Java Developer Assessment Project ("JAir")

Tech specs

The following technologies must be used, everything else is up to you (as long as it makes sense).

- Java 11+
- A Java web container (e.g. Tomcat, Undertow, ...)
- ECMAScript or Typescript (with any libraries/framework you like)
- Maven or Gradle (or Bazel)
- An SQL database like Postgres or MySQL, or in-memory solution like H2

We already provide a skeleton project using Maven, Spring Boot and H2 which you can build upon. But you can also prefer alternatives, don't forget to copy static resources in such case. You can additionally use any tool of your choice.

Overview

You're about to build a new little Java/Web/JS platform, which should test your skills in

- Security (ℹ please take special care of this)
- Concurrency (ℹ also take special care of this)
- CSS-Stylesheets
- Caching on the client side (optional)
- (plus maybe some more)

Part #1 - Implementing the platform

Your platform - let's call it "JAir" - will be used by airlines, and our current customers are Lufthansa, Swissair and Virgin Atlantic.

In our imaginary world, these airlines don't want to employ their own pilots anymore, but want to charter them using our platform, so they can make use of a common pool of pilots.

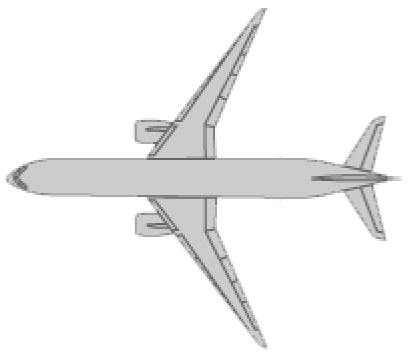
Each airline will have one user on our platform ("lufthansa", "swissair" and "virgin" respectively).

Each airline has several airplanes:

- **Lufhansa** has 12 Boeing 787, and 8 Airbus A350
- **Swissair** has 9 Boeing 787, and 2 Airbus A350
- **Virgin** has 10 Boeing 787

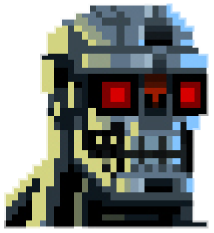
Here are images of the airplanes, which you'll use later:

Boeing 787	**Airbus A350**

And then we have our much valued little pool of pilots:

As 7 of them are currently on vacation (or severely injured), you'll only work with the remaining 5:

James Doolittle	Noel Wien	Robert Hoover	Charles Lindbergh	Charles Yeager
				

Locate these five images, the images of the airplanes and brand logos under resources/static folder, and put them into your DB tables - either as links to files which you store on your server, or as blobs - you choose. Note: You'll design all DB tables you need yourself!)

Functionality

Logging in

When visiting the platform page (https://jair.com), you'll see a login form (keep it as simple as possible). If you don't want to set up HTTPS for your server in the first step, it's ok, this could be done at the end.

User: _____________
Password: ____________

[Submit]

Make sure, that the form **shows an error to the user**, if the user or password was wrong. Also show an error, if the server is not reachable/some other problem occurs.

We don't have any user interface for registration - so just insert the users "lufthansa", "swissair" and "virgin" with a password manually into your database **Question: What do you have to make sure of, so this is safe?**

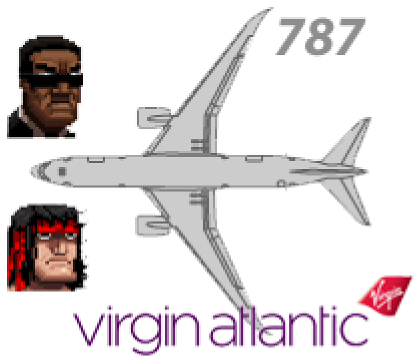
Logged in

After being logged in, a **cookie** should be stored on the client side, which keeps you logged in **for an hour**.

The user will immediately see an overview page (Note: you can either use a real separate page, or have login and overview pages combined in one page, AJAX style).

Overview page

This page shows a list of the 200 most recently booked flights. Each flight should be displayed as a compact composite image like this:

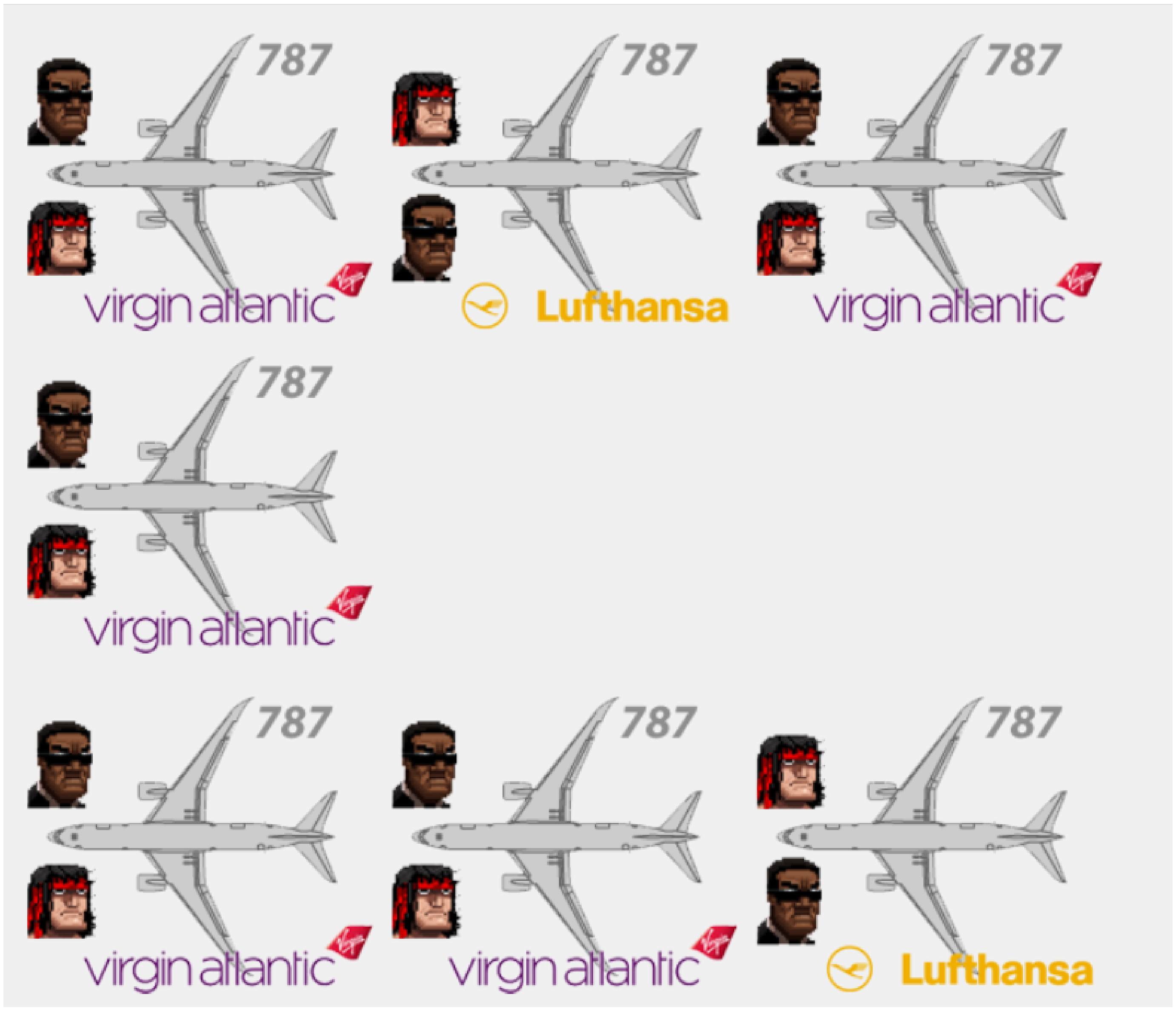


You can create such an image by using the image files from above, and overlaying them (we've already made them transparent, so this should not be too hard). This overlaying should be done on the client side, and it should be easy to achieve with simple CSS.

Please note, that there will be one pilot, and one co-pilot.

- The **pilot always sits on the left hand side** of the cockpit, and therefore is displayed on the bottom left of the composite image,
- whereas the **co-pilot sits on the right hand side** of the cockpit, and is therefore displayed on the top left.

The overview page should **make good use of the screen space**, so it should have more than one such image per line, like this:



Question: What needs to happen, when the browser window gets resized? How can you make sure, that rendering of so many elements is fast?

Also add a logout button somewhere on the overview page.

Charter pilots for a new flight

Somewhere on the overview page, there must be a button which opens a dialog to create a new flight.

The dialog should look something like this:

Airline: Swissair

Airplane: (----Please select-----)

Pilot: (----Please select----)

Copilot: (----Please select----)

[Book now]

Please note, that the Airline field is fixed (determined by the user login). You have to take care of which airplanes an airline has, so the list of airplanes should look like (e.g. for Swissair which has 9 Boeings and 2 Airbus):

```
Boeing 787 #1
Boeing 787 #2
Boeing 787 #3
Boeing 787 #4
Boeing 787 #5
Boeing 787 #6
Boeing 787 #7
Boeing 787 #8
Boeing 787 #9
Airbus A350 #1
Airbus A350 #2
```

But the pilots (and copilots) are charted from a common pool, so the list of available pilots is the same for all airlines (see the 5 pilots we mentioned above).

You do NOT have to take care of any flight departure dates, and you may assume, that **all pilots and all airplanes are always available**. Just make sure, that you **cannot book the same guy both as the pilot and the copilot** for the same flight!

For each pilot, there's a **10% random chance, that he feels sick**, and doesn't want to fly. In that case, the booking fails (just show "Booking failed because XY feels sick" and close the dialog in this case).

Part #2 - Earning medals (Up to you - We really encourage you to do it!)

Now we'll introduce a little special feature in which you can demonstrate, how you deal with concurrency...

Each pilot can earn medals, depending on how many time he/she's been booked: A bronze medal at 10 bookings, silver at 100 and gold at 1000. (For this count, it doesn't matter, if you've been booked as a pilot or copilot). Each medal can only be earned once, so e.g. a pilot doesn't get a second bronze medal at 20 bookings.

Now here's what's important:

- A booking (if it is successfully committed) cannot be undone
- A medal must be awarded immediately when the required number of (successful) bookings is reached. Send out a congratulation email is sent out to the pilot immediately (note: you don't have to actually send it, simply write a log output "Sending email: To: 'Robert Hoover' Subject: 'You just earned your bronze medal'")
- A medal must only be given once (and the email would only be sent out once)
- A medal, once awarded, cannot be taken back.
- It is not allowed - not even for a short period of time - that some pilot has e.g. 99 successful (committed) bookings, but has already been awarded a silver medal.

Question: What do you need to make sure of, so that this always works correctly? In which ways is this a concurrency problem? (Do consider, that multiple users can use the system at the same time... Please make sure, that multiple bookings can be processed concurrently, otherwise the solution would be a little too easy!)

Part #3 - Bonus (Optional)

If you feel, that all of this was too easy... you may, if you want, add the following improvements (maybe partially)

- Let the overview page show much more than the last 200 flights - maybe 10000, or even an indefinite amount by scrolling down further and further
- Try to load these previous flights via chunking (in the background). We know, that this would maybe not even be so great for a real product (**Question: What could be the drawbacks?**), but for this test project it would be interesting.
- Maybe also save them in localStorage or indexedDB. **Question: If you do this, can this be a security issue - and how could you maybe resolve it?**