Object-Oriented

Software Engineering

Instructor : Huang, Chuen-Min

**Teamwork ver.2**

Group3

|  |  |
| --- | --- |
| ID | Name |
| B10523001 | Carol |
| B10523002 | Ivan |
| B10523003 | Hank |
| B10523004 | Rick |
| B10523005 | Aliss |
| B10523006 | Peggy |
| B10523007 | Bess |
| B10423019 | Ivan |
| B10423051  B10423052  B10423054 | Ginobili  Linda  John |
| Date 2018/ 12 / 12 | |

Contents

[1. Before run our interface 2](#_Toc529714948)

[2. Some snapshots of our web 4](#_Toc529714949)

[3. Class Diagram 7](#_Toc529714950)

[4. Patterns and Code 8](#_Toc529714951)

[Memento 8](#_Toc529714952)

[Observer 10](#_Toc529714953)

[Singleton 12](#_Toc529714954)

[State 14](#_Toc529714955)

[5. Evaluate the design by using object-oriented quality metrics 19](#_Toc529714956)

[5. Junit test cases and Junit test suite 22](#_Toc529714957)

[6. Software testing including black box and white box 23](#_Toc529714958)

[Black Box Test 23](#_Toc529714959)

[White Box Test 26](#_Toc529714960)

[7. Invocation chain of our design 28](#_Toc529714961)

[8. Participate In Assignments 29](#_Toc529714962)

# Before run our interface

Because we use external Database-phpMyAdmin. First, you need to install this Database’s software.

|  |
| --- |
| 1. Go to its download page : https://www.phpmyadmin.net/ |
|  |
| 1. Install it, then open it. Create a database named one size, and import our one\_size.sql. Our account and password use default value, so you don’t need to change it.   Account : root  Password : “” |

Second, set Build Path. Decompress mysql-connector-java-8.0.13.zip, then set it.

|  |
| --- |
| 1. Right click project name→Build Path→Configure Build Path… |
|  |
| 1. Follow the step, then you can run our program. |
| 1.  2.  3. |

5.

4.

# Some snapshots of our web

* These are our old interface, but our interface can’t work very successful. So we just put same snapshots in document, and we make a new interface behind those.

|  |
| --- |
| 1. Index |
|  |
| 1. Login |
|  |
| 1. Register |
|  |
| 1. Product Page |
|  |
| 1. Cart Page |
|  |

# Patterns and Code

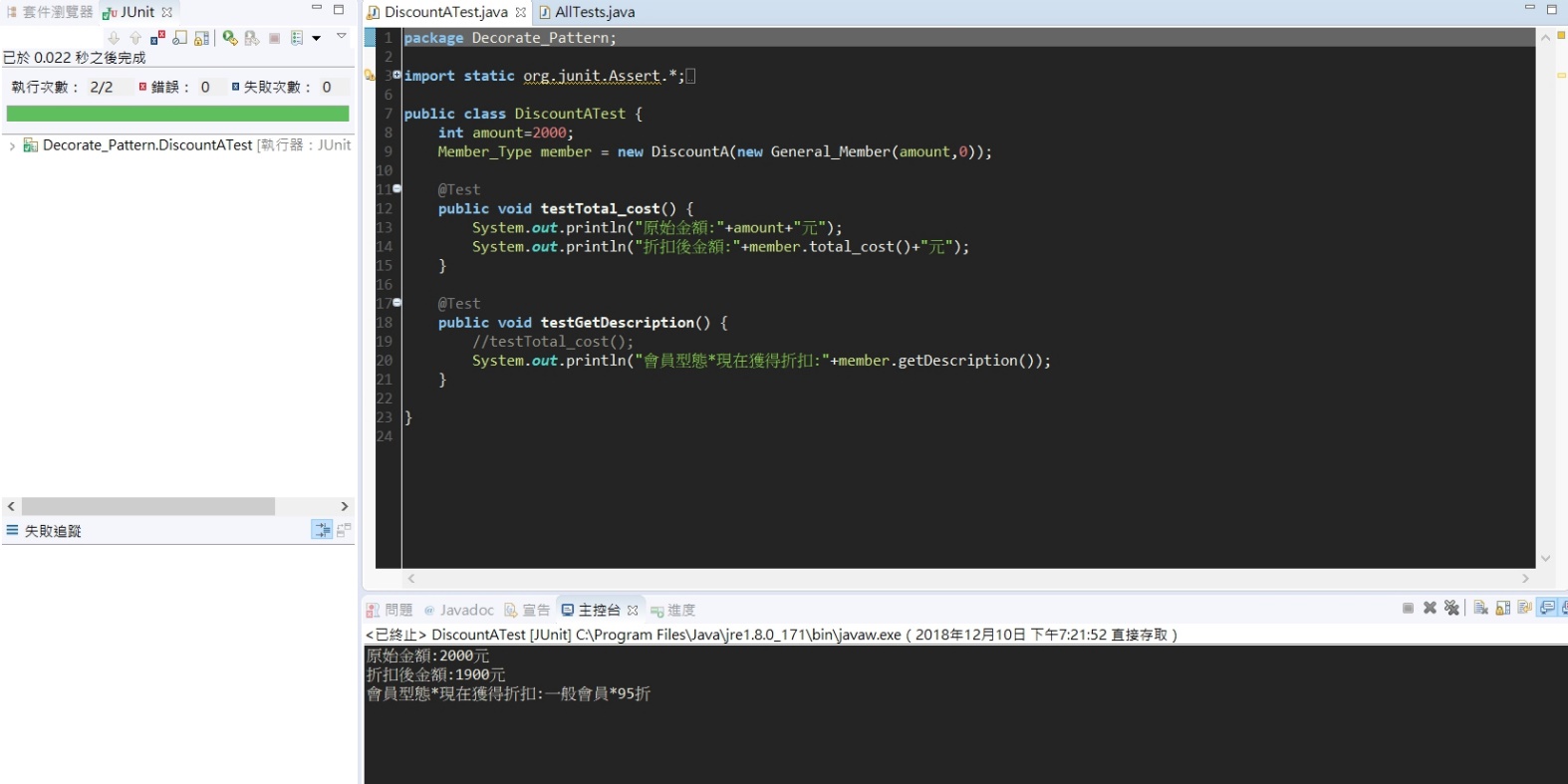
# Evaluate the design by using object-oriented quality metrics

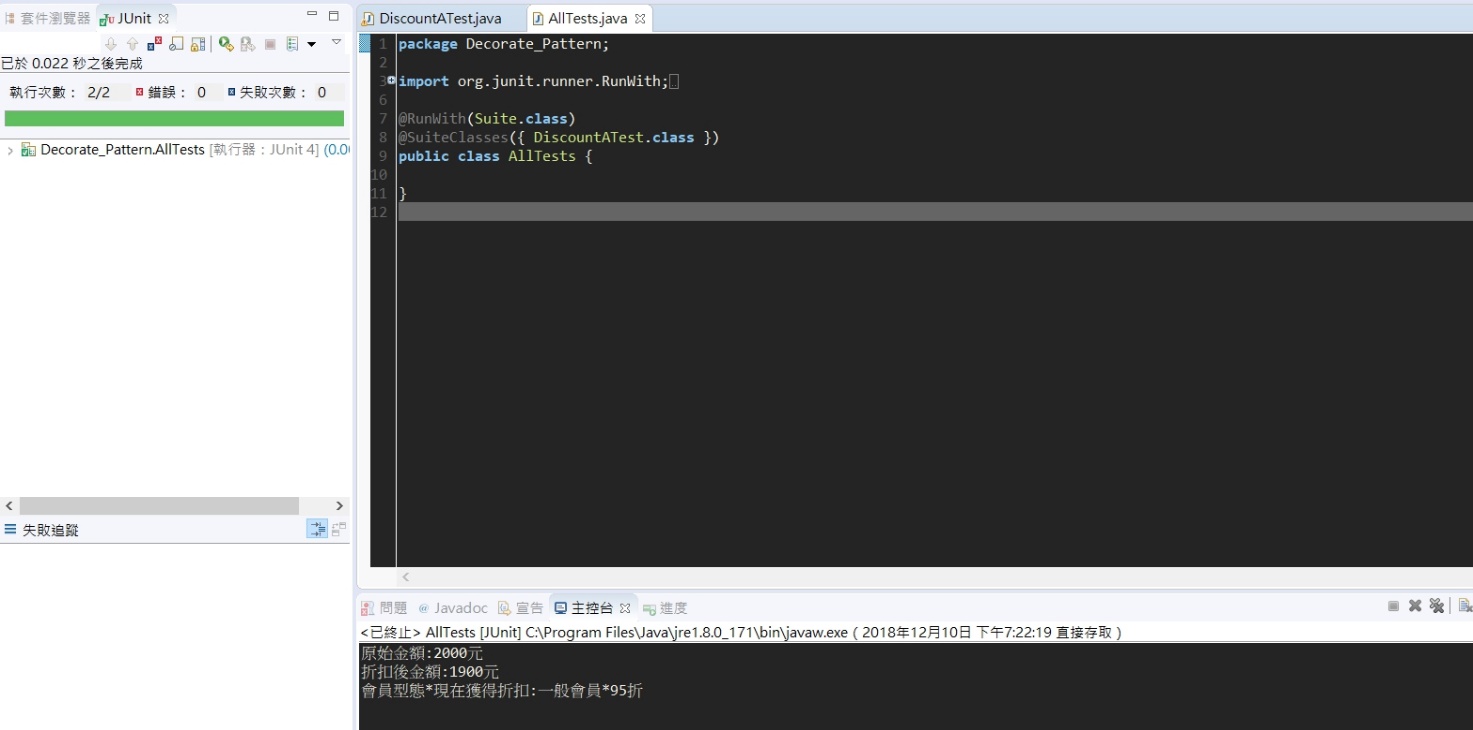
|  |
| --- |
| 1. WMC |
|  |
| 1. DIT |
|  |
| 1. NOC |
|  |
| 1. CBO |
|  |
| 1. RFC |
|  |
| 1. LCOM |
|  |

# **Junit test cases and Junit test suite**

* Junit test case

We use class DiscountATest as the test object, the test results are successful, it means it

doesn’t have errors.

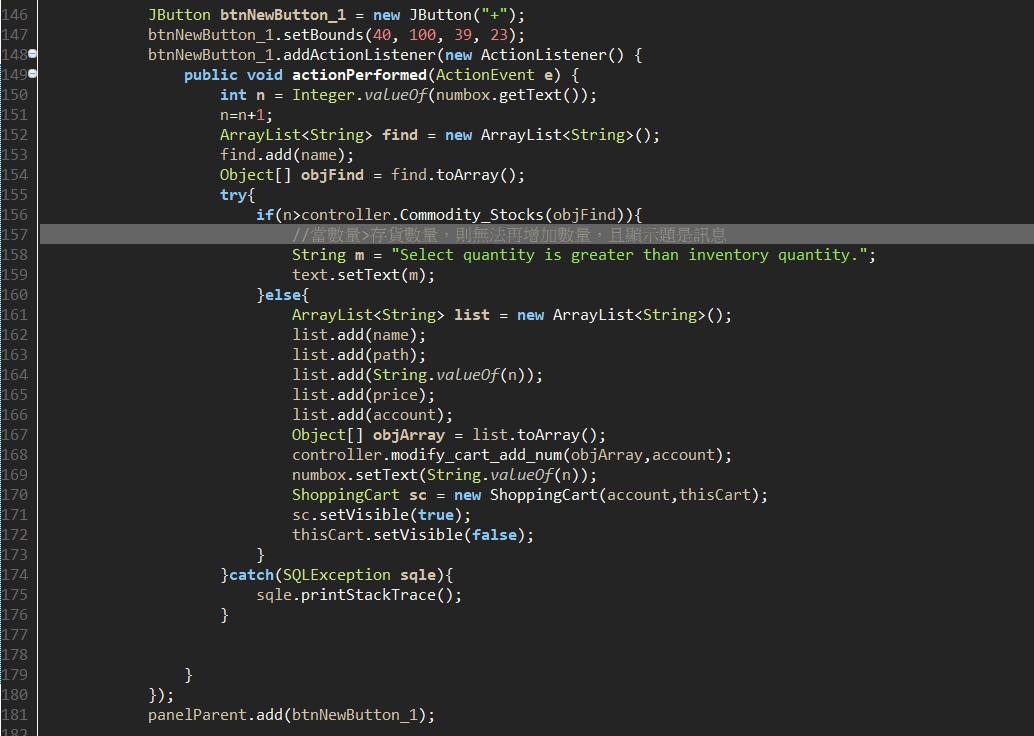
* Junit test suite

1. Software testing including black box and white box

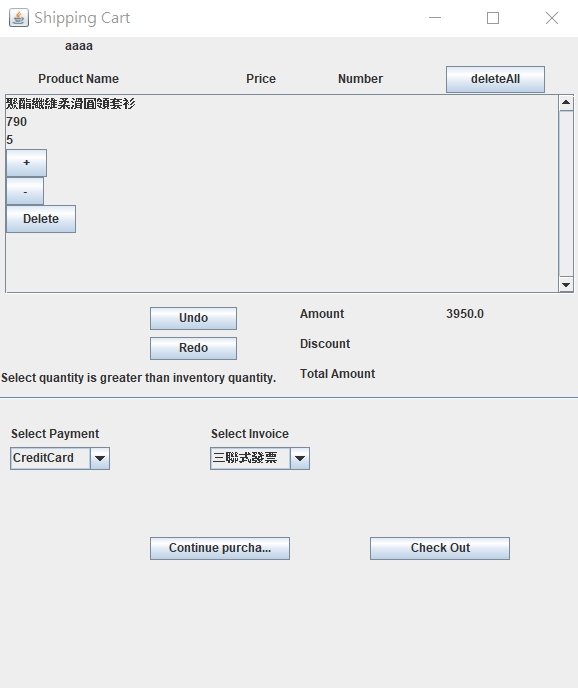
## Black Box Test

1. Boundary Value Analysis



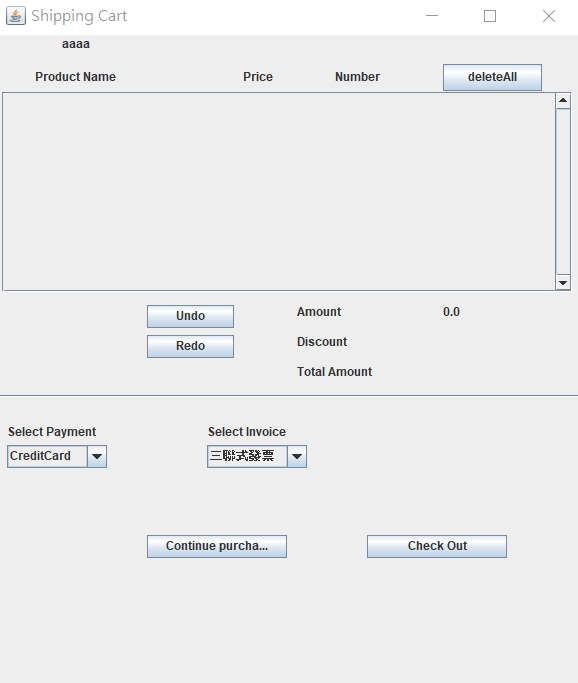


Suppose the inventory of clothes is five.



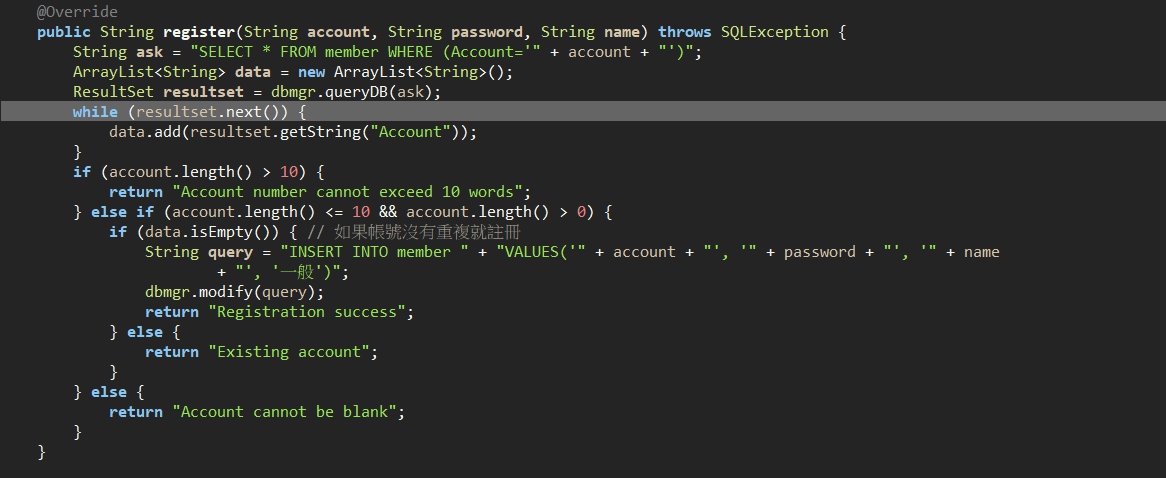
If the consumer orders more than five in the shopping cart.

The system will display "Select quantity is greater than inventory quantity".



If the consumer orders in the shopping cart is zero, the system doesn’t display product information

1. Cause-Effect Analysis



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Condition | Test Case | | | |
| Account.length()>10 | T | F | F | F |
| Account.length()<=10 | - | T | T | T |
| Account.length()>0 | - | T | F | T |
| Data.isEmpty() | - | F | - | T |
| count | 4 | 1 | 2 | 1 |
| result |  |  |  |  |

Account number cannot exceed 10 words

Existing account

Account cannot be null

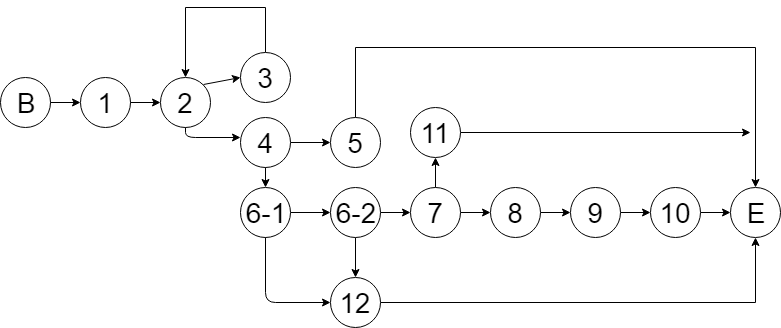
Registration success

## White Box Test

1. Basis Path Testing

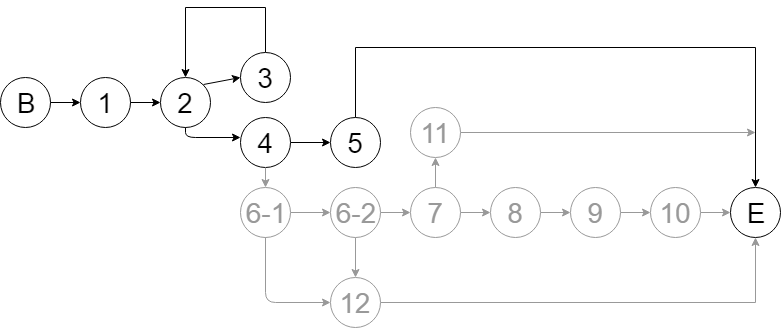
|  |  |
| --- | --- |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12** | **public String register**(String account, String password, String name) throws SQLException {  String ask = "SELECT \* FROM member WHERE (Account='" + account + "')";  ArrayList<String> data = new ArrayList<String>();  ResultSet resultset = dbmgr.queryDB(ask);  **while** (resultset.next()) { // 資料庫裡面有沒有下一筆資料  data.add(resultset.getString("Account")); //把資料庫的結果加到arraylist裡  }  **if** (account.length() > 10) {  return "Account number cannot exceed 10 words";  } **else if** (account.length() <= 10 **&&** account.length() > 0) { //Step 6-1 && Step 6-2  **if** (data.isEmpty()) { // 如果帳號沒有重複就註冊  String query = "INSERT INTO member " + "VALUES  ('" + account + "', '" + password + "', '" + name+ "', '一般')";  dbmgr.modify(query);  return "Registration success";  } **else** {  return "Existing account";  }  } **else** {  return "Account cannot be blank";  }  } |

Path**:**

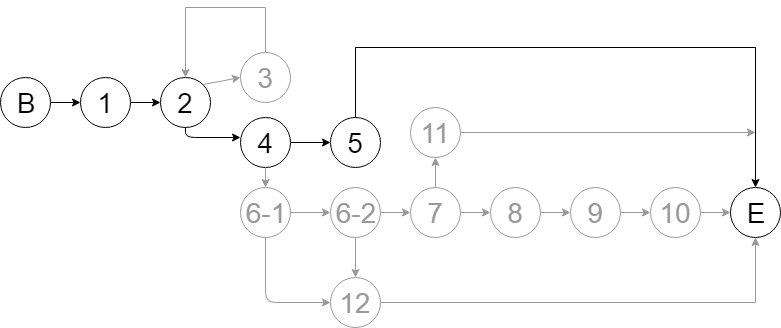
****

Cyclomatic Complexity : 5+1=6

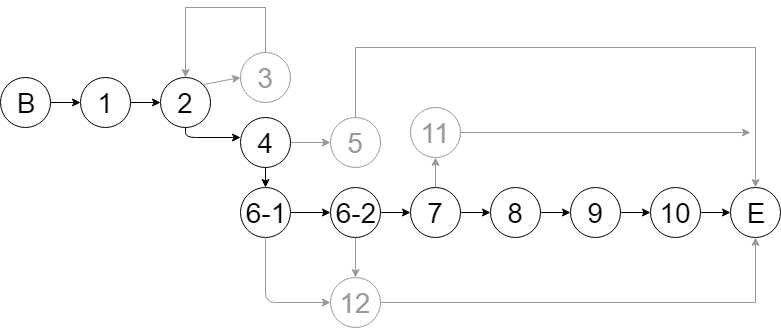
First Path: B🡪1🡪2🡪3🡪2🡪4🡪5🡪E



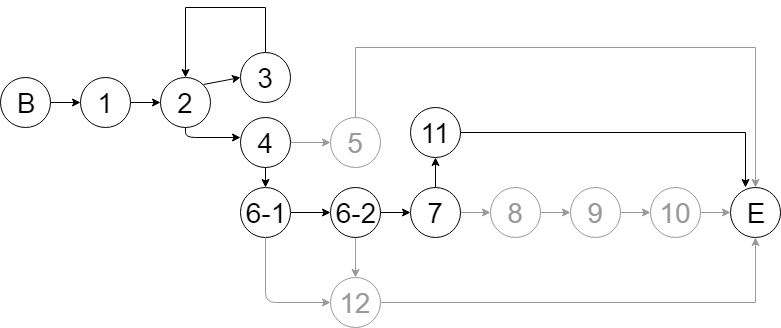
Second Path: B🡪1🡪2🡪4🡪5🡪E



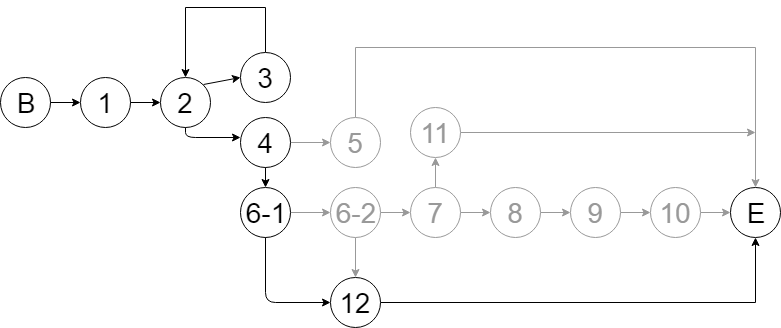
Third Path: B🡪1🡪2🡪4🡪6-1🡪6-2🡪7🡪8🡪9🡪10🡪E



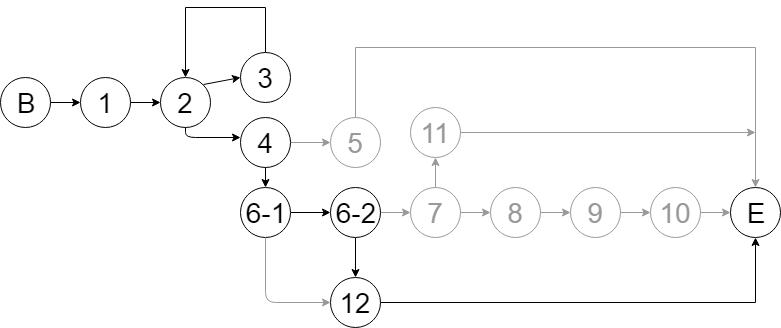
Fourth Path: B🡪1🡪2🡪3🡪2🡪4🡪6-1🡪6-2🡪7🡪11🡪E



Fifth Path: B🡪1🡪2🡪3🡪2🡪4🡪6-1🡪12🡪E



Sixth Path: B🡪1🡪2🡪3🡪2🡪4🡪6-1🡪6-2🡪12🡪E



# **Invocation chain of our design**

# Participate In Assignments

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Percentage (%) | Responsibility |
| B10523001 | Carol | 100% | 1. Design(Interface) 2. Document 3. PowerPoint 4. Participation 80% |
| B10523002 | Ivan | 65% | 1. Design(Interface) 2. Class Diagram 3. Participation 40% |
| B10523003 | Hank | 55% | 1. Design(Interface) 2. Document 3. Participation 20% |
| B10523004 | Rick | 100% | 1. Code(Interface) 2. Class Diagram 3. Participation 100% |
| B10523005 | Aliss | 100% | 1. Design(Interface) 2. Document 3. PowerPoint 4. Participation 100% |
| B10523006 | Peggy | 100% | 1. Design(Interface) 2. Document 3. PowerPoint 4. Participation 90% |
| B10523007 | Bess | 100% | 1. Code(Patterns) 2. Participation 100% |
| B10423019 | Ivan | 30% | 1. Participation 10% |
| B10423051 | Ginobili | 0% | 1. Nothing 2. Participation 0% |
| B10423052 | Linda | 100% | 1. Design(Interface) 2. Document 3. Class Diagram 4. Participation 100% |
| B10423054 | John | 0% | 1. Nothing 2. Participation 0% |