OOIS: Project Evaluation

Course: MSc Computer Science

Module: Object-Oriented Information Systems

Assignment: ePortfolio

Date: Saturday 24th July 2021

Student ID: 126853

Project Evaluation:

Within the Object-Oriented Information Systems module, we were provided with an ‘online store’ scenario. This was required to have a Customer, Seller and Warehouse portal, providing various functions to the user. Within Unit 7, we were required to produce a series of UML diagrams to base our final implementation, consisting of a Class Diagram, an Activity Diagram, and a State Diagram (Highlighting system transitions). Finally, within Unit 11, we were required to implement the designs from Unit 7 in an object-oriented fashion using Python, highlighting cases where a SQL database could be used.

In the system planning stage (Unit 7), a collection of three UML diagrams was completed, highlighting the key elements of each class and documenting the transitions/events that were to occur at a specific point within the program. The first diagram produced was a Class Diagram, which was created based upon key items identified as ‘objects’ within the assignment brief, focusing on the type of relation and method/function visibility. The second diagram produced was an Activity Diagram, which highlights actions that the users can take during the checkout process. The final diagram built was the State Diagram, highlighting the various stages of the order process (Such as In Progress, Awaiting Picking etc.). These diagrams have been handy at the later stages of the development process, as they provide a crucial reference point when creating classes and functions.

When creating the code deliverables for this project, I had ensured that each class was separated into a standalone file, which the requesting class would import. This

ensured that resources remained easily readable and distinct and provided benefits such as code reusability and optimised compilation (Mefford, 2019).

All code produced was intended to conform to the PEP8 style guidelines (Van Rossum, 2001), which is a set of style guidelines produced by a series of developers from the Python Software Foundation. All code was run through the AutoPEP8 (PyPi, n.d.) module, which is available through the Python repository to perform a ‘first fix’ of the code, ensuring it is compliant with the style guidelines. A final manual check was run through the style checker within the PyCharm IDE to ensure it was fully compliant. Implementing this stage into my development process ensures that code is in a standardised format that is easily readable by other Python developers.

When producing the Readme document to accompany the code, I ensured that detailed platform-agnostic documentation on how to execute the code was included. For example, my project makes use of standard Python Virtual Environments, which can ensure that regardless of packages installed on the host machine, you can set up an identical environment to that of my development setup. In addition, this ensures that code is less likely to contain versioning errors, especially when using external modules that could have changed, such as the PrettyTables module.

As testing is an essential part of the development process, each class and method were tested individually during the development process, as well as all together at the end of the project. I compiled a thorough list of items/functions that needed to be tested, which was used to produce a comprehensive test plan. This test plan contained a list of the expected outputs of the system, which was based upon the

system brief and was then compared to the actual outputs of the system, highlighting any inconsistencies. This stage was completed to ensure that all code was functioning as expected, following the provided assignment brief.

In conclusion, I believe that I have thoroughly met all aims and objectives of this module whilst demonstrating my knowledge and understanding of object-oriented techniques, the Python programming language, and the production of UML diagrams. Although the overall object-oriented techniques are the same, this project has highlighted the vast differences in how they are implemented across programming languages I am more familiar with, such as PHP.

References:

Mefford, A. (2019) Should I keep each Python class in a separate file?. Available From: https://www.quora.com/Should-I-keep-each-Python-class-in-a-separate-file [Accessed 24th July 2021].