|  |  |
| --- | --- |
|  | **Cognizant Academy**  **truYum**  **ASP.NET Specification Document**  **Version 0.1** |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** | | **Name** | Ramamoorthy Selvam | Vimalathithan Krishnan | Ramadevanahalli Lingachar, Shashidhara Murthy | | **Role** | Learning Solution Designer | Learning Solution Architect | Learning Solution Lead | | **Signature** |  |  |  | | **Date** |  |  |  | |
|  |

Table of Contents

[1.0 Introduction 4](#_Toc11315152)

[1.1 Purpose of this document 4](#_Toc11315153)

[1.2 Definitions & Acronyms 4](#_Toc11315154)

[1.3 Project Overview 4](#_Toc11315155)

[1.4 Scope 4](#_Toc11315156)

[1.5 Intended Audience 4](#_Toc11315157)

[1.6 Hardware and Software Requirment 4](#_Toc11315158)

[2.0 Design for View Menu Item List Admin (TYUC001) 6](#_Toc11315159)

[2.1 Sequence Diagram 6](#_Toc11315160)

[2.2 ASPX 6](#_Toc11315161)

[3.0 Design for View Menu Item List Customer (TYUC002) 7](#_Toc11315162)

[3.1 Sequence Diagram 8](#_Toc11315163)

[3.2 ASPX 8](#_Toc11315164)

[4.0 Design for Edit Menu Item (TYUC003) 9](#_Toc11315165)

[4.1 Sequence Diagram (Show Edit Menu Item Form) 9](#_Toc11315166)

[4.2 Sequence Diagram (Edit Menu Item) 10](#_Toc11315167)

[4.3 ASPX (EditMenuItem.aspx) 10](#_Toc11315168)

[4.1 ASPX (EditMenuItemStatus.aspx) 11](#_Toc11315169)

[5.0 Design for Add Cart (TYUC004) 12](#_Toc11315170)

[5.1 Sequence Diagram 12](#_Toc11315171)

[5.1 ASPX (MenuItemListCustomer.aspx) 13](#_Toc11315172)

[6.0 Design for View Cart (TYUC005) 14](#_Toc11315173)

[6.1 Sequence Diagram 14](#_Toc11315174)

[6.2 ASPX (Cart.aspx) 15](#_Toc11315175)

[6.1 ASPX (CartEmpty.aspx) 15](#_Toc11315176)

[7.0 Design for Remove Item from Cart (TYUC006) 17](#_Toc11315177)

[7.1 Sequence Diagram 17](#_Toc11315178)

[7.2 ASPX (Cart.ASPX) 18](#_Toc11315179)

[8.0 Standards and Guidelines 18](#_Toc11315180)

[8.1 ASPX 18](#_Toc11315181)

[9.0 Submission 19](#_Toc11315182)

[9.1 Code submission instructions 19](#_Toc11315183)

[10.0 Evaluation Criteria 19](#_Toc11315184)

[11.0 Change Log 20](#_Toc11315185)

# Introduction

## Purpose of this document

The purpose of this document is to define the server side implementation of the truYum application.

## Definitions & Acronyms

|  |  |
| --- | --- |
| Definition / Acronym | Description |
| ASP.NET | Active Server Pages using .NET framework |

## Project Overview

Refer Use Case specification document for understanding the functionality and features.



## Scope

1. Creation of ASP.Net webforms for truYum application

## Intended Audience

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

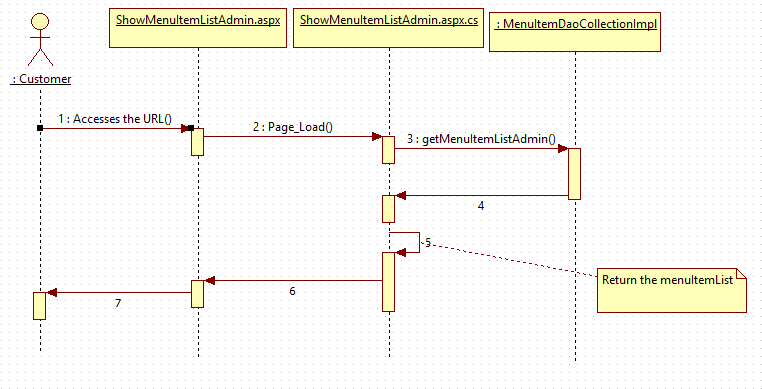
## Hardware and Software Requirment

1. Hardware Requirement:
   1. Developer PC with 4GB Ram
2. Software Requirement
   1. Git
   2. IE or Chrome
   3. .Net Framework 4.5
   4. Visual Studio Professional Edition 2015
   5. SQL Server enterprise edition 2014

# Design for View Menu Item List Admin (TYUC001)

## Sequence Diagram

The following diagram explains the flow of execution when admin accesses the specific URL. Refer steps included after the sequence diagram.



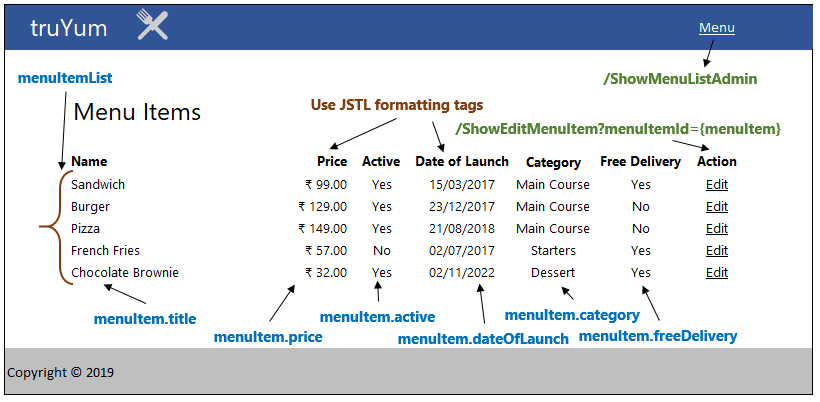
**Steps Explanation**

1. Customer access the ShowMenuItemListAdmin.aspx
2. Instantiate MenuItemDaoCollectionImpl class in Page\_Load of ShowMenuItemListAdmin.aspx.cs and assign it to MenuItemDao reference variable menuItemDao.
3. Invoke menuItemDao.getMenuItemListAdmin() and get the list of menu items.
4. The getMenuItemListAdmin which holds the menuitems list for admin will be bound to gridview control ( available in ShowMenuItemListAdmin.aspx ) using datasource and databind methods.

## ASPX

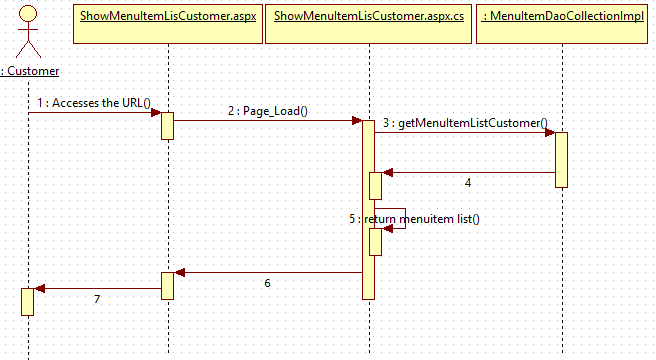
List the menu items for admin.

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed aspx** | **Description** |
| truYum/MenuItemListAdminitem.html | WebContent/ MenuItemListAdminitem.aspx | List the menu items for admin |



# Design for View Menu Item List Customer (TYUC002)

## Sequence Diagram



**Steps Explanation**

1. Customer access the ShowMenuItemListCustomer.aspx
2. Instantiate MenuItemDaoCollectionImpl class in Page\_Load of ShowMenuItemListCustomer.aspx.cs and assign it to MenuItemDao reference variable menuItemDao.
3. Invoke menuItemDao.getMenuItemListCustomer() and get the list of menu items.
4. The getMenuItemListCustomer which holds the menuitems list for customer will be bound to gridview control ( available in ShowMenuItemListCustomer.aspx ) using datasource and databind methods.

## ASPX

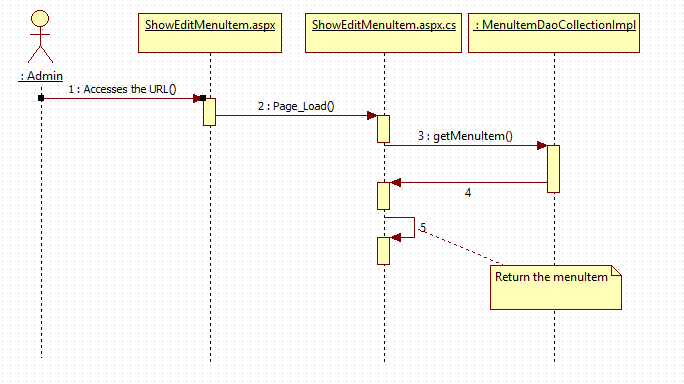
List the menu items for customer.

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed ASPX** | **Description** |
| truYum/MenuItemListCustomer.html | truYum/MenuItemListCustomer.aspx | List the menu items for customer |



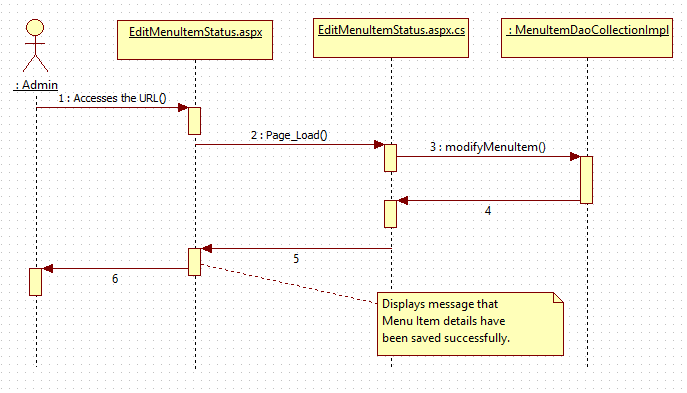
# Design for Edit Menu Item (TYUC003)

## Sequence Diagram (Show Edit Menu Item Form)

**Steps Explanation**

1. Customer clicks on Edit link on a menu item.
2. Customer access the ShowEditMenuItem.aspx
3. The request is receieved in the Page\_load method of ShowEditMenuItem.aspx.cs.
4. Instantiate MenuItemDaoCollectionImpl class in Page\_Load of ShowEditMenuItem.aspx.cs and assign it to MenuItemDao reference variable menuItemDao.
5. The menuItemID that is edited by the user is passed as a input parameter to getMenuItem() method. User is naviaged to ShowEditMenuItem.aspx page which populates the menuitem details in the respective form fields for the passed menuItemId.

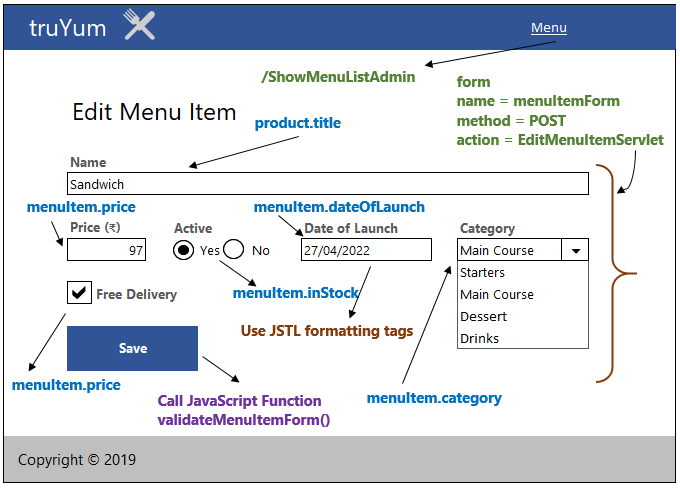
## Sequence Diagram (Edit Menu Item)



Explanation steps are not included for the above diagram as well as subsequent sequence diagrams. Earlier explanation steps can be referred to arrive at the understanding.

## ASPX (EditMenuItem.aspx)

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed ASPX** | **Description** |
| truYum/EditMenuItem.html | truYum/EditMenuItem.aspx | Page with form to edit a menu item |

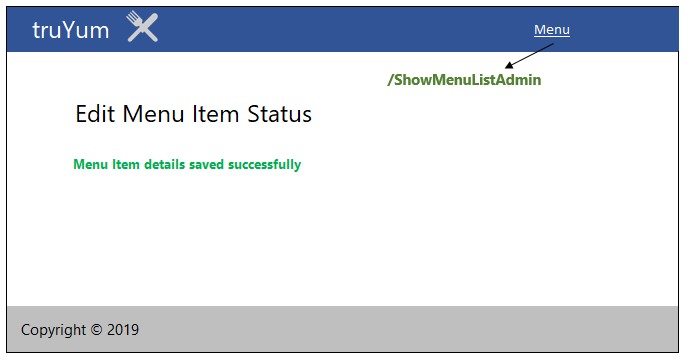


**Category Dropdown Handling**

1. Set the value from menuItem.category as the first item in the drop down.
2. Let the other items be added directly in the aspx with hard coded values.

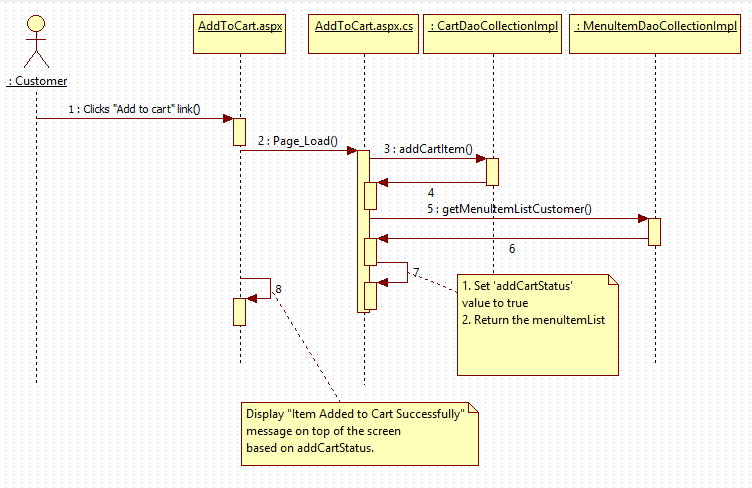
## ASPX (EditMenuItemStatus.aspx)

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed ASPX** | **Description** |
| truYum/EditMenuItemStatus.html | truYum/EditMenuItemStatus.aspx | Display the status of modifying menu item. |



# Design for Add Cart (TYUC004)

## Sequence Diagram



**Steps Explanation**

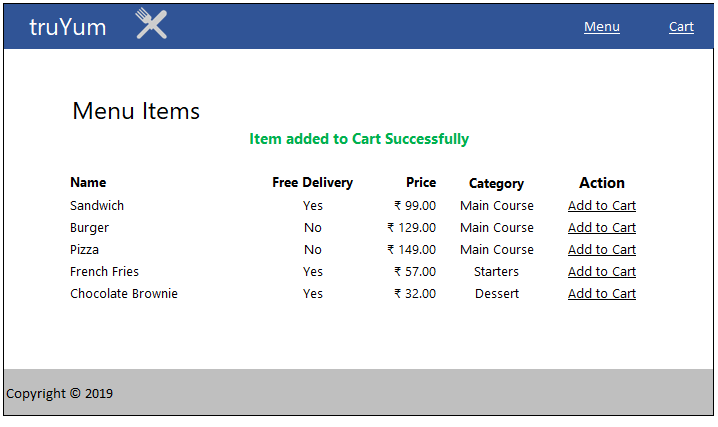
1. Customer clicks on “Add to Cart” link on a menu item from Menu Item List.
2. Customer is directed to AddToCart.aspx?menuItemId=1 URL
3. Set variable userId of type long to value 1. Identifier for a specific user.
4. Read the menuItemId from the request and assign it to a variable
5. Instantiate CartDaoCollectionImpl and assign it to CartDao.
6. Invoke the addCartItem() method passing the userId and menuItemId.
7. Instantiate MenuItemDaoCollectionImpl class in Page\_Load of AddToCart.aspx.cs and assign it to MenuItemDao reference variable menuItemDao.
8. List of MenuItem returned is stored in a variable is bound to gridview control

( available in AddToCart.aspx ) using datasource and databind methods

1. The addCartStatus with value true is stored in a variable and displayed on the page.

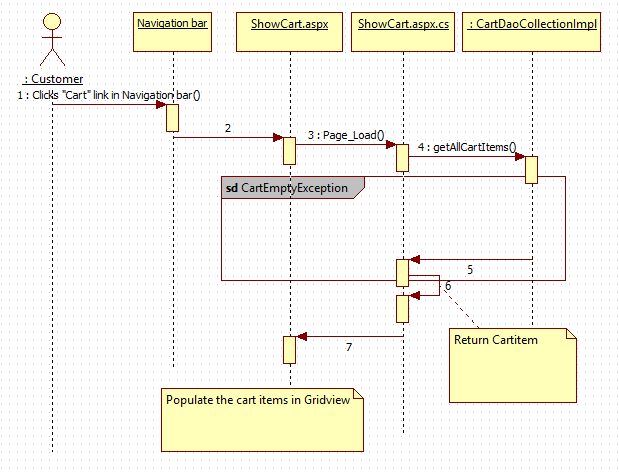
## ASPX (MenuItemListCustomer.aspx)

This aspx is the same the file defined in View Menu Item List Customer section.



# Design for View Cart (TYUC005)

## Sequence Diagram



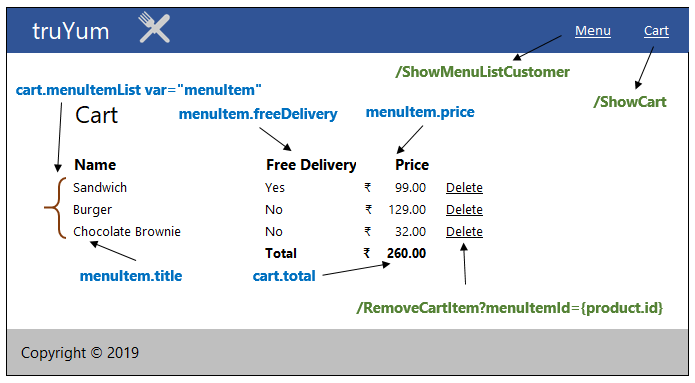
**NOTE:** If getAllCartItems() method throws an exception, then the exception is passed on to CartEmpty.aspx page.

**Steps Explanation**

1. Customer clicks on “Cart” link on the navigation bar.
2. Customer access the /ShowCart.aspx
3. The request is received in Page\_Load method of ShowCart.aspx
4. Set variable userId with type long having value 1
5. Instantiate CartDaoCollectionImpl and assign it to CartDao.
6. Invoke the getAllCartItems() method passing the userId.
7. If the getAllCartItems() does not throw CartEmptyException:
8. List of MenuItem returned is bound to gridview control ( available in ShowCart.aspx ) using datasource and databind methods
   1. Invoke the getAllCartItems() method passing the userId.
   2. If the getAllCartItems() does not throw CartEmptyException:
9. If the getAllCartItems() throws CartEmptyException:
   1. Catch CatchEmptyException and redirect the user to CartEmpty.aspx

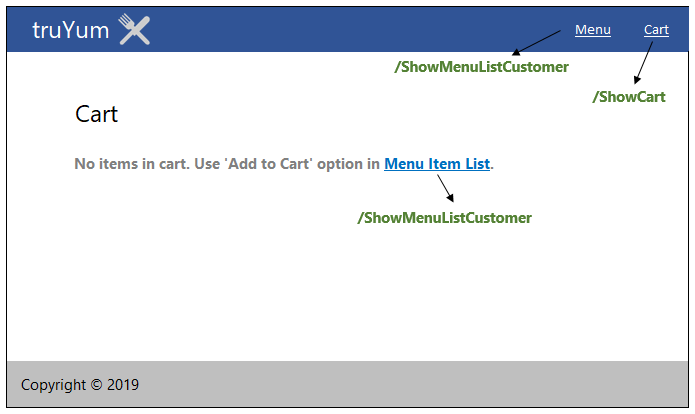
## ASPX (Cart.aspx)

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed ASPX** | **Description** |
| truYum/Cart.html | truYum/Cart.aspx | Displays the cart with the list of menu items added by the customer |



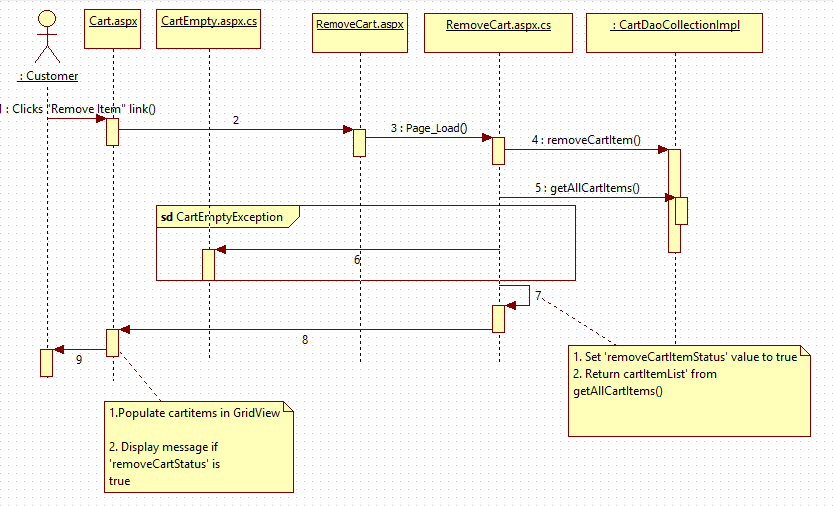
## ASPX (CartEmpty.aspx)

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed ASPX** | **Description** |
| truYum/CartEmpty.html | /CartEmpty.aspx | Displays the cart empty message |



# Design for Remove Item from Cart (TYUC006)

## Sequence Diagram



**Steps Explanation**

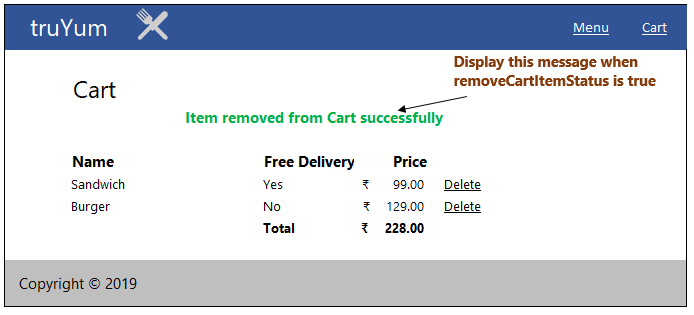
1. Customer clicks on “Remove” link on menu item in Cart.aspx.
2. Customer access the /RemoveCart.aspx?menuItemId=1 URL.
3. The request is received in Page\_Load() method
4. Read the menuItemId from the request and assign it to a variable
5. Instantiate CartDaoCollectionImpl and assign it to CartDao.
6. Set variable userId of type long having value 1.
7. Invoke removeCartItem() method passing the userId and menuItemId.
8. Set the request with variable removeCartItemStatus having value true.
9. Invoke the getAllCartItems() method passing the userId.
10. If the getAllCartItems() does not throw CartEmptyException:
11. List of MenuItem returned is bound to gridview control ( available in RemoveCart.aspx ) using datasource and databind methods

Eg: < To be updated>

* 1. Invoke the getAllCartItems() method passing the userId.
  2. If the getAllCartItems() does not throw CartEmptyException:

1. If the getAllCartItems() throws CartEmptyException:
   1. Catch CatchEmptyException and redirect the user to CartEmpty.aspx

## ASPX (Cart.ASPX)



For other field values refer MenuItemListCustomer.aspx specification in View Menu Item List Customer.

# Standards and Guidelines

## ASPX

1. Every server control should have meaningful Id
2. Server control naming convention to be followed in Id, to explain what type of control is used. Eg: txtUserName
3. Startup page should be set for the web application
4. Startup page should be set for the web application
5. No hard coded values in aspx/ascx files. To be referenced from Web.config file
6. Database connection string should be set in the ConnectionStrings section of Web.config and NOT in the AppSettings

# Submission

## Code submission instructions

Once your code is evaluated by the trainer and all the issues reported by the trainer are corrected, the code needs to be submitted to the remote repository. Follow the steps below to submit the code to remote repository.

1. In Windows Explorer go to the truYum folder
2. Right click on the empty space in the right hand side of Windows Explorer and select “Git Bash here”
3. Execute the following commands

To display the added or modified files

git status

To stage the added or modified files

git add .

To display the staged files

git status

To save the code to local repository

git commit -m "Servlet & JSP"

To transfer the changes from local machine to server

git push origin master

1. Successful execution of the above commands will upload the files to the server repository.
2. Login into <https://code.cognizant.com>
3. Click on the project truYum
4. Check if the files that are uploaded correctly with appropriate folder structure.

# Change Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Changes Made | | | |
| V1.0.0 | Initial baseline created on 20-May-19 by Ramamoorthy Selvam | | | |
| 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** |
|  |  |  |  |