Problem Statement and Goals
Software Eng 4G06

Team 2, Parnas' Pals
Jared Bentvelsen
Bassel Rezkalla
Yuvraj Randhawa
Dimitri Tsampiras
Matthew McCracken

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 —SS]

2 Goals

Goal Descriptions	
Goal Name	blah blah blah
Name	blah blah blah
Name	blah blah blah
Name	blah blah blah
Name	blah blah blah

2.1 Stretch Goals

3 Inputs and Outputs

4 Development Plan

4.1 Team meeting plan

Team meetings will be conducted weekly on non-holiday Mondays at 4:30pm. Further meetings can be scheduled if needed. Meetings will be conducted in person at a McMaster bookable room or library unless specified otherwise.

4.2 Team communication plan

Common team communication and questions will be conducted on **Mircosoft Teams**.
Any online meetings will be conducted on **Mircosoft Teams**.

4.3 Workflow plan

GitHub will be used for all workflow and version control.

4.3.1 Task Scheduling

The **Github Kanban** project board will be used to plan and assign upcomming and ongoing tasks to team developers.
If a task is large or general, sub-tasks will be created and assigend accordingly.
Upcomming project delieverables must include deadlines.

4.3.2 Git Branch Usage

All features, documents and file changes require a named branch describing the change. Branches will follow a similar format to:

$$topic\_or\_type/section/branch\_description$$

4.3.3 Git Commits

Tags will be attached to commits as needed to further display the change description.

Commit Squashing may be used prior to creating pull requests to clean up unnecessary commits.

4.3.4 Git Branch Merging and Pull Requests

The main or centered branch will be protected such that new features and addtions will require a pull request in order to be merged.

Pull Requests require a minimum of 2 approvals to be accepted.

4.3.5 Technial Issues

For all technial questions, concerns and issues, the **Github Issues** feature will be used to open an issue or concern accordingly.

4.4 Team Roles

Team Role Descriptions	
Role	blah blah blah
Front End Expert	Dimitri
Database Expert	Jared
UX/UI Expert	Matt
Role	blah blah blah

4.5 Coding Style and Standards

The **Google Coding Style** will be used as a standard for all code in the project. The style guide for this can be found here. A language specific linter will be enforced.

4.6 Testing

4.7 Risks