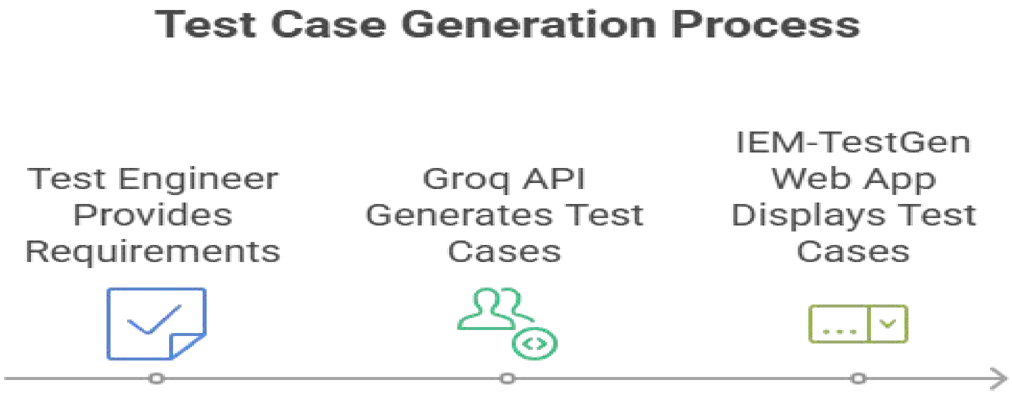
C4 Architecture Documentation

Project: IEM-TestGen – AI-Powered Test Case Generator **Date:** August 15, 2025 **Author:** [Manisha Kasireddy]

—---

1. System Context (Level 1)

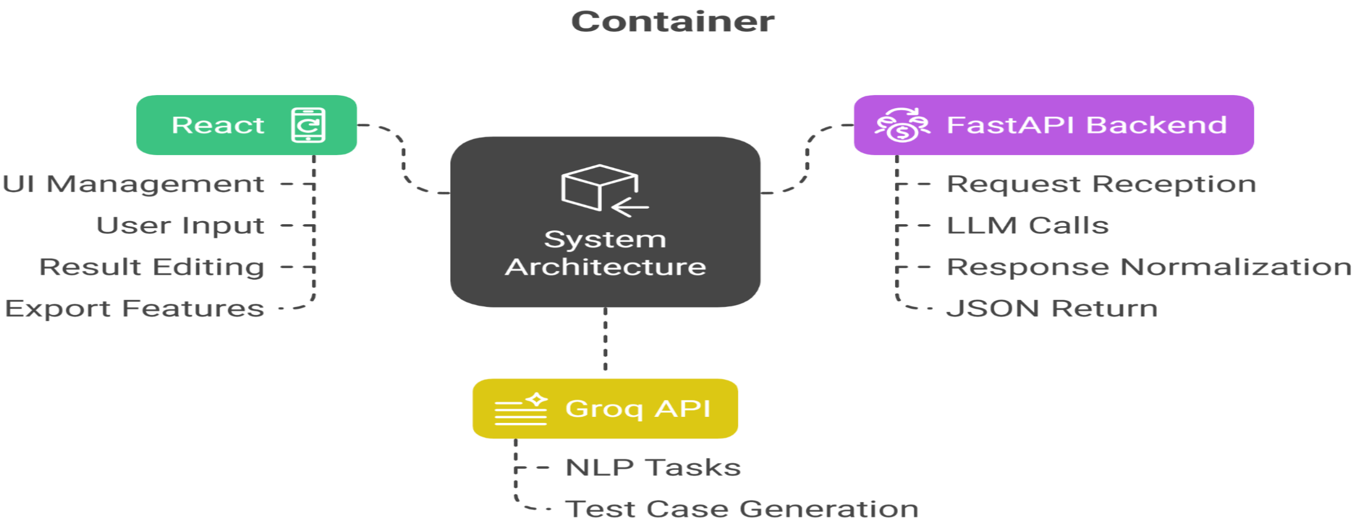
Purpose: Illustrating the relationship between the system and its external actors and dependencies.


Test Case Generation Process
Test Engineer Provides Requirements
Groq API Generates Test Cases
IEM-TestGen Web App Displays Test Cases

Description: - **Test Engineer**: Provides requirement descriptions. - **IEM-TestGen Web Application**: Generates and displays test cases. - **Groq API**: Processes natural language and returns structured test cases.

2. Container Diagram (Level 2)

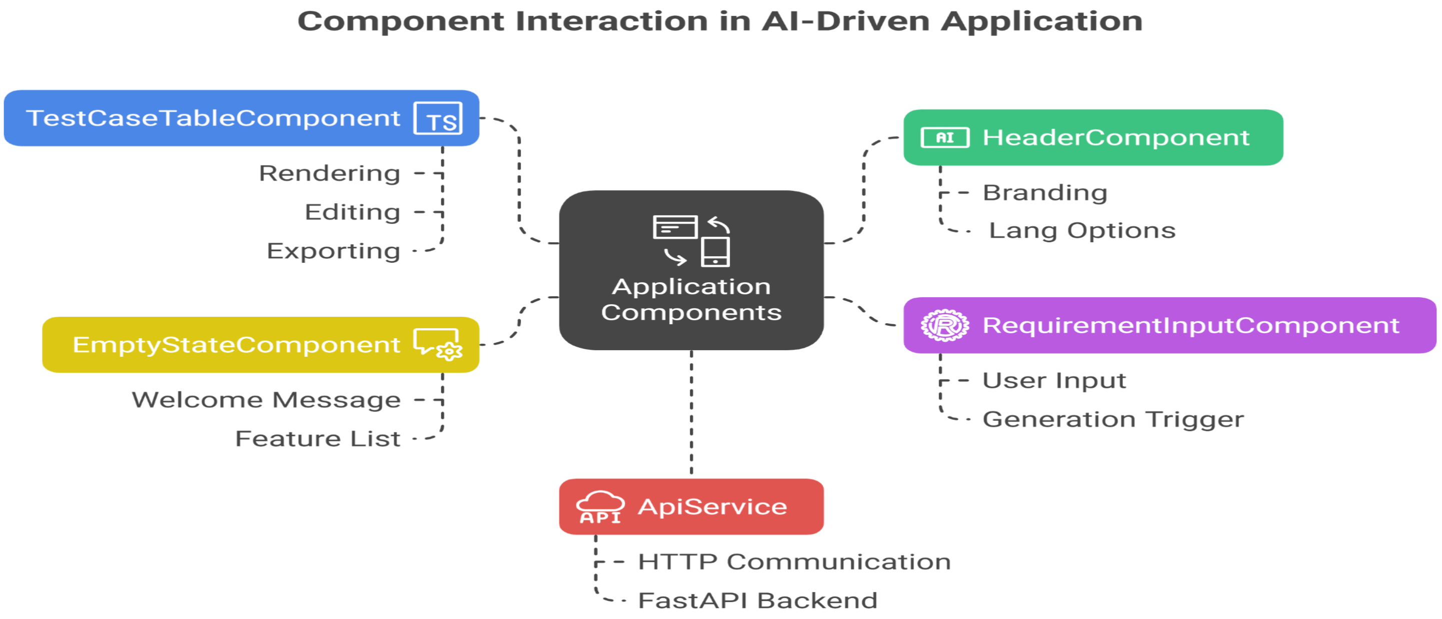
Purpose: Showing the high-level technology architecture, displaying containers that execute code or store data.



Containers: - **React** – Manages UI, user input, result editing, and export features. - **FastAPI Backend** – Receives requests, calls LLM, normalizes responses, and returns JSON. - **Groq API** – External service performing core NLP and test case generation.

3. Component Diagram (React Frontend) (Level 3)

Purpose: Detail internal components of the React container and their interactions.



Components Overview: - **HeaderComponent**: Displays branding and AI badge. - **RequirementInputComponent**: Captures user input and triggers generation. - **EmptyStateComponent**: Shows welcome message and feature list when no cases exist. -**TestCaseTableComponent**: Renders, edits, and exports generated test cases. **ApiService**: Handles HTTP communication with the FastAPI backend.

4. Key Architecture Decisions

Decision	Context	Rationale
React Vite	Rich, interactive UI	Component-based, extensive ecosystem, ease of state management
FastAPI	Lightweight, async API	High performance, automatic docs, simple integration with Python-based LLM clients
Groq GPT API	Core NLP processing	Proven model accuracy for text generation
Serverless Deployment	cost & maintenance	Render reduces infra overhead

5. Quality Attributes & Non-Functional Requirements

Usability: Intuitive interface with clear CTAs, responsive layout.
Performance: < 10s average response time for AI generation.
Scalability: Stateless design allows horizontal scaling.
Security: HTTPS, secure API key management, CORS policies.
Maintainability: Clear module boundaries, documented API, standardized coding styles.