Smart Waiter System Architecture

Meraj Patel #1137491
Pavneet Jauhal #1149311
Shan Perera #1150394

January 10, 2016

Contents

List of Figures

List of Tables

Revision History

Date	**Comments**
October 9, 2015	first draft.

Table 1: Revision History Table

Template

This document uses Volere Template for its organization.

1 Introduction

1.1 Purpose

1.2 Description

1.3 Scope

2 Overview

3 System Architecture

4 Orders

4.1 Overview

Below are key details of ordering component for Smart Waiter application. Detailed architecture review is provided.

4.2 Ordering Structure

There are three primary classes used to hold all vital information regarding ordering. These are: MenuCategories, MenuItems and User. Each class purpose and its correlation with each other is explained below.

4.2.1 MenuCategories Class

This class purpose is to store menu category information of a restaurant menu. This entails category name, picture and a reference to category items. To do so, there are three main variables used within this class to hold this information:

categoryName: Stores category name in the form of String

picURL: Stores URL of category picture

categoryItems: Stores an array of objects that reference MenuItems class

This class is instantiated and called upon when a user successfully scans a barcode. The JSON response acquired from the database is parsed, and information related to menu categories is stored appropriately within this class. The application uses this class to display category information when ”Menu Categories” page is spawned (please see section 5.1.1 to view picture).

4.2.2 MenuItems Class

This class purpose is to store menu item information of a restaurant menu. This means, item name, price and description. Thee main variables are used within this class to hold this information.

itemName: Stores item name

itemPrice: Stores item price

itemDetail: Stores description of item

Objects of this class are instantiated from the MenuCategories class. By successfully scanning the barcode and parsing the JSON response, MenuItem objects are instantiated and used to save item details. These objects are held within categoryItems list seen in Menu Categories class. This application uses this class to display item information of a category when ”Menu Items” page is spawned (please see section 5.1.2 to view picture). These objects are also used in reference to items the user would like to order. This is thoroughly discussed in the proceeding section.

4.2.3 User Class

User class is used to store menu item a user would like to order. For now, there is only one important variable to consider:

userItem: an array that stores objects of MenuItems class. This is used to save menu items a user would like to order.

Only one object is instantiated through the life time of this application. This object represents the user cart of menu items he/she would like to order. This object is called upon when a user decides to add an item to their cart. When this occurs, a copy of the MenuItem object is saved within the userItem variables. This way, there is a list maintained to hold all item a user would like to order. When a user confirms and sends their order, the item information is extracted from userItem list, formatted into a JSON request and send back to the database for further processing. Having this class implemented allows extendibility in the future, as all vital information pertaining to a user can be stored within the class (for example, user settings).

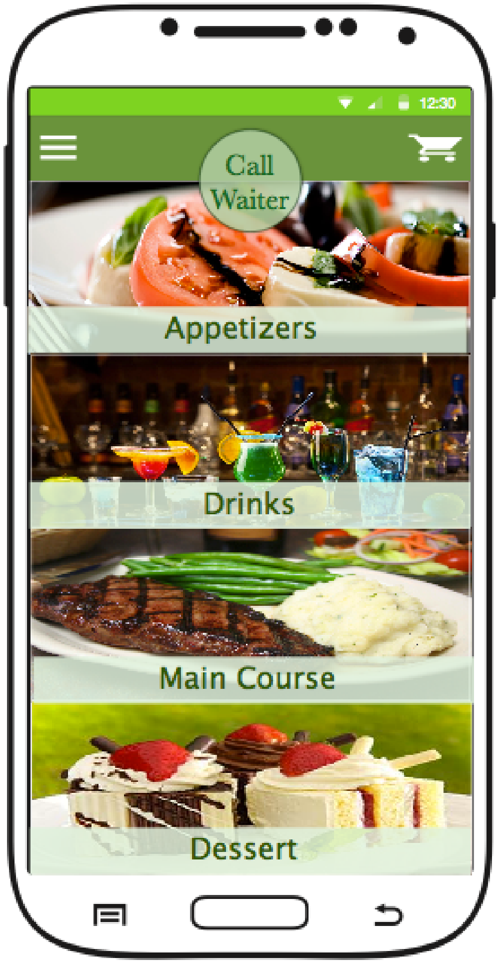
4.3 Accounts Structure

4.4 Transactions Structure

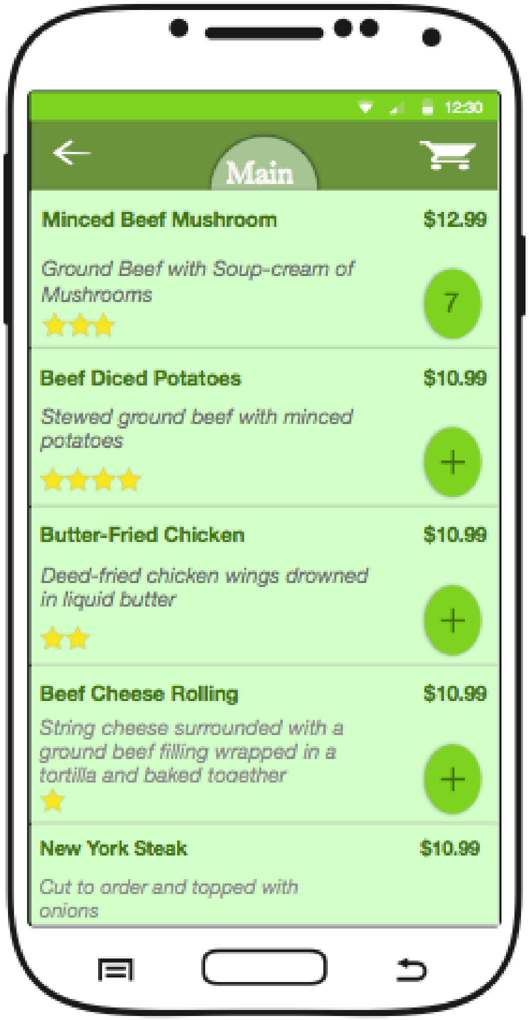
5 User Interface Design

5.1 User Interface Design Overview

5.1.1 Menu Categories



5.1.2 Menu Items



Minced Beef Mushroom $12.99

Ground Beef with Soup-cream of Mushrooms

★★★

Beef Diced Potatoes $10.99

Stewed ground beef with minced potatoes

★★★★

Butter-Fried Chicken $10.99

Deed-fried chicken wings drowned in liquid butter

★★

Beef Cheese Rolling $10.99

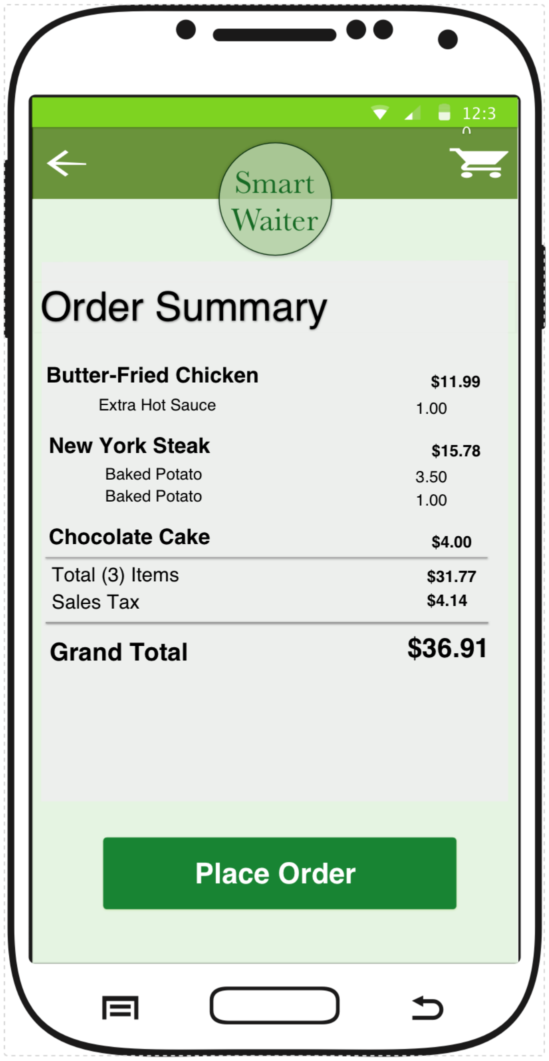
String cheese surrounded with a ground beef filling wrapped in a tortilla and baked together

★

New York Steak $10.99

Cut to order and topped with onions

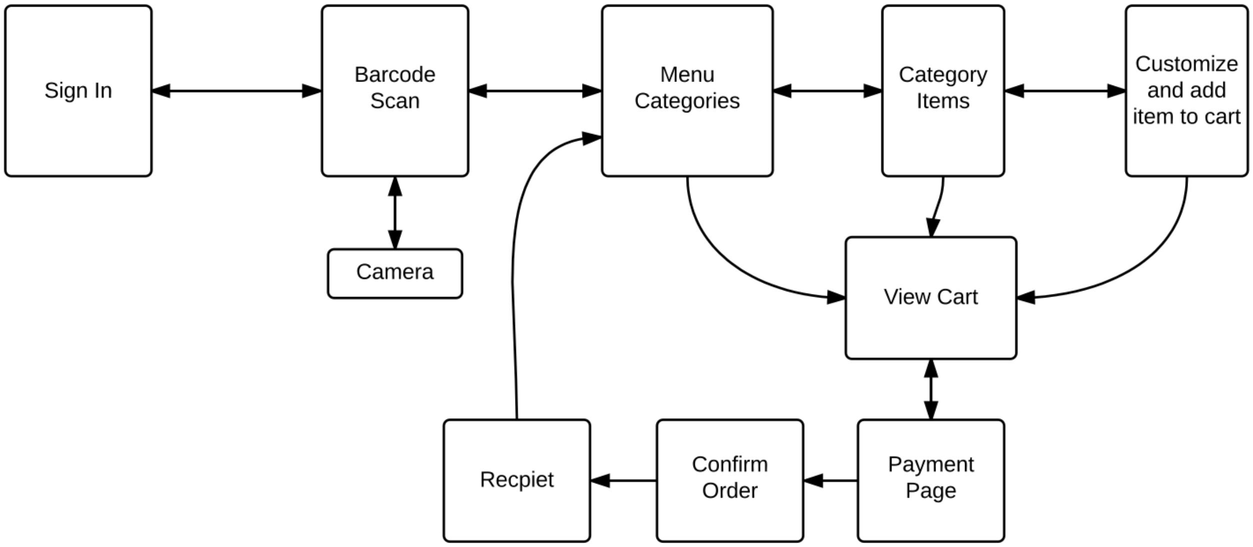
5.1.3 Confirm Order



Order Summary

Butter-Fried Chicken	**$11.99**
Extra Hot Sauce	1.00
New York Steak	**$15.78**
Baked Potato	3.50
Baked Potato	1.00
Chocolate Cake	**$4.00**
Total (3) Items	**$31.77**
Sales Tax	**$4.14**
Grand Total	**$36.91**

5.2 User Interface Navigation Flow



5.3 Use Cases

5.3.1 Sign In Page

User Input	System Response
Enters correct user name and password	Application transitions to Barcode Scan page
Enters incorrect user name and password	Toaster displayed reading ”incorrect login, please try again”
Enters incorrect user name	Toaster displayed reading ”incorrect login, please try again”
Enters incorrect password	Toaster displayed reading ”incorrect login, please try again”
Clicks ”Skip Sign in”	Application transitions to Barcode Scan page
Clicks Back Button on phone	Application Quits

5.3.2 Barcode Scan Page

User Input	System Response
Clicks Scan Barcode	Application transitions to camera so user can scan code. If successful, application will transition to menu page. Otherwise will return to scanning page and display a toaster reading, ”please try again”
Clicks back button on phone	Application transitions to Sign in Page

5.3.3 Menu Categories Page

User Input	System Response
Clicks category	Application transitions Category Items
Clicks back button on phone	Application transitions to Barcode Scan

5.3.4 Category Items Page

User Input	System Response
Clicks Item	Application transitions to Customize Item
Clicks back button on phone	Application transitions to Menu Categories

5.3.5 Customize Item Page

User Input	System Response
Ticks check boxes	None
Enters special instructions in input field	None
Clicks ”Add to Cart”	Transitions to cart page and populate list with item
Clicks back button on phone	Application transitions to Menu items

5.3.6 Cart Page

User Input	System Response
Clicks ”Delete”	Deletes item from list
Clicks ”Submit Order”	Transitions to payment page
Clicks back button on phone	Application transitions to previous page

5.3.7 Payment Page

User Input	System Response
Input valid credit card and clicks ”Process”	Transitions into Confirm Order page
Input invalid credit card and clicks ”Process”	Toaster displayed reading ”invalid credit card”
Clicks back button on phone	Application transitions to previous Cart Page