|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Objects | Properties | Messages | Output |
| As an air traffic controller  So I can get passengers to a destination  I want to instruct the airport to land a plane | Airport  Plane | land(plane) =  plane@Array  plane(@String) | “Land”  Add(@String)  “ | String or array??  plane[@String] |
| As the system designer  So that the software can be used for many different airports  I would like a default airport capacity that can be overridden as appropriate |  |  |  |  |
| As an air traffic controller  To ensure safety  I want to prevent landing when the airport is full | Landing  Plane |  | ‘Land’  ‘Airport full’  Land plane | True/ False  @Boelean (true)  If full, do not land  Else, land  String- Cannot land yet, plane is full |
| As an air traffic controller  So I can get passengers on the way to their destination  I want to instruct the airport to let a plane take off and confirm that it is no longer in the airport | Take off plane |  | ‘Depart’ |  |
| As an air traffic controller  To avoid confusion  I want to prevent asking the airport to let planes take-off which are not at the airport, or land a plane that's already landed |  |  |  |  |

**Instructions**

* Feel free to use google, your notes, books, etc. but work on your own.
* Keep it SIMPLE - it's not nearly as complicated as it first may look.
* You must [submit your challenge](https://airtable.com/shrUGm2T8TYCFAmjN) by 9am Monday morning, wherever you get to.
* Use your own test framework and evidence your test-driven development by committing on passing tests.
* Please write your own README detailing how to install your project, how to run the tests, how you approached the problem and provide screenshots of interacting with your program.
* If you refer to the solution of another coach or student, please put a link to that in your README.
* Please create separate files for every class, module, and spec.

**Steps**

1. Fork this repo, and clone to your local machine
2. npm install to install project dependencies
3. Convert stories into a representative domain model and test-drive your work.
4. Run your tests using npm test or node specRunner.js
5. [Lint](https://eslint.org/docs/user-guide/getting-started) your source code using npx eslint src

**Task**

We have a request from a client to write the software to control the flow of planes at an airport. The planes can land and take off provided that the weather is sunny. Occasionally it may be stormy, in which case no planes can land or take off. Here are the user stories that we worked out in collaboration with the client:

**User Story 1**

As an air traffic controller

So I can get passengers to a destination

I want to instruct the airport to land a plane.

|  |  |  |  |
| --- | --- | --- | --- |
| Objects | Properties | Messages | Output |
| Airport | hangar@Array | Land(plane) | @Array[@planes] |
| Plane | Items @Array[@Item] | addItem(@String) | @Item[item] |

**User Story 2**

As the system designer

So that the software can be used for many different airports

I would like a default airport capacity that can be overridden as appropriate

|  |  |  |  |
| --- | --- | --- | --- |
| Objects | Properties | Messages | Output |
| Airport | capacity |  | @ |
|  |  |  |  |

**User Story 3**

As an air traffic controller

To ensure safety

I want to prevent landing when the airport is full

|  |  |  |  |
| --- | --- | --- | --- |
| Objects | Properties | Messages | Output |
| Airport |  | Airport full | @Boolean TRUE |
|  |  | Space available | @Boolean FALSE |
|  |  |  |  |

**User Story 4**

As an air traffic controller

So I can get passengers on the way to their destination

I want to instruct the airport to let a plane take off and confirm that it is no longer in the airport

|  |  |  |  |
| --- | --- | --- | --- |
| Objects | Properties | Messages | Output |
| Airport |  | takeOff(@plane) | @String confirmed take off |
|  |  | No take off, plane flying | @ |
|  |  |  |  |

**User Story 5**

As an air traffic controller

To avoid confusion

I want to prevent asking the airport to let planes take-off which are not at the airport, or land a plane that's already landed

|  |  |  |  |
| --- | --- | --- | --- |
| Objects | Properties | Messages | Output |
| Airport |  | Land() | @String ‘Plane ‘ |
|  |  | Different airport | @String ‘Plane has landed in a different airport’ |
|  |  |  |  |