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 ii

Revision History ii

1. Introduction 2

1.1 Purpose 2

1.2 Document Conventions 2

1.3 Project Scope 2

1.4 References 2

2. Overall Description 2

2.1 Product Perspective 2

2.2 User Classes and Characteristics 2

2.3 Operating Environment 2

2.4 Design and Implementation Constraints 2

2.5 Assumptions and Dependencies 2

3. System Features 2

3.1 System Feature 1 2

3.2 System Feature 2 (and so on) 2

4. Data Requirements 2

4.1 Logical Data Model 2

4.2 Data Dictionary 2

4.3 Reports 2

4.4 Data Acquisition, Integrity, Retention, and Disposal 2

5. External Interface Requirements 2

5.1 User Interfaces 2

5.2 Software Interfaces 2

5.3 Hardware Interfaces 2

5.4 Communications Interfaces 2

6. Quality Attributes 2

6.1 Usability 2

6.2 Performance 2

6.3 Security 2

6.4 Safety 2

6.5 [Others as relevant] 2

# 

# 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 viea their devices will be referred to as the GUI.

## Intended Audience.

The intended audience of this document could include a multitude of professional positions. 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. 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.

Studying the Effect of Human Mobility on MANET Topology and Routing: Friend or Foe?

doi>[10.1145/2810362.2810370](https://doi-org.ezproxylocal.library.nova.edu/10.1145/2810362.2810370)

Exploring agile

doi>[10.1145/1370143.1370144](https://doi-org.ezproxylocal.library.nova.edu/10.1145/1370143.1370144)

# 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 simplistic but effective GUI for users to interact with.

## Two Use Case Examples.

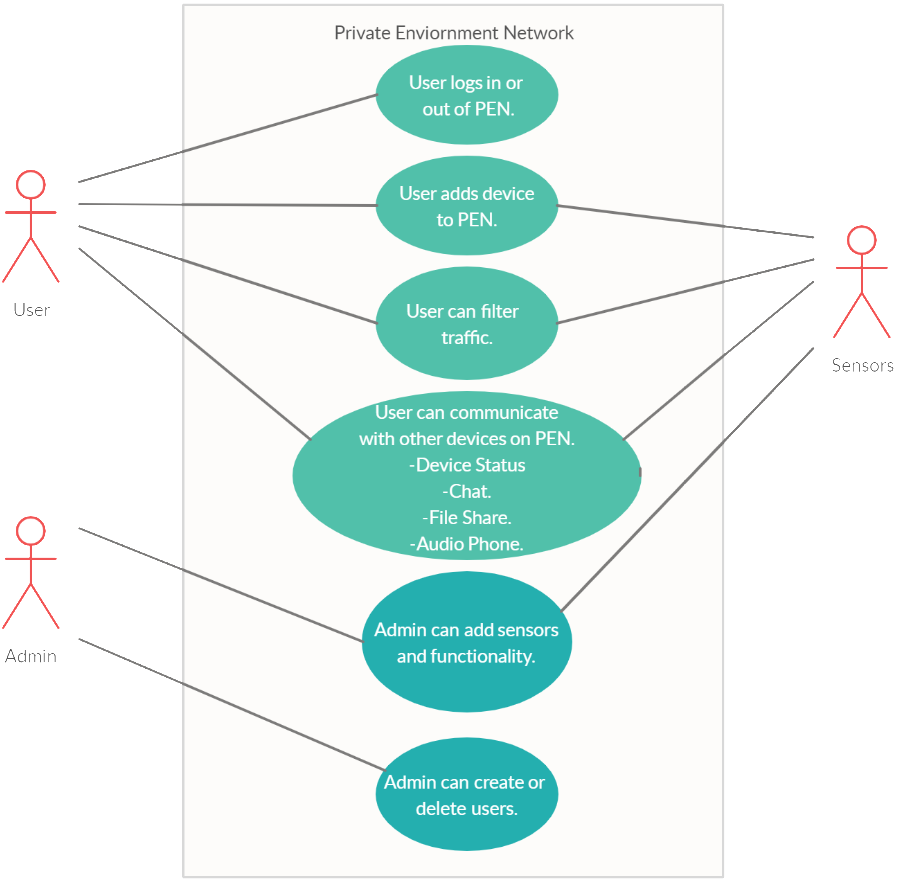
Below are two use case examples of adding a device to the PEN and the phone service.

|  |
| --- |
| **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? |

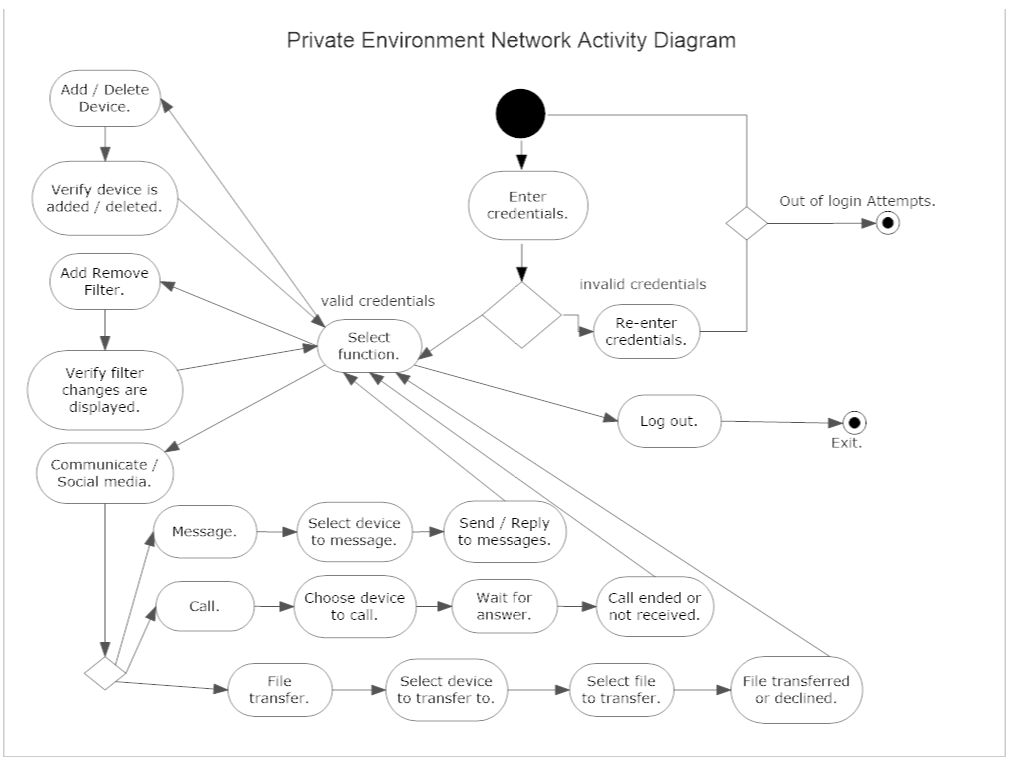
## Use case Diagram.

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 to user is logged out.



## List of Items.

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

|  |  |
| --- | --- |
| **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. |

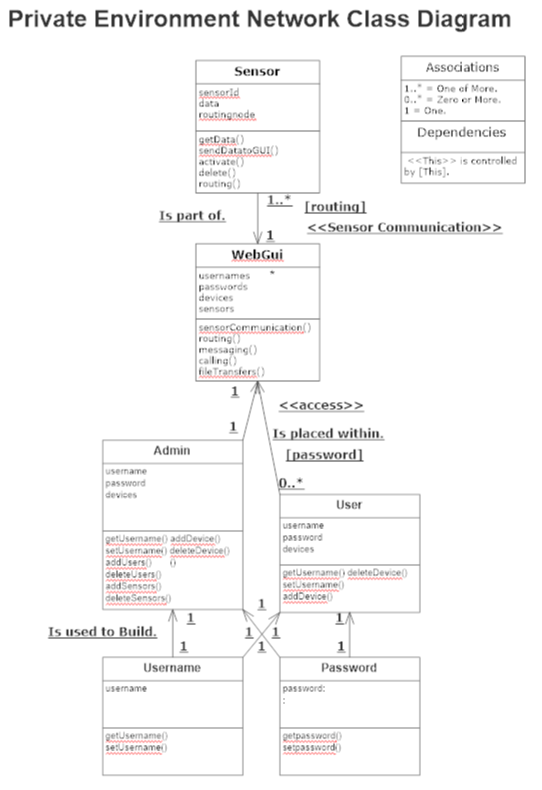
## System List.

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() |

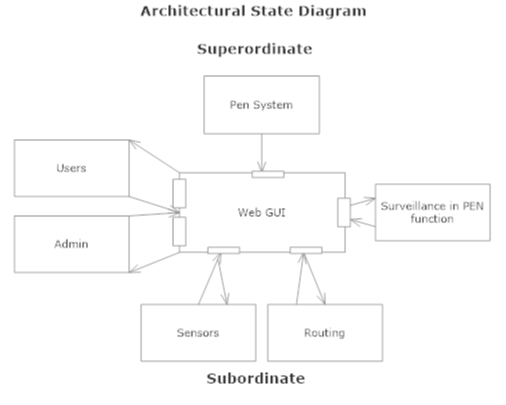
## Class Diagram with Dependencies and Multiplicities.

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.

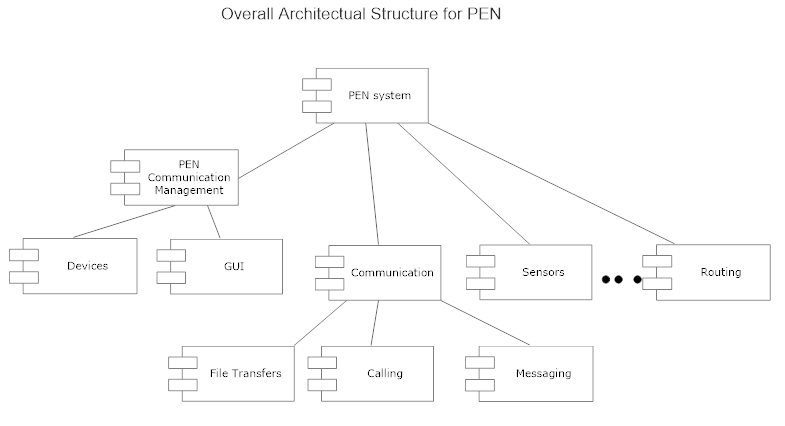


# Design.

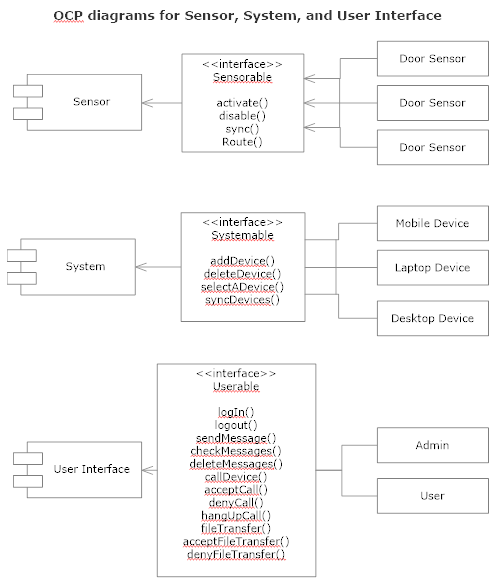
## Architectural State Diagram.



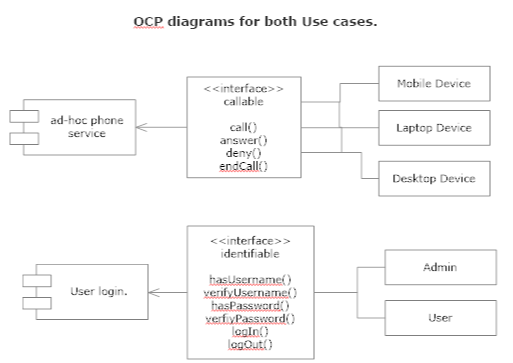
## Overall Architectural State Diagram.



## Class OCP Diagrams.

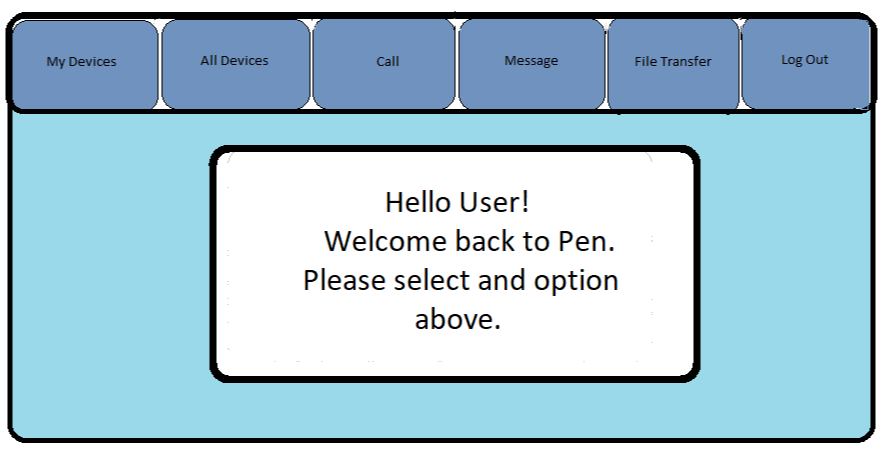


## Use Case OCP Diagrams.

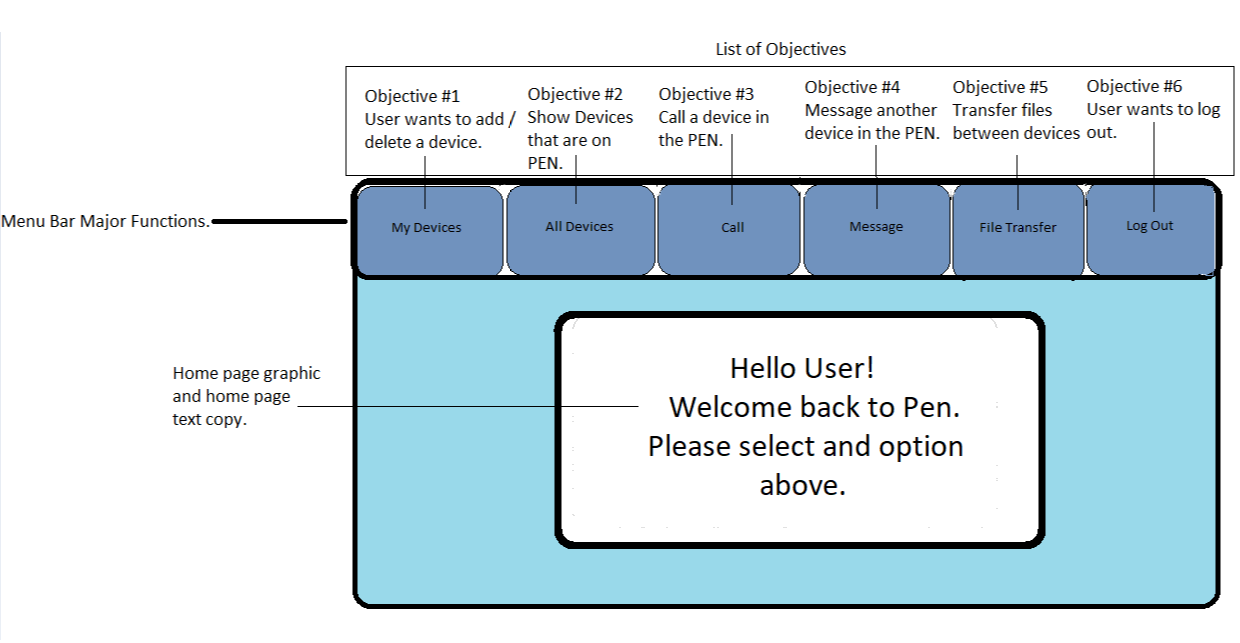


# User Interface.

## Layout Screen.

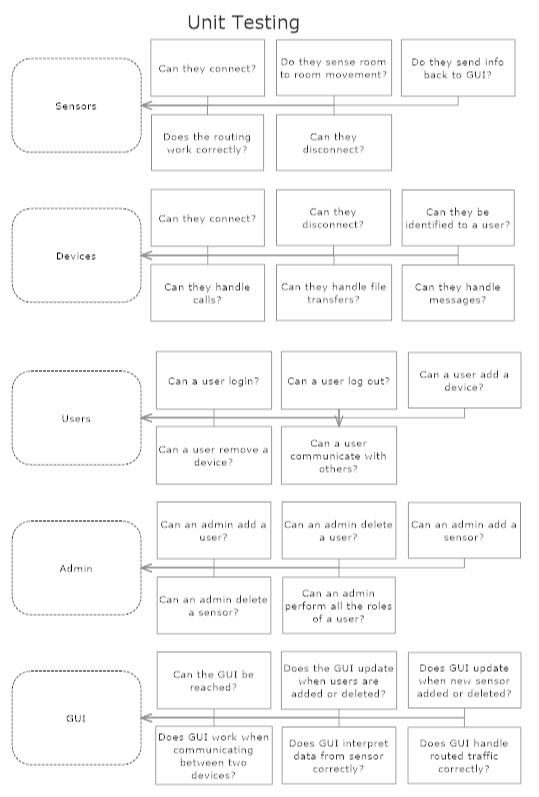


## Layout Screen Sequencing.



# Testing.

## Unit Tests.



## Testing Environments.

