Test Plan for Tangerine™ - Mobile Assessments Made Easy

Group Members:
Ethan - s3897708
John - s3897812
Enrique - s3850685
Riveen - s3896471
Abhinav - s3853479
Ishmum - s3881330

Purpose

Tangerine is an open source software application designed to collect and analyse reading and mathematics assessment data in primary schools. Data can be reviewed on an individual student level, at the school level, or at the regional and even national level through three different platforms. The data gathered and used by teachers, coaches, and ministry officials helps these actors design more effective and targeted education interventions, as gaps and trends can more easily be recognized and addressed.

https://techchange-articulate.s3.amazonaws.com/DIAL/SDG%20Framework/v15/story_html5.html

Test Scope

The primary purpose of this test scope is to verify the Tangerine open source software functions effectively, accurately, and reliably. The tests will simulate real-world conditions and cover the application's core features to ensure that it can be used to assist in early grade education for assessments and research.

The most critical metrics to test are:

1. Verify all assessment related features are working as intended and tested on various Android devices with different specifications.
2. Test various assessment content with different project requirements
3. Ensure that the data collection, the reliability of the data on the servers, data consistency, and accuracy are working as intended.
4. Evaluate the user interface to ensure the ease of use in regards to the navigation and the user interactions.
5. Measure the speed of the data upload in various network speeds as it will be used in areas where the network coverage may deteriorate.
6. Offline functionality to ensure that the data collection and the assessment features are accurate and in sync once an internet connection is provided as there may not always be a reliable network connection in areas where it will be used.

Test Environment

Testers must ensure they have the latest software on their android and computer devices to test the software. An installation guide will be provided with the specific information required to set up the software for testing purposes.

Tests with various android devices specifications with different hardware, screen size, however, all must meet the minimum android requirements to run the software. A diverse range of network coverages will be used to simulate the internet connections when used in real-world applications. This includes a mix of fast and slow connections, different

generations of wireless mobile technology such as 3G, 4G, and 5G as well as the offline mode.

Testers are required to simulate real world scenarios where they will be responsible for data collection for students as well as run through the applications themselves to test how the software would operate from the student perspective. Various sample data sets which will replicate the assignment content that will be used in real-world applications will be provided to testers to ensure that the software will work with different forms of assessments.

Audience

The intended audience of this plan is the Tangerine software test team and developers. The software test team will verify, test, document, and report the functionality of the software. The test lead will be responsible for overseeing the testing effort and all the related documentation to ensure that all the components of the software have been appropriately rested.

Testing Approach

Our testing approach will utilise SpiraPlan to document all requirements traceability, test cases and runs, and results. SpiraPlan provides a full package solution for managing projects and the components related to it such as tasks, issues, releases and requirements Additionally, this software is designed specifically to support agile methodologies such as Scrum and Kanban that allows to track the requirements needed (define by user stories and acceptance criteria), tasks (work that developers do) for a given sprint (a given amount of time to complete a set amount of work to be completed).
https://elearningindustry.com/directory/elearning-software/spiraplan/features

SpiraPlan documents the software requirements for the project such as traceability to test cases, cycles, runs and results. Test cases, cycles and runs combined define the components that need to be tested in the software. SpiraPlan allows testers to record the actual results and allows these results to be traced to the linking requirements. Furthermore, the test process can be easily managed as the status of the tests can be seen (through the percentage of tests completed represented as a bar) and there are tools that provide the data to be exported to various options such as a PDF file or a Word document.
https://www.inflectra.com/SpiraPlan/

SpiraPlan contains all the necessary documents for test documentation such as the test reports, progress reports and software requirement documents which allows ease of access in terms of managing the testing procedures. The testing approach for this test plan is for independent testers to be assigned different sections of the software to test. The test lead is the point of contact for any testing issues. Furthermore, the test lead is responsible for the test planning of releases that are documented on SpiraPlan which include any test cases,

test sets, test runs and reporting the results. Consequently, all testers will report any failures during their tests on SpiraPlan.

Requirements-Based Testing

Requirements will be documented on SpiraPlan (as user stories and acceptance criteria) to allow traceability. As a result, it allows the requirements to be identified for test cases and test sets for the given requirements and the result to be tracked throughout the lifecycle of the software. The suitable tests that express the requirements will be incorporated in the test cycles and these can be done manually or through automation. In addition, the requirements will be tracked through SpiraPlan and the status of requirement-based testing will be updated throughout the testing period.

Manual Testing

Manual testing the Tangerine assessment tool will involve a myriad of approaches to ensure the software is functional for real world applications. Testers will be assigned to different components of the software to test. This entire process will track all the tests that are generated and executed. SpiraPlan will be used to store and overview all the manual test related content. If there are no suitable test cases for a specific feature or the case is lacking enough detail then a tester must inform this issue and assist the test lead and developer with fleshing out the test case. The manual testing of the software can be broken into two aspects, running the software and viewing the stored data and simulating the interaction of the software as a user would. Below is an example of what the manual tester may be tasked to complete:

Testers will start with the basic components including launching the app, creating an account, and subsequently logging in. Data collection and entry will be examined to ensure that the application is able to properly store assessment data and to verify that bugs are not present in an essential component. Inspection of the data analysis and review feature will be conducted to ensure that feature is able to accurately review the assessment data to identify the gaps and trends. Assessment creation will be tested to ensure that this process is seamless, error free, and verify whether users will be able to create a variety of assessment content to tailor to students. Editing assessment will be reviewed to review the changes that can be made to assessment and ensure that these changes are reflected. Manually inputting the assessment data will be conducted by the testers with a variety of inputs related to the content to determine whether the process for entering data is accurate and free from any errors. Additionally, testers must ensure that the data storage is accurate and consistently stored. Testers will verify this process by submitting the sample assessments to confirm that this information is properly transmitted and stored into the servers,

https://rmit.instructure.com/courses/107474/pages/api-testing-ci-slash-cd-testing?module_item_id=5170763

Automated Testing

Automated testing is a critical aspect of the testing process, ideally testing should be automated as it enforces consistency and efficiency. It is essential to ensure that the automated testing suite is in parallel with the current build or release of Tangerine. However, it is possible for the current testing suite to be behind the current version of the software as the requirements of the software may be reviewed or altered. The tester will initiate a test cycle against the latest build to verify if any changes made by the developers have impacted the test suite. Testers will be required to make necessary changes to ensure that the test suite can be successfully executed. While testing, the results will be stored with the provided test results directory that is located in the automated test suite. In the event that the test suite is not compatible with the latest build, a temporary repository will be used instead. Testing must be done with the official test directory to organise all the tests and ease of access for a better overview by the test team. These results will provide a better understanding of the test suites to identify whether tests passed or failed. If there are any failures or abnormal behaviour was encountered then this must be reported to examine and address these issues.

SpiraPlan will be used for the organisation and provide an overview of all the automated tests. Testers are required to connect their test suites to the test requirements to seamlessly examine the test results.

https://rmit.instructure.com/courses/107474/pages/api-testing-ci-slash-cd-testing?module_item_id=5170763

Regression Testing

Regression testing is a type of software testing that is performed to ensure that recent changes or additions to the software do not adversely affect its existing functionality. In the context of Tangerine, regression testing would serve as a critical mechanism to ensure that any modifications or updates to the software do not introduce unprecedented errors or compromise existing functionalities. The main goal here is to maintain the integrity of the application's core features while accommodating changes. The approach involves selecting a strategic subset of test cases that comprehensively cover crucial aspects such as assessment features, data collection processes, user interface elements, and fundamental user interactions.

Executing both manual and automated testing methods is recommended to ensure a robust evaluation. Manual testing would allow for an exploration of user experiences, while automated testing would bring efficiency and consistency, particularly when dealing with repetitive or large-scale testing scenarios. This combination helps verify the stability of the software under various conditions.

The ongoing process of test maintenance is vital in the context of regression testing. After each testing cycle, updates to test cases are necessary to accurately reflect changes in the application. Additionally, new test cases are introduced to accommodate any added features or capabilities. Keeping test data current would ensure alignment with alterations in data.

The core principle behind regression testing is to validate that previously validated functionalities still perform as intended. Any deviations detected during regression testing should be carefully documented and reported. This iterative process ensures that Tangerine's functionality remains intact while allowing for continuous improvement.

Test Maintenance

In order to maintain Tangerine's effectiveness and dependability, test maintenance is an essential component. As the software changes and as new requirements emerge, test maintenance entails updating and improving the test suite, test cases, and related documentation. Following are the test maintenance plan:

Review of Test Cases: Continually check whether the current test cases are still accurate and pertinent. Remove any test cases that are no longer relevant or necessary.
Update Test Cases: Test cases should be updated to account for software changes. The test procedures, expected outcomes, and any other pertinent information must all be updated. Test scenarios must faithfully reflect the application's current state.
Add New Test Cases: If Tangerine is given new features or capabilities, then add new test cases to cover those areas. Make certain that the test suite is thorough and addresses every aspect of the software.

Data Maintenance: Maintain test data so that it is up to date and pertinent. Update the test data as necessary if data structures or formats change.

Regression Testing: Conduct regression testing following updates to make sure that modifications did not adversely affect the functionality of the system. Running the current test suite as part of this ensures that no fresh flaws have been introduced.

Documentation Updates: Update all test-related documentation, such as test plans, test cases, and any traceability matrices, to reflect modifications made to the testing procedure.

Review and Approval: To maintain consistency and quality, make sure that the test lead or project manager reviews and approves each activity related to test maintenance.

As for managing test maintenance tasks, SpiraPlan will remain the main tool. The use of SpiraPlan by testers will allow them to track changes, update and maintain test cases, and keep test documentation up to date.

Item Pass/Fail Criteria

In the Tangerine testing framework, whether test cases pass or fail will be decided based on clear, specific and well-defined criteria. The pass criteria encompass various aspects critical to the application's effectiveness. Assessment features must operate without critical errors, and diverse assessment content with varying project requirements should be successfully tested. Data collection processes must be not only accurate but also reliable, with the data on servers demonstrating precision and consistency.

The user interface plays an integral role, and its success is measured by its ease of navigation as well as intuitiveness. User interactions should be smooth and free of errors to ensure a positive user experience. Furthermore, the speed of data uploads across varying network speeds is a significant metric, especially considering the application's potential use in areas with unreliable network coverage.

Offline functionality is a crucial aspect of Tangerine, and the software must demonstrate the ability to collect and assess data accurately even without an internet connection. Synchronisation should occur seamlessly once connectivity is restored.

Fail criteria include critical functionality failures, user interface issues affecting usability, data upload failures, and compatibility issues.

Test Deliverables

The following test documentations will be maintained and updated throughout testing to make sure activities can be traced back and progress speeds can be checked. They will also be submitted at the conclusion of the testing.

- Test Plan
- Test Cases
- Test Scripts
- Test Data
- Defect Reports
- Test Reports

Resources

Tools used in this test plan

Process	Tool
Automated tests and documentation	Spira

Browsers Used

Browser	Version
Chrome	Latest
Firefox	Latest
Safari	Latest

Devices Used

Device	Version
IPhone	Latest
Android phone	Latest
Local Computer	Running latest Windows update

Test Responsibilities

Test Lead:

Requirement Analysis:

- Understand and review the requirements, functionalities, and features of the Tangerine products.
- Collaborate with stakeholders to understand the business and technical requirements.

Test Strategy Formulation:

- Develop a test strategy that covers all the areas of the software, taking into account its usage in various countries, languages, and the specific methodologies of EGRA and EGMA.
- Determine the testing scope, objectives, and deliverables.

Test Planning:

- Define the timelines, resources, and test environment setup.
- Decide on the necessary tools and technologies to be used.

Test Environment Setup:

- Ensure the availability of devices running Android version 2.3 or higher with the recommended specifications.
- Setup network environments to mimic offline usage and subsequent data uploads when the internet becomes available.

Test Design:

- Design test cases, test scripts, and test data for various scenarios.
- Consider the multi-language aspect and ensure tests are designed for all supported languages.

Functional Testing:

- Test the core functionalities of Tangerine, Tangerine: Class, and Tangerine: Tutor.
- Ensure EGRA and EGMA assessments work as intended.

Usability Testing:

- Test the user interface and experience, especially considering the diverse user base in 65 countries.
- Validate the software's ease of use, especially for classroom observations, surveys, and assessments.

Performance Testing:

- Evaluate the application's performance under various conditions, such as high data volume uploads.
- Validate the software's responsiveness and stability on the recommended device specifications.

Security Testing:

- Assess data security, especially when uploading results to servers or when using RTI's server.
- Ensure that data stored in CSV files and any other format is secure from unauthorised access.

Bug Tracking and Reporting:

- Document all identified issues and communicate them effectively to the development team for resolution.
- Prioritise and manage bug fixes.

Required Staff/Testers:

Test Manager:

- Responsible for overseeing the entire testing process.
- Coordinates with stakeholders, assigns tasks, and ensures the timeline is being followed

Functional Testers:

- Focus on testing the core functionalities of Tangerine and its sub-products.
- Ensure EGRA and EGMA assessments function as intended.

Usability Tester:

- Evaluates user experience and interface.
- Tests the software from an end-user's perspective.

Performance Tester:

- Assess the software's performance, responsiveness, and stability.
- Uses tools and scripts to mimic high data volume scenarios.

Security Tester:

- Focuses on the software's security aspects, ensuring data protection and unauthorised access prevention.

Compatibility and Integration Tester:

- Validates the software on various Android devices and checks for integration of functionalities.

Test Analyst:

- Responsible for designing test cases, scripts, and data.
- Assists in bug tracking and documentation.

Training Needs/Requirements :

All team members must receive training on Tangerine to understand its features and potential use cases. It's important for our functional testers to have a solid understanding of the EGRA and EGMA methodologies, given the software's foundation on these principles. Depending on the selected tools for performance, security, and compatibility evaluations, it's essential that the corresponding staff members are proficiently trained. Considering Tangerine's support for over 100 languages, there's a need to train testers in localization and multilingual environments. Additionally, as the softwares offline capabilities with a subsequent online data upload process, our team should be well-equipped to test these features. Our security testers, in particular, should stay updated with contemporary security protocols to fortify the software against threats. If Tangerine incorporates CI/CD practices, it's vital for the team to know how to integrate their tests within this system.

Test Schedule

Test Phase	Start Date	End Date
Test Planning	2024-01-01	2024-01-05
Requirements Testing	2024-01-06	2024-01-12
Manual Testing	2024-01-13	2024-01-19
Automated Testing	2024-01-20	2024-01-26
Regression Testing	2024-01-27	2024-02-02
Test Report Generation	2024-02-03	2024-02-07

Risks and Contingencies

The following risks may cause issues during testing:

Risk or contingency	**Fix**
Changes to the program that are not reported to the testing team	Make sure all changes to the program are discussed with the testing team as soon as possible
Changes to requirements without discussing with the testing team	Make sure any changes to requirements are discussed with the testing team first
Delays in training personnel on testing	Make sure personnel is trained fast and properly as early as possible
Lack of availability of required software, hardware or tools	Make sure all required resources are collected early
Changes to timeline without discussion with the testing team leading to rushed testing.	Make sure any changes to the timeline are discussed with the testing team before hand

Test reporting and requirements

The reporting and requirements management within the Tangerine testing framework rely on the integration of SpiraPlan, as it serves as the repository for documenting requirements, test cases, test runs, and results. Regular test reports generated within SpiraPlan would provide detailed insights into the status of both manual and automated test cases.

SpiraPlan's support for agile methodologies enhances the efficiency of requirements tracking and task management. The tool's export functionality facilitates the generation of reports in various formats allowing for efficient documentation.

Clear traceability from requirements to test cases is an important factor in this case. SpiraPlan would ensure that this traceability is maintained throughout the testing lifecycle, maintaining transparency and accountability. This would allow for a centralised overview of all tests and their corresponding statuses, streamlining the monitoring and reporting processes. SpiraPlan’s comprehensive capabilities contribute to the overall effectiveness of test reporting and requirements management within the Tangerine testing environment.