				Softwar	re bug report								
niect				Cortwar	c bug report								
ame:	Banking App Prototype	9											
ug ID	Title/Summary	Issue type	Classification	Environment	Step-by-step	Expected result	Actual result	Frequency	Status	Severity	Notes/Comments	Resolution	
ique bug name]	[Give a short and clear description of the issue]	Issue type	Classification	[Describe the exact conditions in which the bug appeared. E.g., OS, OS version, app type, app version, etc.]	[List the exact steps you took before the issue manifested]	[Describe what was supposed to happen]	[Describe what actually happened]	Frequency	To do	Severity	[Additional notes or comments regarding the issue]	[Describe how the issue was resolved for future reference]	
UG-001	Incorrect initial balance set for new accounts	Bug	Syntax	OS: Windows 11 Pro; IDE: IntelliJ IDEA Community Edition 2024.11: SDK: Eclipse Ternurin 19.0.2	Run application 2. Login with cheat login 3. Choose option 1 (Add account) 4. Select account type 1 (SBA) 5. Enter name (just hit enter) 6. Check balance (option 5) 7. Repeat for type 2 Community and 3 Client	Initial balance for all newly created accounts should be 0.0.	On account creation of each type the starting blances were not 0 as expected. SBA is £1, Community £2, Client £3	Always	Done		Caused by passing accountType int to BankAccount constructors balance parameter in Main.java. Affects MainTest testDeposit & MainTest testCheckBalance failures. I rated critical due to the fact there is a risk of major financial loss to the business as the cumalitive affect of adding even £1 could be massive if say 1 million users withdrew the 'free' £1.	Refactored BankAccount and subclasses to use BigDecimal for balance. Modified AccountService.createAccount to initialise balance with BigDecimal.ZERO instead of passing the account type integer. This ensures all new accounts start with zero balance, resolving the incorrect initial balance issue.	
IG-002	Small Business Account withdrawal does not update balance	Bug	Logic	OS: Windows 11 Pro: IDE: IntelliJ IDEA Community Edition 2024.1.f. SDK: Eclipse Temurin 19.0.2	Create SBA (Account 1). 2. Deposit £100 into Account 3. Verify balance is £101 (due to BUG-001) 4. Withdraw £50 from Account 5. Check balance of Account again	Succesful withdrawal should print to console and balance should decrease by £50 to £51.	On withdrawing from an SBA withdrawal prints "Withdrawal successfull" but the balance remains £101.	Always	Done	Major	The withdraw method in SmallBusinessAccount calls an empty private void setBalance(double d) () method, so the balance field is never updated. Breaks core functionality (REQ-007).	I removed the empty setBalance method from the original SBA class and implemented withdraw logic in OverdraftManagement class, ensuring super-setBalance is called correctly to update the balance field after withdrawal. The withdraw method now correctly recognises standard withdrawals and overdraft usage.	
JG-003	Overdraft facility not applied during withdrawals	Bug	Logic	OS: Windows 11 Pro; IDE: IntelliJ IDEA Community Edition 2024.11; SDK: Eclipse Temurin 19.0.2	Create an account of any type 2. Attempt to withdraw or transfer an amount exceeding balance but within the overdraft limit 3. Repeat for other account types	Withdrawal should be successful as there's an overdraft on each account. Balance should become £0 or negative and overdraft use should be tracked.	Withdrawal and transfers should be possible within each account types overdraft limit but it is not possible to use the overdraft limit on any account.	Always	Done	Major	Fails REQ-005. Community/Client accounts need withdraw overridden to handle overdraft. SBA withdraw needs fixing (BUG-002) and potentially refinement of overdraft logic.	Implemented overdraft logic within the OverdraftManagement.withdraw method. This method checks available funds balance + remaining overdraft before allowing withdrawal. Overdraft limits are specific to each account type.	
UG-004	System allows creation of multiple accounts of the same type	Bug	Logic	OS: Windows 11 Pro: IDE: IntelliJ IDEA Community Edition 2024.11: SDK: Eclipse Temurin 19.0.2	Create an SBA account 2. Attempt to create another SBA account 3. Repeat with other account types	For duplicate account creation to be prevented.	System successfully creates the second SBA account without any warning or error.	Always	Done	Critical	Violates REQ-002. Main java account creation logic lacks a check against existing accounts in the accounts list for the type being added.	Added AccountService.accountTypeExistsForUser method. Modified MenucreateBankCount to call this method and prevent the ability to create duplicate account types. The seventy is rated ortical due to the first think of the second types of the second that the second highly vulnerable to automation or coordinated efforts, potentially leading to significant financial losses for the business.	
UG-005	User not prompted to create an account if none exist upon login	Missing feature	Logic	OS: Windows 11 Pro: IDE: IntelliJ IDEA Community Edition 2024.11: SDK: Eclipse Temurin 19.0.2	Ensure no user accounts exist when rerunning 2. Login successfully	To be prompted to create a bank account with the types available after logging in with a newly made account		Always	Done	Minor	Fails REQ-003. Requires adding a check if (atm.accounts. isEmpty()) after successful login in Main,java and implementing the prompt logic.	Implemented Menu.checkAndPromptForBankAccountCreation method to be called after logic in MenurunMainMenu. This imethod checks if accountService.getAccountsForUser checks if the user current has any accounts and if not then it prompts the account creation.	
JG-006	Two signatory functionality missing	Missing feature	Logic	05: Windows 11 Pro; IDE: IntelliJ IDEA Community Edition 2024.11; SDK: Eclipse Temurin 19.0.2	N/A (Functionality cannot be invoked as it doesn't exist). Attempting to conceptualize how it should work reveals no implementation	Accounts requiring two signatories should enforce this rule for specific actions such as withdrawals, transfers.	The TwoSignatoriesRestriction method in BankAccount returns false and is never called. No logic exists anywhere to implement or check for this requirement.	Always	Done	Critical	Fails REO-004. This is a significant feature gap requiring design (how are signatories stored/checked?) and implementation across relevant methods.	Implemented requiresSignatoryAuth() method for the Community account as per the requirements set out by the PO and BA. Added checks in Menu,withdraw, Menu.transfer, and Menu.closeBankAccount to prevent actions on signatory required accounts.	
	BankAccount.transfer method always returns false	Bug	Logic	OS: Windows 11 Pro; IDE: IntelliJ IDEA Community Edition 2024.11; SDK: Eclipse Temurin 19.0.2	Create two accounts (Acc1, Acc2). 2. Deposit funds into Acc1, 3. Perform a transfer from Acc1 to Acc2 (ensure sufficient funds). 4. Observe the console output (shows success) and ideally check the return value programmatically (requires test modification or debugging).	The transfer method should return true when the transfer is successful.	The method returns false even when the transfer completes successfully and balances are updated correctly. The console message correctly indicates success, but the method's return value is wrong.	Always	Done	Minor	The return false; statement at the end of the BankAccount. transfer method is incorrect. It should return true after successfully updating balances. Doesn't break the transfer itself but provides incorrect programmatic feedback. Affects REQ-007 slightly (function exists but flawed).	Refactored BankAccount.transferTo method. It now correctly calls the withdraw method which returns true/false. The transferTo method itself now returns true only if the withdrawal was successful, ensuring correct feedback.	
UG-008	REQ001 Manage account functionality is limited	Suggestion	UI/Usability	OS: Windows 11 Pro; IDE: IntelliJ IDEA Community Edition 2024.11; SDK: Eclipse Temurin 19.0.2	Login 2. Review menu options to see if account managment is implemeted	To be able to manage an account after creation, this should include basic applications such as ability to update username and password and account deletion.	Transfer, Check Balance, Exit are provided. No option to list all accounts or perform account	Always	Done	Minor	Current code base only covers basic transactions. Action: Clarify the requirment with PO/BA. If more features are required, add to code base.	Added "View My Bank Accounts" (option 6) and "Account Settings" (option 7, leading to a submenu with Change Username/Password, Close Account) to MenudisplayMainMenu and implemented corresponding methods (viewMyAccounts, runManageAccountMenu) to provide broader account management capabilities, addressing the ambuiguities of REQ-001.	