**<<Fast Food Delivery System>>**

Software Engineering: Catherine Woods

Submitted By: Aleksandar Zoric (TL\_KCPSD\_Y2)

Submitted on: 20/04/2015

Table of Contents

[**1.** **Introduction** 3](#_Toc369007329)

[**2.** **Functional Requirements** **Error! Bookmark not defined.**](#_Toc369007330)

[**2.1.** **User Requirements** **Error! Bookmark not defined.**](#_Toc369007331)

[**2.2.** **System Requirements** 6](#_Toc369007332)

[**3.** **System Model** 18](#_Toc369007333)

[**4.** **Data Model** 57](#_Toc369007334)

[**5.** **Database Schema** 58](#_Toc369007335)

[**6.** **Program Specifications** 65](#_Toc369007336)

[**7.** **Conclusion** 66](#_Toc369007337)

[**8.** **References:** 67](#_Toc369007338)

# **Introduction**

***Author:*** Aleksandar Zoric

This is a document related to the ‘Fast Food Delivery’ program which, is designed for staff members’ use across all fast food restaurants around the country. It is user friendly and easy to use. All the required functions that a staff member may need to register a new staff member, record food orders, increase stock level, manage reports etc. is available in this program, outline below. Both user and system requirement are documented, which can be viewed to gain a better insight into the program itself and all of its functionalities.

The system and data model are also documented in this document, and also the database schema. These documents can be also viewed, it gives the user a better idea on how the program operates, the files used to store various different information that will be recorded by the admin e.g. staff members file, orders file, reports file etc.

If any issues occur, please contact the author (Above).

# **Functional Components**

# **User Requirements**

The user requirements, will be clearly outlined here. It will describe all the main functions of the software system under the following headings.

## **The system performs staff administration**

* + 1. The system will allow the details of each new staff member to be recorded
    2. The system will allow each current member details to be amended
    3. The system will allow to list all current staff members
    4. The system will allow to cancel or delete a members file

## **The system perform food order management**

* + 1. The system will allow food orders to be recorded
    2. The system will allow food orders to be amended, also an option to dispatch.
    3. The system will generate a listing of a food order
    4. The system will allow for a payment to be recorded
    5. The system will allow for an order to be cancelled

## **The system will perform food management**

* + 1. The system will allow new food to be added
    2. The system will allow for the added food to be amended
    3. The system will allow for food to be removed from the system
    4. The system will perform a listing of all current food items

## **The system will generate reports**

* + 1. The system will generate a sales analysis

# **System Requirements**

The system requirements, minimum and recommended will be clearly outlined throughout this document. For our system, which in this case is fast food deliveries, you will be able to see all required requirement to manage staff, process orders, manage food and manage reports.

## **System Level Diagram**

Customer

Manager

Staff

## **Manage Staff**

## This is where an admin would have permission to register a new staff member, amend a current staff member, list all current staff members or remove a registered staff member.

### **Register Staff**

## This is where a new staff member is registered by the manager

<<Include>>

Manager

Staff

# Activity Diagram

Store staff Details

Assign Staff ID

**N**

Display Confirmation Message

**Y**

Try again?

**N**

Display Error Message

Valid Data?

Enter the new staff details

The admin/staff request the staff file.

## Use Case Narrative

|  |
| --- |
| **Staff** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Register Staff Member** | | **Author:** |
| **Use Case Id** | 01 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Admin/Staff | | |
| **Other Participating Actors** | New Member | | |
| **Description** | This function registers a new staff member. | | |
| **Preconditions** |  | | |
| **Trigger** | A members’ file is requested in order to add a new member. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1**: The manager invokes the register staff function.  **Step 4:** The Manager enters the members details:   * Surname * Forename * DOB * Email * Phone   **Step 5:** The manager confirms that the staff is to be registered | **Step 2:** The system determines the next Staff\_ID and the UI  **Step 6:** The system validates the data entered:   * All fields must be entered * DODB must be in dd/mm/yyyy format   **Step 7**: The system determines the next Staff\_ID  **Step 8**: The Registration Date is assigned the current system date  **Step 9**: The staff details are saved in the Staff File  **Step 10**: The system displays a confirmation message  **Step 11:** The system resets the UI | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Required field not entered** |  | **Step 5:** a required field is not entered  **Step 6**: The system displays an appropriate error message | |
| **DOB invalid** |  | **Step 5:** DOB is in the future  **Step 6**: The system displays an appropriate error message | |
|  |  | | |
| **Conclusions** | A new member has been added to the staff file. | | |
| **Post conditions** | This staff may now use system functions | | |
| **Business Rules** | Staff must be over 18 | | |
| **Implementation Constraints** |  | | |

### **Amend Staff Member**

## This is where a staff members’ details are amended/edited by the manager

<<Include>>

Staff

Manager

## Use Case Narrative

|  |
| --- |
| **Staff** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Amend Staff Member** | | **Author:** |
| **Use Case Id** | 02 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Manager/Staff | | |
| **Other Participating Actors** | Member | | |
| **Description** | This function amends/edits a current staff member’s details. | | |
| **Preconditions** | A member must exist in the members file | | |
| **Trigger** | A staff file is requested in order to amend/edit a new member. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager invokes the amend staff function.  **Step 3:** The admin chooses the member for the details to be amended for.  **Step 5:** The admin updates the required details:   * Surname * Forename * DOB * Email * Phone   Step 6: The admin confirms that the customer is to be amended | **Step 2:** System retrieves summary of all staff details from staff file and displays in UI in order by Staff ID.  **Step 4:** The system displays the staff details.  **Step 6:** The system validates the data entered:   * All fields must be entered * DODB must be in dd/mm/yyyy format * DOB must not be in the future   **Step 7:** The new staff details are updated in the Staff File  **Step 8:** A confirmation message is displayed. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Required field not entered** |  | **Step 6:** a required field is not entered  **Step 7**: The system displays an appropriate error message | |
| **Alternate Scenarios**  **DOB invalid** | **Actor Action System Response**  **Step 6:** DOB is in the future  **Step 7**: The system displays an  appropriate error message. | | |
| **DOB invalid - Format** | **Step 6:** DOB is in an invalid format  **Step 7:** The system displays an  appropriate error message. | | |
| **Conclusions** | A member’s file has been amended. | | |
| **Post conditions** | This staff may now use system functions | | |
| **Business Rules** | A staff member must exist in the staff file before amended. | | |
| **Implementation Constraints** |  | | |

## **List Staff**

## This is where all current staff members’ are listed by a manger.

Manager

## Use Case Narrative

|  |
| --- |
| **Staff** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **List Staff Members** | | **Author:** |
| **Use Case Id** | 03 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Manager/Staff | | |
| **Other Participating Actors** |  | | |
| **Description** | This function lists all current staff. | | |
| **Preconditions** |  | | |
| **Trigger** | A staff listing is required. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager invokes the list staff function.  **Step 3:** The admin orders the list by surname first. | **Step 2:** The system retrieves details of all staff from the staff file and displays the list in alphabetical order.  **Step 4:** The system displays the staff details.  **Step 5:** The system gives the admin the option print the staff member list. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Staff File list Empty** |  | **Step 2:**  The staff file does not contain any staff members.  **Step 3:** The system displays an error message suggesting that the administrator should be informed. | |
|  |  | | |
| **Conclusions** | A staff list is displayed. | | |
| **Post conditions** | The members cannot be amended/edited when viewed like this. | | |
| **Business Rules** | The members list should only contain current members. | | |
| **Implementation Constraints** |  | | |

## **Remove Staff Member**

## This is where a current staff member would be removed by a manager so that the staff file could be updated to the most recent point.

Staff

Manager

## Use Case Narrative

|  |
| --- |
| **Staff** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Remove Staff Member** | | **Author:** |
| **Use Case Id** | 04 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Manager/Staff | | |
| **Other Participating Actors** | Member | | |
| **Description** | This function cancels a request after making one. The system must retrieve the staff file and remove that member from the list. | | |
| **Preconditions** | A member must exist in the staff file. | | |
| **Trigger** | A staff file is requested in order to be able to remove that staff member. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager invokes the remove staff function.  **Step 3:** The admin chooses the staff member to be removed.  **Step 4:** The admin confirms, and the staff member is removed from the staff list. | **Step 2:** System retrieves summary of all staff details from staff file and displays in UI in order by Staff ID.  **Step 5:** The system updates the new staff list and the status is updated to ‘unregistered’. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **No Staff Details found** |  | **Step 2:** The system displays an error message suggesting that the administrator should be informed. | |
|  |  | | |
| **Conclusions** | A member is removed from the staff file list. | | |
| **Post conditions** |  | | |
| **Business Rules** |  | | |
| **Implementation Constraints** |  | | |

## **Process Orders**

This is the system requirement that makes the process for all the food ordered by customers which include recording the food order, editing the order, listing the order, dispatching the order, record payment and cancel an order.

## **Record Order**

This is where a staff member will record the food order made by a customer.

<<Include>>

Staff

Customer

The staff confirms no more food items are required and and ID is assigned

**N**

**Y**

More?

The system adds the total of the order in the basket and asks the user if there is more to be added.

Staff selects a food items and adds to basket

The system retrieves summary details of all food items from the Foods File and displays on the UI.

The staff starts the Record Order process.

Activity Diagram

Activity Diagram Cont.

The system prompts the staff member for customer details

The system validates the customer details

The staff enters the customer details

Display Error Message

**N**

Valid?

**Y**

The system sets the Order Date to the current system date and sets the Order status to ‘placed’.

The system displays a confirmation message.

The system saves the order details in the Orders file and saves details of each item ordered in the Order Items file.

## Use Case Narrative

|  |
| --- |
| **Orders** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Record Food Order** | | **Author:** |
| **Use Case Id** | 05 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** | Customer | | |
| **Description** | This function will record a customer’s food order. | | |
| **Preconditions** | The food ordered must exist the food’s file. | | |
| **Trigger** | A customer makes the food order. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The staff starts the Record Order process.  **Step 3:** Staff selects a food items and adds to basket  **Step 5:** The staff confirms no more food items are required & ID is assigned.  **Step 7:** The staff enters the customer details:   * Name * Address * phone No | **Step 2:** The system retrieves summary details of all food items from the Foods File and displays on the UI.  **Step 4:** The system adds the total of the order in the basket and asks the user if there is more to be added.  **Step 6:** The system prompts the staff member for customer details  **Step 8:** The system validates the customer details:   * All fields are required   **Step 9:** The system sets the Order Date to the current system date and sets the Order status to ‘placed’.  **Step 10** The system saves the order details in the Orders file and saves details of each item ordered in the Order Items file.  **Step 11:** Thesystem displays a confirmation message. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **The Food ordered, missing from the foods’ file.** |  | **Step 8:** A required field is not entered  **Step 5:** The system displays an error message | |
|  |  | | |
| **Conclusions** | A food order has been recorded. | | |
| **Post conditions** |  | | |
| **Business Rules** |  | | |
| **Implementation Constraints** |  | | |

## **Edit Food Order**

## This is where the staff amends/edits the food ordered by a customer.

<<Include>>

Staff

Customer

## Use Case Narrative

|  |
| --- |
| **Orders** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Edit Food Order** | | **Author:** |
| **Use Case Id** | 06 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** | Customer | | |
| **Description** | This function edits/amends the food order made by a customer. | | |
| **Preconditions** | The food ordered must exist the food’s file. | | |
| **Trigger** | A customer changes his/her mind about the order and wishes to change it. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager invokes the amend food order function.  **Step 3:** The staff request the food to be ordered from a customer.   * Main food * Side dip * Any extras * Drink   **Step 6:** The staff confirms and the order is saved in the orders files. | **Step 2:** The system display the UI.  **Step 4:** The system validates all the food entered.  **Step 5:** The system confirms all the food ordered and updates from the current order made.  **Step 7:** The system displays a confirmation message. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **The Food ordered, missing from the foods’ file.** |  | **Step 4:** The food ordered by the customer is missing from the foods’ file or it does not exist.  **Step 5:** The system displays an error message suggesting that the administrator should be informed. | |
|  |  | | |
| **Conclusions** | A food order has been amended and updated from the current order. | | |
| **Post conditions** |  | | |
| **Business Rules** | The orders taken in one day are only stored for 24 hours, meaning the file is refreshed every 24 hours with new data. All order from days before are stored in ‘all\_orders\_file’. | | |
| **Implementation Constraints** |  | | |

## **List Food Order**

## This is where a staff member lists the food order for reviewing purposes before confirming the order.

Staff

## Use Case Narrative

|  |
| --- |
| **Orders** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **List Food Order** | | **Author:** |
| **Use Case Id** | 07 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** |  | | |
| **Description** | This function lists the food order made by a customer. | | |
| **Preconditions** | The food ordered must exist the food’s file. | | |
| **Trigger** | A staff member chooses to list the food ordered by a customer. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The staff member invokes the list food order function.  **Step 3:** The staff member chooses to list the most recent order.  **Step 5:** The staff confirms and the order is displayed in default order.   * Date   **Step 6:** The staff member chooses to display the list in order by time. | **Step 2:** The system display the UI.  **Step 4:** The system retrieves the order from the orders file and displays the time and date of the order.  **Step 7:** The system displays the list in order by time. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Most recent order missing.** |  | **Step 4:** The food ordered by the customer is missing from the orders file or cannot retrieve it.  **Step 5:** The system displays an error message suggesting that the administrator should be informed. | |
|  |  | | |
| **Conclusions** | A most recent food order has been listed. | | |
| **Post conditions** |  | | |
| **Business Rules** | The orders taken in one day are only stored for 24 hours, meaning the file is refreshed every 24 hours with new data. All order from days before are stored in ‘all\_orders\_file’. | | |
| **Implementation Constraints** |  | | |

## **Dispatch Order**

## This is where a staff member dispatches an order made by customer.

## 

Staff

Customer

Activity Diagram

System

Staff

The system displays a confirmation message and marks the order as dispatched.

The system then retrieves the order and prints a customer and staff copy.

The staff member confirms the order to be dispatched and changes its status

The system retrieves the order and displays a dialog to confirm the order.

The staff request the food order to be dispatched.

## Use Case Narrative

|  |
| --- |
| **Orders** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Dispatch Order** | | **Author:** |
| **Use Case Id** | 08 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** | Customer | | |
| **Description** | This function dispatches a food order made by a customer. | | |
| **Preconditions** | The address given by the customer must be local (County). | | |
| **Trigger** | A staff member chooses to dispatch an order. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The staff member invokes the dispatch food order function on the amend order page.  **Step 4:** The staff member confirms the order to be dispatched and the status is changed to ‘dispatched’ in the orders file. | **Step 2:** The system display the UI.  **Step 3:** The system retrieves the order from the orders file and displays a dialog to confirm the order.  **Step 5:** The system displays a confirmation message. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **The system fails to retrieve the order.** |  | **Step 3:** The system displays an error message suggesting that the administrator should be informed. | |
|  |  | | |
| **Conclusions** | A food order is dispatched to the customers address. | | |
| **Post conditions** |  | | |
| **Business Rules** | The orders taken in one day are only stored for 24 hours, meaning the file is refreshed every 24 hours with new data. All order from days before are stored in ‘all\_orders\_file’. | | |
| **Implementation Constraints** |  | | |

## **Record Payment**

## This is where the system records the payment made by a customer for the order made.

Staff

Customer

## Use Case Narrative

|  |
| --- |
| **Orders** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Record Payment** | | **Author:** |
| **Use Case Id** | 09 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** | Customer | | |
| **Description** | This function records the payment into the system made by the customer. | | |
| **Preconditions** |  | | |
| **Trigger** | A staff member chooses the option to record the payment. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The staff member invokes the record payment function.  **Step 4:** The customer chooses to pay.  **Step 5:** The staff member is asked for:   * Total order Amount * Given amount by the customer   **Step 7:** The staff member confirms and the payment is made. | **Step 2:** The system display the UI.  **Step 3:** The system displays the UI for the staff to input the given details.  **Step 6:** The system calculates the total and returns the change due.  **Step 8:** The payment details are stored in the transactions file.  **Step 9:** The system displays a confirmation message. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Details Invalid.** |  | **Step 6:** The details inputted are invalid.  **Step 7:** The system asks the staff member if he/she wants to try again. | |
|  |  | | |
| **Conclusions** | A payment has been made and stored in the transactions file. | | |
| **Post conditions** |  | | |
| **Business Rules** | No payments to be made by card. | | |
| **Implementation Constraints** |  | | |

## **Cancel Order**

## This is where a staff member cancels an order made by a customer for various reasons.

## 

Staff

Customer

## Use Case Narrative

|  |
| --- |
| **Orders** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **List Food Order** | | **Author:** |
| **Use Case Id** | 10 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** | Customer | | |
| **Description** | This function cancels an ordered for various reasons made by the customer. | | |
| **Preconditions** |  | | |
| **Trigger** | A staff member chooses to cancel the food ordered made by the customer. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The staff member invokes the cancel order function.  **Step 4:** The staff member chooses the order from the list he/she wishes to cancel.  **Step 6:** The staff members confirms. | **Step 2:** The system display the UI.  **Step 3:** The system retrieves all orders from the orders file in the last 24 hours and displays the list by default order.   * Order ID   **Step 5:** The system confirms with the staff member to cancel the order.  **Step 7:** The system cancels the order and updates the list in the order file.  **Step 8:** The system displays a confirmation message. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **The list does not update.** |  | **Step 7:** The list in the orders file does not update when an order is cancelled and removed from the list. The order remains in the orders file.  **Step 8:** The system displays a message should the user delete the order manually, if not, an administrator should be informed. | |
|  |  | | |
| **Conclusions** | A most recent food order has been listed. | | |
| **Post conditions** |  | | |
| **Business Rules** | The orders taken in one day are only stored for 24 hours, meaning the file is refreshed every 24 hours with new data. All order from days before are stored in ‘all\_orders\_file’. | | |
| **Implementation Constraints** |  | | |

## **Manage Foods**

## This is where all the food that is sold at this restaurant is managed, these include adding new food to the system, amending/editing a current food item, removing a current food item and listing all food items that are currently in the system.

## **Add Food**

## This is where new food items which will be sold at the restaurant are added to the system.

Manager

## Use Case Narrative

|  |
| --- |
| **Food** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Add Food** | | **Author:** |
| **Use Case Id** | 11 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** |  | | |
| **Description** | This function adds new food items to the system which will be sold at the restaurant. | | |
| **Preconditions** |  | | |
| **Trigger** | The manager chooses to add new food to the system. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager invokes the add food function.  **Step 4:** The manager enters the details for the new added food item.  **Step 7:** The manager confirms with the system, and that he/she’s happy with the details entered. | **Step 2:** The system display the UI.  **Step 3:** The system prompts the user to enter the details of the new food item.   * StockNo * Name * Description * Ingredients * Category * Price   **Step 5:** The system validates all the details given.   * All fields required * Price in euros * Price field must have two digits after the decimal point.   **Step 6:** The system displays a confirmation message to check if the user is done.  **Step 8:** The system displays a confirmation message and saves the new food item in the food items file. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Empty field** |  | **Step 5:** The user has left a field empty.  **Step 6:** The system displays an error message stating that a field has been left empty and some data should be entered. | |
|  |  | | |
| **Conclusions** | A new food items has been added to the system. | | |
| **Post conditions** |  | | |
| **Business Rules** | All food added to the system must go through inspection by the restaurant testing team. | | |
| **Implementation Constraints** |  | | |

## **Amend Food**

## This is where a current food item is edited/amended by a manager.

Manager

## Use Case Narrative

|  |
| --- |
| **Food** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Amend Food** | | **Author:** |
| **Use Case Id** | 12 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** |  | | |
| **Description** | This function amends a current food item. | | |
| **Preconditions** | The food item must exist in the food items file. | | |
| **Trigger** | The manager chooses to amend a current food item in the system. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager invokes the amend food function.  **Step 3:** The manager chooses the food item to be amended.  **Step 6:** The manager confirms with the system, and that he/she’s happy with the details entered. | **Step 2:** The system display the UI.  **Step 4:** The system prompts the user to re-enter the details of the food item.   * Name * Description * Ingredients * Category * Price   **Step 5:** The system validates all the details given.   * All fields required * Price in euros * Price field must have two digits after the decimal point.   **Step 7:** The system displays a confirmation message to check if the user is done.  **Step 8:** The system displays a confirmation message and overwrites the existing detail of a food item with the new details entered. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Empty field** |  | **Step 5:** The user has left a field empty.  **Step 6:** The system displays an error message stating that a field has been left empty and some data should be entered. | |
|  |  | | |
| **Conclusions** | A current food items has been amended. | | |
| **Post conditions** |  | | |
| **Business Rules** | All food added to the system must go through inspection by the restaurant testing team. | | |
| **Implementation Constraints** |  | | |

## **Remove Food**

## This is where a current food item is removed from the system.

Manager

## Use Case Narrative

|  |
| --- |
| **Food** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Remove Food** | | **Author:** |
| **Use Case Id** | 13 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** |  | | |
| **Description** | This function removes a current food item. | | |
| **Preconditions** | The food item must exist in the food items file. | | |
| **Trigger** | The manager chooses to remove a current food item in the system. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager invokes the remove food function.  **Step 3:** The manager chooses the food item to be removed.  **Step 5:** The manager confirms with the system. | **Step 2:** The system display the UI.  **Step 4:** The system displays a confirmation message.  **Step 6:** The system removes the food item from the list.  **Step 7:** The system updates the food items file.  **Step 8:** The system displays a confirmation message that a food item has been removed. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Food items file not updated** |  | **Step 5:** After a food item has been removed, the list of food items in the food items file did not update.  **Step 6:** The system displays an error message suggesting that the administrator should be informed. | |
|  |  | | |
| **Conclusions** | A current food items has been removed from the system. | | |
| **Post conditions** |  | | |
| **Business Rules** | All food added to the system must go through inspection by the restaurant testing team. | | |
| **Implementation Constraints** |  | | |

## **List Food Items**

## This is where the user lists all food items that are currently in the system.

Manager

## Use Case Narrative

|  |
| --- |
| **Food** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **List Food** | | **Author:** |
| **Use Case Id** | 14 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** |  | | |
| **Description** | This function lists all current food items. | | |
| **Preconditions** | The food item must exist in the food items file. | | |
| **Trigger** | The manager chooses to list all current food items in the system. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager invokes the list food function.  **Step 4:** The manager chooses to list the items in order by:   * Price low to high   **Step 6:** The manager confirms. | **Step 2:** The system display the UI.  **Step 3:** The system lists all food items in default order.   * Price high to low   **Step 5:** The system prompts the user if he/she wants to print the list.  **Step 7:** The system sends the list to the default printing device and prints the list.  **Step 8:** The system displays a confirmation message that the list has been printed. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **The system does send the list to the print device.** |  | **Step 5:** After the manager confirms that he/she wishes to have the list printed, the system does not send the list document to the default printing device.  **Step 6:** The system displays an error message suggesting that the administrator should be informed. | |
|  |  | | |
| **Conclusions** | A listing of all current food items are displayed. | | |
| **Post conditions** | The food items cannot be amended or removed when viewed this way. | | |
| **Business Rules** | All food added to the system must go through inspection by the restaurant testing team. | | |
| **Implementation Constraints** |  | | |

## **Manage Reports**

## This is where all the restaurant reports will be managed, these include viewing the sales analysis, customer analysis and editing both reports.

## **View sales analysis**

## This is where the sales analysis can be viewed by the manager only.

Manager

## Use Case Narrative

|  |
| --- |
| **Reports** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **View Sales Analysis** | | **Author:** |
| **Use Case Id** | 15 | | **Date:** |
| **Priority** |  | | |
| **Source** | Requirements | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** |  | | |
| **Description** | This function views the restaurants’ sales analysis. | | |
| **Preconditions** |  | | |
| **Trigger** | The manager chooses to view the sales analysis. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager invokes the view sales analysis function.  **Step 4:** The manager chooses to list the analysis in order by:   * Payment Date   **Step 6:** The manager confirms. | **Step 2:** The system display the UI.  **Step 3:** The system lists the sales analysis in default order.   * Payment Date   **Step 5:** The system prompts the user if he/she wants to print the sales analysis document.  **Step 7:** The system sends the list to the default printing device and prints the document.  **Step 8:** The system displays a confirmation message that the document has been printed. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **The system does send the document to the print device.** |  | **Step 5:** After the manager confirms that he/she wishes to have the document printed, the system does not send the document to the default printing device.  **Step 6:** The system displays an error message suggesting that the administrator should be informed. | |
|  |  | | |
| **Conclusions** | A sales analysis report is displayed. | | |
| **Post conditions** |  | | |
| **Business Rules** | The sales analysis must be daily updated by the manager. | | |
| **Implementation Constraints** |  | | |

# **System Model**

External Entities

Customer

Internal Entities

Manager

Data Stores

D1 Orders File

D2 Staff File

D3 Foods File

D4 Reports File

D5 Payments File

D6 OrderItems File

D7 Customers File

Processes

*P1 Process Orders*

P1.1 Record Order

P1.2 Amend Order

P1.3 List Order

P1.4 Dispatch Order

P1.5 Record Payment

P1.6 Cancel Order

*P2 Process Foods*

P2.1 Add Food

P2.2 Amend Food

P2.3 Remove Food

P2.4 List Food

*P3 Process Reports*

P3.1 View Sales Analysis

*P4 Process Staff*

P4.1 Register Staff

P4.2 Amend Staff

P4.3 List staff

P4.4 Remove Staff

## 

**MANAGER**

MNGR\_NO (pk)

Bonus

**STAFF**

Staff\_No (pk)

Surname

Forename

D.O.B

E-mail

Phone No.

Reg\_Date

Is a

**RESTUARANT**

Rest\_No (pk)

Rname

Tel\_No

Address

1

1..3

**OWNER**

Owner\_No (pk)

Surname

Forename

Phone\_No

E-mail

0..\*

1

Allocated

1

1..\*

Manages

Owned By

**ORDERS**

Order\_No (pk)

Ord\_Items\_Food

Address

Ord\_Date

Ord\_Time

Price

Disptach\_No

1

1..\*

Made By

**REPORTS**

Report\_No (pk)

Rep\_Date

1..\*

1

Is For

**FOOD**

Food\_No (pk)

Stock\_Amount

Date\_Purch

1..\*

1

Is For

# **Data Model**

Level 0 - DFD

Customer

Fast Food Delivery System

Order Request

Order

Level 1 – DFD

D5 Payments File

D6 Customers File

Store

Record Payment

P4

Process Staff

Customer

Order/Cust. Details

Store

D6 OrderItems File

Food retrieved

D3 Foods File

Check Availability

Report Details

Foods Details

Staff Details

D4 Reports File

D2 Staff File

P2

Process Foods

Order Details

P3

Process Reports

Record Order

D1 Orders File

P1

Process Orders

Level 2 - DFD Process P1

P1.6

Cancel Order

Store

D6 OrderItems File

Assign Order as Dispatched

P1.4

Dispatch Order

Delete Order Details

P1.1

Record Order

D1 Orders File

Order Details

P1.2

Amend Order

Order Details

New Order Details

P1.3

List Order

Order Details

P1.5

Record Payment

Customer

Make Payment

D5 Payments File

Record Payment

Level 2 - DFD Process P2

List Food Details

P2.4

List Food Items

Delete Food Details

P2.3

Remove Food

New Food Details

Food Details

P2.2

Amend Food

Food Details

D3 Foods File

P2.1

Add Food

Level 2 - DFD Process P3

D6 OrderItems File

Retrieve Details

Manages Details

Manager

Sales Analysis Details

D4 Reports File

P3.1

View Sales Analysis

Level 2 - DFD Process P4

Delete Staff Member

P4.4

Remove Staff

Staff Details

P4.3

List Staff

New Staff Details

Staff Details

P4.2

Amend Staff

Staff Details

D2 Staff File

P4.1

Register Staff

# **Database Schema**

## **Schema: Fast Food Delivery**

**Relation: Staff**

Attributes:

Staff\_No number(6)

Surname char(15) NOT NULL

Forename char(15) NOT NULL

DOB date

Email char(20)

PhoneNo numeric(11)

Status char(4)

Primary key(Staff\_No)

**Relation: Orders**

Attributes:

Order\_No number(6) NOT NULL

MainDish char(15) NOT NULL

SideDish char(15) NOT NULL

SideDip char(15) NOT NULL

Drink char(15) NOT NULL

Dispatch char(4) NOT NULL

Primary key(Order\_No)

**Relation: Stock**

Attributes:

Stock\_No number(6) NOT NULL

Title char(15) NOT NULL

Description char(15) NOT NULL

Ingridients char(15) NOT NULL

Category char(5) NOT NULL

Price number(6) NOT NULL

Availability char(5) NOT NULL

Primary key(Stock\_No)

**Relation: Payments**

Attributes:

PaymentID numeric(4) NOT NULL

OrderIdPay numeric(4)

Total numeric(4) NOT NULL

Given numeric(4)

ChangeDue numeric(4)

currentDate date NOT NULL

Primary key(Payment\_No

**Relation: Order Items**

Attributes:

OrderID numeric(6) NOT NULL

StockNo numeric(6) NOT NULL

Quantity numeric(4) NOT NULL

Price numeric(4,2) NOT NULL

Primary key(Report\_No

# **Program Specifications**

# **Conclusion**

All functional requirements, flowchart, user and system requirements, DFD’s, data and system model, and the database schema are clearly outlined. The program has the functionality related to current and new staff members, orders taken by customers, which it also keeps a record of, has the functionality to add new food types to the software, increase its stock level etc. and keep a record of all report analysis which may be used by the manager to see the restaurants’ profits and also its losses.

Going through all the functionality of the program, my suggestion where improvement can be made would be to add an extra functionality that keeps a record of all the drivers used to deliver food orders to different customers in order to see how much a certain driver can get deliveries done in a certain amount of time etc. This information may come in use by the manager as he can then tell which driver is best to work on certain days e.g. busy and quiet days.

# **References:**