CO510 Group Work – Stage 2

Version 2018.01.19.01

Introduction

For this stage, and all future stages, you will continue to work in the group to which you were allocated in the Autumn term.

Important: The Workshop sessions on your timetable provide a time and location where you will be able to meet as a group and receive assistance, but **it is essential** that you do not limit your group's collaborations to just that one hour. You must allocate an amount of time each week that fits with the module's workload. This will be particular important in the first few weeks of the term as there are some other activities scheduled in the workshop sessions. Towards the end of the term, the unused lecture slots have also been provided as opportunities for you to hold group meetings.

A group area has been set up for you on the School's gitlab site and you should aim to take advantage of that for project tasks, such as sharing project documents, task allocation, issue tracking and code development. In particular, it offers features for change tracking and version control.

Work for this stage should be based on the material describing the user's requirements to be found only in the following sources:

- The presentation by the client, Miles Roman, in the lecture in week 7.
- The description of the Yuconz Employment Record System made available by the client, Miles Roman.
- Any clarifications and additions provided by the client or module convener.
- The use cases provided at the start of Stage 2.

Clarifying the requirements of Yuconz

At the start of this stage you have been provided with use cases for a subset of the overall system that is required by Yuconz. All groups should use these as the basis of further planning, design and implementation, rather than the ones they developed for Stage 1.

You should **not** assume that these use cases are accurate, complete or consistent. It is your responsibility, therefore, to examine them and ask clarifying questions in order to base your designs and future implementation on an accurate specification.

You may ask questions of Miles via David Barnes. Queries for Miles must be made via email to d.j.barnes@kent.ac.uk with the subject line: Miles Roman

David uses automated email processing and any subject line other than "Miles Roman" (no quotes) will be ignored and not forwarded to him.

You may ask technical questions, which Miles will do his best to answer by consulting his colleagues.

Stage 2 Requirements

For Stage 2 the focus is on planning and design. It will be important that you work out, as a group, how you hope to use the available time over the course of the term to complete the overall assignment. You must:

- Develop plans for how you will organise your group and schedule its effort over the coming term given the timetable of the assessment's stages;
- Develop plans for how you will use your group's gitlab space to coordinate your work and track issues as they arise;
- Develop plans for how you will ensure quality and consistency in everything the group does or produces;
- Analyse and refine the provided use cases;
- Start designing the system in UML from the revised use cases and other descriptions of the system.

It is envisaged that you will not start the coding during this period. Something to bear in mind as you consider coding at a later stage is that it will not be acceptable to submit source code in the form of one or more BlueJ projects.

Your planning documents should address at least the following areas. In all cases simply stating a toolset you are going to use is not sufficient. You must describe the processes involved. The name of each document you create should clearly reflect the purpose of the document; e.g., meeting-minutes, quality-plan, code-reviews, etc.

- Project management: how you are going to plan and manage your work and monitor progress, e.g., Iteration planning, meetings (daily "stand-up meetings", weekly meetings.) Communications methods, e.g., face-to-face, Skype, etc. Will you maintain a GANTT chart or similar? How will meetings and decisions arising from them be minuted?

  Group membership might change from stage to stage. Do your management plans cope with someone leaving the group, for instance?
- QA: How you will ensure consistency and quality assurance in project documents, including software.
- Reviews: how you are going to review documents/software and check and indicate that reviews have taken place. Pair programming, review meetings, etc. How will you review actual progress against planned progress. Will you review action lists from each meeting at the next meeting? How will you adjust plans and processes in the light of review outcomes?
- Version control: You must describe the processes you are using. It is insufficient just to mention a tool set, e.g. gitlab, Google Docs. You must state what functionality the tool provides you with and how that supports the management goals.
- Software tests, fault and bug reporting: The testing you plan on doing and how you will record the test results. The acceptance tests you will perform to demonstrate to the customer that you have delivered the required functionality. You will need to ensure that tests are directly aligned to the use

cases. Use your experience from writing user stories, and the feedback received on this from Stage 1, to guide you in the writing of these.

The UML diagrams and scenarios should be checked and correct/refined, where necessary and appropriate but this should only be done following consultation with Miles Roman to ensure that your understanding is correct. Class and sequence diagrams must be developed from the use cases.

Deliverables

Important: The priority of this stage is the planning documents. While you should certainly make a start on designing the class and sequence diagram elements of the UML model, the limited time available means that it is unlikely that you will be able to complete them. You will not be penalised if they are incomplete, but a reasonable start should have been made on them, nonetheless.

We expect you to use Papyrus to develop the model. All other written documents must be submitted as PDF documents for assessment. All documents must be uploaded via Moodle.

Each group will make a single, egoless submission. In other words, only one member of the group needs to make the submission.

1. The planning documents must be uploaded as a single archive file (.zip) called Documents.zip. This must contain only PDF documents. **The name of each document should clearly reflect the purpose of the document**; e.g., meeting-minutes.pdf, quality-plan.pdf, etc.

   If you have updated the use case scenarios, then include a PDF version of those in this archive.

2. You must submit your UML in the form of a single project as a single archive file (.zip) generated by Papyrus, using the file name Yuconz.zip.

   If you have updated the use case scenarios then they must be submitted as part of the previous item – Documents.zip.

   Before uploading the file into Moodle, check whether it works properly (for example, import the file into another open project in Papyrus) to make sure that the contents are not corrupted. It is your responsibility to make sure that the file is neither incomplete nor corrupted.

Deadline

08:55, Monday 5th February 2018 (week 16)

Weighting

10% of the coursework mark [May be subject to change]

David Barnes
Module convener
Document created: 2018.01.19