

**COT 6931 Client Side Design Document**



Food Giant Sales Flyer Generator



# Revision History

|  |  |  |
| --- | --- | --- |
| Revision # | Date | Changed Items |
| 1 | 02/26/2017 | Initial Document Creation |
| 2 | 03/04/2017 | Added Information on Design Strategies |
|  |  |  |

# Section 1 Introduction

This class is designed to take in user input and send to web services

## Section 1.1 Project Purpose

The Food Giant Flyer Creation program is designed to allow Food Giant store managers more control over selling their products. This program will contain an easy to use interface that allows them to create custom flyers that contain items and pricing of their choosing, after approved by district managers.

## Section 1.2 Client Side Summary

The Client Side section of the Food Giant program will contain the information needed for the manager to log into the Web Service to start the flyer creation process. This program will take in a user name and password and output that data to the server to get their user permissions.

## Section 1.3 Requirements Satisfied

The Client Side Program applies to and satisfies the following requirements:

* SR 1.1
* SR 1.1.1
* SR 1.3.1

# Section 2 Client Side Design Structure

## Section 2.1 Design Overview

The overall program will follow a Model View ViewModel (MVVM) design. This means that each visual element (View) will contain as little code as possible, except for the visual components. The ViewModel classes will primarily drive the logic in the code and instantiate the View classes for the user to see. Any events take by the user (button click, item select, etc.) will inform the View Model of the action and allow it to handle the logic behind what to do on each of these actions.

The Model class will be a “.cs” file that will contain data on what needs to be entered to fulfill a criterion for an object.

To accomplish this effectively, we will be using the Caliburn Micro framework, a completely free Application Program Interface (API) that allows us to very quickly and effectively bind ViewModels, Objects, and Events together with very little redundant code. To use Caliburn, we will enforce class naming for all classes created in the Flyer Generator program. This naming convention is defined as:

1. View – *ClassName*View
2. ViewModel - *ClassName*ViewModel
3. Model - *ClassName*Model

This allows a new team to quickly identify what the class’s overall purpose is just from its name, making maintenance easier.

Variable naming conventions will be enforced and will be defined in a short “Coding Standards” document. This is again to enforce a consistent design and assist with maintainability.

## Section 2.1 Client Side Classes

The Client Side will consist of the follow projects:

LoginControl project

The LoginControl will consist of the following classes:

LoginControlView

LoginControlViewModels

## Section 2.1 FlyerView

The FlyerView will contain all the Visible Components.

The FlyerViewModel will contain all the actions and logic and will reference the FlyerView.

# Section 3 Detailed Design

# Terms of Reference

|  |  |
| --- | --- |
| Term | Definition |
| ASP.NET | Active Server Pages |
| GUI | Graphical User Interface |
| MVVM | Model View View-Model |
| QA | Quality Assurance |
| SDD | Software Design Description |
| SPMP | Software Project Management Plan |
| SQL | Structured Query Language |
| SRS | Software Requirements Specifications |
| STD | Software Test Document |