Jonathan Manos & Travis Pullen
10/28/2015
Software Project Management Plan

1. **Overview** - CLAW Credit Checking System Upgrade

1.1 **Project Summary**

1.1.1 **Purpose, scope, and objectives** - We propose to complete a new implementation of the CLAW credit checking system for Gordon College. The current credit checking system is expensive, outdated, slow, and it also causes lots of added work for the chapel office. 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 . We will develop an application for approved tablet/laptop chapel devices with attached RFID scanners. Project objectives in the application include:

1. Log-in for chapel checker by tapping card to authenticate user – or password lock
2. Select an event from the current time slot to give credits for
3. Quick processing of students – tap, beep, and go
4. Indication of successful scan with nice beep and code displayed
5. Blacklist button for late students to block credit for the student (needs confirmation)
6. Done button to end credit checking for the specific event
7. Automatic syncing and backup of stored student attendance for the chapel event

Certain specifications for this project may change over the course of development, but these are the general goals of the application. Along with these functional requirements, we aim to have the application follow many nonfunctional requirements including:

1. Ease of use – easy to understand and learn the functioning of the app
2. Performance – the application needs to be able to process at least 60 cards per minute, so the student can tap and go in under a second.
3. Reliable – the application successfully stores and uploads card information
4. Supportable – able to be upgraded and well documented

This is the intended course for our CLAW credit checking application for the chapel office.

1.1.2 **Assumptions and constraints** - Assumptions: Jay Whitehouse will provide us with the devices and prox readers we need for development. He will advise us with the best language and IDE for development of the system. He will also show us how to connect to and upload to the Go.Gordon server for uploading student chapel attendance information.

Constraints: The system needs to be developed before the 2016 Spring Semester, so it can be tested and fully implemented for the 2016 Fall Semester.

1.1.3 **Project deliverables** -

Deliverables	**Delivery Date**

System Design	December 4, 2015
System Implementation	January 8, 2015
Test Final Product	March 8. 2015
Final Product	April 8, 2015

1.1.4 **Schedule and budget summary** -

Milestone	**Date of Completion**	**Estimated Time (Actual Time)**
SPMP (Software Project Management Plan)	October 26, 2015	5 Hours (4 Hours)
SIS (Social Impact Statement)	October 26, 2015	5 Hours (2 Hours)
Rapid Prototype	October 26, 2015	2 Hours (2 Hours)
Requirements Analysis	November 9, 2015	10 Hours
User Manual	November 9, 2015	10 Hours
System Design	December 7, 2015	30 Hours
System Implementation	January 11, 2015	100 Hours
Test Final Product	March 11. 2015	5 Hours
Final Product	April 11, 2015	10 Hours

Overall budget for our project: Handled by CTS and the Chapel Office. They will provide the tablets/laptop and prox readers necessary for the implementation of our system.

1.2 **Evolution of the plan** - As we develop the application and learn more about the process needed to give chapel credit to students, the way this system is implemented will be updated. Since we met with the chapel office, we now have an idea of how the program should be structured. We will hold our documents in google drive and update them as necessary.

2. **References** - Chapel Office Project Meeting Notes
https://docs.oracle.com/cd/E19486-01/819-4686/EM-intro.html
http://docs.oracle.com/cd/E19486-01/819-4686/index.html

3. **Definitions** - RFID - **Radio-frequency identification** (**RFID**) is the wireless use of electromagnetic fields to transfer data, for the purposes of automatically identifying and tracking tags attached to objects. The tags contain electronically stored information.

IDE- An integrated development environment (**IDE**) is a software application that provides comprehensive facilities to computer programmers for software development.
SQL - (Structured Query Language) is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).
API - an abbreviation of application program interface, is a set of routines, protocols, and tools for building software applications.
CTS - Center for Technology Services
CASE - stands for **Computer Aided Software Engineering**. It means, development and maintenance of software projects with help of various automated software **tools**.

4. **Project Organization** - An IDE for mobile application development, preferably one with templates for RFID scanning and Microsoft SQL databases. Netbeans has a RFID Software Toolkit so we can likely use that.

4.1 **External interfaces** – Gordon College's Go.Gordon server which contains the list of chapel credits attached to each student's account. We also have to interact with the Chapel Office because it is their requirements that we need to meet.

4.2 **Internal structure** – Jonathan Manos and Travis Pullen will be the main members of the development, documentation, and testing group. Jay Whitehouse will act as a supervisor over our work and an advisor throughout development. Chris Hansen will be the direct client we will be working for so that he overlooks all of our work. Chris Hansen's client is the chapel office so he makes sure that our work is satisfactory to the chapel office.

4.3 **Roles and Responsibilities** - The development team has shared responsibilities of learning how the Gordon databases work and developing code to read the RFID signals from Gordon IDs and uploading it to a database that holds student's chapel credits.
The development team is dually responsible for studying Java API and RFID implementation sources on the web.
The documentation team is dually responsible for working on the necessary documents in development of the system.
The testing team is dually responsible for testing the application thoroughly through development to ensure its efficiency and reliability.

5. **Managerial process plans** - Jay Whitehouse will act as the manager on the project, ensuring that we are working in the right direction through our development. The chapel office is responsible for letting us know what features need to be included and Jay Whitehouse is responsible for looking over and directing our work.

5.1 **Start-up plan**

5.1.1 **Estimation Plan** - Throughout the next two semesters, we are going to have bi-weekly meetings with Jay Whitehouse to talk over the progression of the project and review the work that has been done so far, we will start by developing a scope of work for the project that the chapel office will review and approve. Using this the project is estimated to take 190 hours. We judged that by using our past experience with projects for class. These times will be tracked by us writing down the hours taken to complete each milestone.

5.1.2 **Staffing Plan** - Our staff will be Jonathan Manos and Travis Pullen and we will be in charge of development, documentation, and testing. We will be there for the entire duration of the project. Jay Whitehouse will act as our manager.

5.1.3 **Resource acquisition Plan** - We will need to talk with Chris Hansen or Jay Whitehouse to get our equipment (i.e the tablets/laptops), as well as the ability to access the database. They will also decided on the RFID attachment that we use for implementation of the RFID scanners.

5.1.4 **Project Staff Training Plan** - For training we will read up and practice the language that Jay Whitehouse decides is best to use. Before we start development we will do extensive studying of how RFID scanning hardware is implemented in software and will likely look at examples of RFID implementation to prepare ourselves for development.

5.2 **Work plan**

5.2.1 **Work activities** –

Task	**Dependency**	**Resource**	**Budget**
SPMP (Software Project Management Plan)	Meet with Chapel Office and CTS	Schach	0.00
SIS (Social Impact Statement)	Meet with Chapel Office and CTS	Shneiderman	0.00
Rapid Prototype	SPMP and SIS complete	Schach LucidChart.com	0.00
Requirements Analysis	SPMP and Prototype	Schach	0.00
User Manual	Requirements Analysis	Schach	0.00
System Design	Requirements Analysis, User Manual	Language + APIs IDE Tutorials Schach	0.00
System Implementation	System Design, Requirements Analysis, User Manual, Devices	Device Manuals IDE Tutorials Schach	Cost of 2 prox readers and 2 laptops/tablets for development
Test Final Product	System Implementation	Schach	0.00

Final Product	Test Final Product, SIS	Schach	Any final devices or attachments needed

5.2.2 **Schedule allocation** – Any other dependencies for work tasks will be updated throughout system development.

5.2.3 **Resource allocation** – Resources will be updated throughout system development.

5.2.4 **Budget allocation** - The budget is being handled by the chapel office and CTS.

5.3 **Control plan**

5.3.1 **Requirements Control Plan** – We will prioritize features that should be in the project. There will be features that are must haves and some that would be nice to have, but do not need to be there. We will give these features a grade of either 1 or 2. 1 will be for the must haves, and 2 will be for the nice to have features.

5.3.2 **Schedule Control Plan** – We will be updated via email if the chapel office has any more requirements that they think should be added. Also Jay Whitehouse will let us know if he hears anything,or sees something that should be include in the project, during our bi-weekly meetings.

5.3.3 **Budget Control Plan** – We will be provided with the equipment that is necessary for the project and have no control over the budget that will be spent on the project.

5.3.4 **Quality Control Plan** – We have bi-weekly meetings with Jay Whitehouse. He will look over our work and let us know if our work is up to his standards. Then the project will get passed to Chris Hansen and the chapel office so that they can look at the work and let us know if anything is not up to their standards.

5.3.5 **Reporting Plan** – The stakeholders for the project are the entire Gordon College undergraduate community and the chapel office. We will be reporting all of our information to Jay Whitehouse who will then tell Chris Henson and the chapel office about the progress of the project. If needed we can also have meetings with the chapel office to show them the progress of the project.

5.3.6 **Metrics Collection Plan** – The metrics of the project will fall into a few categories, work, reviews, and changes. For work, the metrics will be collected as hours that we have spent working on completing the project. For reviews this will be done using a separate document that is just for errors that we run into while executing the project, or bugs. Lastly, changes will be another separate document that will contain all the changes and additional requests that the chapel office wants to be included in the project.

5.4 **Risk management plan** - We will have a list of potential risks and how they should be handled. If we come across one that we do not already have listed we will add it to the list and give it a grade on how impactful it is. When we have our bi-weekly meetings we will go over risks and any new ones that we have come across.

5.5 **Closeout plan** - Once the project is completed Jonathan Manos and Travis Pullen will be done working for the team and the completed project will then be handed off to the chapel office so that they can implement the new system and train their employees.

6. **Technical process plans**

6.1 **Process model** - We will use the rapid-prototyping life-cycle model due to the ever changing requirements of our chapel credit checking system. Through development if there are any changes, we can alter the rapid prototype or make a new one altogether to continue working on the system. It is most important that our client is satisfied by the prototype of the system that we will develop.

6.2 **Methods, tools, and techniques** - The language for development has not been decided on yet, Jay Whitehouse will decide on the language that is best suited for RFID development. We will likely develop the GUI first and then develop the functionality for the features of the system.

6.3 **Infrastructure plan** – The infrastructure needed will be around 10 tablet/laptop devices and 10 RFID scanner attachments. These will likely run on an android or windows tablet OS and will be connected to Gordon Net's wifi. The software necessary will be the application we develop along with any software that allows the RFID scanners to work. For a CASE-tool we will use an IDE, likely NetBeans, that is suitable for development of RFID scanning software.

6.4 **Product acceptance plan** - We are to document a scope of work that we will submit to the chapel office, once the reviews are made, we will likely write up another scope of work that is then approved by the chapel office. Afterwards we will show them a rapid prototype to see if they approve of the implementation. Then in our development we will ensure that we go with the criteria that we established in the scope of work and the rapid prototype.

7. **Supporting process plans**

7.1 **Configuration management plan** – Meeting with chapel office to see how they want the application configured and see if they want any additional features that they don't already have.

7.2 **Verification and validation plan** – We will show our work to our other team member Jay Whitehouse to see if it is technically sound, then show to Chris Hansen to show him the functionality and to see if it meets his standard, and then show to the chapel office to see if the system satisfies them and to see if they want any additional small features.

7.3 **Documentation plan** – Work on documents together, using Google docs, so that we think of everything that we need to include. We will update the date on the documents as well, even though google drive keeps record of the changes and who did them. We will also keep copies of past documents in case of any surprise needs.

7.4 **Quality assurance plan** – Test out the application with a few cards at first and then try a small test event, then show the chapel office the same process and how to use it to ensure it meets their standards.

7.5 **Reviews and audits** - We will meet with Jay Whitehouse bi-weekly to show him our progress. He will tell us anything that we need to fix and possible ways of fixing the problem. Once we have been told what should be fixed we will find the next best time that we can get together to work on the project and schedule the time.

7.6 **Problem resolution plan** – Once a problem is brought to the team's attention, the team will meet up and talk about the problem and other things that are affected by this problem. Once the problems have been discussed then the team will discuss how to resolve the problem. Then once the team has come up with a resolution they will fix the problem and continue on with the project as planned.

7.7 **Subcontractor management plan** – Jonathan Manos and Travis Pullen are the subcontractors for this project. We will be responsible for the development of the system. We do not have any subcontractors below us.

7.8 **Process improvement plan** - If we designate a time to meet to perform our weekly documenting and development, it will streamline the whole development process. Working on random shifts can lead to falling behind.

8. **Additional plans** – No additional plans at the present. There is a small possibility that the chapel checking device be on a small laptop as opposed to an android device. Update: the application will now be developed for a tablet or small laptop.