Table 1: Revision History

Date	Developer(s)	Change
3 February 2022	Kehao Huang	Meeting Plan
4 February 2022	Xunzhou Ye	Communication Plan, Roles, Git Workflow, PoC Demo, Technology, Coding Style
4 February 2022	Anhao Jiao	Review and Inspection
12 April 2022	Kehao Huang	Revision 1

SE 3XA3: Development Plan
Finite State Machine Simulator

L02GRP16, NonDeterministic Key
Anhao Jiao (jiaoa3)
Kehao Huang (huangk53)
Xunzhou Ye (yex33)

4 February 2022

1 Team Meeting Plan

- Recurring meetings on Mondays from 9 a.m. to 10 a.m.
- Small discussion sessions will be held right after each lab section.
- Meetings will mostly be held in a virtual setting. In-person meetings will be arranged if necessary.

2 Team Communication Plan

We will mostly be using instant messaging applications like Microsoft Teams, Discord, WeChat to communicate on any problem or issue we encounter throughout the project development.

3 Team Member Roles

- Xunzhou will be the team leader.
- All members share similar expertise on relevant technologies that would be used in the project.

4 Git Workflow Plan

We will primarily follow the Git workflow model as suggested in the lecture to manage the code base. As of the Git workflow model we follow, the central repo holds two main branches with an infinite lifetime: master and develop. We consider master to be the main branch where the source code of HEAD always reflects a production-ready state. We consider develop to be the main branch where the source code of HEAD always reflects a state with the latest delivered development changes for the next release. When the source code in the develop branch reaches a stable point and is ready to be released, all of the changes should be merged back into master somehow. Labels, or tags in Git rather, will be used as required for indicating deliverable submissions. Only the final changes to an deliverable will be merged into the main branch. All other incremental commits will be recorded in other branches like develop branch, feature branch, etc. New branches should be added if in need.

5 Proof of Concept Demonstration Plan

It is well recognized among the group members that the most difficult part is to implement the LaTeX exportation feature. Developing a system to mechanically generate LaTeX snippets require extensive knowledge of both the LaTeX mark-up language as well as the graphing library, `tkiz`. Difficulties in the testing phase also comes from implementing this feature. To test the exportation output, an additional dependency, the LaTeX rendering backend would be needed. Manual inspection of the rendered result would also be required. Fortunately, all necessary tools for testing are widely available. The overall portability of the code base would be supported by utilizing Python virtual environments along with a project requirements file.

6 Technology

Programming Language Python. Written purely in Python.

Primary IDE PyCharm by JetBrains

Testing Framework pytest

Document Generator Doxygen

7 Coding Style

Comply to Google Python Style Guide with PEP484: Type Hints.

8 Project Schedule

Gantt Project on GitLab

9 Project Review

TO BE COMPLETED