**SEG3103 – Assignment 3**

**GUI Based Database Integration Testing**

**By:**

Daniel Godfrey, 4910083

**Presented to:**

Dr Stéphane Somé

Friday April 8th 2011

University of Ottawa

# Test Case Design

## Question 1.1

### Use Case: Connect



### Use Case: Disconnect



### Use Case: Acquisition



## 1.2

### Use Case: Connect

|  |  |  |
| --- | --- | --- |
| Scenario ID | Events | Description |
| S1 | 1, 2a, 2.a.1, 2.a.2, 2, 3 | User logs in with invalid code or password on first attempt. Logs in with valid code and password on second attempt. Human error scenario. |

### Use Case: Disconnect

|  |  |  |
| --- | --- | --- |
| Scenario ID | Events | Description |
| S2 | 1, 2, 3 | User disconnects. Normal course of events. |

### Use Case: Acquisition

|  |  |  |
| --- | --- | --- |
| Scenario ID | Events | Description |
| S3 | 1.a., 3,4,5,6,7,8 | Customer does not have a code and does not wish to register. Common for a “one time” customer. Customer purchases only a single product and only a single quantity. |
| S4 | 1.b, 1.b.1, 1.b.2, 1.b.3, 2, 3, 4, 5, 6, 7, 8 | Customer does not have a code and does wish to register. They then purchase a single product and only a single quantity. |
| S5 | 1, 2, 3.a, 3.a.1, 4, 5, 6, 7.a, 3,4,5,6,7, 8 | Customer with a code purchases 2 products. They purchase multiple quantities of the first product and only a single quantity of the second. |

## 1.3

**Test Case ID:** T1

**Scenario ID:** S1

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | Event | Reaction |
| 1 | Employee enters code: “124” | None |
| 2 | Employee enters password: “456” | None |
| 3 | Employee clicks “connect” | System fails to validate employee and displays an error message. |
| 4 | Employee enters code: “1” | None |
| 5 | Employee enters password: “one” | None |
| 6 | Employee clicks “connect” | System validates employee and terminal is presented ready to accept transactions |

**Pass Criteria:** System fails to validate employee on first login attempts. Employee is connected on second attempt and the terminal is ready to accept transactions.

**Test Case ID:** T2

**Scenario ID:** S2

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON
3. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | Event | Reaction |
| 1 | Employee clicks “disconnect” | System disconnects employee and the terminal is no longer ready to accept transactions |

**Pass Criteria:** Employee is disconnected and the terminal cannot accept transactions.

**Test Case ID:** T3

**Scenario ID:** S3

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON
3. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters product code: “1” | Product name “Beer” and price “1.5” are displayed on terminal. System updates bill. System updates product’s stock quantity. |
| 2 | Employee signals end of transaction. | System computes total and produces bill. |

**Pass Criteria:** Accurate bill is produced with total equal to “1.5”. Product Beer’s stock quantity is updated to 999. Terminal is ready to accept further transactions.

**Test Case ID:** T4

**Scenario ID:** S4

**Setup:**

1. Run baseline.sql
2. Customer with code: “333” does not exist
3. Turn Terminal ON
4. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters new customer code: “333” | None |
| 2 | Employ enters customer name: “Test” | System registers customer. System displays customers name “Test” on terminal |
| 3 | Employee enters product code: “1” | Product name “Beer” and price “1.5” are displayed on terminal. System updates bill. System updates product’s stock quantity. |
| 4 | Employee signals end of transaction. | System computes total and produces bill. |

**Pass Criteria:** Accurate bill is produced with total equal to “1.5”. Product Beer’s stock quantity is updated to 999. Terminal is ready to accept further transactions. A new customer is created with code: “333” and name: “Test”.

**Test Case ID:** T5

**Scenario ID:** S5

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON
3. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters customer code: “1” | Customers name “John Doe” is displayed in terminal |
| 2 | Employee enters product code: “1” | None |
| 2 | Employee enters quantity “2” of product | Product name “Beer” and price “3” are displayed in terminal. System updates the product’s stock quantity |
| 3 | Employee enters product code: “2” | Product name “ Peanuts“ and price “0.5” are displayed in terminal. System updates the product’s stock quantity |
| 4 | Employee signals end of transaction. | System computes total and produces bill. |

**Pass Criteria:** Accurate bill is produced with total equal to “3.5”. Product Beer’s stock quantity is updated to 998 and product Peanut’s stock quantity is updated to 999. Terminal is ready to accept further transactions.

**Test Case ID:** T6

**Scenario ID:** n/a (Invalid Input)

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON
3. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters customer code: “9” | Error Message is Displayed |

**Pass Criteria:** Error message is displayed, bill is not produced, and terminal is ready to accept further actions.

**Test Case ID:** T7

**Scenario ID:** n/a (Invalid Input)

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON
3. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters product code: “9” | Error Message is Displayed |

**Pass Criteria:** Error message is displayed, bill is not produced, and terminal is ready to accept further actions.

**Test Case ID:** T8

**Scenario ID:** n/a (Invalid Input)

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON
3. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters product code: “1” | None |
| 2 | Employee enters quantity “1001” of product | Error Message is Displayed |

**Pass Criteria:** Error message is displayed, bill is not produced, and terminal is ready to accept further actions.

**Test Case ID:** T9

**Scenario ID:** n/a (Boundary)

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON
3. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters product code: “1” | None |
| 2 | Employee enters quantity “1000” of product | Product name “Beer” and price “1500” are displayed in terminal. System updates the product’s stock quantity |
| 3 | Employee signals end of transaction. | System computes total and produces bill. |

**Pass Criteria:** Accurate bill is produced with total equal to “1500”. Product Beer’s stock quantity is updated to 0. Terminal is ready to accept further transactions.

**Test Case ID:** T10

**Scenario ID:** n/a (Boundary Condition)

**Setup:**

1. Successfully execute test case T9

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters product code: “1” | None |
| 2 | Employee enters quantity “1” of product | Error Message is Displayed |

**Pass Criteria:** Error message is displayed, bill is not produced, and terminal is ready to accept further actions.

**Test Case ID:** T11

**Scenario ID:** n/a (Invalid Input)

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON
3. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters product code: “1” | None |
| 2 | Employee enters quantity “0” of product | Error Message is Displayed |

**Pass Criteria:** Error message is displayed, bill is not produced, and terminal is ready to accept further actions.

**Test Case ID:** T12

**Scenario ID:** n/a (Boundary Value)

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters employee code: “” | Error Message is Displayed |

**Pass Criteria:** Error message is displayed, bill is not produced, and terminal is ready to accept further actions.

**Test Case ID:** T13

**Scenario ID:** n/a (Boundary Value)

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON
3. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters product code: “” | Error Message is Displayed |

**Pass Criteria:** Error message is displayed, bill is not produced, and terminal is ready to accept further actions.

**Test Case ID:** T14

**Scenario ID:** n/a (Boundary Value)

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON
3. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters customer code: “” | Error Message is Displayed |

**Pass Criteria:** Error message is displayed, bill is not produced, and terminal is ready to accept further actions.

**Test Case ID:** T15

**Scenario ID:** n/a (Invalid Input)

**Setup:**

1. Run baseline.sql
2. Turn Terminal ON
3. Employee with code: “1” is logged in

**User Events:**

|  |  |  |
| --- | --- | --- |
| Step | External Event | Reaction |
| 1 | Employee enters product code: “1” | None |
| 2 | Employee enters quantity “-1” of product | Error Message is Displayed |

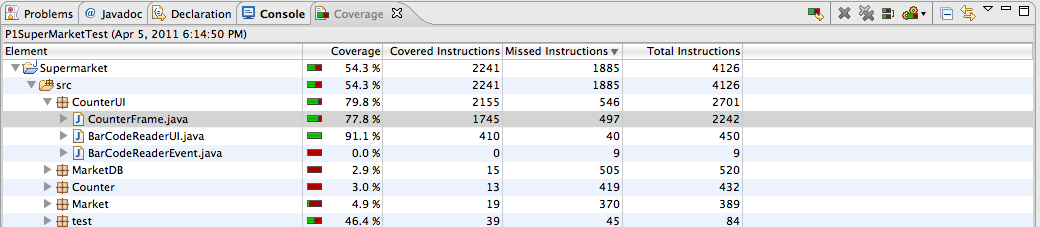
**Pass Criteria:** Error message is displayed, bill is not produced, and terminal is ready to accept further actions.

# Independent Testing of GUI Layer (Class CounterFrame)

**Test Case:** T1

**Result:** PASS

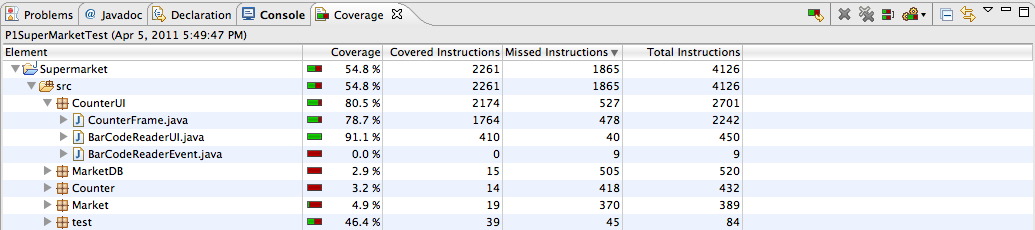
**Code Coverage:** 77.8%



**Test Case:** T2

**Result:** PASS

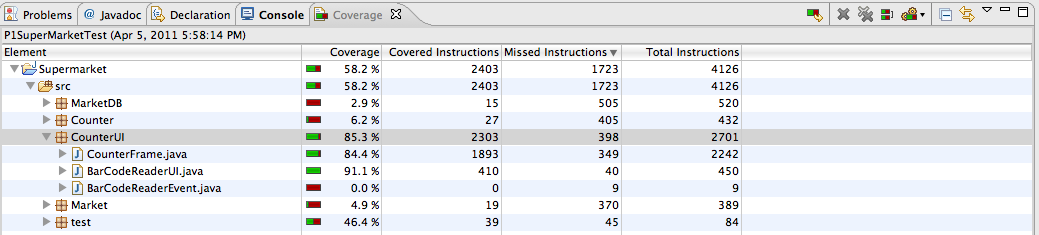
**Code Coverage:** 78.7%

****

**Test Case:** T3

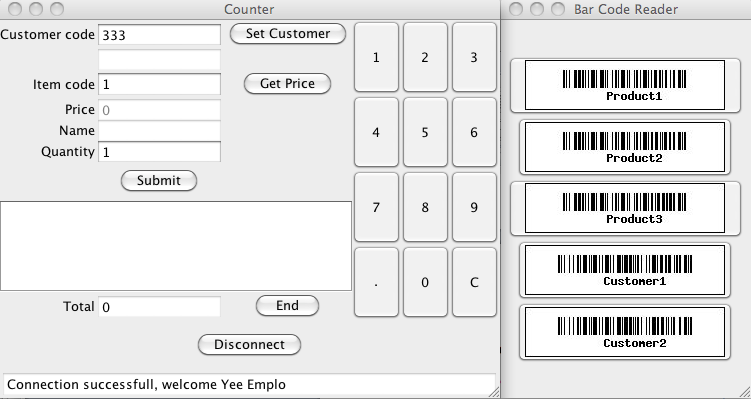
**Result:** Pass

**Code Coverage:** 84.4%

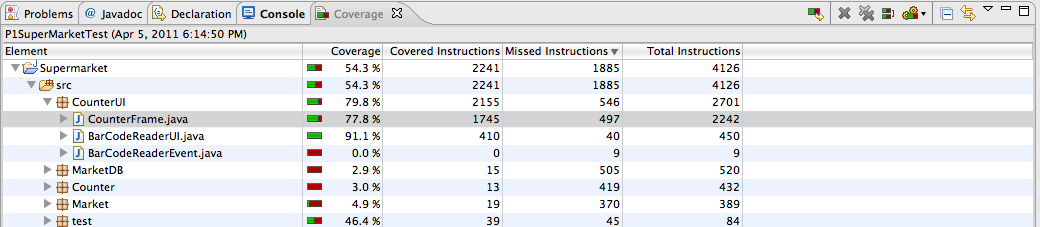


**Test Case:** T4

**Result:** FAIL – Cannot Enter New Customer Name

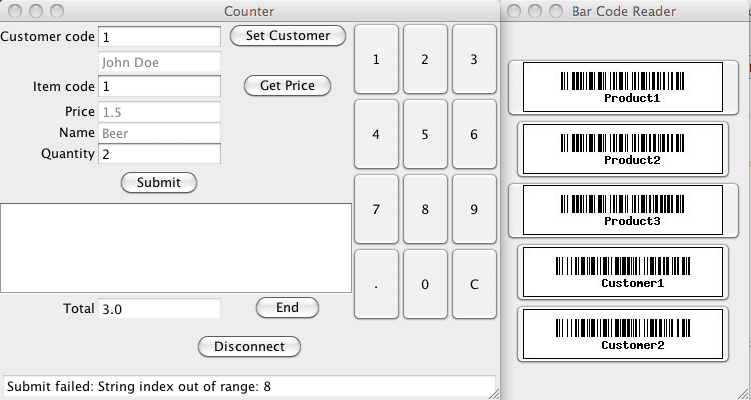


**Code Coverage:** 77.8%

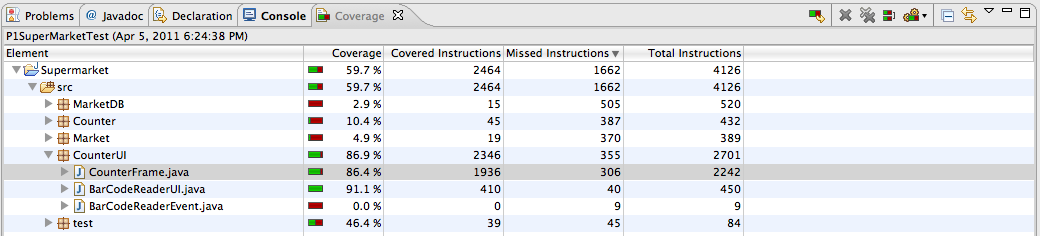


**Test Case:** T5

**Result:** Fail – Bill not updated when adding multiple quantities of an item but adds to total.

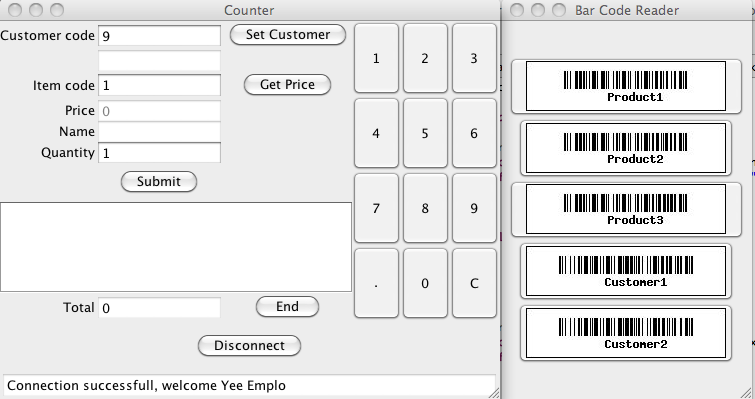


**Code Coverage:** 86.4%

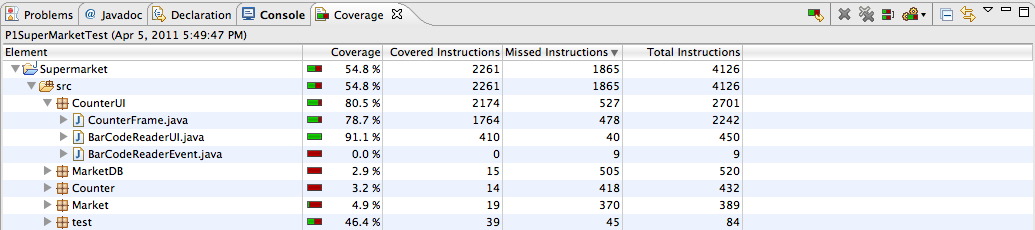


**Test Case:** T6

**Result:** Fail – Entering invalid customer codes do not produce an error message.

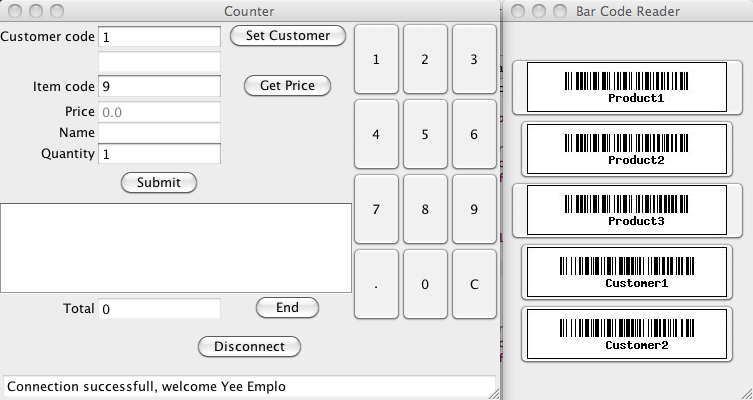


**Code Coverage:** 78.7%

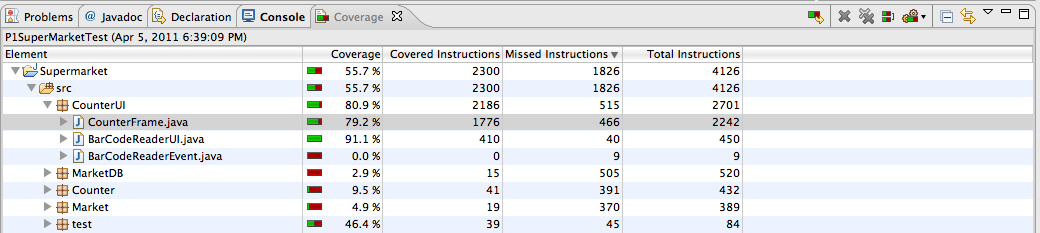
****

**Test Case:** T7

**Result:** Fail – Entering invalid item code does not produce an error message.

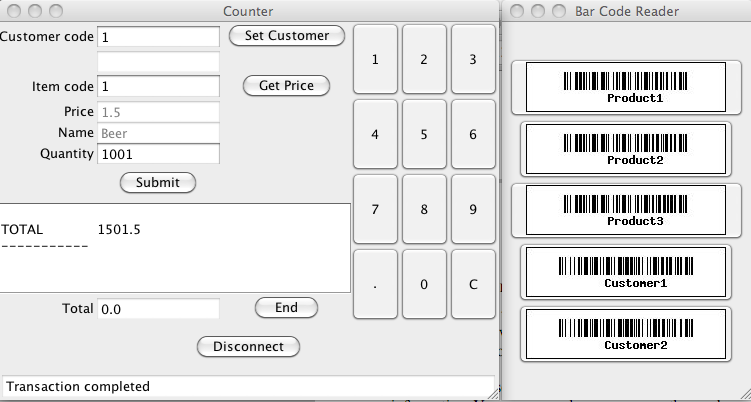


**Code Coverage:** 79.2%

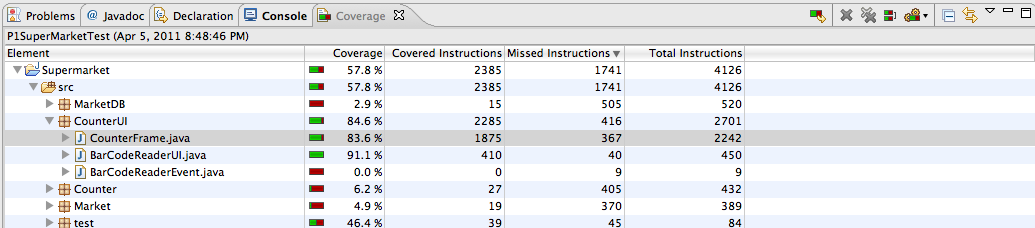


**Test Case:** T8

**Result:** Fail – Item does not appear on bill but total is calculated. Transaction is processed when it shouldn’t be.

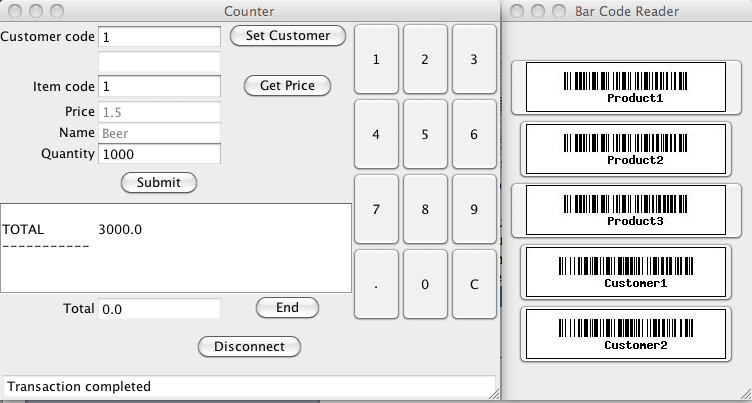


**Code Coverage:** 83.6%

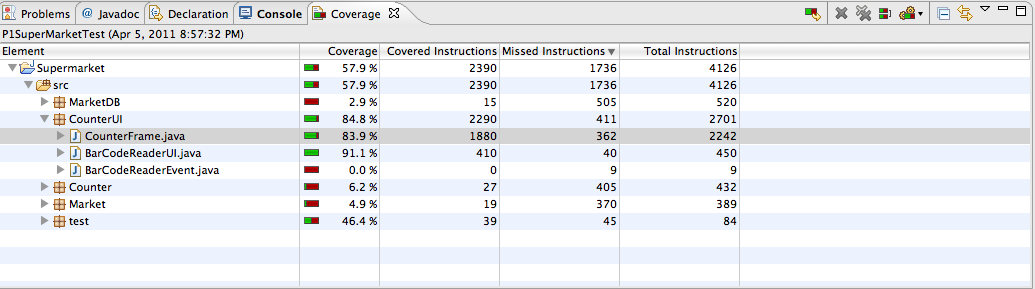


**Test Case:** T9

**Result:** Fail – Item not added to bill but total is calculated.



**Code Coverage:** 83.9%

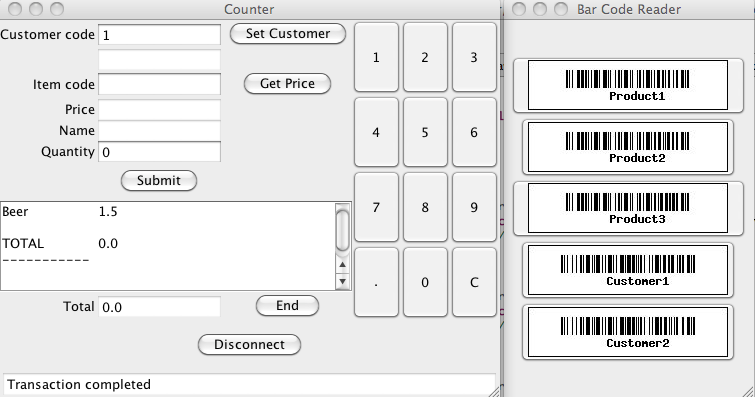
****

**Test Case:** T10

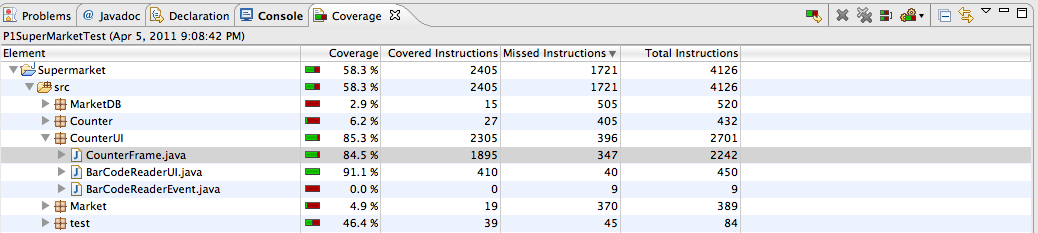
*Omitted As It Does Not Apply At This Iteration Since Actions Do Not Persist.*

**Test Case:** T11

**Result:** Fail – Item is added to bill with total balance as 0.



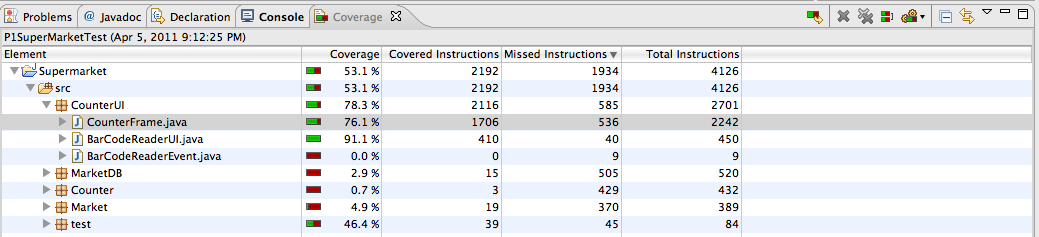
**Code Coverage:** 84.5%



**Test Case:** T12

**Result:** PASS

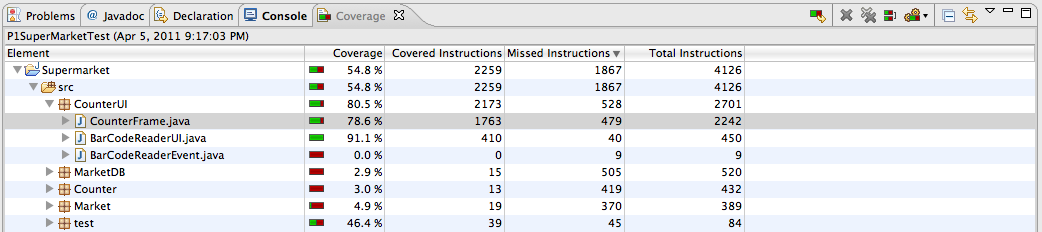
**Code Coverage:** 76.1%

****

**Test Case:** T13

**Result:** PASS

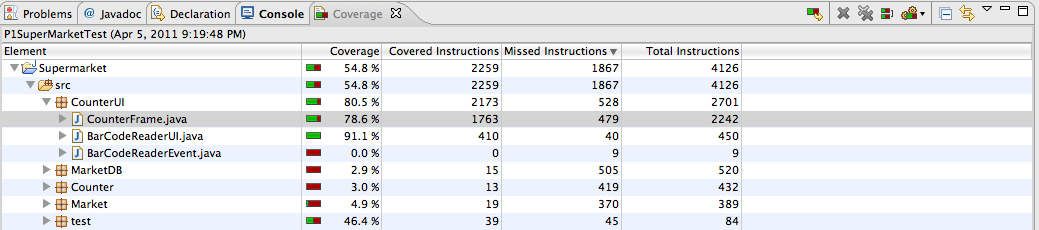
**Code Coverage:** 78.6%



**Test Case:** T14

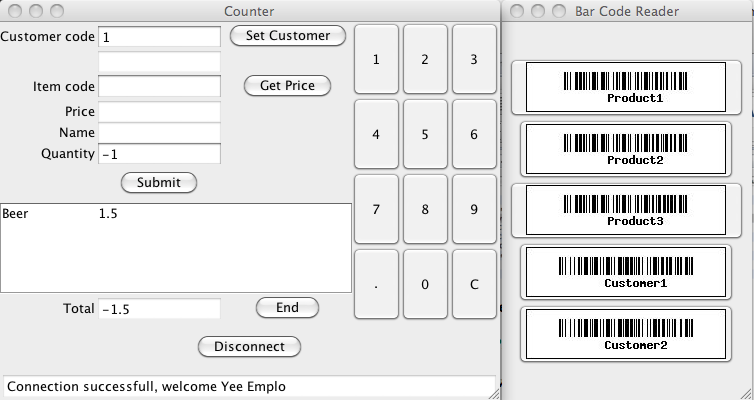
**Result:** PASS

**Code Coverage:** 78.6

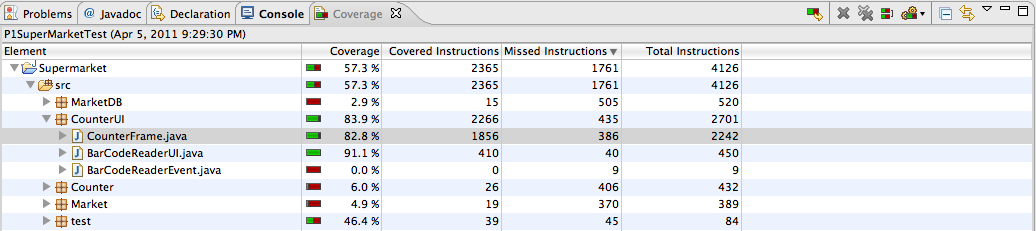


**Test Case:** T15

**Result:** Fail – System allows for negative quantities to be submitted. Results in negative total.



**Code Coverage:** 82.8%

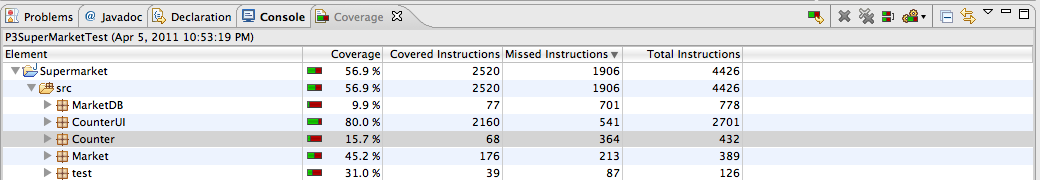


# Integration of GUI and Application Layers (CounterUI, Counter, & Market Packages)

**Test Case:** T1

**Result:** PASS

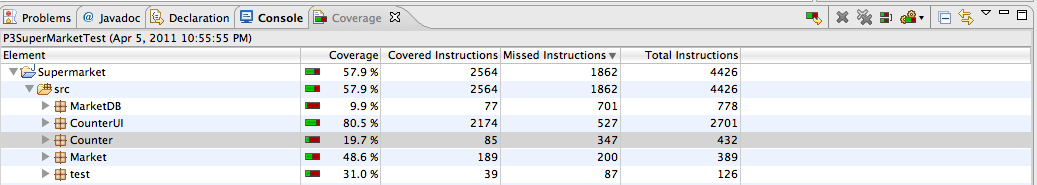
**Code Coverage:** 80%, 15.7%, 45.2%



**Test Case:** T2

**Result:** PASS

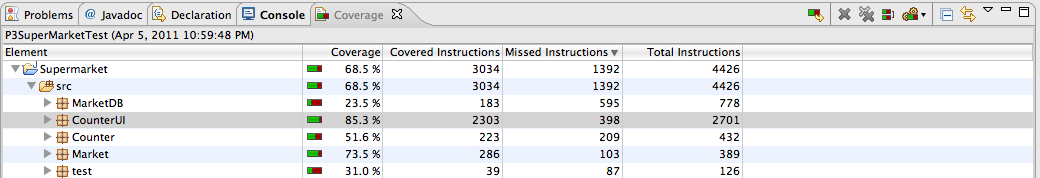
**Code Coverage:** 80.5%, 19.7%, 48.6%

****

**Test Case:** T3

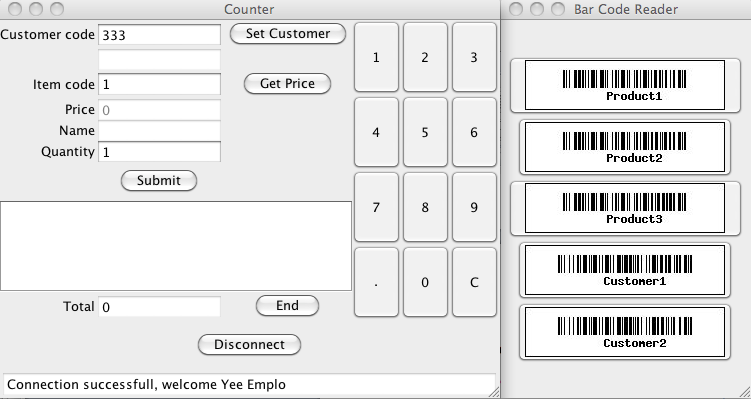
**Result:** Pass

**Code Coverage:** 85.3%, 51.6%, 73.5%

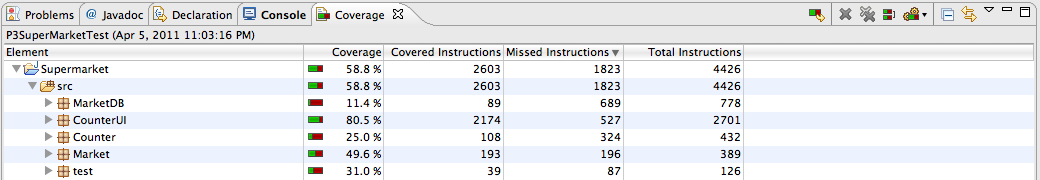


**Test Case:** T4

**Result:** FAIL – Cannot Enter New Customer Name

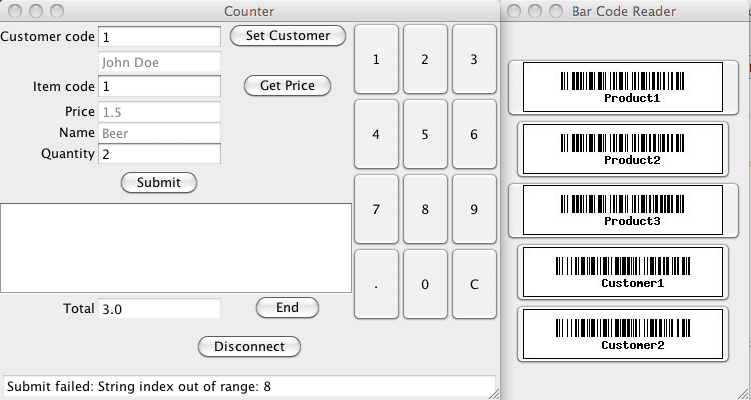


**Code Coverage:** 80.5%, 25%, 49.6%

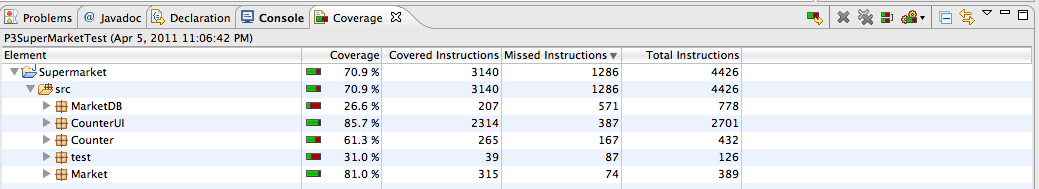


**Test Case:** T5

**Result:** Fail – Bill not updated when adding multiple quantities of an item but adds to total.



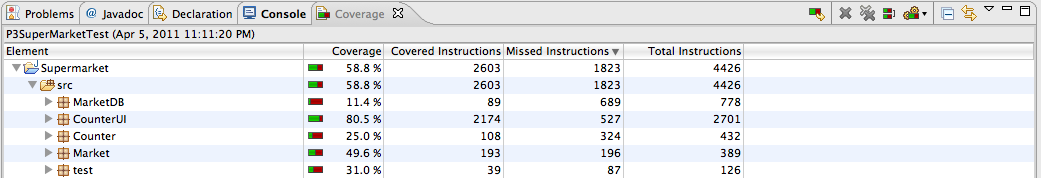
**Code Coverage:** 85.7%, 61.3%, 81%



**Test Case:** T6

**Result:** PASS

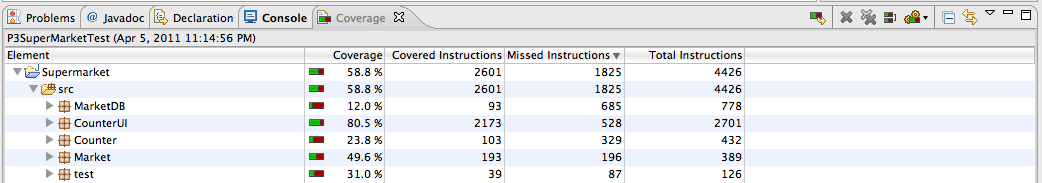
**Code Coverage:** 80.5%, 25%, 49.6%



**Test Case:** T7

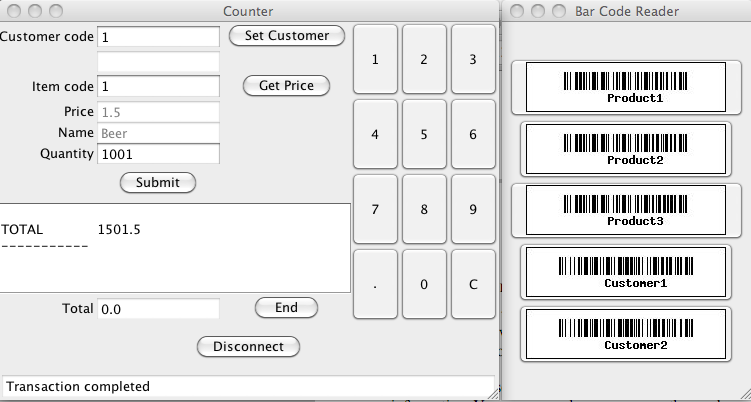
**Result:**  PASS

**Code Coverage:** 80.5%, 23.8%, 49.6%

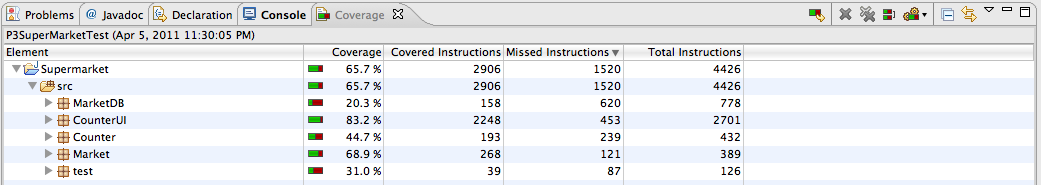


**Test Case:** T8

**Result:** Fail – Item does not appear on bill but total is calculated. Transaction is processed when it shouldn’t be.

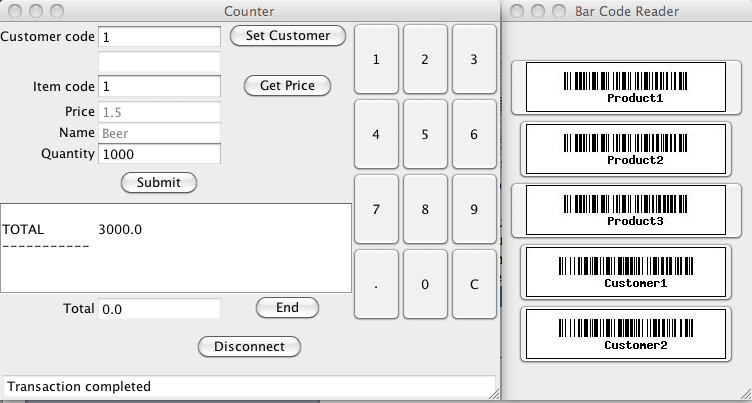


**Code Coverage:** 83.2%, 44.7%, 68.9%

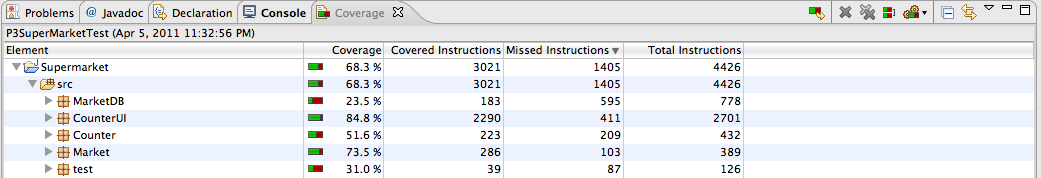


**Test Case:** T9

**Result:** Fail – Item not added to bill but total is calculated.



**Code Coverage:** 84.8%, 51.6%, 73.5%

****

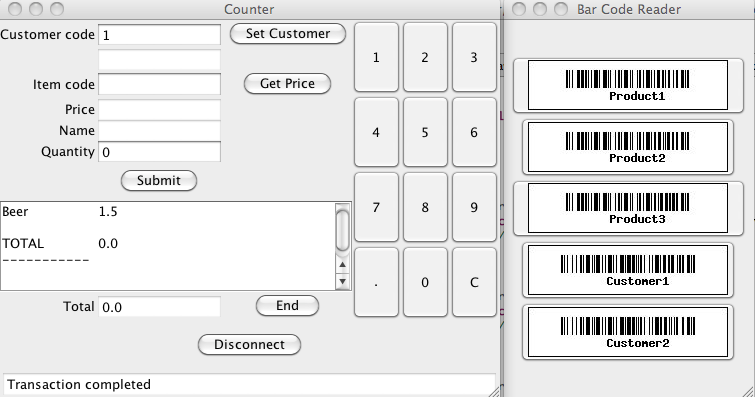
**Test Case:** T10

**Result:** Fail – Setup Failed

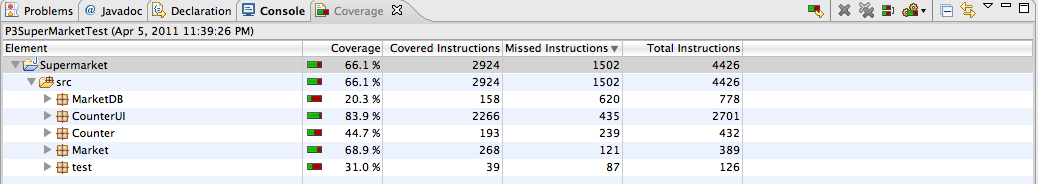
**Code Coverage:** n/a

**Test Case:** T11

**Result:** Fail – Item is added to bill with total balance as 0.



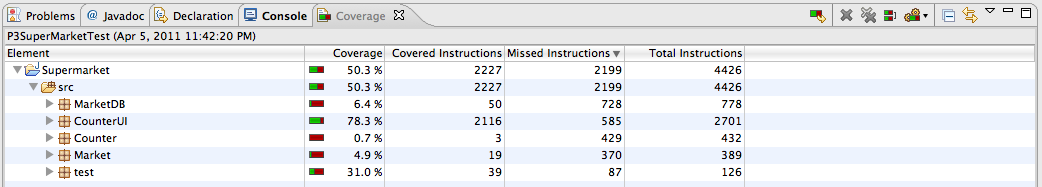
**Code Coverage:** 83.9%, 44.7%, 68.9%



**Test Case:** T12

**Result:** PASS

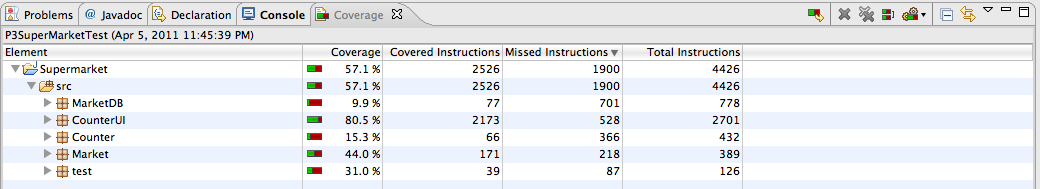
**Code Coverage:** 78.3%, 0.7%, 4.9%

****

**Test Case:** T13

**Result:** PASS

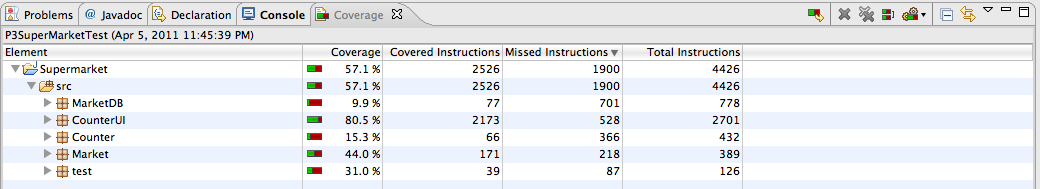
**Code Coverage:** 80.5%, 15.3%, 44%



**Test Case:** T14

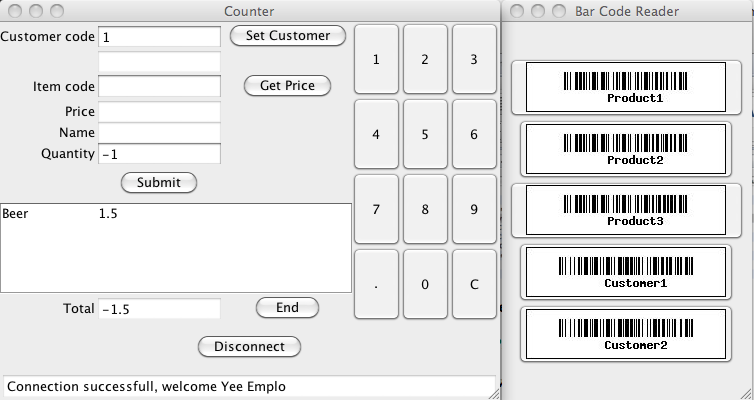
**Result:** PASS

**Code Coverage:** 80.5%, 15.3%, 44%

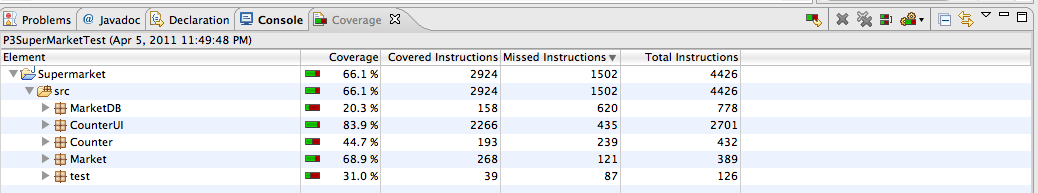


**Test Case:** T15

**Result:** Fail – System allows for negative quantities to be submitted. Results in negative total.



**Code Coverage:** 83.9%, 44.7%, 68.9%

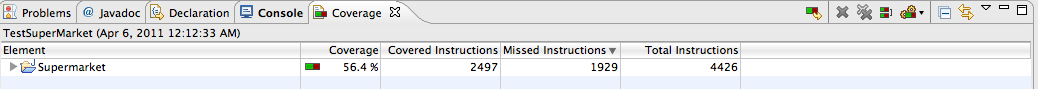


# Complete Integration (SuperMarket Package)

**Test Case:** T1

**Result:** PASS

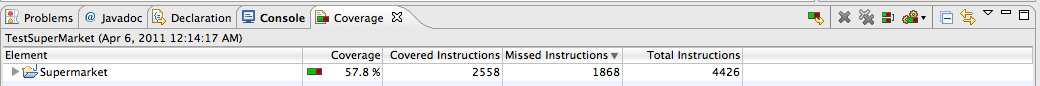
**Code Coverage:** 56.4%



**Test Case:** T2

**Result:** PASS

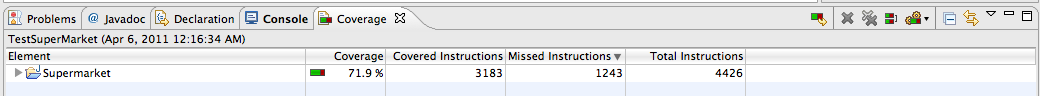
**Code Coverage:** 57.8%

****

**Test Case:** T3

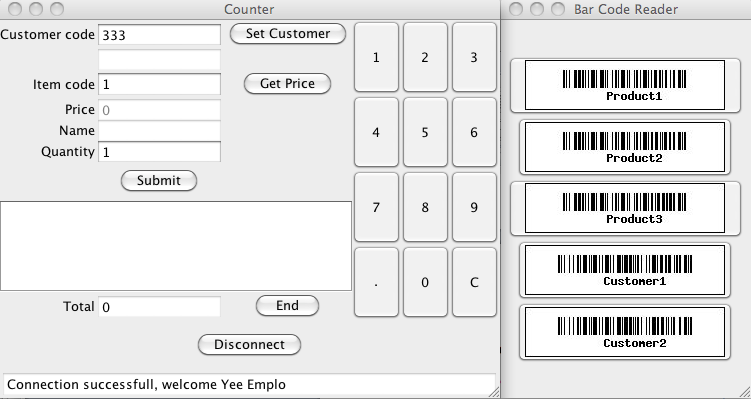
**Result:** Pass

**Code Coverage:** 71.9%

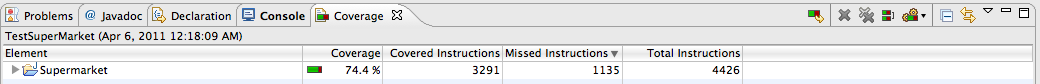


**Test Case:** T4

**Result:** FAIL – Cannot Enter New Customer Name

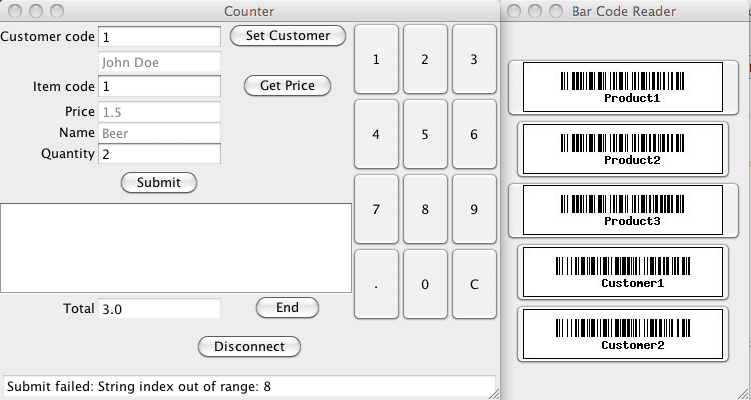


**Code Coverage:** 74.4%

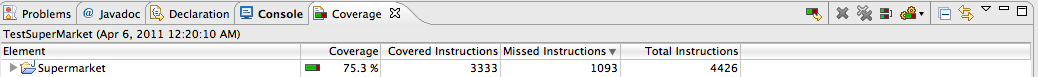


**Test Case:** T5

**Result:** Fail – Bill not updated when adding multiple quantities of an item but adds to total.



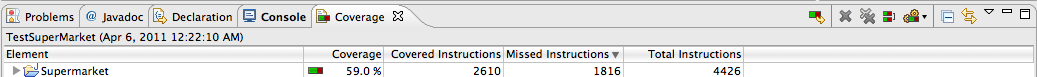
**Code Coverage:** 75.3%



**Test Case:** T6

**Result:** PASS

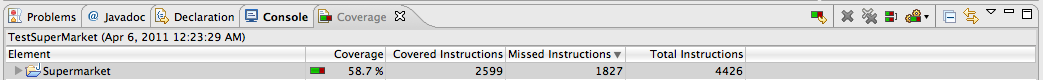
**Code Coverage:** 59.0%



**Test Case:** T7

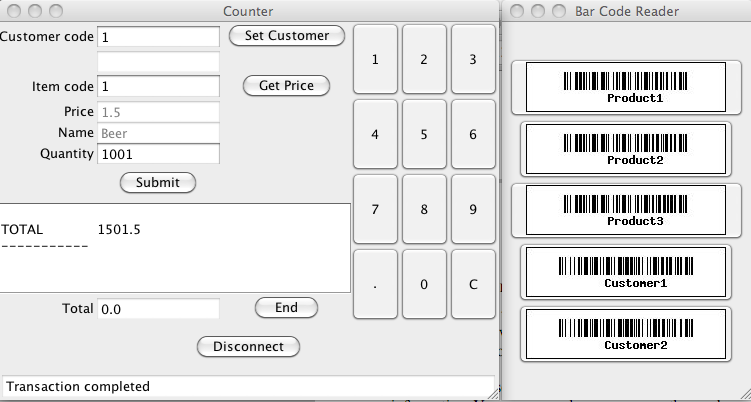
**Result:**  PASS

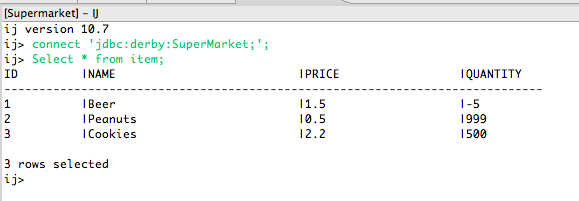
**Code Coverage:** 58.7%



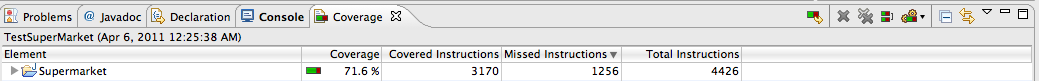
**Test Case:** T8

**Result:** Fail – Item does not appear on bill but total is calculated. Transaction is processed when it shouldn’t be.



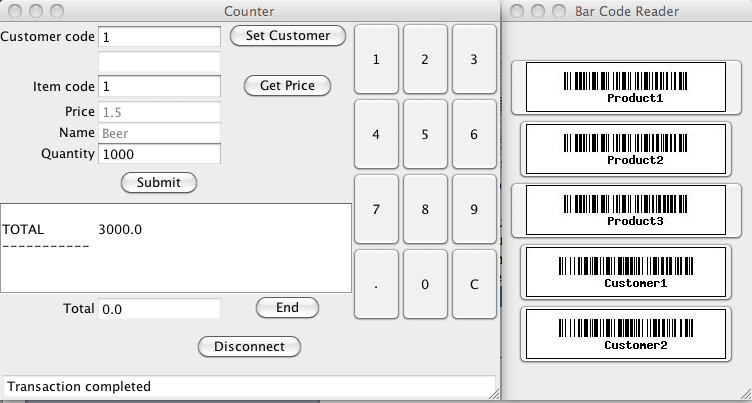


**Code Coverage:** 71.6%

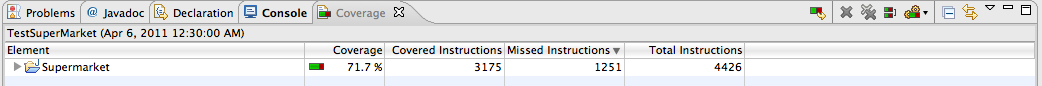


**Test Case:** T9

**Result:** Fail – Item not added to bill but total is calculated.

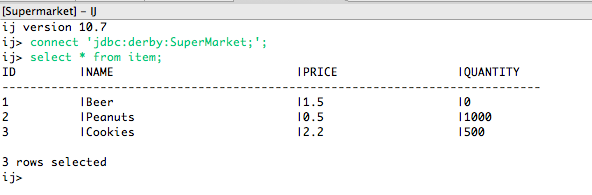


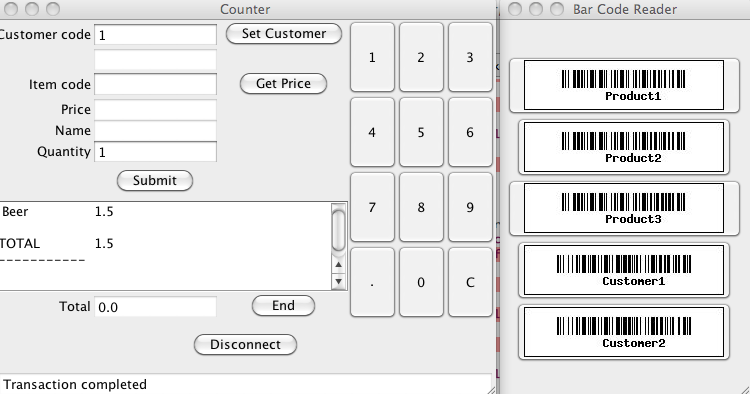
**Code Coverage:** 71.7;

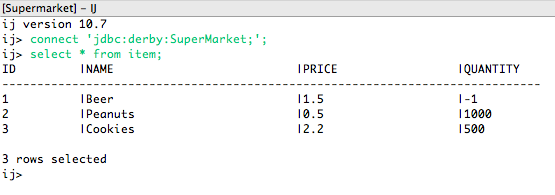
****

**Test Case:** T10

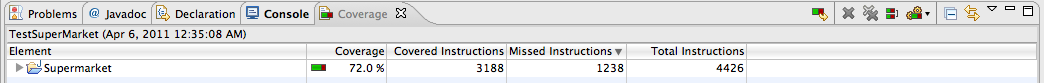
**Result:** Fail – Setup failed plus sale was allowed with 0 inventory





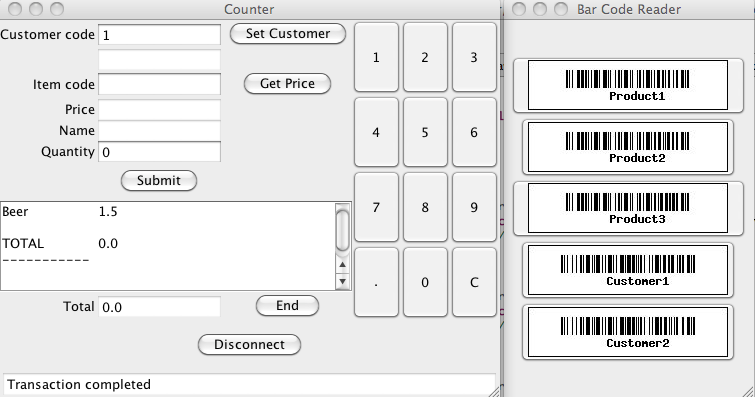


**Code Coverage:** 72%

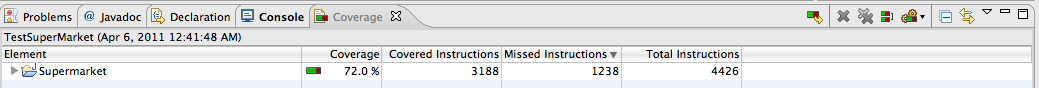


**Test Case:** T11

**Result:** Fail – Item is added to bill with total balance as 0.



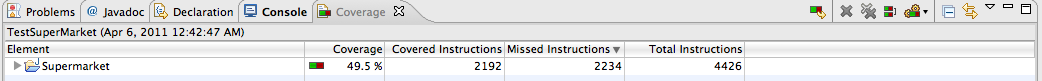
**Code Coverage:** 72%



**Test Case:** T12

**Result:** PASS

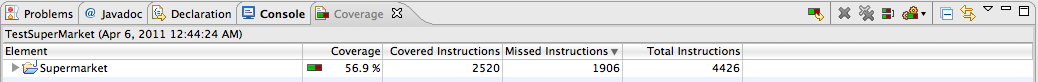
**Code Coverage:** 49.5%

****

**Test Case:** T13

**Result:** PASS

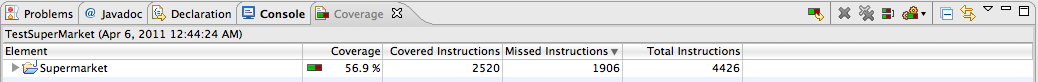
**Code Coverage:** 56.9%



**Test Case:** T14

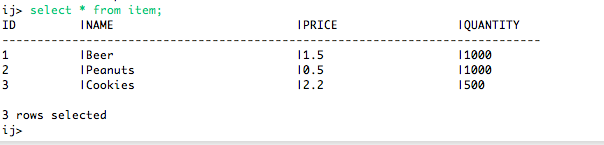
**Result:** PASS

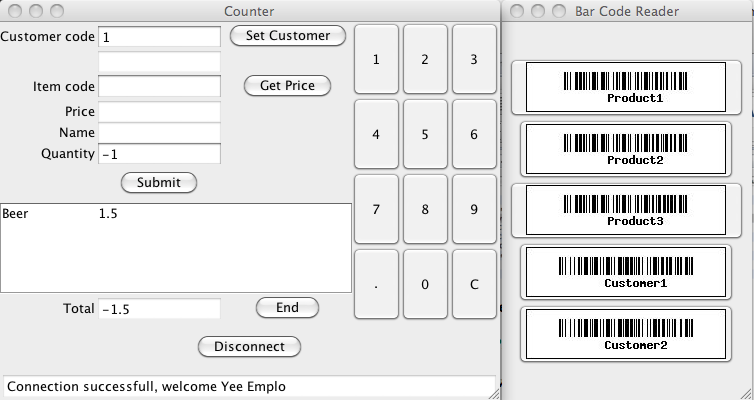
**Code Coverage:** 56.9

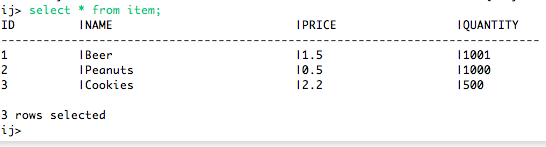


**Test Case:** T15

**Result:** Fail – System allows for negative quantities to be submitted. Results in negative total.







**Code Coverage:** 72%

