With this solution / code that I produced, it meets the required needs because it can get customer input on what they want to see, what they have to enter, what dates they have to enter. It requires input from the user. There are some changes that I would make, and that might be because some things might not work, or might need to be changed to them not running properly.

At the start of the code, it asks the user to input an item they want to choose from that is on the menu. They can select their own item from when they input. It does show them the items first of all, before they make their choice so that they can see all the different items. Then it asks them for their input – which is a user requirement: `item_choice = input("Please enter the number of your choice (1-8): ")`, because they have to input a number, which has been delegated to an item on the menu, and when they input that number, the system then has to recognise that number, from 1-8, which then becomes a system requirement, to recognise the number. If the number was recognised originally / if it was recognised after the first attempt wasn't, then the user can progress onto the next stage.

Once they have gone onto the next stage of this solution, they then have to enter a date range, of when they want to see the data – which would be a start date: `start_date = input('Please enter start date for your time range (DD/MM/YYYY) : ')`

- This way, they can input the start date, and the system can then recognise it, the dates for the data have to be included in the system, so it can recognise it, and then can print off the selected dates. If they enter the correct start date, and the system doesn't recognise it, then it will come up with an error message, saying that: `print("Sorry, you did not enter a valid date")`, and then the user can know it's not the correct dates and will have to re-enter the dates again, like they had to do with the number input. Once the date has been recognised, the user then has to enter an end date for the data they want to see:

```
end_date = input('Please enter end date for your time range (DD/MM/YYYY) : ')
```

- This way, they can then get the end date for the data, and then can have the showing of data that they requested, and then they can view it as long as they want, because it's the dates that they requested.

There could have been a bit more development with this, maybe the user only wanted to see certain dates in between, so maybe don't see 2 days in the time frame they requested, so maybe a bit more development that they get only certain dates for the data, to maybe improve on. Or even, to enter the specific dates that they want, and that way, they get the dates that they requested, instead of a time frame of maybe some dates they didn't want to see.

Then it gets to the showing of the data, where the selected item and dates are shown to the user, this would be a system requirement for them to show it, because it is what the user requested to see. They get the 'extracted data':

```
extracted_data = get_selected_item(item, start_date, end_date)
```

This way, the data is then being processed so that it can be shown for the user, and then the user can see the data that they requested.

And then once the data for the selected item, it gets printed and shown to the user:

```
print("Here is the sales data for {} between dates {} and {}:".format(item,
start_date, end_date))
```

Now that the data has been printed off, the user who requested it can now see it, and can analyse it, do what they want with it, this is because the system works and has tested and run properly.

There would then be parts of the solution where it isn’t working, and maybe the user wants to print off the data, and can then use it for themselves, instead of having to keep accessing the code / solution to get it, if they want to go any further, they get this printed message:

```
print('This part of the program is still under development')
```

This is because it is still being worked on, and needs to be finished. But there would be some further ideas to be imbedded into this, such as the specific dates that the user might want, and not a time frame, and maybe the option of them wanting to print it off, or even getting more data after they get the original. This code meets the basic requirements, because they get the input for the items, they get the selected dates, and then they are printed off, so it has that, and obviously the code can be developed, just like any code, you can always keep adding to it and making it better. There are other things to input apart from the potential add-ons that I suggested.

It is a good code, and obviously can work better, but it is a good basic code, and can be worked by anyone, and inputted by anyone who needs the data or might want to make an analysis of it, this would be because the code works to a basic minimum. It is a good way to make analysis of food sales as well, for what days they sell what food, it would be good for making changes to the menu, and offering deals on specific days for foods that don’t get ordered that much on that day. But it is a good solution and works well.