

Development Plan

Table 1: Revision History

Date	Developer(s)	Change
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...

Put your introductory blurb here.

1 Team Meeting Plan

We plan on doing a large amount of our work on this project asynchronous but we want to meet at least once a week as a group. We created a when2meet page and decided on meeting as a whole team on Wednesdays at 4:30 and whenever else we see fit. These meeting will be held over Discord. We will also be meeting with Marie-Jean once a week to go give her updates and be provided mentor ship and supervision. This meeting does not have a definite time each week and will depend on Marie-Jean's schedule. These meetings will be held over Zoom.

This project consists of two groups, one group (ours) that is responsible for the Natural Language Processing and another that is responsible for the Operations side of things. We will be working closely with the Operations side of things and every couple of weeks we will synchronize our meetings with professor Mosser and the Operations team. Every other week we will also elect one representative to meet with a member of the Operations team without professor Mosser or Marie-Jean to support synchronization.

2 Team Communication Plan

As a team our main form of communication will be through our Discord group chat. We will plan and host meetings there along with discuss any relevant topics. Discord also provides us the functionality to screen share which will be very useful. We also have a separate Discord group chat with the Operations team included to discuss relevant topics with them and plan meetings.

We will also be working closely with the team in Quebec and for communication with them we will be using Mattermost for communication.

3 Team Member Roles

Matthew Curtis	Lead Developer
Jessica Dawson	Lead Developer
Michael Breau	Lead Developer
Benjamin Chinnery	Lead Developer
Yaruo Tian	Lead Developer

4 Workflow Plan

The version control we will be utilizing is git and GitHub. There will be a larger repository that will include the Operations team and material provided by Professor Mosser. This is where all of the coding will take place for our project. We also have a repository just with our team members which is where all of our documentation will be stored for the course. This is where we will work on files such as the Development Plan and the VnVReport for example. Team members will be responsible for creating new branches for there specific work for any given milestone or general development. They will then be responsible for creating a pull request for when they wish to add there work to the main branch in the repository.

This process will be as follows:

- Fork the repo to create your own version
- Create a branch for the changes you wish to make
- Add and commit changes onto your branch
- Push to your branch
- Go to GitHub and click the green "Compare & pull request" button
- Write your description for your pull request of what you did and then click "Create pull request"

After being reviewed by the group your pull request will be merged.

It is strongly encouraged to create multiple commits per pull request with relatively detailed descriptions in case we need to roll/revert back or to find the commit where certain changes were made or where something went wrong in our code. Also it is key that a tag issue is included in commits with the relevant issue(s) in order to keep track of which commit corresponds to which issues. Your commits should be descriptive and also give a rational for the change.

GitHub will also be used for project management the built-in Kanban board for task organization.

We are currently using the issues tab in GitHub to organize our team meetings, all our deliverables and lecture attendance. For our team meetings we will use an issue to keep track of attendance and also jot down our agenda for the meeting and what was discussed. We also have all of our deliverables marked as issues with the corresponding dates attached in order for us to stay on top of things. We also are labelling all of our issues with the corresponding label being "meeting", "deliverable" or "lecture". For each actual coding related deliverable we will have more in depth issues assigned to define tasks, bugs, and changes in order to systematically keep track of and address every step in our software development cycle for each deliverable. This will help us keep track of what issues (tasks) for a deliverable are still needing to be complete and which are closed. It also helps us track where we are and identify and work on any bugs that may arise. These will be labeled by what deliverable they are part of but will also have a label of "task" or "bug" or any other labels we see fit in the future as we progress.

5 Proof of Concept Demonstration Plan

What is the main risk, or risks, for the success of your project? What will you demonstrate during your proof of concept demonstration to convince yourself that you will be able to overcome this risk?

6 Technology

- Specific programming language
- Specific linter tool (if appropriate)
- Specific unit testing framework
- Investigation of code coverage measuring tools
- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
- Specific performance measuring tools (like Valgrind), if appropriate
- Libraries you will likely be using?
- Tools you will likely be using?

7 Coding Standard

8 Project Scheduling

How will the project be scheduled?