Spotify

Performance Testing Strategy

## **Introduction**

The purpose of this document is to describe how the performance requirements for the Spotify system will be tested and verified in a test environment. The document will outline the testing objectives, load model, load profile, performance requirements, performance test types, parameters and data. All deliverables should be used for internal (in-house) usage only and should not be presented to the public.

## **Objectives**

* Verify that future releases will not cause performance degradation, introducing of performance testing improvements
* Measure and analyze end user experience: how quickly application performs from end user standpoint.
* Measure and analyze system behavior under high 80% of capacity production-like load.

## **Requirements**

In a tables below basic requirements for 2 ways of performance testing (API) will be specified.

#### Common

|  |  |  |
| --- | --- | --- |
|  | **Free account** | **Premium account** |
| Acceptable failure rate | ~0% | ~0% |
| Level of acceptable degradation | TBD% | TBD% |

Key business actions - actions, which is common for current and target system should be defined here.

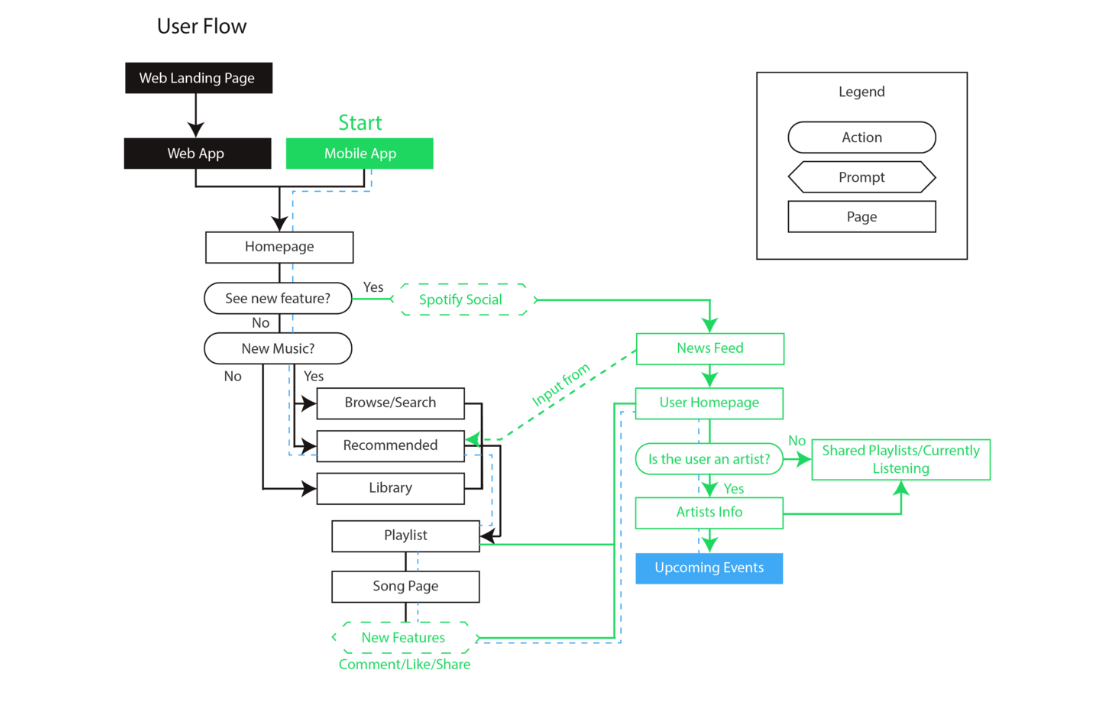
#### API (Backend, Server-Side) - testing of application’s APIs under specified level of load.

|  |  |  |
| --- | --- | --- |
|  | **Free account** | **Premium account** |
| Number of concurrent users | 100 | 100 |
| API response time\* for key business actions (by 90 percentile) | TBD sec | TBD sec |

API response time - server response time (time from moment where api request was sent by a client till last response byte was received).

## **Load model (business scenarios)**

This is the general scenario for Spotify. Will be changed/adjusted during performance testing implementation for each system.



**Schema description:**

In the first testing phase will be checked web application performance with two user account plans: free and premium

## **Test types assumed for conducting**

### API (Backend) performance tests:

#### Validation test (mandatory)

* Purpose:
  + Verify that load script works correctly.
  + Verify than infrastructure for performance tests is prepared correctly
  + Verify that performance tests works as expected, results are reachable in the expected way.
* Load profile:
  + Duration: TBD.
* Entry criteria:
  + Testing environment is prepared.
  + Valid code version is deployed to testing environment.
  + Load test scripts are ready.
  + Load generation infrastructure is prepared.
* Success criteria:
  + This scenario will be successfully completed when:
    - Performance tests work as expected.

#### Response time load test (mandatory)

* Purpose:
  + Measure and analyze performance metrics (response time, error rate) and system behavior under defined load.
  + Compare performance metrics with thresholds/previous results.
  + Load profile:
    - Workload: TBD.
    - Duration: usually: TBD.
    - Target throughput: TBD.
  + Collected metrics:
    - Measurements, collected during performance test by load tool:
      * Response time statistics (Minimum, average, 90% percentiles, maximum response time, Response time standard deviation).
      * Total, pass and fail counts.
      * System throughput.
      * Throughput/response time trends.
  + Entry criteria:
    - Testing environment is prepared.
    - Valid code version is deployed to testing environment.
    - Load test scripts are ready.
    - Load generation infrastructure is prepared.
  + Success criteria:
    - This scenario will be successfully completed when:
      * No unplanned system crashes or exceptions during load test run.
      * Response time metrics meet desired threshold.
      * All observed failures are investigated and understood.

#### Stress (Optional)

For determining how the system or component handles variations in peak load.

#### Scalability (Optional)

For checking system's ability to be readily enlarged. Capacity of ideally scalable system should increase proportionally to the resources allocated for that system. Conducts as a series of ramp-up tests against various system configurations.

#### Stability (Optional)

For checking system's ability to maintain its level of operation. This is a long high load test to analyze if system can work with no issues for a long period of time.

#### Cross-browser testing (Optional)

Time of Javascript parsing and layout rendering are different because each browser has its own javascript engine, rendering engine, mechanism of network interaction

#### User experience testing (Optional)

Emulating of end-user behavior while the application is loaded. Measurements are conducted against the application in normal and loaded state.

## **Environments**

1. Environment for current system – UAT

## **Target contacts**

1. Who is a target contact for performance testing in terms of business perspective? – GUDGUY1
2. Who is a target contact for performance testing in terms of technical perspective? – GUDGUY2