Watermaster Well Database
System Improvements - WWDIM
Design Specification

Version V1.1
Table of Contents

1 Introduction...........................................................................................................1
  1.1 Purpose ..............................................................................................................1
  1.2 Scope ............................................................................................................... 1
  1.3 Definitions, Acronyms and Abbreviations ....................................................... 1
  1.4 References and Supporting Documentations .................................................. 1

2 Design Model.......................................................................................................2
  2.1 Use Case Model ............................................................................................... 2
  2.2 Use Case Realization ....................................................................................... 3
  2.3 Class Diagram(s) ............................................................................................. 10
  2.4 Method listing / Internal interfaces .................................................................. 10

3 Logical Data Model ............................................................................................11
  3.1 ER Diagram ..................................................................................................... 11
  3.2 Data Dictionary ................................................................................................. 12
  3.3 Stored Procedures (SPs)/Services ................................................................... 16

4 Tools and Technologies .....................................................................................20
  4.1 Application Framework ................................................................................... 20
  4.2 Database Connection Pooling ......................................................................... 20

5 Error and Exception Handling ..........................................................................21

6 User Interface Development ............................................................................23

7 Access Control ..................................................................................................24

8 Special Logics and Calculations .......................................................................26

9 Appendices .........................................................................................................27
  9.1 Summary Production Report Template ............................................................ 27
## Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/06/2010</td>
<td>D0.1</td>
<td>Initial Draft</td>
<td>Kalani Tennakoon</td>
</tr>
<tr>
<td>10/13/2010</td>
<td>D0.2</td>
<td>• Incorporating internal review comments • Updating section 5.1.2</td>
<td>Kalani Tennakoon</td>
</tr>
<tr>
<td>10/14/2010</td>
<td>V1.0</td>
<td>• Adding more details to section 2.2.4 • Updating 3.2.2.1, 3.2.1.3, 8 and 9.1 sections according to updates received from MPWMD • Baseline version</td>
<td>Kalani Tennakoon</td>
</tr>
<tr>
<td>11/03/2010</td>
<td>V1.1</td>
<td>• Incorporating client feedback to sections 3.2.1.3 and 3.2.1.5</td>
<td>Kalani Tennakoon</td>
</tr>
</tbody>
</table>

## Reviewers

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Designation</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amali Kumuduni</td>
<td>D0.1</td>
<td>Senior Software Engineer</td>
<td>10/13/2010</td>
</tr>
<tr>
<td>Joe Oliver</td>
<td>V1.0</td>
<td>Water Resources Manager</td>
<td>10/29/2010</td>
</tr>
</tbody>
</table>

## Document Approvals

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MM/DD/YYYY</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Purpose
The purpose of this document is to present the detail design for enhancements to the Seaside Basin Watermaster Database. The Project including the enhancements is named as Watermaster Well Database System Improvements (WWDIM).

1.2 Scope
This document provides design solutions for the enhancements to the Watermaster system targeting only the areas that the enhancements affect upon. Target audience would be all the stakeholders of the project.

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Definitions

1.3.2 Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym / Abbreviation</th>
<th>Denotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRS</td>
<td>Software Requirement Specification</td>
</tr>
<tr>
<td>DB</td>
<td>Data Base</td>
</tr>
<tr>
<td>WWDIM</td>
<td>Watermaster Well Database System Improvements</td>
</tr>
<tr>
<td>MPWMD</td>
<td>Monterey Peninsula Water Management District</td>
</tr>
<tr>
<td>ESRI</td>
<td>Environmental Systems Research Institute</td>
</tr>
<tr>
<td>SFR</td>
<td>Software Functional Requirement</td>
</tr>
<tr>
<td>AJAX</td>
<td>Asynchronous JavaScript and XML</td>
</tr>
<tr>
<td>RIA</td>
<td>Rich Internet Application</td>
</tr>
</tbody>
</table>

1.4 References and Supporting Documentations

<table>
<thead>
<tr>
<th>Document</th>
<th>Version</th>
<th>Author</th>
<th>Document Name/Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>WWDIM Software Requirements Specification</td>
<td>D0.2</td>
<td>Zone24x7</td>
<td><a href="https://svn.zone24x7.lk/svn/WWDIM/tags/Project%20Phases/Requirements/SR-S-RFC/Draft/">https://svn.zone24x7.lk/svn/WWDIM/tags/Project%20Phases/Requirements/SR-S-RFC/Draft/</a></td>
</tr>
</tbody>
</table>
2 Design Model

2.1 Use Case Model

Related to the enhancements considered during this project following can be identified as the various use case models.
2.2 Use Case Realization

2.2.1 Updating data screens

When updating data screens, current system uses the FormView control. The enhancements on this design requires following the same procedure. The EditItemTemplate is used to update the editing scenario and InsertItemTemplate is used for the inserting scenario.

The SelectQuery and UpdateQuery properties of the existing SqlDataSource control are used to get the required value and to update the new value respectively.

2.2.2 Multi-select dropdown

A user control is created for the purpose of selecting multiple values by way of a dropdown consisting of checklist type items. This will be used in the “Well List” screen to perform the required filtering. Information exposed through this user control which is needed when integrating to a new page is listed below.

<table>
<thead>
<tr>
<th>Property/Event/Method</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallingPage</td>
<td>Page</td>
<td>The Page containing the user control</td>
</tr>
<tr>
<td>OnItemsSelected</td>
<td>MultiSelectDropDownDelegate</td>
<td>Event to handle the OnItemsSelected event of the control</td>
</tr>
<tr>
<td>DataSource</td>
<td>object</td>
<td>The DataTable containing</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>the required column names to be bound</td>
<td>string</td>
<td>The description column in data source being bound</td>
</tr>
<tr>
<td>DataTextField</td>
<td>string</td>
<td>The value column in data source being bound</td>
</tr>
<tr>
<td>AutoPostBack</td>
<td>bool</td>
<td>Whether or not a server post back is required after selection</td>
</tr>
<tr>
<td>DataBind</td>
<td>Method</td>
<td>Method to bind the data source</td>
</tr>
</tbody>
</table>

![Data Binding Diagram](image.png)
2.2.3 Adding new Reports

- A new report file should be added to the project

![Adding new item dialog box](image)

  - Add the `DataSource` from the “Website Data Sources” View. The created `TableAdapter` should use the “seaside_basinConnectionString” specified in `Web.config`.

![TableAdapter Configuration Wizard](image)

  - A new Web Form should be added which consists of a `ReportViewer` control and an `ObjectDataSource` control which maps to the `DataSource` created earlier. Sample `.aspx` code is as follows:
The code-behind file of this WebForm should have the Page_SaveStateComplete event implemented as follows, which handles the creation of pdf and excel files.

```csharp
protected void Page_SaveStateComplete(object sender, EventArgs e)
{
    Microsoft.Reporting.WebForms.Warning[] warnings = null;
    String outputType = "PDF";
    String ContentType = "application/pdf";
    string[] streamids = null;
    String mimeType = null;
    String encoding = null;
    String extension = null;
    Byte[] bytes = null;

    if (Request.QueryString["Output"] != null)
    {
```
if (Request.QueryString["Output"] != "1")
{
    outputType = "Excel";
    ContentType = "application/vnd.ms-excel";
}
}

bytes = ReportViewer1.LocalReport.Render(outputType, ",", out mimeType, out encoding, out extension, out streamids, out warnings);

Response.Clear();
Response.Buffer = true;
Response.ContentType = ContentType;
Response.BinaryWrite(bytes);
Response.End();
}

On page load event the following format could be followed to analyse the filter parameters and to load them to the report parameters.

protected void Page_Load(object sender, EventArgs e)
{
    if (Session["accesslevel"] == null)
    Response.Redirect("Login.aspx");

    if (!IsPostBack)
    {
        ReportParameter rpMainTitle = new ReportParameter();
        ReportParameter StartDate = new ReportParameter();
        ReportParameter EndDate = new ReportParameter();

        rpMainTitle.Name = "MainTitle";
        StartDate.Name = "StartDate";
        EndDate.Name = "EndDate";
        string ownername = "";
        if (Request.QueryString["ownerName"] != null)
        {
            ownername = " Well Owner: " + Request.QueryString["ownerName"];}
        else
        {
            ownername = " Well Owner: All Well Owner(s)";
        }

        if (Request.QueryString["WellName"] != null)
        {
            rpMainTitle.Values.Add("Well: " + Request.QueryString["WellName"] + ownername);
        }
        else
        {
            rpMainTitle.Values.Add("Well: All Well(s)" + ownername);
        }

        // Set the report parameters for the report
        ReportViewer1.LocalReport.SetParameters(new ReportParameter[] { rpMainTitle });
    
    if (Request.QueryString["StartDate"] != null)
{  
    StartDate.Values.Add(Request.QueryString["StartDate"]);  

    // Set the report parameters for the report  
    ReportViewer1.LocalReport.SetParameters(new  
    ReportParameter[] { StartDate });  
}

if (Request.QueryString["EndDate"] != null)  
{
    EndDate.Values.Add(Request.QueryString["EndDate"]);  

    // Set the report parameters for the report  
    ReportViewer1.LocalReport.SetParameters(new  
    ReportParameter[] { EndDate });  
}

}

2.2.4 Well information on Map

Adobe Flex is used to create the component to show the map of the “Well” locations. The Flex API provided by ArcGIS Resource Centers and the Samples and development guides provided by the same will be used for implementation. ([http://resources.esri.com/help/9.3/arcgisserver/apis/flex/samples/index.html](http://resources.esri.com/help/9.3/arcgisserver/apis/flex/samples/index.html))

ESRI services will be used to get the map of the required area. The esri:Map and esri:Locator controls and their associated functionality is used to get Map view and to mark locations respectively.
The `addressToLocations(addy, null, new AsyncResponder(onResult, onFault));` method of the `esri:Locator` control is used to mark the exact location of the “Well” which is given as the first parameter. The address is constructed from the following fields.

- Well Location – This will be considered as the street address (No. and street name)
- City – City the well resides
- State - CA

The functionality will be implemented to show the basic information of a “Well” on a mouse click event of that location on the map.

Code snippet:
```javascript
private function onWellClick(event:MouseEvent):void
{
    myMap.infoWindow.label = event.target.name;
    const mapPoint:MapPoint = MapPoint(event.target.geometry);
    const point : Point = myMap.toScreen( mapPoint );
    myMap.infoWindow.show(myMap.toMap(point));
}
```

The following information will be displayed:

- Address location
- Well Master ID
- Well ID and Well name
- Well status


2.3 Class Diagram(s)

The enhancement would be made on top of the existing class structure depicted below.

![Class Diagram](image)

2.4 Method listing / Internal interfaces

A new class will be added to the existing class structure to facilitate the need of logging to the event log during exceptions.

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logger Class</td>
<td>This class is used to log error, information, warning and other events that need to be logged by system. This will use system event log on the server to log these events.</td>
</tr>
</tbody>
</table>

![Logger Class Diagram](image)
3 Logical Data Model

3.1 ER Diagram

The enhancements will be incorporated targeting the current database structure which is as follows.
3.2 Data Dictionary

The following schema changes were done to the tables to facilitate the new improvements.

3.2.1 New tables (Schema and Meta data)

3.2.1.1 tb_well_status

Following table was related to SFR 3.1.2.1 in the referenced SRS document.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>WellStatusID</td>
<td>tinyint</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>varchar(20)</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WellStatusID</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Active</td>
</tr>
<tr>
<td>2</td>
<td>Inactive</td>
</tr>
<tr>
<td>3</td>
<td>Destroyed</td>
</tr>
<tr>
<td>4</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

3.2.1.2 tb_well_type

Following table is related to SFR 3.3.1 and SFR 3.3.5 in the referenced SRS document.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>WelleTypeID</td>
<td>tinyint</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>varchar(100)</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WellTypeID</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Other</td>
</tr>
<tr>
<td>3</td>
<td>Monitor</td>
</tr>
<tr>
<td>4</td>
<td>Producer</td>
</tr>
</tbody>
</table>

3.2.1.3 tb_ownerAdditionalData

Following table is related to SFR 3.3.3 in the referenced SRS document.

The meta-data were entered using the template provided for the Summary Production Report shown in section 9.1. The mapping for the said Producers in the Template, with the current DB Owner entries is given below.

<table>
<thead>
<tr>
<th>Producer specified in the Template</th>
<th>OwnerID on tb_ownercontractor table of the DB</th>
<th>Company Name on tb_ownercontractor table of the DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAW - Coastal Subareas</td>
<td>702</td>
<td>California American Water</td>
</tr>
<tr>
<td>City of Seaside (Municipal)</td>
<td>608</td>
<td>City of Seaside</td>
</tr>
<tr>
<td>Granite Rock Company</td>
<td>607</td>
<td>Granite Rock</td>
</tr>
<tr>
<td>DBO Development No. 27</td>
<td>750</td>
<td>DBO Development</td>
</tr>
<tr>
<td>City of Seaside (Golf Courses)</td>
<td>610</td>
<td>BSL Golf of California</td>
</tr>
<tr>
<td>Sand City</td>
<td>605</td>
<td>City of Sand City</td>
</tr>
<tr>
<td>SNG (Security National Guaranty)</td>
<td>711</td>
<td>Security National Guaranty Inc</td>
</tr>
<tr>
<td>Calabrese (Cypress Pacific Inv.)</td>
<td>604</td>
<td>King Venture</td>
</tr>
</tbody>
</table>
### 3.2.1.4 tb_CAWAquifer

Following table is related to SFR 3.3.3 in the referenced SRS document.

The meta-data were entered using the template provided for the Summary Production Report shown in section 9.1.

<table>
<thead>
<tr>
<th>OwnerID</th>
<th>SubAreaID</th>
<th>Type</th>
<th>BaseOverYieldAlloc</th>
<th>CarryOverFromPrevYr</th>
<th>WaterYear</th>
</tr>
</thead>
<tbody>
<tr>
<td>702</td>
<td>4</td>
<td>SPA</td>
<td>3086.7</td>
<td>495.9</td>
<td>2010</td>
</tr>
<tr>
<td>608</td>
<td>NULL</td>
<td>SPA</td>
<td>253.1</td>
<td>90.9</td>
<td>2010</td>
</tr>
<tr>
<td>607</td>
<td>NULL</td>
<td>SPA</td>
<td>23.8</td>
<td>192.6</td>
<td>2010</td>
</tr>
<tr>
<td>750</td>
<td>NULL</td>
<td>SPA</td>
<td>43.3</td>
<td>NULL</td>
<td>2010</td>
</tr>
<tr>
<td>610</td>
<td>NULL</td>
<td>APA</td>
<td>540</td>
<td>NULL</td>
<td>2010</td>
</tr>
<tr>
<td>605</td>
<td>NULL</td>
<td>APA</td>
<td>9</td>
<td>NULL</td>
<td>2010</td>
</tr>
<tr>
<td>711</td>
<td>NULL</td>
<td>APA</td>
<td>149</td>
<td>NULL</td>
<td>2010</td>
</tr>
<tr>
<td>604</td>
<td>NULL</td>
<td>APA</td>
<td>14</td>
<td>NULL</td>
<td>2010</td>
</tr>
<tr>
<td>749</td>
<td>NULL</td>
<td>APA</td>
<td>31</td>
<td>NULL</td>
<td>2010</td>
</tr>
<tr>
<td>702</td>
<td>2</td>
<td>SPA</td>
<td>246.1</td>
<td>NULL</td>
<td>2010</td>
</tr>
<tr>
<td>751</td>
<td>NULL</td>
<td>APA</td>
<td>251</td>
<td>NULL</td>
<td>2010</td>
</tr>
<tr>
<td>722</td>
<td>NULL</td>
<td>APA</td>
<td>320</td>
<td>NULL</td>
<td>2010</td>
</tr>
<tr>
<td>615</td>
<td>NULL</td>
<td>APA</td>
<td>32</td>
<td>NULL</td>
<td>2010</td>
</tr>
<tr>
<td>612</td>
<td>NULL</td>
<td>APA</td>
<td>41</td>
<td>NULL</td>
<td>2010</td>
</tr>
</tbody>
</table>
### 3.2.1.5 tb_WaterSystem

Following table is related to SFR 3.3.3 in the referenced SRS document.

<table>
<thead>
<tr>
<th>WellSystemID</th>
<th>WellSystemName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Ryan Ranch Unit</td>
</tr>
<tr>
<td>3</td>
<td>Hidden Hills Unit</td>
</tr>
<tr>
<td>4</td>
<td>Bishop Unit</td>
</tr>
<tr>
<td>5</td>
<td>Southern Coastal Subarea</td>
</tr>
<tr>
<td>6</td>
<td>Northern Coastal Subarea</td>
</tr>
<tr>
<td>7</td>
<td>Seaside – Municipal</td>
</tr>
<tr>
<td>8</td>
<td>Seaside – Golf Courses</td>
</tr>
</tbody>
</table>

### 3.2.2 Modifications to existing tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Field Name</th>
<th>Data Type</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tb_well</td>
<td>Well Status</td>
<td>TINYINT</td>
<td>4 (Unknown)</td>
<td>Status of the Well. Refers to tb_well_status.</td>
</tr>
<tr>
<td></td>
<td>MonthlyWaterLevelRequired</td>
<td>BIT</td>
<td>1</td>
<td>For use in Compliance Report</td>
</tr>
<tr>
<td></td>
<td>AnnualWaterQualityRequired</td>
<td>BIT</td>
<td>1</td>
<td>For use in Compliance Report</td>
</tr>
<tr>
<td></td>
<td>QuarterlyProductionRequired</td>
<td>BIT</td>
<td>1</td>
<td>For use in Compliance Report</td>
</tr>
</tbody>
</table>
### Design Specification

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>TINYINT</td>
<td>1 (Blank value)</td>
<td>For use in Compliance Report and Groundwater Level Monitoring Data Report. Refers to <code>tb_well_type</code>.</td>
</tr>
<tr>
<td><strong>Monitored_by</strong></td>
<td>VARCHAR(200)</td>
<td></td>
<td>For use in Groundwater Level Monitoring Data Report.</td>
</tr>
<tr>
<td><strong>WL_mon_freq</strong></td>
<td>VARCHAR(200)</td>
<td></td>
<td>For use in Groundwater Level Monitoring Data Report.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>tb_ownercontractor</strong></th>
<th><strong>Common Name</strong></th>
<th>VARCHAR(150)</th>
<th>Common name for a company</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tb_waterquality</strong></td>
<td><strong>Bicarbonate</strong></td>
<td>VARCHAR(50)</td>
<td>Data input for water quality</td>
</tr>
<tr>
<td></td>
<td><strong>Carbonate</strong></td>
<td>VARCHAR(50)</td>
<td>Data input for water quality</td>
</tr>
<tr>
<td></td>
<td><strong>Bicarbonate Less Than</strong></td>
<td>CHAR(1)</td>
<td>Data validation for Bicarbonate value</td>
</tr>
<tr>
<td></td>
<td><strong>Carbonate Less Than</strong></td>
<td>CHAR(1)</td>
<td>Data validation for Carbonate value</td>
</tr>
<tr>
<td><strong>WaterYear</strong></td>
<td>INT</td>
<td></td>
<td>To record the actual year that the water quality record is attributed to</td>
</tr>
</tbody>
</table>

| **tb_depth** | **ReportMonth** | DATETIME | To record the actual month that the water level record is attributed to |

#### 3.2.2.1 Updating existing contact data for 'Common Name' in 'tb_ownercontractor' table

<table>
<thead>
<tr>
<th>OwnerID</th>
<th>Company Name</th>
<th>CommonName</th>
</tr>
</thead>
<tbody>
<tr>
<td>604</td>
<td>King Venture</td>
<td>Calabrese</td>
</tr>
<tr>
<td>605</td>
<td>City of Sand City</td>
<td>Sand City</td>
</tr>
<tr>
<td>607</td>
<td>Granite Rock</td>
<td>Granite Rock Company</td>
</tr>
<tr>
<td>600</td>
<td>City of Seaside</td>
<td>Seaside - Municipal</td>
</tr>
<tr>
<td>610</td>
<td>BSL Golf of California</td>
<td>Seaside - Golf Courses</td>
</tr>
<tr>
<td>612</td>
<td>Monterey County Parks Department</td>
<td>Laguna Seca Recreation Area</td>
</tr>
<tr>
<td>615</td>
<td>York School</td>
<td>York School</td>
</tr>
<tr>
<td>702</td>
<td>California American Water</td>
<td>California American Water</td>
</tr>
<tr>
<td>711</td>
<td>Security National Guaranty Inc</td>
<td>SNG</td>
</tr>
<tr>
<td>722</td>
<td>Laguna Seca Ranch</td>
<td>Laguna Seca Golf</td>
</tr>
<tr>
<td>749</td>
<td>Mission Memorial Park</td>
<td>Mission Memorial</td>
</tr>
<tr>
<td>750</td>
<td>DBO Development</td>
<td>DBO Development (Target)</td>
</tr>
<tr>
<td>751</td>
<td>Pasadera Country Club, LLC</td>
<td>Pasadera</td>
</tr>
</tbody>
</table>
3.3 Stored Procedures (SPs)/Services

<table>
<thead>
<tr>
<th>Signature</th>
<th>Description</th>
<th>Procedural Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>spGetProductionData</td>
<td>Get Production Data</td>
<td>Stored Procedure to get production data for Production details screen</td>
</tr>
<tr>
<td>spGetComplianceReportData</td>
<td>Get Compliance Report data</td>
<td>Retrieves the information for the Compliance Report</td>
</tr>
<tr>
<td>spGetContactReportData</td>
<td>Get Contact Report Data</td>
<td>Retrieves the information for Contacts Report</td>
</tr>
<tr>
<td>spGetWaterLevelData</td>
<td>Get Groundwater Level monitoring Report Data</td>
<td>Retrieves the information for Groundwater Level monitoring Report</td>
</tr>
<tr>
<td>spGetSummaryProductionReportData</td>
<td>Get Summary Production Report Data</td>
<td>Retrieves the information for Summary Production Report Data</td>
</tr>
<tr>
<td>udfProductionDetail (existing function)</td>
<td>Modify the [udfProductionDetail] function</td>
<td>Function was modified to cater SFR 3.3.2 in the referenced SRS document</td>
</tr>
</tbody>
</table>

**General format of the SP Structures used for WWDIM:**

*This format should be used for the newly added SPs related to the improvements*

**Naming** (following the currently used format): ‘sp<FunctionalityDesc>’

Eg: spGetGroundwaterLevelMonitoringReportData

**Structure:**

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET NOCOUNT ON
GO

-- =============================================
-- Author:  Kalani Tennakoon
-- Create date: 02/17/2009
-- Description: Retrieving data for the Application Data step on the Appeals Module
-- =============================================

IF OBJECT_ID('dbo.spGetAppealsApplicationDataInfo') IS NOT NULL BEGIN
  DROP PROCEDURE dbo.spGetAppealsApplicationDataInfo
  IF OBJECT_ID('dbo.spGetAppealsApplicationDataInfo') IS NOT NULL
    PRINT '<< FAILED DROPPING PROCEDURE dbo.spGetAppealsApplicationDataInfo >>'
  ELSE
    PRINT '<< DROPPED PROCEDURE dbo.spGetAppealsApplicationDataInfo >>'
END
```
GO

CREATE PROCEDURE [dbo].[spGetAppealsApplicationDataInfo]
    @appealID int = NULL,
    @appeableDecID int = NULL
AS
BEGIN
    BEGIN TRY
        DECLARE @currentAppealableDecID int

        -- Get the Appealable Dec ID if the Appeal ID is given
        IF(@appealID IS NOT NULL)
            BEGIN
                SELECT @currentAppealableDecID = FkAppealableDecisionID
                FROM Appeals
                WHERE (AppealID = @appealID)
            END
        ELSE
            BEGIN
                SET @currentAppealableDecID = @appeableDecID
            END

        -- Get Applicant info and Appeal related other info
        IF(@appealID IS NOT NULL)
            BEGIN
                SELECT dbo.Agents.FirstName AS applicantFName,
                       dbo.Address.StreetNo as applicantStreetNo,
                       dbo.Agents.ContactNum as applicantContactNo,
                       dbo.Appeals.AppealDate as appealedDate
                FROM    dbo.Appeals
                        INNER JOIN dbo.Agents
                                ON dbo.Appeals.FkApplicantID = dbo.Agents.AgentID
                        INNER JOIN dbo.Address
                                ON dbo.Agents.FkAgentAddressID = dbo.Address.AddressID
                WHERE dbo.Appeals.AppealID = @appealID
            END
        END TRY
        BEGIN CATCH
            -- Raise an error with the details of the exception
            DECLARE @ErrMsg varchar(4000), @ErrSeverity int
            SELECT
                @ErrMsg   = ERROR_MESSAGE(),
                @ErrSeverity = ERROR_SEVERITY()
            RAISERROR(@ErrMsg, @ErrSeverity, 1);
        END CATCH
    END
END
IF OBJECT_ID('dbo.spGetAppealsApplicationDataInfo') IS NOT NULL
    PRINT '<<< CREATED PROCEDURE dbo.spGetAppealsApplicationDataInfo >>>'
ELSE
    PRINT '<<< FAILED CREATING PROCEDURE dbo.spGetAppealsApplicationDataInfo >>>'
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET NOCOUNT OFF
GO
GRANT EXECUTE ON [dbo].[spGetAppealsApplicationDataInfo] TO [WDDDBSUser]
GO

General format of the function Structures used for WWDIM:

*This format should be used for the newly added functions related to the improvements

    Naming (following the currently used format): ‘fn<Funcionality/ResultDesc>’

        Eg: fnProductionDetail

    Structure:

    -- =============================================
    -- Author:  Lakmal Molligoda
    -- Create date: 10-16-2007
    -- Description: return year list for report params.
    -- =============================================

IF OBJECT_ID('dbo.fnGetAPN') IS NOT NULL
BEGIN
    DROP FUNCTION dbo.fnGetAPN
    IF OBJECT_ID('dbo.fnGetAPN') IS NOT NULL
        PRINT '<<< FAILED DROPPING FUNCTION dbo.fnGetAPN >>>'
    ELSE
        PRINT '<<< DROPPED FUNCTION dbo.fnGetAPN >>>'
END
GO

CREATE FUNCTION [dbo].[fnGetAPN]
    (@inspectionType int,@PermitID int,@RebateID int,@ConservationID int,@WUCID int)
RETURNS varchar(50) AS
BEGIN
    DECLARE @apn varchar(50)
    IF(@inspectionType=1)
        BEGIN
            -- Additional code here
        END
SELECT @apn=FkAPNNum FROM Permits WHERE PermitID=@PermitID
END
RETURN @apn
END

GO
IF OBJECT_ID('dbo.fnGetAPN') IS NOT NULL
    PRINT '<<< CREATED FUNCTION dbo.fnGetAPN >>>'
ELSE
    PRINT '<<< FAILED CREATING FUNCTION dbo.fnGetAPN >>>'
GO

GRANT EXECUTE ON fnBuildAddress TO WDDDBSUser
4 Tools and Technologies

4.1 Application Framework

<table>
<thead>
<tr>
<th>Product</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS Visual Studio 2005</td>
<td>Used as the development environment</td>
</tr>
<tr>
<td>SQL 2005 Server Express Edition</td>
<td>Used for all database functionalities</td>
</tr>
<tr>
<td>SQL Reporting Services</td>
<td>Used for report generation</td>
</tr>
<tr>
<td>IIS 6</td>
<td>Used for hosting the web services used by the system</td>
</tr>
<tr>
<td>Adobe Flex</td>
<td>For the map showing functionality</td>
</tr>
<tr>
<td>AJAX</td>
<td>Used for RIA development, for date selection capabilities</td>
</tr>
</tbody>
</table>

The application will be developed on top of the existing code base which is developed as a “Web Site” type of project.

4.2 Database Connection Pooling

The primary data storage for the WWDIM is a MS-SQL Express 2005 Database Server.
5 Error and Exception Handling

5.1.1 Error Handling

For client side expression/required field validations, the error message will be shown next to the validation field.

Validation error message formats:

- For required field: “Required”
- For data type validation: “Enter a number”

When validation groups and summary tools are used on the web interface side, in the button action it should be validated to check whether the page is valid.

For Example:

```csharp
Page.Validate();
if (Page.IsValid)
{
}
```

This will ensure that the user action would not pass unless page is valid.

To check the session validity following code should be used as in the existing system. This will ensure that the user session is valid or not if not he will be redirected to login page.

Code snippet:

```csharp
if (Session["accesslevel"] == null)
    Response.Redirect("Login.aspx");
```

5.1.2 Exception Handling

All the application exceptions are written to event log on the system. This is done through Logger class.

On event handling functions need to add try catch block and catch the exception and log it to the application log using logger class.

For Example:

```csharp
protected void Page_Load(object sender, EventArgs e)
{
    try
    {
    
    }
    catch (Exception ex)
    {
        Logger.LogErrorEvent(ex);
        throw ex;
    }
}
```
On an exception the existing design will be then used where the user is redirected to an error page when “customErrors” setting in web.config is “On”.

- **GenericErrorPage.htm** - The generic error page will display a generic error message as “Application error occurred. Please try again in a few minutes.” and have a link to redirect user to the Login page.
- **NoAccess.htm** – This page will display an error message as “Application error occurred. Do not have access to this resource file.” and have a link to redirect user to the Login page.
- **FileNotFound.htm** - This page will display an error message as “Application error occurred. Resource file not found.” and have a link to redirect user to the Login page.

```xml
<customErrors mode="On" defaultRedirect="GenericErrorPage.htm">
    <error statusCode="403" redirect="NoAccess.htm"/>
    <error statusCode="404" redirect="FileNotFound.htm"/>
</customErrors>
```

### 5.1.3 Logging and Notifications

Following the current structure, notifications would be considered as User messages displayed on the User Screen. Logging will be used in Exception Handling as mentioned above.
6 User Interface Development

6.1.1 UI Framework/Standards
The enhancements to the screens would be done on top of the same screens. For example the “Well Status” addition to the “Well Details” screen would be seen as follows.

6.1.2 Reports and Charts Formats
The new reports and filter screens related to the same would be similar to the reports being used currently.
7 Access Control

7.1.1 Security Levels
The users will be validated to have authorized privileges. This is built-in to the current system.

7.1.2 User Roles/Groups
Currently the system is defined with user accessing levels as 1, 2, 3 and 4 where level 4 is granted with highest privileges of accessing the system. The enhancements will be done considering these rules, specifying the permissions requested by client, depicted in section 2.1.

7.1.3 Security Configuration Details
To configure the role accessibilities, the following code should be used. The first part prevents the Level 1 user from at least viewing this page while the method call allows the functionality to be modified for different “levels” appropriately. Same design would be followed.

```csharp
protected void Page_Load(object sender, EventArgs e)
{
    if (Session["accesslevel"] == null ||
        Session["accesslevel"].ToString() == "1")
        Response.Redirect("Login.aspx");

    if (Session["accesslevel"] != null)
    {
        ConfigurePageForAccessLevel();
    }

    //User Code
}

private void ConfigurePageForAccessLevel()
{
    if (Session["accesslevel"] != null)
    {
        int accessLevel = int.Parse(Session["accesslevel"].ToString());

        switch (accessLevel)
        {
            case 1:
            case 2:
            default:
                HyperLink1.Visible = false;
                break;
        }
    }
```
case 3:
case 4:
    HyperLink1.Visible = true;
    break;

else
{
    Response.Redirect("Login.aspx");
}
}
8 Special Logics and Calculations

- Where required data should be converted utilizing the relationships:
  
  325,851 gallons = 43,560 cubic feet = 1 acre-foot

- In the enhanced system the subareas will be filtered to show only the below mentioned 4 subareas.
  1. Northern Coastal
  2. Southern Coastal
  3. Northern Inland
  4. Laguna Seca

  Note:
  o The current wells assigned as 'LSS' will be considered as 'Laguna Seca'
  o The current wells assigned as 'SCS' will be considered as 'Southern Coastal'
  o The current wells assigned as 'Dunes' will be assigned as empty
9 Appendices

9.1 Summary Production Report Template

*Note:
- Only the Water master Producers and the related Wells with status ‘Active’ will be considered in this report.
- The category ‘Coastal Subareas’ includes both ‘Northern Coastal’ and ‘Southern Coastal’ subareas.