Team Name: TestFailed

Team Members: Kiran Vennampelli, T S V P P Vishnu Kumar, Guraiah Mekala, Raghu Mylavarapu

Challenge: Context aware testing system for financial ecosystems

Tool Name: WGPRT-1 (WF Generative Payment Rail Testing Tool)

WGPRT Tool Architecture Document

Overview

The WGPRT Tool is a modular Streamlit-based application designed to streamline testing, fraud detection, and risk assessment processes. The application is divided into three primary modules:

1. **Payment Services Testing:** Provides tools to generate BDD test cases for various input formats. Understand the context and re-generate the changed test cases.
2. **Fraud Detection and Risk Scoring:** Implements tools for identifying fraudulent activities and assessing associated risks.
3. **Loan and Credit Risk Assessment:** Facilitates the evaluation of loan applications and credit risk analysis.

Each module is implemented as a separate Python file, dynamically loaded based on user interaction.

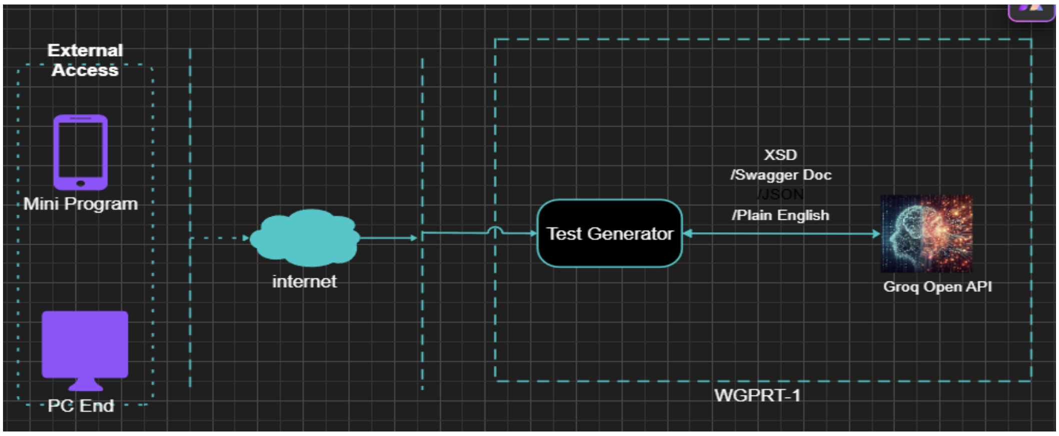
Modules and Features

1. Payment Services Testing

This module provides tools to generate BDD test cases for different input formats using "**Groq Open API technologies**". The following options are available:

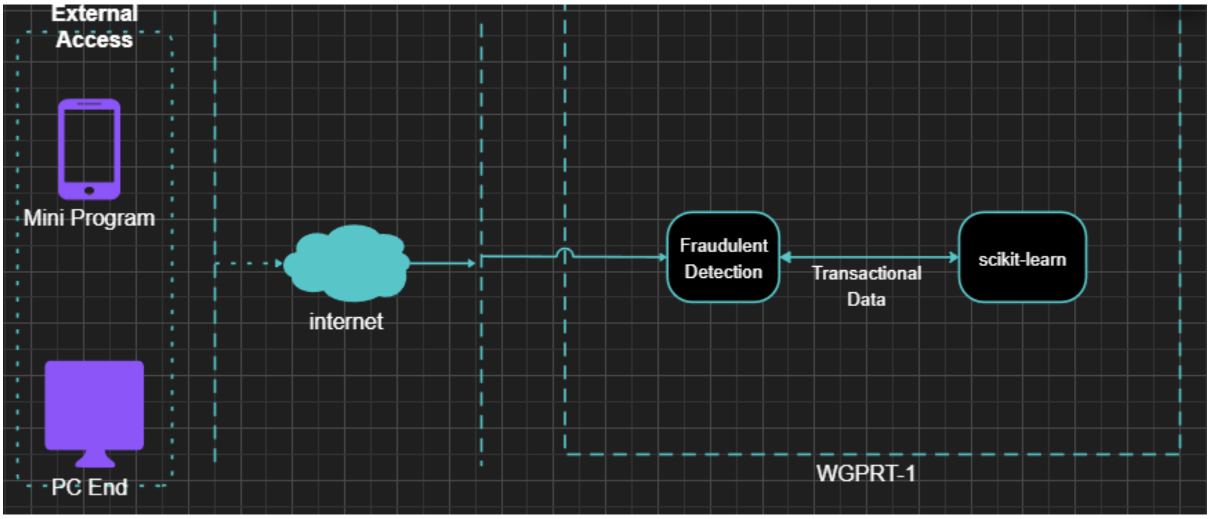
- **XSD Case Generator:** Generates BDD test cases based on uploaded XSD schemas. This extracts XSD schema and generate tests with all possible use cases.
- **API Test Case Generator:** Generates BDD test cases from Swagger JSON files.
- **Swagger URL Test Case Generator:** Generates BDD test cases directly from a Swagger API definition URL.
- **Jira Test Case Generator:** Generates BDD test cases from plain English requirements provided in a text file.

Architecture Diagram:



2. Fraud Detection and Risk Scoring

This module provides tools to detect fraudulent activities and assess associated risks. It allows users to upload datasets and run fraud detection algorithms.



3. Loan and Credit Risk Assessment

This module facilitates the evaluation of loan applications and credit risk analysis. It provides tools to upload loan application data and run risk assessment algorithms.

Navigation and Interaction

Main Navigation

The application uses a sidebar for navigation, allowing users to switch between the three main modules:

1. Test Case Generators
2. Fraudulent Detection
3. Loan Approval

Technologies Used

- **Frontend:** Streamlit (for building the user interface).
- **Backend:**
  - Python (for implementing the logic).
  - Groq Open API (for generating BDD test cases from plain English requirements).
  - Machine learning libraries (e.g., scikit-learn, pandas) for fraud detection and loan approval.

Future Enhancements

1. The Fraud Detection module can be enhanced to dynamically fetch fraudulent conditions or rules from different URLs. This feature will allow the system to stay updated with the latest fraud detection rules and adapt to changing patterns in fraudulent activities.
2. Real-Time Collaboration:
   - Enable multiple users to collaborate on test case generation and analysis in real-time.

Summary of APIs and Technologies

Category	Technology/API	Purpose
Frontend	Streamlit	Provides the user interface for navigation and interaction.
Backend	Python	Implements the core logic for all modules.
	Groq Python Library	Interacts with Groq Open API for AI-based BDD test case generation.
	JSON	Parses Swagger JSON files for API test case generation.
	OS Module	Manages file paths and dynamic file execution.
APIs	Groq Open API	Generates BDD test cases from plain English requirements and Swagger files.
	Swagger/OpenAPI	Parses API definitions for generating BDD test cases.
Machine Learning	scikit-learn	Implements fraud detection and credit risk assessment algorithms.
	pandas	Handles data preprocessing for fraud detection and loan approval.

This document provides a detailed overview of the architecture, modules, and technical implementation of the WGPRT Tool.