ePortfolio (https://michaelbotha-repos.github.io/ePortfolio-MSc/#)

Module Reflection

Unit 1-2: Exploring Key Information System Concepts

The content covered in the first two units presented the fundamentals of an Information System (IS), and the ways ISs might fail. Bourgeois (2014) explained how an IS exists as a combination of a few primary components. I found his book helpful in explaining the holistic process of using information and its representative data to produce value.

Overall, the content covered guided me in learning the major components of an IS, how they interrelate, how they can fail, and their importance to businesses and society in general. Furthermore, I found it crucial to take note that for an IS to be effective, the business processes that undergird and govern the use of them need to be efficient and all-encompassing respectively (Burgeois, 2014). This point and the rest of the learning helped me to see the various IS inefficiencies within my own organisation, where disparate systems produce duplication of activities. Additionally, I learnt how the automation and business workflows that should be provided by my company's ISs have been disconnected by various manual activities not linked to the IS.

<u>Unit 3-4: The Object-Oriented Paradigm</u>

In unit 3 and 4 the Object-Oriented (OO) model of creating and implementing software was covered. Many principles and their related constructs were learnt. For instance, class and instance variables, methods, inheritance, encapsulation, and polymorphism. I experienced further growth by being required to understand design principles for an OO system, where relationships of the envisaged modules or objects had to be formed to produce specific abstractions and functional flows. It was challenging for me to understand the types of relationships: dependency, association, aggregation, and composition, without first seeing these abstract concepts through code examples. Therefore, I had to extensively browse the Internet for appropriate and well explained illustrations.

Understanding the referencing of variables within Python as the interaction of objects significantly improved my grasp of OO programming concepts. Additionally, I can now plan my code a lot better having learnt the modularity and relational principles indicative of OO design and programming. However, the design element can seem very intuitive and producing the right classes with their respective relationships will certainly require further practice.

Unit 5-6: Universal Modelling Language (UML)

Unit 5 and 6 covered the UML and its various notations, as well as the focal points of particular diagrams, namely the Use Case, Class, Object, Sequence, Activity, and State Chart diagrams. UML provides a standardised method to model the previously learnt concepts of OO programming when required to design an OO system (Ambler, 2003).

I needed to apply the concepts learnt from unit 1 up to this point to be able to produce the design for my online store IS required as the mid-module assignment. It was particularly difficult to produce the design, not because of a lack of understanding of the concepts and notations covered, but due to not having sufficient experience with using best practices as it pertains to creating objects and interrelations, especially in the abstract way of a UML representation. Therefore, more exposure to implemented designs as well as experience in creating my own will be required to master this skill. However, I find UML a very useful tool that will aid me in future to model a system during the design phase of the Software Development Lifecycle (SDLC).

Unit 7-8: Relational Database Design

Unit 7 and 8 covered various general concepts pertaining to databases but focused on the Relational Model and the techniques required to design a database of this type. Databases provide an information system with various functionality not easily available through file-based systems (Connolly & Begg, 2014). For instance, better data integrity and consistency, easier sharing of data, and better security (Connolly & Begg, 2014).

I learnt the method to and importance of normalising a relational database to at least the third normal form (3NF), and the various ways one might want to link tables by setting up primary and foreign keys. Therefore, I now feel comfortable with designing a relational database. Furthermore, it was very interesting to relate the concepts learnt back to the resource management IS as well as the SAP management IS used in the company I work for, and to muse on how the underlying database structures should reside.

Unit 9-10: Database Implementation and Manipulation

The concepts learnt in the previous two units were here engaged with in a more pragmatic approach through exposure to the Structured Query Language (SQL), and the open-source MySQL Relational Database Management System (RDBMS). SQL provides the two important facets of a database query language: data definition and data manipulation.

I found creating and manipulating a database using SQL very interesting and straightforward, however applying the last four units' concepts in the development of my Online Store required reengineering, as the database design was not originally accounted for in my first thought processes. When incorporating a database into my system certain functions were a lot easier to perform, but the OO modelling previously used seemed to breakdown slightly. This due to the facilities now available through a separate system where data could be fetched and manipulated at any point in the code, which does not align with the principle of private attributes in OO programming.

Unit 11: Web Development with Python

Unit 11 covered some fundamental principles of web development and introduced the Flask microframework available in Python, commonly used to facilitate back-end functions. A tutorial was followed which guided me in setting up a blogging website, where various principles such as the following were learnt, enabling me to setup up a basic webserver:

- Python virtual environments

- Flask routes

- Flask templates written in Jinja

- Python's native SQLite database library

- Services like password hashing from the werkzeug library

- Jinja and SQLite character escaping security feature

Further practice in Flask is required for me to better grasp the related principles and syntax, as 1 week is a very short timeframe, however essential web principles were understood including the Model, Controller, and View design principle.

Unit 12: Emerging Trends of Information Systems

This week reiterated IS concepts covered throughout the course, with the addition of security fundamentals. Furthermore, I chose to research 5G as my selected area of research and development in ISs, because I already covered Artificial Intelligence (AI), Machine Learning (ML), Cloud computing, Big Data, IoT, and other relevant emerging areas of research in the previous module.

I found it very interesting how mobile technologies like 3G and 4G have revolutionised the IS industry. This knowledge has created more awareness for me regarding the current requirements of a computer scientist when considering the full reaches of ISs. Most of the security principles covered was revision.

References

Ambler, S. (2003) The Elements of UML Style. Cambridge: Cambridge University Press.

Bourgeois, D. (2014) Information Systems for Business and Beyond. Washington: The Saylor Academy.

Connolly, T., Begg, C. (2014) *Database Systems: A Practical Approach to Design, Implementation, and Management*. Pearson Education available

Sommerville, I. (2016) Software Engineering. 10th ed. Essex: Pearson Education Limited.