PROJECT ON

CHINESEBUDFOOD

**Developed by**

**Name :**YANG YUN PENG (David) **Reg.No.:** 2014141340

**Name :**WU YAN YAN(Queen)  **Reg.No:**2014141306



PROJECT TITLE

**BATCH CODE**  ：class2

**START DATE**  ：2016/09/22

**END DATE** ：2017/05/08

**NAME OF THE COORDINATOR**  ：SANA MAHERU

**NAME OF THE DEVELOPER** ：David,

**DATE OF SUBMISSION** ：2017/05/08



CERTIFICATE

This is to certify that this report, titled ChineseBudFood ,embodies the original work done by David, in partial fulfillment of their course requirement at NIIT.

COORDINATOR：Miss SANA MAHERU

ACKNOWLEDGEMENT

Our group is reasonable division of labor, the work is done in the fastest time, code, graphics, text documents, PPT production, are all very good. Initially we plan, is ready to do this project is perfect, at least need to achieve our goals. In the end, spent ten days to complete, thanks to the guidance of NIIT and SANA teacher here.

The project we done in partial fulfillment of the course requirements at NIIT.

Thanks Sana in this year's lead and teach , vivid classroom enable us to better understand the computer language , master the relevant skills and be able to well use them, leading us into in addition to games and social network outside of the computer world .In this year Sana is not only our teacher but also our friend, the style of teaching originates from life ,and higher than life , be called a true mentor, so that we have benefited a lot from her.

REQUIREMENT ELICITATION DOCUMENT

Q1. What are the objectives of your company?

Q2. Please brief us about the stakeholders of the company.

Q3. How many branches does your company have?

Q4. How many order services can you provide for your customers?

Q5. Do you provide any promotional discount for reservation?

If yes, how?

Q6. How do you maintain customer information?

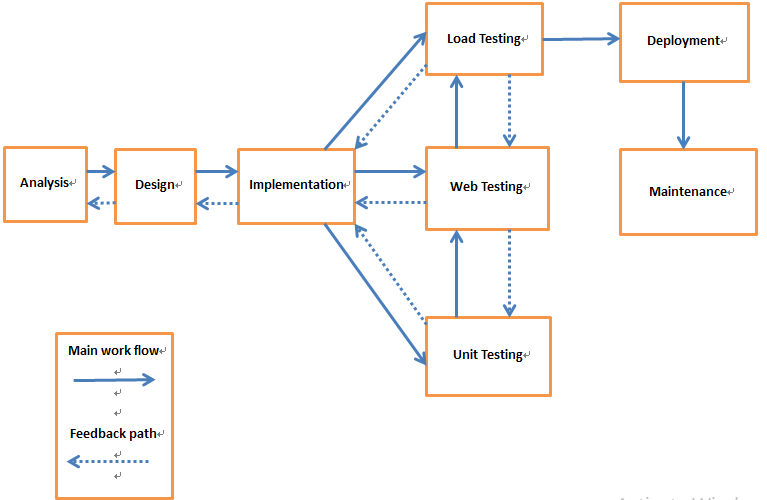
Q7. How do you keep track of how many orders are reserved for a day?

Q8. How do you maintain reservation information?

Q9. How does a customer cancel an order?

Q10. How do you collect feedbacks from customers?

PROJECT LIFE CYCLE MODEL



VISION DOCUMENT

**Vision of the Project**

We will provide a Web application named ChineseBudOnline that can be used by both the customers and the employees of Chinese Bud Food Services.

**About the Company**

Chinese Bud Food Services ,founded in 2009 in San Francisco. The Company provide the food services for the customers

**Requirements Summary**

The management of Chinese Bud Food Services, in the first phase, wants that the ChinsesBudOnline application should enable customers to book tables and place their orders online. In this phase, the orders that need to be delivered

**Project Goals**

The project goals are:

* Analyze the requirements to create solution.
* Design the application based on the results of the analysis phase.
* Create a Web-based application based on the MVC model based on the application design.
* Test the implemented application.
* Host the application on Azure.
* Ensure scalability of the hosted application.

**Project Stakeholders**

The primary stakeholders of the projects are:

* Management of Chinese Bud Food Services.
* Employees of Chinese Bud Food Services.
* Customers of Chinese Bud Food Services.

PROJECT PLAN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Sl No*** | ***Task Name*** | ***Planned Start Date*** | ***Planned Finish Date*** | ***Actual Start Date*** | ***Actual End Date*** | ***Person Responsible*** |
| *1* | *Analyze the project requirements* | *22/09/2016* | *22/09/2016* | *22/09/2016* | *22/09/2016* | *David* |
| *2* | *Design the project* | *29/09/2016* | *13/10/2016* | *29/09/2016* | *13/10/2016* | *David* |
| *3* | *Implement the project* | *20/10/2016* | *17/11/2016* | *20/10/2016* | *17/11/2016* | *David* |
| *4* | *Test the project* | *24/11/2016* | *15/12/2016* | *24/11/2016* | *15/12/2016* | *David* |
| \*David will be working four hours per working day to complete the Project.  \*Saturdays and Sundays as being holidays are not considered as working days. | | | | | | |

WEEKLY STATUS REPORT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Developer: David Phase:1 and 2**  **Period:From** 22/09/2016 **To** 10/11/2016 | | | | |
| **Activity/Artifact** | **Responsibility** | **Planned Completion Date** | **Completion Status(%)** | **Reason for Incompletion** |
| * Create a requirement   elicitation document for the project to gather more specific information on how the system should work.   * Create a presentation of the project life cycle model for the client/stakeholders. * Create a vision document of the project. * Create the project plan. * Create a Weekly status report template. * Create a functional requirement document of the system based on the user requirements gathered from the client. * Model the system using UMLuse case diagrams. * Create Use-case Descriptions * Create the activity diagrams of the system * Create the entity-relationship model. * Create the software requirement specification | Julian  Good  David  Carl  Lelia | 22/09/2016  29/09/2016  20/10/2016  26/10/2016  10/11/2016 | 100%  100%  100%  100%  100% | NA  NA  NA  NA  NA |

|  |
| --- |
| **Functional REQUIREMENT DOCUMENT** |

|  |
| --- |
| 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.  **Search Order Information**   1. The system must allow the users to select a Meal date and selected an adaptable table for the meal. 2. The system should ensure that both the Meal date and a table type is selected to search for an adaptable table. 3. The system should ensure that the selected Meal date should not be earlier than the current date. 4. The system should allow the users to view detail description of a particular table seats and the related services.   **Reserve Tables**   1. The system must allow the users to reserve a cab providing the following information:  * Meal time * The reserved numbers of the seats * Customer name * Customer phone number  1. The system should ensure that meal time, reserved numbers of the seats, customer name and customer phone number are specified during the reservation process. 2. The system should ensure that the customer name should only contain character and spaces. 3. The system should ensure that the customer phone number should be in the (XXX) XXX-XXXX format. 4. The system should display a success message once the reservations process completes.   **View Reservation**   1. The Customer can register the website and get his own account, the account can be your e-mail address , your phone number and also can be a valid name. 2. Once you reserve a table successfully, you can receive the information about your order in detail. It can be in your personal account or in your e-mail Inbox.   **Employee Login**   1. The system must allow an employee to login and administer Meal and reservation details by specifying a user name and password. 2. The system should ensure that the user name and password fields and not 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 Items**  1. The system must allow an employee to add a new item to the restaurant menu.  2. The system must ensure that the details to an item are complete and valid.  3. The system must display the new items details in the ChineseBudOnline application.  **Edit Items**  1. The system must allow an employee to edit an existing item.  2. The system must ensure that the edit details are complete and valid.  3. The system must display the edited items details in the ChineseBudOnline application.  **Delete Items**  1. The system must allow an employee to delete an existing item from the restaurant menu.  2. The system must display a confirmation message before an employee delete an item.  3. The system must update the ChineseBudOnline application without the records of the deleted items.  **Edit Reservation**   1. The system must allow an employee to edit a reservation 2. The system must allow an employee to edit the meal time, reserved numbers of the seats, customer name, and customer phone number. 3. The system must ensure that the edit details are complete and valid.   4 . The system must display the edited reservation details in the Manage Reservation page.  **Cancel Reservation**   1. The system must allow an employee to cancel a reservation. 2. The system must display a confirmation message before an employee cancels a reservation. 3. The system must update the Manage Reservation page without the records of the deleted reservation.   **Blog**   1. The system must allow visitor to see the blog. 2. The system must allow visitor to comment the blog. 3. The system must allow visitor Leave message to blogger. 4. The system must allow the Blogger to manage the blog. |

|  |
| --- |
| **User case diagram for the Customer actor**    **User case diagram for the Employee actor** |

|  |
| --- |
|  |
| **USE CASE DESCRIPTIONS** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use-Case Search Order Information**  *Table 1: Use Case – Search Order Information*   |  |  |  | | --- | --- | --- | | Use-case  Number | UC-01 | | | Use-Case Name | Search Order Information | | | Priority | High | | | Actor | Customer and Employee | | | Description | This use case describes how the Customer or Employee can search for the Order Information | | | Precondition | None | | | Basic course of  Action | **User Action** | **System Response** | | 1. The actor is on the Home Page of the application. 2. The actor Selects a meal date and the reserved numbers of the seats and submits the information. | 1. The system verifies that the meal date is not empty. 2. The system verifies that the meal date is not earlier than the current date. 3. The system displays the UI to reserve a table. 4. Use case ends. | | Alternate course of Action | * 1. If the pickup date is empty the system displays an error message *Meal Date* is required.   2. If the meal date is earlier that the current date the system displays an error message Meal date cannot be earlier than current date and then returns to select a meal date and the reserved numbers of the seats. | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use – Case Reserve Tables**  *Table2:Use Case – Reserve Tables*   |  |  |  | | --- | --- | --- | | Use-case  Number | UC-02 | | | Use-Case Name | *Reserve Tables* | | | Priority | High | | | Actor | Customer and Employee | | | Description | This use case describes how the Customer or Employee actors specify details to reserve a meal. | | | Precondition | None | | | Basic course of  Action | **User Action** | **System Response** | | 1. The actor is on the reservation of the application.  2. The actor specifies the meal time, reserved numbers of the seats, name, and the phone number.  3. The actor clicks the **Book this table** button to reserve a table. | 4. The system verifies that the meal time, reserved numbers of the seats, name, and the phone number are not empty and valid.  5. The system displays message that the reservation has been successfully done.  6. Use case ends. | | Alternate course of Action | * 1. If the meal date, reserved numbers of the seats, name, and the phone number are not empty and valid the system displays an error message. | | |
| **Use Case View Reservation**  *Table3:Use Case – View Reservation*   |  |  |  | | --- | --- | --- | | Use-case Number | UC-03 | | | Use-Case Name | *View Reservation* | | | Priority | High | | | Actor | Customer and Employee | | | Description | This use case describes how the Customer or Employee actors see the details of the reservation. | | | Precondition | None | | | Basic course of  Action | **User Action** | **System Response** | | 1. The actor is on the reservation of the application.  2. The actor specifies the meal time, reserved numbers of the seats, name, and the phone number.  3. The actor clicks the **See the Detail** button to reserve a table. | 4. The system show the customer meal time, reserved numbers of the seats, name, and the phone number.  6. Use case ends. | | Alternate course of Action | * 1. If the customer have emergency to change the order information, he can log in and change the meal date, reserved numbers of the seats, name, and the phone number | | | |
| **Use – Case Employee Login**  *Table4:Use Case – Employee Login*   |  |  |  | | --- | --- | --- | | Use-case Number | UC-04 | | | Use-Case Name | *Employee Login* | | | Priority | High | | | Actor | Employee | | | Description | This use case describes how the Employee actors login into the system. | | | Precondition | None | | | Basic course of  Action | **User Action** | **System Response** | | 1. The actor is on the administrator login page of the application.  2. The actor specifies the administrator login user name and password and clicks the **Login in** button. | 3. The system verifies that the specified user name and password is not empty and valid.  4. The system authenticates the login credentials.  5. The system displays the administration area when the authentication is successful.  6. Use Case ends | | Alternate course of Action | 3.1 If the user name and password fields are empty the system displays an error message *The user name or password provided is incorrect.*  *4.1* If the login credentials do not matched the system displays an error message. | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use – Case Add Items**  *Tables 5: Use Case – Add Items*   |  |  |  | | --- | --- | --- | | Use-case Number | UC-04 | | | Use-Case Name | Add Items | | | Priority | High | | | Actor | Employee | | | Description | This use case describes how the Employee actor adds a new item to the restaurant menu. | | | Precondition | None | | | Post-condition | If the use case was successful, the newly added items detail is displayed in the ChineseBudOnline application. | | | Basic course of  Action | **User Action** | **System Response** | | 1. The actor is on the Add Item Page of the application. 2. The actor selects the item category, then specifies the type of dishes and price and clicks the **Add Item** button. | 1. The system verifies that the item category, type of dishes, and price fields are not empty and valid. 2. The system displays the newly added items details in the ChineseBudOnline application. 3. Use case ends. | | Alternate course of Action | 3.1 If the item category, type of dishes, and price fields are empty or not valid the system displays an error message. | | |
| **Use-case Edit Items**  *Table 6:Use Case – Edit Items*   |  |  |  | | --- | --- | --- | | Use-case Number | UC-05 | | | Use-Case Name | Edit Items | | | Priority | High | | | Actor | Employee | | | Description | This use case describes how the Employee actor edits an existing item detail. | | | Precondition | None | | | Post-condition | If the use case was successful, the updated items detail is displayed in the ChineseBudOnline application. | | | Basic course of  Action | **User Action** | **System Response** | | 1. The actor is on the Edit Item Page of the application. 2. The actor updates the item category, type of dishes and price as required and clicks the Save button. | 1. The system verifies that the item category, type of dishes, and price fields are complete and valid. 2. The system displays the updated items details in the ChineseBudOnline application. 3. Use case ends. | | Alternate course of Action | 3.1 If the item category, type of dishes, and price fields are empty or not valid the system displays an error message. | | |
| **Use-case Delete Items**  *Table 7:Use Case – Delete Items*   |  |  |  | | --- | --- | --- | | Use-case Number | UC-06 | | | Use-Case Name | Delete Items | | | Priority | High | | | Actor | Employee | | | Description | This use case describes how the Employee actor deletes an existing item from the restaurant menu. | | | Precondition | None | | | Post-condition | If the use case was successful, the system updates the ChineseBudOnline application without the records of the deleted items. | | | Basic course of  Action | User Action | System Response | | 1. The actor is on the Delete Item Page of the application.  2. The actor clicks the Delete button. | 3. The system displays a confirmation message before an actor deletes an item.  4. The system displays the ChineseBudOnline application by removing the records of the deleted items.  5. Use case ends. | | Alternate course of Action | NA | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Use-case Number | *UC-07* | | | Use-Case Name | Edit Reservation | | | Priority | High | | | Actor | Employee | | | Description | This use case describes how the Employee actor edits a reservation in the system | | | Precondition | None | | | Basic course of  Action | User Action | System Response | | 1. The actor is on the Edit Reservation page of the application.  2. The actor edits the edit the meal time, reserved number of the seats, names, and customer phone number and clicks the **Delete** button. | 1. The system checks that the edit details are complete and valid. 2. The system displays the edited reservation details in the Manage Reservation page. 3. Use case ends. | | Alternate course of Action | 3.1 If the meal time, reserved number of the seats, names, and customer phone number are empty or not valid the system displays an error message. | |   **Use – Case Edit Reservation**  *Table 8:Use Case – Edit Reservation* |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Use-case  Number | UC-08 | | | Use-Case Name | Cancel Reservation | | | Priority | High | | | Actor | Employee | | | Description | This use case describes how the Employee actor cancels a reservation in the system | | | Precondition | None | | | Post-condition | If the use case was successful, the system displays the Manage Reservation page without the records of the deleted reservation. | | | Basic course of Action | **User Action** | **System Response** | | 1. The actor is on the Cancel Reservation page of the application.  2. The actor clicks the **Cancel** button. | 3. The system displays a confirmation before an actor cancels a cab.  4. The system displays the Manage Reservation page by removing the reservation.  5. Use case ends. | | Alternate course of Action | **NA** | |   **Use Case Cancel Reservation**  *Table 9:Use Case – Cancel Reservation* |
| Use-Case Blog function  *Table 10:Use Case –*   |  |  |  | | --- | --- | --- | | Use-case Number | UC-10 | | | Use-Case Name | Blog | | | Priority | High | | | Actor | Visitor and Blogger | | | Description | This use case describes how the visitor visit and comment the blog and the blogger manage the blog and comments. | | | Precondition | None | | | Post-condition | If the use case was successful, the system updates the ChineseBudOnline application with the comments and messages. | | | Basic course of  Action | User Action | System Response | | 1. The actor is on the blog page.  2. The actor clicks the view or “Hand in” Button | 3. The system displays the blog or the comments.  4. The system show the blogger’s  Blog and comment.  5. Use case ends. | | Alternate course of Action | NA | | |
| **Activity diagram for the View Reservation activity**    **Activity diagram for the Employee Login activity**    **Activity diagram for the Search Order activity**  **Activity diagram for the Reserve Tables activity** |