

Software Requirements Specification

for

Electronic Project Submission Application

Environmental Forensics, Department of Planning, Industry and Environment

Version 0.1

Prepared by A Theuma, C Trinh, D Raman, L Lam, Z Yu

Apollo Consulting

26-08-2020

Table of Contents

1	Introduction	2
1.1	Purpose	2
1.2	Intended Audience and Reading Suggestions.....	2
1.3	Scope	2
1.4	Glossary.....	3
1.5	Document Conventions	3
1.6	References	4
2	Overall Description	4
2.1	Product Perspective	4
2.2	Product Functions	4
2.3	User Classes and Characteristics.....	5
2.4	Operating Environment	5
2.5	User Documentation.....	5
3	Requirements.....	6
3.1	Functional Requirements (System Features).....	6
3.1.1	Submit sample.....	6
3.1.2	Upload photo of a sample.....	6
3.1.3	Accessing old records.....	7
3.2	Design and Implementation Requirements.....	7
3.2.1	User Interfaces	7
3.2.2	Hardware Interfaces	7
3.2.3	Software Interfaces.....	7
3.2.4	Communications Interfaces	7
3.3	Usability Requirements.....	7
3.3.1	Performance Requirements.....	7
3.3.2	Safety Requirements.....	8
3.3.3	Security Requirements.....	8
3.3.4	Software Quality Attributes	8
3.4	Other Non-Functional Requirements.....	8
3.4.1	Firebase Database Performance.....	8
4	Appendix	9
4.1	Appendix A: Current System User Interfaces.....	9

1 Introduction

1.1 Purpose

This purpose of the Software Requirements Specification is to outline our plan to create a new lightweight, cross-platform application for the environmental field officers within the New South Wales Department of Planning, Industry and Environment to access and submit samples to a database on the go and in the office. The requirements specified by the users include a simple, easy-to-use application designed to be intuitive to people of all technological proficiencies, and safe and secure storage of data that can be revisited at any time. This document outlines requirements for version 0.1 of the application.

1.2 Intended Audience and Reading Suggestions

This project is a prototype/mock application for the entry and management of data for the New South Wales Department of Planning, Industry and Environment. The application is restricted to use by the department's personnel and development team members. This has been implemented based on the existing paper forms and screen captures of the application currently in use by a minority of staff.

Reading suggestions are shown below:

- Overall Description: the Department's personnel will need to become accustomed to the application to effectively replace the paper forms.
- System Features: testers will need to understand of the system features to develop effective and meaningful test cases and give useful feedback to the developers.
- External Interface Requirements: software developers will need to know the requirements of the application. The Department's personnel will also need to understand the external interface requirements for the ease-of-use purpose.
- Non-functional and Functional Requirements: the software developers will need to use this as a reference point to determine development goals.

1.3 Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here. An SRS that specifies the next release of an evolving product should contain its own scope statement as a subset of the long-term strategic product vision.>

Commented [AT1]: To be discussed – unsure as to what the corporate goals and business strategy is

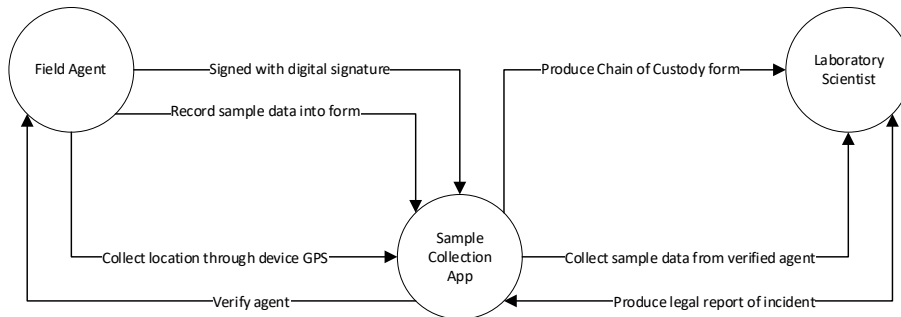


Figure 1 - Level 0 Data Flow Diagram

1.4 Glossary

The following table outlines, in alphabetical order, definitions for all acronyms, abbreviations, and technical terms specific to this project.

Term	Definition
Dart	Dart is a client-optimised programming language for apps on multiple platforms. It was developed and is maintained by Google.
DB	Database
(The) Department	The New South Wales Department of Planning, Industry and Environment
Flutter	Flutter is Google's UI toolkit for building mobile, web, and desktop applications.
Flutter Studio	This is an interactive UI development tool for Flutter. It is able to wire up the basic components and export the designs to Dart code.
Firebase Realtime Database	Firebase allows storing and syncing data with NoSQL cloud database. Data is synced across all clients in real time and remains available when the app goes offline. It offers cross-platform solutions: iOS, Android, and JavaScript SDKs.
Git	Git is an open-source distributed version control system designed to handle all sizes of software development projects with speed and efficiency. Git allows teams to collaborate on the same file (often code) without conflict.
Notion.so	This is the project management tool that provides database, Kanban boards, wikis, calendars, and reminders. It can also be used for knowledge management, note taking and data management.
SQL	Structured Query Language
SSO	Single-Sign-On
UI	User Interface

1.5 Document Conventions

This document is formatted as outlined in the IEEE 29148:2019 standard. Sections and subsections are identified by multi-level numbered lists and bold-faced text has been used to further emphasise these headings. Italicised text has been used to label and recognise diagrams. Hyperlinks are made clear in

blue text colouring. Lists within sections have been bulleted by a horizontal dashed line. Table header rows and columns are indicated in bold type.

1.6 References

List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.

The following are wholly referred to in this document:

- Balsamiq: <https://balsamiq.com/>
- Firebase Realtime Database: <https://firebase.google.com/docs/database/>
- Flutter: <https://flutter.dev/>
- Flutter Studio: <https://flutterstudio.app/>
- Notion.so: <https://www.notion.so>

2 Overall Description

2.1 Product Perspective

The application to be developed is a replacement measure for the paper-based system that has been used in the Department of Planning, Industry and Environment. The goal of the application is to reduce the use of paper in the process of collecting samples at the sites. It will be able to transfer the data of the collected samples from the sites quickly and securely to the laboratory. The application is expected to be able to run on multiple platforms: mobile, web browser, and desktop.

2.2 Product Functions

A high-level summary of the primary functions of the application are as follows:

- FE-1: Record data into the form (including location)
- FE-2: Log and submit the form
- FE-3: Verify users
- FE-4: Create Chain of Custody form with personnel's signature
- FE-5: Transfer from field to laboratory
- FE-6: Generate and export reports with provided form.

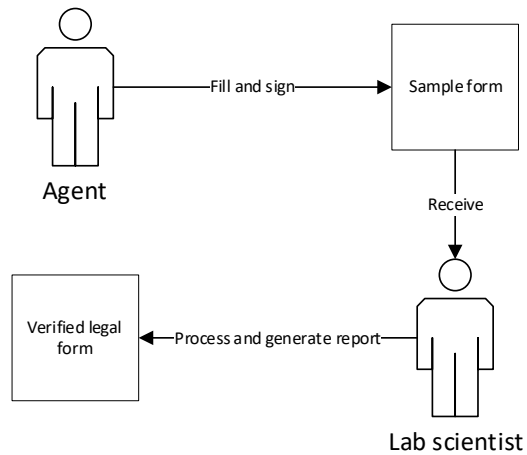


Figure 2- Use Case Diagram

2.3 User Classes and Characteristics

The following users are anticipated to make use of the application:

User	Characteristics
Field Agent	The Field Agent is the person who goes to the incident site to collect samples and record data into the collection form. They need to record time, location, and scientific data of the sample. The Chain of Custody is required to be signed before submission.
Lab Technician	The Lab Technician is the person who receives the data taken by the Field Agent to use for testing. Before obtaining the data, the technician is required to sign the Chain of Custody. The technician also generates the final report.

2.4 Operating Environment

- Mobile phone: iOS and Android OS
 - Smartphone with internet connection
 - Recommended storage is 8GB
 - Ideal screen size is 5"
- Web browser / desktop
 - Internet connection
 - Printer
 - Recommended internet browsers: Chrome, Firefox

2.5 User Documentation

Along with the software, we will also deliver the following items:

- User manual: this will be developed according to the IEEE Standard for Adoption of ISO/IEC 26514:2010

3 Requirements

3.1 Functional Requirements (System Features)

This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.

3.1.1 Submit sample

3.1.1.1 Description and Priority

This feature would allow all users to submit test samples into the database. The priority of this feature is high as this is the subject of the application.

3.1.1.2 Stimulus/Response Sequences

- 1) User chooses to start a new submission
- 2) User adds description to all fields relevant to current test
- 3) User uploads photo of sample
- 4) User submits the test sample
- 5) The system processes the submission
- 6) The system stores it into the database

3.1.1.3 Functional Requirements

- REQ-1: The system shall allow new submissions
- REQ-2: The system shall communicate via the Internet

3.1.2 Upload photo of a sample

3.1.2.1 Description and Priority

This feature would allow the users to upload photos of the individual samples they are submitting to the database. This would be a medium priority feature as it is an essential feature of the application, but it is not a time sensitive task to be focused on until other features are implemented first.

3.1.2.2 Stimulus/Response Sequences

- 1) User selects photo(s) to be uploaded
- 2) User uploads photo(s)
- 3) The system receives the photo(s)
- 4) The system displays the photo(s) as thumbnails

3.1.2.3 Functional Requirements

- REQ-1: The system shall accept image uploads
- REQ-2: The system shall be able to display images

3.1.3 Accessing old records

3.1.3.1 Description and Priority

This feature would allow the user to access previous uploaded and stored records of other samples that are submitted by themselves or by other people. This would be a low priority task as it must be developed first before a feature like this would be usable.

3.1.3.2 Stimulus/Response Sequences

- 1) User opens "Database" page
- 2) User searches for an old sample via queries such as the Sample ID number or keywords
- 3) The system would fetch all relevant entries
- 4) User selects specific entry they wish to view

3.1.3.3 Functional Requirements

- REQ-1: The system shall allow users to access old records
- REQ-2: The system shall be able to process search queries

3.2 Design and Implementation Requirements

3.2.1 User Interfaces

Effort will be put into this project to create an intuitive, simple, and user-friendly graphical interface for non-technical users.

See Appendix A for screen captures of the current user interface.

3.2.2 Hardware Interfaces

Provided in 2.4

3.2.3 Software Interfaces

The following are software components required by the application:

- Firebase Realtime Database: a cloud-hosted database where data is stored as JSON and synchronised in real time to connected users.
- Dart SDK: has libraries and command-line tools that helps developing Dart command-line, server, and non-Flutter web apps.
- Flutter SDK: will use Dart to develop UI for the multiple platform applications

3.2.4 Communications Interfaces

The following are communications functions required by the application:

- Realtime connection: instead of HTTP requests, the database uses data synchronisation.
- HTTP: The http package provides the simplest way to issue http requests. This package is supported on Android, iOS, and the web.

3.3 Usability Requirements

3.3.1 Performance Requirements

Photos of the evidence/sample can be taken and attached to the form that is submitted. Processing the original size of each photo for multiples photos may slow down the devices or even crash.

Compression is recommended in the case where high-quality photos are not required to improve the performance.

3.3.2 Safety Requirements

The data that is entered and processed through the application belong to the Department, thus it is required secure measure to ensure the integrity of the data. The authentication is required at every step of the process. It will use the Department authentication SSO for a more secure and managed log on.

3.3.3 Security Requirements

Authentication is required at every step of the process. It will use the Department's authentication for greater security. The data is only accessible by authorised account with cryptography.

3.3.4 Software Quality Attributes

The following are additional important quality characteristics for the application:

- Adaptability: The application is simple and intuitive so it should not be difficult to adapt it to user's needs.
- Portability: By using Flutter, it should be easily portable to and operating systems.
- Usability: the project is usable after the backend and database are successful developed.
- Maintainability: since the application is simple and does not require complex storage solution, the maintenance of the application is only to take care of the database.
- Reusability: even though the application's code is written in Dart/Flutter, which is not common programming language, the framework is made to be easy to develop and transform.

3.4 Other Non-Functional Requirements

3.4.1 Firebase Database Performance

Firebase Realtime Database is cloud-hosted database solution for the applications.

Expectations:

- Real time syncing for JSON data.
- Devices can collaborate with ease.
- Ability for offline use.
- Strong user-based security.

4 Appendix

4.1 Appendix A: Current System User Interfaces

The screenshot displays a digital form for entering sample information. At the top, a yellow header bar contains the text "Sample 1" and a "Delete sample" button with a close icon. Below the header, the form includes a date and time selection section with a date picker set to "14/08/2020", and dropdown menus for "Hour" (13) and "Minute" (50). A "GPS co-ordinates" section features input fields for "Latitude" and "Longitude", accompanied by a "Locate" button. A large text area labeled "Field results" is positioned below the coordinates. At the bottom, a section titled "Specific tests or test groups requested for Sample 1 (tick required tests)" includes links for "Edit tests list" and "Copy tests", an "Add test" button with a plus icon, and a final "Add sample" button with a plus icon.

Figure 3- Existing Digital Form (New Sample)

CHAIN OF CUSTODY

Submission Project and Chain of Custody Form

Fields marked with * are required

Sample bottles

Type of bottles *

☐ EPA
☐ Other

☐ Sealed *

Shipping container

Type of container *

☐ Esky
☐ White Plastic Box
☐ Other

Type of seal *

☐ EPA
☐ Other

Mode of Transport *

Condition on Depart

☐ Cooled
☐ Frozen
☐ Ambient

Preservation

☐ Alkali
☐ Acid
☐ Other

Sample Collector

☐ Sample bottle/s seal is/are intact *
☐ Shipping container seal is intact *

Representative's name *

Organisation/Employer *

Contact phone no. *

Collected (DD/MM/YYYY)

Hour

Minute

Signature *

Click to Sign

+

Add a sample handler

Go Back

Submit

Figure 4- Existing Digital Form (Chain of Custody)

10