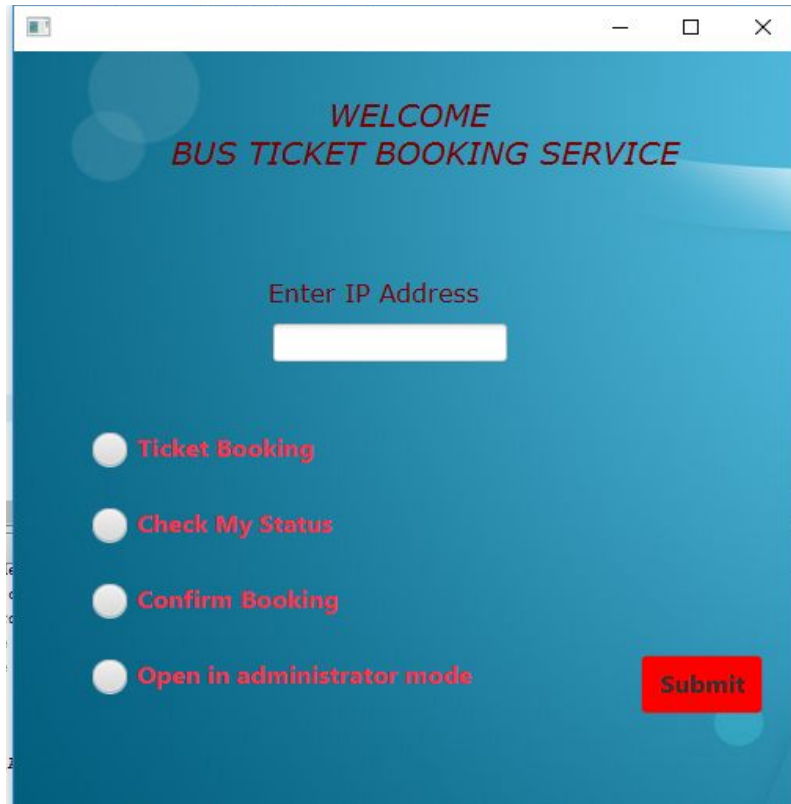


Our java project is an application for online ticket booking system. Basically we've tried to make an application for bus ticket booking for some specific company. Lets first introduce to different pages of our project and how to work with it, in other words here we're trying to make a short user manual for working with this application.

First Page:



WELCOME
BUS TICKET BOOKING SERVICE

Enter IP Address

☐ Ticket Booking

☐ Check My Status

☐ Confirm Booking

☐ Open in administrator mode

Submit

When an ordinary user will first open this application, they can make 3 choice

1. Ticket Booking: by selecting this option, he/she can book a ticket for him/herself.
2. Check My Status: If an user want to know, whether his booking is complete or pending or whether his booking has been cancelled, he has to use this option.
3. Confirm Booking: If an user has already paid for the ticket and wants to get his payment acknowledged, he can use this option. Here he'll need to submit the transaction ID and the password he is given.

If the user is the administrator of the system, then he can use the option "Open in administrator mode" .

And the user need to submit an IP address of the machine where the server is located at. Typically when the server and the client is kept at the same machine, the user should put "127.0.0.1" in this field.

Lets describe the options of the first page in detail. If we want to book a ticket, then this page will appear:

The screenshot shows a web application window titled "Choose Date, Time and Bus Type". The background is a photograph of a yellow bus. Overlaid on the bus image is a booking form with the following elements:

- Leaving from:** A dropdown menu with "Dhaka" selected.
- Going to:** A dropdown menu with "Sylhet" selected.
- Choose the date:** Three dropdown menus for date selection, showing "23", "December", and "2016".
- Time:** Three radio buttons for departure times: "8 am", "2 pm", and "12.30 am".
- Coach Type:** Two radio buttons for coach type: "AC" and "NON-AC". The "NON-AC" option is selected.
- Submit:** A yellow button with the text "Submit".

Here we can select from where we want to start our journey , our destination, date of journey, starting time , and the type of coach we're selecting. Here we've added all files needed to book a ticket for 23-12-2016 to 07-01-2017. The "server" file should be at running condition this time. Before submitting, make sure the server is running.

After pressing, the "submit" button, there should be a page like this,

<input type="checkbox"/> A-1	<input type="checkbox"/> A-2	<input type="checkbox"/> A-3	<input type="checkbox"/> A-4
<input type="checkbox"/> B-1	<input type="checkbox"/> B-2	<input type="checkbox"/> B-3	<input type="checkbox"/> B-4
<input type="checkbox"/> C-1	<input type="checkbox"/> C-2	<input checked="" type="checkbox"/> C-3	<input checked="" type="checkbox"/> C-4
<input checked="" type="checkbox"/> D-1	<input checked="" type="checkbox"/> D-2	<input type="checkbox"/> D-3	<input type="checkbox"/> D-4
<input type="checkbox"/> E-1	<input type="checkbox"/> E-2	<input type="checkbox"/> E-3	<input type="checkbox"/> E-4
<input type="checkbox"/> F-1	<input type="checkbox"/> F-2	<input type="checkbox"/> F-3	<input type="checkbox"/> F-4
<input type="checkbox"/> G-1	<input type="checkbox"/> G-2	<input type="checkbox"/> G-3	<input type="checkbox"/> G-4
<input type="checkbox"/> H-1	<input type="checkbox"/> H-2	<input type="checkbox"/> H-3	<input type="checkbox"/> H-4
<input type="checkbox"/> I-1	<input type="checkbox"/> I-2	<input type="checkbox"/> I-3	<input type="checkbox"/> I-4
<input type="checkbox"/> J-1	<input type="checkbox"/> J-2	<input type="checkbox"/> J-3	<input type="checkbox"/> J-4

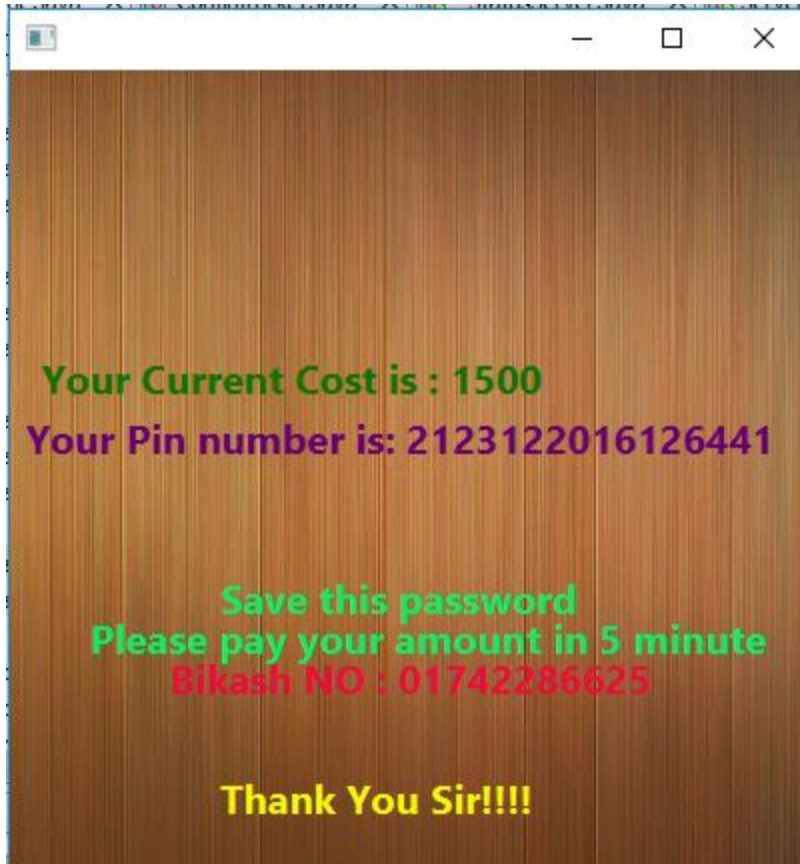
Your cost is: 1500

Previous **Continue**

**Red Marked
Are Booked**

Choose Your Seat

In this page, the red marked tickets are already booked , this tickets can't be booked. You can select the rest of the tickets. After pressing, "continue" there should be page like this.



You have to save the password and pay within 5 minutes, otherwise your booking might get cancelled. Then if you check your status, then you'll find it is "pending" .

WELCOME
BUS TICKET BOOKING SERVICE

Enter IP Address

127.0.0.1

☐ Ticket Booking

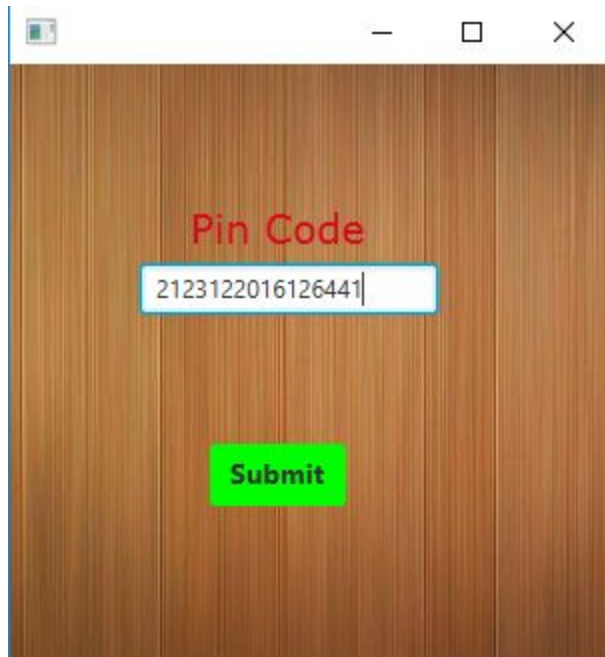
☒ Check My Status

☐ Confirm Booking

☐ Open in administrator mode

Submit

In this page you can select the “Check My Status” option and submit. Then you have to submit the pin number given to you during booking. The file “statusServer” should be at running condition.

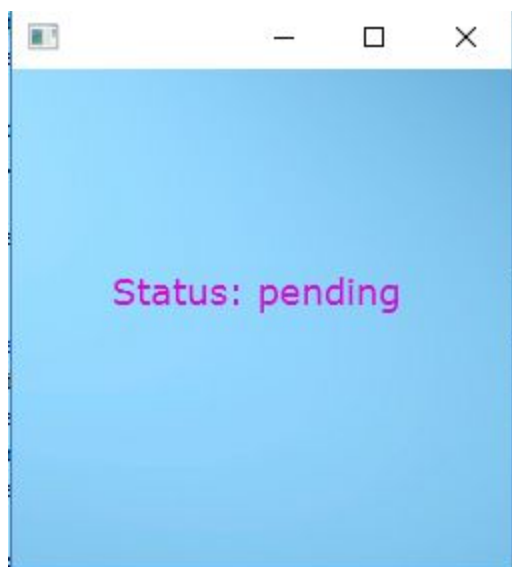


A screenshot of a web application window with a light blue title bar and standard Windows window controls (minimize, maximize, close). The main content area has a dark wood-grain background. At the top, the text "Pin Code" is displayed in red. Below it is a white text input field containing the number "2123122016126441" with a blue border and a cursor at the end. Further down is a green rectangular button with the word "Submit" in white text.

Pin Code

2123122016126441

Submit

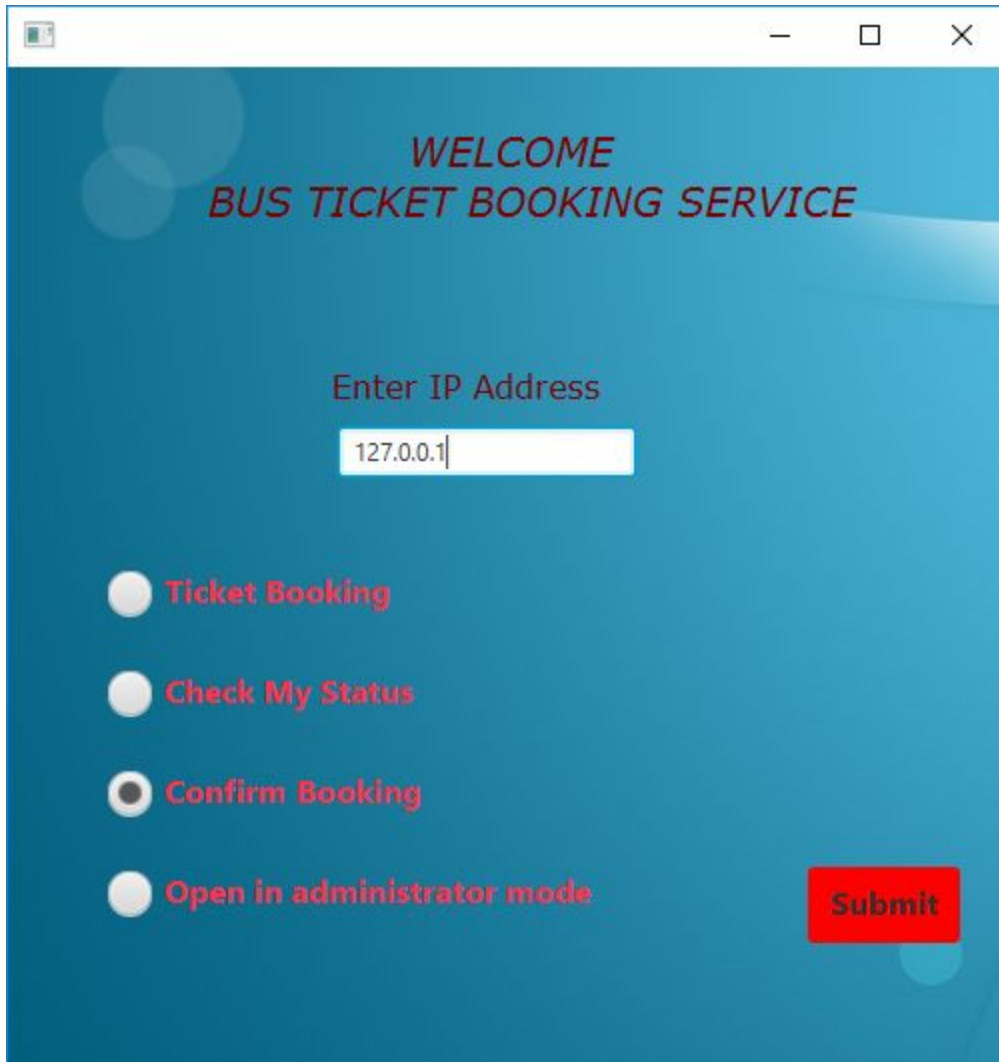


A screenshot of a web application window with a light blue title bar and standard Windows window controls. The main content area has a solid light blue background. In the center, the text "Status: pending" is displayed in a purple font.

Status: pending

When you'll pay , an administrator of the system have to put the information of your payment in the system . After the administrator have submitted these informations, you can submit your payment information .

You have to make sure the file "confirmationServer" should be at running condition.



**WELCOME
BUS TICKET BOOKING SERVICE**

Enter IP Address

127.0.0.1

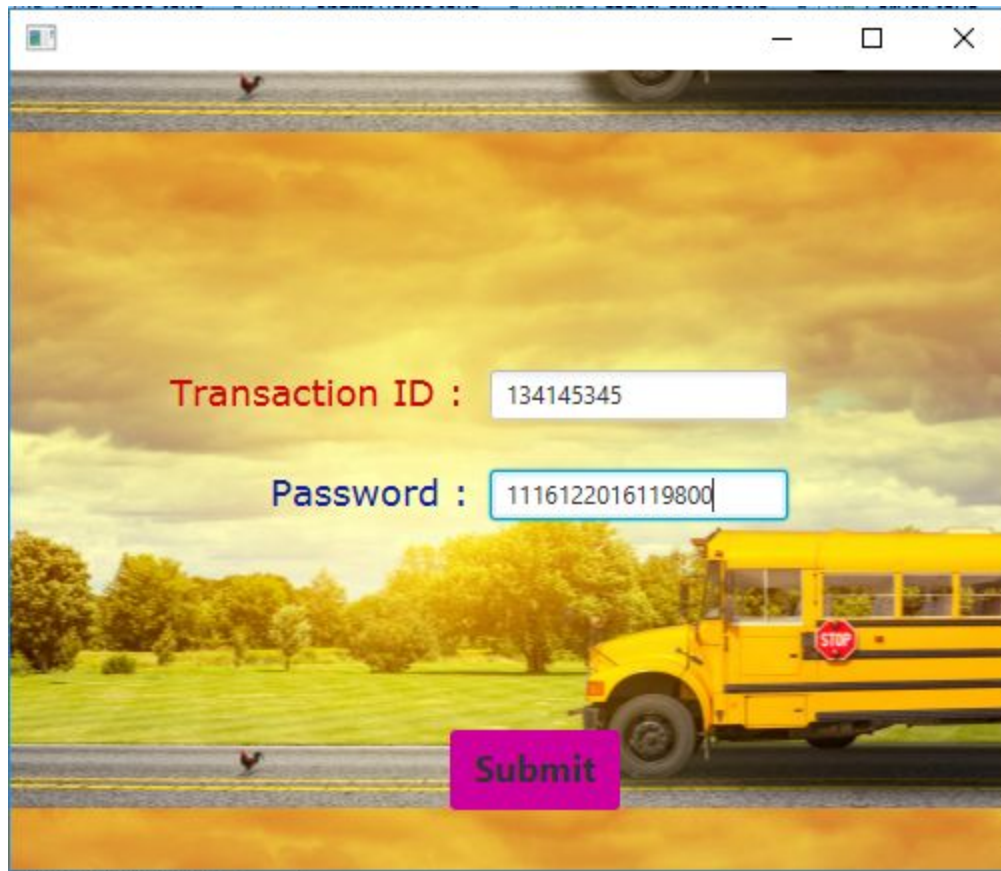
☐ Ticket Booking

☐ Check My Status

☒ Confirm Booking

☐ Open in administrator mode

Submit




Transaction ID :

Password :

If your submission is valid, then later you can check your status and you'll find it "booked" .

Lets now take a look at the administrator mood.

— □ ×

*WELCOME
BUS TICKET BOOKING SERVICE*

Enter IP Address

☐ Ticket Booking

☐ Check My Status

☐ Confirm Booking

☒ Open in administrator mode

Submit

— □ ×

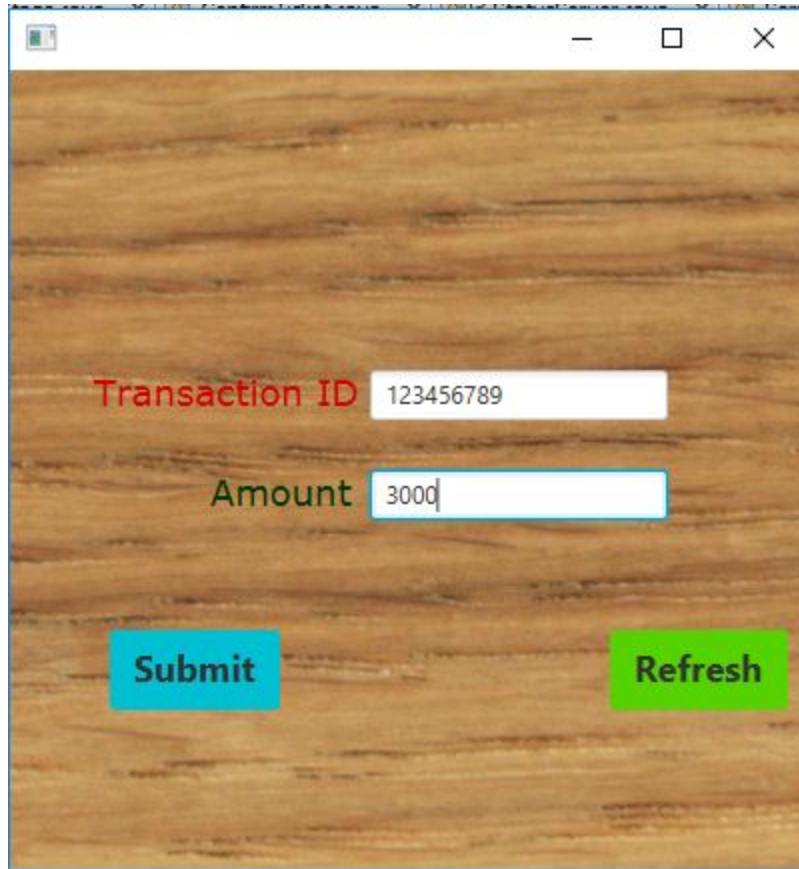
Name

A

Password

1|

Submit



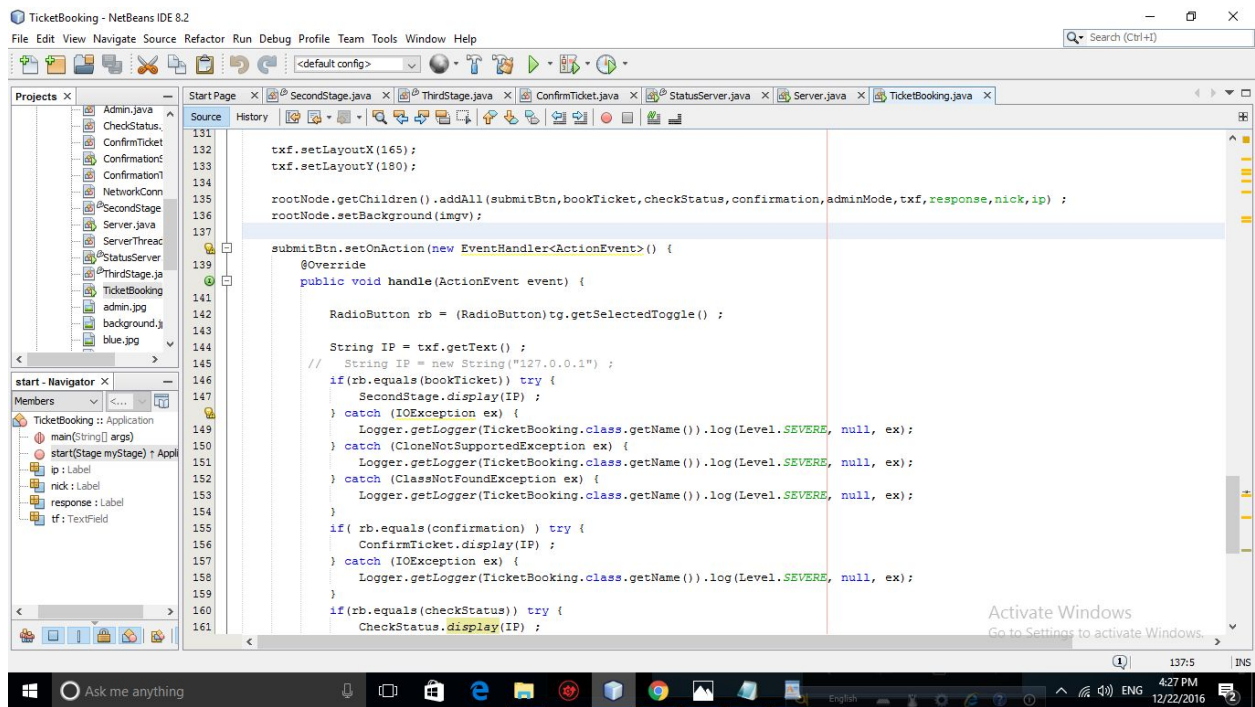
Transaction ID 123456789

Amount 3000

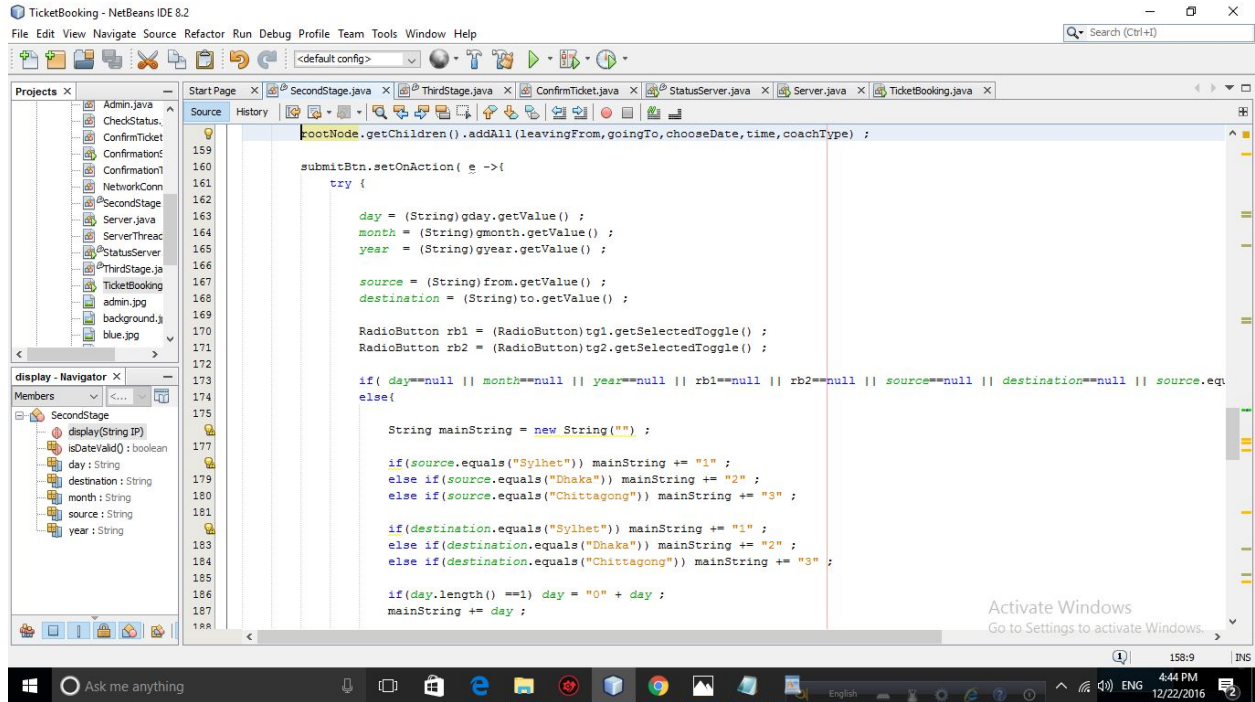
Submit Refresh

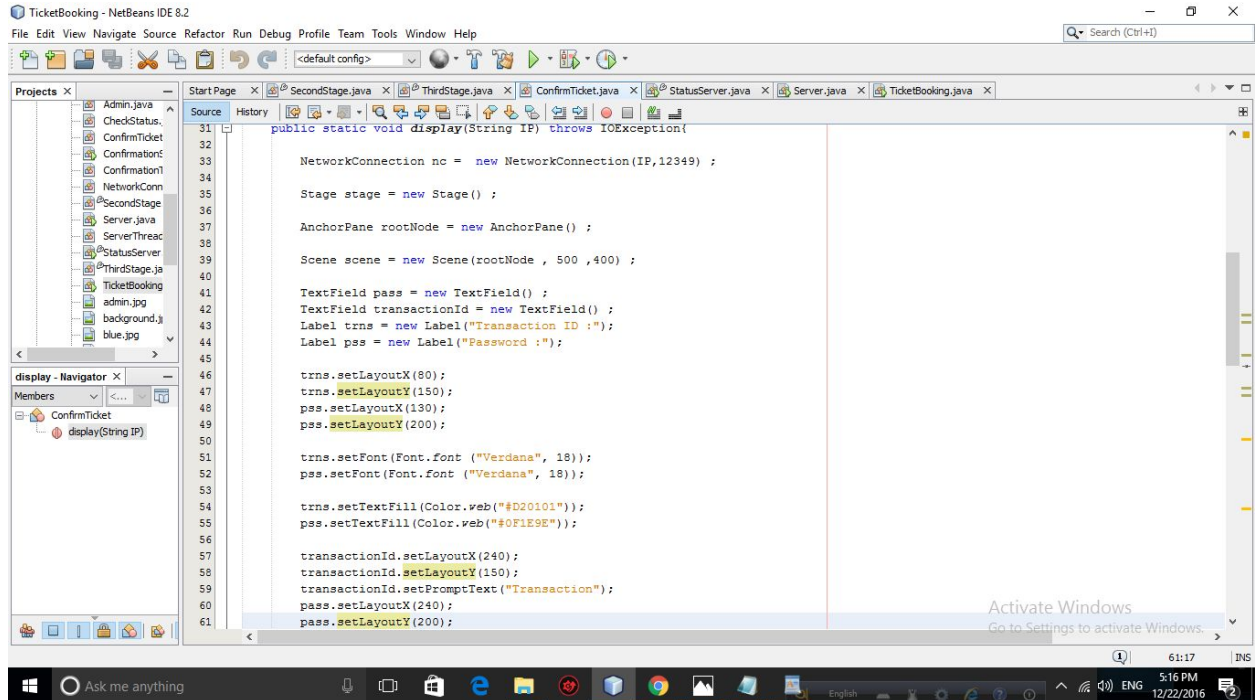
In this page, an administrator of the system can put information about new payment. If he presses “refresh” button, then the pending list will be refreshed. All the booking , which are pending for more than 5 minutes will be cancelled.

Now we're going to show you some important functions of our code , their functionality along with the workflow.



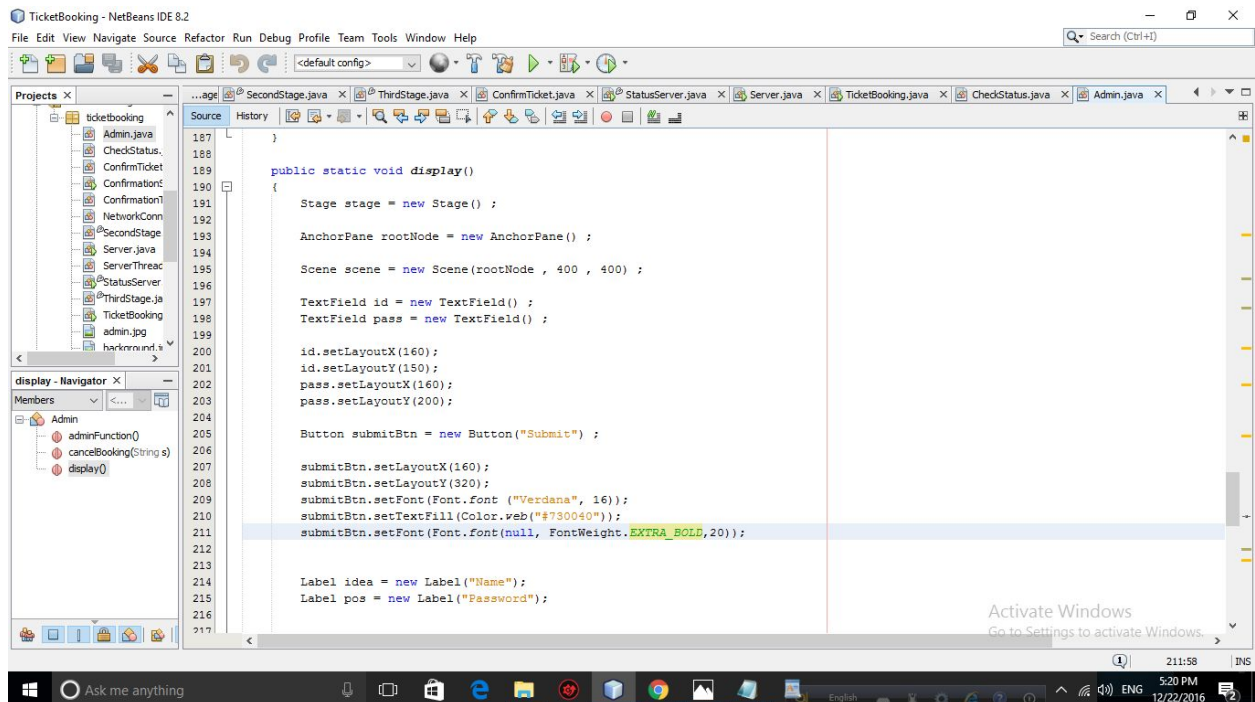
At first, when we'll run the program , it'll start from this first page. Then it'll call the appropriate function according to the client's choice. If the client want to book a ticket, then "secondStage.display()" will be called. And from this function a string will be formed consisting of client's choice, and then function "thirdStage.display()" will be called.



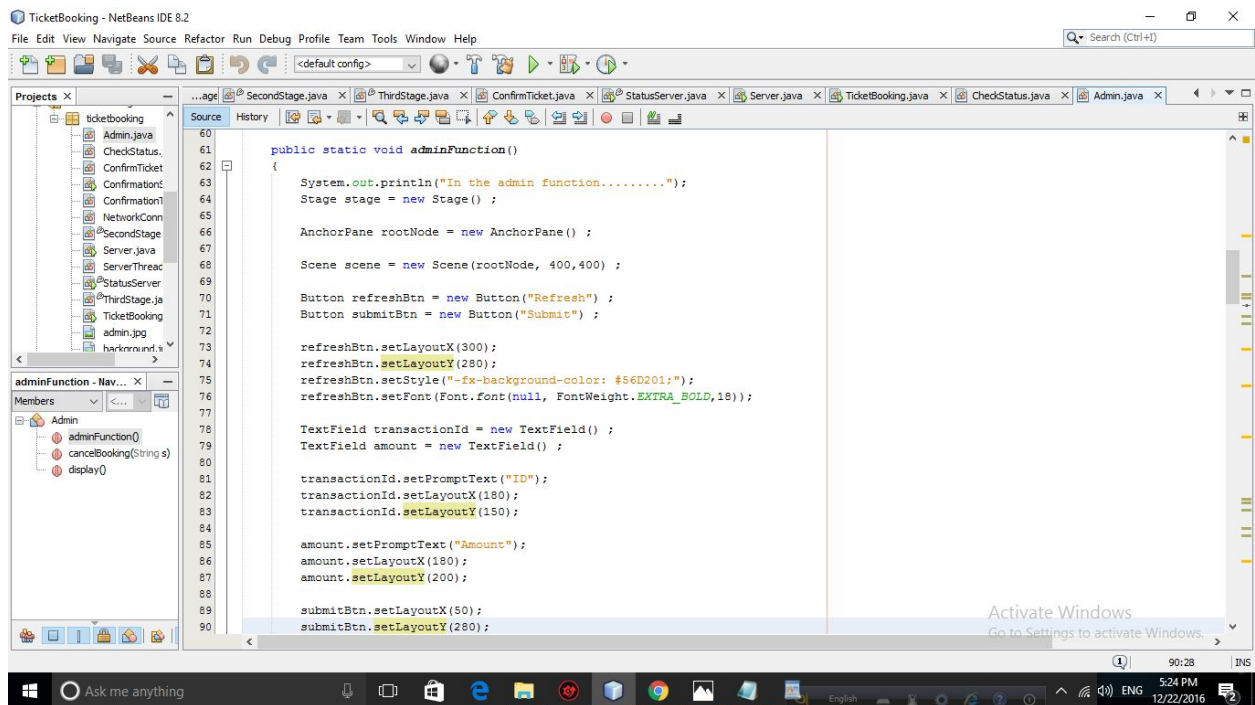


Then if, any client want to check his status whether it is pending/booked, he can check. And in this case the function CheckStatus.display() will be on action .

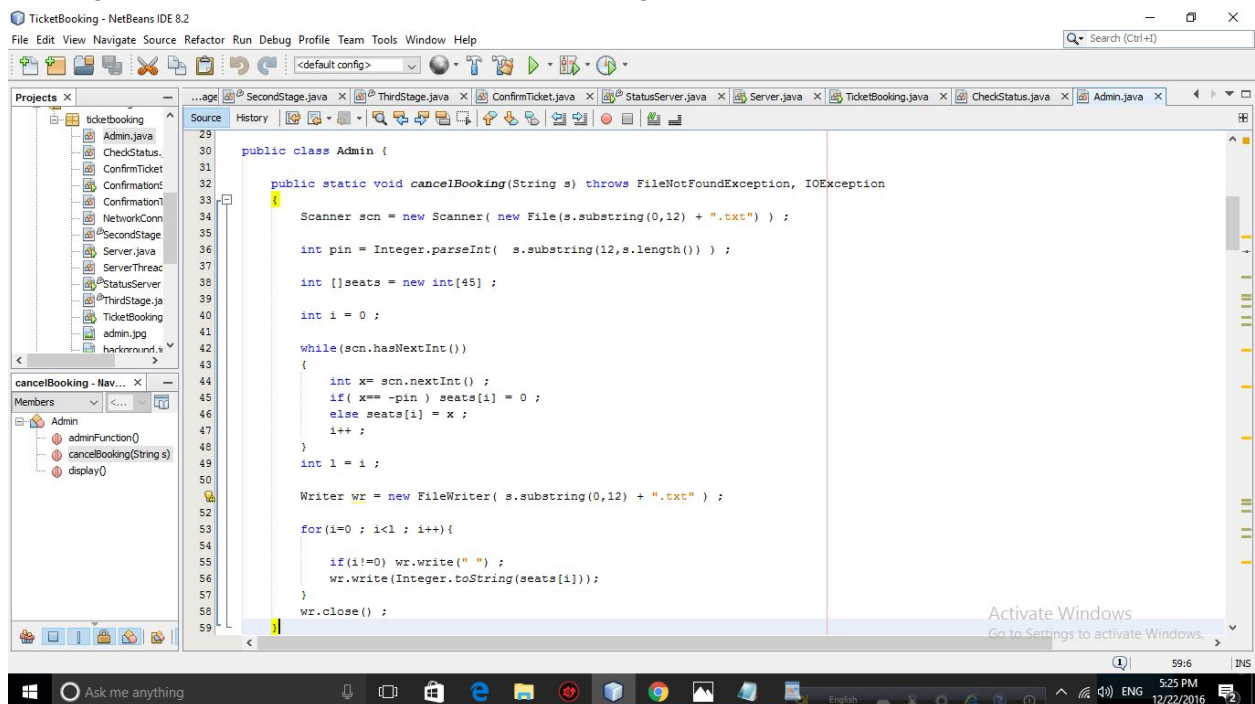
Next if we, talk about administrator, at first, the Admin.display() will be on action.



After the admin have submitted his username and password , he can either put any new payment information in the server or he can refresh the pending list. The function `adminFunction()` will then be called.

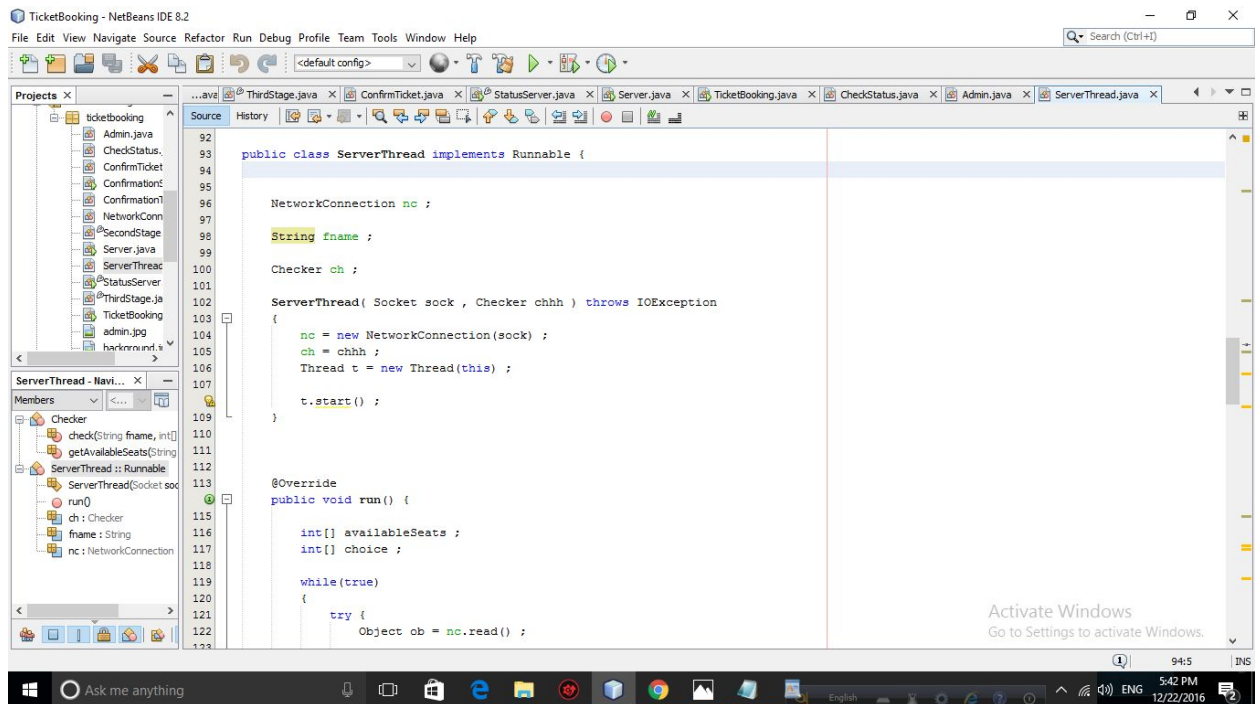


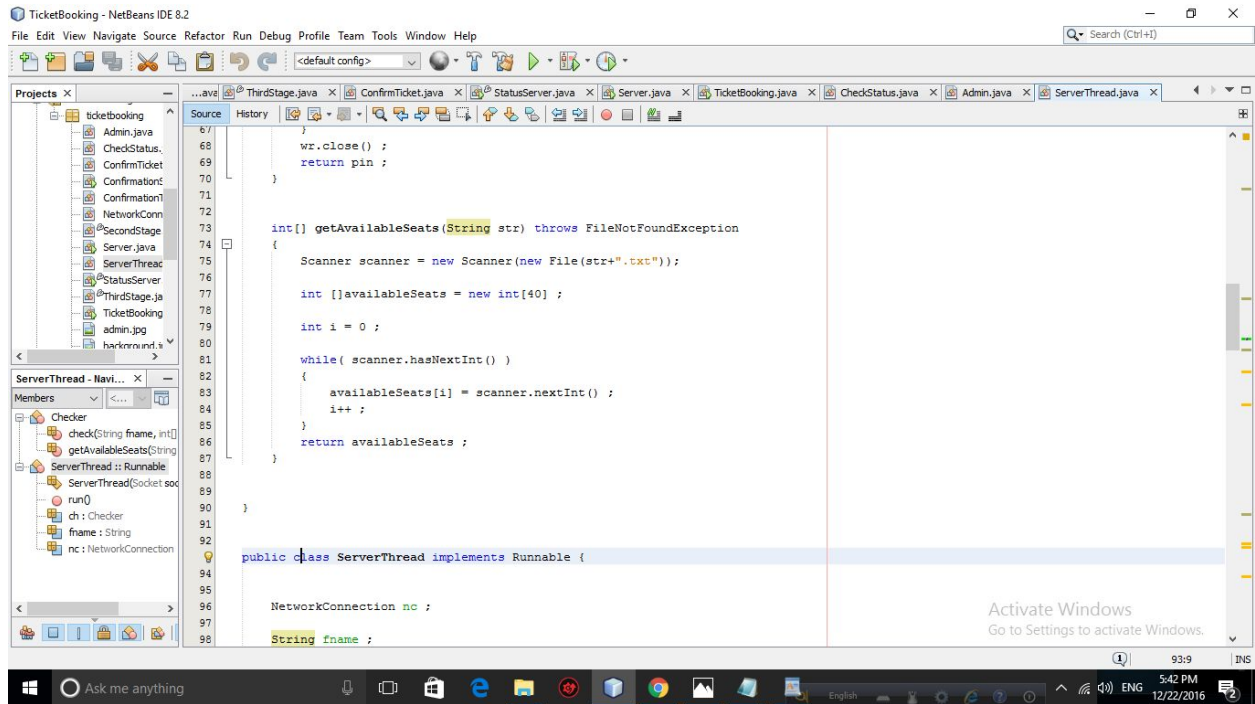
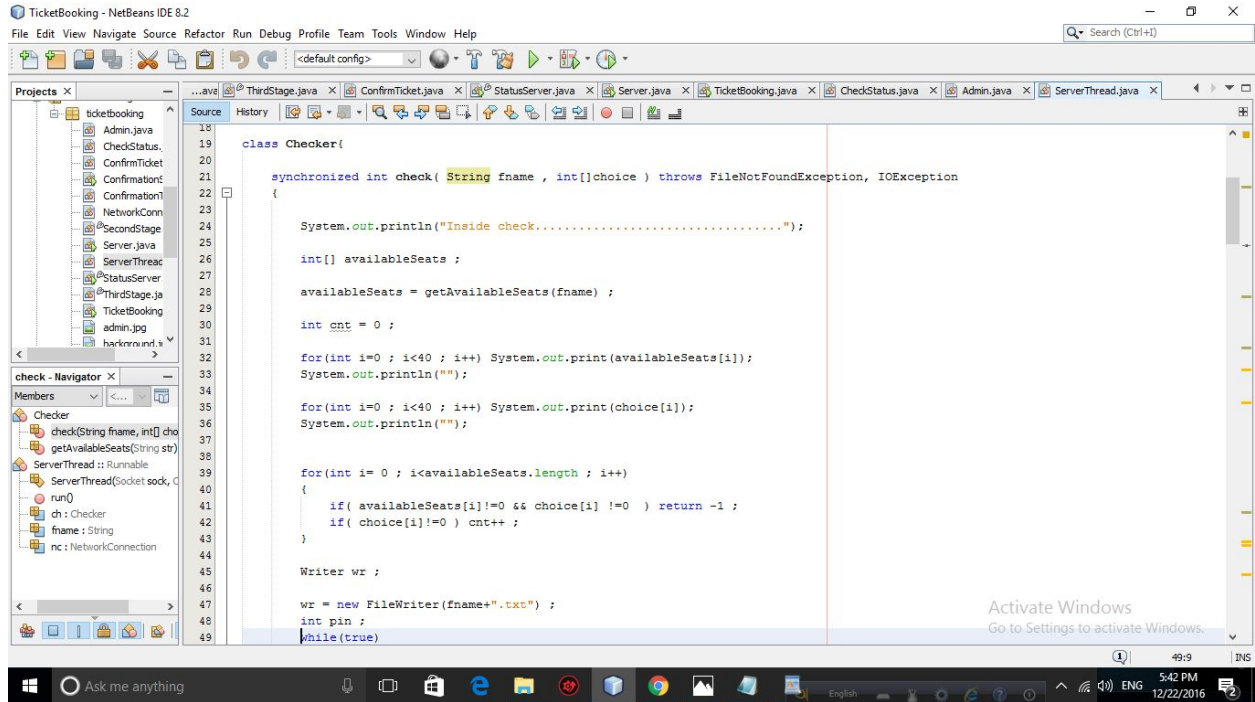
If admin refresh the pending list then, there might be need of cancelling some pending bookings. In this case, the function `cancelBooking()` will be called .



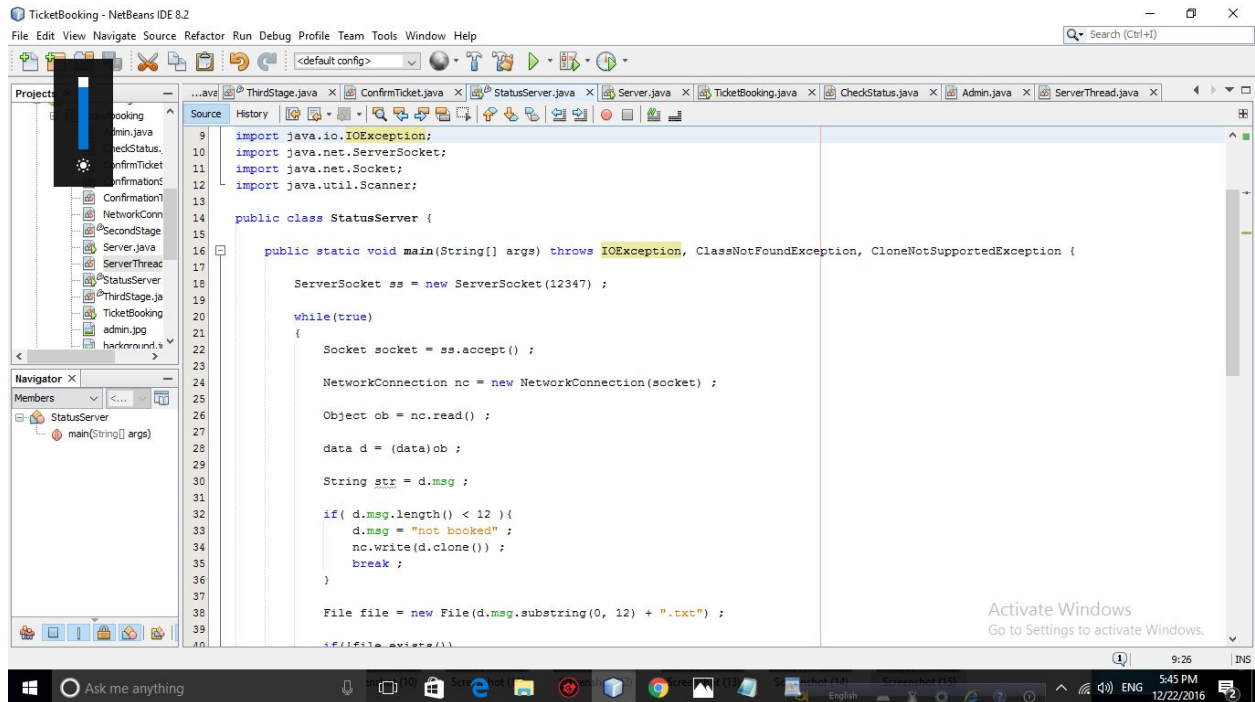
And there are 3 server thread which should be run separately. We strongly recommend to run this before using the app. These files are , “server” , “statusServer” , “confirmationServer” .

The server.main() function works when someone is booking a ticket, and it creates a new thread through a class serverThread. Again, here is a class Checker, and Checker.check() function checks whether clients choice of seat is valid/unbooked, if it is, then book the seats for the client. The Checker.getAvailableSeats() send client the available seats on clients request.

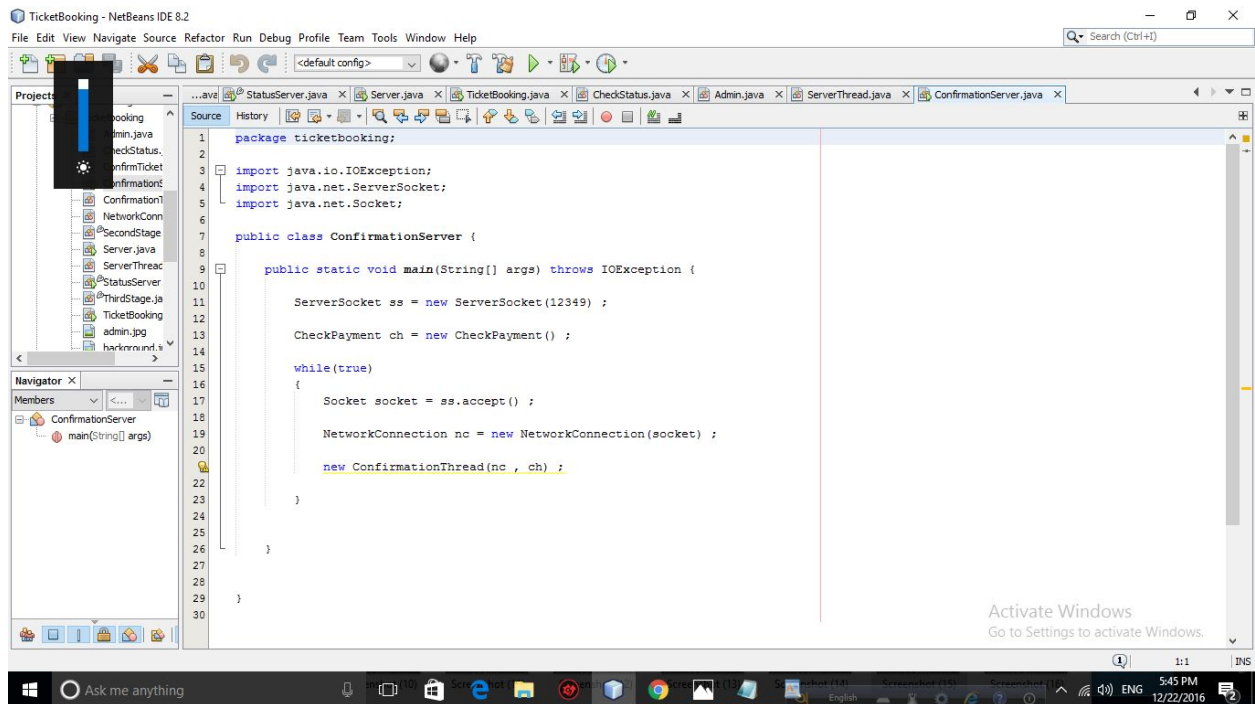




And the statusServer.main() works when a client request the server to check his/her status.



The confirmationServer works when, a client want to put his payment details and confirm his booking. It creates a thread through the class confirmationThread . And advances the process of confirmation.



What we couldn't complete:

1. In our program, an administrator needs to put the payment information in the server. So, it's not fully automated process, as we don't have knowledge about automatic verification of message/e-mail, we couldn't automate this process .
2. In the second stage, if a client chooses some wrong date (ex: chooses a date before the present day) and submit that date for booking a ticket, the system might be crashed, appropriate caution were not taken.
3. The pending list were not automatically refreshed, so if there is a booking pending for more than 5 minutes and the admin does not refresh the list, then a client might confirm his ticket although his confirmation should have been cancelled.

Tools, libraries & additional features:

We didn't use anything outside ordinary java library.