Qt GUI Application:

1. **Inventory Dashboard**
2. **Receipt Scanner** using OpenCV
3. **Sales Prediction** using basic machine learning techniques

Image processing:

**Explanation:**

1. **Image Loading and Conversion**: The image is loaded and converted to grayscale using PIL. This is done to simplify processing by reducing the image to a single channel.
2. **Conversion to NumPy Array**: The image is converted to a NumPy array because OpenCV functions operate on NumPy arrays. This allows you to leverage OpenCV's powerful image processing functions.
3. **Thresholding**: A simple binary threshold is applied where all pixels above a value (128 in this case) are set to the maximum (255, white), and all others are set to 0 (black). This is useful for separating features from the background.
4. **Displaying Images**: Matplotlib is used for displaying images. We use **cmap='gray'** to indicate that these are grayscale images.

Classes and Data Structures:

**Explanation of Each Class:**

1. **InventoryItem**: Manages stock levels for each item. Functions include updating stock levels and restocking items.
2. **Transaction**: Records details of each transaction, such as item sold, quantity, and total price.
3. **ReceiptProcessor**: Uses OpenCV for processing images of receipts. This could be expanded to use OCR technology for extracting text.
4. **Calculator**: Contains methods for calculating integrals and derivatives, useful for financial analyses such as cost projections or rate of change in sales.
5. **Database**: Simulates interaction with a database using a pandas DataFrame. In a real application, you would replace this with actual database queries.

Data processing:

Low stock Alert:

1. Load inventory data from a CSV file, including stock levels and thresholds.
2. Check each item to determine if its current stock is below the threshold.
3. Generate an alert for items with stock below the threshold.