COS30043 – Interface Design and Development

Learning Summary Report

TA QUANG TUNG (104222196)

Self-Assessment Details

The following checklists provide an overview of my self-assessment for this unit.

	Pass (D)	Credit (C)	Distinction (D)	High Distinction (HD)
Self-Assessment				✓

Self-assessment Statement

	Included (please tick)
Learning Summary Report	✓
Use of Bootstrap that demonstrate coverage of core concepts	✓
Use of VueJS that demonstrate coverage of core concepts	✓

Minimum Pass Checklist

	Included (please tick)
Progress on Credit Tasks	✓
All Pass Tasks signed off	✓

Minimum Credit Checklist, in addition to Pass Checklist

	Included (please tick)
Credit and Pass Tasks done, and Progress on Distinction Tasks.	✓
Custom program meets Distinction criteria	✓
Design report with screenshots for custom program	✓

Minimum Distinction Checklist, in addition to Credit Checklist

	Included (please tick)
Research report, and associated pieces	✓
Custom project meets HD requirements	✓

Minimum High Distinction Checklist, in addition to Distinction Checklist

Declaration

I declare that this portfolio is my individual work. I have not copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part of this submission been written for me by another person.

Signature:______________________________

Introduction

This report summarises what I learnt in COS30043 – Interface Design and Development. It includes a self-assessment against the criteria described in the unit outline, a justification of the pieces included, details of the coverage of the unit's intended learning outcomes, and a reflection on my learning.

Overview of Pieces Included

This section outlines the pieces that I have included in my portfolio (the highlighted tasks are the ones that I am most proud of and best demonstrate my learning outcomes):

- 1.1P - Data preparation
- 1.2P - Hello World
- **1.3P - Form Accessibility: A simple HTML form that fully meets accessibility guidelines.**
- 1.4P - Table Accessibility
- 2.1P - Getting bootstrap up and running
- 2.2P - My calculator
- 2.3P - My bootstrap template library
- 3.1P - String test
- 3.2P - Unit lookup
- 3.3C - BMI
- 3.4C - Registration
- 4.1P - Guessing game
- 5.1P - My post
- 5.2P - Menu
- 5.3C - Router
- 6.1C - Form validation
- **6.2D/HD - Custom project: A HD-level custom application made with Vue and Vuetify.**
- 7.1P - Requesting external data
- 7.2P - Retrieving data from a text file
- 8.1P - Student mark
- 8.2C - My router
- 9.1P - Single page application
- **9.2C - Single page application: A single page application made with Vue and Bootstrap that fully supports CRUD features for unit management.**
- 10.1P - Creating a Single Page Application Using Vue CLI
- **10.2HD - HD project: A video tutorial on how to use the TanStack Query library to fetch data with Vue.**

Coverage of the Intended Learning Outcomes

This section outlines how the pieces I have included demonstrate the depth of my understanding in relation to each of the unit's intended learning outcomes.

ILO 1: Apply Design

Apply fundamental design concepts and standards to the development of user interfaces

The following pieces demonstrate my ability in relation to this ILO:

- **1.3P - Form Accessibility and 1.4P - Table Accessibility:** The former features an accessible form with appropriate labels for all inputs and context grouping to group related fields together. The latter features an accessible table with clear headers and scoping.
- **9.2C - Single page application:** This application features a fully responsive dashboard to display and modify unit information. Key UI elements used are tab, table, and pagination.

- **6.2D/HD - Custom project:** This is an advanced application that follows a wide range of UI principles and concepts: mobile-first responsiveness, 12-column system, context grouping, and accessibility.

ILO 2: Use Frameworks

Use contemporary frameworks to create dynamic user interfaces.

The following pieces demonstrate my ability in relation to this ILO:

- **All tasks from 3.1P onwards**: Vue.
- **9.2C - Single page application:** Vue (interactive UI) and Bootstrap (styling).
- **6.2D/HD - Custom project:** Vue (interactive UI) and Vuetify (components and styling).
- **10.2HD - HD project:** TanStack Query (client-side data fetching)

ILO 3: Develop User Interfaces

Design and develop user interfaces optimised for a range of devices and platforms.

The following pieces demonstrate my ability in relation to this ILO:

- **All tasks from 3.1P onwards:** For these tasks, I used a combination of Vue.js and Bootstrap to develop responsive user interfaces.
- **9.2C - Single page application:** The fluid layout for this application adapts to screens as small as 320 pixels in width. The content does not overflow and cause the page to shrink on small screens. This task also makes use of pagination to load content in pages at a time, increasing performance.
- **6.2D/HD - Custom project:** This application is designed with a mobile-first approach. Each type of screen size has a different layout as appropriate.

ILO 4: Evaluate User Interfaces

Evaluate user interfaces with respect to usability and accessibility using appropriate techniques, and propose improvements.

The following pieces demonstrate my ability in relation to this ILO:

- **1.3P - Form Accessibility and 1.4P - Table Accessibility:** This application was jointly validated by AChecker Web Accessibility Checker and the HTML-Validate extension of Visual Studio Code to identify any accessibility problems. Through these tasks, I have learned how to design accessible forms and tables.
- **6.2D/HD - Custom project:** Because this application uses Vuetify, an implementation of Google's Material design specifications, it guarantees conformance to accessibility guidelines.

Reflection

The most important things I learnt:

The most important thing I learned in this unit was using Vue.js to build a single page application. I have learned that the workflow of SPAs is very different from that of a traditional web application, such as one made with PHP. This unit has also introduced me to several important UI design concepts such as the 12-column system and context grouping, which I think will be useful to me in the future.

The things that helped me most were:

I think what helped me the most was practicing the concepts through the lab tasks and custom application. This practice allowed me to get used to the mindset of developing SPAs and better understand the principles taught during the lectures, especially those related to usability and accessibility in the early weeks.

I found the following topics particularly challenging:

Vue.js, the central topic of this unit, was quite challenging to me because this was the first time I had used it. It took me a while to get used to Vue's state and lifecycle system. In the first few assignments, I encountered many problems that took quite a long time to resolve. However, these challenges have given me the chance to hone my self-studying and documentation-reading skills.

I found the following topics particularly interesting:

Despite being quite challenging, Vue.js was really interesting for me to learn because it provided me a point of contrast with React.js. I have had much previous experience with React.js and learning Vue has made me realize they have a lot in common. Namely, both encapsulate state in components, have lifecycle methods, and follow the general pattern of data down, event up.

I feel I learnt these topics, concepts, and/or tools really well:

I think I have learned Vuetify really well in this unit even though it is not officially part of the syllabus. It is a component library for Vue and I used it extensively for my custom app. I found that using Vuetify greatly simplified my development process while also giving me a great-looking UI. The only drawback to this library is how hard it is to customize small details, though this pales in comparison with its convenience.

I still need to work on the following areas:

One area that I will need to work on is testing my Vue UI for accessibility conformance. Testing a simple HTML markup for this is much easier as validators can parse pure HTML. However, once I add the Vue markup, these validators cease to function properly. I think I will need to look for a validation service or library that can evaluate my Vue code.

My progress in this unit was …:

I think I made really good progress in this unit. I was largely able to work independently to solve lab problems and build my custom app. I was able to submit all my assignments on time and demonstrated my work during lab hours with no issues. In retrospect, I think I have learned more things than I had expected.

This unit will help me in the future:

I plan on becoming a full-stack developer in the future so the front-end knowledge acquired in this unit will definitely be useful. I don't intend on using Vue again as I am more of a React person, but the concepts will carry over to make me a better React developer. In fact, I am quite surprised by how much Vue development mirrors React. Aside from the technical similarities such as state management and lifecycle hooks, they both have a Material-inspired UI library (Vuetify vs. Material UI), and data fetching library (TanStack Query).

If I did this unit again I would do the following things differently:

I think I should have focused more on the frontend of my custom app as this is a UI-centric unit. I made the mistake of focusing too much on the custom backend (which is unfortunately not graded) that I didn't have much time for the UI. Thankfully, using Vuetify simplified development and allowed me to quickly construct a responsive and good-looking interface. However, had I not used Vuetify, I would have been in quite a rush to make the frontend functional.

Other…:

None.