SWOT - Sales Tax Workload Distribution System

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.

Sales Tax Workload Distribution System - SWOT

Introduction

The purpose of this document is to define scope and requirements of Workload Distribution System to allocate equitable work amongst officers at Sales Tax Department. Such flow of work is in form of new registrations, refund processing and CST declaration. The proposed system will automate the work allocation to each officer at the Department of Sales Tax on a real time basis, while ensuring equal workload on each officer.

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

Management Summary

Sales Tax Department has a challenge to manage the large influx of work that flows daily in form of new registration requests, refund processing and processing of CST declarations. These are referred as **work types**. This work needs to be allocated to its officer on an equitable basis so that each officer has virtually the same workload with him/her. Since the number of requests runs in to several hundreds and each request has a different processing complexity, this allocation task itself becomes prohibitively complex and results in to delays and customer dissatisfaction. The problem is further compounded as the work type needs a specific skill to complete it; plus each work type has a priority for completion.

To address the above challenge, an effective workload distribution system is envisaged. The proposed system - SWOT will deliver:

1. Provide automated workflow distribution to officers on a real time basis
2. Ensure equitable workload distribution amongst the officer on the basis of work type (i.e. complexity or estimated time needed to complete a work item and it’s priority)

SWOT will be a web-based system and will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

Key Assumptions

1. Complexity of each work type will be determined as (a) on the basis of number of transactions in refund processing & CST declaration and (b) number of warehouses for a new registration request.

2. To further simplify the development without losing the intent of the problem, it is assumed that a work item will only have an unique id, description, work type, a number (of transactions or warehouses), its status (new, assigned, in progress, and completed). It also maintains date/time stamp for every status change for MIS.

3. The skills do not have different levels of expertise. It is a binary attribute i.e. either officer has a specific skill or does not have a specific skill.

4. There are only 3 priorities in for a work type viz., low, medium, and high. A higher priority work is always allocated first.

5. Work items are uploaded using a CSV format in to the system everyday in the morning.

6. The work items will be queued in the inbox of each officer and the officer will be able to pull a work item from the inbox in a FIFO manner. A work item can be pulled only after the current work item at hand has been completed.

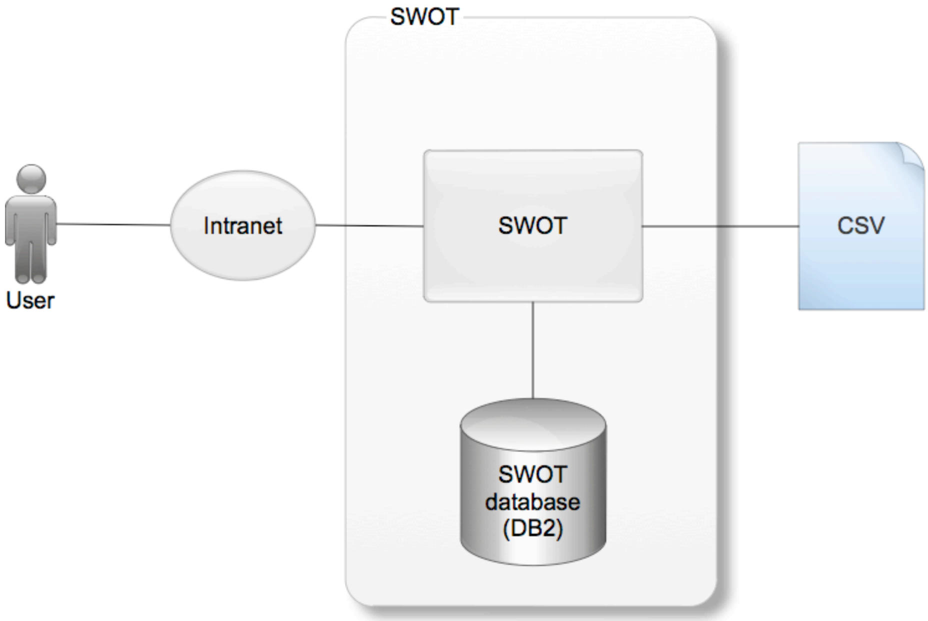
High Level Architecture

SWOT's high level logical architecture is illustrated through the diagram shown in figure 1 below. It will have following categories of users, also referred as officers:

1. Assistant/Deputy Commissioner (to whom work is assigned)

2. Joint/Additional Commissioner (who are consulted as SME and view their dashboards)

3. Commissioner (who has overall access to dashboards)



SWOT Context Diagram

SWOT	The proposed web-based application, SWOT will be accessed by the officers of Sales Tax Department over it's intranet. Users will login to the system to access its services and this will also mark their availability for the day. The admin user will be able to upload incoming flow of work via CSV files.
CSV File(s)	The work load is uploaded in to the system from a CSV file. Such a CSV file is uploaded in to the system everyday morning. The CSV file has 4 columns viz. work item id (always unique), description, work type (new registration, refund or CST declaration) and a number (of transactions or warehouses).
SWOT Database	This will hold all the SWOT data including the work type, skills, officers and their details, and incoming work items.

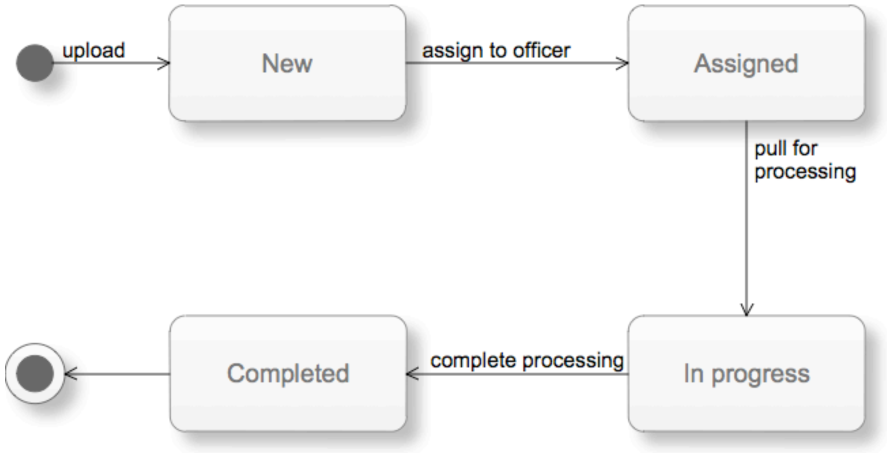
Functional Requirements

The high level functional requirements for the SWOT system are outlined in the State and Use Case diagram described in this section.

SWOT 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.

State Diagram

The following figure outlines the state diagram for a work item in the system.



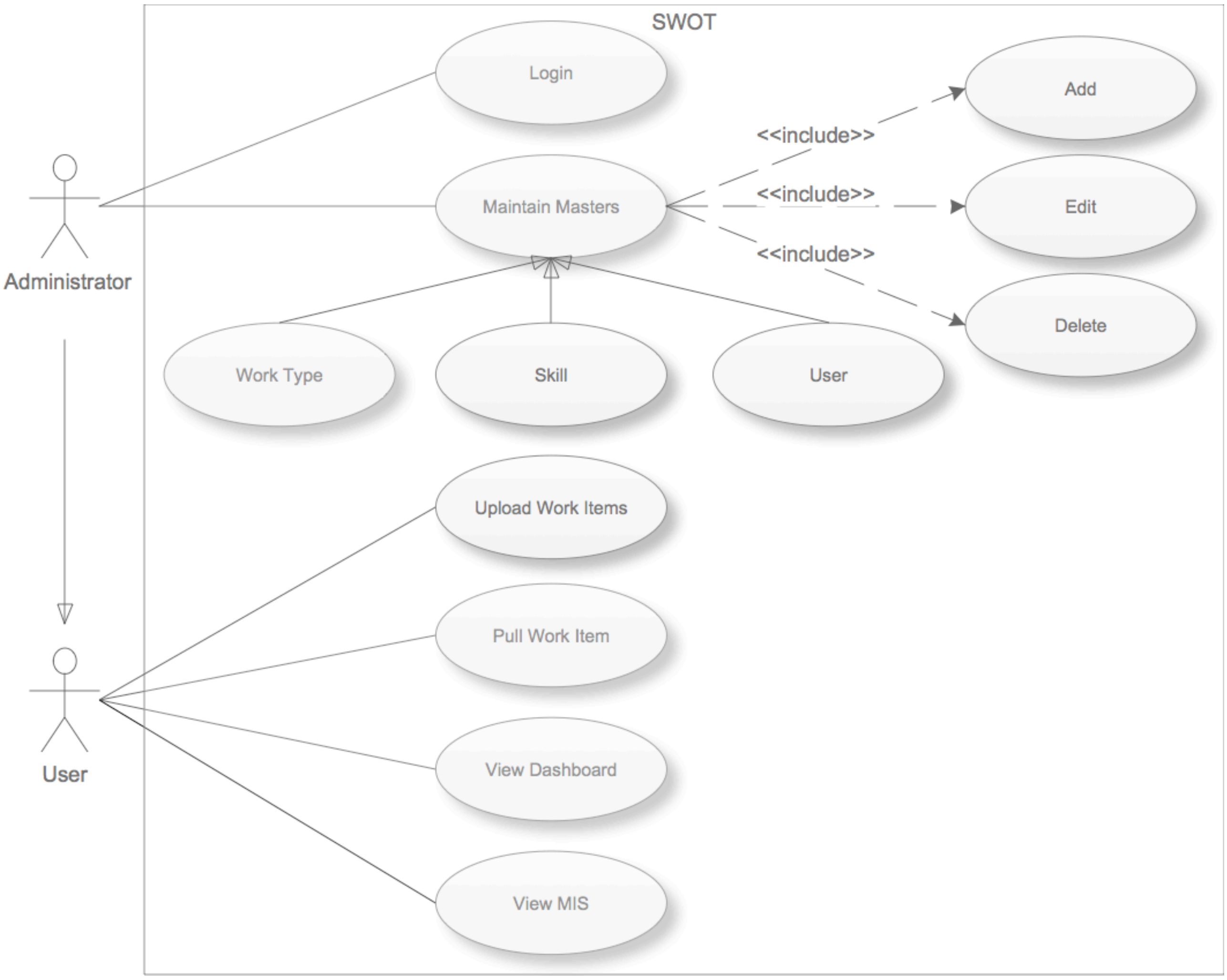
State Diagram

1. All uploaded work items are marked as **new**.
2. Once a work item is assigned to a officer, it's state becomes **assigned**.
3. Item pulled for processing by an officer from his/her inbox, transitions to **in progress** state.

4. Once processing is completed for a work item, it is marked as **completed**.

Use Case Diagrams

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



Use Case Diagram

Use Cases

Login

Use Case Element	Description
Number	UC.01
Application	Login to the system; it is a prerequisite to access any other system services.
Use Case Name	Login
Primary Actor	User
Secondary Actor	None
Pre-condition	None
Trigger	Administrator clicks on the **Login** link on the department's intranet home page
Basic Flow	• System prompts for user-id/password of the user. • User enters the required information and clicks login; correct credential logs the user in to the system and marks him/her as available for the day..
Alternate Flow	• In even of incorrect incorrect credentials, an error is displayed and user is asked to re-enter details.
Output	None

Maintain Masters

Use Case Element	Description
Number	UC.02
Application	Master maintenance in terms of basic operations viz. **add**, **edit**, and **delete**. All master maintenance i.e. work type, priority, skill, and user 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

Use Case Element	Description
Basic Flow	• System presents a list of masters that can be maintained. Administrator selects the desired master. • System prompts for the operation, i.e. add, edit or 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). 2. In case of edit, a list of all master records is presented and 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, a list of all master records along with a check box against each record is presented (to enable multiple selection). 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.

Upload Work Items

Use Case Element	Description
Number	UC.03
Application	Work item data is uploaded in the system everyday morning. It is carried out at 10am everyday (by this time every present user is logged in to the system)
Use Case Name	Upload Work Items
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None
Trigger	Administrator clicks on the **Upload Work Items** menu item on the admin interface page
Basic Flow	• System prompts for the file name to be uploaded. Standard file upload dialog is presented to select a file from the local system. • The selected file data is uploaded in the part master; if the work item id is already present in the SWOT database, it is ignored. The uploaded item(s) are marked as **new**. • The Auto Distribution routine is initiated; each work item, after its assignment to an officer, is marked as **assigned**.

Use Case Element	Description
Alternate Flow	- In event of incorrect CSV format, system gives an error and NO data is uploaded - Operation is cancelled without initiating the Auto Distribution routine.
Output	System displays the number of records uploaded. It also highlights the number of records ignored (i.e. the ones for whom the data was already present).

Pull Work Item

Use Case Element	Description
Number	UC.04
Application	Pulling a work item from the user's inbox for processing.
Use Case Name	Pull Work Item
Primary Actor	User
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **Pull** menu item on the home page
Basic Flow	- The new work item from the user's inbox (if any) is opened for processing. It has 2 options, viz. save and complete. 1. On save, the work item is saved and is returned to user's inbox. Its state remains unchanged (i.e. assigned). Whenever user pulls next, the same item is again opened for processing. 2. On complete, the work item is mark completed and is removed from the user's inbox. Note, in this system only a dummy form is opened displaying work item attributes as defined earlier. There are no operations or data-entry to be performed.
Alternate Flow	None
Output	None

View Dashboard

Use Case Element	Description
Number	UC.05
Application	Viewing online status of work items; the scope depends on the role of the user as outlined in the section on High Level Architecture.
Use Case Name	View Dashboard
Primary Actor	User (administrator is a kind of user)
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **View dashboard** menu item on the home page

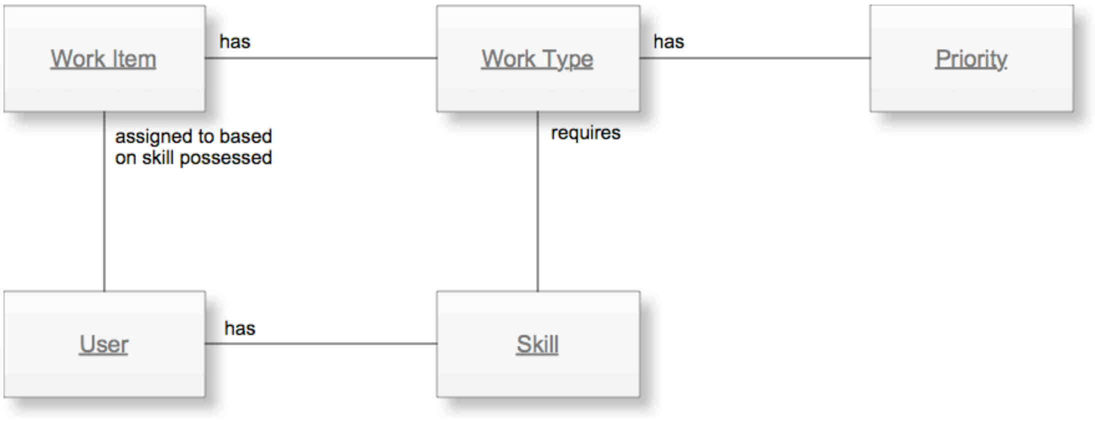
Use Case Element	Description
Basic Flow	- System determines the scope based on the user role to identify the work items to display. - The overall status of identified items is displayed in a summary form i.e. %assigned, %in progress, and %completed (that are completed today only). - All users have an option to view status by work type also. - In case of the commissioner, s/he has the option to view this status by the supervisor also (apart from the total status).
Alternate Flow	None
Output	Dashboard

View MIS

Use Case Element	Description
Number	UC.06
Application	Viewing MIS for all the completed work items
Use Case Name	View MIS
Primary Actor	User
Secondary Actor	None
Pre-condition	None
Trigger	Administrator clicks on the **View MIS** menu item on the home page
Basic Flow	System display the turn around time to complete of each work item for the selected work type and the date range.
Alternate Flow	None
Output	MIS Report

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. Each work type has a work type.
2. Each work type has a priority of completion.
3. Each work type requires a skill to process a work type items.
4. A user has one or more skills.
5. Work item is assigned to a user provided s/he has the skill to process it.
6. Work item has an unique id, description, work type, a number (of transactions or warehouses), its status (new, assigned, in progress, and completed). It also maintains date/time stamp for every status change for MIS.

Workload Distribution

The workload distribution has to be carried out on the basis of the following criteria:

1. An item can be assigned to an officer provided s/he has the skill to perform that work type.
2. Item complexity is determined on the basis details outlined in section on Key Assumptions.
3. Higher priority work type items must be assigned ahead of lower priority work types.

Develop a robust algorithm based on above guidelines and state clearly any additional assumptions made.

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
The action buttons for a New Data Entry Form are uniform for all forms that is allowing data entry	UI

Common Checks	Validation Type
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/