**Site-Stall Project Plan**

Sean Wilson, Noah Tigner, Claire Kolln, Jimmy Lam, Lucas Hyatt

**Tables of Contents**

1. **Management Plan 1**
2. **Milestones/Project Schedule 2**
3. **Monitoring and Reporting 3**
4. **Build Plan 3**
   1. **Rationale behind the Build Plan 4**
5. **Acknowledgements 5**

**1. Management Plan**

Our group has unanimously agreed to approach this project in a democratic format. Important decisions will be decided by vote during our team meetings. Our first decision was to communicate through a program called *Discord*. This can be used through the *Discord* app, and allows the team to use voice chat as well as standard messaging. When we want to meet as a group team meetings will occur in the PSC library.

Our team is organized based on the requirements we have established for the *SiteStall* application. The requirements can be broadly separated into 4 general areas; graphical user interface (GUI) design, data management, blocking functionality, and scheduling. Recognizing these independent components of our project has allowed us to assign individuals to different roles depending on their strengths and interests pertaining to these different requirements.

* **Sean** is responsible for *GUI* development.
  + Anything pertaining to display will be the focus of this role.
* **Noah** will handle the *blocking system functionality* responsible for stalling websites.
  + This role will entail backend development which determines which websites to be “stalled”.
* **Jimmy** and **Claire** will focus on the *data management* module.
  + This role includes handling user data including each user’s respective “blocklists”.
* **Lucas** is responsible for developing dedicated components for *scheduling* within the blocking system module*.*
  + This includes implementing a timer which can be used for immediate stalling, as well as functionality supporting scheduled stalls.

**2. Milestone/Project Schedule**

Development Milestones (TBCB = To be completed by):

1. Initial project ideation; formulate concept of operations and prepare initial presentation.
2. Meet as a group and discuss potential “must-haves” and “could-haves” with regards to our project specification.
3. Divide roles within the group based off of team member ability.
   1. Roles are closely related to initial project requirements.
4. Create repository and connect team members via GitHub.
5. Complete initial documentation for the project, including System Requirement Specification, System Design Specification, and Project Plan.
6. Develop backend implementation for blocking a website. This will likely be run through the terminal.
   1. TBCB Noah Tigner
7. Create an initial user interface to connect to the backend implementation.
   1. TBCB Sean Wils
8. Integrate initial GUI and backend blocking system for completion of prototype 0.1.
9. Complete data management module for storing user data and blocklist functionality.
   1. TBCB Claire Kolln
10. Implement scheduling component within blocking system for tracking and executing scheduled stalls. This will be an updated version of the blocking system entirely.
    1. TBCB Lucas Hyatt
11. Implement user output and metrics for displaying usage of the app within the data manager module.
    1. TBCB Jimmy Lam
12. Finalize an updated version of the GUI to support usage displays and user blocklist data.
13. Connect the updated GUI to the updated data management and blocking system modules. This will signify the completion of prototype 1.0.
14. Finish testing and validation of prototype 1.0.
15. Add reminders and notifications to the working system.
    1. TBCB Claire Kolln + Jimmy Lam
16. Finalize system enhancements and complete prototype 2.0.
17. Send out beta version of prototype for testing among non-classmates.
18. Complete user documentation and programmers documentation.
19. Review, update, and finalize the SRS and SDS documents.
20. Turn in final deliverables.

**3. Monitoring and Reporting**

Monitoring and reporting will be conducted within a spreadsheet called “Tasks and Assignment Breakdown”. As with other documentation files, this file is located within a Google Drive folder and will be updated upon completion of task assignments. This spreadsheet is shared with all members, so when a team member completes a task, they can visit the document. Marking off a task will include the week that it was completed, the final status of the task, the group member(s) who completed it, the date it was assigned and the date it was completed, and the amount of time it took to complete the task. There is an optional “notes” portion which may be used for tracking any need-to-know updates regarding the task. This spreadsheet will function to track our progress throughout the duration of this project. We will monitor this spreadsheet frequently to keep track of who did what and when they did it.

**4. Build Plan**

* A build plan. What is the sequence of steps you will take to build the system? When will each "build" of the system take place

Our group has decided to approach this project with an evolutionary prototyping software development cycle, so most of our “builds” will take the form of a prototype. We plan to have three main prototype builds:

**Version 0.1:** This is the initial prototype which is essentially a basic backend model connected to a user interface. This will include:

* A backend implementation for stalling websites.
* A basic GUI with a user view and buttons for selecting websites to stall.

**Version 1.0**: This will be a more robust and substantial build compared to version 0.1. In this prototype, we will include:

* A data management module for storing user data and “blocklists”
* Scheduling functionality for initiating schedule website stalls.
* An updated user interface to support different users, as well as schedule features.

**Version 2.0**: Following Version 1.0, we will continue testing and perform enhancements to our prototype model. We also have a few features we plan to add to this including:

* A component within data management for sending reminders of upcoming stall periods, as well as notifications to users to enhance program feedback.
* A rewards system for users who use the application

Our goal with this prototype is to continue building upon our last versions, while adding some more engaging features which we feel will enhance the usability of our application.

**4.1 Rationale Behind the Build Plan**

Our team felt comfortable approaching this project through the evolutionary prototyping development cycle because we felt it helped to reduce our risk of venturing away from the requirements we have laid out for this project. Upon completion of a prototype, we are able to perform testing and revisit our specification, to ensure that we are on track with development. Furthermore, each prototype signifies progress within our project. The first prototype will allow us to test our initial backend implementation along with a simple GUI. The second prototype is where we will add the bulk of our requirements, but considering this is not the last deliverable, we will have an opportunity to validate where we are at within the project. Our final prototype will be our deliverable. This prototype will be adding less features, but will greatly improve upon the previous build. These steps will allow our team to test and validate each build, and will allow us to work closely with our project specification. The potential risk of prototyping is that we build a prototype and upon completion, we realize that it functions differently than we envisioned. This could be a result of not foreshadowing how each module would interact together. If this were to happen, we are capable and willing to revisit the design for appropriate updates.

**5. Acknowledgements**

This project plan was designed by the *SiteStall* team, collectively. All key decisions were decided upon during group meetings. The plan is based off of Anthony Hornoff’s “evaluation criteria” document.

<https://classes.cs.uoregon.edu/20W/cis422/Project_Evaluation.html>