Class Relationship Document

Relationships For each Class and what it’s connected to and how they work together

**Enums**

This class pretty much has all the enum that the program uses. Version 1 of thisproject has 2 enums

1. **UserType**
2. **Status**

Classes that **contains** UserType enum are

* User
* CacheHelper

Classes that **asscoiates** with UserType

* RuntimeHelper
* UserHelper
* CustomerDashController
* DashboardController
* LoginController
* RegisterController
* ScenesHandler

Classes that **contains** Status

* Order

Classes that **asscoiates** with Status

* TimeHelper
* CusomerSceneBuilder

**CacheHelper**

This class was designed to hold the state of a variable once logged in to use throughout the program, it only contain static variables.

Classes that **asscoiates** with CacheHelper

* CustomSceneBuilder
* RuntimeHelper
* UserHelper
* CartViewController
* CustomerdashController
* DashboardController
* EditItemController
* LoginController
* RegisterContoller
* ScenesHandler

**CustomSceneBuilder**

This class was developed to handle the loading of dynamic view entities it mostly associates with /uses initial program execution and real time updates. It’s also a class of static functions.

Classes that **associates with /uses** CustomSceneBuilder

* RuntimeHelper
* ScenesHandler

**EmailHelper**

This should be a general class to handle email through the system but for version it isn’t it should be more of a static class.

Classes that **associates with /uses** EmailHelper

* TimeHelper
* CartViewController

**FileHelper**

This class handles everything file related from reading to writing to files.

Classes that **associates with /uses** FileHelper

* UsersHelper

Classes that **contains** FileHelper

* ItemsHelper
* OrdersHelper

**GeneralHelper**

This is a classes that provides general service to the bigger classes.

Classes that **associates with /uses** GeneralHelper

* CustomSceneBuider
* UserHelper
* Item
* EditItemController
* ItemController
* RegisterController

**ItemsHelper**

This is a child class that handles everything that is related to the item model and all the things to you allowed to do to the item model.

Classes that **associates with /uses** ItemsHelper

* CustomSceneBuilder
* RuntimeHelper
* CartViewController
* CustomerdashController
* EditItemController
* ItemController

**OrdersHelper**

This class was designed to handle everything for the order model, it controls what is sent to the order model.

Claases that **associates with /uses** the OrdersHelper

* CustomerSceneBuilder
* RuntimeHelper
* TimeHelper
* CartViewController

**PathHelper**

This class was designed to have all the absolute paths in our project.

Classes that **associates with /uses** PathHelper

* CustomSceneBuilder
* FileHelper
* ScenesHandler

**RuntimeHelper**

This classes basically loads items from the file, locates the current instance of a node in the view an update it, all of the methods in it does particularly that.

Classes that associates with RuntimeHelper

* CustomerdashController
* DashboardController
* ScenesHandler

**TimeHelper**

This class handles to count down of an order and updates the view. The class couldn’t have anything static due to the nature of the operation and that it handle multithreading based on time.

Classes that associates with the TimeHelper

* CustomSceneBuilder

**UserHelper**

This was supposed to be a child class but time was against the group so the current design of class remained.

Classes that associates(uses) with UserHelper

* CartViewController
* LoginController
* RegsiterController

**IDataManipulation (Interface)**

This interface was created to handle data manipulation. This is a dynamic interface so far only 2 types are using, it was created to handle any model that was doing CRUD operations and more.

Class that inherits from this interface

* ItemsHelper
* OrdersHelper

**IDisplayUserError (Interface)**

This interface was created to handle displaying user errors to the view.

Classes that inherits from this interface

* CustomerdashController
* DashboardController
* EditItemController
* ItemCOntroller
* Logincontroller
* RegisterController

IInitWrapper(Interface)

This handle the initialization of components into the project view.

All controller classes inherits from this interface.

**Cart**

This is a model class used to hold some form of data in a certain structure, contents of this class isn’t saved to a file its contents are primarily in memory and it loses its state explicitly when you logout, or terminate the application.

Classes that contain the the Cart or a list of the cart

* ItemsHelper
* OrdersHelper

Classes that associates with the Cart

* CustomerSceneBuilder
* RuntimeHelper
* CartViewController
* CustomerdashController

**Item**

This is a model class used to hold some form of data in a certain structure, contents of this class is saved to a file based on user action.

Classes that contains Item

* CahceHelper
* Order

Classes that associates with Item

* CustomerSceneBuilder
* ItemsHelper
* RuntimeHelper
* TimHelper
* CartViewController
* EditItemController
* ItemController

**Order**

This is a model class used to hold some form of data in a certain structure, contents of this class is saved to a file based on user action.

Classes that associates with Order

* CustomSceneBuilder
* OrdersHelper
* RuntimeHelper
* TimeHelper
* CartViewController

**User**

This is a model class used to hold some form of data in a certain structure, contents of this class is saved to a file based on user action.

Classes that contains User

* RegisterController
* Order – could have used the user id but that wasn’t necessary for this version.

Classes that associates with User

* CartViewController
* UserHelper

**AlertdialogController**

This is a class that handle alerts it was supposed to be generalized with more options like yes / no situations but didn’t have enough time figured I might just build a confirm dialog to combat such event or scenario.

Classes that associates with the AlertdialogController

* CustomerSceneBuilder
* CartViewController
* CustomerdsahController

**CustomerdashController**

Handles what display to the user screen

Classes that associates with CustomerdashController

* CartViewController
* CustomerSceneBuilder

**CartViewController**

Handles what display to the user screen

It doesn’t get called in no other class

**EditItemController**

Handles what display to the user screen

It doesn’t get called in no other class

**DashboardController**

Handles what display to the user screen

It doesn’t get called in no other class

**LoginController**

Handles what display to the user screen

It doesn’t get called in no other class

**RegisterController**

Handles what display to the user screen

It doesn’t get called in no other class

**ItemController**

Handles what display to the user screen

It doesn’t get called in no other class