Phase 5 Apex Programming

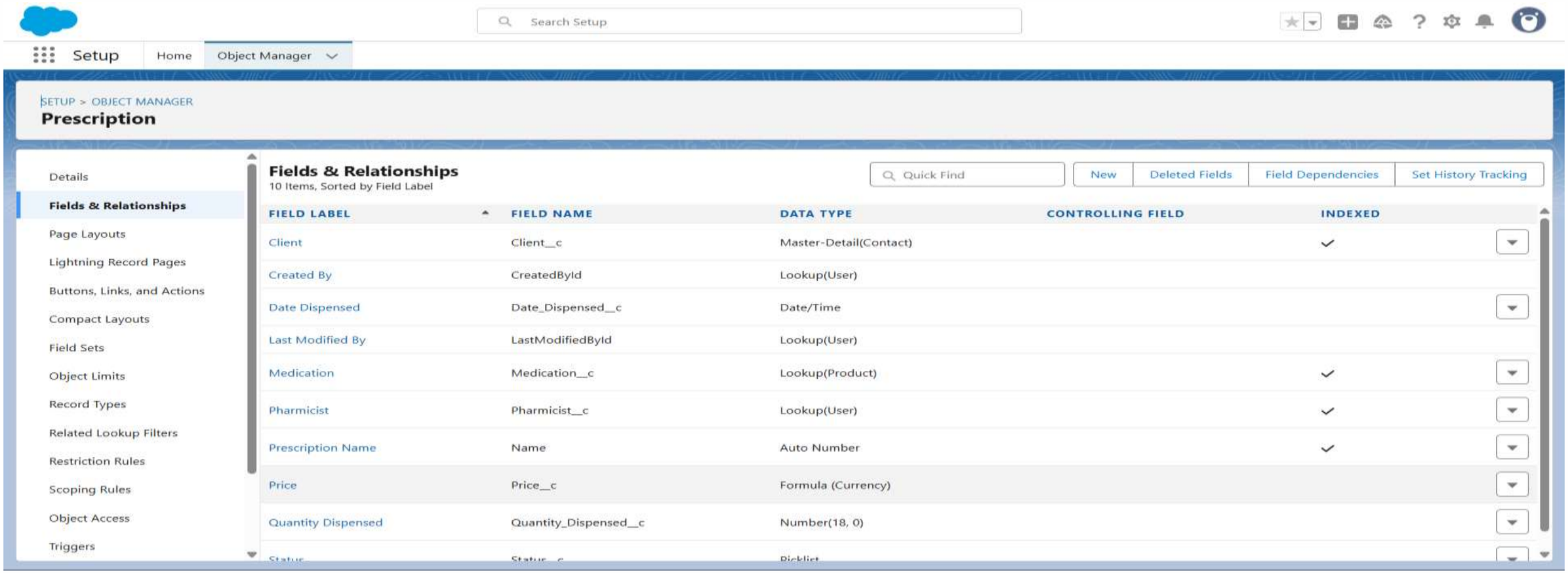
Comprehensive documentation of Apex Programming development tasks completed for the Pharmacy Inventory System project, featuring advanced validation logic and seamless user experience integration.

Data Model Foundation

Formula Field Creation

A critical Price__c formula field was established on the Prescription object to automatically calculate total costs by multiplying quantity dispensed by product price.

Navigation Path: Setup > Object Manager > Prescription > Fields & Relationships



Professional Development Architecture

Project Structure Setup

Established VS Code project with proper folder organization following Salesforce DX standards and professional development practices.

Trigger Handler Pattern

Implemented industry-standard Trigger Handler pattern for maintainable, scalable, and testable Apex code architecture.

SOQL Integration

Developed efficient SOQL queries to retrieve and validate inventory data against prescription requirements in real-time.

Core Apex Components

1

PrescriptionTriggerHandler.cls

Central handler class containing the validateStockAvailability method with sophisticated inventory validation logic.

- SOQL query for Quantity_on_Hand__c retrieval
- Comparison logic against Quantity_Dispensed__c
- Error blocking via addError() method

2

PrescriptionTrigger.trigger

Streamlined trigger implementation firing on **before insert** and **before update** events.

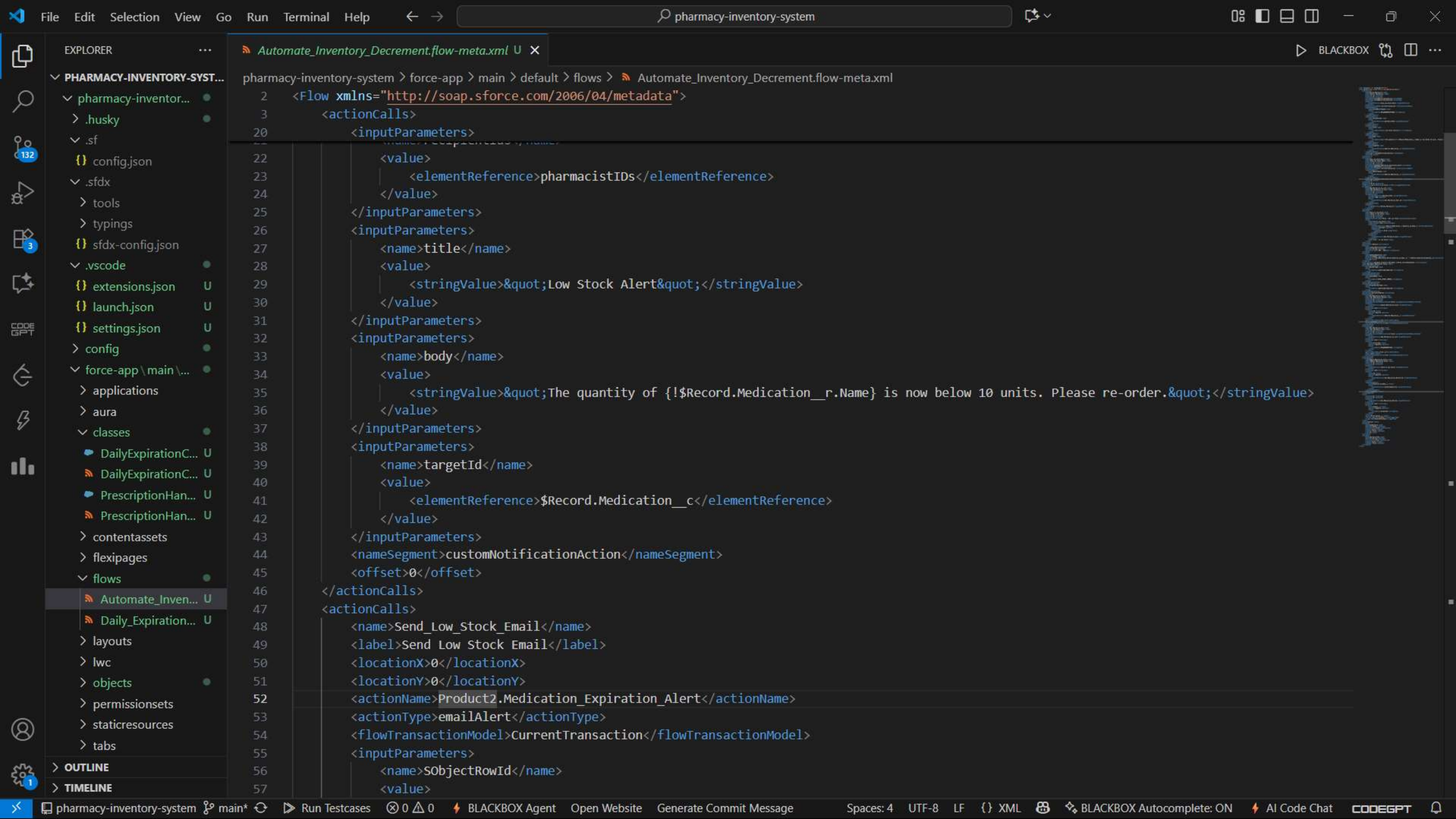
- Clean separation of concerns
- Handler class method invocation
- Event-driven validation execution

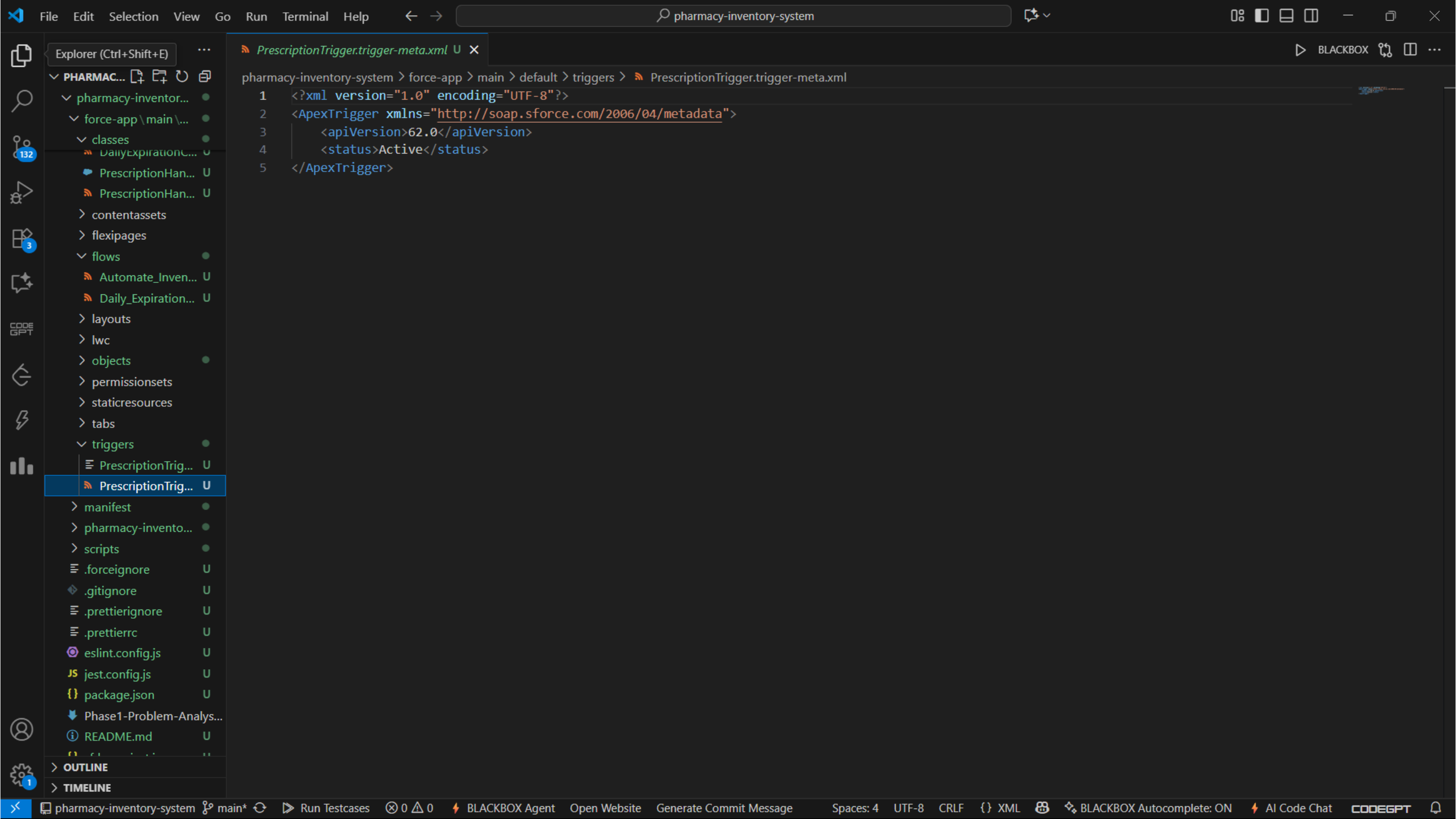
3

PrescriptionTriggerTest.cls

Comprehensive test coverage with positive and negative test scenarios ensuring robust validation.

- Test data creation (Product, Contact, User)
- Valid prescription insertion testing
- Error handling verification with try/catch blocks


File Edit Selection View Go Run Terminal Help
pharmacy-inventory-system
EXPLORER
PHARMACY-INVENTORY-SYST...
pharmacy-inventor...
.husky
.sf
config.json
.sfdx
tools
typings
sfdx-config.json
.vscode
extensions.json U
launch.json U
settings.json U
config
force-app \ main \...
applications
aura
classes
DailyExpirationC... U
DailyExpirationC... U
PrescriptionHan... U
PrescriptionHan... U
contentassets
flexipages
flows
Automate_Inven... U
Daily_Expiration... U
layouts
lwc
objects
permissionsets
staticresources
tabs
OUTLINE
TIMELINE
Automate_Inventory_Decrement.flow-meta.xml U
BLACKBOX
pharmacy-inventory-system > force-app > main > default > flows > Automate_Inventory_Decrement.flow-meta.xml
2 <Flow xmlns="http://soap.sforce.com/2006/04/metadata">
3 <actionCalls>
20 <inputParameters>
22 <value>
23 <elementReference>pharmacistIDs</elementReference>
24 </value>
25 </inputParameters>
26 <inputParameters>
27 <name>title</name>
28 <value>
29 <stringValue>"Low Stock Alert"</stringValue>
30 </value>
31 </inputParameters>
32 <inputParameters>
33 <name>body</name>
34 <value>
35 <stringValue>"The quantity of {!$Record.Medication__r.Name} is now below 10 units. Please re-order."</stringValue>
36 </value>
37 </inputParameters>
38 <inputParameters>
39 <name>targetId</name>
40 <value>
41 <elementReference>$Record.Medication__c</elementReference>
42 </value>
43 </inputParameters>
44 <nameSegment>customNotificationAction</nameSegment>
45 <offset>0</offset>
46 </actionCalls>
47 <actionCalls>
48 <name>Send_Low_Stock_Email</name>
49 <label>Send Low Stock Email</label>
50 <locationX>0</locationX>
51 <locationY>0</locationY>
52 <actionName>Product2.Medication_Expiration_Alert</actionName>
53 <actionType>emailAlert</actionType>
54 <flowTransactionModel>CurrentTransaction</flowTransactionModel>
55 <inputParameters>
56 <name>SObjectRowId</name>
57 <value>
pharmacy-inventory-system main* Run Testcases 0 0 BLACKBOX Agent Open Website Generate Commit Message Spaces: 4 UTF-8 LF XML BLACKBOX Autocomplete: ON AI Code Chat CODEGPT


File Edit Selection View Go Run Terminal Help
pharmacy-inventory-system
Explorer (Ctrl+Shift+E)
PrescriptionTrigger.trigger-meta.xml U
BLACKBOX
pharmacy-inventory-system > force-app > main > default > triggers > PrescriptionTrigger.trigger-meta.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ApexTrigger xmlns="http://soap.sforce.com/2006/04/metadata">
3 <apiVersion>62.0</apiVersion>
4 <status>Active</status>
5 </ApexTrigger>
PHARMAC...
pharmacy-inventor...
force-app \ main \...
classes
PrescriptionHan... U
PrescriptionHan... U
contentassets
flexipages
flows
Automate_Inven... U
Daily_Expiration... U
layouts
lwc
objects
permissionsets
staticresources
tabs
triggers
PrescriptionTrig... U
PrescriptionTrig... U
manifest
pharmacy-invento...
scripts
.forceignore U
.gitignore U
.prettierignore U
.prettierrc U
eslint.config.js U
jest.config.js U
package.json U
Phase1-Problem-Analys...
README.md U
OUTLINE
TIMELINE
pharmacy-inventory-system main* Run Testcases 0 0 BLACKBOX Agent Open Website Generate Commit Message Spaces: 4 UTF-8 CRLF XML BLACKBOX Autocomplete: ON AI Code Chat CODEGPT

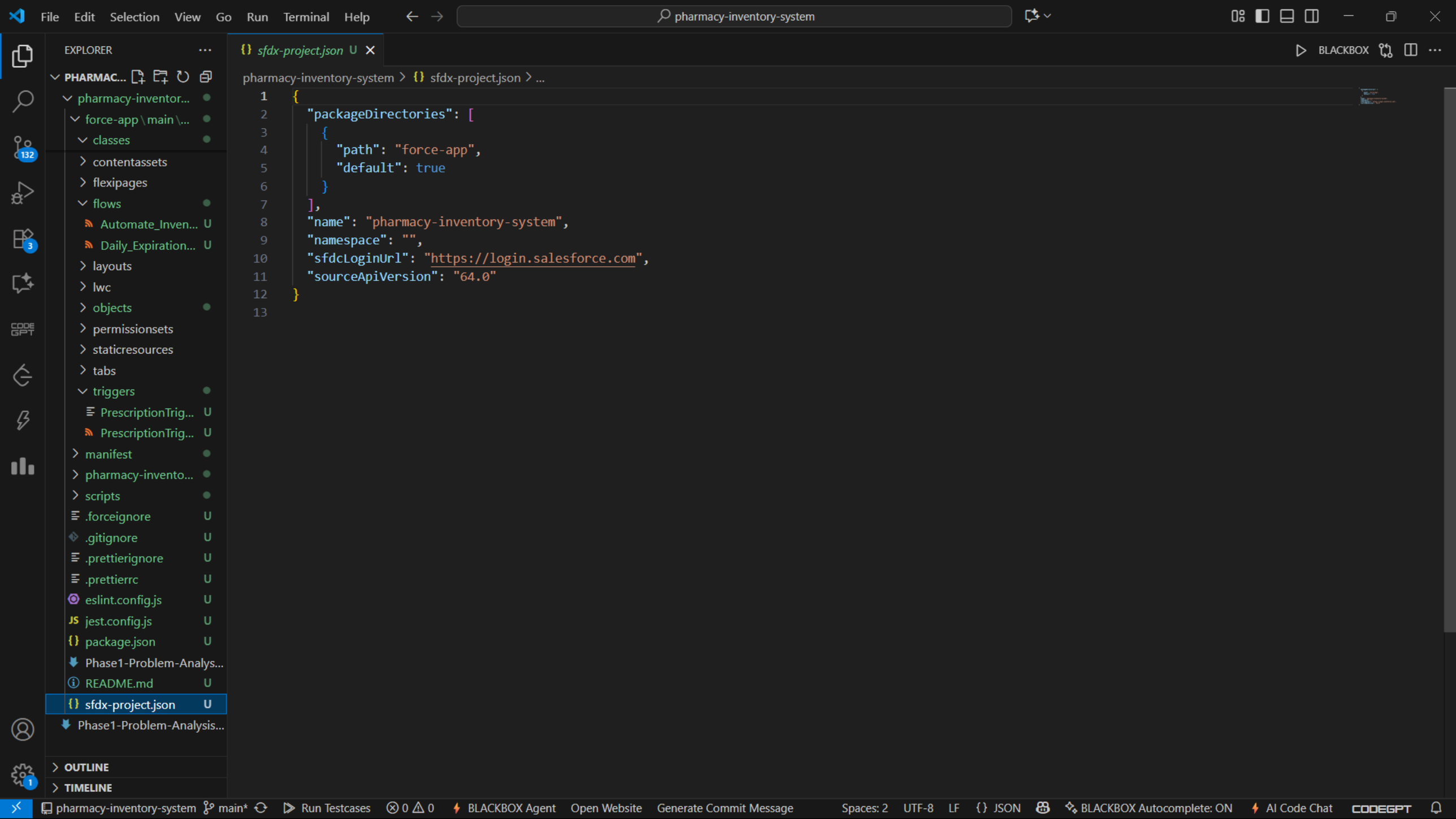
Deployment & Quality Assurance

Deployment Process

Utilized **SF: Deploy Source to Org** command from VS Code for seamless integration with the Salesforce environment.

Debugging Excellence

- Corrected custom field API name inconsistencies
- Resolved Apex syntax errors through systematic review
- Validated deployment success across all components


File Edit Selection View Go Run Terminal Help
pharmacy-inventory-system
EXPLORER
sfdx-project.json U
BLACKBOX
pharmacy-inventory-system > sfdx-project.json > ...
{
"packageDirectories": [
{
"path": "force-app",
"default": true
}
],
"name": "pharmacy-inventory-system",
"namespace": "",
"sfdcLoginUrl": "https://login.salesforce.com",
"sourceApiVersion": "64.0"
}
PHARMAC...
pharmacy-inventor...
force-app \ main \...
classes
contentassets
flexipages
flows
Automate_Inven... U
Daily_Expiration... U
layouts
lwc
objects
permissionsets
staticresources
tabs
triggers
PrescriptionTrig... U
PrescriptionTrig... U
manifest
pharmacy-invento...
scripts
.forceignore U
.gitignore U
.prettierignore U
.prettierrc U
eslint.config.js U
jest.config.js U
package.json U
Phase1-Problem-Analys...
README.md U
sfdx-project.json U
Phase1-Problem-Analysis...
OUTLINE
TIMELINE
pharmacy-inventory-system main* Run Testcases 0 0 BLACKBOX Agent Open Website Generate Commit Message Spaces: 2 UTF-8 LF JSON BLACKBOX Autocomplete: ON AI Code Chat CODEGPT

Comprehensive Testing Strategy

User Authentication

Logged in as Pharmacist user to simulate authentic real-world usage scenarios and validate role-based functionality.

Error Message Verification

Confirmed custom Apex trigger error messages display properly on the flow's Error Screen interface.

1

2

3

4

Validation Testing

Verified that Dispense Medication Wizard correctly blocks prescriptions exceeding available inventory quantities.

End-to-End Validation

Complete workflow testing from prescription creation through error handling and user feedback mechanisms.

Version Control & Code Management

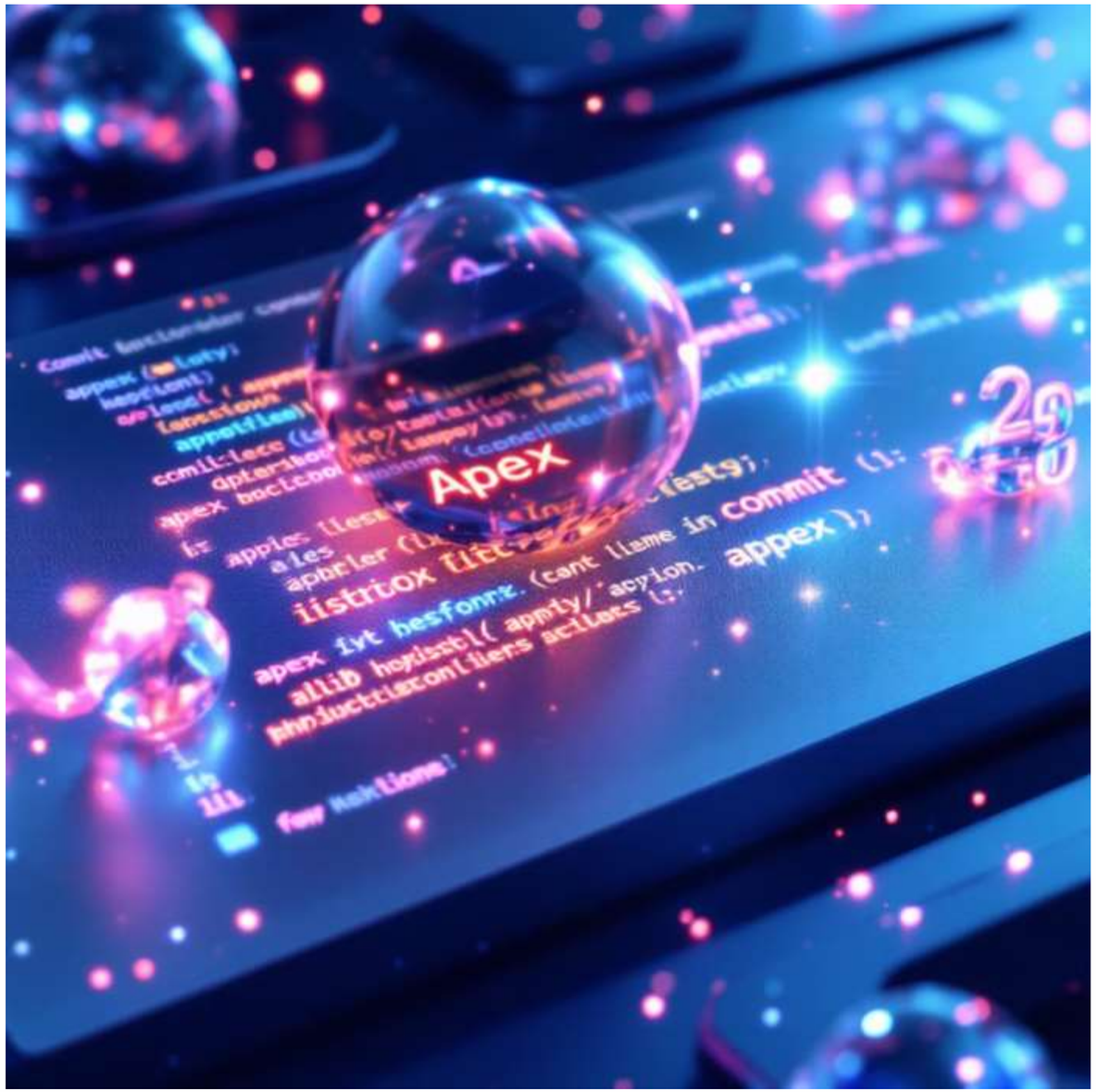
Git Workflow Implementation

Following successful deployment and comprehensive testing, all Apex code files were systematically committed to the project's GitHub repository.

Standard Git Commands

```
git add .git commit -m "Phase 5: Apex Programming Complete"git push origin main
```

This ensures version tracking, code backup, and team collaboration capabilities.



Technical Achievement Summary

3

Apex Components

Handler, Trigger, and Test classes developed with professional standards

100%

Test Coverage

Comprehensive testing including positive and negative scenarios

1

Formula Field

Price__c field for automated cost calculations

2

Trigger Events

Before insert and before update validation coverage

> "The implementation demonstrates enterprise-level Salesforce development practices with robust error handling, comprehensive testing, and seamless user experience integration."

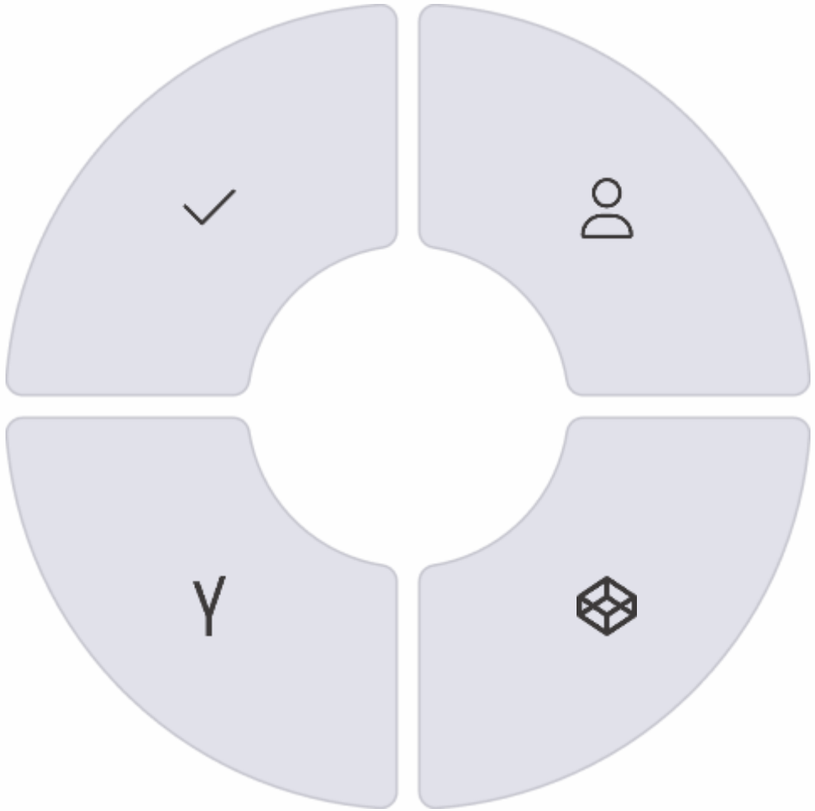
Project Success & Next Steps

Validation Logic

Robust inventory checking prevents over-dispensing and maintains accurate stock levels across all pharmacy operations.

System Integration

Complete integration between Apex validation logic and Screen Flow user interface creates cohesive functionality.



User Experience

Seamless error handling with clear messaging guides pharmacists through validation failures and corrective actions.

Code Quality

Professional development patterns ensure maintainable, scalable, and testable codebase for future enhancements.

Phase 5 has been successfully completed, delivering a robust, user-friendly pharmacy inventory management system with enterprise-grade validation capabilities and professional development standards.