Airport Boarding

Scenario

- We differentiate between three options for check-in:
  - **Normal check-in** with luggage at a normal check-in counter
  - **Expres scheck-in** without luggage at a special check-in counter
  - **Automated check-in** without luggage at a machine
- In addition to the passenger, who representstravelers, there is the check-in representative. The check-in representative isa person who is not the actual passenger, but an agent of the passenger. Thecheck-in representative has the task of performing the check-in with the ticketof the passenger.
- The check-in procedure includes submitting theticket, baggage check-in, seat reservation, and issuing and handing over theboarding pass.
- Passengers who only have hand luggage can useexpress check-in. No baggage check-in is performed.
- During boarding, the boarding pass of thepassenger is verified at the gate.
- Automated check-in is conducted without the helpof a check-in clerk, directly at a machine (screen). Baggage cannot bechecked in.
- The passenger can choose between a normalcheck-in, automated check-in, and express check-in. The passenger walks to thegate and presents his or her boarding pass. The check-in representative canperform a regular check-in, but is not able to perform express check-in andautomated check-in.
- If we take a closer look, we notice that apassenger often travels with luggage, which he or she checks in. Baggagetransportation is responsible for loading luggage into the airplane. Baggagetransportation is carried out by an independent organization, known as ahandling agent. Consequently, it is considered an actor, more specifically, anoutside service provider. It does not matter for our diagram that individualemployees of the partner enterprise perform these tasks.
- Ten minutes before a flight leaves, baggagetransportation requests a passenger list from passenger services, whichincludes every passenger who checked in, but did not board the airplane. On thebasis of this list all affected luggage will be unloaded again from theairplane. If the flight is an international flight, the customs authorities ofthe country in which the destination airport is located also request apassenger list.

UML Diagrams with plantUML codes

1) Class Diagram

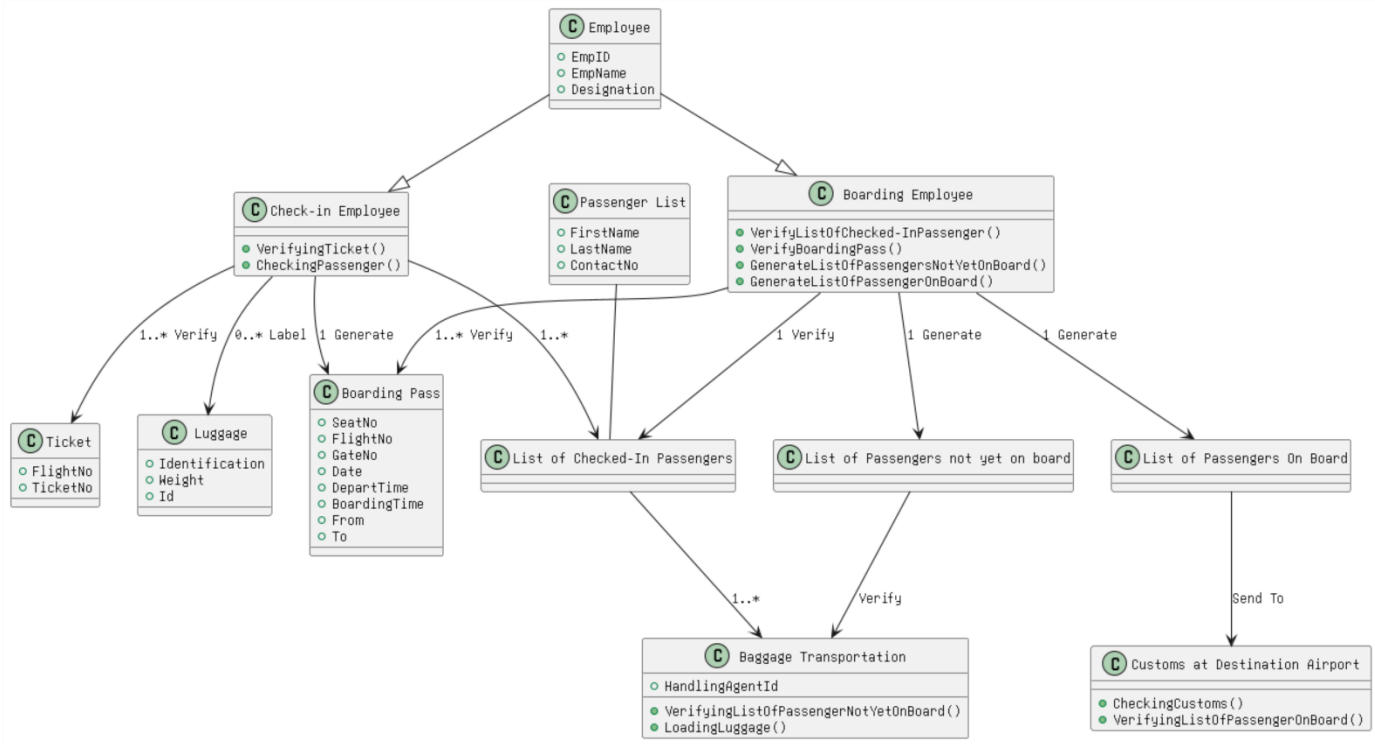
PlantUML code:

```
@startuml
class Employee {
  +EmpID
  +EmpName
  +Designation
}

class Ticket {
  +FlightNo
```

```
  +TicketNo
}

class Luggage {
  +Identification
  +Weight
  +Id
}

class "Check-in Employee" {
  +VerifyingTicket()
  +CheckingPassenger()
}

class "Boarding Pass" {
  +SeatNo
  +FlightNo
  +GateNo
  +Date
  +DepartTime
  +BoardingTime
  +From
  +To
}

class "Boarding Employee" {
  +VerifyListOfChecked-InPassenger()
  +VerifyBoardingPass()
  +GenerateListOfPassengersNotYetOnBoard()
  +GenerateListOfPassengerOnBoard()
}

class "List of Checked-In Passengers"

class "List of Passengers not yet on board"

class "List of Passengers On Board"

class "Baggage Transportation" {
  +HandlingAgentId
  +VerifyingListOfPassengerNotYetOnBoard()
  +LoadingLuggage()
}

class "Passenger List" {
  +FirstName
  +LastName
  +ContactNo
}
```

```
class "Customs at Destination Airport" {
  +CheckingCustoms()
  +VerifyingListOfPassengerOnBoard()
}

Employee --|> "Check-in Employee"
Employee --|> "Boarding Employee"

"Check-in Employee" --> Ticket : "1..* Verify"
"Check-in Employee" --> Luggage : "0..* Label"
"Check-in Employee" --> "Boarding Pass" : "1 Generate"
"Check-in Employee" --> "List of Checked-In Passengers" : "1..*"

"Boarding Employee" --> "Boarding Pass" : "1..* Verify"
"Boarding Employee" --> "List of Checked-In Passengers" : "1 Verify"
"Boarding Employee" --> "List of Passengers not yet on board" : "1 Generate"
"Boarding Employee" --> "List of Passengers On Board" : "1 Generate"

"List of Checked-In Passengers" --> "Baggage Transportation" : "1..*"

"List of Passengers not yet on board" --> "Baggage Transportation" : "Verify"
"List of Passengers On Board" --> "Customs at Destination Airport" : "Send To"

"Passenger List" -- "List of Checked-In Passengers"
@enduml
```

Diagram:

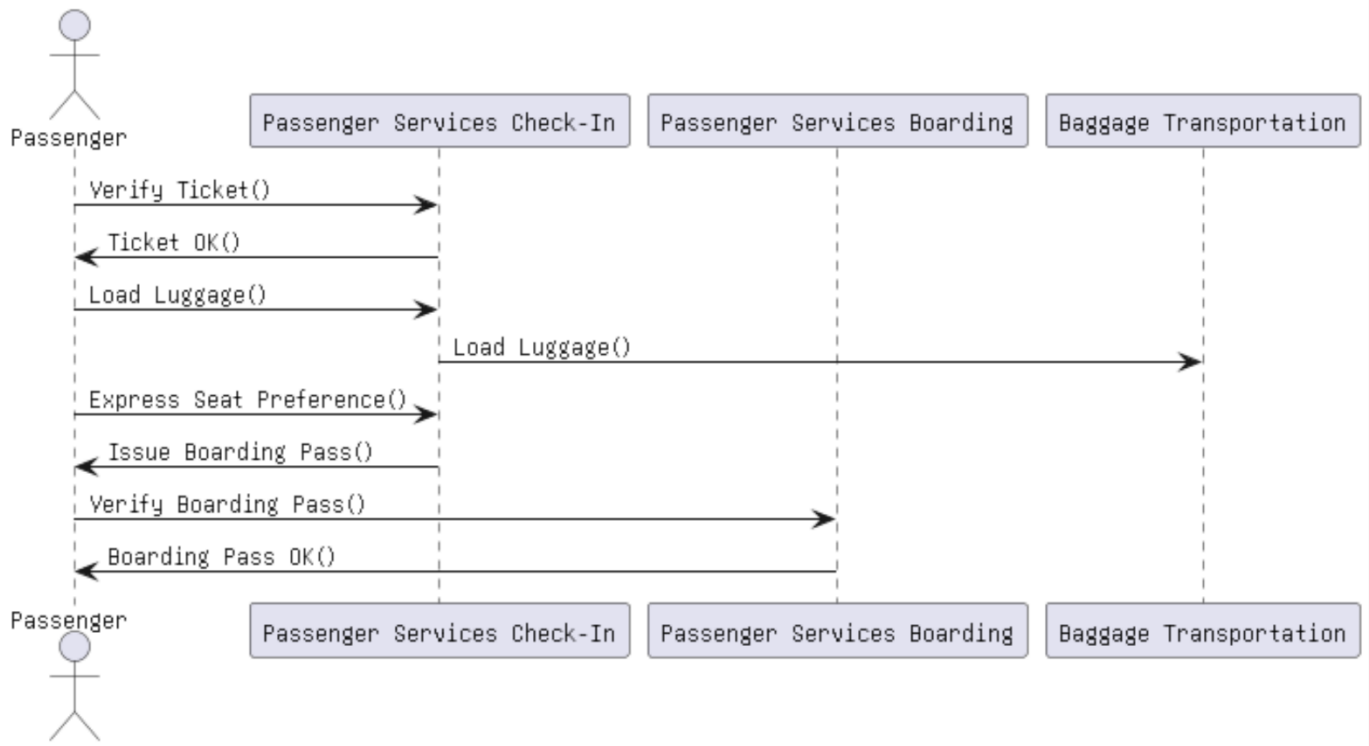


2) Sequence Diagram

PlantUML code:

```
@startuml
actor Passenger
participant "Passenger Services Check-In"
participant "Passenger Services Boarding"
participant "Baggage Transportation"

Passenger -> "Passenger Services Check-In" : Verify Ticket()
"Passenger Services Check-In" -> Passenger : Ticket OK()

Passenger -> "Passenger Services Check-In" : Load Luggage()
"Passenger Services Check-In" -> "Baggage Transportation" : Load Luggage()

Passenger -> "Passenger Services Check-In" : Express Seat Preference()
"Passenger Services Check-In" -> Passenger : Issue Boarding Pass()

Passenger -> "Passenger Services Boarding" : Verify Boarding Pass()
"Passenger Services Boarding" -> Passenger : Boarding Pass OK()
@enduml
```

Diagram:

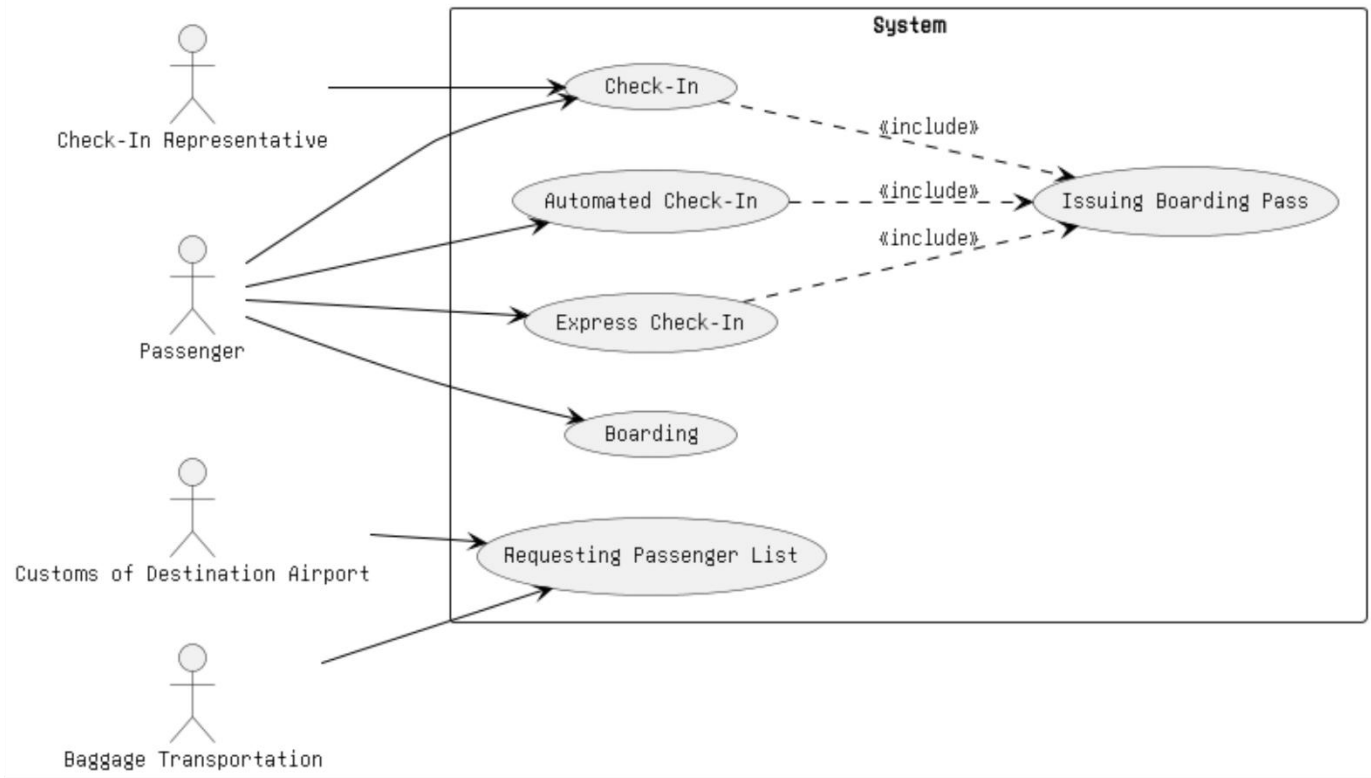


3) Use Case Diagram

PlantUML code:

```
@startuml
left to right direction

actor "Check-In Representative" as CIR
actor "Passenger" as P
actor "Customs of Destination Airport" as CDA
actor "Baggage Transportation" as BT

rectangle "System" {
  usecase "Check-In" as UC1
  usecase "Automated Check-In" as UC2
  usecase "Issuing Boarding Pass" as UC3
  usecase "Express Check-In" as UC4
  usecase "Boarding" as UC5
  usecase "Requesting Passenger List" as UC6

  UC1 -[#black,dashed]-> UC3 : <<include>>
  UC2 -[#black,dashed]-> UC3 : <<include>>
  UC4 -[#black,dashed]-> UC3 : <<include>>
}

P -[#black]-> UC1
P -[#black]-> UC2
P -[#black]-> UC4
P -[#black]-> UC5

CIR -[#black]-> UC1
CDA -[#black]-> UC6
BT -[#black]-> UC6

@enduml
```

Diagram:



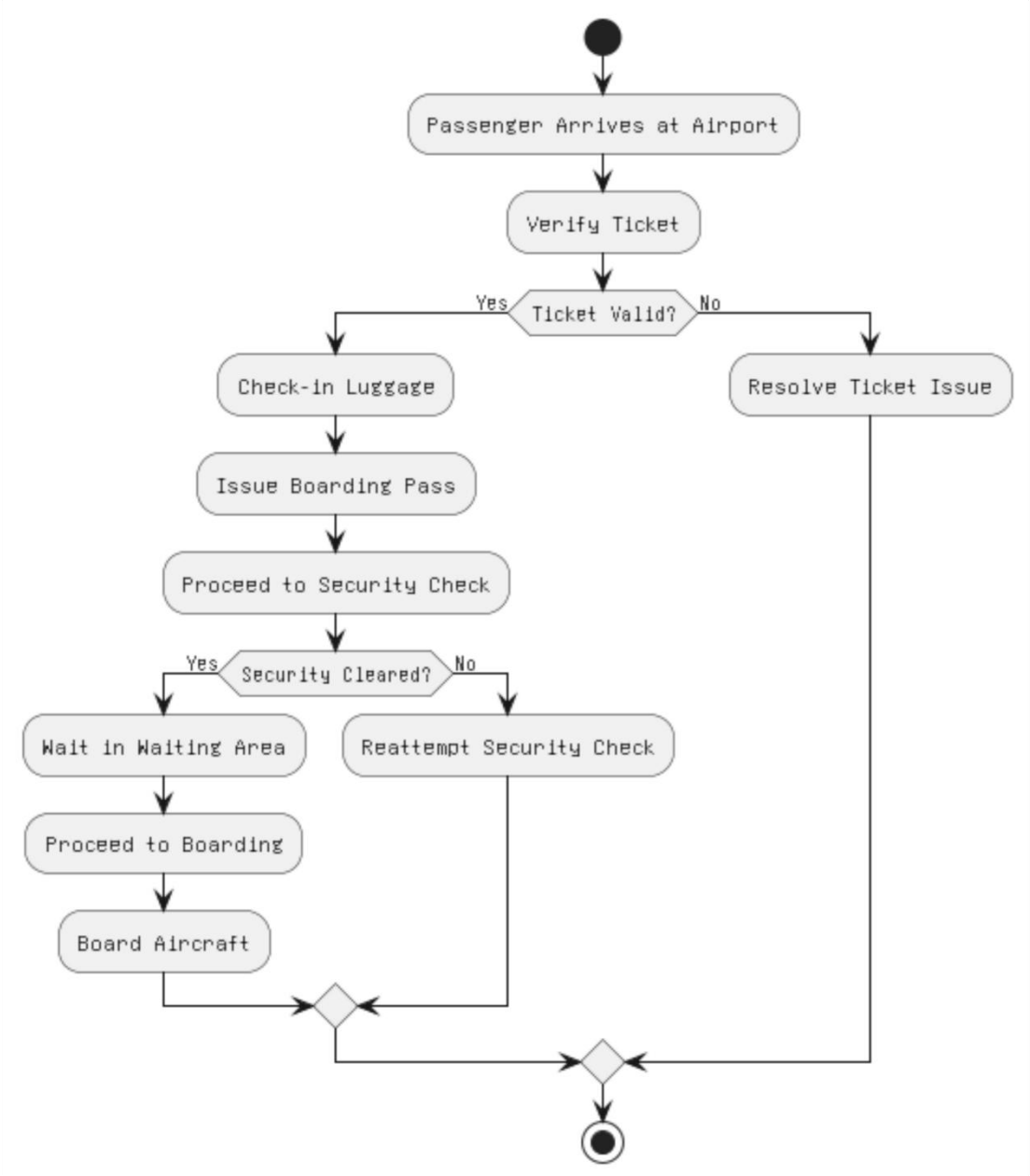
4) Activity Diagram

PlantUML code:

```
@startuml
start
:Passenger Arrives at Airport;
:Verify Ticket;

if (Ticket Valid?) then (Yes)
   :Check-in Luggage;
   :Issue Boarding Pass;
   :Proceed to Security Check;

   if (Security Cleared?) then (Yes)
      :Wait in Waiting Area;
      :Proceed to Boarding;
      :Board Aircraft;
   else (No)
      :Reattempt Security Check;
   endif

else (No)
   :Resolve Ticket Issue;
endif
```

stop
@enduml

Diagram:



5) State Diagram

PlantUML code:

@startuml
[*] --> Idle

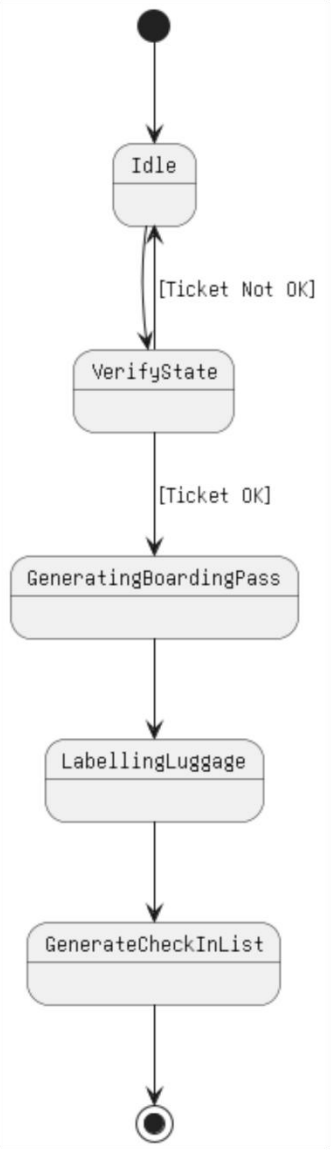
Idle --> VerifyState

VerifyState --> GeneratingBoardingPass : [Ticket OK]
VerifyState --> Idle : [Ticket Not OK]

GeneratingBoardingPass --> LabellingLuggage
LabellingLuggage --> GenerateCheckInList

GenerateCheckInList --> [*]
@enduml

Diagram:



6) Collaboration Diagram

PlantUML code:

```
@startuml
!pragma horizontalLineBetweenDifferentPackageAllowed true

' Define objects to match the layout
object "Baggage Transportation" as Baggage
object "Passenger Services Check-In" as CheckIn
object "Passenger" as Passenger
object "Passenger Services Boarding" as Boarding

' Arrange objects vertically to resemble the provided diagram
Baggage -[#black]-> CheckIn : 4 : Load Luggage()
CheckIn -[#black]-> Passenger : 2 : Ticket OK()
CheckIn -[#black]-> Passenger : 6 : Issue Boarding Pass()

Passenger -[#black]-> CheckIn : 1 : Verify Ticket()
Passenger -[#black]-> CheckIn : 3 : Load Luggage()
Passenger -[#black]-> CheckIn : 5 : Express Seat Preference()

Passenger -[#black]-> Boarding : 7 : Verify Boarding Pass()
Boarding -[#black]-> Passenger : 8 : Boarding Pass OK()
@enduml
```

Diagram: