SYSC 4806 Project Rules

The objective of this project is to apply the engineering methods and design principles taught in this course, towards the development of a Web Application, using the Spring MVC framework or something your team decided to use. If your team decides that Spring is not preferred, your team needs to meet with the instructor to discuss for the rationale behind the decision.

Timeline

There will be three 2-week "**sprints**" where the project will be incrementally built. After each sprint each group must meet with the TA during the lab hours for a short demo (less than 15 min) describing their progress. **If your team cannot meet as a group during lab hours due to conflict, your team needs to contact the TA to schedule a meeting.**

Requirements

Project Setup

Your project should have a repository on **Github**, integrated with CircleCI (or other hosted CI), and be up and running in production on **Heroku** from the start of the project. Please add your TA and your instructor to you project. How? Your TA's github ID can be found on Brightspace under General Info >> Lab Schedules - "Please find your TA based on your session info and your lastname". Your instructor's github id is sthsb.

In the root folder of your project there should be a pom.xml file. The TA should be able to clone the project, and use the pom.xml to compile, test, package, deploy and run the application. The pom.xml file is not required if your group does not use Java; in that case, you need to provide detail instructions on how to launch your app locally.

Development and release process

The project must have a master **branch**, and each contributor should create a new branch for each of the features they implement. Once development is complete, the team member should open **a pull request** and request **code reviews** by at least one other team member. After the reviewer(s) approves the code change, it can be merged into master, re-tested and deployed to Heroku. The TAs are instructed to check for this!

Scrums

Each team member is required to communicate weekly updates following the principles of the **daily scrum meetings**. The purpose of daily scrums is to communicate within the project team. This **communication should also be visible to the TA**, and therefore you will use **GitHub Issues**[1]. Each week a group member should open **a new issue called Weekly scrum – [date]**, then every team member should add comments with their own contribution, i.e. their answers to the questions:

- what have I done this week? (with link(s) to the relevant GitHub Issue(s))

[1] To enable issues in your GitHub repo: go to the "settings" tab of your repo (top right), then check the "issues" checkbox under "Features"

- what will I do next week? (with link(s) to the relevant GitHub Issue(s))

- what is holding me back? (if applicable)

Expected length: 1-2 sentences per item.

Product backlog

Use GitHub Projects ("Projects" tab in your GitHub repo) to create a "Kanban" style view of your Issues, with columns dedicated to "backlog", "in progress", "completed". The README.md file on Github should summarize the current state of the project as per the Kanban, and include a plan for the next sprint. It should also include the **up-to-date schema of the database**[2].

Milestones

- Milestone 1: Early prototype. Give a 10-15 minute demo during the meeting with your TA in the Week of **March 7th**. For this milestone we are looking to see enough functionality to get a feel for the system and how it will work. One important use case should be operational. It should collect data from the back end, do something with it and display the result. The display does not need to be fancy. There should be a GitHub repo, integrated with CircleCI (or other hosted CI), and the app should be up and running on Heroku. We will also inspect the README file, the Issues, the Kanban, the code reviews, the tests, and we will verify that everybody is participating in all aspects of the project (if that is not the case, different team members will end up with different grades).
- Milestone 2: Alpha Release. Give a 10-15 minute demo during the meeting with your TA in the week of **March 21st**. For the alpha release your system should be somewhat usable, although not feature-complete. This means that a user should be able to use several related features of the app and do something reasonably useful. The README on GitHub must be updated with a plan for the next sprint.
- Milestone 3 - Final demo. Project complete. Public demo during the meeting with your TA in the week of **April 4th**. For the final sprint of your project, you must decide on the final scope of the product: a set of features that can be implemented within the given timeline and makes the product usable and useful. The user interface should not have any dangling links to non-implemented features.

Project report: No project report is required.

The information on the GitHub README should provide sufficient information to understand what the project is and how much is implemented. Include a UML class diagram of your Models (and only of the Models!), as well as the corresponding database schema created by the ORM (observe what patterns are being used by the ORM behind the scenes! **You need to know these as preparation for the final exam too…)**. Your diagrams must be in sync with the code that you have produced, so just put these diagrams in version control, and grow/update them along with your code!

[2] IntelliJ can do this for you (https://www.jetbrains.com/help/idea/creating-diagrams.html), once you have an actual persistent database set up. Otherwise you may have to draw it by hand.

Evaluation

The percentages of each milestone of the overall project grade are:

- Milestone 1: 20%
- Milestone 2: 30%
- Milestone 3: 50%

The team and individuals will be judged as follows:

Process, agile practices (60%)

- Proper use of version control, continuous integration, code reviews.
- Everyone in the team is involved in the entire process. Activities causing **silos** will be punished. Examples of such no-no activities are:
    - Team member A takes care of frontend. Team member B takes care of backend. And so on.
    - Most of the development work is only done by a portion of the team.

  Ultimately, we want everyone to have a mind set to work in **cross-functional** teams.

- Tests: unit tests, integration tests

- Deployment frequency: at minimum, 2-3 deployments per week. Every team member should be deploying.

- Information on GitHub README.

- Kanban, Issues, Weekly scrum contributions.

Features (40%)

Status of the project at each milestone, and quality of what has been deployed. The TA will evaluate whether the project **makes reasonable progress**. Since every project is different, we cannot specify a list of expected features to be implemented, or lines of code to be written.

Appendix

Terms used here

1. Sprint: A sprint is a short, time-boxed period when a scrum team works to complete a set amount of work. https://www.atlassian.com/agile/scrum/sprints
2. daily scrum meetings: usually called standup meetings. It needs to be short and quick. https://www.atlassian.com/agile/scrum/standups
3. Product backlog: basically a prioritized todo list. Read more from https://www.atlassian.com/agile/scrum/backlogs
4. More reading about Scrum: https://www.atlassian.com/agile/scrum