Interface Control Document Systems
Competition Phase 2

Revision 2
May 2025

Defense Advanced Research Projects Agency
Biological Technologies Office
675 North Randolph Street
Arlington, VA 22203-2114
TriageChallenge@darpa.mil

Contents

1 Introduction

This document describes the interface to the DARPA Command Post server where teams competing in the DARPA Triage Challenge will submit their Casualty Reports during the competition. The intent is to convey the overall concept of operations for interaction with the Command Post during competition and also to describe the hardware and software interfaces necessary to successfully interact with the server. This document covers the Command Post interface for the Systems Competition only. For Data competition, please refer to its respective ICD document.

2 Overview and Concept of Operations

- The DARPA Command Post provides the sole interface by which teams will transmit competition data to DARPA for purposes of scoring, and, as such, is a critical element for competition participation. To facilitate ease of integration between each competitor and DARPA, the Command Post interface is designed to use widely accepted Internet standards, helping to ensure that teams will have plenty of prior experience building a compliant system and a wide variety of off-the-shelf libraries to choose from to help them do so.
- The interface is utilized by transmitting data formatted in JavaScript Object Notation (JSON) using the Hypertext Transfer Protocol (HTTP) over a local Ethernet network between the competitor client and the DARPA Command Post server. The various functions of the interface are exposed as separate Uniform Resource Identifiers (URIs) that HTTP requests are executed against. Teams will authenticate their messages using an access token that will be provided during registration and setup.
- Additionally, DARPA intends to observe and share motion video of Base Station displays acquired via HDMI capture devices and cameras located in the Staging Area.
- To facilitate deriving operational insights from these interactions, DARPA requests that individual UxSs (unmanned system) are identified as appropriate in report submissions. This is to be accomplished through the use of a “system” field on relevant endpoints as described below. The "system" field will be specified for each UxS during system registration in advance of each competition. It must be unique within the team and no more than 10 characters (alphanumeric and underscores). Each UxS's "name" must remain consistent across all relevant API endpoints for the entire competition event.

2.1 Scoring

During the competition, a team’s score will be based solely on the Casualty Reports that the team submits to the DARPA Command Post via the scoring interface.

When a team is confident that they have identified and localized a new casualty, they will submit a POST request (a standard HTTP protocol operation) with a structure that describes the type and location of the casualty to the appropriate URI for score reports. After receiving and processing the report, the server will return an HTTP response that contains a confirmation of the submitted report and an indication of its effect on the team's remaining reports of that type. If there is a problem with the submitted request, for example an authentication token is not provided or the structure is malformed, a verbose error will be returned instead of an updated score. All properly formatted and authenticated casualty reports are recorded. Fields within

the HTTP response body are used to indicate when a valid report has no effect on the team's score (see POST /api/initial_report). For example, casualty reports made after the official run time has ended or in excess of the allotted report quantity have no effect on the score; this will be indicated by dedicated fields within the HTTP response body.

A separate URI will provide an interface for teams to request their current score, remaining casualty reports, and other information about the current run.

2.2 Base Station Display Capture

In addition to cameras in the Staging Area, DARPA intends to observe and capture the Base Station display(s) via HDMI screen capture devices. The captured footage is anticipated to be used as part of the live-stream broadcast. DARPA intends to provide HDMI cabling in the Base Station Area to capture up to two displays supporting the following options:

- 720p (1280 x 720) max 60 hz
- 1080p/i (1920 x 1080) max 60 hz
- UHD 4k (3840 x 2160) max 60 hz

This will require the Base Station to be capable of mirroring and/or extending its display to available HDMI ports. The capture devices will be capable of inline operation allowing teams to make use of secondary displays.

Verification of this functionality will be incorporated into the team/system on-site checkout procedures.

3 Physical and Network Interface

The DARPA Command Post physical and network interfaces are built on the traditional foundation of Ethernet communications, and so the general concepts should be familiar to all teams. We discuss here the specifics that teams will need to consider for successful interaction with the DARPA Command Post.

- ***Physical Interface:*** All network communications will take place over an Ethernet connection carried by twisted-pair cabling, of sufficient bandwidth for casualty updates, between the DARPA Command Post and the team's Base Station. (IEEE 802.3; twisted-pair cable, e.g., IEEE 803.3ab)

  DARPA will provide one (1) clearly marked Ethernet cable with an RJ-45 connector in the team Base Station area that will be the sole connection to the Command Post. Teams must connect this cable into a dedicated ethernet adapter on the Operator Control Station. Additional interfaces are allowed, however, this interface will be the only interface allowed to communicate with the Command Post.

- ***Network interface:*** Interactions with the Command Post will be handled by a network endpoint on a DARPA-controlled server. The endpoint will have a static IP version 4 address, and teams will be responsible for ensuring that their specific network configuration can successfully communicate

with the network endpoint. Depending on a team's network configuration, this may require a team to, for example, bind a network address to their designated network interface and/or add additional routes to the IP version 4 routing table on their system in order to allow communications with the DARPA Command Post network endpoint.

In the Base Station area during setup, the following will be printed and posted:

- the static IP version 4 address and TCP/UDP port information of the DARPA Command Post network endpoint and
- the IP version 4 configuration information that teams must use to configure their equipment to communicate with the DARPA Command Post network endpoint.

Teams' internal networks should avoid using IP version 4 subnets and addresses that may overlap or conflict with the following subnets reserved for use by the DARPA network:

- 10.0.0.0/15 (10.0.0.0 – 10.1.255.255)
- 10.192.0.0/10 (10.192.0.0 – 10.255.255.255)

The address and port information of the network endpoint will not change during a run; however, it is possible that DARPA may change the address and/or port information between runs.

Network data rate for team interactions may be limited to approximately 100 Mbps, full-duplex between the Base Station area and the DARPA Command Post network endpoint. Submitting data sizes near or over this limit may result in delays.

Network communication between the team Base Station area and the DARPA Command Post will be separate from all other network traffic on-site to ensure security.

3.1 DARPA provided WIFI

DARPA will provide a 5 GHz Wi-Fi network to each team that will provide connectivity with the team's wired OCS connection. The Wi-Fi network will be Wi-Fi 6 (802.11ax), and it is strongly recommended that teams use only Wi-Fi 6 (802.11ax) wireless network adapters to ensure maximum performance.

Each team will have a separate wireless network reserved exclusively for that team's use. Each wireless network will be provided in the respective team garage of each team when that team is not operating on a course. When operating on a course, a team's wireless network will be made available on the course assigned to the team. Consequently, teams should not need to change their wireless network configuration when moving between team garages and courses.

There are some possible caveats when moving a team's wireless network between the team's garage and a course. Stale network state can result, so teams should be prepared to test for and resolve such issues upon each transition between team garages and courses. This is generally a simple matter of clearing the ARP tables on network devices after moving to the new location. ARP tables can be cleared by issuing an appropriate OS command, disabling and reenabling affected wireless NICs, or rebooting affected network stations.

The proper functioning of network communication can be tested using ICMP ECHO (ping) or similar network communications.

4 Protocol Interface

The DARPA Command Post protocols will be built on the common HTTP/1.1[3], and so, the general concepts should be familiar to all teams. HTTP/1.1 utilizes the stream-oriented Transmission Control Protocol[4] (TCP) that handles the underlying mechanisms of forming and using socket connections. HTTP/1.1 allows requests to be targeted at different Uniform Resource Identifiers[5] (URIs) that loosely describe the functionality that is being requested, and the Command Post uses different URIs to allow multiple functions to be provided at the same network endpoint. These URIs are described in detailed protocol descriptions of Sections 5.

Teams must include an authentication token with each HTTP packet that they send in order to act as an additional mechanism for ensuring that submissions were intended (particularly scoring submissions). The "Authorization" header field must include a "bearer" token[6] that will be provided by during system checkout procedures. For example, if the team's token was "flux230{showroom", then they must include an "Authorization" header field with value "Bearer flux230{showroom". Please note that the DARPA Command Post will deviate from standards and will **not** provide a "WWW-Authenticate" response header with error information in the case of an authentication failure.

In an HTTP response, the HTTP status code will be used to indicate the success or failure of the request, and all HTTP response content will be in JavaScript Object Notation[7] (JSON), a self-describing format for representing structured data. Accordingly, the "Content-Type" header field will always be "application/json". Depending on the interface being used, this content could be either:

(1) a JSON-encoded string (i.e., in quotation marks) when there is a human-readable message accompanying this response (such as an error message)
(2) a JSON-encoded object (i.e., key-value pairs enclosed in braces) that contains detailed return information from an interaction and whose content will depend on the interaction

The applicable HTTP status codes are described below in Section 4.1.

4.1 HTTP Status Codes

The HTTP response status codes that will be returned by the Command Post, along with their meaning, are given below:

Code	*Description*	*Meaning*
200	OK	Request was accepted and/or response will be valid.
201	Created	Request was properly formatted and resource (i.e., casualty report) was created.
400	Bad request	JSON parsing failed.
401	Unauthorized	Authentication token does not match expected token for this run (or was not provided).
404	Not Found	URI is not recognized
413	Request Entity Too Large	Upload file size was too large
422	Unprocessable Entity	JSON request had incorrect information for this interaction type.
429	Too many requests	Too many requests have been submitted in a given amount of time.

```
HTTP/1.1 422 Unprocessable Entity
Content-Length: 23
Content-Type: application/json

“Missing field ‘type’”
```

5 Authentication

In order to provide additional competition integrity and protections against unintended operations, DARPA will require an authorized user token assigned to the appropriate team during interactions with the Command Post. This token will be utilized throughout the competition event. These user tokens will be generated during system checkout at each competition event. A default username and password will be provided to teams upon arrival at each competition event during team orientation. This username and password will be updated during system checkout with the DARPA Command Post server. After updating the default password, an authentication token will be provided and expected to be used for all API interactions during the competition event.

6 Scoring Interaction Protocol

All scoring interactions will take place using the network specified in Section 3 and the HTTP protocol described in Section 4. The specific URIs and JSON content related to scoring interactions are defined here. Each function is described in its subsections below with the HTTP function and corresponding URI as section name.

The majority of interactions should use one of the available “POST” functions (Sections 6.2-4) at infrequent times to submit casualty report information for identified casualties. Teams should avoid overusing API requests in order to avoid delays in response times. Note: since the response to the POST functions includes updated status information, there is no need to immediately follow “POST” submissions with GET /api/status calls.

The following API functions are available are described in detail in following subsections:

- GET **/api/status** – Provides current run information at given time of request

- POST **/api/initial_report** – Accepts initial assessment for newly identified casualties or initial supplement.
- POST **/api/update_report** – Accepts updated assessment for previously identified casualties

A test server that implements the scoring API will be provided ahead of the workshop to verify system interactions.

- POST **/api/casualty_image** - Submit an image of a previously identified casualty

6.1 GET /api/status

```
{
    "clock": <float>,
    "team": <string>,
    "user": <string>
}
```

Definitions for the return values of the status response are as follow:

- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *team* represents the unique identifier of the team requesting the status
- *user* represents the authenticated user based on the authentication token provided

Example request:

```
GET /api/status HTTP/1.1
Authorization: Bearer flux230{showroom
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 77
Content-Type: application/json

{ “clock”: 721.4, "team": "dtc1", user: “John B” }
```

6.2 POST /api/initial_report

Submit a report for a newly identified casualty or submit an initial supplement with previously unreported information on an identified casualty.

The request content must be a JSON-encoded object with the following format:

```
{
    "casualty_id": <int>,
    "team": <string>,
    "system": <string>,
    "location":
    {
        "latitude": <float>,
```

```
        "longitude": <float>,
        "time_ago": <int>
    },
    "severe_hemorrhage": <value_at_time>,
    "respiratory_distress": <value_at_time>,
    "hr": <value_at_time>,
    "rr": <value_at_time>,
    "temp": <value_at_time>,
    "trauma_head":  <int>,
    "trauma_torso": <int>,
    "trauma_lower_ext": <int>,
    "trauma_upper_ext": <int>,
    "alertness_ocular": <value_at_time>,
    "alertness_verbal": <value_at_time>,
    "alertness_motor": <value_at_time>
}
```

where **value_at_time** is a sub-object with the following format:

```
{
    "value": <int>,
    "time_ago": <int>
}
```

For fields with timestamps, *time_ago* is a non-negative integer representing the elapsed time (in seconds) since the estimate was made. This relative timestamp is used to determine what ground truth to score the reported value against. *time_ago* may differ between fields in the same report, and it will only pertain to the field it is contained within. For example, *time_ago* within the *location* field will only be used to determine the time at which to compare reported casualty location to the ground truth location.

Each field has the following definitions:

- *casualty_id* is a new, unique integer identifier for the casualty whose condition is being reported
- *team* is a unique string identifier for the team submitting the report
- *system* is a unique string identifier for the unmanned system responsible for the identification of the casualty
- *location* is the coordinates of the reported casualty expressed as *latitude* and *longitude* at time *time_ago* relative to the time the report was submitted.
- *severe_hemorrhage* indicates whether the casualty shows signs of severe hemorrhage or not at time *time_ago* relative to the time the report was submitted.
- *respiratory_distress* indicates whether the casualty shows signs of respiratory distress or not at time *time_ago* relative to the time the report was submitted.
- *hr* indicates the heart rate in beats per minute for the reported casualty at time *time_ago* relative to the time the report was submitted.
- *rr* indicates the respiration rate in breaths per minute for the reported casualty at time *time_ago* relative to the time the report was submitted.

- *temp* indicates the core temperature in degrees Fahrenheit for the reported casualty at time *time_ago* relative to the time the report was submitted.
- *trauma_head* indicates presence of injury on the head for the reported casualty.
- *trauma_torso* indicates presence of injury on the torso for the reported casualty.
- *trauma_lower_ext* indicates the presence and type of injury on one or both lower extremities for the reported casualty.
- *trauma_upper_ext* indicates the presence and type of injury on one or both upper extremities for the reported casualty.
- *alertness_ocular* indicates the presence and type of alertness based on eye movement at time *time_ago* relative to the time the report was submitted.
- *alertness_verbal* indicates the presence and type of alertness based on speech and vocalizations at time *time_ago* relative to the time the report was submitted.
- *alertness_motor* indicates the presence and type of alertness based on gross motor movement at time *time_ago* relative to the time the report was submitted.

The following fields are required in the first POST with a new *casualty_id*:

- *casualty_id*
- *team*
- *system*
- *location*

The following fields are required for any subsequent POST with a previously provided *casualty_id*:

- *casualty_id*
- *team*
- *system*

Remaining health assessment fields are optional. Any included fields must contain all sub-fields; for example, *alertness_verbal* must contain both *value* and *time_ago* sub-fields.

Each field takes specific acceptable values, with semantics for each casualty report field as follows:

- *location*
  - *latitude*: float-precision number with range from -90 and 90 in degrees
  - *longitude*: float-precision number with range from -180 and 180 in degrees
  - *time_ago*: non-negative integer in seconds
- *severe_hemorrhage*
  - *value: 1* indicates the casualty shows signs of the severe hemorrhage
  - *value: 0* indicates the casualty does not show signs of severe hemorrhage
  - *time_ago*: non-negative integer in seconds
- *respiratory_distress*
  - *value: 1* indicates the casualty shows signs of the respiratory distress
  - *value: 0* indicates the casualty does not show signs of respiratory distress
  - *time_ago*: non-negative integer in seconds
- *hr*

    - *value*: non-negative integer indicating heart rate in beats per minute
    - *time_ago*: non-negative integer in seconds
- *rr*
    - *value*: non-negative integer indicating respiratory rate in breaths per minute
    - *time_ago*: non-negative integer in seconds
- *temp*
    - *value*: non-negative integer indicating core temperature in degrees Fahrenheit
    - *time_ago*: non-negative integer in seconds
- *trauma_head*
    - *0 (normal)* indicates absence of injury on the head
    - *1 (wound)* indicates presence of one or more injuries on the head
    - *2 (not testable)* indicates assessment of head injury was not possible
- *trauma_torso*
    - *0 (normal)* indicates absence of injury on the torso
    - *1 (wound)* indicates presence of one or more injuries on the torso
    - *2 (not testable)* indicates assessment of torso injury was not possible
- *trauma_upper_ext*
    - *0 (normal)* indicates absence of injury on both upper extremities
    - *1 (wound)* indicates presence of non-amputation injury on one or both upper extremities
    - *2 (amputation)* indicates amputation of one or both upper extremities
    - *3 (not testable)* indicates assessment of upper extremity injury was not possible
- *trauma_lower_ext*
    - *0 (normal)* indicates absence of injury on both lower extremities
    - *1 (wound)* indicates presence of non-amputation injury on one or both lower extremities
    - *2 (amputation)* indicates amputation of one or both lower extremities
    - *3 (not testable)* indicates assessment of lower extremity injury was not possible
- *alertness_ocular*
    - *value: 0 (open)* indicates eyelids are open and/or blinking spontaneously
    - *value: 1 (closed)* indicates eyelids are closed
    - *value: 2 (not testable)* indicates eyelid assessment was not possible
    - *time_ago*: non-negative integer in seconds
- *alertness_verbal*
    - *value: 0 (normal)* indicates responsiveness to speech prompts with coherent and relevant speech
    - *value: 1 (abnormal)* indicates confused or pain-/distress-related speech and/or vocalizations
    - *value: 2 (absent)* indicates absence of vocalization
    - *value: 3 (not testable)* indicates speech and vocalization assessment was not possible
    - *time_ago*: non-negative integer in seconds
- *alertness_movement*
    - *value: 0 (normal)* indicates walking/standing/sitting unsupported with coordinated movement
    - *value: 1 (abnormal)* indicates lying or sitting supported with minimal limb movement
    - *value: 2 (absent)* indicates absence of limb movement

- *value: 3 (not testable)* indicates movement assessment was not possible
- *time_ago*: non-negative integer in seconds

Complete definitions of report fields and minimum requirements for scoring can be found in the DARPA Triage Challenge Competition Rules.

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "user": <string>,
    "system": <string>,
    "clock": <float>,
    "report_id": <string>,
    "report_timestamp": <string>,
    "report_status": <string>,
    "casualty_id": <integer>
}
```

The semantics of the response fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *user* represents the authenticated user that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *report_id* is a unique identifier for the recorded report response
- *report_timestamp* represents the absolute time that the report was received, following the ISO 8601 combined date and time format. This timestamp is used to award bonus points according to the official scoring guidelines in the DARPA Triage Challenge Competition Rules
- *report_status* indicates the status of the casualty report.
    - "*accepted*" indicates the report was accepted to be evaluated and one allotted report was used
    - "*admin stop*" indicates this report was not scored because it was submitted during an administrative stop
    - "*run not started*" indicates this report was not scored because it was submitted prior to run start
    - "*duplicate id*" indicates this report was not scored because it contained a duplicate *casualty_id* from a previously received report during the same run
    - "*report limit exceeded*" indicates this report was not scored because there were no remaining reports available during the run
    - "*time limit exceeded*" indicates that this report was not scored because it was submitted after the end of the run

- *casualty_id* confirms the casualty identifier in the submitted report

6.3 POST /api/update_report

Submit a report updating the condition of a previously identified casualty.

The request content must be a JSON-encoded object with the following format:

```
{
    "casualty_id": <int>,
    "team": <string>,
    "system": <string>,
    "location":
    {
        "latitude": <float>,
        "longitude": <float>,
        "time_ago": <int>
    },
    "severe_hemorrhage": < value_at_time >,
    "respiratory_distress": < value_at_time >,
    "hr": <value_at_time>,
    "rr": <value_at_time>,
    "alertness_ocular": <value_at_time>,
    "alertness_verbal": <value_at_time>,
    "alertness_motor": <value_at_time>
}
```

The semantics of the fields follow the standards set in Section 6.2 with the following exception:

- *casualty_id* is a previously reported integer identifier used to associate this update report with the initial report on the same casualty

The following fields are required:

- *casualty_id*
- *team*
- *system*
- *location*

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "user": <string>,
    "system": <string>,
    "clock": <float>,
    "report_id": <string>,
    "report_timestamp": <string>,
    "report_status": <string>,
```

```
    "casualty_id": <integer>
}
```

The semantics of the fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *user* represents the authenticated user that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *report_id* is a unique identifier for the recorded report response
- *report_timestamp* represents the absolute time that the report was received, following the ISO 8601 combined date and time format. This timestamp is used to award bonus points according to the official scoring guidelines in the DARPA Triage Challenge Competition Rules
- *report_status* indicates the status of the casualty report.
    - "*accepted*" indicates the report was accepted to be evaluated and one allotted report was used
    - "*admin stop*" indicates this report was not scored because it was submitted during an administrative stop
    - "*run not started*" indicates this report was not scored because it was submitted prior to run start
    - "*no initial report*" indicates this report was not scored because an initial report with the same *casualty_id* has not yet been received during the run
    - "*report limit exceeded*" indicates this report was not scored because there were no remaining reports available during the run
    - "*time limit exceeded*" indicates that this report was not scored because it was submitted after the end of the run
- *casualty_id* confirms the casualty identifier in the submitted report

6.4 POST /api/casualty_image

Submit an image of a previously identified casualty.

The request content must be multipart/form-data with the data object in following format:

```
{
    "casualty_id": <int>,
    "team": <string>,
    "system": <string>,
    "time_ago": <int>
}
```

And with an attached file object with content type of image/png or image/jpeg.

The semantics of the fields are as follow:

- *casualty_id* is a previously reported integer identifier used to associate this update report with the initial report on the same casualty
- *team* is a unique string identifier for the team submitting the report
- *system* is a unique string identifier for the unmanned system responsible for the image
- *time_ago* is a non-negative integer with seconds elapsed since image capture.

The image should be a PNG or JPEG image with file size not exceeding 5MB. In addition, images will only be accepted once per second (rate of 1Hz).

If the image size exceeds 5MB, a HTTP error code of 413 will be returned.
If the request endpoint has submitted at too high of a rate, a HTTP error code of 429 will be returned.

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "user": <string>,
    "system": <string>,
    "clock": <float>,
    "report_id": <string>,
    "report_timestamp": <string>,
    "report_status": <string>,
    "casualty_id": <integer>
}
```

The semantics of the fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *user* represents the authenticated user that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *report_id* is a unique identifier for the recorded report response
- *report_timestamp* represents the absolute time that the report was received, following the ISO 8601 combined date and time format. This timestamp is used to award bonus points according to the official scoring guidelines in the DARPA Triage Challenge Competition Rules
- *report_status* indicates the status of the casualty report.
    - "*accepted*" indicates the report was accepted to be evaluated and one allotted report was used
    - "*admin stop*" indicates this report was not scored because it was submitted during an administrative stop

    - "*run not started*" indicates this report was not scored because it was submitted prior to run start
    - "*no initial report*" indicates this report was not scored because an initial report with the same *casualty_id* has not yet been received during the run
    - "*time limit exceeded*" indicates that this report was not scored because it was submitted after the end of the run
    - "*size limit exceeded*" indicates that the image size exceeds the limit
    - "*rate limit exceeded*" indicates that the rate of image submissions exceeds the limit
- *casualty_id* confirms the casualty identifier in the submitted report

7 Appendix A – Example image submission

Instances of localhost must be changed to actual server IP. Auth_token is valid only for the test submission server. A different value will be provided during test events.

Example Command Line Curl Call

```
curl -X 'POST' \
  'http://localhost/api/casualty_image' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIi-
OiI4M2Q3OGM4ZS04MzhhLTQ0NzctOWM3Yi02N2VmMTZlNWY3MTYiLCJpI-
jowfQ.i4KuwEtc5_6oIYz5TDWcdzl5bMkvCpLZTSZG2Avy84w' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@test.jpeg;type=image/jpeg' \
  -F 'casualty_id=1' \
  -F 'team=TEST_TEAM' \
  -F 'system=UGV_1' \
  -F 'time_ago=0'
```

Example Python Script

```
import requests

server = 'localhost'
url = f'http://{server}/api/casualty_image'
auth_token = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIi-
OiI4M2Q3OGM4ZS04MzhhLTQ0NzctOWM3Yi02N2VmMTZlNWY3MTYiLCJpI-
jowfQ.i4KuwEtc5_6oIYz5TDWcdzl5bMkvCpLZTSZG2Avy84w'
headers = {
    'Authorization': f'Bearer {auth_token}'
}

file_path = 'test.jpeg'

data = {
  'casualty_id': 1,
  'team': 'TEST_TEAM',
  'system': 'UGV_1',
  'time_ago': 0
}

with open(file_path, 'rb') as f:
  files = {
      'file': f
  }
```

```
  response = requests.post(url, files=files, headers=headers, data=data)

if response.status_code == 200:
  print("Successful")
else:
  print(f"Error Type: {response.status_code} Details: {response.text}")
```

8 Appendix B – Standards/References

The following are the applicable standards and references for this document:

- Ethernet: IEEE 802.3; twisted-pair cable, e.g., IEEE 803.3ab
- HyperText Transfer Protocol 1.1: IETF RFC 7230 (https://tools.ietf.org/html/rfc7230)
- TCP/IP: IETF RFC 1122 (https://tools.ietf.org/html/rfc1122)
- URI: IETF RFC 3986 (https://tools.ietf.org/html/rfc3986), specifically the path information
- Bearer Token Authentication: IETF RFC 6750 (https://tools.ietf.org/html/rfc6750)
- JSON: IETF RFC 8259 (https://tools.ietf.org/html/rfc8259)