High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **SecurePay - Payment Security Simulator**. The platform enables finance professionals to simulate, test, and improve payment system security through real-world attack scenarios, secure authentication, and analytics. The design emphasizes modularity, hands-on learning, and robust, open-source technologies.

---

1. System Architecture Overview

Architecture Description:
SecurePay follows a modular, layered architecture with a React-based frontend, a Java Spring backend, and a PostgreSQL database. The system integrates security modules and analytics, supporting extensibility and secure, scalable operations.

Module	Role/Responsibility
Frontend (React)	User interface, scenario selection, results visualization
Backend API (Spring)	Business logic, scenario orchestration, authentication, REST API
Simulation Engine	Executes attack/defense scenarios, manages simulation state
Security Module	Implements secure authentication, OWASP best practices
Analytics Module	Collects, processes, and presents simulation results
Database (PostgreSQL)	Stores users, scenarios, results, and audit logs

---

2. Component Interactions

Interaction Step	Description
1. User logs in via frontend	Credentials sent securely to backend for authentication
2. User selects/launches a simulation scenario	Frontend requests scenario execution via backend API
3. Backend invokes Simulation Engine	Simulation Engine runs scenario, applies security modules
4. Simulation results processed by Analytics Module	Results and logs stored in database, analytics generated
5. Frontend fetches and displays analytics/results	User views outcomes, recommendations, and learning points

---

3. Data Flow Overview

Data Flow	Source	Destination	Purpose
User credentials	Frontend	Backend	Authentication
Scenario configuration/selection	Frontend	Backend	Initiate simulation
Simulation execution data	Backend	Simulation Engine	Run and monitor scenarios
Simulation results/logs	Simulation Engine	Analytics Module/Database	Analysis, storage
Analytics and feedback	Backend	Frontend	Display results and recommendations

---

4. Technology Stack

Layer/Component	Technology/Framework
Frontend	React, JavaScript, OWASP guidelines
Backend	Java, Spring Boot, OWASP Security
Database	PostgreSQL
Security	Spring Security, OWASP best practices
Analytics	Custom Java modules, SQL
Deployment	Docker, Kubernetes (optional), Linux

---

5. Scalability, Reliability & Security

- **Scalability:** Modular architecture supports horizontal scaling of backend and simulation engine. Stateless backend APIs enable load balancing.
- **Reliability:** Persistent storage of all simulation data and logs; robust error handling and monitoring.
- **Security:** End-to-end encryption, secure authentication (Spring Security), strict adherence to OWASP standards, audit logging, and regular security reviews.

End of Document