**Project Name: Green Pig’s Kitchen System**

**Executive Sponsor:**

**Project Manager: Mark Robert M. Aldecimo**

**Business Analyst:**

Date: August 12, 2014

**Table of Contents**

[Approvals 4](#_Toc221528079)

[Use Case List 4](#_Toc221528080)

[1 Login](#_Toc221528081) 5

[1.1 Feature Process Flow / Use Case Model](#_Toc221528082) 6

[1.2 Login](#_Toc221528083) 6

[2 Dish Inventory](#_Toc221528084) 7

[2.1 Feature Process Flow / Use Case Model](#_Toc221528085) 7

[2.2 Create Dish](#_Toc221528086) 8

[2.3 View Dish](#_Toc221528086) 9

[2.4 Delete Dish](#_Toc221528086) 10

[2 Transactions](#_Toc221528084) 11

[3.2 Feature Process Flow / Use Case Model](#_Toc221528086) 12

[3.2 Take Order](#_Toc221528086) 12

[3.3 Process Order](#_Toc221528086) 13

[3.4 Sales Transaction](#_Toc221528086) 14

Revision History

| **Version** | **Date** | **Revision Description** |
| --- | --- | --- |
| .01 |  |  |
| .02 |  |  |
| .03 |  |  |
| .04 |  |  |
| 1.0 |  | Approved Use Case |
|  |  |  |
|  |  |  |

# Approvals

We have carefully assessed the Use Cases for this project. This document has been completed in accordance with the requirements of the System Development Methodology.

MANAGEMENT CERTIFICATION - Please check the appropriate statement.

\_\_\_\_\_\_ the document is accepted.

\_\_\_\_\_\_ the document is accepted pending the changes noted.

\_\_\_\_\_\_ the document is not accepted.

We fully accept the changes as needed improvements and authorize initiation of work to proceed. Based on our authority and judgment, the continued operation of this system is authorized.

(\*=Required \*\*= Submit for Review Approval Not Required)

Executive Sponsor\*\* DATE

Project Sponsor\* DATE

Quality Assurance Manager / Team Lead\* DATE

Business Analyst Manager / Team Lead\* DATE

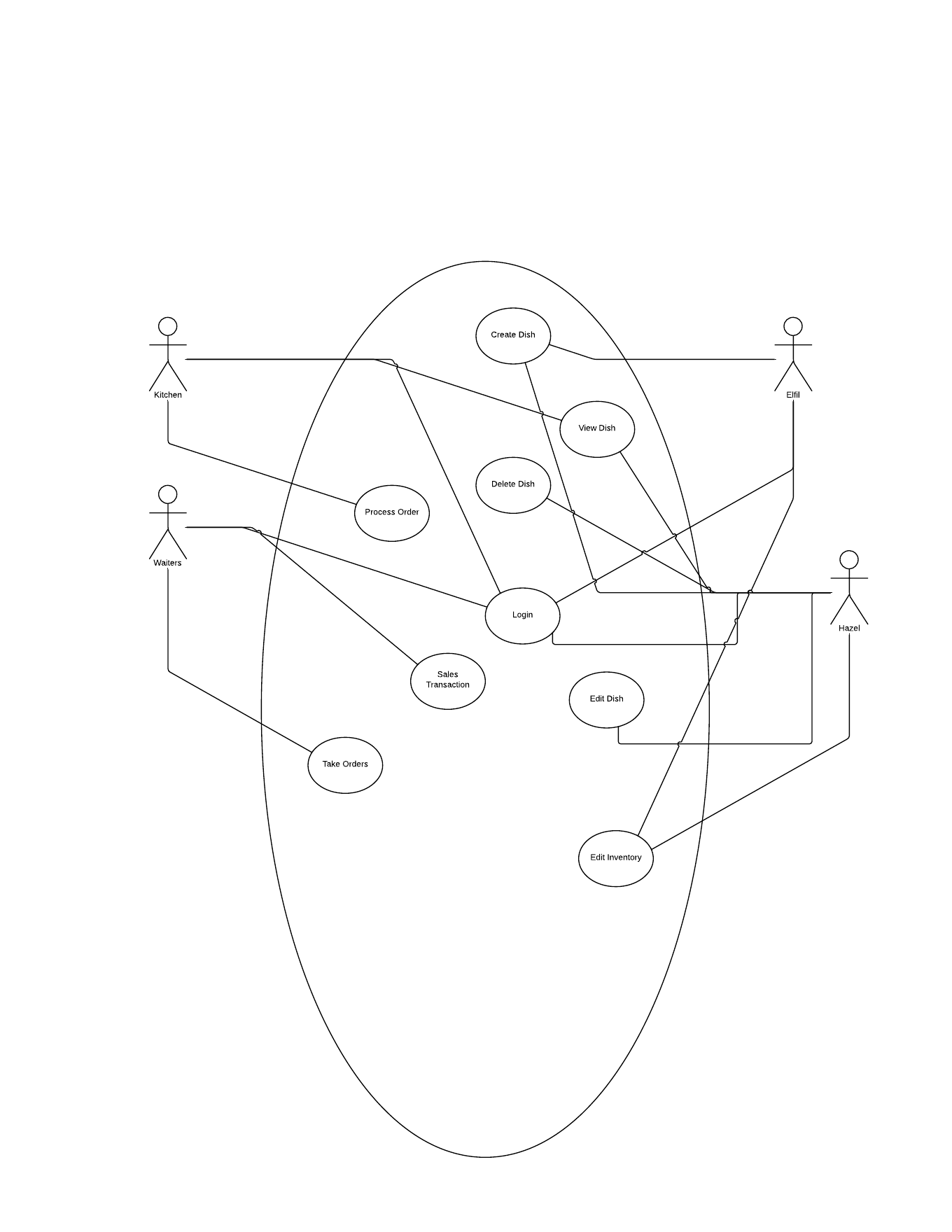
Project Manager DATE

# Use Case List

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Primary Actor** | **Use Cases** |
| 100001 | Elfil | Login |
| 100002 | Hazel | Login, View Dish, Delete Dish, Edit Dish |
| 100003 | Kitchen | Login, Create Dish |
| 100004 | Waiter | Login, Sales Transaction |

# Login

## Login Use Case Model

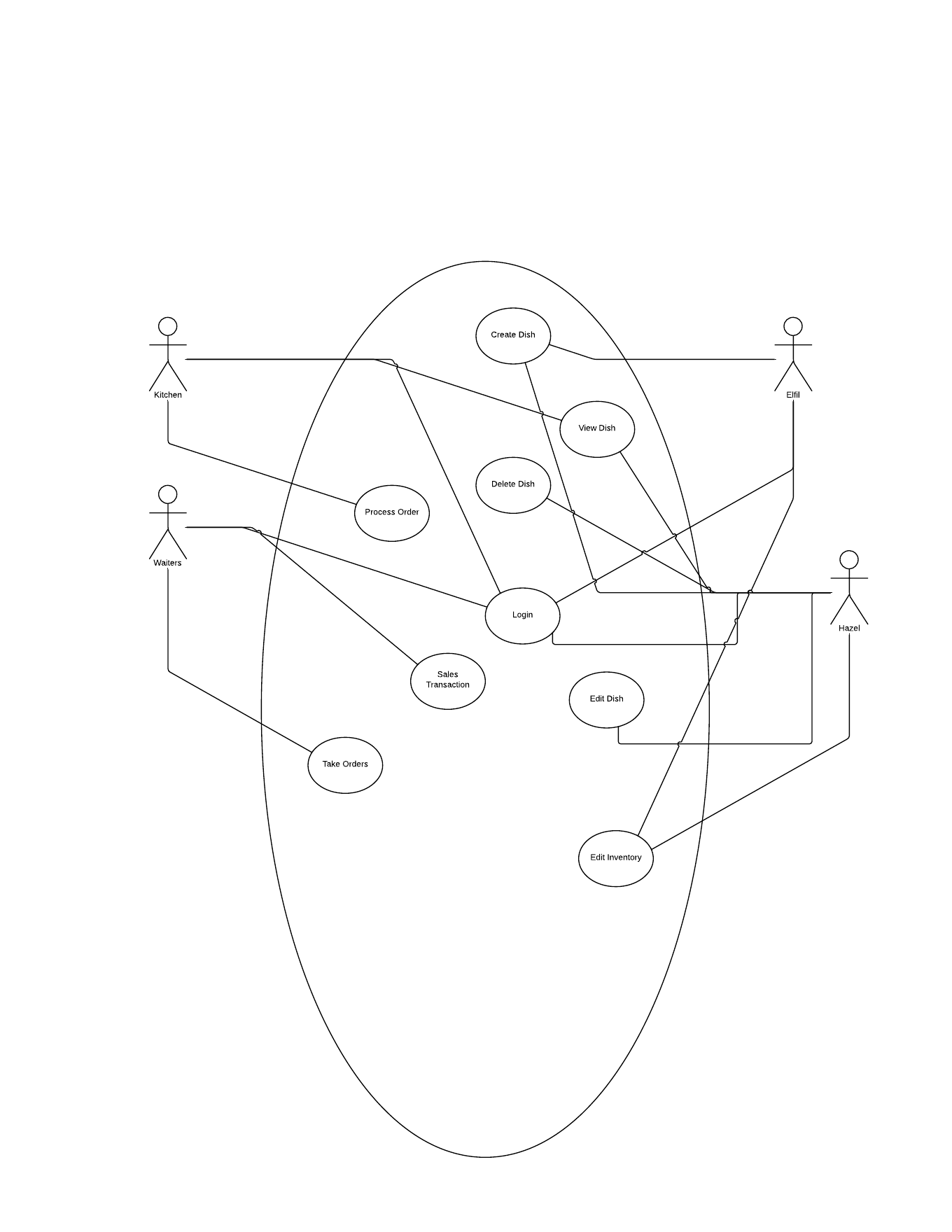


## Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-1.2.1 | | | |
| **Use Case Name:** | Login | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Elfil, Hazel, Kitchen, Waiter | | |
| **Description:** | | This allows the user to login to the system. | | |
| **Trigger:** | | The user wants to login to the system. | | |
| **Preconditions:** | | n/a | | |
| **Postconditions:** | | 1. User is now logged in to the system. 2. User receives an error message stating wrong login credentials. | | |
| **Normal Flow:** | | 1. User inputs the login credentials to the system 2. User hits login button 3. System verfiies the login credentials 4. System authenticates the user 5. System displays the main menu for the user | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | |  | | |
| **Exceptions:** | | [Describe any anticipated **error conditions** that could occur during execution of the use case, and define how the system is to respond to those conditions.  e.g. Exceptions to the Withdraw Case transaction  1a. In step 1 of the normal flow, if the user inputs invalid wrong login credentials.   1. System verifies the login credentials 2. System displays wrong login error message 3. Returns to login menu | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | Once a day | | |
| **Special Requirements:** | | Database is on. | | |
| **Assumptions:** | | User knows his/her own login credentials | | |
| **Notes and Issues:** | |  | | |

# Dish Inventory

## Use Case Model



## Create Dish

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-2.2.1 | | | |
| **Use Case Name:** | Create Dish | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Elfil (primary) | | |
| **Description:** | | Create a dish in the inventory. | | |
| **Trigger:** | | Elfil has a new dish to add in the database of dishes. | | |
| **Preconditions:** | | 1. **Elfil is logged in.** | | |
| **Postconditions:** | | 1. A new dish is added to the database of dishes. 2. System displays invalid values for the dish | | |
| **Normal Flow:** | | 1. Elfil goes to create dish menu. 2. Elfil inputs the correct information for the dish. 3. System adds the dish to the database. | | |
| **Alternative Flows:** | |  | | |
| **Exceptions:** | | 2a. In step 2 of normal flow, if Elfil inputs invalid value for the dish   1. System displays error message 2. Returns to create dish menu | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | Everytime a new dish should be added to the database of dishes. | | |
| **Special Requirements:** | | Database is on | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

## View Dish

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-2.2.2 | | | |
| **Use Case Name:** | View Dish | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Kitchen (primary), Hazel | | |
| **Description:** | | Views the information of the dish. | | |
| **Trigger:** | | User wants to view the infromation of a dish. | | |
| **Preconditions:** | | 1. **User is logged in.** 2. **Dish exists in the database.** | | |
| **Postconditions:** | | 1. System displays the information of a dish 2. System displays error message. | | |
| **Normal Flow:** | | 1. User inputs the dish. 2. System fetches the infromation of the dish from the database 3. System displays that information | | |
| **Alternative Flows:** | |  | | |
| **Exceptions:** | | 2a. In step 2 of normal flow, if Elfil inputs invalid dish   1. System displays error message 2. Returns to view dish menu | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | Everytime a new dish should be added to the database of dishes. | | |
| **Special Requirements:** | | Database is on | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

## Edit Dish

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-2.2.3 | | | |
| **Use Case Name:** | Edit Dish | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Hazel (primary) | | |
| **Description:** | | Edits the information about the dish. | | |
| **Trigger:** | | Hazel wants to change an info about the dish. | | |
| **Preconditions:** | | 1. **Hazel is logged in.** 2. **Dish exists in the database.** | | |
| **Postconditions:** | | 1. Information about the dish is changed 2. System displays “Dish not found” message. | | |
| **Normal Flow:** | | 1. Hazel goes to edit dish menu. 2. Hazel inputs the dish. 3. System verifies the dish. 4. System display the editable information about the dish. 5. Hazel inputs the new information. 6. Hazel hits submit button. 7. System changes the information in the database. | | |
| **Alternative Flows:** | |  | | |
| **Exceptions:** | | 2a. In step 2 of normal flow, if Hazel inputs invalid dish   1. System displays error message 2. Returns to edit dish menu | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | Everytime an information about the dish should be changed. | | |
| **Special Requirements:** | | Database is on | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

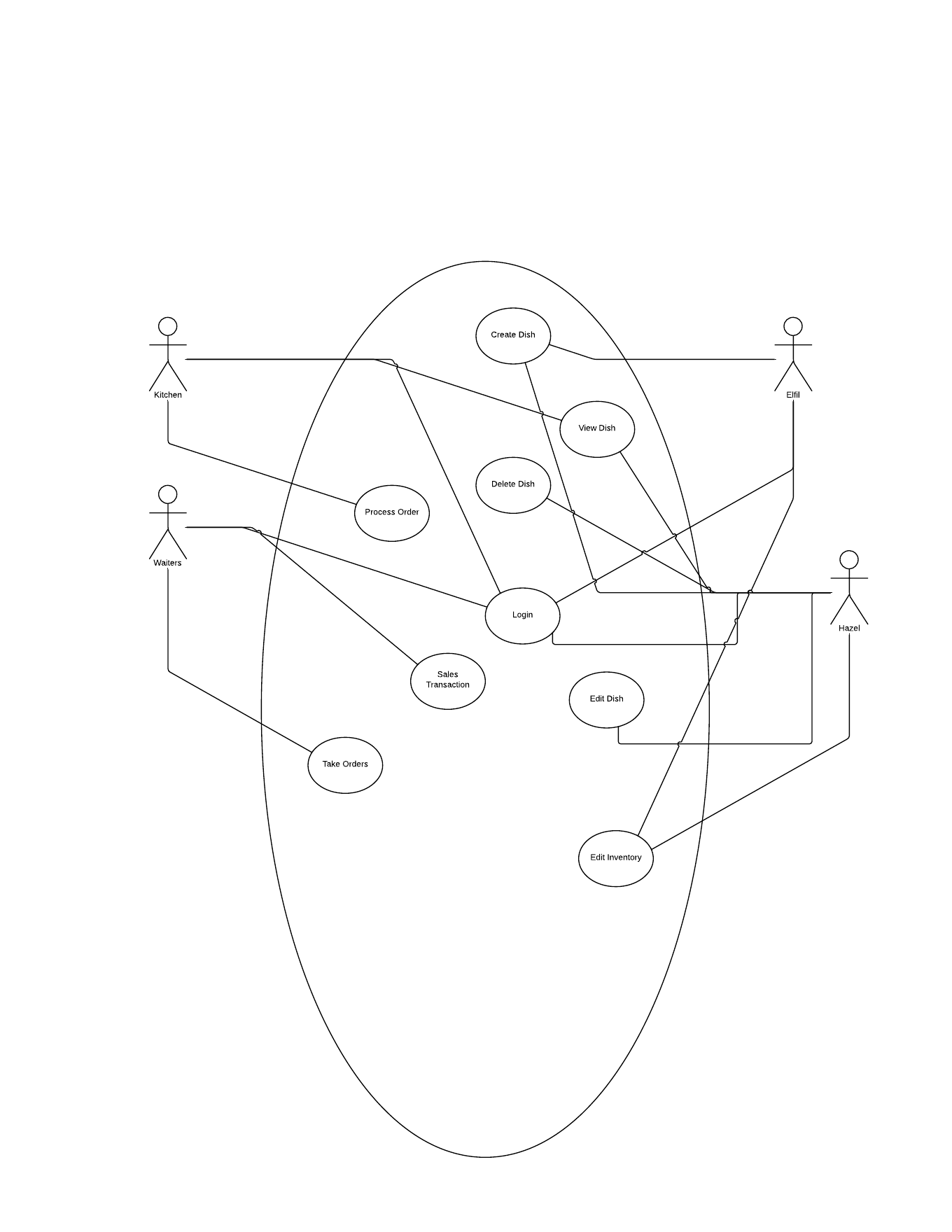
## Delete Dish

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-2.2.4 | | | |
| **Use Case Name:** | Delete Dish | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Hazel (primary) | | |
| **Description:** | | Deletes a dish in the database | | |
| **Trigger:** | | User wants to delete a dish in the database | | |
| **Preconditions:** | | 1. **Hazel is logged in.** 2. **Dish exists in the database** | | |
| **Postconditions:** | | 1. System deletes the dish in the database 2. System displays and error message | | |
| **Normal Flow:** | | 1. Hazel goes to the delete dish menu 2. Hazel inputs the dish she wants to delete 3. System verifies the dish 4. System deletes the dish in the database 5. System displays success message | | |
| **Alternative Flows:** | |  | | |
| **Exceptions:** | | 2a. In step 2 of normal flow, if Hazel inputs invalid dish   1. System displays error message 2. Returns to delete dish menu | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | Everytime a dish should be deleted from the database of dishes. | | |
| **Special Requirements:** | | Database is on | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

# Transaction

## Use Case Model



## Take Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-3.3.1 | | | |
| **Use Case Name:** | Take Order | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Waiter (primary) | | |
| **Description:** | | This allows the waiter to take the orders of the customers and input it to the system | | |
| **Trigger:** | | A customer wants to order a dish. | | |
| **Preconditions:** | | 1. Dish exists in database 2. There is enough ingredients to prepare dish. 3. User is logged in to the system. | | |
| **Postconditions:** | | 1. An order will be input to the system | | |
| **Normal Flow:** | | 1. User goes to take order menu. 2. User inputs order. 3. System receives the order. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | |  | | |
| **Exceptions:** | |  | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | Every time there is an order. | | |
| **Special Requirements:** | | Database is on. | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

## Process Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-3.3.2 | | | |
| **Use Case Name:** | Process Food | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Kitchen (primary) | | |
| **Description:** | | This allows the system to display the orders of the customer and track the preparation of food and the ingredients used. | | |
| **Trigger:** | | A dish is ordered by a customer. | | |
| **Preconditions:** | | 1. Dish exists in database 2. There is an order in the system 3. There is enough ingredients to prepare dish. 4. User is logged in to the system. | | |
| **Postconditions:** | | 1. Number of ingredients used would be deducted to the total number of ingredients 2. System displays “Insufficient ingredients” message 3. Order in the system would be marked complete. | | |
| **Normal Flow:** | | 1. User goes to process order menu 2. User picks an order to process 3. System computes for the total ingredients to be used 4. System verifies if there is enough ingredients 5. System displays a success message. | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | |  | | |
| **Exceptions:** | | 4a. In step 3 of normal flow, if there is no enough ingredients   1. System displays an “Insufficient ingredients” message 2. Returns to process order menu | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | Every time there is an order. | | |
| **Special Requirements:** | | Database is on. | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |

## Sales Transaction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | UC-3.3.3 | | | |
| **Use Case Name:** | View Dish | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | Waiter (primary) | | |
| **Description:** | | Allows the user to make input and track sales. | | |
| **Trigger:** | | A customer would like to pay for the order. | | |
| **Preconditions:** | | **n/a** | | |
| **Postconditions:** | | 1. System tracks the sales 2. System displays a success message. | | |
| **Normal Flow:** | | 1. User inputs the sale. 2. System processes that sale. 3. System displays success message | | |
| **Alternative Flows:** | |  | | |
| **Exceptions:** | |  | | |
| **Includes:** | |  | | |
| **Frequency of Use:** | | Everytime there a customer pays | | |
| **Special Requirements:** | | Database is on | | |
| **Assumptions:** | |  | | |
| **Notes and Issues:** | |  | | |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |