Software Installation and Maintenance Guide for

CLAW Credit Checking System Upgrade

To be submitted to the Department of Mathematics and Computer Science,

Gordon College

in partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science

by

Jonathan Manos and Travis Pullen

Revision date: 4/25/2016

Jonathan Manos & Travis Pullen
4/25/2016

Software Installation and Maintenance Guide

Installation Guide:

Installing the CLAW Credit Checker Application

1. CheckerApp Directory with all of its contents
    a. Can be obtained from github at https://github.com/jonathanmanos/CheckerApp
        i. Use command: git clone https://github.com/jonathanmanos/CheckerApp.git
    b. Can be obtained via USB transfer of the application

Running the CLAW Credit Checker Application

1. Plug the USB RFID Reader into an empty USB slot
    a. If it's the first time – let the device install itself
2. Start the following .exe file:
    a. \CheckerApp\CheckerApplication-master\WpfApplication2\bin\Debug\CheckerSystem.exe

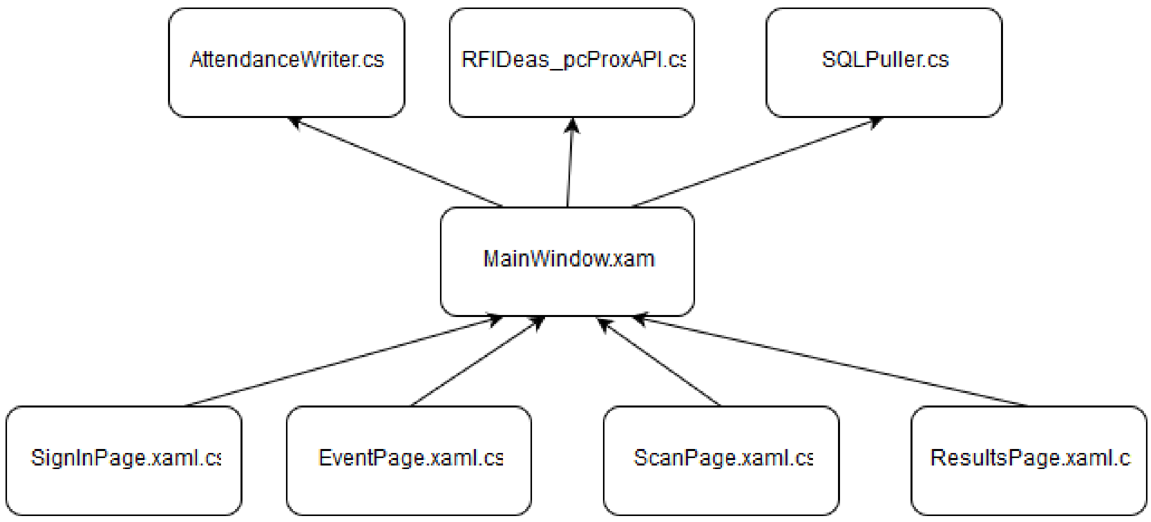
Checking Operation of the Application

All files will be found here: C:_system\

1. For attendance files: C:_system\attendance\
    a. Attendance text files will be generated with unique file paths
2. For Sql files: C:_system\db\
    a. Saved as Checkers.claw, Events.claw, Students.claw

Maintenance Guide:

CLAW Credit Checker Application Structure



Class Descriptions

1. MainWindow.xaml.cs - blank window that holds the various user control pages

Responsible for navigating between the different user control pages:
SignInPage, EventPage, ScanPage, ResultsPage

MainWindow is accessible from other classes by using:
MainWindow.AppWindow

MainWindow holds the attendanceWriter class so to access attendanceWriter one must use:
MainWindow.AppWindow.attendanceWriter

Otherwise you would be using a new instance of the attendanceWriter
MainWindow is also the first location in the program that checks for the
RFID USB device and sets the program up as device connected if it is found

2. AttendanceWriter.cs - class responsible for reading from and writing to text files

Responsible for creating directories and text files

MainWindow holds the attendanceWriter class so to access attendanceWriter one must use:
MainWindow.AppWindow.attendanceWriter
Otherwise you would be using a new instance of the attendanceWriter

Constants for directory names and file paths can be set, as well as the special character used for dividing the sql data text files

3. SQLPuller.cs - class responsible for pulling data from Gordon's attendum database and writing the results to text files using the AttendanceWriter

This class is created and used via the "Update Database" button in the
SignInPage

For updating to work, the user must be signed into the Gordon domain as a Chapel Checker, otherwise Visual Studio must be started remotely as a Chapel Checker

Pulls and Writes 3 Data files:
Events, Checkers, Students

4. RFIDeas_pcProxAPI.cs - class responsible for all RFID USB Device functions

5. SignInPage.xaml.cs - class responsible for both updating the database as well as using the RFID USB scanner to scan cards and check if the given card's id is a chapel checker's id

The results of a scan can be:
Person is a chapel checker - allowing you to proceed to EventPage

Person is not a chapel checker - keeps scanning
Person is not in the database - keeps scanning but prompts to update

6. EventPage.xaml.cs - class responsible for displaying all the events found in the events data file

Responsible for displaying all of the gordon events last pulled from the database

Allows the user to go back to the SignInPage or progress to the ScanPage if an event has been selected

7. ScanPage.xaml.cs - class responsible for displaying the event selected and allowing the user to initiate scanning, at any point a scan for no credit can be selected so the very next scan gives no credit, scanning can also be stopped at any time to progress to the ResultsPage

The results of a scan can be:
Person is not in the database - prompts to update database or ask person to go to CTS
Person is given credit - keeps scanning
Person is given no credit - keeps scanning
If a person has already received credit or nocredit, on a successive scan it will show that it knows the person has already received one of the credits

In order to prevent unnecessary beeping and duplicate entries the system will not display new information about a card that was just scanned unless a different card is scanned then information will be displayed upon scanning the first card again

8. ResultsPage.xaml.cs - class responsible for commanding final cleanup of the attendance file as well as displaying the number of students scanned and the time spent scanning

Class is not essential in the checking process but allows the user to know scanning is complete as well as return to the SignInPage to scan further