CSI 4999 MIM System Software (MIMS) Project Report

**Authors:**

Brandyn Ureel

Daniel Matache

Mark Bruce

Michael Dashe

Katherine Schwartz

### Table of Contents:

[1. Project vision](#_tf3t6rlk72cw)

1.1. Backgrounds

1.2. Socio-economical Impact, Business Objectives, and Gap Analysis

1.3. Security and ethical concerns

1.4. Glossary of Key Terms

[2. Project Execution and Planning](#_d8v85d99d9wp)

2.1. Team Information

2.2. Tools and Technology

2.3. Project Plan

2.4. Best standards and Practices

[3. System Requirement Analysis](#_vl4ezbpu58hj)

3.1. Function Requirements

3.2. Non-functional Requirements

3.3. On-Screen Appearance of landing and other pages requirements.

3.4. Wireframe designs

[4. Functional Requirements Specification](#_kti2dkkyogh0)

4.1. Stakeholders 4.2. Actors and Goals

4.3. User stories, scenarios and Use Cases

4.4. System Sequence / Activity Diagrams

[5. User Interface Specifications](#_mgi0jyqwhs8d)

5.1. Preliminary Design

5.2. User Effort Estimation

[6. Static Design](#_ucdnovompztn)

6.1. Class Model

6.2. System Operation Contracts

6.3. Mathematical Model

6.4. Entity Relation

[7. Dynamic Design](#_b25gjxjgrbxz)

7.1. Sequence Diagrams.

7.2. Interface Specification

7.3. State Diagrams

[8. System Architecture and System Design](#_odzp34axom3p)

8.1. Subsystems / Component / Design Pattern Identification

8.2. Mapping Subsystems to Hardware (Deployment Diagram)

8.3. Persistent Data Storage

8.4. Network Protocol

8.5. Global Control Flow

8.6. Hardware Requirement

[9. Algorithms and Data Structures](#_sttn3m9dikir)

9.1. Algorithms

9.2. Data Structures

[10. User Interface Design and Implementation](#_tv8o6ndx88g3)

10.1. User Interface Design

10.2. User Interface Implementation

[11. Testing](#_q36uk5zfrn8r)

11.1. Unit Test Architecture and Strategy/Framework

11.2. Unit test definition, test data selection

11.3. System Test Specification

11.4. Test Reports per Spring

[12. Project Management](#_es1ftu7opfxc)

12.1. Project Plan

12.2. Risk management

[13. References](#_80ssovbjpwgd)

## 1. Project vision

#### 1.1. Backgrounds

In 2016, law enforcement reported over 600,000 missing children nationwide. At this time, there remain over 80,000 missing from the original total. While this number increases each year, it is paramount to reduce the response time for public awareness of those that have been: (1) reported missing or (2) lured into human trafficking. In addition, the rise of human trafficking burdens the system for the half- million law enforcement (LE) officers nationwide. Unfortunately, missing children and victims of human trafficking are often unreported. Many parents are discouraged by the processes from making an official report to local authorities. Resources for retrieving proper investigatory leads are thereby limited for LE to launch proactive investigations. Instead, LE personnel are isolated to a reactive investigation that may sadly lead only to a recovery of the missing/exploited victim’s remains. It is essential that families of those that have gone missing or been a victim of human trafficking should be given the advantage of online digital tools to help in locating their missing loved ones. This project will address current deficiencies in publicizing information about missing persons and then facilitate enlisting of the public’s assistance in trying to locate these children.

#### 1.2. Socio-economical Impact, Business Objectives, and Gap Analysis

#### 1.3. Security and ethical concerns

The information which will be collected by this application, especially the personal information of the victims and their families, is sensitive. It is therefore imperative that the application strikes the correct balance between informing the public to help locate the victim and protecting the privacy of all innocent people involved. The availability of such information must be limited and security measures to protect it implemented. We will accomplish this by allowing different levels of access for different users, so the family of the victims can share information with law enforcement that is not available to the general public. We must also ensure that the front-end code is secure and non-exploitable, and that querying of the database is done in such a way that a user cannot force application to return information that they shouldn’t have access to.

Also, when gathering information about the victim’s contacts, we must be sure to stay within the bounds of the law. Accessing certain information from third-parties, such as cell service providers, can introduce legal problems and may require a warrant[1]. Therefore, we must ensure the application will only obtain information through legal avenues, such as voluntary input by families of victims.

#### 1.4. Glossary of Key Terms

## 2. Project Execution and Planning

#### 2.1. Team Information

Members:

###### Brandyn Ureel

* + Major: Information Technology

###### Daniel Matache

* + Major: Information Technology

###### Mark Bruce

* + Major: Information Technology

###### Michael Dashe

* + Major: Information Technology

###### Katherine Schwartz

* + Major: Computer Science

#### 2.2. Tools and Technology

* Angular
  + Angular (also known as Angular2) is an open-source web application framework. Angular projects are coded in TypeScript (a superset of JavaScript). It is modular and designed for speed and ease of use.
  + Angular is a powerful framework for building application frontends. It has extensive documentation, and members of our team are already familiar with it.
* Cucumber
  + Cucumber is an open-source software testing tool. There are Cucumber implementations in a variety of different programming languages, including Java, Python, and Ruby. It uses the Gherkin language to define tests. Gherkin is easily readable and allows both programmers and non-programmers to define unambiguous tests.
  + Cucumber provides an adequate testing tool for our project. Tests in Cucumber are easy to define and can be written by any member of the team. Perhaps most importantly, an implementation is available in the language we wish to test (Java).
* Gradle
  + Gradle is an open-source tool that automates the process of creating a software build. It supports many languages and platforms. It prioritizes efficiency and performance. It is designed to cater especially to large and complex builds.
  + Gradle provides an adequate tool for managing and automating our project’s software build.
  + Multi-language, multi-platform tool commonly used and heavily supported by community
* MySql
  + MySQL is a free relational database system. It is commonly used in many domains. Though originally released in 1995, it is still actively maintained. It is currently owned by Oracle Corporation.
  + Relational database system is suitable for our data
  + Available on multiple platforms (Windows, Mac, Linux)
  + Part of XAMPP stack
* PostgreSql
  + PostgreSQL (also referred to as Postgres) is an open-source relational database system. It is used in many domains, including by popular websites such as Reddit and Instagram. Though originally released in 1996, it is still actively maintained by the PostgreSQL Global Development Group.
  + Relational database system is suitable for our data
  + Available on multiple platforms (Windows, Mac, Linux)
  + PostgreSQL is reliable, mature, and free.
* Spring Boot
  + Spring is an open-source application framework for the Java platform. It is modular by design, allowing programmers to choose from a suite of modules that provide various services.
  + Spring Boot allows easy creation of Spring application
  + Spring is a widely-used and powerful Java-based application framework
* XAMPP
  + XAMPP is an Apache web server distribution containing the MySQL database and the PHP scripting language. It is designed to simplify the process of running a local web server for testing purposes.
  + Single solution for entire development server stack
* [Face\_recognition by ageitgey](https://github.com/ageitgey/face_recognition)
  + Face\_recognition is a facial recognition project sourced from GitHub and has a MIT free use licence
  + “The world's simplest facial recognition api for Python and the command line”
  + Requires
    - Python
    - Dlib with Python settings
    - Linux or MacOS
* Adobe XD

#### 2.3. Project Plan

#### 2.4. Best standards and Practices

## 3. System Requirement Analysis

#### 3.1. Function Requirements

* Create an account
* Create a Missing person profile
* E-mail law enforcement
* Log back in
* Print off report
* Citizen submit a report
* Search and display missing persons

#### 3.2. Non-functional Requirements

* Facial recognition search/confirmation
* Uploading photos
* E-mail parents notifications
* Confirmation of users
* Find nearest police station

#### 3.3. On-Screen Appearance of landing and other pages requirements.

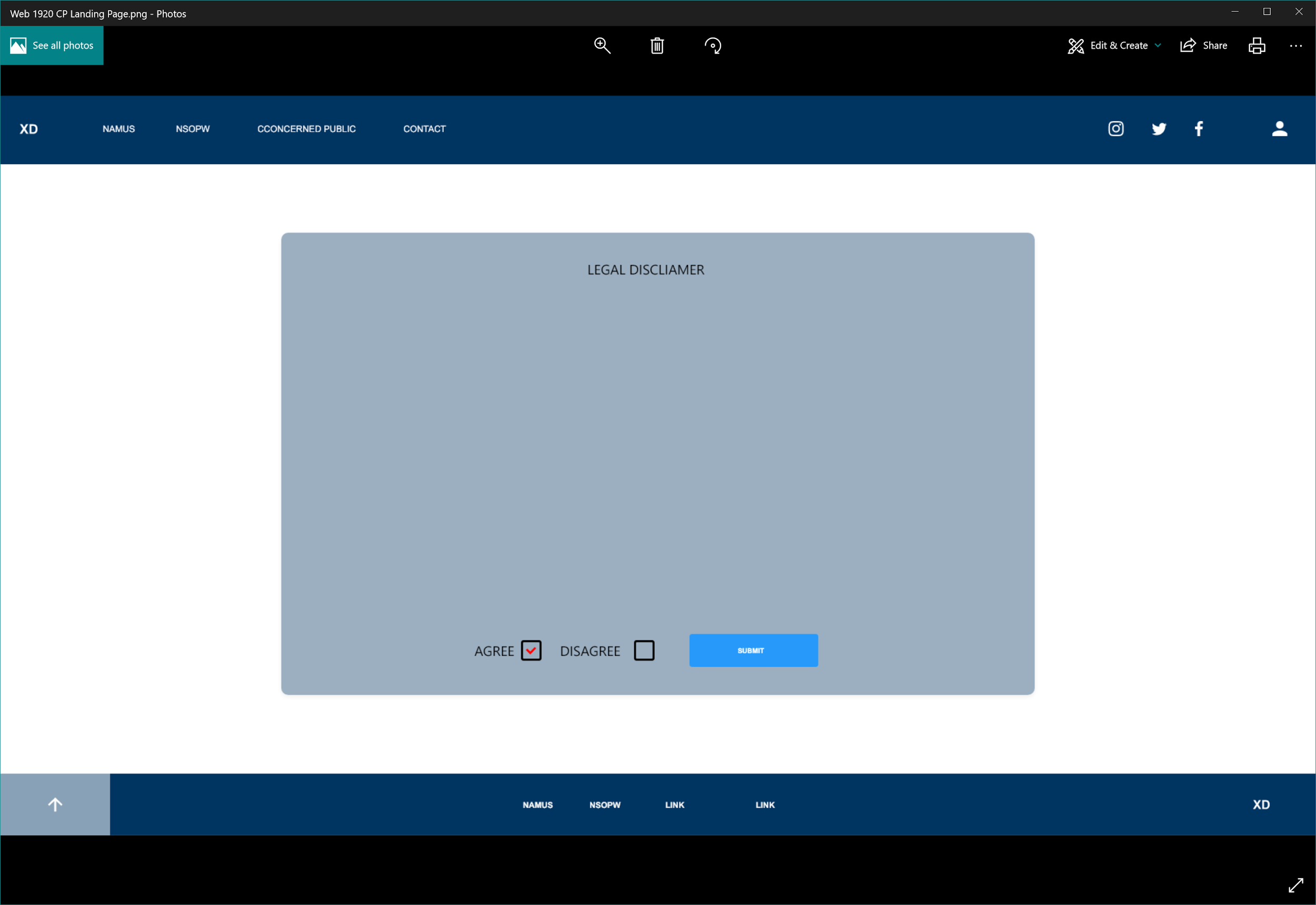
* Have the landing page display a layout of missing children's pictures last seen in the area.
* Have a page to notify users and have them agree to terms of legal disclaimer.
* Have a clean layout that is easy and intuitive to use.
* Have forms easy to understand and intuitive to fill in.
* Make things as straightforward as possible to both parents and Concerned Public

#### 

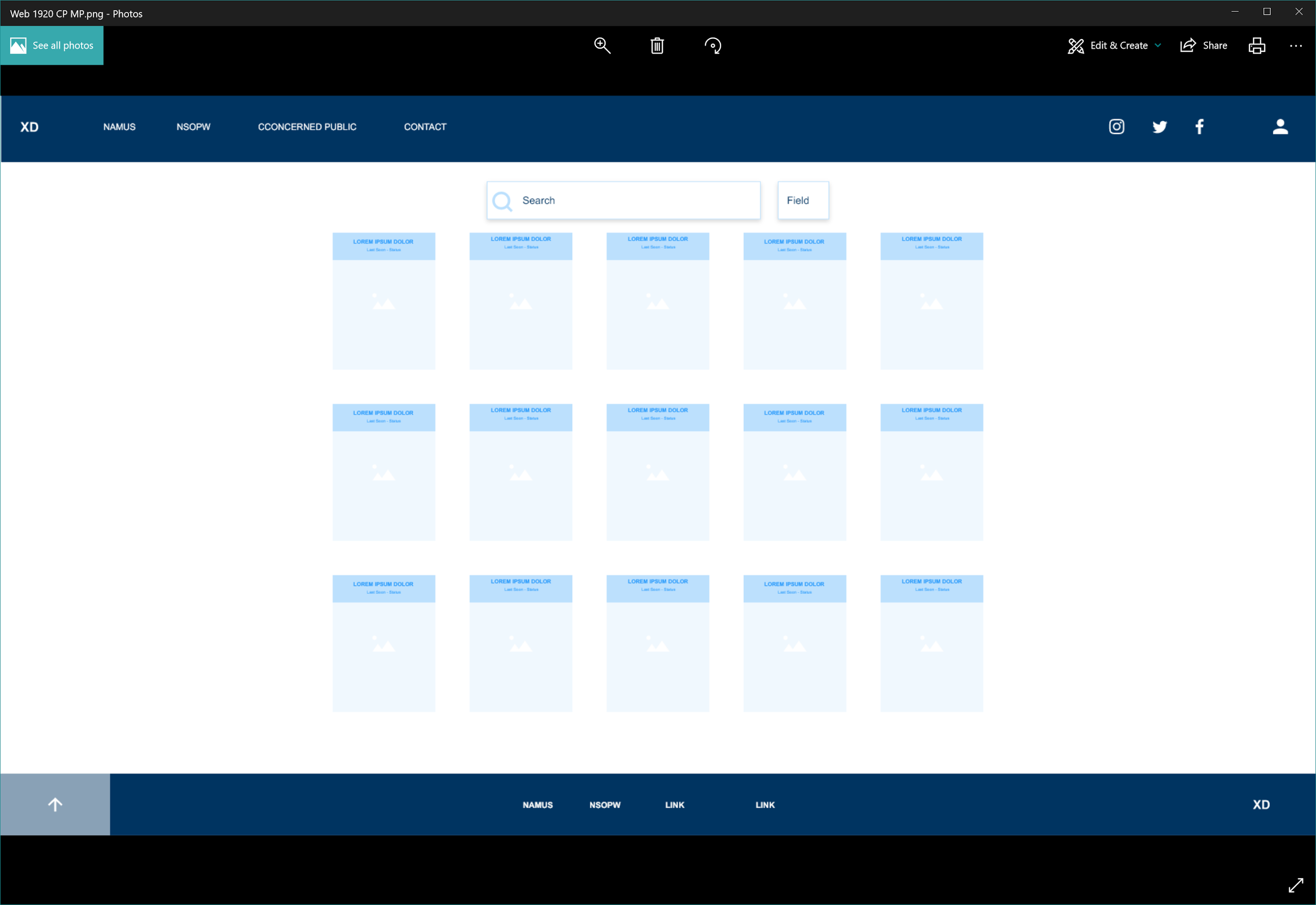
#### 

#### 3.4. Wireframe designs

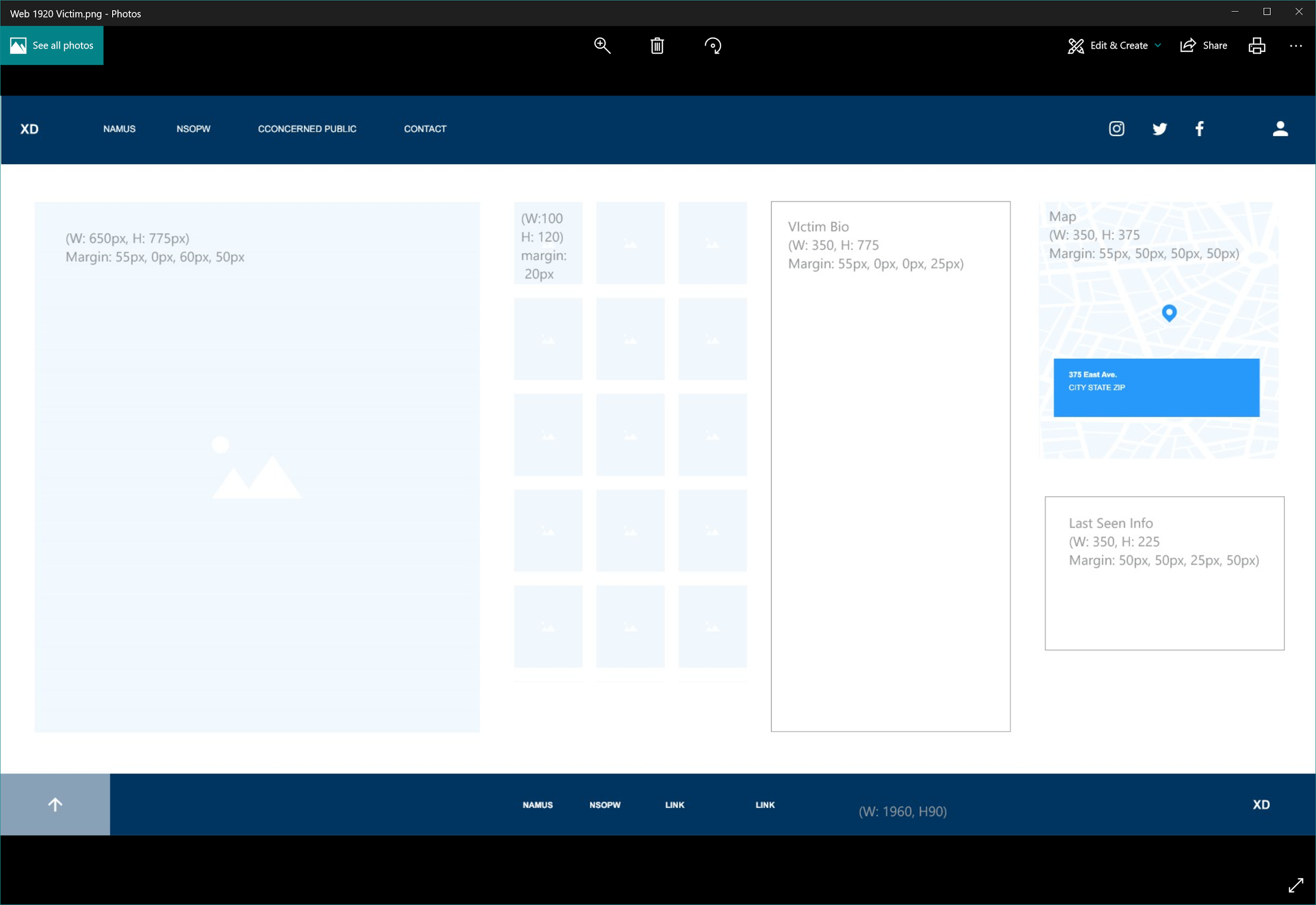
##### Concerned Public



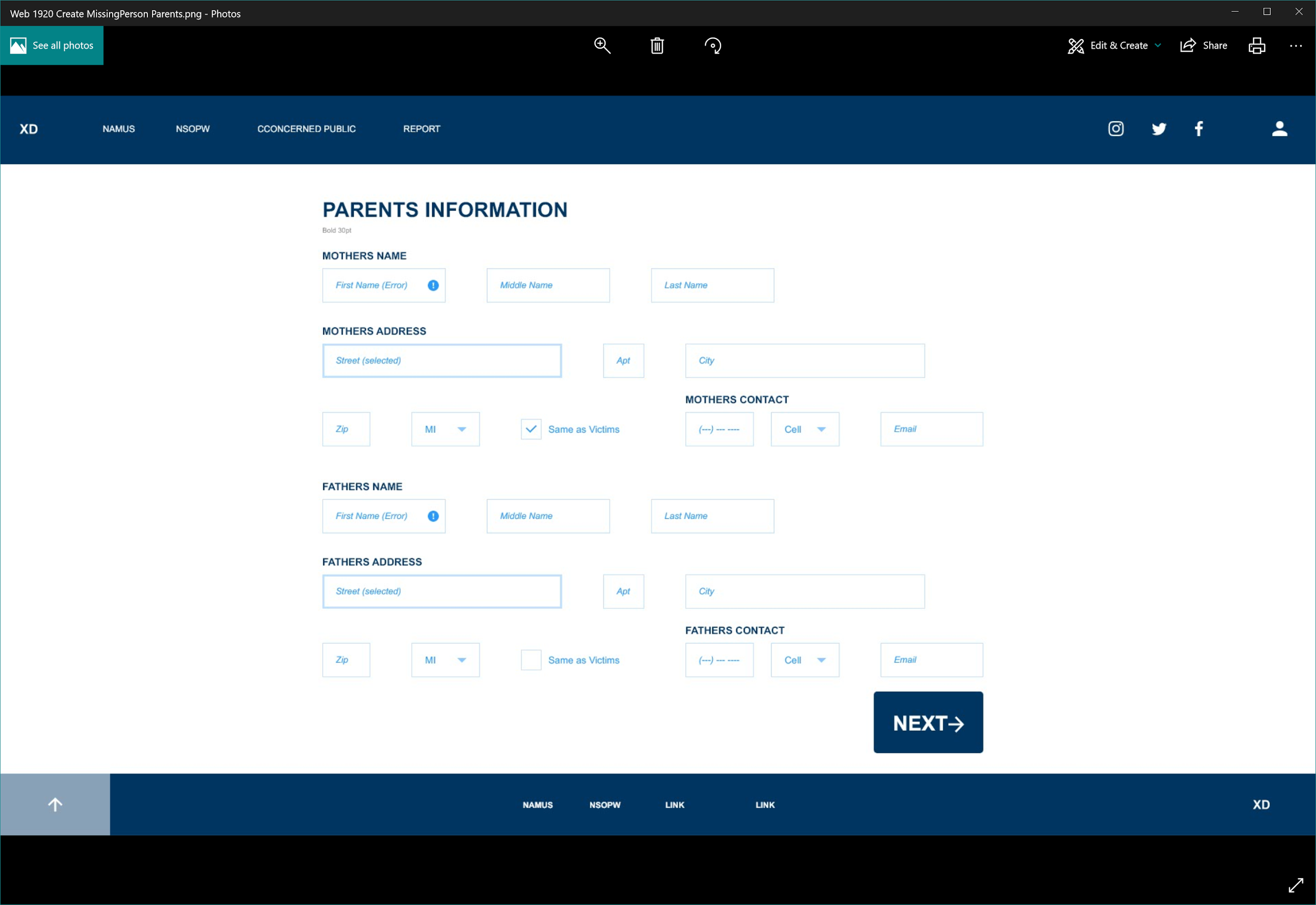
Disclaimer Agreement



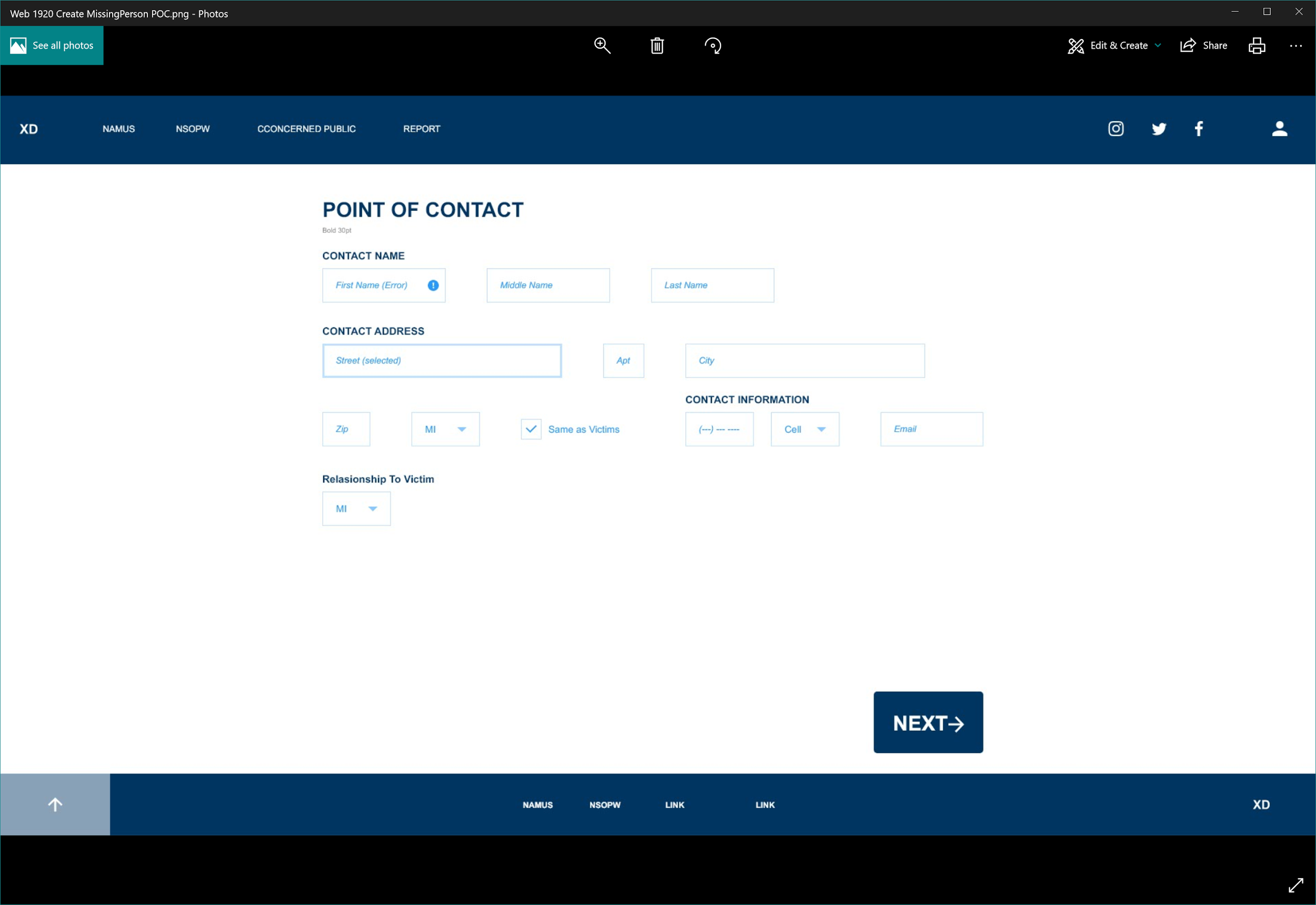
Landing Page



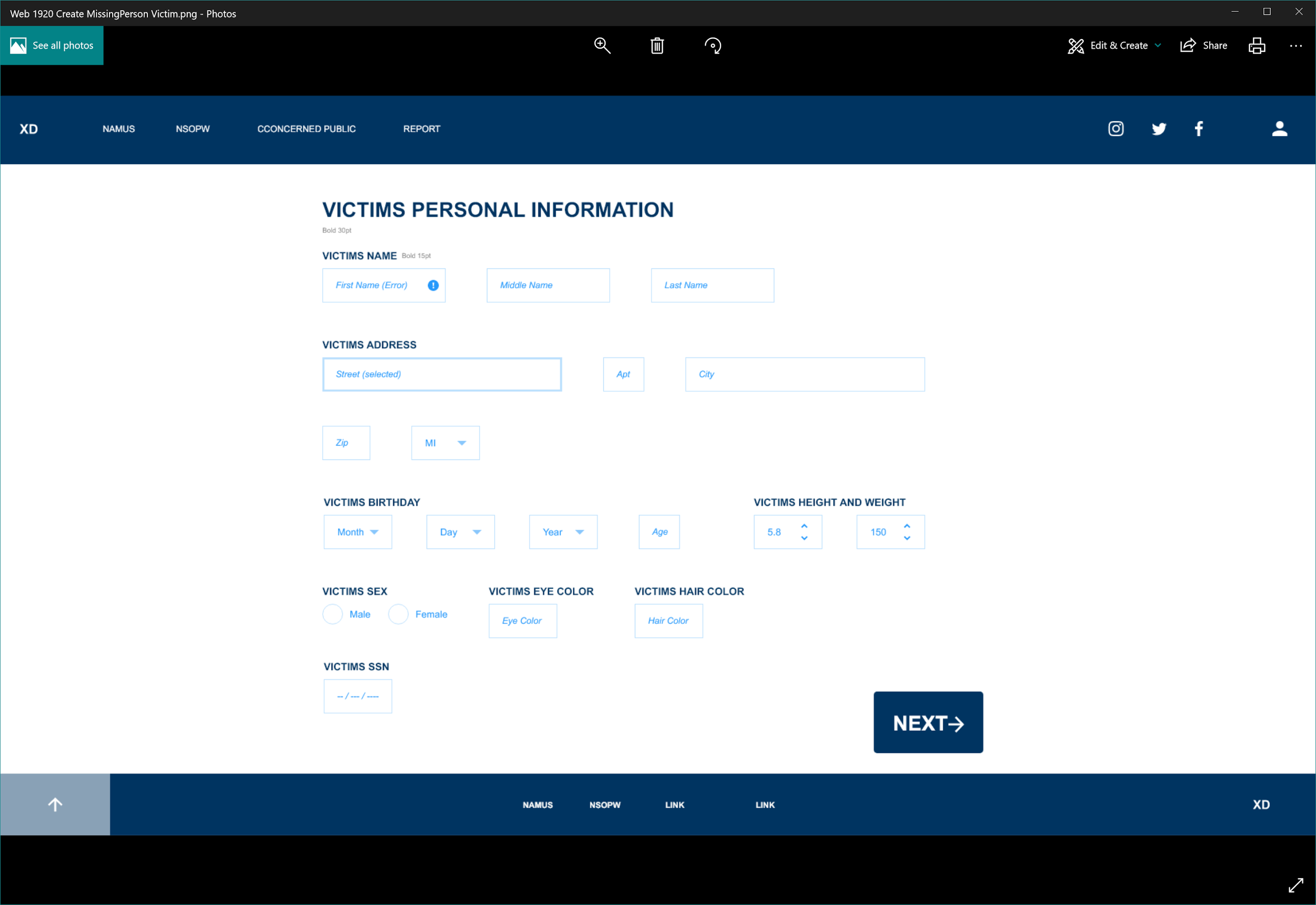
##### Forms



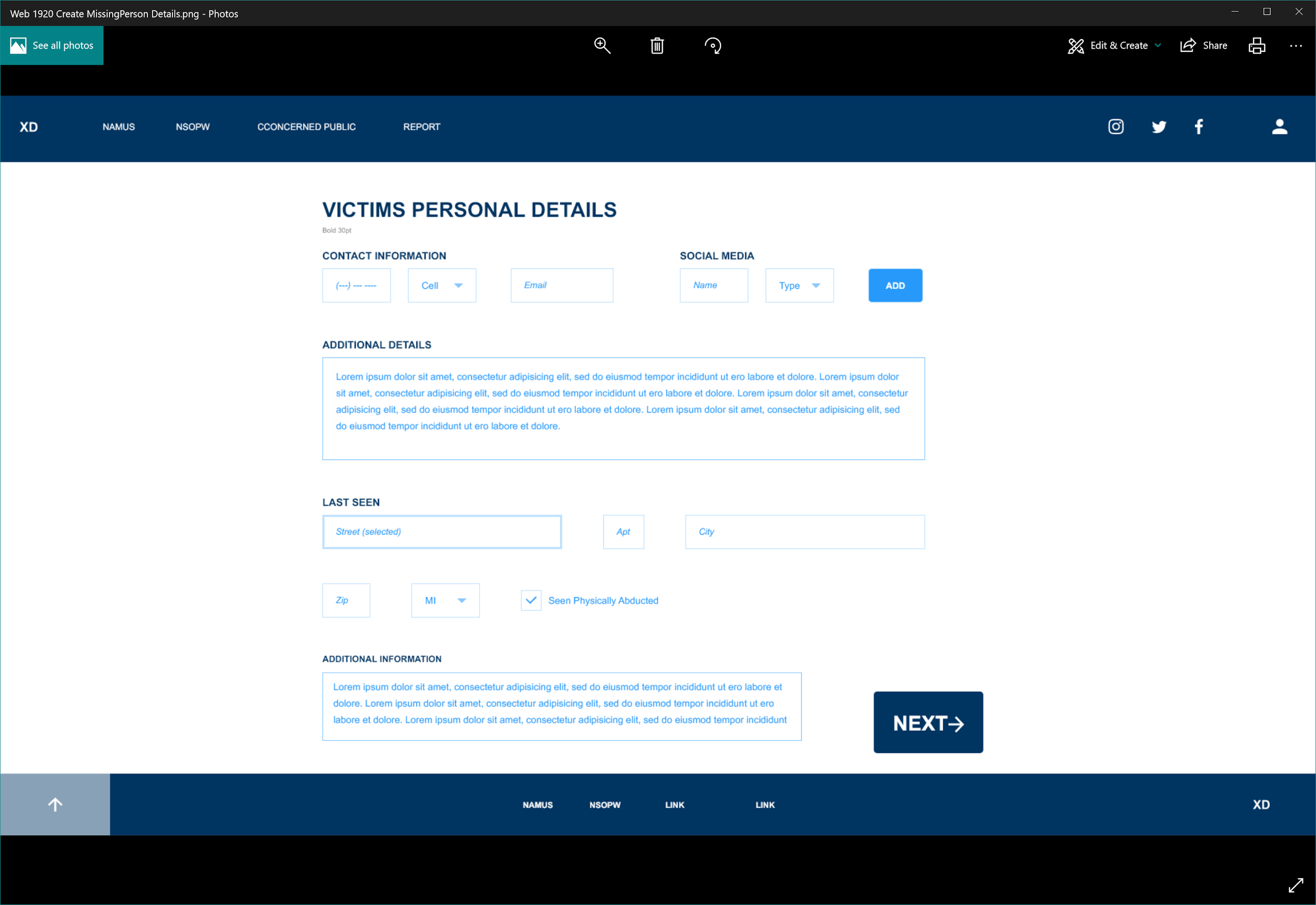
Parents Information Form



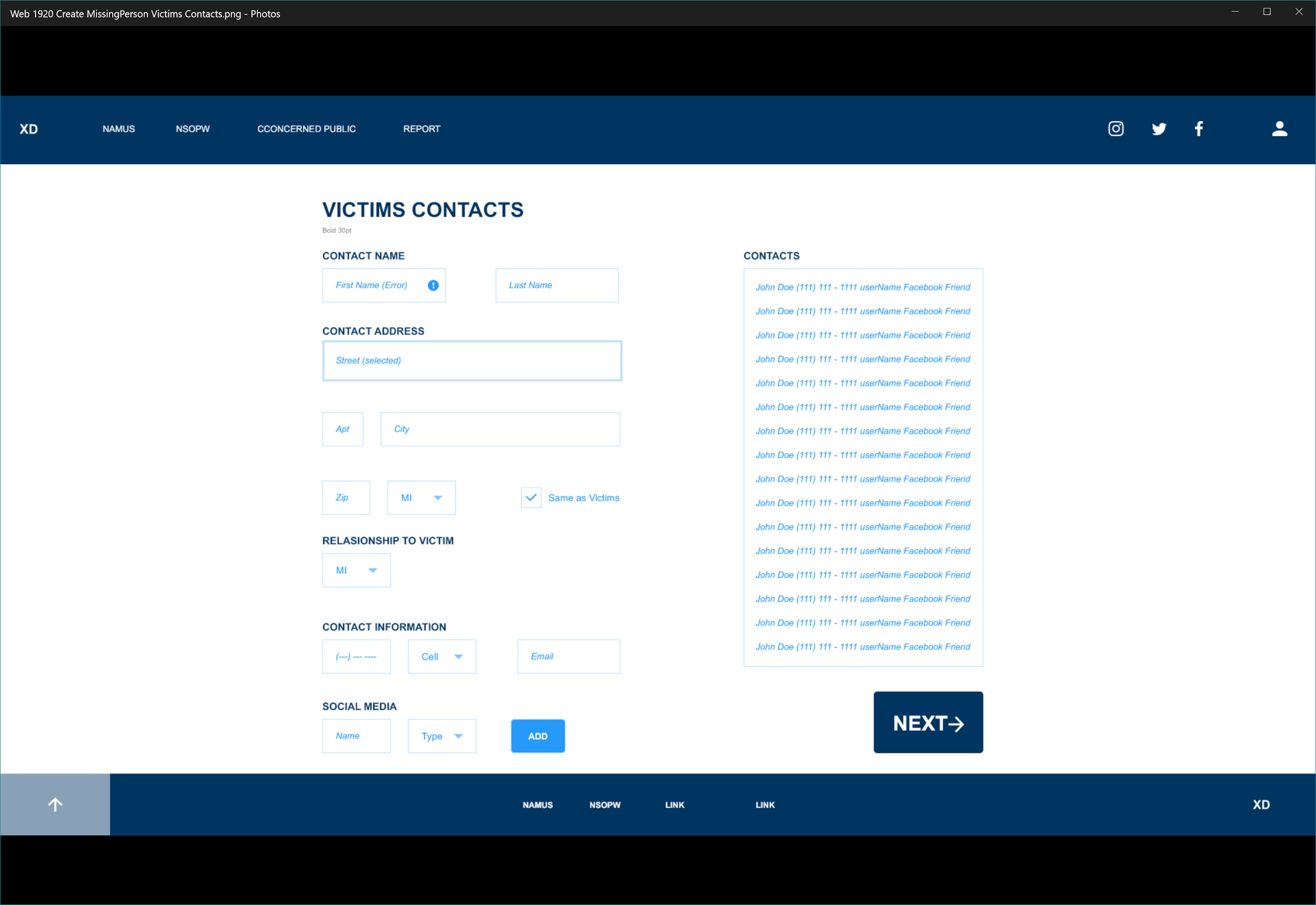
Point of contact Form



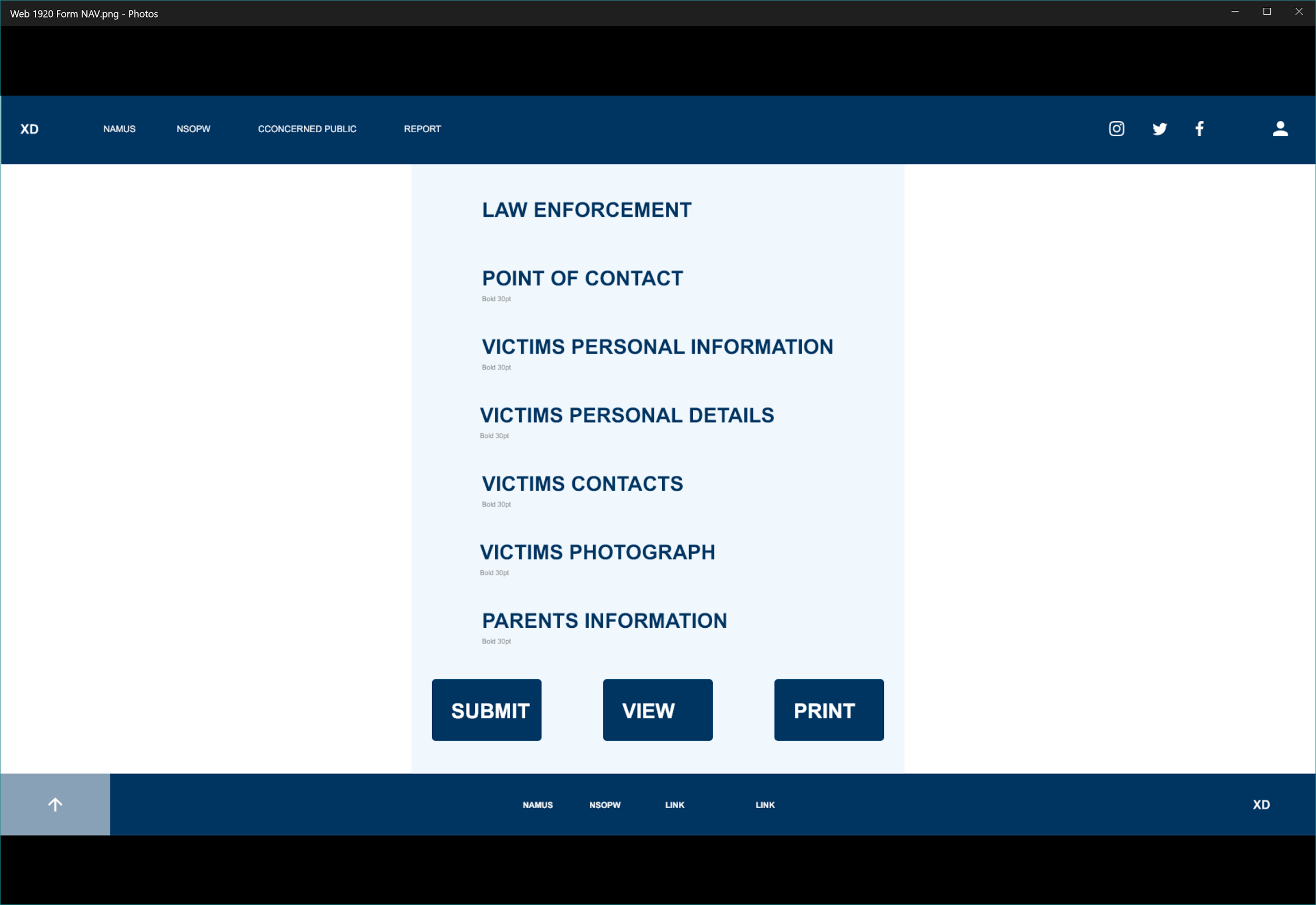
Victims personal Information Form



Victim Personal Details Form

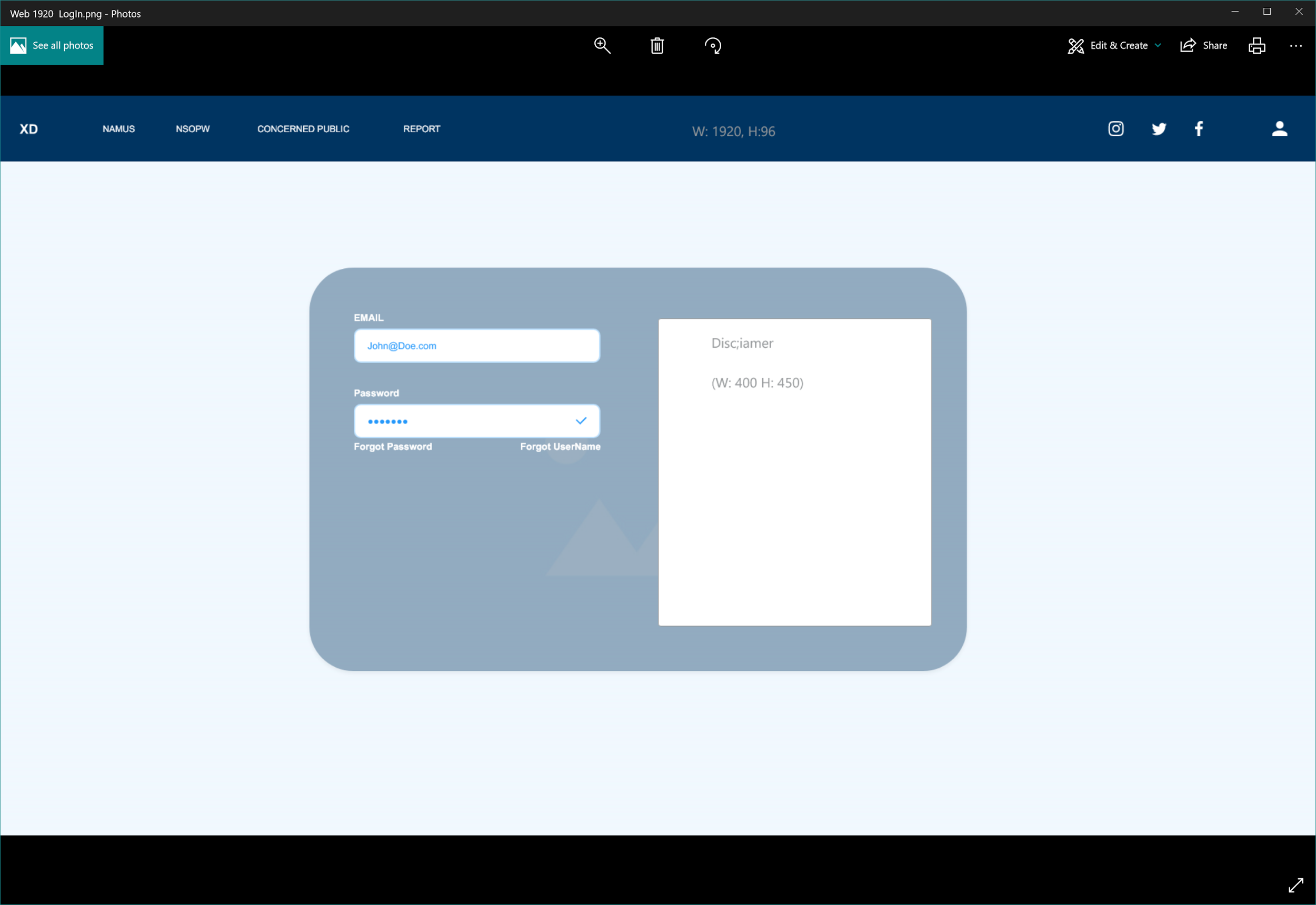


Victims contacts Form

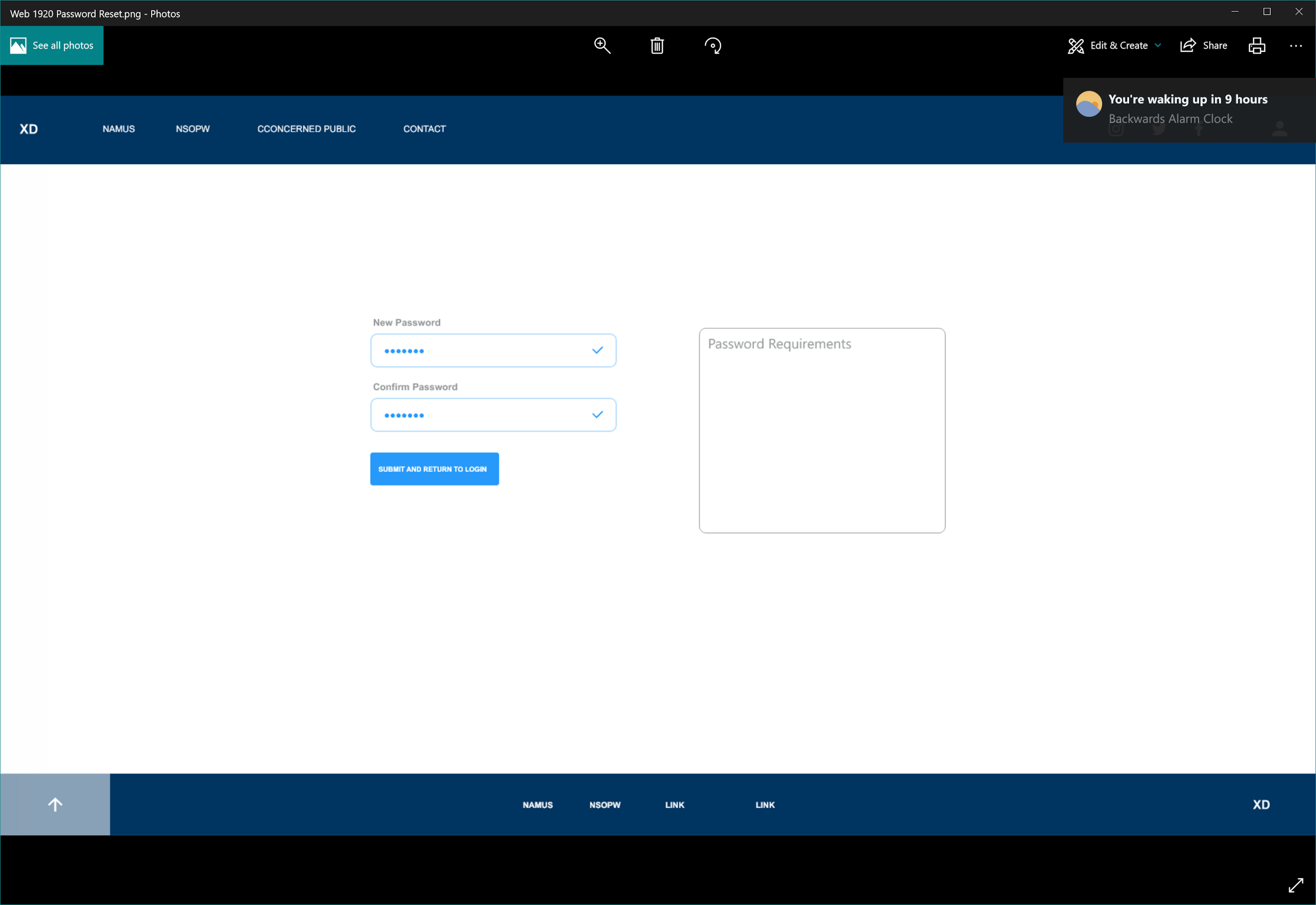


Navigation Page

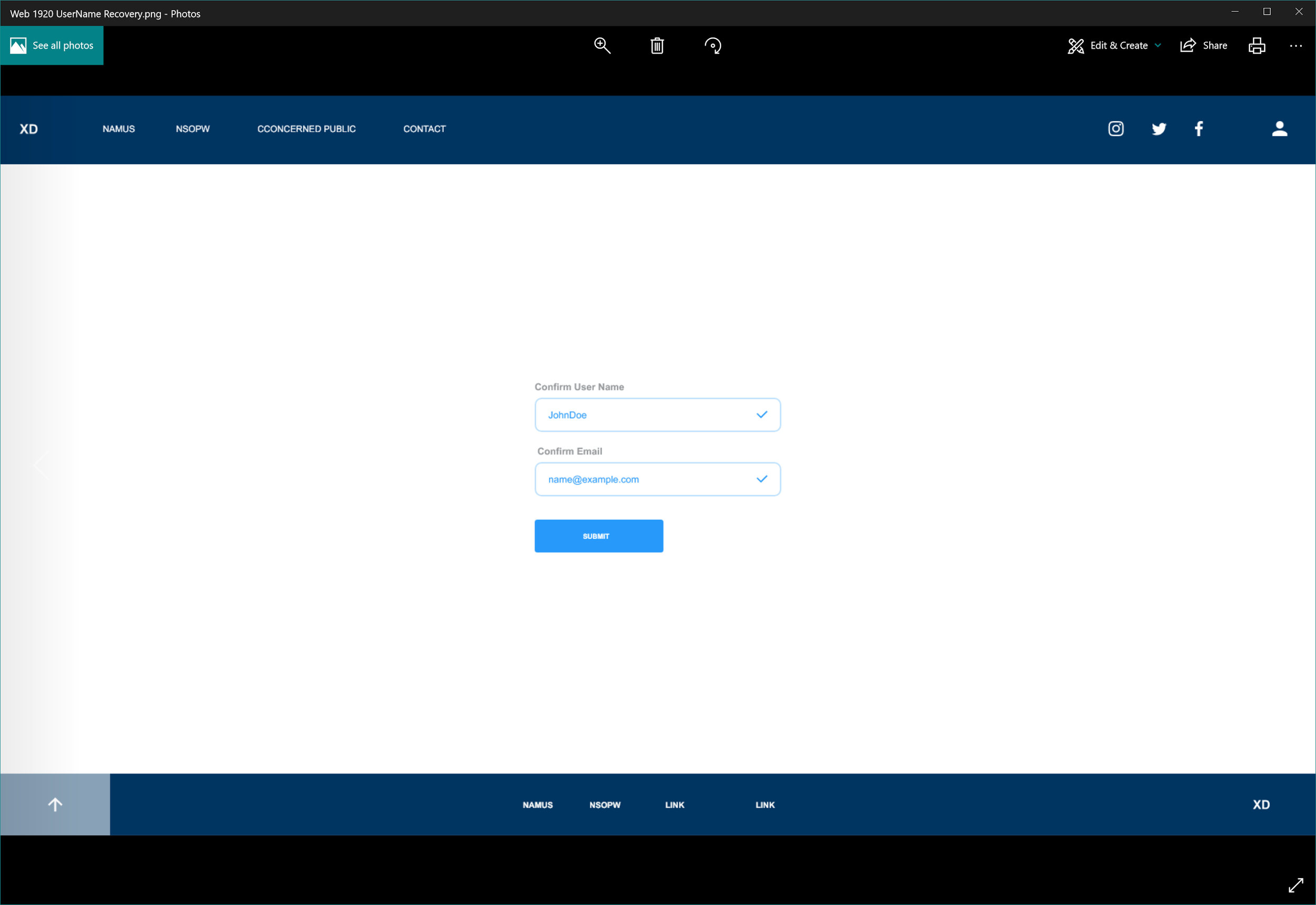
##### LogIn



LogIn page



Password Reset



UserName Recovery

## 4. Functional Requirements Specification

#### 4.1. Stakeholders

#### 4.2. Actors and Goals

#### 4.3. User stories, scenarios and Use Cases

#### 4.4. System Sequence / Activity Diagrams

## 5. User Interface Specifications

#### 5.1. Preliminary Design

#### 5.2. User Effort Estimation

## 6. Static Design

#### 6.1. Class Model

#### 6.2. System Operation Contracts

#### 6.3. Mathematical Model

#### 6.4. Entity Relation

## 7. Dynamic Design

#### 7.1. Sequence Diagrams.

#### 7.2. Interface Specification

#### 7.3. State Diagrams

## 8. System Architecture and System Design

#### 8.1. Subsystems / Component / Design Pattern Identification

#### 8.2. Mapping Subsystems to Hardware (Deployment Diagram)

#### 8.3. Persistent Data Storage

#### 8.4. Network Protocol

#### 8.5. Global Control Flow

#### 8.6. Hardware Requirement

## 9. Algorithms and Data Structures

#### 9.1. Algorithms

#### 9.2. Data Structures

## 10. User Interface Design and Implementation

#### 10.1. User Interface Design

#### 10.2. User Interface Implementation

## 11. Testing

#### 11.1. Unit Test Architecture and Strategy/Framework

#### 11.2. Unit test definition, test data selection

#### 11.3. System Test Specification

#### 11.4. Test Reports per Spring

## 12. Project Management

#### 12.1. 11.1 Project Plan

#### 12.2. 11.2 Risk management

## 13. References

[1] Telephone Records and Privacy Protection Act of 2006, Pub. L. No. 109-476, 120 STAT. 3568 (2006).