# *1Systems II (420-E21-HR)*

# *Final Comprehensive Assignment*

Date assigned: Tuesday, April 4, 2017

Date due: **Wednesday, April 19, 2017 11:50pm**

**Learning Objectives**

Upon successful completion of this assignment, the student will be able to:

* Select an appropriate architecture for an application system
* Select an appropriate system development methodology
* Demonstrate a working knowledge of agile methodology
* Demonstrate analysis of a problem domain and capturing requirements in use case and activity diagrams.
* Demonstrate the creation of use case diagrams, activity diagrams, class diagrams, sequence diagrams and deployment diagrams from system description

To do:

Create a new document named **YourUserName\_E21\_Final Comprehensive Assignment.docx** in your 420-E21 folderin your home drive.

Add a title page to the document, with the course name, assignment name (Final Comprehensive Assignment), your name, the semester, and the date. For each of the questions in the assignment, create a section heading with an appropriate title in the document.

All diagrams are to be in UML syntax.

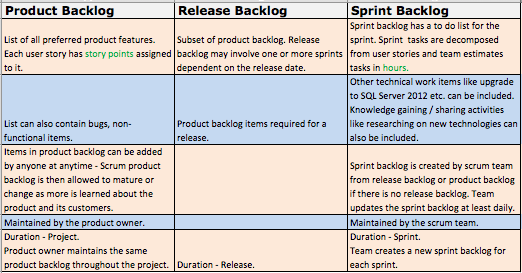
**Part A: System Design**

You are designing a system to help manage agile projects.

Your key systems components/items include: the product backlog, release backlog and a sprint backlog. You may include other components if you see a need for it.

Your key actors must include the Product owner and sprint team. You may include other actors if you see a need for it.

You may need to do research in agile methodologies to complete your modeling and design.



1. Draw a use case diagram showing the abilities of the various actors on the system components. Your use case diagram must cover the actions from project inception to delivery (whole SDLC) and covers who should be able to read, prioritize or update items from any of the backlogs.
2. Create an Activity diagram with swim lanes showing the “Release planning activity” given a Product Backlog. Assume that the Product Backlog is not initially prioritized. The output of this effort is a Release plan with 1 or more sprints contained within the Release.
3. Create an Activity diagram with swim lanes showing the “Sprint activity” given a Release backlog that has been prioritized. (Note that this is everything from beginning to end of the sprint. Sprints are a fixed 2 week duration)
4. Based on your analysis, develop a class diagram, including associations and multiplicities. (You may choose to go through the CRC exercise to determine your classes, but you just need to show me your final Class diagram for this assignment).
5. Create a sequence diagram using the classes (defined in your class diagram) for the “Sprint activity”.
6. Assuming you have a sprint goal of completing the “Sprint Activity”, put together a sprint backlog for the sprint. For coding efforts, show the class.method() on the sprint backlog. Assume the project is up and running and no project setup tasks are required in your estimate.

**Part B: System Architecture**

Heritage College will be hosting a provincial conference for CEGEP teachers and educational administrators next year. In order to support the organization of the conference, they have hired you to develop an event registration system. The college uses .NET, and SQL Server as the standard environment for systems development. They have a web server and database server that run on Windows Server 2008 and a mail server which runs on Windows Server 2011. The web server uses IIS, which requires the .NET framework 4.0, the database server runs SQL Server 2008, and the mail server uses Microsoft Exchange Server.

The user requirements are as follows:

* Administrators can set up the events and schedules
* Educators can view events by topic, author, and time
* Educators can register for events online
* Educators can check their registration status
* Administrators can send email notifications to all registrants
* Administrators can print name tags for registrants
* Administrators can check in registrants at the event
* Educators can view an interactive floor plan
* Educators can view session schedules
* Educators can view session information
* Educators can interact with other educators in attendance

1. What type of architecture should the application use (client/server, two-tier, three-tier, n-tier, etc.)? Justify your answer.

This system would be designed in a three-tier architecture. The system has 3 distinct tiers –the client, the web server and the database server. Those are the parts of the system that will be involved in this system, while the Microsoft Exchange server exists for email, it’s not actually a part of the system. It will get used by the system administrators, but it’s another system that’s separate from the rest of the actual event registration system.

1. Briefly describe how the application should be layered and give an example of what would be in each layer.
2. Draw a deployment diagram for the application.

**Part C: System Development Methodology**

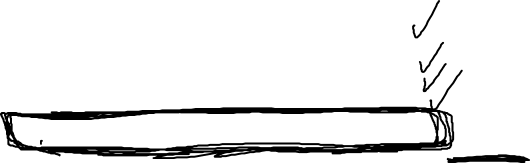
1. An organization works on in-house projects for the US and Canada. The team isn't fluent with any specific project management approach; however the team is passionate with a coach who would help them. The project requirements are well-defined due to the military RFP (Request For Proposal) process which requires them to fully specify the black box requirements prior to requesting proposals. What system development methodology should the organisation use and why? Elaborate on each factor you think leads to the selection of the system development methodology and justify your answer.

**To submit**

When you have completed the assignment, upload it to the Moodle page for this course.

Add a table of contents to the document.

**Marking Scheme**



|  |  |
| --- | --- |
|  | **Out Of** |
| Assignment Document |  |
| Cover page & table of contents | 2 |
| **Part A: System Design** | 0 |
| Use Case Diagram | 5 |
| Release Planning Activity Diagram | 5 |
| Sprint Activity Diagram | 5 |
| Class Diagram | 10 |
| Sequence Diagram | 10 |
| Sprint sizing | 5 |
| **Part B: System Architecture** | 0 |
| Architecture Choice | 4 |
| Layers | 5 |
| Deployment Diagram | 7 |
| **Part C: System Development Methodology** | 0 |
| System Development methodology choice | 4 |
| Proper use of English, handed in properly | 5 |
| **Total** | 67 |