Software Requirements Specification

for

Drawdown Interface (DDI)

Version 1.0 approved

Prepared by
Pranil Dahal
Iovanni Enriquez
Wesley Hwang
Cesar Rojas
Vay Tang

LA County Department of Regional Planning

March 8th, 2019
# Table of Contents

Table of Contents ........................................................................................................................................ 2  
Revision History ........................................................................................................................................ 3  
1. Introduction ............................................................................................................................................. 4  
   1.1. Purpose ........................................................................................................................................... 4  
   1.2. Intended Audience and Reading Suggestions .............................................................................. 4  
   1.3. Product Scope ................................................................................................................................. 5  
   1.4. Definitions, Acronyms, and Abbreviations .................................................................................... 5  
   1.5. References ..................................................................................................................................... 6  
2. Overall Description ................................................................................................................................. 7  
   2.1. Product Perspective ......................................................................................................................... 7  
   2.2. Product Functions ............................................................................................................................ 8  
   2.3. User Classes and Characteristics ................................................................................................. 8  
   2.4. Operating Environment ................................................................................................................. 9  
   2.5. Design and Implementation Constraints ...................................................................................... 9  
   2.6. User Documentation ...................................................................................................................... 9  
   2.7. Assumptions and Dependencies ................................................................................................. 10  
   2.8. Apportioning of Requirements .................................................................................................. 10  
3. External Interface Requirements .......................................................................................................... 11  
   3.1. User Interfaces ............................................................................................................................... 11  
   3.2. Hardware Interfaces ....................................................................................................................... 13  
   3.3. Software Interfaces ........................................................................................................................ 13  
   3.4. Communications Interfaces ......................................................................................................... 13  
4. Requirements Specification ..................................................................................................................... 14  
   4.1. Functional Requirements .............................................................................................................. 14  
   4.2. External Interface Requirements ................................................................................................ 16  
   4.3. Logical Database Requirements .................................................................................................. 16  
   4.4. Design Constraints ....................................................................................................................... 16  
5. Other Nonfunctional Requirements ..................................................................................................... 17  
   5.1. Performance Requirements ........................................................................................................ 17  
   5.2. Safety Requirements ..................................................................................................................... 17  
   5.3. Security Requirements .................................................................................................................. 17  
   5.4. Software Quality Attributes ....................................................................................................... 17  
   5.5. Business Rules ............................................................................................................................. 18  
6. Other Requirements ............................................................................................................................... 19  
Appendix A: Glossary ................................................................................................................................. 20  
Appendix B: Analysis Models .................................................................................................................. 21  
Appendix C: To Be Determined List ....................................................................................................... 22
## Revision History

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Reason For Changes</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iovanni</td>
<td>3/8/2019</td>
<td>Document new APIs</td>
<td>V2</td>
</tr>
<tr>
<td>Cesar</td>
<td>3/8/2019</td>
<td>Document new Components</td>
<td>V2</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Purpose

The purpose of this document is to specify all the requirements of the software Drawdown Interface, referred to as DDI.

This is the Version 1 of this document. It doesn’t include the complete set of requirements.

The missing requirements will be added, and the versions of the documents will be updated accordingly on page 3.

1.2 Intended Audience and Reading Suggestions

This document is intended for project managers, developers, testers, system administrators, and documentation writers. Users of the DDI will have no need to read this document.

This document contains all the requirements of the DDI, categorized into the following types:
   A. External Interface Requirements
   B. Functional Requirements
   C. Non-Functional Requirements

UI developers of this project are recommended to look at the External Interface Requirements (Section 3).

Backend developers are recommended to look at the Functional Requirements (Section 4).

System Administrators are recommended to look at the Non-Functional Requirements (Section 5).

Documentation writers will need to look at the table of contents (Page 2) to determine which section contains the information they seek.
1.3 Product Scope

The name of the product described in this document is Drawdown Interface (DDI). DDI will centralize all data that is related to CUP Deposit Accounts at LA County Department of Regional Planning. DDI will provide a web application that will allow users to access said data.

DDI provides a read-only type of access to the Deposit Accounts data. DDI will not let users edit and delete the data.

Once released, DDI will be used primarily by employees at LA County Department of Regional Planning to see details about plans, deposit accounts, transaction history on deposit accounts and other details in order to perform zoning enforcement related tasks.

1.4 Definitions, Acronyms, and Abbreviations

Plan - a case that is created to track any development that is performed on a property

Deposit Account - is an account that a customer deposits a predetermined amount of money in so that planners can perform work on a property

CUP - Conditional Use Permit
REST - Representational State Transfer
API - Application Programming Interface
HTTP - Hypertext Transfer Protocol
HRHM - HTTP Request Handler Module
TCUP - Testing CUP
TAM - Trend Analysis Module
LAD - Los Angeles Database
JSON - JavaScript Object Notation
PPM - Parameter Processing Module
UI - User Interface
DDI - Drawdown Interface
SSO - Single Sign On
1.5 References
All references used in the creation of this document are listed below.

1.5.1 Controlling Documents
1) There is no document controlling this document.

1.5.2 Applicable Documents
1) Template provided in CS-4961 was used as basic structure and layout for this document.
2) No additional applicable document has been used in the production of this document.

1.5.3 Standards
No Standard has been used in the creation of this document.
2. Overall Description

The Los Angeles County Department of Regional Planning does not have an application that can centralize all data that is related to Deposit Accounts. Because of this, ad hoc reports are constantly being created to track these accounts. DDI will allow users to run custom reports on deposit data, as well as provide as mechanism to drill into more specific data about the respective plans, case, and account.

2.1 Product Perspective

DDI is an interface that receives all of its data from the deposit account database managed by LA County Department of Regional Planning.

For DDI to function properly, said database must be running, and a TCP port must be opened for DDI to latch on to.

All the functional requirements in Section 4 of this document are depending on contingencies that the database:
   A. exists
   B. has a TCP port open for connections
   C. has user credentials created specifically for DDI

The diagram below shows how DDI is directly dependent on the Deposit Account Database.

![Diagram of DDI and Deposit Account Database](image_url)
2.2  Product Functions

DDI will allow users to perform four major functions:

I. Individual Balance by Plan - DDI will list all the accounts categorized by plan. Users can select a plan, and view details about that specific plan. The details will include all the accounts related to the plan, the contact information for each of those accounts, transaction history for each account, etc.

II. Individual Balance by Contact - DDI will list all the accounts organized by contact. For each contact, users can view all the related cases, account balances, transaction history, etc.

III. Non-Matching Account Alert - All account numbers must match a case number. There are entries in the LA County Database where this constraint is not satisfied. DDI will filter out those accounts and allow users to send an email to the respective contact for each account, notifying them to change their information in the database to satisfy this constraint.

IV. All Account Balances - DDI will also list all the accounts and the balances in the database, with various filters. Users can categorize the accounts based on LA County Regions, transactions dates, contact names, etc.

2.3  User Classes and Characteristics

The users for this product will be the employees at LA County Department of Regional Planning, specifically the following sub-divisions:

I. Budget and Accounting
II. Zoning Enforcement
III. System Analysis

Users will not be classified into classes, since all the subdivisions will be using the DDI for all its functions.
2.4 Operating Environment

DDI will run on a server inside the LA County network. It can only be accessed from a client connected to the LA County network. DDI will require:

I. Java 8 to be installed on the server
II. TCP port 80 to be open for HTTP connections
III. Deposit Account Database to be live

2.5 Design and Implementation Constraints

For software developers to implement this product, they must have a local instance of Sql Server Express 2007 in their machine. The purpose of this local database is to replicate the server environment that DDI will be deployed on.

Creation of new acronyms must be authorized by the liaison from LA County, since there are policies that must be taken into consideration while creating acronyms.

2.6 User Documentation

As of now, there are no plans for any user documentation that will be delivered along with the software.

This document will be updated in the future in case any user documentation gets added to the deliverables.
## 2.7 Assumptions and Dependencies

DDI requirements are created based on the assumption that the clients will have Internet Explorer browser version GREATER THAN 8.

DDI user authentication has been designed to fit La County employee LDAP. Any changes in the authentication process will affect the requirements in the future.

DDI uses mssql-jdbc 7.0.0.jre8 version, provided by com.microsoft.sqlserver. Any changes in the version of the Deposit Account database will affect the requirements.

### 2.8 Apportioning of Requirements

I. Requirements for DDI RestAssured test cases will be created after the completions of all the REST APIs.

II. Requirements for Trend Analysis Module will be created after the completion of all the REST APIs.
3. **External Interface Requirements**

This section of the document specifies all the external interfaces of DDI, specifically the web app UI, hardware interfaces, software interfaces and communication interfaces.

### 3.1 User Interfaces

DDI provides a web application to view the four major reports on the home page:

I. All Balances Report
II. Individual Balance by Plans
III. Individual Balance by Contact
IV. Non-Matching Accounts
V. Analysis Tool

All Balances Report page provides user with different filtering options. Users can categorize accounts based on regions, or account/plan/contact names.
Non-Matching Accounts page displays a list of accounts whose account number don’t match a case number. The email on the far right will be a hyperlink and can be used to send an email notification to the person that last accessed the account information, and any other additional people that the user wants added on the list of recipients.

<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
<th>First Name</th>
<th>Last Name</th>
<th>Last Changed On</th>
<th>Last Changed By</th>
<th>Type Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCLP-CF04-114-29254</td>
<td>R. D. GARY</td>
<td></td>
<td></td>
<td>2016-02-10</td>
<td>Anna Yang</td>
<td>Condition Check</td>
<td>eyang@planning seafood.gov</td>
</tr>
<tr>
<td>RCLP-CF01-197-27252</td>
<td>Ryan Shroids</td>
<td></td>
<td></td>
<td>2016-06-13</td>
<td>System Administrator</td>
<td>Condition Check</td>
<td>epk@<a href="mailto:help@seacounty.gov">help@seacounty.gov</a></td>
</tr>
<tr>
<td>RCLP-CF01-776-27065</td>
<td>TOWERICO</td>
<td></td>
<td></td>
<td>2016-07-18</td>
<td>System Administrator</td>
<td>Condition Check</td>
<td>epk@<a href="mailto:help@seacounty.gov">help@seacounty.gov</a></td>
</tr>
</tbody>
</table>

Individual Balances By Contact page displays all the balances of accounts listed in the order by contact. There is a search bar on the top right that allows you to find specific balances with wildcard matching contact names.

Individual Balances by Plan page is extremely similar to the above contact page. The only key difference is that it lists the balances in order of plan number and can be searched by plan as well.
3.2 **Hardware Interfaces**

DDI doesn’t have any hardware interfaces.

3.3 **Software Interfaces**

I. DDI is designed to directly connect to an instance of Sql Server Express 2007 database, with the use of the following connector - com.microsoft.sqlserver - mssql-jdbc - Version 7.0.0.jre8.

3.4 **Communications Interfaces**

I. The data passed from the Backend REST API to Frontend is structured in JSON arrays.
II. The web application provided by DDI is meant to be run on web browsers.
III. DDI provides e-mail interface to send notifications from the Non-Matching Accounts page.
IV. DDI’s Frontend components use HTTP.GET calls to retrieve data from the Backend.
V. DDI provides the user with exporting an entire report to an excel sheet
4. Requirements Specification

4.1 Functional Requirements

This Section collects all DDI Functional Requirements. The Section includes the complete set of functional requirements with explanation and rationale where the statement of the requirement was deemed insufficient or needing additional background/justification.

The in-depth design of the modules mentioned here can be found in the DDI Software Design Document.

An effort has been made to standardize the correlation between the design modules and the requirements to make their access and organization more consistent. Module 2.1 requirements are labeled 3.1, sub-module 2.1.1 requirements are labeled 3.1.1 and so on. The list of requirements are as follows.

<table>
<thead>
<tr>
<th>Requirement No.</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Access Module (DAM) 2.1.1 Requirements</strong></td>
<td></td>
</tr>
<tr>
<td>3.1.1-1</td>
<td>The DAM shall query the LAD</td>
</tr>
<tr>
<td>3.1.1-2</td>
<td>The DAM shall receive data returned by the LAD</td>
</tr>
<tr>
<td>3.1.1-3</td>
<td>The DAM shall accept requests for data from HRHM and TAM</td>
</tr>
<tr>
<td>3.1.1-4</td>
<td>The DAM shall return data to the HRHM</td>
</tr>
<tr>
<td><strong>HTTP Request Handle Module (HRHM) 2.1.2 Requirements</strong></td>
<td></td>
</tr>
<tr>
<td>3.1.2-1</td>
<td>The HRHM shall accept HTTP Requests from Frontend</td>
</tr>
<tr>
<td>3.1.2-2</td>
<td>The HRHM shall request the DAM for data</td>
</tr>
<tr>
<td>3.1.2-3</td>
<td>The HRHM shall return JSON data to the Frontend</td>
</tr>
<tr>
<td>3.1.2-4</td>
<td>The HRHM shall accept HTTP Requests from the TCUP</td>
</tr>
<tr>
<td>3.1.2-5</td>
<td>The HRHM shall return JSON data to TCUP</td>
</tr>
<tr>
<td>3.1.2-6</td>
<td>The HRHM shall accept JSON data from the DAM</td>
</tr>
<tr>
<td>3.1.2-7</td>
<td>The HRHM shall provide “/nonmatching/all” endpoint for retrieving non matching accounts</td>
</tr>
<tr>
<td>3.1.2-8</td>
<td>The HRHM shall provide “/balances/plan/dates” endpoint for retrieving individual balances by plan with date ranges.</td>
</tr>
<tr>
<td>3.1.2-9</td>
<td>The HRHM shall provide “/balances/contact/dates” endpoint for retrieving individual balances by contact with date ranges</td>
</tr>
<tr>
<td>3.1.2-10</td>
<td>The HRHM shall call “/balances/all” endpoint for retrieving all balances report</td>
</tr>
<tr>
<td>3.1.2-11</td>
<td>The HRHM shall call “/api/v1/balances/search/bbc/{name}” for retrieving balances that match the indicated name</td>
</tr>
<tr>
<td>3.1.2-12</td>
<td>The HRHM shall call “/api/v1/accountTransHistory/{accountnumber}” for retrieving the transaction history of the indicated account number</td>
</tr>
<tr>
<td>Requirement No.</td>
<td>Requirement Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>3.1.2-13</td>
<td>The HRHM shall call “/api/v1/allaccountsbycontact/{globalentityaccountid}” for retrieving every account that belongs to the specific contact</td>
</tr>
<tr>
<td>3.1.2-14</td>
<td>The HRHM shall call “/api/v1/allAccountTransHistory/{accountnumber}” for retrieving all the transaction history of every account belonging to the contact</td>
</tr>
</tbody>
</table>

**Testing CUP Module (TCUP) 2.1.3 Requirements**

<table>
<thead>
<tr>
<th>Requirement No.</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.3-1</td>
<td>The TCUP shall accept HTTP Requests from HRHM</td>
</tr>
<tr>
<td>3.1.3-2</td>
<td>The TCUP shall return validation/verification to HRHM</td>
</tr>
<tr>
<td>3.1.3-3</td>
<td>The TCUP shall test every endpoint in the above HRHM requirements</td>
</tr>
</tbody>
</table>

**Trend Analysis Module (TAM) 2.1.4 Requirements**

<table>
<thead>
<tr>
<th>Requirement No.</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.4-1</td>
<td>The TAM shall display statistical data using visual representation such as charts, histograms, etc.</td>
</tr>
</tbody>
</table>

### Requirements Related to Design Module 2.2 and Sub-modules 2.2.1, 2.2.2, etc.

<table>
<thead>
<tr>
<th>Requirement No.</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Interface (UI) 2.2.1 Requirements</strong></td>
<td></td>
</tr>
<tr>
<td>3.2.1-1</td>
<td>The UI shall render and display “IndividualBalancebyPlan” component to the User.</td>
</tr>
<tr>
<td>3.2.1-2</td>
<td>The UI shall render and display “IndividualBalancebyContact” component to the User.</td>
</tr>
<tr>
<td>3.2.1-3</td>
<td>The UI shall render and display “NonMatchingAccounts” component to the User.</td>
</tr>
<tr>
<td>3.2.1-4</td>
<td>The UI shall render and display “AllAccountBalance” component to the User.</td>
</tr>
<tr>
<td>3.2.1-5</td>
<td>The UI shall call functions inside PPM for parsed JSON data.</td>
</tr>
<tr>
<td>3.2.1-6</td>
<td>The UI will pass parameters to the PPM.</td>
</tr>
<tr>
<td>3.2.1-7</td>
<td>The UI will display the appropriate data received from the PPM using components.</td>
</tr>
<tr>
<td><strong>Parameter Processing Module (PPM) 2.2.2 Requirements</strong></td>
<td></td>
</tr>
<tr>
<td>3.2.2-1</td>
<td>The PPM shall call the endpoint mentioned in 3.2.1-7</td>
</tr>
<tr>
<td>3.2.2-2</td>
<td>The PPM shall call the endpoint mentioned in 3.2.1-8</td>
</tr>
<tr>
<td>3.2.2-3</td>
<td>The PPM shall call the endpoint mentioned in 3.2.1-9</td>
</tr>
<tr>
<td>3.2.2-4</td>
<td>The PPM shall call the endpoint mentioned in 3.2.1-10</td>
</tr>
<tr>
<td>3.2.2-5</td>
<td>The PPM shall parse JSON data received from endpoints.</td>
</tr>
<tr>
<td>3.2.2-6</td>
<td>The PPM shall provide parses data to UI for rendering components.</td>
</tr>
<tr>
<td>3.2.2-7</td>
<td>The PPM shall receive data parameters from UI.</td>
</tr>
</tbody>
</table>

### Requirements Related to Design Module 2.3 and Sub-modules 2.3.1, 2.3.2, etc.

<table>
<thead>
<tr>
<th>Requirement No.</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Drill-Down (DD) 2.3.1 Requirements</strong></td>
<td></td>
</tr>
<tr>
<td>3.3.1-1</td>
<td>The All Balances DD displays the transaction history of a selected account</td>
</tr>
<tr>
<td>3.3.1-2</td>
<td>The Plan DD displays the transaction history of the selected plan account to the user</td>
</tr>
<tr>
<td>3.3.1-3</td>
<td>The Contact DD displays the transaction history of the selected contact account to the User</td>
</tr>
<tr>
<td>3.3.1-4</td>
<td>The Flagged Accounts DD displays the balance of the selected account</td>
</tr>
<tr>
<td><strong>Unique Features (UF) 2.4.1 Requirements</strong></td>
<td></td>
</tr>
<tr>
<td>3.4.1-1</td>
<td>Email - User will email a specific individual for information about a non-matching account</td>
</tr>
<tr>
<td>3.4.2-1</td>
<td>Export Excel - User can download the entire report as an excel sheet</td>
</tr>
</tbody>
</table>
4.2 External Interface Requirements

I. All Account Balances
   A. Name of item - Zone Filters
   B. Description of purpose - Filtering accounts based on account type
   C. Source of input or destination of output - Search bar
   D. Valid range, accuracy and/or tolerance - Not Applicable
   E. Units of measure - Not Applicable
   F. Timing - Not Applicable
   G. Relationships to other inputs/outputs - Not Applicable
   H. Screen formats/organization - The table displayed on the screen shall be reduced in size, as the information gets filtered based on the account type.
   I. Window formats/organization - Tabled structure
   J. Data formats - Strings
   K. Command formats - Calls to appropriate REST APIs with the selected zone as a parameter
   L. End messages - HTTP response verification

4.3 Logical Database Requirements

The database is already created and maintained by La County Department of Regional Planning. The developers will not need to make any changes or create any new tables in the database for development purposes.

4.4 Design Constraints

I. The Sql Server Express instance used by DDI must be configured to listen on TCP port 1433, along with a separate user credentials created for the local DDI server to authenticate with.

II. The user credentials must be configured with read-only permission to avoid any possible data corruptions.

III. DDI shall be developed on a private repository. Software developers shall commit new patches on the private github repository, which they shall be added to as contributors.
5. **Other Nonfunctional Requirements**

5.1 **Performance Requirements**

The performance requirements for DDI will be created after our first on-site test. We need the response time of the LA County database to determine the statistical data for performance.

5.2 **Safety Requirements**

In order to avoid any data corruption, the user credentials that DDI uses to access the database must have a read-only permission.

5.3 **Security Requirements**

In order to protect the data in the LA County database, the software shall be deployed in a server that is closed inside the LA County network.

5.4 **Software Quality Attributes**

**Portability** - DDI shall allow System Administrators to easily configure the database URL, ports and hostIP. This way, the software can be ported from place to place if needed.

**Testability** - DDI shall provide REST APIs that can be tested for correctness using different test libraries such as RestAssured.

**Usability** - The web app provided by DDI shall be intuitive, and responsive. For HTTP calls, there shall be a progress meter that gets updated throughout each HTTP call cycle.
5.5 Business Rules

This product is used only by the employees at LA County Department of Regional Planning. Any changes to the product must be authorized by the Principal Application Developer at LA County.

Development milestones shall be reported to the respective IT Manager associated with this project.
6. **Other Requirements**

Not Applicable at the moment.
Appendix A: Glossary
The Acronyms and definitions have been listed in Section 1.4.
Appendix B: Analysis Models

DFDs for DDI can be found in the Software Design Document.
### Appendix C: To Be Determined List

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 4.1</td>
<td>Finish creating the remaining requirements for frontend and backend module.</td>
<td>Complete</td>
</tr>
<tr>
<td>Section 3.1</td>
<td>UI Images for incomplete functionalities</td>
<td>Complete</td>
</tr>
<tr>
<td>Section 4.2</td>
<td>External Interface Requirements</td>
<td>Complete</td>
</tr>
</tbody>
</table>