Project Name	Make an E-commerce Website for Sporty Shoes
Developers Name	Prathamesh Chinchamalatpure
Phase	Implement Frameworks The DevOps way
Part	Writeup

This document contains the following contents:

This project is hosted at https://github.com/Prathameshcgitnew/Phase4ProjectDemo.git

The project is developed by Prathamesh Chinchamalatpure

1.Sprint Planning and Task Achieved in them

This project is planned to be completed in 2 sprint. Task that are assumed to be completed are:

1 Developing flow for project

2 Initializing git repositories to track changes as project progresses

3 Writing Java code to fulfill requirement for the project

4 Testing project by giving diverse input

5 Pushing code to github

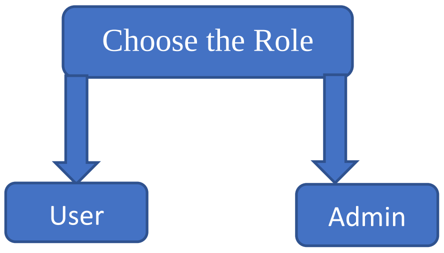
6 Creating document of specification highlighting application appearances , capabilities and user interactions

2. Core concepts required in project

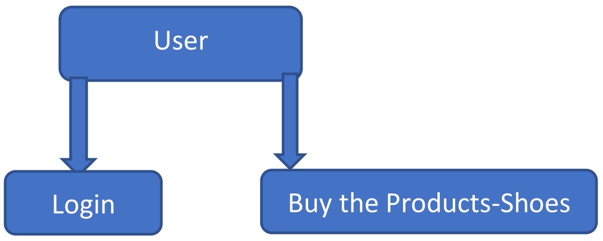
String, Collection Framework , Control constructs, Exception Handling ,Servlet ,JSP , Filter , Spring-core, Spring-MVC , Spring-hibernate Template etc.

3.Functional Flow Of Project

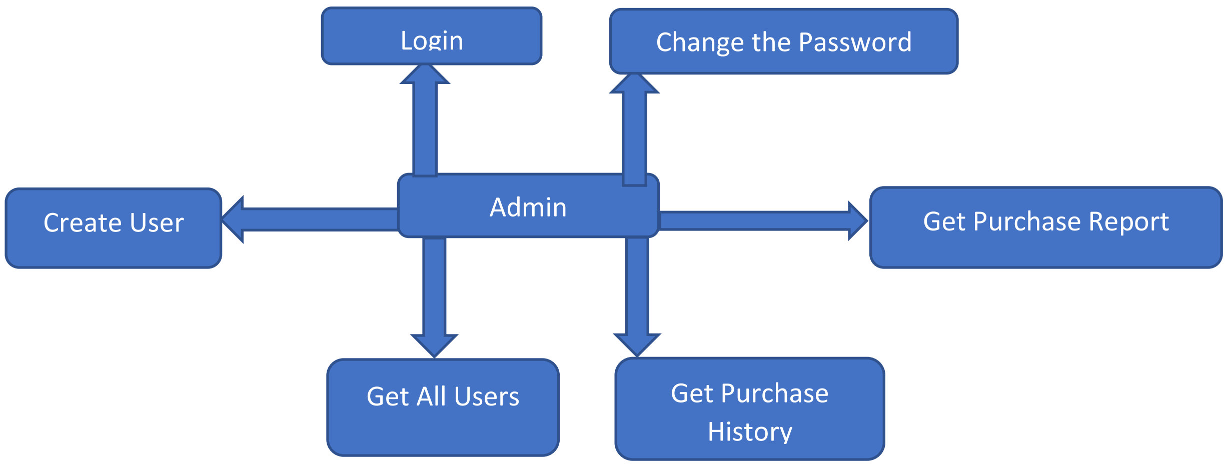
1.



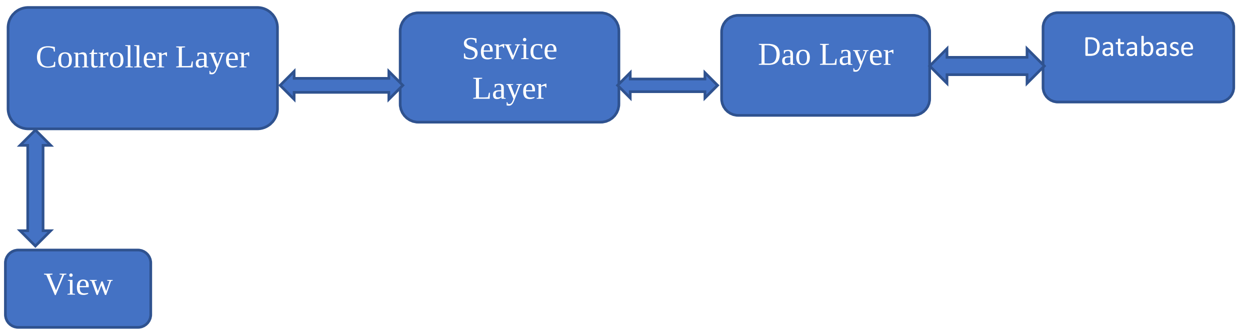
2.



3.



Structural Flow :



4 . Product capabilities, appearance and user interactions

To demonstrate the product capabilities, below are the sub-sections configured to highlight appearance and user interactions for the project:

1 Creating the project in STS

2 Writing a program in Java for the entry point of the application (indexprev.jsp , adminLogin.jsp , userLogin.jsp & AdminController.java , UserController.java)

3 Writing a program in jsp page to display options available for the Admin (LinkPage.jsp and adminpasswordChange.jsp.)

4 Writing a program in in jsp , controllers for options available to User .

5 Writing a program in Java at Dao, Service and Controller Layer.

6 Writing a programs for functional flows (Different Controller Programs)

7 Writing a program to for entities required for database.

8 Pushing the code to the Git repository.

1 Creating the project in STS

1. Open STS.
2. File->New->Project->Dynamic web Project
3. Give it name.
4. Right Click on project Name->Configure-> Convert to maven project
5. Create package and create controller , services , models and jsp pages.

2 Writing a program in Java for the entry point of the application (indexprev.jsp , adminLogin.jsp , userLogin.jsp & AdminController.java , UserController.java)

Indexprev.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<html>
<head>
<meta charset="ISO-8859-1">

<title>Role</title>

<link href="<c:url value="/css/login1.css" />" rel="stylesheet"></link>

</head>
<body>
      <div class="center">
            <div class="center1">
            <h5>Click the button as per your Role</h5>
```

```
                        <div class="center1">
                         <form action="HomePage" method="GET">
                              <button type="submit" >User</button>
                        </form>
                        <form action="AdminLogin" method="POST">
                             <button type="submit">Admin</button>
                        </form>

                        </div>
                        <br /> <br>



                  </div>
            </div>
</body>
</html>
```

adminLogin.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<html>
<head>
<meta charset="ISO-8859-1">

<title>Login</title>

<link href="<c:url value="/css/login1.css" />" rel="stylesheet"></link>

</head>
<body>
      <div class="center">
            <div class="center1">
                  <h1>Admin Login</h1>
                  <form action="adminLoginProcess" method="post">

                        <label>Username : </label> <input type="text"
                              placeholder="Enter Username" name="username"
required> <br>
                        <br> <label>Password : </label> <input

type="password"

                              placeholder="Enter Password" name="password"
required> <br>
                        <br>
                        <div class="center1">

                              <button type="submit">Login</button>
                              <br> <br>
                  </form>
                  <form action="changepasswordProcess">
                        <button type="submit">Change Password</button>
                  </form>

            </div>
            <br /> <br> <input type="checkbox" checked="checked">
            Remember me
      </div>
      </div>
</body>
</html>
```

userLogin.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<html>
<head>
<meta charset="ISO-8859-1">

<title>Login</title>

<link href="<c:url value="/css/login1.css" />" rel="stylesheet"></link>

</head>
<body>
      <div class="center">
            <div class="center1">
                  <h1>User Login</h1>

                 <form action="Login" method="POST">

                  <label>Username : </label> <input type="text"
                        placeholder="Enter Username" name="username"
required> <br>
                  <br> <label>Password : </label> <input type="password"
                        placeholder="Enter Password" name="password"
required> <br>
                  <br>
                  <div class="center1">
                        <button type="submit">Login</button>
                  </div>
                  <br /> <br> <input type="checkbox" checked="checked">
                  Remember me



            </div>
      </div>
</body>
</html>
```

AdminController.java

```
package com.springmvc.controller;


import java.util.List;


import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;

import com.springmvc.dao.AdminDao;
import com.springmvc.dao.AdminDao;
import com.springmvc.model.Admin;
import com.springmvc.service.AdminService;

@Controller
public class AdminController {

	@Autowired
	private AdminDao adminDao;

	@Autowired
	private AdminService adminService;

	@RequestMapping("/adminLoginProcess")
   public String adminLoginValidation(@ModelAttribute Admin admin) {

		System.out.println(admin);
		List<Admin> list=adminDao.getAllAdmins();
		System.out.println(list);
		for(Admin a:list) {
				if(a.getUsername().equals(admin.getUsername()) &&
a.getPassword().equals(admin.getPassword())){


					System.out.println("Admin login successful");
			return "LinkPage";
				}


				}
				return "invalidCredential";
			}


		@RequestMapping(path="/changeThePasswordCheck",method=RequestMethod.POS
T)
		public String changePasswordCheck(@RequestParam("username") String username,
				@RequestParam("password") String password,
				@RequestParam("confirmpassword") String confirmpassword,Model
model
				)
		{
			System.out.println(username);
			System.out.println(password);
			System.out.println(confirmpassword);


			List<Admin> adminlist=adminDao.getAllAdmins();


			for(Admin a:adminlist) {


				if(a.getUsername().equals(username)  &&
password.equals(confirmpassword)) {


					this.adminDao.deleteAdmin();
					Admin newAdmin=new Admin();
```

```
                    // newAdmin.setId(1);
                        newAdmin.setUsername(username);
                        newAdmin.setPassword(confirmpassword);
                        this.adminService.enterAdminCredential(newAdmin);
                        //model.addAttribute("Header", "Please Enter new
credentials");
                        //model.addAttribute("Description", "Please Enter new
credentials");
                        return "adminLogin";
                   }
            }

            return "AdminChangeInvalidCredential";
      }




       }
```

UserController.java

```
package com.springmvc.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import com.springmvc.model.Product;
import com.springmvc.model.PurchaseTableProducts;
import com.springmvc.model.User;
import com.springmvc.service.ProductService;
import com.springmvc.service.PurchaseTableProductsService;
import com.springmvc.service.UserService;

@Controller
public class UserController {

	@Autowired
	private UserService userService;

	@Autowired
	private ProductService productService;

	@Autowired
	private PurchaseTableProductsService purchaseTableProductsService;

	@RequestMapping("/SignInUser")
	public String signInUser() {
		return "userLogin";
	}

	@RequestMapping(path="/processCreateUser" , method=RequestMethod.POST)
	public String processCreateUser(@ModelAttribute User user) {

		user.setUsername(user.getFname()+user.getLname()+user.getId());
		user.setPassword(user.getId()+user.getLname()+user.getFname());
		System.out.println(user);
		this.userService.saveCreatedUser(user);
		return "UserCreatedSuccess";
	}

	@RequestMapping(path="/Login",method=RequestMethod.POST)
	public String userLoginProcess(@ModelAttribute User user,Model model) {

		System.out.println(user);
		List<User> listOfUsers=this.userService.presentAllUsers();
		System.out.println(listOfUsers);
		for(User usern:listOfUsers) {
			/*	System.out.println(usern.getUsername());
			System.out.println(user.getUsername());
			System.out.println(usern.getPassword());
			System.out.println(user.getPassword());*/
			if(usern.getUsername().equals(user.getUsername()) &&
usern.getPassword().equals(user.getPassword())){
				System.out.println("Userl login successful");
				List<Product>
listOfProducts=this.productService.getListOfAllProducts();
				System.out.println(listOfProducts);
				PurchaseTableProducts purTProducts;
				for(Product l:listOfProducts) {
					purTProducts=new PurchaseTableProducts();
					purTProducts.setUsername(user.getUsername());
					purTProducts.setProductname(l.getProductname());
					purTProducts.setCost(l.getCost());
					purTProducts.setCategory(l.getCategory());

	purTProducts.setMyPrimaryKey(user.getUsername()+"_"+l.getProductname());
```

```
        this.purchaseTableProductsService.saveallProductsinPurchaseTable(purTProducts);
                            }
                            model.addAttribute("listOfAllProducts", listOfProducts);
                            model.addAttribute("User", user);
                            return "Home1";
                    }

                }
                return "invalidCredentialNew";
        }
        //username=4_User_New, password=User_New_4
}
```

3 Writing a program in jsp page to display options available for the Admin (LinkPage.jsp and adminpasswordChange.jsp.)

LinkPage.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Admin Link Page</title>
</head>
<body>
     1.<a href="createUser">Create User</a><br>
     2.<a href="getAllUsers">Get All users</a><br>
     3.<a href="purchaseHistory">Get all purchase History</a><br>
     4.<a href="purchaseReport">Purchase report filtered by date and
category</a><br>
</body>
</html>
```

adminpasswordChange.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Change The Admin Password</title>
<link href="<c:url value="/css/login1.css" />" rel="stylesheet"></link>
```

```
</head>
<body>
<div class="center">
            <div class="center1">
                  <h1>Change Admin Password</h1>
                  Please enter valid username.
                  Then enter password
                  <form action="changeThePasswordCheck" method="POST">

                        <label>Username : </label> <input type="text"
                              placeholder="Enter Username" name="username"
required> <br>
                        <br><br><label>Password : </label> <input
type="password"
                              placeholder="Enter Password" name="password"
required> <br>
                        <br>
                        <br> <label>Reenter Password : </label> <input
type="password"
                              placeholder="Confirm Password"
name="confirmpassword" required> <br>
                        <br>

                        <br>
                  <div class="center1">
                        <button type="submit">Change the Password</button>
                  </div>
                  <br />
            </form>
            </div>
      </div>
</body>
</html>
```

7 Writing a program to for entities required for database.

1.Admin.java

```
package com.springmvc.model;


import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;


@Entity

public class Admin {


      @Id
```

```
	@GeneratedValue(strategy=GenerationType.AUTO)
	public int id;
	public String username,password;
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getUsername() {
		return username;
	}
	public void setUsername(String username) {
		this.username = username;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}

	@Override
	public String toString() {
		return "Admin [id=" + id + ", username=" + username + ", password=" +
password + "]";
	}
}
```

2.Order1.java

```
package com.springmvc.model;
```

```
import java.time.LocalDate;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Order1 {
   @Id
   @GeneratedValue(strategy=GenerationType.AUTO)
   public int id;

        public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }

        public String username;
        public String productname;
        public int cost;
        public LocalDate date1;
        public LocalDate getDate1() {
                return date1;
        }
        public void setDate1(LocalDate date1) {
                this.date1 = date1;
        }
```

```
	public String category;
	public String getUsername() {
		return username;
	}
	public void setUsername(String username) {
		this.username = username;
	}
	public String getProductname() {
		return productname;
	}
	public void setProductname(String productname) {
		this.productname = productname;
	}
	public int getCost() {
		return cost;
	}
	public void setCost(int i) {
		this.cost = i;
	}


	public String getCategory() {
		return category;
	}
	public void setCategory(String category) {
		this.category = category;
	}
	@Override
	public String toString() {
		return "Order [id=" + id + ", username=" + username + ", productname=" +
productname + ", cost=" + cost
```

```
                                        + ", date1=" + date1 + ", category=" + category + "]";
          }
}
```

3.Product.java

```
package com.springmvc.model;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Product {

          @Id
          @GeneratedValue(strategy=GenerationType.AUTO)
          String productname;

          int cost;

          String category;

          public String getProductname() {
                    return productname;
          }

          public void setProductname(String productname) {
                    this.productname = productname;
          }
```

```
	public int getCost() {
		return cost;
	}

	public void setCost(int cost) {
		this.cost = cost;
	}

	public String getCategory() {
		return category;
	}

	public void setCategory(String category) {
		this.category = category;
	}

	@Override
	public String toString() {
		return "Product [productname=" + productname + ", cost=" + cost + ", 
category=" + category + "]";
	}
}
```

4.PurchaseTableProducts.java

```
package com.springmvc.model;

import java.util.List;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class PurchaseTableProducts {



        private String username;



        private String productname;



        private int cost;


        private String category;


        @Id
        private String myPrimaryKey;


        public String getMyPrimaryKey() {
                return myPrimaryKey;
        }


        public void setMyPrimaryKey(String myPrimaryKey) {
                this.myPrimaryKey = myPrimaryKey;
        }


        public String getUsername() {
                return username;
}

public void setUsername(String username) {
	this.username = username;
}

public String getProductname() {
	return productname;
}

public void setProductname(String productname) {
	this.productname = productname;
}

public int getCost() {
	return cost;
}

public void setCost(int cost) {
	this.cost = cost;
}

public String getCategory() {
	return category;
}

public void setCategory(String category) {
	this.category = category;
}
```

```
	@Override
	public String toString() {
		return "PurchaseTableProducts [username=" + username + ", productname="
+ productname + ", cost=" + cost
				+ ", category=" + category + "]";
	}



}
```

5.User.java

```
package com.springmvc.model;


import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;


@Entity
public class User {
   @Id
	@GeneratedValue(strategy=GenerationType.AUTO)
   public int Id;



   public String fname;



   public String lname;



   public String username;
```

```
    public String password;




public int getId() {
                    return Id;
            }
            public void setId(int id) {
                    Id = id;
            }
public String getFname() {
            return fname;
}
public void setFname(String fname) {
            this.fname = fname;
}
public String getLname() {
            return lname;
}
public void setLname(String lname) {
            this.lname = lname;
}

public String getUsername() {
            return username;
}
public void setUsername(String username) {
		this.username = username;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}


	@Override
	public String toString() {
		return "User [Id=" + Id + ", fname=" + fname + ", lname=" + lname + ", username="
+ username + ", password="
				+ password + "]";
	}
}
```

1.AdminService.java

```
package com.springmvc.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.springmvc.dao.AdminDao;
import com.springmvc.model.Admin;

@Service
public class AdminService {
	public AdminService(AdminDao adminDao) {
		super();
		this.adminDao = adminDao;
	}



	@Autowired
	private AdminDao adminDao;



	public void enterAdminCredential(Admin admin) {
	    this.adminDao.insertAdmin(admin);
	}
}
```

2.OrderService.java

```
package com.springmvc.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;



import com.springmvc.dao.OrderDao;

import com.springmvc.model.Order1;

@Service
public class OrderService {
	@Autowired
	private OrderDao orderDao;

	public OrderService(OrderDao orderDao) {
		super();
		this.orderDao = orderDao;
	}



	public void enterOrderDetails(Order1 order) {
		this.orderDao.insertOrder(order);
	}

	public List<Order1> getAllOrdersForAllUsers() {
		return this.orderDao.getAllOrderRecords();
	}



}
```

3.ProductService.java

```
package com.springmvc.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.springmvc.dao.ProductDao;
import com.springmvc.model.Product;
```

```
@Service
public class ProductService {



	@Autowired
	private ProductDao productDao;

	public ProductService(ProductDao productDao) {
		super();
		this.productDao = productDao;
	}

	public List<Product> getListOfAllProducts(){

		return this.productDao.getAllProducts();

	}



}
```

4. PurchaseTableProductsService.java

```
package com.springmvc.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;


import com.springmvc.dao.PurchaseTableProductsDao;
```

```
}

import com.springmvc.model.PurchaseTableProducts;

public class PurchaseTableProductsService {

	@Autowired
	private PurchaseTableProductsDao purchaseTableProductsDao;



	public PurchaseTableProductsService(PurchaseTableProductsDao
purchaseTableProductDao) {
		super();
		this.purchaseTableProductsDao = purchaseTableProductDao;
	}



	public void saveallProductsinPurchaseTable(PurchaseTableProducts
purchaseTableProducts) {
		this.purchaseTableProductsDao.saveDataInTable(purchaseTableProducts);
	}

	public List<PurchaseTableProducts> getAllProducts(){
		return this.purchaseTableProductsDao.getAllproducts();
	}

	public void deleteAllPurchasedProducts(String myPrimaryKey) {
		this.purchaseTableProductsDao.deleteAllpurchaseProduct(myPrimaryKey);
	}

}
```

5.UserService.java

```
package com.springmvc.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import com.springmvc.dao.UserDao;
import com.springmvc.model.User;

public class UserService {

	@Autowired
    private UserDao userDao;

	public void saveCreatedUser(User user) {
		this.userDao.insertUser(user);
	}

	public UserService(UserDao userDao) {
		super();
		this.userDao = userDao;
	}

	public List<User> presentAllUsers() {
		List<User> listUsers=this.userDao.getAllUsers();
		return listUsers;
	}
}
```

8.Pushing the code to GitHub repository

1. Open your command prompt and navigate to the folder where you have created your files.

2. Initialize repository using the following command: **git init**

3. Add all the files to your git repository using the following command: **git add .**

4. Commit the changes using the following command: **git commit . -m**

5. Push the files to the folder you initially created using the following command:

   git push -u origin master

Unique Selling Points of the Application

1. Application is multiuser and single admin application. So only admin and users can access it with right credentials.

2. Application is robust for admin flow and user flow . The admin need to login first then only admin can access its internal information.

3. Admin needs to create user first. Share the credential with user and then user can access his or her account.

4. Unauthorized user can not access account or buy the products.

5. Its dynamic nature to generate the history of user transaction and all report with date and category makes it unique.

6. User Interaction is smooth as all navigational options are given.

7. Allowing admin to add any number of users to the list

8. Allowing admin retrieval of time when last date when application was accessed by user

Conclusion

Application enhancement is possible at following point:

1. Application can improved with user interface.

2. Implementation standard can be improved by investing more time in developing same application