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 get extra money in their free time. They will work for the company and receive money based on. LettuceBuy will give them instant updates of newly posted lists and assist them during the delivery process by gps mapping.

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.

# 2. User Requirements

## 2.1 Software Interfaces

Database containing User account information such as Username Password as well as license plate of the driver. This should be reasonably fast to ensure lists are being updated frequently and messages are being sent almost instantly. Our website will be designed using Javascript while the backend server side will written in c++. We will be using HTML and CSS for the styling of the web page. Javascript will be used for the messaging window as well as integration with google map. SQL will be used in order to communicate with the database in which all of the profiles of customers and drivers are stored and for this interaction we will use PHP to link the database and the user interface. The linking between javascript and c++ will be done using the libraries from emscripten.

System able to store list and sent (broadcast) it to the drivers. Once a list is taken the list will be taken down from the board to prevent duplication. Message sessions will be stored in the server but for users it will seem to be a two way communication. Another important interface will be used to start a three way confirm of success and marking the transaction as delivered.

## 2.2 User Interfaces

The software will be able to interact with users based on friendly interface with pictures and simple step to step instruction for each function. All functions or services provided by the server will be accessible to users in a way which a minimum amount of technical informations are provided. All users and drivers are represented by name and picture. Registrations will be on easily accessible (right next to the sign in button) however be differences for user and drivers(drivers will be required a driver's license information as well as the plate number. Gps info such as store location and delivery site will be on a map with searching engine for users to find their targeted stores. List submission will be a simple form with informations like required items, locations of store and transaction site and time limit. Interface for both drivers and clients will be similar with some minor differences. For instance, Drivers will have a functions such as fetching multiple lists on their interfaces and clients will have options like rating for a driver. Message session will be associated with each lists the session is established the second a list is fetched. 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 anyone and we will have no age restrictions. 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 soft rule for all customers to have a contact information listed in the profile so customer-driver communication may occur in between service requests.

Drivers should have legal license plate and have decent general knowledge regarding general groceries. Drivers should 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 type of banking.   
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.

# 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 where there. The server database will contain history of orders and most importantly ratings for each driver. The rating system will be implemented to ensure customer satisfaction and any other scams. We will be implementing a failsafe just so we do not lose the database in case of a server crash. Simple interface to work with (like doodle poll). 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. Also, right when a driver is assigned to a customer in the area, a text based chat will be opened between them so that there can be constant communication between the customer and the driver during the service request.

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.

## 3.2 Non-Functional Requirements

3.2.1 Software Quality Attributes

The software will have the following non-functional requirements:

* Payment transactions: This is a non-functional attribute for our project since we will not focus on how users will pay to the drivers; we will simulate some simple payment process but will not be functional this software itself. One payment method could depend on how many items, their weight, and distance that driver will need to drive.
* 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 an 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)
* Users can save shopping lists in their profile. This will be a very useful feature to save time to users that will potentially use similar shopping lists. Extremely important for families that do groceries every week/month.
* Capability of users to keep track of where their groceries are at; this attribute will use the next feature, GPS location. We can use this feature to send some “alerts” to user such as “groceries are arriving!”.
* A GPS providing the customer and the driver positions of each other will be implemented.
* The driver should be able to view at the history of previous orders by a specific customer before picking
* Provide a mobile friendly website and a website that can support multiple languages.