4/30/2017

Automated Mini-Golf Course Design

By The A-Team

The A-Team

Flavio Aquino, Sharon Tender, and Frank Castillo

# Overview

This document creates the detailed requirements and detailed design for automating a Mini-Golf Course. It will be documented with text, unit tests, UML, or a combination of these methods.

This design solution will automate the task of recording the number of strokes for each hole. It will incorporate the use of a keypad at each hole. The keypad utilizes a swipe-card that identifies the player and automatically records and keeps track of their scores.

The various courses around Tampa Bay all use the same central computer (already installed), connected to the card-swipe stations at each course using a network.  All the hardware has been installed already and the network is up and running.

## Requirements

Develop an automated mini-golf course design with the following capabilities:

* Need #1: Work with existing central network:
  + Card-swipe stations with the keypad
  + Central computer
  + Hardware and network
  + Card made with disposable cardboard (utilize a barcode??)
* Need #2: UI screen at each Card-swipe station with keypad for player score input
* Need #3: UI screen should display:
  + Players name
  + Current score
  + Number of strokes over or under par
* Need #4: Last hole UI screen should have two extra capabilities:
  + Optional printout of their score card
  + A discount coupon printed on the back for next mini-golf game

# Use-Case scenerios

## Use Case - Golfer

### Use Triggers

* Need to record current score
* Need to add score to scorecard
* Need to show current hole score
* Need to check if strokes are over or under par

### Pre-Conditions

* Must have paid for mini-golf round
* Must have swipe card
* Card must be activated by employee
* Card must have the golfer’s name associated with it
  + Name can be printed on the card
  + Name can be associated with it in computer system

### Post-Conditions

* Game data must be persistent after each score entered

### Normal Flow for Golfer

1. Player purchases a game of mini-golf
   1. Employee activates swipe card
   2. Employee assigns player’s name to swipe card
   3. Employee initiates the start of game to the server
2. Player starts the game of mini-golf at the Hole #1
   1. Player sinks the ball in the hole.
   2. Player swipes their assigned card at the card-swipe station
   3. Display screen identifies player by the name activated on the card and displays it
   4. Player enters number of strokes
   5. Card-swipe station displays player’s name, current score, number of strokes under/over par
   6. These events are repeated for each hole until all are played
3. After entering number of strokes for last hole, card-swipe station gives the following options:
   1. Display screen shows player’s name, current score, and final score
   2. Display screen shows two options:
      1. Yes, print scorecard
         1. Prints coupon on back of scorecard
      2. No, do not print scorecard

## Use Case – Mini-Golf Employee

### Employee Use Triggers

* Need to

### Pre-Conditions

* Must have
* Needs a password

### Post-Conditions

* Changes must be persistent

### Normal Flow

1. Steps for Employee

# UML design for classes

# Technology Requirements

< this section is from lecture notes page 216>

## Functional Requirements:

Uptime:

Minimum sustained load:

Burst loads:

Response time:

Failure modes:

## Operational Requirements (non-functional requirements)

System management and monitoring:

Logging

Configuration files

Toggling on/of individual features and services

Access control and rate limits

Graceful degradation

Ability to deploy in a cluster

Queue draining

Backup and restoring data

Instrumented code that allows monitoring and collection of performance data Performance data includes timing data and counters, such as a counter incremented at the top of certain methods.

# System description

< I got this topic from the homework assignment. This might just be a description of the current system described there. If so, that is what I typed below… >

* Server where all data is stored
  + One central location in Tampa Bay area
  + Recommendation #1: Backup server at secondary location
    - Preferably out of state in case of weather event
* Hardware
  + All hardware has been previously installed
    - Card-swipe stations
* Network
  + Up and running