TABLE OF CONTENTS

CHAPTER NO. TITLE PAGE NO.

ABSTRACT

LIST OF FIGURES

LIST OF ABBREVATIONS

1 INTRODUCTION

1.1 OBJECTIVE OF THE PROJECT

2 SYSTEM SPECIFICATION

2.1 HARDWARE REQUIRMENTS

2.2 SOFTWARE REQUIREMENTS

3 SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

3.2 PROPOSED SYSTEM

4 TECHNOLOGIES USED

4.1 JAVA

4.2 ECLIPSE

4.3 ANALYSED TECHNOLOGIES

4.3.1 XMLs

4.3.2 SPRING

4.3.3 JAXB

4.3.4 HIBERNATE

4.3.5 MAVEN

4.3.6 HAZELCAST

5 SYSTEM DESIGN

6 SYSTEM DEVELOPMENT

6.1 LIST OF MODULES

6.2 MODULE DESCRIPTION

6.2.1 QUEUES

6.2.2 PARSER PROGRAM

6.2.3 JAXB PROCESSING

6.2.4 HIBERNATE PROGRAM

6.2.5 DATABASE

6.2.6 SQL LOADING PROCESS

7 SYSTEM IMPLEMENTATION

8 CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

8.2 FUTURE ENHANCEMENT

9 SCREENSHOTS

10 REFERENCES

INTRODUCTION

EFM Operational and Business Intelligence (OBI) is a set of near real-time Financial Messaging Data Marts (DM) for EFM’s (Enterprise Financial Messaging) systems. Since EFM’s primary reason is messaging, the DMs are a repository for information related to financial messages that flow through EFM.

A data mart is the access layer of the data warehouse environment that is used to get data out to the users. The data mart is a subset of the data warehouse that is usually oriented to a specific business line or team.

OBJECTIVE OF THE PROJECT

The main objective of this project is to understand XML Message Parsing and Ways to Load it into Database using Hibernate.

Other objectives are while reading data and writing to database the speed should be greater than forty messages per second from the queue. Multithreading is implemented so that concurrency can be achieved. XML should load from in-memory cache so that time can be saved. This is because reading and writing from memory is faster than reading and writing from disc. The system should not fail in case of changes in the database table. The overall objective of this project is to reduce the expenses for reading messages.

SYSTEM SPECIFICATION

HARDWARE REQUIREMENTS

Processor : Any processor above 500MHz

Ram : 128 Mb

Hard disk : 10 GB

Compact disk : 650 Mb

Input device : Standard Keyboard and Mouse

Output device : Display screen

SOFTWARE REQUIREMENTS

Operating System : Windows Family

Language : Java

Front End : Eclipse

Back End : Oracle SQL Developer

TECHNOLOGIES USED

JAVA

Java is a computer programming language that is concurrent, class based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers “write once, run anywhere” (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture.

Java is a simple, object-oriented, network-savvy, interpreted, robust, secure, architecturally neutral, portable, high-performance, multithreaded, dynamic language.

ECLIPSE

In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications.

By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: Ada, C, C++, COBOL. It can also be used to develop packages. Development environments include the Eclipse Java development tools (JDT) for Java.

ANALYSED TECHNOLOGIES

XMLs

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. It is defined by the W3C's XML 1.0 Specification and by several other related specifications, all of which are free open standards.

The design goals of XML emphasize simplicity, generality and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while many application programming interfaces (APIs) have been developed to aid the processing of XML data.

SPRING

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an alternative to, replacement for, or even addition to the Enterprise JavaBeans(EJB) model. The Spring Framework is open source.

JAXB

Java Architecture for XML Binding (JAXB) allows Java developers to map Java classes to XML representations. JAXB provides two main features: the ability to marshal Java objects into XML and the inverse, i.e. to unmarshal XML back into Java objects. In other words, JAXB allows storing and retrieving data in memory in any XML format, without the need to implement a specific set of XML loading and saving routines for the program's class structure

JAXB is particularly useful when the specification is complex and changing. In such a case, regularly changing the XML Schema definitions to keep them synchronised with the Java definitions can be time consuming and error-prone.

JAXB is a part of the Java SE platform and one of the APIs in the Java EE platform, and is part of the Java Web Services Development Pack (JWSDP). It is also one of the foundations for WSIT.

HIBERNATE

Hibernate ORM (Hibernate in short) is an object-relational mapping framework for the Java language. It provides a framework for mapping an object-oriented domain model to a relational database. Hibernate solves object-relational impedance mismatch problems by replacing direct, persistent database accesses with high-level object handling functions.

Hibernate is free software that is distributed under the GNU Lesser General Public License 2.1.

Hibernate's primary feature is mapping from Java classes to database tables; and mapping from Java data types to SQL data types. Hibernate also provides data query and retrieval facilities. It generates SQL calls and relieves the developer from manual handling and object conversion of the result set.

MAVEN

Maven is a build automation tool used primarily for Java projects. The word maven means "accumulator of knowledge". Maven addresses two aspects of building software: First, it describes how software is built, and second, it describes its dependencies. Contrary to preceding tools like Apache Ant, it uses conventions for the build procedure, and only exceptions need to be written down. An XML file describes the software project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins. It comes with pre-defined targets for performing certain well-defined tasks such as compilation of code and its packaging.

Maven dynamically downloads Java libraries and Maven plug-ins from one or more repositories such as the Maven 2 Central Repository, and stores them in a local cache.  This local cache of downloaded artifacts can also be updated with artifacts created by local projects. Public repositories can also be updated.

The Maven project is hosted by the Apache Software Foundation.

Maven is built using a plugin-based architecture that allows it to make use of any application controllable through standard input. Theoretically, this would allow anyone to write plugins to interface with build tools (compilers, unit test tools, etc.) for any other language. In reality, support and use for languages other than Java has been minimal

HAZELCAST

In computing, Hazelcast is an open source in-memory data grid based on Java.

In a Hazelcast grid, data is evenly distributed among the nodes of a computer cluster, allowing for horizontal scaling both in terms of available storage space and processing power. Backups are also distributed in a similar fashion to other nodes, based on configuration, thereby protecting against single node failure.

Typical use-cases for Hazelcast include:

* Distributed cache, often in front of a database
* storage for temporal data, like web sessions
* in-memory data processing and analytics
* Cross-JVM communication and shared storage

MODULE DESCRIPTION

DatabaseHibernate ProgramParser ProgramQueues

This provides a detailed information about each of the modules and its supported components.

QUEUES

A queue is an example of a linear data structure, or more abstractly a sequential collection. A collection designed for holding elements prior to processing. Besides basic [Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html) operations, queues provide additional insertion, extraction, and inspection operations. Each of these methods exists in two forms: one throws an exception if the operation fails the other returns a special value (either null or false, depending on the operation). Queues typically, but do not necessarily, order elements in a FIFO (first-in-first-out) manner.

Queues are implemented as containersadaptors, which are classes that use an encapsulated object of a specific container class as its underlyingcontainer, providing a specific set of member functions to access its elements. Elements are pushed into the *"*back*"* of the specific container and popped from its *"*front*"*.

PARSER PROGRAM

XML Parser provides way how to access or modify data present in an XML document. Java provides multiple options to parse XML document. There are JAXB and XSLT APIs available to handle XML parsing in Object Oriented way.

JAXB PROCESSING

HIBERNATE PROGRAM

DATABASE

SQL LOADING PROCESS