

**GROUP ASSIGNMENT**

**TECHNOLOGY PARK MALAYSIA**

**AAPP010-4-2-PWP**

**PROGRAMMING WITH PYTHON**

**UCDF2007ICT(ITR), UCDF2007ICT(DI), UCDF2007ICT(SE), UCDF2007ICT, UCDF2007BIT**

**HAND OUT DATE: 02nd JUNE 2021**

**HAND IN DATE: 13TH AUGUST 2021**

**WEIGHTAGE: 100%**

**PREPARED BY:**

1. **YUDHISHTHRA A/L S SUGUMARAN (TP061762)**
2. **CHOONG WEI JUN (TP061867)**

**INSTRUCTIONS TO CANDIDATES:**

1. Submit your assignment online in MS Teams unless advised otherwise
2. Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld
3. Cases of plagiarism will be penalized
4. You must obtain at least 50% in each component to pass this module

Table of Contents

[1. Introduction and Assumptions 2](#_Toc79382182)

[1.1 Assumptions 3](#_Toc79382183)

[2. Program Design 3](#_Toc79382184)

[2.1 Pseudocode 3](#_Toc79382185)

[2.1.1 Utility Functions 3](#_Toc79382186)

[2.1.2 Admin Dashboard Functions 4](#_Toc79382187)

[2.2 Flowchart 8](#_Toc79382188)

[3. Program source code with explanation 9](#_Toc79382189)

[4. Additional features source code with explanation 9](#_Toc79382190)

[5. Screenshots of sample input/output with explanation 10](#_Toc79382191)

[6. Conclusion 10](#_Toc79382192)

[References 11](#_Toc79382193)

# 1. Introduction and Assumptions

The food delivery sector in Malaysia exponentially increased in popularity since the COVID-19 pandemic started. This meant that existing online food services had to rise to the occasion by improving their user interface to cater to the increasing demands and be able to ensure a seamless online food purchasing experience. For this purpose, Spiderman Online Food Service (SOFS) decided to improve their online food ordering service with one main goal in mind, that is for registered customers to be able to include various types of dishes in their order and checkout, thus reducing the time taken to order food.

Spiderman Online Food Services (SOFS) has an Online Food Order Management System that essentially provides

**Reference**

* Due to large number of customers demanding ease to use and fast rental service of cars, SUPER CAR RENTAL SERVICES (SCRS) quoted us to develop python program for online car rental system to enhance effectiveness and efficiency of customers ordering and booking cars for rent. Our program has 3 types of users as requested by SCRS, which are admin, all customer, and registered customers. We assume that “all customers” must use username and password as per global standardization to create an account as “registered customer”. We also assume that registered customers’ personal details are username and password, they can modify their personal details which are username and password. We also assume that “registered customer” can view detail of cars to be rented out, which are car rental price and car name (brand & model) while “all customer” can only view cars available for rent. Next, for the admins, we assume that they can modify the details of each cars (brand, name, description, hourly price, daily price, and remaining count). They can also return a specific car that is rented by a customer. Admins can also look at records of a specific customer, for example, their rental and booking history. Lastly, we assume that the “select and book a car for a specific duration” section is our “shopping cart”. Customers can perform payment to confirm booking that is being included in the “shopping cart” and we let customers confirm their payment via a binary question (yes/no).
* The Super Car Rental Services (SCRS) provides services on online car rental within Malaysia. An online car rental system (OCRS) is planned and developed aiming to implement the service operations involving all customers and administrators. It enables customers to book their desired vehicle with upfront payment while the admins handle the transactions and record statement through the online system. The OCRS is developed to monitor the vehicles, customers, and admins virtually. Booking and payment statements are displayed with specific status and recorded with details which had simplify the business process. The OCRS is centralized based so that the portal is accessible in any locations and anytime. Besides, the system improves the coordination between staffs because redundant data can be avoided if the data is stored and retrieved from the same database files. With the use of the OCRS, the confidential customer’s details can be kept secured with fixed integrity and renting progress will be more efficient and automated.

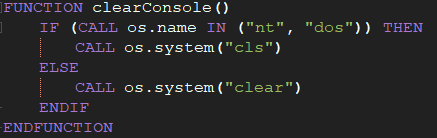
## 1.1 Assumptions

# 2. Program Design

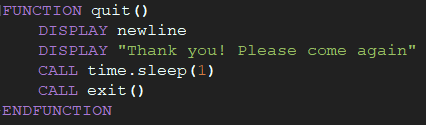
## 2.1 Pseudocode

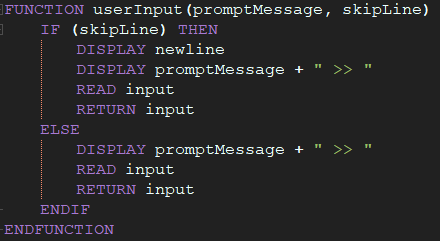
### 2.1.1 Utility Functions

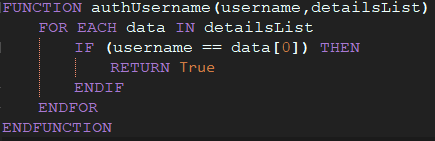
**clearConsole**

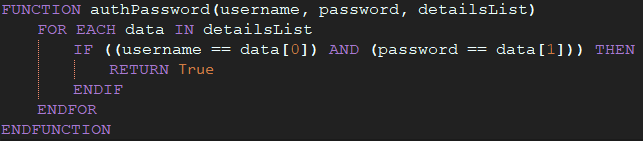
****

**quit**

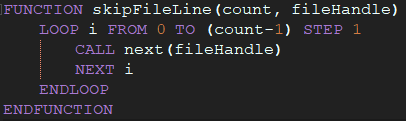
****

**userInput**

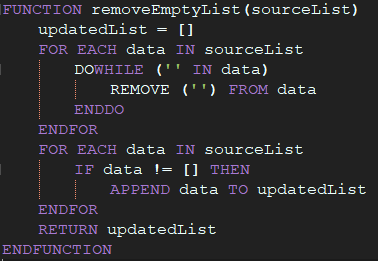
**authUsername**

**authPassword**

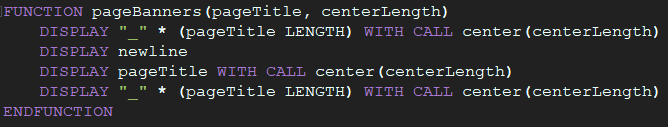
**skipFileLine**

****

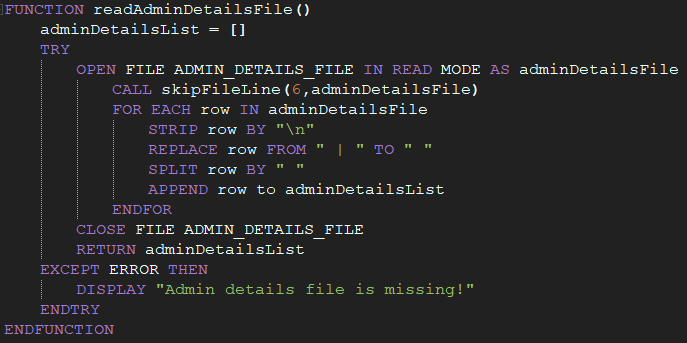
**removeEmptyList**

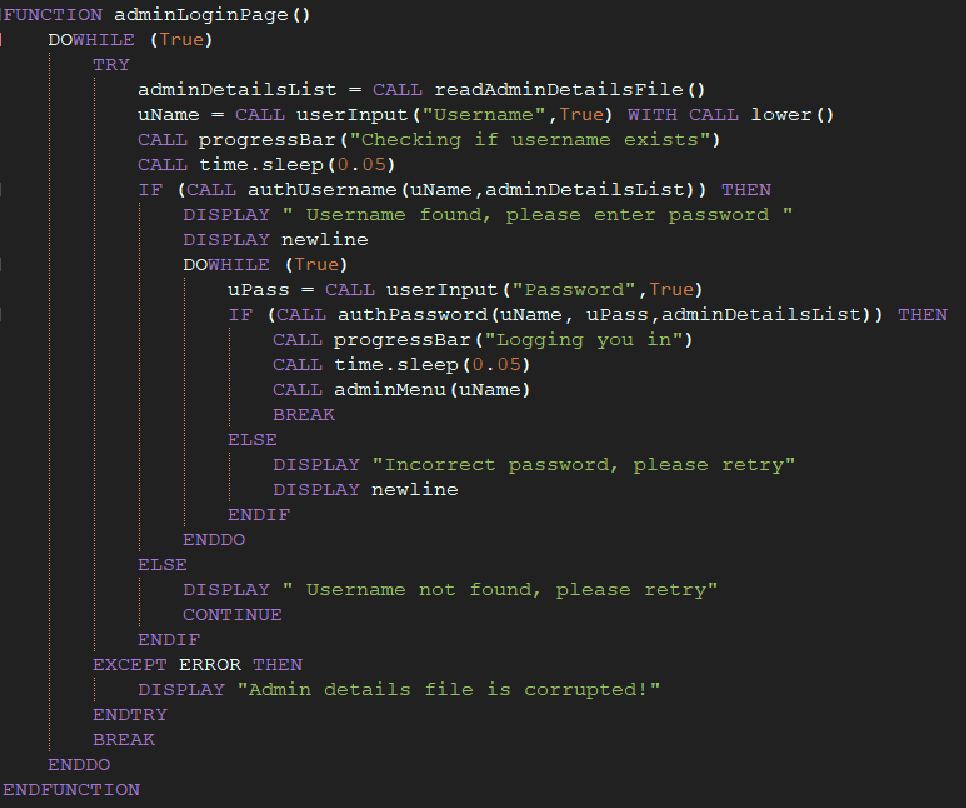
****

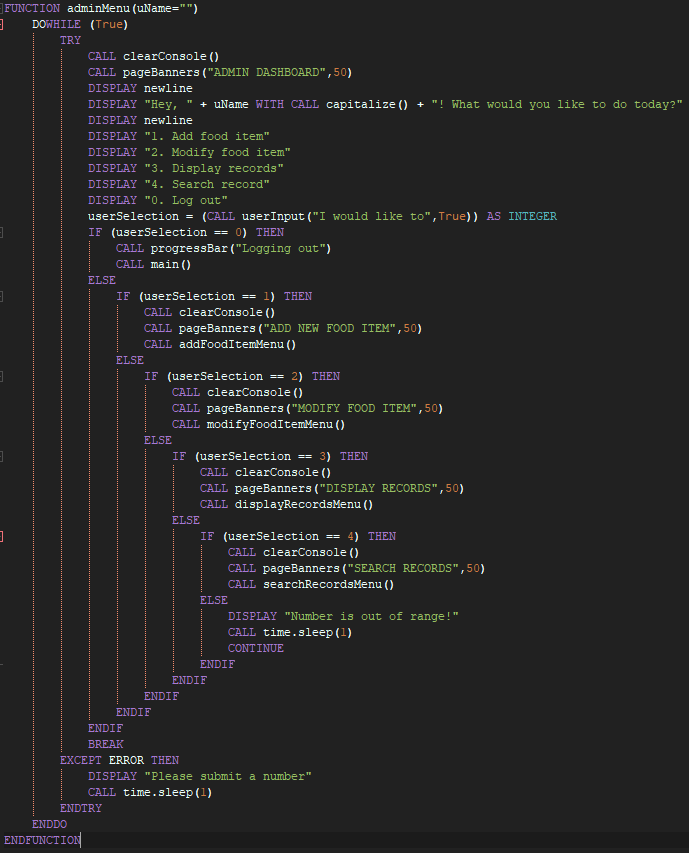
**progressBar**

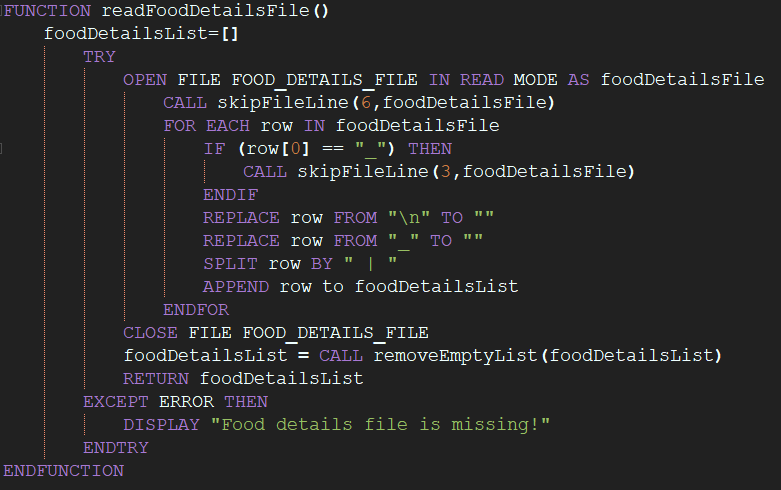
**pageBanners**

### 2.1.2 Admin Dashboard Functions

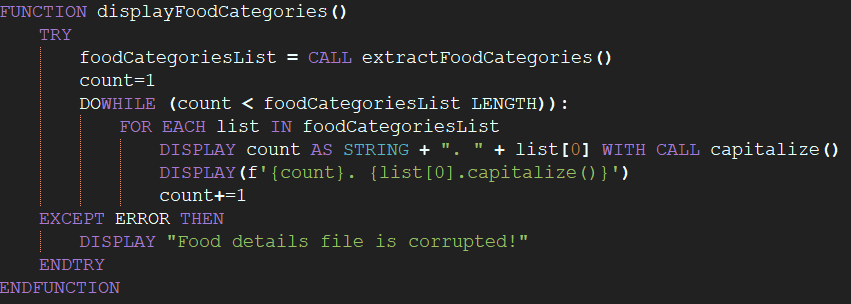
**readAdminDetailsFile**

**adminLoginPage **

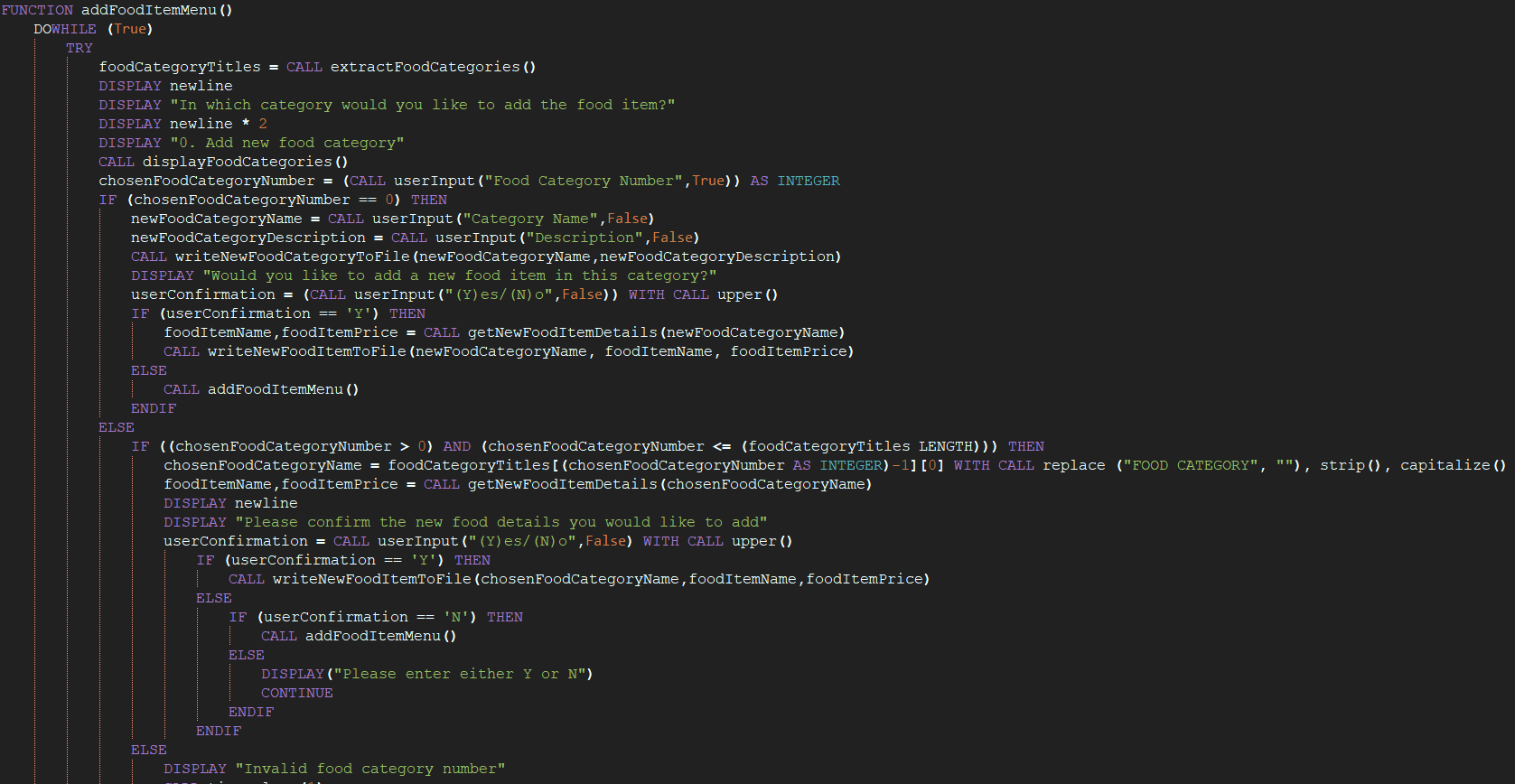
**adminMenu**

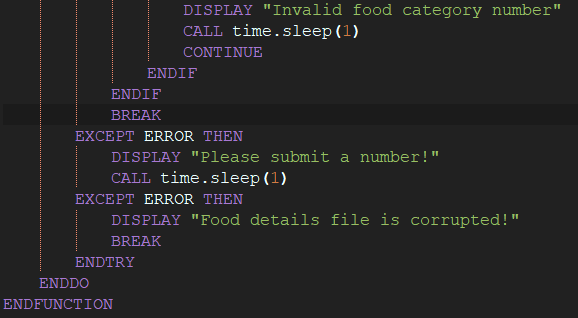
**2.1.2.4 readFoodDetailsFile**

**displayFoodCategories**

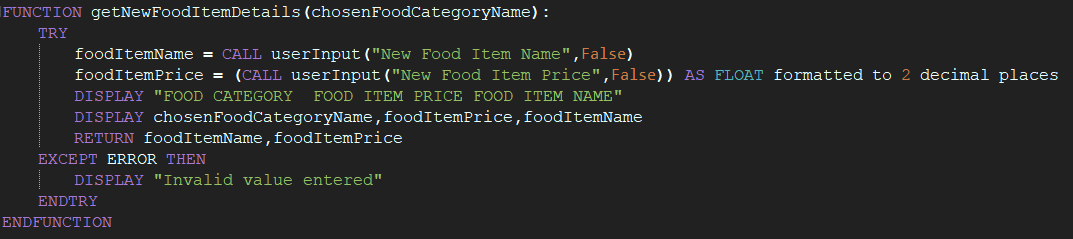
****

**addFoodItemMenu**

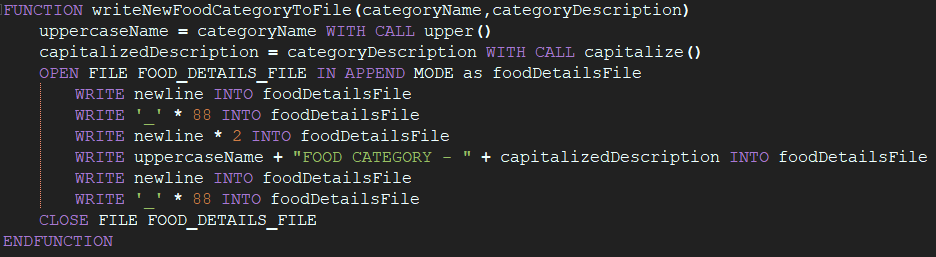
****



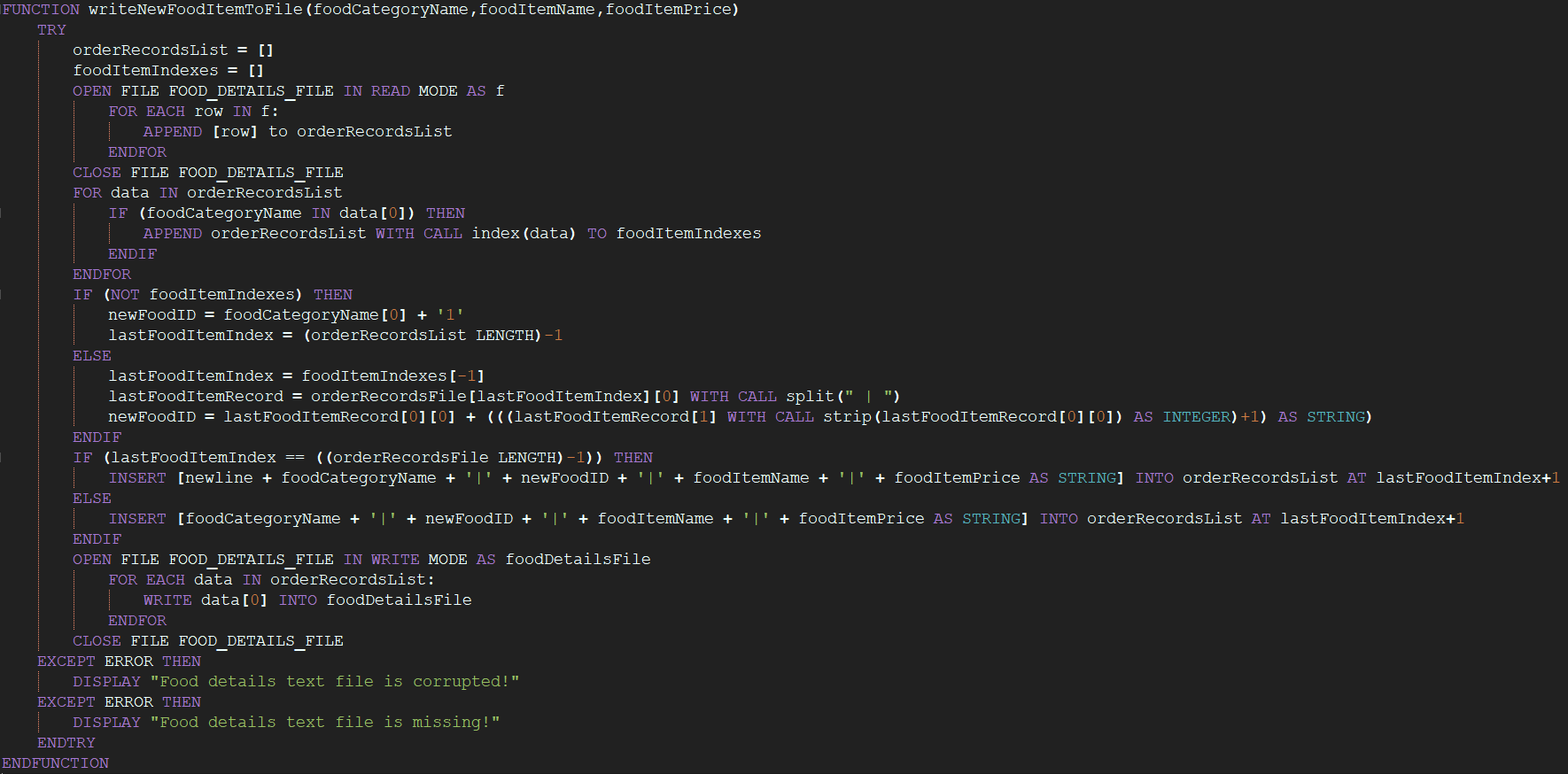
**getNewFoodItemDetails**



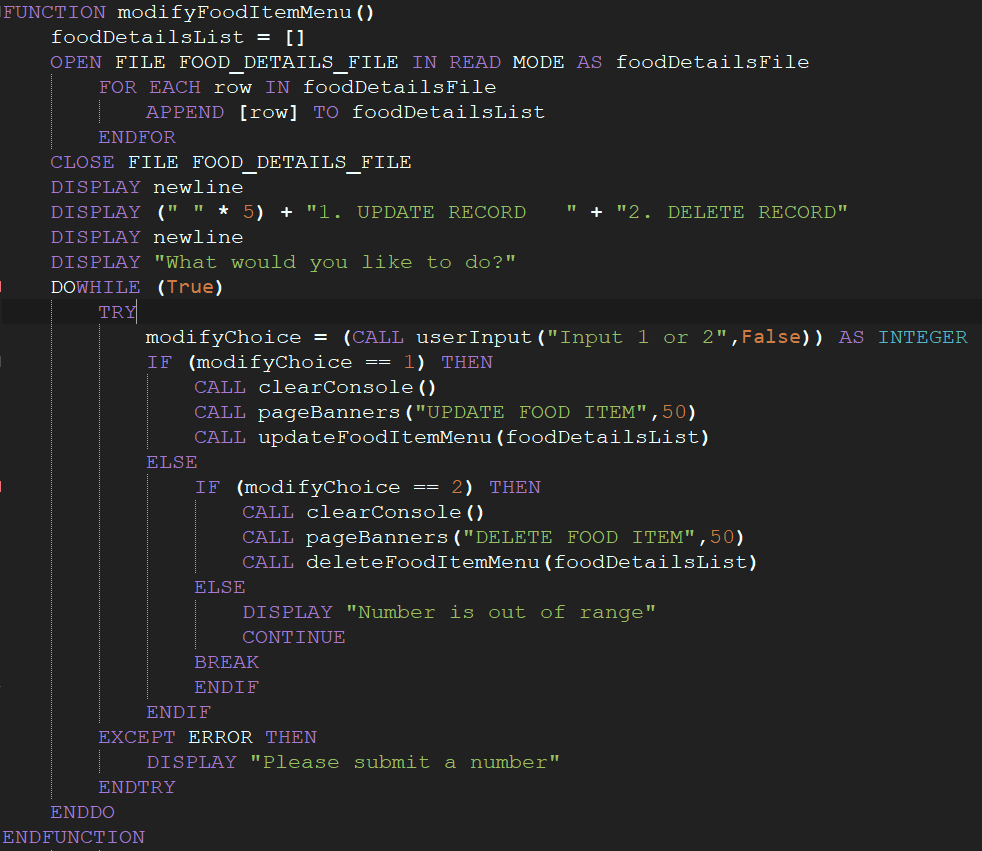
**writeNewFoodCategoryToFile**

****

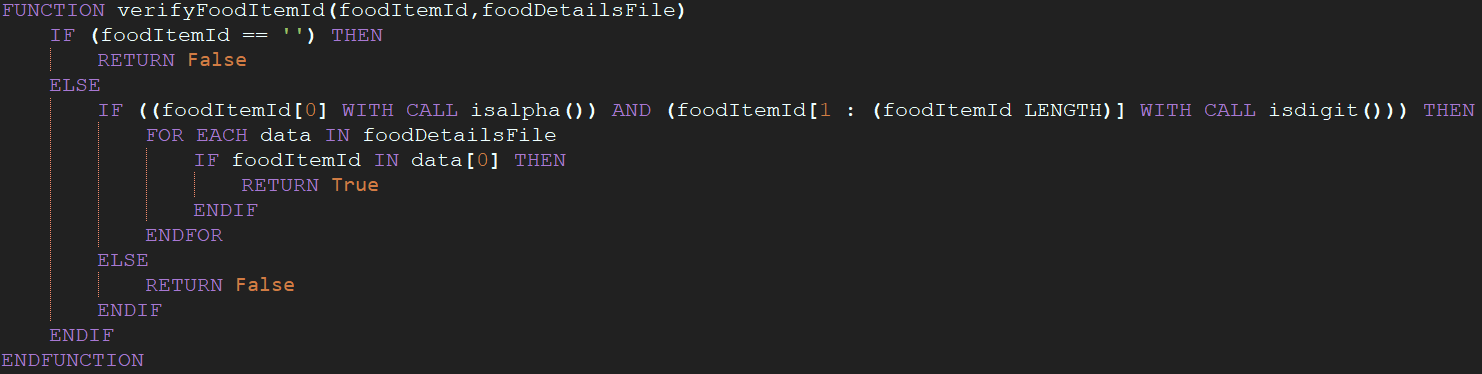
**writeNewFoodItemToFile**



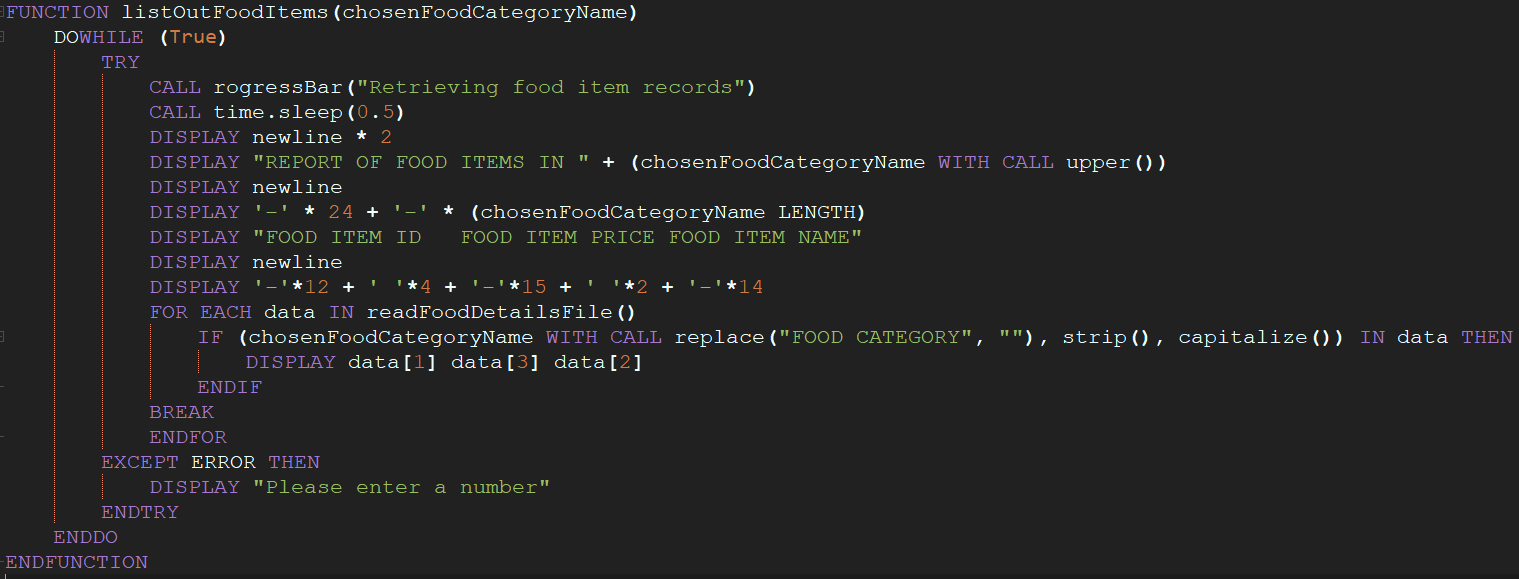
**modifyFoodItemMenu**



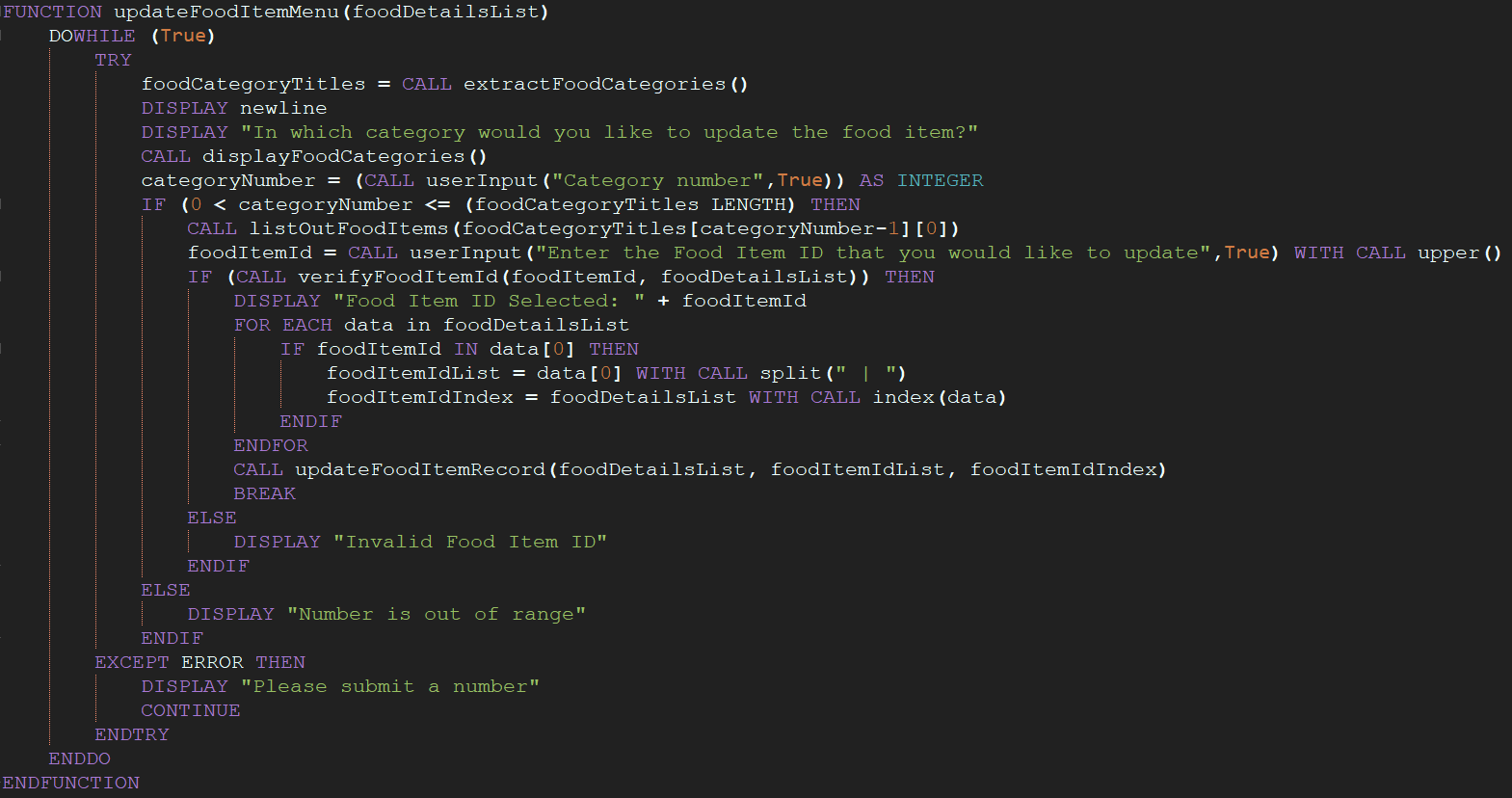
**verifyFoodItemId**



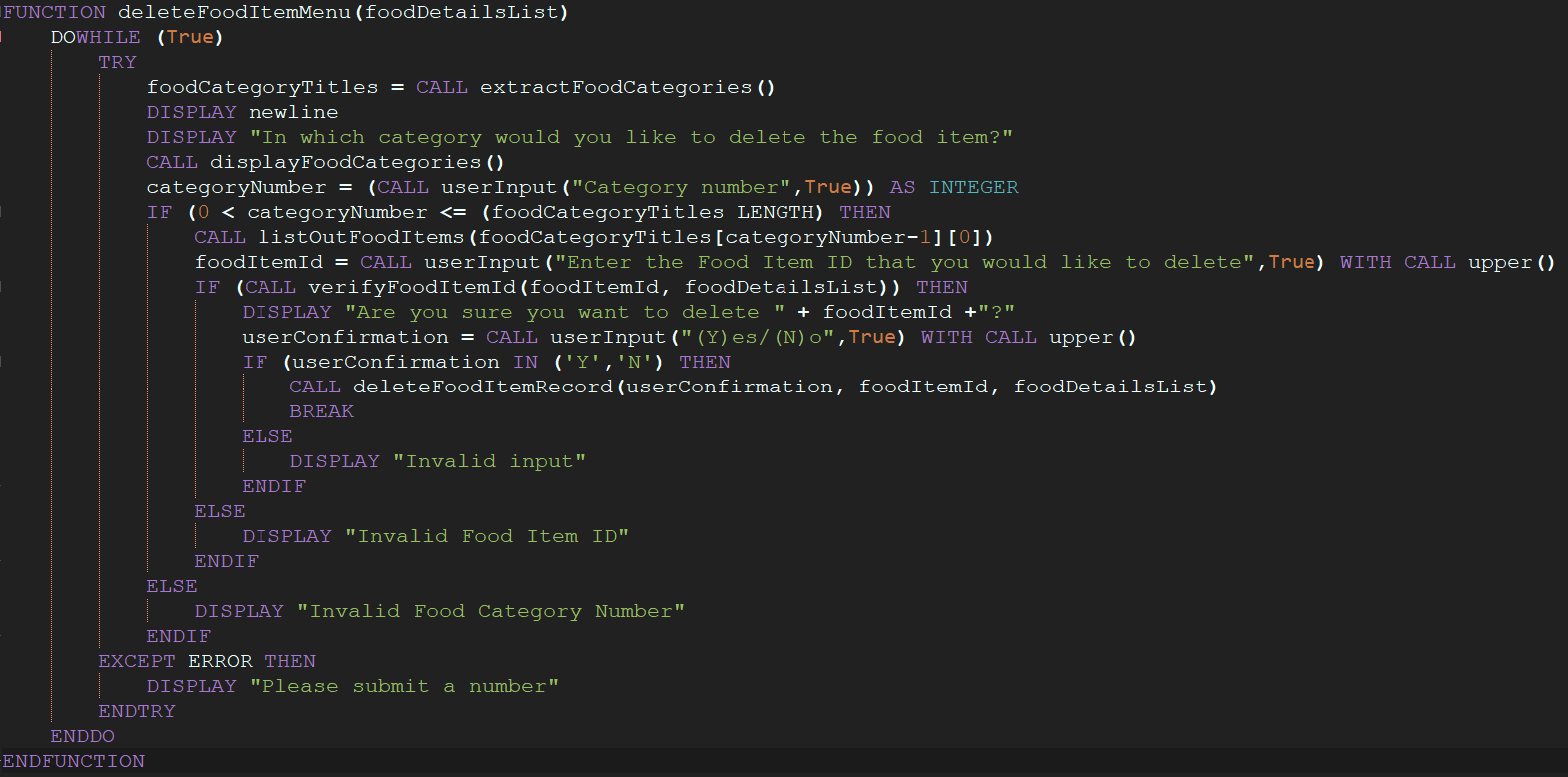
**listOutFoodItems**

****

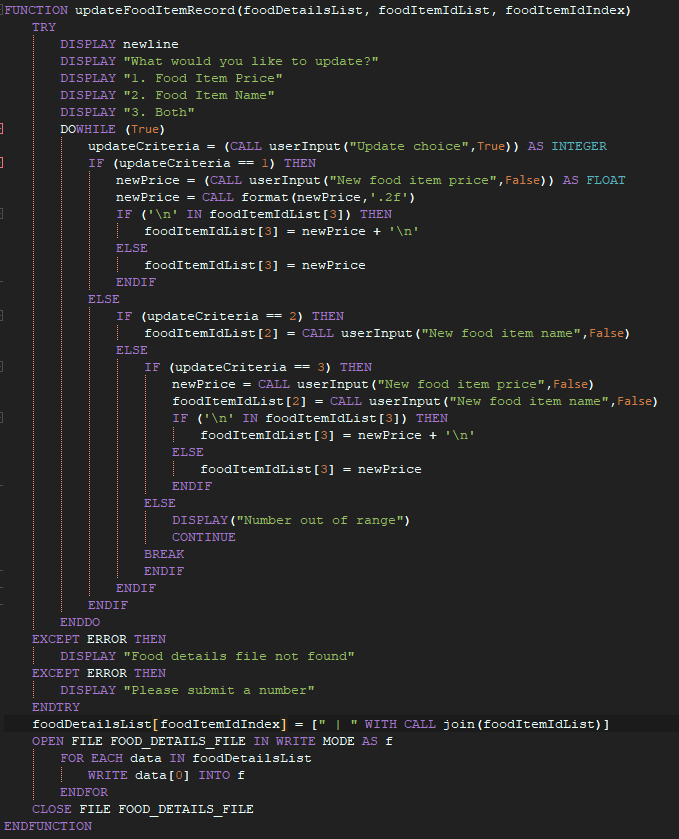
**updateFoodItemMenu**

****

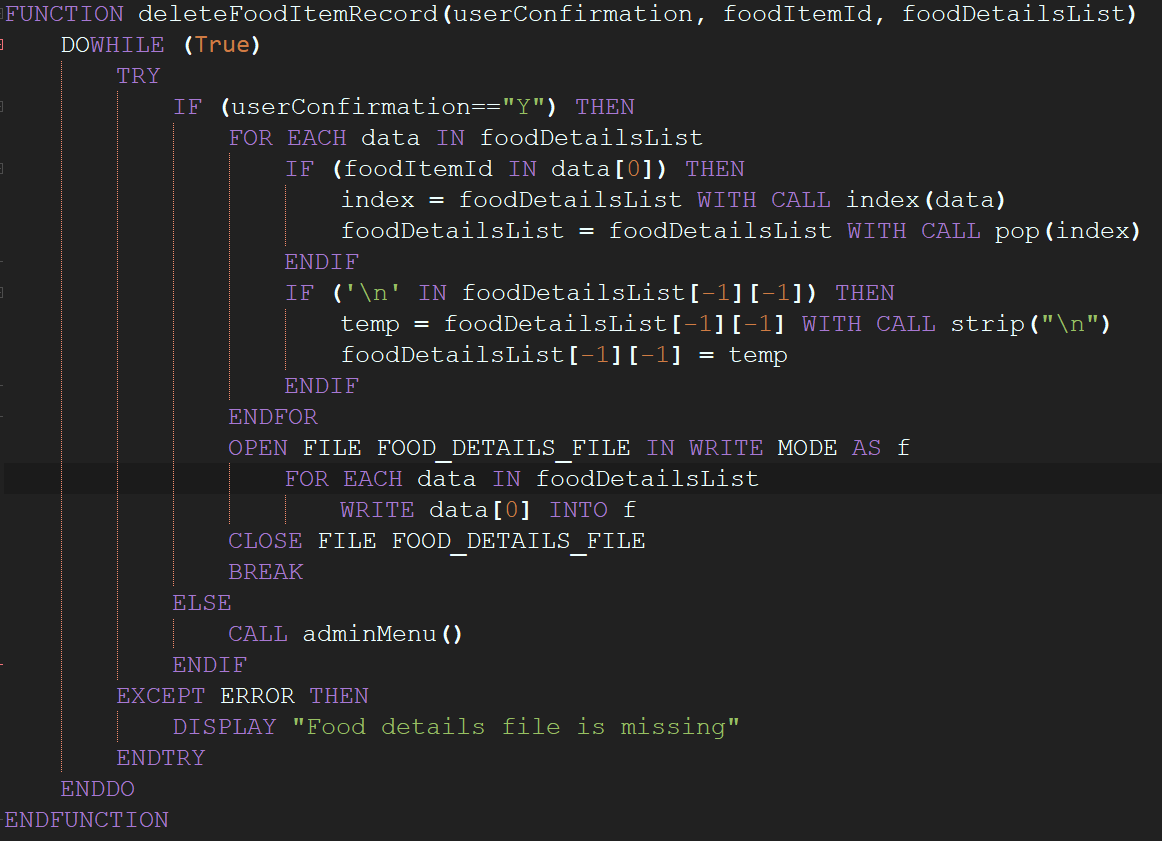
**deleteFoodItemMenu**

****

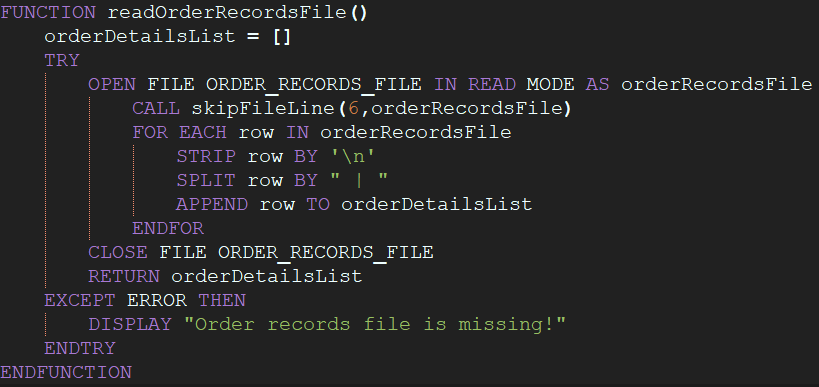
**updateFoodItemRecord**



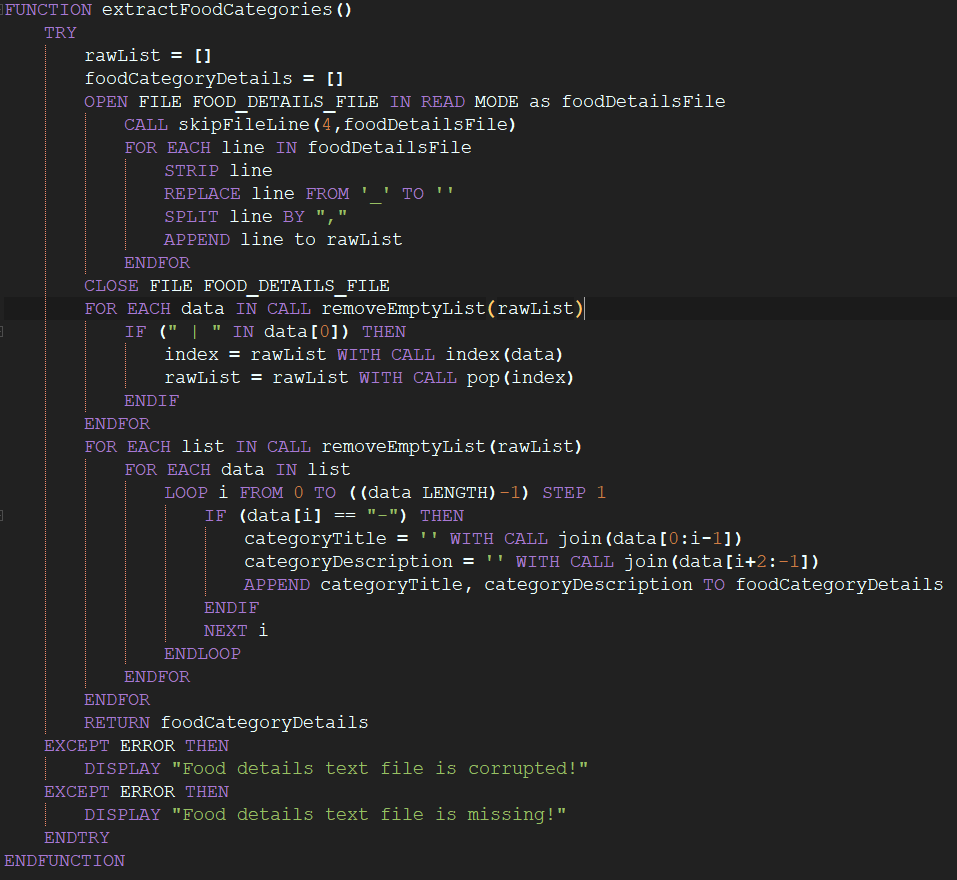
**deleteFoodItemRecord**



**readOrderRecordsFile**



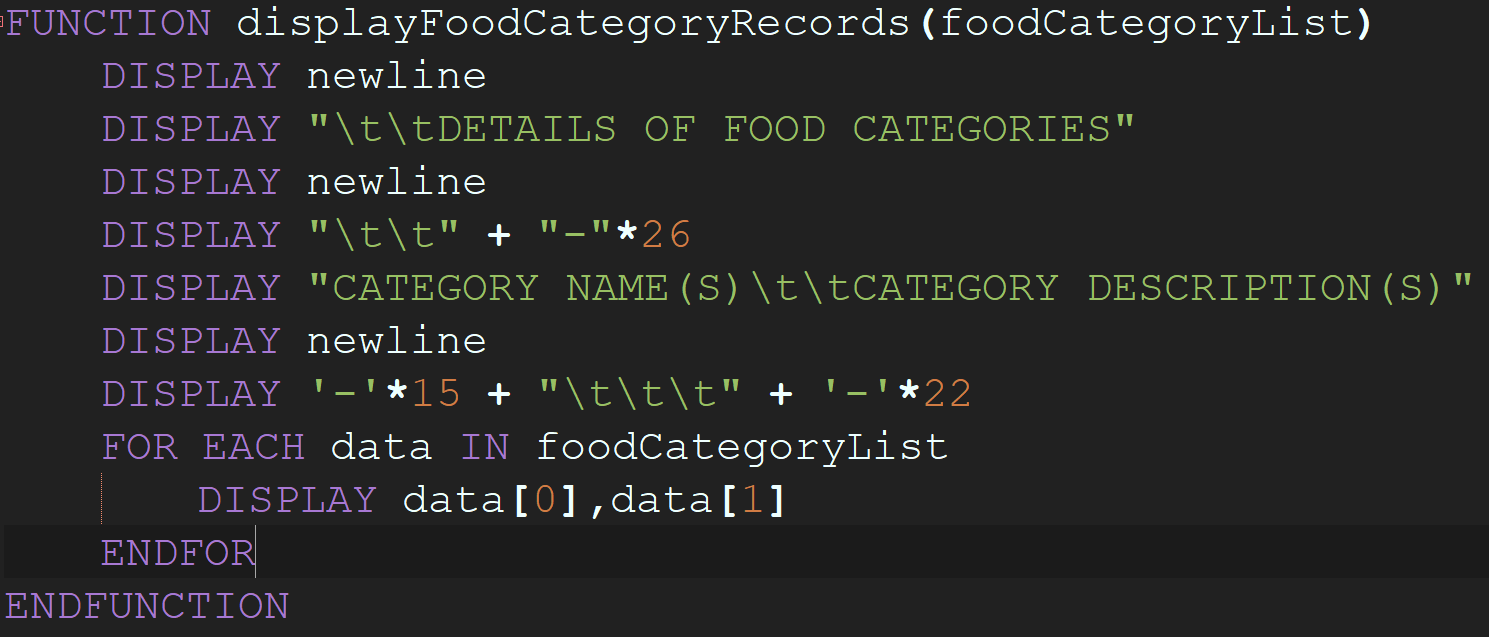
**extractFoodCategories**



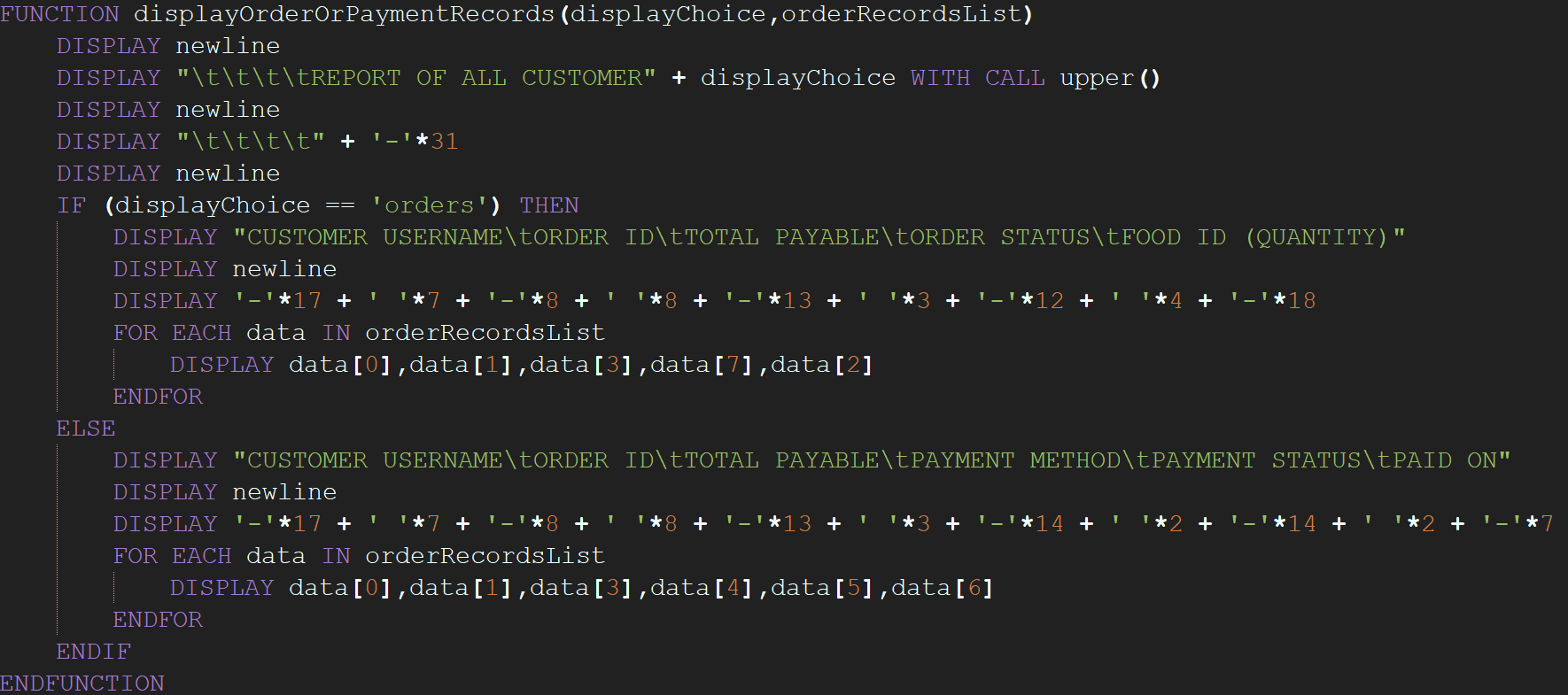
**displayRecordsMenu**

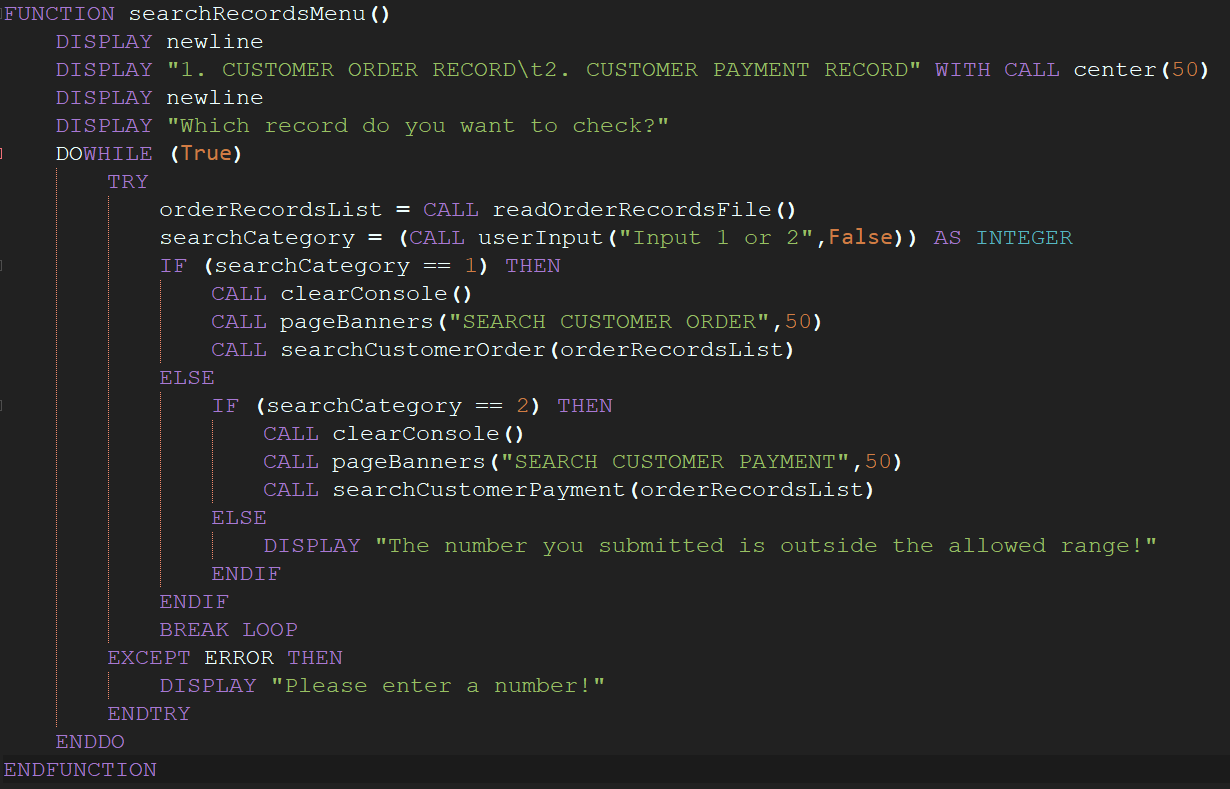


**displayFoodCategoryRecords**

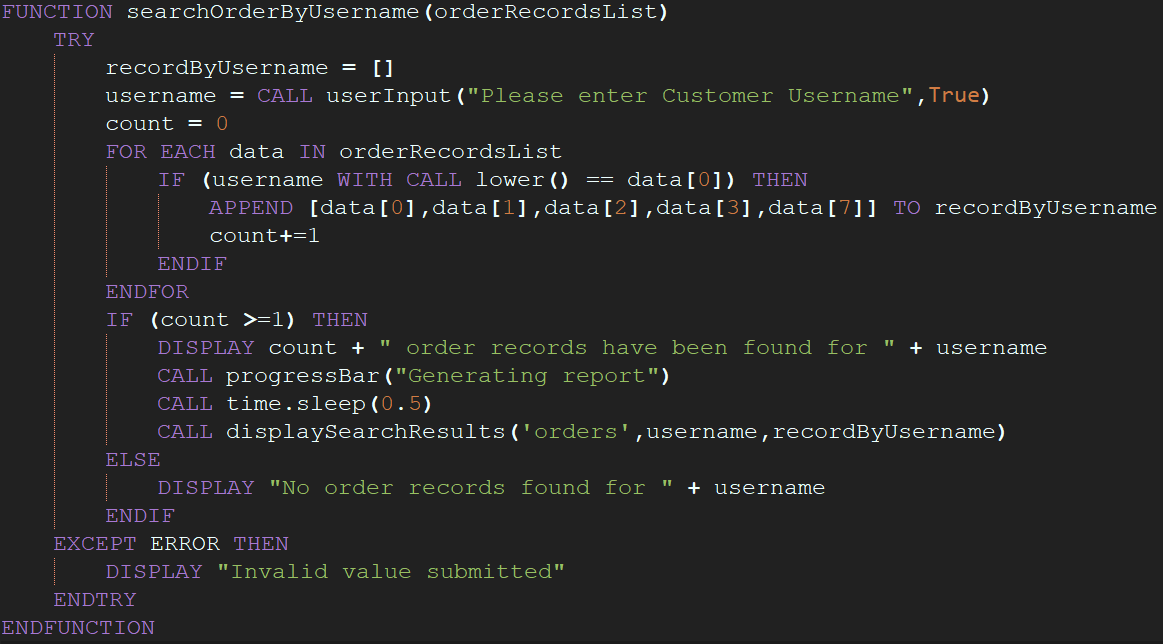
****

**displayOrderOrPaymentRecords**

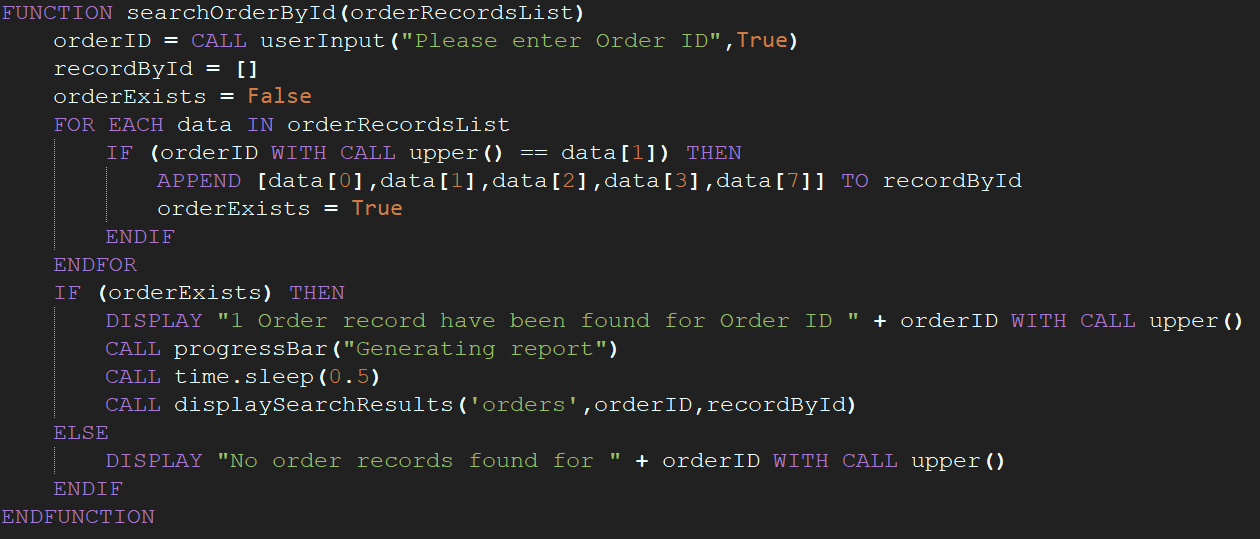
**searchRecordsMenu**



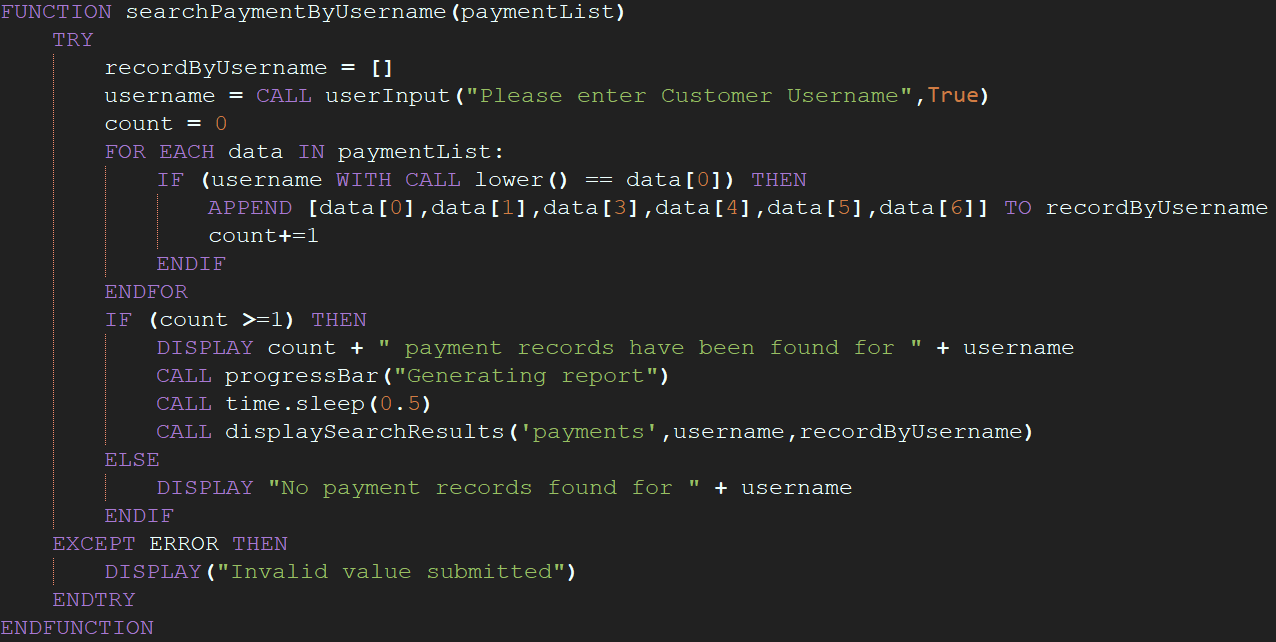
**searchOrderByUsername**

****

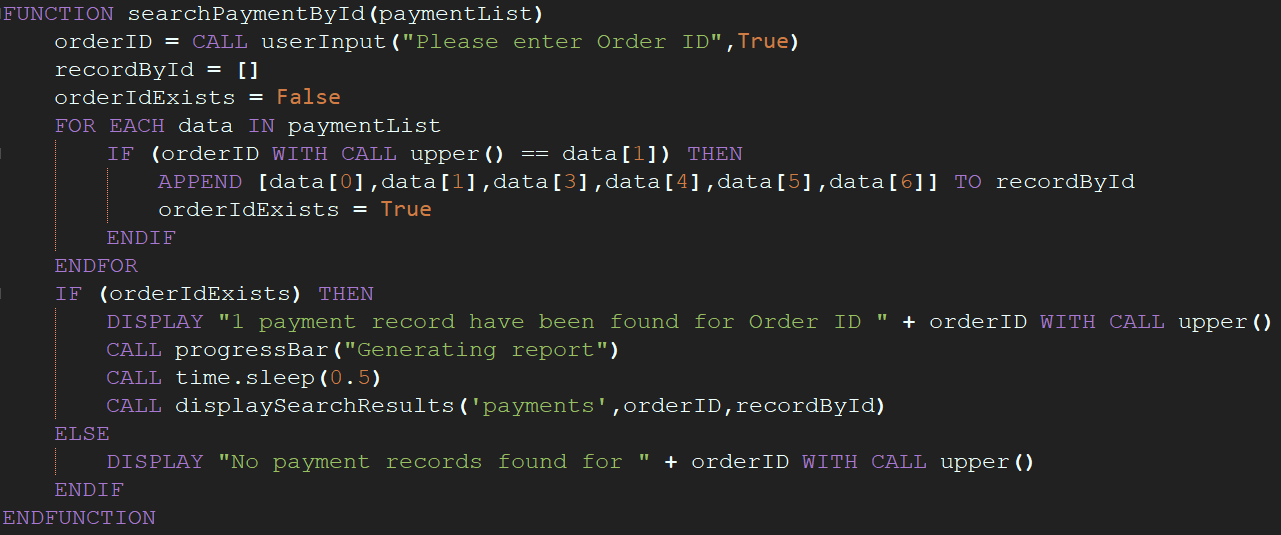
**searchOrderById**

****

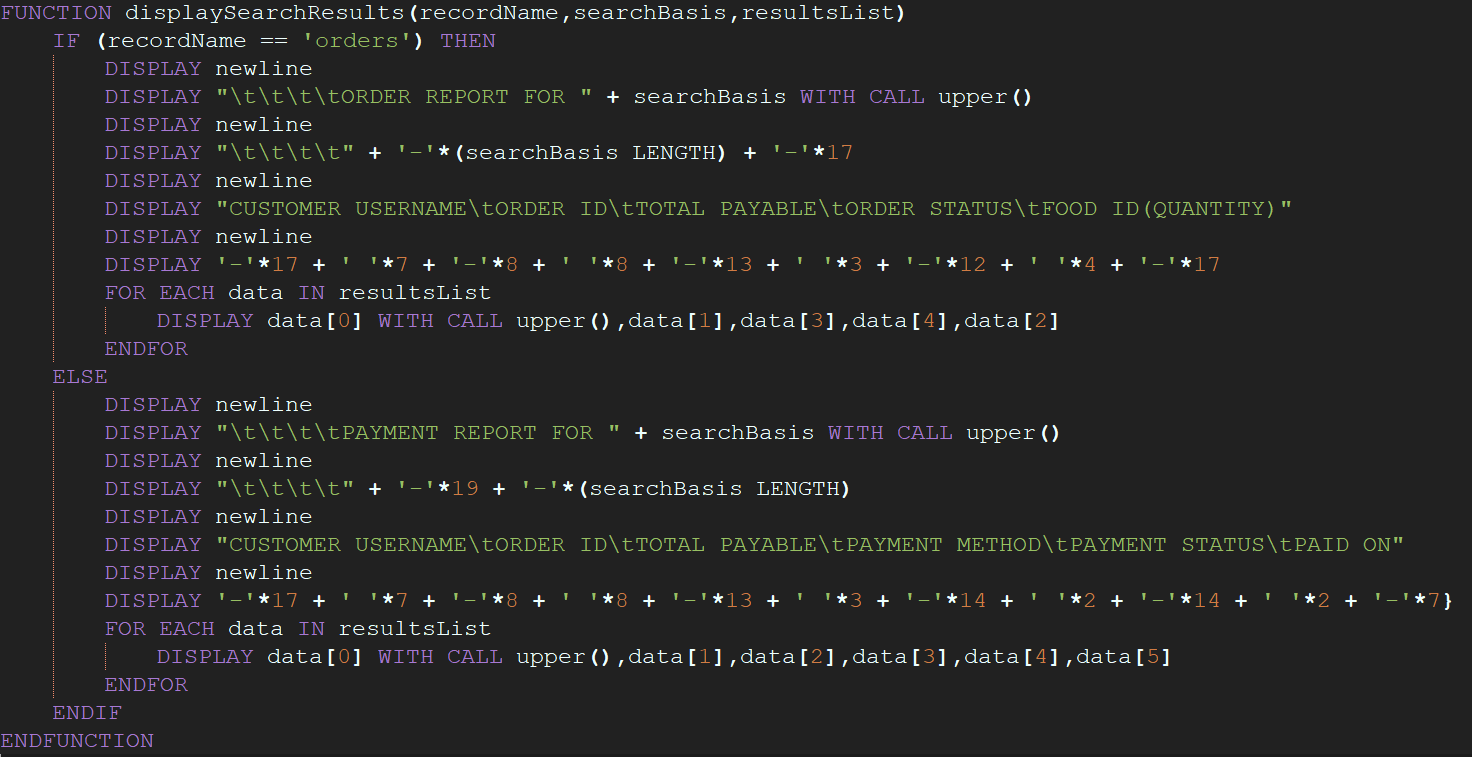
**searchPaymentByUsername**

****

**searchPaymentById**

****

**displaySearchResults**

****

## 2.2 Flowchart

1. Determine problems
2. Draw IPO Chart based on problems
3. Create pseudocode
4. Flowchart

# 3. Program source code with explanation

# 4. Additional features source code with explanation

# 5. Screenshots of sample input/output with explanation

# 6. Conclusion

# References

**There are no sources in the current document.**