**Object-Oriented Design II**

SSE 554

Dr. MacNeil

Spring 2017

**Project II**

Brent Bitler

Matthew Robison

Table of Contents

[Table of Contents 2](#_Toc475639866)

[Table of Figures 3](#_Toc475639867)

[Introduction 4](#_Toc475639868)

[Purpose 4](#_Toc475639869)

[ASP.net MVC Breakdown 5](#_Toc475639870)

[Overview 5](#_Toc475639871)

[Topic 2 6](#_Toc475639872)

[Overview 6](#_Toc475639873)

[Topic 3 7](#_Toc475639874)

[Overview 7](#_Toc475639875)

[Topic 4 8](#_Toc475639876)

[Overview 8](#_Toc475639877)

[Appendix 9](#_Toc475639878)

[Bibliography 9](#_Toc475639879)

[Activity Log 10](#_Toc475639880)

Table of Figures

Figure 1. MVC desgign flow 5

Introduction

## Purpose

For Project II, the purpose was to choose a topic new to us from the text, demonstrate our capabilities through software design, coding, and finally unit testing while using some form of source control. For Project II, we have decided to investigate ASP.net MVC, its functionality, and implementation by creating a sample webpage. This report will chronicle our design strategy and timeline. The features of the project requirements are outlined in the table below.

Table . Project II Composition

|  |  |
| --- | --- |
| Team Composition | Brent Bitler, Matthew Robison |
| Topic | ASP.net MVC |
| Distributed Version Control System | GitHub |
| Programming Language | C# |
| IDE | Visual Studio 2015 |
| TDD Tool | Built-in VS unit test tool |

ASP.net MVC Breakdown

## Overview

ASP.net MVC is a framework designed by Microsoft to facilitate website development, utilizing MVC (Model View Controller) methodologies. The benefit here lies in keeping tasks decoupled and provides a clean separation between the three ideologies. The design pattern of MVC is illustrated in figure 1. It is not a standalone venture, but rather was built on top of ASP.net so developers like us are still able to utilize pre-existing ASP.net functionality.

MODEL

CONTROLLER

VIEW

Figure . MVC desgign flow

The model is the collection of classes in control of the data aspect of the program. The view, as its name hints towards, helps manage the user experience through the UI (User Interface) and is in HTML (Hypertext Mark-up Language). The controller is the collection of classes that deals with user interaction and application conditionals and logic. The rest of this report will chronicle the implementation of an ASP.net MVC application from start to finish.

Application Creation

## Overview

The first step of the application process is to utilize the Visual Studio IDE to begin creating our webpage. We began by selecting an empty ASP.net 4.5.2 template and addings references for MVC. This will provide us with a blank slate for development with minimal predefined content. Here is the starting point for our program after a quick launch. Obviously, we expect a connection error as we have no defined any code thus far.

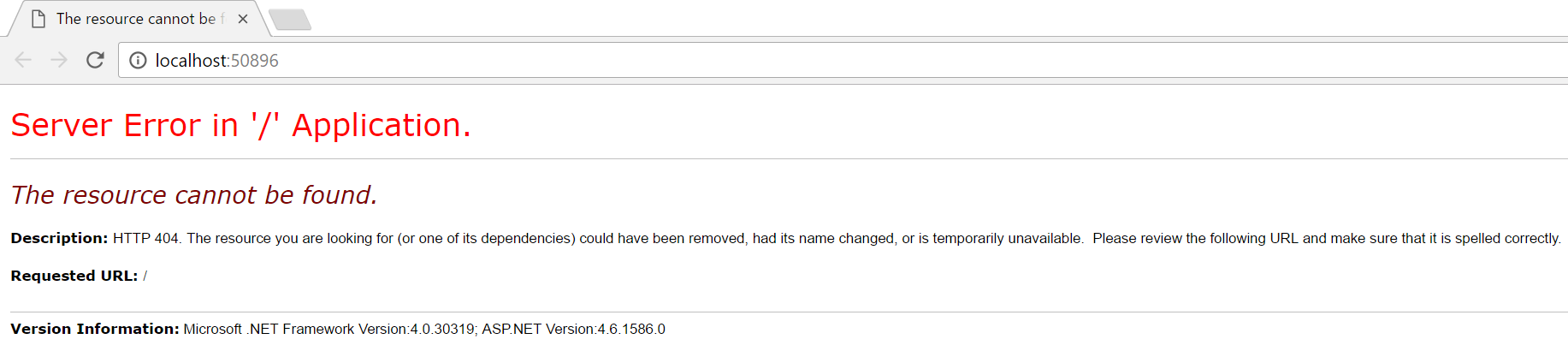


Figure . Initial debug with no code implemented

This will change shortly after adding a controller to the application as shown in figure 3 with a string test output.

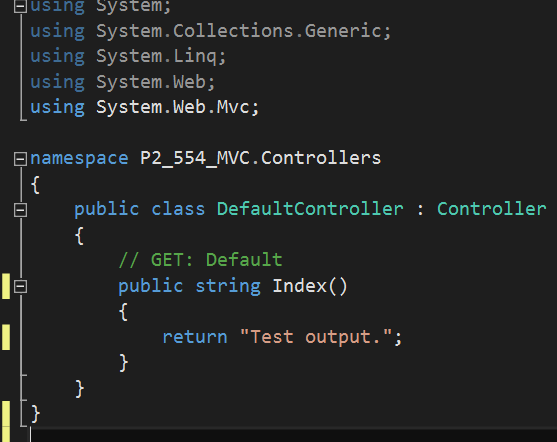


Figure . Initial default controller added with test output text

Now if we run the application we have an established connection with the resulting output shown in figure 4.

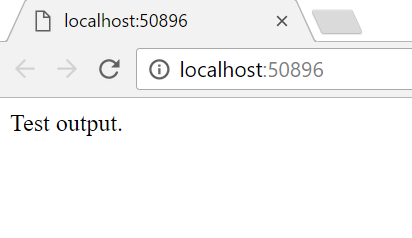


Figure . 404 error resolved

App & Request Life Cycle

## Overview

A life cycle, in general, is a grouping of steps are tasks utilized to handle some sort of request to change the state of an application. The input is a request while the output is a response. In general, MVC consists of two life cycles:

* Application life cycle
* Request life cycle

## Application Life Cycle

The application life cycle is simple the time the application process begins running until the time it stops. Within this timeframe, numerous request life cycles are able to take place to facilitate the application life cycle.

## Request Life Cycle

The other half of this spectrum is the request life cycle, which is the series of events that occurs when an HTTP request is processed by the application. The request is then routed thorugh the URL Routing Module, which is a .NET component able to latch into the application. From here, the MVC framework converts the routing data into a controller able to process requests. At this point, an action invoker searches for and chooses the correct Action method to invoke the controller.

This can be seen as the rebound point for the flow diagram seen in figure 5, which fires off Result Execution. Depending on the type of output it will either be a View engine and subsequently render the view, otherwise the action result will execute on its own. This is what capitulates a real response to the initial HTTP request.

Routing

View Engine

Result Execution

Controller Initialization

Action Execution

Result Execution

RESP

REQ

Figure . MVC Request Life Cycle flow chart

Routing & Controllers

## Routing

This

## Controllers

Appendix

Bibliography

[1] Engelbrecht, Andries P. *Computational Intelligence: An Introduction, (2nd ed.)*. Wiley, 2007.

Activity Log

|  |  |  |
| --- | --- | --- |
| Date | Time (mins) | Description |
| 1/9/17 | 60 | Looked through Naïve GA |
| 1/10/17 | 60 | Looked through and worked Naïve GA |
| 1/11/17 |  |  |
| 1/12/17 | 120 | Research GA’s online for topic ideas |
| 1/13/17 | 180 | Read Ch 9 - GA |
| 1/14/17 | 90 | Read Ch 9 - GA |
| 1/15/17 |  |  |
| 1/16/17 | 120 | Begin constructing class structure of string unscramble program |
| 1/17/17 | 120 | Begin chromosome class |
| 1/18/17 | 120 | Continue working chromosome class, begin developing population class |
| 1/19/17 |  |  |
| 1/20/17 |  |  |
| 1/21/17 | 60 | Finish chromosome class |
| 1/22/17 | 240 | Continue working population class |
| 1/23/17 | 120 | Connector class |
| 1/24/17 | 30 | Come up with initial variable settings |
| 1/25/17 |  |  |
| 1/26/17 | 240 | Begin generating report |
| 1/27/17 | 180 | Put together test data |
| 1/28/17 | 360 | Conclude paper, graphs, etc |
| 1/29/17 | 180 | Final review of paper |
| 1/30/17 | 15 | Submit paper |
|  |  |  |
| TOTAL | 2295 |  |