```
As an air traffic controller
So I can get passengers to a destination
I want to instruct a plane to land at an airport
```

Objects	Messages
Air traffic controller	instruct
Passengers	
Destination	
Plane	land
Airport	

Airport ← land → Plane

```
As an air traffic controller
So I can get passengers on the way to their destination
I want to instruct a plane to take off from an airport and confirm that it is

no longer in the airport
```

Objects	Messages
Air traffic controller	
Passengers	
Destination	
Plane	take_off
Airport	plane_at_airport?

Airport ← take_off → Plane
Plane ← plane_at_airport? → true/false

```
As an air traffic controller
To ensure safety
I want to prevent landing when the airport is full
```

Objects	Messages
Air traffic controller	prevent_landing

safety	ensured?
Airport	airport_full?

Airport ← full? → true/false → prevent_landing if true

_____

```
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	Messages
System designer	override_capacity
Software	be_used
Airport	default_capacity

DEFAULT_CAPACITY ← override_capacity → new_capacity

_____

```
As an air traffic controller
To ensure safety
I want to prevent takeoff when weather is stormy
```

Objects	Messages
Air traffic controller	prevent_takeoff
safety	ensured?
Weather	stormy?

Weather ← stormy? → true/false → prevent_take_off if true

_____

```
As an air traffic controller
To ensure safety
I want to prevent landing when weather is stormy
```

Objects	Messages
Air traffic controller	prevent_landing

safety	ensured?
Weather	stormy?

Weather ← stormy? → true/false → prevent_landing if true