

TicketPayment

Cocoş Tudor-Adrian

E_30423

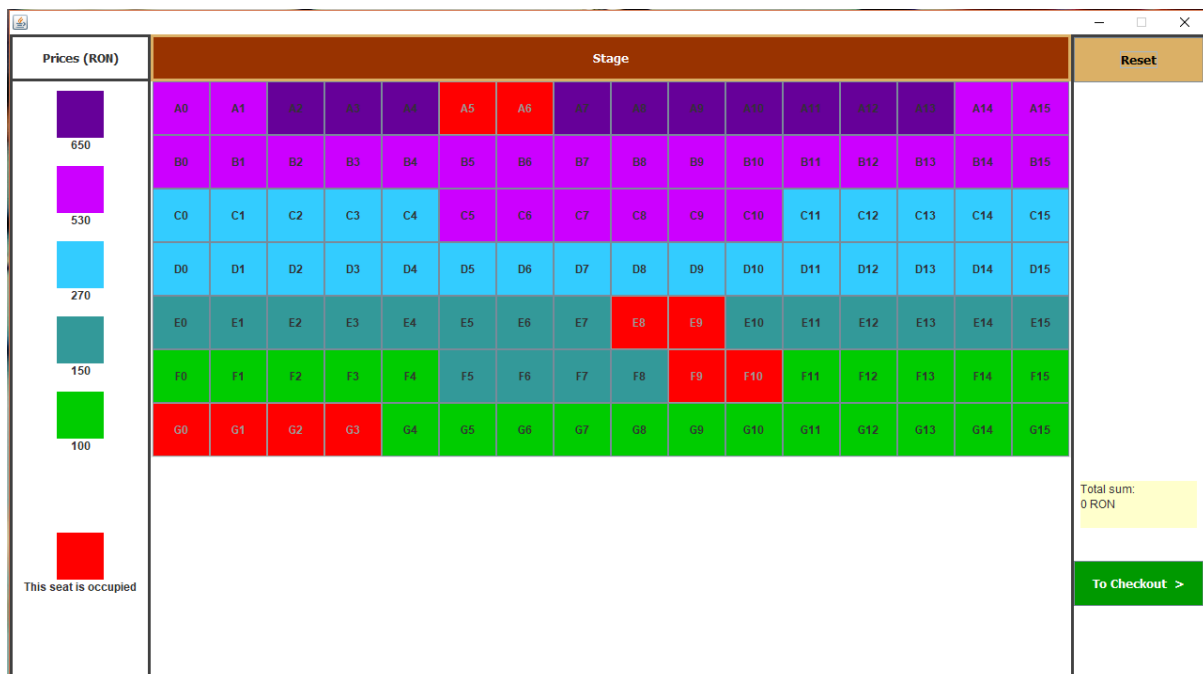
CTI English, 2nd year

I. Introduction

The purpose of this project is to implement an application designed for entertainment facilities like theatres, cinemas or concert halls. It intends to deliver a simple and friendly user interface, friendly to anyone using a computer. It may not look like something new because there are a lot of similar systems in use at ticket-selling companies but some of the applications already implemented lack one of the three main features that should be implemented: a colour coding scheme for seats of different prices, a reset button if you are not happy with the selection and don't want to deselect every seat and a counter that offers you the sum of the currently selected seats so you can have an idea of how much you will end up paying (the last one is especially frustrating at most cinemas where when you try to make reservations they just show you the general prices/day/time-of-the-day/film-type).

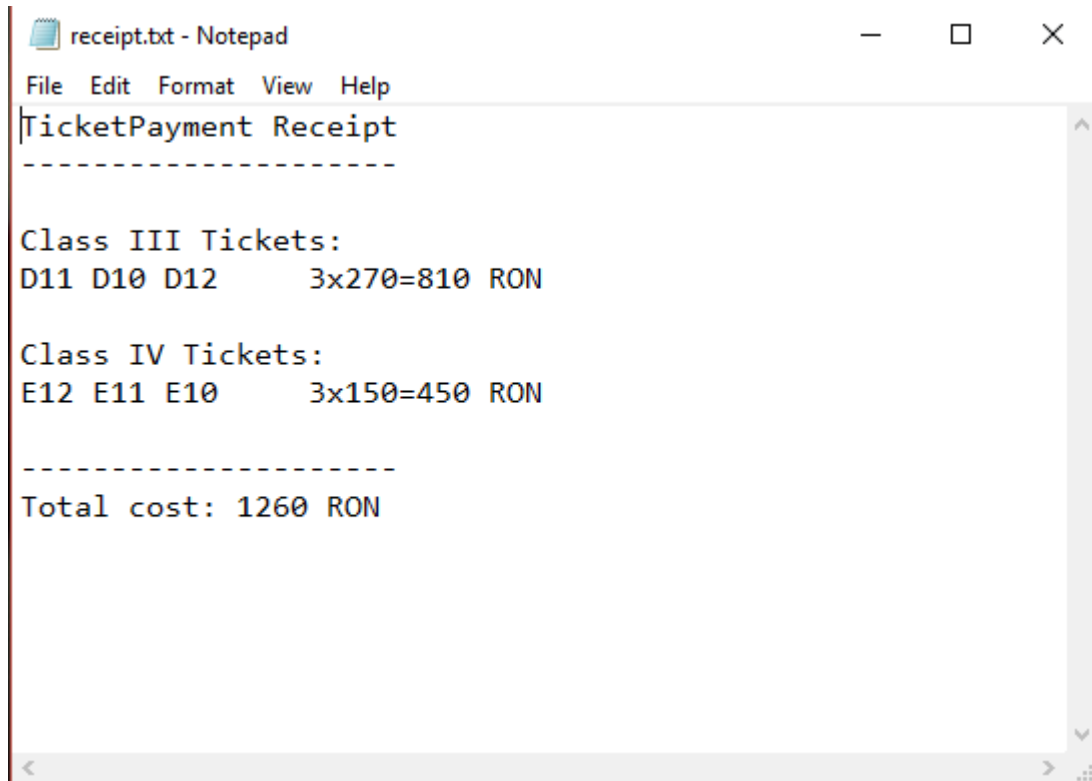
II. Use cases

The usage of the application is straight-forward as its interface:



You just check the available seats (seats that are not red – occupied) and depending on where do you want to sit and your budget you select, by clicking, the desired seats. If you are not happy with one or two seats, for example, you just deselect them by clicking on again. However, if you are dissatisfied with the current selection as a whole and change your mind,

you can easily deselect all of them by clicking the Reset button. In the end you should check the supposed sum with your budget and if everything is alright click the “To Checkout” button to get the receipt.



```
receipt.txt - Notepad
File Edit Format View Help
TicketPayment Receipt
-----

Class III Tickets:
D11 D10 D12      3x270=810 RON

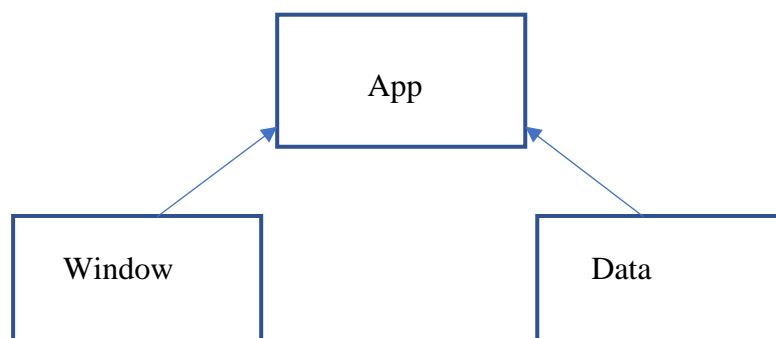
Class IV Tickets:
E12 E11 E10      3x150=450 RON

-----
Total cost: 1260 RON
```

I don't think there are any limit-cases. If all of the seats are sold they should all be red and the user should be unable to select any of them. The integer used for the sum should be more than enough for most currencies as the 2,147,483,647 limit is a very large sum for selling show tickets even if the user decides to buy all the seats.

III. Classes

These are the 3 classes used in this project:



The App class is the one that brings together the application as a whole. It consists of a main method that starts the DATA and GUI segments and an exit method to shut down the application.

The Data class deals with data handling. It inputs information from the seat-data file(data.txt) and the price-data file(prices.txt) making it possible for the GUI to know for example which seats are taken or what prices do certain seats have. It will also compute the data coming from the GUI in order to generate and output the receipt with the seats taken and they price/category and total sum.

The Window class is the GUI part of the application. It generates, based on the information received from Data, the whole interface with panels and buttons and labels. It is also responsible for the action of its elements such as button presses through ActionListeners.

IV. Implementation

The methods used are pretty basic: a read-from-file method or a write-to-file method, but there was a little challenge (a least for me) to generate a panel with a matrix of toggle buttons with identical property definitions and actions, in order not to create them manually and also to be able to create a matrix of any given size: 7 rows by 16 columns for example.

```
JPanel createGridPanel() {
    JPanel p_buttons = new JPanel(new GridLayout(ROWS, COLS));
    char c='A';
    for(int i=0;i<ROWS;i++)
    {
        for(int j=0;j<COLS;j++){
            JToggleButton gb = createGridButton(i, j);
            gb.setOpaque(true);
            gb.setText(c+Integer.toString(j));
            list.add(gb);
            p_buttons.add(gb);
        }
        c++;
    }
    return p_buttons;
}
```

Here each row will get a letter and each column of the row a number generating a classic formation of show tickets: A12, E9, etc.

```
private JToggleButton createGridButton(final int row, final int col) {
    final JToggleButton b = new JToggleButton();
    b.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            JToggleButton gb = getGridButton(row, col);
            if(gb.isSelected()==true)
                chosen.add(gb);
            else
                chosen.remove(gb);
            sum=0;
            for(int k=0; k<chosen.size();k++)
            {
                if(chosen.get(k).getBackground()==Window.Label1.getBackground())
                    sum+=Data.prices[0];
                else
                    if(chosen.get(k).getBackground()==Window.Label2.getBackground())
                        sum+=Data.prices[1];
                    else
                        if(chosen.get(k).getBackground()==Window.Label3.getBackground())
                            sum+=Data.prices[2];
                        else
                            if(chosen.get(k).getBackground()==Window.Label4.getBackground())
                                sum+=Data.prices[3];
                            else
                                if(chosen.get(k).getBackground()==Window.Label5.getBackground())
                                    sum+=Data.prices[4];
            }
            Window.textArea.setText("Total sum: \n"+Integer.toString(sum)+" RON");
        }
    });
    return b;
}
```

For the buttons, at the press of each one of them, I need to compute the current value of the chosen seats.

V. Further improvements

There are a few improvements to be made if this product should be on a market and used in everyday life:

1. Real database link for the seat and price data.
2. A draggable UI without a Windows Frame around it and a dedicated close button.
3. A more modern look by using more complex graphical elements but sticking to a minimalistic aspect so it won't be too clustered and confusing for any user.
4. A pay area with either direct payment via card or links to payment services like Paypal and this one too linked to a database after the transaction that will occupy the bought tickets so no other user may buy them again. In addition to that after the payment was accepted, the user should receive the receipt and the tickets on his/her email address that was filled in before.