[**Display some snapshots of the result in the report.**](#_q8eqnfv5fecr) **1**

[Building blocks of this online shopping:](#_z4fvlscsehzq) 1

[Products page](#_z4fvlscsehzq) 1

[Product info. page](#_z4fvlscsehzq) 1

[Subscribe specific item page](#_z4fvlscsehzq) 2

[Cart page](#_z4fvlscsehzq) 2

[Payment page](#_z4fvlscsehzq) 3

[**You need to evaluate the design quality by using object-oriented quality metrics (WMC, DIT, NOC, CBO, RFC, LCOM). The figure shall be drawn like the provided references below.**](#_3m351kb3yzaj) **3**

[**Create Junit test cases and Junit test suite to test one selected class.**](#_fg6he5rq5q6) **8**

[**Conduct part of the software testing including white box and black box.**](#_l599ie5j9r47) **9**

[While Box](#_xxw6ox487ho5) 9

[Flow graph](#_pbjepldphb1l) 9

[Basis Paths](#_tuyg8xz23j5t) 11

[Black Box](#_xxw6ox487ho5) 11

[Cause-Effect Testing](#_ht5pqdcebh3) 11

[Boundary Value Analysis](#_as7j1o994708) 13

[Deriving Test Cases](#_svbp9r5op4gi) 14

[**Please analyze the invocation chains of your design.**](#_gcqeie2yrzsw) **15**

[public class addtocart extends HttpServlet](#_ujs35etn32wz) 15

[public class deletecartCommand implements command](#_8imj1xsmkum1) 16

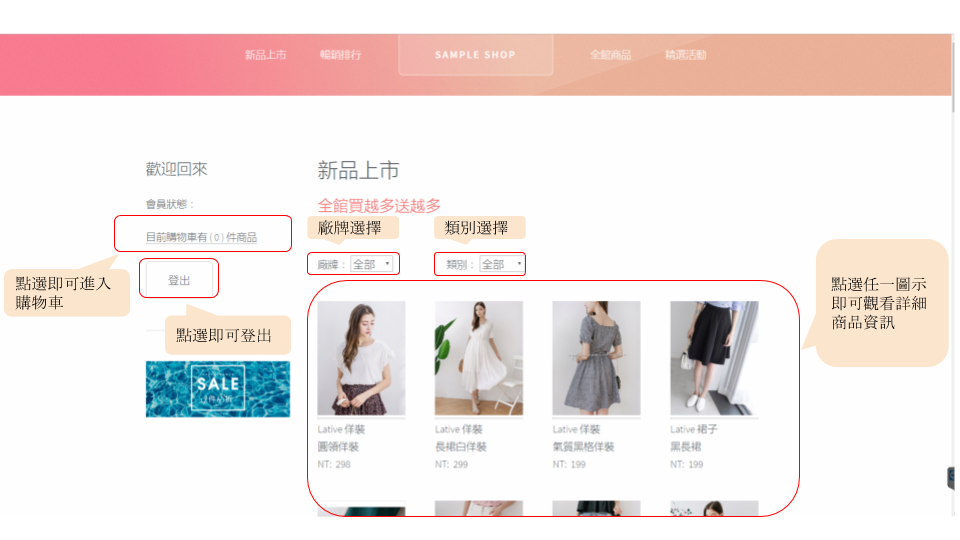
[Count table](#_y3bwtbus52ag) 16

[**Score Sheet**](#_xamjbpunqk9b) **17**

# Display some snapshots of the result in the report.

## Building blocks of this online shopping:

### Products page



### Product info. page



### Subscribe specific item page



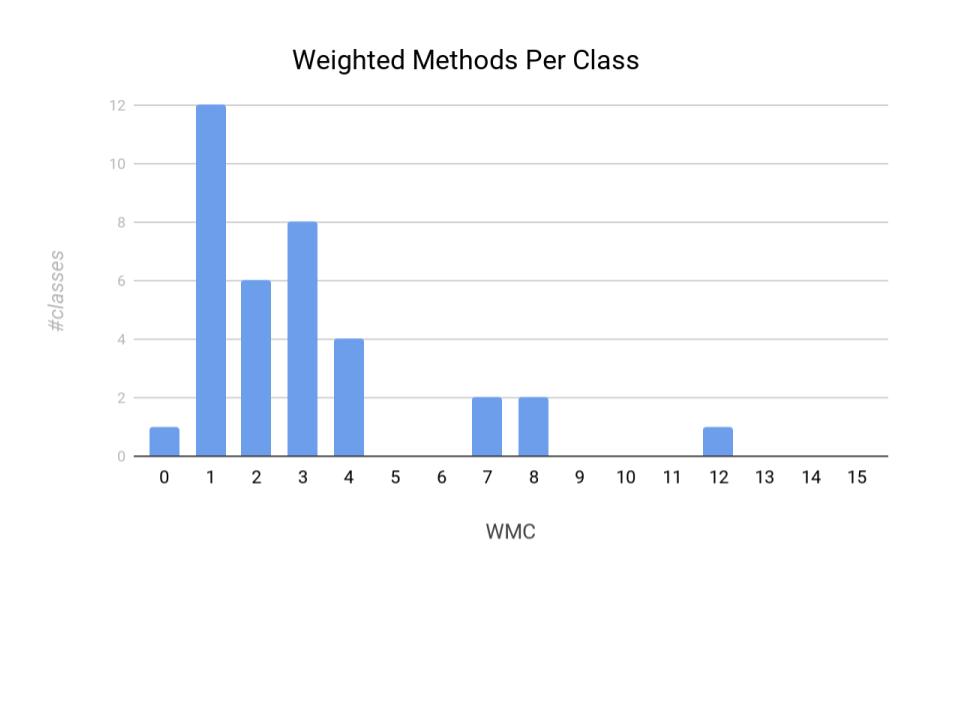
### Cart page

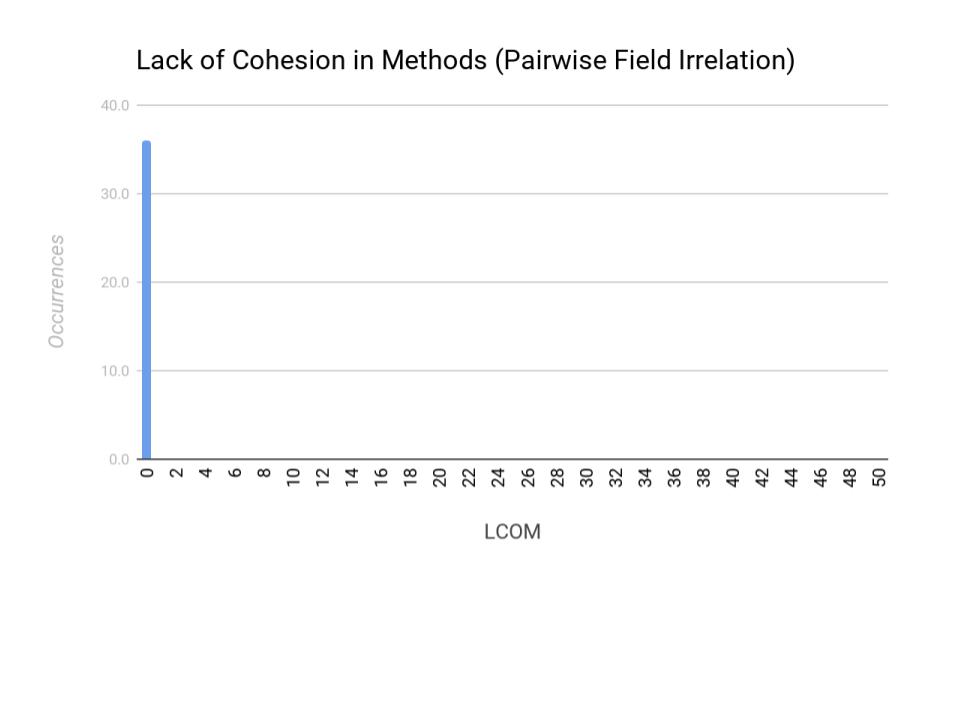
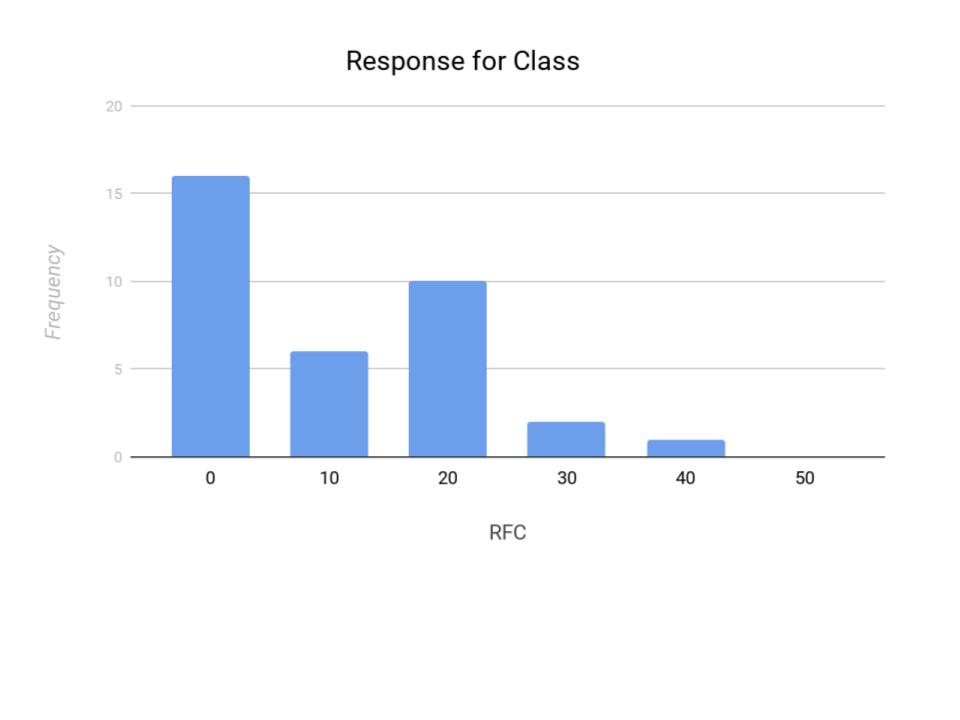
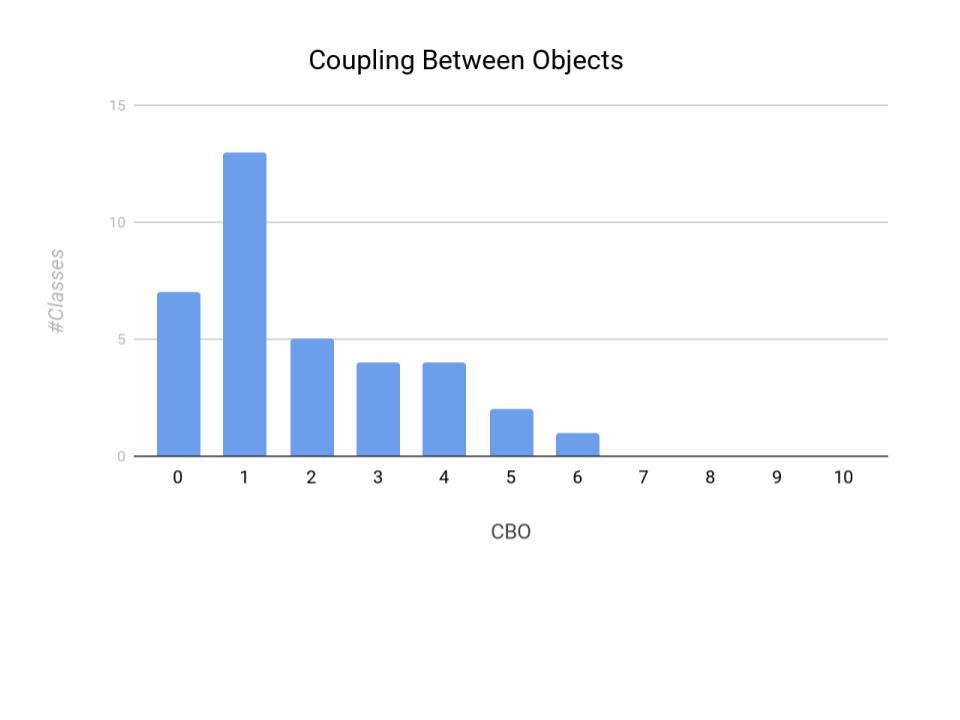
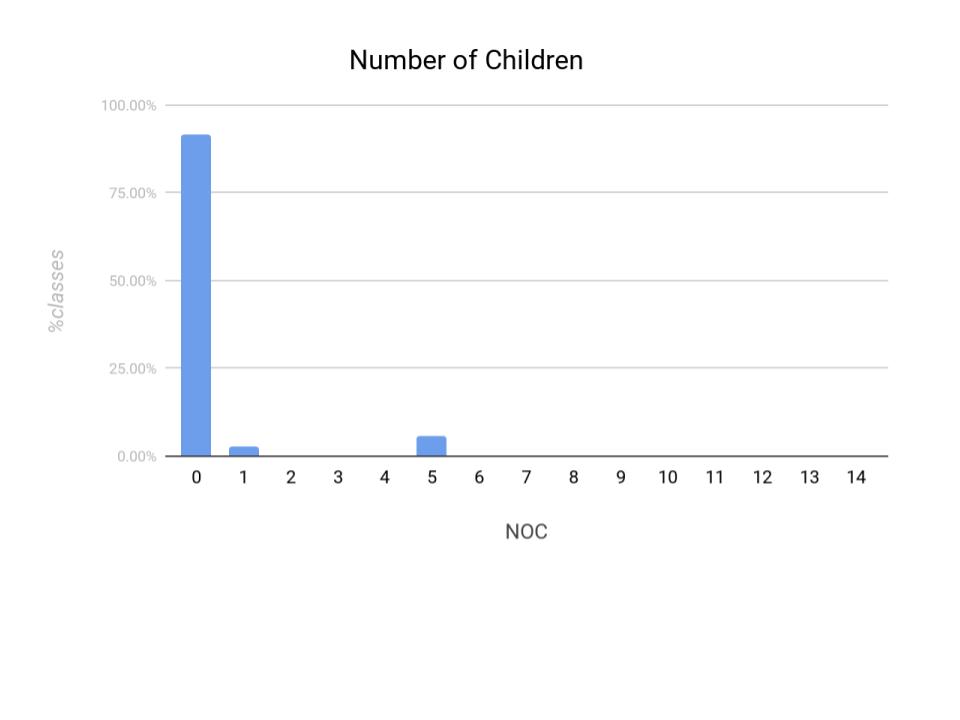
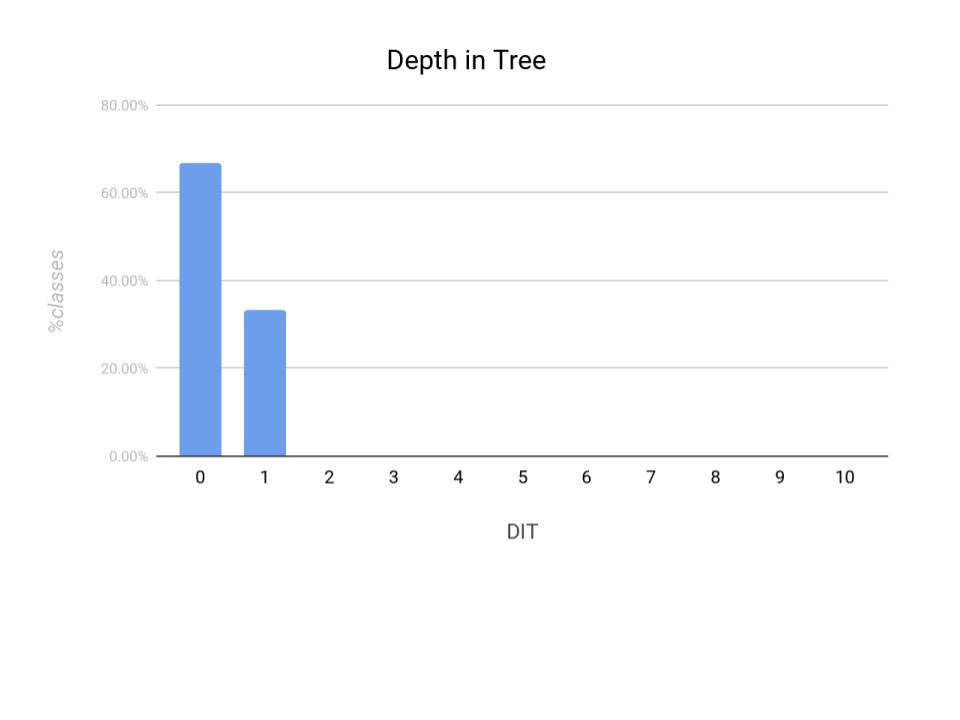


### Payment page



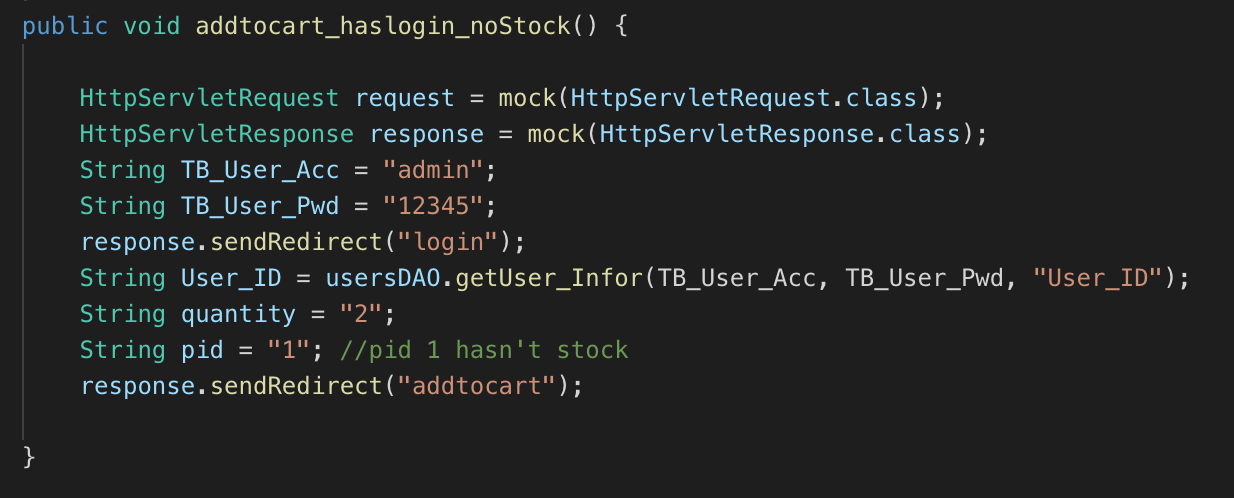
# You need to evaluate the design quality by using object-oriented quality metrics (WMC, DIT, NOC, CBO, RFC, LCOM). The figure shall be drawn like the provided references below.



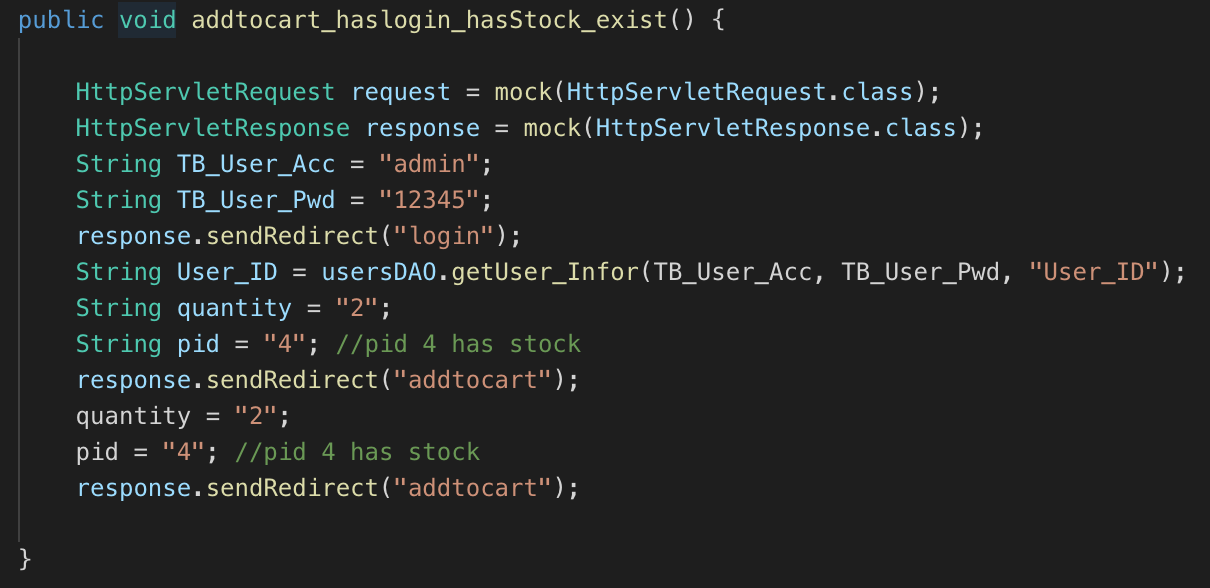


# Create Junit test cases and Junit test suite to test one selected class.

# 





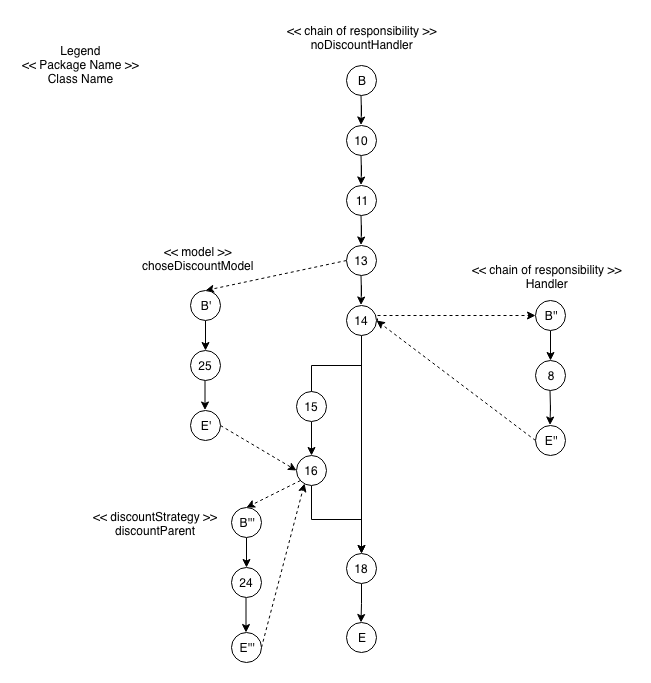


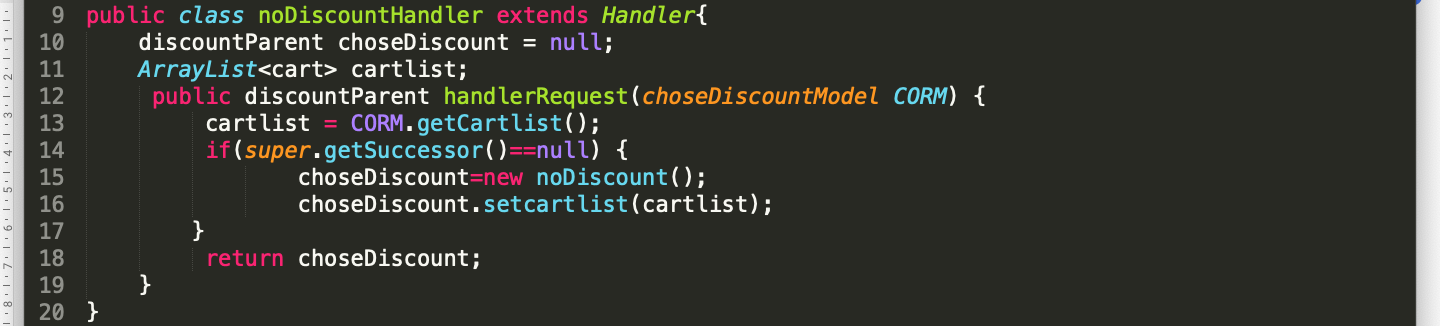
# 

# Conduct part of the software testing including white box and black box.

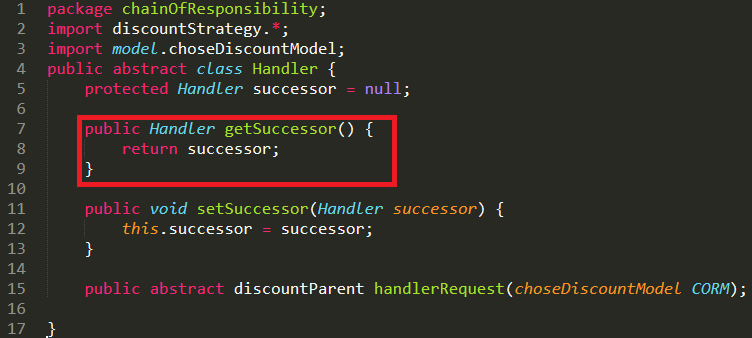
## While Box

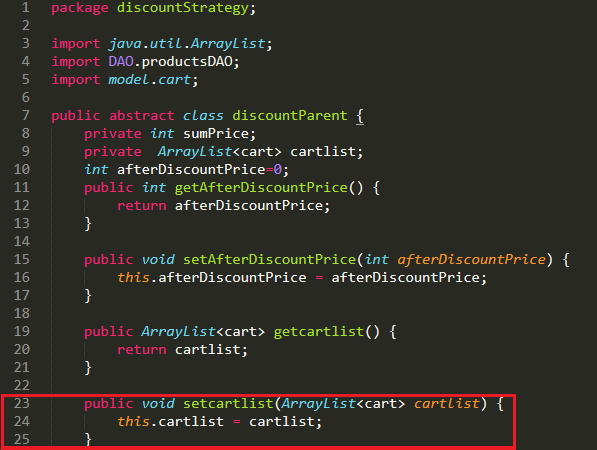
### Flow graph











### Basis Paths

Path1: B 10 11 13 14 15 16 18 E

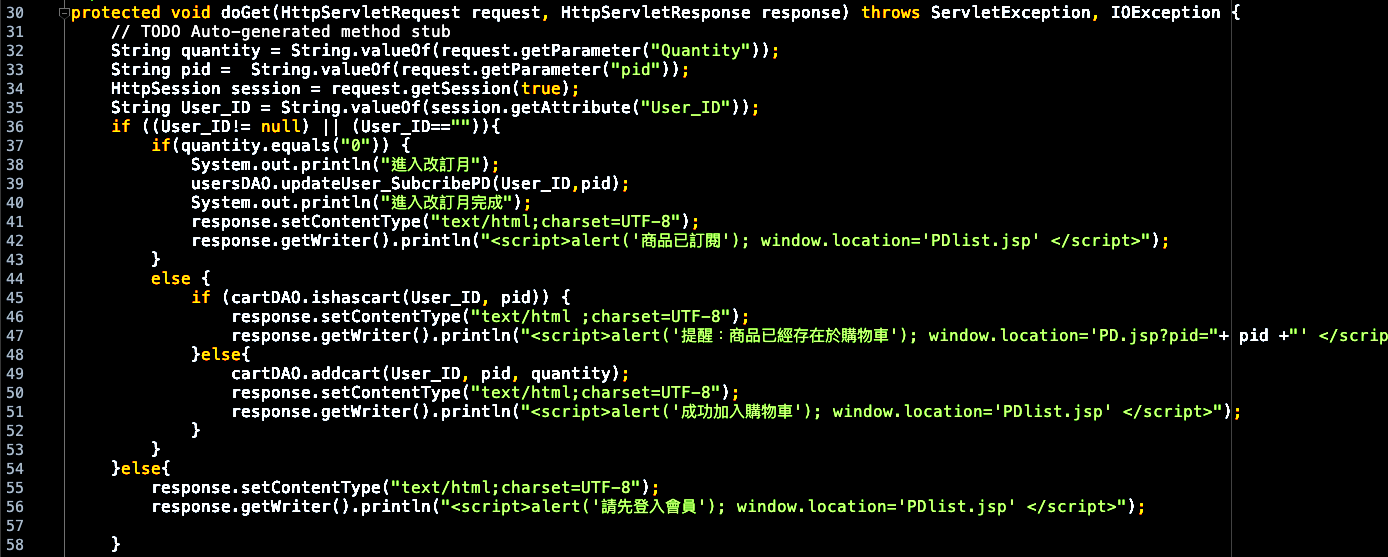
Path2: B 10 11 13 14 18 E

## Black Box

### Cause-Effect Testing

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| login | N | N | N | N | Y | Y | Y | Y |
| in of stock | N | N | Y | Y | N | N | Y | Y |
| item(s) was already in your Cart | N | Y | N | Y | N | Y | N | Y |
| case count | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| add successfully |  |  |  |  |  |  | X |  |
| add-fail | X | X | X | X | X | X |  | X |

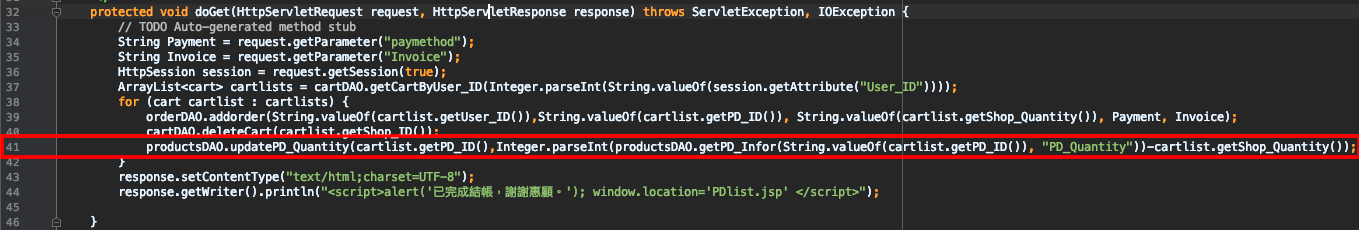
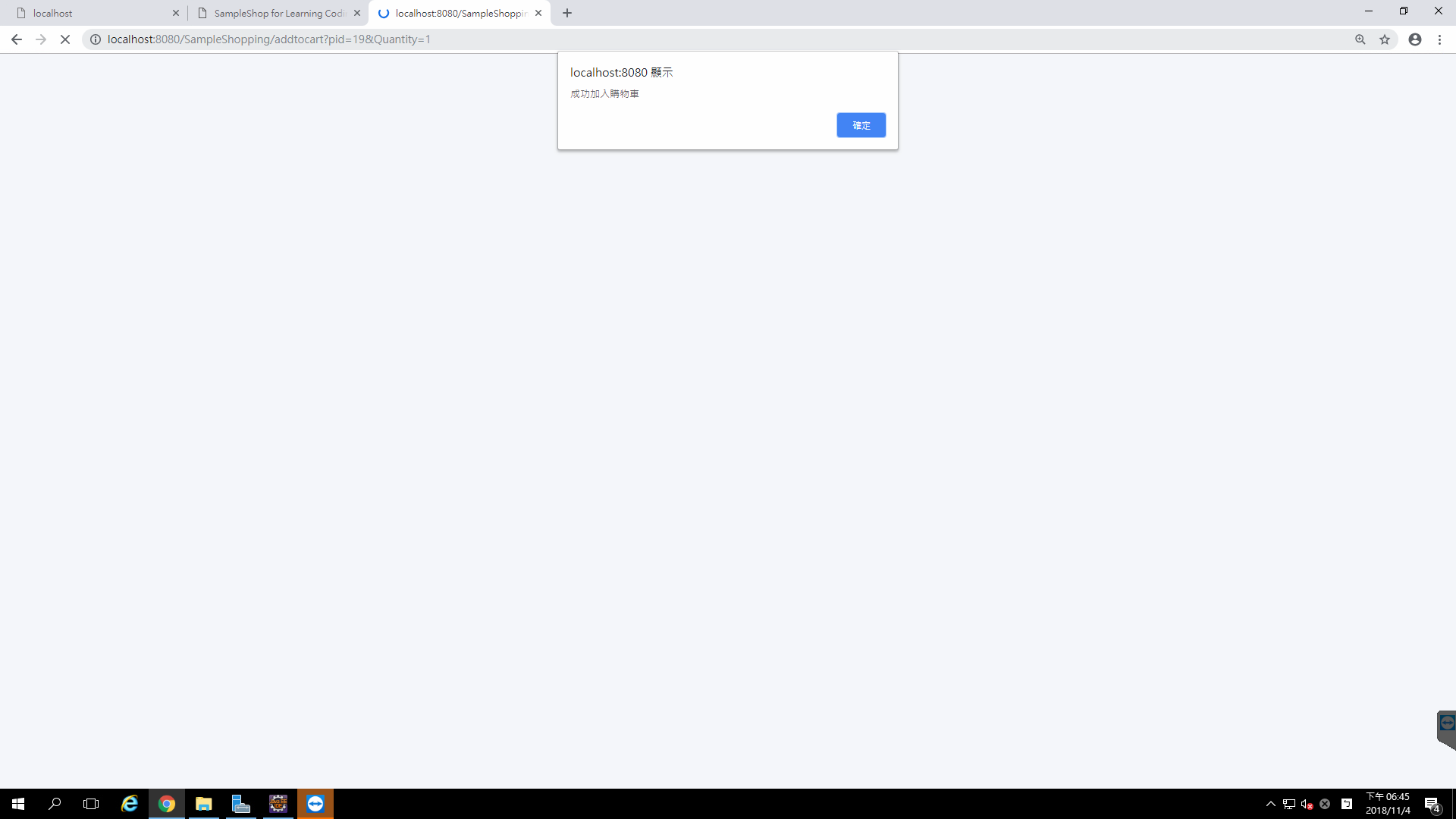
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| login | N | Y | Y | Y |
| in of stock | - | N | Y | Y |
| item(s) was already in your Cart | - | - | N | Y |
| case count | 4 | 2 | 1 | 1 |
| add successfully |  |  | X |  |
| add-fail | X | X |  | X |



### Boundary Value Analysis

m = # in stock before Customer’s choice

|  |  |  |
| --- | --- | --- |
| No. | # of item(s) in stock | output |
| 1 | -1 | Error |
| 2 | 0 | Message :=  ‘sold out’ |
| 3 | 1 | add successfully |
| 4 | m-1 | add successfully |
| 5 | m | add successfully |
| 6 | m+1 | Error |
| 7 | [0,m] | add successfully |



# 

### Deriving Test Cases

m = # in stock before Customer’s choice

n = # in stock after checkout completed

Stock = 1, 2, … n , n<=m

**case 1**: n<=0

Input : Stock = (1,2,3,...m)

expected output : Message := ‘sold out’

**case 2**: n=1 (After checkout completed, only one item in stock)

Input : Stock=(1,2,3,...m)

expected output: Stock= (1).

**case 3.1**: m>n (After checkout completed, the remainder is greater than one)

Input : Stock=(1, 2, 3,...m)

expected output: (1, 2,...n).

**case 3.2**: m=n (An order isn’t completed for some unknown reason)

Input : Stock=(1,2,3,...m)

expected output: Stock=(1,2,3,...n).

# Please analyze the invocation chains of your design.

## public class addtocart extends HttpServlet

|  |  |
| --- | --- |
| doGet(HttpServletRequest request, HttpServletResponse response)→ request.getParameter(**"Quantity"**)→  String.valueOf(request.getParameter(**"Quantity"**)) | 2 |
| doGet(HttpServletRequest request, HttpServletResponse response)→request.getParameter(**"pid"**)→  String.valueOf(request.getParameter(**"pid"**)) | 2 |
| doGet(HttpServletRequest request, HttpServletResponse response)→request.getSession(true) | 1 |
| doGet(HttpServletRequest request, HttpServletResponse response)→session.getAttribute(**"User\_ID")**)→  String.valueOf(session.getAttribute(**"User\_ID"**)) | 2 |
| doGet(HttpServletRequest request, HttpServletResponse response)→usersDAO.updateUser\_SubcribePD(User\_ID,pid) | 1 |
| doGet(HttpServletRequest request, HttpServletResponse response)→response.setContentType(**"text/html;charset=UTF-8"**) | 1 |
| doGet(HttpServletRequest request, HttpServletResponse response)→response.getWriter().println(**"<script>alert('商品已訂閱'); window.location='PDlist.jsp' </script>"**) | 1 |
| doGet(HttpServletRequest request, HttpServletResponse response)→cartDAO.ishascart(User\_ID, pid)→  response.**setC**ontentType(**"text/html ;charset=UTF-8"**)→response.getWriter().println(**"<script>alert('提醒：商品已經存在於購物車');window.location='PD.jsp?pid="+ pid +"' </script>"**) | 3 |
| doGet(HttpServletRequest request, HttpServletResponse response)→cartDAO.ishascart(User\_ID, pid)→cartDAO.addcart(User\_ID, pid, quantity)→  response.setContentType(**"text/html;charset=UTF-8")→response.getWriter().println("<script>alert('成功加入購物車'); window.location='PDlist.jsp' </script>"**) | 4 |
| doGet(HttpServletRequest request, HttpServletResponse response)→response.setContentType(**"text/html;charset=UTF-8")→response.getWriter().println("<script>alert('請先登入會員'); window.location='PDlist.jsp' </script>"**) | 2 |
| doPost(HttpServletRequest request, HttpServletResponse response)→doGet(request, response) | 1 |

## 

## public class deletecartCommand implements command

|  |  |
| --- | --- |
| execute()→cartDAO.getCartByShop\_ID(Shop\_ID)→DC.add(cartDAO.getCartByShop\_ID(Shop\_ID))→cartDAO.deleteCart(Shop\_ID) | 3 |

## Count table (public class addtocart extends HttpServlet, public class deletecartCommand implements command )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Invocation Chain Length | 1 | 2 | 3 | 4 |
| Number of Chain | 5 | 4 | 2 | 1 |

# Score Sheet

|  |  |  |
| --- | --- | --- |
| Name | Score | accountabilities |
| Jenny | 100% | slide,presenter, invocation chain |
| Jason | 100% | code implementation, presenter |
| Kendy | 100% | code implementation, presenter |
| Johnny | 100% | presenter, material collection |
| Howard | 100% | snapshots of the result, presenter |
| James | 85% | audit the project |
| Timothy | 100% | document, presenter, all tasks |
| Jerry | 100% | Black box, presenter |
| Xavier | 100% | White box, presenter |
| Alex | 100% | White box, presenter |

# 