Toolshed Inventory Application

# Project Overview

This project aims to build an application to keep track of community toolshed’s inventory. It also aims to provide a web based reservation system for the toolshed.

This application will provide inventory and member management features for toolshed administrators by providing data on which tools have been checked out to which members. The software will handle everything from keeping track of available tools in a database to renting out those tools to customers, as well as keeping track of checkout and due dates. It will display late fees owed by members, as well as list tools that have been reported broken. The application will allow members of the toolshed to make reservations for tools online, and potentially be notified when their tools are ready for pickup. An extensive list of goals we aim to achieve while developing this product is given in the Product Requirements document in the GIT repository. The primary focus of this application is to allow the community toolshed to efficiently keep track of its inventory and more easily maintain its business.

# Team Organization

This will be a self-directed team. Each team member will be given a task and will have autonomy over deciding how the task will be done. Each team member will then report back to the group to ensure goals are met. We intend to abide by the philosophy of egoless programming, so any mistakes can be corrected in a timely manner.

The team members will initially be responsible for one to two aspects of the project at a time. Once a team member has finished a task, it will be given to another team member to review for errors and enhancements. After review, the task may be reassigned to a team member to correct or enhance an aspect of that task that has already been completed.

Because this is a self-directed team, ultimately everybody is held accountable and responsible for everyone else’s work. After thorough reviews of each other’s features and tasks, it is ensured that we create the best quality application and limit the number of bugs that appear during implementation. We have chosen this method of organizing our team purposefully and to give every team member experience in each area of application development including backend services, the front end, and hosting web sites.

# 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

**Customer Communication**

Communication between the customer and development team is essential to achieve a product that will satisfy the needs of the customer and can be built by the development team in a timely manner. Using the agile methodology, the team will spend time developing this product, and then certain team members will test the product as if they were an end user. Feedback from this process will lead to enhancements and bug fixes throughout the development phase. Switching between a customer and a developer mindset is necessary to gain insight into the usability and intuitiveness of our product.

**Team Communication**

Communication will be necessary when each member takes time to work on the project. While we already know what each team member will do organizationally, it is necessary to know when and what each person will be doing so that everyone else on the team is aware. We will meet weekly to review current goals and the status of the project, and to make new assignments if necessary. Face to face meetings will be scheduled based on the availability of all team members. When our development team is not available to meet, we will communicate through GroupMe, a group messaging app.

# Risk Analysis

* None of the team members have any experience with Amazon Beanstalk. This will present a learning curve, and potentially cause delays in the Deployment of the product. In order to prevent any delays, we will begin studying and researching Beanstalk throughout the development processes.
* As we are using a SQL database, we could potentially lose the master password to the Database which would prevent us from migrating it to another machine. Two team members will keep a copy of this password to ensure it isn’t lost during the completion of the project.
* Many team members have a full class/workload, so they become very busy at various points during the project creation. All team members will communicate about any scheduling issues, to reduce any delays to fulfilling the project requirements.

# Configuration Management

See the README.md in the Git repository.