Performance Testing for an Eyewear App

Introduction:

The client is a leading U.S. eyewear retailer offering fast, affordable, and convenient eye care services. They provide same-day eye exams and an extensive selection of eyeglasses and contact lenses. Through on-site labs, same-day service, and tele-optometry solutions, they deliver a modern, technology-driven experience that makes vision care more accessible.

With a national footprint spanning over 100 locations, the client has become a major player in the vision-care space, providing cost-effective eyewear, comprehensive eye exams, and customer-centric services. They are committed to making quality vision care both affordable and accessible, combining innovative optical technology with competitive pricing and a streamlined in-store experience. Operating as a high-volume, value-driven optical retailer, the client integrates telehealth capabilities, on-site labs, and a diverse product portfolio to deliver efficient, low-cost eye care and eyewear solutions.

Client Details:

Name: Confidential | **Industry:** eCommerce, Retail and Software | **Location:** USA

Technologies:

- **Front end:** HTML5, ReactJS (Next.js), TypeScript, Material UI, CSS,
- **App Layer:** Azure App Service
- **Business/API Layer:** API Management (Premium), Azure Functions
- **Backend Layer:** Azure Cosmos DB, Azure SQL Database
- **Storage Layer:** Azure Storage
- **Middleware:** Azure Service Bus
- **ETL:** Azure Data Factory
- **Tools:** Visual Studio 2022 Professional Edition, Azure Data Studio, ReSharper (Visual Studio Extension for .Net)
- **Storage:** Azure Repository

Technologies Used for Performance:

- **Tool Used for Scripting:** Apache JMeter
- **Tool Used for Load Testing:** Azure Load Testing Service
- **Tool Used for Root Cause Analysis:** Azure Application Insights

Performance Testing for an Eyewear App

Project Description:

The client is a leading eyewear retail brand renowned for delivering exceptional customer service and offering thousands of stylish, affordable frames. Their affiliated network of in-store doctors provides comprehensive eye examinations and detailed vision health assessments, delivering a seamless, full-service experience. An expert team guides customers throughout the selection and purchasing process. The client also offers same-day delivery of prescription glasses where feasible—setting a high standard in customer satisfaction and operational efficiency.

To maintain competitiveness in the rapidly evolving eyewear industry, the client engaged with us to conduct a comprehensive Performance Testing. Their primary objective was to assess the application's reliability, scalability, and stability under various load conditions. The client required a custom performance testing framework capable of running automated daily load tests, including performance regression to consistently monitor the application's health. A key requirement was the generation of actionable performance reports for each execution cycle to help stakeholders make informed decisions about application readiness for release.

Engagement Objectives:
Mindfire's Performance QA team collaborated closely with the client to gain a deep understanding of:

- Application architecture
- Key functional modules
- Current performance challenges and bottlenecks
- Expected workload patterns and usage volumes

The client sought an **end-to-end performance assessment** that would simulate realistic user behaviour across different modules and provide clear insights into system behaviour under varying load scenarios

Approach & Key Activities:

1. Requirements & Environment Setup

- The Performance QA team gathered detailed Non-Functional Requirements (NFRs) through continuous discussions with stakeholders, including user volume, workflow distribution, architecture details, and module-specific performance expectations.

Performance Testing for an Eyewear App

- A dedicated Performance Test Environment was created in collaboration with the DevOps team. This environment mirrored the production setup to ensure high test accuracy and reliability.

2. Tooling & Scripting
   - Apache JMeter, an open-source performance testing tool, was selected for scripting
   - Azure Load Testing service used for executing load scenarios
   - Customized test scripts were created to simulate end-to-end workflows and peak user activity across various application modules.

3. Monitoring & Root Cause Analysis
   - Integrated monitoring was implemented using Azure Application Insights, enabling deep-dive root cause analysis for every performance issue identified.
   - Key metrics included response times (Average, P50, P90, P95), error percentages, throughput, CPU utilization, memory consumption, and API-level performance.

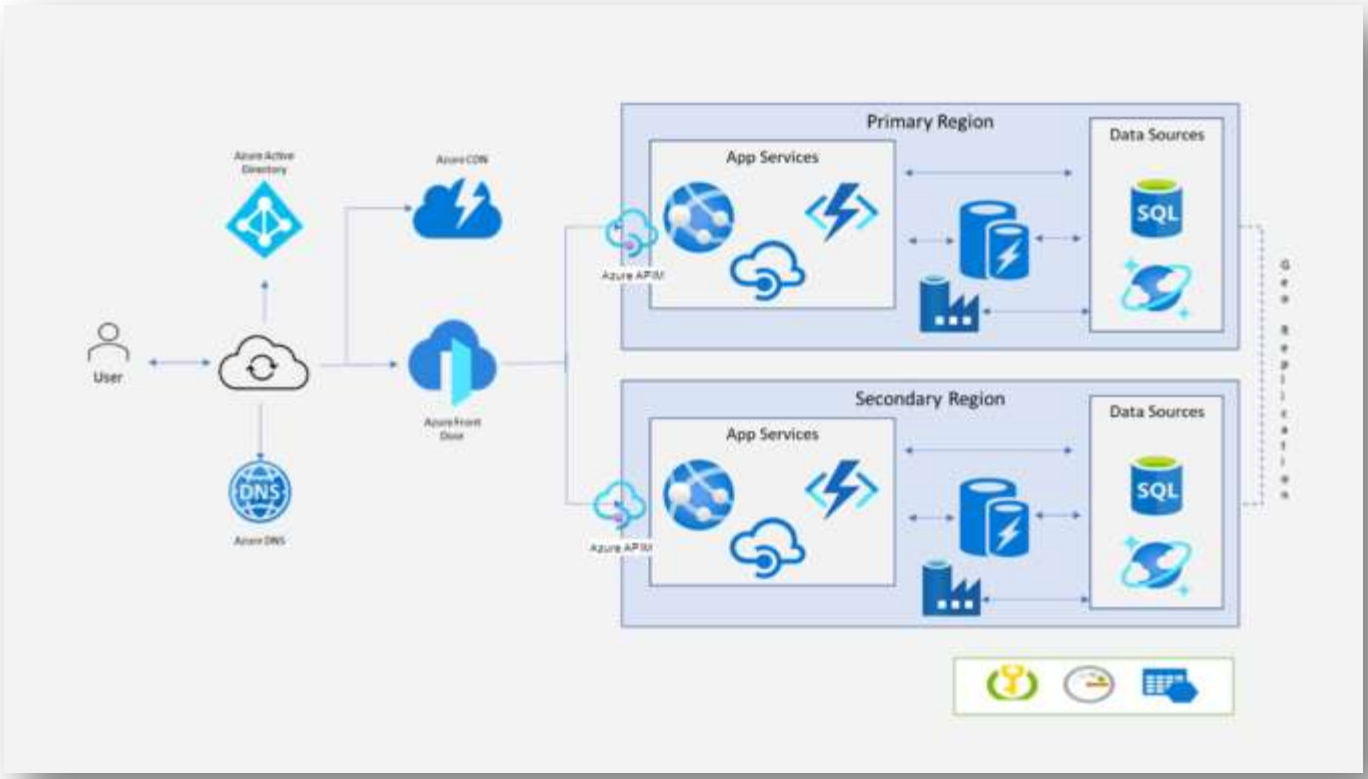
4. CI/CD Integration & Automation
   - The performance test suite was integrated into the client's Azure CI/CD Pipeline, enabling daily automated performance health checks.
   - Stakeholders gained the ability to trigger builds, execute performance tests, and access results directly through the pipeline dashboard.

5. Reporting & Issue Management
   - Detailed performance reports were generated after each execution cycle, providing insights into page-level and API-level performance trends.
   - All performance issues were logged as tickets, tracked rigorously, and validated with the development team until resolution.

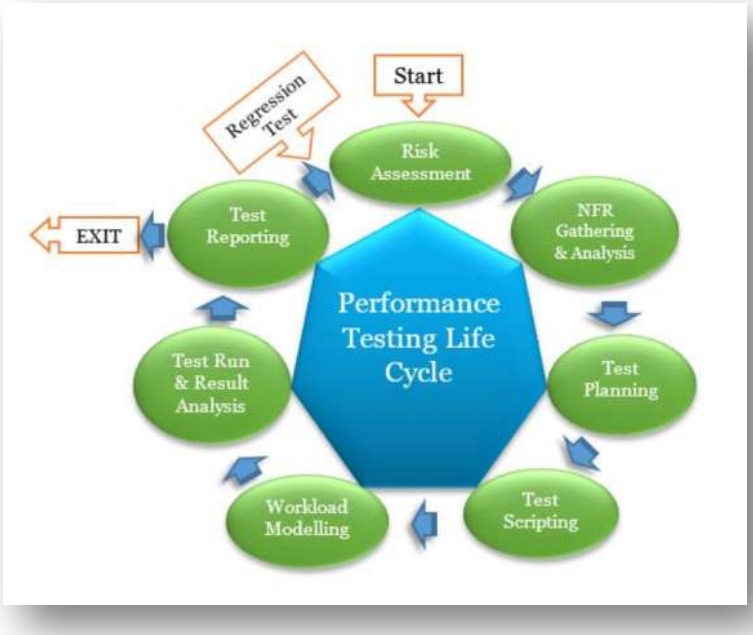
Methodology followed:
We followed Agile Scrum with Daily Standups and Sprint-based deliveries.

Performance Testing for an Eyewear App

Application Architecture Design:



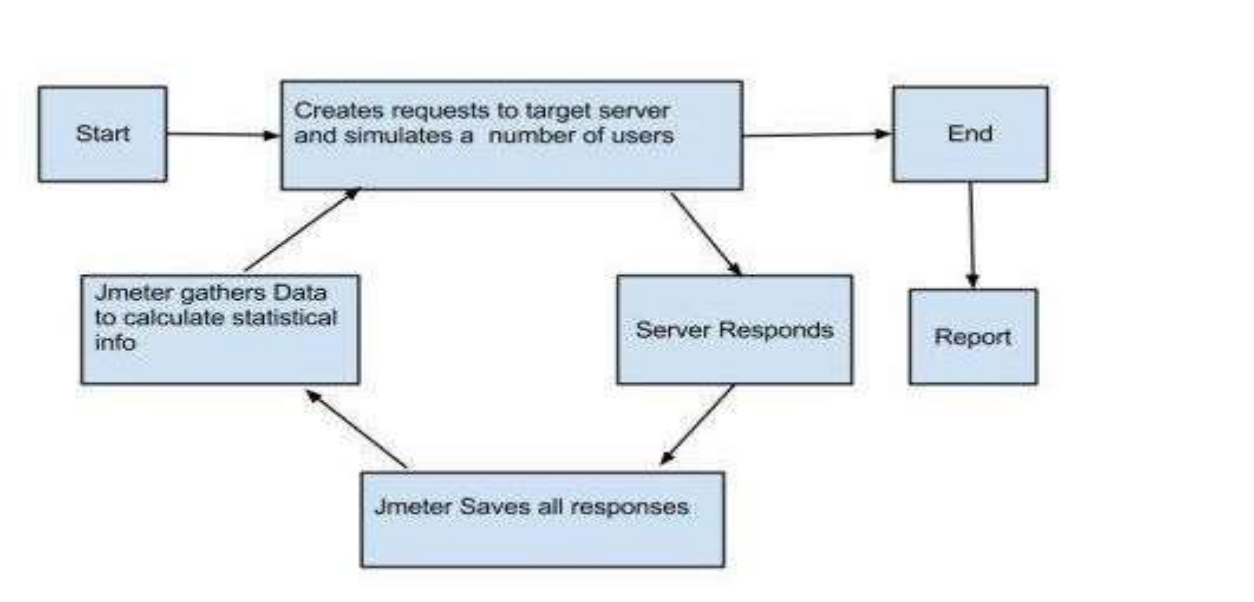
Workflow: Performance Testing



Performance Testing for an Eyewear App

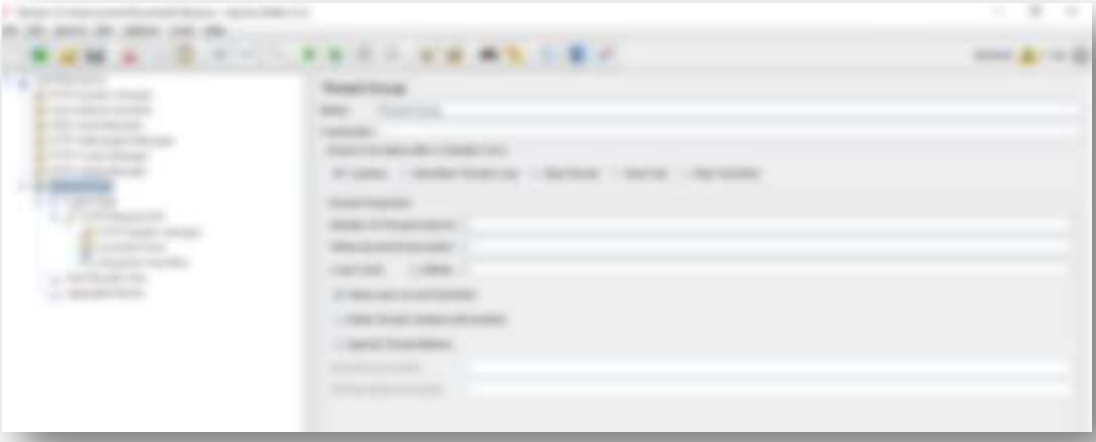
Apache JMeter Workflow:

JMeter simulates a group of users sending requests to a target server, and returns statistics that show the performance/functionality of the target server/application via tables, graphs, etc.



Screenshots:

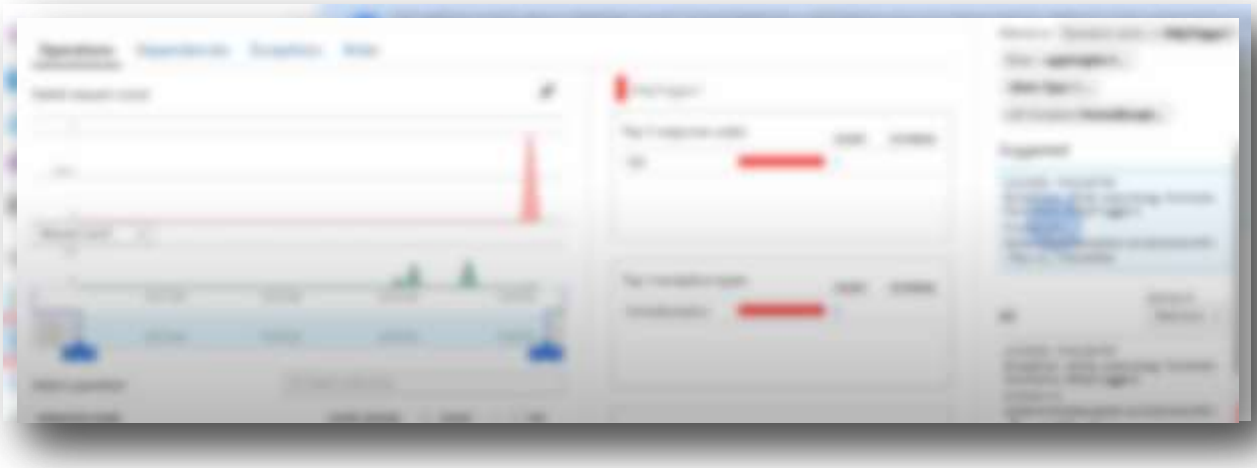
A JMeter test plan looks like as below



Performance Testing for an Eyewear App

Performance Testing Report :

Root Cause Analysis for High Response Time API or Any Error Using Azure Application Insight:



Performance Testing for an Eyewear App