**CS3307 Group Project - Stage #1**

**Fall Session 2017**

**Project Description**

This stage is part of the group project for CS3307.  A full description of the project [can be found here](https://owl.uwo.ca/access/content/group/df1797bc-56ba-4230-9065-cdbd30035a7d/ProjectSpecifications.html).

**Purpose of the Stage**

The general purpose of this assignment is to familiarize yourself with the requirements and scope of the group project for the course, carry out some initial design, and begin tracking your progress towards your project,  In particular this stage will give you experience in:

* developing user stories to capture requirements for a project
* creating UML diagrams for a large software project
* using Jira for simple agile task management

**Assigned**

Sunday, October 1, 2017 (please check the main [course website](http://owl.uwo.ca/) regularly for any updates or revisions)

**Due**

This stage is due Tuesday, October 24, 2017 by 11:55pm (midnight-ish) through an electronic submission through the [OWL site](http://owl.uwo.ca/). If you require assistance, help is available online through [OWL](http://owl.uwo.ca/).

**Late Penalty**

Late assignments will be accepted for up to two days after the due date, with weekends counting as a single day; the late penalty is 20% of the available marks per day. Lateness is based on the time the assignment is submitted.

**Group Effort**

This stage of the project is expected to be a group effort, with each member of the group contributing equally in a reasonable fashion.   Feel free to discuss ideas with other groups in the class; however, your submission must be the work your own group. If it is determined that you are guilty of cheating on the assignment, you could receive a grade of zero, and you may be penalized further by the value of the assignment. That is: this assignment is worth 10% of the overall mark; if you cheat, you could receive a grade of -20%, and your maximum possible mark in the course would be 80%.

**What to Hand in**

Your stage submission, as noted above, will be electronically through [OWL](http://owl.uwo.ca/).  You are to submit all relevant documentation as discussed below.  Only one submission per group is necessary.  (As groups have yet to be fully formed, the group submission aspect of this stage of the project has not been set up in OWL.  This will be done in the near future.)

**Stage Task**

In this stage, you will get started on your projects by writing, estimating, and prioritizing user stories for the requirements of the project.  User stories are commonly used in agile software development methodologies to capture the requirements of a system in short, easy to digest stories rather than the lengthy requirements document (often referred to as "Big Design Up Front") common to many other software development models.  User stories encourage a dialogue between teams and customers to help ensure that teams know the requirements of the system, can easily fit their heads around all the requirements, and can plan and prioritize tasks effectively.

You will also develop a detailed UML class diagram for the classes you expect to need, including attributes and behaviours.  The intent is not to develop a static diagram that can never change, but instead to capture as many of the classes that you expect to need in developing the project so that development iterations can begin quickly.

Finally, your team will create some initial tasks in the project management system we will be using for the course, [Jira](http://www.atlassian.com/software/jira).  This will help you organize and manage your projects, and will give us an excellent tool to keep tabs on your progress over the course of your project.

**User Stories**

In the first part of this stage of your project, you will create user stories for each required feature in the project to assist you in:

* ensuring that you understand the features
* planning and prioritizing the development of each feature
* testing your features once they are implemented to ensure they meet the requirements of the project.

If you have already conceptualized and scoped out additional features for the project, now is a good time to create user stories for them as well, as this will give you more time to obtain feedback and develop these aspects of your project.  So, for each feature, you will need to create:

* The user story itself.  These should be short and simple, and can follow a number of different templates.  (For example, "As a *<role>*, I can *<action with system>* so that *<external benefit>*."  From this I can create a story like "As an OWL instructor, I can create assignments so that students will have work to do to keep them busy."  Not exactly true, but you get the point ...)
* A list of acceptance tests that can be used to verify that the story was developed to meet expectations and has been completed.
* An estimate of the length of the story in story points (a unit that's used to compare the size and complexity of the story relative to other stories; for example, a story estimated at four points is expected to take twice as long as a story estimated at two points).

User stories are intended to be short and to the point.  You should be able to hammer out all the user stories for the project in an evening or two of work.  Get together as a team, go through the [project specification](http://owl.uwo.ca/access/content/group/df1797bc-56ba-4230-9065-cdbd30035a7d/ProjectSpecifications.html), brainstorm the stories, write them up, discuss them, rinse and repeat.

In addition to encoding your stories as tasks in Jira (see below), you should also submit your stories in a PDF or Word document for the TAs to be able to read and process.  You may format this document as a series of cards to follow the user story process, but you do not need to actually submit a series of paper cards for this stage.

**UML Class Diagram**

Create a UML class diagram resulting from your analysis of the [project specification](http://owl.uwo.ca/access/content/group/df1797bc-56ba-4230-9065-cdbd30035a7d/ProjectSpecifications.html).  Your diagram must have:

* Sensible classes
* Attributes, along with their types and visibility
* Methods, along with their parameters, parameter types, return types, and visibility
* Associations, hierarchies, and so on.

The goal here is to capture the key requirements of the project in as complete a manner as possible.  We do not expect that you will know every single method or attribute you might need up front, but it should be clear from your diagram that you have carefully thought through the design of the project and you should be able to sit down after creating the class diagram and be able to start implementing your stories.

You are allowed to use whatever diagramming tool you wish to construct your diagram.  You may submit a source/editable version of the diagram if you like, but you must at least submit a portable PDF version of things so that the TAs can grade things independently of the tools use to create the diagrams in the first place.

**Project Tasks**

Once project groups have been fully formed, each group will have a project and board created for them in our Jira installation.  [Jira](https://www.atlassian.com/software/jira), among other things, is a solid project management tool that is often used in industry.  For CS3307, we will be using a simple [Kanban board](https://www.atlassian.com/agile/kanban) for creating, managing, and tracking the completion of tasks.

For each user story created above, you will create a task in Jira to capture and describe what needs to be done to complete the task (and, thus, the story).  You should also at this point decide which team member will be tackling each task and assign each task out in this fashion.

As you move forward with the project, tasks should be updated and moved along through the board.  Jira can be a powerful tool for you to keep track of your project and to ensure things are on track.  (Keep in mind that for your the next project stage, you should have a minimum of 30% of your user stories completed!)  We will also check in on your progress through Jira to see how things are moving along, so please keep this as current as possible.