

Development Plan

Software Engineering

Team 2, SyntaxSentinals

Lucas Chen

Dennis Fong

Mohammad Mohsin Khan

Julian Cecchini

Luigi Quattrociochi

Table 1: Revision History

Date	Developer(s)	Change
Sept 19	Lucas Chen	Start template

This report outlines the key components of the SyntaxSentials project plan, starting with the adoption of an MIT license for the project. The team's meeting structure is detailed, including weekly meetings and monthly check-ins with the industry advisor. Key roles, such as chair and minute-taker, are assigned, and a meeting agenda template is provided for structured discussions.

The communication plan revolves around GitHub for task management and issue tracking, with detailed steps for issue creation, labeling, and resolution. Team members are assigned roles, and a workflow for merging code changes, performing testing, and pushing updates to production is presented. Additionally, a proof of concept demonstration plan highlights potential project risks. Concluding with expected technologies, coding standards, and a high-level project schedule.

1 Copyright License

The project will be licensed under the MIT License, allowing for open-source use, modification, and distribution with minimal restrictions. Found [here](#).

2 Team Meeting Plan

The team will meet for at least 4 hours every week, 2 hours during the allotted tutorial time on Monday at 2:30 pm. Precise location will be open but the meeting will be virtually, otherwise on McMaster University campus. the other 2 hours allotted would be done virtually over the weekend.

Meetings will be scheduled with our industry advisor depending on their schedule virtually. The plan is to meet for at least 4 hours every month for guidance and to confirm that the team is on the right track. We hope to meet with our advisor for at least 1 hour at the end of every month for a recap. In person meetings with our advisor will be conducted depending on their availability.

Roles

- **Chair:** Dennis will chair all meetings.
- **Minute-taker:** Mohsin will take minutes during all meetings.

Meeting Preparation

Agendas will be collaboratively prepared by the team before meetings with the advisor to ensure we are focused and productive.

2.1 Meeting Agenda Template

Meeting Details

- **Date:** [Insert Date]
- **Time:** [Insert Time]
- **Location/Platform:** [Insert Location or Virtual Platform]
- **Attendees:** [List of Attendees]

Agenda Items

1. **Review of Progress Since Last Meeting**
 - Updates on tasks and milestones
 - Discussion of any challenges or roadblocks
2. **Discussion Points**
 - Key questions or topics for discussion
 - Feedback and suggestions from the supervisor
3. **Next Steps and Action Items**
 - Summary of key takeaways
 - Assignment of new tasks or responsibilities
 - Timeline for upcoming milestones
4. **Any Other Business (AOB)**
 - Additional concerns or questions
 - Future meeting scheduling

Meeting Minutes

- Minutes taken by: Mohammad Mohsin Khan

3 Team Communication Plan

3.1 Attendance Tracking

The attendance issue will be updated for each lecture and meeting that takes place. This issue can be found on the project GitHub page.

3.2 Issue Creation and Tracking

Tasks will be issued via creating a GitHub issue. This approach ensures all task-related discussions and updates are in a centralized place, providing a clear record of progress. Each issue must include a detailed explanation, enabling all team members to understand the task at hand without needing further clarification. Additionally, issues will be categorized using labels to specify their nature. The available labels are:

- **Documentation:** For tasks related to documentation.
- **Bug:** For resolving errors or malfunctions.
- **New Feature:** For implementing new functionalities.
- **Enhancement:** For enhancing existing features.
- **Low Priority:** For tasks that are not urgent.
- **Med Priority:** For tasks that need attention.
- **High Priority:** For tasks that need immediate attention.
- **Spike:** For tasks that require further clarification or research/feedback.

Additional labels will be introduced as needed throughout the project. Issues will be assigned to the team member most suited to the task, based on their expertise. All relevant commits will be displayed under the issue, and it will automatically close once the associated pull request is merged into the main branch. Including a closing tag (e.g., "closes #12") in the pull request ensures proper closure of the issue. This structured process guarantees accountability and keeps tasks traceable.

Step-by-Step Process

1. Open a New Issue

- Write a clear, detailed description to ensure the team fully understands what needs to be done without needing additional questions or clarification.
- At this stage, the team can discuss what needs to be done, identify the problem, and determine the best course of action.

2. Add a Label

- Choose one of the following labels based on the task type:
 - (a) Bug
 - (b) New Feature
 - (c) Enhancement

- (d) Low Priority
- (e) Med Priority
- (f) High Priority
- (g) Spike

3. Assign the Issue

- Assign the issue to the most relevant team member.

Once the above steps are completed, the assignee should follow the established process workflow. After finishing the task and submitting a pull request, they must ensure the pull request provides sufficient detail to demonstrate the task has been resolved. This could include test cases, bug fixes, or other relevant changes. The pull request should also include a closing tag (e.g., "closes #12") to automatically close the issue and ensure proper tracking.

4 Team Member Roles

Role	Assigned
Team Liason & Minute Taker	Mohammad Mohsin Khan
Meeting Chair	Dennis Fong
Leader	Julian
Note Taker	Lucas Chen
Reviewer	Luigi

5 Workflow Plan

5.1 Workflow to merge changes

1. Pull changes from the main branch
2. Run unit/E2E tests and any new test cases required for new code (Must pass a certain threshold to be approved)
3. Create a branch with the following template, branch type/SS/ticket number. (e.g. feature/SS-01, where branch type is feature or fix or spike, SS as SyntaxSentinel and 01 is the ticket number taken from GitHub Issues board)
4. Comment code with branch type followed by a descriptive message (e.g. feat: add database connection)
5. Open pull request
 - (a) Must follow PR template, found on [GitHub project page](#)
 - (b) Automated tests are run using GitHub Actions

- (c) Reviewers ensure that code meets acceptance criteria and does not regress any test cases

- 6. Merge the branch into main

5.2 Workflow to push to higher environment: PROD

- 1. Merge dev branch into SIT branch
- 2. Perform regression test and manually verify complex workflow
- 3. Merge SIT branch into PROD branch
- 4. Validate changes are in production

6 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?

7 Expected Technology

Tool	Version	Explanation
Git	any	Version control system
GitHub	N/A	A tool for version control and project management
Python	TBD	Main programming language for our AI development
VSCode	Any	Code editor
LaTeX	Any	Documentation tool

8 Coding Standard

This project will use linter and PEP8 to ensure code quality is maintained and is readable to others.

9 Project Scheduling

[How will the project be scheduled? This is the big picture schedule, not details. You will need to reproduce information that is in the course outline for deadlines. —SS]

Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. Why is it important to create a development plan prior to starting the project?
2. In your opinion, what are the advantages and disadvantages of using CI/CD?
3. What disagreements did your group have in this deliverable, if any, and how did you resolve them?

Appendix — Team Charter

External Goals

Our team's external goals for this project are:

To create a state-of-the-art code plagiarism detector using ML and NLP that stands out from traditional tools like MOSS. To showcase the project at the Capstone EXPO and win a prize or recognition. To leverage this project as a highlight in job interviews to showcase our technical and teamwork skills.

Attendance

Expectations

All team members are expected to attend meetings on time. Leaving early or missing meetings should only occur in exceptional circumstances, and prior notice must be given whenever possible. Virtual attendance is allowed if in-person is not feasible.

Acceptable Excuse

Acceptable excuses include health issues, family emergencies, academic conflicts with prior notice, or unforeseen personal commitments. Unacceptable excuses are forgetfulness, lack of motivation, or missing meetings due to non-academic social activities.

In Case of Emergency

If a team member encounters an emergency and cannot attend a meeting or complete their assigned tasks, they should immediately inform the team via group chat (or other agreed communication tools) and arrange to cover or reschedule their tasks with another member if possible.

Accountability and Teamwork

Quality

Each team member is expected to come to meetings prepared, having completed their assigned work. The work brought to the team should be of high quality, thoroughly tested, and reviewed where applicable.

Attitude

Team members should approach each task with a positive attitude and be open to suggestions and feedback. Collaboration, respect, and a willingness to listen to others are key expectations. Everyone's ideas should be considered equally.

Stay on Track

To stay on track, the team will use weekly task management tools (such as Trello, Jira or GitHub Issues) to monitor individual progress. Regular check-ins will help ensure members are meeting their targets. If someone consistently underperforms, they will be required to explain their challenges to the team, and corrective actions will be taken, such as redistributing tasks or providing support. Consequences may include minor penalties like bringing coffee to meetings or scheduling a session with the TA or instructor. If a member performs exceptionally well, they may be rewarded with team recognition or other informal incentives like selecting the next team activity.

Team Building

The team will engage in fun activities like online gaming sessions or casual group lunches to build cohesion. Periodically, the team can also celebrate milestones with group outings or social events.

Decision Making

We will aim for consensus in decision-making. However, if disagreements arise and cannot be resolved quickly, we will proceed with a majority vote. For critical decisions, we will ensure that everyone has a chance to voice their opinion, and any unresolved issues can be escalated to the instructor or TA if necessary.