Project Analysis Document

Group 3:

Ashraya Regmi, Jesse Cruse, Tyler Roland, Matthew Stevenson

University of Maryland University College

Revision 1.01

June 8 2017

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Editor** | **Description of Changes** |
| 1.00 | 06/08/2017 | Ashraya Regmi | Original Document |
| 1.01 | 06/09/2017 | Jesse Cruse | * Added risks 2 and 3 * Adjusted revision table * Added potential enhancement 6 * Grammatical/wording adjustments * Added to I/O data, processing steps |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Project Analysis:

1) Analysis

Our project is to create a web application that keeps track of what the customer has purchased in the past and depending on their shopping history, advise them on what they might during their shopping trip to the store.

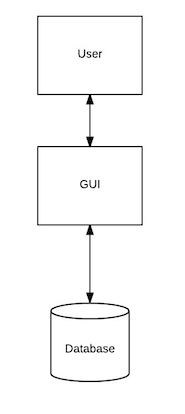
Its functional requirements are as follows:

|  |  |
| --- | --- |
| **Requirement #** | **Description** |
| 1 | This program will have a user interface to interact with the web application. |
| 2 | The program tracks and stores what the user has bought over time. |
| 3 | This program will allow the user to generate a shopping list for regular interval trips as well as special trips. |
| 4 | This program will require a backend database to store purchase and user details. |
| 5 | This program will enable the user to email lists to themselves from the website or print the list. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

To go more into detail in this analysis, we will break it down regarding the output data, input data and its processing function:

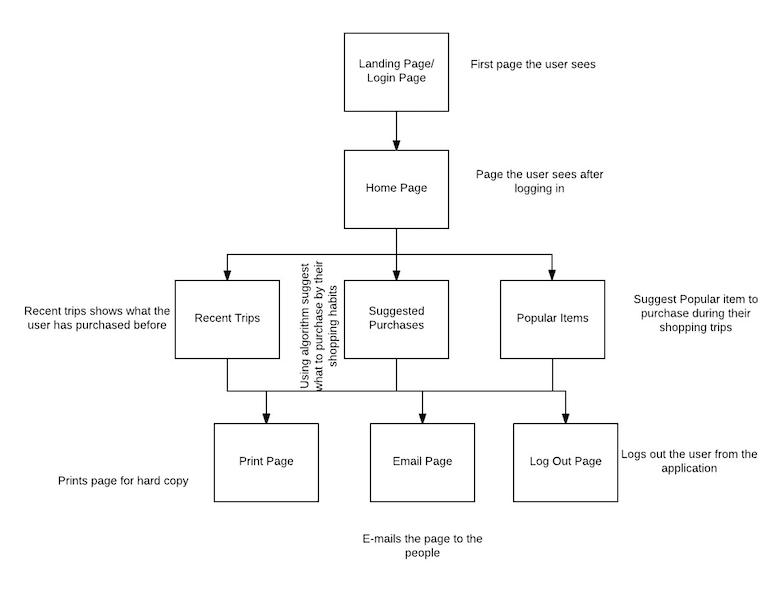
1. Outside system: the end user
2. Input data: Product names, product prices
3. Output data: Shopping list, list of products
4. Data Processing:
   1. Database stores the information of product purchases, including date of purchase.
   2. Algorithmically generate a shopping list based on stored details in the database.
   3. Program displays the generated list on users’ device.

The context diagram shows the above analysis in a diagram below in Figure 1:



**Figure 1 Context diagram**

Based on the above diagram, we can break down our system in to the following: User Input, Graphical User Interface, and the database that stores and processes the information. These systems are necessary to perform the data to convert the input data to the output data and vice versa.



**Figure 2: Subsystem diagram**

Once the user lands on the landing page, they will be provided with a log in page that the user enters the credentials to be authenticated by the database so they can be validated as the user and can proceed with using the website. After logging in, they have the option to see their recent trips and what they bought, suggested purchases based on what they have purchased in the past, and popular items they might be interested in. Also, are the print page, where the user can print the page and use it as a hard copy, email the page to another person, and log out from their account.

**Possible enhancements**:

Some possible enhancements to the application include:

1. Ability to have multiple user accounts
2. Ability to notify/email user what they might need on their next shopping trip
3. Ability to create a shopping list based on the diet of the user
4. Ability for the system to track nutritional information based on the user purchase history for food items
5. Ability to calculate expense per weekly or monthly bases for their shopping trip.
6. Capability to algorithmically reduce/eliminate duplicate items entries.

**Possible risks and risk mitigation:**

Risk 1: Password cracking

Mitigation: Password encryption, when user types the password it is not seen visible.

Risk 2: Duplicate product entries within the database, skewing algorithm results.

Mitigation: For the purposes of this project, the database will be preloaded with items that have been filtered of duplicate entries. Although potentially capable, the algorithm design would have to advance beyond the scope of this project to fully implement duplicate entry mitigation techniques.

Risk 3: Newly entered items with too few date entries to provide accurate purchase forecasting.

Mitigation: For newly entered products, the algorithm makes some assumptions until there is enough purchase history to take over the prediction scheme. The assumption made, for the purposes of this project, is a product purchase interval of five days. Therefore, items that fall under this assumption scheme will continue to populate on shopping lists - although the user will, of course, skip purchasing the item if it is not needed at the time – until there are enough (approximately five) dates in the database to begin predicting a purchase pattern.