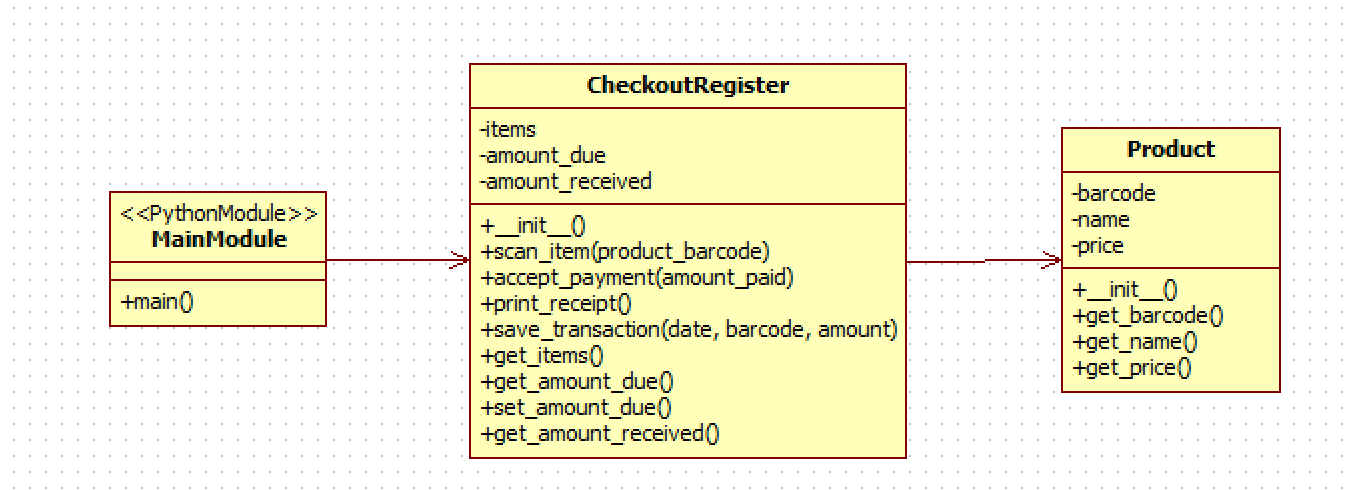
**Assignment 1 – Supermarket Self-Service Checkout**

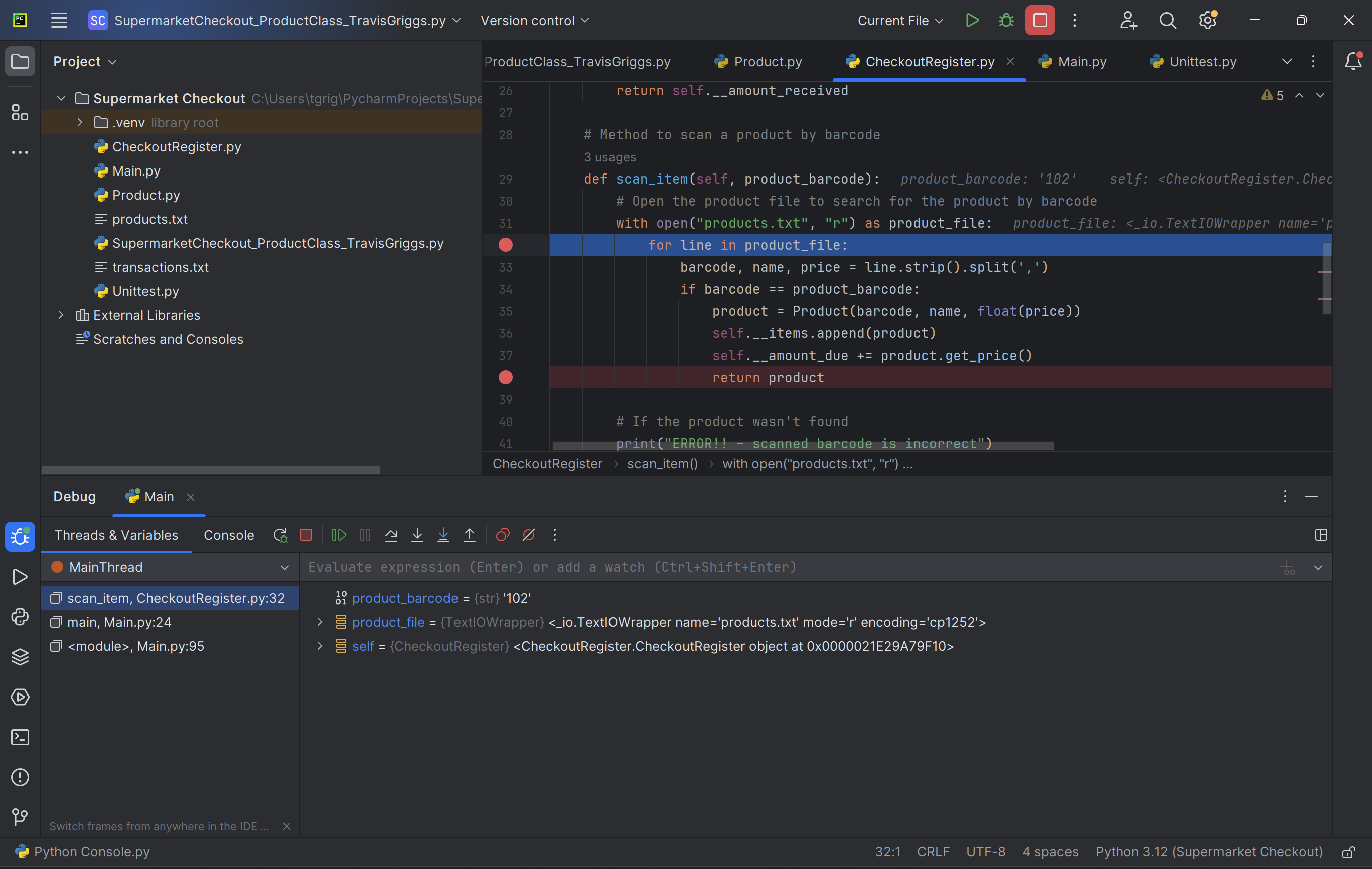
**Part 1 – Design (UML Models)**



**Part 3 –** **Testing (Test Cases and Unit Tests)**

**Debugging**

**Initial item scan:** barcode – 102, Name – Coke Zero, Price – $2.25



**Post stepping through** amount due = $2.25

A screenshot of a computer program

Description automatically generated

**Testing**

**A screenshot of a computer

Description automatically generated**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Description** | **Input** | **Expected Outcome** | **Actual Outcome** | **Pass/Fail** |
| CheckoutRegister\_scan\_item\_valid | Verify that scan\_item() returns the correct product when a valid barcode is scanned | Barcode: "12345" | Product Name: "Test Product"  Amount Due: 5.99 | Product Name: "Test Product"  Amount Due: 5.99 | Pass |
| CheckoutRegister\_scan\_item\_invalid | Verify that scan\_item() returns None when an invalid barcode is scanned | Barcode: "00000" | None | None | Pass |
| CheckoutRegister\_accept\_payment | Verify that accept\_payment() correctly calculates the remaining amount due after a payment | Amount Paid: 7.00 | Amount Received: 7.00  Remaining Amount Due: 3.00 | Amount Received: 7.00  Remaining Amount Due: 3.00 | Pass |
| Product\_init | Verify that the CheckoutRegister is initialized with default values | No input | Amount Due: 0, Amount Received: 0, Items: [] | Amount Due: 0, Amount Received: 0, Items: [] | Pass |
| Product\_get\_barcode | Verify that get\_barcode() returns the correct barcode for a product | Barcode: "12345" | Barcode: "12345" | Barcode: "12345" | Pass |
| Product\_get\_name | Verify that get\_name() returns the correct name for a product | Name: "Test Product" | Name: "Test Product" | Name: "Test Product" | Pass |
| Product\_get\_price | Verify that get\_price() returns the correct price for a product | Price: 5.99 | Price: 5.99 | Price: 5.99 | Pass |