Putt’n Ale Web App

Project Overview

The project aims to build a web-app management system for a Putt-putt golf + bar business.

The goal of the app is to automate certain areas of the business. It will allow players to automatically register for tournaments. While they are playing in the tournament, they will be able to enter their scores and order a drink from the bar without having to leave their hole. The system will track scores in a leaderboard and dole out prize winning automatically when the tournament finishes. Local businesses can register as sponsors for tournaments which means they pay the prize winnings, and advertise their brand in the web app. Staff will be able to view drink orders and fulfill them. Administrators / managers will have the ability to set up tournaments, verify sponsors, and override tournament results in the event of cheating.

The app will be usable from both a phone for players, and from a PC for sponsors and staff.

Team Organization

Scrum master: Adam Alder (but will rotate between phases)

Technical Lead: Sean Collings

Full-Stack Developers: Sean Collings, Jaren Glenn, Adam Alder, Jonathan Gates.

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

Daily communication is conducted on a Discord server that is team created.

The team holds bi-weekly standup meetings on Tuesday and Thursday where we run through what has been completed, and what needs to be completed before the next meeting.

Github will be used as the remote repository host. Additionally, Github's issue tracking and project features will be used for tracking tasks and bugs.

Configuration Management

Refer to the README.md in the Git repository

Risk Analysis

- Database Schema
  - Likelihood - Low / Medium
  - Severity - High
  - Consequences - Incomplete / ineffective data schema will result in a large amount of rework down the line when it needs to be changed.
  - Work-around - None
- User Authentication
  - Likelihood - Low
  - Severity - High
  - Consequences - Users will be unable to interact with the application.
  - Work-around - None
- User Interface

  - Likelihood - Low
  - Severity - High
  - Consequences - Users will be unable to interact with the application in any meaningful way.
  - Work-around - None
- Age Verification
  - Likelihood - Medium
  - Severity - Very High
  - Consequences - Selling alcohol to underage individuals is illegal and could result in a shutdown
  - Work-around - Verify with birthdate and ID (probably don’t require the second one)
- Account Balance
  - Likelihood - Low
  - Severity - Very High
  - Consequences - User’s lose money they didn’t spend, the receive money they didn’t deposit
  - Work-around - None