

# 5.0 Requirements Specification ---

## 5.1 Introduction

This Software Requirements Specification (SRS) documents the requirements for the CaseX Database Website Application. CaseX is an open source web application and database optimized for the management of physical data. This system is uniquely customized as a data management solution for the Los Angeles Police Department Homicide Library Unit, however it can be used for any enterprise seeking physical data management.

At the core of the CaseX application is the Case Record, which contains metadata about a particular homicide case. CaseX has three main functions: efficient upload of Case Record metadata, interactive visualization and query of Case Record Data, and customizable data export into reports or spreadsheets.

The CaseX system architecture is comprised of a web browser-based user interface (front-end), a server (backend), and a database. The frontend consists of various webpages which allows users to accomplish the three main functions of CaseX. The server backend and database will be designed to allow for concurrent access, efficient query, and data validation.

## Outline of Requirements Specifications

### 5.2 CSCI Component Breakdown

### 5.3 Functional Requirements by CSC

### 5.4 Performance Requirements by CSC

### 5.5 Project Environment Requirements

#### 5.5.1 Development Environment Requirements

#### 5.5.2 Execution Environment Requirements

## 5.2 CSCI Component Breakdown

CSCI CaseX is composed of the following CSCs:

### 5.2.1 Frontend CSC -- web-based GUI for CaseX

5.2.1.1 About Page CSU -- page to display information about CaseX to the public

5.2.1.1.1 Login module -- button to take user to the Login Page CSU

5.2.1.2 Login Page CSU -- page for users to log in to accounts

5.2.1.2.1 Login input module -- input fields for username and password

5.2.1.2.2 Login submit button module -- button send inputs to server

5.2.1.3 Header CSU -- top console to display on each of the following pages

5.2.1.3.1 Menu module -- row of buttons to take user to each of the following pages

5.2.1.3.2 User info module -- displays basic information on user currently logged in

5.2.1.3.2.1 Username module -- text display of user's username

5.2.1.3.2.2 Permissions module -- icon representing user's access tier

5.2.1.3.3 Current page module -- text display of currently opened page

5.2.1.3.4 Logout button module -- button to log user out and return to Login Page

5.2.1.4 Home Dashboard Page CSU -- central control page for users

5.2.1.4.1 Links module -- array of buttons to take user to each of the following pages

5.2.1.5 Input Form Page CSU -- page for adding case entries to the database

5.2.1.5.1 Form select module -- interface for selecting which form format to use

5.2.1.5.2 Manual input module -- interface where users can type data into the form

5.2.1.5.3 Form upload module -- button to upload scanned form to server for automatic data parsing

5.2.1.5.4 Submit module -- button to submit completed case entry to the database

5.2.1.6 Data Explorer Page CSU -- page for querying data and viewing query results

5.2.1.6.1 Query module -- section for making queries on database

5.2.1.6.1.1 Search conditions module -- array of conditions for query

5.2.1.6.2 Table display module -- section that displays the returned case entries

5.2.1.6.2.1 Display filter module -- allows users to see only certain columns

5.2.1.6.2.2 Table module -- two-dimensional array display of data

5.2.1.6.3 Graphic display module -- section for visualizing data

5.2.1.7 Case Page CSU -- page for displaying all available data for a single case

5.2.1.7.1 Case selector module -- choose case by DR number

5.2.1.7.2 Display selector module -- choose display as table or form layout

5.2.1.7.3 Toggle edit module -- button to enable / disable editing of case data

5.2.1.7.4 Data display module -- displays case data in selected layout

5.2.1.8 Admin Console Page CSU -- page for management of database and users

5.2.1.8.1 Users management module -- section for modifying users

5.2.1.8.2 Case database management module -- section for modifying case database

### 5.2.2 Server CSC -- Description

5.2.2.1 NodeJS CSU - modules for handling connections and serving web application

**5.2.2.2 API Routes CUS -- modules for handling requests to the server**

**5.2.3 Database CSC -- database components that will store the data**

**5.2.3.1 Queries CSU -- modules will run queries on the stored data**

## 5.3 Functional Requirements by CSC

The CaseX Web App will allow members of the LAPD Homicide Library Unit to record case data that will be stored in the system's database. In the requirements that follow, "user" is understood to be members of the Unit (detectives and officers). Users will have the ability to retrieve and query this same information on demand as well as generate reports on stored data. The system will also have the ability to easily import pre-formatted data.

### 5.3.1 Frontend

5.3.1.1 The Frontend shall display an about page to provide information about CaseX to the public.

5.3.1.2 The Frontend shall display a login page for the user.

5.3.1.2.1 The Frontend shall provide instructions for how to use the login page.

5.3.1.2.2 The Frontend shall provide text entry fields into which the user may type values.

5.3.1.2.3 The Frontend shall provide error checking on the text entry fields such that the user is provided immediate feedback in case of erroneous entry.

5.3.1.3 The Frontend shall direct users to the home dashboard page after successful login.

5.3.1.4 The Frontend shall restrict users without admin access from the admin console page.

5.3.1.5 The Frontend shall include a header bar at the top of the page for all pages except the Login and About pages.

5.3.1.5.1 The header bar shall include the CaseX logo, which will link to the login page.

5.3.1.5.2 The header bar shall include the name of the currently logged in user.

5.3.1.5.3 The header bar shall include an icon which indicates the access level of the user currently logged in.

5.3.1.5.4 The header bar shall include a button to log out, bringing them to the login page.

5.3.1.6 The Frontend shall have a navigation bar to switch between pages.

5.3.1.6.1 The navigation bar shall display a button linking to the home, forms, explorer, and admin pages.

5.3.1.6.2 The navigation bar shall display the name of the current page in a font larger than the navigations buttons for emphasis.

5.3.1.7 The Frontend shall have an input form(s) page.

5.3.1.7.1 The Frontend shall provide a form section for inputting data into a form organized according to the selected format.

- 5.3.1.7.2 The form section shall provide text entry fields into which the user may type values or select options from pulldown menus. The fields will include but are not limited to DR number, victim name, bureau, division, coroner case number and court case number.
- 5.3.1.7.3 The Frontend shall provide the ability to upload scanned forms as an alternative form-filling method.
- 5.3.1.8 The Frontend shall have a data explorer page.
  - 5.3.1.8.1 The data explorer page shall provide the ability to make queries on a database.
  - 5.3.1.8.2 The data explorer page shall save frequently used queries for reuse.
  - 5.3.1.8.3 The data explorer page shall provide an interface to make queries on the data.
  - 5.3.1.8.4 The data explorer page shall display case entries that match the query as a table of columns and rows.
  - 5.3.1.8.5 The data explorer page shall provide the ability to hide columns in the table.
  - 5.3.1.8.6 The data explorer page shall allow the user to export the filtered case data into formats including, but not limited to CSV, Excel spreadsheet, and PDF.
  - 5.3.1.8.7 The data explorer page shall allow the user to link to the individual case page of a particular table entry by clicking on the row.
- 5.3.1.9 The Frontend shall have an individual case page.
  - 5.3.1.9.1 If accessed from the navigation bar, the individual case page shall allow the user to choose a case by a DR number.
  - 5.3.1.9.2 The Frontend shall provide the option to display case data as a table or form layout.
  - 5.3.1.9.3 The Frontend shall allow editing of case data only for users with administrative access.
- 5.3.1.10 The Frontend shall have an admin console page.
  - 5.3.1.10.1 The admin console page shall only be accessible to administrators.
  - 5.3.1.10.2 The admin console page shall have a section for modifying users.
    - 5.3.1.10.2.1 The users section shall allow admins to add and remove users.
    - 5.3.1.10.2.2 The users section shall allow admins to modify users' access level.
  - 5.3.1.10.3 The admin console page shall have a section for modifying case database.
    - 5.3.1.10.3.1 The database modification page shall allow admins to modify the values of active case data.
    - 5.3.1.10.3.2 The database modification page shall allow admins to delete case entries.
    - 5.3.1.10.3.3 The database modification page shall allow admins to modify the name of each attribute (column).
    - 5.3.1.10.3.4 The database modification page shall allow admins to add new attributes and delete existing attributes.
    - 5.3.1.10.3.5 The database modification page shall ask for confirmation from admins before deleting attributes to prevent accidental deletions.

## **5.3.2 Backend**

**5.3.2.1 The Backend shall respond to HTTP requests from the Frontend.**

**5.3.2.2 The Backend shall make a network connection with the database.**

**5.3.2.3 The Backend shall forward queries from the Frontend to the Database.**

**5.3.2.4 The Backend shall forward results of database queries to the Frontend.**

**5.3.2.5 The Backend shall authenticate users.**

**5.3.2.6 The Backend shall grant read and write access in accordance with the access level of the user.**

**5.3.2.6.1 Access levels will include administrator, general access, and read only**

**5.3.2.6.2 Administrators will have complete control over the application, which involves: database structure; all user permissions and privileges; all case data and records.**

**5.3.2.6.3 General access users will have control over input and querying of case data**

**5.3.2.6.4 Read only access users will only be able to query case data**

**5.3.2.7 The Backend shall perform data validation before sending queries to the database.**

**5.3.2.7.1 Validation will include verifying the requested data is of the correct format and verifying syntax (e.g. required fields).**

**5.3.2.8 The Backend shall use Mongoose object modeling for MongoDB documents.**

**5.3.2.9 The Backend shall maintain a network log.**

**5.3.2.10 The Backend network log shall track successful HTTP requests.**

**5.3.2.11 The Backend network log shall track failed HTTP requests.**

**5.3.2.12 The Backend shall respond with a 404 error code when a page or route is not found.**

**5.3.2.13 The Backend shall respond with a 503 error code if the server is overloaded or under maintenance.**

**5.3.2.14 The Backend shall respond with a 504 error code if the server is not receiving a response from the backend servers within the allotted time period.**

**5.3.2.15 The Backend shall respond with a 200 code when an HTTP request is successfully made.**

**5.3.2.16 The Backend shall have a REST API that can be called by the Frontend through an HTTP request.**

**5.3.2.17 The Backend shall import Excel data sheets uploaded by the user.**

**5.3.2.18 The Backend shall parse the Excel data sheets.**

**5.3.2.19 The Backend shall validate the data entered through the Excel data sheet.**

**5.3.2.19.1 Validation will include verifying the correct information is filled out in the Excel cells**

**5.3.2.19.2 Validation will verify that no extraneous data is entered.**

**5.3.2.20 The Backend shall export queried data to an Excel file format.**

5.3.2.21 The Backend shall import data from an Excel file format.

### **5.3.3 Database**

5.3.3.1 The Database shall use MongoDB as the database system.

5.3.3.2 The Database shall perform queries on stored data.

5.3.3.3 The Database shall store case records.

5.4.4.4 The case records will consist of the following information:

- Division
- Bureau
- DR Number
- Victim DR
- Master DR
- Volumes
- Number of cases
- Check out Date
- Notes
- Info added after scan
- Coroner Case #
- Court Case No.
- Overall Adjudication
- Evidence Available/Destroyed
- Date DNA testing completed
- Status DNA sample
- Date DNA testing requested
- DNA CoDIS info
- NIBIN entry date
- Prints: IAFIS/LFIS run date
- Prints Requested date
- Date Occurred
- Date Reported
- RD (Reporting District)
- Case Status
- Case Status Date
- Victim Name
- Victim Sex
- Victim Age
- Victim Description
- Vict supervised release status
- Date Case Last Screened for Evidence
- Date Case Fully Reviewed
- Solvability Factor

- Crime Address
- Weapon / method used
- Motive
- Suspects
- Suspect Age
- Supervised Release Status
- Juvenile Tried as an Adult
- Suspect Names
- Last Modified Date
- Last Modified By

**5.3.3.5 The Database shall store user profile information.**

**5.3.3.6 The user profile will consist of the following information:**

- First Name
- Last Name
- Access Level

**5.3.3.7 The Database shall prevent errors from concurrent data modification.**

**5.3.3.8 The Database shall be hosted on a cloud service.**

**5.3.3.9 The Database shall be accessible only through the CaseX Web Application API.**



## **5.4 Performance Requirements by CSC**

### **5.4.1 User Login should take no more than 5 seconds.**

After inputting login credentials, and successfully passing verification, a user should not wait more than 5 seconds before they are redirected to the home dashboard page.

### **5.4.2 Fast Navigation**

Users should not experience delays while navigating between pages.

### **5.4.3 Quick Search Query Results**

Users should see results of a search within seconds of initiating the search.

### **5.4.4 Modular System Design**

The system design should allow for easy incorporation and modification by other developers.

### **5.4.5 File Upload**

The system should complete PDF file upload and display results within 10 seconds.

### **5.4.6 File Export**

The system should export queried data to an Excel Spreadsheet and initiate download within 10 seconds.

### **5.4.7 Network Crash Recovery**

The system should be able to deal with network crashes.

### **5.4.8 Emergency Backup**

The system should initiate an emergency backup if database or server issues occur.

### **5.4.9 Accessibility**

The system will meet the minimum requirements for software accessibility.

## 5.5 Project Environment Requirements

### 5.5.1 Development Environment Requirements

Category	Requirement
Frontend	Bootstrap, Moment, d3.js
Server	Node.js, npm, Express, Mocha, Jade, ESLint, Istanbul, Mocha, Nodemon, Mongoose, parallelshell, nodemon
Database	MongoDB

### 5.5.2 Execution Environment Requirements

Category	Requirement
Frontend	Web Browser
Server	Cloud-hosted server
Database	Cloud-hosted database