Technical Design Document

Shopping List Project

Team 5

# Version History

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Description** |
| 2016/05/15 | Jared Wooten | Initial Creation |
| 2016/11/29 | Team 5 | Version 0.0 |
|  |  |  |

# Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| SOW | Statement of Work |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# References

1. SOW

# Table of Contents

[1 Version History - Pg. 2](#_gjdgxs)

[2 Acronyms - Pg.](#_30j0zll) 2

[3 References - Pg. 2](#_1fob9te)

[4 Table of Contents - Pg. 3](#_2et92p0)

[5 Introduction - Pg. 4](#_tyjcwt)

[6 Scope - Pg. 4](#_3dy6vkm)

[7 Technical Architecture - Pg. 5](#_1t3h5sf)

[7.1 Use Cases - Pg. 5](#_4d34og8)

[8 Technical Design - Pg. 6](#_2s8eyo1)

[8.1 Design Constraints - Pg. 6](#_17dp8vu)

[8.2 Data Model - Pg. 6,7](#_3rdcrjn)

[8.3 User Interface - Pg. 8](#_26in1rg)

# Introduction

The purpose of this document is to expand upon the SOW [1] and define the technical architecture and high level design required to fulfill project scope. Anything not included in this document is considered out of scope and may incur additional time and costs.

Our customer proposed a Shopping List application designed to provide a simple, straightforward way for users to keep track of their shopping lists. The application will provide each user with the ability to create new lists, add items to each list and check off items from a list. With this software, our customer is able to provide an easy and simple way to keep track of any type of list.

The user will access the application home page, which is where the lists will be seen. From there, a user can choose to view those lists, add a new list, or delete a list. If a user chooses to add a list, they will be prompted to add items to that list. Otherwise, if they choose to view a list, a user will have the ability to check off items they have already purchased, delete items off of the list, delete the list itself or add more items to the list. From the home page the user could also delete lists.

The main goal of the EFC Shopping List Application is to provide users with a better alternative to tracking and maintaining lists, opposed to the traditional paper and pen.

# Scope

This document is structured as follows:

* Technical Architecture
* Technical Design
* Platform Specific Technical Information

# Technical Architecture

**MVC:** We will be using the model view controller design pattern, which is designed to separate the user view’s view (site interface) from the model’s business/domain logic (domain logic is a more general term, but it encompasses what we normally refer to as business logic).

**MSSQL Database**: This component is the database where backend data will be stored. The application will be developed initially using an H2 database for testing purposes, and then migrated to the chosen external database provider.

**Data Model**:The model represents the logical structure of data in the application and the high-level classes associated with it. In other words, the model handles the business and/or domain logic that is built with the Java code we have written and also includes the data we use within our program. The model will include domain objects, which typically consist of the classes we build, including Java beans and POJOs.

**Services**: This component is the assembly that encapsulates the business logic of the app. This component is the assembly that encapsulates the business logic of the app. Domain services are typically written to interact with the data.

**UI**: This component is the main UI of the app defined as an MVC web application. See section 8.3 User Interface.

**Web API:** This component exposes some aspects of the app to external users. It is defined as a Web API 2 web application.

## Use Cases

The application will be written in Java using the Spring Boot MVC framework and Thymeleaf template engine and deployed to Heroku or a similar hosting service. The data will be stored in ClearDB database. Unit testing will be performed manually. The front end will implement HTML, CSS, and possibly JavaScript.

### Login Page

Users don’t need to login as there is no login required to use the app yet. Security configuration has not been set up yet.

### Shopping List CRUD Pages

Users should be able to create, read, update, delete, sort, and check off list items.

# Technical Design

## Design Constraints

The application will be compatible with Chrome browsers. Chrom 54 was used in creation of the application. The interface will be responsive so that users on mobile devices have a good experience.

## Data Model

The Data Model will use a Code First strategy with Data Migrations enabled to facilitate ease of deployment end incremental database development.

### ShoppingList

|  |  |  |
| --- | --- | --- |
| Id (key) | Int | Unique Id |
| UserId (foreign key) | Int | Relates to a User |
| Name | String | Name of shopping list |
| Color | String | Color to display as (hex value) |
| CreatedUtc | DateTimeOffset | Utc created date |
| ModifiedUtc | DateTimeOffset | Utc modified date |

### ShoppingListItem

|  |  |  |
| --- | --- | --- |
| Id (key) | Int | Unique Id |
| ShoppingListId (foreign key) | Int | Relates to a ShoppingList |
| Contents | String | Description of item to shop for |
| Priority | Int (Enum) | One of several enumerated values: “It can wait”, “Need it soon”, and “Grab it now!” |
| IsChecked | Bool | Checked items are purchased/complete |
| CreatedUtc | DateTimeOffset | Utc created date |
| ModifiedUtc | DateTimeOffset | Utc modified date |

### Note

|  |  |  |
| --- | --- | --- |
| Id (key) | Int | Unique Id |
| ShoppingListItemId (foreign key) | Int | Relates to a ShoppingListItem |
| Body | String |  |
| CreatedUtc | DateTimeOffset | Utc created date |
| ModifiedUtc | DateTimeOffset | Utc modified date |

**8.2.4 Shopping List Item Priority**

|  |  |  |
| --- | --- | --- |
| Id(key) | Int | Unique Id |
| Name | String | Description of the priority (“It can wait”, “Need it soon”, and “Grab it now!”) |
| CreatedUtc | DateTimeOffset | Utc created date |
| ModifiedUtc | DateTimeOffset | Utc modified date |

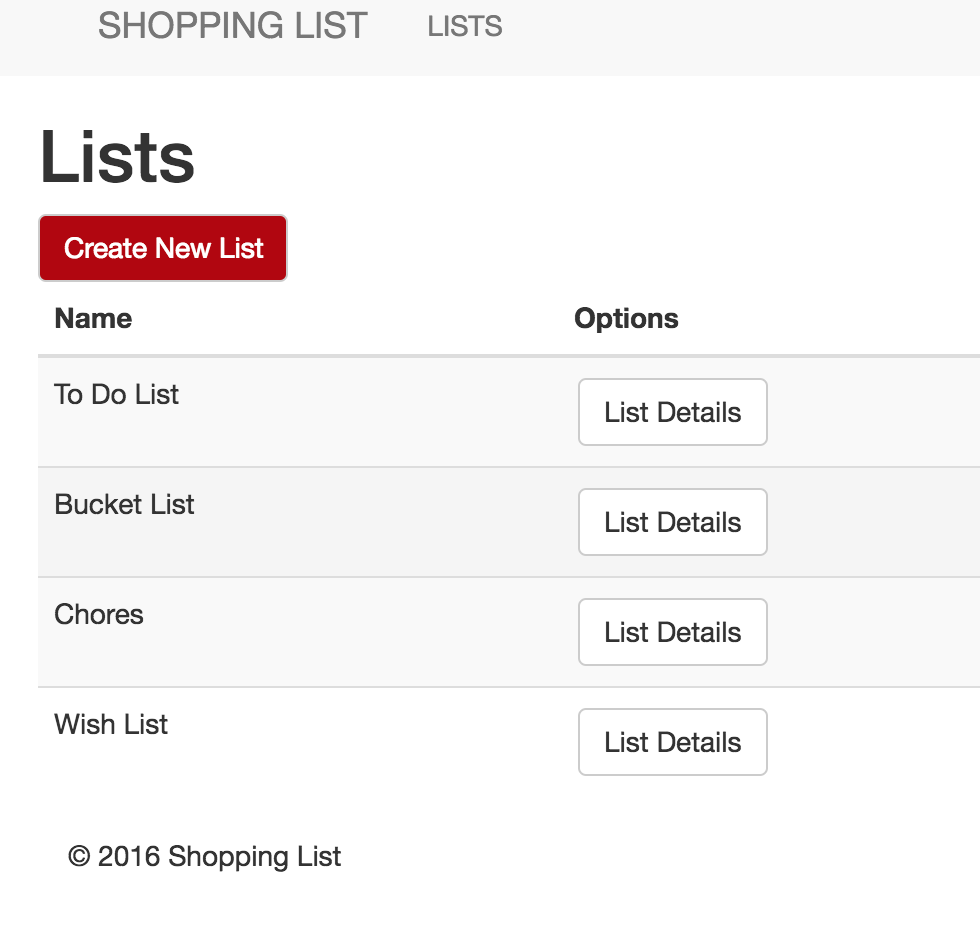
## User Interface

Mockups of the main pages of the app.

### Shopping List – List page

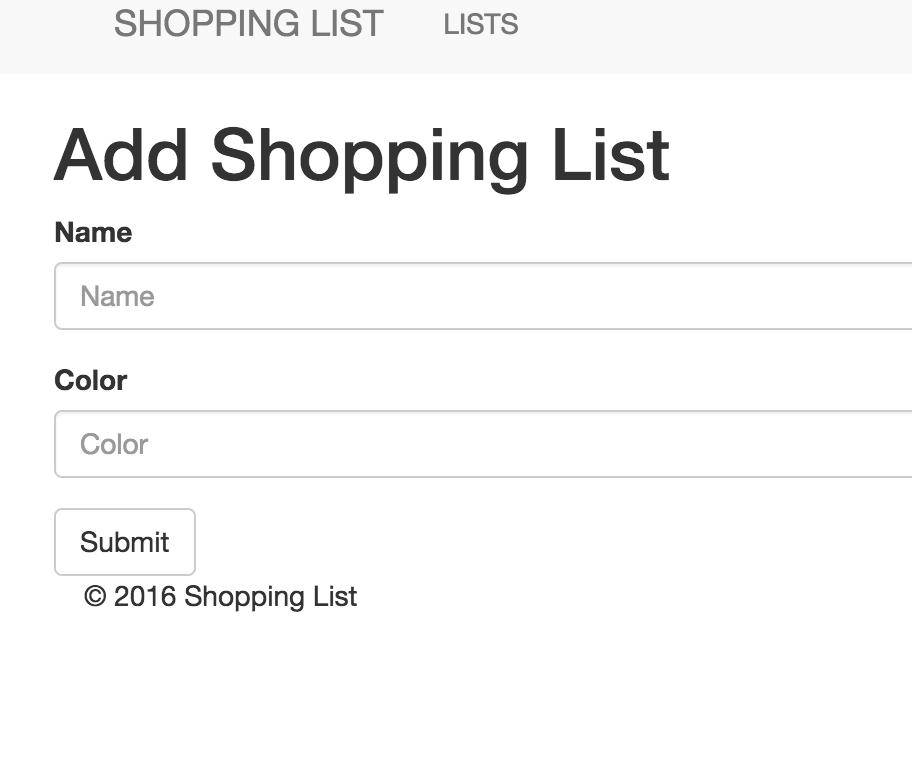
This page lists all shopping lists for the logged in user and has the following functions available:

* Add New List
* (per list item) View List
* (per list item) Delete List



### Shopping List – Add page

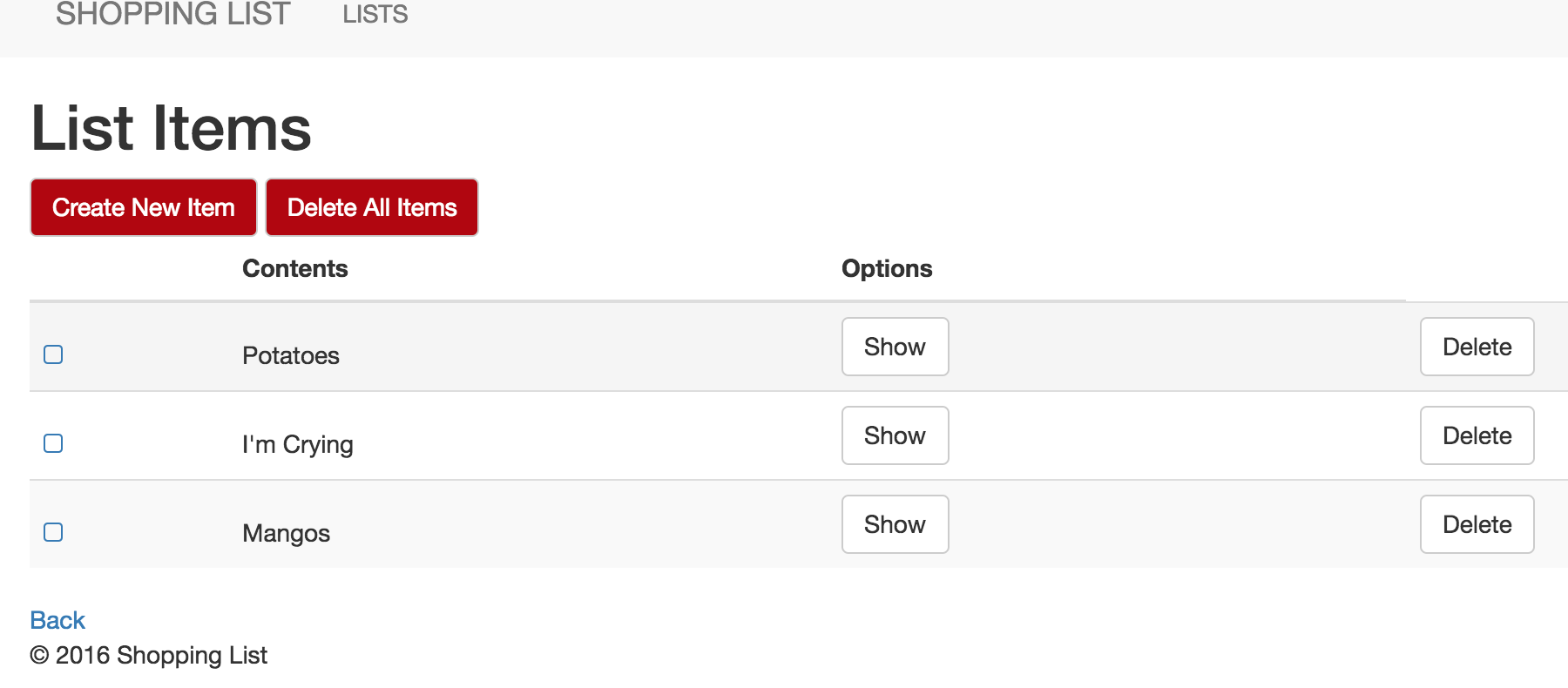
This page is for adding a new shopping list.



### Shopping List - Show page

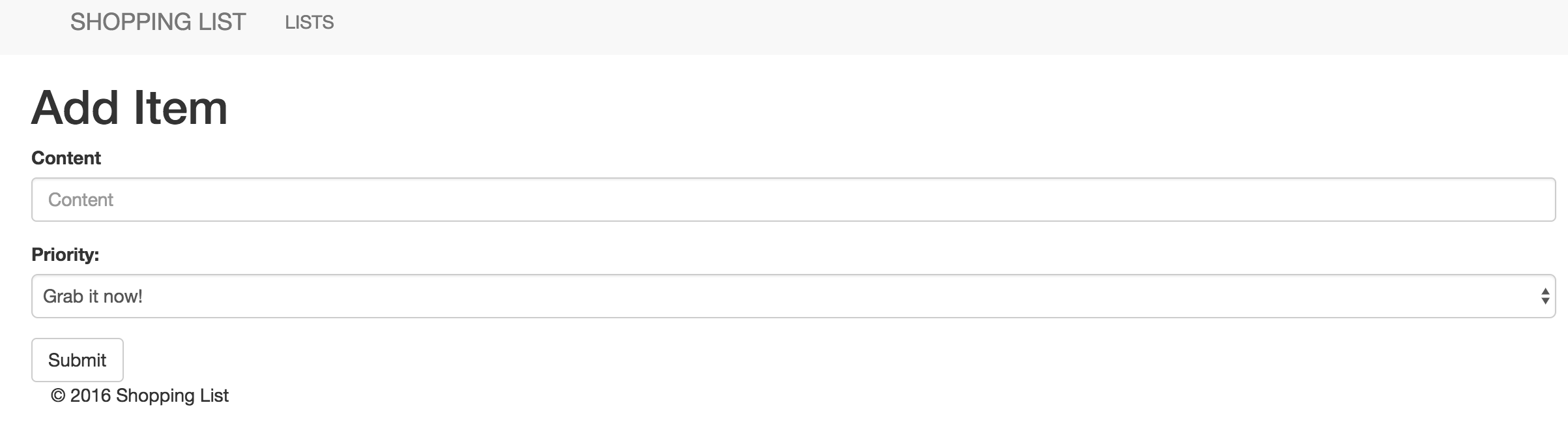
This page lists all shopping list items for the logged in user as well as the following functions:

* Add New Item
* Clear Checked Items
* Sort Items
* Page Items
* (Per list item) Check Off Item
* (Per list item) Add Note to Item
* (Per list item) Delete Item



### Shopping List Item - Add page

This page is for adding a new shopping list item.



# 