# Use Case Diagram Testing

Author: Bing Li

Contents:

[Use Case Diagram Testing 1](#_Toc385345950)

[Testing part: 1](#_Toc385345951)

[The testing input for the function of Xmi Parser: 2](#_Toc385345952)

[The testing output of the function of the Xmi Parser: 2](#_Toc385345953)

[Testing input for the function of Inserting and getting JSON String from database: 3](#_Toc385345954)

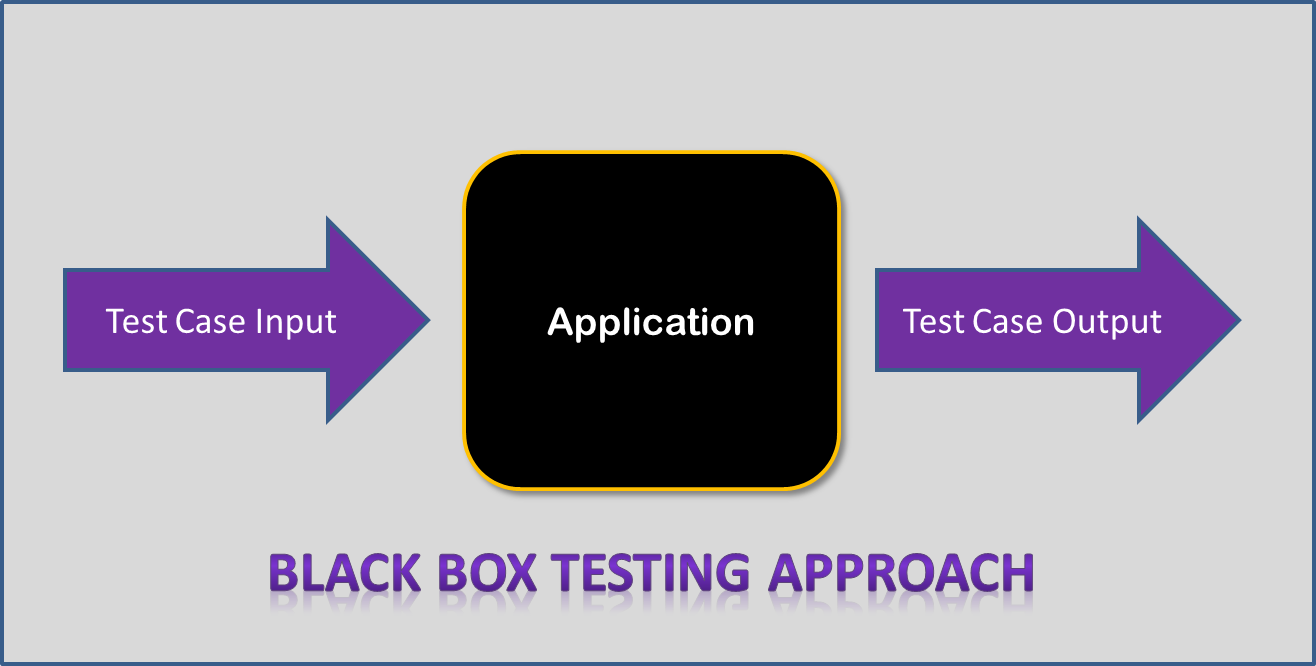
[Testing output for the function of Inserting and getting JSON String from database: 3](#_Toc385345955)

[Testing input for the function of JSON String parser: 3](#_Toc385345956)

[Testing output of the function of the JSON String parser: 4](#_Toc385345957)

[Conclusion: 4](#_Toc385345958)

## Testing part:

For the program of adding use case diagram, our team used black box technique to test the program. The conception of the black box is:

So we can see that the black box testing is a method of testing that examines the functionality of an application without peering into its internal structures. And specific knowledge of the application's code/programming knowledge is not required for the tester.

## The testing input for the function of Xmi Parser:

public void testXMIReader(String xmiFile) throws DocumentException

   {

          XMIParser xmi = XMLReader(xmiFile);

          for(Actor a:xmi.getActors())

          {

                 System.out.println(a.getName());

          }

          for(Association a:xmi.getAssociation())

          {

                 System.out.println(a.getActorName()+" "+a.getUseCaseName());

          }

          for(ExtendAssociation a:xmi.getExtendAssocs())

          {

                 System.out.println(a.getName());

          }

          for(IncludeAssociation a:xmi.getIncludeAssocs())

          {

                 System.out.println(a.getName());

          }

          for(UseCase a:xmi.getUsecases())

          {

                 System.out.println(a.getName());

          }

   }

   public static void main(String []args) throws DocumentException

   {

          XMIParser xmi = new XMIParser();

          xmi.testXMIReader("testnew.xml");

   }

## The testing output of the function of the Xmi Parser:

Actor1

Actor2

Actor3

Actor1 Assignment Project

Actor2 Assignment Project

null null

Actor2 Edit Document

null

null

Assignment Project

Edit Document

Make Comments

Check Permission

## Testing input for the function of Inserting and getting JSON String from database:

public void testInsertJSONString(String jsonString)

   {

          insertJSONString(jsonString);

          System.out.println(getJSONString());

   }

   public static void main(String args[])

   {

          PolicyDataEdit pde = new PolicyDataEdit();

          pde.testInsertJSONString("testjosnString");

   }

## Testing output for the function of Inserting and getting JSON String from database:

testjsonString

## Testing input for the function of JSON String parser:

public void testParser(String jsonString)

   {

          UserDefinedRule udr = parser(jsonString);

          System.out.println(udr.getMaximumOfActors());

          System.out.println(udr.getMustHaveActorName());

          System.out.println(udr.getMustHaveActors());

          System.out.println(udr.getMustHaveAssocs());

          System.out.println(udr.getMustHaveExtendAssociation());

          System.out.println(udr.getMustHaveIncludeAssociations());

   }

   public static void main(String []args)

   {

          JasonParser jp = new JasonParser();

          jp.testParser("testjsonString");

   }

## Testing output of the function of the JSON String parser:

Unexpected character (t) at position 0.

   at org.json.simple.parser.Yylex.yylex(Yylex.java:610)

   at org.json.simple.parser.JSONParser.nextToken(JSONParser.java:269)

   at org.json.simple.parser.JSONParser.parse(JSONParser.java:118)

   at org.json.simple.parser.JSONParser.parse(JSONParser.java:81)

   at org.json.simple.parser.JSONParser.parse(JSONParser.java:75)

   at controller.JasonParser.parser(JasonParser.java:37)

   at controller.JasonParser.testParser(JasonParser.java:317)

   at controller.JasonParser.main(JasonParser.java:329)

0

null

[]

[]

parser.ExtendAssociation@165e6c89

[]

Conclusion: when we did program testing, we did not peer into the internal structure or application’s code. We just input some testing code, and then we got the testing result of our program. So our testing technique is the mode of black box testing.