



# Chue Nang Sao

โปรแกรมจีօນางເສົາ ສໍາຮັບຮ້ານເຫຼັກສູ່ອ



OOP Final Project

# บทคัดย่อ



โปรเจคนี้มีจุดประสงค์เพื่อพัฒนาระบบการเช่า-คืนหนังสือของร้านเช่าหนังสือ  
เนื่องจากผู้พัฒนาได้เคยมีประสบการณ์การใช้บริการจากร้านเช่าหนังสือ แล้วพบว่าระบบ  
ของการเก็บข้อมูลต่างๆของหนังสือ หรือข้อมูลสมาชิกของร้านไม่สามารถตอบความ  
สะดวกให้แก่พนักงานที่ค่อยดูแลและลูกค้าได้ อีกทั้งยังเสี่ยงต่อการสูญหายของข้อมูลดัง<sup>1</sup>  
กล่าว เช่นการเก็บข้อมูลของหนังสือโดยการจดบันทึก หรือการระบุสมาชิกโดยการใช้บัตร  
ผู้พัฒนาจึงได้คิดถึงโปรแกรมสำหรับการเก็บข้อมูลหนังสือที่มีอยู่ในร้าน และข้อมูลของ  
ลูกค้าไว้บนฐานข้อมูล เพื่อให้ง่ายต่อการตอบคำถามลูกค้าเมื่อลูกค้าต้องการเช่า หรือ  
สอบถามเกี่ยวกับหนังสือ และรวมถึงการคำนวณจำนวนเงินทั้งหมดที่ลูกค้าต้องจ่ายในการเช่า  
ครั้งนั้นๆ เพื่อมอบความสะดวกแก่ทั้งพนักงานและตัวลูกค้าเอง





# วัตถุประสงค์



1. เพื่อศึกษาการเชื่อมต่อฐานข้อมูลกับ  
MongoDB และภาษา Java

2. เพื่อพัฒนาโปรแกรมสำหรับร้านเช่าหนังสือ

3. เพื่อนำความรู้ที่ได้เรียนมาจากการ  
ใช้ให้เกิดประโยชน์

1. ได้รู้ถึงวิธีการเชื่อมต่อฐานข้อมูลกับ  
MongoDB และภาษา Java

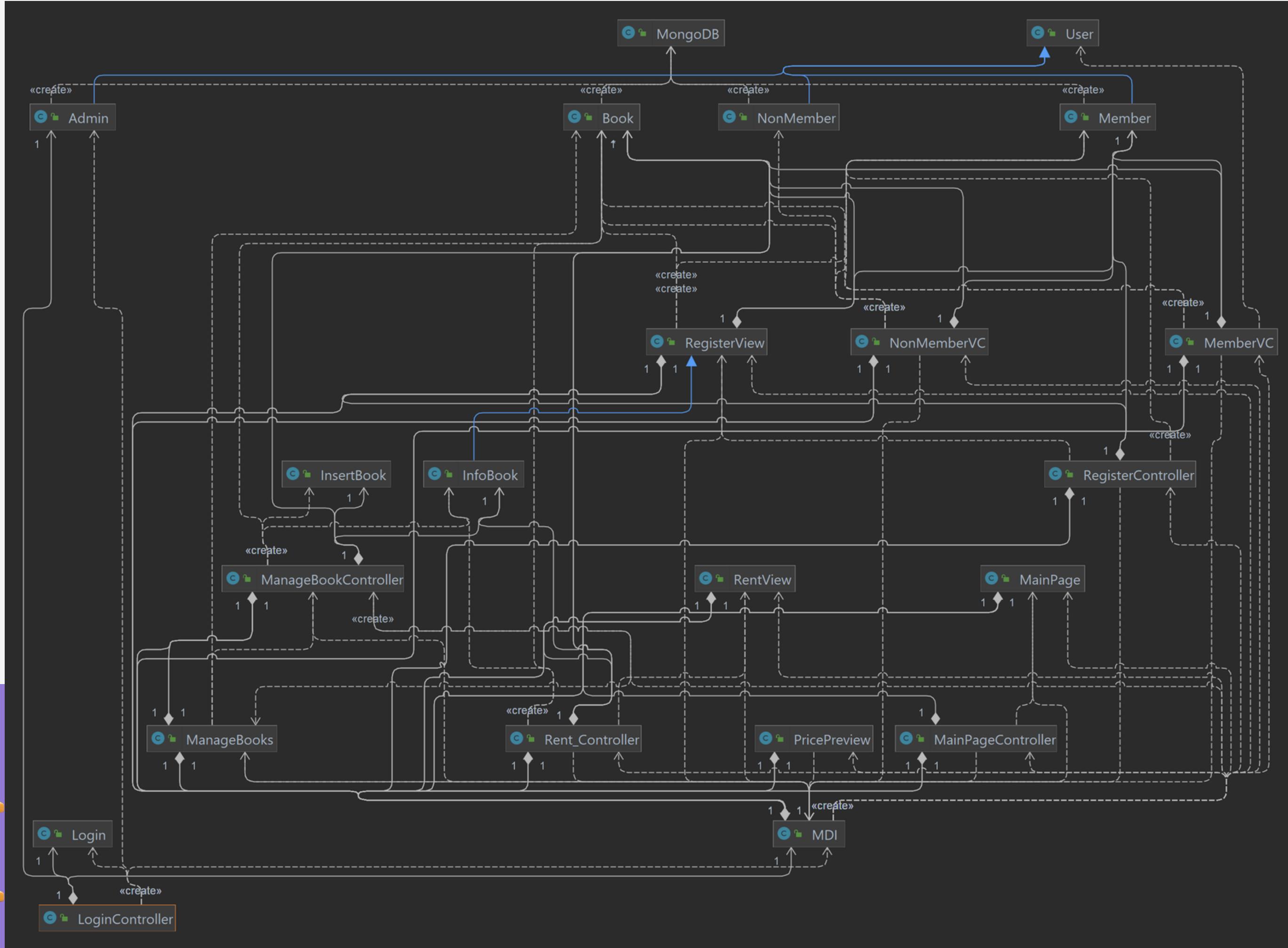
2. การเก็บข้อมูลหนังสือของร้านนั้น  
สูญหายยากขึ้น

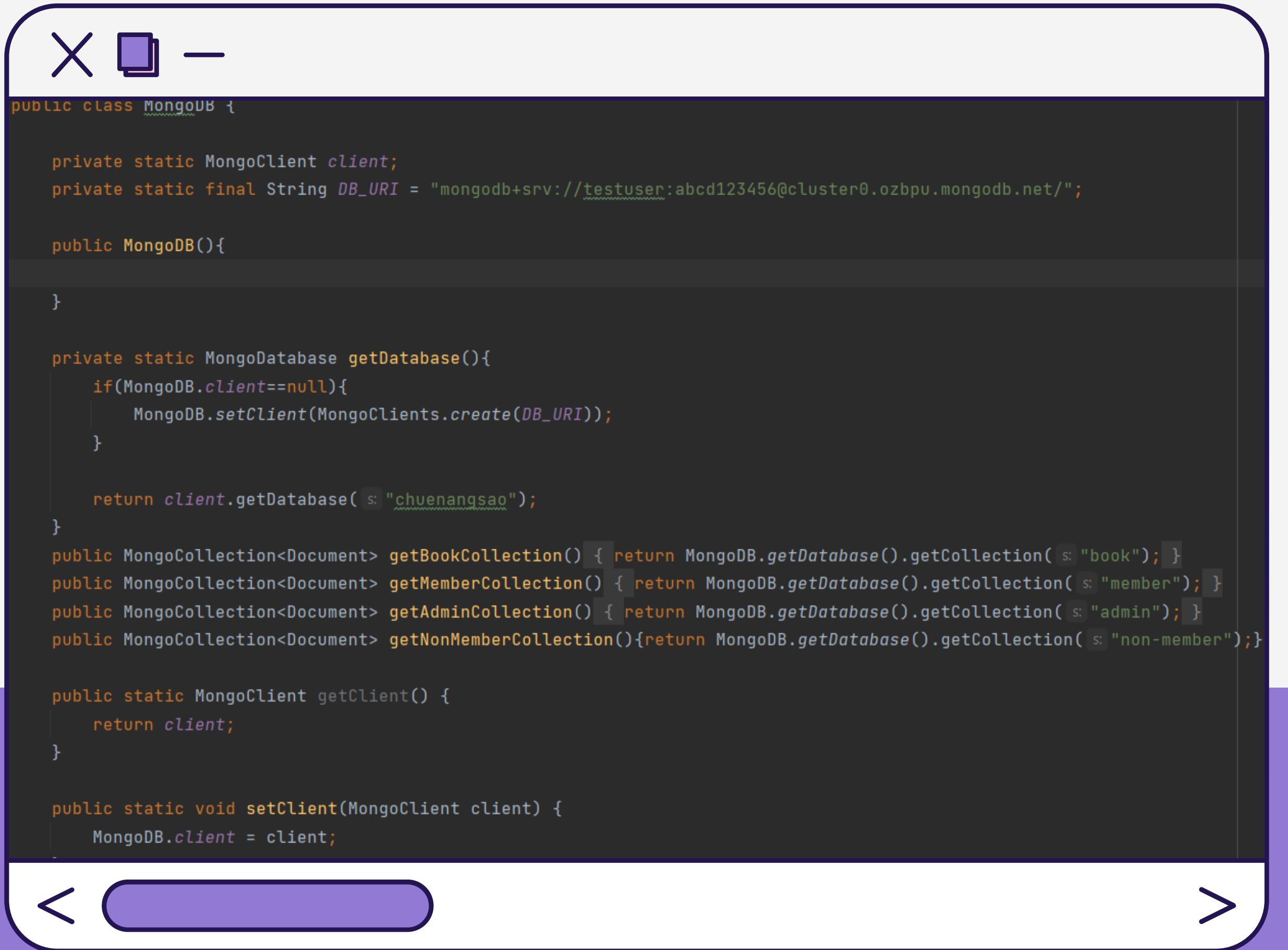
3. พนักงานและลูกค้าได้รับความสะดวก  
สบายมากขึ้นจากการใช้โปรแกรม

ประโยชน์ที่ได้รับ



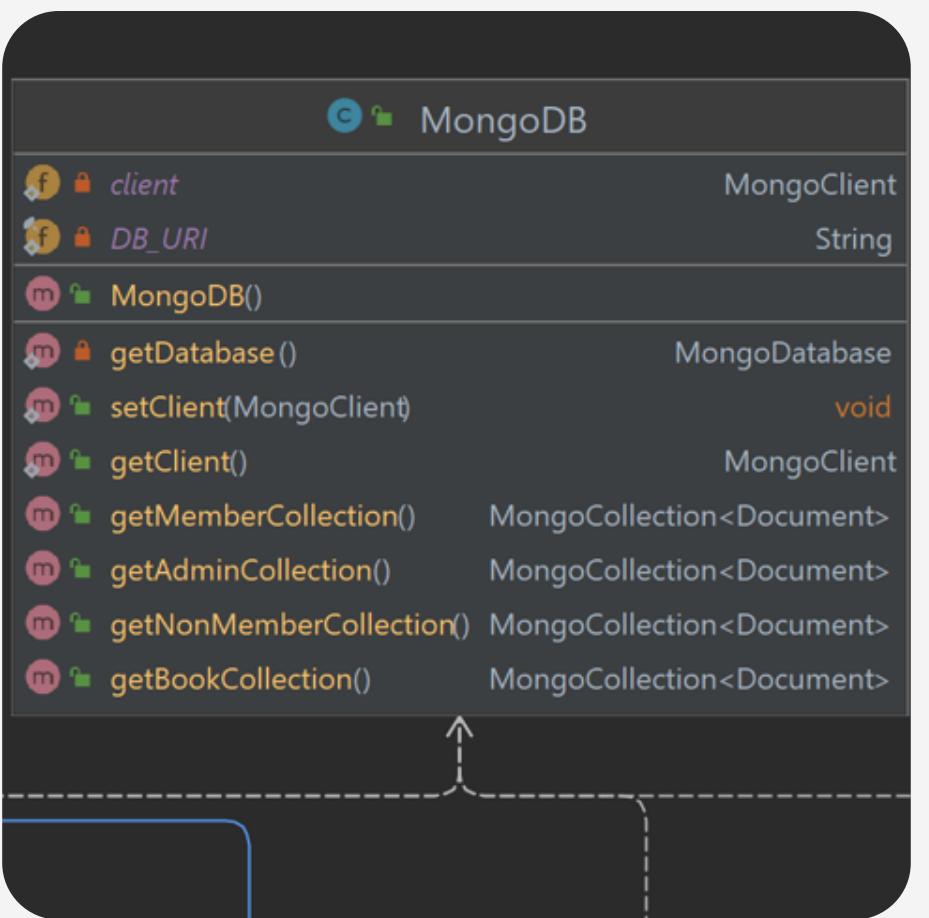
# Class Diagram





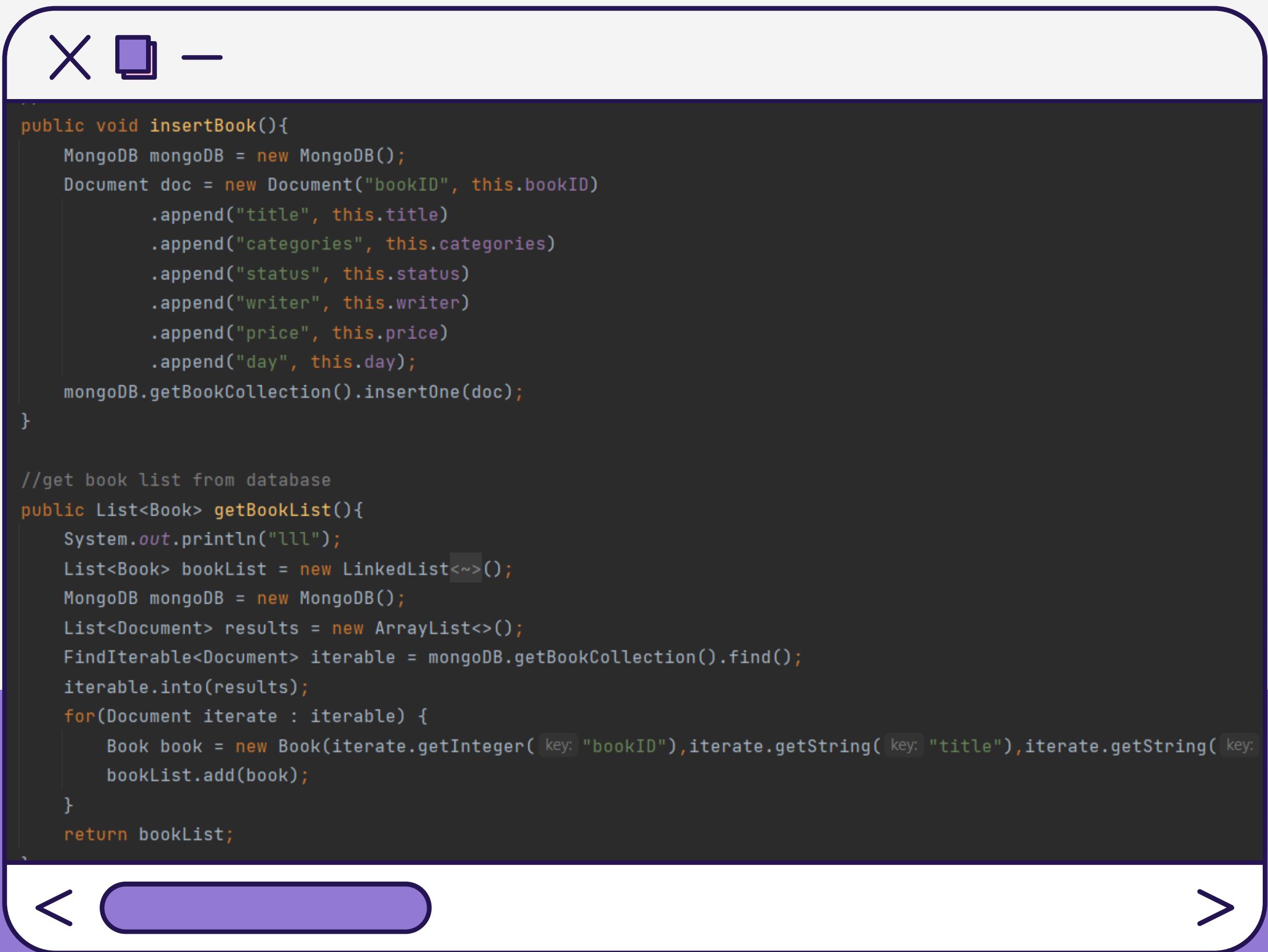
A screenshot of a Java code editor showing a class named `MongoDB`. The code initializes a static `MongoClient` and a static `String` `DB_URI` with a MongoDB connection string. It contains methods to get databases and collections, and static methods to get the client and set a new client. The code is color-coded with syntax highlighting.

```
public class MongoDB {  
  
    private static MongoClient client;  
    private static final String DB_URI = "mongodb+srv://testuser:abcd123456@cluster0.ozbpu.mongodb.net/";  
  
    public MongoDB(){  
    }  
  
    private static MongoDB getDatabase(){  
        if(MongoDB.client==null){  
            MongoDB.setClient(MongoClients.create(DB_URI));  
        }  
  
        return client.getDatabase("chuenangsao");  
    }  
  
    public MongoCollection<Document> getBookCollection() { return MongoDB.getDatabase().getCollection("book"); }  
    public MongoCollection<Document> getMemberCollection() { return MongoDB.getDatabase().getCollection("member"); }  
    public MongoCollection<Document> getAdminCollection() { return MongoDB.getDatabase().getCollection("admin"); }  
    public MongoCollection<Document> getNonMemberCollection(){return MongoDB.getDatabase().getCollection("non-member");}  
  
    public static MongoClient getClient() {  
        return client;  
    }  
  
    public static void setClient(MongoClient client) {  
        MongoDB.client = client;  
    }  
}
```



ส่วนที่เชื่อมต่อ กับ MongoDB  
และดึง Collection





```
public void insertBook(){
    MongoDB mongoDB = new MongoDB();
    Document doc = new Document("bookID", this.bookID)
        .append("title", this.title)
        .append("categories", this.categories)
        .append("status", this.status)
        .append("writer", this.writer)
        .append("price", this.price)
        .append("day", this.day);
    mongoDB.getBookCollection().insertOne(doc);
}

//get book list from database
public List<Book> getBookList(){
    System.out.println("lll");
    List<Book> bookList = new LinkedList<~>();
    MongoDB mongoDB = new MongoDB();
    List<Document> results = new ArrayList<~>();
    FindIterable<Document> iterable = mongoDB.getBookCollection().find();
    iterable.into(results);
    for(Document iterate : iterable) {
        Book book = new Book(iterate.getInteger(key: "bookID"), iterate.getString(key: "title"), iterate.getString(key: "categories"), iterate.getBoolean(key: "status"), iterate.getDouble(key: "price"), iterate.getInteger(key: "day"));
        bookList.add(book);
    }
    return bookList;
}
```

Book	
f	bookID
f	writer
f	status
f	categories
f	price
f	day
f	title
m	Book()
m	Book(int, String, String, boolean, String, double, int)
m	getBookList()
m	editBookDetail()
m	setDay(int)
m	isStatus()
m	setWriter(String)
m	getPrice()
m	setTitle(String)
m	rentBooks(List<Book>)
m	getWriter()
m	getBookID()
m	getDay()
m	getTitle()
m	setBookID(int)
m	setStatus(boolean)
m	returnBook(int)
m	getCategories()
m	setCategories(String)
m	setPrice(double)
m	insertBook()
m	deleteBook(int)

ស៊ុបទីបានទីកន្លែង  
ហើងសើវាក្សានប៉ូមូល



```
X ॥ -
```

```
public boolean isLogin(String id, String password){  
    List<Admin> adminList = new LinkedList<Admin>();  
    MongoDB mongoDB = new MongoDB();  
    Document doc = new Document("id", id);  
    FindIterable<Document> iterable = mongoDB.getAdminCollection().find(doc);  
    for(Document iterate : iterable) {  
        if (iterate.getString("password").equals(password)){  
            return true;  
        }  
    }  
    return false;  
}  
  
public String getPassword() {  
    return password;  
}  
  
public void setPassword(String password) {  
    this.password = password;  
}
```

< >

Admin		
f	password	String
m	Admin()	
m	Admin(String, String, String, String)	
m	setPassword(String)	void
m	getPassword()	String
m	getAdminList()	List<Admin>
m	registerAdmin()	void
m	isLogin(String, String)	boolean

ส่วนที่เอาไว้ตรวจสอบความ  
ถูกต้องในการ Login



```
int index = 0;
double totalprice = 0.00; //ราค่าเช่า (Day x Price)
//for(int i=0;i<RentController.bookList.size();i++){
//totalprice += RentController.bookList.get(i).getPrice() * RentController.bookList.get(i).getDay();
//}

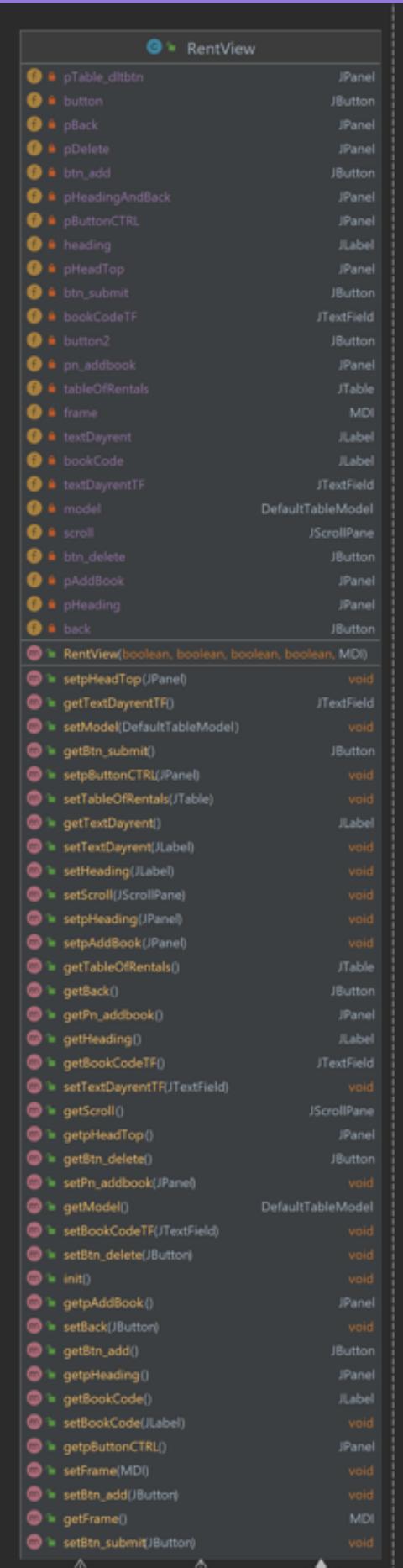
if(ae.getSource().equals(view.getSubmit())){
    //เช็ค ข้อมูล ถ้าถูกต้อง Alert ขึ้น และกลับหน้าเช่าหนังสือ
    if(view.getFname().getText().equals("") || view.getLname().getText().equals("") || view.getAddress().getText().equals("") || view.getCardid().getText().equals("") || view.getContract().getText().equals("")){
        JOptionPane.showMessageDialog(view.getFrame(), message: "Please Insert Name or Address or Contract.");
    }
}

else{
    Member member = new Member(view.getId().getText(),view.getFname().getText(),view.getLname().getText(),view.getAddress().getText(),view.getCardid().getText(),view.getContract().getText());
    member.registerMember();
    JOptionPane.showMessageDialog(view.getFrame(), message: "Register Success.");
    view.getId().setText("");
    view.getFname().setText("");
    view.getLname().setText("");
    view.getAddress().setText("");
    view.getCardid().setText("");
    view.getContract().setText("");
}
}
```

  RegisterController
  mdi
  model
  view
  RegisterController(RegisterView, MDI)
  actionPerformed(ActionEvent) void

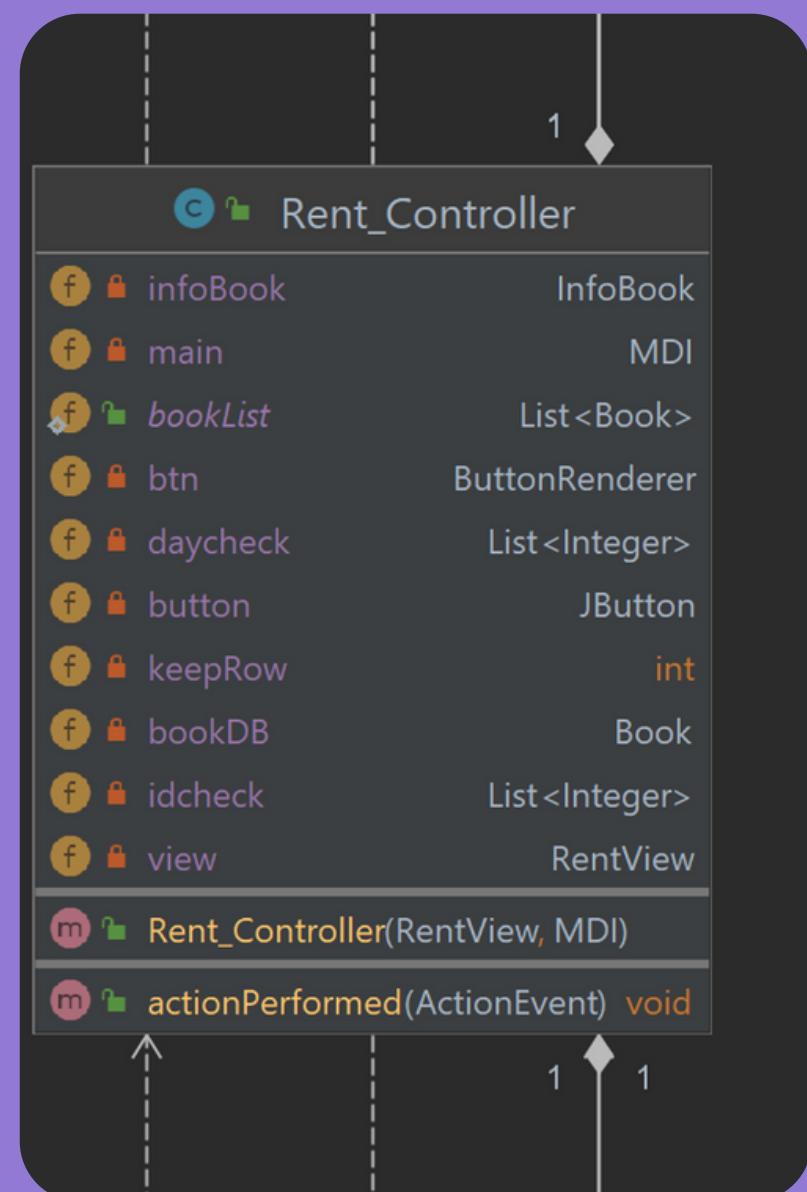
ส่วนของ Register





view

# គ្មារមេខ្មោះ



controller



```
override
public void actionPerformed(ActionEvent ae) {
    //ปุ่ม 1
    if (ae.getSource().equals(main.getRentView().getBtn_add())) { //ถ้ากดปุ่ม Add เพิ่มหนังสือที่จะเช่า
        int search = Integer.parseInt(main.getRentView().getBookCodeTF().getText()); //รับ BookCode
        int days = Integer.parseInt(main.getRentView().getTextDayrentTF().getText()); //รับจำนวนวัน
        if (idcheck.indexOf(search) == -1) { //เช็คว่าที่จะกรอกเข้ากับบันทึกของหนังสือใน DB หรือยัง อันนี้กรณีไม่เข้า
            for (int i = 0; i < booklist.getBookList().size(); i++) { //ลูปตามจำนวนของหนังสือที่มี
                int id = booklist.getBookList().get(i).getBookID(); // ให้รับ ID ของหนังสือจาก DB
                if (id == search) { //เช็คว่ามีหนังสือที่จะยืมไหม (ทำงาน ถ้า ID จาก DB และ search ตรงกัน)
                    book = booklist.getBookList().get(i); //รับหนังสือที่ยืมเข้า
                    break;
                }
            }
        } else { //ถ้าเข้าก็จะกรอกแล้วว่าง TF กรอกแล้วว่าง
            main.getRentView().getBookCodeTF().setText("");
            main.getRentView().getTextDayrentTF().setText("");
        }
    }
}
```

ส่วนของการเพิ่มหนังสือที่  
ต้องการจะเช่าลงในตาราง



X  - 

```
//ปุ่ม 2
else if (ae.getSource().equals(main.getRentView().getBtn_submit())) { //ถ้ากดปุ่ม Submit ส่งไป DB

    //ส่วนของการเชื่อมต่อไปหน้าคำนวนเงิน
    this.main.getRentView().setVisible(false);
    this.main.getRentView().hide();
    this.main.getDesktopPane().remove(this.main.getRentView());
    this.main.getDesktopPane().add(this.main.getPricePreview());
    this.main.getPricePreview().setVisible(true);
    this.main.getPricePreview().show();

    for (int j = 0; j < idcheck.size(); j++) {
        for (int i = 0; i < booklist.getBookList().size(); i++) {
            int id = booklist.getBookList().get(i).getBookID(); //ให้รับ ID ของหนังสือจาก DB ทุกตัวมาเช็ค
            if (idcheck.get(j) == id) { //ໄລเซ็คເອົາໄອດີເຮັມາເທິບກັບหนังສือໃນຮ້ານຕັ້ງແຕ່ເລີ່ມແຮກໄປເລີຍ ຄ້າຕຽນກັນທ່າງນັ້ນ
                book = booklist.getBookList().get(i); //หนังສือທ່ານັ້ນ ບໍ່ມີໃຫຍ່
                book.setDay(daycheck.get(j)); //ຮັບວັນຈາກ list ໄປ set ວ່າຈະຍືນກີ່ວັນ
                book.setStatus(false); //ໃຫ້ set ສະເຕັບສ່ວາລະຍືນກີ່ວັນ
                bookList.add(book); //ເພີ່ມหนังສือຈະຍືນເຂົ້າໄປໃນລົດ
                break;
        }
    }
}
```

<  >

ສ່ວນຂອງການກົດສົ່ງຂໍ້ອມູລັບຫັ້ງສັ່ອ  
ກັ້ງມຸດທີ່ຕ້ອງການເຫຼົາໄປ  
update ບົນຈານຂໍ້ອມູລ





```
//ปุ่ม 4
else if (ae.getSource().equals(main.getRentView().getBtn_delete())) { //ปุ่มการลบข้อมูล
    int index = (main.getRentView().getTableOfRentals().getSelectedRow());
    idcheck.remove(index);
    daycheck.remove(index);
    main.getRentView().getModel().setRowCount(0);
    for (int i = 0; i < idcheck.size(); i++) {
        for (Book checkbook : bookDB.getBookList()) {
            if (idcheck.get(i) == checkbook.getBookID()) {
                Object[] row = {idcheck.get(i), checkbook.getTitle(), daycheck.get(i)};
                main.getRentView().getModel().addRow(row); //นำ row ที่รับข้อมูลต่างๆไปใส่ใน
                main.getRentView().getTableOfRentals().getColumn( identifier: "") .setCellEditor(new DefaultCellEditor(""));
                main.getRentView().getTableOfRentals().getColumn( identifier: "") .setCellEditor(new DefaultCellEditor(""));
            }
        }
    }
}
```

< >

ส่วนของการลบและกดดู  
รายละเอียดของหนังสือ



# គ្រប់ការចែកការងារ

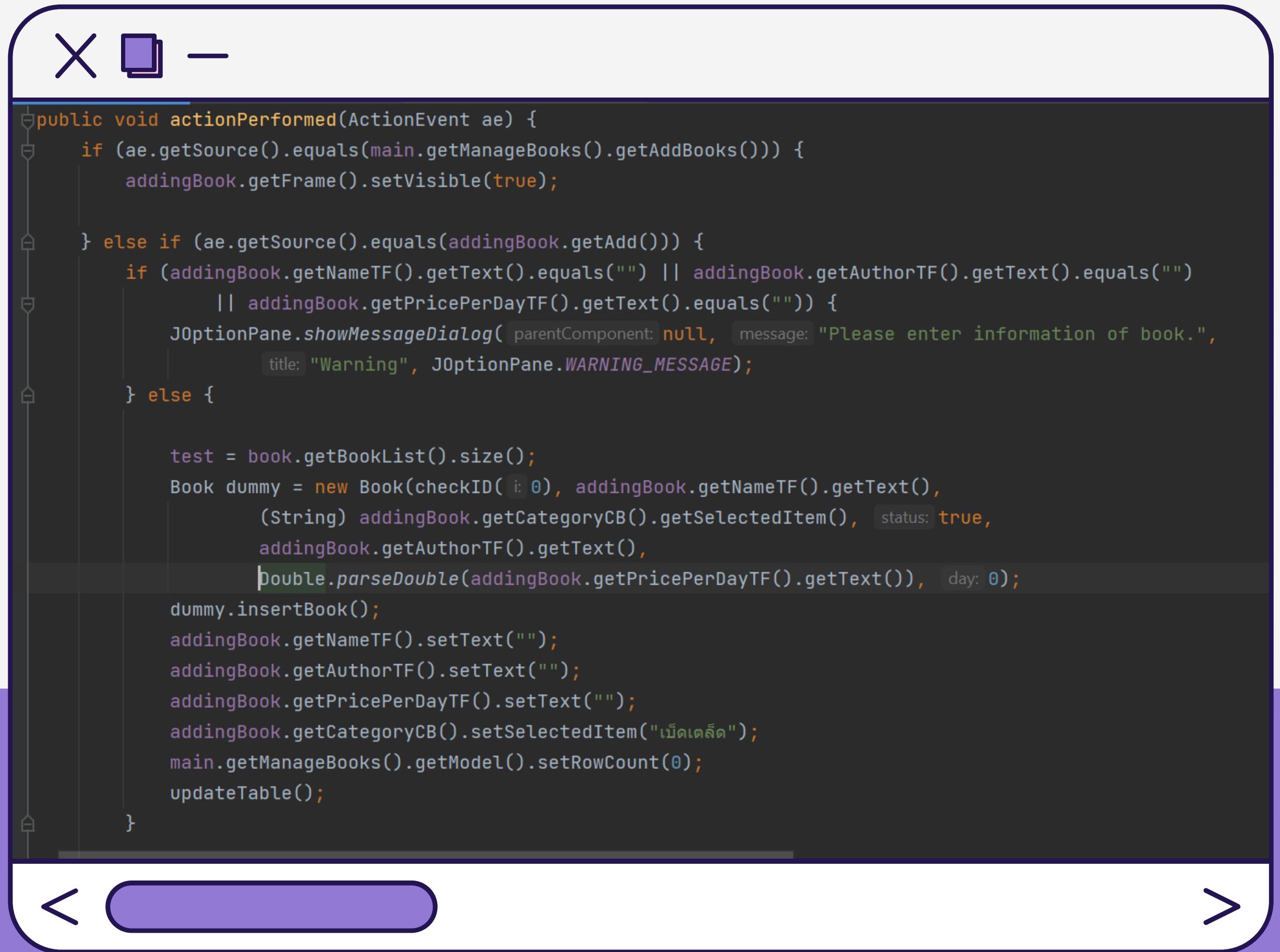


View

```
ManageBooks
+ searchTF JTextField
+ returnBook JButton
+ frame MDI
+ button JButton
+ model DefaultTableModel
+ pHeading JPanel
+ pReturnBook JPanel
+ pSearch JPanel
+ search JButton
+ pAddBook JPanel
+ heading JLabel
+ bookCode JLabel
+ tableOfInfoBooks JTable
+ pHeadingAndSearch JPanel
+ controller ManageBookController
+ searchCB JComboBox
+ scroll JScrollPane
+ bookCodeTF JTextField
+ back JButton
+ pAddBookAndReturnBook JPanel
+ addBooks JButton
+ pBack JPanel
+ pHeadingAndBack JPanel
+ ManageBooks(boolean, boolean, boolean, boolean, ManageBookController, MDI)
+ getBookCodeTF JTextField
+ setModel(DefaultTableModel)
+ getHeading JLabel
+ getSearchCB JComboBox
+ getAddBook JPanel
+ internalFrameOpened(InternalFrameEvent)
+ internalFrameClosing(InternalFrameEvent)
+ getAddBookAndReturnBook JPanel
+ setHeading JLabel
+ getSearch JPanel
+ internalFrameConfigured(InternalFrameEvent)
+ setHeadingAndBack JPanel
+ setAddBook JPanel
+ getHeadingAndSearch JPanel
+ setFrame MDI
+ getFrame JPanel
+ getHeadingAndBack JPanel
+ setpBack JPanel
+ setScroll JScrollPane
+ getBack JButton
+ internalFrameDeiconified(InternalFrameEvent)
+ getReturnBook JButton
+ getHeading JPanel
+ setAddBooks JButton
+ setFrame MDI
+ getBookCode JLabel
+ setBookCodeTR JTextField
+ setSearchTR JTextField
+ setBack JButton
+ getButton JButton
+ init()
+ getScroll JScrollPane
+ getModel DefaultTableModel
+ setSearch JButton
+ setHeading JPanel
+ internalFrameOpened(InternalFrameEvent)
+ internalFrameActivated(InternalFrameEvent)
+ setpAddBookAndReturnBook JPanel
+ setSearchCB JComboBox
+ setpReturnBook JPanel
+ getpBack JButton
+ getpReturnBook JButton
+ setButton JButton
+ setBookCode JLabel
+ setSearch JPanel
+ getController()
+ setTableOfInfoBooks JTable
+ setpHeadingAndSearch JPanel
+ setController(ManageBookController)
+ getTableOfInfoBooks JTable
+ getSearch JButton
+ internalFrameActivated(InternalFrameEvent)
+ getAddBooks JButton
+ getSearchTF JTextField
+ setReturnBook JButton
```

— Controller —

```
ManageBookController
+ button JButton
+ keepRow int
+ addingBook InsertBook
+ test int
+ main MDI
+ infoBook InfoBook
+ book Book
+ manageBooks ManageBooks
+ btn ButtonRenderer
+ ManageBookController(MDI)
+ getBtn() ButtonRenderer
+ getBooks() List<Book>
+ actionPerformed(ActionEvent) void
+ getTest() int
+ getButton() JButton
+ setMain(MDI) void
+ setButton(JButton) void
+ setBook(Book) void
+ getAddingBook() InsertBook
+ checkID(int) int
+ setBtn(ButtonRenderer) void
+ updateTable() void
+ getBook() Book
+ setAddingBook(InsertBook) void
+ setTest(int) void
+ getMain() MDI
```



The screenshot shows a Java code editor with a dark theme. The code is part of a class and handles an actionPerformed event. It checks if the source is a manage books frame or an adding book frame. If the source is the manage books frame, it makes the adding book frame visible. If the source is the adding book frame, it checks if all fields (name, author, category, price per day) are filled. If any field is empty, it shows a warning dialog. Otherwise, it creates a new book object with the provided information, inserts it, and then clears the adding book fields and updates the table.

```
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource().equals(main.getManageBooks().getAddBooks())) {
        addingBook.getFrame().setVisible(true);

    } else if (ae.getSource().equals(addingBook.getAdd())) {
        if (addingBook.getNameTF().getText().equals("") || addingBook.getAuthorTF().getText().equals("") || addingBook.getPricePerDayTF().getText().equals("")) {
            JOptionPane.showMessageDialog( parentComponent: null, message: "Please enter information of book.", title: "Warning", JOptionPane.WARNING_MESSAGE);
        } else {
            test = book.getBookList().size();
            Book dummy = new Book(checkID( i: 0 ), addingBook.getNameTF().getText(), (String) addingBook.getCategoryCB().getSelectedItem(), status: true, addingBook.getAuthorTF().getText(), double.parseDouble(addingBook.getPricePerDayTF().getText()), day: 0);
            dummy.insertBook();
            addingBook.getNameTF().setText("");
            addingBook.getAuthorTF().setText("");
            addingBook.getPricePerDayTF().setText("");
            addingBook.getCategoryCB().setSelectedItem("เบ็ดเตล็ด");
            main.getManageBooks().getModel().setRowCount(0);
            updateTable();
        }
    }
}
```

## ส่วนของการเพิ่มหนังสือ

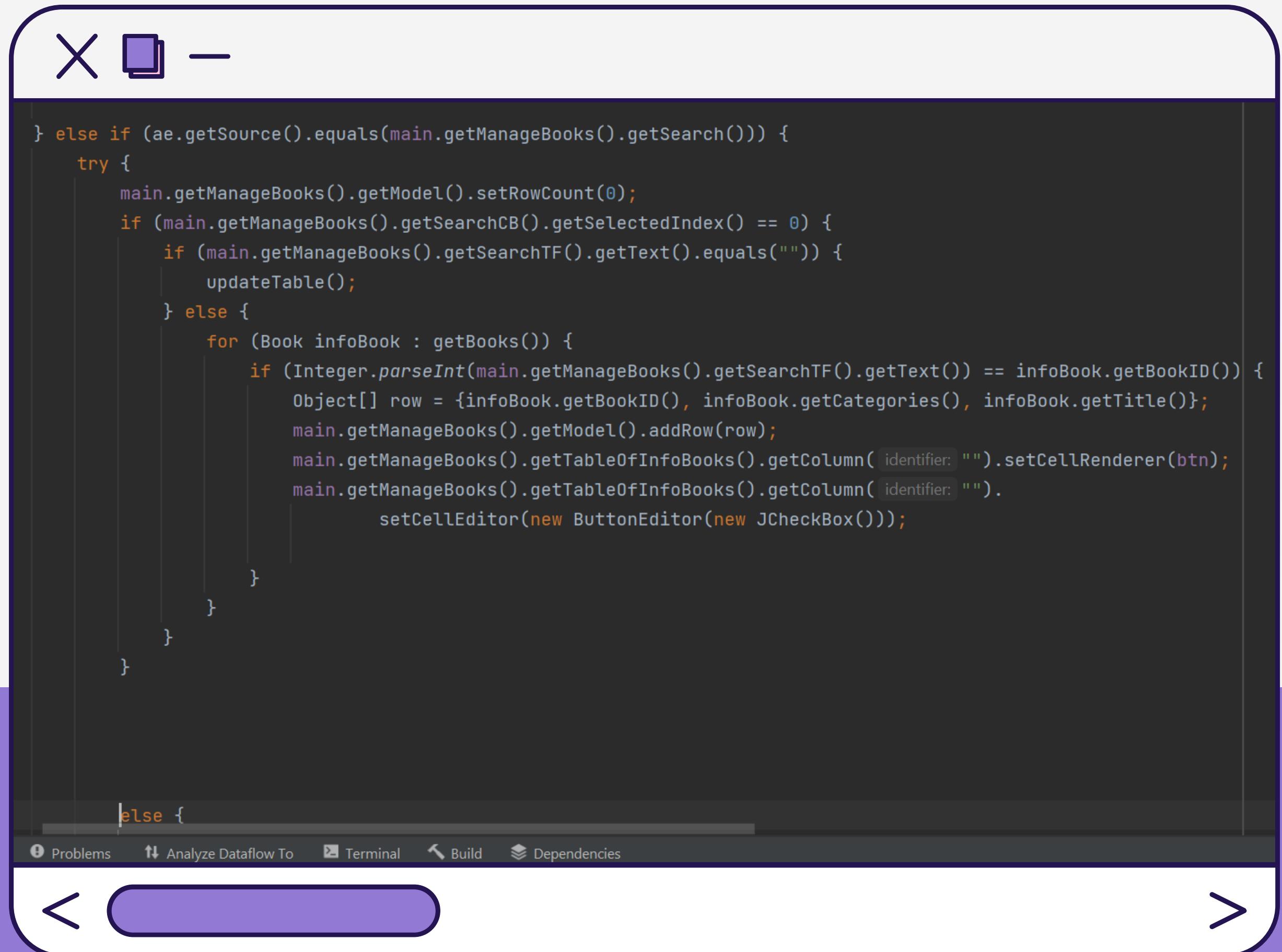




```
public List<Book> getBooks() {  
    Book dummy = new Book();  
    return dummy.getBookList();  
}  
  
public void updateTable() {  
    main.getManageBooks().getModel().setRowCount(0);  
    for (Book infoBook : getBooks()) {  
        if (infoBook.isStatus() == true) {  
            Object[] row = {infoBook.getBookID(), infoBook.getCategories(), infoBook.getTitle()};  
            main.getManageBooks().getModel().addRow(row);  
            main.getManageBooks().getTableOfInfoBooks().getColumn( identifier: "") .setCellRenderer(btn  
            main.getManageBooks().getTableOfInfoBooks().getColumn( identifier: "") .  
                setCellEditor(this.new ButtonEditor(new JCheckBox())));  
        }  
    }  
}
```

ส่วนของ updateTable





```
        } else if (ae.getSource().equals(main.getManageBooks().getSearch())) {  
            try {  
                main.getManageBooks().getModel().setRowCount(0);  
                if (main.getManageBooks().getSearchCB().getSelectedIndex() == 0) {  
                    if (main.getManageBooks().getSearchTF().getText().equals("")) {  
                        updateTable();  
                    } else {  
                        for (Book infoBook : getBooks()) {  
                            if (Integer.parseInt(main.getManageBooks().getSearchTF().getText()) == infoBook.getBookID()) {  
                                Object[] row = {infoBook.getBookID(), infoBook.getCategories(), infoBook.getTitle()};  
                                main.getManageBooks().getModel().addRow(row);  
                                main.getManageBooks().getTableOfInfoBooks().getColumn(identifier: "").setCellRenderer(btn);  
                                main.getManageBooks().getTableOfInfoBooks().getColumn(identifier: "").  
                                    setCellEditor(new ButtonEditor(new JCheckBox()));  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
  
    else {  
        //  
    }  
}
```

Problems Analyze Dataflow To Terminal Build Dependencies

ส่วนการค้นหาหนังสือ





The screenshot shows a Java code editor window with a dark theme. The code is as follows:

```
se {
    for (Book infoBook : getBooks()) {
        if (main.getManageBooks().getSearchTF().getText().equals("")) {
            updateTable();
        } else {
            if (main.getManageBooks().getSearchTF().getText().equals(infoBook.getTitle())) {
                Object[] row = {infoBook.getBookID(), infoBook.getCategories(), infoBook.getTitle()};
                main.getManageBooks().getModel().addRow(row);
                main.getManageBooks().getTableOfInfoBooks().getColumn(identifier: "").setCellRenderer(btn);
                main.getManageBooks().getTableOfInfoBooks().getColumn(identifier: "").setCellEditor(new ButtonEditor(new JCheckBox()));
            }
        }
    }

    h (NumberFormatException e) {
        JOptionPane.showMessageDialog( parentComponent: null,
            message: "Please enter a valid book ID.", title: "Warning", JOptionPane.WARNING_MESSAGE);
    }
}
```

The code implements a search function for books. It iterates through a list of books and adds them to a table if the search text matches the book's title. It also handles the case where no search text is provided by calling the `updateTable()` method. An exception block handles `NumberFormatException` by displaying a warning message to the user.

ส่วนการค้นหาหนังสือ(ต่อ)





```
else if (ae.getSource().equals(button)) {
    int row;
    if (main.getManageBooks().getTableOfInfoBooks().getSelectedRow() == -1) {
        row = keepRow;
    } else {
        row = main.getManageBooks().getTableOfInfoBooks().getSelectedRow();
        keepRow = row;
    }

    for (Book checkBook : getBooks()) {
        if (main.getManageBooks().getTableOfInfoBooks().getModel().getValueAt(row, columnIndex: 0)
            .equals(checkBook.getBookID())) {
            infoBook.getBookId2().setText(String.valueOf(checkBook.getBookID()));
            infoBook.getBookNameTF().setText(checkBook.getTitle());
            infoBook.getCategoryCB().setSelectedItem(checkBook.getCategories());
            infoBook.getAuthorTF().setText(checkBook.getWriter());
            infoBook.getPricePerDayTF().setText(String.valueOf(checkBook.getPrice()));
        }
    }
    infoBook.getFrame().setVisible(true);
}
```

ส่วนของการแสดงข้อมูล  
หนังสือ



```
while(index < member.getMemberList().size()){
    if(member.getMemberList().get(index).getId().equals(id.getText())==false){
        index++;
        check = 0;
    }
    else{
        check = 1;
        break;
    }
}
if(check == 0){
    JOptionPane.showMessageDialog(frame, message: "Invalid Member ID"); //ราชadam Collection หนังสือ
}
else{
    JOptionPane.showMessageDialog(frame, message: Controller.Rent_Controller.bookList.size() + " Book(s)\nTotal Price: " + totalprice * (0.95));
    this.frame.getMemberVC().setVisible(false);
    this.frame.getMemberVC().hide();
    this.frame.getDesktopPane().remove(this.frame.getMemberVC());
    this.frame.getDesktopPane().add(this.frame.getMainPage());
    this.frame.getMainPage().setVisible(true);
    this.frame.getMainPage().show();
    Controller.Rent_Controller.bookList.clear();
}

if(ae.getSource().equals(back)){
    this.frame.getMemberVC().setVisible(false);
    this.frame.getMemberVC().hide();
    this.frame.getDesktopPane().remove(this.frame.getMemberVC());
    this.frame.getDesktopPane().add(this.frame.getPricePreview());
    this.frame.getPricePreview().setVisible(true);
    this.frame.getPricePreview().show();
}
```

MemberVC

f  p_insert	JPanel
f  member	Member
f  accept	JButton
f  p_buttons	JPanel
f  book	Book
f  insert	JLabel
f  back	JButton
f  p_id	JPanel
f  id	JPasswordField
f  f1	Font
f  id_insert	JLabel
f  frame	MDI
m  MemberVC(String, boolean, boolean, boolean, boolean, MDI)	
m  actionPerformed(ActionEvent)	void
m  init()	void

ส่วนการเช่าแบบมีสมาชิก



```
(ae.getSource().equals(next)){
    //เช็ค ข้อมูล ถ้าถูกต้อง Alert ขึ้น และกลับหน้าเช่าหนังสือ
    if(fname.getText().equals("") || lname.getText().equals("") || address.getText().equals("") || contract.getText().equals ""){
        JOptionPane.showMessageDialog(frame, message: "Please Insert Name or Address or Contract.");
    }
    else{
        JOptionPane.showMessageDialog(frame, message: Controller.Rent_Controller.bookList.size() + " Book(s)\nTotal Price: " + totalprice);
        NonMember nonmem = new NonMember(fname.getText(), lname.getText(), address.getText(), cardid.getText());
        nonmem.addNonmemberDetail();
        this.frame.getNonMemberVC().setVisible(false);
        this.frame.getNonMemberVC().hide();
        this.frame.getDesktopPane().remove(this.frame.getNonMemberVC());
        this.frame.getDesktopPane().add(this.frame.getMainPage());
        this.frame.getMainPage().setVisible(true);
        this.frame.getMainPage().show();
        Controller.Rent_Controller.bookList.clear();
    }
}

(ae.getSource().equals(back)){
    this.frame.getNonMemberVC().setVisible(false);
    this.frame.getNonMemberVC().hide();
    this.frame.getDesktopPane().remove(this.frame.getNonMemberVC());
    this.frame.getDesktopPane().add(this.frame.getPricePreview());
    this.frame.getPricePreview().setVisible(true);
    this.frame.getPricePreview().show();
}
```

NonMemberVC

f	p_header	JPanel
f	p_cardid	JPanel
f	book	Book
f	address_header	JLabel
f	p_contract	JPanel
f	contract	JTextField
f	address	JTextArea
f	member	Member
f	cardid	JTextField
f	sp_address	JScrollPane
f	back	JButton
f	cardid_header	JLabel
f	lname_header	JLabel
f	next	JButton
f	fname_header	JLabel
f	fname	JTextField
f	p_fname	JPanel
f	contract_header	JLabel
f	p_buttons	JPanel
f	header	JLabel
f	Iname	JTextField
f	p_Iname	JPanel
f	p_address	JPanel
f	f1	Font
f	frame	MDI
m	NonMemberVC(String, boolean, boolean, boolean, boolean, MDI)	
m	actionPerformed(ActionEvent)	void
m	init()	void
m	isNumber(char)	boolean

ส่วนการเช่าแบบไม่มีสมาชิก





กุลนิษฐ์ จุงาม  
63070011



พัฒนาส่วน View และ  
Controller ของหน้าเช่า  
หนังสือ



เกษมนี สถิติวิการวงศ์  
63070012



พัฒนาส่วนการเชื่อมต่อกับ  
ฐานข้อมูลของ MongoDB  
และ ส่วน Model

ชนกันต์ บุ่มน้อย  
63070027

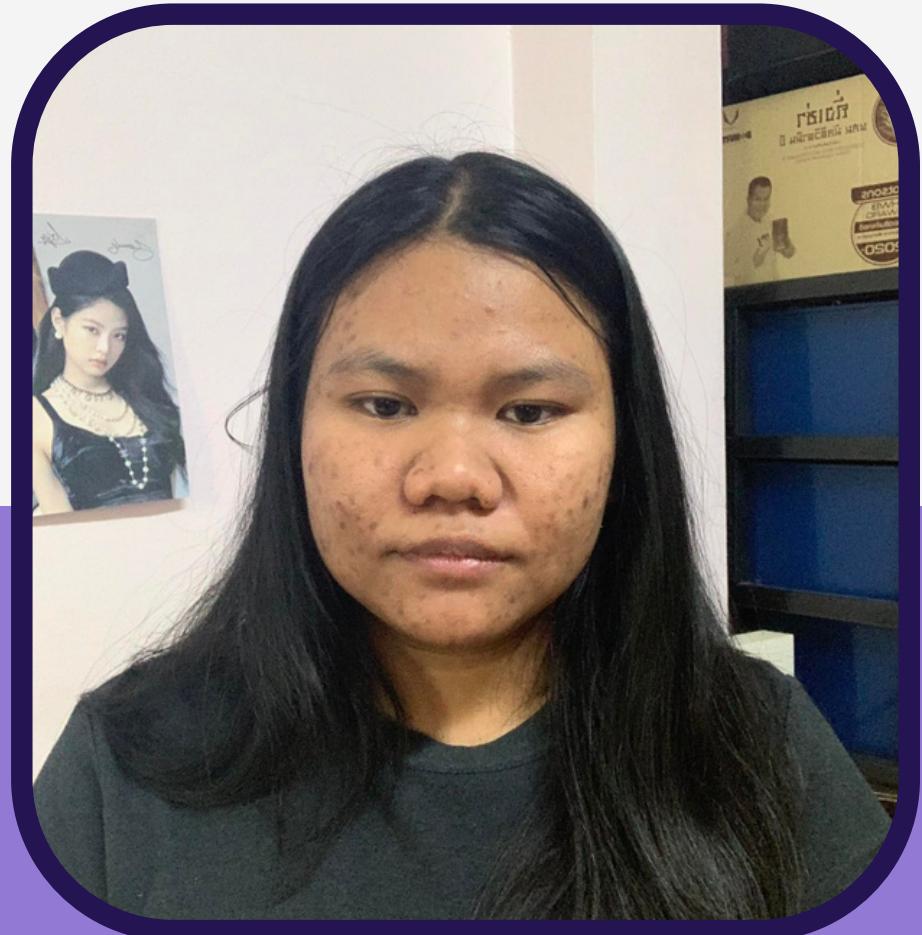


พัฒนาส่วน View และ  
Controller ของหน้าการ  
จัดการหนังสือ

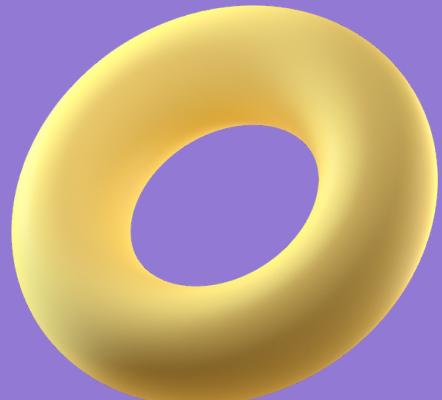




ณัชวลี จำปาสด  
63070061



พัฒนาส่วน View และ  
Controller ของหน้าหลัก  
เข้าสู่ระบบและ สมัครสมาชิก



ยศพล เอี่ยมอ่อน  
63070142



พัฒนาส่วน View และ  
Controller ของสรุปจำนวน  
ที่ลูกค้าต้องจ่าย และ  
ตกแต่ง UI





# Thank you!

