Shopping List Application

SE452 – Object Oriented Enterprise Application Development

Paul Warren

Final Submission

Contents

[Overview 2](#_Toc524376933)

[Requirements 2](#_Toc524376934)

[Use Case 2](#_Toc524376935)

[Description of problem 2](#_Toc524376936)

[Design 3](#_Toc524376937)

[Sequence of major functionality 3](#_Toc524376938)

[Web UI (Common case) 3](#_Toc524376939)

[Table layout 3](#_Toc524376940)

[Deployment 3](#_Toc524376941)

[Discussion of how your design met the requirements 3](#_Toc524376942)

[Discussion of lessons learned 4](#_Toc524376943)

[Milestone Screenshot 4](#_Toc524376944)

[Decision Log 5](#_Toc524376945)

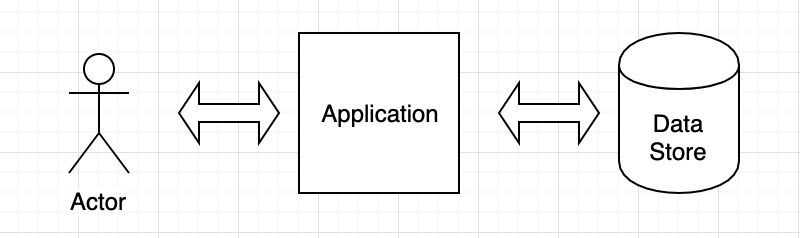
# Overview

The application is a web UI to allow people to create a shopping list and retrieve it later. The application supports basic features like adding, removing and updating items from the list, storing it for later use, and is able to share it with others via email. The application will also allow them to categorize the list by store for easier shopping.

# Requirements

## Use Case

Typical use of application:



User will use the application through the web UI to create necessary changes to their list, which will then be persisted in the data store for later retrieval or editing.

## Description of problem

Create an application that can be used by individuals to create a shopping list. Three basic pieces of functionality are listed below:

1. Add/view/remove/delete things to the list
   1. As a user, I would like to add items to the shopping list.
   2. As a user, I would like to remove items from the shopping list.
   3. As a user, I would like to update items on the shopping list.
   4. As a user, I would like to view items on the shopping list.
2. Categorize by store
   1. As a user, I would like to categorize the items on the shopping list by the store I enter.
   2. As a user, I want to sort items by the store.
3. Share via email
   1. As a user, I want to share my shopping list with other email addresses.

Additional Feature Requests (Currently out of scope)

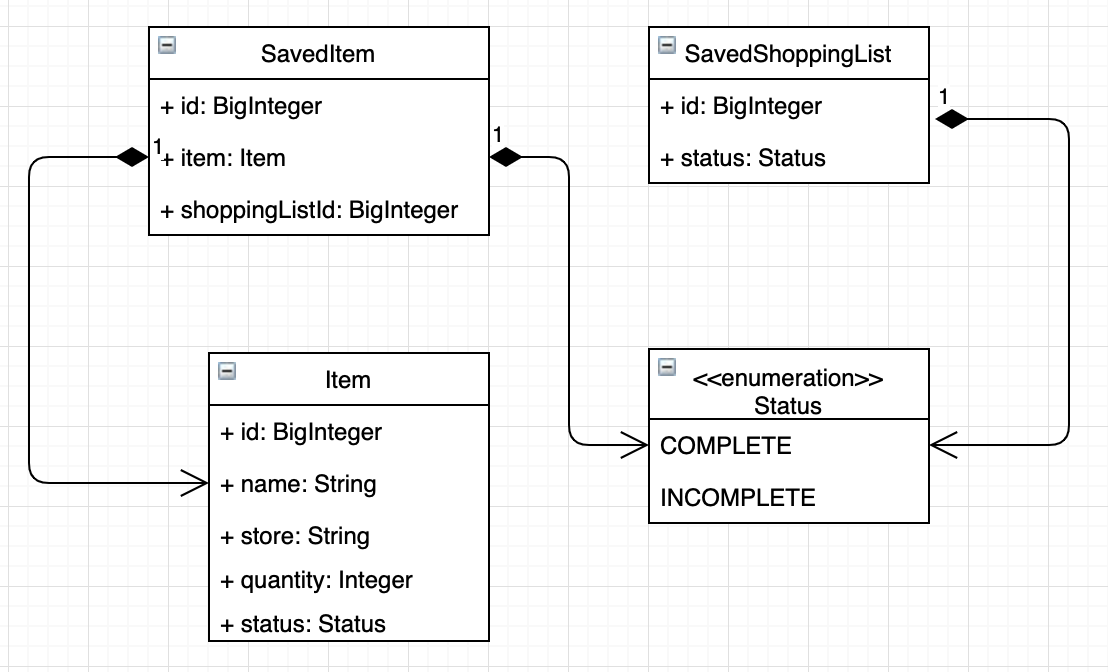
1. Shopping list templates
   1. As a user, I want common items to be pre-populated on my list

# Design

## Sequence of major functionality

### Web UI (Common case)

## Table layout



The documents will be stored in an in-memory MongoDB.

## Deployment

The application will be deployed as a single artifact for simplicity.

# Discussion of how your design met the requirement

The application first lists all available shopping lists. On this page the user can also create a new shopping list. Once the user selects a shopping list they can view the elements already on it, add more, or mark some complete. At the bottom of the page the user can email the shopping lists to different email addresses, and there is a link to return back to the list of all shopping lists for convenience.

# Discussion of lessons learned

Using a templating engine made creating the UI easy. It was especially easy since I could also use the templating engine to also create the email format.

# Appendix

11/12/2018 – The email service is not complete and will send HTML formatted email to the user inputted email address. Clicking on the column headers sorts the values ascending or descending based on the values in the table. One of the issues I have is the row colors don’t change which doesn’t look good.

11/6/2018 – The update/add functionality exists on both the create shopping list and view shopping list page. This means users can edit and change their lists from the UI. The next step will be to add the final functionality, which is to email shopping lists.

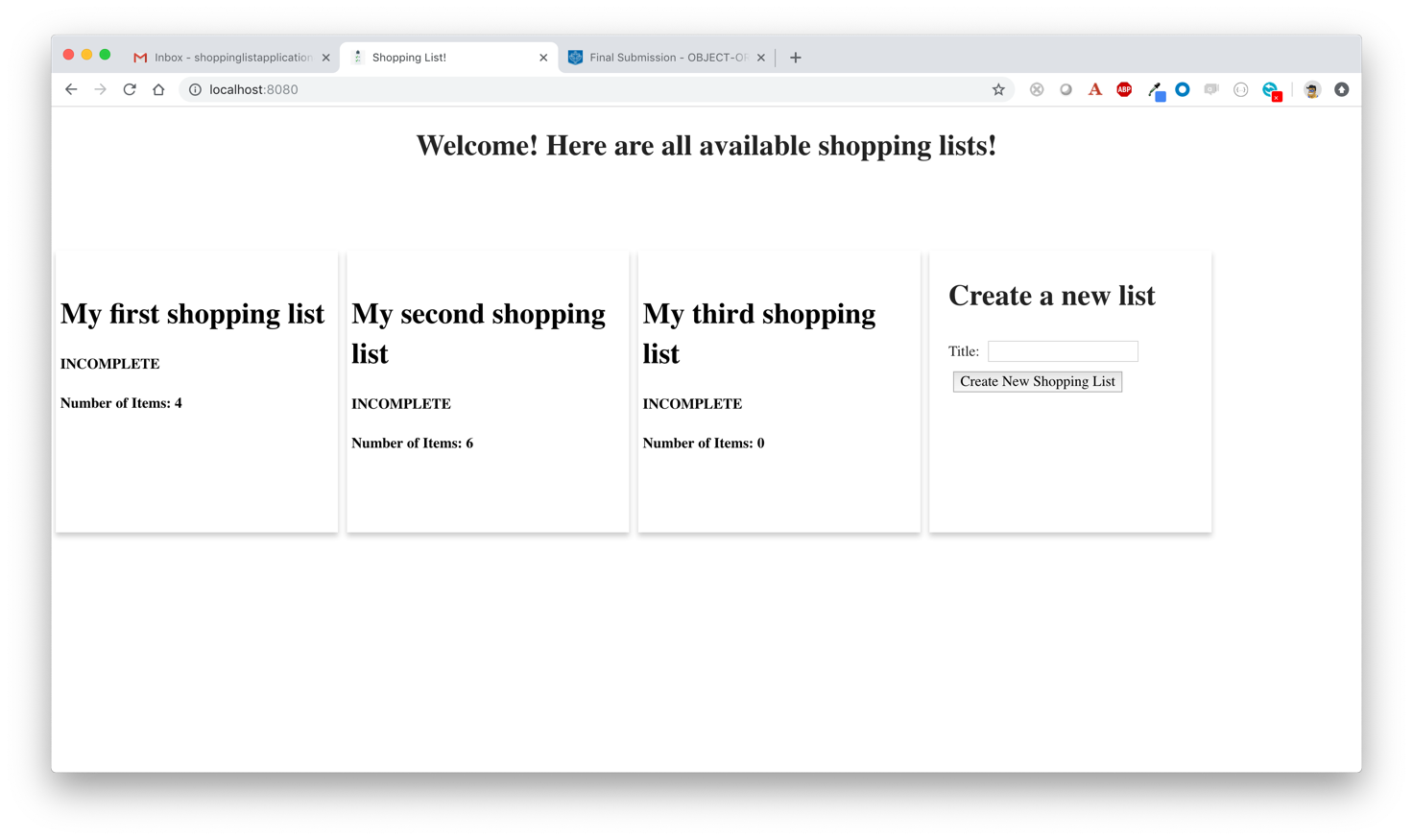
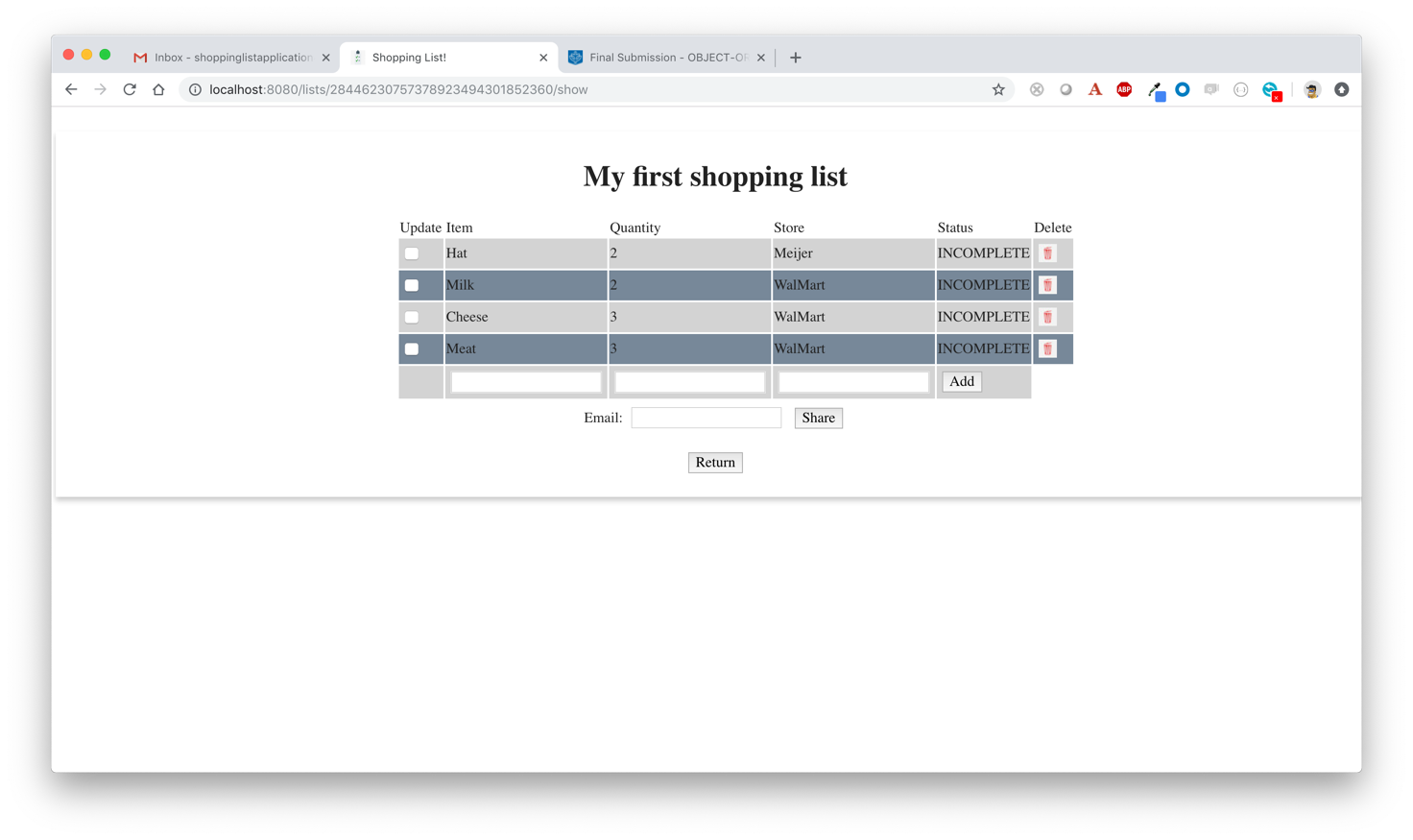
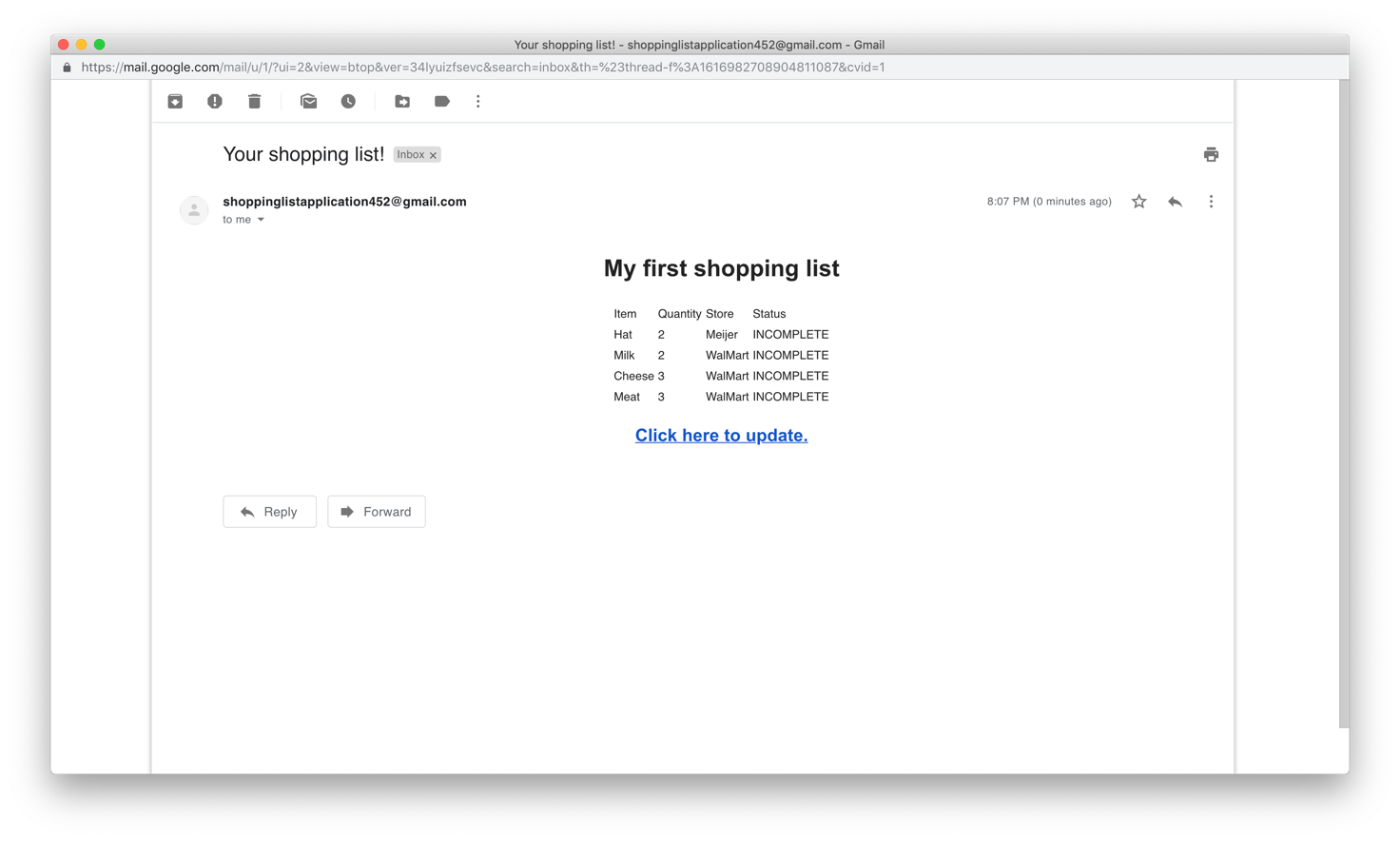
10/23/2018 – Users are able to see all the data currently stored through a web browser. The next steps are to add the submit features in order for users to create new shopping lists, new items, and to update the state of current items. After that to submit the email functionality and configure it to send emails.

9/24/2018 – In order to test the database logic, I added a REST controller to accept network calls. The basic calls exist:

* Create a shopping list
* Create an item in a shopping list
* Edit the item in a shopping list
* Delete an item from a shopping list
* Delete a shopping list

I am also looking to add some feature tests to validate the application works successfully.

Running the application should be as easy as clicking start. The application should download the embedded Mongo dependencies and start the database.



# Decision Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem** | **What was decided** | **Alternatives considered** | **Rationale** |
| Which IDE to use | Intellij | Eclipse, Netbeans | Intellij is free for students. Intellij is powerful and should make development easier. |
| Code repo | Github | None | Backup code to Github in case of catastrophic computer failure |
| Database Technology | MongoDB | H2 | Decided to use spring data mongo since I have not used it before. Document store seems to be fun. |
| Templating Engine | Thymeleaf | JSP, Servlets | Easy with MVC design pattern, Spring has good integration with Thymeleaf |
| Design strategy | Material | other patterns | Material design seemed like something that would fit a shopping list theme |