Table of Contents

1. INTRODUCTION

Customer wants a perfect website that passes the full cycle of manual and automotive testing. Given the specificity of the site it is important to have the same quality of the site as per requirements.

The Test Plan has been created to facilitate communication within the team members. This document describes approaches and methodologies that will apply to the testing process of https://open.spotify.com site. The Test Plan includes but not limited to: objectives, testing types, principles, metrics, responsibilities, entry and exit criteria, schedule. This document has clearly identified what the test deliverables will be, and what is deemed in and out of scope.

2. TEST STRATEGY

2.1. Test Objectives

The objective of the test is to verify that the functionality of https://open.spotify.com works according to the specifications.

The testers will execute and verify the test scripts, identify, report and retest all high and medium severitydefects per the entrance criteria, prioritize lower severity defects for future fixing.

The final product of the test is twofold:

- A production-ready software;
- A set of stable test scripts that can be reused for Functional test execution.

2.2. Test Assumptions

- Exploratory Testing would be carried out once the build is ready for testing
- All the defects would come along with snapshot JPEG format
- The Test Team assumes all necessary inputs required during Test design and execution will besupported by Development/BUSINESS ANALYSTs appropriately
- Test case design activities will be performed by QA Group
- Test environment and preparation activities will be owned by Dev Team
- Dev team will provide Defect fix plans based on the Defect meetings during each cycle to plan. Thesame will be informed to Test team prior to start of Defect fix cycles
- BUSINESS ANALYST will review and sign-off all Test cases prepared by Test Team prior to start of Testexecution
- The defects will be tracked through Jira. Any defect fixes planned will be shared with Test Team priorto applying the fixes on the Test environment
- Project Manager/BUSINESS ANALYST will review and sign-off all test deliverables
- The project will provide test planning, test design and test execution support
- Test team will manage the testing effort with close coordination with Project PM/BUSINESS

ANALYST

- Project team has the knowledge and experience necessary, or has received adequate training in thesystem, the project and the testing processes
- The system will be treated as a black box; if the information shows correctly online and in the reports,it will be assumed that the database is working properly

2.3. Test Principles

- Testing will be focused on meeting the business objectives, cost efficiency, and quality.
- There will be common, consistent procedures for all teams supporting testing activities.
- Testing processes will be well defined, yet flexible, with the ability to change as needed.
- Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
- Testing environment and data will emulate a production environment as much as possible.
- Testing will be a repeatable, quantifiable, and measurable activity.
- Testing will be divided into distinct phases, each with clearly defined objectives and goals.
- There will be entrance and exit criteria.

2.4. Scope

2.4.1. In Scope

Functions to be tested:

- Functional testing of the main page
- Smoke testing of modules: “Main page”, “Sign Up”, “Log In”, “Playlist”
- Performance
- Security
- API

2.4.2. Out of Scope

Not other than mentioned above in section 2.4.1

2.5. Testing types

2.5.1. Manual Exploratory Testing

PURPOSE: The purpose of this test is to make sure critical defects are removed before the nextlevels of testing can start

SCOPE: “Main page”, “Sign Up”, “Log In”, “Search”, “Playlists”

TESTERS: Testing team

METHOD: This exploratory testing is carried out in the application without any test scripts anddocumentation
TIMING: At the beginning of each cycle

2.5.2. Manual Functional Testing

PURPOSE: Functional testing will be performed to check the functions of application. Thefunctional testing is carried out by feeding the input and validates the output from the application

SCOPE: The below table details about the scope of Functional test

Module	Scenarios
Main page	The site must be accessible at the following URL: https://open.spotify.com/
	Verify that the modules in menu are clickable ("Home", "Search", "Sign up", "Log In") and when clicked go to the correct page
Sign Up	Verify that the user can sign up into the site using mail and password
	Verify that the modules "Sign up" has correct link of social media "Sign up with Facebook" and "Sign up with Google"
Log In	Verify that the user can log into the site using an existing mail and password
Main page with user's login	Verify that the modules in menu are clickable ("Home", "Search", "Your Library", "Create Playlist", "Liked Songs", "Upgrade", "Account"
Search	Verify that the user can find a specific query (artist, song, playlist)
Your Library	Verify that the user can add in "Your Library" (artist, song, playlist)
Playlists	Check that the user can create new "Playlist"
	Check that the user can change name of "Playlist"
	Check that the user can add one or more items to a user's playlist
	Check that the user can remove one or more items from a user's playlist.
	Check that the user can follow an existing playlist
	Check that the user can unfollow an existing playlist

TESTERS: Testing Team

METHOD: The test will be performed according to Functional scripts

ENVIRONMENT: OS: Windows 11/64, Browsers: Chrome 106.0.5249.103, Mozilla Firefox 105.0.2 , Microsoft Edge 106.0.1370.34

TIMING: after Exploratory test is completed

2.5.3. Automation Testing

PURPOSE: This test focuses on creating automation scripts based on manual test cases

TESTERS: Testing Team

SCOPE: The same as in Manual Functional Testing

TOOLS: The testing team should use PyCharm as a main IDE, Python as a main language, SeleniumWebDriver as a main framework for test scripts automation, and BrowserStack to run tests in various environments

TIMING: After manual testing is done and all critical issues are resolved

2.5.4. Performance Testing

PURPOSE: This test is performed to measure the speed, responsiveness, stability of the site andalso how well the page is built for optimal performance

TESTERS: Testing Team

SCOPE: https://open.spotify.com

TOOLS: The testing team should use Google Lighthouse, GTMetrix, SpeedLab

BASIC METRICS: Page Load, Speed Index, FCP (first content paint), LCP (largest content paint),TBT (total blocking time), CLS (cumulative layout shift)

TIMING: After manual testing is done and all critical issues are resolved

2.5.5. Security Testing

PURPOSE: This test is performed to reveal current or potential security vulnerabilities

TESTERS: Testing Team

SCOPE: https://open.spotify.com

TOOLS: The testing team should use Snyk and Mozilla Observatory

TIMING: After manual testing is done and all critical issues are resolved

2.5.6. API Testing

PURPOSE: API tests are performed to determine if the developed APIs meet the

expectationswhen it comes to the functionality, performance, reliability and security of the website

TESTERS: Testing Team

SCOPE: https://open.spotify.com

TOOLS: The testing team should use Postman API as a platform for building and using APIs andJavaScript as a language for test scripts

TIMING: After manual testing is done and all critical issues are resolved

2.6. Test Effort Estimate

Testing Type	Estimate (hours)	Start	Finish
Manual			
Automation			
Performance			
Security			
API			

3. EXECUTION STRATEGY

3.1. Entry and Exit Criteria

The entry criteria refer to the desirable conditions in order to start test execution; only the migration ofthe code and fixes need to be assessed at the end of each cycle.

Entry conditions:

- All test hardware platforms should be successfully installed, configured, and functioning properly
- All the necessary documentation, design, and requirements information should be available that willallow testers to operate the system and judge the correct behavior
- All the standard software tools including the testing tools should be successfully installed andfunctioning properly
- Proper test data is available

- The test environment such as, lab, hardware, software, and system administration support should beready
- QA resources have complete understanding of the requirements
- QA resources have sound knowledge of functionality
- Reviewed test scenarios, test cases and RTM

The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.

Exit conditions:

- 100% Test Scripts executed
- 95% pass rate of Test Scripts
- No open Critical and High severity defects
- 95% of Medium severity defects have been closed
- All remaining defects are either cancelled or documented as Change Requests for a future release
- All expected and actual results are captured and documented with the test script
- All defects logged in Jira
- Test Closure Memo completed and signed off

Entry and exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation. All this is input to the project manager for a final "go-no go" decision.

3.2. Test Cycles

- There will be two cycles for functional testing. Each cycle will execute all the scripts
- The objective of the first cycle is to identify any blocking, critical defects, and most of the highdefects. It is expected to use some work-around in order to get to all the scripts
- The objective of the second cycle is to identify remaining high and medium defects, remove thework-around from the first cycle, correct gaps in the scripts and obtain performance results
- Performance, Security and API tests will consist of one cycle

3.3. Validation and Defect Management

- The defects will be tracked through Jira. The technical team will gather information on a daily basis from Jira, and request additional details from the Defect Coordinator. The technical team will work onfixes.
- It is the responsibility of the tester to open the defects, link them to the corresponding scripts, assign an initial severity and status, retest and close the defects; it is the responsibility of the technical team to review Jira on a daily basis, ask for details if necessary and fix the defects.

Defects found during the Testing will be categorized according to the bug-reporting tool “Jira” and the categories are:

Severity	Impact
Highest	▪ This bug is critical enough to crash the system, cause file corruption, orcause potential data loss ▪ It causes an abnormal return to the operating system (crash or asystem failure message appears) ▪ It causes the application to hang and requires re-booting the system
High	▪ Major system component unusable due to failure or incorrectfunctionality ▪ High severity bugs cause serious problems such as a lack of functionality, or insufficient or unclear error messages that can have amajor impact to the user, prevents other areas of the app from being tested, etc ▪ High severity bugs can have a work around, but the work around isinconvenient or difficult
Medium	▪ This Bug will degrade the quality of the System. However there is anintelligent workaround for achieving the desired functionality - for example through another screen ▪ This bug prevents other areas of the product from being tested.However other areas can be independently tested
Low	▪ There is an insufficient or unclear error message, which has minimumimpact on product use

4. TEST ENVIRONMENT

Environment x Support level 1 (browsers):

- Windows 11 : Edge, Chrome (latest), Firefox (latest), Safari (latest)
- Mac OS : Safari (latest)

5. APPROVALS

The Names and Titles of all persons who must approve this plan.

Signature:	
Name:	
Role:	
Date:	

Signature:	
Name:	
Role:	
Date:	

TEST PLAN

Spotify – Web Player
https://open.spotify.com/

Spotify Web Player Playlist functionality

Table of Contents

1. INTRODUCTION

The Test Plan has been created to facilitate communication within the team members. This document describes approaches and methodologies that will apply to the modules: "Main page", "Sign Up", "Log In", "Playlist" of the website https://open.spotify.com/.

This document has clearly identified sub-tasks of website's module testing, types of testing, what devices, environments and tools to be used. It includes test cases with expected and actual results, schedule and executing estimated time each of sub-tasks and entire project.

2. SCOPE

The document mainly targets the GUI, Exploratory, Functional Positive, Negative and Ad-hoc testing of User's Personal Account Address Menu, Performance and Security testing of the website.

3. SUB-TASKS

3.1. WEBSITE MANUAL TEST

Execute manual testing for the User's Personal Account Address Menu based on the next types of testing:

- GUI;
- Exploratory testing;
- Functional Positive testing:
- Functional Negative testing;
- Functional Ad-hoc testing.

GUI Testing:

GUI testing includes testing the UI part. It covers look and feel, error messages, spelling mistakes, GUI guideline violations.

Exploratory Testing:

Exploratory testing will includes a type of software testing where Test cases are not created in advance but QA check system on the fly.

Functional Positive testing:

Checks whether an application behaves as expected with positive inputs.

Functional Negative testing:

Invalid data is inserted to compare the output against the given input.

Functional AD-hoc testing:

Includes an informal testing type with an aim to break the system.

Environment Support:

- Laptop
- OS: Windows 11/ 64
- Browsers (latest versions): Google Chrome, Firefox Mozilla, Microsoft Edge

The following test cases are covers these manual tests:

TC	Test Action	Expected Results
POSITIVE TESTS		
01	The site must be accessible at the following URL: https://open.spotify.com/	Https protocol is active
02	Verify that the user can sign up into the site using a correct mail and password	The user successfully registered on the sites
03	Verify that the user can log into the site using an existing mail and password	The user successfully entered on the sites
04	Validate that menu links lead to the right pages (“Home”, “Search”, “Your Library”, “Liked Songs”)	Each menu item is processed according to its name and URL
05	Verify that the user can find a specific query (artist, song, playlist, album)	Valid request found
06	Check that the user can create new “Playlist”	New playlist successfully created
07	Check that the user can change name of “Playlist”	Playlist name changed successfully
08	Check that the user can add item to a user's playlist	Item added to playlist successfully
09	Check that the user can remove item from a user's playlist.	Item deleted to playlist successfully
10	Check that the user can follow an existing playlist	Playlist successfully added in “Your Library”
11	Check that the user can unfollow an existing playlist	Playlist successfully deleted from “Your Library”
NEGATIVE TESTS		
12	Verify that the Log in with invalid email address is not accepted	Got the error message: “Incorrect username or password.”

13	Verify that the Log in with invalid password is not accepted	Got the error message: "Incorrect username or password."
14	Verify that the Log in with empty email field is not accepted	Got the error message: "Incorrect username or password."
15	Verify that the Log in with empty password field is not accepted	Got the error message: "Incorrect username or password."
AD HOC TESTS		
16	Verify that the registration with invalid (ad hoc) email address (test@mail.comtest@mail.com) is not accepted	Got the error message: "Incorrect username or password."

Testing Tools used:

Process	Tools
Test case creation	Microsoft Word, Microsoft Excel, JIRA
Test case tracking	Microsoft Word, Microsoft Excel, JIRA
Test case execution	Manual
Test case management	Microsoft Excel, JIRA
Test reporting	Microsoft Excel, JIRA

3.2. WEBSITE AUTOMATION TEST

Automation tests are creates by using Selenium WebDriver Python UnitTest frameworks on the basis of Functional manual test cases and executes to check functionality of User's Personal Account Address Menu.

Environment Support:
- Laptop
- OS: Windows 11/ 64
- Browsers (latest versions): Google Chrome, Firefox Mozilla, Microsoft Edge

The following test cases are covers automation tests:

TC	Test Action	Expected Results
POSITIVE TESTS		
01	Verify that the user can log into the site using an existing mail and password	The user successfully entered on the sites
02	Verify that the user can find a specific query (artist, song, playlist, album)	Valid request found
03	Check that the user can create new "Playlist"	New playlist successfully created
04	Check that the user can change name of "Playlist"	Playlist name changed successfully

05	Check that the user can add item to a user's playlist	Item added to playlist successfully
06	Check that the user can remove item from a user's playlist.	Item deleted to playlist successfully
07	Check that the user can follow an existing playlist	Playlist successfully added in “Your Library”
08	Check that the user can unfollow an existing playlist	Playlist successfully deleted from “Your Library”
NEGATIVE TESTS		
09	Verify that the Log in with invalid email address is not accepted	Got the error message: “Incorrect username or password.”
10	Verify that the Log in with invalid password is not accepted	Got the error message: “Incorrect username or password.”
11	Verify that the Log in with empty email field is not accepted	Got the error message: “Incorrect username or password.”
12	Verify that the Log in with empty password field is not accepted	Got the error message: “Incorrect username or password.”
AD HOC TESTS		
13	Verify that the registration with invalid (ad hoc) email address (test@mail.comtest@mail.com) is not accepted	Got the error message: “Incorrect username or password.”

Note: For ease of perception, the tests are numbered starting from the first, separately from manual tests.

Testing Tools used:

Process	Tools
Test case creation	PyCharm, Selenium
Test case tracking	PyCharm, Browsers
Test case execution	Automation
Test case management	PyCharm
Test reporting	HTML reports, Allure reports

Additional automation tests execute on the cloud platform “Browserstack” :

- OS: Windows 10 (Browsers (latest versions): Google Chrome, , Firefox Mozilla, Microsoft Edge)
- OS: macOS Big Sur (Browsers: Safari, version 14.1)

Testing Tools used:

Process	Tools
Test case creation	PyCharm, Selenium
Test case tracking	Browserstack.com
Test case execution	Automation
Test case management	PyCharm
Test reporting	Video files (mp4)

Test results are the same as on local device.

3.3. WEBSITE API TEST

TC	Test Action	Expected Results
POSITIVE TESTS		
19	POST. Create new Playlist	Status: 201 Created
20	PUT. Change Playlist Details	Status: 200 OK
21	POST. Add Items to Playlist	Status: 201 Created
22	DEL. Remove Playlist Items	Status: 200 OK
23	PUT. Follow Playlist	Status: 200 OK
24	DEL. Unfollow Playlist	Status: 200 OK
25	GET. Get Specific User's Playlists	Status: 200 OK
NEGATIVE TESTS		
26.1	POST. Create playlist with invalid name format	“Error parsing JSON.” Status: 400 Bad Request
26.2	POST. Create playlist with invalid public status format	"Error parsing JSON ." Status: 400 Bad Request
27.1	PUT. Change Playlist Details in invalid name format	“Error parsing JSON.” Status: 400 Bad Request
27.2	PUT. Change playlist details in invalid description format	“Error parsing JSON.” Status: 400 Bad Request
27.3	PUT. Change playlist details in invalid public status format	“Boolean value must be either true or false” Status: 400 Bad Request
28.1	POST. Add items to playlist with invalid data in body	“Error parsing JSON.” Status: 400 Bad Request
28.2	POST. Add items to playlist with invalid uris (resource identifier)	“Invalid track uri: spotify:track:(invalid data).” Status: 400 Bad Request
29.1	DEL. Delete Playlist Items with invalid uris	“JSON body contains an invalid track uri: spotify:track:(invalid data).” Status: 400 Bad Request
29.2	DEL. Delete Playlist Items with invalid data in body	“JSON body doesn't conform to specification” Status: 400 Bad Request
29.3	DEL. Delete Playlist Items with empty data in body	“JSON body contains a track without URI.” Status: 400 Bad Request
30.1	PUT. Follow Playlist with empty data in body	“Insufficient scope” Status: 403 Forbidden
30.2	PUT. Follow Playlist with invalid public data	“JSON body does not adhere to the defined endpoint parameters” Status: 400 Bad Request

API tests execute to determine whether the APIs that are developed meet expectations when it comes to functionality, performance, reliability and security for website.

Environment Support:
- Laptop
- OS: Windows 11/ 64
- Browsers (latest versions): Google Chrome (DevTools)

The following test cases are covers API tests:
(There are main tests results in the table)

Testing Tools used:

Process	Tools
Test case creation	Postman, Chrome DevTools
Test case tracking	Postman
Test case execution	Automation
Test case management	Postman
Test reporting	Newman

3.4. WEBSITE AUTOMATION PPERFORMANCE TEST

Performance automation tests execute for measures the speed, responsiveness and stability of the tested website. Tests execute in incognito environments.
After testing reports were generate.

Testing Tools used:

Process	Tools
Test case creation	Google Lighthouse, GTMetrix, Webpagetest, BrowserStack-SpeedLab
Test case tracking	Google Lighthouse, GTMetrix, Webpagetest BrowserStack-SpeedLab
Test case execution	Automation
Test case management	
Test reporting	txt -, html -, pdf - files

3.5. WEBSITE AUTOMATION SECURITY TEST

Security automation tests execute to reveal potential flaws or weaknesses of software and website. Testing focuses on whether the application is designed and configured correctly.
Tests execute in incognito environments.
After testing reports were generate.

Testing Tools used:

Process	Tools
Test case creation	Mozilla Observatory, Snyk
Test case tracking	Mozilla Observatory, Snyk
Test case execution	Automation
Test case management	
Test reporting	txt -, html -, pdf - files

4. TESTS SCHEDULE AND ESTIMATED TIME

Sub-Task Name	Start	Est. time (hours)	Finish
Manual Testing			
Automation Testing			
API Testing			
Performance testing			
Security testing			

Test Execution
QA -
Test phase -
Status -

APPROVALS:

	Project Manager	QA Lead
Signature		