COMP1035 – Lab 04: UML Diagrams II

Today's Task:

1. Develop personas and create a state diagram.
2. Design a low (or high) fidelity prototype.
3. Collaboratively produce markdown report about the diagrams and prototype.
4. Tag the group repository with a clear message upon completion.

Reminder: Software Brief

The university wants a new piece of software for handling module options and allowing students to sign up to optional modules. This includes students from other departments that might want to take the first year. Now, students must collect a form from student services, optionally attend an introduction lecture to a few options, write their choices on the form, and return it to student services. These choices are then checked and, by default, approved if they add up to 120 credits, evenly with a 60-60 credit split each semester, and all from their school and at the right year level. There are many cases, however, where additional approval is required. Some modules require approval from the module convenor, if for example, the student must have taken other pre-requisite optional modules before it. Also, if a student wishes to take a 50-70 credit split, they need approval from the Head of Teaching; this is usually dependent on how well the student is doing and whether they are likely going to be able to handle 70 credits in one semester. Further, if a student wishes to take an introductory module from another department, such as first year Japanese or Introduction to Economics, then the student needs approval from a) the module convenor of that module, and b) from the Head of Teaching in their own school. Approval from the module convenor is often based on limited class sizes, and on deciding whether the student has the pre-requisite learning necessary to understand the classes. This prevents a student from one school taking an advanced subject outside of their discipline, with the likelihood of failing it. Students wishing to take more advanced level foreign languages, for example, may need to provide evidence of language ability if they learned the language outside of the university. The student must gain all these approvals before submitting their forms to student services. These choices must be fed into other university software, two of which are: BlueCastle (student records) and timetabling (for room sizes).

A. Personas – Software of Your Choice [30 mins]

In the lecture, we discussed personas vs stakeholders – those personas are key concepts. You may have a few personas for one stakeholder type (e.g., students) or you may have one persona for a few stakeholder types (e.g., merging lecturer, convenor, etc.). You should also determine the most useful categories of information to include personas for this software, and then generate 4 to 6 personas, identifying which stakeholders they represent. You can find

specialised software if you wish, but PowerPoint/Word are perfectly reasonable tools for this task.

You should also write a short paragraph of reasoning for each persona somewhere (perhaps on the markdown report):

- Which persona need to be built, and which stakeholders they are for?
- What categories of data were important to put in the personas and why?

It is good practice to reason about the "key information" that a diagram or figure in a document is **trying to tell the reader**. You should consider including some explanatory text in all the diagrams or documents you produce.

B. State Diagram [30 mins]

The group should create a state machine diagram (in Visual Paradigm), e.g., application state (submitted, under review etc.), which conveys the possible states it can be and use the arrows to explain what causes the switching between those states.

C. Low-Fidelity (or High) Prototypes [30 mins]

Prototype acts as a bridge between your product's features and the customer's contentment. Prototyping allows you to gain early feedback as you're still in the development stage. Select two use cases from the use case diagram you produced in Lab-02, create a low fidelity prototype, mainly focusing on checking and testing the functionality rather than the visual appearance of the product. The prototypes should clearly refine the **FLOW** of the screen pages.

The group should also write a short paragraph of reasoning of the prototype:

1. Which of this function the use case refers to?
2. Why this function is important for validating with customer?

It is good practice to reason about the "key information" that a diagram or figure in a document is **trying to tell the reader**. You should consider including some explanatory text in all the diagrams or documents you produce.

D. Create Markdown(s) for Task A, B and C in the Group Repository [30 mins]

1. Like markdown task in Lab-03, create **additional markdown file(s)** for Lab-04.
    - The group should make images of your persona, diagram, and prototype above (screenshots or save them as images) and put them in the group repository.

    - REMEMBER to make sure they are **readable and clear**!
- In the markdown(s), make sure you have the following contents clearly visible:
    - What **aspect of the brief** do you think would benefit from a diagram or prototype?
    - **NOTE**: **Do not give bookwork answers** – focus on why the aspect you want to convey is suited to this type of diagram, and why not other types.

2. Your report should contain all relevant information like Task B in Lab-03:

3. Lastly, **do not forget to link this markdown to the main README.md**.

E. Tag Your Repository [5 mins]

Use the git tag process to tag the group repository. This should be the last thing the group do. The group should use a tag name, e.g., "Lab-04-v*X.X*" and tag message. If the group has multiple versions, remember to have the good **version history documented** in the README.md.

F. IMPORTANT: Peer-Review Your Team [5 mins]

Before **12 noon today**, go to Moodle and peers' review your teammates for their contribution to this week's lab work. Everyone **SHOULD** do this for Lab 04 and think carefully about what you are giving yourself and every person in the team.
