

# CO510 Group Work – Stage 3

Version 2018.02.04.01

## Introduction

For this stage you will continue to work in your current groups. In this stage, you must begin the implementation.

Work for this stage must 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 UML model provided at the start of Stage 3 or your own version developed with approval from the client or module convener during Stage 2.
- The scenario descriptions provided at the start of Stage 3.

If you have any uncertainty about the requirements you should ask for clarification.

## Group working

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.

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.

## The use cases

In the light of questions asked by groups during Stage 2, Yuconz has commissioned revised versions of the use case diagrams and scenario descriptions for you to work from. All groups (unless exempted) must use these for Stage 3 onwards. Groups exempted from this requirement are those who amended their Stage 2 diagrams and scenario descriptions following questions to Miles or David about possible errors or inaccuracies during Stage 2. Those groups may, if they wish, use their own versions that they amended during Stage 2.

As at Stage 2, you should **not** assume that these use cases and scenario descriptions are fully 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](mailto: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 should be careful not to ask technical development questions of Miles as he will not be able to answer them.

## Requirements of the final system

At the end of Stage 5 you will deliver a standalone Java application that implements the full functionality of the required system. The application must assume nothing about the existence of supporting software on the target system.

### Stage 3 requirements

For Stage 3 the focus is on baseline implementation of a subset of the elements of the final system. During this stage you may develop as much of the UML model and implementation of the final system as you wish. However, you are required to deliver UML class diagrams, UML sequence diagrams and a standalone Java application, with associated tests, that implements at least the use cases specified in the Authentication use case diagram supplied at the start of Stage 3.

In the light of the Stage 3 requirements and experience gained with the development as it progresses, you should continually review and update your planning and processes that you started at Stage 2. Continual review through experience and changing circumstances is a theme that runs through all stages of this group work.

### Authentication and authorisation

This section provides details to everyone of some of the clarifications Miles provided in response to group questions during Stage 2.

At some point, your system will obviously need to integrate with the existing computer system at Yuconz but the Director of Services Delivery will not allow integration until after the final acceptance testing. This will mean that you will need to 'mock up' authentication and authorisation procedures.

- Authentication enables a user to gain access to the system with a particular authorisation level. The idea is that a user's subsequent actions must be supported by the level of authorisation (credentials) they obtained when they logged in (i.e., were authenticated).
- Authorisation is the process of checking their permission to undertake a requested action, such as reading or modifying a document.
- **All authorisation attempts must be logged by the authorisation server, regardless of whether the authorisation succeeded or failed.**
- There are different levels of authorisation; for example, an HR Employee has rights to view any record; as do Directors, with the exception of personal details records. Individual users only have rights to view their own records, but they may view any record relating to them.
- A particular individual might have different levels of authorization available to them when they log in. For instance, an HR Employee might wish to access the system in the role of an individual user rather than an HR Employee. When accessing records, the rights must be checked against those obtained at the point the user logged into the system.
- A user must not be able to change their authorisation level within a single session. They must first log out and then re-authenticate.
- There is currently no requirement to record failed login attempts.
- There is currently no requirement to time-limit a login session.
- When a member of staff leaves, their login access to the system must be removed, but documents relating to them must not.

- Everyone in the company has a unique 6 character ID consisting of three lower-case alphabetic characters followed by three digits. IDs are not reused when a member of staff leaves.

## Deliverables

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

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

1. We expect your plans from Stage 2 to have required revision during this stage, so 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 must clearly reflect the purpose of the document;** e.g., minutes-template.pdf, quality-plan.pdf, coding-style.pdf, etc.

In contrast to deliverable 2 (below), the idea here is to deliver the plans rather than the implementation of the plans. For instance, you might have a template for meeting minutes, or a design for a GANTT chart. For this deliverable, submit those rather than implementations of those. The minutes of a particular meeting (that use the template) would be part of deliverable 2.

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

2. Evidence supporting the enactment of your plans. For instance, copies of meeting minutes, project planning charts, completed document reviews, 'Issues' lists, etc. Where you wish to refer to information that is not actually a document (e.g., 'Issues' lists in the gitlab repository), a document containing URLs for those documents is acceptable. However, all URLs must be to kent.ac.uk addresses. For instance, URLs to Google Docs are not acceptable.

Upload a single archive file (.zip) called Evidence.zip. This must contain only PDF documents. The name of each document must clearly reflect the purpose of the document.

3. 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 – Evidence.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.

4. Java source code fulfilling the requirements of at least the baseline implementation: the Authentication use case diagram. This must be submitted

in the form of a project that could be imported into a standard IDE, such as Eclipse or Netbeans. Note that BlueJ is **not** considered to be a suitable environment for developing the implementation and BlueJ projects will not be accepted.

The source code should include unit tests for your implementation. Where your implementation requires external files (e.g., a file of user information) these must be included. Upload a single archive file (.zip) containing the source code and any additional external files. Name the file to indicate the IDE you have used, e.g.: source-eclipse.zip, source-netbeans.zip, etc. Java class files are not required.

### Deadline

8:55am, Monday 19<sup>th</sup> February 2018 (week 18).

### Weighting

10% of the coursework mark.

David Barnes  
Module convener