# Requirements

|  |  |
| --- | --- |
| R1 | A user can be registered with matching password and verified password |
| R2 | A user can log in with valid credentials |
| R3 | A recipient can view recipient orders |
| R4 | A recipient can view order progress including transactions against an their own individual order |
| R5 | A driver can view all orders |
| R6 | A driver can add a transaction against an order |
| R7 | A driver can remove a transaction from an order |
| R8 | A seller can create a parcel with a valid value for name, type, weight, and seller |
| R9 | A seller can edit a parcel with a valid value for name, type, weight, and seller |
| R10 | A seller can delete a parcel, which subsequently removes the parcel from any existing order it was on |
| R11 | A seller can create an order with a valid recipient and seller |
| R12 | A seller can edit an order with a valid recipient and seller |
| R13 | A seller can delete an order, which subsequently removes all transactions that existed against that order |
| R14 | A seller can view driver metrics to find total delivery counts and average delivery durations (in days) per driver |
| R15 | A seller can view a list of all registered users |
| R16 | A seller can view details of a specific registered user |
| R17 | An order has zero or more parcels |
| R18 | An order has zero or more transactions |
| R19 | An order has a recipient and seller |
| R20 | An order has zero or one driver |
| R21 | A user has zero or more orders |
| R22 | A user has a role recipient, driver, or seller |
| R23 | An order’s driver must be “None” when no “Picked up” transaction exists |
| R24 | When an order is “picked up”, the driver adding the transaction becomes the named driver on the order |
| R25 | When a “Picked up” transaction is removed, the named driver reverts to “None” |
| R26 | An order’s completed date must be blank when no “Dropped off” transaction exists |
| R27 | When an order is “dropped off”, the order completed date is filled out |
| R28 | When a “Dropped off” transaction is removed, the order completed date is cleared |
| R29 | A seller can view all orders |
| R30 | A seller can view all parcels |
| R31 | A seller can view details of a specific parcel |
| R32 | A seller can add a parcel to an order with a specific quantity |
| R33 | A driver can view order progress including transactions against any order |
| R34 | A seller can view order progress including transactions against any order |

# Test cases

## Normal values

N1 username = “TestUser”, password1 = “pass”, password2 = “pass” (R1)  
N2 username = “recipient”, password = “123” (R2)  
N3 name = “Book”, type = “Large”, weight = 50, sellerID = 3 (R8)  
N4 name = “Pen”, type = “Small”, weight = 100, sellerID = 2 (R9)  
N5 recipientID = 1, sellerID = 3 (R11, R19)  
N6 recipientID = 2, sellerID = 1 (R12, R19)  
N7 orderID = 1, parcelID = 1, quantity=2 (R32)

## Invalid values

I1 username = “TestUser” password1 = “pass”, password2 = “wrong” (R1)  
I2 username = “seller”, password = “abc” (R2)  
I3 name = “Book”, type = “Large”, weight = -1, sellerID = 3 (R8)  
I4 name = “Pen”, type = “Small”, weight = 10, sellerID = -1 (R8)  
I5 name = ”Book”, type = ”Large”, weight = -1, sellerID = 3 (R9)  
I6 name = ”Pen”, type = ”Small”, weight = 10, sellerID = -1 (R9)  
I7 recipientID = -1, sellerID = 1 (R11, R19)  
I8 recipientID = 1, sellerID = -1 (R11, R19)  
I9 recipientID = -1, sellerID = 1 (R12, R19)  
I10 recipientID = 1, sellerID = -1 (R12, R19)  
I11 orderID = -1, parcelID = 1, quantity = 2 (R32)  
I12 orderID = 1, parcelID = -1, quantity = 2 (R32)  
I13 orderID = 1, parcelID = 1, quantity = -1 (R32)

## Boundary values

B1 There are 0 parcels in an order (R17)  
B2 There are 0 transactions against an order (R18)  
B3 An order has no driver (R20)  
B4 An order has 1 driver (R20)  
B5 A user has 0 orders (R21)  
B6 A user has 1 order (R21)  
B7 A user has no role (R22)  
B8 A user has 1 role (R22)

## Special cases

S1 Recipient (Recipient) views list of (Recipient’s) orders (R3)  
S2 Recipient (Recipient) views details of one of (Recipient’s) orders (R4)  
S3 Driver (Driver) views list of all orders (R5)  
S4 Driver (Driver) adds transaction “Picked up” to an order (R6)  
S5 Driver (Driver) adds transaction “Dropped off” to an order (R6)  
S6 Driver (Driver) removes transaction “Picked up” from an order (R7)  
S7 Driver (Driver) removes transaction “Dropped off” from an order (R7)  
S8 Seller (Seller) deletes a parcel (R10)  
S9 Seller (Seller) deletes an order (R13)  
S10 Seller (Seller) views driver metrics for driver totals and average delivery days (R14)  
S11 Seller (Seller) views list of all registered users (R15)  
S12 Seller (Seller) views details of a single registered user (R16)  
S13 An orders driver is updated when a “Picked up” transaction changes (R23, R24, R25)  
S14 An orders completed date is updated when a “Dropped off” transaction changes (R26, R27, R28)  
S15 Driver (Driver) views details of an order (R33)  
S16 Seller (Seller) views details of an order (R34)  
S17 Seller (Seller) views list of all orders (R29)  
S18 Seller (Seller) views list of all parcels (R30)  
S19 Seller (Seller) views details of a specific parcel (R31)

# Test plan

\* Requires manual testing due to dependency injection or unable to unit test this action

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Instruction (Satisfies test cases) | Expected result(s) | Actual result(s) |
| T1 | Create a user with username “TestUser” and matching passwords “pass”. Then log in to the system. (N1, N2) | There is a user called “TestUser” and can be logged in to, taking to Recipient landing page. | RegisterBeanTest  .testRegisterValid()  PASS  Figure 2 RegisterBean Test Results |
| T2 | Create a user with username “TestUser”, password1= “pass” and password2= “wrong”. (I1) | Registration attempt will fail validation and an error message “Login credentials are not correct” will be displayed. | RegisterBeanTest  .testRegisterInvalid()  PASS  Figure 2 RegisterBean Test Results |
| T3 | Create a parcel with values: name = “Book”, type = “Large”, weight = -1, sellerID = 3. (I3) | Creation of parcel fails and an error message requesting the value must be an integer of one or more digits will be displayed. | SellerBeanTest  .testCreateParcelInvalid  Weight()  PASS  Figure 4 SellerBean test results |
| T4 | Create a parcel with values: name = “Pen”, type = “Small”, weight = 10, sellerID = -1. (I4) | Creation of parcel fails, and a null pointer exception is experienced. | SellerBeanTest  .testCreateParcelInvalid  SellerId()  PASS  Figure 4 SellerBean test results |
| T5 | Edit parcel with ID1 with values: name = “Book”, type = “Large”, weight = -1, sellerID = 3. (I5) | Update of parcel fails and an error message requesting the value must be an integer of one or more digits will be displayed. | SellerBeanTest  .testEditParcelInvalidWeight()  PASS  Figure 4 SellerBean test results |
| T6 | Edit parcel with ID1 with values: name = “Pen”, type = “Small”, weight = 10, sellerID =  -1. (I6) | Update of parcel fails, and a null pointer exception is experienced. | SellerBeanTest  .testEditParcelInvalidSellerId()  PASS  Figure 4 SellerBean test results |
| T7 | Create parcel with values: name= “Book”, type= “Large”, weight=50, sellerID=3, then edit the parcel with values: name= “Pen”, type= “Small”, weight=100, sellerID=2.  Create order with values: recipientID=1, sellerID=3. Edit order with values: recipientID=2, sellerID=1.  Add parcel with ID{just created} to order ID{just created} with quantity 2. (N3, N4, N5, N6, N7) | There is a parcel with values: name= “Pen”, type= “Small”, weight=100, sellerID=2, there is an order with recipientID=2, sellerID=1 and the parcel is added to the order with quantity 2. | SellerBeanTest  .testCreateEditParcel  CreateEditOrder  AddParcelToOrderValid()  PASS  Figure 4 SellerBean test results |
| T8 | Create an order with values: recipientID = -1, sellerID = 1. (I7) | Create order fails and a null pointer exception is experienced. | SellerBeanTest  .testCreateOrderInvalid  RecipientID()  PASS  Figure 4 SellerBean test results |
| T9 | Create an order with values: recipientID = 1, sellerID = -1. (I8) | Create order fails and a null pointer exception is experienced. | SellerBeanTest  .testCreateOrderInvalid  SellerId()  PASS  Figure 4 SellerBean test results |
| T10 | Create a temp order, Edit order with ID{just created} with values: recipientID = -1, sellerID = 1. (I9) | Edit order fails and a null pointer exception is experienced. | SellerBeanTest  .testEditOrderInvalid  RecipientId()  PASS  Figure 4 SellerBean test results |
| T11 | Create a temp order, Edit order with ID{just created} with values: recipientID = 1, sellerID = -1. (I10) | Edit order fails and a null pointer exception is experienced. | SellerBeanTest  .testEditOrderInvalidSellerId()  PASS  Figure 4 SellerBean test results |
| T12 | Add parcel with ID1 to order with ID= -1 and quantity 1. (I11) | ParcelID1 is not added to orderID-1, and a null pointer exception is experienced. | SellerBeanTest  .testAddParcelInvalidOrderId()  PASS  Figure 4 SellerBean test results |
| T13 | Add parcel with ID-1 to order with ID1 and quantity 1. (I12) | ParcelID-1 is not added to orderID1, and a null pointer exception is experienced. | SellerBeanTest  .testAddParcelInvalidParcelId()  PASS  Figure 4 SellerBean test results |
| T14 | Add parcel with ID1 to order with ID1 and quantity -1. (I13) | ParcelID1 is not added to orderID1 and an error message is displayed requesting a positive integer value. | SellerBeanTest  .testAddParcelInvalidQuantity()  FAIL: The parcel is added and shows as a negative quantity. (Figure 1 Adding negative quantity parcel to order)  PASS (After modifications)  Figure 4 SellerBean test results |
| T15 | Order with ID=2 has no parcels. View details of order with ID=2. (B1) | Order with ID=2 has no parcels listed. | SellerBeanTest  .testFindOrderParcels()  PASS  Figure 4 SellerBean test results |
| T16 | Order with ID=2 has no transactions. View details of order with ID=2 to check for transactions listed. (B2) | Order with ID=2 has no transactions listed. | SellerBeanTest  .testFindOrderTransactions()  PASS  Figure 4 SellerBean test results |
| T17 | Order with ID=2 has no transactions. View details of order with ID=2 to check its named driver. (B3) | Order with ID=2 has “None” in the driver section. | SellerBeanTest  .testFindOrderTransactions  NoTransactionNoDriver()  PASS  Figure 4 SellerBean test results |
| T18 | Order with ID=2 has at least 1 transaction “Picked up”. (B4) | Order with ID=2 has a named driver that is not “None”. | SellerBeanTest  .testFindOrderTransactions  SomeTransactions()  PASS  Figure 4 SellerBean test results |
| T19 | Register user with values: name = “RecipientWithNoOrders”, matching passwords = “pass”. View recipient orders. (B5) | User named RecipientWithNoOrders has an order total of 0. | RecipientBeanTest  .testViewRecipientOrders  NoOrders()  PASS  Figure 5 RecipientBean test results |
| T20 | Recipient views recipient orders. (B6) | Recipient has an order total greater than or equal to 1 and the order details are valid values. | RecipientBeanTest  .testViewRecipientOrders()  PASS  Figure 5 RecipientBean test results |
| T21\* | A guest user (Not logged in) views the login page. (B7) | Guest may view login page without any missing public assets, including stylesheets and images. | Manual test  PASS  Figure 7 Guest views Login |
| T22\* | A guest user (Not logged in) attempts to navigate to “localhost:8080/ParcelTracker/faces/  viewOrder\_Recipient.xhtml”. (B7) | Filter redirects user back to login page, no secure assets are available to the non-logged in user. | Manual test  PASS  Figure 8 Guest attempts to view restricted page but is returned to login screen |
| T23\* | Recipient attempts to navigate to “localhost:8080/ParcelTracker/faces/  viewOrder\_Recipient.xhtml” while being logged in. (B8) | Expected page viewOrder\_Recipient.xhtml loads correctly. | Manual test  PASS  Figure 9 Recipient views order details page while logged in without being redirected back to login screen |
| T24 | Seller views all users. (S11) | Returns a list of 1 or more users. | SellerBeanTest  .testViewAllUsers()  PASS  Figure 4 SellerBean test results |
| T25 | Seller views details of user with ID=1. (S12) | Returns details of Recipient (User with ID=1). | SellerBeanTest  .testSetAndGetUserDetails()  PASS  Figure 4 SellerBean test results |
| T26 | - Create a recipient user with username “Jimmy”.  - Create a seller user with username “Sally”.  - **Create order** where “Jimmy” is the recipient and “Sally” is the seller.  **-** Run **View recipient orders** when using the application as “Jimmy”.  - View details of the order where Jimmy is the recipient and Sally is the seller. (S1, S2) | Returns a list Jimmy’s orders containing at least the one where Sally is the seller. Then returns details of the selected order without errors. | SellerBeanTest  .textViewRecipientOrdersSpecial()  PASS  Figure 4 SellerBean test results |
| T27 | - Create a recipient user with username “Jimmy”.  - Create a seller user with username “Sally”.  - **Create order** where “Jimmy” is the recipient and “Sally” is the seller.  - Create a driver user with username “Bob”.  - **Add transaction** “Picked up” to the order while using the application as “Bob” (Driver).  - View this order’s details.  - **Add transaction** “Dropped off” to the order while using the application as “Bob” (Driver).  - View this order’s details.  - **Remove transaction** “Dropped off” from the order while using the application as “Bob” (Driver).  - Bob (Driver) Views all orders.  - **View order progress** against the order created. (S3, S4, S5, S6, S7, S13, S14, S15) | First order details represent the order with Bob as the named driver.  Second order details represent the order with a completed date filled in.  Final order details show “Jimmy” is the recipient, “Sally” is the seller, “Bob” is the driver and shows 1x transaction of “Picked up”, and no other transactions. | DriverBeanTest  .testAddAndRemoveTransaction()  PASS  Figure 6 DriverBean test results |
| T28 | - Create a seller user with username “Sally”.  - **Create parcel** with name = “Old item”, type = “Book”, weight=500, sellerID = (ID of Sally).  - **Delete parcel** that was just created. (S8) | Parcel does not exist in database and can not be seen within the application. | SellerBeanTest  .testDeleteParcel()  PASS  Figure 4 SellerBean test results |
| T29 | - Create a recipient user with username “Jimmy”.  - Create a seller user with username “Sally”.  - **Create order** with recipientId=(ID of “Jimmy”), sellerId=(ID of “Sally”).  - **Delete order** that was just created. (S9) | Order no longer exists in database and can not be seen within the application. | SellerBeanTest  .testDeleteOrder()  PASS  Figure 4 SellerBean test results |
| T30 | - Create a driver user with username “Bob”.  - Create a driver user with username “Dave”.  - Create a recipient user with username “Jimmy”.  - Create a seller user with username “Sally”.  - **Create order** “Order1” where “Jimmy” is the recipient and “Sally” is the seller.  - **Create order** “Order2” where “Jimmy” is the recipient and “Sally” is the seller.  - **Create order** “Order3” where “Jimmy” is the recipient and “Sally” is the seller.  - **Add transaction** “Picked up” to “Order1” while using the application as “Bob”.  - **Add transaction** “Dropped off” to “Order1” while using the application as “Bob”.  - **Add transaction** “Picked up” to “Order2” while using the application as “Bob”.  - **Add transaction** “Dropped off” to “Order2” while using the application as “Bob”.  - **Add transaction** “Picked up” to “Order3” while using the application as “Dave”.  - **Add transaction** “Dropped off” to “Order3” while using the application as “Dave”.  - Run **View driver metrics** while using the application as “Sally” (Seller). (S10) | Returns list of metrics showing “Bob” with 2 total deliveries and average days to complete 0 & “Dave” with 1 total delivery and average days to complete 0. | SellerBeanTest  .testViewDriverMetrics()  PASS  Figure 4 SellerBean test results |
| T31 | Seller views all orders. (S17) | Returns a list of 1 or more orders. | SellerBeanTest  .testViewAllOrders()  PASS  Figure 4 SellerBean test results |
| T32 | Seller views details of order with ID=1. (S16) | Returns details of the order with ID=1. | SellerBeanTest  .testFindOrderById()  PASS  Figure 4 SellerBean test results |
| T33 | Seller views all parcels. (S18) | Returns a list of 1 or more parcels. | SellerBeanTest  .testViewAllParcels()  PASS  Figure 4 SellerBean test results |
| T34 | Seller views details of parcel with ID=1. (S19) | Returns details of the parcel with ID = 1. | SellerBeanTest  .testFindParcelById()  PASS  Figure 4 SellerBean test results |

# Appendices

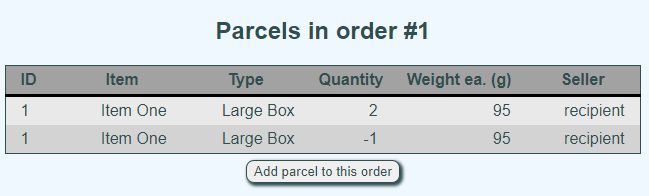


Figure Adding negative quantity parcel to order (FAIL)

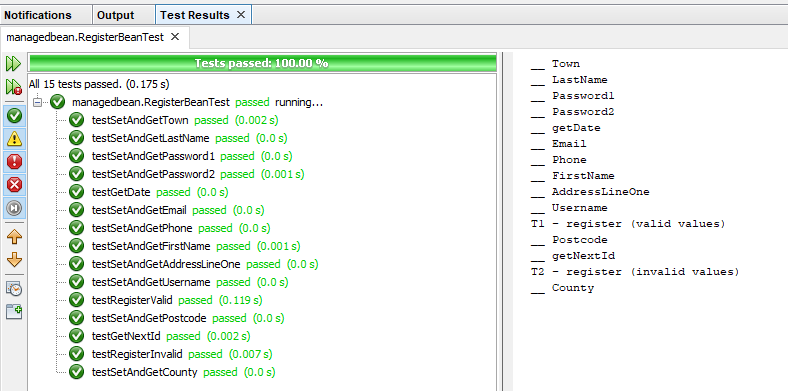


Figure RegisterBean Test Results

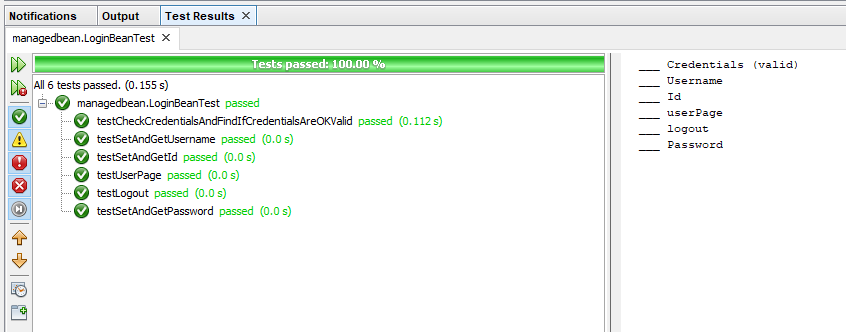


Figure LoginBean test results

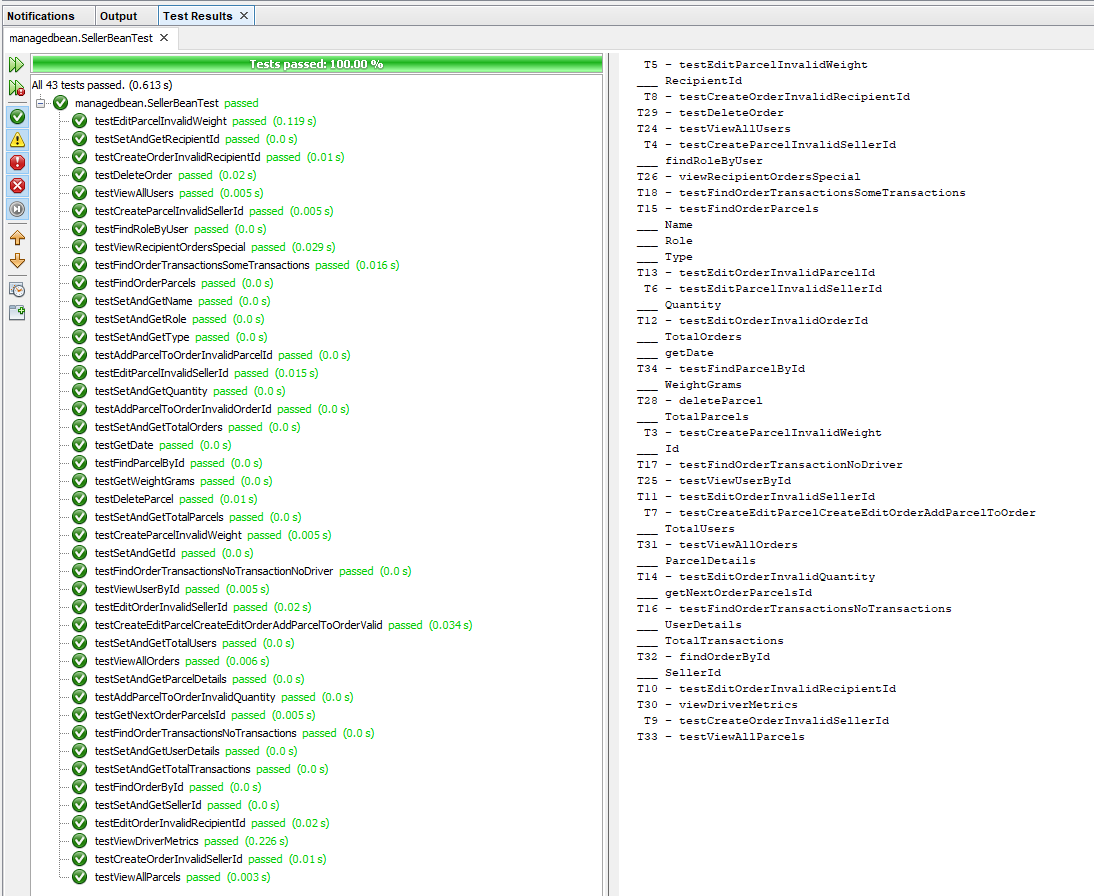


Figure SellerBean test results

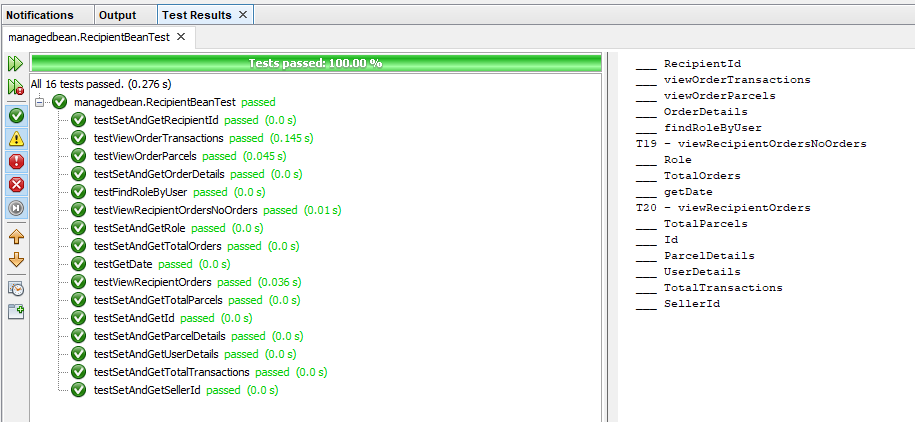


Figure RecipientBean test results

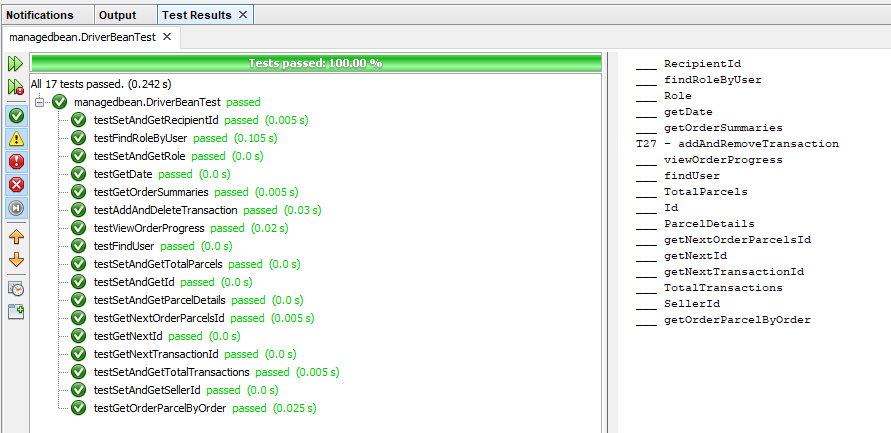


Figure 6 DriverBean test results

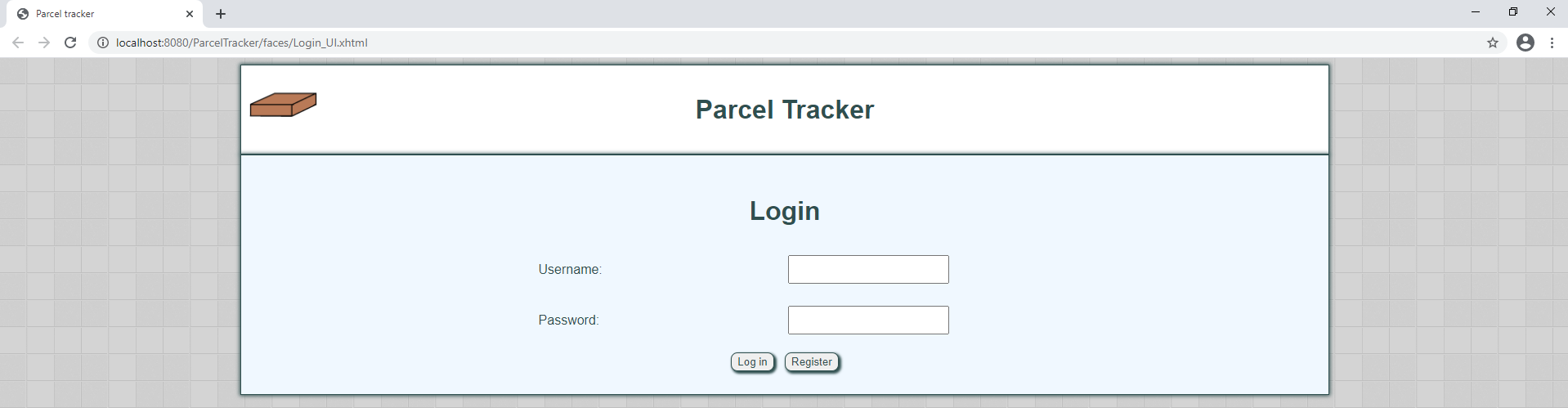


Figure Guest views Login

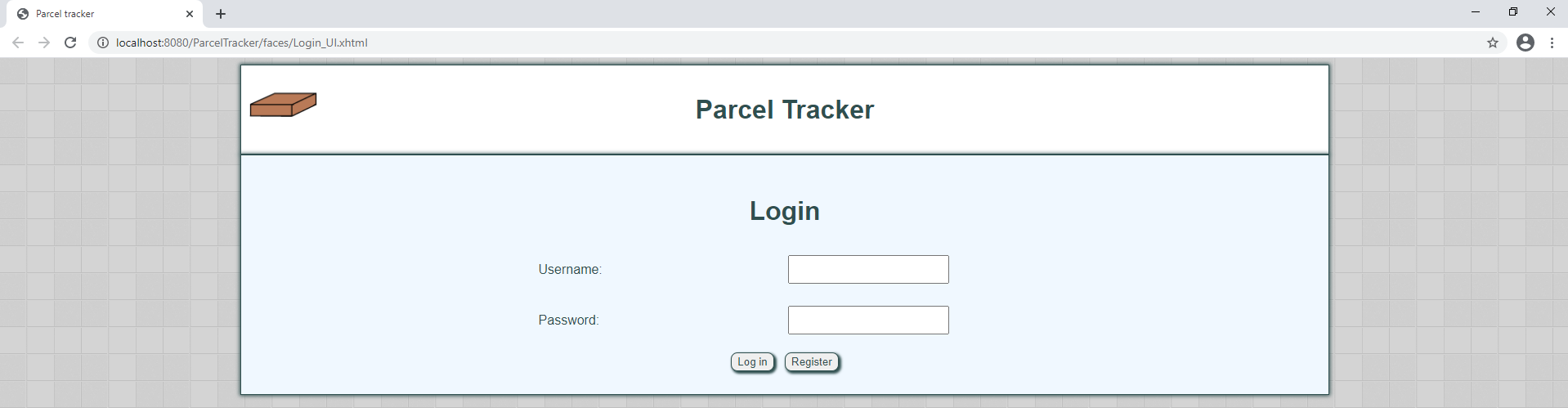


Figure Guest attempts to view restricted page but is returned to login screen

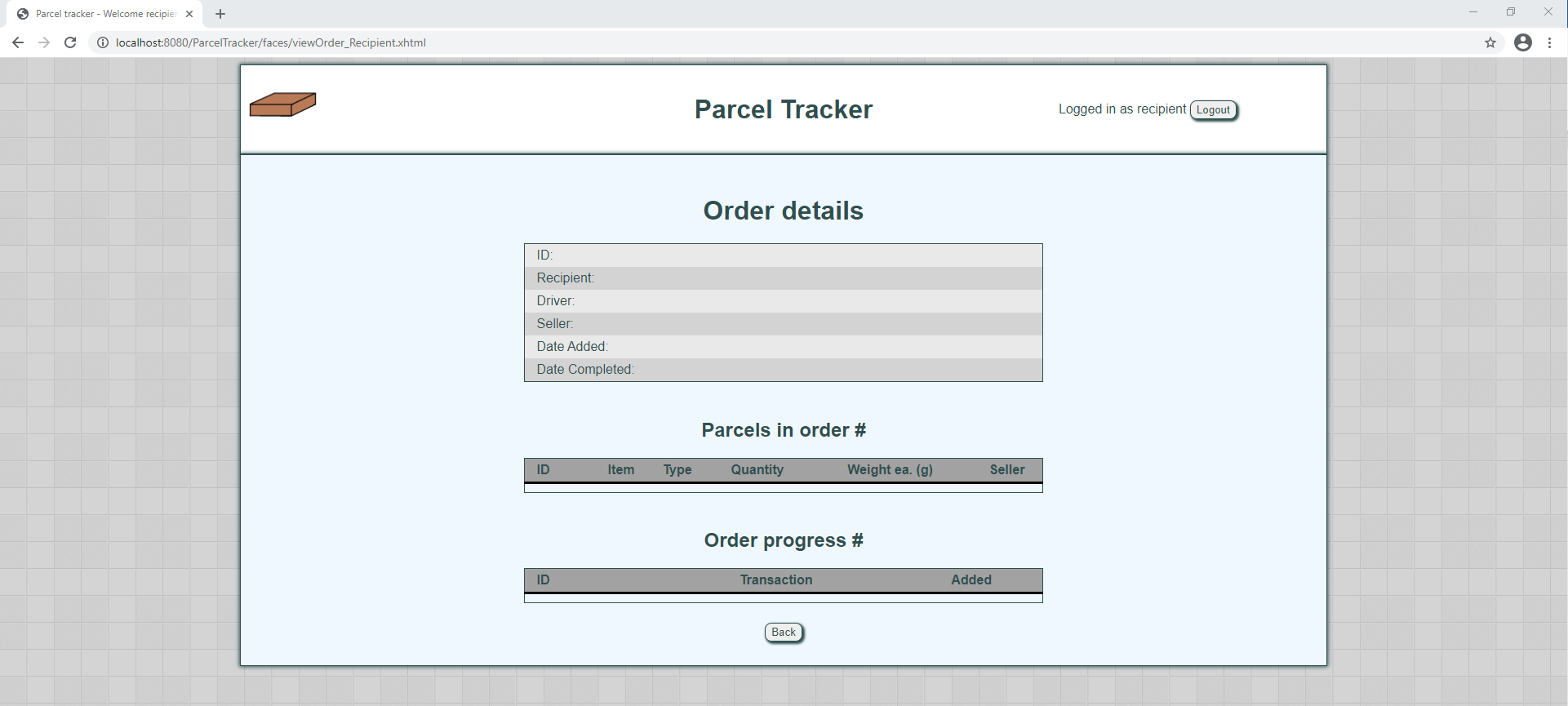


Figure Recipient views order details page while logged in without being redirected back to login screen

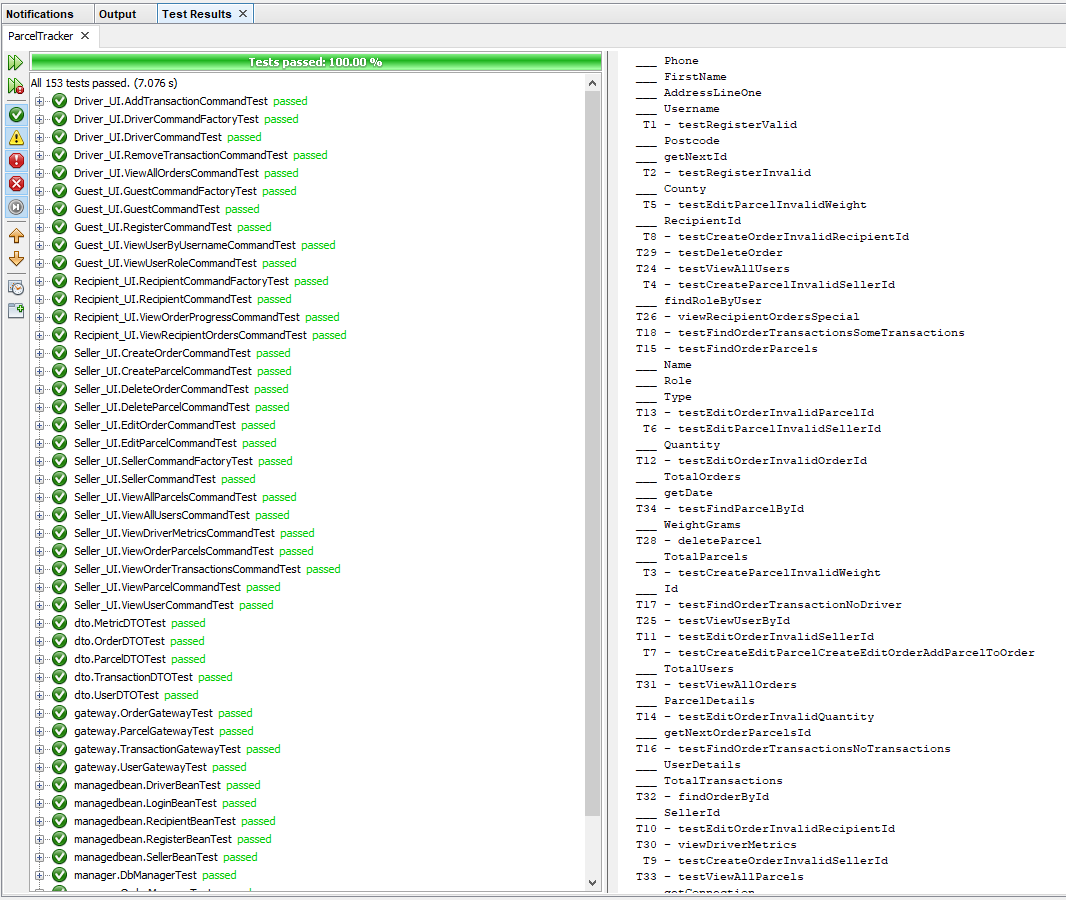


Figure All JUnit tests completed and passed