Brett Nielsen

Cade Rasmussen

Eric Lambert

Zach Ellis

<Tool Coop>

# **Project Overview**

This project will allow the Neighborhood Tool Co-op to manage their inventory while allowing customers to reserve tools. The project will allow for full management of inventory. This includes creating new tools in inventory, adjusting item quantities, marking where the item is located, and storing an image of the tool. There will be full control of customer management. The Neighborhood Tool Co-op will be able to create new customers and manage the customers’ information. The Co-op will be able to create reservations, assign them to a customer, and add tools to the reservation. As tools are checked in and out, the Co-op will be able to track how many are in inventory and know which tools may be checked out to new customers.

# **Team Organization**

Our team is composed of Brett Nielsen, Cade Rasmussen, Eric Lambert, and Zach Ellis. We will all share equal responsibility in the creation of the website application. To stay organized, different features will be assigned to each person. We will adopt a philosophy of egoless programming. This philosophy will guide development as we each take responsibility for our own tasks, and lean on each other for assistance.

# **Software Development Process**

The development will be broken up into five phases. Each phase will be a little like a Sprint in an Agile method and a little like an iteration in a Spiral process. Specifically, each phase will be like a Sprint, in that, work to be done will be organized into small tasks, placed into a “backlog”, and prioritized. Then, using on time-box scheduling, the team will decide which tasks the phase (Sprint) will address. The team will use a Scrum Board to keep track of tasks in the backlog, those that will be part of the current Sprint, those in progress, and those that are done.

Each phase will also be a little like an iteration in a Spiral process, in that each phase will include some risk analysis and that any development activity (requirements capture, analysis, design, implementation, etc.) can be done during any phase. Early phases will focus on understanding (requirements capture and analysis) and subsequent phases will focus on design and implementation. Each phase will include a retrospective.

|  |  |
| --- | --- |
| **Phase** | **Iteration** |
| 1. | Phase 1 - Requirements Capture |
| 2. | Phase 2 - Analysis, Architectural, UI, and DB Design |
| 3 | Phase 3 - Implementation, and Unit Testing |
| 4 | Phase 4 - More Implementation and Testing |

We will use Unified Modeling Language (UML) to document user goals, structural concepts, component interactions, and behaviors.

# **Communication policies, procedures, and tools**

We will meet at least twice a week at 10:30 on Tuesdays and Thursdays. Any communication that is needed outside of these two days will be through Discord messages, or through a group text message. All new code will be written on a separate git branch. This branch will be sent to the team for a code review before a different team member approves and merges the branch to master. Backend code will be written in PHP using Laravel for a framework. We will develop locally using a Homestead Vagrant Box and will deploy code to a server hosted on AWS using a LAMP (Linux, Apache, MySQL, PHP) stack.

# **Configuration Management**

See the README.md in the Git repository.