Optimize

SRS Document

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.

Optimize - hiring process

Introduction

The purpose of this document is to define scope and requirements of a hiring process in a business house where attrition is a grave problem and occurs in large numbers.

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

Management Summary

The company has an increasing attrition problem and it need to take steps to reduce the attrition. After analysis of the problem, it has been determined that the current hiring system does not appear to be capable of handling frequent hiring, resume management and having transparency in complete process of hiring.

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

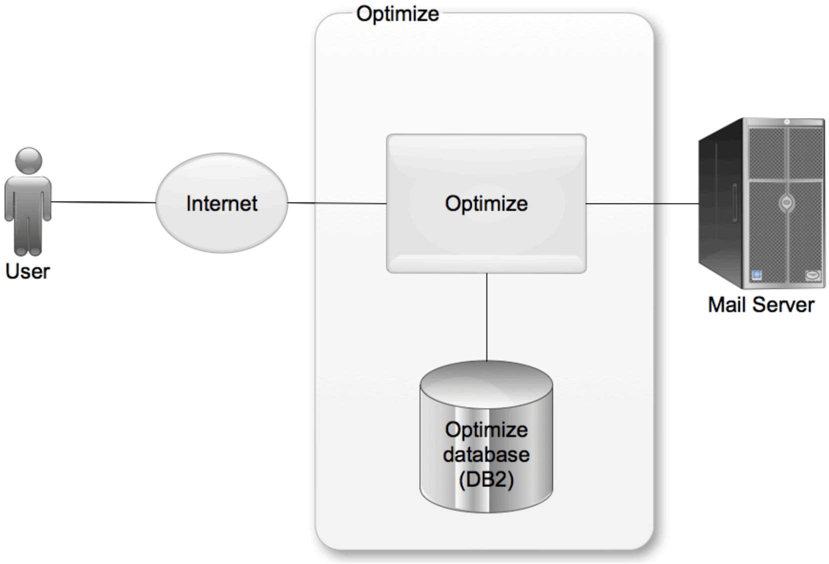
Key Assumptions

1. Resumes are not posted by outside users. Only organizational users can access the system and update the resume bank.
2. Resume's key information will be captured in database for search purposes, the resume copy will stored as an attachment in the system. Only Doc files are accepted as attachments by the system.

High Level Architecture

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

1. HR Admin
2. Hiring Staff
3. Employee



Optimize Context Diagram

Optimize	The system proposes to bring transparency in hiring by ensuring all role players have an access to the resume bank, any kind of action on a profile from resume bank gets recorded, thus history of profile processing is maintained, employee referral feature is also provided to encourage resume building from within the organization.
Optimize Database	Resumes, Skills, Employees, Hiring Process events are recorded in the system
Mail Server	All notifications are routed through the Mail Server

Functional Requirements

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

Optimize 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. Login will be a prerequisite to use Optimize. Internal users will be provided user id/password pair separately.

Once user logs in, menu options shall primarily come from the use case of the role player, e.g.

1.Candidate profile management: Create a new profile, Update an existing profile, Delete a profile

2. Support for uploading a profile in an accepted format.

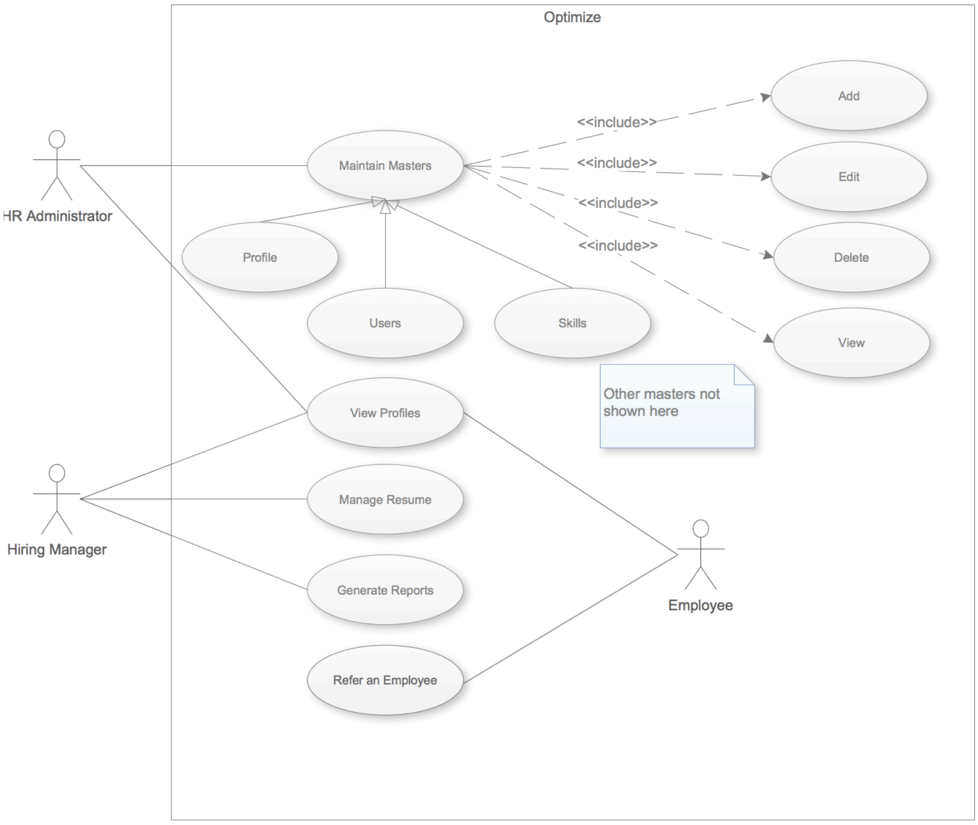
3. Manage the resume lifecycle.

4. Enforce eligibility criteria including blacklisting of a profile.

5. Generate reports based on required filters

Use Case Diagrams

The following figure illustrates the Use Case diagram for the system.



Use Case Diagram

Use Cases

Maintain Masters

Use Case Element	Description
Number	UC.01
Application	Master maintenance in terms of basic operations viz. **add**, **edit**, **delete** and **view**. All master maintenance i.e.Profiles, Skills, List of Universities, Colleges, are child use cases of this Use Case.
Use Case Name	Maintain Masters
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	Administrator clicks on the **Maintain Masters** menu item on the admin interface page
Basic Flow	• System presents a list of masters that can be maintained. Administrator selects the desired master. • System displays a list *view* and links for *add*, *edit* and *delete*. 1. In case add, a new master record data entry form is presented. The master record is saved on clicking the save button provided form clears all the data validations (if any). The list view is updated accordingly. 2. In case of edit, from the list view user is prompted to select the desired record to edit, Selected record is opened for editing. The edited master record is updated on clicking the update button, provided form clears all the data validations (if any). 3. In case of delete, from the list view user selects the check box(s) against each record. Selected records are deleted up on clicking the delete button. However, user is presented a confirmation dialog before deleting the records.
Alternate Flow	• In event of any error, it is clearly displayed and user is asked to renter data or perform operation again.
Output	System displays the details of the successful operation.

View Profile

Use Case Element	Description
Number	UC.02
Application	View a profile from the resume bank
Use Case Name	View Profile
Primary Actor	Hiring Manager
Secondary Actor	HR Admin
Pre-condition	None
Trigger	User clicks on the **View Profile** link on the landing page.

Use Case Element	Description
Basic Flow	• The system displays filter options for viewing the profiles from the Resume Bank. • User may enter the criteria for any one of the following: Name, Domain, Last Organization, Technical Skills, Education • The system displays profiles with these columns and a hyperlink on the name to click open the resume details. • On closing the resume, the view reappears, user may refresh the criteria to look for a separate set of profiles.
Alternate Flow	None
Output	None

Manage Resume

Use Case Element	Description
Number	UC.03
Application	Resume Lifecycle management
Use Case Name	Manage Resume
Primary Actor	Hiring Manager
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the **Manage Resume** link on the landing page
Basic Flow	• The system displays filter criteria to select a resume as in View Profile Use case. • The user can select a profile from the list displayed from the selection criteria entered. The user can select the following states in which the resume can remain in the system. • Shortlisted /Contacted/First round interview completed (Record the findings) /Nth round completed/Selected/On Hold/Rejected In the above process, during any step the Hiring manager/HR can return the resume to the pool depending on their findings. • The system saves the profile record with a reference of hiring manager who has changed its state to one of the above during manage resume. • Also a hiring manager cannot keep the resume in a locked state indefinitely. System will alert the Hiring manager to release the resume for consideration for another position in the company.
Alternate Flow	None
Output	Alert in case resume is locked for more than 15 days by the same hiring manager.

Employee Referral

Use Case Element	Description
Number	UC.04
Application	Existing employees add new profiles to the resume bank or modify the existing ones.
Use Case Name	Employee Referral
Primary Actor	Employee
Secondary Actor	None
Pre-condition	None
Trigger	The user clicks on the **Referral** link on the landing page.
Basic Flow	The system displays the list of referrals entered by the logged in employee. User can either update the existing reference or can create a new candidate profile with information such as: Name, Date of Birth, Years of Experience, Domain, Technical Skills, Soft Skills, Highest Education, Last Organization Worked. Mobile Number, Email id, Attach resume. The system should validate for duplicate profiles in the system. Also an email has to be sent to the candidate whose profile is being referred to after the successful referral action. On submit, a new profile be created in the system or the existing one should be up-dated. An appropriate notification is sent to the Employee if the resume is being referred already in the system. After each successful referral actions an email has to be sent to Employee as well as referee.
Alternate Flow	Pressing Cancel abandons operation, no database gets affected
Output	None

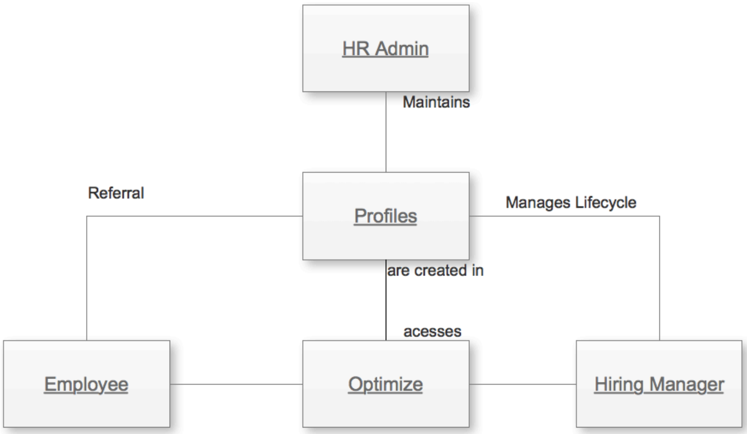
Generate Reports

Use Case Element	Description
Number	UC.05
Application	Reports
Use Case Name	Generate Reports
Primary Actor	Hiring Manager
Secondary Actor	HR Admin
Pre-condition	None
Trigger	The user clicks on the **Generate Reports** link on the landing page.

Use Case Element	Description
Basic Flow	- Multiple filters are displayed by the system for report generation. - The valid filters are Years of experience, University, domain, technology, contacted in last 3 months, contacted in last 6 months, cleared the first round etc. Also the system should give the figures such as how many candidates were contacted in a given period, how many are hired, how many came on board etc.
Alternate Flow	Alert if no records match the criteria
Output	As per criteria selected

Logical Object Model

A high level logical object model of the system is shown below. During technical design it will be transformed into a physical model covering all system entities. Such a diagram will include their relationship and its cardinality.



Logical Object Model

1. Optimize is a hiring system, that allows management of resume bank.
2. The key users of the system are HR admin, Hiring Manager (multiple), employees.
3. The HR admin is responsible for upkeep of resume bank. The resumes are entered with key information in a form and the soft copy attached by the HR admin.
4. The resume can be accessed by Hiring manager (Department user who has the responsibility of hiring), HR admin and the employee who has given referral.
5. The Hiring manager maintains the resume states as part of life cycle management
6. The employee marks resumes for Referral in the resume bank.

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
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/mmm/yyyy on all forms	UI
The color scheme of all forms i.e headers labels , alerts, entry fields are uniform throughout the application	UI

Common Checks	Validation Type
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, Web Sphere 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/