DIGITIZED SALES SYSTEM

MID-TERM MAJOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY
(Computer Science and Engineering)

Submitted By:
Anurag Kumar (1326453)
Pankaj (1323491)

Submitted To.:
Prof. Harkomal Preet Kaur
Assistant Professor, CSE

The Computer Science Department
Swami Vivekanand Institute of Engineering & Technology
Chandigarh-Patiala highway, sector-8, Ramnagar, Banur

"2017"

Abstract

Digitized Sales System is a web application which is used to sell and buy products. The idea of this project is to study the retailer's current working. And propose a solution in order to digitize their current processes and overcome the current issues which are being faced daily due to lack of computerized solution. This need of digitization of their current processes related to order handling will help them in forecasting their business growth.

The customer places an order through a website which is examined and later accepted by retailer/seller. Our Application is looking for a potential supplier against each customer order and sends an invoice which is generated by the system, as it is termed as a mini agreement between Our Application and supplier. As an initial step of the project, the understanding of retailer's current workflow is built. After that the system requirements were identified and documented. Theoretical review of similar systems was made.

This project will work as a 24-hours open retail center for different people. Where customers can buy & sell their belongings or retailers can sell and stock products under them. Even some new innovative features are introduced in this project involving auctioning of project and trading between materials.
.

Acknowledgment

We are highly grateful to Mrs. Kawaljit Kaur, HOD CSE Department, Swami Vivekanand Institute of Eng. & Technology (SVIET), Banur, for providing this opportunity to carry out the major project work at KVCH-IBMCE, Preet Vihar, New Delhi.

The constant guidance and encouragement received from Mrs. Kawaljit Kaur, H.O.D. CSE Department, SVIET Banur has been of great help in carrying out the project work and is acknowledged with reverential thanks. We would like to express a deep sense of gratitude and thanks profusely to Mr. Tanay Kishore Mishra(Cloud Computing) & Mrs. Ruby(Java), without her wise counsel and able guidance, it would have been impossible to complete the project in this manner.

We express gratitude to other faculty members of computer science and engineering department of SVIET for their intellectual support throughout the course of this work.

Finally, We are indebted to all whosoever have contributed in this report work.

ANURAG KUMAR

PANKAJ

Table of Content

Content	Page no.
Introduction	5
Requirement Analysis & System Specification	9
System Design	10
Implementation	13
Biblography	25

Introduction

Introduction to Project

In the emerging global economy, **e-commerce** have increasingly become a necessary component of business strategy and a strong catalyst for economic development. The integration of information and communications technology (**ICT**) in business has revolutionized relationships within organizations and those between and among organizations and individuals. Specifically, the use of **ICT** in business has enhanced productivity, encouraged greater customer participation, and enabled mass customization, besides reducing costs.

With developments in the Internet and Web-based technologies, distinctions between traditional markets and the global electronic marketplace-such as business capital size, among others-are gradually being narrowed down. The name of the game is strategic positioning, the ability of a company to determine emerging opportunities and utilize the necessary human capital skills (such as intellectual resources) to make the most of these opportunities through an e-business strategy that is simple, workable and practicable within the context of a global information milieu and new economic environment. With its effect of leveling the playing field, e-commerce coupled with the appropriate strategy and policy approach enables small and medium scale enterprises to compete with large and capital-rich business**s.**

Scope of this project is to investigate and design a software solution which can facilitate retailers in performing their daily tasks, improving efficiency, and helping them to be more productive. This project will provide a solution through which retailers can easily manage, handle and generate all required information in their respective format when needed. It will help them to manage order details, financial data, and historical data and also in producing documents of different formats for different customers. This solution will help retailers in reducing effort spend on managing orders. It will also provide them opportunity to explore possibility of generating documents, managing financial details and analyzing historical data with use of digitized solution.

Project Category

This project is an Internet Based JAVA Application Development. We are using J2EE technology for development of this project.

Objectives

- Invoice Generation.
- Sending order to supplier.
- Creating customer invoice.
- Generating documents for bank (bill of lading).
- Sending Bill of lading to customer.
- Customer Management.
- Order Trace-ability.
- Financial Details management.

Problem Formulation

In today's fast paced society, it's very hard to be competitive without using cutting-edge technology available in market. After years of business, the data has grown much for retailers. It is becoming a challenge for retailers to manage that data in an effective way. To be more productive in order processing, retailers needs a solution which can facilitate their current processes with use of technology and software.

With increased amount of orders, it is becoming difficult for retailers to manage orders in effective and efficient manner. It is very hard to go through all paper work and backtracking orders. If there is any complain or review of any order, it takes large amount of effort and time to backtrack and fix the problem. This results in loss of resources, increased time, and low output.

Workflow from order quotes, order to invoice and payments are today made manually without the help of a computerized management system. This means a lot of manual work, which leads to the loss of control over operations. Due to higher workloads and more errors, delay in the whole process is experienced on daily basis. No database exists and thus poor ability to pick out statistics on for example the existing order stock.

Identification/Reorganization of Need

As the market and retailer is a very old business after-all, there are many standard workflows that are available for evaluation. Some general practices in retailer market that we consider as general issues to be identified are mentioned below:

- lengthy paper-work for tracing every order and its transaction.
- Availability of a product needs to be evaluated in real-time else leading to inconvenience to the customer.
- Lots of banking documentation for individual product and retailers collection. Customer needs to be provided proper invoice for the product.
- Understanding the customer needs and requirements and stocking product in that fashion.
- Financial management for the organization.

All of these general issues generate needs for an application, which can be formulated in different modules and can be automated for easy utility of these general practices. Leading to a better customer experience towards this business sector. Some of such practice re-organizations are:

- All paper-work are maintained as a transaction log for individual product sell, producing an organized manner of document management virtually.
- Utilizing the architecture of virtual warehousing, we can create supply and demand cycle of any product easily in real-time scenarios.
- All banking bills and documents can be easily managed under secure soft copy dock facility under the application itself. Which will also forward documents to the relative people. Like payment receipt to the customer who paid to the application for any product.
- By creating logs of all customers, and managing them individually, we can easily understand what different categories of customers are, and what those categories require from our application. And further recommending them products using these data.
- A System Admin account can be created for financial management and maintenance. Some regular computations can be automated from the application for ease of use.

Existing System

In the existing system for sales, most of the work is accomplished manually. There are a lot of complexity in data management and lots of loss in previous logs. Due to the fact that these logs are not virtually maintained, they can easily be lost or tampered on.

The banking and billing documentations are maintained with a lot of hard-work. Keeping them arranged properly and reviewing them overall from time to time can be difficult in this system. Most documentation are kept together physically and can be easily tampered as well creating ambiguity in financial management of the system.

Existing system is actually physically present at a location, leaving it restricted only to the geographically close customers. And due to the same fact, it can handle only some definite amount of customers at a time.

The existing system can not handle personnel data of all the customers it get. Which can help in improving the customer and system relationship at a good level.

Proposed System

In the proposed system for digitized sales system. We are going to tackle all issues in the existing system by automating it. Firstly, our application database will have private accounts of all the customers, who will act as both retailers and customers in their nature. Due to private accounts, we can easily understand individual taste of different users, and recommend accordingly.

Due to virtual nature of the application, all transaction log and documentation will be handled virtually using soft copies. All transaction details can be sorted accordingly and provided to respective people. All banking bills and docs can be handled and preserved for ages to come. No loss of logs are possible in this system as the logs are saved in databases which are reliable and secure.

Unique Features of the System

As most of the above mentioned problems are regularly handled these days by many small-scale and large-scale e-commerce companies, we thought that some innovative features should be added to our approach towards this practice. Some of which are:

- Product Trading(exchange of products for other products between retailers/customers).
- A week-term auctioning/bidding over products by the retailer.

Requirement Analysis & System Specifications

Feasibility Study

Technical Feasibility Study: This project contains cloud implementation and J2EE application environment. Many open source cloud service providers are easily available for prototyping purpose, and J2EE development environment(Eclipse) is easily available. The project's technical feasibility is easily achievable.

Economical Feasibility Study: This project requires a cloud deployment using Bluemix and J2EE development environment which both are available to us. JAVA is openly available and Bluemix cloud facility will be provided by the training company. Therefore, economic feasibility is easily achievable.

Operational Feasibility Study: This project requires computational algorithmic structures and basic connectivity establishment for the prototyping purpose which is easily achievable.

S**chedule Feasibility Study:** This project requires at most of 6 months for phase 1 completion.

Facilities required for proposed work

- Most of the facilities required for the accomplishment of this project is software based. Including Eclipse IDE for J2EE development environment which can easily be gathered due to open source nature of this products
- Bluemix for cloud deployment which will be provided to us by the training company.
- And an efficient data-basing implementation like MySQL for integrating to the application and storing all the data logs for the project methodology.

System Design

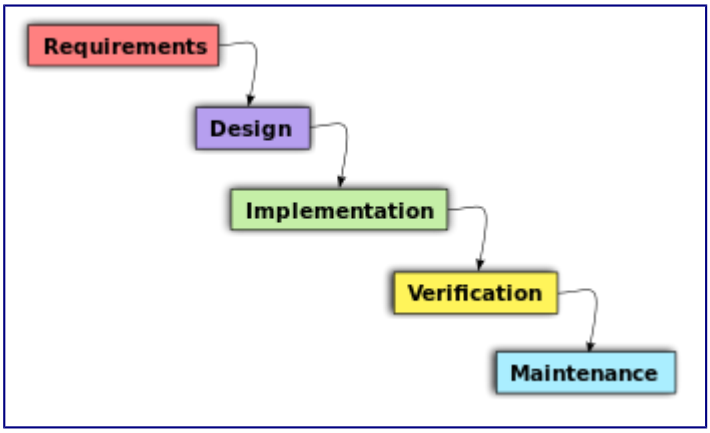
Design Approach

This project is being developed using JAVA technology, specifically J2EE technology. Which decides the design approach of this project to be Object Oriented. Which also suites the nature for our system. As we are digitizing the sales system organization, we are actually creating virtual objects of real life materials. Some of such objects are products, customers, retailers, financial managers, etc.

As most of the system structure is object based, we are going to use Object Oriented Analysis & Design Approach. Where we are going to create different objects for the system and initialize there behaviors and states under certain scenarios. JAVA is an Object Oriented programming technology using the similar OOAD architecture helping us to create this approach easily.

Object-oriented analysis and design (**OOAD**) is a popular technical approach for analyzing, designing an application, system, or business by applying the object-oriented paradigm and visual modeling throughout the development life cycles to foster better stakeholder communication and product quality.

The software life cycle is typically divided up into stages going from abstract descriptions of the problem to designs then to code and testing and finally to deployment. The earliest stages of this process are analysis and design. The analysis phase is also often called "requirements acquisition".



The Waterfall Model.

OOAD is conducted in an iterative and incremental manner, as formulated by the Unified Process.

In some approaches to software development—known collectively as waterfall models—the boundaries between each stage are meant to be fairly rigid and sequential. The term "waterfall" was coined for such methodologies to signify that progress went sequentially in one direction only, i.e., once analysis was complete then and only then was design begun and it was rare (and considered a source of error) when a design issue required a change in the analysis model or when a coding issue required a change in design.

The alternative to waterfall models are iterative models. This distinction was popularized by Barry Boehm in a very influential paper on his Spiral Model for iterative software development. With iterative models it is possible to do work in various stages of the model in parallel. So for example it is possible—and not seen as a source of error—to work on analysis, design, and even code all on the same day and to have issues from one stage impact issues from another. The emphasis on iterative models is that software development is a knowledge-intensive process and that things like analysis can't really be completely understood without understanding design issues, that coding issues can affect design, that testing can yield information about how the code or even the design should be modified, etc.

Although it is possible to do object-oriented development using a waterfall model, in practice most object-oriented systems are developed with an iterative approach. As a result, in object-oriented processes "analysis and design" are often considered at the same time.

The object-oriented paradigm emphasizes modularity and re-usability. The goal of an object-oriented approach is to satisfy the "open closed principle". A module is open if it supports extension. If the module provides standardized ways to add new behaviors or describe new states. In the object-oriented paradigm this is often accomplished by creating a new subclass of an existing class. A module is closed if it has a well defined stable interface that all other modules must use and that limits the interaction and potential errors that can be introduced into one module by changes in another. In the object-oriented paradigm this is accomplished by defining methods that invoke services on objects. Methods can be either public or private, i.e., certain behaviors that are unique to the object are not exposed to other objects. This reduces a source of many common errors in computer programming.

The software life cycle is typically divided up into stages going from abstract descriptions of the problem to designs then to code and testing and finally to deployment. The earliest stages of this process are analysis and design. The distinction between analysis and design is often described as "what vs. how". In analysis developers work with users and domain experts to define what the system is supposed to do. Implementation details are supposed to be mostly or totally (depending on the particular method) ignored at this phase. The goal of the analysis phase is to create a functional model of the system regardless of constraints such as appropriate technology. In object-oriented analysis this is typically done via use cases and abstract definitions of the most important objects. The subsequent design phase refines the analysis model and makes the needed technology and other implementation choices. In object-oriented design the emphasis is on describing the various objects, their data, behavior, and interactions. The design model should have all the details required so that programmers can implement the design in code.

Implementation

Introduction to Languages, IDE, Tools and Technologies used for Implementation

For implementing this project, many technologies were used by us, most of which are explained below, with purpose solved by these technologies, and some information regarding those technologies.

Environments:

STS (Spring Tool Suite) – STS is an eclipse based J2EE IDE used for specific spring framework usage, and as we are using spring to create RestControllers in our web Services, so it is evident that STS is a better match for web service IDE.

Download link: https://spring.io/tools/sts

Spring Tool Suite™

The Spring Tool Suite is an Eclipse-based development environment that is customized for developing Spring applications. It provides a ready-to-use environment to implement, debug, run, and deploy your Spring applications, including integrations for Pivotal tc Server, Pivotal Cloud Foundry, Git, Maven, AspectJ, and comes on top of the latest Eclipse releases.

Included with the Spring Tool Suite is the developer edition of Pivotal tc Server, the drop-in replacement for Apache Tomcat that's optimized for Spring. With its Spring Insight console, tc Server Developer Edition provides a graphical real-time view of application performance metrics that lets developers identify and diagnose problems from their desktops.

The Spring Tool suite supports application targeting to local, virtual and cloud-based servers. It is freely available for development and internal business operations use with no time limits, fully open-source and licensed under the terms of the Eclipse Public License.

Feature Highlights -

Understands your Spring App:

The Spring Tool Suite understands your Spring projects. It parses your configuration files and displays detailed information about the beans that are being defined, their dependencies among each other, used namespaces, and extracts overviews for certain stereotypes like request controllers, aspects, services, and more.

Comprehensive Validations for your Spring Configuration:

Because the Spring Tool Suite understands your Spring projects, it provides a comprehensive set of validations that are being applied automatically. Those validations indicate errors in your configurations directly within the IDE, long before you actually run the app. Finding problems and misconfigurations gets a lot easier.

Refactoring Support for your Spring App:

Refactoring support is one of the most important parts of todays software engineering. Therefore the Spring Tool Suite provides advanced support for refactoring Spring applications. Not only the well-known Java refactorings are reflected in your Spring config files, the IDE adds new refactorings for Spring elements (like renaming of Spring beans, for example).

Code Assists All Over the Place:

It doesn't matter whether you are writing Spring XML configuration files or implement JavaConfig Spring apps, whether you are using the core Spring framework alone or together with all the various additional Spring projects, the Spring Tool Suite provides you with meaningful content-assist all over the place, together with quick-fixes for common errors and problems. You will never program with Spring without those code-assists anymore.

Graphical Viewers and Editors:

Want to get an overview of the bean dependencies in your Spring app? Or wanna visualize and edit Spring Integration, Spring Batch, or Spring Webflow definitions? Check out the graphical editors that come with the Spring Tool Suite, right in your IDE, just one click away from your configuration files.

The Best AOP Support Available

The Spring Tool Suite integrates with the AspectJ language tooling for Eclipse and provides the most comprehensive support for AOP that is available today. Aspects are being recognized, incrementally woven into your system, and visualized directly within the IDE. And see where pointcuts match immediately after saving a file.

IBM Bluemix -IBM's Open Cloud Architecture implementation based on the Cloud Foundry project.

Bluemix™ is the latest cloud offering from IBM®. It enables organizations and developers to quickly and easily create, deploy, and manage applications on the cloud. Bluemix is an implementation of IBM's Open Cloud Architecture based on Cloud Foundry, an open source Platform as a Service (PaaS). Bluemix delivers enterprise-level services that can easily integrate with your cloud applications without you needing to know how to install or configure them. This article gives a high-level description of Cloud Foundry and Bluemix and outlines the features and services that were part of the open beta of Bluemix, which make it a compelling PaaS in the market today.

What is Cloud Foundry?

Cloud Foundry is an open source **platform as a service** (PaaS) that lets you quickly create and deploy applications on the cloud. Because of its open source roots, Cloud Foundry is not vendor specific and does not lock you into proprietary software or cloud infrastructure. Cloud Foundry abstracts the underlying infrastructure needed to run a cloud, letting you focus on the business of building cloud applications. The beauty of Cloud Foundry is that it provides choice. Developers and organizations can choose:

- **Development Frameworks**: Cloud Foundry supports Java™ code, Spring, Ruby, Node.js, and custom frameworks.
- **Application Services**: Cloud Foundry offers support for MySQL, MongoDB, PostgreSQL, Redis, RabbitMQ, and custom services.
- **Clouds**: Developers and organizations can choose to run Cloud Foundry in Public, Private, VMWare and OpenStack-based clouds.

Cloud Foundry's ability to provide choice comes through build-packs, a convenient way to package frameworks and run-times. Build-packs can be community based, custom built, or built from scratch. In other words, if you cannot find a framework or service build-pack that suits your needs, you could modify an existing build-pack or create your own. By using build-packs, companies are able to provide enterprise-level services like the Bluemix cloud offering.

Bluemix is an implementation of IBM's Open Cloud Architecture, based on Cloud Foundry, that enables you to rapidly create, deploy, and manage your cloud applications. Because Bluemix is based on Cloud Foundry, you can tap into a growing ecosystem of runtime frameworks and services. In addition to providing additional frameworks and services, Bluemix provides a dashboard for you to create, view, and manage your applications and services as well as monitor your application's resource usage. The Bluemix dashboard also provides the ability to manage organizations, spaces, and user access.

Bluemix provides access to a wide variety of services that can be incorporated into an application. Some of these services are delivered through Cloud Foundry. Others are delivered from IBM and third party vendors. New and enhanced services are added to the catalog often. To see the current list of runtimes and services, and their status go to the Bluemix catalog.

Some of the commonly used runtimes are:

- Node.js
- PHP
- Python
- Ruby
- J2EE

Some of the Bluemix services available from the expanding catalog include:

Service name	Description
BigInsights for Hadoop	Powered by InfoSphere BigInsights, which is based on open source Hadoop, this service provides the open source capabilities of HBase, Hive, MapReduce, Pig and others, including your own open source packages.
Business Rules	Enables developers to spend less time recoding and testing when the business policy changes. This service minimizes your code changes by keeping business logic separate from application logic.
Cloudant NoSQL DB	Provides access to a fully managed NoSQL JSON data layer that's always on. This service is compatible with CouchDB, and accessible through a simple to use HTTP interface for mobile and web application models.
Data Cache	Improve the performance and user experience of web applications by retrieving information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.
DevOps Auto-Scaling	Enables you to automatically increase or decrease the compute capacity of your application. The number of application instances are adjusted dynamically based on the Auto-Scaling policy you define.
DevOps Delivery Pipeline	Automate builds and deployments, test execution, configure build scripts, and automate execution of unit tests. Automatically build and deploy your application to IBM's cloud platform, Bluemix.
Embeddable Reporting	Use a simple cloud editor then embed reports and dashboards in your web or mobile app using a wide variety of languages such as Node.js or Java using a RESTful API.
Geospatial Analytics	Leverage real-time geospatial analytics to track when devices enter or leave defined regions.
Internet of Things	Lets your apps communicate with and consume data collected by your connected devices, sensors, and gateways.
Mobile Push Notifications	Push information to all application users or to a specific set of users and devices. You can even let users subscribe to specific tags or topics for notification.
MongoDB	A popular NoSQL database
MQ Light	Develop responsive, scalable applications with a fully-managed messaging provider in the cloud. Quickly integrate with application frameworks through easy-to-use

Service name	Description
	APIs.
Redis	A popular distributed dictionary server used by many distributed applications
Secure Gateway	Brings Hybrid Integration capability to your Bluemix environment. It provides secure connectivity from Bluemix to other applications and data sources running on-premise or in other clouds. A remote client is provided to enable secure connectivity.
Sendgrid	Sendgrid's cloud-based email infrastructure relieves businesses of the cost and complexity of maintaining email systems.
Session Cache	Improve application resiliency by storing session state information across many HTTP requests. Enable persistent HTTP sessions for your application and seamless session recovery in event of an application failure.
Single Sign-On	Implement user authentication for your web and mobile apps quickly, using simple policy-based configurations.
SQL Database	Add an on-demand relational database to your application. Powered by DB2, it provides a managed database service to handle web and transactional workloads.
Watson Alchemy API	Leverage natural language processing and computer vision in your apps to deeply understand the world's conversations, documents and photos.
Watson Language Translation	Converts text input in one language into a destination language for the end user. Translation is available among English, Brazilian Portuguese, Spanish, French, and Arabic.
Watson Personality Insights	Derives insights from transactional and social media data to identify psychological traits which determine purchase decisions, intent and behavioral traits; utilized to improve conversion rates.

For organizations, Bluemix provides a cloud platform that requires very little in-house technical know-how as well as cost savings. Bluemix provides the rapid development environment organizations need to react to users' demands for new features. The Bluemix platform and the cloud provide the elasticity and capacity flexibility organizations require when their applications explode in popularity.

GitHub – We used github to host our source code in open source society, which helps us in maintaining the project efficiently and also handles version control and back logs without any help. Openshift saves its cloud application source code in github hosting only which makes it a certain necessity.

Link: https://github.com/

GitHub is a web-based Git repository hosting service. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

GitHub offers both plans for private repositories, and free accounts which are commonly used to host open-source software projects. As of April 2016, GitHub reports having more than 14 million users and more than 35 million repositories, making it the largest host of source code in the world.

GitHub is mostly used for code.

In addition to source code, GitHub supports the following formats and features:

- Documentation, including automatically-rendered README files in a variety of Markdown-like file formats (see README files on GitHub)
- Issue tracking (including feature requests) with labels, milestones, assignees and a search engine.
- Wikis
- Pull requests with code review and comments.
- Commits history.
- Graphs: pulse, contributors, commits, code frequency, punch card, network, members.
- Integrations Directory
- Unified and split diffs.
- Email notifications.

Languages/Libraries:

JAVA 1.8 – JAVA is commonly used in this project for both web service and android application programming. Which makes it an essential part of this project.

Java is a general-purpose 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 compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licences. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

The latest version is Java 8, which is the only version currently supported for free by Oracle, although earlier versions are supported both by Oracle and other companies on a commercial basis.

James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time. The language was initially called *Oak* after an oak tree that stood outside Gosling's office. Later the project went by the name *Green* and was finally renamed *Java*, from Java coffee. Gosling designed Java with a C/C++-style syntax that system and application programmers would find familiar.

Sun Microsystems released the first public implementation as Java 1.0 in 1995. It promised "Write Once, Run Anywhere" (WORA), providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run *Java applets* within web pages, and Java quickly became popular, while mostly outside of browsers, that wasn't the original plan. In January 2016, Oracle announced that Java runtime environments based on JDK 9 will discontinue the browser plugin. The Java 1.0 compiler was re-written in Java by Arthur van Hoff to comply strictly with the Java 1.0 language specification. With the advent of *Java 2* (released initially as J2SE 1.2 in December 1998 – 1999), new versions had multiple configurations built for different types of platforms.

J2EE included technologies and APIs for enterprise applications typically run in server environments, while *J2ME* featured APIs optimized for mobile applications. The desktop version was renamed *J2SE*. In 2006, for marketing purposes, Sun renamed new *J2* versions as *Java EE*, *Java ME*, and *Java SE*, respectively.

In 1997, Sun Microsystems approached the ISO/IEC JTC 1 standards body and later the Ecma International to formalize Java, but it soon withdrew from the process. Java remains a *de facto* standard, controlled through the Java Community Process. At one time, Sun made most of its Java implementations available without charge, despite their proprietary software status. Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System.

On November 13, 2006, Sun released much of its Java virtual machine (JVM) as free and open-source software, (FOSS), under the terms of the GNU General Public License (GPL). On May 8, 2007, Sun finished the process, making all of its JVM's core code available under free software/open-source distribution terms, aside from a small portion of code to which Sun did not hold the copyright.

Sun's vice-president Rich Green said that Sun's ideal role with regard to Java was as an "evangelist". Following Oracle Corporation's acquisition of Sun Microsystems in 2009–10, Oracle has described itself as the "steward of Java technology with a relentless commitment to fostering a community of participation and transparency". This did not prevent Oracle from filing a lawsuit against Google shortly after that for using Java inside the Android SDK (see Google section below). Java software runs on everything from laptops to data centers, game consoles to scientific supercomputers. On April 2, 2010, James Gosling resigned from Oracle.

Spring Framework – It is used in this application for developing web services using RESTcontrollers which is easily applicable by the spring framework libraries. It is widely used now-a-days for similar purpose and also can be used to host the application all by itself on a spring boot server, but we needed to host it on openshift as a cloud application so we didn't opted that feature.

The Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java applications. Spring handles the infrastructure so you can focus on your application.

Spring enables you to build applications from "plain old Java objects" (POJOs) and to apply enterprise services non-invasively to POJOs. This capability applies to the Java SE programming model and to full and partial Java EE.

Examples of how you, as an application developer, can benefit from the Spring platform:

- Make a Java method execute in a database transaction without having to deal with transaction APIs.
- Make a local Java method a remote procedure without having to deal with remote APIs.
- Make a local Java method a management operation without having to deal with JMX APIs.
- Make a local Java method a message handler without having to deal with JMS APIs.

Hibernate ORM – Hibernate is a java library used to handle java persistence interactions between service program and database. It provides a neat implementation of simultaneous transaction handling and data persistence control without any actual SQL command to be written by the developer, increasing the productivity of the team.

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.

Pivotal Virtual local-host server – as a web application is deployed on a server, rather than using a real server that could be very expensive, I used a virtual local-host generated server for deploying the application.

Pivotal Software, Inc. (Pivotal) is a software company based in Palo Alto, California that provides software and services for the development of custom applications for data and analytics based on cloud computing technology. Pivotal Software is a spin-out and joint venture of EMC Corporation and its subsidiary VMware that combined software products, employees, and lines of businesses from the two parent companies including Greenplum, Cloud Foundry, Spring, Pivotal Labs, GemFire and other products from the VMware vFabric Suite.

In July 2012 a GigaOM blog entry speculated on a possible spin out of VMware and EMC that would consolidate some of their cloud computing projects into a new division. The companies confirmed speculation in December of the same year, announcing the initiative with existing technology, people and programs from both companies focused on big data and cloud application platforms under one organization. On April 1, 2013, *The New York Times* reported that Pivotal was official and positioned as a competitor to Amazon Web Services.

At an official event held on April 24, 2013, the organization announced both a $105 million investment from General Electric and its PaaS offering, PivotalCF, a cloud-enabled application platform for private cloud initiatives and public cloud providers.

Oracle Database - (commonly referred to as **Oracle RDBMS** or simply as **Oracle**) It is abbrevated as **Oak Ridge Automatic Computer and Logical Engine** and It is an object-relational database management system produced and marketed by Oracle Corporation.

Larry Ellison and his two friends and former co-workers, Bob Miner and Ed Oates, started a consultancy called Software Development Laboratories (SDL) in 1977. SDL developed the original version of the Oracle software. The name *Oracle* comes from the code-name of a CIA-funded project Ellison had worked on while previously employed by Ampex.

An Oracle database system—identified by an alphanumeric system identifier or SID—comprises at least one instance of the application, along with data storage. An instance—identified persistently by an instantiation number (or activation id: SYS.V_$DATABASE.ACTIVATION#)—comprises a set of operating-system processes and memory-structures that interact with the storage. (Typical processes include PMON (the process monitor) and SMON (the system monitor).) Oracle documentation can refer to an active database instance as a "shared memory realm".

Users of Oracle databases refer to the server-side memory-structure as the SGA (System Global Area). The SGA typically holds cache information such as data-buffers, SQL commands, and user information. In addition to storage, the database consists of online redo logs (or logs), which hold transactional history. Processes can in turn archive the online redo logs into archive logs (offline redo logs), which provide the basis (if necessary) for data recovery and for the physical-standby forms of data replication using Oracle Data Guard.

If the Oracle database administrator has implemented Oracle RAC (Real Application Clusters), then multiple instances, usually on different servers, attach to a central storage array. This scenario offers advantages such as better performance, scalability and redundancy. However, support becomes more complex, and many sites do not use RAC. In version 10*g*, grid computing introduced shared resources where an instance can use (for example) CPU resources from another node (computer) in the grid.

The Oracle DBMS can store and execute stored procedures and functions within itself. PL/SQL (Oracle Corporation's proprietary procedural extension to SQL), or the object-oriented language Java can invoke such code objects and/or provide the programming structures for writing them.

The Oracle RDBMS stores data logically in the form of tablespaces and physically in the form of data files ("datafiles"). Tablespaces can contain various types of memory segments, such as Data Segments, Index Segments, etc. Segments in turn comprise one or more extents. Extents comprise groups of contiguous data blocks. Data blocks form the basic units of data storage.

Maven repository -maven is used for automatically generating a standard project directory structure for a vast amount of varying applications and automate the process of adding dependencies for those projects. As we provide the details for any dependency that we want to be added to the project, maven finds those dependencies online and import them to our projects.

Maven is a build automation tool used primarily for Java projects. The word *maven* means 'accumulator of knowledge' in Yiddish. 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.
Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

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. Currently a plugin for the .NET framework exists and is maintained, and a C/C++ native plugin is maintained for Maven 2.

Alternative technologies like gradle and sbt as build tools do not rely on XML, but keep the key concepts Maven introduced. With Apache Ivy, a dedicated dependency manager was developed as well that also supports Maven repositories.

A central feature in Maven is dependency management. Maven's dependency-handling mechanism is organized around a coordinate system identifying individual artifacts such as software libraries or modules. The POM example above references the JUnit coordinates as a direct dependency of the project. A project that needs, say, the Hibernate library simply has to declare Hibernate's project coordinates in its POM. Maven will automatically download the dependency and the dependencies that Hibernate itself needs (called transitive dependencies) and store them in the user's local repository. Maven 2 Central Repository is used by default to search for libraries, but one can configure the repositories to be used (e.g., company-private repositories) within the POM.

There are search engines such as The Central Repository Search Engine which can be used to find out coordinates for different open-source libraries and frameworks.

Projects developed on a single machine can depend on each other through the local repository. The local repository is a simple folder structure that acts both as a cache for downloaded dependencies and as a centralized storage place for locally built artifacts. The Maven command mvn install builds a project and places its binaries in the local repository. Then other projects can utilize this project by specifying its coordinates in their POMs.

Bibliography

- Department Faculties for Advise and mentor, CSE Dept., SVIET.
- Company trainer & guide, KVCH-IBMCE, Preet Vihar, New Delhi.
- Definitions and explanations for major topics, Wikipedia.
- Project Hosting and maintenance, GitHub.(www.githhub.com)