



# **Locker Management**

SRS Document

<b>Locker Management - Online for Account Holders</b>	<b>3</b>
<b>Introduction</b>	<b>3</b>
<b>Management Summary</b>	<b>3</b>
<b>Key Assumptions</b>	<b>3</b>
<b>High Level Architecture</b>	<b>4</b>
<b>Functional Requirements</b>	<b>5</b>
State Diagram	<b>5</b>
Use Case Diagrams	<b>5</b>
Use Cases	<b>6</b>
<i>Apply for New</i>	<b>6</b>
<i>Report Lost key</i>	<b>7</b>
<i>Request for Change of Co-Signee</i>	<b>8</b>
<i>Request for Surrender</i>	<b>8</b>
<i>Update Status</i>	<b>9</b>
<i>View Request Status</i>	<b>10</b>
<b>Logical Object Model</b>	<b>10</b>
<b>Database Design Guidelines</b>	<b>11</b>
<b>Testing Approach</b>	<b>12</b>
<b>Suggested Reading</b>	<b>12</b>

## 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.

# Locker Management - Online for Account Holders

## Introduction

The purpose of this document is to define scope and requirements of Locker Management System that will enable the bankers to keep electronic records of all the customer requests and their processing update online.

The document details all the high level requirements of the proposed Locker Management solution. This document should be used by the development team to architect the solution the project.

The scope of the project also involves only CSV based integration with the core banking system. The Locker Management system will provide a web based front-end to the customers and branch authorized staff.

## Management Summary

Bank's key business requirement is to implement delivery channel applications like Locker Management. The primary business objectives are:

1. Customers have online access to post requests and view status.
2. Ad-hoc visits to bank for locker services are reduced and customers visits are planned.
3. Bank has electronic records of the transactions, can service customers in a transparent manner.
4. The process and records are audit-able.

Locker Management will be a web-based system and will be accessed via Bank's website. It will be designed & developed to run on IBM WebSphere Application Server and IBM DB2 Universal Database in a 2-tier architecture.

## Key Assumptions

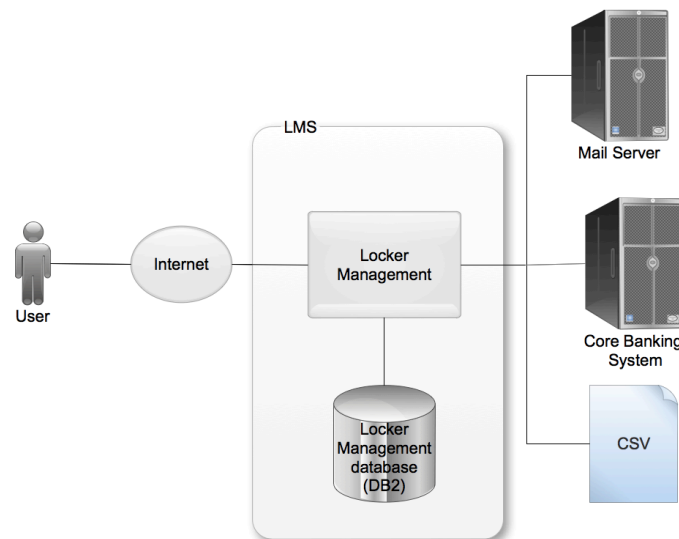
1. The System is assumed for a particular branch as of now, thus there is no need to maintain a branch related masters for the scope of this project.
2. The Administrator is responsible for
  - Uploading CSV of customer updates from the Core Banking System from the backend on daily basis.
  - Uploading Locker Type Master having data such as Locker Type, Annual Charges, Number of Lockers, Assigned, Waiting list count.

- Uploading Locker Inventory Master having data such as Locker Type, Locker Number, Aisle [A.B.C,D...Z], Direction(Left or Right), Row Number, Column Number, Status of 'Assigned', Unassigned'.
3. There are no online payments mechanisms included in the scope of the project
  4. Developer must familiarize with the Locker services provided by the banks and the challenges customer face day in day out to avail of the services.

## High Level Architecture

Locker Management System's high level architecture is illustrated through the context diagram shown in figure below. It will have following categories of users:

1. Branch Staff
2. Customer
3. Administrator



*Locker Management System Context Diagram*

### Core Banking System

This is the backend system used by bank staff while processing customer requests in the office premises. The system maintains complete profile of account holder and up-to date transaction record since the time user holds the account with the bank. This system generates a customer profile CSV for all new customers being added to the Branch.

### Locker Management System

The proposed application over internet that will allow users raise and track requests pertaining to locker services provided by the bank. The system allows the banking users to process the customer requests online to an extent feasible in the ambit of Locker related transactions.

### DBMS

Centralized Repository Locker Holders, specifications, requests etc.

### Mailing System

A system that is used to send notifications or alerts to the external and internal users of the system.

## CSV

CSV is generated by LMS for all customers who have been either issued a locker or have surrendered their lockers. The CSV is accessed by the Core banking system for updating the customer account for the locker related transactions.

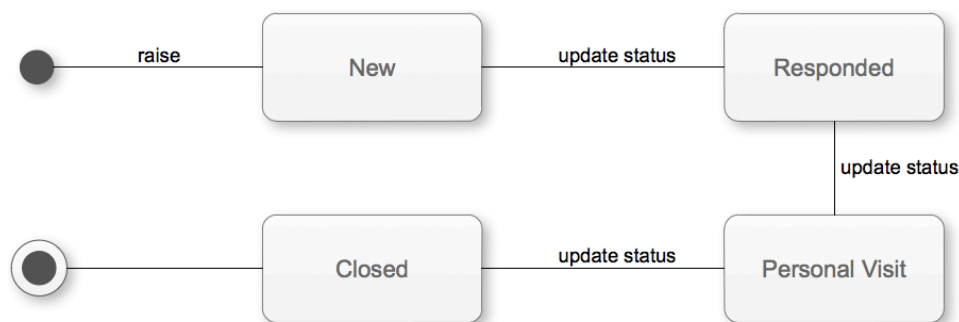
## Functional Requirements

The high level functional requirement for the Locker Management system is represented in the Use Case diagram shown below. The remaining sections in the document describe the major use cases. Locker Management 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 this system. Internal users will be provided user id/password pair separately.

Once user logs in, s/he can view the landing page as per the role assigned to the user. Eg. The Bank Staff shall view options like New Locker Requests, Surrender Requests, Change of Co-Signee, Lost Keys, Visit to Operate. Whereas, the customer shall view My Requests with a list of requests displayed with current status, links to post new request would be provided on the page.

## State Diagram

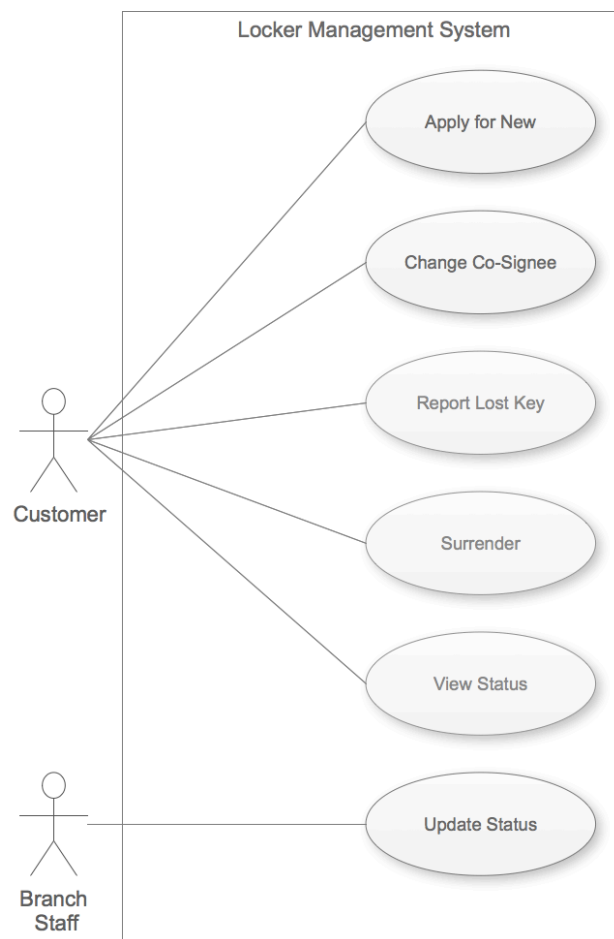
The following figure illustrates the Workflow state transition diagram followed by the requests in the Locker management system.



1. All the requests created in the system are submitted in the New state.
2. The bank staff processes the request by providing the next step details, the request is marked as 'Responded'
3. Next update of status is on personal visit of the customer.
4. Once all pre-requisites are completed to process the request (Offline activities), the request is marked as Closed.

## Use Case Diagrams

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



Use Case Diagram

## Use Cases

### Apply for New

Use Case Element	Description
Number	UC.01
Application	Bank is extending locker related interactions to its existing customers by applying online for the service.
Use Case Name	Apply for Locker
Primary Actor	Customer
Secondary Actor	None
Pre-condition	The user should have a operational bank account in the bank and has internet access to the Banking service for account holders.
Trigger	Account holder clicks on Apply for Locker link on Banking section of website.

Use Case Element	Description
Basic Flow	<ul style="list-style-type: none"> <li>The system displays the type of lockers [Single Shelf, Double Shelf]</li> <li>User selects a type of locker and submits the request</li> <li>The system displays the Waiting list count if greater than zero in locker type master.</li> <li>On submit, the system records the request for new locker with data such as 'Date', Account #, Customer Name, Type of Locker</li> <li>The system also updates the locker type master with Waiting list count +1 for locker type selected by the customer.</li> </ul> <p>Note :The request is not processed online- it will be queued up for the Bank Staff authorized for handling Locker Services in the system.</p> <p>Request will follow a workflow as mentioned in the state diagram.</p>
Alternate Flow	<b>None</b>
Output	None

**Report Lost key**

Use Case Element	Description
Number	UC.02
Application	The customer logs in to report lost key of the locker number.
Use Case Name	Report Lost Key
Primary Actor	Customer
Secondary Actor	None
Pre-condition	Successful login into Locker Management application using account number and password.
Trigger	Customer clicks on menu item Report Lost Key on the Landing Page.
Basic Flow	<p>The system displays the locker number and type details assigned to the customer along with the last visit date:</p> <ul style="list-style-type: none"> <li>A text box to type in the reason for lost is displayed</li> <li>Confirmation with Co-Signee [Yes/No]</li> <li>Do you need a re-issue of key [Yes/No]</li> <li>The user clicks on submit to log this entry or cancel to abandon the entry.</li> <li>The system saves the lost key report with a reference number. The status on Locker record of the customer is marked as 'Blocked for operation'.</li> </ul> <p>Note :The request is not processed online- it will be queued up for the Bank Staff authorized for handling Locker Services in the system.</p> <p>Request will follow a workflow as mentioned in the state diagram.</p>
Alternate Flow	None



Use Case Element	Description
Output	Email alert to bank staff for being vigilant on the requests pertaining to the locker number of this customer.

**Request for Change of Co-Signee**

Use Case Element	Description
Number	UC.03
Application	The co-signee for operating a locker needs a change
Use Case Name	Change of Co-Signee
Primary Actor	Customer
Secondary Actor	None
Pre-condition	Successful login into Locker Management application.
Trigger	Customer clicks on menu item Change of Co-Signee on Landing page.
Basic Flow	<p>The system displays the locker number and type and last operated date.</p> <p>An entry form to capture the following details is displayed:</p> <ul style="list-style-type: none"> <li>Co-Signee 1 (Name from locker record) : [Change / Retain]</li> <li>Co-Signee 2 (Name from locker record): [Change / Retain]</li> </ul> <p>Click on Change on any of these entries will prompt a the Co-Signee to be removed permanently or Replace with another Co-Signee.</p> <p>In case of Replace with another Co-Signee, the system prompts the user to enter the new name and upload the picture.</p> <p>On submit, the change request is logged in to the system with a reference number.</p> <p>Note :The request is not processed online- it will be queued up for the Bank Staff authorized for handling Locker Services in the system.</p> <p>Request will follow a workflow as mentioned in the state diagram.</p>
Alternate Flow	None
Output	None

**Request for Surrender**

Use Case Element	Description
Number	UC.04
Application	Request to surrender the locker.
Use Case Name	Surrender Locker
Primary Actor	Customer
Secondary Actor	None
Pre-condition	Successful login into Locker Management application.
Trigger	Customer clicks on menu item Surrender Locker on the landing page

Use Case Element	Description
Basic Flow	System prompts the confirmation of Surrender and the Last Date of Ownership.  The request is saved on submit with a reference number for surrender.
Alternate Flow	None
Output	None

### Update Status

Use Case Element	Description
Number	UC.04
Application	Processing Status updates for the online direct customer requests for locker services provided by bank
Use Case Name	Update Status
Primary Actor	Bank Staff
Secondary Actor	None
Pre-condition	Successful login into Locker Management application
Trigger	User clicks on menu item Update Requests Status on landing page.
Basic Flow	<p>The System displays various requests list in new state chronologically in the ascending order.</p> <p>The user selects one request at a time for processing. Category of the request defines the processing steps to be navigated in the system.</p> <p><b>New Application Request</b></p> <p>The system displays the request entry filled by the user as mentioned in the Apply for New Locker.</p> <p>During status update, the system displays the waiting number for the Locker type requested for.</p> <p>A button to view the locker type's surrender dates can be used by the user to view what is the probable date for the user to check back for the Locker issue.</p> <p>The user enters the revert by date using date picker as date when locker can be made available as per the surrender dates available.</p> <p>The notification goes to the customer by email and is also displayed online whenever they check status via the system.</p> <p>The request moves to 'Responded' state as per the workflow.</p>

Use Case Element	Description
	<b>Surrender / Co-Signee Change, Lost Key Request</b>  The system displays the request entry filled by the user as mentioned for these categories in the new state.  The user enters the Appointment Date and Time for the user to make it convenient to visit the branch.  A text box stating any specific instruction to be followed for the request to be processed.  The request moves to Responded state, the customer gets a notification via email and can also view the status update online.
Alternate Flow	Cancel will abandon the operation
Output	Email notification to the customers

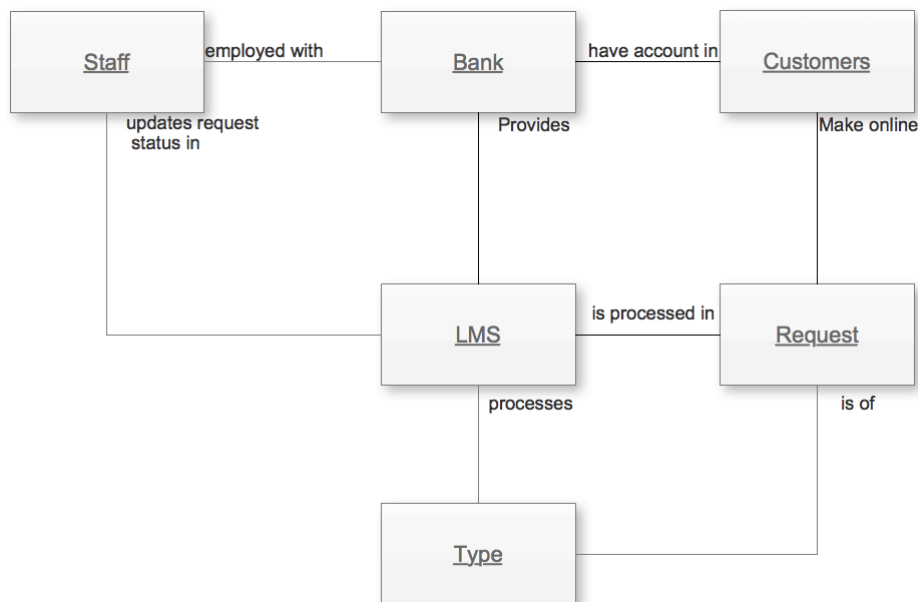
### View Request Status

Use Case Element	Description
Number	UC.05
Application	Customer views the processing status or any activity planned on the request
Use Case Name	View Status
Primary Actor	Customer
Secondary Actor	None
Pre-condition	Successful login into Locker Management application.
Trigger	Customer clicks on menu item View Status on the Landing page
Basic Flow	System displays the requests if any in the chronological ascending order.  The requests record will display the Request Date, Request type, Status, Action Comments, Due on Date.  There is button to acknowledge each request, on click of the button the acknowledgement mail goes to the bank staff queue.
Alternate Flow	None
Output	None

Note : The status update use case here covers the requests in New state. Other states as per the workflow can be automated by the developer if the time permits on similar lines as the New State.

## 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 offers locker management services (LMS) to its existing customers of the core banking operations.
2. The customers can login to the system using their banking account online id and password.
3. The customer can create requests related to Locker management services.
4. The requests are of various types e.g. Apply for New locker, Report Lost Key, Change of Co-Signee, Surrender locker.
5. The bank staff processes the requests in LMS and updates the status online.

## 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
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 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.

Common Checks	Validation Type
RAD - Tackling challenges of software development with Rational Application Developer for WebSphere Software	<a href="http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html">http://www.ibm.com/developerworks/rational/library/08/0926_ackerman-mahate/index.html</a>
IBM Education Assistant - Rational Application Developer 7.5	<a href="http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rad_v7/rad/rad75.html">http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rad_v7/rad/rad75.html</a>
RSA-Overview of Rational Software Architect for WebSphere Software Version 7.5	<a href="http://www.ibm.com/developerworks/rational/library/08/0926_arnold/index.html">http://www.ibm.com/developerworks/rational/library/08/0926_arnold/index.html</a>

Common Checks	Validation Type
Using the new features of UML Modeler in IBM Rational Software Architect Version 7.5	<a href="http://www.ibm.com/developerworks/rational/library/08/0926_diu/index.html">http://www.ibm.com/developerworks/rational/library/08/0926_diu/index.html</a>
Rational Technical Library	<a href="http://www.ibm.com/developerworks/rational/library/">http://www.ibm.com/developerworks/rational/library/</a>