COS30041 – Creating Secure and Scalable Software

Learning Summary Report

Thomas Wright 10470604

Final Portfolio Submission Due

Pass / Credit:	Week 13, Mon, 9:00am
Distinction / High Distinction:	Week 13, Tue, 9:00am

Portfolio Interview Dates

Distinction / High Distinction:	Week 13-14, TBA

[Optional, but Strongly Recommended] Tutor's Feedback on LSR

Timing of the Feedback process and submissions

Grade Description	Pass / Credit	Distinction / High Distinction
LSR only [9.9]	Week 12, Tue (2 June), 09:00	Week 12, Wed (3 June), 09:00
Tutor's final feedback	Week 12, Thu (4 June), 17:00	Week 12, Fri (5 June), 17:00
Final Portfolio	Week 13, Mon (8 June), 09:00	Week 13, Tue (9 Jun), 09:00

Self-Assessment Details

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

	Pass (D)	Credit (C)	Distinction (B)	High Distinction (A)
Self-Assessment (please tick)		✓		

Self-assessment Statement

	Included (please tick)
Learning Summary Report	✓
All Pass Tasks are Compete in Doubtfire	✓

Minimum Pass Checklist

	Included (please tick)
All Credit Tasks are Complete on Doubtfire	✓

Minimum Credit Checklist, in addition to Pass Checklist

	Included (please tick)
Interview booked	
Software proposal is “Complete”	✓
Your custom-built Enterprise application of your own design meets Distinction criteria and standards	
Software Design doc and Software Test report of your custom-built Enterprise application meet Distinction criteria & standards	
Other pieces (please specify)	

Minimum Distinction Checklist, in addition to Credit Checklist

	Included (please tick)
HD Research Proposal is “Complete” on Doubtfire	
A Research report and associated pieces (e.g. source code, if any) that meet HD criteria and standards	
Other pieces (please specify)	

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: Thomas Wright

Portfolio Overview

This portfolio includes work that demonstrates that I have achieve all Unit Learning Outcomes for COS30041 Creating Secure and Scalable Software to a **Credit** level.

Throughout the course of the semester I created multiple enterprise java applications. In task 5.1 I created a scaleable web application that allows employees of a company to login with a username and password, and view & edit their personal information, and emails the employee when their information is updated. In 7.1 I created a similar web-application secured using Glassfish security roles to secure the application.

In tasks 3.1 & 4.1 I used the java persistence library to use object relational mapping to store & access data from a database as objects.

In this subject I have learned how to implement Java EE beans in order to improve the scalability of the application using bean pooling. In task 4.1 I extended the functionality of a stateless bean to provide data access methods, then in task 6.1 I extended a managed bean to create a shopping cart for a digital storefront.

To enforce strict user login requirements in task 5.2C I used JSF validator tags to user enforce password requirements, and a custom validator class to implement a confirm password function. In task 7.1 I used glassfish 4.1.1’s security roles to secure an application using user credentials defined in the glassfish file realm.

In tasks 5.1 and 6.1 I analysed existing enterprise java applications and created, justified, and implemented designs to secure the application and provide additional functionality.

Reflection

The most important things I learnt:

How to write Java code. This is the first unit that I've used java in and given its widespread user throughout the IT industry the exposure to the language will likely be extremely beneficial.

Learning how to use object relational mapping to persist data was useful as using ORM libraries makes it easier to persist business objects.

The things that helped me most were:

One or two of the oracle docs online. This page https://docs.oracle.com/cd/E19798-01/821-1770/gcrkr/index.html in particular helped me create UML diagrams for the design tasks. For the most part however there wasn't a great amount of help available online for the topic, which seemed to be in part because of how old the system is.

I found the following topics particularly challenging:

I didn't find any of the concepts particularly challenging, however I ran into a few errors that slowed me down for really long amount of time. For example, glassfish 4.1 has a bug in which when attempting to create new jdbc resources for configuring a security realm, it throws an error and you are unable to create the resource. I tried to use a payara server instead as apparently it is still supported, unlike glassfish 4.1, however could not get it working and instead had to configure the example resources and pool to use my database.

I found the following topics particularly interesting:

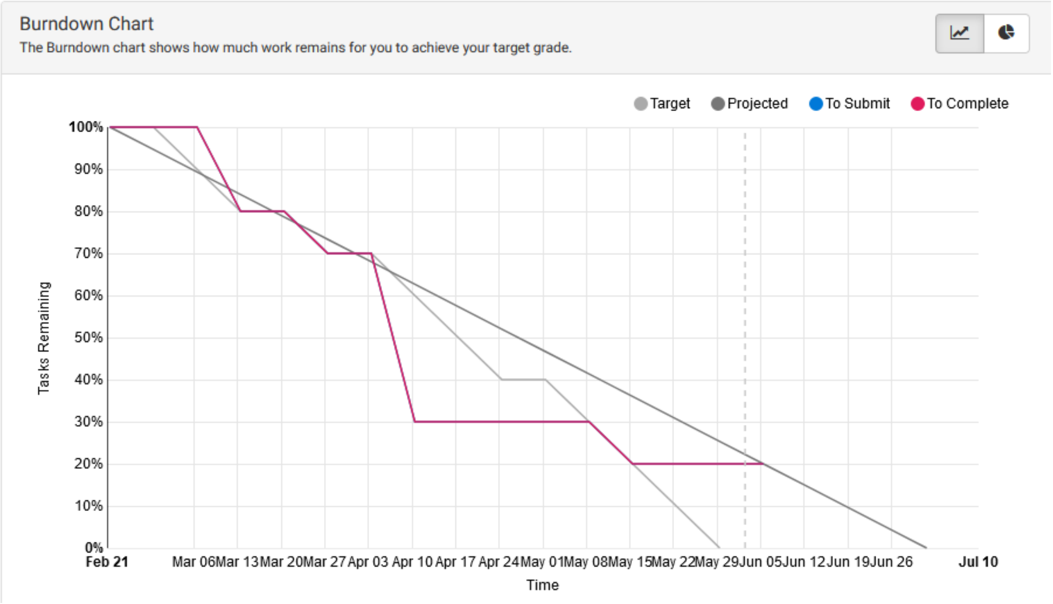
I found the concept of bean pooling interesting, as well as the idea of having objects dedicated for storing session data, or performing stateless operations

I still need to work on the following areas:

Using Java EE security roles. I've been able to put together apps with separate user groups and different functionality for each, however this was done by essentially creating two separate front ends. I'd like to figure out how to use enterprise java to create a dynamic, shared UI that changes based on the user group accessing it.

My progress in this unit was …:

I started out strong and was able get a lot done when the technology was working for me, however I spent dozens of hours troubleshooting issues with Netbeans and Glassfish, and ultimately was unable to pursue a distinction project due to difficulties configuring a jdbc realm with glassfish.



Burndown Chart

The Burndown chart shows how much work remains for you to achieve your target grade.

This unit will help me in the future:

If I’m ever required to use Enterprise Java for an employer in the future then my experience with this unit will have prepared me to work on their application. Using a persistence library will help me in future projects

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

The only thing I would have done differently is I would have tried to get a different server running to avoid the issues I had with glassfish.