



EAST WEST UNIVERSITY

Course Code : CSE435
Section : 02

Test Case Design

For

Bank Management System

Group Members	
Name	Id
Adnan Saif Dipto	2018-1-60-157
Md. Mahmudur Rahman Limon	2018-1-60-253
Fatima Tanjum Tuba	2018-1-60-049

Submitted to

Toukir Ahammed

Lecturer

Department of Computer Science & Engineering

East West University

Test Case Design Specification

1. “Sign-Up” function:

- **Test Case Specification Identifier**
T1
- **Purpose**
To check the functionality of “Sign-Up” module
- **Special Environmental Requirements**
To have accessibility to Internet and to have server in running condition
- **Special Procedural Requirements**
No Needed.
- **Inter-case Dependencies**
No Needed
- **Input Specifications**
User Name: MDMR60253, Phone No: 01749157253, NID No: 1592824588424, Password: qWeR79@1
User Name: TUBA18049, Phone No: 8801718157600, NID no: 1592175598424, Password: mAqaBM4M#
User Name: ASD18157, Phone No: 8801711578897, NID no: 1592166788224, Password: gjeq57\$\$CM
- **Test Procedure**
To press “Sign-Up” button
- **Output Specifications**
Display message, “Registration Successful”.

Boundary Value Checking:

Since there are four variables, the total number of test cases will be $4n + 1 = 17$. In our example, the set of minimum and maximum values is shown below:

	User Name	Phone No	Password	NID No
Min Value	8	11	8	13
Min+ Value	9	12	9	14
Max Value	12	13	16	13
Max- Value	11	12	15	12
Nominal Value	10	12	12	13

Using these values, test cases can be designed as shown below:

Test Case Id	User Name	Phone No	Password	NID No	Expected Output
01	10	11	8	13	Successful
02	10	12	9	14	Unsuccessful
03	10	13	16	13	Successful
04	10	12	15	12	Unsuccessful
05	8	12	8	13	Unsuccessful
06	9	12	9	14	Unsuccessful
07	12	12	16	13	Unsuccessful
08	11	12	15	12	Unsuccessful
09	8	11	12	13	Successful
10	9	12	12	14	Unsuccessful
11	12	13	12	13	Successful
12	11	12	12	12	Unsuccessful
13	8	11	8	13	Successful
14	9	12	9	13	Unsuccessful
15	12	13	16	13	Successful
16	11	12	15	13	Unsuccessful
17	10	12	12	13	Unsuccessful

2. “Sign-In” function:

- **Test Case Specification Identifier**
T2
- **Purpose**
To check the functionality of “Sign-In” module
- **Special Environmental Requirements**
To have accessibility to Internet and to have server in running condition
- **Special Procedural Requirements**
Must be registered.
- **Inter-case Dependencies**
No Needed
- **Input Specifications**
User Name: MDMR60253, Password: qWeR79@1
User Name: TUBA18049, Password: mAqaBM4M#
User Name: ASD18157, Password: gjeq57\$\$CM
- **Test Procedure**
To press “Sign-In” button
- **Output Specifications**
Enter into Dashboard.

Boundary Value Checking:

Since there are two variables, the total number of test cases will be $4n + 1 = 9$. In our example, the set of minimum and maximum values is shown below:

	User Name	Password
Min Value	8	8
Min+ Value	9	9
Max Value	12	16
Max- Value	11	15
Nominal Value	10	12

Using these values, test cases can be designed as shown below:

Test Case Id	User Name	Password	Expected Output
01	10	8	Successful
02	10	9	Successful
03	10	16	Successful
04	10	15	Successful
05	8	12	Successful
06	9	12	Successful
07	12	12	Successful
08	11	12	Successful
09	10	12	Successful

3. “Create Employee ID” function:

- **Test Case Specification Identifier**
T3
- **Purpose**
To check the functionality of “Create Employee ID” module
- **Special Environmental Requirements**
To have accessibility to Internet and to have server in running condition
- **Special Procedural Requirements**
Must be registered and logged in to the system from the manager profile.
- **Inter-case Dependencies**
Must fulfill the “Sign-In” Criteria
- **Input Specifications**
User Name: MDJAD158, Phone No: 01750158253, NID No: 1592158582534, Password: qASH79@7, Password(Manager): gjeq57\$\$CM
User Name: NAM18018, Phone No: 8801677157600, NID no: 1577889998455, Password: NyMaBMX3#, Password(Manager): gjeq57\$\$CM
User Name: ANK10063, Phone No: 8801778855158, NID no: 1592133488889, Password: AinK67\$\$zN, Password(Manager): gjeq57\$\$CM
- **Test Procedure**
To press “Create Profile” button
- **Output Specifications**
Display message, “Profile Successfully Created”.

Boundary Value Checking:

Since there are four variables, the total number of test cases will be $4n + 1 = 21$. In our example, the set of minimum and maximum values is shown below:

	User Name	Phone No	Password	NID No	Password (Manager)
Min Value	8	11	8	13	8
Min+ Value	9	12	9	14	9
Max Value	12	13	16	13	16
Max- Value	11	12	15	12	15
Nominal Value	10	12	12	13	12

Using these values, test cases can be designed as shown below:

Test Case Id	User Name	Phone No	Password	NID No	Password (Manager)	Expected Output
01	10	11	8	13	8	Successful
02	10	12	9	14	9	Unsuccessful
03	10	13	16	13	16	Successful
04	10	12	15	12	15	Unsuccessful
05	8	12	8	13	8	Unsuccessful
06	9	12	9	14	9	Unsuccessful
07	12	12	16	13	16	Unsuccessful
08	11	12	15	12	15	Unsuccessful
09	8	11	12	13	8	Successful
10	9	12	12	14	9	Unsuccessful
11	12	13	12	13	16	Successful
12	11	12	12	12	15	Unsuccessful
13	8	11	8	13	8	Successful
14	9	12	9	13	9	Unsuccessful
15	12	13	16	13	16	Successful
16	11	12	15	13	15	Unsuccessful
17	8	11	8	13	12	Successful
18	9	12	9	14	12	Unsuccessful
19	12	13	16	13	12	Successful
20	11	12	15	12	12	Unsuccessful
21	10	12	12	13	12	Unsuccessful

4. “Remove Employee ID” function:

- **Test Case Specification Identifier**
T4
- **Purpose**
To check the functionality of “Remove Employee ID” module
- **Special Environmental Requirements**
To have accessibility to Internet and to have server in running condition
- **Special Procedural Requirements**
- Must complete ‘Sign In’ function
- **Inter-case Dependencies**
No Needed
- **Input Specifications**
User Name: MDMR60253, Password: qWeR79@1
User Name: TUBA18049, Password: mAqaBM4M#
User Name: ASD18157, Password: gjeq57\$\$CM
- **Test Procedure**
To press “Remove Employee ID” button
- **Output Specifications**
Display message, “ID Removed Successfully”.

Boundary Value Checking:

Since there are two variables, the total number of test cases will be $4n + 1 = 9$. In our example, the set of minimum and maximum values is shown below:

	User Name	Password
Min Value	8	8
Min+ Value	9	9
Max Value	12	16
Max- Value	11	15
Nominal Value	10	12

Using these values, test cases can be designed as shown below:

Test Case Id	User Name	Password	Expected Output
01	10	8	Successful
02	10	9	Successful
03	10	16	Successful
04	10	15	Successful
05	8	12	Successful
06	9	12	Successful
07	12	12	Successful
08	11	12	Successful
09	10	12	Successful

5. “Update Profile” function:

- **Test Case Specification Identifier**
T5
- **Purpose**
To check the functionality of “Update Profile” module
- **Special Environmental Requirements**
To have accessibility to Internet and to have server in running condition
- **Special Procedural Requirements**
- Must complete ‘Sign In’ function
- **Inter-case Dependencies**
No Needed
- **Input Specifications**
Phone No: 01749157253, Password: qWeR79@1
Phone No: 8801718157600, Password: mAqBM4M#
Phone No: 8801711578897, Password: gjeq57\$\$CM
- **Test Procedure**
To press “Remove Employee ID” button
- **Output Specifications**
Display message, “ID Removed Successfully”.

Boundary Value Checking:

Since there are two variables, the total number of test cases will be $4n + 1 = 9$. In our example, the set of minimum and maximum values is shown below:

	Phone No	Password
Min Value	11	8
Min+ Value	12	9
Max Value	13	16
Max- Value	12	15
Nominal Value	12	12

Using these values, test cases can be designed as shown below:

Test Case Id	Phone No	Password	Expected Output
01	12	8	Unsuccessful
02	12	9	Unsuccessful
03	12	16	Unsuccessful
04	12	15	Unsuccessful
05	11	12	Successful
06	12	12	Unsuccessful
07	13	12	Successful
08	12	12	Unsuccessful
09	12	12	Unsuccessful

6. “Create Customer ID” function:

- **Test Case Specification Identifier**
T6
- **Purpose**
To check the functionality of “Create Customer ID” module
- **Special Environmental Requirements**
To have accessibility to the Internet and to have servers in running condition.
- **Special Procedural Requirements**
Must complete ‘Sign In’ function
- **Intercase Dependencies**
No Needed
- **Input Specifications**
User Name: MDMR60253, Phone No: 01749157253, NID No: 1592824588424, Account Number: 8465284901264
User Name: TUBA18049, Phone No: 8801718157600, NID No: 1592175598424, Account Number: 1238763900036
User Name: ASD18157, Phone No: 8801711578897, NID No: 1592166788224, Account Number: 2645890012631
- **Test Procedure**
To press “Create Customer ID” button
- **Output Specifications**
Display message, “ID Created Successfully”.

Boundary Value Checking:

Since there are four variables, the total number of test cases will be $4n + 1 = 17$. In our example, the set of minimum and maximum values is shown below:

	User Name	Phone No	NID No	Account Num
Min Value	8	11	13	13
Min+ Value	9	12	14	14
Max Value	12	13	13	13
Max- Value	11	12	12	12
Nominal Value	10	12	13	13

Using these values, test cases can be designed as shown below:

Test Case Id	User Name	Phone No	NID No	Account Num	Expected Output
01	10	11	13	13	ID Created
02	10	12	14	14	Error
03	10	13	13	13	ID Created
04	10	12	12	12	Error
05	8	12	13	13	Error
06	9	12	14	14	Error
07	12	12	13	13	Error
08	11	12	12	12	Error
09	8	11	13	13	ID Created
10	9	12	13	14	Error
11	12	13	13	13	ID Created
12	11	12	13	12	Error
13	8	11	13	13	ID Created
14	9	12	14	13	Error
15	12	13	13	13	ID Created
16	11	12	12	13	Error
17	10	12	13	13	Error

7. “Remove Customer ID” function:

- **Test Case Specification Identifier**
T7
- **Purpose**
To check the functionality of “Remove Customer ID” module
- **Special Environmental Requirements**
To have accessibility to the Internet and to have servers in running condition.
- **Special Procedural Requirements**
Must complete ‘Sign In’ function
- **Inter-case Dependencies**
No Needed
- **Input Specifications**
Account Number: 8465284901264, Password: qWeR79@1
Account Number: 1238763900036, Password: mAqaBM4M#
Account Number: 2645890012631, Password: gjeq57\$\$CM
- **Test Procedure**
To press “Removed Customer ID” button
- **Output Specifications**
Display message, “ID Removed Successfully”.

Boundary Value Checking:

Since there are two variables, the total number of test cases will be $4n + 1 = 9$. In our example, the set of minimum and maximum values is shown below:

	Account Num	Password
Min Value	13	8
Min+ Value	14	9
Max Value	13	16
Max- Value	12	15
Nominal Value	13	12

Using these values, test cases can be designed as shown below:

Test Case Id	Account Num	Password	Expected Output
01	13	8	ID Removed
02	13	9	ID Removed
03	13	16	ID Removed
04	13	15	ID Removed
05	13	12	ID Removed
06	14	12	Error
07	13	12	ID Removed
08	12	12	Error
09	13	12	ID Removed

8. “Deposit” function:

- **Test Case Specification Identifier**
T8
- **Purpose**
To check the functionality of “Deposit” module
- **Special Environmental Requirements**
To have accessibility to Internet and to have server in running condition
- **Special Procedural Requirements**
Must complete ‘Sign In’ function
- **Inter-case Dependencies**
No Needed
- **Input Specifications**
Account No.: 324789655896, Credit value: 200000
Account No.: 564789655887, Credit value: 500000
- **Test Procedure**
To press “Deposit” button
- **Output Specifications**
Display message, “Deposit Successful”.

Boundary Value Checking:

Since there are two variables, the total number of test cases will be $4n + 1 = 9$. In our example, the set of minimum and maximum values is shown below:

	Account No.	Credit value
Min Value	12	1000
Min+ Value	13	1001
Max Value	12	5000000
Max- Value	11	4999999
Nominal Value	12	2500500

Using these values, test cases can be designed as shown below:

Test Case Id	Account No.	Credit value	Expected Output
01	12	1000	Successful
02	12	1001	Successful
03	12	5000000	Successful
04	12	4999999	Successful
05	12	2500500	Successful
06	13	2500500	Unsuccessful
07	12	2500500	Successful
08	11	2500500	Unsuccessful
09	12	2500500	Successful

9. “Transfer” function:

- **Test Case Specification Identifier**
T9
- **Purpose**
To check the functionality of “Transfer” module
- **Special Environmental Requirements**
To have accessibility to Internet and to have server in running condition
- **Special Procedural Requirements**
Must complete ‘Sign In’ function
- **Inter-case Dependencies**
No Needed
- **Input Specifications**
Account No (sender).: 154789655896, Transfer credit: 20000, Account No. (receiver): 669987196558
Account No. (sender): 544789655887, Transfer credit: 50000, Account No.(receiver): 478965589644
- **Test Procedure**
To press “Transfer” button
- **Output Specifications**
Display message, “Transfer Successful”.

Boundary Value Checking:

Since there are two variables, the total number of test cases will be $4n + 1 = 9$. In our example, the set of minimum and maximum values is shown below:

	Account No. (sender)	Transfer credit	Account No. (receiver)
Min Value	12	5000	12
Min+ Value	13	5001	13
Max Value	12	100000	12
Max- Value	11	99999	11
Nominal Value	12	52500	12

Using these values, test cases can be designed as shown below:

Test Case Id	Account No. (sender)	Transfer credit	Account No. (receiver)	Expected Output
01	12	5000	12	Successful
02	12	5001	13	Unsuccessful
03	12	100000	12	Successful
04	12	99999	11	Unsuccessful
05	12	52500	12	Successful
06	13	52500	13	Unsuccessful
07	12	52500	12	Successful
08	11	52500	11	Unsuccessful
09	12	5000	12	Successful
10	13	5001	12	Unsuccessful
11	12	100000	12	Successful
12	11	99999	12	Unsuccessful
13	12	52500	12	Successful

10. “Withdraw” function:

- **Test Case Specification Identifier**
T10
- **Purpose**
To check the functionality of “Withdraw” module
- **Special Environmental Requirements**
To have accessibility to Internet and to have server in running condition
- **Special Procedural Requirements**
Must complete ‘Sign In’ function
- **Inter-case Dependencies**
No Needed
- **Input Specifications**
Account No.: 324789655896, Credit withdrawn: 20000, Remaining credit: 100000
Account No.: 564789655887, Credit withdrawn: 50000, Remaining credit: 200000
- **Test Procedure**
To press “Withdraw” button
- **Output Specifications**
Display message, “Withdraw Successful”.

Boundary Value Checking:

Since there are three variables, the total number of test cases will be $4n + 1 = 13$. In our example, the set of minimum and maximum values is shown below:

	Account No.	Credit withdrawn	Remaining credit
Min Value	12	500	500
Min+ Value	13	501	501
Max Value	12	1000000	5000000
Max- Value	11	999999	4999999
Nominal Value	12	500250	2500250

Using these values, test cases can be designed as shown below:

Test Case Id	Account No.	Credit withdrawn	Remaining credit	Expected Output
01	12	500	500	Successful
02	12	501	501	Successful
03	12	1000000	5000000	Successful
04	12	999999	4999999	Successful
05	12	500250	500	Successful
06	13	500250	501	Unsuccessful
07	12	500250	5000000	Successful
08	11	500250	4999999	Unsuccessful
09	12	500	2500250	Successful
10	13	501	2500250	Unsuccessful
11	12	1000000	2500250	Successful
12	11	999999	2500250	Unsuccessful
13	12	500250	2500250	Successful