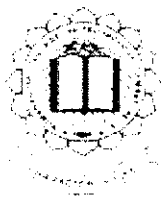
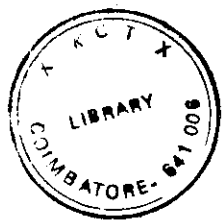




A STUDY ON TESTING AND ANALYSIS OF PROJECT TRACKING SYSTEM OF VEE EEE TECHNOLOGIES, CHENNAI

by

B. SARANYA
Reg.No.71206631046

of

Department of Management Studies
Kumaraguru College of Technology
Coimbatore

A PROJECT REPORT

submitted to the

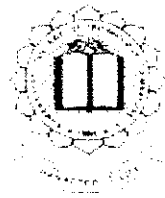
FACULTY OF MANAGEMENT STUDIES

in partial fulfillment of the requirements
for the award of the degree

of

MASTER OF BUSINESS ADMINISTRATION

July, 2008



DEPARTMENT OF MANAGEMENT STUDIES

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE

BONAFIDE CERTIFICATE

Certified that this project work titled **"A STUDY ON TESTING AND ANALYSIS OF PROJECT TRACKING SYSTEM OF VEE EEE TECHNOLOGIES, CHENNAI"** is the bonafide work of **Ms. B. SARANYA (71206631046)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Faculty Guide

Director

Evaluated and vice-voce conducted on 2-7-2008

Examiner I

Examiner II

Declaration

DECLARATION

I, hereby declare that this project work entitled as **A STUDY ON TESTING AND ANALYSIS OF PROJECT TRACKING SYSTEM OF VEE EEE TECHNOLOGIES, CHENNAI**, has been undertaken for academic purpose submitted to Anna University, Chennai in partial fulfillment of the requirements for the award of the degree of Master of Business Administration. The project report is the record of the original work done by me under the guidance of Mr. S. Mohanavel, Senior Lecturer/MBA during the academic year 2007-2008.

I, also declare hereby, that the information given in this report is correct to the best of my knowledge and belief.

Place: Coimbatore

Date: 2-7-2008

B. Saranya

(B. SARANYA)

Acknowledgement

ACKNOWLEDGEMENT

It is inevitable that thoughts and ideas of other people tend to drift into the subconscious when one feels to acknowledge helping derived from others. I acknowledge to all those who have helped me in the preparation of this project work.

I would like to thank the God almighty for his guidance without whom this project wouldn't have become reality.

I wish to express my deep gratitude to our Principal **Dr. Joseph V. Thanikal** for the facilities provided to complete my project work. I wish to express my sincere thanks to **Dr. S. V. Devanathan,** Director, KCT Business School, for his continuous encouragement throughout my project.

I owe my heartfelt gratitude to **Mr. S. Mohanavel,** Senior Lecturer, KCT Business School, for his help and valuable guidance given to me through out my project.

I express my sincere thanks to **Mr. R. Ramakrishnan,** Team Lead and **Mr. S. Benson Edwinraj,** Project Manager, Vee Ece Technologies, Coimbatore for their valuable guidance to complete my project successfully.

Lastly I thank all the employees of the organization who were involved knowingly or unknowingly to make this project work successful.

Vee Eee Technologies

Applying mind in Technologies

Date: 04/05/'08

This is to certify that **Ms. B. Saranya (06MBA46) MBA IInd** year student of **KCT Business School, Kumaraguru College of Technology,** had undergone a Project between **January 19, 2008** to **April 15, 2008** titled **"A Study on Testing and Analysis of Project Tracking System of Vee Eee Technologies, Chennai".**

During this period her performance was **Good.**

Name and Designation

of Organisational Guide

R. Ramakrishnan

Team Lead

4/5/'08

Signature of the of Organisational Guide

Vee Eee Technologies COIMBATORE 98405 77448 www.veetechnologies.com

Executive Summary

EXECUTIVE SUMMARY

Testing is a process used to identify the correctness, completeness and quality of developed computer software. One definition of testing is "the process of questioning a product in order to evaluate it", where the "questions" are things the tester tries to do with the product, and the product answers with its behavior in reaction to the probing of the tester.

In the study, efforts have been made to conduct a detailed analysis of testing of Project Tracking System of Vee Eee Technologies, Chennai. The main objective of the project is to study and analyze various test types on Project Tracking system and to give suggestions to the organization.

Data used for the study is primary in nature as generation of test cases is done manually. The number of test cases generated is 874, where the total number of unit test cases is 601, GUI test cases are 222 and usability test cases are 51.The major tool used to analyze the data collected is MS-Excel.

It is found that the company must concentrate more on unit testing than GUI and usability testing. Unit testing has many fail test cases. This affects the success of the developed product. The programming team must carefully correct the defects to avoid such errors in future.

Table of contents

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Introduction

CHAPTER - 1

INTRODUCTION

1.1. BACKGROUND:

Testing is a process used to identify the correctness, completeness and quality of developed computer software. One definition of testing is "the process of questioning a product in order to evaluate it", where the "questions" are things the tester tries to do with the product, and the product answers with its behavior in reaction to the probing of the tester.

Testing objectives include

1. Testing is a process of executing a program with the intent of finding an error.
2. A good test case is one that has a high probability of finding an as yet undiscovered error.
3. A successful test is one that uncovers an as yet undiscovered error.

Software Development Life Cycle (SDLC) is a methodology that is typically used to develop, maintain and replace information systems for improving the quality of the software design and development process. The typical phases are analysis, estimation, design, development, integration and testing and implementation. The success of software largely depends on proper analysis, estimation, design and testing before the same is implemented.

Testing cannot show the absence of defects, it can only show that software defects are present. The objective is to design tests that systematically uncover different classes of errors and do so with a minimum amount of time and effort.

The benefits include that software functions appear to be working according to specification, performance requirements appear to have been met and data collected during

testing provides a good indication of software reliability and some indication of software quality.

The testing principles are: all tests should be traceable to customer requirements; test should be planned long before testing begins; testing should begin "in the small" and progress towards testing "in the large" and testing should be conducted by an independent third party.

1.1.1. Software Testing Life Cycle:

The software test development life cycle contains the following components:

- ❖ Requirements
- ❖ Use Case Document
- ❖ Test Plan
- ❖ Test Case
- ❖ Test Case Execution
- ❖ Report Analysis
- ❖ Bug Analysis
- ❖ Bug Reporting

Typical interaction scenario from a user's perspective for system requirements studies are testing. In other words, "an actual or realistic example scenario". A use case describes the use of a system from start to finish. Use cases focus attention on aspects of a system useful to people outside of the system also.

- Users of a program are called users or clients.
- Users of an enterprise are called customers, suppliers, etc.

1.1.2. Test Documentation:

Test documentation is a report for managing and maintaining the testing process. Documents produced by testers should answer the following questions:

- What to test? Test Plan
- How to test? Test Specification
- What are the results? Test Results Analysis Report

1.1.3. Software Bug Life cycle:

The software problems are called as bugs. The software bug life cycle is used to identify the stages of its life. Following figure shows an example of the simplest and most optimal software bug life cycle.

Figure 1.1 shows that when a bug is found by a Software Tester, it is logged and assigned to a programmer to be fixed. This state is called open state. Once the programmer fixes the code, he assigns it back to the tester and the bugs enter the resolved state. The tester then performs a regression test to confirm that the bug is indeed fixed and closes. The bug then enters its final state, the closed state.

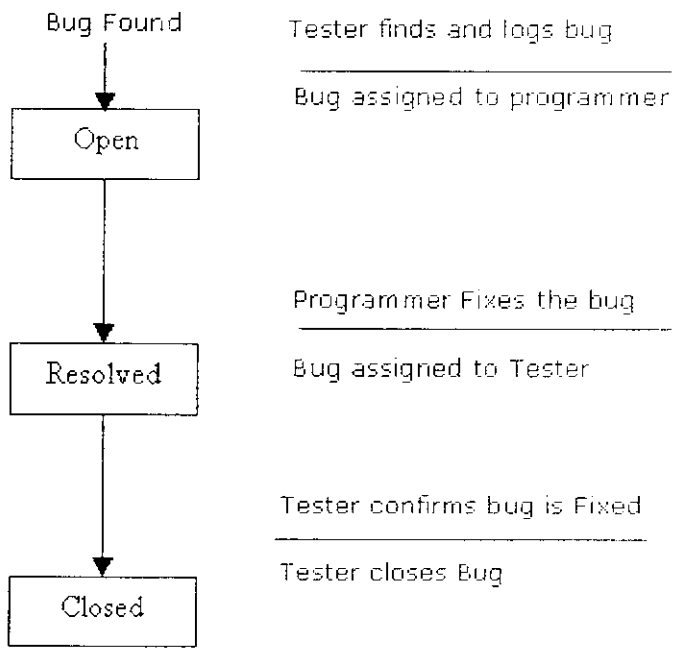


Figure 1.1 Software Bug Lifecycle

1.1.4. Bug Report:

The bug report is prepared for the following purpose:

- Communicate bug for reproducibility, resolution, and regression.
- Track bug status (open, resolved, closed).
- Ensure bug is not forgotten, lost or ignored.

1.1.5. Testing Tools:

➢ **WinRunner:**

WinRunner, is Mercury's interactive enterprise functional testing tool. It is used to quickly create and run sophisticated automated tests on different applications. Winrunner helps to automate the testing process, from test development to execution. It creates adaptable and reusable test scripts that challenge the functionality of application. Prior to a software release, it can run these tests in a single overnight run- enabling to detect and ensure superior software quality.

➢ **Load Runner:**

Load Runner is divided up into three smaller applications. The Virtual User Generator creates scripts that generate a series of actions, such as logging on, navigating through the application, and exiting the program. The Controller takes the scripts that has been made and runs them through a schedule. The Controller is instructed how many users to activate, when to activate them, and how to group the users and keep track of them. The Results and Analysis program gives all the results of the load test in various forms. It allows seeing summaries of data, as well as the details of the load test for pinpointing problems or bottlenecks.

➢ **Test Director:**

Test Director, the industry's first global test management solution, helps organizations deploy high-quality applications more quickly and effectively. Its four modules Requirements, Test Plan, Test Lab, and Defects are seamlessly integrated, allowing for a smooth information

flow between various testing stages. The completely Web-enabled Test Director supports high levels of communication and collaboration among distributed testing teams, driving a more effective, efficient global application-testing process.

- **Silk Test:**

Silk Test is a tool specifically designed for doing regression and functionality testing. It is developed by Segue Software Incorporation. Silk Test is the industry's leading functional testing product for e-business applications, whether Windows based, Web, Java, or traditional client/server-based. Silk Test also offers test planning, management, direct database access and validation, a built in recovery system for unattended testing, and the ability to test across multiple platforms, browsers and technologies.

- **Test Partner:**

The Test Partner Active Data Test Creation wizard provides a non-programmatic way to create data-driven tests. Such tests are useful for testing form-based applications. Using the wizard, test can be recorded and fields can be selected which is to be included in data file using a data table.

- **Rational Software:**

IBM Rational AppScan Tester Edition integrates security into quality assurance testing for improved Web app development. It helps QA teams find and remediate security vulnerabilities while testing for functional and performance issues earlier in the application delivery process.

- Expanded test coverage beyond functional and performance testing
- A single console, metrics and environment for QA teams to manage security and quality testing of Web applications
- Automated security testing and execution as part of the normal QA run, eliminating the need for QA personnel to learn advanced or new security tools

- Supports ability to log security issues directly into your defect tracking tool
- Delivers real-time training on secure testing and coding techniques
- Operating system supported: Windows

1.2. REVIEW OF LITERATURES:

Dhavachelvan. P, Uma. G.V. (2005)[1]: Bringing together agents and other fields of software engineering might be difficult, as the advantages of agent technology are still not widely recognized. Effectiveness claims of agent-oriented software engineering are based upon the strategies for addressing complex systems. Agent technologies facilitate the automated **software testing** by virtue of their high-level decomposition, independency and parallel activation. The informal interpretations of qualitative agent theories are not sufficient to distinguish agent-based approaches from other approaches in software testing.

Itkonen, Juha, Mantyla, Mika V. Lassenius, Casper (2007)[2]: While traditional testing literature emphasizes test cases, Exploratory Testing (ET) stresses the individual tester's skills during test execution and does not rely upon predesigned test cases. In the experiment, 79 advanced software engineering students performed manual functional testing on an open-source application with actual and seeded defects. Each student participated in two 90-minute controlled sessions, using ET in one and Test Case based Testing (TCT) in the other. No significant differences in defect detection efficiency between TCT and ET had been found. The distributions of detected defects did not differ significantly regarding technical type, detection difficulty, or severity. However, TCT produced significantly more false defect reports than ET. The results show no benefit of using predesigned test cases in terms of defect detection efficiency, emphasizing the need for further studies of **manual testing**.

---

[1] Dhavachelvan. P, Uma. G.V, Dept. of Computer Science & Engineering., Anna Univ., Chennai, India. 'COMPLEXITY MEASURES FOR SOFTWARE SYSTEMS: TOWARDS MULTI-AGENT BASED SOFTWARE TESTING', Intelligent Sensing and Information Processing, Proceedings of International Conference on 4-7 Jan. 2005, pp.359-364.

[2] Itkonen, Juha, Mantyla, Mika V. Lassenius, Casper, Helsinki University of Technology, Finland, 'DEFECT DETECTION EFFICIENCY: TEST CASE BASED VS. EXPLORATORY TESTING', Empirical Software Engineering and Measurement, First International Symposium on 20-21 Sept. 2007, Madrid, Spain, pp. 61-70.

Kaufman, L.M. Johnson, B.W. Dugan, J.B. (2002)[3]**:** The existing classes of fault coverage models require a priori distribution for collected data in their analysis. Using these models, analyses can be performed using various assumed distributions. The assumed distributions may not accurately reflect the behavior of the collected data and, as a result, the coverage values predicted by the models may be inaccurate, especially if testing yields little or no failure data. Since the occurrence of an uncovered fault in an ultra-dependable system is a rare event, then statistics of the extremes can be used to quantify uncoverage estimates in such systems. Statistics of the extremes provides for an analysis of rare event data without requiring any a priori knowledge of its distribution.

Martin, Evan, Xie, Tao (2007)[4]**:** Access control policies are increasingly written in specification languages such as XACML. To increase confidence in the correctness of specified policies, policy developers can conduct policy testing with some typical test inputs (in the form of requests) and check test outputs (in the form of responses) against expected ones. Unfortunately, **manual test** generation is tedious and manually generated tests are often not sufficient to exercise various policy behaviors. CIRG is a supporting tool that generates tests based on change-impact analysis. This experimental results show that CIRG can effectively generate tests to achieve high structural coverage of policies and outperforms random test generation in terms of structural coverage and fault-detection capability.

---

[3] Kaufman, L.M, Johnson, B.W, Dugan, J.B, Dept. of Electr. & Comput. Eng., Virginia Univ., Charlottesville, VA, 'COVERAGE ESTIMATION USING STATISTICS OF THE EXTREMES FOR WHEN TESTING REVEALS NO FAILURES', IEEE Transactions, Computers, Jan 2002, Vol. 51, No.1, pp.3-12.

[4] Martin, Evan, Xie, Tao, North Carolina State University, USA, 'AUTOMATED TEST GENERATION FOR ACCESS CONTROL POLICIES VIA CHANGE-IMPACT ANALYSIS', Software Engineering for Secure Systems, SESS '07: ICSE Workshops 2007,Third International Workshop on 20-26 May 2007, Minneapolis, MN, USA, pp-5.

Salima. T.M.S, Ummu, Askarunisha. A, Ramaraj. N. (2007)[5]**:** In software development practice, testing accounts for as much as 50% of total development efforts. Testing can be manual, automated, or a combination of both. **Manual testing** is the process of executing the application and manually interacting with the application, specifying inputs and observing outputs. Manually testing the software is inefficient and costly. It is imperative to reduce the cost and improve the effectiveness of software testing by automating the testing process, which contains many testing related activities using various techniques and methods. In order to automate the process, some ways are required to generate oracles from the specification, and generate test cases to test the target software against the oracles to decide their correctness. Agent technologies facilitate the automated software testing by virtue of their high level decomposition, independency and parallel activation. Usage of agent based regression testing reduces the complexity involved in prioritizing the test cases. With the ability of agents to act autonomously, monitoring code changes and generating test cases for the changed version of the code can be done dynamically.

Saurabh Sinha, Mary Jean Harrold (2000)[6]**:** Analysis techniques, such as control flow, data flow, and control dependence, are used for a variety of software engineering tasks, including structural and regression testing, dynamic execution profiling, static and dynamic slicing, and program understanding. To be applicable to programs in languages such as Java and C++, these analysis techniques must account for the effects of exception occurrences and exception handling constructs; failure to do so can cause the analysis techniques to compute incorrect results and, thus, limit the usefulness of the applications that use them.

---

[5] Salima. T.M.S, Ummu, Askarunisha. A., Ramaraj. N, 'ENHANCING THE EFFICIENCY OF REGRESSION TESTING THROUGH INTELLIGENT AGENTS', Conference on Computational Intelligence and Multimedia Applications, International Conference on 13-15 Dec. 2007, Sivakasi, Tamil Nadu, India, Vol. 1 pp. 103-108.

[6] Saurabh Sinha, Mary Jean Harrold, Coll. of Comput., Georgia Inst. of Technol., Atlanta, GA, 'ANALYSIS AND TESTING OF PROGRAMS WITH EXCEPTION HANDLING CONSTRUCTS', IEEE Transactions, Software Engineering, Sep 2000, Vol. 26, No. 9, pp. 849-871.

Simons, Anthony J.H, Thomson, Christopher. D (2007)[7]: Lazy systematic unit testing with JWalk is compared against regression testing with JUnit, the leading agile testing tool. JWalk produced exhaustive test sets more quickly and recalculated full state and transition coverage, when testing modified or extended classes. For the same time and effort invested, JWalk tested up to two orders of magnitude more paths than **manual tests** created for JUnit by an expert tester.

Subramanyan, Rajesh (2007)[8]: Developing and implementing a successful test automation strategy can provide enormous benefit for a software project. However, automating tests is not cheap or easy. It does not replace the need for **manual testing** or enable to "down-size" the testing group. Automated testing can be made to be cost-effective, if best practices are applied to the process.

Wee Kheng Leow, Siau Cheng Khoo, Yi Sun (2004)[9]: Most research on automated specification-based software testing has focused on the automated generation of test cases. Before a software system can be tested, it must be set up according to the input requirements of the test cases. This setup process is usually performed manually, especially when testing complex data structures and databases. After the system is properly set up, a test execution tool runs the system according to the test cases and pre-recorded test scripts to obtain the outputs, which are evaluated by a test evaluation tool. The system combines the setup process, test execution, and test validation into a single test program for **testing** the behavior of object-oriented classes.

---

[7] Simons, Anthony. J.H, Thomson, Christopher. D, University of Sheffield, 'LAZY SYSTEMATIC UNIT TESTING: JWALK VERSUS JUNIT', Testing: Academic and Industrial Conference Practice and Research Techniques, Mutation, 10-14 Sept. 2007, Windsor, UK, pp. 138.

[8] Subramanyan, Rajesh, SCR US, 'TEST AUTOMATION IN PRACTICE', Computer Software and Applications Conference, COMPSAC, 24-27 July 2007 , Beijing, China, Vol.1, pp.19

[9] Wee Kheng Leow, Siau Cheng Khoo, Yi Sun, Dept. of Comput. Sci., National Univ. of Singapore, 'AUTOMATED GENERATION OF TEST PROGRAMS FROM CLOSED SPECIFICATIONS OF CLASSES AND TEST CASES', Software Engineering, ICSE, Proceedings, 26th International Conference on 23-28 May 2004, pp.96-105.

1.3. STATEMENT OF PROBLEM:

Software development is the translation of a user need or marketing goal into a software product. It is slow, expensive and error prone, often resulting in products with a large number of defects which cause serious problems in usability, reliability and performance. To combat this problem, software testing provides a systematic and empirically-guided approach to control and improve development processes and final products.

In this project three types of testing are done on Project Tracking System. They are unit testing, GUI testing and usability testing. Software unit testing resources to help software testers make sure individual modules or units of source code work properly.

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation.

GUI software testing is the process of testing a product that uses a graphical user interface, to ensure it meets its written specifications. Validation of GUI characteristics against GUI requirements is known as GUI testing. Most clients in client/server and web-based systems deliver system functionality using a graphical user interface (GUI).

Usability testing is also called as 'Testing for User-Friendliness'. It focuses on measuring a human-made product's capacity to meet its intended purpose. Examples of products that commonly benefit from usability testing are web sites or web applications, computer interfaces, documents, or devices. Usability testing measures the usability, or ease of use, of a specific object or set of objects.

1.4. OBJECTIVES OF THE STUDY:

The present study is made for the following objectives:

1.4.1. Primary Objective:

- To generate test cases, perform Unit testing, GUI testing and Usability testing and to study and analyze various test reports on Project Tracking system of Vee Eee Technologies, Chennai.

1.4.2. Secondary Objectives:

- To find out which module has most number of failure rate, the kind of testing which has more fail rate and their composition in the software.
- To find out the bugs and offer solution based on the findings to the programmer of the organization for reworking on the developed application.
- To find out the performance of software based on the test plan, the number of test cases worked and the type of testing.

1.5. SCOPE OF THE STUDY:

Testing of this project is essential to ensure the successful delivery of the developed product. It systematically uncovers different classes of errors in a minimum amount of time and with a minimum amount of effort. Testing is also done to meet the requirements of the user as stated in the specification.

1.6. METHODOLOGY:

Research methodology is a way to systematically solve the research problem. It shows all the details of data which have been used for the project and procedures which have been followed in the study.

1.6.1. Research Scheme:

1. Test Plan Preparation
2. Test Case Generations
3. Testing
4. Analysis

1.6.2. Test Plan:

Version 2.0

Revision History

Date	Version	Description	Author
26/01/2008	1.0	Initial Version	B. Saranya
28/02/2008	2.0	Second Version	B. Saranya

Table of Contents **Page No**

1. Introduction

The test plan outlines the scope, approach, resources and schedule of all testing activities. It identifies the items and features to be tested, types of testing, resource requirements and an approach to project management.

Purpose

The test plan document for the Project Tracking System supports the following objectives:

- Identify existing project information and the software components that should be tested
- List the recommended requirements for test
- Recommend and describe the testing strategies to be employed
- Identify the required resources and provide an estimate of the test efforts
- List the deliverable elements of the test project
- Describe each test case that will be used in the system verification process.

Scope

The Project Tracking System will be unit tested, system tested, GUI tested and usability tested. Unit tests will address functional quality, GUI testing will ensure written specifications and usability testing involves measuring how well test subjects respond in four areas: efficiency, accuracy, recall, and emotional response.

2. Requirements for Test

2.1 Unit Testing

The unit test plan is the overall plan to carry out the unit test activities.

Unit Test Specifications

The unit test plan clearly specifies the scope of unit testing. The basic input/output of the units along with their basic functionality will be tested. The input units will be tested for the format, alignment and the accuracy. The unit test plan gives the rules of what data types are present in the system, their format and their boundary conditions.

Sequence of Testing

The sequences of test activities includes whether to execute positive test cases first or negative test cases first, to execute test cases based on the priority, to execute test cases based on test groups etc. Positive testing is that testing which attempts to show that a given module of an application does what it is supposed to do. Negative testing is that testing which attempts to show that the module does not do anything that it is not supposed to do. An application delivering an error when it should is actually an example of a positive test. A negative test would be the program not delivering an error when it should or delivering an error when it should not. Positive test cases prove that the system performs what is supposed to do while negative test cases prove that the system does not perform what is not supposed to do. Testing the screens, files, database etc., are to be given in proper sequence.

Basic Functionality of Units

The independent functionalities of the units are tested which excludes any communication between the unit and other units. The interface part is out of scope of this test level.

Procedure for Defining the Tests

To ensure that test conditions are defined in an expeditious yet comprehensive manner, the following procedure has been developed:

- Determine the intent of each function
- Determine any input and output for each function

- Define the inputs and outputs that will be used to test each function and document them
- Create positive and negative test conditions and corresponding test data
- Determine the steps to exercise the data

Defining Test Conditions

The following items will be included when defining test conditions as applicable:

- Required fields
- Boundary values
- Data types

Test Validation

Each test condition and its corresponding data must have an expected result to determine if the function behaves as defined in the specification. Any variance from the expected result will be documented in a problem report.

2.2 GUI Testing

The following items are common items to consider when performing GUI Testing:

Menus/Menu Items

- Meaningful menu bar labels (related to calling function)
- Order by frequency of use or importance
- Drop-down menu pulls down upon selection

Buttons

- Meaningful label names
- Consistency of label names
- Consistency of button placement and grouping
- Consistency of button size (width and height)

Toolbars

- Consistency of placement from page to page
- Graphic representation on buttons intuitively represents the function they are calling
- Tool tips used to assist user in learning associated functions

Check Boxes

- Descriptive labels
- Used only when multiple selections are an option (single options use radio buttons)

Color

- Sufficient contrast to reduce eye strain
- Avoid the use of red and green combinations due to color blindness differentiation problems
- Appropriate use for attracting attention

Form Labels

- Sufficient field length to accommodate common screen resolutions

Text Fields

- Field length matches database storage size or scrolls to accommodate

Messages and Message Boxes

- Error message text is consistent throughout the application
- Error messages are meaningful to the user and do not contain technical information (this information should be captured for the technician but unannounced to user)
- Informational messages which interfere with the user's interaction with the product should be limited or allow the user the ability to disable it, once learned

Page Design/Layout

- Graphics and text are not cluttered and difficult to view
- Layout is logical and does not require the user to search for typical functions

Fonts

- Custom fonts should be interchangeable with system default fonts without degradation of the page being viewed

Items to consider when testing:

- Are the menu items descriptive?
- Do the buttons describe the appropriate action?
- Are the button sizes and alignment consistent?
- Are the button placements consistent, e.g., OK on the left and Cancel on the right when placed horizontally?
- What is the overall appearance?
- Are the colors appealing?

- Are colors used consistently when designating specific functionality, e.g., blue underscore to represent a hyperlink?
- Is the text easy to read? Are paragraphs formatted in small sections (four or five sentences) versus several large, lengthy sentences?
- Are the messages informative?
- If the fonts used are not supplied with the typical operating system, are they available dynamically through the application?

2.3. Usability Testing

The following items are common items to consider when performing Usability Testing:

Navigation

The site should provide flexible navigation that enables the user to move freely through the site. Types of navigation include:

- Left-oriented

  Buttons or links are vertically lined up on the left side of the page versus the right side of the page to ensure that variances in screen width do not result in unpredictable button visibility.

- Top-oriented

  Buttons or links are horizontally placed on the top of the page, but must be placed in a manner which does not appear to conflict with the current page headings. Utilization of a separate frame is recommended for this type of button placement.

- Bottom-oriented

  Buttons or links are horizontally placed at the bottom of the page. Most often users must scroll to see them unless they are placed in their own non-scrolling frame.

- Map or center-oriented

  Graphics or designs dominate the page like a map, with the links or buttons placed with the appropriate graphic. It is important that these buttons appear distinguishable from the graphics or text surrounding them to ensure that their use is intuitive.

Site Map (Navigational Aid)

The site could include a site map that aids the user in finding his or her way around the site. A site map facilitates the user's ability to memorize navigation of the site.

Help

Context sensitive help, pop-up windows, tool-tips assist the user in explaining how to perform different tasks or use the application.

Links

Links bring you to another page or to another area on the current page and are consistently and easily identifiable.

Response Time/Performance

Response time is subjective to the user's environment as well as the environment controlling the application under test and any link between the 2 environments. Thus response and performance are difficult to monitor without the use of tools. Unacceptable response times should still be noted, however, as they may warrant the acquisition of tools.

Security

Security should be specific to the type of information it is protecting. It should not hinder the user's ability to easily navigate through the system.

Items to consider when testing:

- Is the application/product easy to use?
- Is the site easy to navigate through?
- Are 'clickable' items obvious to the user?
- Do you know where a button or link will lead? Are the buttons/links labeled properly?
- Is the most important content/information organized in a manner so that the user has it readily available?
- Do you need to scroll down to view important information on a page?
- Do the links bring you to the proper pages? Are you able to distinguish between links that have been visited and not visited?
- Is the navigation from page to page consistent?
- Can you move to other areas within the site without having to retrace your steps backwards? Is the same main navigation toolbar resident on all pages?
- Are navigational aids (site map, help information, documentation) available? Are they intuitive?
- Is the site secure? Can someone else get to your information without entering your username and password?
- What type of response time are you receiving? Is it taking an unacceptable amount of time to retrieve data? Is it taking an unacceptable amount of time to process data?

3. Resources

3.1 Hardware

- **Processor:** Pentium IV
- **RAM:** 128MB
- **Monitor:** Samtron
- **Keyboard:** 104 keys
- **Mouse:** Logitech

3.2 Software

- AJAX(Front End)
- PHP(Interface)
- MY SQL(Back End)

4. Project Milestones

Milestone Task	Start Date	End Date
Plan Test	20/1/2008	25/1/2008
Design Test	26/1/2008	02/2/2008
Implement Test	03/2/2008	12/2/2008
Execute Test	13/2/2008	28/2/2008
Evaluate Test	29/2/2008	05/3/2008

5. Deliverables:

For each test case executed, a test result form will be created. This shall include the name or ID of the test, the use case or supplemental specification to which the test relates, the date of the test, the ID of the tester, required pre-test conditions, and results of the test. Microsoft word will be used to record and report test results.

1.6.3. DATA COLLECTION:

- **Primary data:**

The primary data collected from the organization is:

1. The source code of the Project Tracking System
2. From the source code, test cases were generated manually.

3. The number of test cases generated is 874. The total number of unit test cases is 601. GUI test cases are 222 and usability test cases are 51.
4. Actual outputs were collected across expected outputs.
5. These test results are used for analysis.

➢ **Secondary data:**

- Test Plan
- Test Procedures

1.6.4. TOOLS FOR ANALYSIS:

In order to study and analyze the software test reports of the company the following tool is being used.

- MS-Excel Package

MS-EXCEL (Package):

MS-Excel (full name **MicroSoft Excel**) is a spreadsheet application written and distributed by Microsoft for Microsoft Windows and Mac OS X. It features calculation, graphing tools, pivot tables and, except for Excel 2008 for Mac OS X, a macro programming language called VBA (Visual Basic for Applications). It is overwhelmingly the dominant spreadsheet application available for these platforms and has been so since version 5 in 1993, and is bundled as part of Microsoft Office. MS-Excel is one of the most popular microcomputer applications to date.

Due to Excel's foundation on floating point calculations, the statistical accuracy of Excel has been criticized, as has the lack of certain statistical tools. Excel proponents have responded that some of these errors represent edge cases and that the relatively few users who would be affected by these know of them and have workarounds and alternatives. Excel also supports the second date format based on year 1904 epoch. The latest version 12.0.0 (2008) of MS-Excel has been released on January 15, 2008.

Data Navigation chart:

The following data navigation chart is used to provide flexible navigation that enables the user to move freely through the project.

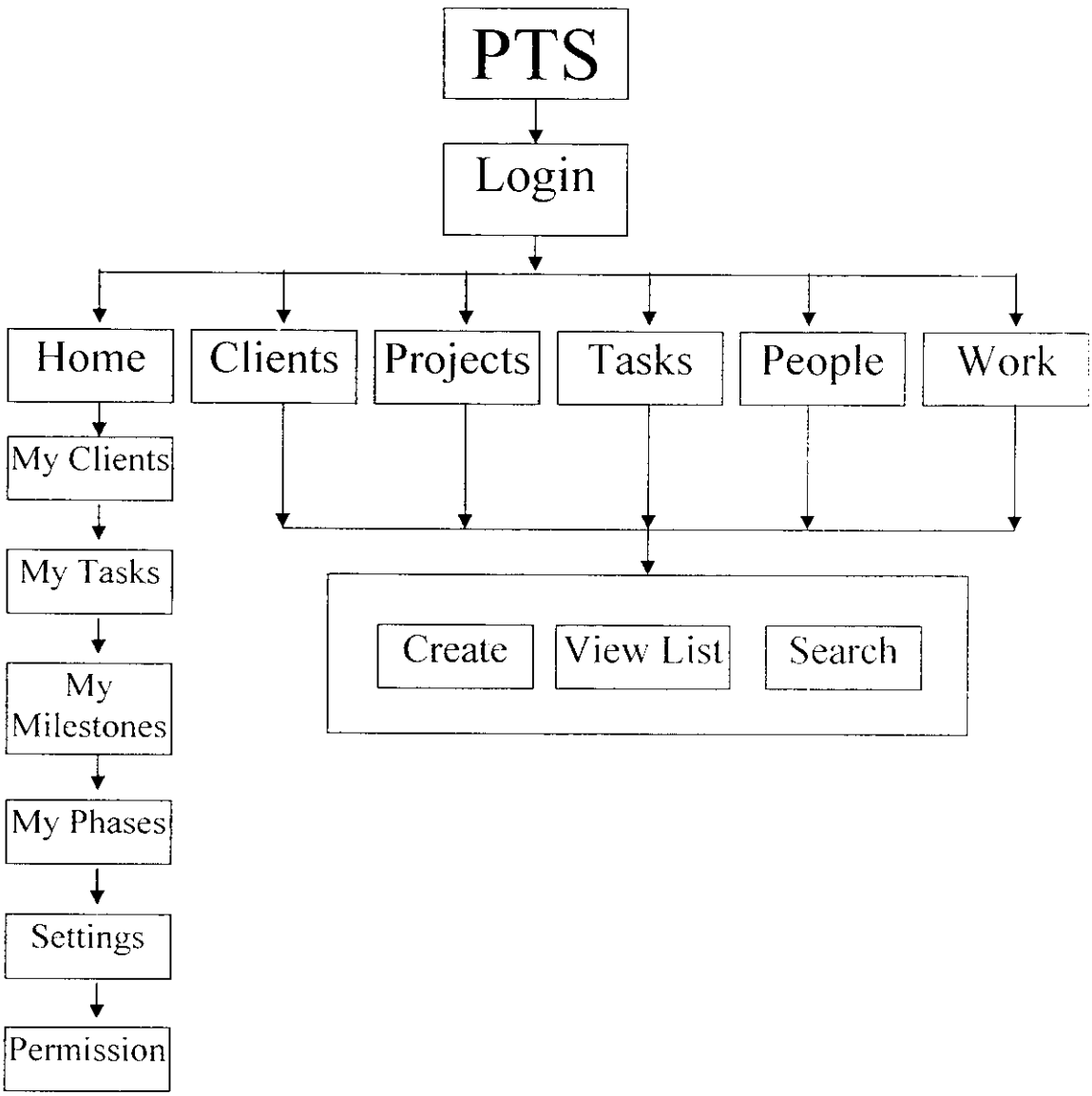


Figure 1.2 Data Navigation chart

1.7. LIMITATIONS:

- The types of testing are limited to unit, GUI and usability testing.
- As the testing is done manually the test cases is generated from the tester's view point. Therefore the result might differ from other individuals.

1.8. CHAPTER SCHEME:

The FIRST CHAPTER is introductory in nature. This chapter tells about the objectives, scope of the study and limitations.

The SECOND CHAPTER conveys about the history of Vee Eee Technologies, Chennai, highlighting the origin and development, product profile, management, services and functional departments of the company.

The THIRD CHAPTER gives the macro and micro scenario with respect to software testing.

The FOURTH CHAPTER presents the data analysis and interpretation.

The FIFTH CHAPTER gives summary of findings and concludes the study with relevant suggestion.

Organisation Profile

CHAPTER - 2

ORGANISATION PROFILE

2.1. HISTORY OF THE ORGANIZATION:

Vec Eee Technologies started in the year 2001, at Chennai which aims to focus for full service Information Technology solution provider. The concern architects, builds, manages and supports business information systems and accounting systems. Vec Eee Technologies delivers exceptional high quality service with great emphasis on customer care. They are one of the fastest growing embedded development companies on the market and expecting to keep growing in the future.

Vec Eee Technologies is headquartered at Chennai in South India. Since its establishment in 2001, the company has had a vision and commitment to produce and deliver quality products adhering to International Standards.

2.1.1. Branches:

The organization which is headquartered at Chennai has its presence throughout Tamilnadu and other states. The branches of the company are found at Madurai, Coimbatore and Hyderabad.

2.2. MANAGEMENT:

The team of embedded professionals is comprised of skilled, experienced problem solvers, not just coders and technicians. The managers and system architects have the business acumen to understand the demanding working standards and produce the high quality products needed. Their teams are comprised of the highest caliber of embedded designers and developers versed in development standards, best practice methodologies and all of the most popular as well as cutting edge development platforms and technologies.

2.3. ORGANIZATION STRUCTURE:

The organization structure of Vee Eee Technologies is a Top Down in nature.

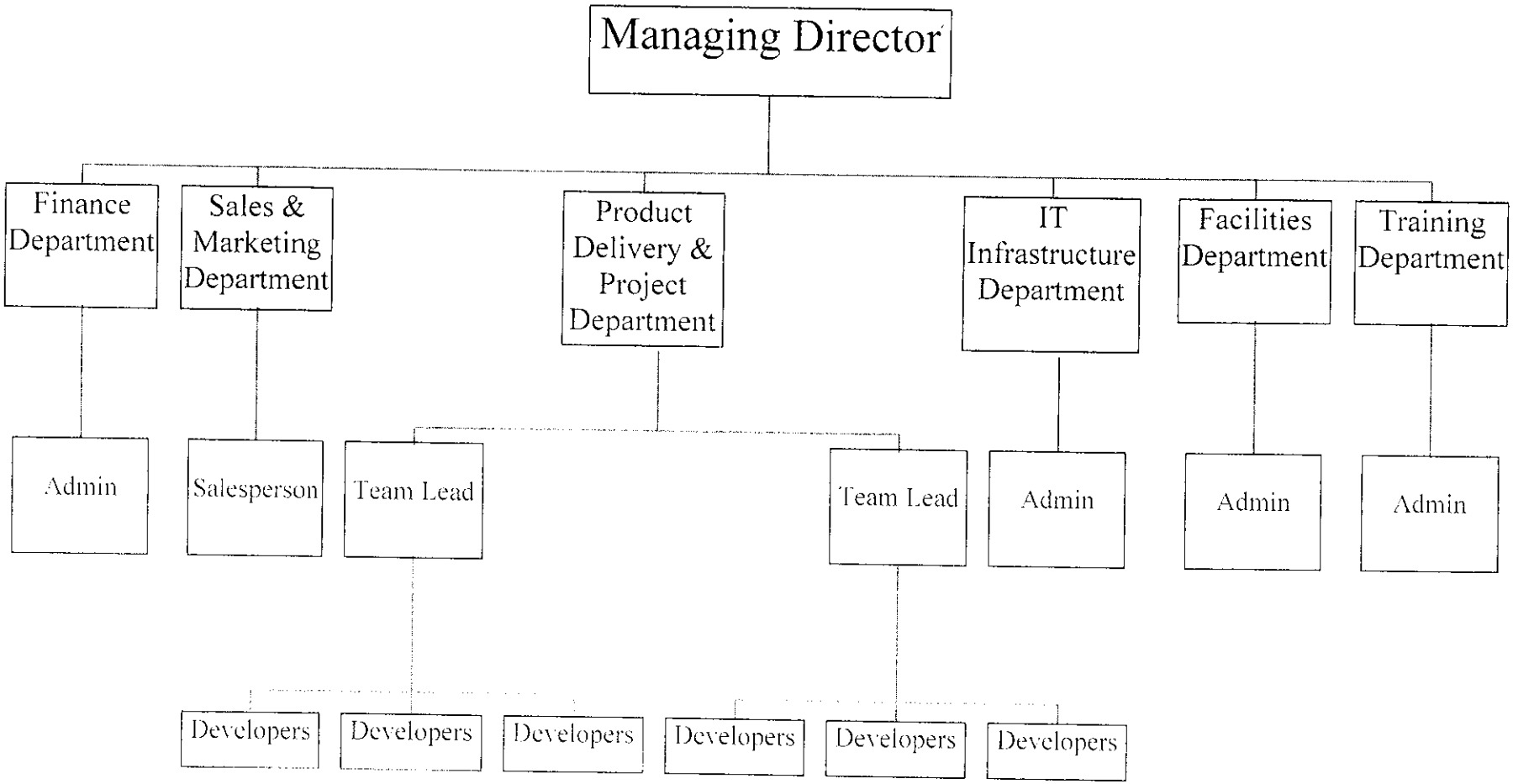


Figure 2.1 Organization Structure

2.4. PRODUCTS PROFILE AND MARKET POTENTIAL

Vee Eee Technologies products and services are designed to help companies meet the challenges they face in today's dynamic and competitive electronics markets. Design cycles continue to shrink and time-to-market is now critical to the success of most new electronic products. Increasingly DAMA (Design Anywhere, Manufacture Anywhere) is becoming the norm.

Global design and manufacturing, crossing continents and time zones, is needed to minimize the time taken to develop, launch and service such products. Electronic devices and systems need to be designed to be right-first-time for manufacturing processes, yet still meet the drive for miniaturization and increased functionality.

Design, purchasing and manufacturing must be flexible and located in the most cost-effective plants, wherever in the world they may be. Purchasing and logistics must be streamlined. And quality must be maintained throughout.

These issues and more are addressed by Vee Eee Technologies product solutions and support services.

2.4.1. Products:

At Vee Eee Technologies they have a rich portfolio of designing FPGA-based systems from concept to implementation. They have designed FPGA-based systems for educational institutions and embedded training centers. The Hardware modules are mapped as fully parallel, time-shared, or micro-coded state machines on FPGA. In the IC Design space, Vee Eee Technologies provides complete architecture modeling, logic design and verification, physical design, and backend operations for supplying fully tested parts.

VLSI team specializes in mapping computationally intensive algorithms in Hardware. The team has experience in mapping algorithms as programmable processor with enough flexibility to map an entire family of these algorithm and protocols on the hardware.

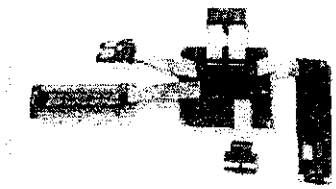


Figure 2.2 Spartan with Interfacing Board

Having successfully completed projects for Engineering Professionals in 2005 – 2007, they are now on a mission of giving our service to the industries who are focusing on the fields in which we are specialized. When it comes to Digital Signal Processing they are specialized in Audio Signal Processing, Digital Image Processing, Wireless Communication, etc. They have worked on various projects under this area.

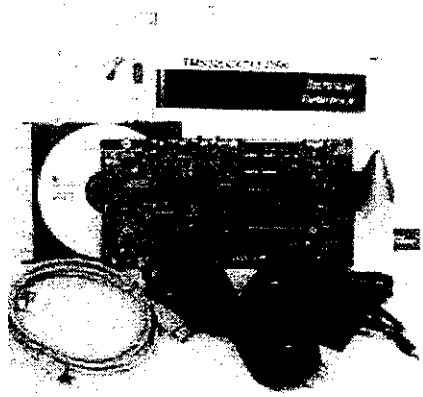


Figure 2.3 TMS320C6713 DSK

Vec Eee Technologies has strong expertise in real- time embedded systems for industry products in the Consumer Electronics, Networking and Wireless segments. Having strong roots in embedded software & hardware solutions, they bring distinctive experiences from engineering experts in R&D, Systems Integration & Product development.

2.4.3. Services:

Some of the services Vee Eee Technologies provides are

- Product concept and specification development.
- Complete project management.
- Electronic hardware design and implementation.
- Embedded firmware design and implementation.
- Logic integration.
- Discrete logic or PALs combined into large FPGAs re-designs for cost reduction, manufacturing or test enhancement.
- Design of test fixtures, test software and diagnostics.
- Digital signal processing

2.4.4. Solutions:

Whether you're a small start-up company or a fortune 500 corporation, they have enough experience and the technology necessary to design a scalable, customized solution for your business infrastructure and need.

The company design cost effective robust technology solutions with excellent usability characteristics for each individual customer. Their custom product development services that will shorten your time to market fill the expertise gaps and help to control run-away development costs.

They offer design solutions in the area of consumer electronics, embedded networking, system integration, testing equipments, digital signal processing and industrial control.

The focus on these areas of system development of the organization provides a unique solution not found in other firms.

2.5. FUNCTIONAL DEPARTMENTS:

2.5.1. Product Delivery & Project Department

Customer satisfaction and delivery of products in stipulated time is the core of any IT organization. Vee Eee Technologies is having a separate product delivery department, which is headed, by one of the experienced professionals from the IT market. This is one of the biggest departments of our company with the total head count of around 150. This department is responsible for monitoring the entire SDLC activities of the company, which includes Requirements gathering, Designing, Development, Testing, Quality Assurance and product delivery as per the market standard. This department will also ensure that the developed applications are delivered to the customers in specified deadline and with high quality. The company focuses more on the fact that the customers are highly satisfied with the final product delivered.

2.5.2. Sales & Marketing Department

Taking our products to the world market plays the vital role in development of the organization to great heights. The sales & marketing team of Vee Eee Technologies is one of the core departments that run our business. There are basically two ways of communications that can be performed. They are the communication between the various departments of our company and the other one is communication between the various clients. This department ensures that the product developed is reached the market and communicating with the clients and stakeholders to improve the business further. An effective 40 member's team at Vee Eee Technologies is running this department successfully for a long term.

2.5.3. Finance Department

Finance is the blood of any business. Finance is that activities which is concerned with acquisition and conversion of capital funds in meeting the financial needs and over all objectives of business enterprises. The main function of this department is to provide finance to various departments. Finance manager controls the finance department.

The turnover of the company in 2007 is Rs.75 Lakhs. There are 10 employees in the finance department. The company is growing in exponentials in both domestic and international markets.

The Advertisement cost for the company in initial stages is very much reduced from 40% to 10%. Vee Eee Technologies provides all provident funds (PF) to all its permanent employees. About 12% of the basic pay is taken for PF from the salary of the employees and Vee Eee Technologies is contributing another 13% for the total benefit. So, the employees will get total of 25% paid after the retirement. Most of the payments are done through the direct deposit of the employees bank salary account. Vee Eee Technologies preferred banks are HDFC and ICICI. We use the e-Payment, and payment management system, which is integrated with SAP. The Salary statement is generated automatically and will be emailed to the employees at the end of the month.

2.5.4. IT Infrastructure Management (IT-ISM)

Applications, operating systems, storage, networks and databases need to support high reliability and availability. IT managers need a view across all components of the application infrastructure to maintain service quality. Legacy systems must be brought forward and enabled for XML, Web services and other new technology. Tools crossing middleware, EAI, message brokers, applications servers and database gateways must be combined to meet today's requirements. In Vee Eee Technologies, we've a separate IS Team they provide all the IT support for executing day-to-day deliverables successfully. We've around 25 people to provide support for all the IT related requirements within the organization.

2.5.5. Facilities Department

This department provides support for all the non-IT requirements for the company, which includes provisioning of various materials related to work environment.

2.5.6. Training Department

Imparting knowledge to the customers about the products we develop is one of the key functionalities of Vee Eee Technologies. We've a team of expertise who are having experience in various technologies and products we use in our production environment. Apart from this, the employees are also involved in various training activities that can help them to improve themselves professionally and personally. This department is responsible for the improvement of enhanced productivity of the employees and converting the organization goals into deliverables.

CHAPTER – 3

MACRO – MICRO ANALYSIS

3.1. GROWTH OF SOFTWARE TESTING – MACRO ANALYSIS:

The evolution of software test engineering can be traced by examining changes in the testing process model and the level of professionalism over the years. The current definition of a good software testing practice involves some preventive methodology.

The separation of debugging from testing was initially introduced by Glenford J. Myers in 1979. Although his attention was on breakage testing, it illustrated the desire of the software engineering community to separate fundamental development activities, such as debugging, from that of verification. Dr. Dave Gelperin and Dr. William C. Hetzel classified in 1988 the phases and goals in software testing in the following stages:

- Until 1956 - Debugging oriented
- 1957-1978 - Demonstration oriented
- 1979-1982 - Destruction oriented
- 1983-1987 - Evaluation oriented
- 1988-2000 - Prevention oriented

3.1.1. Software testing survey shows big spend brings little comfort:

A survey done by The Original Software Group suggests that most companies are uncomfortable with the quality or level of testing done on new and revised software prior to releasing it to users - despite spending an average of over £600,000 a year on the activity.

In October 2004, Original Software, developer of market leading automated software testing solutions, commissioned an independent study of the software testing methodologies and procedures used by a range of companies. The results of the survey indicated clearly that the threat of software failure is being taken no more seriously now than it was four years ago

when a similar study was conducted, despite testing practices that cost the average company over £600,000 per annum.

This study surveyed 130 companies based in Europe, the United States, Canada, and Australia. A link to an online survey was emailed out to 2,000 IT professionals. The data produced by this survey showed that everyone from software developers to IT Directors believe they are expected to produce more with less: develop faster, go live sooner, and yet improve quality to minimize risk and exposure - all with fewer resources.

The study also revealed that IT teams average 18 developers, five Quality Analysts and four User Acceptance Testing staffers, and that IT departments spend an average 33% of their time on software testing activities (developers 33%, QA 36% and UAT 29%). At 13 hrs per person per week, and with a staff cost to companies averaging £ 34 per hour, that is £ 23,264 per person per year. Hence, the total average expenditure related to software testing for the companies that were surveyed is £628,143 per year.

Only 19.4% of respondents said they used some type of automated testing tool, the remainder relying upon manual testing methods. The prevailing reasons given for not automating test processes were: Time (35%); cost (33%); script/data maintenance (26%); and complexity (18%), thus reflecting the pressure on allocation of resources.

3.1.2. A survey of software testing practices in Alberta:

Software organizations have typically de-emphasized the importance of software testing. Results indicate that Alberta-based organizations tend to test less than their counterparts in the United States. The results also indicate that Alberta software organizations tend to train fewer personnel on testing-related topics. This practice has the potential for a two-fold impact: first, the ability to detect trends that lead to reduced quality and to identify the root causes of reductions in product quality may suffer from the lack of testing. This consequence is serious enough to warrant consideration, since overall quality may suffer from the reduced ability to detect and eliminate process or product defects. Second, the organization may have a more difficult time adopting methodologies such as extreme programming. This is significant because other industry studies have concluded that many software organizations

have tried or will in the next few years try some form of agile method. Newer approaches to software development like extreme programming increase the extent to which teams rely on testing skills. Organizations should consider their testing skill level as a key indication of their readiness for adopting software development techniques such as test-driven development, extreme programming, agile modeling, or other agile methods.

3.1.3. A preliminary survey on software testing practices in Australia:

The first survey on software testing practices carried out in Australian ICT industry with a total of 65 organizations from various major capital cities in Australia conducted between 2002 and 2003, focused on five major aspects of software testing, namely testing methodologies and techniques, automated testing tools, software testing metrics, testing standards, and software testing training and education. Based on the survey results, current practices in software testing are reported, as well as some observations and recommendations for the future of software testing in Australia for industry and academia.

3.2. GROWTH OF SOFTWARE TESTING – MICRO ANALYSIS:

3.2.1. Software testing in India:

India is poised to capture a major share of the worldwide software testing market. This is because of the established and dominant IT service sector, presence of organizations with matured processes and practices, and the versatile IT skill-set of testing professionals.

The worldwide testing market is estimated at $13 billion. The global outsourcing testing market opportunity in this year has been estimated at $4.5 billion, of which, nearly $3 billion will be offshored to cheaper destinations.

India has the potential to corner 70 per cent ($1.82 billion) of the outsourced testing market. The compounded annual growth rate for the independent outsourced testing market is estimated at 56 per cent while the independent offshore testing has been estimated at 92 per cent over the next four years.

The size of the testing market in India is estimated to be between Rs. 150 crore (Rs. 1.5 billion) and Rs. 200 crore (Rs. 2 billion). Software companies, from India and abroad, are investing in establishing 'centers of excellence' to effectively tap this growing market.

A survey on the recruitment consultants in IT and ITES sectors conducted recently in Bangalore showed 24 per cent of the recruitment were in software testing -- highest after call centers (47 per cent). It is estimated that Bangalore alone needs about 8,000-10,000 testers and about 16,000 to 18,000 in India.

If software testing needed to be an engineering discipline, it would have to get integrated into the process of production, not as a post-thought. Testing is not only to find defects but also to prevent it, be it in design discipline, code discipline and also to check the behaviour of the product. Hence it is appropriate to call testing, which ensures discipline in all other engineering functions, an engineering discipline, he added.

Moving forward, software testing will enter the next level of maturity. This would demand expertise in specific domains like embedded systems testing, real-time testing, healthcare systems testing, web-application testing, client server application testing, performance testing, connectivity testing, inter-operability testing and performance benchmarking.

3.2.2. India to emerge as global leader in software testing:

India has the potential to garner 70% share of the outsourced testing market. India is on its way to becoming a leader in the software testing market with more and more companies outsourcing their software testing services to the India that has a large availability of testers. Over the last few years, the Indian players have proved that they can offer testing services at a fraction of the cost. Software testing makes sense, since it costs a lot of money and time to find and fix the software bugs or errors in software, once the product is released in the market. Software bugs cost the US economy an estimated $59.5 billion annually, which is equivalent to 0.6 per cent of its gross domestic product. Users and the developers end up paying for the bugs in the software. This loss can be reversed if the companies invest in timely and high quality software testing.

Software testing is not a new phenomenon, but has been a part of any software development or maintenance process. Companies tested the product in-house by establishing separate testing divisions. This was known as detached testing where the testing team did not consist of developers who wrote the codes and therefore they could test the product better. META Group study reveals that the software testing market in India will touch $700 million to $1 billion by 2007.

However, many companies cannot afford to set up a separate testing division. This is where outsourcing comes in. The companies who cannot invest in a specialized testing division have turned to independent software testing providers in offshore locations like India. Outsourcing software testing services has its distinct advantages. Companies who outsource testing service enjoy a reduced cycle time, a shorter time to market and a high quality of service at a far lower cost. Besides, a third party tester is more like a beta user of the product, who can study the software program and find critical faults. A majority of the companies, in the US, Europe and Japan, are now outsourcing testing services either to their own subsidiaries or independent testing companies.

Research firm Gartner estimates the worldwide software testing market at $13 billion and the global market size of outsourced testing services is around $6.1 billion. India according to reports has the potential to corner 70% share of the outsourced testing market. A Meta Group study reveals that the software testing market in India will touch $700 million to $1 billion by 2007. The top ten IT firms in India including Infosys, Wipro, Cognizant Technology Solutions and Tata Consultancy Services (TCS) have been quick in creating independent testing practices in-house.

Testing is however increasingly being seen as a specialized service. IT companies have started positioning themselves as independent software testing (IST) services providers, offering software testing as a standalone service. There are roughly ten IST players in India of a reasonable size with a combined turnover of around Rs 200 crore. The leaders include RelQ, ThinkSoft, ReadyTestGo, Maveric, Applabs, Polaris Application Certification Enterprise (PACE), IQA Tester and Stag Software.

Testing is now comparable with other IT services in terms of billing rates and compensation levels of testing professionals. The testing service market in India is booming with the rapid expansion of Indian IST firms. However, the challenge these companies face is in finding the right talent.

Testing requires a good knowledge of the testing process and knowledge of the domain the product operates in. This kind of talent is not easily available. The testing industry does not have the support of training academies that focus on testing.

IST firms at present depend entirely on in-house capabilities to train beginners. The problem of finding talented testing professionals is compounded by the fact that IT professionals view software testing industry as a stop gap arrangement. To the industry's relief, this perception is changing gradually with more and more IT professionals looking at software testing as a serious career option.

Data Analysis & Interpretation

CHAPTER – 4

DATA ANALYSIS AND INTERPRETATION

In this project, testing and analysis has been done for Project Tracking system. The different types of testing done are unit testing, GUI testing and usability testing. Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation.

GUI software testing is the process of testing a product that uses a graphical user interface, to ensure it meets its written specifications. Usability testing focuses on measuring a human-made product's capacity to meet its intended purpose.

The number of test cases generated is 874. The total number of unit test cases is 601, GUI test cases are 222 and usability test cases are 51.

TABLE – 4.1 TEST RESULTS

TESTING/STATUS	Number of test cases		
	Pass	Fail	Total
Unit	460 (77%)	141 (23%)	601 (100%)
GUI	212 (95%)	10 (5%)	222 (100%)
Usability	48 (94%)	3 (6%)	51 (100%)
Total	720	154	874

Interpretation:

From the above values we could find that the total number of test cases is 874 where the total of pass test cases is 720 and the total of fail test cases is 154. Here the failure rate of 'unit testing' is 23%,'GUI testing' is 5% and for 'usability testing' is 6%.

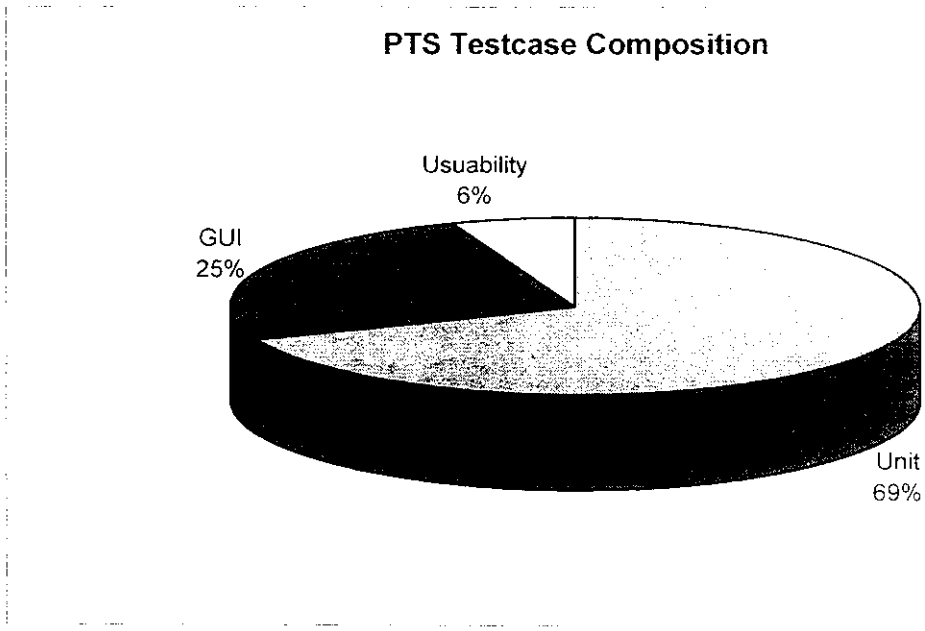


Figure – 4.1 PTS test cases composition

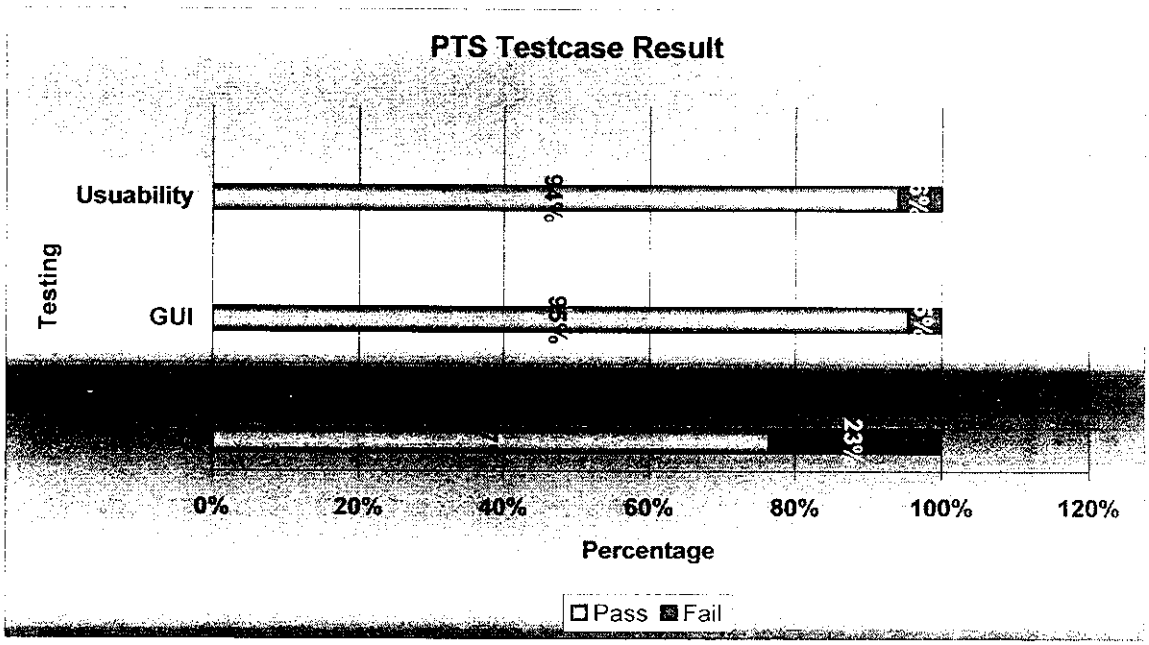


Figure – 4.2 PTS test results

TABLE – 4.2 NUMBER OF TEST CASES FOR UNIT TESTING

FIELD NAME/STATUS	Number of test cases		
	Pass	Fail	Total
pen	2	0	2
ername	4	6	10
ssword	8	5	13
g In	2	0	2
member me	2	0	2
rgot password	7	0	7
me page	1	0	1
sks	14	4	18
it your profile	31	25	56
ients control panel	4	0	4
ients create	5	11	16
ients view list	15	11	26
ients search	3	1	4
oject control panel	4	0	4
oject create	36	14	50
oject view list	7	1	8
oject search	3	0	3
sk control panel	6	0	6
sk create	32	2	34
sk view list	11	3	14
sk search	3	1	4
ople control panel	4	0	4
ople create	28	26	54
ople view list	12	1	13
ople search	5	2	7
w project create	15	3	18
w project view list	8	0	8

New project search	3	0	3
My milestones	12	2	14
Admin control panel	23	1	24
Owner settings	7	4	11
Admin settings	9	2	11
User type permission	6	0	6
Manage create	9	1	10
Manage people	11	1	12
Gantt management	9	0	9
Manage messages	18	3	21
Manage tasks	7	3	10
Manage milestones	10	2	12
Manage asset type	2	1	3
People permission	17	2	19
Work control panel	4	0	4
Work create	23	1	24
Work view list	12	0	12
Work search	6	2	8
Total	460	141	601

TABLE – 4.3 NUMBER OF TEST CASES FOR GUI TESTING

FIELD NAME/STATUS	Number of test cases		
	Pass	Fail	Total
Tasks	7	0	7
Edit your profile	7	1	8
Clients create	4	0	4
Clients view list	5	0	5
Clients search	1	0	1
Project create	19	1	20
Project view list	3	0	3
Project search	1	0	1
Task create	3	1	4
Task view list	6	0	6
People create	9	1	10
People view list	5	0	5
New project create	7	1	8
New project view list	3	0	3
New project search	1	0	1
My milestones	6	0	6
Admin control panel	1	0	1
Owner settings	0	4	4
Admin settings	0	1	1
User type permission	1	0	1
Manage create	14	0	14
Manage project	20	0	20
Gantt management	2	0	2
Manage messages	33	0	33
Manage task	15	0	15
Manage milestone	15	0	15

Manage asset type	6	0	6
People permission	9	0	9
Work create	3	0	3
Work view list	5	0	5
Work search	1	0	1
Total	212	10	222

TABLE – 4.4 NUMBER OF TEST CASES FOR USABILITY TESTING

FIELD NAME/STATUS	Number of test cases		
	Pass	Fail	Total
Home page	12	1	13
Clients create	1	0	1
Project view list	1	0	1
Task view list	2	0	2
People view list	2	0	2
New project create	1	0	1
New project view list	1	0	1
My milestones	1	0	1
Admin control panel	1	0	1
Owner settings	0	1	1
Admin settings	1	0	1
Manage clients	2	0	2
Manage projects	4	0	4
Manage message	6	0	6
Manage task	8	1	9
Manage milestone	1	0	1
People permissions	3	0	3
Work create	1	0	1
Total	48	3	51

TABLE – 4.5 NUMBER OF TEST CASES FOR LOGIN SCREEN

FIELDS / STATUS	OPEN		USER NAME		PASS WORD		FORGOT PWD		REMEMBER ME		LOG IN		TOTAL
	No.	%	No.	%	No.	%	No.	%	No.	%	No.	%	No.
PASS	2	100	4	40	8	62	7	100	2	100	2	100	25
FAIL	0	0	6	60	5	38	0	0	0	0	0	0	11
TOTAL	2	100	10	100	13	100	7	100	2	100	2	100	36

Interpretation:

From the above table the total number of test cases is 36. The total number of pass test cases is 25. The total number of fail test cases is 11. Except Username and Password all the other fields have no fail rate. The fail rate of Username and Password is 60% and 38% respectively.

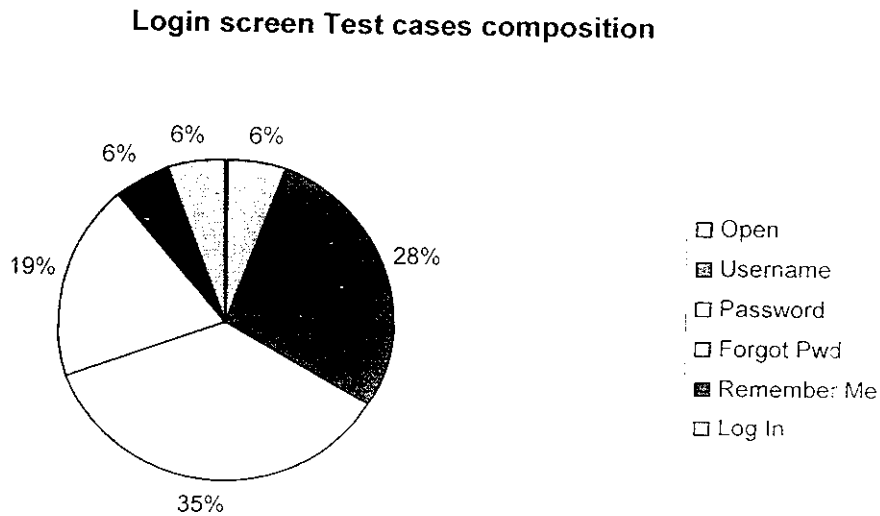


Figure – 4.3 Login screen test cases composition

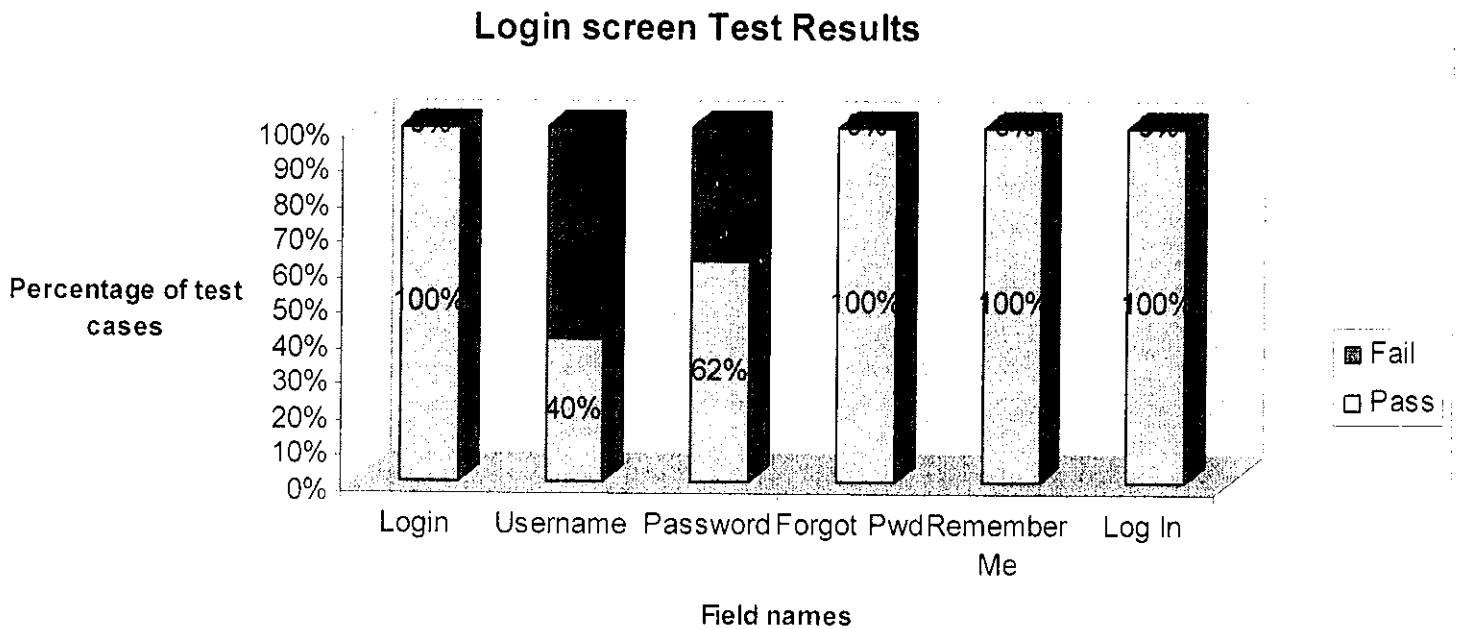


Figure – 4.4 Login screen test results

TABLE – 4.6 TEST RESULTS OF TESTING TYPES FOR LOGIN SCREEN

TESTING/STATUS	Number of test cases		
	Pass	Fail	Total
Unit	25 (69%)	11 (31%)	36 (100%)
GUI	0 (0%)	0 (0%)	0 (0%)
Usability	0 (0%)	0 (0%)	0 (0%)
Total	25	11	36

Interpretation:

From the above table we could find that the total number of test cases is 36 where the total of pass test cases is 25 and the total of fail test cases is 11. Here the failure rate of 'unit testing' is 31%.

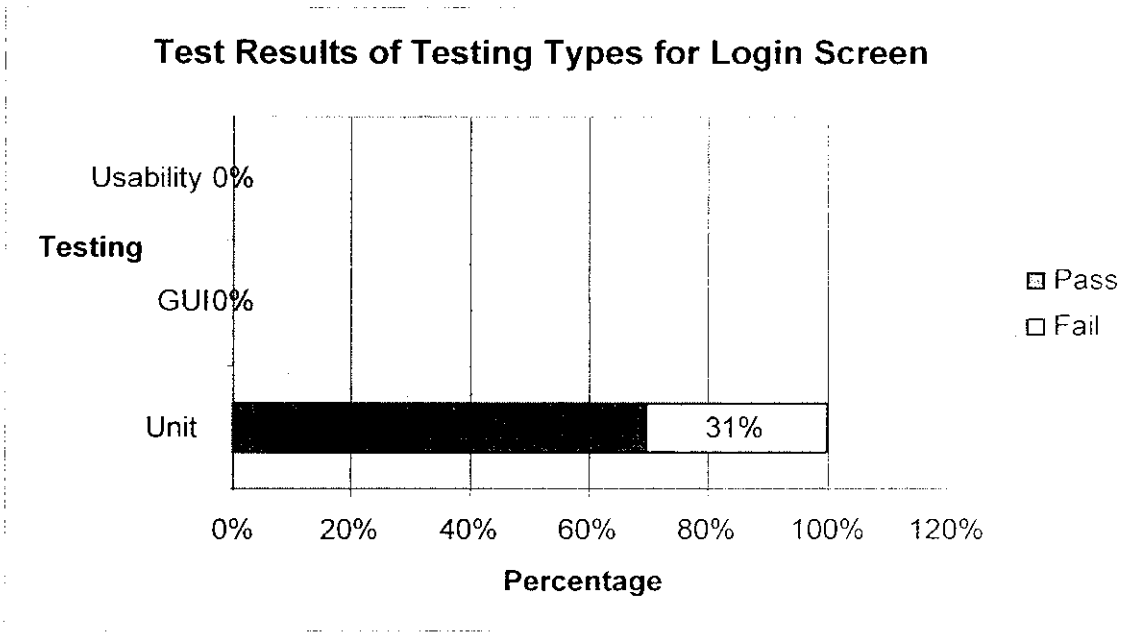


Figure – 4.5 Test results of testing types for Login screen

TABLE – 4.7 NUMBER OF TEST CASES FOR HOME SCREEN

FIELDS / STATUS	HOME PAGE		TASKS		EDIT YOUR PROFILE		TOTAL
	No.	%	No.	%	No.	%	No.
PASS	11	92	24	86	37	59	72
FAIL	1	8	4	14	26	41	31
TOTAL	12	100	28	100	63	100	103

Interpretation:

From the above table the total number of test cases is 103. The total number of pass test cases is 72. The total number of fail test cases is 31. Here all the fields have failure rate. The fail rate of Homepage, Tasks and Edit your Profile is 8%, 14% and 41% respectively.

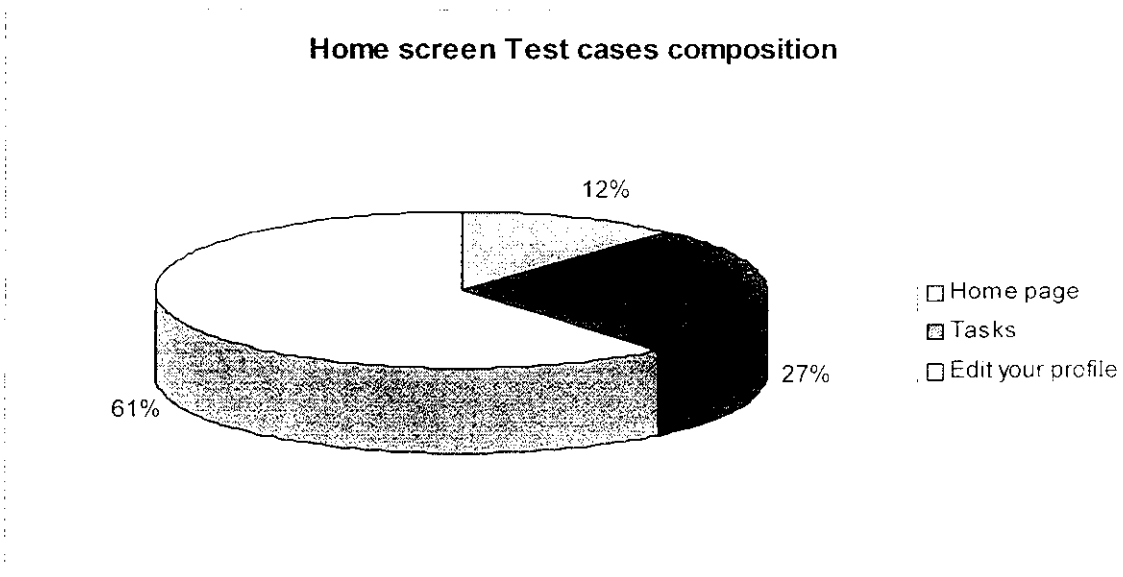


Figure – 4.6 Home screen test cases composition

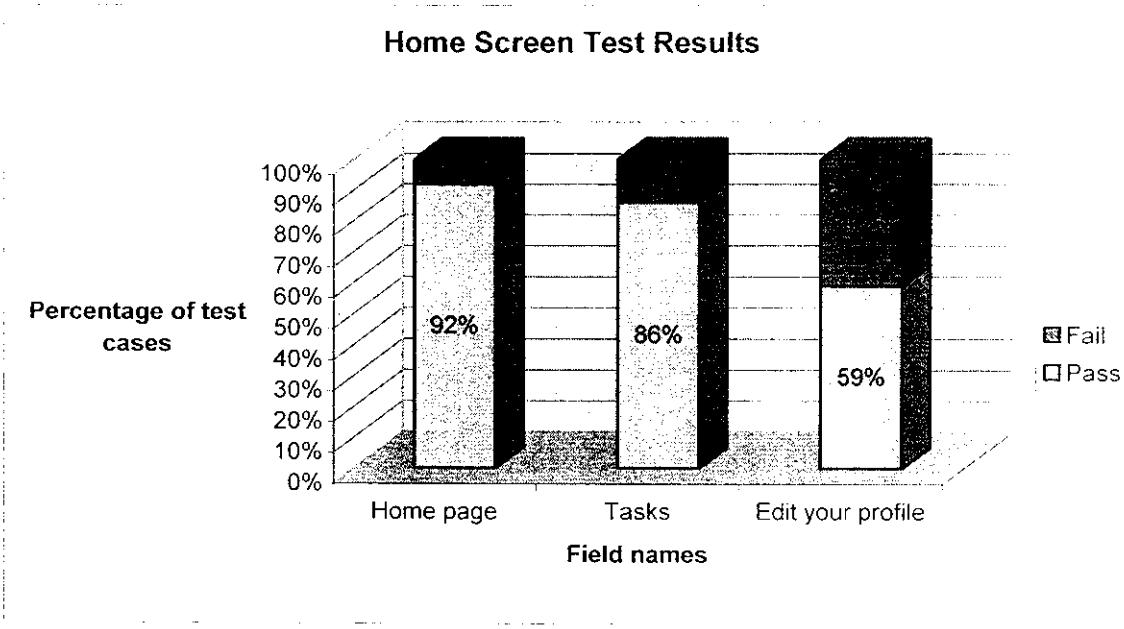


Figure – 4.7 Home screen test results

TABLE – 4.8 TEST RESULTS OF TESTING TYPES FOR HOME SCREEN

TESTING/STATUS	Number of test cases		
	Pass	Fail	Total
Unit	46 (61%)	29 (39%)	75 (100%)
GUI	14 (93%)	1 (7%)	15 (100%)
Usability	12 (92%)	1 (8%)	13 (100%)
Total	72	31	103

Interpretation:

From the above table we could find that the total number of test cases is 103 where the total of pass test cases is 72 and the total of fail test cases is 31. Here the failure rate of 'unit testing' is 39%,'GUI testing' is 7% and for 'usability testing' is 8%.

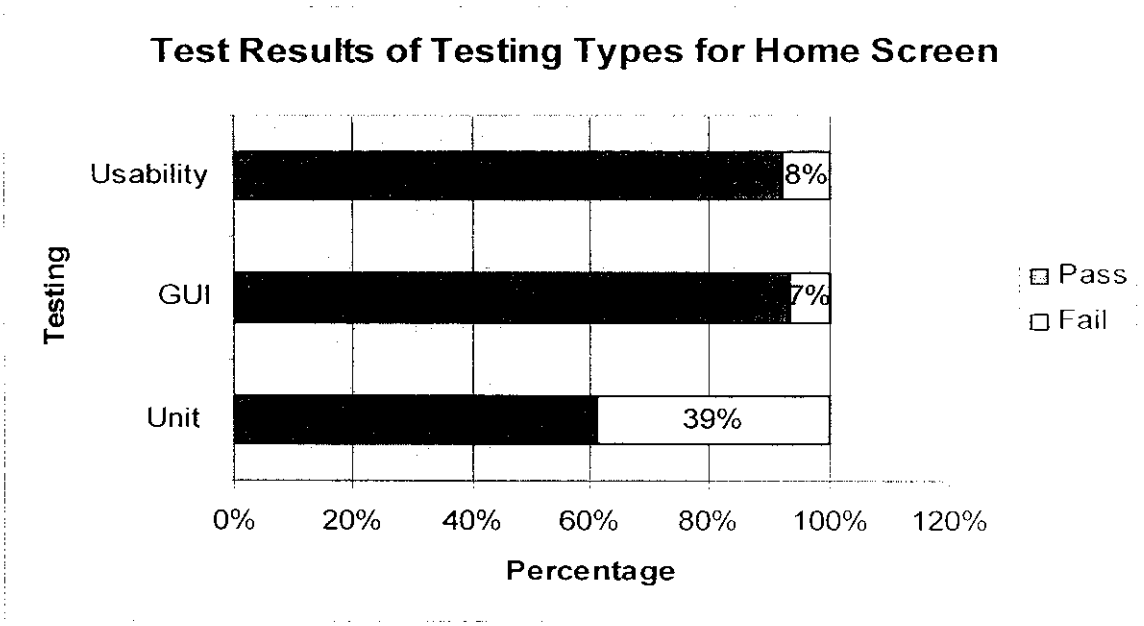


Figure – 4.8 Test results of testing types for Home screen

TABLE – 4.9 NUMBER OF TEST CASES FOR CLIENTS SCREEN

FIELDS / STATUS	CLIENTS CONTROL PANEL		CREATE		VIEWLIST		SEARCH		TOTAL
	No.	%	No.	%	No.	%	No.	%	No.
PASS	4	100	10	48	19	63	4	80	37
FAIL	0	0	11	52	11	37	1	20	23
TOTAL	4	100	21	100	30	100	5	100	60

Interpretation:

From the above table the calculated values are total number of test cases is 60, total pass test cases is 37 and total fail test cases is 23. Here only clients control panel has

100% pass rate. The other fields such as create, view list and search have 52%, 37% and 20% failure rate respectively.

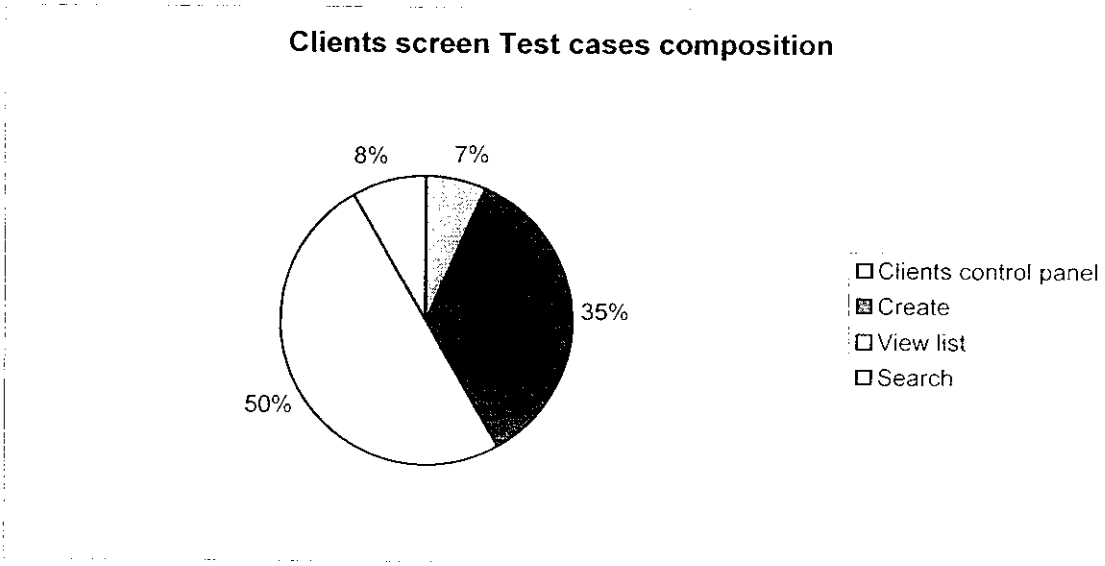


Figure – 4.9 Clients screen test cases composition

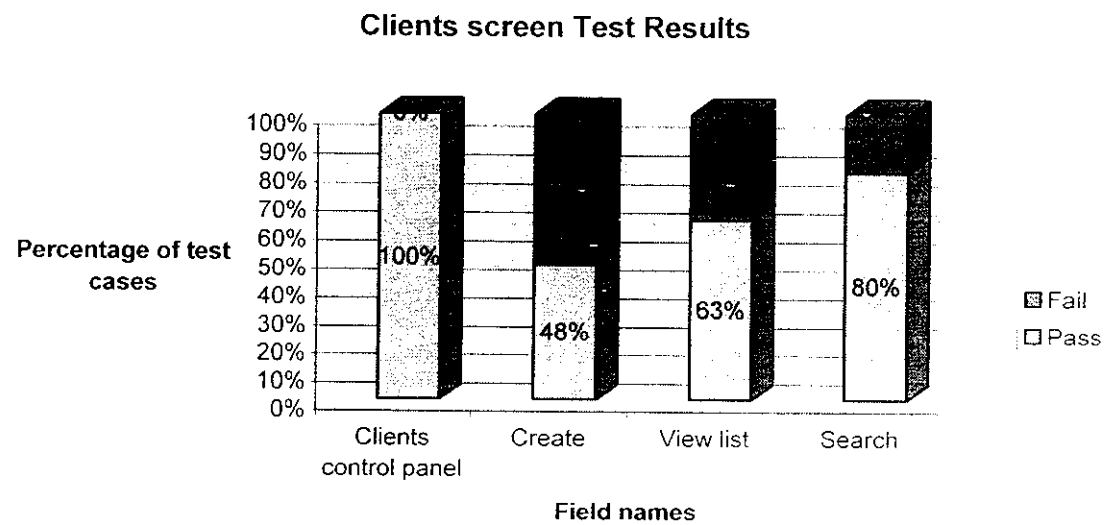


Figure – 4.10 Clients screen test results

TABLE – 4.10 TEST RESULTS OF TESTING TYPES FOR CLIENTS SCREEN

TESTING/STATUS	Number of test cases		
	Pass	Fail	Total
Unit	27 (55%)	22 (45%)	49 (100%)
GUI	10 (100%)	0 (0%)	10 (100%)
Usability	1 (100%)	0 (0%)	1 (100%)
Total	38	22	60

Interpretation:

From the above table we could find that the total number of test cases is 60 where the total of pass test cases is 38 and the total of fail test cases is 22. Here the failure rate of 'unit testing' is 45%.

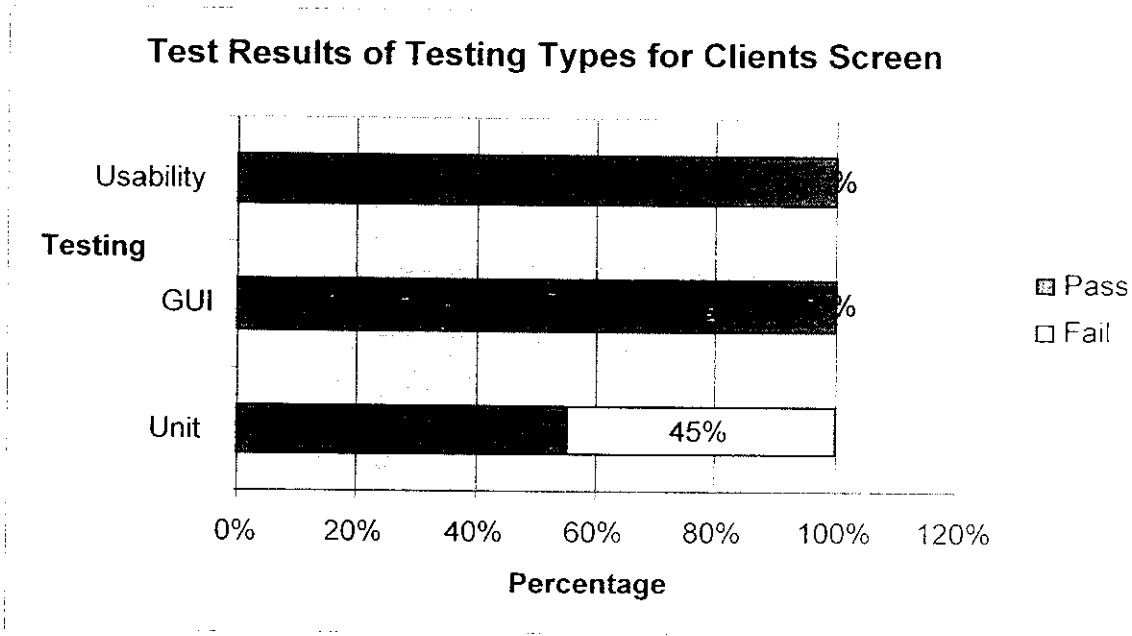


Figure – 4.11 Test results of testing types for Clients screen

TABLE – 4.11 NUMBER OF TEST CASES FOR PROJECTS SCREEN

FIELDS / STATUS	PROJECT CONTROL PANEL		CREATE		VIEWLIST		SEARCH		TOTAL
	No.	%	No.	%	No.	%	No.	%	No.
PASS	4	100	55	79	11	92	4	100	74
FAIL	0	0	15	21	1	8	0	0	16
TOTAL	4	100	70	100	12	100	4	100	90

Interpretation:

From the above table we could find that the total number of test cases found in the projects form is 90 and the total of pass and fail test cases is 74 and 16. Here projects control panel and search has 100% pass rate. The other fields such as create and view list have 21% and 8% failure rate respectively.

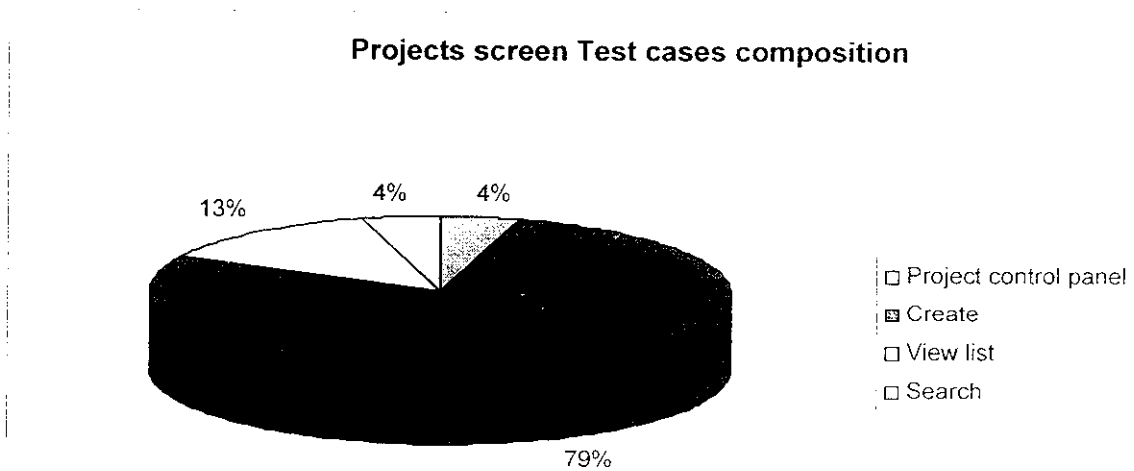


Figure – 4.12 Projects screen test cases composition

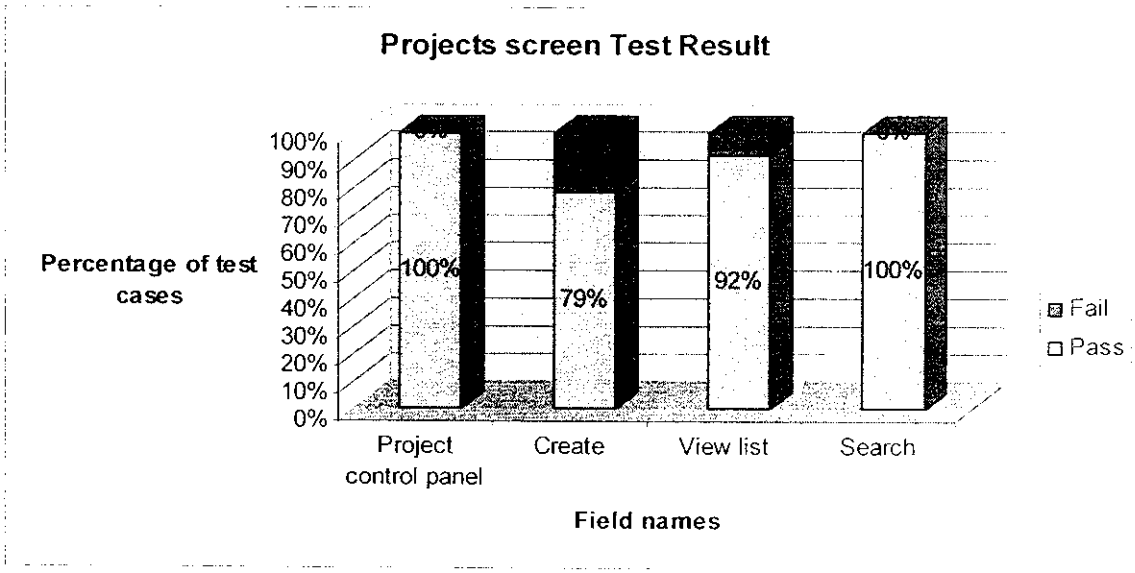


Figure – 4.13 Projects screen test results

TABLE – 4.12 TEST RESULTS OF TESTING TYPES FOR PROJECTS SCREEN

TESTING/STATUS	Number of test cases		
	Pass	Fail	Total
Unit	50 (77%)	15 (23%)	65 (100%)
GUI	23 (96%)	1 (4%)	24 (100%)
Usability	1 (100%)	0 (0%)	1 (100%)
Total	74	16	90

Interpretation:

From the above table we could find that the total number of test cases is 90 where the total of pass test cases is 74 and the total of fail test cases is 16. Here the failure rate of 'unit testing' is 23% and 'GUI testing' is 4%.

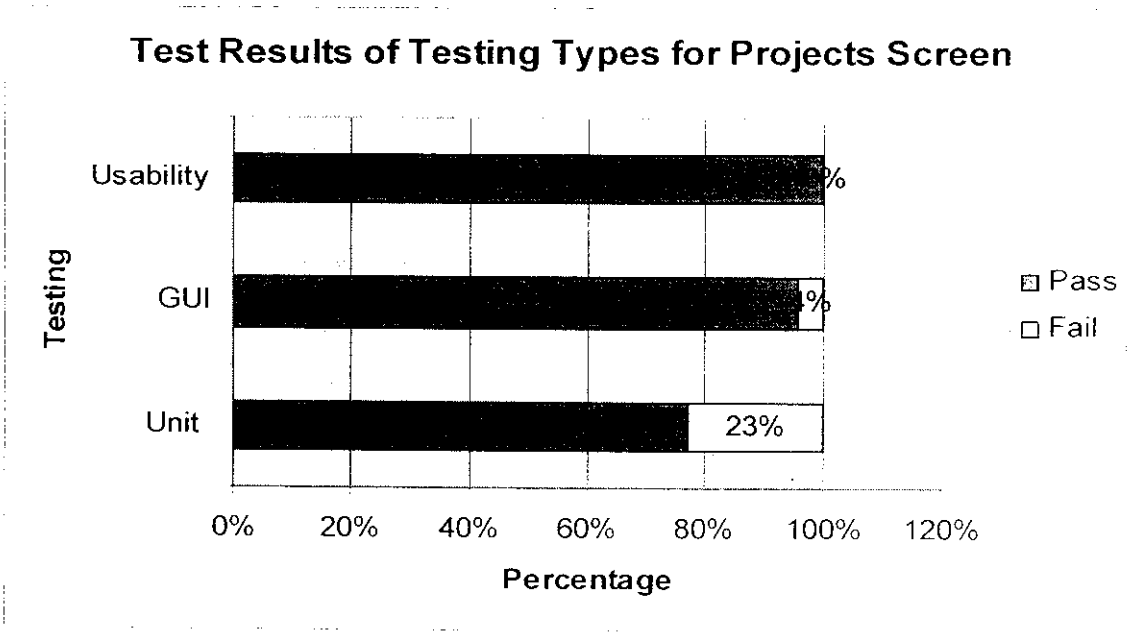


Figure – 4.14 Test results of testing types for Projects screen

TABLE – 4.13 NUMBER OF TEST CASES FOR TASKS SCREEN

FIELDS / STATUS	TASK CONTROL PANEL		CREATE		VIEWLIST		SEARCH		TOTAL
	No.	%	No.	%	No.	%	No.	%	No.
PASS	6	100	35	92	19	86	3	75	63
FAIL	0	0	3	8	3	14	1	25	7
TOTAL	6	100	38	100	22	100	4	100	70

Interpretation:

From the above table we could infer that the total number of test cases found in the 'tasks' form is 70 and the total of pass and fail test cases is 63 and 7 respectively.

Here 'tasks control panel' alone has 100% pass rate. The other fields such as create, view list and search have 8%, 14% and 25% failure rate respectively.

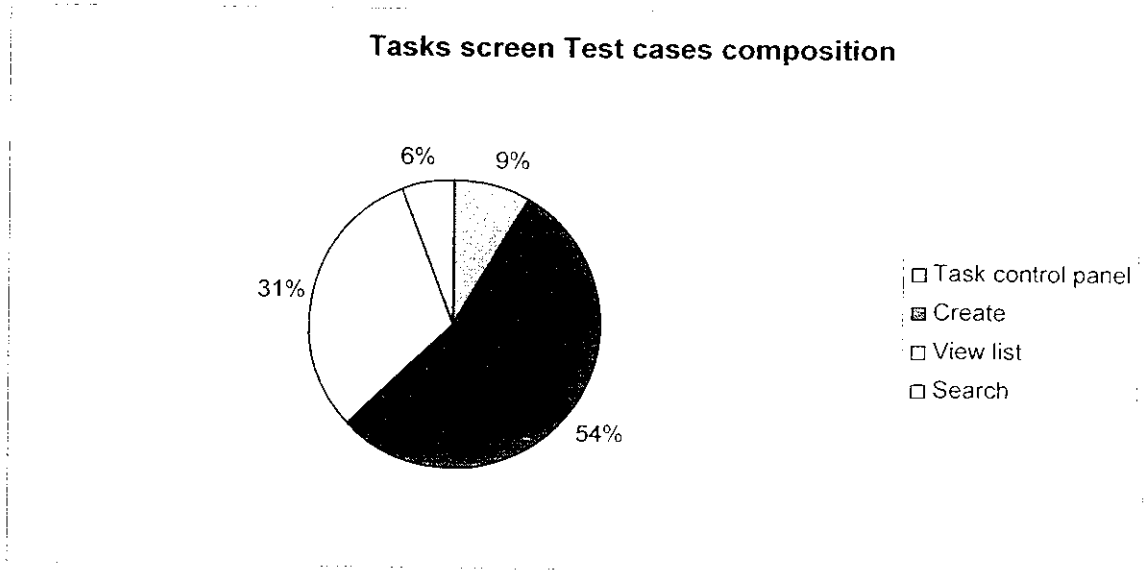


Figure – 4.15 Tasks screen test cases composition

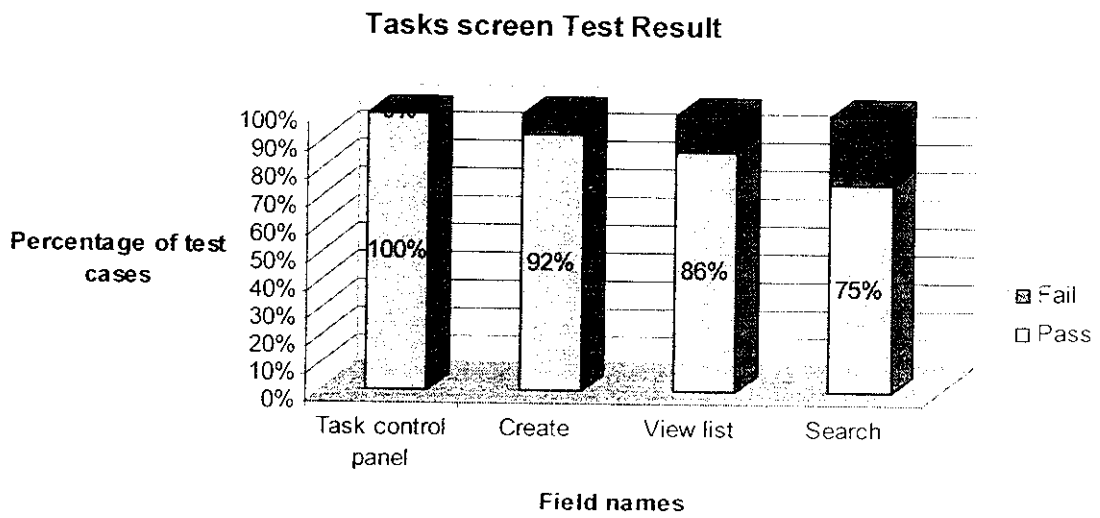


Figure – 4.16 Tasks screen test results

TABLE – 4.14 TEST RESULTS OF TESTING TYPES FOR TASKS SCREEN

TESTING/STATUS	Number of test cases		
	Pass	Fail	Total
Unit	52 (90%)	6 (10%)	58 (100%)
GUI	9 (90%)	1 (10%)	10 (100%)
Usability	2 (100%)	0 (0%)	2 (100%)
Total	63	7	70

Interpretation:

From the above table we could find that the total number of test cases is 70 where the total of pass test cases is 63 and the total of fail test cases is 7. Here the failure rate of 'unit testing' is 10% and 'GUI testing' is 10%.

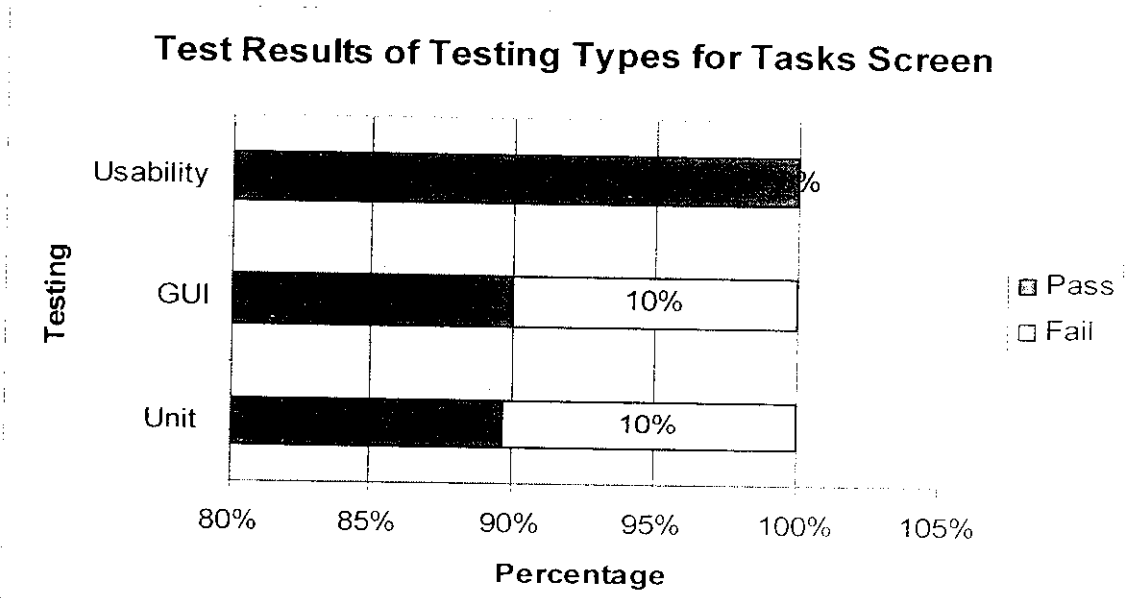


Figure – 4.17 Test results of testing types for Tasks screen

TABLE – 4.15 NUMBER OF TEST CASES FOR PEOPLE SCREEN

FIELDS / STATUS	PEOPLE CONTROL PANEL		CREATE		VIEWLIST		SEARCH		TOTAL
	No.	%	No.	%	No.	%	No.	%	No.
PASS	4	100	37	58	19	95	5	71	65
FAIL	0	0	27	42	1	5	2	29	30
TOTAL	4	100	64	100	20	100	7	100	95

Interpretation:

From the above we could find that the total of pass and fail test cases are 65 and 30 where the total of all test cases is 95. Here only 'people control panel' has 100% pass rate. The other fields such as create, view list and search have 42%, 5% and 29% failure rate respectively. The failure rate of 'create' is more when compared to others.

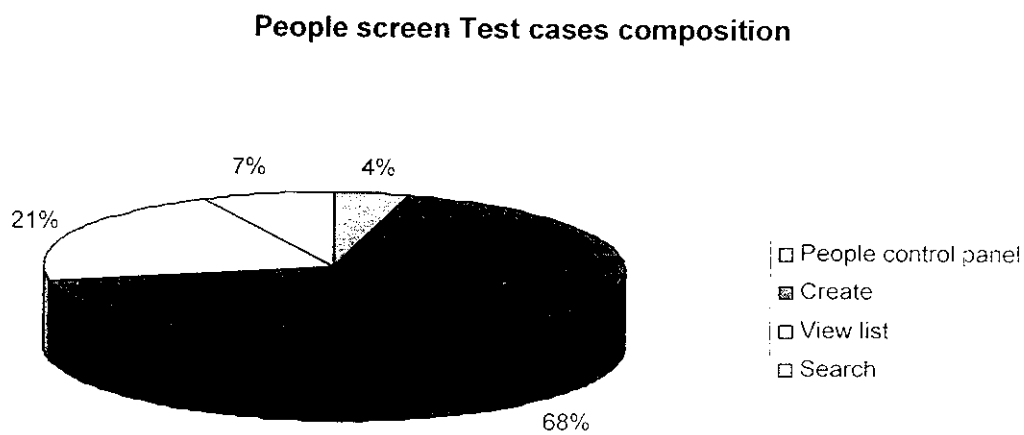


Figure – 4.18 People screen test cases composition

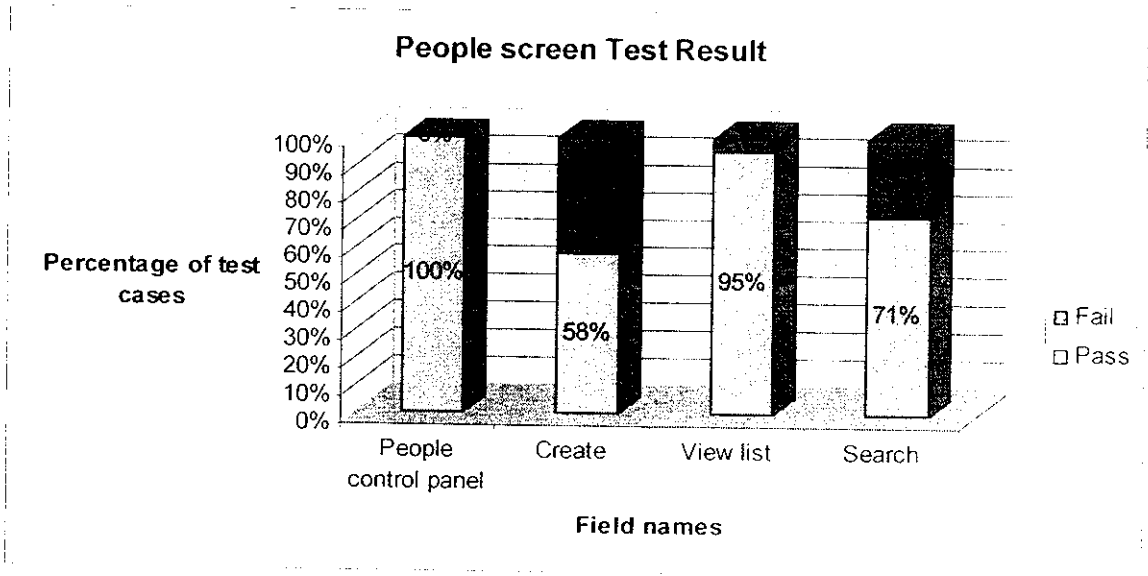


Figure – 4.19 People screen test results

TABLE – 4.16 TEST RESULTS OF TESTING TYPES FOR PEOPLE SCREEN

TESTING/STATUS	Number of test cases		
	Pass	Fail	Total
Unit	49 (63%)	29 (37%)	78 (100%)
GUI	14 (93%)	1 (7%)	15 (100%)
Usability	2 (100%)	0 (0%)	2 (100%)
Total	65	30	95

Interpretation:

From the above table we could find that the total number of test cases is 95 where the total of pass test cases is 65 and the total of fail test cases is 30. Here the failure rate of 'unit testing' is 37% and 'GUI testing' is 7%.

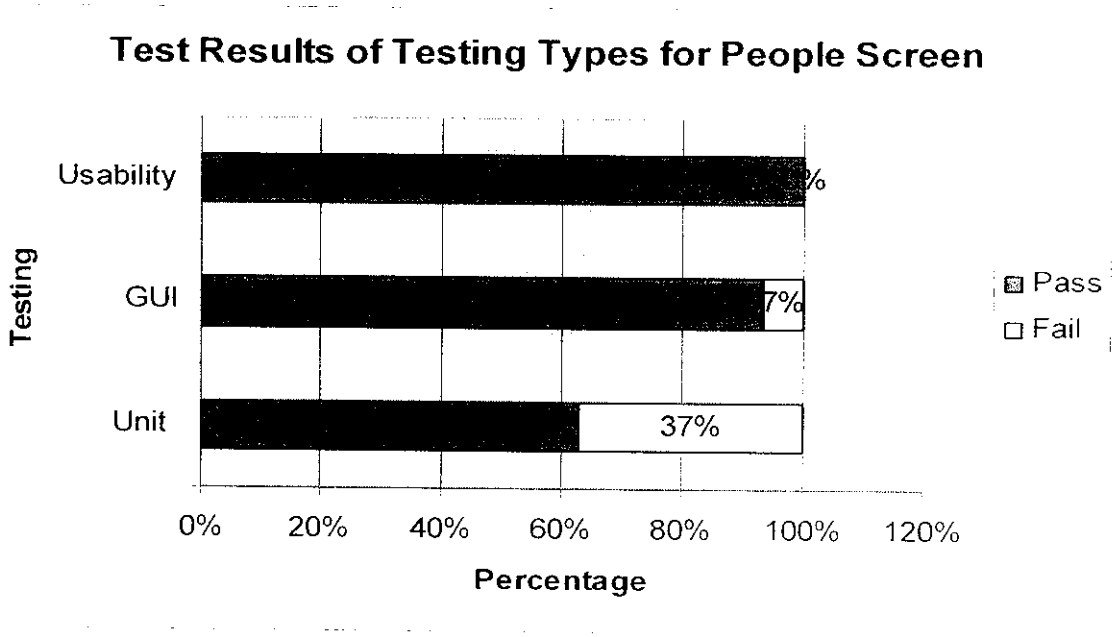


Figure – 4.20 Test results of testing types for People screen

TABLE – 4.17 NUMBER OF TEST CASES FOR NEW PROJECT SCREEN

FIELDS / STATUS	CREATE		VIEWLIST		SEARCH		TOTAL
	No.	%	No.	%	No.	%	No.
PASS	23	85	12	100	4	100	39
FAIL	4	15	0	0	0	0	4
TOTAL	27	100	12	100	4	100	43

Interpretation:

From the above calculated values we could find that the total of pass and fail test cases are 39 and 4 whereas the total of all test cases is 43. Here only "create" has 15% fail

rate. The other fields such as view list and search each have 100% pass rate. The failure rate of this form is less when compared to others.

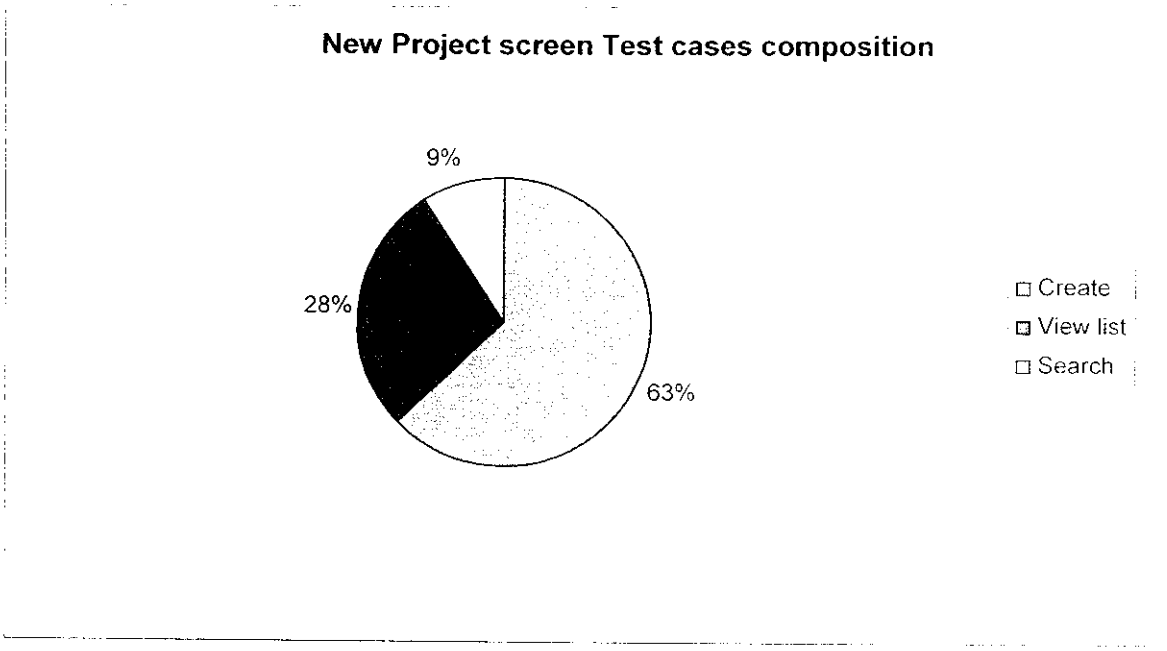


Figure – 4.21 New project screen test cases composition

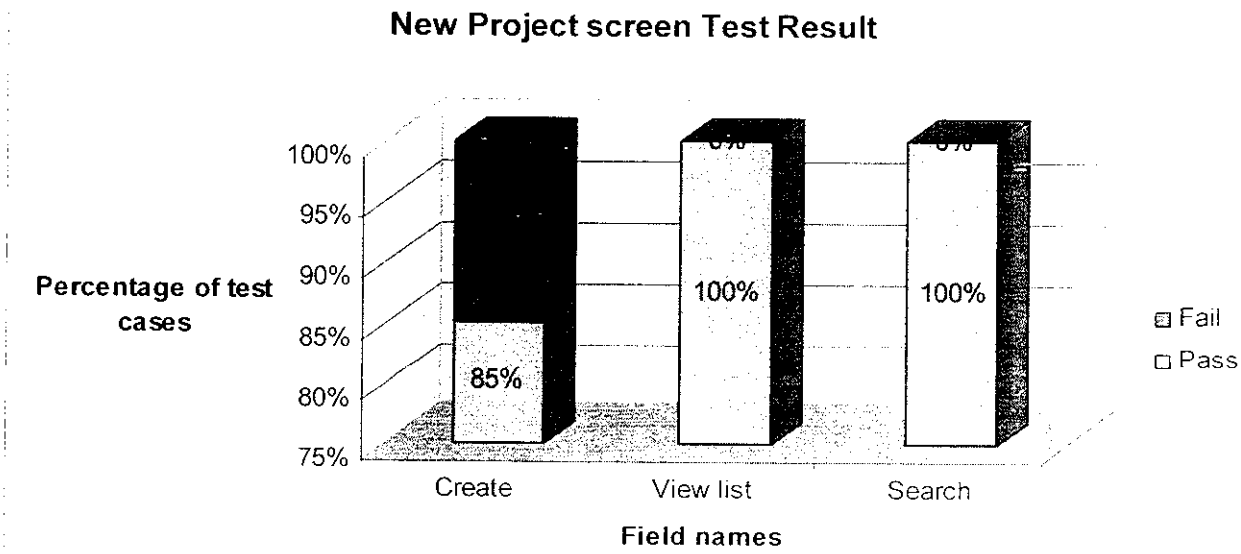


Figure – 4.22 New project screen test results

TABLE – 4.18 TEST RESULTS OF TESTING TYPES FOR NEW PROJECT SCREEN

TESTING/STATUS	Number of test cases		
	Pass	Fail	Total
Unit	26 (90%)	3 (10%)	29 (100%)
GUI	11 (92%)	1 (8%)	12 (100%)
Usability	2 (100%)	0 (0%)	2 (100%)
Total	39	4	43

Interpretation:

From the above table we could find that the total number of test cases is 43 where the total of pass test cases is 39 and the total of fail test cases is 4. Here the failure rate of 'unit testing' is 10% and 'GUI testing' is 8%.

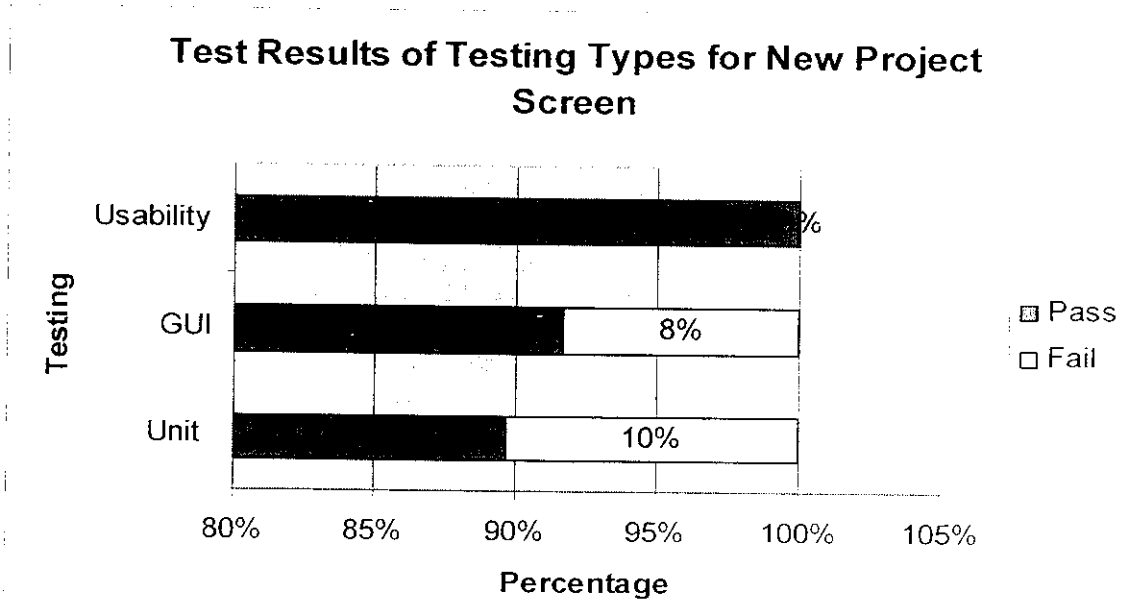


Figure – 4.23 Test results of testing types for New Project screen

TABLE – 4.19 NUMBER OF TEST CASES FOR WORK SCREEN

FIELDS / STATUS	WORK CONTROL PANEL		CREATE		VIEWLIST		SEARCH		TOTAL
	No.	%	No.	%	No.	%	No.	%	No.
PASS	4	100	27	9	17	100	6	75	54
FAIL	0	0	1	4	0	0	2	25	3
TOTAL	4	100	28	100	17	100	8	100	57

Interpretation:

From the above values we could find that the 'work' form has 54 and 3 total number of pass and fail test cases respectively whereas the total of all test cases is 57. Here the failure rate of 'create' is 4% and for 'search' is 25%. The 'work control panel' and 'view list' has 100% pass rate.

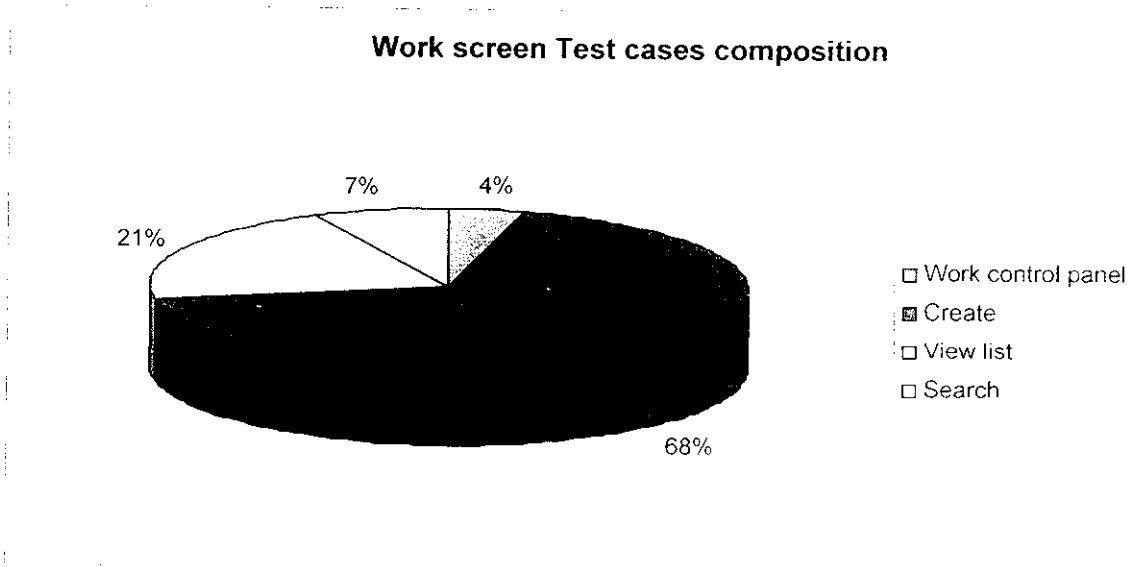


Figure – 4.24 Work screen test cases composition

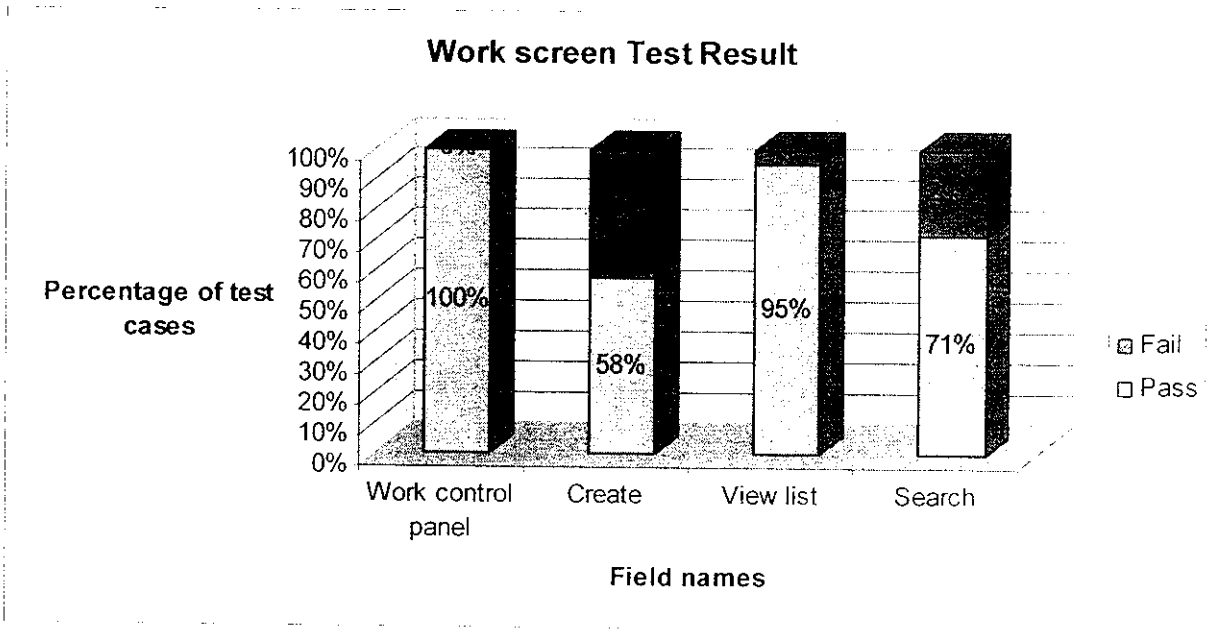


Figure – 4.25 Work screen test results

TABLE – 4.20 TEST RESULTS OF TESTING TYPES FOR WORK SCREEN

TESTING/STATUS	Number of test cases		
	Pass	Fail	Total
Unit	44 (92%)	4 (8%)	48 (100%)
GUI	9 (100%)	0 (0%)	9 (100%)
Usability	1 (100%)	0 (0%)	1 (100%)
Total	54	4	58

Interpretation:

From the above table we could find that the total number of test cases is 58 where the total of pass test cases is 54 and the total of fail test cases is 4. Here the failure rate of 'unit testing' is 8%.

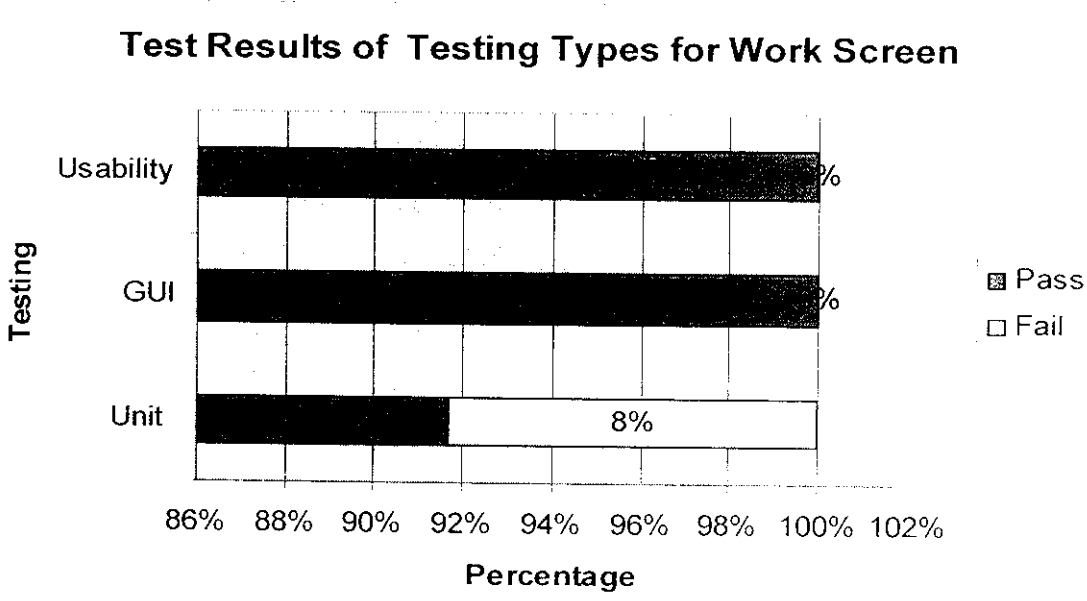


Figure – 4.26 Test results of testing types for Work screen

TABLE – 4.21 NUMBER OF TEST CASES FOR SETTINGS SCREEN

FIELDS / TATUS	ADMIN CONTROL PANEL		OWNER SETTINGS		ADMIN SETTINGS		USERTYPE PERMISSION		MANAGE CLIENTS		MANAGE PROJECTS		GANTT MGT		MANAGE MESSAGES		MANAGE TASKS		MANAGE MILESTONE		MANAGE ASSET TYPE		TOTAL
	No.	%	No.	%	No.	%	No.	%	No.	%	No.	%	No.	%	No.	%	No.	%	No.	%	No.	%	No.
PASS	25	96	7	44	10	77	7	100	25	96	35	97	11	100	57	95	30	88	26	93	8	89	109
FAIL	1	4	9	56	3	23	0	0	1	4	1	3	0	0	3	5	4	12	2	7	1	11	15
TOTAL	26	100	16	100	13	100	7	100	26	100	36	100	11	100	60	100	34	100	28	100	9	100	124

nterpretation:

From the table we could find that the 'settings' form has 109 pass test cases and 15 fail test cases respectively whereas the total of ll test cases is 124. Here the 'user type permissions' and 'Gantt management' alone has 100% pass rate.

TABLE – 4.22 LEGEND FOR SETTTINGS SCREEN

S1 - Admin Control Panel
S2 - Owner Settings
S3 - Admin Settings
S4 - User Type Permission
S5 - Manage Clients
S6 - Manage Projects
S7 - Gantt Management
S8 - Manage Messages
S9 - Manage Tasks
S10 - Manage Milestones
S11 - Manage Asset Type

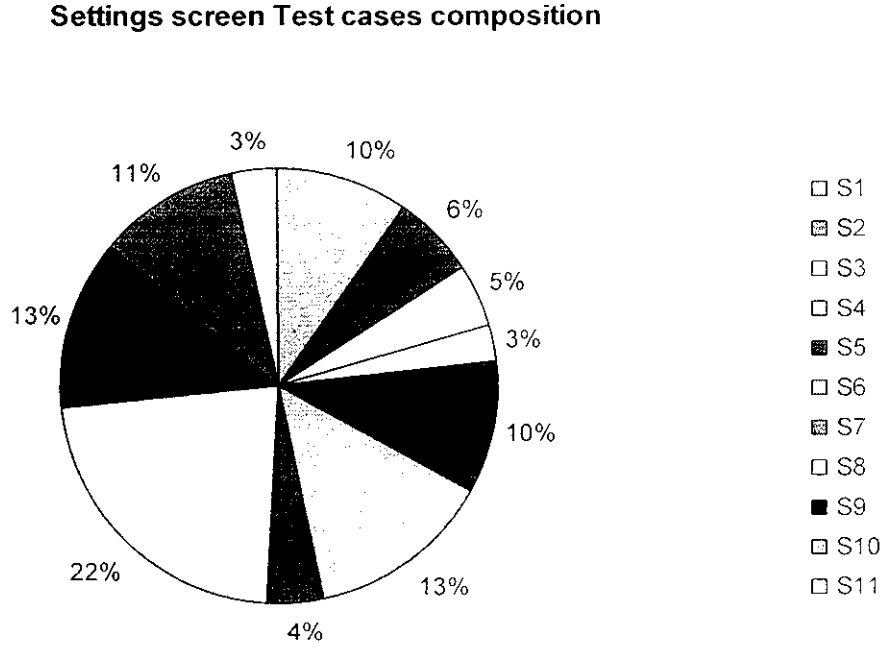


Figure – 4.27 Settings screen test cases composition

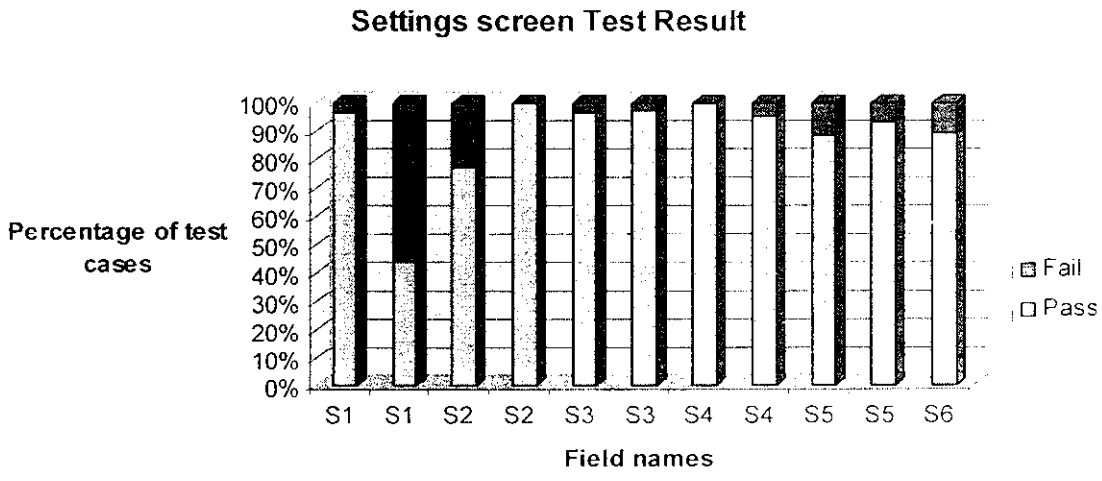


Figure – 4.28 Settings screen test results

TABLE – 4.23 TEST RESULTS OF TESTING TYPES FOR SETTINGS SCREEN

TESTING/STATUS	Number of test cases		
	Pass	Fail	Total
Unit	111 (86%)	18 (14%)	129 (100%)
GUI	106 (95%)	6 (5%)	112 (100%)
Usability	23 (92%)	2 (8%)	25 (100%)
Total	240	26	266

Interpretation:

From the above table we could find that the total number of test cases is 266 where the total of pass test cases is 240 and the total of fail test cases is 26. Here the failure rate of 'unit testing' is 14%, 'GUI testing' is 5% and 'Usability testing' is 8%.

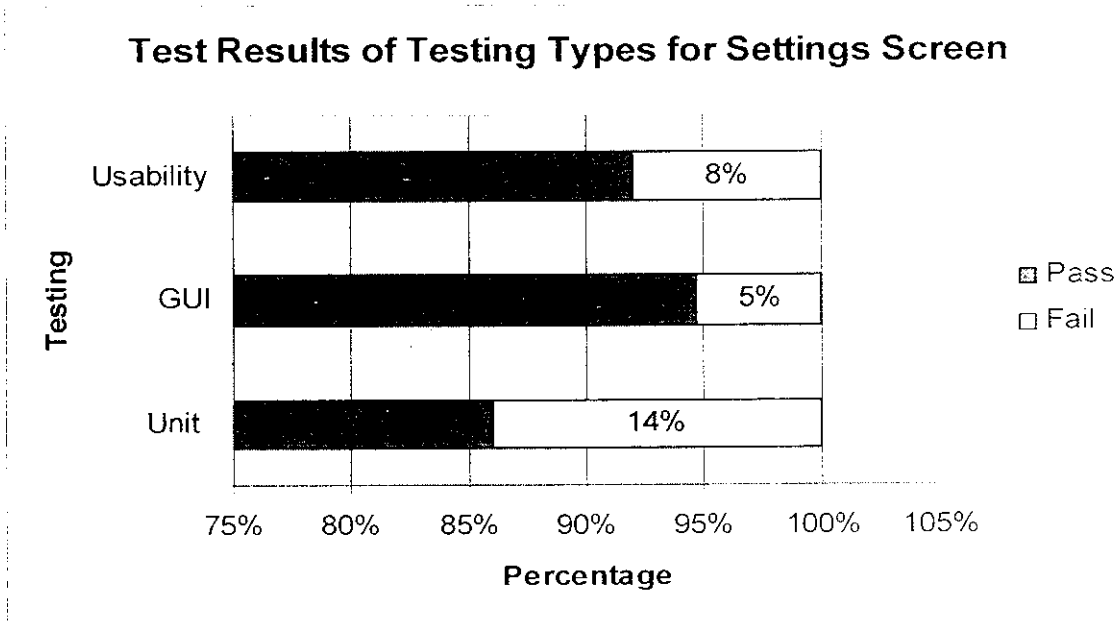


Figure – 4.29 Test results of testing types for Settings screen

Conclusions

CHAPTER – 5

CONCLUSIONS

5.1. FINDINGS:

- From the given data it is known that in all the forms namely Login, Home, Clients, Projects, Tasks, People, New project, My milestones, Settings, Permissions and Work the number of fail test cases is minimum.
- The Unit testing has 23% fail rate. The defects must be rectified to reduce the failure rate.
- The GUI testing has 5% fail rate. But the pass rate is considerable when compared to the fail rate. Therefore it is found that the designing team of the company is good.
- The Usability testing has 6% fail rate. In the same way the test case composition of usability testing is also minimum. It is only 6% of the total testing process. This shows the validity of the company is good.

After considering the above manual testing it is found that implementing project after testing offers the greatest chance of defect detection, and as such the greatest chance of improving overall software quality.

5.2. SUGGESTIONS:

The company must concentrate on the developing programming skills. This will improve the success rate of the testing activity. The programming team is suggested to rework on the bugs of Project Tracking System. They are requested to give proper attention in rectifying the errors. Therefore if proper care is taken by the programming team the minimum error also could be corrected by modifying the form and clearing the defects.

References

REFERENCES

BOOKS AND JOURNAL:

- Glenford J Myers, Tom Badgett, Todd M Thomas, Corey Sandler, "The Art Of Software Testing", John Wiley And Sons Inc, 2004, Second Edition.
- Rex Black, "Managing The Testing Process: Practical Tools And Techniques For Managing Hardware And Software Testing", John Wiley & Sons Inc, 2002, Second Edition.
- Roger S. Pressman, "Software Engineering", Tata Megraw Hill Publishing Private Limited, New Delhi, 2005,Sixth Edition.
- William E. Perry, "Effective Methods For Software Testing", Wiley Computer Publishing, 2001.
- Ieee Journals On Software Maintainence And Computers.

WEBSITES:

- www.softwaredevelopment.ca/bugs.shtml
- http://ieeexplore.ieee.org/xpl/periodicals.jsp
- www.sqatester.com
- www.citeseer.com
- www.altavista.com
- www.yahoo.com
- www.google.com
- www.ibm.com/rational

- www.ibm.com/rational
- http://portal.acm.org/portal.cfm
- www.ebsco.com
- www.amazon.com

Appendices

XSTEPS Project Tracking System

cts

reate a new project

lient

t

chnologies

Projects

Latest Activity

xsteps Technologies - Tech Projects

04-20-2008 Title content

00-00-0000 ~~Title content~~

04-20-2008 Title content

00-00-0000 ~~Title content~~

04-20-2008 Title content

Testing Client - Project Test

✓ 03-10-2008 ~~To Attach File~~

✓ 03-08-2008 To Attach File

lhost ▶ Database: xsteps_pts ▶ Table: xsteps_assets

s_assets

Field	Type	Null	Default	Links to	Comments	MIME
	int(11)	No				
	int(11)	No	0			
	int(11)	No	0			
	int(11)	No	0			
me	datetime	No	0000-00-00 00:00:00			
	int(11)	No	0			
ime	datetime	No	0000-00-00 00:00:00			
	varchar(255)	No				
	varchar(40)	No				
	varchar(40)	No				
	varchar(20)	No				
	text	No				

Type	Cardinality	Field
RIMARY	1	asset_id
NIQUE	1	asset_id
NIQUE	1	asset_id
IDEX	None	asset_id
IDEX	None	asset_id

ge:

age	
56	B
44	B
00	B

Row Statistics:

Statements	Value
Format	dynamic
Rows	1
Row length ø	56
Row size ø	6,200 B
Next Autoindex	117
Creation	Mar 16, 2008 at 11:08 PM
Last update	Mar 16, 2008 at 11:08 PM

Unit Testing

Login

o	S.No	TC.ID	Description	Test Data	Expected Result	Actual Result	Status
	1	L 1.1	Check whether you get login window when you type http://localhost\login in the Url bar		Login window should be displayed	Login window displayed	PASS
	2	L 1.2	Check whether caption of the window is xsteps project management tool		caption must be xsteps project management tool	caption is xsteps project management tool	PASS

ername

o	S.No	TC.ID	Description	Test Data	Expected Result	Actual Result	Status
	3	L 2.1	Check whether the cursor is in the UN field once when you get login window		cursor must be displayed in UN field	cursor is displayed	PASS
	4	L 2.2	Check whether the UN field does not accept more than 40 characters		UN field must accept only 40 characters	UN accepts more than 40 characters	FAIL
	5	L 2.3	UN field should accept only characters and underscore	ab_c	Only characters and underscore must be accepted	underscore accepted	PASS
	6	L 2.4	UN field should accept numbers	abc1	numbers must be accepted	numbers accepted	PASS
	7	L 2.5	UN field should not accept special characters other than underscore	abc&	Only underscore must be accepted and other special characters must	other special characters are accepting	FAIL
	8	L 2.6	UN field should not accept spaces	ab c	spaces must not be accepted in UN field	spaces are accepted	FAIL
	9	L 2.7	Check if UN filed is accepting first character as numeral	1abd	Should not accept, should display an alert	Acceping the data input	FAIL
	10	L 2.8	Check if UN filed is accepting first character as space	abd	Should not accept, should display an alert	Acceping the data input	FAIL

o	TC. ID	Description	Test Data	Expected Result	Actual Result	Status
2	H 2.4	check whether on mousemove on 'no attachment' icon the tooltip is displayed		tooltip must be displayed	tooltip is displayed	PASS
3	H 2.5	check whether on mousemove on 'no comments' icon the tooltip is displayed		tooltip must be displayed	tooltip is displayed	PASS
4	H 2.6	check whether on mousemove on 'no receipt' icon the tooltip is displayed		tooltip must be displayed	tooltip is displayed	PASS
5	H 2.7	check whether on mousemove on 'Edit' icon the tooltip is displayed		tooltip must be displayed	tooltip is displayed	PASS
6	H 2.8	check whether the proper colour is displayed in 'past due' box		check if proper colour is displayed	actual colour is displayed	PASS
7	H 2.9	check whether the proper colour is displayed in 'posted today' box		check if proper colour is displayed	actual colour is displayed	PASS
9	H 2.11	check if on mousemove on 'To attach file' icon the description about the project is displayed		on mousemove the description of project must be displayed	project description is displayed	PASS

profile

No	TC.ID	Description	Test Data	Expected Result	Actual Result	Status
0	H 3.14	check if 'Add Company' button is pressed without entering any text an alert box is displayed		an alert box must be displayed	alert box is displayed	PASS
1	H 3.15	check if 'ok' button is clicked the alert box is closed		alert box must close if 'ok' button is clicked	alert box is closed	PASS
11	H 3.35	check if '@' symbol is not used the email should not be accepted	acd.co.in	an alert must be displayed	alert not displayed	FAIL

page

S.No	TC.ID	Description	Test Data	Expected Result	Actual Result	Status
38	H 1.2	check if logout is clicked it returns to login window		login window must open when logout is clicked	login window opens when logout is clicked	PASS
39	H 1.3	check if tasks is clicked 'My tasks' is displayed		if tasks is clicked 'my task' must be opened	if tasks is clicked 'my task' is opened	PASS
40	H 1.4	check if 'edit your profile' is clicked the profile for editing is displayed		'edit your profile' must be displayed when it is clicked	'edit your profile' is displayed when it is clicked	PASS
41	H 1.5	check if 'Testing client-Project test' is clicked the page is displayed		'Testing client' page must be displayed	page is displayed	PASS
42	H 1.6	check if 'To Attach file' is clicked the page is displayed		'To Attach file' page must be displayed	page is not displayed	FAIL
43	H 1.7	check if 'xsteps Technologies-Tech projects' is clicked the page is displayed		page must be opened	page opens	PASS
44	H 1.8	check if 'Title content' is clicked 'View milestone' page must be displayed		page must be displayed	page is displayed	PASS
45	H 1.9	check if 'home' is clicked the home page is displayed		home page must be displayed	home page displays	PASS
46	H 1.10	check if 'projects' is clicked the home page is displayed		projects page must be displayed	projects page displays	PASS
47	H 1.11	check if 'people' is clicked the home page is displayed		people page must be displayed	people page displays	PASS
48	H 1.12	check if 'admin' is clicked the home page is displayed		admin page must be displayed	admin page displays	PASS

C.ID	Description	Test Data	Expected Result	Actual Result	Status
1.1	Check whether you get login window when you type http://localhost\login in the Url bar		Login window should be displayed	Login window displayed	PASS
1.2	Check whether caption of the window is xsteps project management tool		caption must be xsteps project management tool	caption is xsteps project management tool	PASS

me

C.ID	Description	Test Data	Expected Result	Actual Result	Status
2.1	Check whether the cursor is in the UN field once when you get login window		cursor must be displayed in UN field	cursor is displayed	PASS
2.2	Check whether the UN field does not accept more than 40 characters		UN field must accept only 40 characters	UN accepts more than 40 characters	FAIL
2.3	UN field should accept only characters and underscore	ab_c	Only characters and underscore must be accepted	underscore accepted	PASS
2.4	UN field should accept numbers	abc1	numbers must be accepted	numbers accepted	PASS
2.5	UN field should not accept special characters other than underscore	abc&	Only underscore must be accepted and other special characters must not be	other special characters are accepting	FAIL
2.6	UN field should not accept spaces	ab c	spaces must not be accepted in UN field	spaces are accepted	FAIL
2.7	Check if UN filed is accepting first character as numeral	1abd	Should not accept, should display an alert	Acceping the data input	FAIL
2.8	Check if UN filed is accepting first character as space	abd	Should not accept, should display an alert	Acceping the data input	FAIL
2.9	check whether on pressing enter after typing UN the cursor moves to pwd field		cursor must be displayed in pwd field once enter key is pressed	data is cleared instead of displaying cursor in the pwd field	FAIL
2.10	check if tab key is pressed the cursor is displayed in the pwd field		cursor must be displayed in pwd field once tab key is pressed	cursor is displayed	PASS

rd

C.ID	Description	Test Data	Expected Result	Actual Result	Status
3.1	Check whether the UN field does not accept more than 20 characters		UN field must accept only 20 characters	UN accepts more than 20 characters	FAIL
3.2	pwd field should display everything in asterisk format	***	only asterisks should be displayed	only asterisks is displayed	PASS

Tasks Screen

ntrol panel

TC.ID	Description	Test Data	Expected Result	Actual Result	Status
T 1.1	check if tasks tab is clicked the 'task control panel' is opened		page must be displayed	page displayed	PASS
T 1.2	check if 'My tasks' is clicked the page is displayed		page must be displayed	page displayed	PASS
T 1.3	check if 'Tasks I've sent' is clicked the page is displayed		page must be displayed	page displayed	PASS
T 1.4	check if 'create new tasks' is clicked the 'create task' page is displayed		page must be displayed	page displayed	PASS
T 1.5	check if 'view tasks details' is clicked the 'view task by status' page is displayed		page must be displayed	page displayed	PASS
T 1.6	check if 'search tasks' is clicked the 'search tasks' page is displayed		page must be displayed	page displayed	PASS

TC.ID	Description	Test Data	Expected Result	Actual Result	Status
T 2.1	check whether 'Create task' page is opened as default if 'tasks' tab is clicked		'create task' must open as default	'create task' does not open as default	FAIL
T 2.2	check if a project name is selected the 'For' listbox is enabled		listbox should be enabled	listbox enabled	PASS
T 2.3	check if tab is pressed the cursor must be displayed in the 'Due date' field		cursor must be displayed	cursor not displayed	FAIL
T 2.4	check if a click is made in the 'Due date' field the calendar must be displayed		calendar must be displayed	calendar displayed	PASS
T 2.5	check if past date is selected in the calendar an alert must be displayed		alert must be displayed	alert not displayed	FAIL
T 2.6	check if '>>' is clicked the calendar displays next month		next month must be displayed if '>>' is clicked	next month is displayed	PASS
T 2.7	check if '<<' is clicked the calendar displays previous month		previous month must be displayed if '<<' is clicked	previous month is displayed	PASS
T 2.8	check whether the date format is 'MM-DD-Yr'		'MM-DD-Yr' format must be displayed	date format is displayed properly	PASS
T 2.9	check if 'x' button is clicked the calendar must be closed		calendar must be closed if 'x' is clicked	calendar is closed if 'x' is clicked	PASS

dmin control panel

.No	TC.ID	Description	Test Data	Expected Result	Actual Result	Status
19	S 1.1	check if Home-->Settings is clicked 'Admin control panel' is displayed		page must be displayed	page displayed	PASS
20	S 1.2	check on default the owner settings link is highlighted		owner settings must be highlighted	owner settings not highlighted	FAIL
21	S 1.3	check if tab is clicked 'owner settings' is highlighted		link must be highlighted	link highlighted	PASS
22	S 1.4	check if 'Owner settings' is clicked the owner settings page is displayed		page must be opened	page opened	PASS
23	S 1.5	check if tab is clicked 'Admin settings' is highlighted		link must be highlighted	link highlighted	PASS
24	S 1.6	check if 'Admin settings' is clicked Admin settings page is displayed		page must be opened	page opened	PASS
25	S 1.7	check if shift+tab is clicked 'Owner settings' must be highlighted		link must be highlighted	link highlighted	PASS
26	S 1.8	check if 'People Permissions' is clicked the people permissions page is displayed		page must be opened	page opened	PASS
27	S 1.9	check if the enter key is pressed on the highlighted link the page opens		page must be opened	page opened	PASS
28	S 1.10	check if 'User Type Permissions' is clicked the user type permissions page is displayed		page must be opened	page opened	PASS
29	S 1.11	check if tab is clicked 'User Type permissions' is highlighted		link must be highlighted	link highlighted	PASS
30	S 1.12	check if 'Manage clients' is clicked the manage clients page is displayed		page must be opened	page opened	PASS
31	S 1.13	check if shift+tab is clicked 'User type permissions' must be highlighted		link must be highlighted	link highlighted	PASS
32	S 1.14	check if 'Manage projects' is clicked the manage projects page is displayed		page must be opened	page opened	PASS
33	S 1.15	check if tab is clicked the next link is highlighted		link must be highlighted	link highlighted	PASS
34	S 1.16	check if 'Manage phase' is clicked the manage phase page is displayed		page must be opened	page opened	PASS
35	S 1.17	check if shift+tab is clicked the previous link is highlighted		link must be highlighted	link highlighted	PASS
36	S 1.18	check if 'Gantt management' is clicked the Gantt management page is displayed		page must be opened	page opened	PASS
37	S 1.19	check if 'Manage messages' is clicked the manage messages page is displayed		page must be opened	page opened	PASS

tional Purify pinpoints hard-to-find run time memory errors. It collects coverage, error

d leak data.

sults:

```
Starting Purify'd C:\xampp\xampp-control.exe at 03/30/2008 14:52:17
Starting main
Summary of all memory leaks... {0 bytes, 0 blocks}
Exiting with code 1 (0x00000001)
Program terminated at 03/30/2008 14:52:45
Starting Purify'd C:\xampp\xampp-portcheck.exe at 03/30/2008 14:54:04
Starting main
ABW: Array bounds write in WideCharToMultiByte {1 occurrence}
FMM: Freeing mismatched memory in delete(void *) {16 occurrences}
FMM: Freeing mismatched memory in delete(void *) {1 occurrence}
FMM: Freeing mismatched memory in delete(void *) {1 occurrence}
Summary of all memory leaks... {132 bytes, 3 blocks}
  MPK: Potential memory leak of 50 bytes from 1 block allocated in Shell_NotifyIconW [SHELL
  MLK: Memory leak of 42 bytes from 1 block allocated in LdrLoadDll [NTDLL.dll]
  MPK: Potential memory leak of 40 bytes from 1 block allocated in Shell_NotifyIconW [SHELL
FMM: Freeing mismatched memory in delete(void *) {156 occurrences}
FIM: Freeing invalid memory in LocalFree {36 occurrences}
Exiting with code 0 (0x00000000)
Program terminated at 03/30/2008 14:55:54
Starting Purify'd C:\xampp\xampp_restart.exe at 03/30/2008 14:49:21
Starting main
Summary of all memory leaks... {0 bytes, 0 blocks}
FIM: Freeing invalid memory in LocalFree {36 occurrences}
Exiting with code 0 (0x00000000)
Program terminated at 03/30/2008 14:49:39
Starting Purify'd C:\xampp\xampp_start.exe at 03/30/2008 14:42:09
Starting main
Summary of all memory leaks... {0 bytes, 0 blocks}
FIM: Freeing invalid memory in LocalFree {36 occurrences}
  Address 0x001535e8 points into a HeapAlloc'd block in unallocated region of the default h
  Location of free attempt
    LocalFree      [C:\WINDOWS\System32\KERNEL32.dll]
    IsValidLocale  [C:\WINDOWS\system32\kernel32.dll]
Exiting with code 0 (0x00000000)
```