**Final Documentation**

**for**

**Price Comparison Application**

Capstone: Fall 2012

Lucy Horpedahl, Nathan Noonan, Daniel O’Doherty, and Nick Richardson

**Table of Contents**

**Table of Contents** 2

**1.** **Introduction** #

**1.1 Client** #

**1.2 Application** #

**1.2.1 Original Proposal** #

**1.2.2 Modified Proposal** #

**2.** **Functional and Nonfunctional Requirements** #

**2.1 Functional Requirements** #

**2.2 Nonfunctional Requirements** #

**3.** **Architecture and Design** #

**4.** **Implementation Issues** #

**5.** **Testing (list the tests conducted)** #

**6.** **Lessons Learned** #

**Index of Figures** #

**1.** **Introduction**

**1.1** **Client**

The client for the application we chose to design for this project is Justin Richardson, the owner of a small business, Digital Attitude. The business, which is located in Crossroads Mall, sells Japanese animation collectibles, movies, books, video games, and more.

We spent time at Mr. Richardson’s business, learning about how the store runs. Most, if not all, of his products have barcodes which are scanned in order to read the UPC number of the item. The items are entered into a database and organized by this number so that searching for this number will return the proper item. These barcodes are central to both our original application idea and our updated application.

**1.2** **Application**

**1.2.1** **Original Proposal**

Our initial plan was to design a new point-of-sale system for Mr. Richardson’s store. A point-of-sale system handles nearly all of the necessary operations of a retail business. It allows inventory to be searched and updated, purchases to be made, and customer information to be stored.

We believed that Mr. Richardson did not have a proper point-of-sale system, and we planned to create one that would fit his business’s unique needs. For example, Mr. Richardson buys used items from customers and then sells them in his store. He selects the pricing for these transactions by searching Amazon.com, which is one of the few places that sells most of the items that Mr. Richardson sells in his store.

We planned to implement, as part of the point-of-sale system, a “price checker” that would search Amazon.com and display the current price for the items so that Mr. Richardson would not have to do these searches manually. Upon meeting with Mr. Richardson further, we discovered that his point-of-sale system was adequate, but he told us that the “price checker” feature would be very helpful.

**1.2.2 Modified Proposal**

Because the point-of-sale system would have been a very large project, and the system currently used by the client was sufficient for his needs, we decided to narrow the scope of our project to just design an application that would search Amazon.com and retrieve prices for items in Mr. Richardson’s inventory. We planned to include options for filtering and sorting the output as well as for making changes to the prices in the database through the application itself, rather than having to go back to the point-of-sale system to modify the prices.

The application would need to communicate with Mr. Richardson’s existing database, process the data to retrieve the UPC numbers, communicate with Amazon.com to search for the UPC numbers and retrieve the prices, process the prices to display them on an interface, and ultimately communicate with the database again to write new prices back to the database.

Ultimately, we ran across a number of difficulties in designing this application which caused us to modify certain features, but the basic architecture would remain the same. The rest of this document describes in more detail the work that was done to design this application.

**2.** **Functional and Nonfunctional Requirements**

[*8-12 pages; this would be a much shortened counterpart to what you did in CSCI 4830; you can follow the outline for the requirements specification document in CSCI 4830; be sure to include use cases for the main services you will support*]

**2.1 Functional Requirements**

**2.2 Nonfunctional Requirements**

**3.** **Architecture and Design**

[*8-12 pages; this is also a much shortened counterpart to what you did in CSCI 4830; you can follow the outline for the design document in CSCI 4830; the following should be described: architecture, modules and their interfaces, database (if applicable) - we may have to iterate to get the desired details; some screenshots*]

**3.1 High-Level Architecture**

Figure 1: Architecture

Point-of-sale system

Inventory Database

Price Comparison Application

Amazon.com/Google shops

*Figure 1* shows how the price comparison application interacts with the existing modules. The application must communicate with the existing database in order to read and modify records in the inventory. The database also communicates with the point-of-sale system, and the changes made by the price comparison application must be reflected in the point-of-sale system.

The price comparison application also must communicate with Amazon.com or Google shops in order to search for specific barcodes and retrieve the price at which the item is being sold by various merchants.

First, the application will retrieve all used items from the client’s database. When it has a list of barcodes, the application will send requests to Amazon.com or Google shops for each item. Each request will return a list of offerings for the item, which will tell how much the item is being sold for at various locations online. After retrieving this list, the application will perform statistical calculations to find an average price for each item. These averages, as well as the current price in the client’s database, will be displayed on an interface that will allow the client’s database to be modified if the user chooses.

**4.** **Implementation Issues**

[*1-2 pages; various issues that cropped up in the process of writing the code*]

**5.** **Testing**

[*2-3 pages; list the tests conducted*]

**6.** **Lessons Learned**

[*3-5 pages; what you learned in this project; for example: how had the project help improve your understanding of CS principles? how had your CS background help with this project?; if you had the chance to do it over, what would you do differently; based on your experience, suggest some future extensions*]

**X.** **Section**

**X.1** **Heading**

**X.1.1** **Sub-heading**

**X.1.1.1** **Point**

**Index of Figures**

fig. 1: Architecture #