Design Document

For

Private Environment Network

Version 1.0

Prepared by Eric Webb

Nova Southeastern University

11-17-2019

Table of Contents

Table of Contents 2

1. Introduction 3

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Intended Audience. 3

1.4 Project Scope 3

1.5 References 3

2. Requirements 4

2.1 Product Perspective 4

2.2 Product Features 4

2.3 Two Use Case Examples 4,5

2.4 Use case Diagram 6

2.5 Activity Diagram 7

2.6 List of Items 7,8

2.7 System List 8

2.8 Class Diagram with Dependencies and Multiplicities 9

3. Design 10

3.1 Data Structures and Objects 10

3.2 Architectural Style Pattern 10

3.3 Architectural State Diagram 10

3.4 Overall Architectural State Diagram 11

3.5 Class OCP Diagrams 12

3.6 Use Case OCP Diagrams 13

4. User Interface 14

4.1 Layout Screen 14

4.2 Layout Screen Sequencing 14

5. Testing 15

5.1 Unit Tests 15

5.2 Testing Environments 16

6. Certficate of Authorship 17

# 

# 1. Introduction.

## Purpose.

The purpose of this document is to give the reader a sense of knowledge on how the PEN version 1.0 program works. This could include developers trying to replicate this application, project managers trying to get a better sense of the project, or even testers to better understand how to test this application.

## Document Conventions.

In this document moving forward we will regard the private environment network as a PEN. The guided user interface that users will log into via their devices will be referred to as the GUI.

## Intended Audience.

The intended audience of this document could include a multitude of professional. This could include developers trying to replicate this application, project managers trying to get a better sense of the project, or even testers to better understand how to test this application.

## Project Scope.

The goal of this application will be to have a working private environment network. This PEN will be attainable through agile development and methodologies to properly communicate from the stakeholders to the development team and vice versa. The working PEN will incorporate communications features that are manageable by tiered accounts.

## References.

To begin this project research was done in the topology and routing of MANET’s to familiarize with issues and short fallings incorporated with similar designs and how they were overcame. Building from that time was dedicated into getting familiar with agile methodologies and concepts. These can be further elaborated in the references below.

Gelperin, D. Exploring agile (2008, May 10). doi: 10.1145/1370143.1370144

Medrano-Chavez, A. G., Perez-Cortes, E., & Lopez-Guerrero, M.

Studying the Effect of Human Mobility on MANET Topology and Routing: Friend or Foe? (2015, November 2). doi: 10.1145/2810362.2810370

# Requirements.

## Product Perspective.

The overall perspective of this product will be for consumer private use. The consumer of this product will have full rights to deploy and maintain an instance of this PEN for their own benefit. Because this product is a multiuser product the PEN will have a multiuser product perspective.

## Product Features.

The PEN offers some unique features to make it complete. The first features is that of sensors to locate physical devices. The second feature of these sensors is their capability to route traffic and update nodes as devices traverse the PEN. More features include the ability for devices to link to the PEN’s sensors and a simplistic but effective GUI for users to interact with.

## Two Use Case Examples.

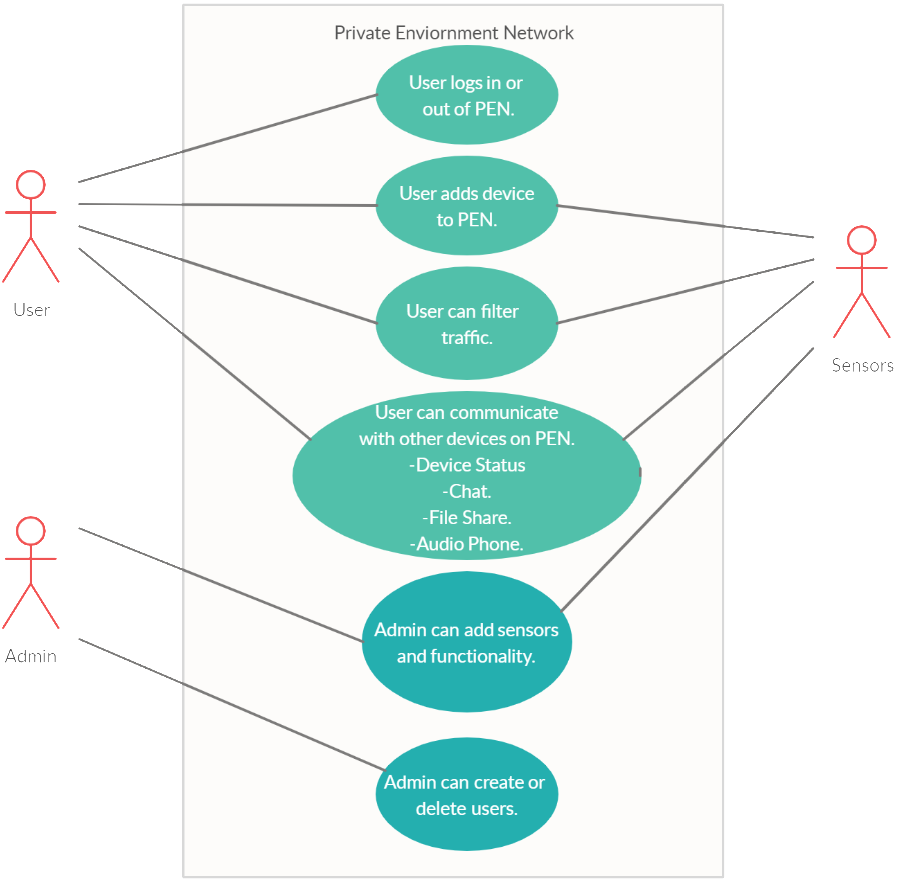
Below are two use case examples of adding a device to the PEN and the phone service.

|  |
| --- |
| **Use Case:** Adding device to PEN  **Primary Actors:** User  **Goal in context:** To add user devices to the private environment network  **Preconditions:** User has account for PEN and has a device on a network that can reach PEN.  **Triggers:** User decides to add their device to pen.  **Scenario:**  User - logs into PEN GUI, Clicks add this device.  Device – Connects to sensors.  User – Verifies PEN web GUI has new device added.  **Exceptions:**  Device cannot connect, verify connection to PEN web GUI, try to add device again.  **Priority:** Essential.  **When Available:** First Increment.  **Frequency of Use:** Only once when connecting device to PEN.  **Channel to Actor:** Via Pen GUI.  **Secondary Actors:** Device to be added, Sensors, System Admin  **Cannels to Secondary Actor:**  Device - Connection to PEN  Sensors – Hardwired and radiofrequencies.  System Admin – Configures PEN for user to be added.  **Open Issues:**  Should there be an indication if a device is online or not?  Should there be a timeout of device on PEN if not used for a while? |

|  |
| --- |
| **Use Case:** ad-hoc phone service.  **Primary Actor:** User  **Goal in context:** A User should be able to call another device in the PEN.  **Preconditions:** Both devices are actively connected to PEN.  **Triggers:** User decides they want to make a call to another user’s device.  **Scenario:**  User1 – See’s User2’s device actively on the network  User1 – Click the audio conference option in PEN GUI for this device.  User2 – Receives request for audio and selects “accept call.”  User1 – Communicates with user2 via audio, session is cancelled when either party ends call.  **Exceptions:**  User2 does not answer call, User1 sees response in PEN GUI saying User2 was not able to answer.  Call gets disconnected, verify both devices are available and re-establish call connection.  **Priority:** Feature chosen at will of user.  **When Available:** After both user devices are connected to PEN.  **Frequency of Use:** Multiple times a day.  **Channel to Actor:** PEN GUI displays call user device option over selected device.  **Secondary Actors:** Second User, Devices, Sensors.  **Cannels to Secondary Actor:**  Secondary User - logs in and connects second device to be called.  Devices - connects to PEN.  Sensors –Knows device is connected and routes information accordingly. (In this case phone traffic.)  **Open Issues:**  Should users have a call log of received and missed calls?  Should there be options to record audio?  Should a timer be set to record how long conversation was? |

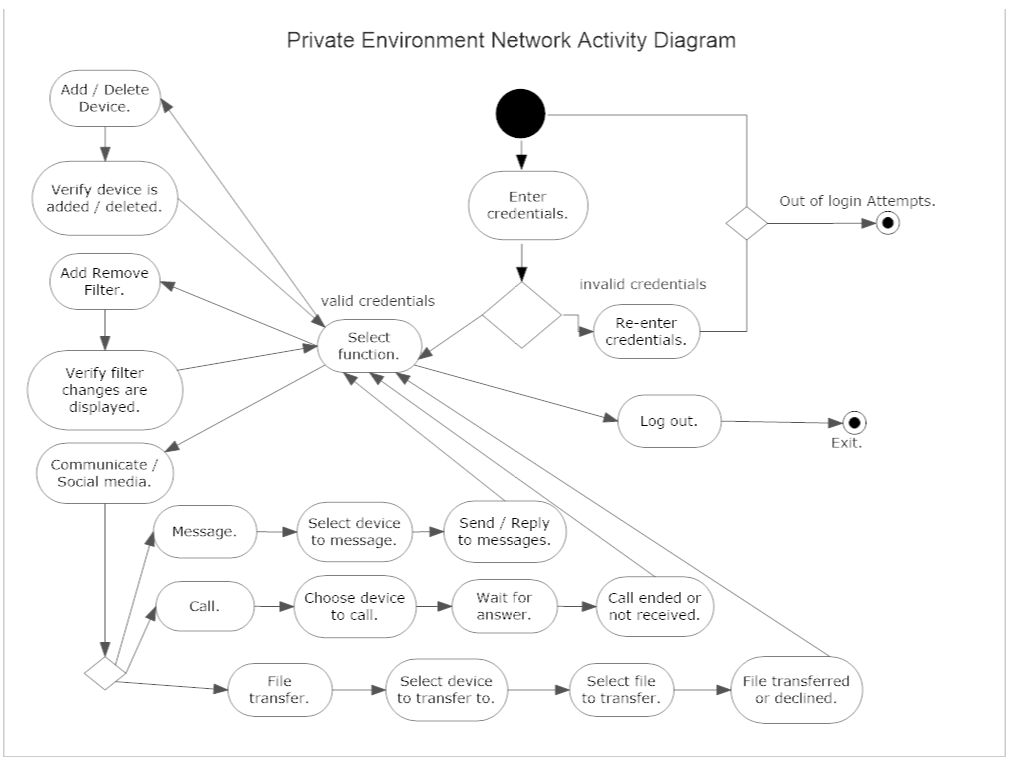
## Use case Diagram.

Below is a high level use case diagram showing a User, Admins, and Sensors as actors with their functionalities to be used.



## Activity Diagram.

Below is an activity diagram representing the activities and flow that will be performed in the PEN. Essentially giving the ability to loop through all functionality till a user is logged out.



## List of Items.

Below is a potential list of items for the system and their general classifications.

|  |  |
| --- | --- |
| **Potential Class** | **General Classification** |
| Admin | Role. |
| User | Role. |
| Username | Thing. |
| Password | Thing. |
| Web GUI | Thing. |
| User Device | Thing. |
| Sensor | External Entity. |
| Installation | Occurrence. |
| Sensor Event | Occurrence. |
| Routing Service | External Entity. |

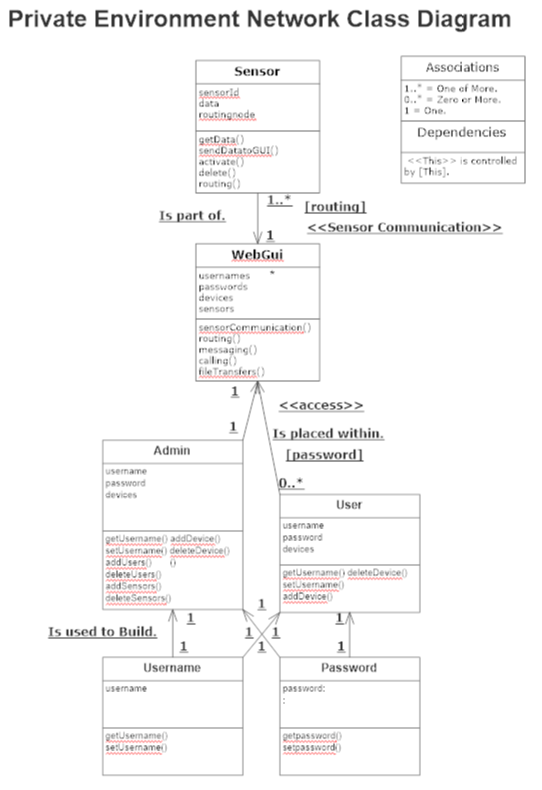
## System List.

Below is a list of attributes to be acknowledged in the system.

|  |
| --- |
| **System** |
| Username |
| Password |
| Email |
| Number of Login Attempts |
| Status of Owner Devices |
| logIn() |
| logout() |
| addDevice() |
| deleteDevice() |
| selectADevice() |
| showDevices() |
| sendMessage() |
| checkMessages() |
| deleteMessages() |
| callDevice() |
| acceptCall() |
| denyCall() |
| hangUpCall() |
| fileTransfer() |
| acceptFileTransfer() |
| denyFileTransfer() |

## Class Diagram with Dependencies and Multiplicities.

Classes will include Sensors, GUI, User, Admin, Username, Password represented in the PEN class diagram represented below. Routing is dependent on Sensor Communication, while GUI access is dependent on the correct password.



# Design.

## Data Structures and Objects.

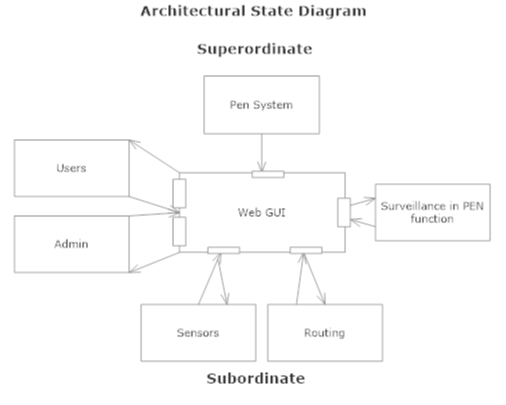
* User and Admins will be objects stored in an ArrayList.
* Sensors will be objects stored in an ArrayList.
* Username and Passwords will be strings stored in key value mappings.

## Architectural Style Pattern.

The PEN will use Main Program Subprogram Architecture. This was chosen because the main program like the GUI invokes other smaller programs like the calling functionality of the sensors and traversing data back to the devices.

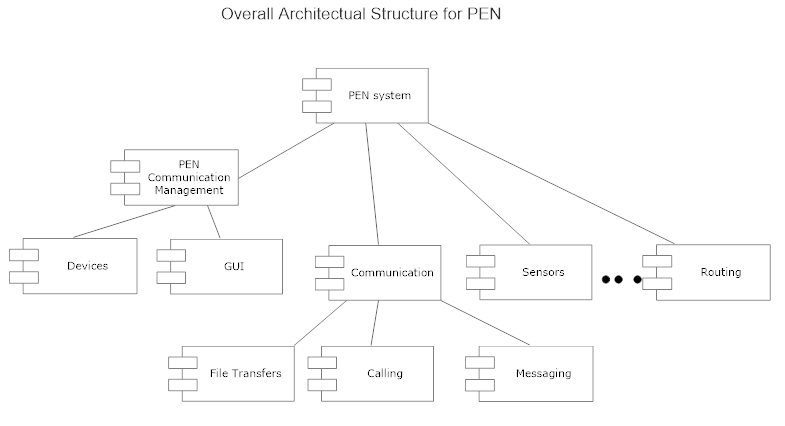
## Architectural State Diagram.

Below is an Architect state diagram reflecting the system and a superordinate and the sensors and routing as subordinates.



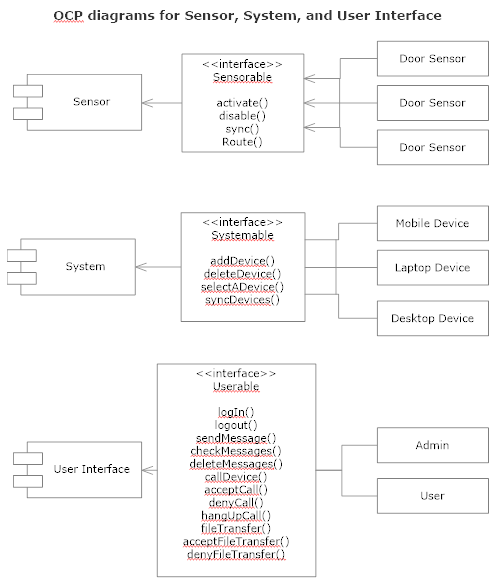
## Overall Architectural State Diagram.

Below is the architectural state diagram reflecting routing and sensing being shared by the same device. Communication handles calling, messaging, and file transfers. The GUI and devices are interwoven into the PEN communication management.



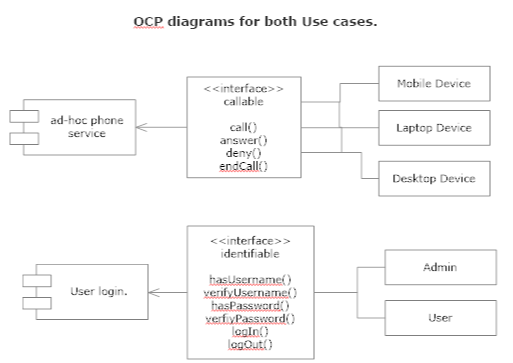
## Class OCP Diagrams.

In the Open close principal diagrams below we see the sensor, system, and user interface. These are shown with their methods and what can use them.



## Use Case OCP Diagrams.

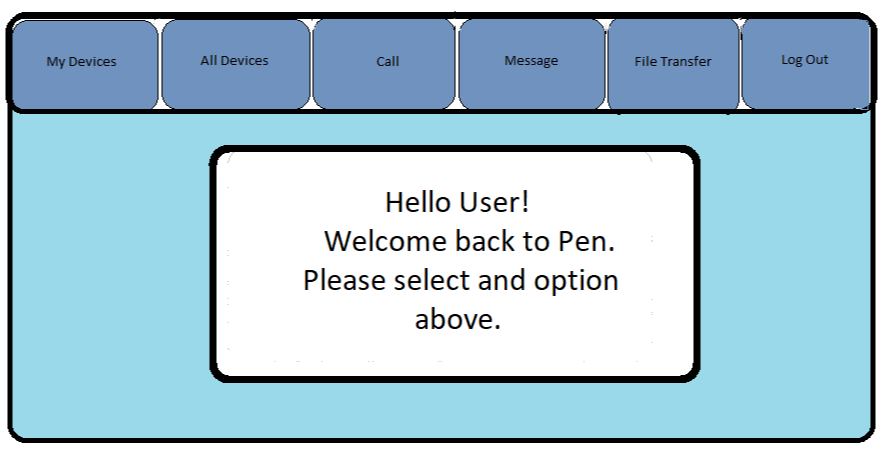
Below is the open close path for the two use cases mentioned in the requirements section above.



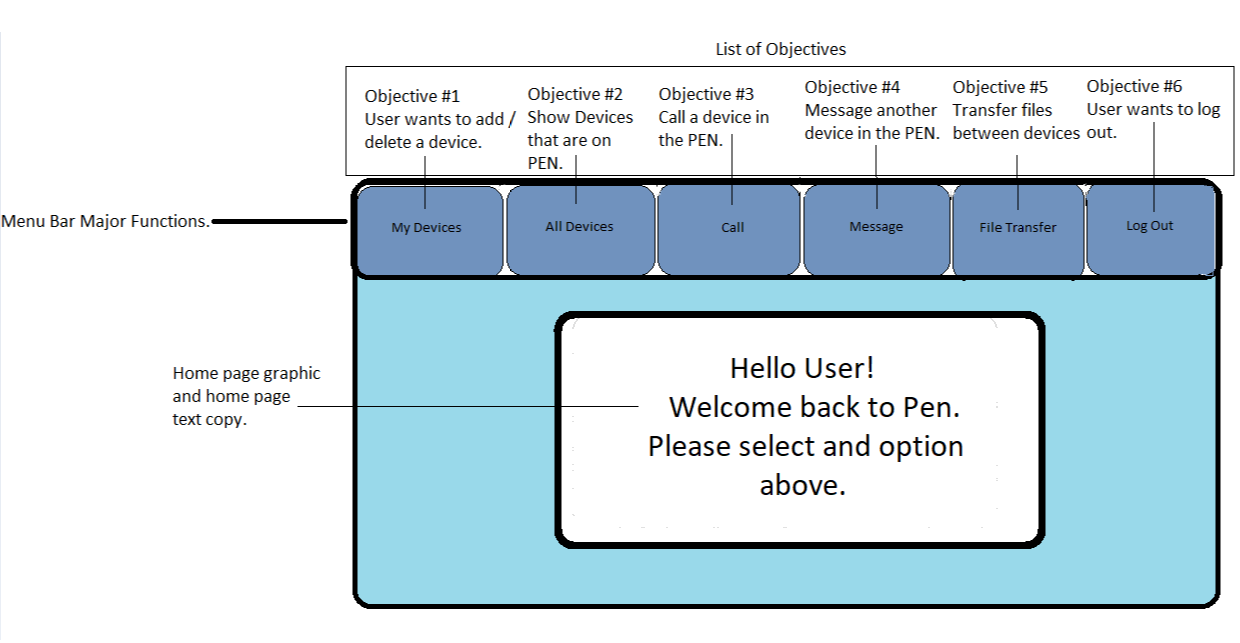
# User Interface.

## Layout Screen.

Below is a fundamental user interface design, There will be a main content area and a top navigational bar. From this bar users will be able to select their devices, all devices, call, message, file transfer, and logout.

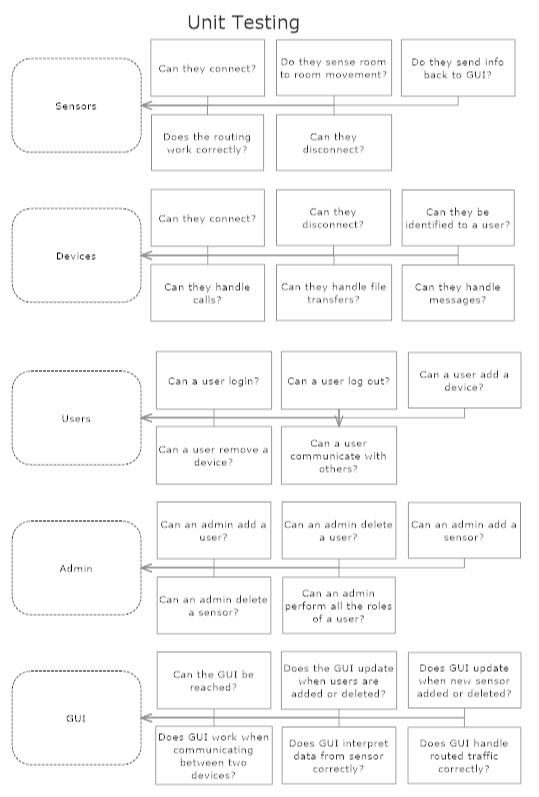


## Layout Screen Sequencing.



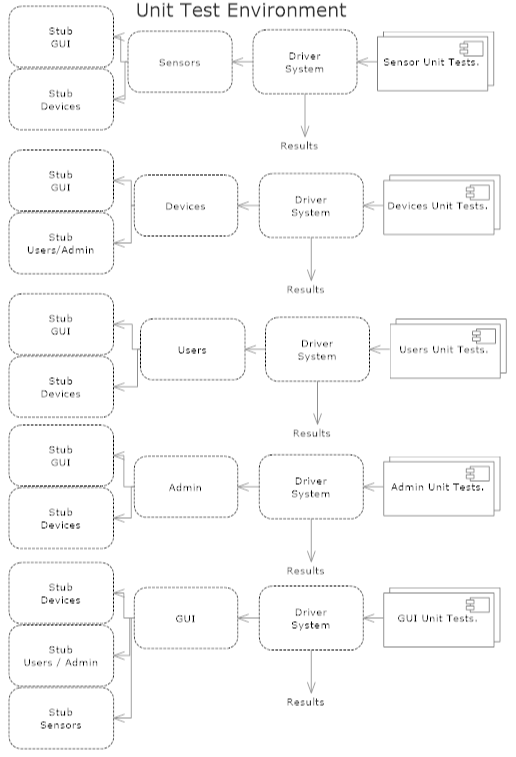
# Testing.

## Unit Tests.

Classes and their testing is illustrated below. Each class will have different tests to adhere to but the Admin class will have all the testing of the User class plus more testing.

## Testing Environments.

Here we see the Driver system administering the tests to the correct classes and what stub programs get called.



**Certification of Authorship**



Submitted to: Professor Jeffrey Kane.

Student’s Name: Eric Webb

Date of Assignment: 11/17/2019

Title of Assignment: Assignment No. 3 – Project

Certification of Authorship: I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas, or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for this course.

Student's Signature: ERIC WEBB