Software Requirements Specifications
Kitchen Keeper

Killer Tech Solutions

Feb 25 2015

Contents

Revision History

Name	Date	Reason for Changes	Version
Initial Draft	29-01-15	Initial Submission	1.0
First Revision	25-02-15	Client Feedback	2.0

Table of Changes from RS 1.0 to RS 2.0

Issue	Change	Section(s) affected
Revision History contained internal versioning	Took out internal version numbers	Revision History
This app seemed to be directed at the elderly but everyone can use it.	Now says that this app is easy to use, not only for elders	1.2, 2.3
Stakeholders not listed and described	Added Stakeholders section	1.3
Glossary difficult to read	Formatted glossary as a table	1.4
Defining the abilities of the employees and administrators.	Clarified that the admins can read and write to the system but the normal employees can only read the system information.	2.3
Defining the two operating environments.	Clarified that there is a user environment and a user environment.	2.4
Document only referenced application for mobile devices but this application is able to be used on internet browsers.	Included internet browsers when mentioning mobile devices	2.4, 4.2.1, 4.3, 5.3.2
A maintainable application code was not going to be released.	A maintainable application code will be released to Food4U.	2.6
Confusion to the specifications of accessing the database.	The Food4U database must be able to reached at all times for users to connect to.	3.1.3
Personal database would update without confirming that the order was received.	Users will confirm that they have received all their ordered food items before personal database updates..	3.2.2

Database information unclear.	Referred to 3.1.2 for clarity.	3.2.3 R2, 3.3.2
Unable to sort items when ordering.	Able to sort items when ordering.	3.2.3 R5
Order sending times are unclear.	Orders are now sent at user specified times.	3.3.1
Different ordering options are unclear.	When creating an order, users specify whether it is a recurring or standard order. The recurring order is more structured and comes every week. The standard order is just a one time order.	3.3, 3.5
Sending orders was not specific enough.	When user sends a standard order, it can either be a same day or next day delivery.	3.4
Ordering system did not display favourite items and frequency of purchased items.	User can select items that he wants to be his “favourites”. These items, along with the most frequently ordered items, can be sorted to appear at the top of the ordering list for faster ordering.	3.3.3
There was no way to edit the personal database while offline.	When the system is offline and the personal database is edited, the changes will be queued. The changes will then be applied after the system is back online and the user confirms them.	3.2.3
System did not notify when it is the day of a food item expiring.	System notifies users a few days in advance of food expiring as well as on the expiry day.	3.6.3
Priorities not correct.	Changed to high priority.	3.7.1, 3.8.1
Differentiating between searching recipe database and searching online for recipes.	Clarified that the system will search both online and in the personal recipe database for recipes.	3.7.3
Recipe search too specific.	Recipe search now searches for recipes with 90% of items in database, not 100%.	3.7.3
Favourite recipes could not be accessed offline.	Favourite recipes now stored to user account instead of online database.	3.8.3

The system automatically deleted items when user selected recipe.	System does not automatically delete items.	3.9.3
No recipe timeout error message.	Timeout error displayed.	3.9.3
Communication interface.	Communication interface is clarified.	4.4
Feedback interface.	There is no need for a feedback interface since feedback can be given in an application store. Section 4.4 Feedback Interface removed.	4.4
Security details.	Added confirmation of the security of user information.	5.3.1
Accounts for mobile and internet browser.	Confirmed that the same account is used for both mobile devices and internet browsers.	5.3.2
Orders and payment security.	Added security before orders and payments are made.	5.3.2
Ability to log out.	Logout button is no longer hidden.	5.3.2
Languages.	This system will be available in 3 languages: English, French, and Mandarin.	6.1.3

1 Introduction

1.1 Purpose

This document describes the requirements of an at-home food management application called “Kitchen Keeper”. The scope of this project includes the at-home user interface and the connected ordering system that Food4U employees will view on their computers.

1.2 Project Scope

Kitchen Keeper is an application made for tablets and mobile devices that keeps track of food in the users’ home. Users can input food that they buy and remove food from their user profile after it is consumed. Food orders are also an important feature of the software. Easier than calling in and ordering, the orders go directly to Food4U and the food is delivered to the users’ homes. This application should increase exposure and uptake of Food4U’s new method of on-demand delivery.

1.3 Stakeholders

Below is a list of all entities who have a vested interest in the project.

- **Users** - Current customers of Food4U. They will need to migrate to the new system.
- **Food4U Admin** - Administrative employees that update and maintain customer records.
- **Food4U Warehouse Staff** - Employees that fill customer orders and prepare them for delivery.
- **Food4U Dispatch Drivers** - Employees that deliver orders to the customer.
- **Food4U Call Centre Staff** - These employees will continue to make changes to the Food4U database, although they will not interact directly with the new system.
- **Food4U IT** - These staff provide technical support to both Food4U staff and customers.
- **Food4U Product Owners** - Oversee the development and deployment of the system.
- **Killer Tech Solutions** - The developers of the system.

1.4 Glossary of Terms

FODC	Food Order Dispatch Control
Food4U Database	The database which stores Food4U’s product catalogue, prices, sales, etc.
GUI	Graphical User Interface
iOS	Mobile operating system for Apple devices
IT	Information Technology
Server	The computer(s) which stores all user information and orders
SQL	Structured Query Language

UERI	User Experience Report Interface
UI	User Interface
User Profile	Information on the Food4U database pertaining to the user, ie. their food inventory, password, etc.

1.5 References

[1] Food4U Delivery Services. (15 January 2015). *Kitchen Keeper Request for Proposal*. *Version 1.0* https://drive.google.com/file/d/0B8M8t5qTYKBqbDhTcVZ3MEZsQlU/view?usp=sharing

[2] IEEE-SA Standards Board. (25 June 1998). *IEEE Recommended Practice for Software Requirements Specifications.* https://connex.csc.uvic.ca/access/content/group/59567bde-894e-4f66-8ae7-2115ba6c44dc/PROJECT/IEEE_Standard_1998.pdf

[3] Killer Tech Solutions. (20 January 2015). *Developer Questions. Version 1.0* https://docs.google.com/document/d/1weYkMgto4HQmaeIzsDHR9g1ugaRDnSnn9aedTPaw4xA/edit?usp=sharing

[4] Government of Canada. (14 January 2015). *Personal Information Protection and Electronic Documents Act (S.C. 2000, c. 5)* http://laws-lois.justice.gc.ca/eng/acts/P-8.6/index.html

[5] Android Developer Resource (27 January 2015). *Dashboards - Platform Versions* http://developer.android.com/about/dashboards/index.html

1.6 Overview

This specification strives to create an understanding between Food4U and Killer Tech Solutions. It contains our interpretation of the product, features, and users (see section 2). Section 3 includes an overview and analysis of the system features. Section 4 contains details about the various entities that Kitchen Keeper must interface with. All additional requirements are presented in sections 5 and 6.

2 Overall Description

This section provides a high level summary of the system; specific requirements will be stated in later sections.

2.1 Product Perspective

Kitchen Keeper is designed to replace Food4U's telephone and website ordering system. This system will connect the users' systems with Food4U employees' systems. The users' interface will be on a tablet and the Food4U's interface will be on a workstation computer that can access the users' order information. They will be connected via the internet.

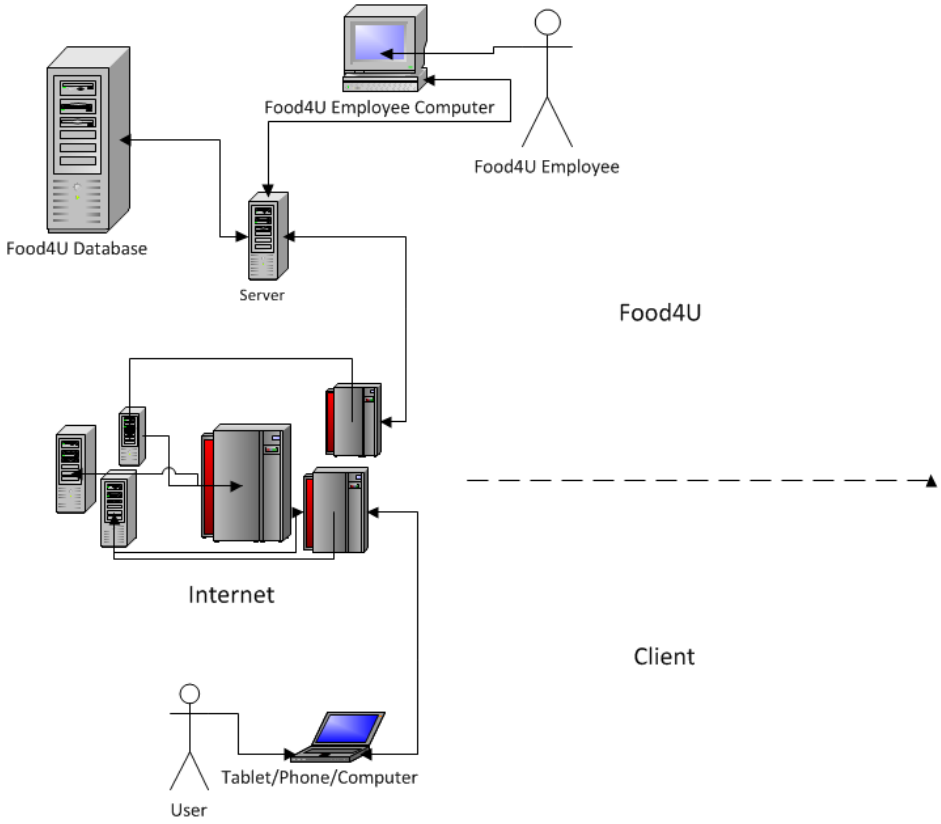

Figure 2.1: Block Diagram of System Interfaces

2.2 Product Features

Kitchen Keeper will keep track of all food in a user's home. It will display this information on a phone, tablet, or web browser as well as other details about the food, including expiry dates, images of the food, etc. (see section 3.1) Users will be able to create scheduled, recurring orders as well as on-demand orders. An important feature is recipe browsing and saving. A user can browse recipes based on the food they currently have, save a recipe they like to their personal database, and enter recipes of their own. There will also be a function to add items to order that are in a recipe but not in the user's fridge or pantry.

2.3 User Classes and Characteristics

There are several different user classes involved. They can be split into casual users and frequent users. Frequent users will expect a slightly different experience than casual users. The frequent users group includes those who rely more heavily on the system. This includes those who cannot go shopping. The elderly, people unable to drive, or people unable to walk to the store are those who will use this service most often. It is most important to satisfy businesses and the elderly. Older people represent a benchmark of who should be able to use this application. Restaurants/businesses are also a large potential market.

Kitchen Keeper will be easy to use. Anyone with little technical expertise will be able to use it, including elders. Elders also have trouble reading small text so readability is a consideration. Casual users are expected to use the on-demand order feature more often.

Food4U employees will also have to use this system on their workstations. There will be two different privilege levels, one for the generic employee and one for administrators and an IT support type user (see section 5.3). The generic employee will only be able to read from the system but administrators and IT support will be able to read and write information to/from the system. Other parties affected include drivers, warehouse staff, and Food Order Dispatch Control (FODC). Although not directly affected, these people will experience different routes and schedules.

Current users, Food4U employees (drivers, warehouse, FODC, and administration), and the developers are the parties concerned with the quality and usability of this application. As such, these groups will be the main focus for this document.

2.4 Operating Environment

The system has two separate operating environments, one for the user and one for the Food4U employees.

The first is the company application. Food4U currently has a database so there exists a server to manage their information and customer information. Our service will need to integrate with the existing database and existing customer information. Company computers run Windows XP at the moment with a possibility of upgrade to Windows 7 in the future.

The application is meant for the convenience of smartphones or tablets. It should be compatible with Android Ice Cream Sandwich (Android versions 4.0 and above account for 91.8% of users [5]) and the latest version of iOS. It must be cross-browser and cross-platform.

2.5 Design and Implementation Constraints

The system must interface with a database that contains all pertinent product and user information. The Food4U portion of the software must be compatible with Windows XP and Windows 7. Another constraint is that there still exists a telephone system and an online ordering system. The new system needs to work with the old systems (telephone/web page) so that deliveries can be efficiently scheduled. Users must be online to use the application.

2.6 Assumptions and Dependencies

Some factors to consider are:

- Food4U has a food item database requiring integration with Kitchen Keeper.
- Food4U wants to have a maintainable codebase, not just a final application.

3 System Features

3.1 Integration with Current Food4U Database

3.1.1 Description and Priority

Integration with the extant database of all the food and information that Food4U has to offer. This is very high priority because the rest of the system relies on this data.

3.1.2 Stimulus / Response Sequences

The existing Food4U database contains item information. It must be integrated and updated to include the following for each item:

- Price
- Approximate expiry date
- Sale status
- Stock information
- Category
- Image

3.1.3 Functional Requirements

REQ-1: The system must be able to access and display all the correct information from the database.
REQ-2: The Food4U database must be available at all times for users to connect to.

3.2 Display all of User's Food

3.2.1 Description and Priority

List all food items that the user currently has and information about those items. This is very high priority.

3.2.2 Stimulus / Response Sequences

The user will be able to view all food items they currently own according to the Food4U database. When an order is received, the user's inventory will update once the user has confirmed he has received each item from the order. When an item is used or expired, the user must manually remove it from their inventory.

3.2.3 Functional Requirements

REQ-1: The system must be able to add and remove items from the users' food inventory on user request.
REQ-2: The system must display all the correct information about each item (see section 3.1.2 for item information).
REQ-3: The system must display all food items that the user has in his house.

REQ-4: The user's inventory must be automatically updated when an order is received, after confirmation.
REQ-5: The display of the user's inventory must be sortable based on the items' information.
REQ-6: If the system is offline and the database is edited, the changes must be queued until system is back online. The changes will then be applied after user confirmation.

3.3 Create Food Orders

3.3.1 Description and Priority

Any food in stock in the Food4U database can be added/removed to the users' orders. Once the user sends the order, they can then be viewed by Food4U, who then delivers those orders. This is a high priority.

3.3.2 Stimulus / Response Sequences

The users must be able to search food products based on name, frequency of purchase, favourite items, and other information (see section 3.1.2). When a user performs a search, the correct items must be shown in response to the search. The user can then add and remove selected items to and from their orders.

3.3.3 Functional Requirements

REQ-1: The system must show all items that can be ordered.
REQ-2: The system must display all relevant items in response to a search.
REQ-3: The system must correctly keep track of the items that have been added/removed from the order list.
REQ-4: The system must keep track of the frequency that each item is ordered.
REQ-5: The system must allow the users to save items to favourites.
REQ-6: The system must allow the user to sort items based on favourite status and ordering frequency.

3.4 Send Food Orders

3.4.1 Description and Priority

Created orders can be sent to Food4U and will be delivered to the user. There are express orders and standard orders. This is a high priority.

3.4.2 Stimulus / Response Sequences

When the user chooses to send their order, it will be seen as active by Food4U. Food4U will create the order for delivery.

3.4.3 Functional Requirements

REQ-1: The order information must be correct when sent.
REQ-2: Food4U will receive the correct orders.

REQ-3: Orders must be sent and received quickly to ensure fast delivery.
REQ-4: The orders must be sent via the internet.
REQ-5: Users can select express or standard order, for same day or next day delivery.

3.5 Create Recurring Orders

3.5.1 Description and Priority

The system will automatically send a specified order on a recurring basis. This is a medium priority.

3.5.2 Stimulus / Response Sequences

The user will create an order that he wants to be delivered on every specific recurring period. The system will then send that same order every time that time has occured. The user can add items and they will stay on the list and be ordered every time until he removes the item.

3.5.3 Functional Requirements

REQ-1: The user must be able to create a specific order of items that they want to be recurring.
REQ-2: The order must be automatically sent every specific period that the user selects.
REQ-3: The system must order all the food on the recurring order every time.
REQ-4: Items will be removed from recurring order upon user request.

3.6 Expiry Dates

3.6.1 Description and Priority

The system will display the expiry dates of the food that the user has. It will warn the user when the food is close to expiring. This is a high priority.

3.6.2 Stimulus / Response Sequences

The system will automatically display an approximate expiry date of the food. This approximate date is from the Food4U database. The user can choose to change the expiry date to the exact expiry date. When the expiry date is close, the system will alert the user.

3.6.3 Functional Requirements

REQ-1: The system must automatically show an approximate expiry date.
REQ-2: The system must recognize an expiry date for each item, manually inputted by the user.
REQ-3: The system must alert the user daily two days before, the day before, and on the expiry date.

3.7 Find Recipes

3.7.1 Description and Priority

The system will enable the user to make a rudimentary web search, as well as search the personal recipe database, to find recipes for use with Kitchen Keeper. This is a high priority.

3.7.2 Stimulus / Response Sequences

The user will search using keywords to find recipes. When a recipe is found to the user's liking, Kitchen Keeper will provide a facility to save that recipe onto the application device.

3.7.3 Functional Requirements

REQ-1: The system must support online searching for recipes.
REQ-2: The user must be able to search the recipe database for recipes.
REQ-3: The system must save recipes specified by the user.
REQ-4: The system must parse recipes such that at least 90% of the items in the recipes correspond to food items extant in the Food4U database.

3.8 Input Recipes

3.8.1 Description and Priority

The system will enable the user to input recipes for storage in the Food4U database. This is a high priority.

3.8.2 Stimulus / Response Sequences

The user will be presented with a list of possible ingredients and a means to select ingredients. Alternatively, the user can input the full name of the ingredient. When an ingredient is selected for inclusion in the recipe, the user will be prompted for the amount, and given options for measurement by quantity, weight, or volume. After all ingredients are added to the recipe, the user will be able to review and name their creation. The recipe will be uploaded and stored in the Food4U database.

3.8.3 Functional Requirements

REQ-1: The system must recognize any ingredient from the Food4U database.
REQ-2: The system must accept new ingredients not yet recorded in the database.
REQ-3: The system must store user inputted and favourited recipes in the user database for future use.

3.9 Suggest Possible Recipes

3.9.1 Description and Priority

A list of items the user currently owns is stored on the Food4U database. From this list, possible recipes comprising currently owned items will be suggested to the user. This is a low priority.

3.9.2 Stimulus / Response Sequences

The user can browse a list of recipes that the user has previously compiled via internet search. Alternatively, the user can browse recipes that have been manually entered. Kitchen Keeper will suggest only recipes that the user has all or almost all the ingredients for. Upon selection of one of the listed recipes, the system is to remove from the list of currently owned items all items required to complete said recipe. This operation must be reversible. Searching for recipes should take no longer than ten seconds.

3.9.3 Functional Requirements

REQ-1: The system must suggest recipes to the user based on their currently owned food items.
REQ-2: The recipe search must time out after ten seconds and display a timeout message.

4 External Interface Requirements

4.1 User Interfaces

This application is designed for tablets, so it must make use of the available screen space. With a large variance in user technological skill and a number of potential use cases, the UI should be designed to provide intuitive navigation and effective error recovery in order to prevent unintended user interaction.

4.1.1 Product Style

All UIs of the system should maintain a style that is consistent with other Food4U products:

- Fonts: Oswald, Open Sans
- Colours: Black, green, and light grey on a white background
- The Food4U logo must be surrounded by whitespace equal to half of the logo's height
- The interface should consist of high-contrast colors and large buttons in order to accommodate the needs of the visually impaired as much as possible

4.1.2 Standard Buttons and Functions

Help:
There should be a help button on every screen to inform the user of the functionality of each button and to give the user tips on how they can use the application.
Home:
There should be a home button on every screen to allow users to navigate more quickly.
Back:
Each screen of the application (except the main screen) should have a back button to return to the previous screen. This allows for faster navigation and easy recovery after making a mistake.

4.1.3 Screens

Tutorial Screen:
This screen will display during the first time a user launches the application (or display later upon user request) to guide the user through basic application functionality.
Inventory Screen:
This screen keeps track of the users current food inventory and allows them to make changes manually.
Home Screen:
The home screen is used for navigating to the other screens. It should be displayed on startup (except the first time the application is run).

Recipes Screen:
This screen provides users with a list of recommended recipes. It will make recommendations based off of whichever recipes have been saved by the user, or what they have the ingredients for. If they are missing ingredients, there should be an option for the user to add the ingredients to their next order. From this screen, the user is also able to input, view, and manage his favourite recipes.

Order Screen:
This screen contains information about the users regular order as well as any one-time orders they have in progress. Orders shall be easily modifiable and there should be an option for quick delivery.

Help Screen (Display tutorials and functionality explanations):
This popup window provides information and tips for the user based on the screen they are currently viewing.

4.1.4 Error message display standards

The system should display messages when unexpected conditions happen to inform the user and also give them descriptive explanations. All the error messages should consist of a plain language description as well as a standard error code. The plain language description should give the user necessary troubleshooting tips and the error code should follow a standard error code table which would be provided to the IT group for customer service purposes.

4.2 Hardware Interfaces

4.2.1 Supported Devices

This system should be able to support common handheld devices (ie. smartphones and tablets) using different operating systems (ie. Android and iOS) as well as personal computers for internet browser use. The system should adjust itself in different screen orientations to produce an intuitive and easy to use layout.

4.2.2 Data transfer

The Kitchen Keeper system should communicate with the online database via the internet and there should be no support for any functionality without an internet connection.

4.3 Software Interfaces

The Food4U application must provide software interfaces to:

- Food4U database for retrieving and updating information on each user's inventory, orders, and recipes
- Various payment methods (ie. credit cards, PayPal)

Food4U administration should have access to some personal user information, the food order database, as well as the Food4U inventory database via administrative accounts accessed through any web browser.

4.4 Communications Interfaces

This system must be able to properly interact with the existing Food4U system that is in place for their employees.

4.4.1 Order Submission Interface

This interface should be used as the basic order communication. All orders sent through the Kitchen Keeper system should be sent directly to the order processing system provided by the client company. Information transferred by this interface would include user credentials (ie. credit card information, delivery address etc.), therefore this communication must be secured to prevent user's information from leaking.

5 Other Non-Functional Requirements

5.1 Performance Requirements

5.1.1 Behaviour Under Loss of User Connectivity

The user application will require internet access to the server to function. It is expected that the application will primarily be used at a user's home where Wi-Fi should be available. If the user is not at home, they have the option to utilize cellular or other Wi-Fi connections to regain connectivity.

- If connection to the server is lost while the application is open, it will immediately notify the user.
- If a connection to the server cannot be established when the application is launched, it will immediately notify the user.

5.1.2 Recipe Search Response Time

After a user chooses the ingredients they want to use in a recipe, they will begin a search for applicable recipes (as outlined in section 3.10). The results of this search (which encompasses both the user's custom recipes as well as recipes from the internet) must return within 10 seconds.

5.2 Safety Requirements

Since the user system is entirely software-based, no safety concerns are anticipated beyond the existing safety considerations of using a mobile electronic device in the presence of kitchen apparatus. These considerations are entirely encompassed by the user domain and will not be affected by the design of Kitchen Keeper software.

5.3 Security Requirements

5.3.1 Scope of User Data

This section provides an overview of the data which is associated with a user and is handled at some point by the system. This provides context for the remainder of section 5.3. Any of this data could potentially be viewed on the user device, transmitted between the user device and the server, or stored on the server. All this data will be stored securely.

- Username and password
- Name
- Shipping address
- Payment information
- Personal list of food at home
- Customized expiry dates for food items
- Personal recipes
- Food4U catalogue searches

- Food orders

5.3.2 User Authentication

- When first entering the application, a user must either sign-in with an existing account, or create a new account. Initially, all users are required to make an account, even if they are existing Food4U customers.
- The user will remain logged in to their account for convenience.
- The user has the option to logout of the account on their device.
- The same user account will be used for tablets and browsers.
- Before sending an order, the user must confirm the order with a password.
- Before editing a recurring order, the user must enter a password.
- Before making a payment, the user must enter a password.

5.3.3 Data Transmission

All user data must be encrypted during transmission over the internet, as it contains confidential information as described in section 5.3.1

5.3.4 User Device Security

- The application must conform to any security standards required for the Android and iOS application markets.
- It is ultimately the user's responsibility to control physical access to their personal device. Some devices support alternate forms of authentication which the user may employ.

5.3.5 Server Security

- All user data stored on the server must be encrypted.
- Passwords stored on the server must be hashed and salted.
- Only authorized Food4U employees may obtain access to stored user data. As such, the system must support two separate employee roles:
    i. A limited role for employees handling orders. This role can only view food orders, the details of the order, and the shipping address.
    ii. An administrative role for IT support and management. They have full read and write access over the user data, this includes making edits or deletions.
- All servers storing user data must be physically located within Canada (see section 5.3.6)

5.3.6 Privacy

The system must conform to any regional restrictions regarding user privacy and security. Specifically, the system will conform to the Canadian Personal Protection and Electronic Documents Act (PIPEDA), which regulates how organizations may collect, use, and disclose personal information [4]. Additional restrictions may apply at the provincial level with Canada, or if the application will be distributed internationally.

5.4 Software Quality Attributes

5.4.1 Availability

- For users to place orders through the system, the database and server must be available to accept user orders.
- For Food4U warehouse staff to prepare orders for delivery, the server must be available to accept employee queries.

5.4.2 Correctness

- The user application must display the correct information for each food item viewed, based on what data is available in the Food4U database.
- The system must not place food orders without user confirmation, payment, and a delivery address.

5.4.3 Portability

The system must be available for Android and iOS tablets. As such:

a. The UI functionality, layout, and style must remain consistent across all platforms.
b. The user application and code-base must meet all quality requirements required for listing on both the Android and iOS application markets.
c. The user application should follow device standards where possible, unless following such a standard would interfere with (a).

5.4.4 Usability

- The user application should be learnable by non-technical users. Ease-of-learning is prioritized over ease-of-use, as it is important not to dissuade new users of the system.
- The user application must provide user help documentation and IT support upon request from any page.

6 Other Requirements

6.1 Accessibility

6.1.1 Text Size

Users with low visual acuity (especially the elderly) may have difficulties reading text that is too small. The system must support these users, so all fonts and icons used in the tablet application must be large enough to be read with ease by the majority of users.

6.1.2 Colour Blindness

The system must be accessible to colour blind users.

6.1.3 Language Support

The system must allow the possibility of expansion to support English, French, and Mandarin.

6.2 Database Requirements

6.2.1 Availability

The Food4U database must be readily available so it can be used to display available food to users and for ordering.

6.2.1 Consistency

The Food4U database must include all the correct items and information about those items so that it can be displayed to the users. Information about items must include: prices, approximate expiry dates, sale items, stock information, category of each item, and an image for each food.

7 Use Cases

7.1 Login

7.1.1 Description

This use case describes how a Food4U customer can login to the Kitchen Keeper System.

7.1.2 Actors

Customers, Kitchen Keeper System

7.1.3 Pre-Conditions

The customer must have an account with Food4U, a username, a password and this information needs to be in the database.

7.1.4 Main Flows

1. The customer opens a not yet logged in Kitchen Keeper System on their device.
2. The system displays a customer login form that accepts a username and a password.
3. **<Enter Username>** The customer inputs their username.
4. **<Enter Password>** The customer inputs their password.
5. **<customer Authentication>** The system authenticates the customer.
6. The system displays a welcome message to the customer.
7. Finish login process.

The figure below presents the sample screen of a login page:

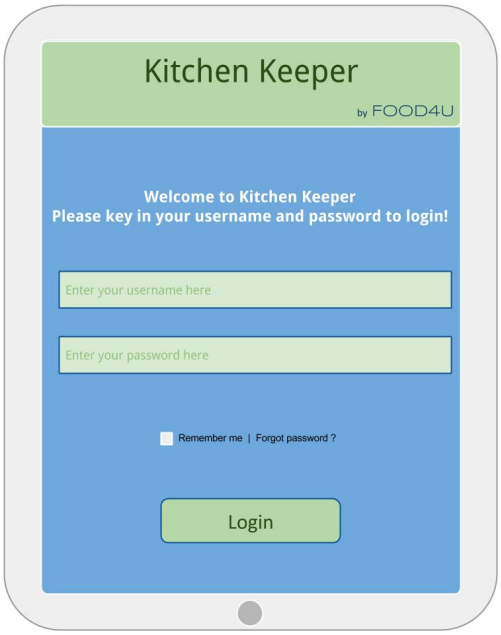

Figure 7.1a: Login

7.1.5 Post-Conditions

The system should keep an account access log to keep information of each account access.

7.1.6 Alternative Flows

A. If the customer's input does not match the database, the system will display an error message: "The username or password you entered is incorrect". Then the customer should correct their input and try again.

The figure below presents the error message of incorrect input:

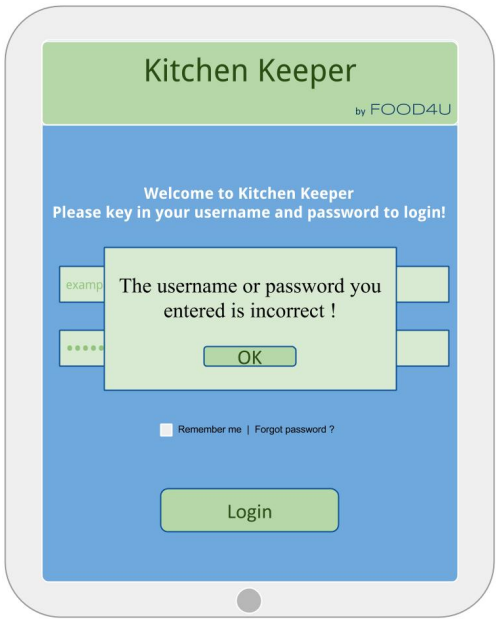

Figure 7.1b: Invalid Login

B. The customer should click "Forgot password" if they forgot their password; the system will direct the customer to a recovery page to reset their password via e-mail.

The figure below presents the reset user account page:

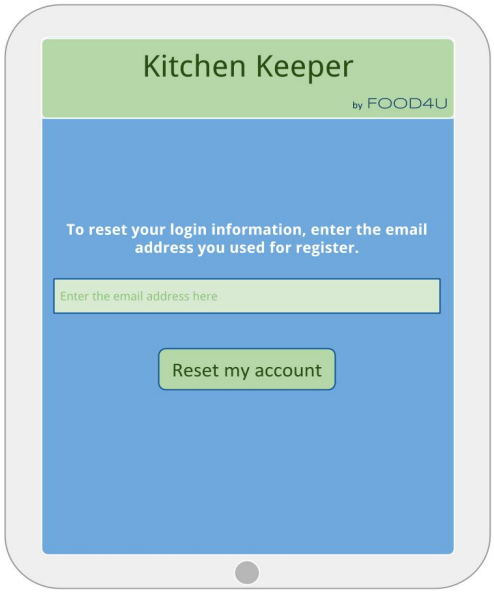

Figure 7.1c: Reset Login Information

7.2 Order Food

7.2.1 Description

This use case describes how a Food4U customer can order food.

7.2.2 Actors

Customers, Kitchen Keeper System, Food4U employees

7.2.3 Pre-Conditions

The customer must have an account with Food4U. They must have a payment option set up or have alternative payment available.

Figure 7.2a: Order Type

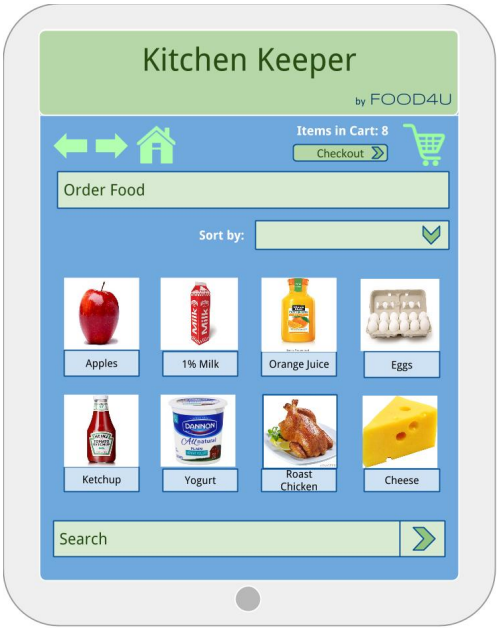

Figure 7.2b: Item Browsing

7.2.4 Main Flows

1. **"Include Login"**
2. The customer selects the option to order food on the main Kitchen Keeper screen.
3. **<Order Types>**The system displays options to edit recurring orders, create a new recurring order, or create a standard order.
4. The customer selects "create a standard order".
5. The system displays the item browsing screen.
6. **<Browse/Search>** The customer searches for the item he desires.
7. The system displays relevant information about the item including an option to "add to cart".
8. The customer selects "add to cart" .
9. The customer selects "checkout".
10. **<Checkout>** The system displays the customer's cart with options to remove items, go back to browse, or checkout.
11. The customer selects "make payment".
12. **"Include Make Payment"**

7.2.5 Post-Conditions

The Food4U system has a new order from the customer and the customer is awaiting their order.

7.2c: Cart View

7.2.6 Alternative Flows

A. At **<Order Types>** If the customer opted to create a new recurring order then

1. The customer selects “create a new recurring order”.
2. The system prompts the customer to select when and how often the order occurs.
3. The customer selects their preferences and rejoins the main flow at **<Browse/Search>**

If the customer opted to edit a recurring order then.

1. The customer selects “edit a recurring order”.
2. The system prompts the customer to select which order .
3. The customer selects the order they wish to edit.
4. The system displays the order list with options to change the name of the list, delete the list entirely, remove items from the list, and add items to the list.
5. The customer clicks the “X” beside an item.
6. The system deletes that item from their list and they are displayed the list without the item.

B. At **<Browse/Search>** If the customer opted to browse items then

1. The system displays items by category.
2. The customer selects which category they wish to browse.
3. The system displays:

7.3 Find Recipes

7.3.1 Description

This use case describes how a Food4U customer can manage their saved recipes and find more.

7.3.2 Actors

Customer, Kitchen Keeper System

7.3.3 Pre-Conditions

The customer must have an account with Food4U. They must have internet access. Their username, password, and saved recipes must be stored in the customer's database.

7.3.4 Main Flows

1. **"Include Login"**
2. The customer navigates to their "Saved Recipes" page. (Figure 7.3a)
3. The system displays the customer's saved recipes and options for the next step.
4. **<Search recipes>** The customer chooses to search for new recipes.
5. The system prompts the customer to enter some recipe keywords.
6. The customer enters one or more recipe keywords.
7. The system generates a list of recipes for the customer based on their search.
8. The customer chooses a recipe.
9. The system displays a list of ingredients and cooking instructions as well as options to order the ingredients or save the recipe. (Figure 7.3b)
10. The customer chooses to order the ingredients for the recipe.
11. **<Order type>** The system asks the customer if they would like the ingredients added to a standard or express order.
12. The customer chooses to add the ingredients to their order.
13. The system adds the ingredients to the next order and prompts the customer to save the recipe for later use.
14. The customer chooses to save the recipe.

Figure 7.3a: View Saved Recipes

Figure 7.3b: View Recipe

7.3.5 Post-Conditions

The customer's saved recipes and orders have been correctly adjusted.

7.3.6 Alternative Flows

A. At **<Search recipes>**, if the customer chose to add their own recipe, then
    1. **<Enter recipe>** The system displays a window for the customer to enter recipe information. (Figure 7.3c)
    2. The customer inputs recipe details and submits.
    3. The system adds the recipe to their saved recipes and brings the customer back to the Find Recipes page.

B. At **<Search recipes>**, if the customer chose to view one of their saved recipes, then
    1. **<Recipe options>**The system displays all recipe information with the option to edit or order the recipe.
    2. The customer chooses to order the recipe.

   Return to **<Order type>**

C. At **<Search recipes>**, if the customer chose to edit the recipe, then
   Return to **<Enter recipe>**

D. At **<Order type>**, if the customer chose to have the ingredients delivered immediately then
    1. The system creates a new order with the ingredients.
    2. **"Make payment"**

Figure 7.3c: Add New Recipe

7.4 Track Inventory

7.4.1 Description

This use case describes how a customer can keep track of items in their food inventory.

7.4.2 Actors

Customer, Kitchen Keeper System

7.4.3 Pre-Conditions

The customer must be logged in and have food items in the inventory

7.4.4 Main Flows

1. The customer clicks the “View Inventory” button.
2. **<View Inventory>** The system displays all food items the customer currently has.
3. **<Confirmation Prompt>** The system will prompt the customer to confirm whether they have received all items of their order or not, with the options “Yes” or “No”

Figure 7.4a: Order Confirmation

4. The customer clicks the “Yes” button.
5. The system updates the inventory and displays all food items the customer currently has.
6. The customer selects a food item that is used or expired.
7. **<Remove Item Prompt>** The system displays information about the selected food item including an option to remove the item.
8. The customer clicks the “Remove Item” button.
9. The system prompts the customer to confirm they want to remove the food item, with the options “OK” and “Cancel”.

Figure 7.4b: Inventory Editor

10. The customer clicks the "OK" button.
11. The system displays a message informing the customer that the item was successfully removed.

7.4.5 Post-Conditions

The customer's food inventory is updated.

7.4.6 Alternative Flows

A. At **<Confirmation Prompt>**, if the customer selects the "No" button then

1. The system displays a phone number and message telling the customer to contact the help line for assistance.

Return to **<View Inventory>**

B. At **<Remove Item Prompt>**, if the customer selects the "Cancel" button then

1. The system displays information about the selected food item.

Return to **<View Inventory>**

7.5 Track Orders

7.5.1 Description

This use case describes how a Food4U customer can track their current and past orders include status as well as delivery information.

7.5.2 Actors

Customers, Food4U database

7.5.3 Pre-Conditions

Customer must be logged in to the application, have previously placed an order, and have gone through the payment process for that order for them to keep track their orders.

7.5.4 Main Flows

1. The customer clicks on the “Order Tracking” button.
2. **<Find Customer Orders>**The system searches the customer’s account and finds all the orders belonging to the customer and displays current active orders on default with the option to choose order history period.
3. **<Find Delivery Information>** Once the order is sent, information about the delivery status will be shown (Figure 7.5). The order of delivery status goes in order: received order information, creating order, order ready for delivery, order on truck, and order received by customer.

7.5.5 Post-Conditions

The customer can update their inventory with their received order.

7.5.6 Alternative Flows

A. Order is not received on time
    1. The customer can call Food4U to figure out the unreceived order.

Figure 7.5: Order Tracking

7.6 Make Payment

7.6.1 Description

This use case describes how a Food4U customer can make a payment for their orders.

7.6.2 Actors

Customer, Food4U database, payment system

7.6.3 Pre-Conditions

The customer must have an existing account and have at least one order pending, having at least one item in their shopping cart. Also, the customer must have a valid payment method: Credit Card, Debit Card, PayPal etc.

7.6.4 Main Flows

1. The customer tries to make a payment for an existing unpaid order by clicking on the "Checkout" button on the item selection screen or the shopping cart inspection screen.
2. **<Select Payment Option>** The customer is presented with several payment options, including Credit, Debit, Paypal, and a drop menu which has any payment methods the customer has previously saved.

Figure 7.6a: Payment Selection

3. **<Submit Payment Information>** The customer inputs their payment information. They supply pertinent information such as card type, name on card, card expiry (as applicable).
4. When all information is entered, the customer clicks "Submit Payment Information," and they are asked to confirm this order via a popup dialog.

Figure 7.6b: Payment Information

7.6.5 Post-Conditions

Payment status of the current order is updated to reflect that the order has been paid for. Payment information is stored if the customer specifies that the system do so.

7.6.6 Alternative Flows

A. At **<Submit Payment Information>,** the customer enters invalid payment information.
    1. The system will not navigate away from the **<Submit Payment Information>**. The customer must enter the correct information before the system will accept it.

B. At **<Submit Payment Information>,** the customer's payment method of choice is rejected.
    1. The system will alert the user that the payment was rejected and prompt for a different payment method

7.7 Summary

The use cases identified in the preceding sections are summarized in the high-level system use case diagram below (Figure 7.7).

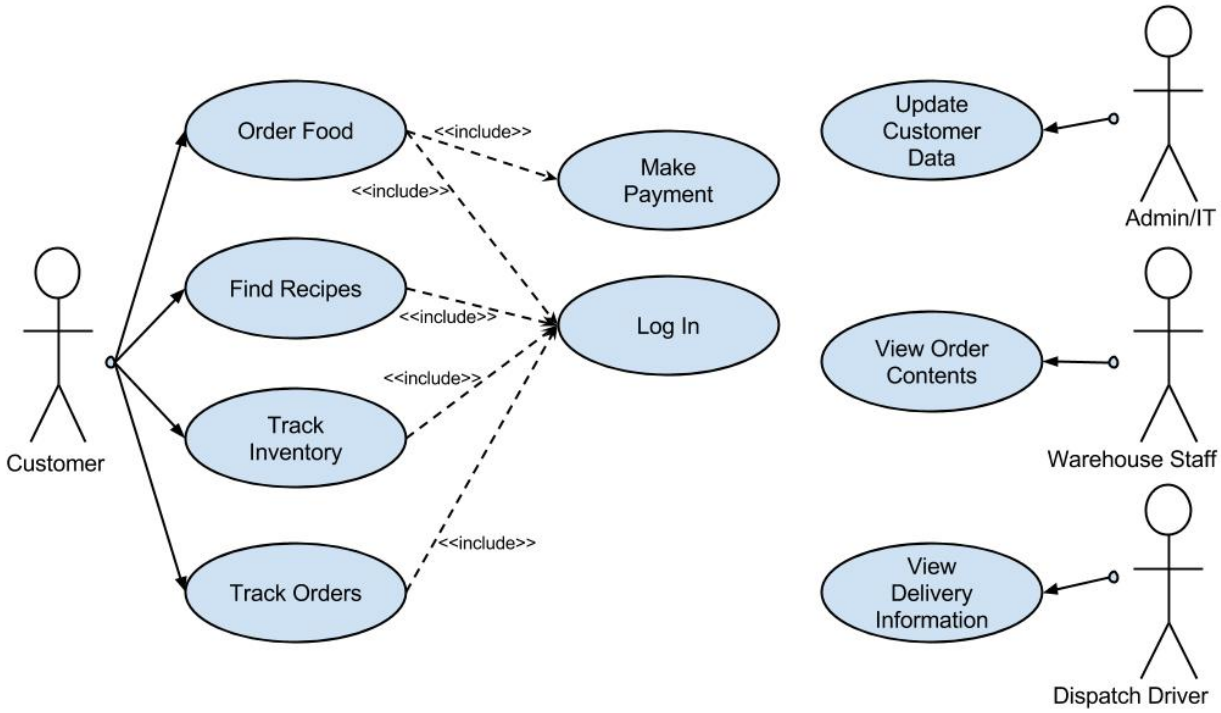

Figure 7.7: Use Case Diagram

8 Domain Model

8.1 Entity-Relationship Diagram

The entity-relationship diagram details how customers and system entities interact with the system and with one another. Entities are listed with their constituent attributes, and cardinality is included for each relationship.

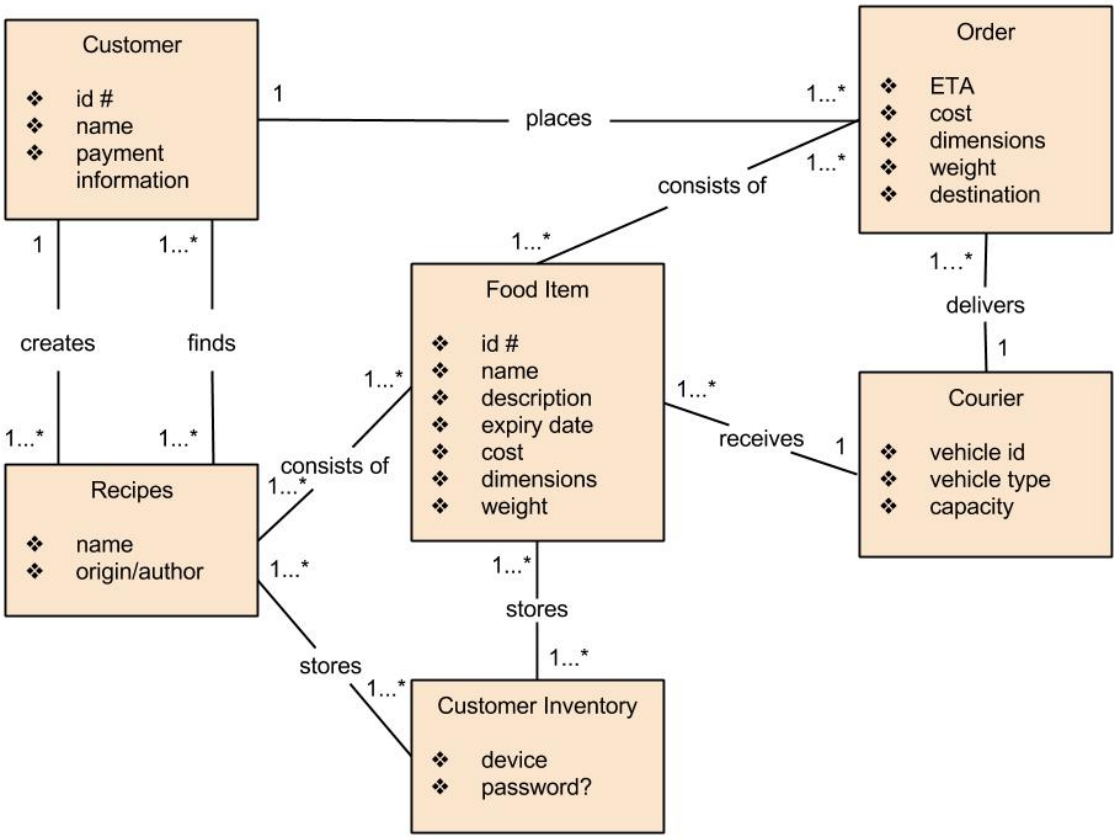

Figure 8.1: Entity-Relationship Diagram for Kitchen Keeper

8.2 Data Flow Diagrams

The Data Flow Diagrams below detail data interactions in the system on 3 levels, each with a progressively higher level of detail.

8.2.1 DFD Level 0 (Context Diagram)

The context diagram shows the system at its highest level (Figure 8.2a). It indicates the external entities which the system must interface with. To make the diagrams more readable, “Customer Data” is an

all-encompassing term which includes: personal/contact/payment information, orders, recipes, and food inventory.

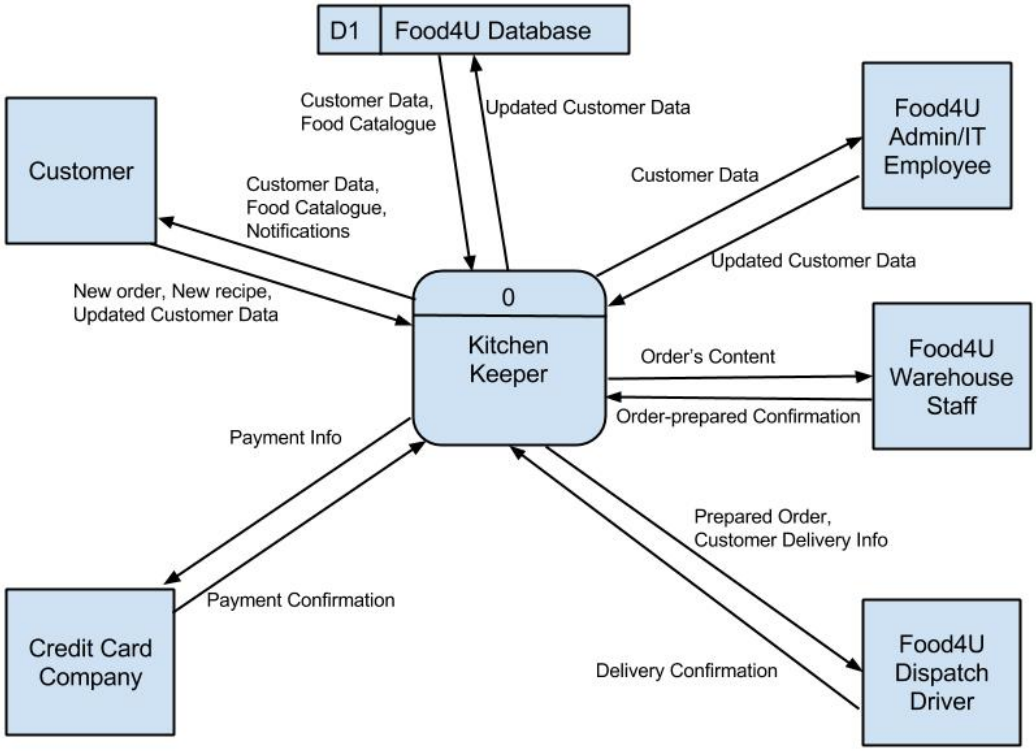

Figure 8.2a: Context Diagram

8.2.2 DFD Level 1

This diagram focuses on the Ordering data flow (Figure 8.2b). It follows an order through from creation to delivery. Some information is omitted from this diagram for the purpose of maintaining simplicity and clarity:

- The Food4U Admin/IT Employee has the ability to edit order details at any time. However, this action would disrupt the remainder of the dataflow. For example, if the order was cancelled or needed to be re-filled.
- The customer has the ability to see the status of their order any time after they have placed it (see section 7.5).

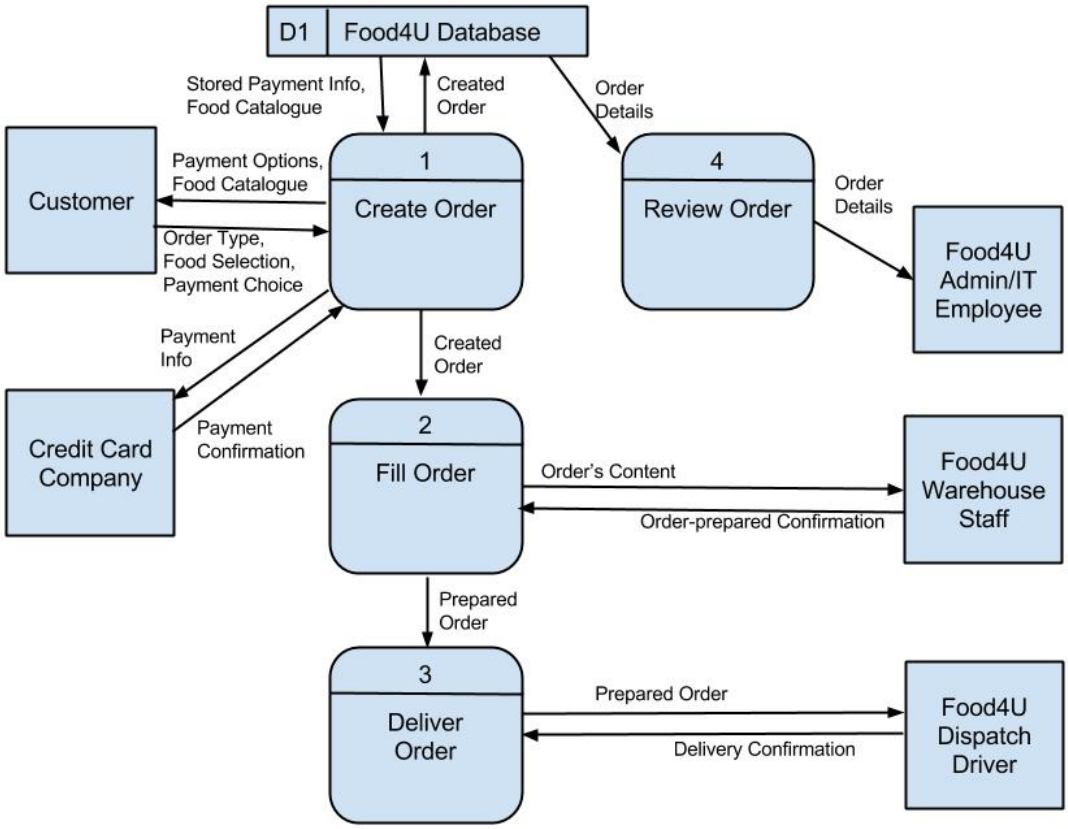

Figure 8.2b: DFD Level 1

8.2.3 DFD Level 2

Here the "Create Order" process from the level 1 DFD is expanded. (Figure 8.2c)

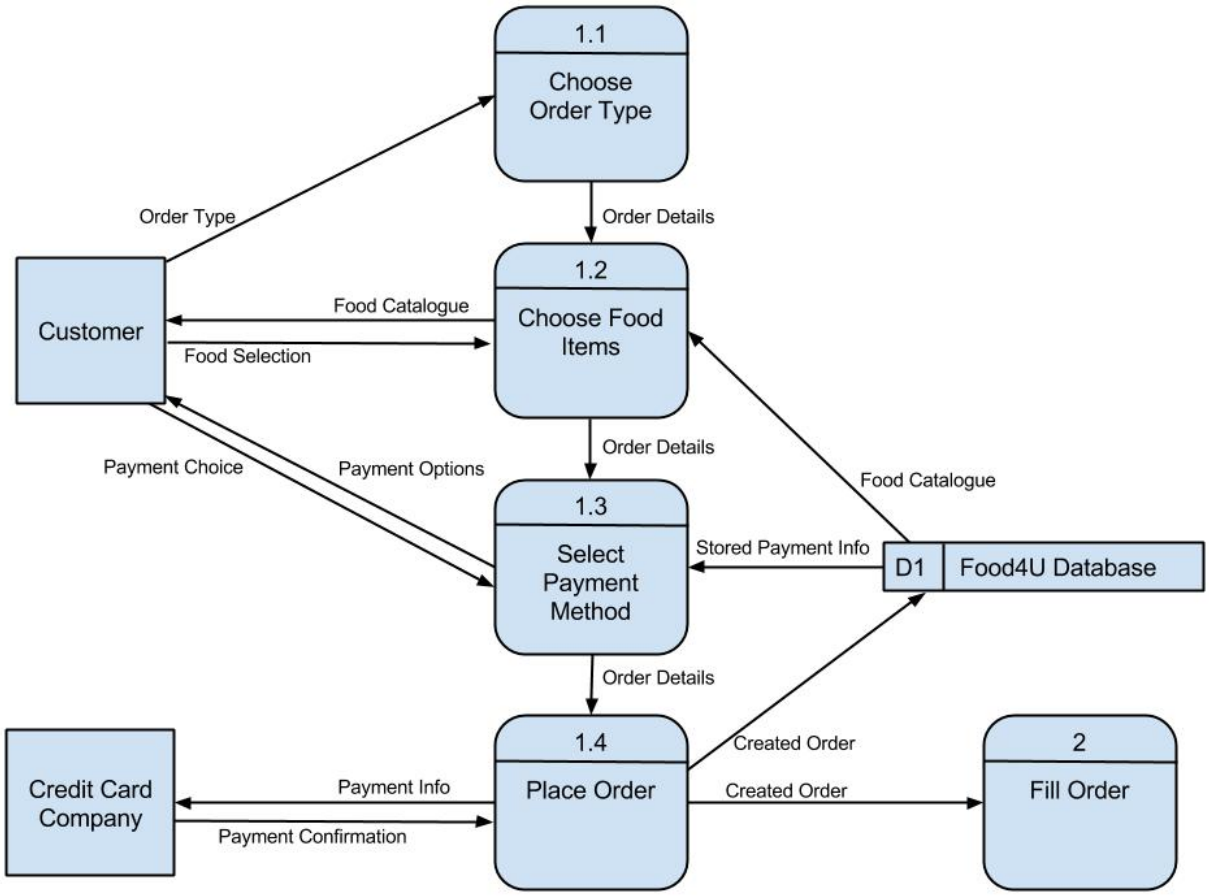

Figure 8.2c: DFD Level 2