Yalla Kora

Test Plan

TABLE OF CONTENTS

1. INTRODUCTION

1.1.Overview

A Test Plan is a document which describes a scope of testing, test strategy, objectives, effort, schedule and resources required. Its main purpose is to guide the whole testing process and used mostly by Project Managers or Tests Engineers.

1.2.Purpose

This Test Plan document for the 'Yalla Kora' supports the following objectives:

- Identify existing project information and software components to be tested.
- Recommendation and description of the testing strategies to be employed.
- Identify required resources and provide a test effort estimate
- List the test project deliverable elements.

The results of test execution will be sent to the customer as reports. All found bugs will be tracked using Trello bug tracker.

1.3.Scope

Testing of mobile application is in the scope of this test plan. The following components and functions would be tested:

1. Check the redirected links to the news
2. Add favorite team
3. Edit the favorite teams list
4. Check the push Notification
5. Play the video files
6. Search for specific team

1.4.References

1. project plan
2. system requirements
3. specification design documentation.

2. Work plan

The parties are agreed to follow the next work plan:

1. Test plan preparation
2. Test plan approval
3. Functional testing and bugs reporting
4. Daily reports preparation
5. Final report preparation

3. Test plan and strategy

3.1. Functional Testing

The objective of functional testing is to make sure that the whole software product works according to the requirements, and no significant errors appear in the application. Functional testing is the most substantial part of software testing. It involves checking of different aspects of the system. A software product must pass all the planned tests. Only in this case its quality can be assured.

Test Objective	**Ensure proper target-of-test functionality**
Technique	Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following: ● The expected results occur when valid data is used. ● The appropriate error or warning messages are displayed when invalid data is used. ● Each rule is properly applied.
Entry Criteria	The application construction is completed. • The test engineers are dedicated. Necessary devices, instruments, and other equipment are acquired. • Test environment is prepared, and the application is released to the test environment. • Test environment is prepared, and the application is released to the test environment.
Completion Criteria	All the planned tests are performed. There are no show-stopping errors. All the errors of high priority and severity are fixed. The test results are evaluated, discussed and approved.

3.2. Test Procedure

Test procedure assumes the next points:

- Reporting of found software bugs.
- Various aspects of the tested software should be checked; this requires executing of
- different testing types.
- The main testing type that would be executed:
- Functional Testing
- UI Testing
- Usability Testing

- Compatibility Testing (4 modern web browsers and devices)
- Regression testing
- Retesting (during the second round if needed)

It also will be checked how the software product is run on the browsers and devices that are supposed to support it, how it starts and stops, how much time does it need to launch.

During this test round next types of testing will NOT be applied:

- Security testing

3.3. Entry Criteria

- planning phase has been finished
- testable units are available
- all functional requirements have been defined
- a unit testing environment has been set up.

3.4. Exit Criteria

- all planned test cases have been covered
- all the bugs found have been reviewed
- performance of key modules has been tested.

3.5. Bug Reporting

Bug reports are created in order to provide the development team and the project managers with exhaustive information about the discovered defects. They must be helpful in determining causes of the errors and correcting them.

Defect Severity can be classified into four categories:

- Critical (blocker) defects are the failure of the complete software system or of a critical subsystem, and no work or testing can be carried out after the occurrence of the defect. It also applies to data loss failures and with processes that leave inconsistent data stored on the database.
- Major defects (and crashes) are those which also causes failure of entire or part of the system, but there are some processing alternatives which allows further operation of the system. It also applies to the system crashing, or aborting, during normal operation of a non-critical flow.
- Minor defects do not result in failure but causes the system to show incorrect, incomplete, or inconsistent results.
- Trivial defects are small errors that do not affect the functionality: typos, grammar mistakes, wrong terminology, etc.
- The information that is indicated in each bug report:
  - the software product name;
  - version number of the software product (if tested on mobile);
  - Each report provides the next information about the defect:

  - summary, which is short description of the problem;
  - location of the defect in the software product;
  - steps to reproduce the error;
  - frequency of the defect occurrence;
  - severity of the defect;
  - additional information about the defect in form of attached screenshots or video records.

Third party software will be used for reporting and maintaining discovered errors. The test team will log in that software all the defects that will be found during the testing process.

4. Resources

4.1. Tools

The following tools will be used for this project:

Name of process	Tool
Defect Tracking	Jira
Test Cases	TFS

4.2. List of Devices

Name of device	OS
iPhone devices	All supported OS
Android devices	All supported OS

5. Risks and Assumptions

5.1. Risks

The following risks exist while testing a mobile application:

- availability of devices
- new features and modification which have not been planned in advance
- changes in requirements
- delays in schedule.

5.2. Assumptions

- each release is accompanied by a note with information about implemented features and their impact on the system
- all “bugs-blockers” receive high priority
- all the bugs found are fixed before the next software release
- all documents are up-to-date and delivered to the testing team in time
- all necessary equipment and tools are provided and ready for testing
- tests schedule is reviewed in case there are any obstacles for testing activity.

6. Test Metrics

- requirement coverage
- test cases coverage
- number of tests executed
- number of defects found (taking into consideration their priorities and severities)
- tests design effort
- total test effort.

7. Logging Tests and Reporting

Each found issue should be properly reported using TFS tool.

8. Roles and Responsibilities

Project Manager

- Managing the whole testing process.
- Providing all the needed resources for the testing activities.

QA Lead

- Collecting and learning the requirements.
- Validation of the documentation.
- Planning the testing works.
- Monitoring the testing activities, making sure that the works are performed according to the plan.
- Reporting about the progress, number and severity of the found errors.

Test Engineer

- QA process / logging found errors into the approved bug tracking system.

9. Deliverables

- tests plan
- test cases documents
- test strategy
- test results
- test summary report.