Spotter

McMaster University

Development Process
SE 4G06 & TRON 4TB6

Group 12

Donisius Wigie - 400143340
Zuhair Makda - 400065864
Artemiy Kokhanov - 400074430
Ridhwan Chowdhury - 400075426
Juwon Adeola - 400075635
Winnie Liang - 400074498

Table of Contents

Revisions

Date	Revision Number	Authors	Comments
October 20, 2021	Revision 0	Zuhair Makda	Created Document and added Sections
October 22, 2021	Revision 0	Artemiy Kokhanov Ridhwan Chowdhury Winnie Liang Donisius Wigie Juwon Adeola Zuhair Makda	Added Roles and completed sections
October 24, 2021	Revision 0	Artemiy Kokhanov Ridhwan Chowdhury Winnie Liang Donisius Wigie Juwon Adeola Zuhair Makda	Final touch-ups and checks for submission

Roles and Responsibilities

The application is primarily software based and all members will touch the code to some extent. Ideally we want everyone in the group to have a good understanding of all aspects of the project so the following roles and responsibilities will be rotated among all members on a bi-weekly basis.

Project Manager:

- Organizes meeting times and locations through the group chat.
- Keeps track of all required deliverables and when they are to be completed and submitted.
- Ensures all documentation is complete and submitted on Avenue before the due/end date.
- Communicates with the Professor/TA when necessary.
- Ensures github issues are resolved within a timely manner.
- Ensures that any required resources and materials for development are collected and set up appropriately.

Scrum Master:

- Runs weekly scrum meetings and informs group of sprint goals, blockers
- Iteration/sprint planning meetings
- Announce outstanding issues and ensure that they're assigned equally among members
- Ensuring scrum framework is followed

All Members:

- Coder:
    - Write readable code which follows a strict style guideline and review process.
    - Work on a suitable architecture for the application.
    - Ensure best practices are followed.
- Code Reviewer:
    - Check backwards compatibility on outstanding pull requests.
    - Handle merge conflicts.
    - Make sure there is sufficient testing and documentation on all pull requests.
    - Suggest changes for improvement.

        - Work with other team members when processing pull requests to ensure changes are non-breaking and all test cases have been passed.
        - Make sure pull requests do not conflict with one another.
- Documentation:
    - Write sufficient documentation in order to maintain code base readability and to make sure all members on the team are on the same page.
    - This should be done asynchronously while writing code.
- QA:
    - Includes writing unit tests, validating documentation, and other types of testing (white box, black box, functional testing).
    - Performing tests
    - Creating github issues to resolve any bugs
    - This should be done asynchronously while writing code.
- Site reliability:
    - Deployment
    - Accessibility
    - Maintenance

Version Control

The primary source code for the application *Spotter* is to be pushed to a github repository that is shared between all members of the project. Within this repository there will be a variety of branches. The master branch will be used for the latest fully functional version of the application where all features have been tried and tested. A dev branch will also be used for pushing forward features that have been worked on but has not been fully tested. This dev branch will act as a pit-stop branch where newly developed features will need to pass specific acceptance tests. Alongside these branches the members of group 12 will use feature branches for implementation of the application's functionality and are encouraged to commit changes greater than 50 lines of code. This is to ensure that the code can easily be reverted if any commits caused unwanted issues and form a steady build up to completion. Before a change can be merged into the master branch, at least one approval from another member is necessary.

Process Workflow

The following steps will outline a general workflow for the development of *Spotter*:

1. Pull all/new code from master and/or dev branch.
2. Create a new feature branch for the designated task of development.
3. Implement the new feature on this new branch following the principles of modularity to ensure low coupling and high cohesion.
4. Once the feature has been implemented to a working state, perform the necessary software testing for the new feature.
5. Push all changes to the newly created branch.
6. Pull any newly added code from the master branch and resolve conflicts if any.
7. Create a pull request to merge the new feature branch into the dev and/or master branch and have another team member review it for approval.
8. Repeat all steps for the implementation of features to the application *Spotter*.

Details on Steps to be Taken

The following steps are to be taken:

1. Create the basic structure of the application (folder structure, empty web application, installation scripts containing all necessary libraries/frameworks).
2. Get a basic application working on all member's devices.
3. Setup testing framework.
4. Find a way to integrate a camera connected to the device to the web browser (there should already be an API built into most modern browsers).
5. Get MediaPipe running on the application.
6. Find a way to pipe video frames from the connected camera to MediaPipe.
7. Come up with methods to translate the coordinates from mediapipe to determine whether or not an exercise was done correctly.
8. Create a way for users to create their own custom constraints for exercises.
9. Deploy the application using AWS.
10. Look into continuous integration.

Testing and documentation should be asynchronous with all steps.

Development Tools and Frameworks

- Github will be used for tracking issues, pending work items, and version control.
- MediaPipe is an open source library that will be used for tracking the user's movements and key landmarks across their body. We chose mediapipe as it is the only lightweight library that we could find that suits our goals.
- VsCode/developer choice of IDE will be used for the development of the website.
- Web Browsers will be used for the testing of the application.
- The React framework will be used for the development of the application's web components.
- AWS will be used for hosting the application publicly.
- Github projects will be used to keep track of tasks.

Handling Changes

- Create a github issue.
- Create a new branch, if necessary, with the name of the change.
- Change the code in question.
- Perform unit testing to ensure the issue is not still prevalent.
- Perform regression or sanity tests to ensure that there are not any new issues created.
- Merge to the Dev/Master branch.
- Close the issue.