PROJECT ON

BAKERS POINT

**DEVELOPED BY：**

XUMINGYU (JASON)

XIONGBINHAN (DRUID)

GAOFEIXIANG (MICHAL)

]

ACKNOWLEDGEMENT

Acknowledgement we would like to thank Sana who supported we with understanding various programming languages,such as C#,java,HTML and so on. During these days in your class, We have learned much from you.

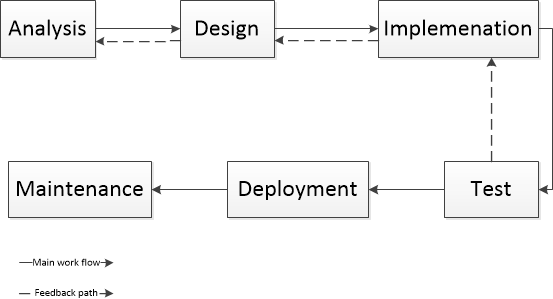
Firstly, you let us know what is the west thinking pattern—straight thinking pattern. And, We think, it is very important to understand the west’s thoughts. As you know, this can help us with our examination and interaction with foreigners.

Secondly,the course provides an introduction to object oriented analysis and design and UML.The software covers the need for Software Development Life Cycle(SDLC) and the various SDLC approaches.It provides a refresher session on object-oriented concepts.In addition,the course describes the concept of requirements gathering and use-case modeling for the proposed software system.The course also explains how to migrate from the requirements analysis phase to the design of software development.

College life is much different from high school life. In college, we should acquire knowledge as well as improve our integrated abilities. There are a lot of English classes for us to learn, such as Oral English, Listening, Reading, Writing, Translation, Grammar, English Literature, and so on. I enjoy these classes very much. Except English, we have many other subjects to learn. Life seems so busy here but it’s so fruitful for me.

Above all, we want to say thanks again to you. And thank you for your time.

PROJECT LIFE CYCLE MODEL



PROJECT PLAN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Sl No*** | ***Task Name*** | ***Planned Start Date*** | ***Planned Finish Date*** | ***Actual Start Date*** | ***Actual End Date*** | ***Person Responsible*** |
| 1 | Analyse the project requirements | 09/22/2016 | 09/22/2016 | 09/22/2016 | 09/22/2016 | Druid Jason Zena Primy Allen |
| 2 | Design the project | 09/29/2016 | 10/27/2016 | 09/29/2016 | 10/27/2016 | Druid Jason Zena Primy Allen |
| 3 | Implement the project | 11/03/2016 | 11/17/2016 | 11/03/2016 | 11/17/2016 | Druid Jason Zena Primy Allen |
| 4 | Test the project | 11/24/2016 | 12/29/2016 | 11/24/2016 | 12/29/2016 | Druid Jason Zena Primy Allen |

**FUNCTIONAL REQUIREMENT DOCUMENT**

Blog is take the network as the carrier, fast, easy, easy to release their own experience, timely and effective to easily communicate with others, and then set of rich and colorful personalized display in an integrated platform. NIIT blog for students to create a note to share collaboration platform for the sharing、learning、 teaching. (Basic Function Requirement)：

Visitor： Read Blog, Comment Blog, Leave message to blogger,  Pay attention to Blog, send private letter, Mark like and give a reward)

Blogger： Post, edit, delete & read blog, Classified, file the blog, Management for comment of blog, Message management, Personal info maintain, To pay attention to other bloggers.

Admin :User Management, Blog management and can recommend blog, Auto Automatic filtering sensitive information, Nonfunctional requirement)：

System can provide faster speed of response to users.

There is no obvious security hole, including SQL Injection & XSS script injection cross the script injection.

System need to compatibility response layout.

System should have expandable and maintainability.

The functional requirements of the system have been derived from the user requirements and from the interviews with the project stakeholders. The functional requirements have been designed to highlight the services that the system should provide. These requirements specify how the developers should design and develop the system in order to meet the expectations of the project stakeholders from the developed system.

**Customer Login**

1. The system allow each custom to login and reserve products by specifying a user name and password.
2. The system should ensure that the user name and password fields are not empty.
3. The system should authenticate the login credentials before providing access to the administration area of the application.

**Search Products**

1. The system must allow customers to search products by keywords or category.
2. The system should allow the customers to view detail description of a particular products.

**View Products**

1. The system must allow the customer to view the products.
2. The system must allow the customer to select a product and view details.

**Place Order**

1. The system should allow customers to place order for one or more products by providing the delivery information.
2. The system should ask customer for a correct phone number for contacting.

**Employee Login**

1. The system allow employee to login and manage order forms and details of products by specifying a user name and password.
2. The system should ensure that the user name and password fields are empty.
3. The system should authenticate the login credentials before providing access to the administration area of the application.
4. The system should provide access to the administration area when the authentication is successful.

**Add Product**

1. The system should allow an employee to add a new product.
2. The system should ensure that the details to a product are complete and valid.
3. The system should display the new product in the Bakers Point Page.

**Delete Product**

1. The system should allow an employee to delete a product from the fleet.
2. The system should display a confirmation message before an employee deletes a product.
3. The system should update the Bakers Point Page without the records of the deleted product.

**Update Product**

1. The system should allow an employee to edit a product.
2. The system should ensure that the edit details are complete and valid.
3. The system should display the updated product details in the Bakers Point Page.

**Mark an order**

1. The system should allow an employee to mark an order as delivered.
2. The marked order should be confirmed by the delivery executive.
3. Once the order be marked the customer should be able to see the change.

**Cancel an order**

1. The system should allow an employee to cancel an order.
2. The system should display a confirmation message before an employee cancel an order.
3. The system should send a confirmation message to the customer before the order cancelled.

**Log off**

1. The system should allow an employee to log off from the system.
2. The system should save the works that employee done before log off.

**UML USE CASE DIAGRAMS**

**User case diagram for the Customer actor**



**User case diagram for the Admin actor**



**Use case diagram for the visitor actor**



**Use case diagram for the blogger actor**



|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-01 | |
| Use-case Name | Customer Login | |
| Priority | High | |
| Actor | Customer | |
| Description | This use case describes how the Customer actors can login. | |
| Precondition | None | |
| Post-condition | If the use case is successful, the actor can login the system. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The actor is on the login page of the application. 2. The actor asks customer to put Username and password and validation to login the system. | 1. The system verifies that the Username is not empty. 2. The system verifies that the password is not empty. 3. The system verifies that the Username and password are correct. 4. The system displays the UI to login. 5. Use case ends. |
| Alternate course of Action | * 1. If the Username or password is empty the system displays an error message Username or Password is required.   4.1 If the Username or password is not correct the system displays an error message User not exists or incorrect password. | |

**USE CASE DESCRIPTION**

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-02 | |
| Use-case Name | Search Login | |
| Priority | High | |
| Actor | Customer | |
| Description | This use case describes how the Customer actors can search products. | |
| Precondition | None | |
| Post-condition | If the use case is successful, the actor can search the products. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The actor is on the Home page of the application. 2. The actor allows customers to search products by keywords or category. | 1. The system verifies that the keyword is not empty. 2. The system verifies that the category is not empty. 3. The system verifies that the detail descriptions are correct. 4. The system displays the UI to search products. 5. Use case ends. |
| Alternate course of Action | * 1. If the keyword is empty the system displays an error message Keyword is required. | |

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-03 | |
| Use-case Name | Place Order | |
| Priority | High | |
| Actor | Customer | |
| Description | This use case describes how the Customer actors can place order. | |
| Precondition | None | |
| Post-condition | If the use case is successful, the actor can place order. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The actor is on the order page of the application.  2. The actor asks customer to provide the delivery information and phone number. | 3. The system verifies that the delivery information is not empty.  4. The system verifies that the phone number is correct.  5. The system displays the UI to place order.  6. Use case ends. |
| Alternate course of Action | * 1. If the delivery information is empty the system displays an error message delivery information is required.   2. If the phone number is not correct the system displays an error message Phone number for contacting is not correct. | |

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-04 | |
| Use-case Name | Employee Login | |
| Priority | High | |
| Actor | Employee | |
| Description | This use case describes how the employee actors can login. | |
| Precondition | None | |
| Post-condition | If the use case is successful, the employee can login the system. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The actor is on the login page of the application. 2. The actor asks employee to put Username and password and validation to login the system. 3. The actor asks employee to login and manage order forms and details of products . 4. The actor provides the employee access to the administration area of the application. | 1. The system verifies that the Username and password is not empty. 2. The system verifies that the Username and password are correct. 3. The system displays the UI to login. 4. The system authenticates the login credentials. 5. The system verifies that the authentication is successful. 6. Use case ends. |
| Alternate course of Action | * 1. If the Username or password is empty the system displays an error message Username or Password is required.   2. If the login credentials do not matched the system displays an error message. | |

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-05 | |
| Use-case Name | Add Product | |
| Priority | High | |
| Actor | Employee | |
| Description | This use case describes how the Employee add a new product. | |
| Precondition | None | |
| Post-condition | If the use case is successful, the newly added product detail is displayed in the Product Page. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The actor is on the Add Product page of the application. 2. The actor selects the product category then specifies the product model and clicks the Add Product button. | 1. The system verifies that the product category ,model and detail are not empty and valid. 2. The system displays the newly added product details in the Product page. 3. Use case ends. |
| Alternate course of Action | * 1. If the product category,model, and detail fields are empty or not valid the system displays an error message User not exists or incorrect password. | |

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-06 | |
| Use-case Name | Delete Product | |
| Priority | High | |
| Actor | Customer | |
| Description | This use case describes how the Customer actors can delete product. | |
| Precondition | None | |
| Post-condition | If the use case is successful, the actor can delete product. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The actor is on the product fleet of the application. 2. The actor clicks the Delete button. | 1. The system displays a confirmation message before an actor deletes a product. 2. The system displays the Bakers Point Page without the records of the deleted product. 3. Use case ends. |
| Alternate course of Action | NA | |

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-07 | |
| Use-case Name | Update Product | |
| Priority | High | |
| Actor | Customer | |
| Description | This use case describes how the Customer actors can update product. | |
| Precondition | None | |
| Post-condition | If the use case is successful, the actor can update product. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The actor is on the update page of the application. 2. The actor edits the details of the product. | 1. The system checks that the edit details are complete and valid. 2. The system displays the edited reservation details in the update page. 3. Use case ends. |
| Alternate course of Action | If the details are empty or not valid the system displays an error message. | |

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-08 | |
| Use-case Name | Mark an order | |
| Priority | High | |
| Actor | Customer | |
| Description | This use case describes how the Customer actors can mark an order. | |
| Precondition | None | |
| Post-condition | If the use case is successful, the actor can mark an order t. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The actor is on the Mark Order page of the application. 2. The actor marks an order as delivered. | 1. The system allows an employee to mark an order as delivered. 2. The system confirms the marked order by the delivery executive. 3. The system allows the customer to see the change once the order be marked. 4. Use case ends. |
| Alternate course of Action | If the order is not be confirmed, the system returns to the Mark Order page.  If the customer can not see the change, the system displays an error message. | |

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-09 | |
| Use-case Name | Cancel an order | |
| Priority | High | |
| Actor | Employee and Customer | |
| Description | This use case describes how the employee can cancel an order. | |
| Precondition | None | |
| Post-condition | If the use case is successful, the actor can cancel an order. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The actor is on theCancel page of the application. 2. The actor asks employee to comfirm the cancel. 3. The actor tells the customer that the order has been canceled. | 1. The system displays a confirmation message before an employee cancel an order. 2. The system sends a confirmation message to the customer before the order cancelled Use case ends. 3. Use case ends**.** |
| Alternate course of Action | * 1. If the employee does not comfirm the cancel order,the system will return to the Cancel page.   4.1 If the customer does not get a message the system will send an error message. | |

|  |  |  |
| --- | --- | --- |
| Use-case Number | UC-10 | |
| Use-case Name | Login off | |
| Priority | High | |
| Actor | Employee | |
| Description | This use case describes how the employee can log off from the system. | |
| Precondition | None | |
| Post-condition | If the use case is successful, the actor can log off from the system. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The actor is on the log off page of the application. 2. The actor asks employee to save the works that the employee done before log off. | 1. The system verifies that the works have been saved. 2. The system displays the UI to login off. 3. Use case ends. |
| Alternate course of Action | * 1. If the works have not been saved the system displays an error message the work have not been saved.. | |

ACTIVITY DIAGRAMS

**Activity diagram for the Login activity**



**Activity diagram for the Search Products activity**



**Activity diagram for the View Products** 

Activity diagram for the Place Order activity

Activity diagram for the Employee Login activity



Activity diagram for the Add Products activity



Activity diagram for the Delete Product activity

Activity diagram for the Update Product activity Activity diagram for the Mark an order activity  Activity diagram for the Cancel an order activity 

Activity diagram for the Log off activity 

**CLASS DIAGRAMS**

**Class diagram for the models**

ENTITY RELATIONSHIP DIAGRAM

****

HARDWARE AND SOFTWARE REQUIREMENTS

**Hardware:** 3.0GHz or faster processor

: 2GB of RAM

: 80GB 7200RPM Hard Drive

: DVD Drive

: 10/100NIC

: Optical wheel mouse

: Keyboard

: 17” Color Monitor

**Operating system**: Windows 8.1(x64)

**Software**: Visual Studio 2012

:Internet Explorer 10