***Group** 10* *Death Star Plans Radio Transmitter*

Major:	**Team members:**
CEG	*Zachary Leyes*
CS	*Luke Kremer*
ITC	*Jeremiah Hackman*
EE	*Julie Peterson-Ramos*
CS	*Khiem Do*

Design Objectives

Objective 1: The proposed solution will identify which of the 100 images contain red circles.

The 10 images with red circles are hidden among 90 decoy images.

Objective 2: The system will transmit 10 images in 600 seconds with encryption.

The transmission will not use Wi-Fi, Bluetooth, cellular signals, or a physical connection. The transmission will include error detection and correction. The images will be encrypted before transmission and decrypted after transmission.

Objective 3: The images will be hosted on a web server that can be accessed through a mobile application.

A mobile application can refer to a web app, iOS app, or Android app. The mobile application will feature a scrollable table displaying the images with red circles. Any user with a smartphone will be able to access this application and view the images.

Objective 4: The web server will host a video. Any user who accesses the web server will be able to view this video.

The first video does not need authentication, and any user can access it.

Objective 5: The web server will host a second video only accessible to authorized users.

Only authorized users can view this video. A secure form of authentication not relying solely on a password will be used to identify users.

Design Assumptions

Assumption 1: The provided room will have a standard 120 V and 60 Hz power supply.

Assumption 2: The Rebel Server will be placed 5 to 10 meters from the lab window.

Assumption 3: Obi-wan will have access to an email account as well as an account on the Rebel website.

Assumption 4: Princess Leia's messages will be in a video format.

Assumption 5: Ambient noise and interference in the radio frequency band chosen will remain low enough for successful communication.

Assumption 6: The Rebel Server can communicate via Wi-Fi with the WSU_EZ network.

Design Requirements

Objective 1: The proposed solution will identify which of the 100 images contain red circles.

Objective 2: The system will transmit 10 images in 600 seconds with encryption.

Objective 3: The images will be hosted on a web server that can be accessed through a mobile application.

Objective 4: The web server will host a video. Any user who accesses the web server will be able to view this video.

Objective 5: The web server will host a second video only accessible to authorized users.

Req No.	Obj No.	Requirement
10	1	The Raspberry Pi shall load images from the USB drive.
20	1	The Raspberry Pi shall identify images with red circles.
30	2	The Raspberry Pi shall have the ability to encrypt images.
40	2	The Rebel Server shall have the ability to decrypt images.
50	2	The Raspberry Pi shall have the ability to transmit and receive data with radio frequency waves.
60	2	The Rebel Server shall have the ability to transmit and receive data with radio frequency waves.
70	2	After receiving a transmission, the Rebel Server shall perform error checking.
80	2	In the case of an error, the Rebel Server shall send a transmission to the Raspberry Pi requesting to resend the data.
90	2	If the Raspberry Pi receives a request to resend a transmission, then it will send the image again.
100	2	The system shall complete all transmissions in 600 seconds or less.
110	4	Any user shall have access to the first video.
120	5	Authorized users shall have exclusive access to the second video.
130	5	The mobile app shall use a secure method of authentication to identify authorized users.
140	3	The mobile app shall display images in a scrollable table.

Req No.	Obj No.	Requirement
150	3	The mobile app shall display up to 10 images.
160	3	The scrollable table shall display the contents of each red circle exactly once.

Definitions:

Encrypt – Protected using a cryptographic method so that intercepted transmissions cannot be read.

Secure Authentication - A form of verification not relying solely on a password.

Scrollable table – A user interface that allows users to view multiple entries by scrolling vertically.

Exclusive Access - Access granted to only a select amount of users and individuals.

Design Constraints

Const No.	*Constraint*
10	The total cost of the project shall not exceed $300 without Faculty Advisor approval.
20	The solution shall use only the provided equipment plus additional components purchased within the budget; a personal mobile application host device is required.
30	The Raspberry Pi and all equipment connected to it must fit within the provided box.
40	The Raspberry Pi and Rebel server shall not communicate with each other using Wi-Fi, Bluetooth, cellular communication, or a wired connection.
50	The Rebel Server and all its components shall be placed at least 5 meters from the Raspberry Pi.
60	The range of frequencies used shall be compliant with FCC regulations.

Definitions:

Mobile Application Host Device – A user-owned smartphone or tablet required to run the companion application.

FCC Regulations – Rules established by the U.S. Federal Communications Commission that governs the legal use of frequency ranges.

Design Standards

Stand. No.	*Standard*
10	The solution shall comply with AES FIPS 197-upd1 Standard for Encryption. https://nvlpubs.nist.gov/nistpubs/FIPS/N-IST.FIPS.197-upd1.pdf
20	The solution shall comply with C95.3-2021: Standard for Radio Frequencies https://ieeexplore.ieee.org/document/9444273
30	The mobile application shall comply with the IEEE/ISO/IEC 23026-2023 standard for website design. https://standards.ieee.org/ieee/23026/10425/=]
40	All Python code shall comply with PEP 8 for coding style. https://peps.python.org/pep-0008/

Definitions:

AES – Advanced Encryption Standard

IEEE – Institute of Electrical and Electronics Engineers

Stand. No	Standard
10	The solution shall comply with AES FIPS 197-upd1 for encryption.
20	The solution shall comply with C95.3-2021 for radio frequencies.
30	The mobile application shall comply with IEEE 23026-2023 for website design.
40	All Python code shall comply with PEP 8 for coding style.

Design Functionality

Top-Level Functional Flow

1. Image Identification & Selection
2. Encryption
3. Wireless Transmission
4. Integrity Checking & Retransmission
5. Server Verification & Storage
6. Decryption at Endpoint
7. End-User Display of Images
8. Public Video Delivery
9. Authenticated Video Delivery

Narrated Functional Breakdown

1. Image Identification & Selection
    - The Raspberry Pi scans all stored images and selects those that have the red circles for upload.
2. Encryption
    - Selected images are encrypted using a cryptographic method.
3. Wireless Transmission
    - Encrypted images are transmitted over a radio frequency channel that does not rely on Wi-Fi, Bluetooth, or cellular networks.
4. Integrity Checking & Retransmission
    - The receiving end checks transmission for errors or loss.
    - Any corrupted or incomplete data is automatically retransmitted.
5. Server Verification & Storage
    - The server validates the completeness and correctness of received data against the integrity metadata.
    - Verified images are stored in preparation for display.

6. Decryption at Endpoint
    - The encrypted images are decrypted at the authorized endpoint (server or application, depending on system design).

7. End-User Display of Images
    - Images are made available to end-users through a mobile application.
    - The application provides scrollable images for browsing.
8. Public Video Delivery
    - One video is published and accessible to all users without authentication.

9. Authenticated Video Delivery
    - A second video is available only to authenticated users.
    - Multi-factor authentication shall be applied to ensure exclusive access.

Bulleted List of Major Functions

- Scan and select images on the Raspberry Pi.
- Encrypt images.
- Transmit images over a radio frequency channel.
- Perform integrity checks and retransmit corrupted data.
- Verify and store data on the server.
- Decrypt images at the authorized endpoint.
- Display images in a scrollable, user-friendly mobile application.
- Publish one video for unrestricted access.
- Publish one video for authenticated access.

Design Impact

1. Cultural

Stealing confidential data violates cultural norms of privacy and security. If this technology became widespread, it could be used to steal private data.

2. Economic

The system uses relatively inexpensive components. However, if the technology became widespread, it could be used to steal sensitive data, which could impose financial costs on victims.

3. Environmental

The system uses small amounts of electricity and processing power and emits low-power radio waves. The overall environmental impact will be negligible.

4. Global

The technology in this solution could be used to steal other sensitive data from the Empire. This could contribute to the galaxy's collective fight against the Empire.

5. Public health

The low frequencies emitted from the transmission device will not affect human health. The system will comply with IEEE standards for radio transmission to ensure its safety.

6. Public safety

The transmission device will not pose a threat to public safety as it has no moving parts, no sharp points, and no deadly emissions.

7. Public welfare

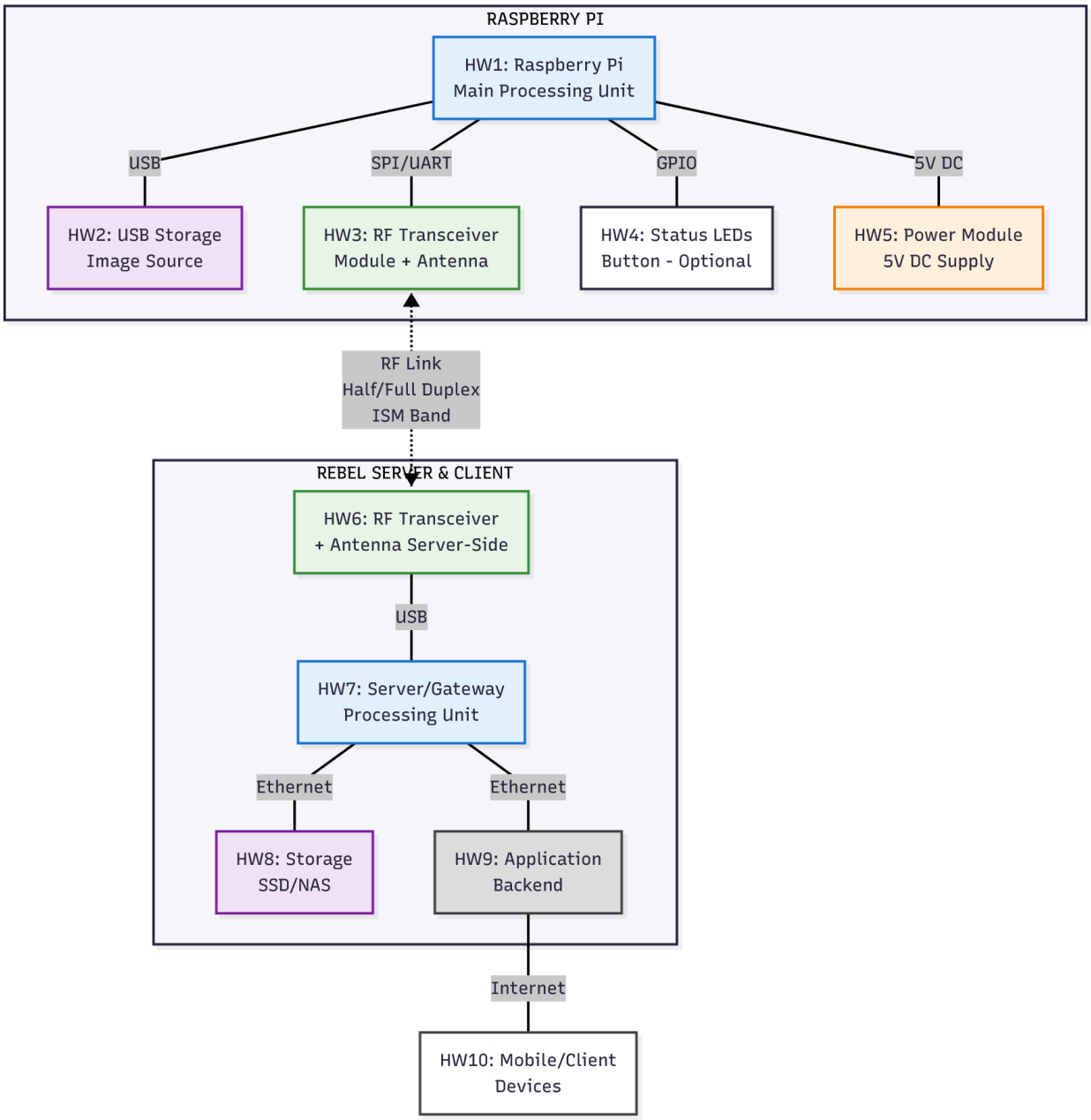
If the technology used in this system became widespread, it could be used to benefit the public by exposing information from other unethical government programs.

8. Social

The technology used to steal the Death Star plans could potentially be used by malicious actors to steal other sensitive information. Individuals may be at risk of data theft.

System Architecture

Hardware Architecture

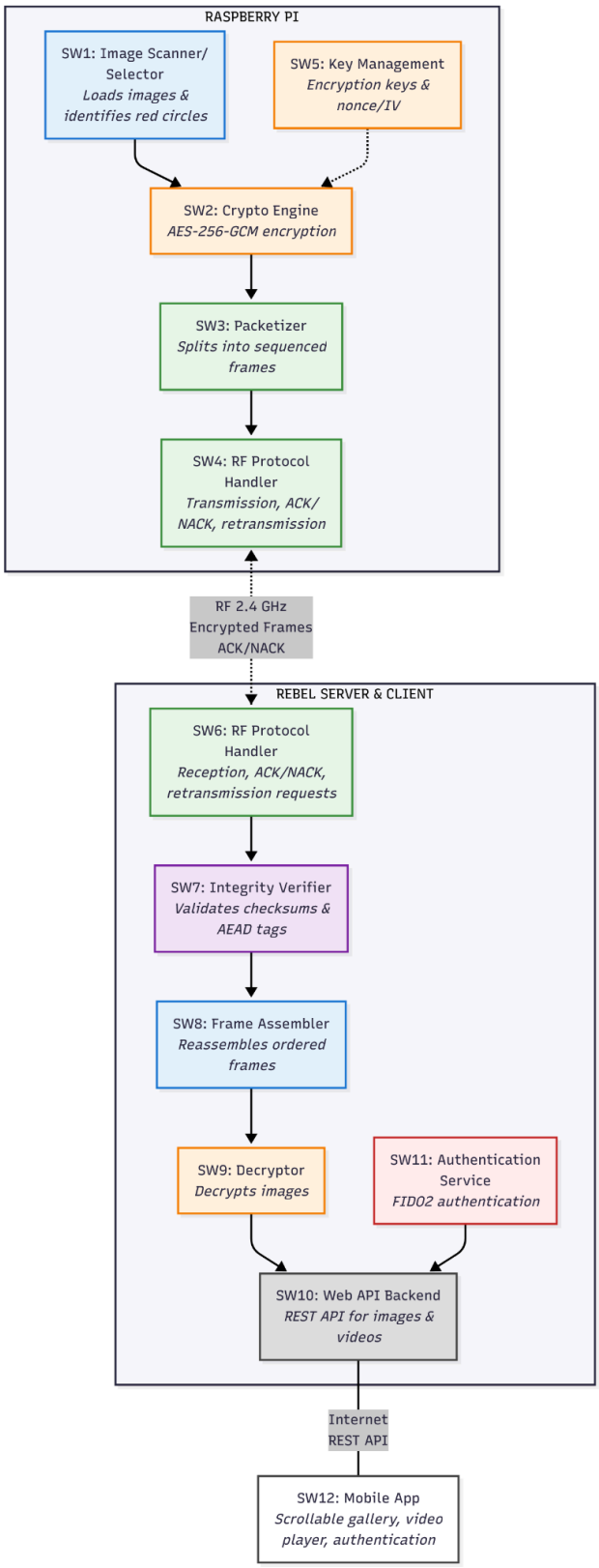


Hardware functional block definitions

- HW1 Raspberry Pi – Single-board computer executing image selection, encryption, packetization, RF link protocol, and retransmission logic.
- HW2 USB Storage – Local mass storage (e.g., USB flash drive) holding the input image set to be scanned/selected.

- HW3 RF Transceiver Module + Antenna (Pi-side) – Sub-GHz/ISM-band transceiver (e.g., FSK/LoRa/OOK capable) connected via SPI or UART; provides non-Wi-Fi/BT/cellular wireless transport.
- HW4 Status I/O (optional) – LED(s)/button for local operator feedback (e.g., TX/RX status, error, retry).
- HW5 Power Module – 5V DC supply for the Pi and RF module (battery or regulated adapter); includes basic EMI filtering.
- HW6 RF Transceiver + Antenna (Server-side) – Matching transceiver for two-way ACK/NACK and control; connected to server via USB/UART.
- HW7 Server/Gateway – Receives RF frames, verifies integrity, issues ACK/NACK, forwards data to backend services; may also perform decryption if server-side endpoint is chosen.
- HW8 Storage (SSD/NAS) – Persistent storage for received ciphertext/plaintext (per design), logs, and metadata.
- HW9 Application Backend – Hosts REST/GraphQL APIs, auth, and content services that the client app consumes.
- HW10 Mobile/Client Devices – User devices that browse/download images and (separately) play videos per access policy.

Software Architecture



Software functional block definitions

Raspberry Pi Software:

- SW1 Image Scanner/Selector – Loads images from USB and identifies red circles using OpenCV
- SW2 Crypto Engine – Encrypts images with AES-256-GCM and generates authentication tags
- SW3 Packetizer – Splits encrypted images into sequenced frames with checksums
- SW4 RF Protocol Handler – Manages transmission, handles ACK/NACK, implements retransmission on errors
- SW5 Key Management – Loads encryption keys and manages nonce/IV generation

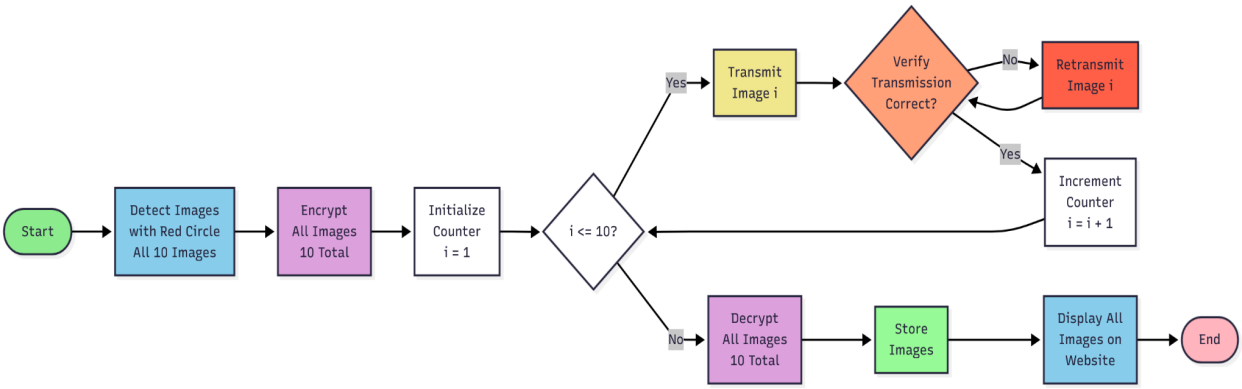
Rebel Server Software:

- SW6 RF Protocol Handler – Receives frames, sends ACK/NACK, requests retransmission on errors
- SW7 Integrity Verifier – Validates checksums and AEAD tags, detects missing/corrupted frames
- SW8 Frame Assembler – Reassembles ordered frames into complete encrypted images
- SW9 Decryptor – Decrypts images and stores plaintext
- SW10 Web API Backend – Serves images and videos via REST API
- SW11 Authentication Service – Implements FIDO2 authentication for protected content

Client Application (Mobile):

- SW12 Mobile App – Displays images in scrollable gallery, plays videos, handles authentication

Encryption and Decryption Breakdown



1. Detect images with red circle - all 10
2. Encrypt the images
3. Transmit Image - one at a time
4. Verify the the transmission is correct for each image
    a. If transmission is incorrect, retransmit image until correct
5. Decrypt images - all 10
6. Store images
7. Display all images on website

We need a flowchart with the details for encryption then another flow chart for decryption including the integrity checks

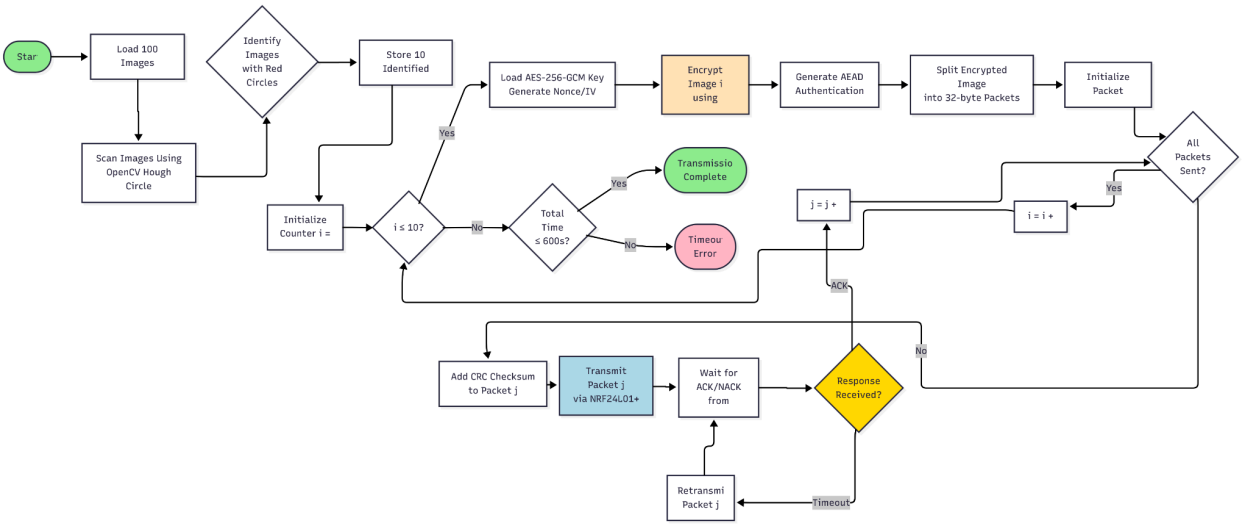
So the encryption should include the start, detecting the images, how encryption will work (AES key) this will be done through OpenSSL, encrypt all ten images, transmit the images in packets one by one (easier for integrity checking and retransmission)

Decryption should include the receiving of the images, verifying the transmission of each packet, retransmission if incorrect transmission, the decryption (will take place once the packets are put together for the whole image), stored on the local computer, displayed on mobile app (in scrollable table), end

Code:

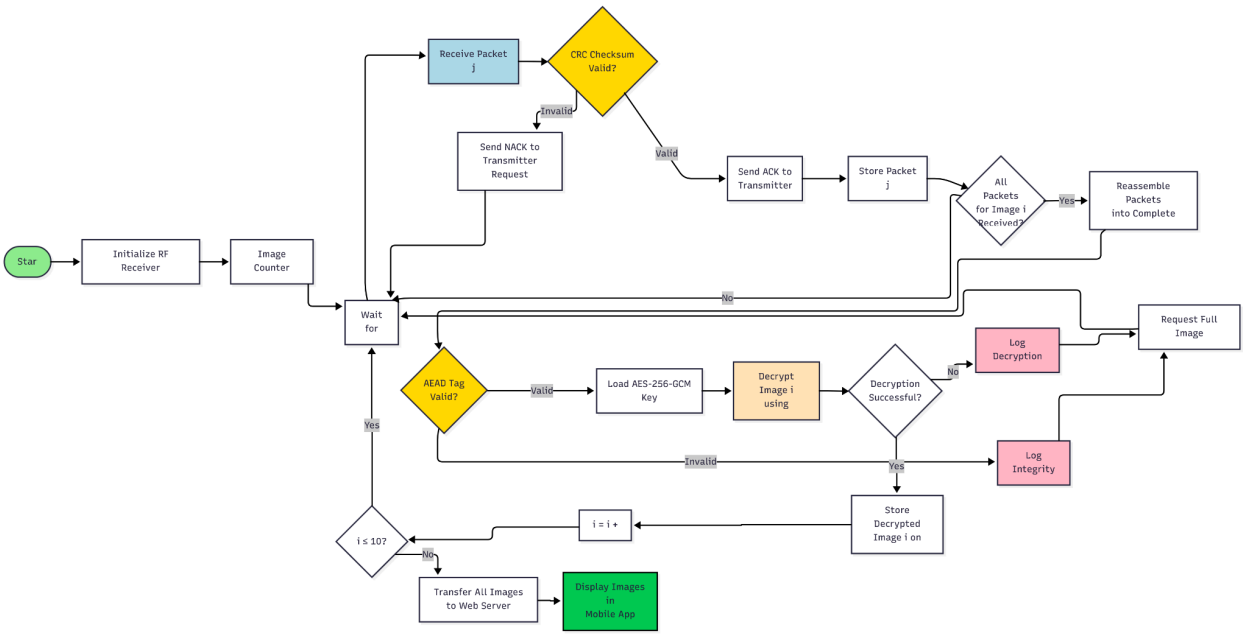
Encryption:
openssl enc -aes-256-cbc -pbkdf2 -in image.png -out encrypted_image.png -pass pass:password.txt
Decryption:
openssl enc -d -aes-256-cbc -pbkdf2 -in encrypted_image.png -out decrypted_image.png -pass pass:password.txt

Encryption:



Starts off with the loading of the images, detecting the images, load the key onto the images, encrypt all ten images with AES key, transmit the images in packets one by one (easier for integrity checking and retransmission), each packet gets a CRC checksum number, transmit each packet to receiver side, verify all packets have been sent and within the time limit

Decryption:



Starts with receiving the packets, verifying the transmission of each packet was successful, reassembling the images, then verify the whole image is correct, decrypt the image using the AES key, repeat for all 10 images, transfer images to web server, display images

4.1

System Design Trades

Criteria	Raspberry Pi 4 (Broadcom BCM2711)	Raspberry Pi 5 (Broadcom BCM2712)	STM32F446RE (ARM Cortex-M4)	BeagleBone Black (TI AM3358)
Requirement 10: Image Scanning	Compliant (OpenCV supported)	Compliant (Enhanced OpenCV performance)	Non-compliant (OpenCV not available)	Compliant (OpenCV supported)
Requirement 40: Transmission < 600s	Compliant (1.5 GHz, 4 cores)	Compliant (2.4 GHz, 4 cores)	Non-compliant (180 MHz, single-core)	Compliant (1 GHz, dual-core)
Requirement 20: Encryption (AES)	Compliant (OpenSSL supported)	Compliant (Enhanced AES performance)	Non-compliant (Limited cryptographic libraries)	Compliant (OpenSSL supported)
Constraint 20: Cost < $300	Compliant ($64)	Non-compliant ($92)	Compliant ($29)	Compliant ($55)
Constraint 60: Fits in Box	Compliant (85x56 mm)	Compliant (85x56 mm)	Compliant (101x76 mm)	Non-compliant (110x56 mm)
Standard 10: AES FIPS 197	Compliant (OpenSSL FIPS)	Compliant (OpenSSL FIPS)	Non-compliant (No FIPS support)	Compliant (OpenSSL FIPS)

Design Trade Notes

1. Raspberry Pi 5 – This device offers the best processing power of all options, as well as the largest memory and storage capacity. The Raspberry Pi 5 has the highest power consumption at 5–12 W and is the most expensive at $92.
2. Raspberry Pi 4 – This device is the previous generation of the Pi series, offering lower processing power and memory capacity compared to the Pi 5. However, it features the same storage capabilities while consuming less power and costing less.
3. STM32F446RE – This microcontroller differs from the single-board computers (SBCs) like the Raspberry Pi models. It has significantly less RAM, flash memory, and processing power, but it excels in real-time hardware control through direct interfacing with peripherals. Its power consumption and price are the lowest among all the options.
4. BeagleBone Black – The BeagleBone Black is a versatile SBC that balances performance and hardware-level control. It offers a 1 GHz TI AM3358 processor, 512 MB

of RAM, and 4 GB of onboard eMMC flash storage. Its moderate power consumption (2–5 W) and price ($53.77) make it a strong mid-range option.

RF Transceiver

Criteria	NRF24L01+ (Nordic Semiconductor)	CC1101 (TI)	HC-12 (Si4438)
Req 70: 3kHz–300GHz	2.4- 2.5 GHz	387-464 MHz	433 MHz
Const 40: Bit Rate	250 kbps - 2 Mbps max	1.2kbps - 500 kbps	9.6 kbps
Req 22: Bidirectional	Yes	Yes	Yes
Req 50: Error Correction	Built-in protocol	Built-in FEC	None
Const 20: Cost <$300	10 ~ $13.99	3 ~ $22.99	1 ~ $9.49
Const 80: >5m Range	10m - 30m (indoors)	100m	500m
Power Supply	1.9-3.6 V	1.8V - 3.6 V	3.3 V

Design Trade Notes

1. CC 1101 ~ This device has multiple frequency ranges but the one listed is the most used. The bit rate has a max of 500 kbps but the device will most likely not reach this max before becoming slow. Compared to the other devices listed, the price is the highest but it does provide three units. This device is compatible with raspberry pi and has a lot of documentation available.
2. NRF24L01+ ~ This device has another frequency range (433 MHz) but the range listed is more commonly used. Compared to the other options this one is the cheapest and offers the most units. This can be beneficial for testing and allow for various options to be done. There is a lot of documentation available for this device and how this device was used for various projects similar to this one.
3. HiLetgo HC-12 ~ This device has one specific frequency for use which can be limiting. This is also the most expensive option since the solution would need two devices. There is not a lot of documentation available compared to the first two but there is some.

Website	Local Computer	WordPress [2]	Wix [3]

Price	Free	$0-59/month	$16-59/month
Hosting	Self-managed on personal device	Fully managed	Fully managed
Customization	Complete control, unlimited	Good with plugins (limited on free plan)	Good with templates and app market
Performance	Excellent (with proper setup)	Good	Good
Support	No support, plentiful online resources	Email, live chat (paid plans)	24/7 support on all plans

Design Trade Notes

1. Local ComputerAW - Highly customizable, able to use complex workflows, and has infinite scalability.
2. WordPress - Limited PHP stack, meant for small websites that scale only slightly, many default themes and plugins.
3. Wix - Drag and drop, launches quickly, limited access and not made for heavy traffic.

Encryption Library

Criteria	OpenSSL (FIPS)	PyCryptodome	libsodium
Req 20: Encryption	AES-256-GCM	AES-256-CBC	XSalsa20
Std 10: FIPS 197	Certified	Not certified	Not certified
Const 20: Cost	Free	Free	Free
Language Support	Python/C	Python	Python/C

Design Trade Notes

1. OpenSSL: Widely used, can be run from the command line or through programming libraries, FIPS certified
2. PyCryptodome: Python library for cryptography tasks
3. libsodium: Cryptography library available in a variety of programming languages

Image Processing Algorithm

Criteria	OpenCV (Hough Circles)	scikit-image (Color	Pillow (Basic Filtering)	YOLOv5 (ML)

		Thresholding)		
Const 20: Cost <$300	Free	Free	Free	GPU needed
Std 10: Compatibility	Python/C++	Python	Python	PyTorch
Ease of Implementation	Complex tuning	Simple	Very simple	Training needed

Design Trade Notes

1. OpenCV: Widely used for a variety of image processing tasks, high performance, extensive documentation
2. scikit-image: Python library for image processing, simple to use
3. Pillow: Fork of PIL (Python imaging library), simple to use
4. YOLOv5: Machine learning model, may be overkill for identifying red circles

Client Application

Criteria	Flutter (Dart)	React Native (JS)	Native (Swift/Kotlin)
Req 80: Scrollable Table	ListView	FlatList	UITableView/RecyclerView
Req 65: 2FA Auth	Firebase Auth	React Native Auth	Manual impl
Const 20: Cost <$300	Free	Free	2x dev effort
Std 30: IEEE 23026	Material Design	Custom UI	Native UI
Development Time	Single codebase	JS bridge	Double work

Design Trade Notes (Client Application)

1. Flutter (Dart): Cross-platform framework with a single codebase for Android/iOS; fast UI performance; uses Material Design by default; easy Firebase integration.
2. React Native (JS): Allows reuse of web development skills in JavaScript; good community support; slightly slower due to JS bridge; simple setup for Firebase Auth.

3. Native (Swift/Kotlin): Best performance and full access to device features; requires two separate codebases; higher development time and cost; ideal for platform-specific optimization.

4.2
HW/SW Design Trade

Microcontroller Selection

Criteria	Raspberry Pi 4 (Broadcom BCM2711)	Raspberry Pi 5 (Broadcom BCM2712)	STM32F446RE (ARM Cortex-M4)	BeagleBone Black (TI AM3358)
Requirement 10: Image Scanning	Compliant (OpenCV supported)	Compliant (Enhanced OpenCV performance)	Non-compliant (OpenCV not available)	Compliant (OpenCV supported)
Requirement 40: Transmission < 600s	Compliant (1.5 GHz, 4 cores)	Compliant (2.4 GHz, 4 cores)	Non-compliant (180 MHz, single-core)	Compliant (1 GHz, dual-core)
Requirement 20: Encryption (AES)	Compliant (OpenSSL supported)	Compliant (Enhanced AES performance)	Non-compliant (Limited cryptographic libraries)	Compliant (OpenSSL supported)
Constraint 20: Cost	Compliant ($68)	Non-compliant ($92)	Compliant ($29)	Compliant ($55)
Constraint 60: Fits in Box	Compliant (85x56 mm)	Compliant (85x56 mm)	Compliant (101x76 mm)	Non-compliant (110x56 mm)
Standard 10: AES FIPS 197	Compliant (OpenSSL FIPS)	Compliant (OpenSSL FIPS)	Non-compliant (No FIPS support)	Compliant (OpenSSL FIPS)

Design Choice Justification:

- **Selected: Raspberry Pi 4 (Broadcom BCM2711)**
  - This choice satisfies all critical requirements (image processing, encryption, transmission time) at the most economical cost ($68).
  - OpenCV and OpenSSL support ensure compliance with Requirement 10 and Standard 10.
  - Alternative options were dismissed due to performance limitations (STM32F446RE), exceeding budgetary constraints (Pi 5), or unsuitable physical dimensions (BeagleBone).

Implementation Details:

- **OS**: Raspberry Pi OS (64-bit) with Python 3.9.
- **Libraries**:
    - `opencv-python` (v4.5.5) for image processing.
    - `pycryptodome` (v3.18.0) for AES-256-GCM encryption.
- **GPIO Pinout**:
    - SPI0 for the RF transceiver (CE0/CE1, MOSI, MISO, SCLK).
    - UART for debug logs.

RF Transceiver Module

Criteria	NRF24L01+ (Nordic Semiconductor)	CC1101 (TI)	HC-12 (Si4438)
Req 70: 3kHz–300GHz	2.4 - 2.5 GHz	387 - 464 MHz	433 MHz
Const 40: Bit Rate	250kbps - 2 Mbps max	1.2kbps - 500 kbps	9.6 kbps
Req 22: Bidirectional	Yes	Yes	Yes
Req 50: Error Correction	Built-in protocol	Built-in FEC	None
Const 20: Cost	10 ~ $13.99	3 ~ $22.99	1 ~ $9.49
Const 80: >5m Range	10m - 30m (indoors)	100m	500m
Power Supply	1.9 - 3.6 V	1.8V - 3.6 V	3.3 V

Design Choice Justification

- **Selected: NRF24L01+ (Nordic Semiconductor)**
    - This module provides the fastest data rate (2 Mbps) among cost-effective alternatives, which is critical for meeting Requirement 40 (600-second deadline).
    - Bidirectional ACK/NACK support addresses Requirement 22 (retransmission capability).
    - **Rejected Options**:
        - HC-12 was too slow (9.6 kbps) and lacked error correction.
        - CC1101, while offering better range, incurred higher cost and increased

setup complexity.

Implementation Details:

- **Frequency**: 2.4 GHz (Channel 76 to avoid Wi-Fi overlap).
- **Data Rate**: 2 Mbps (for 1024x1024 PNGs, approximately ~1 MB/image, requiring ~500 kbps).
- **Python Library**: `RF24` (for NRF24L01+).

Image Processing Algorithm

Criteria	OpenCV	scikit-image	Pillow	YOLOv5
Const 20: Cost	Free	Free	Free	GPU needed
Compatibility	Python/C++	Python	Python	PyTorch

Design Choice Justification:

- **Selected: OpenCV (Hough Circles)**
  - This option offers good accuracy and speed.
  - Tunable parameters allow for optimization specifically for red circle detection.

Encryption Library

Criteria	OpenSSL (FIPS)	PyCryptodome	libsodium
Req 20: Encryption	AES-256	AES-256	XSalsa20
Std 10: FIPS 197	Certified	Not certified	Not certified
Const 20: Cost	Free	Free	Free
Language Support	Python/C	Python	Python/C

Design Choice Justification:

- **Selected: OpenSSL (FIPS)**
  - This is the sole FIPS-compliant option, adhering to Standard 10.
  - OpenSSL is widely used and has extensive documentation.

Mobile App Framework

Criteria	Flutter (Dart)	React Native (JS)	Native (Swift/Kotlin)
Req 80: Scrollable Table	ListView	FlatList	UITableView/RecyclerView
Req 65: 2FA Auth	Firebase Auth	React Native Auth	Manual impl
Const 20: Cost	Free	Free	2x dev effort
Std 30: IEEE 23026	Material Design	Custom UI	Native UI
Development Time	Single codebase	JS bridge	Double work

Design Choice Justification:

- **Selected: Flutter**
  - Flutter enables a **single codebase** for both Android and iOS, leading to cost savings as per Constraint 20.
  - **Built-in Firebase Auth** facilitates 2FA as required by Requirement 65.
  - **Material Design** compliance addresses Standard 30 (IEEE 23026).
  - **Rejected Options**:
    - React Native exhibited slower performance for image rendering.
    - Native development was deemed excessively time-consuming.

Implementation Details:

- **Dependencies**:
  - `firebase_auth` (for 2FA).
  - `cached_network_image` (for efficient image loading).

System Architecture Location: Website

Criteria	Local Computer	WordPress [2]	Wix [3]
Const 20: Cost	Free	$0-59/month	$16-59/month
Hosting	Self-managed on personal device	Fully managed	Fully managed
Customization	Complete control,	Good with plugins	Good with templates

	unlimited	(limited on free plan)	and app market
Performance	Excellent (with proper setup)	Good	Good
Support	No support, plentiful online resources	Email, live chat (paid plans)	24/7 support on all plans

Design Choice Justification:

- **Selected: Local Computer**
  - Local Computer enables a **high scalability** for a website and free, leading to cost savings.
  - **Self management allows** us to change anything we want to change on the Local Computer instance.
  - **Rejected Options**:
    - WordPress, the customization and security systems are not worth the price for the premium version of WordPress. Scalability is not enough for our purposes.
    - Wix, the customization and security systems are not worth the price for just starting use of Wix. Scalability is not enough for our purposes.

4.3

Test and Evaluation Master Plan

Requirements Test Plan

Req No.	Test Method	Evaluation Method	Threshold	Objective
10	Raspberry Pi will be provided with USB drive containing 100 test images	Direct observation that images are loaded into memory without errors	100% of images loaded successfully	The Raspberry Pi shall load images from the USB drive.
20	Run OpenCV Hough Circle detection algorithm on all 100 images	Compare identified images against ground truth list of 10 images with red circles	100% accuracy (all 10 identified, 0 false positives)	The Raspberry Pi shall identify images with red circles.
30	Apply AES-256-GCM encryption to test image using OpenSSL	Verify ciphertext differs from plaintext; measure encryption time	Successful encryption with valid AEAD tag	The Raspberry Pi shall have the ability to encrypt images.
40	Receive encrypted test image and decrypt using matching key	Binary comparison of decrypted image against original image	100% match (bit-for-bit identical)	The Rebel Server shall have the ability to decrypt images.
50	Transmit test packet via NRF24L01+ at 2.4 GHz	Transmit data from Raspberry Pi and confirm that data is correctly received on Rebel Server	Successful transmission detected on correct frequency	The Raspberry Pi shall have the ability to transmit and receive data with radio frequency waves.
60	Configure Rebel Server with matching NRF24L01+ module	Transmit data from Rebel Server and confirm that data is correctly received on Raspberry Pi	Successful reception with 0 bit errors	The Rebel Server shall have the ability to transmit and receive data with radio frequency waves.
70	Transmit frame with known checksum;	Send packet with incorrect data to Rebel	100% error detection rate	After receiving a transmission, the Rebel Server

	intentionally corrupt packet	Server, then confirm that the Rebel Server sends a NACK		shall perform error checking.
80	Inject transmission error and verify NACK sent	Send packet with incorrect data to Rebel Server, then confirm that the Rebel Server sends a NACK	NACK transmitted correctly with frame ID	In the case of an error, the Rebel Server shall send a transmission to the Raspberry Pi requesting to resend the data.
90	Simulate NACK reception at Raspberry Pi	Send NACK to Pi, then check if retransmitted frame is received by Rebel Server	Correct frame retransmitted	If the Raspberry Pi receives a request to resend a transmission, then it will send the image again.
100	Time complete transmission cycle with stopwatch	Record elapsed time from start to completion	≤ 600 seconds total transmission time	It shall transmit in 600 seconds or less.
110	Access public video URL without authentication	Load and play video from multiple devices	Video accessible without login	Any user shall have access to the first video.
120	Attempt to access authenticated video without credentials	Attempt to authenticate with incorrect credentials and confirm that access is denied; authenticate with correct credentials and confirm access is accepted	Unauthorized users blocked; authorized users granted access	Authorized users shall have exclusive access to the second video.
130	Implement FIDO2 two-factor authentication	Test authentication flow with security key and verify non-password based access	FIDO2 authentication successful without password	The mobile app shall use a secure method of authentication to identify authorized users.

140	Load mobile application on mobile device	View images and confirm they are displayed in a scrollable table	Images displayed in scrollable table	The mobile app shall display images in a scrollable table.
150	Count number of images displayed in gallery	Manual count of images and comparison to expected value	Exactly 10 images displayed	The mobile app shall display up to 10 images.
160	Verify each red circle content appears once	Manual inspection of table to confirm no duplicates	Each red circle image appears exactly once	The scrollable table shall display the contents of each red circle exactly once.

Constraints Test Plan

Req No.	Test Method	Evaluation Method	Threshold	Objective
10	The budget spreadsheet will be analyzed to ensure the total cost is $300 or less.	Manual inspection of budget	Total cost no more than $300	The total cost of the project shall not exceed $300 without Faculty Advisor approval.
20	Ensure only appropriate equipment is used.	Documenting of all equipment used	Only provided equipment or components purchased within the budget are used	The solution shall use only the provided equipment plus additional components purchased within the budget; a personal mobile application host device is required.
30	Place the Raspberry Pi and all equipment connected to it into the box.	Visual inspection that ensures no objects or parts are present outside or extend out from the box	All equipment fits within the box	The Raspberry Pi and all equipment connected to it must fit within the provided box.

40	Confirm that all Wi-Fi, Bluetooth, and cellular functionalities are disabled, and no wires directly connect the Raspberry Pi to the Rebel Server.	Visual inspection of computer settings to ensure all wireless communication is disabled	No Wi-Fi, Bluetooth, cellular communication, or wired connections are used.	The Raspberry Pi and Rebel server shall not communicate with each other using Wi-Fi, Bluetooth, cellular communication, or a wired connection.
50	Use a tape measure or other measuring device to determine the distance.	Physical measurement of distance	The Rebel Server is at least 5 meters from the Raspberry Pi.	The Rebel Server and all its components shall be placed at least 5 meters from the Raspberry Pi.
60	Use the transceivers to receive and send test data, and mark the signals that were used during transportation.	Send data with RF transceivers and confirm that it is received correctly	Following the standards set by the FCC, without breaking the said set standards.	The range of frequencies used shall be compliant with FCC regulations.

Standards Test Plan

Req No.	Test Method	Evaluation Method	Threshold	Objective
10	Encrypt and decrypt data using OpenSSL's AES encryption.	Compare encrypted and decrypted data and confirm they differ	Ciphertext differs from plaintext, and data can be accurately decrypted	The solution shall comply with AES FIPS 197-upd1 for encryption.
20	Use the transceivers to receive and send test data, and mark the signals that were used during transportation.	Send data using the RF transceivers and confirm that it is received correctly	Following the standards set by the FCC, without breaking the said set standards.	The solution shall comply with C95.3-2021: Standard for Radio Frequencies

30	Ensure that mobile application follows all appropriate guidelines	Visual inspection that shows the website's content follows user accessibility guidelines as well as security concerns.	The application functions correctly and can display images and videos	The mobile application shall comply with the IEEE/ISO/IEC 23026-2023 standard for website design.
40	Inspect code and ensure it complies with all style guidelines	Review code and verify compliance	All written Python code complies with PEP 8.	All Python code shall comply with PEP 8 for coding style.

4.4

Implementation Plan

The Implementation Plan provides the final details necessary for realization of your project. These details include an interface control specification, budget, schedule, and risk analysis.

Interface Control

Hardware

Raspberry Pi 4 Connections

- **USB 3.0 Port** → HW2USB Storage Device
    - Protocol: USB Mass Storage Class
    - Data Rate: 5 Gbps (USB 3.0)
    - File System: FAT32

GPIO Header (40-pin) → HW3 NRF24L01+ Transceiver

- SPI0 Interface:
    - Pin 19 (MOSI) → NRF24L01+ MOSI
    - Pin 21 (MISO) → NRF24L01+ MISO
    - Pin 23 (SCLK) → NRF24L01+ SCLK
    - Pin 24 (CE0) → NRF24L01+ CSN
    - Pin 22 (GPIO25) → NRF24L01+ CE

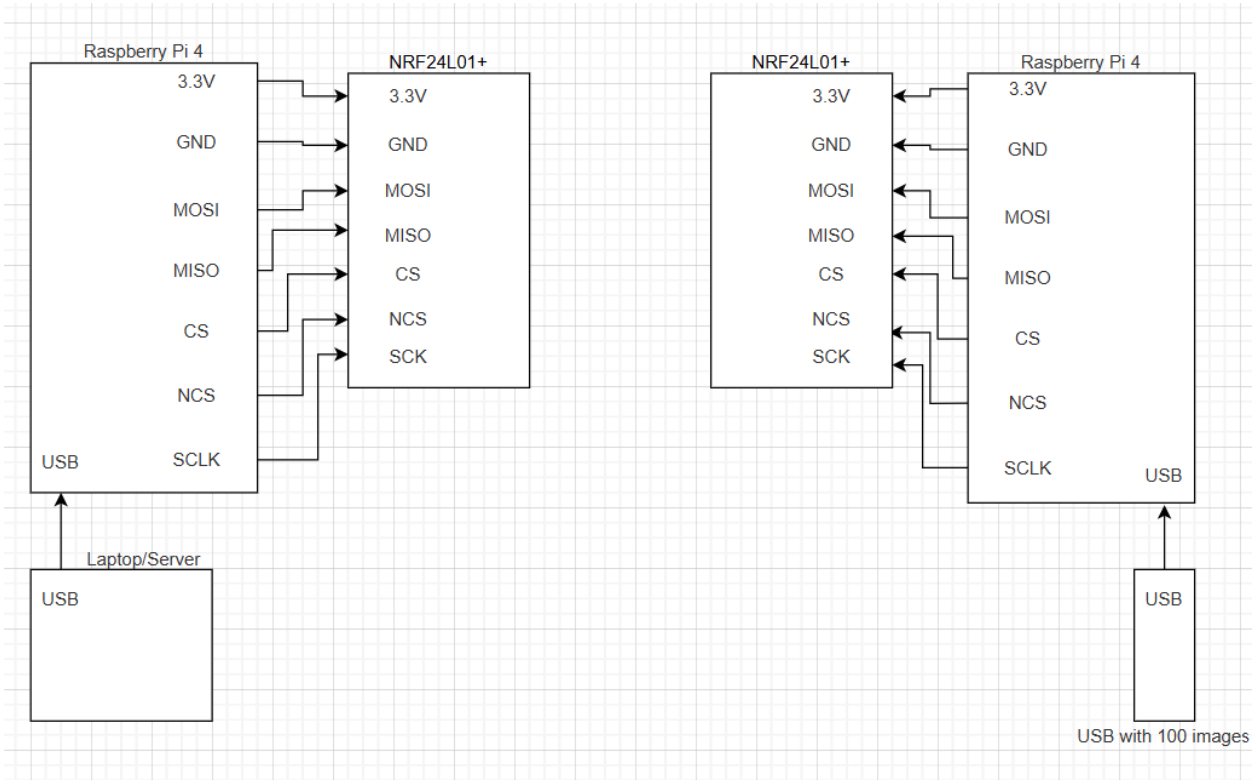
Power:

- Pin 1 (3.3V) → NRF24L01+ VCC
- Pin 6 (GND) → NRF24L01+ GND

UART Interface → Debug Console

- Pin 8 (TXD) → USB-to-Serial Adapter RX
- Pin 10 (RXD) → USB-to-Serial Adapter TX
- Baud Rate: 115200, 8N1

Implementation Plan
Interface Control
Hardware:



Software:

Image Processing Software: Loads 1024 x 1024 PNG images from the USB drive and outputs cropped images.

Image Sender Software: Communicates with the NRF24L01+ receiver by fragmenting the PNG file into 32 byte packets and then sending those packets using a 2.4 GHz radio band transmission.

Image Receiver Software: Communicates with NRF24L01+ transmitter by assembling 32 byte packets into a PNG file.

Mobile application: Fetches PNG image files and MP4 video files from the web server and displays them to the user with a GUI.

Budget

Your hardware purchase budget must be consistent with the ordering process and spreadsheet provided. It should include an itemized list of parts with prices from approved vendors.

Budget:

Vendor	Item #	Description	Qty	Price	Total	Link
Amazon	BC22523	NRF24L01+ Transceivers	1	$13.99	$13.99	https://www.amazon.com/Makerfire-Arduino-NRF24L01-Wireless-Transceiver/dp/B00O9O868G
Amazon	765756931182	Raspberry Pi 4B 4GB	1	$67.99	$67.99	https://www.amazon.com/Waveshare-Accessory-Compatible-Raspberry-Networking/dp/B09TTNF8BT?source=ps-sl-shoppingads-lpcontext&ref_=fplfs&psc=1&smid=AIKUR5C8RLL4Z

Bill of Materials:

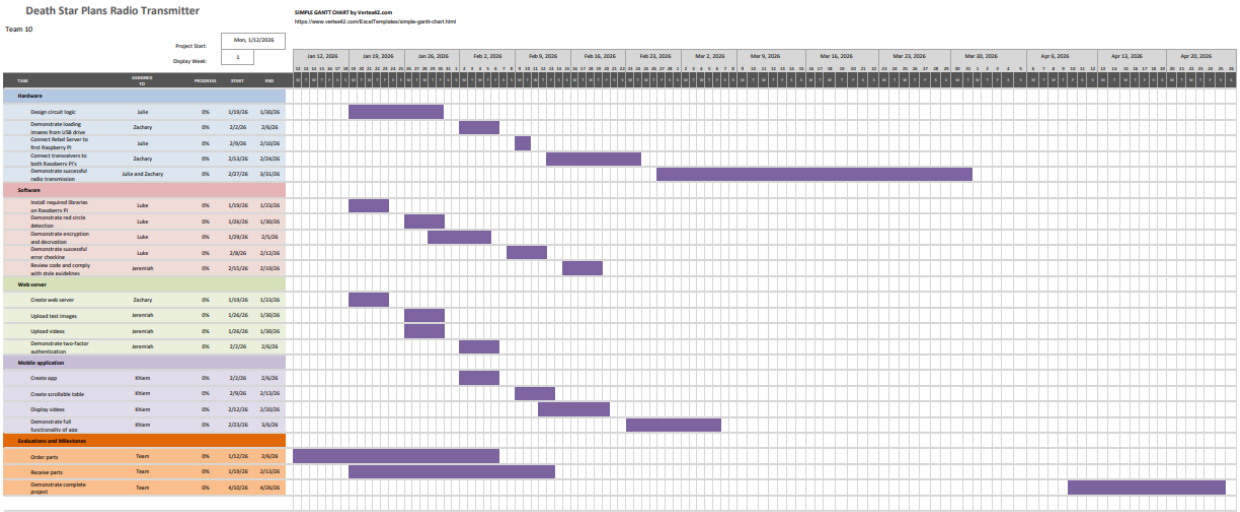
1. Raspberry Pi 4B $30.00
2. Raspberry Pi 4 (Broadcom BCM2711) $67.99
3. NRF24L01+ Transceivers $13.99
4. Jumper Wires $5.97
5. 2 Power Cord $10.00

---------- ----------

Total: $127.95

Schedule:

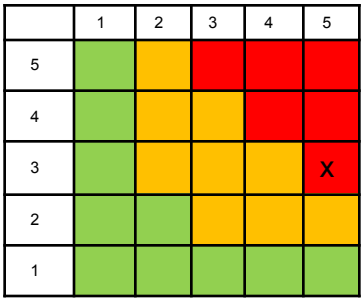
gantt-chart



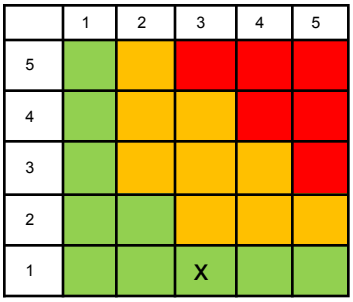
Risk and Mitigations: The graphs show the likelihood of occurrence through the rows and severity of impact through the columns.

- Risk of late/damaged parts. Likelihood is low but the severity of the impact is high. Minimize the likelihood by ordering parts as early as possible and inspecting them as soon as received.

Unmitigated:

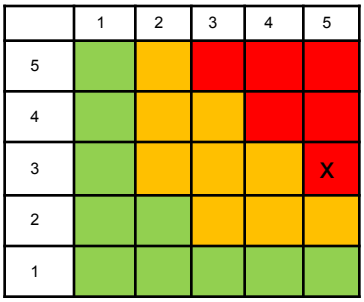


Mitigated:

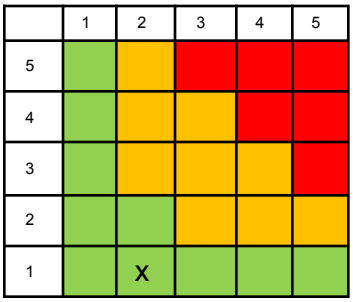


- Risk of incorrect parts being ordered. Likelihood is low and severity is moderate. There is still money left in our budget if we need to order another part.

Unmitigated:

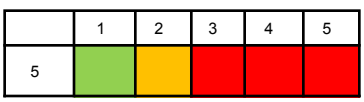


Mitigated:

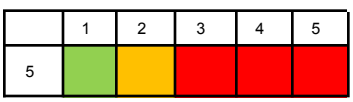


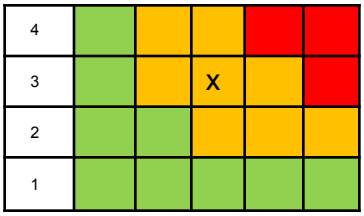
- Risk of insufficient radio bandwidth. Likelihood is moderate and severity is moderate. Order and test radio early to ensure bandwidth will be high enough to transmit images in time limit.

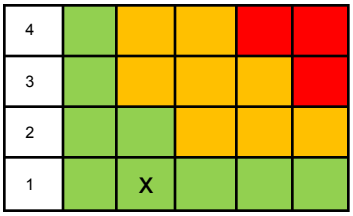
Unmitigated:



Mitigated:



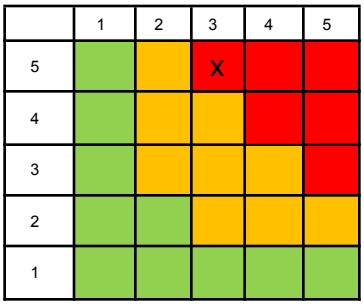


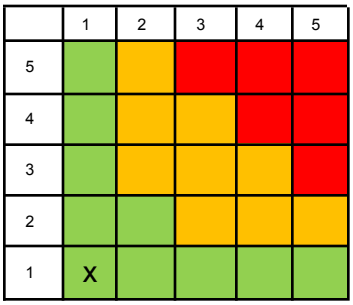


- Risk of violating the FCC regulations. Likelihood is low and severity is high. Minimize the likelihood by constantly checking the guidelines and ensuring the frequencies do not violate them.

Unmitigated:

Mitigated:

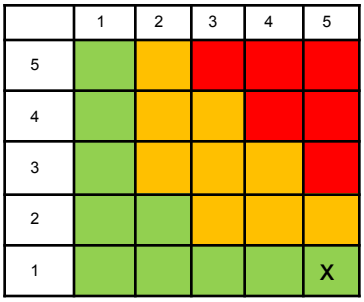


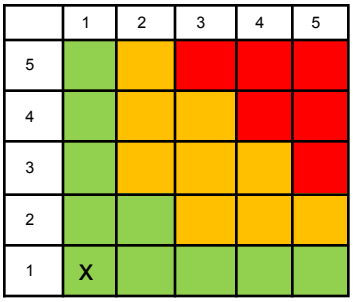


- Risk of electrocution. Likelihood is low and severity is high. Follow proper safety precautions when handling all electronic components.

Unmitigated:

Mitigated:

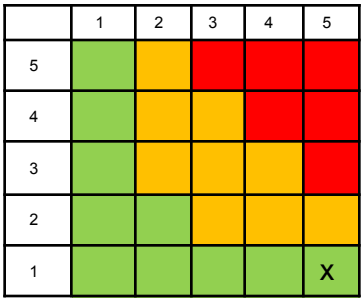


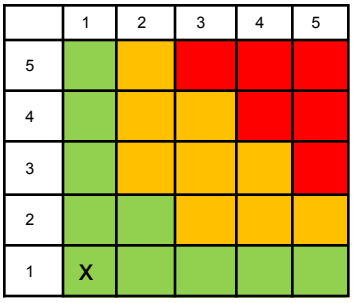


- Risk of mobile application not being fully functional. Likelihood is low and severity is high. Minimize likelihood by creating the application to do the bare minimum to meet requirements before adding anything else.

Unmitigated:

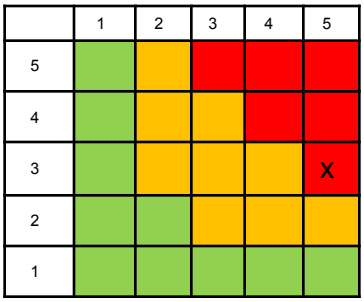
Mitigated:





- Risk of leaked AES key. Likelihood is low and severity is high. Minimize likelihood by replacing compromised keys and ensuring there is another form of encryption and decryption that can be used.

Unmitigated:



Mitigated: