

UNIVERSITY OF JAFFNA
FACULTY OF ENGINEERING
EC5080: SOFTWARE CONSTRUCTION
IN-COURSE ASSESSMENT NO 3

2018/E/073

Movie Reservation System

Multiplexer Theatre Showing Screen 1

SCREEN

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	1	2	3	4	5	6	7	8	9	10
3										
4	1	2	3	4	5	6	7	8		
5	1	2	3	4	5	6	7	8	9	10
6	1	2	3	4	5	6	7	8	9	10
7	1	2	3	4	5	6	7	8	9	10
8	1	2	3	4	5	6	7	8	9	10
9	1	2	3	4	5	6	7	8	9	10
10			3	4	5	6	7	8		

Movie : Ginderdown
Time : July 12, 21.00
Tickets : 0
Total : \$ 0
Seats :

Book Now

☐ Available
☐ Sold
☐ Selected

1) Why you choose that specific IDE and framework to develop your application?

- I selected NetBeans as my IDE. Because GUI applications are easier to build by NetBeans for me. I could able to handle components easily with this IDE. And also most of codes were auto generated. So it is fast and easy.
- I selected swing as my framework. Because it got more component types and features. It has standard GUI library. So it is platform independent. So I could able to work with swing easier in NetBeans IDE.

2) What are the problems you faced and what are solution you found for those problems during implementation?

- There were 84 seats (Buttons). Handling them in the frame was bit harder, but with NetBeans it was easier. Because of that easiness I was able to build the interface of seats, but I again faced a problem that is the button texts were messed. Because of command function are auto generating and allow to access them easily in NetBeans it was little bit easier for my coding.
- Making commands for each buttons were very hard and complex. I found that 84 buttons can easily divided into two parts as normal (seats except last row) and special. So for those buttons I used mainly 2 functions. They also have some depended functions. So I created some functions for get a less complex code.
- To seat numbers I used line and column numbers (ex: 1-1, 10-6). And when printing them in the interface I faced a huge problem that is while entering more seats the label and window were getting larger and it was affected the interface badly. For that I used a JTextArea component to store the seat numbers. I disabled the editing option so user could not type on it. The biggest advantage of this component for me was when user selecting more seats it is able to add all of them with providing scrolling horizontally facility. So we can check seat numbers with it very easily.
- Finally when the window is resizing it was affected the interface badly. So I denied resizing option.

3) Detailed document how your application works?

- There are 84 seats (buttons). Initially they are in gray color.
- When user select a seat (gray colored) following operations will executed.
 - Seat color changes to green,
 - increasing (calculating) the total cost,
 - Increasing the number of tickets,
 - add the number of the seat in a string and
 - Show them in the right hand (panel) of interface.
- User can select multiple number of seats.
- If user select a seat which was already selected (green colored) following operations will executed.
 - Seat color changes to gray,
 - decreasing (calculating) the total cost,
 - Decreasing the number of tickets,
 - remove the number of the seat from the string and
 - Show them in the right hand (panel) of interface.

- After selecting seats user can use 'Book Now' button for the reservation. Then,
 - selected seats (green colored) change to red color,
 - Summary of the reservation will displayed,
 - Updated total cost, number of tickets values and saved seat numbers will initialized and
 - Show them in the right hand (panel) of interface.
- User cannot select any reserved (red colored) seat.
- It will be an unsuccessful reserving if user press 'Book now' button without selecting at least one seat.
- If all seats were reserved user cannot book seats anymore ('Book Now' option will disabled).