

Table 1: Revision History

Date	Developer(s)	Change
Sept. 26, 2016	Christopher, Varun	Initial development plan
Sept. 28, 2016	Varun	Formatting, Introduction and Git Workflow
Sept. 29, 2016	Christopher, Varun	Changes to wording and added Proof of Concept plan section
Dec. 06, 2016	Varun	Fixes from feedback
Dev. 08, 2016	Varun	Review

SE 3XA3: Problem Statement

Spann

Team 5

Christopher Stokes — stokescd

Varun Hooda — hoodav

This document outlines some key points regarding the development of this project. The points discussed related to the way the team will work (meetings, workflow, communication) and how the project will undergo development and progress as time goes on.

1 Team Meeting Plan

The team will meet on a weekly basis on Tuesday afternoons on campus (exact location and time is up to the discretion of the team members). The meetings will allow the team to express any concerns, discuss upcoming deadlines/milestones, and discuss the work plan for the upcoming week. Team members will alternate as the chair for each meeting. The chair will be responsible for creating an appropriate agenda for the weeks meeting and for directing the meeting.

2 Team Communication Plan

The primary means of communication will be Google Hangouts. Issue tracking on gitlab will be used for formally discussing any issues with the project.

3 Team Member Roles

There will be no team leader due the small team size. A team member may lead the team for a particular task if the team member is experienced in that particular task. Roles will be as follows:

Christopher Stokes Primarily work on backend server code, as well as some work on the custom framework that will be used and on some frontend code. Will be the expert on the technologies used on this project.

Varun Hooda Primarily work on frontend code, as well as on the custom framework. Will be the expert on \LaTeX

4 Git Workflow Plan

Git and Gitlab will be used to manage the project's documentation and code base. The team will use a single repository (no forks) with all developers contributing to the same code base. Git branches will be used to reduce conflicts between different incomplete features. The team will attempt to commit and push changes frequently. Labels will be used to differentiate or highlight particular milestones.

5 Proof of Concept Demonstration Plan

The goal of the demonstration plan is to identify any risks that the team may not be able to overcome. In preparation for the demonstration, the team will build a prototype of the final project that will be used to highlight the primary goal of the project. This prototype may highlight the issues previously mentioned, thus help the team move forward by either redefining the project scope or adjusting the project to avoid the risk. The prototype itself will be a simplified version of the final software, including only the key critical features that are vital to the project or features that have a significant risk attached to them.

The goal of the demonstration, essentially, is to highlight any of the following possible issues:

- Implementation problems
 - Feature implementation is beyond the skill level of the team members
 - Feature is infeasible to implement
 - Feature implementation would require too much time
 - Feature implementation would require resources beyond the groups means
- Verify the project can be tested and verified
 - Will there be some feature or part of the project that is impossible/infeasible/impractical to test and ensure correctness of
- All software/hardware dependencies are satisfiable
 - Will the team be able to setup the required supporting hardware and software to allow the project to run
- Issues running the prototype on different devices
 - Will the project be able to provide the same functionality on different platforms, operation systems, and browsers
- The final project will be able to do what is specified
 - The project goals can be implemented and the final project will be able to meet the specification requirements

6 Technology

The project will use modern web technologies for the frontend: HTML5, CSS3 (compiled from LESS) and JavaScript. The backend server will run C# code and use the PostgreSQL object-relational database management system (ORDBMS) to store and retrieve data. A custom JavaScript web framework will also be used to create the frontend website (to avoid manually writing HTML).

For testing, the project will use Postman for the API, Jasmine for JavaScript and Nunit for C#.

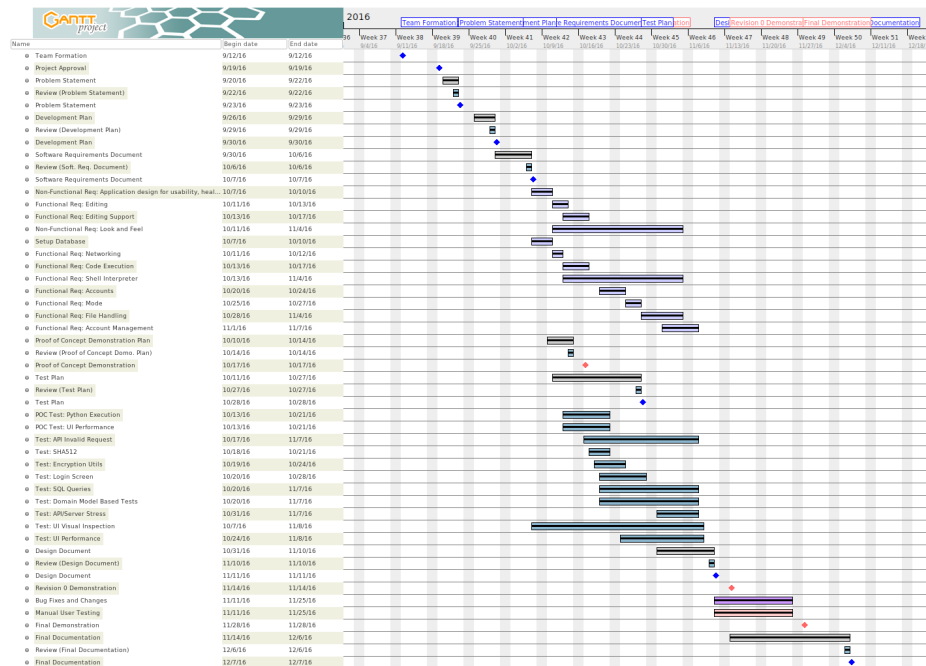
For documentation generation, the project will use C# source code documentation and Doxygen.

7 Coding Style

The project will the following coding styles for all of the code:

- ECMA5 Script Standard for JavaScript
- LESS Standard for LESS files
- Microsoft Coding Conventions for C#

8 Project Schedule



9 Project Review

What was well done

Many aspects of this project were well done. Many of the industry standard technologies were used, C#, IIS, REST API and JavaScript frameworks. The team worked well together, there no issues or arguments or serious differences in opinions. The team also stays well informed and communication was very well.

What was not well done

Due to insufficient knowledge of some of the technologies in use, a lot of the available time went into learning the technologies. This slowed down the development to a certain degree. While this is not a negative point, it is something that hindered the potential of the project in the limited time span. Another thing that was not done as well as it could have been was the documentation of the project. While the documentation did cover what was required, in reflection, spending more time on this part would have resulted into much more thorough documentation of the project as whole. Finally, the testing and testing methods used by the team could be improved. A more throughout set of tests would result in better code coverage, better design, better overall performance and discovery of more bugs.