Lawn Wizard

Project Overview

This app creates a crowdsourcing environment for local yard care work.

Team Organization

Project Manager: Kelsye Anderson (may change over the course of the project)

Designers and Developers: Camila Vulcano, Thomas Lau, James Seelos

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

Group text – Main channel for communication. Used for scrub meetings and setting up future meetings.

Discord – Used for group calls, file sharing, and other collaborative activities.

Google Drive – Storage for files needing collaborative effort and review for Milestone 1.

GitHub – Formal repository used for submissions, version control, data tracking, and communication with Professor Dan Watson and Rob Johnson.

Risk Analysis

- Database Structure
    - Likelihood – Low
    - Severity – Very High
    - Consequences – Ineffective data tracking leading to confusion concerning balances, transactions, account information, etc.

    - Work-Around – None. System loses value and functionality without proper database implementation.

- Login
    - Likelihood – Low
    - Severity – Med-High
    - Consequences – Dissatisfactory customer experience concerning preferences, transaction information, cancellations, and current balance.
    - Work-Around – None
- Verification System
    - Likelihood – Low
    - Severity – Med
    - Consequences – Lack of security concerning exchange of lawn care services
    - Work-Around – Verify by name
- UI
    - Likelihood – Low
    - Severity – Very High
    - Consequences – Inability to interact with users in a clear and efficient way
    - Work-Around – None. System loses value and functionality if users are not able to interact with it.
- Hosting
    - Likelihood – Low
    - Severity – Med

- Consequences – Inability for system to host or serve information essential for system functionality
- Work-Around – Host system through google play store

Configuration management

See the README.md in the Git repository