

Customer BRD – A Web Application for Authorizing Loan Disbursal

**PREPARED BY
Vanshika Sharma (2018BTechCSE120)**

**FACULTY GUIDE
Dr. Alok Kumar**



**Department of Computer Science and Engineering
Institute of Engineering and Technology
JK Lakshmipat University, Jaipur**

May 2022

Customer BRD – A Web Application for Authorizing Loan Disbursal

Practice School – 2

Submitted in partial fulfilment of the requirements
for the degree of

Bachelor of Technology in Computer Science and Engineering

PREPARED BY
Vanshika Sharma (2018BTechCSE120)

FACULTY GUIDE
Dr. Alok Kumar



**Department of Computer Science and Engineering
Institute of Engineering and Technology
JK Lakshmipat University, Jaipur**

May 2022

CERTIFICATE

This is to certify that the Practice School-2 project work entitled **Customer BRD – A Web Application for Authorizing Loan Disbursal** submitted by **Vanshika Sharma (2018BTechCSE120)**, towards the partial fulfilment of the requirements for the degree of **Bachelor of Technology in Computer Science and Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached the level required for being accepted for the Practice School-II examination.



Dr. Alok Kumar

Associate Professor,

Department of Computer Science and
Engineering, IET

JK Lakshmipat University, Jaipur



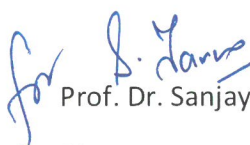
Dr. Gireesh Kaushik

PS2 Coordinator,

Department of Computer Science and
Engineering, IET

JK Lakshmipat University, Jaipur

Date of Submission: May 27, 2022



Prof. Dr. Sanjay Goel

Director

Institute of Engineering & Technology

JK Lakshmipat University, Jaipur

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to *Prof. Sanjay Goel, Director, Institute of Engineering & Technology* for bringing in the opportunity to work with *Nucleus Software Exports Ltd.* and for his valuable time and guidance that made this internship a success.

I take this opportunity to thank my industry mentor, *Ms. Neelam Agarwal*, for trusting me and providing the best of opportunities to grow during my internship tenure. Her exemplary guidance, monitoring, and constant encouragement ignited confidence in me. I would like to express my heartiest thanks to my colleagues and to all those who have helped me directly or indirectly in completing my internship tenure.

I would also like to express my gratitude to thank my faculty mentor, *Dr Alok Kumar*, who has helped me thoroughly with his support, guidance, and inspiration. He has kept me informed of every little detail to help me reach heights by providing the best version of myself to the organization.

Sincerely yours,



Vanshika Sharma (2018BTechCSE120)

ABSTRACT

Software Engineering is a systematic engineering approach to software development. A software engineer is a person who designs, develops, maintains, tests, and evaluates computer software using software engineering concepts. The best method to practice and enhance software skills is internship. This report documents my internship tenure at Nucleus Software Exports Ltd. It highlights various faces of my internship – technical training, team allocation and functional training. NSEL offers banking and lending solutions worldwide. The products and services offered are developed on various platforms like web, mobile and cloud. Every product and service are backed by Java EE – Spring and Hibernate integrated with Oracle database.

The different sections of the report highlight my learning during the technical training and two projects – Lending Management System and Customer BRD, developed in the organization.

Keywords: software engineering, banking, lending, java, spring, hibernate, design patterns

LIST OF FIGURES

S No.	Figure Description	Page No.
1	Nucleus Software's Values	2
2	NSEL Highlights	3
3	FinnOne Neo	4
4	FinnAxia	4
5	LMS Database Schema	7
6	Hibernate Structure	15
7	LMS Class Diagram	17
8	Customer BRD – Screen Navigation Flow Diagram	20

LIST OF ACRONYMS

1. NSEL – Nucleus Software Exports Ltd.
2. JSP – Java Server Pages
3. JPA – Java Persistence API
4. SQL – Structured Query Language
5. PLSQL – Procedural Structured
6. HQL – Hibernate Query Language

TABLE OF CONTENTS

Topic	Page Number
CERTIFICATE	iii
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
LIST OF FIGURES	vi
LIST OF ACRONYMS	vii
TABLE OF CONTENTS	viii
Chapter 1. INTRODUCTION	1
1.1 Objective	1
1.2 Limitations	1
Chapter 2. COMPANY PROFILE	2
2.1 About	2
2.2 Vision	2
2.3 Highlights	3
2.4 Services Offered	3
Chapter 3. TECHNICAL TRAINING	6
3.1 Objective	6
3.2 Structure	6
Chapter 4. STRUCTURED QUERY LANGUAGE – SQL & PLSQL	7
4.1 Structured Query Language – SQL	7
4.1.1 Topics Covered During Training	7
4.1.2 LMS Database Schema	7
4.2 Procedural Structured Query Language – PLSQL	8
4.2.1 Topics Covered During Training	8
Chapter 5. WEB TECHNOLOGIES	9
5.1 HTML	9
5.2 CSS	9
5.3 JavaScript	9
5.4 JQuery	10
Chapter 6. JAVA	11
6.1 Java SE	11
6.1.1 Topics Covered During Training	12
6.2 Java EE	12

6.3 Servlets & JSP	13
6.4 Spring	14
6.5 Hibernate	15
Chapter 7. LENDING MANAGEMENT SYSTEM – MINI PROJECT	17
7.1 Objective	17
7.2 Class Diagram	17
7.3 Functionalities	17
7.4 DESIGN PATTERNS	18
7.4.1 Data Access Object – DAO	18
7.4.2 Singleton Pattern	18
Chapter 8. CUSTOMER BRD – MAJOR PROJECT	19
8.1 Objective	19
8.2 System Flow Diagram	20
8.3 Functionalities	21
8.3.1 Java Persistence API	21
8.3.2 Spring Security	21
8.3.3 Data Validation – JPA Validation	21
8.3.4 SonarLint	22
Chapter 10. CONCLUSION	23
Chapter 11. REFERENCES	24

Chapter 1. INTRODUCTION

1.1 Objective

The objective of internship is to learn and practice industry specific standards in software engineering. The internship tenure at *Nucleus Software Exports Ltd.* provided me the opportunity to learn and grow in the corporate environment, where everything is standardized and done according to the business goals. NSEL offers a variety of products for banking and lending solutions worldwide. The organization has clients across 50+ countries fostering diversity and internationalization both in business management and technical domain. The technology stack across the organization varies vividly following the new age technologies like Spring, Hibernate, DevOps, Artificial Intelligence, Cross Platform App Development, Business Analytics, making the stack robust and diversified. Though the backbone of every product is server and client interaction. To foster that, the organization has a strong and robust Java application support.

The internship broadly categorised into technical training, team allocation interviews followed by functional training on the original product, was an extraordinary experience. The primary objective of this report is to document the entire internship tenure. It focuses on all the technical learning, assignments submitted, projects developed and overall experience.

1.2 Limitations

The limitations of the internship tenure lie in the fact that as trainees or interns we were not subjected to the complete rigorous training done in corporate. As trainees, the scope of work done by us was limited to the basic functionalities drafting. The initial phase focused on learning new age technologies and building mini projects to practice and imply industry standard software engineering practices. In the future, I would like to contribute more towards building core products and services to gain exposure and make learning curve steeper.

Chapter 2. COMPANY PROFILE

2.1 About

Nucleus Software provides lending and transaction banking products to the global financial services industry. As a pioneer in retail and corporate banking software since 1986, the organisation combines deep expertise with an unprecedented track record and a total commitment to building lasting partnerships with our customers. Their software powers the operations of more than 200 Financial Institutions in over 50 countries, supporting retail lending, corporate banking, cash management, mobile and internet banking, automotive finance, and other business areas.

Values:

Nucleus Software has identified five core values - **Integrity, Respect, Result Orientation, Innovation** and **Collaboration** – which form the foundation of our corporate philosophy. From the way staff work together to the way products are developed and foster partnership with the customers to ensure their success, these values underpin everything which is done. The values demonstrate Nucleus Software's commitment to creating a strong corporate culture and long-term partnerships which deliver true value to our customers.

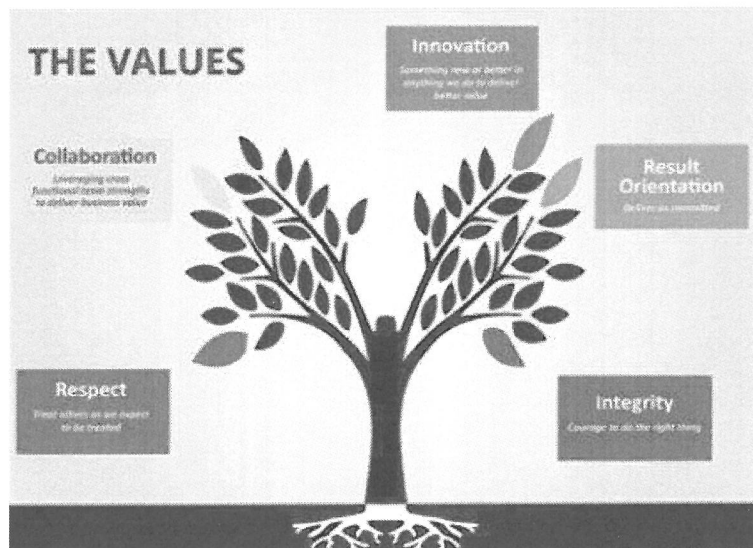


Figure 1: Nucleus Software's Values

2.2 Vision

"Making Financial Services Access Easy and Enriching Worldwide"

2.3 Highlights

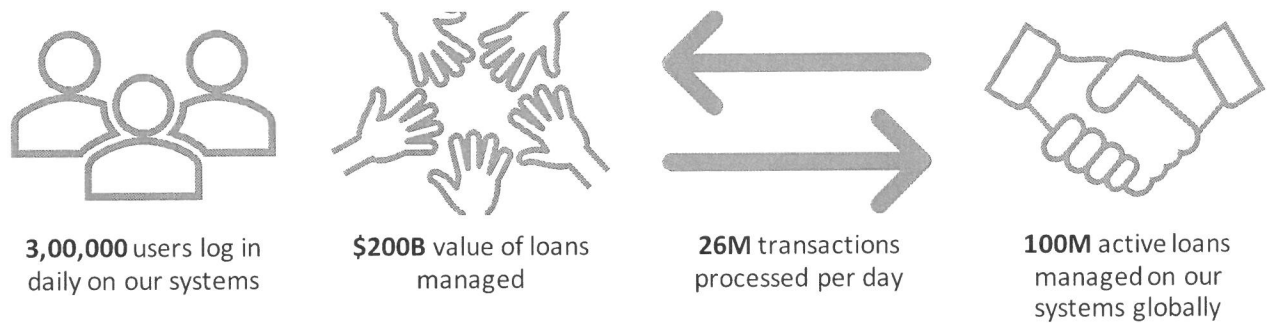


Figure 2: NSEL Highlights

2.4 Services Offered

NSEL offers numerous services to customers globally broadly categorised in 4 channels.

1. Lending

FinnOne Neo is the next-generation lending solution built on an advanced technology platform, designed to shape the future of lending across Retail, Corporate and Islamic sectors. The multi-channel solution helps digitize the complete loan lifecycle end to end, from initial contact with customers and helping make better credit decisions faster to comprehensive loan servicing and sophisticated delinquency management. FinnOne Neo has been designed to meet the challenges of delivering agile and efficient solutions while reducing the cost of operations. With implementations across 50 countries, FinnOne has been recognized as the world's best-selling lending solution for over 10 years.

FinnOne Neo supports both cloud and on-premise deployments. It can be deployed quickly in partnership with leading cloud providers without high upfront capital expenditure. Cloud deployment provides flexibility in scale while optimizing costs by supporting pay-per-use models.

Unique Features:

- Digitization of end to end loan lifecycle
- Ready to be deployed in cloud
- Completely automated and paperless processes
- Dynamic workflow management
- Manages value added products
- Omni-channel capability
- Faster product creation
- Hardware & software agnostic
- Integrated accounting engine and powerful rule engine
- Multi-currency, multi-lingual



Figure 3: FinnOne Neo

2. Transaction Banking

Integrated Transaction Banking Suite – FinnAxia

Nucleus Software's FinnAxia is an integrated global transaction banking solution built on latest Java J2EE technology over Service Oriented Architecture (SOA) platform. With FinnAxia banks can breakdown traditional product silos, launch personalized products/services over multiple channels and achieve operational excellence. FinnAxia, integrated global transaction banking suite, offers the capability to swiftly address corporate customer's liquidity management needs and provides end-to-end solutions from Receivables, Payments, and Liquidity Management to Financial Supply Chain Management, with enhanced customer satisfaction. The solution empowers financial institutions establish and maintain a market-leading presence through the flexible, speedy, and efficient execution of transaction banking processes.



Figure 4: FinnAxia

3. Digital Channels

FinnOne Neo Digital Channels enables banks and financial services companies offer end to end digital lending on mobile and web portal. Powered by integrated modules that can be deployed separately, the solution helps lenders on-board customers faster, deliver better service to customers and simplify collection processes for agents. FinnOne Neo Digital Channels includes mobile applications and web portals for customer acquisition, loan servicing, loan operations, delinquency management and location tracking of field agents. Digital lending capability enables lenders to reduce processing time with paperless

applications, provide instant acknowledgement and resolution of customer queries, launch innovative products faster to market and accept digital payments. With integrations, banks and financial services companies can incorporate automated credit checks, bank statement analysis and customer identity verifications on the go.

We design, develop, and configure computer hardware and software for banks and other organizations in the lending industry.

4. Business Analytics

Predictive Lending Analytics Software: Product Module Overview

To meet changing customer expectations for personalized offerings, and to address a more competitive landscape, banks and other financial services companies have taken advantage of rapid changes in technology to kick start their digital transformation journey. However, offering digitized, automated, and multi-channel loan operations will soon become “table stakes”. To stay ahead of the market and prepare themselves for the next wave of transformation, lenders need to quickly adopt advanced technologies, particularly in analytics.

With cutting edge Machine Learning and Artificial Intelligence capabilities such as text analytics and neural networks, Nucleus Lending Analytics is an easy-to-build platform that helps you make better credit decisions through accurate predictive models. The solution also helps to configure existing decision models such as scorecards and business rules, thereby allowing lenders to monitor their performance and manage the risks associated with their models. Nucleus Lending Analytics addresses key business drivers including generating higher growth by targeting the right customers with customized offerings, increasing profitability by reducing costs and customer churn, enhancing quality of portfolio by enabling strategic decisioning throughout the loan life cycle.

Nucleus Lending Analytics Advantage:

1. **Convenient:** Data science made easy for analysts & business users alike
2. **Fast:** Build the machine learning models and complex business rules with faster turnaround time
3. **Specialized:** Designed for Lending business incorporating 3 decades of domain expertise

Chapter 3. TECHNICAL TRAINING

3.1 Objective

The objective of the technical training was to impart industrial level knowledge of core technologies like SQL, Core Java, Java EE – Servlet, JSP, Spring, Hibernate and many more. The duration of the training was 2.5 months seamlessly divided into 5 sections focusing on main 5 technologies. The highlight of the technical training was topic wise assignments. During the tenure, I submitted nearly 20+ assignments covering all the main topics. All the assignments were a part of **LMS – Mini Project** broken down by different topics and technologies.

3.2 Structure

The structure of the entire duration of technical training focused on several industrial fed technologies. The course plan is as follows:

Table: Technical Training Course Plan		
S. No.	Topic	Days
1	SQL	6
2	PLSQL	5
3	Core Java	19
4	Web Technologies	10
5	Spring Hibernate	15

Chapter 4. STRUCTURED QUERY LANGUAGE – SQL & PLSQL

4.1 Structured Query Language – SQL

Structured Query Language (SQL) is an acronym for Structured Query Language. SQL is a language for interacting with databases. It is the standard language for relational database management systems, according to ANSI (American National Standards Institute). SQL statements are used to conduct operations like updating data in a database and retrieving data from one. Oracle, Sybase, Microsoft SQL Server, Access, Ingres, and other relational database management systems that use SQL are just a few examples.

4.1.1 Topics Covered During Training

Following topics were covered in SQL during training:

1. Introduction to RDBMS
2. SQL Languages – DDL, DML, DCL, DQL, TCL
3. Joins
4. Subqueries
5. Views
6. Index
7. Synonyms

4.1.2 LMS Database Schema

The training focused on building the **Database Schema of Lending Management System**.

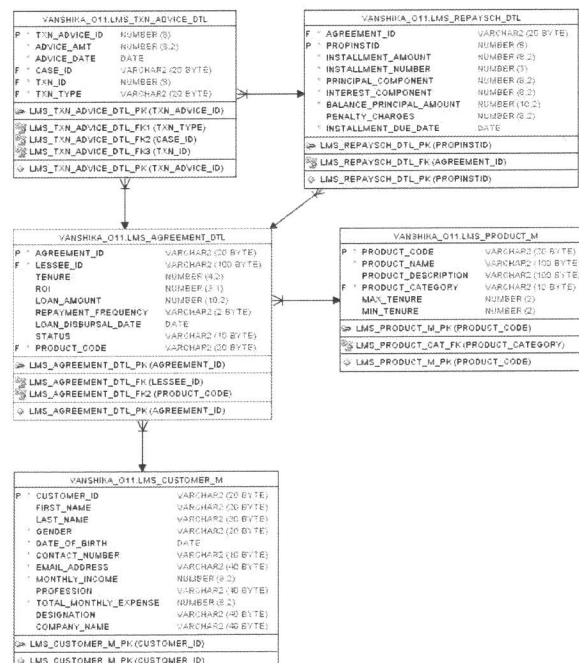


Figure 5: LMS Database Schema

4.2 Procedural Structured Query Language – PLSQL

"Procedural Language Extensions to the Structured Query Language" is what PL/SQL stands for. SQL is a widely used language in relational database management systems for querying and updating data (RDBMS). To circumvent some of SQL's shortcomings, PL/SQL adds additional procedural elements to the language. Furthermore, PL/SQL is a more comprehensive programming language for developing mission-critical Oracle Database applications. PL/SQL is a well-organized and legible programming language. Its constructs succinctly communicate the code's intent. In addition, PL/SQL is a simple language to master.

4.2.1 Topics Covered During Training

Following topics were covered in PLSQL during training:

1. Blocks – Named & Anonymous
2. Procedures & Functions
3. Cursors
4. Exception Handling
5. Collections
6. Packages
7. Triggers

Chapter 5. WEB TECHNOLOGIES

5.1 HTML

Hypertext Markup Language - HTML is the fundamental language of web. It defines the flow, meaning and structure of web content.

The term *hypertext* refers to links that connect online pages inside a single website or between websites. Links are an important part of the Internet. You become an active participant in the World Wide Web by uploading content to the Internet and linking it to other people's pages.

HTML makes use of *markup* to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, , , <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, , , and many others.

5.2 CSS

CSS (Cascading Style Sheets) is a stylesheet language for describing the presentation of an HTML or XML document (including XML dialects such as SVG, MathML or XHTML). CSS specifies how elements should appear on a screen, on paper, in speech, or in other forms of media. According to W3C guidelines, CSS is one of the basic languages of the open web and is standardised across Web browsers. Previously, distinct portions of the CSS specification were developed simultaneously, allowing for versioning of the most recent recommendations.

5.3 JavaScript

JavaScript is a scripting language that allows you to create dynamic computer programmes. It is a lightweight component of web pages whose implementations enable client-side script to interact with users and create dynamic sites. It's an object-oriented programming language that can be interpreted.

A standard version of the basic JavaScript language was defined by the ECMA-262 Specification. The features of JS are:

- JavaScript is an interpreted, lightweight programming language.
- Create network-centric apps with this framework.
- Java complements and integrates with it.
- HTML-complementary and HTML-integrated.
- Open and platform-agnostic

5.4 JQuery

JQuery is a feature-rich JavaScript library that is quick, compact, and cross-platform. Its purpose is to make HTML client-side scripting easier. With an easy-to-use API that works on a variety of browsers, it makes tasks like HTML document traversal and manipulation, animation, event handling, and AJAX relatively straightforward. The primary goal of jQuery is to make it simple to use JavaScript on your website to make it more interactive and appealing. It is also utilised to add movement.

JQuery Features:

- It can also be used to simplify a lot of the more difficult aspects of JavaScript, such as AJAX calls and DOM manipulation.
- It is a JavaScript library that is compact, fast, and light.
- It is platform agnostic.
- It translates to "write less, do more."

In the major project – Customer BRD, JQuery DataTable have been used extensively.

JQuery DataTable:

DataTables is a plug-in for the jQuery Javascript library. It is a highly flexible tool, built upon the foundations of progressive enhancement, that adds all of these advanced features to any HTML table.

Features:

1. Pagination
2. Instant search
3. Multi-column ordering
4. Compatible with different data sources
5. Vivid themes
6. Wide variety of extensions
7. Mobile friendly
8. Fully internationalisable
9. Free open source software

Chapter 6. JAVA

6.1 Java SE

The Java SE is a computing-based platform and used for developing desktop or Window based applications. Thus, core Java is the part of Java SE where the developers develop desktop-based applications by using the basic concepts of Java where JDK (Java Development Kit) is a quite familiar Java SE implementation.

Features of Java:

Object Oriented: In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

Platform Independent: Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform-independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

Simple: Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

Secure: With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

Architecture-neutral: Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

Portable: Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

Robust: Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking.

Multithreaded: With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

Interpreted: Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

High Performance: With the use of Just-In-Time compilers, Java enables high performance.

Distributed: Java is designed for the distributed environment of the internet.

Dynamic: Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.

A heavily used concept of Java, which forms the base of OOPS and several design patterns is **Interface**.

An Interface in Java programming language is defined as an abstract type used to specify the behaviour of a class. An interface in Java is a blueprint of a class. A Java interface contains static constants and abstract methods.

The interface in Java is a mechanism to achieve abstraction. There can be only abstract methods in the Java interface, not the method body. It is used to achieve abstraction and multiple inheritance in Java. In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body. Java Interface also represents the IS-A relationship.

In Java 8, Interfaces have been flourished with features like Dynamic functions.

Several design patterns like DAO, FACADE, Functional Programming and many more important concepts and features have been developed and designed using interfaces.

6.1.1 Topics Covered During Training

1. Introduction and Fundamentals of Java
2. Classes & Objects
3. Inheritance & Composition
4. Exception Handling
5. Packages
6. Inner Classes
7. Functional Programming
8. Data Structure
9. Collection Framework
10. Multithreading
11. IO Programming
12. JDBC
13. Junit
14. Maven

6.2 Java EE

Java Platform, Enterprise Edition (Java EE) is the standard in community-driven enterprise software. Java EE is developed using the Java Community Process, with contributions from industry experts, commercial and open-source organizations, Java User Groups, and countless individuals. Each release integrates new features that align with industry needs, improves application portability, and increases developer productivity. Today, Java EE offers a rich enterprise software platform and with over 20 compliant Java EE implementations to choose from.

Java EE 8 improves APIs and programming paradigms for today's applications, as well as adding capabilities sought by our global community. This release updates compatibility for several industry standards while also simplifying enterprise-ready APIs.

Enhancements include:

- Java Servlet 4.0 API with HTTP/2 support
- Enhanced JSON support including a new JSON binding API
- A new REST Reactive Client API
- Asynchronous CDI Events
- A new portable Security API
- Server-Sent Events support (Client & Server-side)
- Support for Java SE 8 new capabilities (e.g. Date & Time API, Streams API, annotations enhancements)

6.3 Servlets & JSP

A **servlet** is a Java programming language class that is used to improve the capabilities of servers that host request-response programming model applications. Servlets can react to any form of request, but they are most typically used to extend web server-hosted applications. Java Servlet technology defines HTTP-specific servlet classes for such applications.

Interfaces and classes for writing servlets are provided by the *javax.servlet* and *javax.servlet.http* packages. The Servlet interface, which provides life-cycle methods, must be implemented by all servlets. You can utilise or extend the *GenericServlet* class provided with the Java Servlet API to provide generic services, whereas the *HttpServlet* class includes methods for handling HTTP-specific services, such as *doGet* and *doPost*.

JSP is a Java standard technology that allows you to create dynamic, data-driven pages for your Java web applications. The Java Servlet specification is the foundation for JSP. Both technologies are frequently used simultaneously, particularly in older Java web applications. The most obvious difference between them in terms of coding is that servlets start with client-side script or markup and then embed JSP tags to connect your page to the Java backend, whereas JSP starts with client-side script or markup and then embed JSP tags to connect your page to the Java backend.

Features of JSP:

- JSP coding is simple: It is just adding JAVA code to HTML/XML.
- Code length reduction: In JSP, we employ action tags, custom tags, and so on.
- Easier database connection: It's simpler to connect a website to a database, and it's easier to read and write data to the database.
- Create interactive websites: We can do this by creating dynamic web pages that allow users to participate in real time.
- These are browser and server agnostic, making them portable, powerful, adaptable, and easy to manage.

- No Re-Compilation or Redeployment: It is dynamic, secure, and platform independent, therefore there is no need to re-compile.
- Servlet Extension: It has all the functionality of servlets, including implicit objects and custom tags.

6.4 Spring

On any deployment platform, the Spring Framework provides a comprehensive programming and configuration architecture for modern Java-based enterprise applications. Spring's infrastructure support at the application level is a crucial feature: it focuses on "plumbing" enterprise applications so that teams may focus on application-level business logic without having to worry about specific deployment settings.

Features

- Dependency injection, events, resources, i18n, validation, data binding, type conversion, SpEL, and AOP are all core technologies.
- Mock objects, the TestContext framework, Spring MVC Test, and WebTestClient are all used in testing.
- Transactions, DAO support, JDBC, ORM, and XML marshalling are all examples of data access.
- The web frameworks Spring MVC and Spring WebFlux.
- Remoting, JMS, JCA, JMX, email, tasks, scheduling, and cache are all examples of integration.
- Dynamic languages, such as Kotlin and Groovy are supported.

Spring Flexibility

Spring's flexible and comprehensive set of extensions and third-party libraries let developers build almost any application imaginable. At its core, Spring Framework's Inversion of Control (IoC) and Dependency Injection (DI) features provide the foundation for a wide-ranging set of features and functionality.

Spring Security

Spring has a proven track record of dealing with security issues quickly and responsibly. The Spring committers work with security professionals to patch and test any reported vulnerabilities. Third-party dependencies are also monitored closely, and regular updates are issued to help keep your data and applications as safe as possible. In addition, Spring Security makes it easier for you to integrate with industry-standard security schemes and deliver trustworthy solutions that are secure by default.

6.5 Hibernate

Hibernate ORM (or just Hibernate) is a Java programming language object–relational mapping tool. It provides a framework for translating a relational database to an object-oriented domain model. Hibernate solves object–relational impedance mismatch issues by using high-level object handling routines instead of direct, permanent database calls.

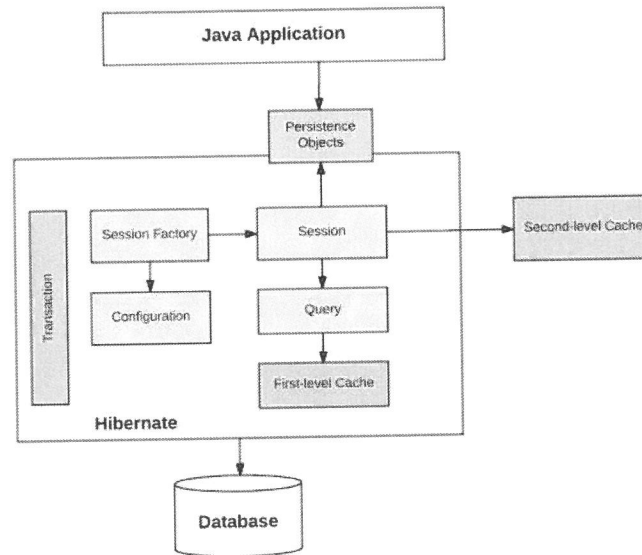


Figure 6: Hibernate Structure

Advantages of Hibernate Framework:

1. **Open Source and Lightweight:** Hibernate framework is open source under the LGPL license and lightweight.
2. **Fast Performance:** The performance of hibernate framework is fast because cache is internally used in hibernate framework. There are two types of cache in hibernate framework first level cache and second level cache. First level cache is enabled by default.
3. **Database Independent Query:** HQL (Hibernate Query Language) is the object-oriented version of SQL. It generates the database independent queries. So you don't need to write database specific queries. Before Hibernate, if database is changed for the project, we need to change the SQL query as well that leads to the maintenance problem.
4. **Automatic Table Creation:** Hibernate framework provides the facility to create the tables of the database automatically. So, there is no need to create tables in the database manually.
5. **Simplifies Complex Join:** Fetching data from multiple tables is easy in hibernate framework.

6. **Provides Query Statistics and Database Status:** Hibernate supports Query cache and provide statistics about query and database status.

Hibernate Query Language - HQL

Hibernate provides the Hibernate Query Language (HQL) as a SQL-inspired language for writing SQL-like queries on Hibernate's data objects. As an object-oriented alternative to HQL, Criteria Queries are given. Criteria Query is used to change the items and provide restrictions on them. The object-oriented equivalent of SQL is HQL (Hibernate Query Language). It generates database-agnostic queries, eliminating the requirement for database-specific queries. Without this functionality, altering the database would necessitate changing individual SQL queries, causing maintenance concerns.

Topics Covered During Training

1. ORM Principles
2. Core API – Session, SessionFactory, Transaction, TransactionFactory, ConnectionProvider, Query
3. Entity
4. Hibernate Dialect
5. Mapping – XML, JPA
6. Entity Instance States
7. Inheritance Mapping
8. Collection Mapping
9. Association Mapping

4. **Junit** testing
5. **Maven** used to build and package the application

7.4 DESIGN PATTERNS

7.4.1 Data Access Object – DAO

The DAO pattern is used to implement the Persistent layer. It usually starts with an interface with generic methods and implementation is provided by corresponding classes as per needs. This approach brings **Loose Coupling** so that the application can switch between persistence layers with minimal changes.

7.4.2 Singleton Pattern

Singleton Pattern follows the definition of a class that has only one instance and provides a global point of access to it. There are two forms of singleton design pattern

- Early Instantiation: creation of instance at load time.
- Lazy Instantiation: creation of instance when required.

Advantage of Singleton Design Pattern lies in the fact that it saves memory because object is not created at each request. Only single instance is reused again and again.

Chapter 8. CUSTOMER BRD – MAJOR PROJECT

8.1 Objective

The objective of project was to develop a spring application to facilitate the customer database. A customer needs to be authorized by specific user having role “checker” to proceed for the loan application. The user with roles “maker” has more privileges like creating, modifying, deleting records of customer data. The “temporary” table contains all the data irrespective of authorization or rejection of record. The “master” table contains only authorized customer records done by a checker.

The technology stack of the project is listed below:

1. Database: Oracle Express Edition
2. Server-Side Scripting: Spring, Hibernate
3. Client-Side Scripting: Java Server Pages (JSP), HTML, CSS, JavaScript, JQuery
4. Security: Spring Security

8.2 System Flow Diagram

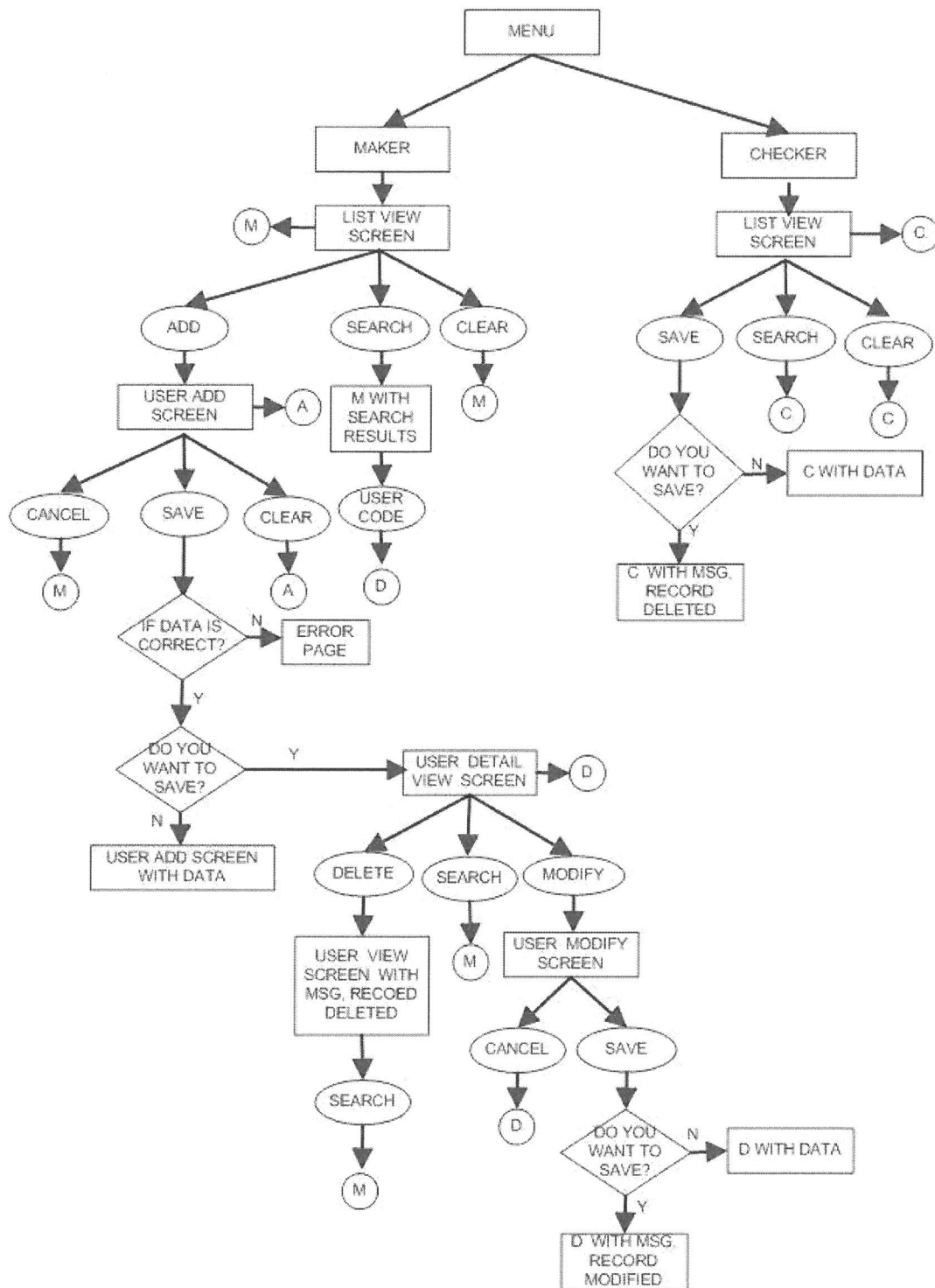


Figure 8: Screen Navigation Flow Diagram

8.3 Functionalities

The following functionalities were provided in the project:

1. A multi-layered architecture with clear separation of front-end, controller logic, business logic, and database layer. Achieved using MVC.
2. Spring Security focusing on user login
3. Interactive user dashboard for both maker and checker
4. Proper transaction management done using JPA Transaction Manager.
5. DAO support for implementing database interaction.
6. Hibernate Caching achieved using lazy and eager loading.

8.3.1 Java Persistence API

The Java Persistence API provides Java developers with an object/relational mapping facility for managing relational data in Java applications. Java Persistence consists of four areas:

- The Java Persistence API
- The query language
- The Java Persistence Criteria API
- Object/relational mapping metadata

8.3.2 Spring Security

Spring Security is an authentication and access-control framework that is both robust and very flexible. For protecting Spring-based apps, it is the de-facto standard.

Spring Security is a framework for providing Java applications with both authentication and permission. The actual value of Spring Security, like all Spring projects, is in how readily it can be modified to meet specific requirements.

Features:

1. Comprehensive and extensible support for both authentication and authorization.
2. Support provided for session fixation, clickjacking, cross site request, forgery and many more.
3. Servlet API Integration
4. Spring Web MVC support

8.3.3 Data Validation – JPA Validation

Validation of data is a typical job that occurs throughout an application's layers, including persistence. The Bean Validation API is supported by the JavaTM Persistence API (JPA) 2.0, allowing data validation to be performed at runtime.

The following JPA validation were used in project

1. Not Null

2. Size
3. Assert True
4. Email

8.3.4 SonarLint

It is Free and Open-Source IDE extension that identifies and helps to fix quality and security issues as we code. It helps in bug detection, instant feedback, and great recommendations.

Its categories each issue in three categories:

1. Code-Smell (A maintainability issue that makes the code confusing and difficult to maintain.)
2. Bug (A coding mistake that can lead to an error or unexpected behaviour at runtime)
3. Vulnerability (A point in the code that is open to attack)

SonarLint works on protocols, which are based on Industry Specific Language/Framework Standard Coding Conventions. The extension also provides a detailed description of the issue.

Rule Description includes:

1. Category of the bug
2. Detailed problem statement and solutions
3. Examples of Compliant and Non-Compliant code blocks

Chapter 10. CONCLUSION

The internship tenure at Nucleus Software Exports Ltd. was an enriching experience. The internship was divided in 3 phases of **technical training, team allocation interview process** and **functional training**. The main technology learnt and practice throughout internship tenure was Java. From Core Java to Advanced Java, concepts were thoroughly preached. The technical training focused on technologies like Java, SQL, Spring and Hibernate. After the completion of technical training many interviews were conducted for business unit allocation. Followed by the functional training wherein product overview was given. Functional training focused on building organisation setup, product, schemes eventually leading upto applying for home loan.

NSEL provides products and services in the domain of banking and lending. The training was focused upon integrating software practices with the financial domain. Henceforth the learning journey in NSEL was an amalgamation of industry software practices in new age technologies and financial domain focused upon banking and lending.

Chapter 11. REFERENCES

- [1] Farrell, J. (2011). *Java programming*. Cengage Learning.
- [2] Hristova, M., Misra, A., Rutter, M., & Mercuri, R. (2003). Identifying and correcting Java programming errors for introductory computer science students. *ACM SIGCSE Bulletin*, 35(1), 153-156.
- [3] Reese, G. (2000). *Database Programming with JDBC and JAVA*. "O'Reilly Media, Inc."
- [4] Singh, R., Bezemer, C. P., Shang, W., & Hassan, A. E. (2016, March). Optimizing the performance-related configurations of object-relational mapping frameworks using a multi-objective genetic algorithm. In *Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering* (pp. 309-320).
- [5] Johnson, R., Hoeller, J., Donald, K., Sampaleanu, C., Harrop, R., Risberg, T., ... & Webb, P. (2004). The spring framework—reference documentation. *interface*, 21, 27.
- [6] Xia, C., Yu, G., & Tang, M. (2009, December). Efficient implement of orm (object/relational mapping) use in j2ee framework: Hibernate. In *2009 International Conference on Computational Intelligence and Software Engineering* (pp. 1-3). IEEE.
- [7] Bikker, J. A., & Haaf, K. (2002). Competition, concentration and their relationship: An empirical analysis of the banking industry. *Journal of banking & finance*, 26(11), 2191-2214.
- [8] Berger, A. N. (2003). The economic effects of technological progress: Evidence from the banking industry. *Journal of Money, credit and Banking*, 141-176.
- [9] Yan, J., Yu, W., & Zhao, J. L. (2015). How signaling and search costs affect information asymmetry in P2P lending: the economics of big data. *Financial Innovation*, 1(1), 1-11.

INTERNSHIP CERTIFICATE

Date: 24/05/2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Vanshika Sharma** has done her internship as Assistant Software Engineer Trainee at Nucleus Software, Noida, from January 12, 2022 to May 30, 2022 for the duration of 4.5 months.

She has worked on a project titled **BRD Maker Checker**. This project was aimed at developing a web-based application using Spring & Hibernate. As part of the project, she has created maker checker dashboards with database integration.

During her internship she has demonstrated her skills with self-motivation to learn new skills. Her performance exceeded our expectations, and she was able to complete the project on time.

We wish her all the best for her upcoming career.

Nucleus Software Exports Ltd.

Ms. Neelam Agarwal

Associate Manager, NSBT


26/5/22

[Signature][Seal of Company]