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

# Architectural Design Specification
# CSE 4316: Senior Design I
# Fall 2022

## Team Mercury
## ARGOOSE-Counter Drone Tracking

James Grumbles
Nirdesh Sakh
Augustine Nguyen
Sanyogita Piya
Mahin Roddur

## Revision History

| Revision | Date | Author(s) | Description |
|---|---|---|---|
| 0.1 | 10.31.2022 | AN | first draft |
| 0.2 | 11.13.2022 | AN, JG, SP, NS, MR | complete draft |

# Contents

## List of Figures



## List of Tables

# 1 INTRODUCTION

ARGOOSE is a system that performs drone detection via sensors. Users of ARGOOSE will be able to set up a perimeter detection zone and receive information on detected drones within the area of operation. The ARGOOSE system detects drones and sends back information on detected drone to users via web interface. System is designed to perform in an open space with acoustic detection. The system is designed to address general drone safety concerns. By providing feedback on drones located in the area, customers can be informed of drone behavior around sensitive locations. General customers are defined as anyone in need of this system's capabilities.

## 2 System Overview

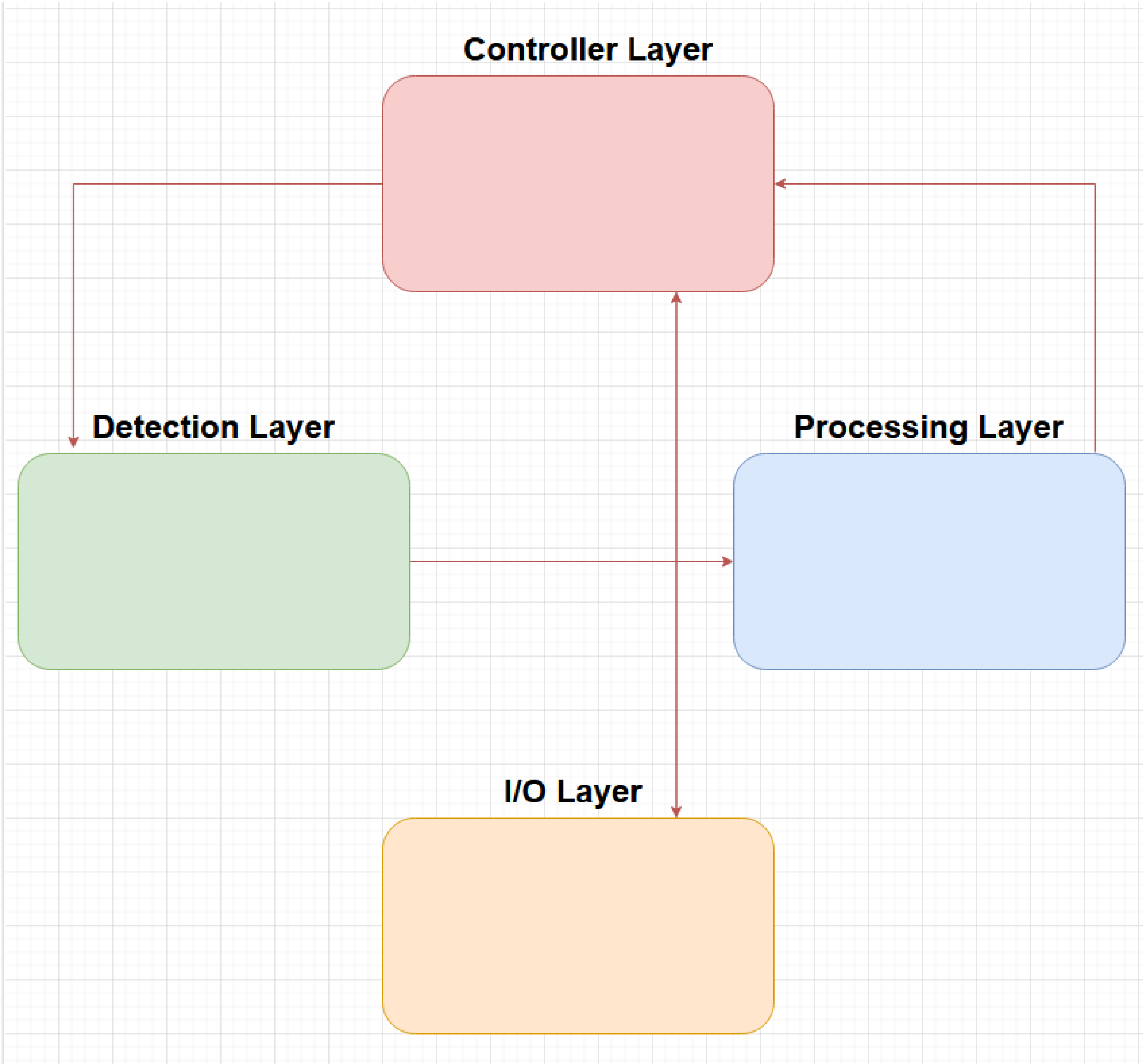


Figure 1: A simple architectural layer diagram

### 2.1 Layer Controller Description

The controller layer's main purpose is to control and regulate the entirety of the system. The main interface is the raspberry pi; it communicates to sensors in the detection layer on when to activate, as well as takes information from the processing layer and sends that information to the drone database subsystem. Additionally, the raspberry pi takes input from users and adjusts settings on the system.

### 2.2 Layer Detection Description

The detection layer consists of sensor subsystems: acoustic arrays and a secondary detector. These sensors take in real-world-analog data and transfers that data over to the processing layer to be digitized and processed further. Also, the secondary detector subsystem is pinged by the controller layer.

## 2.3 Layer Processing Description

Within the processing layer, the analog to digital converter (ADC) is the internal interface, of which it digitizes analog data and sends it to the ODAS subsystem as well as the secondary processor subsystem. Finally, the processed data from the ODAS and secondary subsystems are sent to the controller layer, into the raspberry pi subsystem.

## 2.4 Layer I/O Description

The Input/Output layer's main purpose is to provide the interface to the user to graphically view the detected drones and select individual detected drones to track actively. The layer communicates both ways with the controller layer to receive the data of the detected drones and display it and also send the input from the user.

# 3 Subsystem Definitions & Data Flow

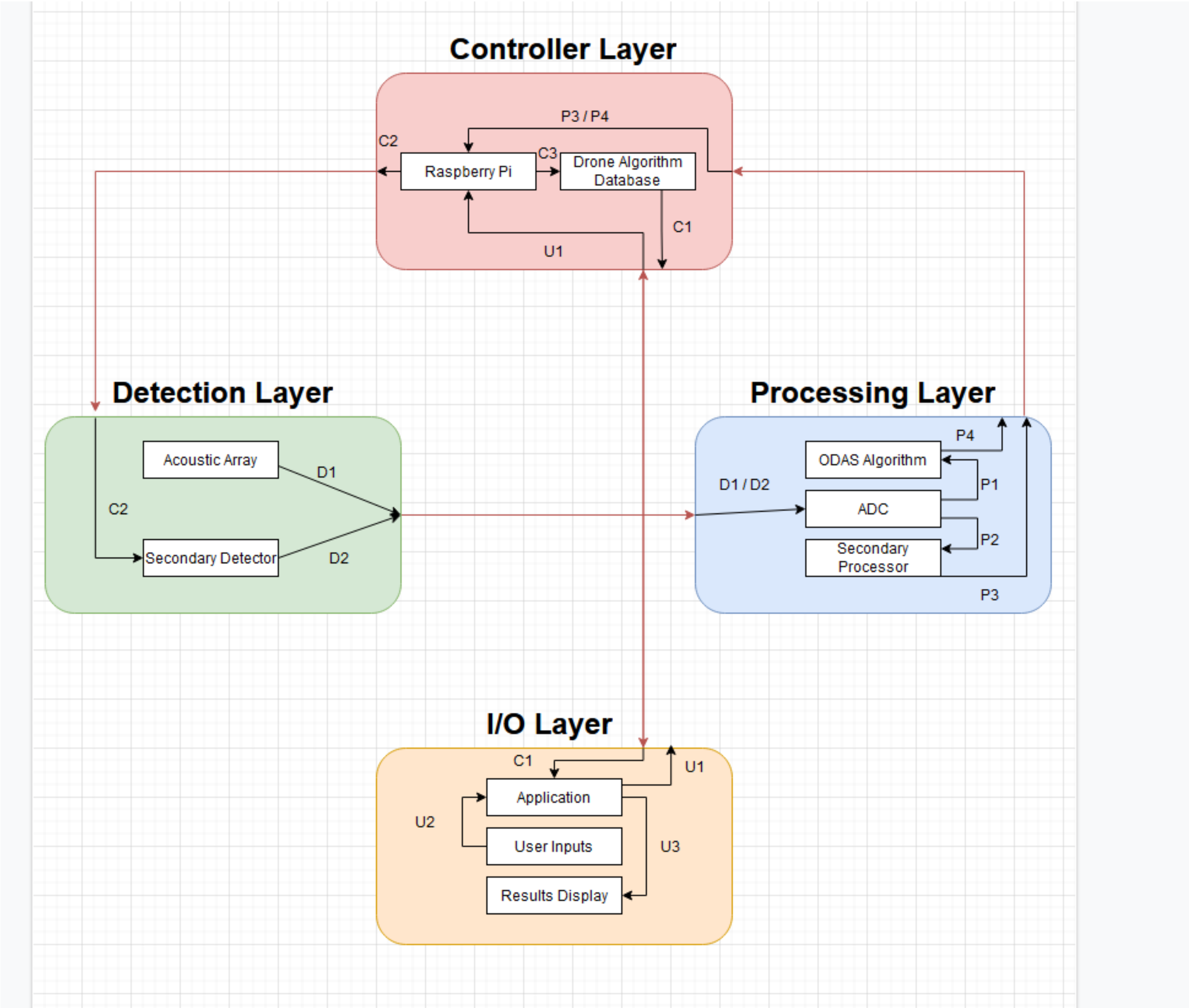


Figure 2: A simple data flow diagram

# 4 Detection Layer Subsystems

The detection layer is the interface between the real-world and the system. Its subsystems consists of sensors and these sensors take in data such as audio, and then is sent to the processing layer. It is also controlled by the raspberry pi subsystem in the controller layer.

## 4.1 Acoustic Array

The acoustic array will consist of a hexagonal printed circuit board with six microphones located at each point of the hexagon. Internally the board will contain the audio to digital converter and be capable of processing the microphone inputs from the array and forwarding that information to a digital component for processing. These requirements are meant by a COTS device called ReSpeaker, which will be the component used to meet these requirements. The subsystem will detect audio inputs from the environment and forward the raw analog audio data to the ADC for processing.

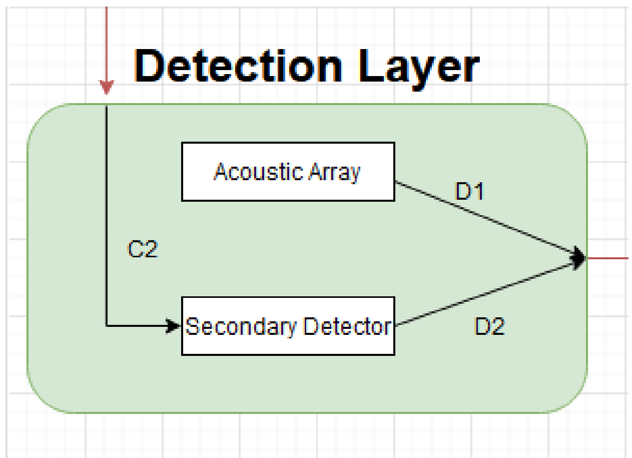


Figure 3: Acoustic Array

### 4.1.1 Assumptions

The ReSpeaker will detect across all six microphones and information will be compiled and properly forwarded to the ADC

### 4.1.2 Responsibilities

The subsystems sole purpose is to process raw audio data from the surrounding environment and compile it into a usable data stream. Each microphone works in tandem with the others to generate positional data, then forwards this information to the ADC to process and push further along the pipeline.

### 4.1.3 Subsystem Interfaces

Table 2: Subsystem interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #RS01 | ReSpeaker Microphone Array | Microphone 1<br>Microphone 2<br>Microphone 3<br>Microphone 4<br>Microphone 5<br>Microphone 6 | Compiled Audio Data to ADC |
| #RS02 | Power Supply | Power In | N/A |

## 4.2 Secondary Detector

The secondary detector is instructed to activate by the raspberry pi when certain conditions are met; once it is activated, the data collected is sent to the ADC for analog to digital conversion.

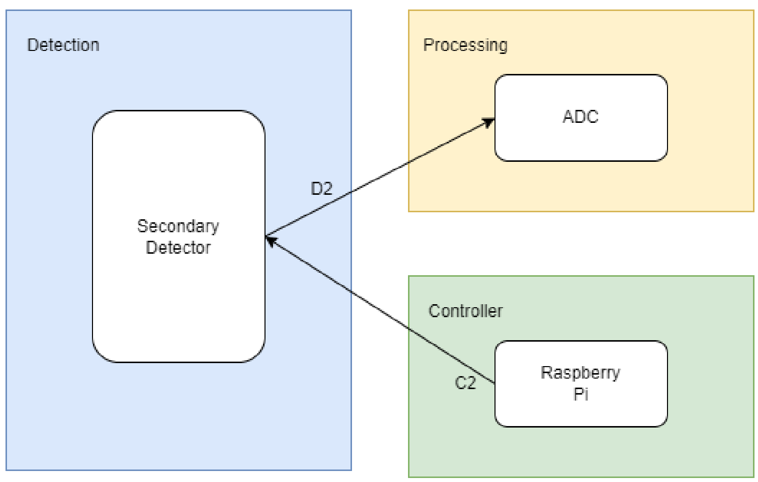


Figure 4: describes secondary detector's relation with the rest of the system

### 4.2.1 Assumptions

The secondary detector takes in some sort of sensory detail. As of yet, this detail has not been determined 100% yet; however, it will most likely be either some form of frequency detector or a visual detector. If the secondary detector is to report to ODAS, then the sensory detail would be auditory.

### 4.2.2 Responsibilities

The main responsibility of the secondary detector is to be an extra method of detection, adding to the precision and accuracy. It is on the standby of the raspberry pi and is processed the same way as the acoustic array.

### 4.2.3 SUBSYSTEM INTERFACES

Table 3: Subsystem interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #C2 | Activation of Secondary Detector | Raspberry pi signaling activation | Secondary Detector activated |
| #D2 | Secondary Detector Data Sending | Analog data collected | Data is sent to the ADC for digital conversion |

# 5 Processing Layer Subsystems

The processing layer takes in analog data, digitizes it, and then processes it. This processed data is then sent to the controller layer.

## 5.1 Analog to Digital Converter (ADC)

The subsystem ADC's main purpose is to convert real-world-analog data and convert it to digital information, ready to be processed. The data that gets fed to ADC comes from the detection layer, with its acoustic array and secondary detector subsystems. Once the ADC has received and converted the data, it transfers the information within its very own processing layer, to the ODAS and secondary processor subsystems.

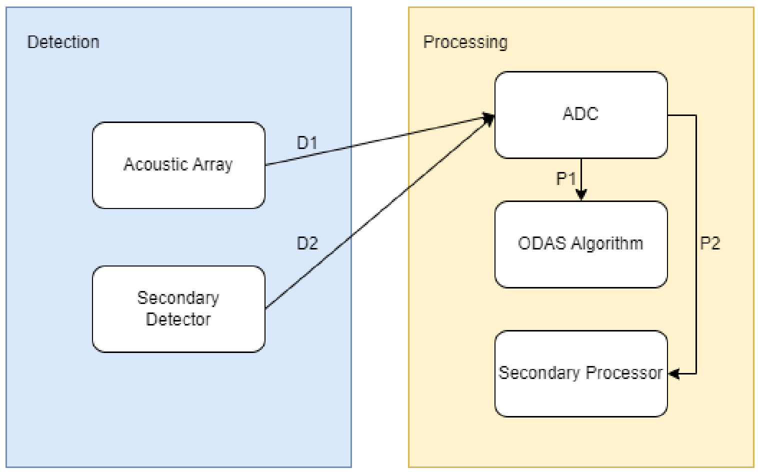


Figure 5: describes the relation of ADC along with other subsystems and layers

### 5.1.1 Assumptions

The data collected from analog sensors is understood by the ADC and is able to be converted, and the converted digital data is able to be interpreted and processed by the ODAS algorithm and the secondary processor.

### 5.1.2 Responsibilities

The ADC subsystem is responsible for connecting the real world with the digital system of the product. The detection layer wouldn't be able to produce any results without its data being processable, and the other subsystems in the processing layer wouldn't have any data to process.

### 5.1.3 Subsystem Interfaces

Table 4: Subsystem interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #D1/D2 | Analog Data Input | Acoustic Data<br>Secondary Detector Data | Digitized Data |
| #P1 | Sending to ODAS | N/A | Digitized Data |
| #P2 | Sending to Secondary Processor | N/A | Digitized Data |

## 5.2 ODAS Algorithm

ODAS is an open source software developed for taking in multiple audio data sources and compiling it into usable data streams. For the purposes of the ARGOOSE system, ODAS will utilize a modified sourcing code to locate the sound of drones that are detected within the area of operation. At default, ODAS detects raw sounds and locations based off of the ReSpeaker array. This information is compiled and displayed on the software to pinpoint location and event data. Data streams will come in from the ADC on detected sound patterns, will be identified and manipulated by ODAS and then forwarded on to the Raspberry pi controller to display and interpret relevant data.

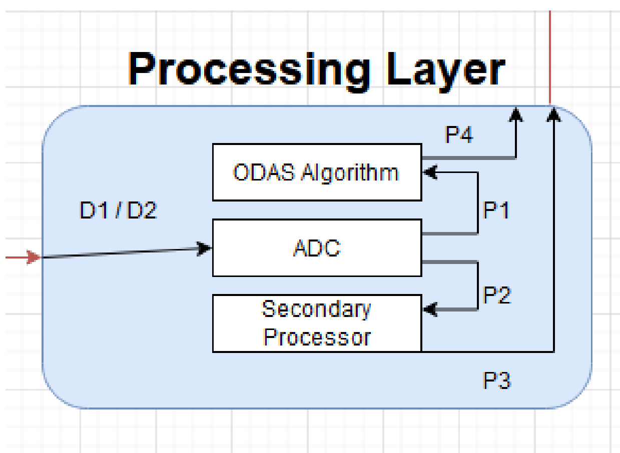


Figure 6: Example subsystem description diagram

### 5.2.1 Assumptions

ODAS algorithmic data processing will accurately detail drone detection and forward data streams that are relevant and interpretable by the Raspberry Pi. Information received from the ADC will be in a format that is accurately analyzable by the ODAS algorithm.

### 5.2.2 Responsibilities

ODAS is responsible for generating the positional and occurrence data from the raw data stream of the ADC. It is the key component is generating the sample necessary for the controller to determine when and where drone detections should occur.

### 5.2.3 Subsystem Interfaces

Table 5: Subsystem interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #RS01 | ODAS Algorithm | ADC Data Stream | Compiled Detection Data to Controller |

## 5.3 Secondary Processing Subsystem

The secondary processing subsystem is responsible for activation of the secondary sensors and detection system after the object is suspected to be a drone.

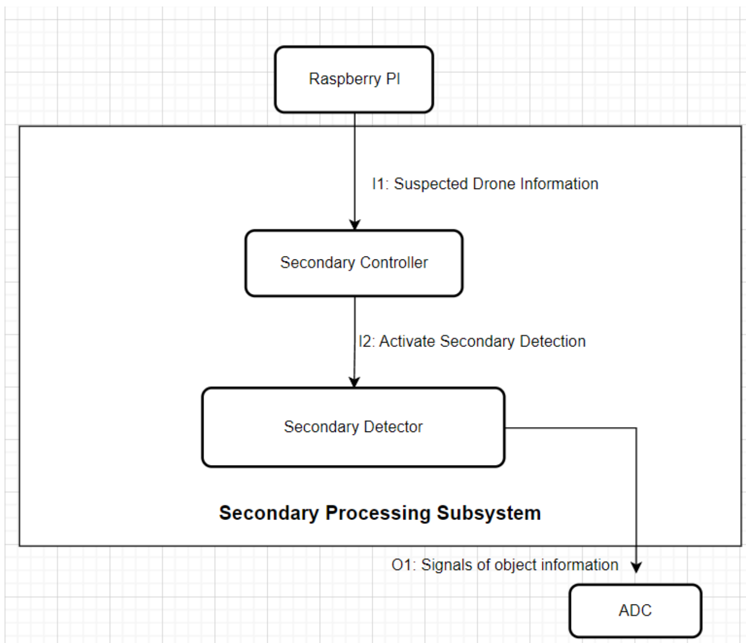


Figure 7: Secondary Processing Subsystem

### 5.3.1 Assumptions

This subsystem assumes that it has an object that is potentially a drone and this information is given by the raspberry pi.

### 5.3.2 Responsibilities

The primary responsibility of this subsystem is to confirm that the object is a drone. It activates the secondary detection sensors which sends the signals to ADC and these signals are sent through ODAS algorithm to the raspberry pi.

### 5.3.3 Subsystem Interfaces

Table 6: Subsystem interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #I1 | Secondary Controller | Drone Information | Signal to secondary detector |
| #I2 | Secondary detector | Activate secondary sensor | N/A |

# 6 Controller Layer Subsystems

The controller layer holds the raspberry pi and drone database subsystem. The raspberry pi controls much of the system's features, as well as acting as an interface between the different layers. The drone database holds all the processed data and sends the information to users.

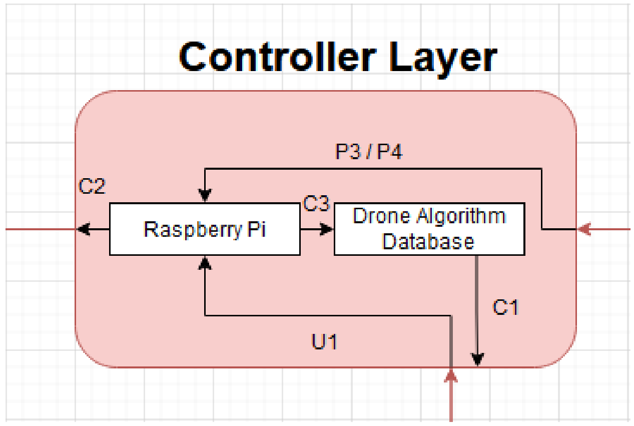


Figure 8: Controller subsystems diagram

## 6.1 Raspberry Pi

In this system, the Raspberry Pi acts as the central hub that controls the traffic of data flow from one subsystem to another. It also activates functions of specific subsystems based on feedback from another subsystem.

### 6.1.1 Assumptions

The Raspberry Pi is assumed to be powered on throughout the active run time of the system. In addition to that, the Pi needs to have a persistent forms of connections with the other subsystems.

### 6.1.2 Responsibilities

Here is an overview of the data flow pathways that Pi is involved in:

- After receiving a positive signal of drone detection from ODAS (which is in the Processing Subsystem), the Pi activates the secondary detector in the Detection Subsystem.
- After receiving confirmation of drone detection from Secondary Processor, it relays relevant drone data to the Drone Algorithm Database.
- The Pi also has a bi-directional connection with I/O Subsystems, particularly with the Application subsystem. It is the point of contact in terms of setting updates and user-logins for the user of the system.

### 6.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here.

Table 7: Raspberry Pi Interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #P4 and C2 | Activation of secondary detector | Signal from ODAS confirming drone presence (P4) | Activation of secondary sensor in Detection subsystems (C2) |
| #P3 and C3 | Relaying drone information to database | Signal from secondary processor (P2) | Drone data relayed to Drone Algorithm Database (C3) |
| #U1 | User interactions relayed by Raspberry Pi | User inputs from Application | Feedback after processing user inputs back to Application |

## 6.2 Drone Algorithm Database

The drone algorithm database stores and manages captured drone data, so that it can be accessed by the application and displayed in a readable format. Logically, it makes sense to have a closer proximity to the controller layer, since most its interactions with other layers and subsystems would be overseen by the Raspberry Pi (the controller of the system).

### 6.2.1 Assumptions

The database is assumed to have a persistent connection in the system's network and the internet. It should be free of any major security issues when data is loaded or accessed by any other subsystem.

### 6.2.2 Responsibilities

Here is an overview of the database's functionality, other than managing the stored data:

- Whenever a new update in drone detection is confirmed by the Pi, it relays relevant information on the detected drone to the database. This is the primary interface through which the database receives updates.
- The other interface link is between the database and the application subsystem in the I/O layer. The application accesses the database through this link to retrieve drone data that it eventually displays to its user.

### 6.2.3 Subsystem Interfaces

Table 8: Drone Algorithm Database interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #C1 | Access link between application and database | Access request by application | Fulfillment of access request |
| #C3 | Reception of new drone data | Information relayed by Pi | Updated database |

# 7 Input Output Layer Subsystems

The input output layer acts an interface between the user and the system. It displays data that is held by the drone database, as well as makes queries to the system.

## 7.1 Application Sub-System

The application subsystem is the Graphical User Interface that interacts with the user to get user inputs if necessary and sends it to the database in form of a query.

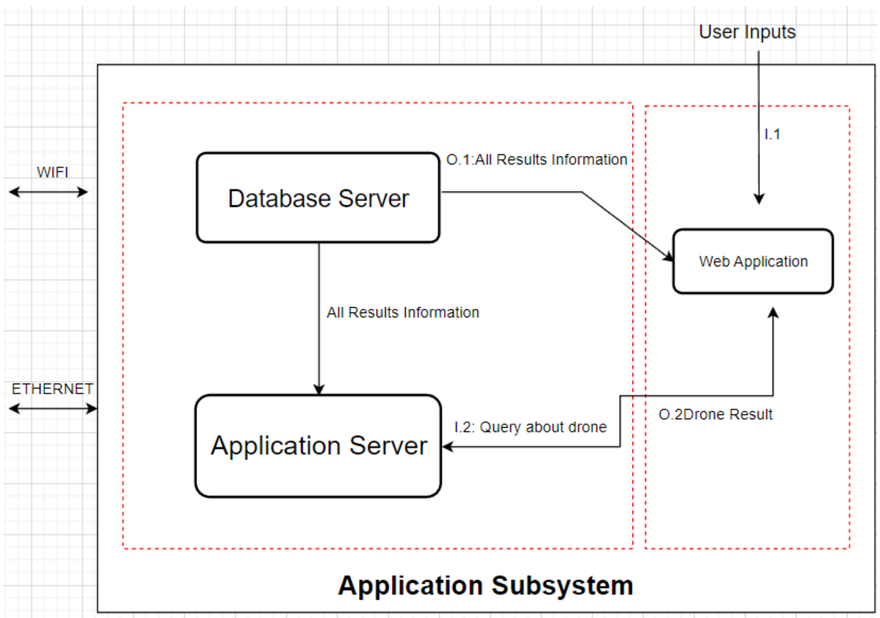


Figure 9: Data flow of Application Subsystem

### 7.1.1 Assumptions

For the operation of this subsystem, it is assumed that the user has access to WIFI or some form of internet and a browser to launch the web application.

### 7.1.2 Responsibilities

This subsystem is responsible for the database interaction with the user. First of all, the data of all the detected drone is displayed to the user then if the user requests for more information on a particular drone then the request is forwarded to the database in form of a query and the database results are displayed.

### 7.1.3 Subsystem Interfaces

Table 9: Subsystem interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #I.1 | User Inputs | Click on particular drone title | Query to the database about the drone |
| #I.2 | Query about drone | The query to the database | Results according to the query |
| #O.1 | List of objects | N/A | List of drone and non drone objects |

## 7.2 Results Page

The results page includes the output of the drones detected in the area which is presented to the user in graphical form.

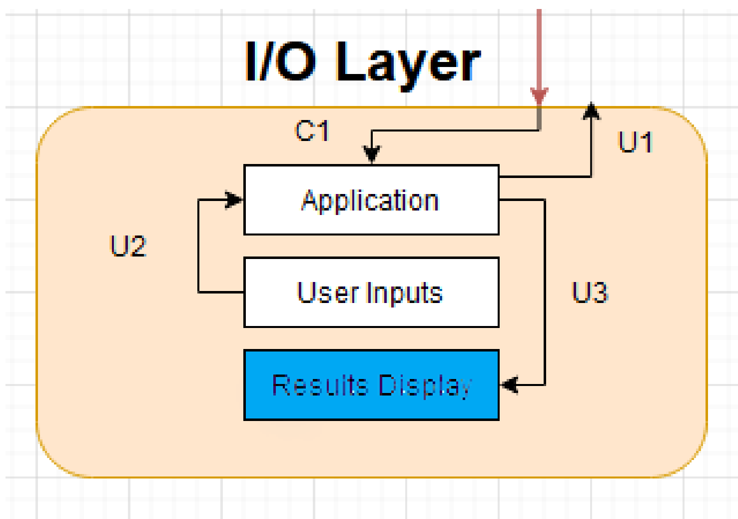


Figure 10: Results Display Subsystem Diagram

### 7.2.1 Assumptions

The results page will only be online after the system is turned on.

### 7.2.2 Responsibilities

The responsibility of the results page is to communicate with the database through the application interface to provide the graphical visualization of the location of the system and any drones detected in the surrounding area. The results page will display the location of the device in the form of circular dot and any detected drones will be shown in triangular shape.

### 7.2.3 Results Subsystem Interfaces

Table 10: Results interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #U3 | Communication with application | detected drones and system location data from the database | Graphical view of the detected drones and location of the system |

## 7.3 User Inputs

The User Inputs page includes the input that can be taken from the user. Here, the user can select a detected drone and verify that it is a false positive. Also, user can select an individual drone and the system will actively track the selected drone in as close to real time as possible.

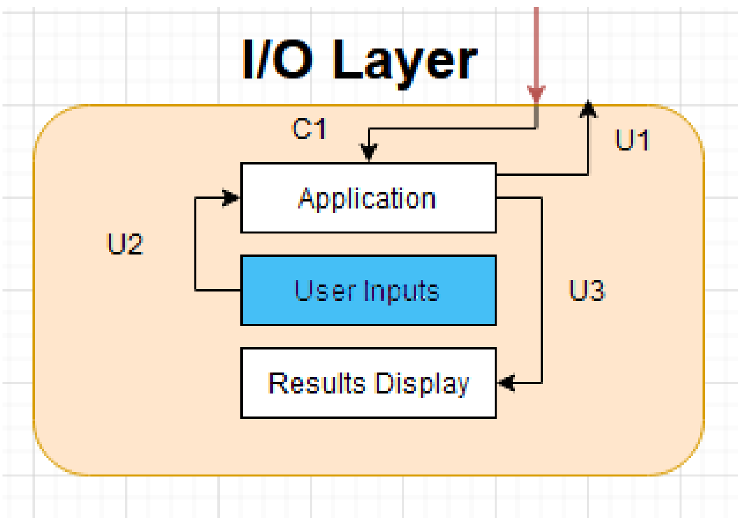


Figure 11: User Inputs subsystem diagram

### 7.3.1 Assumptions

The user can only select one drone at a time for active detection.

### 7.3.2 Responsibilities

The responsibility of the User Inputs is to provide a user interface to the user where they can select a detected drone and verify that it is a false positive. Also, the user can select an individual drone and the system will actively track the selected drone in as close to real time as possible.

### 7.3.3 User Inputs Subsystem Interfaces

Table 11: User Inputs Subsystem interfaces

| ID | Description | Inputs | Outputs |
|---|---|---|---|
| #U201 | Single Drone Active Tracking | Select input from the user | Selected Drone highlighted and actively tracked |
| #U202 | False Positive in detected Drones | Select input from the user | The selected object will not be further tracked |

# REFERENCES