

### Functional requirements

Name	FR#1	Randomly generate a list of flights
Summary	Generate a list of flights where each of the elements has randomly generated attributes and a unique flight number.	
Input	Number of flights to generate	
Output	A list of flights with randomly generated attributes	

Name	FR#2	Search for a flight by any of the criteria
Summary	You can search for flights by date, by hour, by flight number, by city, etc. If there is more than one flight that matches the criteria sought, then the first flight found will be displayed.	
Input	The search criterion	
Output	The first flight that matches the criteria. If no matches are found, then it returns null	

Name	FR#3	Sort flights by any of the criteria
Summary	You can sort the flights by date, by hour, by flight number, by city, etc.	
Input	The sort criterion	
Output	The list of flights is ordered according to the specified criterion	

Name	FR#4	Switch of page
Summary	Show the next page of flights or the previous page of flights depending on the decision that the user takes through some control. This is only possible if the requested page exists.	
Input	The decision of the user. It can be either next or previous.	
Output	The list of flights is ordered according to the specified criterion	

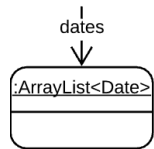
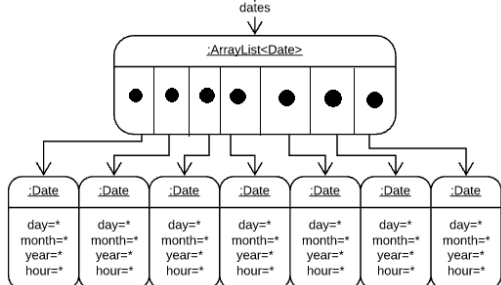
### Non-functional requirements

ID	Description
NFR#1	The time format that will be shown to the user is AM/PM
NFR#2	Create the graphic user interface using javafx

### Traceability

ID	Functional requirement	Method	Class
FR#1	Randomly generate a list of flights	generateFlightList getRandomCity getRandomAirline	Airport
FR#2	Search for a flight by any of the criteria	searchByDate searchByTime searchByAirline searchByFlightNumber searchByDestinationCity searchByBoardingGates	Airport
FR#3	Sort flights by any of the criteria	sortByDateAndTime sortByDate sortByTime sortByAirline sortByFlightNumber	Airport

		sortByDestinationCity sortByBoardingGates	
<b>FR#4</b>	Switch of page	nextPage previousPage setupPage	AirportController

Class	Method	Scenario
FlightTest	setupScenary1()	Empty
DateTest	setupScenary1()	
DateTest	setupScenary2()	
AirportTest	setupScenary1()	Empty

<b>Objective of the test:</b>		Verify that the objects are correctly generated and all the parameters passed to the constructors are correctly assigned		
Class	Method	Scenario	Input	Result
<b>Date</b>	Date	setupScenary1	<ul style="list-style-type: none"> <li>day = sr.nextInt(31)+1;</li> <li>month = sr.nextInt(12)+1;</li> <li>year = sr.nextInt(10)+2020;</li> <li>hour = sr.nextDouble()*24.0;</li> </ul>	A new Date has been successfully created. Each of the attributes of the new Date has correctly assigned the information passed by parameter.
<b>Flight</b>	Flight	setupScenary1	<ul style="list-style-type: none"> <li>date = new Date(1,1,1,1);</li> <li>airline= "HappyAir";</li> <li>flightNumber=124234;</li> <li>destinationCity="The moon"</li> <li>boardingGates=10;</li> </ul>	A new Flight has been successfully created. Each of the attributes of the new Flight has correctly assigned the information passed by parameter.
<b>Flight</b>	Flight	setupScenary1	<ul style="list-style-type: none"> <li>date = new Date(2,2,2,2);</li> <li>airline= "Fly for free!!!";</li> <li>flightNumber=-123;</li> <li>destinationCity="Popayan"</li> <li>boardingGates=-23112;</li> </ul>	The Flight has not been created as the flight number and the boarding gates were invalid.
<b>Airport</b>	Airport	setupScenary1	None	A new Flight has been successfully created.

Objective of the test:		Verify that the airport generates the requested number of flights and it's able to sort them and search them according to different criteria		
Class	Method	Scenario	Entries	Result
Airport	generateFlightList	setupScenary1	• None	The list of flights was created and has the requested number of flights
Airport	sortByDateAndTime	setupScenary1	• None	The list of flights was sorted according to the criterion
Airport	sortByAirline	setupScenary1	• None	The list of flights was sorted according to the criterion
Airport	sortByTime	setupScenary1	• None	The list of flights was sorted according to the criterion
Airport	sortByDestinationCity	setupScenary1	• None	The list of flights was sorted according to the criterion
Airport	sortByFlightNumber	setupScenary1	• None	The list of flights was sorted according to the criterion
Airport	sortByBoardingGates	setupScenary1	• None	The list of flights was sorted according to the criterion
Airport	sortByDate	setupScenary1	• None	The list of flights was sorted according to the criterion
Airport	searchTest	setupScenary1	• None	The searching algorithms find the correct item

