

Comp 47480 Learning Journal – Assignment 2

1. Reflection and Account on Team

During the practical our team first started the assignment by addressing the use case model this was an important part to defining what the library would do. Whilst reading through the instruction we picked out that there were 2 types of member that would be able to access the system. The student and staff would be a generalisation of the model whilst every member could borrow and return books only staff could borrow journals and to return either then there is a dependency that a book would be borrowed in the first place. I found this one the most difficult as I wanted to be able to define more things the library could do. I found it difficult to hone in on just the basics. We then moved forward to the class diagram which we have used more and used to show that the library consisted of items which were either of type book or journal and members that were either of type student or staff in this diagram we were able to define attributes that each class might have for instance a time period due to the fact that we could take out a book for 4 weeks and others were on short term loan. This for the team was relatively straightforward. Subsequently the sequence diagram could be addressed. This diagram was to show an action that might happen within a class on a timeline. The first sequence we decided upon was to borrow an item it the first sequence the system checks that the member has not reached their limit of items they can borrow. The second sequence that we had to design was to send an alert to the user when their book was out of time and needed to be renewed. This functionality required an interface to send the requests to the correct class (as seen in the artefact). In this the system runs a query on all the items in the library and checks the time borrowed against the last borrowed date if it is over the time allowed it then finds the user who has the book out and alerts the member.

2. Reflection on my Learning on UML

After attending the lectures and the practical given about UML, it has highlighted to me my knowledge of UML and the its different models has been quite limited to this point. Previously, I had used UML to depict class diagrams and had little knowledge of the other diagrams that UML encompassed. I was aware that UML could be useful tool in many industries in providing a methodology of how to approach constructing a mock-up of the system that team is implementing and providing a tangible visualization of how the system will work or what the system design will do to senior management that may not understand the system fully if the team was to describe the system by using technical jargon alone. UML is a graphical language used to model systems using

relationships of the components and the dependencies between these components. I have learned that while I have only really used the class diagram. The unified modelling language consists of use case diagrams, class diagrams, sequence diagrams, collaboration diagrams, state chart diagrams, activity diagrams, component diagrams, deployment diagrams. The use case diagram is probably the simplest form of diagram as it just represents what the system can do. It doesn't provide implementation however it does consider dependency of features. It can help in deciding what features are necessary in the system. The domain model /class model is one that I have previously used this I find provides a useful first mock-up of the system by conceptualizing the essential classes and features. It makes use of relationship between components which can be described in three ways association aggregation and generalization. A generalization occurs when there is a class that inherits functionality from the super class. An association is when there is some affiliation between classes and aggregation is when a class belongs to a collection. Arrows provide an idea of the direction of the association and we can provide a way of showing the number of possible instances of the class by providing multiplicities like zero or one instance by using 0..1 notation. The sequence diagram shows how the model interacts with the classes in terms of the operations. These are read top to bottom not left to right as they are depicted like a type of timeline. Collaboration diagrams also provide a representation of interactions however it focuses mainly on the role of the objects. I feel this diagram is more useful as an interaction diagram and I would lean towards using this as I feel it's closer to the class/domain models that I'm comfortable illustrating, so this feels like a good way of providing some modelling of interactions.

Personally, I find some of the UML diagrams simplistic and find it difficult to depict some of the diagrams because I tend to lean towards putting too much of the implementation and complexity into the diagrams where simplicity is necessary. I would much prefer to mock up the bigger picture of the system than to simplify it, to the point that I feel the diagrams can become too simplistic and insignificant to warrant illustration. Although UML is not continually utilized in industry it remains a useful tool where the abstraction of the model is necessary to maintain common system goals and design concepts between programmers where systems can become complex and difficult to understand. It also provides a common language that the everyday person can understand due to its graphical nature. I feel through the course of this practical I have gained a better understanding of UML for demonstrating the model of a system.