|  |  |
| --- | --- |
|  | **Cognizant Academy**  **truYum**  **ADO.NET Specification Document**  **Version 1.0** |
| |  |  |  |  | | --- | --- | --- | --- | |  | **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 3](#_Toc15911968)

[1.1 Purpose of this document 3](#_Toc15911969)

[1.2 Definitions & Acronyms 3](#_Toc15911970)

[1.3 Project Overview 3](#_Toc15911971)

[1.4 Scope 3](#_Toc15911972)

[1.5 Intended Audience 3](#_Toc15911973)

[1.6 Hardware and Software Requirement 4](#_Toc15911974)

[2.0 Class Diagram 5](#_Toc15911975)

[2.1 Data Access Layer 5](#_Toc15911976)

[2.2 Helper.cs 6](#_Toc15911977)

[3.0 DAO for View Menu Item List Admin (TYUC001) 6](#_Toc15911978)

[3.1 MenuItemDaoSql.cs 6](#_Toc15911979)

[4.0 DAO for View Menu Item List Customer (TYUC002) 6](#_Toc15911980)

[4.1 MenuItemDaoSql.cs 7](#_Toc15911981)

[5.0 DAO for Edit Menu Item (TYUC003) 7](#_Toc15911982)

[5.1 MenuItemDaoSql.cs 7](#_Toc15911983)

[6.0 DAO for Add a Menu Item to Cart (TYUC004) 8](#_Toc15911984)

[6.1 CartDaoSql.cs 8](#_Toc15911985)

[7.0 DAO for View Cart (TYUC005) 8](#_Toc15911986)

[7.1 CartDaoSql.cs 8](#_Toc15911987)

[8.0 DAO for Remove Item from Cart (TYUC006) 8](#_Toc15911988)

[8.1 CartDaoSql.cs 8](#_Toc15911989)

[9.0 Standards and Guidelines 9](#_Toc15911990)

[9.1 DAO 9](#_Toc15911991)

[10.0 Submission 9](#_Toc15911992)

[10.1 Code submission instructions 9](#_Toc15911993)

[11.0 Change Log 10](#_Toc15911994)

# Introduction

## Purpose of this document

The purpose of this document is to define the ADO.Net module implementation for truYum project.

## Definitions & Acronyms

|  |  |
| --- | --- |
| Definition / Acronym | Description |
| DAO | Data Access Object |
| ADO.NET | Activex Data Objects |

## Project Overview

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



## Scope

1. Creation of DAO classes and methods for reading and persisting data of truYum application.

## Intended Audience

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

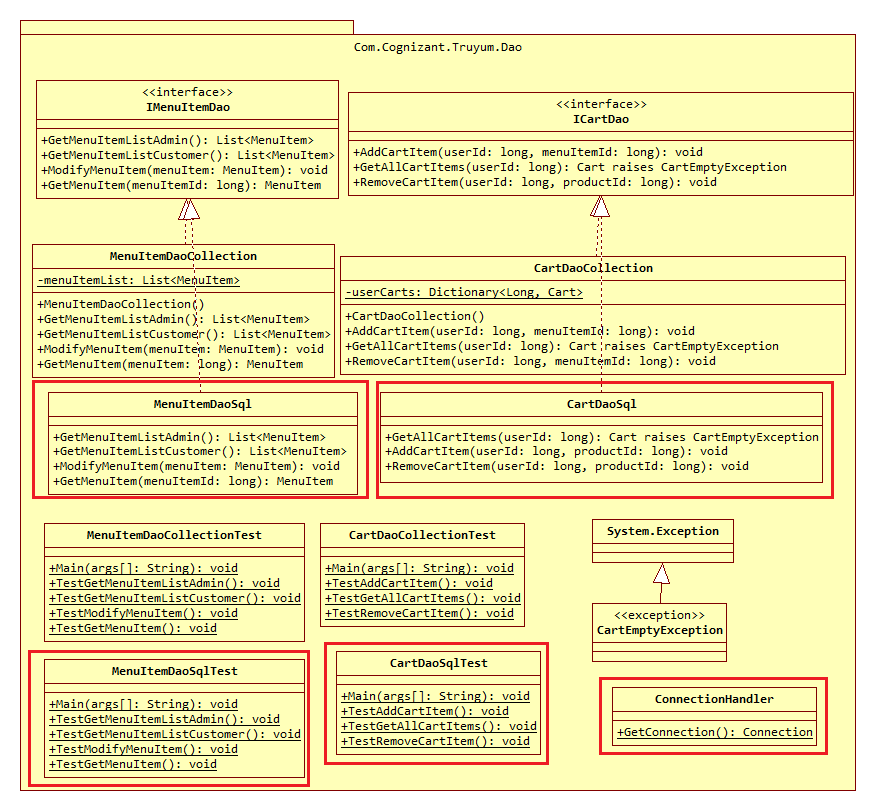
## Hardware and Software Requirement

1. Hardware Requirement:
   1. Developer PC with 8GB 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

# Class Diagram

## Data Access Layer

Refer the diagram below and create classes accordingly.



Dotted arrow represents implementation of an interface.

Make note that GetConnection is a static method.

ADO.Net class library needs to be referenced in the ASP.Net application. The data fetched from database thru ADO.Net should be displayed in the grid in the ASP.Net web pages.

Highlighted classes are the ones that needs to be implemented in this specification.

## Helper.cs

This class will be used by each Dao implementation class for getting the database connection.

The connection details has to be stored in a web.config file. Find the details below:

<connectionStrings>

<add name="connectionString"

connectionString="Data Source=localhost;Initial Catalog=truYum;userid=root;password=password123;Integrated Security=True" providerName="System.Data.SqlClient" />

</connectionStrings>

A static class with a property ConnectionString that gets connection string from the web.config file.

Gets connection using ConfigurationManager Class by referring the name attribute in <connectionStrings> element present in Web.Config file

# DAO for View Menu Item List Admin (TYUC001)

## MenuItemDaoSql.cs

**GetMenuItemListAdmin() of return type List<MenuItem>**

1. Get connection using SqlConnection object
2. Initialize a List of MenuItem
3. Using SqlCommand execute the select query that retrieves all the records from menu\_item table
4. Iterate through the ResultSet
5. For each item in the ResultSet create a new MenuItem instance and add it to the List created in the step 2 and return the List

# DAO for View Menu Item List Customer (TYUC002)

## MenuItemDaoSql.cs

**GetMenuItemListCustomer() of return type List<MenuItem>**

1. Get connection using SqlConnection object
2. Initialize a List of MenuItem
3. Using SqlCommand execute the select query that retrieves the records from menu\_item table applying the following filters:
   1. The menu item is in stock and
   2. The menu item is not past the expiry date
4. Iterate through the ResultSet
5. For each item in the ResultSet create a new MenuItem instance and add it to the List created in the step 2 and return the List

# DAO for Edit Menu Item (TYUC003)

## MenuItemDaoSql.cs

**GetMenuItem(long menuItemId) of return type MenuItem**

1. Get connection using SqlConnection object
2. Execute select query using SqlCommand that retrieves an item from menuItem table based on menuItemId.
3. Create a MenuItem instance and set the values for this menuItem instance from the first item of the ResultSet
4. Return the menuItem created in the previous step

**EditMenuItem(MenuItem menuItem)**

1. Get connection using SqlConnection object
2. Execute update statement using SqlCommand that modifies the values of menuItem table based on menuItemId.
3. Set the parameters of the SqlCommand and execute the statement.

# DAO for Add a Menu Item to Cart (TYUC004)

## CartDaoSql.cs

**AddMenuItem(long userId, long menuItemId) of return type void**

1. Get connection using SqlConnection object
2. Execute insert statement using SqlCommand for inserting data into cart table with userId and menuItemId.

# DAO for View Cart (TYUC005)

## CartDaoSql.cs

**GetMenuItems(long userId) of return type List<MenuItem>**

1. Get connection using SqlConnection object
2. Initialize a List of MenuItem
3. Execute update statement using SqlCommand that joins Cart and MenuItem table to retrieve the list of menu items associated with a specific user.
4. Iterate through the ResultSet
5. For each item in the ResultSet create a new MenuItem instance and add it to the List created in the step 2 and return the List

# DAO for Remove Item from Cart (TYUC006)

## CartDaoSql.cs

**RemoveMenuItem(long userId, long menuItemId) of return type void**

1. Get connection using SqlConnection object
2. Execute delete statement using SqlCommand for delete data into cart table based on userId and menuItemId.

# Standards and Guidelines

## DAO

1. All .Net coding standards are applicable
2. Closure of connection should be done within finally block or thru Using statement

# 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 "jdbc"

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 truYumOnline
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** |
|  |  |  |  |