Software Requirements Specifications
Kitchen Keeper

Killer Tech Solutions

Jan 29 2015

Contents

Revision History

Name	Date	Reason for Changes	Version
Initial Draft	22-01-15	--	0.1
Submission Draft	29-01-15	Initial Submission	1.0

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 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.

1.4 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

1.5 Overview

This specification strives to create an understanding between Food4U and Killer Tech Solutions. It contains our interpretation of the product, features, and users (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 the proceeding sections.

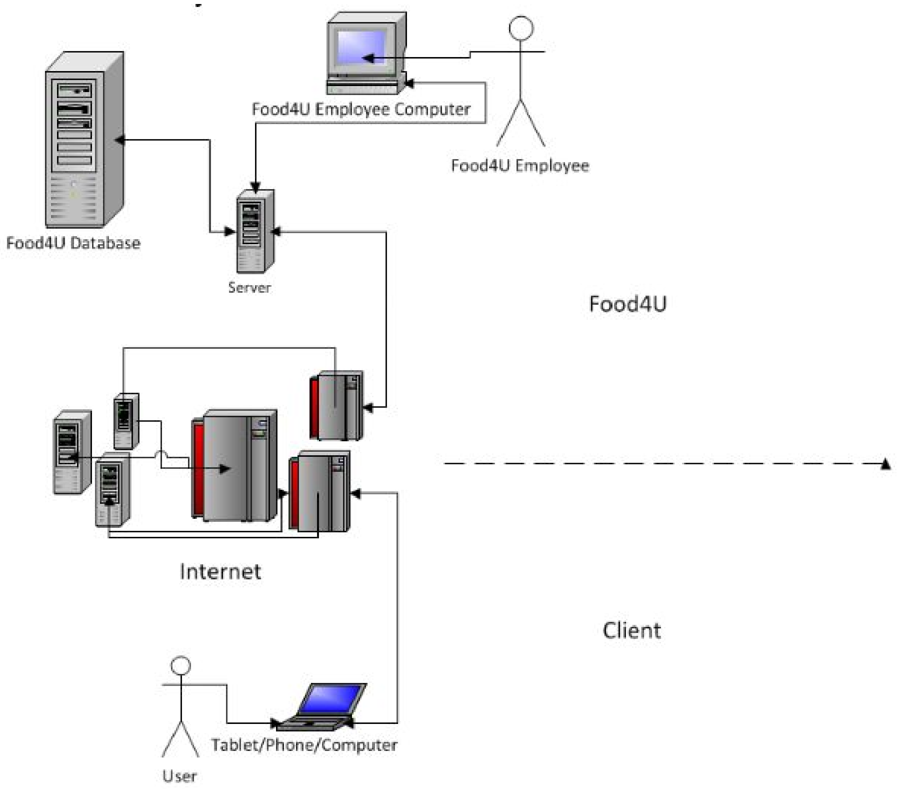
2.1 Product Perspective

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.

Diagram 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 tablet 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 would use this service most often. It is most important to satisfy businesses and the elderly. Older people account for the greatest portion of the current user base and businesses have the largest potential for expansion.

Elderly people are expected to be the primary users of Kitchen Keeper and tend to have little technical expertise. They also often have trouble reading small text. 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). 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.

The first is the company application. Food4U currently has a database so there will be a server of some kind used 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 needs to 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 does not need to have a maintainable codebase, just a final application.

3 System Features

3.1 Integration with Current Food4U Database

3.1.1 Description and Priority

Integration with an 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 system must be able to access the database at all times.

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 automatically. 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 will display all the correct information about each item.
REQ-3: The system displays all the items that the user has in his house.
REQ-4: The user’s inventory is automatically updated when an order is received.

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. The orders 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 and other information. For example, price or category. 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 will display all relevant items in response to a search.

REQ-3: The system will correctly keep track of the items that have been added/removed from the order list.

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. 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 orders will send the correct order information 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.

3.5 Create Recurring Orders

3.5.1 Description and Priority

The system will automatically create recurring orders based on current food that the user has in their home. This is a medium priority.

3.5.2 Stimulus / Response Sequences

The user will select whether or not he wants an automatic recurring order to be created. The user will select the frequency of the recurring order and it will automatically create an order based on what food the user currently has in their house and the specific amounts of foods that the user wants in their house at all times.

3.53 Functional Requirements

REQ-1: The user can turn on or off the automatic ordering system.
REQ-2: The user can create a specific list of items that they want to be recurring.

REQ-3: The system will automatically create a list of items to order based on the food in the house and the users specific order.
REQ-4: The order will automatically be sent every specific period that the user selects.

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 will automatically show an approximate expiry date.
REQ-2: The user can manually input an exact expiry date.
REQ-3: The system will alert the user when an expiry date is within a few days.

3.7 Find Recipes

3.7.1 Description and Priority

The system will enable the user to make a rudimentary web search to find recipes for use with Kitchen Keeper. This is a low 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 user can search online for recipes.
REQ-2: The system will save recipes specified by the user.

REQ-3: The system will parse recipes such that 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 low 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 user can input any ingredient from the Food4U database.
REQ-2: The user can input new ingredients not yet recorded in the database.
REQ-3: Recipes entered are stored in the Food4U 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 an 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: Recipes are to be suggested to the user from currently owned food items.
REQ-2: When creating a meal using a suggested recipe, the system will automatically delete the items used from the user's database.
REQ-3: Recipe search times out after ten seconds.

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.

- **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 function introductions):** 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 purpose.

4.2 Hardware Interfaces

4.2.1 Supported Devices

This system should be able to support common handheld devices (ie. smartphones and tablets) in different operating systems (ie. Android and iOS). 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 internet connection.

4.3 Software Interfaces

The Food4U application must provide software interfaces to:

- Food4U database(s) 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

There is no restriction on the communication protocol to be utilized by Kitchen Keeper. Users should be notified prior to an order.

4.4.1 User Experience Report Interface

There should be an experience report interface in the system for users to report their experience and any feedback after using this system. This is an option for the user to give their input and will be forwarded to the client company as well as the development team through email.

4.4.2 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.

- 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; however, logging out is not recommended and the option should be hidden.

5.3.3 Data Transmission

All user data must be encrypted during transmission over the internet, as it contains confidential information as described in 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 5.3.7)

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 reasonably 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 other languages.

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.

Appendix A: Issues List

The following topics were not clarified in the first requirements elicitation and should be discussed in more detail during the second client meeting on February 5th, 2015.

- How many simultaneous *users* is the system expected to accommodate? What is the expected maximum transactions/second? And what is the maximum reasonable response time?
- How many simultaneous *employees* is the system expected to accommodate? What is the expected maximum transactions/second? And what is the maximum reasonable response time?
- What specific order information must be accessible through the system for Food4U warehouse staff that handle deliveries?