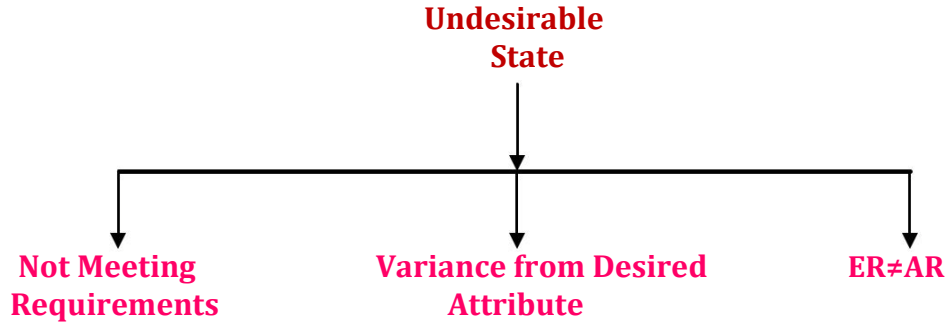
1. TESTING

- It helps to **identify the defects** and **provide quality product** to end user.
- It will happen **D (Development)** to **D (Delivery to End User).**
- The Product/Application/System will be used in **customer point of view** in all positive and negative perception.
- This testing will be done by **TESTERS.**
- A Defect is an **Undesirable State**



2. QUALITY

What is Quality?

- Quality is defined as **meeting the customer's requirements** for the first time and every time.
- Quality is much more than the **absence of defects, reliability, efficiency, easy to use, within budget.**
- **Producer's view** – Quality of product meets requirements.
- **Customer's view** – Quality of product "fit for use"/ meets customer's needs.

Why Quality?

- Quality is the important factor affecting an organization's long term performance.
- Quality improves productivity and competitiveness in any organization.

How you achieve quality?

Right the first time - if anything is done perfectly in the first time itself then we can say that quality is achieved. When the following thing are done then quality can be achieved

Effectiveness - Doing the Right things
Efficiency - Doing the things right

Quality Testing

Verification + Validation

Quality Control

Measures the quality of the product.
Validation is the process of evaluating the final product to check whether the software meets the business needs.

- QC makes sure that we **developed the product right.**
- It is **product oriented**.
- Defect **detection** based.
- This will happen **after the development**
- **Eg:** Testing (Validation)

Software Quality Factors (SQF)

- Correctness
- Reliability
- Functionality
- Efficiency
- Maintainability
- Reusability
- Compatibility
- Effectiveness
- Testability
- Interoperability
- Integrity
- Flexibility
- Usability

Critical Success Factors

4. SOFTWARE TESTING

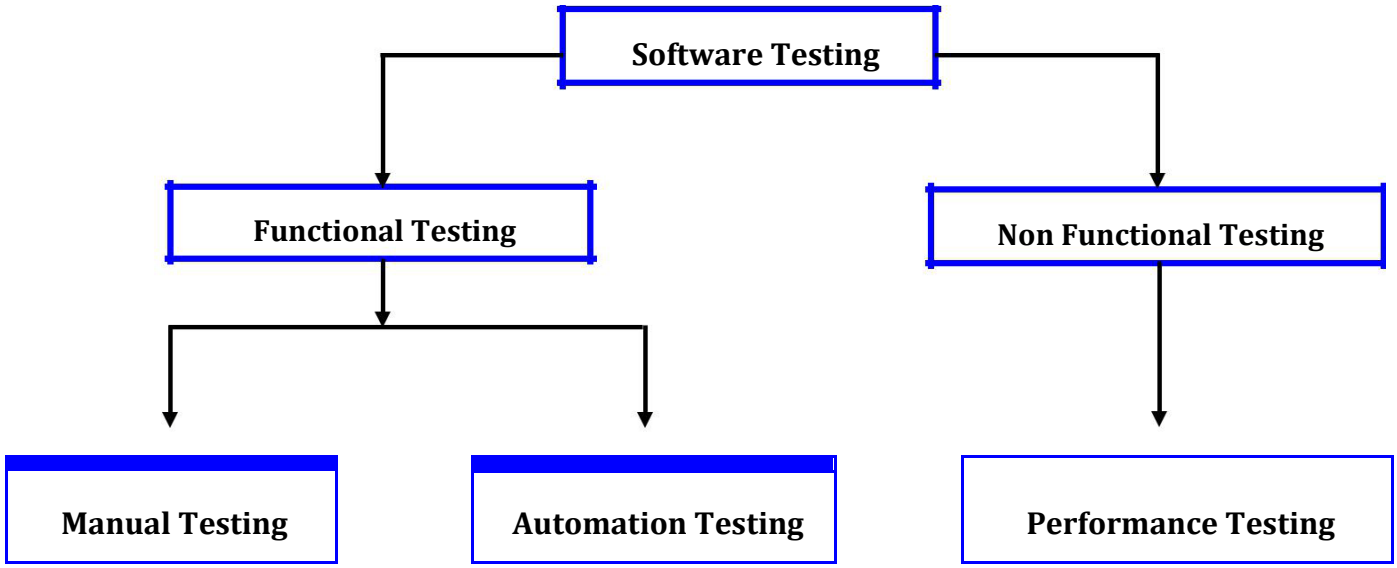
- Software testing is a planned process that is used to identify the correctness, completeness, security and quality of software.
- Software testing is a process of evaluating a system by manual or tool means and verify that it satisfies specified requirements or identify differences between expected and actual results.

Why Software Testing

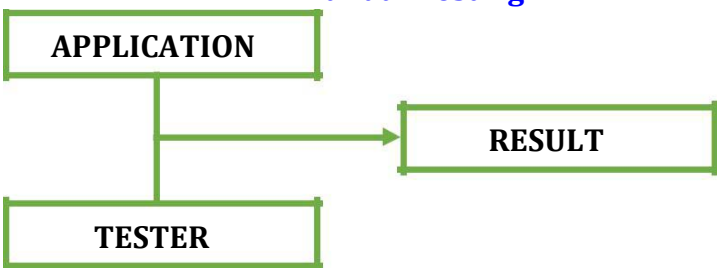
1. To check the reliability of the software.
2. To be ensured that the software does not contain any bug which can become a reason for failure.
3. To check the software was made according to its specification.
4. To check that the software meets its requirements.
5. To check that users are capable of using the software.
6. To check software works with other software and hardware it needs to work with

This testing implemented in two ways

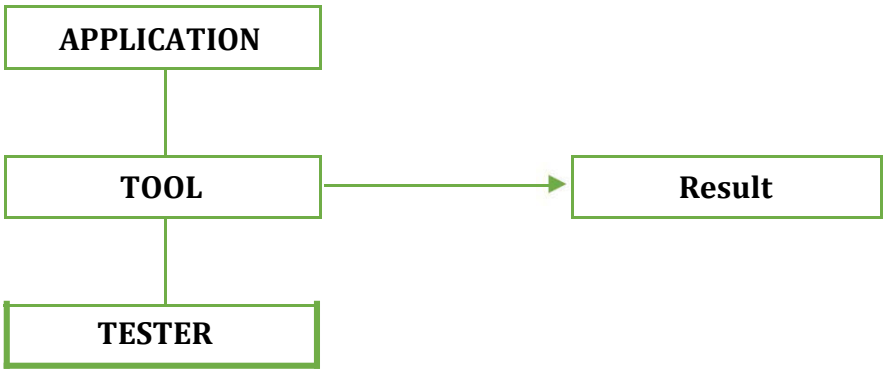
1. Manual Testing
2. Tool Testing



Manual Testing



Tool Testing



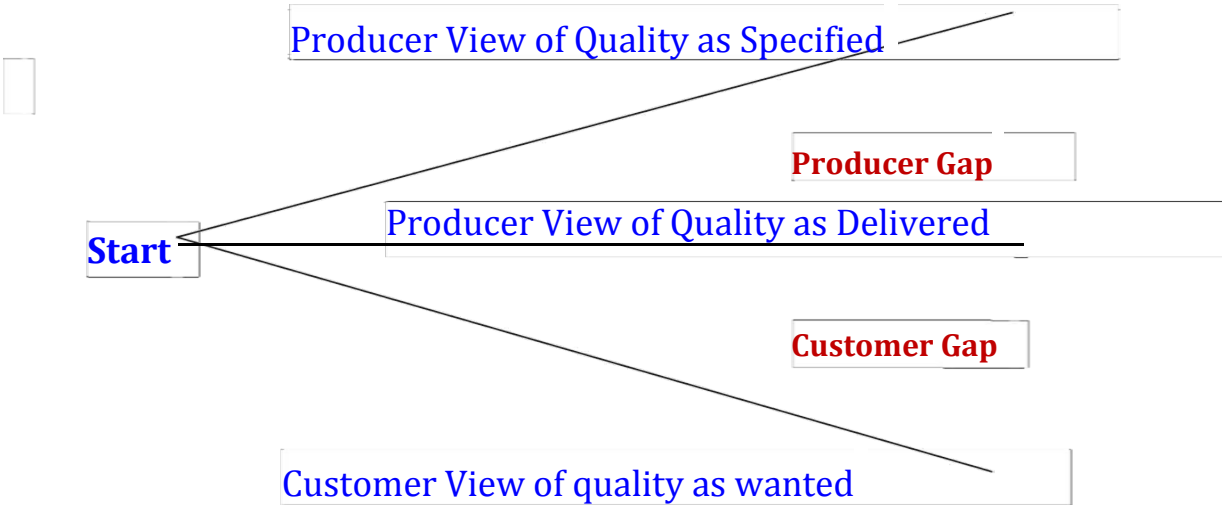
Who Does Testing

1. Software Tester
2. Software Developer
3. Project Lead/Manager
4. End User
5. Information service management
6. Senior organization management

Primary Goal of Testing

- Finding Errors
- Ensure all the functionalities are implemented
- Ensure the software meets the customer requirements
- Ensure the quality of product

Need for Software Testing



Some Key Challenges in Software Testing

1. Testing the Complete Application
2. Relationship with developers
3. Regression testing
4. Testing always under time constraint
5. Understanding the requirements
6. One test team under multiple projects
7. Testers focusing on finding easy bugs
8. Which tests to execute first?
9. Lack of skilled testers
10. Mis-communication or No Communication
11. Requirements are not freezed
12. Application is not Testable
13. Lack of resources
14. Lack of Training

Role of QA

- QA team assures the quality by monitor the whole development process.
- Responsibilities of QA team are planning testing execution Process.
- QA Lead creates the time table and agrees on a quality assurance plan for the document.
- QA team communicated QA process to the team members.
- QA team ensures traceability of test cases to requirements.
-

5. Principles of Testing

1. Testing shows presence of defects

Testing can show the defects are present, but cannot prove that there are no defects. Even after testing the application or product thoroughly we cannot say that the product is 100% defect free. Testing always reduces the number of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness.

2. Exhaustive testing is impossible

Testing everything including all combinations of inputs and preconditions is not possible. So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts. For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (5^{15}) tests. This is very unlikely that the project timescales would allow for this number of tests. So, accessing and managing risk is one of the most important activities and reason for testing in any project.

3. Early testing

In the software development life cycle testing activities should start as early as possible and should be focused on defined objectives.

4. Defect clustering

A small number of modules contains most of the defects discovered during pre-release testing or shows the most operational failures.

5. Pesticide paradox:

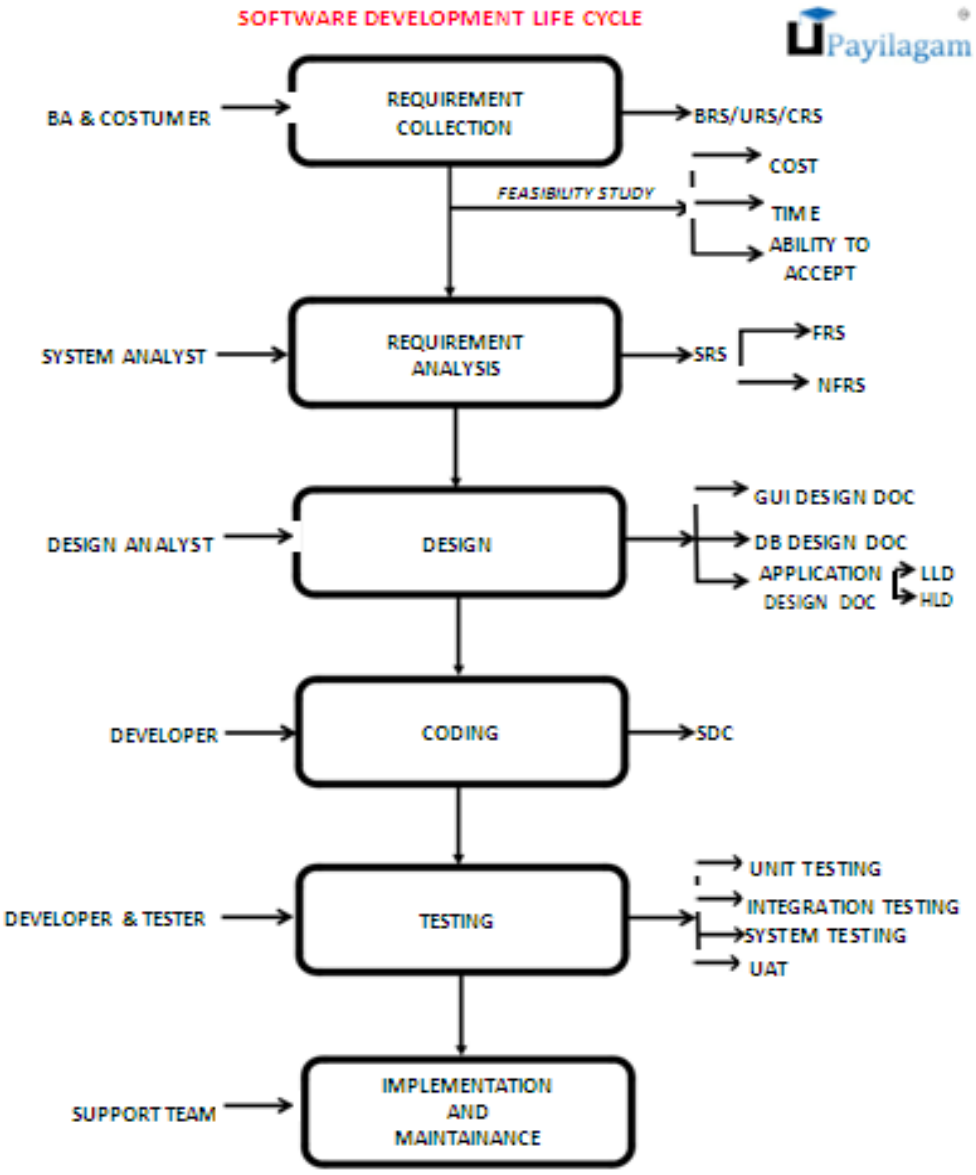
If the same kinds of tests are repeated again and again, eventually the same set of test cases will no longer be able to find any new bugs. To overcome this “Pesticide Paradox”, it is really very important to review the test cases regularly and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

6. Testing is context depending

Testing is basically context dependent. Different kinds of sites are tested differently. For example, safety – critical software is tested differently from an e-commerce site.

7. Absence – of – errors fallacy

If the system built is unusable and does not fulfil the user's needs and expectations then finding and fixing Defects does not help.



SDLC

- It stands for **Software Development Lifecycle.**
- It is the entire process of formal, logical steps taken to develop a software product.
- It describes the activities that take place at each stage of software development process.

SDLC PHASES

Requirement collection:

In this phase with an interaction of Clint BA will collect Clint requirement. Collected information will be documented as BRS (or) customer requirement specifications (CRS).

BRS document describe Clint business needs, like who can access the application and what type of services application has to provide to the user. After preparation of BRS doc Business analyst will perform feasibility study which is called pre project activity. In feasibility testing they verify follow8ing factors to check whether the project (or) product is acceptable or not.

In feasibility study they verify following factors.

1) Finance feasibility.
2) Time feasibility.
3) Ability to accept in terms of technology, environment, Human resources and requirements are reliable or not. If project is acceptable BA intimate to that lint with help of "statement of work" and "service level agreement".

Requirement analysis (or) System analysis:

In this phase system analyst will analyze Clint business needs in BRS document based on that he prepares detailed document SRS.

SRS contains two types of sub document

1. **FRS (Functional requirement spec's):** It describes detailed functionality of the system in order to cover Clint document to understand functionality of application.
2. **NFRS (Non FRS):** It describes about Clint expectations like performance factors, security, compatibility etc.

Design:

In this phase design architect will prepare following documents after analyzing SRS.

1. GUI design doc:

It contains dummy screens of application. A sample application without functionality like prototypes is helpful to know the future implementation of application and as tester point of view it is easy to understand application functionality.

2. Data base design doc:

This document describe about data base structure like tables, rules between those tables.

3. Application design document (or) technical design document:

After architecture of application design architect will prepare two sub docs in TDD.

Two doc's contains in TDD are

1. High level design doc describes no. of modules required for an application and relation between those modules.

Modularization: It is a process of splitting into set of module for easy construction of application.

Module: Means set of similar requirements grouped together.

2. Low level design document: For each module there will in depend low level design document where it describes logic of the each module in order to develop programs.

Coding:

In this phase developer will write the program in order to develop application based on design documents. In general for window base application they use programming languages like java, dotnet, etc. For web based application they use scripting languages PHP, Perl etc.

Testing:

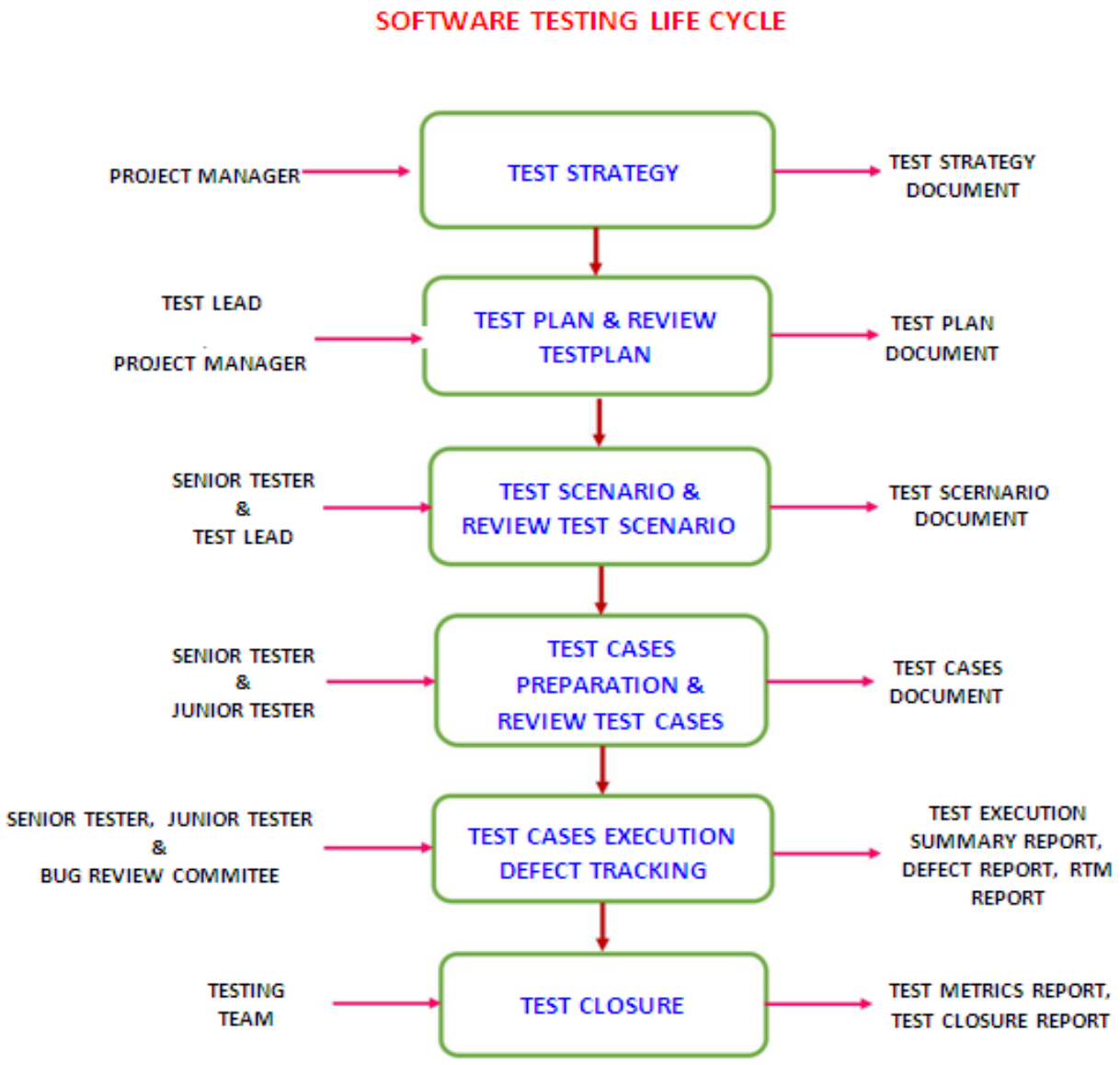
After writing the program development team will perform unit and integration testing using white box testing techniques.

Testing engineers validate application as per Clint requirements and expectations in system testing where they use BBT techniques. After system testing to Clint feedback will perform UAT.

Implementation & Maintenance

Includes implementation of changes that software might undergo over a period of time, or implementation of new requirements after the software is deployed at the customer location. The

maintenance phase also includes handling the residual errors that may exist in the software even after the testing phase. This phase also monitors system performance, rectifies bugs and requested changes are made. Maintenance, often turned support, is a crucial activity for linking the experiences of users/customers with the product delivery organization. We consider perspectives on high tech maintenance from bug fixing through to design focused activities.



TEST INITIATION

STLC starts with Test Initiation. In this stage PM can prepare Word to prepare Test Strategy in IEEE 829 Format. **Test Strategy**. PM can use MS-

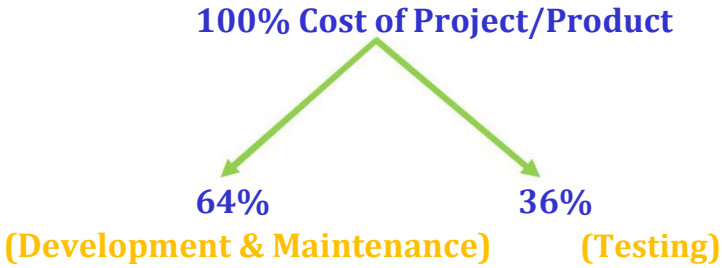
Test Strategy Document:

1. Scope and Objective

PM Can specify importance of testing in current project / product

2. Business Issues

Budget allocation for testing.



3. Test Responsibility Matrix

PM can select reasonable test to be conducted in current Project / Product.

S/W Testing	Yes/No	Comments
Functional Testing	Yes	-
Usability Testing	Yes	
Compatibility Testing	Yes	
H/W Configuration	Yes	
Performance Testing	No	Required Test, Resources not available
Security Testing	No	Required Test, But no Skils
Multi - Language	No	No Requirement

4. Roles and Responsibilities

Jobs for Testing Team & Responsibilities of each job.

Roles	Responsibilities
Test Lead	☐ Prepare Test Plans ☐ Review Test Cases ☐ Involving in Defect Tracking
Senior Tester (Quality Analyst)	☐ Prepare Test Cases ☐ Involving in Defect Reporting
Junior Tester (Quality Controller)	☐ Executing Test Cases ☐ Prepare Defect Reporting

5. Communication & Status Reporting

PM can specify required communication in between testing team members.

Example:

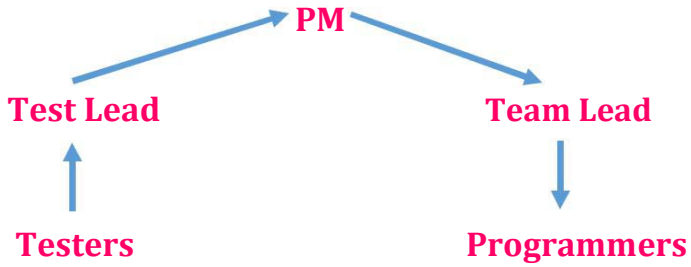
- Jr.Tester can meet Sr.Tester daily.
- Sr.Tester can meet TL weekly twice.
- TL can meet PM weekly once.

5. Test Automation

Current project / product is suitable for testing and also need for automation testing

7. Defect Tracking & Reporting

The process for defect reporting & tracking will be decided by PM



8. Testing Measurements and Metrics

PM can define a set of measurements and metrics to be followed by Testers during testing.

Example:

- 25 to 30 Test cases documents preparation per day
- 15 to 20 Test cases documents execution per day
- 5 Defects detection per day

8. Test Management

Test Management is a part of configuration Management. In general PM is responsible to provide location to Developers and Testers to save their deliverables for future reference.

10. Risk Assumptions

PM can guess challenges will come during testing and identify the solutions to overcome those challenges

Examples:

Risk 1: Lack of time
Risk 2: Lack of Resources
Risk 3: Lack of Documentation
Risk 4: Delays in Delivery
Risk 5: Lack of Skills
Risk 6: Lack of Seriousness to developers
Risk 7: Lack of communication in between developers and testers
Risk 8: Sudden changes in customer requirements.

11. Training Needs

PM can identify need for training to tester for current project / product, Most of the times PM can skip training for budget control.

TEST PLAN

After completion of Team Formation and Risk Identification, Corresponding TL can start Test Plan Preparation in MS-Word by following IEEE 829 Standard Format.

A document that defines the overall testing approach is called Test Plan.

A test plan is a document detailing a systematic approach to testing a system such as a machine or software. The plan typically contains a detailed understanding of the eventual workflow.

IEEE 829 Test Plan Document format classified into 4parts.

- **What to Test?**
- **How to Test?**
- **Who to Test?**
- **When to Test?**

TEST PLAN DOCUMENT

1. **TEST PLAN DOCUMENT ID:**
   Unique number or Name for documents
2. **INTRODUCTION:**
   About current project / Product Testing
3. **FEATURES & MODULES:**
   List of all modules in current project
4. **FEATURES TO BE TESTED:**
   List of modules to be tested
5. **FEATURES NOT TO BE TESTED:**
   List of modules not to be tested
6. **TEST STATEGY:**
   PM will give Strategy

What to Test?

7. **TEST ENVIRONMENT:**
   Required Hardware's and Software's need for testers
8. **TEST DELIVARBLES:**
   List of Testing Documents to be prepared by Tester during testing.
   Eg: Test Scenario, Test Cases, Test Data, Test Scripts, OR, Test Logs, Review Reports.
9. **ENTRY CRITERIA (When to Start Testing)**
   Test Environment Established Test

   Cases Prepared and Reviewed S/W

   Build released from developers

How to Test?

10. **SUSPENSION CRITERIA (When to Interrupt Testing)**
    Major bug found in software build
    More minor bugs in pending Test
    Environment abounded
11. **EXIT CRITERIA**
    Time Exceeded
    All Modules Tested
    All Majors Bugs closed

How to Test?

12. **RESOURCES:**
    List of Selected Tester for Current Project / Product.
13. **RESPONSIBILITIES:**
    Work allocation to testers.

Who to Test?

14. **SCHEDULE:**
    Date and Time.

When to Test?

15. **RISK & ASSUMPTIONS:**
    List of Previously Analyzed Risks and Solutions to overcome them.
16. **APPROVALS:**
    Signatures of the PM & TL.

TEST SCENARIO

- Test Scenario is nothing but Hypothetical Story
- Test Scenarios are just a flow of the application.
- A concept which provides one-line information about what to test.
- It is nothing but a particular functionality of the application need to be tested.
- What To Test
- TL & Senior Tester will prepare Test Scenario in MS-Office by following IEEE 829 Standard Format.

TEST CASES

- Test Cases are nothing but set of procedures which we execute in our system to find the defect
- An input operation and the corresponding expected output" in order to test a small unit of work.
- Test Cases are derived from Test Scenario
- How To Test
- Senior Tester will prepare the test cases.

Example

Scenario

Checking the functionality of Login button is Test scenario

Test Cases

1. Click the button without entering user name and password.
2. Click the button only entering User name.
3. Click the button while entering wrong user name and wrong password.

TEST CASES EXECUTION

- Test execution is culmination of testing activities which involves executing the planned test cases and conducting of the tests.
- Test execution phase broadly involves execution and reporting.
- Test execution consists of following activities to be performed
- Creation of test setup or Test bed
- Execution of test cases on the setup
- Test Methodology used
- Collection of Metrics
- Defect Tracking and Reporting
- Map Defects to Test Cases in RT
- Resting & Regression Testing

DEFECT TRACKING AND REPORTING

Defect Reporting (Bug Life cycle) is the journey of a defect from its identification to its closure. It starts when defect is found and ends when a defect is closed, after ensuring it's not reproduced. Defect life cycle is related to the bug found during testing. The bug has different states in the Life Cycle.

TEST CLOSURE

Testing team will meet, discuss and analyze testing artifacts to identify strategies that have to be implemented in future, taking lessons from the current test cycle. The idea is to remove the process bottlenecks for future test cycles and share best practices for any similar projects in future.

In the review meeting, testing team discuss below factors.

1. Coverage Analysis

a) Module wise coverage.
b) Testing topic wise coverage.

2. Stability in software

Stability of a software indicates, in 20% of testing we can find 80% of bugs and in 80% of testing we can find 20% of bugs.

3. Calculate Bug density

Modules in SUT	% of bugs	
A	20	
B	20	
C	40	----------> Final regression testing
D	20	
	100 % of bugs	

Note

Testing team can re-execute test cases related to high bug density modules on final SUT for "Golden bugs" to detected. This testing is called as "Final regression" testing or Post mortem testing or Pre- acceptance testing or confidence testing. If golden bug was found then testing team will request the developers to fix as early as possible or request customer site people for later patch release.

4. Analysis of differed bugs

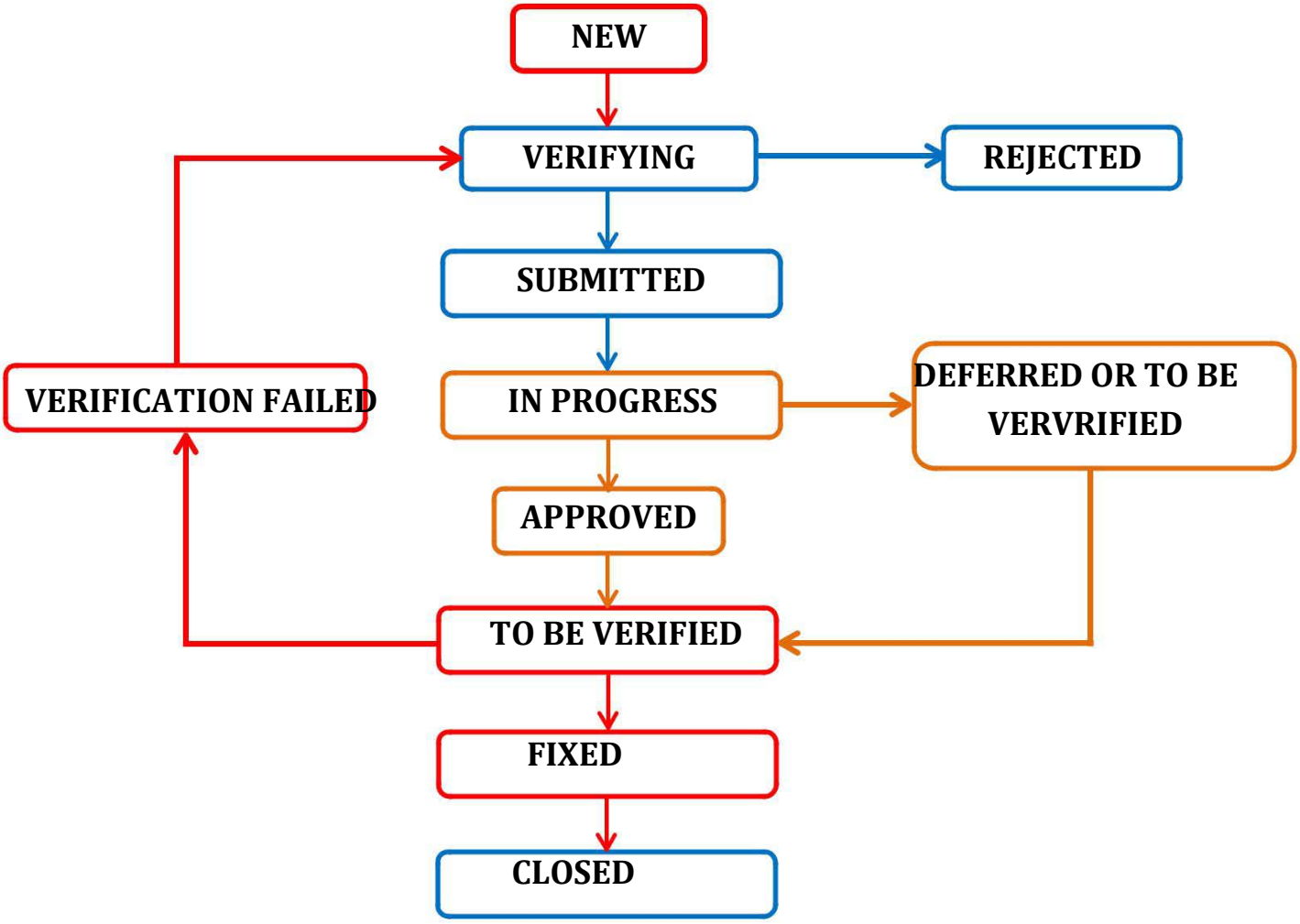
This analysis indicates, whether differed bugs are postpone or not.

The above four points called as "Exit criteria".

5. In software testing final summary report is called as Requirement Traceability Matrix (RTM).

DEFECT LIFE CYCLE FLOW CHART

Defect Life Cycle (Bug Life cycle) is the journey of a defect from its identification to its closure. It starts when defect is found and ends when a defect is closed, after ensuring it's not reproduced. Defect life cycle is related to the bug found during testing. The bug has different states in the Life Cycle.



PROCESS FLOW OF DEFECT LIFE CYCLE

1. Tester will execute the test cases, if tester found any defect then tester will analyse the defect whether it's duplicate or not.
2. If yes, then tester will make updates to original defect.
3. If no, tester will log the defect will keep the status as ***NEW*** and will assign to Bug Review Committee.
4. All defects which are in ***NEW*** status will be reviewed by Bug Review Committee (BA, M, TL and Customer). In this meeting members decide whether the bug is valid or invalid.
5. Then Bug committee validate the defect, review the severity, if it is valid assign to developer, they will keep the status in ***SUBMITTED*** otherwise they reject the defect, will change the status to ***REJECTED***.
6. Then Developer will start working on it, will change the status to ***IN PROGRESS***

7. Once the issue is resolved and validated in development, developer will change the status to ***APRROVED*** and assign to tester.
8. Once fixed defect is ready to test then tester will valid the defect whether it has been fixed or not, Tester will change the status to ***TO BE VERIFIED***.
9. If BUG is not resolved Tester will change the status ***VERIFICATION FAILED*** and will be assigned to Developer repeat the process till BUG has to closed. If the bug is resolved tester will change the status as ***FIXED*** and will assign to BUG Review Committee for closer.
10. Bug Review committee will review and closed the BUG, Only BUG Review Committee (BA, M, TL) has right to keep the status as ***CLOSED*****.**
11. Few defects which are unable to fix in the current release and important for application functionality will be in "***DEFERRED***" status. And will be taken care in next release.

ERROR, DEFECT, BUG, FAILURE, FAULT

ERROR

A human action that produces an incorrect result. Programmatically mistake or syntax mistake leads to error.

DEFECT

If tester found any mismatch between expected and actual value.

BUG

Once developer accepts tester's identified defect that is called bug.

FAILURE

Defect reaches the customer then is called failure.

FAULT

When the product/software successfully launched in the market and running properly but due to any reason if it works unexpectedly is called Fault.

Example

You are driving a car and you are on road while on driving now there is two way on the road
1) left--> mumbai
2) right--> delhi

Now you have to go to delhi it means you have to turn the steering to the right, but by mistake you turn the steering to the left, from that position that is called as "Error" because human interaction is there And now Fault is there till you will not reach the Mumbai, but when you reach Mumbai that is a final stage which is called "Failure" becoz you had to reach Delhi but now you are in Mumbai.

SEVERITY & PRIORITY

SEVERITY

- Impact of the defect
- How severity the bug is affecting the application.

Critical

This defect is causing system failure. Nothing can proceed further. It may also be called as a show stopper

Major

Highly severe defect, is causing the system to collapse, however few parts of the system are still usable, and/or there are a few workarounds for using the system in the collapsed state too

Medium

Is causing some undesirable behavior, however system / feature is still usable to a high degree

Low

Is more of a cosmetic issues, No serious impedance to system functionality is noted

PRIORITY

- Important of the defect
- Informing to developer which defect to be fixed first.

Urgent

Must to be fixed before any other high, medium or low defect should be fixed. Must be fixed in the next build.

High

Must be fixed in any of the upcoming builds but should be included in the release.

Medium

Should take precedence over low priority defects and may be fixed after the release / in the next release.

Low

Fixing can be deferred until all other priority defects are fixed. It may or may not be fixed at all.

Examples for Priority & Severity

High Priority & Low Severity

Company logo is not properly displayed on their website.

High Priority & High Severity

Suppose you are doing online shopping and filled payment information, but after submitting the form, you get a message like "Order has been cancelled".

Low Priority & High Severity

If we have a typical scenario in which the application get crashed, but that scenario exists rarely.

Low Priority & Low Severity

There is a mistake like "You have registered success" instead of successfully, success is written.

BUG LEAKAGE BUG RELEASE

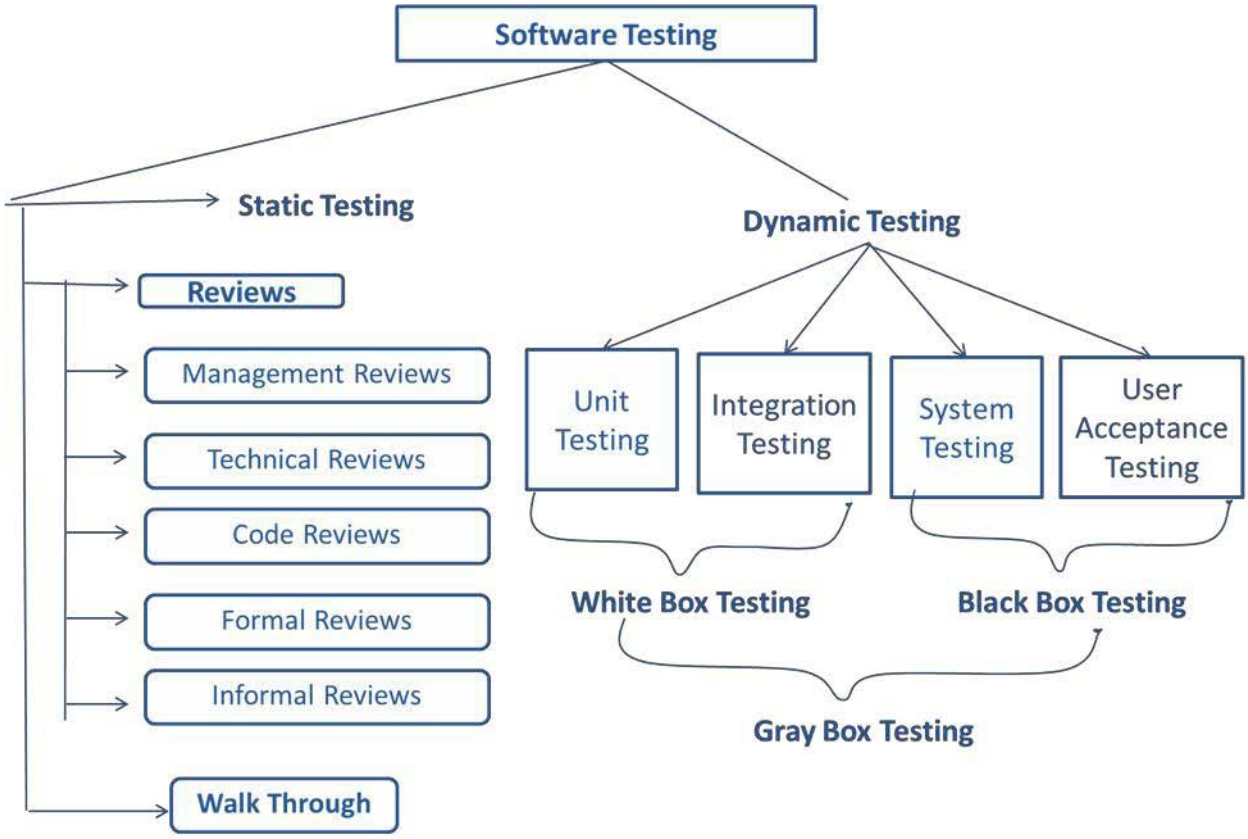
BUG LEAKAGE

When customer or end user discovered a bug which can be detected by the testing team or when a bug is detected which can be detected in previous build then this is called as Bug Leakage.

BUG RELEASE

Is when a build is handed to testing team with knowing the defect is present in the release. The priority and severity of bug is low. It is done when customer want the application on the time. Customer can tolerate the bug in the released then the delay in getting application and cost involved in removing that bug. These bug are mentioned in the Release Notes handed to client for the future improvement chances.

Types of Testing



STATIC TESTING (VERIFICATION)

Static testing is a process of checking we are developing right system or not. Verification performed without executing the system's code and functionality and non- functionality of the system. Also called static analysis.

Walkthrough

Knowledge transfer section are call walkthrough in other word a step by step section conducted by domain expert about a subject or about process to provided common understanding to the team member is call walkthrough.

Review

Examining a project related word or a process related word is calling a review. You can review requirement specification, design specification and code etc. In fact test cases can also be verified.

Objectives of reviews:

- To find the defects in requirements
- To find defects in design
- To identify deviations in process
- To provide valuable suggestions to improve the process.

Types of reviews

1) Management reviews
2) Technical reviews
3) Formal reviews
4) Informal reviews
5) Load review

1. Management reviews

These reviews will be conducted by the high level management or by the middle level management to identify deviations in planed effort to actual effort. If any deviations are identified management will take a necessary corrective action to cover these slippages. So that you can deliver the application in time to customer.

Slippage: It is a deviation between planned works to actual works.

"Daily or weekly project status meeting are nothing but management reviews"

2. Technical reviews

These reviews are conducted among technical member to decide the best approach of implementing a task is any queries on that task is any queries on that task.

3. Formal review

If a review is conducted with a prior plan and by following systemic procedure and proper documentation then these reviews are called formal reviews. Inspection and audits are the best examples for formal reviews.

Inspection:

If a formal review is conducted while executing a task then it is called inspection.

Audit:

If a formal review is conducted after completion of a task to conform that the task is accomplish as per predefined procedures or not is called audit.

4. Informal reviews

If a review is conducted without following any predefined procedure then these reviews are call inspectional reviews. Code reviews are best examples for informal revises.

Code reviews

Review conducted on the source code by developer to check the coding standards is call code review. In general code review will not follow any predefined procedure.

Peer reviews

Reviews conducted among colleagues

DYNAMIC TESTING (VALIDATION)

- Dynamic testing is nothing executing and using the Software/System/Product.
- Checking the Structural and Functional of the Software/System/Product
- Structural means Logic of the program
- Functional means Functionality & Non- Functionality of the program
- Developer & Tester will implement the Dynamic Testing in Four Levels
  1. Unit Testing.
  2. Integration Testing.
  3. System Testing.
  4. User Acceptance Testing.

Levels of Testing

Unit Testing

- Test each module (program, function, procedure, method) individually checking code of units in application is working as expected or not. It is also called Component Testing or Module Testing.
- Follows a white box testing (Logic of the program).
- Developers will be involved.

Steps for Unit Testing

1. Create an Unit Testing Test Plan.
2. Create an Unit Integration Testing Test Cases
3. Prepare the Test Data for Unit Integration Testing
4. Execute the Test Cases
5. Bug fixing and tracking the errors
6. Repeat the cycle as necessary.

Entry Criteria for Unit testing phase are:

1. BRS, LLD and Test Data Should be reviewed and approved
2. Environment should be ready
3. Training for developers should be completed
4. Unit testing coding should ready

Exit criteria for the Unit testing phase are:

1. Unit test has been completed
2. All priority a bug have been fixed and closed
3. Internal documentation has been updated to reflect the current state of the product.

Integration testing

- Once unit testing is conducted the programmer will combine all modules together and will check the interaction between those modules i.e. communicate between the modules this is called integration testing. It also called as Component Integration Testing.
- Follows a Grey box testing (Combination of WBT & BBT)
- Both Testers and Developers will be involved

Steps for Integration Testing

1. Create an Integration Testing Test Plan.
2. Create an Integration Testing Test Cases
3. Prepare the Test Data for Integration Testing
4. Execute the Test Cases
5. Bug fixing and tracking the errors
6. Repeat the cycle as necessary

Entry Criteria for Integration testing phase are:

1. Unit test should be completed and all the major bugs should be fixed.
2. HLD and Test Data should be reviewed and approved.
3. Environment should be ready.
4. Training for developers should be completed.
5. Integration coding should ready.
6. Sanitary testing completed.

Exit Criteria for Integration phase are:

1. Integration Testing has been completed.
2. All priority a bug have been fixed and closed.
3. Internal documentation has been updated to reflect the current state of the product.

System Testing

- Validating both functional and non-functional requirements of system is called system testing
- Follows the black box testing.
- Only Testers will be involved

Steps for System Testing

1. Create a System Testing Test Plan.
2. Create a System Testing Test Cases
3. Prepare the Test Data for System Testing
4. Create and Execute the Automation Test Cases (If Required)

5. Execute the Test Cases
6. Bug fixing and tracking the errors
7. Repeat the cycle as necessary

Entry Criteria for System testing phase are:

1. Integration testing should be completed and all the major bugs should be fixed.
2. Smoke Testing Completed.
3. SRS, Test Plan, Test Scenario, Test Cases, Test Data should be reviewed and approved.
4. Environment should be ready.
5. Training for developers should be completed.
6. Build should ready.
7. Sanitary testing completed.

Exit Criteria for the System testing

1. System Testing has been completed.
2. All priority a bug have been fixed and closed
3. Internal documentation has been updated to reflect the current state of the product.

User Acceptance Testing (UAT)

- Once the system testing is done then customer or domain expert will test the application to conform does the application is good for live environment or not.
- It is depend on the business scenario.
    - **Alpha testing** - It is first level of acceptance testing conducted by domain expert or customer at developer premises.
    - **Beta testing** - It is last level of acceptance testing conducted by domain expert or customer at customer premises to check whether the application is good for live usage.

Entry Criteria for UAT phase

1. System Testing and End to End Testing should be completed.
2. All priority a bug have been fixed and closed
3. Defect Report should be ready.
4. Environment should be ready.
5. User acceptance testing should commence upon approval of the user acceptance Test Lead, Quality Assurance Test Lead and Data Architect/Project Manager who shall ensure that UAT test entrance criteria have been met.

Exit Criteria for UAT phase

1. UAT testing shall be considered complete upon sign-off and approval of the Production Support/Release Manager, Project Manager, UAT Test Lead and business owner who shall ensure that UAT test exit criteria have been met.
2. All required test types have been completed.

TESTING TECHNIQUES

1. White Box Testing
2. Black Box Testing
3. Incremental Testing (Gray Box Testing)

White Box Testing Techniques

White box testing is based on knowledge of the internal logic of an application's code. Tests are based on coverage of code statements, branches, paths and conditions.

- Statement coverage – execute all statements at least once.
- Decision coverage - execute each decision direction at least once.

- Condition coverage – execute each decision with all possible outcomes at least once

Black Box Testing Techniques

Black Box Testing is not based on any knowledge of internal design or code. Tests are based on requirements and functionality

1. Equivalence Partitioning
2. Boundary Analysis
3. Error Guessing

1. Equivalence Partitioning

- It is used to combine same type of test cases related to single functionality / feature / module.
- A subset of data that is representative of a larger class
- For example, a program which edits credit limits within a given range ($10,000 - $15,000) would have 3 equivalence classes
    - Less than $10,000 (invalid)
    - Between $10,000 and $15,000 (valid)
    - Greater than $15,000 (invalid)

2. Boundary Value Analysis

- A test data selection technique in which values are chosen to lie along data extremes. Boundary values include maximum, minimum, just inside/outside boundaries, typical values, and error values.
- A technique that consists of developing test cases and data that focus on the input and output boundaries of a given function
- In the same credit limit example, boundary analysis would test:
    - Low boundary plus or minus one ($9,999 and $10,001)
    - On the boundary ($10,000 and$15,000)
    - Upper boundary plus or minus one ($14,999 and 15,001)

3. Error Guessing

- A test design technique where the experience of the tester is used to anticipate what defects might be present in the component or system under test as a result of errors made, and to design tests specifically to expose them.
- Based on the theory that test cases can be developed based on experience of the Test Engineer
- For example, in an example where one of the inputs is the date, a test engineer might try February 29,2000 or 9/9/99

Incremental Testing

- A disciplined method of testing the interfaces between unit-tested programs as well as between system components
- Testing where components or systems are integrated and tested one or some at a time, until all the components or systems are integrated and tested.
- Incremental Testing Types
    - Top-down
    - Bottom-up

Top-Down

- An integration testing technique that tests the high-level components first using stubs for lower-level called components that have not yet been integrated and that stimulate the required actions of those components.
- Begins testing from the top of the module hierarchy and works down to the bottom using *interim stubs* to simulate lower interfacing modules or programs
- **Stub** is a software component that usually minimally simulates the actions of called components that have not yet been integrated during top-down testing.

Bottom-Up

- An integration testing technique that tests the low-level components first using test drivers for those components that have not yet been developed to call the low-level components for test.
- Begins testing from the bottom of the hierarchy and works up to the top Bottom-up testing requires the development of *driver modules* which provide the test input, call the module or program being testing, and display test output
- **Driver** is a software component or test tool that replaces a component that takes care of the control and/or the calling of a component or system.

Functional Testing

1. Unit Testing

Test each module (program, function, procedure, method) individually checking code of units in application is working as expected or not. It is also called Component Testing or Module Testing.

2. Integration Testing

Once unit testing is conducted the programmer will combine all modules together and will check the interaction between those modules i.e. communicate between the modules this is called integration testing. It also called as Component Integration Testing or Interface Testing.

3. System Testing

Validating both functional and non-functional requirements of system is called system testing.

4. User Acceptance Testing

Once the system testing is done then customer or domain expert will test the application to conform does the application is good for live environment or not.

5. Alpha testing

It is first level of acceptance testing conducted by domain expert or customer at developer premises.

6. Beta testing

It is last level of acceptance testing conducted by domain expert or customer at customer premises to check whether the application is good for live usage.

7. Smoke Testing or Sanitary Testing

It is a group of test cases that establish system is stable and all major functionality is present and works under “normal” conditions.

8. Regression Testing

It is process of testing to make sure that (changes made in fixing bug should not affect the other part of the program) Old programming still works well with new changes.

Regression testing is carried out to examine whether the new code works properly and has not damaged any previously-working functionality

9. Retesting

Retesting is a testing a whether a specified bug has been fixed by the developer or not. That is testing a particular bug alone to find out whether the bug has been fixed or not

10. End to End Testing

Validating all core functionality of the system right from the beginning till end and its data integration as well.

11. Mutation Testing

It is a process of wontedly injecting defects by developer to check whether testers are testing application properly or not.

12. Exhaustive Testing:

Testing functionality with all positive and negative perception is called exhaustive testing and it is also called as in detail testing.

13. Ad Hoc Testing

Testing the application without following any Pre-planned procedure is called Ad-hoc testing.

Non-Functional Testing

1. Usability testing

Validating the user friendliness of the system i.e. how easily end user can understand and use the system.

2. Performance testing

It is a process of checking various efficiency characteristics of the system such as response time, through put, load transaction per minute, transaction needs, resources consumption and stress.

3. Compatibility testing

Validating whether the application is compatible with various hardware and software environments or not.

4. Load testing

Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the systems response time degrades or fails.

5. Stress testing

Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements.

6. Security testing

Validating whether all security conditions are properly implemented in the application or not.

7. Exploratory testing

Exploring the application adding or modifying the existing test cases for better testing.

8. Authorization Testing

Checking does the application has provision to define login account, setting, and changing privileges or not.

9. Authentication Testing or Confidentiality Testing

Validating does the sys is able to recognize the register users and providing right information to the right users or not is called authentication testing.

10. GUI Testing

Checking does the application user inter phase designed professionally or not.

11. Error guessing

With prior experience and knowledge tester will guess the area in application where there could be an error and tester will test the application with that perception.

12. Recovery Testing

Validating does the sys having a permission of back up ,restore options when you restore the back up all the data getting backed up or not and also testing the sys how well it is handling unexpected situations like power failures and sys crashes.

13. Globalization Testing:

Validating does the sys having privatization to change languages. Date and time format according global requirements.

Test Metrics

INTRODUCTION

Metrics are defined as "standards of measurement" and have long been used in the IT industry to indicate a method of measuring the effectiveness and efficiency of a particular activity within a project. Also known as **Software quality metrics**.

There are several test metrics identified as part of the overall testing activity in order to track and measure the entire testing process. These test metrics are collected at each phase of the testing life cycle /SDLC and analyzed and appropriate process improvements are determined and implemented as a result of these test metrics that are constantly collected and evaluated as a parallel activity together with testing both for manual and automated testing irrespective of the type of application.

Complete Metrics data helps in building the "Trustworthy, Capable, Reliable and Predictable" Product or Application.

PROCESS Related Test Metrics

Defect removal efficiency, Review efficiency, etc., Test Case Efficiency, Test Efficiency, Test Effectiveness, etc.

PRODUCT Related Test Metrics

Defect Density, Cumulative weighted defect density, Defect Severity index, etc.

Objectives of Test Metrics

This metric indicates the quality of the product under test. It can be used as a basis for estimating defects to be addressed in the next phase or the next release. This is an Organizational Measurement.

Test Metrics is a mechanism to know the effectiveness of the testing that can be measured quantitatively. It is a feedback mechanism to improve the Testing Process that is followed currently.

When to introduce Test Metrics?

1. Identifying Test Metrics is done at the beginning of the test project.
2. Test Metrics are collected at each phase of the testing.

What are the inputs for Test Metrics?

1. Quantity of test cases prepared / performed.
2. Quantity of Defects found.
3. Size of code developed in KLOC.

Formulae for frequently used Test Metrics

1. Defect Removal Efficiency (DRE) = (E/ E+D) x 100 where

E = Pre-delivery errors (detected during all QC / QA) activities D= Post –delivery Defects

Objective

Reduce Pre and Post Delivery Defects in all Deliveries. Indicates the efficiency of defect removal methods, as well as indirect measurement of the quality of the product.

Unit

Average (%)

2. Review efficiency (RE) = (No. of Defects found in Review) x 100 / Total No. of Defects found before Delivery (both Reviews and Testing)

Objective

Reduce Pre-Delivery Defects.

Unit

Average **(%)**

3. Defect Density = Defects found/Size in KLOC

Objective

This metric indicates the quality of the product under test. It can be used as a basis for estimating defects to be addressed in the next phase or the next version. Reduce defect leakage while coding/design.

Unit

No. of Defects per unit size.
eg: Defects per KLOC

4. Cumulative Weighted Defect Density = No. Of weighted defects (review issues + testing defects) / Product Size (actual in KLOC)

Here **Weighted defects = Major defects + (Minor defects)/3 (Trivial defects)/5**

Objective

To know the weightiness of the issues found during review and testing phase.

Unit

No. of Defects per unit size

5. Defect Severity Index = [Sum of (Defect * Severity Level)]/Total number of defects

Here A number is assigned against each severity level: **4 (Critical), 3 (Major), 2 (Medium), 1 (Minor)**

Objective

Provides a direct measurement of the quality of the product—specifically, reliability, fault tolerance and stability.

Unit

No unit only a real number

6. Test Case Efficiency = (Number of defects detected / Number of test cases run)* 100 Objective

To know the efficiency of the test cases that are being executed in the testing phase. The quality of the test cases can be determined.

Unit

Average **(%)**

Test Efficiency & Test Effectiveness:

These are two vital metrics, which always come to my mind, when I think about metrics.

Let us talk about the meaning of efficiency, most of the definitions state efficiency as the percentage of ratio of output to input of any system using unit measures (delivered /supplied). But I agree to the following definition.

Noun

The ratio of the effective or useful output to the total input in any system

Therefore, efficiency is an attribute which means to maximize the useful output for a given input reducing wastage or losses. Efficiency cannot be more than a 100%, in a sense that a 100% efficient system will have zero losses

7. Test Efficiency (TE)

A Consultant as part of any engagement would want to deliver, the solution on time, within budget, on specs, having an acceptable level of quality, quantified by the customer. To achieve this goal, the team should work with efficiency, which is to constantly show progress to the effort put in. Test efficiency is a quality attribute of the test team, to carry out all testing activities in an efficient manner saving cost and time. To mention few of test efficiency focus areas,

1. Resources
2. Tools
3. People
4. Process
5. Time

Let us look at the broader picture of the metric, test efficiency is not only about test execution alone, but all or most of test activities, like test planning, comprehension, test cases creation, review, execution, defect tracking and closure. TE is not just one single derivation but a number of calculations at each phase and activity of testing. What activities or phases one is interested in, what is particularly measured, depends on lot of other things, the type of project, complexity, availability of resources, the situation, customer requirement etc.

8. Test Effectiveness:

Test effectiveness (TEF) can be calculated for specific set of test activities. For e.g.: Test preparation efficiency will be the time taken for 'X' number of test cases to be prepared, reviewed and reworked to finalize them. There is a catch here, the quality standards of the test cases should be predefined by using defining standards, as I have tried to state a few of them here.

1. The Test cases are complete with respect to Use Cases on which they are based.
2. A tester should be able to execute this test case using only this test case and any directly referenced items given the proper software and hardware configuration.
3. Test data must be specific. For example, don't say "select any menu option to navigate out of current page." Say "Click the Back button." Don't leave any test data to the imagination of the tester.
4. Usually, each test step should contain a single action. E.g. "Save" and "Search" functionalities should be split into 2 steps
5. Test Case names follow the agreed upon naming convention
6. No grammatical mistakes

Let us assume that there are two test teams with equal number of resources with comparable skill set, functional expertise etc. working on the same product, and If test team 'A' has prepared 400 Test cases with agreed quality, for the product in 5 days and for the same product if test team 'B' prepares 400 Test cases in 4 days, then which team can we say is efficient? Definitely team B is efficient, but there is no guarantee that Test team 'B' is effective, we are not sure how many defects can team 'B' uncover as compared to team 'A'. That is our next top.

Test Effectiveness (TEF) Efficacy in contrast to efficiency is focused to just produce the desired result or effect or achievement as such and not the resources or time spent.

Noun Effectiveness means the capability of producing an effect.

Let us look at what Test effectiveness is, Test effectiveness of a technique or a system or a team is the ability to find defects and isolate them, from a product or deliverable. Test

effectiveness is to ensure quality and close the two quality gaps, namely producer's quality gap and customer's quality gap. As definition of quality goes, quality is both process and product quality which is meeting customer requirements and conformance to product specification. These metrics should be quantified, as they closely relate to quality, and for many people the term quality is relative.

Defect originated/Injected

Defect found in phase	Requirement	Design	UT	IT	ST	Total
Requirement	2					**2**
Design	3	12				**15**
UT	2	1	22			**25**
IT	1	2	4	15		**22**
ST	1	2	2	2	6	**13**
Production	1	1	1	2	3	**8**
Total	**10**	**18**	**29**	**19**	**9**	**83**

The table shows the defect origin, on the X-axis, where a defect was injected, where he belongs to? And on the Y-axis, where the defect was detected.

Let us calculate the test effectiveness of the Integration testing activity

Total number of defects of all origin found during Integration testing activity = 22
Total number of defects existing while entering in IT = (10+18+29) – (2+15+25) = 15
Total number of defects injected in the current stage = 19
Effectiveness of IT test phase = Total defects found in this phase/ (No of defects existing + injected)
Effectiveness = 22/ (15+19) * 100 = 64.70%

Things to Remember

1. Keep Test Metrics Simple
2. Create Meaningful Metrics
3. Use Metrics to Manage the Project
4. Track Metrics

Conclusion

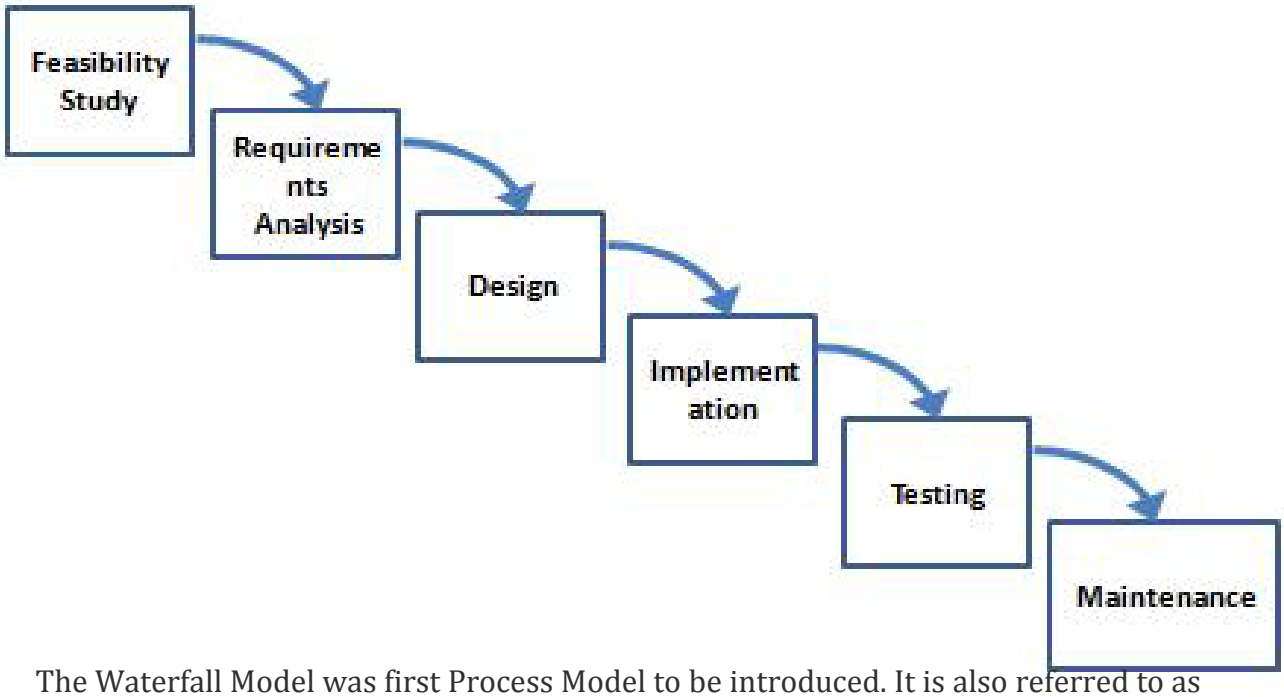
It is not enough to have a set of metrics that are tracked on a regular basis. The metrics must also be reviewed and analyzed regularly, as they can provide value feedback during and after a software development project.

Acronym

DRE - **D**efect **R**emoval **E**fficiency
IT - **I**ntegration **T**esting
KLOC - **K**ilo **L**ines **o**f **C**ode
QC - **Q**uality **C**ontrol
QA - **Q**uality **A**ssurance
RE - **R**eview **E**fficiency
SDLC - **S**oftware **D**evelopment **L**ife **C**ycle
ST - **S**ystem **T**esting
TE - **T**est **E**fficiency
TEF - **T**est **E**ffectiveness
UT - **U**nit **T**esting

TESTING MODELS

1. WATER FALL MODEL



The Waterfall Model was first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model the testing starts only after the development is complete. In **waterfall model phases** do not overlap.

ADVANTAGES

- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.

DISADVANTAGES

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.

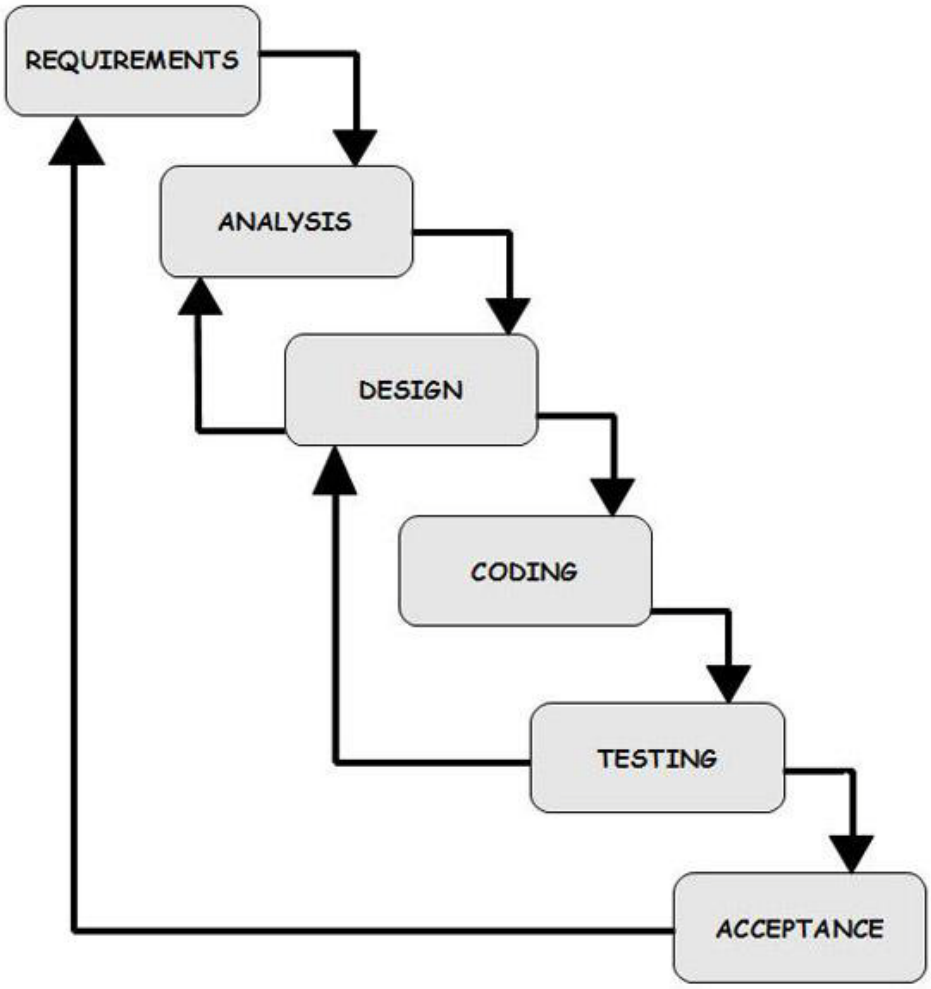
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

When to use the waterfall model

- This model is used only when the requirements are very well known, clear and fixed.
- Product definition is stable.
- Technology is understood.
- There are no ambiguous requirements
- Ample resources with required expertise are available freely
- The project is short.

Very less customer enter action is involved during the development of the product. Once the product is ready then only it can be demoed to the end users. Once the product is developed and if any failure occurs then the cost of fixing such issues are very high, because we need to update everywhere from document till the logic.

2. MODIFIED WATER MODEL



- Systematic & Sequential approach model.
- Similar to WATERFALL model.
- Development always in forward direction.
- Major difference is flexible when compare to pure WATERFALL model, Request for change in requirements at any stage
- It allows to go back to any phase as required and continue the process.

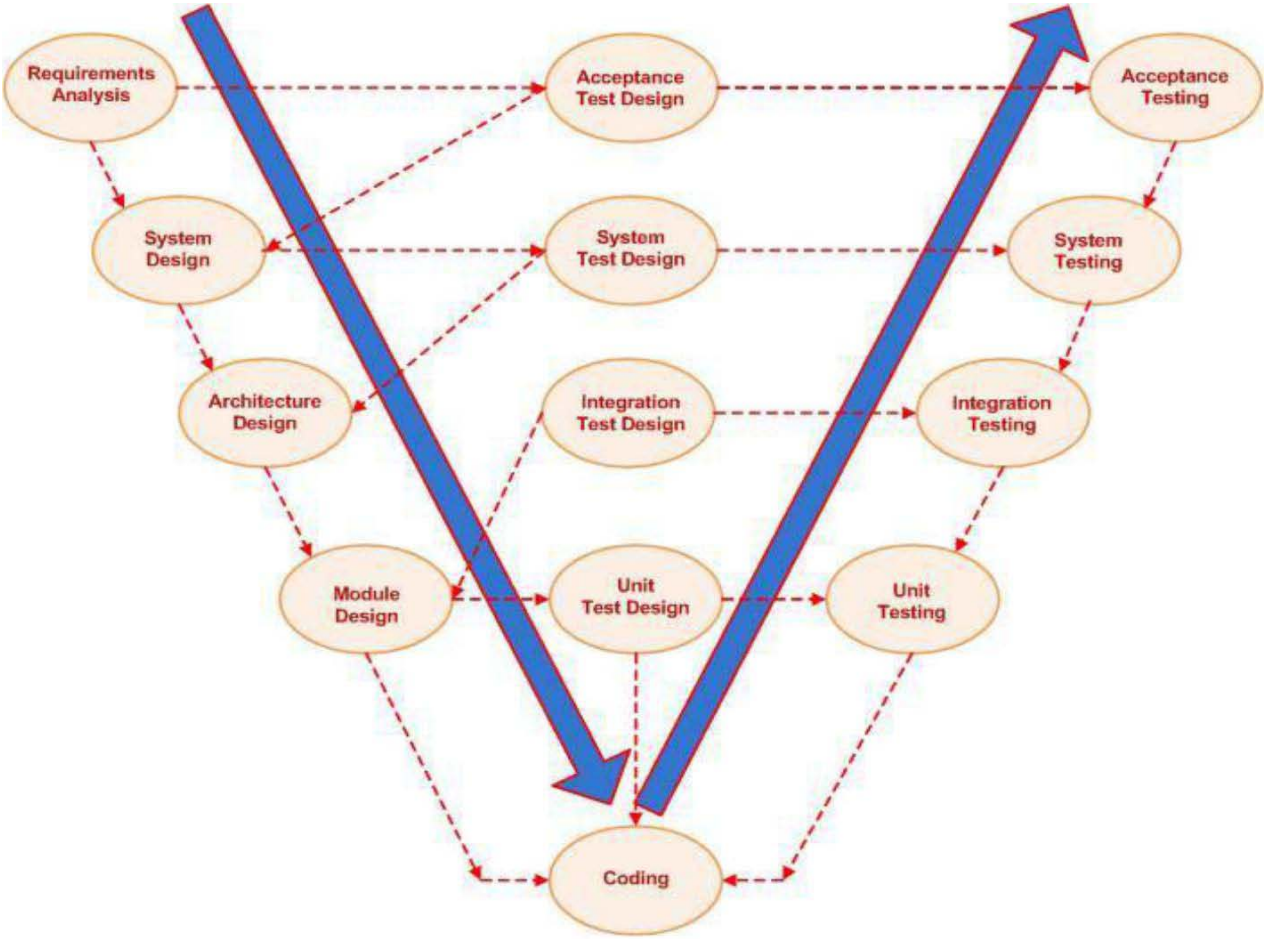
ADVANTAGES

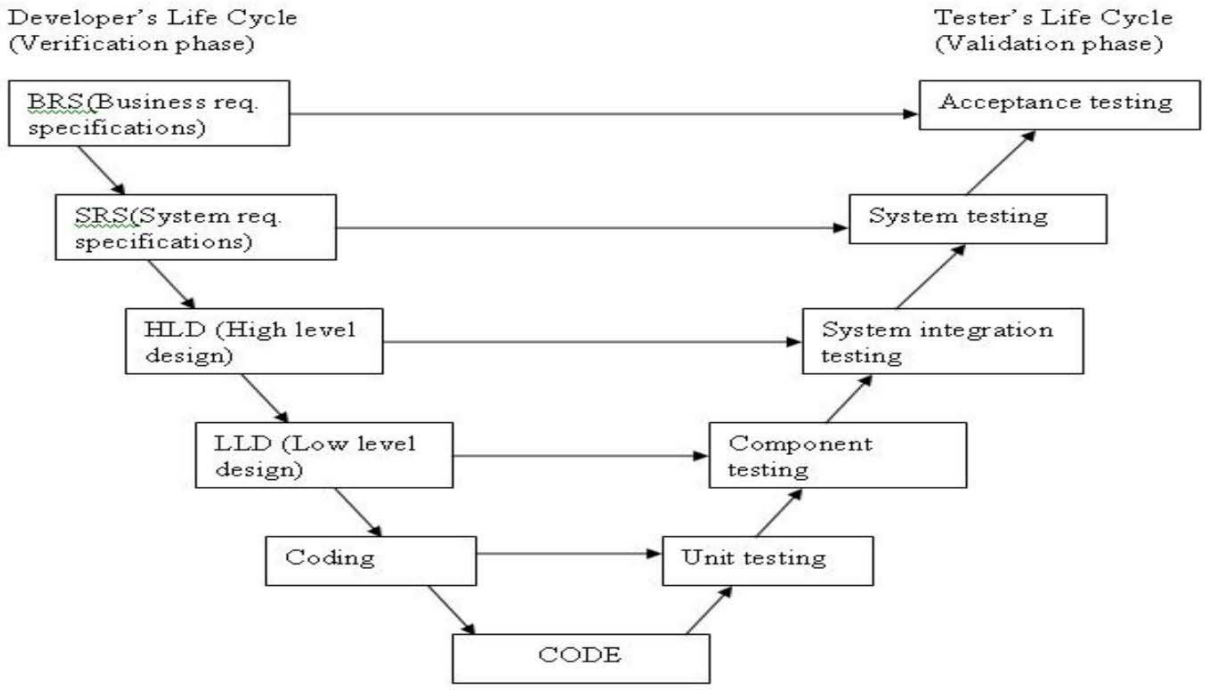
- Useful for the projects where requirement are well known and stable.

DISADVANTAGES

- Comparing to waterfall model, this is not a simple approach

3. V-MODEL





The V-Model (or VEE model) is a systems development model designed to simplify the understanding of the complexity associated with developing systems.

The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing.

"**V**" stands for **V**erification & **V**alidation

ADVANTAGES

- Simple and easy to use.
- Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
- Proactive defect tracking – that is defects are found at early stage.
- Avoids the downward flow of the defects.
- Works well for small projects where requirements are easily understood.

DISADVANTAGES

- Very rigid and least flexible.
- Software is developed during the implementation phase, so no early prototypes of the software are produced.
- If any changes happen in midway, then the test documents along with requirement documents has to be updated.

WHEN TO USE THE V-MODEL

- The V-shaped model should be used for small to medium sized projects where requirements are clearly defined and fixed.
- The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise.
- High confidence of customer is required for choosing the V-Shaped model approach. Since, no prototypes are produced, there is a very high risk involved in meeting customer expectations.

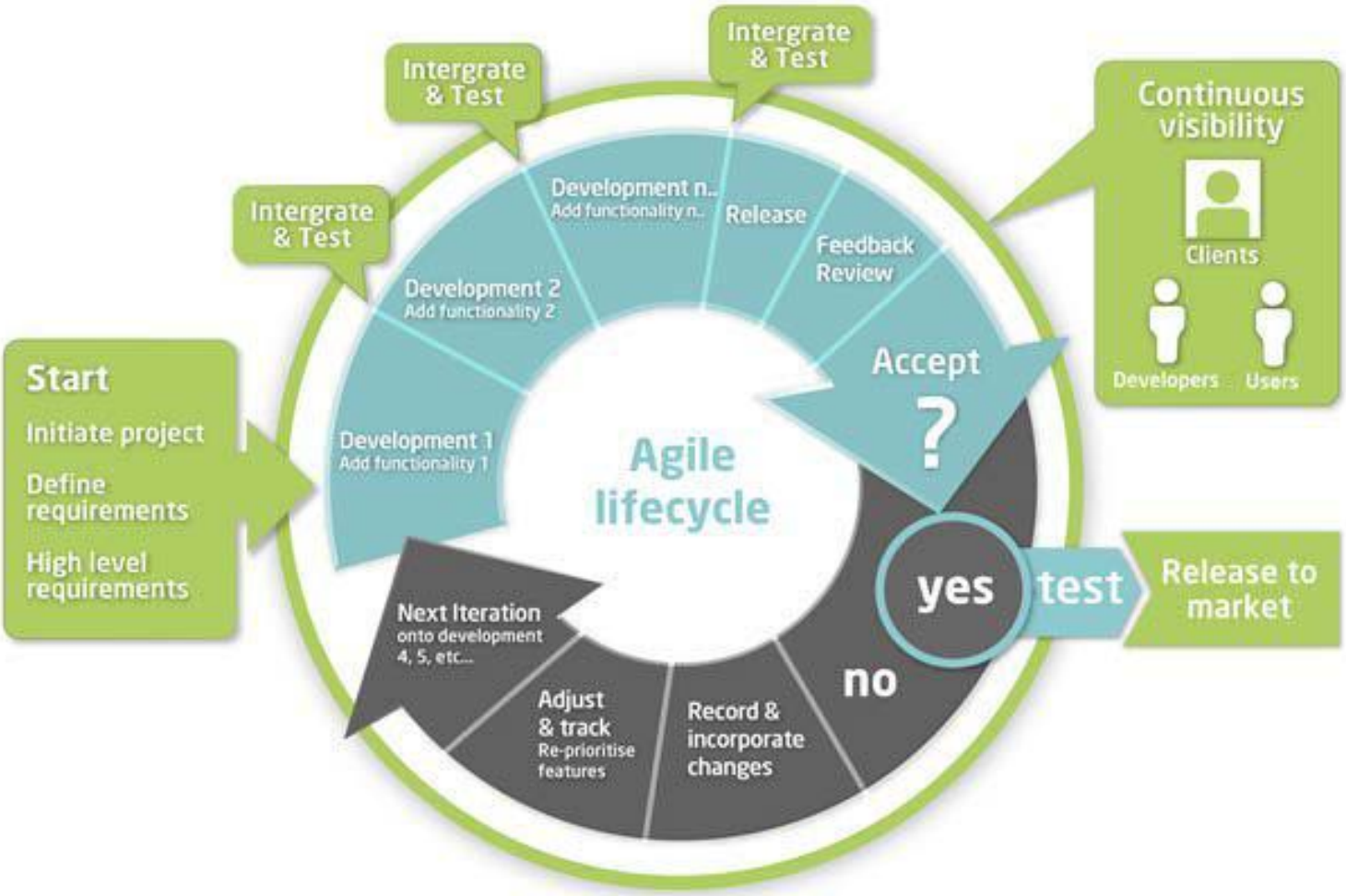
AGILE METHODOLOGIES (SCRUM METHODOLOGIES)

In rugby, 'scrum' (related to "scrimmage") is the term for a huddled mass of players engaged with each other to get a job done.

*Everything depends on **VELOCITY & TIME***

D = V x T

D – Done Feature
V – Resource Capacity
T – Time Line
Timeline will be in weeks not more than 3weeks
Duration of each work is called as **SPRINT**



SECTION 1

SPRINT PLANNING

- Attendances – **Scrum Master, Scrum Developer, Product Owner**

  Product Owner: *In Scrum, the Product Owner is responsible for communicating the vision of the product to the development team.*

  Scrum Master: *The Scrum Master acts as a liaison between the Product Owner and the team. The Scrum Master does not manage the team. Instead, he or she works to remove any impediments that are obstructing the team from achieving its sprint goals.*

  Scrum Developers: *In the Scrum methodology, the team is responsible for completing the work.*

- Discuss on **Product backlogs**:
- Output of this meeting is finalized the **Product Backlogs** document for this **Current Sprint**

SECTION 2

SPRINT COMMITMENT

- Analyzing the User Stories / Requirement
    - ***What to do***
    - ***How to do***
    - ***Who to do***
    - ***When to do***
- **Scrum Master, Scrum Developer and BA** will give the commitment to **Product Owner**
- **Scrum commitment** will be based on Resources
- **Scrum Developers** (Dev Team & Testing Team)

SECTION 3

SPRINT EXECUTION

- **Testing Team**
    - Test Scenario Preparation
    - Test Case Preparation
    - Test Case Review
    - Test Case Execution
- **Daily Standup Meetings**
    - Duration 15mins
    - Scrum master will lead this
    - Task should be boarded on Scrum Board
    - What have you done – Yesterday / Today
    - What are you going to work – Today / Tomorrow
    - Obstacles
    - Every Scrum Developers has to answer the above questions

SECTION 4

SPRINT REVIEW MEETING

- What we have done on Current Sprint
- `Scrum Master, Scrum Developers will demonstrate the product to Product Owner
- Product Review

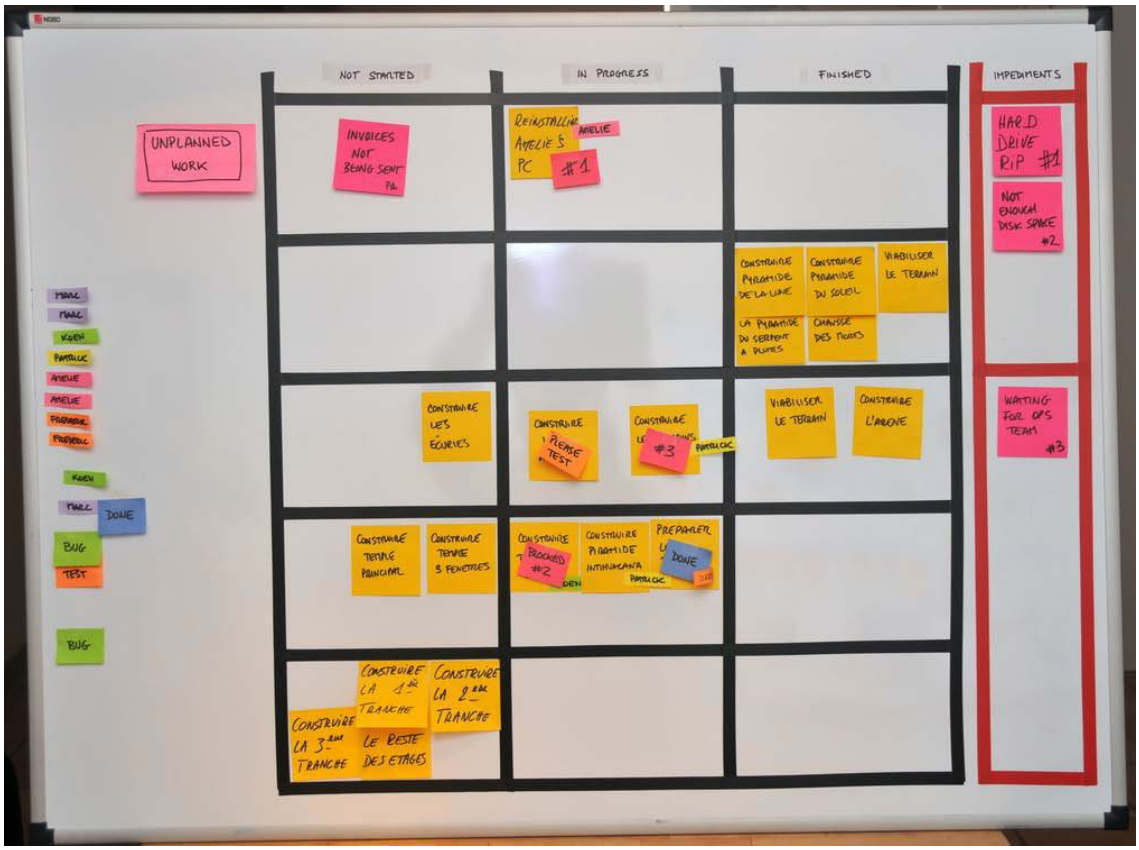
SECTION 5

SPRINT RETROSPECTIVE

- The team discusses the just-concluded sprint and determines what could be changed that might make the next sprint more productive.
    - *What went well during the sprint cycle? – **KEEP DOING***
    - *What could we do differently to improve? – **START DOING***
    - *What went wrong during the sprint cycle? – **STOP DOING***
- **Key elements of the sprint retrospective:**
    - Process improvements are made at the end of every sprint. This ensures that the project team is always improving the way it works.

- The retrospective is a collaborative process among all members, including the team, the product owner, and the ScrumMaster.
- All team members identify what went well and what could be improved.
- The team members discuss the process that they are following and give any suggestions for improvement.
- The team members discuss any other ideas that could improve their productivity.
- The ScrumMaster prioritizes actions and lessons learned based on team direction.
- The retrospective supports team formation and bonding, particularly as any areas of conflict can be identified and dealt with.
- The retrospective helps build the team's sense of ownership and its self-management.
- Process Review

SCRUM BOARD

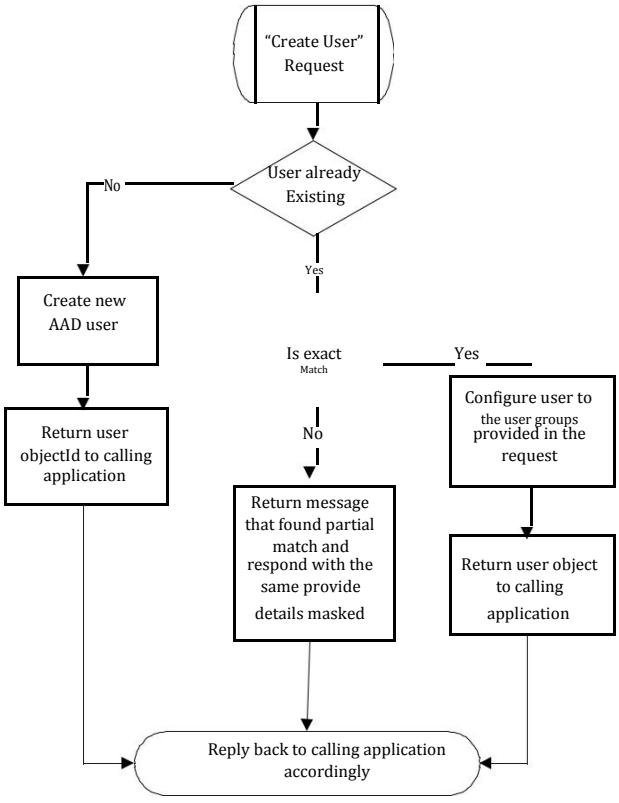


Windows Azure Active Directory User Maintenance Services

Version Control

Version	Date	Description of Change	Author
0.1	21st May 2015	Initial version	Shivanandini Kuncham
0.2	25th May 2015	Updated with review comments	Shivanandini Kuncham
0.3	11th June 2015	Added requestorID in the input request message fields	Shivanandini Kuncham
0.4	29th June 2015	Till date updates added in orange	Shivanandini Kuncham

Create User Service



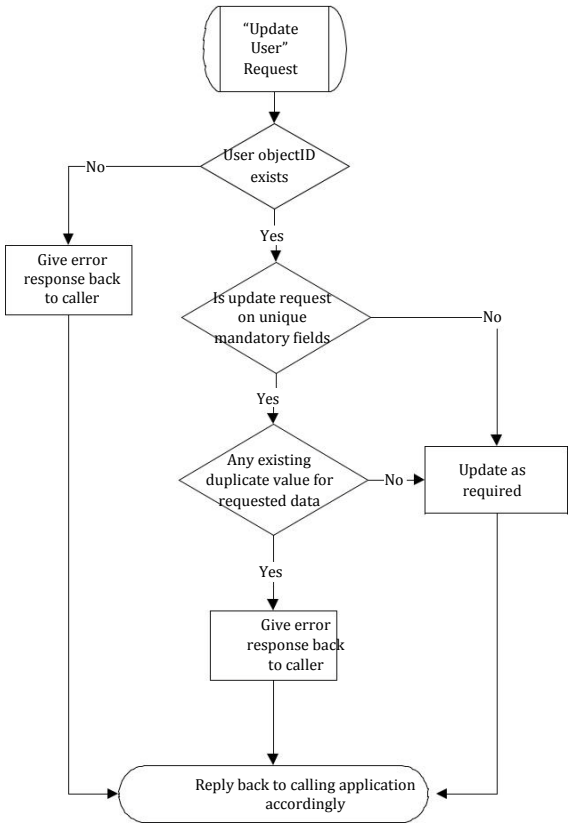
Steps: Search if there is any existing user with the same UserPrincipalName

1. If user exists & if it is exact match (Match with display name, mobile, otherMails) then assign that user with the new group and respond with the user objectID + message that you linked to existing user as you found exact match.

a. Do not link disabled user (*accountEnabled = false*) associated with some groups. In case disabled user without any groups then enable it, link it and add basic license if not associated with one already.
b. otherMails is collection of emails. In case one email address matches it should be considered as exact match
c. In case requested group does not exist throw error back to calling application.
d. If user already associated with the group convey the same in the response
e. On the response back to caller reply with displayname, objectID, masked Mobile & otherMails and all groups user is associated with

2. If user exists & If it is NOT exact match (Case when one or more of these properties not matching - display name, full phone number/mobile, primary email address) then respond back with the partial match user displayname, objectID, masked mobile & otherMails (eg., ******6538, s*********@yahoo.com). Also provide details of all the groups partial match users are associated with.
3. If user does not exist -
   a. Create a new user with the details provided in the input request
   b. Assign user to the groups requested. If groupID does not exist send error response
   c. Assign basic license -- Refer Appendix B section below
   d. Enable MFA by default, unless value provided in input request to override. – Not as part of first release. Part of JIRA EAI-2087
   e. Set password to value passed in input. If empty from source use a random value generated within ES for the same (send the same in response too). (Need to comply with the password policy in AAD)
4. One user create request for a partner use

Update User Service (for both user details and user name)

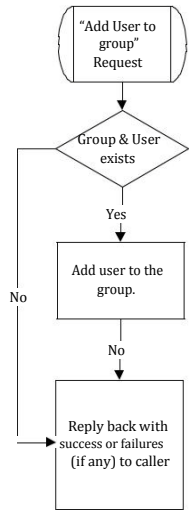


Steps:

1. If objectID does not exist respond with error to the calling application

2. In case requested update is for UserPrincipalName, display user then check if the requested value is already associated with another user, if not then only update otherwise respond back with error to calling application.
3. For updates to any other field update accordingly
4. One update user request per partner user
5. As per new input structure XSD adding users to group is only possible through add user to group service and not through update user service.

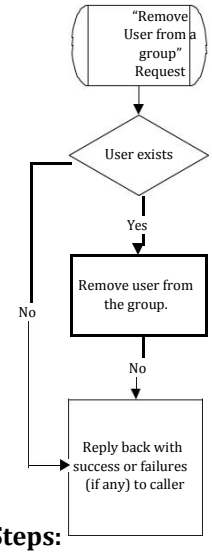
Add User to Group Service



Steps:

1. If user or group do not exist, then reply with error to calling application. In case of user do not exist check for group too and provide consolidated response
2. If user & group exists, add user to the group. Repeat for all the groups mentioned in the request (if any). Respond back with success message
3. One request per partner user but can be with multiple groups

Remove User from a Group Service



Steps:

1. If user do not exist, then reply with error to calling application
2. If user exists, remove user from the group mentioned. Repeat for all the groups mentioned in the request (if any). In case one or many of the groups do not exist remove from the groups which exist and reply back indicating the same

3. *After removing user from group if the user is left with no groups then disable the user (accountEnabled = false)*
4. One request per partner user but can be with multiple group

SOFTWARE REQUIREMENTS SPECIFICATION SAMPLE DOCUMENT

OrangeHRM – My Info Module Live Project

Project Functional Requirement Specification , Version 1

This is a sample SRS document for the live project training. Please read this document and use it as a reference for our live software testing project

1. Purpose of the document:

This is **not** a project plan. It is a guide for system architecture and development, not for phasing, timelines or deliverables.

This document is divided into three sections:

- Project Overview
- Information Architecture
- Site Design

2. Project Overview:

2.1 Audience:

This document is intended as a complete guide for ESS-User in using OrangeHRM 3.0. This document is specially designed for non- specialists; specialists may find the document a useful point of reference. By reading this guide, you will learn how to use OrangeHRM through the elements of the graphical user interface and what's behind some of the advanced features that are not always obvious at first sight. It will hopefully guide you around some common problems that frequently appear for users of OrangeHRM.

2.2 Hardware and Hosting:

OrangeHRM's servers will be hosted at X company's site.

OrangeHRM will be hosted on two servers: One to host the actual website and (language)code, and the other to host the (database name)database.

3. Information Architecture

Log in to the OrangeHRM System using your ESS-User account that has been created by the HR Admin

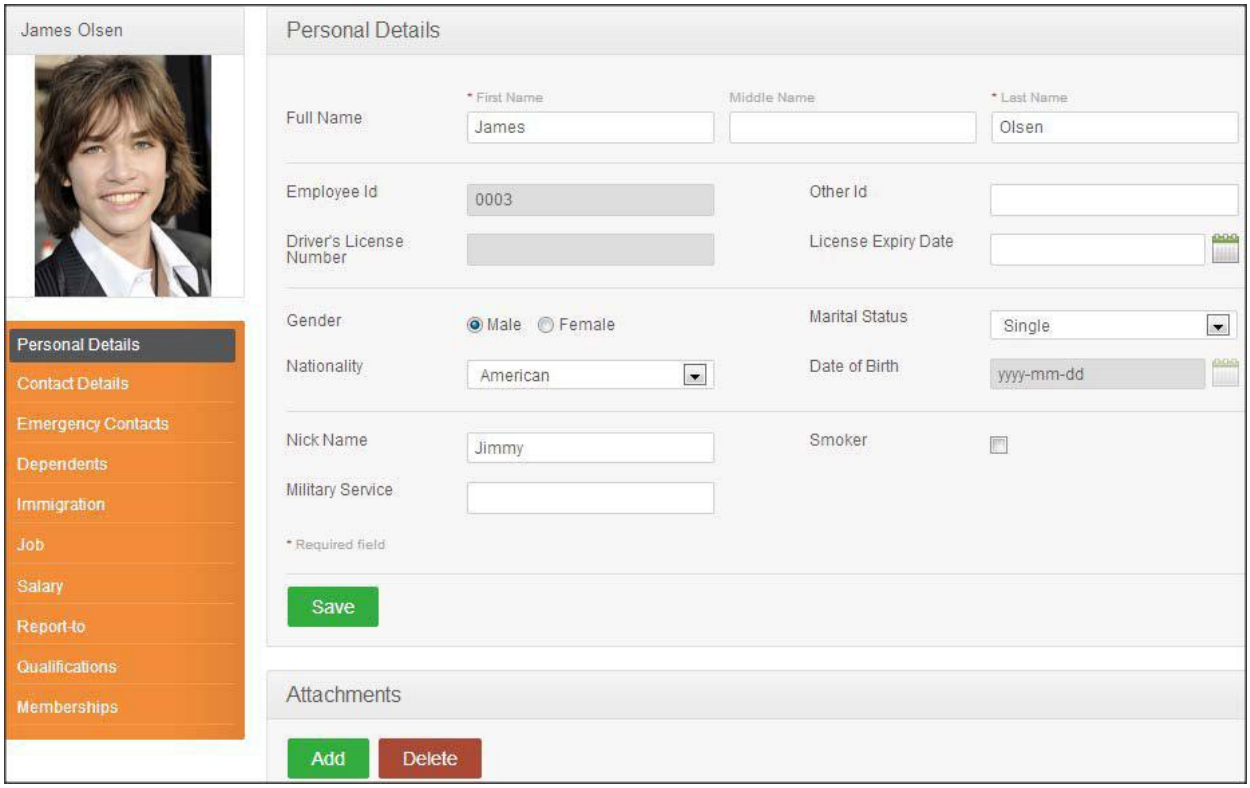
3.1 My info Module

My Info Module is a powerful tool providing employees of the company with the ability to view relevant information such as personal information and updating personal information with an internet enabled PC without having to involve the HR department.

The functionality of this module spans through the entire system, making information available anywhere, anytime. All information is subject to company's defined security policy, where he/she can only view the information he/she is authorized to. An ESS-User can only edit certain fields in the ESS Module, maintaining the security and confidentiality of employee information

3.1.1 My Info Module

When an ESS-User logs into the system for the first time, the first thing they will see is the "Personal Details" screen as shown in Figure 1.1. They are able to edit and enter certain fields.



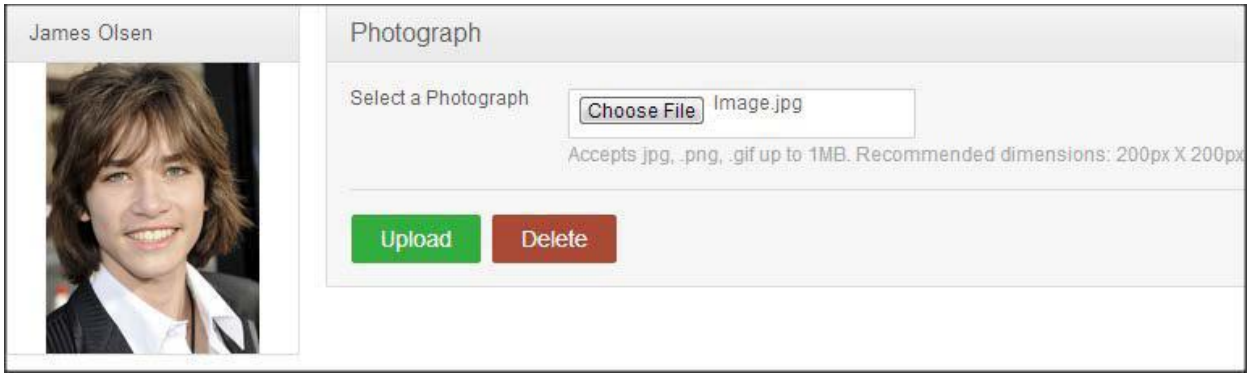
The following are restricted fields where an ESS-User cannot make changes to the following details and need to be populated by the HR Admin and the respective ESS-Supervisor.

Personal Details

- Employee ID
- SSN No
- SIN No
- Driver License No
- Date of Birth

3.1.2 Photograph

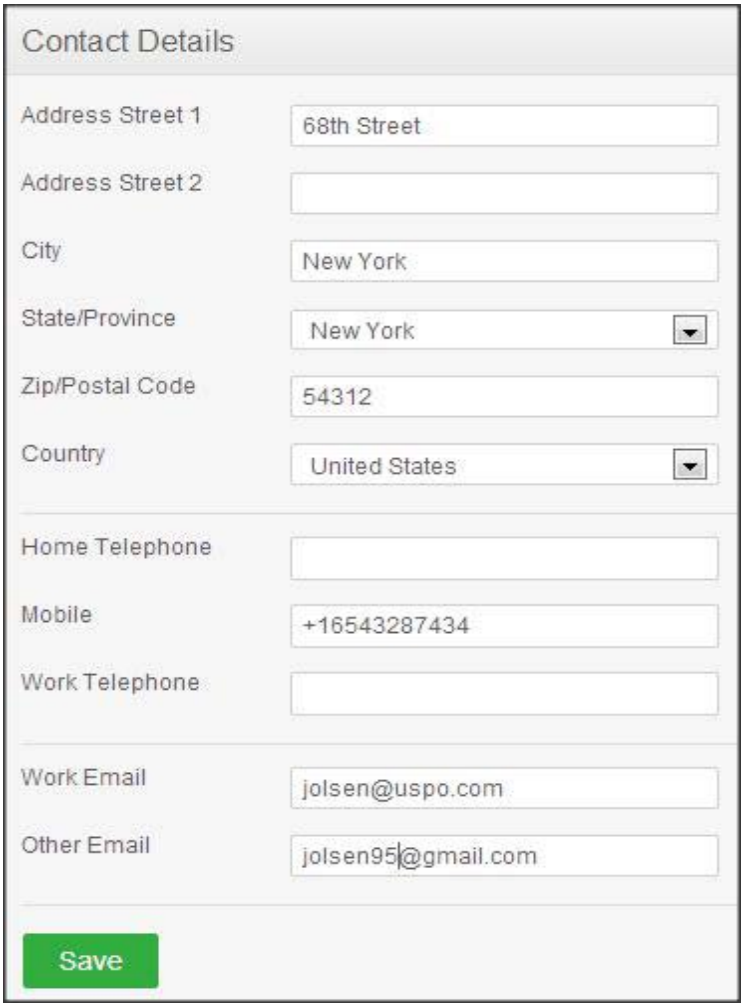
The ESS-User can add a photograph of himself/herself by clicking on the photograph at corner of the screen and the screen as shown in Figure 1.2 will appear.



Click "Browse" and then select a photograph from the relevant path. Click "Upload" once you have selected the picture .The picture selected will be populated on the photograph section.
*Note: You may only upload a maximum size of 1 Megabyte in jpg, png, gif format.

3.1.3 Contact Details

Contact information can be entered from here. Click on "Contact Details" under the Employee Details column and the screen as shown in Figure 1.3 will appear.

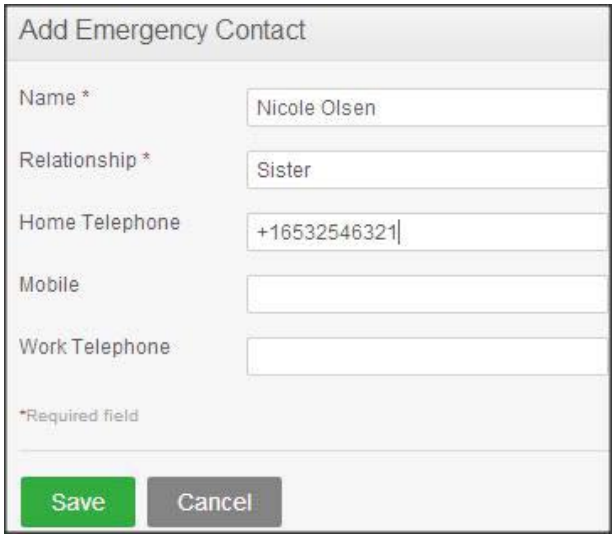


Click “Edit” to enter the information.
You can edit the following:
Country– Select the country from the drop down
Street 1
Street 2

City/Town
State/Province– If the country is United Sates you can select from the drop down or you need to enter it manually
ZIP Code
Home Telephone
Mobile
Work Telephone
Work Email
Other Email
Once you have completed this form click “Save”.

3.1.4 Emergency Contact

Contact details which will be needed during an emergency can be entered here. Select “Emergency Contacts” on the “Personal” column and the screen as shown in Figure 1.4 will appear.



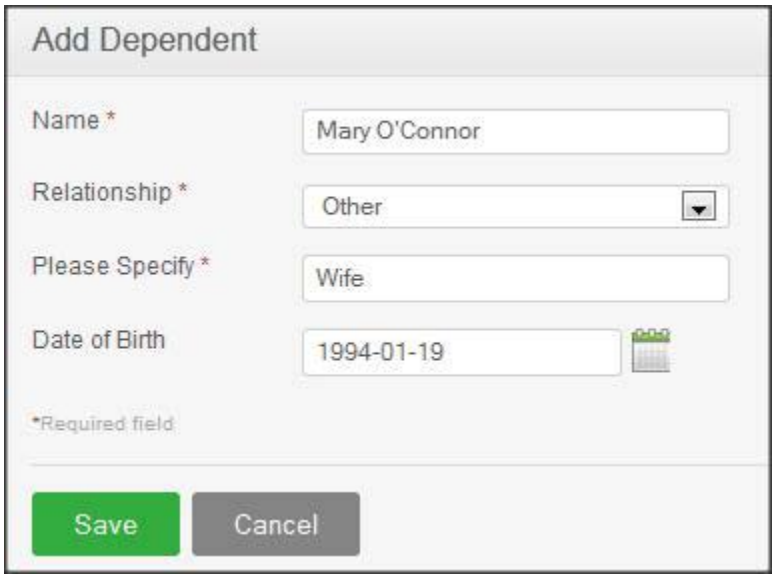
Enter the “Name” of the person you wish the company to contact in case of emergency, your “Relationship” with the contact person provided and a “Home Telephone” or “Mobile Number” the company can reach him/her.

Click “Save” once the fields are added, the emergency contact will be listed as shown in Figure 1.5.

You may add multiple entries of emergency contacts.
To delete an entry, click on the check box next to particular entry. It is also possible to delete multiple entries at the same time by clicking the check box entries you wish to delete and simply clicking “Delete”.
You may also upload any attachment that would support the details you have entered on the form by clicking “Add” under the “Attachment” and selecting a file from a relevant path and upload the following file by clicking “Upload”.

3.1.5 Dependants

If you have any dependents you can enter them here. To add a dependent, click on “Dependents” under the “Personal” column and the screen as shown in Figure 1.6 will appear.



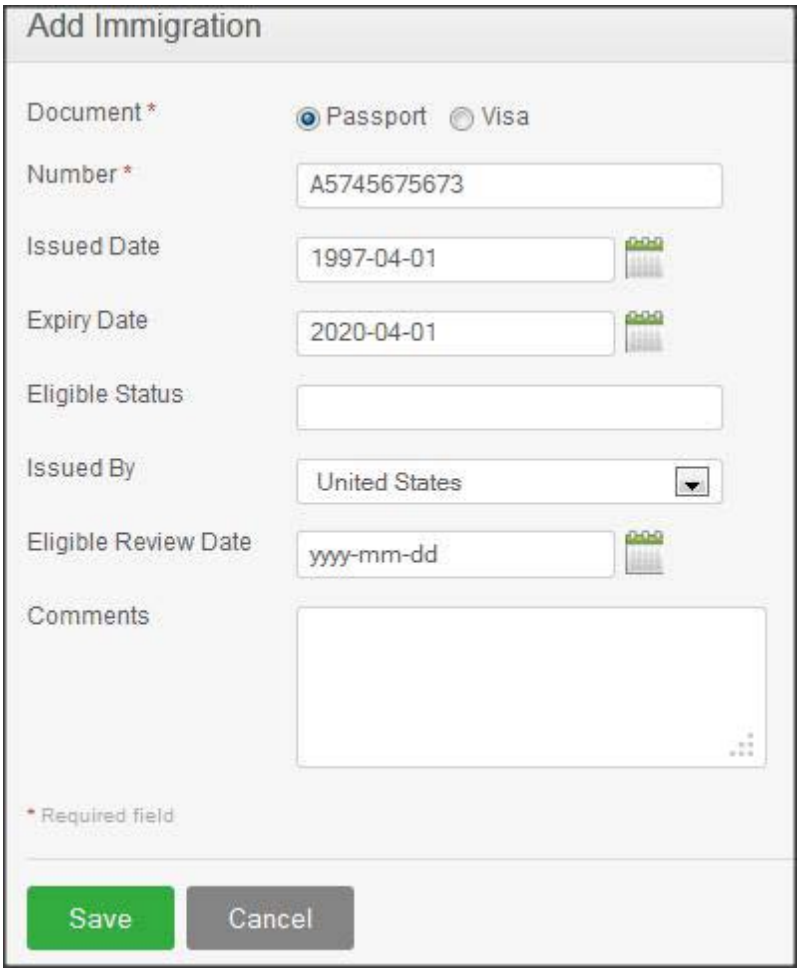
Enter the “Name” of your dependent, the “Relationship” of the dependent to you and his/her “Date of Birth”. Click “Save” once you have entered the following fields and your dependent will be listed as shown in Figure 1.7.

You may add multiple entries of dependants.
To delete an entry, click on the check box next to particular entry. It is also possible to delete multiple entries at the same time by clicking the check box entries you wish to delete and simply clicking “Delete”.

You may also upload any attachment that would support the details you have entered on the form by clicking “Add” under the “Attachment” and selecting a file from a relevant path and uploading the following file by clicking “Upload”.

3.1.6 Immigration

Your immigration information can be entered here. To add your immigration information, select “Immigration” under the “Personal “column and the screen as shown in Figure 1.8 will appear.



Select the document type (Passport or Visa) you wish to add details of, the “Number” whether it is a passport number or a visa number, the “ Issued Date” , “Expiry Date”, the “Eligible Status” of your Passport/Visa and the “Eligible Review Date” as to when the eligibility status was reviewed. You may write a comment if necessary.

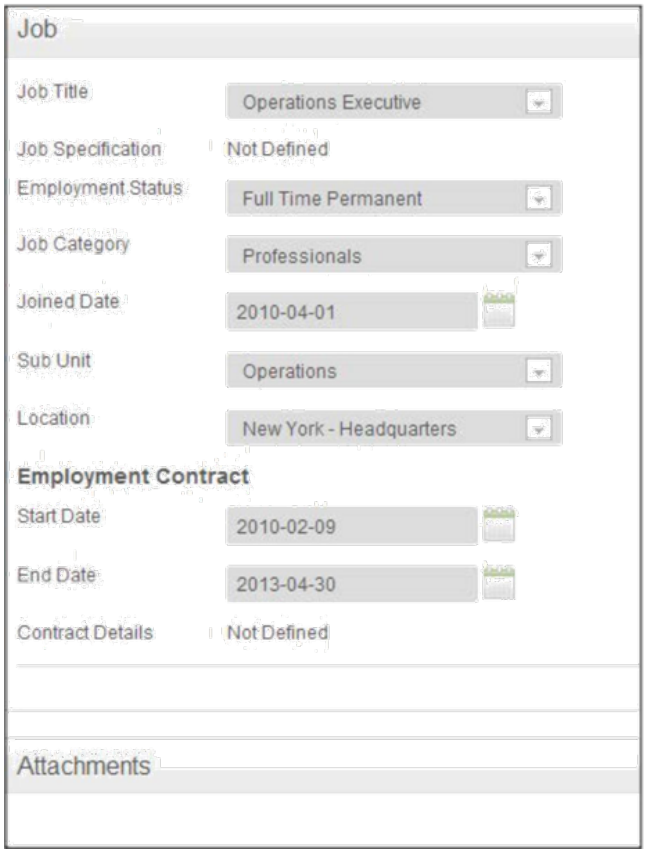
Click “Save” once the fields are added and the following immigration documents will be listed as shown in Figure 1.9.

You may add multiple entries of immigration documents.
To delete an entry, click on the check box next to particular entry. It is also possible to delete multiple entries at the same time by clicking the check box entries you wish to delete and simply clicking “Delete”.
You may also upload any attachment that would support the details you have entered on the form by clicking “Add” under the “Attachment” and selecting a file from a relevant path and uploading the following file by clicking “Upload”.

3.1.7 Job

The ESS-User cannot make changes in the job details. You are only able to view your job details that have been pre-defined by the administrator as shown in Figure 2.0. You are restricted from editing the following fields:

- Job Title
- Jobs Specification
- Employment Status
- Job Category
- Joined Date
- Sub Unit
- Location
- Employment Contract Start Date
- Employment Contract End Date
- Attachments



3.1.8 Salary

The salary information field is completely hidden from the ESS-User as shown in Figure 2.1. Only the HR Admin has access to this information and has to be manually communicated to the ESS-User. You are restricted from editing the following fields:

Salary

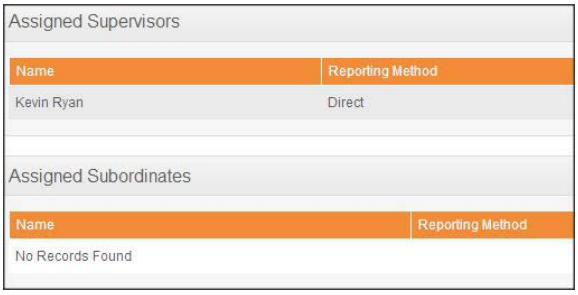
- Salary Component
- Pay Frequency
- Currency
- Amount
- Comments
- Direct Deposit Details
- Attachments

3.1.9 Report To

As an ESS-User, you are only able to view the list of supervisors that you report to and if you are an ESS-Supervisor as well, you will see the list of your subordinates as shown in Figure 2.2.

You are restricted from editing the following fields:

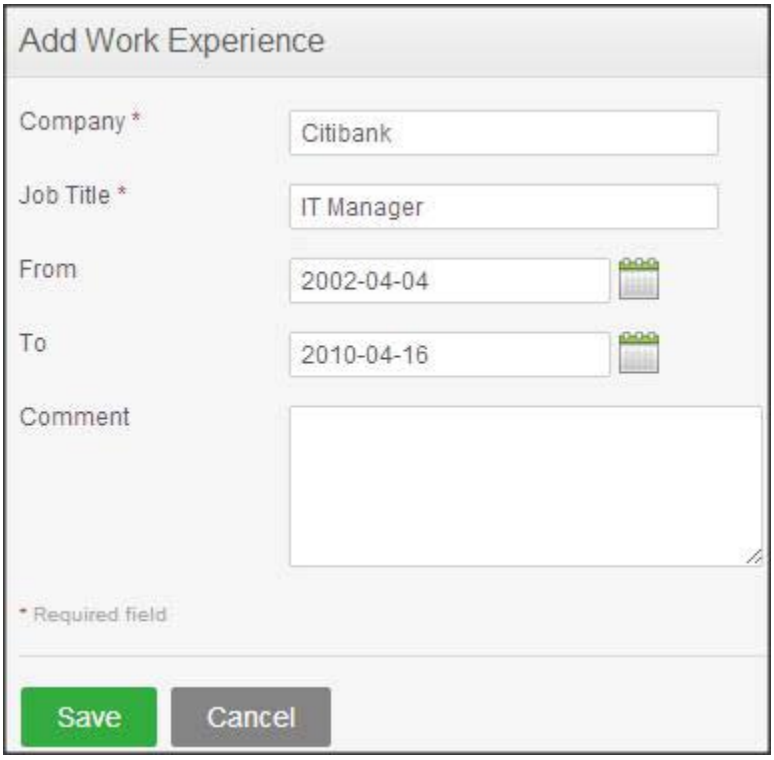
- Assigned Supervisors
- Assigned Subordinates
- Attachments



3.1.10 Qualifications

- Work Experience

Your previous work experiences can be entered here. To enter previous work experiences, click “Add” under “Work Experience” and the screen as shown in Figure 2.3 will appear.

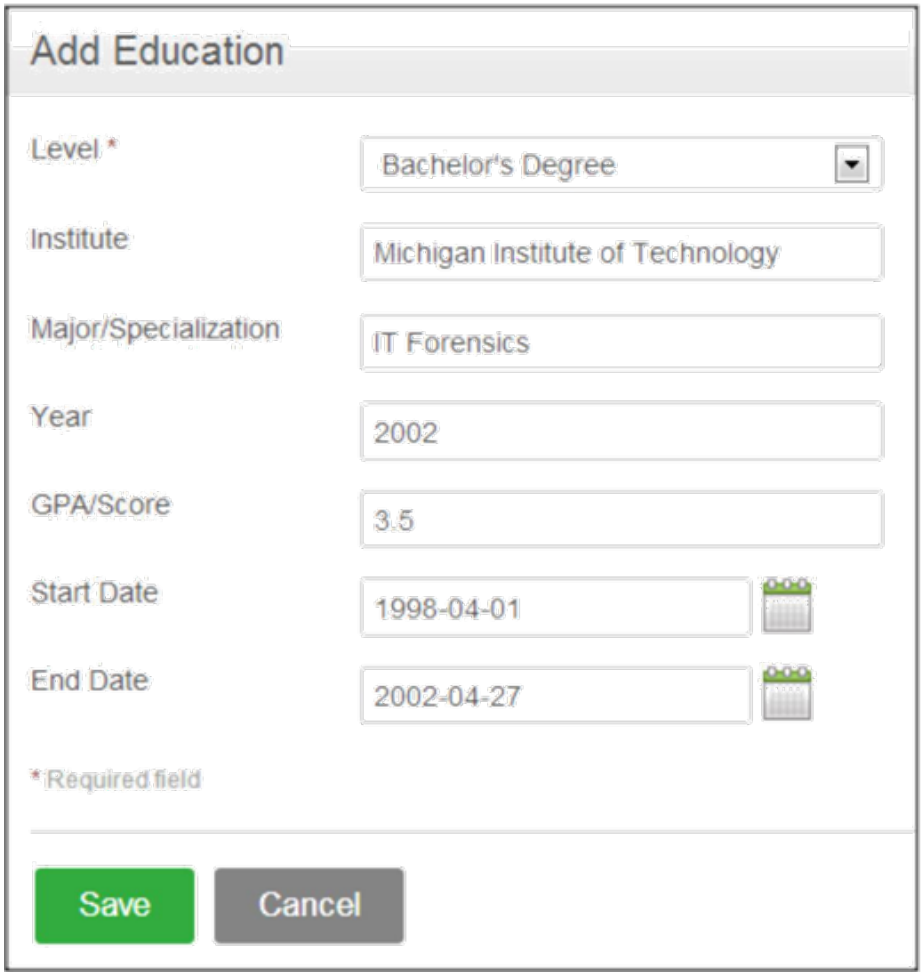


Click “Save” once all the fields are entered and the particular work experience will be listed as shown in Figure 2.4.

You may enter multiple entries of work experience.
To delete an entry, click on the check box next to a particular entry. It is also possible to delete multiple entries at the same time by clicking the check box entries you wish to delete and simply clicking “Delete”.

● Education

You are able to enter details of your education here. To enter education details, click “Add” under “Education” and the screen as shown in Figure 2.5 will appear.

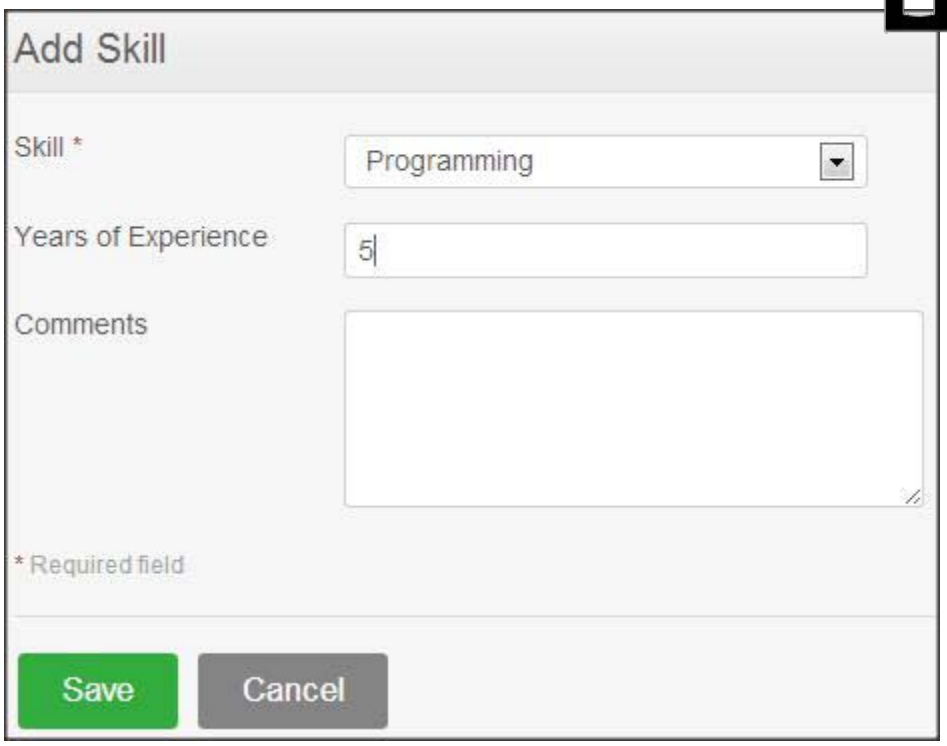


Click "Save" once all the fields are entered and the particular education details will be listed as shown in Figure 2.6.

You may enter multiple entries of education.
To delete an entry, click on the check box next to particular entry. It is also possible to delete multiple entries at the same time by clicking the check box entries you wish to delete and simply clicking "Delete".

● Skills

If you have any special talents or skills they can be entered here. To enter skills, click "Add" under "Skills" and the screen as shown in Figure 2.7 will appear.



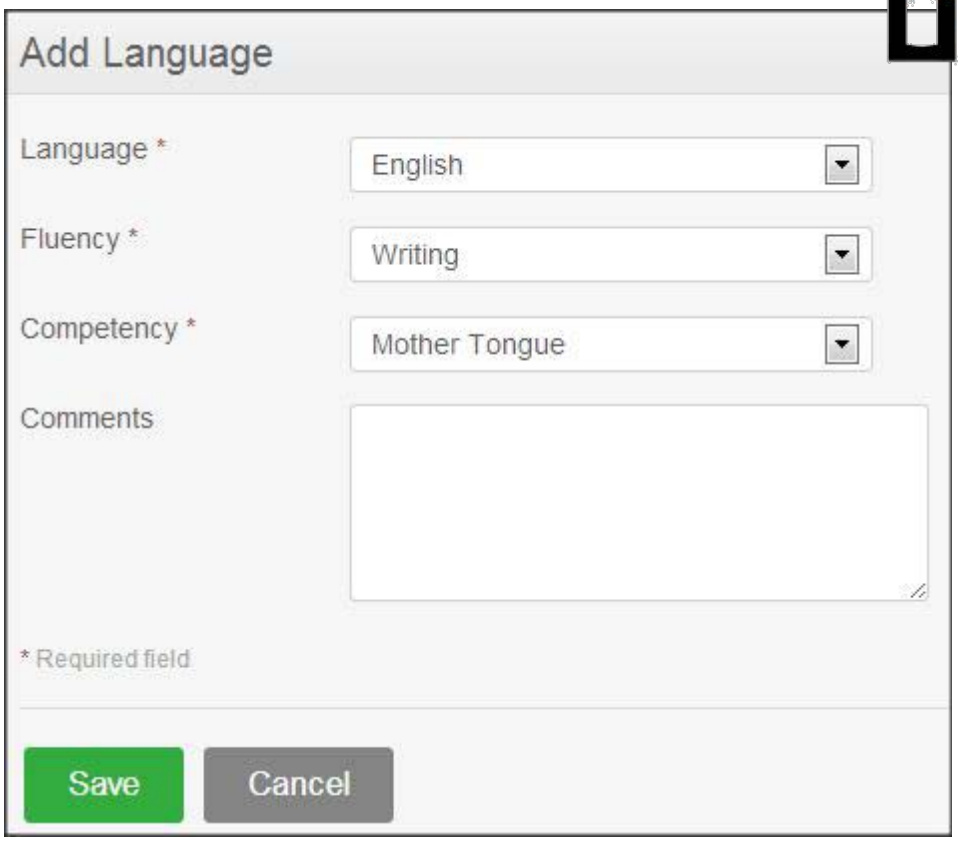
Click “Save” once all the fields are entered and the particular skill will be listed as shown in Figure 2.8.

You may enter multiple entries of skills.

To delete an entry, click on the check box next to particular entry. It is also possible to delete multiple entries at the same time by clicking the check box entries you wish to delete and simply clicking “Delete”.

● Languages

You can enter the various languages that you are competent in, with the level of competency. To enter your language of competency, click “Add” under “Language” and the screen as shown in Figure 2.9 will appear.

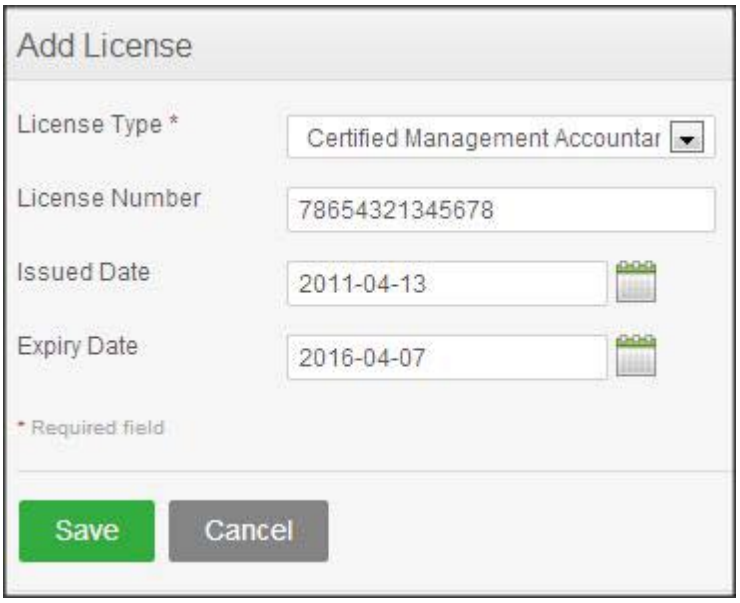


Click “Save” once all the fields are entered and the particular language of competency will be listed as shown in Figure 3.0.

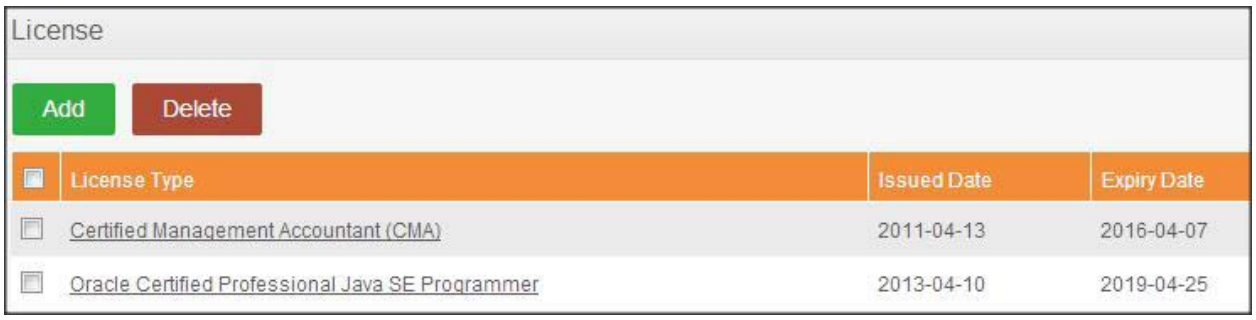
You may enter multiple entries of languages.
To delete an entry, click on the check box next to particular entry. It is also possible to delete multiple entries at the same time by clicking the check box entries you wish to delete and simply clicking “Delete”.

● License

Here you can enter the licenses that you may have. To enter licenses, click “Add” under “License” and the screen as shown in Figure 3.1 will appear.



Click "Save" once all the fields are entered and the particular license will be listed as shown in Figure 3.2

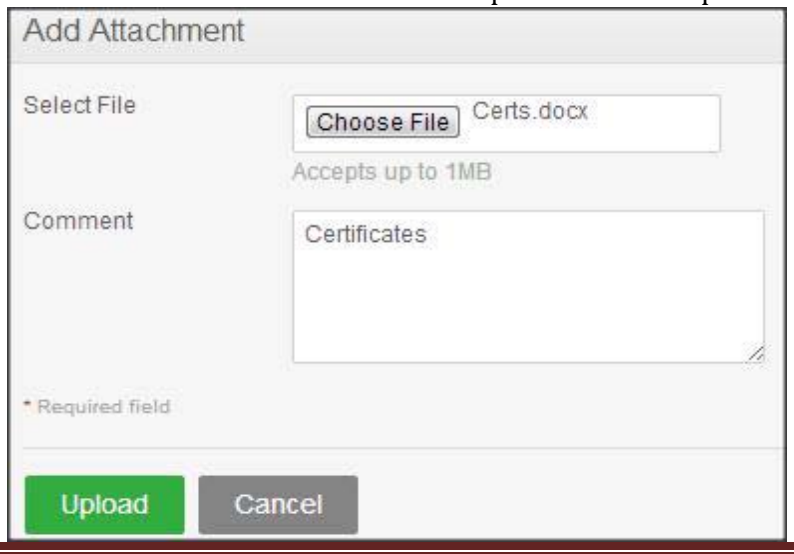


You may enter multiple entries of licenses.
To delete an entry, click on the check box next to particular entry. It is also possible to delete multiple entries at the same time by clicking the check box entries you wish to delete and simply clicking "Delete".

●Attachments

Any supporting documents regarding your qualification that you think is needed by the management can be attached here. Please note that each document cannot exceed 1 megabyte, but you can attach more than on document. To add an attachment, click "Add" under attachment and the screen as shown in Figure 3.3 will appear.

Click "Browse" and select the file from the relevant path and click "Upload" to upload it.

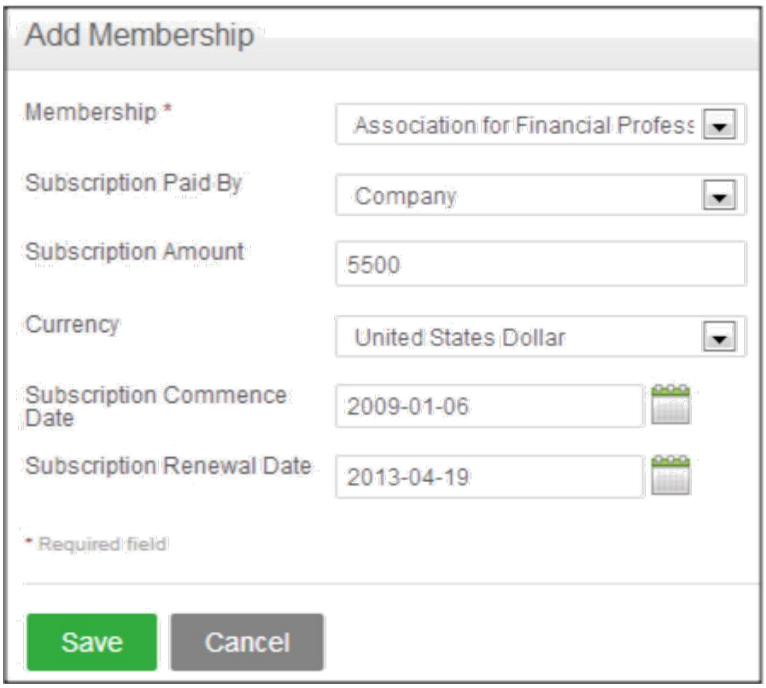


Once you have uploaded the file, the file will be listed as shown in Figure 3.4

You may upload multiple attachments.
To delete an entry click on the check box next to the particular entry and click "Delete".
Multiple selections can be deleted simultaneously.

3.1.11 Membership

If you are a members of any committee, institute etc. those details can be entered here. To enter membership details, go to My Info>>Personal>>Membership and click "Add" and the screen as shown in Figure 3.5 will appear.



Click “Save” once all the fields are entered and the particular membership detail will be listed as shown in Figure 3.6.

You may enter multiple entries of memberships.
To delete an entry, click on the check box next to particular entry. It is also possible to delete multiple entries at the same time by clicking the check box entries you wish to delete and simply clicking “Delete”.
You may also upload any attachment that would support the details you have entered on the form by clicking “Add” under the “Attachment” and selecting a file from a relevant path and upload the following file by clicking “Upload”.

4. Site Design

4.1 Aesthetic/HTML Requirements and Guidelines

OrageHRM must deliver a compelling visitor experience. However, it cannot sacrifice usability and accessibility.

The web site ‘look’ must conform to the following requirements:

- The site should be HTML 4.0 compliant.
- All pages must download in less than 10 seconds over a 56k modem connection.- Perfomrance req
- All pages must fit in a web browser displayed on a computer set to 640 x 480 pixels.
- All pages must use a web safe color palette.
- The site must be compatible with Internet Explorer 4, 5 and 5.5, and with Firefox 4-6, as well as Google Chrome 4.0 and later.
- The site must conform to the WAI Accessibility Guidelines outlined at http://www.w3.org/TR/WAI-WEBCONTENT/ , wherever possible.
- All site pages should be available for search engine robots.
- All pages that use static images should be displayed correctly.

5. Sign-Off Document

The following parties have read and agree with this Requirements Definition document for the OrangeHRM application account module functionality.
After approval of this Requirements Definition phase, any significant changes in the scope of this project will require validation of existing project costs and schedules.

_____ _____

Name Date
Business Lead

_____ _____

Name
Project Manager

Date

TEST PLAN

Test Plan (a Real Sample)

Live Project Training - OrangeHRM

Note: This is a sample test plan created on real time software testing live project – for training conducted by www.Payilagam.com on following page:

Version: 1.0
Created: 02/05/2014
Last Updated: 02/05/2014
Status: DRAFT (The status would change to finalized post the BA, PM and dev team review and sign off)

Revision and Signoff Sheet

Document History - To maintain a list of changes being made

Version	Date	Author	Description of Change
1	02/14/2014	Kamal Subramani	Draft
2	02/14/2014	Siva Subramani	Draft - Reviewed

Approvers List - To track who has reviewed and signoff on the Test plan

Name	Role	Approver / Reviewer	Approval / Review Date

Reference Documents - Clearly mark the document used as an input to create the test plan

Version	Date	Document Name
1.0		ORANGEHRM VERSION 3.0 – MY INFO MODULE - FSD

FUNCTIONAL TESTING SCOPE

User	Scenarios	Sub Levels	Complexity	No. of Test cases	Negative Test Cases	Expecting Additional Test Cases
Employee	Login Page	Login	Medium	1	1	
Employee	My information	Add info	Complex	10	1	1
		Delete info	Complex	10	1	
		Verify the info display	Medium	5		
Employee	Photograph	Add photograph	Complex	5	1	1
		Replace	Complex	5	1	
		Verify photograph	Medium	3		
Employee	Aesthetics	HTML requirements	Medium	1		
		Performance	Complex	3		
		Accessibility	Complex	10		
		Image display	Medium	10		

DFRT EXECUTION CYCLE

Functional	Start Date	End Date	Available time
Test Execution	Monday, March 10, 2014	Tuesday, March 25, 2014	(12 working days)
First navigation, Employee module(Cycle 1)	Monday, March 10, 2014	Friday, March 14, 2014	
Development team 1 Day	Monday, March 17, 2014	Monday, March 17, 2014	
Retest	Tuesday, March 18, 2014	Tuesday, March 18, 2014	
First navigation, Employee (Cycle 2)	Wednesday, March 19, 2014	Friday, March 21, 2014	
Development team 1 Day	Monday, March 24, 2014	Monday, March 24, 2014	
Retest	Tuesday, March 25, 2014	Tuesday, March 25, 2014	

1. INTRODUCTION

1.1. Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the

OrangeHRM Version 3.0 – My Info Module.com site. The document introduces:

- Test Strategy: rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).
- Execution Strategy: describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.
- Test Management: process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk and mitigation, team roster)

1.2. Project Overview

My Info Module is a powerful tool providing employees of the company with the ability to view relevant information such as personal information and updating personal information with an internet enabled PC without having to involve the HR department.

The functionality of this module spans through the entire system, making information available anywhere, anytime. All information is subject to company's defined security policy, where he/she can only view the information he/she is authorized to. An ESS-User can only edit certain fields in the ESS Module, maintaining the security and confidentiality of employee information

1.3. Audience

- Project team members perform tasks specified in this document, and provide input and recommendations on this document.
- Project Manager Plans for the testing activities in the overall project schedule, reviews the document, tracks the performance of the test according to the task herein specified, approves the document and is accountable for the results.
- The stakeholders' representatives and participants (individuals as identified by the PMO Leads) may take part in the UAT test to ensure the business is aligned with the results of the test.
- Technical Team ensures that the test plan and deliverables are in line with the design, provides the environment for testing and follows the procedures related to the fixes of defects.
- Business analysts will provide their inputs on functional changes.

2. TEST STRATEGY

2.1. Test Objectives

The objective of the test is to verify that the functionality of ORANGEHRM VERSION 3.0 – MY INFO MODULE works according to the specifications.

The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing via CR.

The final product of the test is twofold:

- A production-ready software;
- A set of stable test scripts that can be reused for Functional and UAT test execution.

2.2. Test Assumptions

Key Assumptions

- Production like data required and be available in the system prior to start of Functional Testing
- In each testing phase, Cycle 3 will be initiated if the defect rate is high in Cycle 2.

General

- Exploratory Testing would be carried out once the build is ready for testing
- Performance testing is not considered for this estimation.
- All the defects would come along with a snapshot JPEG format
- The Test Team will be provided with access to Test environment via VPN connectivity
- The Test Team assumes all necessary inputs required during Test design and execution will be supported by Development/BUSINESS ANALYSTs appropriately.
- Test case design activities will be performed by QA Group
- Test environment and preparation activities will be owned by Dev Team
- Dev team will provide Defect fix plans based on the Defect meetings during each cycle to plan. The same will be informed to Test team prior to start of Defect fix cycles
- BUSINESS ANALYST will review and sign-off all Test cases prepared by Test Team prior to start of Test execution
- The defects will be tracked through HP ALM only. Any defect fixes planned will be shared with Test Team prior to applying the fixes on the Test environment
- Project Manager/BUSINESS ANALYST will review and sign-off all test deliverables
- The project will provide test planning, test design and test execution support
- Test team will manage the testing effort with close coordination with Project PM/BUSINESS ANALYST
- Project team has the knowledge and experience necessary, or has received adequate training in the system, the project and the testing processes.
- There is no environment downtime during test due to outages or defect fixes.
- The system will be treated as a black box; if the information shows correctly online and in the reports, it will be assumed that the database is working properly.
- Cycle 3 will be initiated if there are more defects in Cycle 2.

Functional Testing

- During Functional testing, testing team will use preloaded data which is available on the system at the time of execution
- The Test Team will be perform Functional testing only on ORANGEHRM VERSION 3.0 – MY INFO MODULE

UAT

- UAT test execution will be performed by end users (L1, L2 and L3) and QA Group will provide their support on creating UAT script.

2.3. Test Principles

- Testing will be focused on meeting the business objectives, cost efficiency, and quality.
- There will be common, consistent procedures for all teams supporting testing activities.
- Testing processes will be well defined, yet flexible, with the ability to change as needed.
- Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
- Testing environment and data will emulate a production environment as much as possible.
- Testing will be a repeatable, quantifiable, and measurable activity.
- Testing will be divided into distinct phases, each with clearly defined objectives and goals.
- There will be entrance and exit criteria.

2.4. Data Approach

- In functional testing, ORANGEHRM VERSION 3.0 – MY INFO MODULE will contain pre-loaded test data and which is used for testing activities.

2.5. Scope and Levels of Testing

2.5.1. Exploratory

PURPOSE: the purpose of this test is to make sure critical defects are removed before the next levels of testing can start.

SCOPE: First level navigation, dealer and admin modules

TESTERS: Testing team.

METHOD: this exploratory testing is carried out in the application without any test scripts and documentation

TIMING: at the beginning of each cycle.

2.5.2. Functional Test

PURPOSE: Functional testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

Scope: The below excel sheet details about the scope of Functional test. Note: The scope is high level due to changes in the requirement.

To keep the document easily fragmented and categorized, the scope has been embedded as separate document. If you prefer you can insert a table here itself. The scope is created based on the Test scenarios that were identified in the previous article.

Functional Testing Scope.xlsx

TESTERS: Testing Team.

METHOD: The test will be performed according to Functional scripts, which are stored in HP ALM.

TIMING: after Exploratory test is completed.

TEST ACCEPTANCE CRITERIA

1. Approved Functional Specification document, Use case documents must be available prior to start of Test design phase.
2. Test cases approved and signed-off prior to start of Test execution
3. Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects
4. Test environment with application installed, configured and ready to use state

Sign-off	Readiness
•Approved Functional Specification Document •Approved Use cases •Approved Test cases	•Development completed & unit tested •Application deployed and system ready for testing on Test environment •Production like data is available to test all functionalities. •Defect fixes planned based on Defect triage (Unit Testing) and evaluation criteria

TEST DELIVERABLES

S.No.	Deliverable Name	Author	Reviewer
1.	Test Plan	Test Lead	Project Manager/ Business Analyst's
2.	Functional Test Cases	Test Team	Business Analyst's Sign off
3.	Logging Defects in HP ALM	Test Team	Test Lead/ Programming Lead(Vijay)
(4.	Daily/weekly status report	Test Team/ Test Lead	Test Lead/ Project Manager
5.	Test Closure report	Test Lead	Project Manager

MILESTONE LIST

The milestone list is tentative and may change due to below reasons

a) Any issues in the System environment readiness
b) Any change in scope/addition in scope
c) Any other dependency that impacts efforts and timelines

Testing generally is not carried out in one cycle. Based on the testing scope, we can estimate how much time it takes and establish the time lines as you can see in the below embedded excel sheet.

DFRT Execution Cycle.xlsx

2.5.3. User Acceptance Test (UAT)

PURPOSE: this test focuses on validating the business logic. It allows the end users to complete one final review of the system prior to deployment.

TESTERS: the UAT is performed by the end users (L1, L2 and L3).

METHOD: Since the business users are the most indicated to provide input around business needs and how the system adapts to them, it may happen that the users do some validation not contained in the scripts. Test team write the UAT test cases based on the inputs from End user (L1,L2 and L3 users) and Business Analyst's.

TIMING: After all other levels of testing (Exploratory and Functional) are done. Only after this test is completed the product can be released to production.

TEST DELIVERABLES

S.No.	Deliverable Name	Author	Reviewer
1.	UAT Test Cases	Test Team	Business Analyst's Sign off

2.6. Test Effort Estimate

This document lists out all the activities that have to be performed by the QA team and estimates how many man-hours each activity is going to take.

New_Detailed DRFT

3. Test estimate v1.xlsx

Note: this estimate is for the TCOE team only Testing Schedule

4. EXECUTION STRATEGY

4.1. Entry and Exit Criteria

- The entry criteria refer to the desirable conditions in order to start test execution; only the migration of the code and fixes need to be assessed at the end of each cycle.

- The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.
- Entry and exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation. All this is input to the project manager for a final "go-no go" decision.
- Entry criteria to start the execution phase of the test: the activities listed in the Test Planning section of the schedule are 100% completed.
- Entry criteria to start each cycle: the activities listed in the Test Execution section of the schedule are 100% completed at each cycle.

Exit Criteria	Test Team	Technical Team	Notes
100% Test Scripts executed			
95% pass rate of Test Scripts			
No open Critical and High severity defects			
95% of Medium severity defects have been closed			
All remaining defects are either cancelled or documented as Change Requests for a future release			
All expected and actual results are captured and documented with the test script			
All test metrics collected based on reports from HP ALM			
All defects logged in HP ALM			
Test Closure Memo completed and signed off			
Test environment cleanup completed and a new back up of the environment			

4.2. Test Cycles

- There will be two cycles for functional testing. Each cycle will execute all the scripts .
- The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects. It is expected to use some work-around in order to get to all the scripts.
- The objective of the second cycle is to identify remaining high and medium defects, remove the work-around from the first cycle, correct gaps in the scripts and obtain performance results.

- UAT test will consist of one cycle.

4.3. Validation and Defect Management

- It is expected that the testers execute all the scripts in each of the cycles described above. However it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. This is especially relevant in the second cycle, when the Business analyst's join the TCOE in

the execution of the test, since the BUSINESS ANALYSTs have a deeper knowledge of the business processes. If a gap is identified, the scripts and traceability matrix will be updated and then a defect logged against the scripts.

- The defects will be tracked through HP ALM only. The technical team will gather information on a daily basis from HP ALM, and request additional details from the Defect Coordinator. The technical team will work on fixes.
- It is the responsibility of the tester to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the Defect Manager to review the severity of the defects and facilitate with the technical team the fix and its implementation, communicate with testers when the test can continue or should be halt, request the tester to retest, and modify status as the defect progresses through the cycle; it is the responsibility of the technical team to review HP ALM on a daily basis, ask for details if necessary, fix the defect, communicate to the Defect Manager the fix is done, implement the solution per the Defect Manager request.

Defects found during the Testing will be categorized according to the bug-reporting tool "Mercury HP ALM" and the categories are:

Severity	Impact
1 (Critical)	☐ This bug is critical enough to crash the system, cause file corruption, or cause potential data loss ☐ It causes an abnormal return to the operating system (crash or a system failure message appears). ☐ It causes the application to hang and requires re-booting the system.
2 (High)	☐ It causes a lack of vital program functionality with workaround.
3 (Medium)	☐ This Bug will degrade the quality of the System. However there is an intelligent workaround for achieving the desired functionality - for example through another screen. ☐ This bug prevents other areas of the product from being tested. However other areas can be independently tested.
4 (Low)	☐ There is an insufficient or unclear error message, which has minimum impact on product use.
5(Cosmetic)	☐ There is an insufficient or unclear error message that has no impact on product use.

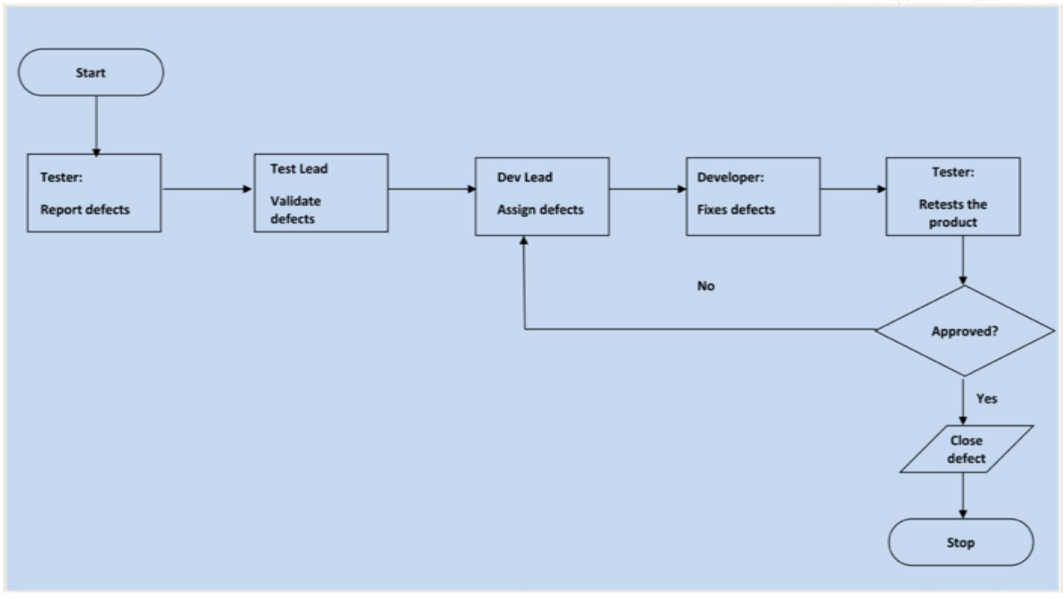
4.4. Test Metrics

Test metrics to measure the progress and level of success of the test will be developed and shared with the project manager for approval. The below are some of the metrics

Report	Description	Frequency
Test preparation & Execution Status	To report on % complete, %WIP, % Pass, % Fail Defects severity wise Status – Open, closed, any other Status	Weekly / Daily (optional)
Daily execution status	To report on Pass, Fail, Total defects, highlight Showstopper/ Critical defects	Daily
Project Weekly Status report	Project driven reporting (As requested by PM)	Weekly – If project team needs weekly update apart from daily and there is template available with project team to use.

4.5. Defect tracking & Reporting

Following flowchart depicts Defect Tracking Process:



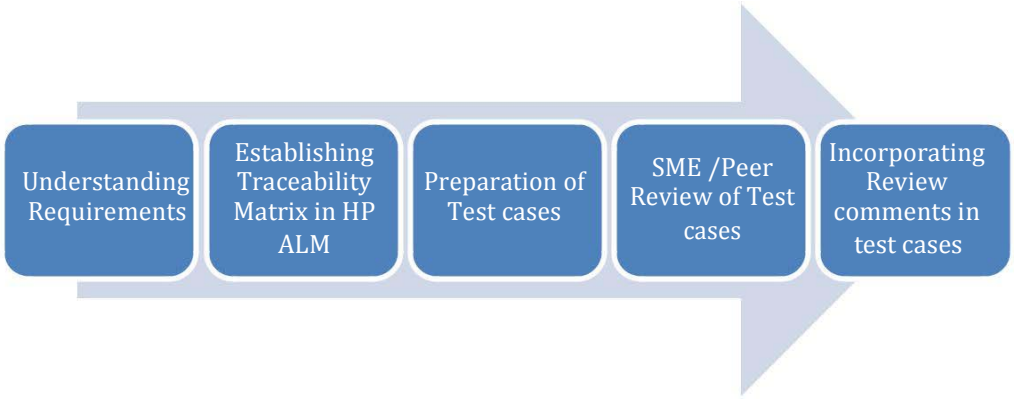
5. TEST MANAGEMENT PROCESS

5.1. Test Management Tool

HP Application Lifecycle Management is the tool used for Test Management. All testing artifacts such as Test cases, test results are updated in the HP Application Lifecycle Management (ALM) tool.

- Project specific folder structure will be created in HP ALM to manage the status of this DFRT project.
- Each resource in the Testing team will be provided with Read/Write access to add/modify Test cases in HP ALM.
- During the Test Design phase, all test cases are written directly into HP ALM. Any change to the test case will be directly updated in the HP ALM.
- Each Tester will directly access their respective assigned test cases and update the status of each executed step in HP ALM directly.
- Any defect encountered will be raised in HP ALM linking to the particular Test case/test step.
- During Defect fix testing, defects are re-assigned back to the tester to verify the defect fix. The tester verifies the defect fix and updates the status directly in HP ALM.
- Various reports can be generated from HP ALM to provide status of Test execution. For example, Status report of Test cases executed, Passed, Failed, No. of open defects, Severity wise defects etc.

5.2. Test Design Proces



- The tester will understand each requirement and prepare corresponding test case to ensure all requirements are covered.
- Each Test case will be mapped to Use cases to Requirements as part of Traceability matrix.
- Each of the Test cases will undergo review by the BUSINESS ANALYST and the review defects are captured and shared to the Test team. The testers will rework on the review defects and finally obtain approval and sign-off.
- During the preparation phase, tester will use the prototype, use case and functional specification to write step by step test cases.
- Testers will maintain a clarification Tracker sheet and same will be shared periodically with the Requirements team and accordingly the test case will be updated. The clarifications may sometimes lead to Change Requests or not in scope or detailing implicit requirements.
- Sign-off for the test cases would be communicates through mail by Business Analyst's.
- Any subsequent changes to the test case if any will be directly updated in HP ALM.

5.3. Test Execution Process

- Once all Test cases are approved and the test environment is ready for testing, tester will start a exploratory test of the application to ensure the application is stable for testing.
- Each Tester is assigned Test cases directly in HP ALM.
- Testers to ensure necessary access to the testing environment, HP ALM for updating test status and raise defects. If any issues, will be escalated to the Test Lead and in turn to the Project Manager as escalation.
- If any showstopper during exploratory testing will be escalated to the respective development SPOCs for fixes.

- Each tester performs step by step execution and updates the executions status. The tester enters Pass or Fail Status for each of the step directly in HP ALM.
- Tester will prepare a Run chart with day-wise execution details
- If any failures, defect will be raised as per severity guidelines in HP ALM tool detailing steps to simulate along with screenshots if appropriate.
- Daily Test execution status as well as Defect status will be reported to all stakeholders.
- Testing team will participate in defect triage meetings in order to ensure all test cases are executed with either pass/fail category.
- If there are any defects that are not part of steps but could be outside the test steps, such defects need to be captured in HP ALM and map it against the test case level or at the specific step that issue was encountered after confirming with Test Lead.
- This process is repeated until all test cases are executed fully with Pass/Fail status.
- During the subsequent cycle, any defects fixed applied will be tested and results will be updated in HP ALM during the cycle.

As per Process, final sign-off or project completion process will be followed

5.4. Test Risks and Mitigation Factors

Risk	Prob.	Impact	Mitigation Plan
SCHEDULE Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date.	High	High	• The testing team can control the preparation tasks (in advance) and the early communication with involved parties. • Some buffer has been added to the schedule for contingencies, although not as much as best practices advise.
RESOURCES Not enough resources, resources on boarding too late (process takes around 15 days.	Medium	High	Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing.
DEFECTS Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve.	Medium	High	Defect management plan is in place to ensure prompt communication and fixing of issues.
SCOPE Scope completely defined	Medium	Medium	Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing.
Natural disasters	Low	Medium	Teams and responsibilities have

Risk	Prob.	Impact	Mitigation Plan
			been spread to two different geographic areas. In a catastrophic event in one of the areas, there will resources in the other areas needed to continue (although at a slower pace) the testing activities.
Non-availability of Independent Test environment and accessibility	Medium	High	Due to non availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution.
Delayed Testing Due To new Issues	Medium	High	During testing, there is a good chance that some "new" defects may be identified and may become an issue that will take time to resolve. There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved. If these issues become showstoppers, it will greatly impact on the overall project schedule. If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution.

5.1. Communications Plan and Team Roster

5.2. Role Expectations

The following list defines in general terms the expectations related to the roles directly involved in the management, planning or execution of the test for the project.

	Roles	Name	Contact Info
1.	Project Manager		
2.	Test Lead		
3.	Business Analyst		
4.	Development Lead		

	Roles	Name	Contact Info
5.	Testing Team		
6.	Development Team		
7.	Technical Lead		

5.2.1. Project Management

- Project Manager: reviews the content of the Test Plan, Test Strategy and Test Estimates signs off on it.

5.2.2. Test Planning (Test Lead)

- Ensure entrance criteria are used as input before start the execution.
- Develop test plan and the guidelines to create test conditions, test cases, expected results and execution scripts.
- Provide guidelines on how to manage defects.
- Attend status meetings in person or via the conference call line.
- Communicate to the test team any changes that need to be made to the test deliverables or application and when they will be completed.
- Provide on premise or telecommute support.
- Provide functional (Business Analysts) and technical team to test team personnel (if needed).

5.2.3. Test Team

- Develop test conditions, test cases, expected results, and execution scripts.
- Perform execution and validation.
- Identify, document and prioritize defects according to the guidance provided by the Test lead.
- Re-test after software modifications have been made according to the schedule.
- Prepare testing metrics and provide regular status.

5.2.4. Test Lead

- Acknowledge the completion of a section within a cycle.
- Give the OK to start next level of testing.
- Facilitate defect communications between testing team and technical / development team.

5.2.5. Development Team

- Review testing deliverables (test plan, cases, scripts, expected results, etc.) and provide timely feedback.
- Assist in the validation of results (if requested).
    - Support the development and testing processes being used to support the project.
    - Certify correct components have been delivered to the test environment at the points specified in the testing schedule.
    - Keep project team and leadership informed of potential software delivery date slips based on the current schedule.
    - Define processes/tools to facilitate the initial and ongoing migration of components.
    - Conduct first line investigation into execution discrepancies and assist test executors in creation of accurate defects.
    - Implement fixes to defects according to schedule.

6. TEST ENVIRONMENT

ORANGEHRM VERSION 3.0 – MY INFO MODULE's servers will be hosted at X company's site. RANGEHRM VERSION 3.0 – MY INFO MODULE will be hosted on two servers: One to host the actual website and (language) code, and the other to host the (database name) database.

A windows environment with Internet Explorer 8, 9 and 10, and with Firefox 27.0, as well as Google Chrome 32.0 and later should be available to each tester.

7. APPROVALS

The Names and Titles of all persons who must approve this plan.

Signature:	
Name:	
Role:	
Date:	

Signature:	
Name:	
Role:	
Date:	

TEST SCENARIO

By

Kamal Subramani

Payilagam

Project Name	**OrangeHRM Version 3.0 – My Info Module**
Reference Document	**Project Functional Requirement Specification ,**
Created by	**Version 1**
Date of creation	Prepared By: Kamal Subramani
Date of review	**04-Feb-14**
	10-Feb-14

Test scenario ID	Requirement- reference document index	Test scenario description	Importance	No. of test cases
TS_MI_01	Section 3, Page 3	validate if the user is able to enter the Orange HRM system with a successful ESS-User account	High	3
TS_MI_MIM_01	3.1.1	validate if the user is able to see the "Personal Details" on logging in the first time	Medium	
		validate if the user is able to edit the fields other than the following and save changes Personal Details ● Employee ID ● SSN No ● SIN No ● Driver License No ● Date of Birth	High	20
TS_MI_P_01	3.1.2	Validate if the user can upload a picture from the site of the format (jpg, png, gif)	Medium	3
TS_MI_P_02	3.1.2	Validate if the user can upload a picture of the size less than 1 MB	Medium	2
TS_MI_P_03	3.1.2	Validate if the user can replace a picture when a new picture is uploaded	Low	2
TS_MI_Aesthetics_01	4.1	validate if the site is HTML 4.0 compliant (We do not know how to test this, but we still need to make a note of this. Sometimes, we might need the dev team's help to run this kind of test cases)	Medium	We do not know how many test cases this will result it, so leaving it empty
TS_MI_Aesthetics_02	4.1	validate if all pages download in less than 10 seconds over a 56k modem connection(Performance requirement -but to be noted just as well)	Medium	
TS_MI_Aesthetics_03	4.1	Validate if all pages fit in a web browser displayed on a computer set to 640 x 480 pixels	Medium	
TS_MI_Aesthetics_04	4.1	Validate if all pages use a web safe color palette	Medium	
TS_MI_Aesthetics_05	4.1	validate if the site is compatible with Internet Explorer 4, 5 and 5.5, and with Firefox 4-6, as well as Google Chrome 4.0 and later(gives an indication that when we set up the test env we need all the versions of the browsers installed on test machines)	Medium	
TS_MI_Aesthetics_06	4.1	validate if the site confirms to the WAI Accessibility Guidelines outlined at http://www.w3.org/TR/WAI- WEBCONTENT/ (Accessibility Testing requirement)	Medium	
TS_MI_Aesthetics_07	4.1	validate if all pages of the site are available for search engine robots	Medium	
TS_MI_Aesthetics_08	4.1	validate if the static images on the site are displayed correctly	Medium	

TEST CASES

By

Kamal Subramani

Payilagam

Quality Assurance Test Cases								
Company Confidential Copyright @ 2006		MindQ Systems						
Project name		**E-Banking**						
Module		**Admin**						
Component		**Admin**						
Document References		**E-Banking_Admin_FRS_V1.2, E-Banking_admin_test scenarios_v1.0**						
Author		**Suresh Babu**						
Test Case ID/Name	**Test Case Description**	**Step Name**	**Step Description**	**Input Fields**	**Expected Result**	**Actual Results**	**TestCaseResults**	**TC Priority**
	RF-Home Page							
TC001_Ebanking_RF Home Page_Visitor Links	Verify all the visitor links functionality in RF Home page	Step1	Enter URL and click on go	URL	System displays Home page of Ranford Bank			
		Step2	Click on Home		System displays Home page of Ranford Bank			
		Step3	Click on Personal Banking		System displays information about Personal Banking			
		Step4	Click on Corporate Banking		System displays information about Corporate Banking			
		Step5	Click on International Banking		System displays information about International Banking			
		Step6	Click on About Us		System displays information about Ranford Bank			
		Step7	Click on Customer Service		System displays Customer service information of Ranford Bank			
		Step8	Click on Internet Banking FAQ's		System displays some FAQ's about Internet Banking.			
		Step9	Click on Terms & Conditions		System displays terms & conditions of Internet banking			
		Step10	Click on Privacy		System displays privacy commitment of Ranford Bank			
		Step11	Click on Disclaimer		System displays notice and copyright of this site.			
		Step12	Click on Site Map		System displays site map of Ranford Bank			
TC002_Ebanking_RF Home Page_Admin Login_valid	Verify Admin Login with valid username & Password.	Step1	Enter URL & Click on go	URL	System displays Ranford Home Page with login facility			
		Step2	Enter username,password &Click on Login	Username & Password	System displays Admin Home Page			
TC003_Ebanking_RF Home Page_Admin Login_invalid	Verify Admin Login with invalid username & password	Step1	Enter URL & Click on go	URL	System displays Ranford Home Page with login facility			
		Step2	Enter invalid username or invalid password & click on login	Username & Password	System displays correspoding error message.			
TC004_Ebanking_RF Home Page_Banker Login_valid	Verify Banker Login valid Username & Password	Step1	Enter URL & click on go	URL	System displays home page with login facility			
		Step2	Select respective branch		System display the selected branch			

		Step3	Enter valid username,Password & click on Login	Username & Password	System displays Banker module			
TC005_Ebanking_RF Home Page_Banker Login_invalid	Verify Banker Login with invalid Username or invalid password.	Step1	Enter URL & click on go	URL	System displays home page with login facility			
		Step2	Select respective branch		System display the selected branch			
		Step3	Enter invalid username or invalid password & click on login	Username & Password	System displays corresponding error message.			
TC006_Ebanking_RF Home Page_Customer Links	Verify all CustomerLogin Links	Step1	Enter URL & click on go	URL	System displays home page with login facility			
		Step2	Click on Personal		System displays login page for Personal baking			
		Step3	Click on Corporate		System displays login page for Corporate banking			
		Step4	Click on International		System displays login page for International banking			
	Admin Links							
TC007_Ebanking_Admin Home Page_Admin Links	This testcase is validates all admin home page links	Step1	Enter Valid URL & click on go	URL	System displays home page with login facility			
		Step2	Enter username, Password & Click on Login	Username & Password	System displays Admin module			
		Step3	Click on Branches		System displays Branches details Page			
		Step4	Click on Roles		System displays Roles details Page.			
		Step5	Click on Users		System displays Users details Page.			
		Step6	Click on Employees		System displays Employees Details Page.			
		Step7	Click on Home		System displays Home page of Admin module.			
		Step8	Click on Change Password		System displays Change Password Page.			
		Step9	Click on Logout		Logout from the system and displays Ranford Home Page.			
	Branches							
TC008_Ebanking_Admin_Branches_New Branchcreation_valid	Verify creation of a new branch with valid data.	Step1	Enter URL & click on go	URL	System displays home page with login facility			
		Step2	Enter valid Username,Password & click on login	Username & Password	System displays Admin module			
		Step3	Click Branches		System displays branches details page			
		Step4	Click New Branch		System displays New branch Entry Page.			
		Step5	Enters necessary fields with valid data and click on Reset	Branch Name Address1 Address2 Address3 Area Zip code	System clears all fields			

		Step6	Enter necessary fields with valid data & click on Submit	Branch Name Address1 Address2 Address3 Area Zip code	System displays a message "New Branch Created successfully with Branch ID" with "Ok" Button			
TC009_Ebanking_Admin_Branches_New Branchcreation_Cancel	Verify creation of a new branch by clicking Cancel	Step1	Enter URL & click on go	URL	System displays home page with login facility			
		Step2	Enter valid Username,Password & click on login	Username & Password	System displays Admin module			
		Step3	Click Branches		System displays branches details page			
		Step4	Click New Branch		System displays New branch Entry Page.			
		Step5	Enters necessary fields with valid data and clicks on Cancel	Branch Name Address1 Address2 Address3 Area Zip code	System closes the new branch entry page and displays branches details page.			
TC010_Ebanking_Admin_Branches_New Branchcreation_invalid	Verify new branch creation with invalid data.	Step1	Enter URL & Click go	URL	System displays home page with login facility			
		Step2	Enter valid Username,Password & click on login	Username & Password	System displays Admin module			
		Step3	Click Branches		System displays branches details page.			
		Step4	Click New Branch		System displays New branch Entry Page			
		Step5	Enter necessary fields with invalid data & click on Submit	Branch Name Address1 Address2 Address3 Area Zip code	System displays corresponding Error message			
TC011_Ebanking_Admin_Branches_New Branchcreation_mandtory fields	Verify branch creation by leaving mandatory fields.	Step1	Enter URL & Click go	URL	System displays home page with login facility			
		Step2	Enter valid Username,Password & click on login	Username & password	System displays Admin module			
		Step3	Click Branches		System displays branches details page.			
		Step4	Click New Branch		System displays New branch Entry Page			
		Step5	Entering data by leaving mandatory fields & click on Submit	Branch Name Address1 Address2 Address3 Area Zip code	System displays corresponding Error message			
TC012_Ebanking_Admin_Branches_New Branchcreation_duplicate	Verify branch creation with duplicate data.	Step1	Enter URL & Click go	URL	System displays home page with login facility			
		Step2	Enter valid Username,Password & click on login	Username& password	System displays Admin module			

		Step3	Click Branches		System displays branches details page.			
		Step4	Click New Branch		System displays New branch Entry Page			
		Step5	Entering fields with duplicate data & click on Submit	Branch Name Address1 Address2 Address3 Area Zip code	System displays corresponding Error message			
TC013_Ebanking_Admin_Branches_Branch updation_valid	Verify branch updation with valid data.	Step1	Enter URL & click go	URL	System displays home page with login facility			
		Step2	Enter Username,Password & click on login	Username & password	System displays Admin module			
		Step3	Click Branches		System displays branches details page			
		Step4	Select required Branch and Click on Edit		System displays branch updation page.			
		Step5	modify necessary fields with valid data & click on Update	Branch Name Address1 Address2 Address3 Area Zip code	System displays a message " Branch Updated successfully			
TC014_Ebanking_Admin_Branches_Branch updation_invalid	Verify branch updation with invalid data.	Step1	Enter URL & click go	URL	System displays home page with login facility			
		Step2	Enter Username,Password & click on login	Username & Password	System displays Admin module			
		Step3	Click Branches		System displays branches details page			
		Step4	Select required Branch and Click on Edit		System displays branch updation page.			
		Step5	modify necessary fields with invalid data & click on Update	Branch Name Address1 Address2 Address3 Area Zip code	System displays corresponding error message.			
TC015_Ebanking_Admin_Branches_Branch updation_duplicate	Verify branch updation with existing data.	Step1	Enter URL & click go	URL	System displays home page with login facility			
		Step2	Enter Username,Password & click on login	Username & Password	System displays Admin module			
		Step3	Click Branches		System displays branches details page			
		Step4	Select required Branch and Click on Edit		System displays branch updation page.			
		Step5	modify necessary fields with duplicate data & click on Update	Branch Name Address1 Address2 Address3 Area Zip code	System displays corresponding error message.			

TC016_Ebanking_Admin_Branches_Branch updation_mandatory fields	Verify branch updation by leaving mandatory fields.	Step1	Enter URL & click go	URL	System displays home page with login facility			
		Step2	Enter Username,Password & click on login	Username & Password	System displays Admin module			
		Step3	Click Branches		System displays branches details page			
		Step4	Select required Branch and Click on Edit		System displays branch updation page.			
		Step5	Entering data by leaving mandatory fields & click on Update	Branch Name Address1 Address2 Address3 Area Zip code	System displays corresponding error message.			
TC017_Ebanking_Admin_Branches_Branch deletion_dependency	Verify branch deletion with is having dependency.	Step1	Enter URL & click go	URL	System displays home page with login facility			
		Step2	Enter Username,Password & click on login	Username & Passowrd	System displays Admin module			
		Step3	Click Branches		System displays branches details page.			
		Step4	Select required Branch and click on Delete		System displays a confirmation “Are you sure you want to delete this record?” with “Ok” and “Cancel” buttons			
		Step5	if you want to delete then click on “ok” .		If dependency exists, system will not allow to delete and displays a message “its a shared record cannot be deleted”.			
		Step6	if you do not want to delete then click on "Cancel".		System closes the branch Deletion page and displays branches details page			
TC018_Ebanking_Admin_Branches_Branch deletion_without dependency	Verify branch deletion without any dependency	Step1	Enter URL & click go	URL	System displays home page with login facility			
		Step2	Enter Username,Password & click on login	Username & Password	System displays Admin module			
		Step3	Click Branches		System displays branches details page.			
		Step4	Select required Branch and click on Delete		System displays a confirmation “Are you sure you want to delete this record?” with “Ok” and “Cancel” buttons			
		Step5	if you want to delete then click on “ok” .		System Deletes Selected Branch (if there is no dependency) and displays ”Branch Deleted Successfully” message			
		Step6	if you don't want to delete then click on "Cancel"		System closes the branch Deletion page and displays branches details page			
TC019_Ebanking_Admin_Branches_Searc	Verify Search & Clear.	step1	Enter URL & click go	URL	System displays home page with login facility			

h & Clear		step2	Enter Username,Password & click Login	Username & Password	Ststem displays Admin module.			
		step3	Click Branches		System displays Branch details page			
		step4	Select Proper country/state/city and Click Search		System displays branches based on the selection criteria.			
		step5	Click on Clear		System cleared fields and displays all Branch details			
	Roles							
TC020_Ebanking_Admin_Roles_New Role_valid	Verify new role creation with valid data.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Roles		System displays Role details page			
		step4	click on New Roles		System displays New Role Entry Page.			
		step5	Enter necessary fields with valid data and click on Reset	Role Name Role Description	System clears all fields.			
		step6	Enter necessary fields with valid data and click on Submit	Role Name Role Description	System displays a message "New Role Created successfully with Role Id"			
TC021_Ebanking_Admin_Roles_New Role_Cancel	Verify new role creation by clicking Cancel	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Roles		System displays Role details page			
		step4	click on New Roles		System displays New Role Entry Page.			
		step5	Enter necessary fields with valid data or with out entering any fields click on Cancel	Role Name Role Description	System closes the new role entry page and displays roles details page			
TC022_Ebanking_Admin_Roles__New Role_invalid	Verify new role creation with invalid data.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Roles		System displays Role details page			
		step4	click on New Roles		System displays New Role Entry Page.			
		step5	Enter necessary fields with invalid data and click on Submit	Role Name Role Description	System displays corresponding error message			
TC023_Ebanking_Admin_Roles_New Role_mandatory fields	Verify new role creation by leaving mandatory fields.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Roles		System displays Role details page			
		step4	click on New Roles		System displays New Role Entry Page.			

		step5	Enter data by leaving mandatory fields & click on Submit	Role Description Role Type	System displays corresponding error message			
TC024_Ebanking_Admin_Roles_New Role_duplicate	Verify new role creation by giving duplicate data.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Roles		System displays Role details page			
		step4	click on New Roles		System displays New Role Entry Page.			
		step5	Enter all fileds with duplicate data & click on Submit	Role Description Role Type	System displays corresponding error message			
TC025_Ebanking_Admin_Roles_update Role_valid	Verify role updation with valid data.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & password	Ststem displays Admin module.			
		step3	Click Roles		System displays Roles details page			
		step4	click edit on the respective role to be edited		System displays Roles Update Page			
		step5	Update necessary fields with valid data & click on Update	Role Name Role Description	System displays a message "Roles Updated Successfully"			
TC026_Ebanking_Admin_Roles_update Role_invalid	Verify role updation with invalid data.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & password	Ststem displays Admin module.			
		step3	Click Roles		System displays Roles details page			
		step4	click edit on the respective role to be edited		System displays Roles Update Page			
		step5	Update necessary fields with invalid data & click on Update	Role Name Role Description	System displays corresponding error message			
TC027_Ebanking_Admin_Roles_update Role_mandatory fields	Verify role updation by leaving mandatory fields.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & password	Ststem displays Admin module.			
		step3	Click Roles		System displays Roles details page			
		step4	click edit on the respective role to be edited		System displays Roles Update Page			
		step5	Update role by leaving mandatory fields & click on Update	Role Name Role Description	System displays corresponding error message			
TC028_Ebanking_Admin_Roles_update Role_duplicate	Verify role updation with duplicate data.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & password	Ststem displays Admin module.			
		step3	Click Roles		System displays Roles details page			
		step4	click edit on the respective role to be edited		System displays Roles Update Page			

		step5	Update role by entering duplicate data fields & click on Update	Role Description Role Type	System displays corresponding error message			
TC029_Ebanking_Admin_Roles_delete Role_dependency	Verify role deletion with dependency.	step1	Enter URL & click on go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & password	Ststem displays Admin module.			
		step3	Click Roles		System displays Roles details page			
		step4	click on delete on the respective role to be deleted		System displays a confirmation “Are you sure you want to delete this record?” with “Ok” and “Cancel” buttons			
		step5	if you want to delete then click on “ok” .		If dependency exists, system will not allow to delete and displays a message “its a shared record cannot be deleted”.			
		step6	if you do not want to delete then click on "Cancel".		System closes the Role Deletion page and displays Role details page			
TC030_Ebanking_Admin_Roles_delete Role_without dependency	Verify role deletion without any dependency.	step1	Enter URL & click on go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & password	Ststem displays Admin module.			
		step3	Click Roles		System displays Roles details page			
		step4	click on delete on the respective role to be deleted		System displays a confirmation “Are you sure you want to delete this record?” with “Ok” and “Cancel” buttons			
		step5	if you want to delete then click on “ok” .		System Deletes Selected Branch (if there is no dependency) and displays ”Role Deleted Successfully” message			
		step6	if you don't want to delete then click on "Cancel"		System closes the Role Deletion page and displays Role details page			
	Users							
TC031_Ebanking_Admin_Users_New User_valid	Verify User Creation with valid data	step1	Enter URL & click on go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & password	Ststem displays Admin module.			
		step3	Click Users		System displays user details page.			
		step4	click on New user		System displays New user creation Page.			
		step5	enters necessary fields with valid data and clicks Reset.	Customer Name User Name Login Password Transaction Password	System clears all fields.			

		step6	Enter necessary fields with valid data and click on Submit.	Customer Name User Name Login Password Transaction Password	System displays a message "New User Created successfully with User ID"			
TC032_Ebanking_Admin_Users_New User_Cancel	Verify User Creation with valid data by clicking Cancel	step1	Enter URL & click on go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & password	Ststem displays Admin module.			
		step3	Click Users		System displays user details page.			
		step4	click on New user		System displays New user creation Page.			
		step5	Enters necessary fields with valid data or with out entering any fields clicks Cancel.	Customer Name User Name Login Password Transaction Password	System closes the new user creation page and displays user details page.			
TC033_Ebanking_Admin_Users_New User_invalid	Verify User Creation with invalid data	step1	Enter URL & click on go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & password	Ststem displays Admin module.			
		step3	Click Users		System displays user details page.			
		step4	click on New user		System displays New user creation Page.			
		step5	Enter necessary fields with invalid data and click on Submit.	Customer Name User Name Login Password Transaction Password	System displays corresponding error message			
TC034_Ebanking_Admin_Users_New User_mandatory fields	Verify User Creation by leaving mandatory fields	step1	Enter URL & click on go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Users		System displays user details page.			
		step4	click on New user		System displays New user creation Page.			
		step5	Entering data by leaving mandatory fields & click on Submit.		System displays corresponding error message			
TC035_Ebanking_Admin_Users_New User_duplicate	Verify User Creation by entering duplicate data.	step1	Enter URL & click on go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Users		System displays user details page.			
		step4	click on New user		System displays New user creation Page.			
		step5	Enter duplicate data & click on Submit.	User Name Login Password Transaction Password	System displays corresponding error message			
TC036_Ebanking_Admin_Users_Update	Verify User Updation with valid data.	step1	Enter URL & click on go	URL	System displays home page with login facility			

User_valid		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Users		System displays users details page.			
		step4	click edit for any user		System displays user updation page.			
		step5	modify necessary fields with valid data and click on Update.	User Name Login Password Transaction Password	System displays a message "User Updated successfully"			
TC037_Ebanking_Admin_Users_Update User_invalid	Verify User Updation with invalid data.	step1	Enter URL & click on go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Users		System displays users details page.			
		step4	click edit for any user		System displays user updation page.			
		step5	modify necessary fields with invalid data and click on Update.	User Name Login Password Transaction Password	System displays corresponding error message			
TC038_Ebanking_Admin_Users_Update User_mandatory fields	Verify User Updation by leaving mandatory fields.	step1	Enter URL & click on go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Users		System displays users details page.			
		step4	click edit for any user		System displays user updation page.			
		step5	Modify by leaving mandatory fields & click on Update.	User Name Login Password Transaction Password	System displays corresponding error message			
TC039_Ebanking_Admin_Users_Update User_duplicate	Verify User Updation by entering duplicate data.	step1	Enter URL & click on go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Users		System displays users details page.			
		step4	click edit for any user		System displays user updation page.			
		step5	Modify with duplicate data & click on Update.	User Name Login Password Transaction Password	System displays corresponding error message			
TC040_Ebanking_Admin_Users_Delete User	Verify User Deletion.	step1	Enter URL & click on go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Users		System displays users details page.			
		step4	click Delete for any user		System displays a confirmation window with "Yes" and "No" options.			
		step5	if you want to delete then click on "OK"		System deletes the user.			

		step6	if you don't want to delete then click on "Cancel"		System closes the User Deletion page and displays User details page			
TC041_Ebanking_Admin_Users_Search & Clear	Verify Search & clear.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Users		System displays users details page.			
		step4	Select Proper country/state/city		System displays Users based on the selection criteria.			
		step5	Click on Clear		System cleared fields and displays all User details			
	Employees							
TC042_Ebanking_Admin_Employee_New Employee_valid	Verify Employee Creation with valid data.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Employee		System displays employee details page.			
		step4	click on New Employee		System displays New Employee Entry Page.			
		step5	Enter necessary fields with valid data and clicks Submit.	Employee Name Login Password	System displays a message "Employee Created Successfully with Employee ID"			
		step6	Enters necessary fields with valid data and clicks Reset.	Employee Name Login Password	System clears all fields.			
TC043_Ebanking_Admin_Employee_New Employee_Cancel	Verify Employee Creation by clicking Cancel	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Employee		System displays employee details page.			
		step4	click on New Employee		System displays New Employee Entry Page.			
		step5	Enters necessary fields with valid data or with out entering any fields clicks Cancel.	Employee Name Login Password	System closes the new Employee entry page and displays Employee details page.			
TC044_Ebanking_Admin_Employee_New Employee_invalid	Verify Employee Creation with invalid data.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Employee		System displays employee details page.			
		step4	click on New Employee		System displays New Employee Entry Page.			
		step5	Enter necessary fields with invalid data and clicks Submit.	Employee Name Login Password	System displays corresponding error message			
TC045_Ebanking_Admin_Employee_New	verify Employee Creation by leaving mandatory fields	step1	Enter URL & click go	URL	System displays home page with login facility			

Employee_mandatory fields		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Employee		System displays employee details page.			
		step4	click on New Employee		System displays New Employee Entry Page.			
		step5	Enter data by leaving mandatory fields & click on Submit.		System displays corresponding error message			
TC046_Ebanking_Admin_Employee_New Employee_duplicate	verify by entering duplicate data fields	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Employee		System displays employee details page.			
		step4	click on New Employee		System displays New Employee Entry Page.			
		step5	Enter duplicate data fields & click on Submit.	Employee Name Login Password	System displays corresponding error message			
TC047_Ebanking_Admin_Employee_Employee update_valid	verify Employee Updation by entering valid data.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Employee		System displays employee details page.			
		step4	click on Edit Employee		System displays Employee Updation Entry Page.			
		step5	click edit on the respective employee details to be edited		System displays Employee Update Page.			
		step6	Update necessary fields with valid data and clicks Update.	Employee Name Login Password	System displays a message "Updated Successfully"			
TC048_Ebanking_Admin_Employee_Employee update_invalid	verify Employee Updation by entering invalid data.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Employee		System displays employee details page.			
		step4	click on Edit Employee		System displays Employee Updation Entry Page.			
		step5	click edit on the respective employee details to be edited		System displays Employee Update Page.			
		step6	Update necessary fields with invalid data and clicks Update.	Employee Name Login Password	System displays corresponding error message			
TC049_Ebanking_Admin_Employee_Employee update_mandatory fields	verify Employee Updation by leaving mandatory fields.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Employee		System displays employee details page.			
		step4	click on Edit Employee		System displays Employee Updation Entry Page.			
		step5	click edit on the respective employee details to be edited		System displays Employee Update Page.			

		step6	Update an employee by leaving mandatory fields & click on Update.	Employee Name Login Password	System displays corresponding error message			
TC050_Ebanking_Admin_Employee_Employee update_duplicate	verify Employee Updation by entering duplicate data fields.	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Employee		System displays employee details page.			
		step4	click on Edit Employee		System displays Employee Updation Entry Page.			
		step5	click edit on the respective employee details to be edited		System displays Employee Update Page.			
		step6	Update an employee by entering duplicate data fields & click on Update.	Employee Name Login Password	System displays corresponding error message			
TC051_Ebanking_Admin_Employee_Employee Deletion	verify Employee Deletion	step1	Enter URL & click go	URL	System displays home page with login facility			
		step2	Enter Username,Password & click on Login	Username & Password	Ststem displays Admin module.			
		step3	Click Employee		System displays employee details page.			
		step4	click Delete on the respective role to be deleted.		System displays a message “Are you sure you want to delete this record” with “Ok” and “Cancel” buttons			
		step5	if you want to delete then click on "OK"		System deletes the Employee.			
		step6	if you don't want to delete then click on "Cancel"		System closes the Employee Deletion page and displays Employee details page.			

Project Name	OrangeHRM Version 3.0 – My Info Module
Reference Document	Project Functional Requirement Specification , Version 1
Created by	© Prepared by: Kamal Subramani
Date of creation	13-Feb-14
Date of review	20-Feb-14

Test case ID	Test Objective	Precondition	Steps:	Test data	Expected result	Post-condition
TC_MI_01	Successful Employee login to OrangeHRM portal Check the screenshot to get an idea of what screen we are testing  Pic 1.jpg	1. A valid ESS-User account to login to be available 2. Orange HRM 3.0 site is launched on a compatible browser	1. In the login Panel, enter the username 2. Enter the Password for the ESS-User account in the password field 3. Click "Login" button	"A valid username" Enter the actual data in your real time situation "A valid Password"	The user is logged in successfully. There is only one expected result for the entire test case. However, that does not have to be so. If it makes sense that for every step, you want to write the result of exactly what happens with it, please free to have an expected result for each test step	For first time users personal information is displayed. Note: This info is only additional. Just as a pointer to the tester
TC_MI_02	Error message on unsuccessful Employee login to OrangeHRM portal launched	1. A ESS-User name to login to be available 2. Orange HRM 3.0 site is launched on a compatible browser	In the login Panel, enter 1. the username 2. Enter the Password for the ESS-User account in the password field 3. Click "Login" button	"A valid username" "A invalid Password"	An Error message is displayed and the user is not logged in to the Orange HRM portal. "<Exact Error Message>" In the test case, it is not enough when we say, 'that an error is displayed'- in addition to that, we will have to mention the exact error message that is going to be encountered by the user- This information can generally be found in FRD(SRS). If not, look in the technical design document or Use cases.- Check the test next case, where we write Expected result, step wise.	As you can see, post condition can be left empty when there is nothing else to add
TC_MI_MIM_01	First time user login-information display check  Pic 2.jpg	1. A ESS-User name to login to be available for a first time user of the site. 2. Orange HRM 3.0 site is launched on a compatible browser	1. In the login Panel, enter the username 2. Enter the Password for the ESS-User account in the password field 3. Click "Login" button 4. Check the fields on the "Personal information Page"	"A valid username" "A valid Password"	The user is logged in successfully and the personal information page is displayed Check if the following fields are disabled for entry in Personal Details: • Employee ID • SSN No • SIN No	These fields are grayed out and cannot be modified

						• Driver License No • Date of Birth	

TC_MI_MIM_02	Personal details- modification with valid values- "First Name"	1. Orange HRM 3.0 site is launched	o 1. Check the fields on the "Personal information Page" 2. Change the field, "First Name" with enter a valid new name in this field 3. Click on "Save"		The users information is displayed "A valid new value for the first name field" The first name field needs to now show the new value entered	

* This is how all the other field's modification can be tested. Be sure to include negative test cases, where an invalid data can be entered as the new value and an error message is observed

TC_MI_P_01	Check the upload of a JPG format image  	1. Orange HRM 3.0 site is launched on a compatible browser and a ESS User account holder is logged in to the site 2. A valid image to upload that is JPG in format and less than 1 MB in size is available on the local machine a location	Click on the photograph displayed at the top left corner of the page		The "Photograph screen" will be displayed	This page will contain options to select and upload pictures.
			Click on "Choose a file" button		You will be able to browse your local machine for images	
			Choose a image file of type "JPG" that is less than 1 MB	Name of the image Location-path on the machine	The file name is selected in the "Choose a file" box	
			Click on upload		The file gets uploaded and the older image is replaced	It takes 2-5 depending on the size of the image for this change to complete and the page to refresh with the new image.
TC_MI_P_02	Check the upload of a PNG format image	1. Orange HRM 3.0 site is launched on a compatible browser and a ESS User account holder is logged in to the site 2. A valid image to upload that is PNG in format and less than 1 MB in size is available on the local machine a location	Click on the photograph displayed at the top left corner of the page		The "Photograph screen" will be displayed	This page will contain options to select and upload pictures.
			Click on "Choose a file" button		You will be able to browse your local machine for images	
			Choose a image file of type PNG that is less than 1 MB	Name of the image Location-path on the machine	The file name is selected in the "Choose a file" box	
			Click on upload		The file gets uploaded and the older image is replaced	It takes 2-5 depending on the size of the image for this change to complete and the page to refresh with the new image.

*include another similar test cases for all the valid formats

TC_MI_P_03	Check the upload of a invalid format of the picture (may be a .doc file) that is less than 1 MB	1. Orange HRM 3.0 site is launched on a compatible browser and a ESS User account holder is logged in to the site 2. A doc file is available that is less	Click on the photograph displayed at the top left corner of the page		The "Photograph screen" will be displayed	This page will contain options to select and upload pictures.
			Click on "Choose a file" button		You will be able to browse your local machine for images	

		than 1 MB	Choose the doc file that is less than 1 MB	Name of the image Location-path on the machine	The file name is selected in the "Choose a file" box	
			Click on upload		An error message is displayed that the format is not supported- <The exact error message>	
TC_MI_P_04	Check the upload of a valid format by over the size of 1 MB	1. Orange HRM 3.0 site is launched on a compatible browser and a ESS User account holder is logged in to the site 2. A JPG file is available that is more than 1 MB	Click on the photograph displayed at the top left corner of the page Click on "Choose a file" button Choose the JPG file that is more than 1 MB Click on upload	Name of the image Location-path on the machine	The "Photograph screen" will be displayed You will be able to browse your local machine for images The file name is selected in the "Choose a file" box An error message is displayed that the size is over the limit - <The exact error message>	This page will contain options to select and upload pictures.

* Similarly, try writing test cases for multiple valid and invalid formats and sizes

RTM

By

Kamal Subramani

Payilagam

Sample Requirements Traceability Matrix

Prepared By: KAMAL SUBRAMANI

BRD- Section	FSD- Section	Test scenario ID	Test case ID	Status	Defects
1- Loan Process	1.1- New users	TS_Loan_001- Validate the "Apply Loan" feature as a new user	TC_newuser_01	Passed	
			TC_newuser_02	Passed	
			TC_newuser_03	Failed	Defect_01, Defect_02
		TS_Loan_002- validate the "Apply Loan" feature as a already existing user	TC_newuser_04	Passed	
			TC_newuser_05	Blocked	Defect_01
			TC_newuser_06	Failed	Defect_03
			TC_newuser_07	Passed	
		TS_Loan_003- For a new user in the "Apply loan", check the guest customer option and apply loan	TC_newuser_08	Passed	
		TS_Loan_004 - For a new user in the "Apply loan", check the Register option and apply loan	TC_newuser_09	Passed	
	1.2- Existing users	TS_Loan_005- Login to the loan portal as an already a customer with a loan and check the information displayed	TC_Exist_User_01	Passed	
		TS_Loan_006-Check the Loan whose status is "Sent for review"	TC_Exist_User_02	Passed	
		TS_Loan_007-Check the Loan whose status is "Reviewed and Accepted"	TC_Exist_User_03	Passed	
		TS_Loan_008-Check the Loan whose status is "Reviewed and deleted"	TC_Exist_User_04	Passed	
2- Ease of use	2.1- Ease os use	TS_Loan_009-Check for a visitor if the information on the site is accessible in less than 3 clicks or not	TC_EasyUse_01	Passed	
		TS_Loan_010-Check for a registered user if the information on the site is accessible in less than 3 clicks or not	TC_EasyUse_02	Passed	
		TS_Loan_011-Check for a banker if the information on the site is accessible in less than 3 clicks or not	TC_EasyUse_03	Passed	

TEST CASES EXECUTION RESULT

By

Kamal Subramani

Payilagam

Smoke Testing

Does the link http://opensource.demo.orangehrm.com/ open up without any issues after code deployment?
Are there any broken links in the first level of navigation?
Are the user IDs we have to continue testing functional?
* The most important point of check at this point is, whether we have a system available to test or not.

Sanity Testing

***The most critical business functionality is tested in this stage.**

1. ESS-User account holder can login to the OrangeHRM system
2. Are the personal details visible?
3. Is the "Edit" button on the personal details page functional?
4. Is the personal details page's save option working?
5. Photo upload working or not? (upload one sample valid photo to check this)

.....and so on

A rule of thumb where sanity testing is concerned is that, even if one sanity test case fails, the build will not be accepted for testing.

Exploratory Test Results

*** Even though exploratory testing is non-documentation led, the results(issues identified) have to be documented. Informally initially and then raise a defect**

Exploratory testing can be performed based on the project knowledge and experience.
A sample exploratory testing result can be as follows:

Visited the OrangeHRM demo page
http://opensource.demo.orangehrm.com/ from Google search results. For no reason it showed me the login page in Vietnamese language instead of English. When I refreshed the same page it showed me the login page in English.
See screenshot attached.


OrangeHRM issue 1.jpg

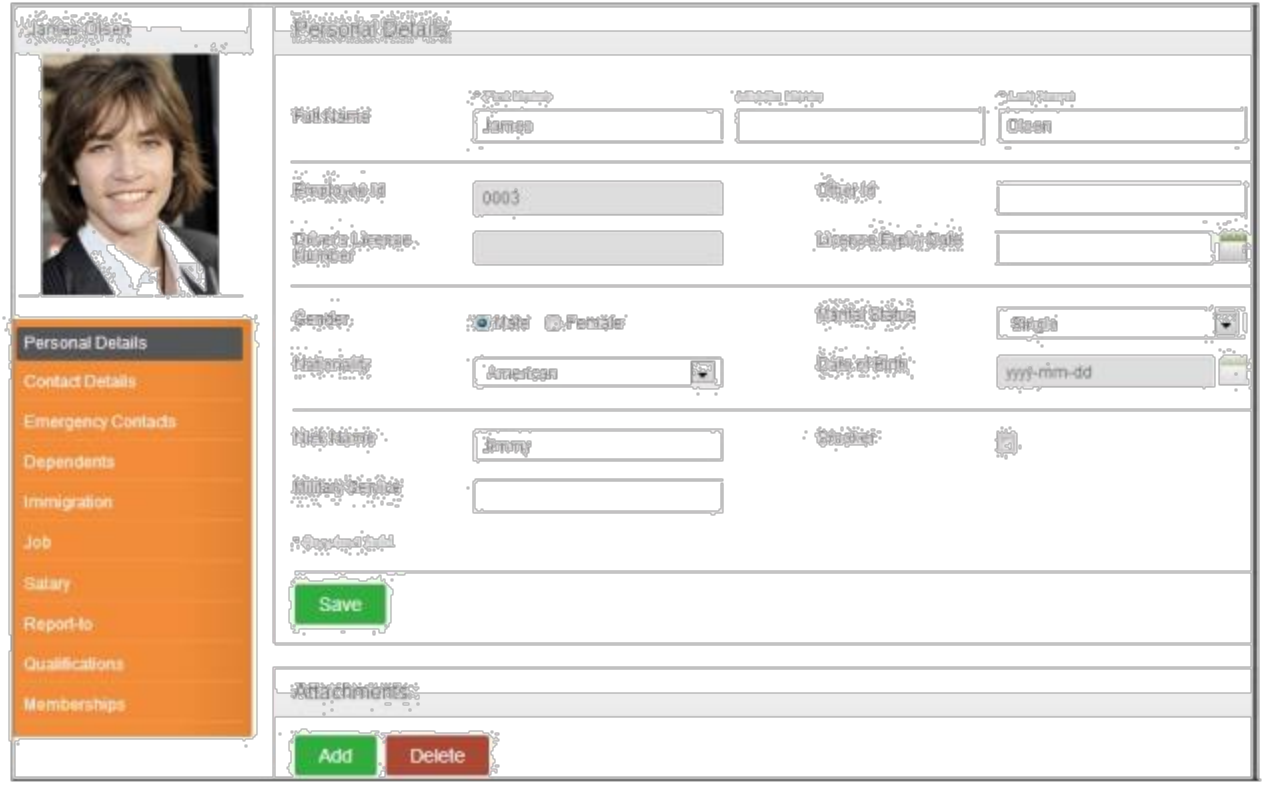
Functional Test

Project Name	**OrangeHRM Version 3.0 – My Info Module**
Reference Document	**Project Functional Requirement Specification , Version 1**
Created by	**© Prepared By: Kamal Subramani**
Date of creation	**13-Feb-14**
Date of review	**24-Feb-14**

Test case ID	Test Objective	Precondition	Steps:	Test data	Expected result	Post-condition	Status		Actual Result
TC_MI_01	Successful Employee login to OrangeHRM portal Check the screenshot to get an idea of what screen we are testing Pic 1.jpg	1. A valid ESS-User account to login to be available 2. Orange HRM 3.0 site is launched on a compatible browser	1. In the login Panel, enter the username	"A valid username"	The user is logged in successfully.	For first time users personal information is displayed.		**Pass**	
			2. Enter the Password for the ESS-User account in the password field	"A valid Password"					
			3. Click "Login" button						
TC_MI_02	Error message on unsuccessful Employee login to OrangeHRM portal	1. A ESS-User name to login to be available 2. Orange HRM 3.0 site is launched on a compatible browser	1. In the login Panel, enter the username	"A valid username"	An Error message is displayed and the user is not logged in to the Orange HRM portal. "<Exact Error Message>"			**Pass**	
			2. Enter the Password for the ESS-User account in the password field	"A invalid Password"					
			3. Click "Login" button						
TC_MI_MIM_01	First time user login-information display check Pic 2.jpg	rst time user of the site.2. Orange HR	1. In the login Panel, enter the username	"A valid username"				**Pass**	
			2. Enter the Password for the ESS-User account in the password field	"A valid Password"					
			3. Click "Login" button		The user is logged in successfully and the personal information page is displayed				
			4. Check the fields on the "Personal information Page"		Check if the following fields are disabled for entry in Personal Details: • Employee ID • SSN No • SIN No • Driver License No • Date of Birth	These fields are grayed out and cannot be modified			
TC_MI_MIM_02	Personal details- modification with valid values- "First Name"	1. Orange HRM 3.0 site is launched o	1. Check the fields on the "Personal information Page"		The users information is displayed			**Pass**	
			2. Change the field, "First Name" with enter a valid new name in this field	"A valid new value for the first name field"					
			3. Click on "Save"		The first name field needs to now show the new value entered				

TC_MI_P_01	Check the upload of a JPG format image  	1. Orange HRM 3.0 site is launched on a compatible browser and a ESS User account holder is logged in to the site 2. A valid image to upload that is JPG in format and less than 1 MB in size is available on the local machine a location	Click on the photograph displayed at the top left corner of the page		The "Photograph screen" will be displayed	This page will contain options to select and upload pictures.	**Pass**	
			Click on "Choose a file" button		You will be able to browse your local machine for images			
			Choose a image file of type "JPG" that is less than 1 MB	Name of the image Location-path on the machine	The file name is selected in the "Choose a file" box			
			Click on upload		The file gets uploaded and the older image is replaced	It takes 2-5 depending on the size of the image for this change to complete and the page to refresh with the new image.		
TC_MI_P_02	Check the upload of a PNG format image	1. Orange HRM 3.0 site is launched on a compatible browser and a ESS User account holder is logged in to the site 2. A valid image to upload that is PNG in format and less than 1 MB in size is available on the local machine a location	Click on the photograph displayed at the top left corner of the page		The "Photograph screen" will be displayed	This page will contain options to select and upload pictures.	**Blocked (could not obtain a png file)**	
			Click on "Choose a file" button		You will be able to browse your local machine for images			
			Choose a image file of type PNG that is less than 1 MB	Name of the image Location-path on the machine	The file name is selected in the "Choose a file" box			
			Click on upload		The file gets uploaded and the older image is replaced	It takes 2-5 depending on the size of the image for this change to complete and the page to refresh with the new image.		
TC_MI_P_03	Check the upload of a invalid format of the picture that is less than 1 MB	1. Orange HRM 3.0 site is launched on a compatible browser and a ESS User account holder is logged in to the site 2. A exe file is available that is less than 1 MB	Click on the photograph displayed at the top left corner of the page		The "Photograph screen" will be displayed	This page will contain options to select and upload pictures.	**Fail**	The message of 'Invalid file type' should appear (as only JPG, GIF and PNG files supported) after the save button. But it uploads the complete exe file and then gives a message that the file type is invalid.
			Click on "Choose a file" button		You will be able to browse your local machine for images			
			Choose the doc file that is less than 1 MB	audacity.exe on C://	The file name is selected in the "Choose a file" box			
			Click on upload		An error message is displayed that the format is not supported- <The exact error message>			
TC_MI_P_04	Check the upload of a valid format by over the size of 1 MB	1. Orange HRM 3.0 site is launched on a compatible browser and a ESS User account holder is logged in to the site 2. A JPG file is available that is more than 1 MB	Click on the photograph displayed at the top left corner of the page		The "Photograph screen" will be displayed	This page will contain options to select and upload pictures.	**No run**	
			Click on "Choose a file" button		You will be able to browse your local machine for images			
			Choose the JPG file that is more than 1 MB	Name of the image Location-path on the machine	The file name is selected in the "Choose a file" box			
			Click on upload		An error message is displayed that the size is over the limit - <The exact error message>			


OrangeHRM
OPEN SOURCE HR MANAGEMENT
HR for ALL
LOGIN Panel
Admin
LOGIN
OrangeHRM ver 2.7 © OrangeHRM Inc. 2005 - 2012 All rights reserved.


James Olsen
Personal Details
Full Name:
James
Olsen
Employee Id
0003
Driver's License Number
Gender:
Male
Female
Nationality
American
Nick Name
Jimmy
Military Service
Save
Attachments
Add
Delete
Personal Details
Contact Details
Emergency Contacts
Dependents
Immigration
Job
Salary
Report-to
Qualifications
Memberships

James Olsen

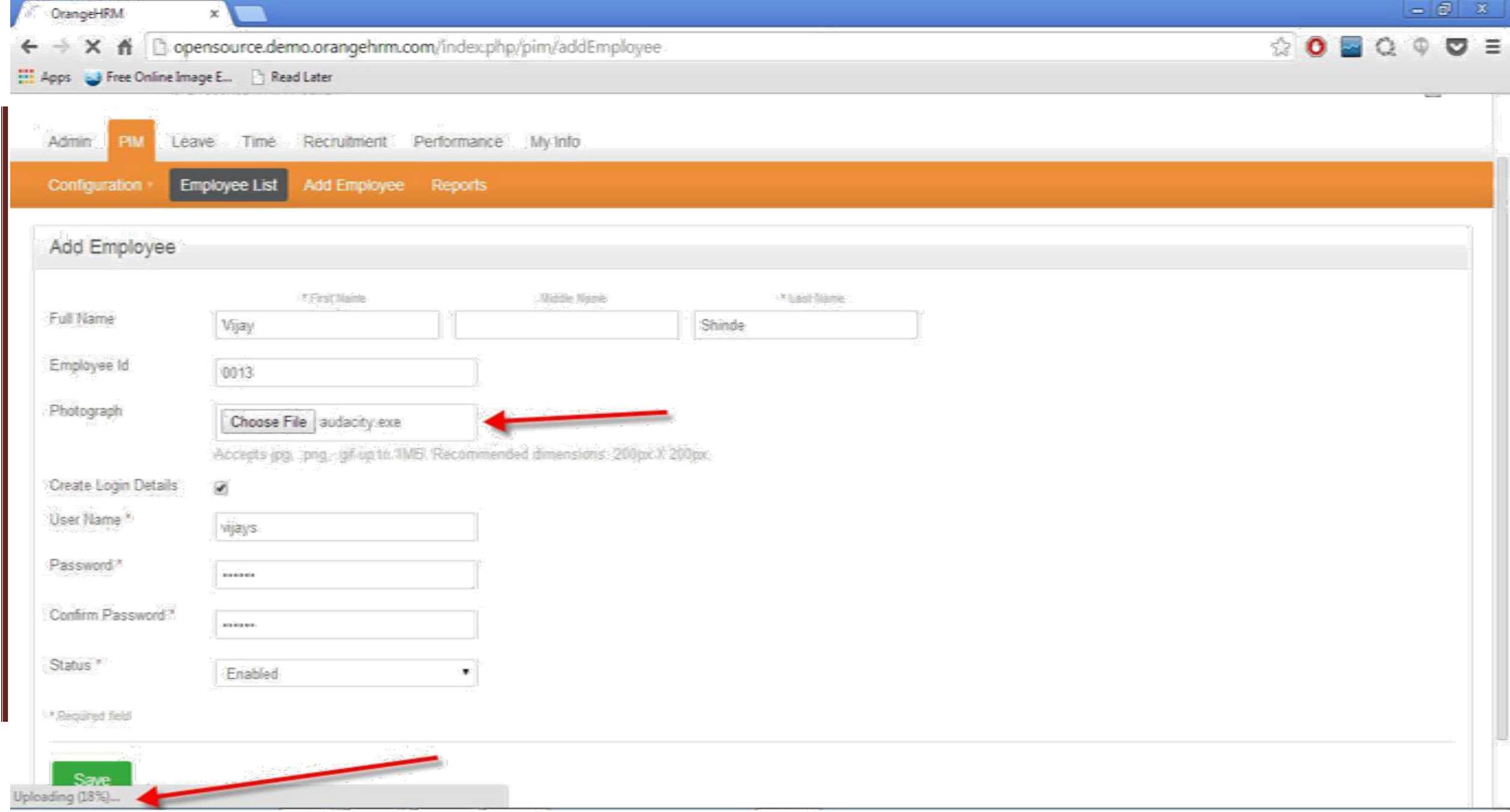
Photograph

Select a Photograph

Choose File | Image.jpg

Accepts jpg, png, gif up to 2MB. Recommended dimensions: 250px x 200px

Upload | Delete


OrangeHRM
opensource.demo.orangehrm.com/index.php/pim/addEmployee
Apps
Free Online Image E...
Read Later
Admin
PIM
Leave
Time
Recruitment
Performance
My Info
Configuration
Employee List
Add Employee
Reports
Add Employee
* First Name
Middle Name
* Last Name
Full Name
Vijay
Shinde
Employee Id
0013
Photograph
Choose File
audacity.exe
Accepts jpg, .png, .gif up to 1MB. Recommended dimensions: 200px X 200px
Create Login Details
User Name *
vijays
Password *
Confirm Password *
Status *
Enabled
* Required field
Save
Uploading (18%)...

DEFECT REPORT

By

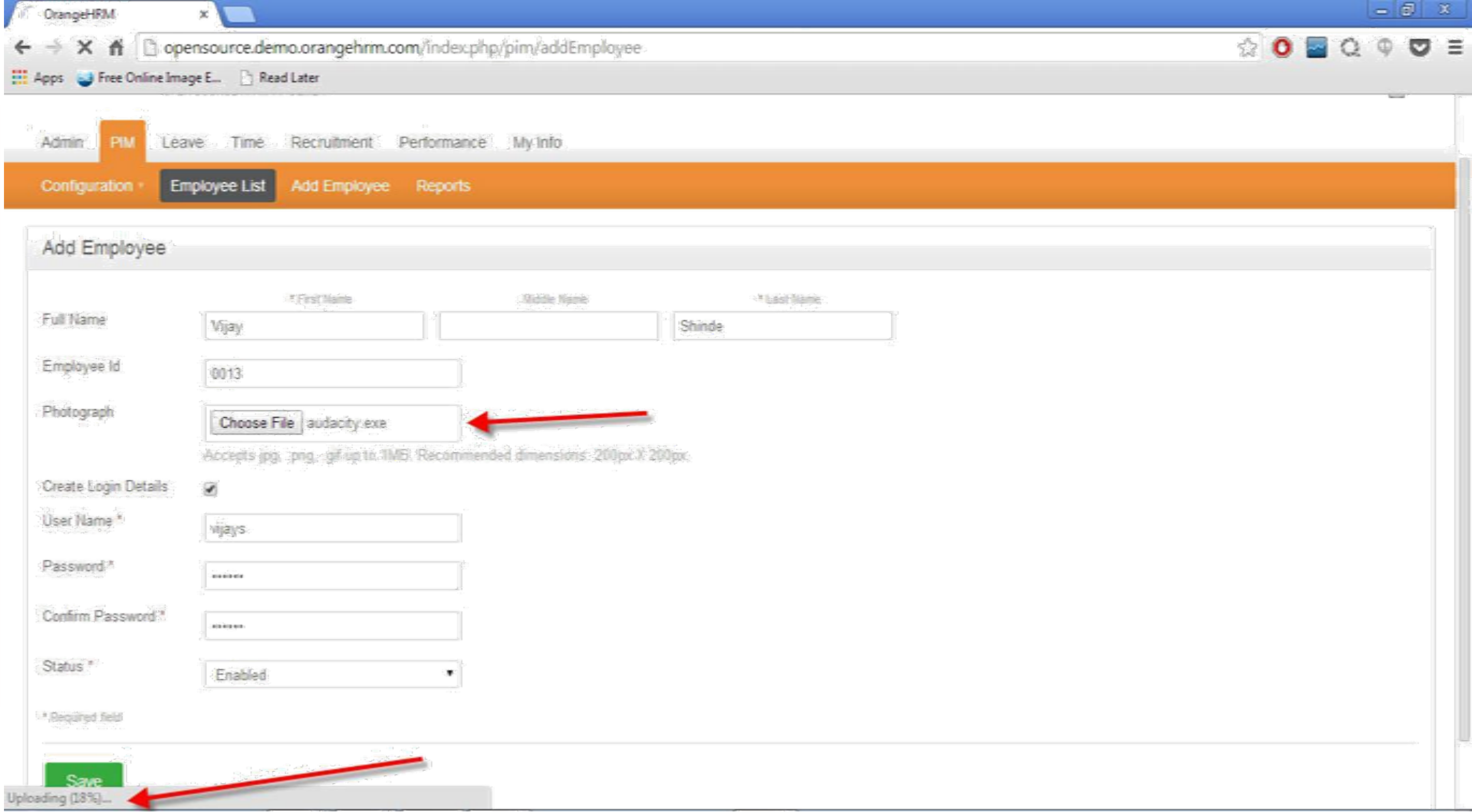
Kamal Subramani

Payilagam

OrangeHRM defect report- Cycle 1 Testing

Defect ID	Defect Description	Steps to reproduce	Severity	Status	Created by	Assigned to	Screenshot
1	Launching OrangeHRM from google search results page causes the site to be open in the vietnamese language	1. Launch google.com and search for Orange HRM website 2. click on the the OrangeHRM demo page http://opensource.demo.orangehrm.com/ from Google search results. Expected result: The OrangeHRM page should be displayed in English Actual Result: The site is displayed in vietnamese Note: The site goes back to being displayed in English, on refreshing the page.	1-medium	Open	QA team member	QA lead	OrangeHRM issue 1.jpg
2	The invalid file format error message for invalid photographs is appearing after the upload is complete	1. launch OrangeHRM and login as a employee 2. Click on the photograph displayed at the top left corner of the personal details page 3. Click on "Choose a file" button and choose a ".exe" file that is less than 1 MB 4. Click Upload Expected result: An error should be displayed that it is an invalid file format. Actual Result: The message of 'Invalid file type' should appear (as only JPG, GIF and PNG files supported) after the Upload button. But it uploads the complete exe file and then gives a message that the file type is invalid.	1- Medium	Open	QA team member	QA team lead	OrangeHRM issue 2.jpg


OrangeHRM
OPEN SOURCE HR MANAGEMENT
(Username : Admin | Password : admin)
ĐĂNG NHẬP Bảng điều chỉnh
Tên đăng nhập
Mật khẩu
ĐĂNG NHẬP
HR for ALL


OrangeHRM
opensource.demo.orangehrm.com/index.php/pim/addEmployee
Apps
Free Online Image E...
Read Later
Admin
PIM
Leave
Time
Recruitment
Performance
My Info
Configuration
Employee List
Add Employee
Reports
Add Employee
Full Name
Vijay
Shinde
Employee Id
0013
Photograph
Choose File
audacity.exe
Create Login Details
User Name *
vijays
Password *
Confirm Password *
Status *
Enabled
* Required field
Save
Uploading (18%)...

SAMPLE STATUS REPORT

By

Kamal Subramani

Payilagam

SUMMARY

Overall progress of the QA cycle(Ontime, delayed, Stopped)	On time
Total number of test cases	100
Number of testers	5
Test cycle duration	5 days

The gap in the total executed and planned can be absorbed by the QA team

Status for 08/31/2013	
Number of test cases planned	20
Number of test cases executed	18
Number of test cases executed overall	78
Number of defects encountered today	2
Number of defect encountered so far	10
Number of critical defects- still open	3

Overall status	
Number of test cases planned	100
Number of test cases executed	78
Pass Percentage of the defects	98%
Defects density	2.5 per day
Critical defects percentage	20%

DETAILED STATUS

Module	Scenarios	Sub Levels	Complexity	Responsible tester	Date of Execution(Can be past, present or future date)	Status(Pass/Fail/ blocked/not executed)	Defect ID- Brief description	Severity	Status
Admin	Login Page	Login	Medium	Tester A	31-08-13	Pass			
Admin	Country Management	Add country	Complex	Tester B	31-08-13	Pass			

		Delete country	Complex	Tester B	31-08-13	Pass			
		Verify the list display	Medium	Tester B	31-08-13	Pass			
Admin	City management	Add City	Complex	Tester A	31-08-13	Pass			
		Delete City	Complex	Tester A	31-08-13	Pass			
		Verify the list display	Medium	Tester A	31-08-13	Pass			
Admin	Car Make Management	Add Car Make	Complex	Tester B	31-08-13	Fail	1028: Cannot add car make, 404 error on clicking the link	1- High	Open
		Delete car make	Complex	Tester B	31-08-13	Blocked	1028		
		Verify the list display	Medium	Tester B	31-08-13	Pass			
Admin	Car Model Management	Add Car Model	Complex	Tester X	31-08-13	Pass			
		Delete car model	Complex	Tester X	31-08-13	Pass			
		Verify the list display	Medium	Tester X	31-08-13	Fail	1029: List not displayed. Table contents empty always	2- Medium	Assigned
Admin	Featured car	Select all	Medium	Tester X	01-09-13	Not executed			
		Select New	Complex	Tester X	31-08-13	Pass			
		deselect some	Complex	Tester X	31-08-13	Pass			
		deselect all	Medium	Tester A	01-09-13	Not executed			

BUG REPORT

Defect ID	Defect Description	Steps to reproduce	Severity	State	Created by	Assigned to
1028	Cannot add car make, 404 error on clicking the link	1. Login to the site using the admin credentials 2. Click on the "Car Make Management" menu option from the left side navigation 3. The list of the makes of the cars available will be displayed 4. There will also be a link "Add new car make" present on the top(header) of the list 5. Click on the link Expected result: The page where the admin can enter a new make for the car should be displayed Actual Result: A 404 error causes the system to crash. The site has to be relaunched to recover from this state.	1- High	Open	Tester X	Dev representative

TEST SUMMARY REPORT

By

Payilagam

Sample Test Summary Report Created and published by:

Test Summary Report

1. Purpose

This document explains the various activities performed as part of Testing of 'ABCD transport system' application.

2. Application Overview

'ABCD transport system' is a web based Bus ticket booking application. Tickets for various buses can be booked using the online facilities. Real time passenger information is received from a 'Central repository system', which will be referred before booking is confirmed. There are several modules like Registration, Booking, Payment and Reports which are integrated to fulfill the purpose.

3. Testing Scope

<This section explains about the functions/modules in scope & out of scope for testing; Any items which are not tested due to any constraints/dependencies/restrictions. **Example:** A functionality verification which needs connectivity to a third party application cannot be tested, as the connectivity could not be established due to some technical limitations. This section should be clearly documented, else it will be assumed that Testing covered all areas of the application>

a) In Scope

Functional Testing for the following modules are in Scope of Testing

- Registration
- Booking
- Payment

b) Out of Scope

Performance Testing was not done for this application.

c) Items not tested

Verification of connectivity with the third party system 'Central repository system' was not tested, as the connectivity could not be established due to some technical limitations. This can be verified during UAT (User Acceptance Testing) where the connectivity is available or can be established.

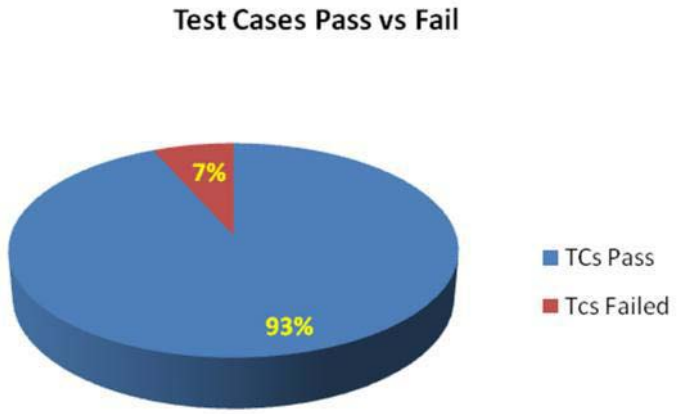
4. Metrics

<Metrics will help to understand the test execution results, status of test cases & defects etc. Required Metrics can be added as necessary. Example: Defect Summary-Severity wise; Defect Distribution-Function/Module wise; Defect Ageing etc.. Charts/Graphs can be attached for better visual representation>

d) No. of test cases planned vs executed

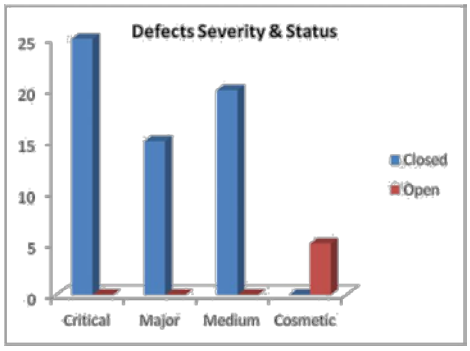
e) No. of test cases passed/failed

Test cases planned	Test cases executed	TCs Pass	Tcs Failed
80	75	70	5



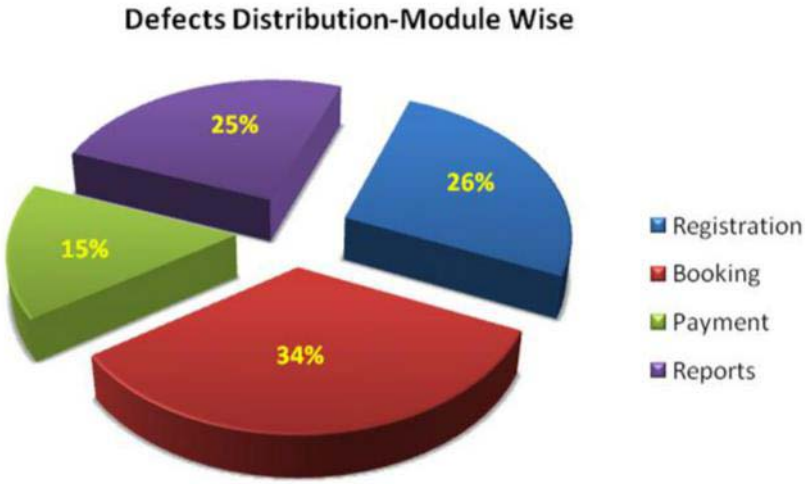
f) No of defects identified and their Status & Severity

	Critical	Major	Medium	Cosmetic	Total
Closed	25	15	20	0	60
Open	0	0	0	5	5
					65



g) Defects distribution – module wise

	Registration	Booking	Payment	Reports	Total
Critical	6	7	5	7	**25**
Major	4	5	2	4	**15**
Medium	6	8	2	4	**20**
Cosmetic	1	2	1	1	**5**
Total-->	**17**	**22**	**10**	**16**	**65**



5. Types of testing performed

a) Smoke Testing

This testing was done whenever a Build is received *(deployed into Test environment)* for Testing to make sure the major functionalities are working fine, Build can be accepted and Testing can start.

b) System Integration Testing

- This is the Testing performed on the Application under test, to verify the entire application works as per the requirements
- Critical Business scenarios were tested to make sure important functionalities in the application works as intended without any errors.

c) Regression Testing

- Regression testing was performed each time a new build is deployed for testing which contains defect fixes and new enhancements, if any.
- Regression Testing is being done on the entire application and not just the new functionalities and Defect fixes.
- This testing ensures that existing functionalities works fine after defect fix and new enhancements are added to the existing application.
- Test cases for new functionalities are added to the existing test cases and executed.

<Describe the various types of Testing performed for the Project. This will make sure the application is being tested properly thro testing types agreed as per Test Strategy>

6. Test Environment & Tools

<Provide details on Test Environment in which the Testing is carried out. Server, Database, Application URL etc. If any Tools were used like Quality Center (now HP ALM) for logging defects>

Application URL	http://abcd.2345.com
Apps Server	192.168.xxx.22
Database	Oracle 12g
HP QC/ALM	192.168.xxx.22

7. Lessons Learnt

<This section is used to describe the critical issues faced and their solutions (how they were solved during the Testing). Lessons learnt will help to make proactive decisions during the next Testing engagement, by avoiding these mistakes or finding a suitable workaround >

SI. No	Issues faced	Solutions
1	Somke testing test cases required to be executed manually each time	Smoke test cases were automated and the scripts were run, which ran fast and saved time
2	Initially, Few testers were not having rights to change defect status in HP QC/ALM. Test lead need to perform this task.	Rights were obtained from Client, by explaining the difficulty.

8. Recommendations

<Any workaround or suggestions can be mentioned here.>

- Admin control for defect management tool can be given to Offshore Test lead/manager for providing access to Testing team.
- Each time the onsite Admin need not be contacted for requests whenever they arise, thereby saving time due to the geographical time zone difference.

9.Best Practices

<There will be lot of activities done by the Testing team during the project. Some of them could have saved time, some proved to be a good & efficient way to work, etc. These can be documented as a 'Value Add' to show case to the Stakeholders.

Example: A repetitive task done manually every time was time consuming. This task was automated by creating scripts and run each time, which saved time and resources.

- Smoke test cases were automated and the scripts were run, which ran fast and saved time.
- Automation scripts were prepared to create new customers, where lot of records need to be created for Testing.
- Business critical scenarios are separately tested on the entire application which are vital to certify they works fine.

10. Exit Criteria

<Exit Criteria is defined as a Completion of Testing by fulfilling certain conditions>

a) All test cases should be executed – **Yes**
b) All defects in Critical, Major, Medium severity should be verified and closed – **Yes**.
c) Any open defects in trivial severity – **Action plan prepared with expected dates of closure.**

Example: No Severity1 defects should be 'OPEN'; Only 2 Severity2 defects should be 'OPEN'; Only 4 Severity3 defects should be 'OPEN'. Note: This may vary from project to project. Plan of Action for the Open defects should be clearly mentioned with details on when & how they will be addressed and closed.>

11. Conclusion/Sign Off

<This section will mention whether the Testing team agrees and gives a Green signal for the application to 'Go Live' or not, after the Exit Criteria was met. If the application does not meet the Exit Criteria, then it can be mentioned as – "The application is not suggested to 'Go Live'. In this scenario, It will be left with the decision of Senior Management and Client and other Stakeholders involved to take the call on whether the application can 'Go Live' or not.>

As the Exit criteria was met and satisfied as mentioned in Section 10, this application is suggested to 'Go Live' by the Testing team. Appropriate User/Business acceptance testing should be performed before 'Go Live'.

12. Definitions, Acronyms, and Abbreviations

<This section mentions the meanings of Abbreviated terms used in this document and any other new definitions>


SQL

What is SQL?

- SQL stands for **S**tructured **Q**uery **L**anguage
- SQL allows you to access a database
- SQL is an ANSI standard computer language
- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert new records in a database
- SQL can delete records from a database
- SQL can update records in a database
- SQL is easy to learn

SQL is a Standard - BUT....

SQL is an ANSI (American National Standards Institute) standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database. SQL works with database programs like MS Access, DB2, Informix, MS SQL Server, Oracle, Sybase, etc.

Unfortunately, there are many different versions of the SQL language, but to be in compliance with the ANSI standard, they must support the same major keywords in a similar manner (such as SELECT, UPDATE, DELETE, INSERT, WHERE, and others).

Note: Most of the SQL database programs also have their own proprietary extensions in addition to the SQL standard!

SQL Database Tables

A database most often contains one or more tables. Each table is identified by a name (e.g. "Customers" or "Orders"). Tables contain records (rows) with data.

Below is an example of a table called "Persons":

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

The table above contains three records (one for each person) and four columns (LastName, FirstName, Address, and City).

SQL Queries

With SQL, we can query a database and have a result set returned.

A query like this:

```
SELECT LastName FROM Persons
```

Gives a result set like this:

LastName
Hansen
Svendson
Pettersen

Note: Some database systems require a semicolon at the end of the SQL statement.

SQL Data Manipulation Language (DML)

SQL (Structured Query Language) is a syntax for executing queries. But the SQL language also includes a syntax to update, insert, and delete records.

These query and update commands together form the Data Manipulation Language (DML) part of SQL:

- **SELECT** - extracts data from a database table
- **UPDATE** - updates data in a database table
- **DELETE** - deletes data from a database table
- **INSERT INTO** - inserts new data into a database table

SQL Data Definition Language (DDL)

The Data Definition Language (DDL) part of SQL permits database tables to be created or deleted. We can also define indexes (keys), specify links between tables, and impose constraints between database tables.

The most important DDL statements in SQL are:

- **CREATE TABLE** - creates a new database table
- **ALTER TABLE** - alters (changes) a database table
- **DROP TABLE** - deletes a database table
- **CREATE INDEX** - creates an index (search key)
- **DROP INDEX** - deletes an index

SQL The SELECT Statement

The SELECT Statement

The SELECT statement is used to select data from a table. The tabular result is stored in a result table (called the result-set).

Syntax

```
SELECT column_name(s)
FROM table_name
```

Select Some Columns

To select the columns named "LastName" and "FirstName", use a SELECT statement like this:

```
SELECT LastName,FirstName FROM Persons
```

"Persons" table

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

Result

LastName	FirstName
Hansen	Ola
Svendson	Tove
Pettersen	Kari

Select All Columns

To select all columns from the "Persons" table, use a * symbol instead of column names, like this:

```
SELECT * FROM Persons
```

Result

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

The Result Set

The result from a SQL query is stored in a result-set. Most database software systems allow navigation of the result set with programming functions, like: Move-To-First-Record, Get-Record-Content, Move-To-Next-Record, etc.

Semicolon after SQL Statements?

Semicolon is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.

Some SQL tutorials end each SQL statement with a semicolon. Is this necessary? We are using MS Access and SQL Server 2000 and we do not have to put a semicolon after each SQL statement, but some database programs force you to use it.

The SELECT DISTINCT Statement

The DISTINCT keyword is used to return only distinct (different) values.

The SELECT statement returns information from table columns. But what if we only want to select distinct elements?

With SQL, all we need to do is to add a DISTINCT keyword to the SELECT statement:

Syntax

```
SELECT DISTINCT column_name(s)
FROM table_name
```

Using the DISTINCT keyword

To select ALL values from the column named "Company" we use a SELECT statement like this:

```
SELECT Company FROM Orders
```

"Orders" table

Company	OrderNumber
Sega	3412
W3Schools	2312
Trio	4678
W3Schools	6798

Result

Company
Sega
W3Schools
Trio
W3Schools

Note that "W3Schools" is listed twice in the result-set.
To select only DIFFERENT values from the column named "Company" we use a SELECT DISTINCT statement like this:

SELECT DISTINCT Company FROM Orders

Result:

Company
Sega
W3Schools
Trio

SQL The WHERE Clause

The WHERE clause is used to specify a selection criterion.

The WHERE Clause

To conditionally select data from a table, a WHERE clause can be added to the SELECT statement.

Syntax

```
SELECT column FROM table
WHERE column operator value
```

With the WHERE clause, the following operators can be used:

Operator	Description
=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern

Note: In some versions of SQL the <> operator may be written as !=

Using the WHERE Clause

To select only the persons living in the city "Sandnes", we add a WHERE clause to the SELECT statement:

```
SELECT * FROM Persons
WHERE City='Sandnes'
```

"Persons" table

LastName	FirstName	Address	City	Year
Hansen	Ola	Timoteivn 10	Sandnes	1951
Svendson	Tove	Borgvn 23	Sandnes	1978
Svendson	Stale	Kaivn 18	Sandnes	1980
Pettersen	Kari	Storgt 20	Stavanger	1960

Result

LastName	FirstName	Address	City	Year
Hansen	Ola	Timoteivn 10	Sandnes	1951
Svendson	Tove	Borgvn 23	Sandnes	1978
Svendson	Stale	Kaivn 18	Sandnes	1980

Using Quotes

Note that we have used single quotes around the conditional values in the examples.
SQL uses single quotes around text values (most database systems will also accept double quotes).
Numeric values should not be enclosed in quotes.
For text values:

```
This is correct:
SELECT * FROM Persons WHERE FirstName='Tove' This is
wrong:
SELECT * FROM Persons WHERE FirstName=Tove
```

For numeric values:

```
This is correct:
SELECT * FROM Persons WHERE Year>1965
This is wrong:
SELECT * FROM Persons WHERE Year>'1965'
```

The LIKE Condition

The LIKE condition is used to specify a search for a pattern in a column.

Syntax

```
SELECT column FROM table
WHERE column LIKE pattern
```

A "%" sign can be used to define wildcards (missing letters in the pattern) both before and after the pattern.

Using LIKE

The following SQL statement will return persons with first names that start with an 'O':

```
SELECT * FROM Persons
WHERE FirstName LIKE 'O%'
```

The following SQL statement will return persons with first names that end with an 'a':

```
SELECT * FROM Persons
WHERE FirstName LIKE '%a'
```

The following SQL statement will return persons with first names that contain the pattern 'la':

```
SELECT * FROM Persons
WHERE FirstName LIKE '%la%'
```

SQL The INSERT INTO Statement

The INSERT INTO Statement

The INSERT INTO statement is used to insert new rows into a table.

Syntax

```
INSERT INTO table_name
VALUES (value1, value2,....)
```

You can also specify the columns for which you want to insert data:

```
INSERT INTO table_name (column1, column2,...)
VALUES (value1, value2,....)
```

Insert a New Row

This "Persons" table:

LastName	FirstName	Address	City
Pettersen	Kari	Storgt 20	Stavanger

And this SQL statement:

```
INSERT INTO Persons
VALUES ('Hetland', 'Camilla', 'Hagabakka 24', 'Sandnes')
```

Will give this result:

LastName	FirstName	Address	City
Pettersen	Kari	Storgt 20	Stavanger
Hetland	Camilla	Hagabakka 24	Sandnes

Insert Data in Specified Columns

This "Persons" table:

LastName	FirstName	Address	City
Pettersen	Kari	Storgt 20	Stavanger
Hetland	Camilla	Hagabakka 24	Sandnes

And This SQL statement:

```
INSERT INTO Persons (LastName, Address)
VALUES ('Rasmussen', 'Storgt 67')
```

Will give this result:

LastName	FirstName	Address	City
Pettersen	Kari	Storgt 20	Stavanger
Hetland	Camilla	Hagabakka 24	Sandnes
Rasmussen		Storgt 67	

SQL The UPDATE Statement

The Update Statement

The UPDATE statement is used to modify the data in a table.

Syntax

```
UPDATE table_name
SET column_name = new_value
WHERE column_name = some_value
```

Person:

LastName	FirstName	Address	City
Nilsen	Fred	Kirkegt 56	Stavanger
Rasmussen		Storgt 67	

Update one Column in a Row

We want to add a first name to the person with a last name of "Rasmussen":

```
UPDATE Person SET FirstName = 'Nina'
WHERE LastName = 'Rasmussen'
```

Result:

LastName	FirstName	Address	City
Nilsen	Fred	Kirkegt 56	Stavanger
Rasmussen	Nina	Storgt 67	

Update several Columns in a Row

We want to change the address and add the name of the city:

```
UPDATE Person
SET Address = 'Stien 12', City = 'Stavanger'
WHERE LastName = 'Rasmussen'
```

Result:

LastName	FirstName	Address	City
Nilsen	Fred	Kirkegt 56	Stavanger
Rasmussen	Nina	Stien 12	Stavanger

SQL The Delete Statement

The Delete Statement

The DELETE statement is used to delete rows in a table.

Syntax

```
DELETE FROM table_name
WHERE column_name = some_value
```

Person:

LastName	FirstName	Address	City
Nilsen	Fred	Kirkegt 56	Stavanger
Rasmussen	Nina	Stien 12	Stavanger

Delete a Row

"Nina Rasmussen" is going to be deleted:

DELETE FROM Person WHERE LastName = 'Rasmussen'

Result

LastName	FirstName	Address	City
Nilsen	Fred	Kirkegt 56	Stavanger

Delete All Rows

It is possible to delete all rows in a table without deleting the table. This means that the table structure, attributes, and indexes will be intact:

```
DELETE FROM table_name
or
DELETE * FROM table_name
```

SQL ORDER BY

The ORDER BY keyword is used to sort the result.

Sort the Rows

The ORDER BY clause is used to sort the rows.

Orders:

Company	OrderNumber
Sega	3412
ABC Shop	5678
W3Schools	2312
W3Schools	6798

Example

To display the companies in alphabetical order:

```
SELECT Company, OrderNumber FROM Orders
ORDER BY Company
```

Result:

Company	OrderNumber
ABC Shop	5678
Sega	3412
W3Schools	6798
W3Schools	2312

Example

To display the companies in alphabetical order AND the ordernumbers in numerical order:

```
SELECT Company, OrderNumber FROM Orders
ORDER BY Company, OrderNumber
```

Result:

Company	OrderNumber
ABC Shop	5678
Sega	3412
W3Schools	2312
W3Schools	6798

Example

To display the companies in reverse alphabetical order:

```
SELECT Company, OrderNumber FROM Orders
ORDER BY Company DESC
```

Result:

Company	OrderNumber
W3Schools	6798
W3Schools	2312
Sega	3412
ABC Shop	5678

Example

To display the companies in reverse alphabetical order AND the ordernumbers in numerical order:

SELECT Company, OrderNumber FROM Orders
ORDER BY Company DESC, OrderNumber ASC

Result:

Company	OrderNumber
W3Schools	2312
W3Schools	6798
Sega	3412
ABC Shop	5678

SQL AND & OR

AND & OR

AND and OR join two or more conditions in a WHERE clause.
The AND operator displays a row if ALL conditions listed are true. The OR operator displays a row if ANY of the conditions listed are true.

Original Table (used in the examples)

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Svendson	Stephen	Kaivn 18	Sandnes

Example

Use AND to display each person with the first name equal to "Tove", and the last name equal to "Svendson":

```
SELECT * FROM Persons
WHERE FirstName='Tove'
AND LastName='Svendson'
```

Result:

LastName	FirstName	Address	City
Svendson	Tove	Borgvn 23	Sandnes

Example

Use OR to display each person with the first name equal to "Tove", or the last name equal to "Svendson":

```
SELECT * FROM Persons
WHERE firstname='Tove'
OR lastname='Svendson'
```

Result:

LastName	FirstName	Address	City
Svendson	Tove	Borgvn 23	Sandnes
Svendson	Stephen	Kaivn 18	Sandnes

Example

You can also combine AND and OR (use parentheses to form complex expressions):

```
SELECT * FROM Persons WHERE (FirstName='Tove' OR
FirstName='Stephen') AND LastName='Svendson'
```

Result:

LastName	FirstName	Address	City
Svendson	Tove	Borgvn 23	Sandnes
Svendson	Stephen	Kaivn 18	Sandnes

SQL IN

IN

The IN operator may be used if you know the exact value you want to return for at least one of the columns.

```
SELECT column_name FROM table_name
WHERE column_name IN (value1,value2,..)
```

Original Table (used in the examples)

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Nordmann	Anna	Neset 18	Sandnes
Pettersen	Kari	Storgt 20	Stavanger
Svendson	Tove	Borgvn 23	Sandnes

Example 1

To display the persons with LastName equal to "Hansen" or "Pettersen", use the following SQL:

```
SELECT * FROM Persons
WHERE LastName IN ('Hansen','Pettersen')
```

Result:

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

SQL BETWEEN

BETWEEN ... AND

The BETWEEN ... AND operator selects a range of data between two values. These values can be numbers, text, or dates.

```
SELECT column_name FROM table_name
WHERE column_name
BETWEEN value1 AND value2
```

Original Table (used in the examples)

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Nordmann	Anna	Neset 18	Sandnes
Pettersen	Kari	Storgt 20	Stavanger
Svendson	Tove	Borgvn 23	Sandnes

Example 1

To display the persons alphabetically between (and including) "Hansen" and exclusive "Pettersen", use the following SQL:

```
SELECT * FROM Persons WHERE LastName
BETWEEN 'Hansen' AND 'Pettersen'
```

Result:

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Nordmann	Anna	Neset 18	Sandnes

IMPORTANT! The BETWEEN...AND operator is treated differently in different databases. With some databases a person with the LastName of "Hansen" or "Pettersen" will not be listed (BETWEEN..AND only selects fields that are between and excluding the test values). With some databases a person with the last name of "Hansen" or "Pettersen" will be listed (BETWEEN..AND selects fields that are between and including the test values). With other databases a person with the last name of "Hansen" will be listed, but "Pettersen" will not be listed (BETWEEN..AND selects fields between the test values, including the first test value and excluding the last test value). Therefore: Check how your database treats the BETWEEN....AND operator!

Example 2

To display the persons outside the range used in the previous example, use the NOT operator:

```
SELECT * FROM Persons WHERE LastName
NOT BETWEEN 'Hansen' AND 'Pettersen'
```

Result:

LastName	FirstName	Address	City
Pettersen	Kari	Storgt 20	Stavanger
Svendson	Tove	Borgvn 23	Sandnes

SQL Aliases

With SQL, aliases can be used for column names and table names.

Column Name Alias

The syntax is:

SELECT column AS column_alias FROM table

Table Name Alias

The syntax is:

SELECT column FROM table AS table_alias

Example: Using a Column Alias

This table (Persons):

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

And this SQL:

SELECT LastName AS Family, FirstName AS Name FROM

Persons

Returns this result:

Family	Name
Hansen	Ola
Svendson	Tove
Pettersen	Kari

Example: Using a Table Alias

This table (Persons):

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

And this SQL:

```
SELECT LastName, FirstName
FROM Persons AS Employees
```

Returns this result:
Table Employees:

LastName	FirstName
Hansen	Ola
Svendson	Tove
Pettersen	Kari

SQL Join

Joins and Keys

Sometimes we have to select data from two or more tables to make our result complete. We have to perform a join.
Tables in a database can be related to each other with keys. A primary key is a column with a unique value for each row. The purpose is to bind data together, across tables, without repeating all of the data in every table.
In the "Employees" table below, the "Employee_ID" column is the primary key, meaning that **no** two rows can have the same Employee_ID. The Employee_ID distinguishes two persons even if they have the same name.
When you look at the example tables below, notice that:

- The "Employee_ID" column is the primary key of the "Employees" table
- The "Prod_ID" column is the primary key of the "Orders" table
- The "Employee_ID" column in the "Orders" table is used to refer to the persons in the "Employees" table without using their names

Employees:

Employee_ID	Name
01	Hansen, Ola
02	Svendson, Tove
03	Svendson, Stephen
04	Pettersen, Kari

Orders:

Prod_ID	Product	Employee_ID
234	Printer	01
657	Table	03
865	Chair	03

Referring to Two Tables

We can select data from two tables by referring to two tables, like this:

Example

Who has ordered a product, and what did they order?

```
SELECT Employees.Name, Orders.Product
FROM Employees, Orders
WHERE Employees.Employee_ID=Orders.Employee_ID
```

Result

Name	Product
Hansen, Ola	Printer
Svendson, Stephen	Table
Svendson, Stephen	Chair

Example

Who ordered a printer?

```
SELECT Employees.Name
FROM Employees, Orders
WHERE Employees.Employee_ID=Orders.Employee_ID
AND Orders.Product='Printer'
```

Result

Name
Hansen, Ola

Using Joins

OR we can select data from two tables with the JOIN keyword, like this:

Example INNER JOIN

Syntax

```
SELECT field1, field2, field3
FROM first_table
INNER JOIN second_table
ON first_table.keyfield = second_table.foreign_keyfield
```

Who has ordered a product, and what did they order?

```
SELECT Employees.Name, Orders.Product
FROM Employees
INNER JOIN Orders
ON Employees.Employee_ID=Orders.Employee_ID
```

The INNER JOIN returns all rows from both tables where there is a match. If there are rows in Employees that do not have matches in Orders, those rows will **not** be listed.

Result

Name	Product
Hansen, Ola	Printer
Svendson, Stephen	Table
Svendson, Stephen	Chair

Example LEFT JOIN
Syntax

```
SELECT field1, field2, field3
FROM first_table
LEFT JOIN second_table
ON first_table.keyfield = second_table.foreign_keyfield
```

List all employees, and their orders - if any.

```
SELECT Employees.Name, Orders.Product
FROM Employees
LEFT JOIN Orders
ON Employees.Employee_ID=Orders.Employee_ID
```

The LEFT JOIN returns all the rows from the first table (Employees), even if there are no matches in the second table (Orders). If there are rows in Employees that do not have matches in Orders, those rows **also** will be listed.
Result

Name	Product
Hansen, Ola	Printer
Svendson, Tove	
Svendson, Stephen	Table
Svendson, Stephen	Chair
Pettersen, Kari	

Example RIGHT JOIN
Syntax

```
SELECT field1, field2, field3
FROM first_table
RIGHT JOIN second_table
ON first_table.keyfield = second_table.foreign_keyfield
```

List all orders, and who has ordered - if any.

```
SELECT Employees.Name, Orders.Product
FROM Employees
RIGHT JOIN Orders
ON Employees.Employee_ID=Orders.Employee_ID
```

The RIGHT JOIN returns all the rows from the second table (Orders), even if there are no matches in the first table (Employees). If there had been any rows in Orders that did not have matches in Employees, those rows **also** would have been listed.
Result

Name	Product
Hansen, Ola	Printer
Svendson, Stephen	Table
Svendson, Stephen	Chair

Who ordered a printer?

```
SELECT Employees.Name
FROM Employees
INNER JOIN Orders
ON Employees.Employee_ID=Orders.Employee_ID
WHERE Orders.Product = 'Printer'
```

Result

Name
Hansen, Ola

SQL UNION and UNION ALL

UNION

The UNION command is used to select related information from two tables, much like the JOIN command. However, when using the UNION command all selected columns need to be of the same data type.

Note: With UNION, only distinct values are selected.

```
SQL Statement 1
UNION
SQL Statement 2
```

Employees_Norway:

Employee_ID	E_Name
01	Hansen, Ola
02	Svendson, Tove
03	Svendson, Stephen
04	Pettersen, Kari

Employes_USA:

Employee_ID	E_Name
01	Turner, Sally
02	Kent, Clark
03	Svendson, Stephen
04	Scott, Stephen

Using the UNION Command

Example

List all different employee names in Norway and USA:

```
SELECT E_Name FROM Employees_Norway
UNION
SELECT E_Name FROM Employees_USA
```

Result

Name
Hansen, Ola
Svendson, Tove
Svendson, Stephen
Pettersen, Kari
Turner, Sally
Kent, Clark
Scott, Stephen

Note: This command cannot be used to list all employees in Norway and USA. In the example above we have two employees with equal names, and only one of them is listed. The UNION command only selects distinct values.

UNION ALL

The UNION ALL command is equal to the UNION command, except that UNION ALL selects all values.

```
SQL Statement 1
UNION ALL
SQL Statement 2
```

Using the UNION ALL Command

Example

List all employees in Norway and USA:

```
SELECT E_Name FROM Employees_Norway
UNION ALL
SELECT E_Name FROM Employees_USA
```

Result

Name
Hansen, Ola
Svendson, Tove
Svendson, Stephen
Pettersen, Kari
Turner, Sally
Kent, Clark
Svendson, Stephen
Scott, Stephen

SQL Create Database, Table, and Index

Create a Database

To create a database:

CREATE DATABASE database_name

Create a Table

To create a table in a database:

```
CREATE TABLE table_name
(
column_name1 data_type,
column_name2 data_type,
.....
)
```

Example

This example demonstrates how you can create a table named "Person", with four columns. The column names will be "LastName", "FirstName", "Address", and "Age":

```
CREATE TABLE Person
(
LastName varchar,
FirstName varchar,
Address varchar,
Age int
)
```

This example demonstrates how you can specify a maximum length for some columns:

```
CREATE TABLE Person
(
LastName varchar(30),
FirstName varchar,
Address varchar,
Age int(3)
)
```

The data type specifies what type of data the column can hold. The table below contains the most common data types in SQL:

Data Type	Description
integer(size) int(size) smallint(size) tinyint(size)	Hold integers only. The maximum number of digits are specified in parenthesis.
decimal(size,d) numeric(size,d)	Hold numbers with fractions. The maximum number of digits are specified in "size". The maximum number of digits to the right of the decimal is specified in "d".
char(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis.
varchar(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis.

Create Index

Indices are created in an existing table to locate rows more quickly and efficiently. It is possible to create an index on one or more columns of a table, and each index is given a name. The users cannot see the indexes, they are just used to speed up queries.

Note: Updating a table containing indexes takes more time than updating a table without, this is because the indexes also need an update. So, it is a good idea to create indexes only on columns that are often used for a search.

A Unique Index

Creates a unique index on a table. A unique index means that two rows cannot have the same index value.

```
CREATE UNIQUE INDEX index_name
ON table_name (column_name)
```

The "column_name" specifies the column you want indexed.

A Simple Index

Creates a simple index on a table. When the UNIQUE keyword is omitted, duplicate values are allowed.

```
CREATE INDEX index_name
ON table_name (column_name)
```

The "column_name" specifies the column you want indexed.

Example

This example creates a simple index, named "PersonIndex", on the LastName field of the Person table:

```
CREATE INDEX PersonIndex
ON Person (LastName)
```

If you want to index the values in a column in **descending** order, you can add the reserved word **DESC** after the column name:

```
CREATE INDEX PersonIndex
ON Person (LastName DESC)
```

If you want to index more than one column you can list the column names within the parentheses, separated by commas:

```
CREATE INDEX PersonIndex
ON Person (LastName, FirstName)
```

SQL Drop Index, Table and Database

Drop Index

You can delete an existing index in a table with the DROP statement.

ROP INDEX table_name.index_name

Delete a Table or Database

To delete a table (the table structure, attributes, and indexes will also be deleted):

DROP TABLE table_name

To delete a database:

DROP DATABASE database_name

Truncate a Table

What if we only want to get rid of the data inside a table, and not the table itself? Use the TRUNCATE TABLE command (deletes only the data inside the table):

TRUNCATE TABLE table_name

SQL ALTER TABLE

ALTER TABLE

The ALTER TABLE statement is used to add or drop columns in an existing table.

```
ALTER TABLE table_name
ADD column_name datatype
ALTER TABLE table_name
DROP COLUMN column_name
```

Note: Some database systems don't allow the dropping of a column in a database table (DROP COLUMN column_name).

Person:

LastName	FirstName	Address
Pettersen	Kari	Storgt 20

Example
To add a column named "City" in the "Person" table:

ALTER TABLE Person ADD City varchar(30)

Result:

LastName	FirstName	Address	City
Pettersen	Kari	Storgt 20	

Example
To drop the "Address" column in the "Person" table:

ALTER TABLE Person DROP COLUMN Address

Result:

LastName	FirstName	City
Pettersen	Kari	

SQL Functions
SQL has a lot of built-in functions for counting and calculations.

Function Syntax
The syntax for built-in SQL functions is:

SELECT function(column) FROM table

Types of Functions
There are several basic types and categories of functions in SQL. The basic types of functions are:

- Aggregate Functions
- Scalar functions

Aggregate functions
Aggregate functions operate against a collection of values, but return a single value.

Note: If used among many other expressions in the item list of a SELECT statement, the SELECT must have a GROUP BY clause!!

"Persons" table (used in most examples)

Name	Age
Hansen, Ola	34
Svendson, Tove	45
Pettersen, Kari	19

Aggregate functions in MS Access

Function	Description
AVG(column)	Returns the average value of a column
COUNT(column)	Returns the number of rows (without a NULL value) of a column
COUNT(*)	Returns the number of selected rows
FIRST(column)	Returns the value of the first record in a specified field
LAST(column)	Returns the value of the last record in a specified field
MAX(column)	Returns the highest value of a column
MIN(column)	Returns the lowest value of a column
STDEV(column)	
STDEVP(column)	
SUM(column)	Returns the total sum of a column
VAR(column)	
VARP(column)	

Aggregate functions in SQL Server

Function	Description
AVG(column)	Returns the average value of a column
BINARY_CHECKSUM	
CHECKSUM	
CHECKSUM_AGG	

COUNT(column)	Returns the number of rows (without a NULL value) of a column
COUNT(*)	Returns the number of selected rows
COUNT(DISTINCT column)	Returns the number of distinct results
FIRST(column)	Returns the value of the first record in a specified field (not supported in SQLServer2K)
LAST(column)	Returns the value of the last record in a specified field (not supported in SQLServer2K)
MAX(column)	Returns the highest value of a column
MIN(column)	Returns the lowest value of a column
STDEV(column)	
STDEVP(column)	
SUM(column)	Returns the total sum of a column
VAR(column)	
VARP(column)	

Scalar functions

Scalar functions operate against a single value, and return a single value based on the input value.

Useful Scalar Functions in MS Access

Function	Description
UCASE(c)	Converts a field to upper case
LCASE(c)	Converts a field to lower case
MID(c,start[,end])	Extract characters from a text field
LEN(c)	Returns the length of a text field
INSTR(c)	Returns the numeric position of a named character within a text field
LEFT(c,number_of_char)	Return the left part of a text field requested
RIGHT(c,number_of_char)	Return the right part of a text field requested
ROUND(c,decimals)	Rounds a numeric field to the number of decimals specified
MOD(x,y)	Returns the remainder of a division operation
NOW()	Returns the current system date
FORMAT(c,format)	Changes the way a field is displayed
DATEDIFF(d,date1,date2)	Used to perform date calculations

SQL GROUP BY and HAVING

Aggregate functions (like SUM) often need an added GROUP BY functionality.

GROUP BY...

GROUP BY... was added to SQL because aggregate functions (like SUM) return the aggregate of all column values every time they are called, and without the GROUP BY function it was impossible to find the sum for each individual group of column values.

The syntax for the GROUP BY function is:

SELECT column,SUM(column) FROM table GROUP BY column

GROUP BY Example

This "Sales" Table:

Company	Amount
W3Schools	5500
IBM	4500
W3Schools	7100

And This SQL:

SELECT Company, SUM(Amount) FROM Sales

Returns this result:

Company	SUM(Amount)
W3Schools	17100
IBM	17100
W3Schools	17100

The above code is invalid because the column returned is not part of an aggregate. A GROUP BY clause will solve this problem:

```
SELECT Company,SUM(Amount) FROM Sales
GROUP BY Company
```

Returns this result:

Company	SUM(Amount)
W3Schools	12600
IBM	4500

HAVING...

HAVING... was added to SQL because the WHERE keyword could not be used against aggregate functions (like SUM), and without HAVING... it would be impossible to test for result conditions.

The syntax for the HAVING function is:

```
SELECT column,SUM(column) FROM table
GROUP BY column
HAVING SUM(column) condition value
```

This "Sales" Table:

Company	Amount
W3Schools	5500
IBM	4500
W3Schools	7100

This SQL:

```
SELECT Company,SUM(Amount) FROM Sales
GROUP BY Company
HAVING SUM(Amount)>10000
```

Returns this result

Company	SUM(Amount)
W3Schools	12600

SQL The SELECT INTO Statement

The SELECT INTO Statement

The SELECT INTO statement is most often used to create backup copies of tables or for archiving records.

Syntax

```
SELECT column_name(s) INTO newtable [IN externaldatabase]
FROM source
```

Make a Backup Copy

The following example makes a backup copy of the "Persons" table:

```
SELECT * INTO Persons_backup
FROM Persons
```

The IN clause can be used to copy tables into another database:

SELECT Persons.* INTO Persons IN 'Backup.mdb' FROM

Persons

If you only want to copy a few fields, you can do so by listing them after the SELECT statement:

SELECT LastName,FirstName INTO Persons_backup FROM

Persons

You can also add a WHERE clause. The following example creates a "Persons_backup" table with two columns (FirstName and LastName) by extracting the persons who lives in "Sandnes" from the "Persons" table:

```
SELECT LastName,Firstname INTO Persons_backup
FROM Persons
WHERE City='Sandnes'
```

Selecting data from more than one table is also possible. The following example creates a new table "Empl_Ord_backup" that contains data from the two tables Employees and Orders:

```
SELECT Employees.Name,Orders.Product
INTO Empl_Ord_backup
FROM Employees
INNER JOIN Orders
ON Employees.Employee_ID=Orders.Employee_ID
```

SQL The CREATE VIEW Statement

A view is a virtual table based on the result-set of a SELECT statement.

What is a View?

In SQL, a VIEW is a virtual table based on the result-set of a SELECT statement.
A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database. You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from a single table.
Note: The database design and structure will NOT be affected by the functions, where, or join statements in a view.

Syntax

```
CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition
```

Note: The database does not store the view data! The database engine recreates the data, using the view's SELECT statement, every time a user queries a view.

Using Views

A view could be used from inside a query, a stored procedure, or from inside another view. By adding functions, joins, etc., to a view, it allows you to present exactly the data you want to the user.
The sample database Northwind has some views installed by default. The view "Current Product List" lists all active products (products that are not discontinued) from the Products table. The view is created with the following SQL:

```
CREATE VIEW [Current Product List] AS
SELECT ProductID,ProductName
FROM Products
WHERE Discontinued=No
```

We can query the view above as follows:

```
SELECT * FROM [Current Product List]
```

Another view from the Northwind sample database selects every product in the Products table that has a unit price that is higher than the average unit price:

```
CREATE VIEW [Products Above Average Price] AS
SELECT ProductName,UnitPrice
FROM Products
WHERE UnitPrice>(SELECT AVG(UnitPrice) FROM Products)
```

We can query the view above as follows:

```
SELECT * FROM [Products Above Average Price]
```

Another example view from the Northwind database calculates the total sale for each category in 1997.
Note that this view select its data from another view called "Product Sales for 1997":

```
CREATE VIEW [Category Sales For 1997] AS
SELECT DISTINCT CategoryName,Sum(ProductSales) AS CategorySales
FROM [Product Sales for 1997]
GROUP BY CategoryName
```

We can query the view above as follows:

```
SELECT * FROM [Category Sales For 1997]
```

We can also add a condition to the query. Now we want to see the total sale only for the category "Beverages":

```
SELECT * FROM [Category Sales For 1997]
WHERE CategoryName='Beverages'
```