# Software Requirements Specification
# Team: OpenCrowd

## 1. Introduction

LettuceBuy is a web based application which provides convenience for customers as well as an opportunity for “drivers” to make extra money. Due to the limited amount of time that we have for this project, we will be limiting our functionality which will be explained further into this document. The purpose of our project is to allow customers to order online and maintain a text-based communication with the driver for more instructions/items. Both customers and drivers will have their profiles stored onto a server database which is also where the ratings for each driver resides.

### 1.1 Purpose

Time management is one of the biggest concerns that people currently have. Students, professors, workers or any other person would rather want to avoid doing groceries to be able to do other activities that might be more productive or relaxing in general. LettuceBuy is an easy to use that aims to provide a more convenient way of obtaining groceries and a new way of marketing for farms and locally grown products. This software will allow clients to post their lists online and have fresh and healthy groceries delivered to their door step. This will be a very beneficial route to providing people with cars an easy opportunity to earn extra money in their free time. They will work for the company and receive money based on how far they had to go and how many items were needed to be purchased. LettuceBuy will provide them instant updates of newly posted lists and provide them with the ability to constantly give updates to the user while a list is being worked on.

This document will describe features and requirement specifications of the LettuceBuy. Providing a guideline for the development process, and will be revisited during the verification stage and after each prototype is completed as we are using Agile development strategy combined with an iterative and incremental approach as prototypes are central to our website’s ability to succeed.

1.2 Scope

*In Scope:*

a. Register/Login scripts with proper redirection based on whether the user has an active list at the moment.
b. A convenient way of inputting new lists for clients to use.
c. Update order before list is fetched by drivers. Once fetched, disable ability to update lists.

d. Drivers should be able to fetch list or drop it once they have fetched it. The dropped list should go back to the pool of lists
e. Driver has buttons to update status of list so client is constantly informed of what is happening. Driver has the ability to confirm transaction which prompts the client to testify that they did receive the groceries. Driver is frozen until client presses a button to confirm.
f. Update credentials for both client and drivers. Usernames and passwords cannot be changed.

*Out of Scope:*

a. No banking or actual transactions will take place on the website.

# 2. User Requirements

## 2.1 Software Interfaces

Database containing all information of a user account such as username, password and home address. Our website will be designed using PHP for the backend server side with a sqlite3 database housed on a Google Cloud platform. We will be using HTML and CSS for the styling for the web pages in order to improve the GUI. Sqlite3 will be used in order to communicate with the database in which all of the profiles of customers and drivers as well as current shopping lists are stored. To make this possible, we will use PHP to link the database and the user interface.

System should able to store list and broadcast it to the drivers. Once a list is fetched, the list will be taken down from the pool of lists to prevent duplication or multiple drivers fetching the same list bug. Another important interface will be used to start a three way confirm of success and marking the transaction as delivered. The clients must have an easy time inputting their items onto the cart as well as deleting if necessary.

## 2.2 User Interfaces

Registrations links will be easily accessible (top right of the index page). List submission will involve a simple form with information like items (empty list cannot be submitted), locations of store (can be left empty if no preference). The driver will automatically receive the address of the client for where Interface for both drivers and clients will be similar with some minor differences. Drivers will have buttons that are mobile-friendly that can be used to inform the client where they currently are with the overall transaction. After successful transaction, users can tap the finish button and a three way confirm will be established.

## 2.3 User Characteristics

This website is intended for any adults and teenagers looking to save time when it comes to grocery shopping. Initially we will be assuming that our users are on desktops although we will be providing everyone with a mobile-friendly version of the website if given time. But anyone with a comparable device and slight experience with internet should be able to type in a http address to get to the site and navigate in the website using touch screen or a mouse. We will be enforcing a hard rule for all customers to have a contact information listed in the profile so customer-driver communication may occur in between service requests. All credentials during registration are required because they are essential when it comes to transactions.

Drivers must be able to know the name or recognize the picture of the items.

## 2.4 Assumptions and Dependencies

We are assuming that all payments and transactions have been paid online prior to delivery since our project will most likely not deal with any types of banking or money related transactions. We also assume that all users will know English in the beginning but there is a possibility of allowing multiple languages for the website if given time. It is assumed that drivers should have legal license plate and have decent general knowledge regarding general groceries.  User must have a device, phone or a computer to be able to access the website.

# 3. System Requirements

## 3.1 Functional Requirements

Team OpenCrowd:-

- Haoxian Lin
- Hassaan Khan
- Victor Arango
- Bao Vo

We want to be able to store profiles for both customers as well as drivers onto the server's database. Simple interface to work with (like doodle poll) for both clients and drivers. Expanding the amount of grocery stores that drivers can be asked to go to multiple grocery stores and the variety of items that the customer can choose from. Initially, the order placing during the first prototype or even the second depending upon progress as a whole, will be strictly text based which later will be covered with an interface to polish our code. We want to provide driver with the ability to tell the client where he is at in the process of delivering at the press of a button. Although our website was not made for mobile, it is quite mobile friendly.

We want to ensure the fact that the driver has flexibility as well which is why we will be making sure to include functions that allow the driver to personally select his/her own customers based on the list and how far they are willing to driver. Name of customers is not revealed until a list is fetched to maintain privacy. List ID can be shared between clients and drivers if it is a mutual transaction. All inputs are checked for validity depending upon the situation. For example, a client cannot submit an empty list to the pool of lists. There needs to be appropriate responses to various situations, such as false login credentials should return a red flag with a message displaying on the page. Security should be a priority, therefore passwords should be encrypted when being placed inside the database.

## 3.2 Non-Functional Requirements

### 3.2.1 Software Quality Attributes

The software will have the following non-functional requirements:

- Option to see detail information of any item in a specific store nearby; the specific benefits to a product and its characteristics should be displayed when a user clicks on an item.
- While users are making their shopping list, the website will suggest other items that people might buy as well (i.e. if someone picks milk in their shopping list, the website could suggest to buy cereal).
- Provide a mobile friendly website and a website that can support multiple languages.

## 3.3 Non-Implemented Functions

- GPS system was not implemented as no team member was experienced enough with Google maps API or any other to be able to implement that in time.
- Clients at this moment cannot pick their own drivers. This would have required the use of location API in order to know which drivers live close to a particular client. An algorithm of matching client's city and state strings with that of driver's was looked into but it seemed to defeat the purpose.
- Forgotten password was not implemented as other core functions had higher priority towards the end of the project.
- Chat system was not implemented even though it was originally in the plan. The point of this website is to provide fast service and chatting does not serve that purpose. If any information regarding the lists needs updating, one can simply call the opposing party by looking at the phone number displayed on screen.
- Email was not implemented in the final project due to lack of knowledge when it comes to servers. Original plan was to implement email and have user validation after registration happen through there. We also wanted to have an email sent to both client and drivers of the transaction details such as items purchased by driver, ID of list and Client/Driver info respectively.

- The driver and clients should be able to view the history of their previous orders.