PROGRAM PROCEDURE

USER LOGIN

1. USER LOGIN
   1. NAME CODE

* UserLogin
  1. MAJOR COMPONENT
     1. HTML
* ExistingUserLoginAccount.cshtml – Full view
* CustomerAccountDashboard.cshtml – Full view
  + 1. JAVASCRIPT
* UserAccountLogin.Js
  + 1. C# CONTROLLER
* LoginUserAccountController.cs
  + 1. C# BUSINESSLOGIC
* LoginExistingUserDataLogic
  + 1. C# BUSINESSMODEL
* IGetLoginReturnFDataFromDB – use to extract return value
* LoginReturnInformationObjectModel – method return type
  + 1. C# DATAACCESS
* LoginReturnInformationDataAccess – return value
* LoginInformationFromUserObjectModel – Parameter via Constructor
  + 1. SQL SEVER
* spGetUserDataByLoginInfomation – stored procedure
  1. PROCEDURE
     1. CLICK “CUSTOMER LOGIN”
        1. HTMLs
* Call controller via href at \_LayoutMainPage (Line 26)
* The view will render “ExistingUserLoginAccount.cshtml” using \_LayoutMainPage
  + 1. TYPING @ LOGIN UI INPUT
       1. JAVASCRIPT **– “UserAccountLogin.js”**

**Validate The Real Time**

* Get all input element via query selector all
* Use foreach loop to add event listener
* Add event listener “keyup” to track down input and validate using regex pattens
* Perform Regex function “Validate” and assign to bool variable “isvalid” – noite: Regex function is predefined in library.js
* Is “isValid” is false, add to classlist “invalid” this is predefined at CSS, this will make the border red, depends on class seetings.
  + - 1. CLICK “LOGIN”
      2. CREATING NAME AND NAME INITIAL FOR DISPLAY
      3. C# BUSINESS LOGIC
* Add property to “LoginReturnInformationObjectModel”

public string DisplayName { get; set; }

public string NameInitial { get; set; }

* In business logic “LoginExistingUserDataLogic”, combine the data of from other property , FirstName and LastName using StringBuilders to Create NameDisplay and NameInitialDisplay
* Conditions:

-“LoginStatusNumberCode” property must be equl to 1 (means success login)

-“FirstName” and “LastName” property is not null

* If successful and condition meet, retrieve data at layout view

“\_LayoutUserAccount.cshtml” using

“@modelBusinessModel.ObjectModel.LoginReturnInformationObjectModel” declaration. This is possible because the return view is using this layout.