

SE 3XA3: Development Plan
Synergy Inventory Management System (SIMS)

Team #33, 'Sick Ideas'
Nathan Coit – 400022342
Lucas Shanks – 400029943
Cameron Van Ravens – 400020215

December 4, 2017

Table 1: Revision History

Date	Developer(s)	Change
11/28/17	Cam Van Ravens, Nathan Coit, Lucas Shanks	Initial Revision
12/04/17	Cam Van Ravens, Nathan Coit, Lucas Shanks	Revision 1

This is the development plan for the **Synergy Inventory Management System (SIMS)**, which covers the plans for Team meetings and communications, states member roles, as well as covers plans for Git workflow and the technologies involved.

1 Team Meeting Plan

Regular team meetings are to be held weekly on Thursdays at 14:30, with a duration of 1 hour. All team members are required to attend these meetings. Any changes to a specific weekly meeting date must be communicated and approved by all team members, as well as made note of in that week's Meeting Minutes.

Regular team meetings will cover the following topics:

- Individual deliverables progress
- Team milestone progress
- Outstanding issues and setbacks
- Any proposed changes (regarding development, requirements, team management, etc.)

2 Team Communication Plan

All team communications will be made through a Facebook group chat, if not conducted in person. Team members will be responsible for keeping themselves up-to-date with this chat, as well as using it for all outgoing team communications. All communications regarding the team or project must be accessible by all team members.

3 Team Member Roles

The team member roles will be as follows:

- **Nathan Coit**
 - Developer (Frontend Connectivity)
 - Meeting Scribe
 - Documentation Expert
- **Lucas Shanks**
 - Developer (Frontend UI)
 - QA Manager

- **Cameron Van Ravens**
 - Developer (Backend API)
 - Project Manager
 - Git & Technology Expert

4 Git Workflow Plan

The Git workflow will be conducted as follows:

- **Production** (or master branch)
 - This branch will contain a stable build of the project at all times, as this branch will contain the version of the project to be deployed.
 - No direct work or edits may be made to this branch. All changes must be made through a pull request.
 - Pull requests to this branch must pass all tests and be approved by at least two team members before being approved. *All Pull Requests to this branch can only be made from **Development**.*
- **Staging**
 - ~~– This branch will contain a build of the project to be staged, before it can issue a pull request to the **Production** branch.~~
 - ~~– This branch will be subject to testing through Continuous Integration, and will trigger testing upon receiving any changes to its code base.~~
 - ~~– Pull requests to this branch must be approved by at least two team members.~~
- **Development**
 - This branch will contain a build of the project waiting to be tested and deployed.
 - Pull requests to this branch can be made from any of the team members main branches
- **Individual Main Branches**
 - Each team member will have their own main branch to work from, forked from the development branch.
 - Each team member may create branches from their branch, and merge back to their own branch without audit from other team members.
 - Team members are forbidden from pulling changes from another team member's branch

5 Proof of Concept Demonstration Plan

The Proof of Concept demonstration will present a slimmed-down version of the platform which will store and retrieve user inventory data from the database, and display it in a prototype of the management interface. A significant risk in the development of the Proof of Concept concerns its implementation, however this will be resolved through prioritizing development tasks.

6 Technology

The technologies used for this project will be:

- **NodeJS** - The backend language
- **ExpressJS** - The webserver used for the backend
- **PostgreSQL** - The database used for the project
- **HTML, CSS, Typescript** - Used for the frontend user interface

The Javascript and Typescript used in this implementation will follow the new ECMAScript 6th edition features.

7 Coding Style

The coding style that will be followed for this project is Google's coding style for Javascript. This document can be found at <https://google.github.io/styleguide/jsguide.html>. This coding style will be enforced using a technology called *Linting*, which will analyze and report styling errors to the developer when attempting to run their code.

8 Project Schedule

For the project schedule, please refer to this Gantt Chart.

9 Project Review

In reflection of this Development Plan over the development of this project, it was very effective. Having the defined branches allowed developers to not worry about breaking things for everyone and allowed for more work to be done. The goals of setting up CI/CD had to be dismissed, as the project proved to be too large and featured to write enough tests for effectively using CI in the time constraint, as well as the SSH port being blocked on the GitLab server's firewall made it very difficult to set up a truly automated deployment. Overall, we believe that this development plan was greatly beneficial to our group in allowing us to get this project done efficiently, organized, and painlessly.