**Project Title** : eTohfa (Online Gift Shop)

**Team** : Shital Meshram

Varsha Gaikwad

**SRS Document :**

1. **Introduction**
   1. **Purpose:**

The Online Shopping System for products shop web application is intended to provide complete solutions for vendors as well as customers through a single get way using the internet. It will enable vendors to setup online shops, customer to browse through the shop and purchase them online without having to visiting the shop physically.

* 1. **Scope:**

This system allows the customer's to maintain their shopping cart for add or remove the product over the internet. Shoppers will be able to manage their products stock and its status i.e the product is available of out of stock.

* 1. **Definitions:**
* OSS => Online Shopping System
* SRS => Software Requirement Specification
* GUI => Graphical User Interface
* Portal => Personalized Website
* Stackholder => The person who will participate in the System. And Onwer of system Ex. Customer, Adminstrator , Shopper
* UML => Software Engineering Notation for visualising System in the form diagrams
* SSL => Secure Socket Layer used for providing restricted access to application.
* BOD => Board Of Directors (Management).
* RDBMS => Relational Dadabase Management System.
* CLUSTERS => Group of independent servers.
  1. **Overview:**

This System provides an easy solution to customer's to buy the product without going to the shop and also shop owner to sale their products online. It also give the verities to gifts so that you can gift to an individual.

* 1. **Additional Information:**

The system work on internet server, so it will be operated by any end user for the buying purpose with secure platform. This system protects the integrity of the sellers and buyers, provides easy buying policies and offers.

* 1. **General Description:**

The Online shopping application helps to manage the items in the shoppers' carts and also helps customers to purchase. The online shopping system will use the internet as the sole method for selling goods to its consumers.

* 1. **Functional Requirement:**

This section provides requirement overview of the system. Various functional modules that can be implemented by the system will be-

* + 1. **Description:**

Registration if customer wants to buy the product then he/she must be registered, Unregistered user can not get to shopping cart. Login Customer logins to the system by entering valid user id and password for shopping. End User can Browse Products , their categories as well, he/ she can add products to her/his wishlist. Payment for customer; there are many of secure billing will be prepaid as debit or credit cart, cash on delivery. Logout after the payment of the product the customer will logged out. Report Generation after all transaction the system can generate the portable file (.pdf) then sent one copy to customer's Email- address and another one for the system database to calculate the monthly transaction. The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the RDBMS (also known as the back-end). A client/server system is a distributed system in which, Some sites are client sites and others are server sites.

All the data resides at the server sites.

All applications execute at the client sites.

* 1. **Technical Issues:**

This system will work on client-Server architecture. It will require an internet server.

The system should support some commonly used browser such as Chrome etc.

Interface Requirement Various interfaces for the product could be

1. Login Page
2. Registration form

There will be a screen displaying information about product that the shop having.

The customers may select the different options which will be open in another screen as

1. Login Page
2. Registration Form
3. Product Page
4. Shopping Cart
5. Shipping Details
6. Purchase history
7. Account Settings
8. Payment Gateways

* 1. **Hardware Interface:**

The System must run over the internet, All the hardware shall require to connect to internet will be hardware interface for the system.

e.g. modem, WAN, LAN

Specialized Server Infrastructure Hardware

The system should use distrubuted servers i.e cloud for managing large amount of data so as to make it appear as single unit for end-user.

The system should have proper clusters for backup.

* 1. **Software Interface:**

The system is on server so it requires the any scripting language like JSP or PHP or ASP, ETC.

The system should be able to exchange data using XML, JASON or any advance technology.

The system require DataBase also for the store the any transaction of the system like MySql or oracle, or SQL server etc.

System also require DNS (Domain Name space) for the naming on the internet.

<http://www.eTohfa.in>

<http://www.amazon.in>

At the end-user need web browser for interact with the system.

* 1. **Performance Requirement:**

There is no performance requirement in this system, because the server request and respone to client is totally based on internet connection of enduser.

* 1. **Design Constrains:**

This system should be developed using Standard Web Page Development Tool , which confirms GUI standards such like HTML, XML, JSON,etc.

The system should support various RDMS and Cloud Technologies.

* 1. **Non-Functional Requirements**
     + 1. **Security**

**SSL**

The System use SSL (Secure Socket Layer) in all transactions that include any confidential customer information.

The system must automatically log out all customers after a period of inactivity.

The system should not leave any cookies on the customer's computer containing user's password.

The system's back-end servers shall only be accessible to authenticated administrators.

Sensitive data will be encrypted before being sent over insecure connections like internet.

The proper firewalls should be developed to avoid intrusions from the internal or external sources.

* + - 1. **Reliability:**

The system provides storage of all databases on redundant computers with automatic switchover.

The main pillar of reliability of the system is the backup of the database which is continuously maintained and update to reflect the most recent changes.

* + - 1. **Availability:**

The system should be available at all times. meaning the user can access it using web browser, only restricted by the down time of the server on which the system runs. In case of a of a hardware failure or database corruption, a replacement page will be shown.

uptime : It mean 24 \* 7 availability

100%--------------

99.9%

99.999%

99.9999%

* + - 1. **Maintainability:**

A commercial database is used for maintaining the database and application server takes care of the site.

The maintainability can be done efficiently.

* + - 1. **Portability:**

The application is HTML and scripting language based (Javascript). So the end user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available or will be available in the future. An end-user is used this system on an OS either it is Windows or Linux. The System shall run on PC, Laptops and PDA. etc.

The technology should be transferable to different environments easily.

* + - 1. **Accessibility:**

Only registered users should be allowed to process the orders after authentications.

Only GUI access of the system should be permitted to end users.

* + - 1. **Policies:**

The system should adhere to all the legal formalities of the particular countries.

The system should maintain security related to sensitive data.

* + - 1. **Efficiency:**

The system should provide good throughput and response to multiple users without burdening the system by using appropriate number of servers.

* + - 1. **Safety:**

Software should not harm ethical and environmental conditions of the end users machine.

* + - 1. **Modularity:**

The system should have user friendly interface.

It should be easily updated, modified and reused.

* 1. **Operational Scenario:**
     + 1. **Customer Interaction:**

The Customer want to buy item. The system shows all product categories to customer. If customer select item then those items are listed in shopping cart for buying. The payment will be made with credit card or debit card. If customer wants to cancel the order before shopping then he or she can cancel it. Customer can see the buying report on account details. Customer will receive email about purchase done.

* + - 1. **Shopper Interaction:**

* + - 1. **Staff Interaction:**

* + - 1. **BOD (Board Of Directors)**

* 1. **Preliminary Schedule: :**
* Login
* Manage customer database Browse category
* add or remove item from cart
* Manage customer database
* update item category
* approve/reject shop creation
* shipping order
* Logout
* Give feedback
* Payment :

By Credit Card, By Debit Card, By online banking or By Cash on Delivery

* Visit Site
* Create new account
* View account details
* Cancel order before shipping
* Registration
* Order tracking
* Return Policies
* Customer Support

**Application Architecture:**

Application = Logic + data

Logic =(UI Logic + Business Logic + DataAccess Logic)

Data =( structured data , Non Structured data)

Desktop application Architecture and Online Application Architecture

**Online Application:**

Web based Application :

deployed on web and accessed by users from anywhere

ShoppingApplication------Web portal-- used remotely by endusers, vendors, employees

**Logic:**

**UI Logic:**

Web Pages + HTML controls + Web Components (Angular)

Navigation : (UI Routing) HTML Links, Routing mechansim

Data Binding : DOM + JSP tags (JSTL) + {{}} ngModel,

Event Binding : action handlers

HTTP Request: GET:----------------doGet

POST:---------------doPost

PUT:-----------------doPut

DELETE:-----------doDelete

Client Side UI----------------HTML, CSS, javaScript, bootstrap

UI (Client Side UI Framework)

Angular, React, Vue,..........

**Web Logic: ( Server Side processing)**

Server UI---------------JSP, servelet, ( classical java web technology)

spring MVC ( to take adv of MVC design Pattern using ready made frmwrk)

Model, View, Controller

Router

(SOA layer)

Spring Boot api

CRUD REST API

ORM Technique:Hibernate ( ORM),

JPA,

JDBC ( database Connectivity)

State management:

Client Side state management

cookies, querystring, form collection, hidden variables

local storage, session storage, Web sql,

Server Side state management

session, Cache,

database

**Business Logic:**

Java console application will be used to test your business Logic

Core Java:

will contain

1.business query processing

2.business operation managment

3.Business data manipulation

from eTohfa Online gift shop point of view

1. ProductCatalog
   1. Product Inventory
      1. AddProduct
      2. updateProduct
      3. deleteProduct
   2. Product Category
      1. getAllProduct
      2. getProductCategory
      3. addProductCategory
      4. deteletProductCategory

1. Shopping Cart:
   * 1. Adding to cart
     2. Removing from cart
     3. Getting all element from cart
2. Order Processing
   * 1. Buy
     2. Cancel
     3. Address Details
     4. Total bill
     5. getAllOrders
     6. getOrderByCatagory
     7. getOrderHistory
3. Receiver Details
   * 1. Insert Receivers details
     2. GetReceivers details
     3. updateReceiversDetails
     4. deleteReceiversDetails
4. Shopper
   * 1. getAllShopper
     2. getShopperbyShopTitle
     3. getShopperById
5. Payment Processing
   * 1. getPaymentDetails
     2. checkPayments
     3. getCustomerByOrderPayment
     4. getAllPayment
     5. getPaymentbyCustomer
     6. getPaymentSuccessfullReport
6. Shipping
   * 1. Get Address Details
     2. Get Shipping Status
     3. GetShippedOrder
     4. getUnshippedOrder
7. CustomerMembership
   * 1. Registration
     2. Validate User
     3. Get Person Details
     4. Change password
     5. Forget password
     6. Edit profile
8. Claims
   * 1. Login
     2. Logut

**Tables:**

1. Employee/staff:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Empid (PK) | Ename | Email | Contact no | role |
|  |  |  |  |  |

1. Customer:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Customerid (PK) | firstName | LastName | Email | Password | Address | Contact No |
|  |  |  |  |  |  |  |

1. Shopper:

|  |  |  |  |
| --- | --- | --- | --- |
| ShopperId (pk) | ShopName | Email | Contact No |
|  |  |  |  |

1. Product:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ProductId(PK) | Name | Description | Price | Image | categoryId(FK) | ShopperId(FK) |
|  |  |  |  |  |  |  |

1. Category

|  |  |
| --- | --- |
| categoryId(PK) | CategoryName |
|  |  |

1. Order:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| orderId(PK) | Qty | Total Amt | CustomerId(FK) | ReceiversId(FK) |
|  |  |  |  |  |

1. Shipment:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ShipId(PK) | ShippingDate | status | OrderId(FK) | empId(FK) |
|  |  |  |  |  |

1. Payment:

|  |  |  |  |
| --- | --- | --- | --- |
| PaymentId(PK) | PaymentDate | Amount | OrderId(FK) |
|  |  |  |  |

1. ReceiverDetails (gift receiver):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| receiverId(PK) | receiverName | Address | ContactNo | CustomerID(FK) |
|  |  |  |  |  |

1. Bank details
2. Accid
3. Account No
4. IFSC
5. balance

**Data :**

Structured Data:

RDBMS

fields

tables

contstraints

Add some dummy records in your newly created database

write reusable SQL queries against those database tables to check bussiness Queries

Test those SQL queries on existing dummy database you built

Unstrctured:

NoSQL

Social Media

MongoDB