NRCS

# Volunteer Tracker

## Design Document/Application Plan

---

**Version 1.0**

Author: Keith Stewart

| Design Document/Application Plan | | NRCS Volunteer Tracker | |
|---|---|---|---|
| **Project Developer** | Keith Stewart | **Project Type** | U.S. Federal Government (FWS) |
| **Contractor PM** | Keith Stewart | **Government PM** | REDACTED |
| **Chief Programmer** | Keith Stewart | **Date** | 18 February 2016 |
| **Project File Location** | | Internal | |
| **Document Name** | | DesignDocumentNRCSV1.0.docx | |

# Table of Contents



# Figures



# Tables

## Introduction

This design document will serve as the controlling document for the NRCS Volunteer Tracker portal application. This document is also the standardized application plan template describing the primary flow and functionality of the application software. Post project deployment, this document will serve as the Technical Reference for the software application. Hereinafter, throughout this document, the NRCS Volunteer Tracker Software application will be referred to as the NRCSv software.

Key high-level components of this document include the following:

1. NRCSv application information architecture
2. NRCSv application functionality
3. NRCSv application data structure/organization
4. NRCSv application schedulers
5. NRCSv application key included library files

Each component (1—5) is described in greater detail in the corresponding sections of this document.

## 1. Information Architecture

The information architectural framework for the NRCSv application is divided into two functional categories consisting of the Application Architecture and the Network Architecture. Since the application architecture is specific to the NRCSv software, greater detail will be provided for the application architecture category. Future Federal Government plans (2016) specify that the NRCSv application will migrate to and reside on either a DOI or NPS Cloud network architectural framework. As such, this document will describe the "as is" network architecture and any specifics relevant to the NRCSv. Future updates to this document should provide details relative to the DOI/NPS Cloud network architecture upon deployment to the Cloud.

### 1.1. Application Architecture

The application architecture refers to the type of *application processing middleware* that processes the NRCSv application as well as the specific *design and organization* of the information components (core code base) that make up the NRCSv application. The following sections will discuss each of these three application architecture areas.

#### 1.1.1. Application Processing Middleware

To process the multiple code components (pages, templates, included files, scripts, database queries, etc.) for this web application, a processing middleware is required. The NRCSv application uses Adobe® ColdFusion™ Standard Version 10 installed on a Windows Server 2008 R2 Enterprise edition software consisting of Dual Core 2.4 Ghz processors on a 64-bit machine. Additionally, the ColdFusion software connects to the Volungeer.gov GSS Microsoft® SQL Server™ 2012 edition database through native and ODBC data connectors (see details in Section 2—Data Structures below). *Important note about ColdFusion Version 10.*[1]

#### 1.1.2. Generalized Core Code Design

The NRCSv application is a collection of multiple ColdFusion templates that perform intended functions as specified in the application design. Templates are considered the core code base for the software application. Each of these templates can be abstracted in terms of a contained

[1] The NRCS Volunteer Tracker application uses ColdFusion version 10 for production, however, it was also developed and tested on ColdFusion version 11 Standard Edition. In the event of a ColdFusion upgrade to version 11, the application is expected to perform without issues and without modifications.

set of programming instructions that are executed when the template is accessed by a web browser or when the template is called/requested by another template or an AJAX synchronous scrip call. In general terms, each template involves three main component areas when processing:

1) Component 1: The code Template file with instructions about what function(s) to execute.
2) Component 2: The global variables file (Application.cfm) which includes key global application variables, including user authentication.
3) Component 3: The included templates and/or other files that provide information and rules to help the Template (Component 1) perform its function(s).

Figure #1 and Table #1 below detail the typical application architecture core code design used in the NRCSv application. In Table #1, a sample scenario depicting a user login is demonstrated to explain the interaction of the application design. Note that this is a *stylized* simple-case scenario; this type of scenario is routinely common throughout the application.

**Figure 1: Core Code Processing/Functionality**

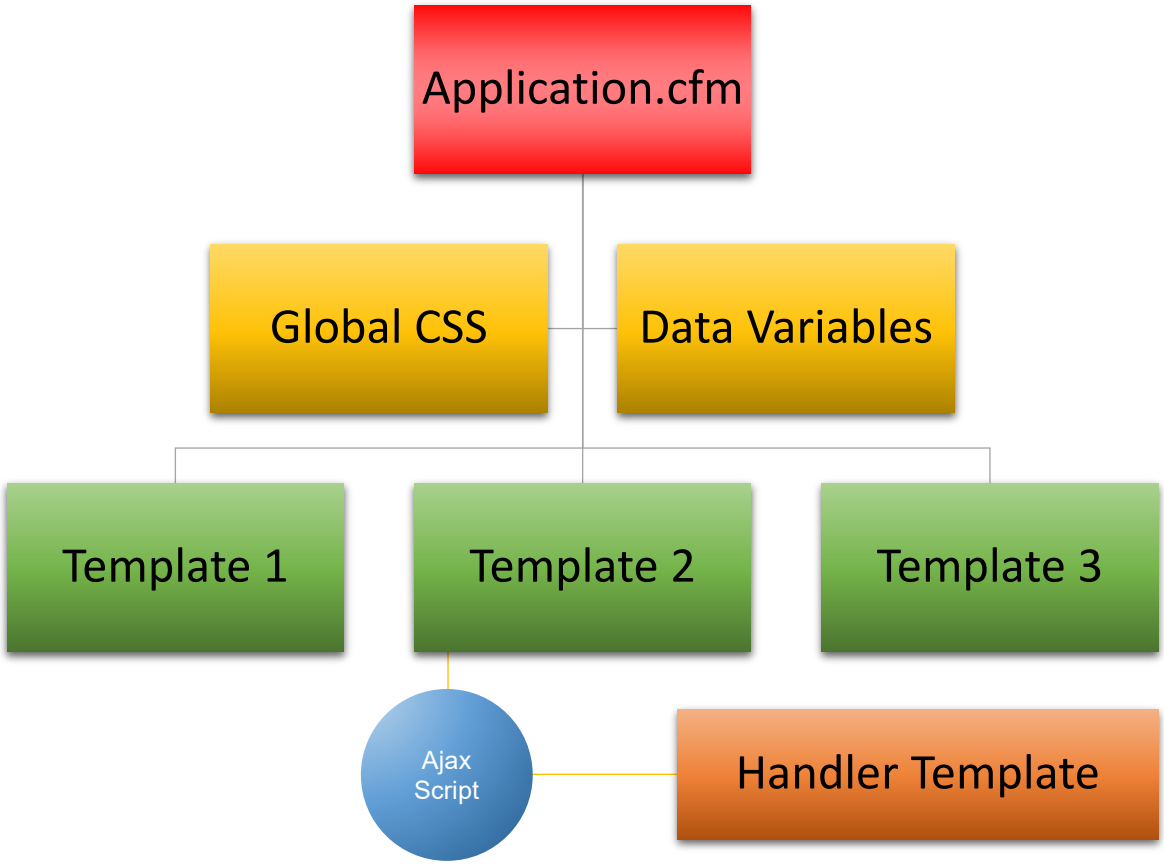



**Table 1: Core Code Details**

| Figure 1 Item | Description |
|---|---|

| | |
|---|---|
| Application.cfm | **Processed File:** global variable file for NRCSv application; controls data source, session variables, error handling, system encryption key, and application security filtering |
| Global CSS | **Included File:** specifies style and display formatting rules |
| Data Variables | **Included File(s):** may be more than one file; checks if user is authenticated and sets authenticated user-specific variables; may include database query results in arrays, structures, session variables, QoQ (query of query) and JavaScript files |
| Template 1--3 | **Generated File(s):** dynamically generated ColdFusion files based on the Application.cfm, Global CSS, and Data Variables files and on the core cfm code in the template. This can include multiple templates based on the user interface and function being performed |
| Ajax Script | **Semi Generated Script(s):** AJAX scripts which run asynchronously on the server and/or client machine, generally called on a user click/action, these include core functional script calls to ColdFusion templates. Most AJAX scripts are partially generated via ColdFusion for variable/parameter handling |
| Handler Template | **Generated Files(s):** these are specialized ColdFusion template files which perform some specialized function. Generally, handler templates do NOT stand on their own, but rather perform some function and return the results to the calling application. Generated files are generally in HTML 5.0. |

**Typical Application Scenario: User Login Example**

1) User navigates to the Login page (template 1)
2) Application.cfm sets the data source as well as other variables and the application name (VT)
3) The Global CSS file sets the style and formatting for page display
4) The Data Variables file checks to see if the user is already authenticated:
    a. If user is authenticated, user is redirected to the intended NRCSv page (Template 3)
    b. If user is NOT authenticated, user is presented with username and password login fields (Template 1)
5) Template 1 is the user login page for non-authenticated users. The template checks to see if the connection is an https (secure) connection, if not, redirected to the https login page. User is required to enter username and password. Upon submission redirected to Template 2.
6) AJAX scripts call helper templates to check for login integrity and prevent cross-site scripting attacks.
7) Template 2 processes (validates) the username and password based on the parameters specified in the Application.cfm file (data source). If the username and password result in one and only one record, the Data Variables page is updated with authenticated user-specific session variables and the user is redirected to the intended NRCSv page (Template 3). If the username and password result in anything other than one and only one record, the user is redirected back to Template 1, the Data Variables page sets an internal counter, and the user is allowed two more tries (based on the internal counter) entering a correct username and password combination. After 3 unsuccessful attempts, the Data Variables file is updated with a "restriction" variable and the user is prohibited within the application for a given period. The process is complete.

**IMORTANT INFORMATION:** Key to all ColdFusion applications (as well as all web applications) is the global file which sets application parameters. For the NRCSv application, the global file is the Application.cfm file. The following KEY application parameters are established in the NRCSv Application.cfm file

| Application Parameter | Value |
|---|---|
| Application name | VT |
| CFToken and CFID | Dynamically set by ColdFusion per session |
| Data Source | volunteersql (virtual name, actual DSN set in CF Administrator) |
| ErrorEmail | support@volunteer.gov |

| | |
|---|---|
| todaydate | NOW() Date/Time function to get current server time (EST) |
| vDOM (virtual domain) | #cgi.server_name# (CF variable to detect current server) |
| vURL (virtual URL) | https://#vDOM#/NRCS |
| vFile (virtual file upload location) | #ExpandPath( '../' )# (not currently used for NRCSv) |
| REMOVED | REMOVED |
| Request Errors | ErrorRequest.cfm |
| Exception Errors | ErrorException.cfm |
| UA (User Agent) | #cgi.HTTP_USER_AGENT# (CF variable to detect user browser) |

Red = Sensitive: DO NOT DISCLOSE except to authorized developers

### 1.1.3. Core Code Organization

As is standard in web application software, the core component files (templates, etc.) are organized in logical folder groupings within a web-accessible directory structure on the web server. For the NRCSv application, file structure follows a common organization that is utilized across all Volunteer.gov GSS software applications. Figure #2 and Table #2 detail this core code organization for the NRCSv application.

**NOTE:** All the below code files are physically on the REDACTED server within the REDACTED top-level web directory and the REDACTED second-level web subdirectory. When referring to the NRCSv application, the working directory will be assumed to be the top-level application directory, even though it is contained in the web folders mentioned above.

**Figure 2: Code Organization (Directory Structure)**


**Table 2: Code Organization Details**

| Figure 2 Folder | Description |
|---|---|
| NRCS | This is the application top directory folder that contains the primary code files for the NRCSv application. Note that there are other internal ColdFusion processing files which are outside this folder and which are shared by all applications on the Volunteer.gov server |
| /library | This is an application sub directory folder that contains any required JavaScript files and other external connector files (Google maps keys/APIs, search engines, etc.). Generally, files within the /library are script files, text files, or cfm files. Note: we do not include CSS files in the library; these are placed in the top web folder for accessibility to templates. |
| /images | This is an application sub directory folder that contains images used throughout the NRCSv application. Note: some images may be external to this application as specified in the Global CSS file. Generally, we try to avoid external images because they can cause slower page performance. |

| Figure 2 Folder | Description |
|---|---|
| /handlers | This is an application sub directory folder that contains key functional "helper" code files/templates which perform specialized queries, display, calculations, etc. Normally, these code files could be contained in their parent templates, but because they may involve complex processes and processes that are used by multiple templates, it is a common programming practice to separate "reusable code blocks" into individual files for easier use and access. |
| cfm template files | These are the actual ColdFusion template files that are the core code for the NRCSv software application. These files reside in the top directory folder (NRCSv). NOTE: there are two critical files and two sub-critical files which are always within this directory:<br><br>**Critical Files (always processed):**<br><br>*Application.cfm:* The global variables file. This file is processed every time a cfm file is accessed; this sets the application variables required for the application to function.<br><br>*vtStyle.css:* linked file which includes all the CSS rules for the application. The global CSS file is CSS3 compliant.<br><br>**Sub-Critical Files (processed when called):**<br><br>*validateMe.cfm:* Included file, checks to validate if a user is authenticated or not, processed for every page requiring an authenticated user. If authentication fails, all subsequent processing is aborted and the user is redirected to https login page. The *validateMe.cfm* file contains user specific parameters and user agent parameters. NOTE: in general, to speed up cfm processing, we convert locked session variables to unlocked template variables in this file on a template-by-template basis for performance enhancement. This file is processed only when called. The following key logic applies to the *validateMe.cfm* template:<br><br>  <br>The following user-specific parameters are converted from a session structure (session.GateKeeper.*x* where *x* is the session variable) to page variables so that they can be called/accessed without setting a CFLOCK for each request.<br><br>• AL (user access level)<br>• ST (user State location from address)<br>• myAgency (user Agency, in this case NRCS)<br>• WS (a string variable with all user work stations. This can be a comma separated list for more than 1 workstation.)<br>• VTID (user unique ID)<br>• MyPartner (user Agency, in this case NRCS)<br>• FN (user first name)<br>• LN (user last name)<br>• thisuser (unique user ID)<br><br>In addition to the above files, this directory includes all the other ColdFusion templates required to perform the NRCSv application functions. On rare occasions, we may create additional sub-folders for cfm templates when there is a clear advantage to segregating functionality within a sub-folder. |

## 1.2. Network Architecture

As of 21 February 2016, the NRCSv software application resides on a DOI/IBC server in the Reston Data Center. For a detailed Network Architecture Diagram, consult the appropriate DOI/IBC Network Operations Administrator. For security, Network Architecture is NOT presented as part of this document. Furthermore, it is *anticipated* that sometime 2016, the NRCSv software application

will be deployed with the DOI/NPS Cloud Network framework. Details of the DOI/NPS Cloud Network should be directed to the appropriate DOI/NPS Cloud Network administrator. For the purposes of the NRCSv application, it should be noted that the current DOI/IBC strategy is to force all Network traffic from the IBC Netscaler to the Volunteer.gov server via port 443 so that all traffic is via and encrypted SSL layer. The IBC also manages the SSL Certificate for the Volunteer.gov server where the NRCSv application is resident.

## 2. Key Application Functional Features

The following sections provide basic details about the core application logic. This is for the benefit of support and development teams who are responsible for managing the application. The following sections are listed based on key functional application areas.

### 2.1. User Access and Privileges

Users are authenticated based on their username and password combination. Privileges are assigned based on their user level with National Administrators being the top-level application managers and volunteers being the base level. A brief listing of responsibilities follows:

**National Administrators:** responsible for managing all volunteers and groups, managing work activities, managing administrators, viewing coordinator emails, managing offices, and also viewing MLRA reports, Year-End reports, and Volunteers with 4000+ hours of accumulated time.

**State Administrators:** responsible for volunteers and volunteer groups within their respective state (state is designated by their administrator).

**Area Administrators:** responsible for volunteers and volunteer groups within their respective areas (areas are designated by their administrators).

**Office Administrators:** responsible for volunteers and volunteer groups within their respective office list (offices lists are designated by their administrator).

**Volunteers:** responsible for entering their own time, printing/downloading their record of time, and maintaining their own contact information

**Group Volunteers:** group volunteers do not have access to the system; they are entered by administrators and assigned time.

Based on the above user categories, when a user is authenticated, they are directed to the main portal page based on the following strategy:

| User Level | Redirection |
|---|---|
| Administrators | admin.cfm |
| Volunteers | vol.cfm |

All administrative functions are available from the admin page and are restricted based on the user level. Volunteers can only enter time and manage their contact information.

To make the process of user authentication efficient, all application pages (except the home page/login page: index.cfm) include the validateMe.cfm include template. This template checks for the existence of a session variable in the GateKeeper session structure. The GateKeeper session structure parameters are set upon a successful login. The validateMe page runs a check for authentication, if passed, it then sets a CF-Lock (ColdFusion machine lock of multithreading) and converts the GateKeeper session variables to page variables so that a new CF-Lock does not have to be set every time the app requests a GateKeeper variable (VTID, Name, State, User Level, etc.). This allows for the segregation of appropriate access points within the admin page; basically, administrators only see or are allowed to do what their pre-defined user level allows.

If a user is unable to authenticate (after 3 unsuccessful login attempts), they are redirected to the /tossOff subdirectory and a session variable is set which disallows them to login again. This session variable is set for 24 hours but can be reset by closing all browser instances and starting the web browser again. We do NOT disclose this to users, but this functionality is available for support teams as needed.

## 2.2. Reporting

Reporting is also coupled with user level, where National Administrators have access to several reports that other administrators do not have access to. In most reporting and other administrative functions, the NRCSv application uses asynchronous AJAX calls generally to CFM template files in the /handlers subdirectory to accomplish reporting. Most of these reports are numbered based on the report category being accessed. For example, there are a total of 6 standard reports (configurable with dates and offices and other parameters) which all use AJAX calls to their respective handler file (R01, R02, R03, etc.). There are a total of 8 custom reports which follow the same logic but are more customizable than the standard reports. All reports use AJAX to call their respective handler file (C01, C02, C03, etc.).

Furthermore, each report (standard and custom) includes a feature which allows authenticated users to download the report into an Excel file format. These excel files are created in real-time and are NOT stored on the server. The user is required to either save or open the Excel file on their own client. The Excel files are generated based on the ColdFusion CFDOCUMENT strategy. Most Excel generation files include the naming convention *xxx*2XLS.cfm in their name structure.

When a user has generated a custom report, they are also able to save their custom report to their “Favorites”. This functionality inserts the report type together with their report variables and parameters to the MyFav table along with the user’s unique VTID. The reports are then accessed by appending the variables to the report type via URL variables.

## 2.3. Other Administrative Functions

National Administrators are also authorized to perform the following tasks on the NRCSv application:

**Manage Offices:** Archive or un-archive offices, add new offices, move offices to other states, change office designations to National/State/MLRA, and edit any address information about offices. Because the NRCS supplies the OIP designation for each office, it is their responsibility to ensure the integrity of OIP numbering. There are no provisions in the application to authenticate OIP numbers and Administrators are prohibited from creating duplicate OIPs.

**Manage Coordinator Email:** Manage/Edit/Create email addresses for each State Coordinator.

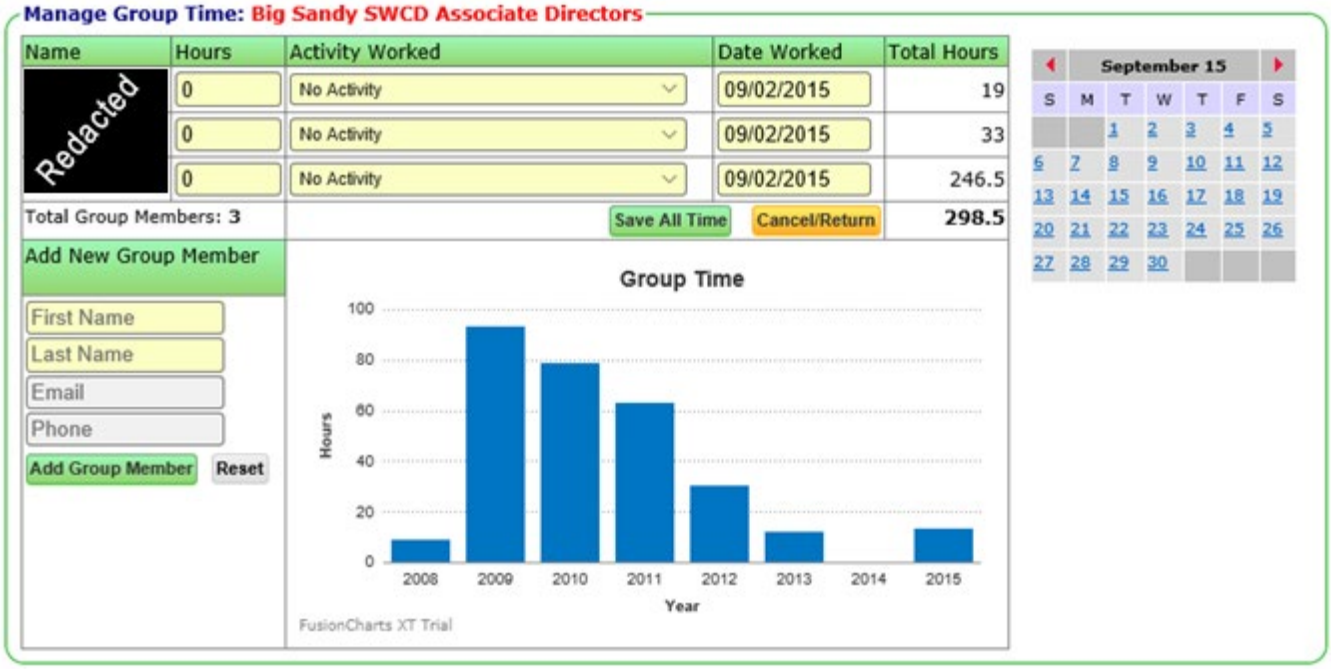
**View Administrators:** View and download to Excel all administrators and access administrators to manage their accounts.

**Manage Work Activities:** View/Edit/Create work activities. These work activities will be visible in the time sheet drop down work activity lists for all users. Note: for system integrity, we do not allow for the deletion of work activities. If a work activity is no longer needed, it can be set to NOT visible in the manage work activities area.

**Verify Time:** National Administrators can verify or un-verify ANY time entry in the system.

**Manage Groups:** Create new groups, manage existing groups, add groups members, and enter group time. The Groups administrative functions are covered in a separate design document available only to administrators. A sample screen of the Groups Time management portal is displayed in Figure #3 below.

**Figure 3: National Administrators—Group Time Management**




## 2.4. Script/Chart File Locations

The NRCSv application uses several JavaScript files for presentation control and AJAX initiation. These files are located within the /library subdirectory. There is also a sortable.js file which allows for the sorting of a data table based on a user click if the table is tagged with the controlling CSS class (sortable). Section 5 of this document list the basic function of each JavaScript file and other /library files.

In addition, the home page (index.cfm) of the NRCSv application includes an interactive USA map with the number of total volunteers included for each state. The map is in the /charts subdirectory/. This map is generated from a FusionCharts™ JavaScript component and populated dynamically from a nightly volunteer query (SCHHours.cfm). The data table below the USA map is also generated from this scheduled file (SCHHours.cfm). The controlling FusionCharts scripts and map

engine is shared with other applications on Volunteer.gov and is included in the fusioncharts/ folder which is above the top-level NRCSv folder within the REDACTED web server directory. Control of the USA map is via the fusion charts .JS include in the <head> section of the index file.

Additionally, within the Group Time templates (addTime.cfm), there are included FusionChart bar charts indicating the total hours worked for each group per year. Again, these charts are controlled via the FusionChart included JavaScript files located in the <head> section of the admin.cfm page and via a template include (/charts/gTime.cfm) within the addTime.cfm template.

### 2.5. Authentication Considerations

The authentication process on the NRCSv application occurs via a username and password combination authentication process where passwords are encrypted via a SHA 512 HASH and the login form is always processed via port 443 via SSL certificate. The specific process is as follows:

1) index.cfm login page (SSL username and Password required)
2) validation of username and password in the nValidate.cfm template.

In the form field from the index.cfm page to the nValidate.cfm page, the NRCSv application includes a form field named VGUUID and a value of the unique ColdFusion CFTOKEN value which is generated per user session. The CFTOKEN is encrypted via AES and the stored encryption key (REDACTED variable). When the form is submitted, the nValidate.cfm page then decrypts the VGUUID value and compares it to a generated CFTOKEN. If there is a match, the processing is allowed to proceed. If there is NOT a match, the user agent is then redirected to the sorry.cfm page and processing is stopped. This VGUUID is used as a means of ensuring that the login attempt originated from the index.cfm page on the server. This also prevents cross site scripting and/or hijacking of the login process.

### 2.6. Password Reset

The NRCSv application includes a Password Reset function whereby a user can reset their password. The process works as follows:

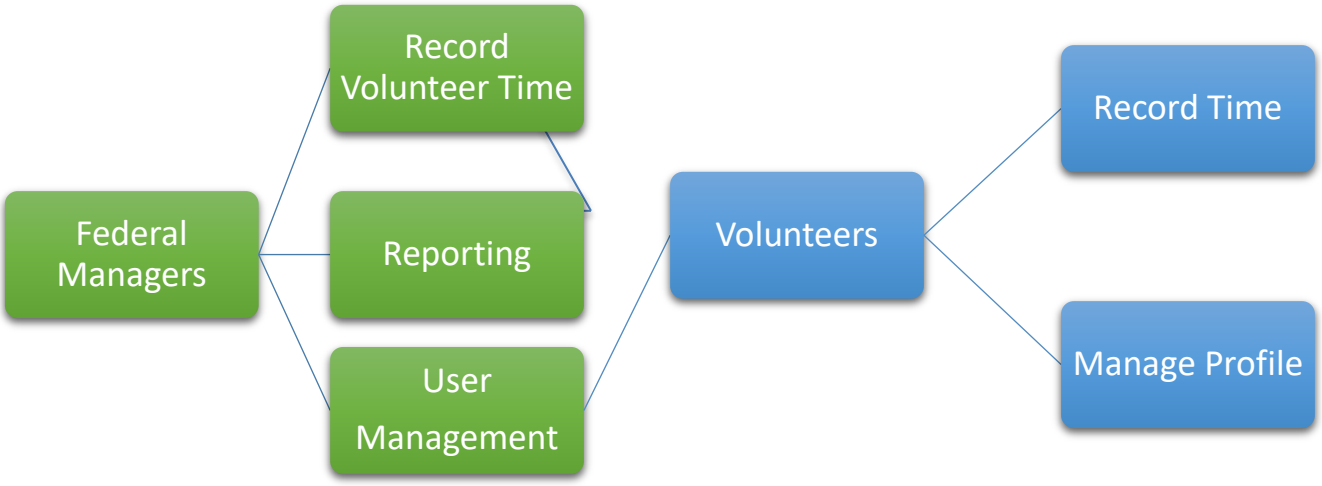
1. On the home page (index.cfm), user clicks the Reset button and follows the instructions
2. The user is redirected to the NRCS.cfm reset page. The user must enter their email address or username in the field provided.
3. A unique encrypted VGUUID form variable is created (see 2.5 above), and the username entered by the user is passed to the checkReset.cfm template via SSL. If the decrypted VGUUID does not match the CFTOKEN value, the processing is aborted and the user agent is directed to the sorry.cfm page.
4. If the username entered matches ONE and ONLY ONE username in the VT_Users table, and a password reset email is generated and sent to the user account email address.
5. The temporary password is set in the database with a 2-hour expiration. If the user fails to reset their password within the 2-hour time limit, the user must again request a password reset as above.
6. The password reset email includes a temporary password that is encrypted along with instructions for resetting the password.

## 3. Data Structure

The NRCSv application is primarily a data-driven software user interface and management tool. Key functionality includes roles-based content and user management allocated to two broad categories of users including Federal Government Managers and Non-Federal Government Volunteers. For the NRCSv application, Federal Government Managers are primarily responsible for entering Non-Federal Volunteer time; there is limited Non-Federal Government access to this application.

Within the Federal Managers roles, there are several levels of roles-based users and associated functions. Within the Non-Federal Volunteer users, roles-based functions are limited to entering time and printing/saving a record of time only. This non-federal volunteer role applies to individuals; specifically, an individual non-federal volunteer can only manage their own time and profile record. Figure # 4 and shows a high-level relationship among the users and primary roles.

**Figure 4: Generalized Roles-Based Data Structure**




Expanding from the Generalized Roles-Based Data Structure (Figure #4), a more detailed relationship view of the roles within the NRCSv application are presented in Figure #5 below (Detailed Roles-Based Functions). These functions are key to designing a manageable and productive database structure.

**Figure 5: Detailed Roles-Based Data Structure**

| **(1) NATIONAL ADMINISTRATORS** | |
|---|---|
| **(2) State Administrators**<br>Manage users, volunteers, and groups within their designated *state*.<br>**Restrictions:** managing offices, managing coordinator email, viewing administrators, managing work activities, viewing volunteer 4000-hour reports, viewing Year-End reports, viewing MLRA Office/Volunteer reports | **(3) Area Administrators**<br>Manage users, volunteers and groups within their designated *area based on offices codes* (OIP).<br>**Restrictions:** managing offices, managing coordinator email, viewing administrators, managing work activities, viewing volunteer 4000-hour reports, viewing Year-End reports, viewing MLRA Office/Volunteer reports |
| **(4) Office Administrators**<br>Manage users, volunteers and groups within their designated *area based on offices codes* (OIP). | **(5) Volunteers** |

| | |
|---|---|
| **Restrictions:** managing offices, managing coordinator email, viewing administrators, managing work activities, viewing volunteer 4000-hour reports, viewing Year-End reports, viewing MLRA Office/Volunteer reports | Can only enter their own time as well as update their contact information. Can only enter time for offices they are assigned to.<br>**Restrictions:** All. Can only manage their own time and record. Cannot manage or view any other information or users. |

**NOTE:** Figure 5 explanation: National Administrators transcend all users, reports, and functions. The numbers in brackets (1), refer to the account authority level, 1 being the highest and 5 the lowest. Each level can manage other users at the same level or below, with the exception that level 5 (Volunteers) can only manage their own accounts; they cannot manage other volunteers. Additionally, the NRCSv system includes a Regional Administrator role. This role would be directly below the National Administrator. The Regional Administrator role is NOT in use; it is reserved for possible future use.

**Special Volunteer Role Note:** The NRCS staff create volunteer roles and assign workstations (OIP) for volunteers. Volunteers do not directly create their own accounts. Additionally, the NRCS enters time for MOST volunteers.

Based on the primary roles-based functionality of the NRCSv application, data structure and data relationships are key to functionality and performance of the application and user interface. Accordingly, the following data structures (data elements, keys, and data types) have been created in the Volunteer.gov database for the NRCSv application. At the bottom of each data table, please see important notes relative to that table. Each of the following data tables represent the actual tables in the NRCSv database.

**Table 3: Data Table: Users**

| SQL Name: VT_Users | | | |
|---|---|---|---|
| **#** | **Data Name** | **Data Type/Length** | **Description** |
| 1 | VTID | int | Primary key, auto increment |
| 2 | userID | [redacted] | user ID (generally email address) |
| 3 | PWD | [redacted] | password, SHA 512 HASH REDACTED |
| 4 | fname | [redacted] | user first name |
| 5 | lname | [redacted] | user last name |
| 6 | email | [redacted] | email address |
| 7 | phone | nvarchar 50 | phone |
| 8 | street1 | nvarchar 150 | street address line 1 |
| 9 | street2 | nvarchar 150 | street address line 2 |
| 10 | street3 | nvarchar 150 | street address line 3 |
| 11 | city | nvarchar 100 | user city |
| 12 | state | char 2 | user state |
| 13 | zip | nvarchar 10 | user zip code |
| ~~14~~ | ~~country~~ | ~~char 2~~ | no longer used for NRCS |
| 15 | gender | char 1 | gender |
| ~~16~~ | ~~dob~~ | ~~nvarchar 20~~ | no longer used for NRCS |
| 17 | dateCreated | smalldatetime | account creation date |
| 18 | status | char 10 | 1=active 0=archived |

| SQL Name: VT_Users | | | |
|---|---|---|---|
| # | Data Name | Data Type/Length | Description |
| ~~19~~ | ~~workSites~~ | ~~nvarchar 4000~~ | no longer used for NRCS |
| 20 | adminLevel | int | user access level (1) |
| 21 | partner | nvarchar 50 | agency (default NRCS) |
| ~~22~~ | ~~regionName~~ | ~~nvarchar 50~~ | no longer used for NRCS |
| ~~23~~ | ~~areaName~~ | ~~nvarchar 150~~ | no longer used for NRCS |
| 24 | officeName | nvarchar 4000 | comma separated list of work sites (OIP) |
| ~~25~~ | ~~groupName~~ | ~~nvarchar 250~~ | no longer used for NRCS |
| 26 | bgCheck | bit | user/volunteer background check 1=yes 0=no |
| 27 | student | bit | is a student 1=yes 0=no |
| 28 | international | bit | is international 1=yes 0=no |
| 29 | hosted | bit | hosted volunteer 1=yes 0=no |
| 30 | TempPassword | [redacted] | temporary password SHA HASH 512 |
| 31 | TPWexpire | datetime | temporary password expire date/time |

## Table Notes (VT_Users)

**Discussion:** several data items in this table were previously used by NON NRCS volunteer tracking on Volunteer.gov. These data items are listed above in gray background; they are not used in the NRCSv application.

**Database:** Volunteer.gov Server (DSN: volunteerSQL, SQL Name: VG)

| Table Color Key | |
|---|---|
| Primary Keys | |
| Foreign Keys | |
| Obsolete Data | |
| Redacted Data | |

**(1) adminLevel (user access levels) values:**

| Level | Role | Authority Level |
|---|---|---|
| 6 | National Administrator | Highest |
| 5 | Regional Administrator | Not used |
| 4 | State Administrator | |
| 3 | Area Administrator | |
| 2 | Office Administrator | |
| 0 | Volunteer | Lowest |

**status (account status) values:**

0 archived
1 active

**CRITICAL:**

[redacted]

### Table 4: Data Table: Work Activities

| SQL Name: vt_wa | | | |
|---|---|---|---|
| # | Data Name | Data Type/Length | Description |
| 1 | WAID | Int pk | Primary key, auto increment |

| SQL Name: vt_wa | | | |
|---|---|---|---|
| **#** | **Data Name** | **Data Type/Length** | **Description** |
| 2 | wactivity | nvarchar 400 | work activity name |
| 3 | wstatus | bit | work activity status 1=display 0=do not display |

**Table Notes (vt_wa)**

**Discussion:** The vt_wa table includes work activities that are displayed (or not displayed) for volunteer time sheets. These are editable by national administrators.

| Table Color Key | |
|---|---|
| Primary Keys | |
| Foreign Keys | |
| Obsolete Data | |

## Table 5: Data Table: Favorite Reports

| SQL Name: vtFav | | | |
|---|---|---|---|
| **#** | **Data Name** | **Data Type/Length** | **Description** |
| 1 | FVID | Int pk | primary key, auto increment |
| 2 | VTID | Int fk | link to user ID table |
| 3 | sDate | smallDateTime | start date for report |
| 4 | eDate | smallDateTime | end date for report |
| 5 | OfficeName | nvarchar 1000 | comma separated list of office names (OIP) |
| 6 | FavName | nchar 250 | name of favorite report |
| 7 | FavLink | nvarchar 250 | link to report type |
| 8 | vType | nvarchar 100 | type of volunteer for custom report 6 |
| 9 | DateAdded | smallDateTime | date favorite report added/updated (not implemented) |

**Table Notes (vtFav)**

**Discussion:** The vtFav table includes links and report parameters for custom reports generated by users and saved to their favorite list.

| Table Color Key | |
|---|---|
| Primary Keys | |
| Foreign Keys | |
| Obsolete Data | |

## Table 6: Data Table: Groups

| SQL Name: vtGroups | | | |
|---|---|---|---|
| **#** | **Data Name** | **Data Type/Length** | **Description** |
| 1 | GID | int | primary key, auto increment |
| 2 | state | nvarchar 25 | group state |
| 3 | groupName | nvarchar 250 | group name |
| ~~4~~ | ~~nationalG~~ | ~~bit~~ | no longer used by NRCS |
| 5 | regionName | nvarchar 25 | default National (otherwise, not used) |
| ~~6~~ | ~~areaName~~ | ~~nvarchar 50~~ | no longer used by NRCS |
| 7 | officeName | nvarchar 50 | OIP number for office group associated with |
| 8 | leaderFName | nvarchar 100 | group leader first name |

| SQL Name: vtGroups | | | |
|---|---|---|---|
| # | Data Name | Data Type/Length | Description |
| 9 | leaderLName | nvarchar 100 | group leader last name |
| 10 | leaderEmail | nvarchar 150 | group leader email address |
| 11 | leaderPhone | nvarchar 25 | group leader phone number |
| 12 | dateAdded | smalldatetime | date group added |
| 13 | dateUpdated | smalldatetime | date group updated |
| 14 | archived | bit | is group archived 0=no 1=yes |
| 15 | recurring | bit | recurring group =1, one-time group =0 |
| 16 | leaderAdd1 | nvarchar 50 | group leader address line 1 |
| 17 | leaderAdd2 | nvarchar 50 | group leader address line 2 |
| 18 | leaderCity | nvarchar 50 | group leader city |
| 19 | leaderState | nvarchar 2 | group leader state |
| 20 | leaderZip | nchar 10 | group leader zip code |

**Table Notes (vtGroups)**

**Discussion:** The vtGroups table includes all the specific information about volunteer group within the NRCSv application.

| Table Color Key | |
|---|---|
| Primary Keys | |
| Foreign Keys | |
| Obsolete Data | |

## Table 7: Data Table: Group Time

| SQL Name: vtGroupTime | | | |
|---|---|---|---|
| # | Data Name | Data Type/Length | Description |
| 1 | TID | Int pk | primary key, auto increment |
| 2 | GID | Int fk | foreign key, link from the Group Table |
| 3 | GUID | Int fk | foreign key, link from Group Users table, 0 for One-time group |
| 4 | DateWorked | smallDateTime | date worked |
| 5 | hoursWorked | float | hours worked |
| ~~6~~ | ~~locationName~~ | ~~nvarchar 250~~ | no longer used by NRCS |
| 7 | trainingState | Char 2 | State where training occurred |
| 8 | dateEntered | smallDateTime | date time was entered |
| 9 | activityWorked | nvarchar 100 | activity worked |
| 10 | groupNumber | float | number in group (0 if recurring, number if One-time group) |

**Table Notes (vtGroupTime)**

**Discussion:** The vtGroupTime table links through shared foreign keys to the vtGroups table via the GID key and the vtMembers table via the GUID key.

| Table Color Key | |
|---|---|
| Primary Keys | |
| Foreign Keys | |
| Obsolete Data | |

## Table 8: Data Table: Group Members

| SQL Name: vtMembers | | | |
|---|---|---|---|
| # | Data Name | Data Type/Length | Description |
| 1 | GUID | Int pk | Primary key, auto increment |
| 2 | GID | Int fk | Foreign key, link from groups table |
| 3 | vtMemberFname | nvarchar 50 | group member first name |
| 4 | vtMemberLname | nvarchar 50 | group member last name |
| 5 | vtMemberEmail | nvarchar 150 | group member email address |
| 6 | vtMemberPhone | nvarchar 50 | group member phone number |
| 7 | dateAdded | smallDateTime | date group member added |
| 8 | dateUpdated | smallDateTime | date group member last updated |
| 9 | archived | bit | is member archived (1=yes 0=no) |

**Table Notes (vtMembers)**

**Discussion:** The vtMembers table links through a shared foreign key to the vtGroups table via the GID key. This table includes group members time and contact information. It should be noted that group members are added via NRCS staff and do NOT have access to this application. Also, this group members table is for *Recurring Groups* only. One-time groups do not include a listing of names, only a listing of total members and total hours and date/work location.

| Table Color Key | |
|---|---|
| Primary Keys | |
| Foreign Keys | |
| Obsolete Data | |

### Table 9: Data Table: Volunteer Time

| SQL Name: myTime | | | |
|---|---|---|---|
| # | Data Name | Data Type/Length | Description |
| 1 | TID | int | Primary key, auto increment |
| 2 | ba | nvarchar 500 | Office Name from NRCSOIP table |
| 3 | AID | int | Foreign key, link from NRCSOIP table |
| 4 | time1 | float | hours time entry for Sunday |
| 5 | time2 | float | hours time entry for Monday |
| 6 | time3 | float | hours time entry for Tuesday |
| 7 | time4 | float | hours time entry for Wednesday |
| 8 | time5 | float | hours time entry for Thursday |
| 9 | time6 | float | hours time entry for Friday |
| 10 | time7 | float | hours time entry for Saturday |
| 11 | comments | nvarchar 500 | work activity |
| 12 | week | smalldatetime | week start (Sunday) for time entry |
| 13 | eid | int | Foreign key, link to VT_Users table |
| 14 | myCompany | nvarchar 200 | volunteer coordinator (administrator email address |
| 15 | tdate | nvarchar 15 | date time was entered updated |
| 16 | reg | float | total hours worked for period (week) |
| 17 | notes | nvarchar 25 | Agency (default NRCS) |
| 18 | verify | bit | has time been verified 1=yes 0=no |
| 19 | volState | char 2 | state volunteer activity occurred |

| SQL Name: myTime | | | |
|---|---|---|---|
| # | Data Name | Data Type/Length | Description |
| 20 | verifyBy | nvarchar 150 | time verified by (administrator name) |
| 21 | verifyDate | smalldatetime | date time was verified |
| ~~22~~ | ~~AwardType~~ | ~~nvarchar 100~~ | not used by NRCS |
| ~~23~~ | ~~AwardDate~~ | ~~smalldatetime~~ | not used by NRCS |
| ~~24~~ | ~~Award~~ | ~~bit~~ | not used by NRCS |

**Table Notes (myTime)**

**Discussion:** The myTimes table links through a shared foreign key to the VT_Users table via the EID key and the NRCSOIP table (work locations) table through the AID key. Time is recorded on a weekly timesheet from Sunday—Saturday for a given week. This table includes volunteer hours for *individual volunteers*, separate from the Group Timetable.

| Table Color Key | |
|---|---|
| Primary Keys | |
| Foreign Keys | |
| Obsolete Data | |

**Table 10: Data Table: NRCS OIP (Offices)**

| SQL Name: NRCSOIP | | | |
|---|---|---|---|
| # | Data Name | Data Type/Length | Description |
| 1 | NRCSID | int | primary key, auto increment |
| 2 | Address | nvarchar 255 | OIP street address |
| 3 | Agency | nvarchar 255 | agency name (NRCS) |
| 4 | City | nvarchar 255 | OIP city address |
| 5 | OfficeName | nvarchar 255 | OIP office name |
| 6 | OfficeType | nvarchar 255 | OIP office type |
| 7 | OIP | float | numeric OIP number supplied by NRCS |
| ~~8~~ | ~~SiteID~~ | ~~float~~ | numeric site number supplied by NRCS (not used) |
| 9 | SiteName | nvarchar 255 | site-specific office name |
| 10 | State | nvarchar 255 | OIP state address |
| 11 | ZipCode | nvarchar 255 | OIP zip code address |
| 12 | MarkDel | int | is OIP archived (1=yes, 0=no) |
| 13 | cEmail | nvarchar 150 | email address of coordinator for OIP |
| 14 | nationalGroup | int | is national office (1=yes, 0=no) |

**Table Notes (NRCSOIP)**

**Discussion:** The NRCSOIP table includes information about each NRCS office. This table was supplied by the NRCS. The table is editable by National Administrators.

| Table Color Key | |
|---|---|
| Primary Keys | |
| Foreign Keys | |
| Obsolete Data | |

## 4. Application Schedulers

The NRCSv application includes one ColdFusion scheduler task configured in the ColdFusion administrator as follows:

**Table 11: Application Scheduler Table**

| Schedule Name | File Name | Frequency | Details |
|---|---|---|---|
| Get NRCS Hours | SCHHours.cfm | Daily 2:00 AM EST | Generates a query of individual volunteer and group volunteer counts per state comprehensive of the entire application time frame. This query is then written to a text file (nrcsvol.txt). This text file is then included in the charts/USA.cfm chart which is displayed on the home page. |

## 5. Key Included Library Files

The NRCSv application includes several JavaScript files and associated "helper" files for document formatting and functionality. Table #12 below provides a brief explanation of these files:

**Table 12: Library Files Explanation**

| File Name | Location | Brief Explanation |
|---|---|---|
| checker.js | /library | checks for password integrity for account creation |
| dateCheck.js | /library | validates dates entered into the reports start/end date fields |
| fetchForm.js | /library | asynchronous AJAX form submission |
| fetchNew.js | /library | asynchronous dynamic AJAX URL submission |
| hover.cfm | /library | CFM template which creates a hover effect for help |
| Menu.js | /library | menu display/formatting |
| scrollMe.js | /library | creates a floating DIV object (used in Calendar on Groups Page) |
| sortable.js | /library | creates a clickable/sortable data table based on table columns |
| states.cfm | /library | converts State abbreviations into corresponding state full name |
| fusioncharts.js | ../../fusioncharts | creates maps/charts for display |

Examples of the FusionChart™ dynamically generated charts/maps are displayed below in Figures #6 and #7 for reference.

**Figure 6: Interactive Volunteer Count Map (FusionCharts™)**

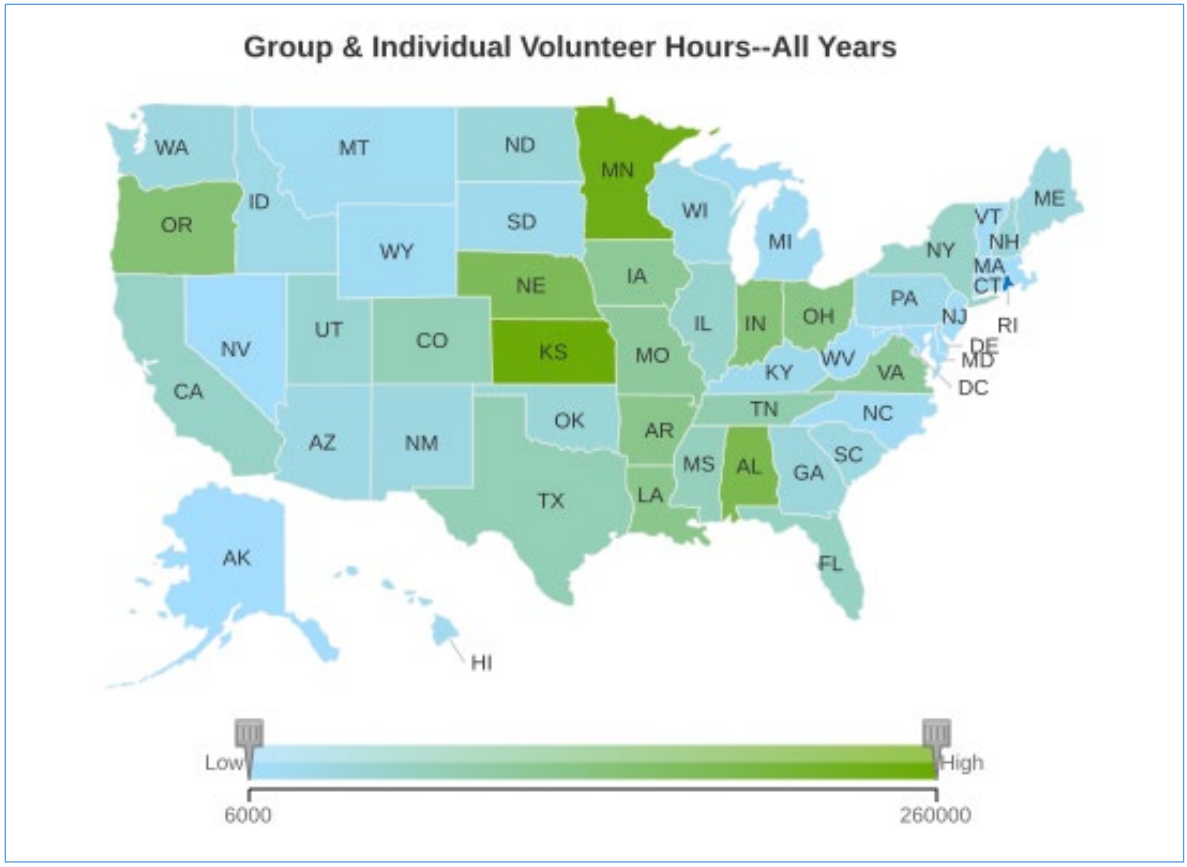
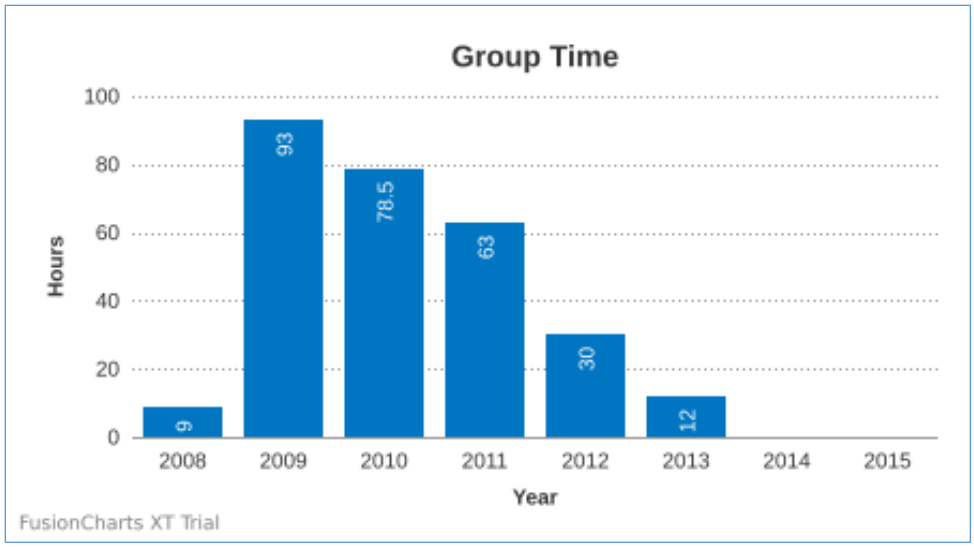



**Figure 7: Interactive Group Time Chart (Example from VA Group)**



## 6. Developer Information

The NRCSv app was developed under contract between the USDA NRCS and eSpherical.com, Inc. Contact information for the developer is provided below.

**DEVELOPER**
eSpherical.com. Inc.
Keith Stewart (Chief Developer)
2200 12th Court North, Suite 724
Arlington, VA 22201
REDACTED

----------========== END OF DOCUMENT ==========----------