Requirements Engineering

Take-Away System

Submitted By: Krystian Lewandowski (T00212237)

Computing with Software Development

Date Submitted: dd/mm/yyyy

**Table of Contents**

[1. Introduction/overview 4](#_Toc55406478)

[2. Functional Components 5](#_Toc55406479)

[3. User Requirements 6](#_Toc55406480)

[3.1. Take-Away System will manage the Menu 6](#_Toc55406481)

[3.2. Take-Away System will process orders 6](#_Toc55406482)

[3.3. Take-Away System will perform administrative reporting 6](#_Toc55406483)

[4. System Requirements 7](#_Toc55406484)

[4.1. System Level Use Case Diagram 7](#_Toc55406485)

[4.2. Manage Menu 8](#_Toc55406486)

[4.2.1. Add Menu Item 8](#_Toc55406487)

[4.2.2. Remove Menu Item 11](#_Toc55406488)

[4.2.3. Update Menu Item 14](#_Toc55406489)

[4.2.4. View Menu 17](#_Toc55406490)

[4.3. Manage Orders 19](#_Toc55406491)

[4.3.1. Make order 19](#_Toc55406492)

[4.3.2. Cancel Order 22](#_Toc55406493)

[4.3.3. Update Order 24](#_Toc55406494)

[4.4. Perform Admin 26](#_Toc55406495)

[4.4.1. Set Opening Time 26](#_Toc55406496)

[4.4.2. Yearly Item Analysis 28](#_Toc55406497)

[4.4.3. Yearly Revenue Analysis 30](#_Toc55406498)

[5. System Model 33](#_Toc55406499)

[5.1. Level-0 DFD 33](#_Toc55406500)

[5.2. Level-1 DFD 33](#_Toc55406501)

[5.3. Level-2 DFD (Process P1: Title) 33](#_Toc55406502)

[5.4. Level-2 DFD (Process P2: Title) 33](#_Toc55406503)

[5.5. Level-2 DFD (Process P3: Title) 33](#_Toc55406504)

[6. Data Model (Class Diagram) 34](#_Toc55406505)

[6.1. Class Diagram 34](#_Toc55406506)

[6.2. Relational Schema 34](#_Toc55406507)

[6.3. Database Schema 34](#_Toc55406508)

[7. Conclusion 35](#_Toc55406509)

[8. Appendices 36](#_Toc55406510)

[8.1. Appendix A – Title 36](#_Toc55406511)

[8.2. Appendix B – Title 36](#_Toc55406512)

# Introduction/overview

**Leave this section until last……**

# Functional Components

This section presents the functional components of the propsoed software system. Give a very brief overview of the logical structure of the software…. 2-3 sentences

# User Requirements

This section describes the user requirements (functional components) as high-level abstract statements.

## Take-Away System will manage the Menu

* + 1. Take-Away System will add a food item
    2. Take-Away System will update food item
    3. Take-Away System will remove a food item
    4. Take-Away System will display the menu

## Take-Away System will process orders

* + 1. Take-Away System will add orders
    2. Take-Away System will remove orders
    3. Take-Away System will update orders

## Take-Away System will perform administrative reporting

* + 1. Take-Away System will set opening times
    2. Take-Away System will produce an Item Analysis chart
    3. Take-Away System will produce a Yearly Revenue Analysis chart

# System Requirements

The low-level system requirements of the system are presented in this section of the document.

## System Level Use Case Diagram

The following system level use case diagram illustrates the high-level system requirements.

Manager

Sales Clerk

Customer

## Manage Menu

### Add Menu Item

All food items available to the customers must be recorded in the system.

This function adds an item to the menu.

Use Case Diagram

Manager

<<includes>>

<<extends>>

Activity Diagram

Yes

Initialize UI

Input Menu Data

**Valid Data?**

No

Display Error Message

Save Menu Data

Display Confirmation Message

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Menu Item** | |
| **Use Case Id** | 001 | |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function adds a new item to the menu | |
| **Preconditions** |  | |
| **Trigger** | Add Menu Item option selected from main menu | |
| **Expected Scenario** | **Manager Action** | **System Response** |
|  | **Step 3:** Input the following data:   * Item\_Name String(30) * Item\_Description String (60) * Item\_Price numeric(4,2) | **Step 1:** Assign next item Id  **Step 2:** Display the UI to add a menu Item.  **Step 4:** Validate data:   * All fields entered * Price greater than 0 * Item name is unique * Item name only letters   **Step 5:** Set Item\_Status to Available ‘A’  **Step 6:** Save data to database:   * Item\_Id * Item\_Name * Item\_Description * Item\_Price * Item\_Status   **Step 7:** Display Confirmation message  **Step 8:** Reset UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Data Input** |  | **Step 4:** Display Appropriate Error Message e.g. “Invalid Price Entered”  **Step 5:** Return to Step 3 |
| **Conclusions** | An item is added to the menu | |
| **Post conditions** |  | |
| **Business Rules** | Item Name must only contain letters | |
| **Implementation Constraints** | Data Entered Must Be Unique | |

### Remove Menu Item

Use Case Diagram

Manager

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Remove Menu Item** | |
| **Use Case Id** | 002 | |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function removes an item from the menu | |
| **Preconditions** |  | |
| **Trigger** | Remove Menu Item option selected from main menu | |
| **Expected Scenario** | **Manager Action** | **System Response** |
|  | **Step 3:** Select Item to remove  **Step 5:** Confirm selection | **Step 1:** Retrieve menuItems from database  **Step 2:** Display UI.  **Step 4:** Ask user for confirmation    **Step 6:** Remove the Item from Menu  **Step 7:** Display Confirmation message.  **Step 8:** Reset UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid data entered** |  | **Step 7:** Display Appropriate Error Message e.g. “Invalid Price Entered”  **Step 8:** Return to Step 6 |
| **Conclusions** | Item detail is removed from the menu | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** | At least one item must be in the database | |

### Update Menu Item

Use Case Diagram

Manager

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update Menu Item** | |
| **Use Case Id** | 003 | |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function updates an item in the menu | |
| **Preconditions** |  | |
| **Trigger** | Update Menu Item option selected from main menu | |
| **Expected Scenario** | **Manager Action** | **System Response** |
|  | **Step 3:** Select Item to change  **Step 5:** Confirm selection  **Step 6:** Input the following data:   * Item\_Name String(30) * Item\_Description String (60) * Item\_Price numeric(4,2) | **Step 1:** Retrieve items from database  **Step 2:** Display the UI to change a menu Item.  **Step 4:** Ask user for confirmation  **Step 7:** Validate data:   * All fields entered * Price greater than 0 * Item name is unique * Item name only letters   **Step 8:** Alter data in database:   * Item\_Id * Item\_Name * Item\_Description * Item\_Price   **Step 9:** Display Confirmation message.  **Step 10:** Reset UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid data entered** |  | **Step 7:** Display Appropriate Error Message e.g. “Invalid Price Entered”  **Step 8:** Return to Step 6 |
| **Conclusions** | Item detail is changed in the menu | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** | At least one item must be in the database | |

### View Menu

This function will display the menu to the user.

Use Case Diagram

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **View Menu** | |
| **Use Case Id** | 004 | |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function displays the menu.  This is to allow the manager to inspect that everything on the menu is displaying as it should and no errors were entered | |
| **Preconditions** |  | |
| **Trigger** | View Menu option selected from main menu | |
| **Expected Scenario** | **Manager Action** | **System Response** |
|  | **Step 3:** View Items  **Step 4:** Interact with menu | **Step 1:** Retrieve items from database  **Step 2:** Display UI.  **Step 5:** Show Confirmation messages.  **Step 6:** Reset UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The menu has been verified by the manager | |
| **Post conditions** |  | |
| **Business Rules** | Manager must verify menu before it is displayed to customers | |
| **Implementation Constraints** | At least one item must be present in the menu | |

## Manage Orders

### Make order

Use Case diagram

Customer

Sales Clerk

<<includes>>

<<includes>>

<<extends>>

Activity Diagram

Retrieve Items from Database

Make Order

**Confirm Order**

No

Clear Order Details

Yes

Add Order

Display Confirmation Message

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Make Order** | |
| **Use Case Id** | 005 | |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Sales Clerk | |
| **Other Participating Actors** | Customer | |
| **Description** | This function adds a new order | |
| **Preconditions** |  | |
| **Trigger** | Make Order option selected from main menu | |
| **Expected Scenario** | **Sales Clerk Action** | **System Response** |
|  | **Step 4:** Input Customer’s order and details:   * Item * Amount * Name String(40) * Phone No String (10)   **Step 6:** Confirm Order | **Step 1:** Retrieve menu data  **Step 2:** Generate Order ID numeric(4)  **Step 3:** Display UI  **Step 5:** Ask user for confirmation  **Step 7:** Save Data:   * Order ID * Item * Amount * Name * Phone No   **Step 8:** Display Confirmation Message  **Step 9:** Reset UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Unconfirmed Order** |  | **Step 6:** Display Message e.g. “Order Cancelled”  **Step 7:** Return to step 4 |
| **Conclusions** | An Order is added. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Cancel Order

This function will remove an order from the orders file.

Use Case Diagram

Customer

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Cancel Order** | |
| **Use Case Id** | 006 | |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Customer | |
| **Other Participating Actors** |  | |
| **Description** | This function removes an order | |
| **Preconditions** | Customer has made an order | |
| **Trigger** | Cancel Order option selected from main menu | |
| **Expected Scenario** | **Customer Action** | **System Response** |
|  | **Step 3:** Input following Data:   * Order ID * Name * Phone No | **Step 1:** Retrieve orders  **Step 2:** Initialise UI  **Step 4:** Validate Data  **Step 5:** Remove Order from database  **Step 6:** Output confirmation message  **Step 7:** Reset UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Data Entered** |  | **Step 5:** Display Message e.g. “Order Does not exist”  **Step 6:** Return to step 3. |
| **Conclusions** | An Order is removed. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** | At least one order must be present. | |

### Update Order

This function will alter an existing order

Use Case Diagram

Customer

<<extends>>

Sales Clerk

<<includes>>

<<includes>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update Order** | |
| **Use Case Id** | 007 | |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Customer | |
| **Other Participating Actors** |  | |
| **Description** | This function alters an order.  First the user has to input their information then an option to amend the order is given | |
| **Preconditions** | Customer has made an order | |
| **Trigger** | Update Order option selected from main menu | |
| **Expected Scenario** | **Customer Action** | **System Response** |
|  | **Step 3:** Input following Data:   * Order ID * Name   **Step 5:** Change order details e.g.   * Add items * Change phone number | **Step 1:** Retrieve orders  **Step 2:** Initialise UI  **Step 4:** Validate Data  **Step 6:** Validate Order Data  **Step 7:** Save changes in database  **Step 8:** Output confirmation message  **Step 9:** Reset UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Data Entered** |  | **Step 5:** Display Message e.g. “Order Does not exist”  **Step 6:** Return to step 3. |
| **Conclusions** | An Order is updated in the Orders File. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** | At least one order must be present in the orders file | |

## Perform Admin

### Set Opening Time

This function will allow the manager to set the times between which orders will be processed.

Use Case Diagram

Manager

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Set Opening Time** | |
| **Use Case Id** | 008 | |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function sets the times for which the system will process orders | |
| **Preconditions** |  | |
| **Trigger** | Set Opening Time option selected from main menu | |
| **Expected Scenario** | **Manager Action** | **System Response** |
|  | **Step 2:** Input the following data:   * Opening Time * Closing Time | **Step 1:** Initialise UI  **Step 3:** Validate Times  **Step 4:** Save times  **Step 5:** Display confirmation message  **Step 6:** Reset UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Time entered** |  | **Step 4:** Display Error Message e.g. “Opening time after Closing time”  **Step 5:** Return to Step 2 |
| **Conclusions** | The system will process orders between the specified times. | |
| **Post conditions** |  | |
| **Business Rules** | Last order to be processed at 11p.m. at the latest. | |
| **Implementation Constraints** |  | |

### Yearly Item Analysis

This function will output the top five most popular items on the menu and the bottom five items on the menu.

Use Case Diagram

Manager

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Yearly Item Analysis** | |
| **Use Case Id** | 009 | |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function will output item statistics based on item sales | |
| **Preconditions** |  | |
| **Trigger** | Yearly Item Analysis option selected from main menu | |
| **Expected Scenario** | **Manager Action** | **System Response** |
|  | **Step 2:** Input year for analysis  **Step 5:** Click exit button | **Step 1:** Initialise UI  **Step 3:** Validate Year  **Step 4:** Output following data:   * 5 most popular items along with their price and quantity sold * 5 least popular items along with their price and quantity sold   **Step 6:** Show confirmation message  **Step 7:** Reset UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Year entered** |  | **Step 4:** Display Error Message e.g. “You can only view data from the last 5 years onsite.”  **Step 5:** Return to Step 2 |
| **Conclusions** | The system outputs item data for the specified year | |
| **Post conditions** |  | |
| **Business Rules** | Data older than 5 years is to be stored in a separate archive database offsite and removed from current database | |
| **Implementation Constraints** |  | |

### Yearly Revenue Analysis

This function will display the income for a year using a bar chart, showing the profits per month.

Use Case Diagram

Manager

<<includes>>

<<extends>>

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Yearly Revenue Analysis** | |
| **Use Case Id** | 010 | |
| **Priority** |  | |
| **Source** |  | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function will output a bar chart with the proceeds per month | |
| **Preconditions** |  | |
| **Trigger** | Yearly Item Analysis option selected from main menu | |
| **Expected Scenario** | **Manager Action** | **System Response** |
|  | **Step 2:** Input year for analysis  **Step 6:** Click exit button | **Step 1:** Initialise UI  **Step 3:** Validate Year  **Step 4:** Retrieve Sales Data from Sales Table  **Step 5:** Output bar chart \*Appendix A  **Step 7:** Show confirmation message  **Step 8:** Reset UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Year entered** |  | **Step 4:** Display Error Message e.g. “You can only view data from the last 5 years onsite.”  **Step 5:** Return to Step 2 |
| **Conclusions** | The system outputs revenue data for the specified year | |
| **Post conditions** |  | |
| **Business Rules** | Data older than 5 years is to be stored in a separate archive database offsite and removed from current database | |
| **Implementation Constraints** |  | |

# System Model

The following dataflow diagrams have been produced for the system:

## Level-0 DFD

Take-Away

System

Customer

Make Order

Process Order

## Level-1 DFD

## Level-2 DFD (Process P1: Title)

## Level-2 DFD (Process P2: Title)

## Level-2 DFD (Process P3: Title)

# Data Model (Class Diagram)

Brief introduction……

## Class Diagram

Include class diagram / Object Model – UML Class Diagram here

Class diagram shows objects & attributes (NO METHODS)

## Relational Schema

Relational schema for the data requirements - Using ***bracket notation***

## Database Schema

A definition of the database to be implemented.

This includes primary key, foreign key and other constraints to be implemented.

# Conclusion

# Appendices

## Appendix A – Sales Bar Chart

## Appendix B – Title

Might include:

* **Lookup / Reference tables**
* **Sample reports / Listings**