­

4/22/2025

Exfiltrate & Disseminate Death Star Plans

Team: 12

Project Sponsor:

* Dr. Michael Raymer
* Dr. Weisong Wang
* Billy Kelly
* Dr. Thomas Wischgoll
* Dr. Zhiqiang Wu

Primary Email  
someonewhocares@wright.edu

Document Tracking Info:

Document Version: Version 2

Date: 4/19/2025

Final Approval Date: xx/xx/xxxx

Final Approval By: <name> <email>

Authors: Kalen Tullis, Simran Cheema, Divyesh Ambaliya, Liliane Owens, Nathaniel Rieder

Wright State University

TABLE OF CONTENT

[1 Executive Summary 2](#_Toc184026197)

[1.1 Purpose of this document 2](#_Toc184026198)

[1.2 Design Scope 2](#_Toc184026199)

[1.3 Intended Audience and Document Overview 2](#_Toc184026200)

[1.4 Definitions, Acronyms, and Abbreviations 2](#_Toc184026201)

[1.5 Design Document Conventions 3](#_Toc184026202)

[1.6 References and Acknowledgments 3](#_Toc184026203)

[2 Problem Statement 5](#_Toc184026204)

[2.1 Historical Introduction 5](#_Toc184026205)

[2.2 Alternative Approaches 6](#_Toc184026206)

[2.3 Impact of Success 9](#_Toc184026207)

[3 Context of Design Solution 10](#_Toc184026208)

[3.1 Design Objectives 10](#_Toc184026209)

[3.2 Design Assumptions 11](#_Toc184026210)

[3.3 Design Requirements 12](#_Toc184026211)

[3.4 Design Constraints 12](#_Toc184026212)

[3.5 Design Standards 13](#_Toc184026213)

[3.6 Design Functionality 14](#_Toc184026214)

[3.7 User Characteristics 14](#_Toc184026215)

[3.8 Operating Environment 14](#_Toc184026216)

[4 Technical Approach 15](#_Toc184026217)

[4.1 Hardware 15](#_Toc184026218)

[4.2 Software 17](#_Toc184026219)

[4.3 Budget 19](#_Toc184026220)

[5 Appendix: Test and Evaluation Master Plan and Report 20](#_Toc184026221)

# 1 Executive Summary

This document outlines the completed system design for the Rebel Server Communication System, a secure short-range radio solution for covert data transmission between a Raspberry Pi and a Rebel Server. The system transmits encrypted images across a monitored space without reliance on Wi-Fi, Bluetooth, or cellular networks. This document adheres to the design standards required by Wright State’s College of Engineering Senior Design program and details the final project scope, motivation, implemented design specifications, testing procedures, and evaluation of results.

## 1.1 Purpose of this document

## The purpose of this document is to describe the construction, implemented functions, and final technical specifications of the Rebel Server Communication System. The hardware architecture includes a Raspberry Pi, Rebel Server, and short-range radio transceivers used for secure data transmission. The software architecture includes image identification, encryption, and checksum verification, which together enabled reliable, authenticated communication between devices without physical connections.

## 1.2 Design Scope

The Rebel Server Communication System was built to operate within a secure environment, though testing demonstrated a reliable transmission range of approximately 2 meters. This foundational setup still demonstrates the potential of short-range radio as a discrete, secure communication alternative, offering applications in highly monitored facilities, research labs, and secure data-handling environments.

## 1.3 Intended Audience and Document Overview

This document is intended for project sponsors, technical advisors, security engineers, and future developers interested in secure communication solutions. Section 1 provides a summary, Sections 2 and 3 describe the system’s purpose and context, while Sections 4 and 5 present the implemented technical approach and the testing results that verified system performance. Readers seeking a complete technical overview are encouraged to review the entire document.

## 1.4 Definitions, Acronyms, and Abbreviations

* IEEE: Institute of Electrical and Electronics Engineers
* FCC: Federal Communications Commission
* ISM Band: Industrial, Scientific, and Medical Band, used for short-range communication
* Checksum: A computed value used to verify data integrity
* Short-Range Radio: A low-power radio used for close-range data transmission
* Raspberry Pi: A small computer capable of many applications.
* Rebel server: Computer server that the Raspberry Pi will be transmitting data to.

## 1.5 Design Document Conventions

Citations are in IEEE format.

## 1.6 References and Acknowledgments

[1] A. Marget, “Data Encryption: How It Works & Methods Used,” Unitrends, Jan. 27, 2022. Available: <https://www.unitrends.com/blog/data-encryption>

[2] Abdi, Masoud, and Saeid Nahavandi. “Multi-Residual Networks: Improving the Speed and Accuracy of Residual Networks.” ArXiv.org, 2016, [https://arxiv.org/abs/1609.05672. Accessed 18 Oct. 2024](https://arxiv.org/abs/1609.05672.%20Accessed%2018%20Oct.%202024).

[3] Daniel, W, and Edward Fine. “An Evaluation of Training Size Impact on Validation Accuracy for Optimized Convolutional Neural Networks.” SMU Data Science Review, vol. 1, no. 4, 2018, p. 12, <https://scholar.smu.edu/cgi/viewcontent.cgi?article=1046&context=datasciencereview>

[4] IBM, “What are Convolutional Neural Networks? | IBM,” www.ibm.com, 2024. Available: <https://www.ibm.com/topics/convolutional-neural-networks>

[5] “Infrared Explained: The Science Behind the Rays | Lenovo US,” Lenovo.com, 2021. Available: <https://www.lenovo.com/us/en/glossary/what-is-infrared/?orgRef=https%253A%252F%252Fwww.google.com%252F>. [Accessed: Oct. 16, 2024]

[6] J. Nesfield, “StackPath,” www.electronicdesign.com, Jun. 24, 2019. Available: <https://www.electronicdesign.com/markets/automation/article/21808186/sending-data-over-sound-how-and-why>

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 2015. Available: <https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf>

[8] L. Matzen, Z. Gastelum, B. Howell, “Effects of machine learning errors on human decision-making: manipulations of model accuracy, error types, and error importance,” Cogn. Research 9. Aug. 26, 2024. Available: <https://doi.org/10.1186/s41235-024-00586-2>

[9] Makkar, Tanya, et al. “Analogizing Time Complexity of KNN and CNN in Recognizing Handwritten Digits.” IEEE Xplore, 1 Dec. 2017, <https://ieeexplore.ieee.org/document/8313707/similar>. Accessed 18 July 2021.

[10] M. Paszynski, KranzlmüllerD., V. V. Krzhizhanovskaya, J. J. Dongarra, and P. Sloot, Computational science - ICCS 2021: 21st international conference, Krakow, Poland, June 16-18, 2021: proceedings. Part II. Cham: Springer, 2021.

[11] O. Abood and S. Guirguis, "A Survey on Cryptography Algorithms," Int. J. Sci. Res. Publ., vol. 8, no. 7, pp. 495–516, 2018, doi: 10.29322/IJSRP.8.7. 2018.p7978.

[12] Openstax, “14.1 Speed of Sound, Frequency, and Wavelength - Physics | OpenStax,” openstax.org. Available: <https://openstax.org/books/physics/pages/14-1-speed-of-sound-frequency-and-wavelength>

[13] P. Borisovas, “Twofish encryption: What is it and how does it work? | NordVPN,” nordvpn.com, Mar. 24, 2024. Available: <https://nordvpn.com/blog/what-is-twofish-encryption/>

[14] Pytorch Team, “ResNet”. Pytorch, <https://pytorch.org/hub/pytorch_vision_resnet>.

[15] Puneet, “What is the Twofish encryption algorithm? | Encryption Consulting,” Encryption Consulting, Sep. 23, 2020. Available: <https://www.encryptionconsulting.com/education-center/what-is-twofish/>

[16] RYAN D’AMATO, “ICT Projects - Software-Defined Radio: At The Forefront Of Data Transmission,” ICT Projects, Jul. 10, 2023. Available: <https://ictprojects.mt/2023/2023/software-defined-radio-at-the-forefront-of-data-transmission/>. [Accessed: Oct. 09, 2024]

[17] R. Sanders, “Elliptic Curve Cryptography: What is it? How does it work?,” Keyfactor, Nov. 29, 2022. Available: <https://www.keyfactor.com/blog/elliptic-curve-cryptography-what-is-it-how-does-it-work/>

[18] “SATEL – Radio technology for various data transfer needs,” SATEL, Aug. 19, 2024. Available: <https://www.satel.com/products/>. [Accessed: Oct. 17, 2024]

[19] skullsinthestars, “How fast can we communicate with light and radio waves?,” Skulls in the Stars, Jul. 22, 2023. Available: <https://skullsinthestars.com/2023/07/22/how-fast-can-we-communicate-with-light-and-radio-waves/>. [Accessed: Oct. 17, 2024]

[20] TensorFlow. “Convolutional Neural Network (CNN) | TensorFlow Core.” TensorFlow, 2019, [www.tensorflow.org/tutorials/images/cnn](http://www.tensorflow.org/tutorials/images/cnn).

[21] “Time complexity and optimality of kNN,” nlp.stanford.edu, Apr. 07, 2009. Available: <https://nlp.stanford.edu/IR-book/html/htmledition/time-complexity-and-optimality-of-knn-1.html>

[22] towkeershah40, “Infrared Light for Transmission,” Geeks for Geeks, Oct. 10, 2022. Available: <https://www.geeksforgeeks.org/infrared-light-for-transmission/>. [Accessed: Oct. 14, 2024]

[23] Wikipedia, “Advanced Encryption Standard,” Wikipedia, May 07, 2019. Available: <https://en.wikipedia.org/wiki/Advanced_Encryption_Standard>

[24] “Train K-Nearest Neighbor Classifier (Image Analyst)—ArcGIS Pro | Documentation,” Arcgis.com, 2024. Available: <https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/train-k-nearest-neighbor-classifier.htm>. [Accessed: Oct. 09, 2024]

[25] “Transfer Data with Animated QR codes,” Dynamsoft Developers Blog, Sep. 30, 2021. Available: <https://www.dynamsoft.com/codepool/transfer-data-with-animated-qr-codes.html>. [Accessed: Oct. 14, 2024]

[26] “1.6. Nearest Neighbors,” nam04.safelinks.protection.outlook.com. Available: [https://scikit-learn.org/stable/modules/neighbors.html](https://scikit-learn.org/stable/modules/neighbors.html )

[27] X. Chen et al., “Covert Communications: A Comprehensive Survey,” IEEE Communications Surveys & Tutorials, vol. 25, no. 2, pp. 1173–1198, Jan. 2023, doi: <https://doi.org/10.1109/comst.2023.3263921>.

# 2 Problem Statement

The challenge was to securely exfiltrate sensitive Death Star plans from a restricted, highly monitored environment to a Rebel server. Our team developed and implemented a system that transfers these plans from a Raspberry Pi device to the Rebel Server without detection by the guards. The server then securely reveals the plans on a mobile application.

The transmission was required to be wireless, avoiding Wi-Fi, cellular, and Bluetooth to reduce the risk of detection. It also had to be encrypted and completed within 10 minutes. The solution prioritized security, cost-effectiveness, and successful data transfer without alerting the Rebel guards, as any detection could compromise the mission and expose sensitive information.

**Existing Solutions**

1. Standard Wireless Communication Protocols: Wi-Fi, Bluetooth, and cellular networks are common but easily detectable in restricted environments.
2. Infrared and QR Code-based Communication: These solutions work for certain applications but have limitations in range, data capacity, and require a line of sight.

**Implemented Solution**

To meet the requirements, we implemented a secure, short-range radio-based communication system to transmit encrypted data between the Raspberry Pi and the Rebel Server without detection. Unlike standard wireless protocols, our system used cost-effective, low-power radio transceivers (CC1101 modules) designed specifically for limited-range data transmission.

Although the intended range was 5 meters, testing demonstrated a reliable transmission range of approximately 2 meters, which required placement adjustments to meet communication needs. Despite this limitation, the system successfully transmitted encrypted image data wirelessly without relying on Wi-Fi, Bluetooth, or cellular. This approach demonstrates a feasible and adaptable method for undetected data transfer in secure or monitored environments.

## 2.1 Historical Introduction

There has always been a high importance in secure, wireless data transmissions. This has only increased as wireless data transmission becomes more commonly used for moving data. These transmissions need protection to prevent access from malicious entities to the data. This is even more necessary for critical data such as military transmissions. This data needs to be transferred with protection from any breach to prevent its misuse. This means that measures need to be taken to guarantee the safe transmission of data.

## 2.2 Alternative Approaches

**Communication:**

Alternative solutions include using Software Defined Radio, QR Codes, Infrared Light and sound waves for data transmission. Software Defined Radio (SDR) transmits data by generating and modulating radio waves through software, allowing dynamic adjustments to frequency, bandwidth, and protocols for wireless communication over long or short distances. SDR offers reliability with a moderate time, but it is out of budget and will cost more than $300. QR codes transmit data visually by encoding it into a grid of black and white squares. A scanning device decodes the image into text, URLs, or other information. This approach requires line-of-sight, which makes it less effective. Infrared light transmits data by modulating invisible light waves, typically using LED emitters and photo detectors to send and receive binary signals in line-of-sight environments. Infrared communication also offers a line-of-sight method for secure and low power data transfer but may require precise alignment between devices and have limited range. Sound waves transmit data by encoding it into audible or ultrasonic frequencies, which are emitted by speakers and decoded by microphones or specialized receivers. Sound waves are cost effective and do not require line of sight but there are chances of interference to the data transfer because of background noises.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Communication [req#20]*** | ***Reliability*** | ***Time [req#23]*** | ***Cost [req#80]*** | ***Line of Sight [opt]*** |
| *Software Defined Radio* | *High [16]* | *Moderate [16]* | *$300-400* | *No* |
| *QR codes* | *High [25]* | *Fast [25]* | *$150-300* | *Yes* |
| *Infrared light* | *Moderate [5]* | *Fast [22]* | *$20-40* | *Yes* |
| *Sound waves* | *Moderate [6]* | *Moderate [12]* | *$50* | *No* |
| *Short range radio* | *High [18]* | *Moderate [19]* | *$100-200* | *No* |

Alternative communication methods considered included Software Defined Radio (SDR), QR codes, infrared light, and sound waves. SDR offered high reliability and flexibility but exceeded our project’s budget of $300. QR codes and infrared provided fast, secure transmission but required line-of-sight and were limited by range. Sound waves were cost-effective but highly susceptible to environmental interference.

We implemented short-range radio using CC1101 modules, which offered sufficient flexibility and moderate bandwidth within budget constraints. Although our planned range was 5 meters, testing demonstrated reliable transmission at approximately 2 meters with no obstructions. This solution proved practical for our environment and successfully enabled secure wireless communication without detection.

**Encryption:**

For encryption alternative approaches include RSA, DES, Blowfish, TwoFish and ECC. RSA offers high security, but it is very slow, has a small key length and consumes a lot of energy. DES is also very similar to RSA, but it does not have a high security level and is moderate. Blowfish offers low energy consumption and high security, but it has a moderate speed. TwoFish has a variety of choices when it comes to key length and has High security. But it is slow in speed and consumes more energy than needed. ECC is also another good choice, but its speed and key length varies based on the ECC method used, making it more difficult to use.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Encryption [req#3]*** | ***Speed [req#23]*** | ***Key Length [req#10]*** | ***Energy Consumption[opt]*** | ***Security Level [opt]*** |
| *AES* | *Fast [23] [11]* | *128,192 or 256 bits [1]* | *Low [1]* | *High [1][11]* |
| *RSA* | *Slow [11]* | *1024 or 2048 bits [1]* | *Moderate [1]* | *High [1] [11]* |
| *DES* | *Slow [11]* | *56 bits [1]* | *High [1]* | *Moderate [1] [11]* |
| *BLOWFISH* | *Moderate [10]* | *32 to 448 bits [2]* | Low [10] | *High [10]* |
| *TWOFISH* | *Slow [13]* | *128, 192, or 256 bits. [15]* | Moderate [15] | *High [15]* |
| *ECC* | *Varies [11]* | *Varies* | Very Low [17] | *Very High [17] [11]* |

We evaluated multiple encryption options, including RSA, DES, Blowfish, TwoFish, and ECC. Based on speed, security, and energy efficiency, AES was selected as the most appropriate algorithm for our system. While we initially considered AES in CBC mode, we implemented AES in ECB mode using the Python pycryptodome library due to its lower resource requirements on the Raspberry Pi. AES offered fast encryption, flexible key lengths, and acceptable memory usage for our constrained hardware setup. Though ECB mode is generally less secure than CBC, the lack of IV transmission and the small, compressed image sizes in our system made ECB a reasonable trade-off between simplicity and effectiveness.

**Image Identification**

Image Identification alternatives include using a custom coded heuristic, K-Nearest Neighbor algorithm, and Convolutional Neural Network. A custom heuristic would be a quick coded solution using the assumptions provided but suffer greatly in accuracy. K-Nearest Neighbor (KNN) algorithm memorizes the location of data points and computes an average of nearest neighbors to determine classification. The KNN algorithm is computationally slow and suffers on more complicated datasets. The Convolutional Neural Network (CNN) is a machine learning algorithm that inputs data through layers of artificial neurons to make decisions. CNN would also be an acceptable solution but is more general use than image recognition. A Residual Network is a subset of CNN that uses skip connections which is beneficial for deep neural networks.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Image Identification [req#10]*** | ***Accuracy [req#10]*** | ***Time complexity [opt]*** | ***Existing Libraries [opt]*** |
| *Custom Image Heuristic* | *Low* | *Estimated O(n2)* | *No* |
| *K-Nearest Neighbor* | *96%*  *MNIST [9]* | *O(nd) [21]* | *Yes [26]* |
| *Convolutional Neural Network* | *90 - 98.21%*  *Fashion MNIST [3]* | *O(n) [4]* | *Yes [20]* |
| *Residual networks* | *96.27%*  *CIFAR-10 [2]* | *O(n) [7]* | *Yes [14]* |

Image recognition is an important step in determining Death Star images from the total dataset. Numerous machine learning algorithms can be used for image recognition with varying results. The choice of algorithm is important to find the compromise between accuracy and computationally expensive. Accuracy is defined as a percentage of correct predictions made from all predictions by the model. Error rate is the complement percentage determining number of incorrect predictions [8].

|  |  |
| --- | --- |
| (Equation 1) |  |
| (Equation 2) |  |

Accuracy can be interpreted as a range given the stochasticity built in the algorithms. The time complexity of the algorithm is important to efficiently recognize Death Star images and determine algorithm feasibility. The time complexity measures the time to classify an image not including time for training/validation. Time complexity is important, but time is not a limiting factor for image recognition therefore making it an optional criterion. Existing libraries for these algorithms are also an optional criterion as some of these algorithms have more than one. Existing libraries will greatly cut down on development time and allow efficient testing of algorithms using variable input parameters. Custom Image Heuristic would use computation shortcuts and assumptions about image contents. This algorithm may be competitively time efficient but does not use all available information within the image. The KNN accuracy was measured using MNIST dataset, the CNN accuracy was measured using fashion MNIST dataset and Residual Network accuracy was measured using CIFAR-10 dataset. The most complex of these datasets would be CIFAR-10 because it uses the largest images, tied for most classes, and uses color images. With these 3 criteria the chosen algorithm is Residual Network because of its competitive accuracy, competitive time complexity, and existing library. The Residual Network is also shown to be more effective on complex datasets compared to KNN and CNN due to its skip connections. These skip connections allow it to average out poor learned features and focus on the meaningful ones.

The solution was implemented using a Residual Network for image recognition. The neural network was able to detect all death stars with extra false positives. The accuracy can be improved with an improved training dataset. The images recognized by the neural network are saved in a separate folder in preparation for image transfer.

## 2.3 Impact of Success

1. Cultural

The Death star can destroy entire cultures in an instant. If it is destroyed, we preserve any of the potential targeted cultures.

1. Economics

When the Death Star destroys a planet, the planet’s population and the planet’s resources are destroyed, placing a permanent stop on all trading in and out of that planet. If the Death Star is destroyed, this event is prevented from happening.

1. Environmental

By destroying the Death Star, we could prevent the total destruction of many environments, preserving them from complete loss.

1. Galactic

The Death Star represents a constant threat of power over everyone in the galaxy. By destroying it, we would give the citizens a new hope for a future without threat to their life.

1. Public health

When the Death Star destroys a planet, it leaves behind a massive amount of space dust and debris. These remains would be highly irradiated and be hazardous for the immediate and surrounding areas, where the debris would lie.

1. Public safety

The Death Star represents a constant threat to public safety, and by destroying it we can remove this threat.

1. Public welfare

The Death Star is a constant source of fear for all citizens in the galaxy. If it is destroyed, the citizens can live on without this fear.

1. Social

Destroying the Death Star would be a show of strength from the rebel alliance. This will lead to an increase in rebel sympathy from citizens as we demonstrate our capabilities against the empire. This sympathy can cause a split in populations between those who sympathize and those who do not.

# 3 Context of Design Solution

## 3.1 Design Objectives

***Objective 1:*** The proposed solution will identify the 10 death star images.

All Death Star images are being received in a pool of 100 different images. From these images the solution will identify only the 10 Death Star images with weaknesses outlined in a red circle. Each image will be delivered as a PNG with a shape of 1024x1024 pixels on a USB Type A thumb drive.

***Objective 2:*** The proposed solution will transmit the images to the server images discreetly.

The Death Star guards are monitoring all Wi-Fi, Bluetooth, and cellular signals. Also, no physical connection between the server and the raspberry pi is possible. The transmitting of images must be done within a ten-minute period and avoid monitored methods or else the guards will detect the transmission.

***Objective 3:*** The proposed solution will transmit the images to the server securely and verifiably.

All communication between the Raspberry Pi and the Rebel server must be encrypted to ensure the transmitted data's integrity. The raspberry pi will receive confirmation of lossless data transmission from the server, which will be situated at least 5m from the raspberry pi. Non-verified images will be retransmitted to the server until all images are verified as sent.

***Objective 4:*** A mobile app will display Death Star weaknesses.

The mobile app will be hosted on the Rebel Server where weaknesses will be accessible to the rebel alliance. The images will show only the relevant section of the image originally outlined in a red circle. All weaknesses will be visible in a scrollable grid.

## 3.2 Design Assumptions

***Assumption 1:*** The Raspberry pi and the rebel server will have uninterrupted access to an American standard 120V wall outlet.

***Assumption 2:*** The team will be provided the one hundred 1024 x 1024 8-bit color-depth PNG images on a USB flash drive.

***Assumption 3:*** The 10 death star PNGs will be distinctly different from the remaining 90 PNGs.

The 10 death star images will be different in composition and framing by featuring a light-colored circle on a dark-colored background.

***Assumption 4:*** The 10 death star images will have a red circle to identify the death star weaknesses.

***Assumption 5:*** There will be no radio frequency interference outside permitted FCC limits within the building.

***Assumption 6:*** There will be no other team using the same radio frequency during the data transmission to and from the server.

***Assumption 7:*** The team will be provided with a Raspberry pi 4 model B.

***Assumption 8:*** The team will be provided with a computer mouse, a monitor, and a keyboard.

***Assumption 9:*** The team will be provided with a server running a recent version of Linux.

***Assumption 10:*** The server will have multiple USB type A ports.

***Assumption 11:*** The server will be no less than 5 meters away from the designed solution.

***Assumption 12:*** The server and solution will be in adjacent rooms separated by a door with a glass window.

***Assumption 13:*** The team will be allowed to set up the required software and hardware for the raspberry pi and server.

## 3.3 Design Requirements

|  |  |  |
| --- | --- | --- |
| Req No. | Obj No. | Requirement |
| 1 | 1,2,3,4 | The design solution shall use a raspberry pi. |
| 2 | 2,3,4 | The design solution shall use a server. |
| 3 | 4 | The design solution shall use a mobile app. |
| 10 | 1 | The raspberry pi shall identify the 10 Death Star images from a set of 100 1024x1024 pixel images in PNG format. |
| 11 | 1 | The rebel server shall crop only the 10 Death Star images to show only the death star weaknesses. |
| 20 | 2 | The raspberry pi and rebel server shall use a wireless signal for data transmission except for Wi-Fi, Bluetooth, or cellular from the raspberry pi to the server. |
| 21 | 2 | The raspberry pi shall transmit data to the server through a physical barrier. |
| 22 | 2 | The server shall be no less than 5 meters away from the physical barrier. |
| 23 | 2 | The raspberry pi shall transmit the data to the server in no more than ten minutes after starting. |
| 24 | 2 | The designed solution shall conform to Federal Communications Commission standards. |
| 30 | 3 | The raspberry pi shall receive confirmation of lossless data transmission from the rebel server. |
| 31 | 3 | The raspberry pi shall retransmit the Death Star weakness in the event of transfer failure. |
| 40 | 3 | The raspberry pi and the rebel server shall encrypt all communication between them. |
| 50 | 4 | The app shall run on a mobile phone operating system. |
| 51 | 4 | The mobile app shall display Death Star weaknesses |
| 60 | 4 | All weaknesses shall be visible in a scrollable grid. |
| 61 | 4 | All images on the app shall display only the weaknesses. |
| 70 | 4 | The rebel server shall host the app. |
| 80 | 1,2,3,4 | The designed solution shall cost no more than 300 USD |

## 3.4 Design Constraints

|  |  |
| --- | --- |
| Const No. | Constraint |
| 10 | The designed solution shall transmit data in no more than 10 minutes. |
| 20 | The designed solution shall be no less than 5m away from the server. |
| 30 | The designed solution shall conform to Federal Communications Commission regulations and standards. |
| 40 | The designed solution shall cost no more than 300 USD. |
| 50 | The app shall run on a mobile device. |
| 60 | The designed solution shall use a wireless signal, except for Wi-fi, Bluetooth, or cellular. |
| 70 | The designed solution shall have no physical connection between the Raspberry pi and rebel server. |
| 80 | The rebel server shall use a Linux operating system. |
| 90 | The designed solution shall have at least one USB port reserved for the thumb drive containing the images. |
| 100 | The Raspberry pi shall transmit encrypted data. |
| 110 | The rebel server shall host the mobile app. |

## 3.5 Design Standards

|  |  |
| --- | --- |
| *Stand. No* | Standard |
| 10 | IEEE1900.1-2019 - Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management |
| 20 | ISO/IEC 24775-1:2021 technology — Storage management. |
| 30 | ISO/IEC 27002:2013- Information security, cybersecurity and privacy protection — Information security controls: This standard ensures the data transmitted between pi and server is encrypted and secure |
| 40 | IEEE C95.7-2005- Radio Frequency Safety Programs, 3 kHz to 300 GHz |
| 50 | IEEE Std 802.15.3-2003 - Information technology Telecommunications and information exchange between systems Local and metropolitan area networks and specific requirements |
| 60 | IEEE 802.15.4-2011 - Low-Rate Wireless Personal Area Networks (LR-WPANs) |
| 70 | ISO/IEC/IEEE 8802-11:2018 -Telecommunications and exchange between information technology systems and requirements for local and metropolitan area networks. |
| 80 | IEEE 10.1109/UBMK52708.2021.9558922 Performance Evaluation of JPEG Standards, WebP and PNG in Terms of Compression Ratio and Time for Lossless Encoding |

20: This standard ensures interoperability between storage systems, such as file transfer protocols between the Raspberry Pi and the Rebel Server.

30: This standard ensures the data transmitted between pi and server is encrypted and secure.

40: This standard defines the safety of radio frequencies and how to prevent use of incorrect frequencies.

50: This standard defines the protocol and compatible interconnection of data and multimedia communication equipment via 2.4 GHz radio transmissions in a Wireless Personal Area Network (WPAN) using low power and multiple modulation formats to support scalable data rates.

60: This standard is commonly used in radio communication systems, such as Zigbee or other short-range wireless communication protocols, which may be applicable if you're using a low-power, short-distance radio setup for data transmission between the Raspberry Pi and Rebel Server.

70: This standard is aimed at providing wireless connectivity for fixed, portable, and moving stations within a local area, offers regulatory bodies a means of standardizing access to one or more frequency bands for the purpose of local area communication (LAN).

## 3.6 Design Functionality

1. *Death Star image recognition*

A raspberry pi will take a thumb drive containing 100 photos and locate the 10 photos of the Death Star.

1. *Wireless data transmission*

A non-detectable wireless signal will send the 10 Death Star photos to the rebel server, and the server will alert the raspberry pi if data is not received. If the data is not received, it will be re-transmitted. All transmissions between the Pi and the server will be encrypted.

1. *Prepare images for display*

The photos now on the server will be decrypted and cropped to display only the Death Star weaknesses within the red circle in each image.

1. *Mobile App*

The cropped photos will be broadcasted to a mobile app. The app will display the weaknesses in a scrollable grid. The app will be hosted on the rebel server.

## 3.7 User Characteristics

The intended users are personnel or organizations operating in high-security or restricted environments where secure, covert communication is essential. These users require a system for transferring sensitive data discreetly over short distances without detection by traditional monitoring systems. Typical users may include security teams, research lab personnel, and intelligence operators who need reliable, low-cost data transmission solutions that avoid Wi-Fi, Bluetooth, or cellular networks, allowing them to securely transfer data without physical connections or detection.

## 3.8 Operating Environment

* Operating System:

The system operates in a Linux-based environment, compatible with both the Raspberry Pi and Rebel Server.

* Hardware:

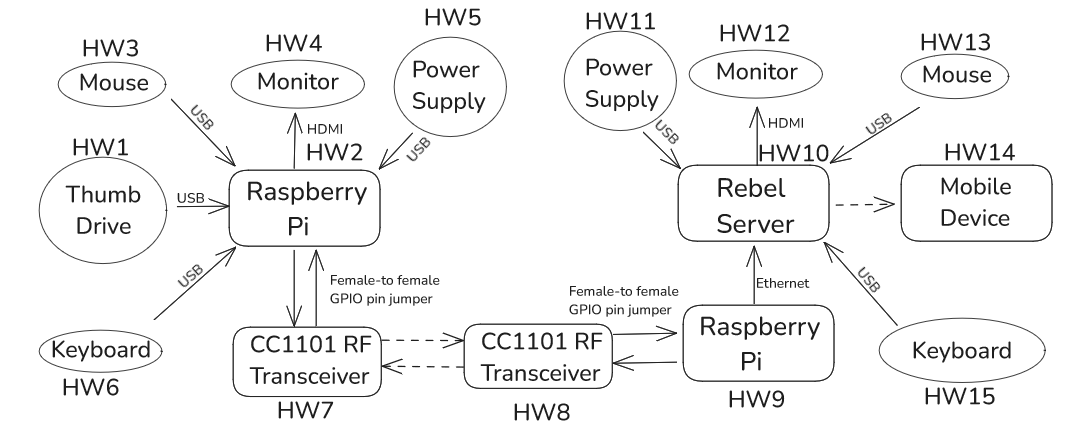
The system operates without interference from other devices using the same frequency the device is transmitting on.

* Environment:

The system is intended to be used in a secure, restricted space, such as a research lab or high-security facility, where traditional wireless communication methods are restricted or monitored.

# 4 Technical Approach

## 4.1 Hardware



HW1: Thumb drive containing the 100 photos that our design processes.

HW2: The Raspberry Pi that runs all the software involved in photo recognition and checksum verification.

HW3: A computer mouse that allows us to interact with the Raspberry Pi.

HW4: A computer monitor that displays the Raspberry Pi.

HW5: A standard wall outlet power supply that powers the Raspberry Pi.

HW6: A computer keyboard that allows us to interact with the Raspberry Pi.

HW7: A transceiver chip that communicates with the Raspberry Pi on the server side.

HW8: A transceiver chip that communicates with the other Raspberry Pi.

HW9: This Raspberry Pi communicates with the other one to allow communication between the Server and HW2.

HW10: The server where the photos are cropped, and the mobile app will be hosted.

HW11: A standard wall outlet power supply that powers the server and the connected Raspberry Pi.

HW12: A computer monitor that displays the server.

HW13: A computer mouse that allows us to interact with the server.

HW14: Any android mobile device that accesses the mobile app.

HW15: A computer keyboard that allows us to interact with the Server.

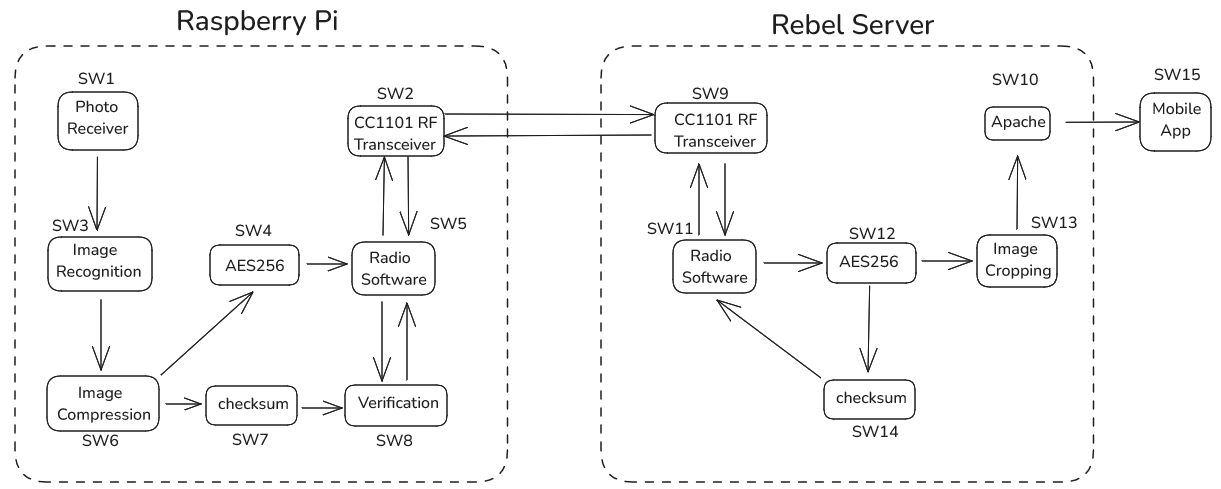
**Hardware Design Choices:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Short range radio Transceivers | Design Options | | | |
| Criteria | Single chip 433/868/915 MHz Transceiver nRF905 | *ALPHA RF Transceiver* | *CC1101 Multi-Frequency RF Transceiver Modules* | *WWZMDiB 2Pcs 433M Transmitter + Receiver Kit for Remote Control Raspberry Pi Arduino* |
| *Bandwidth* | Low | Low | Moderate | Low |
| *Cost* | $9.99 | $10.99 | $8.99 | $6.99 |
| *Frequency Range* | 433/868/915 MHz | 433/868/915 MHz | 433/868/915 MHz | 433 MHz |
| *Accessory* | Antenna | None included | Antenna | Male to Female Jumper Wires included |
| *Compatibility* | Moderate | Low | High | Low |

The CC1101 Multi-Frequency RF Transceiver Modules stand out as the best choice compared to the other options because of its versatility, higher compatibility, and better overall performance. Unlike the nRF905, which is limited to lower data rates and fewer supported frequencies, the CC1101 has multi-frequency support (433, 868, and 915 MHz), making it more adaptable. It also provided moderate bandwidth, allowing for more reliable data transfer over a short distances and in more complex environments. Additionally, the CC1101 has compatibility and source code for Raspberry pi which made it the more reliable choice than the other options. The included antenna support ensured more stable transmission, and its wide frequency range helped with testing different frequency for transmission, unlike the simpler and more limited WWZMDiB or ALPHA RF transceivers, which may not offer the same level of scalability or performance in demanding applications.

Overall, CC1101’s flexibility, reliable performance, and integration made it the most effective and cost-efficient choice for our secure transmission system. The final system achieved reliable encrypted transmission at approximately 2 meters using the CC1101 modules, with a transmission rate optimized for speed and stability.

## 4.2 Software



SW1: This is the built-in software on the Raspberry pi that accessed the 100 PNG images.

SW2: The transceiver chip that transmitted and received data with the rebel server side.

SW3: Software that identified the Death Star images from the images in the provided thumb drive.

SW4: This encrypted the Death Star images before they are transmitted.

SW5: Software on the Raspberry Pi that generated signals for transmission and read signals that were received.

SW6: This compressed the Death Star images after they have been recognized by SW3.

SW7: This generated a checksum value that was used to verify the compressed images being transmitted.

SW8: This compared the checksum values generated by the Raspberry Pi and the server verifying a successful transmission.

SW9: The transceiver chip that transmitted and received data with the Raspberry Pi side.

SW10: Software running on the rebel server that hosted the mobile app.

SW11: Software on the rebel server that generated signals for transmission and read signals that are received.

SW12: This decrypted the Death Star images after they are transmitted.

SW13: This detected, and crop the images down to, the red circles around the Death Star weaknesses in each photo.

SW14: This generated a checksum value for the image the rebel server has received from the Raspberry Pi side for verification.

SW15: This is the mobile app that ran on an android mobile device.

**Software Design Choices:**

**Residual Network**

|  |  |  |  |
| --- | --- | --- | --- |
| Residual Network | Design Options | | |
| Criteria | *TensorFlow* | *Pytorch* | *Microsoft Cognitive Toolkit* |
| Raspberry Pi Support [1] | *Yes* | *Yes* | *No* |
| Active Development | *Yes* | *Yes* | *No* |
| Ram Requirement | *8-16 GB* | *2-8 GB* | *8-16 GB* |

Many implementations of a residual network are fundamentally the same. Compatibility with our hardware restrictions is a more measurable factor in deciding on a machine learning library. Microsoft Cognitive Toolkit is an older library that helped shape machine learning but is no longer actively supported. TensorFlow and Pytorch are more modern and supported libraries. Pytorch is the machine learning library of choice because of its slightly lower ram requirements. It may be feasible to test two different solutions using both TensorFlow and Pytorch, but ultimately it is determined by the capabilities of the hardware.

The final implementation used a predefined residual network (resNet) from Pytorch. This provided an easy-to-use model to train on death star predictions. The final fully connected layer of the model was modified to output 2 possibilities. The model was then trained on Deathstars, and images not related to space as per the assumptions.

**Server and Mobile App**

The receiving raspberry pi would have a module to receive the transmitted images and host them on a file server. The raspberry pi uses Samba to host an SMB fileserver. The mobile app provides the appropriate username and password to connect and pull the images from the server. The images are then displayed in a scrollable grid to the user. The mobile app was developed using Android Studio and tested on Android Mobile OS.

**AES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AES | Design Options | | | |
| Criteria | OpenSSL | Python Library | Java Library | C++ Library |
| Suitability | Requires more resources | Good option for Raspberry Pi due to low resource requirements | Comprehensive library for various encryption algorithms, including AES. | May require optimization for limited resources |
| Installation | sudo apt-get install openssl | pip install pycryptodome | sudo apt install default-jre-headless  sudo apt install openjdk-17-jdk | Install via package manager or build from source. Example: sudo apt-get install libcrypto++-dev |
| Usage Example | Command-line utility for encryption. | from Crypto.Cipher import AES | Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding"); | #include <cryptlib.h> <br>#include <aes.h>  Use classes from the library for AES encryption. |
| License | Open Source (Apache License 2.0) | Open Source (Public Domain) | Open Source (Bouncy Castle License) | Open Source (Boost License) |

During implementation, we selected the Python pycryptodome AES library for its low resource usage, easy integration, and fast setup on the Raspberry Pi platform. Other options such as OpenSSL and Java libraries require more system resources or more complex setup pipelines. C++ libraries like Crypto++ were considered, but the additional effort for Raspberry Pi performance made them less practical for our time and hardware constraints.

The final system used AES in ECB mode, implemented via Python, to encrypt each 48-byte image chunk before transmission. This approach balanced speed, simplicity, and acceptable security for the controlled communication environment.

## 4.3 Budget

A screenshot of a computer

Description automatically generated

# 5 Appendix: Test and Evaluation Master Plan and Report

**Timeline:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Testing and Evaluation Methods:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement | Test Method | Evaluation Method | Threshold | Objective |
| 1 | Raspberry Pi is in use | Observation | The Pi displays powered LED. |  |
| 2 | Server is in use | Observation | Server can boot to a desktop |  |
| 3 | Mobile app is in use | Observation | Mobile app is accessible |  |
| 10.1 | Image recognition Libraries installed | Successful run of python script testing imports | Python script runs with no errors | Python can recognize all installed libraries. |
| 10.2 | Image recognition can run with any image of assumed format | Give software an image of assumed image. | Software identifies the image is of assumed format. | Software can identify all images of assumed format. |
| 10.3 | Image recognition can recognize a single death star image | Give software a Death Star image. | Software identifies the image as a Death Star image. | Software can always determine when an image contains the Death Star. |
| 10.4 | Image recognition can recognize a single non-death star image | Give software a non-Death Star image. | Software identifies the image as a non-Death Star image. | Software can always determine when an image does not contain a Death Star. |
| 10.5 | Image recognition can identify 10 death star images out of 100 images | Give software 100 photos that contains 10 Death star images | Model can recognize 80% of Death Star images | Model can recognize 100% of Death Star images. |
| 11 | Server can crop images | Software will crop any image. | The output is a smaller image | The server will crop images to our desired size. |
| 11.1 | Server can crop images based on a red circle | Software will crop image of assumed format and content | The output is content outlined in a red circle | The script will save multiple cropped images if there are multiple red circles. |
| 20.1 | Transceivers are recognized by Pi and Server | Run software to get transceiver status | Transceiver is recognized by kernel | Pi and Server can recognize the state of their own transceivers |
| 20.2 | Data can be encoded in carrier wave | Script can encode the letter “a” | Script on transmitter says the data was sent. | The software can successfully convert data into a transmittable radio wave. |
| 20.3 | Receivers can decode carrier wave | Receiver collects the sent “a” data | Receiver receives data from the transmitter | The Pi can receive data transmitted from the rebel server and vice versa. |
| 20.4 | Pi and Server can transfer any data | Pi and Server will transmit an image to each other. | A test image is transmitted. | The Raspberry Pi can transmit any image to the rebel server. |
| 20.5 | Pi and Server can transfer an image of assumed format | Transmission script works to transmit a single death star image | Received image is viewable on server | Transmission script can transmit a set of death star images |
| 20.6 | Communication can occur in both pi-to-server and server-to-pi | Observation of transmission output | The same image can be sent back and forth |  |
| 21 | Pi and Server can communicate in separate rooms | Observation | Pi and Server are placed in opposite rooms | Not to be detected by the guards while transmitting the data. |
| 22 | Pi and Server can communicate while separated by more than 5 meters | Observation | At least 5 meters | Transmit over 5-to-10-meter distance |
| 23 | Pi will transfer 1 full image in less than 1 minute | Timing image transmission using a stopwatch | Images take at most 1 minute per image transmission | Images take at most 30 seconds per image transmission |
| 23.1 | Pi will transfer 10 full images continuously. | Timing speed of complete transmission. | The 10 images are sent within 10 minutes. | The complete image transmission takes no more than 10 minutes. |
| 24 | Already tested in standards table |  | The standards table is tested successfully. | The final design is safe to use and operate. |
| 30.1 | Pi and Server download the same checksum package | Observation | Pi and Server use same checksum algorithm |  |
| 30.2 | Server can transmit checksum to Pi | Pi and Server can transmit “hello world” to each other | Hello World, is transmitted | Image checksum is transferred |
| 31.1 | Pi can compare checksums | Observation | Pi saves both Server and Pi checksums | A script determines the confirmation of lossless data transmission from the server |
| 31.2 | Pi can retransmit images | Tested in 20.3 |  | Pi is signaled to retransmit images based on checksums |
| 31.3 | Server can recognize failed images | Observation of resulting images | Server replaces latest image | Server can continue receiving images while computing checksum of previous image. Failed images are retransmitted at the end of transmission. |
| 40.1 | Pi can encrypt data | Encryption library can encrypt a text file | Text file is encrypted | Images are encrypted |
| 40.2 | Server can support the same encryption library | Same test as 40.1 |  |  |
| 50 | Mobile app will launch on Android phone OS | Observation | Mobile app will launch hello world on Android phone OS | Mobile apps are useable on Android and IOS phone systems. |
| 51 | Mobile app will display images | Observation | Mobile app displays hello world | Mobile app displays images |
| 60 | Mobile app will display images in a grid | Observation | Mobile app displays rectangular components in a grid | Mobile app displays images in a grid |
| 61 | Mobile app displays cropped images from server | Observation | Mobile app displays text from server input | Mobile app displays Death Star images from the server |
| 70 | Server hosts all mobile app content | Tested in 61 | Mobile app content is dependent on server | The server will display the content for the mobile app |
| 80 | Tested in Constraint 40 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Constraint | Test Method | Evaluation Method | Threshold | Objective |
| 10 | Already tested in Requirement 23 |  | 10 minutes | Data transmission shall take less than 10 minutes. |
| 20 | Already tested in Requirement 22 |  | 5 meters | Data can be transmitted across a short distance. |
| 30 | Already tested in standards table |  | The standards table is tested successfully. | The final design is safe to use and operate. |
| 40 | Record expenses in the budget. | Observation | $300 | The design cost less than $300 |
| 50 | Already tested in Requirement 50 |  | App on mobile phone | App displays 10 images |
| 60 | Already tested in Requirement 20 |  | A discrete signal is used | Data is transmitted using RF transceiver |
| 70 | See if the Pi has a physical connection to the server. | Observation | 0 physical connections | There are no physical connections between Pi and server. |
| 80 | Check the OS of the device | Observation | Linux OS in use | The Pi and server use a Linux OS |
| 90 | Check available USB ports on the Pi | Observation | 1 open USB port | The Pi will have at least 1 USB port open. |
| 100 | Already tested in Requirement 40 |  | All data between pi and server is encrypted. | All data between pi and server is encrypted. |
| 110 | Already tested in Requirement 70 |  | The app is hosted on the server | The app is hosted on the server. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Standard | Test Method | Evaluation Method | Threshold | Objective |
| 10 | Document review to align with IEEE1900.1-2019 terms | Glossary check for terminology compliance | Complete use of IEEE-compliant terms | Consistency in terminology across all project documentation |
| 20 | Validate data storage and file transfer between the Raspberry Pi and the Rebel Server. | Perform file transfer tests to verify interoperability and data storage. | Successful data transfer between systems | Smooth, error-free file transfers without interoperability issues. |
| 30 | Review and verify encryption protocols on the Raspberry Pi and Rebel Server. | Confirm that all data transmissions are securely encrypted per ISO/IEC 27002:2013. | Data encryption that prevents unauthorized access. | Complete encryption compliance, ensuring robust data security |
| 40 | Use transmission devices in standard range of operations. | Check that the device is operating in the standard frequency range shown on the data sheet. | All frequencies fall within the safe 3 kHz to 300 GHz range. | Full compliance with RF safety guidelines, ensuring no unsafe frequencies are used. |
| 50 | Test RF Transceiver configurations to validate WPAN data transmission at 2.4 GHz. | Ensure RF Transceiver transmits data using low power and supported modulation formats. | Functioning WPAN within the 2.4 GHz range. | High-quality WPAN transmission with low interference and stable data rates. |
| 60 | Conduct tests on the RF Transceiver system to ensure it meets LR-WPAN transmission standards. | Use low-power, short-range communication protocols to test data transfer stability. | Reliable, low-power transmission with no loss of data. | Consistent, high-quality LR-WPAN communication across the required distance. |
| 70 | Assess the RF Transceiver for compatibility with ISO/IEC/IEEE 8802-11:2018 to verify local and metro connectivity standards. | Check for transmission compatibility over multiple network bands. | RF Transceiver operates within local and metro connectivity requirements. | Reliable and standardized network communication across various distances. |
| 80 | Test JPEG, WebP, and PNG image formats on the Raspberry Pi to analyze compression and decompression time. | Measure transmission time and compression ratio for each image format. | Compression achieves fast, lossless image transmission. | Optimal balance of high compression ratio and low transmission time. |

**Risk and Mitigations:**

* Risk of signal interference during transmission. The likelihood is moderate, and severity is high. Mitigation: Test RF transceiver at multiple frequencies to identify which is the best option for stable transmission. Schedule and conduct transmission testing in the environment to identify and mitigate potential interferences
* Risk of data transmission delays due to encryption processing. The likelihood is moderate, and severity is moderate. Mitigation: optimize encryption code to reduce processing time and test alternative algorithms to balance security with speed. Compress images to lessen the data load to make sure there is faster processing on the Raspberry Pi.
* Risk of slow data transmission due to data size. The likelihood is moderate, and the severity is high. Mitigation: compress images to a smaller size for faster data transmission.
* Risk of incorrect image identification from USB. The likelihood is moderate, severity is high. Mitigation: test image recognition software extensively on varied datasets before doing final transmission. Use redundancy checks to make sure selected images are part of the target set type and prevent errors in the identification process.
* Risk of signal attenuation due to physical barrier. The likelihood is high, and severity is moderate. Mitigation: Adjust RF transceiver signal strength settings within a safe limit to compensate for attenuation. Select transmission frequencies correctly for passing through the glass and conduct testing to find the correct positioning and power adjustments for good transmission.
* Risk of Data Synchronization Issues Between Server and Mobile App. The likelihood is moderate, and severity is high. Mitigation: Include logs for tracking data exchange between the app and the server to debug and resolve inconsistencies quickly.
* Risk of Unauthorized data access during communication. Th likelihood is moderate, and the severity is high. Mitigation: Implement end-to-end AES encryption to secure all transmitted data and conduct regular security audits of both the raspberry pi and the rebel server.
* Risk of data interception during transmission. The likelihood is moderate, and the severity is high. Mitigation: use secure frequency channels for RF transmission and enable device authentication protocols. Conduct RF penetration testing to identify vulnerabilities.
* Risk of prolonged troubleshooting for RF communication setup. The likelihood is moderate, and the severity is moderate. Mitigation: Monitor progress against milestones weekly and adjust resources or tasks to prevent delays and associated costs.
* Risk of installation errors delaying software setup. The likelihood is moderate, and the severity is moderate. Mitigation: Create a step-by-step installation guide for drivers and libraries, and test installation procedures beforehand to identify and address potential errors.
* Risk of exceeding the $300 budget due to unforeseen hardware replacements. The likelihood is low, and the severity is high. Mitigation: Purchase additional components if necessary, use affordable alternatives, and maintain a contingency fund within the budget.
* Risk of increased costs due to extended project timeline. The likelihood is moderate, and the severity is moderate. Mitigation: Monitor progress against milestones weekly and adjust resources or tasks to prevent delays and associated costs.