

Computer Science Department

Capstone Project Proposal

## **Market Basket Web App**

*Marouane Benchater*

**CSC 520**

CSC 521 Project Advisor

*Dr. Bo Hatfield*

## **Contents**

Student Objectives	3
Problem Specification	5
Solution Processes and/or Design	6
Requirement Collection and Analysis	7
Identifying Needs of Customers	
Identifying Target Specifications	
3 Tier Top Level Architecture Diagram	9
Detailed Design	10
Use Case Diagram	
Preliminary ER-Diagram (SQL)	11
UI Mock-Ups	12
Benchmark Specification	15
Tool List	16
Time Schedule	17
Grading Scheme for CSC521	18
List of Deliverables	19

## **Student Objectives**

The objective of doing this capstone project is to:

- Learn and be efficient with creating a web app using tools like: Php, SQL, HTML...
- Practice software development through the life cycle of a complete system,
- Design and implement a UI front-end for the end users,
- Enhance my skills on relational database design and implementation,
- Enhance the server-side programming skills.

## **Motivation:**

During quarantine, a lot people could not go out and shop as they were afraid to get sick.

That's when I came up with the idea of a Web App where people can order and get their grocery delivered to their doors without having to wait in line to get into stores or risking getting sick.

The plan is: to learn and implement all the steps mentioned in this proposal in order to build a Web App that allows users to shop for food online and save time.

### **What the Web App will accomplish:**

The app will have items that are carried by the supermarket, shoppers will be able to

browse any item, add it to their cart, and remove it from their cart.

it will display “deals of the week” which is items that are on sale. After the shopper is

done shopping they can check out and pay for their order.

## **Problem Specification**

The benefits of this project are both non-financial and financial. Some of these benefits

are: Faster shopping: this web app is going to make shopping a lot faster for users.

Customers will no longer have to leave their houses and drive to the supermarket, it will be one click away.

No more waiting in queue: the web app will have a feature that shows when the supermarket is busy so customers will avoid it in those times in case they decide to go out and shop themselves.

Search: searching for a product will be easy and instant; if an item is out of stock the admin will be able to update the stock so there will be no wasting of shoppers' time.

## **Solution Processes and Design**

The supermarket web application's back end is going to consist of a database that holds all the useful information containing items, prices, aisle numbers, deals of the week.

These attributes are going to be put together to form meaningful data for the shoppers when they use this web app. The back end will also be designed to display a login page that requires a password to access the web app in order to shop (shoppers) or to manage (admin).

As for the front end, it will be very clear and easy to use for both shoppers and admins. A simple homepage with buttons on the side for categories such as: items, cart, deals of the week...

## **Requirement Collection and Analysis**

### **Stakeholder Identification:**

The Stakeholders are:

- Users: shoppers that can access the web app. (shoppers will have the ability to browse items, add items to cart, delete items from cart, view deals of the week...)
- Admin: store managers. (Admins will have the ability to manage the webapp such as to add items to database, delete existing items, manage aisles, and update the deals of the week...).

### **Identifying customer's needs:**

- Less time shopping: by shopping online users avoid waiting in a queue.
- Easy to use app: the app will have a simple to use front end, with the most seller items in the homepage. With just few clicks to order, instead of wasting time physically browsing in the supermarket.
- Conveniency: The food will be delivered to the shoppers doors after they check out. Very convenient to people that don't have the time to go out and shop.

### **Functionalities for the shoppers:**

#### **Group 1**

- Login and shop: shoppers can login using their email and password.

- Search for items: Shoppers can search for any items either by browsing items or typing a keyword in the search bar.
- View deals of the week: users can click on “deals of the week” to view what’s on sale.
- Check for supermarket hours: the WebApp will have the supermarket hours.

## **Group 2**

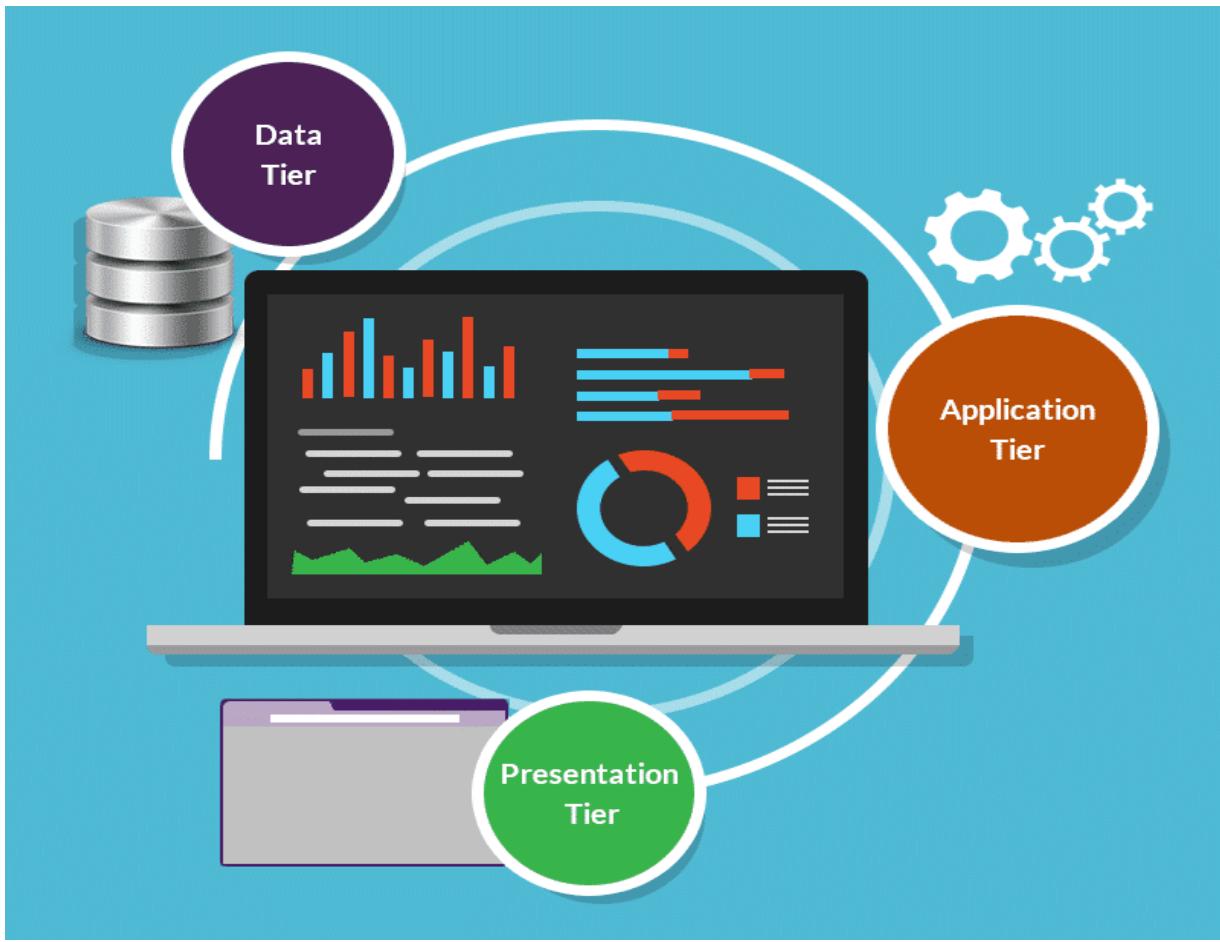
- Add/Remove items to/from cart: while users browse they can click on an item and add it to the shopping cart. If the user wants to delete an item from the cart, they can click on “shopping cart” and delete it.

## **Functionalities for the admin:**

- Login: Admins can login using their emails and passwords.
- Add new items: Store managers can add new items to the database.
- Delete existing items: When an item is discontinued, the admin can delete it from database.
- Update prices: admins can change item prices
- Manage aisles: move items from an aisle to the other, remove/add items to aisles
- Update deals of the week: managers can update items on sale.

The application must go under several tests ensuring that it doesn’t have any bugs. A database is needed for items, aisles, and prices. The app needs to be updated frequently by the admin. The goal is to maintain an easy-to-use application, up to date, and develop it when needed, fixing any problem that can be encountered.

### 3 Tier Top Level Architecture Diagram

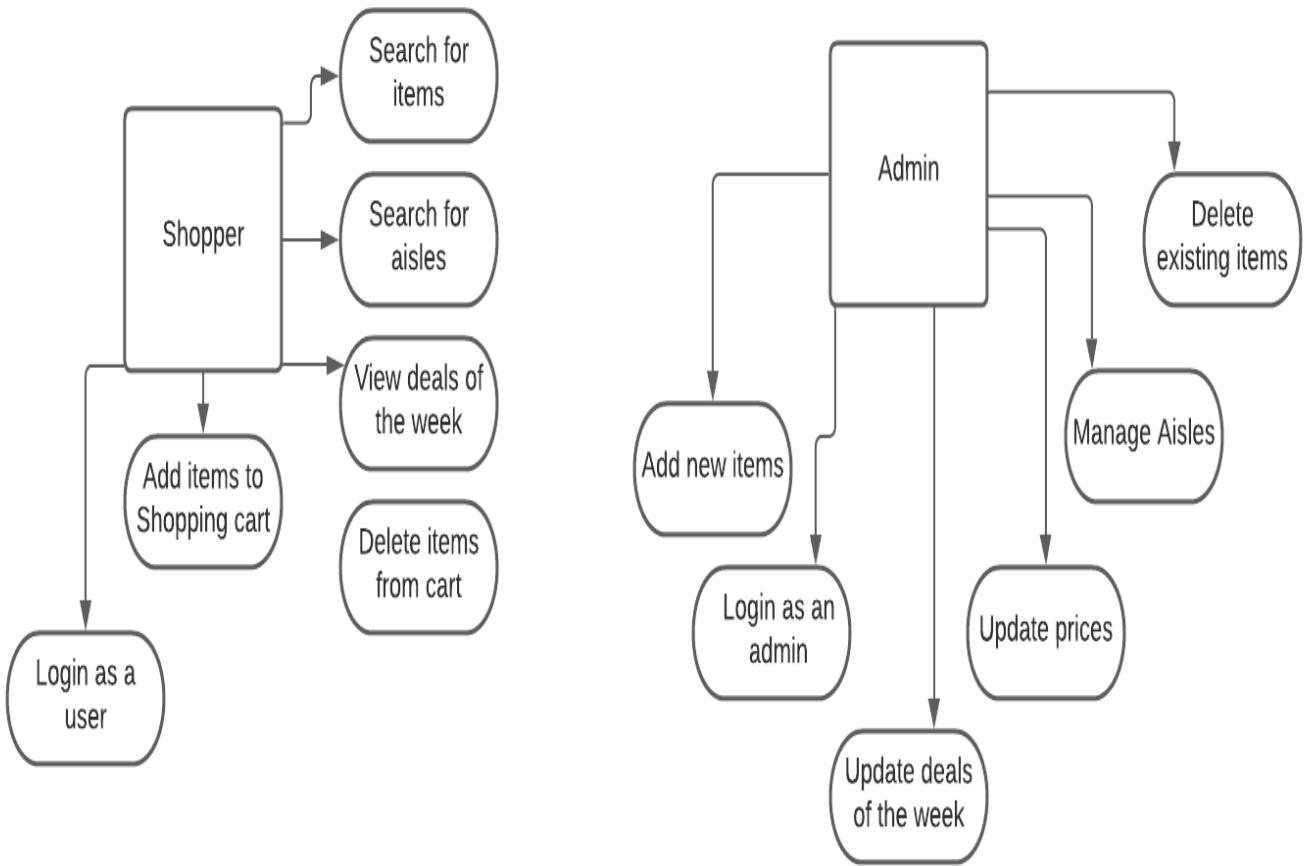


**Presentation Tier** The presentation tier is the front-end layer in the 3-tier system and consists of the user interface.

**Application Tier** The application tier contains the functional business logic which drives the application's core capabilities.

**Data Tier** The data tier comprises of the database/data storage system and data access layer.

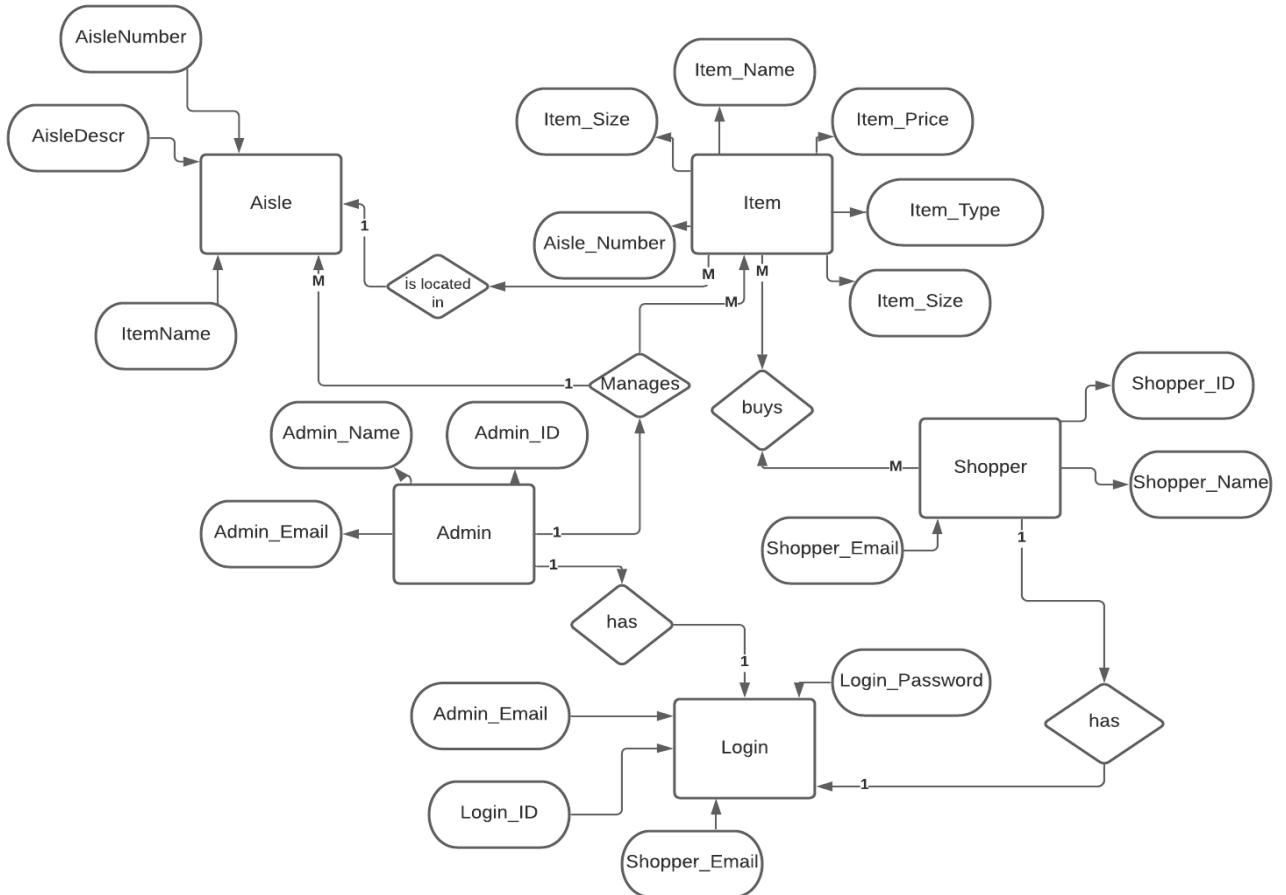
## Use Case Diagram



---

The use case diagram above describes functionalities for both admins and users.

## Preliminary ERD

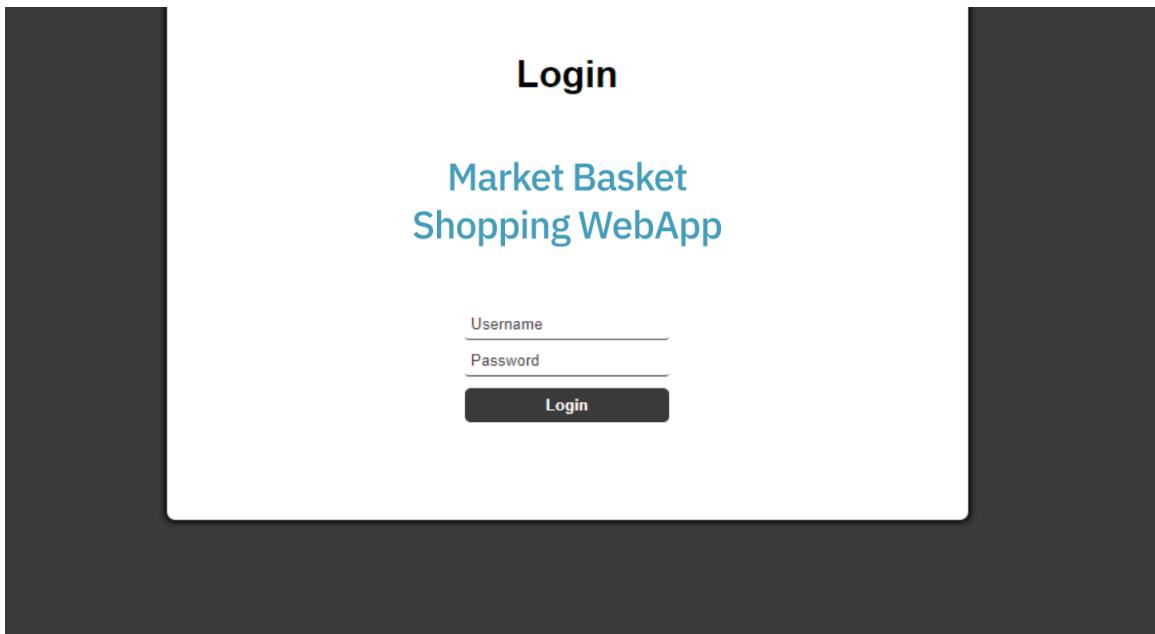



---

The ERD shows the relationship between entities (Aisle, Item, and Price) stating their attributes. The relationship between Aisle and Item is a on to many, while the relationship between Item and Price is a one to one.

## UI Mock-Ups

### Prototype



---

This is how the login page should look like, it consists of a login button a prompts the user to put in a Username/Email and Password.



The screenshot above is how the homepage would look like to a user/shopper. It consists of buttons for each function that this webapp has. (Shopping Cart, Aisles, Items, Deals Of The Week, Hours of Operation)



The screenshot above is how the Webapp would look like when an Admin/Store Manager logs in. it consists of functionalities like (Add/Delete Items, Manage Aisles, Update Prices, Update Deals Of The Week and Update Hours of Operation)

## **Benchmark Specification**

- **Benchmark 1:** Database design and implementation + testing
- **Benchmark 2:** Implementing shopper's functionality group 1 + Testing (Front-End and Server Side)
- **Benchmark 3:** Implementing shopper's functionality group 2 + Testing (Front-End and Server Side)
- **Benchmark 4:** Implementing admin's functionality + Testing (Front-End and Server Side)
- **Benchmark 5:** Integrated Testing + Report Writing

## Tool List

### **Front-End Development Tools:**

HTML

CSS

JavaScript

### **Server-sign Programming Tools:**

PHP

### **Back-End Database Tools:**

MySQL

phpMyAdmin

### **IDEs:**

Notepad++

### **Hardware Tools:**

Linux server

MySQL server

### **Other tools:**

FileZilla

## **Time Schedule**

**Benchmark 1:** Should be done by third week of February

**Benchmark 2:** From third week of February to second week of March

**Benchmark 3:** From second week of March to fourth week of March

**Benchmark 4:** First two weeks of April

**Benchmark 5:** Third and fourth week of April

## **Grading Scheme**

**Benchmark 1:** 15%

**Benchmark 2:** 25%

**Benchmark 3:** 15%

**Benchmark 4:** 20%

**Benchmark 5:** 15%

**Presentation –** 10%

## **List of Deliverables**

- Original proposal and presentation files
- Amendments to the proposal
- Appropriately commented source code
- Documentation of project functionality
- Sample output
- Executables
- Presentation documents
- Project journal
- Project post mortem
- Presentation of the completed project