**TEST PLAN**

**Group 1**

| **Mobile App Testing** | Version: **1.0** |
| --- | --- |
| **Test Plan** | Date: **13/01/2024** |
| Prepared by: **Group 1** | |

# Revision History

| **Date** | **Version** | **Author** | **Description** |
| --- | --- | --- | --- |
| 13/01/2024 | 1.0 | Lê Minh Vương | Prepare for I |
| 13/01/2024 | 1.0 | Hồ Trọng Nghĩa | Prepare for II |
| 13/01/2024 | 1.0 | Nguyễn Hữu Đức Hiếu | Prepare for III |
| 13/01/2024 | 1.0 | Nguyễn Thị Thúy | Prepare for IV |
| 13/01/2024 | 1.0 | Đinh Quốc Chương | Prepare for V |
| 13/01/2024 | 1.0 | Đinh Quốc Chương | Prepare for VI |
| 13/01/2024 | 1.0 | Hồ Trọng Nghĩa | Prepare for VII |
| 13/01/2024 | 1.0 | Nguyễn Thị Thúy | Prepare for VIII |
| 13/01/2024 | 1.0 | Nguyễn Hữu Đức Hiếu | Prepare for IX |

# Table of Contents

[**Revision History 1**](#_m8kamfke9e8o)

[**Table of Contents 2**](#_rn8owo6c2dbx)

[**I. Introduction 3**](#_qzyb7b4hyp7h)

[**II. Test Strategy 3**](#_81gtzsed9rcc)

[**III. Test Planning 4**](#_qbwsml816saa)

[**IV. Test Design 5**](#_29j0rb2lmaqs)

[**V. Test Execution 7**](#_bnch1up775te)

[**VI. Test Closure 8**](#_sc65gpzdivu1)

[**VII. Risk and Contingencies 9**](#_z5jyrw4osvph)

[**VIII. Approval 10**](#_fkpx2ndm42cc)

[**IX. Appendices 10**](#_a1kfvd67819d)

# Introduction

* Purpose: To ensure the functionality, usability and security of the mobile application.
* Scope: The testing will cover:
  + Task management.
  + Priority management.
  + Notify to the user.

# Test Strategy

* Testing Levels:
* Unit Testing:
* Verify each module/component independently.
* Validate the functionality of individual units.
* Integration Testing:
* Ensure that the modules work together as expected.
* Test the data flow and communication between different components.
* System Testing:
* Verify the overall functionality of the system.
* Check the interactions between different systems.
* Acceptance Testing:
* Validate if the app meets the specified requirements.
* Ensure that the app fulfils business use cases.
* Testing Types:
* Cross-OS Testing:
* Test the app on different mobile operating systems (iOS, Android).
* Ensure consistent functionality across various OS versions.
* Functional Testing:
* Validate the creation of tasks, setting priorities, and receiving notifications.
* Ensure that all features work according to specifications.
* GUI Testing:
* Verify the app's Graphical User Interface (GUI) for consistency and responsiveness.
* Confirm that all UI elements are functioning correctly.
* Usability Testing:
* Evaluate the app's user-friendliness.
* Test if users can easily navigate, create tasks, and set priorities.
* Accessibility Testing:
* Ensure the app is accessible to users with disabilities.
* Test screen readers, voice commands, and other accessibility features.
* Security Testing:
* Check for vulnerabilities in data storage and transmission.
* Verify secure authentication and authorization mechanisms.
* Database Testing:
* Validate the app's interactions with the database.
* Confirm data integrity and accuracy.
* Performance Testing:
* Evaluate the app's response time under different load conditions.
* Check for memory leaks and overall performance efficiency.
* Test Environment:
* Devices:
* Test on a variety of mobile devices to ensure compatibility.
* Include a mix of popular devices and different screen sizes.
* Operating System:
* Test on the latest versions of iOS and Android.
* Include testing on different OS configurations.
* Network Conditions:
* Test the app's performance under various network conditions (3G, 4G, Wi-Fi).
* Check if the app functions seamlessly with intermittent connectivity.
* Test Data:
* Create a diverse set of test data to cover different scenarios.
* Include edge cases and boundary values in the test data.

# Test Planning

1. **Test Deliverables:**

* Test Plan Document: Detailed document outlining the testing strategy, scope, resources, and schedule.
* Requirement Traceability Matrix : Mapping of test cases to the corresponding requirements.
* Test Data: Sets of input data for various test scenarios.
* Test Environment Setup: Documentation specifying the required hardware, software, and network configurations.

1. **Test :**

* Unit Testing: Individual components are tested in isolation.
* Integration Testing: Verify the interaction between integrated components.
* System Testing: Validate the entire system against specified requirements.
* Acceptance Testing: Ensure the system meets user acceptance criteria.

1. **Test Schedule:**

- Test Planning Phase (Duration: x days):

* Define testing objectives, scope, and strategy.
* Identify resources and allocate responsibilities.
* Develop a high-level test schedule.

- Test Design Phase (Duration: x days):

* Create detailed test cases based on requirements.
* Develop test data sets.
* Review and finalize the test plan.

- Test Execution Phase (Duration: x days):

* Execute unit tests after each component is developed.
* Conduct integration tests to verify component interactions.
* Execute system tests covering end-to-end scenarios.
* Perform acceptance tests with user involvement.

- Test Closure Phase (Duration: x days):

* Analyze test results and document defects.
* Prepare a test summary report.
* Conduct a lessons learned session.

1. **Resource Planning:**

* Test Manager: Responsible for overall test strategy, planning, and coordination.
* Test Engineers: Develop test cases, execute tests, and report defects.
* Developers: Collaborate with testers during unit and integration testing.
* System Administrators: Set up and maintain the test environment.
* Users: Participate in acceptance testing and provide feedback.

# Test Design

* Test Scenarios:

1. User Login:

* Goal: Ensure users can log in successfully.
* Scenario:
* Enter valid login information.
* Enter invalid login information.
* Check out the "Forgot Password" feature.

1. Create New Task:

* Goal: Ensure users can create new tasks simply and effectively.
* Scenario:
* Create tasks with valid information.
* Check for task creation with invalid data.

1. Set priority for each task:

* Goal: Test your ability to Set priority for each task.
* Scenario:
* Check the priority set for tasks.
* Check if priority tasks are in the right place.

1. Receive notifications:

* Goal: Ensure users receive the correct notifications.
* Scenario:
* Create an important task and check for notifications.
* Check for not receiving notifications.

1. Compatible with Multiple Devices and Operating Systems:

* Goal: Ensure the application runs smoothly across multiple devices and operating systems.
* Scenario:
* Test on mobile phones.
* Tested on iOS, Android operating systems.

1. Easy to Use without Detailed Instructions:

* Goal: Make sure users can use the application without detailed instructions.
* Scenario:
* Test the intuitiveness of the user interface.
* Make sure there's online support or simple instructions.

1. Security of User Information:

* Goal: Protect users' personal and work information.
* Scenario:
* Check the security of login data.
* Check the task information storage process.

1. Supports Multiple Languages and Regions:

* Goal: Make sure your app supports multiple languages and regions.
* Scenario:
* Switch language and check for correct display.
* Check multilingualism in notifications.
* Test Cases: Detailed cases for login, execute test cases for task creation, priority setting, and notification delivery.

1. Test Login:

* Valid: Enter correct username and password.
* Invalid: Enter incorrect username or password.

1. Test Create Task:

* Valid: Enter complete information and check if the task has been created successfully.
* Invalid: Leave the required information field blank (Title/content).

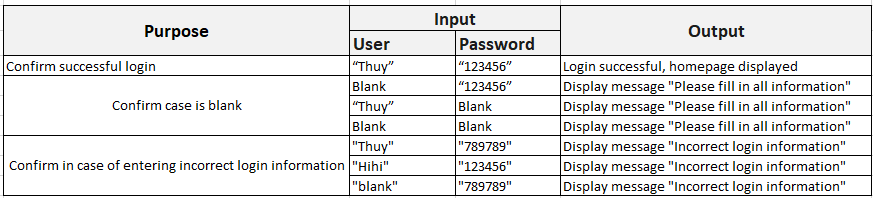
1. Test set priority:

* Valid: Choose a priority for the task and save.
* Invalid: No priority level selected.

1. Test get notifications:

* Check notifications when there's an important task.
* Test Data: Valid and invalid user credentials, task data and receive notifications methods.

*Ví dụ: Test design cho chức năng login*

**

# Test Execution

* **Test Execution Schedule:** Start Date - dd/mm/yyyy, End Date - dd/mm/yyyy

| **Activity** | **Start Date** | **End Date** | **Duration** |
| --- | --- | --- | --- |
| Test Planning | dd/mm/yyyy | dd/mm/yyyy | x days |
| Test Design | dd/mm/yyyy | dd/mm/yyyy | x days |
| Cross-OS Testing   * Android Device * iOS Device * Compatibility Testing | dd/mm/yyyy | dd/mm/yyyy | x days |
| Functional Testing   * Creating tasks * Setting priorities * Receiving notifications * Login & Register | dd/mm/yyyy | dd/mm/yyyy | x days |
| GUI Testing | dd/mm/yyyy | dd/mm/yyyy | x days |
| Usability Testing | dd/mm/yyyy | dd/mm/yyyy | x days |
| Accessibility Testing | dd/mm/yyyy | dd/mm/yyyy | x days |
| Security Testing | dd/mm/yyyy | dd/mm/yyyy | x days |
| Database Testing | dd/mm/yyyy | dd/mm/yyyy | x days |
| Performance Testing | dd/mm/yyyy | dd/mm/yyyy | x days |
| Test Reporting | dd/mm/yyyy | dd/mm/yyyy | x days |
| **Total** | | | **n days** |

* **Test Execution Environment:** Test on mobile devices including Android and IOS.
* **Defect Reporting:** The entire process of mobile app testing process should be recorded and double checked before crafting the final report. Testers must submit the most ideal bug report that includes all the information which helps developer to be able to fix the bug and then finally close the bug. The bug report should include the following:
* Types or levels of testing performed
* Time taken to complete tests
* Incidence reports and the stats derived from them.
* Overall quality of mobile application. Include key points describing whether the mobile application is fit for use, whether it meets acceptance criteria.

→ Submit accurate and complete bug report using bug tracking & reporting tools like Jira, Mantis, etc.,

# Test Closure

* **Criteria for Test Completion:**
* The application should meet the requirements specified by the stakeholders.
* The application should be tested for its functions of creating tasks, setting priorities, and receiving notifications., usability, performance,..
* The application should be tested on multiple devices with different screen sizes, operating systems, and hardware.
* The application should be tested for cross-platform compatibility.
* The application should be tested for application security vulnerabilities.
* The application should be tested for performance to ensure speed, scalability, and stability.
* The application should be tested for accessibility.
* **Test Summary Report:** The following is a summary of the simulation test results::

| **Type of Testing** | **Number of Test Cases** | **Number of Defects** | **Description of Defects** | **Found Status** |
| --- | --- | --- | --- | --- |
| Cross-OS Testing   * Android Device * iOS Device * Compatibility Testing | X | Y |  | Passed / Failed |
| Functional Testing   * Creating tasks * Setting priorities * Receiving notifications * Login & Register | X | Y |  | Passed / Failed |
| GUI Testing | X | Y |  | Passed / Failed |
| Usability Testing | X | Y |  | Passed / Failed |
| Accessibility Testing | X | Y |  | Passed / Failed |
| Security Testing | X | Y |  | Passed / Failed |
| Database Testing | X | Y |  | Passed / Failed |
| Performance Testing | X | Y |  | Passed / Failed |

* Issues:
* Functional do not work smoothly
* Compatibility issues
* Usability issues
* Performance issues
* Security issues
* Cross-platform compatibility
* Accessibility issues
* Recommendations: Based on the test results, the following recommendations are made:
* Fix the defects found during testing.
* Improve the performance of the application by optimizing the code.
* Enhance the user interface to improve usability.

# Risk and Contingencies

* Identify Risks:
* Device Fragmentation:
* Risk: The mobile app may behave differently on various devices due to the diversity of Android and iOS devices.
* Impact: Inconsistencies in user experience, functionality, or performance.
* Probability: High.
* Contingency Plan: Implement device-specific testing, prioritising popular devices and leveraging emulator testing to cover a broader range. Regularly update the device matrix based on user analytics.
* Data Security Breach:
* Risk: Potential vulnerability leading to a breach of user data during transmission or storage.
* Impact: Compromised user privacy and loss of trust in the app.
* Probability: Medium.
* Contingency Plan: Conduct thorough security testing, implement encryption standards, and collaborate with cybersecurity experts for penetration testing. Swiftly address and patch any identified vulnerabilities.
* Unpredictable Network Conditions:
* Risk: The app may not perform optimally under varying network conditions, leading to potential usability issues.
* Impact: Degraded user experience and dissatisfaction.
* Probability: Medium.
* Contingency Plan: Implement network simulation tools during testing to replicate different network scenarios. Optimise the app's performance by employing techniques such as caching and offline functionality.
* Contingency Plans:
* Device Fragmentation:
* Contingency Action: Regularly update the test device matrix based on market share and user analytics.
* Responsibility: Testing team.
* Timeline: Monthly reviews and updates.
* Data Security Breach:
* Contingency Action: Engage with cybersecurity experts for regular penetration testing and implement security patches promptly.
* Responsibility: Security team and development team.
* Timeline: Immediate action upon identification of vulnerabilities.
* Unpredictable Network Conditions:
* Contingency Action: Use network simulation tools during testing to replicate various conditions and optimise the app for performance.
* Responsibility: Testing and development teams.
* Timeline: Continuous monitoring and optimization during the testing phase.

# Approval

* Approvals: QA Lead, Project Manager.

# Appendices

* **Glossary:**
* Definitions of key terms and acronyms used in the test plan to ensure clarity and understanding.

| **#** | **Glossary** | **Meaning** |
| --- | --- | --- |
| 1 | PM | Project Manager |
| 2 | QA | Quality Assurance |
| 3 | GUI | Graphical User Interface |

* **References:**
* Requirements Document: The document outlining the functional and non-functional requirements to be tested.
* Design Specification: Reference to the system design specifications for a better understanding of the architecture.
* Test Strategy Document: Overview of the testing approach, objectives, and scope.
* Project Schedule: A schedule outlining project milestones, including testing phases and deadlines.