**WELCOME** PLAN

Version 1.0

**Table of Contents**

[Infrastructure Setup 3](#_Toc527147291)

[Communication Environment 3](#_Toc1759853725)

[Setup New PC Guideline 3](#_Toc1597945812)

[BOSS Application 3](#_Toc1702022721)

[SO Website 3](#_Toc259648025)

[SOAR 4](#_Toc1283047850)

[Theoretical Training 4](#_Toc1728025914)

[SO Business 4](#_Toc1740831574)

[Development Process 4](#_Toc1646996843)

[BOSS & NHibernate 5](#_Toc873776602)

[SOAR 5](#_Toc1413796293)

[SO Website 5](#_Toc1242925443)

[Practical Training 5](#_Toc675219203)

[NHibernate 5](#_Toc1976115802)

[BOSS 6](#_Toc1358636911)

[SOAR 8](#_Toc814787745)

[SO Website 8](#_Toc1991353165)

This document is to help newcomers get familiar with School Outfitters.

# Infrastructure Setup

Purpose: Can run and debug source codes in dev environment, also can start the communication

## Communication Environment

* Outlook 365

## Setup New PC Guideline

* Check it out: [Setup New PC.docx](https://schooloutfitters-my.sharepoint.com/:w:/p/mark_nguyen/EUCW8G-5FRJIl48RsUqQVL0B_81KHcCKqQsqNIOAKb2QpA?email=SCHOOLOUTFITTER%40schooloutfitters.com&e=Fw33Ty)

## BOSS Application

* Visual Studio Community
* Tortoise SVN
* SQL Management Studio
* Janus Winforms (note: Run administrator)  
  \\sovn-server\Software\For Developer\Recommended\Janus Winforms\Version #2 - In Use\SetupWinformsSuitev2(Licensed).msi
* Crystal Report (note: Run administrator)  
  \\sovn-server\Software\For Developer\Recommended\Crystal Report XI - In Use\cr\_xi\_dev.exe
* BOSS Client  
  \\sovn-server\Software\For Developer\Recommended\BOSS Client - V1.6.0
* Check out Master and Dev source code branches

## SO Website

* Visual Studio Community 2022 with installation details below:
  + ASP.NET and web development
  + .NET desktop development
  + .NET:
    - .NET 5
    - .NET 6
    - .NET 2.1
    - .NET 3.1
    - .NET 4.5.2
    - .NET 4.6.2
    - .NET 4.8
    - .NET 4.8 targeting pack
    - .NET SDK
  + IIS Express
  + Nuget package manager
  + C# and Visual Bassic Roslyn compilers
  + MS Build
  + Development activities
    - ASP.NET and web development prerequisites
    - C# and Visual Basic
    - IntelliCode
    - Razor Language Services
* ASP.NET MVC 3 SP1
  + \\sovn-server\Software\Development\MVC\AspNetMVC3Setup-V3.0.20105.0.exe
  + \\sovn-server\Software\Development\MVC\AspNetMVC3Setup-Upgrade to V3.0.50813.0.msi
* Node v10.19.0 (npm 6.13.4) 32 bit  
  \\sovn-server\Software\Development\Nodes\node-v10.19.0-x86.msi
* Python 2.7.17 32 bit   
  \\sovn-server\Software\Development\Nodes\python-2.7.17.amd64.msi  
  *Add Python installed path to the System Environment*
* Get SO Website source code master or develop branch  
  <https://schooloutfitters.visualstudio.com/SO%20Projects/_git/SchoolOutfitters>
* CMD to “run npm install” at root folder: \SchoolOutfitters\
* CMD to “run npm install” at folder: \SchoolOutfitters\FutureVision.Website\
* CMD to execute run webpack command at root folder: \SchoolOutfitters\  
  “node node\_modules/webpack/bin/webpack.js --config webpack.config.vendor.js”  
  “node node\_modules/webpack/bin/webpack.js --config webpack.config.classroom.js”

## SOAR

* Visual Studio Community
* .NET Core 3.1
* Check out Master and Dev source code branches

# Theoretical Training

Purpose: To understand SO business, development process and overall system

## SO Business

* SO history & goals
* SO organization charts
* SOVN organization charts
* Business workflow

## Development Process

* SO Agile Process
* Overall System

## BOSS & NHibernate

* NHibernate
* Source code structure
* Database structure

## SOAR

* Source code structure

## SO Website

* Source code structure
* Database structure

# Practical Training

## NHibernate

Develop a small (Console, Winforms or Web) application that uses NHibernate to CRUD (Create, Read, Update and Delete) data

* Create two tables in database
  + TrainingOrder
    - Id: UNIQUEIDENTIFIER
    - OrderNumber: VARCHAR(20)
  + TrainingOrderItem
    - Id: UNIQUEIDENTIFIER
    - Order\_ID\_FK: UNIQUEIDENTIFIER (this is foreign key to TrainingOrder)
    - ProductSku: VARCHAR(50)
    - ItemPrice: MONEY
* Map two tables to classes Order and OrderItem in the application
* Design a new form (see mock-up below for example) to allow users to CRUD. It requires you use NHibernate to perform these CRUD activities.
  + Create New Order: A new order with items will be saved into database – if there is already the same order number in database, it should raise a message and doesn’t allow users to create it
  + Load Existing Order: Order items of the order entered in the text field will be loaded and populated into the grid.
  + Update Existing Order: Update order items for the order entered in the text field – if there is not the order in database, it should raise a message and doesn’t allow users to update it
  + Add Item: Add a new blank row so that users can enter info for the item
  + Remove Item: Remove the selected item row

Graphical user interface, text, application, email

Description automatically generated

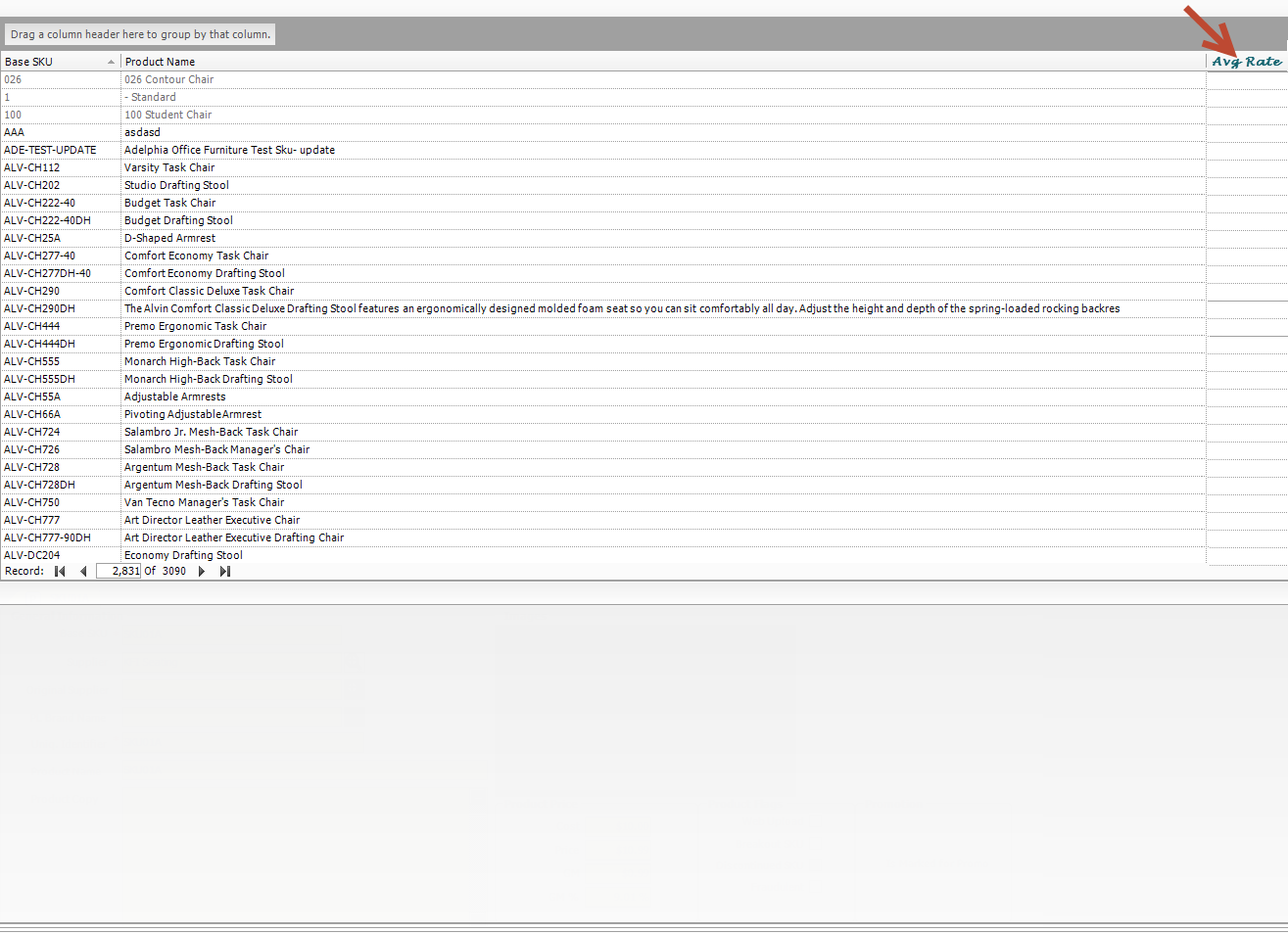
## BOSS

* Create new Product Review module under Marcom as screenshot below

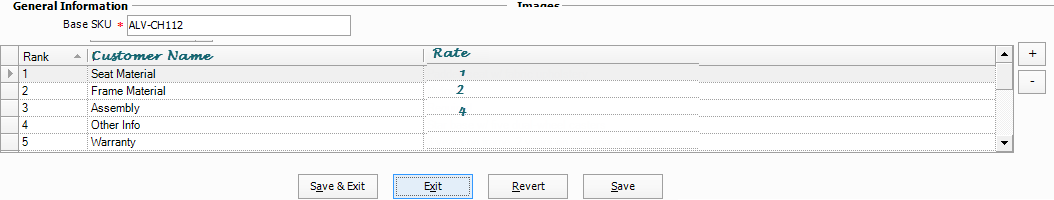
Graphical user interface, text, application

Description automatically generated with medium confidence

* The list view will display corresponding product reviews based on the selected node



* A new detailed screen will display when a record double-clicked



* Click +/- to add and remove rating
* Click Save/Save & Exit to save the rating

## SOAR & BOW

* SOAR
  + Create a new api in the ProductApi service
    - Request
      * GET api/products?fromDate={fromDate}&toDate={toDate}
    - Response
      * Base Sku
      * Product Name
      * Product Cost
      * Product Price
* BOW
  + Add new node named “Product - 1 Week” on the left navigation
  + When the node is clicked, BOW calls the API above to get products created within 1 week and populate the data into a list view by default.
  + There are two filters From Date and To Date on the list view. After selecting them and hitting the button “Process”; BOW re-submits a request to the API above to get the corresponding products.

## SO Website

* Create new table BlogArticle in database

Graphical user interface, application, table

Description automatically generated

* Create a landing page
  + This page should display all blog articles
  + The URL to access this page is https://[domain]/blog
  + The articles should be paginated as 10 articles per page.
  + There should be an ability to switch page to Grid or List view like the mockup below. This should be behaved asynchronously without reloading the full page
  + This page should be used Cache like the current Category page

Graphical user interface, website

Description automatically generated

Graphical user interface, text, website

Description automatically generated

* Create an article page
  + The URL page will be The URL to access this page is https://[domain]/blog/[url-friendly-name]
  + The page should use Cache as well

Graphical user interface, text

Description automatically generated

* Blog module to manage for the Content
  + Create a simple Blog management from the admin to manage the articles
  + User can create/update/delete an article to display on the website
  + Using 3rd TinyMCE to edit the content of the Article
  + You can ignore the Status column, publish all articles to the landing page
  + Action column should be included “Edit” and “Remove” link to edit or remove an article
  + Click “Create a new blog” to open a new page or modal popup window to create a new blog

Graphical user interface

Description automatically generated