Loan Management System
Internal Design Specification

Version 2.1

Revision History

Date	Version	Description	Author
July 21, 2009	0.02	Software Architecture Document	
July 25, 2009	0.03	Software Architecture Document	

Table of Contents

1. Introduction

The purpose of this document is to outline the architecture and design of the Loan Management System application. It will serve as the work document for developers to implement the solution. The document will describe the overall architecture as well as each individual component.

1. Requirements

The purpose of the LMS application is to gather all customer information and requests in the centralized database.

The main goal of the project is to reduce Total Cost of Ownership (TCO) by the following means:

- LMS should be implemented as Web application to avoid the need to update it with the new version locally.
- It should be implemented using open source frameworks and products only.
- It should handle moderate load with a few concurrent users, but the technology chosen to implement it should be applicable in the case of significant load increase.

1. Architecture

1. *High Level View*

Hight level *physical* architecture of the application is represented on the diagram 3.1

Diagram 3.1

Web Browser is any standard web browser: Internet Explorer, Firefox, Opera or Google Chrome.
It could be run on any supported operating system: Windows, Mac-OS X, Linux etc.

Web Server is a physical server with installed server software stack: server operating system,
Java-enabled web server, all libraries necessary to run LMS application and the application itself.

Communication between Web Browser and Web Server utilize standard HTTP protocol. It could be established over any network infrastructure: Internet or Intranet (internal network of a company).

High level logical view at the internal design of the application is represented on the diagram 3.2

Diagram 3.2

LMS application implemented with proven 3-Tier architecture.

1. 3.1.1.Presentation Layer

Presentation Layer includes all user interface (UI) and its elements: frames, panels, buttons, input fields, menus etc.. It is implemented with JavaScript code generated from Java source code by Google Web Toolkit (GWT) at build time and executed in browser.

1. 3.1.2.Business Logic Layer

Business Logic Layer is implemented with a set of Java components which are deployed on Tomcat servlet container. It consists from components that process incoming requests from browser, business logic components,data access components, transaction support components and some other components. Detailed composition of this layer is described in part 3.2

1. 3.1.3.Data Layer

Data Layer is represented by open source relational database management system MySQL.
Its purpose is to provide reliable data storage, guarantee data integrity and efficiently process requests from the application.

1. *Detailed View*

More detailed composition of the Business Logic Layer is represented on the Diagram 3.3
It show only important type of components, which roles is described below.

Servlet is a standard Java component that can process browser requests itself or delegate it to other components. In the LMS application it serves as an adapter between HTTP request and the rest of Java code. It parses HTTP request which was created by JavaScript RPC call in browser and creates Java objects for further processing. It is implements according to GWT conventions.

Object Mapper serves as dependency management tool and is implemented by Dozer library. It just creates copies of the objects passed by *Servlet* and originated in JavaScript code in browser. This is due to restrictions of GWT library.

Business Logic Components is the core of the application logic. These components integrate
a set of simple operations on data in a single transactional operation, provide exception handling and logging. They utilize Spring Framework transactional services.

Data Access Layer is set of components that simplify development by translating method calls in SQL queries and representing query result in the form of Java objects. These components utilize *Hibernate* Object Relational Mapping framework as well as Spring Framework transactional services.

Diagram 3.3

1. Frameworks

The full list of frameworks and libraries used in the project are listed in the following table

	Name	Version	Class	Sponsor	Link	
1	MySQL	5.1.6	RDBMS	Sun Microsystems	http://www.mysql.com/	

2	Tomcat	5.5	Servlet Container	Apache Foundation	http://tomcat.apache.org/	
3	Spring Framework	2.5.6	IoC Container	Spring Source Inc.	http://www.springsource.org/	
4	Hibernate	3.3.1	ORM Framework	JBoss Inc.	https://www.hibernate.org/	
5	Google Web Toolkit	1.6.4	Cross compiler	Google Inc.	http://code.google.com/webtoolkit/	
6	GWT-EXT	2.0.5	Widget Library	Sanjiv Jivan	http://code.google.com/p/gwt-ext/	
7	Dozer	5.0	Object Mapper	-	http://dozer.sourceforge.net/	

1. ***Description and Purpose***
    1. **MySQL**

MySQL is the most widely used open source *relational database management system* (RDMS).
It runs as a server providing multi-user access to a number of databases. Libraries for accessing MySQL databases are available in all major programming languages. In this project standard JDBC interface is used.

1. **Tomcat**

Tomcat is a Servlet container. It implements Java Servlet and Java Server Pages specification
and provides “pure-Java” HTTP web server environment to run Java web applications.

1. **Spring Framework**

Spring Framework is a *De-facto standard* framework for creating Java applications.

It helps developers to create well designed, testable application of any complexity by encouraging the use of *Inversion of Control/Dependency Injection* (IoC/DI) design pattern.

1. **Hibernate**

Hibernate is the most widely used *object-relational mapping* (ORM) framework.
It's primary purpose is mapping from Java objects to database tables (and from Java data types to
SQL data types). Hibernate generates the SQL calls and relieves the developer from manual result set handling and object conversion, keeping the application portable to all supported SQL databases.

1. **Google Web Toolkit**

Google Web Toolkit (GWT) is a set of tools that allow developers to create cross-browser web application. Using GWT, developers can develop and debug AJAX applications in the Java language using the Java development tools of their choice. When the application is built, the GWT cross-compiler translates the Java source code to optimized JavaScript code for each browser type (Internet Explorer, Firefox, Opera, Chrome)

1. **Database Structure**

1. Data Definition Language (DDL) *Statements*

These are extracts form DDL statements, which shows database table structure.

Table 'EVENTS'

```
EVENT_ID int(10) unsigned NOT NULL AUTO_INCREMENT,
USER_ID int(10) unsigned NOT NULL DEFAULT '0',
TIMESTAMP timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
MESSAGE varchar(1024) NOT NULL,
PRIMARY KEY (EVENT_ID)
```

Table 'USERS'

```
USER_ID int(10) unsigned NOT NULL AUTO_INCREMENT,
USERNAME varchar(16) NOT NULL,
PASSWORD varchar(16) NOT NULL,
FIRST_NAME varchar(32) DEFAULT NULL,
LAST_NAME varchar(32) DEFAULT NULL,
COMPANY varchar(32) DEFAULT NULL,
PHONE varchar(16) DEFAULT NULL,
E_MAIL varchar(32) DEFAULT NULL,
PRIMARY KEY (USER_ID),
UNIQUE KEY I_USERNAME (USERNAME)
```

Table 'CUSTOMERS'

```
CUSTOMER_ID int(10) unsigned NOT NULL AUTO_INCREMENT,
TITLE varchar(4) NOT NULL,
FIRST_NAME varchar(32) NOT NULL,
LAST_NAME varchar(32) NOT NULL,
STREET varchar(32) NOT NULL,"
CITY varchar(32) NOT NULL,"
PROVINCE varchar(3) NOT NULL,"
POSTAL_CODE varchar(7) NOT NULL,"
MONTHLY_INCOME double NOT NULL,"
MONTHLY_DEBT double NOT NULL,
PRIMARY KEY(`CUSTOMER_ID`),
UNIQUE KEY I_NAME (FIRST_NAME ,LAST_NAME)
```

1. *Entity Relationship Diagram*

Entity relationship is represented on the Diagram 5.1

Diagram 5.1

1. User Interface

1. *Login Window*

Login allows the user to login in order to use the application

1. 6.1.1.Access method

- Using URL *dev-hitekschool.com/lms/loans/210{n},* where *n* is build number from 1 to 8
- By clicking 'Logout' button when it is accessible

1. 6.1.2.Window Layout

Fields:

- Branch - TextField, mandatory
- Username - TextField, mandatory
- Password - TextField, mandatory

Buttons:

- Login – validates Username and Password for the given Branch

If Username or Password is wrong for the given Branch error message window is displayed.

If Username and Password are correct for the given Branch Home Page is opened.

1. *Home Page*

1. 6.2.1.Access method

Browser will navigate to the Home Page after successful login.

1. 6.2.2.Main Menu

- Button Users – will navigate to the Users Page
- Button Customers – will navigate to the Customers Page
- Button Settings – is disabled
- Button Policies – is disabled

1. 6.2.3.Navigation Bar

- Action Menu is disabled
- Help Menu:
- Help Topics menu item shows message “Help is not implemented”
- About menu item shows About window

- Logout button invalidates session and navigates to Login window

1. 6.2.4.Search panel

- Search User by Username – TextField
- Search Customer by Last Name – TextField

1. *Users Page*

1. 6.3.1.Access Method

Browser will navigate to the Users Page after clicking on the Users button at the Home Page.

1. 6.3.2.Navigation Bar

Action Menu:

- Add User menu item navigates to the Add User page
- Edit User menu item navigates to the Edit User page
- Delete User menu item invokes delete user command after confirmation

- Help Menu is identical to the Help Menu at the Home Page
- Logout button is identical to the Logout button at the Home Page
- Home button navigates to the Home Page

1. *Add/Edit User Page*

- Username Text Field, mandatory
- Password Text Field, mandatory
- First Name text field

- Last Name text field
- Company text field
- Phone text field
- E-mail text field
- Save button invoke save operation for the user if all field values are valid and closes panel and navigates back to the Users Page
- Cancel button closes panel and navigates back to the Users Page

1. ***Customers Page***

1. 6.5.1.Access Method

Browser will navigate to the Customers Page after clicking on the Customers button at the Home Page.

1. 6.5.2.Navigation Bar

- Action Menu

- Add Customer menu item navigates to the Add Customer page
- Edit Customer menu item navigates to the Edit Customer page
- Delete Customer menu item invokes delete customer operation after confirmation

- Help Menu is identical to the Help Menu at the Home Page
- Logout button is identical to the Logout button at the Home Page
- Home button navigates to the Home Page

1. ***Add/Edit Customer Page***

1. 6.6.1.Customer Personal Data

- Title – select box
- First Name - text field,
- Last Name - text field, mandatory
- Street – text box
- City – text box
- State/Province – text box
- ZIP/Postal Code – text box

1. 6.6.2.Customer Financial Data

- Monthly Income – text box
- Monthly Debt – text box
- Monthly Payments – text box, disabled, calculated

1. 6.6.3.Loan Terms

- Term Years - radio button
- Returned Customer - check box
- Rate - text box, disabled, calculated
- Prequalified for Principal of - text box, disabled, calculated
- At - text box, disabled, calculated