



Abdulhaq Zulfiqar

Shahmeer

Usman Asif

SE-G

SQE

Saba Kanwal

BLACK BOX TESTING

Contents

Splash.java:.....	9
Key Input Fields:	9
1. Image Path Input	9
Test Cases for Image Path Input.....	9
2. Screen Dimensions Input	10
Test Cases for Screen Dimensions.....	10
3. Delay/Timing Input	10
Test Cases for Delay/Timing	11
WithdrawMoney.java:	11
Key Input Fields:	11
1. Account Number Input	11
Test Cases for Account Number Input.....	12
2. Withdrawal Amount Input	12
Test Cases for Withdrawal Amount Input.....	12
3. Withdrawal Date Input.....	13
Test Cases for Withdrawal Date Input	13
WithdrawMoney.java:	14
Key Input Fields:	14
1. Account Number Input	14
Test Cases for Account Number Input.....	14
2. Withdrawal Amount Input	15
Test Cases for Withdrawal Amount Input.....	15
3. Withdrawal Date Input.....	16
Test Cases for Withdrawal Date Input	16
ViewCustomer.java:	17
Key Input Fields:	17
1. File Content Input	17
Test Cases for File Content Input.....	17
2. Table Data Input.....	18
Test Cases for Table Data Input	18
3. Table Column Properties	19

Test Cases for Table Column Properties	19
UISwitchListener.java:.....	19
Key Elements to Test:	19
1. Look and Feel Property Change	20
Test Cases for Look and Feel Property Change	20
2. Component Validation	20
Test Cases for Component Validation	21
3. Event Handling Behavior	21
Test Cases for Event Handling	22
NewAccount.java:.....	22
Key Input Fields:	22
1. Account Number Input	22
Test Cases for Account Number Input.....	23
2. Customer Name Input.....	23
Test Cases for Customer Name Input	23
3. Deposit Amount Input	24
Test Cases for Deposit Amount Input.....	24
4. Deposit Date Input	25
Test Cases for Deposit Date Input.....	25
FindAccountName.java:	25
Key Input Fields:	25
1. Customer Name Input.....	26
Test Cases for Customer Name Input	26
2. File Content Input	27
Test Cases for File Content Input.....	27
3. Search Button Behavior	27
Test Cases for Search Button Behavior.....	28
FindName.java:	28
Key Input Fields:	28
1. Customer Name Input.....	28
Test Cases for Customer Name Input.....	29
2. File Content Input	29

Test Cases for File Content Input	30
3. Search Button Behavior	30
Test Cases for Search Button Behavior	30
FindAccount.java:	31
Key Input Fields:	31
1. Account Number Input	31
Test Cases for Account Number Input.....	31
2. File Content Input	32
Test Cases for File Content Input.....	32
3. Search Button Behavior	33
Test Cases for Search Button Behavior.....	33
DepositMoney.java:.....	33
Key Input Fields:	33
1. Account Number Input (txtNo)	34
Test Cases for Account Number Input	34
2. Deposit Amount Input (txtDeposit)	34
Test Cases for Deposit Amount Input.....	35
3. File Content Input (Bank.dat)	35
Test Cases for File Content Input.....	35
4. Save Button Behavior (btnSave)	36
Test Cases for Save Button Behavior	36
DeleteCustomer.java:	37
Key Input Fields:	37
1. Account Number Input (txtNo)	37
Test Cases for Account Number Input.....	37
2. Deletion Button Behavior (btnDel)	38
Test Cases for Deletion Button Behavior	38
3. File Content Input (Bank.dat)	38
Test Cases for File Content Input.....	39
4. Confirmation Dialog for Deletion (deletion())	39
Test Cases for Confirmation Dialog.....	39
5. File Saving After Deletion (deleteFile())	40

Test Cases for File Saving After Deletion.....	40
6. Cancel Button Behavior (btnCancel)	40
Test Cases for Cancel Button Behavior	40
BankHelp.java:	40
Key Input Fields:	40
1. Filename (filename) - Path to the HTML file	41
Test Cases for Filename Input	41
2. Hyperlink URL (url)	41
Test Cases for Hyperlink URL Input	42
3. Hyperlink Handling (linkActivated).....	42
Test Cases for Hyperlink Handling	42
4. Cursor Change During Loading (linkActivated())	43
Test Cases for Cursor Change	43
5. Document Loading (html.setPage())	43
Test Cases for Document Loading	44
6. Error Handling During File Read (IOException)	44
Test Cases for Error Handling	44
7. Cancel Button Behavior (btnCancel)	45
Test Cases for Cancel Button Behavior	45
BankSystem.java:	45
1. Account Management.....	45
1.1 Create a New Account.....	45
Inputs:.....	45
Conditions:.....	45
Test Cases:	46
1.2 Delete an Account.....	47
Inputs:.....	47
Conditions:.....	47
Test Cases:	47
2. Transactions	47
2.1 Deposit Money	47
Inputs:.....	47

Conditions:.....	47
Test Cases:	48
2.2 Withdraw Money	48
Inputs:.....	48
Conditions:.....	48
Test Cases:	49
3. Search	49
3.1 Search by Account Number	49
Inputs:.....	49
Conditions:.....	49
Test Cases:	49
3.2 Search by Name	50
Inputs:.....	50
Conditions:.....	50
Test Cases:	50
4. View	51
4.1 View One by One	51
Inputs:.....	51
Conditions:.....	51
Test Cases:	51
4.2 View All Customers	51
Inputs:.....	51
Conditions:.....	51
Test Cases:	52
5. Help	52
5.1 Help Content	52
Inputs:.....	52
Conditions:.....	52
Test Cases:	52
5.2 Shortcut Help	52
Inputs:.....	52
Conditions:.....	52

Test Cases:	53
6. Deposit Money	53
6.1 Inputs:.....	53
Conditions:.....	53
Test Cases:	53
7. Withdraw Money	54
7.1 Inputs:.....	54
Conditions:.....	54
Test Cases:	54
8. Delete Customer	55
8.1 Inputs:.....	55
Conditions:.....	55
Test Cases:	55
9. Change Background Color	56
9.1 Inputs:.....	56
Conditions:.....	56
Test Cases:	56
10. Print Customer Balance	56
10.1 Inputs:.....	56
Conditions:.....	56
Test Cases:	56
11. Quit BankSystem	57
11.1 Inputs:.....	57
Conditions:.....	57
Test Cases:	57
12. Change Layout Style	58
12.1 Inputs:.....	58
Conditions:.....	58
Test Cases:	58
13. Apply Theme (e.g., Aqua, Green)	58
13.1 Inputs:.....	58
Conditions:.....	58

Test Cases:	58
14. Help Contents	59
14.1 Inputs:.....	59
Conditions:.....	59
Test Cases:	59
15. Help on Shortcuts.....	59
15.1 Inputs:.....	59
Conditions:.....	59
Test Cases:	60
16. About BankSystem	60
16.1 Inputs:.....	60
Conditions:.....	60
Test Cases:	60
Conclusion: Black Box Testing for BankSystem.....	60
1. Functionality Validation:.....	61
2. User Interface (UI) Testing:.....	61
Boundary Value Analysis (BVA) & Equivalence Class Partitioning (ECP):	61
3. Error Handling & Robustness:	61
4. Cross-functional Test Coverage:	61
Key Learnings:	61
• Thorough Test Case Design:.....	61
• Error Handling and User Guidance:	61
• Consistency in UI:.....	62
• Efficiency in Testing:.....	62
What We Learned as a Team:	62
1. Collaboration:	62
2. Real-World Application	62
3. Attention to Detail:.....	62
4. Tools and Techniques:.....	62
Final Thoughts:	62

Splash.java:

Key Input Fields:

1. **Image Path:** Path provided for the splash image ("Splash.jpg").
2. **Screen Dimensions:** The window is dynamically positioned based on screen dimensions.
3. **Delay/Timing:** Simulated processing loop with `for (int i = 0; i <= 1000; i++) {}`.

1. Image Path Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Image File Path	Existing image file (Splash.jpg)	Non-existent file path	Path does not exist	Valid existing file
	Correct format (e.g., .jpg)	Unsupported formats (.txt)	Empty string	.jpg extension

Test Cases for Image Path Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid image path	"Splash.jpg"	Splash screen displays correctly	IP1, V1
TC002	Invalid - Non-existent file	"Invalid.jpg"	Error: File not found	IP2, B1
TC003	Invalid - Unsupported format	"Splash.txt"	Error: Unsupported file format	IP3, V2
TC004	Invalid - Empty string	""	Error: File path cannot be empty	IP4, B3

2. Screen Dimensions Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Screen Width and Height	Values > 0 (e.g., 1920x1080)	Negative values	Width or height = -1	Width/Height = 1
	Standard resolutions (e.g., 1080p)	Extremely large values	Width or height = 0	Maximum reasonable size

Test Cases for Screen Dimensions

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC005	Valid screen dimensions	1920x1080	Splash positioned at screen center	SD1, V1
TC006	Invalid - Negative dimensions	- 1920x1080	Error: Invalid screen dimensions	SD2, B1
TC007	Invalid - Zero dimensions	0x0	Error: Screen dimensions cannot be 0	SD3, B2
TC008	Boundary - Minimum valid dimensions	1x1	Splash displayed correctly	SD4, B3
TC009	Boundary - Maximum valid dimensions	7680x4320	Splash displayed correctly	SD5, V2

3. Delay/Timing Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Processing Time (loop counter)	Loop completes correctly	Exceeds threshold	Negative loop iterations	Reasonable iterations

Test Cases for Delay/Timing

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC010	Valid processing loop	0 to 1000	Loop completes and transitions	DT1, V1
TC011	Invalid - Excessive iterations	0 to 1000000	Unresponsive or delayed transition	DT2, B1
TC012	Invalid - Negative loop count	-100 to 100	Error: Invalid loop iteration	DT3, B2
TC013	Boundary - Minimum iterations	0	Loop completes and transitions	DT4, B3

WithdrawMoney.java:

Key Input Fields:

1. **Account Number** (txtNo): Input must be numeric and correspond to a valid account in the database.
2. **Withdrawal Amount** (txtWithdraw): Input must be numeric and should not exceed the current balance.
3. **Withdrawal Date** (cboMonth, cboDay, cboYear): Input must be a valid date.

1. Account Number Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number (txtNo)	Existing numeric account number	Non-numeric input	Negative numbers	Minimum valid account number (e.g., 1)
		Account does not exist	Empty string	Maximum account number present in the database

Test Cases for Account Number Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid account number	"12345"	Account details are retrieved successfully	AN1, V1
TC002	Invalid - Non-numeric input	"ABCDE"	Error: Account number must be numeric	AN2, V2
TC003	Invalid - Account does not exist	"99999"	Error: Account number does not exist	AN3, B1
TC004	Invalid - Empty input	""	Error: Account number cannot be empty	AN4, B2
TC005	Boundary - Minimum valid account	"1"	Account details are retrieved successfully	AN1, B3
TC006	Boundary - Maximum valid account	"99999"	Account details are retrieved successfully	AN1, B4

2. Withdrawal Amount Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Withdrawal Amount (txtWithdraw)	Numeric, less than current balance	Non-numeric input	Negative values	Minimum valid withdrawal amount (e.g., 1)
		Greater than current balance	Empty string	Maximum equal to current balance

Test Cases for Withdrawal Amount Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC007	Valid withdrawal amount	"500"	Amount successfully withdrawn	WA1, V1
TC008	Invalid - Non-numeric input	"ABC"	Error: Withdrawal amount must be numeric	WA2, V2

TC009	Invalid - Exceeds current balance	"1500"	Error: Cannot withdraw more than current balance	WA3, B1
TC010	Invalid - Empty input	""	Error: Withdrawal amount cannot be empty	WA4, B2
TC011	Boundary - Minimum valid withdrawal	"1"	Amount successfully withdrawn	WA1, B3
TC012	Boundary - Maximum valid withdrawal	"1000"	Amount successfully withdrawn	WA1, B4

3. Withdrawal Date Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Withdrawal Date (cboMonth, cboDay, cboYear)	Valid date	Invalid date	Nonexistent dates (e.g., 30 Feb)	Earliest valid date (e.g., 01 Jan 2000)
		Empty selection	Future date	Latest valid date (e.g., 31 Dec 2015)

Test Cases for Withdrawal Date Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC013	Valid date	"15 August 2010"	Date accepted	WD1, V1
TC014	Invalid - Nonexistent date	"30 February 2010"	Error: Invalid date	WD2, V2
TC015	Invalid - Empty date selection	""	Error: Date must be selected	WD3, B1
TC016	Boundary - Earliest valid date	"01 January 2000"	Date accepted	WD1, B3
TC017	Boundary - Latest valid date	"31 December 2015"	Date accepted	WD1, B4

TC018	Invalid - Future date	"15 August 2025"	Error: Cannot select a future date	WD2, B2
-------	-----------------------	------------------	------------------------------------	---------

WithdrawMoney.java:

Key Input Fields:

1. **Account Number (txtNo):** Numeric input representing a valid account number.
2. **Withdrawal Amount (txtWithdraw):** Numeric input that must not exceed the current balance.
3. **Withdrawal Date (cboMonth, cboDay, cboYear):** A valid date.

1. Account Number Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number (txtNo)	Existing numeric account number	Non-numeric input	Negative numbers	Minimum valid account number (e.g., 1)
		Account does not exist	Empty string	Maximum account number in database

Test Cases for Account Number Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid account number	"12345"	Account details are displayed successfully	AN1, V1

TC002	Invalid - Non-numeric input	"ABCDE"	Error: Account number must be numeric	AN2, V2
TC003	Invalid - Account does not exist	"99999"	Error: Account does not exist	AN3, B1
TC004	Invalid - Empty input	""	Error: Account number cannot be empty	AN4, B2
TC005	Boundary - Minimum valid account	"1"	Account details are displayed successfully	AN1, B3
TC006	Boundary - Maximum valid account	"50000"	Account details are displayed successfully	AN1, B4

2. Withdrawal Amount Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Withdrawal Amount (txtWithdraw)	Positive numeric value \leq balance	Non-numeric input	Negative numbers	Minimum valid withdrawal amount (e.g., 1)
		Greater than current balance	Empty string	Maximum valid withdrawal amount = balance

Test Cases for Withdrawal Amount Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC007	Valid withdrawal amount	"500"	Amount successfully withdrawn	WA1, V1
TC008	Invalid - Non-numeric input	"ABC"	Error: Withdrawal amount must be numeric	WA2, V2
TC009	Invalid - Exceeds current balance	"1500"	Error: Cannot withdraw more than current balance	WA3, B1

TC010	Invalid - Empty input	""	Error: Withdrawal amount cannot be empty	WA4, B2
TC011	Boundary - Minimum valid withdrawal	"1"	Amount successfully withdrawn	WA1, B3
TC012	Boundary - Maximum valid withdrawal	"1000"	Amount successfully withdrawn	WA1, B4

3. Withdrawal Date Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Withdrawal Date (cboMonth, cboDay, cboYear)	Valid date	Invalid date	Nonexistent dates (e.g., 30 Feb)	Earliest valid date (e.g., 01 Jan 2000)
		Empty selection	Future dates beyond current year	Latest valid date (e.g., 31 Dec 2015)

Test Cases for Withdrawal Date Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC013	Valid date	"15 August 2010"	Date accepted	WD1, V1
TC014	Invalid - Nonexistent date	"30 February 2010"	Error: Invalid date	WD2, V2
TC015	Invalid - Empty date selection	""	Error: Date must be selected	WD3, B1
TC016	Boundary - Earliest valid date	"01 January 2000"	Date accepted	WD1, B3
TC017	Boundary - Latest valid date	"31 December 2015"	Date accepted	WD1, B4
TC018	Invalid - Future date	"15 August 2025"	Error: Cannot select a future date	WD2, B2

ViewCustomer.java:

Key Input Fields:

1. **File Content:** Bank.dat file containing customer records.
2. **Table Data:** Populated from the Bank.dat file.

1. File Content Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
File Content (Bank.dat)	File exists, contains valid records	File does not exist	File is empty	File contains one record
	Records formatted correctly	Corrupted file format	Records missing mandatory fields	File contains maximum supported records (500)

Test Cases for File Content Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid file with records	"Bank.dat"	Records are displayed successfully in the table	FC1, V1
TC002	Invalid - File does not exist	"Missing.dat"	Error: File not found	FC2, V2
TC003	Invalid - File is empty	"Empty.dat"	Error: Records file is empty	FC3, B1
TC004	Invalid - Corrupted file format	"Corrupted.dat"	Error: Cannot read file	FC4, V3
TC005	Boundary - One record in file	"OneRecord.dat"	One record displayed successfully	FC1, B2

TC006	Boundary - Maximum records	"MaxRecords.dat"	All 500 records displayed successfully	FC1, B3
--------------	----------------------------	------------------	--	---------

2. Table Data Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Table Data	Valid customer data (Account No., Name, Date, Balance)	Missing Account No.	Account No. is empty	Account No. = 1
		Missing Bank Balance	Balance = negative value	Maximum valid Account No.

Test Cases for Table Data Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC007	Valid table data	"12345, John, 01-Jan-2010, 1000"	Data displayed correctly in the table	TD1, V1
TC008	Invalid - Missing Account No.	", John, 01-Jan-2010, 1000"	Error: Account No. is required	TD2, V2
TC009	Invalid - Missing Bank Balance	"12345, John, 01-Jan-2010, "	Error: Bank Balance is required	TD3, V3
TC010	Invalid - Negative Bank Balance	"12345, John, 01-Jan-2010, - 1000"	Error: Bank Balance cannot be negative	TD4, B1
TC011	Boundary - Minimum Account No.	"1, Alice, 01-Jan-2010, 500"	Data displayed correctly in the table	TD1, B2
TC012	Boundary - Maximum Account No.	"50000, Bob, 01-Jan-2015, 1500"	Data displayed correctly in the table	TD1, B3

3. Table Column Properties

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Column Properties	Correct column names and widths	Missing column name	Column width = 0	Minimum valid column width
		Incorrect column widths	Column width exceeds limit	Maximum valid column width

Test Cases for Table Column Properties

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC013	Valid column properties	"Account No., Name, Date, Balance"	Table displayed with correct column widths and names	CP1, V1
TC014	Invalid - Missing column name	"Account No., , Date, Balance"	Error: Column name is required	CP2, V2
TC015	Invalid - Column width = 0	Width = 0	Error: Column width must be greater than 0	CP3, B1
TC016	Boundary - Minimum column width	Width = 100	Table displayed correctly	CP1, B2
TC017	Boundary - Maximum column width	Width = 500	Table displayed correctly	CP1, B3

UISwitchListener.java:

Key Elements to Test:

1. **Look and Feel Property Change:** Ensures that the lookAndFeel property is correctly identified and updates the component's UI.

2. **Invalid Property Names:** Ensures the listener ignores unrelated property changes.
3. **Component Validation:** Ensures the component is properly validated and repainted.

1. Look and Feel Property Change

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Property Name	"lookAndFeel"	Any other property name	Empty property name	Exact match "lookAndFeel"
Component Update Behavior	UI updates successfully	No update performed	Null component reference	Non-null component reference

Test Cases for Look and Feel Property Change

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid property name	"lookAndFeel"	UI is updated, component repainted	LAF1, V1
TC002	Invalid - Unrelated property name	"backgroundColor"	No action performed	LAF2, V2
TC003	Invalid - Empty property name	""	No action performed	LAF3, B1
TC004	Boundary - Exact match "lookAndFeel"	"lookAndFeel"	UI is updated, component repainted	LAF1, B2

2. Component Validation

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
------------	------------------	--------------------	--------------------	------------------

Component Reference	Non-null JComponent reference	Null reference	Component not initialized	Fully initialized component
Validation and Repaint Behavior	Component validates and repaints successfully	No validation or repaint occurs	Partial repaint	Complete repaint

Test Cases for Component Validation

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC005	Valid component reference	Initialized JPanel instance	Component validated and repainted successfully	CV1, V1
TC006	Invalid - Null component reference	null	Error or no action performed	CV2, V2
TC007	Invalid - Partially initialized component	Partially initialized JPanel	Component fails to repaint	CV3, B1
TC008	Boundary - Fully initialized component	Fully initialized JPanel	Component validated and repainted successfully	CV1, B2

3. Event Handling Behavior

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
PropertyChangeEvent Object	Properly initialized with "lookAndFeel"	Event with null source	Event with no property name	Event with valid property name

Test Cases for Event Handling

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC009	Valid event	PropertyChangeEvent with "lookAndFeel"	Listener updates UI	EH1, V1
TC010	Invalid - Null event source	PropertyChangeEvent with null source	Error or no action performed	EH2, V2
TC011	Invalid - Event without property name	PropertyChangeEvent with empty property name	No action performed	EH3, B1
TC012	Boundary - Valid property name event	PropertyChangeEvent with "lookAndFeel"	Listener updates UI	EH1, B2

NewAccount.java:

Key Input Fields:

1. **Account Number (txtNo):** Numeric input for a new account.
2. **Customer Name (txtName):** Text input for the customer's name.
3. **Deposit Amount (txtDeposit):** Numeric input for the initial deposit.
4. **Deposit Date (cboMonth, cboDay, cboYear):** Input for the account's opening date.

1. Account Number Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number (txtNo)	Unique numeric account number	Non-numeric input	Negative numbers	Minimum valid account number (1)

		Duplicate account number	Empty string	Maximum account number allowed
--	--	--------------------------	--------------	--------------------------------

Test Cases for Account Number Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid account number	"12345"	Account is created successfully	AN1, V1
TC002	Invalid - Non-numeric input	"ABCDE"	Error: Account number must be numeric	AN2, V2
TC003	Invalid - Duplicate account	"12345"	Error: Account number already exists	AN3, B1
TC004	Invalid - Empty input	""	Error: Account number cannot be empty	AN4, B2
TC005	Boundary - Minimum valid account	"1"	Account is created successfully	AN1, B3

2. Customer Name Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Customer Name (txtName)	Alphanumeric name, ≤ 50 characters	Contains special characters	Name exceeds character limit	Minimum valid name length (1 character)
		Empty input		Maximum valid name length (50 characters)

Test Cases for Customer Name Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC006	Valid customer name	"John Doe"	Account is created successfully	CN1, V1

TC007	Invalid - Contains special chars	"John@123"	Error: Name cannot contain special characters	CN2, V2
TC008	Invalid - Empty input	""	Error: Name cannot be empty	CN3, B1
TC009	Boundary - Minimum valid length	"J"	Account is created successfully	CN1, B2
TC010	Boundary - Maximum valid length	50 chars	Account is created successfully	CN1, B3

3. Deposit Amount Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Deposit Amount (txtDeposit)	Positive numeric value	Non-numeric input	Negative deposit	Minimum valid deposit (1)
		Empty string		Maximum valid deposit value

Test Cases for Deposit Amount Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC011	Valid deposit amount	"1000"	Account is created successfully	DA1, V1
TC012	Invalid - Non-numeric input	"ABC"	Error: Deposit amount must be numeric	DA2, V2
TC013	Invalid - Negative deposit	"-500"	Error: Deposit amount must be positive	DA3, B1
TC014	Invalid - Empty input	""	Error: Deposit amount cannot be empty	DA4, B2

TC015	Boundary - Minimum valid deposit	"1"	Account is created successfully	DA1, B3
-------	----------------------------------	-----	---------------------------------	---------

4. Deposit Date Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Deposit Date (cboMonth, cboDay, cboYear)	Valid date	Invalid date	Nonexistent dates (e.g., 30 Feb)	Earliest valid date (01 Jan 2000)
		Empty date selection	Future date beyond current year	Latest valid date (31 Dec 2015)

Test Cases for Deposit Date Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC016	Valid date	"15 August 2010"	Date accepted, account created	DD1, V1
TC017	Invalid - Nonexistent date	"30 February 2010"	Error: Invalid date	DD2, V2
TC018	Invalid - Empty date selection	""	Error: Date must be selected	DD3, B1
TC019	Boundary - Earliest valid date	"01 January 2000"	Date accepted, account created	DD1, B2
TC020	Boundary - Latest valid date	"31 December 2015"	Date accepted, account created	DD1, B3

FindAccountName.java:

Key Input Fields:

1. **Customer Name (txtName):** Text input for searching the account holder.
2. **File Content (Bank.dat):** The data file containing records of customers.

1. Customer Name Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Customer Name (txtName)	Alphanumeric name matching a record	Name does not match any record	Empty name	Minimum valid name length (1 character)
		Contains special characters		Maximum valid name length (e.g., 50 characters)

Test Cases for Customer Name Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid customer name	"John Doe"	Account details are displayed successfully	CN1, V1
TC002	Invalid - Name not found	"Jane Smith"	Error: Customer does not exist	CN2, V2
TC003	Invalid - Empty name input	""	Error: Name field cannot be empty	CN3, B1
TC004	Invalid - Contains special characters	"John@123"	Error: Name cannot contain special characters	CN4, V3
TC005	Boundary - Minimum valid length	"A"	Account details are displayed successfully	CN1, B2
TC006	Boundary - Maximum valid length	"NameWith50Chars"	Account details are displayed successfully	CN1, B3

2. File Content Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
File Content (Bank.dat)	File exists and contains valid data	File does not exist	File is empty	File contains one record
		Corrupted file format		File contains maximum records (500)

Test Cases for File Content Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC007	Valid file with records	"Bank.dat"	Records are loaded successfully	FC1, V1
TC008	Invalid - File does not exist	"Missing.dat"	Error: File not found	FC2, V2
TC009	Invalid - File is empty	"Empty.dat"	Error: Records file is empty	FC3, B1
TC010	Invalid - Corrupted file format	"Corrupted.dat"	Error: Cannot read file	FC4, V3
TC011	Boundary - One record in file	"OneRecord.dat"	One record is displayed successfully	FC1, B2
TC012	Boundary - Maximum records in file	"MaxRecords.dat"	All records are loaded successfully	FC1, B3

3. Search Button Behavior

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
------------	------------------	--------------------	--------------------	------------------

Search Button (btnFind)	Enabled and performs search	Disabled	Null input in txtName	Valid name input present
Cancel Button (btnCancel)	Closes the window	No effect		

Test Cases for Search Button Behavior

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC013	Valid search operation	"John Doe"	Account details are displayed successfully	SB1, V1
TC014	Invalid - Search button disabled	Search disabled state	No action performed	SB2, V2
TC015	Invalid - Empty name input	""	Error: Name field cannot be empty	SB3, B1
TC016	Cancel button operation	Button clicked	Window is closed	CB1, V1

FindName.java:

Key Input Fields:

1. **Customer Name (txtName):** Text input for searching the customer.
2. **File Content (Bank.dat):** The data file containing records of customers.

1. Customer Name Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Customer Name (txtName)	Alphanumeric name matching a record	Name does not match any record	Empty name	Minimum valid name length (1 character)

		Contains special characters		Maximum valid name length (e.g., 50 characters)
--	--	-----------------------------	--	---

Test Cases for Customer Name Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid customer name	"John Doe"	Account details are displayed successfully	CN1, V1
TC002	Invalid - Name not found	"Jane Smith"	Error: Customer does not exist	CN2, V2
TC003	Invalid - Empty name input	""	Error: Name field cannot be empty	CN3, B1
TC004	Invalid - Contains special characters	"John@123"	Error: Name cannot contain special characters	CN4, V3
TC005	Boundary - Minimum valid length	"A"	Account details are displayed successfully	CN1, B2
TC006	Boundary - Maximum valid length	"NameWith50Chars"	Account details are displayed successfully	CN1, B3

2. File Content Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
File Content (Bank.dat)	File exists and contains valid data	File does not exist	File is empty	File contains one record

		Corrupted file format		File contains maximum records (500)
--	--	-----------------------	--	-------------------------------------

Test Cases for File Content Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC007	Valid file with records	"Bank.dat"	Records are loaded successfully	FC1, V1
TC008	Invalid - File does not exist	"Missing.dat"	Error: File not found	FC2, V2
TC009	Invalid - File is empty	"Empty.dat"	Error: Records file is empty	FC3, B1
TC010	Invalid - Corrupted file format	"Corrupted.dat"	Error: Cannot read file	FC4, V3
TC011	Boundary - One record in file	"OneRecord.dat"	One record is displayed successfully	FC1, B2
TC012	Boundary - Maximum records in file	"MaxRecords.dat"	All records are loaded successfully	FC1, B3

3. Search Button Behavior

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Search Button (btnFind)	Enabled and performs search	Disabled	Null input in txtName	Valid name input present
Cancel Button (btnCancel)	Closes the window	No effect		

Test Cases for Search Button Behavior

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
--------------	-------------	-------	------------------	------------------

TC013	Valid search operation	"John Doe"	Account details are displayed successfully	SB1, V1
TC014	Invalid - Search button disabled	Search disabled state	No action performed	SB2, V2
TC015	Invalid - Empty name input	""	Error: Name field cannot be empty	SB3, B1
TC016	Cancel button operation	Button clicked	Window is closed	CB1, V1

FindAccount.java:

Key Input Fields:

1. **Account Number (txtNo)**: Numeric input for searching the customer account.
2. **File Content (Bank.dat)**: The data file containing records of customers.

1. Account Number Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number (txtNo)	Numeric account number matching a record	Non-numeric input	Negative numbers	Minimum valid account number (1)
		Account number does not exist	Empty input	Maximum account number allowed

Test Cases for Account Number Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid account number	"12345"	Account details are displayed successfully	AN1, V1

TC002	Invalid - Non-numeric input	"ABCDE"	Error: Account number must be numeric	AN2, V2
TC003	Invalid - Account does not exist	"54321"	Error: Account does not exist	AN3, V3
TC004	Invalid - Empty input	""	Error: Account number cannot be empty	AN4, B1
TC005	Boundary - Minimum valid account	"1"	Account details are displayed successfully	AN1, B2

2. File Content Input

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
File Content (Bank.dat)	File exists and contains valid data	File does not exist	File is empty	File contains one record
		Corrupted file format		File contains maximum records (500)

Test Cases for File Content Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC006	Valid file with records	"Bank.dat"	Records are loaded successfully	FC1, V1
TC007	Invalid - File does not exist	"Missing.dat"	Error: File not found	FC2, V2
TC008	Invalid - File is empty	"Empty.dat"	Error: Records file is empty	FC3, B1
TC009	Invalid - Corrupted file format	"Corrupted.dat"	Error: Cannot read file	FC4, V3
TC010	Boundary - One record in file	"OneRecord.dat"	One record is displayed successfully	FC1, B2

TC011	Boundary - Maximum records in file	"MaxRecords.dat"	All records are loaded successfully	FC1, B3
-------	------------------------------------	------------------	-------------------------------------	---------

3. Search Button Behavior

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Search Button (btnFind)	Enabled and performs search	Disabled	Null input in txtNo	Valid account number present
Cancel Button (btnCancel)	Closes the window	No effect		

Test Cases for Search Button Behavior

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC012	Valid search operation	"12345"	Account details are displayed successfully	SB1, V1
TC013	Invalid - Search button disabled	Search disabled state	No action performed	SB2, V2
TC014	Invalid - Empty input	""	Error: Account number cannot be empty	SB3, B1
TC015	Cancel button operation	Button clicked	Window is closed	CB1, V1

DepositMoney.java:

Key Input Fields:

1. **Account Number (txtNo):** Numeric input for identifying the customer account.
2. **Deposit Amount (txtDeposit):** Numeric input for specifying the deposit amount.
3. **File Content (Bank.dat):** The data file containing records of customers.

1. Account Number Input (txtNo)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number (txtNo)	Numeric account number matching a record	Non-numeric input	Negative numbers	Minimum valid account number (1)
		Account number does not exist	Empty input	Maximum account number allowed

Test Cases for Account Number Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid account number	"12345"	Account details are displayed successfully	AN1, V1
TC002	Invalid - Non-numeric input	"ABCDE"	Error: Account number must be numeric	AN2, V2
TC003	Invalid - Account does not exist	"54321"	Error: Account does not exist	AN3, V3
TC004	Invalid - Empty input	""	Error: Account number cannot be empty	AN4, B1
TC005	Boundary - Minimum valid account	"1"	Account details are displayed successfully	AN1, B2

2. Deposit Amount Input (txtDeposit)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Deposit Amount (txtDeposit)	Positive numeric value	Non-numeric input	Negative numbers	Minimum valid deposit (1)

		Zero value		Maximum allowed deposit
--	--	------------	--	-------------------------

Test Cases for Deposit Amount Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC006	Valid deposit amount	"1000"	Deposit processed and balance updated successfully	DA1, V1
TC007	Invalid - Non-numeric input	"ABC"	Error: Deposit amount must be numeric	DA2, V2
TC008	Invalid - Negative deposit	"-500"	Error: Deposit amount cannot be negative	DA3, B1
TC009	Invalid - Zero deposit	"0"	Error: Deposit amount cannot be zero	DA4, V3
TC010	Boundary - Minimum deposit	"1"	Deposit processed and balance updated successfully	DA1, B2

3. File Content Input (Bank.dat)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
File Content (Bank.dat)	File exists and contains valid data	File does not exist	File is empty	File contains one record
		Corrupted file format		File contains maximum records (500)

Test Cases for File Content Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC011	Valid file with records	"Bank.dat"	Records are loaded successfully	FC1, V1

TC012	Invalid - File does not exist	"Missing.dat"	Error: File not found	FC2, V2
TC013	Invalid - File is empty	"Empty.dat"	Error: Records file is empty	FC3, B1
TC014	Invalid - Corrupted file format	"Corrupted.dat"	Error: Cannot read file	FC4, V3
TC015	Boundary - One record in file	"OneRecord.dat"	One record is displayed successfully	FC1, B2

4. Save Button Behavior (btnSave)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Save Button (btnSave)	Enabled with valid input	Disabled or invalid input	Missing deposit amount	Valid account and deposit
Cancel Button (btnCancel)	Closes the window	No effect		

Test Cases for Save Button Behavior

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC016	Valid save operation	"12345", "1000"	Deposit processed and balance updated successfully	SB1, V1
TC017	Invalid - Save button disabled	Disabled state	No action performed	SB2, V2
TC018	Invalid - Missing deposit amount	"12345", ""	Error: Deposit amount cannot be empty	SB3, B1
TC019	Cancel button operation	Button clicked	Window is closed	CB1, V1

DeleteCustomer.java:

Key Input Fields:

1. **Account Number (txtNo):** Numeric input for the account number that the user wants to delete.
2. **File Content (Bank.dat):** The data file containing customer records to be deleted.

1. Account Number Input (txtNo)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number (txtNo)	Numeric account number matching a record	Non-numeric input	Negative numbers	Minimum valid account number (1)
		Account number does not exist	Empty input	Maximum account number allowed

Test Cases for Account Number Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid account number	"12345"	Account details are displayed successfully	AN1, V1
TC002	Invalid - Non-numeric input	"ABCDE"	Error: Account number must be numeric	AN2, V2
TC003	Invalid - Account does not exist	"54321"	Error: Account does not exist	AN3, V3
TC004	Invalid - Empty input	""	Error: Account number cannot be empty	AN4, B1
TC005	Boundary - Minimum valid account	"1"	Account details are displayed successfully	AN1, B2

2. Deletion Button Behavior (btnDel)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Delete Button (btnDel)	Enabled with valid input	Disabled with invalid input	Missing account number	Account exists and is valid
Cancel Button (btnCancel)	Closes the window	No effect		

Test Cases for Deletion Button Behavior

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC006	Valid delete operation	"12345"	Account is deleted and message shown	DB1, V1
TC007	Invalid - Delete button disabled	Disabled state	No action performed	DB2, V2
TC008	Invalid - Missing account number	""	Error: Account number cannot be empty	DB3, B1
TC009	Cancel button operation	Button clicked	Window is closed	CB1, V1

3. File Content Input (Bank.dat)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
File Content (Bank.dat)	File exists and contains valid data	File does not exist	File is empty	File contains one record
		Corrupted file format		File contains maximum records (500)

Test Cases for File Content Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC010	Valid file with records	"Bank.dat"	Records are loaded successfully	FC1, V1
TC011	Invalid - File does not exist	"Missing.dat"	Error: File not found	FC2, V2
TC012	Invalid - File is empty	"Empty.dat"	Error: Records file is empty	FC3, B1
TC013	Invalid - Corrupted file format	"Corrupted.dat"	Error: Cannot read file	FC4, V3
TC014	Boundary - One record in file	"OneRecord.dat"	One record is displayed successfully	FC1, B2

4. Confirmation Dialog for Deletion (deletion())

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Confirmation Dialog (deletion())	User confirms deletion	User cancels deletion	Invalid action or exception	Valid confirmation

Test Cases for Confirmation Dialog

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC015	User confirms deletion	YES	Account is deleted and message shown	CD1, V1
TC016	User cancels deletion	NO	No changes made, deletion is canceled	CD2, V2

5. File Saving After Deletion (deleteFile())

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Save After Deletion (deleteFile())	Successfully saves after deletion	File I/O error	File corruption during write	Successfully saved file after deletion

Test Cases for File Saving After Deletion

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC017	Valid save after deletion	"12345"	File is updated with record deleted	FS1, V1
TC018	Invalid - File write error	"12345"	Error: Problem with file saving	FS2, V2

6. Cancel Button Behavior (btnCancel)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Cancel Button (btnCancel)	Closes the window successfully	No effect		

Test Cases for Cancel Button Behavior

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC019	Cancel operation	Button clicked	Window is closed successfully	CB1, V1

BankHelp.java:

Key Input Fields:

1. **Filename (filename):** Path to the HTML file to be loaded in the JEditorPane.
2. **Hyperlink URL (url):** Hyperlinks embedded in the HTML file.

1. Filename (filename) - Path to the HTML file

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Filename (filename)	Valid HTML file path	Invalid or non-existent path	Empty or null path	Minimum file size (1 KB)
	Valid file format (.html)	Invalid file format	Malformed URL path	Maximum file size

Test Cases for Filename Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC001	Valid file path and format	"help.html"	HTML content loaded and displayed	FN1, V1
TC002	Invalid - file path does not exist	"missing_file.html"	Error: File not found	FN2, V2
TC003	Invalid - non-HTML file format	"file.txt"	Error: Unsupported file format	FN3, V3
TC004	Invalid - Empty path	""	Error: No file path provided	FN4, B1
TC005	Boundary - Minimal valid HTML file	"small_help.html"	HTML content loaded successfully	FN5, B2

2. Hyperlink URL (url)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Hyperlink (url)	Valid URL embedded in the HTML	Invalid URL format	Missing URL (empty link)	URL that points to valid resources

Test Cases for Hyperlink URL Input

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC006	Valid hyperlink URL in the HTML file	"http://example.com"	URL is activated, content loaded successfully	HU1, V1
TC007	Invalid URL in HTML file	"http://invalid.url"	Error: Invalid URL, no content loaded	HU2, V2
TC008	Hyperlink points to an empty URL	""	Error: No URL specified	HU3, B1
TC009	Boundary - Hyperlink pointing to a local file	"file://help.html"	Local file is loaded successfully	HU4, B2

3. Hyperlink Handling (linkActivated)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Hyperlink Handling (linkActivated)	Valid link activation	Invalid link activation	Empty or malformed link	URL links to external pages

Test Cases for Hyperlink Handling

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC010	Valid link activation	"http://example.com"	Link is activated, page is loaded successfully	LH1, V1
TC011	Invalid link activation (non-existent URL)	"http://invalid.url"	Error: Link cannot be opened, invalid URL	LH2, V2

TC012	Link pointing to an empty page	"http://example.com/empty"	Error: Empty page, no content displayed	LH3, B1
--------------	--------------------------------	----------------------------	---	---------

4. Cursor Change During Loading (linkActivated())

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Cursor change during hyperlink loading	Valid cursor change to waiting	No cursor change during loading	Invalid cursor handling	Cursor resets after loading

Test Cases for Cursor Change

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC013	Valid cursor change to waiting cursor	"http://example.com"	Cursor changes to wait while loading page	CC1, V1
TC014	Invalid cursor handling during loading	"http://invalid.url"	Cursor remains unchanged on invalid URL	CC2, V2

5. Document Loading (html.setPage())

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Document loading (html.setPage())	Valid HTML content loading	Invalid content loading	File loading errors	Maximum content size loaded

Test Cases for Document Loading

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC015	Valid HTML page loading	"http://example.com/page1.html"	Content is displayed successfully	DL1, V1
TC016	Invalid HTML page URL	"http://invalid.url"	Error: Unable to load the page, error displayed	DL2, V2
TC017	Large HTML page loading	"http://example.com/large_page.html"	Content loads fully, page is large but rendered correctly	DL3, B2

6. Error Handling During File Read (IOException)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
IOException handling during file reading	File successfully read	Error in file reading	File reading errors	Large files handled well

Test Cases for Error Handling

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
--------------	-------------	-------	------------------	------------------

TC018	Valid file reading	"help.html"	File is loaded and displayed successfully	EH1, V1
TC019	Invalid file reading (corrupted file)	"corrupted.html"	Error: Cannot read file, fallback to previous state	EH2, V2

7. Cancel Button Behavior (btnCancel)

Conditions	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Cancel Button (btnCancel)	Closes the window	No effect when clicked	Invalid interaction	

Test Cases for Cancel Button Behavior

Test Case ID	Description	Input	Expected Outcome	New Tags Covered
TC020	Valid cancel operation	Button clicked	Window is closed successfully	CB1, V1

BankSystem.java:

1. Account Management

1.1 Create a New Account

Inputs:

- **Account Number (txtNo):** Must be unique and numeric.
- **Name (txtName):** Alphabetic characters.
- **Deposit Amount (txtDeposit):** Numeric value greater than zero.

Conditions:

Field	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
-------	------------------	--------------------	--------------------	------------------

Account Number	Unique, numeric > 0	Non-numeric, already existing account number, or empty	0, -1, abc	1 (minimum valid), 12345
Name	Length ≤ 50, alphabetic	Numeric, special characters, empty	Length = 0	Length = 1, Length = 50
Deposit Amount	Greater than 0	Negative, 0, or non-numeric	-1, 0, abc	1, 10000

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC001	Valid new account	No: 100, Name: John, Deposit: 500	Success message; account created	V1, V2, V3
TC002	Invalid - Duplicate account number	No: 100, Name: John, Deposit: 500	Error: Account number already exists	IV1
TC003	Invalid - Non-numeric account number	No: abc, Name: John, Deposit: 500	Error: Account number must be numeric	IV2
TC004	Boundary - Minimum valid account number	No: 1, Name: John, Deposit: 500	Success message; account created	VB1
TC005	Boundary - Minimum valid deposit amount	No: 101, Name: Jane, Deposit: 1	Success message; account created	VB2
TC006	Invalid - Zero deposit amount	No: 102, Name: Jane, Deposit: 0	Error: Deposit amount must be > 0	IV3
TC007	Invalid - Empty name	No: 103, Name: ` `, Deposit: 500	Error: Name cannot be empty	IV4
TC008	Boundary - Maximum valid name length	No: 104, Name: A... (50 chars), Deposit: 500	Success message; account created	VB3

1.2 Delete an Account

Inputs:

- **Account Number (txtNo):** Numeric and must exist in the database.

Conditions:

Field	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number	Numeric, exists in database	Non-existent, non-numeric, empty	0, -1, abc	1, 12345

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC009	Valid account deletion	No: 100	Success: Account deleted	V1
TC010	Invalid - Non-existent account number	No: 999	Error: Account does not exist	IV1
TC011	Invalid - Non-numeric account number	No: abc	Error: Account number must be numeric	IV2
TC012	Boundary - Minimum valid account	No: 1	Success: Account deleted	VB1

2. Transactions

2.1 Deposit Money

Inputs:

- **Account Number (txtNo):** Numeric and must exist in the database.
- **Deposit Amount (txtDeposit):** Numeric value greater than zero.

Conditions:

Field	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
-------	------------------	--------------------	--------------------	------------------

Account Number	Numeric, exists in database	Non-existent, non-numeric, empty	0, -1, abc	1, 12345
Deposit Amount	Greater than 0	Negative, 0, or non-numeric	-1, 0, abc	1, 10000

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC013	Valid deposit	No: 100, Amount: 500	Success: Deposit added successfully	V1, V2
TC014	Invalid - Non-existent account	No: 999, Amount: 500	Error: Account does not exist	IV1
TC015	Invalid - Non-numeric account number	No: abc, Amount: 500	Error: Account number must be numeric	IV2
TC016	Invalid - Zero deposit amount	No: 100, Amount: 0	Error: Deposit amount must be > 0	IV3
TC017	Boundary - Minimum valid deposit	No: 101, Amount: 1	Success: Deposit added successfully	VB1

2.2 Withdraw Money

Inputs:

- **Account Number (txtNo):** Numeric and must exist in the database.
- **Withdraw Amount (txtWithdraw):** Numeric value greater than zero and \leq current balance.

Conditions:

Field	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number	Numeric, exists in database	Non-existent, non-numeric, empty	0, -1, abc	1, 12345
Withdraw Amount	Greater than 0 and \leq balance	Negative, 0, > balance, or non-numeric	-1, 0, abc	1, balance

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC018	Valid withdrawal	No: 100, Amount: 500	Success: Withdrawal processed	V1, V2
TC019	Invalid - Non-existent account	No: 999, Amount: 500	Error: Account does not exist	IV1
TC020	Invalid - Non-numeric account number	No: abc, Amount: 500	Error: Account number must be numeric	IV2
TC021	Invalid - Withdraw amount > balance	No: 100, Amount: 2000	Error: Insufficient balance	IV3
TC022	Invalid - Zero withdraw amount	No: 100, Amount: 0	Error: Withdraw amount must be > 0	IV4
TC023	Boundary - Maximum withdraw amount	No: 101, Amount: Balance	Success: Withdrawal processed	VB1

3. Search

3.1 Search by Account Number

Inputs:

- **Account Number (txtNo):** Numeric and must exist in the database.

Conditions:

Field	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number	Numeric, exists in database	Non-existent, non-numeric, empty	0, -1, abc	1, 12345

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
--------------	-------------	-------	------------------	------

TC024	Valid search	No: 100	Success: Account details displayed	V1
TC025	Invalid - Non-existent account	No: 999	Error: Account does not exist	IV1
TC026	Invalid - Non-numeric account number	No: abc	Error: Account number must be numeric	IV2
TC027	Invalid - Empty account number	No: `` (empty)	Error: Account number cannot be empty	IV3
TC028	Boundary - Minimum valid account	No: 1	Success: Account details displayed	VB1

3.2 Search by Name

Inputs:

- **Name (txtName):** Alphabetic and must exist in the database.

Conditions:

Field	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Name	Alphabetic, exists in DB	Non-existent, numeric, special chars, empty	Length = 0, 123	Length = 1, Length = 50

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC029	Valid search by name	Name: John	Success: Account details displayed	V1
TC030	Invalid - Non-existent name	Name: Jane	Error: Name does not exist	IV1
TC031	Invalid - Name with numbers	Name: John123	Error: Name must contain only letters	IV2
TC032	Invalid - Empty name	Name: `` (empty)	Error: Name cannot be empty	IV3

TC033	Boundary - Minimum valid name length	Name: A	Success: Account details displayed	VB1
--------------	--------------------------------------	---------	------------------------------------	-----

4. View

4.1 View One by One

Inputs:

- No direct inputs; sequential navigation through account records.

Conditions:

Field	Valid Partitions	Invalid Partitions
Navigation	Forward and backward allowed	Invalid navigation commands
Records Existence	Database has records	Database is empty

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC034	Valid navigation	Forward navigation	Success: Next record displayed	V1
TC035	Valid navigation	Backward navigation	Success: Previous record displayed	V2
TC036	Invalid navigation - Empty database	Empty database	Error: No records to display	IV1

4.2 View All Customers

Inputs:

- No direct inputs; fetches all customer records from the database.

Conditions:

Field	Valid Partitions	Invalid Partitions
Records Existence	Database has records	Database is empty

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC037	Valid - Records exist	Database with records	Success: All customer details displayed	V1
TC038	Invalid - No records	Empty database	Error: No customer records available	IV1

5. Help

5.1 Help Content

Inputs:

- No direct inputs; fetches static HTML help content.

Conditions:

Field	Valid Partitions	Invalid Partitions
File Availability	Help file exists	Help file missing or corrupted

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC039	Valid - Help content available	Help file exists	Success: Help content displayed	V1
TC040	Invalid - Help file missing	Missing file	Error: Help content could not be loaded	IV1

5.2 Shortcut Help

Inputs:

- No direct inputs; fetches static shortcut help content.

Conditions:

Field	Valid Partitions	Invalid Partitions
-------	------------------	--------------------

File Availability	Shortcut help file exists	Shortcut help file missing or corrupted
--------------------------	---------------------------	---

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC041	Valid - Shortcut help available	Shortcut file exists	Success: Shortcut content displayed	V1
TC042	Invalid - Shortcut file missing	Missing file	Error: Shortcut content could not be loaded	IV1

6. Deposit Money

6.1 Inputs:

- **Account Number (txtNo):** Numeric, must exist in the database.
- **Deposit Amount (txtDeposit):** Positive numeric value.

Conditions:

Field	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number	Numeric, exists in DB	Non-existent, non-numeric, empty	0, -1, abc	1, 12345
Deposit Amount	Positive (> 0) numeric	Negative, zero, non-numeric, empty	0, -1, abc	1, 9999999

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC043	Valid deposit	No: 100, Amt: 500	Success: Amount deposited successfully	V1, V2
TC044	Invalid - Non-existent account	No: 999, Amt: 500	Error: Account does not exist	IV1
TC045	Invalid - Non-numeric account number	No: abc, Amt: 500	Error: Account number must be numeric	IV2

TC046	Invalid - Empty account number	No: `` (empty), Amt: 500	Error: Account number cannot be empty	IV3
TC047	Invalid - Zero deposit amount	No: 100, Amt: 0	Error: Deposit amount must be greater than 0	IV4
TC048	Boundary - Minimum valid deposit	No: 100, Amt: 1	Success: Amount deposited successfully	VB1

7. Withdraw Money

7.1 Inputs:

- **Account Number (txtNo):** Numeric, must exist in the database.
- **Withdraw Amount (txtWithdraw):** Positive numeric value, less than or equal to account balance.

Conditions:

Field	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number	Numeric, exists in DB	Non-existent, non-numeric, empty	0, -1, abc	1, 12345
Withdraw Amount	<= Balance, > 0 numeric	Greater than balance, zero, negative, empty	0, -1, abc	1, balance

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC049	Valid withdrawal	No: 100, Amt: 200	Success: Amount withdrawn successfully	V1, V2
TC050	Invalid - Non-existent account	No: 999, Amt: 200	Error: Account does not exist	IV1
TC051	Invalid - Non-numeric account number	No: abc, Amt: 200	Error: Account number must be numeric	IV2

TC052	Invalid - Empty account number	No: `` (empty), Amt: 200	Error: Account number cannot be empty	IV3
TC053	Invalid - Withdraw amount > balance	No: 100, Amt: 5000	Error: Insufficient funds	IV4
TC054	Boundary - Minimum valid withdraw	No: 100, Amt: 1	Success: Amount withdrawn successfully	VB1

8. Delete Customer

8.1 Inputs:

- **Account Number (txtNo):** Numeric, must exist in the database.

Conditions:

Field	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number	Numeric, exists in DB	Non-existent, non-numeric, empty	0, -1, abc	1, 12345

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC055	Valid delete	No: 100	Success: Customer account deleted	V1
TC056	Invalid - Non-existent account	No: 999	Error: Account does not exist	IV1
TC057	Invalid - Non-numeric account number	No: abc	Error: Account number must be numeric	IV2
TC058	Invalid - Empty account number	No: `` (empty)	Error: Account number cannot be empty	IV3
TC059	Boundary - Minimum valid account	No: 1	Success: Customer account deleted	VB1

9. Change Background Color

9.1 Inputs:

- **Color Selection:** GUI-based input for choosing a valid color.

Conditions:

Field	Valid Partitions	Invalid Partitions
Color Input	Any color in color picker	Invalid or corrupted GUI action

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC060	Valid color change	Select Blue	Success: Background changes to blue	V1
TC061	Invalid - No color selected	Click Cancel	No change in background color	IV1
TC062	Invalid - Corrupted action	Invalid color picker	Error: Unable to change background color	IV2

10. Print Customer Balance

10.1 Inputs:

- **Account Number (txtNo):** Numeric, must exist in the database.

Conditions:

Field	Valid Partitions	Invalid Partitions	Invalid Boundaries	Valid Boundaries
Account Number	Numeric, exists in DB	Non-existent, non-numeric, empty	0, -1, abc	1, 12345

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
--------------	-------------	-------	------------------	------

TC063	Valid balance print	No: 100	Success: Account balance printed	V1
TC064	Invalid - Non-existent account	No: 999	Error: Account does not exist	IV1
TC065	Invalid - Non-numeric account number	No: abc	Error: Account number must be numeric	IV2
TC066	Invalid - Empty account number	No: `` (empty)	Error: Account number cannot be empty	IV3
TC067	Boundary - Minimum valid account	No: 1	Success: Account balance printed	VB1

11. Quit BankSystem

This functionality allows the user to exit the application. It's a simple action tied to the window close event.

11.1 Inputs:

- **Close Window Action (via File Menu or Window Menu)**

Conditions:

Field	Valid Partitions	Invalid Partitions
Exit Action	User selects "Quit" or closes the window	User cancels or presses the close button in error

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC068	Valid close via menu	Select Quit	Success: Application closes gracefully	V1
TC069	Valid close via window	Close the window	Success: Application closes gracefully	V2
TC070	Invalid close (canceled)	Close window and cancel	Error: Application does not close	IV1

12. Change Layout Style

12.1 Inputs:

- **Layout Style (UI) Change:** Change among available styles (e.g., Metal, Motif, Windows).

Conditions:

Field	Valid Partitions	Invalid Partitions
Layout Style	Metal, Motif, Windows	Invalid selections (null, system crash, etc.)

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC071	Valid style change to Metal	Select Metal	Success: UI changes to Metal style	V1
TC072	Valid style change to Windows	Select Windows	Success: UI changes to Windows style	V2
TC073	Invalid style change	Select invalid option	Error: No change to UI, invalid input	IV1

13. Apply Theme (e.g., Aqua, Green)

13.1 Inputs:

- **Theme Selection:** Select a theme from available options.

Conditions:

Field	Valid Partitions	Invalid Partitions
Theme Selection	Aqua, Green, etc.	Invalid theme input, null selection

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
--------------	-------------	-------	------------------	------

TC074	Valid theme selection (Aqua)	Select Aqua	Success: UI changes to Aqua theme	V1
TC075	Valid theme selection (Green)	Select Green	Success: UI changes to Green theme	V2
TC076	Invalid theme selection	Select invalid theme	Error: No change to UI, invalid input	IV1

14. Help Contents

14.1 Inputs:

- **Help Menu Selection:** Access the help content section of the system.

Conditions:

Field	Valid Partitions	Invalid Partitions
Help Access	User selects Help menu	User cancels or doesn't select help

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC077	Access Help Contents	Select Help Contents	Success: Help content displayed	V1
TC078	Invalid access to Help	No selection or cancel	Error: Help content not displayed	IV1

15. Help on Shortcuts

15.1 Inputs:

- **Shortcut Help Access:** View keyboard shortcuts used in the system.

Conditions:

Field	Valid Partitions	Invalid Partitions
Shortcuts Help	User accesses help	User cancels or doesn't access

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC079	Access Shortcut Help	Select Help Shortcuts	Success: Display keyboard shortcut information	V1
TC080	Invalid access to Shortcuts	No selection or cancel	Error: Shortcuts not displayed	IV1

16. About BankSystem

16.1 Inputs:

- **About Menu Selection:** View program info and credits.

Conditions:

Field	Valid Partitions	Invalid Partitions
About Info	User selects About menu	User cancels or doesn't select

Test Cases:

Test Case ID	Description	Input	Expected Outcome	Tags
TC081	Access About Information	Select About	Success: Program details and credits displayed	V1
TC082	Invalid access to About	No selection or cancel	Error: Program info not displayed	IV1

Conclusion: Black Box Testing for BankSystem

The **Black Box Testing** process for the **BankSystem** application allowed us to evaluate the system's functionalities from the user's perspective, ensuring that the application behaves as expected without focusing on the internal code. We focused on testing the system's input/output behavior, ensuring that every feature performs as intended under different conditions.

Here's a breakdown of our learnings and insights from the process:

1. **Functionality Validation:** We verified that all the system's core functionalities, such as account creation, deposit, withdrawal, searching, deleting, printing, and the various UI customizations, perform correctly. This also included menu-based and toolbar-based operations, ensuring smooth navigation and interaction.
2. **User Interface (UI) Testing:** Several test cases focused on ensuring that the user interface responds properly to user actions. We confirmed that the system handles input constraints correctly and provides appropriate feedback in case of invalid input.

Boundary Value Analysis (BVA) & Equivalence Class Partitioning (ECP):

- **BVA** helped us identify critical points of interaction such as minimum and maximum values, validating edge conditions like the lowest/highest allowed balance or deposit amounts.
 - **ECP** ensured we tested the application's response to both valid and invalid user inputs by categorizing input values into equivalence classes and testing them accordingly.
3. **Error Handling & Robustness:** We ensured the application gracefully handles incorrect inputs or user errors. This includes scenarios like empty fields, invalid customer ID, exceeding withdrawal amounts, etc. The application should prompt users with meaningful error messages, enhancing the user experience.
 4. **Cross-functional Test Coverage:** We confirmed that all menu options, toolbars, and context-based actions such as right-click menus work as intended. Special focus was given to the dynamic behaviors like window opening/closing, theme changes, and UI transitions.

Key Learnings:

- **Thorough Test Case Design:** The importance of designing comprehensive test cases that cover both expected and edge-case scenarios was reinforced. This ensured that no critical functionality was missed and that the application is fully validated from a user-centric perspective.
- **Error Handling and User Guidance:** A key learning from testing was the significance of having robust error handling. The system should not just handle

failures silently but provide useful feedback that helps users understand what went wrong and how to correct it.

- **Consistency in UI:** The importance of maintaining consistent behaviors across different user interactions, such as menu selections, toolbar actions, and pop-up menus, was highlighted. Testing showed that the UI must be responsive and intuitive, making it easy for users to navigate.
- **Efficiency in Testing:** The process of categorizing input conditions into equivalence classes and focusing on boundary values helped optimize the testing process. This approach saved time and effort while ensuring comprehensive coverage.

What We Learned as a Team:

1. **Collaboration:** Working as a team allowed us to break down complex features into manageable tasks. We were able to delegate and tackle different functionalities simultaneously, ensuring all aspects of the system were covered.
2. **Real-World Application:** Testing real-world scenarios, such as managing large databases or performing transactions, gave us practical insights into how users will interact with the system in production. We learned how to anticipate user behavior and handle errors in a way that is seamless for the end-user.
3. **Attention to Detail:** The testing process sharpened our attention to detail. We needed to ensure that small issues, like validating input constraints or ensuring buttons are aligned and clickable, were not overlooked. Each test case presented an opportunity to refine the application further.
4. **Tools and Techniques:** We gained valuable hands-on experience with tools and methodologies like **BVA**, **ECP**, and **Test Case Design**. Understanding how to efficiently execute these methodologies within the scope of Black Box Testing was invaluable.

Final Thoughts:

The **Black Box Testing** exercise has not only helped us validate the core functionalities of the **BankSystem** application but also strengthened our understanding of how to design and execute comprehensive testing strategies. The process was essential for ensuring that the application is both user-friendly and reliable, providing a seamless experience for its users while meeting functional requirements.

This experience has been instrumental in improving our skills in systematic testing, error handling, and ensuring that applications behave as intended from a user perspective. It has also helped us understand the importance of providing clear user guidance, intuitive design, and maintaining consistency across various features of the system.