**Criterion B: Design**

### Flowchart(s)

The following flowchart outlines the workflow of application when a user interacts with it.

Diagram

Description automatically generated

### Algorithms

### Since this is a web application for comparing retail prices and knowing when the best deals are on the market, I will be using a user based approach for this. Meaning, if users would like to set a price alert, they will be required to sign up for an account. My intent with this is so that I can send alert notifications to valid email addresses. To help me in this, I will be using Firestore’s password management system. Also, I will need to read and write data, mainly to help me retrieve information from multiple retail sites about products.

### Data Structures

Since I am making use of price alerts, and having to store a lot of user information that will be stored in the cloud database Firestore, I will be employing the use of a Map. Using Firestore as my backend service will allow me to create new users with ease and write to the database without any issues.

### Objects / UML diagram

I will be making use of a User class that stores basic information like email, name etc. It will also have an arraylist of nodes or maps that is used to store the price point of an item the user would like to look out for.

Include UML class diagram (is a) if you're creating your own classes, or extending other classes with variables and getters/setters/other methods. Include a UML instance diagram (has-a) as well if you have multiple classes that use other classes in the field lists. Including a discussion of why you chose this design helps, listing the classes and what they are used for (a table of with class name and description won't count against your word count), and why you chose this organization of objects over something else..

If you're not using OOP, you should. If you're still not, you should include info on how you did your procedural decomposition.

### UI flows

I have not discussed about user flow yet with my client. I will be discussing this in the near future.

### Test Plan

|  |  |  |
| --- | --- | --- |
| **Success Criteria** | **Test Plan** | **Expected Outcome** |
| 1, 7 | 1. Upon start-up, test the security of the login page by entering random characters. 2. Users signs up for an account when “create account” is pressed. | 1. User stays on the login page until valid credentials are input. 2. Users are redirected to sign up page and then back to the login after successful creation of an account. |
| 2, 3, 6 | 1. Search up common household products. 2. Search products with typos in their name. 3. Type part of a product name in search menu. | 1. Listings from all online retailers are presented in a list view. The listings are presented in a hierarchical fashion where product listings at the top are the “best deals.” All listings have a price next to it. 2. If no product is found on retail sites, return feedback to user saying to type in a valid product name. 3. Returns accurate listings for desired product. |
| 4, 5 | 1. Search up common household products and click “set price alert” 2. Search up another common household products and click “set price alert”. Setting it to something very simple that can be met quite easily. | 1. Allows the user to select a price point for a product. When said price point is satisfied, an email notification is sent to the user’s inbox. 2. User should receive emails for all the products they are currently looking out for. |
| 6 | 1. Search up another product that is offered at many different retailers. | 1. When list view is presented, upon user clicking on a specific listing, they should be redirected to its respective web page where they can purchase it. |

**If you're missing any of these items, or if items are very brief, you'll likely lose a whole point for each (out of 6 total).**