SUCCESS CASE DIAGRAM MariaDB (Data Store) UserManagementEntry (Application Layer) UserManagementManager (Business Layer) ILoggingService (Service Layer) UserManagementDataAccess (Data Access Layer) UserManagementService (Service Layer) SingleOperationRequest(CREATE : String) : String userManagementManager = new UserManagementManager() return UserManagementManager createAccountManager.CreateAccountInput(accountInfo : Map<String, String>) : Boolean | IsValidRequest(request : String[]) : Boolean userManagementService = new UserManagementService() UserManagementService() return UserManagementService userManagementService.CreateAccountRequest(accountInfo : Map<String, String>) : Boolean userManagementDataAccess = new UserManagementDataAccess() return UserManagementDataAccess user Management Data Access. Is Account Active (username: String): BooleanEstablishMariaDBConnection() : Boolean Query Processing: SELECT username FROM accounts WHERE username SelectAccount(searchAccountDML : String) : Boolean LIKE usernameInput False : Boolean False : Boolean Query Processing: INSERT accounts userManagementDataAccess.CreateAccount(accountInfo : Map<String, String>) : Boolean InsertAccount(createAccountDML : String) : Boolean VALUES (typeInput, usernameInput, passwordInput, emailInput, ableInput) True : Boolean True : Boolean userManagementLoggingService = new UserManagementLoggingService() UserManagementLoggingService() return UserManagementLoggingService LogRequestAsync(operation : String, isSuccess : Boolean) : Boolean Refer to the Logging Diagram True : Boolean "UM operation was successful" : String (Subsequent Logging Process Here)

UIMessage : String **<**-----

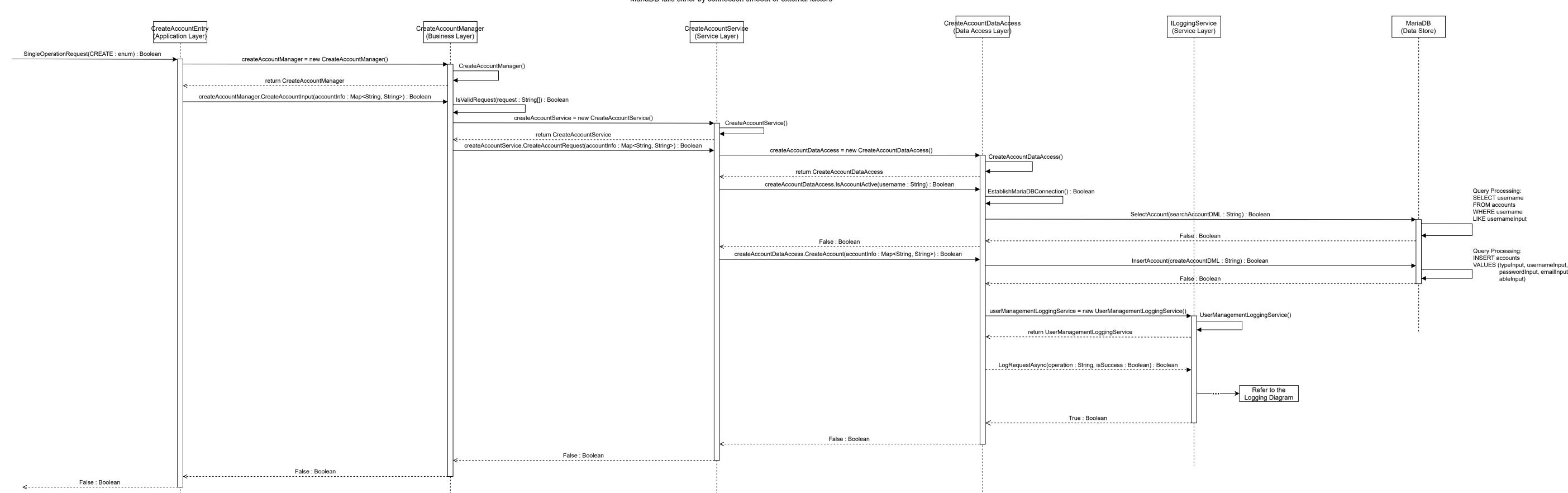
> **Success Scenario**  Single Request is made for the CREATE operation
>  New instance of CreateAccountManager is created Input account info 4. Check if request is valid, the following is checked: - does the input follow the appropriate regex - does the input contain all necessary info for the operation request 5. New instance of CreateAccountService is created 6. Call the CreateAccount Service to process operation with the validated input 7. New instance of CreateAccountDataAccess is created 8. Check if the account already exists using the username 9. Establish the MariaDB connection in DataAccess if not already opened Perform a SELECT query on the Accounts table in the database:
>  SELECT username
>  FROM accounts WHERE username LIKE usernameInput Returns FALSE if not found. 11. Return FALSE to the Service if no account already exists, and continue 12. Create the account with the account info 13. Perform a INSERT query on the Accounts table in the database: INSERT accounts VALUES (typeInput, usernameInput, passwordInput, emailInput, ableInput) Returns TRUE if account is successfully created. 14. Return TRUE to the Service if the account was successfully created 15. Return TRUE to the Manager if the request was successfully performed 16. Return TRUE to System and follow with the appropriate Logging for this operation

#### FAILURE CASE DIAGRAMS

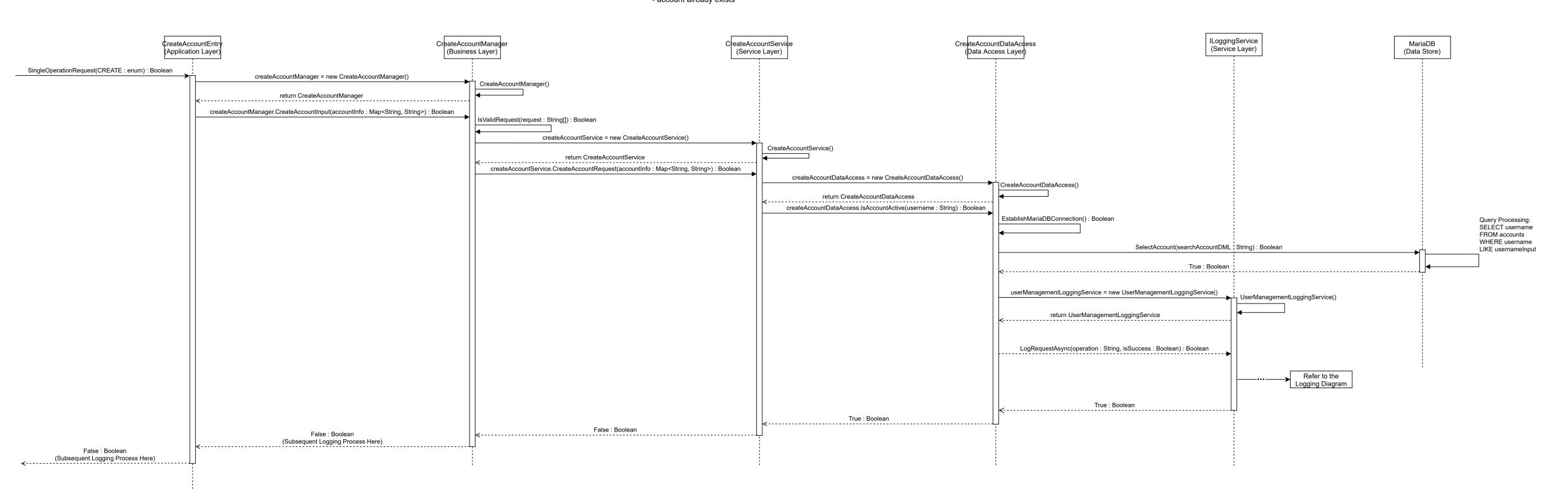
Fail Case 13

New Account is not Inserted into MariaDB

Reasons:
- incorrect DML syntax
- MariaDB fails either by connection timeout or external factors



### Fail Case 10a New Account already exists in MariaDB Reasons: - account already exists



## Fail Case 10b Could Not Process Query Reasons: - incorrect DML syntax - MariaDB fails either by connection timeout or external factors

CreateAccountE (Application Lay		countManager CreateAc ness Layer) (Servi	ce Layer) (Data A	ountDataAccess ccess Layer)	ILoggingService (Service Layer)	MariaDB (Data Store)
OperationRequest(CREATE : enum) : Boolean	createAccountManager = new CreateAccountManager()					
	return CreateAccountManager	CreateAccountManager()				
	createAccountManager.CreateAccountInput(accountInfo : Map <string, string="">) : Boolean</string,>	IsValidRequest(request : String[]) : Boolean				
		createAccountService = new CreateAccountService()	CreateAccountService()			
		return CreateAccountService  createAccountService.CreateAccountRequest(accountInfo: Map <string>): Boolean</string>	ŭ .			
			createAccountDataAccess = new CreateAccountDataAccess()	CreateAccountDataAccess()		
			return CreateAccountDataAccess  createAccountDataAccess.IsAccountActive(username : String) : Boolean			
				EstablishMariaDBConnection() : Boolean		Qi SE FF
				SelectAccount(searchAc	countDML String) : Boolean	₩ LII
				Erro	or : String	
				userManagementLoggingService = new UserManagementLoggingSer	vice() UserManagementLoggingService()	
				return UserManagementLoggingService		
				LogRequestAsync(operation : String, isSuccess : Boolean) : Boole	an	
					Refer to the Logging Diagram	
			France Office	True : Boolean		
	False : Boolean	False : Boolean	Error : String	. <u>L</u>		
False : Boolean	(Subsequent Logging Process Here)	- 니				

# Fail Case 9 MariaDB Connection could not be Established Reasons: - EstablishMariaDBConnection() returns False -- NOTE: continue to call this method until True if still no connection after 5 seconds, log the error and end the process

	countManager CreateActes Layer) CreateActes CountManager (Serv	countService ILoggir (Service	ngService CreateAcco ce Layer) (Data Ac	untDataAccess MariaDB (Data Store)
SingleOperationRequest(CREATE : enum) : Boolean  createAccountManager = new CreateAccountManager()  return CreateAccountManager	CreateAccountManager()			
createAccountManager.CreateAccountInput(accountInfo : Map <string, string="">) : Boolean</string,>	IsValidRequest(request : String[]) : Boolean			
	createAccountService = new CreateAccountService()  return CreateAccountService	CreateAccountService()		
	createAccountService.CreateAccountRequest(accountInfo : Map <string, string="">) : Boolean</string,>	createAccountDataAccess = new CreateAccount return CreateAccountDataAccess	:	CreateAccountDataAccess()
		createAccountDataAccess.IsAccountActive(us		EstablishMariaDBConnection() : Boolean
		False : Boolean  userManagementLoggingService = new UserManagementLoggingService()	UserManagementLoggingService()	
		return UserManagementLoggingService		
		LogRequestAsync(operation : String, isSuccess : Boolean) : Boolean	Refer to the	
		True : Boolean	Logging Diagram	
False : Boolean	False : Boolean			
(Subsequent Logging Process Here)				

## Fail Case 4 Invalid Request Reasons: Inputs do not follow their respective regex Missing required input(s) or does not match the inputs required for request

CreateAccountEntry Create (Application Layer) (Bu	AccountManager ILco	oggingService ervice Layer)	CreateAccountService	CreateAccountDataAccess (Data Access Layer)	MariaDB (Data Store)
SingleOperationRequest(CREATE : enum) : Boolean	· · · · · · · · · · · · · · · · · · ·	ervice Layer)	(Service Layer)	(Data Access Layer)	(Data Store)
return CreateAccountManager  createAccountManager.CreateAccountInput(accountInfo : Map <string, string="">) : Boolean</string,>	IsValidRequest(request : String[]) : Boolean				
	userManagementLoggingService = new UserManagementLoggingService()  return UserManagementLoggingService	Oserivariagement.coggingService()			
	LogRequestAsync(operation : String, isSuccess : Boolean) : Boolean				
		Refer to the Logging Diagram			
False : Boolean	True : Boolean				
< <del>-</del>					