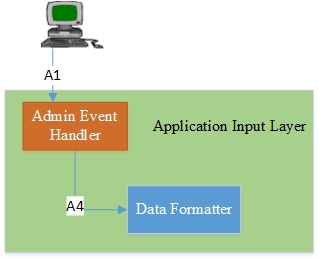
# Application Input Layer

This section will present detail description of the Application Input Layer. This layer takes the input from the two types of the users namely the Administrator (the admin) and the local users (the user) via the Web Application and the Android™ application. Each subsystem of this layer will be presented below.

## Administrator Event Handler Subsystem



**Fig. 6 -** Administrator Event Handler

### General Description:

The Admin Event Handler accepts inputs from the administrator through the web application. This subsystem will determine the type of messages needed to be delivered to the rest of the system in order to satisfy the request or an action received from the administrator.

### Assumptions:

All administrative interactions will be unique which will allow the Admin Event Handler to properly respond to the event.

### Responsibilities:

The Admin Event Handler will detect the type of action that has occurred through the admin user interface of the application. The Admin Event Handler will trigger a response message and sent it to the Input Data Formatter. The response will be unique to a particular type of actions.

### Inter-Layer Interfaces:

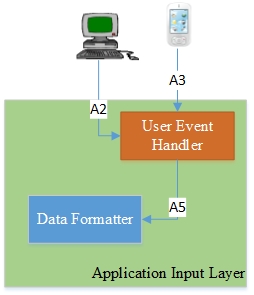
This subsystem does not interact with other layers.

### Public Interfaces:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **Information required** | **Information Returned** |
| adminEventListener | The Admin Event Handler will wait to receive messages from the administrator through the web application. | None | Admin Input Data (New user information, login credentials, etc.) |

**Table. 6 –** Public Interface (Admin Event Handler)

## User Event Handler Subsystem



**Fig. 7 -** User Event Handler

### General Description:

The User Event Handler accepts inputs from the user through the Web Application and Android™ application. This subsystem determines the type of messages needed to be delivered to the rest of the system in order to satisfy any request or action received from the user.

### Assumptions:

All user interactions will be unique which will allow the User Event Handler to properly respond to the event.

### Responsibilities:

The User Event Handler will detect the type of action that has occurred through the user interface of the application. The User Event Handler will trigger a response message and send it to the Data Formatter. The response will be unique to a particular type of actions.

### Inter-Layer Interfaces:

This subsystem does not interact with other layers.

### Public Interfaces:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **Information required** | **Information Returned** |
| userEventListener | The User Event Handler will wait to receive messages from the user through the web application or Android™ Application. | None | User Input Data (login credentials) |

**Table. 7 –** Public Interface (User Event Handler)

## https://lh6.googleusercontent.com/xr2D5lHweQLgB2Q0UEwFirlmL3cEBwXOuo7Ficgcv9EOZ2pLIsN5peQAi_FOlcokRDPbUaJt7oMiwJMJP0ZIsrHVSarj4gWeGO1qNQKXd5oDmucEFImxkFHphkz-WkpV8gData formatter

**Fig. 8 -** Data Formatter

### General Description

The Data Formatter accepts input from the Admin Event Handler and the User Event Handler. The data will be pre-processed by the formatter and converted into the format which is acceptable for the rest of the system. Then, the formatted data will be sent to the Application Data Controller which is necessary for creating a proper response for both Admin and the User’s interaction through the internet service.

### Assumptions

The Data Formatter must be able to accept various forms of input from the Admin Event Handler and the User Event Handler.

### Responsibilities

The Data Formatter should accept the digital inputs in various forms and translate it into an acceptable form. The formatter will structure each data in such a manner that they will be easy to process and send it to the processing layer (Application Data Controller).

### Inter-Layer Interfaces:

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Description** | **Information required** | **Information Returned** |
| sendFormattedData | The Data Formatter will send formatted data to the Application Data Controller in the Processing Layer through the internet. | Event data generated by the Android™ Application and the Web Application. | None |

**Table. 8 –** Inter-Layer Interface (Data Formatter)

### Public Interfaces:

This subsystem uses the internet to pass its data to the Processing Layer.

## Data Flow Definitions

The following table will show the breakdown of the data flows in the architectural data flow diagram presented in section 9.1. This table will show each of the data flows, what the data is, and where the data is going. The table will also show the information that is passed between each of the subsystems so that it can give a better understanding of the system and not only the layers of the system.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data**  **Element ID** | **Description** | **Source** | **Sink** |
| **A1** | Admin enters any input into the Web application or requests functionality from the application. | Admin | Admin Event Handler |
| **A2** | User enters any input into the web application or requests functionality from the application. | User | User Event Handler |
| **A3** | User enters any input into the Android™ application or requests functionality from the application | User | User Event Handler |
| **A4** | An event from the admin is processed and sent to the Data Formatter to format the request into information usable by another subsystem. | Admin Event Handler | Data Formatter |
| **A5** | An event from the user is processed and sent to the Data Formatter to format the request into information usable by another subsystem. | User Event Handler | Data Formatter |
| **A6.1** | After the message has been formatted it is passed using the internet. | Data Formatter | Internet |
| **A6.2** | The Internet passes the information it received from Data formatter to the Application Data Controller. | Internet | Application Data Controller |
| **D1** | The fully processed information has will be passed to the Database Manager in order to get ready to be stored. | Main Processor | Database Manager |
| **D2** | The formatted data will be passed to the Database (i.e. scan disk, hard drive) which will store the information. | Database Manager | Database |
| **D3** | The needed stored data is requested from the Database (i.e. scan disk, hard drive) and is formatted into data that is readable by the system. | Database | Database Manager |
| **D4** | When stored data is requested it is passed from the Database Manager to the Main Processor. | Database Manager | Main Processor |
| **H1** | The Hardware Event Handler will listen the input in the form of a button press and then pass the appropriate information. | Controller | Hardware Event Handler |
| **H2** | The Hardware Event Handler will get signal from battery about its power. | Battery | Hardware Event Handler |
| **H3** | The Hardware Event Handler will get information about drone movement from the Drone (Pilot Module). | Drone(Pilot Module) | Hardware Event Handler |
| **H4** | The Video Input Handler will get video frame as input from the video camera. | Video Camera | Video Input Handler |
| **H5** | The hardware event that has been occurred is transferred to the Radio Frequency Handler either process the data or pass it to the system for high level processing. | Hardware Event Handler | Radio Frequency  Handler |
| **H6** | The video event that has been occurred is transferred to the Radio Frequency Handler which will pass it to the Hardware Data Controller. | Video Input Handler | Radio Frequency  Handler |
| **H7** | The processing is done to the command it received from the controller and passed to the drone. | Radio Frequency  Handler | Drone |
| **H8** | The video event and hardware event that has been occurred is relayed to the Hardware Data Controller for action to be taken by the system. | Radio Frequency  Handler | Hardware Data Controller |
| **O1** | The output data is passed to the Output Formatter so that it can be put in the correct format. | Output Controller | Output Formatter |
| **O2.1** | The formatted data is passed to the internet. | Output Formatter | Internet |
| **O2.2** | The internet will act like a bridge and pass the output data to the admin so that they can view the output information, either an android app or the web application. | Internet | Admin |
| **O2.3** | The internet will act like a bridge and pass the output data to the user so that they can view the output information, either an android app or the web application. | Internet | User |
| **P1** | Once hardware data has been received and analyzed, it is passed to the Main Processor for performing the requested action. | Hardware Input Controller | Main Processor |
| **P2** | Once the application data has been received and analyzed, it is passed to the Main Processor for performing the requested action. | Application Data Controller | Main Processor |
| **P3** | The processed information will be passed to the Output Controller so that it can be output. | Main Processor | Output Controller |

## Producer-Consumer Relationship Table

The following table will show the relationships between data flows in a different way and helps to give a different perspective from producer consumer relationships.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Consumer | Admin | User | Drone | Batteries | Video Camera | Controller | Hardware Event Handler | Video Input Handler | Radio Frequency Handler | Admin Event Handler | User Event Handler | Data Formatter | Hardware Data Controller | Application Data Controller | Main Processor | Database Manager | Hard Disk | Output Controller | Output Formatter | Internet |
| Producer |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Admin |  |  |  |  |  |  |  |  |  |  | A1 |  |  |  |  |  |  |  |  |  |  |
| User |  |  |  |  |  |  |  |  |  |  |  | A2 A3 |  |  |  |  |  |  |  |  |  |
| Drone |  |  |  |  |  |  |  | H3 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Batteries |  |  |  |  |  |  |  | H2 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Video Camera |  |  |  |  |  |  |  |  | H4 |  |  |  |  |  |  |  |  |  |  |  |  |
| Controller |  |  |  |  |  |  |  | H1 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Hardware Event Handler |  |  |  |  |  |  |  |  |  | H5 |  |  |  |  |  |  |  |  |  |  |  |
| Video Input Handler |  |  |  |  |  |  |  |  |  | H6 |  |  |  |  |  |  |  |  |  |  |  |
| Radio Frequency Handler |  |  |  | H7 |  |  |  |  |  |  |  |  |  | H8 |  |  |  |  |  |  |  |
| Admin Event Handler |  |  |  |  |  |  |  |  |  |  |  |  | A4 |  |  |  |  |  |  |  |  |
| User Event Handler |  |  |  |  |  |  |  |  |  |  |  |  | A5 |  |  |  |  |  |  |  |  |
| Data Formatter |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | A6.1 |
| Hardware Data Controller |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | P1 |  |  |  |  |  |
| Application Data Controller |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | P2 |  |  |  |  |  |
| Main Processor |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | D1 |  | P3 |  |  |
| Database Manager |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | D4 |  | D2 |  |  |  |
| Hard Disk |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | D3 |  |  |  |  |
| Output Controller |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | O1 |  |
| Output Formatter |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | O2.1 |
| Internet |  | O2.2 | O2.3 |  |  |  |  |  |  |  |  |  |  |  | A6.2 |  |  |  |  |  |  |

# Testing Considerations

This section will describe how each layers of the system will be validated. The each layer has specified functions and properties which will be performed without compromising the stand-alone principle of those layers and their independency will not be broken. As a development team we will be comparing each layer to the guiding principles used to create the layers to ensure they are consistent and represent the principles stated in the document above.

## Application Input Layer

### Modularity:

The Application Input Layer must be an independent layer which doesn’t depend upon the internal mechanisms of other layers. The layer must be a standalone layer of the system.

### Internal-System Interaction:

The subsystems of this layer will only interact and communicate with the Processing Layer of the architecture.

### External-System Interaction:

The subsystems of this layer will be directly interactive with the user and administrator. The administrator must be able to set user settings, add new local users, and set other local settings of the system.

## Hardware Input Layer

### Modularity:

The Hardware Input Layer must be an independent layer which does not depend upon the internal mechanisms of other layers. The layer must be a standalone layer of the system.

### Internal-System Interaction:

The subsystems of this layer will interact and communicate with the Processing Layer of the architecture.

### External-System Interaction:

The subsystems of this layer will be gathering data from the system components like drone battery, controller, video camera, and drone. This layer will gather data from the system components like battery power of the drone, information about controller, video, and other data from the pilot module. It will also transmit the information to the drone about its flight situation like it is in autopilot or human override.

## Processing Layer

### Modularity:

The Processing Layer must be an independent layer which does not depend upon the internal mechanisms of other layers. The layer must be a standalone layer of the system.

### Internal-System Interaction:

The subsystems of this layer will receive data from the Application Input Layer, the Database Layer, and the Hardware Input Layer. Similarly, the subsystems of this layer will only send data to the Presentation Layer and Database Layer of the system.

### External-System Interaction:

The subsystems of this layer will not interact with any external resources except internet through which it receives the data send by user and admin. The data is passed to it by Data Formatter subsystem of the Application Input Layer which uses internet to pass data to this layer.

## Database Layer

### Modularity:

The Database Layer must be an independent layer which does not depend upon the internal mechanisms of other layers. The layer must be a standalone layer of the system.

### Internal-System Interaction:

The subsystems of this layer will only send and receive data from the Processing Layer.

### External-System Interaction:

There will be a Hard Drive of the host computer which will act as the data storage of the system. It will be the only external interaction this layer will have.

## Presentation Layer

### Modularity:

The Presentation Layer must be an independent layer which does not depend upon the internal mechanisms of other layers. The layer must be a standalone layer of the system.

### Internal-System Interaction:

The subsystems of this layer will interact with the Processing Layer from which it will receive data. This layer will not receive and send data to any other layer of the system.

### External-System Interaction:

The subsystem of this layer will send the processed data as well as packaged data to the corresponding output to the admin and user through the internet.