

Chemotherapy Ordering Management System (COMS)

Technical Manual

Version 1.2



September 2012

Department of Veterans Affairs

Revision History

[illegible]

Table of Contents

Introduction.....	1
History.....	1
Application Overview	1
Architecture.....	4
Model-View-Controller (MVC) Framework	4
Ext JS.....	4
General Note.....	4
Ext JS Model-View-Controller-(MVC) Software Design Pattern	4
Standard Model Template	5
Dependent Model Template	7
Source Model Template.....	8
Standard Store Template	8
Standard View Template	9
Standard Controller Template	9
Web Services.....	10
COMS Secure Shell	14
Medical Domain Web Services.....	15
VA FileMan.....	15
Databases.....	15
Service Calls – General Information	15
Application Breakdown.....	16
Patient Tab	17
Patient Selection	17
Patient Information	19
Chemotherapy Template Order Source.....	26
Order Entry Management.....	28
Nursing Documentation.....	33
Orders	44
Template Authoring.....	45
Messages.....	50
Admin	52

List of Figures

1	Patient Selection	17
2	Confirmation of Returned Patient.....	19
3	Patient Information for Service Calls.....	20
4	Chemotherapy Template Order Source Display.....	26
5	Order Entry Management Display.....	28
6	Nursing Documentation Header Display.....	31

7	ND Module, General Information Panel.....	32
8	ND Module, Assessment Panel.....	34
9	ND Module, IV Site Panel.....	36
10	ND Module, Treatment Panel.....	38
11	ND Module, Infusion Reactions Panel.....	40
12	ND Module, Discharge Instructions Panel.....	42
13	Orders Tab.....	44
14	Template Authoring.....	46
15	Add New Drug Regimen.....	50
16	Messages Tab.....	51
17	Admin Tab.....	52

Introduction

The Chemotherapy Ordering Management System (COMS) application enhances the clinical environment and safety for oncology patients through development and implementation of an automated ordering and management process available within the Veterans Health Administration (VHA) clinical practice setting. The application satisfies the unique needs of chemotherapy ordering and standardizes capabilities to meet direct entry of chemotherapy orders consistent with oncology practice. COMS interfaces and interacts with existing applicable VHA health care systems, modules and processes within Computerized Patient Record System (CPRS) and Veterans Health Information Systems and Technology Architecture (VistA) databases. COMS is a web-based application consisting of Hypertext Precursor (PHP), Java Script, Simple Object Access Protocol (SOAP), and Representational state transfer (REST) web services.

History

The VHA has one of the largest cancer populations in the country; it is also the fastest growing group of patients within the VHA. A uniquely high-risk and high-complexity domain of health care, Oncology has not been effectively implemented within the existing VHA Electronic Health Record primarily due to the lack of functionality required for the specialty. This creates a clinical environment with minimal standardization and limited direct order entry of chemotherapy. VHA's oncology processes are a mix of paper-based and computer-based practices, presenting potential error, adverse events, and inefficiencies. For these reasons, the VHA Office of Health Information (OHI) Patient Safety Workgroup rated this issue as having a high level of patient safety risk. Accordingly, an initiative within VHA's Innovations Program sought to enhance the clinical environment and safety for oncology patients through development of the COMS application as part of VHA's Strategic Incubation.

VHA provides oncology services at a multitude of different locations by integrated oncology care teams consisting of, but not limited to, a physician, pharmacist, and nurse. Teams typically provide care on an outpatient basis, although some patients may require hospitalization. The COMS application supports oncology healthcare teams in ordering, preparing, and documenting the administration of chemotherapy. It consists of a general application interface and several modules – Chemotherapy Template Order Source, Order Entry Management, Flow Sheet, Nursing Documentation and End of Treatment Summary – that serve a specific purpose and provide functionality to support users in executing their roles and responsibilities in the treatment process. COMS also offers exportability of chemotherapy templates to further facilitate VHA-wide standardization of chemotherapy regimens.

Application Overview

COMS accommodate local facility policies for clinical preferences/processes and implements several VA defined system-wide protocols. These protocols include a national set of chemotherapy order templates, standardization of the calculation method for medication dosage and dose rounding, and a standard documentation format for chemotherapy treatment plans, administration, and summaries. COMS creates and manages chemotherapy templates; clears and places medication orders in VistA; enables nursing documentation; displays a temporal flow sheet for relevant clinical data, medications administered, and user assessment of response to treatment; and creates a treatment summary for the current healthcare team, referring/primary care providers, and other clinical and support staff. The COMS application

fulfills legal/professional requirements, fosters Joint Commission compliance and enhances patient safety with documentation ultimately stored in VA's electronic health record.

This manual instructs users on COMS application functionality, navigation of the application, and effective use of COMS to record and reflect treatment provided to oncology patients. COMS consists of five clinical modules: Chemotherapy Template Order Source (CTOS), Order Entry Management (OEM), Nursing Documentation (ND), Flow Sheet (FS), and End of Treatment Summary (EoTS) plus miscellaneous functionality.

The CTOS module permits the oncology provider to download a chemotherapy regimen template from a central library and modify it for local use. It also enables authorized users to create original, new templates or create a template from an existing one. The CTOS module affords flexibility to enable the user to assign a user-friendly template name. These templates may be applied to a particular patient record to generate an order sheet for the regimen and used throughout the remaining modules of the COMS application. During the process of applying a template to a patient, the oncology provider has the opportunity to identify the effective date; select the body weight and body surface area formula to use for medication dosage calculations; categorize the regimen for the patient as curative or palliative care; indicate whether the patient/regimen are part of a clinical trial and specify the name of the clinical trial, if applicable; identify patient amputation(s); and document the current performance status of the patient.

The OEM module permits the oncology provider to prescribe or modify an order for pre-therapy, therapy, and post-therapy medications from any template currently applied to a specific patient. Oncology providers utilize the OEM module to tailor templates – currently applied to patients based on diagnosis and other considerations – as a prescription for an appropriate course of therapy to meet chemotherapy treatment goals. For any future administration date within the prescribed regimen, oncology providers may edit medication dosages after the initial order and change the administration date. The OEM module also facilitates and documents communication among the oncology provider, pharmacist, and nurse as it relates to modifying the patient's medication order and any subsequently required provider approval for changes to the original order while it progresses from "ordered" through "administered" status. Altogether, the OEM module enables oncology providers to tailor pre-therapy, therapy, and post-therapy medications for each individual patient and facilitates treatment plan communication and coordination among the healthcare team.

The ND module supports oncology nursing with six activity-specific panels to convey information previously obtained and to facilitate the documentation of new information relevant to medication administration and oncology care. The nurse may view relevant historic and current clinical data, record the provision of chemotherapy, and document nursing assessments of the patient since the previous administration and throughout the current treatment. The ND module provides the capability to annotate overall regimen and administration day specific information, including vital signs, symptom assessments and infusion reactions. It also facilitates documenting the administration of the regimen's prescribed medications with functionality to verify and modify pre-populated data from the OEM module for medication administered. The ND module enables the nurse to document administration dates and start/stop times, verify patient identity and medication dosing, and annotate discharge instructions provided to the patient. Relevant entries within the ND module are automatically populated into the Flow Sheet and may be included in the End of Treatment Summary.

The FS module offers a snapshot of care with an efficient display of relevant information and patient-centered documentation of chemotherapy administration. The healthcare team may view relevant clinical data, the disease response/patient reaction to the chemotherapy, pertinent laboratory results, and an overview of administered medications. Organized in administration day columns, the FS module displays pre-therapy, therapy, and post-therapy information for medications and dosages administered to the

specific patient. The COMS application eliminates dual entry by retrieving this documentation from the ND module. The FS module also supports direct entry of disease response, toxicity/side effects, and other annotations with free text comments for any administration day within the regimen. The healthcare team may view the FS module throughout the regimen and the provider is presented relevant FS module information while generating the End of Treatment Summary.

The capstone module of the COMS prototype is the End of Treatment Summary. The EoTS module supports the oncology provider creation, and healthcare team viewing, of the summary of care rendered and results achieved throughout the specified treatment regimen. The EoTS module also enables the provider to stop a treatment regimen. Following the conclusion or discontinuation of a regimen and its applied template, the provider typically generates the treatment summary. To aid in generation of the summary report, the EoTS module retrieves relevant information from various COMS modules and pre-populates several sections of the treatment summary for provider consideration when preparing the narrative of disease response and toxicity side effects in the provider report section. It also supports an overall provider report with free text entry to communicate patient and regimen specific assessment to the current and future healthcare team. The EoTS module enhances documentation and communication by facilitating the generation and viewing of oncology regimen treatment summaries for the current healthcare team, referring/primary care providers, and other clinical and support staff.

COMS miscellaneous functionality is the underpinning that supports the five clinical modules and general application performance. Not specifically associated with any particular module, COMS miscellaneous functionality is integral to overall application effectiveness. Through patient specific functionality, users are provided general and historic information from VistA and various COMS modules. General functionality enables coordination and monitoring of provider orders from the point of order, through clearing of the order, through finalizing and dispensing medications, to administration of the prescribed medications. Extensive administrative functionality enables COMS administrators to setup and maintain the application's database contents, accessibility, and interoperability while tailoring COMS to accommodate local facility policies for administrative and clinical preferences and processes. COMS miscellaneous functionality also facilitates the setup and configuration of the application.

Architecture

The COMS application is comprised of two separate but interconnected components. The first component is the backend or database component while the frontend component serves as the user interface. The database was developed in PHP and is responsible for all interaction with the Structured Query Language (SQL) Database, the Medical Domain Web Services (MDWS) system for the saving/retrieval of patient records, chemotherapy templates, medications, and pharmaceutical orders. The user interface was developed in JavaScript and provides all the controls, windows, and forms for user interaction to create, edit, and view the information necessary for COMS to function effectively and efficiently.

Model-View-Controller (MVC) Framework

Both the user interface and database components were developed using the Model-View-Controller (MVC) pattern. This is a software design pattern used to separate the presentation of information from the underlying architecture and business rules.

The “Model” portion of the MVC pattern defines the data and business rules. The “Controller” portion manages the input and converts it to instructions for either the “Model” or “View” portion. The “View” portion is the output representation or display of the data in terms the application user can understand, such as panels and tabs of information.

Ext JS

Ext JS is a JavaScript library and application framework developed by Sencha (<http://www.sencha.com/products/extjs/>) for use in designing robust user interface centric, web-based applications. For the COMS application the Ext JS library allowed permits division of the application into the MVC pattern with the panels, forms and tabs for the end user, defining various data models for communicating with a backend component and the controls necessary for the application to communicate with the backend database.

General Note

Any textual value available in a Drop Down control (also termed a “Combo Box” or simply “Combo”), for example the name of a medication, will be stored as the Globally Unique Identifier (GUID) for the record containing that textual value from the appropriate table. In this manner, if a textual value needs to change (e.g. due to a typographical error in the string), none of the saved records pointing to that textual value would need to change.

Ext JS MVC Software Design Pattern

- A **model** is an object representing data or business rule. Models persist through the data package and may link to other models through associations.
- A **view** serves as a window displaying the data. Grids, trees, and panels are all views. A **controller** serves as the “glue” between the model and the view, controlling the timing

and passing of models to views for application display. A controller contains all the programming code for rendering views, instantiating Models, and any other application logic.

- **Event handlers** should be included as part of a Controller for the timing and passing of models. Views do not contain event handlers; they only fire the events.
- **Hierarchy of controllers** determines the handling of information. Each controller contains dedicated views and models. When designing interaction between controllers, developers consider application wide events. For example, if the View in Controller1 fired an event to be handled by Controller2, the application is designed to for Controller1 to fire a generic application event and Controller2 to handle it for updating its views. In this instance of interaction between the two controllers, Controller1 has a higher/earlier hierarchy followed by Controller2
- **Validations** are contained within controller models to ensure the data is consistent with designed expectations. If the data is consistent, the model validation will pass and permit the controller to View the information for display. If the data is not consistent with designed expectations, the controller's model should invalidate the data and not provide a View for the user.
- **Model specific data manipulations** are similar to validations and are contained within the Controller's Model. Accordingly, the application manipulates the data according to designed specifications before the Controller permits a View for the user.
- **Data querying logic** must also reside within a model. For a model as an abstraction to the system that processes input routed by controllers and updates its state, it serves as a proxy class for the backend database system. All queries to the database should go through a model. Any business processes in the application should be formulated within a model only.

Standard Model Template

A standard model template is used when there are no dependencies for the model. Within COMS, an example of a standard model template is as follows:

```
Ext.define("COMS.model.<Name of Model>", {
    extend: "Ext.data.Model",
    alias: "model.<Name of Model>",
    fields: [
        "Field1",          // default field type is string, no name/type object required
        "Field2",
        { "name" : "Field3", "type" : boolean },
        ...
    ],
    "proxy": {
        "type": "rest",
        // The following parameters should be set to false, to ensure that no
        // additional parameters are passed to the service call
        "filterParam" : false,
        "groupParam" : false,
        "pageParam" : false,
        "startParam" : false,
        "limitParam" : false,
        // The following parameter prevents the "noCache" parameter
```

```

        // from being passed to the service call, however, the service call code
        // itself should always ensure that it sends current data back.
        // Set HTTP Headers to NoCache???
"noCache" : false,
    // Note, there is an additional parameter which might get passed
    // to the service call in the case of a model being used by a combo box.
    // This is the "query" parameter which must be set to false in the
    // ComboBox config itself, e.g. "queryParams" : false
api: {
    read: <READ URL>,           // uses HTTP GET
    update: <UPDATE URL>,       // uses HTTP PUT
    create: <CREATE URL>        // uses HTTP POST
},
"reader": {
    "type": "json",
    "root": "records",
    "successProperty": "success",
    "totalProperty": "total",
    "messageProperty": "message"
},

// Any functions which are specific to the proper retrieval/handling/validating of
// the data should be included in the model file itself.
"afterRequest" : function (request, success) {
    if ("read" === request.action) {
        this.readCallback(request);
    }
    else if ("create" === request.action) {
        this.createCallback(request);
    }
    else if ("update" === request.action) {
        this.updateCallback(request);
    }
    else if ("destroy" === request.action) {
        this.deleteCallback(request);
    }
},

"readCallback" : function (request) {
    if (!request.operation.success) {
        Ext.Msg.show({
            title: "Warning",
            msg: "Could not load Data. Please try again.",
            buttons: Ext.Msg.OK,
            icon: Ext.Msg.WARNING
        });
    }
}
}

```

```

    }
  });

```

Dependent Model Template

Conversely, a dependent model template is used when a given model contains an array of objects which are defined in a separate model. The following is an example of a dependent model template used within the COMS application:

```

Ext.define("COMS.model.<Name of Model>", {
  extend: "Ext.data.Model",
  alias: "model.<Name of Model>",
  // If this model is dependent on one or more additional models
  // then include the "uses" array
  uses: [
    "COMS.model.<Dependent Model Name>"
  ],

  fields: [
    "Field1",      // default field type is string, no name/type object required
    "Field2",      // This is an array of "<Dependent Model Name>" objects
    { "name": "Field3", "type": boolean },
    ...
  ],

  // The "hasMany" property defines which field(s) is/are dependent
  // on which additional models
  hasMany: {
    model: "COMS.model.<Dependent Model Name>",
    name: "Field2"
  }
  "proxy": {
    "type": "rest",
    api: {
      read: <READ URL>,           // uses HTTP GET
      update: <UPDATE URL>,       // uses HTTP PUT
      create: <CREATE URL>        // uses HTTP POST
    },
    "reader": {
      "type": "json",
      "root": "records",
      "successProperty": "success",
      "totalProperty": "total",
      "messageProperty": "message"
    }
  }
});

```

Source Model Template

A source model template is used when a given model is used by a dependent model for an array of objects defined in a separate model. Model specific functions should be included in the Model file. Within the COMS application, an example of a source model template is as follows:

```
Ext.define("COMS.model.<Name of Model>", {
    extend: "Ext.data.Model",
    alias: "model.<Name of Model>",
    // If this model is related to one or more additional models
    // then include the "uses" array
    uses: [
        "COMS.model.<Name of Model which is dependent>"
    ],
    fields: [
        "Field1"
    ],

    // The " belongsTo" property defines which model(s) are dependent upon this one
    belongsTo: {
        model: "COMS.model.< Name of Model which is dependent >"
    }
    "proxy": {
        "type": "rest",
        api: {
            read: <READ URL>,           // uses HTTP GET
            update: <UPDATE URL>,       // uses HTTP PUT
            create: <CREATE URL>        // uses HTTP POST
        },
        "reader": {
            "type": "json",
            "root": "records",
            "successProperty": "success",
            "totalProperty": "total",
            "messageProperty": "message"
        }
    }
});
```

Standard Store Template

The COMS application also utilizes a standard store template. In this section the standard store template technical aspects and utility are discussed.

```
Ext.define('COMS.store. <Name of Store>', {
    extend : 'Ext.data.Store',
```

```

        model : <Name of Model>
    });

```

Note: The “autoLoad” option should not be used. Additional options are declared after the model.

Simplified method to access a data store by name

The COMS application accesses a data store by name through its database “get” functionality in the backend component.

If a store is included in the list of stores for a controller:

```

stores : ["Store1", "Store2"])

```

Then simply reference the store by name via the controller’s “get” function:

```

this.get<StoreName>Store();

```

For example:

```

this.getStore1Store();
this.getStore2Store();

```

However, this will only work if the store is declared in the stores list. For example given the above stores list one can call *this.getStore1Store()*, *this.getStore2Store()* but a user may not call *this.getStore3Store()* until declared in the stores list.

Standard View Template

The COMS application utilizes a standard view template in the frontend/user interface component.

```

Ext.define("COMS.view.<Application Section Folder Name>.<View Name>", {
    "extend" : "Ext.tab.Panel",
    "alias" : "widget.<View Name>",
    "name" : "<View Name>",
    "autoEl" : { tag : "nav" },
    "items" : [
        { "xtype" : "AddLookups", "title" : "Add LookUps" },
        { "xtype" : "container", "html" : <HTML Content> }
    ]
});

```

Views are designed and accomplished through individual JavaScript (JS) files in a folder descriptive of the application section where they are located (e.g. Patient Tab, Orders Tab, and Template Authoring Tab). Each view will contain a unique Alias and Name and will utilize Hypertext Markup Language (HTML) 5 tags (e.g. section, header, footer, navigation, etc) to promote better document structure. A complete list of standard view templates is available at

http://www.w3schools.com/html/html5_new_elements.asp .

Standard Controller Template

The COMS application also uses standard controller templates. Similar to the views, controllers are designed and achieved through individual JS files in a folder descriptive of the application section where

they located in (e.g. Patient Tab, Orders Tab, and Template Authoring Tab). An example of standard controller templates within the application is as follows:

```
Ext.define("COMS.controller. <Application Section Folder Name>.<View Name>", {
    "extend": "Ext.app.Controller",
        // Include any stores, views and Data Models managed by this controller
    "stores": [ "LookupStore" ],
    "views": [ "Management.AdminTab" ],
    "models": ["LookupTable"],

        // Include any local references needed by this controller
        // a local reference can be accessed via the "get" function of the controller
        // e.g. this.getLookup()
    "refs": [
        { "ref": "<Ref Name>", "selector": "<Selection Query>" },
    ],

    "init" : function() {
        // Assign any event handlers to specific objects and events
        this.control({
            "<Selection Query>" : {
                select : this.<Local Function Name>
            },
        });
    },

        // define the functions used in this controller to manage the specific events.
    "<Local Function Name>" : function(<Parameters>){
    }
})
```

Web Services

The World Wide Web Consortium (W3C, located at <http://www.w3.org/>) defines a “Web service” as “a software system designed to support interoperable machine-to-machine interaction over a network”. COMS utilizes two distinct type of Web Services; Simple Object Access Protocol (SOAP) and REpresentational State Transfer (RESTful).

The COMS application uses RESTful Web Services to transfer information between the frontend/user interface and the backend/database components. A RESTful Web Service allows a simple structure to exchange information between these two components. For the user interface to obtain information from the database component, the web browser sends an http “GET” request to the database component via a simple Uniform Resource Identifier (URI) or “web address” such as <http://coms.va.gov/Patients/viewall>

This web address communicates with the database component to retrieve a list of all patients in the system and to return specific information about those patients in the form of a JavaScript Object Notation (JSON) record, as follows:

```

{
  "success": true,
  "total": 41,
  "records": [
    {
      "id": "B521F525-6099-E111-8812-000C2935B86F",
      "name": "PATIENT FOURHUNDREDFIFTYFIVE",
      "DOB": "04/07/1935",
      "Gender": "M",
      "Age": "77",
      "DFN": "100455",
      "TemplateName": "2012-3-0001-ABCD-PACLITAXEL INJ,CONC 200-20120711",
      "TemplateDescription": "NSCLC - Paclitaxel Single Agent",
      "TemplateID": "BFF16C4E-74CB-E111-A078-000C2935B86F",
      "TreatmentStart": "07/11/2012",
      "TreatmentEnd": "10/03/2012",
      "Goal": "Curative",
      "ClinicalTrial": "",
      "WeightFormula": "Adjusted Weight",
      "BSAFormula": "Haycock",
      "PAT_ID": "AA5760E7-79CB-E111-A078-000C2935B86F",
      "BSA_Method": "Haycock",
      "PerformanceStatus": "1",
      "TreatmentStatus": "On-Going - Rest Day",
      "Amputations": []
    },
    ... 40 more records similar to the one above
  ]
}

```

The above JSON object communicates the model portion (specifically the “COMS.model.PatientInfo” model) of the user interface that the call to retrieve the patient information was successful, that it returned 41 patient records and then lists each of the 41 individual patient information records. The actual retrieval of the patient information from the database component is the task of the PHP code. The PHP code uses its own models, views and controllers to query the SQL Database and/or MDWS to obtain the information and (using the model’s data specific data manipulator) combine it into the data sent to the user interface.

Actions to save a new record to the database component are done via an http “POST” request. This action is also achieved via a simple URI with the data inside a JSON object. COMS accomplishes requests to update an existing record are via an http “PUT” request.

COMS uses the following service calls to communicate between the user interface and database components:

Service Name: ActiveWorkflows - Retrieve a list of the currently active workflows
 Service Name: AddCTOS - Apply a Chemotherapy Template to a specified patient
 Service Name: AddEoTS - Create a new End of Treatment Summary Record

Service Name: AddFlowSheetRecords - Update the changes made to the current Flowsheet to the patient's record for this treatment regimen

Service Name: AddLookup - Add new Data to the Lookup Table

Service Name: AddND_Assessment - Add a new Nursing Documentation Assessment record to the patient's record for this treatment regimen

Service Name: AddND_GenInfo - Add a new Nursing Documentation General Information record to the patient's record for this treatment regimen

Service Name: AddND_IVSite - Add a new Nursing Documentation IV Site Assessment record to the patient's record for this treatment regimen

Service Name: AddND_React_Assess - Add a new Nursing Documentation Reaction Assessment record to the patient's record for this treatment regimen

Service Name: AddND_Treatment - Add a new Nursing Documentation Treatment Assessment record to the patient's record for this treatment regimen

Service Name: AddPatientTemplate - Apply a new Template to a patient

Service Name: AddVitals - Add a new Vitals record to the patient's record for this treatment regimen

Service Name: AdminGlobals - Return a list of global variables

Service Name: AdminUsers - Return (or create a new) list of Users allowed to access the system

Service Name: Allergies - Return a list of all Allergies for a specified patient

Service Name: CTCAE_Data - Return the Common Terminology Criteria for Adverse Events data; retrieved from http://evs.nci.nih.gov/ftp1/CTCAE/CTCAE_4.03_2010-06-14.xls

Service Name: CTCAE_SOC - Return the Common Terminology Criteria for Adverse Events (CTCAE) System Organ Class (SOC) groupings as listed in the National Cancer Institute Center for Bioinformatics documentation

Service Name: CTOS - Return the Template Data for the specified Template

Service Name: DeleteLookup - Delete specific data from the Lookup Table

Service Name: DeleteTemplate - Delete a specific Treatment Template from the application

Service Name: DiseaseStage - Return list of all Stages used in "Select Disease Stage" drop down menu fields

Service Name: DiseaseType - Retrieve a list of disease types, used in the "Select Disease Type" drop down menu fields

Service Name: DrugRegimen - Return the information for a particular drug regimen (e.g. drug, dosage, and route of administration)

Service Name: DrugUnits - Return a list of drug units for use in the Units drop down menu fields

Service Name: Drugs - Return a list of drugs for use in the Drugs drop down menu fields

Service Name: Edit_OEMRecord - Edit a specific Order Entry Management (OEM) record to change a specific treatment order

Service Name: EmotegenicLevel - Return a list of Emotegenic Levels for use in the Emotegenic Level drop down menu fields

Service Name: EoTS - Return a specific End of Treatment Summary

Service Name: FlowSheetRecords - Return a specific Flow Sheet for a specific patient

Service Name: FluidType - Return a list of Fluid Types for use in the Fluid Types drop down menu fields

Service Name: HydrationDrug - Return a list of medications for use in the medication drop down menu fields

Service Name: Infusion - Return a list of Infusion Types for use in the infusion type drop down menu fields

Service Name: LabInfoResults - Return a list of all laboratory results for a specified patient

Service Name: Lookups - Generic service to return records for a specific LookupType in the Lookup Table

Possible Lookup Types are included in the following table:

Lookup Type ID	Lookup Type	Description
1	Diagnosis	Diagnosis Type Lookup
2	Drug	Drug Type Lookup
3	TreatmentIndicator	Treatment Indicator Selector Values
4	Regimen	Template Selector Values with Template Name in Description
5	TIProtocol	Treatment Indicator Protocol Selector Values
6	DiseaseCat	Disease Category Selector values
7	DiseaseType	Disease Types (aka Cancer Types) with Abbreviations in Description
8	DCBlood	Disease Blood Category Selector values
9	References	LookUp for References with URI in Description
10	PerformanceStatus	Patient's Performance Status with Sequence in Description
11	Unit	Medication Unit Measurement
12	Route	Regimen Route Type
13	Emetogenic	Emetogenic Level
14	LabTest	Laboratory Test Types
15	Health Care Provider	Health Care Provider Type
16	Specimen	Specimen Type
17	Lab Test Site	Laboratory Test Site
18	TimeFrameUnit	Various Time Frame Units
19	Total_Courses_Max	Lists the maximum number of treatment cycles that may be repeated for a given regimen
20	Cycle_Length_Max	Lists the maximum length (days/weeks) of any cycle within a treatment regimen
21	MasterTemplateRefXRef	Cross Reference from Master Template to References Lookups
22	TemplateSource	Location of Templates
23	DiseaseStage	Disease Stage
24	User	User Name/ID
25	TemplateAlias	Alias for template name
26	NonFormaDrug	Add Non-Formulary Drug
27	PatientAllergies	Allergies For a Patient
28	FluidType	Fluid Types
29	Allergies	Type of Allergy
30	PatientAmputations	Amputations for a Patient

Service Name: MDWSMatch - Return patient information from the MDWS system for the patient specified by the Data File Name (DFN) ID

Service Name: MegaMDWS - Start a search in the MDWS system to retrieve all the latest information on the specified patient to store in the COMS database

Service Name: MedsNonRounded - Return a list of medications which should NOT be rounded when calculating medication dosages

Service Name: ND_Assessment - Return a specific Nursing Documentation Assessment record from the patient's COMS record for this treatment regimen

Service Name: ND_GenInfo - Return a specific Nursing Documentation General Information record from the patient's COMS record for this treatment regimen

Service Name: ND_IVSite - Return a specific Nursing Documentation IV Site Assessment record from the patient's COMS record for this treatment regimen

Service Name: ND_React_Assess - Return a specific Nursing Documentation Reaction Assessment record from the patient's COMS record for this treatment regimen

Service Name: ND_Treatment - Return a specific Nursing Documentation Treatment Assessment record from the patient's COMS record for this treatment regimen

Service Name: OEMRecords - Return list of Order Entry Management (OEM) Records for specified patient

Service Name: Orders - Return a list of all current pharmaceutical orders in the COMS application

Service Name: PatientTemplate - Return a list of Template information for treatment regimens previously applied to the specified patient

Service Name: Patients - Retrieve information for a list of or specific patient used in the "Select Patient from CRPS" drop down menu fields

Service Name: ReadND_Treatment - Returns the Nursing Documentation Treatment records from the patient's COMS record for this treatment regimen

Service Name: Reasons - Retrieve a list of reasons for a workflow

Service Name: References - Retrieve a list of references for template definitions used in the authoring of a template

Service Name: SavePatient - Save all the current information for the current patient; does not include personally identifiable information (PII)

Service Name: Templates - Return list of templates available for specified source (i.e. National, Local, or My Templates)

Service Name: TimeFrameUnit - Return a list of Time Frame Units (days, weeks, months) which are used in the "Cycle Length" drop down menu fields

Service Name: Vitals - Return list of Vital Signs (e.g. blood pressure, height, and weight) for specified patient

COMS uses RESTful based Web Services to communicate between the user interface and database components. However, the MDWS system utilizes SOAP based Web Services to present its information to any requesting service, such as COMS.

COMS Secure Shell

Secure Shell (SSH) 2.0 is a cryptographic network protocol for secure data communication, remote shell services or command execution, and other secure network services between two networked computers connected via a secure channel. COMS uses SSH to connect to a VistA instance and transmit data regarding the patient and pharmacy orders generated within the application. This connection permits the

shell opening of a MUMPS database and allows the execution of FileMan commands for executing routines to read and write data in MUMPS. The PHP framework uses the phpseclib, mcrypt, gmp or bcmath to encrypt the connection between COMS and VistA. phpseclib is designed to be fully interoperable with Open Secure Sockets Layer (OpenSSL), using the standardized cryptography methods to transmit the data to VistA. Net_SSH2 library permits the exchange of COMS generated data and VistA patient records, notes, and pharmacy orders.

Medical Domain Web Services

Medical Domain Web Services (MDWS) is a suite of web services that exposes medical domain functionality, Medical Domain Objects (MDO). MDWS provides a web service function to allow an issuing VistA Remote Procedure Call (RPC). The web service MDWS uses the Simple Object Access Protocol (SOAP).

The MDWS suite of web services provides the client application developer with common SOAP web services and documentation necessary to create client applications. In turn, this incorporates the organization's business rules and provides access to data from multiple VistA instances and other disparate data sources. MDWS fosters COMS interoperability with VistA/CPRS and serves as a critical link in overall COMS functionality within the VHA clinical environment.

VA FileMan

COMS utilize SSH2's ability from within PHP to connect to a VistA instance and transmit the generated data for the patient and pharmacy orders. After the connection channel is established, COMS executes VA FileMan commands to navigate through VistA and execute commands to perform standard "silent" routines. These "silent" routines define, enter, and retrieve information from the local VistA instance for use within the COMS application and posting of information into VistA's electronic health record.

Databases

The COMS application uses MS SQL Server for storage of local data. COMS sends data and retrieves data from the SQL Databases via PHP based RESTful Web Service Calls. All data calls into the SQL Server are made via Stored Procedures designed in accordance with industry standard leading practices to reduce the probability of SQL Injection attacks and data corruption.

Service Calls – General Information

The default URI for all service calls is specified entirely in lower case.

Valid:

<http://example.com/patients>

Invalid:

<http://example.com/Patients>

Service calls return data as a valid JSON object as specified in Request For Comment (RFC) #4627 (<http://www.ietf.org/rfc/rfc4627.txt>), as shown in the following example:

Valid:

```
{ "name": "Simon" }
```

Invalid:

```
{ name: "Simon" }
```

```
{ 'name': "Simon" }
```

```
{ "name": 'Simon' }
```

Additionally, service calls issued with an http “GET” request have two basic return conditions. The return condition may be either, a “successful” return or a “failure” return as follows:

Service call returning a “successful” return condition:

```
{  
  "success" : true,  
  "total" : INT - representing the # of records returned  
  "records" : [] - array of 1 or more records  
}
```

Service call returning a “failure” return condition:

```
{  
  "success" : false,  
  "message" : "" - message detailing the reason for the failure  
}
```

If the service call was unable to return any records because none matched the search criteria, then a message of "No Records Found" should be returned with a success of false.

As such, all stores should be configured with a “REST” proxy and a “JSON” reader:

```
"proxy": {  
  "type": "rest",  
  "reader": {  
    "type": "json",  
    "root": "records",  
    "successProperty": "success",  
    "totalProperty": "total",  
    "messageProperty": "message"  
  }  
}
```

Note: When data is POSTed back to the server via any JSON Object, the service will automatically include the time/date stamp of the posting. The service will also include the User ID of the user who initiated the data POST back to the server.

This information is not listed in any Ext JS models as it is generated by the backend service call.

Application Breakdown

Patient Tab

The Patient Tab is the default view when accessing the COMS application. It is used to access patient-specific functionality, but not required to access Orders, Template Authoring, Messages, or Admin functionality.

Patient Selection

The COMS application provides three different methodologies for patient selection. Users may accomplish patient selection by Administration Date(s), CPRS Query, or Patient Selection from CPRS, as shown in Figure 1.

The screenshot shows the 'Chemotherapy Ordering Management System (COMS)' interface. At the top, there's a navigation bar with tabs: Patient, Orders, Template Authoring, Messages, and Admin. The 'Patient' tab is selected. Below the navigation bar, there's a 'Patient Selection' window. Inside this window, there are three search methods: 1. 'Enter a range of Administration Dates to search - From: 08/23/2012 To: []' with a link 'Select Patient by Administration Date(s)'. 2. 'OR Enter Patient Identification (SSN) to query CPRS: []' with a link 'Query CPRS for Patient'. 3. A note box stating: '(Note: For testing purposes, there are hundreds of patients available between 0010 and 0603. To search for a patient, use the spelling of the number for a last name and the number. For example: FiveHundredTwenty, Patient would be f0520 or OneHundredThirty, Patient would be o0130)'. Below the note box, there's a dropdown menu labeled 'Select Patient from CPRS:'. At the bottom right of the window, there are links for 'Help' and 'Logout'.

Figure 1: Patient Selection

Select Patient by Administration Date(s)

To select a patient by administration date(s), COMS utilizes the “Patients” Service Call. The application passes one or more dates (the “From” date if only one date passed) and will return a list of patients who have Administration Dates within the specified date range. By default, the service will return all patients who have future Administration Dates.

Options for the Patients Service Call include the following:

- `http://example.com/patients` – Return all patients who have Administration Dates \geq today
- `http://example.com/patients/mm_dd_yyyy` – Return all patients who have Administration Dates \geq the date passed
- `http://example.com/patients/mm_dd_yyyy/mm_dd_yyyy` – Return all patients who have Administration Dates \geq the first date passed but \leq the second date passed

Data is returned in the form of a standard Ext JS Data Store as a JSON Object based on the “PatientList” Model.

```
PatientList = [           // Array of Patients used to select a specific patient.
  {
    "PatName" : "", // Patient Name
    "PatID" : "",   // GUID to identify this specific patient
  }
]
```

]

Query CPRS for Patient

The “CPRSPatient” Service Call is used to query the Computerized Patient Record System (CPRS) using the CPRS Data File Name (DFN). This provides a Patient Index into the MDWS system and returns a single Patient Record in the form of a standard Ext JS Data Store as a JSON Object based on the “PatientList” Model. If, the DFN passed does not uniquely define a single patient, a list of patients will be returned by the service call.

The CPRSPatient Service Call for the COMS application is as follows:

- http://example.com/cprs_patient/##### – Return a specific patient whose DFN matches that passed. If the DFN passed is not complete, CPRS will return all patients matching the partial DFN.

Select Patient from CPRS

If either the “Patients” service call or the “CPRSPatient” service call returns a single patient, COMS presents a link to confirm the specified patient., as shown in Figure 2. Otherwise, COMS presents a selection or ComboBox to select a specific patient from the returned list.

The screenshot displays the 'Chemotherapy Ordering Management System (COMS) Proof of Concept - Demo System' interface. At the top, there is a navigation bar with tabs for 'Patient', 'Orders', 'Template Authoring', 'Messages', and 'Admin'. Below this, a 'Patient Selection' dialog box is open. It contains two search methods: 'Enter a range of Administration Dates to search - From: [] To: []' with a link 'Select Patient by Administration Date(s)', and 'OR Enter Patient Identification (SSN) to query CPRS: [b0080] Query CPRS for Patient'. A note box explains that for testing purposes, hundreds of patients are available between 0010 and 0603, and provides examples: 'FiveHundredTwenty, Patient' would be f0520 or OneHundredThirty, Patient would be o0130. At the bottom, it says 'Please click here to confirm this is the patient you want: EIGHTY-PATIENT BCMA', where the text 'EIGHTY-PATIENT BCMA' is circled in red. A 'Help Logout' link is at the bottom right.

Figure 2: Confirmation of Returned Patient

Regardless, selecting/confirming a specific patient will launch a series of service calls to retrieve all COMS and VistA information for the specified patient.

- <http://example.com/patient/data/GUID> - Retrieve basic details for specified patient
- <http://example.com/patient/mdws/GUID> - Retrieve all current MDWS data for specified patient
- <http://example.com/patient/labs/GUID> - Retrieve laboratory results for specified patient
- <http://example.com/patient/vitals/GUID> - Retrieve vital signs history for specified patient
- <http://example.com/patient/template/GUID> - Retrieve template history for specified patient
- <http://example.com/patient/oem/GUID> - Retrieve Order Entry Management Results for specified patient
- http://example.com/patient/template_data/GUID - Retrieve details on the current template applied to the specified patient

The data returned by this series of service calls are used to populate the individual sections of the Patient Tab as depicted in the following sections.

Patient Information

The “Patient Information” section of the “Patient” tab contains multiple panels to display data returned from the series of executed service calls. Figure 3 depicts the information displayed from the returned service calls.

Patient Orders Template Authoring Messages Admin

Patient Selection

Enter a range of Administration Dates to search - From: 08/23/2012 To: [Select Patient by Administration Date\(s\)](#)

OR

Enter Patient Identification (SSN) to query CPRS: [Query CPRS for Patient](#)

(Note: For testing purposes, there are hundreds of patients available between 0010 and 0603. To search for a patient, use the spelling of the number for a last name and the number. For example: **FiveHundredTwenty, Patient** would be **f0520** or **OneHundredThirty, Patient** would be **o0130**).

Select Patient from CPRS: PATIENT FOURHUNDREDFIFTYFIVE

Patient Information for - PATIENT FOURHUNDREDFIFTYFIVE

Patient Information

Gender:	M	Age:	77	Amputee:	None
BSA Weight Method:	Adjusted Weight	BSA Method:	Haycock	BSA:	1.93 Calculate BSA Show Calculations
Template:	2012-3-0001-ABCD-PACITAXEL INJ, CONC 200-20120711 NSCLC - Paclitaxel Single Agent Open Template in Chemotherapy Template Order Source (CTOS) Tab				
Regimen Status:	On-Going - Admin Day	Regimen Start Date:	07/11/2012	Regimen End Date:	10/03/2012
Type(s) of Cancer:					
Allergies:	Name	Type	Comment		
Clinical Trial:	NOT a clinical trial				

Treatment Regimens & Summaries (1 Record)

	Template Name	Start Date	End Date		
Current Template:	NSCLC - Paclitaxel Single Agent	07/11/2012	10/03/2012	Show Details	Stop Treatment

Patient History (56 Records)

Vital Statistics

Date	Temp	Pulse	BP	Resp	Pain	SP O ₂	PS	Height in Inches	Weight in lbs.	BSA			
										Weight Form.	Weight in KG	Method	BSA
08/23/2012	98.4	76	146/84	12	4		1	70	172	Adjusted Weight		Haycock	1.93 m ²
08/13/2012	98.7	72	146/92	10	5		1		174	Adjusted Weight		Haycock	

Laboratory Information (No Records Available)

Collection Date	Info	Name	Result	Acceptable Range	OUT of Range	Comment
-----------------	------	------	--------	------------------	--------------	---------

Figure 3: Patient Information from Service Calls

MDWS Data Service Call

The “MDWSData” service call issues a series of internal service calls to the MDWS system to retrieve any VistA/CPRS data for the specified patient. The “MDWSData” service call only returns the status of the success/failure of the service calls placed to MDWS to the frontend. All data retrieved is stored in the COMS internal database on the backend. An example of the MDWS data service call is as follows:

- <http://example.com/patient/mdws/GUID>

The GUID passed to the “MDWSData” service call is the “PatID” (consisting of the first letter of the patient’s last name and last four digits of the patient’s social security number). This is obtained from the “PatientList” service call when the patient was selected. Data is returned in the form of a standard Ext JS Data Store as a JSON Object based on the “PatientInformation” Model. The “MDWSData” service call returns a standard message body as follows:

```
{
  "success" : true,          // could return false if the service call failed
  "message" : "MDWS Msg: <Details on the status of the MDWS service calls made>"
}
```

Patient Data Service Call

The “Patient Data” service call returns some basic information from VistA/CPRS about the selected patient for use in the COMS application. The patient data service call is composed as follows:

- <http://example.com/patient/data/GUID>

The GUID passed to the “Patient Data” service call is the “PatID” obtained from the “PatientList” service call when the patient was selected. Data is returned in the form of a standard Ext JS Data Store as a JSON Object based on the “PatientInformation” Model, as follows.

```
PatientInformation = {          // This is a single element data structure
  "PatName" : "",              // Patient Name
  "PatID" : "",                 // GUID to identify the specific patient
  "PatTreatID" : "",           // Patient Treatment ID - uniquely identify all information
                                // related to this particular treatment regimen
  "Gender" : "",                // Patient Gender (M/F)
  "Age" : "",                   // Patient Age (in years) - integer
  "Amputations" : [             // Any amputations the patient has
    {
      "Amputation" : "",        // Amputation Description –
                                // derived from the allowable amputations table
      "PctBodyMass" : ""        // % of body mass the amputation removed,
                                // used in BSA Calculations
    }
  ],
  "BSA_WeightMethod" : "",       // BSA Weight Method –
                                // derived from the allowable BSA weight methods table
                                // specified when template applied
  "BSA_Method" : "",            // BSA Method - derived from the allowable BSA
                                // methods table specified when template applied
}
```



```

"BSA_Value" : "",          // BSA Value, calculated based on available criteria
"BSA_Calculations" : "",   // Actual formula used to derive the current BSA Value
"PatHeight" : "",          // Patient's Height (in Inches) used for the BSA Calculation
"PatWeight" : "",          // Patient's Weight (in Pounds) used for the BSA Calculation
"TemplateID" : "",         // Template ID – GUID to identify the specific template
                           // used in this treatment regimen
"TemplateDesc" : "",       // Template description
"TemplateName" : "",       // Template name
"TemplateAppliedDate" : "" // Date the template was applied
"OEM_ID" : ""              // GUID for the OEM Record for this patient/treatment
"RegimenStatus" : "",      // Regimen Status for the current calendar day –
                           // derived from the Regimen Status table
                           // (On-Going - Admin Day, On-Going - Treatment Day,
                           // Started, Ended, ???)
"NextAdminDate" : "",      // Future administration date derived from template
"RegimenStartDate" : "",   // Date the Regimen was started (First day of the first cycle,
                           // Not necessarily the date the template was applied)
"RegimenEndDate" : "",     // Date the Regimen was originally calculated to end
                           // (based on cycle duration and number of cycles)
"RegimenStopDate" : "",    // Date the Regimen was stopped (if it was stopped
                           // prematurely)
                           // this would be different than the Regimen_EndDate)
"CancerTypes" : [         // Types of cancer the patient has
  {
    "Type" : "",           // Cancer Type - derived from list of cancer types
    "Stage" : ""           // Stage - derived from the list of stages for each
                           // particular cancer type
  }
],
"Allergies" : [           // Any allergies the patient has,
                           // if the patient has NO allergies then this field is ""
  {
    "Name" : "",           // Name of the allergy
    "Type" : "",           // Type of allergy (food, medication, etc)
    "Comment" : ""         // Any comments in the patients record
  }
],
"ClinicalTrial" : ""      // If this is a clinical trial, then this would be the
                           // description of the trial, or it is ""
}

```

Within COMS, the "Patient Information" data is rendered in the "Patient Information" Panel via an Ext JS library xTemplate.

After rendering the Patient Information, links may be available for Body Surface Area (BSA) calculations. If a template has been applied to that patient, links are available for “Show Calculations” and “Open Template”. The “Show Calculations” link will display a pop-up with the calculations

performed to calculate the BSA. The “Open Template” link will display the currently applied Template information in the “Chemotherapy Template Order Source” tab.

Patient Treatment History Service Call

The “Patient Treatment History” service call returns a list of previously applied templates with links to treatment history/results associated with the applied templates. If there is a template currently applied to the patient this template information is also returned as the “CurrentTemplate”. The patient treatment history service call is as follows:

- <http://example.com/patient/treatments/GUID>

The GUID passed to the “Patient Treatment History” service call is the “PatID” obtained from the “PatientList” service call when the patient was selected. Data is returned in the form of a standard Ext JS Data Store as a JSON Object based on the “PatientTreatmentHistory” Model, as follows.

```
PatientTreatmentHistory = [ // This is an array of individual Patient Treatment History JSON Objects
{
    "CurrentTemplate" : True/False,
    "PatTreatID" : "", // Patient Treatment ID
                        // GUID to uniquely identify all information related
                        // to this particular treatment regimen
    "TemplateID" : "", // Template ID - GUID to identify the specific
                        // template used in this treatment regimen
    "TemplateDesc" : "", // Template description
    "TemplateName" : "", // Template name
    "TemplateAppliedDate" : "" // Date the template was applied
    "RegimenStatus" : "", // Regimen Status for the current calendar day
                        // derived from the Regimen Status table
                        // (On-Going - Admin Day, On-Going - Treatment Day,
                        // Started, Ended, ???)
    "RegimenStartDate" : "", // Date the Regimen was started (First day of the
                        // first cycle, Not Necessarily the date the template
                        // was applied)
    "RegimenEndDate" : "", // Date the Regimen was originally calculated to
                        // end (based on cycle duration and number of cycles)
    "RegimenStopDate" : "" // Date the Regimen was stopped (if it was stopped
                        // prematurely this would be different than the
                        // Regimen_EndDate)
                        // If "Regimen_StopDate" = "" then the treatment is
                        // still going on or treatment has been stopped
    "EoTS_ID" : "" // GUID for the End of Treatment Summary Record,
                        // if "EoTS_ID" = "" then EoTS has not been generated
}
]
```

The “PatientTreatmentHistory” data is rendered in the “Treatment Regimens & Summaries” panel via an Ext JS library xTemplate.

After rendering the Treatment Regimens & Summaries, each template will have the following links displayed:

- Show Details
- Stop Treatment
- Generate End of Treatment Summary
- Show End of Treatment Summary

Show Details: Selection will display the details of the Template and treatment performed information in the following COMS clinical modules within the Treatment Details panel:

Chemotherapy Template Order Source – Template Details
Order Entry Management – Individual orders for treatment
Nursing Documentation – Details of the treatment performed and patient assessment
Flowsheet – High level overview of the treatment performed and snapshot of ongoing care

Stop Treatment: Available for the Current Template only, selection will prompt the user to confirm that the treatment the patient is currently undergoing is to be stopped. If the user confirms stoppage, COMS will flag the treatment as stopped, and this link will be removed.

Generate End of Treatment Summary: Available when a treatment has ended but an End of Treatment Summary (EoTS) has not been generated, selection of this link will initiate the EoTS and guide the user through the process of generating an End of Treatment Summary.

Show End of Treatment Summary: Available when an End of Treatment Summary has been generated, selection of this link will display a read-only version of the End of Treatment Summary in a popup window.

Patient Vitals Service Call

The “Patient Vitals” service call returns patient vital signs data collected from VistA via the MDWS system. If the patient is currently undergoing a treatment, the vital signs returned are those taken during the treatment period. If the patient is currently not undergoing treatment, the vital signs returned are those obtained since the last treatment or all those contained within the VistA database. Data is retrieved from the COMS internal database, made available from the most recent “MDWSData” service call. The patient vitals service call is as follows.

- <http://example.com/patient/vitals/GUID>

The GUID passed to the “Patient Vitals” service call is the “PatID” obtained from the “PatientList” service call when the patient was selected. Data is returned in the form of a standard Ext JS Data Store as a JSON Object based on the “PatientVitalsHistory” Model, as follows.

PatientVitalsHistory = [// This is an array of individual Patient Vitals History JSON Objects

```
{
    "Date" : "",           // Date the measurement was taken
    "Temp" : "",           // Temperature in degrees Fahrenheit
    "Pulse" : "",          // Pulse in beats per minute
```

```

"BP" : "",           // Blood Pressure taken as Systolic / Diastolic mm of Hg
"Respiration" : "",  // Patient respirations in breaths per minute
"Pain" : "",         // Level of pain using the standard Comparative Pain Scale
                        // ranging from 0 - 10
"SPO2" : "",        // Saturation of Peripheral Oxygen in the blood
"PS" : [             // Performance Status - Using the ECOG
                        // (Eastern Cooperative Oncology Group) Scale
    {
        "PS_Level" : "",
        "PS_Desc" : ""
    }
],
"Height" : "",       // Patient height in inches
"Weight" : "",       // Patient weight in pounds
"BSA_WeightMethod" : "", // BSA Weight Method - derived from the allowable
                        // BSA weight methods table; specified when template
                        // applied
"BSA_Method" : "",   // BSA Method - derived from the allowable
                        // BSA methods table; specified when template applied
"BSA_Value" : "",    // BSA Value, calculated based on available criteria
"BSA_Calculations" : "" // Actual formula used to derive the current BSA Value
}
]

```

The “PatientVitalsHistory” data is rendered in the “Patient History” Panel via an Ext JS library xTemplate.

Laboratory Results Service Call

The “Laboratory Results” service call returns laboratory results data collected from VistA via the MDWS system. If the patient is currently undergoing a treatment, the laboratory results returned are those taken during the treatment period. If the patient is not currently undergoing a treatment, the results returned are since the last treatment or all those contained in the VistA database. Data is retrieved from the COMS internal database, made available from the most recent “MDWSData” service call. The laboratory results service call is as follows:

- <http://example.com/patient/labs/GUID>

The GUID passed to the “Laboratory Results” service call is the “PatID” obtained from the “PatientList” service call when the patient was selected. Data is returned in the form of a standard Ext JS Data Store as a JSON Object based on the “PatientLabs” Model, as follows.

```

PatientLabs = [       // Array of individual Laboratory Results JSON Objects
    {
        "Date" : "",   // Date of the laboratory test
        "Info" : "",   // Information on the laboratory test
        "Name" : "",   // Name of the test
        "Specimen" : "", // Specimen used for the test
    }
]

```

```

        "Result" : "",           // Results of the test
        "AccptRange" : "",       // What the acceptable range of the test is
        "InRange" : "",          // true if the results are within the acceptable range
        "Comment" : "" // Comments
    }
]

```

The “PatientLabs” data is rendered in the “Laboratory Information” panel via a Grid control from the Ext JS library. By default, data is grouped on the “Specimen” data, but may be changed and/or sorted through the Grids controls.

Chemotherapy Template Order Source

The “Chemotherapy Template Order Source” (CTOS) module within the “Patient” tab contains multiple tabs to display specific areas of information regarding the patient’s treatment. The CTOS module contains data obtained via the Patient Template Data Service Call rendered via an Ext JS xTemplate. This information is presented in the COMS Chemotherapy Template Order Source display, as shown in Figure 4.

Chemotherapy Template Order Source	Order Entry Management	Nursing Documentation	Flowsheet	
CANCER CHEMOTHERAPY IV ORDER SHEET Max Number of Cycles: 4 Cycle Length: 3 Weeks Chemotherapy Regimen Name: 2012-3-0001-ABCD-PACLITAXEL INJ,CONC 200-20120711 Emetogenic level: Low Febrile Neutropenia risk: 5 % Reference:				
Pre Therapy				
Instructions: Patient to take on chemotherapy days				
Sequence #	Drug	Dose	Route	Administration Day
1	DEXAMETHASONE TAB	20 mg	Oral	1
	Fluid/Volume:		Infusion Time:	
Take on chemotherapy days				
2	DIPHENHYDRAMINE CAP,ORAL	50 mg	Oral	1
	Fluid/Volume:		Infusion Time:	
Take on chemotherapy days				
3	RANITIDINE TAB	50 mg	Oral	1
	Fluid/Volume:		Infusion Time:	
For gastric reflux/distress, take on chemotherapy days				
Therapy				
Instructions: Use non-PVC containers for final dilution and 0.22 filter and tubing sets for administration				
Sequence #	Drug	Dose	Route	Administration Day
1	PACLITAXEL INJ,CONC	200 mg/m2	IVPB	1
	Fluid/Volume:	500	Infusion Time:	3 hrs / 0 min
Administer in 500ml Normal Saline, D5W, or Ringers Solution				
Post Therapy				
Instructions: For nausea/vomiting				
Sequence #	Drug	Dose	Route	Administration Day
1	DEXAMETHASONE TAB	20 mg	Oral	1
	Fluid/Volume:		Infusion Time:	
Take on chemotherapy days				
2	DIPHENHYDRAMINE CAP,ORAL	50 mg	Oral	1
	Fluid/Volume:		Infusion Time:	
Take on chemotherapy days				
3	RANITIDINE TAB	50 mg	Oral	1
	Fluid/Volume:		Infusion Time:	
For gastric reflux/distress, take on chemotherapy days				
<input type="button" value="Edit Template"/>				

Figure 4: Chemotherapy Template Order Source Display

Template Data Service Call

The “Template Data” service call retrieves details on the specified template. Typically this is the template currently applied to a patient, a template previously applied to a patient, or a template for review prior to applying to a patient. The template data service call is as follows:

- http://example.com/template_data/GUID

The GUID passed to the “Template Data” service call is the “TemplateID”. This is obtained either from the “PatientList” service call when the patient was selected if reviewing the template currently applied to the patient, or the “Patient Treatment History” service call to review a previously applied template. Data

is returned in the form of a standard Ext JS Data Store as a JSON Object based on the “Template Data” Model, as follows.

Template Data Model

```
{
  "id": "",
  "RegimenName": "",
  "Disease": "",
  "DiseaseStage": "",
  "CycleNumMax": "",
  "CycleLength": "",
  "CycleLengthUnit": "",

  // Emotegenic Level (Low, Med, Moderate, High)
  " ELevelName ": ""

  // Febrile Neutropenia Risk
  "FNRisk" : "", // Pct (whole number only, 0-100)
  "FNRecom" : "", // Text for recommendation

  "References": [],

  "PreTherapy" : {}, // JSON object using the Therapy Model for the Pre Therapy
                    // portion of the treatment
  "Therapy" : {}, // JSON object using the Therapy Model for the Therapy
                  // portion of the treatment
  "PostTherapy" : {} // JSON object using the Therapy Model for the Post Therapy
                    // portion of the treatment
}
```

Order Entry Management

Chemotherapy Template Order Source				Order Entry Management				Nursing Documentation				Flowsheet			
Order Entry Management (OEM) Information - for Patient: EIGHTY-PATIENT BCMA															
Regimen:		2012-3-0001-ABCD-PACLITAXEL INJ, CONC 200-20120711													
Description		NSCLC - Paclitaxel Single Agent													
Treatment Start:		08/24/2012													
Treatment End:		11/16/2012													
Neutropenia Risk:		5%		Recommendation:		The 2005 Update Committee agreed unanimously that reduction in febrile neutropenia(FN) is an important clinical outcome that justifies the use of CSFs, regardless of impact on other factors, when the risk of FN is approximately 20% and no other equally effective regimen that does not require CSFs is available. Primary prophylaxis is recommended for the prevention of FN in patients who are at high risk based on age, medical history, disease characteristics, and myelotoxicity of the chemotherapy regimen. CSF use allows a modest to moderate increase in dose-density and/or dose-intensity of chemotherapy regimens. Dose-dense regimens should only be used within an appropriately designed clinical trial or if supported by convincing efficacy data. Prophylactic CSF for patients with diffuse aggressive lymphoma aged 65 years and older treated with curative chemotherapy (CHOP or more aggressive regimens) should be given to reduce the incidence of FN and infections. Current recommendations for the management of patients exposed to lethal doses of total body radiotherapy, but not doses high enough to lead to certain death due to injury to other organs, includes the prompt administration of CSF or pegylated G-CSF									
Emesis Risk:		Low		Recommendation:		ASCO No antiemetic administered routinely pre- or postchemotherapy. NCCN No routine prophylaxis; consider using antiemetics listed under primary prophylaxis as treatment.									
Goal		Curative													
Performance Status		0-Fully active, able to carry on all pre-disease performance without restriction Change Performance Status													
Select Admin Day to view: <input type="text"/>															
Cycle 1 (of 4); Admin Day: 1 Date: 08/24/2012															
Pre Therapy Patient to take on chemotherapy days															
Drug		Dosing								Administration Time					
DEXAMETHASONE TAB (20 mg) <i>Take on chemotherapy days</i>		Drug		Dose		Administration									
		DEXAMETHASONE TAB		20 mg		Oral									
DIPHENHYDRAMINE CAP, ORAL (50 mg) <i>Take on chemotherapy days</i>		Drug		Dose		Administration									
		DIPHENHYDRAMINE CAP, ORAL		50 mg		Oral									
RANITIDINE TAB (50 mg) <i>For gastric reflux/distress, take on chemotherapy days</i>		Drug		Dose		Administration									
		RANITIDINE TAB		50 mg		Oral									
Therapy Use non-PVC containers for final dilution and 0.22 filter and tubing sets for administration															
Drug		Dosing								Administration Time					
PACLITAXEL INJ, CONC (200 mg/m2) <i>Administer in 500ml Normal Saline, D5W, or Ringers Solution</i>		Drug		Dose		Calculated Dose		Administration		9:00 AM					
		PACLITAXEL INJ, CONC		200 mg/m2		0 mg		IVPB							
		Fluid Type		Fluid Volume		Flow Rate		Infusion Time							
		Normal Saline		500 ml		167 ml/hr		3 hrs / 0 min							
Post Therapy For nausea/vomiting															
Drug		Dosing								Administration Time					
PROCHLORPERAZINE TAB (10 mg) <i>Take 2 tablets (10mg total) by mouth every six hours as needed for nausea/vomiting</i>		Drug		Dose		Administration									
		PROCHLORPERAZINE TAB		10 mg		Oral									
Digital Signature: Doctor															
Digital Signature: Co-Signer (Optional)															
Digital Signature: Pharmacist															

Figure 5: Order Entry Management Display

Order Entry Management Service Call

The “Order Entry Management” (OEM) service call returns all the OEM data listed in the Order Entry Management module. This information is available only if the patient is undergoing a treatment. If the patient is not currently undergoing a treatment or a request has not been made to view any historical treatment information, this service call cannot be issued. If the service call cannot be issued, the Order Entry Management module will display a “No Information Available” message. Figure 5 depicts the COMS Order Entry Management display. The OEM service call is as follows:

- <http://example.com/oem/GUID>

The GUID passed to the “OEM” service call is the “OEM_ID” obtained from the “PatientList” service call when the patient was selected. Data is returned in the form of a standard Ext JS Data Store as a JSON Object based on the “OEM” Model, as follows.

OEM (Order Entry Management) Model

```
{
    "id" : "",                // GUID for this record

    "FNRisk" : "",            // Febrile Neutropenia Risk
                                // Pct (number only, 0-100)
    "FNRecom" : "",          // Text for recommendation

                                // Emesis Risk
    "ELevelID" : "",          // GUID for the Emesis Risk record Information
    "ELevelName" : "",        // Level name (e.g. "Low", "Medium", "High", etc)
    "ELevelRecomASCO" : "",   // American Society of Clinical Oncology (ASCO)
                                // Recommendation
    "ELevelRecomNCCN" : "",   // National Comprehensive Cancer Network (NCCN)
                                // Recommendation
    "numCycles" : "",         // Number of treatment cycles in this template
    "DaysPerCycle" : "3",     // # of days in a treatment cycle
                                // (Note a cycle may be measured in days, weeks,
                                // months but this is always days)
    "Goal" : "",              // Goal for this particular treatment; curative or
                                // palliative (set at time template is applied to patient)
    "ClinicalTrial" : "",     // If this is a clinical trial then this is the trial description
                                // otherwise it's blank
    "Status" : "Ordered",     // status
    "PerformanceStatus" : "", // Performance Status, initially set at time template is
                                // applied, but can be changed during the treatment
    "LastPSChange" : "",      // Date the Performance Status was last changed
    "OEMRecords" : []         // Array of OEM Admin Day Models, as follows:
}
```

OEM Admin Day Model

```

{
    "id" : "",           // GUID for this record
    "Cycle" : "",        // Cycle # (1-N)
    "Day" : "",          // Day in Cycle # (1-N)
    "AdminDate" : "",    // Calendar Date (mm/dd/yyyy)
    "PreTherapy" : {},   // JSON object using the Therapy Model for the Pre Therapy
                        // portion of the treatment
    "Therapy" : {},      // JSON object using the Therapy Model for the Therapy
                        // portion of the treatment
    "PostTherapy" : {}   // JSON object using the Therapy Model for the Post Therapy
                        // portion of the treatment
}

```

Therapy Model

```

{
    "instr" : "",        // Instructions for this type of therapy
    "Therapy" : []       // Array of individual Therapy Order Models
}

```

Therapy Order Model

```

{
    "id" : "",           // GUID for this record
    "Order_ID" : "",     // GUID for the Order Record linked to this record
    "Instructions" : "",  // Instructions for this particular treatment
    "Med" : "",          // Medication for this treatment
    "MedID" : "",        // GUID for the Medication
    "Sequence" : "",     // Order in which this medication is to be given
    "AdminTime" : "",    // When on the Admin Day the medication is to be administered
    "Dose" : "",         // Dosage to be administered
    "DoseUnits" : "",    // Units of Measure for the Dosage
    "AdminMethod" : "",  // Method of administering the medication
    "BSA_Dose" : "",     // Dosage to be administered based on Body Surface Area
    "FluidType" : "",    // Type of fluid to be used for IV administration methods
    "FluidVol" : "",     // Fluid Volume to be used for IV administration methods
    "FlowRate" : "",     // Rate of flow of fluid (plus medication) to be used for
                        // IV administration methods
    "InfusionTime" : ""  // How long it should take to administer the medication if
                        // delivering via IV administration methods; automatically
                        // calculated as FluidVol / FlowRate
}

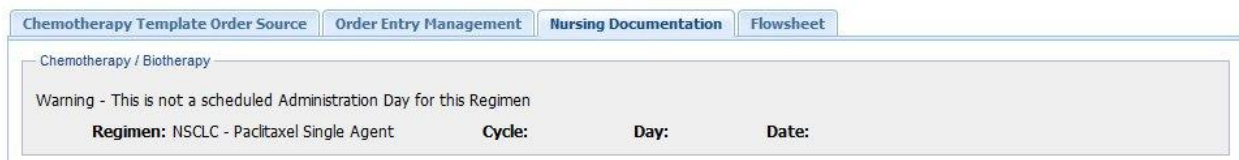
```

The “Order Entry Management” data is rendered in the “Order Entry Management” module within the “Patient Information” section of the “Patient” tab via an xTemplate component from the Ext JS library. A Combobox control for selecting a particular Administration Day within the entire Treatment period, or “Show All” Administration Days is available for filtering the “Order Entry Management” data.

Nursing Documentation

The “Nursing Documentation” module set contains a number of activity-based panels for nurses to enter information on patient treatment and assessment.

At the top of the “Nursing Documentation” module is the “Chemotherapy/Biotherapy” header. This header lists the current Regimen name, Cycle, Day, and Date as well as a notice if the current day is not a scheduled Administration Day, as shown in Figure 6.



The screenshot shows a web application interface with four tabs: "Chemotherapy Template Order Source", "Order Entry Management", "Nursing Documentation", and "Flowsheet". The "Nursing Documentation" tab is active. Below the tabs is a header section titled "Chemotherapy / Biotherapy". Inside this header, there is a warning message: "Warning - This is not a scheduled Administration Day for this Regimen". Below the warning, there are four fields: "Regimen: NSCLC - Paclitaxel Single Agent", "Cycle:", "Day:", and "Date:".

Figure 6: Nursing Documentation Header Display

Nursing Documentation – General Information Panel

The “General Information” panel of the Nursing Documentation module contains a collapsible section for displaying “Laboratory Information”. This section mirrors the “Laboratory Information” section from the “Patient Information” panel, presented in the ND module for the ease of view by oncology nurses.

After “Laboratory Information”, the next section is nurses to complete “Patient Identification” and “Patient Teaching”. The next section is for nursing verification of medication dosing. The remaining two sections are for nurses to record vital signs information and view historic vital signs. The ND module, General Information panel is shown in Figure 7.

Chemotherapy Template Order Source Order Entry Management **Nursing Documentation** Flowsheet

Chemotherapy / Biotherapy

Warning - This is not a scheduled Administration Day for this Regimen

Regimen: NSCLC - Paclitaxel Single Agent **Cycle:** **Day:** **Date:**

General Information Assessment IV Site Treatment Infusion Reactions Discharge Instructions

— Laboratory Information —

Patient Identification

Patient identification verified with 2 information sources?: Yes: ☐ No: ☐

Consent obtained?: Yes: ☐ No: ☐

Comment:

Patient Teaching

Education assessment complete?: Yes: ☐ No: ☐

Pre-procedure plan reviewed with patient/significant other, questions answered?: Yes: ☐ No: ☐

Dual Verification of Dosing

Sign to Verify
Sign to Verify

Vital Signs

Temp.: °F **Pulse:** **BP:** / **Patient Gender:** Male

Height: inches (cm) **Resp:** **SP O2%:** **Age:** 77

Weight: lbs (kg) **Pain:** **BSA:** Show [Calculations](#)

Save Cancel

Vital Signs - Historical

Date	Temp	Pulse	BP	Resp	Pain	SP O2	PS	Height in Inches	Weight in lbs.	BSA			
										Weight Form.	Weight in KG	Method	BSA

Figure 7: ND Module, General Information Panel

Oncology nurses populate the fields contained within the General Information panel. All information is then posted back to the server via the “NursingDoc GenInfo” service call when the user clicks the “Save” button. The Nursing Documentation General Information Service Call is as follows:

- <http://example.com/NursingDoc/GenInfo>

The “NursingDoc GenInfo” service call posts data to the COMS database in the form of a standard Ext JS Data Store as a JSON Object based on the “GenInfo” Model, as follows:

GenInfo Model

```
{
    "PatID" : "",                // GUID to identify this specific patient
    "PatIDGood" : <BOOL>,        // true indicates that the patient ID has been verified
    "ConsentGood" : <BOOL>,      // true indicates consent obtained
    "Comment" : "",              // Any comments entered
    "EducationGood": <BOOL>,     // true indicates Education Assessment Complete
    "PlanReviewed": <BOOL>      // true indicates Plan reviewed and all questions
                                // answered
}
```

Upon successful completion of the “POST” request, the “NursingDoc GenInfo” service call replies back with a standard JSON object response of the form as follows:

```
{
    "success": "true",
    "message": "Gen Info Save Successful"
}
```

If the “success” parameter is false, the “message” parameter will indicate the reason for the save failure.

COMS provides two (2) buttons for “Dual Verification of Dosing”. This functionality permits two (2) separate electronic signatures by two separate individuals to confirm that the calculated dosing is correct.

COMS utilizes “Vital Signs” entries to post the patient’s current vital signs back to the application’s database. . All Vital Signs are then displayed in the “Vital Signs – Historical” table; this mirrors the “Vital Signs” section of the “Patient Information” panel. The vital signs entered are posted back to the server via the “Patient Vitals” service call when the nurse clicks the “Save” button.

Note that the “Save” button can issue two (2) posts. One post is for the “Patient Identification” and “Patient Teaching” while the other is for the “Vital Signs”. If the “Patient Identification” and “Patient Teaching” or “Vital Signs” sections have not been populated, the “Save” button will only do a post back to the server for the populated section(s).

Nursing Documentation – Assessment Panel

The “Assessment” panel of the Nursing Documentation module contains a collapsible section for displaying “Notes on Assessment Events”. This is a block of static text.

Following the “Notes on Assessment Events” section, COMS provides a section for completing the nurse’s assessment of adverse events the patient experienced since the last treatment. This section consists of a series of check boxes. When checked/selected, COMS will display a Combobox listing the possible CTCAE levels for the event and a comment section to further qualify the adverse event.

Figure 8: ND Module, Assessment Panel

Upon completing the Assessment panel form, the nurse user must click the “Save” button for the data to be posted back to the server as part of the patient’s oncology record.

Populated fields within the form are posted back to the server via the “NursingDoc Assessment” service call upon clicking the “Save” button. The Nursing Documentation Assessment service call is as follows:

- <http://example.com/NursingDoc/Assessment>

The “NursingDoc Assessment” service call posts data in the form of a standard Ext JS Data Store as a JSON Object based on the “ND_Assessment” Model, as follows:

ND_Assessment Model

```
{
    "PatID" : "",           // GUID to identify this specific patient
    "AssessmentDetails" : [] // Array of individual “Assessment” Model objects
}
```

Assessment Model

```
{
    "sequence",          // The order that this record is to be displayed (
                        // fatigue = 1,
                        // anorexia = 2,
                        // nausea = 3,
                        // vomiting = 4,
                        // diarrhea = 5 all others as needed
    "fieldLabel",        // The label for the assessment: "Fatigue", "Anorexia",
                        // Nausea", etc
    "choice",            // The value chosen for this assessment: true, false, null
    "comments",          // The user entered comments
    "levelChosen"        // The level of the Assessment: 1, 2, 3, 4 etc.
                        // based on the type of assessment
}
```

Upon successful completion of the “POST” request the “NursingDoc Assessment” service call replies back with a standard JSON object response in the following form:

```
{
    "success": "true",
    "message": " Assesment Info Save Successful"
}
```

If the “success” parameter is false, the “message” parameter will indicate the reason for the save failure.

Nursing Documentation – IV Site Panel

The “IV Site” panel of the Nursing Documentation module contains several sections for nursing users to populate, as follows:

- IV Access
- Site Appearance
- Brisk blood return verified
- Comments

Chemotherapy Template Order Source Order Entry Management **Nursing Documentation** Flowsheet

Chemotherapy / Biotherapy

Warning - This is not a scheduled Administration Day for this Regimen

Regimen: NSCLC - Paclitaxel Single Agent **Cycle:** **Day:** **Date:**

General Information **Assessment** **IV Site** Treatment Infusion Reactions Discharge Instructions

IV Access

Date Accessed:

Device:

Gauge:

Location:

Site Appearance

☐ Absence of symptoms ☐ Pain ☐ Swelling ☐ Erythema ☐ Line Disconnected/Port De Accessed

Comments:

Brisk blood return verified

Pre treatment: Yes: ☐ No: ☐

During treatment: Yes: ☐ No: ☐

Post treatment: Yes: ☐ No: ☐

Comments:

Comments:

Figure 9: ND Module, IV Site Panel

Upon completing the form, the user must click the “Save” button for the data to be posted back to the server and COMS database.

Populated fields within the form are posted back to the server via the “NursingDoc IVSite” service call upon clicking the “Save” button. The Nursing Documentation IV Site service call is as follows:

- <http://example.com/NursingDoc/IVSite>

The “NursingDoc IVSite” service call posts data in the form of a standard Ext JS Data Store as a JSON Object based on the “ND_IVSite” Model, as follows:

ND_IVSite Model

```
{
    "PatID" : "",                // GUID to identify this specific patient
    "DateAccessed" : "",        // Date – mm/dd/yyyy format
    "Device" : "",              // From the “Device” Combobox
    "Gauge" : "",               // From the “Gauge” Combobox
}
```



```

        "Location" : "", // From the "Location" Combobox
        "Appearance" : [], // Array of Site Appearance Check Boxes checked
        "Appearance_Comments" : "", // Comments
        "BBR_Pre" : <BOOL>, // T/F for Brisk Blood Return verified before treatment
        "BBR_During" : <BOOL>, // T/F for Brisk Blood Return verified during treatment
        "BBR_Post" : <BOOL>, // T/F for Brisk Blood Return verified after treatment
        "BBR_Comments" : "", // Comments
        "Gen_Comments" : "" // General Comments
    }

```

Upon successful completion of the “POST” request, the “NursingDoc IVSite” service call replies back with a standard JSON object response in the following form:

```

{
    "success": "true",
    "message": " IV Site Info Save Successful"
}

```

If the “success” parameter is false, the “message” parameter will indicate the reason for the save failure.

Nursing Documentation – Treatment Panel

As shown in Figure 10, the “Treatment” panel of the Nursing Documentation module is a form grid for the user to document medication administration. The user confirms pre-populated entries and, if necessary, edits a cell or selects from a cell Combobox, to accurately record the treatment administered. The user must populate the “Start Time” (and the “End Time” if the medication is administered by IV) for each medication then select the “Sign to Verify” link. Selection of this link will prompt the user for his/her VistA access/verify code credentials for digital signature/verification of the individual medications administered.

Chemotherapy Template Order Source Order Entry Management **Nursing Documentation** Flowsheet

Chemotherapy / Biotherapy

Regimen: NSCLC - Paclitaxel Single Agent **Cycle:** 1 **Day:** 1 **Date:** 09/21/2012

General Information Assessment IV Site **Treatment** Infusion Reactions Discharge Instructions

Treatment Complete

Medication Given

Treatment Administered							
Medication	Dose	Units	Route	Start Time	End Time	Comments	Signature
Pre Therapy							
DEXAMETHASONE T...	20	mg	Oral				Sign to Verify
DIPHENHYDRAMINE...	50	mg	Oral				Sign to Verify
RANITIDINE TAB	50	mg	Oral				Sign to Verify
Therapy							
PACLITAXEL INJ,CO...	200	mg	IVPB				Sign to Verify
Post Therapy							
PROCHLORPERAZIN...	10	mg	Oral				Sign to Verify

Treatment Complete

Figure 10: ND Module, Treatment Panel

When the user accesses the “Treatment” tab, COMS will issue an “Orders” service call to retrieve all the dispensed orders for this patient. The application applies a filter to the Ext JS Data Store for the “Orders” service call to filter medications for display. Only those medications with an “orderstatus” of “Dispensed” and “PatID” of the current selected patient are returned by the “Orders” service call for display on the “Treatment” panel.

At the same time a “Treatment” service call is issued via an http “GET” request to retrieve the current treatments which have been applied to the current patient. The treatment service call is as follows:

- <http://example.com/Treatment/<GUID>>

The GUID passed to the “Treatment” service call is the “PatID” obtained from the “PatientList” service call when the patient was selected. Data is returned in the form of a standard Ext JS Data Store as a JSON Object based on the “Treatment” Model noted below.

Treatment Model

This “Treatment Model” supports both http “GET” and http “PUT” requests. Accordingly, the “GET” request will return a standard Ext JS “GET” service request as follows:

```
{
  "success" : true,
  "total" : INT - representing the # of records returned
  "records" : [] - array of 1 or more records of the “Treatment Model”
}
```

Single Treatment Model object

```
{
  "PatID" : "",           // GUID to identify this specific patient
  "PatTreatID" : "",      // Patient Treatment ID - uniquely identify all information
                          // related to this particular treatment regimen
  "TemplateID" : "",       // GUID to identify the specific template used in this
                          // treatment regimen
  "TreatmentID" : "",      // GUID for this particular Treatment Record
  "Cycle" : "",           // Cycle # (1-N)
  "Day" : "",             // Day in Cycle # (1-N)
  "AdminDate" : "",       // Calendar Date (mm/dd/yyyy)
  "Med" : "",             // Medication given
  "MedID" : "",           // GUID for the Medication
  "StartTime" : "",       // Date/Time (mm/dd/yyyy hh:mm am/pm) the med was
                          // administered
  "EndTime" : "",         // Date/Time (mm/dd/yyyy hh:mm am/pm) the med was
                          // stopped (only for IV meds)
  "Dose" : "",            // Dosage administered
  "DoseUnits" : "",       // Units of Measure for the Dosage
  "AdminMethod" : "",     // Method of administering the medication
  "Comments" : "",       //
  "User" : "",            // ID of the user who digitally signed this record
  "Date" : "",            // Date/Time (mm/dd/yyyy hh:mm am/pm) of signature
  "Med_Original" : "",    // Original values for the Med, Dose, Units and Administration
  "Dose_Original" : "",   // method
  "Units_Original" : "",  //
  "AdminMethod_Original" : "", //
}
```

When the “Treatment” service call sends data back to the server via an http “PUT” request, only a single Treatment Model record is sent.

As each medication treatment is digitally signed, a “Treatment” service call is issued via an http “PUT” to post a single Treatment Model record back to the server and backend database.

Nursing Documentation – Infusion Reactions Panel

The “Infusion Reactions” panel of the Nursing Documentation module contains four (4) collapsible sections to document infusion reactions to the chemotherapy treatment. Each section has a series of check boxes. When the user checks a box, COMS will display a comment field for entering additional details of the selected Infusion Reaction. Figure 11 shows the four Infusion Reaction sections as follows:

- Section 1: Extravasation
- Section 2: Cytokine-Release Syndrome
- Section 3: Hypersensitivity or Anaphylaxis
- Section 4: Other

Chemotherapy Template Order Source Order Entry Management **Nursing Documentation** Flowsheet

Chemotherapy / Biotherapy

Warning - This is not a scheduled Administration Day for this Regimen

Regimen: NSCLC - Paclitaxel Single Agent Cycle: Day: Date:

General Information Assessment IV Site Treatment **Infusion Reactions** Discharge Instructions

Extravasation

☒ Topical heating applied

Frequency:

☐ Topical cooling applied

☐ Interventions

☐ Antidotes

☐ Measurements

☐ Edema

☐ Erythema

☐ Discomfort with movement

☐ Other

Cytokine-Release Syndrome

☐ Fever

☐ Chills

☐ Rigors

☐ Nausea

☐ Hypotension

☐ Tachycardia

☐ Asthenia

☐ Headache

☐ Rash

☐ Tongue and Laryngeal Edema

☐ Dyspnea

☐ Other

Hypersensitivity or Anaphylaxis

☐ Uneasiness or Agitation

☐ Chest Tightness

☐ Hypotension

☐ Dyspnea

☐ Wheezing

☐ Urticaria

☐ Periorbital or facial edema

☐ Abdominal

☐ Cramping

☐ Diarrhea

☐ Nausea

☐ Other

Other

☐ Other

Save Cancel

Figure 11: ND Module, Infusion Reactions Panel

The “Infusion Reactions” panel employs an “Infusion” service call for both “GET” and “POST” http requests in the following format:

http “GET” Request

- [http://example.com/ Infusion/GUID](http://example.com/Infusion/GUID)

http “POST” Request

- [http://example.com/ Infusion](http://example.com/Infusion)

The GUID passed to the “Infusion” service call for the http “GET” request is the “PatTreatID” obtained from the “PatientInformation” data obtained from the “Patient Data” service call when the patient was selected.

For the http “GET” request, COMS returns data in the form of a standard Ext JS Data Store as a JSON object based on the “Treatment Infusion” Model. This JSON object contains all the Infusion Reaction records for the current treatment regimen. Conversely, the http “POST” request sends a single record based on the “Infusion” Model to the server/backend database and a single “Status” message is returned upon completion of the service call to indicate the record was saved.

Treatment Infusion Model

```
{
    "success" : <BOOL>,
    "total" : <INT>,          // Representing the # of records returned
    "records" : []           // Array of 1 or more “Infusion Model” records
}
```

Infusion Model object

```
{
    "PatID" : "",             // GUID to identify this specific patient
    "PatTreatID" : "",        // Patient Treatment ID - uniquely identify all information
                                // related to this particular treatment regimen
    "TemplateID" : "",         // GUID to identify the specific template used in this
                                // treatment regimen
    "TreatmentID" : "",        // GUID for this particular Treatment Record
    "Cycle" : "",              // Cycle # (1-N)
    "Day" : "",                // Day in Cycle # (1-N)
    "AdminDate" : "",          // Calendar Date (mm/dd/yyyy)
    "User" : "",               // ID of the user who digitally signed this record
    "Date" : "",               // Date/Time (mm/dd/yyyy hh:mm am/pm) of signature
    "InfusionData" : []        // Array of 1 or more “Infusion Data” records
}
```

Infusion Data Model object

```
{
    "ReactionType" : "",       // The Infusion Reaction Type checked
    "Comment" : ""             // Any comments entered for this specific Infusion Reaction
                                // Type
}
```

Upon successful completion of the “POST” request, the “Infusion” service call replies back with a standard JSON object response of the following form:

```
{
    "success": "true",

```

```

    "message": " Infusion Reaction Info Save Successful"
}

```

If the “success” parameter is false, the “message” parameter will indicate the reason for the save failure.

Nursing Documentation – Discharge Instructions Panel

The “Discharge Instructions” panel of the Nursing Documentation module is a form for the user to enter information related to the patient’s discharge. This includes any patient education, future appointments, and discharge instructions the nurse provided to the patient. Figure 12 shows the Discharge Instructions panel.

Chemotherapy Template Order Source Order Entry Management **Nursing Documentation** Flowsheet

Chemotherapy / Biotherapy

Warning - This is not a scheduled Administration Day for this Regimen

Regimen: NSCLC - Paclitaxel Single Agent **Cycle:** **Day:** **Date:**

General Information Assessment IV Site Treatment Infusion Reactions **Discharge Instructions**

Patient Education

Patient Education: Yes: ☐ No: ☐

Comments:

Follow up: Inpatient: ☐ Outpatient: ☐

Next Chemotherapy Appt.:

Next Clinic Appt.:

Laboratory Test(s) Scheduled

Discharge Instructions

☐ Patient was given Chemotherapy discharge instructions.

Select Instructions:

No Instructions available at this time

Comments:

Save Cancel

Figure 12: ND Module, Discharge Instructions Panel

The “Discharge Instructions” tab employs a “Discharge” service call. This allows for both “GET” and “POST” http requests in the following format:

http “GET” Request

- <http://example.com/Discharge/GUID>

http "POST" Request

- <http://example.com/Discharge>

The GUID passed to the "Discharge" service call for the http "GET" request is the "PatTreatID" obtained from the "PatientInformation" data obtained from the "Patient Data" service call when the patient was selected.

For the http "GET" request, COMS returns data in the form of a standard Ext JS Data Store as a JSON object based on the "Discharge Instructions" Model. This JSON object contains the Discharge Instructions record for the current treatment regimen. The http "POST" request sends a single record based on the "Discharge Instructions" Model to the server/backend database and a single "Status" message is returned upon completion of the service call to indicate the record was saved.

Discharge Instructions Model

```
{
    "success" : <BOOL>,
    "total" : <INT>,          // Representing the # of records returned (should == 1)
    "records" : []           // Array of 0 or 1 "Discharge" record
}
```

Discharge Model object

```
{
    "PatID" : "",             // GUID to identify this specific patient
    "PatTreatID" : "",        // Patient Treatment ID - uniquely identify all information
                                // related to this particular treatment regimen
    "TemplateID" : "",        // GUID to identify the specific template used in this
                                // treatment regimen
    "TreatmentID" : "",       // GUID for this particular Treatment Record
    "Cycle" : "",             // Cycle # (1-N)
    "Day" : "",               // Day in Cycle # (1-N)
    "AdminDate" : "",         // Calendar Date (mm/dd/yyyy)
    "User" : "",              // ID of the user who digitally signed this record
    "Date" : "",              // Date/Time (mm/dd/yyyy hh:mm am/pm) of signature
    "Education" : <BOOL>,     // Patient Education checked
    "EduComments" : "",       // Comments for Patient Education
    "Followup" : "",          // Inpatient/Outpatient Followup
    "NextChemo" : "",         // Date for next Chemo Apt (mm/dd/yyyy)
    "NextClinic" : "",        // Date for next Clinic Apt (mm/dd/yyyy)
    "LabTest1" : "",          // Date for Lab Tech Apt (mm/dd/yyyy)
    "LabTest2" : "",          // Date for Lab Tech Apt (mm/dd/yyyy)
    "GivenInstructions" : <BOOL>, // Patient was given instructions
    "Instructions" : [],      // List of instructions given
    "Comments" : ""           // Comments
}
```

Upon successful completion of the “POST” request, the “Discharge” service call replies back with a standard JSON object response of the following form:

```
{
  "success": "true",
  "message": " Discharge Instructions Info Save Successful"
}
```

If the “success” parameter is false, the “message” parameter will indicate the reason for the save failure.

Orders

COMS utilizes the “Orders” tab to display all outstanding pharmacy orders for the next three (3) calendar days, as shown in Figure 13. These orders will have an order status of ordered, cleared, finalized, in-coordination, dispensed, or administered for each medication prescribed by the oncology provider for the specified patient.

Patient Orders Template Authoring Messages Admin												
Name	Admin Day	Type	Drug	Dosage	Units	Route	Fluid/ Volume	Flow Rate	Instructions	Order Status	Set Status	
Admin Date: 09/12/2012												
FOURHUNDREDFIFTYFIVE PATIE...	1	Pre Therapy	DEXAMETHASO...	20	mg	Oral	0		Take on chemothera...	Ordered		Update
FOURHUNDREDFIFTYFIVE PATIE...	1	Pre Therapy	DIPHENHYDRA...	50	mg	Oral	0		Take on chemothera...	Ordered		Update
FOURHUNDREDFIFTYFIVE PATIE...	1	Pre Therapy	RANITIDINE TAB	50	mg	Oral	0		For gastric reflux/di...	Ordered		Update
FOURHUNDREDFIFTYFIVE PATIE...	1	Post Thera...	PROCHLORPER...	10	mg	Oral	0		Take 2 tablets (10m...	Ordered		Update
FOURHUNDREDFIFTYFIVE PATIE...	1	Therapy	PACLITAXEL INJ...	386	mg	IVPB	500	167	Administer in 500ml ...	Ordered		Update
FIVEHUNDREDFIFTEEN PATIENT	1	Pre Therapy	DEXAMETHASO...	20	mg	Oral	0		Take on chemothera...	Ordered		Update
FIVEHUNDREDFIFTEEN PATIENT	1	Pre Therapy	DIPHENHYDRA...	50	mg	Oral	0		Take on chemothera...	Ordered		Update
FIVEHUNDREDFIFTEEN PATIENT	1	Pre Therapy	RANITIDINE TAB	50	mg	Oral	0		For gastric reflux/di...	Ordered		Update
FIVEHUNDREDFIFTEEN PATIENT	1	Post Thera...	PROCHLORPER...	10	mg	Oral	0		Take 2 tablets (10m...	Ordered		Update
FIVEHUNDREDFIFTEEN PATIENT	1	Therapy	PACLITAXEL INJ...	394	mg	IVPB	500	167	Administer in 500ml ...	Ordered		Update
FIVEHUNDREDFIFTY PATIENT	1	Pre Therapy	DEXAMETHASO...	20	mg	Oral	0		Take on chemothera...	In-Coordination		Update
FIVEHUNDREDFIFTY PATIENT	1	Pre Therapy	DIPHENHYDRA...	50	mg	Oral	0		Take on chemothera...	Ordered		Update
FIVEHUNDREDFIFTY PATIENT	1	Pre Therapy	RANITIDINE TAB	50	mg	Oral	0		For gastric reflux/di...	Ordered		Update
FIVEHUNDREDFIFTY PATIENT	1	Post Thera...	PROCHLORPER...	10	mg	Oral	0		Take 2 tablets (10m...	Ordered		Update
FIVEHUNDREDFIFTY PATIENT	1	Therapy	PACLITAXEL INJ...	330	mg	IVPB	500	167	Administer in 500ml ...	Ordered		Update

Figure 13: Orders Tab

Orders Service Call

The “Orders” service call returns all pharmacy order data for the next three (3) calendar days, collected from VistA via the MDWS system.

- <http://example.com/Orders>

COMS returns data in the form of a standard Ext JS Data Store as a JSON object based on the “PharmacyOrder” Model as follows:

```
PharmacyOrder = [           // Array of individual Pharmacy Order JSON Objects

    {
        "PatID" : "",        // Patient ID - GUID from DB Table
        "Last_Name" : "",
        "Cycle" : ,
        "templateID" : "",    // GUID
        "adminDay" : ,
        "adminDate" : "",     // mm/dd/yyyy
        "drug" : "",
        "type" : "",
        "typeOrder" : 1,
        "dose" : 20,
        "unit" : "",
        "route" : "",
        "fluidVol" : 0,
        "flowRate" : "",
        "instructions" : "",
        "Order_ID" : "",      // GUID
        "orderstatus" : "",
        "orderid" : ""        // GUID
    }
]
```

The “Pharmacy Order” data is rendered in the “Orders” Tab via a Grid control from the Ext JS library. By default, the data is grouped on the “Administration Date” data. However, users may change and/or sort the data through the Grids controls.

Template Authoring

COMS provides the “Template Authoring” tab for authorized users to create new Chemotherapy Order Templates either from an existing template or creating one from scratch. Figure 14 shows the Template Authoring tab.

Disease Selection

The “Disease Selection” Combobox is used to select the type of cancer for treatment by a specific regimen. This Combobox is linked to the “Stage Selection” Combobox to select a particular stage (or progression) of the specified cancer. Accordingly, the “Stage Selection” Combobox is dependent upon the type of cancer selected in the “Disease Selection” Combobox.

Disease Type Service Call

The “Disease Type” service call is used to obtain the list of cancers and associated stages available in the application. COMS utilizes this service call to populate the data store for the “Disease Selection” Combobox as follows:

- <http://example.com/LookUp/view/DiseaseType>

Data is returned in the form of a standard Ext JS Data Store as a JSON object based on the following “DiseaseType” Model:

```
{
  "id" : "",
  "name" : "",           // Name of this form of cancer
  "description" : "",    // Description of this form of cancer
  "stages" : [""]       // List of stages for this particular cancer, used as a feed for the
                        // “Stage Selection” Combobox
}
```

Emetogenic Level Selection

The “Emetogenic Level” service call is used to obtain the list of Emetogenic levels (i.e. capacity to induce emesis or vomiting). It is used to populate the data store for the “Emetogenic Level Selection” Combobox as follows:

- <http://example.com/LookUp/view/Emetogenic>

COMS returns data in the form of a standard Ext JS Data Store as a JSON object based on the following “EmetogenicLevel” Model:

```
{
  "id": "",           // GUID
  "name": "",         // Level name (“Minimal” – “High”)
  "level": "",        // Emetogenic Level (level 1 – level 4)
  "description" : ""  // Description for the level
}
```

Emetogenic Level reference pages

More information on Emetogenic Levels is available on the following clinical websites:

<http://theoncologist.alphamedpress.org/content/4/3/191.full>

<http://www.healthprolink.com/mobile/tools/Oncology/Emetogenicscale.htm>

Emesis level predicts the percentage of patients who experience emesis (i.e. vomiting) without administration of an effective emesis prophylaxis.

- 1 = < 10%
- 2 = 10-30%
- 3 = 30-60%
- 4 = 60-90%
- 5 = > 90%

<http://emedicine.medscape.com/article/1355706-treatment>

Further, emesis levels have defined descriptions to qualify the numeric values as the following paragraph notes: Level (Risk) : (Desc)

level 4 (High): More than 90% of patients who receive these chemotherapy agents experience nausea and vomiting. Carmustine, cisplatin, cyclophosphamide (>1500 mg/m²), dacarbazine, dactinomycin, mechlorethamine, streptozotocin Serotonin-receptor antagonist, dexamethasone, and aprepitant

level 3 (Moderate): Nausea and vomiting occurs in 30-90% of patients who receive these chemotherapy agents. Carboplatin, cyclophosphamide (< 1500 mg/m²), cytarabine (>1 g/m²), daunorubicin, doxorubicin, epirubicin, idarubicin, ifosfamide, irinotecan, oxaliplatin Serotonin-receptor antagonist, and dexamethasone

level 2 (Low): Nausea and vomiting occurs in 10-30% of patients who receive these chemotherapy agents. Bortezomib, cetuximab, cytarabine (< 1 g/m²), docetaxel, etoposide, fluorouracil, gemcitabine, methotrexate, mitomycin, mitoxantrone, paclitaxel, pemetrexed, topotecan, and trastuzumab Serotonin-receptor antagonist

level 1 (Minimal): Less than 10% of patients who receive these chemotherapy agents experience nausea and vomiting. Bevacizumab, bleomycin, busulfan, 2-chlorodeoxyadenosine, fludarabine, rituximab, vinblastine, vincristine, and vinorelbine. No antiemetic routinely administered

<http://www.cancer.gov/cancertopics/pdq/supportivecare/nausea/HealthProfessional/Table3>

Clinical guidelines also provide Antiemetic Recommendations by Emetic Risk Categories.

Emetic Risk Category ASCO Guidelines NCCN Guidelines

High (>90%) risk

ASCO Guidelines - Three-drug combination of a 5-HT₃ receptor antagonist, dexamethasone, and aprepitant is recommended as pre-chemotherapy treatment.

For patients receiving cisplatin and all other agents of high emetic risk, the two-drug combination of dexamethasone and aprepitant is recommended for prevention of delayed emesis.

NCCN Guidelines –Pre-chemotherapy, a 5-HT₃ receptor antagonist (ondansetron, granisetron, dolasetron, or palonosetronb), dexamethasone (12 mg), and aprepitant (125 mg) recommended, with or without lorazepam.

For prevention of delayed emesis, dexamethasone (8 mg) on days 2 - 4 plus aprepitant (80 mg) on days 2 and 3 recommended, with or without lorazepam on days 2 - 4.

Moderate (30%-90%) risk -

ASCO Guidelines - For patients receiving an anthracycline and cyclophosphamide, the three-drug combination of a 5-HT3 receptor antagonist, dexamethasone, and aprepitant recommended prechemotherapy; single-agent aprepitant recommended on days 2 and 3 for prevention of delayed emesis. For patients receiving other chemotherapies of moderate emetic risk, the two-drug combination of a 5-HT3 receptor antagonist and dexamethasone recommended prechemotherapy; single-agent dexamethasone or a 5-HT3 receptor antagonist recommended on days 2 and 3 for prevention of delayed emesis.

NCCN Guidelines - For patients receiving an anthracycline and cyclophosphamide and selected patients receiving other chemotherapies of moderate emetic risk (e.g., carboplatin, cisplatin, doxorubicin, epirubicin, ifosfamide, irinotecan, or methotrexate), a 5-HT3 receptor antagonist (ondansetron, granisetron, dolasetron, or palonosetronb), dexamethasone (12 mg), and aprepitant (125 mg) recommended, with or without lorazepam, prechemotherapy; for other patients, aprepitant is not recommended. For prevention of delayed emesis, dexamethasone (8 mg) or a 5-HT3 receptor antagonist on days 2 - 4 or, if used on day 1, aprepitant (80 mg) on days 2 and 3, with or without dexamethasone (8 mg) on days 2 - 4, recommended, with or without lorazepam on days 2 - 4.

Low (10%-30%) risk -

ASCO Guidelines - Dexamethasone (8 mg) recommended; no routine preventive use of antiemetics for delayed emesis recommended.

NCCN Guidelines - Metoclopramide, with or without diphenhydramine; dexamethasone (12 mg); or prochlorperazine recommended, with or without lorazepam.

Minimal (<10%) risk -

ASCO Guidelines - No antiemetic administered routinely pre- or postchemotherapy.

NCCN Guidelines - No routine prophylaxis; consider using antiemetics listed under primary prophylaxis as treatment.

Adding a new Medication to a Therapy Treatment

The “Add New Drug Regimen” pop-up dialog box is a form used to create a new Drug Regimen record for pre-therapy, therapy, or post-therapy administration. Figure 15 shows this pop-up dialog box.

Figure 15: Add New Drug Regimen

When the user completes this form, COMS issues an http “POST” request to push the data.

Med Regimen Service Call

- <http://example.com/MedRegimen>

COMS passes this “MedRegimen” service call via a standard Ext JS Data record as a JSON object based on the following “MedRegimen” Model:

MedRegimen Model

```
{
  "id" : "",           // GUID for this record
  "MedType" : "",      // Type of medication (inpatient or outpatient)
  "AdminDays" : "",    // List of days this medication is to be administered
                        // given as a single #,
                        // comma separated list of #'s
                        // or a range of #'s
  "Instructions" : "", // Instructions for this particular treatment
  "Med" : "",          // Medication for this treatment
  "MedID" : "",        // GUID for the Medication (is this needed in this model?)
  "Sequence" : "",     // Order in which this med is to be given
  "AdminTime" : "",    // When on the Admin Day the med is to be administered
  "Dose" : "",         // Dosage to be administered
  "DoseUnits" : "",    // Units of Measure for the Dosage
  "AdminMethod" : "",  // Method of administering the medication
  "FluidType" : "",    // Type of fluid to be used for IV administration methods
  "FluidVol" : "",     // Fluid Volume to be used for IV administration methods
  "FlowRate" : "",     // Rate of flow of fluid (plus medication) to be used for
                        // IV administration methods
  "InfusionTime" : ""  // How long it should take to administer the medication if
                        // delivering via IV administration methods
}
```

Messages

COMS utilizes the “Messages” tab to display any healthcare team messages intended for a specific role ID. Messages are not necessarily intended for a specific user. Figure 16 shows the “Messages” tab.

Patient Orders Template Authoring Messages Admin							
My Messages							
Date Sent	Time	To	From	CC	Subject	Action	
August 24, 20...	1117	pharmacist10@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTY-PATIENT BCMA		
August 24, 20...	1117	pharmacist9@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTY-PATIENT BCMA		
August 24, 20...	1117	pharmacist8@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTY-PATIENT BCMA		
August 24, 20...	1117	pharmacist7@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTY-PATIENT BCMA		
August 24, 20...	1117	pharmacist6@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTY-PATIENT BCMA		
August 24, 20...	1117	pharmacist5@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTY-PATIENT BCMA		
August 24, 20...	1117	pharmacist4@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTY-PATIENT BCMA		
August 24, 20...	1117	pharmacist3@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTY-PATIENT BCMA		
August 24, 20...	1117	pharmacist2@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTY-PATIENT BCMA		
August 24, 20...	1117	pharmacist1@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTY-PATIENT BCMA		
August 24, 20...	1117	pharmacist10@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTY-PATIENT BCMA		
August 6, 2012	1731	pharmacist9@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTYONE-PATIENT BCMA		
August 6, 2012	1731	pharmacist8@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTYONE-PATIENT BCMA		
August 6, 2012	1731	pharmacist7@dbi...	programmer1@dbi...	programmer1@dbi...	INFO: Order Notification to Pharmacist for patient EIGHTYONE-PATIENT BCMA		
Refresh							

Figure 16: Messages Tab

Messages Service Call

The “Messages” service call returns all messages for the role ID listed for the current user. The messages service call is as follows:

- http://example.com/Messages/Filtered/RID/ROLE_ID

COMS returns data in the form of a standard Ext JS Data Store as a JSON object based on the following “Message” Model.

```

Message = [                                // Array of individual Message JSON Objects
{
    "mid" : INT,                            // Message ID, Integer representing the id of this message
    "MTo" : "",                             // Email address of the intended recipient of the message
    "CC" : "",                              // Email address of any carbon/courtesy copy recipients
    "Subject": "",                          // Subject of the message
    "Message": "",                          // Body of the message
    "Date": {                               // Date info of when the message was sent
        "date" : "2012-08-24 11:17:39",
        "timezone_type" : 3,
        "timezone" : "America/New_York"
    },
    "MFrom" : "",                          // Email address of the user originating the message action
    "rid" : 26,
    "wid" : 75,
    "dateSent" : "August 24, 2012",
    "timeSent" : "1117",
    "timeZone" : "America/New_York",
},

```

```

    "MStatus" : "Unread",
    "OpenLink" : "https://coms-uat.dbitpro.com/showMessage.php?mid=638"
  },

```

The “Message” data is rendered in the “Messages” tab via a Grid control from the Ext JS library. By default, the data is sorted on the “Date”. However, users may change and/or sort data through the Grid’s controls.

Admin

COMS provides various administrative functionality for maintaining and tailoring the application for local facility preferences. Figure 17 shows the “Admin” tab.

Figure 17: Admin Tab

Add Lookups

The “Lookup” service call returns all data for the specified data group as follows:

- <http://example.com/LookUp/viewall>

COMS returns data in the form of a standard Ext JS Data Store as a JSON object based on the specific “Lookup” Model. The “Lookup” Model for allergies contained within the application is as follows:

```

{
  "id": "29",
  "value": "Allergies",
  "type": "0",
  "description": "Type of Allergy"
},

```

Delete Template

- <http://example.com/Lookup/Templates>

```

{
  "id": "BFF16C4E-74CB-E111-A078-000C2935B86F",
  "type": "4",
  "name": "PACLITAXEL INJ,CONC 200",

```



```

    "description": "NSCLC // Paclitaxel Single Agent",
    "totnum": "4",
    "length": "3",
    "unit": "Weeks",
    "coursenum": "0",
    "emolevel": "Low",
    "fnrisk": "5",
    "version": "1",
    "regimenid": "BBF16C4E-74CB-E111-A078-000C2935B86F"
  }

```

- <http://example.com/Lookup/Templates/GUID>
Use “DELETE” HTTP command to delete the template specified by the passed GUID

Global Variables

- <http://example.com/Admin/Globals>

```

{
  "success": true,
  "total": 1,
  "records": [
    {
      "domain": "coms-uat.dbitpro.com",
      "sitelist": "355"
    }
  ]
}

```

COMS Users

- <http://example.com/Admin/Users>

```

{
  "rid": 29,
  "username": "01vehu",
  "role": "Administrator",
  "DisplayName": "Administrator",
  "Email": "admin@dbitmail.com",
  "cprsUsername": "01vehu",
  "cprsPass": "vehu01"
},

```

Active Workflows

- <http://example.com/Admin/ActiveWorkflows>

```
{
  "ID": 13,
  "WorkflowID": 13,
  "WorkflowName": "Cancelled",
  "Active": 1,
  "Reason": "Notification",
  "NoSteps": 1,
  "ReasonNo": 14,
  "LastIssued": 46,
  "Body": "Your order has been cancelled."
},
```

Medications Not Rounded

- <http://example.com/Admin/MedsNonRounded>

```
{
  "success": true,
  "total": 1,
  "records": [
    {
      "Lookup_ID": "94A0B573-A48C-E111-A87B-000C2935B86F",
      "Name": "ABACAVIR SOLN,ORAL",
      "NonRounding": 1
    }
  ]
}
```

Rounding Rules

- <http://example.com/Admin/Rounding>

```
{
  "success": true,
  "total": 1,
  "records": [
    {
      "pctRound": ""
    }
  ]
}
```