Group 21 - Just Eat Delivery App

Student Name	Student Number
Anastasiya Bogoslovskaya	**20332234**
Abdelaziz Abushark	**20332134**
Ali Al Ani	**19314029**
Saif Ali	**20333159**
Noel Austin	**20333159**
Adam Beatty	**20332203**

Contents

A. Background Research

We were assigned trade unions as our domain to work on. But as a group we decided it would be better if we opted for a more modern domain that we could relate to and what caught our eyes was the delivery systems that have become a part of our everyday lives. We picked the biggest platform used by the majority of the group which was Just Eat. They have been around the longest and focus more on restaurant descriptions and ratings from customers creating a reliable network their user's trust.

To communicate amongst the group, we opted for WhatsApp as our main source as everyone was on the platform and the notifications can be sent, received and accessed from your phone. It also has a call feature and sending voice messages which we thought would help if things needed explanation to make sure everyone understood the task and what we all had to do. We created a shared google docs which we could all access simultaneously and edit. There we put in a list of everything that needed to be done and who was responsible for each section. We could then see how everyone was progressing and proofread and help if needed. To contact each other we used collaborate ultra for our timetabled slot and we also met up on zoom and in-person outside the assigned hours to make sure the work was getting done in a timely manner.

For the charts we again choose a digital approach due to the nature of being able to edit in real time and everyone being able to access it simultaneously regardless of where they were which only aided our collaboration and joint efforts. We found lucid charts easy to use as a group as it had all the features included and easy to access and make the various diagrams.

To research the domain, we thought it would be best to test it out in the real world so whoever didn't have the app downloaded it. We explored through it and saw the different features and how it all worked. We thought this would help in creating the diagrams as we had direct contact with the system. While consulting our notes we began working on the system adding the main features we thought were most applicable as it is a very big system with various features to make the user experience as pleasant and easy as possible.

We examined various examples from other delivery services online and tried to find common trends and themes that fit in with the lecture slides and material provided to find the best Use Case diagram names. This research process helped us see a clearer picture of the main parties and processes involved in the system. The Use Cases were discussed collectively and then after the main actors and 8 ovals were decided it was finished off and finalised. Having the Use Case diagrams completed helped with creating the Class Diagram as it was based on one another. Once again being done on lucid charts as we had previously used it

and thought it worked great, so we didn't want to change something that was working. The 15 most relevant Class diagrams were then chosen by the group and drawn up.

After completing them we used the same resources to complete the Activity diagrams. We also had a group working on the ethics canvas.
When approaching the ethics canvas, we had to take into consideration every possible way our system affects people, businesses and communities. We had to take a step back and consider these effects on a wider scale. We found the most concerning ethical issue to be the storing and protection of personal data and that's why canvas is more centred around that.

B. Ethic Canvas

Just Eat is an app that doesn't target a core demographic. We explored how this app would affect individuals and groups and found. We found drawing up this ethics canvas very beneficial as it helped us understand exactly how Just Eat affects the world and who is likely to benefit from it but also helped identify who it will negatively affect.

Here is a textual description of our ethics canvas:

Individuals affected:

People of all ages use this app to order food. However, we thought that it's possible older people who are less in tune with technology would be less likely to use the app. Restaurants and takeaways would make use of this app and therefore their employees would be affected.

Behaviour:

Staff will now spend less time taking orders on the phone which means they will spend their time more efficiently. The app is also more convenient for customers as generally people will prefer to order food on an app rather than on a phone, however we did feel it would make people more reliant on their phone.

Relations:

Just Eat will result in less live interaction between the business and the customers as the whole ordering process will be carried out online, the app may also result in people staying in for food more rather than going out to a restaurant as ordering a takeaway is now even easier.

What can we do:

First of all, we need to ensure the app is well trusted and we need to use and store the data in a safe and ethical manner (i.e., use it for the purpose given and nothing more). This will be helped with strong cyber security, as it will ensure the data is kept safe. The app also represents the restaurants we are delivering for; therefore, we must provide a good service as it will reflect poorly on the restaurants if we do not.

Worldviews:

May be an issue with the app and the use of location that comes with that. Secondly the app may result in an increase in the consumption of fast food as it is now easier to get, this could lead to public health problems in the future.

Group conflicts:

competition between restaurants will increase as the app will provide a wider range of options for the consumer.

Groups affected:

Small businesses can benefit the most from this system. The exposure and accessibility they can potentially receive from this app can greatly improve their business. It's possible that restaurants who don't offer any takeaway service will struggle with not being able to

make use of the Just Eat app, with many people preferring to stay home and order a takeaway instead. This is especially relevant during the pandemic. A company that builds phone applications may be consulted to create/improve the layout of the Just Eat app.

Product or service failure:

Personal data such as names, home addresses, phone numbers and email addresses are all at risk of being leaked in the event of a data breach. If the Just Eat does not function correctly it is possible that food orders can be lost or entered incorrectly. Also, errors in payment can occur if the app doesn't work as intended.

Problematic use of resources:

The user's personal data is saved on the app when they create an account. This includes information like their name, address, phone number and email address. With the increased convenience of ordering fast food that Just Eat provides, it may result in more people becoming overweight as this type of food tends to be less healthy than a home cooked meal.

The visual Ethics Canvas Attached below:

Ethics Canvas, Group: 21 Title: Just Eat Date: 12/10/2021

Individuals Affected:

- Used by men and women of all ages.
- Likely used less by the elderly who are less in touch with technology.
- Restaurants, takeaways and their employees are affected.

Behavior:

- Staff will spend less time taking orders by phone.
- More convenient for customers who rather order by app then on phone.
- Results in people becoming more reliant on there phone.

Relations:

- Less live interaction between the business and the customer as the whole ordering process is now done online.
- May result in people staying in more as food can be delivered to your door very easily.

What can we do?:

- The biggest ethical issue is probably the storing of personal data.
- However, with strong cybersecurity, the user's data can be stored safely and consensually.
- If our system is not designed to work effectively and efficiently, countless businesses that use the app will be very negatively affected.

Worldviews:

- May be issues with the app using your location and personal data.
- Consumption of fast food may increase, as its easier to get.

Group Conflicts:

- Will result in more competition between restaurants as customers will have a greater choice.

Groups affected:

- Small businesses can benefit the most from this system, exposure and accessibility they receive from this app will improve their business.
- It may result in restaurants who do not offer a takeaway service struggling, as people may prefer the option of staying in and ordering a takeaway.
- May need to consult with an app building company to improves the layout etc.

Product or Service Failure:

- Names, phone numbers, email addresses and home addresses are all at risk in the event of a data breach.
- If the app doesn't work properly, food orders can be lost, errors in payment can occur.

Problematic Use of Resources:

- Personal data is saved securely when the user creates an account on the app
- may result in more people being overweight as takeaway food tends to be less healthy than a home cooked meal.

C. Use case diagram

Name:	**Opening of app**
Participating Actor	**Customer**
Entry Condition	**Customer has a smartphone** **Customer has internet**
Exit Condition	**Customer has opened the app**
Normal Scenario	**1) Customer turns on their smartphone** **2) Customer unlocks their smartphone** **3) Customer finds app** **4) Customer clicks and opens app**
Error Scenario	**Customer has no internet** **1) App opens but asks to connect to internet for any information**
Description	**The customer would use a technological device such as a smartphone or computer to access the Just Eat application. From opening the app, the customer can then proceed to choose the food/restaurant he wishes to order/from.**

Name:	**Order Food**
Participating Actor	**Customer**
Entry Condition	**Customer has a smartphone** **Customer has internet** **Customer has opened the app**
Exit Condition	**Customer ordered food**
Normal Scenario	**1) Customer has selected restaurant** **2) Customer has selected food** **3) Customer has selected drink**

Error Scenario	**Customer doesn't know what to order**
Description	**After the customer has opened the Just Eat app successfully as well as choose the restaurant from which they want to order food from, the customer can proceed to pick out the food they wish to order. from the specific restaurant they chose. If the customer decides to choose card as**

Name:	**Pay for Food**
Participating Actor	**Customer**
Entry Condition	**Customer has money**
Exit Condition	**Customer has paid for food.**
Normal Scenario	**1) Customer chooses whether they pay with cash or card** **2) Customer pays with card if they chose card** **3) Customer waits for driver to pay for cash otherwise.**
Error Scenario	**Customer has no money to pay for food.** **Customer's card is invalid** **Customer inserted incorrect bank details.**
Description	**Once the customer has successfully chose restaurant as well as chose the food they want to order it is up to the customer to pay for the food they wish to have delivered. Once the customer is ready to pay for food, they are given two payment options (cash or card) straight after.**

Name:	**Accept Payment**
Participating Actor	**Just Eat**
Entry Condition	**Customer has paid**
Exit Condition	**Just Eat accepts payment/payment method**
Normal Scenario	**1) Customer has paid for food.** **2) Just Eat has accepted their payment**
Error Scenario	**Customer's card had insufficient funds.**
Description	**At this stage of the case, the 'Just eat' ordering system accepts the payment made by the customer for his/her food. Whether the customer has decided to pay cash/ card it is upto the 'Just Eat' app to determine whether or not the form of payment is accepted.**

Name:	**Refund**
Participating Actor	**Restaurant**
Entry Condition	**Restaurant has issue with payment** **Restaurant cannot make food**
Exit Condition	**Just issues refund**
Normal Scenario	**Restaurant cannot make food (insufficient food supply)** **Issues refund through the just eat app back to the customer** **Customer has money returned**
Error Scenario	**1) Restaurant has problems with refunding money**
Description	**In the scenario where the customer declines the order, a refund can be issued to the customer.**

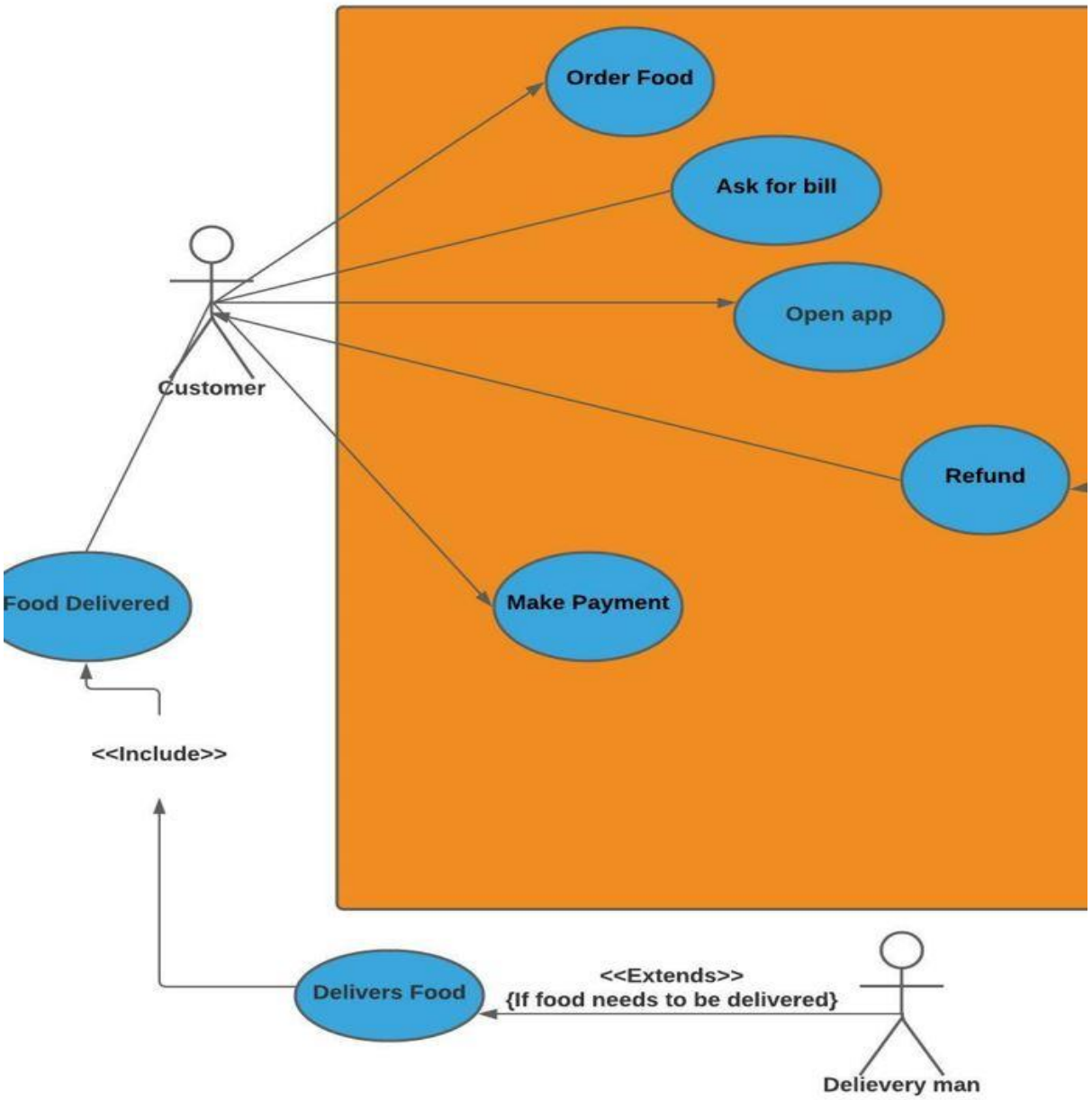
Name:	**Confirm Order**
Participating Actor	**Just Eat**
Entry Condition	**Customer has ordered food** **Restaurant has accepted payment** **Restaurant can make food**
Exit Condition	**Order Confirmed to the customer**
Normal Scenario	**1) Customer has ordered food** **2) Customer has paid for food** **3) Restaurant has accepted payment** **4) Restaurant has given Just Eat the go-ahead for the order** **5) Just Eat confirms the order with the customer**
Error Scenario	**Order confirmation doesn't go through** - **Restaurant system crash** - **Customer loses internet connection** - **Restaurant cannot make food**
Description	**Once the customer has paid for his food as well as Just Eat accepting the customer's form of payment. Just Eat confirms the order with the restaurant and the customer too, letting them know that their food is going to be prepared and delivered.**

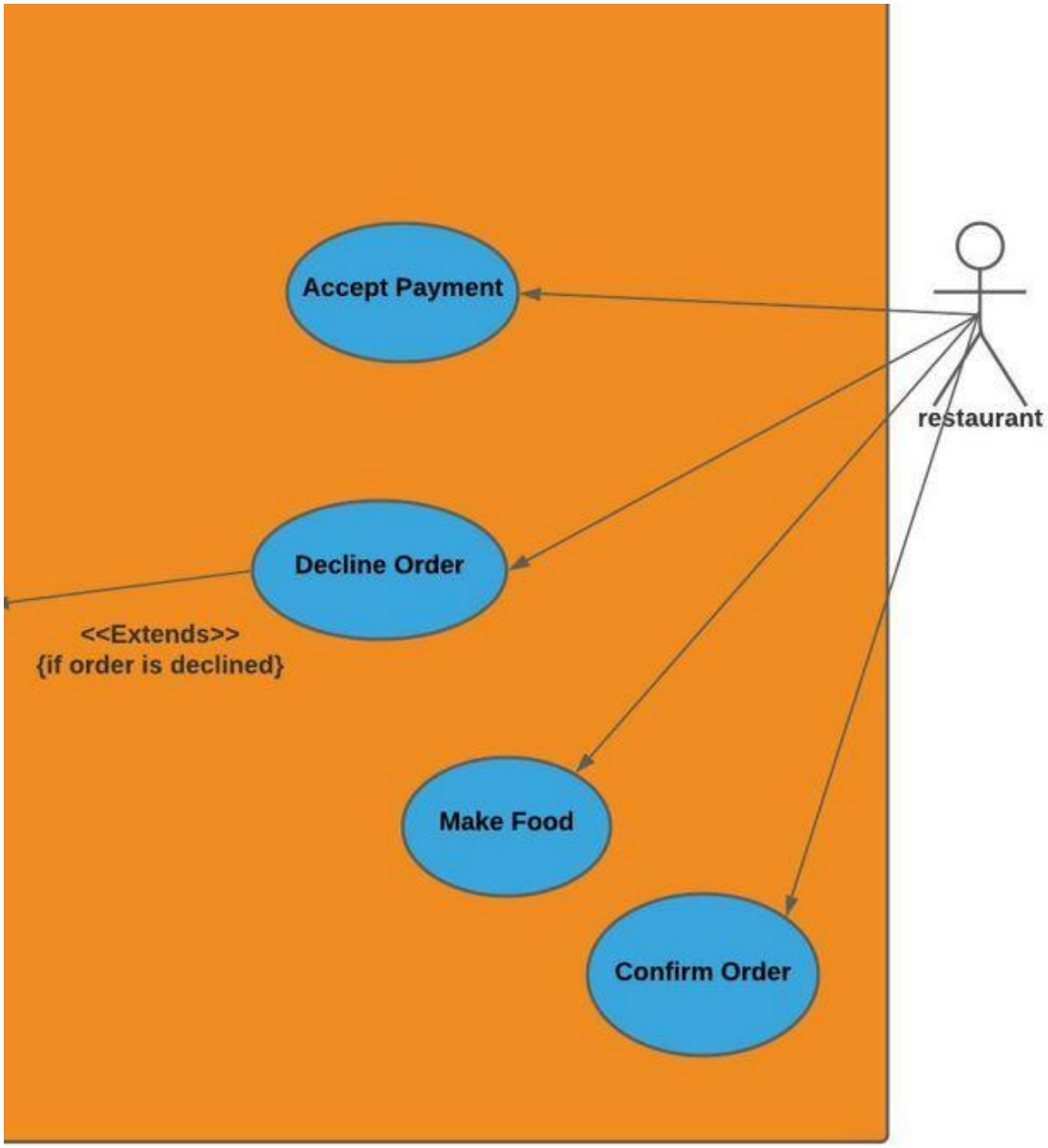
Name:	**Make food**
Participating Actor	**Restaurant**
Entry Condition	**Restaurant has been given order from customer**

	Restaurant has accepted payment from customer **Restaurant has sufficient food and staff to make order**
Exit Condition	**Food is made for delivery**
Normal Scenario	**Restaurant has been given an order of food and drink from a Just Eat customer** **Restaurant begins to make the food from scratch for delivery**
Error Scenario	**Restaurant runs out of ingredients** - **While making food, restaurant cannot continue as they run out of ingredients** **Restaurant cannot make food as they have already closed**
Description	**In the event where the customer has ordered his food and Just eat has successfully processed it. A summary of the order is then provided to the restaurant, where they are then responsible for the making of the food that is requested to them by the customer.**

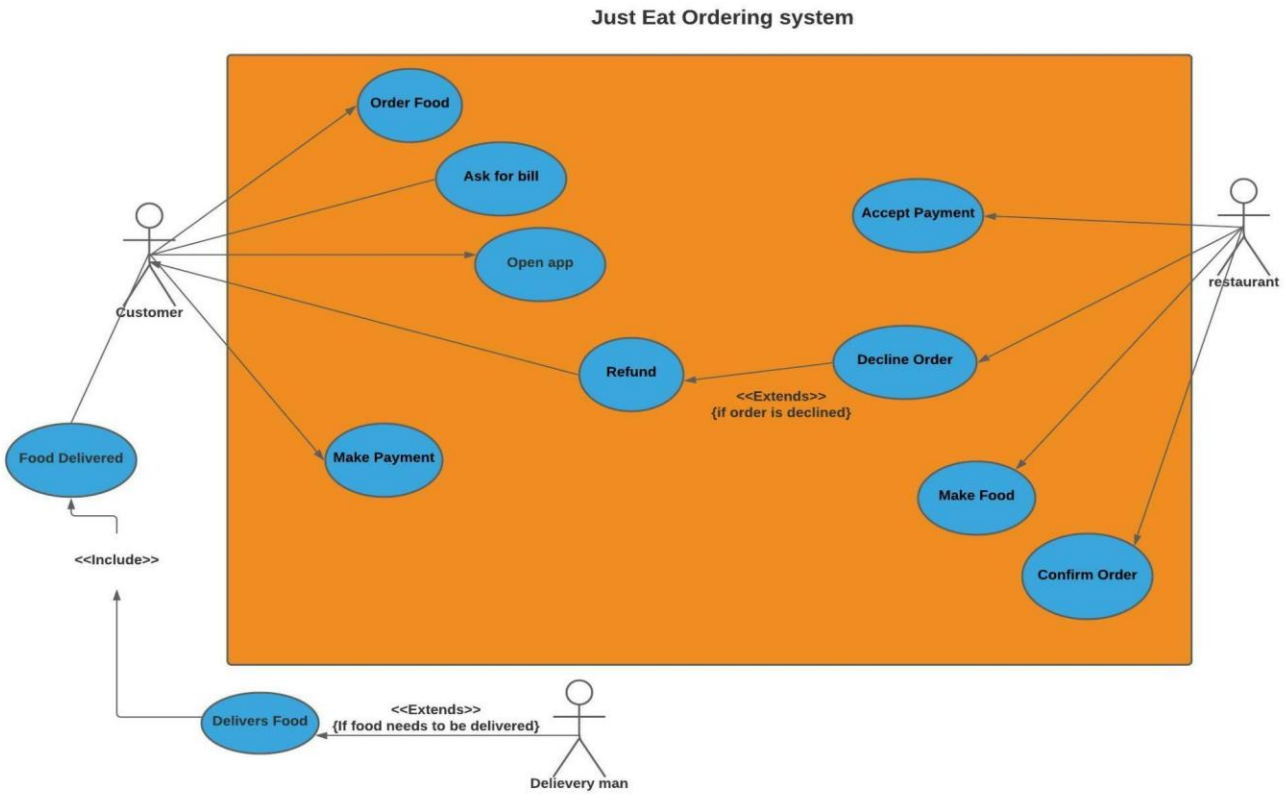
Name:	**Delivers Food**
Participating Actor	**Delivery driver**
Entry Condition	**Food has been prepared by restaurant** **Delivery driver has a car or bike** **Delivery driver has been given prepared food** **Delivery driver knows the address of the customer**
Exit Condition	**Delivery driver delivers the food to the customer**
Normal Scenario	**1) Delivery driver collects food from the restaurant as well as the address of the customer.** **2) Delivery driver enters their car/bike** **3) Delivery driver Drives/cycles to customer address** **4) Gives customer their order** **5) Collects cash if payment method was cash**
Error Scenario	**Delivery driver's car breaks down** **Delivery driver forgets to collect food** **Delivery driver cannot find the address of the customer** **Customer won't open the door for the delivery driver**
Description	**Once the restaurant has completed making the food. They are obligated to pass on the order to the delivery man, and the delivery man must finally send the food to the customer.**

Attached below parts of Use case diagram:


Order Food
Ask for bill
Open app
Refund
Make Payment
Customer
Food Delivered
<<Include>>
Delivers Food
<<Extends>>
{If food needs to be delivered}
Delievery man


Accept Payment
restaurant
Decline Order
<<Extends>>
{if order is declined}
Make Food
Confirm Order

Attached below the final full Use case diagram:



D. Class diagram

When we started doing the class diagrams, we looked back at our use-case diagram and system then we came up with classes to maintain both our use-case and classes diagrams, so we had to decide what are the best classes to include on our class diagram as we want to keep our diagram understandable.

Before we began implementing the class diagram, we did research on how to transition from use-case diagram to class-diagram to make sure of the suitability and accessibility of each field/method in each class. We started the procedure by reviewing and analysing the relationship between all the classes to ensure that it reflected anything relevant to

our project, because we originally planned to tie and connect almost every class to the other. Finally, the only most important connections were filtered out and left in the final version of the diagram.

Down below is the description of how the system works:

The User (customer) enters just eat application

The User Signs up to be member in just eat application

Suggestions windows will pop up that has latest discounts, restaurants, and food suggestions

The User maybe choose between search or previous orders (if they used the app before)

If the user chooses previous order, they will be directed into a new window

If the user chooses to search a search bar will pop up

The User will order their food

Just eat app will give them the cost of the food and the delivery

Just eat app payment tap will pop up and has two options either credit card or cash on delivery

If the User chooses credit card, Just eat up will pop a page to allow user enter their details

If the User chooses Cash on delivery, Just eat app will send a message to delivery man

Just eat app will give the User their confirmation number and estimation time for their ready to be ready

Restaurant prepares the food and notifies Just eat that the order is ready

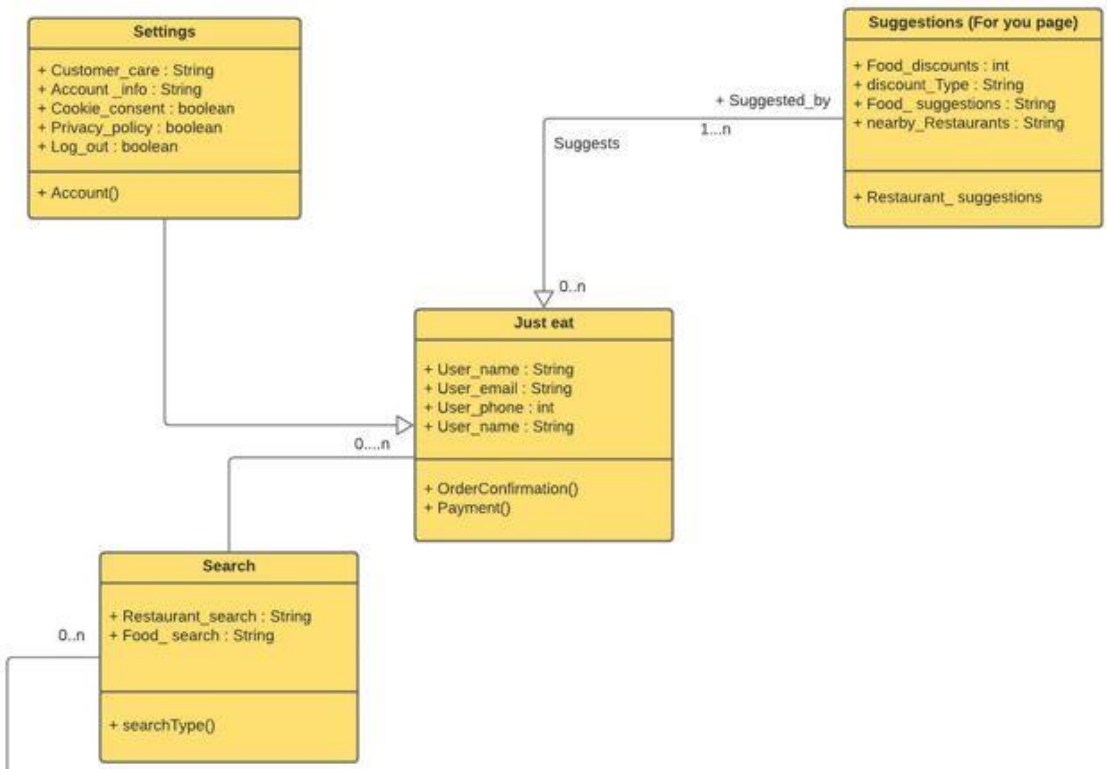
Just eat will notify one of the nearest drivers to collect the food

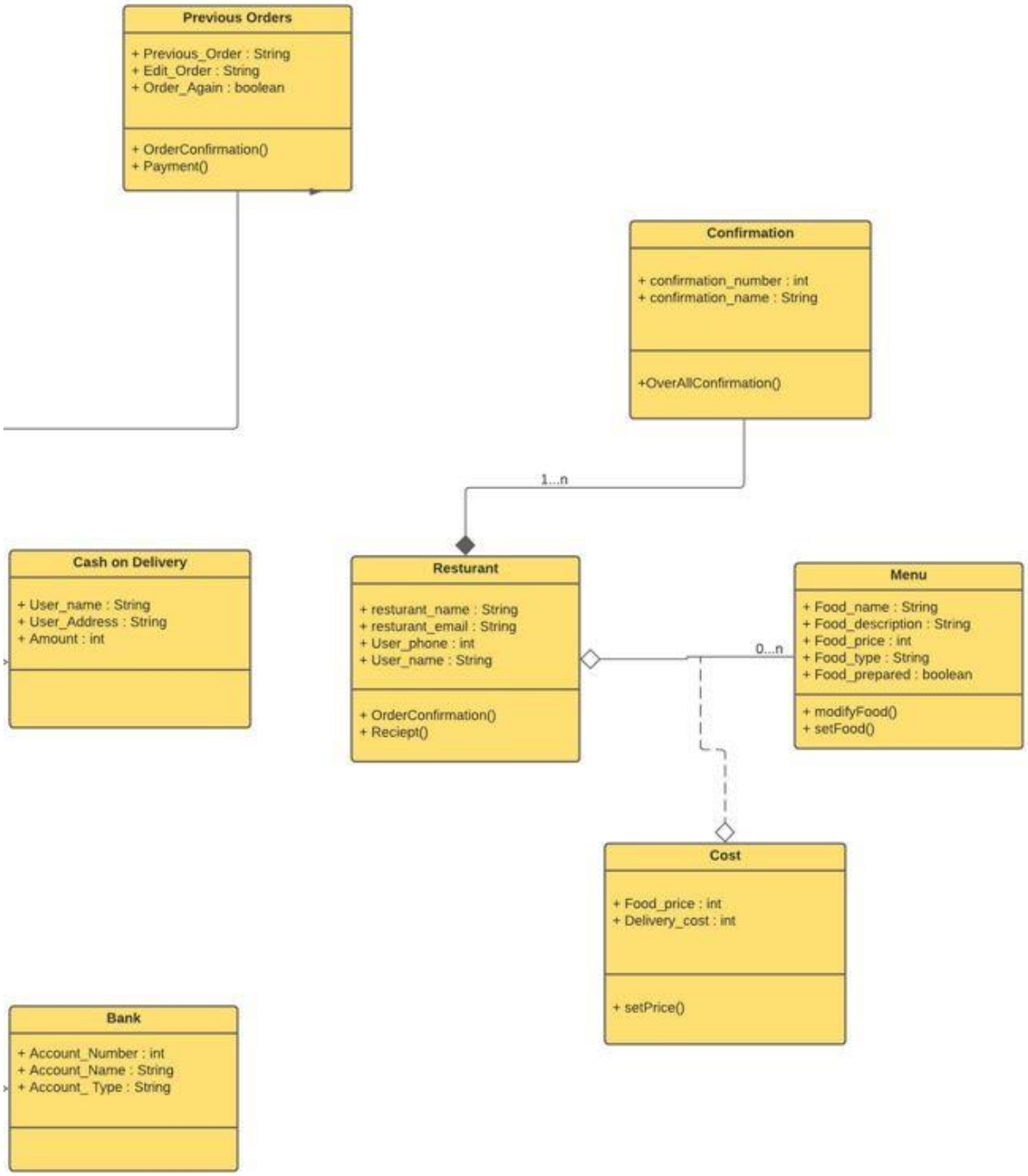
Delivery man will deliver the food to the User

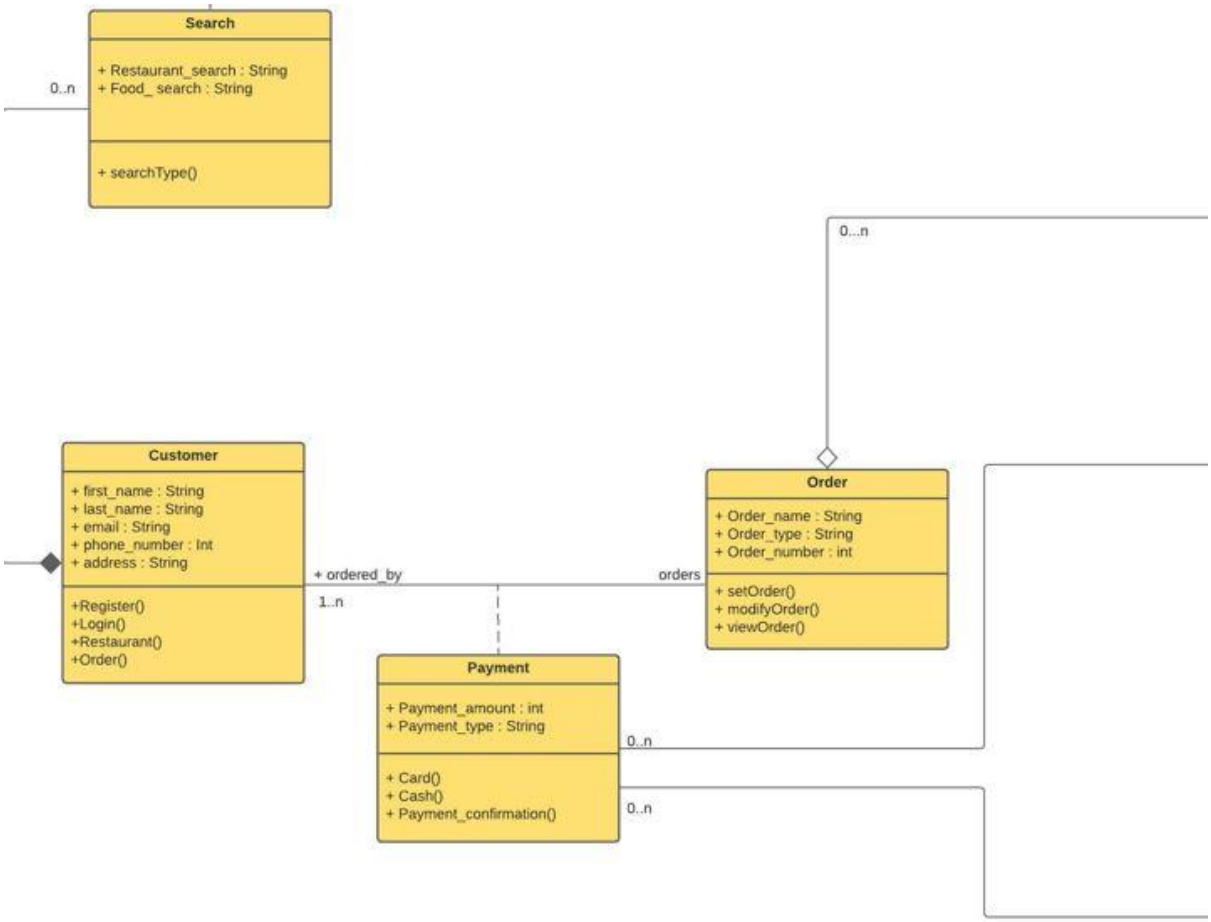
If User chooses cash on delivery, user pays money to delivery man

User rates the services out of 5 on Just eat app

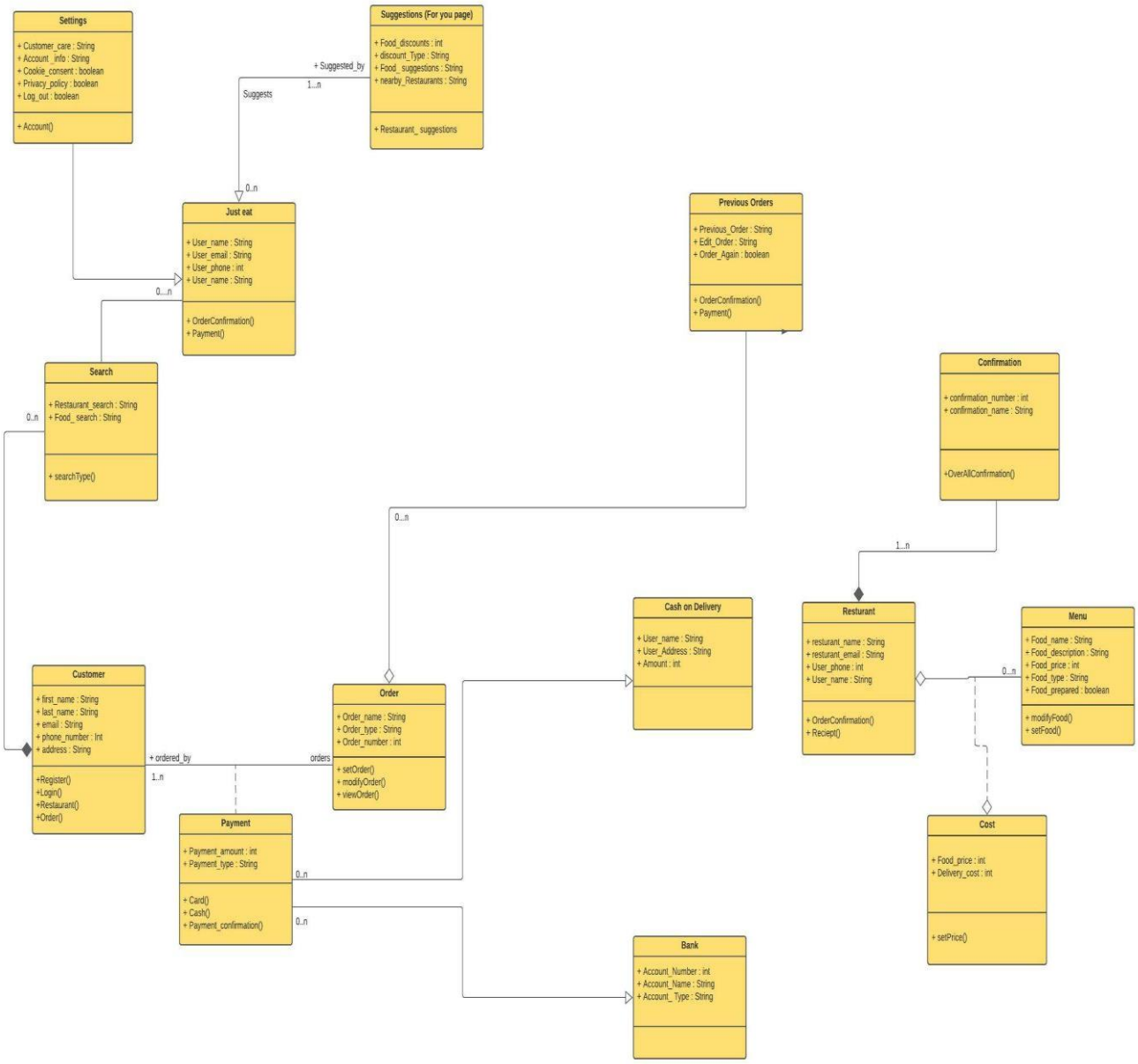
Attached below parts of the Class diagram:




Previous Orders
+ Previous_Order : String
+ Edit_Order : String
+ Order_Again : boolean
+ OrderConfirmation()
+ Payment()
Confirmation
+ confirmation_number : int
+ confirmation_name : String
+OverAllConfirmation()
1...n
Cash on Delivery
+ User_name : String
+ User_Address : String
+ Amount : int
Resturant
+ resturant_name : String
+ resturant_email : String
+ User_phone : int
+ User_name : String
+ OrderConfirmation()
+ Reciept()
0...n
Menu
+ Food_name : String
+ Food_description : String
+ Food_price : int
+ Food_type : String
+ Food_prepared : boolean
+ modifyFood()
+ setFood()
Cost
+ Food_price : int
+ Delivery_cost : int
+ setPrice()
Bank
+ Account_Number : int
+ Account_Name : String
+ Account_ Type : String

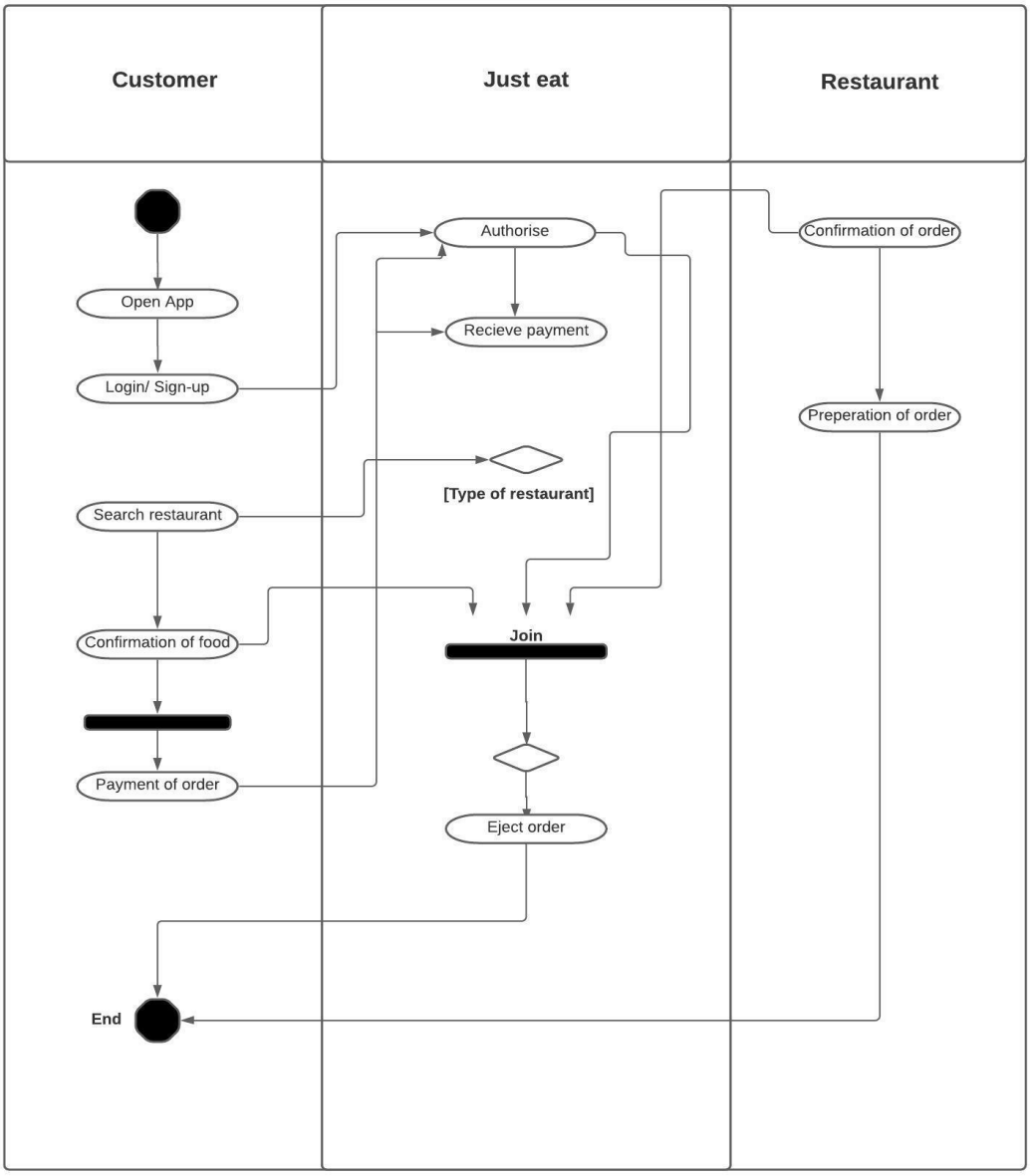

Search
+ Restaurant_search : String
+ Food_ search : String
+ searchType()
0..n
0...n
Customer
+ first_name : String
+ last_name : String
+ email : String
+ phone_number : Int
+ address : String
+Register()
+Login()
+Restaurant()
+Order()
+ ordered_by
1..n
orders
Order
+ Order_name : String
+ Order_type : String
+ Order_number : int
+ setOrder()
+ modifyOrder()
+ viewOrder()
Payment
+ Payment_amount : int
+ Payment_type : String
+ Card()
+ Cash()
+ Payment_confirmation()
0..n
0..n

Attached below the full Class diagram:



E. Activity diagrams

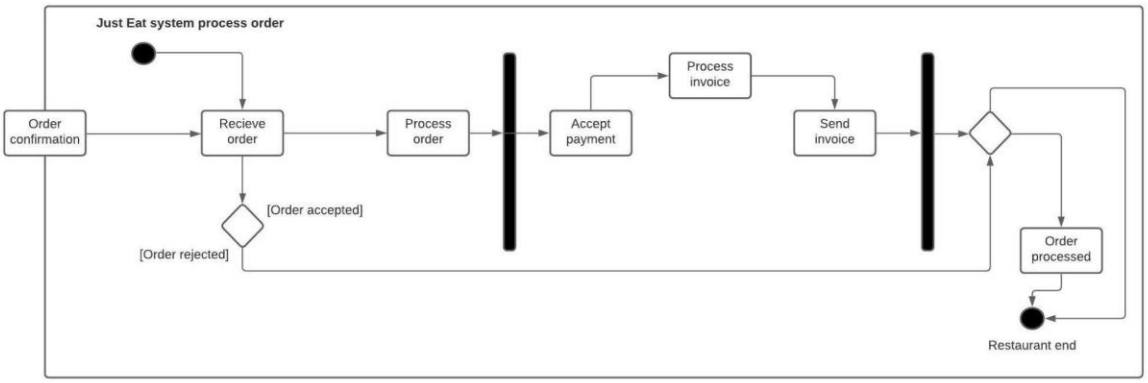
1st Activity diagram:



Activity diagram 1 description:

This diagram focuses on the interaction between 3 parties (Customer, Just eat, Restaurant)as this diagram shows that when the customer aka the user opens the application they are prompted to login to their account or sign up to a new one. The customer then chooses what restaurant they want to eat from and orders their food. The just eat app then authorizes the payment and confirms that it's gone through. The food order is sent to the restaurant to prepare ending this activity diagram.

2nd Activity diagram:



Activity diagram 2 description:

The second activity diagram focuses on the order system. Once the customer has confirmed his/her order. The order system goes through a series of processes. The order is first received and must be processed where it reaches a block of decision. If the order cannot be processed, it will then be rejected, and the confirmation order would then end. If the order is processed payment must be accepted and then the invoice is printed and sent to the restaurant where the order will be processed by hand. This is where the activity diagram ends.

F. Contributions:

Ali Al Ani:

- Made Use case, class and activity diagrams
- Made Use case in-depth explanation
- Created a draft for use, class and activity diagrams
- Created the presentation for our assignment
- Presented the presentation
- Participated in group discussions regarding the project
- Contributed in the description of use case, class and activity diagrams
- Was present throughout all the tutorials and made sure the group is on the same page.

Saif Ali:

- Made Use case, class and activity diagrams
- Made Use case in-depth explanation
- Created a draft for use, class and activity diagrams
- Created the presentation for our assignment
- Presented the presentation
- Participated in group discussions regarding the project
- Contributed to the description of use case, class and activity diagrams

Abdelaziz Abushark:

- Worked on the research of Use case, class and activity diagrams
- Contributed on making Use case, class and activity diagrams
- Created a draft of Use case, class and activity diagrams
- Helped recording for the presentation and contributed on the script for the presentation
- Formatted report by adding the templates, pics, etc.
- Actively Participated in group discussions regarding the project
- Wrote the whole description for class diagram

- Managed to evenly distribute different roles to members regarding project.
- Contributed to the description of use case and activity diagrams

Noel Austin:

- Researched the ethics part of the project
- Contributed on adding bullet points to the ethics canvas
- Created the final ethic canvas diagram
- Made the description for ethics canvas
- Made the description for the 1st activity diagram
- Participated in group discussions regarding the project
- Participated in the script writing process of the presentation

Adam Beatty:

- Researched the ethics part of the project
- Contributed on adding bullet points to the ethics canvas
- Created a draft for ethic canvas and reviewed it
- Participated in group discussions regarding the project
- Participated in the script writing process of the presentation

Anastasiya Bogoslovskaya:

- Background research for the whole project e.g. Just Eat
- Participated in group discussions regarding the project
- Wrote the background research
- Wrote and reviewed the strengths and weaknesses for the project
- Contributed on adding bullet points to the ethics canvas
- Contributed on making Use case diagram
- Reviewed the work and help put finishing touches

G. Strengths and Weaknesses:

A - The actors in the UML diagram:

Type	Strengths	Weaknesses
Customer	**- The system remembers the customer and their favourite restaurants, foods, log in details and card details for easier use of the system repetitively.** **- Easy to read and see the different functions available to do customer** **- Labels are meaningful and names are easy to read and self-explanatory**	**- Lacks the customer's point of view and does not take their needs into consideration**
Just Eat System	**- Helps communicate the functionality of the system** **- Provides structure to the activity** **- Can see the user needs** **- Helps structuring and clarifying procedures**	**- not executable** **- things with similar graphical representation are difficult to distinguish between** **- Difficult to model complex things** **- What happens when errors occur and how will you rectify them**
Delivery Man	**- Simple use cases (delivers food)**	**-No flexibility** **- High level of abstraction** **- Might not be giving accurate times of the delivery taking place because didn't factor in traffic and queues**

Restaurant	**- Provides structure to the activities** **- Shows the different process they are involved in and all the things it needs to complete**	**- Difficult to get an overall overview** **- Very specific function to only represent the functions of the system**

B - General UML diagram

Strengths of the UML diagrams:

- Easy to read and understand
- Easy to follow and self-explanatory as simple language is used
- Communicates the functionality of the system.
- Provides structure to the activity
- See the user needs from the system
- Helps planning and development
- Shows different points of view
- Standardized and can be read by all parties
- Helps in structuring and clarifying procedures
- Labels are meaningful and names are easy to read

Weaknesses of the UML diagrams:

- Not executable
- High level of abstraction
- Lack of the users point of views
- Difficult to model complex things
- Difficult to get an overall overview
- Things with similar graphical representations are difficult to distinguish between

- Very specific function to only represent the functions of the system