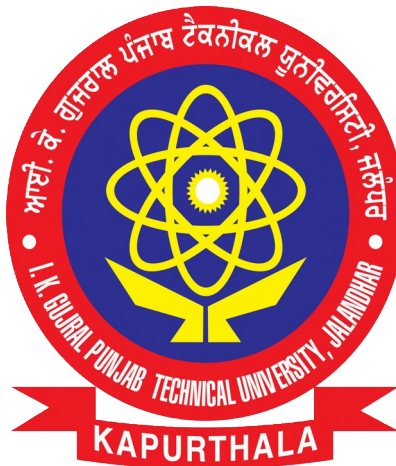


DIGITIZED SALES SYSTEM

FINAL 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.:

Mrs. Kawaljit Kaur
Head Of Department, CSE

The Computer Science Department
Swami Vivekanand Institute of Engineering & Technology
Chandigarh-Patiala highway, sector-8, Ramnagar, Banur
“2017”

Table of Content

	Page no.
<i>Acknowledgment</i>	<i>i</i>
<i>Abstract</i>	<i>ii</i>
<i>Certificate</i>	<i>iii</i>
<i>About Company</i>	<i>v</i>
<i>List of Figures</i>	<i>vi</i>
<i>List of Abbreviations</i>	<i>vii</i>
CHAPTER 1: INTRODUCTION	1
1.1 Introduction to Project	1
1.2 Project Category	2
1.3 Objectives	2
1.4 Problem Formulation	2
1.5 Identification/Reorganization of Need	3
1.6 Existing System	4
1.7 Proposed System	4
1.8 Unique Features of the System	4
CHAPTER 2: Requirement Analysis and System Specification	5
2.1 Feasibility study	5
2.2 Software Requirement Specification Document	6
2.3 SDLC model to be used	11
Chapter 3. System Design	15
3.1 Design Approach	15
3.2 Structured Analysis	17
3.3 User Interface Design	23
3.4 Database Design	25
3.4.1 ER Diagrams	26
3.4.2 Database Manipulation	26
3.4.3 Database Connection Controls and Strings	28
3.5 Methodology	30

Chapter 4. Implementation, Testing, and Maintenance	31
4.1 Introduction to Languages, IDE's, Tools and Technologies used for Implementation	31
4.2 Coding standards of Language used	42
4.3 Testing Techniques and Test Plans	47
Chapter 5. Results and Discussions	50
5.1 Snapshots of System	50
5.2 Back Ends Representation	55
5.2.1 Snapshots of Database Documents	55
Chapter 6. Conclusion and Future Scope	57
References/Bibliography	58

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

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.

Certificates



May 19, 2017

PROVISIONAL CERTIFICATE

This is to certify that **Mr. /Ms. Anurag Kumar, B.Tech** student of "**SVIET**", has successfully completed 6 Months **Industrial Training** at KVCH towards partial fulfillment of his academic requirement.

He worked on the project "**Digitized Sale System**" on **CLOUD COMPUTING** technology.

His performance was excellent during the tenure.

We wish him a bright and successful career.



Manager
Pooja Arora

KVCH IBM-CE & ORACLE WD
C-109 Sec:2,Noida
9266625607
corporate@kvch.in
www.kvch.in

www.kvch.in

NOIDA

📍 C-109, Sector-2, Noida- 201301
☎ 9266625607 📧 training@kvch.in

DELHI

📍 G-18, 2nd Floor, Preet Vihar- 110092
☎ 9212260630 📧 delhi.trg1@kvch.in

K V Computer Home Pvt Ltd

CIN : U74899DL1991PTC044952
Helpline Number : 9510860860

May 19, 2017

PROVISIONAL CERTIFICATE

This is to certify that **Mr. /Ms. Pankaj, B.Tech** student of "**Swami Vivekanand Institute of Engineering & Technology**", has successfully completed 6 Months **Industrial Training** at KVCH towards partial fulfillment of his academic requirement.

He worked on the project "**DIGITIZED SALES SYSTEM**" on **JAVA** technology.

His performance was excellent during the tenure.

We wish him a bright and successful career.



Manager
Pooja Arora

KVCH IBM-CE & ORACLE WD
C-109 Sec:2,Noida
9266625607
corporate@kvch.in
www.kvch.in

www.kvch.in

NOIDA

📍 C-109, Sector-2, Noida- 201301
☎ 9266625607 ✉ training@kvch.in

DELHI

📍 G-18, 2nd Floor, Preet Vihar- 110092
☎ 9212260630 ✉ delhi.trg1@kvch.in

K V Computer Home Pvt Ltd

CIN : U74899DL1991PTC044952
Helpline Number : 9510860860

About Company



KVCH is a 26 years old company having expertise on multiple training domains. The company established in 1991 & joins hands with various major giants like CMC LTD, TCS, Tech Mahindra, CRIS, HCL & many other govt bodies.

The journey took over a very long period once the company got privatization in 1991. The Company got the authentic unique identification no. U74899DL1991PTC044952 from Ministry of Company affairs, India in 1991.

KVCH is associated with IBM & Oracle as Business Partner.

In a world, where there is growing demand for skilled IT professionals, the key to transforming today's students into tomorrow's working professionals is to develop industry capabilities right from foundation level.

The KVCH training modules provides an opportunity in experiential learning to work on projects based on real-world problems, rather than industry simulations. Make the right beginning with a project that requires practical knowledge to back-up the theories you have mastered.

Key Features of KVCH

- Choose from a variety of industry aligned projects based on real world challenges solved by IBM using cutting edge technologies.
- Blend IT skills, knowledge and industry best practices.
- Learn and work with enterprise class tools, and methodologies.
- Get expert coaching and mentoring.
- Get yourself evaluated and certified by IBM.

List of Figures

S.No.	Figure description	P.No.
1	System Environment	6
2	Use Case: User Authentication	6
3	Use Case: Buy Product	7
4	Use Case: Order Checkout	7
5	Use Case: Review Product	7
6	Use Case: Add Product	8
7	Use Case: Remove Product	8
8	Use Case: Edit Product	8
9	Use Case: Search Product	9
10	Use Case: Edit Profile	9
11	Use Case: Create/Delete Profile	9
12	Logical Structure of Application	10
13	Agile Technology	11
14	The Waterfall Model	15
15	DFD Context Flow Diag	17
16	DFD level 0	18
17	Structured Flow Chart	22
18	Bsic User Interface Interlink Design	23
19	ER Diagram	26
20	Login Page	50
21	Home Page	51
22	Product Desc Page	52
23	Order checkout page	53
24	Payment gateway page	54
25	Product document JSON format	55
26	Order document JSON format	56
27	User document JSON format	56

List of Acronyms

EA	: Enterprise Architecture
FDD	: Function Driven Development
SDLC	: Software Development Life Cycle
DSS	: Digitized Sales System
SQL	: Structured Query Language
IEEE	: Institute of Electrical and Electronics Engineering
IT	: Information Technology
RMP	: Risk Management Planning
SaaS	: Software as a Service
SRS	: Software Requirement Specification
SDD	: Software Design Document

CHAPTER 1: INTRODUCTION

1.1 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 businesses.

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.

1.2 Project Category

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

1.3 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.

1.4 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 need 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.

1.5 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.

1.6 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.

1.7 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.

1.8 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.

CHAPTER2: Requirement Analysis & System Specifications

2.1 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.

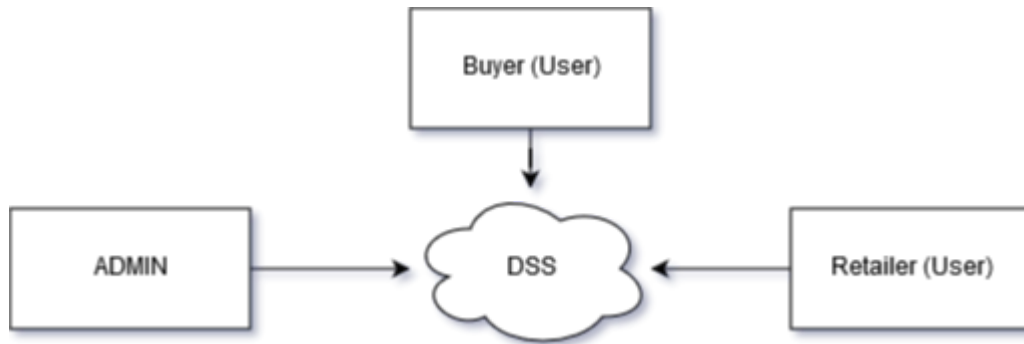
Schedule Feasibility Study: This project requires at most of 6 months for phase 1 completion.

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.

2.2 Software Requirement Specifications Document

System Environment



Our Project has only 3 kinds of Actors. Which are Buyer, Retailer and Administrator. As these name suggest, they do there relevant jobs. Admin has lowest Abstraction over the system and can manipulate it according to his will. Buyers are here to buy products of the website and retailers are here to sell their products using this website. At some instances, Buyers and retailers can be the same User. Hence, both are evolved to create a hybrid User that can sell and buy simultaneously using this application website.

Functional Requirement Specification

Use Case:

Diagram:

Brief Description:

Use Case: User Authentication

Diagram:

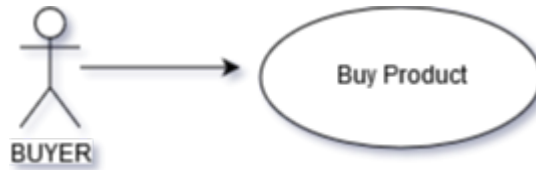


Brief Description:

Users will provide there email ID and password for authentication of themselves. Post-condition will include leading to home page of the application with user registered as online. If the combination of ID and password doesn't make, the authentication attempt will fail and user will not be registered as online.

Use Case: Buy Product

Diagram:



Brief Description:

Buyer (User) will specify a product that he wants to buy with its quantity which will be added to the user's currently active order. Which is commonly referred as cart.

Use Case: Order Checkout

Diagram:



Brief Description:

Buyer's (User) currently active order will be checked out for availability of supplies and total cost. After which Buyer will be asked to enter shipping information and finally payment service. After payment process, invoice or the content bought will be provided to the Buyer by mail and Retailer will be informed about his sell.

Use Case: Review Product

Diagram:



Brief Description:

Buyer who have ordered a product can review it in few days to tell other buyers about the quality of product and service of this company.

Use Case: Add Product

Diagram:



Brief Description:

Retailer/Admin can add a new product to the collection of the application. That product will be put in a category and Buyers will be recommended new products for better experience. Product will also require inventory amount available at the Retailer's.

Use Case: Remove Product

Diagram:



Brief Description:

Retailer/Admin can remove any present product from the collection in the application at any time. A retailer can only remove their own products but admin can remove any product from himself or any other retailer. All details regarding that product will be lost and it will be removed from any buyers cart available, if added.

Use Case: Edit Product

Diagram:



Brief Description:

Retailer/Admin can edit any present product from the collection in the application at any time. A retailer can only edit their own products but admin can edit any product from himself or any other retailer. All details regarding that product will be edited and it will be updated from any buyers cart available, if added.

Use case: Search Product

Diagram:



Brief Description:

Any user entity can search for a product available in the applications collection. Search can be done on the bases of similarly named products, similarly named categories, solid category search, under price range search, etc.

Use case: Edit Profile

Diagram:



Brief Description:

Retailer/Buyer can edit their respective profile information at any time using this functionality. They will enter their updated information and it will be modified in the database regarding their own information set. Any user can't handle any other user's information.

Use case: create/delete profile

Diagram:



Brief Description:

Any user can create their profile or delete their profile. To place any order or sell any product, you would require to add a profile and when user delete their respective profile, their data will get lost. That is, if a retailer had any product still available, that product will get removed, and all logs of orders from a buyer will be lost.

User Characteristics

A Buyer is supposed to traverse over products in the application collection, and add products they want to buy in the current order. And at the end, finish the order by checking out.

A Retailer is supposed to input different product available with him, and also the inventory amount of those products. They are supposed to stay alert about sell of there products and invoice regarding those sells.

Non-Functional Requirement

This application is web based, so we require a good internet connection between the application servers and the end-user. The speed of the End-User's connection will depend on the hardware used rather than characteristics of this system. We also need concurrency between product and their inventory, so that no false orders can be put into the system.

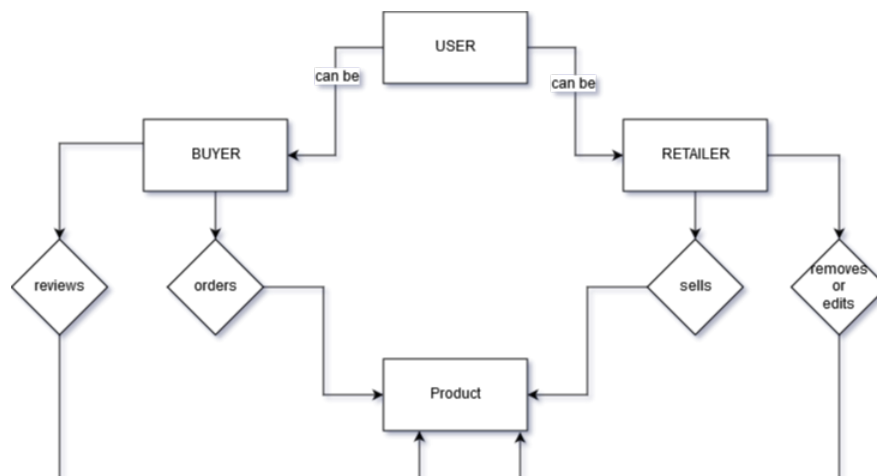
Security Requirement

Security will be provided at the server side of the application. For better handling of data leaks, cloud entity containers will handle firewall and security themselves. User authentication is the first step in this application, leading to data integrity and authentication.

Look and Feel Requirement

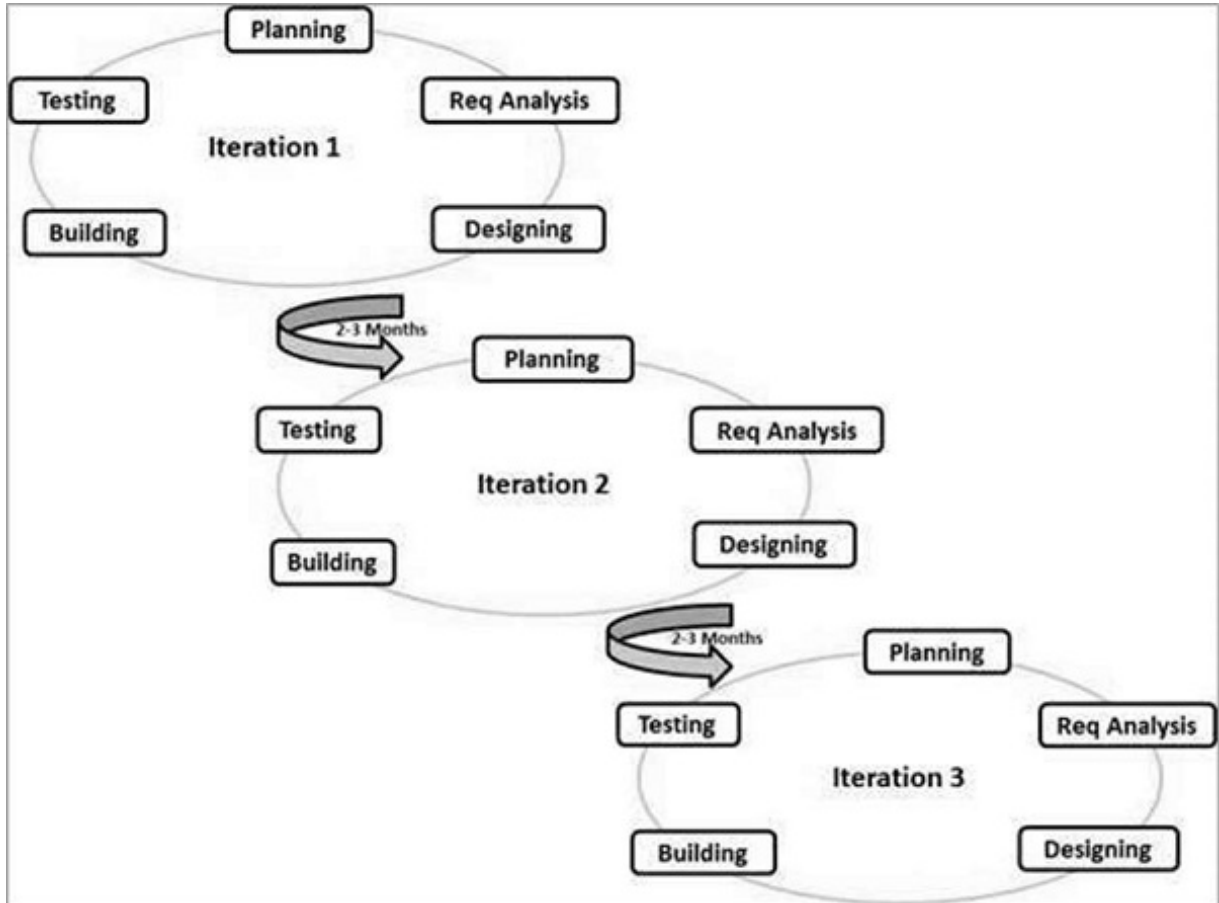
Look and Feel of the application needs to be responsive in nature. This application should have smooth transitions between PCs and Mobile devices. This will be achieved using efficient CSS coding. Bootstrap is one of the standard responsive CSS library built by Twitter inc.

Logical Structure of Application



2.3 SDLC model to be Used

SDLC model that is being used to complete this project is the standard **Agile Methodology**.



Agile methodologies arose from the need to develop software applications that could accommodate the fast-paced evolution of the Internet. Agile is, in some way, a variant of iterative life cycle where deliverables are submitted in stages. The main difference is that agile cuts delivery time from months to weeks. Companies practicing agile are delivering software products and enhancements in weeks rather than in months. Moreover, the agile manifesto covered development concepts aside from the delivery life cycle, such as collaboration, documentation, and others.

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks.

Every iteration involves cross functional teams working simultaneously on various areas like –

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

The Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

The most popular Agile methods include Rational Unified Process (1994), Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995). These are now collectively referred to as Agile Methodologies, after the Agile Manifesto was published in 2001.

Following are the Agile Manifesto principles –

- Individuals and interactions – In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- Working software – Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
- Customer collaboration – As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
- Responding to change – Agile Development is focused on quick responses to change and continuous development.

Agile methods are being widely accepted in the software world recently. However, this method may not always be suitable for all products. Here are some pros and cons of the Agile model.

The advantages of the Agile Model are as follows –

- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required.
- Easy to manage.
- Gives flexibility to developers.

The disadvantages of the Agile Model are as follows –

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- There is a very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.

Agile is based on the adaptive software development methods, whereas the traditional SDLC models like the waterfall model is based on a predictive approach. Predictive teams in the traditional SDLC models usually work with detailed planning and have a complete forecast of the exact tasks and features to be delivered in the next few months or during the product life cycle.

Predictive methods entirely depend on the requirement analysis and planning done in the beginning of cycle. Any changes to be incorporated go through a strict change control management and prioritization.

Agile uses an adaptive approach where there is no detailed planning and there is clarity on future tasks only in respect of what features need to be developed. There is feature driven development and the team adapts to the changing product requirements dynamically. The product is tested very frequently, through the release iterations, minimizing the risk of any major failures in future.

Customer Interaction is the backbone of this Agile methodology, and open communication with minimum documentation are the typical features of Agile development environment. The agile teams work in close collaboration with each other and are most often located in the same geographical location.

CHAPTER 3: System Design

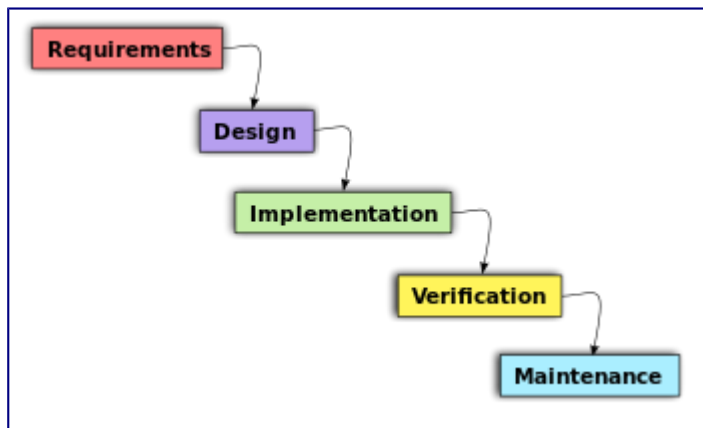
3.1 Design Approach

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.

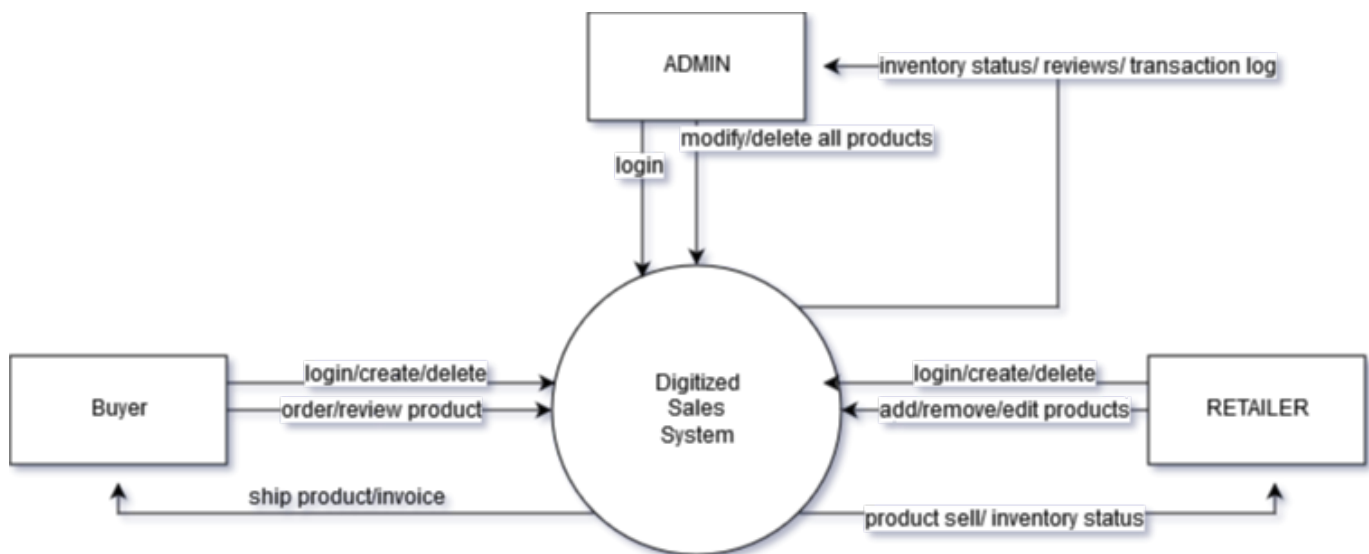
Although it is possible to do object-oriented development using a waterfall model, in practice most object-oriented systems are developed with an iterative approach. As a result, in object-oriented processes "analysis and design" are often considered at the same time.

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.

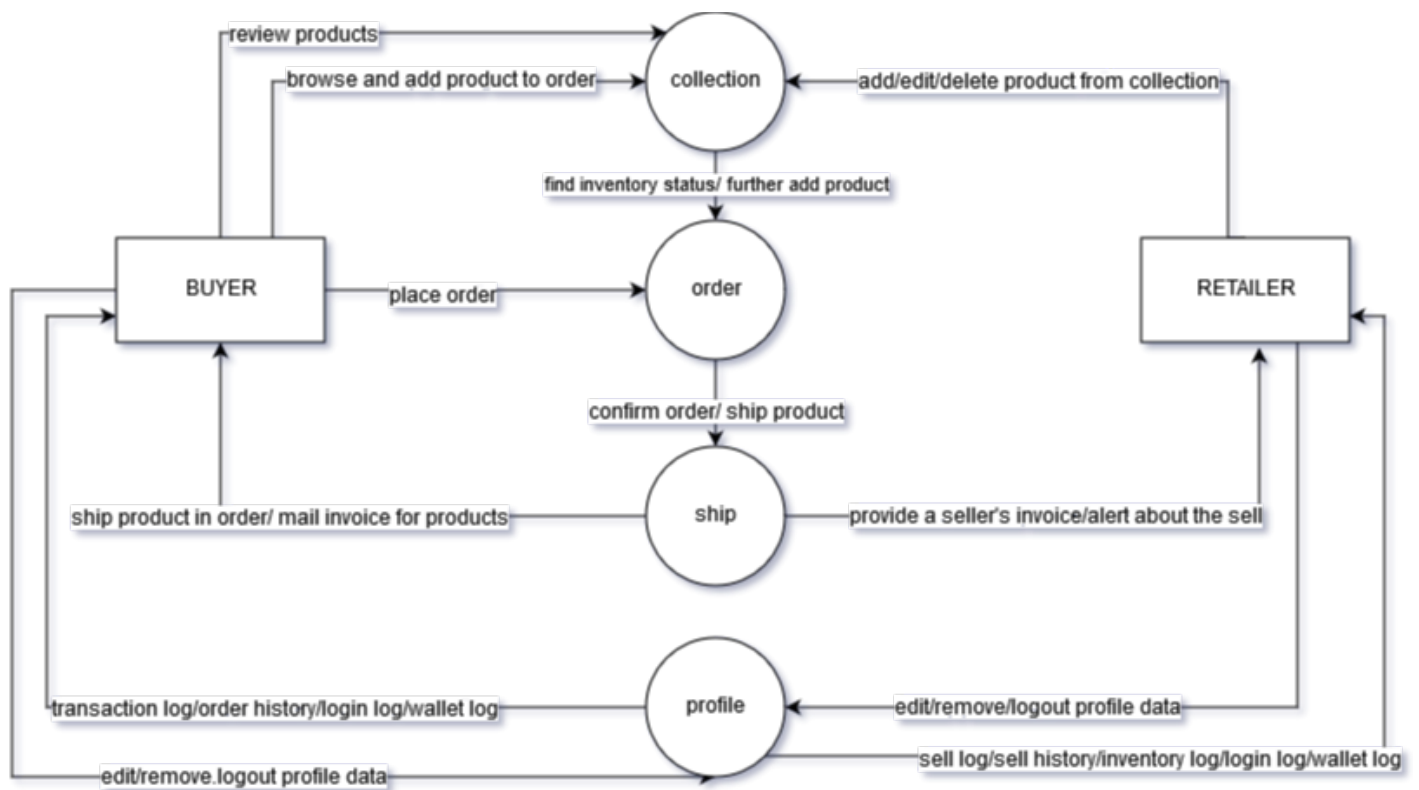
3.2 Structured Analysis

DFD Context Flow Diagram



Data Flow Diagram of DSS

DFD level 0 Diagram



DFD level 0

Data Dictionary
Entity – User

Attribute Name	Required	Type	Default Value
userId	YES	VARCHAR	N/a
UserPass	YES	VARCHAR	N/a
Email	YES	VARCHAR	N/a
IsAdmin	YES	BOOLEAN	False
FName	YES	VARCHAR	N/a
LName	NO	VARCHAR	N/a
Address1	YES	VARCHAR	N/a
Address2	NO	VARCHAR	N/a
City	YES	VARCHAR	N/a
State	YES	VARCHAR	N/a
PostalCode	YES	VARCHAR	N/a
Province	YES	VARCHAR	N/a
Country	YES	VARCHAR	INDIA
Phone	YES	VARCHAR	N/a
Lang	YES	VARCHAR	Eng
FavCategory	NO	VARCHAR	N/a
ShowPetTips	YES	VARCHAR	true

Entity – Product

Attribute Name	Required	Type	Default Value
Name	YES	VARCHAR	N/a
ProductId	YES	VARCHAR	N/a
Description	NO	VARCHAR	N/a
Category	YES	Category(Entity)	N/a
SearchKey	YES	VARCHAR ARR	N/a
ImgURL	NO	VARCHAR	N/a
UnitCost	YES	NUMBER	0.0
Quantity	YES	NUMBER	0
Avalabilty	YES	BOOLEAN	False
ReorderQuantity	NO	NUMBER	0
Supplier	YES	Supplier(Entity)	N/a

Entity – Order

Attribute Name	Required	Type	Default Value
UserId	YES	VARCHAR	N/a
OrderDate	YES	DATE	System current
OrderStatus	YES	VARCHAR	“incomplete”
Cctype	NO	VARCHAR	N/a
CcNumber	NO	VARCHAR	N/a
CcExpiryMonth	NO	VARCHAR	N/a
CcExpiryYear	NO	VARCHAR	N/a
BillFName	YES	VARCHAR	N/a
BillLName	YES	VARCHAR	N/a
BillAddress1	YES	VARCHAR	N/a
BillAddress2	NO	VARCHAR	N/a
BillCity	YES	VARCHAR	N/a
BillState	YES	VARCHAR	N/a
BillProvince	YES	VARCHAR	N/a
BillPostalCode	YES	VARCHAR	N/a
BillCountry	YES	VARCHAR	INDIA
Shiptobillingaddress	YES	BOOLEAN	TRUE
ShipFName	NO	VARCHAR	N/a
ShipLName	NO	VARCHAR	N/a
ShipAddress1	NO	VARCHAR	N/a
ShipAddress2	NO	VARCHAR	N/a
ShipCity	NO	VARCHAR	N/a
ShipState	NO	VARCHAR	N/a
ShipProvince	NO	VARCHAR	N/a
shipPostalCode	NO	VARCHAR	N/a
ShipCountry	NO	VARCHAR	N/a
Subtotal	YES	DOUBLE	0.0
ShipMethod	YES	VARCHAR	“NORMAL”
ShipCost	YES	DOUBLE	0.0
Total	YES	DOUBLE	0.0
Items	YES	Item(Entity)	N/a

Entity – Supplier

Attribute Name	Required	Type	Default Value
Name	YES	VARCHAR	N/a
SupplierStatus	YES	VARCHAR	“IDLE”
Address1	YES	VARCHAR	N/a
Address2	NO	VARCHAR	N/a
City	YES	VARCHAR	N/a
State	YES	VARCHAR	N/a
PostalCode	YES	VARCHAR	N/a
Province	YES	VARCHAR	N/a
Country	YES	VARCHAR	INDIA
Phone	YES	VARCHAR	N/a

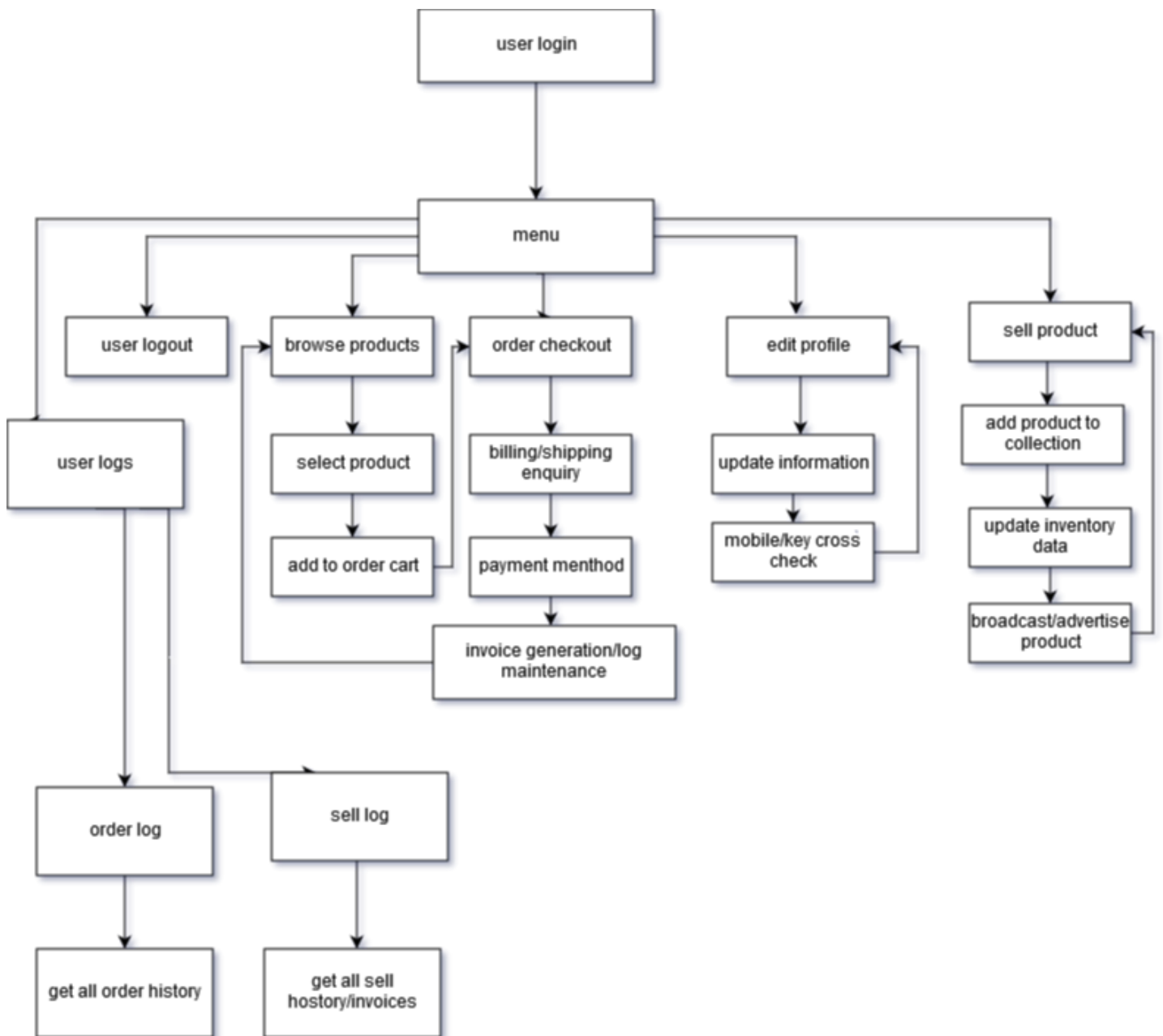
Entity – Item

Attribute Name	Required	Type	Default Value
Quantity	YES	NUMBER	0
LineTotal	YES	DOUBLE	0.0
ProductID	YES	VARCHAR	N/a

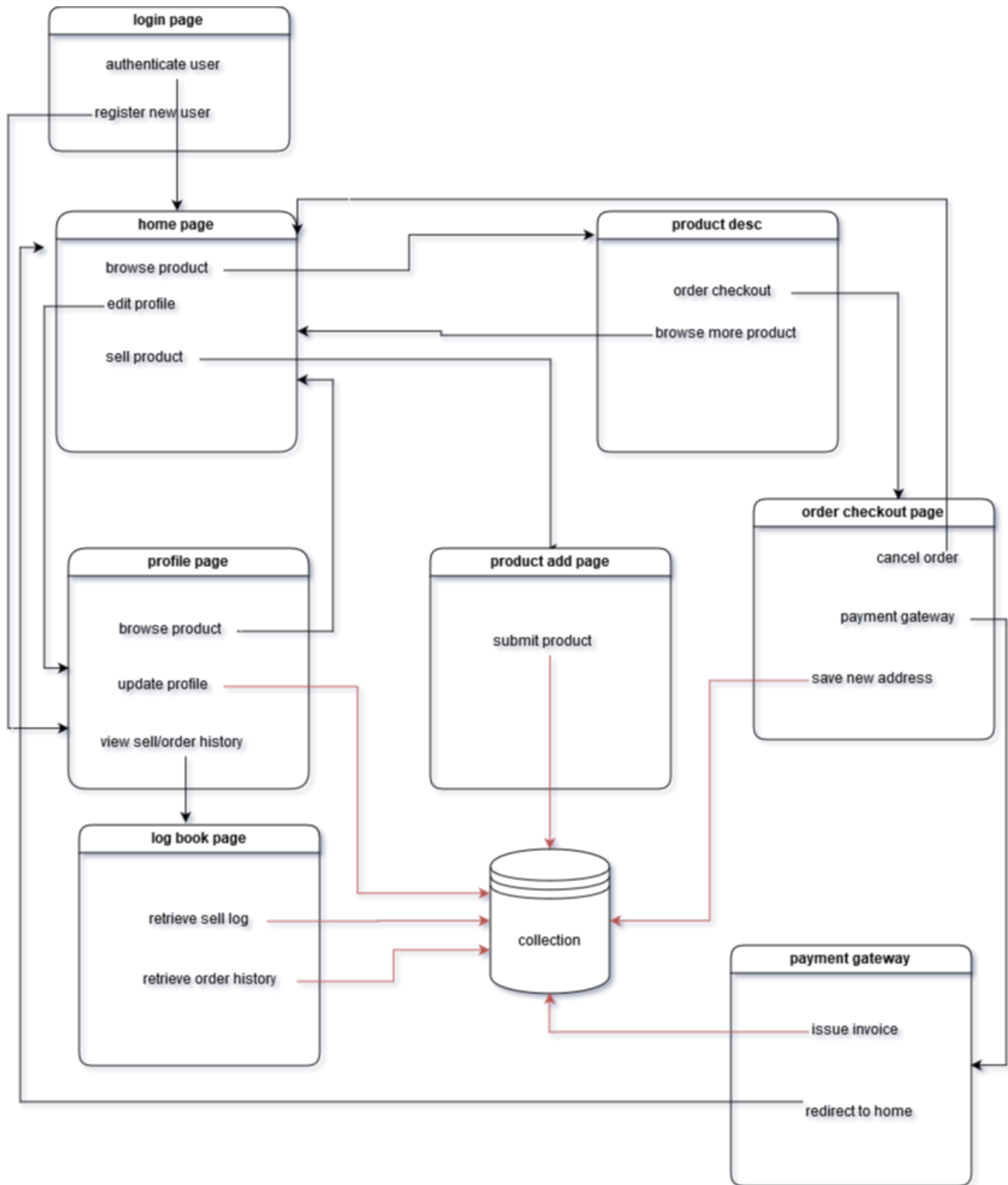
Entity – Category

Attribute Name	Required	Type	Default Value
_id	YES	VARCHAR	RANDOM
_rev	YES	VARCHAR	RANDOM
CategoryID	YES	VARCHAR	N/a
Name	YES	VARCHAR	N/a
Description	NO	VARCHAR	N/a
ImgURL	NO	VARCHAR	N/a
BannerURL	NO	VARCHAR	N/a

Structured Flow Chart



3.3 User Interface Design



Basic User Interface Interlink design

Basic approach behind the UI Design would be Material Flat Look & Feel Design. Flat design is a minimalist UI design genre, or design language, commonly used in graphical user interfaces (such as web applications and mobile apps), especially in such graphical materials as posters, arts, guide documents, publishing products.

Flat design is a style of interface design emphasizing minimum use of stylistic elements that give the illusion of three dimensions (such as the use of drop shadows, gradients or textures) and is focused on a minimalist use of simple elements, typography and flat colors. Designers may prefer flat design because it allows interface designs to be more streamlined and efficient. It is easier to quickly convey information while still looking visually appealing and approachable. Additionally, it makes it easier to design an interface that is responsive to changes in browser size across different devices. With minimal design elements, websites are able to load faster and resize easily, and still look sharp on high-definition screens. As a design approach, it is often contrasted to skeuomorphism and *rich design*. (Though it should be noted that flat design can use skeuomorphs just as much as a realistically designed UI).

Major Interfaces

Login Page, where the user will start his journey of this application. First most page of this application. New Users could register themselves or old users could reconnect to the system. Register option will lead to profile page while login process will lead to home page of the application.

Home page, where user enters after login process. Here, user can browse between products under different categories to create his order. Any product view will lead to the product descriptive page. User can filter the product to be viewed according to different fields like company name, cost bars, review popularity, latest products, etc.

Profile Page, where user can edit or delete their existing profile and its relative information. It also provides with functionalities such as order history log and sell history log. This helps user to maintain accountability for every transaction made from this system. You can also mention your preferences here, in your profile for better recommendation of products.

Order Checkout Page, where all items that user want to buy are accumulated in an order entity and final cost is calculated. User interacts with this interface in step by step manner. Firstly, billing and shipping addresses are noted from the user, or attached according to historical data of addresses. Then, final amount with added shipment cost is calculated and user is forwarded to the payment gateway page.

Payment Gateway Page, where user arrives with an order in hand, and an amount to be paid to the system itself. User selects his preferred payment method from the list of methods provided by the payment gateway. Then final payment is preceded. Invoice of that order is mailed to users mail ID and billed under provided billing address of the order.

Log Book Page, where user logs regarding sell product or order product is viewed. This page retrieves all the transitional log history of the user and presents it in front of himself. This helps the user to maintain accountability for every amount transaction with this system.

Product Add Page, when user wants to become a retailer and sell products, he uses this page. User enters information about the product he wants to sell and with the inventory amount available to him. Then that data is added to the collection of products for other users to view.

Product Desc Page, where every information about a product in the collection is presented. Details like retailer information, delivery charges, descriptive knowledge, etc are mentioned in this view page. Snapshots of the actual product is also available for much better description of product. Reviews of different users relative to this product is available here. Any user can come to this page and add the relate product to their order checkout list.

Basic technologies being used for user interface design in this project are:

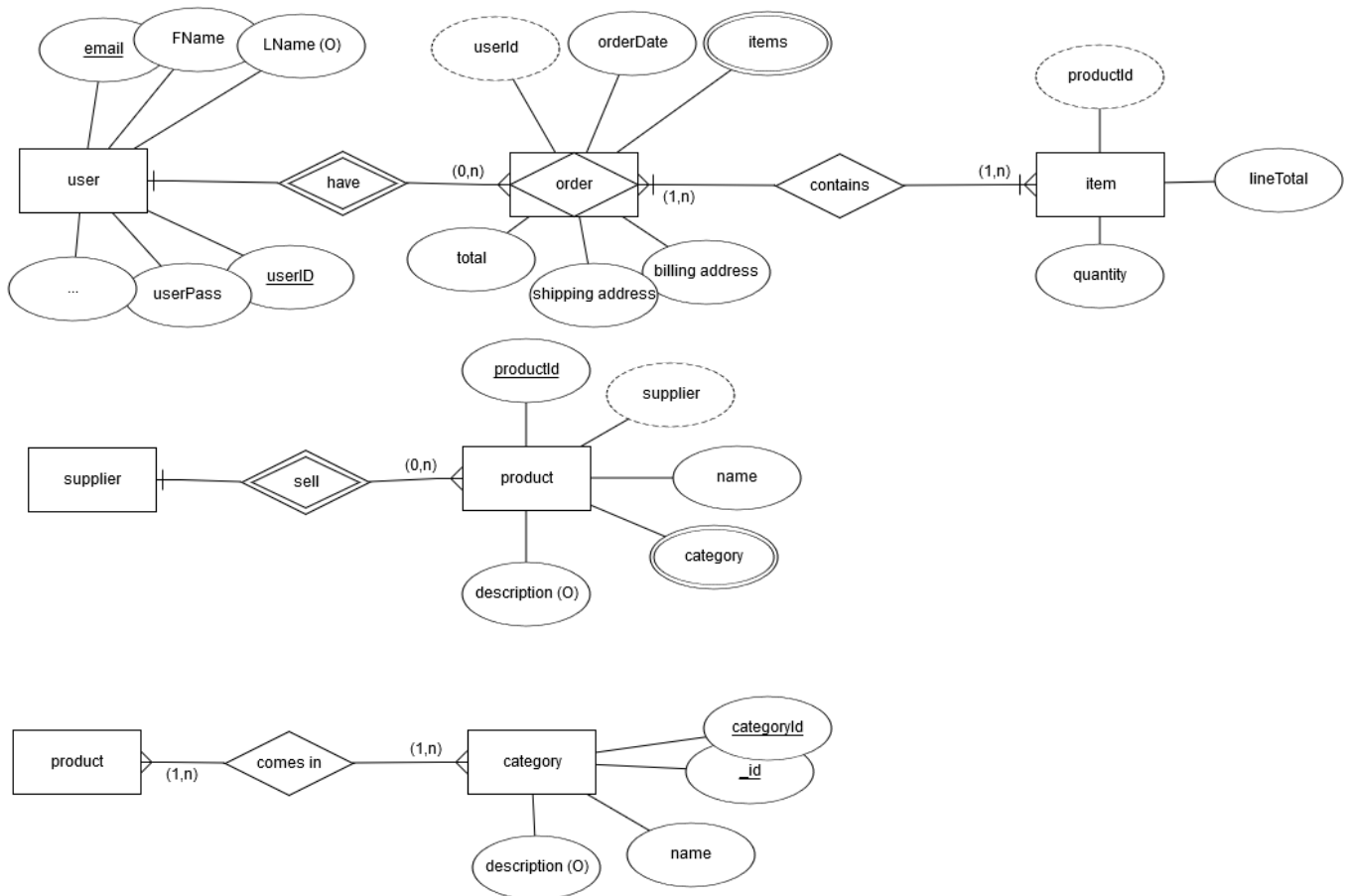
- HTML
- CSS
- JavaScript(JS)
- JQUERY
- Bootstrap by twitter inc.
- AJAX
- JSP

3.4 Database Designer

Database is the most essential part of any project application. It is the module that provided our project to power of persistence. Any application that remembers anything about is because of some sort of database.

Database Management System used in this project is one that is easily available at side of IBM Bluemix, which is **Cloudant NOSQL Database** Management System. This tool persist in the form of JSON documents. Hence no SQL command is required here. Using JSON instead of traditional SQL relative databases is much less memory consuming. As constraints and abstractions can only be made at application level, it gives way more power and processing speed to the database server.

3.4.1 ER Diagrams



ER Diagrams

3.4.2 Database Manipulation

The Cloudant NoSQL database service on Bluemix offers instances of Cloudant that you can bind and use immediately in your application. The current shared plan is an instance that offers up to 2 GB of transfers and up to 50,000 API calls free per month; a paid option allows unlimited transfers and API calls.

Note that the shared plan is actually a paid plan, but it's free if your monthly transfer and API quotas qualify.

1. Create an instance of the Cloudant NoSQL database service using the shared plan and name it "laurenlandscapesdb":

```
cf create-service cloudantNoSQLDB Lite laurenlandscapesdb
```

2. Bind this instance of the Cloudant NoSQL database service to the running app:

```
cf bind-service your_app_name laurenlandscapesdb
```

```
cf restage your_app_name
```

Add Cloudant NOSQL using Bluemix DashBoard GUI, we follow few STEPS.
Visit your Bluemix dashboard.

1. Click **Catalog**.
2. Check the **Data & Analytics** category.
3. Select **Cloudant NoSQL DB**.
4. Enter a descriptive name in the **Service name** field.
5. In the **Connect to** drop-down list, select the app you want to access Cloudant.
6. Click **Create**.
7. From your dashboard, open the Cloudant service instance.
8. Locate your database details under the **Service Credentials** tab.
9. Back on the **Manage** tab, click **Launch** to open the Cloudant dashboard.

Get the database credentials for the bound Cloudant NoSQL database service instance from the deployed app:
`cf env your-app-name`

This will display the environment variables available to the deployed app. The bound Cloudant NoSQL database service information will show up in VCAP_SERVICES. Look for something similar to this:

```
"VCAP_SERVICES": {  
  "cloudantNoSQLDB": [  
    {  
      "credentials": {  
        "host": "88396eb3-c56d13d6593-bluemix.cloudant.com",  
        "password": "3f7c34a",  
        "port": 443,  
        "url": "https://sfafsabluemix:sfafsafa@8-8400-ec56d13d6593-bluemix.cloudant.com",  
        "username": "232-bluemix",...}]
```

3.4.3 Database Connection Controls and Strings

Bluemix provides a set of libraries(.jar libs) that can be used for calling cloudant client from inside of a JAVA source code without interfering with the HTTP sessions. Other way for Database Connection and Control is with the help of HTTP APIs.

HTTP API

All requests to Cloudant go over the web, which means any system that can speak to the web, can speak to Cloudant. All language-specific libraries for Cloudant are really just wrappers that provide some convenience and linguistic niceties to help you work with a simple API. Many users even choose to use raw HTTP libraries for working with Cloudant.

Cloudant supports the following HTTP request methods:

- GET

Request the specified item. As with normal HTTP requests, the format of the URL defines what is returned. With Cloudant this can include static items, database documents, and configuration and statistical information. In most cases the information is returned in the form of a JSON document.

- HEAD

The HEAD method is used to get the HTTP header of a GET request without the body of the response.

- POST

Upload data. Within Cloudant's API, POST is used to set values, including uploading documents, setting document values, and starting certain administration commands.

- PUT

Used to put a specified resource. In Cloudant's API, PUT is used to create new objects, including databases, documents, views and design documents.

- DELETE

Deletes the specified resource, including documents, views, and design documents.

- COPY

A special method that can be used to copy documents and objects.

If the client (such as some web browsers) does not support using these HTTP methods, POST can be used instead with the X-HTTP-Method-Override request header set to the actual HTTP method.

Method not allowed error

Example error message

```
{  
  "error": "method_not_allowed",  
  "reason": "Only GET,HEAD allowed"  
}
```

If you use an unsupported HTTP request type with a URL that does not support the specified type, a 405 error is returned, listing the supported HTTP methods, as shown in the example.

JSON

Cloudant stores documents using JSON (JavaScript Object Notation) encoding, so anything encoded into JSON can be stored as a document. Files like images, videos, and audio are called BLObs (binary large objects) and can be stored as attachments within documents.

Distributed Systems

Cloudant's API enables you to interact with a collaboration of numerous machines, called a cluster. The machines in a cluster must be in the same datacenter, but can be within different 'pods' in that datacenter. Using different pods helps improve the High Availability characteristics of Cloudant.

An advantage of clustering is that when you need more computing capacity, you just add more machines. This is often more cost-effective and fault-tolerant than scaling up or enhancing an existing single machine.

For more information about Cloudant and distributed system concepts, see the CAP Theorem guide.

Replication

Replication is a procedure followed by Cloudant, CouchDB, PouchDB, and others. It synchronizes the state of two databases so that their contents are identical.

You can continuously replicate. This means that a target database updates every time the source database changes. Testing for source changes involves ongoing internal calls. Continuous replication can be used for backups of data, aggregation across multiple databases, or for sharing data.

Continuous replication can result in a large number of internal calls. This might affect costs for multi-tenant users of Cloudant systems. Continuous replication is disabled by default.

3.5 Methodology

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques.

Participants:

Source Participants of this projects can be from the corpus of any human user. As anyone with money can buy for sell products, everybody that qualifies above condition can be a source participant. But target participants need to lesser than that as to study whole corpus is a very lengthy task. For this application methodology study, we will use student crowd of our own college as target participants. Studying the structural behavior of our application in the campus zone itself.

Procedures:

Major entity management and executive nature of the application is being studied. Most important of which all are product addition, deletion, order and edit. Here products will be added to the collection of the application's list of products. They can be edited by their respective suppliers/retailers or the main ADMIN of the application. Other users can buy this product by adding it to their order list and checking out that order.

Users can be buyers and retailers both, as we want to achieve the capabilities of user being a trader for goods. Users also can edit their profile details for better recommendation of products and automated billing and shipping system for the user.

Users can check their order history for accountability of application transactions, and also if user is a retailer they can check sell history log for maintaining knowledge of inventory and resupply as soon as possible.

The Physical part of this project includes delivery and retrieval power of the system. Different users can put products of sale, leading to retrieval force for this application being needed. And as this project is a sales system, it also requires a delivery force for delivering products bought by users at their shipping addresses. All the rest of functionalities as virtual and can be easily operated on distributed networks.

CHAPTER 4: Implementation

4.1 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 today's 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-assists 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

Service name	Description
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 REST controllers 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 opt 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 abbreviated 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 10g, 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.

IBM Cloudant NOSQL Database: Used in this project for data persistence purposes. This is most essential part of this project. It is one of the services provided by Bluemix for application development.

Cloudant is an IBM software product, which is primarily delivered as a cloud-based service. Cloudant is an open source non-relational, distributed database service of the same name. Cloudant is based on the Apache-backed CouchDB project and the open source BigCouch project.

Cloudant's service provides integrated data management, search, and analytics engine designed for web applications. Cloudant scales databases on the CouchDB framework and provides hosting, administrative tools, analytics and commercial support for CouchDB and BigCouch. Cloudant's distributed CouchDB service is used the same way as standalone CouchDB, with the added advantage of data being redundantly distributed over multiple machines.

Cloudant was acquired by IBM from the start-up company of the same name. The acquisition was announced on February 24, 2014, The acquisition was completed on March 4 of that year.

Cloudant was founded by Alan Hoffman, Adam Kocoloski, and Michael Miller. The three met in the physics department at MIT where they worked with large data sets from experiments such as the Large Hadron Collider and the Relativistic Heavy Ion Collider. In early 2008 their ideas for fixing the "big data problem" caught the attention of Silicon Valley-based Y Combinator, which resulted in \$20,000 seed funding. The company also received an early seed round of \$1 million from Avalon Ventures in August 2010. Cloudant was designed for cloud computing, automatically distributing data across multiple servers in addition to scaling the database to accommodate web applications.

In August 2010, Cloudant released free BigCouch under an Apache License(2.0). Cloudant offered services including support, consulting services and training. Cloudant delivered their first product in the third quarter of 2010. Cloudant has over 2500 customers for its hosted service as of January 2011.

In November 2010, Cloudant was recognized as one of '10 Cool Open-Source Startups' by CRN. Cloudant was regularly recognized in the local Boston startup community, named as one of the 'Top 5 Database Startups' and 'Top Ten Cloud Computing Startups' in Boston's popular technology column by Joe Kinsella, 'High Tech in the Hub. On February 24, 2014, IBM announced an agreement to acquire Cloudant. The acquisition closed in March, after which Cloudant joined IBM's Information and Analytics Group.

In September, 2016, IBM Cloudant completed the donation of the BigCouch project to The Apache Software Foundation, resulting in the release of Apache CouchDB 2.0. CouchDB 2.0 incorporates many of the improvements made by Cloudant and BigCouch to the original CouchDB project, including clustering capabilities, a declarative query language and performance enhancements.

4.2 Coding Standards for Languages Used

JAVA Coding Standards

Each Java source file contains a single public class or interface. When private classes and interfaces are associated with a public class, you can put them in the same source file as the public class. The public class should be the first class or interface in the file.

All source files should begin with a c-style comment that lists the class name, version information, date, and copyright notice:

```
/*
 * Classname
 *
 * Version information
 *
 * Date
 *
 * Copyright notice
 */
```

The first non-comment line of most Java source files is a package statement. After that, import statements can follow. For example:

```
package java.awt;

import java.awt.peer.CanvasPeer;
```

Four spaces should be used as the unit of indentation. The exact construction of the indentation (spaces vs. tabs) is unspecified. Tabs must be set exactly every 8 spaces (not 4). Avoid lines longer than 80 characters, since they're not handled well by many terminals and tools.

Java programs can have two kinds of comments: implementation comments and documentation comments. Implementation comments are those found in C++, which are delimited by `/*...*/`, and `//`. Documentation comments (known as "doc comments") are Java-only, and are delimited by `/**...*/`. Doc comments can be extracted to HTML files using the javadoc tool.

Implementation comments are meant for commenting out code or for comments about the particular implementation. Doc comments are meant to describe the specification of the code, from an implementation-free perspective. to be read by developers who might not necessarily have the source code at hand.

Comments should be used to give overviews of code and provide additional information that is not readily available in the code itself. Comments should contain only information that is relevant to reading and understanding the program. For example, information about how the corresponding package is built or in what directory it resides should not be included as a comment.

Discussion of nontrivial or nonobvious design decisions is appropriate, but avoid duplicating information that is present in (and clear from) the code. It is too easy for redundant comments to get out of date. In general, avoid any comments that are likely to get out of date as the code evolves.

Note: The frequency of comments sometimes reflects poor quality of code. When you feel compelled to add a comment, consider rewriting the code to make it clearer.

Comments should not be enclosed in large boxes drawn with asterisks or other characters. Comments should never include special characters such as form-feed and backspace.

Simple Statements

Each line should contain at most one statement. Example:

```
argv++;      // Correct
argc--;      // Correct
argv++; argc--; // AVOID!
```

Compound Statements

Compound statements are statements that contain lists of statements enclosed in braces "{ statements }". See the following sections for examples.

- The enclosed statements should be indented one more level than the compound statement.
- The opening brace should be at the end of the line that begins the compound statement; the closing brace should begin a line and be indented to the beginning of the compound statement.
- Braces are used around all statements, even single statements, when they are part of a control structure, such as an if-else or for statement. This makes it easier to add statements without accidentally introducing bugs due to forgetting to add braces.

return Statements

A return statement with a value should not use parentheses unless they make the return value more obvious in some way. Example:

```
return;

return myDisk.size();

return (size ? size : defaultSize);
```

if, if-else, if else-if else Statements

The if-else class of statements should have the following form:

```
if (condition) {
    statements;
}
```

```
if (condition) {  
    statements;  
} else {  
    statements;  
}
```

```
if (condition) {  
    statements;  
} else if (condition) {  
    statements;  
} else {  
    statements;  
}
```

for Statements

A for statement should have the following form:

```
for (initialization; condition; update) {  
    statements;  
}
```

An empty for statement (one in which all the work is done in the initialization, condition, and update clauses) should have the following form:

```
for (initialization; condition; update);
```

When using the comma operator in the initialization or update clause of a for statement, avoid the complexity of using more than three variables. If needed, use separate statements before the for loop (for the initialization clause) or at the end of the loop (for the update clause).

while Statements

A while statement should have the following form:

```
while (condition) {  
    statements;  
}
```

An empty while statement should have the following form:

```
while (condition);
```

do-while Statements

A do-while statement should have the following form:

```
do {  
    statements;  
} while (condition);
```

switch Statements

A switch statement should have the following form:

```
switch (condition) {  
case ABC:  
    statements;  
    /* falls through */  
case DEF:  
    statements;  
    break;  
case XYZ:  
    statements;  
    break;  
default:  
    statements;  
    break;  
}
```

Every time a case falls through (doesn't include a break statement), add a comment where the break statement would normally be. This is shown in the preceding code example with the `/* falls through */` comment.

Every switch statement should include a default case. The break in the default case is redundant, but it prevents a fall-through error if later another case is added.

try-catch Statements

A try-catch statement should have the following format:

```
try {  
    statements;  
} catch (ExceptionClass e) {  
    statements;  
}
```

A try-catch statement may also be followed by finally, which executes regardless of whether or not the try block has completed successfully.

```
try {  
    statements;  
} catch (ExceptionClass e) {  
    statements;  
} finally {  
    statements;  
}
```

4.3 Test Techniques and Test Plans

In every company, testing is the most vital and precious stage in the software development life cycle. But, the technique by which it is performed, dissimilar from one company to another company. Software testing has turn into a part of programming and it is good to begin testing from the first stage, to avoid complexity by repairing the errors at the last stage. Also in software development life cycle (SDLC) testing includes pick up consistency, interpretation and other major elements, which may explain under software requirement specification. Client may stay further for software delivery, but they don't akin to work with imperfection software. It is desirable to perform the testing process from the first stages, with consider to the Software Development Life Cycle to elude any difficulty.

- Recognition of Error and Faults

Testing step is one step which resolves the errors and faults in the software application. These errors may be in unit level or in system level. After going through so many testing the application will be free of errors that may be disturbing the application.

- Statistics to Shareholders and Status of Organization

Testing stage helps to know the condition of product and work standards. The stakeholders get better data through testing stage about utility value too.

- Enhancement in Product Standards

Testing can help to know the real result and the probable result. It also helps to pick up the standards of the software. With proper testing an application can come out of bugs and build up ideal software for the end-users.

- Technical Significance

Testing segment is significant for technical characteristics of any SDLC, as the software then completed with technically satisfied.

- To Succeed of any Contentious Programmers

Ideal testing functions and tools aid to evolve up the product in business and keep programmers away from the other contestant. Going through all stages of testing, the software application will be more bugs free, protected and technically sound.

- Free from any Risk

Whenever going to develop any software, testing is an essential part. When develop software without any testing then it may cause lots of risks to the end users. To free everyone from any risk, it is essential that to go under all testing stages.

- Enhanced Standards

Appropriate tested application provides additional assurance of build up with best software. Moreover, it refines standards of application as incessant and all types of testing stages have prepared a protected and harmless software application that could be worn by the end users.

- Confirmation and Corroboration

One of the major targets of testing stage in SDLC is for confirmation and corroboration. Testing is greatly used in confirmation and corroboration method. Depending on the result we can compare among standards of several software application.

- Credibility Evaluation

Testing stage also insist this important issue. If the software application has gone through all the testing types (like unit testing, regression testing etc.), the application will surely be a reliable one. So, testing evaluate credibility of software application. Testing provides the greatest analytical process to give equipped testing on product ensuing in a credible product.

- Demonstrate Accessibility and Feasibility

One of the most significant targets of testing is to demonstrate the product is both accessible and functional. Accessibility testing is where the application is delivering to a select assembly of users and their functioning with the application is noticed. All type of a user's communication with the application, like easiness of applies and whenever users are getting troubles, are preserved and examined.

- Avoid Fault Immigration

In the first stage of SDLC, most of the faults have been found. If the faults can be noticed earlier, then these may be prohibited from immigrating to the following progress stage. If the errors could be discover previously then the saving of software development cost will be vast.

- Commercial Significance

A full tested software application will have excellent business aspects. As all are like to work with reliable and trusted application in commercial.

Three testing techniques will be used in this project. Unit, Integration and System Testing.

Unit Testing will be used for testing every individual Module independent to the system for any fault or bug. Integration Testing will be done to verify that all modules are working correctly with each other. And finally, System Testing will be done to validate the system as a whole to reach the project's functional requirements. JUnit Testing techniques will be implemented parallel to the source as to get better testing assertions.

JUnit is a Regression Testing Framework used by developers to implement unit testing in Java, and accelerate programming speed and increase the quality of code. JUnit Framework can be easily integrated with either of the following –

- Eclipse
- Ant
- Maven

JUnit test framework provides the following important features –

- Fixtures
- Test suites
- Test runners
- JUnit classes

Fixtures is a fixed state of a set of objects used as a baseline for running tests. The purpose of a test fixture is to ensure that there is a well-known and fixed environment in which tests are run so that results are repeatable. It includes –

- setUp() method, which runs before every test invocation.
- tearDown() method, which runs after every test method.

A test suite bundles a few unit test cases and runs them together. In JUnit, both @RunWith and @Suite annotation are used to run the suite test. Test runner is used for executing the test cases.

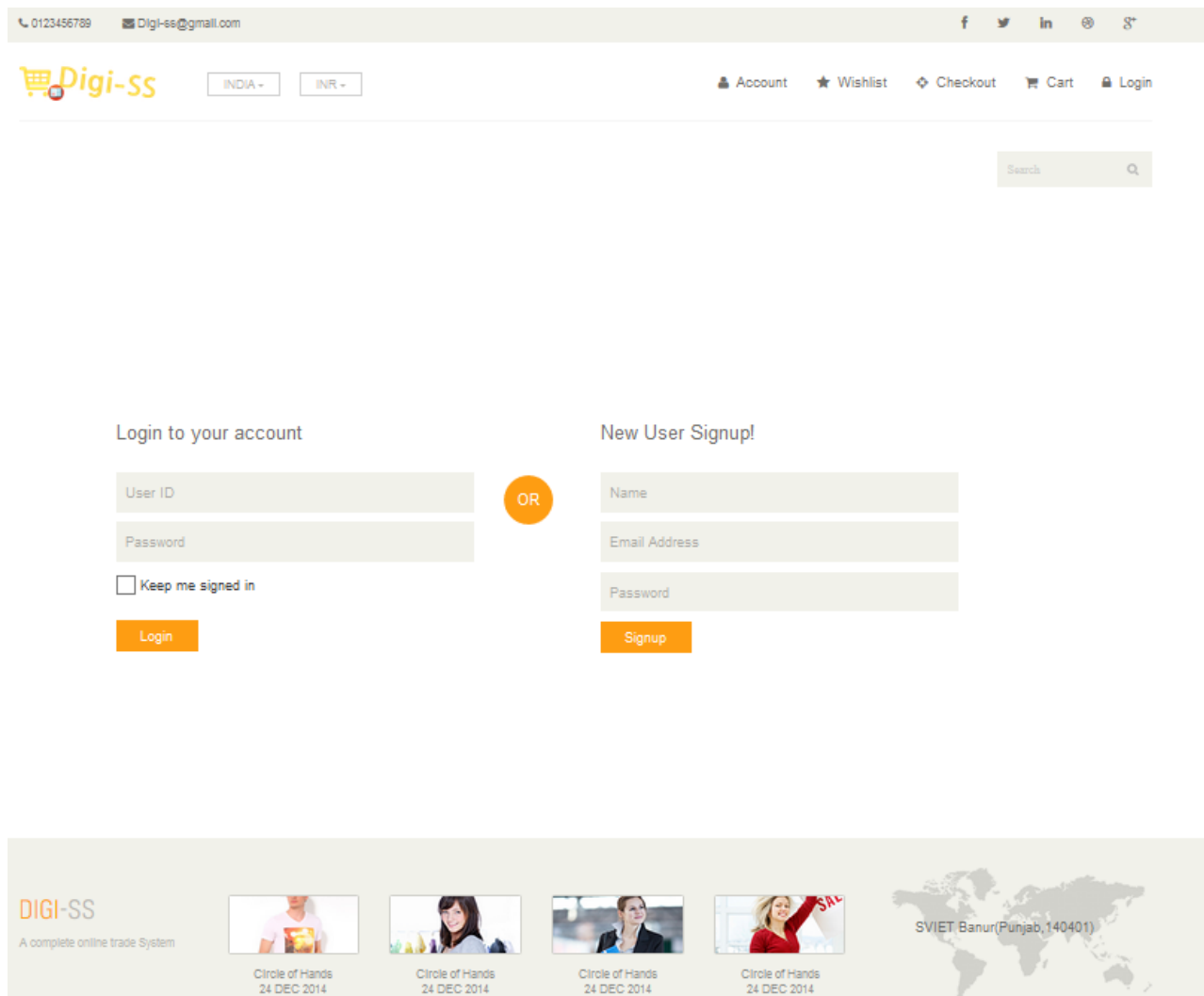
JUnit classes are important classes, used in writing and testing JUnits. Some of the important classes are –

- Assert – Contains a set of assert methods.
- TestCase – Contains a test case that defines the fixture to run multiple tests.
- TestResult – Contains methods to collect the results of executing a test case.

CHAPTER 5: Results and Discussions

As a result to our effort, we have finally created the Digitized Sales System(DSS) that we were required to build. Due to this rigorous documentation and engineering approach towards application development, we could easily build a corporate level application. Our basic functionalities are complete which included user profiling, product shopping and selling at same place by similar kind of users, transitional logging, invoice forwarding and maintenance. And we have set further functional goals like trading implementation, bidding structure implementation, Tone Analysis on product reviews and better intelligent recommendation engine building and deployment.

5.1 Snapshots of System



The screenshot displays the login and registration interface of the Digi-ss system. At the top, a header bar includes contact information (0123456789, Digi-ss@gmail.com), social media icons, and navigation links (Account, Wishlist, Checkout, Cart, Login). Below the header, the Digi-ss logo is accompanied by country (INDIA) and currency (INR) dropdowns, and a search bar. The main content area is divided into two sections: 'Login to your account' and 'New User Signup!'. The login section features input fields for 'User ID' and 'Password', a 'Keep me signed in' checkbox, and a 'Login' button. The registration section includes input fields for 'Name', 'Email Address', and 'Password', and a 'Signup' button. An 'OR' button is positioned between the two sections. The footer contains the Digi-ss logo, a tagline 'A complete online trade System', four promotional images for 'Circle of Hands' dated '24 DEC 2014', and a world map with the address 'SVIET Banur(Punjab,140401)'.

0123456789 Digi-ss@gmail.com

f t in

Digi-ss INDIA - INR -

Account Wishlist Checkout Cart Login

Search

Login to your account

User ID

Password

☐ Keep me signed in

Login

OR

New User Signup!

Name

Email Address

Password

Signup

DIGI-SS

A complete online trade System

Circle of Hands 24 DEC 2014

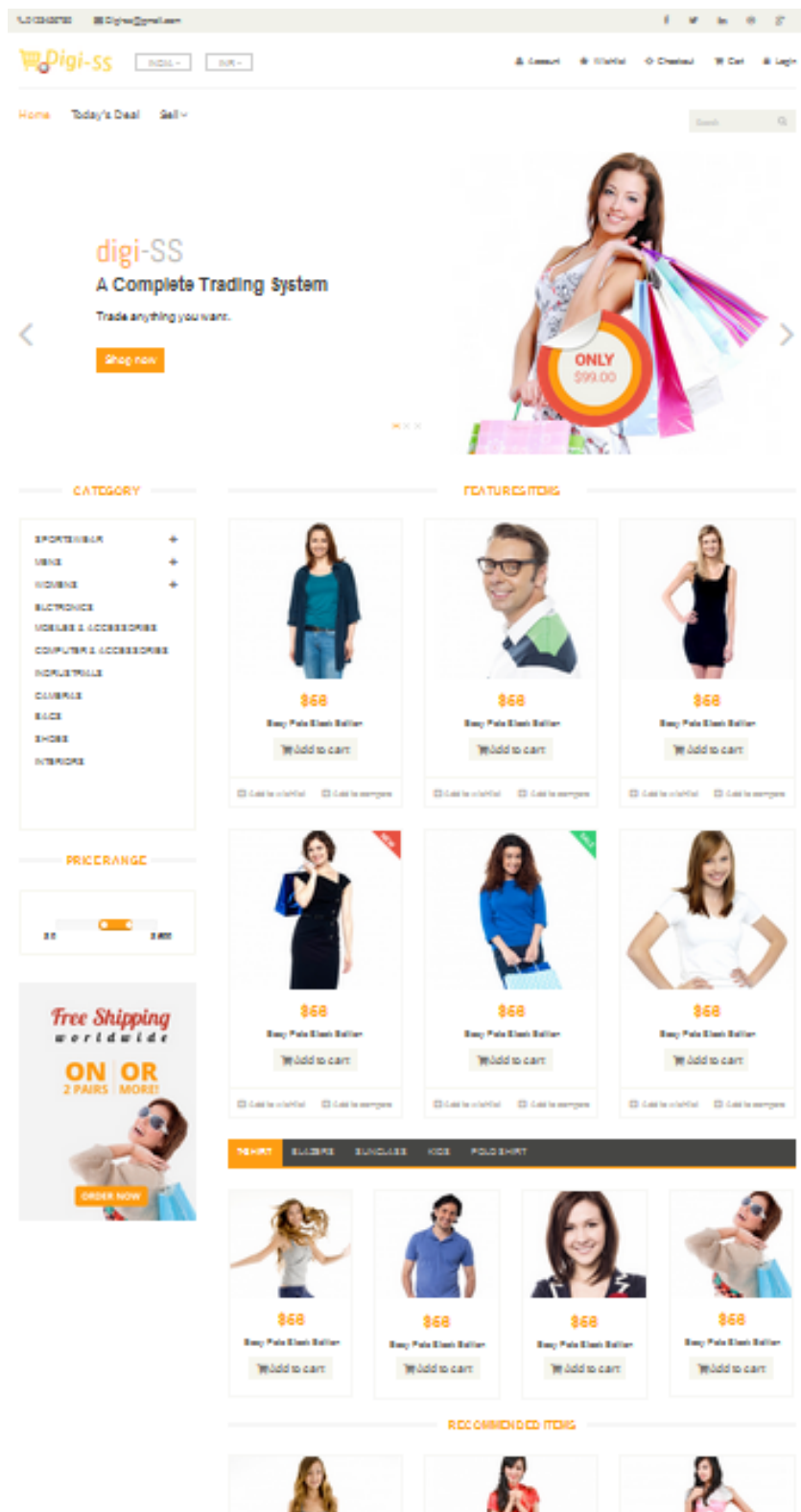
Circle of Hands 24 DEC 2014

Circle of Hands 24 DEC 2014

Circle of Hands 24 DEC 2014

SVIET Banur(Punjab,140401)

Login Page



home page

0123456789
Digi-ss@gmail.com
f
w
in
g

Digi-ss
INDIA
INR
Account
Wishlist
Checkout
Cart
Login





Home
Today's Deal
Sell
Search

CATEGORY
SPORTSWEAR
MENS
WOMENS
KIDS
FASHION
HOUSEHOLDS
INTERIORS
CLOTHING
BAGS
SHOES

BRANDS
ACNE (50)
GRÜNE ERDE (56)
ALBIRO (27)
RONHILL (32)
ODDMOLLY (5)
BOUDESTIUN (9)
RÖSCH CREATIVE CULTURE (4)

PRICE RANGE
\$ 0
\$ 800

Free Shipping worldwide

NEW
Anne Klein Sleeveless Colorblock Scuba
Web ID: 1089772
★★★★☆
US \$59
Quantity: 3
Add to cart
Availability: In Stock
Condition: New
Brand: E-SHOPPER
Like
Tweet
Pin It
Share
1.7M
ZOOM

DETAILS
COMPANY PROFILE
TAG
REVIEWS (5)

EUGEN
12:41 PM
31 DEC 2014

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Write Your Review
Your Name
Email Address
Rating: ★★★★★
Submit




Product Description Page

0123456789
Digi-ss@gmail.com
f
w
in
g
+

Digi-ss
INDIA
INR
Account
Wishlist
Checkout
Cart
Login

Search

Home
Shopping Cart

Item	Price	Quantity	Total
 Colorblock Scuba Web ID: 1089772	\$59	+ 1 -	\$59
 Colorblock Scuba Web ID: 1089772	\$59	+ 1 -	\$59
 Colorblock Scuba Web ID: 1089772	\$59	+ 1 -	\$59

What would you like to do next?
Choose if you have a discount code or reward points you want to use or would like to estimate your delivery cost.

☐ Use Coupon Code
☐ Use Gift Voucher
☐ Estimate Shipping & Taxes
Country: United States Region / State: Select Zip Code:

Get Quotes
Continue

Cart Sub Total	\$59
Eco Tax	\$2
Shipping Cost	Free
Total	\$61

Update
Check Out

order ckckout page

payment gateway page

5.2 Back Ends Representation

Due to the use of IBM Cloudant NOSQL Database, we do not have tables, but have documents. Different entities are saved in different databases in forms of JSON format documents. JSON: **J**ava**S**cript **O**bject **N**otation. JSON is a syntax for storing and exchanging data. JSON is text, written with JavaScript object notation. Since the JSON format is text only, it can easily be sent to and from a server, and used as a data format by any programming language.

5.2.1 Snapshots of Database Documents

```
3  {
4      "name": "name",
5      "productId": "id13543",
6      "description": "desc",
7      "category": { "_id": "id15313843434",
8      "_rev": "rev15345435",
9      "categoryId": "id11111",
10     "name": "cat_name",
11     "description": "cat_desc",
12     "imgUrl": "imgurl2.com",
13     "bannerUrl": "bannerurl.co.in"},
14     "searchKey": ["key1", "key2", "key3"],
15     "imgUrl": "imgurl.com",
16     "unitCost": "20.21",
17     "quantity": "20",
18     "availability": true,
19     "reorderQuantity": "0",
20     "supplier": { "name": "name",
21         "supplierStatus": "status",
22         "address1": "add 1",
23         "address2": "add 2",
24         "city": "city",
25         "postalCode": "110011",
26         "province": "province",
27         "country": "INDIA",
28         "phone": "1234567891"}
29 }
```

product document JSON format

```

4  { "userId":"id15315",
5    "orderDate":"date-month-year","orderStatus":"status",
6    "ccType":"ccType","ccNumber":"ccnumber","ccExpiryMonth":"mm","ccExpiryYear":"yyyy",
7    "billFName":"F Name","billLName":"L Name",
8    "billAddress1":"add 1","billAddress2":"add 2",
9    "billCity":"city","billState":"state","billProvince":"province","billPostalCode":"110011",
10   "billCountry":"INDIA",
11   "shipToBillingAddress":false,
12   "shipFName":"F Name","shipLName":"L Name",
13   "shipAddress1":"add 1","shipAddress2":"add 2",
14   "shipCity":"city","shipState":"state","shipProvince":"province","shipPostalCode":"110011",
15   "shipCountry":"INDIA",
16   "subTotal":"20.21",
17   "shipMethod":"method","shipCost":"1.0",
18   "total":"21.21",
19   "items":[{"quantity":"2","lineTotal":"5","productId":"id"},
20           {"quantity":"2","lineTotal":"5","productId":"id"},
21           {"quantity":"2","lineTotal":"5","productId":"id"}]
22 }

```

order document JSON format

```

4  { "userId":"id12123546468",
5    "userPass":"pass156834384",
6    "email":"email@gmail.com",
7    "isAdmin":true,
8    "fName":"F Name",
9    "lName":"L Name",
10   "address1":"address1",
11   "address2":"address2",
12   "city":"city",
13   "state":"state",
14   "postalCode":"110011",
15   "province":"province",
16   "country":"INDIA",
17   "phone":"1234567891",
18   "lang":"eng",
19   "favCategory":"category",
20   "showPetTips":true}

```

user document JSON format

CHAPTER 6: Conclusion and Future Scope

digi-ss is a Digitized Sales System developed under guidance of KVCH trainers for final year project submission. It is hosted on cloud foundry created by IBM also know as BLUEMIX cloud foundry. It uses Liberty as bluemix server hosting and cloudbant NOSQL as database management service provided by bluemix itself. Main jobs possible under this application are user profiling, product sell and buy, product reviewing and critics. This was a 6 months training project which will close in july' 17.

For future scopes, we have certainly thought of trading purchases between pair of users. Where a pair will decide to trade their products for others products and application system will just reverse the ownership of those products, leading to completion of trade. Trade being a long lost business venture can be easily virtualized and profited upon. And other scope that we have thought of is bidding of a product between users. Where a product could be open for bidding for a certain interval of time period, and traditionally, highest bidder takes the product. We can also build better intellectual recommendation engine for recommending products to the user in future versions. Using machine learning models and concepts, it can be easily implemented.

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.github.com)