The University of British Columbia

L o c a l l y

Project Requirements, Vision, and Scope

Angy Chung

Anna Gudimova

David Jung

Mo Kiani

Andy Lin

Alena Safina

November 18, 2016

Table of Contents

Product Vision Statement

Locally will allow users to view nearby vendors, locally grown produce being sold at farmer's markets, and check what's in season. The data will come directly from the vendors themselves as they can update their daily produce stock through our application. It differs from other solutions as it is a mobile application which allows convenient, easy, on-the-go checking and updating.

Problem Statement:

The problem of	Not knowing what is in season or where to purchase local produce if/when it's in stock
affects	People who wish to purchase produce grown locally that supports local farmers as part of a healthy lifestyle, and the farmers and vendors who want to reach this demographic
the impact of which is	Having more barriers, hassle in finding local, healthy foods; vendors and farmers miss profit opportunity
a successful solution would be	A mobile platform where consumers can browse produce for nearby local farmers markets and pick out healthy alternatives

Product Position Statement:

For	Android users
Who	eat local/organic foods or sell them
Our system	is a software mobile application
That	will connect consumers and vendors of locally grown produce
Unlike	other solutions which are mainly web-based or other mobile applications which are not based in Vancouver
Our product	is a user-friendly mobile platform that is portable and easily available on the go whenever customers or vendors wish to use it

Project Overview

Users: The users of Locally fall into two categories: consumers and vendors.

***Consumers**:* Consumers are members of the general public who wish to lead healthy lifestyles and support local food producers. Their technology experience may vary; however, our mobile application will not expect a high level of technical expertise. It will have a user-friendly interface and should be straightforward for any smartphone user to navigate.

Consumers will want to use our application to view nearby farmer's markets and their stock and a list of in season produce.

***Vendors**:* Vendors will be farmers/vendors from the Greater Vancouver area selling fresh locally grown produce who operate at farmer's markets. They may be required to have a higher level of technical experience, but only in terms of managing their vendor account and updating their stock and business information. We plan to minimize the technical expectations required of them by providing a simple to use GUI for managing their account information. Vendors will want to use our application to directly reach their customers by providing information about their business and products for sale.

Feature List:

Consumer Interface:

- Search bar for users to search a certain type of produce and see where it is being sold
- List of farmer's markets in Vancouver displaying information such as market hours, location, whether or not it is currently open
- List of a market's vendors (this view opens up after selecting a certain market)
- Vendor's profile page that includes information such as vendor name, description, and a list of the items they're currently selling
- Calendar to display dates, locations, and times of upcoming markets with links to vendor profiles and location of markets on the map
- Map that displays all markets with indicators of which ones are currently open
- Grocery list for users to add items which they can then easily check if they are being sold at a certain market

Vendor Interface:

- Vendors can create accounts to register their businesses
- Vendors can update their stock list to reflect what is in season and what they are selling and also provide a description to consumers about their business

Constraints:

Our product will run on the Android system and therefore one of the major constraints is that our users and vendors must use Android in order to interact with our application. This means that iOS users will be unable to access our application at the moment but there may be plans for iOS support in the future. We will be using Google Maps API for displaying currently available markets and thus our project will have a dependency on their API. Our

product will also be relying on Amazon Web Services so we will be constrained by their policy for free services in the areas of storage, time, and amount of GET and PUT requests.

Scope and Limitations:

Considering our time constraint of eleven to twelve weeks, we have decided to limit the scope of our Android application to include users and vendors in the Vancouver area only and there will be room for expansion in the future if necessary. Given the time constraint, we will not be implementing features that notify the user of nutritional values of the food they are searching, although this may be an additional feature in the future. Another feature that we currently will not be supporting but can be implemented in the future is user accounts to store user preferences and comments on vendors and specific produce.

Assumptions and Dependencies:

We use AWS' DynamoDB to store information about vendors, markets and items in stock. The reason we decided on DynamoDB was that it is a non-relational database and has an excellent Android SDK. Furthermore, load balancing and growth are handled by Amazon.
In order to store vendor credentials we use AWS' Cognito. We use this to register, allow logins, and verify vendor details.
For handling image resources we use AWS' S3. This means the users will be required to be connected to initially load resources, but we are using Picasso for loading images which handles caching.

Use Cases

Use Case 1: Create a Vendor Profile

Identification: The vendor creates an account to add new stock items to their profile
Primary Actor: The Vendor
Stakeholders and Interests:

- The Vendor: The vendor creates an account to keep customers informed about what produce they have in stock

Preconditions:

- The phone is connected to the Internet to allow the app database updates

Postconditions:

- The vendor has access to their account and permissions to modify their stock

Main Success Scenario:

1. The consumer clicks “Use as Vendor” on the navigation panel and enters personal details along with the farmers market name to create a new account
2. The vendor receives a phone call from our representative to confirm their identity and allow the access to their account
3. Vendor now can access their account and has permissions to update their stock

Extensions and Alternate Flows:

Alternate Flow: Vendor Tries to Log in Before Their Identity is Verified

1. If the vendor tries to log in with their new account credentials before phone call confirmation they will not be able to do so

Open Issues:

- How will we verify vendors?

Closed: In order to become a registered vendor, we will contact them personally in order to validate their account. Upon completion we can then edit their information to verified and they can begin using our services.

- Should every vendor have the permission to add farmer’s market events to the calendar?

Closed: Vendors are not allowed that permission.

- Should we allow for vendors to delete farmer market events from the calendar? What if those events already have vendor participants assigned?

Closed: Vendors are not allowed that permission.

Use Case 2: Enter New Stock Item

Identification: The vendor enters a new stock item by selecting it from a list the system displays.

Primary Actor: The Vendor

Stakeholders and Interests:

- The Vendor: The vendor wishes to add an additional stock item for sale for their business in order to inform and attract their customers.
- The Customer: The customer wishes to view what product a vendor currently has in stock or search where/whether a specific produce item is being sold.

Preconditions:

- The vendor must have a registered vendor account/profile and must currently be signed into the system - this will give them access to personal stock and permission to update it

Postconditions:

- The new stock item will be displayed on the vendor’s page.

- If a customer searches for that stock item, the vendor's page will be shown as an available result.

Main Success Scenario:

1. The vendor tells the system to enter a new stock item.
2. The system displays the "enter stock item" screen to the vendor. The screen contains a dynamic list of fields
3. The vendor searches for their stock item within a field.
4. The system displays a list of matching stock items to the vendor, using a scatter-gather technique to display the suggestions.
5. The vendor selects their desired item. New empty fields are automatically added if the vendor has completed previous.
6. Steps 3) to 5) may be repeated for additional stock items.
7. The vendor completes entering stock items.
8. The system updates the vendor's stock information.
9. The system displays a request completed message to the vendor.

Extensions and Alternate Flows:

Alternate Flow: Delete Stock Item

1. If at line 1) vendor decides to delete an item, then
2. The vendor does the delete action on that item and the process returns to line 1) of the main scenario.

Open Issues:

- Reconsider how the system should handle finding no matching items for the vendor.

Closed: Simply displays that no vendors in the market carry those items

- How the system will handle duplicate/erroneous entries

Closed: Check to ensure that the item is not added multiple times. Erroneous entries are not possible as the valid items will already be defined.

- Should the system also allow the vendor to specify item quantity?

Closed: No, too much granularity gets in the way of use. The vendor can simply delete the item from their stock if its sold out.

Use Case 3: View Nearby Markets

Identification: The consumer searches for nearby farmers markets in their area.
Primary Actor: The Consumer
Stakeholders and Interests:

- The Consumer: The consumer wishes to find a list of nearby farmers markets in their vicinity

- The Vendor: The vendor wishes to have the markets where they sell produce to appear in the search results

Preconditions:

- The system has records of markets and their locations

Postconditions:

- The consumer has viewed markets nearby them, and now knows at a glance which are open, closed, or closing soon

Main Success Scenario:

1. The consumer tells the system to find nearby markets.
2. The system determines the consumer's location.
3. The system displays a list of markets close to the consumer's location.
4. The consumer selects a market to view more information about it

Extensions and Alternate Flows:

Alternate Flow: Handle Turned Off Location Services

1. If at line 2) of View Nearby Markets, the consumer does not have their location services turned on, then:
2. The system will prompt the vendor to turn on their location services
3. The consumer turns on their location services
4. The use case restarts at 3) in View Nearby Markets.

Alternate Flow: User Denies Location Services

1. If at line 2) of Handle Nearby Markets, the consumer denies location services, then
2. The system searches for last known location, if none found, distance features are turned off and defaults to Vancouver.
3. The use case restarts at 3) in View Nearby Markets.

Extension: User wants to view list of vendors in market

1. At line 4) of View Nearby Markets, user taps on Market name and is taken to list of vendors for that market.

Open Issues:

- How many markets should be displayed to the user?

Closed: Not enough farmers markets in the Vancouver area for this to be an issue - all will be displayed

- What market information will be displayed?

Closed: Name of market, distance from user to market if the user has location enabled, address of the market and opening hours for the day. Will be taken to List of Vendors of the market by tapping on market name.

- How the market list should be filtered or displayed (e.g. map, text list, calendar)

Closed: Main Screen markets will be shown as Cards, in side screens markets will be a list

- Adding a distance range to the search filter

Closed: All markets will be shown

- Should consumers be able to add favourites? (may require consumers to have accounts)

Closed: Consumers will not have accounts

Use Case 4: Search For Stock

Identification: The consumer creates a grocery list consisting of the items that they need.
Primary Actor: The Consumer
Stakeholders and Interests:

- The Consumer: The consumer wishes to search for the locations where they can purchase the items they are searching for.
- The Vendor: The vendor wishes to have their stock item and name appear in the search results of the customer if they have listed that item in stock.

Preconditions:

- The system has records of vendors, their locations, and the items they have listed as being in stock

Postconditions:

- The consumer has viewed vendor(s) in the farmer's market that sell the items they searched for

Main Success Scenario:

1. The consumer constructs a grocery list from an extensive list of produce and selects the market they want to search in.
2. The system displays which items are available in that market and the vendors that carry that item.
3. The consumer selects a vendor to view more information about them.
4. Steps 1) to 3) can be repeated if the user wishes to search for the items in a different market.

Extensions and Alternate Flows:

Alternate Flow: No Vendors Found

1. If at line 2) of Search for Stock, the system cannot find any vendors selling those items.
2. The system will alert the consumer of having found no results in that market.
3. The use case restarts at 2) in Search for Stock.

Open Issues:

- Should we have a "go back" button so that user could return to the search results after they see a vendor's profile?

Closed: Users should be able to return and view the search results again.

Use Case 5: Check the Calendar

Identification: The consumer checks the calendar to read market description and get directions
Primary Actor: The Consumer
Stakeholders and Interests:

- The Consumer: The consumer sees the calendar to check the upcoming farmer's market events, as well as their locations, description, and get Google maps directions

Preconditions:

- The system has records of farmer's market events, their dates, locations, description
- The phone is connected to the Internet to allow the use of maps

Postconditions:

- The consumer has viewed farmer's market dates, locations, description, and directions

Main Success Scenario:

1. The consumer clicks on the market name in the calendar view
2. The system displays a market page containing market description, location, dates it's open and directions link that gets opened in Google maps
3. Step 2) can be repeated to check more markets.

Extensions and Alternate Flows:

Alternate Flow: Consumer Chooses to Sync Market Hours

1. If the consumer wants to use their existing Google Calendar attached to their Google account, then the consumer can go to settings and select the markets they wish to add to their Google Calendar