Deliverable #1 Template : Software Requirement Specification (SRS)

SE 3A04: Software Design II – Large System Design

Tutorial Number: T03
Group Number: G6
Group Members:

- Cass Braun
- Nehad Shikh Trab
- Savvy Liu
- Tvesha Shah
- Victor Yu

IMPORTANT NOTES

- Be sure to include all sections of the template in your document regardless whether you have something to write for each or not
  - If you do not have anything to write in a section, indicate this by the *N/A*, *void*, *none*, etc.
- Uniquely number each of your requirements for easy identification and cross-referencing
- Highlight terms that are defined in Section 1.3 (**Definitions, Acronyms, and Abbreviations**) with **bold**, *italic* or <u>underline</u>
- For Deliverable 1, please highlight, in some fashion, all (you may have more than one) creative and innovative features. Your creative and innovative features will generally be described in Section 2.2 (**Product Functions**), but it will depend on the type of creative or innovative features you are including.

1 Introduction

This SRS will provide visibility over software requirements for Gaim, an AI-powered wildlife identification application. This document will discuss the purpose of Gaim, the scope of the application, its objectives, production description, any constraints, user characteristics, product requirements, and use case diagrams.

1.1 Purpose

- Specify the purpose of the SRS.
  inform public about wildlife
- Specify the intended audience for the SRS.
  Nature enthusiats
  hunters - 1.27 mill canadians hunted, 1.4 mill sport shooting

1.2 Scope

1.2.1 Software Products

1.2.1.1 Decision-Tree Expert (Mammal, Bird, and Fish domain)

This expert gives guided questions that the user can answer. At the end, based on the answers given by the user, generates and gives to the forum its best guess at what the animal could be based on its domain. It also gives an accuracy score to the forum. If it has no animal in its database which meets the required attributes, it returns no answer.

1.2.1.2 Image Expert

Based on a user-uploaded image, this expert generates and gives to the forum its best guess on what the animal could be. At the same time, it generates an accuracy score. If the accuracy score is too low, it returns nothing to the forum.

1.2.1.3 Freeform Writing Expert

Based on user-given freeform text, this expert generates and returns to the forum its best guess to as what the animal could be. At the same time, it generates an accuracy score. If the accuracy score is too low, it returns nothing to the forum.

1.2.1.4 Animal Characteristics DBMS

This service maintains a list of species in Ontario including their attributes, phylogenetic classification, and their at-risk status. It allows other classes to sort and access this information.

1.2.1.5 Forum

The forum accepts responses from the experts as their guesses of what the animal could be, as well as an accuracy score. Based on these accuracy scores, it selects an answer to present to the user.

1.2.1.6 Animal Report Generator

The animal report generator receives the answer from the Forum and generates a report to show to the user about the animal.

1.2.2 Applications

1.2.2.1 Benefits

This product will allow users to be able to more easily identify animals found in Ontario. It will then give them relevant information such as habitat, circadian rhythms, and at-risk status. It will allow users to make informed decisions about their actions surrounding the identified animal.

1.2.2.2 Objectives

This software will reliably generate a report based on any of the available input methods. It will be secure. This product will efficiently determine what animal the user is asking about based on any of the available input methods.

1.2.2.3 Goals

- This product will generate an answer within 10 seconds of completion of user input
- This product will be complete by
- This product will always generate an answer with an accuracy score greater than 90%
- Identify the software product(s) to be produced, and name each (e.g., Host DBMS, Report Generator, etc.)

- Explain what the software product(s) will do (and, if necessary, also state what they will not do).
- Describe the application of the software being specified, including relevant benefits, objectives, and goals.

1.3 Definitions, Acronyms, and Abbreviations

- Provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS.
- This should be in alphabetical order.

1.4 References

- Provide a complete list of all documents referenced elsewhere in the SRS.
- Identify each document by title, report number (if applicable), date, and publishing organization.
- Specify the sources from which the references can be obtained.
- Order this list in some sensible manner (alphabetical by author, or something else that makes more sense).

1.5 Overview

- Describe what the remainder of the document/SRS contains.
  (e.g. "Section 2 discusses...Section 3...")

2 Overall Product Description

- This section should describe the general factors that affect the product and its requirements.
- It does not state specific requirements.
- It provides a *background* for those requirements and makes them easier to understand.

2.1 Product Perspective

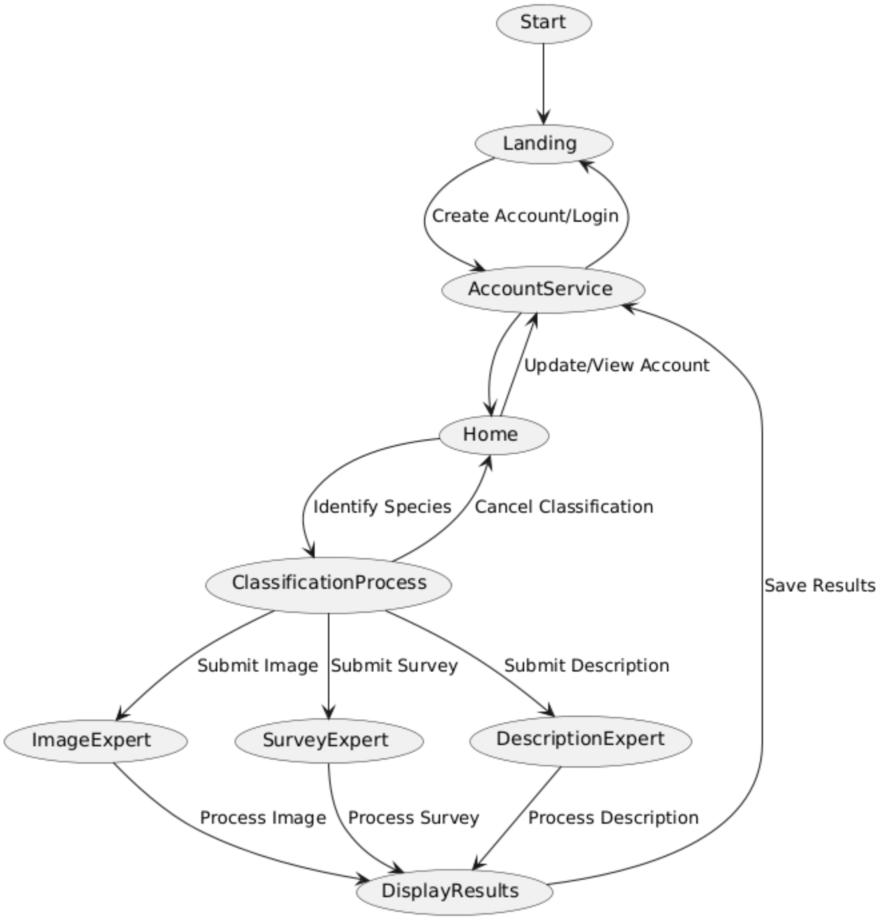
- Put the product into perspective with other related products, i.e., context
- If the product is independent and totally self-contained, it should be stated here
- If the SRS defines a product that is a component of a larger system, then this subsection should relate the requirements of that larger system to the functionality of the software being developed. Identify interfaces between that larger system and the software to be developed.
- A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful

2.2 Product Functions

- Provide a *summary* of the major functions that the software will perform.
  - **Example**: An SRS for an accounting program may use this part to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those functions requires.
- Functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time
- Present the functions in a list format - each item should be one function, with a brief description of it
- Textual or graphical methods can be used to show the different functions and their relationships
  - Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables

The goal of the system is to identify endangered and out of season species for 3 main domains: aerial, aquatic, and land. There will be 3 main methods for identifying specifics and a backend decision tree to determine accuracy and provide a result. The methods will be: description, survey and image. The main modules of this app will be the following: Identifying species (submitting photo, writing a description, filling a questionnaire, getting result, viewing accuracy score), Account Services (creating an account, viewing past inquiries, updating information, logging in), Community Discussion (asking questions, posting pictures, liking other users posts in the hunting community).

Modules	Functions
Identify Species: Image	Write Description: Allows the user to submit a description of the animal they have seen View Result: Allows user to see the identified species and corresponding accuracy score Save Result: Allows user to store output
Identify Species: Description	Submit Image: Allows the user to upload an image of animals they see View Result: Allows user to see the identified species and corresponding accuracy score Save Result: Allows user to store output
Identify Species: Survey	Fill out Questionnaire: Allows user to submit inquiry based on predesigned questions View Result: Allows user to see the identified species and corresponding accuracy score Save Result: Allows user to store output
Account Services	Create Account: Allows the user to create an account Update Account: Allows the user to update their account information Login: Allows user to login/logout of their account Search History: Allows user to view past searches



2.3 User Characteristics

2.3.1 Hunters

Hunters primarily live in rural areas, sometimes with limited access to educational resources and diverse perspectives. For this reason, they have a majority lower education level as those who leave behind their provincial life to gain education in the city may decide to stay there and live behind their rural life of hunting. They have much experience with wildlife and its behaviours, many having learned hunting from their parents who in turn learned it from their parents before them. They have, in general, a deep respect for wildlife and a vast knowledge of species and their behaviours. However, they may be limited in their knowledge of which species are currently designated to be at risk.

2.3.2 Nature Enthusiasts

Nature enthusiasts cover a broad spectrum of people from casual national park-goers to experienced portagers. Their education level varies from those who have not graduated high school to those with PhDs and MDs. However, in Canada it tends towards those with undergraduate degrees as that is the most common education level. Their experience level varies greatly, however it would tend towards those who camp once or twice a year. Potentially, they have attended a seminar run by the national parks and are able to identify a small subset of species. In majority, our user can identify broad spectrums of species (such as moose, deer, squirrel, bird), but is unable to tell the different nuances between species.

- Describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise

- Since there will be many users, you may wish to divide into different user types or personas

2.4 Constraints

1. **Data Availability:** The accuracy of Gaim depends on an up-to-date and comprehensive database of wildlife species, including their characteristics and at-risk status. Obtaining and maintaining such a database may require collaboration with wildlife organizations and adherence to licensing restrictions. Especially since this is supposed to be Ontario-specific, it will be difficult to obtain such a database.

2. **Internet Dependency:** Even though some features can be used locally, features such as cloud-based image processing, AI model updates, and community discussions require an active internet connection. Users in remote locations with limited to no internet connectivity may experience reduced to no functionality.

3. **Regulatory and Ethical Constraints:** Since the application deals with wildlife identification, it must comply with Canadian environmental laws and conservation policies. The app must ensure that it does not inadvertently promote illegal activities such as hunting endangered species. Even though the app will provide endangerment status, the list will need to be up-to-date and may require regular checks to stay updated.

4. **AI Model Accuracy and Bias:** The AI models used for species identification must be trained with diverse datasets to avoid misclassifications. Ensuring high accuracy while maintaining real-time performance is a key challenge, as incorrect identifications may mislead users. This can be especially critical with endangered species and protected species.

5. **Scalability:** As Gaim gains more users, the system must be capable of handling increased data processing demands and community interactions. Future versions may require cloud infrastructure scaling, which could introduce additional costs.

2.5 Assumptions and Dependencies

- Audio recordings are clear and free of excessive background noise.

- Images and videos are of sufficient resolution, clarity, and lighting for accurate identification.

- The app will only be used in Canada (no need to consider species from other regions).

- We have access to cloud or local computational resources to handle image and sound processing.

- Assume APIs are secure and functional

- Assume user has internet connection

2.6 Apportioning of Requirements

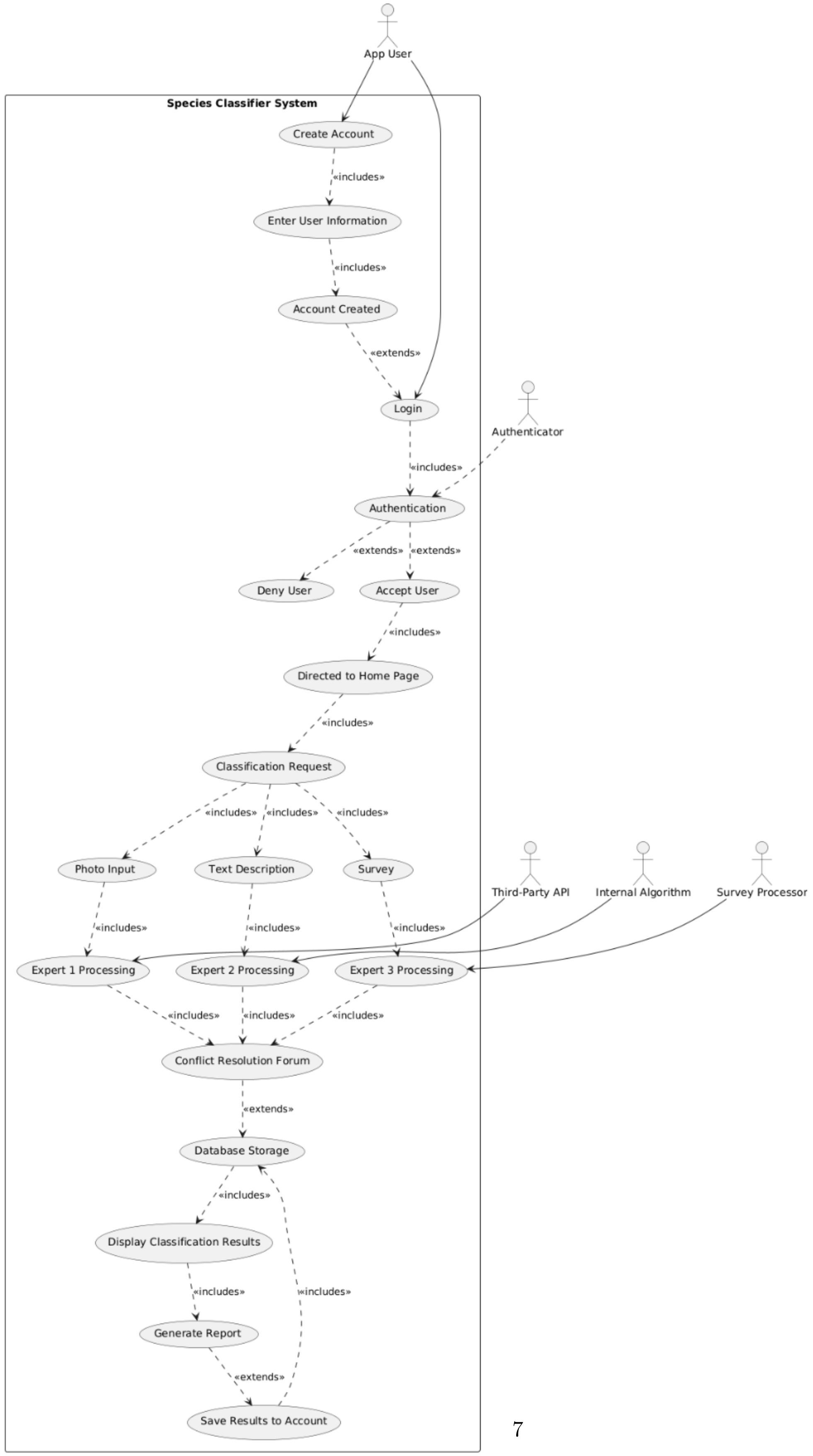
- The first version of the app will not take into consideration animals whose appearances change by season.

- The first version of the app will be developed in English.

- Verify with wildlife experts to check accuracy of our results.

- Expand database to take into account species from other regions as the app grows.

3 Use Case Diagram



- Provide the use case diagram for the system being developed.
- You do not need to provide the textual description of any of the use cases here (these will be specified under "Highlights of Functional Requirements").

4 Highlights of Functional Requirements

- Specify all use cases (or other scenarios triggered by other events), organized by Business Event.
- For each Business Event, show the scenario from every Viewpoint. You should have the same set of Viewpoints across all Business Events. If a Viewpoint doesn't participate, write N/A so we know you considered it still. You can choose how to present this - keep in mind it should be easy to follow.
- At the end, combine them all into a Global Scenario.
- Your focus should be on what the system needs to do, not how to do it. Specify it in enough detail that it clearly specifies what needs to be accomplished, but not so detailed that you start programming or making design decisions.
- Keep the length of each use case (Global Scenario) manageable. If it's getting too long, split into sub-cases.
- You are *not* specifying a complete and consistent set of functional requirements here. (i.e. you are providing them in the form of use cases/global scenarios, not a refined list). For the purpose of this project, you do not need to reduce them to a list; the global scenarios format is all you need.
- Red text below is just to highlight where you need to insert a scenario - don't actually write it all in red.

Main Business Events: List out all the main business events you are presenting. If you sub-divided into smaller ones, you don't need to include the smaller ones in this list.

Viewpoints: List out all the viewpoints you will be considering.

Interpretation: Specify any liberties you took in interpreting business events, if necessary.

BE1. Business Event Name #1

VP1. Viewpoint Name #1
Insert Scenario Here

VP2. Viewpoint Name #2
Insert Scenario Here

Global Scenario:
Insert Scenario Here

BE2. Business Event Name #2

VP1. Viewpoint Name #1
Insert Scenario Here

VP2. Viewpoint Name #2
Insert Scenario Here

Global Scenario:
Insert Scenario Here

5 Non-Functional Requirements

- For each non-functional requirement, provide a justification/rationale for it.
  Example:
  SC1. *The device should not explode in a customer's pocket.*
  Rationale: Other companies have had issues with the batteries they used in their phones randomly exploding [insert citation]. This causes a safety issue, as the phone is often carried in a person's hand or pocket.
- If you need to make a guess because you couldn't really talk to stakeholders, you can say "We imagined stakeholders would want...because..."
- Each requirement should have a unique label/number for it.
- In the list below, if a particular section doesn't apply, just write N/A so we know you considered it.

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF-A1.

5.1.2 Style Requirements

LF-S1.

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH-EOU1.

5.2.2 Personalization and Internationalization Requirements

UH-PI1.

5.2.3 Learning Requirements

UH-L1.

5.2.4 Understandability and Politeness Requirements

UH-UP1.

5.2.5 Accessibility Requirements

UH-A1.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR-SL1. The response time of all external APIs (expert calls, database updates, report generation, etc.) should be within 0.5 seconds. **Rationale:** The average API response time is between 100ms to 500ms [1] and this is known as the typical expectation of performance. A response time in this range is expected by industry standards and is required for a competitive user experience. We would ideally want limited delays, to promote efficiency and satisfaction, and this API response time will ensure that.

PR-SL2. App start-up time should be around 2 seconds. **Rationale:** The average start-up time in the industry is around 1.5 to 4 seconds [2]. This is the time frame that most systems and users typically wait for. Creating an application that does not fit these requirements will cause us to lose user interest.

PR-SL3. Account updates should be within 3 seconds. **Rationale:** When a user updates their account to save a report, update personal data or look for items previously stored our system should efficiently fetch this information from our database. This is the most time it should take for old information to be requested, searched and displayed on the user screen [3].

5.3.2 Safety-Critical Requirements

PR-SC1. The app should provide safe suggestions for user reports. **Rationale:** Since our app targets natural areas and focuses on the hunting community recommendations we give during our report generation must be as safe as possible. Focus there is always room for the unexpected but without safety, we lose the trust of our users.

5.3.3 Precision or Accuracy Requirements

PR-PA1. The app must be accurate in image classification around 85 percent. **Rationale:** The goal of this app is to ensure species are correctly identified. While this application can be used by a diverse demographic, the hunting community is a large potential user base. For their specific needs, it is essential to ensure accuracy when looking at animal classification based on submitted images. The goal of our product should aim for the current industry standard which is 85 percent [4].

PR-PA2. The app must be accurate in description classification by 80 percent. **Rationale:** Classification models that take in description input have an average 80 percent accuracy rate in responses [5]. This is something we should aim to uphold in the internal model to ensure we are consistent with industry standards.

5.3.4 Reliability and Availability Requirements

PR-RA1. The system must have an availability of 99.999 **Rationale:** 99.999 is a well-known figure for availability in the software industry [6]. An application or product should almost always be available to users and all updates and fixes should be done intelligently during off hours or in chunks (so the entire service is not unavailable all at once).

5.3.5 Robustness or Fault-Tolerance Requirements

PR-RFT1. The system must be able to handle unexpected inputs **Rationale:** If a user accidentally submits an invalid input or unexpected entry to any of our experts our product should be able to respond and prompt the user for more information. We need to ensure that expected crashes do not occur due to minor input errors.

PR-RFT2. The system must allow users to view saved reports even when there is no network. **Rationale:** While it may not always be possible to make new searches in areas with no network it should be possible to view previous searches made. Especially for those hunting wifi and network is not a 100 percent ensured guarantee so having that option to view previous searches is a huge advantage.

5.3.6 Capacity Requirements

PR-C1. The system must be able to 5 simultaneous users. **Rationale:** Ideally the system would be able to work with hundreds if not thousands of users at once. However, given the scope and time frame of the project, this number is more verifiable with the current team size.

5.3.7 Scalability or Extensibility Requirements

PR-SE1. The system must adhere to solid principles and design patterns. **Rationale:** To ensure the application is best set up for long-term growth we must adhere to correct solid principles and design patterns when designing and ultimately building this product. The goal would be to create an effective application that is low coupled, correctly implemented independent modules and open for extensibility in the future.

5.3.8 Longevity Requirements

PR-L1. The system must update species information every month (update new animals in the area, seasonality, etc.) **Rationale:** There are continuous discoveries, regulation changes and seasonal differences to keep in mind for species classification. It is essential that our database updates during key moments and ensures our lists of species and huntable species are accurate and up to date.

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE-EPE1.

5.4.2 Requirements for Interfacing with Adjacent Systems

OE-IA1. The system must send and receive information from the user to our backend database and external API experts (e.g. for image classification) **Rationale** : To have a well-functional appreciation our system must be able to communicate effectively based on user input and return an accurate response from external connectors.

5.4.3 Productization Requirements

OE-P1.

5.4.4 Release Requirements

OE-R1. The app must be compatible with Android 14 and above and Apple iOS 17 and above. **Rationale** : This will ensure the most range to reach the most number of users.

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS-M1.

5.5.2 Supportability Requirements

MS-S1.

5.5.3 Adaptability Requirements

MS-A1.

5.6 Security Requirements

5.6.1 Access Requirements

SR-AC1.

5.6.2 Integrity Requirements

SR-INT1.

5.6.3 Privacy Requirements

SR-P1.

5.6.4 Audit Requirements

SR-AU1.

5.6.5 Immunity Requirements

SR-IM1.

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

CP-C1.

5.7.2 Political Requirements

CP-P1.

5.8 Legal Requirements

5.8.1 Compliance Requirements

LR-COMP1.

5.8.2 Standards Requirements

LR-STD1.

A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.