|  |  |
| --- | --- |
| **Project Case** |  |
| COMP7066  Expert System |
| **Computer Science** | **E212-COMP7066-ZZ01-00** |
| ***Valid on*** *Even Semester Year 2020/2021* | **Revision 01** |

1. Seluruh kelompok tidak diperkenankan untuk:

*The whole group is not allowed to:*

* + - Melihat sebagian atau seluruh proyek kelompok lain,

*Seeing a part or the whole project from another groups*

* + - Menyadur sebagian maupun seluruh proyek dari buku,

*Adapted a part or the whole project from the book*

* + - Mendownload sebagian maupun seluruh proyek dari internet,

*Downloading a part or the whole project from the internet,*

* + - Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,

*Working with another theme which is not in accordance with the existing theme in the matter of the project,*

* + - Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + - Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai kelompok** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the group is proved to the actions described in point 1 above, the score of the group which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan proyek, segala jenis pengumpulan proyek di luar jadwal tidak dilayani.

*Pay attention to the submission schedule for the project, all kinds of submission outside the project schedule will not be accepted*

1. Bila Anda tidak membaca peraturan ini, maka Anda dianggap telah membaca dan menyetujuinya

*If you have missed to read these regulations, so you are considered to have read and agreed on it*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* |
| 40% | 60% |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| • Java 8  • Eclipse Oxygen 3 with JESS 71p2 |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri dan proyek untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment and project collection for this subject are described as follows:*

|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* |
| CLP | CLP, JAVA, CLASS |

**Soal**

*Case*

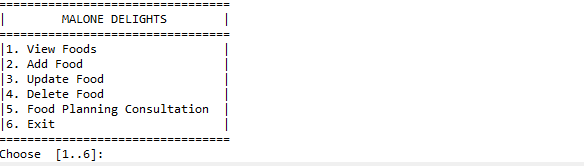
**Malone Delights**

**Malone Delights** is a restaurant created by a fan of the series “**The Office**” which was built with the idea of one of the show’s character called Kevin Malone that spilled a pantry full of salsa on the office floor. The restaurant provides a variety of food which includes **Mexican, Chinese, and Italian**. The restaurant also opens a catering option for personal up to large scale purposes. Since this is a new made restaurant, the management staff hasn’t had a grasp of the idea of what the people would want. So instead, the restaurant specifically calls you as their expert system programmer. He asks you to build an application that can give an expert solution for helping **Malone Delights** gets the feedback from customer and ease the process of managing the menu overall.

Here are the descriptions of this application:

# The application consists of **6 main menus**:

* + - 1. **View Foods**
      2. **Add Food**
      3. **Update Food**
      4. **Delete Food**
      5. **Food Planning Consultation**
      6. **Exit**



**Figure 1. Main menu screen**

# If user chooses **menu 1** “**View Foods**”, then:

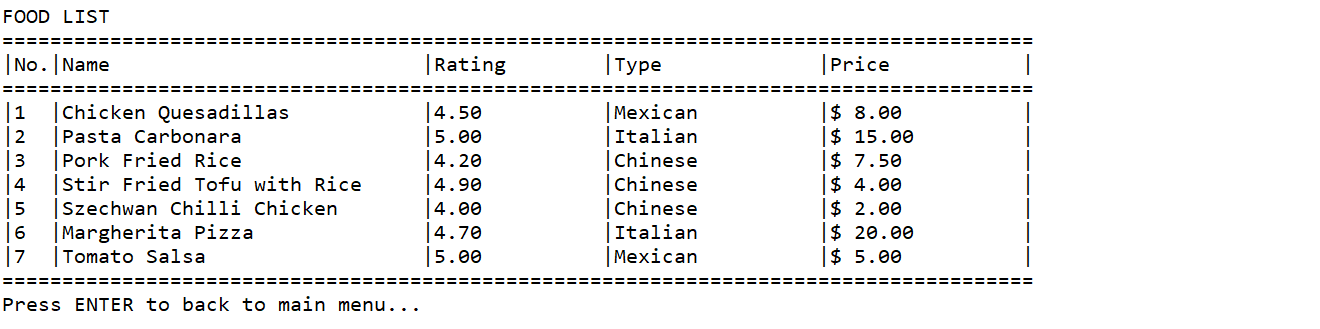
## The application will ask user to input whether they want to view the **Foods** or the **Food Plans**. **Validate** input must be **numeric** and **between** **0 and 2**.

## If user chooses **0**,the application will **go** **back** to **main menu**.

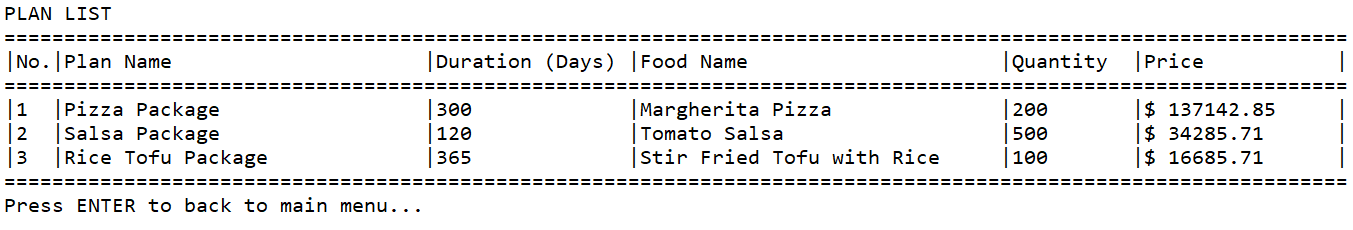
## Otherwise, then the application will **show all** the **selected artist facts**.



**Figure 2. Choose food/food plan screen**



**Figure 3. View food screen**



**Figure 4. View food plan screen**

# If user chooses **menu 2** “**Add Food**”, then:

## The application will ask user to input whether they want to view the Foods or the Food Plans. **Validate** input must be **numeric** and **between** **0 and 2**.

## If user chooses **0**,the application will **go** **back** to **main menu**.

## 

**Figure 5. Choose category to insert screen**

## Otherwise, then:

## If user chooses **sub menu 1** (“**Food**”), then the application will ask user to input:

### **Name**. **Validate** input must be **between 5 and 28** **characters**.

### **Rating**. **Validate** input **must be float** and must be **between** **0.0 and 5.0**.

### **Type**. **Validate** input must be **either** “**Italian**”, “**Chinese**”, or “**Mexican**” (**Case Sensitive**).

### **Price**. **Validate** input must be a **float** and **between the minimum and maximum for each type of food**.

|  |  |  |
| --- | --- | --- |
| **Food Type** | **Minimum Price** | **Maximum Price** |
| **Italian** | $15.00 | $80.00 |
| **Chinese** | $1.50 | $20.00 |
| **Mexican** | $5.00 | $25.00 |

## If user chooses the **sub menu 2** (“**Food Plan**”), then the application will ask user to input:

### **Name**. **Validate** input **length** must be **less than 20** **characters and ends with ‘Plan’ but cannot only be ‘Plan’**.

### **Duration**. **Validate** inputmust be an **integer** and **must be between 10 and 3650 days**.

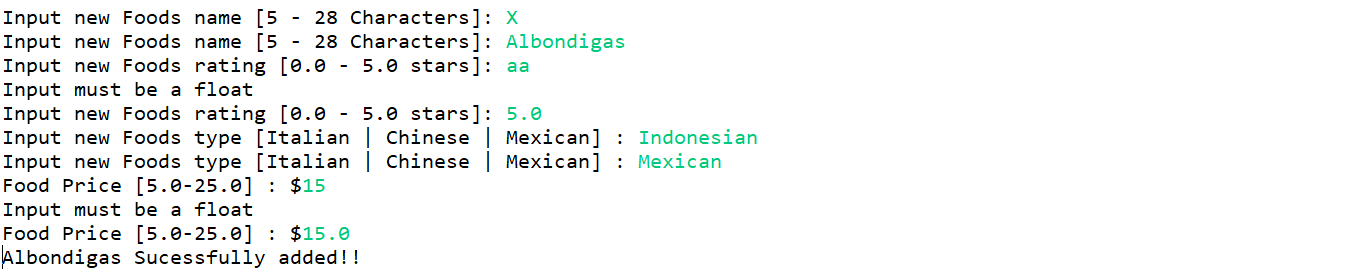
### **Food Name**. First, **show the list of foods** available on the menu. Then, **validate** input must be **a food that exists in the menu**.

### **Food Portion Per Week**. **Validate** input must be **integer** and **must be more than 10 portions**.

### **Plan Price**. The plan price is calculated by this formula

**Plan Price = (Food Price \* 0.8) \* (Duration / 7) \* Food Portion**

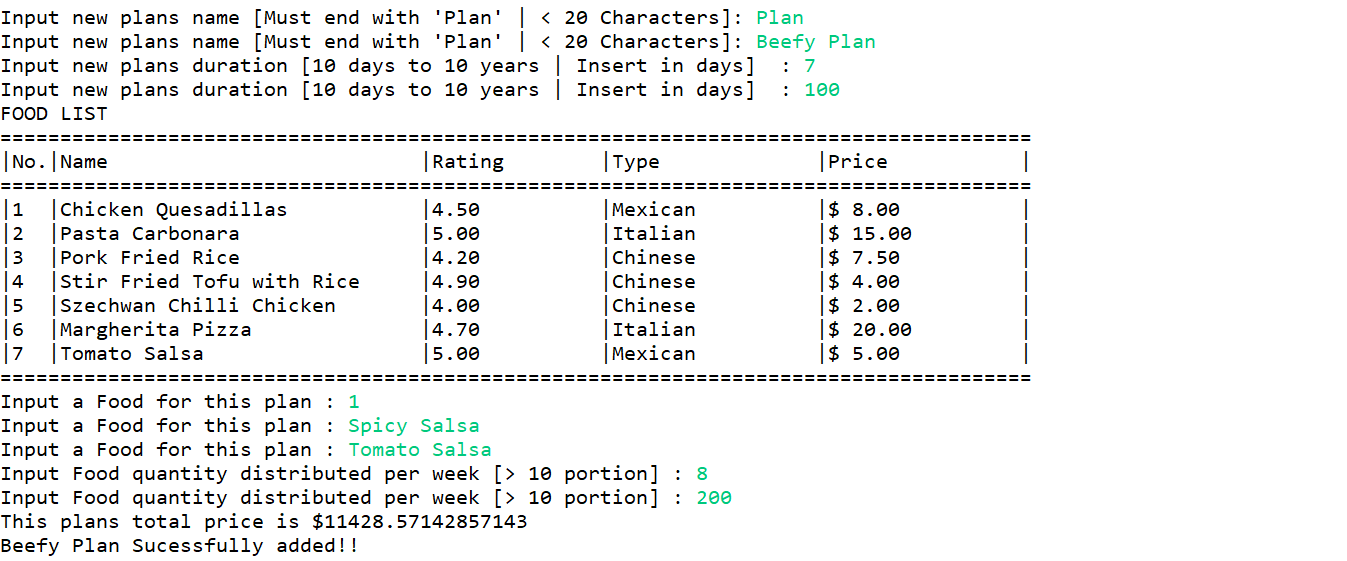
### After that, the application will **insert** **the data** into the **selected category** and **show success message along with the name of the data**.



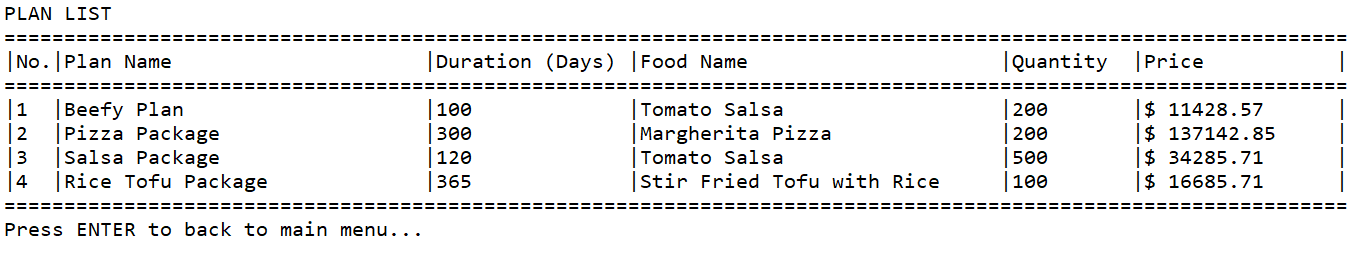
**Figure 6. Food validation screen**

## 

**Figure 7. View food after adding a new food screen**



**Figure 8. Add new food plan validation screen**



**Figure 9 . View food plan after adding new plan screen**

# If user chooses **menu 3** “**Update Food**”, then:

## The application will ask user to input whether they want to update the **Foods** or the **Food Plans**. **Validate** input must be **numeric** and **between** **0 and 2**.

## If user chooses **0**,the application will **go** **back** to **main menu**.

## 

**Figure 10. Choose category to update**

## Otherwise, then:

## If user chooses **sub menu 1** (“**Food**”), then the application will ask user to input:

### **Show the list of food available first.**

### **Name**. **Validate** input must be **between 5 and 28** **characters**.

### **Rating**. **Validate** input **must be a float** and must be **between** **0.0 and 5.0**.

### **Type**. **Validate** input must be **either** “**Italian**”, “**Chinese**”, or “**Mexican**” (**Case Sensitive**).

### **Price**. **Validate** input must be a **float** and **between the minimum and maximum for each type of food**.

**NOTE**: Updating foods won’t change the food plan data.

|  |  |  |
| --- | --- | --- |
| **Food Type** | **Minimum Price** | **Maximum Price** |
| **Italian** | $15.00 | $80.00 |
| **Chinese** | $1.50 | $20.00 |
| **Mexican** | $5.00 | $25.00 |

## If user chooses the **sub menu 2** (“**Food Plan**”), then the application will ask user to input:

### **Show the list of food plans available first.**

### **Name**. **Validate** input **length** must be **less than 20** **characters and ends with ‘Plan’, but cannot only be ‘Plan’**.

### **Duration**. **Validate** inputmust be an **integer** and **must be between 10 and 3650 days**.

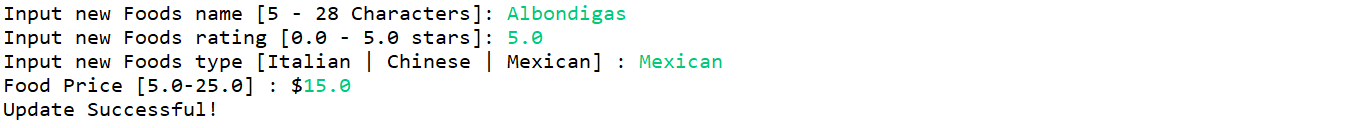
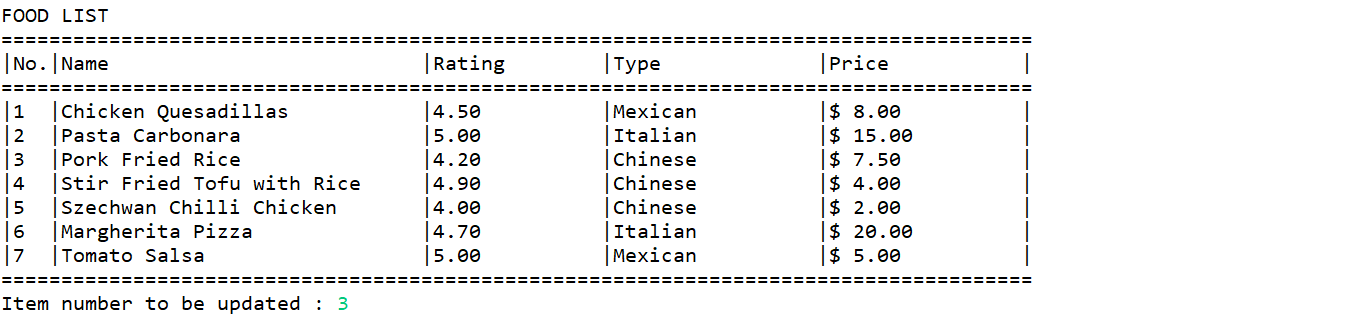
### **Food Name**. First, **show the list of foods** available on the menu. Then, **validate** input must be **a food that exists in the menu**.

### **Food Portion Per Week**. **Validate** input must be an **integer** and **must be more than 10 portions**.

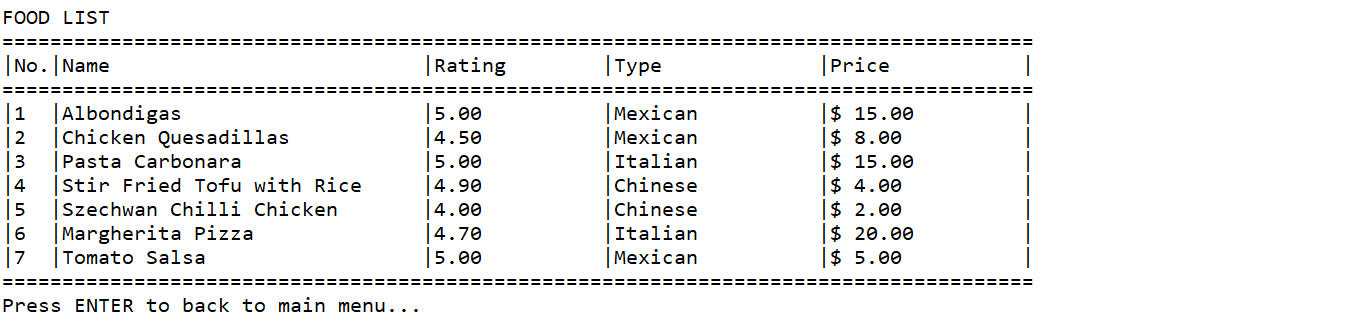
### **Plan Price**. The plan price is calculated by this formula

**Plan Price = (Food Price \* 0.8) \* (Duration / 7) \* Food Portion**

### After that, the application will **update** the **selected** **data by number** and **show success message**.

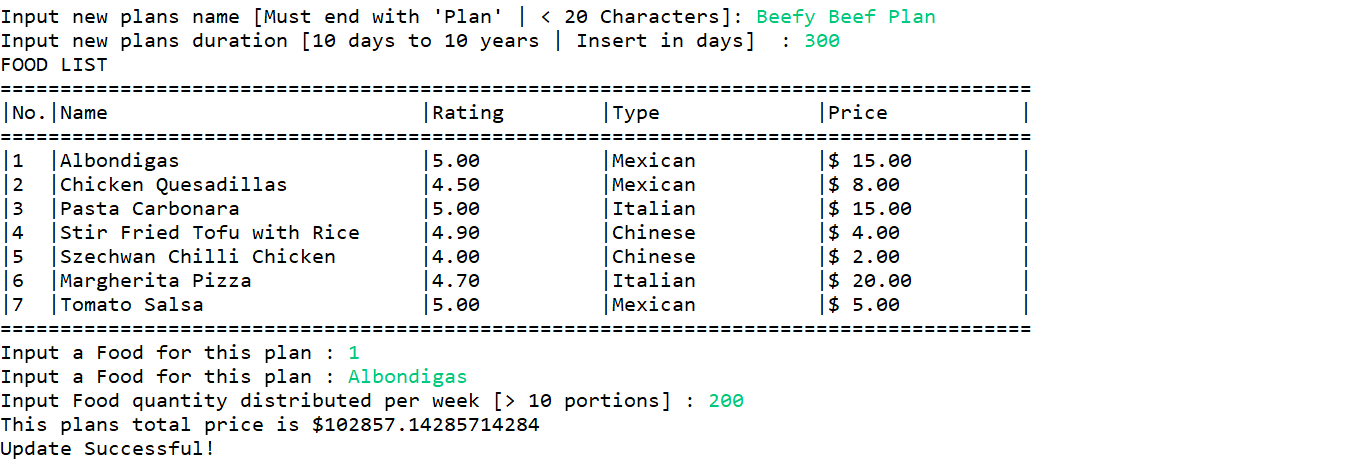


**Figure 11. Update Food validation screen**

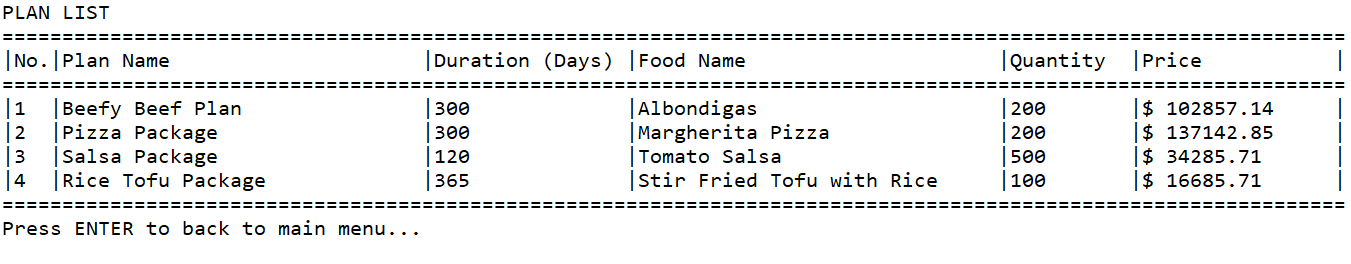


**Figure 12. View Food after Updating Pork Fried Rice**

### 



**Figure 13. Update Food Plan validation screen**



**Figure 14 . View Food Plan after Updating Beefy Plan**

# If user chooses the **menu 4** “**Delete Food**”, then:

## The application will ask user to input whether they want to view the Foods or the Food Plans. **Validate** input must be **numeric** and **between** **0 and 2**.

## If user chooses **0**,the application will **go** **back** to **main menu**.

## 

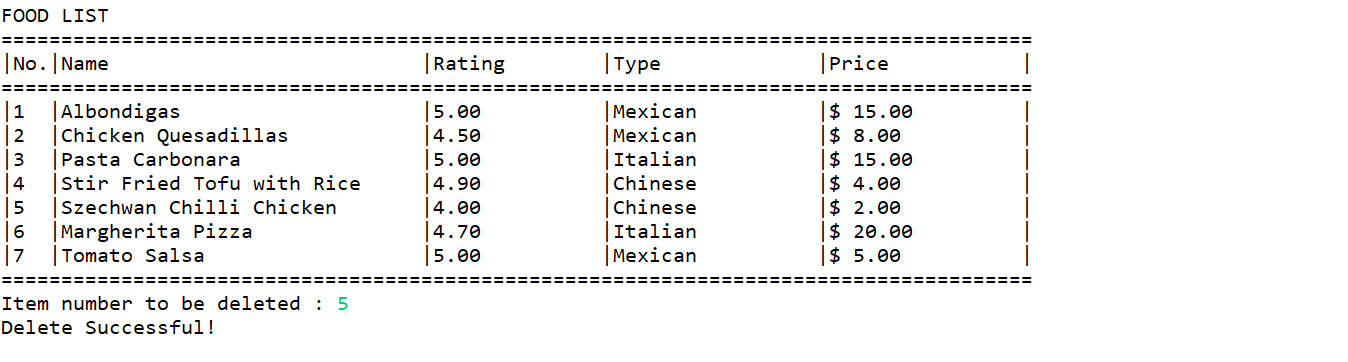
**Figure 15. Choose category to delete screen**

## Otherwise, then:

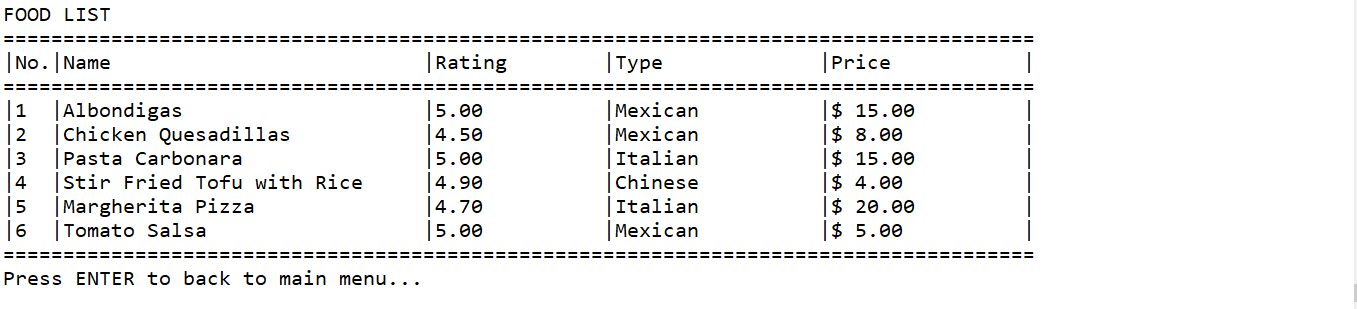
## If user chooses **sub menu 1** (“**Food**”), then the application will show the list of Foods available on the menu. The input must be between 1 and the amount of foods.

## If user chooses the **sub menu 2** (“**Food Plan**”), then the application will show the list of Food plans available on the menu. The input must be between 1 and the amount of food plans.

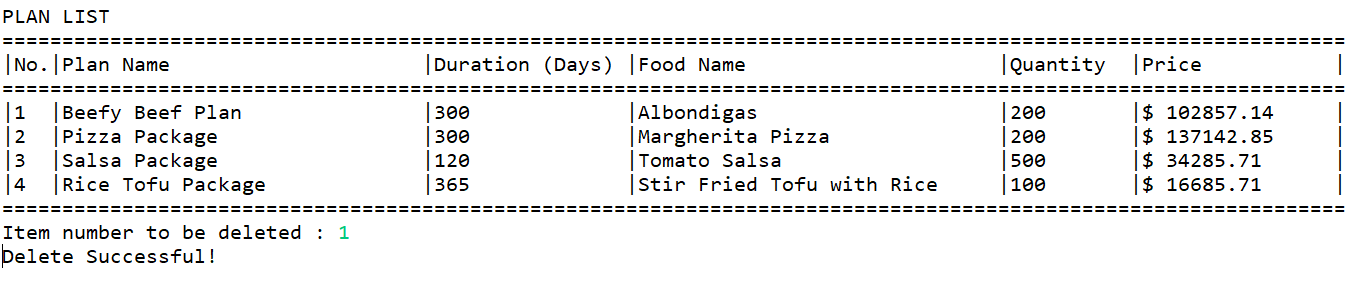
### After that, the application will **update** the **selected** **data by number** and **show success message**.



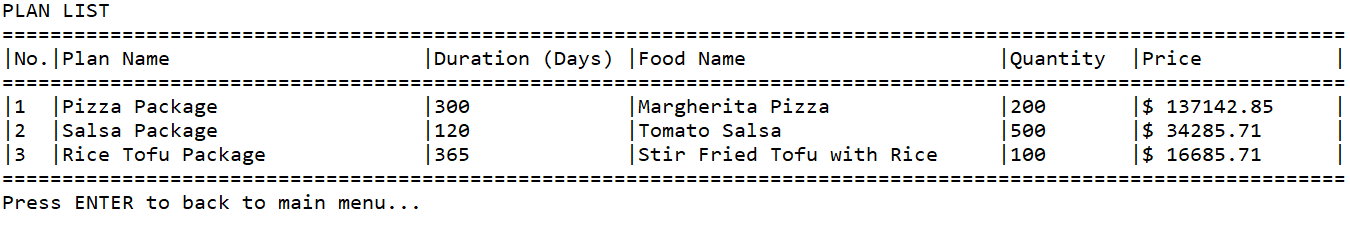
**Figure 16. Delete food**



**Figure 17. View food after deleting Chili Chicken**



**Figure 18 . Delete Food Plan**



**Figure 19 . View Food Plan after Deleting Beefy Beef Plan**

# If user chooses **menu 5** “**Food Planning Consultation**”, then:

## The application will ask user to input whether they want to view the Foods or the Food Plans. **Validate** input must be **numeric** and **between** **0 and 2**.

## If user chooses **0**,the application will **go** **back** to **main menu**.

## 

**Figure 20. Choosing a category to consult**

## If the user chooses the menu 1 **(“Foods”)** The application will ask user to input:

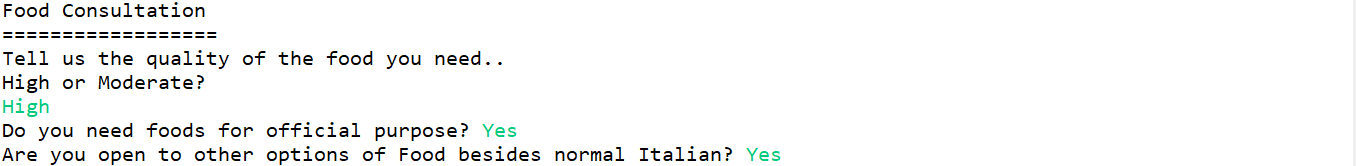
## **Quality of food**. **Validate** input must be **either High or Moderate (Case Sensitive)**.

## **Official Purposes (only shown when quality of food is High)**. **Validate** input must be **either** “**Yes**”or“**No**”. (**Case Sensitive**).

## **Versatile (only shown when quality of food is High)**. **Validate** input must be **either** “**Yes**”or“**No**”. (**Case Sensitive**).

## **Budget**. **Validate** input must be **either** “**Unlimited**”**,** “**High**”**, or “Low”**. (**Case Sensitive**).

## 



**Figure 21. Find Food validation screen**

## Then, the application will **search** for the corresponding food that fulfills these following criteria:

### If the **quality of food is “High”,** then:

* **Ask the user** if the food is for **official purpose** or not.
* If the answer is “**Yes**”, then **ask the user** if **they are okay with any kind of food or not** (**versatility option**).
* If the answer is “**No**”, then **ask the user** the **budget scale** for their meal.
  + If the **quality of food is “Moderate”,** then ask the user immediately for the budget scale for their meal.

### If the **quality is High**, **it is for official purposes, and user is versatile**, then show **all foods** available.

### If the **quality is High**, **it is for official purposes**, **and user is not versatile**, then show **all** **Italian** **foods** available.

### If the **quality is High**, **it is for personal purposes**, **and user’s budget is Low**, then show **all** **foods** with the price less than **$4**.

### If the **quality is High**, **it is for personal purposes**, **and user’s budget is High**, then show **all** **foods** with the price less than **$100**.

### If the **quality is High**, **it is for personal purposes**, **and user’s budget is Unlimited**, then show **all** **foods**.

### If the **quality is Moderate, and user’s budget is Low**, then show **all** **foods** with the price less than **$3**.

### If the **quality is Moderate, and user’s budget is High**, then show **all** **foods** with the price less than **$15**.

### If the **quality is Moderate, and user’s budget is Unlimited**, then show **all** **foods**.

## If the user chooses the menu 2 **(“Food Plans”)** The application will ask user to input:

## **Needs**. **Validate** input must be **either** “**Personal Purposes**”, “**Small Scale**”or“**Large Scale**”. (**Case Sensitive**).

## **Budget**. **Validate** input must be **either** “**Unlimited**”**,** “**High**”**, or “Low”**. (**Case Sensitive**).

## **Quality of food**. **Validate** input must be **either High or Moderate (Case Sensitive)**.

## 

**Figure 22. Find Food Plan Validation Screen**

## Then, the application will **search** for the corresponding food plan that fulfills these following criteria:

### If the **quality of food is “High”,** then the food type in the plan must be “**Italian**”.

### If the **quality of food is “Moderate”,** then the food type in the plan must be “**Mexican**” or “**Chinese**”.

### If the **need is for “Personal Purposes”,** then the plans food portion/week **must be less than 50**.

### If the **need is for “Small Scale”,** then the plans food portion/week **must be less than 1000**.

### If the **need is for “Large Scale”,** then the plans food portion/week **must be less than 5000**.

### If the **budget is “Small”,** then the plans price **must be less than $10000**.

### If the **budget is “Medium”,** then the plans price **must be less than $50000**.

### If the **budget is “Large”,** then the plans price **must be less than $100000**.

### If the **budget is “Unlimited”,** then show **all the food plans** based on the **quality of the food** **without price restriction**.

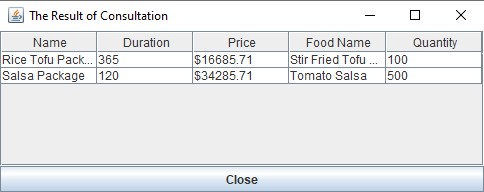
## After that, the application will **show** the **result of consultation** in the **GUI** (**Graphical User Interface**) view with the following criteria:

## If there are **foods/plans** that **fulfills** the **criteria**, **show all data** that **fulfills** the **criteria**.

* Otherwise, then **show a text that says, ‘Unable to Recommend’**.
* Then, the application will **go back** to the **main menu**.



**Figure 23. Consultation result form when foods that fulfills criteria exists**



**Figure 24. Consultation result form**

## 

**Figure 25. Consultation result form when a food that fulfills criteria doesn’t exists**

* + - If user chooses **menu 6** “**Exit**”, then the application will **be terminated**.
    - These following facts must be **included** when the application **started**:

## **Food Facts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Rating** | **Type** | **Price** |
| Tomato Salsa | 5.0 | Mexican | $5.00 |
| Margherita Pizza | 4.7 | Italian | $20.00 |
| Szechwan Chili Chicken | 4.0 | Chinese | $2.00 |
| Stir Fried Tofu With Rice | 4.9 | Chinese | $4.00 |
| Pork Fried Rice | 4.2 | Chinese | $7.50 |
| Pasta Carbonara | 5.0 | Italian | $15.00 |
| Chicken Quesadillas | 4.5 | Mexican | $8.00 |

## **Food Plan Facts**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Duration** | **Price** | **Food Name** | **Food Quantity** |
| Rice Tofu Package | 365 | $16685.71 | Stir Fried Tofu with Rice | 100 |
| Salsa Package | 120 | $34285.71 | Tomato Salsa | 500 |
| Pizza Package | 300 | $137142.85 | Margherita Pizza | 200 |

References:

* <https://cdn.shopify.com/s/files/1/1009/0836/t/2/assets/not_available.jpg?3654727700511648903>