|  |  |
| --- | --- |
|  | **Cognizant Academy**  **truYum**  **Servlet & JSP Specification Document**  **Version 1.0** |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** | | **Name** | Chandrasekaran Janardhanan | Vimalathithan Krishnan | Ramadevanahalli Lingachar, Shashidhara Murthy | | **Role** | Learning Solution Designer | Learning Solution Architect | Learning Solution Lead | | **Signature** |  |  |  | | **Date** | 22 May 2019 | 22 May 2019 | 17 Jun 2019 | |
|  |

Table of Contents

[1.0 Introduction 4](#_Toc12281353)

[1.1 Purpose of this document 4](#_Toc12281354)

[1.2 Definitions & Acronyms 4](#_Toc12281355)

[1.3 Project Overview 4](#_Toc12281356)

[1.4 Scope 4](#_Toc12281357)

[1.5 Intended Audience 4](#_Toc12281358)

[1.6 Hardware and Software Requirment 4](#_Toc12281359)

[1.7 Eclipse Configuration for Tomcat Server 5](#_Toc12281360)

[1.7.1 Add Tomcat Server to Eclipse 5](#_Toc12281361)

[1.7.2 Add Tomcat library to eKart project 5](#_Toc12281362)

[2.0 Class Diagram 7](#_Toc12281363)

[2.1 Servlets 7](#_Toc12281364)

[3.0 Design for View Menu Item List Admin (TYUC001) 8](#_Toc12281365)

[3.1 Sequence Diagram 8](#_Toc12281366)

[3.2 Servlet 8](#_Toc12281367)

[3.3 JSP 9](#_Toc12281368)

[4.0 Design for View Menu Item List Customer (TYUC002) 10](#_Toc12281369)

[4.1 Sequence Diagram 10](#_Toc12281370)

[4.2 Servlet 10](#_Toc12281371)

[4.3 JSP 11](#_Toc12281372)

[5.0 Design for Edit Menu Item (TYUC003) 11](#_Toc12281373)

[5.1 Sequence Diagram (Show Edit Menu Item Form) 11](#_Toc12281374)

[5.2 Sequence Diagram (Edit Menu Item) 12](#_Toc12281375)

[5.3 Servlet 13](#_Toc12281376)

[5.4 JSP (edit-menu-item.jsp) 13](#_Toc12281377)

[5.1 JSP (edit-menu-item-status.jsp) 14](#_Toc12281378)

[6.0 Design for Add Cart (TYUC004) 15](#_Toc12281379)

[6.1 Sequence Diagram 15](#_Toc12281380)

[6.1 Servlet 16](#_Toc12281381)

[6.1 JSP (menu-item-list-customer.jsp) 16](#_Toc12281382)

[7.0 Design for View Cart (TYUC005) 17](#_Toc12281383)

[7.1 Sequence Diagram 17](#_Toc12281384)

[7.2 Servlet 18](#_Toc12281385)

[7.3 JSP (cart.jsp) 18](#_Toc12281386)

[7.1 JSP (cart-empty.jsp) 18](#_Toc12281387)

[8.0 Design for Remove Item from Cart (TYUC006) 20](#_Toc12281388)

[8.1 Sequence Diagram 20](#_Toc12281389)

[8.2 Servlet 21](#_Toc12281390)

[8.3 JSP (cart.jsp) 21](#_Toc12281391)

[9.0 Standards and Guidelines 21](#_Toc12281392)

[9.1 JSP 21](#_Toc12281393)

[9.2 Servlets 22](#_Toc12281394)

[10.0 Submission 22](#_Toc12281395)

[10.1 Code submission instructions 22](#_Toc12281396)

[11.0 Change Log 23](#_Toc12281397)

# 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 |
| JSP | Java Server Pages |
| JSTL | Java Standard Tag Library |

## Project Overview

Refer truYum-use-case-specification.docx for understanding the functionality and features.

## Scope

1. Creation of JSP and Servlets 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. JDK 1.8
   3. Eclipse IDE for Enterprise Java Developers 2019-03 R
   4. Apache Tomcat 9

## Eclipse Configuration for Tomcat Server

### Add Tomcat Server to Eclipse

In this module the eKart application will be deployed and tested in Tomcat Server. Tomcat is integrated with Eclipse and the following steps will guide for configuring Tomcat Server in Eclipse.

1. Extract Tomcat 9 zip file to D:
2. Open Eclipse
3. Select Window > Show View > Servers
4. Click on the link “No servers are available. Click this link to create a new server..”
5. Expand “Apache” and select “Tomcat v9.0 Server”
6. Click “Next”
7. Click “Browse” and select the folder where Tomcat 9 had been extracted
8. Click “Next” and then click “Finish”
9. Now the Tomcat Server will be visible in the “Servers” view.

### Add Tomcat library to eKart project

Once the Tomcat Server is configure in Eclipse, we need to ensure that the Tomcat library is available to truYum project. Follow the steps below to get this done.

1. Right click on truYum project and select the options below
2. Properties > Build Path > Libraries > Add Library .. > Server Runtime > Next > Apache Tomcat v9.0
3. Click “Apply & Close”

# Class Diagram

## Servlets

Refer the diagram below and create classes accordingly. In addition to the classes below, all the classes defined in Java Specification Document as well is applicable.



# 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. Admin access the /ShowMenuItemListAdmin URL
2. The servlet receives the request in doGet() method
3. Instantiate MenuItemDaoCollectionImpl and assign it to MenuItemDao reference variable menuItemDao.
4. Invoke menuItemDao.getMenuItemListAdmin() and get the list of menu items.
5. The request is set with menuItemList stored in the previous step
6. The request is forwarded to menu-item-list-admin.jsp

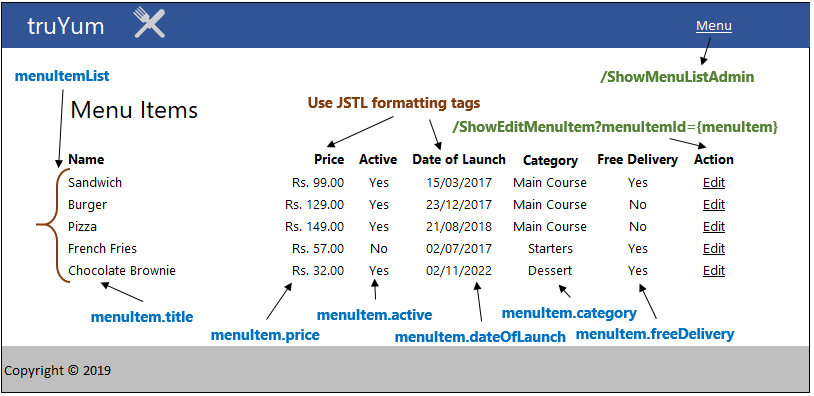
## Servlet

| **Servlet Class Name** | **URL** | **Description** |
| --- | --- | --- |
| ShowMenuItemListAdminServlet | ShowMenuItemListAdmin | This URL is the home page for admin. This URL will be link for ‘Menu Items’ in navigation bar. This servlet fetches the list of menu items for admin and returns the data to JSP. |

## JSP

List the menu items for admin.

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed JSP** | **Description** |
| WebContent/menu-item-list-admin.html | WebContent/menu-item-list-admin.jsp | List the menu items for admin |



# Design for View Menu Item List Customer (TYUC002)

## Sequence Diagram



**Steps Explanation**

1. Customer access the /ShowMenuItemListCustomer URL
2. The servlet receives the request in doGet() method
3. Instantiate MenuItemDaoCollectionImpl and assign it to MenuItemDao reference variable menuItemDao.
4. List of MenuItem returned is stored in a variable
5. The request is set with menuItemList stored in the previous step
6. The request is forwarded to menu-item-list-customer.jsp

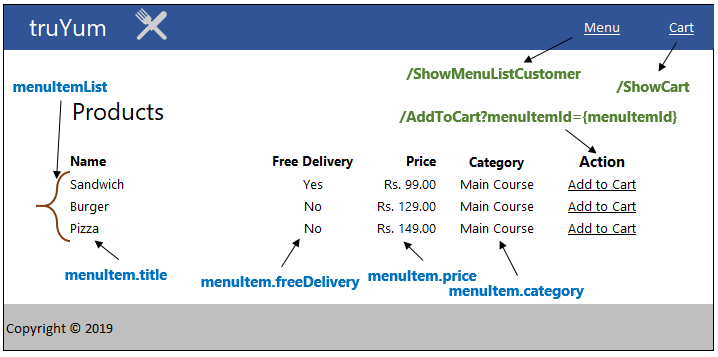
## Servlet

| **Servlet Class Name** | **URL** | **Description** |
| --- | --- | --- |
| ShowMenuItemListCustomerServlet | ShowMenuItemListCustomer | This URL is the home page for customer. This URL will be link for ‘Menu’ in navigation bar. This servlet fetches the list of menu items for customer and returns the data to JSP. |

## JSP

List the menu items for customer.

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed JSP** | **Description** |
| WebContent/menu-item-list-customer.html | WebContent/menu-item-list-customer.jsp | 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 URL
3. The servlet receives the request in doGet() method
4. Instantiate MenuItemDaoCollectionImpl and assign it to MenuItemDao reference variable menuItemDao.
5. MenuItem returned is stored in request
6. The request is forwarded to edit-menu-item.jsp

## Sequence Diagram (Edit Menu Item)



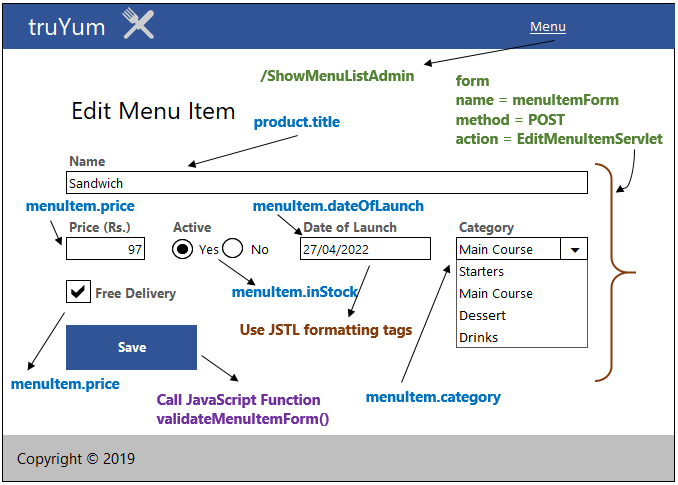
Explanation steps are not included for the above diagram as well as subsequent sequence diagrams. Earlier explanation steps can be referred to arrive at the understanding.

## Servlet

| **Servlet Class Name** | **URL** | **Description** |
| --- | --- | --- |
| ShowEditMenuItemServlet | ShowEditMenuItem?menuItemId=1 | The “Edit” link for each menu item listed in Menu Item List page has to point to this Servlet. This servlet will returns the specific menu item to JSP page for populating the fields in Edit Menu Item page. The parameter id represents the menu item which the Admin has clicked in Menu Item Admin page |
| EditMenuItemServlet | EditMenuItem | This servlet is invoked when Admin clicks save in the Edit Menu Item page. The form data is read by the servlet and save the data. |

## JSP (edit-menu-item.jsp)

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed JSP** | **Description** |
| WebContent/edit-menu-item.html | WebContent/edit-menu-item.jsp | Page with form to edit a menu item |

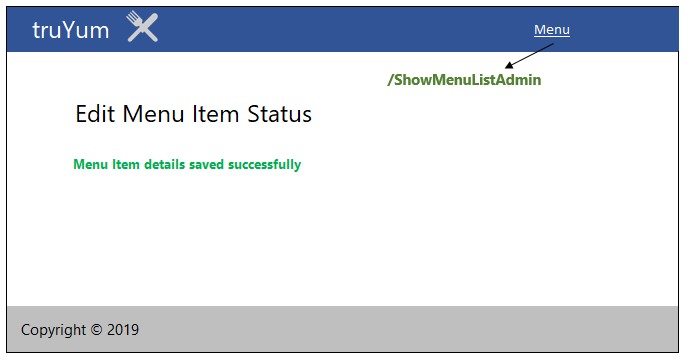


**Category Dropdown Handling**

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 JSP with hard coded values.

## JSP (edit-menu-item-status.jsp)

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed JSP** | **Description** |
| WebContent/edit-menu-item-status.html | WebContent/edit-menu-item-status.jsp | Display the status of modifying menu item. |



# Design for Add Cart (TYUC004)

## Sequence Diagram



**Steps Explanation**

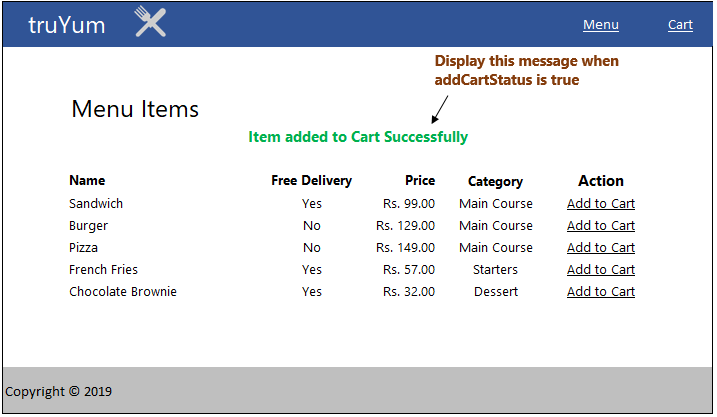
1. Customer clicks on “Add to Cart” link on a menu item from Menu Item List.
2. Customer access the /AddToCart?menuItemId=1 URL
3. The servlet receives the request in doGet() method\
4. Set variable userId of type long to value 1. Identifier for a specific user.
5. Read the menuItemId from the request and assign it to a variable
6. Instantiate CartDaoCollectionImpl and assign it to CartDao.
7. Invoke the addCartItem() method passing the userId and menuItemId.
8. Instantiate MenuItemDaoCollectionImpl and assign it to MenuItemDao reference variable menuItemDao.
9. List of MenuItem returned is stored in a variable
10. The request is set with menuItemList stored in the previous step
11. The request is set with addCartStatus with value true.
12. The request is forwarded to menu-item-list-customer.jsp

## Servlet

| **Servlet Class Name** | **URL** | **Description** |
| --- | --- | --- |
| AddToCartServlet | AddToCart?menuItemId=1 | This servlets adds the selected menu item into the cart of the Customer. The parameter menuItemId denotes the menu item chosen by the customer. |

## JSP (menu-item-list-customer.jsp)

This JSP 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 ShowCartItemServlet forwards the request to cart-empty.jsp.

**Steps Explanation**

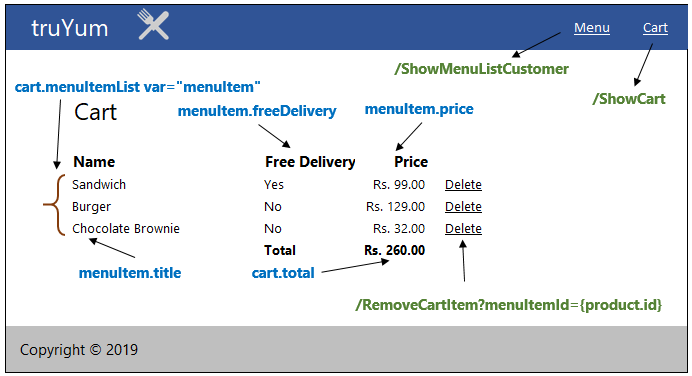
1. Customer clicks on “Cart” link on the navigation bar.
2. Customer access the /ShowCart URL
3. The servlet receives the request in doGet() method
4. Set variable userId with type long having value 1
5. Instantiate CartDaoCollectionImpl and assign it to CartDao.
6. Invoke the getAllCartItems() method passing the userId.
7. If the getAllCartItems() does not throw CartEmptyException:
   1. List of MenuItem returned is stored in a variable
   2. The request is set with menuItemList stored in the previous step
   3. The request is forwarded to cart.jsp
   4. Invoke the getAllCartItems() method passing the userId.
   5. If the getAllCartItems() does not throw CartEmptyException:
8. If the getAllCartItems() throws CartEmptyException:
   1. Catch CatchEmptyException and forward the screen to cart-empty.jsp

## Servlet

| **Servlet Class Name** | **URL** | **Description** |
| --- | --- | --- |
| ShowCartServlet | ShowCart | This servlets get the list of menu items from the customer’s cart for displaying in the Cart page |

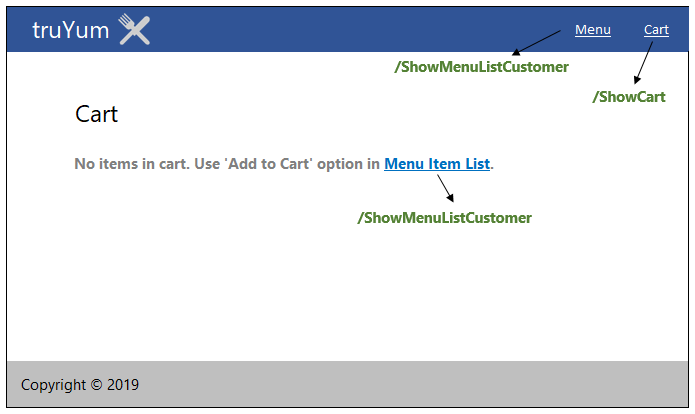
## JSP (cart.jsp)

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed JSP** | **Description** |
| WebContent/cart.html | WebContent/cart.jsp | Displays the cart with the list of menu items added by the customer |



## JSP (cart-empty.jsp)

|  |  |  |
| --- | --- | --- |
| **HTML** | **Renamed JSP** | **Description** |
| WebContent/cart-empty.html | WebContent/cart-empty.jsp | Displays the cart empty message |



# Design for Remove Item from Cart (TYUC006)

## Sequence Diagram



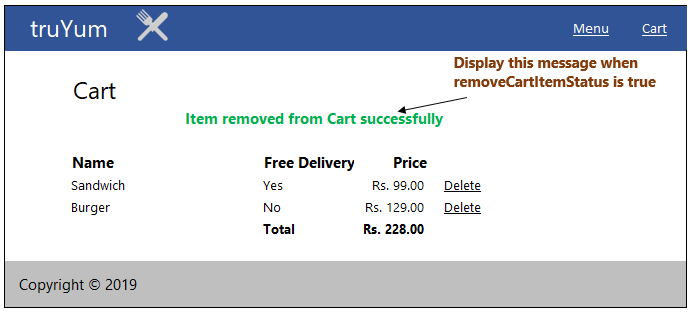
**Steps Explanation**

1. Customer clicks on “Remove” link on menu item in cart.jsp.
2. Customer access the /RemoveCart?menuItemId=1 URL.
3. The RemoveCartServlet receives the request in doGet() 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:
    1. List of MenuItem returned is stored in a variable
    2. The request is set with menuItemList stored in the previous step
    3. The request is forwarded to cart.jsp
    4. Invoke the getAllCartItems() method passing the userId.
    5. If the getAllCartItems() does not throw CartEmptyException:
11. If the getAllCartItems() throws CartEmptyException:
    1. Catch CatchEmptyException and forward the screen to cart-empty.jsp

## Servlet

| **Servlet Class Name** | **URL** | **Description** |
| --- | --- | --- |
| RemoveCartServlet | RemoveCart?menuItemId=1 | This servlet removes the selected menu item from the cart of the Customer. The parameter menuItemId denotes the menu item chosen by the customer. |

## JSP (cart.jsp)



For other field values refer menu-item-list-customer.jsp specification in View Menu Item List Customer.

# Standards and Guidelines

## JSP

1. Dynamic content should be always generated using JSTL
2. Usage of Scriptlets should be completely avoided
3. Indentation should be inline with the standards defined for HTML
4. Use formatting JSTL tags for displaying currency and dates

## Servlets

1. All coding standards applicable for Java are applicable here
2. Servlet should not contain any business logic

# 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 <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** |
|  |  |  |  |