Instagram

Test plan

# Introduction

## Purpose of this Test plan

The purpose of this document is to specify performance requirements and conditions for Instagram application. The document will outline scenarios, test cases, parameters and data used in evaluating the capacity of the included features.

## Test plan identification

|  |  |
| --- | --- |
| Application Name | Instagram |
| Test Cases (scripts) | Anonymous user flow  Viewer user flow  Creator user flow  Seller user flow |

# Test description

## Test objectives

* + 1. Check application’s availability on selected environment
    2. Get basic response time for each tested item with no load
    3. Define capacity in term of number of concurrent virtual users
    4. Check system counters from involved servers

## Items to be tested

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Action** | **Test case (script)** | **Notes** |
| Login | Login | Anonymous, Viewer, Creator, Seller | Will be completed once for each test user, except Anonymous case |
| Open Home Page | Open Home Page | Viewer, Creator, Seller | For Anonymous users – Login page will be opened, other features are not available for anonymous users |
| View News Feed | See more posts | Viewer |  |
| View Random Story | Open Story | Viewer |  |
| View Next Story | Open Story | Viewer |  |
| View separate post | Open Post | Viewer, Creator, Seller | Own posts and other posts have different options |
| Create Post | Create Post | Creator |  |
| Create Story | Create Story | Creator |  |
| Like Post | Leave “Like” | Viewer |  |
| Comment Post | Send Comment | Viewer, Seller |  |
| Like Story | Leave “Like” | Viewer |  |
| Comment Story | Send Comment | Viewer |  |
| Open My Account | Open Account | Creator, Seller |  |
| Open random account | Open Account | Viewer |  |
| Follow another account | Follow account | Viewer, Creator |  |
| View Followers | Open Followers List | Viewer |  |
| View Followings | Open Followings List | Viewer |  |
| Search Account by Name | Search | Viewer |  |
| Search Tags | Search | Viewer, Creator |  |
| Open Activity | Open Activity | Creator, Seller |  |
| Open Direct | Open Direct | Creator, Seller |  |
| Send direct message | Send message | Creator, Seller |  |

1. Home Page (News Feed)

This is the first page opened after entering the application.

Main operations:

* Create new Post
* Open Direct
* Create new Story
* View Stories
* See more posts (scroll via feed). Each post can be liked, commented or sent.

1. Search and Recommended

Main Operations:

* Search field. Search can be completed by name of user, username or tags.
* See more recommended posts (scroll to load more). Contains posts from unfollowed accounts with content based by interests of user.

1. Activity Page

Contains information about last actions other users did to current user account or posts (new Followers or likes for posts). If user has closed account – that page will contain “Follow Request” tab.

1. Account

Main operations:

* Open Post. Each post can be liked, commented or sent. For “My Account”: contains “Edit” option where user has opportunity to update description or add tagged people to photo or video.
* View Followers List
* View Followings List
* For “My Account”: Edit Profile. After opening of Edit user can update avatar photo, their name, username, bio and links.
* For other accounts: Follow or Unfollow. Depends on current relationship between current user and user which account page is opened.

## Items not to be tested

All other features not listed in previous point.

## Test data

1. Set of users with different content level (number of posts)
2. Each user should have some number of posts: from 0 to 100 for Viewer user, from 400 to 1000 for Creator user and from 50 to 500 for Seller user.
3. Each user should have some Followers and Followings.
4. Each post can contain from 1 to 10 photos or videos and text message.

## Test user roles

1. Anonymous user – can only see Log In page. Other features are not available.
2. Logged in users:

* Viewer – common user who opens Instagram couple times per day to check new stories and posts of their friends and chosen bloggers.
* Creator (business account) – user who regularly creates new posts and stories, communicate with followers in Comments and Direct.
* Seller (business account) – user who is selling anything in their account, usually spends time in Instagram in Direct to communicate with possible customers and sometimes adds new posts and stories.

# Non-Functional Requirements

* Highly scalable
* High consistency
* High availability
* High reliability
* User data should be durable. Any uploaded photos should never be lost
* The maximum latency for generating the News Feed is 150 ms.

# Suspension criteria and resumption requirements

## Suspension criteria

* **Not valid or not stable build** of application
* **The need for updates of test scripts** because of significant changes in functionality of application
* **Testing tools issues** - Load Generator, Metric Collector, Visualizer, CI/CD
* **Test environment issues**. Wrong configuration, access problems
* **Test data issues**. Not reached number of users, absence of certain user type accounts, not enough number of posts and stories.

## Resumption criteria

* **Test plan is completed** (updated) and approved.
* **Correct and stable build is running** on chosen test environment.
* Required **test scripts are completed and valid**. Smoke test verified all key steps.
* All **testing tools are confirmed to work** during the test.
* **Test environment has correct configuration** which is verified and approved for usage in current test.
* **Test data satisfies requirements** to run tests with different types of user and posts.

# Test deliverables

Main expected test deliverables are:

* Test strategy and plan
* Basic test scripts and scenarios
* Baseline of main metrics for deployed functionality: response times for the pages, throughput (requests per second), system resources consuming
* Test reports and analysis

# Testing tasks

To run each test case next steps are required:

1. Basic scripting
2. Basic scenarios creation
3. Setting up load generation tools
4. Setting up monitoring and collecting tools
5. Deployment stable version to environment
6. **Smoke** testing. Is used for general check of application availability on selected environment and confirms that test script is valid.
7. Running set of performance testing:
   * 1. **Capacity**. First testing to determine load model for other types of testing.
     2. **Regular load** test. Is completed regularly for each change of application or configuration. Runs with ~50% of capacity.
     3. **Volume.** Checks system and application behavior with different levels of load: low - ~30% of capacity, middle - ~60% of capacity, and high - ~90% of capacity. Is used to find possible issues with data (new photos/videos are not added properly, existing posts have problems with loading or photo/video quality)
     4. **Longevity.** Low load test to check system and application behavior under constant load and to find possible memory leaks. Runs with ~20% of capacity
     5. **Scalability**. Not required for all test cases. Checks application behavior with different configurations.

# Test environments

For performance testing a dedicated environment is needed. The configurations of the servers should be as much as close to production’s ones.

Expected servers:

* Front-End servers
* Back-End servers
* Databases (PostgreSQL, Hive, Pgbouncer, Redis, Memcache)

To perform partial and full end-to-end performance testing it’s recommended to have a separated database and other services, i.e. they should not be located on the same server where related services are being tested by another team.

**Test environment specifications:** Close to Production environment as much as possible.

# Risks

* A significant difference in configuration from the production environment
* Performance testing results can be essentially different even in case of minor difference in think times, arrival rate and test duration
* During the execution of the tests, some major performance or functional problems that may require code changes, creation of a new build may be discovered and in that case it may be necessary to repeat the load test from the beginning
* Load test should be performed against a build that is solid enough, and that has been functionally tested, after code is complete. Failure to follow this rule may result on rework to update test scripts for every new build, plus the load test may need to be repeated from the beginning. This will affect the schedule
* Performance testing tool is not capable of identically reproducing real life scenarios - so results could only be trusted as having limited reliability level
* Network/systems latency issues
* Environment’s unavailability