AGROWEBPR - A WEB-BASED APPLICATION FOR CONNECTING PUERTO RICO'S FOOD SUPPLY CHAIN

By

Pedro X. Vazquez Rosario

A project report submitted in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING
in
COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2014

Approved by:

_____   _____
Manuel Rodriguez Martinez, Ph.D.   Date
President, Graduate Committee

_____   _____
Pedro Resto, Ph.D.   Date
Member, Graduate Committee

_____   _____
Pedro Rivera, Ph.D.   Date
Member, Graduate Committee

_____   _____
Lynette Orellana, Ph.D.   Date
Graduate Representative

_____   _____
Pedro I. Riviera, Ph.D   Date
Chairperson of the Department

ABSTRACT

This work presents distributed three-tier architecture based web application, named AgroWebPR, providing ubiquitous and secured access to the distribution and purchasing of agricultural products. In this era of technology it's important to be able to have communication via these means, one of which is the Internet. AgroWebPR provides a platform that lets all aspects of the supply chain to communicate, whether it is selling or purchasing a variety of products. The application is built using the PHP programming language with the back-end in the form of Apache Web Server. End users have the facility to add product that they grow (farmers) or manufacture and determine price, quantity, and availability dates, all through the use of this application. AgroWebPR accommodates farmers, manufacturers and point-of-sale locations thus, improving the supply chain communication for agricultural products. The number of users involved in selling-purchasing activities is expected to increase as more users become familiar with the application.

RESUMEN

Este trabajo presenta la arquitectura "three-tier" distribuida y basada en aplicación web, nombrada AgroWebPR proveyendo un acceso ubicuo y seguro para la distribución y compra de productos agrícolas. Es importante poder establecer una comunicación que vaya a la par con la era tecnológica que vivimos. El Internet como medio de transmisión y su servicio el World Wide Web (WWW) permite de forma sencilla el acceso remoto a varios archivos proveyéndole así al agricultor, fabricantes y puntos de venta un acceso casi inmediato a las diferentes negociones para la venta de sus productos. AgroWebPR provee una plataforma que permite a todos los aspectos de la cadena de suministro comunicarse, que se está vendiendo de determinados productos o para ordenar una gran cantidad de producto. La aplicación está construida utilizando el lenguaje de programación PHP con el "back-end" en la forma de un Apache Web Server. Los usuarios finales tienen la facilidad de añadir productos que crecen (granjeros) o fabricaron y determinar el precio, cantidad y fechas disponibles durante todo el uso de esta aplicación. AgroWebPR se utilizará para dar cabida a agricultores, fabricantes y puntos de ventas a que aumenten proporcionalmente la cantidad de productos que pueden vender en un determinado período de tiempo. Por otra parte, la cantidad de clientes también aumentará debido a la cantidad de usuarios que accederán a la aplicación web. Esta aplicación tiene la capacidad para que en un futuro puedan integrarse funcionalidades adicionales que fueren necesarias basadas en las necesidades de los usuarios.

ACKNOWLEDGEMENTS

Table of Contents

Figure List

1. Introduction

1.1 Justification

At present, over 90 percent of the agricultural products consumed in the Puerto Rican diet are imported. This represents a significant risk to food security if we think on the possibility of a major natural incident in the region, which could motivate the supplying countries to withhold their Puerto Rican deliveries, leaving the island population short of the needed goods. Much is discussed in the local news about the need to restart the island's agricultural production, which could play a significant role in increasing the island's employment figures.

On the other hand, local consumers are always searching for lower prices, which allow agricultural imports from nearby Latin-American countries to compete favorably with local products. It is believed that for local production to be reignited successfully, the following issues need to be improved upon: (a) farmer and worker know-how; (b) methods and time requirements used for planting, growing and harvesting; (c) coordination between supply and demand; (d) definition and development of agricultural product value-adding niches; and (e) coordination of supply chain activities to local and external (e.g. Mainland USA) markets.

The work to be performed in this project has a direct impact on items (c) and (e) above. It establishes communication between farmers, agricultural product value-adders, and the large and small product sellers. The web-based application should allow early conversations between farmers and customers (i.e. value-adders and

points of sale) regarding product quantity, timing and pricing. At least some of the uncertainty in demand estimation and pricing is removed, reducing risk especially for the local farmer. It is believed that this enhanced coordination is instrumental for improving competitiveness of local production against imports. This project addresses the need for an application, which can go through every aspect of the supply chain: from farmers, to the processing plants, and all the way to the point-of-sale. It should also provide buyers with the ability to search for products and their variations that are currently on demand. AgroWebPR provides an easy-to-use interface for the users of the application.

1.2 Problem

Agricultural product imports, estimated between 85 to 92 percent of the total consumption by Puerto Ricans, arrive to the island from 56 other countries. The incoming business volume breaks down as follows: 76 percent from United States, 4 percent from China, 3 percent from Canada, and the remaining 17 percent from the region (Dominican Republic, Costa Rica, and others) [9]. At the same time, Puerto Rican farmers do not focus on competition and in general lack the desire to be more productive and cost-effective, and it seems that the majority of local consumers are looking to purchase imports instead of home grown products due to price differences.

Local farmers tend to feed their family with their own harvest. With the assertive flow of food imports it is a steep challenge for local farmers to compete

with imports pricing. And as mentioned, there is no effective way to communicate through the local supply chain. Typically the farmer sells to a distributor, who goes with the product to the point-of-sale to negotiate quantity and price, but only after the product is loaded on the truck. These last minute rush negotiations are highly uncertain and on occasions forces the farmer to lose money. The negative experience might motivate the farmer to replace the current product, with very little understanding about the new supply and demand realities.

1.3 Proposed Solution: AgroWebPR

AgroWebPR is a web-based application that will establish the proper links from farmers, to processing plants, to distributors, and to the point-of-sale. The goal is to build an application that would provide a proper way for everyone to communicate, eliminating the inefficiencies that currently exist. Besides providing elegant search schemes for products and their variations, the application also focuses on providing a proper visual searching mechanism. Figure 1.1 presents the home page for the web site.
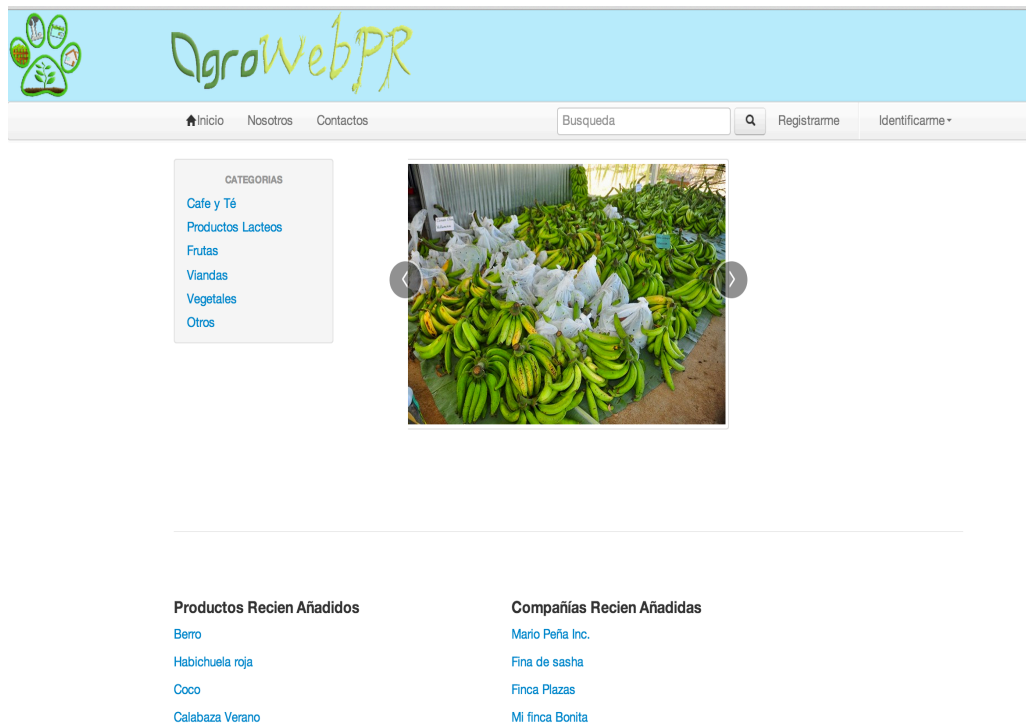
Figure 1: AgroWebPR Interface

The application features open source packages for scalable distributed computing. The web comments provide the user interface whereas the database management system stores/executes user profiles and product information.

Users will be able buy products pertinent to location, which can be identified by an address on a Global Positioning System (GPS); a mapping application is used to enhance the search and to display the results. Furthermore, to enhance the search and the display the results, users have the ability to search for products, product types, locations, and business names. The application shall be scalable with the usage of web services for data transfer.

1.4 Contribution

AgroWebPR aims to provide a tailor-made solution outfitted according to the need of the agricultural sector to add, maintain and securely access all the product data. This project aims to contribute in the following aspects:

- Establish a way for farmers to have a direct link to the processor, distributor, and point-of-sale via presenting their product information.

- Easy-to-use web application to create a link for everyone in the supply chain

- Secured access to farmer data

- Visual representation of the availability of products for extensive periods of time, taking into consideration planting-to-harvesting lead-times

- Visual search for supply and demand results for available products and their variations by location (i.e. town or city)

- Flexible application architecture for scalability

- View and analysis of historical data on harvest and sales by region

- Increase in local agricultural production (which minimizes imports) given the ability to identify supply sources, quantities, and to negotiate pricing early.

1.5 Objectives

1.5.1 General Objectives

This project aims to help the planning and growth of the agricultural sector by creating a scalable web application on top of it using PHP and JQuery.

1.5.2 Specific Objectives

- To provide the farmer crop and harvest information and have it available to be seen by potential buyers.

- To design a database in MySQL to store the data from all the members of the supply chain and configuration data for the software application.

- To design a PHP web application to provide a flexible and scalable framework for development.

- To authenticate the end users to assure secured access and privacy to all the business proprietary data.

- To provide visual search and show results using Google Maps and JavaScript to provide location information.

1.6 Outline

Chapter 2 provides an outlook into the technologies being implemented to appreciate the application and compare to other works of similar natures. Chapter 3 describes key architectural concepts of AgroWebPR along with a system overview. Chapter 4 provides details on the technical aspects of the notional application

architecture and its components. Chapter 5 proses thoughts for future enhancements. Finally, chapter 6 concludes this project report with a detailed listing of achieved results.

2. Background

2.1 Technologies

- *Model-View-Controller Architecture*

  The best way to build an application that is web based is to use a Model-View-Controller (MVC) pattern. MVC architecture is suitable at all scales, whether it relates to small, medium or large enterprise-level applications. A multi-tier application consists of logically separated layers with each layer responsible for accomplishing a separate logical process. Each tier in a multi-tier application can be deployed in a separate machine and is able to communicate with other respective tiers in the application. The greatest advantage to separating the application into discrete components is that you end up with software that developers can update, tweak, and manipulate without requiring major rewrites and aggressive re-factoring.

- *PHP*

  It is a language used to write instructions for a web server, much like HTML, CSS and JavaScript, which also makes instructions for the end user's web browser. It has evolved to include a command line interface capability and can be used in standalone graphical applications [5]. PHP generally runs in web servers by inputting its code and creating web pages using HTML as output. It can also output images, PDF files, and even Flash movies. It also can be deployed on most web servers and on almost every operating system, and has the ability to be used with many relational database management systems.

- *JavaScript & jQuery*

  It is "the" client-side scripting language for web sites and web applications. It is characterized as object oriented, dynamic and weakly-typed scripting language and is ubiquitously used in the web pages for form validation, functionality extension, rendering, and much more. JavaScript executes in the client web browser providing dynamic web pages and improved user interfaces. The usage of JavaScript is not limited to web pages; several extensions to JavaScript are available which provide additional functionalities and support.

  JQuery [1] is a fast and compact JavaScript library, which provides a simplified way to traverse HTML documents, handle events, animate and perform Ajax interactions. It is currently the most popular JavaScript library in use which also enables creation of plugins over the base JavaScript library. Visual Studio, a premiere integrated development environment for development in .NET, bundles jQuery along with it.

- *Google Maps*

  It is a free (non-commercial) web mapping service application, which provides many map-based services and is also available in mobile version. The maps can be viewed in different forms along with traffic, photos, videos and other information. Besides being available online it can also be embedded in web pages and has various web services available to consume geocoding, elevation, directions, and information about places from client applications. The embedding can be done using JavaScript API provided also by Google. The maps can be manipulated in various ways using different utilities in addition to the capability of

adding a variety of contents to the map. The APIs provide great deal of flexibility in the way contents can be added and manipulated on top of the many default features. The JavaScript Maps API V3 is the latest version [3]. The main disadvantage of Google Maps is its inability to depict the Polar Regions. Another product from Google, Google Earth, circumvents this limit with additional globe-viewing features. The points on the earth's surface can be identified using latitude and longitude data and in Google Maps a projection is made using latitude and longitude data converting them to a (x, y) coordinate, which is used to represent a pixel in the map [2, 3]. Figure 2.1 displays a snapshot of the Google Maps user interface.
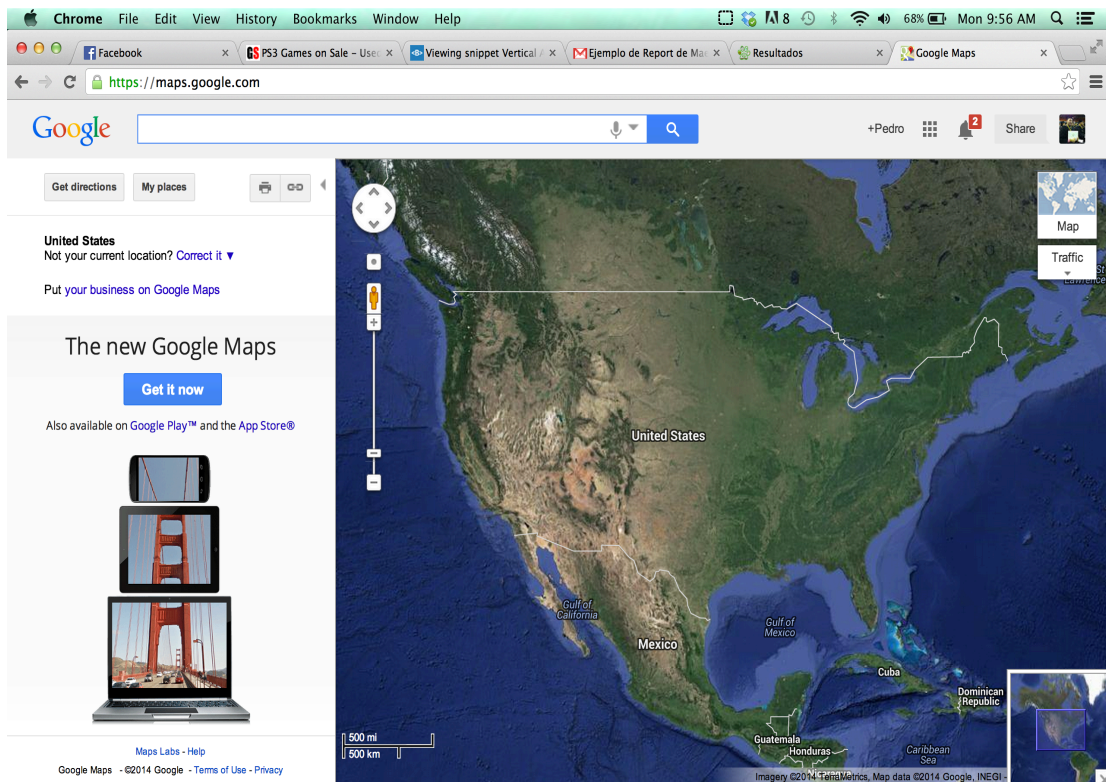

Figure 2.1: Google Maps Interface

10

- *MySQL*

  This is one of the most popular Open Source SQL database management systems, developed, distributed, and supported by Oracle Corporation [5]. Since MySQL software is Open Source anybody can download it at no charge. Furthermore, it is a client/server system, which consists of a multi-threaded SQL and can support several different client programs and libraries. It also has a wide range of application programming interfaces (APIs) and administrative tools. MySQL is used as a database engine supporting thousands of users and hence is highly scalable, robust, and requires little or no attention. All the tools that come with it make administration and SQL programming tasks much easier.

- *Google Charts*

  It's a Google API that is easily embedded into a developer's application by the use of JavaScript. Developers can load the libraries, list the data to be charted, select options to customize charts, and create a chart object with an id. Then by simply creating a <div> with the id that was created developers can display Google Charts. Google Charts provides many charts ranging from Line Charts, Bar Charts, Pie Charts, etc. Thus, given developers a wide range of chart options that they can manipulate by using their own data. To view users data Google Charts utilizes the DataTable class, making it easy to switch between chart types. The DataTable provides methods for sorting, modifying, and filtering

data, and can be populated directly from your web page, a database, or any data provider supporting the Chart Tools Datasource protocol [11].

2.2 Related Work (Literature Review)

Pick-a-Pepper.com [6] is a website that connects local farmers, gardeners, and food artisans with local consumers, restaurants, and institutions. It is a commercial product created by Emma O'Connell in September 2011. Pick-a-Pepper lets farmers who have a bounty to share, no matter the size of the connection to all different types of consumers throughout the web. Pick-a-Peper.com also does not charge its members any fees as the slogan states that the producer should keep every penny of their hard earned dollars. The website is divided into to two parts: Sell and Buy. Sell lets farmers register free of charge and lets them add products they have for sale. For Buy non-member can browse by farms or products, but their search engine has a few issues. As a non-member you are required to enter your zip code to search for locally produced goods in your area. A drawback of this approach is that if there is no farm located in your area you will get error messages. Pick-a-Pepper also features a paying option through paypal.com; thus, farmers need to have a corporate PayPal account to be able to receive payments for their products.

LocalHarvest.com [7] is a website that lets their members connect to small farms, farmer markets, and other local food sources. It is a commercial product created by Guillermo Payet in 1998 for the Buy Local movement. LocalHarvest lets users search from a wide variety of products such as pet needs, livestock, flowers, gift

baskets, etc. Their focus is on every aspect of agricultural activities and benefits businesses that focuses on a wide arrange of products. LocalHarverst features tabs where member can choose from shops, where they can see all the products that are available or farms, where you can search for a farm in your area by searching via zip/city or state. When searching for a product you can see an extensive description on how that product came to existence. Also, when a member decides on a certain product they will be able to see all the farms that have it available, and a tag if the farmer specified local pickup only. LocalHarvest also features the option to pay via credit card as well as via PayPal. They also support a map feature that presents its members with plot points of all farms, farmer markets, restaurants, and others across the United States. LocalHost even has tabs for forums, blogs and newsletters to have their members connected with each other.

FarmFresh.com [8] is a website, which provides the information of around 75 local farm members, and includes all the products available. FarmFresh was created by Fraser Valley Farm Direct Marketing Association (FVFDMA), formed in 1994, and promotes the market of farmers selling direct to an end point consumers. As soon as you access FarmFresh you are greeted with the option of choosing for three different indexes; namely, Farm, Product and Feature. They all produce the same result: a list of the index you chose with name, city, website, and summary. When accessing farmer's details you can see all the basic information as in the search with a Google Map with the location of the farm. The product index presents users with all the products that are available. Once you choose a product it will show users the farms that grow it. FarmFresh does not feature a member logging or a

PayPal account, which means that no established farm can add their information to the page without contacting the website administrator.

Even though these websites have contributed great value to the agricultural sector, they still lack functionality when compared to AgroWebPR. The mentioned websites only cater for the search of farms that are located in the United States, while AgroWebPR will cater to the needs of the Puerto Rican farms, processing lines, distributors and points of sale. Also, some features that are in the other websites are out of the scope of this project report focused on AgroWebPR; for instancing the PayPal feature.

3. AgroWebPR System Overview

This chapter defines the design of the application and the implementation choices and details.

3.1 Product Overview

The core functionality of AgroWebPR is to provide users a fast and reliable way to sell their agricultural products online. This application is built to cater the need of farmers, manufacturers, distributor and point of sale, and provides a secured distribution channel for them. The application will store the users information as well as the products farmers will have available during a given period of time. The application will also have support for users to restock and edit their products at any given time. In addition, the application will have the ability to search via products, user companies, town, and product type (e.g. vegetables). Furthermore, the application must provide users with a means of communicating though the application itself; i.e. users can negotiate asking prices for their products.

3.2 Major Tools and Components

- **JavaScript and jQuery** - This scripting language is used to provide dynamic outlook to the web pages along with validation and Google Maps integration and manipulation. Google Maps JavaScript API v3 is used which is the latest API library for this mapping application. JQuery provides several animation and UI components. The jQuery controls require JavaScript to function and assume that the JavaScript is enabled in the browser.

- **Google Maps** - This popular web mapping application provides the infrastructure upon which the user can define the search region and the search results are also shown therein. Figure 3.1 samples how AgoWebPR implements the Google Maps feature.

Figure 3-1 Google Maps implementation sample.

- **Apache Web Server** – an Apache Foundation open source web server.

- **MySQL DBMS** – an open source SQL database management system.

- **Linux Operating System** – an UNIX variant open source operating system.

- **PHP** – a server-side scripting language designed for web development applications.

- **Google Charts** – a software package written in JavaScript to display high quality graphical charts using users own data.

3.3   Design Overview

This section gives an overview of the architecture chosen. It also presents the design description of the system interface.

3.3.1 System Architectural Design

The entire application is based upon a three-tier computing model architecture in which the system functionality and associated workloads must be allocated and segregated as a multi-tier architecture between three separated processes. All in all, there are 3 tiers in this designed model. This model of development helps in the creation of a flexible and reusable application infrastructure. Each layer is developed as a separate project to each other and can be deployed independently in different hosting machines. Later, in order to change the implemented technology or for enhancing the application, the specific layer changed needs to be added. These processes are identified as: the Presentation Tier (PT), the Application Logic Tier (ALT) and the Data Management Tier (DMT). A depiction of the Three-tier architecture is presented in Figure 3.1.

The **PT** is the topmost level of the application and focuses on displaying information. It communicates with other tiers via the Graphical User Interface (GUI). The second tier (i.e. the **ALT**) is pulled out from the presentation tier and, as its own layer; it controls an application's functionality. The third tier, the **DMT**, focuses on the database servers. Here the information is stored and retrieved to be used in the ATL tier to later be displayed to the user.

Figure 3-2 Overview of the architecture for AgroWebPR

AgroWebPR implements these processes using the concepts of the Model-View-Controller (MVC) design patter. [10] MVC is a clear division between domain objects (ATL) that model our perception of the real world, and presentation objects (PT) that are the GUI elements we see on the screen. This insures that processes are

18

independent from each other, which benefits the testing of each role separately. Figure 3.3 samples a Model-View-Controller.

- **View-** Role implemented in the Presentation Tier. The user is built on HTML5, JavaScript, and JQuery. It provides users with the graphical interface in which the user will be able to view all the aspects of AgroWebPR.

- **Controller-** Role implemented in the ALT. The controller is built on PHP 5, which collaborate with the Apache Web Server.

- **Model-** Role implemented in the Data Management Tier. The model keeps the user information and the activities that they will be implementing. User product information will also be kept such as price, quantity, the expected harvest date, and the actual harvest date. All data are received from the user via web forms and placed in a MySQL database.
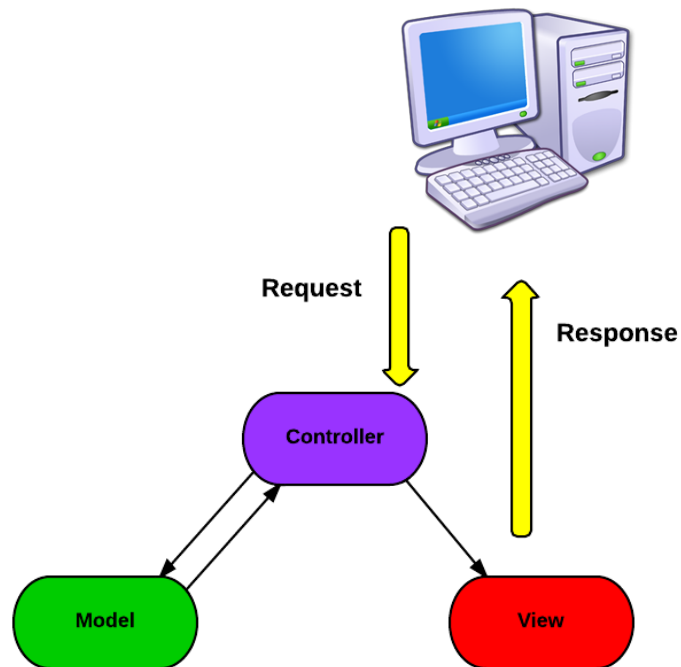
Figure 3-3 Overview of the MVC for AgroWebPR

3.4 Presentation Tier Overview

This is the topmost tier in the architecture stack, which provides the user interface for the users to interact upon. The system is a basic web client that will run on all major web browsers. Furthermore, all the queries for search, results of searches, and all other information are displayed for the user in this layer using HTML5 web pages. The default-landing page for any user should be an introductory page and users should log-in to the system to further use certain features of the application. Other features such

as searching will be accessible to everyone. Users that have created a profile in the system will have the ability to sell/buy products, as well being able to add other users to favorites, contacting them though the system and viewing reports. Primarily, users will need to create an account to be able to see everything AgrowebPR has to offer, but some features will be available for non-users. Non-users of the application can have access to the AgroWebPR search engine.

The application will have searching functionality where users can define certain parameters. When a user enters the desired parameter whether it is products, product type, town, or company name, the search result will contain values satisfying this. Furthermore, all searches are categorized based on the name of a given product. Each product will display the user with its specific information, which includes price, location, company, and quantity. Finally, all products will utilize Google Maps with a marker indicating the town where the product is located. The specific information for each marker has the town name and zip code of that given product.

Currently, AgroWebPR only gives non-users the access to search but for those users who register in the page, they will be given full access to the application. Primarily, users will need to access their own profile page that features a side menu with all the features a registered user; this includes: Personal Messages, Offers, Add Production, Reports, Edit Profile and Contacts. Each one will be discussed in this section. Figure 3.4 presents the profile page.

Another feature that the users profile contains is the ability to edit or delete products that were previously added to the system. This means that if a product had all its quantity purchased, the user could click on edit and restock that product with new quantities. If a

user needs to correct the date that a product will be available, such information is easily updated. The delete button does not require much explanation, since it is used to delete a product the user does not intend to sell. The profile also displays a favorites table with the name of companies the user had prior contact with. Also, the favorites table features a delete button that gives the user the option to delete a company from the favorite list.
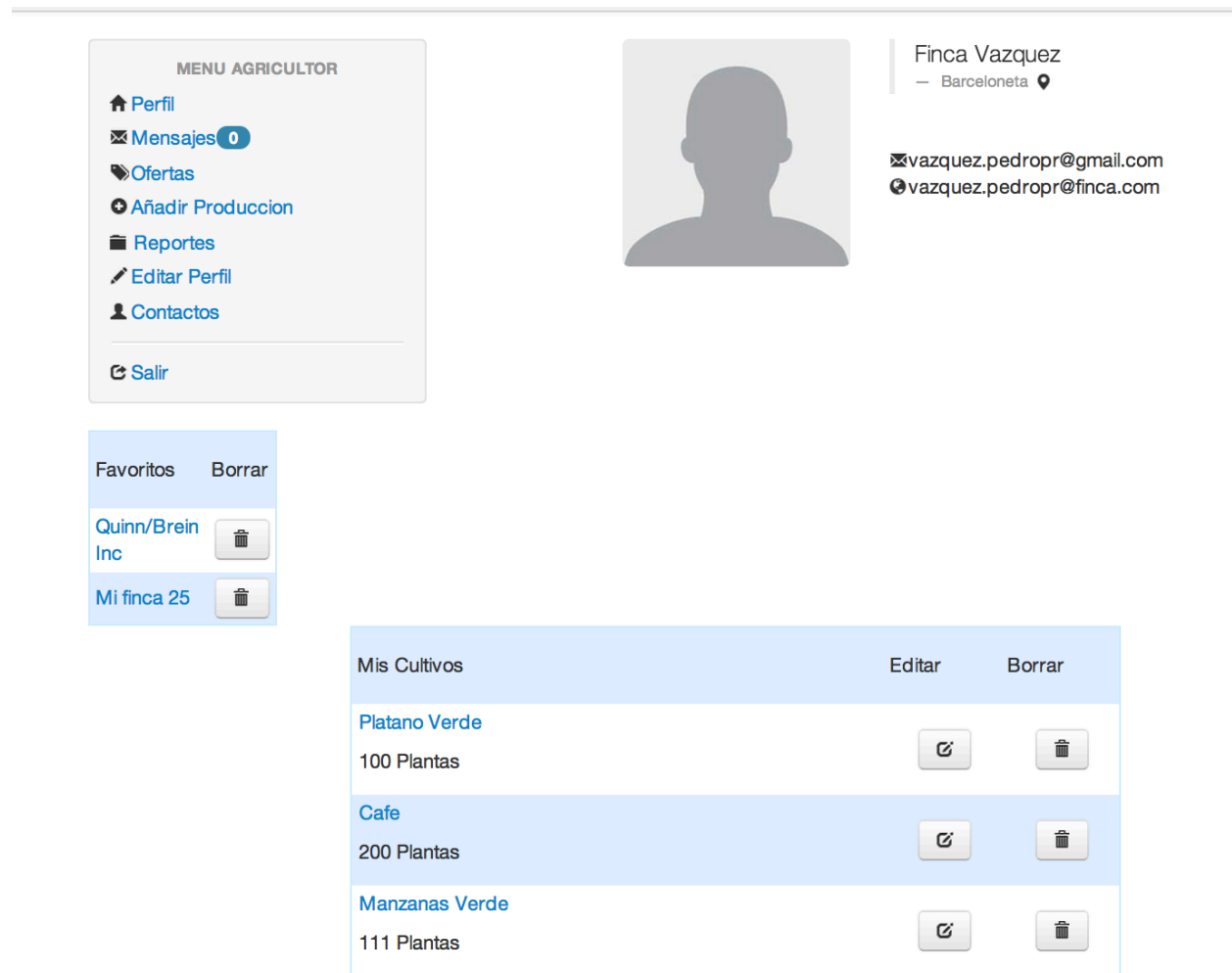

Figure 3-4 AgroWebPR Profile Page

Other features registered users will have access to include: (1) adding new products, (2) defining expected harvest date, (3) defining the actual harvest date, (4) estimated quantity available, (5) suggested selling price, (6) product type, and (7) per-unit price. Since selling products is one of the most important functions of this application, it was important to give users an easy and clear way for adding their products. The first thing that user will see is a dropdown box that will show an exhaustive list of products that are currently grown in Puerto Rico, which is aligned with AgroWebPR's focus in Puerto Rico's agriculture.

In addition, to provide an estimate on product supply, users will be required to submit the estimated harvest date. The success in predicting the harvest date will be captured by capturing from the user the actual harvest date. These two pieces of information will be captured through the use of two calendars.

Reports are another important feature available, which help users keep track of the business activity; i.e., how many products have been sold or purchased. By utilizing Google Charts users will display bar chart showing products and quantities or prices. Average price information for any given product will help suppliers in suggesting selling prices for upcoming product harvests. Figure 3.5 shows an AgroWebPR report example.
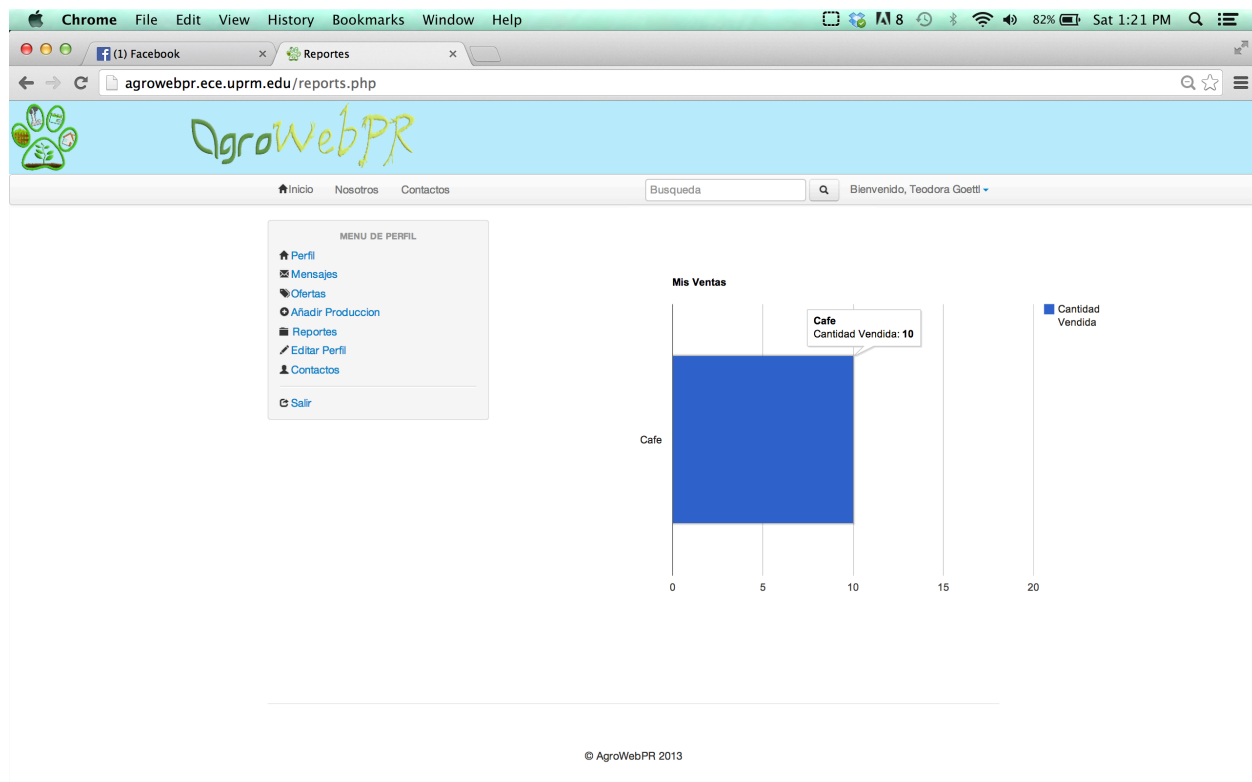
Figure 3-5 AgroWebPR Reports Page

The last major function is product purchasing; i.e. making offers for upcoming harvests or finished value-added products. The buyer initiates this process, by going to the products page and clicking on offer. The user can make an offer by inputting the desired quantity and the date when the product is needed. The buyer will receive confirmation of the offer via the profile page. The supplier will later need to evaluate the offer and accept or refuse it, providing feedback on the decision. An offer page will be available to see all offers that the buyer has made, as well as all supplier-selling commitments. Figure 3.6 presents the offer page.

Figure 3-6 AgroWebPR Offer Page

3.5 Application Logic Tier Overview

The Application Logic Tier (ATL) of AgroWebPR contains the core application processing logic that is integral to the functionality of all the layers. This layer serves as an interface between the data management tier and the presentation tier application providing an abstract representation of the data and its access logic. This is deployed as a separate assembly, which sends requests to the data access layer via the end points; similarly it interacts with the presentation layer using the Apache Server. These follow Http bindings to transfer serializable objects over Http.

25

The main aspect of the application tier is its ability to display the user interface via using the client code (scripts) and web documents. The web host that will be used by the presentation tier will depend on the web documents to show all the components to the AgroWebPR GUI. Many parts of these web documents are written in JavaScript and JQuery, which in part are both client-scripting languages. The remaining web documents consist of limitedly static and dynamically generated web forms using server-side scripts. These server-side scripts are written in the PHP language, which also are embedded to run directly in a web document.

All the content to be used by the ATL is made available to the presentation tier, which is published using an Apache Web Server as mention before. Thus, after a user inputs a request, the Apache Web Server has to interpret the PHP code before responding. This process occurs repeatedly, given that all HTML web documents have the ability to generate all content sent to the presentation tier.

3.6 Data Management Tier Overview

This layer abstracts the storage layer and provides the requesting service data from the database or the file system. This layer defines interface for interacting with the SQL server to read the configuration settings and data, and an interface for interacting with the file system containing the AgroWebPR information. This tier also hosts the service provided by the Database Management System (DBMS) that will be used by AgroWebPR [10]. The DBMS maintains and uses the data stored in the AgroWebPR database. The Data Management Tier uses the DBMS to allow the Application Logic Tier to make use of the data.   MySQL is the most used open-source DBMS available in the public domain and AgroWebPR uses MySQL DBMS to store all of its data.

Defining database table structures that would put all the data and its information looks perfect for this type of DBMS. The data managed by AgroWebPR includes only data submitted information; data types used are standard data types like real numbers, dates, and character strings.

4. AgroWebPR System Overview

4.1 Presentation Tier IMPLEMENTATION

This Chapter goes into more detail on how AgroWebPR was implemented. Figure 4.1 depicts the menu component diagram.
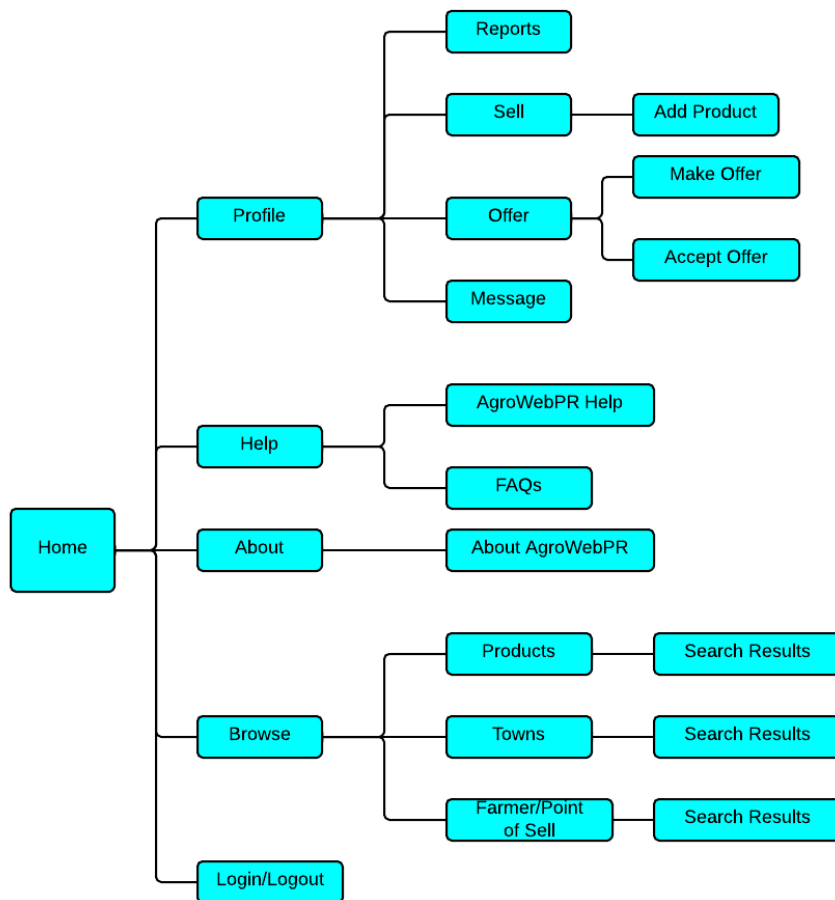

Figure 4-1 AgroWebPR Menu Component Diagram.

4.1.1 System Interface Description

All pages in the application use the same format. All pages are contained in a Master Page allowing creation of a consistent layout throughout the application. The master page has a header section, a body section, and a footer section. The header section contains the title, login bar, menu bar, and a search bar. The body section holds the contents that are defined in different HTML5 pages. Besides using HTML5 combined in the web document, JQuery will also be used to further enhance the user experience in the page. Furthermore, PHP will be included in portions of the web document to generate templates dynamically.

The header and footer are consistent and the only region where the content changes is in the body section or page specific content area. This region is used to display content available to a specific page; e.g. the search page. This area is also used to display a generic message in case of any exceptions occurring within the application.

Page Header: The Page Header is displayed at the top of all pages. It contains the title section, login controls, menu control and the side menu control. The side menu contains link to all the features in this web application. As mentioned in the requirements chapter, the visibility of the menu items depends upon the authorization roles assigned to the logged-in user.

Page Footer: The footer section contains the external links to UPRM site, Department of Agriculture site, and the sitemap along with the copyright declaration of the application.
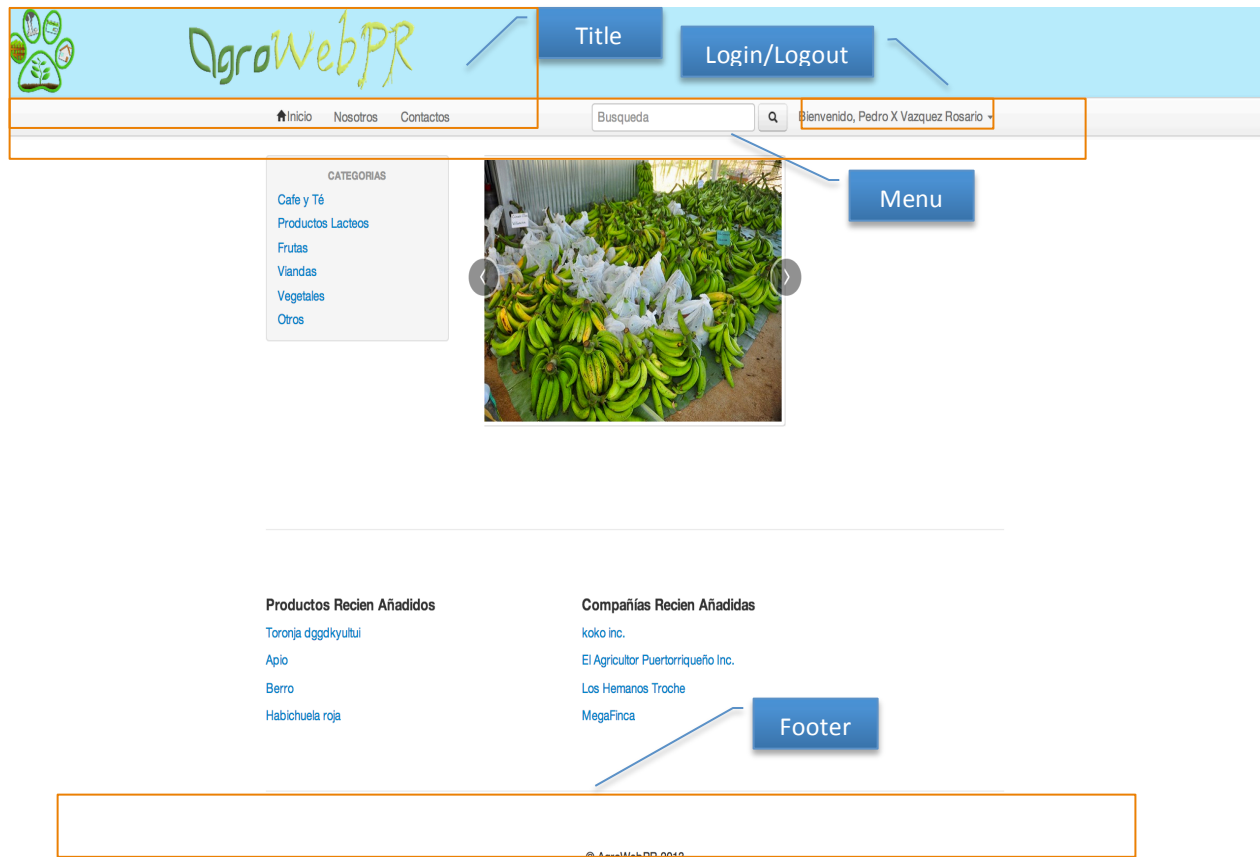
Figure 4-2 Rendered View of Master Page sans content.

4.1.2 Design consistency

The look and feel of the controls being used will be consistent throughout the application. For example, the color and font chosen for the text and button will be consistent and may only vary upon context difference. Likewise the display for notification and validation messages will also be consistent. Users will always be hinted upon wrong input and be properly notified thereof.

Error details will be hidden from the users and only friendly generic message notifying of a failure or cause of failure will be shown. Users will be able to return to a previous page after occurrence of an error.

4.1.3 Accessing AgroWebPR

Users should be required to login to the system to use certain features of the application. Since users are assigned different roles, which depend on their occupation, the contents and region of the application accessible to the users will not vary. These roles are as follows: Farmer, Manufacturer, and Point-of-Sale.   All users have administrator level access, which provides user with site wide access whether it is selling or buying products, communicating with other users, adding users to favorites, and verifying personal reports. Users who wish to make use of these features will be required to register though AgroWebPR registration page.

There is a system wide menu providing access to different modules of the application; i.e. if the user has registered in the page. The display of the main menu will change as a result, by displaying the user with a side menu. This side menu will be displayed and will be present throughout all major registered user displays. Users who have not registered will have basic level of searching though the page. Moreover, if an unregistered user tries to purchase a product, a login screen will be displayed, hinting that the user does not have the required credentials to access this area of the website. Users should not be able to jump to unauthorized pages by directly accessing a valid URL in the application. Direct access URL in this context refers to specifying the URL in the browser's address bar.

4.1.4 Searching through AgroWebPR

Besides searching via products, it is important for users to have other search options, such as (1) multiple company names versus (2) specific company name. Other search criteria include product type, which was implemented so that users could have a wider scope of results. As an example, if a user inputs Vegetables in the search bar, the results will be all products that are vegetables. Lastly, users could also decide to search via towns in Puerto Rico. This option is important for those users that wish to purchase products that are relatively close to where they live and discarding products that are being harvested too far away.

After users input the desired parameter they wish to search for, a search result page is accessed. This page features a list of the products based on the search criteria. Furthermore, all search results are displayed in tabular format with a picture (if the users included when adding the product) and the price, quantity, and location of that product. Also, included in the search result is a Google map with the location of that product. Figure 4.3 depicts the code for adding Google Maps feature.

```
1  <div class="span7">
2          <iframe width="500" height="250" frameborder="0"
   scrolling="no" marginheight="0" marginwidth="0"
3                      src="https://www.google.com/maps?
   f=q&amp;source=s_q&amp;hl=en&amp;geocode=&amp;q=<?php echo
   "$myrow[town]";?>,+Puerto
   +Rico&amp;aq=0&amp;oq=barceloneta&amp;ie=UTF8&amp;hq=&amp;hnear=
   <?php echo "$myrow[town]";?>,+Puerto
   +Rico&amp;t=m&amp;z=14&amp;output=embed"></iframe>
4
5          </div>
```

Figure 4-3 Code snippet of Google Maps Feature.

4.1.5 Selling and Purchasing Products

The most important feature for users is the ability to sell and purchase products. For users to wish to sell their agricultural products they will need to register, afterwards they will have to edit their profile information so that other users can see information such as their phone number and location. After finishing this process they will have to access the add product form, in which users will be required to fill out all the information presented to them, except adding an image to the product, which is optional. All information that the users will be input is standard for an agricultural product. After completing this information, products will be added to your list of products where you can edit the quantity of production and the date that they were cultivated and harvested. Also, you will be able to delete the product if you wish to do so.

Those companies who wish to purchase products for others will also be required to register. For this project, AgroWebPR does not require a PayPal feature or type of Shopping Cart, however there will be an Offer feature. To make an offer, the buyer will need to access the product page and specify the amount and the date for which the product is needed. A request will be later sent to the supplier to determine if they are willing to provide the needed products, creating an electronic contract. In the event of the supplier rejecting the offer, they will be required to provide an explanation via a text box labeled comments. For the convenience of buyers, there will be an area called offer where they will be able to manage new offers as well as the offer history with that particular supplier. Figure 4.4 provides an example on how to make an offer.

Figure 4-4 How to make an Offer.

4.1.6 Reports Implementation

As mentioned in a prior section, AgroWebPR uses Google Charts to create custom user reports. User can view three distinct reports as mentioned in Chapter 3. By using certain data from the AgroWebPR, Google Charts is able to produce a variety of reports. For AgroWebPR uses bar charts to display data graphically. The way AgroWebPR uses Google Charts through JavaScript that is embedded into the web forms. By using the Data Table class that comes standard in the Google Visualization API, the table is automatically populated by using a Query on the Product Line Table. Figure 4-5 depicts the code for adding the Google Charts feature.

```
1  // Load the Visualization API and the piechart package.
2      google.load('visualization', '1', {'packages':
.  ['corechart']});
3
4      // Set a callback to run when the Google Visualization API
.  is loaded.
5      google.setOnLoadCallback(drawChart);
6
7      function drawChart() {
8          // Create our data table out of JSON data loaded from
.  server.
9          var data = new google.visualization.DataTable(<?=
.  $jsonTable?>);
10         var options={
11             title: 'Mis Ventas',
12             is3d: 'true',
13             width: 800,
14             height:> 600
15         };
16
17         // Instantiate and draw our chart, passing in some
.  options.
18         var chart = new
.  google.visualization.BarChart(document.getElementById('chart_div
.  '));
19         chart.draw(data,options);
```

Figure 4-5 Code snippet of Google Charts Feature.

4.2   Data Management Tier Implementation

The database was designed first and based on the database entity model. Relation

constraints were created in the database prior to the model creation. As mentioned in a

prior chapter, AgroWebPR uses a MySQL database. MySQL is the most used open-

sourced DBMS in the world.

4.2.1 DBMS Tables

Figure 4-6 shows the Entity Relationship Model (ERM) diagram for the AgroWebPR database. The core table will be (company) profile, which stores all users information. The table will act as the main table to connect to all other tables by using the primary key profile_id, which will be a unique integer for each, an every user. Since users will be selling products, it was necessary to create a table that would store that information, which was named product_line. Thus, a profile can be associated with many product_line's or production making it a one-to-many relationship. Likewise, the offer table and the Product Line table will be a one-to-many relationship since a distinct offer can only be given to one product at a time. On the other hand, the relationship between the Offer table and the Profile is a many-to-many and the primary key profile_id will be given to distinct roles in the table, which are the actual profile_id and the company_id. Lastly, the final two tables will follow a similar pattern to the prior table meaning that both will harbor a many-to-many relationship and the primary key profile_id will given to distinct roles in both tables.
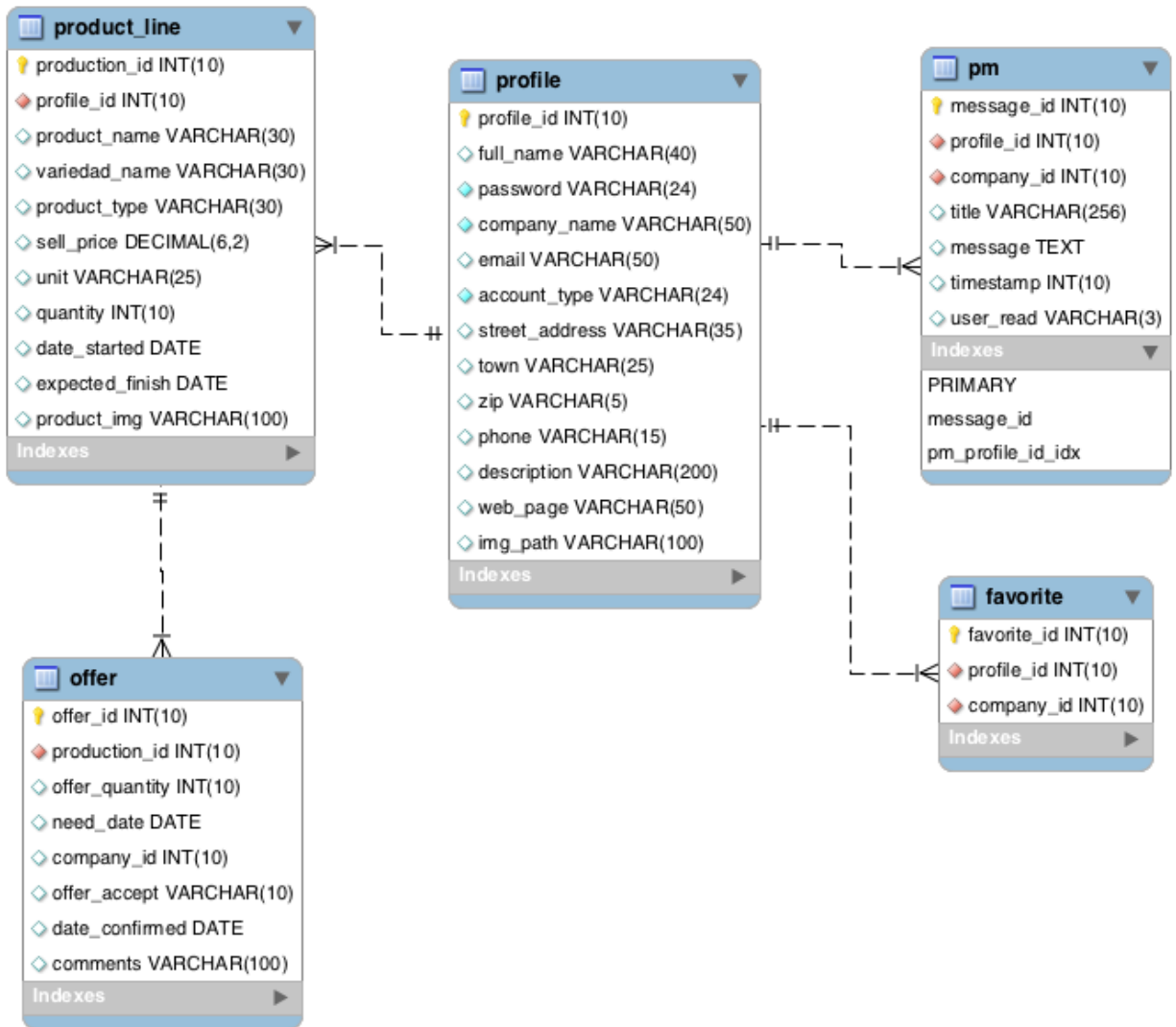
Figure 4-6 ERM diagram for user data.

5. Conclusions

5.1 Conclusions

For this project, three-tier architecture and the Model-View-Controller application was created for facilitating the communication between all entities of the supply chain. The open-source application was develop to provide farmers a means to sell their product straight to the point of sale or the value-adding processors across the Internet. AgroWebPR provides its users with more than just the ability to purchase products; it gives users a vast variety of uses such as reports using Google Charts, searching locations with Google maps, and messaging capabilities from user to user.

The web view was created using a combination of HTML and JQuery; thus, a GUI is provided for users to navigate with ease. Buyers have the ability to search through multiple products available at various farms, triggering an offer on them. Suppliers can provide visibility to available products, quantity, and time of sale using a simple web document that stores and retrieves the information in a simple database management system. MySQL was such DBMS used to store all information obtained via the users information and inputs.

The development of AgroWebPR has provided the opportunity for developing firsthand knowledge in the design and development of an enterprise application architecture that works together with various technologies that have been integrated. The handling of large remote sensing data files in the web application environment has provided a valuable insight towards the importance of optimization in file handling. MySQL has facilitated the handling of agricultural data. Furthermore, working with the

client side of PHP as well as JQuery helped acquire a better knowledge about the usage of this technology and its capabilities. Finally, several new features in HTML5 and CSS3 were learned during the development process.

6. Future works

6.1 Google Maps Implementation

The Google Maps implementation was not completed. A next step would use Google Maps as a way to determine which products are closest to you based on farm location. The result would be the exact distance in miles of the nearest supplier with the desired product that the buyer searched for. Thus, users can see what products are closest to their current location and the exact location. Transportation management software's are available to help define the best route to visit a predefined number of suppliers.

6.2 Mobile version

Mobile applications have taken over when compared to personal computers. Cellular phones should be the main receiver and sender of AgroWebPR communication activities, which should accelerate the response to offers and messages.

7. Bibliography

[1]   Bibeault, B., & Katz, Y. (2008). *Jquery in Action* (2nd ed.). Greenwich, CT, USA: Manning Publications Co.

[2]   Google   Inc.   (n.d.).   *Google   Maps*.   Retrieved   June   20,   2013,   from http://maps.google.com/

[3]   Google Inc. (n.d.). *Google Maps API Family*. Retrieved June 20, 2013, from http://code.google.com/apis/maps/index.html

[4]   MySQL   Inc.   (n.d.).   *What   is   MySQL?*   Retrieved   June   20,   2013,   from http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html

[5]   PHP Inc. (2009). *PHP: What can PHP do?* Retrieved June 20, 2013, from http://www.php.net/manual/en/intro-whatcando.php

[6]   Pick-A-Pepper, Inc. (2011). *Pick-A-Pepper* Retrieved September 3, 2013, from https://pick-a-pepper.com/index.php

[7]   LocalHarvest, Inc. (1999). *LocalHarvest* Retrieved September 4, 2013 from http://www.localharvest.org/

[8]   Fraser Valley Farm Direct Marketing Association (1997). *FarmFresh* Retrieved September 4, 2013 from http://www.bcfarmfresh.com/

[9]   José   E.   Maldonado   (2012)   *La   Seguridad   Alimentaria   y   Los   Alimentos Transgénicos   en   Puerto   Rico*   Retrieved   June   22,   2013,   from http://www.miprv.com/la-seguridad-alimentaria-y-los-alimentos-transgenicos-en-puerto-rico/

[10]   Ramakrishnan, R., & Gehrke, J. (2003). Database Management Systems. New York: McGraw-Hill.

[11]   Google Inc. (n.d.). Google Charts API. Retrieved January 15, 2014, from https://developers.google.com/chart/

[12]   Mike McGrath (2012). *SQL for web developers, programmers & students (3rd Edition).* Warwickshire, United Kingdom: In Easy Steps Limited.

[13]   Mike McGrath (2012). *PHP & MySQL (3rd Edition).* Warwickshire, United Kingdom: In Easy Steps Limited.

[14]   Mike McGrath (2012). *JavaScript create functions for the web (3rd Edition).* Warwickshire, United Kingdom: In Easy Steps Limited.

[15]   Mike McGrath (2012). *HTML5 take control of the web (3rd Edition).* Warwickshire, United Kingdom: In Easy Steps Limited.

[16]   Mike McGrath (2012). *CCS3 add style to your web pages (3rd Edition).* Warwickshire, United Kingdom: In Easy Steps Limited.