# Project Two Report

*Group 7 - Shannon Trudeau, Andrew Popovich, Justin Cotner*

## 1. Reflection on our Design and the Use of Actors

When first planning our actor-based design, we didn't take into account very much the individual actor's message queues that come built into the Akka system. We changed our design to take advantage of these message queues so that we could remove some of the unnecessary messages that were being passed between actors. Other than those changes (mentioned below) our design from Part 1 matched the implementation exactly.

However, as a whole the use of actors in this project has made the design and implementation of the airport security check simulation significantly easier than what it would have been through the use of traditional concurrency mechanisms. Specifically this is due to the fact that shared mutable state is shared between actors through the use of immutable messages, which means that we did not have to account for maintaining that state throughout the design and implementation of the system. In other words, we did not have to worry about the problems associated with the use of traditional concurrency methods such as deadlocks and race conditions because of how messages between actors are immutable. Also, the system was easier to design because the use of the actor paradigm enabled us to think about the process as if each actor represented a person, as opposed to having to think about the process in terms of the interaction of threads or other code mechanisms. As a result the design and implementation appropriately reflects this airport security process through the use of interactions that would likely occur in the real world. Overall, we think that the actor paradigm should be used in concurrent systems whenever it makes sense to break up functionality in terms of actors that communicate and work with messages, such as with systems that have clear distinctions behind the responsibilities for their modules and the interactions between their modules.

## 2. Changes to Original Design

- We needed to add a jail actor which we didn't initially anticipate. Originally, we planned on having the jail represented as a simple list of passengers that was shared across all of the security actors. However, this changed was due to the requirement that there should be no shared mutable state and that any communication between actors needed to occur as a message.
- Removed BodyScanReady message because it created a circular dependency with the QueueActor, in which it would have been difficult to instantiate these two actors. Instead we simply use the BodyScanActor's message queue for holding passengers as they wait to go through the Body Scanner.

## 3. Actor-Based Design

The following information shows the updated design that was originally created in Part 1 of the project.

| Name: Document Check Actor |
| --- |
| **State information (What does the actor know?):** |

- Has a list of Queue Actors
- Probability of randomly failing paperwork checks - default is 20%

**Responsibilities (What does the actor do?):**

- Check paperwork, pass or fail
- Direct passenger to the next security line if pass
- Inform the lines that it is the end of the day

| **Messages Received** | | | |
|---|---|---|---|
| **Message class** | **Sender** | **Contents** | **Resulting action or effect** |
| Passenger | Main | Passenger information | Pass or fail the passengers paperwork |
| EndOfDay | Main | None | Sends the message along to all of the lines |
| **Messages Sent** | | | |
| **Message class** | **Recipient** | **Contents** | **Purpose and trigger** |
| Passenger | A Queue Actor | Passenger information | Purpose: Add the passenger to the queue<br>Trigger: A passenger has passed the document check |
| EndOfDay | All Queue Actor | None | Purpose: Tells the lines no more passengers will be coming.<br>Trigger: Received the EndOfDay message. |

---

Name: Queue Actor

**State information (What does the actor know?):**

- ConcurrentLinkedQueue - A queue representing a line of passengers waiting to go through the body scanner and bag scanner
- Reference to the line's Bag Scan Actor
- Reference to the line's Body Scan Actor

**Responsibilities (What does the actor do?):**

- Maintains a queue of passengers in a line
- Passes along the passengers bags to the Bag Scan Actor as soon as the actor gets in line
- Passes along the passenger to the Body Scan Actor when the Body Scan Actor is ready for the next person

**Messages Received**

| Message class | Sender | Contents | Impact or effect |
|---|---|---|---|
| Passenger | Document Check Actor | Passenger's information | Add the passenger to the queue |
| EndOfDay | Document Check Actor | Nothing | Send to other actors in the line |

**Messages Sent**

| Message class | Recipient | Contents | Purpose and trigger |
|---|---|---|---|
| Passenger | Body Scan Actor | Passenger's information | Purpose: Queue sends the next passenger to the body scanner<br>Trigger: The Body Scan Actor in the line is ready for the next passenger |
| Bag | Bag Scan Actor | Passenger's ID | Purpose: Send the passenger's bags to the Bag Scan Actor so that the bags can be checked<br>Trigger: A Passenger enters the queue |
| EndOfDay | | nothing | Purpose: Shutdown, send message to other actors in the line<br>Trigger: EndOfDay message is received from the Document Check Actor |

---

| Name: Bag Scan Actor |
|---|
| **State information (What does the actor know?):**<br>● Reference to the Security Actor<br>● Probability of passengers randomly failing - default is 20%<br>● Integer representing the queue number that this actor resides in |
| **Responsibilities (What does the actor do?):**<br>● Scan the passenger. Pass or Fail them<br>● Send the result to the Security Actor for the queue that this actor resides in |

| **Messages Received** | | | |
|---|---|---|---|
| **Message class** | **Sender** | **Contents** | **Impact or effect** |
| EndOfDay | Queue Actor | Nothing | Ppropagate message to the Security Actor |
| Bags | Queue Actor | Passenger ID | The passenger has placed their bags on the bag check, right after entering the queue and the bags are scanned |
| **Messages Sent** | | | |
| **Message class** | **Recipient** | **Contents** | **Purpose and trigger** |
| Result | Security Actor | Passenger's information, pass/fail boolean, and either 0 or 1 to distinguish the message sender (Body Scan Actor or Bag Scan Actor) | Purpose: Send the results to the Security Actor<br>Trigger: A bag has been scanned |

---

| Name: Body Scan Actor |
|---|
| **State information (What does the actor know?):**<br>● Reference to the Security Actor |

- Reference to the Queue Actor
- Probability of passengers randomly failing - default is 20%
- Integer representing the queue number that this actor resides in

**Responsibilities (What does the actor do?):**

- Scan the passenger. Pass or Fail them
- Send the result to the Security Actor for the queue that this actor resides in
- Send a message back to the Queue Actor so that another passenger can be sent to this actor

| **Messages Received** | | | |
|---|---|---|---|
| **Message class** | **Sender** | **Contents** | **Impact or effect** |
| Passenger | Queue Actor | Passenger information | Passenger is at the head of the queue and ready to be scanned |
| EndOfDay | Queue Actor | None | Propagates message to the rest of the actors for the line |
| **Messages Sent** | | | |
| **Message class** | **Recipient** | **Contents** | **Purpose and trigger** |
| Result | Security Actor | Passenger's information, pass/fail boolean, and either 0 or 1 to distinguish the message sender (Body Scan Actor or Bag Scan Actor) | Purpose: Send report to this queue's Security Actor<br>Trigger: A Passenger has went through the body scan |
| EndOfDay | Security Actor | None | Purpose: Tell Security actor no more passengers will be coming.<br>Trigger: The EndOfDay message is received from the queue |

---

Name: Security Actor

**State information (What does the actor know?):**

- Reference to Jail Actor- a reference to the Jail Actor
- HashMap - representation of the passengers that have gone through at least once scanner and the results of both scanners. Key - Passenger's name or ID. Value - Array containing two booleans to store the result from each scanner
- Integer representing the number of Scan Actors that have shut down in the line

**Responsibilities (What does the actor do?):**

- Receive the results from both the Bag Scan Actor and Body Scan Actor and store those results by passenger
- Once both results are in for a passenger check those results and either let that passenger through or send him/her to the Jail Actor
- Once both Scan Actors have shut down the Security Actor shuts down

| **Messages Received** | | | |
|---|---|---|---|
| **Message class** | **Sender** | **Contents** | **Impact or effect** |
| Result | Bag Scan Actor<br>Body Scan Actor | Passenger's information, pass/fail boolean, and either 0 or 1 to distinguish the message sender (Body Scan Actor or Bag Scan Actor) | If it the first result for the Passenger, add it to the HashMap. Otherwise update the HashMap and send the passenger to Jail if applicable. |
| EndOfDay | Body Scan Actor | None | Once both of the Scan Actors in the Secuity Actor's queue have shut down then this actor sends an EndOfDay message to the Jail Actor |
| **Messages Sent** | | | |
| **Message class** | **Recipient** | **Contents** | **Purpose and trigger** |
| Passenger | Jail Actor | Passenger's information | Purpose: Add the passenger to the Jail<br>Trigger: A passenger does not pass one of the scans |

| | | | |
|---|---|---|---|
| EndOfDay | Jail Actor | None | Purpose: Tell the Jail Actor to send the prisoners to permanent holding at the end of the day<br>Trigger: The EndOfDay Message has been received by both of the Scan Actors |

---

| Name: Jail Actor |
|---|
| **State information (What does the actor know?):**<br>● List of Passengers - representation of passengers that did not pass the security check<br>● Integer representing the number of queues |
| **Responsibilities (What does the actor do?):**<br>● Receive the passengers from each Security Actor that did not pass the security check<br>● Shut down and send the jailed passengers to permanent holding at the end of the day |

| **Messages Received** | | | |
|---|---|---|---|
| **Message class** | **Sender** | **Contents** | **Impact or effect** |
| Passenger | Security Actor | Passenger's information | Add the passengers to the list of passengers to send to jail |
| EndOfDay | Security Actors | None | Once all of the Security Actors have sent their EndOfDay message send all of the passengers jailed to permanent holding and shutdown all actors |

# 3. Actor Collaboration Diagram