|  |  |
| --- | --- |
|  | **Cognizant Academy**  **truYum**  **FSE .Net Core Specification Document**  **Version 1.0** |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** | | **Name** | Seshadri M R | Vimalathithan Krishnan | Ramadevanahalli Lingachar, Shashidhara Murthy | | **Role** | Learning Solution Designer | Learning Solution Architect | Learning Solution Lead | | **Signature** |  |  |  | | **Date** |  |  |  | |
|  |

Table of Contents

[1.0 Introduction 3](#_Toc19716868)

[1.1 Purpose of this document 3](#_Toc19716869)

[1.2 Definitions & Acronyms 3](#_Toc19716870)

[1.3 Project Overview 3](#_Toc19716871)

[1.4 Scope 3](#_Toc19716872)

[1.5 Intended Audience 3](#_Toc19716873)

[1.6 Hardware and Software Requirement 3](#_Toc19716874)

[2.0 Forking and Cloning 4](#_Toc19716875)

[3.0 TYUC001, TYUC002 – Active Menu Item List 4](#_Toc19716876)

[4.0 TYUS004 – Search Menu item by name 4](#_Toc19716877)

[5.0 TYUC003 – Edit menu item 5](#_Toc19716878)

[6.0 TYUS001, TYUS002 – Responsive to multiple devices 5](#_Toc19716879)

[7.0 Submission 5](#_Toc19716880)

[7.1 Code submission instructions 5](#_Toc19716881)

[8.0 Change Log 6](#_Toc19716882)

# Introduction

## Purpose of this document

The purpose of this document is to define the user interface specification for truYum project.

## Definitions & Acronyms

|  |  |
| --- | --- |
| Definition / Acronym | Description |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheet |
| RWD | Responsive Web Design |
| UX Design | User Experience Design |

## Project Overview

Refer truyum-user-stores\_DotNetCore.xlsx for understanding the functionality and features.

## Scope

Develop truYum application using .Net core

## Intended Audience

* Product Owner
* Scrum Master
* Application Architect
* Project Manager
* Test Manager
* Development Team
* Testing Team

## Hardware and Software Requirement

1. Hardware Requirement:
   1. Developer Desktop PC with 8GB RAM
2. Software Requirement
   1. Visual Studio 2017
   2. Chrome Browser

# Forking and Cloning

1. **Fork project:** <https://code.cognizant.com/dotnet/truYum-Core>

# TYUC001, TYUC002 – Active Menu Item List

**Create New Controller for Food item operations**

1. Create a new model class with appropriate properties for FoodItem
2. Create a new Controller with Read write actions and NOT scaffolding
3. Hardcode the sample list of food items to be used in the Index action method

**Create New Food Menu View**

1. Add view to the Index Action method
2. Hardcode the sample list of food items in the component
3. Render the food items in the view using a Partial view, sending the model
4. Render the partial view inside a div in Index view

**Create Food Item detail partial View**

1. Create a partial view \_FoodItem
2. Use tag helpers to render the content, using DIV, instead of tabular format
3. Have Edit link for every line item

# TYUS004 – Search Menu item by name

**Modify Index View for Search**

1. Add a textbox with Id for search menu item by the name
2. Use placeholder to specify that it’s a search by menu item name

**Implement search using Ajax call**

1. Use document.ready to link a function to be triggered on keyup for the search textbox
2. Create a HttpGet action method in the controller to perform the search operation
3. Take in the search string as input parameter to search the hardcoded food item list
4. Set this to a PartialView with the filtered list as model
5. If the search string is null or empty, return the entire list
6. In the Ajax success call, set the parent DIV of partial view with the content using the HTML method

# TYUC003 – Edit menu item

**Create Edit Food Item View**

1. Create a Food Item edit view to Edit the Name, Price and IsActive properties of the Food item
2. In the Edit action method, filter the hardcoded list to filter the food menu item as per the Id
3. In the Edit action method of HttpPost, get the data of the modified model and set it in TempData converting it to a JSON string
4. Use RedirectToAction MVC method to Index page.
5. Use ViewBag to get the data from TempData and write it on the web page

# TYUS001, TYUS002 – Responsive to multiple devices

**Configure project to use UI template**

1. Update index.html with the template header and footer
2. Update the components template with appropriate elements from UI template

# Submission

## Code submission instructions

Use git add, commit and push commands to upload your code into remote GltLab repository.

During commit, give the commit message as “DotNet core with MVC”.

# Change Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Changes Made | | | |
| V1.0.0 | Initial baseline created on <dd-Mon-yy> by <Name of Author> | | | |
| Vx.y.z | <Please refer the configuration control tool / change item status form if the details of changes are maintained separately. If not, the template given below needs to be followed> | | | |
| **Section No.** | **Changed By** | **Effective Date** | **Changes Effected** |
|  |  |  |  |