CS 4800 - Software Requirements Specification

I. Overview

Company: Legumes, LTE - Nicole Casartelli (CEO), Luis Aldeco, Ryan Atienza, Selena Aungst, Matthew Plascencia

Client: Professor Edwin Rodriguez

Product: Chickpea - Online grocery order and delivery service

II. Introduction

For this product, Legumes, LTE was requested to make a product that solves many issues presented by other online grocery delivery systems. Other services are difficult to use, do not allow customers to order from multiple stores, have poor quality mobile browser translations, or have any number of other issues that arise when using popular services of this kind. For this Spring 2021 release product, we were tasked to create a product that solves these problems, and Chickpea will be our solution.

III. Background

Chickpea is an online grocery order and delivery service that allows customers to order groceries from participating stores in their area and receive their order on their doorstep or have it brought to their car at the store(s) of their choice. For our delivery option, customers can purchase items from multiple stores in one order and it will be seamlessly handled and delivered by one driver as one order. For our pickup option, customers can schedule multiple time windows to pick up their order from each store at which they purchased items. Chickpea will have an attractive and intuitive design that is responsive to any screen size it is loaded on, making it perfect for ordering online at home, or checking delivery updates on mobile on your way to the store.

IV. Project Description

Chickpea will have users at several different levels: “Customers”, “Stores”, and “Handlers”. Customers and Handlers will interact with the frontend of the site, purchasing items and receiving updates about order delivery. Stores will interact with the database, updating inventory and pricing as well as receiving order requests.

Product Users	
Customers	Purchasing groceries
Stores	Grocery vendors
Handlers	Shopping for / delivering groceries

Product Use Cases	
Update inventory	Update delivery status
Update Handler status	Update Handbasket when item is added/removed
Update substitutions	Update account information
Update an order	Search Categories
Search Stores	Search Items
Cancel an order	Create account
Alert store of order	Aggregate purchases from multiple stores

---

V. Scope/Audience

Chickpea has a specific target audience in mind, modern mobile users who want to take advantage of, or require, the convenience of grocery delivery. The product will allow users to place complex food orders from a variety of grocery stores on

Client Initials: [illegible]

the go. It caters to the users who may be busy and who value simplicity. The UI will present a modern, simple design that has a younger audience in mind, although it will be simple enough to be accessible to older users.

VI. Budget

Chickpea's budget will primarily be dedicated to website hosting and maintenance and maintenance. Legumes, LTE will be compensated in deserving grades.

Service Names	Description	Cost (Monthly)	Cost (Annual)
MongoDB Atlas	Tier M60, chosen for an expected simultaneous user base of 25,000, with room for occasional usage spikes. Database infrastructure and server	$2,847	$34,164
Glitch Premium	Host for Chickpea's front end, there is only one Premium Tier		$96
Domain Name	*Chickpea.shop*, a custom domain name. For the purposes of this demo the domain will instead be *chickpea.glitch.me*		$60
			Annual Total: $34,320

VII. Requirements

The following is a collection of all the expected functions of Chickpea:

FUNCTIONAL:

FR-1: The system will validate a Customer login in response to a login attempt

Client Initials: [illegible]

- Chickpea must be able to receive a Customer’s login information, encrypt the data, and store the data for later use. When a Customer attempts to login, Chickpea must be able to reference the database in order to either accept or deny that particular login attempt.

FR-2: The system will recommend stores based on the Customer’s location

- Once the Customer is logged in, the system will detect their location (assuming the Customer gives the product adequate permission) and will scan the area around a given radius (that the Customer can decide within reason) for possible grocery stores. Once the grocery stores have been located, the system will present the list of available stores to the Customer.

FR-3: The system will filter items in response to Customer input in a search bar as well as based on predefined criteria

- Once the Customer selects a Store, they will be presented with the Store’s current inventory. The Customer will also have the option to search for particular items using a search bar. If a desired item is not found by the store, the system will analyze the Customer’s keywords and suggest related items or another store that may carry that product. Along with a search bar, the Customer may also simply filter a Store’s products by category like vegetables or dairy. In order to do this, the system will have to store products with certain labels. New products will have to be tagged.

FR-4: The system will update item stocks accordingly whenever purchases are completed or stores are restocked

- Live updates will be sent to the system whenever a Customer makes a purchase. This update will take into account what the Customer bought and the quantity and will deduct that much from the product’s stock. Inversely, if a store gets a new shipment, that will also be included in the system.

FR-5: The system will update a Customer's Handbasket in response to a Customer adding items to it and will combine transactions from multiple stores into one simple transaction for the Customer

Client Initials: ER

- The system will calculate all the corresponding taxes and shipping costs (assuming the order is for delivery) so that the customer only sees a single total cost, regardless of how many stores they may have shopped at. More specific data, such as cost for ordering at each store, will be provided if the customer desires.

FR-6: The system will log a transaction in response to a Customer completing a purchase

- Customers will be able to see their "receipts" from former purchases and keep logs on how much they have spent over time.

FR-7: The system will allow Customers to select a driver to complete their delivery

- If the Customer wishes to choose a particular delivery driver, they will be able to select from the ones who are active at the moment and nearest to the store.

FR-8: The system will display delivery data, maps, and ETA

- Once the Customer has placed their order, they will be able to see the route of the suggested route that the delivery driver will make as well as their expected arrival time (ETA).

FR-9: The system will allow Customers to set favorite items and stores

- The system will allow for Customers to "star" or "favorite" their favorite products. All of their favorite products will be kept in a separate tab in which they can also see what stores near them offer it.

FR-10: The system will allow Customers to edit their account info if necessary

- Customers will be able to change certain aspects of their account like preferred address or nickname. Other areas, like email or age, cannot be changed.

FR-11: The system will allow Customers to cancel orders

- In the event that the Customer wishes to cancel an order, the system should issue a refund if a transaction was processed and mark it as cancelled. Stores' item stocks must be updated to reflect this cancellation.

Client Initials: ER

NONFUNCTIONAL:

NR-1: The system must have a record of its Customers and relevant information such as their names, addresses, and payment information

- In order for the system to associate its Customers with specific transactions, it is necessary to maintain a list of registered Customers. This record will also be needed when completing deliveries.

NR-2: The system must have a record of its Handlers and relevant information such as their names and locations

- A record for registered Handlers and their relative locations is necessary as it enables the system to offer suitable suggestions that are nearby a particular Customer.

NR-3: The system must have a record of the stores it interacts with and their respective inventory statuses

- It is necessary to track the inventories of the stores the system interacts with in order to provide the most accurate and up-to-date stock counts for items. Customers should not need to worry about item miscounts.

NR-4: The system is quick to load on mobile devices

- Customers will be relying on Chickpea to obtain their essentials, and so the system must load and be ready for use within 5 seconds.

NR-5: The system's UI is optimised for mobile devices

- Chickpea aims to improve on the unsatisfactory Customer experience the client has experienced with similar grocery delivery apps on mobile. Thus, the system's UI will primarily be optimized for mobile devices to encourage its use on those devices.

NR-6: The system will have a brief loading screen in between different pages

- A loading screen between pages allows the system to give Customers indication that they are making progress within the app. Loading screens should last no more than 5 seconds.

NR-7: The system is accessible for the visually impaired via descriptive alternative text for images

Client Initials: ER

- Some Customers may have visual impairments and thus rely on screen reading technology to navigate applications. Alternative text for images will aid those Customers in that process.

VIII. Technical Specifications

The system will be purely web-based and accessed through web browsers, which dictates that JavaScript will be the primary programming language that drives the client side. It will rely on the Bootstrap CSS framework to build and style the graphical user interface.

The system will be hosted by the web hosting service Glitch. The web host will be configured such that it will make use of the Node.JS runtime environment and Express.JS web application framework to handle communication between the client (the web browser) and the backend of the system (the database).

The system uses the database service MongoDB Atlas to handle storing relevant data pertaining to the system's Customers, grocery store inventory and logistics, deliveries, and transactions. The MongoDB Atlas database will reside on and be maintained by AWS.

Legumes LTE will primarily use the Visual Studio IDE to develop the system.

IX. Signatures

Client Signature: Edwin Rodriguez Date: 2/26/2021

CEO Signature: Nicole Casartelli Date: 2/26/2021

Client Initials: ER