

**Graduate Diploma in IT**

**Level 7**

**GIT701 System Development Integration II**

**Project Report:**

**2. Design**

(Worth 20% of final Mark)

**Final Result: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Assessor Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |
| --- | --- |
| **Student Name** | **Diego Evangelisti** |
| **Student ID** | **YIB00002MN** |
| **Student Signature** |  |
| **Student Name** | **Saeed Masoodnia** |
| **Student ID** | **YIB0000306** |
| **Student Signature** |  |
| **Assessor** | **Cassie Po** |

**Table of Content**

**INTRODUCTION…………………………………………………………………………………………..3**

**SCOPES AND OBJETIVES…………………………….………………………………………………..4**

**DELIVERABLES……………………………………………………………………………………………5**

**PROCESSES………………………………………………………………………………………………….6**

**TIMEFRAME AND MILESTONES……………….…………………………………………………..7**

**TEAM MEMBERS: RESPONSABILITIES……….………………………………………………..8**

**DOCUMENTS UPDATED……………………………………………………………………………….9**

RISK REGISTER**…………………………………………………………………………………………….…9**

COST REPORT **…………………………………………………………………………………………….12**

GANTT CHART**……….………………………………………………………………………………………13**

**REFERENCES………………………………………………………………………………………………14**

# Introduction

Based on the fact that the application is made for smartphones, and specifically for apple users, the design was planned to follow some pre-establish templates, buttons, etc that are already proved and tested. All of apple cell phones users are familiar with some design and functionalities and we found interested to keep them, with two purposes:

1. Get user attention and make the app easy to use and navigate throughout all pages.
2. Make our design easier to develop (less time consumer when designing it)

We want the users to find an easy to navigate application, that will allow them to find products and have the change to compare price easily, create lists and find the nearest branch for both shops (Countdown and New World Metro).

The following sections will explain the colour pattern selection, user flow, workflows, mark-ups, backend (includes database explanation, hosting and deployment of the project) and testing part.

**Structure and Navigation**

The following graph shows how the user can navigate throughout the application based on the decision made. It is called the user flow and explains mainly step by step the path the user will take when using the app.

The available pages are:

* **Start screen:** It shows background image related with doing shopping, the application name, a brief description and a button to start using the app.
* **Home screen**. In this page some random products are shown and on top there is a text search to enter the product name the user wants to find (see below: fixed bar on top).
* **List creation**. The first time the user should create a new list so then the product search can start.
* **Add items:** Includes search box to find a product by its name.
* **Decision table**: Compare products that matched with the product name search from Countdown and New World metro. The first products are the cheapest one available in each market.
* Product information: The data for each product is represented here. A big picture of it, price, quantity selection, product name, shop name and a brief description for some of them.
* **Products list:** Shows all the products chosen by the user. Here those products can be managed as follows:
  + - Delete from the list
    - Increment or decrement the quantity
    - The summarized price per each item is shown and the total price from the list as well.

Fixed bar on top:

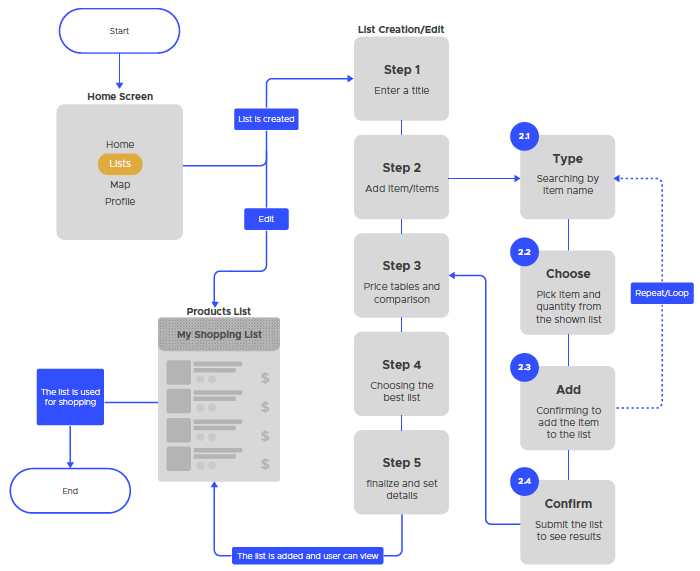
Text search that allows users to look for products by its product name. The products obtained are all that contains that specific word inserted in the search area.

Fixed bar on bottom:

On the bottom of every page there is a fixed bar that contains four options:

* + Home: Takes the user to the Home screen previously explained.
  + List: Show the products list previously mentioned.
  + Map: Show all markers with all branches’ location from both shops in a google map.
  + Profile: For future purposes. Should show the user information.

**User flow**



**System design**

1. **Brainstorming:**

Since the very beginning of this project we performed a brainstorming to come up with the best decisions about UI/UX attributes.

The following pictures show the first approaches taken during the preliminary sections.

Pictures here:

Brainstorming

1. **Paper sketches:**

Once we had our design planned, Saeed started to perform the sketches in order to get closer to our final design. Some examples of it:

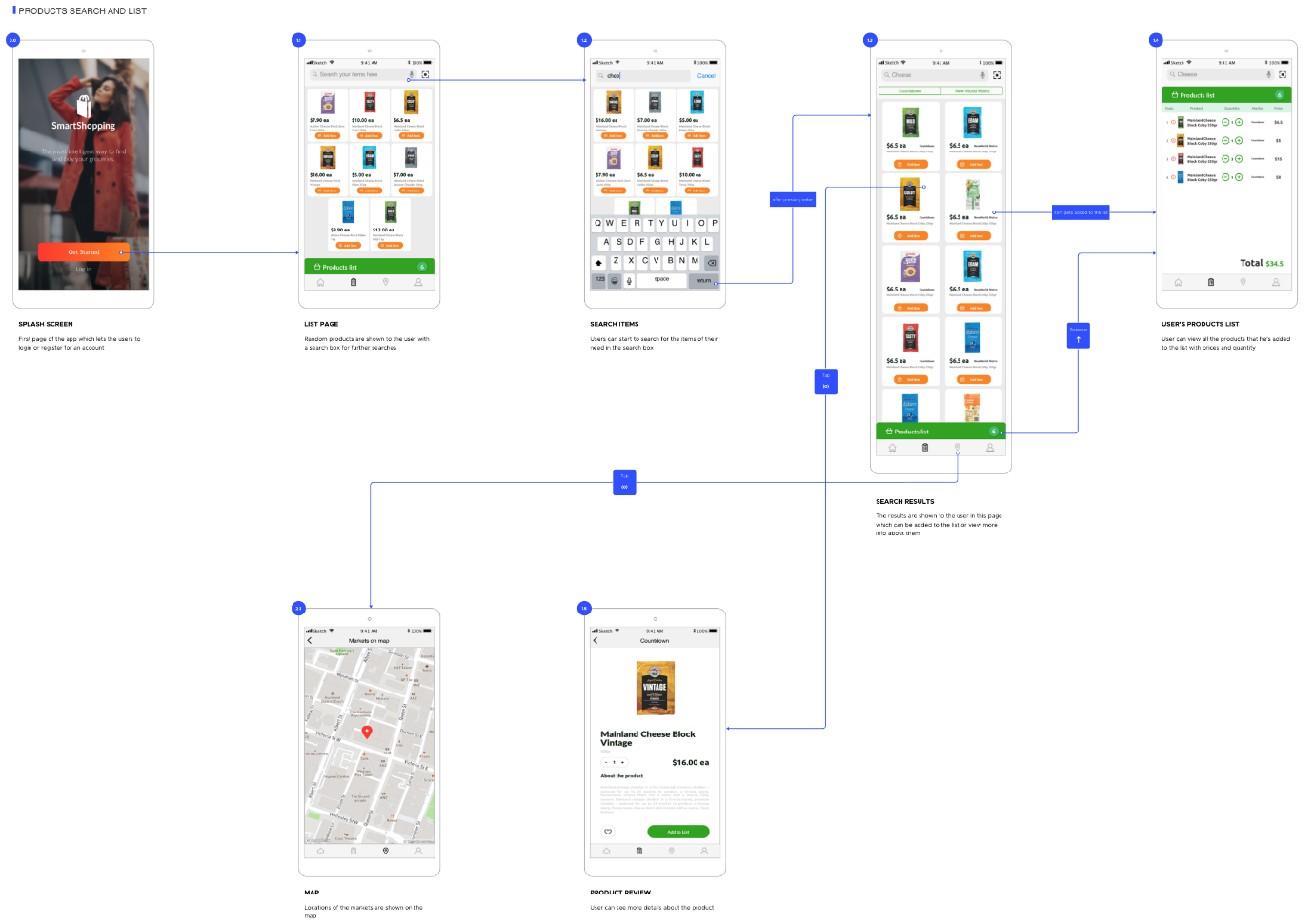
Pictures here:

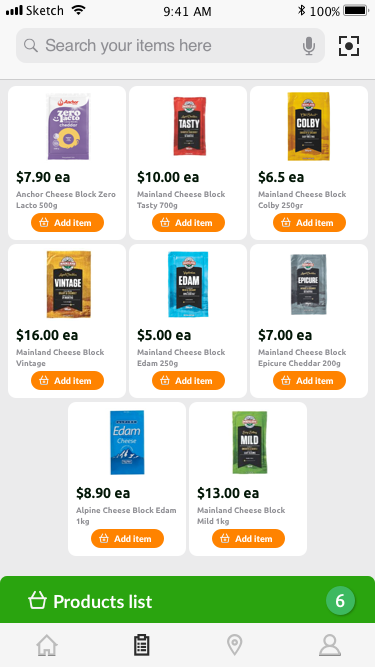
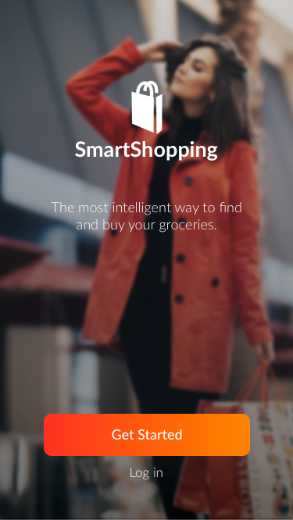
1. **Digital sketches**

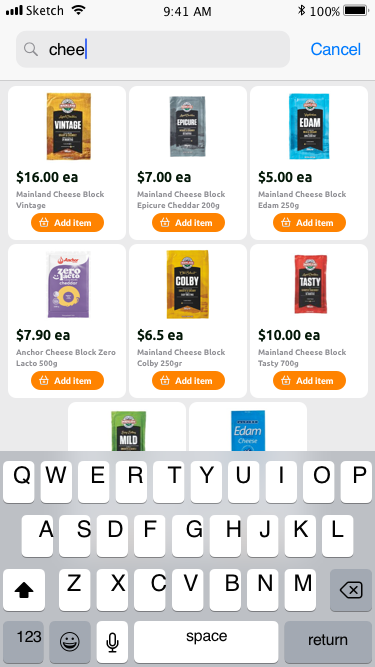
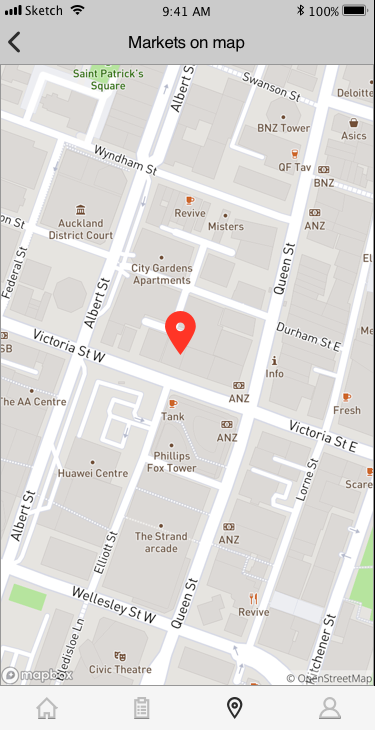
After sketching in paper all draws were translated to digital format using the Sketch software application, a powerful tool for building UI/UX design and saving the data in its own format (sketch format).

1. **Wireframing:**

After some iterations and discussions with the rest of the team, the final design was ready, and this is the result, which includes the wireframing with the user interaction included:

****

****

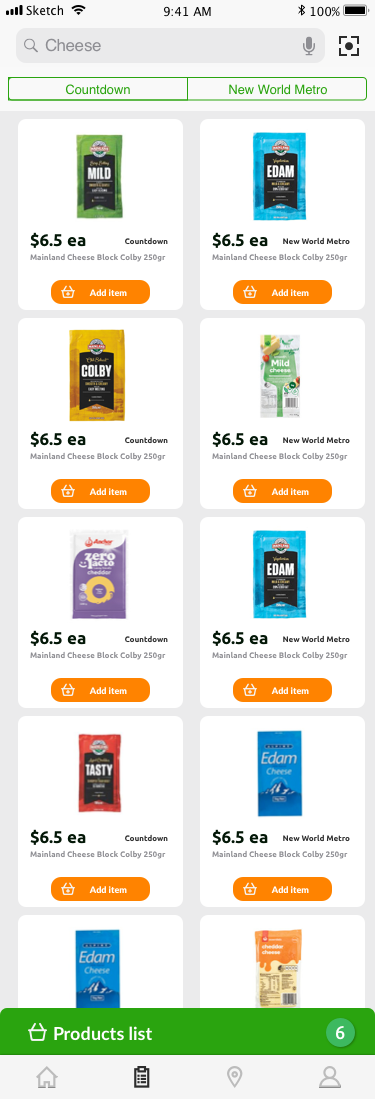
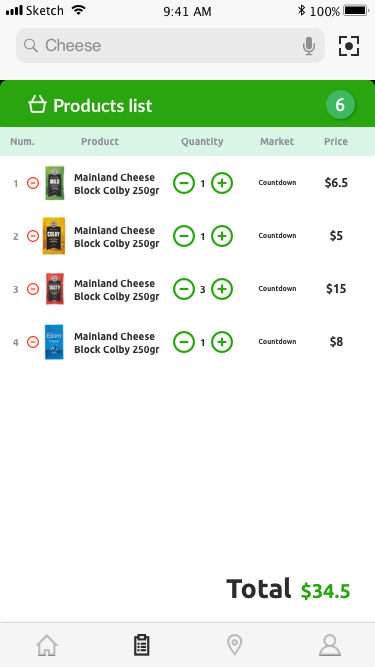
****

**4. Product search**

**3. Product search**

1. **Splash page**

**2. Product search**

****

**6. Product search**

**5. Product search**

**Design Users evaluation:**

The UI/UX was tested by three different people…

[EXPLAIN THE RESULTS HERE]

**Implementation**

**Functionality:**

* **Front end pages:**

**[ Explain how the front works, tools, etc]**

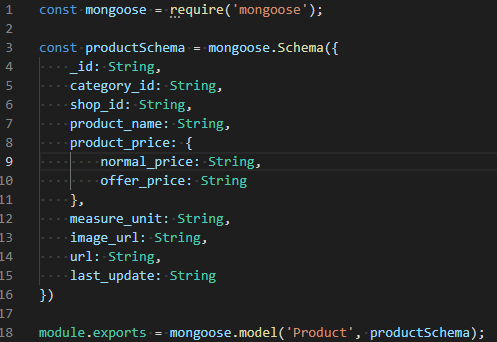
* **Databases access:**

For storing the data, we have chosen MongoDB (non-relational database), and we worked with Mongoose, a library for managing MongoDB in Node.js, the runtime application for back end purposes.

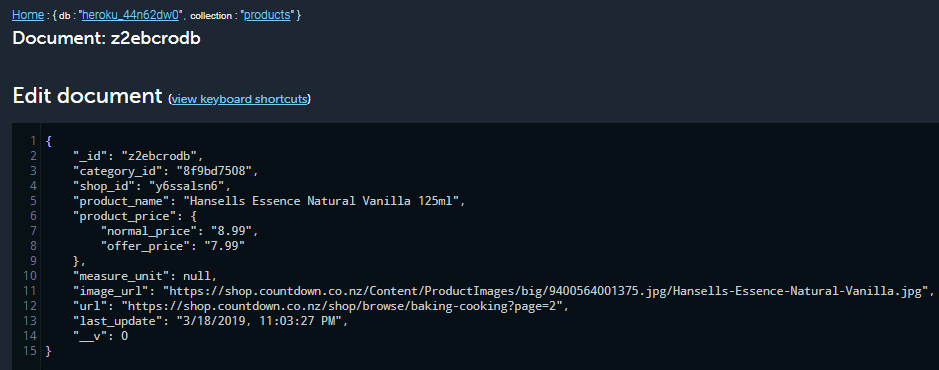
First of all, an ER Diagram (Entity Relationship Diagram) was performed in order to define all entities and its relationships needed.

The final result:

Mongoose uses schema and models to give the developer the possibility to structure the data, similar than the concept of schemas and tables for relational databases. Then, each model created is called a Collection, and each collection have documents (records) inside. In order to clarify this concept, we include the following example:



That schema and model is needed to create the data structure for the Product entity and setting up of all attributes Then, when products are created, the data type is similar than JSON (JavaScript Object Notation) format:



**The Collections we created are the following:**

* **Users**
* **Products**
* **Categories**
* **Shops**
* **Branches**
* **Lists**

# References

PayScale, 2018. Entry Level Software Developer Salary (New Zealand). Retrieved from: <https://www.payscale.com/research/NZ/Job=Software_Developer/Salary/89c3dc87/Entry-Level>

Udemy, 2019. Complete iOS App Development Bootcamp. Retrieved from: <https://www.udemy.com/ios11-app-development-bootcamp/>

Udemy, 2019. Node.js: The Complete Guide to Build RESTful APIs. Retrieved from: https://www.udemy.com/nodejs-master-class/