TEST PLAN

Project ID: PRJC_TST
Author: Kseniya Akulovich
Date of creation: August 5, 2022

REVISION HISTORY			
Date	**Changes made**	**Changed by**	**Approved by**
August 5, 2022	Document creation	Kseniya Akulovich	

TABLE OF CONTENTS

RELATED DOCUMENTS

Document ID	Description

ABBREVIATIONS AND DEFINITIONS

Abbreviation	Definition

1. General Information

This document describes the scope of work required for completing the "PRJC_TST" project by cataloging the objectives, test strategy and approach, testing specifics, estimations (incl. risk mitigation) and resources allocated to the project given above.

The main purpose of this project is to create the "MyTastes" web application that will provide users with an opportunity to share their taste in music, literature, and movies online.

The "MyTastes" web application (EN) will consist of three modules: Audio, Films, and Library. A registered user can add info on audio, films and books in their profile and also rate the experience that the user had with such an item. To find someone with the same taste in music, literature, or movies, the user needs to filter other accounts using the search parameters in the application.

2. Project Scope

2.1 Features To Be Tested

#	Module Name	Description	Reference
1.	Audio	The 'Audio' module is designed to provide any user with an opportunity to upload info about the audios they've listened to in order to share their opinion. Once all the info about a certain song is uploaded to the profile via the form, it'll be shown in the user's list of audios with all the relevant information. All the data can be edited later by the user. The profile itself will be available for other users through the search (see 'MyTastes' specification).	#R_26-45 ('MyTastes' specification)
2.	Films	The 'Films' module is designed to provide any user with an opportunity to upload info about the films they've watched in order to share their opinion. Once all the info about a certain movie is uploaded to the profile via the form, it'll be shown in the user's list of movies with all the relevant information. All the data can be edited later by the user. The profile itself will be available for other users through the search (see 'MyTastes' specification).	#R_46-62 ('MyTastes' specification)
3.	Library	The 'Library' module is designed to provide any user with an opportunity to upload info about the books they've read in order to share their opinion. Once all the info about a certain book is uploaded to the profile via the form, it'll be shown in the user's list of books with all the relevant information. All the data can be edited later by the user. The profile itself will be available for other users through the search (see 'MyTastes' specification).	#R_1-25 ('MyTastes' specification)
4.	Profile management	The user can populate their profile with personal info: name, date of birth, links to social media etc. (see 'MyTastes' specification), and configure profile availability.	#R_63-75 ('MyTastes' specification)
5.	Search	The user can see other profiles; the user needs to go to search, enter the name of any item and set the rating for that item in the search filter; the output will contain profiles that have rated this item the same way (see 'MyTastes' specification).	#R_76-90 ('MyTastes' specification)

2.2 Features Not To Be Tested

#	Module Name	Description	Reference
1.	Chat widget	The chat widget will be integrated as a third-party component.	See 'MyTastes' specification

2.3 Third-Party Components

#	Name	Role	Reference
1.	Apache HTTP Server	Web Server	See 'Environment Configuration'
2.	Apache Tomcat 9	Application Server	See 'Environment Configuration'
3.	MySQL 8.0.22	Database	See 'Environment Configuration'
4.	phpMyAdmin	Intended to handle the administration of a MySQL	See 'Environment Configuration'
5.	PHP	Works with Apache to build dynamic web pages	See 'Environment Configuration'
6	Chat widget	Will be used to contact support	See 'Environment Configuration'

3. Test Strategy And Approach

The "MyTastes" application will be mainly tested using the black box approach without knowledge of the internal structure or program source code; automated testing with the involvement of QA Automation Specialists might be considered a future option in the case of project expansion.

3.1 Entry/Suspension/Resumption/Exit/Acceptance Criteria

○ *Entry criteria:* new build. The testing process may not be initiated if there is an issue with build installation

○ *Suspension criteria:* critical path testing will begin only if at least 90% of smoke test cases are passed; critical path testing will be stopped if there is at least 50% failure rate for CP test cases. Moreover, the testing process will be stopped if the build does not contain the specified change;

○ *Resumption criteria:* more than 50% of bugs found during the previous iteration are fixed;

○ *Exit criteria:* more than 80% planned for the current iteration test cases are executed

○ *Acceptance criteria*: 100% success of test cases for smoke testing and 90% success of test cases for critical path testing; all critical and major bugs are fixed.

3.2 Testing: Types And Levels

The following testing types will be used to test the "MyTastes" web application:

Testing Type	Definition	Reasoning
Alpha testing	Simulated or actual operational testing by potential users/customers or an independent test team at the developers' site, but outside the development organization	The team will use this testing type to provide an internal acceptance testing and ensure that the product being developed is of high quality, before it is handed over to the Customer.

Application (Business-logic) tier testing	Testing scope focused around clear business needs and demands of the Customer according their business rules	The team will evaluate how particular business restrictions and demands implemented in the final product (such as company's color scheme, impossibility to download audio/films/books from the site etc.)
Black-box testing	Testing process that revolves around the lack of knowledge about architectural and code solutions of the application	The team will evaluate the quality of all the features implemented without interactions with their internal code.
Cross-browser testing	A set of tests which to show compatibility with browsers and browser versions	The team will test if the application works and is depicted correctly in the following browsers: Google Chrome, Firefox, and Opera.
Dynamic testing	Code execution testing (with an application started and its functions)	The team will evaluate the integrity of the application, its functional and non-functional performance quality, all kinds of user/usage scenarios execution.
Exploratory testing	An informal test design technique where the tester actively controls the design of the tests as those tests are performed and uses information gained while testing to design new and better tests.	The team will use the exploratory testing approach scope in order to facilitate the most in-depth coverage of quality criteria (e. g. exploring steps to reproduce bugs, applying wide data range, etc.).
Functional testing	A type of testing that verifies that the application functionality works correctly (the correct implementation of functional requirements).	Our team will verify that each feature (function) of the application operates in accordance with functional specification (user stories, use case, requirements specification).
GUI testing	A testing type aimed to check the interfaces of an application or its components.	Our team will ensure that all the information in the graphical user interface (fonts size and types, alignments, labels, icons and pictures size, etc.) is unified and matches the requirements (mockups).
Domain testing (Equivalence partitioning, Boundary value analysis)	A testing technique based on equivalence classes and boundary conditions. It is used to identify efficient and effective test cases when multiple variables can or should be tested together.	Our team will generate combinations of values and provide multiple execution of test cases using these generated combinations as input data.
Installation testing	A type of testing aimed to identify defects that affect the installation phase of an application.	The team will follow the positive installation scenario according to the special instruction and will deploy the application.
Integration testing	A type of testing where software modules are integrated logically and tested as a group.	The team will review the quality of mutually dependent components of the application (e.g., integration with chat widget, co-dependencies between the forms for adding editing and lists of added items etc.).
Localization testing	A testing type aimed to ensure the correctness and quality of the adaptation of the product for use in a particular language, taking into account national and cultural characteristics.	Our team will test appropriate linguistic and cultural aspects for a particular locale, in particular date, time, unit formats, etc.
Manual testing	Testing process revolves around test cases that are representative when executed manually.	Our team will not use automation scripts to test this application, since it is out of scope of this project.
Module testing, Component testing	Testing of each individual application component separately.	The team will decompose the application (for instance, forms for

		addition/editing, list of added items, profile management etc.) and focus on nuclear functions of individual application component.
Negative testing	A type of testing to show the system behavior when non-standard operations (sometimes invalid ones) performed or data used that may potentially lead to errors.	Our team will follow alternative and exceptional scenarios (for example, adding an invalid/corrupted file) that should end up with an error, as well as use invalid data (where appropriate, for example, for input areas) to check whether the system responds as it should.
New feature testing	Testing of the new features of the application.	Our team will examine specifications corresponding to the new feature and create test cases exclusively to test this new feature.
Pairwise testing	A black box test design technique in which test cases are designed to execute all possible discrete combinations of each pair of input parameters.	Our team will use special tools (such as PICT) and create test cases which are built on the principle of testing pairs of values of parameters (variables) instead of trying to test all possible combinations of all values of all parameters.
Positive testing	A type of testing to show that the application works properly and as intended; a tester carries out their activities strictly as instructed.	Our team will examine the application and use valid data, follow typical scenarios and do not try to cause errors, deviations or any incorrect output. In the context of positive testing, we will consider use cases that do not result in errors or do not imply any incorrect input/output.
Presentation tier testing	The scope of testing that covers end-user and “outer world” communication processes of the application. The assessment is not limited to usability, responsiveness, compatibility etc. criteria.	The team will evaluate the quality of the front-end structure, visual and functional criteria.
Regression testing	A type of testing aimed at verifying the fact that previously working functionality has not been affected by errors caused by changes in the application or its environment.	Once the bug has been fixed, our team will make sure that the process of it being fixed has not caused any errors in other parts of the application.
Re-testing	Re-execution of the previously failed tests to confirm the fix.	Once the bug has been fixed, our team will verify that it does not occur anymore by re-running the tests failed previously.
Static testing	General testing w/o code execution (incl. documentation, requirements review, prototyping, code review, etc.).	The team will assess and improve the requirements, related models and documentation (mockups).
System testing	All the parts of the application are integrated into the working system and its quality is assessed.	The scope of team’s testing process will cover mutual functionality of the whole application and its components integrated.
Testing based on test cases	Testing process that is built around predefined test cases suits and sets in order to facilitate the most comprehensive and traceable approach to achieve quality criteria.	The team will create test cases suits and sets in order to cover and meet the whole agenda of quality and acceptance criteria.

Usability testing	Testing whether the end-user understands how to use the product as well as how much the end user enjoys using the product.	Our team will ensure that all interface elements are placed correctly in the application, their intended purpose is easily understandable; all links/buttons should be clickable and cue a certain event.
Web application testing	Testing process uses certain approach and method which is particularly effective for web applications (e.g. compatibility, performance etc.).	As the product is a web application, our team is intended to test the standard functionality which is typical for such kinds of applications.

Smoke Testing

is performed to quickly assess the readiness of the product for deeper testing. The team will test main functions of the ‘MyTastes’ application to perform build acceptance on the most important server/client configuration. Therefore, this type of testing will be performed on Win 10Pro x64 + Google Chrome 104.0.5112.79/81, Mozille Firefox 103.0, Opera 89.0.4447.83. Please see “Entry/Suspension/Resumption/Exit/Acceptance criteria” for more details on criteria relevant to accept the build.

Critical Path Testing

Provided that smoke testing has been completed successfully and the relevant criteria have been met, the team will proceed with critical path testing. The purpose thereof is to detect bugs that would affect the major functionality considered to be the most important one for the product users. This testing will be performed manually by following steps indicated in the relevant test cases.

Extended Testing

The purpose of extended testing is to detect defects related to the non-typical but still possible and likely usage scenarios (such as uploading a corrupted file, reaction of the application to any manual input in the fields where a value is mainly chosen from an integrated mechanism, like choosing a date from a date picker, entering values into a stepper instead of using arrows etc.). This type of testing will be performed both according to test cases and using ad hoc testing scenarios.

3.3 Bug Tracking And Definitions

JIRA is a bug tracking system that will be used on this project.

Severity:

Blocker: Blocks development and/or testing work, production could not run.
Critical: Crashes, loss of data, severe memory leak.
Major: Major loss of function.
Minor: Minor loss of function, or other problem where easy workaround is present.
Trivial: Cosmetic problem like misspelt words or misaligned text.

Priority:

ASAP: If a defect has a high impact on a user and business performance, it is essential to fix it as soon as possible.

High: The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed.

Medium: During the normal course of the development activities defect should be resolved. It can wait until a new version is created.

Low: The Defect is an irritant but repair can be done once the more serious Defect has been fixed.

4. Risk Evaluation And Mitigation

#	Risk Description	Mitigation Measures	Status	Probability Rate, %
1.	Unavailability of team members due to sick days/vacations/other matters	Shared responsibilities between team members with equal roles, proper workload distribution	Mitigated	40
2.	Dependency on third-party components	Having professionals with an expertise in using the indicated third-party components and keeping in touch with additional workforce that might provide professional advice	Controlled	30
3.	Changes in the scope of the project	Arranging efficient engagement of the Customer to clarify and prioritize requirements.	Active	20
4.	Force majeure events resulting in access limitations or failure to use delivery tools and other possible processes interferences	Utilizing tools that can be accessed locally in order to decrease the number of tools that might be forbidden to access to	Active	60

5. Resources

5.1 Team And Scope Of Responsibilities

Key Project Resources:

#	Name	Contact Information
1.	Project Manager	John Smith, johnsmith@ourcompany.com
2.	Project Coordinator	Anna Doe, annadoe@ourcompany.com
3.	Key Developer	Marianne Krasowski, mariannekrasowski@ourcompany.com
4.	Test Lead	Peter Parker, peterparker@ourcompany.com

Test Team:

#	Name	Role	Responsibilities	Contact Information
1.	Peter Parker	Test Lead	Test plan creation, definition and further optimization of the test strategy (approaches, methods, tools etc.) to be employed and quality criteria, standardization of processes and approaches, estimation of testing	peterparker@ourcompany.com

			effort and testing progress in general, identification of delivery issues related to testing and resource needs, assignment of work to the testing team members and keeping track of their performance, provision of TRRs	
2	Kurt Meyer	Tester	Requirements analysis, development and maintenance of test cases, functional and non-functional testing, bug reporting and validation of bug fixes, peer review with regard to testing artifacts, reporting on testing status	kurtmeyer@ourcompany.com
3.	Elizabeth Arrington	Tester	Requirements analysis, development and maintenance of test cases, functional and non-functional testing, bug reporting and validation of bug fixes, peer review with regard to testing artifacts, reporting on testing status	elizabetharrington@ourcompany.com

5.2 Test Environment

Servers:
We've chosen cloud deployment for the database/application/web servers with the following configurations:
- Hardware: Cloud, 4Cores, 16GB RAM, AllFlash 70GB;
- Software: WinSrv 2019 x64, MySQL 8.0.22 + phpMyAdmin (DB)/Apache Tomcat 9 (app) + PHP/Apache HTTP Server (web);

Clients:
Three machines with the following configurations:
- Hardware: 8Cores, 32GB RAM, SSD 1 TB/4Cores, 16GB RAM, SSD 0.5TB/2Cores, 8GB RAM, HDD 0.5TB;
- Software: Win 10Pro x64, Google Chrome, Mozilla Firefox, Opera (see 'Smoke Testing' for detailed configurations)

5.3 Test Tools

#	Tool	Reasoning
1.	Confluence	Requirements and documentation tracking
2.	JIRA	Bug-tracking system for bug reporting and storage of test cases
3.	PICT	All-pairs testing of search mechanisms
4.	FlashBack Express Recorder	Screenshots and screencasts of defects
5.	Microsoft Teams	Communication on the project

6. Test Artifacts

#	Type	Created by	Frequency	Method of delivery
1.	Test plan	Test Lead	Before the actual testing process	To be published on Confluence
2.	Test cases	Testers	Before the actual testing process	Created in MS Excel/JIRA
3.	Bug reports	Testers	Upon finding a bug	To be published in JIRA
4.	Test result reports	Testers	Weekly	To be published on Confluence

7. Schedule

#	Activity	Start Date	End Date	Created by
1.	Creation of the test plan	Once all the requirements have been clarified	The test plan is ready and approved by the Customer	Test Lead
2.	Creation of the test cases	As soon as the TP is ready	Test cases are developed and peer-reviewed	Test Team
3.	Environment set-up	Environment configuration is known	The environment is configured	Test Team
4.	Build installation	After the build is ready to be installed	The build is installed and ready to be tested	Test Team
5.	Execution of smoke testing	The build is installed and ready to be tested	All the critical and major bugs revealed during this stage are fixed and verified	Test Team
6.	Execution of critical path testing	Smoke testing is completed and meets its exit criteria, the build is ready to be tested for its main functionality	All the critical and major bugs revealed during this stage are fixed and verified	Test Team
7.	Execution of extended testing	Critical path testing is completed and meets its exit criteria, the build is ready to be tested for rare scenarios and exceptions	All the critical and major bugs revealed during this stage are fixed and verified	Test Team
8.	TRR creation	Once a sprint has come to the end or there is an urgent need to report the progress	TRR is provided to the Customer	Test Lead

Please note that more detailed work estimations (in h/story points) will be provided in a separate document.