**CDMR Technical Design**

**Version 1.0**

|  |  |  |  |
| --- | --- | --- | --- |
| **Technical Design – <name of design>** | | | |
| Process Category |  | Start Date | 12/05/2016 |
| Author | Siva Sajjala | End Date |  |
|  |  | Deliverable ID |  |
|  |  |  |  |

Modification History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Section** | **Description** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Contents**

[1 Executive Summary 3](#_Toc468885285)

[1.1 Overview 3](#_Toc468885286)

[1.2 Requirements/Design 3](#_Toc468885287)

[1.3 Out of scope 3](#_Toc468885288)

[1.4 Process Models 4](#_Toc468885289)

[1.5 Source System(s) & Target Systems 5](#_Toc468885290)

[2 Technical & Component Summary 5](#_Toc468885291)

[2.1 Runtime Architecture Overview 6](#_Toc468885292)

[2.2 Integration Services 7](#_Toc468885293)

[2.3 UI components: 7](#_Toc468885294)

[2.4 Task Engine & Tasks 7](#_Toc468885295)

[2.5 CDMR Database 7](#_Toc468885296)

[2.6 Logging 8](#_Toc468885297)

[3 Components 8](#_Toc468885298)

[3.1 Customer Lookup API: 8](#_Toc468885299)

[3.2 Important Services 9](#_Toc468885300)

[3.3 Important JSP pages: 11](#_Toc468885301)

[3.4 Important Documents 12](#_Toc468885302)

[3.5 Database Tables 13](#_Toc468885303)

[3.6 Data Model 14](#_Toc468885304)

[3.7 Configuration Files 14](#_Toc468885305)

[4 Error Parameters 15](#_Toc468885306)

# Executive Summary

## Overview

The purpose of the project is to automate the approval work flow for CDMR (Credit Debit Request Memo) application using Java EE platform and provide higher financial control, audit capabilities.

At higher level, CDMR includes the following features.

* Hierarchy based on defined roles, assigned and maintained by administrator.
* Worked flow designed based on sales approval
* Allows Approvers to tag the requisitions with proper reason codes.
* Provides inbox facility for approvers to list all tasks waiting for their approvals
* Search facility for all CDMR users to perform search based on customer criteria

## Requirements/Design

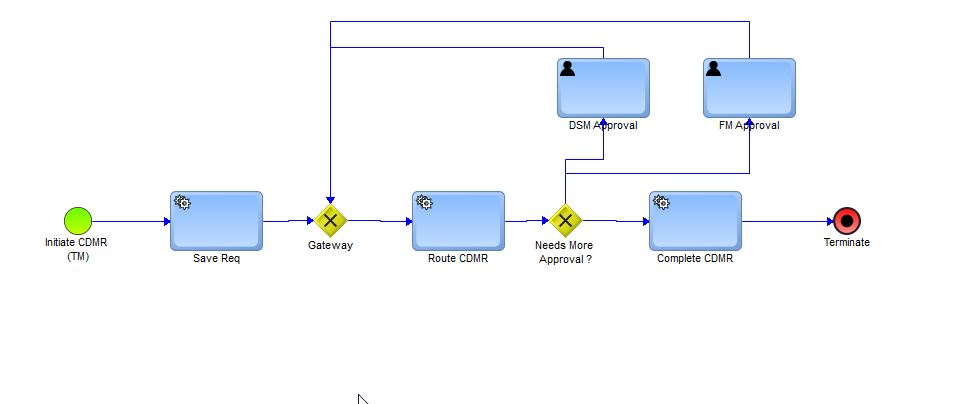
This document will contain information on the on process work flow and UI portions of the CDMR solution. The CDMR solution contains Java UI, Process/Task management and MySQL database layers.

## Out of scope

Providing Admin functionality to manage the CDMR requestors and Approvers list in the application is considered out of scope for this release. This feature will be considered as part of future releases along with other business enhancements.Functional Summary

## Process Models

Main work flow:



## Source System(s) & Target Systems

The CDMR BPM solution interacts with the below source and target systems to complete the CDMR process.

1. **CDMR UI –** Built in on the Java UI platform, users log into this system to input new CDMR requisitions, save the updates, Search the requisitions, or respond to tasks assigned to them by the BPM Process. These interactions in the UI layer are the “source” of the majority of activity in the CDMR solution
2. **Inbox**– Inbox functionality is used to display all CDMR tasks assigned to an approver (DSM & FM).
3. **Process Management –** This functionality is built using Java 1.8 platform to save, update and route the requisitions, manage the tasks. This is really the backbone for the entire application which keeps track of the CDMRs that are filed by the sales department and also make sure they are going thru different levels of approvals before it is finally posted to customer.

1. **Customer Lookup** – In house REST API is built to lookup the customer details from MySQL database based on the user input from UI. Product lookup web service in webMethods are used to access the PRSIM system to get product information that is needed by the UI.

# Technical & Component Summary

This section explains the tools, techniques, and approaches used to construct the CDMR solution. For information on the specific components built, see the “Components” section of this document.

## Runtime Architecture Overview



## Integration Services

Service layer is built using Java EE platform. All core processing services use hibernate framework to access the database objects. Tomcat hosts these services and execute. Main functionalities include routing, process approvals, search, inbox etc.

## UI components:

There are mainly 5 UI pages that users interact with. These are built using JSP, HTML technology. Users logs in to create CDMR page to initiate the requests, Approvers use the inbox and CDMR details pages to approve/reject a CDMR and All other viewers get to know the status of requisitions using search facility. Java script and Jquery is used for form validations. All UI interactions communicate with service layer thru servelets.

## Task Engine & Tasks

Main services for this functionality are QueueTask and UpdateTask. Assigns a unique number when a user gets assigned with a task. This task is referred on approver’s inbox page.

## CDMR Database

To meet the requirements of the CDMR solution, a database schema was created to hold the contents of each requisition’s customer, invoice and adjustment details as it was processed through the Process Model. It is necessary to maintain the data in custom database table for CDMR solution for the following reasons:

1. Search-ability – Customer, Invoice and adjustment details for a requisition needs to be searched every time users use the application. Due to this reason, custom database tables are designed to store customers, invoices and adjustment details, status information.
2. Future proof –
   1. Reporting can be built on top the database layer which gives the user to generate any financial reports or adhoc reporting.
   2. Having the CDMR data in a well formed database gives ultimate control on archiving and porting the data in the future.
   3. For details on CDMR tables used to drive functionality, refer to section 3.4.

## Logging

Logging is done thru 2 ways in the CDMR solution.

1. Server – Tomcat EE is used as the web app container. By using tomcat container most of the server logging is automatically taken care by Tomcat. Any server related errors/exceptions/warning and information is written at the following logs.

/opt/tomcat/logs/catalina.out

/opt/tomcat/logs/localhost\_access\_log.txt

1. App Log: As the review process for a division is progressing, all important events are logged in application log file ‘cdmr.log’ using log4J framework. Log file is stored at /opt/tomcat/logs/cdmr.log folder.

# Components

## Customer Lookup API:

This is REST based web service created to get the customer details from CDMR database. Input is customer number, given as parameter for the web service URL. Service is able to provide the response in either JSON or XML format.

Service URI: <http://138.68.9.11:8080/cdmr/rest/customer>

GET: [http://138.68.9.11:8080/cdmr/rest/customer/{param}/](http://138.68.9.11:8080/cdmr/rest/customer/%7bparam%7d/)

Example: <http://138.68.9.11:8080/cdmr/rest/customer/1000/>

Customer Result:

JSON Response:

{

"add1": "100",

"add2": "Manpower Place",

"city": "Milwaukee",

"custName": "Experis",

"custNum": 1000,

"phone": "414-961-1000",

"state": "WI",

"zip": "53212"

}

XML output:

<customer>

<add1>100</add1>

<add2>Manpower Place</add2>

<city>Milwaukee</city>

<custName>Experis</custName>

< custNum>1000</custNum>

<phone>414-961-1000</phone>

<state>WI</state>

<zip>53212</zip>

</customer>

For more details on HTTP error codes, please refer to the document on github repository.

<https://github.com/SSAJJALA/CreditDebitRequest/blob/master/CustomerLookup.md>

## Important Services

| Service | Notes |
| --- | --- |
| com.cdmr.approval.RouteCDMR | RouteCDMR program is the routing engine for CDMR application. This class identifies the next approver and assigns the task in the approver's inbox. |
| com.cdmr.controller.CDMRDetailsServlet | CDMRDetailsServlet controller servlet for cdmrDetails.jsp page. Take care of approvals and rejections from UI |
| com.cdmr.controller.CreateCDMRServlet | CreateCDMRServlet controller servlet for createCdmr.jsp page. Takes care of customer lookup, invoice lookup, submit and calculate action buttons from UI.. |
| com.cdmr.controller.InboxServlet | InboxServlet controller servlet for inbox.jsp page. Gets the cdmr details for a task (on click) and directs to details page. |
| com.cdmr.controller.SearchServlet | SearchServlet is the controller servlet for search.jsp. Gets the cdmr details for a requisition id (on click) and directs to details page. |
| com.cdmr.requisition.GetRequisition | GetRequisition prepares the cdmr requisition details based on requisition ID. |
| com.cdmr.requisition.SaveRequisition | SaveRequisition class saves a CDMR requisition to the database. REQUISITION, CDMR, CDMR\_ADJUSTMENT, COMMENT tables gets updated. |
| com.cdmr.requisition.UpdateRequisition | UpdateRequisition class updates an existing CDMR requisition to the database. REQUISITION, CDMR, CDMR\_ADJUSTMENT, COMMENT tables gets updated. |
| com.cdmr.Task.QueueTask | QueueTask generates the task ID for approval. This class is to create new unique number for task id and also assigns this new task to an user. Database tables involved in this class are TASK AND TASK\_ASSIGNMENT. |
| com.cdmr.Task.UpdateTask | UpdateTask updates the task status. |
| com.cdmr.ui.CalculateCDMR | This is the main service before a CDMR gets created. User hits the ‘Calculate’ button on UI and that action invokes this service call. Based on the invoice and adjustment details, it calculates the CDMR net adjustment amount including tax, allowance and charges. |
| com.cdmr.ui.CompleteTask | CompleteTask closes an active task upon approval from user. |
| com.cdmr.ui.GetCDMRDetails | This gets the details of a CDMR based on either requisition id or task id. |
| com.cdmr.ui.GetInbox | Pull all tasks for an user that are active. |
| com.cdmr.ui.Search | Search facility gets the cdmr details based on the search criteria that user is going to select on UI. It can search based on requisition id or customer name or status or all. Basically it formulates the dynamic query and runs this query on CDMR database to fetch the results. |
| com.cdmr.ui.SubmitCDMR | After calculation is done, this is the immediate next functionality. Submit action button calls this service. Saves the details to the database, generates an unique id (called Requisition id). This requisition id is the key for entire life cycle of that CDMR. |
| com.cdmr.ui.ValidateUser | Validates the user against CDMR\_USER table. |
| com.cdmr.webservices.CustomerLookupConsumer | This is the web service consumer to get the customer details from REST API (CustomerLookup). Maps the JSON response to ‘Customer’ object. |
| com.cdmr.webservices.InvoiceLookup | Service to get the invoice details from INVOICE and INVOICE\_DETAIL table. |

## Important JSP pages:

|  |  |
| --- | --- |
| JSP Page | Notes |
| cdmrDetails.jsp | UI page allows the CDMR users to display the details of a requisition. Approvers can either approve or reject the cdmr. Other users can  just view the details of a requisition. |
| createCdmr.jsp | This page allows the users to create a new CDMR requisition and submit for approvals. |
| inbox.jsp | Allows approvers to check the list of tasks waiting for their action. Navigates to CDMR details page upon clicking the task id. |
| index.jsp | This is the main landing page for CDMR application after authorised user logs in to the application. Displays all navigation links. |
| search.jsp | This gives the users to search for CDMRs based on 4 search fields. Filter by Requisition id or status or customer name or all.  Searches the CDMR database with the specified criteria and display them along with status. |

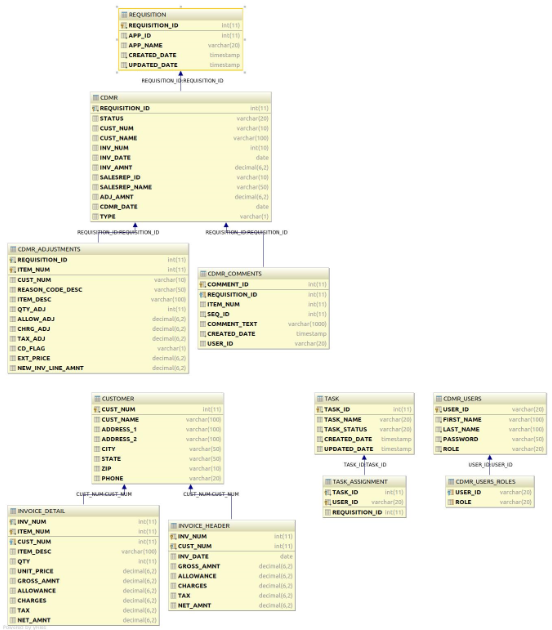
## Important Documents

|  |  |
| --- | --- |
| Document | Notes |
| com.cdmr.Data.CDMR | The main document structure that contains all the information about a particular CDMR request and this is the document that drives the whole CDMR process |
| com.cdmr.Data.Comment | Stores the comments by the user |
| com.cdmr.Data.CDMRAdjustments | Contains the adjustment data |
| com.cdmr.Data.Customer | Contains customer information. |
| com.cdmr.Data.InvoiceDetails | Contains invoice details. |
| com.cdmr.Data.InvoiceHeader | Invoice header details. |
| com.cdmr.Data.TaskResponse | Response of the task by an approver |
| com.cdmr.Data.UiAdjData | Used for capturing and manipulating adjustments on UI |

## Database Tables

|  |  |
| --- | --- |
| Table | Notes |
| REQUISITION | The highest level table in the CDMR schema and is a solution-agnostic table (meaning it’s not just for CDMR). This table essentially contains just the ReqID and minimal ancillary information that applies to any solution that uses this table. |
| CDMR | It essentially contains the CDMR header information. |
| CDMR\_ADJUSTMENTS | A one-to-many child table to the CDMR table which contains all adjustment details. Reason codes, Extended price, allowance adjustment amount, charges etc are important fields. |
| CDMR\_COMMENTS | One-to-many child table to the CDMR table which stores the comments put in by the user. |
| CDMR\_USERS | Users table for storing the CDMR user ids and names. |
| CDMR\_USER\_ROLES | Role table to specify the roles for each of the users in CDMR\_USER table. |
| CUSTOMER | Contains customer data. |
| INVOICE\_HEADER | Contains invoice header data. |
| INVOICE\_DETAIL | Contains invoice details data. |
| TASK | Stores task details. |
| TASK\_ASSIGNMENT | Stores task assignments. |

## Data Model

******

## Configuration Files

All files are located in the /opt/tomcat/webapps/cdmr/WEB-INF/classes directory

|  |  |
| --- | --- |
| Config File | Notes |
| cdmr.properties | Contains CDMR specific configuration parameters. Includes config parameters customer lookup web service URL, error codes, UI button actions etc. |

# Error Parameters

The following table lists the error parameters that are used to either display online or use it in services when an error occurs.

|  |  |
| --- | --- |
| Error Parameter | Error Origin and Description |
| customerLookupError\_01 | Customer number invalid |
| customerLookupError\_02 | Customer not found |
| customerLookupError\_03 | Unable to lookup customer due to technical issues |
| customerLookupError\_04 | Customer lookup webservice not available |
| customer\_02 | Customer not found |
| invoice\_01 | Invoice not found |
| invoice\_02 | Invalid invoice. Invoice details not found |
| validation\_01 | please enter valid adjustment qty/reason code/credit debit flag |