DSI Application

Version 1.4

Table of Contents

1. Introduction 3

1.1 Problem Statement 3

1.2 Product Vision Statement 3

2. Requirements Overview 3

2.1 User Stories 3

3. Technical Overview 4

3.1 Design Constraints 4

3.2 Solution Strategy 4

4. System Design 5

4.1 Context and Scope 5

4.2 Container Diagram 6

4.3 Component and Runtime Design(s) 6

5. Glossary 6

# Introduction

This document will go into detail the problem details, requirements, strategy, and design of the DSI application. This section will define the existing problem, and what the vision of the solution looks like.

## Problem Statement

This document provides a comprehensive overview of the DSI Application and the process that was used to construct it. The document captures the importance of the solution, system requirements, a high-level schedule, system architecture, and the software quality plan. The purpose of this project is creating an application for Death Scene Investigators to fill out a digital form of records from the field

## Product Vision Statement

For Kentucky coroners who need a method to accurately report death scene investigations, the progressive web application is a software that will improve data collection from death scenes over inefficient pen-and-paper forms. Our product offers significant improvements over the old web-based reporting system (CRI) by providing a responsive UI for the investigators, improving mobility of the application by caching forms that cannot be sent immediately to a database due to a lack of wireless connection, and ultimately increasing the overall efficiency of the National Violent Death Reporting System.

# Requirements Overview

This project must satisfy all given needs of coroners who will be using this application. It should replace the use of paper documents for all DSI cases in Kentucky, so it must contain every feature possible now and more to show the value of switching systems. In this section, we will outline the concerns of users and acceptance criteria needed to confirm this will provide a valuable experience.

## User Stories

## This section includes selected epics and user stories created for this project. Each epic is accompanied by acceptance criteria and any important notes.

## Epic 1: As a coroner, I should view the form from any device. To do this, it needs to comply with the PWA formatting policy Acceptance: The form can be accessed from a mobile device, tablet, and laptop.

## Epic 2: As a ­, I should be able to view the Original DSI form as a Digital form Note: The categories given in the initial form should be transferred in a friendly manner so coroners can recognize what is needed in each area of the digital form. Acceptance: a coroner should approve the layout, or category names should be exactly the same as the initial form.

## Epic 3: As a database administrator, I should be able to view the forms using the database. We are creating a backend using Firebase but may have to connect to a different backend in the future. The forms for the final product should be sent to the specified location for later review. Acceptance: A different location will be used for initial testing; if we can access the final database, we can send a test document and determine if it uploads successfully.

## Epic 4: As a coroner, I want to be able to upload files so that I can record physical evidence Notes: This should include audio and/or visual files with a different section for each. Local photos from the native device will also need to be accepted, such as from the device camera if possible. A separate section will be provided for file uploads. Acceptance: Given that I want to remove files, I should be able to remove a file that was previously uploaded Acceptance: Given that I want to upload multiple files, I should be able to see all files that have been uploaded. Previously uploaded files should not be overwritten. Acceptance: Given that I want to upload a file if a file does not upload successfully, I should be notified that the upload could not be completed Acceptance: Given that I want to upload multiple files, I should upload a file even after a file has already been cached.

## User Story 1: As a coroner, I want to be able to use this PWA offline so that I can work in areas with poor internet connectivity Notes: This is a feature of all PWA’s, but it is a vital feature for our application that needs to be checked. Acceptance: A user can access the PWA without an internet connection Acceptance: Data should not be lost upon losing internet connection.

## User Story 2: As a coroner, I want to be able to submit my work so that it can be uploaded to the CDC database Acceptance: Given that I submitted the form, I should be notified whether or not the file was successfully submitted and the reasoning (too large, no connection, other error) Acceptance: If the form fails to submit, the data should still be on the device.

## User Story 3: As a coroner, I want to be able to save and close the PWA so that I can continue where I left off later using the save feature. Acceptance: If the application is closed and reopened, information should be retained. Acceptance: A save button is visible for information to be locally saved on the device.

## User Story 4: As a coroner, I should be able to upload a photo from my local device storage. Note: This can be the photos application for iOS or the file’s location from Windows or Linux device. This is a story within the epic of uploading files. Acceptance: A photo should be selected from the device and submitted with the form Acceptance: The photo should be viewable from the final submission form in the database Acceptance: The chosen photo should be viewable from the PWA to confirm the correct image was selected Acceptance: an error should be displayed if the photo was not uploaded or selected correctly.

## User 1: As a database administrator managing information for the national database, this is easier to handle and analyze than paper documents. The PWA will accomplish this by aggregating the submitted forms.

## User 2: As a coroner, this improves the clarity of my writing and reduces reprints. The PWA will accomplish this by being a digital form to enter information. This allows me to attach photos and other files instead of remembering later directly. The PWA will accomplish this by allowing file uploads. This PWA will allow me to work using a digital form even when an internet connection is poor in the local area.

## User 3: As an Investigator, it will improve my workflow by not requiring physical documents to be transferred, thereby improving timelines for investigations.

# Technical Overview

## Design Constrains

For this project, there are several design constraints of importance. The users of this application may work in areas without internet for extended periods of time, therefore this application should be able to work without internet access. To accomplish this goal, our group has chosen to create a progressive web application (PWA) due to its ability to cache data and work offline.

Using a PWA creates some additional constraints, such as ensuring HTTPS capability, determining an application icon, and creating a web manifest file. This satisfies an additional constraint of security. Due to the nature of the data being processed, HTTPS is required to comply with regulations. Therefore, the PWA will satisfy this constraint as well.

Another constraint for this project requires a specific set of pre-determined questions be asked and answered by users. To comply with this constraint, a digital form needs to be created to mimic the existing forms. Finally, users of this application may want to upload additional information not concerning the specific questions. To allow for this capability, users need to be allowed to upload files with their form. This constraint will be satisfied by allowing for files uploads near the end of the digital form.

## Solution Strategy

Quality: There are several considerations that need to be made for developing this application. The first question we had is what privacy laws we need to follow concerning the investigation data being recorded and submitted through this application. Some privacy laws, such as HIPPA, may require specific security for an application concerning health data. We have asked Dr. Brown about such concerns and are waiting on her input. This means that it is important to make sure people can access someone else's data using the PWA, and that the method of submitting the data from the PWA is secure. The PWA will be a one-way submission so it cannot be used to access the information someone else submitted. That will be handled on the database end for the CDC. Another concern is maintainability. When the software needs to be updated or improved, it is much easier to maintain a single codebase than to update a separate codebase for every platform. The PWA handles this issue by being a "one-size fits all" application that can be updated once to satisfy every platform.

The approach to this project is to make a PWA that can be used online or offline, and when online it can submit the form to a larger database. To work towards this solution, the framework being used for this project will be Next.js building on React. A template for a Next.js PWA will be used as the foundation. The use of Next.js allows a simplified approach and provides existing documentation and framework for us to build from. Additionally, it should have faster load times for users. A form library will be used to implement the form filling requirements.

# System Design

The overall design is to use a PWA to take in all of the needed data from the scene. Then package it up into a secured JSON file and send it to the backend. Where in the file is received and added to the account which can be accessed globally and by anyone with proper authentication.

In figure 1 and figure 2, the key can be summarized by a solid box represents a portion of the DSI software, a dotted box represents an external product, a human figure represents user(s), and arrows represent data transfer or usage of a system.

## Context and Scope

Diagram

Description automatically generated

Figure 1: Context and Scope

Figure 1 shows the overall context and scope for this application. The users in our diagram are the Coroners and Investigators who will be using the PWA we create, and the database administrators who will interface with the existing backend where the completed forms will be sent.

In this system, the Coroners and Investigators will be using the DSI software system to fill out forms. Once this is complete, they will submit the form to be uploaded to the backend UKY database. They will not have access to the forms on the database. The database administrators will have access to this backend database to analyze the data that was submitted through these forms.

## Container Diagram

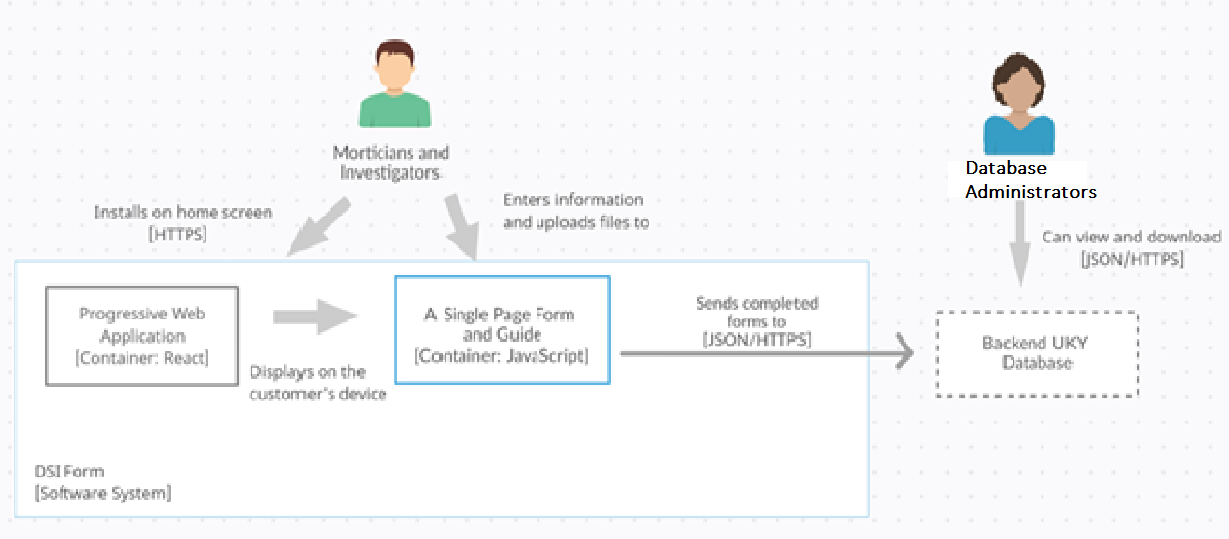


Figure 2: Container Diagram

## The container diagram decomposition shown in figure 2 follows the structure we discussed with our customer. Utilizing a Progressive Web Application, we are sending data to a specific endpoint for the database administrators to analyze data. The database administrators will be working out of the UK Statistics department as identified in figure 2. The backend database is shown outside of the container as it should be already implemented for our frontend to send data to.

## The Coroners and Investigators need to be able to download the PWA, enter information into the form, then upload the document to the endpoint. This follows the container diagram shown and they interface with the PWA and then with the form displayed in the application.

## For data interfacing, there are only three points where data needs to be transmitted. The JSON/HTTPS format for the backend database was already implemented, and HTTPS is used for the PWA to be downloaded from the internet to the users’ device. The UK statistics department can interface with the backend using the same method we use to communicate data to the backend. Although that is not within the expected scope of this project, it is important to include so that the expectations of the final application are clear for future progress.

## Components or Runtime Design(s)

One of the primary components that will be used the Google Firebase Cloud Firestore backend. This was chosen because of its up time, ease of use, built in authentication and the level of provided documentation. The last part is to the team the most important as we will be handing this project off and having a reliable backend that is simple to update and keep current is paramount. With the Google solution that will not be an issue as it is easy to modify on the fly and to remount to a new group or external persons as well. And another added bonus of this approach is in the cost of the solution, it is scaled to the number of requests and access’s which for the project the less is better in all regards.

# Glossary

|  |  |
| --- | --- |
| Term | Definition |
| PWA | A progressive web application, in this case a form with pages that does not need an active connection. |
| JSON  HTTPS  Firebase  React  Next.js | A file format that uses human readable text; based on pairs of data entry’s  An internet communication protocol used for secure communication by encrypting data.  A platform developed by Google for creating mobile and web applications.  A free and open-source front-end JavaScript library for building user interfaces based on UI components.  A open-source web development framework built on top of Node.js enabling React based web applications functionalities such as server-side rendering and generating static websites. |