

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

1.2 Inputs and Outputs

[Characterize the problem in terms of “high level” inputs and outputs. Use abstraction so that you can avoid details. —SS]

1.3 Stakeholders

1.4 Environment

[Hardware and software environment —SS]

2 Goals

3 Stretch Goals

4 Challenge Level and Extras

[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]