Team Project - Sprint 1

Due: Friday, June 14 @ 23:59

The main goal of this deliverable is to give you a chance to:

1. Get you to build the first presentable version of your prototype (i.e. Demonstrate working software)
2. Require you to prioritize tasks and make product decisions (the deadline is fixed, you get to choose what to build by the deadline)
3. Encourage you to reflect on your team (and individual) work habits

Deliverables

The following files should be submitted to your team repo in a folder called `doc/sprint1` (You will need to create this folder): Note that they're are best viewed in Markdown:
The following files should be submitted to your team repo,

1. Project-planning
    1. RPM.md - The release planning meeting document RPM.md, document the Release Planning Meeting. This document has to indicate clearly the release goal, the scope of the project (at least in terms of epics/key features) and the participants.
    2. iteration-01.plan.md - Summarize your first planning meeting
2. sprint-01.review.md -Summarize your first sprint review meeting.

STANDUPS

Standups are a way for you to communicate with your team what you've worked on, what you will be working on and any blockers you encountered.

For your project, standups/team updates are done on the CSCC01 Slack server within your team's #standup channel.

Each standup should follow this format:
[Standup Date] - Sprint # Standup #

1. What did you work on since the last standup?
2. What do you commit to next?
3. When do you think you'll be done?
4. Do you have any blockers?

Each team is required to post a minimum of 5 standups per sprint and ALL standup updates must answer the necessary questions and are of good quality.

We encourage you to do more than five to ensure team communication and to resolve blockers. In addition, they should be spaced out

SYSTEM DESIGN DOCUMENT

The System Design document must be prepared in a format that we can read (PDF, MS-Word, md, html) and it must be stored in the same folder doc/sprint1.

You will likely learn and use a new technology in the course of the project. It is therefore likely that the system design you provide here will undergo major changes as you work on the project. Don't panic! Get together and spend some time brainstorming. Use your TA to receive feedback

- Include a high-level description of your classes using CRC Cards: what they are, what their responsibilities are, and what is the interaction between them.

- You can use the following template for the CRC Cards:

<table>
<tr><td colspan="2">Class Name</td></tr>
<tr><td colspan="2">Parent Class(if any): List the parent class if applicable
Subclass (if any): List all the subclasses separated by comma</td></tr>
<tr><td>Responsibilities
• <i>List of responsibilities</i></td><td>Collaborators:
• <i>List all your collaborator</i></td></tr>
</table>

- The description of system interaction with the environment should indicate any dependencies or assumptions made about the operating environment of the system. E.g. OS, programming language compilers and virtual machine, DB's, network configuration, etc.

- Describe the architecture of the system that is the most abstract view of how your system is divided into components and how those components are interconnected. The architecture should be described with a diagram showing components and how they are related (or equivalent in words). Beware of designs based on a large number of components, they may signal a design that is overly complex.

- The system decomposition should relate the system architecture to the detailed design, to identify the role of each component in the higher-level architectural view. Description of strategy for dealing with errors and exceptional cases (e.g. invalid user input, network or external system failure) that might arise in the use of the software. For anticipated errors and exceptions, a summary of how the software will respond in these situations.

The following tools may be useful in generating your software architecture diagram:

- Draw.io/mxGraph - http://www.draw.io
- LovelyCharts - http://my.lovelycharts.com/

- Draw Anywhere - http://www.drawanywhere.com/
- Creately - http://creately.com
- Diagrammr - http://www.diagrammr.com/
- Grapholite - http://grapholite.com/
- LucidChart - http://www.lucidchart.com

As with the previous deliverable, we created starter templates for you:

- Iteration-01.plan.md
- sprint-01.review.md

DEMO WITH THE TA

Please make sure to stay within the scope identified during the planning meeting. You should be able to demo your software at the end of sprint 1.

During the demo, all team members must be present. The TA will mark the attendance for everyone on the team and ask you to show your working software. The TA should be able to use the software to the extent of the feature(s) you have already installed. You should demo **at least two features** to your TA to receive full marks.

Afterwards, you should conduct your Sprint 1 retrospective meeting. Appoint a note taker and document your observations about sprint 1 in a document named sprint-01.review.md (template provided above) in doc/sprint1.

Marking Evaluation

Sprint 1 is worth 18% of the final project mark. Please review the marking scheme for the full mark breakdown and items that are needed for submission.

Additionally, there will be a peer evaluation submission on Quercus. The peer evaluation must be submitted before 11:59pm on the last day of your sprint (when the sprint is due).