Landfill e-Forms Application
Software Design Document

Brian Becerra   Jason Salazar   Krystal Lo   Mark Canseco

April 2025

Contents

Version History

User	Date	Reason for Changes	Version
Krystal Lo	5/6/25	Update for snapshot 1	1.0
Jason Salazar	5/6/25	Update for snapshot 2	2.0
Mark Canseco	5/6/25	Update for snapshot 3	3.0
Brian Becerra	5/6/23	Update for snapshot 4	4.0

1 Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the Landfill e-Forms Application (LEF). This document will outline the system architecture, user interface, and intended audience for the application.

1.2 Scope

The scope of this document includes the design and implementation of the LEF. This document was created by the development team and is intended to be used as a reference for the project.

1.3 Intended Audience

The intended audience for this document includes the development team, project managers, and stakeholders involved in the LEF project.

1.4 Overview

The LEF application will be a web-based application and a mobile application that allows users to submit and manage landfill-related forms electronically. The application will provide a user-friendly interface and will be designed to streamline the process of submitting and managing forms.

1.5 References

See the references section for a list of documents and resources referenced in this document.

1.6 Definitions, Acronyms, and Abbreviations

See the glossary section for definitions of terms, acronyms, and abbreviations used in this document.

2   System Architecture

2.1   Workflow

The workflow of the LEF application consists of the following steps:

- User submits a form through the mobile application.

- The form data is sent to the web application for processing.

- The web application verifies the data and stores it in the database.

- Users can view and manage submitted forms through the web application.

- Users can generate reports based on the submitted forms.

2.2   Site Breakdown

The LEF application will consist of the following components:

- **Web Application:** The main interface for users to submit and manage forms.

- **Mobile Application:** The interface for users to submit forms on the go.

- **Database:** The backend storage for all form data and user information.

- **Server:** The server that hosts the web application and database.

- **API:** The interface for communication between the mobile application and web application.

2.3   Architecture Overview

The mobile application follows a Model-View-ViewModel (MVVM) architectural pattern. The **View** consists of the UI screens for data entry and display. The **ViewModel** handles the logic and data preparation for the UI. The **Model** represents the data and interacts with the local database and the remote server. The "Audit" data entry screen and list view each have their corresponding ViewModels.

2.4   Data Flow - New Audit

1. The user interacts with the "New Audit" screen, entering data into the form fields.

2. The View binds user input to properties in the Audit ViewModel.

3. Upon saving, the ViewModel validates the data and interacts with the local data storage (Model) to contain the information in the local database.

4. When the user initiates synchronization, the ViewModel retrieves the locally stored audit data and send it to the backend API endpoint.

3   User Interface

3.1   How to Use

3.1.1   Web Application

Open the web application in a browser and log in using your credentials. The main dashboard contains icons for the following features:

- **Andriod Data Sync:** Upload data from the mobile application and download data from the web application.

- **Unverified Forms:** View data passed from the mobile application that has not been verified yet.

- **Exceedances:** View current exceedances and their status, as well as conduct repairs.

- **Reports:** View reports and filter them by type, location, and date.

- **Equipment:** Manage equipment inventory and manage equipment types.

- **Users:** View and manage users and user groups. Add, edit, assign user roles, and delete users.

- **Application Settings:** (Visible only for administrators) Manage security token expiration times, adjust user account information, and change super administrator passwords.

- **Information:** View information about the application, release notes, and user manuals.

3.1.2   Mobile Application

If installing through Android Studio, we must first set up the Android Studio environment.Start Android Studio and select "Open an existing Android Studio project". Select the folder where the repository was cloned and select "landfill-android-app".

For application installation, a phone with Android 5.1 (Lollipop) or higher is required.

- To install through Android Studio, connect the phone to the computer. Go to Build > Build APK and the application will be installed on the phone.

- To install through File Download, ensure no device is connected and go to Build > Build APK. The .apk file will be downloaded inside the project folder. To retrieve the .apk file, go to "landfill-android-app > app > build > outputs > apk". Send the .apk file to the selected device and open it to install.

3.2   Database Explanation

The database for the LEF is handled by Microsoft SQL Server. It will store all data related to the forms, users, and equipment. The following tools are used to manage the database:

- **Microsoft SQL Server:** The relational database management system.

- **Microsoft SQL Server Management Studio (SSMS)**

- **DBeaver (Alternative):** Universal database tool.

Glossary

LEF - Landfill e-Forms Application

UI - User Interface

DB - Database

References

Ascent - Project. (2017). Cysun.org. https://ascent.cysun.org/project/project/view/72
Ascent - Project. (2022). Cysun.org. https://ascent.cysun.org/project/project/view/44