

FORWARD FOCUS

Test Plan Architecture

04/28/20

1. Introduction	2
1.1 Objective	
1.2 Purpose	
1.3 Scope	
1.4 Acronyms and Definition	
2. Features to be Tested	4
3. Features Not to be Tested	4
4. Testing Strategy	6
4.1 Testing Types	6
4.1.1 Data and Database Integrity Testing	
4.1.2 Unit Testing	
4.1.3 System/Integration Testing	
4.1.4 Acceptance Testing	
5. Approach	8
6. Item Pass/ Fail Criteria	10
6.1 Suspension Criteria	
6.2 Resumption Criteria	
6.3 Approval Criteria	
7. Testing Process	11
7.1 Test Deliverables	
7.2 Testing Tasks	
7.3 Responsibilities	
7.4 Scheduling	
8. Environmental Requirements	12
8.1 Hardware	
8.2 Software	
8.3 Security	
8.4 Tools	
8.5 Risks And Assumptions	
9. References	12

1.

This is the Master Test Plan for Forward Focus Project. This plan will address only those items and elements that are related to the Career Guidance process, both directly and indirectly affected elements will be addressed. The primary focus of this plan is to ensure that the new Forward Focus application provides the same level of information and detail as the current system while allowing for improvements and increases in data acquisition and level of details available (granularity).

The project will have three levels of testing:-

- 1) Unit Testing,
- 2) System/Integration Testing
- 3)Acceptance Testing

1.1 Objective

- Checks of the functions
- Interface testing and integration testing
- Plan a list that needs to be revised after testing

1.2 Purpose

This Test Plan document supports the following objectives:

- Identify existing project information and the software that should be tested.
- List the recommended test requirements (high level).
- Recommend and describe the testing strategies to be employed.
- Identify the required resources and provide an estimate of the test efforts.
- List the deliverable elements of the test activities.

1.3 Scope

It is critical that all system and subsystem interfaces be tested as well as system performance at this early stage. The interfaces between the following subsystems will be tested:

- 1) Student Login and Registration
- 2) Profile and Score Evaluation
- 3) University Suggestions

The details for each level are addressed in the approach section and will be further defined in the level specific plans.

1.4 Acronyms and Definition

Project – activities which will lead to Forward Focus System production.

Client – the individual or company that is constructing this Forward Focus System for.

User – the person or individuals who are actually going to communicate with the Forward Focus network.

Use case – defines a goal-oriented system-to-actor interaction. A case of use can define several variants, called scenarios, resulting in different paths through the use case and usually different results.

Scenario- one way through a case of usage

Actor- consumer or other device of software that derives interest from a case of use.

Developer- the person or organization that develops the system, sometimes also called the supplier.

Stakeholder- everyone has an interest in the project and its progress. This includes clients, consumers, users, creators, testers, executives and managers.

2. Features To Be Tested

This is a description of what is to be checked from the viewpoint of the USERS on what the program is doing. This is not a program technical definition but a view of the functions by USERS.

The interfaces between the subsystems to be tested are:

- Module 1: Login
- Module 2: Registration
- Module 3: Profile Evaluation
- Module 4: Score Evaluation
- Module 5: University Suggestion
- Module 6: Download Resource

The external interfaces to the following devices to be tested are:

- Local Computer
- Remote Computer

The most critical performance measures to be tested are:

- Response time for remote login to the system.
- Response time to access the Forward Focus software.
- Response time to access the Resources.
- Student response time when a system is loaded with multiple students.
- Student response time when simultaneous access to the resources provided.

3. Features Not to be Tested

It is a list of what is NOT to be checked from both the user's point of view on what the program is doing and a view of the management of configuration / version control. This is not a program technical overview but a view of the functions by USERS.

The features which are yet to be released are not tested in the current version of testing. If the feature has a very low risk and the deadline is due soon, such features should be given low priority and should not be tested in the current version.

Considering these factors , following are the features not to be tested in current version:

- Giving a mock live with the time constraints, provided by team ForwardFocus. This feature is to be included in the next version of software hence it should not be tested
- Logging into the system using third party applications.
- Live-Lectures by experts.
- Chat-Box 24x7 for doubts
- Post a doubt or a blog helpful for the community

These features are not tested because these features are going to be included in the next version of the software.

4. Testing Strategy

The Test Strategy presents the recommended approach to the testing of the software applications. The previous section on Features to be Tested described what will be tested; this describes how it will be tested.

The main considerations for the test strategy are the techniques to be used and the criterion for knowing when the testing is completed.

In addition to the considerations provided for each test below, testing should only be executed using known, controlled databases, in secured environments.

UNIT Testing will be done by the developer and will be approved by the development team leader. Proof of unit testing (test case list, sample output, data printouts, defect information) must be provided by the programmer to the team leader before unit testing will be accepted and passed on to the test person. All unit test information will also be provided to the test Person.

SYSTEM/INTEGRATION Testing will be performed by the test manager and development team leader with assistance from the individual developers as required. No specific test tools are available for this project. Programs will enter into System/Integration test after all critical defects have been corrected. A program may have up to two Major defects as long as they do not impede testing of the program (I.E. there is a work around for the error).

ACCEPTANCE Testing will be performed by the actual end users with the assistance of the test manager and development team leader. The acceptance test will be done in parallel with the existing manual process after completion of the System/Integration test process.

4.1 Testing Types

4.1.1 Data and Database Integrity Testing

The databases and the procedures in the database should be checked as different systems. These systems are to be checked without the applications (as the data interface). In order to identify the tools / techniques that may exist to support the tests identified below, further research on the DBMS needs to be carried out.

Test Objective: Ensure the methods and processes for accessing Database work properly and without corruption in data.

Technique:

- Invoke method and process of accessing each database, seeding each with valid and invalid data (or data requests).
- Inspect the database to ensure that the data is stored as planned, that all database events have occurred correctly, or check the returned data to ensure that the correct data has been retrieved (for the correct reasons)

Completion Criteria:

All methods and processes for accessing databases work as planned and without abuse in the data.

Special Consideration:

- Testing can require a DBMS development environment or drivers to enter or edit data in the databases directly.
- Processes should be hand invoked.
- Small or minimally sized databases (limited number of records) can be used to make any non-acceptable incidents more noticeable.

4.1.2 Unit Testing

User Interface testing verifies the relationship between a user and the device. UI Testing's purpose is to ensure that the User Interface provides the user with the correct access and navigation through the application functions. Furthermore, UI Testing ensures that artifacts within the UI function as intended and comply with corporate or industrial standards.

Test Objective:

Be sure to test the following:

- Application navigation represents business functions and specifications properly, including window to window, field to field, and access methods (tab keys, mouse movements, accelerator keys);
- Window objects and features, such as menus, size, position, state, and focus, comply with the standards.

Technique:

Create / modify checks for each window to check proper navigation and object states for each window and interface operation.

Completion Criteria:

Every window has been tested successfully to remain compatible with the benchmark version or within appropriate standards

Special Consideration

- Not all properties can be opened to custom and third party objects.

4.1.3 System/Integration Testing

Application research would concentrate on any target specifications that can be specifically related to usage cases (or business functions), and business laws. The purposes of these tests are to verify the proper acceptance, processing and retrieval of data, as well as the correct application of business rules. This type of testing is based on black box techniques, i.e. verification of the application (and its internal processes) by interacting with the application through the GUI and analyzing the output (results). An outline of the tests recommended for each application is identified below:

Test Objective:

Ensure proper search, data entry, collection, and retrieval of applications.

Technique:

- Use valid and invalid data to verify the following: Execute each use case, use case flow or function:
- The results expected occur when using valid data.
- When invalid data is used, the correct error / alert messages are displayed.
- Each business rule is applied appropriately.

Completion Criteria:

- All planned experiments were carried out.
- All defects found have been remedied.

4.1.4 Acceptance Testing

Acceptance testing, a research method conducted to assess whether or not the software program has met the requirements for the requirement. The main aim of this test is to determine the compliance of the program with the business specifications and to check if it has met the necessary standards for delivery to end users.

Test Objective:

The acceptance test cases are performed against the test data or using an acceptance test file, and the tests are then compared to the anticipated ones.

Technique:

The acceptance test is carried out as the last stage of pre-deployment testing. Alpha Testing is carried out by peers in the company itself as part of uncovering any missed errors or missing functions and is

typically a black-box test process. Beta research is carried out outside of the organisation. These involve typically consumer testing and end-user testing.

That includes general consumer research and end-user testing tests.

Completion Criteria:

- These users must check to verify fulfillment of non-technical specifications and effective integration of interfaces.

5. Approach

Bear in mind the degree on which the program can be managed and the sensitivity of certain errors that may involve uncalled mishaps. Since we choose feature driven development as our guiding process, we have identified some important modules to be built in iterations. Before integrating these modules, these modules can be checked separately for errors and results. Our strategy was with the modules checked first by a team member who planned the same and later by a different team member, and then change features by feedback. Those tests are called units. The research involved checking code and debugging some syntax and logical errors. Specific features like Login and Add New Student may not work efficiently if the database scales to practical if not configured numbers. Such features, modulated, continue to work with integration in its product, such as Online . These features are to be tested for effective and true system integration in system design.

For such unique experiments the research team consisted of already participating team members and peer teams. Unit and integration testing were carried out by dividing the team into developer and non-developer for each feature and both teams conducted detailed tests as described above. It has reported some non-technical reviews for the consumer Graphical Design and Configuration. Graphics and aesthetics not important in the field of work for a software, the team focused more on the development of an error-free and banal system.

6. Item Pass/Fail Criteria

If a certain consumer follows the instructions of the program and does not make a mistake when performing one's own functionalities, then it can be defined as a PASS criterion and an FAIL criterion results in the opposite of the information listed previously.

6.1 Suspension Criteria

- If the user enters a wrong username or password then the system suspends the username or password to a pop up representing the wrong username.
- When the student enters the correct username and password , and if the entered time is already gone then the system goes into suspended state until a correct delay time is given.
- Whenever there are clashes for assigned student profiles, a student may get a message to wait for a certain period of time due to high traffic. In such a case, that particular test session goes in a suspended state until the traffic settles down and that student is assigned a test.

6.2 Resumption Criteria

- After suspension from the login state, the system needs to resume from asking for a correct username password and thereby displaying the respective screens afterward.
- For the invalid updated delay time added, the system will again ask for a valid argument and then carry forward the whole procedure.
- After the test clashes are sorted out by the scheduler of the system, it takes the user to the final stage of approval for the given test.

6.3 Approval Criteria

- Firstly , the user must have the right username and password in order to be able to use the resources provided.

- Forward Focus staff has to enter valid student details while deleting a student from the given access.
- Updated Test Time should be after the original time and not a time that has already gone by.
- Thereafter, the test gets assigned to a particular student if and only if the status from the given student is 'Yes, Start'.
- After checking the valid scores and percentage of a student, then only the university list suggestion list must be popped up.

7. Testing Process

7.1 Test Deliverables

The deliverables from the test processes are documented from the test plan and minutes of meetings. Deliverables can be defined from the test case document in which the status of each and every test case changes to 'yes' or 'no' according to the results. The actual result column may also be viewed as deliverables for test case papers.

7.2 Testing Tasks

Duplicate classes are created before unit testing each and every feature of the project, so that they can be tested on another machine. For example, A class called 'Test' is duplicated and then moved to a different computer where it is independently tested. That makes the testing process much simpler and easier to work.

7.3 Responsibilities

The responsibilities were handed out across the group. Those who work on the software design were not interrupted in terms of checking the software 's functionality. Testing was performed internally by an individual outside the party, as well as externally. The group members tested the features that they did not have to construct so they can give a true review of the features tested. The documents were distributed equally among the group members as well as the coding procedure.

7.4 Schedule

Because we have introduced the software development process for User Driven Design, the testing is performed only when an user is completely designed and implemented in its proper

manner. After each feature 's production has finished, testing is performed individually on that feature. The average time needed to check each of the apps came out in about 5 minutes each.

8. Environmental Requirements

8.1 Hardware Requirements

- No specific hardware is needed, the software is built and tested on our personal computers.

8.2 Software Requirements

- Visual Code Studio
- Python(Django) compiler
- PostgreSQL
- Google Chrome(Web browser)
- Sublime Text

8.3 Security

- The testing is performed on a different system. Backup is taken of the previous stable versions hence malfunctioning can be avoided hence if any type of error is detected or the version fails the test cases, the team can refer to previous versions and proceed further.

8.4 Tools

- **Visual Code Studio:** The tool was used to develop the software as well as for testing. The tool was used to develop and test the django code. This software prints errors with the line numbers and very useful suggestions which makes testing easier.
- **PostgreSQL:** The tool was used to enter data and view data sets and fetch entries.
- **Google Chrome :** The tool was used to view all HTML-CSS pages and make changes accordingly.
- **Sublime Text:** The tool was used to develop HTML pages.

9. References

1. Jmpovedar.files.wordpress.com. 2020. [online] Available at: <<https://jmpovedar.files.wordpress.com/2014/03/ieee-829.pdf>> [Accessed 15 May 2020].
2. Eriksson, U., 2020. *Test Plan Document, Attributes Of Test Plan Document With Example*. [online] ReQtest. Available at: <<https://reqtest.com/testing-blog/sample-test-plan/>> [Accessed 15 May 2020].
3. Johnson, C., 2020. *The Complete Guide To Writing Test Strategy [Sample Test Strategy Document]*. [online] Software Testing Material. Available at: <<https://www.softwaretestingmaterial.com/test-strategy/>> [Accessed 15 May 2020].