# Software Requirements Specification

# for

# Test Repository of Agri Data (TReAD)

Version <0.1.1>

Prepared by

|  |  |  |
| --- | --- | --- |
| Abraham Darius Llave |  | [adsllave@outlook.com](mailto:adsllave@outlook.com) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
| Date: | 07 Dec 2016 |

Contents

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgements 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3 Specific Requirements 5

3.1 External Interface Requirements 5

3.1.1 User Interfaces 5

3.1.2 Hardware Interfaces 5

3.1.3 Software Interfaces 5

3.1.4 Communications Interfaces 5

3.2 Functional Requirements 6

3.3 Behaviour Requirements 6

3.3.1 Use Case View 6

4 Other Non-functional Requirements 7

4.1 Performance Requirements 7

4.2 Safety and Security Requirements 7

4.3 Software Quality Attributes 7

5 Other Requirements 8

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 0.0.1\_init-rough-draft | Llave, Abraham Darius S. | Initial release | 07/DEC/2016 1633 |

# Introduction

## Document Purpose

The purpose of this document is to provide the software requirement specification report for the TReAD software, outlining documentation in both short and highly technical forms.

## Product Scope

The TReAD is a simple web-based application that enables users to create, retrieve, update and delete rice research survey data on a relational database hosted online/on the “cloud”. In addition, geographic visualization with respect to where these were sourced is also offered.

## Intended Audience and Document Overview

This project is a test as part of the recruitment process for Specialist – Web Development position with closing date 21 Nov 2016, of the International Rice Research Institute (IRRI). As such, the intended audience are the following – staff under Staffing Services of its Human Resources Services Department (HRS), project leader and some panelists who have technical background in web development/software engineering/programming.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| AJAX | Asynchronous JavaScript and XML |
| CSS | Cascading Style Sheets |
| DB | Database |
| ERD | Entity-Relationship Diagram |
| HTML | HyperText Transfer Mark-up Language |
| HTTP | HyperText Transfer Protocol |
| HTTPS | Secure HTTP |
| JSON | JavaScript Object Notation |
| PostgreSQL | PostgreSQL Relatonal Databse Management System |
| REST | Representational State and Transfer |
| SQL | Server Query Language |
| W3C | World-Wide Web Consortium |
| xHTML | Extensible HTML |
| XML | Extensible Mark-up Language |

## Document Conventions

TBA

## References and Acknowledgements

TBA

# Overall Description

## Product Perspective

The project is a web application for use mainly over the public Internet over the HTTP(S) protocol, to be utilized for the creation, retrieval, update and deletion of agricultural research survey data involving <fields> as well as providing a feature for geographical visualization of where these were sourced. It aims to let users easily and efficiently manage such data and access to the rendered visualization using common web browsers found in various fairly-aged devices as of late 2016.

## Product Functionality

1. Create, update, retrieve, manage and delete users.
2. Create, update, retrieve and delete survey data.
3. Offer geographic visualization on where the data were sourced. Levels can be whole ASEAN, by country, by district.
4. Retrieving data involves a filtering facility/ies where we data being sought can be specified by country, province, among other details.

## Users and Characteristics

* Administrator

This user can modify or delete accounts.

* Uploaders

This user mainly deals with the creation, modification or deletion of survey data.

* Public

All users who can access the web application without authentication for the purposes of being identified as one or more of the preceding user types. These only have the privilege of retrieving data.

## Operating Environment

The system is intended to operate using the predominant client/server architecture for a web-based application. There would be a single physical and virtual server, and database. The server should be able to host at least Apache web server version 2.2.17; PHP Scripting Language version 7 and PostgreSQL Relational Databse Management System version 9.3. For the client side, a web browser supporting around 90% of W3C-HTML5 specification as of 2013, with JavaScript and Cookies enabled should be used to avail of client features as well as conform to modern web design principles such as Responsive Web Design and accessibility. Browsers dating earlier than that are not guaranteed to work with the application whether in part or in full.

## Design and Implementation Constraints

1. For the most part, interaction between the logic and database side of the application will be via another application subset dedicated for the purpose, utilizing the RESTful services principle.
2. A simple (virtual and/or physical server) with at least 384MB of random access memory, free long-term storage space of at least 1GB and generic x86 or x86\_64 processor with <FLOPS performance> and guaranteed at least 10Mbps Internet connection can suffice for the purpose. There would be no need for advanced features such as load balancing, as of the moment.
3. The “LAPP” (Linux, Apache, PHP, PostgreSQL) stack will be used for the whole suite of application.

## User Documentation

During turn-over of the software, there will be at least a user manual in PDF format to be turned over to the appropriate person, in addition to other possible means like wiki pages. The author will be also available to provide further explanation/reasonable updates or fixes as requested.

## Assumptions and Dependencies

1. “Uploader” users have limited to novice computer skills.
2. JavaScript and Cookies should be enabled in the user’s web browser.
3. Supports W3C-HTML5 standards of late 2013.
4. Internet connection between client and server is quite decent, at least 30 KB/sec with minimal variation.
5. Application is not meant for large-scale use, i.e., requiring advanced technologies such as load balancers on the server-side.
6. Data can be viewed by the public without authorization.

# Specific Requirements

## External Interface Requirements

### User Interfaces

TBA

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., Cancel) that will appear on every screen, error message display standards, and so on. Define the software components for which a user interface is needed.

TO DO: The least you can do for this section is to describe in words the different User Interfaces and the different screens that will be available to the user. Those who will be able to provide optional Graphical User Interface screenshots, will be rewarded by extra marks.>

### Hardware Interfaces

TBA

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.

TO DO: Please provide a short description of the different hardware interfaces. If you will be using some special libraries to communicate with your software mention them here. In case you have more than one hardware interface divide this section into subsections.>

### Software Interfaces

TBA

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>

### Communications Interfaces

TBA

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >

## Functional Requirements

TBA

*< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions with specific explanations regarding every function.*

*TO DO: Brake the functional requirements to several functional areas and divide this section into subsections accordingly. Provide a detailed list of all product operations related to these functional areas.*

## Behaviour Requirements

### Use Case View

TBA

<A use case defines a goal-oriented set of interactions between external actors and the system under consideration. Since sometimes we will not be able to specify completely the behaviour of the system by just State Diagrams, we use use-cases to complete what we have already started in section 3.3.1.

TO DO: Provide a use case diagram which will encapsulate the entire system and all possible actors. Do not include detailed use case descriptions (these will be needed when you will be working on the Test Plan), but make sure to include a short description of what every use-case is, who are the actors in your diagram. For more information please refer to your UML guide>

# Other Non-functional Requirements

## Performance Requirements

TBA

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide at least 5 different performance requirements based on the information you collected from the client. For example you can say “1. Any transaction will not take more than 10 seconds, etc…>

## Safety and Security Requirements

TBA

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

* Provide at least 3 different safety requirements based on your interview with the client or, on your related research, and again you need to be creative here.
* Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>

## Software Quality Attributes

TBA

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, & etc…Do not forget to include such attributes as the design for change. Please note that you need to include at least 2 quality attributes, but it is the mere minimum and it will not receive the full marks.>

# Other Requirements

<This section is **Optional.** Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

TBA

Appendix A – Data Dictionary

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist your Lab Instructor to determine the effort put forth to produce this document>