Teen Court Database

Software Design Document | Current Version 1.0.4

Prepared By:

Andrew Thompson

Robert Reilly

Revision History

Date	*Author*	*Version*	*Comments*
9/26/12	*Andrew Thompson*	*1.0.0*	*Initial version*
10/30/12	*Andrew Thompson*	*1.0.1*	*Updated User Access and Defendant*
11/20/12	*Andrew Thompson*	*1.0.2*	*Added Court Program and Volunteer*
12/2/12	*Andrew Thompson*	*1.0.3*	*Added SQL schema information*
12/5/12	*Andrew Thompson*	*1.0.4*	*Added database table for each section*

Table of Contents

1.0 Overview

The Teen Court Database is a web application used to track the progress of juvenile defendants within the local teen, or youth court program. Court administrators will have the ability to generate reports, view statistics and track volunteer information. Since this is intended to be web application, access will be obtained with any internet capable computer or mobile device.

This document contains the necessary components for the design and implementation of the Teen Court Database application.

1.1 Scope

The scope of this project is to develop a web based system that allows court administrators the ability to track information for juvenile defendants within the teen court programs.

1.2 Purpose

The purpose of this document is to describe what the various components of the system are and how they will be developed. The major components of the application are described in various subsections:

- *Technologies Used:* Lists the specific languages, plugins, or tools used to develop the component.
- *Component Overview:* Describes the data and database tables within the component.
- *Phase Overview:* Describes what work is done during specific phases of development.
- *Architecture Overview:* Describes how the component will function.
- *Design Overview:* Describes the look and feel of the component.

Any possible development issues or constraints will also be described.

1.2.1 Domain Name and Web Space

This application will be developed under the domain name teencourtdb.com. Web and database server hosting will be provided by the Baltimore based website design firm, eCoastStudios. Michael Roth is the main contact for this project.

1.2.2 Server Environment and Database

The development server and production server will be on Linux servers with the latest version of PHP and MySQL database. A hosting control solution, cPanel, will allow the development team access to web space management.

1.2.3 Web Application

The main application will provide all the requirements gathered from stakeholders and listed in the Software Requirements Document.

1.2.4 Security

In addition to individual user access based upon email addresses, it is recommended to obtain a security certificate/key to provide a Secure Socket Layer (SSL) for data encryption between the client and server. Also, various sensitive data will be encrypted in the database.

1.3 Systems Goals

This project's goals are to give teen/youth court program administrators and their employees an accessible and easy to use solution to track participants in youth court programs. By tracking defendant data, statistics can be generated to prove these types of alternate judicial processes work. This may lead to more funding and more courts taking this approach.

1.4 System Overview and Diagram

Users will access this application through qualified web browsers offered on most major internet capable devices, such as personal computers, tablets and smart phones. The application will offer cross-platform and cross-browser support. When a user's browser makes a connection to the website, the pages and data will be served to the client. The diagram in figure 1 shows the system's major components.

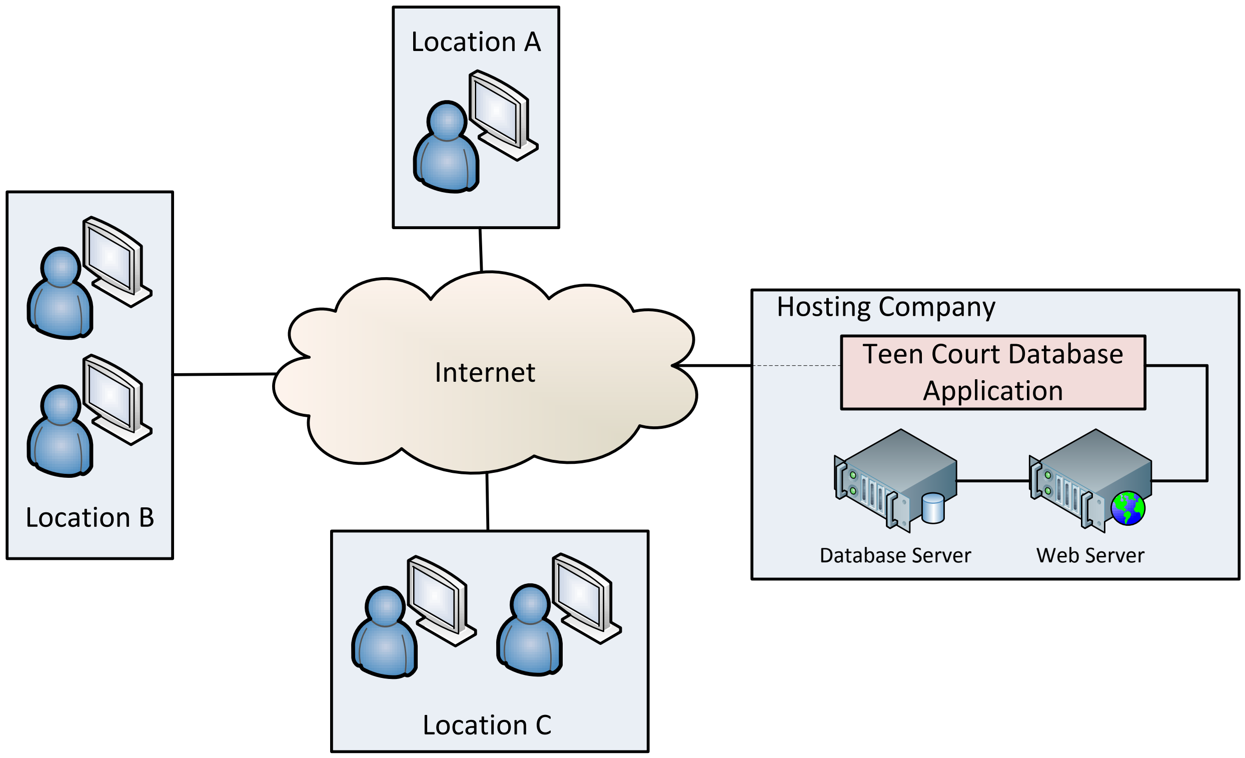

Figure 1: System Diagram

1.5 Technologies Overview

This application will be developed using the latest version of PHP and take an Object-Oriented approach. PHP is popular and widely used HTML-embedded scripting language that allows developers to quickly build dynamically generated web pages. More information: can be found at http://us3.php.net/.

A data access layer will be implemented to help secure information between the web and database server. This will most likely be PHP's built in PHP Data Object (PDO). For more information, visit http://www.php.net/manual/en/book.pdo.php.

For data storage, the latest version of MySQL will be used with this application. This is a popular open-source relational database management system (RDBMS). More information can be found at http://www.mysql.com/.

JSON, JavaScript Object Notation, will be used for formatting the data of SQL results. http://www.json.org/

A JavaScript Framework, JQuery, will be used to ensure cross-browser JavaScript support. To manipulate client webpages without making additional requests to the server, AJAX, or Auto-synchronous JavaScript and XML, will be used and is built into the framework. More information can be found at http://www.jquery.com.

JQueryUI , a JavaScript Framework that generates client-side widgets will be used to provide a more functional user interface. More information about this framework is available at http://www.jqueryui.com.

2.0 Project Overview

This section will describe team member roles and how the project will be managed.

2.1 Team Members and Roles

Andrew Thompson will be the Team Leader and Lead Developer. Robert Reilly will be Lead Tester and Developer.

2.2 Project Management Approach

The Scrum Team is scheduled to have Sprint Meetings every Tuesday and Thursday at 10:00 AM. These meetings will not last longer than 15 minutes. Additional Meetings will be scheduled as needed. The Project Lead, Andrew Thompson, is the primary contact with the Product Owner, Marlene Todd.

The project backlog will be tracked within Trello – an online collaboration and organization tool. Scrum Masters and Scrum Team Members are required access to this site. The Product Owner will not need to access this site. The Scrum Team will manage Sprint Backlog priorities based upon the Product Owners input. Sprints will last between two and three weeks. Sprint Reports will be sent to the Scrum Master and Product Owner at the end of scheduled Sprints.

The online repository, GitHub, will provide source control, source-code browser, and a project wiki. Scrum Team Members and the Scrum Master are required to have access to this site.

2.3 Phase Overview

This system will be developed in phases, they are as follow:

2.3.1 User Stories and Requirements Gathering

During this phase, the requirements for the system will be gathered from user stories. The product owner informs the team of any requirements, limitations or constraints. This information can be found in the Software Requirements Document and 3.0 Requirements.

2.3.2 Database Schema and Class Design

The database schema, or design, will be developed during the second phase. The team needs to mimic the existing Access Database currently being used by Lawrence County Teen Court, while making sure the design is as efficient as possible. The major class objects will also be designed during this phase; this includes defendants, volunteers, court listings, workshops and any other objects that can be derived from the requirements.

2.3.3 Prototype

The prototype will provide limited functionality of the final application and act as a demonstration for the client. This will allow the team to make changes without major code rework. All major areas of the application should be built for the client review.

2.3.4 Application Development

This phase will be the actual development phase. All aspects of the application will be implemented to the client's specifications and requirements. Some unit testing will take place during this phase.

2.3.5 Testing

Some aspects of testing will take place during the Application Development phase. This is to ensure proper data entry for the application. Individual domain constraints will be tested along with data accuracy and program functionality.

2.3.6 Delivery

This phase is where the application will be turned over to the client. The web files and database will already be on the production server. User manuals for each user level will also be delivered in PDF format and hard copy. The user accounts used to access the website and database will also be turned over.

2.4 Terminology and Acronyms

- AJAX: Asynchronous JavaScript and XML is a set of development tools to provide the client side to communicate with the server without refreshing the page.
- CAPTCHA: A CAPTCHA is a program that protects websites against bots by generating and grading tests that humans can pass but current computer programs cannot.
- Client: An application or system that accesses a service provided by a server.
- Court Program: A local city, county or community youth or teen court that uses this web application.
- Database: An organized set of data and associated data structures.
- DBMS: A Database Management System allows an interface to manipulate a database.
- Domain Name: An identification string that identifies represents an Internet Protocol (IP) resource. A Domain Name Server (DNS) handles the translation from Domain Name to IP address.
- JSON: JavaScript Object Notation is a lightweight data-interchange format.

- JQuery: JavaScript Framework that allows client manipulation.
- JQueryUI: JavaScript Framework that allows various client widgets to provide a more functional user interface.
- MySQL: A popular and reliable Database Management System (DBMS).
- NAYC: National Association of Youth Courts provides resources for over 1,050 youth and teen court programs throughout the country.
- PHP: PHP Hypertext Preprocessor is an HTML-embedded scripting language capable of Object Orientated Programming (OOP) principles.
- Server: A computer hardware system and software that serves as a dedicated host for one or more services.
- SSL: Secure Socket Layer is a cryptographic protocol that provides communication security over the Internet.
- Web Application: A software application that is developed in a browser-supported programming language and is accessed by users over an Intranet or Internet.
- Web Hosting: A hosting service on the Internet that allows website owners to access their site and typically provides web space management tools or control panels.

3.0 Requirements

The purpose of this web application is to allow youth and teen courts to track participants within their programs. In order to achieve this, several requirements are specified from the stakeholders.

3.1.1 User Access

Only individuals working for a particular court program may have access to the application. Defendants, their families, and volunteers will **not** be allowed to view **any** information contained in the application. The different levels are described in section 4.1 User Account and Access.

3.1.2 Court Program & Data Access

The authorized users shall only have access to information pertaining to their particular court program. Court program administrators are able to adjust options pertaining to their individual programs. These options are covered in further detail in section 4.2 Court Program. Application Administrators will have the additional access to site-wide statistics and options to add new court programs. These levels of access are described in section 4.1 User Account and Access.

3.1.3 Defendant

The defendant is the primary data gathering point for this application. A defendant can be added, updated, deleted, and expunged. Each court program and their users can only access defendant information they enter. The different information fields are described in section 4.3 Defendant.

3.1.3.1 Defendant Expunging

Each court program will have one of three options to expunge defendant information: sealed, partial expunge and full expunge. The sealed option retains defendant data within the database but restricts the data to be displayed in results other than demographic statistics. Partial expunging removes the defendant's personal information from the database but leaves citation information for statistics gathering. Full expunge removes all traces of the defendant within the system.

3.1.4 Volunteer

Volunteers are those individuals who assist with the teen court program. Generally they are assigned as court members or jury participants. The ability to track hours, status and volunteer history will be implemented. More information can be found in 4.4 Volunteer.

3.1.5 Workshop

Various workshops can be created and populated from open defendants. More information can be found in 4.6 Workshops.

3.1.6 Reports and Statistics

All necessary and legal documents will be generated for the defendants. Statistics reporting will include discoverable data, demographics data, and both local and regional statistics.

4.0 Design and Implementation

The web application will be developed as several different areas or components. Components may interact with others depending upon their type or usage. Each component is described in detail below.

4.1 User Account and Access

This section describes the User Account and User Access development process.

4.1.1 Technologies Used

PHP, MySQL, AJAX, JQuery and JQuery UI. Registration page uses an additional Captcha plugin.

4.1.2 Component Overview

Every user accessing the system will be required to have the following information in the database:

- ID of the Court Program they belong to
- Name (First and Last)
- Email Address (encrypted in the database)
- Password (encrypted in the database)
- Phone numbers
- Account Type
    - Application Administrator
    - Program Administrator
    - Program Manager
    - Program User

4.1.2.1 Database Tables Used

The following MySQL tables are used: user, user_log, user_phone, program

4.1.3 Phase Overview

This phase will be the design and implementation of the different levels of users able to access the application. The tables for the database will be designed and the various web pages and scripting will be implemented.

4.1.4 Architecture Overview

This phase will implement the *index.php* and *main.php* pages that will handle the user access and *registration.php* that will be used for new user account registration.

The primary concern with user access is ensuring the user passwords cannot be accessed through unauthorized means. Prevention against hacking and brute force access scripts must be high priority. There will be a maximum of 3 login attempts before the user is locked out of the system. They must then wait a certain amount of time before trying again. This prevents scripted hacking attempts.

The email and password will be protected by using the SHA-256 cryptographic hash function with a random salt, or additional string value. In addition to the encrypted password, there will also be a limited number of login attempts to prevent brute force access using automated scripts.

New users will register their account from the registration page. The users will be given a court access code by their individual administrators and will need this to register with the application. The access code will be matched with the database to ensure it is a valid code. As the user enters a password, it will be checked against a strength algorithm, using AJAX, to make sure they are using a somewhat secure.

To avoid bots from spamming the registration page, a CAPTCHA system will be implemented to prove a human is registering for the application.

4.1.4.1 User Registration

Figure 2 shows the architecture diagram of the registration process.

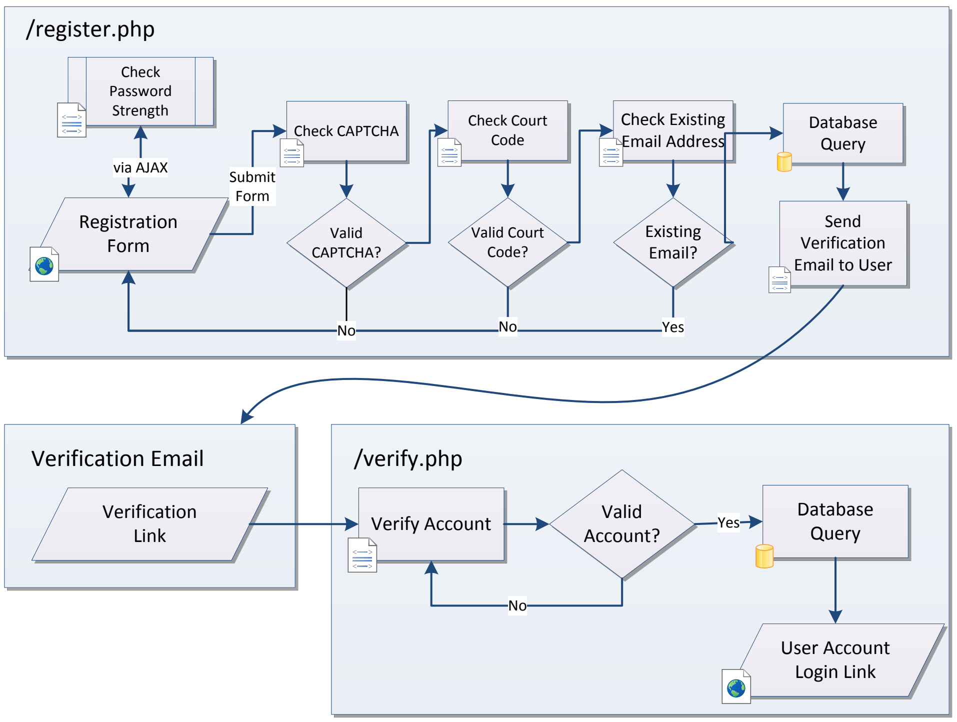

Figure 2: User Registration Architecture Diagram

4.1.4.2 User Access

Figure 3 is the architecture diagram of the user access implementation.

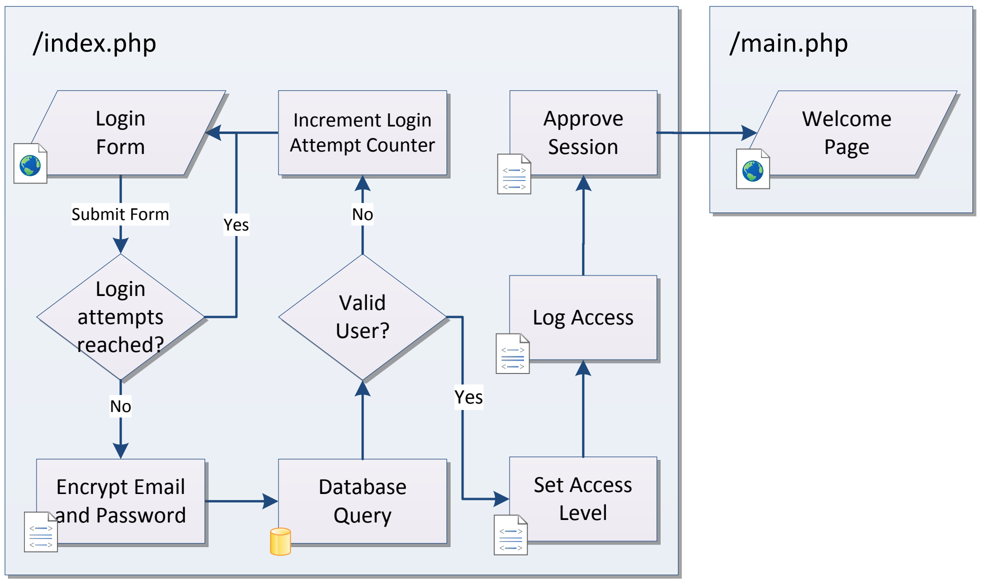

Figure 3: User Access Architecture Diagram

4.1.5 Design Details

These pages use CSS and JQuery UI elements, such as buttons and a custom theme, to provide unique styling.

4.2 Court Program

4.2.1 Technologies Used

4.2.2 Component Overview

4.2.2.1 Database Table Schemas

4.2.3 Phase Overview

4.2.4 Architecture Overview

4.2.5 Design Details

4.3 Defendant

This section describes the Defendant development process. The Defendant is the main data object for this application. Several fields will be propagated via dropdown boxes that the court administrator has access to.

4.3.1 Technologies Used

PHP, MySQL, AJAX, JSON, JQuery.

4.3.2 Component Overview

The following information is required for all Defendants:

- Primary Defendant Information
  - Name
    - First Name
    - Middle Initial
    - Last Name
  - Citation Number (from citation information)
  - Citation Date (from citation information)

    - Citation Time (from citation information)
    - Home phone number
    - Date of Birth
    - Court Case number
    - Agency Case number
    - Expunged date
    - Closed date
- Personal Information
    - Address
        - Physical Address
        - Mailing Address
    - School Data
        - School Name
        - Grade level
    - Physical Description
        - Height
        - Weight
        - Eye Color
        - Hair Color
        - Race
    - Driver's License Data
        - Number
        - Issue State
- Parental Information
    - Parental Relationship
    - Name
        - First Name
        - Middle Initial
        - Last Name
        - Suffix
    - Phone Number
        - Home Number
        - Work Number
    - Address
        - Physical Address
        - Mailing Address
    - Field to specify if living with that parent or not
- Citation Information
    - Offense Date
    - Offense Time
    - Day of Week
    - (Race/Age/Sex is auto filled from Personal Data)
    - Location
        - Address Number
        - City
        - State
        - Zip Code
        - Common Place

        - Cross Street
    - Citing Officer Data
        - Name
        - Identification Number
    - Mirandized checkbox
    - Offense Data
        - Statue
        - Title
        - Type
    - Vehicle Data
        - License Number
        - License State
        - Make
        - Type
    - Drugs or Alcohol Involved
    - Stolen/Vandalized Data
        - Description of items stolen or vandalized
        - Value or Amount
- Intake Information
    - Intake Date
    - Intake Time
    - Reschedule Date
    - Reschedule Time
    - Intake Interviewer
    - Referred to Juvenile – Not Qualified (Date)
    - Dismissed / No Complaint (Date)
- Court Information
    - Court Information
        - Court Date
        - Court Time
        - Court Type
        - Court Location
    - Judge
    - Court Officers
        - Defense Attorney
        - Prosecuting Attorney
        - Bailiff
        - Court Clerk
        - Exit Interviewer
    - Advisor (Multiple)
    - Jury (Multiple)
    - Parent Present (Multiple)

4.3.2.1 Database Table Schemas

Defendant, school, intake,

4.3.3 Phase Overview

The initial layout will be developed during the design of the prototype. Actual implantation takes place during the Application Development phase.

4.3.4 Architecture Overview

4.3.5 Design Details

This section will be compromised of two main areas, the top defendant information and a multiple-tab area containing additional information described above. Once the primary information is added, the user will have access to the additional tabbed areas.

4.4 Volunteer

4.4.1 Technologies Used

4.4.2 Component Overview

4.4.2.1 Database Table Schemas

4.4.3 Phase Overview

4.4.4 Architecture Diagram

4.4.5 Data Flow Diagram

4.4.6 Design Details

4.5 Courts

4.5.1 Technologies Used

4.5.2 Component Overview

4.5.2.1 Database Table Schemas

4.5.3 Phase Overview

4.5.4 Architecture Overview

4.5.5 Design Details

4.6 Workshops

4.6.1 Technologies Used

4.6.2 Component Overview

4.6.2.1 Database Table Schemas

4.6.3 Phase Overview

4.6.4 Architecture Overview

4.6.5 Design Details

4.7 Reports and Statistics

4.7.1 Technologies Used

4.7.2 Component Overview

4.7.2.1 Database Table Schemas

4.7.3 Phase Overview

4.7.4 Architecture Overview

4.7.5 Design Details

4.8 Surveys

4.8.1 Technologies Used

4.8.2 Database Table Schemas

4.8.3 Component Overview

4.8.4 Phase Overview

4.8.5 Architecture Overview

4.8.6 Design Details

4.9 Program Administration Options

4.9.1 Technologies Used

4.9.2 Component Overview

4.9.2.1 Database Table Schemas

4.9.3 Phase Overview

4.9.4 Architecture Overview

4.9.5 Design Details

4.10 Application Administration Options

4.10.1 Technologies Used

4.10.2 Component Overview

4.10.2.1 Database Table Schemas

4.10.3 Phase Overview

4.10.4 Architecture Overview

4.10.5 Design Details

4.11 Miscellaneous

5.0 System and Unit Testing

5.1 Overview

5.2 Dependencies

5.3 Test Setup and Execution

6.0 Development Environment

This section describes programs used to develop the application, where it will be hosted and what kind of environment the server is.

6.1 Development IDE and Tools

For MySQL schema model and database management, MySQL Workbench will be used: http://dev.mysql.com/downloads/workbench/. Aptana Studio 3 will be used as the IDE for development of the web application and the prototype: http://www.aptana.com/products/studio3. The Aptana IDE has integrated GitHub support.

6.2 Source Control

GitHub will be used for the primary file repository and source control. Builds will be pushed to GitHub after every editing session. The link is: https://github.com/SDSMT-CSC/TCD.

6.3 Dependencies

6.4 Build Environment

The build environment is a Linux server running Apache HTTP Web Server and MySQL Database Management System. The servers are located in Dallas, TX. The login credentials are:

- Domain Name:
  - http://teencourtdb.com
- CPanel Access for hosting management:
  - URL: http://teencourtdb.com/cpanel
  - Username: teencour
  - Password: ifHQS14zvcN2
- FTP Access for file upload/download:
  - Username: teencour
  - Password: ifHQS14zvcN2
- Dedicated IP address: 50.22.71.90

6.5 Development Machine Setup

The application will be built on the existing web space. In the event of any hosting or connection issues that may arise during development, the team has access to a private, temporary server with the same server environment. This way progress on the application will not be hindered while the issue is resolved.

7.0 Release | Setup | Deployment

7.1 Deployment Information and Dependencies

7.2 Setup Information

7.3 System Versioning Information

8.0 End User Documentation

Appendix I: List of Figures

Appendix II: Supporting Information and Details

II.1 Database Schema

```
USE `teencour_data` ;

-- -----------------------------------------------------
-- Table `teencour_data`.`program`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`program` (
  `programID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(150) NOT NULL ,
  `PAddress` VARCHAR(150) NULL ,
  `pCity` VARCHAR(45) NULL ,
  `pState` CHAR(2) NULL ,
  `pZip` VARCHAR(25) NULL ,
  `mAddress` VARCHAR(150) NULL ,
  `mCity` VARCHAR(45) NULL ,
  `mState` CHAR(2) NULL ,
  `mZip` VARCHAR(25) NULL COMMENT 'Mailing address zip code' ,
  `added` TIMESTAMP NOT NULL DEFAULT now() ,
  PRIMARY KEY (`programID`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `teencour_data`.`user`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`user` (
  `userID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `type` INT UNSIGNED NOT NULL ,
  `firstName` VARCHAR(75) NOT NULL ,
  `lastName` VARCHAR(75) NOT NULL ,
  `email` VARCHAR(255) NOT NULL ,
  `password` VARCHAR(255) NOT NULL ,
  `createDate` DATETIME NULL ,
  `loginDate` TIMESTAMP NOT NULL DEFAULT now() ,
  PRIMARY KEY (`userID`) ,
  INDEX `programID` (`programID` ASC) ,
  CONSTRAINT `fk_user_court`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `teencour_data`.`user_phone`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`user_phone` (
  `phoneID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `userID` INT UNSIGNED NOT NULL ,
  `phoneNum` VARCHAR(45) NOT NULL ,
  `ext` VARCHAR(5) NULL ,
  `type` VARCHAR(25) NULL ,
  PRIMARY KEY (`phoneID`) ,
  UNIQUE INDEX `phoneID_UNIQUE` (`phoneID` ASC) ,
  CONSTRAINT `fk_user_phone_user1`
    FOREIGN KEY (`userID` )
    REFERENCES `teencour_data`.`user` (`userID` ) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `teencour_data`.`school`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`school` (
  `schoolID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `schoolName` VARCHAR(45) NULL ,
  `address` VARCHAR(45) NULL ,
  `city` VARCHAR(45) NULL ,
  `state` VARCHAR(45) NULL ,
  `zip` VARCHAR(45) NULL ,
  `phone` VARCHAR(45) NULL ,
  `contact` VARCHAR(45) NULL ,
  `schoolcol` VARCHAR(45) NULL ,
  PRIMARY KEY (`schoolID`, `programID`) ,
  INDEX `fk_school_information_court1_idx` (`programID` ASC) ,
  CONSTRAINT `fk_school_information_court1`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `teencour_data`.`defendant`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`defendant` (
  `defendantID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `firstName` VARCHAR(45) NOT NULL ,
  `lastName` VARCHAR(45) NOT NULL ,
  `middleName` VARCHAR(45) NULL ,
  `homePhone` VARCHAR(45) NULL ,
  `dob` INT NOT NULL ,
  `courtCaseNumber` VARCHAR(45) NULL ,
  `agencyCaseNumber` VARCHAR(45) NULL ,
  `expungeDate` DATETIME NULL ,
  `closeDate` DATETIME NULL ,
  `schoolID` INT UNSIGNED NULL ,
  `grade` VARCHAR(45) NULL ,
  `height` VARCHAR(45) NULL ,
  `weight` VARCHAR(45) NULL ,
  `eyeColor` VARCHAR(45) NULL ,
  `hairColor` VARCHAR(45) NULL ,
  `sex` CHAR(1) NULL ,
  `race` VARCHAR(45) NULL ,
  `licenseNum` VARCHAR(45) NULL ,
  `licenseState` VARCHAR(45) NULL ,
  `added` TIMESTAMP NOT NULL DEFAULT now() ,
  PRIMARY KEY (`defendantID`, `programID`) ,
  INDEX `fk_defendant_court1_idx` (`programID` ASC) ,
  INDEX `fk_defendant_school1_idx` (`schoolID` ASC) ,
  CONSTRAINT `fk_defendant_court1`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) ,
  CONSTRAINT `fk_defendant_school1`
    FOREIGN KEY (`schoolID` )
    REFERENCES `teencour_data`.`school` (`schoolID` ) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `teencour_data`.`volunteer`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`volunteer` (
  `volunteerID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `firstName` VARCHAR(45) NULL ,
  `lastName` VARCHAR(45) NULL ,
  `phone` VARCHAR(45) NULL ,
  `email` VARCHAR(255) NULL ,
  PRIMARY KEY (`volunteerID`, `programID`) ,
  INDEX `fk_volunteer_court1_idx` (`programID` ASC) ,
  UNIQUE INDEX `volID_UNIQUE` (`volunteerID` ASC) ,
  CONSTRAINT `fk_volunteer_court1`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `teencour_data`.`court_location`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`court_location` (
  `locationID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `name` VARCHAR(45) NULL ,
  `address` VARCHAR(45) NULL ,
  `city` VARCHAR(45) NULL ,
  `state` CHAR(2) NULL ,
  `zip` VARCHAR(45) NULL ,
  PRIMARY KEY (`locationID`, `programID`) ,
  INDEX `fk_trial_location_program1_idx` (`programID` ASC) ,
  UNIQUE INDEX `locationID_UNIQUE` (`locationID` ASC) ,
  CONSTRAINT `fk_trial_location_program1`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) )
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `teencour_data`.`court`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`court` (
  `courtID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `defendantID` INT UNSIGNED NOT NULL ,
  `locationID` INT UNSIGNED NOT NULL ,
  `type` VARCHAR(45) NOT NULL ,
  `date` DATETIME NOT NULL ,
  `closed` TINYINT(1) NULL ,
  PRIMARY KEY (`courtID`, `programID`) ,
  INDEX `fk_jury_court1_idx` (`programID` ASC) ,
  INDEX `fk_jury_defendant1_idx` (`defendantID` ASC) ,
  INDEX `fk_trial_trial_location1_idx` (`locationID` ASC) ,
  CONSTRAINT `fk_jury_court1`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) ,
  CONSTRAINT `fk_jury_defendant1`
    FOREIGN KEY (`defendantID` )
    REFERENCES `teencour_data`.`defendant` (`defendantID` ) ,
  CONSTRAINT `fk_trial_trial_location1`
    FOREIGN KEY (`locationID` )
    REFERENCES `teencour_data`.`court_location` (`locationID` ) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `teencour_data`.`court_position`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`court_position` (
  `positionID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `position` VARCHAR(45) NOT NULL ,
  PRIMARY KEY (`positionID`) ,
  CONSTRAINT `fk_jury_position_court1`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `teencour_data`.`court_member`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`court_member` (
  `courtID` INT UNSIGNED NOT NULL ,
  `volunteerID` INT UNSIGNED NOT NULL ,
  `positionID` INT UNSIGNED NOT NULL ,
  `hours` DECIMAL(8,2) NULL ,
  PRIMARY KEY (`courtID`, `positionID`, `volunteerID`) ,
  INDEX `fk_jury_pool_jury1_idx` (`courtID` ASC) ,
  INDEX `fk_trial_members_trial_position1_idx` (`positionID` ASC) ,
  CONSTRAINT `fk_jury_pool_volunteer1`
    FOREIGN KEY (`volunteerID` )
    REFERENCES `teencour_data`.`volunteer` (`volunteerID` ) ,
  CONSTRAINT `fk_jury_pool_jury1`
    FOREIGN KEY (`courtID` )
    REFERENCES `teencour_data`.`court` (`courtID` ) ,
  CONSTRAINT `fk_trial_members_trial_position1`
    FOREIGN KEY (`positionID` )
    REFERENCES `teencour_data`.`court_position` (`positionID` ) )
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `teencour_data`.`user_log`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`user_log` (
  `logID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `userID` INT UNSIGNED NOT NULL ,
  `action` VARCHAR(45) NOT NULL ,
  `ip_address` VARCHAR(45) NOT NULL ,
  `date` TIMESTAMP NOT NULL DEFAULT now() ,
  PRIMARY KEY (`logID`) ,
  UNIQUE INDEX `logID_UNIQUE` (`logID` ASC) ,
  CONSTRAINT `fk_user_log_user1`
    FOREIGN KEY (`userID` )
    REFERENCES `teencour_data`.`user` (`userID` ) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `teencour_data`.`guardian`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`guardian` (
  `guardianID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `defendantID` INT UNSIGNED NOT NULL ,
  `firstName` VARCHAR(45) NULL ,
  `lastName` VARCHAR(45) NULL ,
  `address` VARCHAR(150) NULL ,
  `city` VARCHAR(45) NULL ,
  `state` VARCHAR(45) NULL ,
  `zip` VARCHAR(45) NULL ,
  `homePhone` VARCHAR(45) NULL ,
  `emailAddress` VARCHAR(45) NULL ,
  `employer` VARCHAR(45) NULL ,
  `workPhone` VARCHAR(45) NULL ,
  `livesWith` TINYINT(1) NULL ,
  PRIMARY KEY (`guardianID`) ,
  UNIQUE INDEX `parentID_UNIQUE` (`guardianID` ASC) ,
  INDEX `fk_parents_defendant1_idx` (`defendantID` ASC) ,
  CONSTRAINT `fk_parents_defendant1`
    FOREIGN KEY (`defendantID` )
    REFERENCES `teencour_data`.`defendant` (`defendantID` ) )
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `teencour_data`.`custom_fields`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`custom_fields` (
  `fieldID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `fieldName` VARCHAR(45) NULL ,
  `required` TINYINT(1) NULL ,
  PRIMARY KEY (`fieldID`, `programID`) ,
  UNIQUE INDEX `fieldID_UNIQUE` (`fieldID` ASC) ,
  INDEX `fk_custom_fields_court1_idx` (`programID` ASC) ,
  CONSTRAINT `fk_custom_fields_court1`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) )
ENGINE = InnoDB;
   -- -----------------------------------------------------
-- Table `teencour_data`.`custom_data`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`custom_data` (
  `customID` INT UNSIGNED NOT NULL ,
  `defendantID` INT UNSIGNED NOT NULL ,
  `value` VARCHAR(45) NULL ,
  PRIMARY KEY (`customID`) ,
  INDEX `fk_custom_data_custom_fields1_idx` (`customID` ASC) ,
  CONSTRAINT `fk_custom_data_custom_fields1`
    FOREIGN KEY (`customID` )
    REFERENCES `teencour_data`.`custom_fields` (`fieldID` ) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `teencour_data`.`survey`
-- -----------------------------------------------------
CREATE   TABLE IF NOT EXISTS `teencour_data`.`survey` (
   `surveyID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
   `description` MEDIUMTEXT NULL ,
   `created` TIMESTAMP NOT NULL DEFAULT now() ,
   `sent` DATETIME NULL ,
   PRIMARY KEY (`surveyID`) ,
   UNIQUE INDEX `surveyID_UNIQUE` (`surveyID` ASC) )
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `teencour_data`.`survey_questions`
-- -----------------------------------------------------
CREATE   TABLE IF NOT EXISTS `teencour_data`.`survey_questions` (
   `questionID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
   `surveyID` INT UNSIGNED NOT NULL ,
   `question` VARCHAR(255) NULL ,
   `type` INT UNSIGNED NULL ,
   PRIMARY KEY (`questionID`, `surveyID`) ,
   UNIQUE INDEX `questionID_UNIQUE` (`questionID` ASC) ,
   INDEX `fk_survey_questions_survey1_idx` (`surveyID` ASC) ,
   CONSTRAINT `fk_survey_questions_survey1`
      FOREIGN KEY (`surveyID` )
      REFERENCES `teencour_data`.`survey` (`surveyID` ) )
ENGINE = InnoDB;
   -- -----------------------------------------------------
-- Table `teencour_data`.`survey_answers`
-- -----------------------------------------------------
CREATE   TABLE IF NOT EXISTS `teencour_data`.`survey_answers` (
   `surveyID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
   `userID` INT UNSIGNED NOT NULL ,
   `questionID` INT UNSIGNED NOT NULL ,
   `answer` VARCHAR(45) NULL ,
   PRIMARY KEY (`surveyID`, `userID`, `questionID`) ,
   INDEX `fk_survey_answers_survey1_idx` (`surveyID` ASC) ,
   INDEX `fk_survey_answers_survey_questions1_idx` (`questionID` ASC) ,
   INDEX `fk_survey_answers_user1_idx` (`userID` ASC) ,
   CONSTRAINT `fk_survey_answers_survey1`
      FOREIGN KEY (`surveyID` )
      REFERENCES `teencour_data`.`survey` (`surveyID` ) ,
   CONSTRAINT `fk_survey_answers_survey_questions1`
      FOREIGN KEY (`questionID` )
      REFERENCES `teencour_data`.`survey_questions` (`questionID` ) ,
   CONSTRAINT `fk_survey_answers_user1`
      FOREIGN KEY (`userID` )
      REFERENCES `teencour_data`.`user` (`userID` ) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `teencour_data`.`survey_options`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`survey_options` (
  `optionID` VARCHAR(45) NOT NULL ,
  `questionID` INT UNSIGNED NOT NULL ,
  `option` VARCHAR(45) NULL ,
  PRIMARY KEY (`optionID`) ,
  INDEX `fk_survey_options_survey_questions1_idx` (`questionID` ASC) ,
  CONSTRAINT `fk_survey_options_survey_questions1`
    FOREIGN KEY (`questionID` )
    REFERENCES `teencour_data`.`survey_questions` (`questionID` ) )
ENGINE = InnoDB;

  -- -----------------------------------------------------
-- Table `teencour_data`.`court_defendant_jury`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`court_defendant_jury` (
  `courtID` INT UNSIGNED NOT NULL ,
  `defendantID` INT UNSIGNED NOT NULL ,
  `positionID` INT UNSIGNED NOT NULL ,
  `hours` DECIMAL(8,2) NULL ,
  PRIMARY KEY (`courtID`, `defendantID`, `positionID`) ,
  INDEX `fk_jury_member_def_trial1_idx` (`courtID` ASC) ,
  INDEX `fk_jury_member_def_defendant1_idx` (`defendantID` ASC) ,
  INDEX `fk_court_jury_court_position1_idx` (`positionID` ASC) ,
  CONSTRAINT `fk_jury_member_def_trial1`
    FOREIGN KEY (`courtID` )
    REFERENCES `teencour_data`.`court` (`courtID` ) ,
  CONSTRAINT `fk_jury_member_def_defendant1`
    FOREIGN KEY (`defendantID` )
    REFERENCES `teencour_data`.`defendant` (`defendantID` ) ,
  CONSTRAINT `fk_court_jury_court_position1`
    FOREIGN KEY (`positionID` )
    REFERENCES `teencour_data`.`court_position` (`positionID` ) )
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `teencour_data`.`statute`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`statute` (
  `statuteID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `code` VARCHAR(50) NULL ,
  `title` VARCHAR(50) NULL ,
  `description` TEXT NULL ,
  PRIMARY KEY (`statuteID`, `programID`) ,
  INDEX `fk_citation_court1_idx` (`programID` ASC) ,
  UNIQUE INDEX `statuteID_UNIQUE` (`statuteID` ASC) ,
  CONSTRAINT `fk_citation_court1`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `teencour_data`.`court_common_place`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`court_common_place` (
  `commonPlaceID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `commonPlace` VARCHAR(45) NULL ,
  `address` VARCHAR(45) NULL ,
  `city` VARCHAR(45) NULL ,
  `state` CHAR(2) NULL ,
  `zip` VARCHAR(45) NULL ,
  `closeTo` VARCHAR(45) NULL ,
  PRIMARY KEY (`commonPlaceID`, `programID`) ,
  INDEX `fk_court_common_place_court1_idx` (`programID` ASC) ,
  CONSTRAINT `fk_court_common_place_court1`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) )
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `teencour_data`.`citation_officer`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`citation_officer` (
  `officerID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `firstname` VARCHAR(45) NULL ,
  `lastname` VARCHAR(50) NULL ,
  `idNumber` VARCHAR(50) NOT NULL ,
  `phone` VARCHAR(45) NULL ,
  PRIMARY KEY (`officerID`, `programID`) ,
  INDEX `fk_citation_officer_court1_idx` (`programID` ASC) ,
  UNIQUE INDEX `officerID_UNIQUE` (`officerID` ASC) ,
  CONSTRAINT `fk_citation_officer_court1`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `teencour_data`.`citation`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`citation` (
  `citationID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `defendantID` INT UNSIGNED NOT NULL ,
  `statueID` INT UNSIGNED NOT NULL ,
  `officerID` INT UNSIGNED NOT NULL ,
  `date` TIMESTAMP NULL ,
  `address` VARCHAR(45) NULL ,
  `city` VARCHAR(45) NULL ,
  `state` CHAR(2) NULL ,
  `zip` VARCHAR(45) NULL ,
  `mirandized` TINYINT(1) NULL ,
  `drugsOrAlcohol` TINYINT(1) NULL ,
  `commonPlaceID` INT UNSIGNED NULL ,
  `entered` TIMESTAMP NOT NULL DEFAULT now() ,
  PRIMARY KEY (`citationID`, `defendantID`) ,
  INDEX `fk_citation_court_common_place1_idx` (`commonPlaceID` ASC) ,
  INDEX `fk_citation_defendant1_idx` (`defendantID` ASC) ,
  INDEX `fk_citation_statute1_idx` (`statueID` ASC) ,
  INDEX `fk_citation_citation_officer1_idx` (`officerID` ASC) ,
  UNIQUE INDEX `citationID_UNIQUE` (`citationID` ASC) ,
  CONSTRAINT `fk_citation_court_common_place1`
    FOREIGN KEY (`commonPlaceID` )
    REFERENCES `teencour_data`.`court_common_place` (`commonPlaceID` ) ,
  CONSTRAINT `fk_citation_defendant1`
    FOREIGN KEY (`defendantID` )
    REFERENCES `teencour_data`.`defendant` (`defendantID` ) ,
  CONSTRAINT `fk_citation_statute1`
    FOREIGN KEY (`statueID` )
    REFERENCES `teencour_data`.`statute` (`statuteID` ) ,
  CONSTRAINT `fk_citation_citation_officer1`
    FOREIGN KEY (`officerID` )
    REFERENCES `teencour_data`.`citation_officer` (`officerID` ) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `teencour_data`.`citation_vehicle`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`citation_vehicle` (
  `vehicleID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `citationID` INT UNSIGNED NOT NULL ,
  `licenseNumber` VARCHAR(45) NULL ,
  `licenseState` CHAR(2) NULL ,
  `make` VARCHAR(45) NULL ,
  `model` VARCHAR(45) NULL ,
  `color` VARCHAR(45) NULL ,
  PRIMARY KEY (`vehicleID`, `citationID`) ,
  INDEX `fk_citation_vehicle_citation1_idx` (`citationID` ASC) ,
  UNIQUE INDEX `vehicleID_UNIQUE` (`vehicleID` ASC) ,
  CONSTRAINT `fk_citation_vehicle_citation1`
    FOREIGN KEY (`citationID` )
    REFERENCES `teencour_data`.`citation` (`citationID` ) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `teencour_data`.`citation_stolen_items`
-- -----------------------------------------------------
CREATE   TABLE IF NOT EXISTS `teencour_data`.`citation_stolen_items` (
   `itemID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
   `citationID` INT UNSIGNED NOT NULL ,
   `name` VARCHAR(45) NULL ,
   `value` DECIMAL(18,2) NULL ,
   PRIMARY KEY (`itemID`) ,
   INDEX `fk_citation_stolen_items_citation1_idx` (`citationID` ASC) ,
   CONSTRAINT `fk_citation_stolen_items_citation1`
      FOREIGN KEY (`citationID` )
      REFERENCES `teencour_data`.`citation` (`citationID` ) )
ENGINE = InnoDB;
   -- -----------------------------------------------------
-- Table `teencour_data`.`court_guardian`
-- -----------------------------------------------------
CREATE   TABLE IF NOT EXISTS `teencour_data`.`court_guardian` (
   `courtID` INT UNSIGNED NOT NULL ,
   `guardianID` INT UNSIGNED NOT NULL ,
   PRIMARY KEY (`courtID`, `guardianID`) ,
   INDEX `fk_trial_guardian_guardian1_idx` (`guardianID` ASC) ,
   INDEX `fk_trial_guardian_trial1_idx` (`courtID` ASC) ,
   CONSTRAINT `fk_trial_guardian_guardian1`
      FOREIGN KEY (`guardianID` )
      REFERENCES `teencour_data`.`guardian` (`guardianID` ),
   CONSTRAINT `fk_trial_guardian_trial1`
      FOREIGN KEY (`courtID` )
      REFERENCES `teencour_data`.`court` (`courtID` ) )
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `teencour_data`.`intake_information`
-- -----------------------------------------------------
CREATE   TABLE IF NOT EXISTS `teencour_data`.`intake_information` (
   `defendantID` INT NOT NULL ,
   `description` VARCHAR(50) NOT NULL ,
   `date` TIMESTAMP NOT NULL ,
   `inteviewer` VARCHAR(45) NULL ,
   `referred` TINYINT(1) NULL ,
   `dismissed` TINYINT(1) NULL ,
   PRIMARY KEY (`defendantID`) )
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `teencour_data`.`workshop`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`workshop` (
  `workshopID` INT UNSIGNED NOT NULL AUTO_INCREMENT ,
  `programID` INT UNSIGNED NOT NULL ,
  `date` DATETIME NULL ,
  `title` VARCHAR(45) NULL ,
  `description` TEXT NULL ,
  `instructor` VARCHAR(45) NULL ,
  `officerID` INT NULL ,
  PRIMARY KEY (`workshopID`, `programID`) ,
  INDEX `fk_workshop_program1_idx` (`programID` ASC) ,
  CONSTRAINT `fk_workshop_program1`
    FOREIGN KEY (`programID` )
    REFERENCES `teencour_data`.`program` (`programID` ) )
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `teencour_data`.`workshop_roster`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`workshop_roster` (
  `workshopID` INT UNSIGNED NOT NULL ,
  `defendantID` INT UNSIGNED NOT NULL ,
  `completed` DATETIME NULL ,
  INDEX `fk_workshop_roster_workshop1_idx` (`workshopID` ASC) ,
  INDEX `fk_workshop_roster_defendant1_idx` (`defendantID` ASC) ,
  PRIMARY KEY (`workshopID`, `defendantID`) ,
  CONSTRAINT `fk_workshop_roster_workshop1`
    FOREIGN KEY (`workshopID` )
    REFERENCES `teencour_data`.`workshop` (`workshopID` ),
  CONSTRAINT `fk_workshop_roster_defendant1`
    FOREIGN KEY (`defendantID` )
    REFERENCES `teencour_data`.`defendant` (`defendantID` ) )
ENGINE = InnoDB;

-- -----------------------------------------------------
-- Table `teencour_data`.`volunteer_position`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `teencour_data`.`volunteer_position` (
  `volunteerID` INT UNSIGNED NOT NULL ,
  `positionID` INT UNSIGNED NOT NULL ,
  PRIMARY KEY (`volunteerID`, `positionID`) ,
  INDEX `fk_volunteer_position_court_position1_idx` (`positionID` ASC) ,
  INDEX `fk_volunteer_position_volunteer1_idx` (`volunteerID` ASC) ,
  CONSTRAINT `fk_volunteer_position_court_position1`
    FOREIGN KEY (`positionID` )
    REFERENCES `teencour_data`.`court_position` (`positionID` ),
  CONSTRAINT `fk_volunteer_position_volunteer1`
    FOREIGN KEY (`volunteerID` )
    REFERENCES `teencour_data`.`volunteer` (`volunteerID` ) )
ENGINE = InnoDB;
```

Appendix III: Progress | Sprint Reports

This section will contain a complete list of all of the period progress and/or sprint reports which are deliverables for the phases and versions of the system.

III.1 Sprint 1 Progress Report

Download the complete Sprint 1 report from GitHub: Sprint Report 1.pdf

III.2 Sprint 2 Progress Report

Download the complete Sprint 2 report from GitHub: Sprint Report 2.pdf

III.3 Sprint 2 Progress Report

Download the complete Sprint 3 report from GitHub: Sprint Report 3.pdf