Table 1: Revision History

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

Development Plan
REVITALIZE

Team 13, REVITALIZE
Bill Nguyen and nguyew3
Syed Bokhari and bokhars
Hasan Kibria and kibriah
Youssef Dahab and dahaby
Logan Brown and brownl33
Mahmoud Anklis and anklism

[Put your introductory blurb here. —SS]

1 Team Meeting Plan

2 Team Communication Plan

3 Team Member Roles

4 Workflow Plan

The main repository is named REVITALIZE. The implementation will follow the feature-branch model. Feature development will take place in a branch other than the main branch. This encapsulation makes it easy for our team to work on a feature without disturbing the main codebase. That way, the main branch will never contain broken code. The feature-branch model will make it simple for our team to initiate discussions around a branch by making pull requests to comment on each other's work.

The documentation will follow the centralized model. The main branch will serve as a point of entry for all changes to the documentation. All changes will be committed to this branch.

To report bugs and request modifications to implementation or documentation, we will create GitHub issues and assign them to the people working on a specific problem. This will also serve as a way of holding team members accountable for the issues they have to tackle in the future. Labels will be used to classify

issues and pull requests to help create a standard workflow in our repository.

If required, milestones will be created to track progress on groups of issues or pull requests in our repository to better manage our project through viewing milestones' view dates, completion percentages, and lists of open and closed issues and pull requests associated with the milestones. Using milestones could become a necessity as our project size and complexity increases with time.

5 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?

6 Technology

- Specific programming language
- Specific linter tool (if appropriate)
- Specific unit testing framework
- Investigation of code coverage measuring tools
- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
- Specific performance measuring tools (like Valgrind), if appropriate
- Libraries you will likely be using?
- Tools you will likely be using?

7 Coding Standard

8 Project Scheduling

[How will the project be scheduled? —SS]