Blue-Force Tracking using

Commodity Smart Phone Technology

# Software Engineering System

# Design Document

Version 1.0

17/10/2017

Prepared by

2017-SP5-1

Danielle Heinrich

Table of Contents

[1. Introduction 7](#_Toc496039832)

[1.1 Purpose of the SDD 7](#_Toc496039833)

[2. Business Requirements 8](#_Toc496039834)

[2.1 Executive Summary 8](#_Toc496039835)

[2.2 Project Purpose/Justification 8](#_Toc496039836)

[2.2.1 Business Need/Case 8](#_Toc496039837)

[2.2.2 Business Objectives 9](#_Toc496039838)

[2.3 Project Description 10](#_Toc496039839)

[2.3.1 Project Objectives and Success Criteria 11](#_Toc496039840)

[2.3.2 Requirements 11](#_Toc496039841)

[2.3.3 Constraints 13](#_Toc496039842)

[2.3.4 Assumptions 13](#_Toc496039843)

[2.3.5 Preliminary Scope Statement 14](#_Toc496039844)

[2.4 Risks 14](#_Toc496039845)

[2.5 Project Deliverables 14](#_Toc496039846)

[2.6 Summary Milestone Schedule 15](#_Toc496039847)

[2.7 Summary Budget 16](#_Toc496039848)

[2.8 Project Approval Requirements 17](#_Toc496039849)

[2.9 Project Manager 17](#_Toc496039850)

[3. System Requirements 19](#_Toc496039851)

[3.1 Functional Requirements 19](#_Toc496039852)

[3.1.1 Android device application 19](#_Toc496039853)

[3.1.2 Situational awareness display application 19](#_Toc496039854)

[3.1.3 Backend server 19](#_Toc496039855)

[3.2 Non-Functional Requirements 20](#_Toc496039856)

[3.2.1 Usability 20](#_Toc496039857)

[3.2.2 Performance 20](#_Toc496039858)

[3.2.3 Security 20](#_Toc496039859)

[3.2.4 Design 20](#_Toc496039860)

[3.2.5 Supportability 21](#_Toc496039861)

[3.2.6 Implementation 21](#_Toc496039862)

[4. Design Considerations 22](#_Toc496039863)

[4.1 Goals and Guidelines 22](#_Toc496039864)

[4.2 Development Methods & Contingencies 23](#_Toc496039865)

[5. Operational Scenarios 24](#_Toc496039866)

[5.1 Key System Actors 24](#_Toc496039867)

[5.2 Use Case Descriptions 24](#_Toc496039868)

[5.3 Use Case Diagrams 25](#_Toc496039869)

[6. System Architecture and Architecture Design 28](#_Toc496039870)

[6.1 Logical View 28](#_Toc496039871)

[6.2 Physical View 31](#_Toc496039872)

[6.3 Information Architecture 32](#_Toc496039873)

[6.3.1 Records Management 32](#_Toc496039874)

[6.4 Internal Communications Architecture 41](#_Toc496039875)

[6.5 System Architecture Diagram 43](#_Toc496039876)

[7. System Design 44](#_Toc496039877)

[7.1 Database Design 44](#_Toc496039878)

[7.1.1 Data Objects and Resultant Data Structures 44](#_Toc496039879)

[7.2 Data Conversion 44](#_Toc496039880)

[7.3 External Tools (APIs) 44](#_Toc496039881)

[7.4 User Interface Design 45](#_Toc496039882)

[Appendix A: Record of Changes 46](#_Toc496039883)

[Appendix B: Acronyms 47](#_Toc496039884)

[Appendix C: Glossary 48](#_Toc496039885)

[Appendix D: Referenced Documents 49](#_Toc496039886)

[Appendix E: Approvals 50](#_Toc496039887)

[Appendix F: Additional Appendices 51](#_Toc496039888)

[Appendix G: Declaration of Contribution 52](#_Toc496039889)

List of Figures

[Figure 1: Commodity Blue-Force Situation Awareness 9](#_Toc496039753)

[Figure 2: Gantt chart schedule of milestones 14](#_Toc496039754)

[Figure 3: Actor operating mobile Android device use case diagram 23](#_Toc496039755)

[Figure 4: Actor operating situational awareness display use case diagram 24](#_Toc496039756)

[Figure 5: Developer operating backend server use case diagram 25](#_Toc496039757)

[Figure 6: Logical Component Architecture 26](#_Toc496039758)

[Figure 7: Sequence Diagram – registering a device and visualising it on a web-based browser 27](#_Toc496039759)

[Figure 8: Entity Relationship Diagram 28](#_Toc496039760)

[Figure 9: High Level System Architecture 29](#_Toc496039761)

[Figure 10: Deployment Diagram 30](#_Toc496039762)

[Figure 11: Complete list of Traccar database tables 32](#_Toc496039763)

[Figure 12: Device forwarding location data to Traccar Server 40](#_Toc496039764)

[Figure 13: Device forwarding location data to Traccar Server and a PCAP file 40](#_Toc496039765)

[Figure 14: Device forwarding location data to Traccar Server and an external server 41](#_Toc496039766)

[Figure 15: Layers and Packages 41](#_Toc496039767)

List of Tables

[Table 1: Business objectives supporting DST Group and NSID strategic plans 9](#_Toc496039768)

[Table 2: Preliminary milestone schedule 15](#_Toc496039770)

[Table 3: Summary budget 16](#_Toc496039772)

[Table 4: Android application use case descriptions 24](#_Toc496039773)

[Table 5: Situational awareness display use case descriptions 24](#_Toc496039774)

[Table 6: Server backend use case descriptions 25](#_Toc496039775)

[Table 7: Data 33](#_Toc496039784)

[Table 8: application\_settings (PK = id) 35](#_Toc496039786)

[Table 9: application\_settings imported keys 35](#_Toc496039787)

[Table 10: device\_icons (PK = id) 35](#_Toc496039788)

[Table 11: device\_icons imported keys 35](#_Toc496039789)

[Table 12: device\_icons exported keys 35](#_Toc496039790)

[Table 13: devices (PK = id) 36](#_Toc496039791)

[Table 14: devices imported keys 36](#_Toc496039792)

[Table 15: devices exported keys 36](#_Toc496039793)

[Table 16: groups (PK = id) 37](#_Toc496039794)

[Table 17: groups imported keys 37](#_Toc496039795)

[Table 18: groups exported keys 37](#_Toc496039796)

[Table 19: groups\_users (PK = user\_id, group\_id) 37](#_Toc496039797)

[Table 20: groups\_users imported keys 37](#_Toc496039798)

[Table 21: pictures (PK = id) 37](#_Toc496039799)

[Table 22: pictures exported keys 38](#_Toc496039800)

[Table 23: positions (PK = id) 38](#_Toc496039802)

[Table 24: positions imported keys 38](#_Toc496039803)

[Table 25: positions exported keys 38](#_Toc496039804)

[Table 26: reports (PK = id) 38](#_Toc496039805)

[Table 27: reports exported keys 39](#_Toc496039806)

[Table 28: reports\_devices (PK = report\_id, device\_id) 39](#_Toc496039807)

[Table 29: reports\_devices imported keys 39](#_Toc496039808)

[Table 30: reports\_groups (PK = report\_id, group\_id) 39](#_Toc496039810)

[Table 31: reports\_groups imported keys 39](#_Toc496039811)

[Table 32: reports\_users (PK = user\_id, report\_id) 39](#_Toc496039812)

[Table 33: reports\_users imported keys 39](#_Toc496039813)

[Table 34: ui\_state (PK = id) 39](#_Toc496039814)

[Table 35: ui\_state imported keys 39](#_Toc496039815)

[Table 36: user\_settings (PK = id) 40](#_Toc496039816)

[Table 37: user\_settings exported keys 40](#_Toc496039817)

[Table 38: users (PK = id) 40](#_Toc496039818)

[Table 39: users imported keys 41](#_Toc496039819)

[Table 40: users exported keys 41](#_Toc496039820)

[Table 41: users\_devices (PK = users\_id, devices\_id) 41](#_Toc496039821)

[Table 42: users\_devices imported keys 41](#_Toc496039822)

## Introduction

The System Design Document pertains to a proposed commodity Blue Force tracking system. As it stands, the project sponsor does not possess an existing commodity tracking system. The document outlines the high-level system design and the low-level detailed design specifications for the commodity Blue Force Tracking system under development. Considerations with regards to design are discussed, in addition to the provision of operational scenarios, system architecture diagrams and system design schemas.

### Purpose of the SDD

The System Design Document details and tracks the necessary information required to effectively define architecture and system design in order to give the development team guidance on the architecture of the system to be developed. Generally, the intended audience for such a document would be the project team and development team, but for the purpose of this project, the document is written with the project sponsor and other external agencies in mind. As the proposed project may be further developed by the project sponsor, it is imperative that data relating to decisions, architecture, and design of the project are available for future reference.

## Business Requirements

### Executive Summary

Blue Force Tracking is a system that provides military commanders and troops with information in regards to the location of friendly and hostile forces (Smithsonian 2017). The system provides timely and accurate real-time visualisation of a combat zone, which enhances situational awareness and lowers the risk of friendly fire (Smithsonian National Air and Space Museum 2012). Blue Force Tracking systems comprise of the following:

* A computer to show location information
* A satellite terminal and antenna to send location and other data
* A global positioning satellite (GPS) receiver to ascertain its own bearings
* Command-and-control software to give and receive orders, along with other support functions
* Mapping software that displays the Blue Force Tracking device, along with the locations of other friendly and enemy forces (Wikipedia 2017)

This project will be looking at making use of commodity hardware to simulate Blue Force Tracking systems, which are otherwise unavailable for use in Defence Science and Technology Group (DST Group) trials. GPS locations from Android devices are to be sent to a server, which in turn will generate a situational awareness display and pass a stream of collated GPS positions to a data diode. This will allow for the emulation of Blue Force Tracking systems in future trials, enhancing situational awareness and permitting near real time analysis of trial events as they unfold.

### Project Purpose/Justification

The purpose of this project is to construct a simulated Blue Force Tracking system that can be utilised during science and technology trials. Trial events need to emulate real military operations as much as possible and providing a system to emulate that of Blue Force Tracking systems will increase trial realism.

#### Business Need/Case

A major undertaking of the project sponsor group (Information Architectures) within the National Security and Intelligence, Surveillance and Reconnaissance Division (NSID) of DST Group is to explore and develop Integrated Intelligence, Surveillance and Reconnaissance (ISR) software to support Defence and its coalition partners (DST Group 2017). In order to validate and test the exemplar software produced, experimental trials are conducted. These trials are typically conducted with Defence service personnel alongside DST Group personnel and must imitate real Defence operations as much as possible.

In Australia, Defence utilise specialised Blue Force GPS Tracking systems for operations. As a customer of the project sponsor group, Defence have requested the capabilities provided by a Blue Force Tracking system during trials to assist in providing a realistic military setting by establishing ground truth, which is the actual outlook of a tactical situation.

Unfortunately, the Blue Force Tracking system and respective devices used by Defence are not readily available for use in science and technology trials. As such, the project sponsor group are lacking the functionality that these GPS tracking systems provide. Therefore, there is a need to emulate the capability of Blue Force GPS Tracking systems, which would be used to enhance near real-time situational awareness of troops during trials and allow near real-time analysis of the trial events as they unfold. The emulated system would also enrich the genuineness of experimental trials, which may in turn lead to improved ISR software as it would allow for the testing of exemplar applications in a more realistic setting.

#### Business Objectives

This project supports not only the DST Group strategic plan, but also the NSID strategic plan in several ways. The business objectives for the project which relate to the strategic plans are outlined in Table 1.

Table 1: Business objectives supporting DST Group and NSID strategic plans

|  |  |  |
| --- | --- | --- |
| **Strategic Plan** | **Component** | **Business Objectives for Project** |
| DST Group | Vision   * Support and transform Australia’s defence and national security | Support defence and national security of Australia by allowing personnel to conduct trials in a more realistic environment and review their actions in near real-time. Near real-time analysis could lead to more informed operational decisions |
| Values   * Innovation – actively looking for better ways of doing business | Improve how business is done by providing a means in which personnel can conduct trials in a more realistic environment |
| Strategic Initiatives   * Strategic engagement with client focus | Client focus is the primary driver for this project as it has been derived through the request of Defence, who are a major client of DST Group |
| NSID | Mission   * To enhance Australia’s capability to produce accurate, relevant, timely and actionable information to enable decision-making superiority for Defence and to support whole-of-Government national security science and technology coordination and delivery | Increase the volume of accurate, relevant, timely and actionable information by providing ground truth capabilities during trials. Improve decision-making by enhancing the near real-time situational awareness of troops during trials and allowing near real-time analysis of the trial events as they unfold |
| Values   * Openness – within security constraints, encouraging free information exchange and research collaboration across organisational boundaries enabling new capabilities for the benefit of Australian Defence and national security | Encourage free information exchange and research collaboration across organisations by utilising free and open source software, working with external organisations for the development of the project and allowing the University of South Australia to further modify and use the software solution without restriction |
| Core Roles   * Sustainment – supports sustainment, improvement and operational effectiveness of Defence ISR systems | Boost operational effectiveness of ISR systems by allowing for the testing of exemplar applications in a more realistic setting, which can lead to improved software |
| Science and Technology Capabilities Goals   * Integrated ISR Enterprise Analysis and Experimentation * Integrated ISR Architecture Development and Demonstration | By improving the realism of science and technology experimental trials. Provides test data that can be fed into exemplar software to shape possible system integration with Blue Force Tracking systems. |
| Science and Technology Capabilities Goals   * Multi Source Data Analysis and Information Fusion | By enhancing the near real-time situational awareness of troops during trials and allowing near real-time analysis of the trial events as they unfold |

### Project Description

There are two major components to this project. Firstly, Android devices with an application capable of capturing and forwarding periodic GPS updates and other pertinent data to an internet connected server will simulate the functionality of a Blue Force Tracking system being able to receive information relating to friendly and enemy forces; the Android devices will represent friend or foe systems. The second component will consist of a server, which will generate a web-based situational awareness display of the Android devices, as well as pass a stream of the collated GPS positions and relevant data of devices to a data diode. While traditional Blue Force Tracking systems are able to determine and send their own location, the simulated server will not have this functionality. In addition, the server will not feature command-and-control software and will therefore not be capable of giving or receiving orders to actor devices.

Figure 1: Commodity Blue-Force Situation Awareness



The tracking devices along with the server and situational awareness visualisation will emulate the functionality of Blue Force Tracking systems for use in future trials.

Figure 1 illustrates how the system will operate during a trial. From the left of the diagram, Android devices are carried by actor vehicles and people as they move around the exercise area. These devices, loaded with the afore mentioned application, will send periodic position update reports to an internet connected server. On the right-hand side, the server will parser the messages from the devices, generate a situational awareness display and pass a stream of the collated device data to a cross-domain diode. In the top right of the diagram, an actor with an internet connected device will be able to access the situational awareness display produced by the server using a web browser.

In order to reduce the software development efforts, both the Android application and the internet connected server application may be derived from existing open source software.

#### Project Objectives and Success Criteria

Develop a simulated commodity-based Blue Force Tracking system for the purpose of science and technology trials within 13 weeks.

#### Requirements

Input from the project sponsor has emphasised a number of high-level project requirements which act as guidelines within which the project must conform:

* The system must emulate the functionality of Blue Force Tracking systems
  + In Blue Force Tracking systems, the system knows its own location, as well as the location of other devices. For the project, the backend server does not need to know its own location
  + Android devices do not need to know the location of other Android devices
* The server shall support up to 20 Android devices at a time, but an extensible solution that supports a greater number of devices is desirable
* Android devices must have the following configurable attributes:
  + Location upload intervals, with a minimum upload time of 1 second
  + Server IP address and port
* Location data sent to the server must be collaborated and output as a UDP stream
  + Collaborated data should handle the data of multiple Android units simultaneously
  + Data output as a UDP stream needs to be of a format specified by the client
* Location data must be sent from an Android device, reach the server and be output as a UDP stream within 5 seconds
* The situational awareness display should be web-based so it is accessible from all internet connected devices using a web browser
* Access to the situational awareness display should require authorisation and authentication credentials
* The situational awareness display should be configurable and allow the user to stop tracking certain Android devices and change the displayed icons of devices to represent friendly or hostile units
* The situational awareness display should feature a configurable time zone, but if this is not possible, UTC time should be used
* Extant open standards for serving and visualising information should be used to enable the reuse of what is developed
* The situational analysis visualisation should make use of internet connected foundation imagery services
* The data stream going to the cross-domain diode should be a UDP stream
  + The diode should be treated as a system boundary.
  + The diode will expose a UDP server socket.
* The server shall implement a REST service to allow for the request of track data for a given period of time
* The server should log the received messages
* The user must be able to turn on and off the GPS location upload functionality on Android devices
* User experience needs to be simple so that personnel with minimal experience can operate both the Android devices and web-based situational awareness display with little instruction
* The backend software should be packaged in such a way to allow installation on standalone networks
* It is proposed that a free AWS compute server or similar is used to host the server backend software
* Stretch goal: there should be functionality to allow for the replay of recorded messages

#### Constraints

A number of limitations in regards to people, time and equipment available for the project have been identified and are detailed below:

* As the project is to be undertaken by just one person, the skill set and technical knowledge required will be confined to that individual. The lack of a cross-functional team will mean that the full potential of deliverables may not be met – for example, as the individual has minimal networking expertise, the networking portion of the project may consist of only basic, simplified functionality. The lack of team also means the number of man-hours available to the project will be far fewer than that of a team-based project.
* Another constraint on time is the fact that the project is to span approximately 13 weeks, a period that is non-negotiable. This is due to the fact that the project is undertaken as part of a University assessment, and is confined to a semester of work. Therefore, there is no room for extension.
* Testing multiple devices concurrently may also prove to be problematic due to the lack of team – it will not be possible to gather simultaneous, differing test data using the Android devices with just one person. The devices will either have to capture track data with different timestamps, or capture stationary track data with identical timestamps. Ideally, you would have several devices capturing active, moving GPS tracks at the same time to test that the system functions as expected.
* As the number of Android devices supplied is limited and largely unvarying, this will reflect in the test GPS data supplied to the server. Limited test data will mean that GPS data sent from other commodity devices will be untested and unaccounted for on the server side. Should the project expand to use different commodity devices, their capabilities will be unknown and may produce unpredicted results. The small number of similar devices provided mean it will not be possible to develop ‘generalised’ software – the software will only be guaranteed to work with the devices provided.
* The Android devices provided for use during trials will be of low processing power and will feature low resolution displays, with the likely vertical resolution being 480 pixels. Therefore, the application chosen will need to be computationally lean, as well as offer support for low resolution devices.

#### Assumptions

The following assumptions highlight expected conditions of the project, without it having been specifically stated:

* The choice of development tools used to build the software will be at the discretion of the project team
* If the developed software is based on open source software, the licencing agreements for the derived software will remain unchanged
* The commodity Android devices and related hardware will be in functioning condition and have access to mobile networks, Wi-Fi and an application market so that client side software can be installed and managed
* If the server is hosted on a cloud-based, virtual machine, the machine will be available for the duration of the project and resizable as required

#### Preliminary Scope Statement

The scope of this project is to include the following:

* Delivery of a commodity-based, mock Blue Force Tracking system as outlined in the requirements section, for use by personnel in experimental science and technology trials.
* Delivery of relevant documentation.
* Delivery of a test data stream.

Completion of the project will be established when software and test data to be delivered is tested and stable, and documentation fulfils its purpose. Stretch goals, where stated, do not need to be achieved for the project to be characterised as complete.

### Risks

The following high-level risks have been found to apply to this project:

* There is a risk that due to the strict deadline of the project, it may not be completed to a satisfactory level in time. This risk can be mitigated by clearly defining the requirements before commencing the project and holding discussions with the project sponsor if it is believed that some requirements may be too difficult for one person to accomplish in the timeframe given.
* Another risk is that the delivered software will not be suitable for the purpose of science and technology trials as intended. This may be due to low quality software, or the fact that the software cannot be readily tested for ‘heavy load’ situations, in which there are a large number of devices reporting their location data simultaneously. The risk of producing low quality software can be minimised by clearly defining the requirements and ensuring a suitable number of use cases have been explored. Ensuring the project sponsor and academic supervisor are kept notified of major milestones in development should assist in ensuring the project is advancing as intended. Reducing the risk associated with the software not running as intended under ‘heavy load’ post-completion may not be possible due to the constraints listed earlier – a lack of team members and devices means the ability to ‘stress test’ is not readily available.
* The availability and suitability of a free cloud compute server for hosting backend and web interface services could also pose a risk to the project. Free server instances are likely to offer limited infrastructure or software capabilities, which may prove inadequate for the purposes of the project. To alleviate the risk of using an ill-equipped compute server, research should be conducted into the offerings and limitations of cloud providers. The project sponsor has also offered to assist in establishing a more suitable solution if a free server proves inadequate.

### Project Deliverables

* Android devices loaded with software capable of capturing and forwarding periodic GPS updates and other pertinent data
* Backend server
* PCAP file of captured UDP stream
* Documentation
  + Information with regards to design and architecture decisions
  + User guides
  + Instructions on standing up the server
  + WADL definition for REST interface

### Summary Milestone Schedule

A preliminary schedule of all high-level project milestones is described in Table 2.

Table 2: Preliminary milestone schedule

|  |  |
| --- | --- |
| **Milestone Name and Description** | **Expected Completion Date** |
| Delivery of System Requirements Document | 13 August 2017 |
| Delivery of Project Proposal Presentation Slides | 13 August 2017 |
| Project Proposal Presentation | 15 August 2017 |
| DST Group Trial   * This trial can be used for testing of the system, if needed * Trials run for three weeks, commencing at the given date | 16 October 2017 |
| Delivery of Final Project Presentation Slides | 29 October 2017 |
| Delivery of Fair Day Poster | 29 October 2017 |
| Delivery of System Design Document | 5 November 2017 |
| Delivery of Final Deliverables | 5 November 2017 |
| Delivery of Fair Day Presentation Slides | 5 November 2017 |
| Project Presentation | 5 November 2017 |
| Project Fair Day | 9 November 2017 |
| DST Group Trial   * The trial for which the system is intended to support * Trials run for three weeks, commending at the given date | 13 November 2017 |

Figure 2 outlines a refined schedule in which development tasks, in addition to high-level milestones, are plotted against time frames.

Figure 2: Gantt chart schedule of milestones



### Summary Budget

Table 3: Summary budget

|  |  |
| --- | --- |
| **Component** | **Cost** |
| Android smartphones | On loan from sponsor |
| Pre-paid mobile sim cards | On loan from sponsor |
| Pre-paid recharge vouchers | Provided by sponsor |
| Cloud compute server | Free – if a free server is not suitable, a more suitable solution is to be established in consultation with the project sponsor |
| Open source software | Free |
| Development hardware | The project team has the choice to use personally owned, University-based, or hardware provided by the sponsor. With the exception of personally owned hardware, the other choices will bear no cost on the project team |
| Development software | The project team has the choice to use personally owned, University-based or sponsor-based development software. With the exception of personally owned software, the other choices will bear no cost on the project team |
| Networking and internet connectivity | The project team has the choice to use personally owned, University-based or sponsor-based network facilities and internet provision. With the exception of personally owned network and internet functionality, the other choices will bear no cost on the project team |
| Travel | Meetings with the client and academic supervisor are to be held at either the University of South Australia Mawson lakes campus or on-site at DST Group Edinburgh. These costs are expected to be minimal and will be borne by the project team |

### Project Approval Requirements

The requirements which must be met to gain project approval include:

* The project is to emulate the capability of Blue Force Tracking systems
* The project must include a configurable application loaded on Android devices, used to report location data to a server
* The server must be packaged in such a way to allow for it to be installed on a different host machine
* The server is to receive and log data from the Android devices, visualise the devices on a configurable web-based display, and stream received data to a specified UDP port
* The server must feature a REST service endpoint and allow for the request of logged data
* The project must be extensible
* All high-level project requirements outlined in section 2.3.2 have been met

### Project Manager

As this project is being undertaken solely by Danielle Heinrich, that person will also act as the project manager.

The project manager will be responsible for all client communication, management and development obligations.

To ensure optimal client communication, the project manager will initially liaise with the sponsor at least once a week, or as often as reasonably necessary. Once the project requirements and scope have been defined and authorisation to move ahead with the project has been sought from the sponsor, it is expected that client communication will occur less frequently.

Meetings with the academic supervisor will also be held on a weekly basis to start, with the rate of communication expected to fall as the project matures.

Whenever meetings with the client or academic supervisor are sought, the project manager will prepare required documentation, such as meeting minutes and clarification questions, ahead of time. Following a meeting, a summarisation of meeting discussions will be sent to the respective persons – this will assist in reiterating the understanding of project specifications between parties, as well as acting as a log of evidence.

The project manager will also generate the project schedule, referring to it weekly to track the progress of project milestones. If milestones are at risk of running over their allotted period of time, the project manager will investigate the cause of delay and either reallocate project resources or update the schedule accordingly. Conversely, if milestones are completed ahead of time, the schedule can be updated to reflect this, and additional time can be instead allocated to other elements of the project.

## System Requirements

### Functional Requirements

#### Android device application

The system shall:

FR1: Present an interface in which an actor can configure if their device transmits location data, where the device transmits location data to and how often the device transmits location data.

FR2: Allow the actor to toggle whether location data is sent or not

FR3: Allow the actor to configure the IP address and port of the server to which the location data must be sent

FR4: Allow the actor to configure the upload interval time for the data sent to the server

#### Situational awareness display application

The system shall:

FR5: Present an interface in which an actor can view the representative positions of Android devices on a map

FR6: Allow the actor to toggle the display of individual devices on a map on or off

FR7: Allow the actor to stop tracking a device, so the application no longer receives data pertaining to that device

FR8: Allow the actor to start tracking a device, so the application receives data pertaining to that device

FR8.1: By default, the device should have an ‘unknown’ icon

FR8.2: By default, the device should be displayed on the map

FR9: Allow the actor to assign a friendly, hostile or unknown icon to a device

FR9.1: By default, devices should have an ‘unknown’ icon

FR10: Allow the actor to display only friendly, hostile or unknown icons (or any combination of these groupings) on the map

FR10.1: By default, all groupings are displayed at the same time

FR11: Allow the actor to configure the time zone

FR11.1: By default, the time zone should be UTC

FR12: Allow the actor to replay previously recorded location data

FR12.1: The actor must select a date and time period

#### Backend server

The system shall:

FR13: Allow the developer to install the backend server on a host machine

FR14: Allow the developer to conduct a REST service request to get previously logged data

FR14.1: The developer must select a date and time period

FR15: Allow the developer to configure the IP address and port of the UDP socket to which the streaming location data is sent

FR16: Collaborate received location data and output as a stream

FR17: Log received messages

### Non-Functional Requirements

#### Usability

The system shall:

NFR1: Allow the user to zoom in and pan around the situational awareness display map

NFR2: User interfaces must be simple and intuitive to use, with little instruction necessary

#### Performance

The system shall:

NFR3: Ensure the server can support up to 20 mobile devices at a time

NFR4: Ensure the server is extensible to support a greater number of mobile devices in the future

NFR5: Ensure the mobile application can configure the upload interval time to a minimum of 1 second

NFR6: Location data must be sent from a mobile device, reach the server and be output as a stream within 5 seconds

#### Security

NFR7: Ensure access to the web-based situational awareness display is protected by authorisation and authentication controls

#### Design

NFR8: Mobile phone application must run on an Android operating system

NFR9: Mobile phone application must operate on low performance CPUs (approximately 1.3GHz)

NFR10: Mobile phone application must operate on low screen resolution (approximate vertical resolution of 480 pixels)

NFR11: Ensure the collaborated output stream from the server is formatted according to client specifications

NFR12: Ensure the situational awareness display application is web-based and accessible from Internet connected devices using a web browser

NFR13: Ensure the backend software is packaged to allow for the installation on standalone networks

#### Supportability

NFR14: Ensure the installation process of the backend server is configured in such a way to require minimal user input

NFR14.1: Installation documentation should be provided separately

#### Implementation

NFR15: Ensure the collaborated output stream from the server is sent over UDP

NFR16: Ensure the system uses open standards for serving and visualising information

NFR17: Ensure the situational awareness display makes use of internet connected foundation imagery services

NFR18: A free Amazon Web Services compute server should be used to host the backend software

## Design Considerations

### Goals and Guidelines

The design of the commodity-based Blue Force tracking system is dictated by several goals and guidelines, which have been derived from the requirements outlined in section 3. Goals and guidelines of particular importance to the design of the system are:

* The system must emulate the functionality of Blue Force Tracking systems, with a few minor exceptions, such as the system not needing to know its own location. This goal is vital and represents the purpose of the project – to construct a simulated system that can be utilised during science and technology trials to increase realism.
* The system must use open standards for serving and visualizing information and make use of internet connected foundation imagery services. This goal is required to enable reuse of the developed software.
* The user experience needs to be simple so both Android devices and the web-based situational awareness display can be operated with minimal instruction. With regards to the situational awareness display, this includes zoom and pan functionality, and the ability to toggle devices on or off. The display should also offer the ability to configure device icons to reflect military symbols. The Android devices should have configurable settings for identifier, reporting interval, server address and server port. The reasoning behind this goal is the fact that the devices and web-based display are to be used by Defence service and DST Group personnel in experimental trials. Personnel may be unfamiliar with the device application and situational awareness display being developed, and trial conditions may be stressful, warranting a simple, intuitive software experience.
* The performance and compatibility of Android devices is a major driver behind the design decisions of the application software. It goes without saying that the tracking client application must be compatible with Android devices, but it must also work on devices with a low-performance CPU and low screen resolutions. This is necessary as the project sponsor will be purchasing a number of cheap, low-quality Android devices for use during experimental trials, for which the application will need to be installed onto.
* The system must be extensible. Initially, the system should support a minimum of 20 devices reporting location data simultaneously. However, this number may increase in the future and the system should be designed to support such a need.
* The system must implement a REST service interface to allow for the request of data for a given period of time. This guideline is at the request of the project sponsor.

In addition to the goals and guidelines listed above, while the fact that the UDP stream output must be in a specific, three-line format does not have significant architectural implications, it does impact implementation details. The same could be said of many of the requirements listed in section 3. There is one further guideline which dominates the design decision – the fact that the system can be based on free and open source software. This is a critical directive and investigation into the suitability of existing software warrants its own document. A Technology Scan and Feasibility Analysis document (Appendix D) was produced which explores the association between the system requirements and the features provided by a number of free and open source applications. Finally, the document makes recommendations on whether existing software should be developed upon, and if so, which software is most suitable.

### Development Methods & Contingencies

Following the thorough investigation described in the Technology Scan and Feasibility Analysis document (Appendix D), it was determined that development of the project would be built upon free and existing open-source software – namely, Traccar Server, Traccar Web UI and Traccar Client. As such, the fundamental system and software design is already in place. However, it was initially necessary to conduct requirements gathering and define operational scenarios to determine which, if any, software would best suit the needs of the project. The System Requirements document (Appendix D) produced in the initial stages of project development provides system requirements, constraints and deliverables which have been used to determine which software would best suit the needs of the project.

While much of the system and software design is already in place, it is not well documented. Instead, the architectural diagrams in this report aim to dissect the existing software being built upon and provide insight into how they have been designed. Such diagrams will provide guidance to the development team as they must build upon the existing design to better suit the needs of the project sponsor, as well as providing reference material for the client and external persons.

As the fundamentals of the software have already been designed, the risk of unforeseen contingencies arising is heightened. Mature software may prove difficult to manipulate to fit the client’s needs as there may be tight coupling between classes or methods, poorly written or hard to understand code, or poor design decisions. The fact that the backend server is authored by a different developer than the web interface adds an extra layer of dimensionality to the problem – changes made in either domain may not be adopted as intended. Exploring and describing the design is likely the best defense against unknown factors that may be present in the existing software. This will better equip the development team with system-wide knowledge so that they may produce effective workarounds to any difficulties faced.

## Operational Scenarios

### Key System Actors

Trial Actors

Software Developers

### Use Case Descriptions

Table 4: Android application use case descriptions

|  |  |
| --- | --- |
| Toggle GPS on/off | Actor toggles the ability to send their GPS location on or off. The system either starts or stops sending location data to a server. |
| Configure server IP and port | Actor configures the IP address and port of the server to which the location data must be sent and the system updates accordingly. |
| Configure upload interval time | Actor configures the interval time for which location data is sent to the server and the system updates accordingly. |

Table 5: Situational awareness display use case descriptions

|  |  |
| --- | --- |
| Login | Actor enters their credentials and the system verifies authentication. If successful, the system proceeds to display the main application. |
| Toggle displaying device | Actor toggles the ability to display a device and the system shows or hides the device from the display accordingly. |
| Stop tracking device | Actor selects a device to stop tracking. The system removes the device from the display and no longer logs its location data. |
| Start tracking device | Actor inputs a unique device identifier and the system starts logging the location data of the device with this identification. The system also shows the device on the display. |
| Assign icon to device | Actor selects an icon to assign to a device. The system updates the display to reflect the icon selected. |
| Toggle friendly, hostile and unknown devices | Actor toggles the ability to display devices based on their icon and the system updates the display to reflect the choice. |
| Configure time zone | Actor chooses a time zone and the system updates accordingly. |
| Replay recorded messages | Actor enters and date and time period and the system displays and replays recorded location data. |

Table 6: Server backend use case descriptions

|  |  |
| --- | --- |
| Install | Actor installs server on a host machine. The system is installed and running. |
| Conduct REST service request | Actor inputs a date and time period to conduct a REST service request. The system returns all logged data from that time. |
| Configure UDP stream IP and port | Actor configures an IP address and port to which streaming data must be sent. The system forwards data to this address. |

### Use Case Diagrams

Figure 3: Actor operating mobile Android device use case diagram

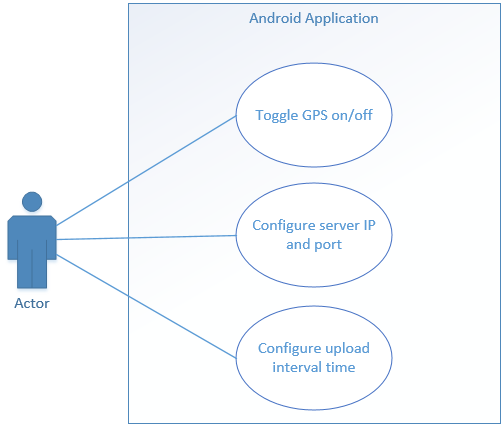


Figure 4: Actor operating situational awareness display use case diagram



Figure 5: Developer operating backend server use case diagram



## System Architecture and Architecture Design

The system is comprised of three applications – Traccar Client, Traccar Server and Traccar Web UI. Traccar Client sends location and identifying information to Traccar Server, which in turn serves information to Traccar Web UI, which provides a web-based interface for users to interact with. While a user does have the ability to interact with Traccar Server without a visual interface through the use of its built in RESTful interface, the client specifically requested a visual display. Likewise, Traccar Server can receive location data from any configurable GPS tracking device – it is not restricted to receiving communication from Android devices. However, as noted previously, the client requires an application capable of functioning on Android appliances.

This particular decomposition was chosen as a result of a technology and feasibility scan, the details of which can be found in a document by the same name (Appendix D).

### Logical View

Figure 6: Logical Component Architecture

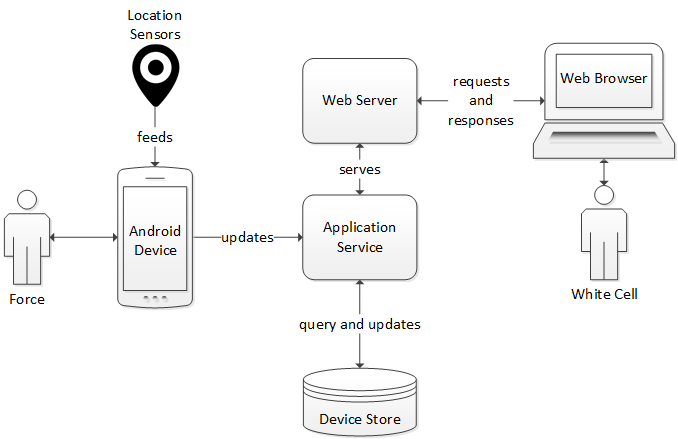


Figure 6 provides a high-level overview of the components of the system and how they interact with one another.

1. Starting from the left, the ‘Force’ user configures Traccar Client, which has been installed on an Android device, with identifying information and an address and port number to send data to.
2. The device receives location data from GPS signals via its onboard location sensors. Location and identifying information is then sent to the server specified by the address and port.
3. The server, with Traccar Server installed, then updates the datastore with the data received from the device.
4. From the right, the ‘White Cell’ actor interacts with a web browser and views visual location data of devices by configuring the identifying information of the device they wish to track.
5. The browser sends a request to the Traccar Web UI application.
6. Information needed by the Traccar Web UI is served up through Traccar Server, which queries the datastore for the required data.

Figure 7: Sequence Diagram – registering a device and visualising it on a web-based browser

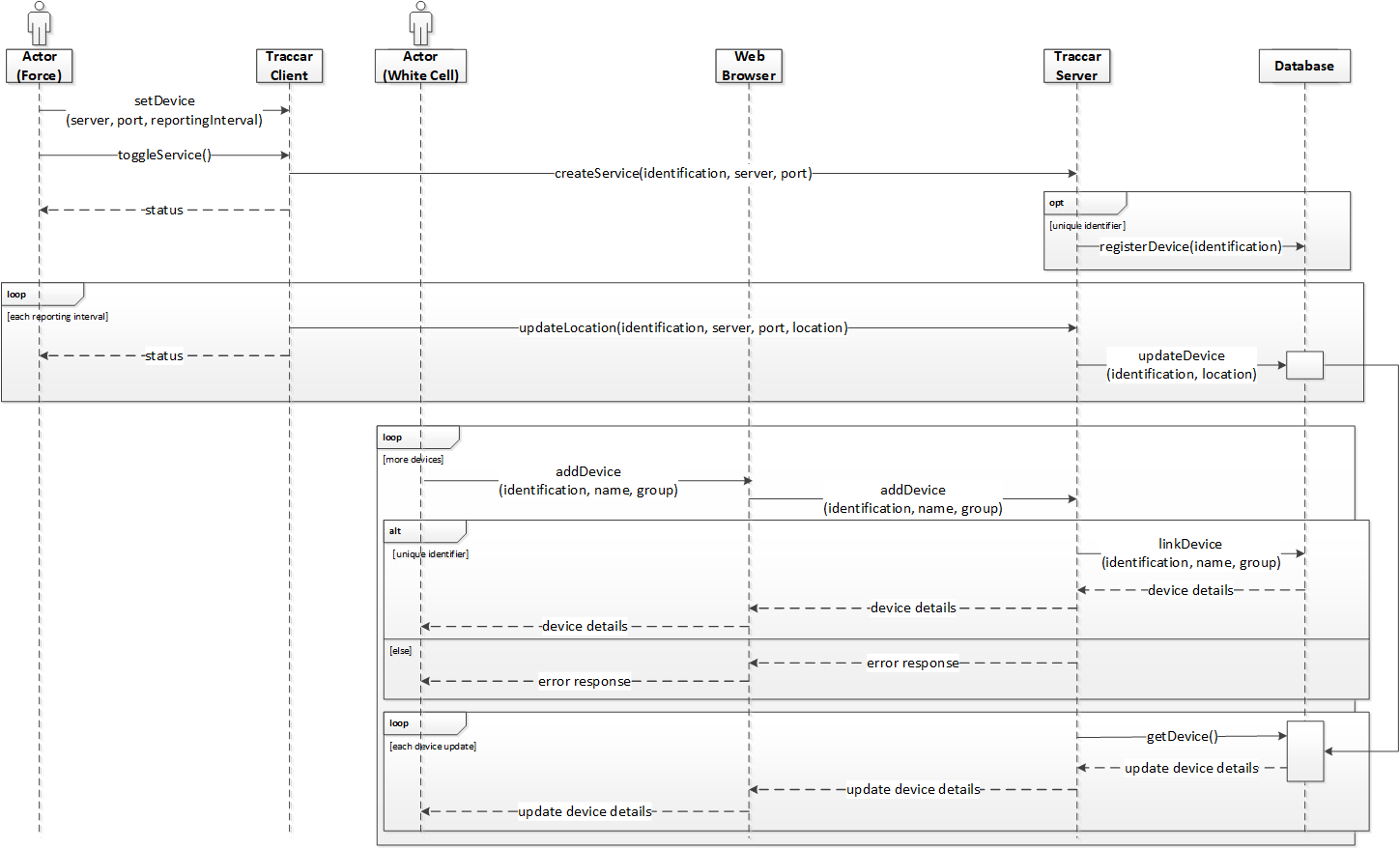


Figure 7 represents the sequence of messages sent between actors and the system in order to register a device and visualise it on a web browser.

Sending location data from a device:

1. The actor (Force) configures the Traccar Client software installed on a device.
   1. At a minimum, the address and port of the server to forward data to must be configured.
   2. The reporting interval can be configured as necessary.
   3. The device identifier is generated automatically but can be manually set.
2. The actor (Force) toggles the Traccar Client service on.
3. Provided the Android Location Services are functional and GPS sensors are running, Traccar Client creates ‘Service Created’ and ‘Location Update’ statuses.
4. Provided the correct server configuration has been applied, Traccar Client configuration settings are sent from the device to the server specified to create a service.
5. If the Traccar Client device identifier is new and unique, the device is registered and stored in the database.
6. At each reporting interval, provided the Android Location Services are functional and GPS sensors are running, Traccar Client creates ‘Location Update’ statuses.
   1. Provided the correct server configuration has been applied, device location data is sent to the server.
   2. Device location data is stored in the database.

Visualising devices on a web based situational awareness display:

1. For each device, the actor (White Cell) configures the details of the device to display.
   1. At a minimum, the device identifier and name must be configured.
   2. Group and other attributes can be configured as necessary.
2. The configuration details are sent as a POST request payload to the Traccar Server REST API.
3. If the device identifier is unique, the configuration details are linked to the relevant device.
   1. If the device identifier is already in use within the web-based application, an error message is returned.
   2. If the device identifier is unique for the web application instance, device details and status are returned.
4. The server checks if a device has been updated. Whenever device data is updated, the device data is retrieved from the database and pushed to the web interface via a websocket.

Figure 8: Entity Relationship Diagram

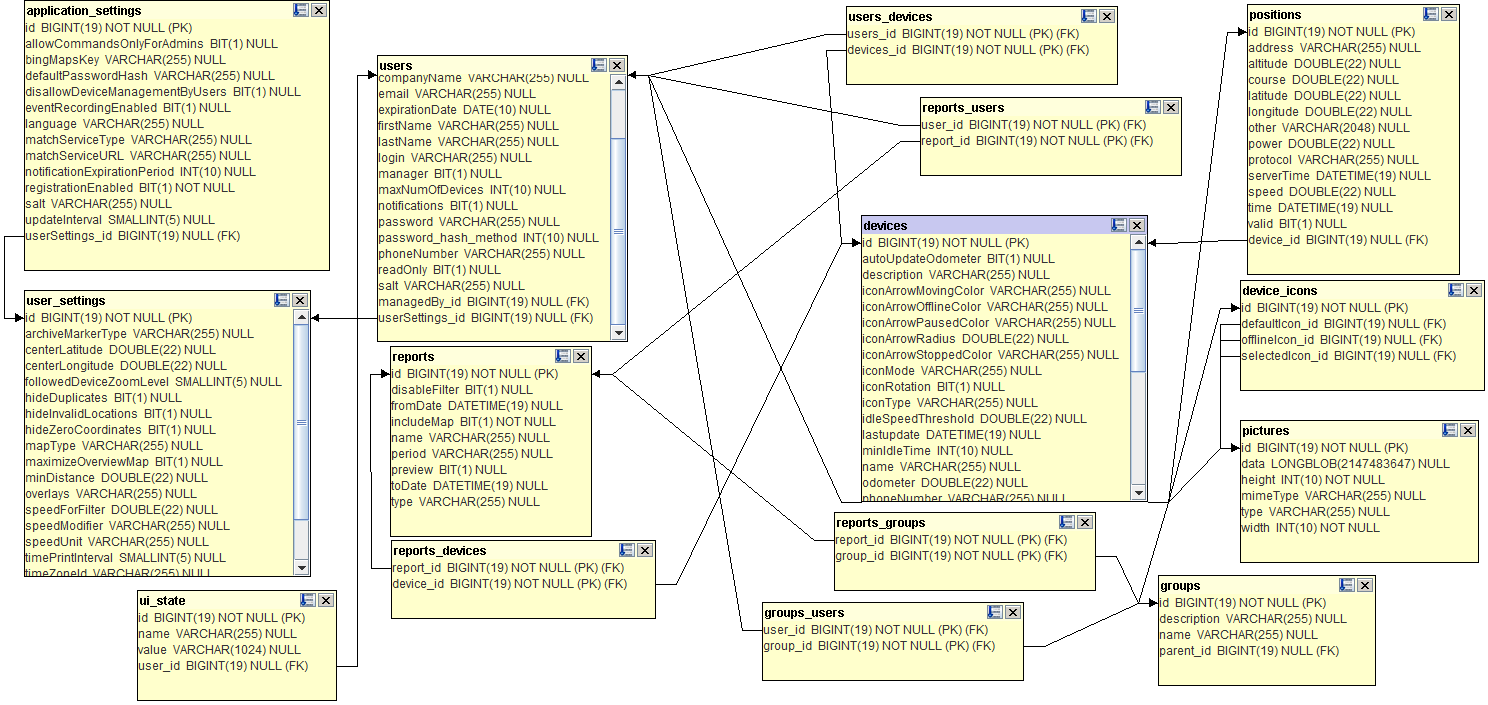


Figure 8 represents the relationship between objects within the system. The tables representing objects are further defined in Section 6.3.

### Physical View

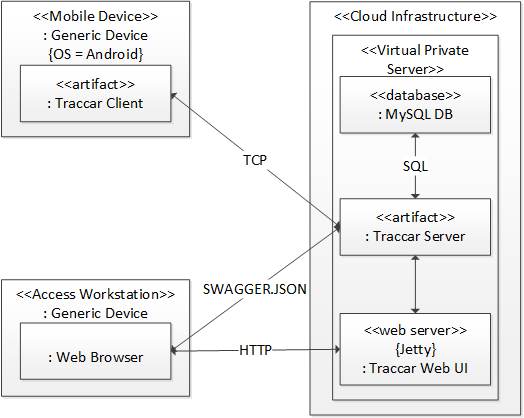
Figure 9: High Level System Architecture



Figure 9 represents a high-level overview of the input and output processes of the system.

1. Location Sensors provide location information to Android devices.
2. Android devices receive location information from sensors and provide a Location Services API for use by installed applications.
3. The Traccar Client Application utilises the Android Location Services API to retrieve location data. The location data is then parsed and forwarded through an Internet connected network using TCP protocol to a server.
4. The Traccar Server is capable of receiving, decoding, and storing TCP messages. It can also serve web interface files and provides an API for access to the database and internal state of the server.
5. The Traccar Web UI generates a situational awareness display on an Internet connected Web Based Browser, as well as streaming received messages to an Internet connected Data Diode using UDP protocol.

Figure 10: Deployment Diagram



### Information Architecture

#### Records Management

By default, Traccar Server uses an embedded H2 database. However, this database is included for development purposes only and the author strongly recommends implementing a more stable database for release. MySQL was chosen as the database to standup for release purposes. The configuration file was modified to utilise the MySQL database as follows:

<entry key='database.driver'>com.mysql.jdbc.Driver</entry>

<entry key='database.url'>jdbc:mysql://[HOST]:3306/[DATABASE]?useSSL=false&amp;allowMultiQueries=true&amp;autoReconnect=true&amp;useUnicode=yes&amp;characterEncoding=UTF-8&amp;sessionVariables=sql\_mode=''</entry>

<entry key='database.user'>[USER]</entry>

<entry key='database.password'>[PASSWORD]</entry>

##### Data

Table 7 describes some of the data that is supplied to the system, including the format and supplier. This list is not a complete list, and instead serves to highlight some of the more important data feeds. A more thorough list of data stored by the system can be found in section 6.3.1.3.

Table 7: Data

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Format** | **Supplied By** | **Interface** |
| Device Identification | Automatically generated from device IMEI number but can be manually input | Either device IMEI or user input | Traccar Client |
| Server Address | Default address provided, but can be manually input | User input | Traccar Client |
| Server Port | Default port provided, but can be manually input | User input | Traccar Client |
| Frequency | Default value provided, but can be manually input | User input | Traccar Client |
| Device Name | Manually input | User input | Traccar Web UI |
| Device Protocol | Automatically generated | Generated according to the port that device data is forwarded to | Traccar Server |
| Device Status | Automatically generated | Generated according to the time of last update from the device | Traccar Server |
| Device Position (including latitude, longitude, course, altitude, speed and time of fix) | Automatically generated | Generated according to position received from external sensors | Traccar Client |
| Device Icon | Default icons provided, but additional icons can be added | User input | Traccar Web UI |
| Group Name | Manually input | User input | Traccar Web UI |
| User (including name, password, phone number and devices) | Manually input | User input | Traccar Web UI |
| Application and User Settings (including map tile-set, filters, timeout) | Default values are provided, but can be manually provided | User input | Traccar Web UI |

##### Manual/Electronic Inputs

Not Applicable

##### Master Files

Figure 11 shows a complete list of the database tables created by Traccar Server. Although the web interface has been modified to remove any reference to maintenance, notifications, and sensors, it was decided that the database would not be modified to reflect this. This was due to the fact that the original Traccar Server and Traccar Web UI configure the database entries in a slightly different fashion from one another. Configuring the entries must be done in a particular order to successfully install Traccar Web UI – failure to do so results in corrupt table creation. Therefore, a decision was made to make no further alterations to the database scheme.

Figure 11: Complete list of Traccar database tables

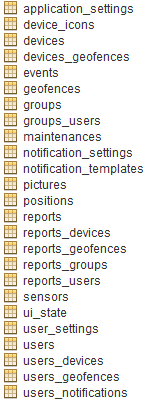


Table 8 through to table 42 describe the data held in each table of relevance to the modified web interface, including the column names, data types and primary and foreign keys.

Table 8: application\_settings (PK = id)

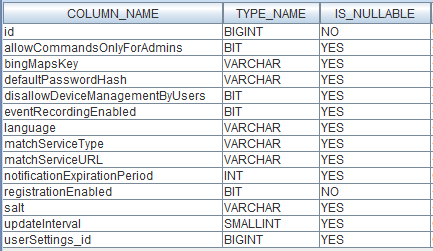


Table 9: application\_settings imported keys



Table 10: device\_icons (PK = id)

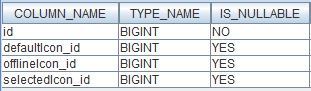


Table 11: device\_icons imported keys

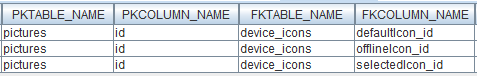


Table 12: device\_icons exported keys



Table 13: devices (PK = id)

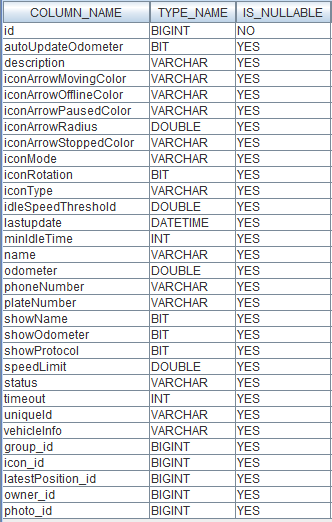


Table 14: devices imported keys

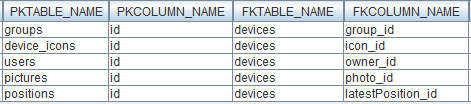


Table 15: devices exported keys

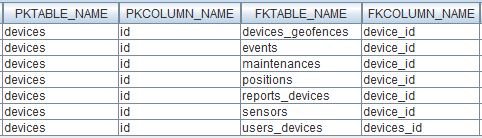


Table 16: groups (PK = id)

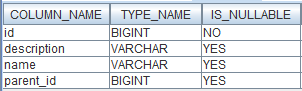


Table 17: groups imported keys



Table 18: groups exported keys

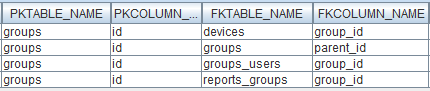


Table 19: groups\_users (PK = user\_id, group\_id)



Table 20: groups\_users imported keys



Table 21: pictures (PK = id)

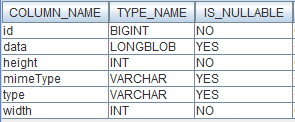


Table 22: pictures exported keys

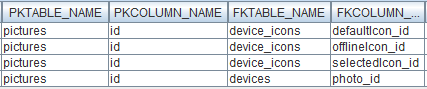


Table 23: positions (PK = id)

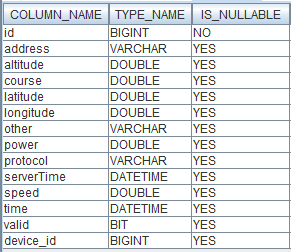


Table 24: positions imported keys



Table 25: positions exported keys



Table 26: reports (PK = id)

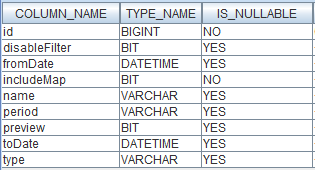


Table 27: reports exported keys

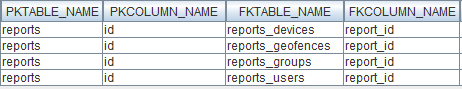


Table 28: reports\_devices (PK = report\_id, device\_id)



Table 29: reports\_devices imported keys



Table 30: reports\_groups (PK = report\_id, group\_id)



Table 31: reports\_groups imported keys



Table 32: reports\_users (PK = user\_id, report\_id)



Table 33: reports\_users imported keys



Table 34: ui\_state (PK = id)

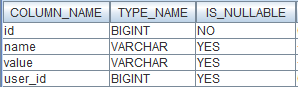


Table 35: ui\_state imported keys



Table 36: user\_settings (PK = id)

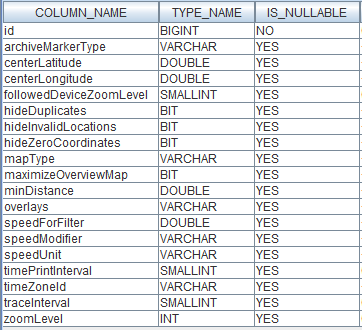


Table 37: user\_settings exported keys



Table 38: users (PK = id)

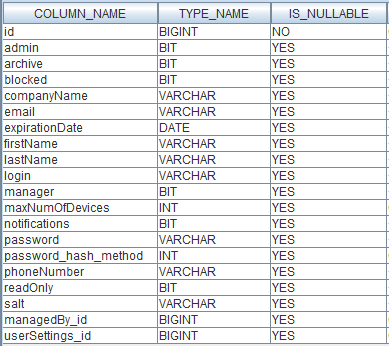


Table 39: users imported keys



Table 40: users exported keys

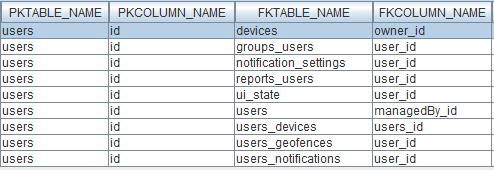


Table 41: users\_devices (PK = users\_id, devices\_id)



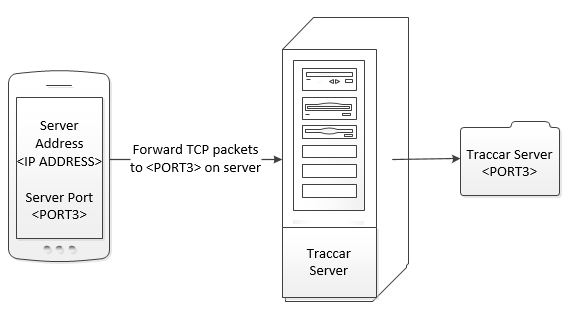
Table 42: users\_devices imported keys



### Internal Communications Architecture

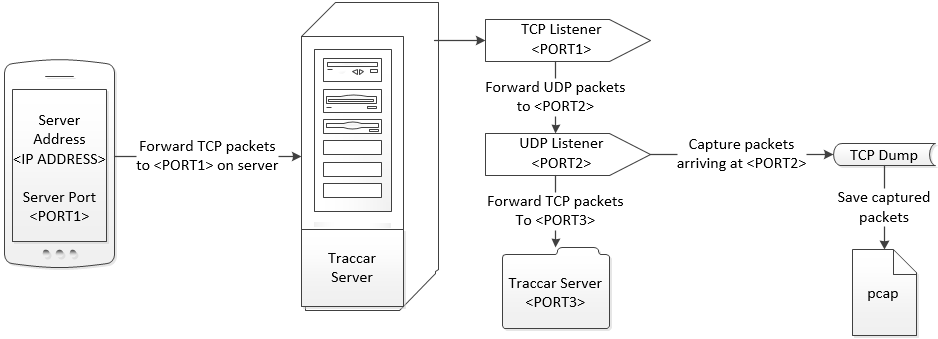
Figure 12 shows how devices forward location data to Traccar Server by default – the port configured on the device must match a port on the server application. Traccar Server is capable of receiving data through many different ports, with each port associated with a different message protocol. Traccar Server uses the port number to identify the decoder required for the incoming data – for example, all data arriving on port 5055 will be decoded according to OsmAnd standards.

Figure 12: Device forwarding location data to Traccar Server



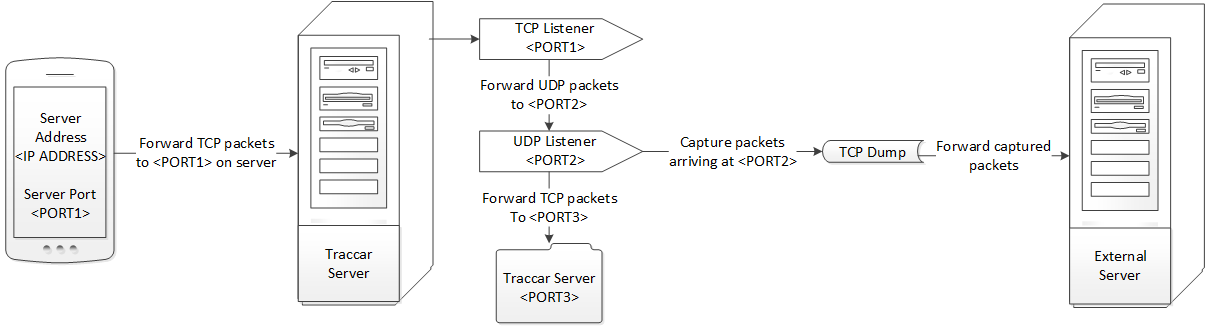
In order for the server to both forward location data to the Traccar application, as well as save the captured data as a UDP stream, the device must forward information to a port that is not in use by Traccar Server. Instead, a device must forward its data to a TCP listener, which in turn forwards the data to a second listener in UDP format. The second listener forwards the data to Traccar Server in TCP format, while a packet capture tool listens and saves all data arriving on this listener. Figure 13 outlines this process.

Figure 13: Device forwarding location data to Traccar Server and a PCAP file



Forwarding packets to an external server or data diode follows the same pattern as above, with the only exception being that the packet capture tool pipes its capture through to an external server address. Figure 14 charts this process.

Figure 14: Device forwarding location data to Traccar Server and an external server



### System Architecture Diagram

Figure 15: Layers and Packages

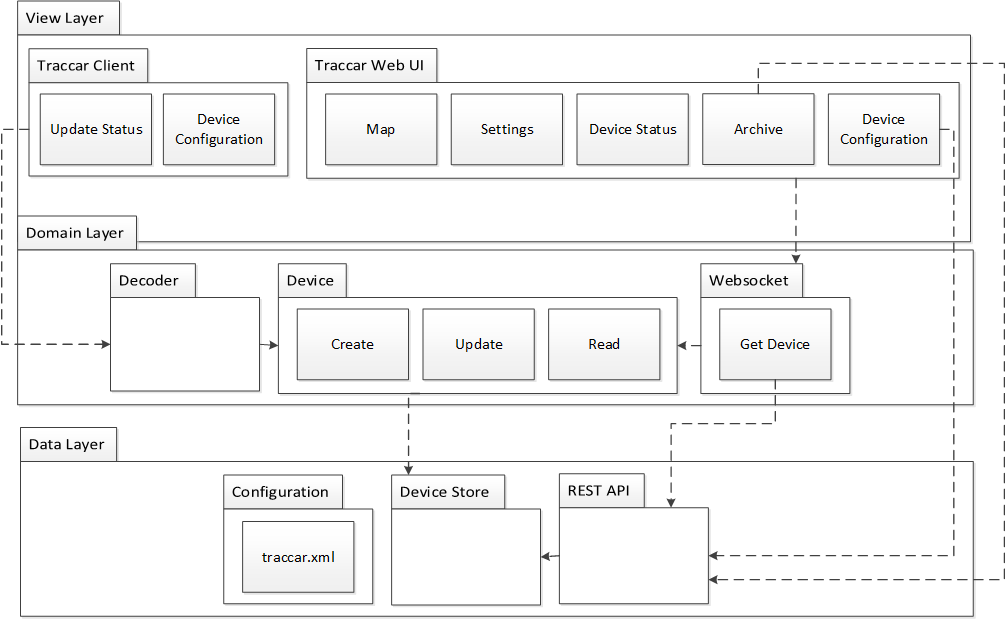


Figure 15 provides an overview of the layers and packages that comprise of the complete system. The view layer represents both Traccar Client and Traccar Web UI. Traccar Server is embodied by the domain and data layer.

## System Design

### Database Design

See section 6.3.

#### Data Objects and Resultant Data Structures

See section 6.1.

### Data Conversion

Not Applicable

### External Tools (APIs)

Traccar Web UI implements a simplified version of a RESTful API, which can be accessed by third-party applications.

The prefix for all method calls is '/traccar/rest' and all requests must be prefaced by an initial login:

* http://<SERVER IP ADDRESS>:<SERVER PORT>/traccar/rest/login?payload=["<USERNAME>","<PASSWORD>"]

Once logged in, all methods in the DataService interface class are exposed by the API.

The API supports both POST and GET methods - POST requests should contain method arguments serialised in a JSON array within the request body, while GET requests should provide a payload parameter with a serialised JSON array as the value. Method call responses are serialised in JSON and each response will return a HTTP status code.

The methods available in the DataService interface class include the following:

* User authenticated()
* User login(String login, String password, boolean passwordHashed)
* User login(String login, String password)
* boolean logout()
* User register(String login, String password)
* List<User> getUsers()
* User addUser(User user)
* User updateUser(User user)
* UserSettings updateUserSettings(UserSettings userSettings)
* User removeUser(User user)
* List<Device> getDevices()
* Device addDevice(Device device)
* Device updateDevice(Device device)
* Device removeDevice(Device device)
* Map<User, Boolean> getDeviceShare(Device device)
* void saveDeviceShare(Device device, Map<User, Boolean> share)
* List<Position> getPositions(Device device, Date from, Date to, boolean filter)
* List<Position> getLatestPositions()
* ApplicationSettings getApplicationSettings()
* void updateApplicationSettings(ApplicationSettings applicationSettings)
* void saveDefaultUserSettigs(UserSettings userSettings)
* UserSettings getDefaultUserSettings()
* void saveRoles(List<User> users)
* List<GeoFence> getGeoFences()
* GeoFence addGeoFence(GeoFence geoFence)
* GeoFence updateGeoFence(GeoFence geoFence)
* GeoFence removeGeoFence(GeoFence geoFence)
* Map<User, Boolean> getGeoFenceShare(GeoFence geoFence)
* void saveGeoFenceShare(GeoFence geoFence, Map<User, Boolean> share)
* String sendCommand(Command command)

An additional method was also added at the request of the project sponsor, which allows the user to access all location data messages in TLT-2H protocol format:

* List<String> getAllMessages(Date from, Date to)

An example of its usage follows:

* http://<SERVER IP ADDRESS>:<SERVER PORT>/traccar/rest/getAllMessages?payload=[<FROM>,<TO>]
* http://<SERVER IP ADDRESS>:<SERVER PORT>/traccar/rest/getAllMessages?payload=[%222015-01-01%2000:00:00%20GMT%22,%222018-01-01%2023:59:59%20GMT%22]

### User Interface Design

A Before and After document was created to illustrate the differences in the Traccar Client and Traccar Web UI interface before and after development. The primary goal of the interface design was to meet all the requirements of the project sponsor, while also removing any interface options that were in excess of requirements. The notable deletions from the interface were vehicle maintenance records, notifications and sensor information. A copy of the Before and After document can be found in Appendix D.

Appendix A: Record of Changes

Table 43 - Record of Changes

| Version Number | Date | Author/Owner | Description of Change |
| --- | --- | --- | --- |
| 1.0 | 17 October 2017 | Danielle Heinrich | First draft |

Appendix B: Acronyms

Table 44 - Acronyms

| Acronym | Literal Translation |
| --- | --- |
| CPU | Central Processing Unit |
| DST Group | Defence Science and Technology Group |
| GPS | Global Positioning Satellite |
| NSID | National Security and Intelligence, Surveillance and Reconnaissance Division |
| PCAP | Packet Capture |
| REST | Representational State Transfer |
| TCP | Transmission Control Protocol |
| TLT-2H | TLT-2H |
| UDP | User Datagram Protocol |
| UTC | Coordinated Universal Time |
| WADL | Web Application Description Language |

Appendix C: Glossary

Table 45 - Glossary

| Term | Acronym | Definition |
| --- | --- | --- |
| Central Processing Unit | CPU | The key component of a computer system, which contains the circuitry necessary to interpret and execute program instructions. |
| Defence Science and Technology Group | DST Group | The Australian government's lead agency responsible for applying science and technology to safeguard Australia and its national interests. |
| Global Positioning Satellite | GPS | A radio navigation system that allows land, sea, and airborne users to determine their exact location, velocity, and time 24 hours a day, in all weather conditions, anywhere in the world. |
| National Security and Intelligence, Surveillance and Reconnaissance Division | NSID | The division within DST Group that the project sponsor falls into. |
| Packet Capture | PCAP | A protocol for Internet communication that allows a computer or device to receive incoming signals from another device and convert those signals into usable information. |
| Representational State Transfer | REST | A way of providing interoperability between computer systems on the Internet. |
| Transmission Control Protocol | TCP | A standard that defines how to establish and maintain a network conversation via which application programs can exchange data. |
| TLT-2H | TLT-2H | A GPS communication protocol |
| User Datagram Protocol | UDP | An alternative communications protocol to TCP used primarily for establishing low-latency and loss tolerating connections between applications on the Internet. |
| Coordinated Universal Time | UTC | The time standard commonly used across the world. |
| Web Application Description Language | WADL | Designed to provide a description of HTTP-based Web applications that can be processed by machines. |

Appendix D: Referenced Documents

Table 46 - Referenced Documents

| Document Name | Document Location and/or URL | Issuance Date |
| --- | --- | --- |
| Technology Scan and Feasibility Analysis | https://github.com/Danielle-Heinrich/ICT-Project/raw/master/Technology%20Scan%20and%20Feasibility%20Analysis.docx | 24/08/2017 |
| System Requirements | https://github.com/Danielle-Heinrich/ICT-Project/raw/master/System%20Requirements%20V2.0.docx | 26/09/2017 |
| Before and After | https://github.com/Danielle-Heinrich/ICT-Project/raw/master/Before%20and%20After.docx | 05/10/2017 |

DST Group 2016, Defence Science and Technology Strategic Plan 2013-2018, Australian Government, viewed 30 July 2017, <https://www.dst.defence.gov.au/sites/default/files/basic\_pages/documents/dst-strategic-plan-2016-update.pdf>.

DST Group 2016, National Security and Intelligence, Surveillance and Reconnaissance Division Strategic Plan 2016-2020, Australian Government, viewed 30 July 2017, <https://www.dst.defence.gov.au/sites/default/files/publications/documents/National%20Security%20and%20ISR%20S%26T%20Strategy%202016-2020.pdf>.

DST Group 2017, *Information Integration,* Defence Science and Technology Group, Australian Government, viewed 09 August 2017, < https://www.dst.defence.gov.au/capability/information-integration>.

Smithsonian 2017, *Blue Force Tracker System,* Time and Navigation, Smithsonian, viewed 30 July 2017, <https://timeandnavigation.si.edu/multimedia-asset/blue-force-tracker-system>.

Smithsonian National Air and Space Museum 2012, *The Blue Force Tracker System,* video, YouTube, 5 April 2013, viewed 30 July 2017, <https://www.youtube.com/watch?v=gsjl\_bKuc\_o>.

Wikipedia 2017, *Blue Force Tracking,* Wikipedia, viewed 30 July 2017, <https://en.wikipedia.org/wiki/Blue\_Force\_Tracking>.

Appendix E: Approvals

The undersigned acknowledge that they have reviewed the SDD and agree with the information presented within this document. Changes to this SDD will be coordinated with, and approved by, the undersigned, or their designated representatives.

Not Applicable

Table 47 - Approvals

| Document Approved By | Date Approved |
| --- | --- |
| Name: <Name>, <Job Title> - <Company> | Date |
| Name: <Name>, <Job Title> - <Company> | Date |
| Name: <Name>, <Job Title> - <Company> | Date |
| Name: <Name>, <Job Title> - <Company> | Date |

Appendix F: Additional Appendices

Not Applicable

Appendix G: Declaration of Contribution

The following is a declaration of your individual contributions towards this group assessment. If any contribution does not meet the assessment requirements, the course coordinator may adjust individual marks up or down, depending on the level of contribution made.

**Team Member 1**

Name: Danielle Heinrich

I contributed 100% towards this assessment.

I worked on all sections/questions.