# **Process Plan Document LettuceBuy**

# 1. Introduction

LettuceBuy is a platform that aims to improve the overall time management of the working millennials as well as convenience of customers by providing them a revolutionary way of obtain groceries especially when time is of the essence. We also hope that LettuceBuy can provide a new way of marketing for farms and locally grown products.

## 1.1 Definitions and Acronyms

Driver: services providers, drivers, farms, retail stores.

Clients: consumers of end products.

Lists: generated by the clients including types and quantity of products that they wish to obtain.

Stores: where drivers get their products from.

Delivery site: where the transaction between driver and clients happens.

Processing time: Time starting from when drivers accepted a list to finishing the delivery.

Total time: Time from client submitting a list to the finishing of delivery.

## 1.2 Project Scope

LettuceBuy is essentially an ‘uber-like’ service creating jobs for people in an area of high demand, time management. An important feature that should not be overlooked is how immensely helpful this project will be for the disabled and the elderly**.** A vital requirement will be to provide a mobile friendly website that allows customers to order groceries online from nearby stores of their choosing through only a few keystrokes.

We will intentionally ignore part of the project related to money and secure banking as that is out of this project’s scope. We will be focusing on implementing an initial product that can meet the basic specification needs then work on adding bells and whistles to the rest of the project.

* Goals
  + Improve time management and productivity of working classmen
  + Fast responding website
  + Simple interface to work with (like doodle poll)
  + Expand to multiple grocery stores for GPS (store name will be text based at first)
  + Finish a stable product by the end of the semester
* Tasks
  + Link between server and shopping list request
  + Link between server and driver that will receive list
  + Storing images of products from local stores (will be text based in the beginning)
  + Store driver database on server to be accessed by shoppers
  + Text session between driver and customer until delivery is complete
  + Rating system for drivers only (Drivers with rating lower than 2 stars will be banned permanently, License locked)
  + Three way confirmation for each transaction, a receipt must be submitted.
  + Flag system to flag bad customers or drivers?
  + Provide a detailed manual containing instructions for customers and drivers
* Resources
  + Local store’s websites for product images
  + Location data to know where the delivery site is?
  + List of stores within the area.

## 1.3 Assumptions made

Customers that order groceries online are assumed to be adults.

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.

## 1.4 Schedule

The team consists of 4 members:-

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

Note: The dates below are subjected to change based on weekly progress. They are firm dates.

*Table 1 shows the expected number of prototypes to be made along with a estimation of efforts*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # of Prototype | Start Date  mm/dd/yyyy | End Date  mm/dd/yyyy | Total  Effort | Description of Change |
| Prototype 1 | 9/26/2016 | 10/26/2016 | 192 mh/w | Version 0.0 |
| Prototype 2 | 10/29/2016 | 11/9/2016 | 192 mh/w | Version 0.5 |
| Prototype 3 | 11/10/2016 | 11/20/2016 | 140 mh/w | Version 1.0 |

In between each of these prototypes, we will undergo heavy testing to improve our software as much as possible as we progress through. We want to eliminate bugs as fast as possible and from the beginning. Final stable product should be relatively bug free. Description of change will change as prototypes are made and tested. We want to reserve the last week or so for extensive debugging.

# 2. Process Description

## 2.1 Project Lifecycle

The lifecycle of this project will be suited to two software process models, the Iterative-and-Incremental and Agile Development model.

We will use the Iterative-and-Incremental model in first part of the project. We will develop this part of the project in different stages or increments.The stages are Requirements, Design, Implementation/code, Test Plan Document, User manual/documentation, and Version 0.0 which will be discussed in more detail in the next section. Each stage of the project can be done separately but they could be modified if there is feedback from another stage that requires modifications. For instance, if we find out that there is another component that should be added to include a new feature in the project, we could do so by going back to our requirements document. Since requirements can change throughout this semester, this model is ideal especially because not all requirements are known. At the end of the first part of the project we will have a version 0.0 from which we will start creating new prototypes that will undergo heavy testing.

The second part of the project will start when all required documents are almost done. Here we will have a very good base to work on improving the idea and will look for continuous improvements in new prototypes. We will also design different test cases scenarios where basic functionalities of our project should be met and thus a continuous change in its structure could be rapidly tested to insure its proper behavior. When the project is running, we would like to get feedback from “customers” who in our case will be peer students. After project satisfactorily work, we can make new versions of it by introducing new features that we may believe will be more appealing to the potential customers.

## 2.2 Process Activities

1. **Requirements Document**
   1. **Activity Description:**This activity is the stage of the project where all requirements are listed and focuses on what we will build. Specify software and external requirements. This is perhaps the most important activity and will be undergo multiple times during the life cycle of the software. At this stage we will not only on what’s needed but also what new features we can add to the project as well as functionality required for these new features.
   2. **Input criteria:** Check feasibility of requirements by understanding time that will be needed to develop.
   3. **Output criteria:** The completion of this document will mark every essential component of this software that needs to be implemented. This document coupled with the SRS will and our prototypes will showcase which features have been implemented in the working version of the software.
2. **Design Document**
   1. **Activity Description:** This document will contain what we design in the Design Stage of the project. Here the overall software architecture and website layout will be discussed and we will agree on its specifications. Overall structure of the software will be established. Server side architecture should be easily manageable with focus of stability. GUI should be simplistic, intuitive that users can be able to use basic functions of the app without reading any documentations. The team should also focus on increasing the marketing value in this stage, designing features that are not only useful but also economical. All team member should understand this document thoroughly to avoid future miscommunication.
   2. **Input criteria:** Software architecture needs to be very detail to avoid programming issues. Interfaces of different parts of the project will need to be stated clearly in great detail to avoid potential bug and ensure the parallelity of the project development.
   3. **Output criteria:** Designs should be planned to be accomplish in a reasonable amount of time to not exceed due dates nor end of the semester.
3. **Implementation/code**
   1. **Activity Description:** The designed software architecture from the last stage will be followed by the creation of the code from team members. Then it could be implemented to the project and be ready to test. The code needs to be easily scalable for future requirement and be able to work with large data bases.
   2. **Input criteria:** Code should match the already designed software architecture which can be modified if needed. Code should be check for at least two different team members. Documentation of the Code will be written simultaneously which include documentation of the code as well as new ideas that could be implemented in the next version.
   3. **Output criteria:** An easy to follow code with corresponding documentation to ease future testing/debugging.
4. **Test plan document**
   1. **Activity Description:** Software test cases to check basic functionalities of the project. It will have different test scenarios to check specific behaviors on the system. This task and be started after the initial requirement document.
   2. **Input criteria:** Test should check all possible scenarios to check proper behavior. The test should be a well documented code.
   3. **Output criteria:** Code that can easily test different cases and will provide results that are easy to read.
5. **User manual/documentation** 
   1. **Activity Description:** Team will write an easy to follow guide that shows all the services available for users. These documentations should be readable to users with minimal technical informations. This task should not be focused heavily in the early stage of the project.
   2. **Input criteria:** Concise and clear information of services that are provided to the users.
   3. **Output criteria:** Written document describing our service functionalities, pictures should be used for easy understanding.
6. **Version 0.0**
   1. **Activity Description:** First prototype of our project. Basic software requirements should be met. User interfaces could be simple the focus should be building a solid framework for the software, debugging heavily on the Server, User, Driver three way communication.
   2. **Input criteria:** Information from all the previous stages to integrate the first functional version.
   3. **Output criteria:** First version of service from which we will expect to get issues, feedback, and possible new improvements from team discussion and peer students. These feedbacks and Commands will be applied to the next prototype.

# 3. Roles & Management Structure

Roles for every team member with delegated tasks are listed below.

## 3.1 Team member names

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

## 3.2 Team Organization

|  |  |
| --- | --- |
| **Name** | **Role** |
| Hassaan Khan | Developer |
| Victor Arango | Developer |
| Bao Vo | Supervisor |
| Haoxian Lin | Project Manager |

## 3.3 Task List with respect to roles

The planning phase of the project should be worked on by all members to maximize creativity and productivity. The person who is responsible for a certain task has the authority when it comes to decision making and evaluating all feedbacks and comment from other members. The decision can only be vetoed If and only if all three other team members disapprove the idea. The person responsible should listen to the team. The person also has the responsibility to ensure the progress of the task.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task** | **Person Responsible** | **Estimated Effort** | **Start Date**  **mm/dd/yyyy** | **End Date**  **mm/dd/yyyy** |
| **1** | **Initial stage design and planning (requirements, overall structure).** | **Haoxian Lin**  **Hassaan Khan**  **Victor Arango**  **Bao Vo** | **30 mh/w** | **9/14/2016** | **9/30/2016** |
| **2** | **Account, Registration Database & Interface; Client & Driver** | **Hassaan Khan**  **Victor Arango**  **Haoxian Lin** | **25 mh/w** | **10/1/2016** | **10/7/2016** |
| **3** | **List: Communication with server and interface** | **Bao Vo**  **Hassaan Khan** | **9 mh/w** | **10/8/2016** | **10/15/2016** |
| **4** | **Location sharing, Map integration GUI Design: client site** | **Victor Arango**  **Bao Vo** | **15 mh/w** | **10/16/2016** | **10/21/2016** |
| **5** | **Web interface Html and CSS styling, JavaScript** | **Hassaan Khan**  **Haoxian Lin**  **Bao Vo** | **23mh/w** | **10/22/2016** | **10/30/2016** |
| **6** | **Message System**  **(Text session)** | **Haoxian Lin**  **Hassaan Khan**  **Bao Vo** | **15 mh/w** | **10/31/2016** | **11/7/2016** |
| **7** | **Searching algorithm and automatic tree generation** | **Hassaan Khan**  **Victor Arango** | **25 mh/w** | **11/8/2016** | **11/11/2016** |
| **8** | **System safety and reliability (Backing up database)** | **Haoxian Lin** | **5 mh/w** | **11/12/2016** | **11/14/2016** |
| **9** | **Testing** | **Haoxian Lin**  **Hassaan Khan**  **Victor Arango**  **Bao Vo** | **28 mh/w** | **11/15/2016** | **12/05/2016** |

These dates are firm and are subjected to change whenever the team deems it necessary. We will however, be going through these set of steps in order to complete our project.

# 4. Estimates

## 4.1 Effort Estimate

Each team member is expected to work at least 12 hours each week. After visually planning and describing interfaces, the project should have at least three individual parts that can be worked on in parallel. We expect the initial stage of the project to be slower for team members to obtain the fundamental knowledge needed for the project such as understanding the basics in UNIX, C++, and network programming. The estimation of efforts given above in previous sections are rough estimates and can change depending upon new requirements that come up and various other factors.