**Software Design Document (SWDD) Template**

Background

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and data necessary for the implementation phase. The SWDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SWDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage—i.e., the detailed design stage—more detailed data structures are defined and algorithms are developed for the defined architecture.

This template is an annotated outline for a software design document adapted from the *IEEE Recommended Practice for Software Design Descriptions*. The *IEEE Recommended Practice for Software Design Descriptions* have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. For your own information, please refer to IEEE Std 1016[[1]](#footnote-1) for the full *IEEE Recommended Practice for Software Design Descriptions*.

Team 4

**Health Monitoring Application**

#### Software Design Document

Name (s): Sai Dutt, Manish Kumar  
Section:   
Workstation:

Date: (11/20/2024)

**TABLE OF CONTENTS**

1.0 INTRODUCTION 4

1.1 Purpose 4

1.2 Scope 4

1.3 Overview 4

1.4 Reference Material 4

1.5 Definitions and Acronyms 4

2.0 SYSTEM OVERVIEW 4

3.0 SYSTEM ARCHITECTURE 4

3.1 Architectural Design 4

3.2 Decomposition Description 5

3.3 Design Rationale 5

4.0 DATA DESIGN 5

4.1 Data Description 5

4.2 Data Dictionary 5

5.0 COMPONENT DESIGN 5

6.0 HUMAN INTERFACE DESIGN 5

6.1 Overview of User Interface 5

6.2 Screen Images 6

6.3 Screen Objects and Actions 6

7.0 REQUIREMENTS MATRIX 6

8.0 APPENDICES 6

## INTRODUCTION

## Purpose

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

This software design document describes the architecture and detailed system design of the application for health-conscious users that want to optimize their health based on their location.

## Scope

The application provides real-time data on environmental risks such as air quality, water quality and pollen levels. It includes features such as data encryption. The app targets individuals with health concerns by enabling informed decisions regarding environmental hazards.

## Reference Material

* + - IEC 62304 Standards for Software Lifecycle
    - Software Requirements Specification (SRS)

## SOFTWARE OVERVIEW

The application provides users with real-time monitoring of environmental factors such as AQI,

PM2.5, PM10, pollen levels, and water quality. It supports data visualization, and reports.

Built using React Native, the app adheres to industry standards for safety and security

## SOFTWARE CLASSIFICATION

EnviroSafe is classified as **Class A Software** under IEC 62304. This classification ensures

## the app serves as a passive safety tool without direct involvement in life-critical decisions.

## SOFTWARE ARCHITECTURE

## Architectural Design

The system architecture includes the following high-level modules:

• **User Interface (UI):** Displays environmental data and alerts.

• **Backend Services:** Handles API integrations, and user authentication.

• **Data Management:** Stores user-generated reports and configuration data.

• **Security Layer:** Implements AES-256 encryption for data transmission and storage.

A diagram of a person's life cycle

Description automatically generated

## Decomposition Description

• **Frontend:** Manages user interactions, displaying air and water quality indices, and

hazard alerts.

• **Backend Services:** Communicates with external APIs for environmental data,

and stores user data.

## Design Rationale

React Native was chosen for its cross-platform capabilities. Firebase was selected for backend

development due to its robust features such Authentication and Cloud Messaging. The

architecture ensures modularity and scalability.

## COMPONENT DESIGN

**UI (Frontend)**

* **Functionality:** Displays environmental data (AQI, water quality, pollen index), hazard alerts, and user-generated reports.
* **Key features:** 
  + Dynamic dashboards for real-time environmental metrics.
  + Navigation menus for quick access to settings, and reports.

**Auth (Backend-Firebase)**

* **Functionality:** Manages user authentication, ensuring secure login and signup processes.
* **Key Features:**
  + Email/password-based authentication.
  + Password reset and account recovery mechanisms.

**External APIs**

* **Functionality:** Provides real-time environmental data from trusted third-party sources.
* **Key Features:**
  + Access to AQI, pollen levels, and water quality metrics.
  + Regular updates to ensure accurate data representation.
  + APIs: OpenWeatherMap, Google Maps, and water quality APIs.

## HUMAN INTERFACE DESIGN

## Overview of User Interface

The user interface (UI) of the application is designed to ensure ease of use, and accessibility while delivering real-time environmental risk data. The design prioritizes intuitive Software Design Document navigation, seamless interaction, and visual clarity for users to quickly understand and act on the environmental information provided.

The UI adheres to material design principles for consistency across iOS and Android platforms, using React Native to ensure cross-platform compatibility.

## Screen Images

Screens screenshot of a medical application

Description automatically generated

1. [↑](#footnote-ref-1)