**Slide 1**

Hello, my name is \_\_\_\_, and I will be doing this presentation on behalf of Group 14. The domain we have chosen to cover is Ryanair, specifically a ticket booking system, and in this presentation, I am going to be covering our UML Use Case diagram.

**Slide 2**

The first thing we did as a team was to try and decide what the actors of the system were, and list them out on a table. This helped us decide what the roles and possible use cases were, as we could put ourselves in an actor’s situation and think about what they might want to do with the system. For example, we agreed that as a passenger we might want to buy a ticket. Hence for a passenger, we knew that their role was that of an individual purchasing a ticket from the system. We went through this process with each actor to come up with the table shown on this slide.

**Slide 3**

After we decided what each actor’s role was, we went on to figure out what the use cases were. This involved both thinking of every action that would take place in the system, as well as putting ourselves in the actor’s situation and thinking about what we would want to do. From this we could create a table of use cases and the participating actors. Again, using the passenger as an example, we would want to get a ticket. This would mean we would want to choose a ticket and pay for the ticket. We may also have an account on the Airline’s Website or App and may want to login to them. We would also have to check-in at the airport. Hence, Passenger is a participating actor in the use cases ‘Login to Airline App/Website’, ‘Choose Ticket’, ‘Make Payment’, and ‘Check-in’.

**Slide 4**

From the tables in the previous slides, we could start working on the diagram. We had to change the diagram a few times as we discovered new use cases (and got advice in the group collab room). We also discovered relations between different use cases, such as functionalities that use cases might have in common (and use cases that do not interact with the actors).

**Slide 5**

We decided that there were too many use cases on the diagram and decided to reduce it down to 8 use cases. This gave us the diagram in this slide, which we continued to work on based on the advice we got (in the group collab room).

**Slide 6**

This led us to our current use case diagram, as shown on this slide. We decided to add colour to the text to make it easier to differentiate between the names of actors, use cases, use case relations, and the name of the system. This final diagram has 8 use cases and 9 actors. Starting from the actor ‘Passenger’, the passenger may do multiple things. They could login to the airline’s website or app, where they would then choose a ticket. They could also go to a price comparison website to find cheaper tickets. The website would use an algorithm to choose a ticket at a low price for the Passenger. They may also go to a ‘travel agent’, which will choose a ticket for them, or they may get a ticket at the ticket desk before the flight. Lastly, the Passenger would make the payment for the ticket, and check-in. The bank would verify the payment made by the Passenger. External services may offer services such as Car hire or accommodation bookings when the Passenger is choosing a ticket. An external Algorithm is used to decide what the cost of a ticket is. The Airline would issue a boarding pass to the passenger when they check in.

**Strengths and weaknesses**

A strength of this design is that booking a flight is easy and relatively quick.

One of the main weaknesses of our design is that there is nothing in place in the system for if the Passenger wishes to cancel or change flights once the ticket has been purchased. Another possible weakness is that the Passenger cannot buy tickets from the site themselves without an account. This could potentially limit the amount of customers the Airline gets.

***Note: May have to stop at this point due to the five minute time limit. Just skim these slides if so.***

**Slide 7**

Over the next few slides, I’ll be covering the descriptions of the use cases in our diagram, which are the ones that we thought were most important.

This is the use case ‘Login to Airline App/Website’. It describes the initial interaction the passenger has with the Ryanair IT system in order for them to gain access to the rest of the system. This use case has a single participating actor, the Passenger, who may want to buy a ticket through their own account. This use case has three preconditions. These are that the participating actor must have created an account for the website/app, have access to the internet, and must sign in to access flight information (I feel this isn’t a precondition, but what the use case actually is). There are also three exit conditions. These are that the actor does purchase a ticket, the actor does not purchase a ticket, or the actor chooses not to create an account. Under normal circumstances, if the actor does not have an account they must create one. The actor must use their own username and password to log in. Once logged in, information is provided about tickets, and the actor has the ability to navigate through the site. Error scenarios for this use case include the IT system not recognising the username and/or password and denying the actor access to their account, and internet access failing for the actor.

**Slide 8**

This is the use case ‘Choose Ticket’. It describes how a ticket is purchased in the IT system. A ticket can be purchased by any actor participating in this use case, of which there are four: these are the Passenger, the Travel Agent, the Ticket Desk, and an Algorithm to find cheap fares. This has a single entry condition, that is that the actor must be affiliated with the app/website through an account or work for the company. There are four exit conditions for this use case. The actor chooses the ticket they would like to purchase, or chooses not to purchase a ticket. If the actor is the Ticket Desk or the Travel Agent, they pick a ticket on behalf of the Passenger. Under normal circumstances, the actor chooses the ticket they want and the price is displayed, the actor chooses a ticket and their choice is processed at the Ticket Desk, the Travel Agent displays various packages to the user and buys the ticket based on what the user picks, or an Algorithm checks the fluctuating prices of tickets and notifies the user when the price is at its lowest, and the user can then buy that ticket. There are two error scenarios we could think of. There could be a technical error and the tickets aren’t displayed properly, such as displaying incorrect prices, tickets not appearing when they should etc., or the IT system could process an incorrect, different ticket from the one the actor chose.

**Slide 9**

This is the use case ‘Book External Services (Car Hire/Accommodation)’. It is provided by an external actor but is part of the IT system. It has a single actor, External services. These services can choose whether or not they want to be affiliated with the IT system and whether or not they want the airline to promote their services. The airline would charge a commission for this. There is a single entry condition for this use case. The Passenger, or the Travel Agent, Ticket Desk, or Price Comparison Website on behalf of the Passenger, chooses that they would like to purchase external services along with their ticket. There are two exit conditions to this use case, which are that the actor chooses what external service or services they want, or the actor chooses that they do not want external services. Under normal circumstances, the actor is shown a range of services to pick from, they choose which external services they want and the price is displayed along with the price of the ticket. There is one Error scenario, which is that there may be a technical error and the services are displayed incorrectly or the accumulated price is incorrect.

**Slide 10**

The use case ‘Make Payment’ describes what happens after the Passenger has chosen which services and ticket they would like to purchase, or after these have been chosen on their behalf by the Ticket Desk, the Travel Agent, or an Algorithm from a ticket comparison website. In this use case, the Passenger is asked to pay for the ticket and services if they chose any, which includes asking for credit card details. This use case has a single participating actor, the Passenger, and a single entry condition, which is that a ticket and possibly external services have been chosen. There is also a single exit condition, which is that the Payment details are entered (second possible exit condition that the Passenger decides to cancel?). Under Normal circumstances, the payment details are entered correctly. There is an error scenario, where the payment details are not entered correctly or the IT system does not recognise the payment details as valid.

**Slide 11**

The use case ‘Verify Payment’ describes what happens once payment details have been entered by a Passenger. Here, the bank associated with the payment details the passenger gave validates the payment details, and the money is extracted from the Passenger’s account. A confirmation of payment is then sent. This use case has a single participating actor, the Bank. There is one entry condition, which is that the passenger has entered payment details, and one exit condition, which is that the Bank confirms the details and processes the order. Under normal circumstances, the Bank confirms the details and processes the order successfully. Possible Error scenarios include when money is not taken from the account, or the details can’t be verified. Another error scenario is that the Passenger has insufficient funds.

**Slide 12**

This is the use case ‘Check In’. Once the option to Check In becomes available, the Passenger may Check In through the airline app /website or at the ticket desk. Details are taken, such as the Passenger’s passport number, and once these have been confirmed by the airline IT system, a boarding pass is issued. There are three participating actors in this use case, the Passenger, the Travel Agent, and the Airline IT system (Ticket Desk?). The entry condition of this use case is that a ticket for a flight must have been purchased by either the Passenger, or by the Travel Agent (or Ticket Desk?) on behalf of the Passenger. There are two exit conditions. The ticket order is found and the boarding pass is issued, or the ticket order is not found and the boarding pass is not issued. Under normal circumstances, the Passenger checks in, is assigned a seat number and is given other information on the boarding pass. There are four error scenarios to this use case. These are that the ticket order cannot be found, the flight is overbooked, the flight is cancelled, or the Passenger checks in late.

**Slide 13**

This is the use case ‘Issue Boarding Pass’. This describes what happens after the Passenger has checked in successfully. A boarding pass is issued to the passenger by the Airline, confirming details such as flight time, boarding time, gate number and seat number. An email of the boarding pass is sent to the customer and it can also be found on the website/app when the Passenger logs in. There is a single participating actor in this use case, that is the Airline. The entry condition is that the passenger must have successfully checked in with a purchased ticket, and the exit condition is that the boarding pass is issued. Under normal circumstances, the Passenger is issued the boarding pass with information on their flight. There are three error scenarios to this, which are that the flight may be overbooked, the flight is cancelled, or the seat they are assigned is double booked.

**Slide 14**

This is the use case ‘Decide Cost’. This is where an algorithm collects information on the passenger and creates a profile based on this information. It also gathers information, such as time of year, how busy certain flight routes are, and so on. Based on this information, a price for the ticket is decided. There are two participating actors in this use case, the Airline, and the Algorithm used to collect information and decide cost. There is one entry condition, and that is that a ticket must have been chosen by the Passenger or on behalf of the Passenger (price decided before or after a ticket is chosen?). The exit condition is that the cost is decided and displayed to the customer. Under normal circumstances, the cost is decided based on the information collected by the Algorithm, and the cost is displayed to the customer. One Error scenario to this is that there may be an IT error and the cost cannot be calculated or displayed.

(Notes and thoughts: Slide 6 may have too much to do with order of actions, which is more associated with activity diagrams?)