**CHAPTER 5: Software Testing Document**

5.1. Introduction

5.1.1. Purpose

The primary purpose of this chapter is the process to check whether the software is defect-free or not. It is the process of verification and validation of software service or application by checking whether it is meeting the user requirements and what all is implemented as per the characteristics. Software testing plays a vital role in the process of developing a high quality software. Testing is necessary because we all make mistakes. Some of those mistakes are unimportant but some of them are expensive and dangerous. It contains the following sections:

* Scope of Testing
* Testing Tool and Environment
* Resources and responsibilities
* Test strategy: Test approach, test stage
* Test schedule
* Feature to be tested
* Feature not to be tested
* Defect Log
* Test report

5.1.2. Scope of testing

* There are four main stages of testing that need to be completed before a program can be cleared for use: unit testing, integration testing, system testing, and acceptance testing.

|  |  |  |
| --- | --- | --- |
| ID | Test Stages | Description |
| 1 | Unit testing | During this first round of testing, the program is submitted to assessments that focus on specific units or components of the software to determine whether each one is fully functional. The main aim of this endeavor is to determine whether the application functions as designed. In this phase, a unit can refer to a function, individual program or even a procedure, and a White-box Testing method is usually used to get the job done. One of the biggest benefits of this testing phase is that it can be run every time a piece of code is changed, allowing issues to be resolved as quickly as possible. It’s quite common for software developers to perform unit tests before delivering software to testers for formal testing. |
| 2 | Integration testing | Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions. This is particularly beneficial because it determines how efficiently the units are running together. Keep in mind that no matter how efficiently each unit is running, if they aren’t properly integrated, it will affect the functionality of the software program. In order to run these types of tests, individuals can make use of various testing methods, but the specific method that will be used to get the job done will depend greatly on the way in which the units are defined. |
| 3 | System testing | System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets Quality Standards. System testing is undertaken by independent testers who haven’t played a role in developing the program. This testing is performed in an environment that closely mirrors production. System Testing is very important because it verifies that the application meets the technical, functional, and business requirements that were set by the customer. |
| 4 | Acceptance testing | The final level, Acceptance testing (or User Acceptance Testing), is conducted to determine whether the system is ready for release. During the Software development life cycle, requirements changes can sometimes be misinterpreted in a fashion that does not meet the intended needs of the users. During this final phase, the user will test the system to find out whether the application meets their business’ needs. Once this process has been completed and the software has passed, the program will then be delivered to production. |

* Type of testing

The following type of testing is used in this project are:

* + GUI test
  + Function test
  + Regression test
  + Acceptance test include alpha test
  + Unit test

5.1.3. Range of testing

Team performs all functions defined in the SRS based on the approved version.

5.2.1 Testing Tools and Environment

**5.2.1.1. Testing tools**

* **Front-end and Project Testing**
  + Chrome Developer Tool: To view logs.
  + Trello: Manage bug.
  + Microsoft Excel, Git: Manage testcase
* **API testing**
  + PostMan: to manage the list of all APIs and manually test APIs’ result
* **Unit test**
  + Jest: Unit test for JavaScripts
* **UI testing**
  + React-Native-CLI client: View app when development

**5.2.1.2. Testing environment**

* **Device 1: Laptop Dell** 
  + CPU: i7 7300hq
  + Ram: 8gb
  + Gpu: gtx1050ti
* **Device 2: Laptop HP**
  + CPU: i7 10510U
  + Ram: 8gb
  + Gpu: Geforce MX250
* **Device 3: Laptop Asus**
  + CPU: i7 9750H
  + Ram: 8gb
  + Gpu: GTX1650
* **Other device: Mobile phone**
  + OPPO F7
  + Samsung Galaxy A21s
  + Samsung Galaxy A80
  + OPPO A52

5.2.2. Resources and Responsibilities

|  |  |  |
| --- | --- | --- |
| ID | Resources | Responsibilities |
| 1 | Project Manager | - Responsible for project schedule and overall success of the project  - Review test-case and report |
| 2 | Tester | - Preforming the actual system testing  - Manage test resource and assign test tasks  - Create test plan  - Create test cases  - Create test report  - Execute test  - Test log report |
| 3 | Developer | - Create unit test and integration test scripts  - Fix bugs |

5.2.3. Test Strategy

**5.2.3.1. Test Model**

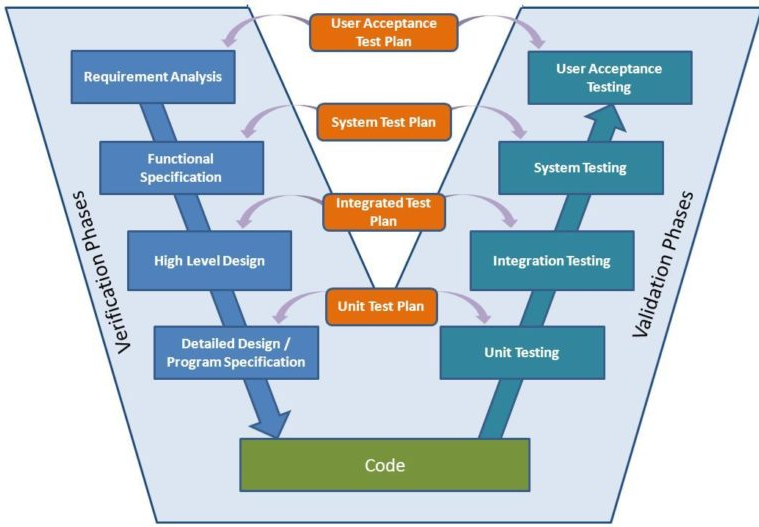


Figure 94 - V-model

Overall, we choose V-Model to implement testing process. With V-Model, software development is separated into two appropriate phase’s groups: development and testing. In this model, the verification and validation will be done side by side. It emphasizes the strict process flow to develop a quality product. The errors occurred in any phase will be corrected in that phase. Proactive defect tracking defects, which are found at early stages even, may be in the development phase before application being tested.

Pet Dating API has 2 levels of test:

* Unit testing: Automation tests that cover logic of Models and Libraries
* API testing: Automation tests that involve testing APIs directly (in isolation) to determine whether APIs return the correct response (in the expected format) for a broad range of feasible requests, react properly to edge cases such as failures and unexpected/extreme inputs.
* Pet Dating Front-end works mostly with GUI instead of logic and it depends on Pet Dating API, so that Pet Dating Front-end apply system testing which coverage of whole Pet Dating system.

**5.2.3.2. Test types**

The following type of testing is used in this project are:

* + - Unit test:

**UNIT TESTING** is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class.

* Unit test also includes database testing to verify constraint, transaction, default value, data types, data format, and check null and junk characters which are mentioned in database design and software requirement.
* Test case will have to cover all logic branch that function or method could execute with difference data input. Another alternative logic branch should be covered if not, that logic branch should be detected at API testing level.
* Implemented function's error message and database error message will be included in this test.
  + - API test:

**API TESTING** is a software testing type that validates Application Programming Interfaces (APIs). The purpose of API Testing is to check the functionality, reliability, performance, and security of the programming interfaces. In API Testing, instead of using standard user inputs (keyboard) and outputs, you use software to send calls to the API, get output, and note down the system's response. API tests are very different from GUI Tests and won't concentrate on the look and feel of an application. It mainly concentrates on the business logic layer of the software architecture.

* Involves testing APIs directly to determine if they meet expectations for functionality, reliability, performance, and security. API testing will test all of individual implemented API of PetDating API.
* Test case will verify constraint of data which be mention in Business rule
* Basically, almost all API test cases are executed as automation test. After that all API with standard sample datasets will be saved and confirmation tests will be executed by using Postman with developer's local database.
  + - UI test:

**Graphical User Interface testing** is mainly about ensuring that the UI functions in the right, an app follows its written specifications and that defects are identified. Other than that, we check that the design elements are good. This involves checking the screens with the controls like menu bars, toolbars, color, fonts, sizes, icons, content, buttons, etc. How do they respond to user input? GUI test will be performed fully on all screens.

* This test targets to cover the verification of the overall look and feel of the OCFU system including initial position, font, text size, color, focus, initial button, tab order, label, screen sizes and sentences width.
* Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, it must be able to provide inputs to the input fields.
* Check if Error Messages are displayed correctly.
* Check if Font used in application is readable.
* Check if the alignment of the text is proper.
* Check if the Color of the font and warning messages is aesthetically pleasing.
* Check if images have good clarity.
* Check if images are properly aligned.
* Check the positioning of GUI elements for different screen resolution.
  + - Regression testing

**Regression Testing** is a type of testing that is done to verify that a code change in the software does not impact the existing functionality of the product. This is to make sure the product works fine with new functionality, bug fixes or any change in the existing feature. Previously executed test cases are re-executed in order to verify the impact of change.

* + - Acceptance testing
      * This test type will be executed by tester with designed test cases, acceptance test is a test type conducted to determine if the requirements of a specification or contract are met
      * It also includes alpha testing; alpha testing takes place at close relation user’s site and are free test to detect bug and strange behavior. By that, development team will improve UX and UI of system

**5.2.3.3. Test stage**

The table below describes the stages in which common tests are executed:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type of test | Stage of test | | | |
| Unit test | Integration test | System test | Acceptance test |
| Unit tests | x | x | x |  |
| API tests | x | x |  |  |
| UI tests |  |  | x | x |
| Regression tests | x | x | x | x |
| Acceptance tests |  |  | x | x |

**5.2.3.4. Test schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| Task | | Start date | End date |
| Phase 1 | Create test plan | 10/06/2010 | 11/06/2020 |
| Create and execute unit test | 12/06/2020 | 28/06/2020 |
| Create and execute integration test | 12/06/2020 | 28/06/2020 |
| Phase 2 | Create test plan | 30/06/2020 | 01/07/2020 |
| Create and execute unit test | 01/07/2020 | 28/07/2020 |
| Create and execute integration test | 03/07/2020 | 30/07/2020 |
| Phase 3 | Create test plan | 20/07/2020 | 21/07/2020 |
| Create and execute system test | 25/7/2020 | 22/08/2020 |
| Create and execute acceptance test | 28/07/2020 | 15/08/2020 |

**5.2.3.5. Deliverables**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Deliverables | Responsibilities | Delivered date |
| 1 | Test plan | Test Leader | 21/07/2020 |
| 2 | Unit test | Developer | 28/07/2020 |
| 3 | Integration test | Tester + Developer | 30/07/2020 |
| 4 | System test | Tester + Developer + PM | 12/08/2020 |
| 5 | Acceptance test | All member | 15/08/2020 |
| 6 | Defect logs | All member | 18/08/2020 |
| 7 | Final test summary report | Test Leader | 20/08/2020 |

5.2.4. Features to be tested

* All features that are listed in Requirement that application have.
* GUI of the Mobile application.

5.3. Test case

5.3.1. UnitTest

Unit tests are done by the developers in order to test individual units, and approved by the test leader.

This is done to ensure that important functions, such as algorithms for verb conjugation and name conversion, return the correct values.

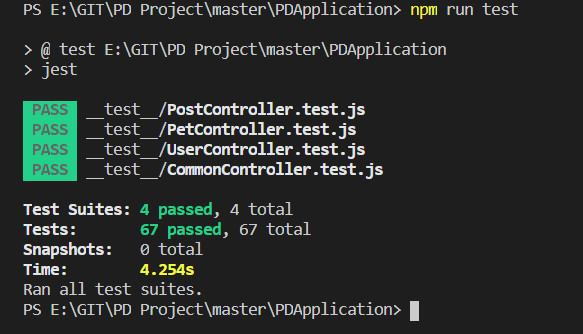


Figure 95 - Unit test with jest

5.3.2. System testing

Detail test cases will be described in **“Petdating\_Test Case\_ver01.xls”** file.

5.3.3. Acceptance test

Acceptance Testing is a level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. But our project will use the Check Lists as a substitute for Acceptance testing.

The content of the Checklist is shown in the table below

|  |  |  |  |
| --- | --- | --- | --- |
| No | Checklist | Yes | No |
| General | | | |
| 1 | Text does not have grammatical and spelling errors. | √ |  |
| 2 | All buttons are functional. | √ |  |
| 3 | All mandatory fields are validated and indicated by asterisk (\*) symbol. | √ |  |
| 4 | All error messages are displayed using red color. | √ |  |
| 5 | All "Delete" functions ask for confirmation. | √ |  |
| GUI and usability | | | |
| 6 | Screen design follows the standard of the project. | √ |  |
| 7 | Waiting icon appear for waiting display data. | √ |  |
| 8 | Color red is used for all error messages. | √ |  |
| 9 | Project state is displayed and colored with appropriate state | √ |  |
| 10 | Design style is friendly and easy to use | √ |  |
| 11 | The text easy to understand. Don't use slang, acronyms, and abbreviations. | √ |  |
| 12 | The static text is clear, concise, and meaningful. | √ |  |
| 13 | The screen designed to fit with multiple screen size. | √ |  |

5.3.4. Defect Log

Excel and github are used to manage bugs:

* When a bug is found, testers change the status and assign for developer.
* Developers fix bugs and change status of bugs to resolve.
* If the bug is fixed then tester change the status to close, otherwise will be re-open again to fix.

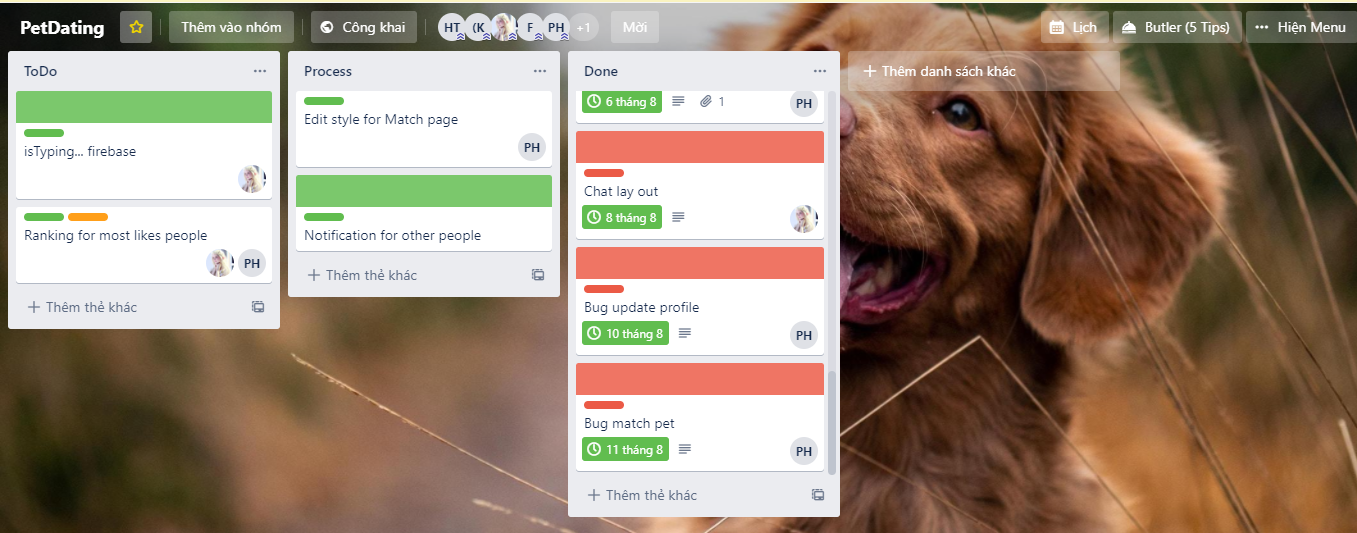


Figure 96 - Defect logs with Trello

5.4. Test Report



### Unit test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Function Name | Number of test case (Estimate) | Pass | Fail |
| 1 | dateFormat | 10 | 10 | 0 |
| 2 | reversDF | 4 | 4 | 0 |
| 3 | getAge | 6 | 6 | 0 |
| 4 | timeFormat | 5 | 5 | 0 |
| 5 | petValidation | 10 | 10 | 0 |
| 6 | captionValidation | 6 | 6 | 0 |
| 7 | userValidation | 12 | 12 | 0 |
| 8 | phoneValidation | 14 | 14 | 0 |
| Total | | 67 | 67 | 0 |

### Integration and system tests

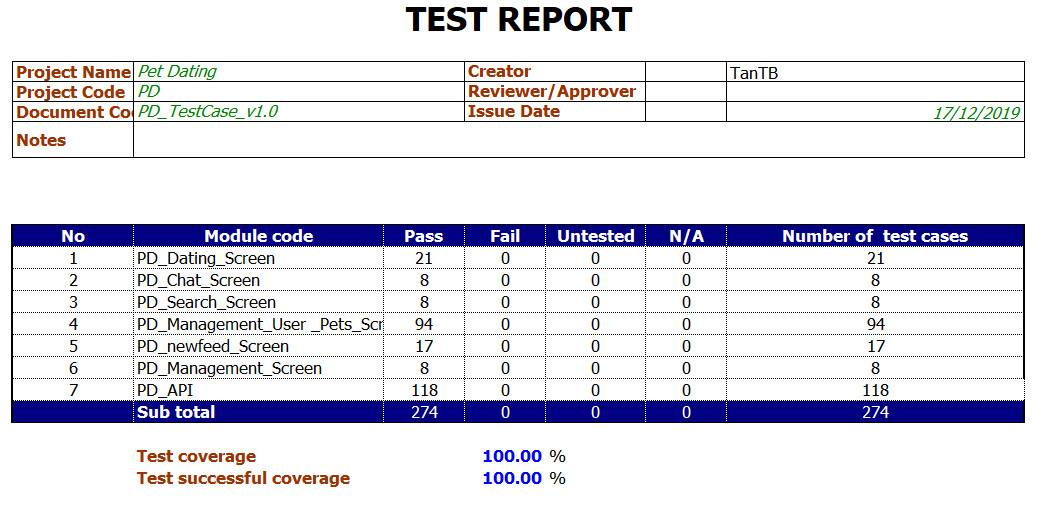


Figure 97 - Integration and system test report