A Reflection on UML on a Postgraduate Course in Computer Science

The OOP module asked us to prepare a selection of UML diagrams both as individual pieces of work and a set of diagrams to support the summative assignments. Throughout my career, I have used a specific set of UML or related diagrams and upon starting this module I thought I had gained enough knowledge to support me. Going through this module has given me a clearer insight into the use of UML and how it applies to the varying levels of design steps or stages of the development cycle. Ore importantly and while it is a very interpretative area and abstract it has taught me how critical some if not all UML diagram types are applicable in the stages of development or a project.

When I started the course my experience of UML had been simply to use activity diagrams or state machine diagrams or in some cases a hybrid of the two to define either business or system processes or decision points. I quickly found how relative UML diagrams are to varying levels of design stages. From a professional perspective, only certain roles would typically use certain diagrams for example at the design stage and as stated I would typically use an activity diagram but then once a design is complete the design pack would be handed to a development team who would then extract the steps into class or sequence diagrams. One thing I found apparent is that UML is very abstract and that users of UML will quite often abstract further or perhaps not go into the level of detail that is standard on the UML model. This would quite often be based on the person's role, the company's development process or the standards or skill levels of individuals.

The course did teach me some of the fundamental concepts of UML and how it facilitates a much clearer understanding of system design. However, I do feel that as UML is a skill in its own right and is more applicable to business analysts and architects that a 12-week part-time course would perhaps not deliver a sufficient amount of knowledge or skill to be able to execute UML or in fact to the criteria or level required for this course. As stated previously UML is a very abstract and interpretive model or framework. The coding assignment we had to complete and utilise UML diagrams was also very high level and ambiguous, so I found it hard to separate the focus of delivering quality OOP code and UML at the same time and having a solid grasp on both.

I know I can deliver UML diagrams, but it is an area I need to investigate and learn further. Whilst I can rely on my professional experience to translate and surface certain diagrams for solutions and designs, I do need to take the feedback I have received on this module to further develop my understanding. I will however apply the additional knowledge I have received in what I feel are my stronger areas of knowledge to my professional work and therefore add value to the models, design and architectural approaches of my work.