## **Introduction**

For developing the automated test for the website <https://www.ryanair.com/ie/en/> I used:

* **For testing**: Junit, Selenium, Cucumber BDD
* **Programming language**: Java
* **Design pattern**: Page Object Pattern
* **IDE**: Intellij Idea
* **Web Browser**: Chrome 79.0

I chose to create two type of tests: tests using **Selenium** and tests using **Cucumber** tests. They are in two different files and there is no connection between them.

## **How to run the tests**

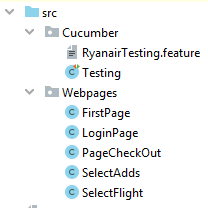
There are 2 options to run the tests:

1. Open the project in an IDE and run the Testing class (for Selenium tests) or Runner class (for Cucumber tests) ; please add all the jar files to Dependencies
2. Go to the project folder and open the cli. Run the command:

**java -jar "*full-path-to-the-jar-file*** \**untitled\out\artifacts\Test2Adults1Kid\_jar\Test2Adults1Kid.jar" "*full-path-to the project*\untitled\src\Cucumber\RyanairTesting.feature"**

## **Project structure and methods explanations**

I created a class for each page that is used in the booking process. For each page I created different methods that can be used for creating multiple type of tests.



***Cucumber*** folder contains three files:

1. ***Testing*** contains the Selenium tests; it can be run without adding Cucumber jars (I created it just to have two types of tests).

* **testOneAdult()**
* **testTwoAdultsOneKid()**

1. ***RyanairTesting*** contains the features for running Cucumber (the scenario)
2. ***RyanairRunner*** is the runner function for Cucumber

***Webpages*** folder contains the classes for every page of the website.

1. **LoginPage** class : login form

* **setLogin(email, password)** test the login form. The credentials for login are sent to the method through the parameters *email* and *password*

1. **FirstPage** class : the dashboard for searching the flight

* **setReturnTicket** : buy a return ticket
* **setOneWayTicket** : buy a single ticket
* **setFrom(fromCity)** : sets the departure city through the parameter fromCity
* **setTo(toCity)** : sets the arrival city through the parameter toCity
* **setDepartureDate(month, day)** : sets the departure date through the parameters month and day
* **setReturningDate(month, day)** : sets the arrival date through the parameters month and day
* **setNumberPassengers(adults, children, teens, infants):** sets the number of passengers depending on their type
* **search** : press the search button
* **searchFlight(fromCity, to, adults, teens, children, infants, monthF, dayF, monthR, dayR)** : the method sums up all the methods presented below to be able to call one single function for a flight search

1. **SelectFligtPage** class: flight selection, package selection and passengers names selection

* **setFlightFrom** : select the departure flight
* **setFlightTo** : select the return flight
* **setPackageFrom(bag)** : select the package for the departure flight through the *bag* parameter; it can be “value”, “regular” or “plus”
* **setPackageTo(bag)** : select the package for the return flight through the *bag* parameter; it can be “value”, “regular” or “plus”
* **setPackageValue** : it selects the “value” package for both flight
* **setPackageRegular** : it selects the “regular” package for both flight
* **setPackagePlus** : it selects the “plus” package for both flight
* **setPassengerTitle (title, id, type)** : sets the title for each passenger through the parameters given; *title* can be “Mr”, “Ms” or “Mrs”, *id* is the number that refers to the passenger of type *type* (ex. Id=0, Passenger 0 of type ‘A’ - adult)
* **setFirstName(firstName, id, type)** : sets the first name for each passenger through the parameters given
* **setSecondName(secondName, id, type)** : sets the second name for each passenger through the parameters given
* **next** : goes to the next page
* **setPassenger(title, firstName, secondName, id, type)** : the method sums up all the methods presented below to be able to call one single function for all the information needed

1. **SelectAddsPage** class: select seats, select bags, select other extra other things

* **setSeats** : if there are no children on board, the method will automatically select the random seats; if there are children on board, the method will select the suggested seats
* **setBags(x, noPassengers)**: it chooses between the “included” bag and “priority” bag for each passenger;
* **setNoAdds** : add no extra packages and finish the booking

1. **SelectAddsPage** class: select seats, select bags, select other extra other things

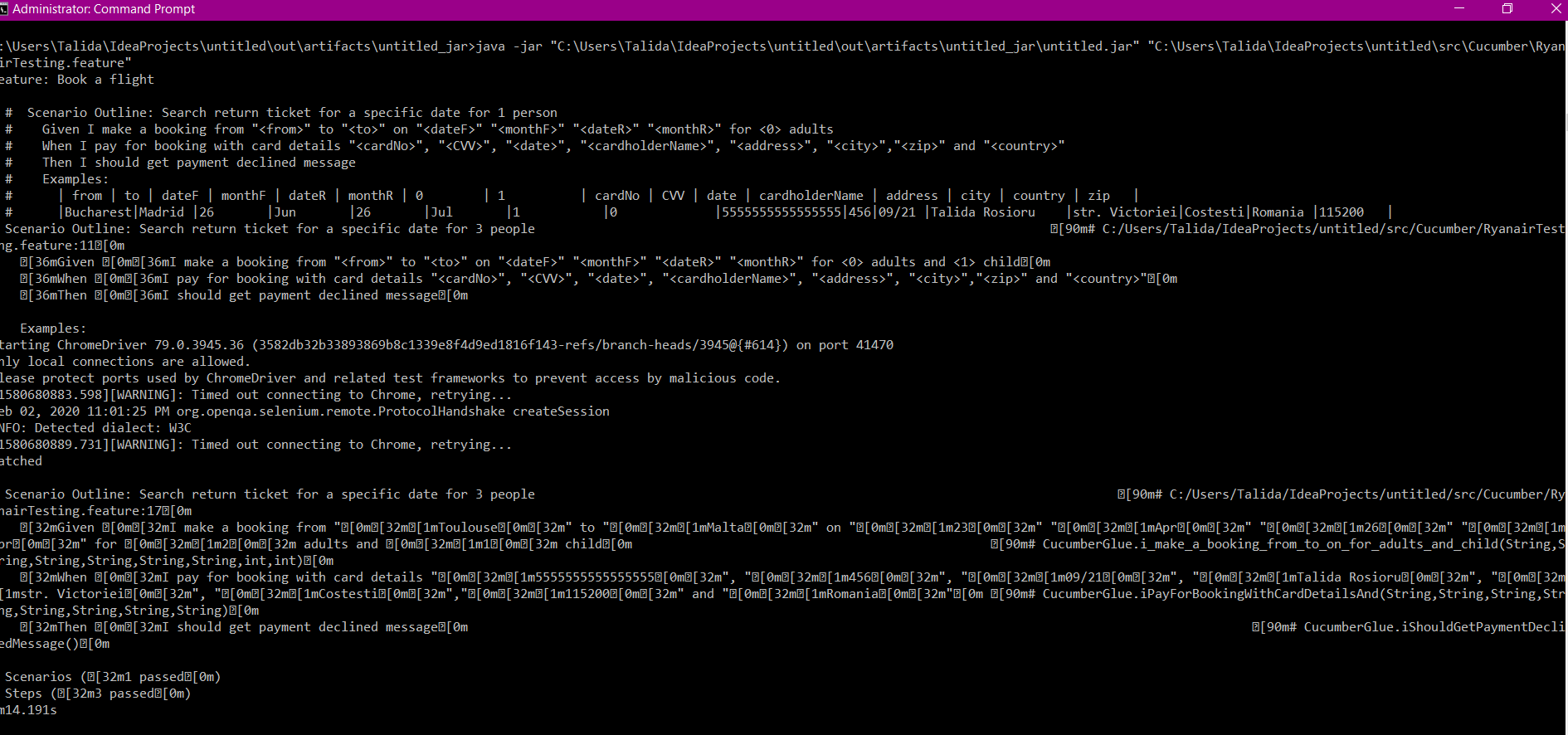
* **setCheckOut** : go to the basket and proceed check out
* **setPhone :** it sets the phone number
* **setInsurance** : it adds the insurance
* **setNoInsurance** : it doesn’t add any insurance
* **setCard(zipCode, cardNumber, expiryDate, CVV, address, city, cardholderName)** : sets the details of the cardthrough the parameters
* **setCurrency** : sets the currency to EUR
* **pay** : press the paying button

## **Test reports**

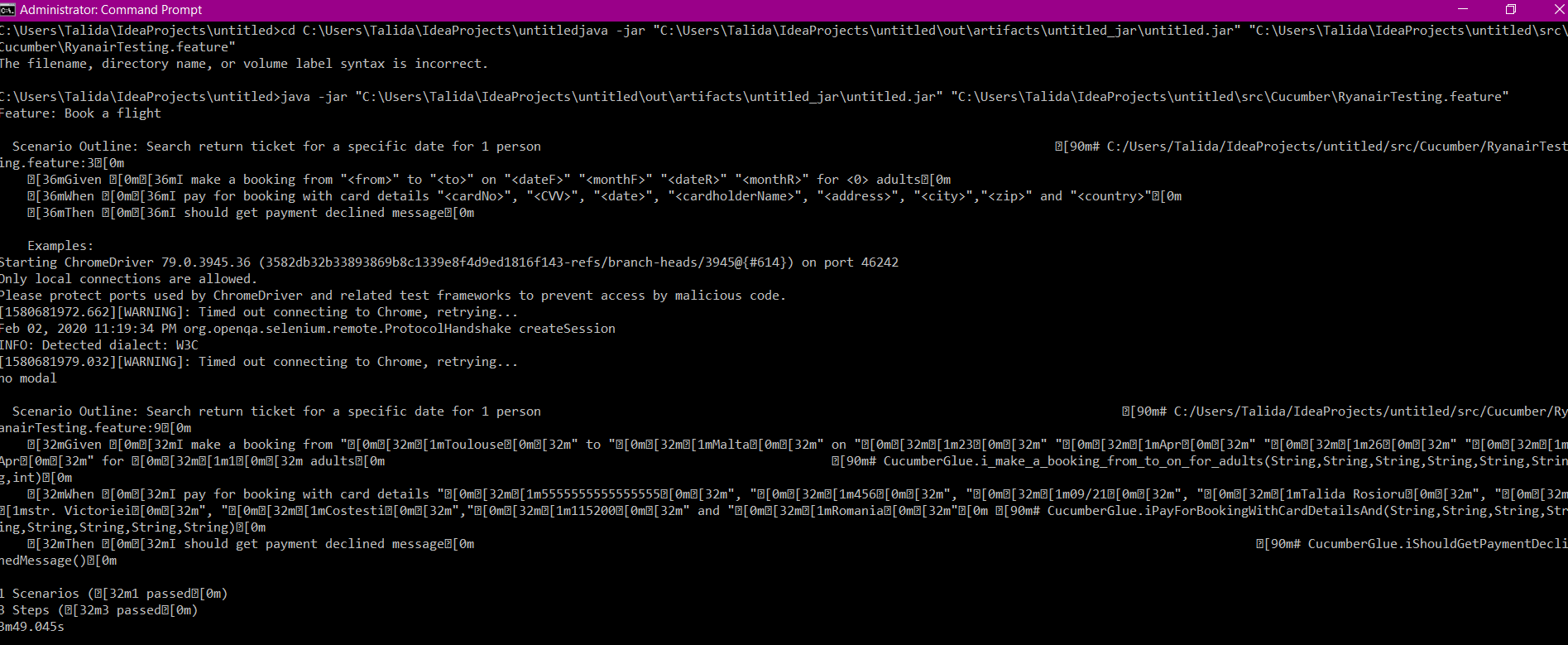
***Runner class (using Cucumber)***



* 2 adults 1 kid cli



* 1 adult cli



***Testing class (using Selenium)***



