**COMP 3059 – Capstone Project I**

**Software Requirements Analysis and Design Assignment**

Table of Contents

[**1.0 Introduction** 2](#_Toc118812413)

[**1.1 Purpose** 3](#_Toc118812414)

[**1.2 Scope** 3](#_Toc118812415)

[**2.0 System Overview** 2](#_Toc118812416)

[**2.1 Project Perspective** 2](#_Toc118812417)

[**2.2 System Context** 2](#_Toc118812418)

[**2.3 General Constraints** 2](#_Toc118812419)

[**2.4 Assumptions and Dependencies** 2](#_Toc118812420)

[**3.0 Functional Requirements** 3](#_Toc118812421)

[**3.1 Features** 3](#_Toc118812422)

[**3.2 Use Cases** 10](#_Toc118812423)

[**3.2.1 Use Case #1** 10](#_Toc118812424)

[**3.2.2 Use Case #2** 11](#_Toc118812425)

[**3.2.3 Use Case #3** 11](#_Toc118812426)

[**3.2.4 Use Case #4** 12](#_Toc118812427)

[**3.2.5 Use Case #5** 12](#_Toc118812428)

[**3.2.6 Use Case #6** 13](#_Toc118812429)

[**3.2.7 Use Case #7** 13](#_Toc118812430)

[**3.2.8 Use Case #8** 14](#_Toc118812431)

[**3.2.9 Use Case #9** 14](#_Toc118812432)

[**3.2.10 Use Case #10** 15](#_Toc118812433)

[**4.0 Non-Functional Requirements** 24](#_Toc118812434)

[**5.0 Logical Database Requirements** 25](#_Toc118812435)

[**6.0 Other Requirements** 29](#_Toc118812444)

# **1.0 Introduction**

## **1.1 Purpose**

Goodie Foodie is a mobile application built for users with an interest in cooking. This application allows users to share their cooking experience whether they are a complete beginner or a pro. There are several functions within this application, including viewing other posts, tracking their ingredients’ expiration dates, and buying groceries online.

This application also allows grocery store suppliers and delivery staff to be users. Suppliers will be notified if their inventory is running low and to view orders. The delivery staff will also be able to see the orders they are to deliver.

## **1.2 Scope**

Goodie Foodie Mobile Application allows users to search or view recipes. Users can also create, edit, view, and delete posts. They can give comments or likes. The application can keep track of the expiration date for ingredients users had input. It should notify the user of an ingredient near its expiration date.

Users can also do online grocery shopping. Once the order is placed, users can see the status of their order.

Suppliers should also be notified if their inventory is low, but whether the suppliers can restock their ingredients in a timely manner is not within our control.

Our objective is to make cooking enjoyable for everyone, whether they are a beginner or not with the functionalities our app can provide.

# **2.0 System Overview**

## **2.1 Project Perspective**

Goodie Foodie (GF) is a replacement for existing systems, and bunch of cooking apps in the market. This app compiles all its recipe videos, guides, posts and more ways to assist home cooks and develop their skills. GF has functions that are unique which are tracking food expiration dates and online grocery shopping.

## **2.2 System Context**

With the development approach, we have to consider the nature of business and future to decide if it is just web based mobile app development. So, with this, we need to strategize the UI needs. For Device compatibility and screen size, every business wishes to give a massive first impression to their end users. Device compatibility is probably the most important factor for a mobile application. With flawless service and compatible on multiple devices. OS selection is very important. Apple and Android require two different development procedures.

## **2.3 General Constraints**

* There may be many other similar applications out there and people may not want to use our app.
* Users may find only one or a few functions that are useful and may look for other apps to replace ours.
* Users may not find the UI to their liking, and we may have to redesign it.
* Users may not use our app because our service is not faster than other competitors

## **2.4 Assumptions and Dependencies**

* The project scope will not change once the stakeholders sign off on the statement.
* All the members should have completed their respective tasks according to the schedule in order for the project to be finished by the expected deadline.
* Resource assumptions include necessary resources such as people and materials that have a significant impact on the failure or success of a project.
* We should be performing unit testing as frequently as possible.

# **3.0 Functional Requirements**

## **3.1 Features**

The features of this software project will be discussed in 3 parts, general users, grocery stores suppliers and delivery people.

This system will allow anyone to register as a user. Users can login, search recipes, share posts, leave comments, like posts, receive ingredients’ expiration date notification, send feedback, register to become a delivery person, shop groceries online etc.

As for the grocery stores suppliers, this system allows them to login as a supplier, manage sell items, categorize items, view orders placed by users, and view and modify sell items’ inventory.

Finally, delivery people can see the pick-up and drop-off addresses of new orders, accept orders, receive money once they have delivered the items to users.

**3.1.1 Feature Table #1**

|  |  |  |  |
| --- | --- | --- | --- |
| User Feature | Input | Processing | Output |
| Register | Enter personal information, accept the terms and policies and click register button. | System will check if the same phone number exist from the database. If not, a new user will be added to the database. | If phone number already exist, a prompt message “This phone has been registered” will pop out. Otherwise, will be directed to the login page. |
| Login | Enter username and password and click login button. | System will check if user with the same username and password exists from the database. | If user exist, then will be directed to the home page of this application. Otherwise, a prompt message “Wrong username and password” will pop out. |
| Search Recipe | Enter an ingredient, recipe name or recipe type (ex: cuisine type) and click search button. | System will select all the recipes that match the search requirement from the database. | A list of recipes will be displayed. |
| View shared posts | Click on post’s home button | System will select all the publicly shared posts from the database. | A list of posts will be displayed. |
| View posts created by user itself | Click on “My posts” button. | System will select all the posts created by user from the database. | A list of posts will be displayed. |
| Create posts | Click add new post button, then enter or upload contents and click save button. | System will add a new post to the database. | System will redirect to “My posts” page where the newest post will be displayed as well. |
| Edit posts | Click the edit button of the selected post, and then modify contents and click save button. | System will first select the post from the database, then display the contents and save the modified contents to the database. | System will redirect to “My posts” page where the modified post will be displayed with the changes. |
| Delete posts | Click the delete button of the selected post and confirm deletion. | System will delete the post from the database. | System will redirect to “My posts” page and the deleted post will no longer be displayed. |
| Share / Hide posts | Change the view setting to public / private. (When a new post is created the default is public.) | System will first select the post from the database, then save the modified view setting to the database. | System will redirect to “My posts” page where the modified post will be displayed with different view setting. |
| Leave comments | Click the “leave a comment” button then enter contents and click submit button. | System will add the new comment to the database. | System will display contents of the post along with the new comment. |
| Like posts | Click the like button of shared posts. | System will add new post like to the database. | System will display the current number of likes of the post. |
| View all ingredients’ expiration date list | Click on expiration date’s home button. | System will select all ingredient expiration dates made by user from the database. | A list of ingredients’ expiration date will be displayed in table format. |
| Receive ingredients’ expiration date notification | Click add ingredient expiration date button and enter an ingredient and expiration date. Then set the preferred notification time (ex: 3 days prior to expiry) and click save button. | System will add the new ingredient expiration date to the database. And automatically send notification when the time comes. | Receive a notification of the expiry date of the ingredient. |
| Edit ingredients’ expiration date | Click edit button of selected ingredient expiration date and modify expiration date or preferred notification time and click save button. | System will first select the ingredient expiration date from the database, then display the contents and save the modified contents to the database. | System will redirect to ingredients’ expiration date page where the modified ingredient expiration date will be displayed with the changes. |
| Delete ingredients’ expiration date | Click delete button of selected ingredient expiration date and confirm deletion. | System will delete the ingredient expiration date from the database. | System will redirect to ingredients’ expiration date page and the deleted ingredient expiration date will no longer be displayed. |
| Send feedback | Click on send feedback button and enter contents then click submit button. | System will send feedback to customer service email. | System will generate a prompt about feedback being received message to current page. |
| Register as a delivery person | Click “become a delivery partner” button and enter the direct deposit information. Then accept the terms and policies and click submit. | System will add user to the delivery person table in the database. | A delivery person home button (more features) will be added to the software. |
| View all online grocery stores | Click on shop home button. | System will select all grocery stores from the database. | A list of grocery stores will be displayed. |
| View items in online grocery stores | Click the selected grocery store. | System will select all items of selected grocery store from the database. | A list of items will be displayed. |
| Search items in online grocery stores | Enter item name and click search button. | System will select all the items that match the search requirement from the database. | A list of items will be displayed. |
| View items in shopping cart | Click shopping cart icon. | System will select all the items that are stored inside the cart object. | A list of items will be displayed. |
| Add items to shopping cart | Enter the quantity and click add to cart button of selected item. | System will add new item and quantity to the cart object. | N/A |
| Edit items in shopping cart | Modify item quantity of selected item. | System will modify the quantity of the selected item inside the cart object. | A list of items with any modification will be displayed. |
| Remove items from shopping cart | Click delete button of selected item. | System will delete item and quantity from the cart object. | A list of items without the deleted item will be displayed. |
| Checkout shopping cart | Click checkout button. | System will display checkout page where user must enter necessary information. | System will display checkout page for user to enter delivery address and payment method. |
| Make online payment | Enter payment method and click place order button. | System will process online payment, if successful then a new order will be created and added to the database. | If payment is successful, the system will redirect to order page. Otherwise, a prompt of “Transaction Declined” will pop out. |
| Receive electronic receipts | N/A | If payment is successful, the system will generate and send electronic receipt to user’s email. | User receiving electronic receipt email from the application. |
| View all orders | Click order’s home button. | System will select all orders placed by the user from the database. | A list of orders will be displayed. |
| Track order delivery status | Click on the desired order. | System will select the selected order placed by the user from the database and display content. | The order information including status will be displayed. |

**3.1.2 Feature Table #2**

|  |  |  |  |
| --- | --- | --- | --- |
| Supplier Feature | Input | Processing | Output |
| Login | Enter username and password and click login button. | System will check if supplier with the same username and password exists from the database. | If supplier exist, then will be directed to the supplier home page of this application. Otherwise, a prompt of “Wrong username and password” will pop out. |
| View all sell items | Click sell items button. | System will select all the sell items from this supplier in the database. | A list of sell items will be displayed. |
| Add sell items | Click add new item button, then enter or upload contents and click save button. | System will add new item to the database. | System will redirect to sell item page where the newest item will be displayed as well. |
| Edit sell items | Click the edit button of the selected item, and then modify contents and click save button. | System will first select the item from the database, then display the contents and save the modified contents to the database. | System will redirect to sell item page where the modified item will be displayed with the changes. |
| Delete sell items | Click the delete button of the selected item and confirm deletion. | System will delete the item from the database. | System will redirect to sell item page and the deleted item will no longer be displayed. |
| View all categories | Click category button. | System will select all the categories from this supplier in the database. | A list of categories will be displayed. |
| Add new categories | Click add new category button, then enter category name and click save button. | System will add the new category to the database. | System will redirect to category page where the newest category will be displayed as well. |
| Edit categories | Click the edit button of the selected category, and then modify category name and click save button. | System will first select the category from the database, then display the contents and save the modified contents to the database. | System will redirect to category page where the modified category will be displayed with the changes. |
| Delete categories | Click the delete button of the selected category and confirm deletion. | System will delete the category from the database. | System will redirect to category page and the deleted category will no longer be displayed. |
| Categorize items | Edit sell items and select a category for the items. (If a category is deleted then the items under that category would no longer be listed on the online grocery store. Items must be assigned to a category for users to buy.) | Same as edit sell items process. | Same as edit sell items output. |
| Receive new orders | N/A | System will generate new orders after users’ payments are completed. Then send notification to suppliers automatically. | Receive new order notification from the system. |
| View orders | Click on order home button. | System will select all orders that bought items from the supplier from the database. | A list of orders will be displayed. |
| Change orders’ status | Click on the selected order and change the status then click save button. | System will first select and display content of the selected order. Then the system will save the changes to the database. | System will redirect to order page where the modified order will be displayed with the changing status. |
| View items’ inventory | Click on inventory button. | System will select all the data from the inventory table that belongs to the supplier in the database. | A list of items’ inventories will be displayed in table format. |
| Modify item inventory | Modify inventory numbers of selected items and click save button. (Items of 0 inventory will display out of stock on the online grocery store) | System will save the changes to the database. | A list of items’ inventories with the changes will be displayed in table format. |

**3.1.3 Feature Table #3**

|  |  |  |  |
| --- | --- | --- | --- |
| Delivery Person Feature | Input | Processing | Output |
| See the pick-up and drop-off addresses of new orders | Click on the “new orders” button. | System will select all the unassigned orders from the database. | A list of unassigned orders will be displayed. |
| Accept orders | Click accept order button of selected order. | System will assign the delivery person to the order and save changes in the database. | System will redirect to the accepted order detail page. |
| View accepted orders | Click on users’ orders button. | System will select all the orders that are assigned to the delivery person from the database. | A list of orders will be displayed. |
| Receive money once order has been delivered | Click on the selected user’s order and change the order’s status to delivered then click save button. | System will first select and display content of the selected order. Then the system will save the changes to the database. | Delivery person will automatically receive money in its account within 2 business days. |

## **3.2 Use Cases**

### **3.2.1 Use Case #1**

|  |  |
| --- | --- |
| **Use Case Name** | Search Recipes |
| **XRef** | 3.3.4 Activity Diagram 3.3.4.1 Search Recipes |
| **Trigger** | Users click the search button. |
| **Precondition** | The user is on the *Recipe* page linked from the Goodie Foodie Mobile App Main Page. |
| **Basic Path** | 1. User enters an ingredient, recipe name or recipe type (ex: cuisine type) in the search engine. 2. The system will select the recipes that match the text that the user has typed in the search engine from the database. |
| **Postcondition** | A list of recipes or no match found message being displayed. |
| **Exception Paths** | The User may abandon the search at any time. |
| **Other** | None |

### **3.2.2 Use Case #2**

|  |  |
| --- | --- |
| **Use Case Name** | Share / Hide posts |
| **XRef** | 3.3.4 Activity Diagram 3.3.4.2 Share / Hide posts |
| **Trigger** | Users change the view setting of the post to public / private. |
| **Precondition** | The user is on the *My Posts* page linked from the Goodie Foodie Mobile App Post Page. |
| **Basic Path** | 1. The system will select and display a list of post created by the user itself from the database. 2. User clicks on the post that wishes to change view setting. 3. The system will select the post and display the details from the database. 4. User clicks edit button. 5. User changes the view setting and click save. 6. The system will save the modified view setting to the database. |
| **Alternative Paths** | Alternate steps of Basic Path step 2-5   1. User clicks on the Public/Private toggle button of the post. |
| **Postcondition** | A list of posts created by the user being displayed. |
| **Exception Paths** | The attempt may be abandoned at any time. |
| **Other** | None |

### **3.2.3 Use Case #3**

|  |  |
| --- | --- |
| **Use Case Name** | Receive ingredients’ expiration date notifications |
| **XRef** | 3.3.4 Activity Diagram 3.3.4.3 Receive ingredients’ expiration date notifications |
| **Trigger** | Users add ingredient expiration dates. |
| **Precondition** | The user is on the *Track Expiration Date* page linked from the Goodie Foodie Mobile App Main Page. |
| **Basic Path** | 1. User clicks add ingredient expiration date button. 2. User selects an ingredient and expiration date. 3. User enters the preferred notification time (ex: 3 days prior to expiry) and clicks save button. 4. The system will add the new ingredient expiration date to the database. |
| **Postcondition** | User receives a notification of the expiration date of the ingredient according to preferred notification time. |
| **Exception Paths** | The attempt may be abandoned at any time. |
| **Other** | None |

### **3.2.4 Use Case #4**

|  |  |
| --- | --- |
| **Use Case Name** | Checkout shopping cart |
| **XRef** | 3.3.4 Activity Diagram 3.3.4.4 Checkout shopping cart |
| **Trigger** | Users click on the checkout button. |
| **Precondition** | The user is on the *Shopping Cart* page linked from the Goodie Foodie Mobile App Shop Grocery Page. |
| **Basic Path** | 1. User clicks on the checkout button. 2. The system will display a checkout page where users must enter necessary information. 3. Users fill out the delivery address and payment methods. 4. Users click the place order button. 5. The system will process online payment. 6. The system will create a new order and update the database. |
| **Postcondition** | The newest order’s detail is being displayed. |
| **Exception Paths** | 1. The attempt may be abandoned at any time. 2. Transaction failed. |
| **Other** | None |

### **3.2.5 Use Case #5**

|  |  |
| --- | --- |
| **Use Case Name** | Register as a delivery person |
| **XRef** | 3.3.4 Activity Diagram 3.3.4.5 Register as a delivery person |
| **Trigger** | The user clicks the become a delivery partner button. |
| **Precondition** | The user is on the *Main* page of the Goodie Foodie Mobile App. |
| **Basic Path** | 1. Users click “become a delivery partner” button. 2. Users enter back account information. 3. Users accept the terms and policies and click submit. 4. The system will add users to the delivery person table in the database. |
| **Postcondition** | A link to the delivery person page is added on the Main page. |
| **Exception Paths** | 1. The attempt may be abandoned at any time. 2. The user is already registered as a delivery person. |
| **Other** | None |

### **3.2.6 Use Case #6**

|  |  |
| --- | --- |
| **Use Case Name** | Categorize items |
| **XRef** | 3.3.4 Activity Diagram 3.3.4.6 Categorize items |
| **Trigger** | The supplier assigns a category to the item and clicks the save button. |
| **Precondition** | The user is on the *Item* page linked from the Goodie Foodie Mobile App Supplier Page. |
| **Basic Path** | 1. Suppliers click on the item to categorize. 2. The system will display the item details. 3. Suppliers click on the edit button. 4. Suppliers select a category for the item and click save. 5. The system saves and updates the database. |
| **Postcondition** | A list of items sold by the supplier being displayed. |
| **Exception Paths** | 1. The attempt may be abandoned at any time. 2. Suppliers cannot find a matching category. |
| **Other** | None |

### **3.2.7 Use Case #7**

|  |  |
| --- | --- |
| **Use Case Name** | Change order status |
| **XRef** | 3.3.4 Activity Diagram 3.3.4.7 Change order status |
| **Trigger** | The supplier clicks the change status button. |
| **Precondition** | The supplier is on the *Order Detail* page linked from the Goodie Foodie Mobile App Order Page. |
| **Basic Path** | 1. Suppliers select a preferred order status and click save. 2. The system saves and updates the database. |
| **Postcondition** | The updated order detail is being displayed. |
| **Exception Paths** | 1. The attempt may be abandoned at any time. 2. Cannot find a matching status. |
| **Other** | None |

### **3.2.8 Use Case #8**

|  |  |
| --- | --- |
| **Use Case Name** | Modify items inventory |
| **XRef** | 3.3.4 Activity Diagram 3.3.4.8 Modify items inventory |
| **Trigger** | The supplier clicks the save button. |
| **Precondition** | The supplier is on the *Inventory* page linked from the Goodie Foodie Mobile App Main Page. |
| **Basic Path** | 1. Suppliers modify items’ inventory numbers from the inventory table and click save button. 2. The system saves and updates the database. |
| **Postcondition** | The updated inventory table is being displayed. |
| **Exception Paths** | 1. The attempt may be abandoned at any time. 2. Suppliers entered numbers less than zero. |
| **Other** | None |

### **3.2.9 Use Case #9**

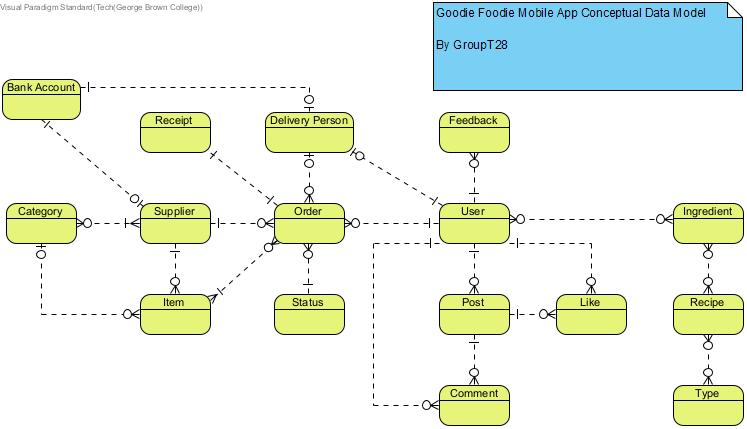
|  |  |
| --- | --- |
| **Use Case Name** | Accept orders |
| **XRef** | 3.3.4 Activity Diagram 3.3.4.9 Accept orders |
| **Trigger** | The delivery person clicks the accept button. |
| **Precondition** | The user is on the New O*rder* page linked from the Goodie Foodie Mobile App Delivery Person Page. |
| **Basic Path** | 1. The system assigns the delivery person to the order. 2. The system saves and updates the database. |
| **Postcondition** | The order detail is being displayed. |
| **Exception Paths** | The attempt may be abandoned at any time. |
| **Other** | None |

### **3.2.10 Use Case #10**

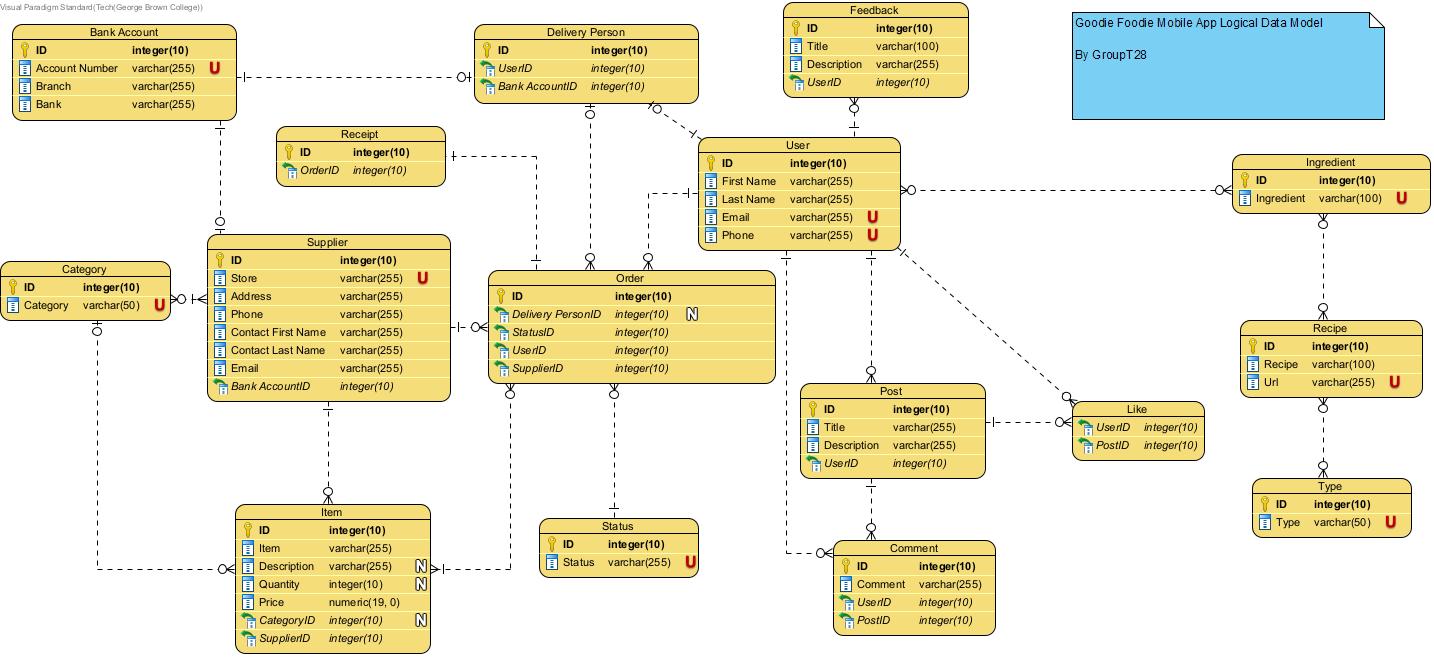
|  |  |
| --- | --- |
| **Use Case Name** | Delivery partner receive payment |
| **XRef** | 3.3.4 Activity Diagram 3.3.4.10 Delivery partner receive payment |
| **Trigger** | The delivery person changes the order’s status to delivered. |
| **Precondition** | The delivery person is on the *Order Detail* page linked from the Goodie Foodie Mobile App Accepted Order Page. |
| **Basic Path** | 1. Delivery People select order status “Delivered” and click save. 2. The system saves and updates the database. 3. The system will transfer the amount of delivery fee directly to delivery people's back accounts accordingly within 2 business days. |
| **Postcondition** | Delivery people receiving money. |
| **Exception Paths** | 1. The attempt may be abandoned at any time. 2. Transaction failed. |
| **Other** | None |

**3.3 Data Modelling and Analysis**

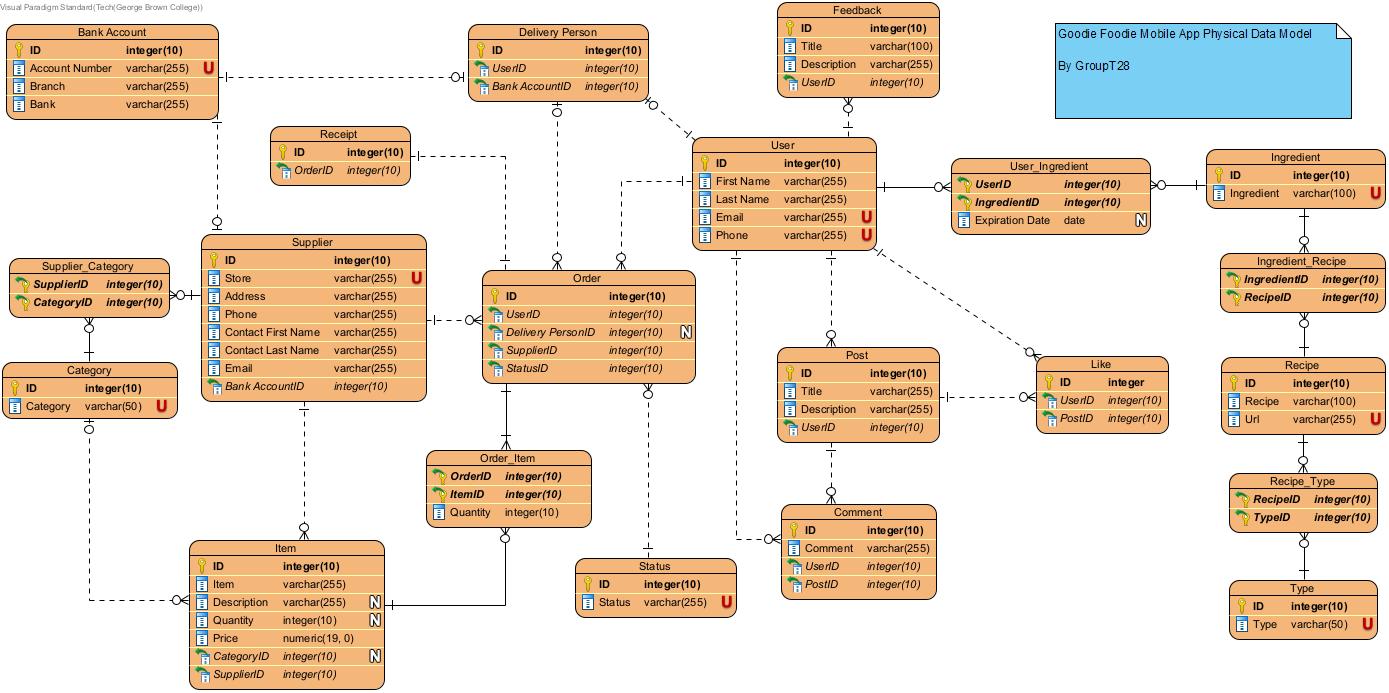
**3.3.1 Conceptual Data Model**



**3.3.2 Logical Data Model**

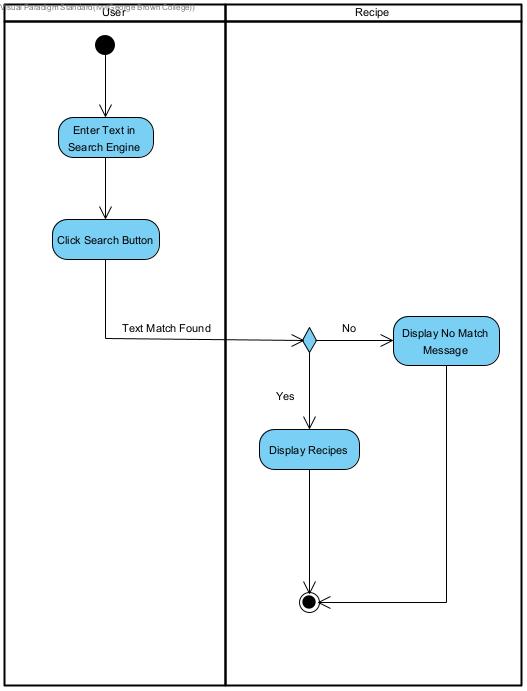


**3.3.3 Physical Data Model**

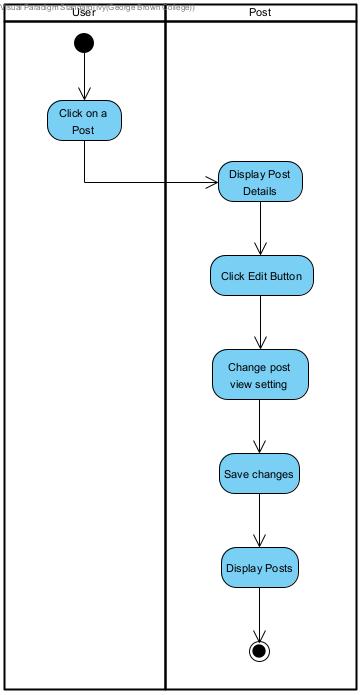


**3.3.4 Activity Diagram**

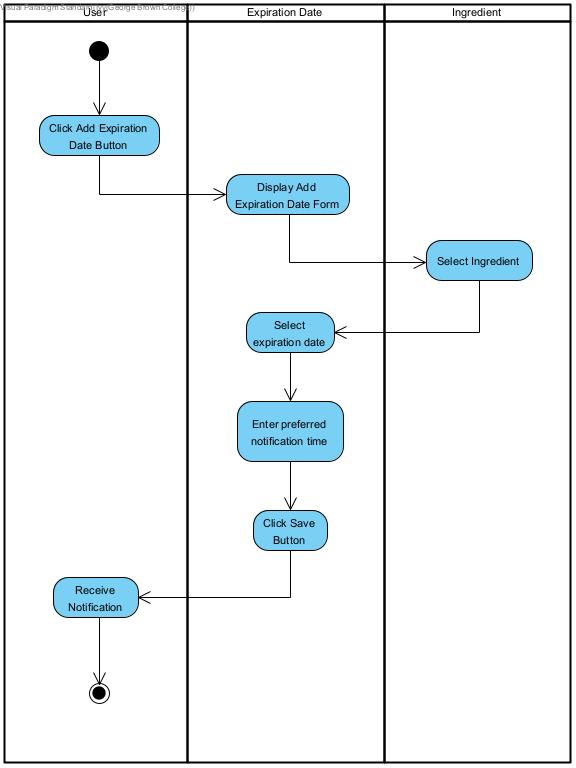
**3.3.4.1 Search Recipes**



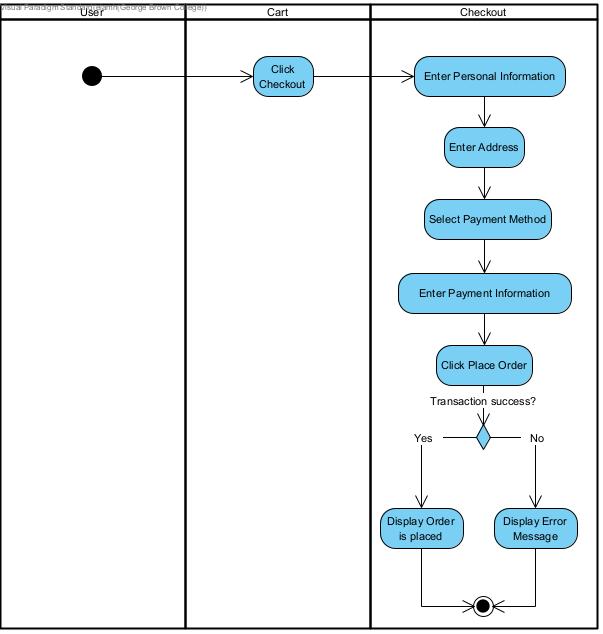
**3.3.4.2 Share / Hide posts**



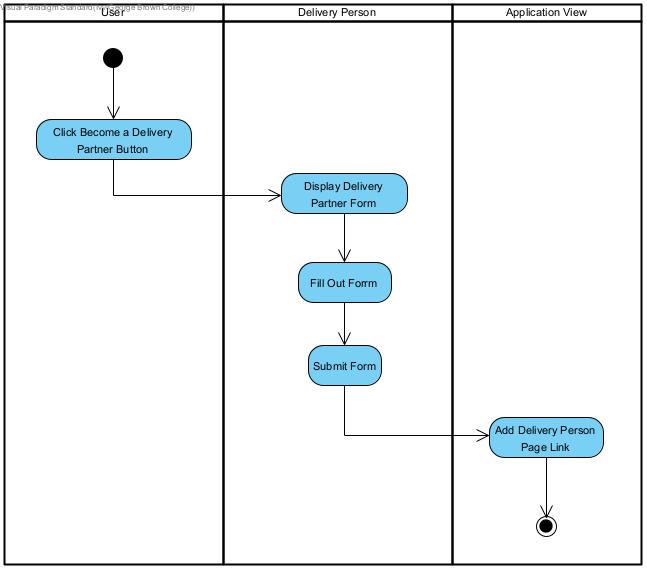
**3.3.4.3 Search Receive ingredients’ expiration date notifications**



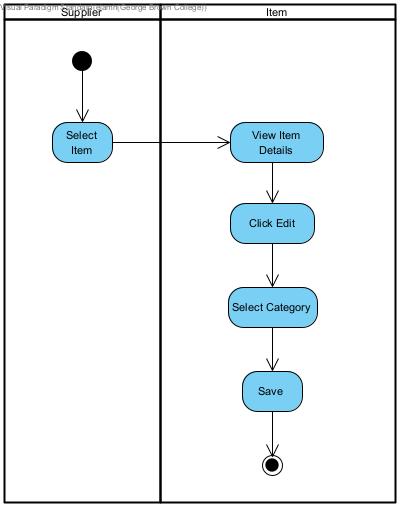
**3.3.4.4 Checkout shopping cart**



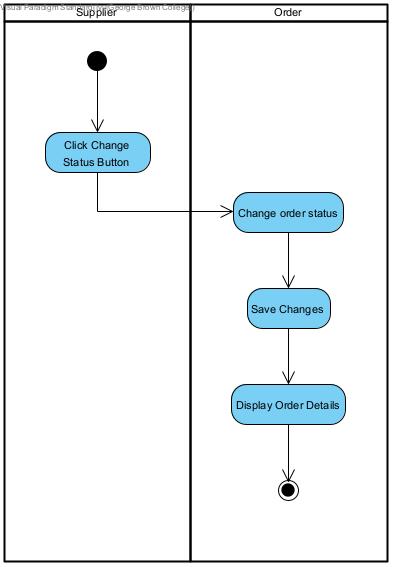
**3.3.4.5 Register as a delivery person**



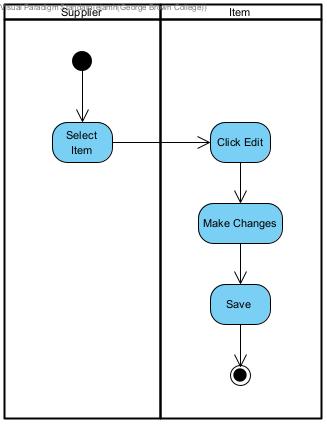
**3.3.4.6 Categorize items**



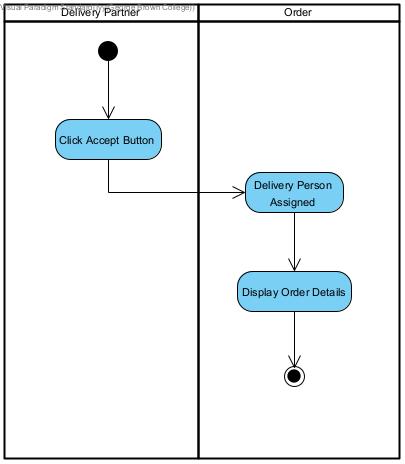
**3.3.4.7 Change order status**



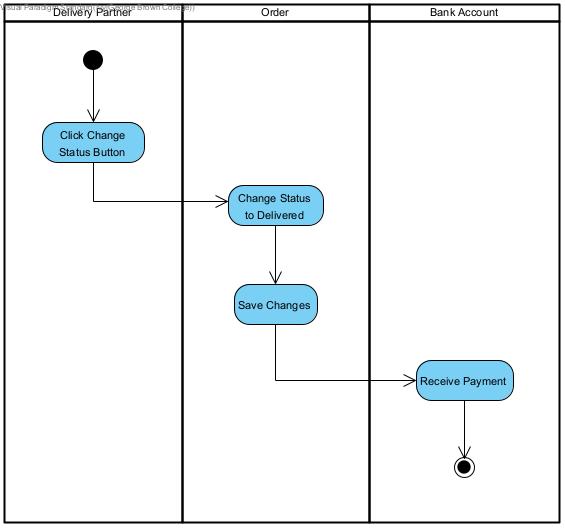
**3.3.4.8 Modify items inventory**



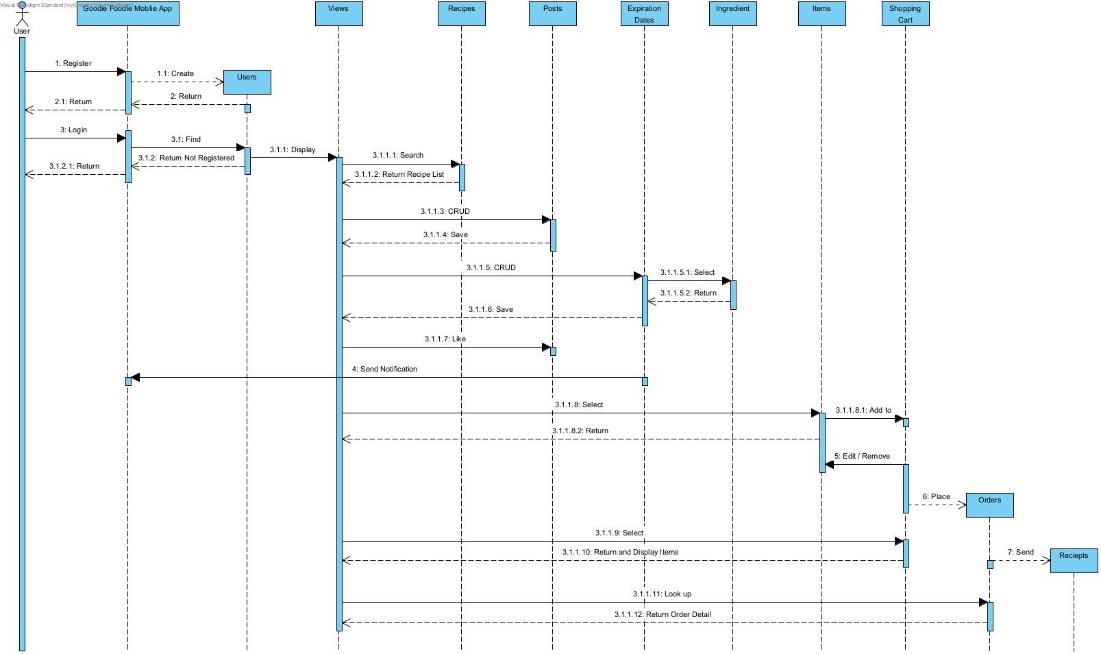
**3.3.4.9 Accept orders**



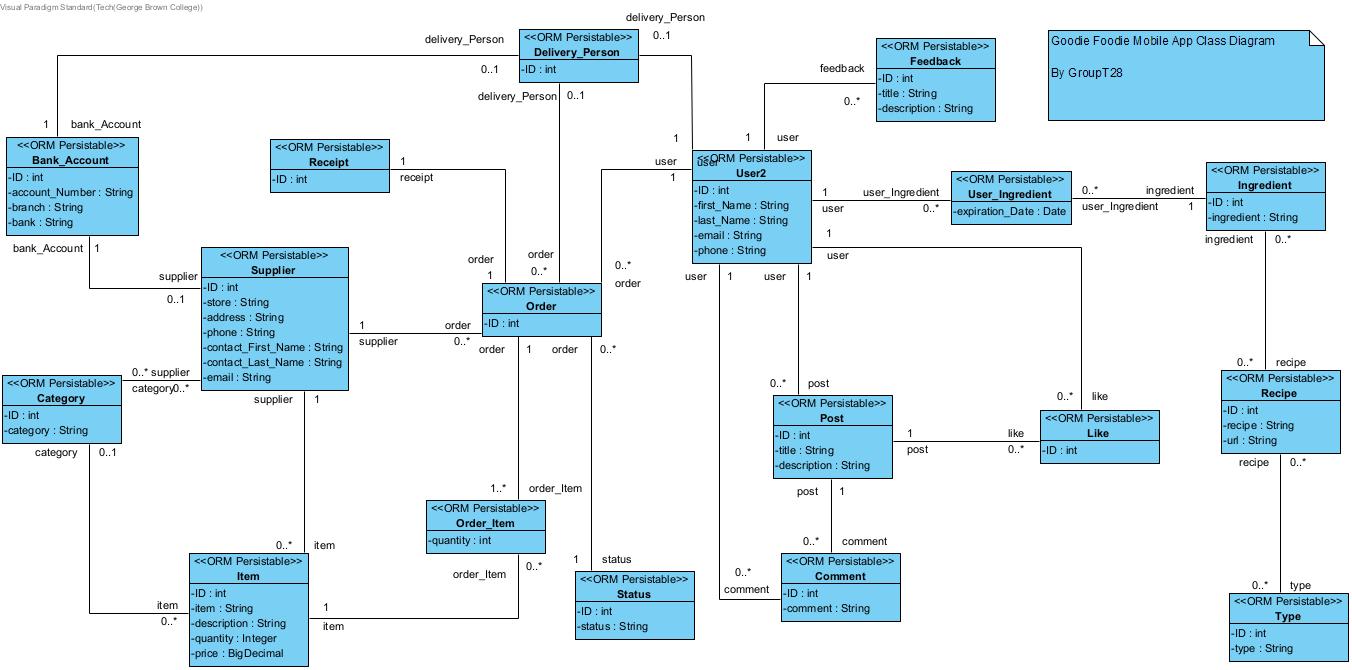
**3.3.4.10 Delivery partner receives payment**



**3.3.5 Sequence Diagram**

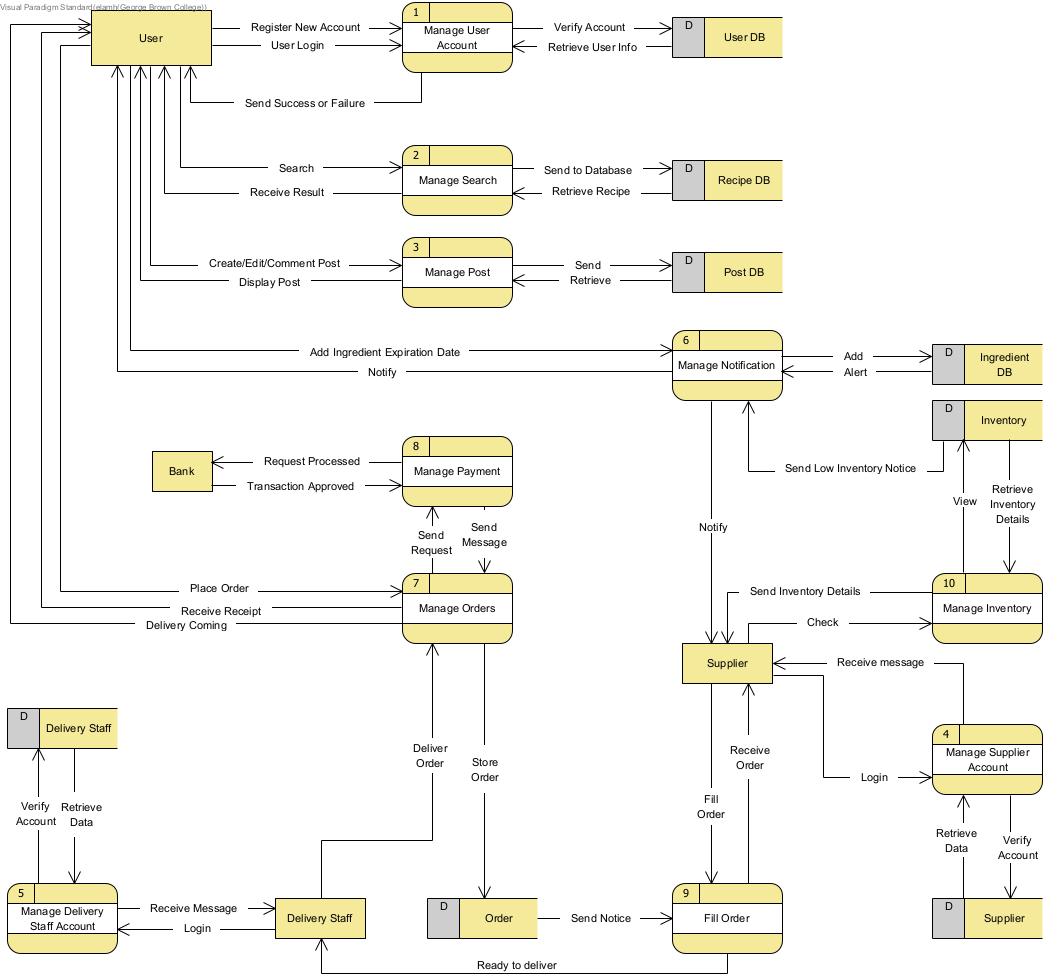


**3.3.6 Class Diagram**



**3.4 Process Modelling**

**3.4.1 Data Flow Diagram**



## **4.0 Non-Functional Requirements**

Performance: Each request should be processed within 10 seconds.

Usability: The application should be easy to navigate with a user-friendly interface that contains easy to use buttons. The time to load from one page to another should not exceed 3 seconds.

Security: User information should be protected and encrypted to prevent malicious external threats.

Portability and compatibility: The application must be compatible with android.

Reliability and availability: The application should be loaded within 3 seconds when the number of users is greater than 100.

Maintainability: Updates will occur regularly when necessary to give Users a better experience.

## **5.0 Logical Database Requirements**

## **5.0.1 User Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name of user (First name, Last name) | Varchar | Name of user |  |
| ID | Integer | ID number of Goodie Foodie App member |  |
| Email Address | Varchar | Email Address of user |  |
| Phone | Varchar | User phone |  |

**5.0.2 User Ingredient Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| User ID | Integer | Users use self-ingredient to search for the recipe |  |
| Ingredient ID | Integer | Ingredient buys by user |  |
| Expiration Date | Date | Expiration date tracking makes customers know their purchased ingredients will expire. |  |

**5.0.3 Ingredient Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Varchar | Name of ingredient |  |
| ID | Integer | ID number of ingredients |  |

**5.0.4 Recipe Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Varchar | Name of Recipe |  |
| ID | Integer | ID number of recipe |  |
| URL | Varchar | URL to link the general Recipe web side | User clicks on the URL link and the website will display all the descriptions of how the recipe is cooked |

**5.0.5 Type Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | Type of ID |  |
| Type | Varchar | Showing a kind of recipe such as clafoutis [French], plum pudding [British], and Indian pudding (NOT Native American) [American] |  |

**5.0.6 Feedback Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | ID number of feedback |  |
| Title | Varchar | Title of feedback |  |
| Description | Varchar | Description of feedback |  |
| User ID | Integer | Users get feedback from the other user comment on the dish |  |

**5.0.7 Post Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | ID number of Post |  |
| Title | Varchar | The post of title |  |
| Description | Varchar | The post of description that user writes in comment |  |
| User ID | Integer | Users can post complete foods in the post area |  |

**5.0.8 Comment Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | ID number of Comment |  |
| Comment | Varchar | Users can write any comment to post comments made by other users |  |
| User ID | Integer | Users receive the comment |  |
| Post ID | Integer | Users can look at the comment |  |

**5.0.9 Like Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| User ID | Integer | User receives likes from other users on posts |  |
| Post ID | Integer | Posts will show people who have clicked the like button |  |

**5.0.10 Delivery Person Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | ID of delivery | Including name |
| User ID | Integer | User id will show to the Delivery Person |  |
| Bank Account ID | Integer | User bank account will be shown to the delivery person if they have a deposit |  |

## **5.0.11 Order Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | ID of order |  |
| Delivery Person ID | Integer | Order checking the delivery person id | Checking the name of the delivery person |
| Status ID | Integer | Status entity |  |
| User ID | Integer | User entity |  |
| Supplier ID | Integer | Supplier entity |  |

## **5.0.12 Status Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | ID of status |  |
| Status | Varchar | Follow the status of the order |  |

## **5.0.13 Receipt Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | ID of receipt |  |
| Order ID | Integer | Showing up all information of Delivery Person ID, Status ID,  User ID and Supplier ID |  |

## **5.0.14 Bank Account Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | ID of bank account |  |
| Account Number | Varchar | Digital number of bank account |  |
| Branch | Varchar | Branch name of bank |  |
| Bank | Varchar | Bank |  |

## **5.0.15 Supplier Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | ID of integer |  |
| Store name | Varchar | Name of store |  |
| Address | Varchar | Address of store |  |
| Phone | Varchar | Phone number of store |  |
| Contact (First name and Last name) | Varchar | Contact full name |  |
| Email | Varchar | Email of contact |  |
| Bank Account ID | Integer | Bank account entity |  |

## **5.0.16 Category Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | ID of Category |  |
| Category | Varchar | Supplier and Item |  |

## **5.0.17 Item Data Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| ID | Integer | ID of Item |  |
| Item | Varchar | Show up all Item in store |  |
| Description | Varchar | Description of the Item |  |
| Quantity | Integer | Quantity of the Item |  |
| Price | Numric | Price of Item |  |
| Category ID | Integer | Category Entity |  |
| Supplier ID | Integer | Supplier Entity |  |

## **6.0 Other Requirements**

None.

**7.0 Approval**

The signatures below indicate their approval of the contents of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Role | Name | Signature | Date |
| Business Analyst | Loman Chan Shum | LC | Nov 8, 2022 |
| Database Administrator | Elam He | EH | Nov 8, 2022 |
| System Analyst | Hsin Yu Ivy Tsai | IT | Nov 8, 2022 |
| Project Manager | Alex Tsang | AT | Nov 8, 2022 |
| Technical Reviewer | Hui Qiu | HQ | Nov 8, 2022 |