The End

- Cinema -

FOCUSKY

Group members

Yu Ren: Software Design, Core Code, Build Database

Jing Bian: Build Database, UI

Hao Dong: UI, Core Code

Xingzhou Li: presentation, Slides, Ui

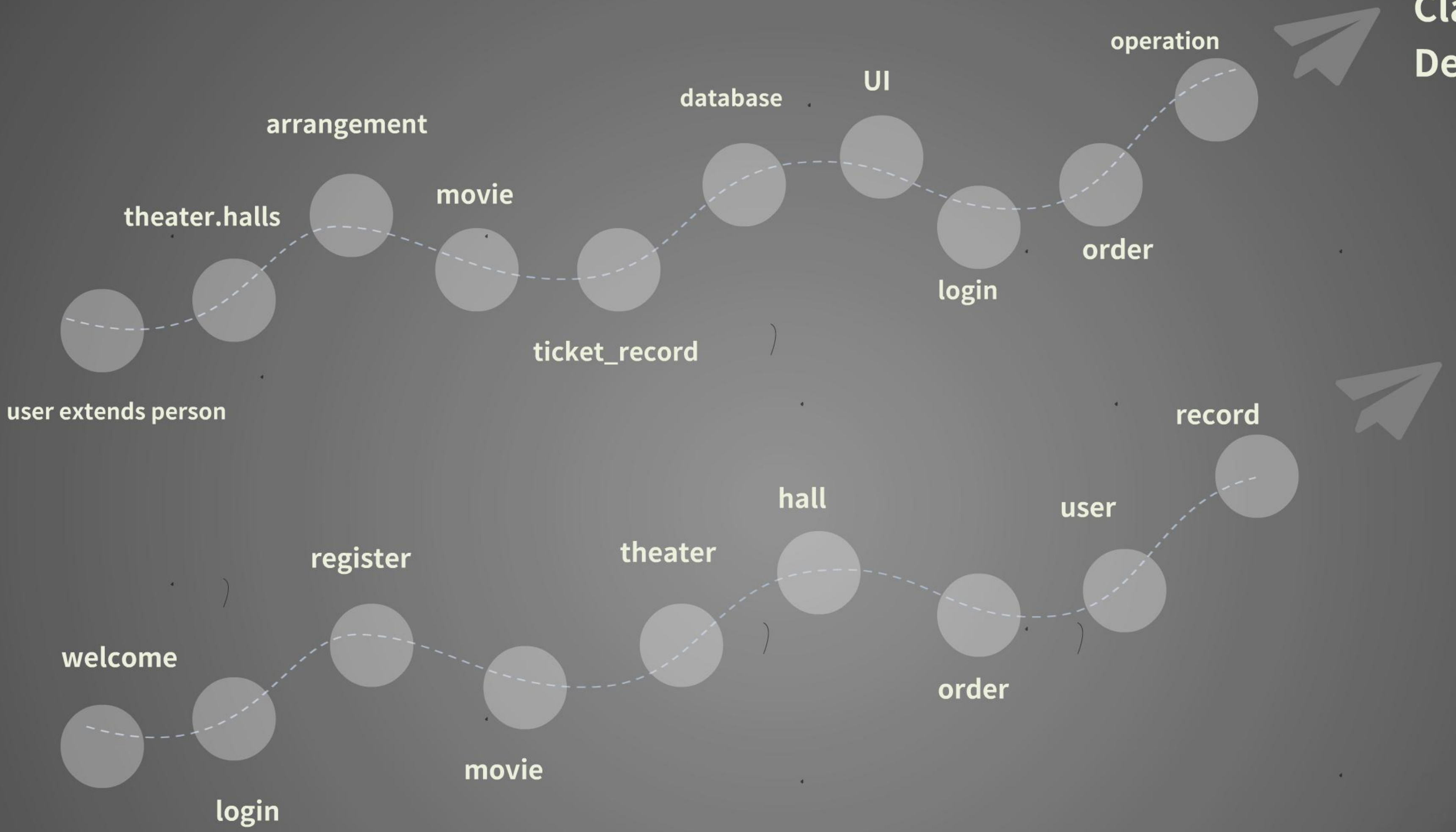
Jinnuo Che: Software Design, Core Code

Function Design

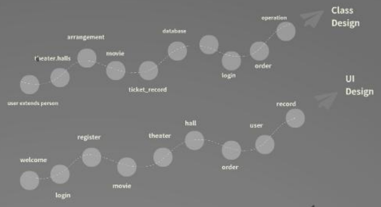
1. Users Management: Log In & Sign Up
2. Tickets Purchase:
    1)movie->theater->StartTime->NumberOfTicket
    2)theater->movie->StartTime->NumberOfTicket
3. Record Management:
    1)Look Up Purchase History
    2)Refund Tickets

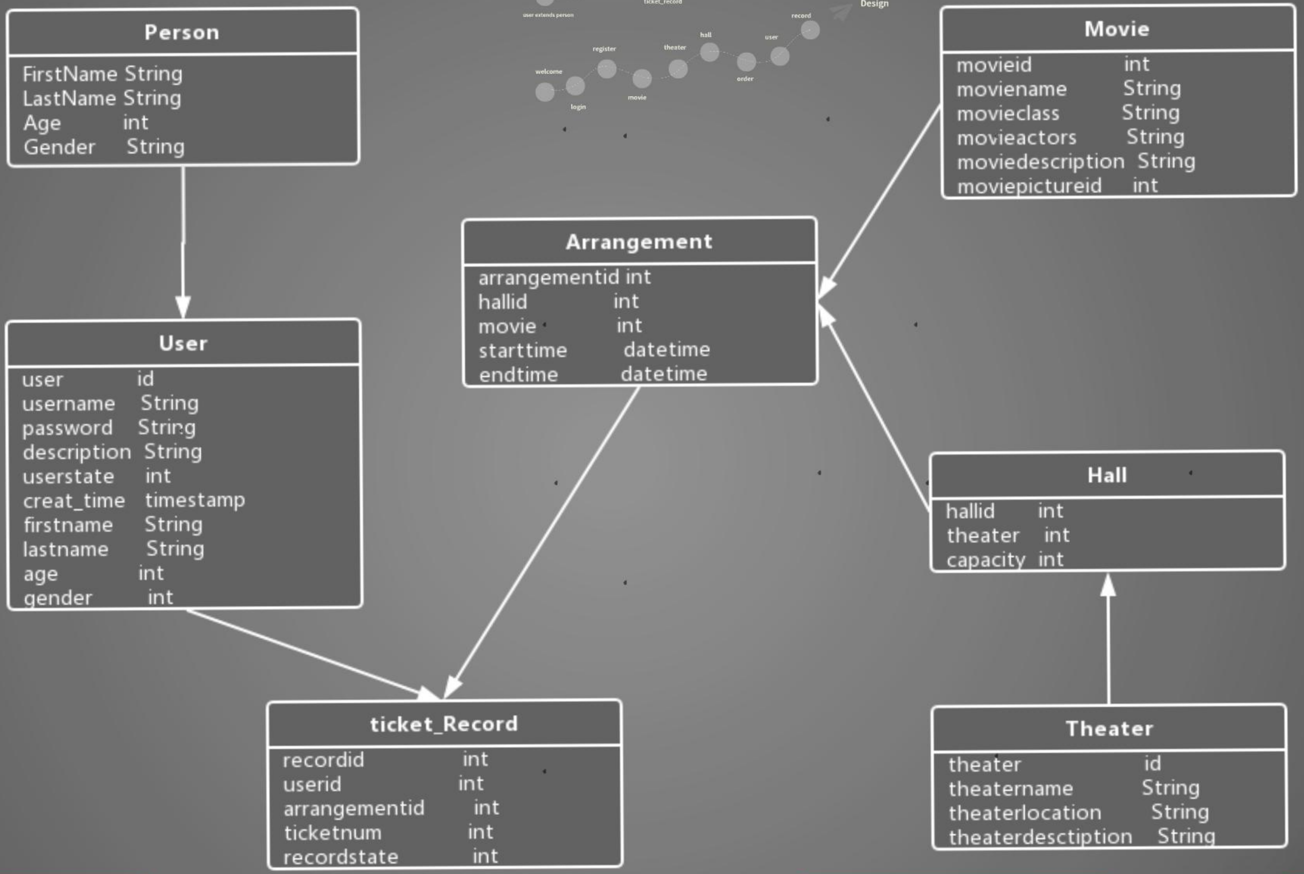
Class Design

UI Design



Database Design





Person

FirstName	String
LastName	String
Age	int
Gender	String

User

user	id
username	String
password	String
description	String
userstate	int
creat_time	timestamp
firstname	String
lastname	String
age	int
gender	int

Arrangement

arrangementid	int
hallid	int
movie	int
starttime	datetime
endtime	datetime

Movie

movieid	int
moviename	String
movieclass	String
movieactors	String
moviedescription	String
moviepictureid	int

Hall

hallid	int
theater	int
capacity	int

ticket_Record

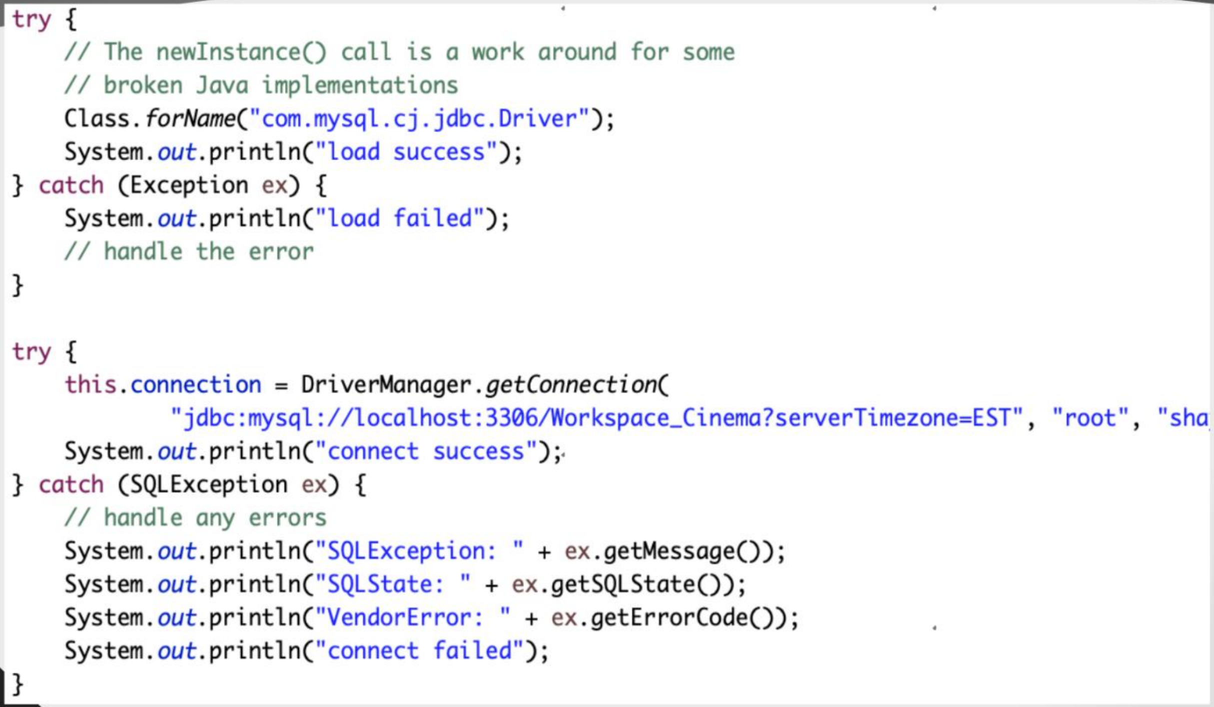
recordid	int
userid	int
arrangementid	int
ticketnum	int
recordstate	int

Theater

theater	id
theatername	String
theaterlocation	String
theaterdesctiption	String

Technological Challenges

Time deviation in JDBC

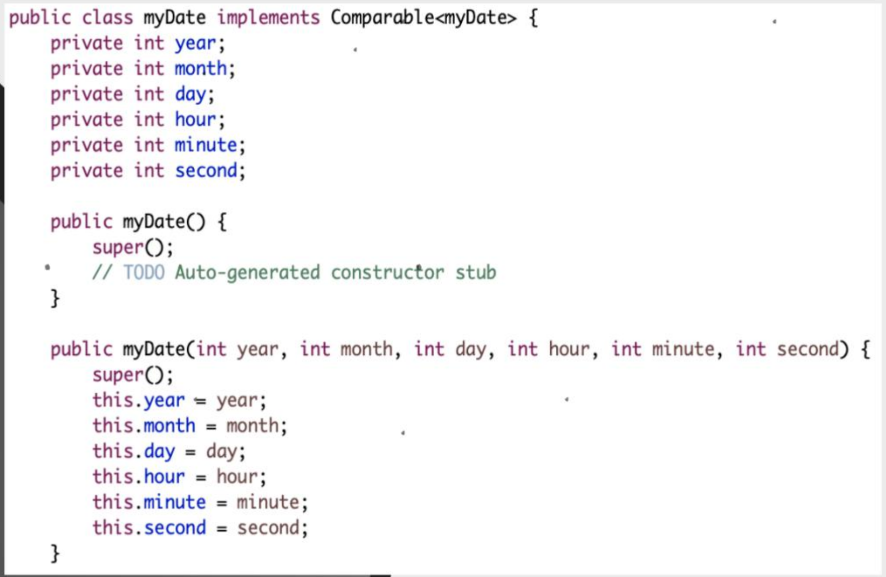


```
try {
    // The newInstance() call is a work around for some
    // broken Java implementations
    Class.forName("com.mysql.cj.jdbc.Driver");
    System.out.println("load success");
} catch (Exception ex) {
    System.out.println("load failed");
    // handle the error
}

try {
    this.connection = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/Workspace_Cinema?serverTimezone=EST", "root", "sha
    System.out.println("connect success");
} catch (SQLException ex) {
    // handle any errors
    System.out.println("SQLException: " + ex.getMessage());
    System.out.println("SQLState: " + ex.getSQLState());
    System.out.println("VendorError: " + ex.getErrorCode());
    System.out.println("connect failed");
}
```

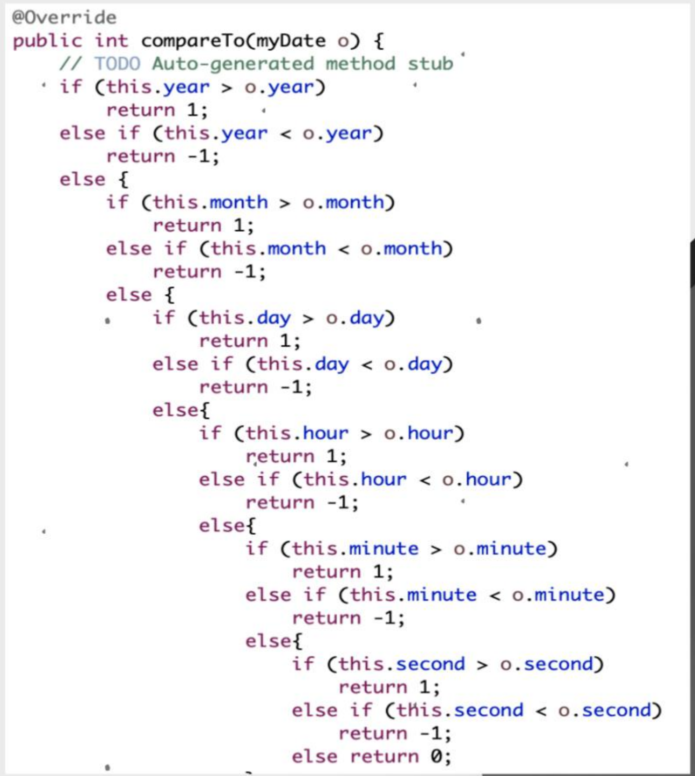
FOCUSKY

Technological Challenges

"myDate " Implements Comparable



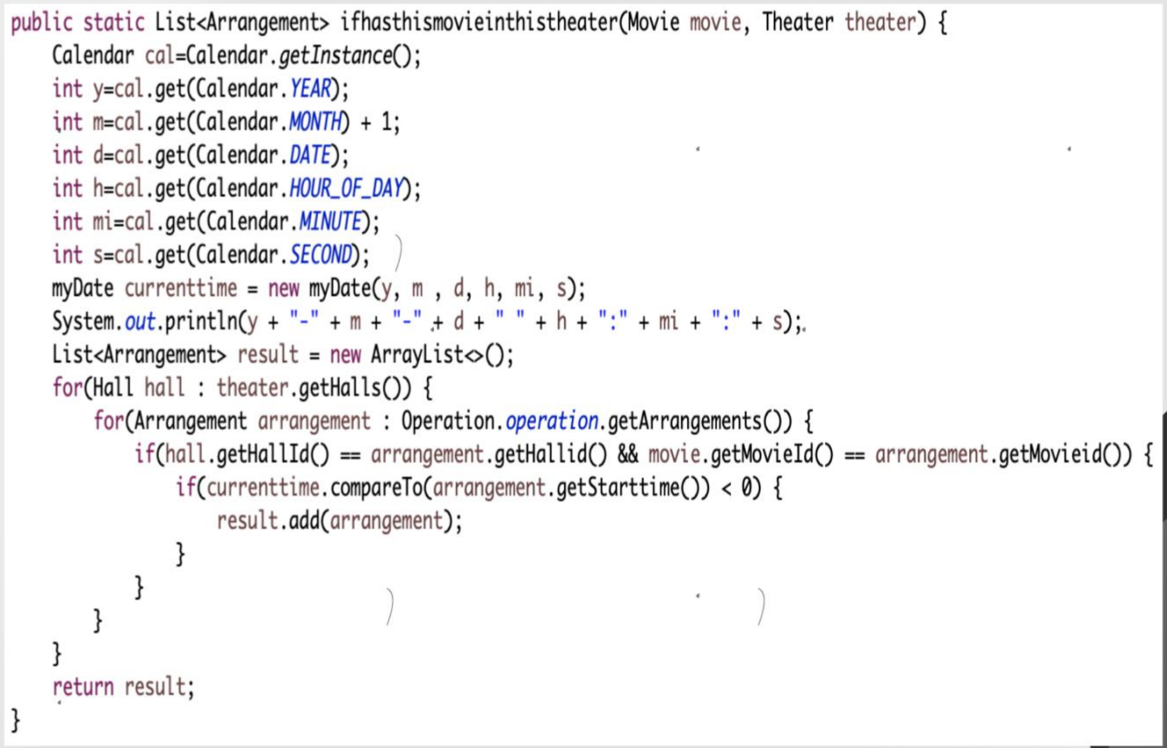
```java
public class myDate implements Comparable<myDate> {
    private int year;
    private int month;
    private int day;
    private int hour;
    private int minute;
    private int second;

    public myDate() {
        super();
        // TODO Auto-generated constructor stub
    }

    public myDate(int year, int month, int day, int hour, int minute, int second) {
        super();
        this.year = year;
        this.month = month;
        this.day = day;
        this.hour = hour;
        this.minute = minute;
        this.second = second;
    }
```



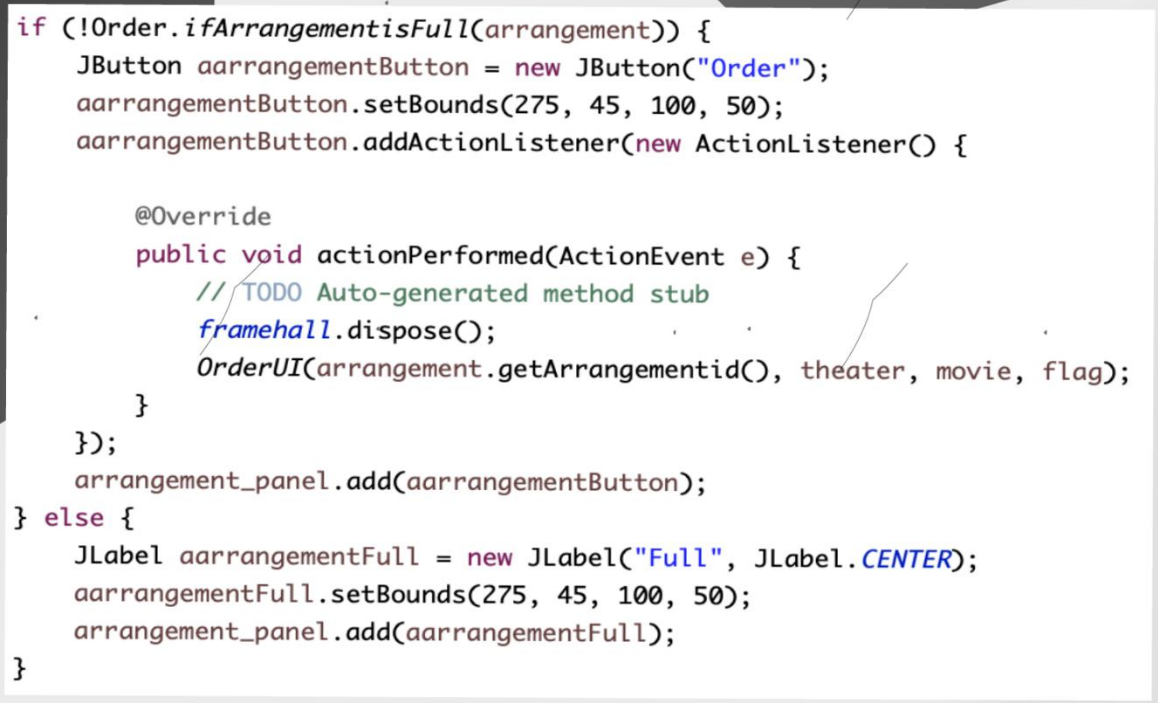
```java
@Override
public int compareTo(myDate o) {
    // TODO Auto-generated method stub
    if (this.year > o.year)
        return 1;
    else if (this.year < o.year)
        return -1;
    else {
        if (this.month > o.month)
            return 1;
        else if (this.month < o.month)
            return -1;
        else {
            if (this.day > o.day)
                return 1;
            else if (this.day < o.day)
                return -1;
            else{
                if (this.hour > o.hour)
                    return 1;
                else if (this.hour < o.hour)
                    return -1;
                else{
                    if (this.minute > o.minute)
                        return 1;
                    else if (this.minute < o.minute)
                        return -1;
                    else{
                        if (this.second > o.second)
                            return 1;
                        else if (this.second < o.second)
                            return -1;
                        else return 0;
```

Technological Challenges

Arrangement Display



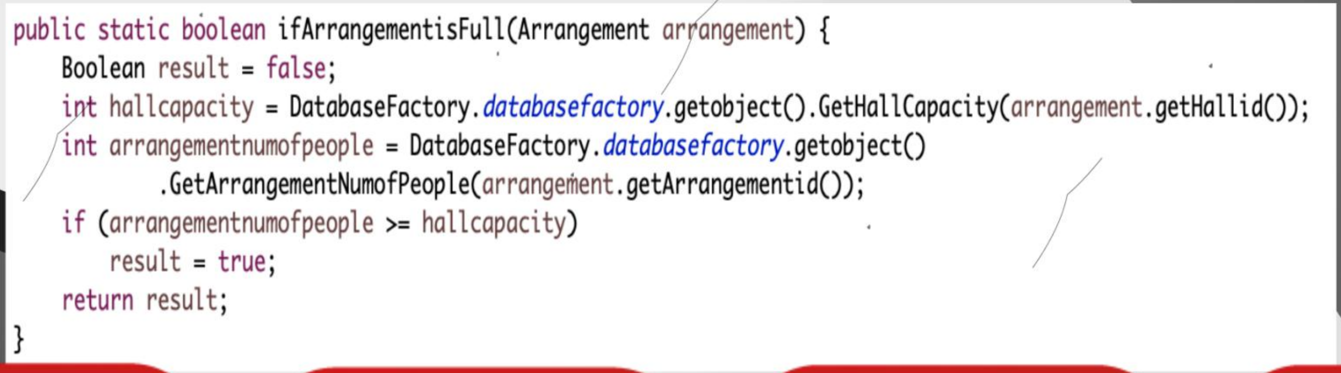
```java
public static List<Arrangement> ifhasthismovieinthistheater(Movie movie, Theater theater) {
    Calendar cal=Calendar.getInstance();
    int y=cal.get(Calendar.YEAR);
    int m=cal.get(Calendar.MONTH) + 1;
    int d=cal.get(Calendar.DATE);
    int h=cal.get(Calendar.HOUR_OF_DAY);
    int mi=cal.get(Calendar.MINUTE);
    int s=cal.get(Calendar.SECOND);
    myDate currenttime = new myDate(y, m , d, h, mi, s);
    System.out.println(y + "-" + m + "-" + d + " " + h + ":" + mi + ":" + s);
    List<Arrangement> result = new ArrayList<>();
    for(Hall hall : theater.getHalls()) {
        for(Arrangement arrangement : Operation.operation.getArrangements()) {
            if(hall.getHallId() == arrangement.getHallid() && movie.getMovieId() == arrangement.getMovieid()) {
                if(currenttime.compareTo(arrangement.getStarttime()) < 0) {
                    result.add(arrangement);
                }
            }
        }
    }
    return result;
}
```



```
if (!Order.ifArrangementisFull(arrangement)) {
    JButton aarrangementButton = new JButton("Order");
    aarrangementButton.setBounds(275, 45, 100, 50);
    aarrangementButton.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent e) {
            // TODO Auto-generated method stub
            framehall.dispose();
            OrderUI(arrangement.getArrangementid(), theater, movie, flag);
        }
    });
    arrangement_panel.add(aarrangementButton);
} else {
    JLabel aarrangementFull = new JLabel("Full", JLabel.CENTER);
    aarrangementFull.setBounds(275, 45, 100, 50);
    arrangement_panel.add(aarrangementFull);
}
```

What will happen

if the hall is full ?

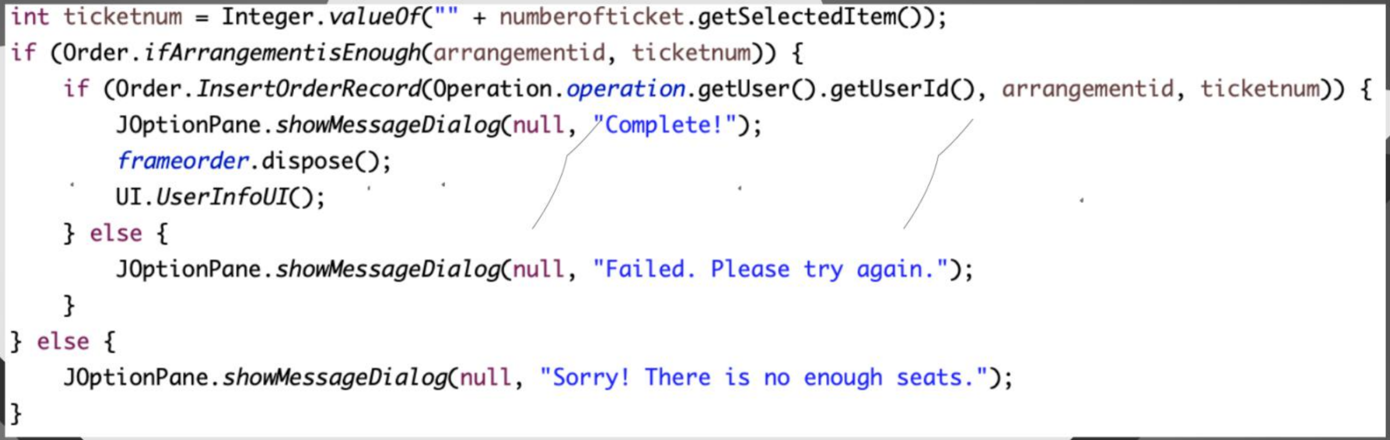


```
public static boolean ifArrangementisFull(Arrangement arrangement) {
    Boolean result = false;
    int hallcapacity = DatabaseFactory.databasefactory.getobject().GetHallCapacity(arrangement.getHallid());
    int arrangementnumofpeople = DatabaseFactory.databasefactory.getobject()
            .GetArrangementNumofPeople(arrangement.getArrangementid());
    if (arrangementnumofpeople >= hallcapacity)
        result = true;
    return result;
}
```

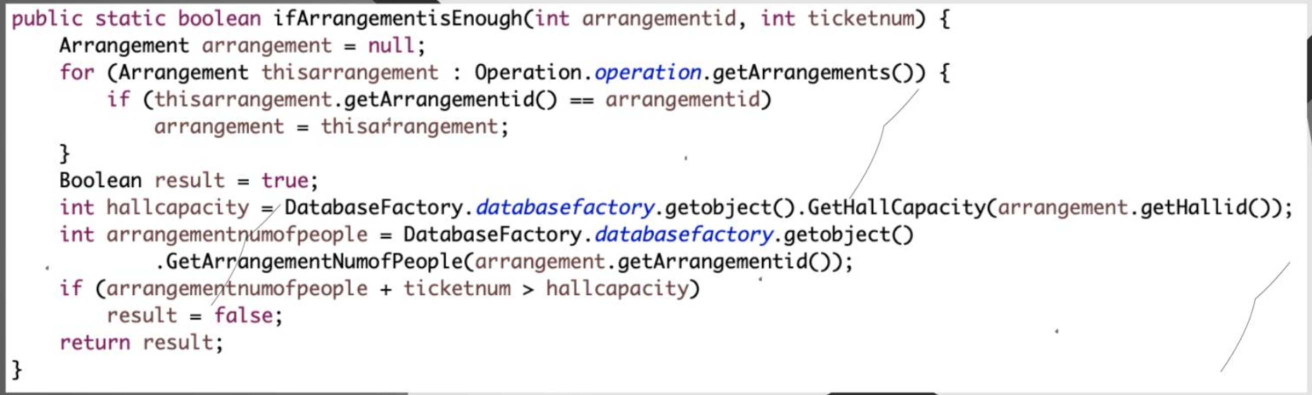
Technological Challenges

what will happen

if the hall is going to full?



```java
int ticketnum = Integer.valueOf("" + numberofticket.getSelectedItem());
if (Order.ifArrangementisEnough(arrangementid, ticketnum)) {
    if (Order.InsertOrderRecord(Operation.operation.getUser().getUserId(), arrangementid, ticketnum)) {
        JOptionPane.showMessageDialog(null, "Complete!");
        frameorder.dispose();
        UI.UserInfoUI();
    } else {
        JOptionPane.showMessageDialog(null, "Failed. Please try again.");
    }
} else {
    JOptionPane.showMessageDialog(null, "Sorry! There is no enough seats.");
}
```

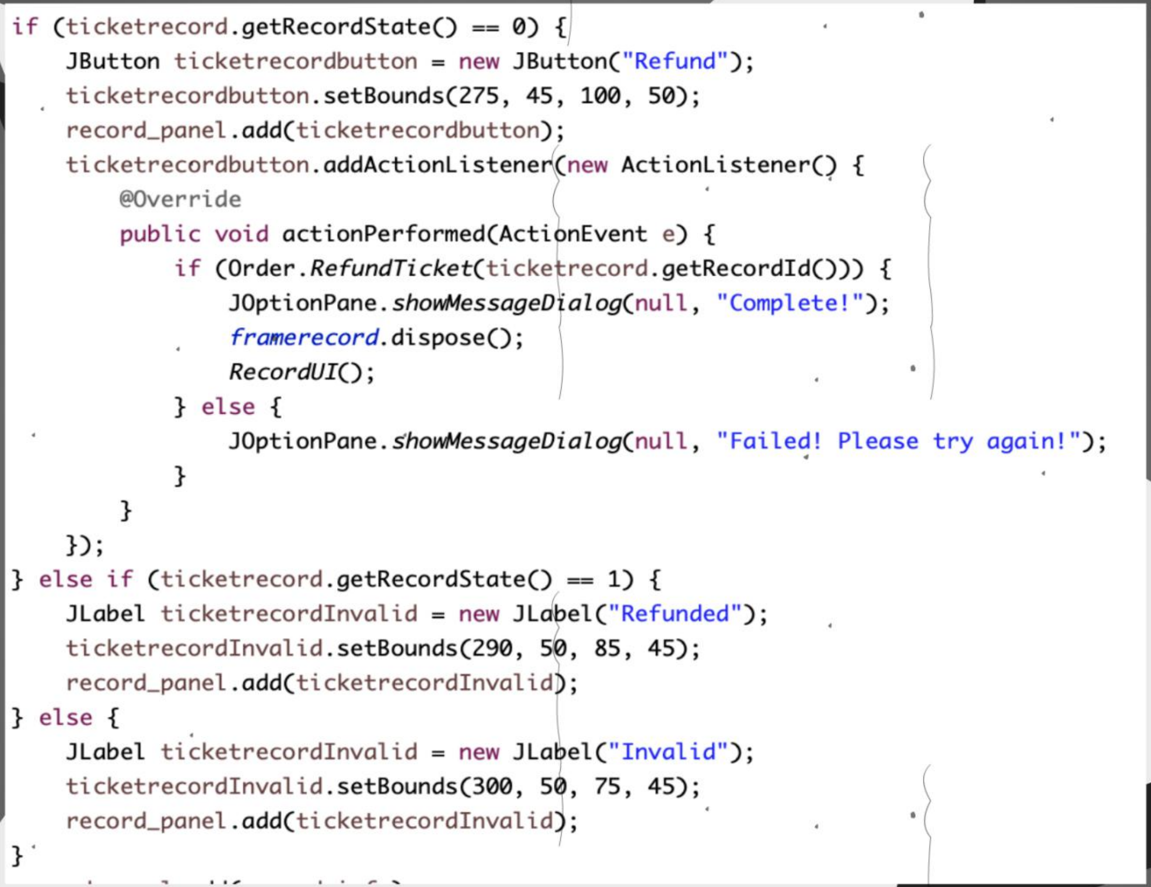


```java
public static boolean ifArrangementisEnough(int arrangementid, int ticketnum) {
    Arrangement arrangement = null;
    for (Arrangement thisarrangement : Operation.operation.getArrangements()) {
        if (thisarrangement.getArrangementid() == arrangementid)
            arrangement = thisarrangement;
    }
    Boolean result = true;
    int hallcapacity = DatabaseFactory.databasefactory.getobject().GetHallCapacity(arrangement.getHallid());
    int arrangementnumofpeople = DatabaseFactory.databasefactory.getobject()
            .GetArrangementNumofPeople(arrangement.getArrangementid());
    if (arrangementnumofpeople + ticketnum > hallcapacity)
        result = false;
    return result;
}
```

FOCUSKY

Technological Challenges

If users **refund tickets**, how to deal with the **records state?**



```
if (ticketrecord.getRecordState() == 0) {
    JButton ticketrecordbutton = new JButton("Refund");
    ticketrecordbutton.setBounds(275, 45, 100, 50);
    record_panel.add(ticketrecordbutton);
    ticketrecordbutton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (Order.RefundTicket(ticketrecord.getRecordId())) {
                JOptionPane.showMessageDialog(null, "Complete!");
                framerecord.dispose();
                RecordUI();
            } else {
                JOptionPane.showMessageDialog(null, "Failed! Please try again!");
            }
        }
    });
} else if (ticketrecord.getRecordState() == 1) {
    JLabel ticketrecordInvalid = new JLabel("Refunded");
    ticketrecordInvalid.setBounds(290, 50, 85, 45);
    record_panel.add(ticketrecordInvalid);
} else {
    JLabel ticketrecordInvalid = new JLabel("Invalid");
    ticketrecordInvalid.setBounds(300, 50, 75, 45);
    record_panel.add(ticketrecordInvalid);
}
```

Improvements

01 Show **seats map** in each hall

02 **Fuzzy search**

03 Add **wallets** to each user

04 **Filter** movies according to types or actors...

The End

- Cinema -