



OD Setup

SRS Document

ODS -Organization Directory Setup	3
Introduction	3
Management Summary	3
Key Assumptions	4
High Level Architecture	4
Functional Requirements	4
Use Case Diagrams	5
Use Cases	6
<i>Upload Masters</i>	6
<i>Setup Organization Units</i>	7
<i>Add Employees</i>	8
<i>Employee Transfer</i>	9
<i>Record Employee Exit</i>	11
Logical Object Model	11
Database Design Guidelines	12
Testing Approach	12
Suggested Technical Reading	13

Disclaimer

This Software Requirements Specification document is a guideline. The document details all the high level requirements. The document should be used as a guideline by the students to design the Solution Architecture for the project. The document also describes the broad scope of the project and high level logical object model. But while developing the solution if the developer has a valid point to add more details being within the scope specified then it can be accommodated after consultation.

ODS - Organization Directory Setup

Introduction

The purpose of this document is to define scope and requirements of an Organization Directory Setup Module that can be used in any of the business applications being implemented for customers by Prime Software Systems.

This document should be used by the development team to architect the solution the project.

Management Summary

As a software company, re-usability, quick deployment and implementation are some of the key factors that make software projects profitable. Deploying over 20 new customized business workflow applications was a challenge as the Organization Setup consumed fair amount of time during the rollouts thus adding a chunk of time and issues to the project timeline. The technical team decided to work on an a Generic Organization Directory Setup Module to be used with the solutions being delivered. ODS - organization directory setup is focussed on making the process simple even though the organization structures vary from company to company. ODS should be able to map the Organization requirements effectively. .

The scope of SOD for workflow applications will covers

1. Units of the Organization Directory
2. Define hierarchy within Business Units
3. Employee Masters
4. Employee Mapping to Units
5. Employee Reporting
6. Employee Events like - Transfer or Leaving

The proposed solution will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

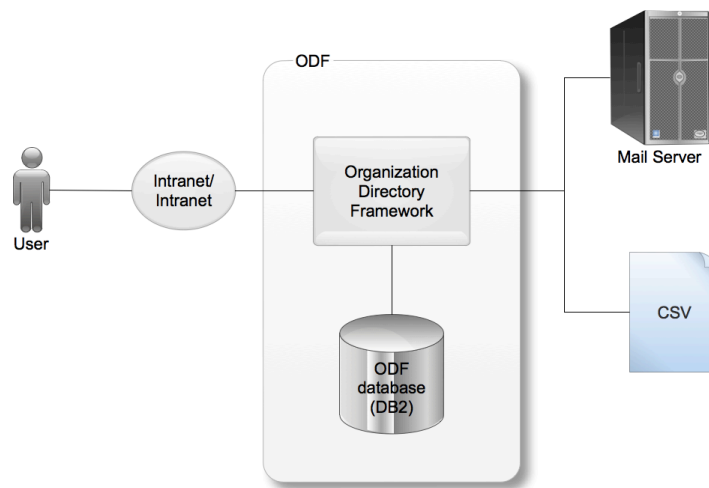
Key Assumptions

1. The developer has familiarized him/her self with workflow applications need for a robust Organization Directory (OD) Structure and the components of an OD.
2. List of Countries and Cities can be upload as master from backend.
3. Employee master data with Employee id and Employee Name is assumed to be entered from backend.

High Level Architecture

ODS high level architecture is illustrated through the context diagram shown below. It will have following categories of users:

1. HR Admin
2. Administrator



ODS Context Diagram

ODS	The Organization directory structure facilitates operations spanning in various regions such as countries or cities and all of them are related in a hierarchy. Employee events such as Transfer or leaving are handled as a special case for the workflow apps to keep functioning without any break.
ODS Database	Database stores information entered in OD
CSV	Employee Master, Type of Location, Location, Designation
Mail Server	All notifications are routed via this mail server

Functional Requirements

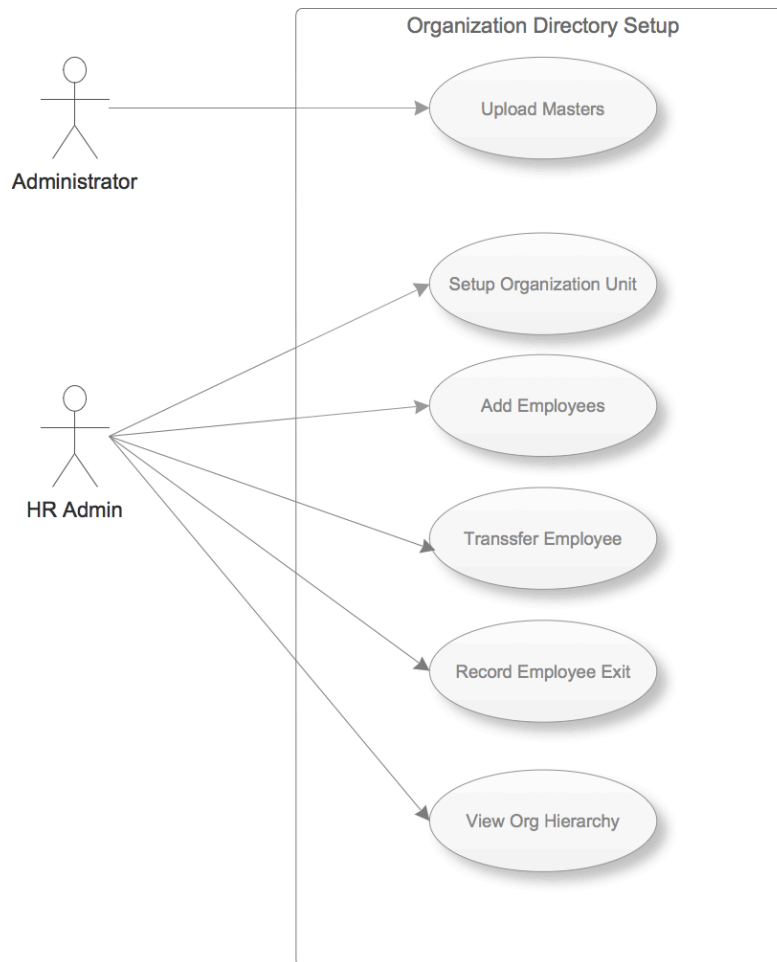
The high level functional requirements for the ODS are outlined in the Use Case diagram described in this section.

ODS will provide a secure user-id/password based secured login mechanism to access its services. The details of this are not outlined here. The development team is expected to create these keeping in mind the general practices followed by the web applications. Once user logs in, s/he can view Home page of ODS with the two options

1. Organization Units
2. Employees

Use Case Diagrams

The following figure illustrates the Use Case diagram for the system. The MiS use cases are not detailed here.



Use Case Diagram

Use Cases

Upload Masters

Use Case Element	Description
Number	UC.01
Application	<p>Masters in the application are uploaded as CSV files.</p> <p>The CSV upload module should be able to upload the following masters as and when the feed is required.</p> <p>Location Type Master contains Type id, Type (Corporate office, Branch office, Ware house, Plant, Service Center, Sales Office etc..)</p> <p>Location Master contains Location id, Add1, Add2, City, Country, Location Type</p> <p>Designation Master contains Designation id, Designation</p> <p>Employee id contains Employee id, Employee name, Designation</p> <p>Level Master contains Level id, Level No</p>
Use Case Name	Upload Masters
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the Master Upload on the landing page.
Basic Flow	<p>System prompts for the file name to be uploaded. Standard file upload dialog is presented to select a file from the local system.</p> <ul style="list-style-type: none"> The selected file data is uploaded in the related masters for new records; if an existing record is encountered, the old details are replaced with the new details.
Alternate Flow	<ul style="list-style-type: none"> In event of incorrect CSV format, system gives an error and NO data is uploaded. Operation is cancelled
Output	System displays the number of records uploaded. It also highlights the number of records updated (i.e. already existing ones being replaced)

Setup Organization Units

Use Case Element	Description
Number	UC.02
Application	<p>Organization comprises of various units that we usually relate to as “Head Office or Corporate Office”, “Business Unit or Branch office”, “Functions or Department”.</p> <p>Each of these units will have a head referred to as either CEO, BU head etc. You will notice that all the units mentioned here are either at the same level or appear one below the other.</p> <p>This forms a hierarchy in the org structure and thus requires levels to be denoted so that the workflow kind of applications that execute business rules such as Department head has a limit of approving 1 Lac worth of spend, but in case the expense is exceeding that value, then either Function head or CEO will have to approve before the authorization is granted in the application. Similar case could be with a leave approval, in absence of the department head, let's say the rule is Branch head will approve in case of branches and Function head will approve in case of Corporate office. Apart from approvals, the escalations on SLAs being missed while processing also get highlighted in similar kind of hierarchy. These kinds of business requirements entail setting up of hierarchical structure for the workflow applications.</p>
Use Case Name	Setup OU
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the Setup Organization Unit on the Admin menu of ODS landing page.

Use Case Element	Description
Basic Flow	<ul style="list-style-type: none"> The system follows a wizard like approach to setup the organization. Step 1 - Select the Location Type from the pick list of location master (Tip: At the top level is either the Corporate Office or Head office) <ul style="list-style-type: none"> Enter Name (Company Name) Select the Head for this Location (Employee picker is displayed for selection) Click on Save will save the record as Org Unit Id, Org Unit Name, Level as '0', Unit head, location id, Type id, parent Org Unit id. (Parent Org Unit id will be same as Org Unit id in this case). <p>Note : If there were subsidiaries for this organization then the same step is repeated with Parent Org Unit id as the Top Org Unit</p> <ul style="list-style-type: none"> Step 2 - Define the Business Units in step 2 (Eg. Manufacturing Plant) <ul style="list-style-type: none"> Select location type from pick list by default the previous location type will be displayed and saved along with the record Enter Name : <Automobile Manufacturing> Select the Head for this Unit (Employee picker is displayed for selection) Click on Save will save the record as Org Unit Id, Org Unit Name, Level as '1', Unit head, Location id, Parent Org Unit id as the previous Org Unit id. The step 2 is repeated to create a hierarchy of operations by adding Functions , then departments under the functions. The level number will stored as previous level number +1
Alternate Flow	<ul style="list-style-type: none"> Cancel to abandon the operations, data is saved till the point user has saved in the hierarchy building.
Output	<ul style="list-style-type: none"> None

Add Employees

Use Case Element	Description
Number	UC.03
Application	Employee belongs to a location and a department. System allows user to add employees to a location
Use Case Name	Add Employees
Primary Actor	HR admin
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the Add Employee on the ODS landing page.

Use Case Element	Description
Basic Flow	<ul style="list-style-type: none"> System displays the list of Business Units for adding employees. [As a full fledged application, the employee profile would be detailed, for workflow apps purpose, a very minimalistic data of an employee is being recorded to the existing list) Select Business Unit [Departments - Marketing] The system displays two sections with a divider, Section 1 List of employees, Section 2 Employees Added to the BU. The user is able to drag and drop names of employees to the BU list. Remove button on mouse hover of employee record in the added list will remove from only that list. Remove feature is not available on the source list Click on Save will save the employee records with the selected Org Unit id. Implying the employees now belong to this business unit and operate from the same location as the business unit is selected. The reporting manager now becomes the head of the business unit - in this case its the department head. <p>Note In case of the employee selected and added to the BU is the Head of a BU, then reporting manager would be the BU head of the Org Unit one level above.</p>
Alternate Flow	<ul style="list-style-type: none"> In event of any error, it is clearly displayed and user is asked to reenter data or perform operation again.
Output	None

Employee Transfer

Use Case Element	Description
Number	UC.04
Application	Employee Transfer impacts reporting and location details of the employee. In workflow applications, its an important event to be handled in the Org Directory
Use Case Name	Employee Transfer
Primary Actor	HR Admin
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the Transfer on the ODS landing page.

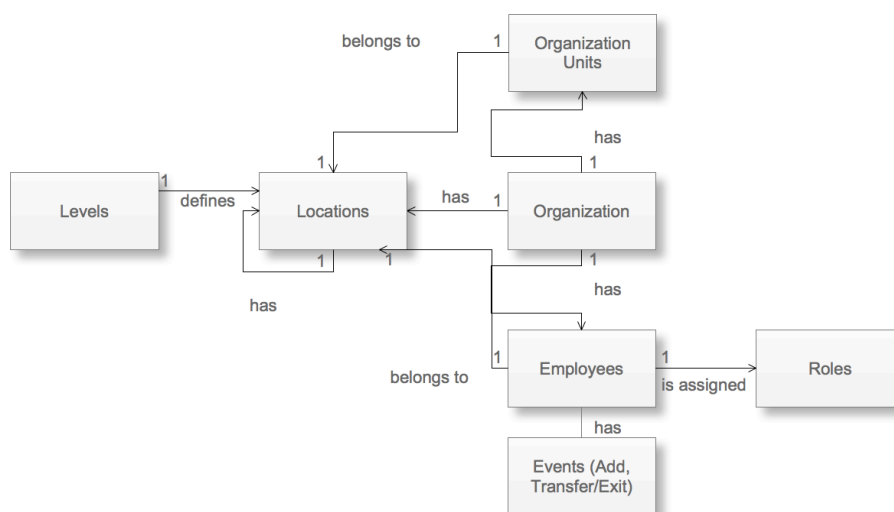
Use Case Element	Description
Basic Flow	<ul style="list-style-type: none"> • The system displays list of Business Units to select from. • User selects one BU from the list (e.g. department in a manufacturing plant) • The list of employees in the department are displayed in the selection window. • One or more employees can be selected if the Transfer is to a single location else one by selection is done. • User selects the Transfer to location from the list. • All BUs belonging to the selected location are displayed. The user selects one BU for transferring the employee(s) to it. • Clicking on Update will update the employees of the Org Unit id (new BU selected), the reporting manager is updated with New Org Unit head. <p>(Implications in a business application of this event are far reaching eg. all the leave applications or expense claims or budget approvals or indents etc will go to the new org unit heads bucket for approval).</p>
Alternate Flow	<ul style="list-style-type: none"> • Cancel abandons the operation, no database.
Output	<ul style="list-style-type: none"> • Email notification to new org unit head is sent informing of a transfer done on the system.

Record Employee Exit

Use Case Element	Description
Number	UC.05
Application	Employee leaving impacts all business applications as there are authorizations attached to employees role. of the employee. Access and authorizations to create or approve any of the workflow documents gets revoked.
Use Case Name	Employee Exit
Primary Actor	HR Admin
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the Exit on the ODS landing page.
Basic Flow	<ul style="list-style-type: none"> The system displays list of Business Units to select from. User selects one BU from the list (e.g. department in a manufacturing plant) The list of employees in the department are displayed in the selection window. Select one employee from the list to deactivate. Clicking on Deactivate will update the flag for employee as “Deactivated”. <p>(Implications in a business application of this event are far reaching eg. the selection list of employees will not show the deactivated employee, nor will the employee be able to login.)</p>
Alternate Flow	<ul style="list-style-type: none"> Cancel abandons the operation, no database.
Output	<ul style="list-style-type: none"> Email notification to org unit head is sent informing of deactivation

Logical Object Model

During the technical design it will be transformed into a physical model covering all system entities. Such a diagram will include their relationship and its cardinality.



1. Organizations may operate from 1 or more locations.

2. Locations follow a hierarchy that is extensible. The top of the hierarchy is Organization's Corporate Office.
3. Each level of hierarchy defines the type of the location.
4. Organization has many employees. Each employee belongs to a location.
5. Each employee r has 1 or more roles assigned to them.
6. Roles determine the access rights of the user.

Database Design Guidelines

This involves the transformation of the use cases, state diagrams, and logical object model into detailed and optimized physical database table designs.

Typically persistent classes will map to table(s) with their attributes as columns of the table. In some cases a high level object may map in to a master-child table. Invoice is one such example where it maps in to "invoice_header" and "invoice_line_item" table.

Associations between two persistent objects are realized as foreign keys to the associated objects. A foreign key is a column in one table that contains the primary key value of the associated object.

Similarly, a standard technique in relational modeling is to use an intersection entity to represent many-to-many associations. Following is a broad checklist for physical database database design:

1. Database must be properly normalized except those instances where de-normalization help improves performance. This option must be used with special care.
2. All persistent classes that use the database for persistency must map to database structures.
3. Many-to-many relationships must have an intersecting table.
4. Primary keys should be defined for each table, unless there is a performance reason not to define a primary key.
5. Indexes should be defined to optimize access.
6. Data and referential integrity constraints should be defined.

Testing Approach

Quality of the software can be achieved with basic hygiene and consistency followed during design and development of User Interface(UI), Navigation, Validations as per the business process requirement.

To ensure the project delivers acceptable quality to the customer, its important to create a checklist of the conventions to be followed across. Common checks as below are for your reference during design and development:

Common Checks	Validation Type
Page Title is valid for the feature being provided on the page	UI
Order of the Data Entry Fields is logical as per the functionality being provided by the feature	UI
Order of the Display only Fields makes viewing and understanding easy for the user	UI
Spellings and Correctness of Label for the Data Entry and Display fields	UI
The labels are not wrapping onto another row thereby adding a blank row on the page	UI
The fields with drop down are displayed in single row instead of drop down coming on the next row	UI

Common Checks	Validation Type
Data Entry field basic validations are working i.e Text field /Numbers / Dates allow data for their type only	Functional
The dates are following a standard format dd/mm/yy on all forms	UI
The color scheme of all forms i.e headers labels , alerts, entry fields are uniform throughout the application	UI
The action buttons for a New Data Entry Form are uniform for all forms that is allowing data entry	UI
The action buttons are performing the desired action e.g. "submit" is creating a new record if there are no errors and recording all the input fields, whereas 'cancel' is not creating a new record in the database	Functional
The links provided on the forms are opening correctly.	Functional
The data feed mechanism for Read and Write files is generating a log with count of entries.	Navigation

Suggested Technical Reading

The project is aimed at making the student understand concepts of Design and Development using IBM Rational tools, WebSphere Application Server and DB2 Database. The following reading reference is easy to understand and should be read to get a clear understanding of capabilities of the tools and how you would leverage them to execute a project.

Technical Reference	URL to access
RAD - Tackling challenges of software development with Rational Application Developer for WebSphere Software	http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html
IBM Education Assistant - Rational Application Developer 7.5	http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rad_v7/rad/rad75.html
RSA-Overview of Rational Software Architect for WebSphere Software Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_arnold/index.html
Using the new features of UML Modeler in IBM Rational Software Architect Version 7.5	http://www.ibm.com/developerworks/rational/library/08/0926_diu/index.html
Rational Technical Library	http://www.ibm.com/developerworks/rational/library/