DOCUMENT STORAGE SYSTEMS, INC.

# TECHNICAL OVERVIEW

## Maternity Tracker Dashboard

**Document History**

| Version | Date | Description |
|---|---|---|
| 1.0 | 4/17/2014 | Initial |
| 1.1 | 5/6/2014 | Added Interface Descriptions |

This document is a high-level technical description of the Maternity Tracker MCC Dashboard system. It is meant to familiarize the reader with the technical details of the design, architecture, and functionality of the system for the purposes of implementation and integration into the VA systems.

## TABLE OF CONTENTS

## GENERAL SYSTEM OVERVIEW

The Maternity Tracker is a set of applications and enhancements which facilitate tracking of maternal care in the perinatal period. The purpose of the system is to provide the Maternity Care Coordinator (MCC) and other key providers with the tools necessary to ensure that the veteran receives optimal care during pregnancy. The system also collects relevant data that is necessary for program tracking and analysis. The system consists of several components including, enhancements to VistA and CPRS, a web-based dashboard application, and a patient portal prototype.

## COMPONENTS OF THE SYSTEM

There are 4 main components of the maternity tracker system: the pregnancy and confirmation prototype, the MCC dashboard, the MCCD exchange document, and the patient portal.

This document concerns one part of the maternity tracker solution, the MCC dashboard. The MCC dashboard is a browser-based solution that allows an MCC, or other key provider, to track the care of a pregnant veteran during the perinatal period. The primary method employed to achieve this is to allow creation and management of a small list of patients and to show maternity-specific data for those patients. The dashboard is used for the following purposes:

1. To view and enter information about the current pregnancy.
2. To view and enter information about past pregnancies.
3. To view and enter other medical information relevant to pregnancies.
4. To document the MCC contact with the patient.
5. To view and enter basic delivery and newborn information.
6. To generate and import CDA exchange documents.
7. To track and assist with the provisioning of patient education materials
8. To ensure that lab tests, ultrasounds, and consults are ordered and completed.
9. To help track patient enrollment in community maternity programs such as TXT4BABY, birthing classes, and other pregnancy groups and classes.
10. To assist in the tracking of pregnancy outcomes by collecting and displaying data in-aggregate about the current and past patients in the system.

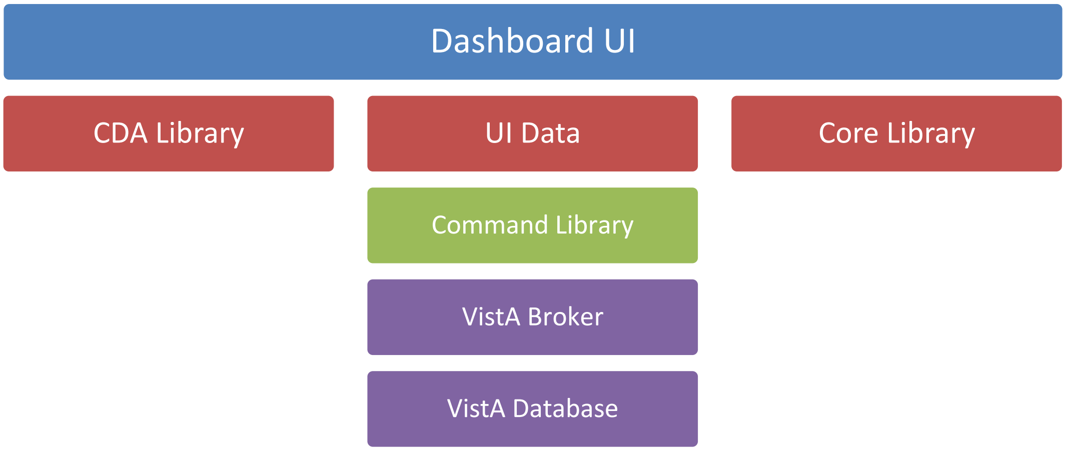
## DEVELOPMENT TOOLS AND LANGUAGES

The MCC Dashboard has been created with the following tools, components, languages and technologies:

1. Microsoft Visual Studio 2012 – Primary Development Environment
2. Microsoft .NET Framework 4.5 – Framework Version
3. Microsoft ASP.NET MVC – Project/Application Type
4. C# - Primary language
5. C# Razor Syntax – Server Side UI processing

6. Twitter Bootstrap version 3.0 – User Interface Styling and Functionality
7. JQuery 1.9.1 – Client-Side Scripting
8. HTML 5, CSS 3,  JavaScript – Client-Side UI

## SYSTEM DESIGN AND ARCHITECTURE



The solution is broken into the following components:

1. Dashboard UI – ASP.NET MVC project which handles all navigation, presentation and UI logic. This application utilizes an application design pattern called Model View Controller (MVC)
2. UI Data - .NET Library consisting of data objects used by the UI and data access code for interfacing with the data layer. This library utilizes the "Repository" design pattern.
3. Command Library – A .NET library which provides access to VistA data through objects called "commands"
4. Vista Broker – A .NET library which provides core functionality for accessing a VistA database.  This project/library includes a "broker" which sends messages to and retrieves data from VistA.  Also, any objects which are used system-wide related to Vista are placed here.
5. Core – A .NET library which provides core functionality (not related to VistA) for the solution.  This library contains basic functionality for the entire solution such as logging, tracing, and error handling.
6. CDA Library – A .NET library which provides object related to the creation and processing of CDA documents.  This library is used to simplify the CDA document exchange process within the dashboard.

## INTERFACE DESCRIPTIONS

**Direct Messaging** – The system will utilize the VA developed Direct Messaging application to send and receive secure messages with non-VA providers and facilities.  The dashboard will *contain a link to the direct messaging application. (e.g. http://direct.va.gov)*

**Secure Messaging** – The system will utilize the VA developed Secure Messaging API to allow direct communication between the MCC and the patient. At the time of this writing this interface has not been developed. It is anticipated to make web service calls to the secure messaging server for the following functions:

1. Retrieve messages sent to the MCC from a patient.
2. Send a message to a MHV patient.

**VistA** – An interface is being developed in Vista to provide data for the dashboard. This interface is in development. It is anticipated that the interface will provide the following classes of data to the dashboard:

1. General patient demographic data
2. Maternity specific clinical data
3. Maternity tracking information including which patients are flagged and tracked.
4. General clinical data for CDA generation
5. Laboratory Orders and Result
6. TIU Notes
7. Alerts and Clinical Reminders
8. Orders, Consults, and Ultrasounds

**Dashboard Web Services** – Web services will be developed which will provide data to the patient portal related to maternity care. The following classes of data are anticipated:

1. Patient Education Materials
2. Patient Pregnancy Data
3. Patient-entered pregnancy data
4. Patient enrollments (pregnancy classes, TXT4BABY, etc.)

## EXTERNAL DEPENDENCIES

This section includes a general discussion of the external dependencies of the MCC Dashboard. Release notes and a separate System Requirements document will detail exact version dependencies where there is one.

The MCC dashboard acts as a consolidator of system to simplify the usage of existing technology from within the dashboard.

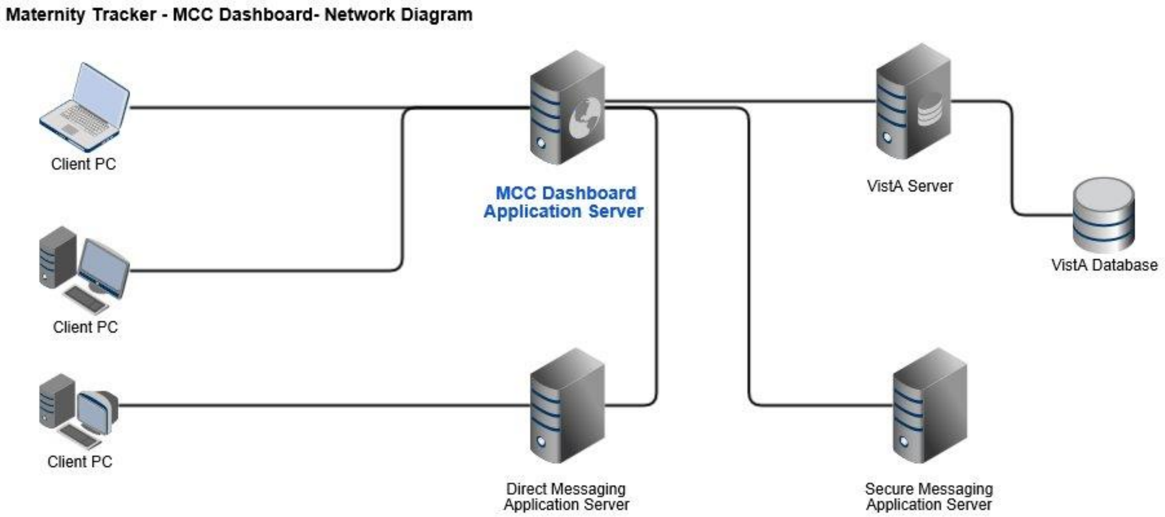
- **VistA** – The bulk of the data for the system resides in a VistA database. The dashboard requires a server name and port number to access VistA data.
- **Maternity Tracker VistA Updates** – The maternity tracker utilizes the **DSIO** namespace within VistA. These updates provide the functionality for the dashboard to work. A version of the dashboard will be dependent upon a specific version (KID build) of the DSIO Vista updates.
- **CPRS version 29+** – While the dashboard does not integrate directly with CPRS, some of the features of the dashboard depend on CPRS enhancements and features.

- **Direct Messaging** – The MCC dashboard generates, imports, and processes CDA documents for exchange, but it does not implement the exchange itself. Instead it relies upon an existing Direct Messaging application to exchange documents and messages with Non-VA providers and organizations. The exact link to the messaging application can be set in the application configuration file.
- **Secure Messaging** – The MCC dashboard provides a subset of the secure messaging functionality using the newly developed (at the time of this writing) API.

## SYSTEM REQUIREMENTS

- **Windows Server 2008 R2 –** Web Server
- **Current Browser** – Internet Explorer 9+, Google Chrome, Mozilla Firefox

## NETWORK DIAGRAM


Maternity Tracker - MCC Dashboard- Network Diagram
Client PC
Client PC
Client PC
MCC Dashboard Application Server
VistA Server
VistA Database
Direct Messaging Application Server
Secure Messaging Application Server

## PERFORMANCE

The performance of the MCC dashboard is expected to be dependent on a few factors.

1. *Vista Performance* – All of the data related functionality of the dashboard involves remote procedure calls (RPC's) to VistA. There is very little processing occurring on the data returned from the RPC's. Because of this VistA performance will be critical in a well-performing dashboard UI.
2. *Network Latency (Server to Server)* – Ideally, the MCC dashboard will be installed on a server which is collocated with the VistA database such that latency can be reduced to a minimum. The dashboard would be expected to have reduced response times and slower performance if the network infrastructure is overly taxed or at maximum capacity.

3. *Network Latency (Client to Server)* – Because the dashboard application does not run locally on a client pc, the network performance between the client and the application server can be a significant limiting factor regarding performance of the dashboard. Much of the application operations will be processed on the server.
4. *Server Performance* – The performance of the server will affect the performance of the dashboard. If the dashboard is installed onto a dedicated server with current hardware, the server is not expected to contribute to performance problems. If, however, the application is installed onto an existing server with other applications running on it, then those applications may end up taking system resources and slowing down the dashboard.

## SECURITY

The MCC dashboard is an intranet application that is not accessible to the internet at-large. Because of this it inherits some of the security features that are built into the VA intranet itself. Security has been built into the application which prevents any unauthorized user from accessing the system.

ASP.NET Security Features

ASP.NET is a widely used web application development platform which contains security features built into it. Several of those built-in features have been implemented in this project.

- **HTTPS** – The MCC Dashboard utilizes encrypted communications via the HTTPS protocol. This means that all data is encrypted as it flows from the web-server to the client browser.
- **Authorize Attribute** – Each of the web pages or actions that the user accesses via the application contain an “Authorize” attribute which indicates that only an authorized user may access the page or resource. If an anonymous user attempts to access one of these pages or resources, then the user will automatically be redirected to a login page where they must enter VistA credentials.
- **Session Timeout** – When a user logs into the system using their VistA credentials, the system creates a “session” for the user. This session has a timeout value, which is configurable, that automatically terminates the session on expiration. This can help prevent unauthorized access in the event that a user logs into the dashboard and then walks away from their workstation or device.

VistA Security Features

- The MCC dashboard utilizes VistA security to prevent unauthorized access to the data. The user must provide, before accessing any VistA data, an access and verify code. These credentials will identify the user on the system and restrict access to the data just as it would through CPRS or a roll-and-scroll interface.
- The MCC dashboard contains an application context which much be added to a user’s menu context before they may utilize the MCC dashboard. This provides an additional layer of security where a VistA user does not have access to the dashboard by default; but must be given this access by the appropriate security personnel.

## SYSTEM SETUP AND CONFIGURATION

These are the required steps to install the MCC Dashboard on a Windows 2008 R2 server:

1. Add IIS Role using server configuration
    a. Select ASP.NET option.
2. Install the Web Platform Installer
3. Change Default App Pool in IIS configuration to .NET 4.0
4. Run asp_net_regiis.exe –I to install and register the asp.net 4.0 version of the application pool.
5. Install .NET Framework 4.5 using Web Platform Installer
6. Copy installation files to any local folder on the server.
7. Install the dashboard application by running the following command from an administrative command prompt:
    a. va.gov.artemis.deploy.cmd /T (Tests the installation)
    b. Verify that no errors are found
    c. Va.gov.artemis.deploy.cmd /Y (Performs the installation)
    d. Verify that no errors are found
8. Create the \Content\CDA folder in the default installation folder (if it does not exist)
    a. The default will be c:\inetpub\wwwroot\artemis\content\cda
    b. Give read/write permission to this folder to the IIS_IUSRS group

Some of the dashboard configuration is handled by editing the “web.config” file located in the root of the installation folder. The following is a description of the settings which are set up there:

1. VistA Server Settings
    a. vistaServer – The name of the vista server name
    b. vistaListeningPort – The port number
2. Home Page Settings
    a. WelcomeMessage – The exact text of the welcome message
    b. ProductDescription – Text of the product description
    c. AccessHelpLink – A link pointing to help getting access to the application
3. AppContext – Default is DSIO GUI CONTEXT
4. CDA Settings
    a. cdaManufacturerModelName – The manufacturer of the software
    b. cdaSoftwareName – The name of the software
    c. cdaProviderOrganizationPhone – The central phone number for the local VA hospital

## TERMINOLOGY, DEFINITIONS, ACRONYMS

The **Maternity Care Coordinator (MCC)** is a newly defined role within the VA focused on coordinating the care of pregnant veterans; especially those who receive care from a Non-VA provider or facility.

The **MCC Dashboard** is the software application that is the primary focus of this document.

A **tracked patient** is a patient that is included in the primary list of patients in the MCC dashboard.

**CDA** stands for Consolidated Document Architecture. This is an HL7 standard for representing and exchanging clinical data. (HL7 Website)

**MVC** stands for Model-View-Controller. This is an industry standard design pattern which specifies how applications are to be organized and coded. The Microsoft implementation of MVC has been used in this case. (MVC on Wikipedia)

**VistA** – Veterans health Information Systems and Technology Architecture (Vista Wikipedia)

**CPRS** – Computerized Patient Record System

**Direct Messaging** – A simple, secure, scalable, standards-based way for participants to send authenticated, encrypted health information directly to known, trusted recipients over the internet. (Direct Project)

**Secure Messaging** – A web-based message system that allows participating VA veterans and VA health care teams to bi-directionally communicate non-urgent, health-related information in a private and safe computer environment.

**ASP.NET** – Microsoft web-application development platform. (ASP.NET MVC)

**HTTPS –** Encrypted version of the ubiquitous Hypertext Transfer Protocol. (more information)