TECHNICAL UNIVERSITY OF CLUJ-NAPOCA ROMANIA

Faculty: Automation and Computer Science
Specialization: Computer Science in English
Subject: Object-Oriented Programming
Project: Airline Reservations System

Laboratory Teacher: Baka Aron

Students: Antonescu Cristina,
Deac Denisa

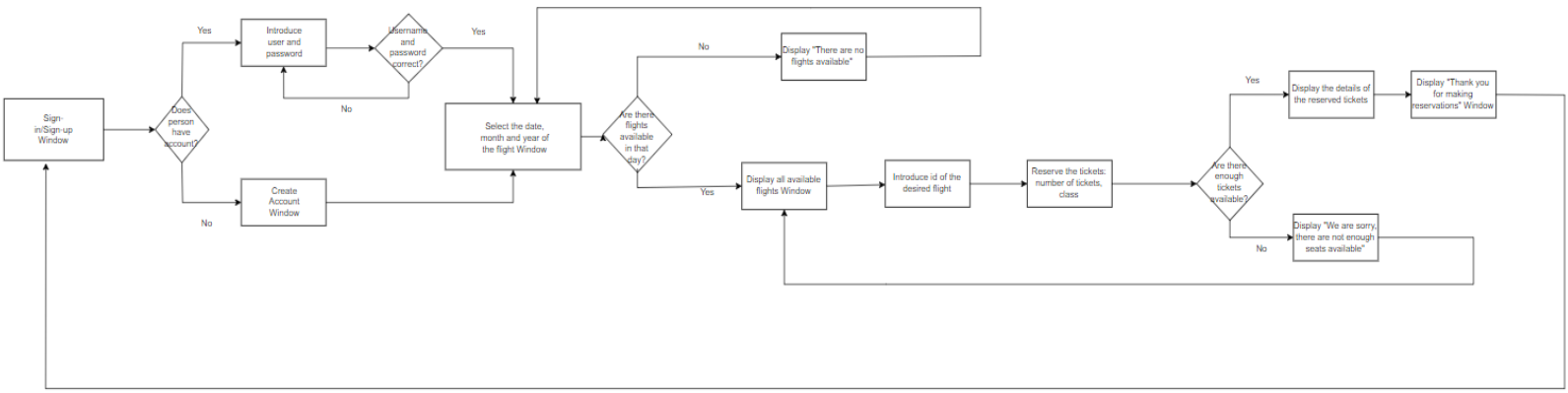
Table of contents:

1. Project Description

Our project consists of a basic airline ticket reservations system. The user, recognized by the system through the account they have created, can make reservations for their preferred date, month and year of the flight and the respective category (business or economy). The project has 8graphical interfaces, to make it more user-friendly. The data is stored in a database, consisting of 2 tables (1 for user, 1 for flights).

2. Use Cases

The diagram below shows the use cases of our application:

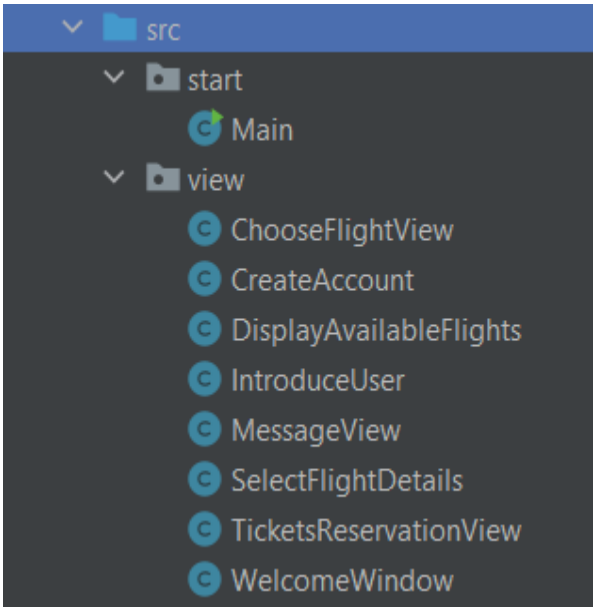


The user can make a reservation by first logging into the account they have already created or creating a new account. Then the user can select the date, month and year of the flight they want to make a reservation for. If there are available flights on this desired date, then the user can introduce the id of the flight they want to book and then the number of tickets and the class. If there are no available flights on the desired date, the user can select another date. Moreover, if there aren't enough tickets, the user can try to make another reservation. After the reservation has been made successfully, the user will see at the end the details of the reservation.

3. Solution

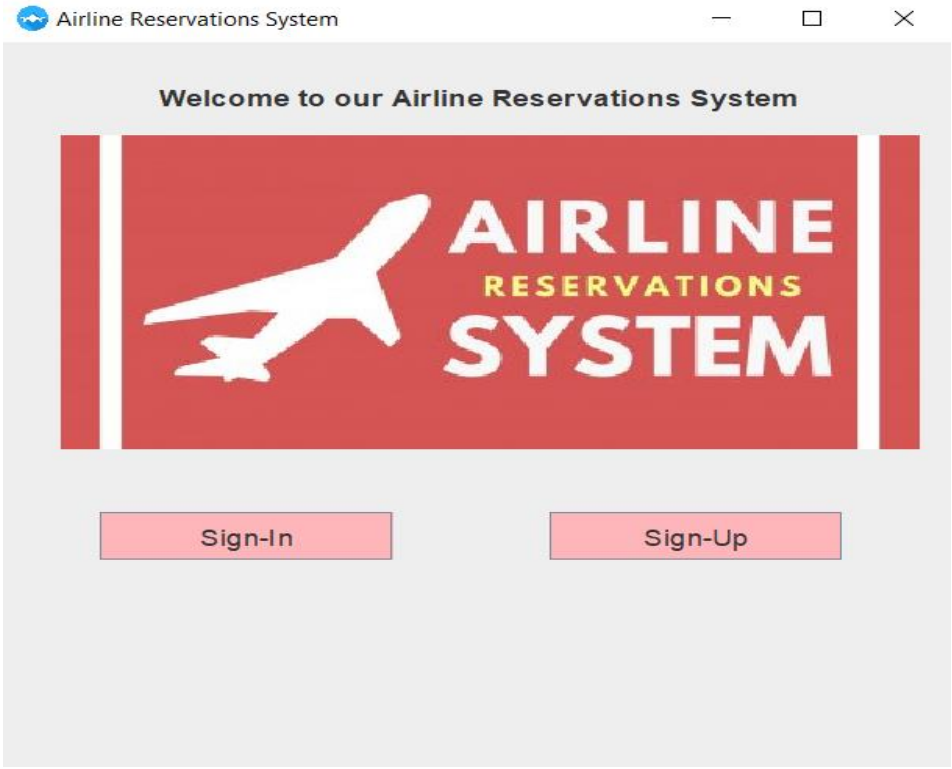
Our application has 10 graphical interfaces based on the use cases presented above, which are very easy to use and a database, in which we store relevant data and is updated when a reservation is made or when a new account is created. For the graphical interfaces, we used Java Swing, because we have more experience with it and for the database we used PostgreSQL, since we use it at the laboratory as well.

The project has the following classes:

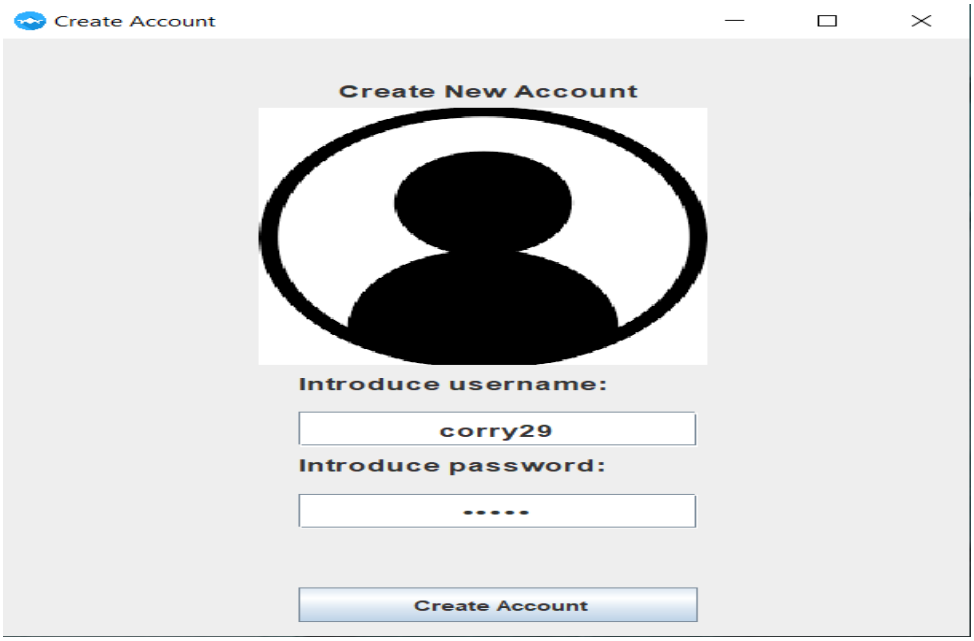


The start package contains the Main class, in which the WelcomeWindow is called. The view package contains all the interfaces. We connected the database to each of these classes and manipulated the data provided by the user in each class, because it was more straightforward for us.

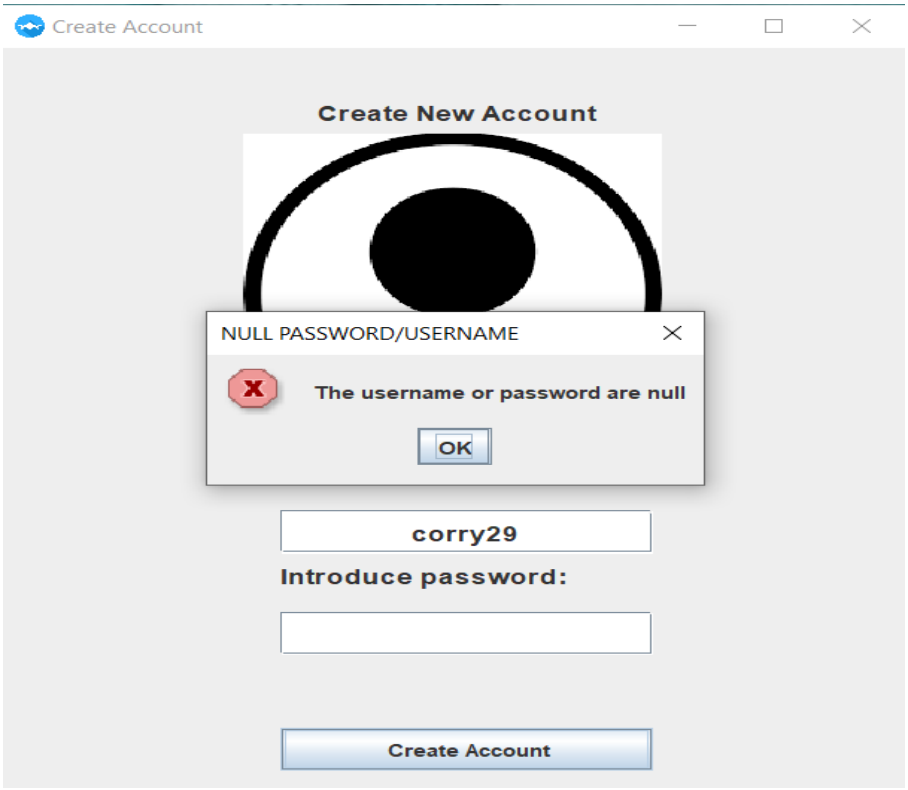
When the programme is run, the WelcomeWindow window appears.



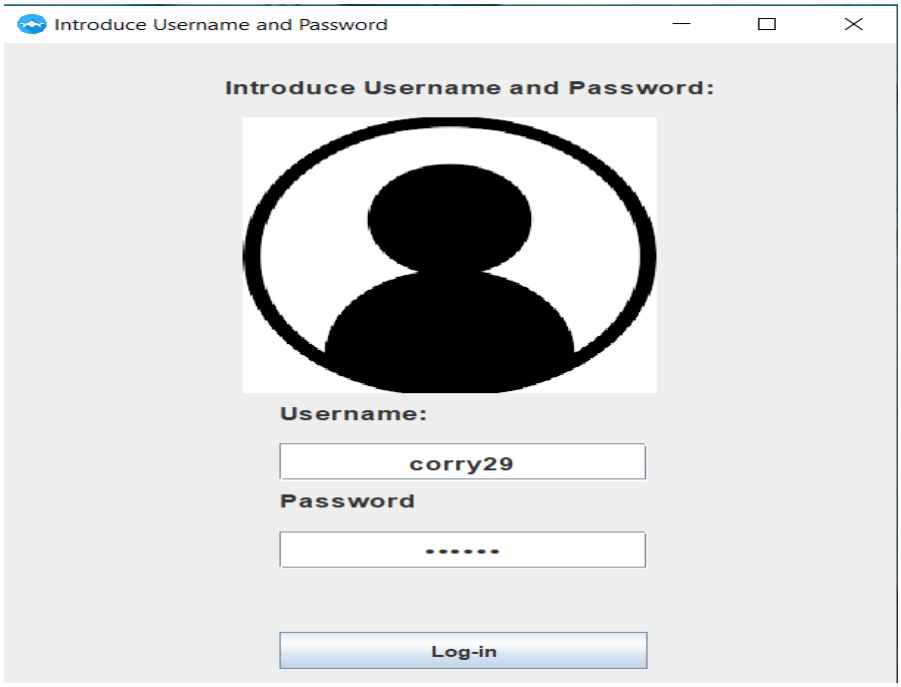
The user can either sign-in or sign-up. Depending on the choice, the CreateAccount window will appear or the IntroduceUser window.

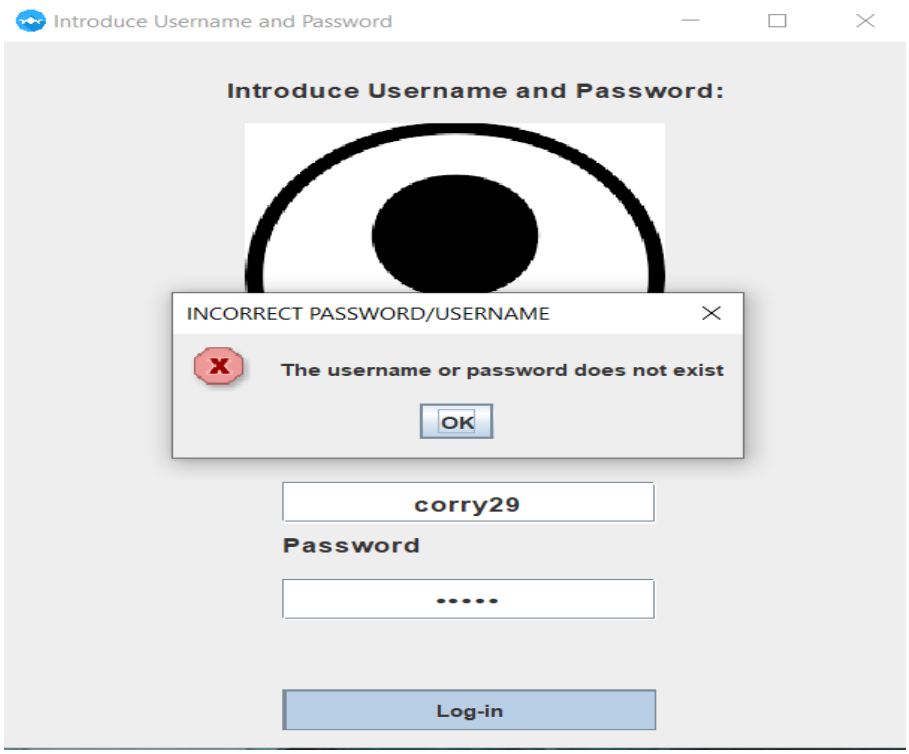


In the CreateAccount window, the user must introduce a username and a password, which will be introduced in the database and then will be asked to log-in. If the user presses the Create Account button and the username and/or password are null, an error message will appear.

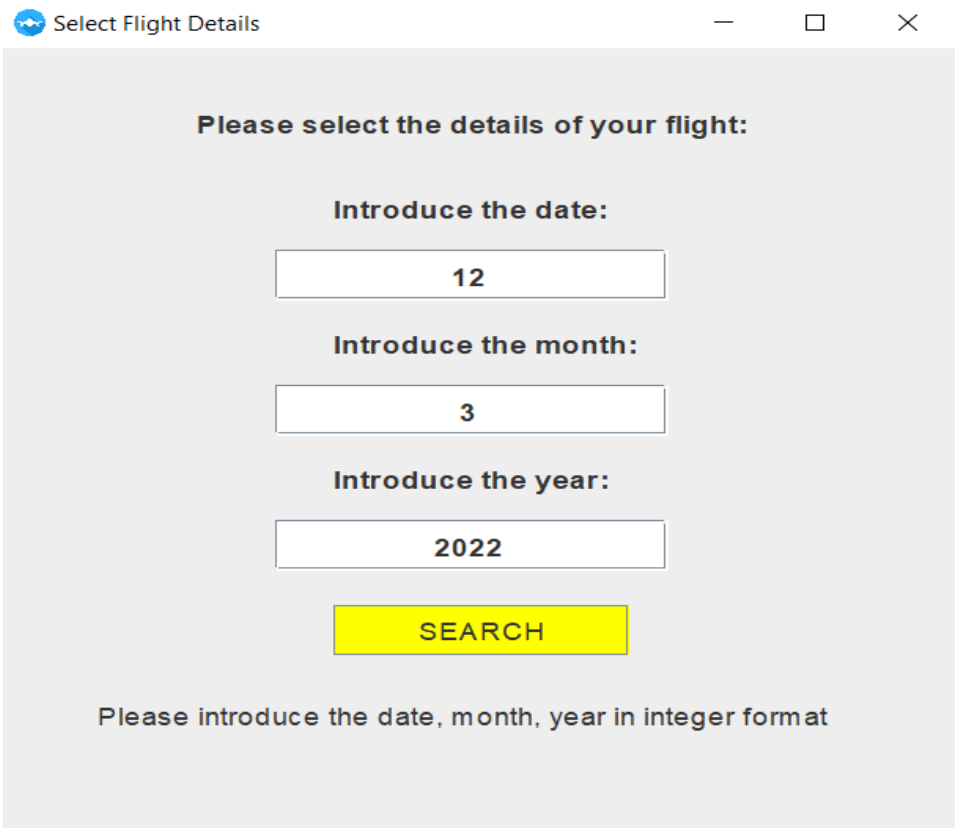


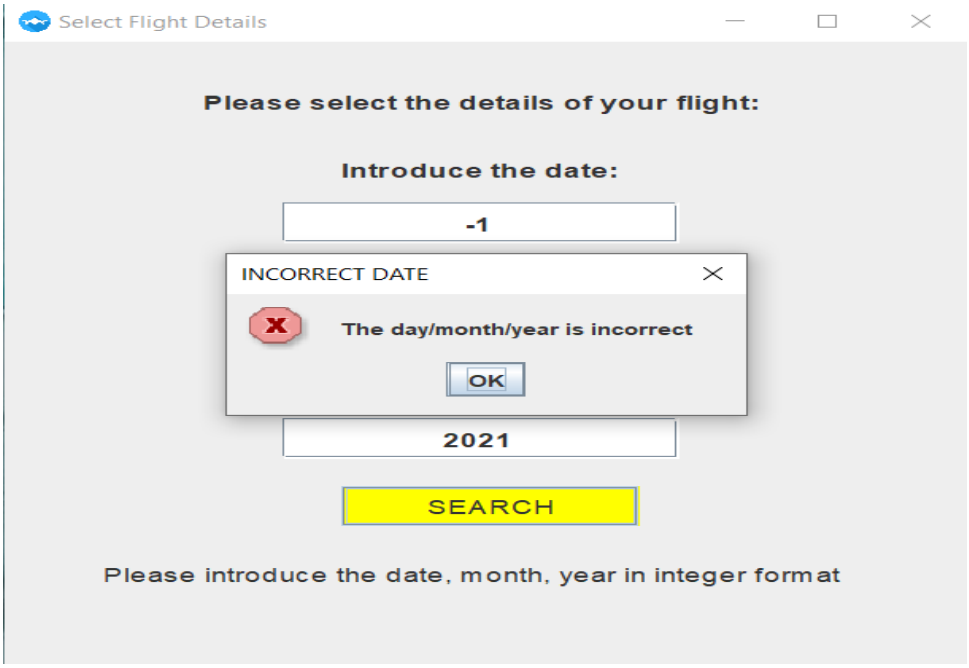
In the IntroduceUser window, the user must introduce the username and the password. If the combination is correct (it exists in the database), then the SelectFlightDetails window will appear after pressing the Log-In button, else, an error message will appear.


Introduce Username and Password
Introduce Username and Password:
Username:
corry29
Password
Log-in

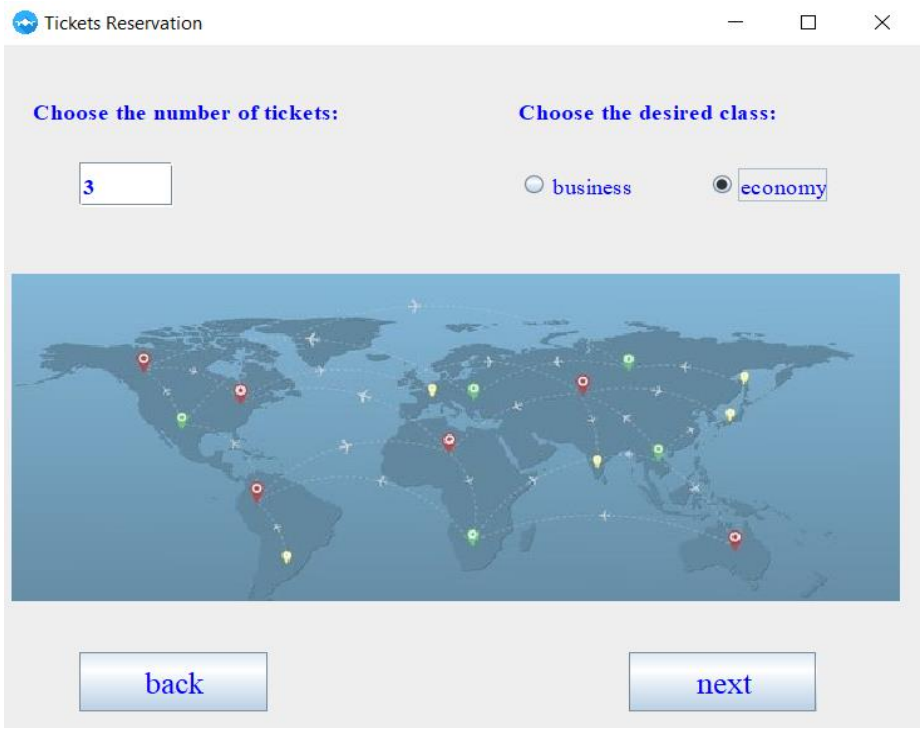

Introduce Username and Password
Introduce Username and Password:
INCORRECT PASSWORD/USERNAME
The username or password does not exist
OK
corry29
Password
Log-in

In the SelectFlightDetails window, the user must introduce the day, month and year that they want to make a booking for. The day, month and year must be introduced in integer format and if the date combination is not correct (for instance, 30th February or -11th August), an error message will be displayed. If there are any available flights on that date, then the DisplayAvailableFlights window will appear. If there are no available flights, then an error message will appear.

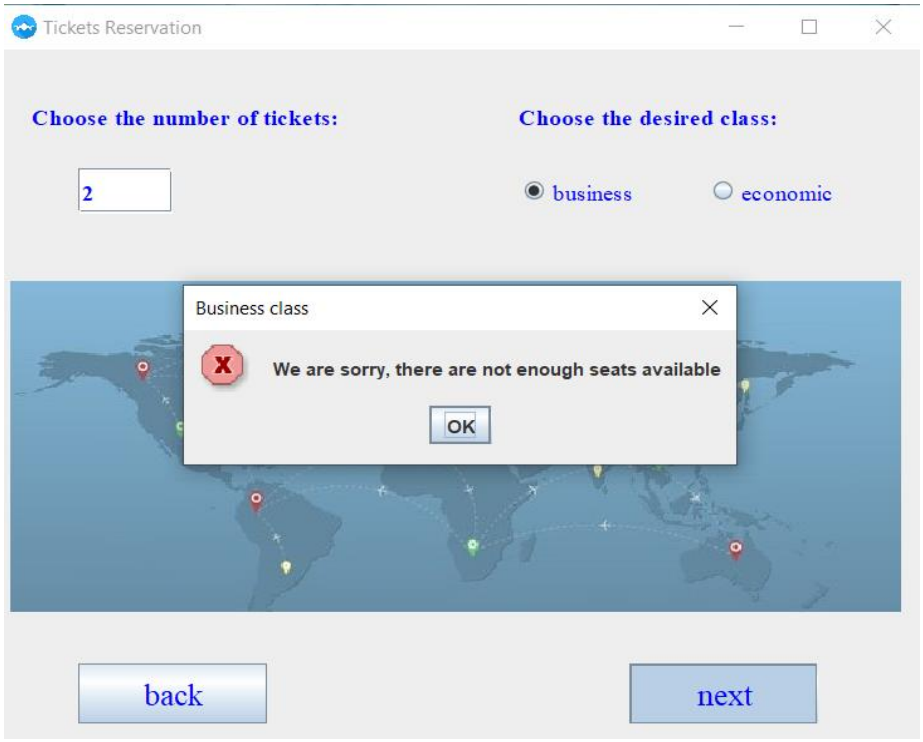


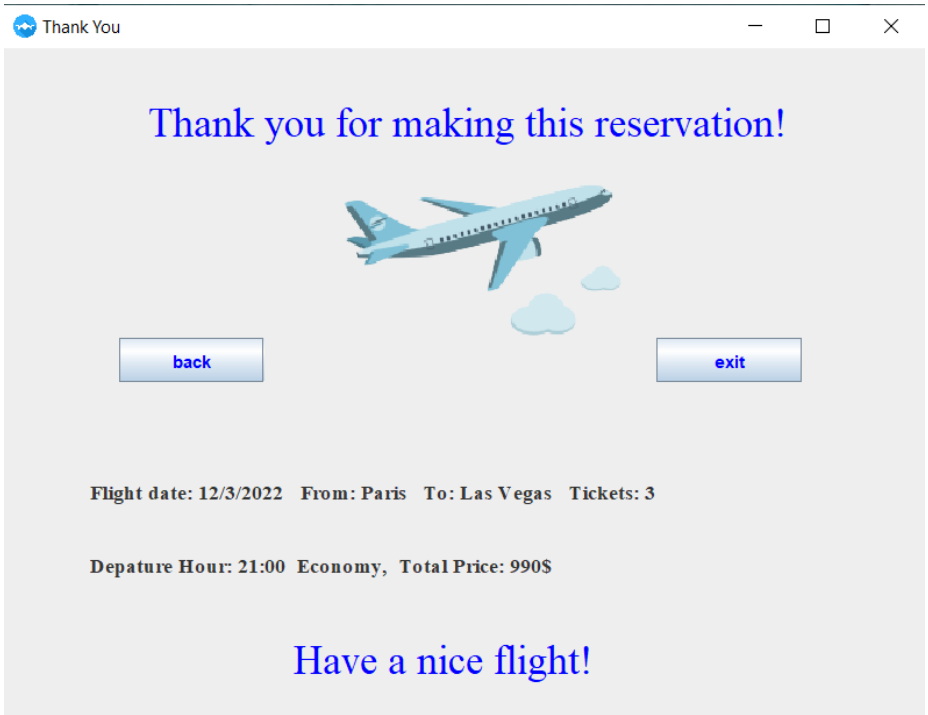


In the DisplayAvailableFlights window, all available flights stored in the database on the given date will be displayed in a table format, with labels as the header. By pressing the Make a Reservation button, the ChooseFlightView window will appear. In the ChooseFlightView, the user introduces the id of the desired flight. The ChooseFlightView window has 2 buttons, BACK and NEXT. By pressing BACK, the DisplayAvailableFlights window will appear again. By pressing NEXT, the TicketsReservationView window will appear.



In the TicketsReservationView, the user can introduce the number of tickets they want to book and the class (business or economy). If there are enough tickets, then the MessageView window will appear, which displays the details of the reservation and the database will be updated. If there aren't enough tickets, then an error message will appear.


Tickets Reservation
Choose the number of tickets:
2
Choose the desired class:
business
economic
Business class
We are sorry, there are not enough seats available
OK
back
next


Thank You
Thank you for making this reservation!
back
exit
Flight date: 12/3/2022 From: Paris To: Las Vegas Tickets: 3
Depature Hour: 21:00 Economy, Total Price: 990$
Have a nice flight!

The database we used for the project is a fairly simple one, which has two tables: accounts, which contains the users of the application and flights, which contains information about the flights stored in our system.

accounts [airline_reservation@localhost]

11 rows | Tx: Auto | DDL

WHERE | ORDER BY

	account_id	a_username	a_password
1	1	ana12	sunny
2	2	marius	123
3	3	carlaK	kkk
4	4	denisa	ddeni
5	5	anam	mmm
6	6	cris11	criss
7	7	bogdana	mmm
8	8	mikeK	koraline
9	9	luka@12	1234
10	10	mariusss	sss
11	11	corry29	corryc

flights [airline_reservation@localhost]

10 rows | Tx: Auto | DDL | CSV

WHERE | ORDER BY

	flight_id	flight_day	flight_month	flight_year	departure_city	destination_city	seats_economy
1	3	13	6	2022	Tokyo	Dubai	297
2	2	22	5	2022	Los Angeles	London	81
3	1	12	3	2022	Las Vegas	New York	50
4	6	22	5	2022	Dubai	Washington	90
5	7	22	5	2022	Rome	Madrid	40
6	8	13	3	2022	Sofia	Bucharest	100
7	9	12	3	2022	Berlin	Ottawa	300
8	10	12	3	2022	Ottawa	Berlin	300
9	4	12	3	2022	Las Vegas	Paris	490
10	5	12	3	2022	Paris	Las Vegas	91

seats_business	price_economy	price_business	flight_hour
179	100	400	06:30
183	300	1000	19:00
187	200	500	12:00
130	500	700	19:40
40	280	500	18:20
50	150	200	10:00
100	399	499	15:30
100	450	550	23:40
179	300	750	13:00
0	330	750	21:00

To create the database, we used Datagrip, since we use it at the laboratory as well and for the application we used IntelliJ, as it is very user-friendly and we have used it over the course of the semester.

4. Conclusion

Although our project was not very complex, it was good practice with Java Swing and the knowledge we have accumulated over the course of the semester and we also learned how to connect a database to an application. There are some aspects that could be improved at our airline reservations system, to make it even better for the user, but we are going to detail that at the next point.

5. Further Improvements

To make things easier for the airline company, a user account should contain the following information, apart from username and password: full name, address, telephone number, email. After a reservation has been made, this should be stored in a separate database and the user must select after making a reservation the method of payment. Another way to improve the functionality of the programme would be to add the admin feature, which allows users that are admins to add flights to the database and see the reservations. Also, the user can have the option to select the flight depending on their desired day or desired destination and departure city.