Project Sprint 1

I hope everyone has fun and gains a lot of valuable experience. The Product Owners, TAs and I are here to help you achieve both of these goals. In class we talked about how an Agile team works on a software development project: the process, the roles, etc. The project is where you put all this knowledge in practice. This document outlines how your work will be assessed and what is expected from Sprint 1 of your project.

Project Planning

Include all aspects of your project plan. Among other artifacts, we will definitely want to see:

- The release planning meeting document `RPM.md`
- The product backlog `PB.md`. You should take the copy from sprint 0, reflect any changes suggested during release planning meeting, (if no changes it is OK to keep the `PB.md` you already have). Place the updated `PB.md` under `doc/sprint1`.
- Once you complete the Release Planning Meeting, you should also do the Sprint 1 Planning Meeting. Those meetings must be held during the tutorial Tuesday/Thursday, Oct 8/10. You will have a chance to discuss with your TAs as well, who will play the role of the Scrum Master. Here it is what we are looking for:
  - Documentation: the meeting minutes must be written in a file named `sprint1.md` under `doc/sprint1` folder.
  - Participation: Everyone must participate. Record the participants in `sprint1.md`.
  - Sprint Backlog: document your sprint backlog as a section of `sprint1.md`. As a reminder, your sprint backlog must include user stories you intend to implement during sprint 1.
  - Break down each user story into tasks and do assign the tasks to team members.

Project Tracking using Jira

- Chose a numerical series to be used for recording the point estimation of your user stories.
- Make sure all your stories are prioritized and estimated.
- Prioritizing is simply done (in Jira) by moving them up and down.
- Create your sprint 1. Set the sprint starting date to Oct 8, 2019 and Sprint end date to Oct 22, 2019.
- Make sure to estimate your stories and record story points on Jira.
- Start your sprint 1.
- Make sure to use the story ID in your git commit messages so you get your activities added to your stories.
- The tasks that you have listed in `sprint1.md` should also be recorded on your tracker and of course the name of the person who records the task must match the name of the person to whom the task has been assigned.

System Design

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 are probably not seasoned developers (yet!), and 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. You will also receive feedback on this from me and the TA.

- 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:

```
Class name: Classname
Parent class (if any):
Classname Subclasses (if any): List all the subclasses separated by a comma.
Responsibilities:
*g
*h
*i
Collaborators:
*j
*k
*l
```

- The description of system interaction with 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 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.

Inteview with the TA

Please make sure to implement at least three major features of your project. You should be able to demo your software end of sprint 1, Oct. 22 (Oct 24) , 2019 during your tutorial. By Oct 22, 10am, you must have already commited to github the required documents and finished (on the tracker) the stories you are demo-ing.
During the demo, all the team must be present. The TA will mark the attendance for everyone on the team and ask you to show (no more than 3-5 min!) your working software. The TA should be able to use the software to the extent of the feature(s) you have already installed.
Once the other teams present to the TA, you should conduct your Sprint 1 retrospective meeting. Appoint a scribe and document your observations about sprint 1 in a document named `SR1.md` in `doc/sprint1`. You should record:

- the participants in the meeting

- unfinished tasks and group them into stories; add them to `SR1.md` in the form of new user stories. Update your `PB.md` and save the updated copy in `doc/sprint2` (yes, sprint2 - prepare for next sprint!)
- what are practices that you should continue during next sprint
- what are some new practices that you might want to use during next sprint
- what are (if any) harmful practices you should stop using during next sprint
- what was your best/worst experience during sprint 1

Marking (9% of your final mark, see below for the breakdown)

Team work

This deliverable is worth 8% of your final mark. Your work will be graded based on the following criteria:

- 5 marks Planning Meetings (as documented in `RPM.md` and `sprint1.md` - namely - do your documents include all requred elements listed above?).
- 10 marks for correctly formulated user stories, correct recording on Pivotal Tracker of all user story elements, and break down of stories into tasks. The breakdown will be marked based on `sprint1.md` and Pivotal Tracker records..
- 10 marks for correct tracking on Pivotal Tracker. Namely - are all your stories record on the tracker? Did you start all stories for sprint one did you finish them? Do the commits on github match the tasks listed in `sprint1.md`? Is the commit dome by the person to whom the task ahs been assigned to?
- 10 marks for System Design. You may want to generate a PDF file for this that explains your CRC Cards and your software architecture diagram. Make sure your report looks professional. Do not forget to include a cover page and a table of contents. Think of a user who chooses to read your report in electronic format — include links where appropriate. Following tools may be useful in generating your software architecture diagram:

```
1. Draw.io/mxGraph - http://www.draw.io
2. LovelyCharts - http://my.lovelycharts.com/
3. Cacoo - http://cacoo.com
4. Draw Anywhere - http://www.drawanywhere.com/
5. Creately - http://creately.com
6. Diagrammr - http://www.diagrammr.com/
7. Grapholite - http://grapholite.com/
8. Gliffy - http://www.gliffy.com
9. LucidChart - http://www.lucidchart.com
```

- 10 marks if your demo works without failures for the user stories marked as finished in the tracker. The TA will check your feature, try it on your laptop, and mark you on the spot on this item.
- Finally, your sprint amrk will be scaled according to the peer evaluation percent assigned to you by your team mates. Please see next item for the details of computation.

Peer Evaluation

- Peer evaluation: please submit on blackboard as required in Peer Evaluation document published on the course web site. The completion of Peer Evaluation docuemnt is worth 1% of your final mark.

- Usage of peer evaluation to scale your mark is done as follows. Each team member will evaluate you using the peer evaluation form based on 6 criteria, for a total of 30 points. The number of points you receive from each peer, divided by 30, and multiplied by 100, will be converted to a percentage. We will average all percentages assignned to you and use this percentage to scale your mark.
- Keep in mind you are not allowed to ask who assigned you what number. Those numbers are strictly confidential. No exceptions.