.connectDB();  
        String query = "DELETE FROM `bookinfo` WHERE callnumber=?";  
        PreparedStatement pst = con.prepareStatement(query);  
        pst.setString(1, tf_callnumber4.getText());  
        pst.executeUpdate();  
        AdminGUI.updateSearchTable();  
        JOptionPane.showMessageDialog(null, "ลบข้อมูลหนังสือสำเร็จ");  
    }catch(Exception ex){  
        JOptionPane.showMessageDialog(null, ex);  
    }  
}
```

เมื่อกดปุ่มลบข้อมูลหนังสือ จะลบข้อมูลหนังสือนั้นๆ ใน Database  
และสั่งให้ AdminGUI อัปเดต JTable ใน หน้าค้นหาหนังสือ



โค้ดและคำอธิบายโค้ด

## Class UserGUI



```
public static void logout() {
    frame.dispose();
    new LoginForm();
}
```

```
public class UserGUI {
    private UserMenu menu;
    private static JFrame frame;
    private static JPanel pa_center;
    private UserSearch pa_search;
    private UserBorrow pa_borrow;
    private UserHistory pa_history;

    public UserGUI(int userid) {
        frame = new JFrame("Library System");
        menu = new UserMenu(userid);
        menu.setBorder(BorderFactory.createMatteBorder(0, 0, 0, 1, Color.black));

        pa_center = new JPanel();
        pa_search = new UserSearch();
        pa_borrow = new UserBorrow(userid);
        pa_history = new UserHistory(userid);

        pa_center.setLayout(new CardLayout());
        pa_center.add(pa_search, "panel_search");
        pa_center.add(pa_borrow, "panel_borrow");
        pa_center.add(pa_history, "panel_history");

        frame.add(menu, BorderLayout.WEST);
        frame.add(pa_center, BorderLayout.CENTER);

        frame.setSize(1280, 720);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

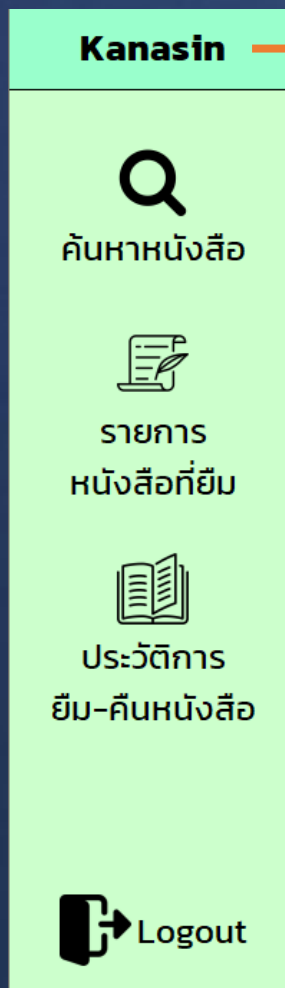
```
public static void getCenterPanel(String panel_name) {
    CardLayout card = (CardLayout) (pa_center.getLayout());
    card.show(pa_center, panel_name);
    System.out.println("Change to " + panel_name);
}
```





โค้ดและคำอธิบายโค้ด

## Class UserMenu



```
public void setUserFirstname(int userid){  
    try {  
        Connection con = Connect.connectDB();  
        String queryname = "SELECT fname FROM userinfo WHERE userid = " + userid;  
        Statement st = con.createStatement();  
        ResultSet rs = st.executeQuery(queryname);  
        if (rs.next())  
            lb_userFirstname.setText(rs.getString("fname"));  
    } catch (SQLException ex) {  
        JOptionPane.showMessageDialog(null, ex);  
    }  
}
```

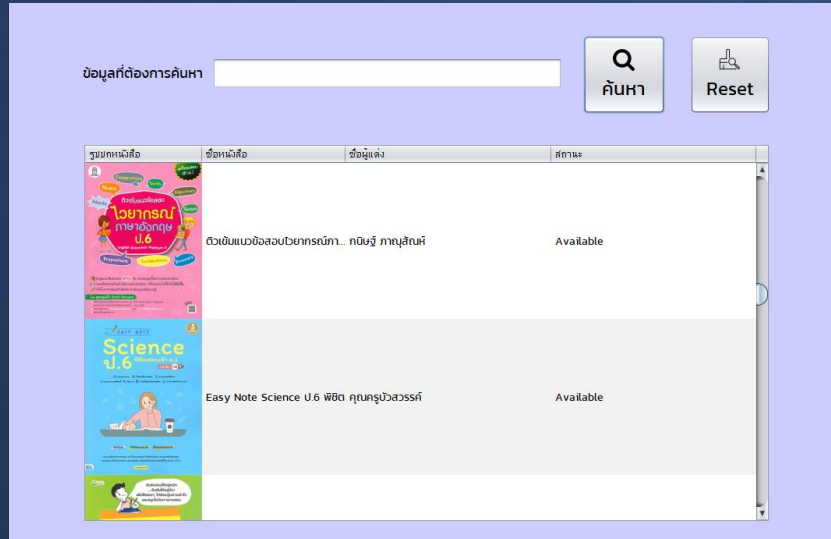
นำ parameter userid ที่ส่งมาจาก UserGUI ไปค้นหา ชื่อผู้ใช้ ใน Database  
เพื่อนำมาแสดงในแถบเมนู





# โค้ดและคำอธิบายโค้ด

## Class UserSearch



ให้ Column รูปปกหนังสือมีการแสดงผล Cell นั้น  
เป็นค่าของ JLabel (label ที่เป็นรูปภาพ)

```
String querybook = "SELECT * FROM bookinfo WHERE CONCAT(bname, bookauthor) LIKE '%" + searchText + "%'";
```

ข้อมูลในตารางจะใช้วิธีสร้าง ArrayList ที่คล้ายกันกับ Class AdminSearch  
และเมื่อกดค้นหาจะพิจารณา ชื่อหนังสือและชื่อผู้แต่ง พร้อมกัน

```
public void showBookTable(ArrayList<Book> list) {
    DefaultTableModel model = (DefaultTableModel) JTable_bookinfo.getModel();
    JTable_bookinfo.getColumnModel().setCellRenderer(new MyTableCellRenderer());
    Object[] row = new Object[4];
    model.setRowCount(0);
    for (int i = 0; i < list.size(); i++) {
        ImageIcon image = new ImageIcon(new ImageIcon(list.get(i).getImgbook()).getImage().getScaledInstance(150, 200, Image.SCALE_SMOOTH));
        JLabel imagelb = new JLabel();
        imagelb.setIcon(image);
        row[0] = imagelb;
        row[1] = list.get(i).getBname();
        row[2] = list.get(i).getBookauthor();
        if (list.get(i).getStatus() == true) {
            row[3] = "Available";
        } else {
            row[3] = "UnAvailable";
        }
        model.addRow(row);
    }
}
```

```
class MyTableCellRenderer implements TableCellRenderer {
    public JLabel getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column){
        return (JLabel) value;
    }
}
```



# โค้ดและคำอธิบายโค้ด

## Class UserBorrow

รายการหนังสือที่ยืมอยู่ปัจจุบัน

ชื่อหนังสือ	วันที่ยืม	วันที่คืน
Effective Java	08/12/2563	22/12/2563
The Hardware Hacker	08/12/2563	15/12/2563

```
public void setBorrowTable(int userid){
    try {
        Connection con = Connect.connectDB();
        String queryhistory = "SELECT bname, borrowdate, returndate FROM borrowhistory "
            + "INNER JOIN bookinfo ON borrowhistory.callnumber = bookinfo.callnumber "
            + "WHERE userid = " + userid + " AND returned = 0";
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(queryhistory);
        DefaultTableModel model = (DefaultTableModel) JTable_presentborrow.getModel();
        Object[] row = new Object[3];
        while (rs.next()){
            row[0] = rs.getString("bname");
            row[1] = rs.getString("borrowdate");
            row[2] = rs.getString("returndate");
            model.addRow(row);
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

แสดงข้อมูล หนังสือที่ยืมอยู่ปัจจุบัน โดยอาศัยค่า userid ในการค้นหารายการยืมหนังสือของ ผู้ใช้ที่ login อยู่ และเลือกเฉพาะรายการยืมหนังสือที่ยังไม่คืน



# โค้ดและคำอธิบายโค้ด

## Class UserHistory

### ประวัติการยืมหนังสือ

ชื่อหนังสือ	วันที่ยืม	วันที่คืน	สถานะ
เบย์ihu วิทยาศาสตร์	06/12/2563	16/12/2563	Available
Effective Java	08/12/2563	22/12/2563	Unavailable
The Hardware Hacker	08/12/2563	15/12/2563	Unavailable

```
public void setHistoryTable(int userid){
    try {
        Connection con = Connect.connectDB();
        String queryhistory = "SELECT bname, borrowdate, returndate,status FROM borrowhistory "
            + "INNER JOIN bookinfo ON borrowhistory.callnumber = bookinfo.callnumber "
            + "WHERE userid = " + userid;
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(queryhistory);
        DefaultTableModel model = (DefaultTableModel) JTable_borrowhistory.getModel();
        Object[] row = new Object[4];
        while (rs.next()){
            row[0] = rs.getString("bname");
            row[1] = rs.getString("borrowdate");
            row[2] = rs.getString("returndate");
            if (rs.getBoolean("status") == false){
                row[3] = "Unavailable";
            }
            else {
                row[3] = "Available";
            }

            model.addRow(row);
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

แสดงข้อมูล หนังสือที่เคยยืมโดยอาศัยค่า userid ในการค้นหารายการยืม  
หนังสือของ ผู้ใช้ที่ login อยู่ พร้อมกับแสดงสถานะของหนังสือนั้นๆ

# หน้าที่การทำงานภายในกลุ่ม

## 1. นายคณาสิน อมรภิตติสาร

- ศึกษาข้อมูลการทำฐานข้อมูล
- ทำการทำงานของ GUI ในฝั่ง Admin panel เพิ่ม-ลบหนังสือ
- ทำการทำงานของ GUI ในฝั่ง Admin panel คืบหนังสือ



## 2. นายรัชพนธ์ ศรีอากาศไกรแสง

- หาหนังสือและเพิ่มหนังสือลงฐานข้อมูล
- ทำการทำงานของ GUI ในฝั่ง Admin panel เพิ่ม-คืบหนังสือ



## 3. นายประธาน นาเวียง

- หาหนังสือและเพิ่มหนังสือลงฐานข้อมูล





# หน้าที่การทำงานภายในกลุ่ม

## 4. นายภัทรพล เกรรัตน์พันธิกุล

- ทำ Class Diagram
- ทำ User GUI



## 5. นายภูวนัตถ์ จันทรมี

- ศึกษาข้อมูลการทำฐานข้อมูล
- ทำ GUI ทุกส่วน
- ทำการทำงานของ GUI ทุกส่วนที่เป็น Jtable

