

## **UML**

**Use Case:** Order pick up

**Primary Actor:** Student / Food pantry staff

**Scope:** Order placed by the student

**Level:** User goal

### **Brief:**

The student places an order using the online order form. The system tells the student when the order will be ready for pick up. The student picks up the order from the food pantry using his/her student ID at the time of pick up.

### **Stakeholders:**

Administrator, Food Pantry Manager, Food Pantry Staff, Student

**Post conditions:** Food is received by user (leaves pantry). Database is updated.

**Preconditions:** User has an account. Food ordered has a quantity >0 in database system.

**Triggers:** Form successfully submitted to server. Order gets prepared for pick up.

Basic Flow:

1. User log in.
2. Fill out required fields in form.
3. Press the submit button.
4. Database updated.
5. Pantry staff prepares food for pickup.
6. User picks up order.
7. Database updated.

Extensions:

Cancel – The user cancels their order. The database is updated accordingly (quantity+)

Timeout – The user's order will be automatically canceled if form takes X time to complete.

API

1. Log in process;

```
{
```

```
  "action": "log in"
```

```
  "username": "User1"
```

```
  "password": "****"
```

```
}
```

=>

```
{
```

```
  "success": "true"
```

```
  "message": "Welcome username!"
```

```
}
```

OR

```
{
```

```
  "success": "false"
```

```
  "message": "Incorrect username and/or password. Please try again."
```

```
}
```

2. Fill in Form (Place Order);

```
{
```

“action”: “Fill required fields”

“field x (Item Name)”: “some data”

“field Y (Item Quantity)”: “some quantity”

}

=>

{

“success”: “true”

“action”: “Order is sent to database”

}

OR

{

“success”: “false”

“message”: “Invalid name/quantity of item.”

}

3. Order Sent to Database

{

“action”: “successfully filled in form data is sent to database”

“field x (Item Name)”: “compared to database name field”

“field Y (Item Quantity)”: “compared to database quantity field”

}

=>

{

“success”: “true”

“Form action”: “Order placed!”; “Pick up date set”

“Database action”: “item quantity updated (subtracted)”

}

OR

{

“success”: “false”

“message”: “Item not in stock.”

4. Order pick up:

{

“action”: “order successfully picked up”

“field x (Student ID)”: “compared to database student ID”

“field Y (Student receipt)”: “update database receipt field”

}

=>

{

“success”: “true”

“Form action”: “Order picked up!”

“Database action”: “item receipt updated (y/n)”

}

OR

{

“success”: “false”

“message”: “Order not picked up.”