Price Comparison Application Software Plan

Vinayak Panchal
Vinayak_Panchal@student.uml.edu
University of Massachusetts, Lowell

Nikhil Swami
Nikhil_Swami@student.uml.edu
University of Massachusetts, Lowell

1. Introduction

The Internet and networking have increased the availability of information immeasurably. Since the information is so vast it's harder for users to retrieve exact information. Our Price Search engines have been developed to facilitate in one such price comparison information retrieval.

I. Project overview

This project is to create a prototype Price Comparison Software. Our Price comparison software will provide shopping tools and search products from various websites. It will simply provide a simple searching tool which will compare and display the various prices from websites. Our shopping engine will collect product information, including pricing, from participating retailers and then display that collective information on a single results page in response to a shopper's search query. In this way, shoppers can compare each retailer's price, shipping options, and service on a single page and choose the merchant that offers the best overall value.

II. Project deliverables

1. Preliminary Project Plan	2016.02.29
2. Requirements Specification	2016.03.07
3. Analysis [Object model, Dynamic model, and User interface]	2016.03.07
4. Architecture Specification	2016.03.07
5. Component/Object Specification	2016.03.16
6. Source Code	2016.03.16 - 2016.04.16
7. Test Plan	2016.04.16 - 2016.04.23
8. Final Product w/ Demo	2016.04.24 - 2007.04.25

III. Evolution of this document

This document will be updated as the project progresses. Updates should be expected in the following sections:

i. ***References*** - updated as necessary

ii. ***Technical Process -*** this section will be revised appropriately as the requirements and design decisions become clearer

iii. ***Schedule –*** as the project progresses, the schedule will be updated accordingly

Revision History

Revision	Date	Updated By	Update Comments
0.1	2016.02.29	Nikhil & Vinayak	First Draft
0.2	2016.03.21	Nikhil	eBay API and data parsing and retrieval
0.3	2016.03.28	Vinayak	Amazon Dev key and product advertising API

0.4	2016.04.01	Nikhil	Combining API calls
0.5	2016.04.07	Vinayak	Display Results
0.6	2016.04.11	Nikhil	Design
0.7	2016.04.17	Nikhil & Vinayak	Design Implementation and formating

IV. References

2. Project organization

I. Project Scope

To deliver a working comparison search engine, which can lookup, search and visualize the data retrieved from amazon and eBay API for the user entry and give them the best possible choice from these sites.
The user will have his own login and profile which he can sign in on to find his previous queries and list which he can store. He can modify his profile and user settings.

II. Process model

The process used for this project will be a Hybrid Waterfall model such that each stage of the waterfall allows us to update the project plan and other deliverables for missing areas or correctness.

Organizational structure

Team Members –

i. Nikhil Swami
ii. Vinayak Panchal

Week/Deliverable	Team Leader	Deliverable Description
1	Nikhil & Vinayak	Project Plan
2	Nikhil & Vinayak	Requirements Specification
3	Nikhil & Vinayak	Analysis
4	Nikhil & Vinayak	Architecture Spec
5	Nikhil & Vinayak	Component/Object Specification
6	Nikhil & Vinayak	Source Code
7	Nikhil & Vinayak	Test Plan
8	Nikhil & Vinayak	Final Deliverable

III. Project responsibilities

.

Team member assignments per deliverable according to expertise

1. Project Plan – Entire Team
2. Requirements Specification – TBD
3. Analysis – TBD
4. Architecture Spec – TBD
5. Component/Object Specification – TBD

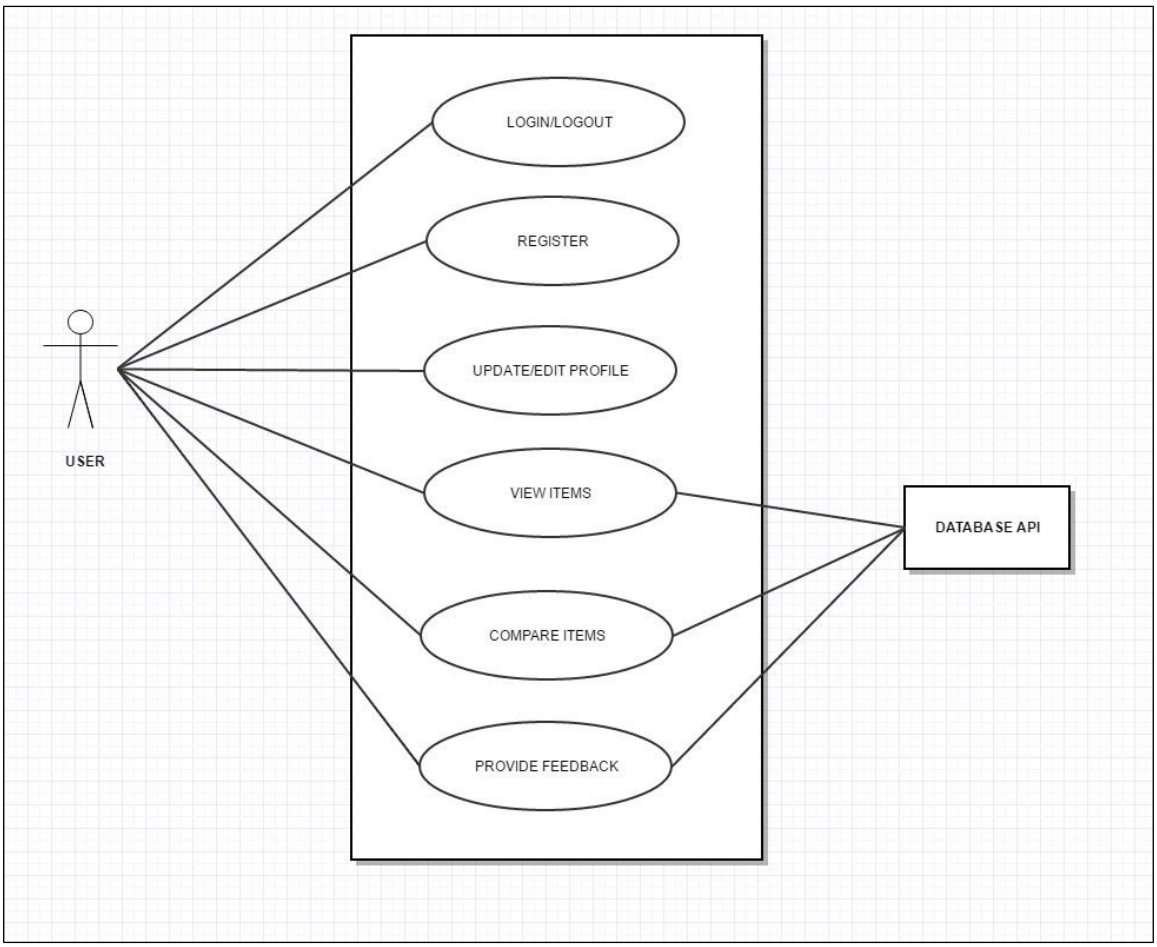
6. Source Code – TBD
7. Test Plan – TBD
8. Final Deliverable – Entire Team

IV. Excluded

Actual buying and trading of items will not be covered in this project. Transactions rights are not present and no credit card details or information is taken from user.

3. Software Development Process

3.1 Software Development Process Overview



USE CASE DIAGRAM

3.2 GENERAL PLANS FOR SOFTWARE DEVELOPMENT

The Price Comparison Project software development will apply the following general methods

a. The project will follow the defined processes documented in Section 2 to conduct software requirements analysis and manage the Software Requirements Specification (SRS).
b. The project will only design and develop software to meet requirements set at the beginning of the project. It will deliver a workable comparison search engine with the capability of comparing products from multiple sites
c. The project will adhere to the standards required by the SDP for design, coding, and test methods for new software

3.3 SYSTEM INTERFACES

3.3.1 User Interfaces

● The user interface for the system will allow the user to easily generated documents, search for documents, and modify documents. The user should be presented with all main functions on the first user interface page to allow for the user to select the function to use without the need to navigate inward to find it. The interface will need to use tab focus marks to allow for navigation using a keyboard as much as possible to alleviate stress on users' arms and hands caused by changing constantly from keyboard to mouse. It will be accessible through a web interface to allow for centralized hosting and use by various operating systems.

3.3.2 Software Interfaces

The software will need to interface with the API's and pull data from it and present it in a specific display view. The connection should be able to extract information from XML and generate required information.

4. SCHEDULES AND ACTIVITY NETWORK

5.1 Non-functional Requirements

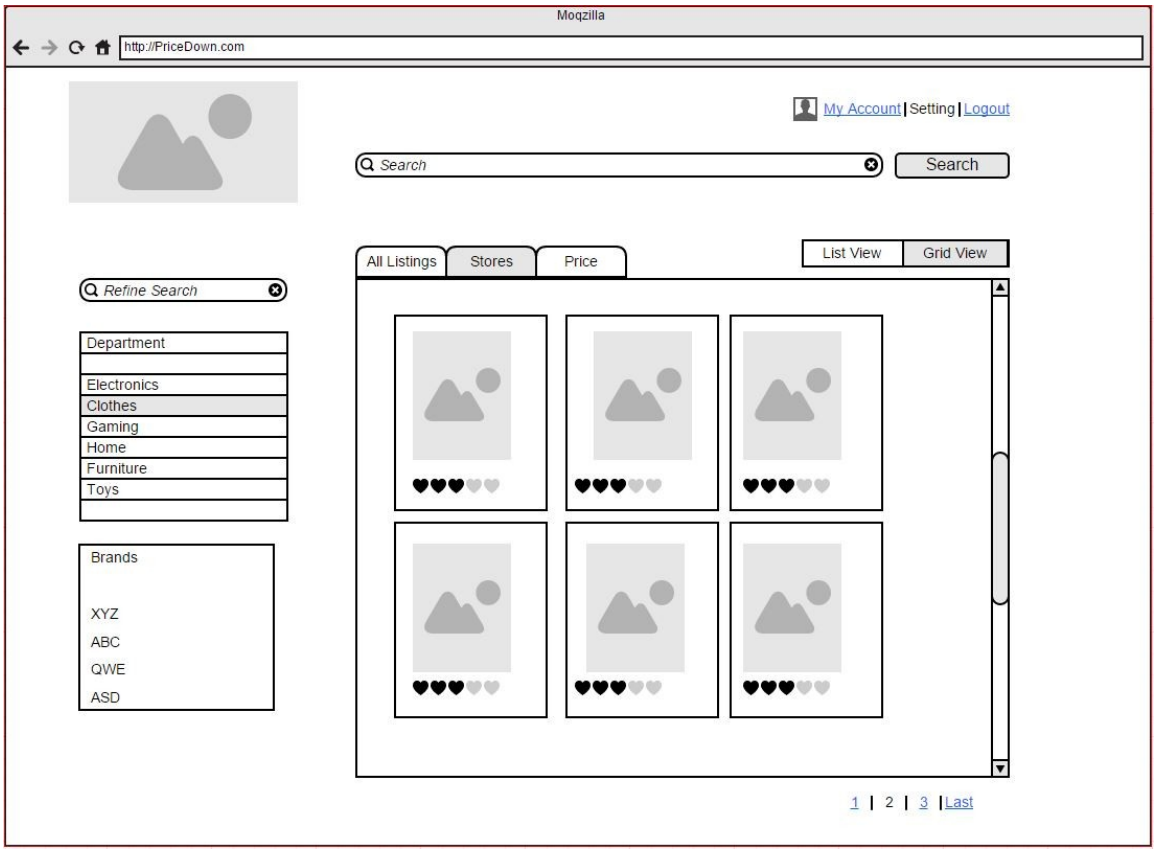
Performance Requirements

● The system should be able to generate previews of documents within 15 seconds of user request.

● The system should be able to be multi-tasking to allow multiple users, up to multiple simultaneous users per interface instance to interact with the system without having to wait on others to finish working with the system.

● The system should be able to hold and search through large amounts of documents. The data structures used for the system will be fairly simple consisting of a few fields to hold document types and their related codes, XML instances with an id, and an audit log table, however the size of the simple data structures could potentially be quite large.

5. DIAGRAMS

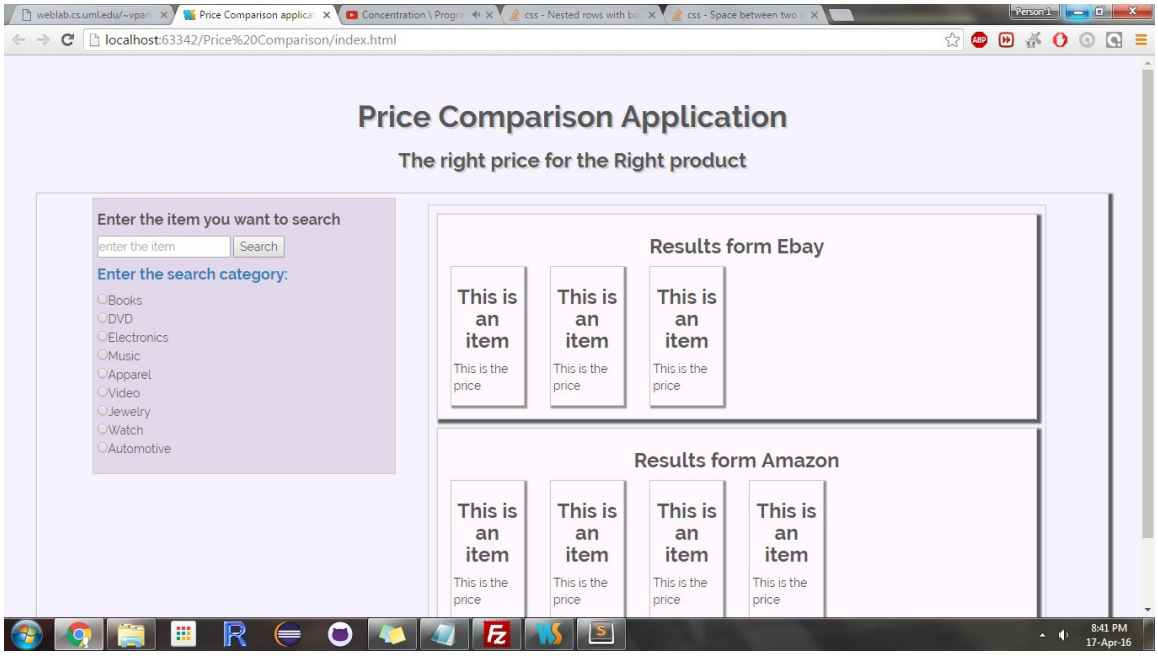

Moqzilla
http://PriceDown.com
My Account | Setting | Logout
Search
Search
All Listings
Stores
Price
List View
Grid View
Refine Search
Department
Electronics
Clothes
Gaming
Home
Furniture
Toys
Brands
XYZ
ABC
QWE
ASD
1 | 2 | 3 | Last

Code sample:

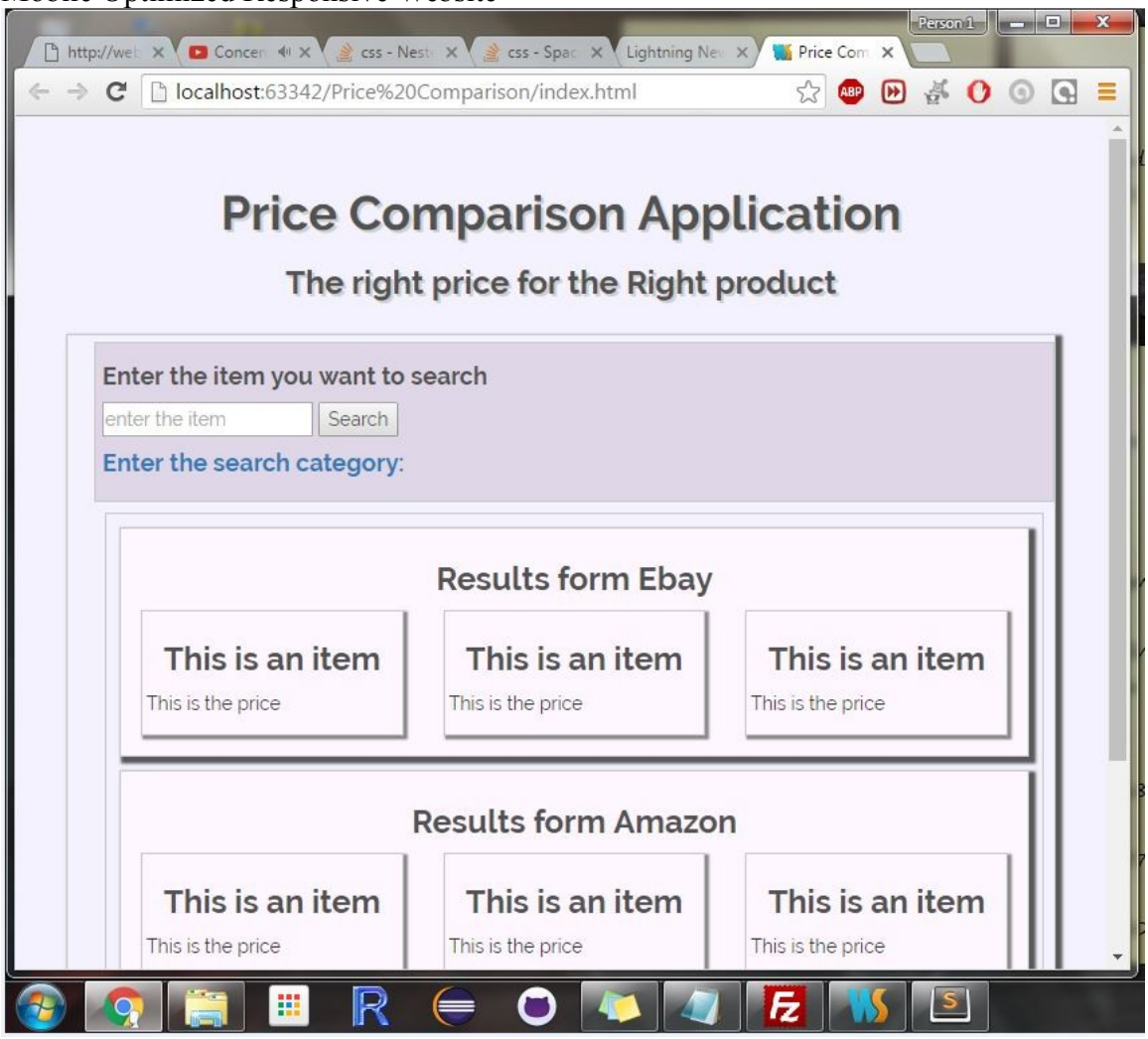
```php
<?php
error_reporting(E_ALL);  // Turn on all errors, warnings and notices for easier debugging

// API request variables
$endpoint = 'http://svcs.ebay.com/services/search/FindingService/v1';  // URL to call
$version = '1.0.0';  // API version supported by your application
$appid = 'student-products-PRD-038ccaf50-b2ee97ab';  // Replace with your own AppID
$globalid = 'EBAY-US';  // Global ID of the eBay site you want to search (e.g., EBAY-DE)
$query = 'harry potter';  // You may want to supply your own query
$safequery = urlencode($query);  // Make the query URL-friendly
$i = '0';  // Initialize the item filter index to 0


// Create a PHP array of the item filters you want to use in your request
$filterarray =
  array(
    array(
    'name' => 'MaxPrice',
    'value' => '25',
    'paramName' => 'Currency',
    'paramValue' => 'USD'),
    array(
    'name' => 'FreeShippingOnly',
    'value' => 'true',
    'paramName' => '',
    'paramValue' => ''),
    array(
    'name' => 'ListingType',
    'value' => array('AuctionWithBIN','FixedPrice','StoreInventory'),
    'paramName' => '',
    'paramValue' => ''),
  );

  // Generates an indexed URL snippet from the array of item filters
function buildURLArray ($filterarray) {
  global $urlfilter;
  global $i;
  // Iterate through each filter in the array
  foreach($filterarray as $itemfilter) {
    // Iterate through each key in the filter
```

PHP Hypertext Preprocessor file lengt

ScreenShots



Mobile Optimized Responsive Website



Responsive Website 2