

UML Report

Group 14

Adam Mulvihill

Caoilfhionn Ní Dheoráin

Emmet Morrin

Emer Murphy

Background Research:

We started off with searching general information about Ryanair and flight booking systems. We found information from websites such as nbaa.org (National Business Aviation Association), ryanair.com, and articles explaining how ticket prices were calculated. We also found information about how covid had an impact on these systems.

These links and pieces of information were shared amongst the group.

Fortunately for our domain, Ryanair, we had prior experience with how the booking system of flights works.

We were able to use these experiences to shape our UML diagram from the viewpoint of a customer, and our background research to shape our diagram from the position of ryanair itself.

This allowed us to make a UML diagram which encompassed the whole system.

How we undertook the task:

We used lucid charts for our UML and activity diagrams. It was easy to use and allowed us to share and edit the diagrams we were working on simultaneously.

Some files that we had to share with each other were done over google docs.

The majority of our communication with this project happened over discord and the scheduled collaborate sessions.

We split the project into multiple parts, and each person did different pieces of the project.

We had meetings where we would discuss what was to be done and who was to do it outside of the scheduled team meetings.

Links:

- nbaa.org/flight-department-administration/sms
- nbaa.org/aircraft-operations/scheduling
- lovin.ie/travel-food/21-facts-about-ryanair-that-will-blow-your-mind
- youtu.be/H_akzwzghWQ
- campsystems.com/Product-Services
- <https://www.ryanair.com/content/dam/ryanair/help-centre-pdfs/eu261-.pdf>
- ryanair.com/gb/en/useful-info/help-centre/fees
- en.wikipedia.org/wiki/Air_traffic_control
- cnbc.com/2018/08/03/how-do-airlines-price-seat-tickets.html
- sheffield.com/articles/how-air-traffic-control-systems-work
- illum.in.usc.edu/the-algorithm-behind-plane-ticket-prices-and-how-to-get-the-best-deal

Ryanair Ticket buying system and ethics

It's important that our information system considers ethics because Ryanair is a huge company; its scale and size makes their decisions very important. Ryanair's choices can have huge effects on other people and the environment. Ryanair has a lot of influence on the society that we live in and the planet that we live on. It also creates a good image for the company and improves the service.

Our information system for Ryanair uses big data to develop a deep understanding of the customer. Our information system collects, accesses and analyzes the customers personal data. The ethical questions surrounding the use of people's data are difficult to answer. For Ryanair to offer a one-to-one personalized service for the individual Ryanair must compromise on privacy in some way. Customers give access to their data when they click and agree to the terms and conditions. Our information System is going to be transparent with the customer and let them know what their data will be used for.

Our information system will make purchasing flights easier so the impact on the environment from flights will increase. Further along in our design of the information system, we could add an offset system, that allows customers to pay a little bit extra for their flight ticket so the carbon dioxide from the flight will be offset. There are positive and negative effects of tourism, our information system could have some resources to educate customers on eco-tourism.

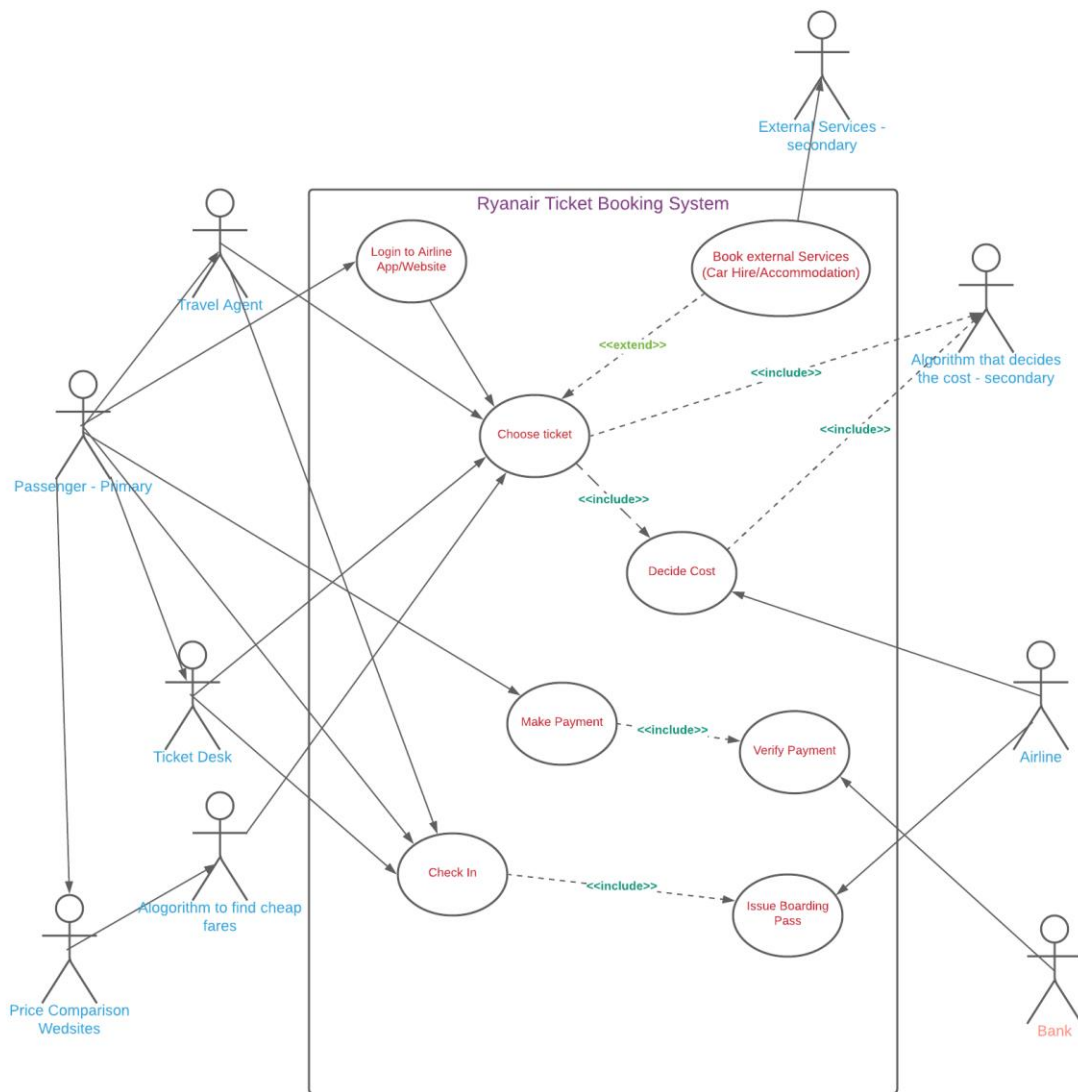
Table of Actors

| Actor | Role |
|-------------------------------|---|
| Passenger | An individual who will, purchase a ticket |
| Travel Agents | A company that provides travel services to the general public on behalf of travel suppliers |
| Price Comparison Websites | E.g Skyscanner, Kayak |
| Ticket Desk | |
| External Services | E.g Car Hire /Accommodation |
| Algorithm that decides cost | Profiles passengers and monitors nature of travel |
| Airline | E.g Ryanair |
| Algorithm to find cheap fares | Monitors price fluctuation |
| Bank | Deals with Payments |

Use-Case/Actor Table

| Use Case Name | Participating Actors |
|------------------------------|------------------------------------|
| Login to Airline App/Website | Passenger/Travel Agent |
| Check-in | Passenger/Travel Agent/Ticket Desk |
| Make Payment | Passenger |
| Issuing Boarding Pass | IT System/Airline |
| Decide Cost | Algorithm that decides cost |
| Book External Services | External Services |
| Verify Payment | Bank |
| Choose Ticket | Passenger |

UML Case Diagram



Use-Case Text Description for Normal and Error Scenario

| | |
|-----------------------------------|---|
| Use-Case Name | Login to Airline App/Website |
| Participating Actor/Actors | Passenger |
| Precondition | <ul style="list-style-type: none"> -The individual must have created an account -Sign in is required to access flight information -Access to internet |
| Exit Condition | <ul style="list-style-type: none"> -Individual doesn't buy a ticket -Individual chooses not to create an account -Individual does choose to purchase a ticket |
| Normal Scenario | <ul style="list-style-type: none"> -If an individual doesn't have an account they must create one -Individual uses their username and password to log in -Ticket information is provided -User has the ability to navigate through the site |
| Error scenario | <ul style="list-style-type: none"> -The IT system doesn't recognise the username and password and denies the individual access to their account -Internet access fails |
| Description | This use case 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 they need access to this use case. The information the user provides is validated for security purposes |

| | |
|-----------------------------------|--|
| Use-Case Name | Choose Ticket |
| Participating Actor/Actors | <ul style="list-style-type: none"> -Passenger -Travel Agent -Ticket Desk -Algorithm to find cheap fares |
| Entry Condition | Individual/Company must be in some way affiliated with the the App/Website through an account or they may work for the company |
| Exit Condition | <ul style="list-style-type: none"> -Individual/Company chooses which ticket they would like to purchase -Ticket Desk picks a ticket on behalf of the Passenger -Travel Agent picks a ticket on behalf of the Passenger -Individual/Company chooses not to buy a ticket |
| Normal Scenario | <ul style="list-style-type: none"> -Individual/Company chooses the ticket they want and the price is displayed -Individual/Company chooses the ticket they want and it is process at the ticket desk -Algorithms checks the fluctuating prices of tickets and notifies the user when the price is at its lowest, ticket is bought through the price comparison website -Travel Agent displays various packages to the user and based on what one the passenger picks, they buy that ticket |
| Error scenario | <ul style="list-style-type: none"> -Technical error and the tickets aren't displayed properly/ prices are wrong ect. -IT system processes the incorrect ticket |
| Description | This use case describes how a ticket is purchased in the IT system. A ticket can be purchased by 4 of the participating actors. A return or single ticket can be bought. |

| | |
|-----------------------------------|---|
| Use-Case Name | Book External Services (Car Hire/Accommodation) |
| Participating Actor/Actors | External Services |
| Entry Condition | The passenger or 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(s) |
| Exit Condition | -The individual/ company chooses what external service(s) they want -The individual/ company chooses they don't want external service(s) |
| Normal Scenario | The individual/ company is shown a range of services, they choose which external service(s) they want and the price is displayed along with the price of the ticket |
| Error scenario | Technical error and the services are displaying incorrectly or price is being accumulated incorrectly |
| Description | This use case is provided by an external actor yet it is part of the IT system. External Companies 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. |

| | |
|-----------------------------------|--|
| Use-Case Name | Make Payment |
| Participating Actor/Actors | Passenger |
| Entry Condition | A ticket (and possibly external services) are chosen |
| Exit Condition | -Payment details are entered |
| Normal Scenario | Payment details are entered correctly |
| Error scenario | Payment details are entered incorrectly or the IT system doesn't recognise the payment details as valid |
| Description | After the Passenger has chosen which services and/or tickets they would like to purchase, the IT system asks the passenger to pay for these. Details of credit card, ect. are required |

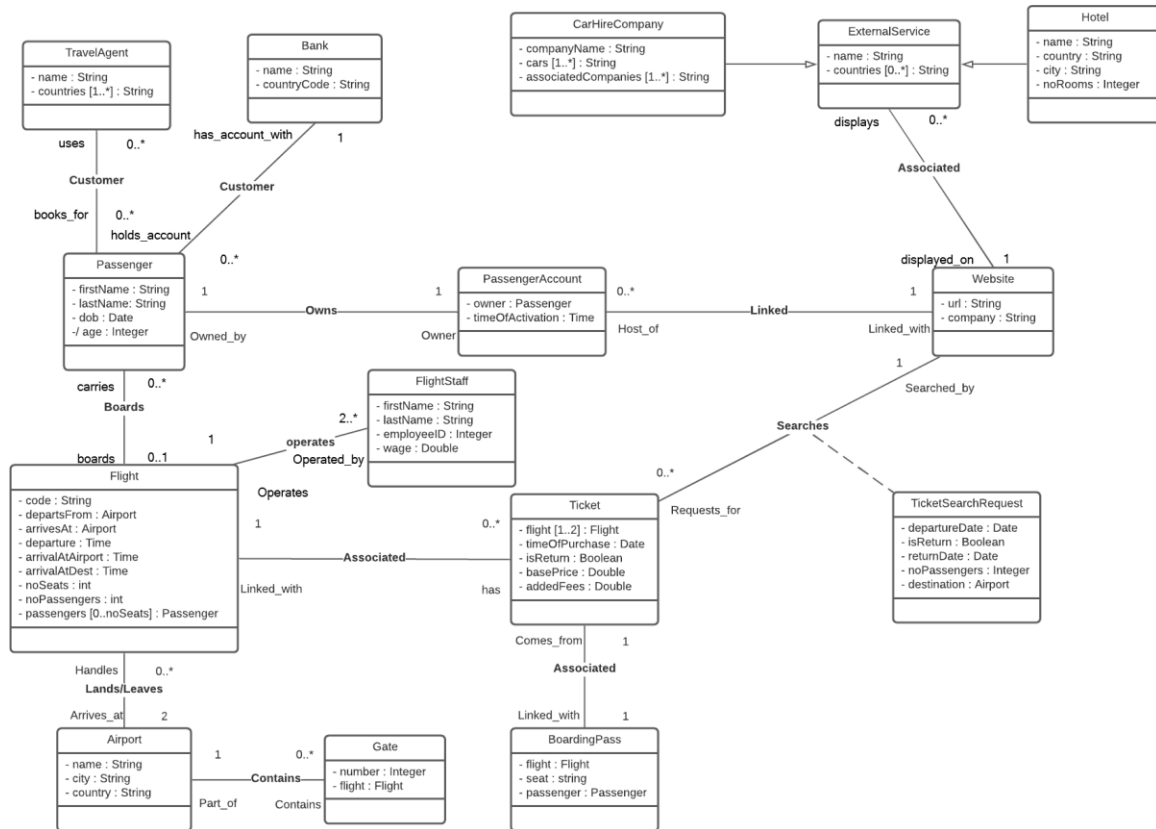
| | |
|-----------------------------------|--|
| Use-Case Name | Verify Payment |
| Participating Actor/Actors | Bank |
| Entry Condition | Payment details must have been entered by the passenger |
| Exit Condition | Payment is verified by the bank and the order is confirmed |
| Normal Scenario | Bank confirms the details and processes the order |
| Error scenario | -Money isn't taken from the account, details can't be verified -Insufficient funds |
| Description | Once payment details are entered, the bank validates these to continue with the process. The money is then extracted from the account and a confirmation of payment is sent. |

| | |
|-----------------------------------|--|
| Use-Case Name | Check In |
| Participating Actor/Actors | Passenger/Travel Agent/Airline IT System |
| Entry Condition | A ticket for a flight must have been purchased |
| Exit Condition | -Ticket order found and boarding ticket issued -Order not found and boarding pass not issued |
| Normal Scenario | Individual/company checks in, is assigned a seat number and other information on the boarding pass |
| Error scenario | -Ticket order can't be found -Flight is overbooked -Flight is cancelled -Passenger checks in late |
| Description | When the option becomes available, the passenger may check in through the airline app/ website or at the ticket desk. Details are taken(passport number, ect.) and once these have been confirmed by the airline IT System, a boarding pass is issued. |

| | |
|-----------------------------------|--|
| Use-Case Name | Issue Boarding Pass |
| Participating Actor/Actors | Airline |
| Entry Condition | Individual/company must have checked in with a purchased ticket |
| Exit Condition | Boarding pass issued with seat number ect. |
| Normal Scenario | Individual assigned a seat number and boarding pass |
| Error scenario | -Flight is overbooked -Flight is cancelled -Seats double booked |
| Description | Once the passenger has checked in successfully, a boarding pass is issued by the Airline confirming details such as the 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 they login |

| | |
|-----------------------------------|--|
| Use-Case Name | Decide Cost |
| Participating Actor/Actors | -Airline -Algorithm that decides the cost |
| Entry Condition | A ticket must have been chosen |
| Exit Condition | Cost is decided and displayed to the customer |
| Normal Scenario | The cost is picked based on the algorithm and displayed to the customer |
| Error scenario | IT error and cost can't be calculated or displayed |
| Description | The algorithm collects information on the passenger. It creates a profile based on this and figures out which category of passenger they belong to. It also takes other factors into account like the time of year of the flight, how many times the passenger has visited the website, ect. Based on all the information a price is calculated. |

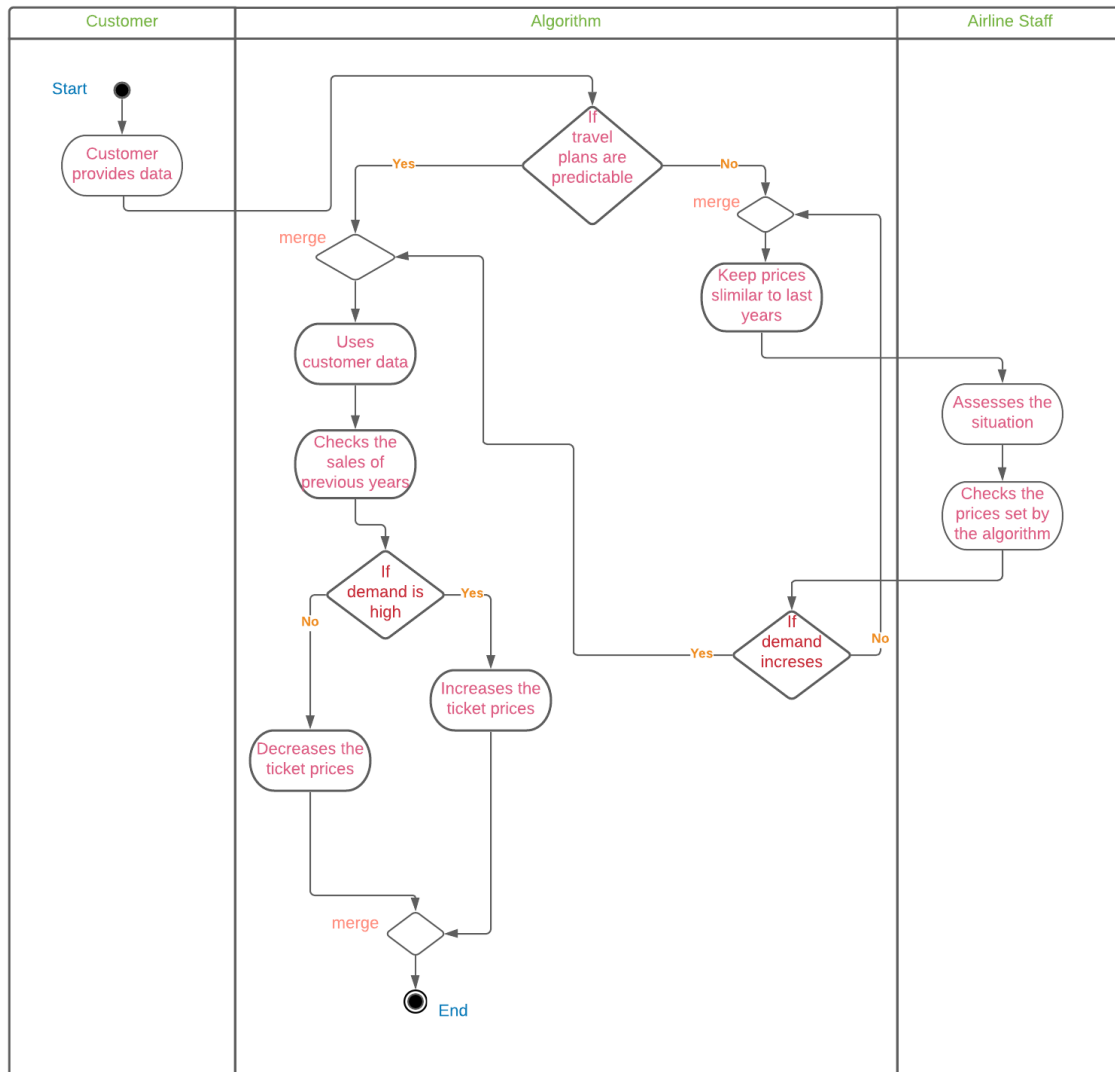
UML Class Diagram



A few small design decisions were made while this diagram was being made. Passenger was made into its own class, such that information about passengers could be stored. The class TicketSearchRequest was made to have the information about what people search for in one place, allowing it to later be stored more easily for the ticket pricing algorithm. Since Hotel and CarHireCompany both had attributes in common, they were both made subclasses of a class ExternalService. These may both later have functionality in common, which could also be put into the class ExternalService instead of being put into both classes separately. Lastly, the Airport class was added, which holds information about Airports such as their name and where they are. This would make it easier for relevant flights and their corresponding tickets to be found when a passenger searches for a ticket.

Activity Diagram - Decide Cost

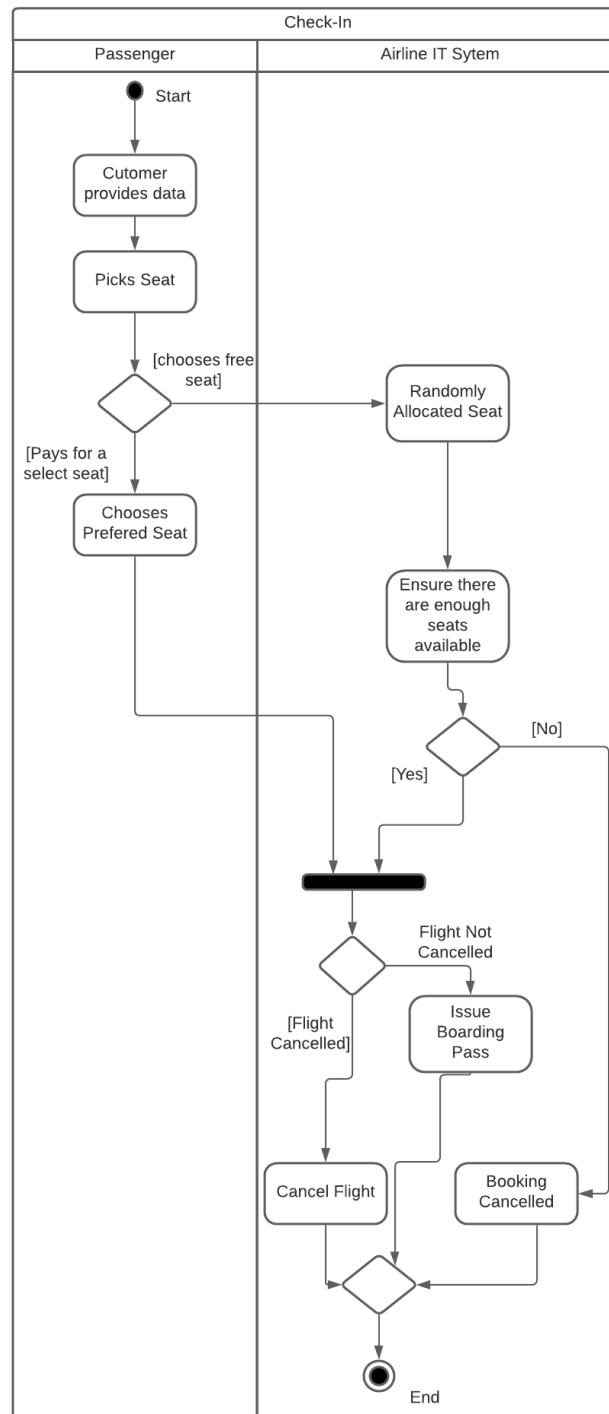
Activity Diagram for the Algorithm That Decides Cost



In our information system, the algorithm that decides the cost of the tickets is designed so that planes are full, but seats aren't sold too quickly because the airline doesn't want to miss out on selling high-priced last-minute tickets. The activity diagram starts when the customer provides data, this is taken when they use the website to book flights. With big data, airlines can predict what the year is going to be like, which seasons will be busier etc. However, this is not the case if there is an unpredictable event that causes a sharp decrease in demand, for example the coronavirus pandemic. If the travel plans of passengers aren't predictable the algorithm keeps the prices similar to the previous year. During unpredictable times when the demand for flights is down, passengers are flying because they have to fly instead of flying because the flight tickets are cheap so decreasing the flight tickets won't increase demand. These are strange, unpredictable times so our activity diagram has a staff member assess the situation and check that the computer is pricing correctly for the circumstance. If the travel starts to increase and demand becomes predictable, the algorithm uses the customer's data and data from the sales of previous years. Airlines don't want to have a big difference in demand from season to season because they have a certain number of staff

and a certain number of planes that they can't change for season to season. If the demand is high the algorithm increases the price of the tickets to make more money else if demand is low the algorithm decreases the ticket prices to increase demand. The algorithm then ends.

Activity Diagram - Check-In



This diagram explains the activity that happens when a passenger checks-in their ticket. When checking in, certain information is provided by the passenger(passport number, ect.). After this the customer can then decide whether they want to be randomly allocated a seat or to pay for a selected one. If they choose to be randomly allocated a seat, the process is a

little more complex. The seats are given to the paying passengers first, and out of the empty seats that are left, one assigned to the other passengers. That is if there are more seats available. If there are no seats left the airline will either cancel the booking or they may assign you with a temporary boarding pass and when you arrive at the gate the employee will check if a seat has become available(due to other passengers not showing up ect.) or they may assign you to the next flight. We choose not to include this case in the activity diagram as it was not part of the IT system and deals more with the employees of Ryanair. Once passengers have been assigned seats, they are issued a boarding pass. As long as the flight is still going ahead, this brings us to the end of the activity diagram.

Listing of Who Did What

| | |
|--|--|
| Description of Background Research | Emmet Morrin |
| UML Case Diagram | Caoilfhionn Ní Dheoráin |
| Use Case Text Description Normal/Error Scenarios | Caoilfhionn Ní Dheoráin |
| Presentation Slides | Caoilfhionn Ní Dheoráin |
| Presentation Script | Adam Mulvihill |
| Presenter | Emmet Morrin |
| Strengths and Weaknesses | Caoilfhionn Ní Dheoráin and Adam Mulvihill |
| UML Activity Diagram - Check In | Caoilfhionn Ní Dheoráin |
| UML Activity Diagram - Check In Description | Caoilfhionn Ní Dheoráin |
| UML Activity Diagram - Algorithm | Emer Murphy |
| Ethics Canvas | Emer Murphy |
| UML Class Diagram | Adam Mulvihill |
| Design Decisions | Adam Mulvihill |

Strengths and weaknesses

A strength of this design is that booking a flight is easy and relatively quick. A customer has many different options for buying a ticket and can choose whatever one is appropriate for their situation. It's also quite a quick process for the passenger. The passenger doesn't have to communicate with the system too much. They just need to pick a ticket, pay for it and check in. It's not a lengthy process. The actions they have to take are executed quickly. The verification of the payment and the check in happen almost instantly, meaning that it is accessible and almost effortless to purchase from the Ryanair booking system. We have also thoroughly described the UML design through Class and Activity diagrams. These display exactly when things take place and what kind of attributes and operations belong to them. After we read our UML report, we as group members were all sure that we clearly understood the Ryanair Booking System.

One of the main weaknesses of our design is that there is nothing in place in the system if the Passenger wishes to cancel or change flights once the ticket has been purchased. From our research, we concluded it's actually an extremely difficult process and can almost never be completely cancelled. For example, a passenger will always encounter some sort of fee for choosing to cancel or change the booking. Perhaps we could have explained this process in our design, but as a group we decided it would have taken from the simplicity of our design, one of the major strengths of our system. Another possible weakness is that a passenger cannot buy tickets from the site themselves without an account. This could potentially limit the amount of customers the airline gets. However, from the research we have undertaken, it is quite necessary for the airline to ensure customers have accounts in order to profile them and use the information within the algorithms.