VOTAFUN

TEST PLAN

Version 1.1
01/11/2023

REVISION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Ng Yue Jie Alphaeus	26/10/2023	Ng Yue Jie Alphaeus	26/10/2023	Initial test plan draft
1.1	Roy Lau Run-Xuan	26/10/2023	Ng Yue Jie Alphaeus	01/11/2023	Addition of relevant sections

TABLE OF CONTENTS

1 Test Plan Identifier

This document refers to VotaFun test plan, which is currently version 1.1. The test plan has documented the general steps involved in the testing process, as well as the items that the QA team will test.

2 Introduction

The test plan provides an overview of the testing approach, strategy, schedule, tasks and deliverables for the testing process of VotaFun.

The objectives of this test plan are to

- Identify features of VotaFun to test.
- Describe our testing approach and techniques used
- Describe the criteria for when test cases fail or pass
- Describe the testing tasks the QA team will perform
- Describe the deliverables at the end of the testing phase
- Identify the resources and effort needed, the schedule for testing and any risks for testing
- Outline the responsibilities of each team member during testing

The QA team will perform four different types of tests. They are Unit Testing, Integration Testing, System Testing and User Acceptance Testing. The QA team will be using the white box approach for testing.

3 Test Items

The following lists the different items of VotaFun that the QA team will test.

3.1 Unit Testing

The team will perform unit testing for individual components of VotaFun in isolation. The QA team will perform unit testing automatically using test scripts by setting up a CI/CD pipeline during the development phase. This testing will verify that individual components of VotaFun work as intended.

3.1.1 Database

3.1.1.1 Manipulate application data

The database must be able to manipulate VotaFun's application data, such as room, question, option and user data. Manipulation refers to the following cases:

- Create a new room, question, option or user
- Sets the room properties
- Change a room host
- Add a user to a room
- Remove a user from a room
- Add a vote to an option

3.1.1.2 Retrieve application data

The database must be able to retrieve VotaFun's application data, such as the room, question, option and user data.

3.1.1.3 Store application data

The database must be able to store VotaFun's application data, such as the room, question, option and user data. Storing of application data can be done after any manipulation of the application data.

3.1.1.4 Format application data

After retrieving VotaFun's application data, such as the room, question, option and user data, the database must be able to format the data into a dictionary.

3.1.2 LLM

3.1.2.1 Extracting information

The LLM component must be able to extract the question, options or activities, given a properly formatted question and options or activities.

3.1.2.2 Generate reply

The LLM component must be able to generate a formatted reply to be sent to ChatGPT, given the past questions and prompts.

3.1.2.3 Retry logic

The LLM component must run the retry logic if there are fewer than two options for a question.

3.1.3 Frontend

3.1.3.1 Room creation

The Frontend must be able to navigate to the room creation page from the landing page by clicking on the 'Create Room' button, the component must then be able to redirect the browser to the 'Join Room' page.

3.1.3.2 Join room

The Frontend component must be able to navigate to the room creation page from the landing page. The Join Room button will only be enabled once the Room ID has been keyed in, after which the Join Room' button can be clicked, the component must then be able to redirect the browser to the 'Join Room' page. Within the 'Join Room' page, the frontend component must then be able to navigate to the 'Lobby Page' by entering the username details and then clicking on the 'Join' button.

3.1.3.3 Kick user

The frontend component must be able to kick any user in the Lobby or during Session if the current user is the host. The kick icon must appear for every participant in the room excluding themselves. Clicking on the kick icon will boot the target user from the lobby and their username will no longer be shown in the participant view.

3.1.3.4 Vote options

The frontend component must be able to display 4 options to vote for during a voting round. These options should appear without any abnormalities (eg. Overflowing text)

3.1.3.5 Set room properties

The Frontend component must be able to configure the room properties in the ‘Lobbu Page’ if the current user is the host. The frontend component should be able to modify the Room Activity dropdown, the Room Location dropdown and also the Number of Users. Any changes to these properties must be reflected to the Participants in the Room.

3.1.3.6 Game round

The Frontend component must be able to start a countdown timer of 15 seconds upon the start of a game round. Once the timer has reached 0, it should stop and the component should display a loading indicator signifying that the next prompt is being generated. While the voting round is ongoing, the user is able to vote on any of the 4 options. After making a vote, all of the options will be grayed out signifying that the user has already voted.

3.2 Integration Testing

For integration testing, the different components of VotaFun will be combined and tested together. Integration testing tests the functionality of VotaFun and ensures that all components can work together and as intended.

3.2.1 Vote options

Users must be able to vote for an option at each round. After the user has voted, all the options will be disabled.

3.2.2 Create room

The room host must be able to create a room for other users to join. The application will present the room host with a shareable link for other users to join.

3.2.3 Join room

Users must be able to join a room using the URL from the room host.

3.2.4 Kick users

The room host must be able to kick users from the game lobby. The kicked user will return to the home page.

3.2.5 Set room properties

The room host must be able to set different room properties such as the room location, room activity and the maximum capacity of the room. Other participants are notified of the change in room location and room activity.

3.2.6 Sockets

The socket component must handle different functionality. These functionality includes:

- Connect (join) a room
- Disconnect (leave) a room
- Broadcast room properties change
- Broadcast room host change
- Handle users voting for an option

- Close room when there are 0 users

3.3 System Testing

For system testing, the QA team will test the entire VotaFun application together.

3.3.1 Game round

VotaFun will present users with a question and at least two options for the first five rounds. VotaFun will present users with four activities for the last round and the final activity with the most votes. The game must end within six rounds.

3.4 User Acceptance Testing

The project team will release a beta application of VotaFun to clients for testing. The quality manager will provide a series of tests and the reasons for that test case for the client. The quality manager will note the feedback received from the client to improve VotaFun and try to incorporate it into VotaFun before release.

4 Features To Be Tested

The following table lists the different features the QA team will test from the user's point of view.

Each feature has a risk level, and it can be either High, Medium or Low. High risks are features essential to the users, and the QA team will test these features first. Medium risks are features not as necessary to VotaFun, and the QA team will test these features after the testing of high risks features. Finally, features which are low risk are those that are not important, and the QA team will prioritise them last.

Features	Risk
Vote options	High
Game round	High
Room creation	High
Joining rooms	High
Kick users	Medium
Set room properties	High
Sockets	High

5 Features Not To Be Tested

The QA team will test all the features implemented in VotaFun.

6 Approach

The test approach specifies how the QA team will perform testing for VotaFun. This section outlines the overall testing techniques, objectives and the team approach to testing.

6.1 Objectives

The objective for testing is to ensure that the team finds defects in VotaFun before release and minimise the defect slippage ratio. It also helps the team validate that VotaFun has implemented all functionalities as written in the SRS.

6.2 Testing techniques

6.2.1 White box testing

Our team will use white box testing for VotaFun. More specifically, we are looking at code coverage. Code coverage helps the team to monitor which parts of the code have been executed by the tests and which are not.

There are different types of code coverage, like statement coverage, branch coverage and line coverage. Our team will look at statement coverage as the primary metric, which tells the team how many statements the automated test has covered during testing.

6.2.2 Automated testing

The developer team has set up a continuous integration/ continuous delivery (CI/CD) pipeline using Heroku. Whenever developers push new features or code, the CI/CD pipeline automatically runs these tests. Developers can see what tests pass and fail, allowing them to fix them before merging into the main branch.

The team will use automated testing frameworks to test frontend and backend components as much as possible. The backend components (database and LLM) will use Pytest as the testing framework, while the frontend components will use Jest. The QA team will also use Playwright to perform integration and system tests.

6.2.3 Regression testing

Regression testing involves re-running previously executed test cases to ensure that VotaFun works even if the developers modify any code. Our team performs automated complete regression testing with the CI/CD pipeline and re-runs all tests when code developers push code to GitHub.

Having these automated regression tests allows all code changes to be constantly validated. It is also more efficient and will speed up the testing process, allowing the team to catch issues quickly.

6.3 Testing steps for VotaFun

6.3.1 Test planning

The QA team will first define the testing objectives, approach, and scope. They will also identify the test deliverables and estimate the resources and effort needed for the testing process. Additionally, the QA team will also identify risks in testing as well as develop contingency plans. The QA team will document all these in the test plan.

6.3.2 Test design

In this task, the QA team will develop different test cases. Each test case must specify the execution steps, expected and actual results, inputs and any pre and post-conditions. If necessary, the QA team will also write test scripts during this step to

perform automated testing of components. The QA team must ensure that the test cases have a comprehensive coverage of the code.

6.3.3 Test execution

The QA engineers will first prepare the testing environment. Next, the QA team will follow the execution steps for each test case and compare the expected result and actual results. If test scripts are available, the QA team will run these test scripts as well.

The QA team must record the status of each test case if the test case fails or passes, as well as when the QA team performs the test. The QA team must also document the steps to reproduce the bug.

6.3.4 Report defects and perform corrective actions

The QA team will notify the developers of failed test cases. Developers are responsible for implementing the fix for each bug found.

Once the developers have fixed bugs found, the QA team will re-run those failed test cases and document the test result if it passes or fails. The QA team will perform regression testing as well to ensure that passed test cases do not fail.

6.3.5 Test reporting

The QA manager will create a report that summarises all testing activities done by the QA team. The report will include test results, defects found, defects status and so on. The project manager will approve the test report.

7 Item Pass/Fail Criteria

The QA engineers must follow the steps while executing each test case and compare the expected and actual results at each step. If the output is different from the expected results, the test case will be regarded as failed, and this shall be reported to the developer team. A test case is regarded as a pass only when the test output matches the expected results for all execution steps of the test.

Testing for VotaFun is regarded as complete when these criteria are met:

- All unit, integration and system testing are complete
- Client has provided feedback during the User Acceptance Test

8 Suspension Criteria And Resumption Requirements

8.1 Suspension Criteria

Suspension criteria are conditions when the QA engineers should stop any testing activities temporarily. Having these suspension criteria ensures that testing results are meaningful without any wasted effort from the QA team. Testing results are reliable, as the QA team performs testing in a controlled environment each time.

The following are cases where the QA engineers should stop testing temporarily:

- The test environment is found to be not stable.
- A critical defect is found in VotaFun.
- Required resources are not available (Database, Internet connection).
- Actual outputs differ from expected outputs during the execution of test cases.

Whenever the QA engineers suspend testing, they will document and notify the QA manager of the reason for suspension. The QA team can only resume testing once the resumption requirements are met.

8.2 Resumption Requirements

Resumption requirements are requirements to meet before the QA engineers can resume testing after a testing suspension. The QA engineers should wait for these requirements to be met before resuming testing as it ensures that testing does not resume too soon, which can lead to unreliable results.

The following are cases where testing can continue after the suspension criteria:

- Test environment is confirmed to be stable for testing.
- Critical defects found in VotaFun are fixed.
- Required resources are available and functioning as expected.
- Actual output is the same as the expected output during the execution of test cases.

Before the QA engineers resume testing, they must verify that the requirements are met and document them. The QA manager will approve the resumption of testing activities before the QA engineers can continue testing.

9 Test Deliverables

Test deliverables refer to documents or items created during the testing phase. These deliverables act as a record of the testing process, as well as the results of testing. The following are the different test deliverables for VotaFun.

9.1 Test plan

The test plan is a document that provides a high-level overview of VotaFun's testing process. It highlights the objectives, what features and items the QA team will test, the testing approach, the resources needed and the schedule.

9.2 Test cases

Test cases are a set of executable steps for a single test scenario. It contains a description of the test case, pre and post-conditions, steps to perform the test, expected and actual results, and the status of the test case.

9.3 Test report

The test report summarises the testing process, and outcomes of the tests.

9.4 Defect report

The QA team will document all bugs and issues found during the testing phase, the steps to reproduce the bugs and also the outcomes when the QA team re-runs the failed test. Developers will also document the corrective actions to rectify the bugs.

10 Testing Tasks

The QA team will complete the following testing tasks during the testing process for VotaFun:

- Test planning
- Identify features and items to test
- Create test cases and test scripts
- Conduct testing
- Create test report

11 Environmental Needs

The QA engineers must meet the following requirements or have the following software to test VotaFun:

- Internet connection
- git (version control)
- docker
- Integrated development environment (IDE)
- Linux, Mac OS or Windows desktop
- Internet browser (Firefox, Chrome, Safari)
- Test cases

12 Responsibilities

Role	Responsibilities
Project Manager	● Oversees and approves of the test plan. ● Priorities VotaFun's features the QA team will test. ● Monitor and collect information on the test results.
QA Manager	● Overseas the entire testing phase of the project. ● Draft the test plan, test cases and test report. ● Ensures that the developers fix bugs found during testing.
QA Engineer	● Perform testing on VotaFun. ● Report on bugs and issues found during testing.
Lead frontend/ backend developer	● Provide training on testing frameworks. ● Provide support for testing efforts.

13 Staffing And Training Needs

The QA manager has to be familiar with the test plan and the different test cases. They will be responsible for overseeing the testing phase and writing of the documentation. The QA engineers will be the ones performing the tests and will work together with the frontend and backend developers if there are any bugs or issues found. The lead developers will provide materials and training on the testing frameworks. Lead developers must also be familiar with VotaFun's software components.

The developers must train the QA engineers on how to set up VotaFun locally. The QA engineers also must be familiar with Python and TypeScript testing frameworks.

14 Schedule

VotaFun's testing phase will last three weeks, from 12 October 2023 to 2 November 2023. The following outlines the tasks the team will perform during these three weeks and the deadlines for each task.

14.1 Creating test cases

Due to the tight timeline in the testing phase, the QA team has developed some test cases ahead of schedule during the development phase. The QA team will also use this time to develop additional test cases to ensure that test cases cover all scenarios. The QA team will complete this task on 16 October 2023.

14.2 VotaFun testing

After the QA team has developed the test cases and the project manager approves the test cases, VotaFun's testing begins. The QA team will raise bugs and issues found to the developers. The developer team will fix any bugs or issues found. After the developers fix the bugs, the QA team will rerun failed test cases. Testing will conclude on 26 October 2023.

14.3 Documentation

The QA team will compile the test cases and test coverage reports. The testing results and the test report will be documented and checked into VotaFun's Mediawiki. The team will finish documentation on 2 November 2023.

15 Risks And Contingencies

The following are the risks and contingencies during testing for VotaFun.

Risk ID	RISK 10
Risk Description	There may be a lack of testing resources like testers that may hinder testing efforts.
Risk Response	Mitigate
Probability of Occurrence	20%
Impact Level	Medium
Contingencies	The developers will perform testing if needed, and the QA team will request a deadline extension if the timeline allows. The project manager will also allocate enough resources to the testing phase.

Risk ID	RISK 11
Risk Description	There may be incomplete testing. Test cases may not cover all scenarios, leading to undiscovered bugs and issues.
Risk Response	Mitigate
Probability of Occurrence	40%
Impact Level	High

Contingencies	The QA team must ensure that all test cases cover all use cases. The QA team must also design test cases that maximise test coverage.

Risk ID	RISK 12
Risk Description	There may be a tight development schedule for VotaFun, leaving little time for testing. A tight schedule may also lead to incomplete testing, as the QA team may not have time to run all test cases.
Risk Response	Mitigate
Probability of Occurrence	40%
Impact Level	Medium
Contingencies	The QA manager will develop test cases early in the project. For unit tests, the QA manager will design test cases for VotaFun's CI/CD pipeline. The QA team will also prioritise test cases and ensure that high risk features are tested first.

Risk ID	RISK 13
Risk Description	Human errors during testing may lead to wrong testing results. The QA manager may not properly design the test cases, or testers may not follow the execution steps in the test case.
Risk Response	Mitigate
Probability of Occurrence	30%
Impact Level	High
Contingencies	The QA manager will provide clear steps for executing test cases. The QA manager will also work with the developer team to ensure that the test cases designed will cover all scenarios.

16 Approvals

The project manager will approve the entire testing process for VotaFun, including the approval of the test cases designed by the QA manager. The QA manager will approve the testing results from the QA engineers and draft the test report from the testing results. Once the test report is drafted and approved by the project manager, the project manager will store the test report in VotaFun's MediaWiki.

17 References

The following table lists the various documents referenced in this test plan. The documents listed in the table can be found in VotaFun's MediaWiki.

Document Name	Issuance Date	Version
Project Proposal	7/9/2023	1.0
Use Case Model	7/9/2023	1.3
System Requirement Specification	21/9/2023	1.3
Quality Plan	21/9/2023	1.0
Test Cases and Requirements Test Coverage Report	01/11/2023	1.1