Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4316: Senior Design II
Summer 2016

TrafficNetPeons
Product Name: Hawkeye

Peyton Casper
Rabin Dhaubanjar
Kyle Edelmann
Jose Hernandez
Ruchitha Thalakola

Revision History

Revision	Date	Author(s)	Description
0.1	1.01.2016	GH	document creation
0.2	1.05.2016	AT, GH	complete draft
0.3	1.12.2016	AT, GH	release candidate 1
1.0	1.20.2016	AT, GH, CB	official release
1.1	1.31.2016	AL	added design review requests

CONTENTS

List of Figures

List of Tables

1 Introduction

The Detailed Design Specification (DDS) document will describe in detail the system architecture of the TrafficNet camera. The DDS will discuss the major components of the camera system and how they correspond to each other. The camera will have the ability to allow the user to view its stream and control the zoom, tilting, and panning aspects on a web interface. The camera will stream at least 10 FPS, include day/night control, maintain 10x zoom, be weather resistant, and will be powered from a solar panel and battery. The structure of the software system includes the web interface layer, wireless communication layer, base station layer, and camera layer. Each of these architectural layers include subsystems that have their own individual interfaces and interact with other subsystems.

2 System Overview

In our system there are four main layers, which are classified as web interface layer, wireless connection layer, base station layer, camera layer. The user can communicate with the remote surveillance camera with the help of the user interface. The Ethernet and black box will bridge between the user and the remote camera. User interface will include all kinds of resources that user is capable of using. In the system streaming video is made available and some of the controls that user can use through his computer. In the camera layer we have our camera module, which can be rotated and tilted 180 degrees, which makes it 360- degree total movement possible. In the network layer we will be using Ethernet connection. The source of the power for the camera layer and black box is base station. The main power source for the base station is solar battery.

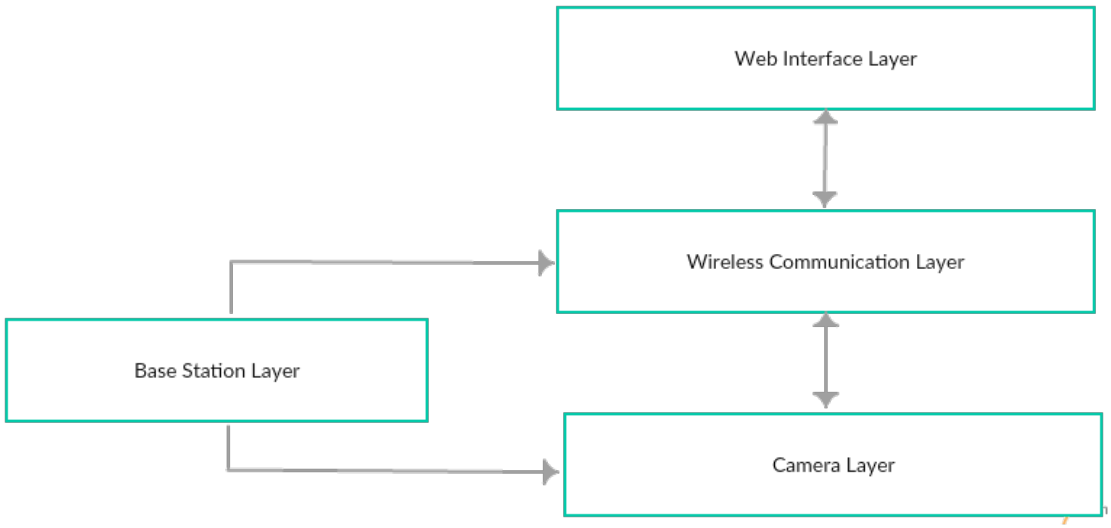

Figure 1: System architecture

3 Base Station Layer Subsystems

The following sections are dedicated to the Base Station Layer of this project. This layer is dedicated to providing power to the Camera and Wireless Communication Layers autonomously through the Solar Battery Subsystem.

3.1 Base Station Layer

The base station is comprised of the Camera Housing, shielding the Camera Layer hardware from the elements, and the Solar Power Supply it is connected to through Micro USB 2.0.

3.2 Base Station Layer Hardware

One Voltage Regulator supplied by TrafficNet, LLC connected to a Solar Cell and battery will serve as the power supply for the Camera Layer by providing 5v DC to the Raspberry Pi Zero.

3.3 Solar Battery Subsystem

The Solar Battery Subsystem actually consists of the Solar Cell, Solar Battery, and Voltage Regulator supplied by TrafficNet, LLC. Together this supplied system controls the energy consumption of the Raspberry Pi Zero and limits its supplied voltage to 5v DC.

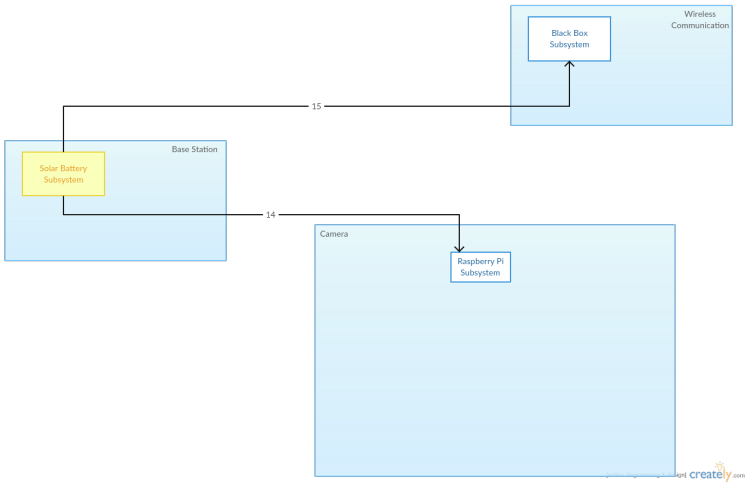

Figure 2: Camera Module Subsystem Description

3.3.1 Solar Battery Subsystem Hardware

The Solar Cell is provided by TrafficNet, LLC. The Voltage Regulator is a standard incorporation of the Solar Cell which keeps the Solar Battery at a stable 5v DC which the Raspberry Pi Zero can natively utilize.

4 Camera Layer Subsystems

The following sections are dedicated to the Camera Layer of this project. This layer includes Raspberry Pi Zero, database, stepper motor drivers, stepper motor, and camera module subsystems. The Camera Layer manages the actions set forth by the user in the Web Interface layer. The operating system used is Raspbian, while the programming language used is python. Besides the stepper motors components, a custom PCB board is used.

4.1 Camera Layer

The Raspberry Pi Zero will execute commands based on the user in the Web Interface layer. Once the user sets forth a command, the RPi Zero will execute and control the movements of the camera using the stepper motor driver to then control the stepper motors.

4.2 Camera Layer Hardware

Since the RPi zero has its own unique operating system, the operating system that will be used is Raspbian.

4.3 Camera Layer Software Dependencies

Besides standard libraries that were installed, lwiringpi library had to be installed to allow functionality of the C++ code we were using at first to test the camera layer. Mjpeg will be used to as the video compressor since it would be the easier router for the amount of time to complete the project in the required time.The functionality is also dependent on the custom PCB stepper motor driver to control the stepper motors.

4.4 Raspberry Pi Zero Subsystem

The Raspberry Pi Zero is controlled by user input in the web interface layer, which is then responsible for controlling all other subsystems in the camera layer.

4.4.1 Raspberry Pi Subsystem Hardware

The Rpi Zero has a 1Ghz single-core CPU, 512MB RAM, micro USB powered, and a 40 pin header HAT.

4.4.2 Raspberry Pi Subsystem Operating System

The required operating system is Raspbian.

4.4.3 Raspberry Pi Subsystem Programming Languages

The programming language used to control the Raspberry Pi Zero is Python and C++.

4.4.4 Raspberry Pi Subsystem Data Structures

For testing purposes and programming, the Raspberry Pi Zero is connected to a monitor via and HDMI and VGA connector.

4.5 Database Subsystem

The Rpi Zero has a wireless network card for internet connectivity, so all the information sent and received by the Raspberry Pi Zero is stored in the database, which is used to store logs of important information.

4.6 Stepper Motor Driver Subsystem

The stepper motor driver is essentially software installed on the Raspberry Pi Zero to control the stepper motors and allow the stepper motors to work.

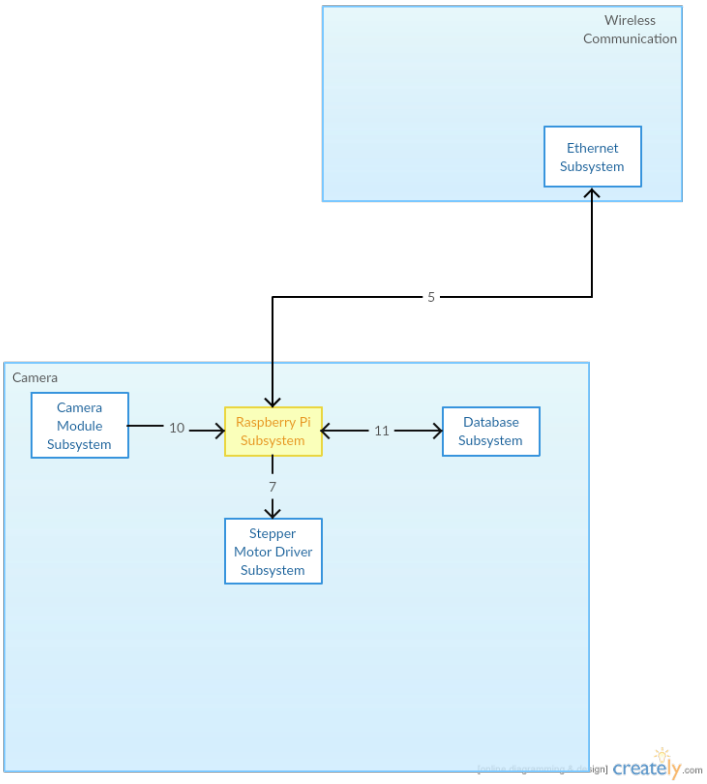

Figure 3: Raspberry Pi Subsystem Description

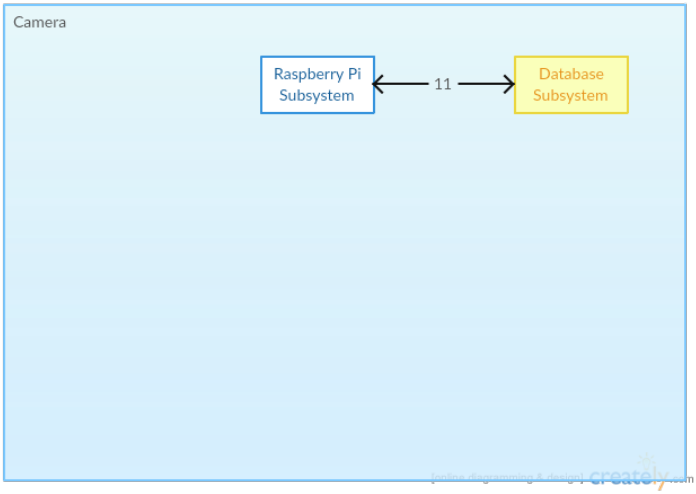

Figure 4: Database Subsystem Description

4.6.1 Stepper Motor Driver Subsystem Software Dependencies

The stepper motor driver is dependent on the Raspberry Pi Zero programmed by the developer.

4.6.2 Stepper Motor Driver Subsystem Programming Languages

The stepper motor driver uses Python programming language to control the stepper motors.

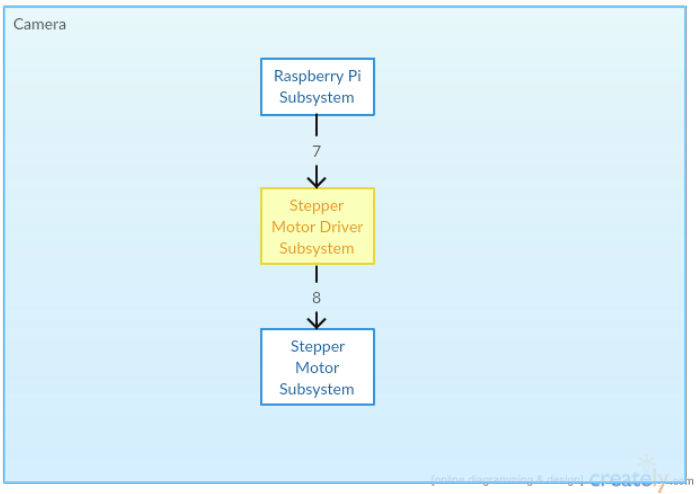

Figure 5: Stepper Motor Driver Subsystem Description

4.7 Stepper Motor Subsystem

The stepper motor driver is essentially software installed on the RPi Zero to control the stepper motors. The stepper motors are connected to the GPIO in the RPi Zero through wires. The stepper motors are controlled through the user web interface. The stepper motors can pan and tilt the camera module.

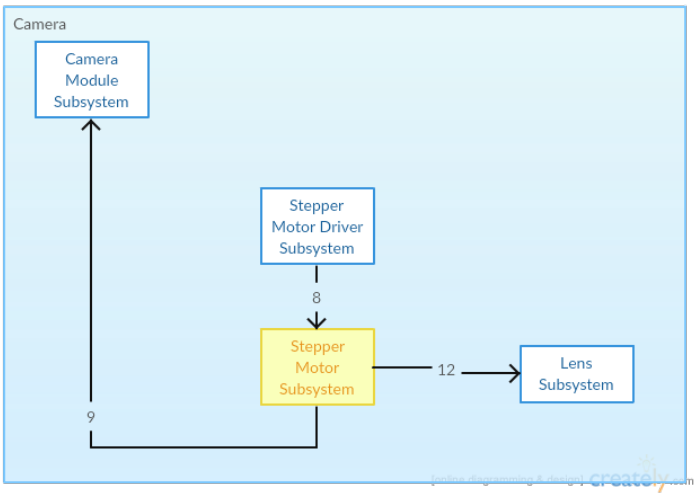

Figure 6: Stepper Motor Subsystem Description

4.7.1 Stepper Motor Subsystem Hardware

The team created a custom PCB board HAT which allows more room to house the stepper motors inside the camera dome. The HAT mounts on top of the RPi Zero and then female to female wires running from the HAT to the stepper motors allows the connection between the two.

4.7.2 Stepper Motor Subsystem Software Dependencies

The stepper motors are dependent on the stepper motor driver and the way the developer programmed the functionality of the stepper motors.

4.7.3 Stepper Motor Subsystem Data Structures

A stepper motor is a brushless, synchronous electric motor that converts digital pulses into mechanical shaft rotations. Each rotation of a stepper motor is divided into a set number of steps.

4.8 Camera Module Subsystem

The camera module is controlled by the stepper motor, which moves the camera (pan and tilt) certain degrees based on user interactions.The camera module is responsible for capturing images of the surrounding landscape. The camera module will provide live feedback of its surroundings when the user request the information from the web interface.

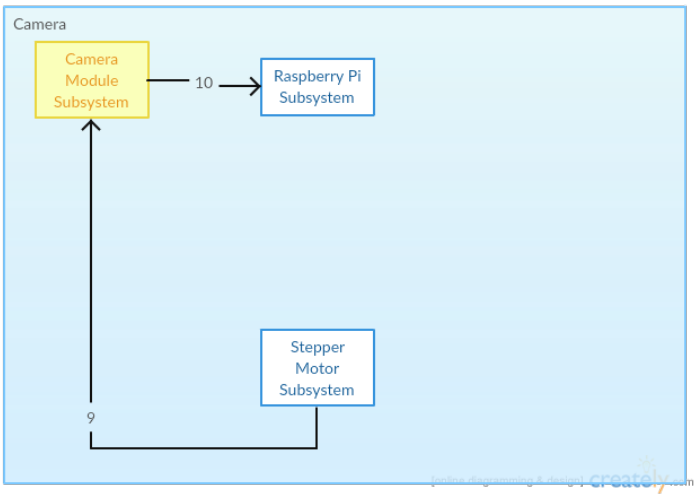

Figure 7: Camera Module Subsystem Description

4.8.1 Camera Module Subsystem Hardware

The camera module is a special connection that can only be made to the RPi Zero.

4.8.2 Camera Module Subsystem Software Dependencies

Since the camera module can only be used by the RPi zero, the camera module depends on the pre-installed software of the RPi zero.

4.8.3 Camera Module Subsystem Data Structures

The camera module has a five megapixel fixed-focus camera that supports 1080p, 720p and VGA90 video modes, as well as stills capture. It attaches via a 15cm ribbon cable to the CSI port on the RPi Zero.

5 Wireless Communications Layer Subsystems

The following sections are dedicated to the Wireless Communications Layer of this project. This layer includes the Ethernet Subsystem and the Black Box Subsystem. The Wireless Communications Layer manages the communications demands between the Web Interface Layer and the Camera Layer via UDP transport protocol over a wireless network provided the consumer by TrafficNet LLC.

5.1 Wireless Communications Layer

User issued commands will be sent from the Web Interface Layer to the Black Box Subsystem via UDP/IP Transfer Protocol, and subsequently forwarded by the Ethernet Subsystem to the Raspberry Pi Zero for execution.

5.2 Wireless Communications Layer Hardware

Specifically, the Wireless Communications Layer is comprised of a TBD Wireless Network Adapter of TrafficNet, LLC's choosing and the Ethernet Adapter connected to the Raspberry Pi Zero. The two components work in tandem to provide a means through which the Raspberry Pi Zero can establish communication with the Web Interface Layer.

5.3 Layer Software Dependencies

It is necessary for the appropriate software to be present on the Raspberry Pi Zero to provide the end connection for the video streaming server to operate through. This software runs the UDP/IP Transfer Protocol which allows the video to be streamed wirelessly to the Web Interface Layer. It is the TrafficNet.py program described in the Camera Layer.

5.4 Black Box Subsystem

The Black Box is a placeholder name for the indeterminate Wireless Network Adapter to be incorporated by the vendor TrafficNet, LLC at a later date. However, it is crucial to the operation of the Wireless Communications Layer. Regardless of the specifics of the chosen WNA, its function will remain the same. Without it there can be no connection established between the ports of the user's machine and the Raspberry Pi Zero.

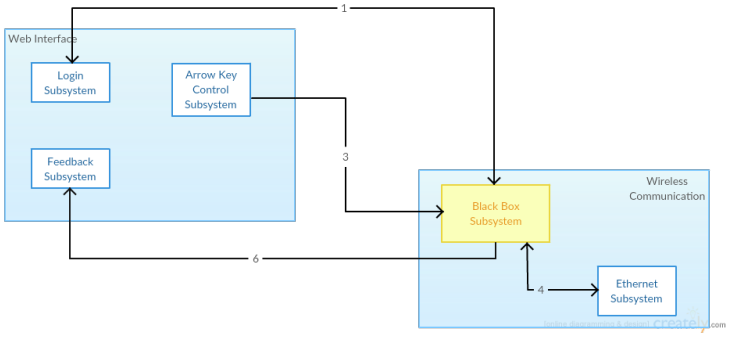

Figure 8: Black Box Subsystem description diagram

5.4.1 Black Box Subsystem Hardware

The specific WNA at this time is unknown.

5.5 Ethernet Subsystem

The Raspberry Pi Zero has an Ethernet Adapter connected to it. This is Ethernet Subsystem of the Wireless Communications Layer. It connects the Raspberry Pi Zero to the WNA of the Black Box Subsystem.

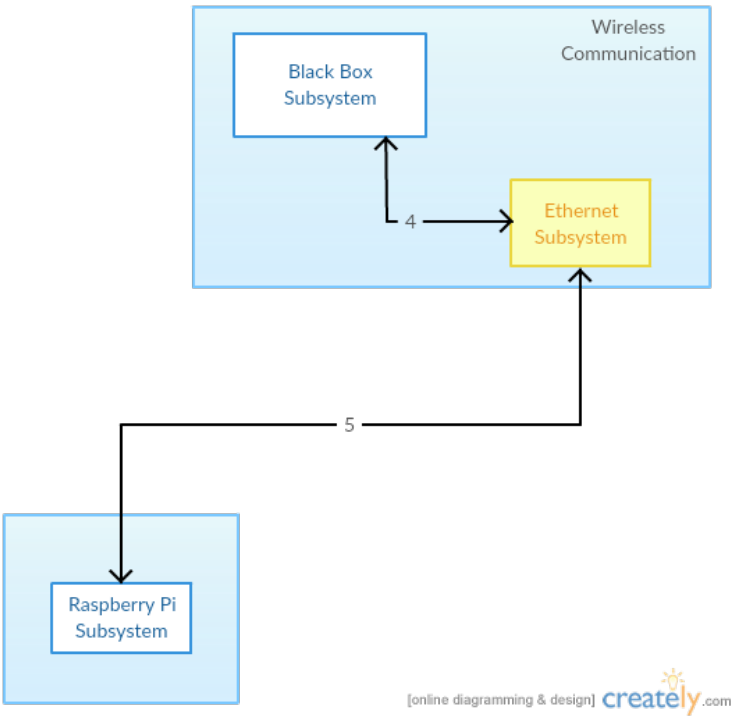

Figure 9: Database Subsystem Description

5.6 Ethernet Subsystem Hardware

The Ethernet Adapater is a Micro-USB 2.0 to Ethernet Port OTG Cable capable of communication speeds up to 100Mbps.

5.7 Ethernet Subsystem Software Dependencies

The Ethernet Adapter comes with pre-installed drivers for 802.3 Network Standards.

6 Web Interface Layer Subsystems

This section details the web interface layer and the three subsystems that it encompasses. The web interface layer includes the login, feedback and arrow key interface subsystems.

6.1 Layer Hardware

The Web Interface layer runs as a web server ontop of a Raspberry Pi Zero. The Raspberry Pi Zero runs off Raspbian OS which is a flavor of Debian Jessie.

6.2 Layer Operating System

This layer relies on Raspbian OS which is a flavor of Debian Jessie.

6.3 Layer Software Dependencies

The Web Interface relies on jQuery, HTML, CSS, Python, Python Flask, and the GPIO Zero Python library.

6.4 Login Subsystem

The login sybststen will receive internet access from the black box system and be the first interface that the user will arrive at. The user will be required to access the system using a password or username stored in a database on the Raspberry Pi.

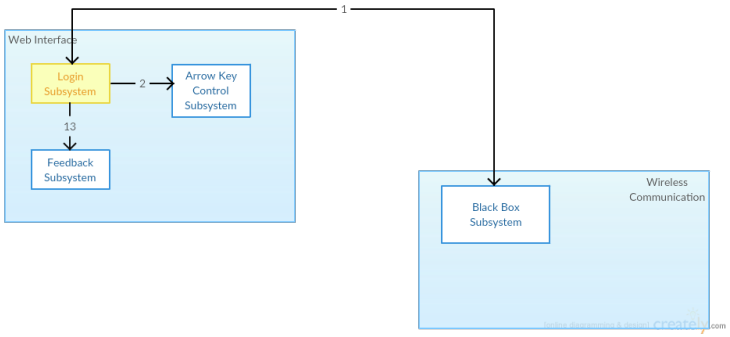

Figure 10: Diagram of the Login Subsystem

6.4.1 Login Subsystem Software Dependencies

The Login Subsystem relies on jQuery, HTML, CSS, Python, and Python Flask.

6.4.2 Login Subsystem Programming Languages

Javascript, HTML, CSS, and Python were used to create this subsystem.

6.4.3 Login Subsystem Data Structures

The Login Subsystem formats all requests to the server in a JSON format to send to the server.

6.5 Arrow Key Control Subsystem

The arrow key control subsystem is reached after a successful validation from the login subsystem and provides the ability to pan and tilt the camera.

6.5.1 Arrow Key Control Subsystem Software Dependencies

The Arrow Key Control Subsystem relies on jQuery, HTML, CSS, Python, Python Flask, and the GPIO Zero Python Library.

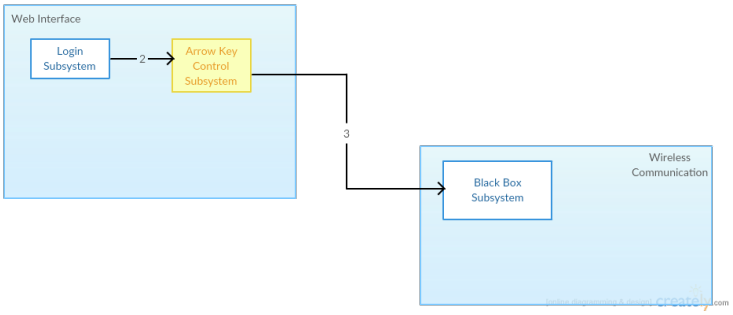

Figure 11: Diagram of the arrow key control subsystem

6.5.2 Arrow Key Control Subsystem Programming Languages

Javascript, HTML, CSS, and Python were used to create this subsystem.

6.5.3 Arrow Key Control Subsystem Data Structures

The Login Subsystem formats all requests to the server in a JSON format to send to the server.

6.6 Feedback Subsystem Subsystem

The feedback subsystem is responsible for relaying the camera feed back to the user once they have been successfully validated by the login subsystem.

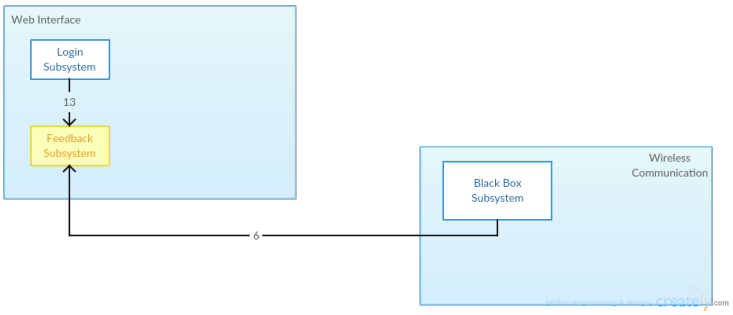

Figure 12: Diagram of the feedback subsystem

6.6.1 Feedback Subsystem Software Dependencies

The Feedback Subsytem relies on UV4l to provide the camera stream back to the user.

6.6.2 Feedback Subsystem Data Structures

The Feeback Subsystem formats the camera stream as an mjpeg stream to send back to the user.

7 Appendix A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.