Demonstration System Specification (draft)

# Introduction

The int

## Purpose

The purpose of this document is to provide a reference point and scoping model for a demonstration version of the Protocol and eCRF Build Tool. It will also outline the scalability of the system.

## Scope

1. Identify
2. Explain what the software product(s) will, and, if necessary, will not do
   1. Will be a one way tool to develop and maintain the integrity of the eCRFs for a clinical trail
   2. The demonstration build will be fully scalable to allow
3. Will not do:
   1. Store any confidential data
4. Describe the application of the software being specified. As a portion of this, it should:

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **CDISC** |  |
| **ODM** |  |
| **CDASH** |  |
| **SDM** |  |

## References

1. CDISC, page 1 lsdjflsdjflkjsdflksjd

## Overview

1. Describe what the rest of the SRS contains
2. Explain how the SRS is organized.

# General Description

This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.

## Product Perspective

This subsection of the SRS puts the product into perspective with other related products or projects. (See the IEEE Guide to SRS for more details).

## Product Functions

The functions that the software will perform:

* Capture the system requirements for export to other electronic data capture tool;

## User Characteristics

Principle Investigator/Researcher

* Has responsibility for writing the protocol and developing the study.
* Will be able to log in
* Will be able to export the build to the available EDC system
* Will have access to the library of eCRFs built by administration

Administration

* Has responsibility for maintaining the datasets, items and items groups

## General Constraints

This subsection of the SRS should provide a general description of any other items that will limit the developer’s options for designing the system. (See the IEEE Guide to SRS for a partial list of possible general constraints).

## Scalability

### Future Development

The primary function of the tool is to establish the system requirements for capturing study data. The tool is built around a fixed set of variables that can be utilised in numerous datasets to provide a limitless supply of eCRFs. The tool takes the key elements of a study to outline the structure, workflow and timing of the subsequent eCRFs.

You essentially have the workings of a system that can develop other data capture tools. Numerous applications (clinical or non-clinical) could be developed. The protocol could be seen as the administration element for other applications to build from.

## Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

# Specific Requirements

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.

Each requirement in this section should be:

1. Correct
2. Traceable (both forward and backward to prior/future artefacts)
3. Unambiguous
4. Verifiable (i.e., testable)
5. Prioritized (with respect to importance and/or stability)
6. Complete
7. Consistent
8. Uniquely identifiable (usually via numbering like 3.4.5.6)

Attention should be paid to the carefully organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

## External Interface Requirements

### User Interface

### Hardware Interface

### Software Interface

### Communications Interface

## Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

### Functional Requirements or Features #1

#### Introduction

#### Input

#### Procession

#### Outputs

#### Error Handling

### Use Cases

#### Use Case #1

#### Use Case #2

### Classes / Objects

## Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

### Availability

* The system will be available to all clinical researchers

### Security

The system will hold no clinical information;

The main security features will need to be abdicate to :

* + Sensitivities surrounding corporate drug companies
  + User details for

### Maintainability

Maintenance will initially be high – the first few months will require the construction of the items and item groups that form the basis of the library. This will require some level of validation and testing.

It is expected the continued maintenance will be low – Once the primary items, item groups and forms have been constructed updates will be minimal;

New items and item groups will be study dependant;

## Design Constraints

Export –

## Database Requirements

Will a database be used?

## Audit Trail

The system will require a detailed audit trail. As studies are developed strict version controls will be required to maintain the validations.

## Help Desk Support

## Administration Features

## Output Requirements

### eCRF applications

### Analysis

## Training

## Licences

## Data Integrity

## Test and Evaluation

## Accessibility and User Service

## Data Validation

## Other Requirements

# Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.

## Sequence Diagrams

## Data Flow Diagrams

## State Transition Diagrams

# Change Management

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

# Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.

Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.