Aglie Avengers

Team Contract

UTSC CSCC01H3 - Summer 2023

Adam Badar
Aditya Kulkarni
Aryan Thakur
Balraj Kharol
Rahul Sharma
Shashwat Piyush Doshi
Shawn Santhoshgeorge

Contents

1 Introduction

This document is the team's contract. It sets forth the team's expectations and responsibilities. All team members must agree to and abide by this document for the duration of the project.

2 Communication

We will use Discord as our primary communication tool. It will be used for general updates regarding the project and other communications. Members are expected to **respond within 24 hours any day of the week.**

We will use Zoom for our Daily Standup, Sprint Planning, and Sprint Retrospective Meeting,

3 Meetings

3.1 Running Meetings

The minutes of the meeting will be collected into Confluence. The individual responsible for taking the minutes is done in an alphabetical order and will also be in charge of running the meetings.

3.2 Daily Standup

This is a **mandatory** meeting that will occur **Daily at 8 PM** and will be used to update the team on the following:

- What did you do yesterday?
- What do you plan to work on today?
- Do you require any assistance with your task(s)?

Come prepared with answers to these questions above.
We can potential extend our Standup to include the Sprint Planning

3.3 Sprint Planning

This is a **mandatory** meeting that will occur biweekly, with exact date and time to be decided at a later time for each, or it can occur at the same time after a Daily Standup. This meeting will be used to plan out the next sprint and what each team member will be required to be working on.

3.4 Sprint Retrospective Meeting

This is a **mandatory** meeting that will occur biweekly at the end of each sprint and will be used as a time of reflection and learning for the next sprint. The exact time and date for this meeting will be decided at least 1 week before. If you have suggestions for improving the next sprint, come prepared

4 Version Control

4.1 Git Flow

Git Flow is a common git branching strategy that helps with agile software development and will be used for the duration of this project.

It creates 2 branches called `main` and `develop` which are both responsible for recording the history of the project. They also have specific responsibilities. The `main` branch stores the latest stable version of the software while `develop` stores all the latest complete features from the `feature/*` branches.

When a new feature is to be built, you are required to start by branching off from the latest `develop` branch to create a `feature` branch. When the feature is complete a PR will be created to the `develop` branch, never to the `main` branch. If the PR is approved the branch commits will be Squash and Merged to its parent branch.

A more visual flow of this can be found here Git Flow Cheat Sheet
Source: Gitflow Workflow | Atlassian Git Tutorial

4.2 Branch Naming Conventions

A branch should begin with a category, reference, and brief description and will follow the below pattern:

{category/reference/description}

Category

- `feature`: Adding or Modifying a feature
- `bugfix`: Fixing a Bug in `develop`
- `hotfix`: Fixing a Bug in Production, and/or Release
- `chore`: Everything Else (Documentation, Formatting, Adding Tests and others)

Reference

This points to an ticket number and must follow this format `{FIN-#}`, but if no reference exist then add `non-ref`

Description

This is meant to be a brief description of the branch

4.3 Commit Messages

When creating a commit, follow the Conventional Commits Guidelines as it will be used for creating CHANGELOGs for each release of the software and makes it easier to communicate changes to the team. Ensure that you read up on the Conventional Commits

4.4 Testing

Before a PR is setup, ensure that your feature is well-tested to ensure that new changes have not broken current features, here are some things look out for and suggested tools to assist:

Frontend Testing

- Responsiveness: Ensuring that components and pages are designed to scale to the user's screen size, orientation and other preferences is important to providing a great user experience. The apps below will help with simulating these user preferences and provide a way to visually inspect changes.
  - Responsively App
  - Responsive Viewer (Chrome Only)
- Consistency: Ensuring that interactions on the site should be the same for similar components to prevent users from getting confused.
- Accessibility: Ensuring the site is available for all users is important, especially for individuals with disabilities and special needs. The site below will test for various issues including accessibility.
  - PageSpeed Insights
- Cross-Browser Compatibility: Ensuring the user experience is consistent no matter the Browser being used. It would also be more effective if the test was run on various OS and devices combinations

API Testing

We will be using Postman for API Testing below are some cases to ensure your API call can handle.

- Basic Positive Tests: Ensure the API meets the functionality and acceptance criteria
- Extend Positive Tests with Optional Parameters
- Negative Tests: Ensure the API can properly handle errors when providing no input, invalid input, and other cases
- Destructive Tests: Ensure the API call is robust enough to handle intentional attempts to break the API

Source: REST API Testing Strategy: What Exactly Should You Test?

4.5 Documentation

Before a PR is set up, ensure the changes made are well documented. Follow the conventions that exist to ensure consistency. Make sure it is clear and simple for any team member to be able to understand it

5 Division of Work

We will equally distributed the stories amongst all the team members during each Sprint Planning Day. Ideally, team members would be able to choose stories that they would like to work on, but if multiple team members want the story it will be given to an individual with the least number of task(s) first else given to an individual who has worked on this previously. Some stories may require assistance and in this case we recommend pairing up with someone who has worked on a similar stories.

To evaluate how long a story will take we will use the Fibonacci Estimation Technique.

6 Submission of Work

When it comes time to submit work for marking at the end of each Sprint, we will follow the release method of Git Flow

1. Create a branch called `release/(MAJOR.MINOR.PATCH)` at least 1 day before the due date.
2. All features for this release must have been completed at least 1 day before the due data.
3. All team members are required to review each others stories and see if there are any bugs or missing documentation and meets the acceptance criteria for it
4. Repeat above step until all the team members approve the changes

The `MAJOR.MINOR.PATCH` is related to Semantic Versioning.

Once it is approved by the team it will be squashed and merged into both `develop` and `main` branches, with one for `main` being tagged. This will be done by Rahul Sharma and Shashwat Piyush Doshi.

7 Contingency Plan

7.1 Missing Meeting(s)

If a team member is missing meeting(s) due to external factor(s), they must notify the team members at least one day ahead with their valid reason(s). Repeatedly missing meetings will result in a discussion with the team to resolve the issue and can involve the Instructor if required.

7.2 Drop Course

If a team member decides to drop the course, they are required to alert the Team and the Instructor regarding their decision at the earliest possible moment. The task list of this member will be evenly distributed to the remaining team members and this individual will be required to close up any currently in progress task(s) and provide a brief summary of the task(s) to those who will take it on.

7.3 Academic Dishonesty

If a team member believes another member(s) has committed academic dishonesty, they will promptly report it to the Instructor and/or the Teaching Assistants. Any further discussion will involve the entire team, the Instructor and/or Teaching Assistants regarding the next steps.

7.4 Other Issue(s)

For any issues that do not fall into the above categories and could not be resolved within the team, we will bring the issue(s) to the Instructor and/or the Teaching Assistants to be resolved.

8 Agreement

By providing my name here digitally, I acknowledge that I have read and understood the team agreement thoroughly. I agree to these terms and intend to fulfill them to the best of my abilities.

Digital Signatures:

- Adam Badar (2023/05/23)
- Aditya Kulkarni (2023/05/23)
- Aryan Thakur (2023/05/23)
- Rahul Sharma (2023/05/23)
- Balraj Kharol (2023/05/23)
- Shashwat Piyush Doshi (2023/05/23)
- Shawn Santhoshgeorge (2023/05/23)