Software Requirements Specification

For

Transaction Management System

Version 2.1 approved

Prepared by Ritika Chaudhary

ABC BANK

<date created>

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Product Scope 1

1.3 Overview 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Class and Characteristics 2

2.4 Operating Environment 3

3. Functional Specifications 3

3.1 Create New Transaction 3

3.2 Search Existing Transaction 4

3.3 Check Session User Authorizations 4

4. External Interface Requirements 4

4.1 User Interfaces 4

4.2 Software Interfaces 4

5. Other Nonfunctional Requirements 5

5.1 Performance Requirements 5

5.2 Security Requirements 6

5.3 Software Quality Attributes 6

Appendix A: Analysis Models 7

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Ritika Chaudhary | 10/09/2016 | Requirements changed | 2.0 |
| Ritika Chaudhary | 12/09/2016 | Schema Changed | 2.1 |

# Introduction

## Purpose

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's goal and its user interface, hardware and software requirements. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

This document gives detailed functional and nonfunctional requirements for the transaction management system for ABC bank. This product will support online transaction creation and search for the bank.

Currently, every time a user needs to search for a transaction it is done manually by looking at the physical records of the bank. This is a time consuming activity. Also whenever a user needs to create a transaction it is entered manually into the physical records of the bank. This is not feasible for large scale eateries.

## Product Scope

The purpose of this system is to create convenient and easy to use online system for transaction management in a bank. The system will be based on a relational database with its transaction create and search functionality.

This Product will automate creation and search of a banking transaction.it will expose APIs for search and creation of transactions. In addition it will include the setting up of a User Interface for the same.

Transaction management system provides search of the details of transactions based upon different criteria. Further it provides for the creation of transactions of various types.

After the search is done, a list of transactions is displayed to the user. If a transaction is created a success message or message is displayed to the user.

Transaction creation is not allowed on a holiday with respect to country/region of transaction.

## Overview

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product.  General description of the project is discussed in section 2 of this document.  Section 3 gives the functional requirements, data requirements and constraints and assumptions made while designing the Transaction Management API.  It also gives the user viewpoint of product.  Section 3 also gives the specific requirements of the product.  Section 4 discusses the external interface requirements and gives detailed description of functional requirements. Section 5 is for supporting information.

# Overall Description

## Product Perspective

The Transaction Management database System stores the following information as shown below.

* Transaction Details

It includes transaction date, account number, amount and transaction type id.

* Loan Type Transaction Details

It includes transaction id, AccountFrom and Account To.

* Deposit Type Transaction Details

It includes deposit details for the transaction. It has transaction id, account from and account to.

* Forex Type Transaction Details

It includes from currency, to currency, transaction id, account from and account to.

* Demat Type Transaction Details

It includes units, unit cost, buy or sell option, account from and account to with transaction id.

## Product Features

The major features of transaction management system as shown in below **entity–relationship model** (**ER model**)

ADD ER

## User Class and Characteristics

Users of the system should be able to navigate to our User Interface after Authentication. All the users are then allowed to perform a search on transactions. For create transaction functionality, the transaction details (transaction type, account number, amount) are validated whether the user is allowed to perform the transaction or not.

Users should be able to perform the following function:

* Transaction Details Search
  + Transaction Date
  + Type of Transaction
  + Account Number
  + Date range
* Transaction Creation
* Loan Disbursal
* Fixed Deposit
* Saving Accounts transactions(Deposit/Withdrawal)

## Operating Environment

Operating environment for the TRANSACTION MANAGEMENT SYSTEM is as listed below

* centralized database
* client/server system
* operating  system : windows xp
* database: oracle
* platform: JAVA

# Functional Specifications

This section provides the functional overview of the product. The project will require HTML and CSS as front end and at the back end ORACLE database will be running. Various functional modules that can be implemented by the product will be:

1. Create new transaction

2. Search existing transaction

3. Authentication

## Create New Transaction

Users of the system should be able to choose the type of transaction to perform. They should be able to navigate to a specific HTML page. User should then be able to enter details of transaction in a form and submit it.

Validation on transaction details submitted is performed. Accordingly a success or error message is displayed. Amounts of the accounts involved in the transaction should be updated.

In loan disbursal type of transaction, the loan approval will be checked. If valid, the required amount will be debited from the bank’s current liquidity account and will be credited into the savings account of the applicant.

In case of Fixed Deposit amount being transferred to a savings account on maturity, verification is done to make sure that account has actually matured. Then, that deposit will be closed and the amount will be transferred to the specified saving account.

In savings account transaction type, dependent on whether it is deposit or withdrawal the balance in the account will be suitably changed.

After the transaction, a success message or an error message will be displayed.

## Search Existing Transaction

The users should be able to perform a search based on criteria provided. The SQL query will be generated and run on the transactions database and a JSON object with the results will be sent back to the calling function. User should be able to see the results on screen.

## Check Session User Authorizations

The privileges of the session user must be checked before any functionality is exposed. This will involve calling the BANK STRUCTURE API with the session user’s category.

# External Interface Requirements

## User Interfaces

* The interface must be highly intuitive or interactive because there will be no assistance for the user who is operating the System.
* The interface should sync well with the other connected components within the bank.
* Front End Client:

The system is a web based application clients are requiring using modern web browser such as Mozilla Firefox 1.5, PHP.

* Web Server:

The web application will be hosted on one of the Apache server.

* Back End:

We use backend as ORACLE.

## Software Interfaces

* The development environment should be connected to SVN repositories for continuous code integration and backups.
* In addition to the native Oracle database of our transactions module, our system will interact with the Customer Account Oracle database (to receive and update account information), the User Access control Oracle database (to verify if user is allowed to access functionality) and the Reference Oracle database (to obtain trade related information).
* The API should provide information to the reporting database in the form of a JSON.

# Other Nonfunctional Requirements

## Performance Requirements

The steps involved in creating the Transaction Management System are explained below.

1. **Flow Chart**

Flowcharts are used in designing and documenting simple processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help understand a process, and perhaps also find flaws, bottlenecks, and other less-obvious features within it. There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions. The two most common types of boxes in a flowchart are:

* processing step, usually called *activity*, and denoted as a rectangular box
* decision, usually denoted as a diamond

1. **E-R Diagram**

E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

* ENTITIES:  Specifies distinct real-world items in an application.
* PROPERTIES/ATTRIBUTES: Specifies properties of an entity and relationships.
* RELATIONSHIPS:  Connects entities and represent meaningful dependencies between them.

1. **Normalization**

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a Database is not properly designed it can gives rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

## Security Requirements

The user must be authenticated to ensure only authorized users have access to the system. Any customer related information must be protected carefully.

Transaction is a critical process so connectivity to the server should be maintained at all times to provide security to users.

## Software Quality Attributes

The code should be robust, complete, adaptive, modular and reusable. It should also scale easily and integrate well with the other modules of the banking system being deployed.

* AVAILABILITY:  The database should be available when required.
* CORRECTNESS:  The transactions made should be correctly executed. The results displayed for search should be correct.
* MAINTAINABILITY: The transaction tables should be maintained properly along with the related tables.
* USABILITY: The code should be reusable.
* ADAPTABLITY: The code should be able to be changed in future changes.
* INTEGRATION: The code integrates well with other modules of the banking system.

Appendix A: Analysis Models

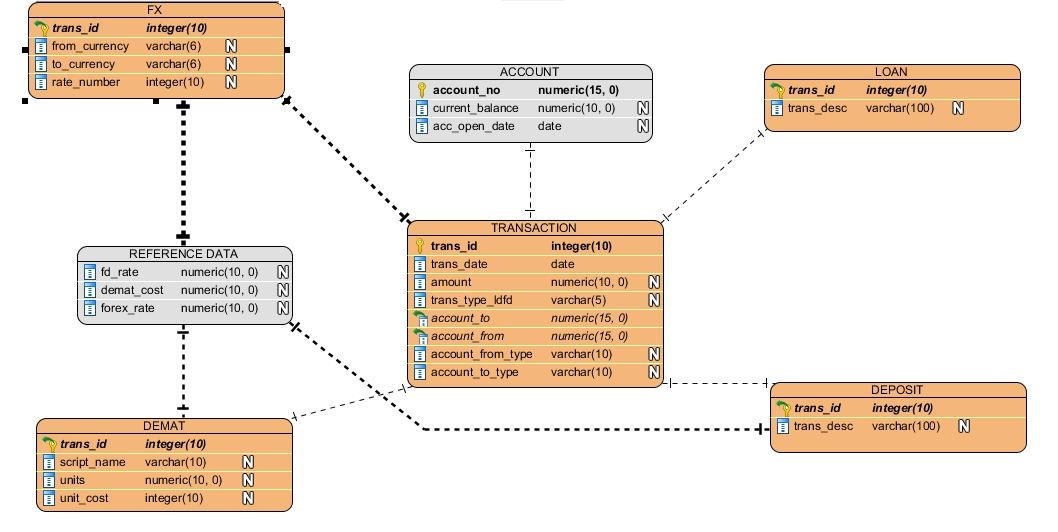


Figure 1: ER Diagram

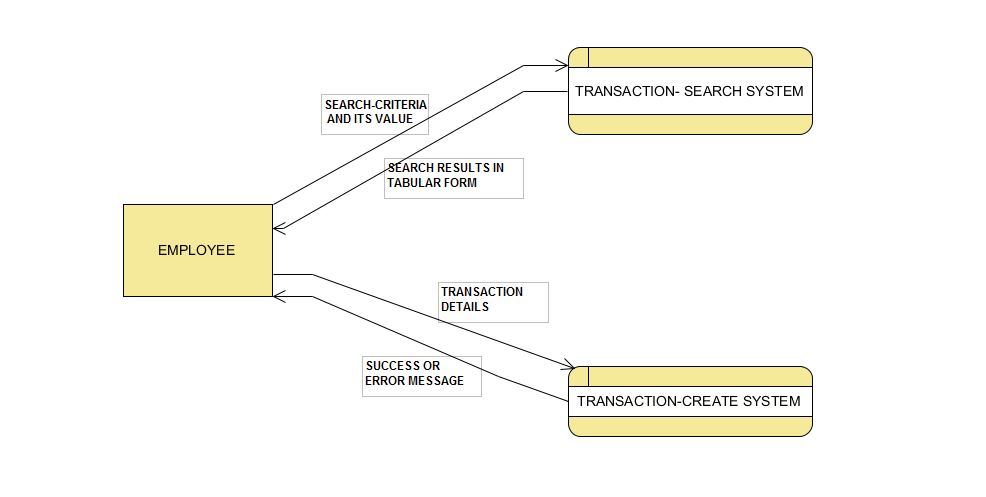


Figure 2: Data Flow Diagram Level 1

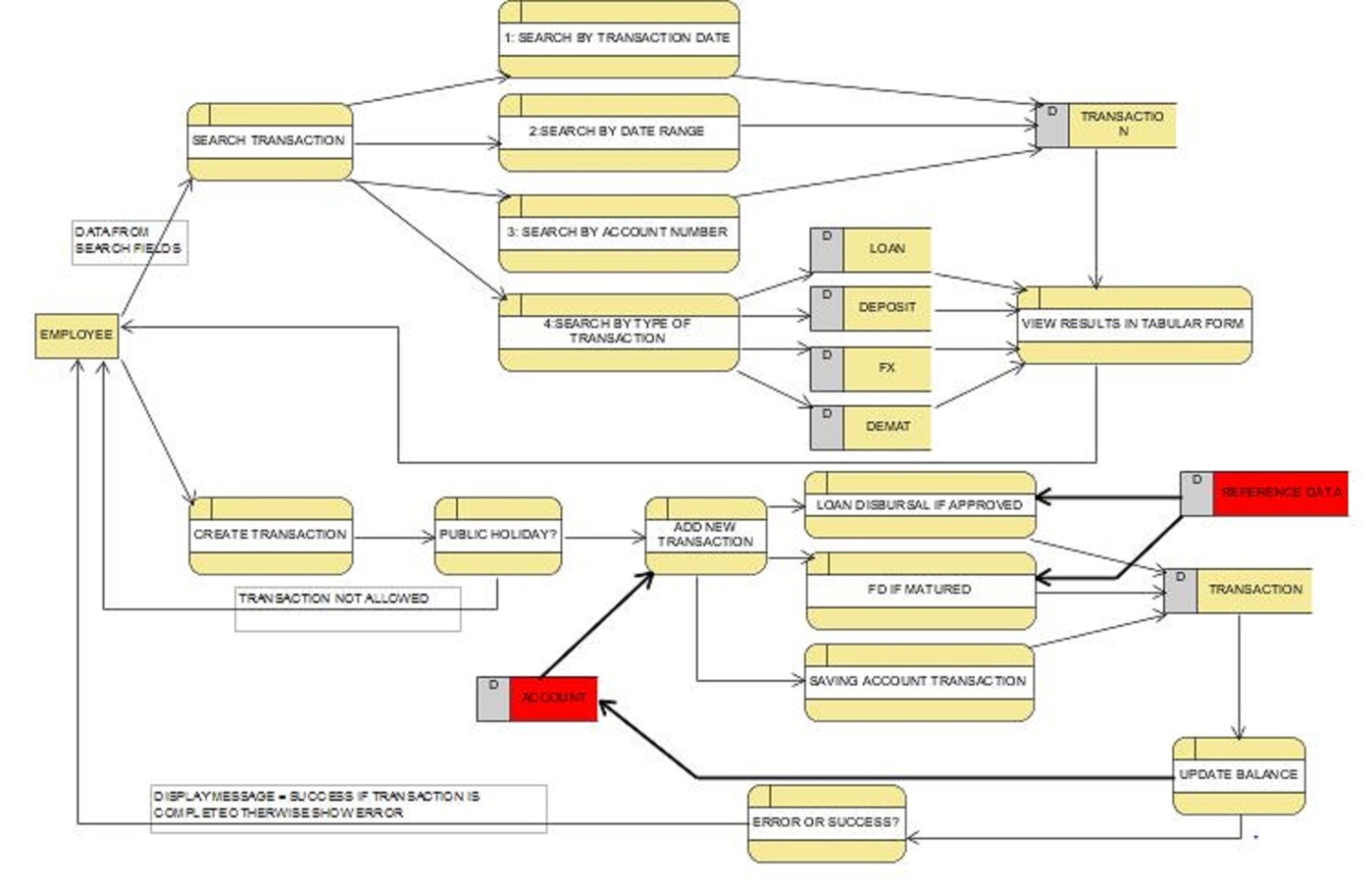


Figure 3: Data Flow Diagram Level 2

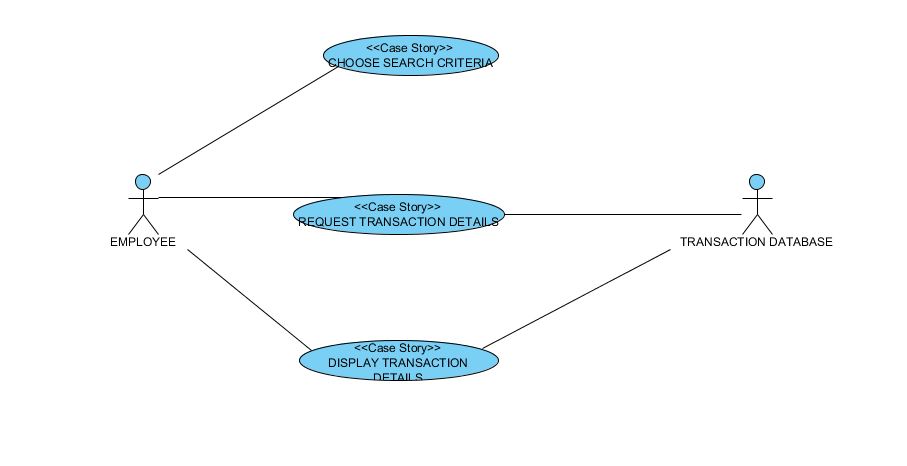


Figure 4: Search Use case diagram

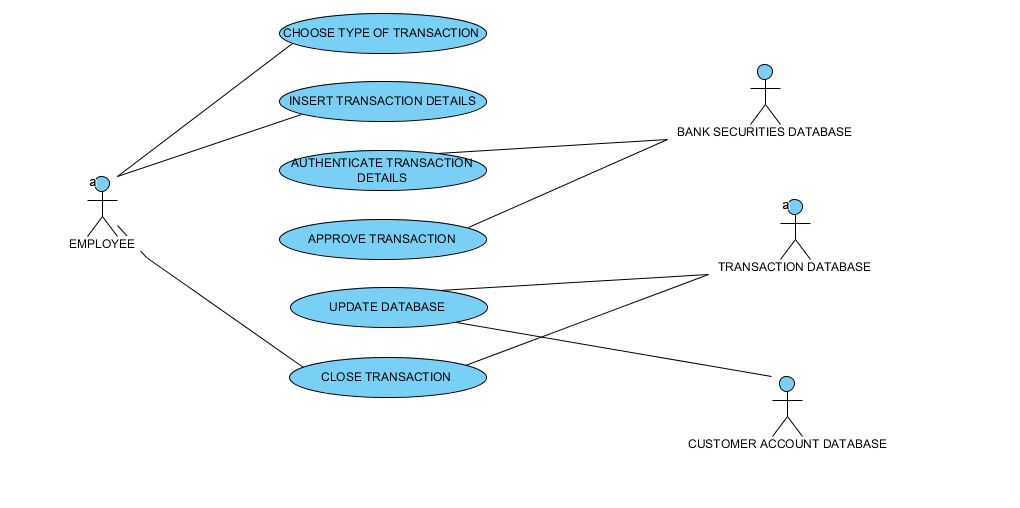


Figure 5:Transaction Create use case diagram