NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing and Cyber Systems

Team Standards

October 4th, 2023

Team IQ

Project Sponsor: Erika Konrad

Faculty Mentor: Igor Steinmacher

Team Members:

Logan Samstag (Team Lead)

Nicholas Persley

Kristiana Kirk

Elian Zamora-Rivera:

Robin Pace

NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing and Cyber Systems

Overview

The purpose of this Team Standards document is to define the roles of all the team members, as well as lay out the standards and expectations for how code will be written and maintained throughout the project. This document will also outline the core technologies in use, as well as a justification for why we chose each technology. This document will also outline expectations for mentor meetings, team meetings, and client meetings. This document will serve as the “one stop shop” for standards and information about the project.

Team Members & Roles

Logan Samstag: Team Leader/ Back-End Engineer

Nicholas Persley: Back-End Engineer/Release Manager

Kristiana Kirk: Front-End Engineer/Recorder

Elian Zamora-Rivera: Front-End Engineer

Robin Pace: Back-End Engineer/Quality Assurance

Team Leader: Coordinate & run team meetings, review all deliverables before final submission, communication with the client.

Back-End Engineer: Coordinate design of back-end components and API interactions. This role is responsible for the flow of the application.

Front-End Engineer: Coordinate design of front-end components and overall user experience. This role is also responsible for the design of the team website.

Release Manager: Reviews code and deliverable changes as well as manages cleanup/maintenance of the GitHub repository and Kanban board.

Quality Assurance: Coordinate the quality of all the deliverables.

NAU NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing and Cyber Systems

Team Meeting Expectations

Meeting Time: Wednesdays 8pm-9pm in Engineering - rm. 101

Agenda Structure: Begin with a brief overview from each team member of accomplished tasks if needed. Next, review the task report from the mentor meeting and assign tasks to team members for the upcoming week. After administrative items, we will move into a more informal working session.

Minutes: Kristiana will take all meeting minutes and upload minutes for each meeting to the GitHub repository.

Discussion-making Process: All the decisions are to be run by all the team members. All the opinions from the respective team members would be taken into consideration, although during a conflict 4/5 majority would be preferred in all cases.

Attendance: All team members are expected to be actively present on time for team meetings. Each team member is allowed two absences from team meetings but must provide all team members with at least 2 hours' notice. Absences without notice and reason will be considered under unusual circumstances, by taking a majority vote from the remaining present members. If a tie occurs, the project mentor will be the tiebreaker vote.

Conduct: All team members are expected to be always courteous, kind, and professional. All group members should bring any issues that need to be discussed to the group during the team meeting. We will start with an informal discussion with the problematic team member. If the issue continues, the team leader will bring the issue to the group during a team meeting for discussion. If the issue persists, the team will discuss the issue with the team mentor and CS Faculty Sponsor if needed.

NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing and Cyber Systems

Tools

Version Control: Git/GitHub will be used.

Branching

The action of merging into the *main* branch will not occur until a version based on semantic versioning (https://semver.org/) is met. That is, a major/minor version is to be released or a patch is necessary to be injected into the program. The branch to merge into *main* will follow the naming scheme “release_vx.x.x”. Each teammate will create branches based on this for their respective work. In order to merge into *main*, a pull request must be created and assigned to the Release Manager (Nicholas Persley) and at least one other person working on the respective side of the program (i.e., front-end or back-end). Finally, when a new branch is created, it will follow a naming scheme of “<type>_<name>”. Where “type” is a descriptive generalization of what it is meant for. For example, this could be a bug fix, feature, etc. And “name” is a descriptive title of what the branch is changing/adding (e.g., “ServerConnections”). An example of a development branch would then be “bugfix_Serverconnections”.

Communication: Discord will be used at the main communication channel. Various channels within it will be used.

- *urgent* - will be used as emergency help. If someone is to send a message within this channel, it should capture the attention of everyone available at the time, and respond as soon as possible.
- *reminders* - will include announcements of important information, things like due dates, uncompleted/need to complete tasks, etc.

- *front-end/back-end* - including their voice variants, these channels will be used to discuss front-end and back-end, respectively. It goes without saying that all team members may access, read, and write into these channels as needed.
- *upcoming_meetings* - will be used to read and write notes taken for a meeting to remind the team members of what needs to be done.

Artificial Intelligence Assistants: The use of artificial intelligence software to write any documentation is prohibited. The use of small help like Grammarly is acceptable as long as it does not rewrite a relatively large amount of a document. Please see the course syllabus for the courses' for the higher-priority rules on this.

Coding

Languages: HTML, CSS, JavaScript, and anything else will be added at a later date.

Styles: The code one writes should follow the general guidelines provided

- Lines of code will be eighty (80) characters maximum, or within reasonable limits beyond it to provide good readability
- Braces will be put on a new line, aside from styling languages like HTML/CSS
- The team will use camel-case and snake case for any constants
- Variables will be defined within the for-loop header instead of at the top of the function

Team Self Review

During each team meeting, all team members will perform a self-review for the previous week. We will take about 5 minutes at the start of each team meeting for each team member to answer the following questions:

- What did you accomplish this week?
- What do you plan to work on this week?
- What do you plan to have completed by the next meeting?
- Do you have any issues/blockers you are facing?

We hope to do this in an informal discussion format to allow team members to voice concerns and ask questions in a non-judgmental space. The primary goal of the reflection is to offer advice to other team members, and provide a time for discussion on more specific tasks that may not have otherwise been discussed.