

## **SOEN 6011**

### **Software Engineering Process**

#### **Project Title: Career Services Application**

**Due date: July 12, 2023**

The objective of this project is to provide a career services platform where the students (people looking for jobs) will be able to post and update their portfolio and the employers should be able to post jobs. With the help of this, you can carry out a job head hunting in your own comfort. This benefits job-seeking candidates and employers to shortlist their potential job and employee respectively. This can also help employers find specific candidates as per their requirements along with job seekers being able to tailor their profile as per the market. All you require is a PC or a mobile device with Internet to access the website online. To get to this online career services system all the Students and Employers will create an account and use their email and password to login. Students will create/modify their profile and will view and apply to job postings. Employers will create and manage their job posts, review applications from candidates. The login credentials for this system are under high security and nobody will have the capacity to crack it easily. Employers will receive notifications when students apply to their job offers, and students when they are called for interview.

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

Apart from these primary users mentioned in the description, other kinds of users could be considered. Additional marks will be given for teams who are thinking out of the box and include other users and functionalities for eventual implementation when properly justified.

This project will help you to get a taste of software project management skills first hand. You will follow the Agile development approach; take advantage of GitHub distributed version control plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis to support your project management process. Because of the short span of this project, you are not expected to deliver a marketable product, but the result should be at least a compelling middle-fidelity prototype that could serve as the basis for defining a real product. Check these two links on prototypes fidelity quite helpful: A Guide to Prototype Fidelity:

<https://www.webfx.com/blog/web-design/design-mockup-fidelity/>

<https://www.webfx.com/blog/web-design/wireframes-vs-prototypes-difference/> ).

The duration of the project is 6 weeks; the development process is an adapted Agile with 1.5 to 2.5 weeks long iterations, 4 iterations in total. The first sprint is for training and setting up your development environment.

Please create a public repository in GitHub named <team\_name-Soen6011summer2023> with all team's deliverables in this project (organized by subfolders: Sprint 1, Sprint 2, Sprint 3, and Sprint 4). This repository is going to have all documents related to your work such as README file, project reports, meeting minutes respectively, user stories, etc.. You can use the Wiki or any other GitHub feature, to organize your documents.

- The README file is supposed to show the title of the project, its description, the members of the team and the technology to be used (look at the details in the grading scheme attached).
- The plan for the next sprint (or open issues) can be written as an "issue" on GitHub (Open issue for next sprint and can be of any kind)
- Record all meetings you hold for the project and post their "minutes" in the corresponding folder.
- The report for sprint 1 that you will submit on Moodle should **ONLY** contain:
  - o Cover page with important details of the team
  - o A link to your GitHub repository. If you create a private repository, add the instructor (JoumanaDargham) and your TA.

Note that all other relevant information must be present on your GitHub account.

If you are not familiar with how to write meeting minutes and user stories you might need to investigate and read the books and supplementary material in the course outline.

The following table summarizes the sprint1 work:

***Sprint 1 Requirements:***

Activities	Sprint Details	Weight
<b>Sprint 1:</b> 1. GitHub Initiation 2. README file submission 3. Task Breakdown 4. Commits of all members	<b>Sprint 1 Deliverables that can be tracked on your GitHub:</b> <ul style="list-style-type: none"> <li>• README file with               <ol style="list-style-type: none"> <li>1. Description of the Project</li> <li>2. Team Members and roles</li> <li>3. Project Approach and Technology</li> </ol> </li> <li>• Document(s) with               <ol style="list-style-type: none"> <li>1. 4 user stories backlog for Sprint1</li> <li>2. Task Breakdown (details of the stories with team members responsibility, if known, for the current sprint).</li> <li>3. Open issues or plan for next Sprint</li> </ol> </li> <li>• Meetings Minutes file (minutes files should be named &lt;teamName_Sprint#_meetingnumber_meetingdate&gt;)</li> </ul>	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

<b>Process general</b>	Total=100
Done enough work (end to end of the system complete and working)	create a repo; show the description of the project, team members. 10 points: Update the <b>readme.md</b> to show: ◦the info of your <b>team members (real name and GitHub user name, role/Strengths of the member in the project)</b> ◦what <b>languages &amp; techniques</b> do you plan to use (if haven't decided, then write something like TBD) 4~8: If any parts listed above are missed
A clear goal of the project	10 points: Update the <b>readme.md</b> to show the <b>description / objective / core features</b> of the project 4~8: The description / objective is vague or unclear
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: Create/document high-level user stories for 4 core features. 4~8: If you missed the user stories for some features/ if the stories 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
<b>Repo</b>	
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 <b>&amp; everyone should participate in the contributions</b> (i.e., we are able to see the contribution somewhere in your repo) If no assignees, <b>-2 from the group marks</b> If someone didn't contribute to the repo, <b>-2 from the mark of that member</b>
Team work	10 points: the work reflects good coordination and collaboration. Meaningful commits and messages. -2.. -5 if it not all members are concerned with the work

# 1. Reference

[1] <https://www.linkedin.com/learning/learning-git-and-github-14213624>

[2] <https://docs.github.com/en/get-started/quickstart/hello-world>