# Project proposal SIT305:

UniShop

By Harry L(218241616)

## Overview:

This app is a centralized location-based shopping app the looks at the stores in the area and finds the items that each place sells. It would work by looking into a saved data base of locations that each contain a list of items. From there you’d be able to search for items and you’d see the current price and availability for that item.

## Purpose:

the point of this app is to make it so that myself and others have an easy way to look at the stores in the area and compare them without the need to look up all of their individual websites or go to each I person.

The reason for this was when I’d go shopping, I would see that an item I just purchased was cheaper at Coles where my gf just purchased what I’d payed more for at the Woolworths. Along with the need to know if an item is in stock as there is uses that can arise from having to travel to somewhere far for a product, only to find it out of stock.

It’s design is it to ultimately save time and money for those who need to save and is targeted to allow people to make choices on their purchasing long before they leave to the store with plenty of time left over to.

### Target audience:

The main people that would use this app is those who are needing to save money on items. This would include uni-students, low income families and anyone who is looking to save on their everyday shopping.

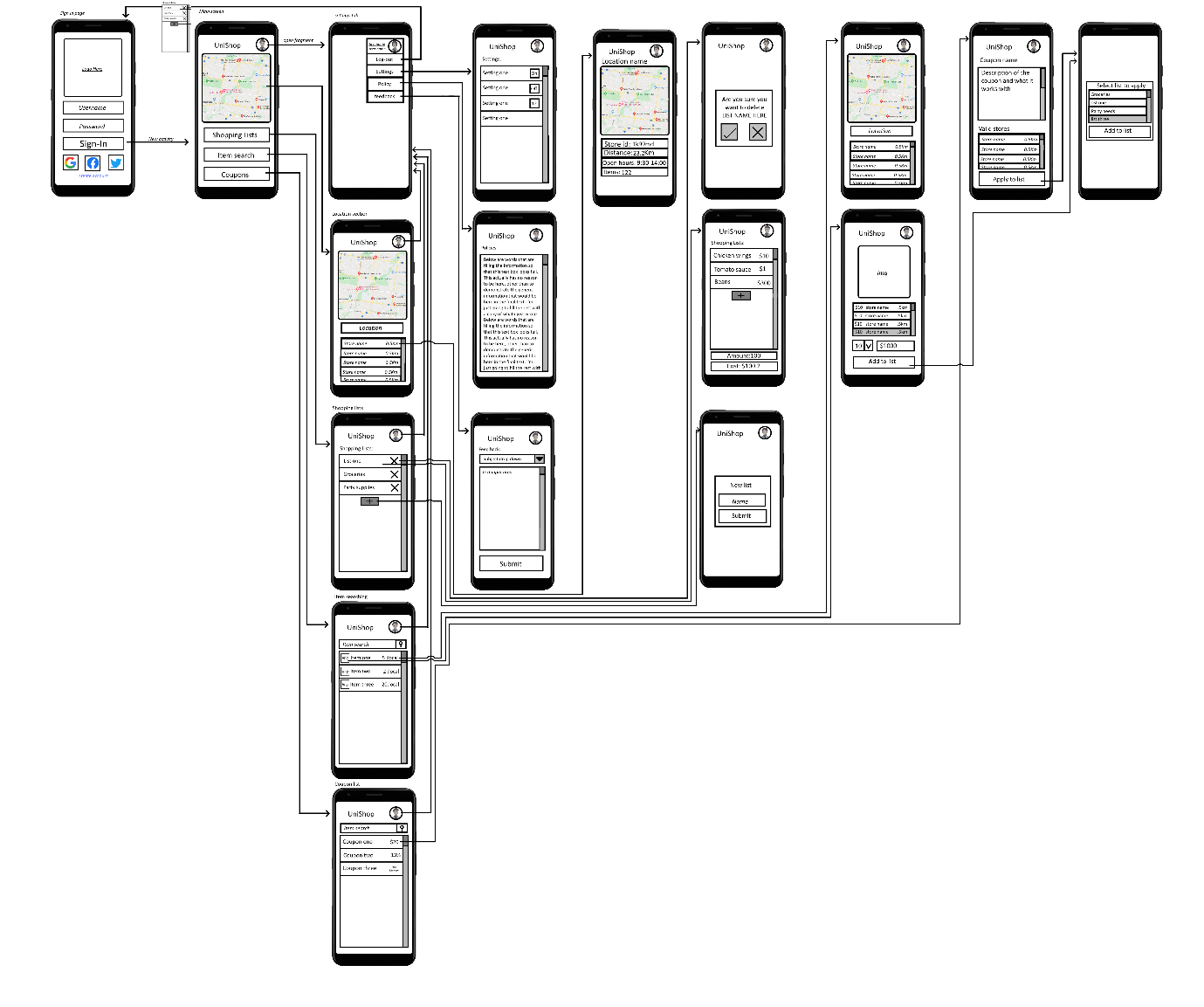
### Creativity:

The way that apps main goal is to be centralized much like you’d use something like the petrol spy app. The ability to have everything in one location and the ability to only need one app for online shopping/ planning shopping lists.

## Features:

The app is going to have a list of multiple features that would help it be worth the pricing. Below are the items that would be used to help justify the cost.

* Web crawler for tough websites
* Link to major brands
* Location based ability to search within given radius for all known grocery shops
* User based shopping lists
* Coupons/ discount register
* Api integration where possible
* History of item sale price

Design: 

## Data:

Runtime variables:

the variable that are created when running is the username, this is because the rest are pulled either from a webhost/api (see api/class structure) or internal storage.

Permanent storage data:

the database of items is to be stored on the web, along with the stores and their stockpile so as to reduce the size of the app and to better have up to date information for the user. This means that’s what’s left is for the users lists to be pulled from the internal storage which allows access to the last grabbed values and means there is semi offline use.

## Api/class structure:

3 main apis that I was going to implement but haven’t been able to due to the following reasons.

The google places but as that is charged when you use it. This charge meant that I wasn’t able to do testing without causing myself financial issues.

The second was the use of the map view which while implemented to the page, had issues with the views being grey when no other testing showed that happening.

The third and main api that I wanted to use was the Woolworths api which, due to lack of response on their end, I was unable to implement their api.

The api that I have been able to implemented quite well was the SQLite system in which I was able to substitute the information I would have used from both of them and created my own temporary files that will be place holder till I’m able to better acquire resources.

When it comes to these SQLite tables that I implemented, I focused around the use of 5 different main ones:

-DatabaseHelperItems

-DatabaseHelperStores

-DatabaseHelperStoreContents

-DatabaseHelperShoppingLists

-DatabaseHelperShoppingListContents

With the main items being stored in the items helper, I was able to replace the need for the Woolworths api as long as I could have the list filled to a decent amount before pushing out the app. the second list which focused on stores also worked in a similar way but so as to removed the need for google places and having to get the location information from the search feature. The final of these three lists make it so that you can store the entire list of items within a store without having to save an array to the tables.

The final 2 were focused around the use of keeping a table with all the id’s and names of the lists in the system so as to not forget a list. That list is then used to get the list of contents which contains the item id and quantity of said items so as to allow for better organizing and stat calculation.

External resources required:

all resources were created by myself and thus requires no licensing.