Computerized Internal Evaluation Test

A project report submitted in partial fulfillment requirements
for the award of Degree of

MASTERS OF COMPUTER APPLICATIONS (2015-2018)

Department of Computer Science and Application
Panjab University Regional Centre,
Sri Muktsar Sahib

Submitted by: Submitted
to:

Gurinder Kaur **Dr. Manish Kumar**
MCA 6th sem.
1504

Project Undertaken At :

Table of Content

1. Introduction

1.1 Introduction:

This Web Application provides facility to conduct online examination world wide.

It saves time as it allows number of students to give the exam at a time and displays the results as the test gets over, so no need to wait for the result. It is automatically generated by the server.

Administrator has a privilege to create, modify and delete the test papers and its particular questions.

User can register, login and give the test with his specific id, and can see the results as well.

1.2 Goal or aim of the Project:

Computerized Internal Evaluation Test project intends to provide and integrate and automated software of any institution that wishes to revolutionize the management of student service through computerization. The aim of this project is to computerize the entire process to conducting internal examination of students.

1.3 Usefulness of Project:

It is web application developed using PHP. **Computerized Internal Evaluation Test (CIET)** PHP Final Year project built over the open source platform, automates the process of conducting internal examinations for engineering students. It has log-in form where it can accept user login information (user name and password), then authenticates the user, upon the successful login authentication, it gives those privileges for which he/she is authorized for. Here the user can be a student or a head of the department (admin). The student uses Computerized Internal Evaluation Test (CIET**)** to write an exam and views his/her examination result. Head of The Department uses Computerized Internal Evaluation Test (CIET**)** to view student’s performance.

2. Modules

Computerized Internal Evaluation Test project intends to provide and integrate and automated software of any institution that wishes to revolutionize the management of student service through computerization. The aim of this project is to computerize the entire process to conducting internal examination of students

There are two modules:

1) ***Student***
2) ***Administrator***

2.1 Student Module

- Can login
- Can give exam
- Can get result
-

2.2Admin Module

- Can authorize the student
- Can manage the student
- Can view result of student

Chapter 3 Feasibility Study

Feasibility study is done so that an ill-conceived system is recognized early in definition phase. During system engineering, however, we concentrate our attention on four primary areas of interest. This phase is really important as before starting with the real work of building the system it was very important to find out whether the idea thought is possible or not.

3.1Types of Feasibility

3.1.1 Economic Feasibility:

An evaluation of development cost weighted against the ultimate income or benefit derived from the developed system.

My project "CIET" is economical feasible as it doesn't cost more. It can be used on any computer platform without the installation or any other extra resources.

Software Cost:

Wamp : Free
Notepad++ : Free

3.1.2 Technical Feasibility:

It is a measure of the practically of a specific technical solution and the availability of technical resources and expertise .The proposed system uses HTML , CSS ,JavaScript as front end. Hardware used in this project are- – i3 processor 2.30 GHz, 4 GB RAM, 400 GB hard disk. The technical feasibility is in employing computers to the organization. the organization has not technical difficulty in adding this system.

Tools Used

- **Tools used at Front End**

✓ HTML

✓ CSS

- **Tools used at Back End**

✓ PHP
✓ AJAX
✓ MySql Database

3.1.3 Operational Feasibility:

A study about the operational aspects of the system. The project is operationally feasible. This project is made for the convenience of the admin . It will greatly reduce the inaccuracy of data ,easy modification,. So because of the above stated advantages the user of the system will not be reluctant. So my project "CIET" performs all the operations, Such as conduct online tests , students get feedback . So it has operational feasibility.

3.1.4 Other Feasibility Dimensions:

3.1.4.1 Time based study:- This is an analysis of the time required to achieve

Time Duration

For study	**10 days**
Designing	**15 days**
For development	**50 days**
Testing	**05days**
Total Time	**80 days**

3.1.4.2 Legal feasibility:

Determine whether the proposed system conflicts with legal requirements. E.g. a data processing system must comply with the local Data Protection Acts .The proposed system is legally feasible.

4.Design

4.1Data Flow Diagrams:

Data Flow Diagram is the Graphical description of the system data and how the process transform the data. Data Flow Diagram depicts information flow, the information flow and the transforms that are applied as data moves from the input to output. It is the starting point of the design place that functionally decomposes the requirement specification functionally decomposes the requirement specification down to the level of details. Then a DFD describe what data flow logical rather than how they are processed.

Unlike detail flowchart, Data Flow Diagrams to supply details description of the modules but graphically describes a systems data and how the data interacts with the system to construct a Data Flow Diagrams we use:

- Arrows
- Circles
- Squares

An arrow identifies the data flow in motion.it is a pipeline through which information is flown file the rectangle in the flowchart. A circle stands for process that convert data information. As open ended but represents a data stores data at rest or a temporary repository of data. A Square defines a source or destination of system data.
Rule for constructing a Data Flow Diagram

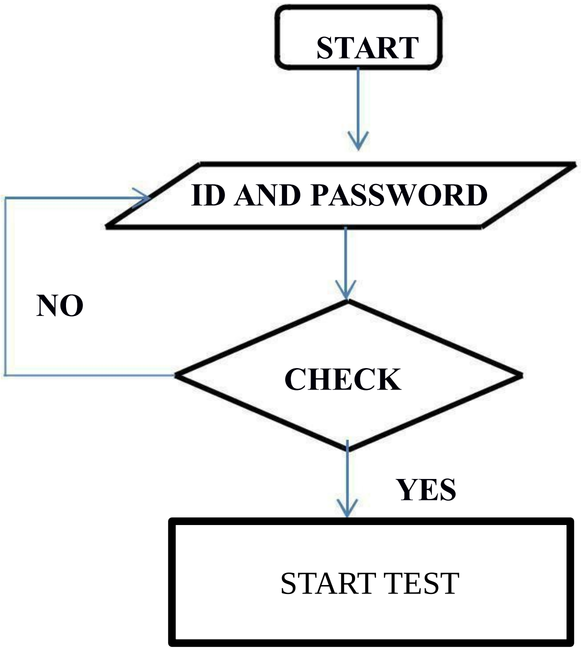
- Arrows should not cross each other.
- Squares, circles and file must be a name.
- Decomposed data flow squares and circles can be same names.
- Choose meaningful names for data flow.
- Draw all data flows around the outside of diagram.

Uses case Diagram

A use case diagram at its simplest is a representation of a users interact with the System and depicting the specification of a use case. A use case diagram can party the different type of uses of a system and various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

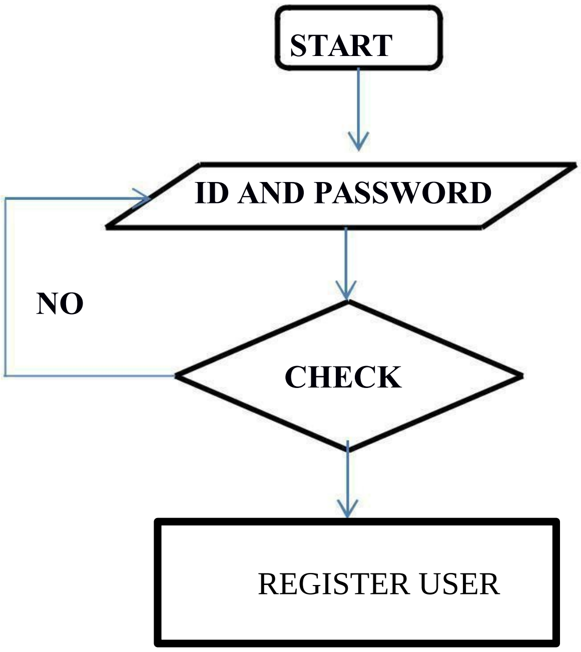
4.2Flow Chart:

4.2.1 For Student



4.1.2 For ADMIN

Flowchart



5.Tool and Techniques to be used

Language

- PHP

Font End

- HTML
- CSS

Back End

- **MySQL**

6.Requirements

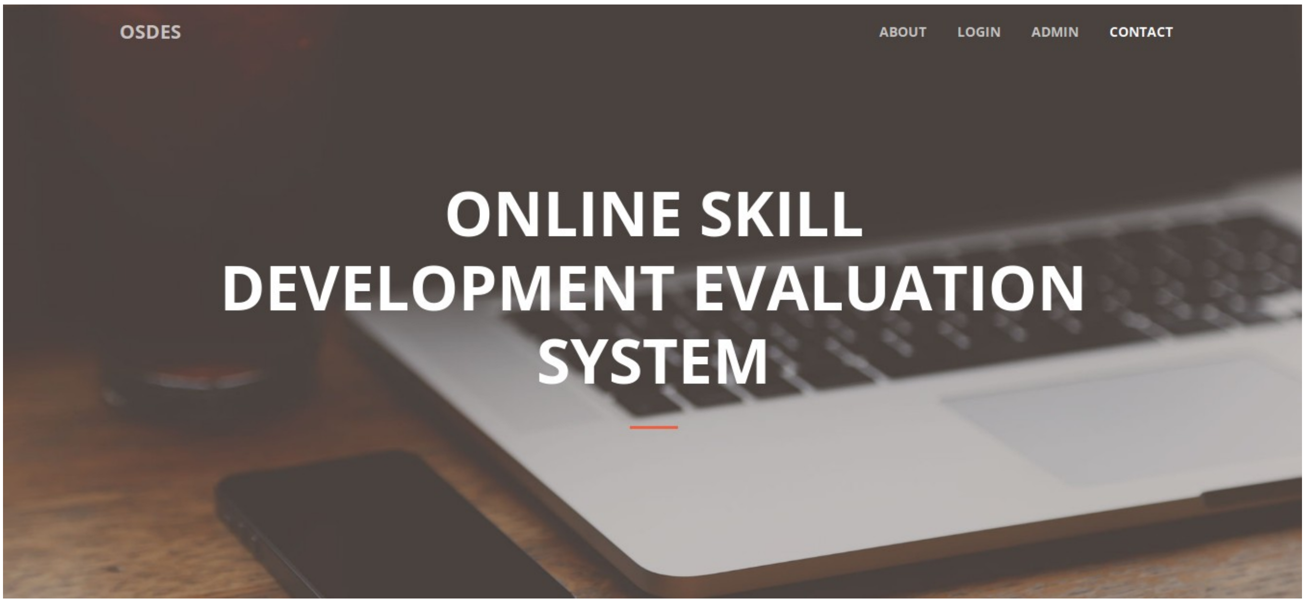
5.1 Hardware Requirements

•	Operating System	Window 7
•	Hardware	Processor P3, p5
•	RAM	Minimum RAM 512 MB
•	Hard disk	Minimum 80 GB

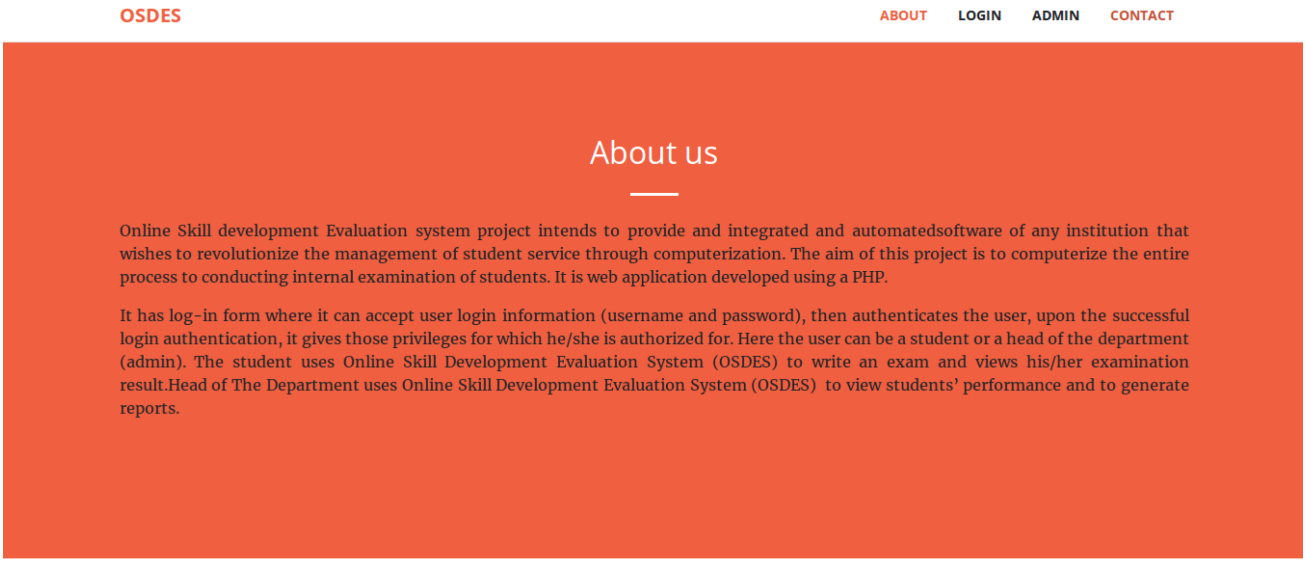
5.2 Software Requirements

•	Server	Xamp Server(Apache Tomcat)
•	Front End	MYSQL
•	Back End	Minimum 80 GB
•	IDE	Sublime Text
•	Browser	Internet Explorer ,Safari ,Mozilla

7.Implementation



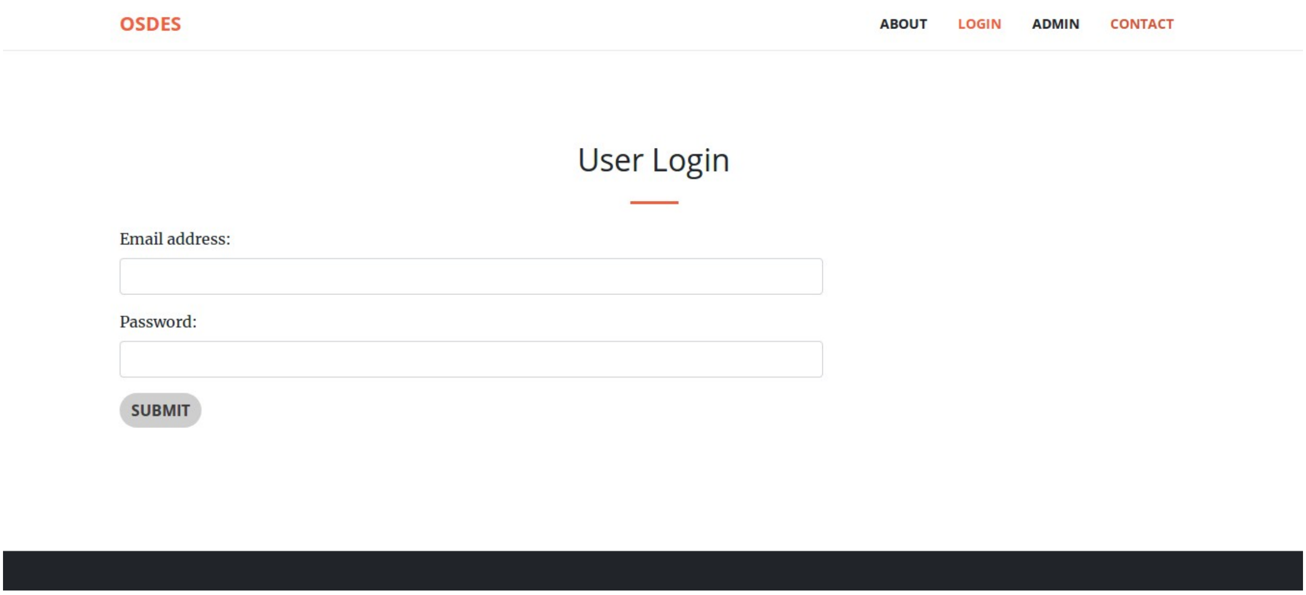
ONLINE SKILL DEVELOPMENT EVALUATION SYSTEM



About us

Online Skill development Evaluation system project intends to provide and integrated and automatedsoftware of any institution that wishes to revolutionize the management of student service through computerization. The aim of this project is to computerize the entire process to conducting internal examination of students. It is web application developed using a PHP.

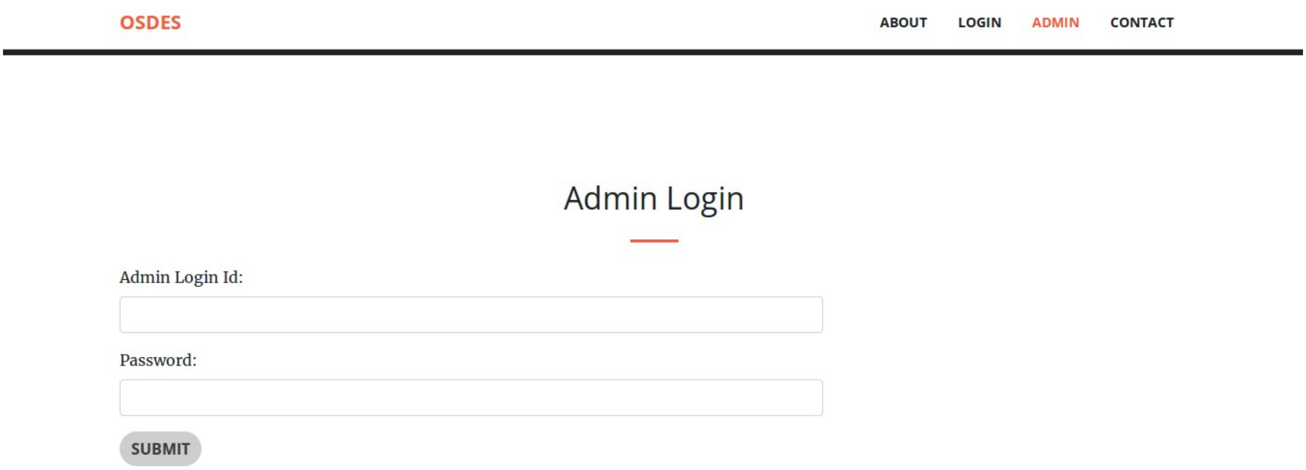
It has log-in form where it can accept user login information (username and password), then authenticates the user, upon the successful login authentication, it gives those privileges for which he/she is authorized for. Here the user can be a student or a head of the department (admin). The student uses Online Skill Development Evaluation System (OSDES) to write an exam and views his/her examination result.Head of The Department uses Online Skill Development Evaluation System (OSDES) to view students' performance and to generate reports.



User Login

Email address:

Password:



Admin Login

Admin Login Id:

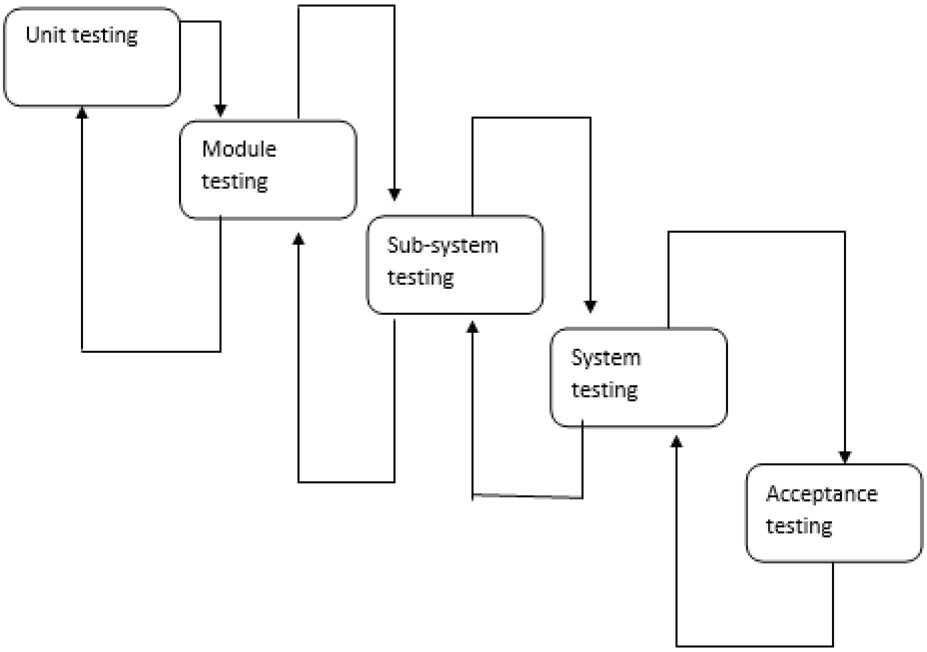
Password:

8.Testing

8.1 Testing Procedure:

System testing

Here the System testing involved is the most widely used testing process consisting of five stages as shown in the figure. In general, the sequence of testing activities is component testing, integration testing, and then user testing. However, as defects are discovered at any one stage, they require program modifications to correct them and this may require other stages in the testing process to be repeated.



(Component testing) (Integration testing) (User testing)

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of the software. The results of testing are used later on during maintenance also.

Testing is vital to the success of the system. System testing makes a logical assumption that if the parts of the system are correct, the goal will be successfully achieved. In adequate testing or non-testing leads to errors that may not appear until months or even years later:

This creates two problems:

1. The time lag between the cause and the appearance of the problem.
2. The time interval effect of the system errors on files and the records on the system.

A small error can conceivably explode into a much larger problem. Effective testing early in the process translates directly into long term cost savings from a reduced number of errors.

Another reason for system testing is its utility as a user oriented vehicle before implementation. The best program is worthless if it does not meet the user requirements. Unfortunately, the user's demands are often compromised by efforts to facilitate program or design efficiency in terms of processing time or design efficiency.

Thus in this phase we went to test the code we wrote. We needed to know if the code compiled with the design or not? Whether the code gave the desired outputs on given inputs? Whether it was ready to be installed on the user's computer or some more modifications were needed?

Through the web applications are characteristically different from there software counterparts but the basic approach for testing these web applications is quite similar. These basic steps of testing have been picked from software engineering practices. The following are the steps, we undertook:

1. The content of the Internet site is reviewed to uncover Content errors. Content Errors covers the typographical errors, grammatical errors, errors in content consistency, graphical representation and cross referencing errors.
2. The design model of the web application is reviewed to uncover the navigation errors. Use cases, derived as a part of the analysis activity allows a web designer to exercise each usage scenario against the architectural and navigational design. In essence these non-executable tests help to uncover the errors in navigation.
3. When web applications are considered the concept of unit changes. Each web page encapsulates content navigation links, content and processing elements (Forms, Scripts, Servlets and JSP's as in our case). It is not always possible to test each of these individually. Thus is the base of the web applications the unit to be considered is the web page. Unlike the testing of the algorithmic details of a module the data that flows across the module
4. interface, page level testing for web applications is driven by content, processing and links encapsulating the web page.
5. The Assembled web application is tested for overall functionality and content delivery. The various user cases are used that test the system for errors and mistakes.
6. The Web application is tested for a variety of environmental settings and is tested for various configurations and upon various platforms.
7. The modules are integrated and integration test are conducted.
8. Thread based testing is done to monitor the regression tests so that the site does not become very slow is a lot of users are simultaneously logged on.
9. A controlled and monitored population of end users tests Intranet application, this all comprises of the User Acceptance Testing.

Because web application evolves continuously, the testing process is an ongoing activity, conducted by web support staff in our case the Organization's IS people who will finally update and manage the application.

PSYCHOLOGY OF TESTING

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't work. Testing is the process of executing a program with the intent of finding errors.

TESTING OBJECTIVES:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

- ✓ Testing is a process of executing a program with the intent of finding an error.
- ✓ A successful test is one that uncovers an as yet undiscovered error.
- ✓ A good test case is one that has a high probability of finding error, if it exists.
- ✓ The tests are inadequate to detect possibly present errors.
- ✓ The software more or less confirms to the quality and reliable standards.

LEVELS OF TESTING

In order to uncover the errors present in different phases, we have the concept of levels of testing. The basic levels of testing are

Client Needs Acceptance Testing

Requirements System Testing

Design Integration Testing

Code Unit Testing

Unit testing

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications, testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of a module. Each module has been tested by giving different sets of inputs. When developing the module as well as finishing the development, the module works without any error. The inputs are validated when accepting them from the user.

Integration Testing

After unit testing, we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions.

In this project the main system is formed by integrating all the modules. When integrating all the modules I have checked whether the integration effects working of any of the services by giving different combinations of inputs with which the two services run perfectly before Integration.

SYSTEM TESTING

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see if software meets its requirements.

Here entire project “Company Mailer ” has been tested against requirements of project and it is checked whether all requirements of project have been satisfied or not.

ACCEPTANCE TESTING

Acceptance Testing is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system; the internal logic of program is not emphasized.

Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

WHITE BOX TESTING

This is a unit testing method, where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors.

I tested step wise every piece of code, taking care that every statement in the code is executed at least once. The white box testing is also called Glass Box Testing.

I have generated a list of test cases, sample data, which is used to check all possible combinations of execution paths through the code at every module level.

White-box test focuses on the program control structure. Test cases are derived to ensure that all statement in the program control structure. Test cases are derived to ensure that all statement in the program control structure. Test cases are derived to ensure that all statement in the program has been executed at least once during testing and that all logical conditions have been exercised. Basis path testing, a white box technique, makes use of program graphs (or graph matrices) to derive the set of linearly independent test that will ensure coverage. Condition and data flow testing further exercising degrees of complexity.

BLACK BOX TESTING

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details at statement level. Here the module will be treated as a block that will take some input and generate output. Output for a given set of input combinations are forwarded to other modules.

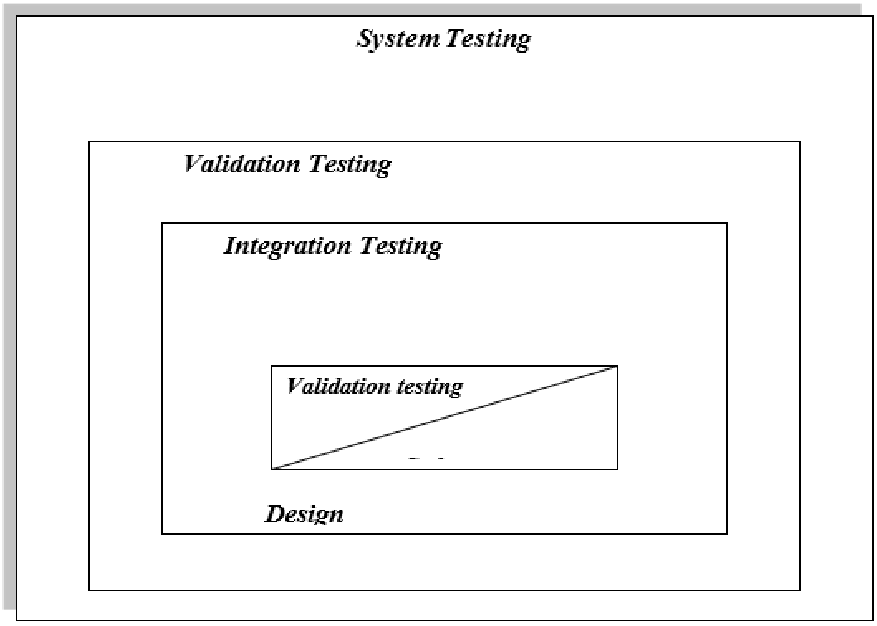
Black-box test are designed to uncover errors functional requirement without regard to the internal workings of a program. Black-box testing techniques focus on the information domain of the software, deriving test cases by partitioning the input and output domain of a program in manner that provides through test coverage. The black-box test is used to demonstrate that software functions are operational, that input is properly produced, and that the integrity of external information are maintained. A black-box test examines some fundamental aspect of a system with little or no regard for the integral logical structure of the software.

Graph based testing methods explore the relationship between and behavior of program objects. Equivalence partitioning divides the input classes of data are likely to exercise specific software function. Boundary values analysis probes the program's ability to handle data at the limits of acceptability.

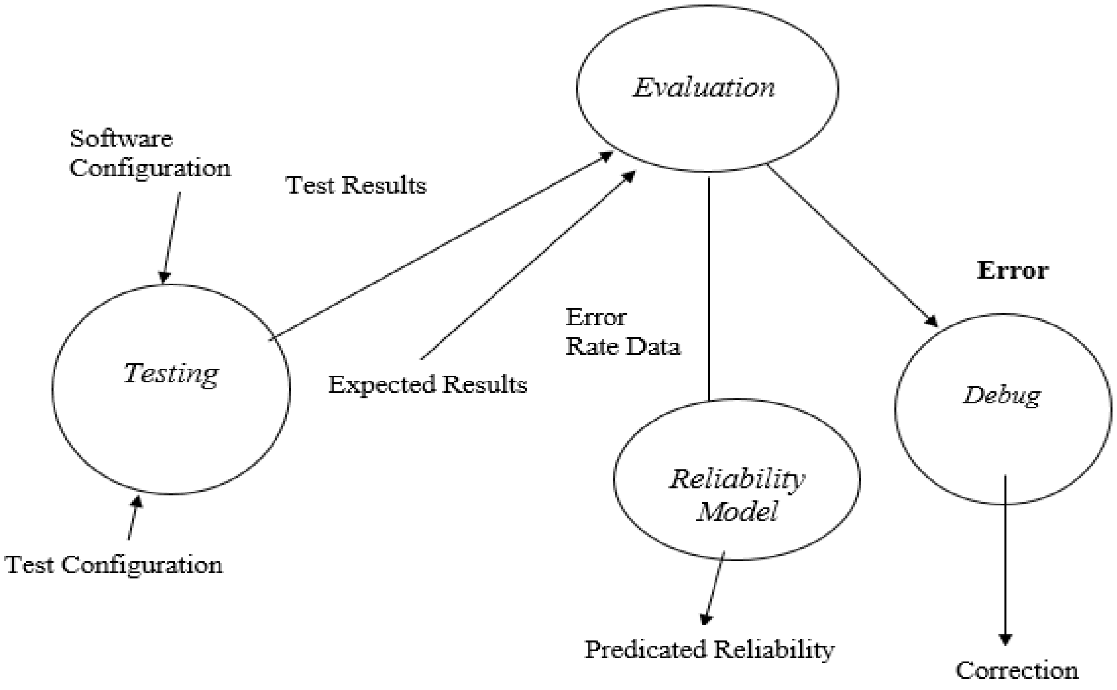
TEST INFORMATION FLOW

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vortex of the spiral and, concentrates on each unit, component of the software as implemented in source code. Testing progresses moving outward along the spiral to integration testing, where the focus is on designed the construction of the software architecture. Taking another turn outward on spiral, we encounter validation testing, where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole. To test computer software, we spiral out along stream lines that broaden the scope of testing with each turn.

Considering the process from a procedural point of view, testing within the context of software engineering is actually a series of four steps that are implemented sequentially. The steps are shown in Figure. Initially, tests focus on each component individually, ensuring that it functions properly as unit. Hence, the name unit testing. Unit testing makes heavy use of white-box testing techniques, exercising specific paths in module's control structure to ensure complete coverage and maximum error detection.



INFORMATION FLOW OF DATA FOR TESTING



Scope

A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot established that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions.

The scope of software testing often includes examination of code as well as execution of that code under various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of the software development, a testing organization may be different from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

Functional versus non-functional testing

Functional Testing refers to the activities that verify a specific action or function of the code. These are usually found in the code requirements documentation, although some development methodologies work from use cases or user stories. Functional tests tends to answer the question of “can the user do this” or “does this particular feature work.”

Non-functional Testing refers to aspects of the software that may not be related to a specific function or user action, such as scalability or other performance, behavior under certain constraints or security. Testing will determine the flank point, the point at which extremes of scalability or performance leads to unstable execution.

Chapter 9 Advantages & Disadvantages

9.1) Advantages

- Security of data
- Ensure data accuracy
- Minimize manual data entry
- Minimum time required

9.2) Disdvantages

- 1.Internet connection is must .
- **2.** User must have an account registered on the website for test.

10.Maintence

Maintenance or enhancement can be classified as: -

- ✓ **Corrective**
- ✓ **Adaptive**
- ✓ **Perfective**

Corrective maintenance means repairing processing or performance failures or making changes because of previously uncorrected problems or false assumptions.

Adaptive maintenance means changing the program function.

Perfective maintenance means enhancing the performance or modifying the program(s) to respond to the user's additional or changing needs.

Maintenance is actually the implementation of the post implementation review plan. As important as it is, many programmers and analysts are reluctant to perform or identify themselves with the maintenance effort. There are psychological, personality and professional reasons for this. In any case, a first class effort must be made to ensure that software changes are made properly and in time to keep the system in tune with user specifications.

Maintenance is costly. One way to reduce maintenance costs is through maintenance management and software modification audits. Software modification consists of program rewrites system level updates, and re-audits of low ranking programs to verify and correct the soft spots.

The outcome should be more reliable software, a reduced maintenance backlog, and higher satisfaction and morale among the maintenance staff.

In "CIET", care will be taken towards maintenance; Loop Holes will be eradicated from the system from time to time according to changing requirements with lesser cost

Evaluation

The evaluation phase ranks project proposals and determines the one best suited, Evaluation of the system is performed to identify its strengths and weaknesses. The actual evaluation can occur along any of the following dimensions:

- ❖ Operational Evaluation: Assessment of the manner in which the system functions, including case of use, response time, overall reliability and level of utilization.
- ❖ Organizational Impact: Identification and measurement of benefits to the organization in such areas as financial concerns, operational efficiency and competitive impact.
- ❖ User Manager Assessment Evaluation of the attitudes of senior and user manager within the organization, as well as end-users.

Development Performance: Evaluation of the development process in accordance with such yardsticks as overall development time and effort, conformance to budgets and standards and other project management criteria

Future Scope of the Project

- ❖ The project has been completed successfully and tested and no issue involved till now. Currently the system is worked, but there is a requirement to improve the system.

❖ In general, scope creep happens when new elements are added to a project that’s already been approved, but no consideration is given to increasing the budget, adding more time to the schedule and/or adding more resources to compensate for the revised project

❖ I evaluated some applications that provide similar features to the user.

Online seen
Chat facility
Sticky notes

4.1 Future Scope for Modification

There is one drawback of this system is the database access problem. The database is not much secured as it still has the risk of illegal access.The future enhancement would thus include an update file to be managed properly and it should be signed and approved by the administrator regularly. Regular backups are needed for proper functioning of the system. The backups can be daily, weekly or after every 15 days or after a month. This ensures that if the functioning of any module stops then there is recovery available for it. This could be done by the ODL Administrator.

There is a need of Rollback and Commit functions to make and save the updates made in the database should be there. In case of any inconsistency in the database, the database can be brought to a consistent state without data lose.

This is a first step for the good cause and there is many more ideas revolving in mind that I hopefully include in future. In near future this project can be extended and make it more useful.

11.References/Bibliography

- ✓ www.w3schools.com/php
- ✓ www.tutorialspoint.com
- ✓ php.net/manual/en/index.php
- ✓ **http://www.google.com**