**SOEN 6011**

**Software Engineering Process**

**Project Title: Career Services Application**

**Due date: July 19, 2023**

The following are the categories of main users:

1. Employer – create an account/register, login, browse candidates, add job offers, select candidates that have already applied to an offer for an interview, track the post
2. Student/Candidate – create an account/register, login, and build/upload a resume; browse and apply for job offers; track applications
3. Admin – Managing all user profiles, postings and tracking

The following are the 4 core features:

* Employer – browse candidates, add job offers, select candidates that have already applied to an offer for an interview
* Student/Candidate – create profile, and build/upload a resume; browse and apply for job offers
* Tracking: Track the post by employers, track application by candidates
* Admin – Managing all user profiles, postings and tracking

The following table summarizes the sprint2 work:

***Sprint 2 Requirements:***

|  |  |  |
| --- | --- | --- |
| Activities | Sprint Details | Weight |
| **Sprint 2:**   1. README   file  submission  2. Task Breakdown  3. Commits of  all  members | **Sprint 2 Deliverables that can be tracked on your GitHub:**   * README file with updated Info if necessary   1. Description of the Project   2. Team Members and roles   3. Project Approach and Technology * Document(s) with   1. User stories backlog for Sprint3   2. Task Breakdown (details of the stories with team members responsibility, if known, for the current sprint).   3. Open issues or plan for next Sprint      * Meetings Minutes file (minutes files should be named   <teamName\_Sprint#\_meetingnumber\_meetingdate>   * Implementation of 2 core features | 5% |
| All members of the team should contribute to at least one commit or initiate requests and issues. These should be clearly traceable on Github. By the end of the project, the instructor and the markers should be able to trace the work of every member of the team. |
|  | | |

# Grading Scheme

|  |  |
| --- | --- |
| **Process general** | Total=100 |
| Done enough work (end to end of the system complete and working) | 10 points: Work on 2 of the 4 core features  Note: you need to finish one core feature and have major progress on the second feature (needs to be working with some minor details)  4: if you only have progress on one core feature  8: if you have implemented one core feature and there is no work done for the second |
| Story description, Points, Risks, Priority | 10 points: Create issues about user story details and allocate story points to the project schedule. Include risk management process (impact, probability and counter measures) 4~8: The content is unclear / vague/ missing elements |
| Frequent meetings (scrum) with meeting minutes available | Do you often have discussion or meetings? Did you document your meeting minutes? 10 points: Document your meeting at the wiki page (recommended) of your repo or under the issues. (at least one meeting) 4~8: The content is unclear / vague/ missing elements. |
| Task breakdown | 10 points: Allocate points to tasks as per user story breakdown.  4~8: If you missed the user stories/tasks for some features/ if the stories/task breakdown are unclear. |
| Finish the tasks in the current sprint on time and plan for next sprint | 10 points: Finish the tasks before the DDL and  create some issues (a few sub user story of core features/story cards) under the milestone of the next sprint (you can still commit after the DDL if the changes you do are beyond the requirement of this sprint) -(4~8): If some tasks were just finished after the DDL  -(4~8): If you don't have issues under the next milestone or they are unclear |
|  |  |
| **Repo** |  |
| Commit logs descriptive | Meaningful commits description 10 points: Most of the commit messages are meaningful and correctly describe the commit 4~8: Several meaningless commit messages (e.g., asdasd; aaaaa; 123) 0: Most (more than 3) of the commit message are meaningless |
| Referenced with issue | commits referenced with issue numbers 10 points: Have at least one commit referenced with an issue 4~8: Have commits referenced with issues, but they are not related. |
| Discussion / reviews on github items | 10 points: Do pull requests & review them. 0: Have pull requests but no review |
| Work assigned to individuals | 10 points: Assign some pull request / issue to members to work on them  & **everyone should participate in the contributions** (i.e., we are able to see the contribution somewhere in your repo) If no assignees, **-2 from the group marks** If someone didn't contribute to the repo, **-2** **from the mark of that member.** |
| Team work | 10 points |

# Reference

1. <https://www.linkedin.com/learning/learning-git-and-github-14213624>
2. <https://docs.github.com/en/get-started/quickstart/hello-world>