Tool Database Project

# Project Overview

This project aims to build a system for handling a tool rental system for use by a community tool storage service.

The rental system will be used by the owners of the tool storage service as well as community members. It aims to simplify the tool borrowing process and reduce inefficiencies. The system will include a database of tools and their rental status linked to a website where users will be able to view and checkout tools. Anonymous users, regular accounts, and administrative accounts will have different website permissions. The administrative account will be held by the owners and employees of the tool storage service.

# Team Organization

The team will practice an egoless programming strategy, without any set leadership or specific programming roles. Individual responsibilities will be decided upon during team meetings according to individual skills and experience. Assignments will also be made with the aim of equitably distributing the workload. In the event of an emergency such as a natural disaster, any group member may assume complete control over the project.

# 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 be communicating through the web service “Slack”, and our version control system will be Git. We will host our project on GitHub.

Team meetings will occur every Tuesday and Thursday during class. If needed, additional meetings will be scheduled as team member availability dictates.

# Risk Management

We anticipate few risks from this project. The following are potential issues we may run into over the course of this project:

**Git inexperience.** Many team members are unfamiliar with the intricacies of Git, such as forking and merging. If we fail to adequately learn how to use Git effectively, there may be issues with our version control.

**Web Development Rustiness**. While some team members are confident in their ability to program the web interface of the project, others are less familiar with web development principles and will need to review in order to contribute to the assignment.

During team meetings, we will communicate our progress and understanding of git and web development, ensuring that these risks are mitigated.

# Configuration Management

See the README.md in the Git repository.