1. Introduction

1.1 Purpose

*Identify the purpose of this SDD and its intended audience. (e.g. “This software design document describes the architecture and system design of XX. ….”).*

The purpose of this software design document is to describe the architecture and system design of the Environmental Data Collection & Recording Agent (EDCRA). It is intended for software designers seeking to write, implement or extend the system.

1.2 Scope

*Provide a description and scope of the software and explain the goals, objectives and benefits of your project. This will provide the basis for the brief description of your product.*

This product is an extension of CBSR's more primitive software, tempmon, which was implemented (hastily) to read and record temperatures from a single specific temperature transmitter. The goal of EDCRA is to remedy the shortcomings of tempmon, including its restrictive hardware components and user friendliness. The result of this product should be a widely accessible system usable on any platform and with a much wider selection of environmental monitoring equipment.

1.3 Overview

*Provide an overview of this document and its organization.*

This document is organized into eight sections: Introduction, System Overview, System Architecture, Data Design, Component Design, Human Interface Design, Requirements Matrix, and Appendices.

The Introduction section, the section you're reading right now, provides a high level view of the software. It introduces the product by name and describes its purpose and goals. It also describes the rest of the document (see: this section).

The System Overview section gives a general description of the functionality, context and design of this project. It also provides the background of the project, including a more thorough description of its predecessor software, tempmon.

The System Architecture section describes the architecture of this project, consisting of a modular program structure and the relationships between the modules to give a complete picture of the system. It also includes the motivations behind the choice of architecture.

The Data Design section explains how the architecture translates into data structures. This includes descriptions of how data is stored, processed and organized. It also includes an alphabetical list of the data structures of the software (by category) with their types and descriptions.

The Component Design section examines the data structures in a systematic way, describing in much higher detail how each such structure functions.

The Human Interface Design section gives an overview of the user interface of the system, including examples contained in screen images and descriptions of how the user will interact with the data structures of this software using components of the user interface.

The Requirements Matrix section provides a cross-reference that traces components from the software requirements specification (SRS) of this project to the data structures and interactions described in this document.

The Appendices section consists of all supporting information in understanding this document. This includes all definitions, acronyms, and a reference manual.

1.4 Reference Material

*[Optional]*

*List any documents which were used as sources of information for the test plan.*

1.5 Definitions and Acronyms

*[Optional]*

*Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SSD. These definitions should be items used in the SSD that are most likely not known to the audience.*

2. System Overview

*Give a general description of the functionality, context and design of your project. Provide any background information if necessary.*

Background: CBSR chose to develop this product because the environmental monitoring landscape was rife with overpriced equipment paired with software that was overly complex and not at all user-friendly. Because CBSR is primarily a bio-banking company and required environmental monitoring, when it came time to decide between one of the available options and none were compliant with our needs we decided to develop one of our own: an open-source solution compliant with the hardware we already had in place. The first rendition, tempmon, was relatively successful despite not having a development plan, but only for satisfying the needs of our own company. A new product, namely this one, needed to be developed to remedy the issues with the original solution. Specifically we seek to create a product that is easy to implement for those without technical expertise yet easy to extend for those who have it.

3. System Architecture

3.1 Architectural Design

*Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high level overview of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with eachother in order to achieve the desired functionality. Don't go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was composed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagrams if required.*

3.2 Decomposition Description

*Provide a decomposition of the subsystems in the architectural design. Supplement with text as needed. You may choose to give a functional description or an object oriented description. For a functional description, put top-level data flow diagrams (DFD) and structural decomposition diagrams. For an OO description, put subsystem model, object diagrams, generalization heirarchy diagrams (if any), aggregation heirarchy diagrams (if any), interface specifications, and sequence diagrams here.*

3.3 Design Rationale

*Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade-offs that were considered. You may discuss other archtectures that were considered, provided that you explain why you didn't choose them.*

4. Data Design

4.1 Data Description

*Explain how the information domain of your system transforms into data structures. Describe how the major data or system entities are stored, processed and organized. List any database or data storage items.*

4.2 Data Dictionary

*Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and their attributes, methods and method parameters.*

5. Component Design

*In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or psuedocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or psuedocode.*

6. Human Interface Design

6.1 Overview of User Interface

*Describe the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.*

6.2 Screen Images

*Display screenshots showing the interface from the user's perspective. These can be hand-drawn or you can use an automated drawing tool. Just make them as accurate as possible.*

6.3 Screen Objects and Actions

*A discussion of screen objects and actions associated with those objects.*

7. Requirements Matrix

*Provide a cross-reference that traces components and data structures to the requirements in your SRS document.*

*Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.*

8. Appendices

*[Optional]*

*Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.*