

9-2 Final Project - System Proposal Document

John C. Aghadiuno

Southern New Hampshire University

Professor Allyson Heisey

07/01/2018

Contents

Introduction.....	4
Background.....	4
Problem Statement.....	4
Audience.....	5
Systems Requirements.....	5
Requirements Models.....	5
Output.....	5
Input.....	6
Process.....	6
Performance.....	6
Control.....	7
Data Process Model.....	8
Data Flow Diagram.....	9
Data Dictionary.....	10
Object Modelling.....	13
Use Cases and Actors.....	14
Use Case Diagram.....	15
State Transition.....	16

Systems Design.....	17
Specification	17
Requirements Models	20
Data Design.....	22
Logical Data Model	23
Data Flow Diagram.....	30
Data Dictionary.....	31
Object Models.....	34
User Interface Design	36
Sitemap	36
Homepage Template	36
People Search Template	37
Person Details Template.....	38
Case Details Template.....	39
System Architecture	40
Feasibility Analysis.....	42
Project Plan	46
Work Breakdown Structure.....	46
Project Monitoring and Control Plan.....	47

Definition of Done	47
Timeline	49
Resource Sheet.....	50
Cost Sheet	51
References.....	52

Introduction

Background

CPCS currently maintains three separate “case management” systems in use by staff (i.e., CASEY, TRIS, and CMS). They share a core set of functionalities, mandated by statute and ethical obligation. They also attempt, with varying degrees of success, to address a separate tier of functions aimed at improving staff efficiency and practice, as well as the allocation of resources. The agency’s recent growth and a desire to better leverage our data for the good of our clients has resulted in the addition of functions to this separate tier. Design of the current systems, however, impedes implementation of such expanded functionality.

Problem Statement

Due to deficiencies in the current system design and the lack of standard data input practices, most of the data collected for purposes other than case “counts” (i.e., everything except case numbers and runsheet narratives) is of poor quality, and in their current form they do not seem to be either complete or accurate. The data are inconsistent when considered across multiple offices/working groups (even within the same division), as various groups adhere to differing practices. Additionally, the current system introduces demonstrably unnecessary work

and complexity (e.g., duplicate data entry). Consequently, users have had to adapt their workflows to the system rather than having the system support the best workflows.

Based on the above deficiencies in the current systems, it is necessary to construct a system built on a more robust understanding of our data and user needs. The purpose of this document is to outline the business requirements for such a solution, one that would replace the existing systems with a single, well-designed system capable of ensuring the following:

- **Improved efficiency:** ensure that users are required to enter data in the system only once, and that they can enter what they need, and that they can easily find the information they require
- **Improved data quality:** create robust data validation mechanisms that ensure the system can use the data that users enter
- **Effective, user-friendly design:** ensure that users are able to easily view necessary system information, and navigate seamlessly through the system
- **Improved system flexibility:** ensure the system is adaptable to the evolving needs of CPCS.

Audience

The intended audience for this document shall be the Project Sponsors, Project Team, IT Development team, Quality Assurance team and other interested parties, including anyone who will be involved in support and maintenance of this application.

Systems Requirements

Requirements Models

Output

- The system must provide UIs for reporting current and historical case data statewide.

- The system must provide functionality for filtering and aggregating case data.
- The system must be simple and intuitive, and unambiguous output.
- The system must provide authentication to access its resource.
- The system must provide authorization and implement access control to its resource.
- The system must provide secure access online and be available to any device.
- The system must allow access to customizable reports based on staff roles.

Input

- The system must allow for case management by staff.
- The system must store case activity and event logs.
- The system must capture relevant information about staff reporting preferences.
- The system must allow management to assign staff to roles.

Process

- The system must integrate with the existing case systems.
- The system must be able to generate reports based on user provided metrics.
- The system must be able to pull data from existing systems and aggregate data.
- The system must have built-in algorithms to enforce agency rules and objectives.
- The system must be able to export data in various formats.

Performance

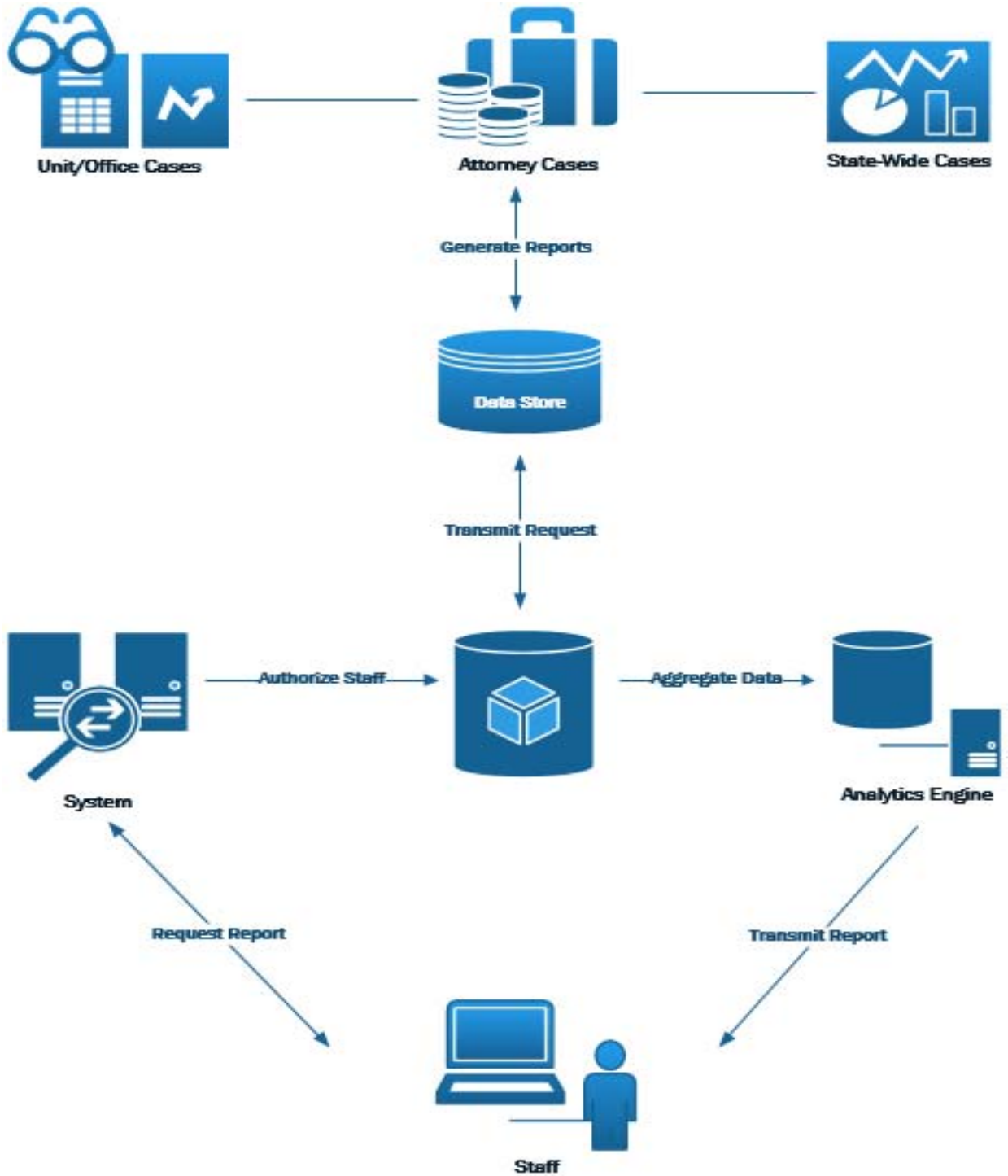
- The system must be able to scale and support thousands of users simultaneously.

- The system must be able to handle core business processes.
- The system must have a friendly interface, easy to use and accessible on any device.

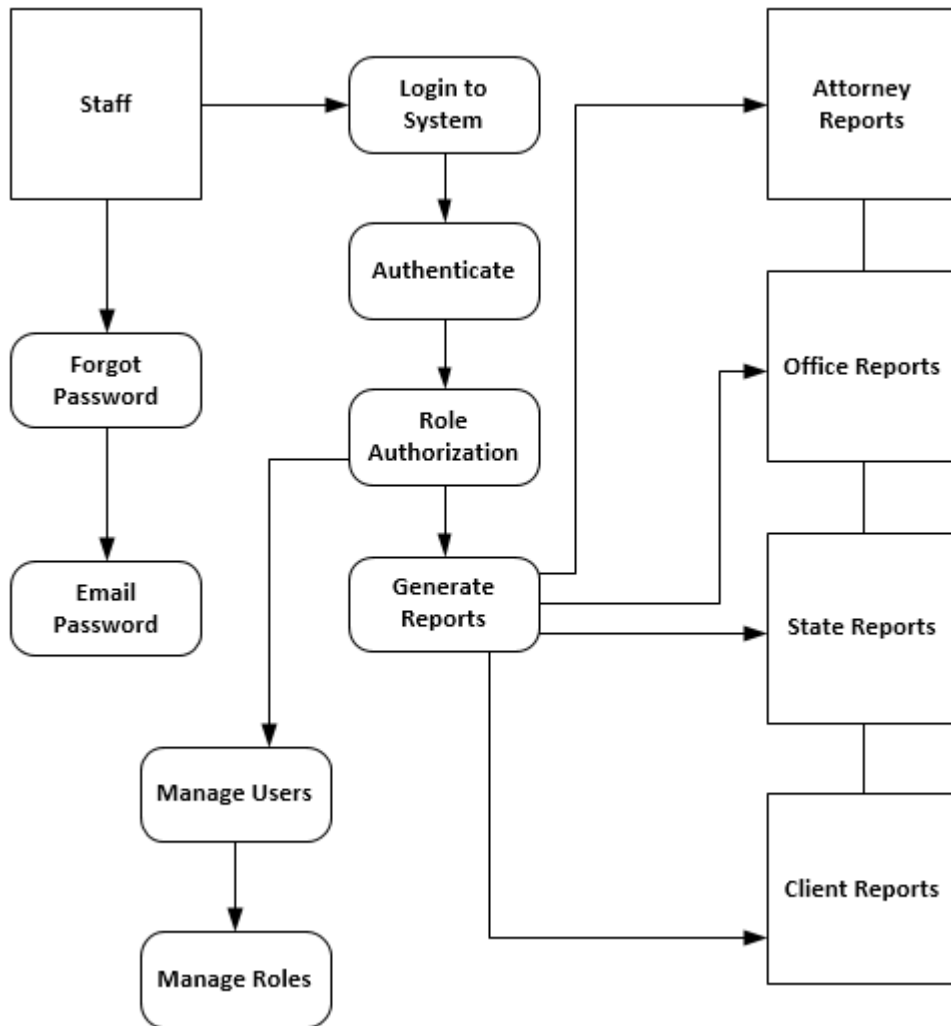
Control

- The system must provide a secure single log-in enterprise-wide.
- The system must provide authentication, authorization and access control mechanisms consistent with business rules.
- The system must provide for the security, integrity, reliability, retention and back-up of data in accordance with applicable state laws, governing bodies, industry standards and business rules.

Data Process Model



Data Flow Diagram



Data Dictionary

[-]	[🔍]	zeno.Case (System-Versioned)
[-]	[📁]	Columns
	🔑	CaseId (PK, int, not null)
	[📄]	CaseDescription (nvarchar(500), null)
	[📄]	OldCaseNumber (nvarchar(50), null)
	🔗	CaseTypeId (FK, int, null)
	🔗	DispositionId (FK, int, null)
	[📄]	CaseOpenDate (date, not null)
	[📄]	CaseCloseDate (date, null)
	[📄]	CaseNextDate (date, null)
	[📄]	CaseFiledDate (date, null)
	[📄]	CaseNote (nvarchar(1000), null)
	[📄]	IsRemoved (bit, not null)
	🔗	CaseOpenTypeID (FK, int, null)
	[📄]	BailAmount (nvarchar(50), null)
	[📄]	BailConditions (nvarchar(150), null)
	[📄]	BailAdmitted (bit, not null)
	[📄]	IsCurrentlyInCustody (bit, not null)
	🔗	CurrentCustodyLocation (FK, int, null)
	[📄]	LastUpdateUser (nvarchar(100), not null)
	[📄]	ValidFrom (datetime2(2), not null)
	[📄]	ValidTo (datetime2(2), not null)
[+]	[📁]	Keys
[+]	[📁]	Constraints
[+]	[📁]	Triggers
[+]	[📁]	Indexes
[+]	[📁]	Statistics

Select a page

- 🔗 General
- 🔗 Permissions
- 🔗 Change Tracking
- 🔗 Security Predicates
- 🔗 Extended Properties

Connection

Server:
cpcs-zeno.database.windows.net

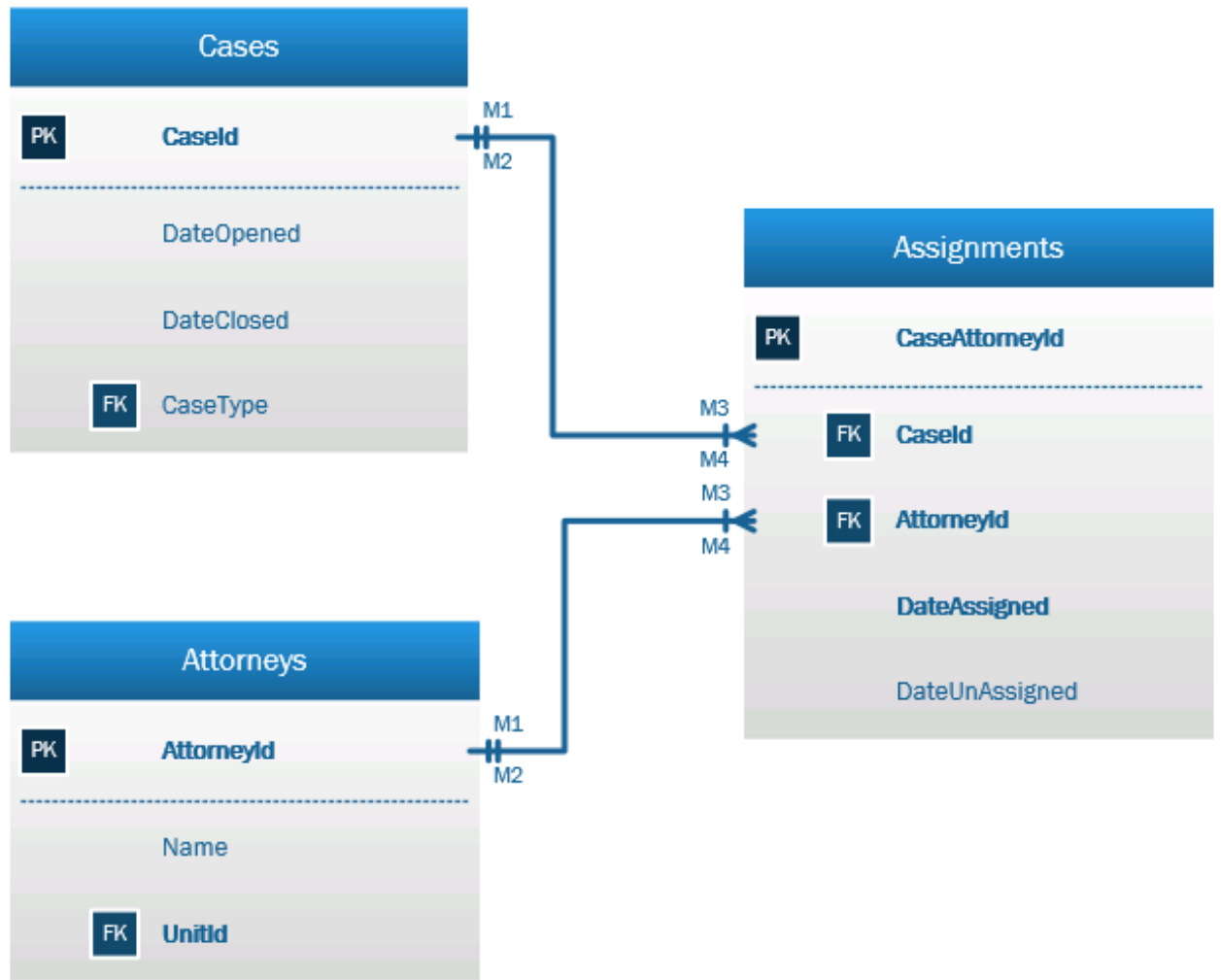
Connection:

Script ? Help

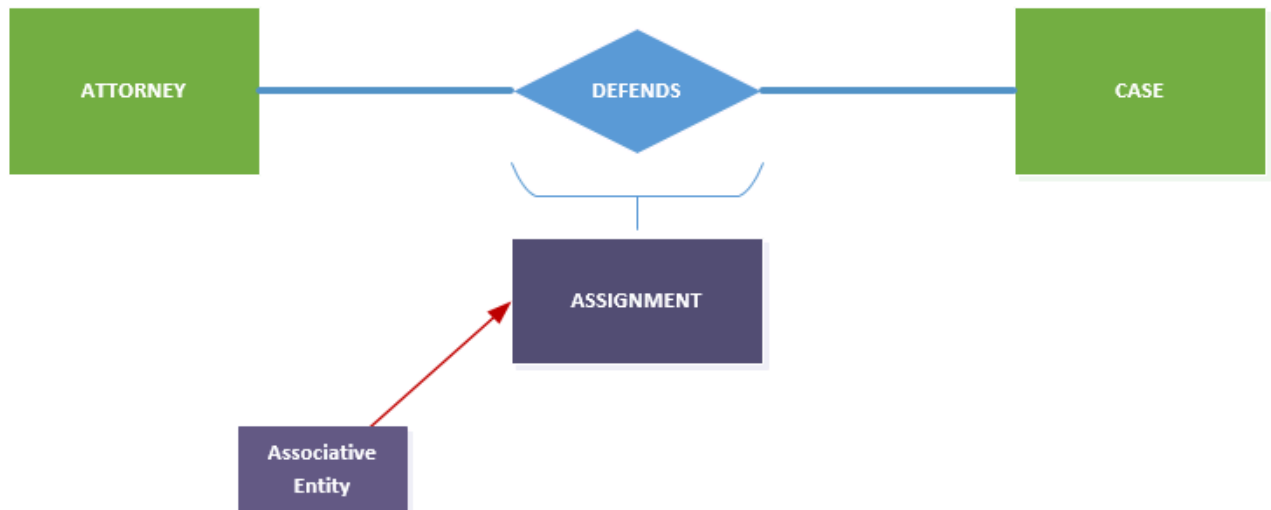
🔍 ⬆ ⬇ 📄

Current connection parameters	
Database	Zeno_DEV
Server	cpcs-zeno.database.windows.net
User	jaghadiuno@publiccounsel.net
Description	
Schema	zeno
Name	Case
Created date	5/10/2018 3:21 PM
System object	False
GraphDB	
Table is a node table	False
Table is an edge table	False
Options	
Table temporal type	SystemVersioned
Durability	SchemaAndData
Memory optimized	False
ANSI NULLS	True
Quoted identifier	True

Associative Entity



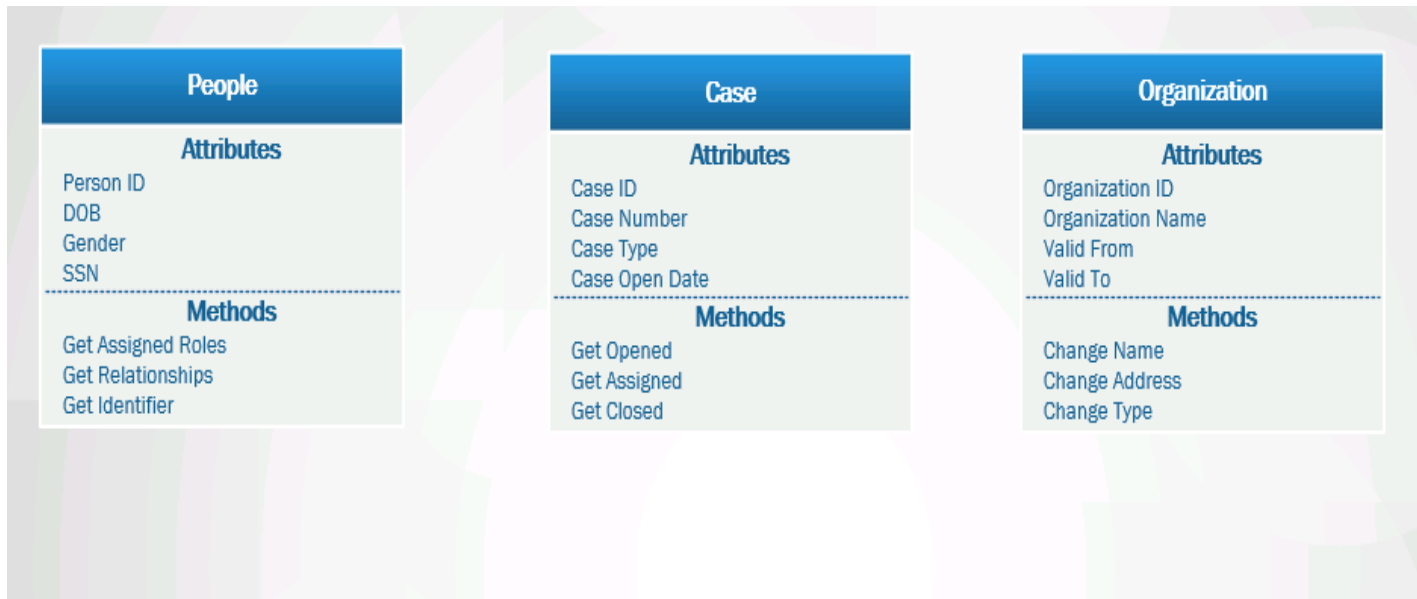
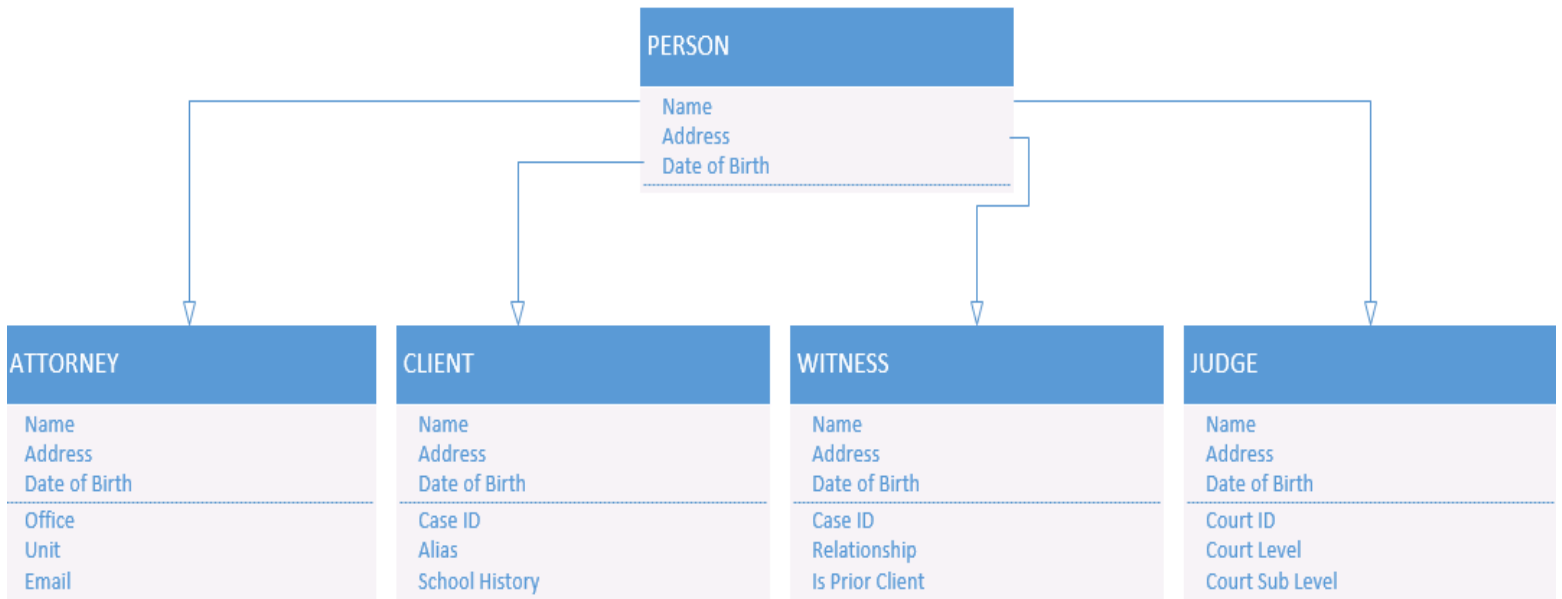
Notation Entity



Object Modelling

Use appropriate object modeling techniques and tools to describe the system requirements.

Every person in the database will inherit from the PERSON class, including attorneys, clients, investigators, judges, defendants, witnesses, etc. While they inherit and share a common set of characteristics, as can be seen in the models below, they can possess their own characteristics.



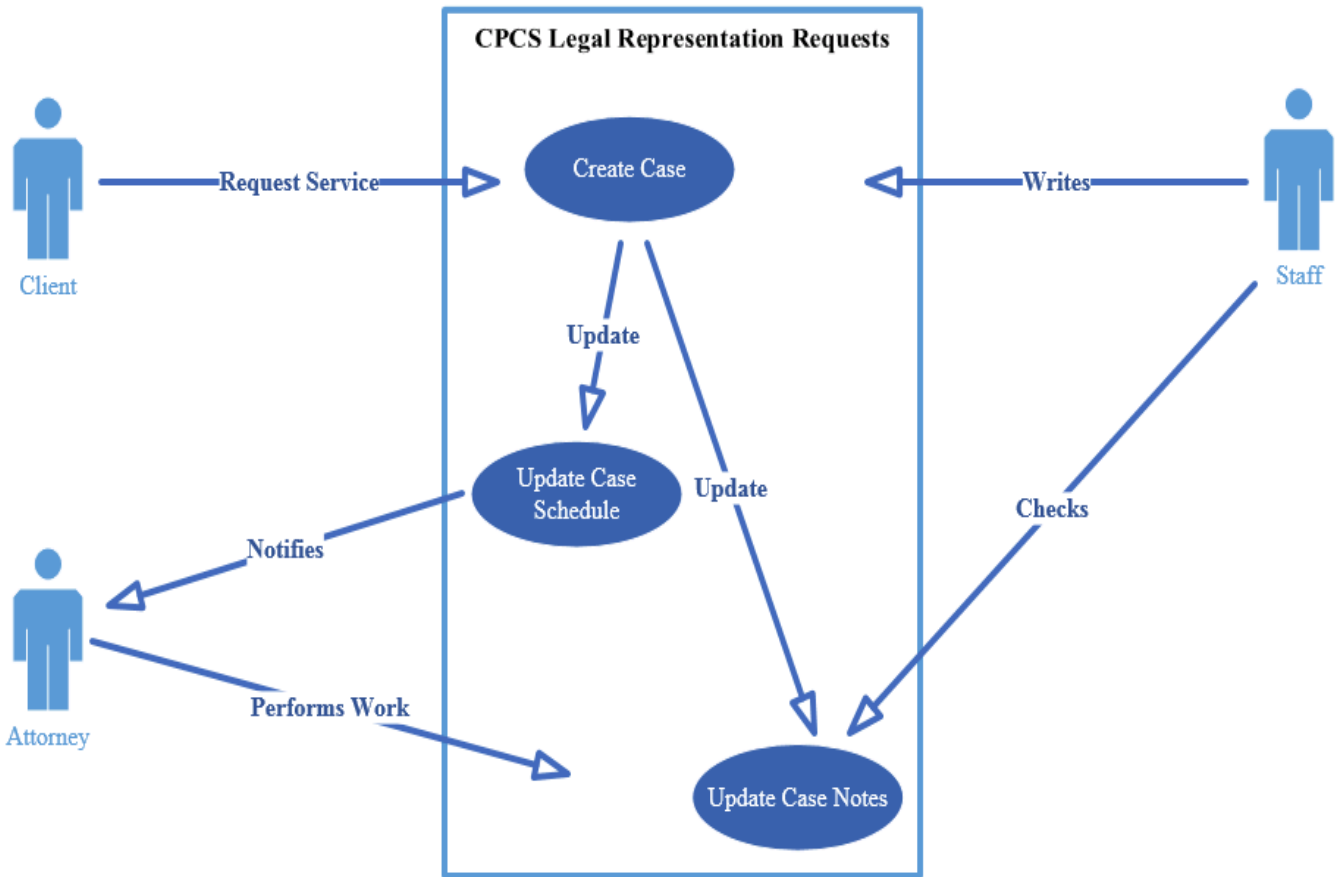
Use Cases and Actors

Primary use cases and actors for the ZENO system include:

1. ATTORNEY (actor) can VIEW OWN CASE REPORTS (use case).

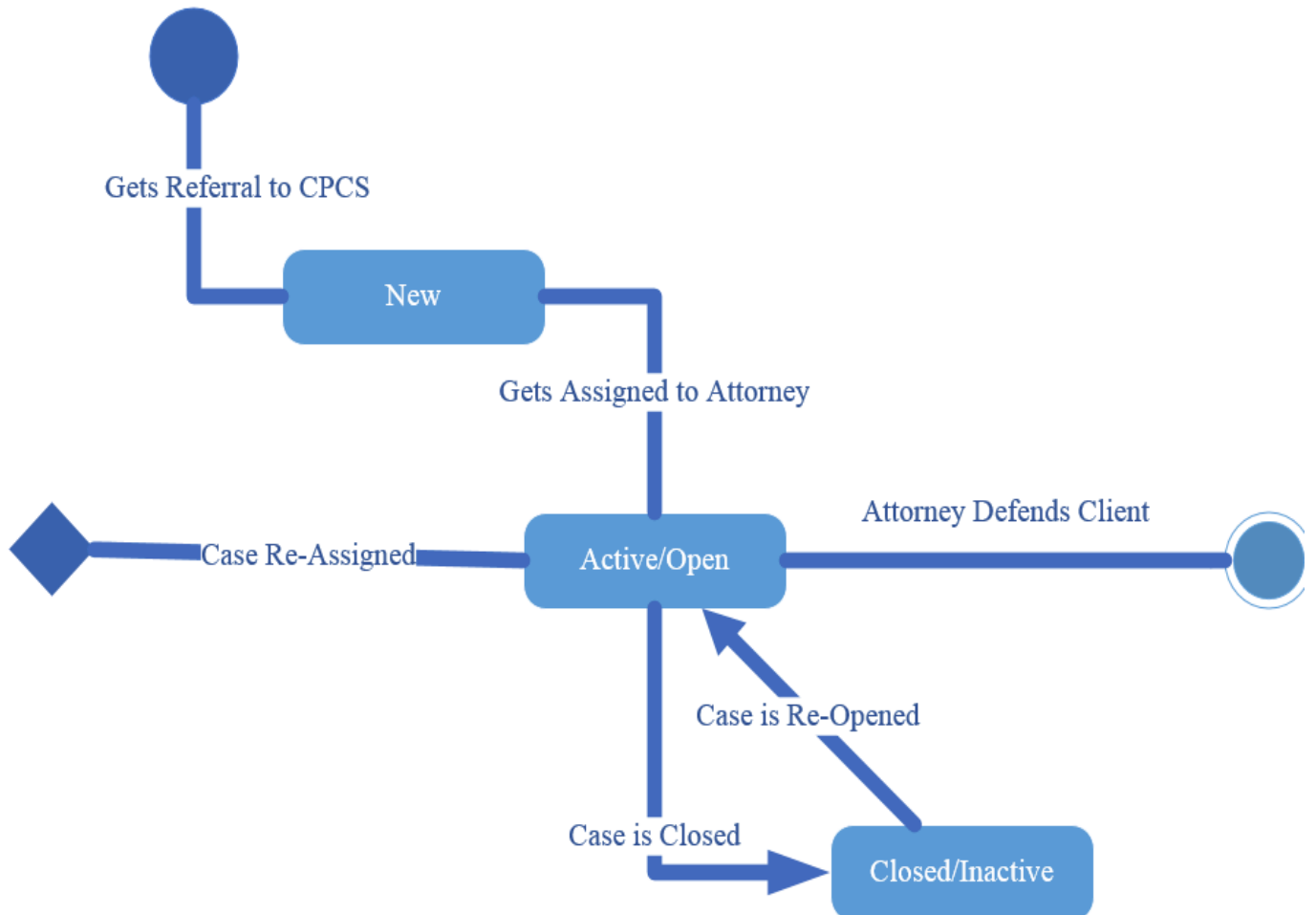
- 2. MANAGER (actor) can VIEW ALL CASE REPORTS (use case).
- 3. OFFICE ASSISTANT (actor) can PRINT CASE REPORTS (use case).

Use Case Diagram



State Transition

State Transition Diagram for Client Case



Systems Design

Specification

<p>The reporting application will provide UIs for reporting current, and/or historical case data statewide. These reports will also provide additional functionality for filtering and aggregating case data.</p>	
1	<p>Given a set of filters (e.g., Division, Unit, Office, date) the <i>Events</i> screen will return a list or graphical display of aggregate counts for the given filters grouped by filtered agency entities. When comparing multiple date ranges, aggregate counts will be replaced by differences.</p>
2	<p>Filtering characteristics should include:</p> <ul style="list-style-type: none"> ● Period (e.g., custom, today, yesterday, last week...) <ul style="list-style-type: none"> ○ From (Starting Date) ○ To (Ending Date) ● Comparison (i.e., no comparison, compare to previous period, and compare to previous year) <ul style="list-style-type: none"> ○ When not “no comparison” <ul style="list-style-type: none"> ■ From (Starting Date) ■ To (Ending Date) ● Division (e.g., PD, CAFL, YAD, MH) ● Unit (e.g., trial, appeals) ● Office (e.g., Lowell District) ● Staff Detail
3	<p>The <i>Period</i> filter should present a set of predefined periods including:</p> <ul style="list-style-type: none"> ● Custom ● Today ● Yesterday ● Last Week ● Last Month ● Q2 FY2017 ● Q1 FY2017 ● Q4 FY2016 ● Q3 FY2016 . . . <p>Where the divisions by quarter are the preceding 8 quarters, excluding the current quarter.</p>

4	The values of the <i>From</i> and <i>To</i> dates associated with <i>Period</i> are automatically set after selecting a period. If they are altered, the <i>Period</i> selection is set to <i>Custom</i> .
5	The <i>Comparison</i> selection can be: <ul style="list-style-type: none"> ● No comparison ● Custom ● Compare to previous period ● Compare to previous year
6	The <i>Comparison's From</i> and <i>To</i> fields only appear when <i>Comparison</i> is not equal to "No comparison"
7	The values of the <i>From</i> and <i>To</i> dates associated with <i>Comparison</i> are automatically set after selecting a <i>..previous period</i> or <i>...previous year</i> . If they are altered, the selection is set to <i>Custom</i> .
8	The span of time covered by the <i>From</i> and <i>To</i> dates associated with <i>Comparison</i> must be the same as that covered by the <i>From</i> and <i>To</i> dates associated with <i>Period</i> .
9	The <i>Division</i> , <i>Unit</i> , and <i>Office</i> fields should be pulldown menus.
10	All pulldown values should default to ALL.
11	The <i>Division</i> , <i>Unit</i> , and <i>Office</i> pulldowns should allow for multiple selections.
12	The available <i>Unit</i> options should be contingent of selected <i>Divisions</i> , and the available <i>Office</i> options should be contingent on the selected <i>Unit</i> . This is to avoid logical impossibilities that would lead to null results (e.g., <i>Division</i> = CAFL and <i>Office</i> = Lowell District).
13	<i>Staff Detail</i> should be a pulldown, defaulting to all, and allowing for multiple selects (e.g., attorney, SSA, investigator).
14	Filtered results should be displayed in four slices: <ul style="list-style-type: none"> ● Mixed Division, Unit and Office List ● Staff List ● Graph ● Cases List
15	The <i>Mixed Division, Unit and Office List</i> should display <ul style="list-style-type: none"> ● A nested list of the selected Divisions, Units and Offices ● As well as a TOTALS row at the top, aggregating values for the following rows. Each row should contain columns noting <ul style="list-style-type: none"> ● Start: The number of open cases at the Start of the Period ● Asd: The number of cases Assigned during the Period that are not Probation or

	<p>Bail</p> <ul style="list-style-type: none"> ● Prob: The number of Probation cases assigned during the Period ● Bail: The number of Bail cases assigned during the Period ● Tch: The number of Touched Cases (start + asd + prob + bail) ● Clsd: The number of cases that were closed during the Period ● End: The number of cases open at the end of the Period.
16	When fetching search results the system should display a loading message to make clear that a search is underway. Ideally, this would take the form of a progress bar to indicate how much time is left.
17	<i>Mixed Division, Unit and Office List</i> counts should count each individual case, not each time a staffer touches a case. So if two attorneys worked the same case, it would show up only once.
18	The content of the <i>Mixed Division, Unit and Office List</i> is contingent on the filtering selections.
19	The content of the <i>Staff List</i> is contingent on those rows selected in the <i>Mixed Division, Unit and Office List</i> . It should be possible to select more than one such row.
20	The <i>Staff List</i> should display rows for each involved staff member with subdivisions for offices. Office subdivisions should not include counts. However, each staff row should include the same columns and counts as the <i>Mixed Division, Unit and Office List</i> but tailored for the staff member.
21	The <i>Staff List</i> should count each time a staffer touches a case. So if two attorneys worked the same case, it would show up as a single case in each attorney's row.
22	The report will show either <i>Details</i> or <i>Graph</i>
23	When <i>Comparison</i> is not <i>No Comparison</i> , the counts in rows will show the difference between the older period and the newer period. For example, if period 1 had 20 touched cases and period 2 had 19, the displayed number would be -1.
24	When <i>Comparison</i> is not <i>No Comparison</i> , the Graph will display lines for period 1 and period 2.
25	Graphed data will plot counts for the number of open cases. That is, the Y axis should start with the number of open cases at the beginning of the period, going up with newly opened cases and going down with newly closed cases.
26	The graph will contain data for the most granular selection above. For example, if Lowell District is selected in <i>Mixed Division, Unit and Office List</i> and there no selection in the <i>Staff List</i> , the counts will be those of all Lowell District cases matching

	the filtering criteria. If, however, Attorney Smith is selected in the Staff List, the counts will be those of Attorney Smith's cases.
27	The start and end dates on graphed data will be set by the min and max opening dates found in matching cases.
28	The granularity of dates in graphed data will allow for user editing. <i>For example, the x axis could show individual days, weeks, months or quarters, depending on the user selection. These delineations would count cases in groupings. So, if the divisions were by weeks, the Y axis would display counts per week.</i>
29	The granularity of date data in graphed data should be a function of the difference between the start and end date of the graph. <i>For example, if the time spans days, the level of detail should be days, if it spans years, quarters or months would be appropriate.</i>
30	The <i>Case Grid</i> should display a row for each case encompassed by the above selections, with columns for <ul style="list-style-type: none"> ● Case Number ● Docket Number ● Client Name ● The "most important" charges (limited by space) ● The case's office ● The opening date ● The attorney(s) on the case ● The closing date ● An indicator if there was an adverse disposition on any charge
31	The results should include a <i>total number of matches</i> as an output.
32	When the total number of matches is sufficiently small, there should be the option to output results to a csv or xls file.
33	When a case number is displayed, it should be a hyperlink to the case's details as seen in the current system (i.e., CASE, TRIS or CMS).

Requirements Models

Output

- The system must provide UIs for reporting current and historical case data statewide.

- The system must provide functionality for filtering and aggregating case data.
- The system must be simple and intuitive, and unambiguous output.
- The system must provide authentication to access its resource.
- The system must provide authorization and implement access control to its resource.
- The system must provide secure access online and be available to any device.
- The system must allow access to customizable reports based on staff roles.

Input

- The system must allow for case management by staff.
- The system must store case activity and event logs.
- The system must capture relevant information about staff reporting preferences.
- The system must allow management to assign staff to roles.

Process

- The system must integrate with the existing case systems.
- The system must be able to generate reports based on user provided metrics.
- The system must be able to pull data from existing systems and aggregate data.
- The system must have built-in algorithms to enforce agency rules and objectives.
- The system must be able to export data in various formats.

Performance

- The system must be able to scale and support thousands of users simultaneously.

- The system must be able to handle core business processes.
- The system must have a friendly interface, easy to use and accessible on any device.

Control

- The system must provide a secure single log-in enterprise-wide.
- The system must provide authentication, authorization and access control mechanisms consistent with business rules.
- The system must provide for the security, integrity, reliability, retention and back-up of data in accordance with applicable state laws, governing bodies, industry standards and business rules.

Data Design

This document will define the data architecture for the CPCS Case Management System.

The data architecture for a system is comprised of the following:

- Logical/physical data models that detail the structure and relationships between the major entities of the system. The data models defined during data architecture may not be fully flushed out to include every possible column or data point required. However, the primary entities and their relationships will be completed.
- Standards and best practices to be used in designing and developing the data architecture.
- A data migration strategy for migration of data from existing databases into the new database.

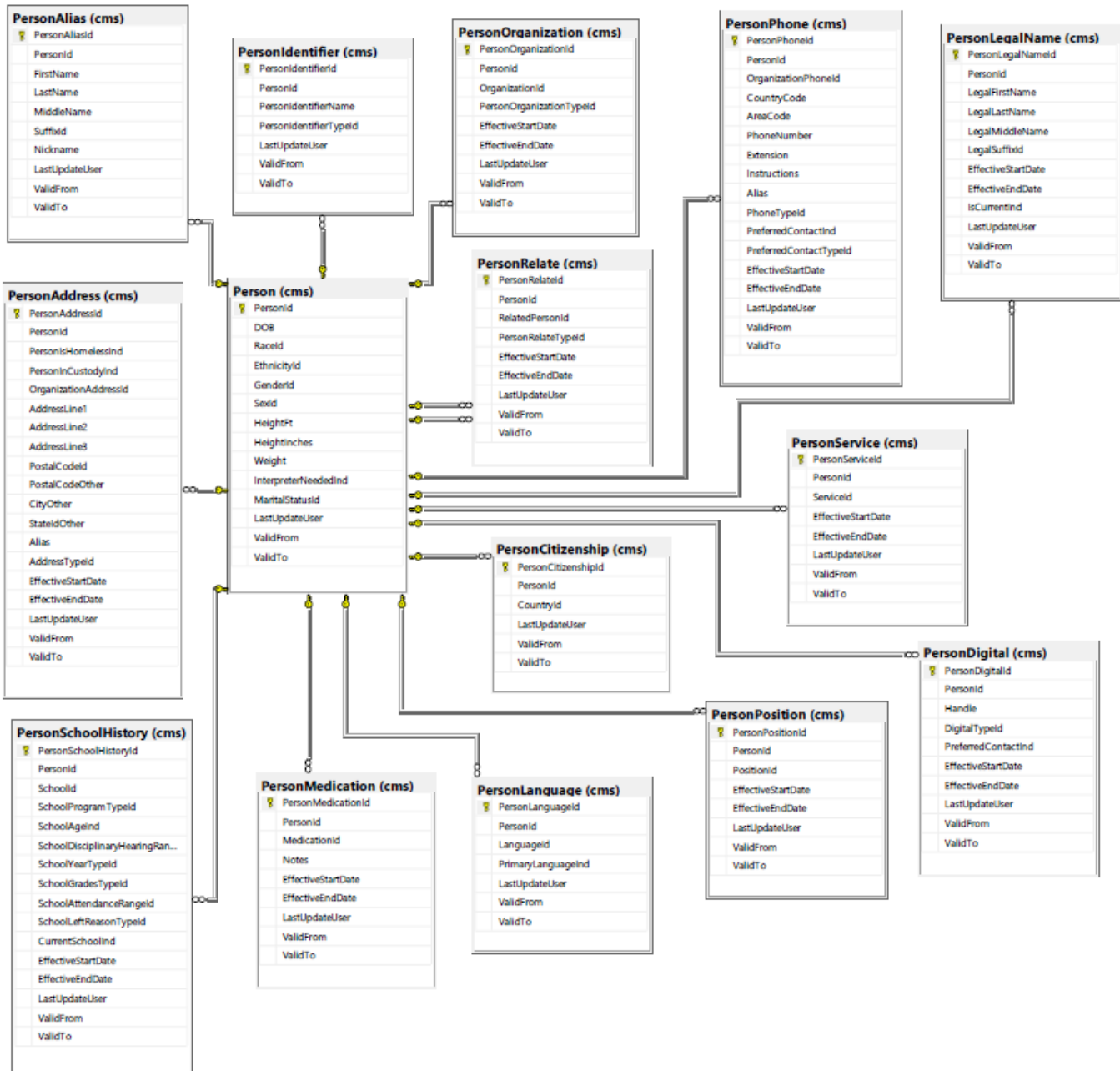
Logical Data Model

In order to develop the data model for this project, an SSDT (SQL Server Data Tools) Database Project was created and used in order to define a set of schemas and tables that will become the persistence layer of the system. In this way, a more “logical” data model was not generated in favor of creating a detailed physical data model that will have immediate use in development of the system. However, the following diagrams and descriptions will act as a logical data model, describing the major entities of the system and how they relate to each other.

Person

The person entity and its assorted one-to-many related entities will store personal and demographic information with various degrees of completeness for any person who has a role in the Case Management System. Examples of person include, but are not limited to, a client, a CPCS attorney, a non-CPCS attorney, a judge, a witness, and a victim. As each person is entered into the system, the system/UI will incorporate workflow and other mechanisms in order to minimize duplicate entry of the same person into the system as much as possible. The details of the duplicate check mechanism are not part of the data architecture documentation. Although there is the concept of a Person being in one or more “position” (see PersonPosition entity below), that will be primarily used as a filtering mechanism when searching for a specific Person. It is when related to other entities that a Person will take on a role as part of that relationship through the PersonRelateType entity. See the Person Main Entity Diagram below followed by descriptions of the nature and usage of each entity.

Person Entity Diagram

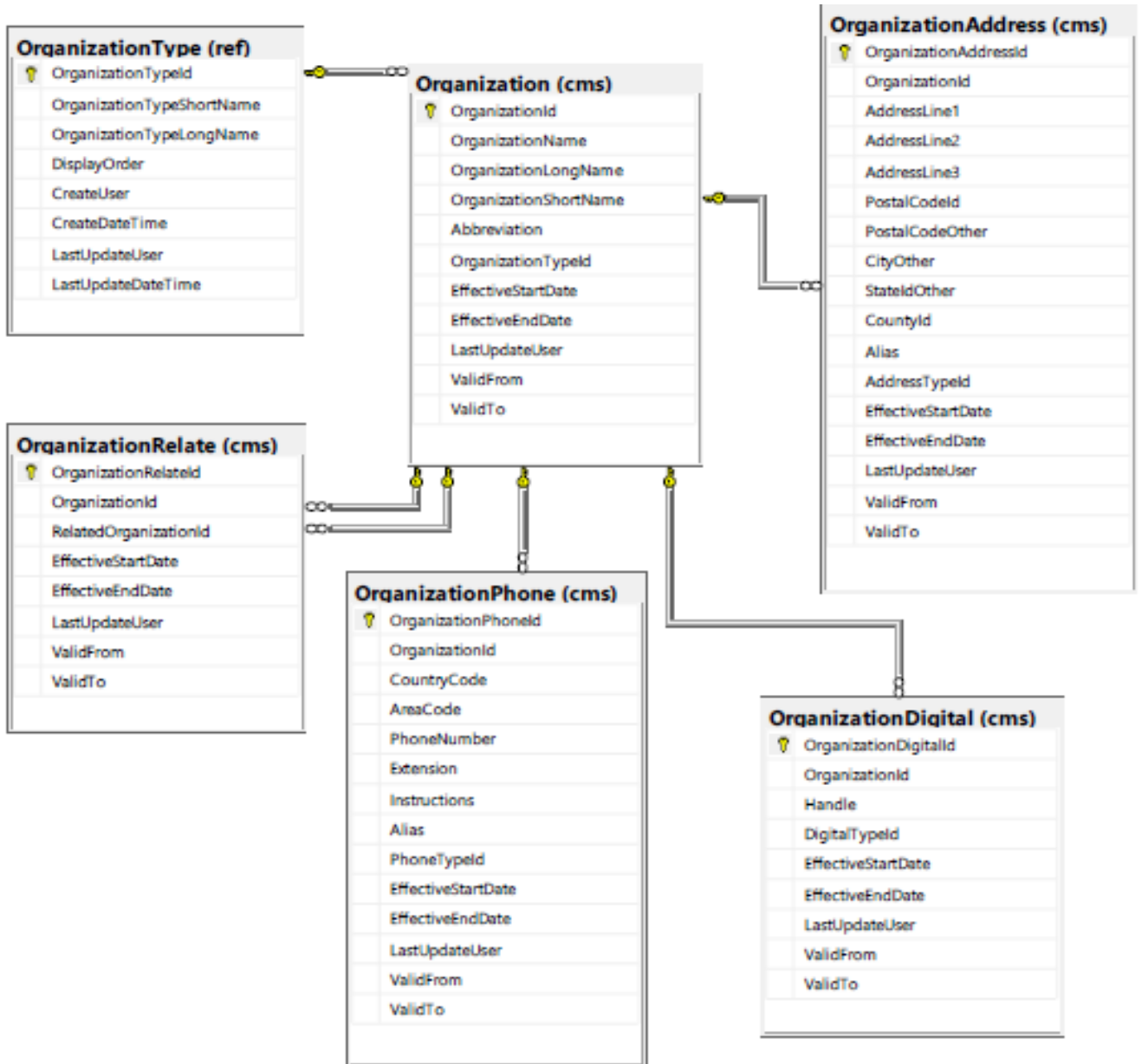


Organization

The organization entity and its assorted one-to-many related entities will store details about any conceptual or physical organization of persons that will be used in the system. There

will be a special set of Organization records that will conceptualize an organization structure for CPCS itself. Although that organizational structure is currently loosely modelled as divisions with regions, units and offices below them, the OrganizationRelate/OrganizationType concept will allow for the hierarchy to be defined at any number of levels over time, while preserving the history of how it was structured at any earlier point in time. This flexible Organization relationship concept can also be used to model flat or hierarchical relationships outside of the CPCS realm. For example, if there is a hierarchy to the Massachusetts court system courts, the courts could be related this way. See the Organization Main Entity Diagram below followed by descriptions of the nature and usage of each entity.

Organization Main Entity Diagram



Matter

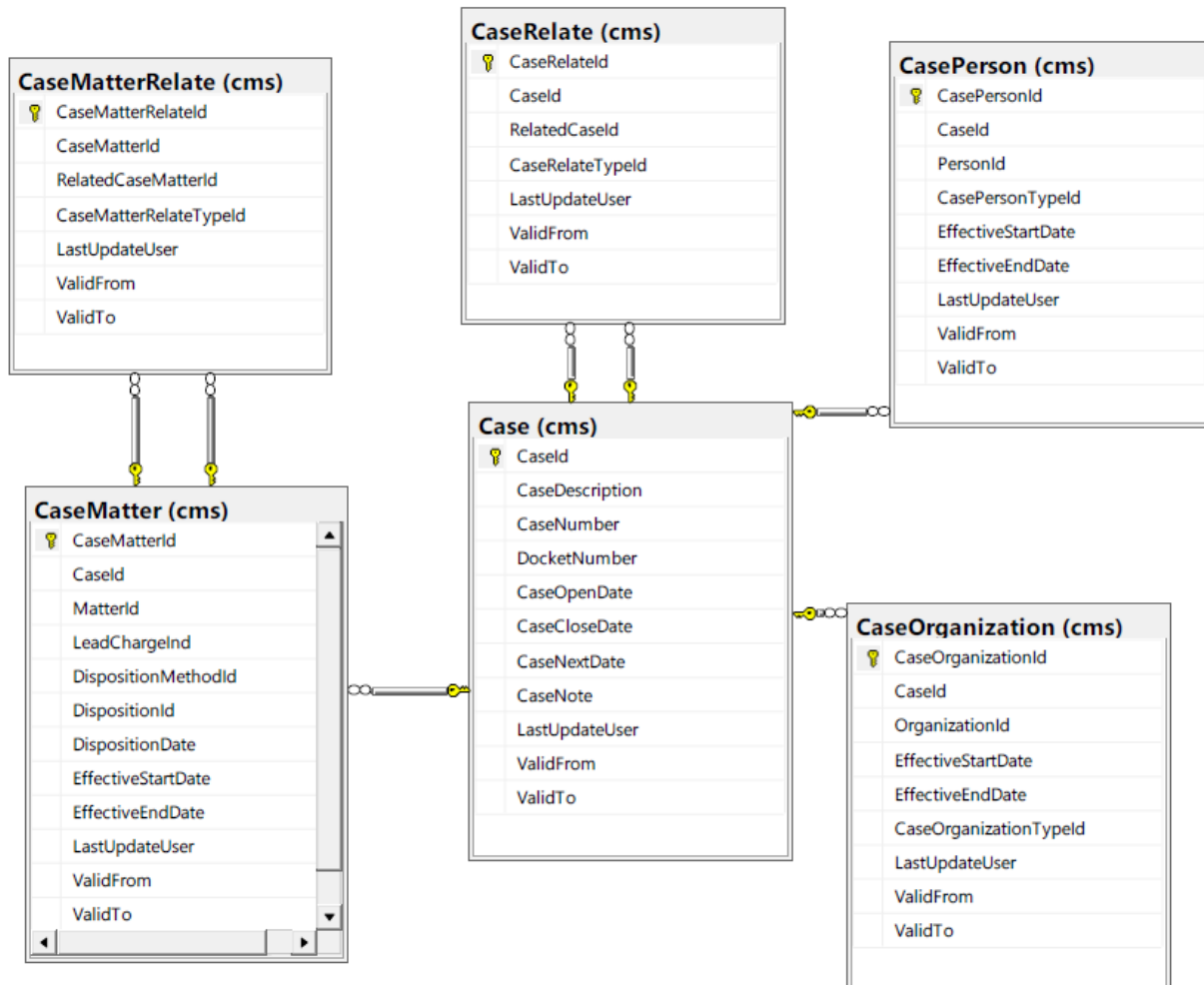
The Matter entity will store details about the possible matters that could be associated with a case. The Matter entity will store both criminal and civil matter data. Think of the Matter

entity as more of a specialized reference table. The Matter entity is somewhat equivalent to the current “Charge Index” in the current system, in that it contains the definitions of the types of matter that could be related to any Case in the system but does not hold the actual instances of matters for each case. See the Matter Main Entity Diagram below followed by descriptions of the nature and usage of each entity. Since the Matter entity contains mostly pointers to the reference tables described below, the Matter Main Entity Diagram depicts both the Matter entity and its related reference tables.

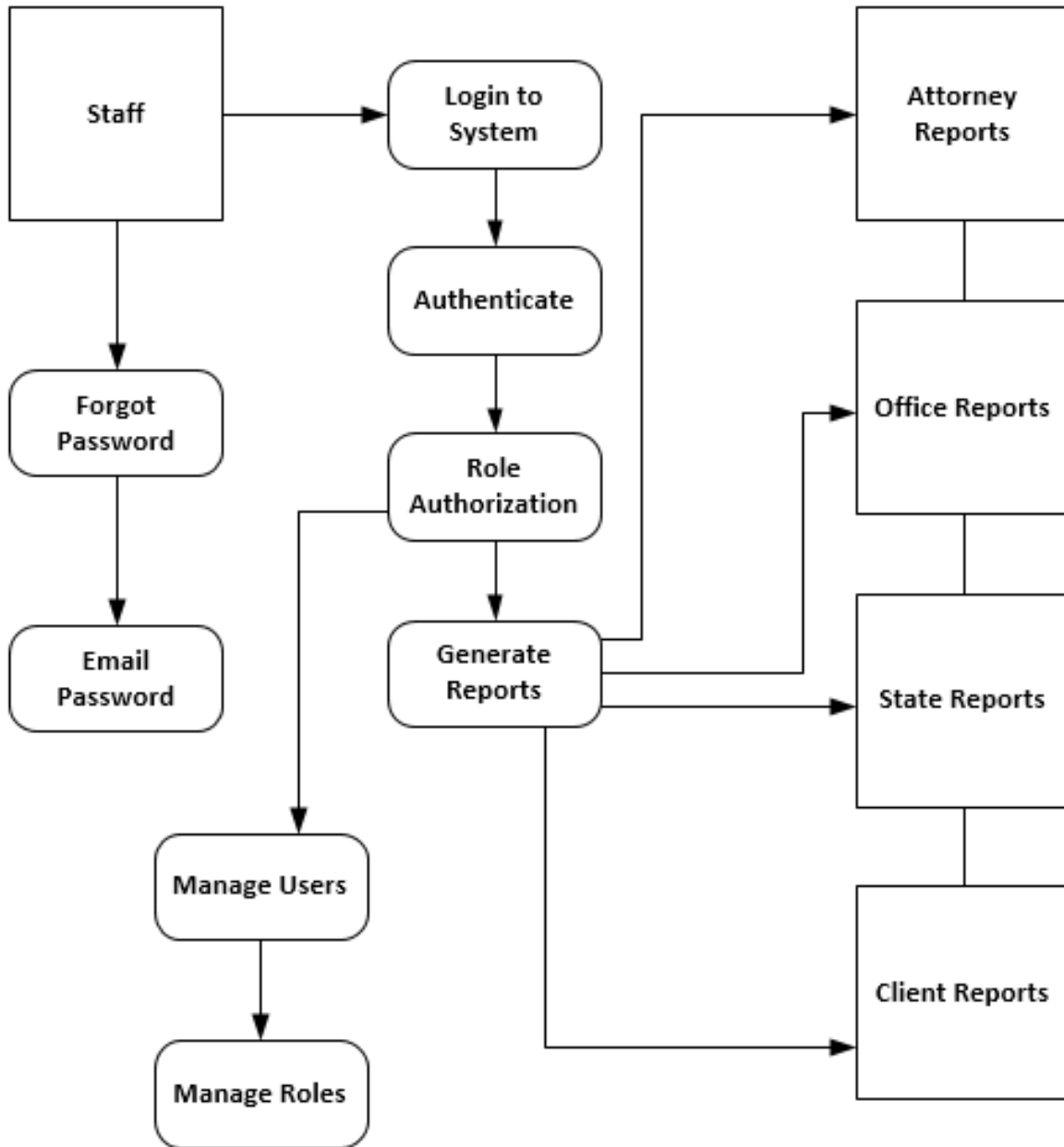
Case

The Case entity and all of its related entities will store all of the information about each case that is handled by CPCS in the Case Management system. See the Case Main Entity Diagram below followed by descriptions of the nature and usage of each entity.

Case Main Entity Diagram



Data Flow Diagram



Data Dictionary

[-]	[🔍]	zeno.Case (System-Versioned)
[-]	[📁]	Columns
	[🔑]	CaseId (PK, int, not null)
	[📄]	CaseDescription (nvarchar(500), null)
	[📄]	OldCaseNumber (nvarchar(50), null)
	[🔗]	CaseTypeId (FK, int, null)
	[🔗]	DispositionId (FK, int, null)
	[📄]	CaseOpenDate (date, not null)
	[📄]	CaseCloseDate (date, null)
	[📄]	CaseNextDate (date, null)
	[📄]	CaseFiledDate (date, null)
	[📄]	CaseNote (nvarchar(1000), null)
	[📄]	IsRemoved (bit, not null)
	[🔗]	CaseOpenTypeID (FK, int, null)
	[📄]	BailAmount (nvarchar(50), null)
	[📄]	BailConditions (nvarchar(150), null)
	[📄]	BailAdmitted (bit, not null)
	[📄]	IsCurrentlyInCustody (bit, not null)
	[🔗]	CurrentCustodyLocation (FK, int, null)
	[📄]	LastUpdateUser (nvarchar(100), not null)
	[📄]	ValidFrom (datetime2(2), not null)
	[📄]	ValidTo (datetime2(2), not null)
[+]	[📁]	Keys
[+]	[📁]	Constraints
[+]	[📁]	Triggers
[+]	[📁]	Indexes
[+]	[📁]	Statistics

Select a page

- General
- Permissions
- Change Tracking
- Security Predicates
- Extended Properties

Connection

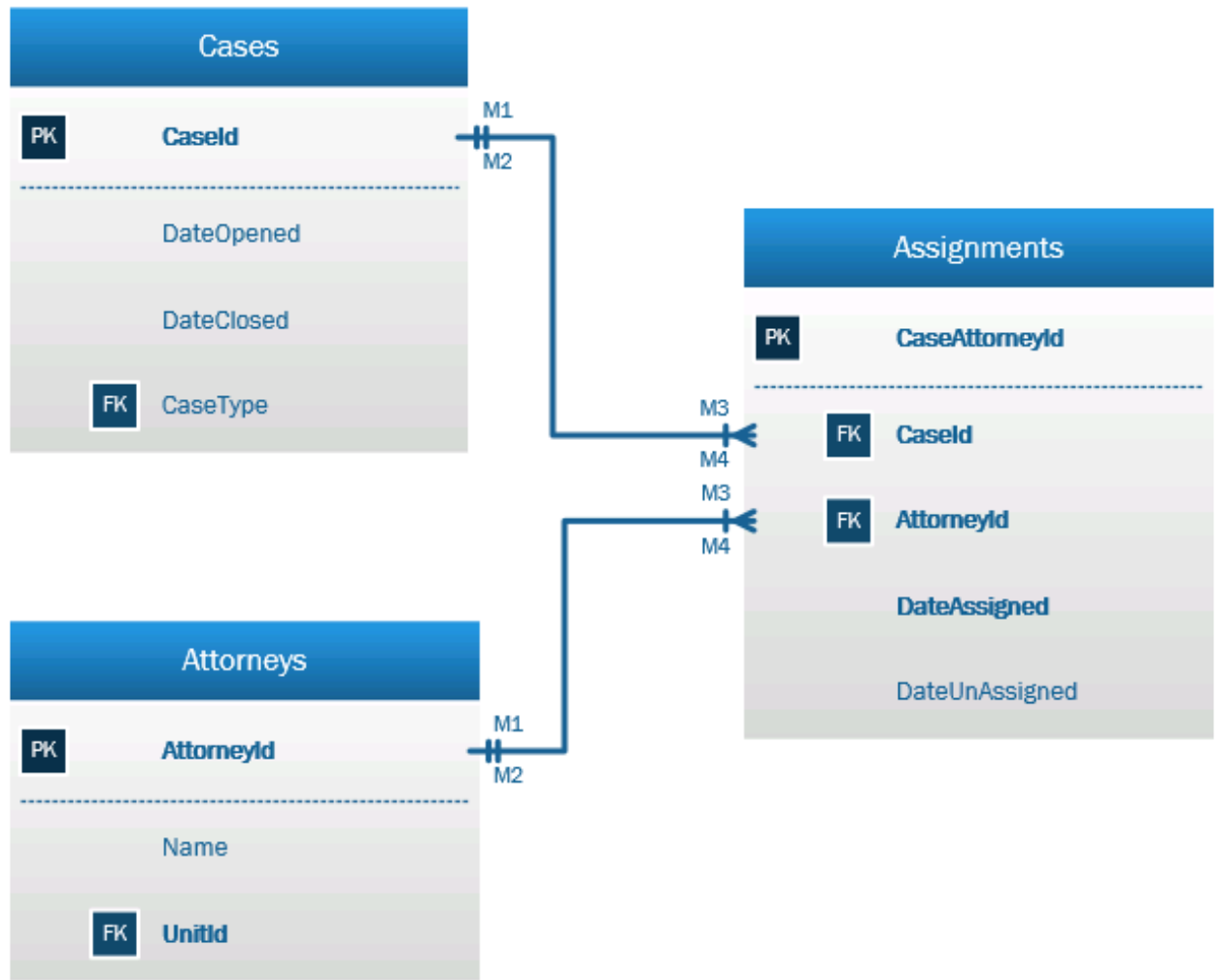
Server:
cpcs-zeno.database.windows.net

Connection:

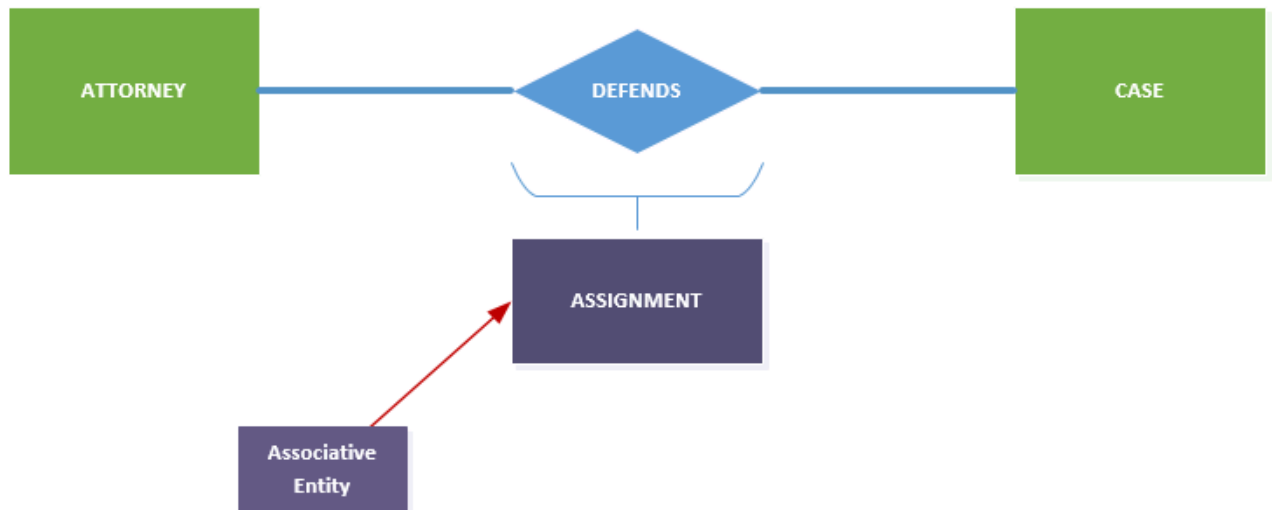
Script Help

Current connection parameters	
Database	Zeno_DEV
Server	cpcs-zeno.database.windows.net
User	jaghadiuno@publiccounsel.net
Description	
Schema	zeno
Name	Case
Created date	5/10/2018 3:21 PM
System object	False
GraphDB	
Table is a node table	False
Table is an edge table	False
Options	
Table temporal type	SystemVersioned
Durability	SchemaAndData
Memory optimized	False
ANSI NULLs	True
Quoted identifier	True

Associative Entity

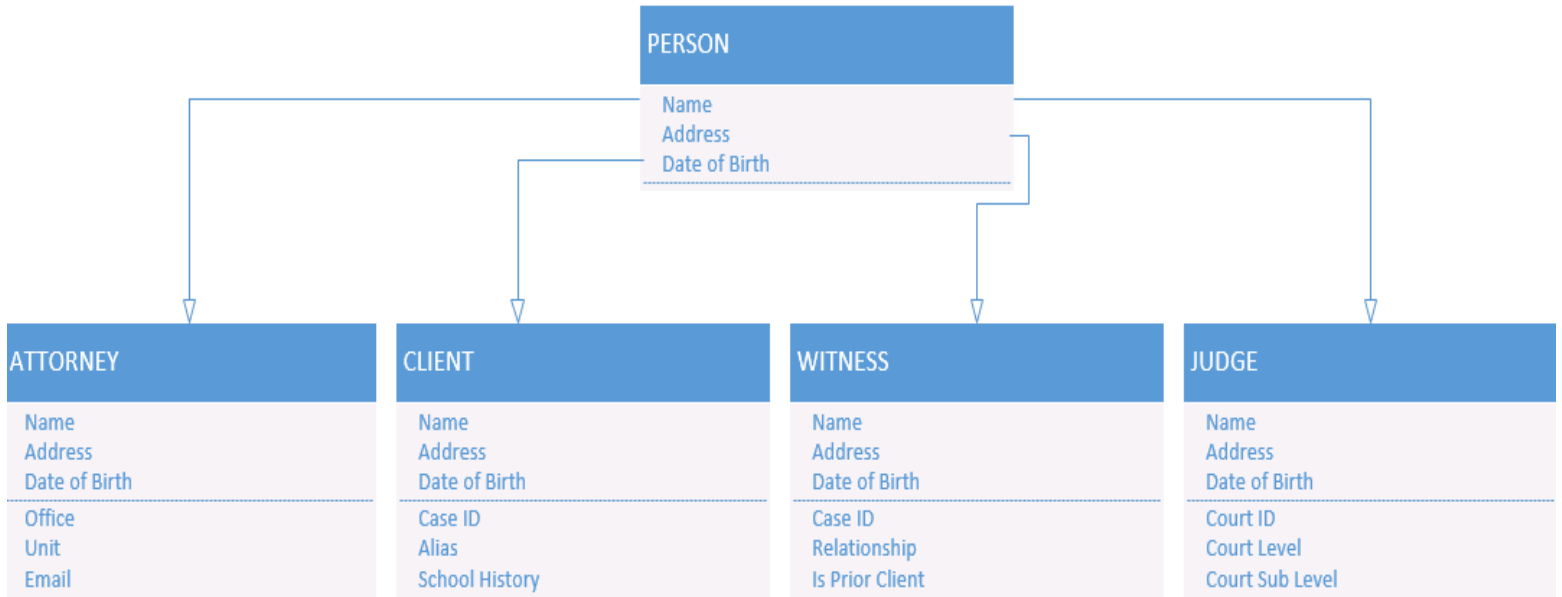


Notation Entity



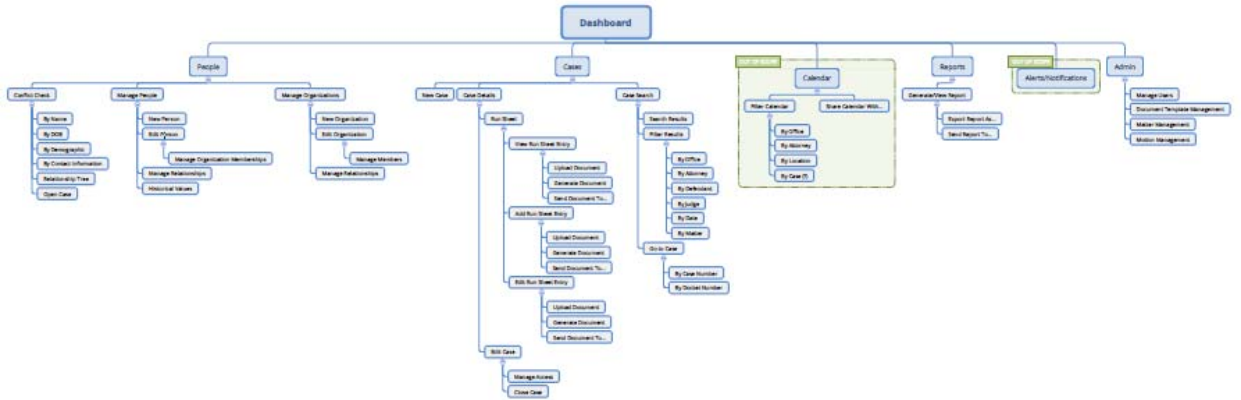
Object Models

Every person in the database will inherit from the PERSON class, including attorneys, clients, investigators, judges, defendants, witnesses, etc. While they inherit and share a common set of characteristics, as can be seen in the models below, they will possess, in varying degrees, their own characteristics.



User Interface Design

Sitemap



Homepage Template

People Search

First Name

Last Name

Date of Birth From To

City/Town

Gender Any

Race Any

[Advanced Search...](#)

Cases Case or Docket Number

Status	Case No.	Docket No.	Client	Opening Date	Last Updated	
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>
Open	123456	123456	Jonathan Doe	11/27/2017	11/27/2017	<input type="button" value="edit"/> <input type="button" value="delete"/>

My Cases

Reports

- Weekly Report
- Monthly Report
- Quarterly Report
- Annual Report
- Custom Report 1

People Search Template

The screenshot displays a web browser window with the address bar showing `https://zeno.publiccounsel.net/`. The page content is titled "Search Results" and includes a link for "Add New Person". The search form is organized into four main sections:

- Name:** Fields for First Name (containing "John"), Middle Name/Initial, Last Name (containing "Doe"), and Sounds Like.
- Personal:** Fields for Date of Birth (with a calendar icon), Gender (dropdown menu), Race (dropdown menu), and Last 4 of SSN.
- Contact:** Fields for Address, City/Town, State (dropdown menu with "MA" selected), ZIP Code, and Phone Number.
- Case:** Fields for Case Number, Docket Number, Opening Date (with a calendar icon), and Role (dropdown menu).

A "Search" button is located below the form fields. Below the search button, a message states "No results found." and a "Create a New Person" button is visible. On the left side of the page, there is a vertical sidebar with several icons: a briefcase, a person silhouette, a document, a bell, and a gear. At the bottom of this sidebar is a circular button labeled "IN".

Person Details Template

CPCS ZENO
https://zeno.publiccounsel.net/

Person Details
✖

Jonathan Michael Doe

Personal Information | Contact Information | Case History | Known Associates | Organizations

PERSONAL INFORMATION

Date of Birth: 05/01/1975	Place of Birth: New Bedford	Gender: Male	Marital Status: Married
Ethnicity: American	Race: Caucasian	Language: Spanish	Citizenship: US
SSN: XXX-XX-1234	Height: 6'0"	Weight: 205 lbs	PCF: 1234567

Also Known As

- John Doe
- Johnny Doe
- Jack Doe

CONTACT INFORMATION

Address History

Address	City/Town	State	ZIP	Last Updated
CURRENT 123 Thompson St	Springfield	MA	01020	06/2017
883 Main St, Apt 2	Holyoke	MA	01013	06/2015
294 Merrimack Ave, #8	Lowell	MA	01850	06/2007
45 Prospect St, 3	Providence	RI	02860	06/2005
99 Main St	Pawtucket	RI	02860	06/2002

Phone Numbers

PRIMARY 617-555-1234	Mobile
617-555-1234	Work

CASE HISTORY

Case Number	Docket Number	Role	Date Opened	Date Closed	Attorney	Judge
123456	234567	Defendant	11/01/2017	Still Open	Virginia Cunningham	Walter Manning
123456	234567	Defendant	03/09/2015	12/15/2015	Robert Smith	Rebecca Washington
123456	234567	Spouse/Domestic Partner	03/09/2015	12/15/2015	Robert Smith	Rebecca Washington
123456	234567	Victim	03/09/2015	12/15/2015	Robert Smith	Rebecca Washington
123456	234567	Defendant	03/09/2015	12/15/2015	Robert Smith	Rebecca Washington

KNOWN ASSOCIATES

First Name	Middle Name	Last Name	Relationship	Date of Birth	Age	Address	City/Town	Gender	Race
Jane	Barbara	Doe	Wife	05/01/1975	42	123 Thompson St	Springfield	Female	Caucasian
Jane	Barbara	Doe	Wife	05/01/1975	42	123 Thompson St	Springfield	Female	Caucasian
Jane	Barbara	Doe	Wife	05/01/1975	42	123 Thompson St	Springfield	Female	Caucasian
Jane	Barbara	Doe	Wife	05/01/1975	42	123 Thompson St	Springfield	Female	Caucasian
Jane	Barbara	Doe	Wife	05/01/1975	42	123 Thompson St	Springfield	Female	Caucasian
Jane	Barbara	Doe	Wife	05/01/1975	42	123 Thompson St	Springfield	Female	Caucasian
Jane	Barbara	Doe	Wife	05/01/1975	42	123 Thompson St	Springfield	Female	Caucasian
Jane	Barbara	Doe	Wife	05/01/1975	42	123 Thompson St	Springfield	Female	Caucasian
Jane	Barbara	Doe	Wife	05/01/1975	42	123 Thompson St	Springfield	Female	Caucasian
Jane	Barbara	Doe	Wife	05/01/1975	42	123 Thompson St	Springfield	Female	Caucasian

ORGANIZATIONS

Role	Organization Name	Address
Member	Central Organization of America	123 Main Street, Braintree, MA
Member	Kwanis International	123 Main Street, Brockton, MA

Open New Case
Edit Person
Cancel

Case Details Template

https://zamo-public.courtsnet.net/
CPCS ZENO

Person Details

Case #123456 – Jonathan Doe

Case Information | [Run Sheet](#) | [Charges \(5\)](#) | [Participants \(2\)](#) | [Bail](#)

CASE INFORMATION

Client: Jonathan Doe	Case Type: Criminal	Opening Date: 10/10/2017	Case Opening: In court
Lead Charge: DRUG DISTRIBUTE OR POSSESS WITH INTENT, CLASS A, SUBSQ OFF. c. 94C s. 32(b)			
Attorney Assigned: Alex Montgomery	Date Assigned: 12/08/2017		
Assisting Attorney: None	Prosecutor: James Colston		
Social Worker: None	Investigator: None		
Court: Barstable District Court	Department: District Court		
Judge: Aubrey Washington	Is Prior Client: No		

Notes: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

RUN SHEET Showing 5 of 27 entries

Date	Staff Member	Category	Details	Outcome
12/29/2017	Alex Montgomery	Non-court event	Client call	N/A
12/29/2017	Alex Montgomery	Non-court event	Client call	N/A
12/29/2017	Alex Montgomery	Non-court event	Client call	N/A
12/29/2017	Alex Montgomery	Non-court event	Client call	N/A
12/29/2017	Alex Montgomery	Non-court event	Client call	N/A

CHARGES

Docket No	Offense	Date	Court Level	Disposition
★ 1234 CR 001234	DRUG DISTRIBUTE OR POSSESS WITH INTENT, CLASS A, SUBSQ OFF. c. 94C s. 32(b)	12/8/2017	District Court	
1234 CR 001234	DRUG DISTRIBUTE OR POSSESS WITH INTENT, CLASS B, SUBSQ OFF. c. 94C s. 32A(b)	12/8/2017	District Court	
1234 CR 001234	CONSPIRACY TO VIOLATE DRUG LAW c. 94C s. 40	12/8/2017	District Court	
1234 CR 001234	DRUG DISTRIBUTE OR POSSESS WITH INTENT CLASS D c. 94C s. 32C(a)	12/8/2017	District Court	
1234 CR 001234	DRUG DISTRIBUTE OR POSSESS WITH INTENT CLASS E c. 94C s. 32D(a)	12/8/2017	District Court	

PARTICIPANTS

Name	Role	City	Case Weight
Holly Archibald	Co-defendant	Boston	0
James Winchester	Co-defendant	Brockton	0

BAIL

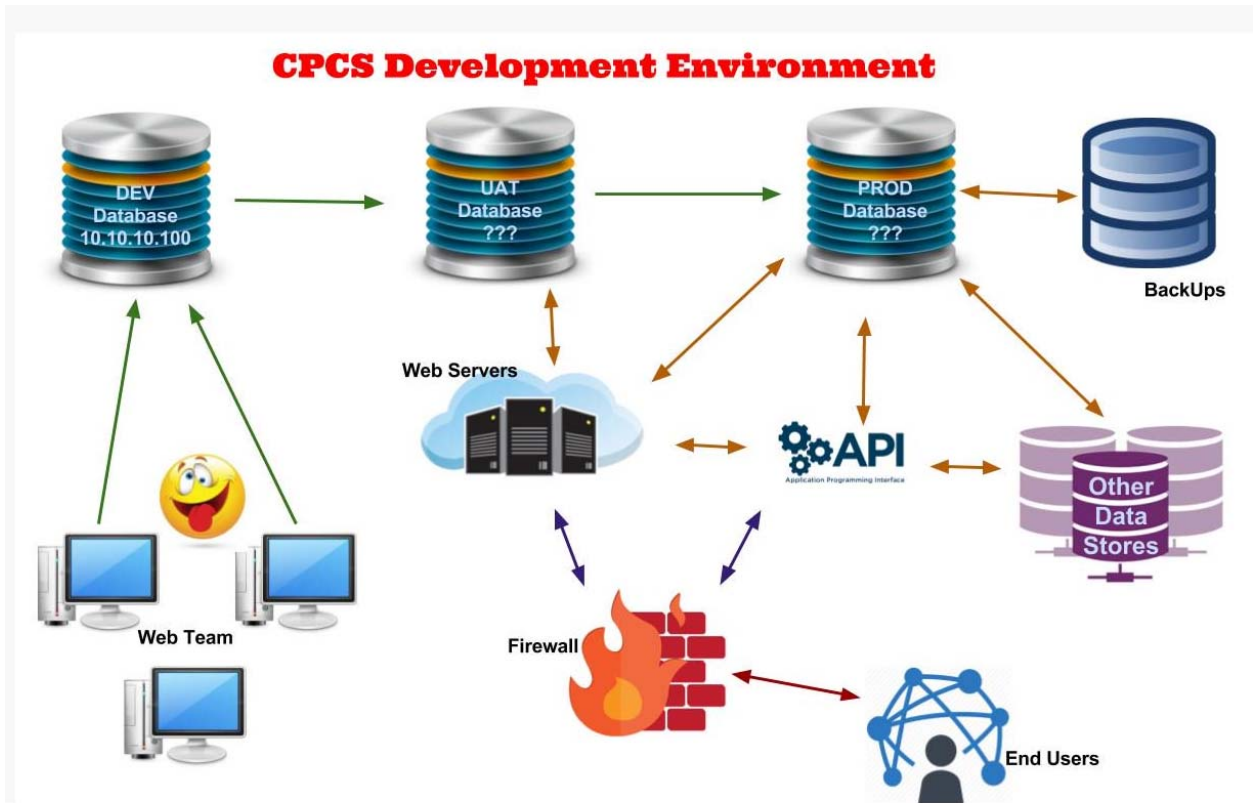
Held in Custody: Yes	Held at Location: Barstable County Jail and House of Correction
Bail: No	Amount: N/A
Bail Conditions: N/A	Released: No

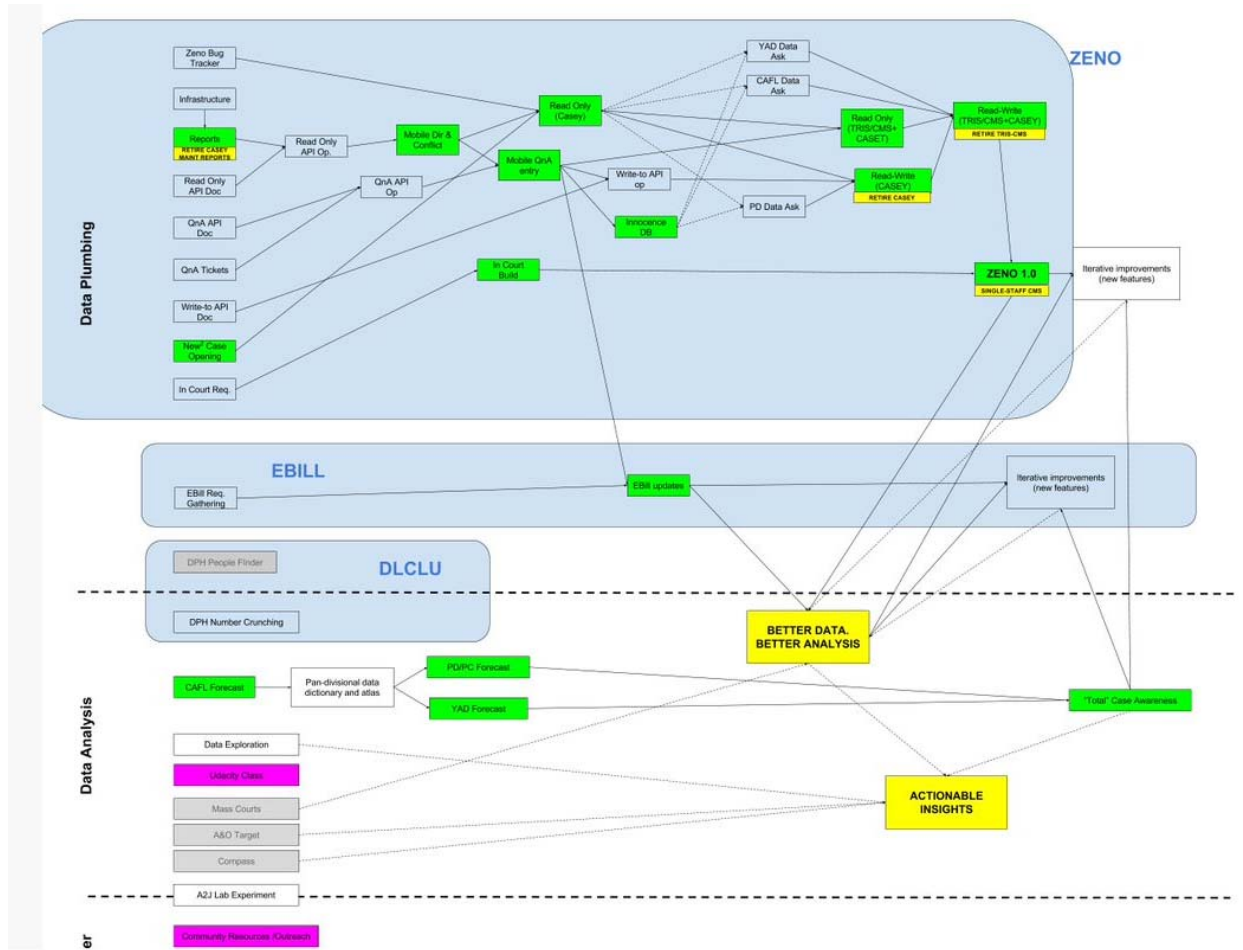
Arrignment Dates

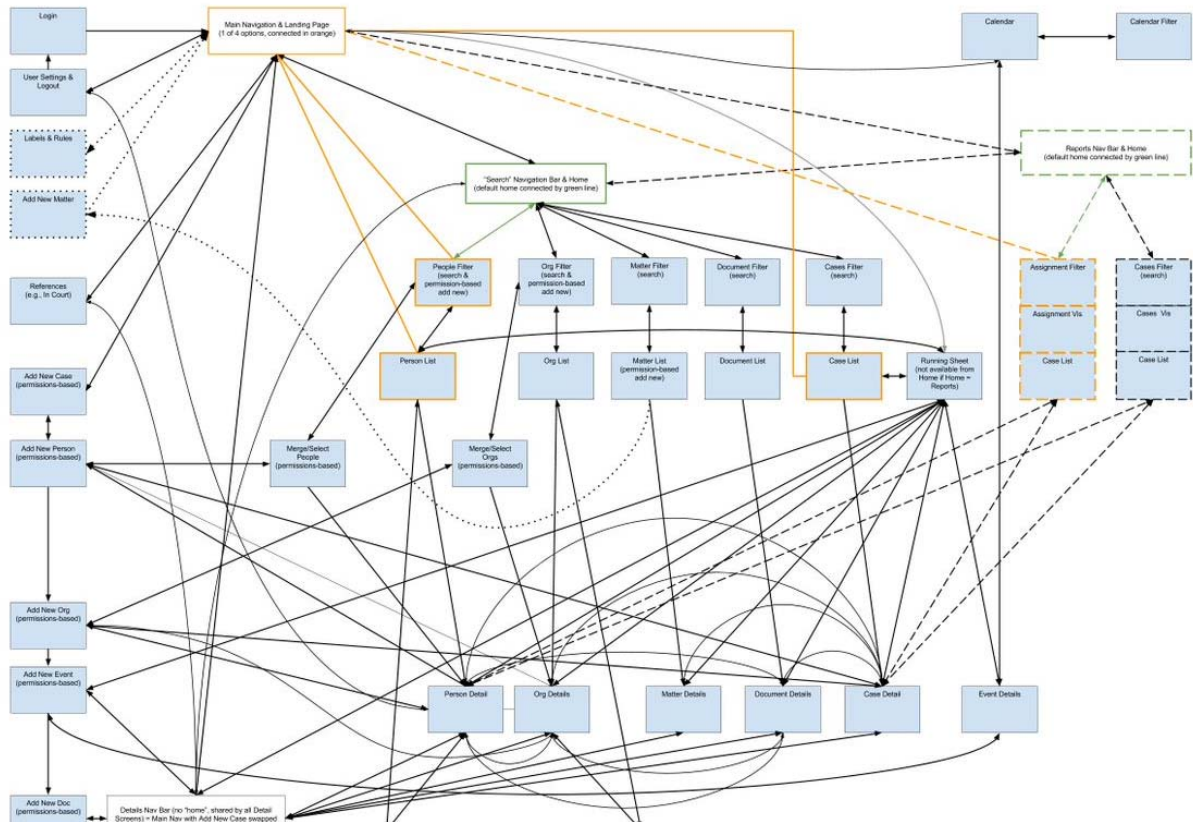
District Arrignment Date: N/A	Superior Arrignment Date: N/A
Juvenile Complaint Arrignment Date: N/A	Juvenile Incident Arrignment Date: N/A

[Edit Case](#) |
 [Add Run Sheet Entry](#) |
 [Close Case](#) |
 [Re-open Case](#) |
 [Cancel](#)

System Architecture







Feasibility Analysis

For the past four years CPCS has collected data on user needs regarding case management. This has involved both formal and informal conversations, including insights gained from direct user feedback, multiple listening tours, surveys, and formal meetings for the Gideon Project. Since May, we have conducted thirty-six new interviews, and five group lunch talks, as part of site visits. Interviews have included sessions with Deputy Chief Counsels, Managing Directors, AICs, AAs, SSAs, Social Workers, Investigators, Directors of SSAs/social workers, PD's lead investigator, and a paralegal. These recent meetings included members from PD, CAFL, YAD, MH, and A&O in offices across the commonwealth, including Boston, Lawrence, Norwood, Quincy, Roxbury, and Worcester. All told, it is estimated that the entire

multi-year process has involved in-person meetings with roughly a hundred staff members in offices from Boston to Pittsfield. This is in addition to a little over two hundred anonymous survey replies. Consequently, we have spoken with members of every practice area, including representatives of every user type. All of these have served to inform the current set of functional requirements.

Findings

We have identified eight high-level themes that apply to the system as a whole, and four areas of focus that help to define specific groupings of features. The focus areas are themselves comprised of specific requirements and a summary is presented below.

Themes

- Centralization (Facilitates Collaboration and Avoids Duplicate Data Entry)
- Mobile Access
- Granular Data Collection & Flexible Data Elements
- Data Aggregation & Analysis (e.g., Standard Reporting and Predictive Modeling)
- User Interface & Workflow Support
- Permissions, Security, & Privacy
- Reliability of Data and System (Including Technical Performance and Data Quality)
- Smooth Transition to a New System (Data and Training)

Focus Areas & Features

People & Groups				
Requirement(s)	In Current?	Complexity	Impact	Priority
The ability to find, edit, and add information on people and groups, including the relationships of people to each other, to cases, and to groups.	in part	med	high	high
Radically improved search: fuzzy/statistical search/match (esp. conflict check)	no	high	high	high

Public person and group profiles w/ user comments	no	low	med	low
---------------------------------------------------	----	-----	-----	-----

Case Management, Document Management, and Calendaring				
Requirement(s)	In Current?	Complexity	Impact	Priority
The ability to find, edit, and add information on cases. Such information includes: <ul style="list-style-type: none"> Relationships of people and groups to cases Runsheets, including fine-grain action tracking (e.g., did you argue a motion today?) Calendars Documents (external and internal) Matter dispositions and procedural history 	in part	high	high	high
The system should be capable of producing documents from user-defined templates.	in part	high	high	med
Improved collaboration: <ul style="list-style-type: none"> In-system referrals (investigators et al.) Separate communications channel for supervision Internal notifications of involved parties (e.g., letting SSA know when a runsheet entry was made) 	in part	med	high	high

Reports & Research				
Requirement(s)	In Current?	Complexity	Impact	Priority
Standard reporting nomenclature (e.g., agreement on what one counts when counting cases)	no	med	high	high
Static reports (e.g., trends & counts by office, employee et al.)	in part	med	high	high

Dynamic Reports (e.g., Boolean queries of most data fields)	in part	high	med	high
The ability to export or make available via API information for consumption by external systems (e.g., export to csv, Excel)	in part	low	high	high
Access to CPCS research material (e.g., training and synced charge info) and pointers to external resources (e.g., linking to statutory language or model jury instructions).	no	med	med	low

Administration				
Requirement(s)	In Current?	Complexity	Impact	Priority
Permissions based on relationships to cases and groups' membership (roles)	no	med	high	high
Workflow management & business rules <ul style="list-style-type: none"> Contextual delivery of best practices (e.g., serving up appropriate documents and presenting links to additional resources) Data entry support 	no	high	high	low
Flexible nomenclature	no	med	high	high
Adaptive presentation <ul style="list-style-type: none"> Custom Home Screen / Dashboard Data filters for specific roles and divisions 	no	med	med	high

Project Plan

Work Breakdown Structure

	Task Name	Work	Duration	Start	Finish	Type
1	Project - ZENO	768 hrs	63 days	Tue 8/7/18	Thu 11/1/18	Fixed Duration
2	Initiation	48 hrs	10 days	Tue 8/7/18	Mon 8/20/18	Fixed Duration
3	Review Project Notes	16 hrs	4 days	Tue 8/7/18	Fri 8/10/18	Fixed Units
4	Create Business Case Document	24 hrs	6 days	Mon 8/13/18	Mon 8/20/18	Fixed Units
5	Create Schedule	8 hrs	2 days	Mon 8/13/18	Tue 8/14/18	Fixed Units
6	Obtain User Signoff to Proceed	0 hrs	0 days	Mon 8/20/18	Mon 8/20/18	Fixed Units
7	Requirements	320 hrs	20 days	Tue 8/21/18	Mon 9/17/18	Fixed Duration
8	Fact Find	160 hrs	20 days	Tue 8/21/18	Mon 9/17/18	Fixed Units
9	Review Existing System	160 hrs	20 days	Tue 8/21/18	Mon 9/17/18	Fixed Units
10	Design	120 hrs	15 days	Tue 9/18/18	Mon 10/8/18	Fixed Duration
11	Design User Interface	40 hrs	5 days	Tue 9/18/18	Mon 9/24/18	Fixed Units
12	Design Data Flow	40 hrs	5 days	Tue 9/25/18	Mon 10/1/18	Fixed Units
13	Design User Processes	40 hrs	5 days	Tue 10/2/18	Mon 10/8/18	Fixed Units
14	Prototype	160 hrs	10 days	Tue 10/9/18	Mon 10/22/18	Fixed Duration
15	Create Web Page Prototype	80 hrs	10 days	Tue 10/9/18	Mon 10/22/18	Fixed Units
16	Create iOS Prototype	80 hrs	10 days	Tue 10/9/18	Mon 10/22/18	Fixed Units
17	Verification	80 hrs	5 days	Tue 10/23/18	Mon 10/29/18	Fixed Duration
18	Review Solution Architecture	40 hrs	5 days	Tue 10/23/18	Mon 10/29/18	Fixed Units
19	Validate Prototype	40 hrs	5 days	Tue 10/23/18	Mon 10/29/18	Fixed Units
20	Finalize	40 hrs	3 days	Tue 10/30/18	Thu 11/1/18	Fixed Duration
21	Determine Recommendation	24 hrs	3 days	Tue 10/30/18	Thu 11/1/18	Fixed Units
22	Create Executive Presentation	16 hrs	2 days	Tue 10/30/18	Wed 10/31/18	Fixed Units

Project Monitoring and Control Plan

Definition of Done

The Definition of Done is a set of agreed-upon definitions for when a given software feature is regarded as truly completed (feature complete), and ready to be delivered and closed out as a work item. The definition, as agreed upon by all key stakeholders, consists of far more than just the code, as regarded by the developer, being completed; rather, each delivered feature is subjected to the same standards for quality and delivery to ensure a greater level of efficiency and verifiability when developing a given feature.

To ensure quality of delivery of features, CPCS defines the following checklist items as its Definition of Done:

1. Code Completed

The code is considered complete by the developer. All to-do items present in the code are resolved, and the developer considers it completed and that it fully addresses the scope of the User Story in question.

2. Code Commented, Checked In, and run against current Source Control Version

The developer has sufficiently documented the code, based on the agreed-upon code documentation standards, and checked it into source control. Normally, this means the code check-in has been associated with the relevant work item in the relevant project management utility. Additionally, the developer has gotten the latest version of the code branch in question and ensured that their code still runs against that branch.

3. Builds without Errors or Warnings

The developer has produced code that compiles without errors. Additionally, it is a standard best practice that production-ready code should also build without any compiler

warnings whatsoever. Typically, consistent checks against this are performed on a nightly or on-demand basis by a Continuous Integration process, that automatically compiles the application as of its current version.

4. Unit Tests Written and Passing

The developer has produced accompanying unit tests, encompassing an acceptable level of code coverage (60-75% code coverage would be recommended in general, with the level reflective of how critical and complex the component is) for the given component. These unit tests must all pass.

5. Code Reviewed, and meets Development Standards

The developer has produced code that meets the agreed-upon code quality standards, as determined by a peer review. The peer review will ensure that the code not only meets general syntax and stylistic guidelines, but also that the code adheres to general principles of abstraction and DRY (Don't Repeat Yourself).

6. Any Build/Deployment/Configuration Changes are

Implemented/Documented/Communicated

Any configuration changes necessary (application-level configuration, or permissions changes, or credentials) are documented and communicated to the team.

7. Hours for the Task or User Story are set to Zero, and the work item is marked as “completed”

Once the item has passed the previous enumerated items, the work item can be marked as completed.

8. The Code is deployed to a Test Environment, and System Tests are completed

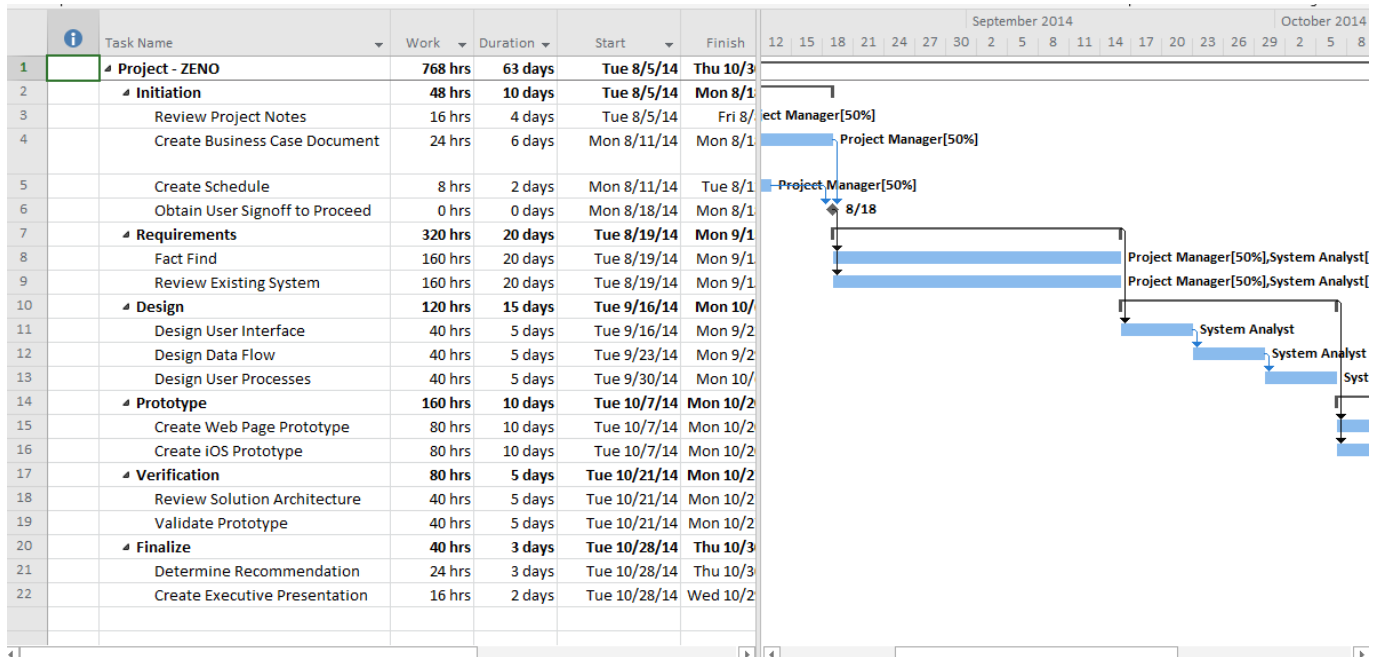
After the item is marked as completed, it must be deployed to a dedicated test environment for further vetting and ensured that the system still functions as expected.

9. The Code is verified in the Test environment and signed off on

A person other than the developer who implemented the code (typically a dedicated QA resource) tests and verifies the code, and ensures that it matches expectations, as set out by the requirements for the work item. If available, UAT signoff may be incorporated into the development process, but is not necessary to consider an item to be “done”.

10. The work item is marked as closed, and considered “done”

Timeline

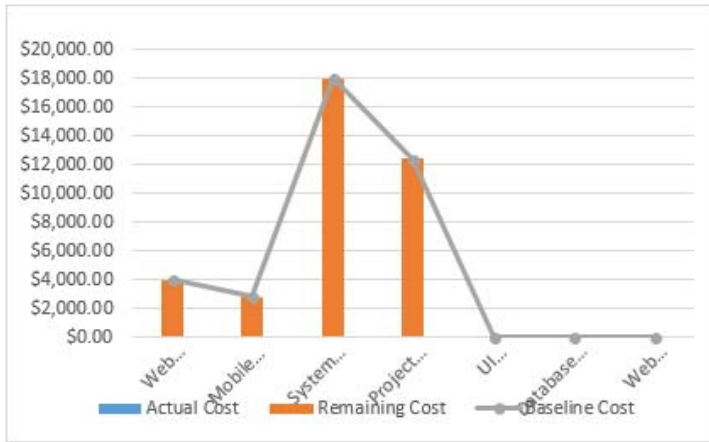


Cost Sheet

RESOURCE COST OVERVIEW

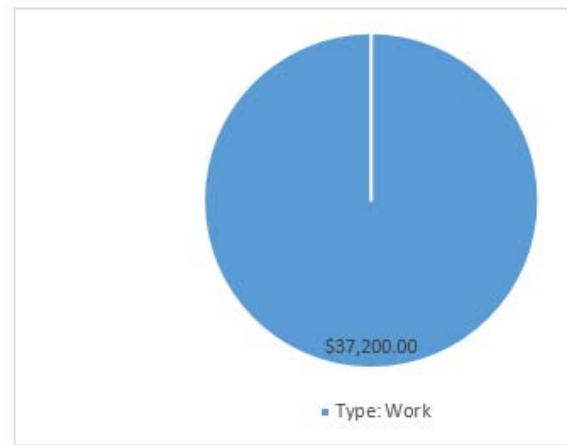
COST STATUS

Cost status for work resources.



COST DISTRIBUTION

How costs are spread out amongst different resource types.



COST DETAILS

Cost details for all work resources.

Name	Actual Work	Actual Cost	Standard Rate
Web Developer	0 hrs	\$0.00	\$50.00/hr
Mobile Developer	0 hrs	\$0.00	\$35.00/hr

Name: Duration: Effort driven Manually Scheduled Previous Next

Dates: Start: Finish:

Constraint: Date: Task type: WBS code:

Current Baseline Actual Priority: % Complete:

References

Rosenblatt, H., & Scott, T. (2017). *Systems Analysis and Design* (11th ed.). Retrieved from <https://mbsdirect.vitalsource.com/#/books/9781337424202/>