



## ABACUS Elaboration Document

Uni Eats

Team Members:



66  
100.  
Janet

Document Date: 28 August 2020

Review mode still on  
when saved as pdf →  
makes it difficult to  
view diagrams + designs

**Commented [Janine1]:**

The Elaboration Document is a tangible outcome of the Elaboration Phase (Analysis and Design) and expands on the initial requirements definition as done in the Requirements Document. In essence, the requirements doc becomes the elaboration doc

No need to retype all the narratives, etc – just copy and paste and make the required changes as recommended after the Requirements document assessment

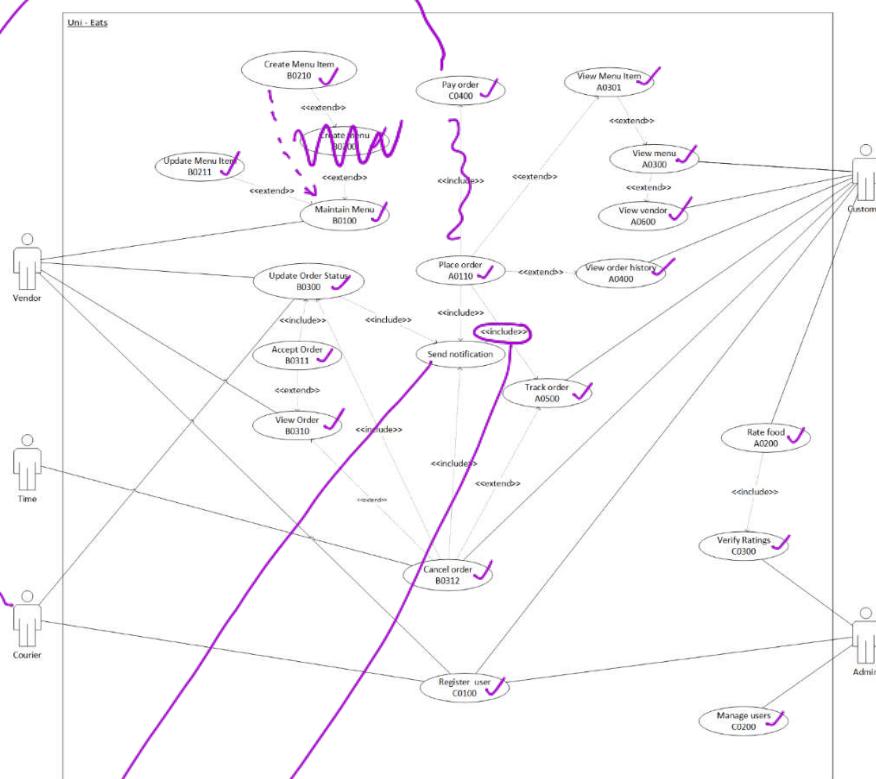
## TABLE OF CONTENTS

---

<b>1</b>	<b>FUNCTIONAL REQUIREMENTS.....</b>	<b>3</b>
1.1	Analysis Use Case Model .....	3
1.2	Use Case Glossary and Responsibilities .....	4
<b>2</b>	<b>UI PROTOTYPES.....</b>	<b>6</b>
2.1	Team UI Guidelines .....	6
2.2	Team Design for.....	8
2.2.1	Login / Logout .....	8
2.2.2	Dashboard / Landing Page / Main Page.....	8
2.3	UI Designs & Updated Analysis Use Case Narratives.....	9
2.3.1	Designed by [REDACTED] .....	9
2.3.2	Designed by [REDACTED] .....	15
2.3.3	Designed by [REDACTED] .....	20
<b>3</b>	<b>DATA REQUIREMENTS .....</b>	<b>26</b>
	Implementation Ready Class Diagram .....	26

# 1 FUNCTIONAL REQUIREMENTS

## 1.1 Analysis Use Case Model



- why did you decide to keep it here? I recommended it to be removed in Req. Doc.

use case ID?

### Commented [Janine2]:

This will be an updated diagram based on feedback from the Requirements document assessment.

### Steps in Constructing the Analysis Use Case Model:

- 1.Identify, define and document new actors – or refine by showing inheritance between actors.
- 2.Identify, define and document new use cases.
- 3.Identify, define and document <<include>> and <<extend>> use cases.
- 4.Update the use case model diagram.

### <<Include>> USE CASES

- 1.A use case whose functionality can be shared by other use cases (common behavior)
- 2.The <<include>> relationship occurs when you have behavior that is common to several use cases
- 3.<<includes>> are used when there are several use cases performing similar functionality
- 4.These are then extracted into an abstract use case, and this abstract use case can be referenced or used by any other use case requiring the functionality (e.g. printing reports).
- 5.Arrow pointing from original to use case it is using

### <<Extends>> USE CASES

- A use case that adds value to another use case
- <<extends>> is used when you want to describe extra functionality that is available when required or behavior that is done only under certain circumstances
- An extension use case can only be invoked by the use case it is extending (but a given use case may have many extends relationships, if very complex)
- Arrow pointing from extension use case to use case it is extending

**VERY NB: Please include Use Case Nr on the use case diagram as well**

## 1.2 Use Case Glossary and Responsibilities

Team Member Responsible: [REDACTED]	
Use Case Id	Use Case Name
A0110	Place Order
B0300	Update Order Status
C0100	Register User
C0200	Manage Users
Queries/Reports	
A0300	View Menu
A0301	View Menu Item

**Commented [NJ((CS3):** Decide who will be taking responsibility for the design and development of which specific use cases.

Each team member must have a minimum of 2 use cases that ADDS/UPDATES/DELETEs from a database table

**Commented [NJ((CS4):** Each team member should have a minimum of 1 report/query

Team Member Responsible: [REDACTED]	
Use Case Id	Use Case Name
B0100	Maintain Menu
B0200	Create Menu
B0210	Create Menu Item
B0211	Update Menu Item
C0400	Pay Order
Queries/Reports	
A0500	Track Order
B0310	View Order

**Commented [NJ((CS5):** Decide who will be taking responsibility for the design and development of which specific use cases.

Each team member must have a minimum of 2 use case that ADDS/UPDATES/DELETEs from a database table

**Commented [NJ((CS6):** Each team member should have a minimum of 1 report/query

Combined these 2, make more sense

**Team Member Responsible:** [REDACTED]

Use Case Id	Use Case Name
A0200	Rate Food
B0311	Accept Order
B0312	Cancel Order
C0300	Verify Ratings
<b>Queries/Reports</b>	
A0400	View Order History
A0600	View Vendor

**Commented [NJ((CS7):** Decide who will be taking responsibility for the design and development of which specific use cases.

Each team member must have a minimum of 2 use case that ADDS/UPDATES/DELETEs from a database table

**Commented [NJ((CS8):** Each team member should have a minimum of 1 report/query

## 2 UI Prototypes

### 2.1 Team UI Guidelines

#### Navigation and control

Navigation is allocated in the header of the screen with tabs that are relevant to the respective user being displayed. The logo is situated in the top left side of the header, the logo will by default navigate the user to the home page. A pop-up menu option is also provided on the top left side of the screen (three bar icon) that will give the user options to view their profile, order history, log out etc.

#### Support for user efficiency

The UI design aims to provide complete efficiency for all users. Input controls such as radio buttons, drop down lists and buttons are implemented to help users complete a task in the minimum time possible. Data sheet views are also provided to enhance a customer-admin query. User efficiency is also supported by a search bar. For example, if a user's role is a customer, they can search for a meal or vendor they seek.

Upon registering, a user can choose what type of role they wish to fulfil (customer, vendor, courier). For each type of user, respective navigation is provided in the header that is customized to the appropriate user.

#### Recognition rather than recall

The website makes use of search boxes located near the top of the screen. The search box allows users to search for their desired food items. A user does not have to remember which vendor offers which product, as users can search for their desired product and the system will return every vendor that offers that product. The website also makes use of menus which will give users more cues as to what types of foods are available. This helps recognition as more cues are provided to help the user recall information from memory.

The consistently placed buttons and screen layout, minimalist features and recognizable icons promote recognition. This consistency ensures that the user only needs to learn how to use the system once and having more features to remember will decrease the user's memorability of using the website.

Uni-Eats user interface is designed in such a way so that it resembles other more recognizable food delivery websites. If our users have had experience with other delivery systems and understand how they work, then it will be easier for users to recognize and understand how the system works. This capitalizes on the user's mental model on how they understand food delivery websites. Familiar interface metaphors such as the cart symbol are utilized, a cart symbol is used to resemble a shopping cart and the user adds items to this cart just which is the same flow of buying from a physical store.

#### Aesthetic design

The Uni-Eats website provides a consistent layout throughout the system. Consistency is achieved by using a fixed colour scheme (the website will use Nelson Mandela University colours as this is their main target domain) and the same web page layout throughout i.e. consistent positioning of elements etc. White space is maximised wherever possible allowing the pages to not look too busy and causing the user to be overwhelmed.

**Commented [NJ((CS9):** As a group decide on a default UI style that your screens will need to adhere to (search for style sheet ideas online)

Think about menu placement, font style, button placement, naming conventions, colour schemes, how error messages are handled

When your design are assessed we will look at the following aspects:

#### **Navigation & Control**

How easy is it for the user to navigate through the system?  
Have you used tabs, drop-down menus, buttons, or some other way to assist the user's access / navigation to the functionality provided by the system?

#### **Support for User Efficiency**

How easy is it for the user to complete the task?  
Use of sub forms, tabs, datasheet views or similar functionality  
Relevant content should be included (not unnecessary data, but as much as possible to assist with task)  
Is the utility/functionality correct and all-inclusive?

#### **Recognition rather than recall**

Lookups, searches, drop-down lists  
Consistent screen design within subsystem  
Use of buttons, controls and handling of events

#### **Aesthetic Design**

Layout of screen elements  
Use of colour, White space (not too much, or too little)  
Avoid cluttering of screen or wasted space

#### **Error prevention / recovery**

Provide meaningful messages

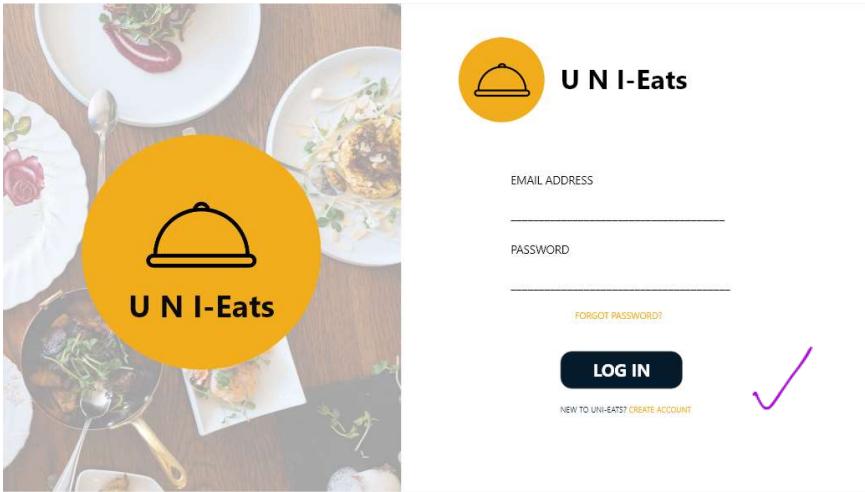
### Error prevention / recovery

The system aims to prevent users from making mistakes which result in errors. It is designed so that the text displayed is visible, understandable, and unambiguous.

Error messages are made to be user friendly. Errors are unavoidable so it is best that these error messages are not condescending or patronizing to the user. Because the user domain will consist of users with different levels of technological competencies, it is important that the error messages provide enough information to help the users fix the error. The system provides concise explanations of what error has occurred, why it has occurred and what does that mean for the user. The users will receive notifications when an error has occurred. This is important as feedback is important for users in understanding what is right and what is wrong.

## 2.2 Team Design for

### 2.2.1 Login / Logout



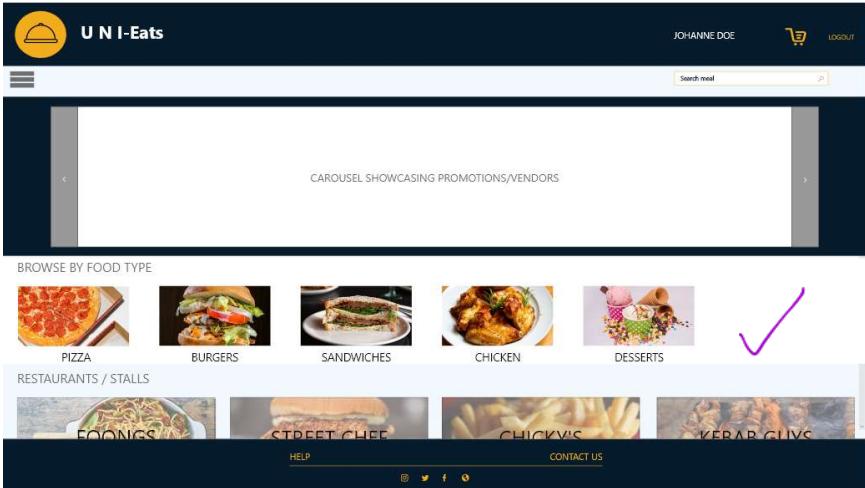
#### Commented [NJ((CS10):

Designs for the following screens should be included here and not under a specific narrative

Login / Logout Feature

Dashboard / Landing Page – this would be the 1<sup>st</sup> screen after a user has logged on

### 2.2.2 Dashboard / Landing Page / Main Page



Is this in an email, is it a flag in DB that is set  $\Rightarrow$  is this the "Send Notification" use case?

## 2.3 UI Designs & Updated Analysis Use Case Narratives

### 2.3.1 Designed by [REDACTED]

Use Case ID	Use Case Name
A0110	Place Order
Primary Business Actors	Other participating Actors
Customer	B0300-
Description	Customer creates an order. The customer can create a new order by selecting specific menu items that they want; or they can choose to replicate a previous order by means of reordering using the order from their order history.
Pre-Conditions	User must be logged in and have access to the Uni-Eats web application.
Triggers	The customer can also trigger A0110 by selecting order item which is seen after triggering A0300 (View Menu) or by triggering A0400 (View Order History) $\xrightarrow{\text{A301}}$
Post-Conditions	Order is placed.
Basic Flow of Events	<ol style="list-style-type: none"> <li>1. System sends order request to Vendor.</li> <li>2. Triggers B0300 (Update Order Status).</li> <li>3. <del>System triggers A0500. (Track Order)</del> <math>\xrightarrow{\text{not removed as recommended in Req. Doc}}</math></li> </ol>
Alternate Flow of Events	<ol style="list-style-type: none"> <li>1. Something goes wrong.           <ol style="list-style-type: none"> <li>1.1 Error message displayed.</li> <li>1.2 A0300 (View Menu) triggered.</li> </ol> </li> </ol>
Initial UI design	

Use Case ID	Use Case Name
A0300	View Menu
Primary Business Actors	Other participating Actors
Customer, Admin	
Description	Users can view the menu.
Pre-Conditions	The user must be logged in to the Uni-Eats site.
Triggers	User selects the option to View Menu of a specific vendor.
Post-Conditions	Vendor menu is displayed to the user.
Basic Flow of Events	<ol style="list-style-type: none"> <li>1. System displays menu.</li> </ol>

### Commented [NJ((CS11):

Your designs can be paper based and then you include scanned images of the designs here

OR

You could do the design in your chosen development environment

OR

you may make use of the following UI prototyping tools to assist you with your designs:

#### Downloadable Tools

Balsamiq (<https://balsamiq.com>)  
Justinmind (<http://www.justinmind.com>)  
Wireframesketcher (<http://wireframesketcher.com>)

#### Online Tools

Fluidui ([www.fluidui.com](http://www.fluidui.com))  
Mockflow ([www.mockflow.com](http://www.mockflow.com))

Include screen snips of the design for each use case completed by each team member. Refer to the narratives that was described in your Requirements document. Some use cases might be designed on one UI or split over multiple designs.

Make sure that you provide annotated notes to your designs

**Commented [JN12]:** The Analysis Use-Case Narratives are derived from the requirements business use-case narrat... [1]

**Commented [JN13]:** Repeat tables below for every use case that the specific team member will be responsible for as ... [2]

**Commented [Janine14]:** Must correspond with Business Use Case model

**Commented [Janine15]:** Identify the users/systems that initiate the use case

**Commented [Janine16]:** ... [3]

**Commented [Janine17]:** Used to describe the overall intent of the use case.

**Commented [Janine18]:** ... [4]

**Commented [Janine19]:** List the event / condition that trigger the use case

**Commented [Janine20]:** List the end results expected by the user. ... [5]

**Commented [Janine21]:** ... [6]

**Commented [JN22]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There ... [7]

**Commented [JN23]:** Include a screen design that will help the reader visualize the behavior of the use case. Anno... [8]

**Commented [Janine24]:** Repeat tables below for every use case

**Commented [Janine25]:** Must correspond with Business Use Case model

**Commented [Janine26]:** Identify the users/systems that initiate the use case

**Commented [Janine27]:** ... [9]

**Commented [Janine28]:** Used to describe the overall intent of the use case.

**Commented [Janine29]:** ... [10]

**Commented [Janine30]:** List the event / condition that trigger the use case

**Commented [Janine31]:** List the end results expected by the user. ... [11]

**Commented [Janine32]:** ... [12]

\* Look at your design, then see if your steps / flow matches with what is happening here, also take into account links from other use cases, specifically A301

	2. User clicks on Menu Item and triggers A301. (View Menu Item)
Alternate Flow of Events	alt2 - return to Home Page
Initial UI design	

**Commented [JN33]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.

Use Case ID	Use Case Name
A0301	View Menu Item
Primary Business Actors	Other participating actors
Customer, Admin, A0300	
Description	Users can view the menu item.
Pre-Conditions	The user must be logged in to the Uni-Eats site.
Triggers	User selects the option to view menu item from a menu view. (A0300)
Post-Conditions	Vendor menu is displayed to the user.
Basic Flow of Events	<ol style="list-style-type: none"> <li>System displays menu item.</li> <li>Information about the meal as well as an option to order is displayed. <i>Add to Cart</i></li> <li>User click returns to menu view and triggers A0300 - (View Menu)</li> </ol>
Alternate Flow of Events	alt3 - user clicks cancel, returns to A300
Initial UI design	

**Commented [JN34]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

**Commented [Janine35]:** Must correspond with Business Use Case model

**Commented [Janine36]:** Identify the users/systems that initiate the use case

**Commented [Janine37]:** List the users/systems that receive messages/report from the use case

**Commented [Janine38]:** Used to describe the overall intent of the use case.

**Commented [Janine39]:** List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

**Commented [Janine40]:** List the event / condition that trigger the use case

**Commented [Janine41]:** List the end results expected by the user.

For example, An order is created

**Commented [Janine42]:** The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Commented [JN43]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.

**Commented [JN44]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

Use Case ID	Use Case Name
B0300	Update order status
Primary Business Actors	Other participating Actors
Vendor	Courier
Description	Maintains the order status.
Pre-Conditions	User must be logged in. A vendor must have received an order.
Triggers	A0110 has completed. B0211 (accept order) or B0220 (cancel order) has been called.
Post-Conditions	Order Status is updated.
Basic Flow of Events	<ol style="list-style-type: none"> <li>Vendor accepts order and status changed to Order Accepted by Vendor.</li> <li>System notifies relevant parties on status change.</li> </ol> <p><i>↳ Is this the send notification?</i></p>
Alternate Flow of Events	<ol style="list-style-type: none"> <li>Vendor rejects order and status changed to Order Rejected by Vendor.</li> <li>System notifies relevant parties on status change.</li> <li>Courier accepts order delivery and status changed to Order Accepted for Delivery.</li> <li>System notifies relevant parties on status change.</li> <li>Vendor starts preparation and status changed to Order Preparation Started.</li> <li>System notifies relevant parties on status change.</li> <li>Preparation of order complete and status changed to Preparation Completed.</li> <li>System notifies relevant parties on status change.</li> <li>Order Picked up for delivery and status changed to Order out for Delivery.</li> <li>System notifies relevant parties on status change.</li> <li>Order delivered and status changed to Delivered</li> <li>System notifies relevant parties on status change.</li> <li>Order paid for and status changed to Order Paid.</li> <li>System notifies relevant parties on status change.</li> <li>System generates error message</li> </ol>

Commented [NJ((CS45]): Remove and replaced by Update Order Status

I cannot see a "Delivered" status on design → Is the Courier detail involved then?

↳ How does this happen?  
Does Vendor get view of new orders, select 1, then checks button

↳ Does not match the radio button options, only 4 on design



Is this use case a "view" or a place where an action is taken or a combination

*Initial UI design*

FOONGS ▾

Item	Time	Quantity	Status
Order #2381	14:28		Preparation Accepted Cancel Completed
Chicken chow mein	1		○ ○ ○ ○
Honey chicken	2		○ ○ ○ ○
Order #2382	14:53		Preparation Accepted Cancel Completed
Chicken chow mein	1		○ ○ ○ ○

HELP CONTACT US

**Commented [JN46]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

Use Case ID	Use Case Name
C0100	Register User
Primary Business Actors	Other participating Actors
Customer	Vendor, Courier
Description	A user must be registered to use the system. Options to register as are customer, vendor or courier.
Pre-Conditions	None
Triggers	User selects Register User option.
Post-Conditions	The system has created a new profile for the user.
Basic Flow of Events	<ol style="list-style-type: none"> <li>System displays the different user profiles.</li> <li>User enters details regarding their respective user role.</li> <li>User selects <u>register</u>. <i>Sign up</i></li> <li>System stores the information.</li> <li>System displays status message.</li> <li>System redirects user to home page.</li> </ol>
Alternate Flow of Events	<ol style="list-style-type: none"> <li>If something goes wrong.</li> <li>Error message displayed.</li> <li>User is redirected back to Registration page.</li> </ol>
Initial UI design	

**Commented [Janine47]:** Must correspond with Business Use Case model

**Commented [Janine48]:** Identify the users/systems that initiate the use case

**Commented [Janine49]:** List the users/systems that receive messages/report from the use case

**Commented [Janine50]:** Used to describe the overall intent of the use case.

**Commented [Janine51]:** List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

**Commented [Janine52]:** List the event / condition that trigger the use case

**Commented [Janine53]:** List the end results expected by the user.

For example, An order is created

**Commented [Janine54]:** The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Commented [JN55]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.

**Commented [JN56]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

*Can user upload a photo? No upload link given  
where is it stored in DB,*

Use Case ID	Use Case Name
C0200	Manage Users
Primary Business Actors	Other participating Actors
Admin	
Description	User accounts including vendor, customer and courier are managed/maintained.
Pre-Conditions	Admin must be logged into the Uni-Eats web application. The admin can take the following options: <ul style="list-style-type: none"> <li>• Update a user's details</li> <li>• Create/remove a user</li> </ul>
Triggers	Admin logs into admin account and navigates to manage users. Admin selects manage users.
Post-Conditions	The appropriate user account has been updated/added/removed.
Basic Flow of Events	<ol style="list-style-type: none"> <li>1. Admin selects manage users.</li> <li>2. Admin selects user to be updated.</li> <li>3. Admin will either update, create or remove a user.</li> <li>4. The system's database is updated to reflect these changes.</li> </ol>

1. will you be able to sort the columns by clicking at top?
2. Maybe also allow to enter Last Name , Vendor Name to Search by i.s.o. User ID

**Initial UI design**

User ID	First name	Last name	User type	Email	Campus
37827	Jane	Doe	Customer	jdoe@gmail.com	North campus

User ID	First name	Last name	User type	Email	Campus	Phone number
37827	Jane	Doe	Courier	jdoe@gmail.com	North campus	083456789

User ID	Name	Phone number	User type	Email	Campus	Cuisine
37820	Foobgs	0745678999	Vendor	foongs@gmail.com	North campus	Asian

**Commented [JN57]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

### 2.3.2 Designed by [REDACTED]

Use Case ID	Use Case Name
A0500	Track Order
Primary Business Actors	Other participating Actors
Customer	A0400
Description	Customer can track any active orders.
Pre-Conditions	Customer must have at least one active order.
Triggers	Customer selects the option to Track Order.
Post-Conditions	Customer can track any of their active orders.
Basic Flow of Events	<ol style="list-style-type: none"> <li>User Clicks on track order.</li> <li>System shows the Customer the current state of the order.</li> <li>System shows Order Description.</li> </ol>
Alternate Flow of Events	what about Cancel (B312)
Initial UI design	<p>The screenshot shows the UNI-Eats mobile application interface. At the top, there's a navigation bar with a logo, the text 'UNI-Eats', and a user profile for 'GARETH'. Below this is a search bar. The main screen is titled 'Track Order' and shows 'Order #4235'. The details for the order are listed: 'Impossible Burger', 'Quantity 2', 'Status Being Prepared', and two collection points: '62534' and '123007'. A handwritten note in pink ink on the right side of the screen says 'rather show descriptive info like Courier Name'. At the bottom of the screen, there are buttons for 'HELP' and 'CONTACT US'.</p>

**Commented [JN58]:** Repeat tables below for every use case that the specific team member will be responsible for as indicated in the Use Case Glossary

You will copy this over from the Requirements Document

**Commented [Janine59]:** Must correspond with Business Use Case model

**Commented [Janine60]:** Identify the users/systems that initiate the use case

**Commented [Janine61]:** List the users/systems that receive messages/report from the use case

**Commented [Janine62]:** Used to describe the overall intent of the use case.

**Commented [Janine63]:** List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

**Commented [Janine64]:** List the event / condition that trigger the use case

**Commented [Janine65]:** List the end results expected by the user.

For example, An order is created

**Commented [Janine66]:**

The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Commented [JN67]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.

**Commented [JN68]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

Use Case ID	Use Case Name
B0100	Maintain Menu
Primary Business Actors	Other participating Actors
Vendor	
Description	Vendor can update their menus. Updating the menu involves adding new items into the menu and removing items from the menu.
Pre-Conditions	The user must be logged in as a vendor. Vendors must be registered with the university. The Vendor must already have a menu in the system.
Triggers	Vendor selects the option to maintain their menu.
Post-Conditions	Vendor menu is updated.
Basic Flow of Events	<ol style="list-style-type: none"> <li>Vendor selects the Update Menu option.</li> <li>Vendor selects Create Menu item option.</li> <li>Vendor inputs new menu item details.</li> <li>System updates menu with new menu item.</li> <li>System displays updated menu to the Vendor.</li> <li>Vendor confirms changes.</li> <li>System saves menu.</li> </ol>

To create  
New  
Menu  
∴ B200

**Commented [Janine69]:** Must correspond with Business Use Case model

**Commented [Janine70]:** Identify the users/systems that initiate the use case

**Commented [Janine71]:** List the users/systems that receive messages/report from the use case

**Commented [Janine72]:** Used to describe the overall intent of the use case.

**Commented [Janine73]:** List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

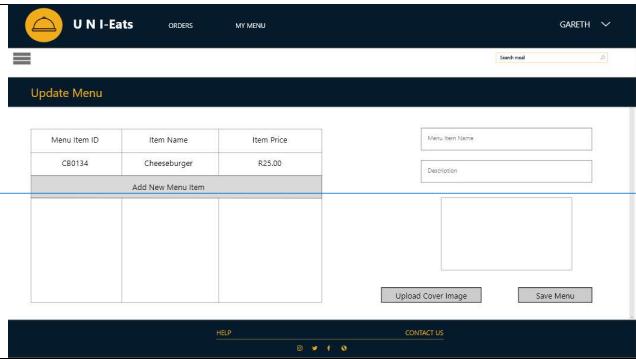
**Commented [Janine74]:** List the event / condition that trigger the use case

**Commented [Janine75]:** List the end results expected by the user.

For example, An order is created

**Commented [Janine76]:**

The main flow of events covers what "normally" happens when the use case is performed.

Alternate Flow of Events	<ol style="list-style-type: none"> <li>2. Vendor selects Remove Menu Item option.</li> <li>3. Vendor selects menu item to remove.</li> <li>4. System removes menu item.</li> <li>5. System displays updated menu to the Vendor.</li> <li>6. Vendor confirms changes.</li> <li>7. System saves menu.</li> </ol> <p style="margin-top: 10px;">2. Vendor selects Update menu Item option.</p> <ol style="list-style-type: none"> <li>1. Vendor selects Update menu Item option.</li> <li>2. Vendor selects Menu item to update.</li> <li>3. System triggers B0211.</li> <li>4. System displays updated menu to the Vendor.</li> <li>5. Vendor confirms changes.</li> <li>6. System saves menu.</li> </ol>
Initial UI design	
Use Case ID	Use Case Name
B0200	Create Menu
Primary Business Actors	Other participating Actors
Vendor	
Description	Vendor can create a new menu .
Pre-Conditions	The user must be logged in as a vendor.
Triggers	Vendor selects the option to Create a new Menu from B0100. (Maintain menu)
Post-Conditions	Vendor menu is created.
Basic Flow of Events	<ol style="list-style-type: none"> <li>3. System creates new menu.</li> <li>4. System displays menu to the Vendor.</li> </ol> <p style="margin-top: 10px;">3. Trigger B0210. (Create menu item)</p> <p style="margin-top: 10px;">4. Add new menu items</p>
Alternate Flow of Events	<ol style="list-style-type: none"> <li>1. Display form to complete</li> <li>2. Enter info</li> </ol>

should this not be within B0211  
rather  
+ no Remove button ?

**Commented [JN77]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.

**Commented [JN78]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

**Commented [Janine79]:**  
Must correspond with Business Use Case model

**Commented [Janine80]:**  
Identify the users/systems that initiate the use case

**Commented [Janine81]:**  
List the users/systems that receive messages/report from the use case

**Commented [Janine82]:**  
Used to describe the overall intent of the use case.

**Commented [Janine83]:**  
List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

**Commented [Janine84]:**  
List the event / condition that trigger the use case

**Commented [Janine85]:**  
List the end results expected by the user.

For example, An order is created

**Commented [Janine86]:**  
The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Commented [JN87]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.

**Initial UI design**

The screenshot shows a web-based application for creating menu items. At the top, there's a navigation bar with 'UNI-Eats', 'ORDERS', 'MY MENU', and a user profile for 'GARETH'. Below the navigation is a search bar. The main area is titled 'Create Menu Item' and contains two input fields: 'Item Name' and 'Short Description'. There's also a placeholder 'Upload Cover Image'. A prominent blue button labeled 'Create Menu Item' is at the bottom right. A handwritten note in pink says 'Add Items to menu' next to the button.

**Commented [JN88]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

Use Case ID	Use Case Name
B0210	Create menu item
Primary Business Actors	Other participating Actors
Vendor	
Description	Vendor can create new menu items.
Pre-Conditions	The user must be logged in as a vendor.
Triggers	Vendor selects the option to Update Menu.
Post-Conditions	Vendor menu is updated.
Basic Flow of Events	<ol style="list-style-type: none"> <li>System asks Vendor for item to add.</li> <li>Vendor inputs menu item details.</li> <li>System creates menu item.</li> </ol>
Alternate Flow of Events	<ol style="list-style-type: none"> <li>If an error occurs.             <ol style="list-style-type: none"> <li>System displays error message.</li> <li>System reverts to 1.</li> </ol> </li> </ol>
<b>Initial UI design</b>	<p>The screenshot shows the same 'Create Menu Item' interface as above, but with additional fields: 'Price' and 'Categories' (with options like American, Asian, Chinese, Greek, Italian, and French). A handwritten note in pink points to the 'Price' field with the text 'Show currency, or is'. Another note below it asks 'this shown as soon as you enter a number?'</p>

**Commented [Janine89]:**  
Must correspond with Business Use Case model

**Commented [Janine90]:**  
Identify the users/systems that initiate the use case

**Commented [Janine91]:**  
List the users/systems that receive messages/report from the use case

**Commented [Janine92]:**  
Used to describe the overall intent of the use case.

**Commented [Janine93]:**  
List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

**Commented [Janine94]:**  
List the event / condition that trigger the use case

**Commented [Janine95]:**  
List the end results expected by the user.

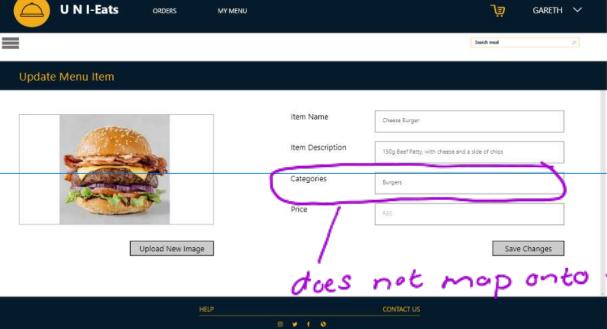
For example, An order is created

**Commented [Janine96]:**  
The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Commented [JN97]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.

**Commented [JN98]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

Use Case ID	Use Case Name
B0210	Update menu item
Primary Business Actors	Other participating Actors
Vendor	
Description	Vendor can update their menu.
Pre-Conditions	The user must be logged in as a vendor.
Triggers	Vendor selects the option to Update Menu.
Post-Conditions	Vendor menu is updated.
Basic Flow of Events	<ol style="list-style-type: none"> <li>System asks Vendor for items to be updated and displays menu item details.</li> <li>Vendor inputs menu item details.</li> <li>System displays menu to the Vendor.</li> <li>System asks vendor to confirm changes.</li> <li>System saves changes.</li> </ol>
Alternate Flow of Events	<p>1. If an error occurs.      1.1 System displays error message.      1.2 System reverts to 1.</p> <p style="text-align: right;"><i>What about Remove Item (see B100)</i></p>
Initial UI design	 <p><i>does not map onto the multiple categories you would be able to choose from in B210 → change the view of this attribute</i></p>

**Commented [Janine99]:**  
Must correspond with Business Use Case model

**Commented [Janine100]:**  
Identify the users/systems that initiate the use case

**Commented [Janine101]:**  
List the users/systems that receive messages/report from the use case

**Commented [Janine102]:**  
Used to describe the overall intent of the use case.

**Commented [Janine103]:**  
List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

**Commented [Janine104]:**  
List the event / condition that trigger the use case

**Commented [Janine105]:**  
List the end results expected by the user.

For example, An order is created

**Commented [Janine106]:**  
The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Commented [JN107]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.

**Commented [JN108]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

*does not map onto the multiple categories you would be able to choose from in B210 → change the view of this attribute*

Use Case ID	Use Case Name
B0310	View Order
Primary Business Actors	Other participating Actors
Vendor	
Description	A vendor can view an order that a customer has placed.
Pre-Conditions	Customer must have successfully placed an order at that specific vendor.
Triggers	Vendor selects view order option.
Post-Conditions	A customer's order is displayed to the vendor.
Basic Flow of Events	<ol style="list-style-type: none"> <li>The system displays order information.</li> <li>Vendor views order and B0211. (accept order)</li> </ol>

**Commented [NJ((CS109):** What happens after order information is displayed?  
Vendor would then Accept Order (call B0211)

*how ?  
do they select a button*

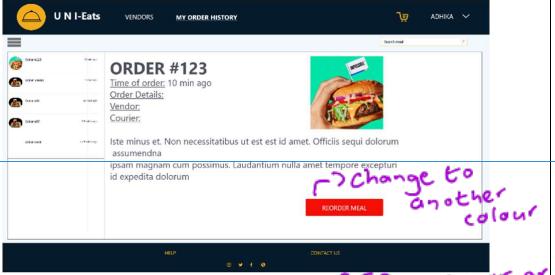
Alternate Flow of Events	1. Vendor views order and B0220. (cancel order) <i>now?</i>															
<b>Initial UI design</b>	<p>View Order</p> <p>Order Details</p> <p>Order #4235</p> <table border="1"> <thead> <tr> <th></th> <th>Quantity</th> <th>Status</th> <th>Courier ID</th> <th>Collection Point</th> </tr> </thead> <tbody> <tr> <td>Impossible Burger</td> <td>2</td> <td>Being Delivered</td> <td>62354</td> <td>1230007</td> </tr> <tr> <td>Impossible Burger</td> <td>1</td> <td>Cancelled - No Ingredients</td> <td></td> <td></td> </tr> </tbody> </table>		Quantity	Status	Courier ID	Collection Point	Impossible Burger	2	Being Delivered	62354	1230007	Impossible Burger	1	Cancelled - No Ingredients		
	Quantity	Status	Courier ID	Collection Point												
Impossible Burger	2	Being Delivered	62354	1230007												
Impossible Burger	1	Cancelled - No Ingredients														

**Commented [JN110]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

Use Case ID	Use Case Name										
C0400	<del>Pay Order</del> <i>Receive Payment</i>										
Primary Business Actors	Other participating Actors										
A0600	<i>Courier</i>										
Description	A customer pays cash for their order once it has been delivered to them										
Pre-Conditions	Customer must have the Uni-Eats web application open and is logged in. The checkout process must be completed by the customer.										
Triggers	The courier receives a cash payment from the customer for their respective order.										
Post-Conditions	The customer's order is paid for, this payment is reflected in the database.										
Basic Flow of Events	<ol style="list-style-type: none"> <li>1. Courier receive cash from customer.</li> <li>2. Courier selects receive payment.</li> <li>3. The system's database is updated to reflect payment from customer.</li> </ol> <p>* not updated as per recommendation from Req.doc</p>										
<b>Initial UI design</b>	<p>MY DELIVERIES</p> <p>GARETH</p> <p>Pay Order</p> <p>Order Details for Selected Order:</p> <table border="1"> <tr> <td>Order ID</td> <td>#54326</td> </tr> <tr> <td>Order Status</td> <td>Delivered</td> </tr> <tr> <td>Collection Point</td> <td>1230007</td> </tr> <tr> <td>Order Items</td> <td>Impossible Burger Chocolate Pancakes</td> </tr> <tr> <td>Order Total</td> <td>R 86.50</td> </tr> </table> <p>Confirm Receipt of Cash</p>	Order ID	#54326	Order Status	Delivered	Collection Point	1230007	Order Items	Impossible Burger Chocolate Pancakes	Order Total	R 86.50
Order ID	#54326										
Order Status	Delivered										
Collection Point	1230007										
Order Items	Impossible Burger Chocolate Pancakes										
Order Total	R 86.50										

**Commented [JN111]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

### 2.3.3 Designed by [REDACTED]

Use Case ID	Use Case Name	
A0400	View Order History	
Primary Business Actors		Other participating Actors
Customer		
Description	Customer can view previous orders they have made.	
Pre-Conditions	Customer must be logged on.	
Triggers	Customer selects the option to View Order History.	
Post-Conditions	Customer can see the orders that they have made in the past.	
Basic Flow of Events	1. System displays the customer's order history. 2. User selects order. 3. A0500 is triggered. <i>(list of previous order)</i> <i>why?</i> <i>display selected order</i> <i>select to Order again (A110)</i>	
Alternate Flow of Events	1. User has not made any previous orders. 1.1 System displays error message and prompts user to place an order. 1.2 Revert to Main Menu.	
Initial UI design		

Use Case ID	Use Case Name	
A0600	View Vendor	
Primary Business Actors		Other participating Actors
Customer		
Description	Customers can view information about the vendor, as well as average rating.	
Pre-Conditions	User must be logged in as customer.	
Triggers	Customer selects View Vendor button from displayed vendors.	
Post-Conditions	Customer is taken to a page that shows the details of the selected vendor.	
Basic Flow of Events	1. System displays information about the vendor.	
Alternate Flow of Events		

D) would this be from Home Page  
that you would see list

**Commented [JN112]:** Repeat tables below for every use case that the specific team member will be responsible for as indicated in the Use Case Glossary

You will copy this over from the Requirements Document

**Commented [Janine113]:**  
Must correspond with Business Use Case model

**Commented [Janine114]:**  
Identify the users/systems that initiate the use case

**Commented [Janine115]:**  
List the users/systems that receive messages/report from the use case

**Commented [Janine116]:**  
Used to describe the overall intent of the use case.

**Commented [Janine117]:**  
List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

**Commented [Janine118]:**  
List the event / condition that trigger the use case

**Commented [Janine119]:**  
List the end results expected by the user.

For example, An order is created

**Commented [Janine120]:**  
The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Commented [JN121]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.

**Commented [JN122]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

**Commented [Janine123]:**  
Must correspond with Business Use Case model

**Commented [Janine124]:**  
Identify the users/systems that initiate the use case

**Commented [Janine125]:**  
List the users/systems that receive messages/report from the use case

**Commented [Janine126]:**  
Used to describe the overall intent of the use case.

**Commented [Janine127]:**  
List anything that must be complete prior to the initiation of the use case.

[... 14]

**Commented [Janine128]:**  
List the event / condition that trigger the use case

**Commented [Janine129]:**  
List the end results expected by the user.

[... 15]

**Commented [Janine130]:**  
The main flow of events covers what "normally" happens when the use case is performed.

[... 16]

**Commented [JN131]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case.

[... 17]



**Commented [JN132]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

Use Case ID	Use Case Name
A0200	Rate Food
Primary Business Actors	Other participating Actors
Customer	
Description	Customer can rate the food/service provided by a vendor. (ratings are done on a five-star rating system, customers also have the option of writing a comment)
Pre-Conditions	Customer has ordered from the vendor who they want to rate.
Triggers	Customer has completed an order with a Vendor. <i>Select Rate Food</i>
Post-Conditions	The rating given by customer will contribute to the overall rating that a vendor has on the system after being verified by Administrator. The rating and feedback will also be available to the vendors. <i>1. Display vendors</i> <i>2. Select</i>
Basic Flow of Events	<ol style="list-style-type: none"> <li>3+ User selects stars for rating and then has the option of writing a comment.</li> <li>4. User selects submit button.</li> <li>5. System displays confirmation message.</li> <li>6. System sends feedback to Admin and triggers <i>C0400</i>. <i>(Verify Ratings)</i></li> <li>7. System updates vendor's overall rating.</li> </ol>
Alternate Flow of Events	<ol style="list-style-type: none"> <li>5. System displays an error message.</li> <li>5.1 System allows user to reenter feedback.</li> <li>5.2 Revert to 3.</li> </ol>
Initial UI design	<p><i>→ would user select from here?</i></p>

**Commented [Janine133]:** Must correspond with Business Use Case model

**Commented [Janine134]:** Identify the users/systems that initiate the use case

**Commented [Janine135]:** List the users/systems that receive messages/report from the use case

**Commented [Janine136]:** Used to describe the overall intent of the use case.

**Commented [Janine137]:** List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

**Commented [Janine138]:** List the event / condition that trigger the use case

**Commented [Janine139]:** List the end results expected by the user.

For example, An order is created

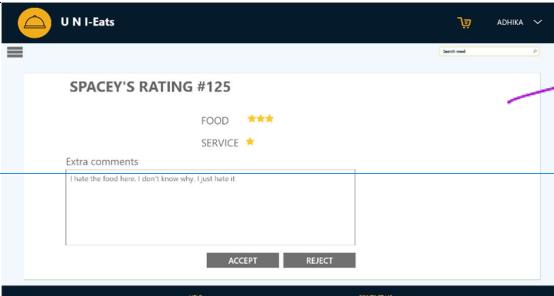
**Commented [Janine140]:** The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Commented [JN141]:** Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.

**Commented [JN142]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

*not updated based feedback from  
Req Doc*

Use Case ID	Use Case Name
C0300	Verify Rating
Primary Business Actors	Other participating Actors
A0600	
Description	A customer pays cash for their order once it is placed.
Pre-Conditions	Customer must have the Uni-Eats web application open and is logged in. The checkout process must be completed by the customer.
Triggers	The courier receives a cash payment from the customer for their respective order.
Post-Conditions	The customer's order is paid for, this payment is reflected in the database.
Basic Flow of Events	<ol style="list-style-type: none"> <li>Courier receives cash from customer.</li> <li>Courier selects receive payment.</li> <li>The system's database is updated to reflect payment from customer.</li> </ol>
Initial UI design	

Steps for Pay Order

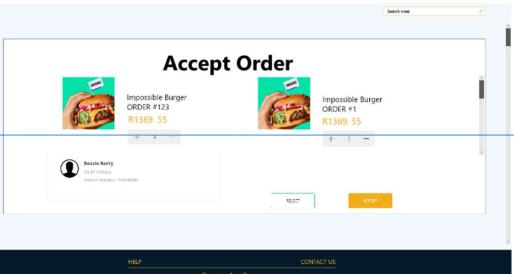
1st get list of reviews to verify

**Commented [JN143]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

Use Case ID	Use Case Name
B0311	Accept order
Primary Business Actors	Other participating Actors
Vendor	
Description	A vendor can choose to accept or reject an incoming order.
Pre-Conditions	A customer has placed an order at the respective vendor. Vendor must be logged in to the Uni-Eats web application.
Triggers	A500(Track Order) triggers B0311. Called from B310
Post-Conditions	A customer's order is accepted and is a pending state. (i.e. order is yet to be prepared)
Basic Flow of Events	<ol style="list-style-type: none"> <li>The system calls B0300.</li> <li>System displays that order is being prepared. (incoming orders are displayed on vendor's screen from oldest orders to newest orders. Orders are prepared according to this list – that is orders that were made earlier will be prepared before orders that were made later)</li> </ol>

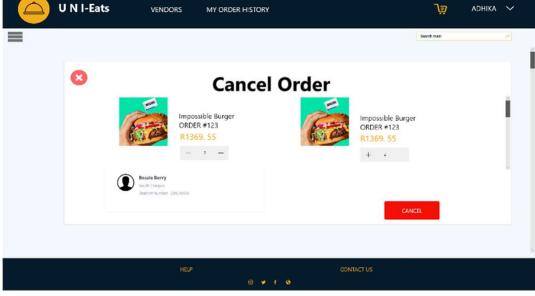
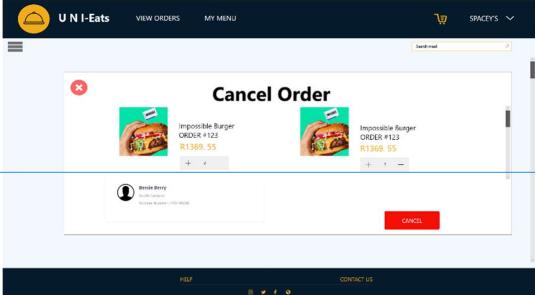
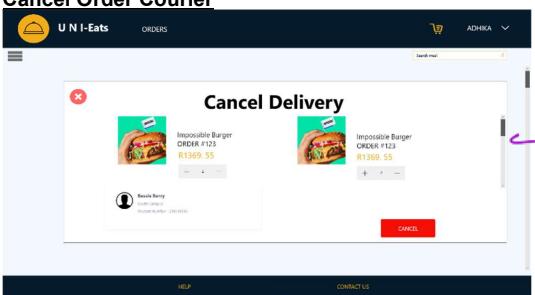
1. Show you screen(p23)  
2 User Accepts

this belongs to B300 logic

<b>Alternate Flow of Events</b>	
<b>Initial UI design</b>	

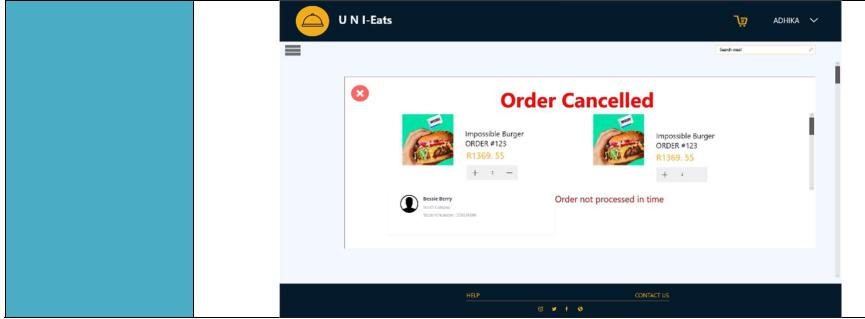
**Commented [JN144]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

Use Case ID	Use Case Name
B0312	Cancel Order
Primary Business Actors	Other participating Actors
Vendor, Courier, Customer, Time	
Description	<p>An order can be cancelled in several ways.</p> <p>A customer can manually cancel an order. This is only possible in the period before the order has started being prepared by the vendor.</p> <p>A vendor may cancel an order at any time. A penalty will be incurred for canceling an order after a certain amount of time</p> <p>A courier may cancel an order only before they accept an order.</p> <p>If a certain amount of time has passed before a vendor accepts an order, a courier accepts a delivery, or a customer fails to pick up their order then the system will automatically cancel the order</p>
Pre-Conditions	Consumer must have placed an order
Triggers	<p>Customer pressing the cancel order button. (from A500)</p> <p>Vendor pressing the cancel order button. (from B310)</p> <p>Courier pressing the cancel order button.</p> <p>Order not accepted in the time limit. what is the limit?</p>
Post-Conditions	Order is canceled and deleted from the system.
Basic Flow of Events	<ol style="list-style-type: none"> <li>Customer cancels order and system calls B0300.</li> <li>System notifies Vendor and Courier that Customer has canceled order.</li> <li>Returns to main menu.</li> </ol>
Alternate Flow of Events	<ol style="list-style-type: none"> <li>System cancels order. (vendor)</li> <li>System calls B0300.</li> <li>System notifies Customer and Courier that Vendor has canceled order.</li> <li>Returns to main menu.</li>   <li>System cancels delivery acceptance of order. (courier)</li> <li>System notifies Customer and Vendor that Vendor has canceled delivery.</li> </ol>

	<p>3. System calls B0300.</p> <p>4. Returns to main menu.</p> <p>1. Either Vendor has not accepted order, or no Couriers have accepted the delivery and enough time has passed.</p> <p>2. System cancels the order.</p> <p>3. System notifies Customer, Vendor and Courier that order has been canceled.</p> <p>4. System calls B0300.</p> <p>5. Returns to main menu.</p>
	<p><b>Cancel Order Customer</b></p>  <p><b>Cancel Order Vendor</b></p>  <p><b>Cancel Order Courier</b></p>  <p><b>Cancel Order Time</b></p>
<b>Initial UI design</b>	

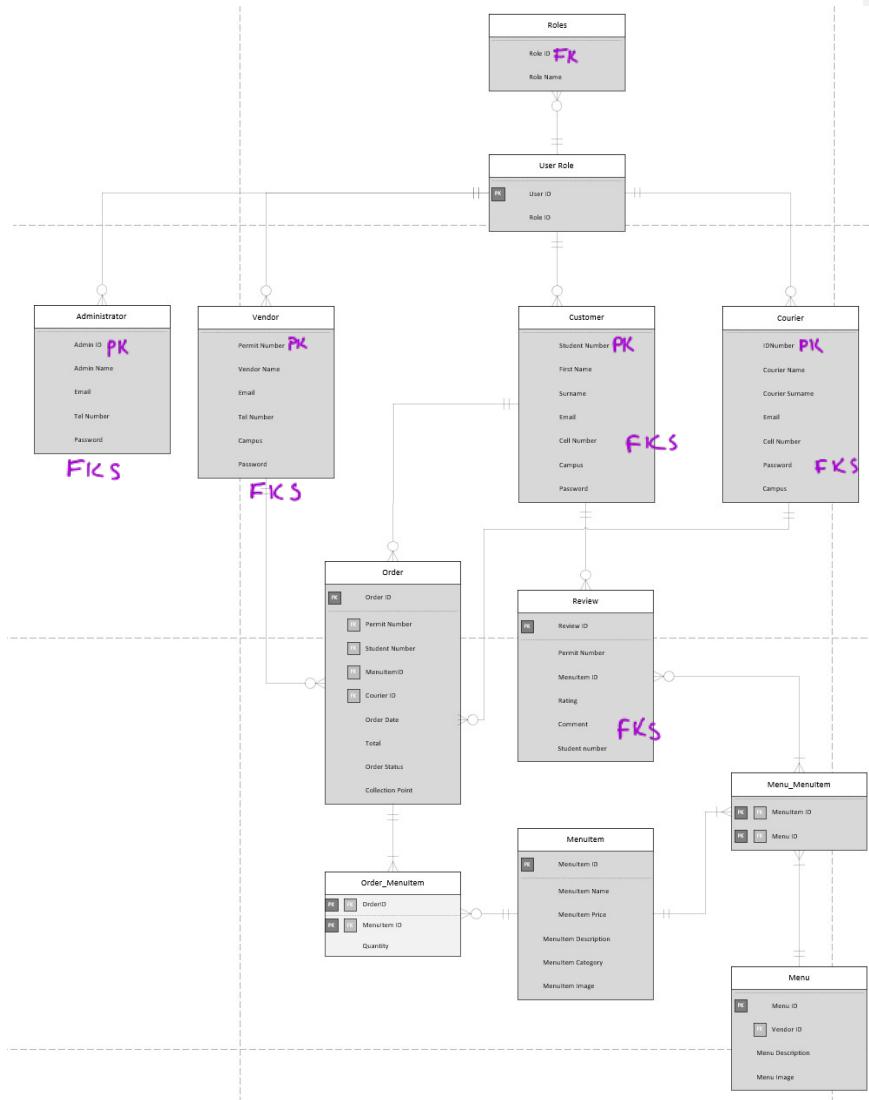
**Commented [JN145]:** Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

→ would a new  
courier then  
not be  
selected ?



### 3 DATA REQUIREMENTS

#### Implementation Ready Class Diagram



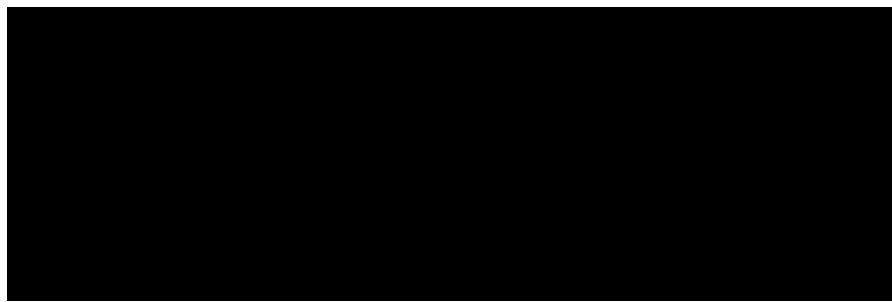
**Commented [NJ((CS146): Draw an implementation ready Entity Relationship Diagram (ERD) using UML notation**

Refer to your WRDBV notes for an explanation of this diagram

You may include the attributes in a separate table and only show the Pks and Fks on the diagram

**Module code:** WRRV301

We,

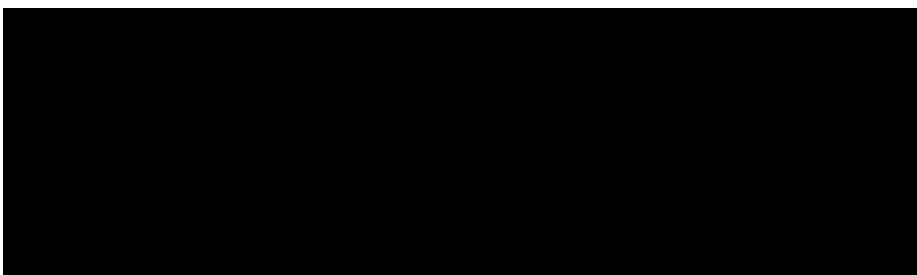


hereby declare that this submission is our own, original work.

We further declare that:

1. No part of this submission has been copied from another person/group,
2. We did / did not work with another person/group on this submission,
3. We acknowledged all consulted sources in the text and submitted a list of references, and
4. Parts without references are entirely our own work
5. That we have all equally contribute to the work or as indicated in the contribution % above.

*(Delete any of the above if not applicable)*



**Page 9: [1] Commented [JN12]** **Janine Nel** **2019/05/13 21:31:00**

The Analysis Use-Case Narratives are derived from the requirements business use-case narratives, which are expanded to include the use case's typical course of events and its alternate courses

Update the narratives accordingly based on the feedback you received from the Requirements Document Assessment

**Page 9: [2] Commented [JN13]** **Janine Nel** **2019/05/13 21:34:00**

Repeat tables below for every use case that the specific team member will be responsible for as indicated in the Use Case Glossary

You will copy this over from the Requirements Document

**Page 9: [3] Commented [Janine16]** **Janine Nel** **2010/02/09 20:03:00**

List the users/systems that receive messages/report from the use case

**Page 9: [4] Commented [Janine18]** **Janine Nel** **2010/02/09 20:36:00**

List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

**Page 9: [5] Commented [Janine20]** **Janine Nel** **2010/02/09 20:39:00**

List the end results expected by the user.

For example, An order is created

**Page 9: [6] Commented [Janine21]** **Janine Nel** **2010/02/09 20:44:00**

The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Page 9: [7] Commented [JN22]** **Janine Nel** **2020/03/19 20:07:00**

Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.

**Page 9: [8] Commented [JN23]** **Janine Nel** **2019/05/13 21:38:00**

Include a screen design that will help the reader visualize the behavior of the use case. Annotate / provide explanatory notes where necessary.

**Page 9: [9] Commented [Janine27]** **Janine Nel** **2010/02/09 20:03:00**

List the users/systems that receive messages/report from the use case

**Page 9: [10] Commented [Janine29]** **Janine Nel** **2010/02/09 20:36:00**

List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

**Page 9: [11] Commented [Janine31]** **Janine Nel** **2010/02/09 20:39:00**

List the end results expected by the user.

For example, An order is created

**Page 9: [12] Commented [Janine32]** **Janine Nel** **2010/02/09 20:44:00**

The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Page 15: [13] Commented [Janine76]** **Janine Nel** **2010/02/09 20:44:00**

The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Page 20: [14] Commented [Janine127]** **Janine Nel** **2010/02/09 20:36:00**

List anything that must be complete prior to the initiation of the use case.

For example, the user may be required to be logged on

**Page 20: [15] Commented [Janine129]** **Janine Nel** **2010/02/09 20:39:00**

List the end results expected by the user.

For example, An order is created

**Page 20: [16] Commented [Janine130]** **Janine Nel** **2010/02/09 20:44:00**

The main flow of events covers what "normally" happens when the use case is performed.

The basic flow is often represented as a numbered list that describes the interaction between an actor and the system. Decision points in the basic flow branch off to alternate flows.

**Page 20: [17] Commented [JN131]** **Janine Nel** **2020/03/19 20:07:00**

Alternate Flows capture variations to the basic flows, such as user decisions or error conditions. There are typically multiple alternate flows in a single use case. Some alternate flows rejoin the basic flow at a specified point, while others terminate the use case.