Value Track - High Value Transaction Tracking

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.

High Value Transaction Tracking System - Value Track

Introduction

The purpose of this document is to define scope and requirements of automating the tracking of high value transactions like Bank Guarantees, Letter of Credit, Foreign Exchange, Treasury etc for the bank's business customers. The Value track system shall be an online application maintaining branch wise track of these transactions. The proposed system will provide visibility of high value transactions and their status at branch and head office level.

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

Management Summary

A bank with widespread operations is struggling with control over high value transactions happening for its customers in the form of Bank Guarantees, Letter of Credit, Foreign Exchange, Treasury etc. There is no centralized system to take care of these transactions. A web based system is proposed that will record the high value transactions taking place in the branches.. The proposed system - Value Track will:

1. High visibility into banks liabilities taken up during High value transactions
2. Easy to inform customers for the required documentation or fee as and when due.
3. Branch wise view available as well as the data can be rolled up based on the transaction type at the head office level.

It was proposed to develop a web based online system with one key objectives in mind - "to get control on the high value transactions happening at various locations of the bank". Value Track will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

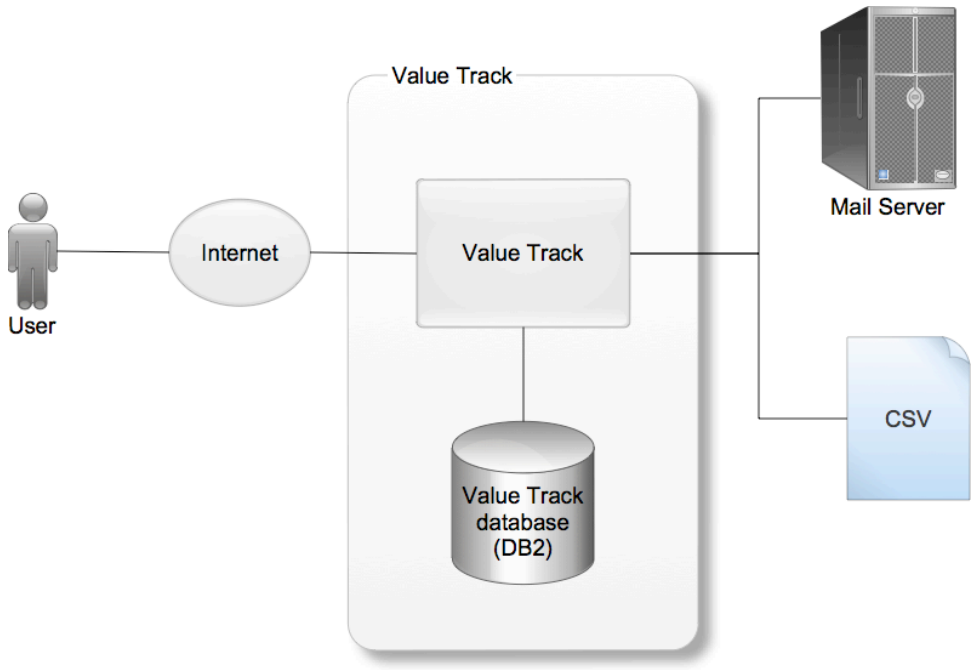
Key Assumptions

1. There is no customer interaction with the system.
2. While it is nice to have e-mail alerts & reminders to notify the customers of any requirements or meetings in regard to the High value transaction, these features will not be incorporated in this system. However, adventurous developers are welcome to add such features.
3. Complexity of a High Value tracking system has been taken away for the project duration purposes.
4. There is no linking with core banking system. The customer profiles are uploaded as CSV file generated by the core banking system after business hours. Any new accounts created for the day are pushed into CSV file. Value track reads the file from a shared location and uploads data at the trigger of admin user.
5. The branch logs into the application by providing branch code. The users are mapped to the branch code.

High Level Architecture

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

1. Administrator
2. Branch Staff
3. Branch Head
4. Head Office User



Value Track Context Diagram

Value Track	The proposed web-based application, Value Track will be accessed by the bank staff at the branches and administrator, HO officer at the head office. At the branch office users shall be create the high value transaction record in the Value Track application. Head office users view the dashboard as a consolidated picture and can also drill down to branch level data.
Value Track Database	This will hold all the Value Track data including the masters like Customers, Type of transactions, documents and daily transactions of high value
CSV	All the masters are uploaded into the system through the CSV format. Branches master CSV contains Branch Code, Branch Address1, Address2, City High Value transactions(HVT) are the type of services available to their account holders for fulfilling their work related requirement such as setting up and Letter of Credit for a consignment to be imported etc. The Structure of HVT master shall be HVT Code, HVT Description, Bank Charges in % HVT also has list of documents to be submitted by the customer, the list is captured via CSV with columns such as HVT Code, Document Title Customer Profile is an output in CSV format from the core banking system. The profile is captured as Account Number, Customer Name, Address1, Address2, City, Mobile Number, E-mail id
Mail Server	Mail server will be used for sending communication via email to the customer

Functional Requirements

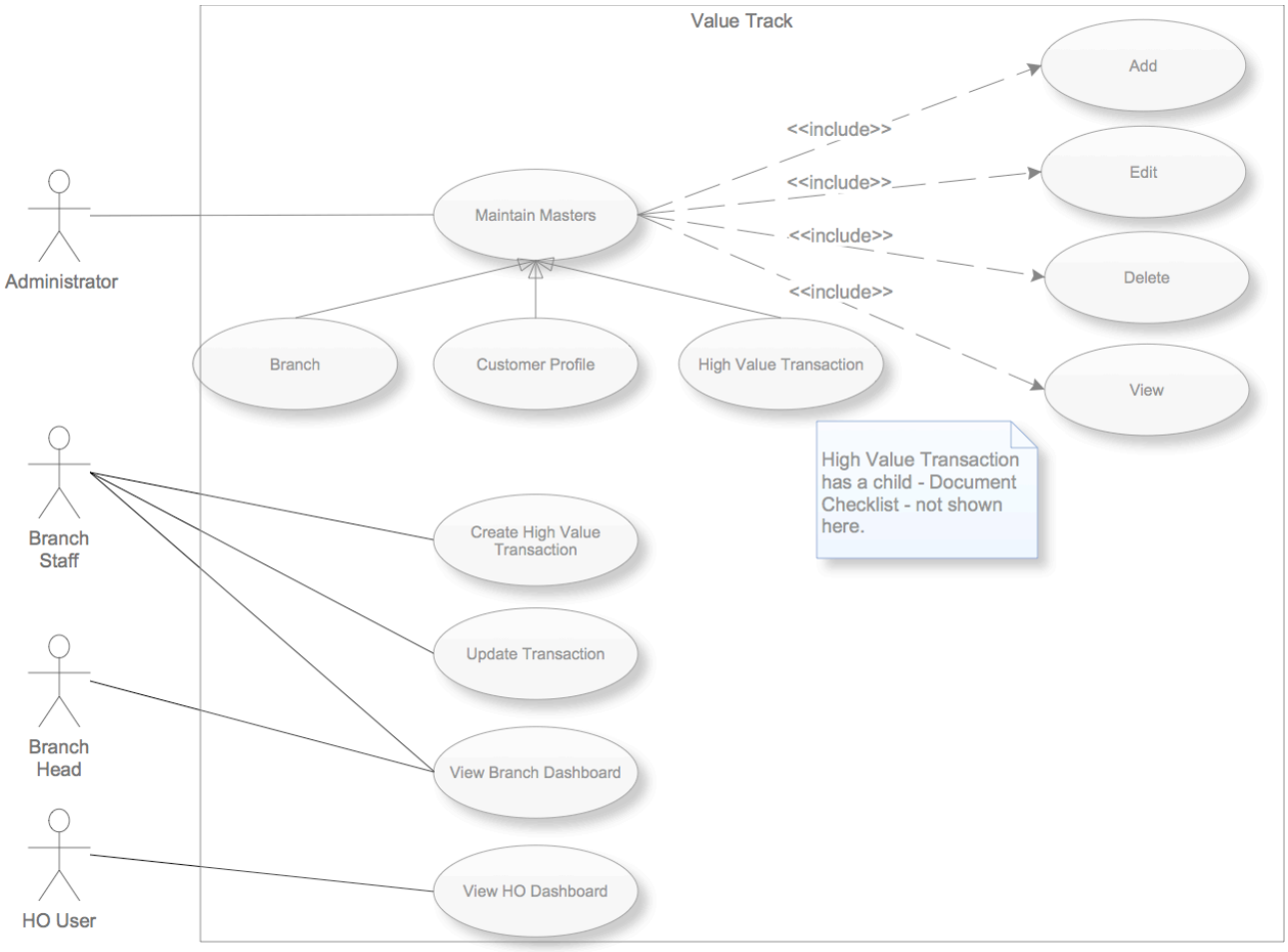
The high level functional requirements for the Value Track system are outlined in the Use Case diagram described in this section.

Value Track will provide a secure user-id/password based secured login mechanism to access its data. 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 Value Track. Internal users will be provided user id/password pair separately

Once the bank staff logs in, s/he can view todays High Value Transactions and their status from “HVT Today”. Note, this page is only visible after bank staff logs in to the system.

Use Case Diagrams

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



Use Case Diagram

Use Cases

Master Maintenance

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. High Value transaction master, HVT Document checklist, Customer profile are child use cases of this Use Case.
Use Case Name	Master maintenance
Primary Actor	Administrator
Secondary Actor	None
Pre-condition	None

Use Case Element	Description
Trigger	Administrator clicks on the **Master Maintenance** link 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 *upload, add*, *edit* and *delete*. 1. In case of upload, a dialogue window prompts the user to select the file to be uploaded. The selected file is uploaded. The upload procedure validates for existing entries, if found, they are replaced by the new details, new records get added to the existing master. 2. 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. 3. 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). 4. 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	Upload of exiting entries, replaces the existing data with new entry Delete of a master record checks for its usage in the transaction table. If present, does not delete and displays and alert ‘Master record has been used in transaction, cannot delete”
Output	None

Create High Value Transaction

Use Case Element	Description
Number	UC.02
Application	Customer approaches the bank to use high value services. The high value tracking system updates existing customer profiles on daily basis. Thus customer profile is present in the high value system, the record is pulled using the account number and the high value transaction type is selected by the bank staff to proceed with full transaction.
Use Case Name	Create High Value Transaction
Primary Actor	Branch Staff
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **Create New HVT** link on “HVT Today page”

Use Case Element	Description

Basic Flow	• A transaction form is displayed.
	• Enter customer account number, the system pulls up data from customer profile and displays the name of the account holder, address and phone number.
	• Selects the High value transaction as required by the customer. A drop down is displayed from the HVT Master.
	• List of documents required for the purpose are displayed with check box to update the status. HVT document checklist is used to display the list of documents.
	• Enter the following fields to complete the transaction record.
	  ◦ Value of the transaction
	  ◦ Effective date
	  ◦ Beneficiary Name
	  ◦ Beneficiary Address
	  ◦ Contact Number
	  ◦ Bank charges for the HVT (computed as Value of HVT * Bank Charges % from the master)
	  ◦ Total Payment Due
	• Enter the target date for receiving the documents to process the HVT request.
	• On Save of this request, HVT transaction id is generated by the system

Alternate Flow	Click Cancel will abandon the HVT creation, database does not get updated.
Output	Transaction id that is communicated via email to the customer along with due date for documents and total payment due for the transaction.

Update High Value Transaction

Use Case Element	Description
Number	UC.03
Application	To update status of documents or modify existing transaction information
Use Case Name	Update High Value Transaction
Primary Actor	Bank Staff
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **Update HVT** link on “HVT today page”

Use Case Element	Description

Basic Flow	• System displays search page that allows user to enter customer account number. The High Value transactions under processing for the searched account number are displayed. The user selects the transaction to update.
	• A form in edit mode is displayed with following fields as Read only.
	  • Account Number
	  • Account Holder's Name
	  • Address
	  • Contact Number
	  • HTV type
	• The user can edit the following information:
	  • List of documents required for the purpose are displayed with check box to update the status. User can upload a scanned copy of the document.
	  • Value of the transaction
	  • Effective date
	  • Beneficiary Name
	  • Beneficiary Address
	  • Contact Number
	  • Bank charges for the HVT (computed as Value of HVT * Bank Charges % from the master)
	  • Total Payment Due (Computed as Value of the transaction + Bank charges)
	  • Target date for documentation to process the HVT request.
	• Click on Update, the system marks the status of the transaction as "Ready for Processing" and if checklist of documents has content and is checked by the user, the status is marked as "Awaiting Documents"

Alternate Flow	Click on Cancel will not save any changes
Output	E-Mail stating HTV transaction has moved into processing.

View Branch Dashboard

Use Case Element	Description
Number	UC.04
Application	To view the status of High value transactions in the bank's branch
Use Case Name	View Branch Dashboard
Primary Actor	Branch Staff
Secondary Actor	Branch head
Pre-condition	None

Use Case Element	Description
Trigger	User clicks on the **View Dashboard** link on “HVT today page”

Basic Flow	• The system displays various perspectives of High Value transactions in sections using following criteria.
	• For a period : Start Date & End Date [Display default month beginning to be taken as start date and end date should be today’s date]
	  • High Value Transaction Type, Count and Total Value
	  • Pending for documents, Count and Total Value
	  • Under processing, Count and Total Value
	  • Total of Bank Charges for transactions whose effective date is falling between the reporting period
	  • Date Wise break up of total value

Alternate Flow	If the period selected does not have any data, show alert of No data found for the selected period.
Output	None

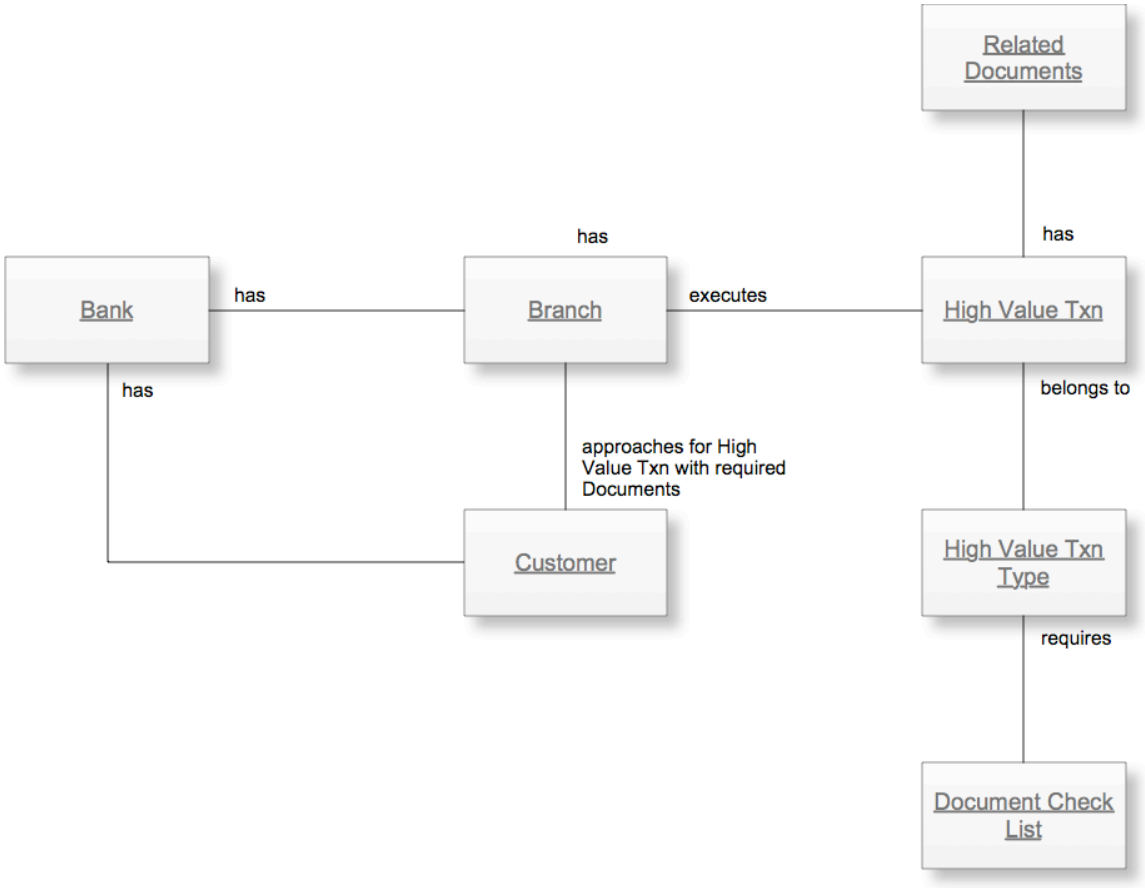
Head Office Dashboard

Use Case Element	Description
Number	UC.05
Application	The head office users reporting needs on High Value transactions is met with this feature. They can view a consolidated picture and also look up branch wise data.
Use Case Name	Head Office Dashboard
Primary Actor	Head Office Users
Secondary Actor	None
Pre-condition	None
Trigger	User clicks on the **View Head office Dashboard** link on home page

Use Case Element	Description
Basic Flow	The user selects the view type [Consolidated or for a Branch] Select branch from the list of branches displayed from the branch master. Selects the period as Start Date & End Date [Display default month beginning to be taken as start date and end date should be today's date] Consolidated View type displays • High Value Transaction Type, Count and Total Value • Branch code, Branch Name, Total Value, total bank charges Consolidated View type displays • Branch Name, Branch Head • High Value Transaction Type, Count and Total Value • Pending for documents, Count and Total Value • Under processing, Count and Total Value • Total of Bank Charges for transactions whose effective date is falling between the reporting period • Date Wise break up of total value
Alternate Flow	None
Output	None

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. A bank has multiple branches across cities in the country.
2. Branches service customers having account in their bank.
3. Branches provide high value transaction services to their existing customers.
4. Customers approach the branch office with documents for bank to execute a high value transaction
5. The High value transaction is of various types
6. There are set of documents associated with a type of High value transaction.

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/