**Software Requirements Specification**

**for**

NotWegman’s Online

**Version 1.0 approved**

**Prepared by: Thomas Rogers**

**Thomas Mease**

**Lisa Skiles**

**Adam Diemicke**

**//ToDo Ventures LLC**

**October 23, 2017**

**Table of Contents**

[**Introduction**](#_qbkrxygwbd94) **3**

[**Purpose**](#_3znysh7) **3**

[**Document Conventions**](#_2iyfo33sul2c) **3**

[**Intended Audience and Reading Suggestions**](#_g3x2ose7wgm) **3**

[**Product Scope**](#_lve1wyfu3338) **3**

[**References**](#_oimk2m655hpx) **3**

[**Overall Description**](#_xt1pl0dhqnbl) **4**

[**Product Perspective**](#_2s8eyo1) **4**

[**Product Functions**](#_vozvsocaa00w) **4**

[**User Classes and Characteristics**](#_6cic4rf59n8k) **4**

[**Operating Environment**](#_hbtofdprvty6) **5**

[**User Documentation**](#_ddw4nr8z7jj7) **5**

[**Assumptions and Dependencies**](#_awf765ox2ds3) **5**

[**External Interface Requirements**](#_b2k6jvw6l6ep) **5**

[**User Interfaces**](#_2jxsxqh) **5**

[**Software Interfaces**](#_b46u8brj12bj) **6**

[**Communications Interfaces**](#_hxmgntsw33du) **6**

[**System Features**](#_v6ik378fg6xe) **6**

[**Order Groceries**](#_ma3lbwl88uxd) **6**

[**Order Food**](#_3dylqrev8467) **6**

[**Cart**](#_3ykqz5o97ukc) **7**

[**Checkout**](#_2a0od8c2fg53) **8**

[**Login**](#_6h20joz87jf) **9**

[**Other Nonfunctional Requirements**](#_3whwml4) **9**

[**Performance Requirements**](#_2bn6wsx) **9**

[**Safety Requirements**](#_xycgacgnzvhz) **10**

[**Security Requirements**](#_kerh8lwitxp) **10**

[**Software Quality Attributes**](#_gbcqfkw1ckgw) **10**

[**Business Rules**](#_ltltl3uib2al) **10**

[**Other Requirements**](#_ufnx0shcj706) **10**

# **Introduction**

## **Purpose**

NotWegman’s Online is an online shopping system specifically for groceries and make-to-order foods. The system will contain a database of items available for purchase and a customizable food menu to order special made foods. Users will be able to add/remove items from their cart, receive a time estimate for their order, and be able to choose between delivery or pick-up. In addition, the user may have a membership or student qualifications allowing them to get discounts or free shipping. (*Version 1.0*)

## **Document Conventions**

There are no special or specific document or typographic conventions that are needed to easily and readily comprehend the SRS Document.

## **Intended Audience and Reading Suggestions**

The intended audience for this NotWegman’s Online SRS Document is the developers, staff, testers, clients, and the overall //ToDo Ventures LLC team. This SRS Documents is not intended for distribution beyond the //ToDo Ventures LLC and the clientele’s team. The SRS Document contains specifications regarding the NotWegman’s Online Program’s overall description, specifications, features, and requirements. The suggested sequence for reading is the Introduction (1), Overall Description (2), System Features (4), External Interface Requirements (3), Other Nonfunctional Requirements (5), Other Requirements (6) for all team members.

## **Product Scope**

The purpose of the NotWegman’s Online customer interface is to extend the market of NotWegman’s to customers who would not typically partake in traditional grocery shopping due to lifestyle or time constraints. The interface is designed to be easy and quick to use for both customers and employees. Refer to the attached use case diagram in Figure 1 below.

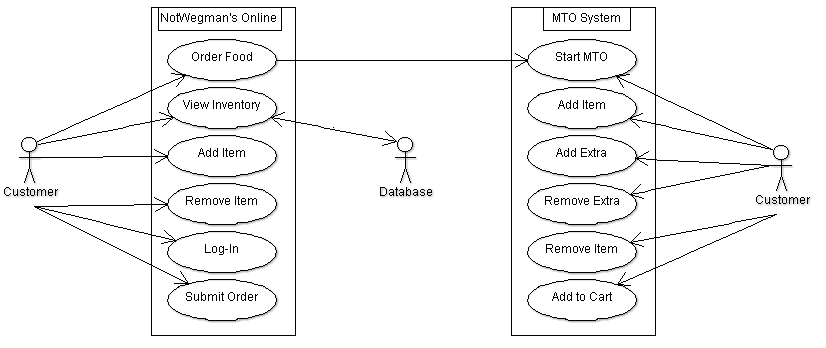
## **References**

This SRS Document does not refer to any other documents or any web addresses.

# **Overall Description**

## **Product Perspective**

NotWegman’s is a stand alone, new software and this document will describe the subsystems relating to the customer/external user’s experience.



**Figure 1 - System Environment**

## **Product Functions**

The product functions are as follows:

* The user can log into the system, which will check for username and password
* The user can add generic foodstuffs to a virtual cart
* The user can custom-order specialty foods, with each addition having a price
* The user can check out and pay, choosing pickup or delivery
* The program will print out a final price and time until the order is ready

## **User Classes and Characteristics**

The //ToDo Ventures LLC team anticipates new and old customers to utilize our product. Subsets of the customer include members, non-members, and students. All customers will be able to utilize the grocery ordering and MTO food services. Members will be able to enter in their member ID and receive discounts. Students will be able to reference their school and receive free shipping. Non-members will not receive any special features.

## **Operating Environment**

The operating environment for NotWegman’s Online will be exclusive to recent versions of Windows and Mac, as they require usage of Microsoft Excel’s files, which are not supported directly on Linux or Ubuntu.

## **User Documentation**

There will be a brief user manual provided with the software to allow for easy use. Please refer to the document for further information or assistance.

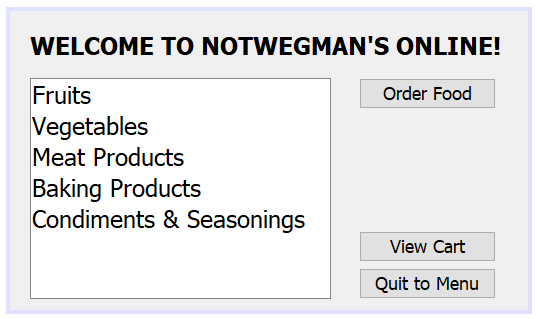
## **Assumptions and Dependencies**

It is assumed that NotWegman’s already has in place a system for employee use and delivery.

# **External Interface Requirements**

## **User Interfaces**

An early of example of the User Interface is included below:



The interface is made to be very user-friendly, and contains a list of items, and labeled buttons that are designed to easily navigate the user to what their intended action is, including “Order Food,” “View Cart,” and “Quit to Menu.”

## **Software Interfaces**

One of the tools NotWegman’s Online will be utilizing will be a tool that enables the program to read through an excel spreadsheet that contains the data for the grocery items. The Jar file gives the developers ability to parse and search through data with ease. The data will exist offline included in the program’s files. It will be updated whenever a software update is available.

## **Communications Interfaces**

There are no online communications that exist for the program.

# **System Features**

## Order Groceries

* + 1. View Inventory
       1. The user will be able to view all the items contained in NotWegman’s inventory. This is a high priority function and all other functions pertaining to ordering groceries will stem from viewing the inventory.
       2. When the user starts to use the program, they will instantly be shown the inventory.
       3. REQ-1: The software must be operable.

REQ-2: The software must be up to date. This will ensure that the correct inventory is displayed

* + 1. Add Items to Cart
       1. While viewing the inventory, the user will be able to add an item they want to purchase to the cart. This is a high priority function.
       2. When viewing the inventory, there will be an “Add to Cart” option for every item listed. The user will be able to press this button and the item will be added.
       3. REQ-1: The item must be in stock.

## Order Food

* + 1. Add Item
       1. The user will be able to add an item to their food order from a database of items available from NotWegman’s. This is a high priority function.
       2. Upon being shown the grocery inventory, there will be an option to view the food menu.
       3. REQ-1: The customer must be able to add multiple items to each order.
    2. Add Extra
       1. The user will be allowed the option to add extra features to their item at an additional cost. This is a low priority function.
       2. These options will be presented when the user selects the item to add to their order.
       3. REQ-1: There must be a base item present to add an extra to.
    3. Remove Extra
       1. The user will be able to remove any extra features they may have added to their order, subtracting its cost from the total price. This is a low priority function.
       2. This option will be present while the user is selecting their additional features for the order.
       3. REQ-1: An extra must have been added to an item to remove it.
    4. Remove Item
       1. The user will be able to remove an item from their food order, subtracting its cost from the total order. This is a high priority function.
       2. This option will be present when the customer reviews their food order before adding it to their cart for purchase.
       3. REQ-1: An item must have been added to the order to remove it.
    5. Add to Cart
       1. The user will be able to add their completed food order to the cart. This a high priority function.
       2. This option will be present after the customer starts a food order.
       3. REQ-1: An order containing items must have been created to add it to the user’s cart.

## Cart

* + 1. Edit Item
       1. The user will be able to change a custom item that already exists in the cart or change the quantity ordered. This will be a low priority function.
       2. The user will access their cart and be able to click edit for the custom ordered food or be able to click on a drop down box allowing the user to choose quantity they would like to order.
       3. REQ-1: An item must be present in the cart to edit it.

REQ-2: The item count cannot exceed the amount in stock.

* + 1. Remove Item
       1. The user will be able to remove an item from the cart. This is a high priority function.
       2. The user will access their cart and click a button that will remove the item from the cart.
       3. REQ-1: An item must be present in the cart to remove it.

## Checkout

* + 1. Login
       1. Checks the user’s inputted Username and password against the database. This has moderate priority.
       2. The user will click the checkout button while viewing the cart and be prompted to enter their login credentials.
       3. REQ-1: There must be an item present in the cart to checkout.
    2. Continue as Guest
       1. Allows the user to bypass the login function. This is a high priority function.
       2. When prompted to login, the user may select the “Continue as Guest” option. This will send them right to choosing between pickup/delivery.
       3. TBD
    3. Pickup/Delivery
       1. The user will choose pickup to pickup in store or delivery to have the order delivered straight to the user’s home. This is a high priority function.
       2. After entering in their login credentials or selecting continue as guest, the user will choose between pick-up or delivery.
       3. TBD
    4. Submit Order
       1. The user will click a submit button. The system will then calculate the total price based on their login credentials and print a receipt.
       2. After choosing pickup or delivery, the user will be able to submit the order and provide payment information.
       3. TBD
    5. Payment Information
       1. The user will enter their payment information, choosing between credit, debit, or cash. The user will provide card information or whether they will give cash upon receiving their order.
       2. The user will select credit, debit, or cash from a list of options. If credit or debit is chosen, the user will enter their card information. If cash is chosen, the user will provide a cash amount upon delivery/pickup.
       3. TBD

## Login

* + 1. Enter Username/Password
       1. The user enters a username and password, which are then checked against the database of user information. This is a moderate priority function.
       2. When checking out, the user will be prompted to enter in their login credentials. This can be bypassed by choosing “Continue as Guest.”
       3. REQ-1: The user must have an account with their data stored in the database.
    2. Enter Student ID
       1. The user enters their student email. If it contains an email domain that is in the databases to check against, they will receive TBD perks. This is a moderate priority function.
       2. When checking out, the user may enter a valid college email address to receive TBD perks.
       3. REQ-1: The college domain must be contained in our database.

REQ-2: The email must be a valid college email address. This will be checked through a verification email.

# Other Nonfunctional Requirements

## Performance Requirements

The product must be capable of performing several tasks related to the purchase and shipment of the items in the NotWegman’s inventory. This includes the ability to confirm usernames, passwords, and emails, as well as the ability to store as many items in the virtual cart as the user desires and calculate the price of the order correctly, taking into account the possible discounts.

## Safety Requirements

Potential consequences of an online ordering system are that the package could end up late, lost, or damaged, the customer could accidentally order items not in stock, or delivered items could be incorrectly assembled. To protect NotWegman’s from liability issues, a disclaimer and waiver should be required to proceed with purchase, and employees should be trained to manage situations that may arise.

## Security Requirements

A separate database holding the usernames and passwords of all known users will be used to keep users anonymous. The system will ask for usernames and passwords of all non-guest users as to keep purchases secure.

## Software Quality Attributes

The system will aim to be easy to use and understand, allowing the user to easily search for foods and add them to their order. The system will allow for ease of access and the user will find the system intuitive and easy to use on return visits and updates to the system.

## Business Rules

The customer/user will access the entirety of the interface. The interface will connect to the rest of the system, including the database of foods and the build-your-own food segment. The user will under no circumstances be permitted to see through the facade and into the rest of the program, but the facade will access the rest for the user.

# Other Requirements

The NotWegman’s Online project does not have any other requirements or needs other than the ones explicitly mentioned within the SRS Document.

**Appendix A: Glossary**

There are no special terms or phrases used in the length of this SRS Document that would require any specific defining.

**Appendix B: Analysis Models**

See section 2.1 for a use case diagram of the system.