**SolarProx**

**Force of Mercury**

**Trevor Ryan**

**Austin Carper**

**Kurt Neimayer**

**Jacob Sanford**

**Design Document #1**

****

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Main Author(s) | Version Description | Date Completed |
| 1.0 | Trevor Ryan  Austin Carper  Kurt Neimayer  Jacob Sanford | Design Document #1 Rough draft. This is the initial version for Design Document #1 | 02/28/2021 |

**Table of Contents**

[**1.0 Introduction**](#_heading=h.awttgun60cha) **4**

[**1.1 Overview**](#_heading=h.yp2mvetpe15d) **4**

[**1.2 Deliverables**](#_heading=h.k12beldy866i) **4**

[**1.3 Project Schedule**](#_heading=h.j0pxbpad3fsp) **5**

[**1.4 Definition of Terms**](#_heading=h.olsjnpyuhw9i) **6**

[**2.0 Project Organization**](#_heading=h.t0hh8x1atokq) **7**

[**2.1 Software Life Cycle Model**](#_heading=h.e24z5mnzwd2a) **7**

[**2.2 Team Responsibilities**](#_heading=h.97z5w2yoh6ur) **7**

[**3.0 Managerial Process**](#_heading=h.nhol72ppvid5) **9**

[**3.1 Team Reporting/Monitoring**](#_heading=h.646mex6vgaef) **9**

[**4.0 Technical Process**](#_heading=h.9osu6uj4vyry) **10**

[**4.1 Tools**](#_heading=h.4ow5145p4lqx) **10**

[**4.2 Documentation Strategy**](#_heading=h.4ow5145p4lqx) **10**

[**5.0 Risk Analysis**](#_heading=h.ocl75gewrv6u) **11**

[**5.1 Risk Identification**](#_heading=h.od81lm1w39fb) **11**

[**5.2 Risk Planning**](#_heading=h.o9z9a7yafhdt) **11**

[**6.0 Design Architecture**](#_heading=h.3y68jjlplu4s) **13**

# 1.0 Introduction

## 1.1 Overview

SolarProx is a web application that will implement the Proxmox interface into a web application where students, who will be the primary users, can revert machines back to a certain state, view notes created by the administrator, and enter flag values into a field to earn points for each box. It will be designed for the students in network security classes at Mount Union but can also be used with whatever group the administrator would like. The application will run on our client’s virtual lab which is the same lab that is used currently for the network security classes. The application will also allow the administrator access to add and remove machines from the active machine pool, enter notes for each machine, add and remove users, along with being able to create snapshots for each machine.

This Design Document will be used to explain all of the high-level functions of our program along with the methodology we will be using throughout our development. This document will show identified risks that could hinder our project along with analyzing these risks along with explaining how we can prevent these risks. This document will also show our project’s data flow diagram and how we expect our program to flow.

## 1.2 Deliverables

This section outlines the due dates for documents that will be created throughout the build process.

**January:**

January 13th: Project topic selection

January 27th: Software Requirements Specification draft due

**February:**

February 8th: Software Requirements Specification final version due

February 10th: Press release #1 draft due

February 10th: Project schedule draft due

February 15th: Press release #1 final version due

February 17th: Project schedule final version due

February 22nd: Introductory project presentation

**March:**

March 1st: Design document #1 draft due

March 8th: Design document #1 final version due

March 22nd: Design document #2 draft due

March 29th: Design document #2 final version due

March 31st: Status meeting/presentation

**April:**

April 7th: Press release #2 draft due

April 12th: Press release #2 final version due

April 14th: User’s Manual draft due

April 21st: User’s Manual final version due

April 28th: Final report due

## 1.3 Project Schedule

This section outlines the due dates for the project itself.

**January:**

**February:**

February 12th: Data diagram

February 12th: UML diagram

February 12th: Website sketches

February 15th: Review website layout with client

February 19th: Establish basic Proxmox connection

February 19th: Establish basic Active Directory connection

February 26th: Build data containing classes

February 26th: Build page layout prototype

**March:**

March 1st: Meet with client for product review

March 19th: Build website and attach to tomcat

March 26th: Quality assurance testing

March 31st: Submit initial test to client

March 31st: Prototypes due

**April:**

April 2nd: Penetration testing for feedback

April 20th: Fix errors from feedback

April 24th: General improvements

April 26th: Project due

## 

## 1.4 Definition of Terms

* Penetration Testing - A security term for finding vulnerabilities and documenting how they were found and what damage can occur by exploiting them
* Proxmox - Software used to help manage a group of virtual machines
* Active Directory - A system on Windows servers that is used to help manage permissions and users over a network of machines.
* Virtual Machine - Software that creates another instance of an operating system to emulate a computer without being a separate physical machine
* Flag- A flag is a value stored in a .txt file within a machine for users to find. The flag value can then be placed in a field so that the user can earn points

# 

# 2.0 Project Organization

## 2.1 Software Life Cycle Model

Force Of Mercury will be using the Design to Schedule Life Cycle Model with parts of the Evolutionary Delivery Life Cycle Model. SolarProx will have certain features that need to be functional within a certain time so having the Design to Schedule Life Cycle Model will ensure that the highest priority features will be completed within the given time period. Force Of Mercury also chose to incorporate parts of the Evolutionary Delivery Life Cycle Model so that certain features can be reviewed with our client.

## 2.2 Team Responsibilities

* All Students
  + Stay in contact with the group
  + Work together to complete required documents
* Trevor Ryan - Team Leader
  + Organizes meeting times
  + Communicates with client
  + Has the final say on documents before submission
  + Divides up work when necessary
  + Updates the Kanban board with new assignments as they are produced
* Jacob Sanford - Lead Architect
  + Lead programmer in the project
  + Focused on managing the general code structure of the project including producing diagrams
  + Refactor code to be easier to read and more efficient
* Kurt Neimayer - Lead Tester
  + Writes unit tests and runs them to detect potential issues
  + Test the site for unexpected issues before initial release
  + Report all issues to the team in a timely manner
* Austin Carper - Configurations Manager
  + Manages new commits to GitHub and general maintenance.
  + Maintains and produces blog post
  + Manages storing and backing up important documents
* Ken Smith - Client and lab host
  + Provided how he wants the project to look
  + Provided what desired features should be in the initial release
  + Provides the team with the lab environment to host the web application
  + Reviews features and decides if they are complete or if they need more work

# 

# 3.0 Managerial Process

## 3.1 Team Reporting/Monitoring

In order to judge our progress for the project, Force of Mercury will present any progress to the client for the client to view. After getting feedback, Force of Mercury will either decide whether that feature is ready to go or if that feature needs to be worked on some more. If the feature is good to go, Force of Mercury will then move onto the next feature for the project. These meetings will occur for each feature until the project is ready to be completed.

# 4.0 Technical Process

## 4.1 Tools

* GitHub - Used for version control, configurations management, host our blog, and maintaining a Kanban board
* StarUML - Used for documenting code and producing diagrams.
* Maven - Used for building compiled code and controlling dependencies
* IntelliJ IDE - Used as the development environment to write the software.
* Discord - Used to contact the client as well as communicate as a team and store documents.
* Windows Server 2016 - Used to control active directory.
* OpenVPN - Allows the connection to the lab environment from outside the network.
* Apache Tomcat 9 Server - Used to run the software and manage development
* Proxmox - Used to manage virtual machines within the network including restarting them and producing or loading snapshots of the virtual machines

## 4.2 Documentation Strategy

* Log information in the StarUML file to allow for documentation of all classes and methods. These will then be easily navigable on a local site view, allowing for information about the methods as well as additional comments and information about how they function. This information will be publicly accessible at the submission of the project.
* Code will consist of having comments when necessary for code that may appear complex or use a mixture of additional languages such as making machine calls to Active Directory or Proxmox to help understand the code when looking at the source.
* GitHub commits will contain information about what changes happened on the branch to allow further information about what changes occurred without needing to read the source code.
* Produce easy to read code by following proper naming conventions with detailed names for methods, classes, and variables to produce easier to read code. All code will also be formatted using IntelliJ’s automatic formatter to produce similar spacing to enforce readability.

# 5.0 Risk Analysis

## 5.1 Risk Identification

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk Name** | **Likelihood** | **Impact** | **Overall Risk** |
| Getting caught up on learning new materials | High | Medium | High |
| Procrastination | Medium | High | High |
| Mount Union’s internet going down short term | High | Low | Medium |
| Mount Union’s internet going down long term | Low | High | Medium |
| Client’s lab becoming damaged | Low | High | Medium |
| Anyone in the group becoming sick and unable to work at full capacity | Medium | Low | Low |
| Anyone in the group getting caught up at their jobs | Medium | Low | Low |
| Client’s lab environment going down and we are unable to contact him | Low | Medium | Low |

## 5.2 Risk Planning

* **Getting caught up on learning new materials:**
  + Ensure we are learning new material quickly
  + Don’t get too caught up learning new materials
  + Assign who does what so others can move on
  + Help each other understand the material
* **Procrastination:**
  + Ensure we are sticking to schedule
  + If we end up off schedule, work hard to get back on schedule
* **Mount Union’s internet going down short term:**
  + Contact professor and client to let them know what is happening
  + Work on what we can without internet
  + Switch over to ethernet to continue working or go to Mount Union’s computer lab
* **Mount Union’s internet going down long term:**
  + Contact professor and client to let them know what is happening
  + Figure out when internet goes back online
  + Go to Mount Union’s lab or connect to ethernet on campus
* **Client’s lab becoming damaged**
  + Figure out the extent to the damage
  + Find out how long the lab will be unavailable
  + Work on project locally for the time being
* **Anyone in the group becoming sick and is unable to work to their full capacity:**
  + Person who is sick should do as much as they are able to do
  + Everyone else should work normally and possibly pick up some of the tasks that were assigned to the sick person
* **Anyone in the group gets caught up at their jobs:**
  + That person should make up their time
  + Everyone else should work normally
  + Others can help the person who is caught up at work with their tasks
* **Client’s lab environment goes down and we are unable to contact him:**
  + Keep reaching out to the client until we figure out what is going on
  + Figure out how long it will be down once we get in contact with the client
  + Continue doing work outside of the lab environment

# 6.0 Design Architecture

The design documents for the Data Flow Diagram, Package Diagram, and UML diagram per package can be accessed through the StarUML local based website zip. The files of the diagrams can be accessed via our blog website under the second blog post. To use the site, launch the file called index, and use the navigation controls on the left-hand side to adjust which diagram and information you are looking at.