

**System Architecture**

Department of Veterans Affairs,

Pre-Procedure Checklist Tool

VA118-11-RQ-0508

**Volume I**

**Intellica Corporation**

**209 W. Poplar**

**San Antonio, Texas 78212**

**12/05/2011**

# Table of Contents

[Table of Contents 2](#_Toc320518513)

[Table of Figures 3](#_Toc320518514)

[Change Log 4](#_Toc320518515)

[Introduction 4](#_Toc320518516)

[System Overview 5](#_Toc320518517)

[System Architecture Overview 7](#_Toc320518518)

[Overall System Architecture 16](#_Toc320518519)

[Software Development Components 17](#_Toc320518520)

# Table of Figures

[Figure 1 – Overall System Composition 7](#_Toc320518521)

[Figure 2 – Client Tier 8](#_Toc320518522)

[Figure 3 – Application Web Server 8](#_Toc320518523)

[Figure 4 – Web Application Views 9](#_Toc320518524)

[Figure 5 – Web Application Data Access 10](#_Toc320518525)

[Figure 6 – Logic Module 11](#_Toc320518526)

[Figure 7 – Pre-Procedure Checklist Database 12](#_Toc320518527)

[Figure 8 – Stored Procedures 13](#_Toc320518528)

[Figure 9 – Accessing the Decision Support Windows Service 14](#_Toc320518529)

[Figure 10 – Decision Support Web Service 15](#_Toc320518530)

[Figure 11 – Overall System Architecture 16](#_Toc320518531)

[Figure 12 – VAPPCT.DA Class Library Project 17](#_Toc320518532)

[Figure 13 - VAPPCT.UI Class Library Project 17](#_Toc320518533)

[Figure 14 - VAPPCT Web Application 18](#_Toc320518534)

[Figure 15 - Decision Support Windows Service Project 19](#_Toc320518535)

# Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version # | Author | Revision Description |
| 12/05/2011 | 1 | Craig Rebo | Created |
| 12/07/2011 | 2 | Craig Rebo | Added Web App, Class Libraries and Web Service. |
| 03/26/2012 | 3 | Craig Rebo | Removed Decision Support Web Service and added a windows service in its place, updated architecture diagrams. |
| 07/16/2012 | 4 | Craig Rebo | Updated Communicator Service description and added VAPPCT.Data class library |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

This document describes the overall system architecture for the Pre-Procedure checklist tool.

# System Overview

The mission of the Department of Veterans Affairs (VA), Office of Information & Technology (OI&T), San Francisco Veterans Affairs Medical Center is to provide benefits and services to veterans of the United States. In meeting these goals, OI&T strives to provide high quality, effective, and efficient Information Technology (IT) services to those responsible for providing care to the Veterans at the point-of-care as well as throughout all the points of the veterans’ health care in an effective, timely and compassionate manner. VA depends on Information Management/Information Technology (IM / IT) systems to meet mission goals.

The VHA Innovation Program is a cooperative effort between the Health and Medical Information Office (CHIO / HMIO) of VHA’s Office of Health Information (OHI) with VA’s Office of Information and Technology (OI&T) to catalyze and support VA’s next generation of Health Information Technology (HIT). VHA Innovations Program Goals are to provide tools, resources and processes that:

1. Encourage innovations to emerge from the field
2. Allow innovations to evolve based on collaborative constructive review
3. Provide a safe harbor for innovations to be piloted
4. Vet Innovations from business and technical perspective

The Pre-Procedure prototype application is a multi-user ASP.NET Web application that contains all the functionality to communicate with CPRS (Computerized Patient Record System) / VistA through the Medical Domain Web Services (MDWS) interface and provides the following functionality:

1. Authenticate users and provide user roles
2. Allow for the entry of checklist items
3. Create checklist templates
4. Display single and multi-patient views based on selected patient attributes
5. Edit checklist and provide override capability
6. Allow for the generation of the documentation of the current checklist
7. Provide for a comprehensive audit trail
8. Retrieve data such as patient demographics, problem lists and scheduled procedures
9. Allow the user to use standard Boolean logic for items in the checklist

Important quality attributes include security, usability, availability and performance. The ability to load the single patient view in under 10 seconds and load the multi-patient view in less than 20 seconds is critical.

# ****System Architecture Overview****

The Pre-Procedure checklist prototype is an ASP.NET Web application that saves and retrieves Pre-Procedure checklist data to and from an Oracle database. The application also communicates with a Decision Support Windows Service to save and retrieve data to and from VistA.

## ****System Composition****

The system architecture consists of the following:

1. A Web Browser capable of navigating and displaying ASP.NET pages and running client side javascript.
2. A Microsoft Windows 2008 Server running Internet Information Server (IIS) and ASP.NET. This is the application server that hosts the Web application.
3. A Microsoft Windows 2008 Oracle database server that holds all Pre-Procedure checklist data. This server hosts the Decision Support Windows service that is responsible for accessing VistA interfaces and updating checklist data at specified intervals.

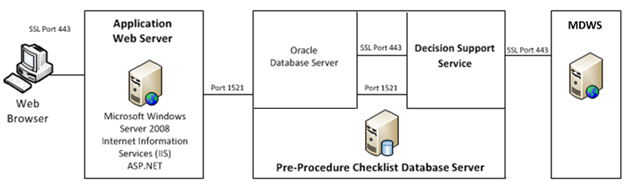


Figure 1 – Overall System Composition

## ****Sources of System Components****

The Pre-Procedure checklist is a combination of Government-Furnished-Equipment (GFE), Commercial-Off-The-Shelf (COTS) components, Government-Off-The-Shelf (GOTS) components and code developed from scratch.

### ****Architecture of the Client Tier****

Users access the Pre-Procedure checklist application using a thin client Web browser. The Web browser must be capable of navigating and displaying ASP .NET pages and running client side javascript.

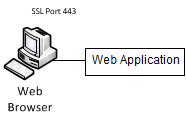


Figure 2 – Client Tier

### ****Architecture of the Application Web Server****

The application Web server is responsible for serving dynamic ASP.NET content to the users Web browser. Access to the Web server is through SSL Port 443. Users will use MDWS credentials to Authenticate to the Web server. The Web server operating system is Microsoft Windows 2008 Server with IIS and ASP.NET.

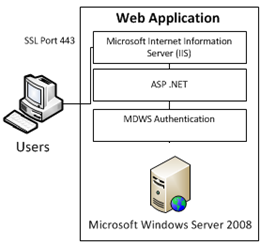


Figure 3 – Application Web Server

### Application Server Components

#### ASPX View pages

The application uses ASP.NET master pages to create a consistent look and feel across the application. In addition to this, all of our master pages derive from a single base master page. This base page contains common functions that are used across all pages. It also acts as a central controller for security, auditing and page flow. A developer creates a View by developing the user interface in an ASPX page. The Controller functions are implemented in the code-behind file for the ASPX page (.aspx.cs).

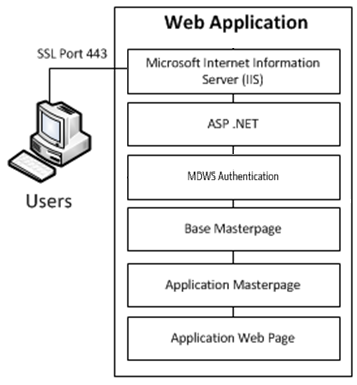


Figure 4 – Web Application Views

#### Data Access

The application accesses data through a set of Application Data Access Classes developed in C#. These classes will use the General Data Access Classes to insert, update and retrieve data to and from the Oracle database. The General Data Access components contain tools for connecting to Oracle. They also contain tools for passing parameters to Oracle stored procedures and retrieving .NET DataSets. ALL data will be accessed via Oracle stored procedures through the Data Access layer. This will simplify auditing and centralized data access business rules as well as connection to data sources. The General Data Access Classes access the Oracle database using the System.Data.OracleClient library through port 1521.

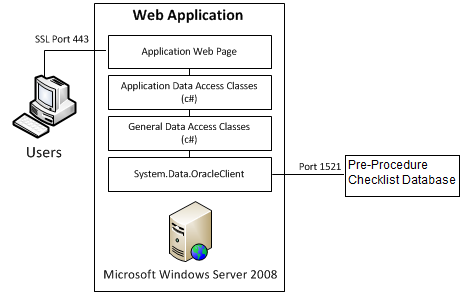


Figure 5 – Web Application Data Access

#### Logic Module

The component that parses the logic for the logic module will reside on the Web server in the form of a C# class library. The logic Module allows the user to use standard Boolean logic for items in the checklist.

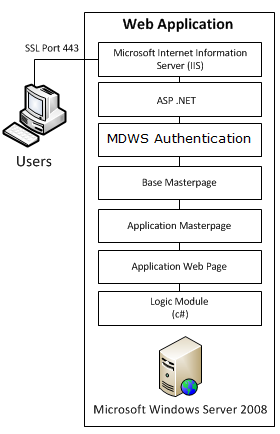


Figure 6 – Logic Module

### ****Architecture of the Pre-Procedure Checklist Database Server****

The Pre-Procedure Checklist Database server is responsible for storing all Pre-Procedure related data. The database server operating System is Windows Server 2008. As stated above, ALL data will be accessed via Oracle stored procedures through the Data Access layer.

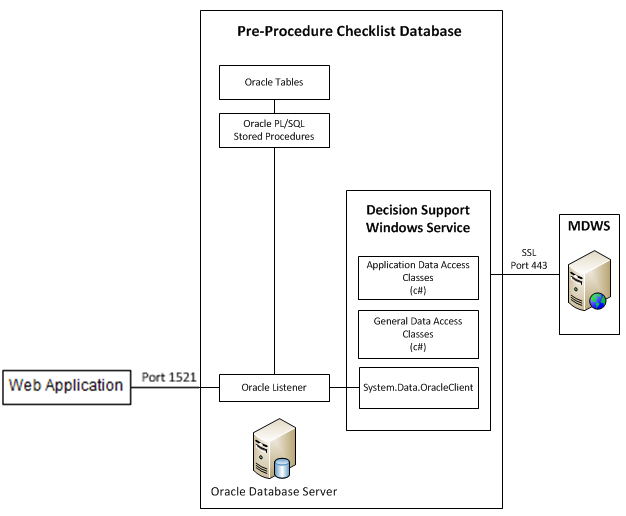


Figure 7 – Pre-Procedure Checklist Database

### Pre-Procedure Database Server Components

#### Stored Procedures

Each functional area of the application has an associated Oracle Stored procedure package containing procedures and functions used to store and retrieve data. The PCK\_COMMON package contains general functions and procedures used by all other packages to save and retrieve data and status information.

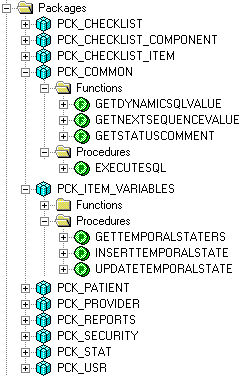


Figure 8 – Stored Procedures

#### Scheduled Tasks

Scheduled tasks, such as updating data in the Pre-Procedure checklist database or backing up audit data, are implemented using Oracle Jobs that call stored procedures.

### ****Architecture of the Decision Support Server****

The Decision Support Windows Service uses the VAPPCT.Data layer to access MDWS. This allows the Service to use the same calls that the Web application uses to store MDWS data and update checklist state. This service is installed on the Web Server and runs at specified intervals to retrieve results from MDWS and update checklist state.

# 

Figure 10 – Decision Support Web Service

# Overall System Architecture

The following is the overall System Architecture for the Pre-Procedure checklist prototype. The ASP.NET Web application communicates with the MDWS and the database to store and retrieve data. The database communicates with the Decision Support Windows service to access VISTA.

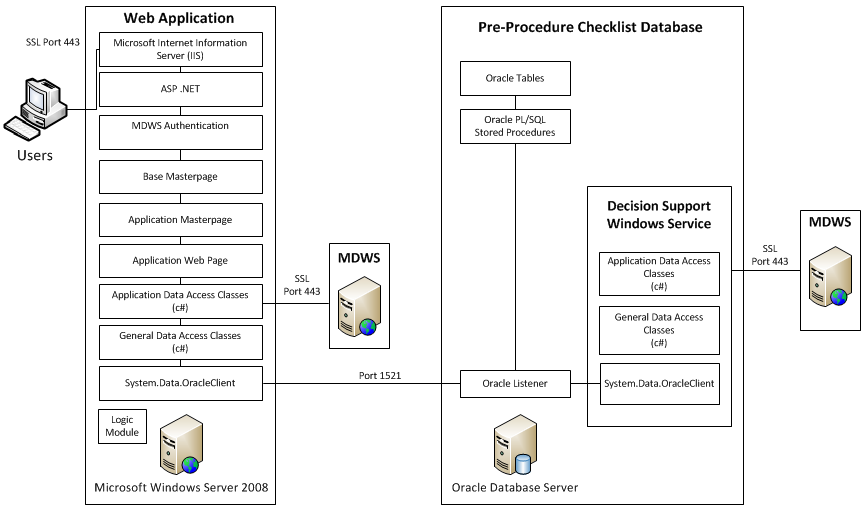


Figure 11 – Overall System Architecture

# Software Development Components

## C# Class Libraries

VAPPCT.DA – general data access utilities for connecting to Oracle, executing stored procedures and retrieving datasets. This class library will be used by both the web app and the decision support web service.

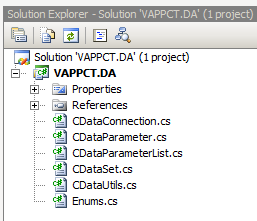


Figure 12 – VAPPCT.DA Class Library Project

VAPPCT.UI – general user interface utilities used from the Web application.

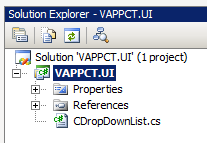
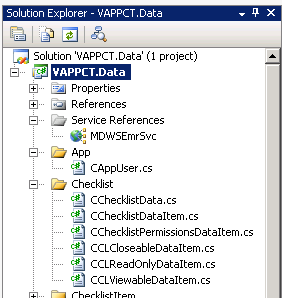


Figure 13 - VAPPCT.UI Class Library Project

VAPPCT.Data – Data layer for the application. This library is used by the Web Application and the Decision Support Windows Service to access VAPPCT and MDWS data.



## ASP .NET Web Application

VAPPCT – Web application, application and data classes are separated, all master pages derive from our controller class CBaseMaster. VAPPCT.UI, VAPPCT.DA and VAPPCT.Data class libraries are linked in. CAppDBConn derives from CDataConnection class. Error pages are stored in a sub directory.

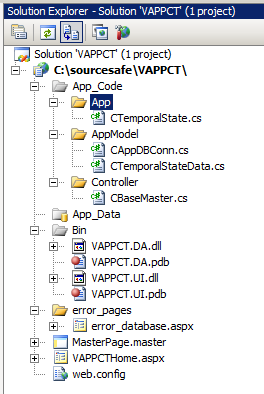


Figure 14 - VAPPCT Web Application

## Decision Support Windows Service

VAPPCTComm – Windows service for communicating with MDWS. VAPPCT.DA and VAPPCT.Data class libraries will are linked in. This service is installed on the Web Server and runs at specified intervals to retrieve results from MDWS and update checklist state.

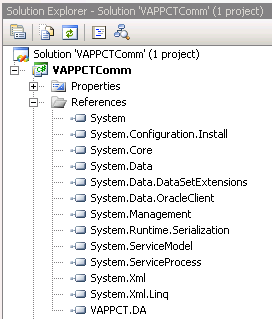


Figure 15 - Decision Support Windows Service Project