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 all lists ever added in the database
- ○ Store driver database on server to be accessed by shoppers

Originally thought that a brief chat session between the client and the driver would be useful but as the project developed, it seemed like time was of the essence and a chat session would actually only contradict our earlier statement. The client can simply call the driver and inform any new changes that need to be made to a shopping list.

- ● Resources/Design
  - ○ PHP for backend and HTML/CSS for frontend
  - ○ Slight use of Javascript for specific functionalities regarding inputting new lists

Our main plan was to use C++ for the backend but after doing some research, PHP was the obvious choice as it was more appropriate for our project.

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.

We are assuming that both parties are on PCs and not mobile. (The website is very mobile friendly even though it is not coded for any mobile devices)

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 an 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 0.1
Prototype 3	11/10/2016	12/13/2016	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.

First prototype will include basic skeleton of code without considering any significant cases. Login and registering will not take into account any already existing usernames in the database. All of the pages will be raw PHP files with no GUI in sight. The second prototype will take into account a lot more of the cases with an improvement to GUI. A lot more functionality will be provided such as extra required credentials during login. Clients will be able to submit lists and drivers will be able to 'fetch' the lists being uploaded onto the database.

The last prototype should be a working stable product and should not deviate too much from its original purpose. There will be changes in the in the overall design and some intended features will be deemed unnecessary as the semester moves forward.

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**
   a. **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.
   b. **Input criteria:** Check feasibility of requirements by understanding time that will be needed to develop.
   c. **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**
   a. **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.
   b. **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 parallel of the project development.
   c. **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**
   a. **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.
   b. **Input criteria:** Code should match the already designed software architecture which can be modified if needed. Code should be checked by 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.
   c. **Output criteria:** An easy to follow code with corresponding documentation to ease future testing/debugging.
4. **Test plan document**
   a. **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.

b. **Input criteria:** Test should check all possible scenarios to check proper behavior. The test should be a well-documented code.
c. **Output criteria:** Code that can easily test different cases and will provide results that are easy to read.

5. **User manual/documentation**
a. **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 information. This task should not be focused heavily in the early stage of the project.
b. **Input criteria:** Concise and clear information of services that are provided to the users.
c. **Output criteria:** Written document describing our website functionalities, pictures should be used for easy understanding.

6. **Version 0.0**
a. **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.
b. **Input criteria:** Information from all the previous stages to integrate the first functional version.
c. **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	Input shopping list (improved)	Hassaan Khan	8 mh/w	12/1/2016	12/2/2016
4	Driver side fetch with multiple cases	Hassaan Khan	3 mh/w	11/05/2016	11/05/2016
5	Driver side confirm transaction	Victor Arango	4 mh/w	12/8/2016	12/9/2016
6	Driver side status changes	Hassaan Khan	3 mh/w	12/9/2016	12/9/2016
7	Updating credentials	Victor Arango	7 mh/w	12/1/2016	12/2/2016
8	List: Communication with server and interface	Bao Vo Hassaan Khan	9 mh/w	10/8/2016	10/15/2016
9	Web interface HTML and CSS styling, JavaScript-Integration	Hassaan Khan Bao Vo	23mh/w	12/8/2016	12/10/2016
10	Testing	Haoxian Lin Hassaan Khan Victor Arango Bao Vo	28 mh/w	12/12/2016	12/14/2016

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, PHP, 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.

5. Quality Plan

The quality control process consists of the following:-

SRS Review – This document is essential to our project as it determines what languages the front end and backend should be in. It will be reviewed by every team member.
Design Review – The overall design will be much more clear after the second prototype is made as it will include key features of this website. It will be reviewed by the project manager.
Unit Testing – Each team member is to thoroughly test any new feature that they implement in order to minimize problems during integration. Travis will also be used in order to perform test cases on each test.
System Testing – System testing will be done through the use of two devices in order to make sure that the interaction between client and driver is working through a server.

6. Project Tracking

In order to monitor and record all changes to our project, we will be using GitHub version control system. This will be essential when it comes to parallelizing our work and integrate all working functionalities before each prototype. All unit testing will prior to integration and all working code should be pushed onto GitHub as soon as possible.

7. Project Reflection

Overall, the project was successful as core functionalities with apt responses to multiple situations were implemented. What we would do differently is work a lot harder on the design document as it is one of the most important documents out there. That document defines the starting point which is the hardest step of a project. It also sets the work flow of the entire group which is essential in a cooperative environment. We needed to have done some more research to have a better time in implementation as none of us had any experience in Javascript or Ajax

which would have improved our ability to implement certain core functions that had to be completed the long way. Our lack of knowledge in networks disallowed us to have our Google cloud server send an email to another email address which could have served an important role to finish transactions as mentioned earlier as well as for validation purposes. We needed one more prototype between the second and the last in order to sort all bugs out.