ERROR404
Elaboration Document

NMU FOODIES

$\frac{82}{100}$

Team Members:

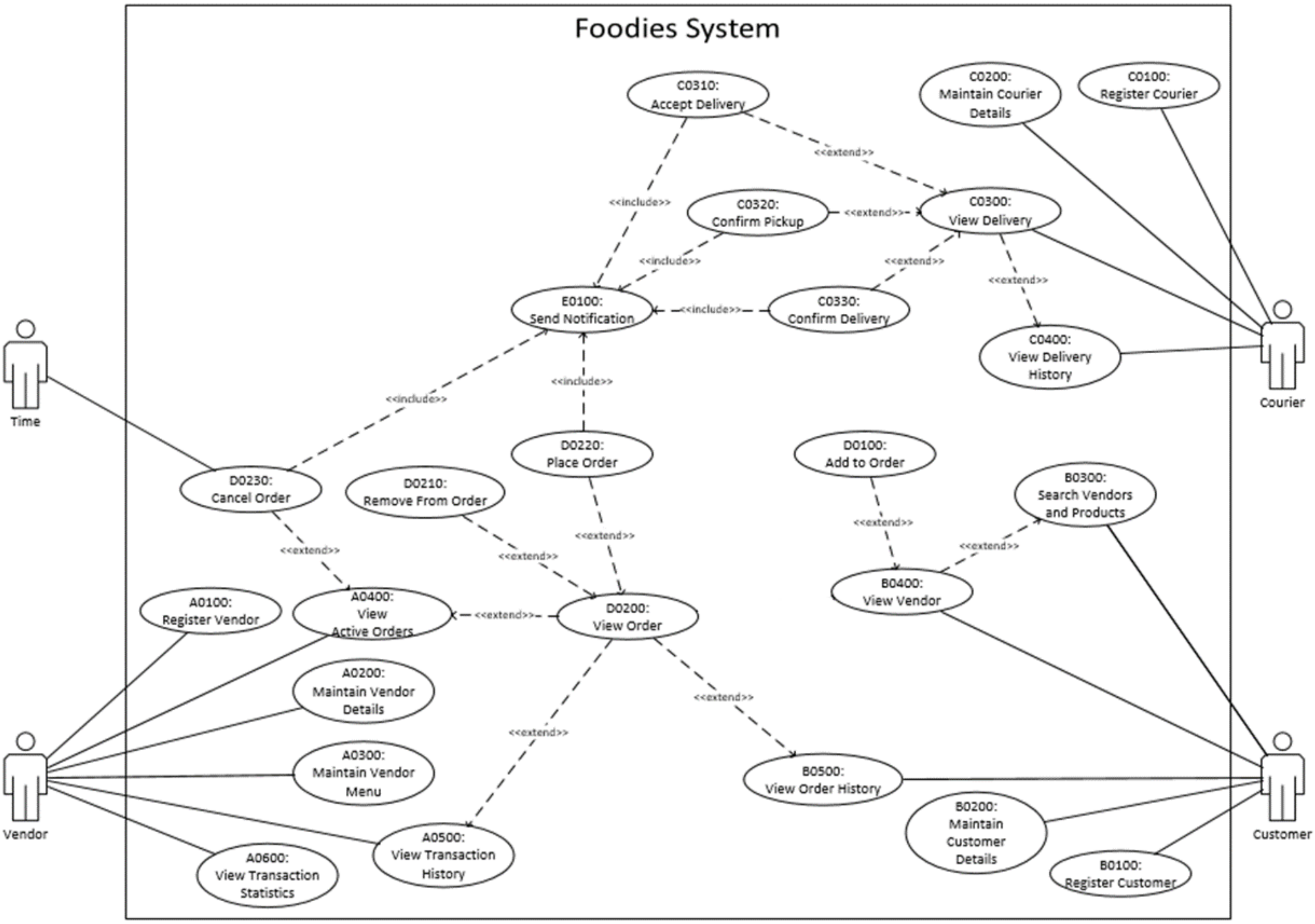
[redacted]

Document Date: 27 Aug 2020

Table of Contents

1 FUNCTIONAL REQUIREMENTS

1.1 Analysis Use Case Model



1.2 Use Case Glossary and Responsibilities

Team Member Responsible: [redacted]	
Use Case Id	**Use Case Name**
A0100 ✓	Register Vendor
A0200 ✓	Maintain Vendor Details
A0300 ✓	Maintain Vendor Menu
Queries/Reports	
A0400 ✓	View Active Orders
A0500 ✓	View Transaction History
A0600 ✓	View Transaction Statistics

Team Member Responsible: [redacted]	
Use Case Id	**Use Case Name**
B0100 ✓	Register Customer
B0200 ✓	Maintain Customer Details
B0300 ✓	Search Vendors and Products
Queries/Reports	
B0400 ✓	View Vendor
B0500 ✓	View Order History

Team Member Responsible: [redacted]	
Use Case Id	**Use Case Name**
C0100 ✓	Register Courier
C0200 ✓	Maintain Courier Details
C0310 ✓	Accept Delivery
C0320 ✓	Confirm Pickup
C0330 ✓	Confirm Delivery
Queries/Reports	
C0300 ✓	View Delivery
C0400 ✓	View Delivery History

Team Member Responsible: All Team Members	
Use Case Id	**Use Case Name**
D0100 ✓	Add To Order
D0210 ✓	Remove From Order
D0220 ✓	Place Order
D0230 ✓	Cancel Order
Queries/Reports	
D0200 ✓	View Order
E0100 ✓	Send Notification

2 UI Prototypes

2.1 Team UI Guidelines

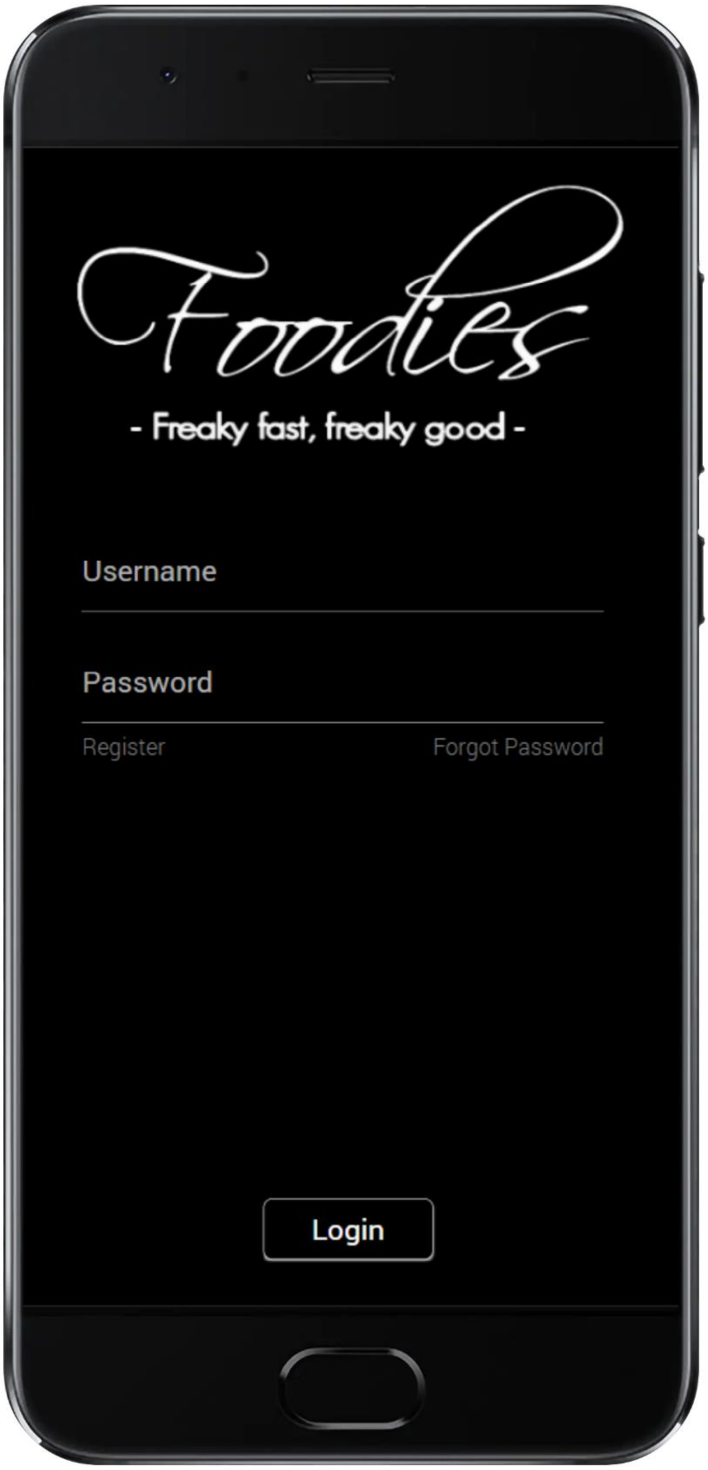
The NMU Foodies application will include a navigation bar fixed at the bottom of the screen allowing for navigation to the three main widgets/views/screens of the application viz. *Home, Account* and *Orders*. The navigational controls will take the form of icon buttons and the currently selected icon button on the navigation bar will be styled differently as to emphasize its selected state. The system can further be navigated by means of the built-in android system navigation as well as a *back* button positioned at the top-left-hand corner of certain screens. Navigation within certain screens – e.g. navigation between tabs in a tab view – will also be implemented alongside other navigational controls such as buttons used for redirecting the user to different sections of the application. This navigation scheme is fairly consistent with most modern-day mobile applications and should be fairly easy to learn. This navigation scheme ensures usability and an enjoyable user experience.

With regards to user-input, appropriate controls will be implemented where necessary: date-time pickers will be used for the capturing of times and dates, drop-down menus, radio-groups and list-boxes will be implemented when multiple options are available to the user and text fields will be formatted and styled correctly when sensitive information needs to secured/protected from prying eyes. A search bar will be used for lookup and along with buttons and all other controls, will adhere to a consistent styling theme which will be implemented system wide. A large amount of controls and other on-screen elements will follow a somewhat materialistic design incorporating the subtle use of rounded corners to create a modern appeal. The system UI will primarily be dark with pops of colour to draw the user's attention to critical UI elements while also being easy on the eyes. The choice of our controls, their appearance as well as their consistent positioning across screens will ensure that UI concept of *affordance* is adhered to.

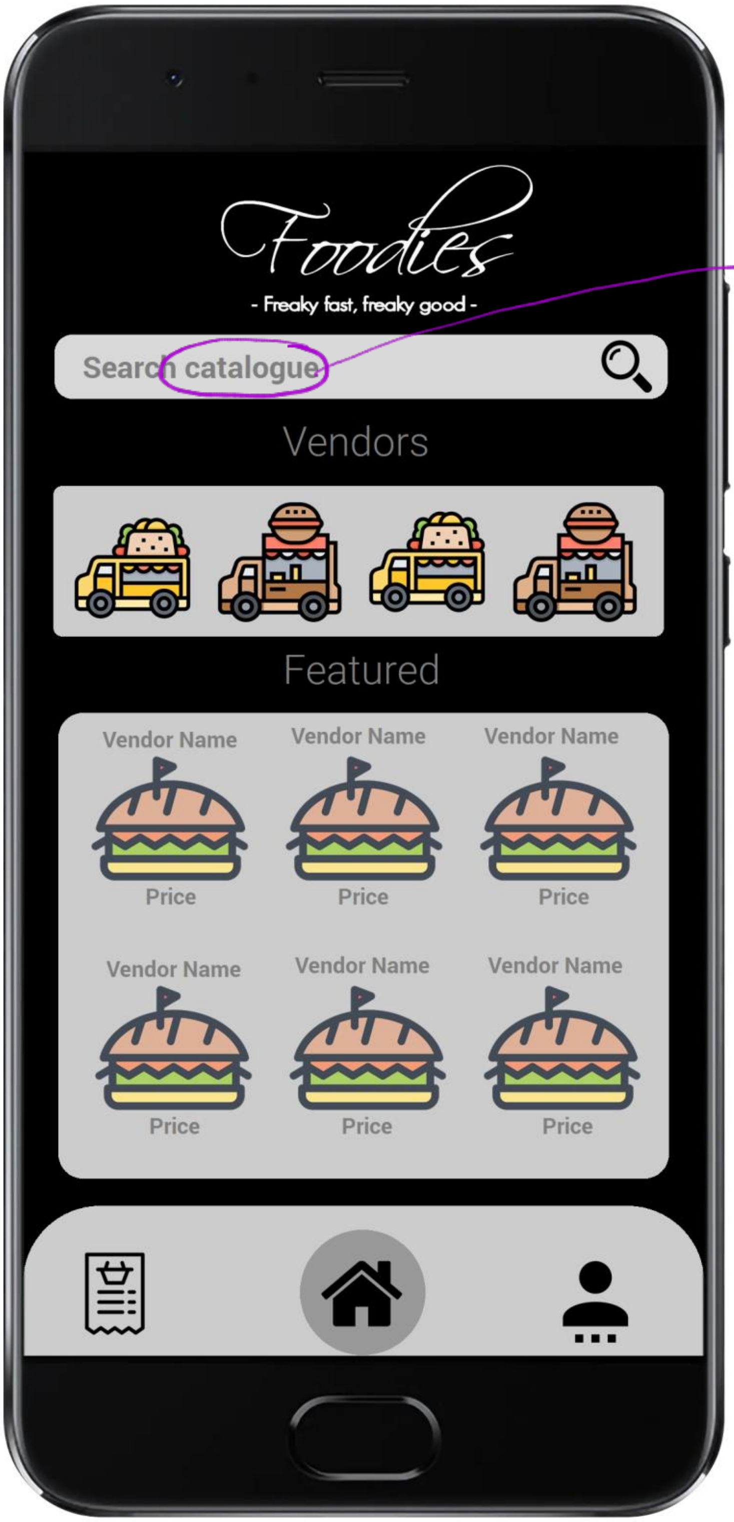
The system will provide consistent and user-friendly feedback in the form of on-screen real-time messages and visual updates (which could pertain to numerous things other than merely errors and confirmation messages). Alerts, i.e. pop-up dialogs, will be used in order to confirm user action or to convey messages. These alerts will be centered on the screen and the underlying widget (on which the alert will be drawn) will be faded/blurred in order to emphasize the pop-up dialog. Error catching will be handled accordingly in the code but front-end acknowledgement via error messaged as well as the disabling of controls under certain circumstances will ensure that the user is always aware of the consequences of their actions and won't be able to perform any actions that could compromise the system in any way.

2.2 Team Design for

2.2.1 Login / Logout



2.2.2 Dashboard / Landing Page / Main Page



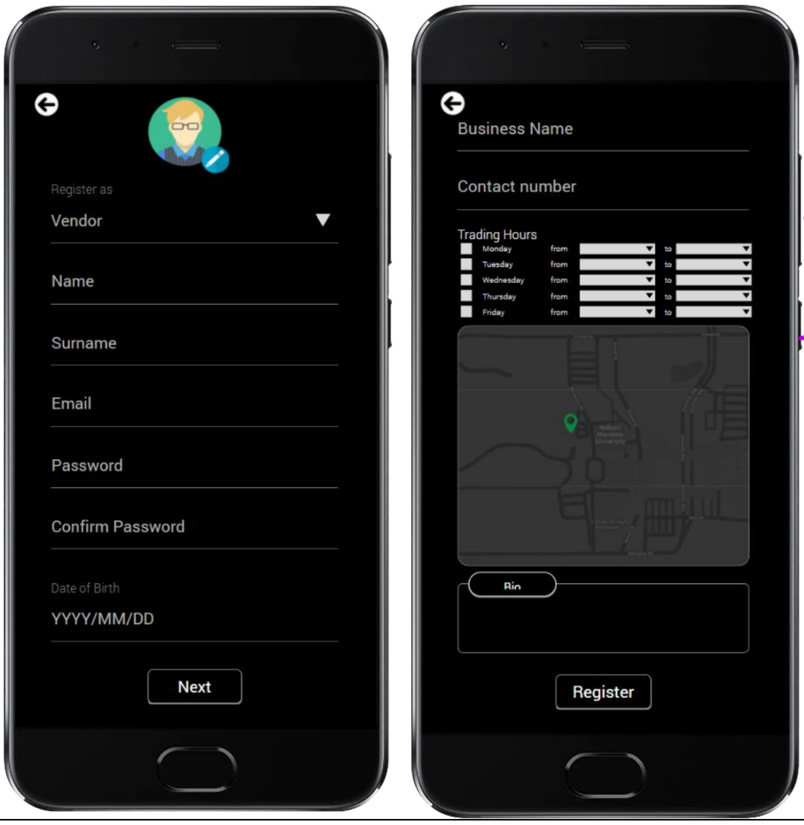
Vendor menus (then it is within context of app)

2.3 UI Designs & Updated Analysis Use Case Narratives

2.3.1 Designed by [redacted]

Use Case ID	Use Case Name
A0100	Register Vendor
Primary Business Actors	**Other participating Actors**
Vendor	
Description	Vendors can setup their accounts. Relevant business information such as Vendor Name, Vendor contact number, Vendor trading hours, Vendor location (campus) and Vendor manager's personal information such as name, surname, email address and password will be stored for various verification procedures.
Pre-Conditions	The vendor must have installed the system on a device.
Triggers	Vendor taps on the application's icon in their application drawer on their device and at the landing screen, clicks the "Register" button.
Post-Conditions	An account is created for the Vendor. An identification number is created for the Vendor and stored with other Vendor information in a database. An email is sent to the Vendor's manager congratulating him/her for registering.
Basic Flow of Events	1. Select Vendor user type on the registration screen. 2. System displays the registration screen with fields into which relevant details should be entered. 3. Vendor inputs Vendor Information: name, contact number, trading hours, location. 4. The Vendor inputs the manager's information: name, surname, email address and password (twice to ensure correctly inputted password). 5. The Vendor taps the "Register" button. 6. System validates entered details. 7. System sends email to user containing confirmation code 8. System requests user to enter confirmation code. 9. User enters confirmation code 10. The system displays a message informing the user of successfully registering. 11. Home Screen is displayed
Alternate Flow of Events	5. The Vendor taps the "Cancel" button. 6. System returns to the landing screen.
Alternate Flow of Events	7. System informs user that details provided are invalid or insufficient. 8. Return to basic flow step 2.
Alternate Flow of Events	10. The system displays error message due to incorrect confirmation code. 11. Return to basic flow step 8.

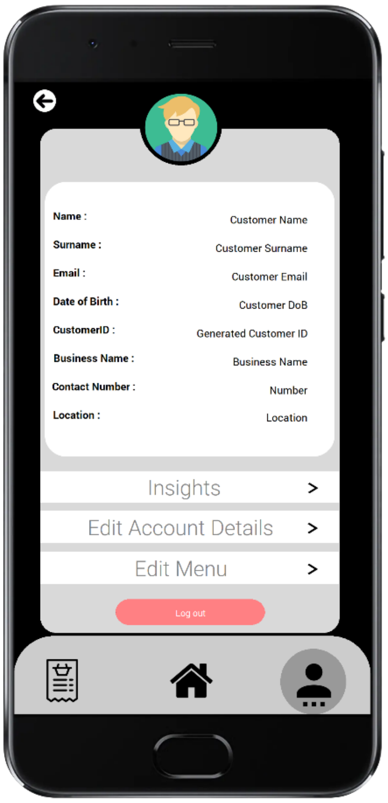
have you thought about how this would be implemented

Initial UI Design	

Is there provision for multi-values in db?

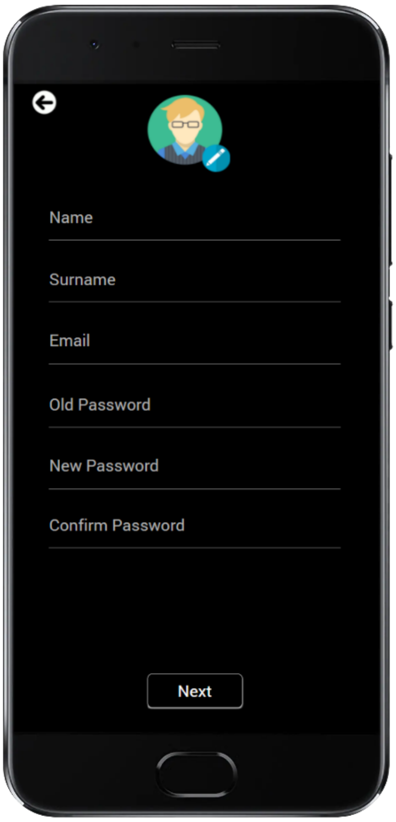
Use Case ID	**Use Case Name**
A0200	Maintain Vendor Details
Primary Business Actors	**Other participating Actors**
Vendor	
Description	Allows the vendor to maintain vendor related account details by navigating to a screen where the vendor can manage the vendor's details.
Pre-Conditions	A Vendor user must be logged in.
Triggers	If the vendor details are declared insufficient by the system or when the Vendor clicks a "Edit Details" button on the interface.
Post-Conditions	The Vendor's details are updated in the database so that the updated details can be displayed on the interface for all users. Vendor user is returned to his account screen (Vendor Screen).
Basic Flow of Events	1. The system displays the Update Vendor Details screen. 2. The Vendor enters the necessary details in the provided fields. 3. The Vendor clicks the "Update" button. 4. The system validates the entered details 5. The system displays a message confirming that the details were successfully updated. 6. Vendor is returned to the Account Screen.
Alternate Flow of Events	3. The Vendor clicks the "Cancel" button. 4. Return to basic flow step 6.
Alternate Flow of Events	5. The system displays a message informing the user that entered details are not valid or that the changes could not be saved. 6. Return to basic flow step 1.

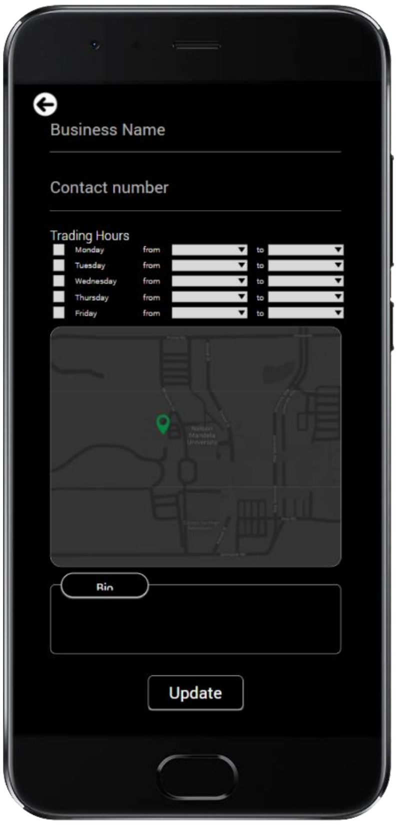
Initial UI Design

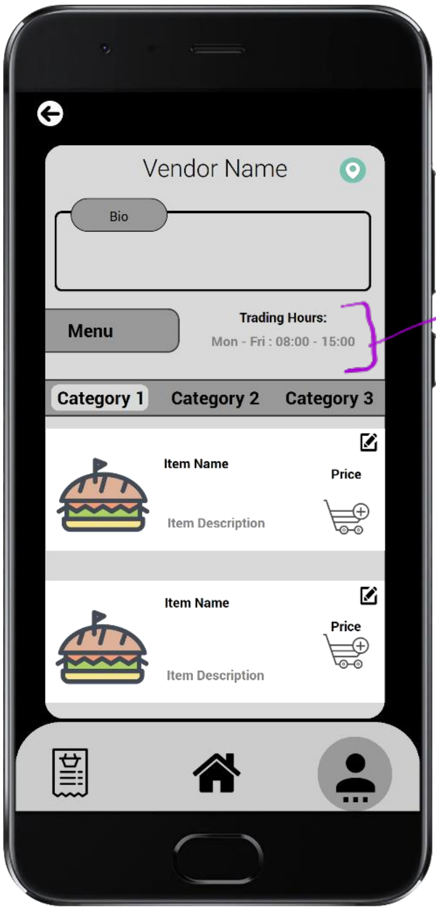


Name :	Customer Name
Surname :	Customer Surname
Email :	Customer Email
Date of Birth :	Customer DoB
CustomerID :	Generated Customer ID
Business Name :	Business Name
Contact Number :	Number
Location :	Location

good integration points

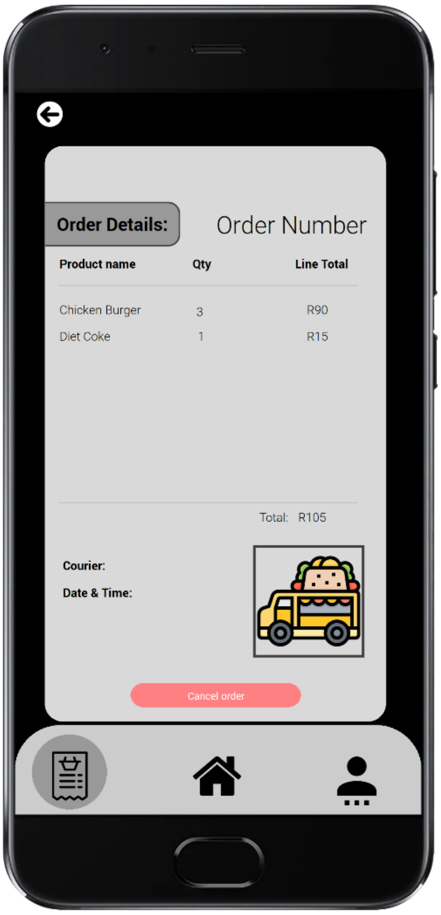


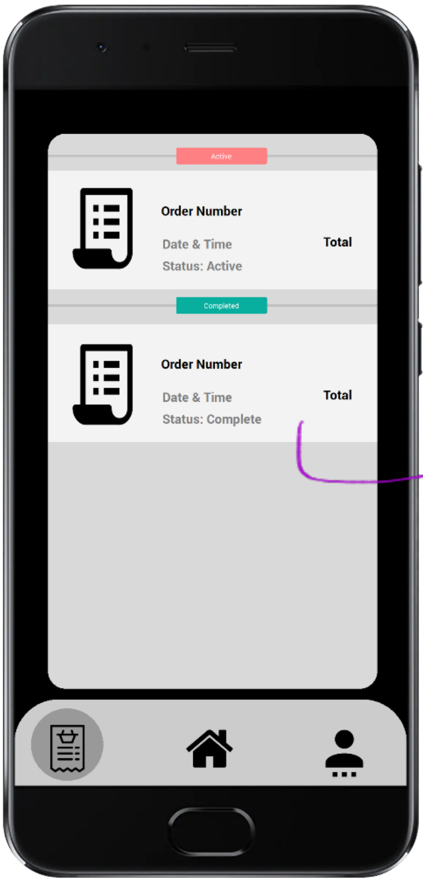


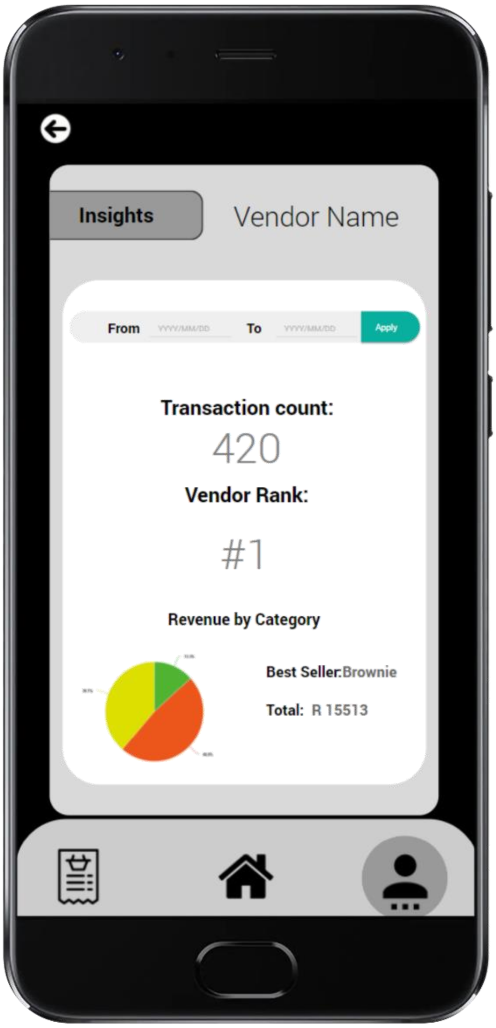
Use Case ID	**Use Case Name**
A0300	Maintain Vendor Menu
Primary Business Actors	**Other participating Actors**
Vendor	
Description	Allows the Vendor to maintain the business menu by allowing the vendor to add and remove items to and from the menu. It also allows for the editing/updating of existing menu items.
Pre-Conditions	The Vendor user must be logged in.
Triggers	The Vendor clicks the Edit Menu button.
Post-Conditions	The vendor's menu is updated in the database so that the updated menu can be displayed on the interface for Vendors and Customers.
Basic Flow of Events	1. The system displays the Update Menu screen. 2. The Vendor adds, removes and/or edits existing food items in the menu. 3. The Vendor clicks the "Save Changes" button/icon when done editing a product. 4. The system displays a message confirming that the menu was successfully updated. 5. The Vendor is returned to the Menu Screen.
Alternate Flow of Events	3. The Vendor clicks the "Cancel" button/icon while editing a product. 4. Return to basic flow step 5.
Initial UI Design	

on previous view you have multiple values, how is this view displayed if i.e. Trading Hrs only Mon, Wed, Fri ?

✓

<table>
<tr><th>Use Case ID</th><th colspan="2">Use Case Name</th></tr>
<tr><td>A0400</td><td colspan="2">View Active Orders</td></tr>
<tr><th colspan="2">Primary Business Actors</th><th>Other participating Actors</th></tr>
<tr><td colspan="2">Vendor</td><td></td></tr>
<tr><th>Description</th><td colspan="2">Allows the Vendor to view incoming orders initiated by the Customer.</td></tr>
<tr><th>Pre-Conditions</th><td colspan="2">A Vendor user must be logged in.</td></tr>
<tr><th>Triggers</th><td colspan="2">Vendor clicks on the “Orders” button or a New Order alert notification in the notification panel. where is this on your design ?</td></tr>
<tr><th>Post-Conditions</th><td colspan="2">The Orders Screen (containing orders placed with the vendor) is displayed.</td></tr>
<tr><th>Basic Flow of Events</th><td colspan="2">1. The system displays all the Vendor’s active orders in a list.</td></tr>
<tr><th>Alternate Flow of Events</th><td colspan="2">1. The system displays all the Vendor’s active orders in a list.
2. The user invokes D0200.</td></tr>
<tr><th>Initial UI Design</th><td colspan="2">
Order Details: Order Number
Product name | Qty | Line Total
Chicken Burger | 3 | R90
Diet Coke | 1 | R15
Total: R105
Courier:
Date &amp; Time:
 </td></tr>
</table>

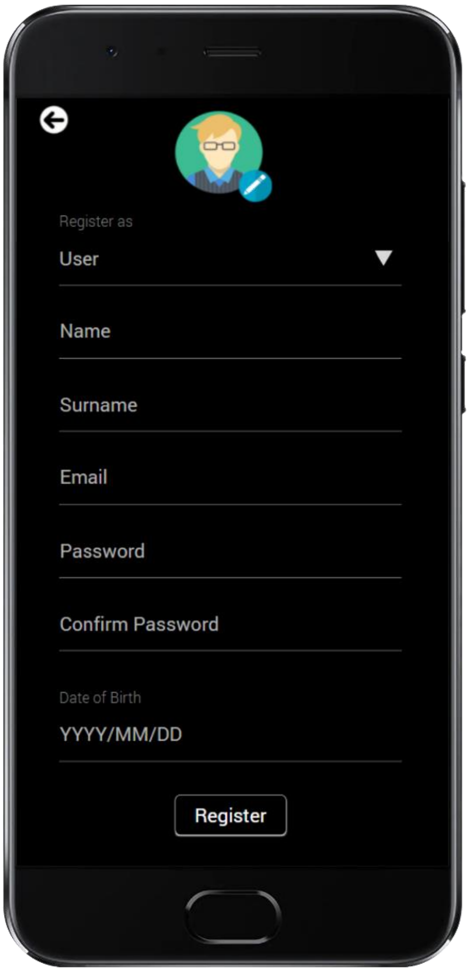
Use Case ID	**Use Case Name**
A0500	View Transaction History
Primary Business Actors	**Other participating Actors**
Vendor	
Description	The Vendor can review previous transactions for book-keeping purposes.
Pre-Conditions	The Vendor must be logged in.
Triggers	The Vendor clicks the "Orders" button.
Post-Conditions	A report compiled by the system on all the transactions that have occurred.
Basic Flow of Events	1. The system displays the transactions that have occurred.
Alternate Flow of Events	1. The system displays the transactions that have occurred. 2. The user invokes D0200
Initial UI Design	 Active Order Number Date & Time Status: Active Total Completed Order Number Date & Time Status: Complete Total allow more filtering here, maybe per month, year

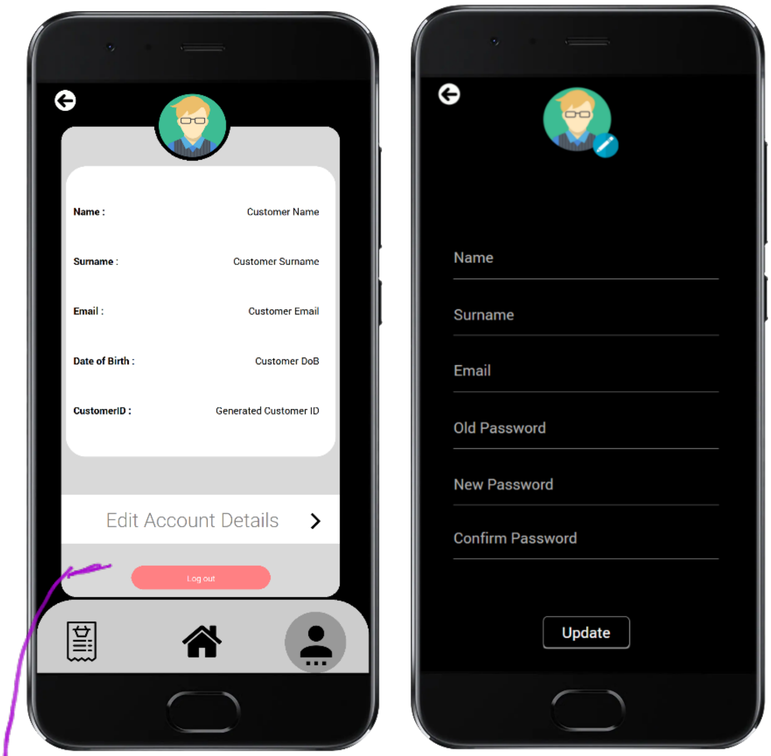
Use Case ID	**Use Case Name**	
A0600	View Transaction Statistics	
Primary Business Actors		**Other participating Actors**
Vendor		
Description	The Vendor can review the statistical performance of the business and determine whether their making profits or losses.	
Pre-Conditions	The Vendor must be logged in.	
Triggers	The Vendor clicks the “Insights” button. ✓	
Post-Conditions	A report compiled by the system on the performance of the business such as the total number of orders, revenue by category and overall rank at different time projections are displayed.	
Basic Flow of Events	1. User enters a “From” date 2. User enters a “To” date 3. User clicks apply 4. The system displays the business transaction statistics of the Vendor based on the date entered.	
Initial UI Design	 Insights Vendor Name From YYYY/MM/DD To YYYY/MM/DD Apply Transaction count: 420 Vendor Rank: #1 Revenue by Category Best Seller: Brownie Total: R 15513	✓

2.3.2 Designed by [redacted]

Use Case ID	Use Case Name
B0100	Register Customer
Primary Business Actors	**Other participating Actors**
Customer	
Description	Customers can setup their accounts. Relevant information such as name, surname, email address and password is stored. The email and password will be required to login to the system as part of the authentication and authorization flow.
Pre-Conditions	The Customer must have installed the system on a device.
Triggers	Customer taps on the application's icon in their application drawer on their device and at the landing screen, clicks the "Register" button.
Post-Conditions	An account is created for the Customer. An identification number is created for the Customer and stored with other Customer information in a database. An email is sent to the Customer congratulating him/her for registering.
Basic Flow of Events	1. Select Customer user type on the Registration screen. 2. System displays the registration screen with fields into which relevant details should be entered. 3. Customer inputs their name, surname, email address and password (twice to verify correctness). 4. Customer taps the "Register" button. 5. System validates entered details. 6. System sends email to Customer containing confirmation code 7. System requests Customer to enter confirmation code. 8. Customer enters confirmation code. 9. The system displays a message informing the Customer of successfully registering. 10. Home Screen is displayed
Alternate Flow of Events	4. Customer taps the "Cancel" button. 5. System returns to the landing screen.
Alternate Flow of Events	6. System informs user that details provided are invalid or insufficient. 7. Return to basic flow step 2.
Alternate Flow of Events	9. The system displays error message due to incorrect confirmation code. 10. Return to basic flow step 7.

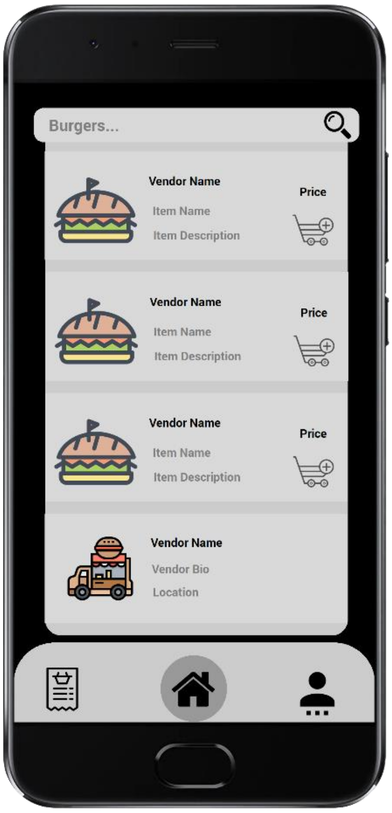
See comment in A100

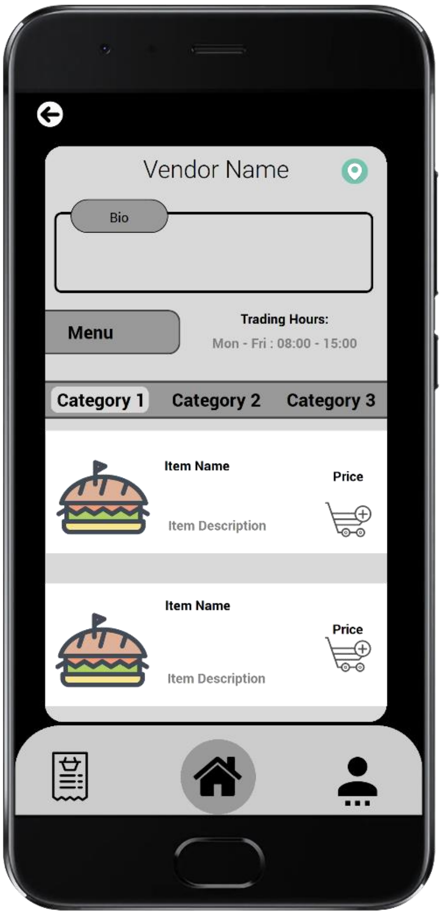
Initial UI Design	

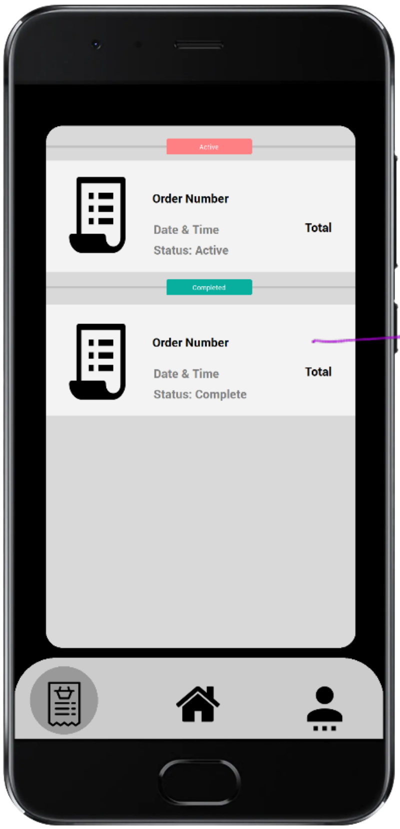
Use Case ID	Use Case Name
B0200	Maintain Customer Details
Primary Business Actors	**Other participating Actors**
Customer	
Description	Allows the customer to maintain customer related details by navigating to a screen where the customer can manage the customer's details.
Pre-Conditions	A customer user must be logged in.
Triggers	If the customer details are insufficient or if the customer selects an "Edit Details" button on the interface.
Post-Conditions	The Customer's details are updated in the database. The Customer is returned to his account screen.
Basic Flow of Events	1. The system displays the Update Customer Details screen. 2. The customer enters the necessary details in the fields provided. 3. The Customer clicks the "Update" button 4. The system validates the entered details 5. The system displays a message confirming that the details were successfully updated. 6. The Customer is returned to the Account Screen.
Alternate Flow of Events	3. The Customer clicks the "Cancel" button. 4. Return to basic flow step 6.
Alternate Flow of Events	5. The system displays a message informing the user that entered details are not valid or that the changes could not be saved. 6. Return to basic flow step 1.
Initial UI Design	 Name : Customer Name Surname : Customer Surname Email : Customer Email Date of Birth : Customer DoB CustomerID : Generated Customer ID Edit Account Details Log out Name Surname Email Old Password New Password Confirm Password Update

✓

maybe add in link to View Order History (B500)

Use Case ID	Use Case Name
B0300	Search Vendors and Products
Primary Business Actors	**Other participating Actors**
Customer	
Description	Allows the Customer to perform a look-up on the Vendors that are registered on the system and view the products that each Vendor offers.
Pre-Conditions	The Customer must be logged in.
Triggers	The Customer taps on the search icon after typing in a keyword in the search panel.
Post-Conditions	The system displays search results consisting of Vendors and/or Products matching the search criteria.
Basic Flow of Events	1. The system builds a list consisting of all products and vendors matching the search criteria. 2. The systems displays a list to the user containing search results.
Alternate Flow of Events	3. Option to call B0400
Alternate Flow of Events	3. The system displays message informing the user of no results found.
Initial UI Design	 Burgers... Vendor Name Item Name Item Description Price Vendor Name Item Name Item Description Price Vendor Name Item Name Item Description Price Vendor Name Vendor Bio Location

Use Case ID	**Use Case Name**
B0400	View Vendor
Primary Business Actors	**Other participating Actors**
Customer	Vendor
Description	A list of Vendors displayed on the Home Screen be interacted with by means of tapping on the tiles in the list to then view a screen containing information about the vendor and the products sold by the vendor.
Pre-Conditions	The Customer must be logged in.
Triggers	The Customer taps on a Vendor on the screen or invoked from B0300
Post-Conditions	The Vendor's business information and menu is displayed.
Basic Flow of Events	1. The system retrieves all data pertaining to the Vendor and the Vendor's catalog.
Alternate Flow of Events	2. User invokes D0200
Initial UI Design	 Vendor Name Bio Menu Trading Hours: Mon - Fri : 08:00 - 15:00 Category 1 Category 2 Category 3 Item Name Price Item Description Item Name Price Item Description

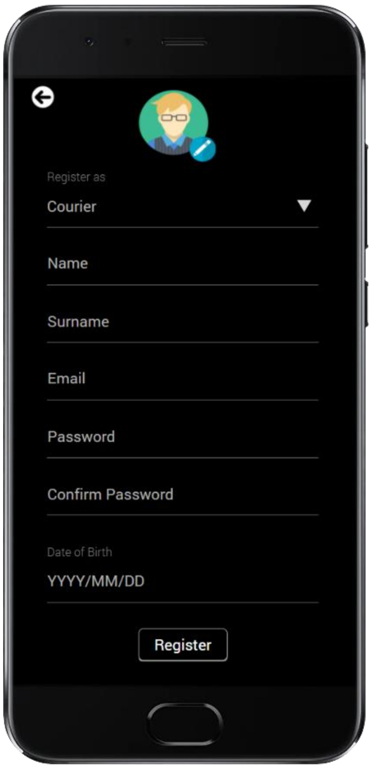
Use Case ID	**Use Case Name**
B0500	View Order History
Primary Business Actors	**Other participating Actors**
Customer	
Description	Allows Customer to view previously placed orders and the details pertaining to them.
Pre-Conditions	The customer must be logged in.
Triggers	The customer taps “Orders” button on the interface
Post-Conditions	A list of the orders previously made by the customer is displayed.
Basic Flow of Events	1. The system retrieves all the data relating to orders placed by the Customer. 2. The system builds and displays a list of orders made by the Customer.
Alternate Flow of Events	3. User invokes D0200
Initial UI Design	

Filter on vendor, dates as well

2.3.3 Designed by [redacted]

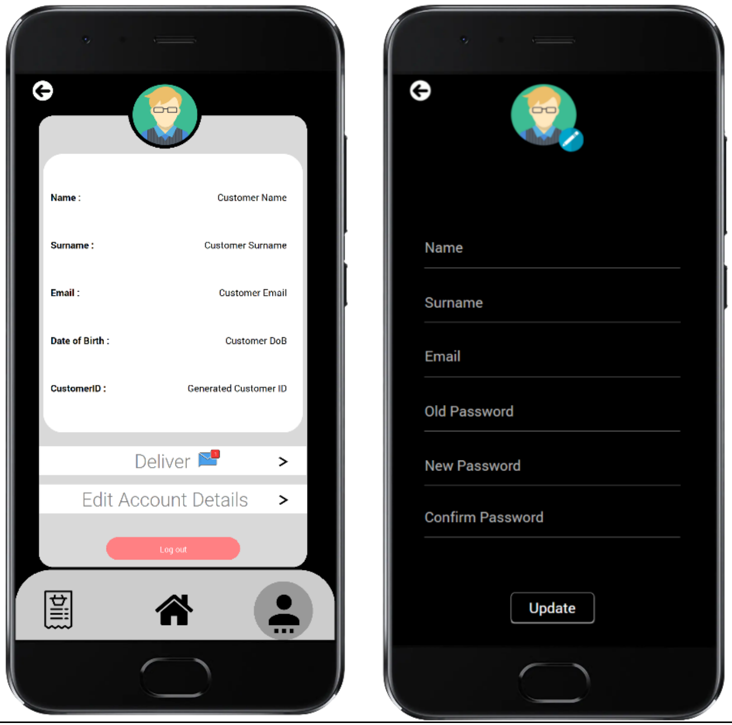
Use Case ID	**Use Case Name**
C0100	Register Courier
Primary Business Actors	**Other participating Actors**
Courier	
Description	A Courier can setup an account. All information such as name, surname, email address and password are stored. The email and password will be required to login to the system as part of the authentication and authorization flow.
Pre-Conditions	The Courier must have installed the system on their device.
Triggers	The Courier taps on the application icon in their application drawer on a device and clicks on the "Register" button on the landing page.
Post-Conditions	A Courier account is created (details saved/stored) and can be used to access the system.
Basic Flow of Events	1. Select Courier user type on the Registration screen. 2. System displays the registration screen with fields into which relevant details should be entered. 3. The Courier inputs their name, surname, email address and password (twice to ensure it has been inputted correctly). 4. Courier taps the "Register" button. 5. System validates entered details. 6. System sends email to Courier containing confirmation code. 7. System requests Courier to enter confirmation code. 8. Courier enters confirmation code. 9. The system displays a message informing the Courier of successfully registering. 10. Home Screen is displayed.
Alternate Flow of Events	4. Courier taps the "Cancel" button. 5. System returns to the landing screen.
Alternate Flow of Events	6. System informs user that details provided are invalid or insufficient. 7. Return to basic flow step 2.
Alternate Flow of Events	9. The system displays error message due to incorrect confirmation code. 10. Return to basic flow step 7.

same comment as A100

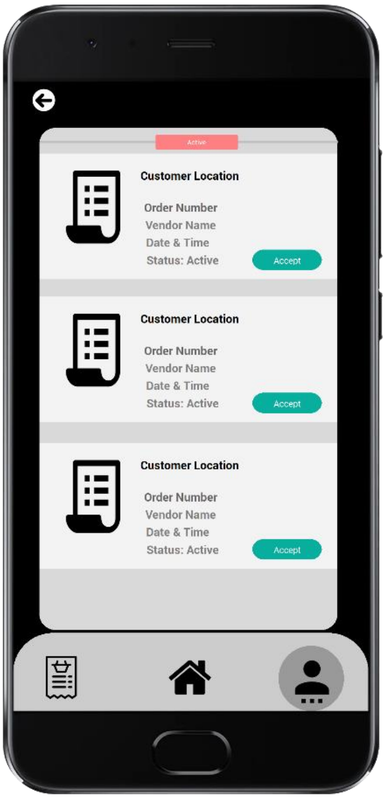
Initial UI Design	

Register as: Courier

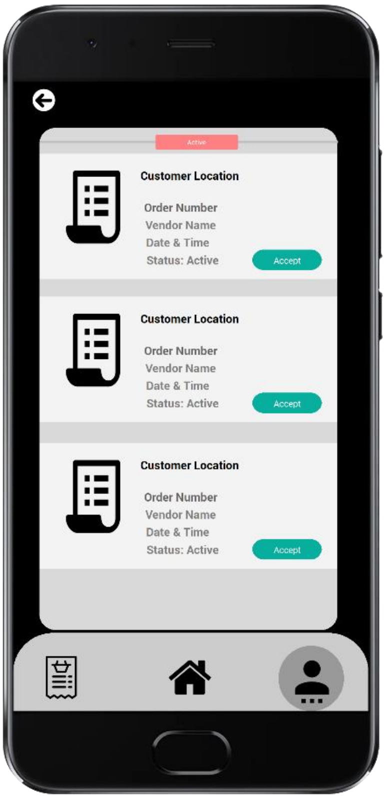
Use Case ID	**Use Case Name**
C0200	Maintain Courier Details
Primary Business Actors	**Other participating Actors**
Courier	
Description	Allows the Courier to maintain courier related details by navigating to a screen where the courier can manage the courier's details.
Pre-Conditions	The Courier must be logged in.
Triggers	If the Courier details are insufficient or if the courier selects an "Edit Details" button on the interface.
Post-Conditions	The Courier's details are updated in the database and updated details are displayed in relevant areas of the system.
Basic Flow of Events	1. The system displays the Update Courier Details screen. 2. The Courier enters the necessary details in the fields provided. 3. The Courier clicks the "Update" button 4. The system validates the entered details 5. The system displays a message confirming that the details were successfully updated. 6. The Courier is returned to the Account Screen.
Alternate Flow of Events	3. The Courier clicks the "Cancel" button. 4. Return to basic flow step 6.
Alternate Flow of Events	5. The system displays a message informing the user that entered details are not valid or that the changes could not be saved. 6. Return to basic flow step 1.

Initial UI Design	 Name : Customer Name Surname : Customer Surname Email : Customer Email Date of Birth : Customer DoB CustomerID : Generated Customer ID Deliver > Edit Account Details > Log out Name Surname Email Old Password New Password Confirm Password Update

Use Case ID	**Use Case Name**
C0300	View Delivery
Primary Business Actors	**Other participating Actors**
Courier	
Description	Allows Courier to view information pertaining to a delivery. The delivery could be one found in a list of available delivery jobs, one found in the Courier’s list of accepted/active deliveries or one from the Courier’s delivery history.
Pre-Conditions	The Courier must be logged in.
Triggers	The Courier taps on a delivery tile in a list.
Post-Conditions	Details regarding a delivery are displayed. If the delivery is an available one, an “Accept Delivery” button is also displayed. If the delivery is an already accepted one, depending of the delivery status, either a “Confirm Pick-up” or “Confirm Delivery” button is displayed.
Basic Flow of Events	1. The system retrieves data regarding the selected delivery. 2. The system displays the delivery information on the screen. 3. The system displays an “Accept Delivery” button. 4. Calls C0310
Alternate Flow of Events	3. The system displays a “Confirm Pick-up” button. 4. Calls C0320
Alternate Flow of Events	3. The system displays a “Confirm Delivery” button. 4. Calls C0330

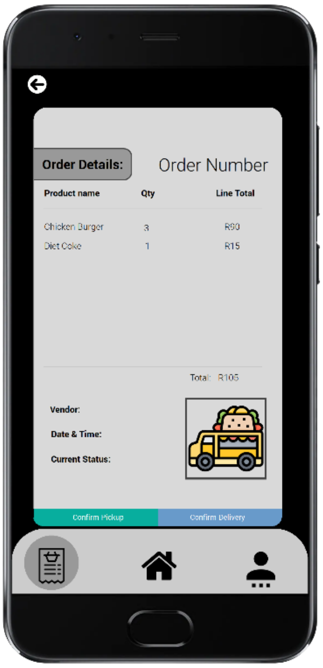
Initial UI Design	 Active Customer Location Order Number Vendor Name Date & Time Status: Active Accept Customer Location Order Number Vendor Name Date & Time Status: Active Accept Customer Location Order Number Vendor Name Date & Time Status: Active Accept ✓

Use Case ID	**Use Case Name**
C0310	Accept Delivery
Primary Business Actors	**Other participating Actors**
Courier	
Description	Allows the Courier to indicate availability to collect and deliver the order placed by the Customer from the Vendor.
Pre-Conditions	The Courier must be logged in and viewing an available delivery job.
Triggers	The Courier taps on “Accept Delivery” button (Invoked from C0300)
Post-Conditions	The Vendor and Customer are made aware of the Courier’s availability status. The pending order is placed with the Vendor adding it to the Vendor’s active orders and the delivery is added to the Courier’s list of active deliveries.
Basic Flow of Events	1. The system requests the Courier to confirm acceptance. 2. The Courier confirms acceptance. 3. The system adds the order to the Customer’s order history. 4. The system calls E0100 to send a notification to the Customer. 5. The system adds the order to the Vendor’s active orders. 6. The system calls E0100 to send a notification to the Vendor. 7. The system directs the Courier to the Active Deliveries screen.

Initial UI Design	

✓

Use Case ID	**Use Case Name**	
C0320	Confirm Pickup	
Primary Business Actors		**Other participating Actors**
Courier		
Description	Allows the Vendor and Customer to be made aware that the Courier has collected the order.	
Pre-Conditions	The Courier must be logged in and viewing an active delivery that has not been picked up yet.	
Triggers	The Courier taps the “Confirm Pickup” button. (Invoked from C0300)	
Post-Conditions	Vendor and Customer are made aware that the order is on its way by a notification. Delivery status is updated to reflect pickup	
Basic Flow of Events	1. The system requests the Courier to confirm pickup. 2. The Courier confirms pickup. 3. The system updates the delivery status of the order to “collected”. 4. The system updates to order status to “complete” 5. The system removes the order from the Vendor’s active order list. 6. The system calls E0100 to send notification to the Vendor. 7. The system calls E0100 to send notification to the Customer. 8. The system directs the Courier to the Active Deliveries screen.	

Initial UI Design	 ✓

Order Details: Order Number

Product name	Qty	Line Total
Chicken Burger	3	R90
Diet Coke	1	R15

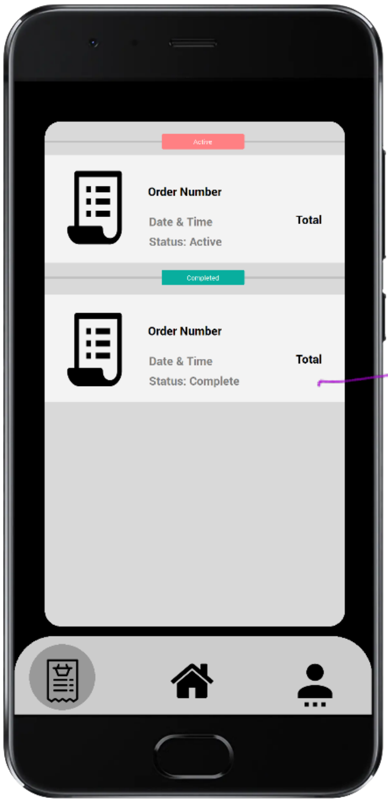
Total: R105

Vendor:

Date & Time:

Current Status:

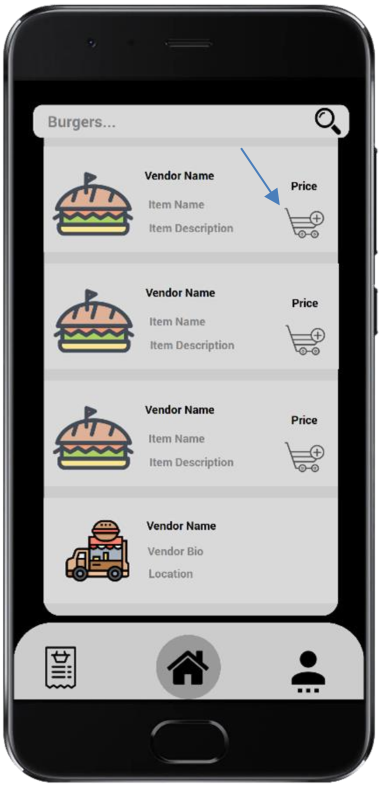
Use Case ID	**Use Case Name**
C0400	View Delivery History
Primary Business Actors	**Other participating Actors**
Courier	
Description	Allows the Courier to view a complied report consisting of the past deliveries made by the courier.
Pre-Conditions	The Courier must be logged in.
Triggers	The Courier taps “View Delivery History” button.
Post-Conditions	The system displays a list of past deliveries made by the Courier.
Basic Flow of Events	1. The system displays a list of past deliveries made by the courier.
Alternate Flow of Events	2. User invokes C0300

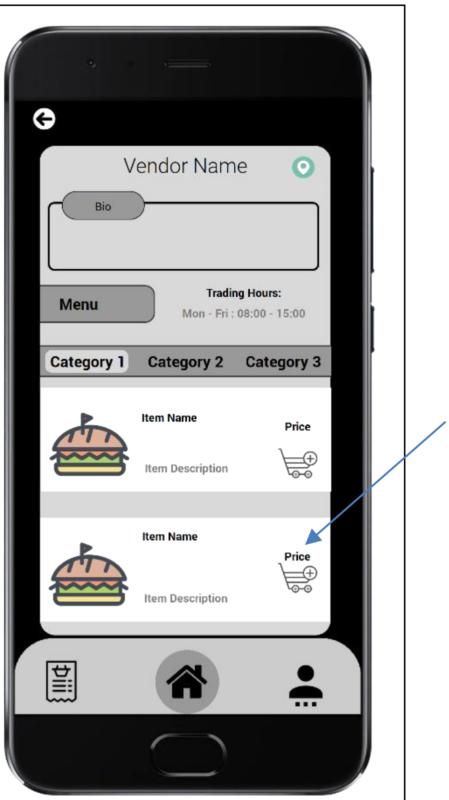
Initial UI Design	 Active Order Number Date & Time Status: Active Total Completed Order Number Date & Time Status: Complete Total more Filtering options

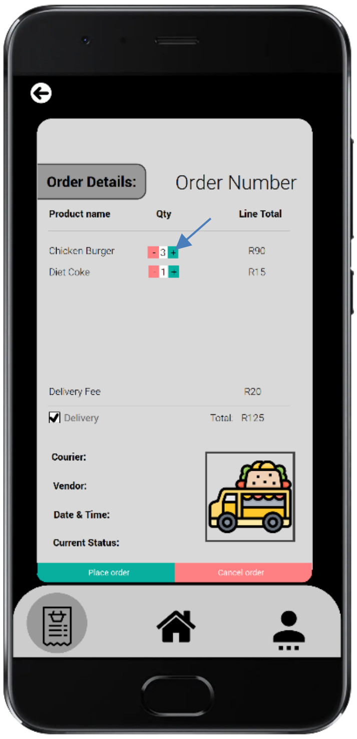
2.3.4 Designed by All Team Members

Use Case ID	**Use Case Name**
D0100	Add To Order
Primary Business Actors	**Other participating Actors**
Customer	
Description	Allows the customer to add a menu item (from the Vendor they are currently viewing) to the current order.
Pre-Conditions	The customer must be logged in and viewing the vendor or viewing their current order.
Triggers	The customer taps the “Addition” icon next the menu item in the menu (list of products) or the order line of the order they’re viewing. (Invoked from B0400 or B0300)
Post-Conditions	An order is created if one does not exist already and the menu item is added to the order.
Basic Flow of Events	1. The menu item is added to the order on a new line. 2. Order details are updated. 3. Notify user of order change.
Alternate Flow of Events	1. An order is created. 2. Return to basic flow step 1.
Alternate Flow of Events	1. The quantity of menu item on the order is updated. 2. Return to basic flow step 2.

Initial UI Design





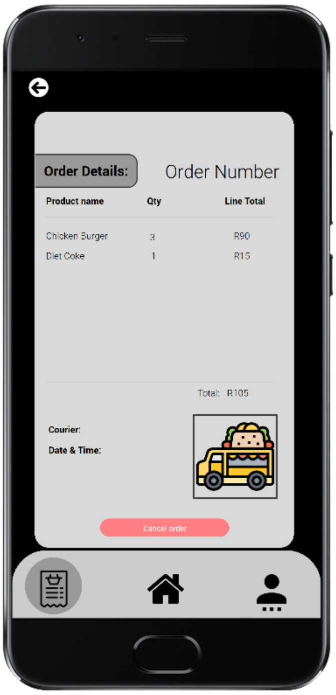
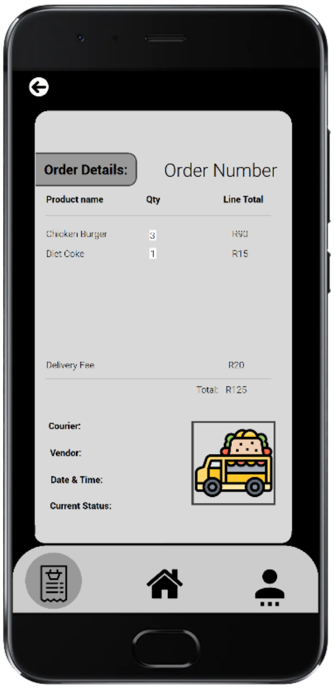


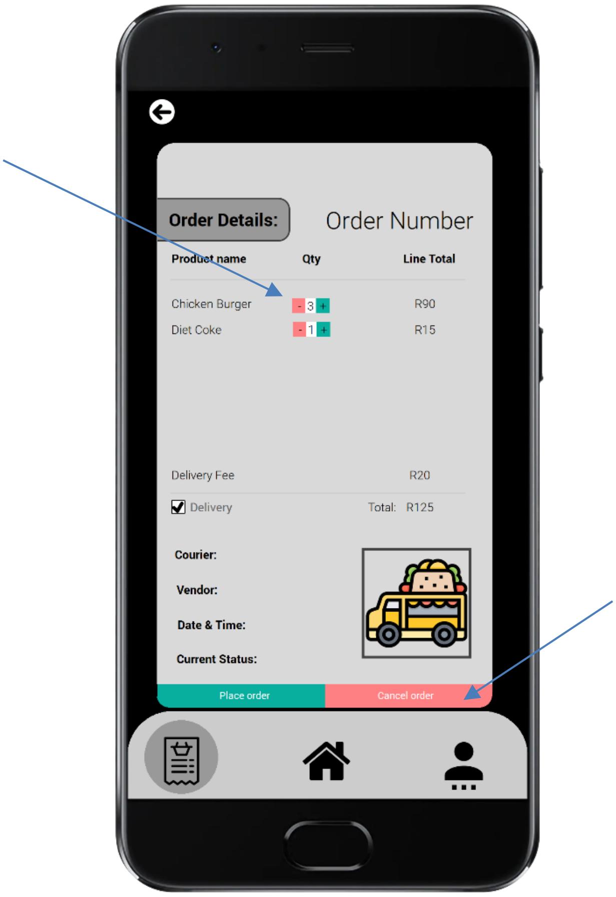
Product name	Qty	Line Total
Chicken Burger	3	R90
Diet Coke	1	R15

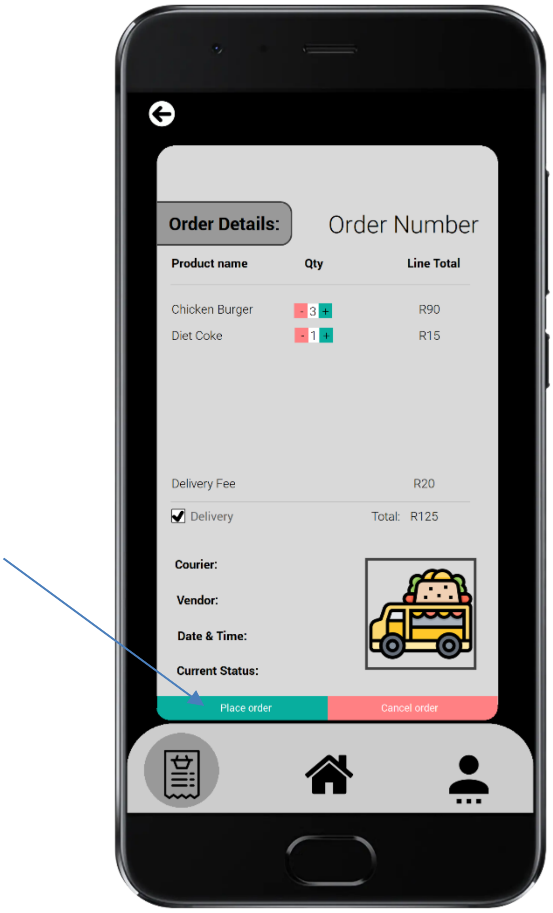
Delivery Fee R20

☑ Delivery Total: R125

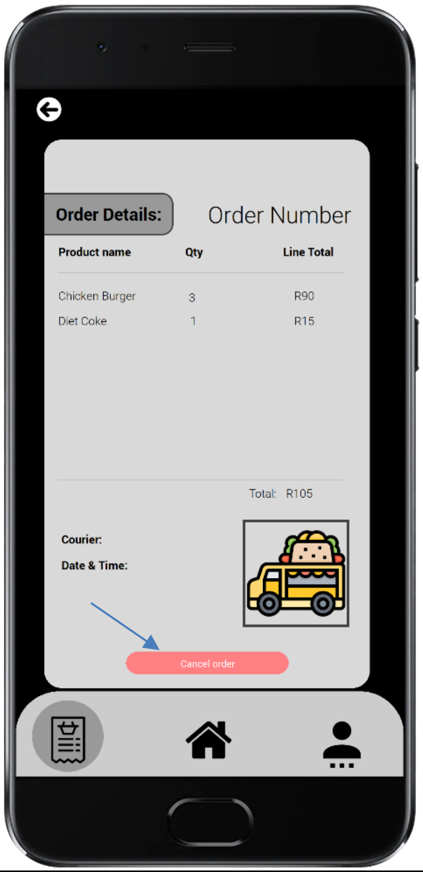
✓

Use Case ID	Use Case Name
D0200	View Order
Primary Business Actors	**Other participating Actors**
Customer, Vendor	
Description	Allows the customer to view information about an order that they are actively building or one that appears in their order history. Allows the vendor to view information about an active (i.e. placed) order or one that appears in their transaction history.
Pre-Conditions	The Customer must be logged and be in the process of building an order from a vendor they are viewing or the customer must be viewing orders in their order history. The Vendor must be logged in and viewing their active orders or transaction history.
Triggers	The customer taps “View Order” button to view the order they are actively building or the customer simply taps on one of the orders in the list of orders in their order history. The vendor taps on one of the orders in their list of active orders or the list of orders in their transaction history. (Invoked from A0400, A0500 or B0500)
Post-Conditions	The system displays order information along with appropriate controls depending on the type of user.
Basic Flow of Events	1. Menu items on order are displayed in the form of lines containing info such as the name of the menu item, the product code, the quantity and the line total. 2. The order total is calculated. 3. The order total is displayed.
Alternate Flow of Events	4. User Invokes D0220
Alternate Flow of Events	5. User Invokes D0210
Initial UI Design	 Order Details: Order Number Product name - Qty - Line Total Chicken Burger - 3 - R90 Diet Coke - 1 - R15 Total: R105 Courier: Date & Time: *Figure 2:Vendor view*  Order Details: Order Number Product name - Qty - Line Total Chicken Burger - 3 - R90 Diet Coke - 1 - R15 Delivery Fee - R20 Total: R125 Courier: Vendor: Date & Time: Current Status: *Figure 1: Customer view after order is placed*

Use Case ID	Use Case Name
D0210	Remove From Order
Primary Business Actors	**Other participating Actors**
Customer	
Description	Allows the customer to discard a food item from the order.
Pre-Conditions	The customer must be logged in and have an active order which they are viewing. This order must have at least one menu item/product.
Triggers	The customer taps a “-” icon next to the menu item of active order being currently viewed (Invoked from D0200)
Post-Conditions	A menu item/product is removed from the order and if the order is “empty”, the order is no longer displayed in the order list.
Basic Flow of Events	1. The order line pertaining to the removed product is removed. 2. The order total is updated and displayed.
Alternate Flow of Events	1. The quantity of involved order line is decremented by one. 2. The involved order line total is updated and displayed. 3. Return to basic flow step 2.
Alternate Flow of Events	2. The order is removed from the orders list. 3. User returned to the Home screen.
Initial UI Design	 **Order Details:** Order Number **Product name** **Qty** **Line Total** Chicken Burger - 3 + R90 Diet Coke - 1 + R15 Delivery Fee R20 ☑ Delivery Total: R125 **Courier:** **Vendor:** **Date & Time:** **Current Status:**

Use Case ID	Use Case Name
D0220	Place Order
Primary Business Actors	**Other participating Actors**
Customer	
Description	Allows the customer to finalize the order. The customer can choose whether the order should be delivered to a specific location or if the order will be collected in person.
Pre-Conditions	The customer must be logged in and viewing an order they are actively building.
Triggers	The customer taps a “Place Order” button on the Order view (Invoked from D0200)
Post-Conditions	The vendor is alerted to begin preparing the order if the order is to be collected by the user. If the order is to be delivered, nearby couriers are notified of the request.
Basic Flow of Events	1. The Customer selects to collect the order in person. 2. The order is added to the vendor’s active order list. 3. Calls E0100 to send notification to vendor. 4. The order is added to the customer’s order history. 5. Calls E0100 to send notification to customer. 6. Returns customer to the vendor screen.
Alternate Flow of Events	1. The Customer selects to have the order delivered. 2. Calls E0100 to send a delivery request notification to nearby couriers. 3. Return to basic flow step 6.
Initial UI Design	 Order Details: Order Number Product name / Qty / Line Total Chicken Burger - 3 + R90 Diet Coke - 1 + R15 Delivery Fee R20 ☑ Delivery Total: R125 Courier: Vendor: Date & Time: Current Status: ✓

Use Case ID	Use Case Name
D0230	Cancel Order
Primary Business Actors	**Other participating Actors**
Vendor	Time
Description	Allows the Vendor to cancel or decline an order made by a customer or allows the system to cancel an overdue order that was never completed.
Pre-Conditions	The vendor must be logged in and have an active order which they are viewing from the Active Orders Screen or the system must have an active order.
Triggers	The vendor taps a “Cancel Order” button or the system determines that an order is overdue by comparing timestamps.(Invoked from A0400)
Post-Conditions	The order is cancelled and the order status has been set to “cancelled” The order will no longer be displayed on the vendors active order list in the Active Orders screen. Relevant users will have been received a notification informing them of the cancellation. If the vendor cancelled the order, the vendor will have been returned to the Active Orders screen.
Basic Flow of Events	1. The system prompts the user to confirm the cancellation. 2. The user taps the “Yes” button to confirm the cancellation. 3. The system sets the order’s status to “cancelled”. 4. Calls E0100 5. The vendor is returned to the Active Orders screen.
Alternate Flow of Events	1. The system cancels the order and sets the order status to “cancelled”. 2. Return to basic flow step 5.
Alternate Flow of Events	3. The user taps the “No” button to cancel the cancellation. 4. Return to basic flow step 5.

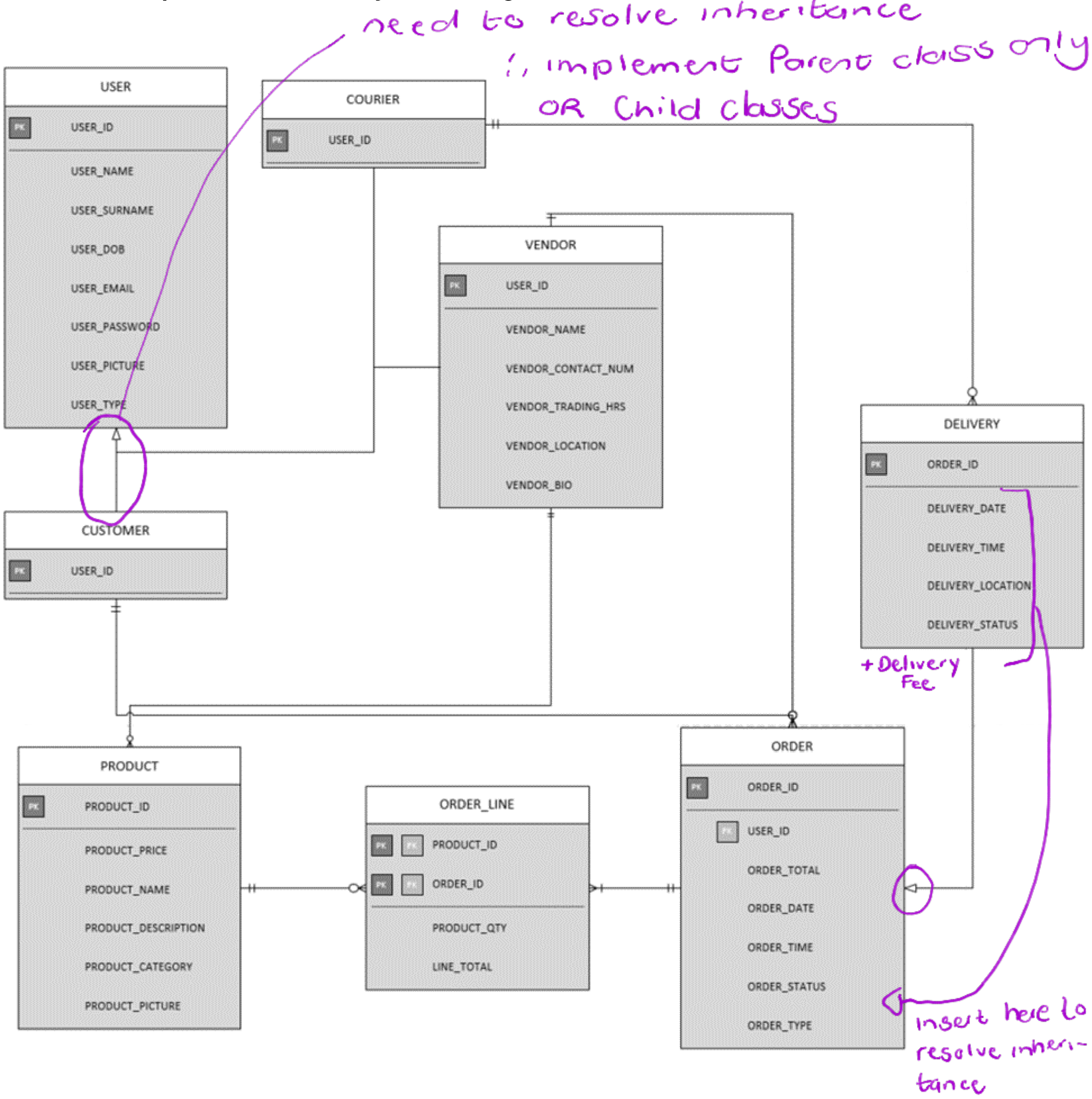
Initial UI Design	 Order Details: Order Number Product name / Qty / Line Total Chicken Burger / 3 / R90 Diet Coke / 1 / R15 Total: R105 Courier: Date & Time: Cancel order ✓

Use Case ID	Use Case Name
E0100	Send Notification
Primary Business Actors	**Other participating Actors**
State	
Description	The system alerts its user(s) when an action has been performed requiring relevant user to be informed thereof.
Pre-Conditions	
Triggers	The system cancels an overdue order, vendor manually cancels an order, customer places an order or courier updates delivery status. (Invoked from D0230, D0220, C0310, C0320 or C0330)
Post-Conditions	Relevant users receive the notification
Basic Flow of Events	1. Notification content is generated. 2. Notification is sent to relevant users.

Where would you see the notification?

3 DATA REQUIREMENTS

Implementation Ready Class Diagram



need to resolve inheritance
∴ implement Parent class only
OR Child classes

+ Delivery Fee

insert here to resolve inheritance

Department of Computing Sciences

Plagiarism Declaration - Elaboration Document

Module code: WRRV301

We,

Name	Student Number	Contribution % [Total = 100]
[redacted]	[redacted]	33.33%
[redacted]	[redacted]	33.33%
[redacted]	[redacted]	33.33%

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 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 indiciated in the contribution % above.

[redacted]