PHASE 3: Data Modelling & Relationships

Debug Log Analysis and Bug Fixing in Salesforce Apex

1. Objective of Phase 3

In this phase, we design the data structure needed for writing, testing, and debugging Apex code in a Salesforce DX environment. Although this project focuses mainly on Apex debugging, proper data modelling is essential because Apex code often works with objects, fields, and relationships.

The goal of this phase is to define the objects and fields required for the Apex class and test class that will be debugged using the Apex Replay Debugger.

2. Objects Used in the Project

Since the debugging module in this project uses a sample Apex class named **AccountService**, the primary Salesforce object used is:

a. Standard Object: Account

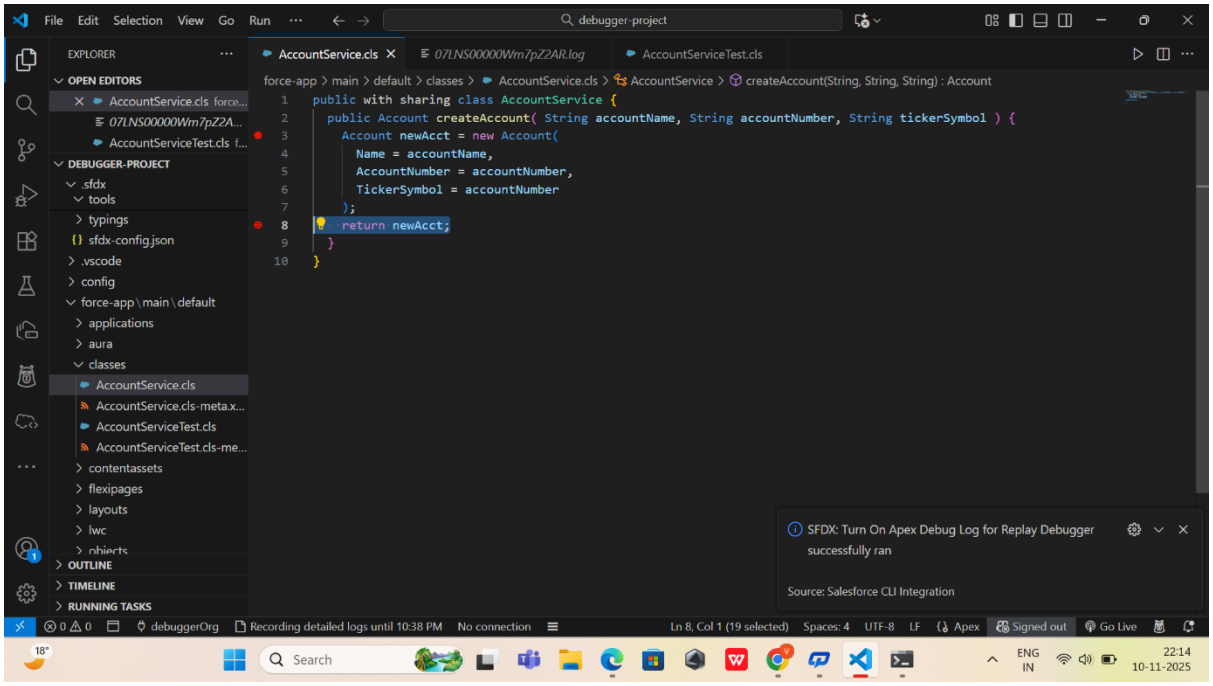
The Account object is used to:

- Create sample records in Apex
- Test field assignments
- Trigger checkpoints and breakpoints
- Validate field values during debugging

Key fields used:

- Name
- AccountNumber
- TickerSymbol
- Industry
- Phone

These fields are referenced in Apex code and examined through heap dumps during debugging.



3. Relationships

This project **does not require complex object relationships** because the focus is debugging Apex logic, not building a relational data model.

But if extended, the following relationships could be included:

a. Account → Contact (One-to-Many)

Useful for testing parent-child data handling in Apex.

b. Account → Opportunity (One-to-Many)

Useful for debugging business logic around sales processes.

c. Account → Case (One-to-Many)

Useful for debugging service-related flows.

Current Project Status:

➞ No relationship objects are mandatory.

➞ Only the Account object is used to demonstrate debugging concepts.

4. Data Modelling for Apex Test Class

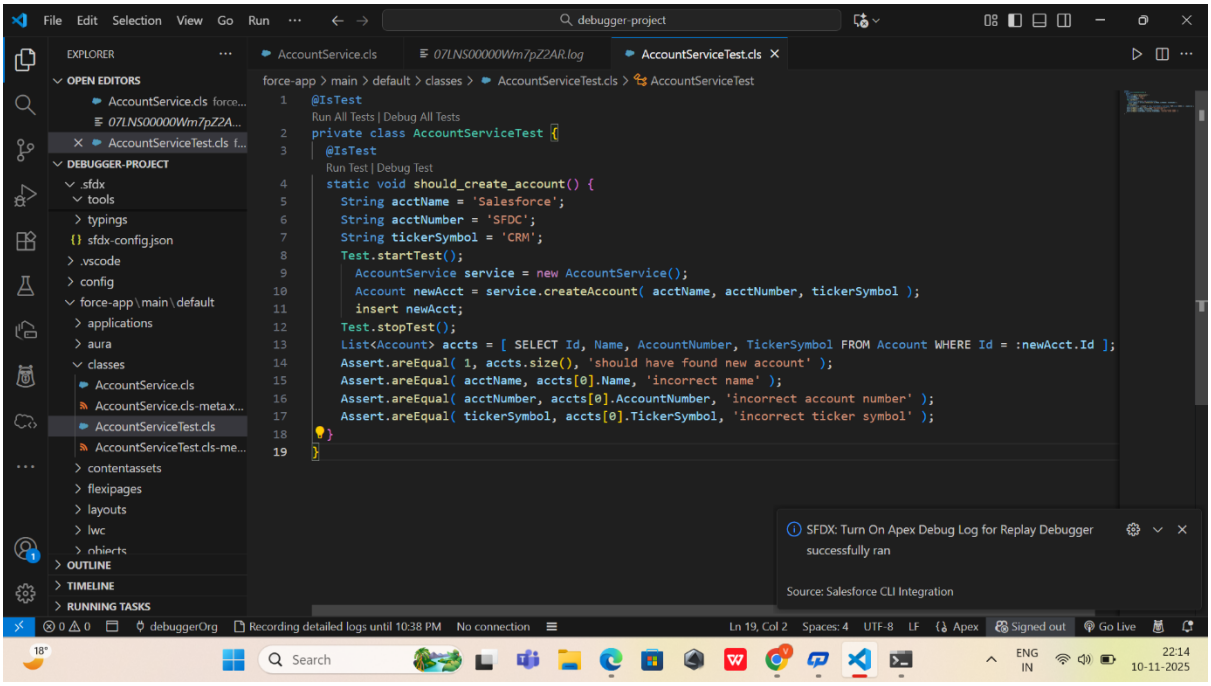
The **Apex test class (AccountServiceTest)** creates test data required for debugging:

What is created in test class?

- A sample Account record
- Field values to test (e.g., Name, AccountNumber, TickerSymbol)
- Assertions to check expected vs actual results

Why this matters for debugging?

- Test methods help generate debug logs
- Replay Debugger pauses on lines interacting with model fields
- Checkpoints capture heap dumps of Account object state



5. Outcome of Phase 3

At the end of this phase:

✓ Data model for debugging is clearly identified
✓ Standard “Account” object structured for use in Apex code
✓ Required fields are selected for debugging use cases
✓ No complex relationships required for the testing module
✓ Test data modelling ready for Apex test execution