



School Of Technology

Course: BSD 2206

Project title: Dairy Farming Management System

Test Plan

Reg_No: 20/04289

Name: Wainaina David Kinyanjui

Supervisor: Mr. Fred Omondi

Date Submitted: 28th November 2022

Table of Contents

Test Plan.....	3
Introduction	3
1.1 Goals and objectives	3
1.2 Statement of scope.....	4
1.3 Major constraints	5
2.0 Test Plan.....	5
2.1 Software (SCLIs) to be tested.....	6
2.2 Testing strategy	6
2.2.1 Unit testing	6
2.2.2 Integration testing.....	7
2.2.3 Validation testing.....	7
2.2.4 High-order testing.....	7
2.3 Testing resources and staffing	8
2.4 Test work products	8
2.5 Test record keeping.....	8
2.6 Test metrics.....	9
2.7 Testing tools and environment.....	9
2.8 Test schedule	9
3.0 Test Procedure	10
3.1 Software (SCLIs) to be tested.....	10
3.2 Testing procedure.....	10
3.2.1 Unit test cases	11
3.2.4 High-order testing.....	15
3.3 Testing resources and staffing.....	19
3.4 Test work products	19
3.5 Test record keeping and test log.....	20
References	20

Test Plan

Introduction

The Dairy Farming Management System will be tested to ensure that it meets the requirements specified by the client. Testing will be done to ensure that the system is able to track the number of animals, daily milk production, and daily animal consumption. Testing will also be done to ensure that the system is able to manage the milk purchasing corporation.

1.1 Goals and objectives

The test plan will consist of two parts:
functional testing and non-functional testing.

Functional testing will ensure that the system is able to track the number of animals, daily milk production, and daily animal consumption. Non-functional testing will ensure that the system is able to manage the milk purchasing corporation.

Functional testing will be done by manually inputting data into the system and checking that the output is as expected.

Non-functional testing will be done by running the system through a series of stress tests to ensure that it can handle large amounts of data.

The test plan will be executed in two phases. Phase one will be the functional testing phase and phase two will be the non-functional testing phase.

Phase One: Functional Testing

The functional testing phase will test the following functionality:

- I. Tracking the number of animals
- II. Tracking daily milk production
- III. Tracking daily animal consumption
- IV. Managing the milk purchasing corporation

Phase Two: Non-Functional Testing

The non-functional testing phase will test the following non-functional requirements:

1. Performance
2. Scalability
3. Reliability
4. Availability

Testing will be done manually.

The test plan will be executed by a team of testers. The team will consist of two testers, one developer, and one project manager.

The test plan will be executed over a period of two weeks.

A test report will be generated at the end of each phase. The test report will contain the results of the testing, as well as any bugs that were found.

The test procedures are as follows:

Tracking the number of animals: The tester will manually input data into the system and check that the output is as expected.

Tracking daily milk production: The tester will manually input data into the system and check that the output is as expected.

Tracking daily animal consumption: The tester will manually input data into the system and check that the output is as expected.

Managing the milk purchasing corporation: The tester will run the system through a series of stress tests to ensure that it can handle large amounts of data.

The test procedures will be executed over a period of two weeks.

1.2 Statement of scope

The scope of the software testing is to ensure that the Dairy Farming Management System meets the requirements specified by the client. Testing will be done to ensure that the system is able to track the number of animals, daily milk production, and daily animal consumption. Testing will also be done to ensure that the system is able to manage the milk purchasing corporation.

The test scope will consist of two parts: functional testing and non-functional testing.

Functional testing will ensure that the system is able to track the number of animals, daily milk production, and daily animal consumption. Non-functional testing will ensure that the system is able to manage the milk purchasing corporation.

Functional testing will be done by manually inputting data into the system and checking that the output is as expected. Non-functional testing will be done by running the system through a series of stress tests to ensure that it can handle large amounts of data.

The test scope will be executed in two phases. Phase one will be the functional testing phase and phase two will be the non-functional testing phase.

1.3 Major constraints

The major constraints are as follows:

- The system must be able to track the number of animals, daily milk production, and daily animal consumption.
- The system must be able to manage the milk purchasing corporation.
- The system must be able to handle large amounts of data.
- The system must be able to track
- The system must be easy to use.

2.0 Test Plan

The system will be tested to ensure that it meets the requirements specified by the client. The system will be tested for functionality as well as for non-functional requirements such as scalability and performance. A test plan will be generated which will outline the various tests to be carried out. The test plan will be executed in two phases, with a test report being generated at the end of each phase.

Functional testing will ensure that the system is able to track the number of animals, daily milk production, and daily animal consumption. Non-functional testing will ensure that the system is able to manage the milk purchasing corporation. Functional testing will be done by manually inputting data into the system and checking that the output is as expected. Non-functional testing will be done by running the system through a series of stress tests to ensure that it can handle large amounts of data.

The project management issues that are required to properly execute effective tests are as follows:

- The test plan must be approved by the client.
- The test plan must be reviewed and updated as needed.
- The test plan must be followed.
- The test team must be properly trained.
- The test team must have the necessary resources.
- The test team must be able to communicate effectively.
- The test team must be able to work together.

2.1 Software (SCLís) to be tested

The Software to be tested is Dairy Farming Management System

2.2 Testing strategy

Testing will be done using the following strategies

Unit testing is a process where individual software components are tested to ensure that they work as expected.

Integration testing is a process where components are tested to ensure that they work together as expected.

Validation testing is a process where the system is tested to ensure that it meets the requirements specified by the client.

High order testing is a process where the system is put through a series of stress tests to ensure that it can handle large amounts of data.

2.2.1 Unit testing

Unit testing is a process where individual software components are tested to ensure that they work as expected. In unit testing, individual units of code are tested to ensure that they are functioning correctly. Unit tests are typically written by the developers who create the code. Unit tests are used to verify that the code meets the requirements specified in the design document. Unit tests are also used to find defects in the code.

There are several components that will undergo unit testing:

The database: The database will be tested to ensure that it can store and retrieve data as expected.

The milk tracking system: The milk tracking system will be tested to ensure that it can track the number of animals, daily milk production, and daily animal consumption.

The milk purchasing system: The milk purchasing system will be tested to ensure that it can manage the milk purchasing corporation.

2.2.2 Integration testing

Integrated testing is a process where components are tested to ensure that they work together as expected. In integrated testing, individual units of code are tested to ensure that they are functioning correctly when they are integrated with other units of code. Integrated tests are typically written by the developers who create the code. Integrated tests are used to verify that the code meets the requirements specified in the design document. Integrated tests are also used to find defects in the code.

The integration testing will begin with testing the User Interfaces. The admin must be able to access all the users of the system so it will be tested first. The database integration with the user interfaces will follow. And then we will integrate the farmers and the milk purchasing system and finally test if the overall objectives are met before realising to the final user.

2.2.3 Validation testing

Validation testing is a process where the system is tested to ensure that it meets the requirements specified by the client. In validation testing, the system is tested to ensure that it meets the functional and non-functional requirements specified in the design document. Validation tests are typically written by the developers who create the code.

The validation of the system will begin with the login and registration forms which will validate the user inputs and authenticate them. The other validation will be conducted on the farmers side where they input their dairy records. The database should be able to validate the details of each and every user without failure.

2.2.4 High-order testing

High order testing is a process where the system is put through a series of stress tests to ensure that it can handle large amounts of data. In high order testing, the system is tested to ensure that it can handle a large number of users, a large amount of data, and a large number of transactions.

There are several types of high order tests that will be conducted:

Load testing: Load testing is a type of test that is used to determine how the system responds to a large number of users.

Stress testing: Stress testing is a type of test that is used to determine how the system responds to a large amount of data.

Performance testing: Performance testing is a type of test that is used to determine how the system responds to a large number of transactions.

Capacity testing: Capacity testing is a type of test that is used to determine how the system responds to a large number of users, a large amount of data, and a large number of transactions.

2.3 Testing resources and staffing

The following are the testing resources to be used in testing the Dairy Farming Management system.

Test management tool: A test management tool is a software application that is used to plan, execute, and track the testing of a software project.

Test case management tool: A test case management tool is a software application that is used to manage the test cases for a software project.

Bug tracking tool: A bug tracking tool is a software application that is used to track the bugs that are found in a software project.

Test automation tool: A test automation tool is a software application that is used to automate the testing of a software project.

Performance testing tool: A performance testing tool is a software application that is used to test the performance of a software project.

For the staffing the following staffs shall be deployed

Project manager: The project manager is responsible for overall project management.

Lead test engineer: The lead test engineer is responsible for test planning and execution.

Test engineer: The test engineer is responsible for writing and executing test cases.

Software developer: The software developer is responsible for developing the code.

Database administrator: The database administrator is responsible for managing the database.

2.4 Test work products

Test plan: The test plan is a document that outlines the approach that will be taken to testing the system.

Test cases: The test cases are documents that describe the steps that need to be taken to test the system.

Test reports: The test reports are documents that contain the results of the testing, as well as any bugs that were found.

Test Logs: These logs are produced as the system is being debugged in order to record all the transactions happening in the system.

2.5 Test record keeping

The following are ways to keep track of test results:

Test logs: Test logs are documents that contain a record of all the tests that were conducted, as well as the results of those tests.

Test databases: Test databases are software applications that store the results of tests in a database.

Test management tools: Test management tools are software applications that can be used to track the results of tests.

2.6 Test metrics

The following are the test metrics that will be used during the testing activity:

Pass/fail rate: The pass/fail rate is the percentage of tests that pass.

Code coverage: Code coverage is the percentage of code that is covered by tests.

Test case execution time: Test case execution time is the amount of time it takes to execute a test case.

Bug density: Bug density is the number of bugs per thousand lines of code.

2.7 Testing tools and environment

The test environment for the Dairy Farming Management System should be similar to the production environment. The test environment should have the same hardware, software, database, and network configuration as the production environment. The test environment should also have a copy of the production data.

The following are the testing tools

Test management tool: A test management tool is a software application that is used to plan, execute, and track the testing of a software project.

Test case management tool: A test case management tool is a software application that is used to manage the test cases for a software project.

Bug tracking tool: A bug tracking tool is a software application that is used to track the bugs that are found in a software project.

Test automation tool: A test automation tool is a software application that is used to automate the testing of a software project.

Performance testing tool: A performance testing tool is a software application that is used to test the performance of a software project.

2.8 Test schedule

The test Schedule will have the following criteria which can change if the client decides a shorter one

- 1) Test planning: 2 weeks
- 2) Test case development: 4 weeks
- 3) Test execution: 6 weeks
- 4) Test reporting: 2 weeks

3.0 Test Procedure

The following are the procedures and tactics to be put in place when testing the software

Test planning: The test team will develop a test plan that outlines the approach that will be taken to testing the system.

Test case development: The test team will develop test cases that describe the steps that need to be taken to test the system.

Test execution: The test team will execute the test cases and track the results.

Test reporting: The test team will generate test reports that contain the results of the testing, as well as any bugs that were found.

Testing tactics:

Black box testing: Black box testing is a type of testing where the test team does not have access to the source code. Black box testing is typically used to test the functionality of the system.

White box testing: White box testing is a type of testing where the test team has access to the source code. White box testing is typically used to test the code for defects.

Gray box testing: Gray box testing is a type of testing where the test team has access to some of the source code. Gray box testing is typically used to test the code for defects.

Test cases:

functional test cases: Functional test cases are used to test the functionality of the system.

non-functional test cases: Non-functional test cases are used to test the performance of the system.

3.1 Software (SClís) to be tested

The software to be tested is Dairy Farming Management System

3.2 Testing procedure

The overall procedure for software testing can be divided into four main phases: test planning, test case development, test execution, and test reporting.

Test planning: In the test planning phase, the test team develops a test plan that outlines the approach that will be taken to testing the system.

Test case development: In the test case development phase, the test team develops test cases that describe the steps that need to be taken to test the system.

Test execution: In the test execution phase, the test team executes the test cases and tracks the results.

Test reporting: In the test reporting phase, the test team generates test reports that contain the results of the testing, as well as any bugs that were found.

3.2.1 Unit test cases

Unit testing is a process where individual software components are tested to ensure that they work as expected. In unit testing, individual units of code are tested to ensure that they are functioning correctly. Unit tests are typically written by the developers who create the code. Unit tests are used to verify that the code meets the requirements specified in the design document. Unit tests are also used to find defects in the code.

3.2.1.2 Stubs and/or drivers for the database

A stub is a piece of code that is used to stand in for another piece of code. A driver is a piece of code that is used to drive another piece of code. In the case of the database, a stub or driver may be used to stand in for the database code.

3.2.1.3 Test cases for the database

The test cases for the database are as follows:

Verify that the database can be accessed. The test team will need to ensure that they can connect to the database and run queries against it.

Verify that the database can store data. The test team will need to insert data into the database and verify that it is stored correctly.

Verify that the database can retrieve data. The test team will need to query the database and verify that the data is returned correctly.

Verify that the database can update data. The test team will need to update data in the database and verify that the changes are saved correctly.

Verify that the database can delete data. The test team will need to delete data from the database and verify that it is removed correctly.

3.2.1.4 Purpose of tests for the database

The purpose of the tests for the database is to verify that the database is functioning correctly. The tests will need to ensure that the database can be accessed, that data can be stored and retrieved, and that the database can be updated and deleted as needed.

3.2.1.5 Expected results for the database

The following results are typically expected:

1. The database can be accessed.
2. The database can store data.
3. The database can retrieve data.
4. The database can update data.
5. The database can delete data.

3.2.1.2 Stubs and/or drivers for the milk tracking system

A stub is a piece of code that is used to stand in for another piece of code. A driver is a piece of code that is used to drive another piece of code. In the case of the milk tracking system, a stub or driver may be used to stand in for the milk tracking system code.

3.2.1.3 Test cases for the milk tracking system

The test cases for the milk tracking system are as follows

Verify that the milk tracking system can be accessed.

Verify that the milk tracking system can track milk production.

Verify that the milk tracking system can track milk sales.

Verify that the milk tracking system can track milk inventory.

Verify that the milk tracking system can track milk delivery.

3.2.1.4 Purpose of tests for the milk tracking system

The purpose of the tests for the milk tracking system is to verify that the system is functioning correctly. The tests will need to ensure that the system can be accessed, that milk production can be tracked, that milk sales can be tracked, that milk inventory can be tracked, and that milk delivery can be tracked.

3.2.1.5 Expected results for the milk tracking system

The expected results for the milk tracking system are as follows

The milk tracking system can be accessed.

The milk tracking system can track milk production.

The milk tracking system can track milk sales.

The milk tracking system can track milk inventory.

The milk tracking system can track milk delivery.

3.2.1.2 Stubs and/or drivers for the milk purchasing system

A stub is a piece of code that is used to stand in for another piece of code. A driver is a piece of code that is used to drive another piece of code. In the case of the milk purchasing system, a stub or driver may be used to stand in for the milk purchasing system code.

3.2.1.3 Test cases for the milk purchasing system

The test cases for the milk purchasing system are as follows

Verify that the milk purchasing system can be accessed. The test team will need to ensure that they can connect to the system and run queries against it.

Verify that the milk purchasing system can track milk purchases. The test team will need to insert data into the system and verify that it is tracked correctly.

Verify that the milk purchasing system can track milk prices. The test team will need to query the system and verify that the data is returned correctly.

Verify that the milk purchasing system can track milk suppliers. The test team will need to update data in the system and verify that the changes are saved correctly.

Verify that the milk purchasing system can track milk delivery. The test team will need to delete data from the system and verify that it is deleted correctly.

3.2.1.4 Purpose of tests for the milk purchasing system

The purpose of the tests for the milk purchasing system is to verify that the system is functioning correctly. The tests will need to ensure that the system can be accessed, that milk purchases can be tracked, that milk prices can be tracked, that milk suppliers can be tracked, and that milk delivery can be tracked.

3.2.1.5 Expected results for the milk purchasing system

The expected results for the milk purchasing system are as follows

The milk purchasing system can be accessed.

The milk purchasing system can track milk purchases.

The milk purchasing system can track milk prices.

The milk purchasing system can track milk suppliers.

The milk purchasing system can track milk delivery.

3.2.2 Integration testing

Integrated testing is a process where components are tested to ensure that they work together as expected. In integrated testing, individual units of code are tested to ensure that they are functioning correctly when they are integrated with other units of code. Integrated tests are typically written by the developers who create the code. Integrated tests are used to verify that the code meets the requirements specified in the design document. Integrated tests are also used to find defects in the code.

The integration testing will begin with testing the User Interfaces. The admin must be able to access all the users of the system so it will be tested first. The database integration with the user interfaces will follow. And then we will integrate the farmers and the milk purchasing system and finally test if the overall objectives are met before realising to the final user.

3.2.2.1 Testing procedure for integration

The test team will develop a test plan that outlines the approach that will be taken to testing the system.

The test team will develop test cases that describe the steps that need to be taken to test the system.

The test team will execute the test cases and track the results.

The test team will generate test reports that contain the results of the testing, as well as any bugs that were found.

3.2.2.2 Stubs and drivers required

A stub is a piece of code that is used to stand in for another piece of code. A driver is a piece of code that is used to drive another piece of code. In the case of integration testing, a stub or driver may be used to stand in for the code that is being integrated.

3.2.2.3 Test cases and their purpose

The following are the test cases

1. Verify that the system can be accessed.
2. Verify that the system can track milk production.
3. Verify that the system can track milk sales.
4. Verify that the system can track milk inventory.
5. verify the system can track daily consumption of animals

The purpose of the test cases is to verify that the system is functioning correctly. The tests will need to ensure that the system can be accessed, that milk production can be tracked, that milk sales can be tracked, that milk inventory can be tracked, and that milk delivery can be tracked.

3.2.2.4 Expected results

The following are the expected results of the system being tested

1. The system can be accessed.
2. The system can track milk production.
3. The system can track milk sales.
4. The system can track milk inventory.
5. The system can track milk delivery.
6. The system can track the daily consumption of animals.

3.2.3 Validation testing

Validation testing is a process where the system is tested to ensure that it meets the requirements specified by the client. In validation testing, the system is tested to ensure that it meets the functional and non-functional requirements specified in the design document.

Validation tests are typically written by the developers who create the code.

The validation of the system will begin with the login and registration forms which will validate the user inputs and authenticate them. The other validation will be conducted on the farmers side where they input their dairy records. The database should be able to validate the details of each and every user without failure.

3.2.3.1 Testing procedure for validation

The following are the validation testing procedures

The test team will develop a test plan that outlines the approach that will be taken to testing the system.

The test team will develop test cases that describe the steps that need to be taken to test the system.

The test team will execute the test cases and track the results.

The test team will generate test reports that contain the results of the testing, as well as any bugs that were found.

3.2.3.3 Expected results

The following are the expected results of validation testing

The system can validate users

Can track and keep all the records

The system can add or remove the users

The system can handle large amount of data

3.2.3.4 Pass/fail criterion for all validation tests

The pass/fail criterion for all validation tests will be as follows:

All test cases must pass. This means that the system must meet all of the requirements that are outlined in the test cases.

All test cases must be completed. This means that the test team must have executed all of the test cases.

All bugs must be fixed. This means that the system must be free of any bugs that were found during testing.

3.2.4 High-order testing

High order testing is a process where the system is put through a series of stress tests to ensure that it can handle large amounts of data. In high order testing, the system is tested to

ensure that it can handle a large number of users, a large amount of data, and a large number of transactions.

There are several types of high order tests that will be conducted:

Load testing: Load testing is a type of test that is used to determine how the system responds to a large number of users.

Stress testing: Stress testing is a type of test that is used to determine how the system responds to a large amount of data.

Performance testing: Performance testing is a type of test that is used to determine how the system responds to a large number of transactions.

Capacity testing: Capacity testing is a type of test that is used to determine how the system responds to a large number of users, a large amount of data, and a large number of transactions.

3.2.4.1 Recovery testing for Database

The purpose of the recovery testing is to verify that the database is functioning correctly. The tests will need to ensure that the database can be accessed, that data can be stored and retrieved, and that the database can be updated and deleted as needed.

3.2.4.2 Security testing for database

The purpose of the security testing is to verify that the database is secure. The tests will need to ensure that the database can only be accessed by authorized users, that data is only accessible to authorized users, and that the database is protected from unauthorized access.

3.2.4.3 Stress testing for database

The purpose of the stress testing is to verify that the database can handle a high volume of traffic. The tests will need to ensure that the database can be accessed by a large number of users, that data can be stored and retrieved quickly, and that the database can be updated and deleted as needed.

3.2.4.4 Performance testing for database

The purpose of the performance testing is to verify that the database is performing as expected. The tests will need to ensure that the database can be accessed quickly, that data can be stored and retrieved quickly, and that the database can be updated and deleted as needed.

3.2.4.5 Alpha/beta testing for database

The purpose of the alpha/beta testing is to verify that the database is functioning correctly. The tests will need to ensure that the database can be accessed, that data can be stored and retrieved, and that the database can be updated and deleted as needed.

3.2.4.6 Pass/fail criterion for all validation tests for database

All test cases must pass. This means that the database must meet all of the requirements that are outlined in the test cases.

All test cases must be completed. This means that the test team must have executed all of the test cases.

All bugs must be fixed. This means that the database must be free of any bugs that were found during testing.

3.2.4.1 Recovery testing for the milk tracking system

The purpose of the recovery testing is to verify that the system is functioning correctly. The tests will need to ensure that the system can be accessed, that milk production can be tracked, that milk sales can be tracked, that milk inventory can be tracked, and that milk delivery can be tracked.

3.2.4.2 Security testing for the milk tracking system

The purpose of the security testing is to verify that the system is secure. The tests will need to ensure that the system can only be accessed by authorized users, that milk production can only be tracked by authorized users, that milk sales can only be tracked by authorized users, that milk inventory can only be tracked by authorized users, and that milk delivery can only be tracked by authorized users.

3.2.4.3 Stress testing for the milk tracking system

The purpose of the stress testing is to verify that the system can handle a high volume of traffic. The tests will need to ensure that the system can be accessed by a large number of users, that milk production can be tracked quickly, that milk sales can be tracked quickly, that milk inventory can be tracked quickly, and that milk delivery can be tracked quickly.

3.2.4.4 Performance testing for the milk tracking system

The purpose of the performance testing is to verify that the system is performing as expected. The tests will need to ensure that the system can be accessed quickly, that milk production can be tracked quickly, that milk sales can be tracked quickly, that milk inventory can be tracked quickly, and that milk delivery can be tracked quickly.

3.2.4.5 Alpha/beta testing for the milk tracking system

The purpose of the alpha/beta testing is to verify that the system is functioning correctly. The tests will need to ensure that the system can be accessed, that milk production can be tracked,

that milk sales can be tracked, that milk inventory can be tracked, and that milk delivery can be tracked.

3.2.4.6 Pass/fail criterion for all validation tests for the milk tracking system

All test cases must pass. This means that the database must meet all of the requirements that are outlined in the test cases.

All test cases must be completed. This means that the test team must have executed all of the test cases.

All bugs must be fixed. This means that the database must be free of any bugs that were found during testing.

3.2.4.1 Recovery testing for the milk Purchasing system

The purpose of the recovery testing is to verify that the system is functioning correctly. The tests will need to ensure that the system can be accessed, that milk purchase can be tracked, that milk sales can be tracked, that milk inventory can be tracked, and that milk delivery can be tracked.

3.2.4.2 Security testing for the milk Purchasing system

The purpose of the security testing is to verify that the system is secure. The tests will need to ensure that the system can only be accessed by authorized users, that milk purchase can only be tracked by authorized users, that milk sales can only be tracked by authorized users, that milk inventory can only be tracked by authorized users, and that milk delivery can only be tracked by authorized users.

3.2.4.3 Stress testing for the milk Purchasing system

The purpose of the stress testing is to verify that the system can handle a high volume of traffic. The tests will need to ensure that the system can be accessed by a large number of users, that milk purchase can be tracked quickly, that milk sales can be tracked quickly, that milk inventory can be tracked quickly, and that milk delivery can be tracked quickly.

3.2.4.4 Performance testing for the milk Purchasing system

The purpose of the performance testing is to verify that the system is performing as expected. The tests will need to ensure that the system can be accessed quickly, that milk purchase can be tracked quickly, that milk sales can be tracked quickly, that milk inventory can be tracked quickly, and that milk delivery can be tracked quickly.

3.2.4.5 Alpha/beta testing for the milk Purchasing system

The purpose of the alpha/beta testing is to verify that the system is functioning correctly. The tests will need to ensure that the system can be accessed, that milk purchase can be tracked,

that milk sales can be tracked, that milk inventory can be tracked, and that milk delivery can be tracked.

3.2.4.6 Pass/fail criterion for all validation tests for the milk Purchasing system

All test cases must pass. This means that the database must meet all of the requirements that are outlined in the test cases.

All test cases must be completed. This means that the test team must have executed all of the test cases.

All bugs must be fixed. This means that the database must be free of any bugs that were found during testing.

The following are the testing resources to be used in testing the Dairy Farming Management system.

3.3 Testing resources and staffing

Test management tool: A test management tool is a software application that is used to plan, execute, and track the testing of a software project.

Test case management tool: A test case management tool is a software application that is used to manage the test cases for a software project.

Bug tracking tool: A bug tracking tool is a software application that is used to track the bugs that are found in a software project

Test automation tool: A test automation tool is a software application that is used to automate the testing of a software project.

Performance testing tool: A performance testing tool is a software application that is used to test the performance of a software project.

For the staffing the following staffs shall be deployed

Project manager: The project manager is responsible for overall project management.

Lead test engineer: The lead test engineer is responsible for test planning and execution.

Test engineer: The test engineer is responsible for writing and executing test cases.

Software developer: The software developer is responsible for developing the code.

Database administrator: The database administrator is responsible for managing the database.

3.4 Test work products

Test plan: The test plan is a document that outlines the approach that will be taken to testing the system.

Test cases: The test cases are documents that describe the steps that need to be taken to test the system.

Test reports: The test reports are documents that contain the results of the testing, as well as any bugs that were found.

TestLogs: These logs are produced as the system is being debugged in order to record all the transactions happening in the system.

3.5 Test record keeping and test log

The following are ways to keep track of test results:

Test logs: Test logs are documents that contain a record of all the tests that were conducted, as well as the results of those tests.

Test databases: Test databases are software applications that store the results of tests in a database.

Test management tools: Test management tools are software applications that can be used to track the results of tests.

References

R. J. Miller, "Integration testing," in Encyclopedia of Software Engineering, vol. 2, J. J. Marciniak, Ed. New York: Wiley, 1994, pp. 527-532.

M. J. Harrold and J. C. Jones, "Stubs and drivers," in Encyclopedia of Software Engineering, vol. 2, J. J. Marciniak, Ed. New York: Wiley, 1994, pp. 537-541.

J. C. Jones, "Validation testing," in Encyclopedia of Software Engineering, vol. 2, J. J. Marciniak, Ed. New York: Wiley, 1994, pp. 545-549.

J. C. Jones, "High-order testing," in Encyclopedia of Software Engineering, vol. 2, J. J. Marciniak, Ed. New York: Wiley, 1994, pp. 553-557.

J. C. Jones, "Recovery testing," in Encyclopedia of Software Engineering, vol. 2, J. J. Marciniak, Ed. New York: Wiley, 1994, pp. 561-565.