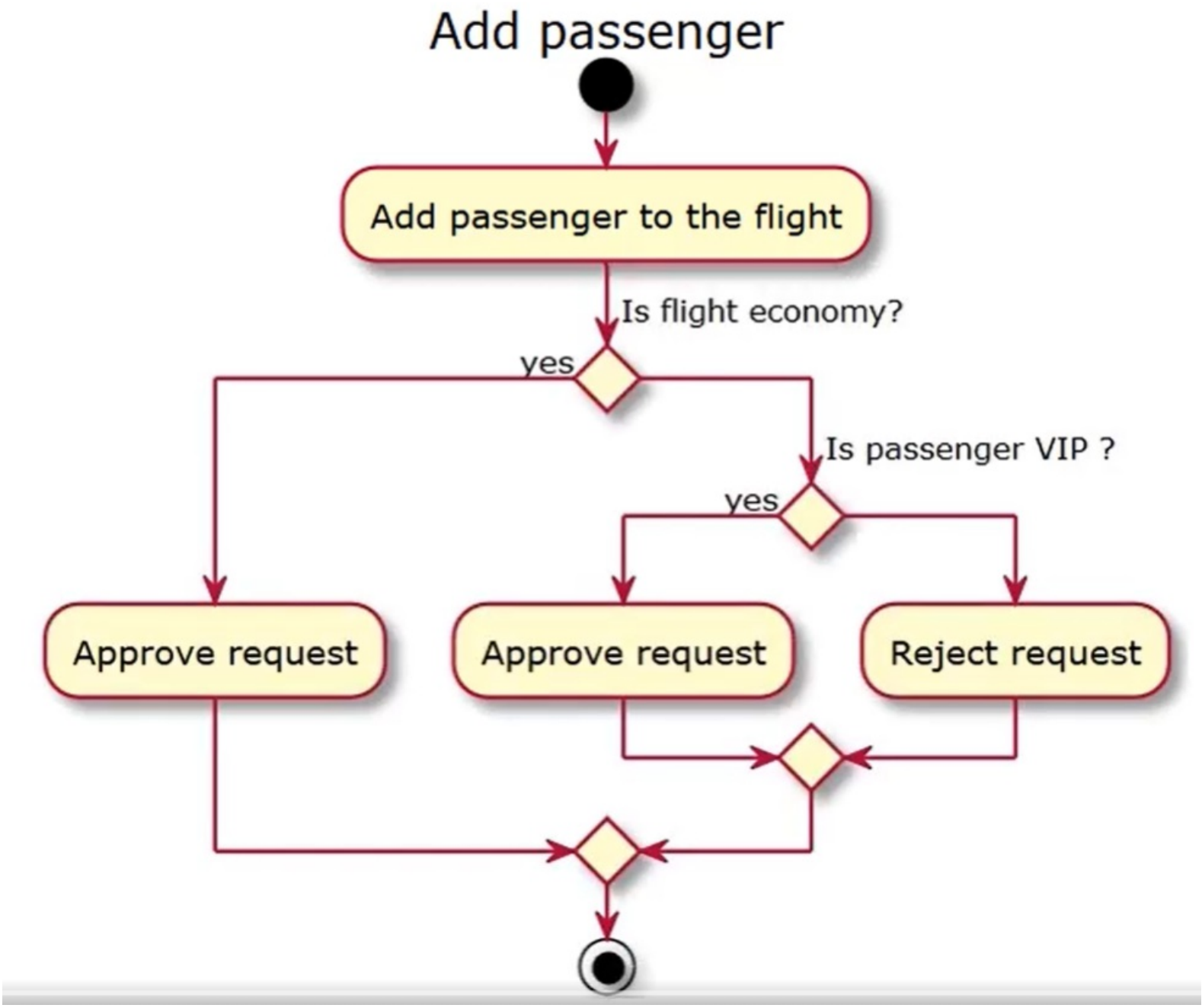
Flight Management

Assumptions

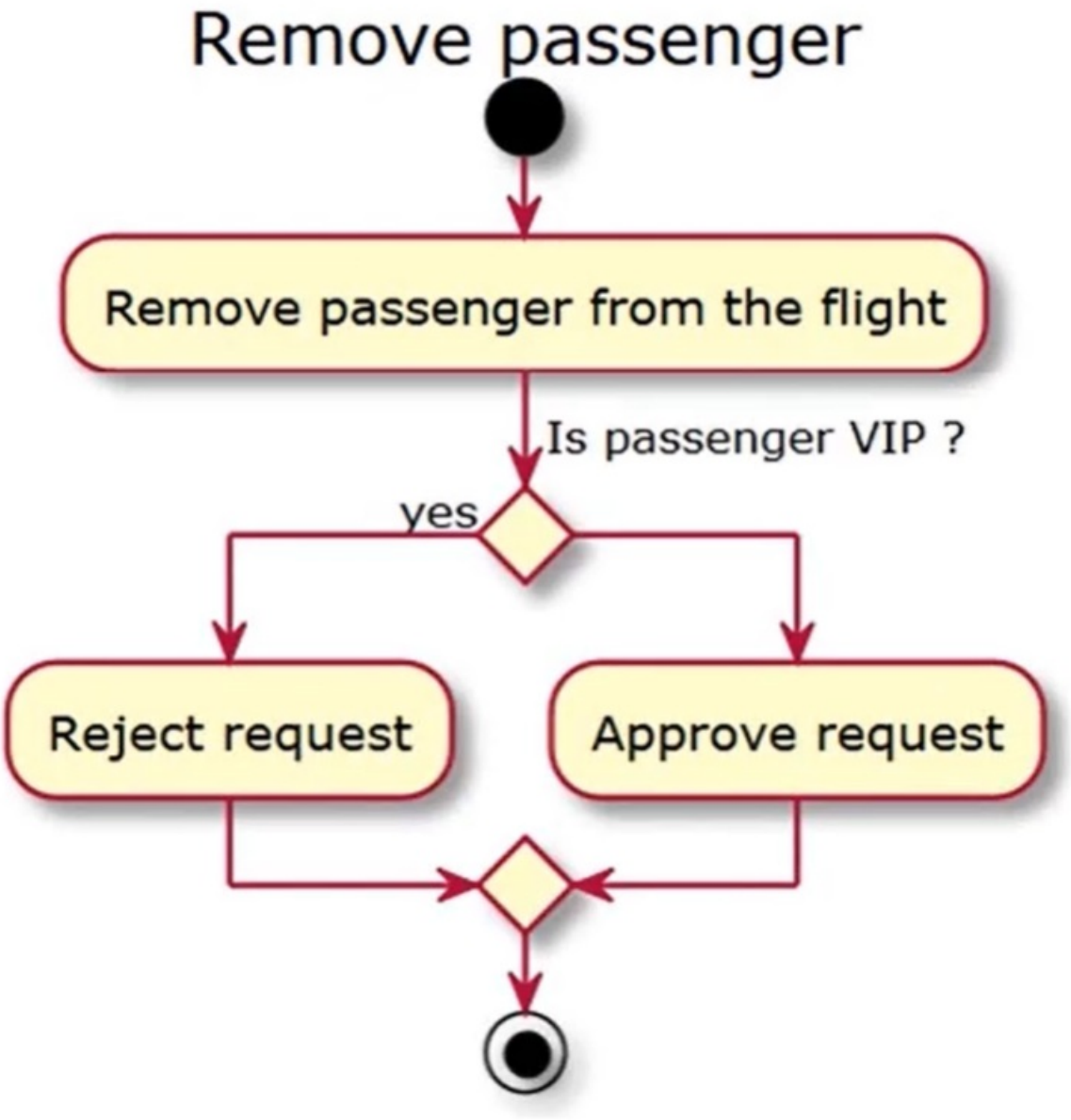
1. Flights can be of multiple types but at present only business and economy are supported
2. If the flight is an economy flight both VIP and non-VIP passengers maybe added into it
3. If the flight is a business flight only VIP passengers maybe added into it
4. While removing passengers from a flight VIP passenger cannot be removed

Functionalities of the application

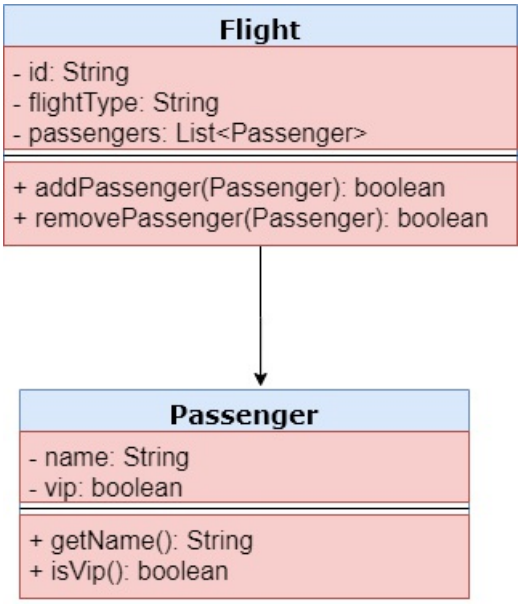
1. Add Passenger



1. Remove Passenger



Model classes



REST Url's

HTTP Method	URL	Parameters	Remarks
POST	/flights/business		
POST	/flights/economy	Accepts an object of passenger. Use Lombok for boilerplate code and appropriate logging	
DELETE	/flights/business		
DELETE	/flights/economy		

Requirements

1. Use Lombok for boilerplate code
2. Use logging level INFO for successful completion of the function and ERROR for failed execution
3. The controller should have an exception handler that will process the InvalidFlightOperation exception, in this logging level ERROR must be used. Log the error message and the error stack trace as well.
4. Configure the log output format as shown in the sample. Sample format of the log

Mon 13:03:12 09/11/2020 [restartedMain] [INFO] [TomcatWebServer] - Tomcat started on port(s): 8080 (http) with context path ''

1. All the warnings and errors must be written into a file called *application.log* located at the folder D:\logs
2. Set the application default log level to INFO
3. Create a ConsoleAppender and FileAppender to complete these tasks

Test Scenarios

1. Add a economy flight
2. Add a business flight
3. Add a VIP passenger to economy flight
4. Add a VIP passenger to business flight
5. Add a non-VIP passenger to economy flight
6. Add a non-VIP passenger to business flight
7. Remove a VIP passenger to economy flight
8. Remove a VIP passenger to business flight
9. Remove a non-VIP passenger to economy flight
10. Remove a non-VIP passenger to business flight

Requirements

1. Create a test class called FlightUnitTest and perform unit tests using junit
2. Perform code quality analysis using sonarqube
3. Perform code coverage analysis using jacoco