|  |
| --- |
|  |
| **DS Team** |
| .NET Internship Training |



.NET Internship Training

SCC VN - DS

Internal Use

Document Control

|  |  |
| --- | --- |
| Document Control | |
| Document Name: | Internship Training |
| Status (Master / Draft / Withdrawn): | Master |
| Document Author: | Khanh Nguyen |
| Document Owner: | Khanh Nguyen |
| Approved by: |  |
| Next Planned Review Date: |  |

|  |  |  |
| --- | --- | --- |
| Version Control | | |
| Version Number | Version Date | Comments / Amendments |
| 1.0 | 07/03/2019 | Initial verson created |
| 1.1 | 26/02/2020 | Add VueJs, change M2 to DS |
| 1.2 | 06/11/2020 | Give final exam in detail |
| 1.3 | 25/03/2022 | Remove VueJs |
| 1.3 | 25/03/2022 | Add Angular |
| 1.3 | 25/03/2022 | Adjust Final Exam: use angular for the front-end |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Distribution List | | |
| Name or Location | Role | Company / Department |
| Khanh Nguyen | Principal .Net Developer | DS |
| Ky Nguyen | Senior .Net Developer | DS |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[1. Introduction 4](#_Toc99085456)

[1.1. Requirements 4](#_Toc99085457)

[1.2. Objectives 4](#_Toc99085458)

[2. Content 4](#_Toc99085459)

[2.1. Programming language 4](#_Toc99085460)

[a. C#.NET 4](#_Toc99085461)

[b. JavaScript 4](#_Toc99085462)

[c. HTML 4](#_Toc99085463)

[d. CSS 4](#_Toc99085464)

[2.2. Frameworks 4](#_Toc99085465)

[a. JQuery 4](#_Toc99085466)

[b. Bootstrap 3 4](#_Toc99085467)

[c. KnockoutJS 4](#_Toc99085468)

[d. Angular 5](#_Toc99085469)

[e. Entity Framework 5](#_Toc99085470)

[3. Planning 5](#_Toc99085471)

[3.1. Git 5](#_Toc99085472)

[3.2. Javascript / Jquery 5](#_Toc99085473)

[3.3. KnockoutJS 5](#_Toc99085474)

[3.4. Angular 5](#_Toc99085475)

[3.5. Bootstrap 3 5](#_Toc99085476)

[3.6. Front End exercise 5](#_Toc99085477)

[3.7. C# & Web application 6](#_Toc99085478)

[3.8. MVC 6](#_Toc99085479)

[3.9. Entity Framework 6](#_Toc99085480)

[3.10. Back End exercise 6](#_Toc99085481)

[3.11. Final exercise 6](#_Toc99085482)

[4. Final Practice Project goals 7](#_Toc99085483)

[5. Final Practice Project Specifications 7](#_Toc99085484)

[5.1. Purchase order list 7](#_Toc99085485)

[5.2. Purchase order detail 8](#_Toc99085486)

[5.3. Sending emails for purchase orders: (NICE TO HAVE) 10](#_Toc99085487)

[5.4. Database schemas: 12](#_Toc99085488)

[6. Document Management & Control 13](#_Toc99085489)

# Introduction

This document is training plan for DS internship programme

## Requirements

Trainee should meet some basic knowledge to participate the training

* Have knowledge about C#, OOP
* Have knowledge about HTML, JavaScript

## Objectives

After complete training, trainee is expected to be able to use below criteria in order to work in DS project

* Familiar with JQuery, Knockout JS, Bootstrap
* Understand MVC, MVVM pattern
* Basic level in using GIT

# Content

## Programming language

### C#.NET

* OOP
* LINQ
* Generic Method

### JavaScript

* Variables
* JSON
* Statements

### HTML

* DOM
* Tags

### CSS

* Classes
* CSS3 vs CSS4

## Frameworks

### JQuery

* Syntax
* Methods, Selector

### Bootstrap 3

* Grid
* Web Responsive

### KnockoutJS

* Observable
* Binding
* Validation

### Angular

* Components
* Directives
* Templates
* Dependency Injection

### Entity Framework

* Object-relational mapper
* Dependency Injection

# Planning

Duration 312 hours with trainer’s effort is 22 hours

## Git

Trainee (8h) – trainer (1.5h)

In order to access materials and doing study, trainee should use Git as the source management

## Javascript / Jquery

Trainee (8h) – trainer (1h)

## KnockoutJS

Trainee (24h) – trainer (3h) <https://knockoutjs.com/>

Trainee will learn about KnockoutJS, as an MVVM framework.

- What is KnockoutJS, MVVM pattern

- Why to use

- How to use

- Exercises

## Angular

<https://angular.io/docs>

## Bootstrap 3

Trainee (15h) – trainer (1h) <https://getbootstrap.com/docs/3.0/getting-started/introduction/>

To complete the front-end development, the trainee will learn about HTML and CSS, especially Bootstrap – a CSS framework

- What is web responsive?

- Why & How to use

- Exercises

## Front End exercise

Combination KnockoutJS + Bootstrap: trainee (24h) – trainer (1h)

After complete the front-end development training, the trainee will have exercises to combine skills

## C# & Web application

Trainee (16h) – trainer (2h)

In this section, trainees will learn MVC pattern. But first, they will have exercises to consolidate the C#.net & web application

- They will learn LINQ, Generic

- They will learn what is web application, how it works

- They will learn web-server, web service, web api

- Exercises

## MVC

Trainee (24h) – trainer (3h) <https://www.asp.net/mvc>

- What is MVC

- Why to use

- How to use

- Exercises

## Entity Framework

Trainee (16h) – trainer (2h)

In Database layer:

- Learn to use Entity Framework (Code First)

- What’s Code First

- What’s Database First

## Back End exercise

Combine MVC + Entity Framework: trainee (24h) – trainer (2h)

## Final exercise

Trainee (200h) – trainer (4h)

In the end, trainees will have an exercise to use all skills they’ve learnt

A Web Application uses Bootstrap to provide the web-responsive ability, use JQuery to provide the enhancement UI, use KnockoutJS to interact with the data changes. The server side will use MVC to implement the business logic, web service/ web api. The connection to DB will use Entity Framework to provide an object mapping for business logic.

Besides, the web application should contain:

- KnockoutJS with observableArray/compute/validation

- KnockoutJS with data-binding/loop statement

- Bootstrap 3, web responsive

- Web service with GET/POST request and return the JSON object

Trainer’s responsibilities:

Apply Scum with 5 sprints within 5 weeks

Apply review CLEAN code

# Final Practice Project goals

Trainee (200h) – trainer (4h)

In the end, trainees will have an exercise to use all skills they learnt. A Web Application uses Bootstrap to provide the web-responsive ability, use JQuery to provide the enhancement UI, use Angular to interact with the data changes. The server side will use MVC to implement the business logic, web service/ web api. The connection to DB will use Entity Framework to provide an object mapping for business logic.

Besides, the web application should contain:

Angular

Bootstrap, web responsive

Web service with GET/POST request and return the JSON object

Apply Scum with 5 sprints within 5 weeks

Apply review CLEAN code

# Final Practice Project Specifications

A small site manages purchasing orders with 3 main functions:

1) Purchase order list

2) Purchase order detail

3) Sending emails for purchase orders

## Purchase order list

Not include column “ETA date” and “Qty Open” column



*PBI 1: Load Purchase Order (PO) List*

**Scenario 1: Load the list as screenshot**

I can:

- See List PO with detail information.

- If a PO sent emails then “Sent Email” column would be ticked.

**Scenario 2: navigate to PO Detail screen when click on the Order No**

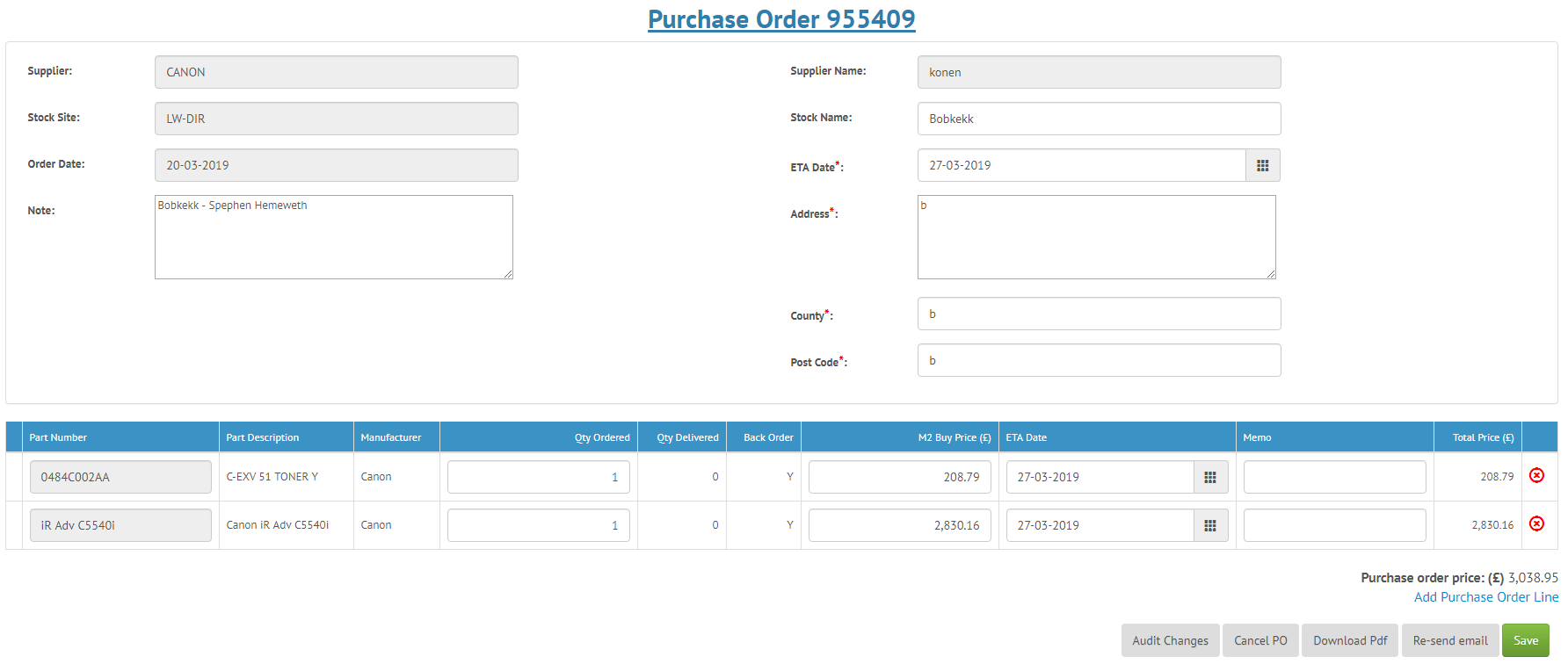
## Purchase order detail

Not include ETA Date

Not include “Download Pdf”/ “Cancel PO”/ “Audit Changes” function

Order date: editable -> not greater than current date

Re-send email will redirect to Sending emails page (screen #3)



**PO Title: Purchase Order XXXX**

**PO Head:**

- Supplier:

- Supplier Name:

- **Stock Site\***: stock site number

- **Stock Name:** stock site name

- **Order Date\***: order date (format: DD-MM-YYYY)

- Note:

- Address:

**PO Line:**

- Part Number:

- Part Description:

- Manufacturer:

- **Qty Order:** Must > 0

- **Buy Price:** Must > 0

- **Order Date:** >= Order Date of POH

- Memo:

**Total Price**: sum of all POLs’ total price

**Buttons:**

- Cancel PO

- Save

*PBI 1. Edit Purchase Order (PO)*

**Scenario 1: Edit detail Purchase Order (PO) from PO Management screen**

Given I am at detailed PO screen

I can:

- Edit information of a selected PO. I see that **Supplier, Supplier Name, Stock Site, Order Date** fields cannot be editable but **Note, Address, County, Post Code** fields can be editable. Then I update some editable fields

- Update **Qty Ordered, Buy Price, Ordered Date, Memo** fields for each PO line, then I click on "Save" button. Updated information is saved successfully.

**Scenario 2: Cannot edit if the PO is cancelled**

Given I am at detailed PO screen and the PO is already cancelled

I can:

- See warning message under the PO title saying that "**Warning:** This PO cannot be modified because already cancelled."

- Not update the fields (POH and POL) as all the fields and buttons become disable and read-only

**Scenario 3: Add more PO line**

Given I am at detailed PO screen

I can:

- See option to add more POL to the current PO. And when I choose to add new PO line,

Then I see new line appears at end of lines table. Part Number drop down list just displays Part Numbers which have not been used in any other PO line. And when I choose to save current PO with added line(s) fulfilled required fields as **Part Number**, Required **Qty** and **Buy Price**

Then I see new line(s) are saved successfully.

**Scenario 4: Remove PO line**

Given I am at detailed PO screen

I can:

- Select one PO line then I would like to remove this PO line by clicking on [-] icon at each PO line. Then I see that this PO line is removed

Note: The PO must have at least one PO line (Warning: "The PO must have at least one PO line")

*PBI 2: View PO*

**Scenario 1: View detail Purchase Order (PO) from PO Management screen**

Given I am at view detailed PO screen

I can:

- See detailed information of a selected PO (POH)

- See List PO Lines with detail information

**Scenario 2: Verify duplicate PO lines on each PO**

Given I am at view detailed PO screen

I can:

- See there are no duplicated Part Number

*PBI 3: Cancel PO*

**Scenario 1: Cancel PO**

Given I am at detailed PO screen. I would like to cancel my PO because my order is no longer required.

I can:

- Click on "Cancel" button. System will display a confirmation message: "Do you want to cancel this PO?" Yes - No.

If user clicks on Yes button, system will have below actions:

\* **Qty Ordered** of all POLs are to be zero

\* **Buy Price** of all POLs are to be zero

If user clicks on No button, nothing happens

## Sending emails for purchase orders: (NICE TO HAVE)

Not include “Reset default template” function

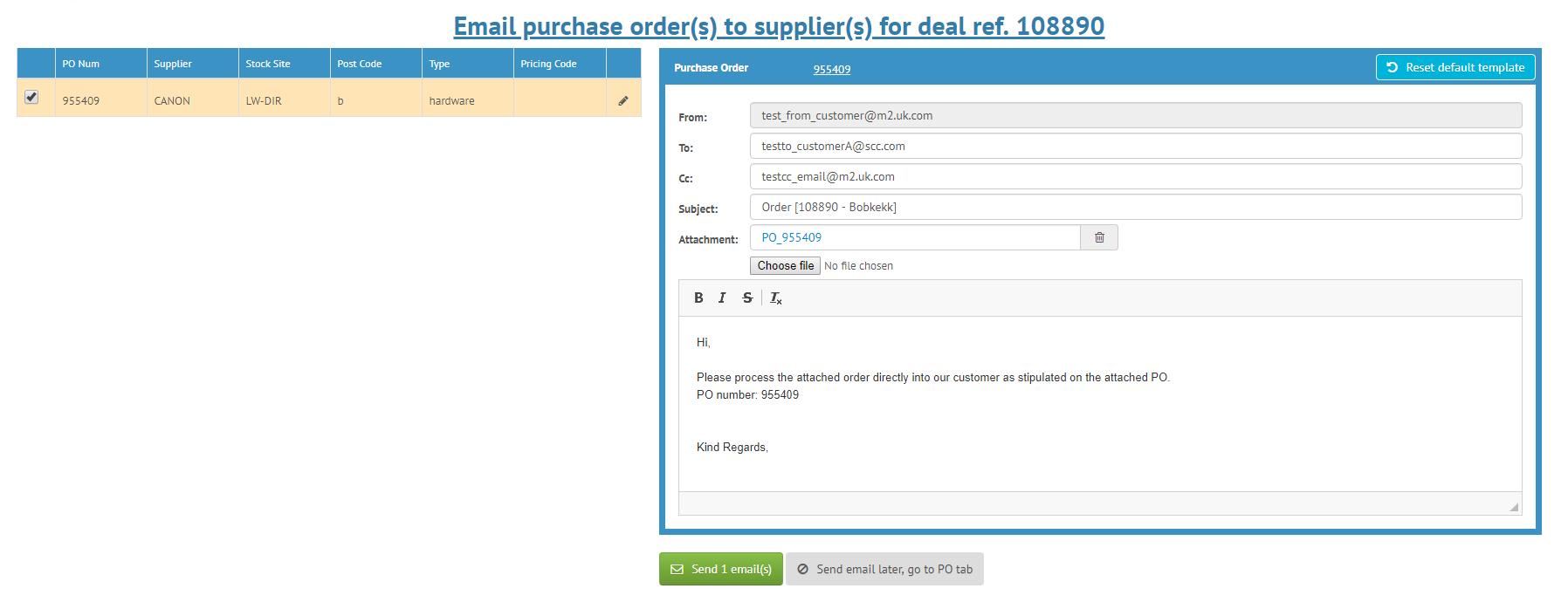
Not include the “Attachment” function

The body of email doesn’t require to use rich text box (WYSIWYG)

“Send email later, go to PO tab” will redirect to the Purchase order list page (screen #1)

Email addresses have to be validated

Provide the ability that let user navigate back to PO Detail screen



*PBI 1: View default email*

**Scenario 1: Load default values**

From address: depends on StockSite

- StockSite: LW-DIR: lw-dir@abc.com

- StockSite: LW-OUT: lw-out@abc.com

- StockSite: LW-MAN: lw-man@abc.com

To address: depends on Supplier

- Supplier: CANON: canon@abc.com

- Supplier: RICOH: ricoh@abc.com

- Supplier: EPSON: epson@abc.com

Subject: text with format Order - <Order Number> <Stock Name>

Contain: text with following information

- First section is as the screenshot

- Second section contains the list of PO Lines in rows whose format is <Part Name>

<Quantity> <Price>

I can:

- See “From”, “To”, “Subject”, “Contain” are fill correctly and follow format

(FROM depend on Stock site Email, TO depend on Supplier Email, Subject with format Order - <Order Number><Stock Name>, Contain).

*PBI 2: Edit email’s content*

Given I am at detailed Email screen

I can:

- I see a text area and I can modify email content.

- To/Cc to multiple addresses, split by comma (‘,’)

**Scenario 1: From address is not editable, the rest is editable**

**Scenario 2: To/Cc addresses support multiple addresses, split by comma (‘,’)**

*PBI 3: Send or not send email*

Given I am at detailed Email screen

I can:

- Click button send and then email will be send/cc to addresses.

- When I click send email later button I will be redirect to PO List.

**Scenario 1:** Send email to addresses and mark the PO as Sent (showed on PO List). Validation is applied to email address (correct format), subject (required), content (required)

**Scenario 2:** Send email later and redirect to PO List

## Database schemas:

PurchaseOrder table

PurchaseOrderLine table

Part table

Supplier table

StockSite table

# Document Management & Control

The document owner, listed in the Document Control section, is responsible for ensuring that a record of its approval, review and change is maintained.

Users of this document are encouraged to provide feedback to the document owner so that they can capture opportunities for improvements as part of the document review process. Revisions should be made only by the document owner. Feedback should be made via e-mail.

Specialist Computer Centres Plc (SCC) Proprietary Rights are included in the information disclosed herein. Recipient by accepting this document agrees that neither this document nor the information disclosed herein or any part thereof shall be reproduced or transferred to other documents or used or disclosed to others for any other purpose except where specifically authorised in writing by SCC