EAST WEST UNIVERSITY

Course Code : CSE435
Section : 02

Mini Project Report

on

Test Case Analysis 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

Contents **Page**

Test Plan Identifier

TPI-001

In this project we have decided to identify each feature with unique and different names.

Name	Test ID
Sign Up	UT-101.01
Sign In	IT-101.01
Create Employee ID	CT-101.01
Remove Employee	RT-101.01
Update Profile	PT-101.01
Create Customer ID	AT-101.01
Remove Customer	ET-101.01
Deposit	DT-101.01
Transfer	TT-101.01
Withdraw	WT-101.01

Function Identifier

All the functions in this project have been identified with a unique ID.

Serial No.	Function Name	Function ID/FID
01	Sign Up	FID-01
02	Sign In	FID-02
03	Create Employee ID	FID-03
04	Remove Employee	FID-04
05	Update Profile	FID-05
06	Create Customer ID	FID-06
07	Remove Customer	FID-07
08	Deposit	FID-08
09	Transfer	FID-09
10	Withdraw	FID-10

Introduction

The project is titled as Bank Management System. The project is created for the bank managers and employees. Here we will be testing the features of the software. The goal is to test the features of the software to determine the efficiency, correctness, and effectiveness.

Test Item

Here we will mainly focus testing plans and procedures on the requirement specification of the system. As it gives the functionalities of the software. But we will not conduct testing procedures on design specification, database, and security of the software. As it is our first time testing the software we are identifying the test version 1.0.

Features to be Tested

1. Sign Up
2. Sign In
3. Create Employee ID
4. Remove Employee ID
5. Update Profile
6. Create Customer ID
7. Remove Customer ID
8. Transfer
9. Deposit
10. Withdraw

Features not to be Tested

1. Database Connect ability
2. Security

Approach

Testing Methods

We will use black box testing method to test our software. We will design test cases for the software features. Then we will manually test them.

Testing tools

We will not be using any testing tools as we are manually testing the software features.

Item Pass/Fail Criteria

1. For Sign In/Sign Up, proper information of username, password, email, address etc must be given in order to pass, invalid information will result in failing.
2. The passing criterias for creating employee ID are: the ID must be created from a manager profile and the name, password, email etc must follow proper format. Invalid information and keeping empty fields will result in failing.
3. To remove employee ID, the operation must be conducted from a manager profile and fill the next steps. Not following the steps will result in failing.
4. To create and remove customer ID the passing criterias are: it can be from a manager or employee profile, account number, name, password, signature etc must be in a valid format. Invalid information will result in failing.
5. To transfer/deposit/withdraw the passing criterias are: it must be from a manager or employee account and account number, signature, credit value / transfer amount/ withdrawal amount etc must be valid. Invalid information will result in failing.

Test Deliverables:

- Report on Test Planning
- Report on Test Design
- Report on Execution and Evaluation of the result
- Summary

Responsibilities:

All the plans and strategies are made by us. We are working together as a unit to complete the project.

Schedule:

Test Design and Planning **Day 0**				
	Sign-Up, Sign-In **Day 1 - 4**			
		Creating Customer/Employee ID, Removing Customer/Employee ID and Update Profile **Day 5-10**		
			Deposit, Transfer, Withdraw **Day 11-14**	
				Reviewing All the Completed Tasks **Day 15-16**

Planning Risks and Contingencies:

We have created a schedule for all the procedures and tasks. But in any case, if we fail to meet the schedule, then it will be a major planning risk as we will be falling behind schedule. For this there are some contingency plans.

- We can decrease the number of test cases and keep the important ones. This way, we can keep up with the schedule but the testing will not be to the point accurate as some of the test cases will be untested.
- We can extend the schedule. This way all the test cases will be tested but the whole project will be delayed due to the extension of time.

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
  Must be from a manager ID
- **Inter-case Dependencies**
  No Needed
- **Input Specifications**

User Name(employee): MDMR60253, Password(manager): gjeq57$$CM
User Name(employee): TUBA18049, Password(manager): gjeq57$$CM
User Name(employee): ASD18157, Password(manager): 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: mAqaBM4M#
  Phone No: 8801711578897, Password: gjeq57$$CM
- **Test Procedure**
  To press “Update Profile” button
- **Output Specifications**
  Display message, “Profile Updated 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

Execution and Evaluation of the Result:

Test Incident Report Identifier

TIR-01

Summary

Sign up function and Test case T1

Incident Description

It describes the following:

1. Date and time: 21/05/2023 09.10PM
2. Testing Personnel Names: Adnan
3. Environment : Online Environment with SQL database
4. Testing Inputs
5. Expected Outputs
6. Actual Outputs
7. Anomalies detected during the test
8. Attempts to repeat the same test

Test Inputs	Expected Outputs	Actual Outputs	Anomalies Detected	Attempts to repeat the same test
Adnan 01911112222 973046862504 adnan60157 A12345@123456 adnan@gmail.com	"Registration Successful"	"Registration Successful"	nil	
Mahmud 0184555401 973040222534 mdmahd60253 liM45$44268 limon@gmail.com	Enter correct 11 character Phone number	Enter correct 11 character Phone number	nil	
Tuba 01946224118 973040490494 tubat60049 TubT&00449 tuba@gmail.com	"Registration Successful"	"Registration Successful"	nil	

Test Summary Report Identifier:

TS1

Description:

Serial No	Function Name	Identifier	Function ID	Status
01	Sign Up	T1	FID-01	Pass

Evaluation:
The Sign up function has been tested successfully. There are no bugs in the function.

Test Incident Report Identifier

TIR-02

Summary

Sign In function and Test case T2

Incident Description

It describes the following:

1. Date and time: 21/05/2023 09.20PM
2. Testing Personnel Name: Adnan
3. Environment : Online Environment with SQL database
4. Testing Inputs
5. Expected Outputs
6. Actual Outputs
7. Anomalies detected during the test
8. Attempts to repeat the same test

Test Inputs	Expected Outputs	Actual Outputs	Anomalies Detected	Attempts to repeat the same test
adnan60157 Aa2345@123456	“Welcome to The Dashboard”	“Welcome to The Dashboard”	nil	
mdmahd60253 liM45$44267	Enter correct username or password	Enter correct username or password	nil	
tubat60049 TubT&00449	“Welcome to The Dashboard”	“Welcome to The Dashboard”	nil	

Test Summary Report Identifier:

TS2

Description:

Serial No	Function Name	Identifier	Function ID	Status
02	Sign in	T2	FID-02	Pass

Evaluation:

The Sign-in function has been tested successfully. There are no bugs in the function.

Test Incident Report Identifier

TIR-03

Summary

"Create Employee ID" function and Test case T3

Incident Description

It describes the following:

1. Date and time: 21/05/2023 09.35PM
2. Testing Personnel Name: Adnan
3. Environment : Online Environment with SQL database
4. Testing Inputs
5. Expected Outputs
6. Actual Outputs
7. Anomalies detected during the test
8. Attempts to repeat the same test

Test Inputs	Expected Outputs	Actual Outputs	Anomalies Detected	Attempts to repeat the same test
MDJAD158 01750158253 1592158582534 qASH79@7 gjeq57$$CM	"Profile Successfully Created"	"Profile Successfully Created"	nil	
NAM18018 8801677157600 1577889998455 NyMaBMX3 gjeq57$$CG	Enter correct password	"Profile Successfully Created"	It accepts the password and creates the id	3
ANK10063 8801778855158	"Profile Successfully	"Profile Successfully	nil	

1592133488889 AinK67$$zN gjeq57$$CM	Created”	Created”		

Test Summary Report Identifier:
TS3

Description:

Serial No	Function Name	Identifier	Function ID	Status
03	Create Employee ID	T3	FID-03	Bug Found

Evaluation:
The “Create Employee ID” function could not be tested successfully. There is a bug in the function. The bug causes the system to accept the wrong password for manager id and creates the employee profile.

Test Incident Report Identifier

TIR-04

Summary

Remove Employee ID function and Test case T4

Incident Description

It describes the following:

1. Date and time: 21/05/2023 11.40PM
2. Testing Personnel Names: Limon
3. Environment : Online Environment with SQL database
4. Testing Inputs
5. Expected Outputs
6. Actual Outputs
7. Anomalies detected during the test
8. Attempts to repeat the same test

Test Inputs	Expected Outputs	Actual Outputs	Anomalies Detected	Attempts to repeat the same

				test
ASD18157 gjeq57$$CM yes	“ID Removed Successfully”	“ID Removed Successfully”	nil	
MDMR60253 gjeq57$$CM no	“Termination not Confirmed”	“ID Removed Successfully”	It removes employee ID with choice of termination set to “no”	3
TUBA18049 gjeq57$$M yes	“ID Removed Successfully”	“ID Removed Successfully”	nil	

Test Summary Report Identifier:
TS4

Description:

Serial No	Function Name	Identifier	Function ID	Status
04	Remove Employee ID	T4	FID-04	Bug Found

Evaluation:
The Remove Employee ID function could not be tested successfully. There is a bug in the function. The bug is that the Remove Employee ID function operates after the termination confirmation set to “no’.

Test Incident Report Identifier

TIR-05

Summary

Update Profile function and Test case T5

Incident Description

It describes the following:

1. Date and time: 22/05/2023 12.20AM
2. Testing Personnel Names: Limon
3. Environment : Online Environment with SQL database

4. Testing Inputs
5. Expected Outputs
6. Actual Outputs
7. Anomalies detected during the test
8. Attempts to repeat the same test

Test Inputs	Expected Outputs	Actual Outputs	Anomalies Detected	Attempts to repeat the same test
01749157253 qWeR79@1 limon@gmail.com	"Profile Updated Successfully"	"Profile Updated Successfully"	nil	
8801718157600 mAqaBM4M# adnan@gmail.com	"Profile Updated Successfully"	"Profile Updated Successfully"	nil	
8801711578897 gjeq57$$CM tuba@gmail.com	"Profile Updated Successfully"	"Profile Updated Successfully"	nil	

Test Summary Report Identifier:
TS5

Description:

Serial No	Function Name	Identifier	Function ID	Status
05	Update Profile	T5	FID-05	Pass

Evaluation:
The Update Profile function has been tested successfully. There are no bugs in the system.

Test Incident Report Identifier

TIR-06

Summary

Create Customer ID function and Test case T6

Incident Description

It describes the following:

1. Date and time: 22/05/2023 01.05AM
2. Testing Personnel Names: Limon
3. Environment : Online Environment with SQL database
4. Testing Inputs
5. Expected Outputs
6. Actual Outputs
7. Anomalies detected during the test
8. Attempts to repeat the same test

Test Inputs	Expected Outputs	Actual Outputs	Anomalies Detected	Attempts to repeat the same test
gjeq57$$CM Limon MDMR60253 01749157253 1592824588424 8465284901264 limon1@gmail.com	"ID Created Successfully"	"ID Created Successfully"	nil	
gjeq57$$CM Tuba TUBA18049 8801718157600 1592175598424 1238763900036 tuba1@gmail.com	"ID Created Successfully"	"ID Created Successfully"	nil	
gjeq57$$CM Adnan ASD18157 8801711578897 1592166788224 2645890012631 adnan@gmail.com	"ID Created Successfully"	"ID Created Successfully"	nil	

Test Summary Report Identifier:

TS6

Description:

Serial No	Function Name	Identifier	Function ID	Status
06	Create Customer ID	T6	FID-06	Pass

Evaluation:
The Create Customer ID function has been tested successfully. There are no bugs in the system.

Test Incident Report Identifier

TIR-07

Summary

“Remove Customer ID” function and Test case T7

Incident Description

It describes the following:

1. Date and time: 22/05/2023 01.05AM
2. Testing Personnel Names: Adnan
3. Environment : Online Environment with SQL database
4. Testing Inputs
5. Expected Outputs
6. Actual Outputs
7. Anomalies detected during the test
8. Attempts to repeat the same test

Test Inputs	Expected Outputs	Actual Outputs	Anomalies Detected	Attempts to repeat the same test
8465284901264 qWeR79@1	“ID Removed Successfully”	“ID Removed Successfully”	nil	
1238763900036 mAqaBM4M#	“ID could not be found in Database”	“Profile Updated Successfully”	The ID is not in the database but the bug is creating wrong display	5

			message	
2645890012631 gjeq57$$CM	“ID Removed Successfully”	“ID Removed Successfully”	nil	

Test Summary Report Identifier:
TS7

Description:

Serial No	Function Name	Identifier	Function ID	Status
07	Remove Customer ID	T7	FID-07	Bug Found

Evaluation:
The “Remove Customer ID” function could not be tested successfully. There is a bug in the function. The bug causes the system to display “ID Removed Successfully”, but the ID is not in the database and there is no chance of removing an ID which is not in the database.

Test Incident Report Identifier

TIR-08

Summary

Deposit function and Test case T8

Incident Description

It describes the following:

1. Date and time: 22/05/2023 1:16 PM
2. Testing Personnel Names: Tuba
3. Environment : Online Environment with SQL database
4. Testing Inputs
5. Expected Outputs
6. Actual Outputs
7. Anomalies detected during the test
8. Attempts to repeat the same test

Test Inputs	Expected Outputs	Actual Outputs	Anomalies Detected	Attempts to repeat the same

				test
324789655896 200000	“Deposit Successful”	“Deposit Successful”	nil	
564789655887 500000	“Deposit Successful”	“Deposit Successful”	nil	
564789655887 500000	“Deposit Successful”	“Deposit Successful”	nil	

Test Summary Report Identifier:
TS8

Description:

Serial No	Function Name	Identifier	Function ID	Status
08	Deposit	T8	FID-08	Pass

Evaluation:
The Deposit function has been tested successfully. There are no bugs in the function.

Test Incident Report Identifier
TIR-09
Summary
Transfer function and Test case T9
Incident Description
It describes the following:

1. Date and time: 22/05/2023 1.22AM
2. Testing Personnel Names: Limon
3. Environment : Online Environment with SQL database
4. Testing Inputs
5. Expected Outputs
6. Actual Outputs
7. Anomalies detected during the test
8. Attempts to repeat the same test

Test Inputs	Expected Outputs	Actual Outputs	Anomalies Detected	Attempts to repeat the same

				test
154789655896 20000 669987196558	“Transfer Successful”	“Transfer Successful”	nil	
544789655887 500000 478965589644	“Transfer Failed: Amount Exceeded”	“Transfer Failed: Amount Exceeded”	nil	

Test Summary Report Identifier:
TS9

Description:

Serial No	Function Name	Identifier	Function ID	Status
09	Transfer	T9	FID-09	Pass

Evaluation:
The Update Profile function has been tested successfully. There are no bugs in the system.

Test Incident Report Identifier

TIR-10

Summary

“Withdraw” function and Test case T10

Incident Description

It describes the following:

1. Date and time: 22/05/2023 01.25AM
2. Testing Personnel Names: Adnan
3. Environment : Online Environment with SQL database
4. Testing Inputs
5. Expected Outputs
6. Actual Outputs
7. Anomalies detected during the test
8. Attempts to repeat the same test

Test Inputs	Expected Outputs	Actual Outputs	Anomalies Detected	Attempts to repeat the same

				test
324789655896 20000 100000	“Withdraw Successful”	“Withdraw Successful”	nil	
564789655887 50000 200000	“Withdraw Successful”	“Withdraw Successful”	nil	

Test Summary Report Identifier:
TS10

Description:

Serial No	Function Name	Identifier	Function ID	Status
10	Withdraw	T10	F10	Pass

Evaluation:
The Withdraw function has been tested successfully. There is no bug in the system.

Summary

In this project, we have tested different features for the bank management system software. Our aim was to test these features in order to determine how efficient, flawless and effective the software is. We have chosen 10 features such as “Sign-up/Sign-in”, “create employee ID”, “Update profile” and so on. At first we identified each feature with unique and different names such as CT-101.01 for “Create Employee ID”, PT-101.01 for “Update Profile” etc. After that, All the functions in this project have been identified with a unique feature ID (FID). “Black-box” testing method was used to manually test each feature of our software. Some specific pass/fail criteria were assigned to check the functionality and effectiveness of the test items. For each item, a test case design specification was created where we assigned test case specification identifier, special environmental and procedural requirements, Intercase dependencies, input specifications, test procedure and output specifications. Boundary value checking method was used to calculate the expected output based on the given inputs. Finally, for execution and evaluation of the result, a set of test inputs were given with the following expected output and actual output for bug detection and retesting the features.

All the plans and strategies were executed by us. We worked together as a unit to complete the project. This project was also done in a bounded schedule for testing all the procedures and tasks in such a way that if we fail to meet the schedule, then it will be a major planning risk as we will be falling behind schedule.