**SOEN 6011**

**Software Engineering Process**

**Project Title: Career Services Application**

**Due date: August 9, 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 sprint 4 work:

***Sprint 4 Requirements:***

|  |  |  |
| --- | --- | --- |
| Activities | Sprint Details | Weight |
| **Sprint 4:**  1. Complete Testing  2. Complete organization of Project  3. Complete all required Features  4. Commits of  all  members | **Sprint 3 Deliverables that can be tracked on your GitHub:**   * Document(s) with   1. Complete all User stories for Sprint4.   2. Finalize Project and Prepare for presentation!   3. Working Continuous integration, Setup test environment and working tests      * Meetings Minutes file (minutes files should be named   <teamName\_Sprint#\_meetingnumber\_meetingdate>   * Implementation of all required features * Complete testing required for Sprint 4 | 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: Finished all the core features and additional features & they all work well in your environment.  0-6: If some features don't work well. |
| Story description, Points, Risks, Priority | 10 points: Document the issues about the additional features with risk & priority & story points for user stories  (Label them or write them inside the issue to indicate or use wiki to document them)  4~8: The content is unclear / vague/ missing elements. If you missed the user stories/tasks for some features/ if the stories/task breakdown are unclear. |
| Frequent meetings (scrum) with meeting minutes available | Do you often have discussion or meetings? Did you document your meeting minutes?  10 points: Final meeting minutes update  4~8: The content is unclear / vague/ missing elements. |
| UAT | 10 points: UATs written for all features (core + additional).  4~8: No UAT’s or Missing for some features |
| Continuous Integration | 10 points: working test environment for continuous integration and working tests  4~8: No initial Setup of Continuous Integration |
| Unit Testing | 10 points: Unit Testing is done for some modules and features  4~8: No unit testing found or hardly any tests written |
|  |  |
| **Repo** |  |
| Commit logs descriptive and referenced with issue and pull requests review | Meaningful commits description  10 points: Most of the commit messages are meaningful and correctly describe the commit. Have at least one commit referenced with an issue. Do pull requests & review them.  4~8: Several meaningless commit messages (e.g., asdasd; aaaaa; 123) with their respective pull requests  0: Most (more than 3) of the commit message are meaningless. Have commits referenced with issues, but they are not related. Have pull requests but no review. |
| Code Quality | 10: Well organized and high quality code with comments  8: Generally good  0-6: Spotted multiple problems with quality of code |
| Directory and package structure | 10 points: Files with similar functionality are organized in packages.  -(4~8): Packages contain some files that are not related to the same functionality  -(4~8): No or very bad package structures |
| Teamwork | 10 points |