System Design Document 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: 12/15/2015

Document accepted on _____ by _____

(Date) (Departmental Representative)

Travis Pullen and Jonathan Manos 12/15/15

Chapel Credit Checker System Design

1. Introduction

1.1 Purpose of the system -

The purpose of this new implementation is to increase the productivity of checkouts at chapel events, decrease the costs of credit checking equipment, decrease the work needed to upload information, and add on more features to the new checking devices. The current credit checking system is expensive, outdated, slow, and it also causes lots of added work for the chapel office. This application could save the college money and time and reduce the traffic in bigger chapel events, which would also improve the student morale when attending chapel. For further information on the purpose of the system please refer to the requirements analysis and/or project proposal documents.

1.2 Design goals –

The system will be designed for a Windows 10 platform, so that it can be used on laptops, tablets, and phones though phone implementation may be later in the system's lifecycle. The system will also have an interface that is easy to use and quick to learn. The application main priorities are:

1. Quick processing of students – tap, beep, and go
2. Reliable storage and upload of event attendance information
3. Well written code and documentation for supportability
4. Simple, easy to understand interface for the checker

For further information on design goals refer to the functional and nonfunctional requirements listed in the requirements analysis document.

1.3 Definitions, acronyms, and abbreviations -

RFID - Radio frequency identification (RFID) is the wireless use of radio waves to transfer data, for the purposes of automatically identifying and tracking tags attached to objects. The tags contain electronically stored information.
CLAW - Christian Life and Worship

1.4 References -

1. Chapel Office Project Meeting Notes
2. Pluralsight.com C# Tutorials and Lessons
3. pcProx SDK examples
4. Jay Whitehouse

1.5 Overview -

The current CLAW credit checking system is outdated and in need of an overhaul being slow and over 10 years old as well as using more expensive and slower technology. An upgraded CLAW credit checking system that our team will develop will utilize faster

modern technology that also happens to be cheaper to improve costs and efficiency of chapel checkouts. Using RFID attachments to prox read Gordon IDs will drastically decrease checkout times and new devices will have a Wi-Fi connection to automatically upload student information to the Go Gordon server. The development process should be fun and challenging and provide a great benefit for the chapel office and the Gordon College student body.

2. Current software architecture -

The current system that Gordon College has in place is to scan the barcode on the back of the student id using Pocket PC's. This is very slow because the id needs to be handed to the chapel checker, then they have to hold the id at an angle that can be read by the scanner. Some id's have a faded barcode that causes the scanner to take a while to read the barcode, if it reads it at all. Also the Pocket PC's are expensive to upkeep and tend to not be as reliable as they should be.

3. Proposed software architecture

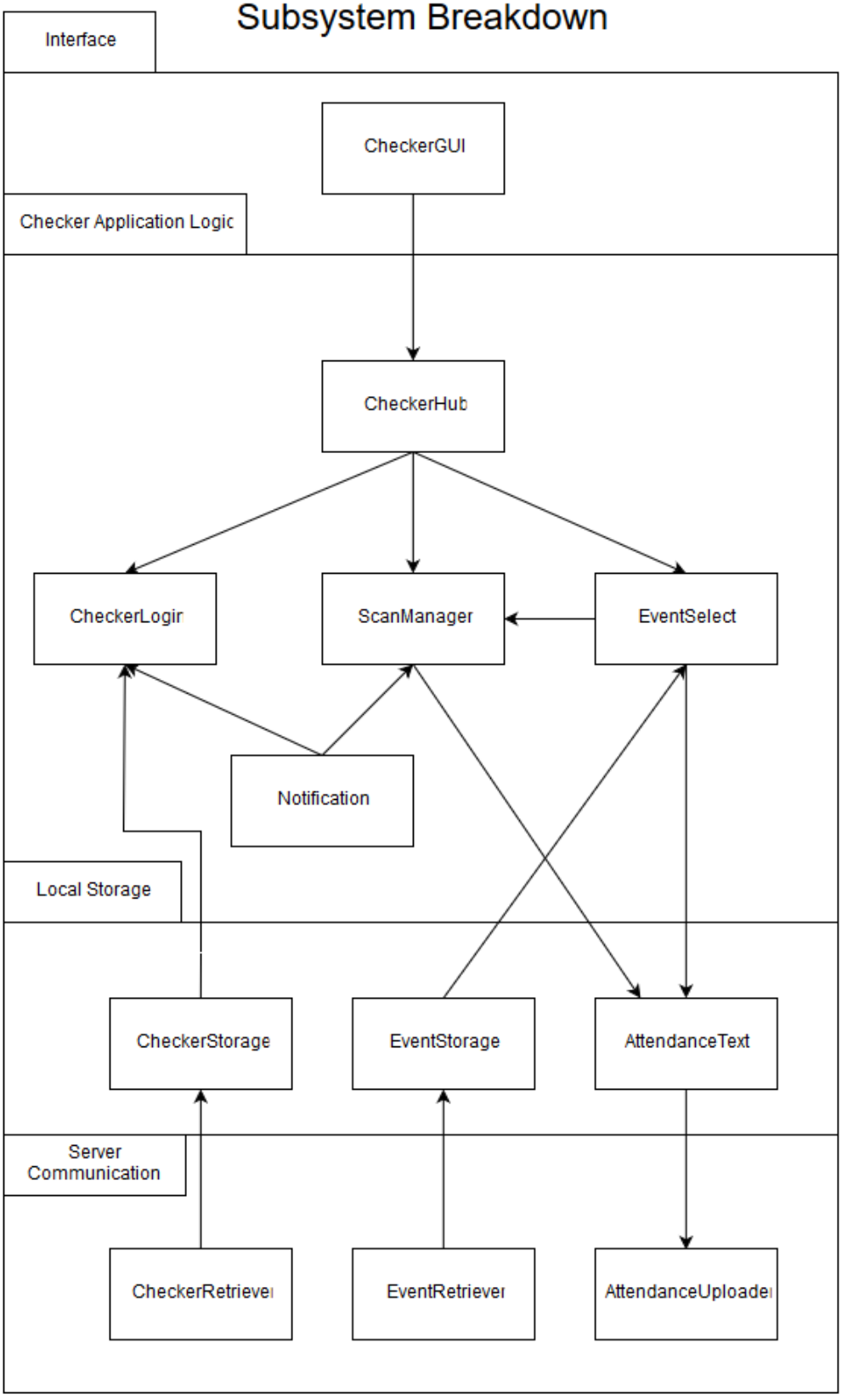
3.1 Overview –

The proposed system will speed up the scanning of students' ids and ease the process of uploading chapel attendance information. The system will use laptops/tablets (2 in 1 systems) that have an RFID USB plug-in. The checker will login to the application by scanning their own ID. The checker will then select the event that they are checking for and proceed to the scanning phase. The student will tap their card to the RFID sensor and the application will report audio and visual confirmation. The sensor will read the prox id and then the application can match it to the student ID, and record the student ID in the event attendance text document. After all the students have scanned, the checker ends the event which uploads the text document to Gordon servers to give students chapel credit.

3.2 Subsystem decomposition –

For the Chapel checking system there is need for a few subsystems. There is Gordon database, a subsystem in itself, that the chapel checking system will need to communicate with to retrieve and upload information. In the chapel checking system it will decomposed into four subsystems. A subsystem that focuses on the interface of the device and what the user interacts with. A subsystem that holds the logic of the checking system in terms of authentication, event selection, scanning, and writing to text file. A subsystem that holds the communication of the system from retrieving information from Gordon servers and uploading information to Gordon servers. And then one last subsystem that acts as a local database on the chapel checking system holding chapel checker information, event information, and attendance text files.

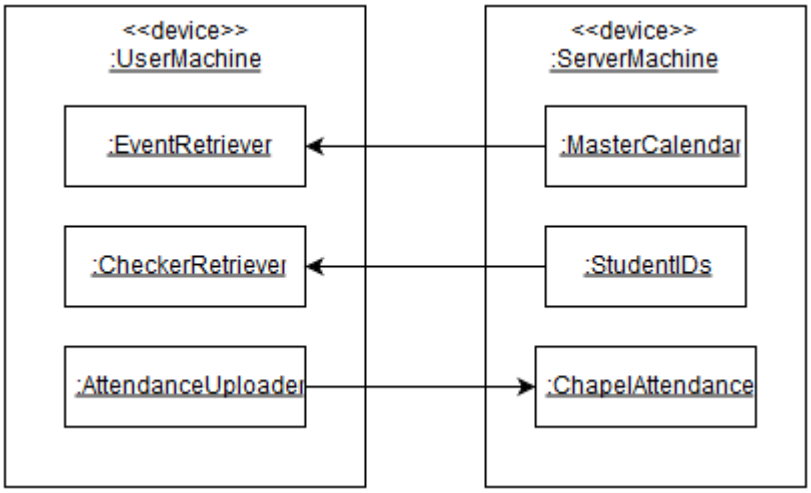
Chapel Checking System
Subsystem Breakdown



3.3 Hardware/software mapping –

The single chapel checking system will run on a windows 10 device as well as make use of a RFID prox reader USB attachment. A software application pcProxConfig that comes with the RFID reader attachment offers testing and scanning options with the RFID reader. To develop with the RFID reader's system, the USB device comes with pcProxSDK which has an API with commands for development. The chapel checking system will make use of the provided software and development resources.

The system will also communicate with Gordon Servers to access databases for information for the chapel system's operation. The system will access the master calendar for event information. The system will access Student IDs to translate hexadecimal prox IDs to student ID decimal numbers. The system will access the Chapel Attendance database on the Go Gordon server to upload event attendance. These are the areas where the system interacts with other hardware.



3.4 Persistent data management –

The text document for event attendance is only needed until the information is uploaded to the server so it is not persistent data.

The system normally will interact with Microsoft SQL databases to access student information and master calendar event information. To ensure the system is portable and does not necessitate a connection during operation, all chapel events should be stored in a database on the system. We will likely use a Microsoft SQL database or make a class for events to store event and checker id information.

The event database will only store events with chapel credit and will contain the ID, name, date, and time of the event. The date and time of the event can be used by the system to show which events are available for scanning. The ID of the event can be identifying the correct text document with the chapel attendance information. Checker information will just be a list of ID's that are approved chapel checkers. This data needs to be persistent so it will need to be saved on the application.

3.5 Access control and security -

Only chapel checkers will have access to the system and the computer will need network access to scan the checker's ID and confirm that the student has an active chapel checker student job. Who is on the chapel checker list is controlled by CTS and it is the responsibility of the chapel office and CTS to ensure only eligible student checkers are on the list. Otherwise a list of valid login ProxIDs would need to be stored locally on each checker's computer.

Security is not a significant issue for the project since the only information be transferred is public event information and student prox ID numbers. The attendance text file will contain the student prox ID, which is not the student's actual ID number, which will be translated when uploaded to the Gordon server. The prox ID number is unique to the ID card.

Access Matrix:

Actors ↓ Objects →	Login	Event	Scan	SaveUpload
ChapelChecker	Sign in by scanning card	Select event from current date	Scan people's cards by tapping them	Click done to finalize event processing

3.6 Global software control -

Connection issues should not be able to occur during the actual event. The checker scans their card which is authenticated by translating their proxID and verifying by communicating with Gordon servers. In case there is no connection the computer could locally store chapel checker proxIDs for the system to verify with. This avoids the issue of no connection for chapel checker authentication.

Synchronization and concurrency issues cannot occur during system operation since each system will operate individually and locally. The system will utilize event-driven control. The concurrency issue comes during the scanning of cards and how to deal with multiple people tapping their card at the same time. In that event, one card will be read or a misread will happen. In this case, people can rescan their cards just in case and since the system will check the text document of student information for duplicates. However, the system will have a cooldown after each scan so the likelihood of concurrent scans is unlikely. On concurrent uploads, the application should be able to handle an upload failure and try again periodically.

3.7 Boundary conditions –

On start-up of the system, the application will connect to the Gordon servers and attempt updating event and chapel checker information in the system's local database. Then the application should be in its initial state of needing chapel checker authentication to continue operation. The application will have event information and local checker ProxIDs loaded up. The system should show the chapel checker login page and should connect to the Gordon servers if connection is available. In the case that the system cannot

connect to Gordon servers, the locally stored ProxIDs of chapel checkers would be used to authenticate the user.

During normal operation of the application, in the case of a bad read of a card, the system will notify the user to rescan their card. A visual and audio cue will notify the user of a bad scan.

Before shut-down of the system, the application should have already saved and uploaded the attendance file and returned to the starting state of the initial login page. On shut-down, if the text document was not uploaded then the system should mark the text file so that on the next start of the system it attempts uploading the file.

4. Subsystem services - Not necessary as of yet.

Glossary

RFID - Radio frequency identification (RFID) is the wireless use of radio waves to transfer data, for the purposes of automatically identifying and tracking tags attached to objects. The tags contain electronically stored information.
CLAW - Christian Life and Worship
CTS – Center for Technology Services
USB – Universal Serial Bus