**Group 19:**

Nguyễn Võ Thanh Tâm – ITITWE19012

Trần Thị Huỳnh Như – ITITIU19174

Đỗ Hoàng Nhung – ITITIU19176

**ORIENTED-OBJECT PROGRAMING REPORT**

Project: Movie tickets booking

**I. Abstract:**

- Supporting the cinema to manage, organize the schedule as well as movie rooms.

- Supporting customer to book tickets, opt the showtime and cinema seats.

**II. Information:**

**- Users:**

+ Manager

+ Customer

**- Function:**

+ Manager:

• Add movie and edit movie information

• Organize the movie schedule

• Arrange movie rooms

+ Customer:

• Read description, watch trailer, get information relating to showroom and schedule

• Book tickets

• Manage the order

**III. Materials and methods:**

**- Design database:**

+ hour: store the given time slot

+ timer: store date and time slot

+ movie: store movie information

+ showCalendar: save id of room, timer and movie

+ bill: save tickets bill

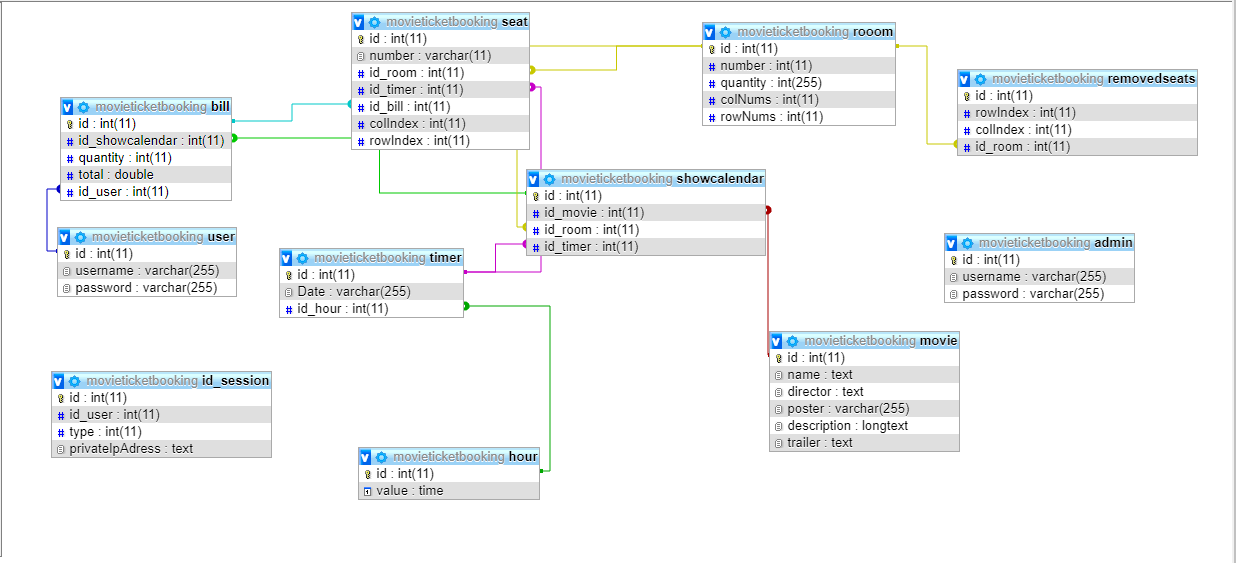
+ room: save room information

+ removedSeat: save seating chart

+ user: customer account

+ admin: manager account

+ Id\_session: save logging in - user information



**- Design theme:**

+ Some tools in java:

• Jcombobox: <https://stackoverflow.com/questions/4822928/jcombobox-string-item-visible-and-integer-key-inherent>

• Label: to display image, movie information,…

• Button: event click, action performed, how to disabled button

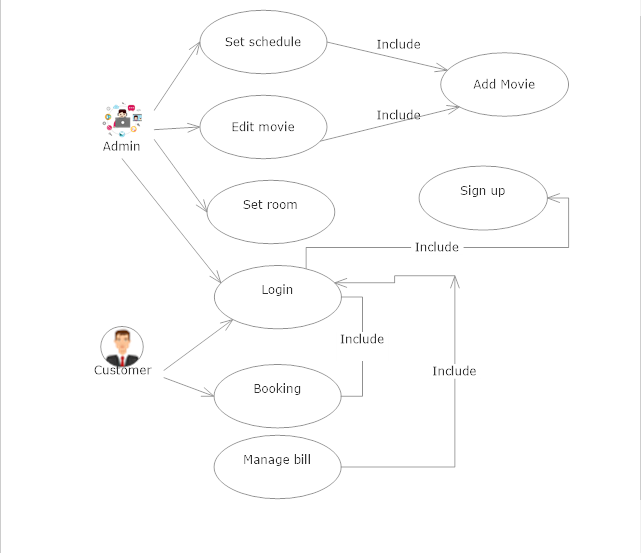
• Jtextfield: to input movie information

• jText Area: to display movie description

• Jtable: to display list of orders

• Messager dialog: to display error massage

**- User case:**



**- Algorithm for function:**

+ Upload image in java: <https://www.youtube.com/watch?v=UusZGBkV6HI>

+ Slide show : <https://www.youtube.com/watch?v=pN1uoJD_uSE>

+ How to avoid schedule conflict?

Schedule consists of 3 attributes: Movie title, schedule, room

• Select title, then room, then schedule → use statement select and where

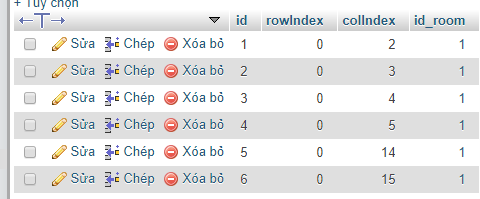
• When schedule conflict happens, schedule which are chose will be saves in a vector → when another schedule display, corresponding will be hidden.

+ Create room:

• To save seat chart, we need matrix (row × column) and used seat

• Initialize a (m × n) matrix corresponding to the number of rows and columns, it will be saved in database in form of the number of rows and columns.

• To recognize the booked seat, we need to assign to seat buttons: it is “none” → index at that position equals 0 in database, it is “booked” → index equals 1.

Eg: 

The index in room 1: (0,2), (0,3), (0,4), (0,5), (0,14), (0,15) will be deleted.

+ Login id session system:

**IV. Result:**

**1. Create room**

- Managers can build seat chart by create a matrix, then they will choose some available positions for use in that matrix

**2. Decentralize access:**

- Devide the system into 2 theme: admin and customer, each have their own account

- Develop login system

**3. Adjust information slide**

- Admin can adjust information slide of movie

**4. Login system**

- Maintain login state

**V. References**

1. CGV Application: <https://play.google.com/store/apps/details?id=com.cgv.cinema.vn&hl=vi>

2. Movie ticket booking application: <https://www.youtube.com/watch?v=xk2a979kHSc&t=54s>

3.Stackoverflow