Problem Statement and Goals
ProgName

Team #, Team Name
Student 1 name
Student 2 name
Student 3 name
Student 4 name

Table 1: Revision History

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

1 Problem Statement

[You should check your problem statement with the problem statement checklist. —SS]

[You can change the section headings, as long as you include the required information. —SS]

1.1 Problem

The current methods that exist for plagiarism detection of code has many flaws that make it easily bypassable. A coder can add lines of code that do nothing to some original snippet of code, and the MOSS (Measure of Software Similarity) plagiarism detector will not detect any form of plagiarism between the two. This is an inherent problem that arises from the use of algorithmic solutions; these solutions lack semantic understanding. With how easy it is to plagiarize, students who shouldn't be graduating are able to graduate by putting in a fraction of the work of other students.

1.2   Inputs and Outputs

The problem will take two snippets of code to compare for plagiarism. The desired output is a decision on whether or not the two snippets were plagiarized from one another. [Characterize the problem in terms of "high level" inputs and outputs. Use abstraction so that you can avoid details. —SS]

1.3   Stakeholders

The main stakeholders in this project are professors in any computing and software department, and students enrolled in courses where coding is prevalent. Professors have been identified as stakeholders since they are the people who will be looking out for plagiarism to ensure people don't do it in their courses. This project provides a tool to give professors the ability to make better predictions on plagiarism. Students have also been identified as stakeholders, since students mostly work very hard to complete their degree. It is unfair to them when there are those who can plagiarize and achieve the same results as them, with half the effort.

1.4   Environment

This solution will operate on a device, where two files will be fed to a model. The model will leverage hardware provided on the cloud.

[Hardware and software environment —SS]

2   Goals

3   Stretch Goals

4   Challenge Level and Extras

This project has been assigned a difficulty level of general, and may be subject to change. The aim is to use well known techniques that have been extensively researched, which may push the difficulty to an advanced level, depending on the complexity and feasibility of the research.

The team intends to build an interface to support the project as an extra. More ideas for extras can be added in the future.

[State your expected challenge level (advanced, general or basic). The challenge can come through the required domain knowledge, the implementation or something else. Usually the greater the novelty of a project the greater its challenge level. You should include your rationale for the selected level. Approval of the level will be part of the discussion with the instructor for approving the project. The challenge level, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

[Teams may wish to include extras as either potential bonus grades, or to make up for a less advanced challenge level. Potential extras include usability testing, code walkthroughs, user documentation, formal proof, GenderMag personas, Design Thinking, etc. Normally the maximum number of extras will be two. Approval of the extras will be part of the discussion with the instructor for approving the project. The extras, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

Appendix — Reflection

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. What went well while writing this deliverable?

2. What pain points did you experience during this deliverable, and how did you resolve them?

3. How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?