**Project Requirements Specification**

**Course:** CS514 Software Engineering

**Project Name:** Online Food Ordering Web Site

**Name**

Tiffany Valens

**Professor:** John Ma

**Last updated:** <June 14, 2013>

**Deliverable Number:** <#1>

**Document Revision Number:** <v0.1>

**<Online Food Ordering Website>**

|  |  |
| --- | --- |
| **Table of Contents** | |
| **Project Component** | **Page#** |
| 1. Introduction | **3** |
| 1. Business Case and Project Vision | **3** |
| 1. Clients | **3** |
| 1. System Actors | **4** |
| 1. Scope of Work | **5** |
| 1. Functional Requirements | **6** |
| 1. Non-Functional Requirements | **7** |
| 1. Constraints | **8** |
| 1. Assumptions and Dependencies | **9** |
| 1. Application Testing | **9** |
| 1. Data Entities and Relationships | **10** |
| 1. Architecture Diagram | **15** |
| 1. Database Design | **16** |
| 1. Use Case Reports | **17** |

1. **Introduction**

As today’s people is relying on efficiency, convenient processes, products, and services in order to save time, effort, energy, and increase ease in accessibility, the internet and its boom have provided means to cater to people needs without having to go to the physical stores to buy products or services, this includes food service. Given a variety of options to choose from participated vendors, potential customers just need to go online to complete their purchases. The prevailing “brick and mortar” restaurants which have physical existence are not required for business owners. The Online Food Ordering (OFO) Web Site operates from cyberspace.

**SYSTEM CONCEPT**

1. **Business Case & Project Vision**

The Online Food Ordering (OFO) Web Site facilitates food vendors to set up online restaurants. Customers can browse through the restaurants, and a system administrator admit and reject new restaurants and maintain lists of restaurant categories. Our OFO Web Site is designed to manage items in the restaurant and also to help customers purchase them without having to visit the restaurants physically, using internet as the sole method for selling food to its customers.

This website allows the customer to easily explore any choices of food from local and regional cuisines in the U.S. to unique and exotic culinary dishes sold by vendors with backgrounds from all around the world, with occasional experimental and fusion food creations provided by the sellers. The food fair is on-going every day with no time limitations and the customers can have the order delivered to their homes with paying delivery fees to the vendors, or the food can be picked up at some locations that the vendors provide if the customers wishes to do so. Depending on the restaurants, some shops open 24 hours. The customer has a complete control of his/her shopping experience and provided with pictures of items sold in the menu of each vendor.

(obj)This project looks for bridging the gap between the seller and the customer. As a multiple merchant venue, a good amount of flexibility is offered in the design process so that every online restaurant owner can choose to use the basic templates or own design templates for his/her own window which customer can visit to browse and subsequently buy the products from. The delivery of product is managed directly by restaurant owners.

1. **Clients**

Most restaurant owners have a physical foundation that often causes quite a number of problems. Multiple store fronts are expensive to maintain as store prices are forced to rise. By using our product, food vendors are at an advantage because their costs are significantly lower than their competitors, allowing restaurant owners to sell the same food at a lower price. This OFO Web Site also acts as a platform for small business owners to thrive and gain a competitive edge in the restaurant business since the overall costs of store maintenance are relatively low. To preserve and assure the quality of product sold by vendors, the customer can rate the items purchased from each vendor and provide feedback for the food and services, as part of the community response of eating experience. As internet is now used rapidly and people are accustomed to using it, they regard ordering products and services online as a time-saving and cost-saving experience, which is the very essence of our online food court system.

1. **System Actors**

The OFO Web Site’s clients are business owners and customers/guests. The actors who interact with the system are:

* **Web Administrator**: The Web Administrator is the super user and has complete control over all the activities that can be performed. The application informs the administrator of all restaurant creation requests, and the administrator can then approve or reject them. The administrator also manages the list of available product categories. The administrator can also view and delete entries in the guestbook.
* **Restaurant Owner:** Any user can create a restaurant and submit the request through the application. Once the Web Administrator approves it, the requester is notified, and from then on is assigned the role of Restaurant Owner. The Restaurant Owner is responsible for setting up his/her online restaurant and maintaining it. The job includes managing the sub-categories of the items in the restaurant, listing or deleting items from his store, viewing reports that give details of the sales and orders specific to his restaurant. The Restaurant Owner can also decide to close shop and remove it from the food court.
* **OFO Customer/Guests:** An OFO Customer can look through the restaurants and choose products to place in a virtual shopping cart. The shopping cart details can be viewed and items can be added or removed from the cart. To place an order, the customer is prompted to login. Also, the customers can modify personal information (such as phone number and delivery address) stored by the application. The customer can also view the status of any previous orders.
* **Employees:** This a customable option for restaurant owners who wish to delegate administration of his/her enterprise to employees who work in the accounts department under an Accounts manager to perform the accounting activities of the restaurant, sales department under a Sales manager looks after the sale of food, and the food delivery department under a Food delivery manager to overlook distributing activities if deemed necessary.

1. **Scope of Work**

Initial functional requirements are:

* Free and secure registration for new customer and profile management facilities for existing Customers
* Searching through the e-food court to see the items available for purchase for each category of food like Burger, Sub, Pizza/Italian, Sushi/Japanese, Thai, Chinese, Mediterranean, etc.
* Fast, easy, quick, and reliable searching mechanisms for finding particular types of food and drinks.
* Building a shopping cart so that customers can add ‘n’ number of items to the cart and checkout with the entire shopping carts.
* Email updates to current customers of OFO about new food items from restaurants where they have past purchases and recommendations of similar food they browse.
* Displaying ‘Most Purchased Items” and “Highest Rated Items” in each restaurant.
* Strategic data and graphs for Administrators and Restaurant owners about items that are popular in the menu and geographical regions with most orders.
* Restaurant employees carry out their internal responsibilities that involve processing orders, updating order’s status and answering client’s queries online or by the phone, assuring home delivery, getting customer’s feedback.
* Feedback mechanism, so that customers can give feedback for the product or service which they have acquired, this includes rating of individual products by relevant customers, also feedback for the performance of particular vendors, and the entire OFO website as well.
* Electronic payment for online transactions using all popular credit cards, debit cards, PayPal, money orders, and other relevant payment options, as available from time to time.

Initial non-functional requirements are:

* Secure access of confidential data using SSL.
* 24 hours a day, 7 days a week system availability.
* Better component design to achieve optimal performance at peak time.
* Space for advertisement where it is effectively noticed by customers and as a source of revenue.
* Each customer has a personalized web page based on his/her recent purchases, as part of a unique restaurant front concept in hopes of drawing in as many return customers as possible.

1. **Functional Requirements**

## Programming languages:

* JAVA EE: Java Enterprise Edition 6 is Oracle’s enterprise Java computing platform for extending the Java Platform, Standard Edition (Java SE), providing an API for object- relational mapping, distributed and multi-tier architectures, and web services, based largely on modular software components running on an application server.
* HTML, XML: Hyper Text Markup Language and Extensible markup Language are the main markup languages for creating web pages, displaying and storing information in a web browser. It provides a means to create the structure of text-based information in a document and to allow images and objects to be embedded and can be used to create interactive forms.
* JavaScript: A prototyped-based scripting language that is used to make web pages interactive, create dynamic web content and user interface.

Tools & Development Environment:

* ECLIPSE J2EE: Eclipse is a toolkit which is designed for assembling complex projects, providing fully dynamic web application utilizing Enterprise JavaBeans (EJB). This comprise of EJB tools, CMP, data mapping tools, and a universal test client that is capable of testing EJBs.
* Apache Tomcat 7.0.18 Server: Apache Tomcat is an open source web server and Servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run in.

## Product perspective

OFO is geared towards vendors who want to reach out to the maximum cross-section of customer and common people who can be potential customer. This project envisages OFO as the tool that simplifies the process of purchasing food from restaurants, as the ultimate food place for customers to browse and order food online, and as the platform for catering and restaurant businesses to attract more customers and go big without having the necessity to own and maintain physical stores or dining locations. OFO should be user-friendly, ‘quick to learn’ and reliable software for the above purpose. OFO is intended to be a stand-alone product and should not depend on the availability of other software. It should run on both UNIX and Windows based platform.

User Interface Design

Siding with standard values of a good user interface whether it is designed for a WebApp, or a traditional software application, this website is designed to exhibit the following characteristics:

* Easy to use.
* Easy to maintain.
* Easy to learn.
* Easy to navigate.
* Intuitive.
* Consistent.
* Efficient.
* Error-free/ less bugs.
* Functional.

Aiming to provide the end-user with a satisfying and rewarding experience, this OFO web site follows all these principle of effective user interface design. Like an effective interface, OFO is visually apparent and accommodating, instilling in its users a sense of control. Users quickly see the breadth of their options, grasp how to achieve their goals, and do their work. It does not concern the user with the inner workings of the system and the users have the full option to undo activities at any time to remove item from the shopping cart. Like effective applications and services, OFO performs a maximum of work, while requiring a minimum of information from the users.

1. **Non-Functional Requirements**

A number of attributes of software can serve as requirements; the following items provide a partial list. Known as non-functional requirements or quality attributes, these are characteristics the system must possess, but that might diffuse through the design.

* 1. ***Availability***

The system is available at all times assuring the user can access it using a web browser, only restrained by the down time of the server on which the system runs. In case of a hardware failure or database corruption, a replacement page will be shown and backups of the database should be retrieved from the server and saved by the administrator. Then the service is restarted.

* 1. ***Reliability***

The reliability of the separate components determines the reliability of the overall program. The backup of the database comprises the main pillar of reliability of the system is continuously preserved and updated to reflect current changes. The implementation is J2EE oriented, thus the system will be operating inside a container. The overall stability of the system depends on the stability of container and its underlying operating system.

* 1. ***Maintainability***

A commercial database is used for conserving the database and the application server oversees the site. In case of a failure, a re-initialization of the program is performed. The software design is being done with modularity in mind so that maintainability can be done efficiently.

* 1. ***Portability***

Due to the application is J2EE based, this website is compatible with all systems which have a native Java implementation. The end-user part is fully portable and any system using any web browser should be able to utilize the features of this application, including any hardware platform available in the present system and in the future.

* 1. ***Security***
* The user's IP is logged.
* Passwords are saved encrypted in the database to safeguard the user's privacy.
* Sensitive data encrypted before being sent over insecure connections like the internet.
* Certain functions are assigned to certain modules only.
* Data integrity are checked for critical variables.
  1. ***Portability***

Due to the application is J2EE based, this website is compatible with all systems which have a native Java implementation. The end-user part is fully portable and any system using any web browser should be able to utilize the features of this application, including any hardware platform available in the present system and in the future.

* 1. ***Security***
* The user's IP is logged.
* Passwords are saved encrypted in the database to safeguard the user's privacy.
* Sensitive data encrypted before being sent over insecure connections like the internet.
* Certain functions are assigned to certain modules only.
* Data integrity are checked for critical variables.

1. **Constraints**

* Limited to HTTP/HTTPS.
* No supportability of back up so availability will get affected.
* No multilingual support

1. **Assumptions and Dependencies**
   * The details related to the product, customer, payment and service transaction provided manually.
   * Administrator has existed in the system.
   * Roles and tasks are predefined.
2. **Application Testing**

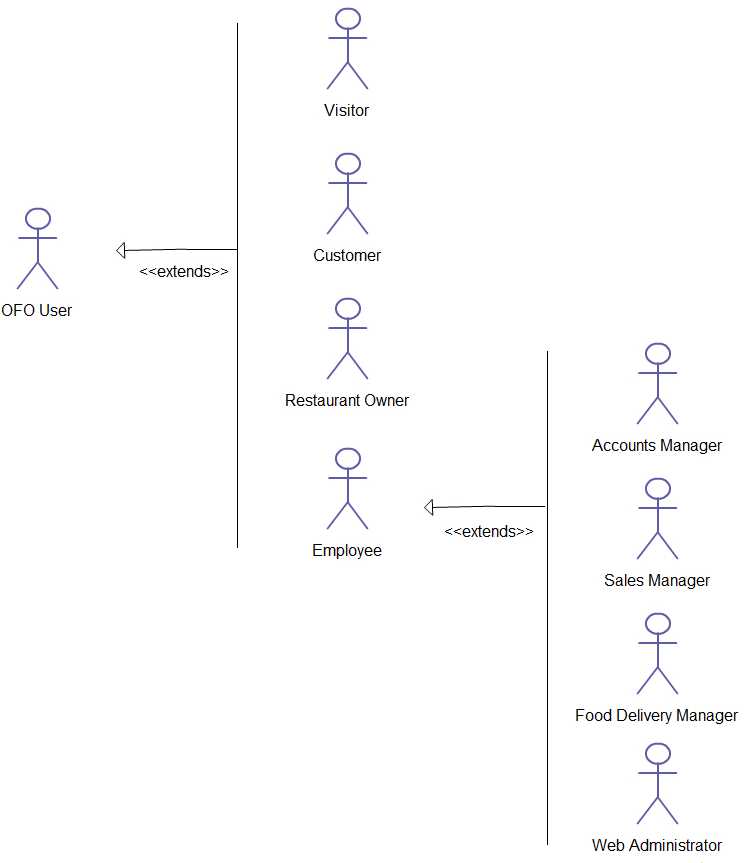
Testing Web application (WebApp) is very important process in ensuring that the end users stay true in utilizing OFO website as their favorite app and prevent them to go elsewhere for the content and function they need. The WebApp testing comprises of a collection of related activities with its main goal to eliminate software problems by uncovering errors in WebApp content, function, usability, navigability, performance, capacity, and security. To accomplish this, a testing strategy that encompasses both reviews and executable testing is applied throughout the Web engineering process.

The testing process begins by focusing on user-visible aspects of the WebApp and proceeds to test that exercise technology and infrastructure. In some instances a WebApp test plan is produced. In every instance, a suite of test cases is developed for every testing step and an archive of test results is maintained for future use.

Quality is incorporated into a web application as a consequence of good design. In this website as well, the following quality dimensions have always been the cap stones in the development process:

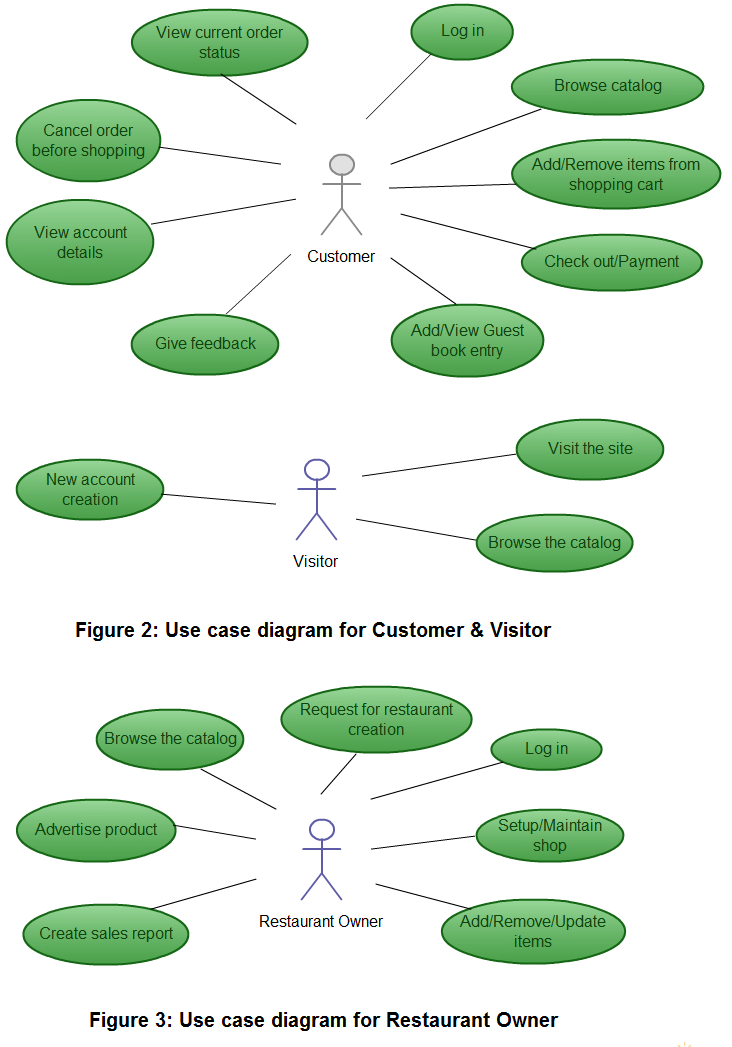
* ***Content*** is evaluated at both the syntactic and semantic level.
* ***Function*** is tested to uncover errors that indicate lack of conformance to customer requirements.
* ***Structur****e* is assessed to ensure that it properly delivers content and function, that it is extensible, and that it can be supported as new content or functionality is added.
* ***Usability*** is tested to ensure that each category of user is supported by the interface.
* ***Navigability*** is tested to ensure that all navigational syntax and semantics are exercised to uncover any navigational errors.
* ***Performance*** is tested under a variety of operating conditions to ensure that the system is responsive to user interaction and operates without unacceptable operational degradation in all situations.
* ***Compatibility*** is tested by executing the web application in a variety of different hosts on both the client and server sides.
* ***Interoperability*** is tested to ensure that the WebApp properly interfaces with other applications and/or databases.
* ***Securit****y* is tested by assessing potential vulnerabilities and attempting to exploit each.

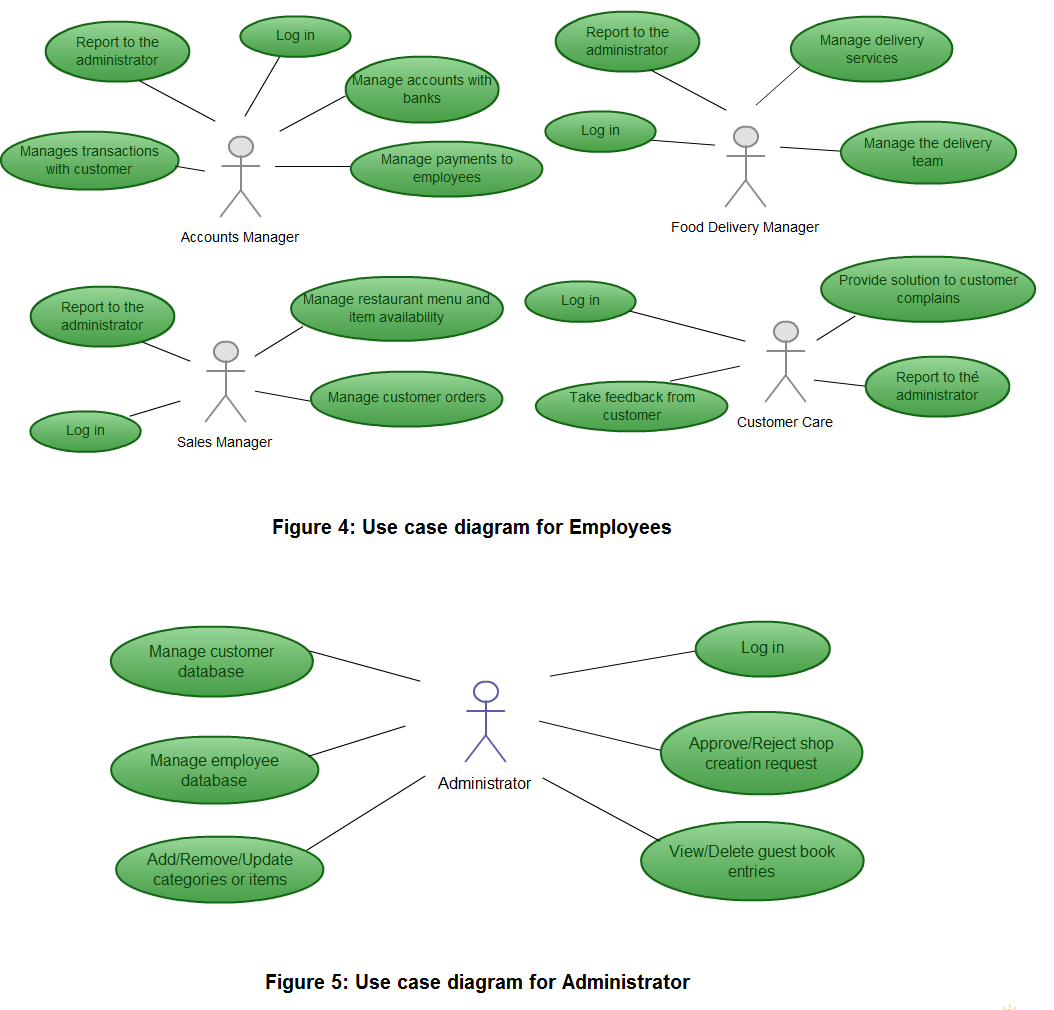
**11. Data Entities and Relationships**



**Figure 1. User hierarchy**

.





The overall description of the system, as represented in the above use-case diagrams:

**Administrator:**

* **Database Management:** Control the database and keep track of all records of customers and employee details.
* **Contact and Giving Permission to Vendors:** Communicate with vendors and give permission to sell their product under the site after testing the quality of food and drinks.
* **View all details:** View the details of all employees and control the whole site.
* **Advertising the Site:** Responsible for regular advertising and marketing of the site.

**Customers:**

* **Login:** Customers must have a valid login ID to enter into the site.
* **View and edit personal account:** Can view/edit his personal details, payment details, anddetails about services provided.
* **Choosing and comparing products:** Can view all available products and can compare them and make a choice for purchasing products.
* **Use a shopping cart:** Can choose items from the restaurant menu, add them into the cart, and discard no longer wanted items.
* **Buying:** Can purchase any product through valid credit card, debit card, or PayPal.
* **Giving Feedback to Customer Care:** Can provide feedback to the 24X7 Customer Care Service center about their impression for the site and services.
* **Use the Guest book:** Customers can read the guestbook of each restaurant and also leave a guestbook entry.

**Visitors:**

* **Visiting the Site:** Everyone can visit the site without registration.
* **Register:** New users can sign up by creating new ID.

**Restaurant Owner:**

* **Acquiring Permission from Administrator:** Vendors must get permission from the Administrator for selling their products under the site. Administrator will test quality of each food and beverage according to its market prices to authorize vendor for selling purpose.
* **Consulting with Administrator:** Can consult with the Administrator regarding product’s quality and advertisements.
* **Advertising Vendor’s Own Products:** Responsible for making advertisements of his products, as the site is not responsible for any kind of advertisements about products.

**Sales Manager:**

* **View customer details:** View the personal details of the customer.
* **Managing customer orders:** Responsible for properly checking the selected product according to the customer’s choice and transferring the correct customer order to the restaurant owner/kitchen for processing. Also inform updates to the restaurant owner/kitchen when order cancellation occurs or changes made to the order during acceptable timeframe.
* **Contacting with Customers regarding sales:** Provide order confirmation from the restaurant and manage all phone-in and online orders. Give updates to Customers when their order is ready for delivery.
* **Managing the restaurant menu and item availability:** Change, adjust the menu per restaurant owner’s request and keep track of each item’s availability for selling purpose.
* **Contacting with Administrator:** Informing administrator any changes to the menu and items availability.

**Food Delivery Manager:**

* **Managing Delivery Services:** Arrange restaurant orders to be delivered to the customer addresses by assigning delivery tasks to personnel according to the areas they are delivering.
* **Supervising the Delivery Team:** Monitor the performance of delivery personnel, ensuring that they deliver customer orders to the correct addresses within the estimated delivery time.
* **Consulting with Administrator:** Reporting any changes related to delivery services: areas covered, delivery time and fees to the Administrator for updating the database.

**Accounts Manager:**

* **Regulating Payments:** Keep track of all the payment transactions made by the customers and update the payment information.
* **Consulting with Banks:** Responsible for contacting the banks for the validation of the account number provided by the customer while purchasing and make the transaction from the given account.
* **Consulting with Administrator:** Consult with the Administrator about the payment details of the customers for the updating of the database.

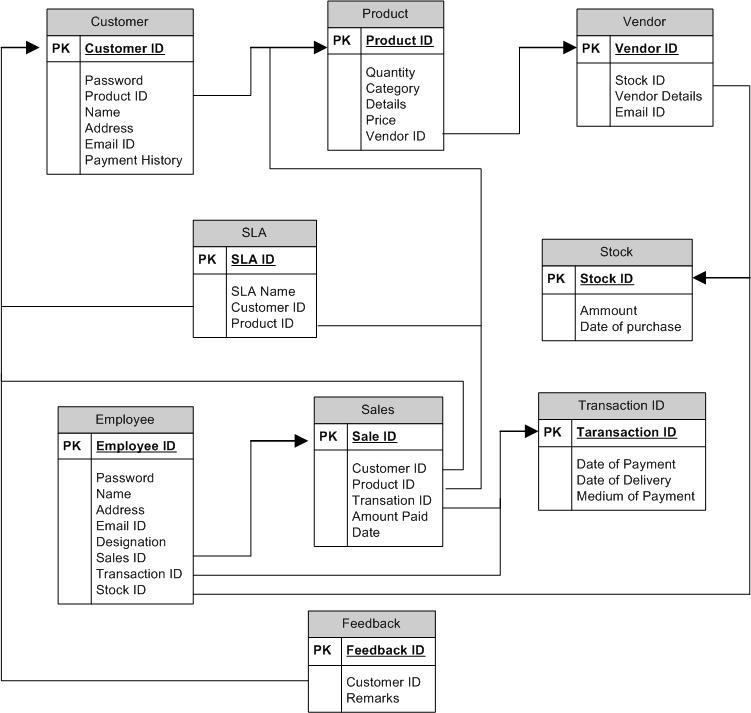
**Customer Care:**

* **Accepting Feedback from the Customers:** Responsible for receiving complaints, queries and feedback from the customers.
* **Providing Solutions to Customers:** Provide feasible solutions to the customers on their complaints and queries.

**12. Architecture Diagram**

## arch

**13. Database Design**



# 14. Use Case Reports

* Administrators:

**Database Management:** Control the database and keep track of all records of customers and employee details.

* **Preconditions:** Administrator log in is completed.
* **Normal flow of events:**

1) Usual check of the database by the Administrator.

2) Updating the database (if required).

* **Alternate flow of events:** None.
* **Post Condition:** Continuously updated database.

**Contact and Giving Permission to Vendors:** Communicate with restaurant owners and give permission to sell their product under the site after testing the product’s quality.

* **Preconditions:** 1) Log in is completed.

2) Vendor contacts with Administrator.

* **Normal flow of events:** Negotiation is successful.
* **Alternate flow of events:** Negotiation is failed.
* **Post Condition:** possibilities of new product items

**Contacting Business Partners:** Responsible for contacting with Business Partners who will sponsor the site and help in conducting the business process.

* **Preconditions:** 1) Log in is completed.

2) Business Partner contacts with Administrator.

* **Normal flow of events:** Negotiation is successful.
* **Alternate flow of events:** Negotiation is failed.
* **Post Condition:** possibilities of new sponsors and raise in investments.

**Advertising the Site:** Responsible for making advertisements for the site.

* **Preconditions:** Log in is completed.
* **Normal flow of events:** 1) Contacting different media.

2) Making advertisements for the site.

* **Alternate flow of events:** None.
* **Post Condition:** Site is promoted.

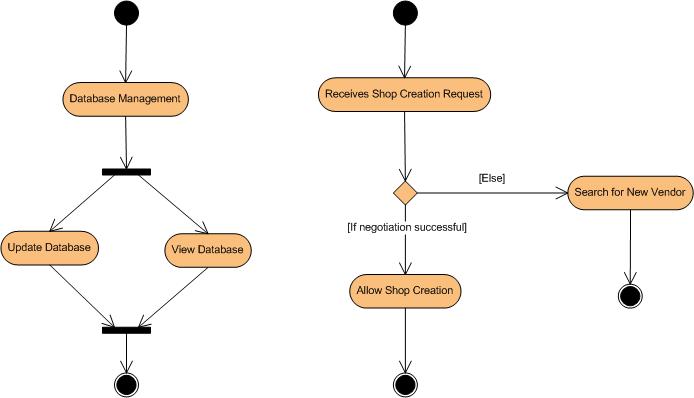
**View all details:** View the details of all employees and control the whole site.

* **Preconditions:** Log in is completed.
* **Normal flow of events:**

1) Administrator views the details of all employees.

2) Controls the whole site.

* **Alternate flow of events:** None.
* **Post Condition:** Everything is checked and under control.



* Customers:
* **Preconditions:** Customer must have a valid user ID.
* **Normal flow of events:**

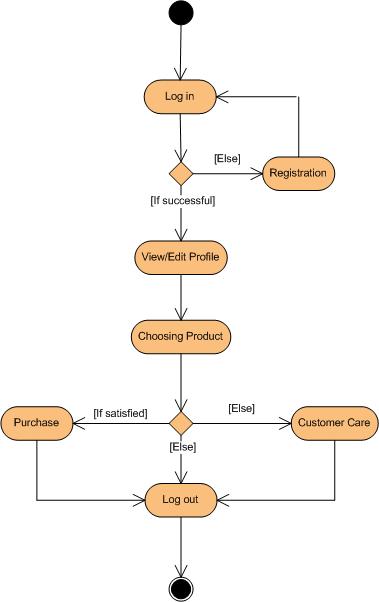
1. Log in.
2. View and edit Own Details
3. Choosing and comparing products
4. Purchasing
5. Logout

* **Alternate flow of events:**

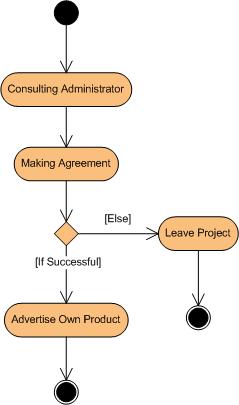
1) New customer registration

2) Complaining to Customer Care

* **Post Condition:** Customer goals accomplished.

****

* Visitors:
* **Preconditions:** Log in is completed.
* **Normal flow of events:** Visiting the Site.
* **Alternate flow of events:** None.
* **Post Condition:** Proper separation between customers and window-shoppers.
* Vendor:
* **Preconditions:** Can consult with the Administrator regarding product’s quality and advertisements.
* **Normal flow of events:** Can consult with the Administrator regarding product’s quality and advertisements.
* **Alternate flow of events:** Can withdraw from the project.
* **Post Condition**: Various attractive items for customers.

****

* Sales Manager:

Sales Manager can view customer details and responsible for managing sales to customers, viewing product stocks and contacting with the administrator.

* **View Customer Details:** View personal details of the customers.
* **Managing Sales to Customers:** Responsible for properly checking the selected product according to the customer’s choice and transferring the correct customer order to the restaurant owner/kitchen for processing.
* **View Restaurant Menu:** Adjust the menu per restaurant owner’s request and keep track of each item’s availability for selling purpose.
* **Contacting with Administrator:** Informing administrator any changes to the menu and items availability.

**Name of the use case:** View customer details.

**Description:** View the personal details of the selected customer.

**Precondition:** Sales manager is already logged in.

**Normal flow of events:**

Select customer.

The details of customer viewed.

**Alternate flow of events:** None

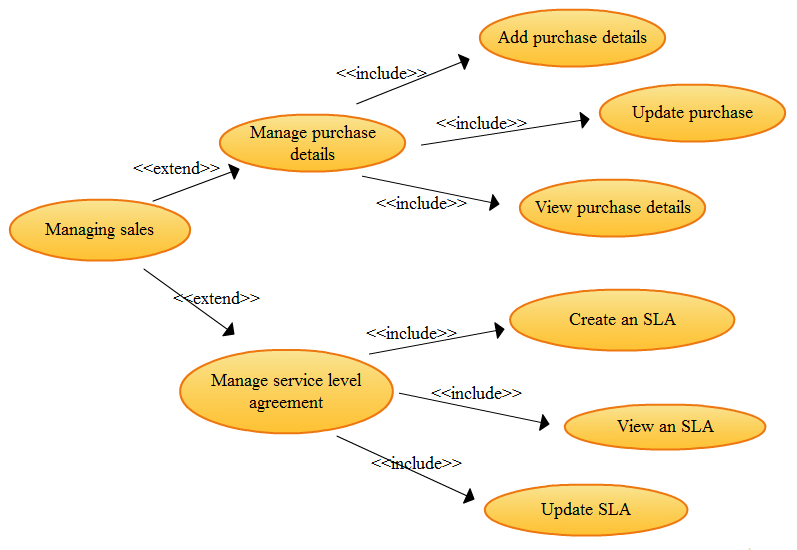
**Post condition:** None.

Ask for customer

Customer details

Select customer

**Managing Sales to Customers:**



**Name of the use case:** Add/update and view purchase details.

**Description:** Store the details of the items sold, customer ID, and any changes in customer order details can be made and view purchase details.

**Precondition:** Sales Manager is already logged in. The customer is registered and the products are already listed.

**Normal flow of events:**

Select a customer.

Select an item from the menu.

Enter/update purchase details.

Save new data.

**Alternate flow of events:**

If the customer is not registered, ask for registration.

If the item is not available, send error message.

**Post condition:** Sale ID is generated.

Available

Select customer

Select product

Ask for registration

Send error message

Registered

Validation

Not registered

Availability

Not available

Sale product

Add/update data

View purchase details

**Name of the use case:** Create/update a service level agreement.

**Description:** Store the details of the services provided to a customer, duration of the services and details of the terms and conditions

**Precondition:** Sales manager is already logged in. The product and the services to be provided are already present.

**Normal flow of events:**

Select product.

Select services

Enter details of the service level agreements.

Add / update the data.

**Alternate flow of events:**

If the item is not present, send error message.

**Post condition:** SLA is created / updated.

Ask for SLA

Details

Select product

Send error message

Availability

Ask for services

Not available

Available

Select service

Add / update data

**Name of the use case:** View service level agreement.

**Description:** To see the details of the agreement.

**Precondition:** Sales manager is logged in. The product and the services to be provided are already present.

**Normal flow of events:**

Select restaurant product.

Select date.

The details of the SLA are shown to the sales manager.

**Alternate flow of events:**

If the item is not present, send error message.

**Post condition:** None.

View SLA

Details

Select product

Send error message

Availability

Ask for services

Not available

Available

Select service

* Delivery Manager:

Delivery Manager is responsible for arranging delivery services to the customers, receiving items from restaurant, managing the delivery personnel, and consulting with the administrator.

* **Managing Delivery Services:** Arrange restaurant orders to be delivered to the customer addresses by assigning delivery tasks to personnel according to the areas they are delivering.
* **Supervising the Delivery Team:** Monitor the performance of delivery personnel, ensuring that they deliver customer orders to the correct addresses within the estimated delivery time.
* **Consulting with Administrator:** Reporting changes related to delivery services: areas covered, delivery time and fees to the Administrator for updating the database.

**Name of the use case:** Consulting with the administrator about delivery services to customers.

**Description:** Contacting the administrator about vendor request in modifying delivery services.

**Precondition:** Purchase Manager is already logged in.

**Normal flow of events:**

Report recent changes to the administrator.

Implement changes in the database.

Delivery services are updated.

**Alternate flow of events:** None.

**Post condition:** None.

Report changes to the administrator.

Modify the database accordingly

Accepted

Delivery services are updated

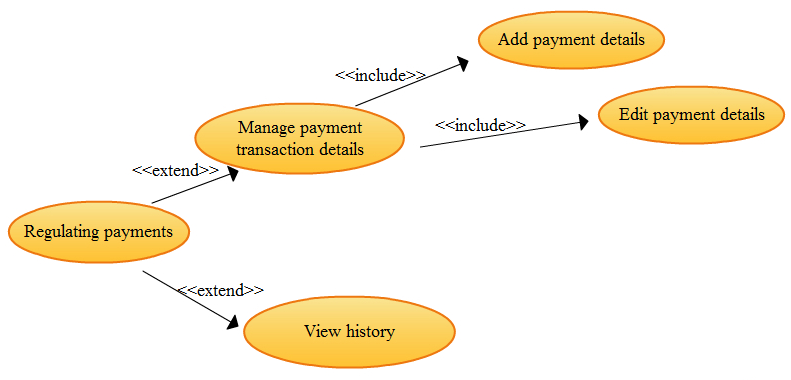
Received

* Accounts Manager:

Accounts Manager is responsible for receiving customer payments, managing customer payment details and consulting with the administrator.

* **Regulating Payments:** Keep track of all the payment transactions made by the customers and update the payment information.
* **Consulting with Banks:** Responsible for contacting the banks for the validation of the account number provided by the customer while purchasing and make the transaction from the given account.
* **Consulting with Administrator:** Communicate with the Administrator about any payment transaction problems.

**Regulating Payments**



**Name of the use case**: Add / edit payment transaction details

**Description**: All the payment transaction details are entered or edited.

**Precondition**: Accounts manager has logged in.

**Normal flow of events:**

Select the customer.

Select the product.

Select transaction ID.

Enter / edit the details of payment.

Save the payment details.

**Alternate flow of event:** None.

**Post condition**: None.

Select customer

Select product

Select

Transaction ID

Enter / edit

Payment details

Save

Payment details

**Name of the use case:** View history.

**Description:** View the payment details of the selected customer.

**Precondition:** Accounts manager is already logged in.

**Normal flow of events:**

Select customer.

The payment details of customer is reviewed.

**Alternate flow of events:** None

**Post condition:** None.

Ask for customer

Customer payments details

Select customer

**Name of the use case:** Consulting with bank and consulting with administrator.

**Description:** Contacting the bank for the customer payment and in case of any problem consulting with the administrator.

**Precondition:** Accounts manager is already logged in.

**Normal flow of events:**

Select transaction ID.

View transaction details.

Contact bank.

Receive payment.

Manage payment.

**Alternate flow of events:**

If any problem contact with administrator.

**Post condition:** None.

Select

Transaction ID

Contact bank

Receive payment

View

Transaction

details

Contact Administrator

Success

Problem

Manage payment

* Customer Care:

Responsible for getting feedback from customers and providing solutions to them.

* **Receiving Feedback from the Customers:** Responsible for admitting complaints, queries and feedback from the customers.
* **Providing Solutions to Customers:** Provide feasible solutions to the customers on their complaints and queries.

**Name of use case:** Getting feedback and providing solutions.

**Description:** To gain feedback from customers about products and services provided and giving solutions accordingly.

**Normal flow of events:**

Select customer.

Get feedback.

Provide solutions.

**Alternate flow of events:**

 If customer is not registered ask for registration at first.

**Post condition:** None.

Select customer

Send error message

Check

Get feedback /

Query

Not registered

Registered

Select service

Provide solutions