

**UC-473 Food Ordering System**

**Detailed Use case**

|  |  |
| --- | --- |
| **Layer** | Food order Usecase |
| **Author** | Anum |
| **Project Name** | Food Ordering System |
| **Creation Date** | Tuesday 28th 2018 |

**1 UC-473 Food Order**

**(Process ID:473)**

**1.1 Description**

This process allows the authorized users to access the system.

Develop a reusable login module in order to reduce cost of the software development as it can be reused

**1.2 Primary Actors**

• Public User: Public Users include end users and management to whom the end users report.

**1.3 Trigger**

• Public User Access the system.

• System Access the system.

**1.4 Pre Conditions**

• User has required authorization to access system

**1.5 Post Conditions**

• User gets to access the system**2 Flow of Events**

**2.1 Basic Flow**

1. Enter the location and desired restaurant.
2. The main form open on which enter the personal information like name email etc.
3. Then on the same page, select the desired food and customize the quantity.
4. Click on the order now button.
5. The “Thank You” window will be show.

**2.1.1 Basic Flow Process Diagram**

End Process

Order foor

Information is valid + at least 1 item must be selected

Menu + Personal information form

Enter the location and desired restaurant

****

****

****

****

**2.1.2 Related Business Rules**

1. A user can only be registered once
2. After three tries, lock user account
3. Restaurant must be in his/her city.
4. User can customize the quantity.
5. User can edit the order.

**2.2 Alternative Flows**

**Is this the third failed attempt? Passed**

1. If Is this the third failed attempt? condition is true : show Account Locked Error Message .

2. The use case ends

**Select the restaurant**

1. Enter the desired restaurant of your city.

**Personal information and Item Selection**

1. Enter your personal information.

2. Select the desired item.

**If zero order is selected**

1. At least one item must be selected, if no item is select then generate error.
2. Personal information must be valid.

**Order successfully**

After successfully place the order, notify about that the order has been placed successfully.

The use case ended.

**3 Related Information**

**3.1 Priority**

• Medium

**3.2 Secondary Actors**

• System Software Application

**3.3 Extended Use Cases**

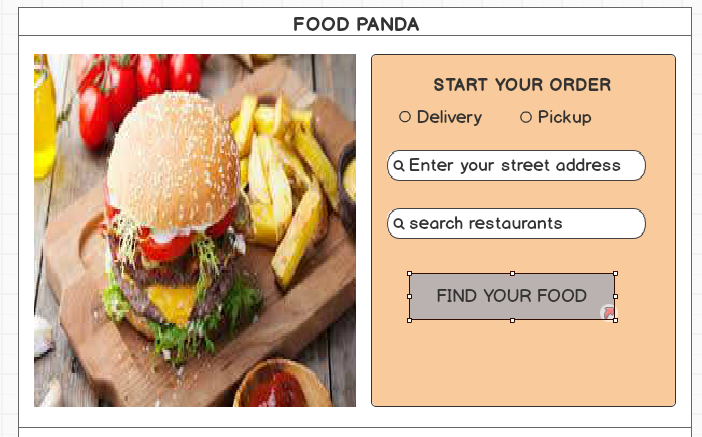
No item is select.

**3.4 Included Use Cases**

None

**3.9 UI Information**

**• Start your order**

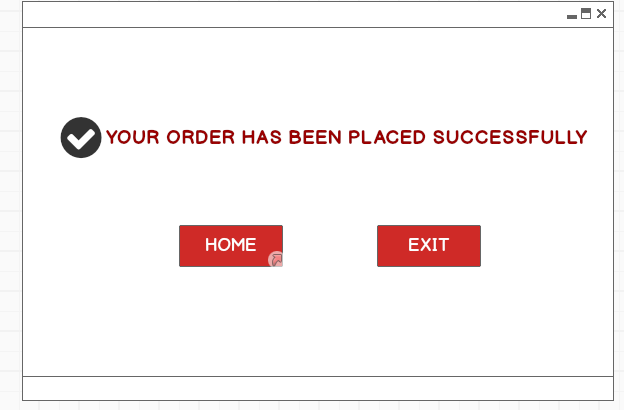


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Description** | **Characteristics** | **Field Length** | **Default Value** |
| Address | Street address of user | Data Type: String |  |  |
| Restaurant | Restaurant name | Data Type:String |  |  |

**• Select menu**



**• Confirm notification**



**4 Supplementary Information**

**4.3 Notes**

**User can reset account's password**: User will be able to change the password of his account whenever he wishes. :9/1/2008

**After three tries, lock user account**: After three invalid attempts, user's accont will be locked. :6/26/2008

**Support 100 concurrent logins**: Merged 'Support 100 concurrent logins' with 'Support 100 concurrent logins' :12/29/2008

**4.4 Dependencies**

None

**4.5 Constraints**

* Password should be eight characters long.
* Only one restaurant must be selected at a time.
* The quantity must be at least 1.

**5 Glossary**

**5.1 Technical Specifications**

**5.1.1 Use Case**

A use case defines a functional requirement that is described as a sequence of steps, which include actions performed by a system and interactions between the system and actors. Use cases address the question of how actors interact with a system, and describe the actions the system performs.

**5.1.2 Actor**

An actor is external to a system, interacts with the system, may be a human user or another system, and has goals and responsibilities to satisfy in interacting with the system. Actors address the question of who and what interacts with a system.

**5.1.3 Trigger**

A trigger specifies the event that gets the use case started.

**5.1.4 Pre-Condition**

A pre-condition describes what the system should ensure is true before the system allows the use case to begin. This is useful for telling the programmers what conditions they don't have to check for in their code.

Example: the location and the desired restaurant name must be enter first.

**5.1.5 Post-Condition**

A post-condition is a statement of what the world should look like after execution of an operation.

**5.1.6 Flow Diagram**

Flowchart diagrams can be generated for a selected layer/project. New knowledge objects and relations can also be created in this Visio Layout. Flowchart diagrams are generated according to the semantic (shapes defined for each knowledge object) applied.

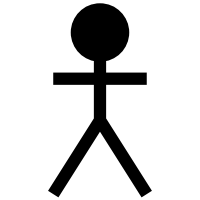
**5.1.7 Basic Flow**

At a minimum, each use case should convey a primary scenario, or typical course of events, also called 'basic flow' or 'happy flow'. The main basic course of events is often conveyed as a set of usually numbered steps.

**5.1.8 Alternate Flow**

Use cases may contain secondary paths or alternative scenarios, which are variations on the main theme.

Trigger



**Actor**