

**COT 6931 Server Side Design Document**



Food Giant Sales Flyer Generator



Table of Contents

[Revision History 2](#_Toc476470252)

[Section 1 Introduction 3](#_Toc476470253)

[Section 1.1 Project Purpose 3](#_Toc476470254)

[Section 1.2 Server Side Summary 3](#_Toc476470255)

[Section 1.3 Requirements Satisfied 3](#_Toc476470256)

[Section 2 Server Side Design Structure 3](#_Toc476470257)

[Section 2.1 Design Overview 3](#_Toc476470258)

[Section 2.1 Server Side Classes 4](#_Toc476470259)

[Section 2. Additional Components 5](#_Toc476470260)

[Section 3 Detailed Design 5](#_Toc476470261)

[Section 3.1 Flyer Creator Classes 5](#_Toc476470262)

[Overview of Flyer Creator Class 6](#_Toc476470263)

[View 6](#_Toc476470264)

[ViewModel 6](#_Toc476470265)

[Section 3.2 Created Flyer Page Classes 6](#_Toc476470266)

[View 6](#_Toc476470267)

[ViewModel 6](#_Toc476470268)

[Section 3.3 Database Structure 6](#_Toc476470269)

[Section 3.4 Database Maintainer Classes 7](#_Toc476470270)

[Overview of Database Maintainer Class 7](#_Toc476470271)

[View 7](#_Toc476470272)

[ViewModel 8](#_Toc476470273)

[Terms of Reference 8](#_Toc476470274)

# 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 retrieve data from a database and show these selections to a user in order to populate a page with images of the selected items. Also, this will allow a user to set a price for each item.

## 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 Server Side Summary

These Server Side Projects are designed to retrieve data from a database and show these selections to a user in order to populate a page with images of the selected items. Also, this will allow a user to set a price for each item.

## Section 1.3 Requirements Satisfied

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

* SR 1.2
* SR 1.2.1
* SR 1.3
* SR 1.3.1
* SR 1.3.2
* SR 1.4
* SR 1.4.1
* SR 1.4.2
* SR 2.1
* SR 2.2
* SR 2.3
* SR 2.4
* SR 3.1
* SR 3.2

# Section 2 Server 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 Server Side Classes

The Server Side will consist of a single project, FoodGiantFlyer. The FoodGiantFlyer project will consist of the following classes:

Views:

* FlyerCreatorView
* DatabaseMaintainerView

ViewModels:

* FlyerCreatorViewModel
* DatabaseMaintainerViewModel

Models:

* FlyerDataModel

Others:

* AppBootstrapper
* DatabaseInterface

## Section 2. Additional Components

In addition to the classes listed in the previous section, the Server Side program will contain the following components:

* Database – FoodGiantItemSQLDatabase.mdf. This database will contain all data for the Food Giant Items entered in by an admin. This is an integral part of the FlyerCreator classes.
* Images Folder – Located In subfolder under exe folder. This folder contains all added images by an admin. This is an integral part of the FlyerCreator classes.
* ASP.NET Folder – Located In subfolder under exe folder. This folder contains all Food Giant templates added by an admin. This is an integral part of the FlyerCreator classes.

These components will be discussed further in the detailed design section where applicable to the classes that require their usage.

# Section 3 Detailed Design

This section will detail the purpose of each class, the SRS requirements it affects and a description of the classes’ components.

## Section 3.1 Flyer Creator Classes

The Flyer Creator class handles displaying the data from the database to the user, and allowing them to select all the elements involved in generating a flyer.

After the manager has selected the items they want to use for the flyer, this class will pass the selected items to the asp.net page to generate a printable flyer.

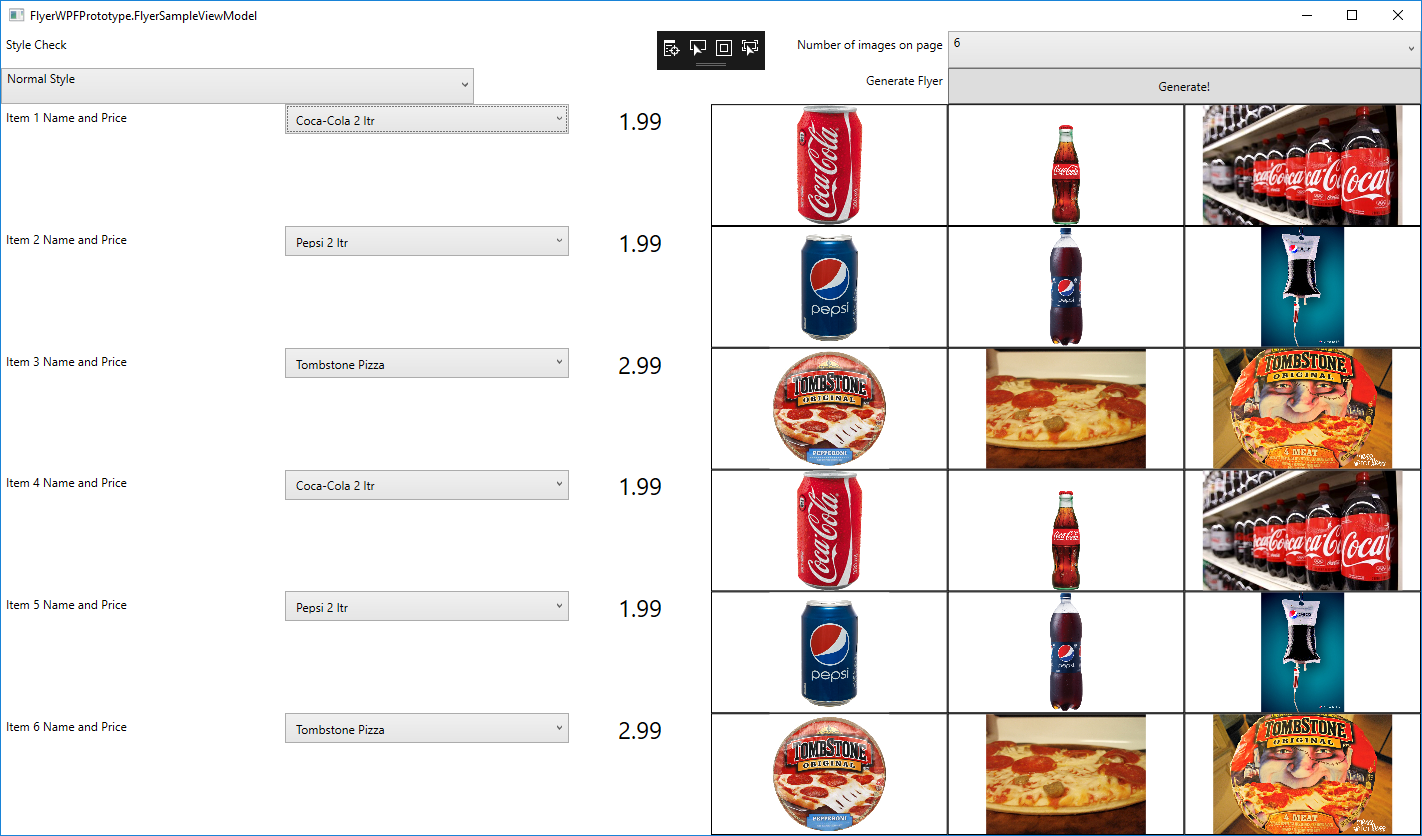


Image 1.1 Flyer Creator Prototype Picture

### Section 3.1.1 Overview of Flyer Creator Class

These classes satisfy requirements

* SR 1.2
* SR 1.2.1
* SR 1.4
* SR 1.4.1
* SR 1.4.2
* SR 3.1
* SR 3.2

### Section 3.1.2 View

The FlyerCreator will contain all the Visible Components that the user can interact with.

### Section 3.1.3 ViewModel

The FlyerCreatorViewModel will be the primary C# class used for the Food Giant Program

The FlyerCreatorViewModel will contain all the actions and logic and will reference the FlyerCreatorView. This class will populate data from the SQL database into the View for the user to see and select.

## Section 3.2 Created Flyer Page Classes

These classes satisfy requirements:

* SR 1.3
* SR 1.3.1
* SR 1.3.2

### Section 3.2.1 Overview of Created Flyer Page Class

### Section 3.2.2 View

### Section 3.2.3 ViewModel

## Section 3.3 Database Structure

The database satisfies requirements:

* SR 2.1
* SR 2.2

The SQL Database will contain the following tables:

* FlyerHistory
* ItemList

TODO This section

The FlyerHistory Table’s purpose is to

The FlyerHistory Table will contain the following fields:

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Allow Nulls |
| ID (Primary Key, Unique) | Int | No |
| Item Name (Unique) | Text | No |

The ItemList Table’s purpose is to

The ItemList Table will contain the following fields:

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Allow Nulls |
| ID (Primary Key, Unique) | int | No |
| Item Name (Unique) | Text | No |
| Item Category | Varchar(50) | No |
| Image Name 1 | Varchar(50) | No |
| Image Name 2 | Varchar(50) | Yes |
| Image Name 3 | Varchar(50) | Yes |

Int ID – Unique Identifier for each database item

Test Item Name – Name of sale item

store the data for the item name and

## Section 3.4 Database Maintainer Classes

### Section 3.4.1 Overview of Database Maintainer Class

These classes satisfy requirements

* SR 2.3
* SR 2.4

### Section 3.4.2 View

### Section 3.4.3 ViewModel

# Section 4 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 |