Context-Aware Testing System for Financial Ecosystems

Team: AI_Challengers

Overview

- The **Context-Aware AI-Driven Testing System** is an advanced, multi-agent AI-powered framework designed to automate financial test generation, execution, fraud detection, and compliance validation. This system leverages Generative AI with LLM's, NLP, and ML models to create dynamic test cases from various financial data sources, ensuring optimal accuracy, security, and efficiency in financial transactions.
- Financial institutions operate within a highly complex and interconnected digital ecosystem that includes applications, APIs, databases, and cloud services. These systems require continuous validation to ensure compliance, security, and reliability. However, traditional testing methodologies rely heavily on static test cases, mock-based environments, and manual updates, leading to inefficiencies, high maintenance costs, and increased defect leakage into production.

Problem Statement

- Financial institutions operate within complex ecosystems comprising multiple interconnected systems, applications and API's. Testing these ecosystems often relies on static mock tools leading to inefficiencies and high maintenance cost. Testers struggle to create real-world financial transaction test scenarios covering fraud detection, regulatory compliance, and risk assessment..
- Frequent updates to AML (Anti-Money Laundering), GDPR, PCI-DSS, SOX, and other compliance frameworks make manual compliance testing inefficient. Banking operations, fund transfers, KYC validation, and credit risk assessments involve multiple transaction layers that require end-to-end testing. Rapid changes in pricing algorithms, fraud detection mechanisms, and AI-driven financial decision-making introduce new test case requirements. Manual compliance testing fails to detect violations proactively.
- Traditional test automation frameworks lack parallel execution capabilities and efficient resource allocation. Running full regression test suites requires significant computing resources, increasing costs. Manual intervention for debugging and failure analysis adds delays and overhead. Manual compliance testing fails to detect violations proactively.

Proposed Solution

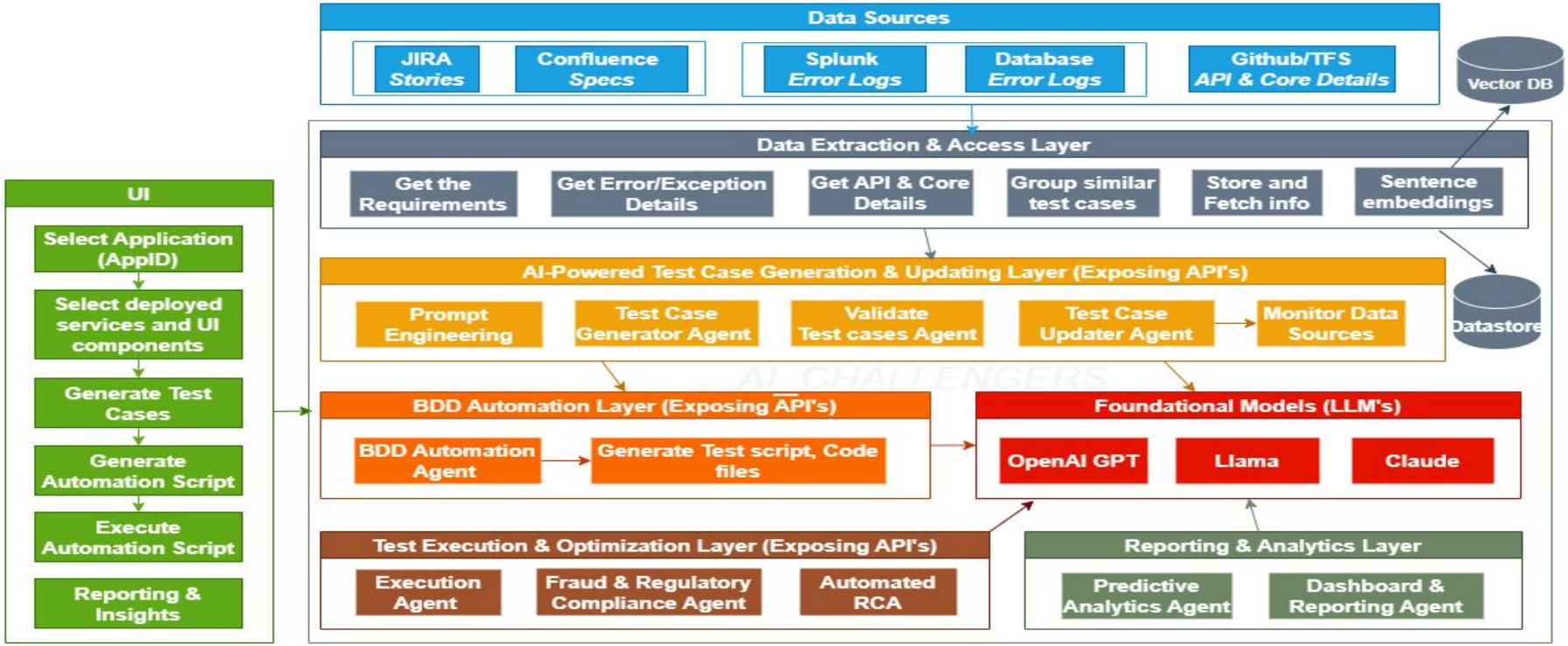
The proposed solution is a **multi-agent, AI-driven testing framework** that dynamically generates, validates, executes, and optimizes test cases based on evolving system contexts. It integrates various data sources, extracts relevant information, and leverages Generative AI for intelligent test generation, validation, execution, and reporting.

Layer	Details
UI	**User selects the Application and Services from UI. Triggers the Test case generation, automation script, execute tests and reports & analytics.**
Data Collection & Preprocessing	**Fetching the relevant information from multiple sources:** 1. JIRA & Confluence documents (Engineering specs, user stories). 2. Splunk & Banking Logs (Error logs, transaction failures, security alerts). 3. API Contracts & Financial Workflows (Payments, loan approvals, customer interactions). 4. Database Exception Logs (SQL errors, fraud alerts, failed transactions). **Preprocessing:** NLP-based Data Extraction - Extracts test scenarios from documents and logs. Vector Similarity Search - Detects duplicate test cases. Data Transformation - Converts unstructured text into structured test case format.
AI-Powered Test Case Generation & Validation	1. Test Case Generator Agent processes the extracted data using an LLM (e.g., ChatGPT, LLaMA) to create test cases. 2. Generated test cases are sent to the Validation Agent for optimization. 3. Validation Agent processes test cases and sends them to LLM for optimization. 4. LLM returns optimized test cases.

Proposed Solution

Layer	Details
Automatic Test Case Update	1. Continuous monitoring of JIRA, Confluence, logs, and API definitions detects any changes. 2. When a change is detected, the affected test cases are flagged for modification. 3. Updated data is re-processed through the Test Case Generator Agent and Validation Agent. 4. Modified test cases replace outdated ones, ensuring accuracy and relevance.
BDD Test Automation	1. UI triggers BDD Automation Agent to generate BDD test scripts. 2. BDD Automation Agent uses LLM to convert test cases into executable scripts.
Tests Execution and Optimization	1. Multi-threaded execution for faster test runs. 2. High-risk transactions (large-value transfers, cross-border payments) tested first. 3. AI will adjust the testing strategy dynamically based on previous execution results. 4. AI will detect why a test failed and suggest fixes. 5. AI will generate fraudulent transaction patterns to test financial fraud detection system and compliance validation
Reporting and Analytics	1. The Predictive Analytics Agent will analyze past test results, identify risks, and optimize future test cases.

Architecture Diagram



Tech Stack

Frontend (UI): ReactJS **Backend (APIs):** FastAPI **Data Processing:** MongoDB, FAISS
AI & ML: LangChain, AutoGen, Hugging Face, Llamma, OpenAI, Claude TensorFlow, NLP
Automation & Testing: Cucumber, SpecFlow

Benefits and Efficiency

Enhanced Test Accuracy & Coverage

- Automatically creates real-world financial test cases for transactions, fraud detection, compliance, and risk assessments. Dynamically updates test cases based on system changes, regulatory updates, and evolving financial risks. Tests payment processing, loan approvals, KYC validation, and fraud prevention across banking workflows.
- Automated AI-driven test generation: **70% faster**

Reduced Costs & Testing Effort

- AI self-updates test cases, reducing manual test creation and maintenance efforts. Eliminates redundant testing, saving costs on infrastructure, manual QA, and test execution time.
- AI-driven cost reduction: **20-50% savings**

Stronger Security, Fraud Detection & Risk Mitigation

- AI detects fraudulent activities like money laundering, insider fraud, and suspicious transactions. Ensures adherence to AML, PCI-DSS, GDPR, and banking regulations through continuous compliance testing. Prioritizes test scenarios based on financial risk impact and transaction sensitivity.
- AI-driven fraud detection: **50% more accurate,** Automated regulatory compliance testing: **Audit cycle 60% shorter**

Faster & Smarter Test Execution

- Runs functional and performance tests autonomously, reducing test cycle times.Identifies. system failures, security vulnerabilities, and performance bottlenecks before deployment.AI pinpoints issues and suggests fixes for failed test scenarios.
- Parallel AI-based execution: **40-60% faster**