A PROJECT REPORT ON

ONLINE EXAMINATION SYSTEM

SUBMITTED BY

PRANJAL SAIKIA

(Session 20014-2015)

Branch- M.Sc (IT), ID-CS13MS0316

(Session 2013-2015)

Submitted in partial fulfillment for the award of the Master Degree of

Master of Science in Information Technology

KAZIRANGA UNIVERSITY

Jorhat-785006, Assam.

Organizational Guide

Mr. Bharadwaj Choudhury
Asst. Dean, School of Computing Sciences.
ACKNOWLEDGMENT

“Task successful” makes everyone happy. But the happiness will be gold without glitter if we didn’t state the persons who have supported us to make it a success.

Success will be crowned to people who made it a reality but the people whose constant guidance and encouragement made it possible will be crowned first on the eve of success.

This acknowledgement transcends the reality of formality when we would like to express deep gratitude and respect to all those people behind the screen who guided, inspired and helped me for the completion of our project work.

I consider myself lucky enough to get such a good project. This project would add as an asset to my academic profile.

I would like to express my thankfulness to my project guide, Mr. Bharadwaj Choudhury ASST. Dean of Computing Sciences, Kaziranga University, Jorhat- Assam, for his constant motivation and valuable help through the project work, and I express my gratitude to Tripti paul, for his constant guidance and co-operation throughout the project.

I also extend my thanks to my Team Members for their co-operation during my course.

Finally I would like to thanks my friends for their co-operation to complete this project.

PRANJAL SAIKIA

School of computing Sciences

Branch- M.Sc (IT)

ID-CS13MS0316
DECLARATION

I hereby declare that the project work entitled “ONLINE EXAMINATION SYSTEM” submitted on the partial fulfillment for the 2nd Year of Master of Sciences (Information Technology) from Kaziranga University and approved by University Grants Commission (UGC) has been carried out by our self under the academic guide and, for whom the project is going to implement at Kaziranga University. No part of the project has been submitted for any degree or diploma of any institution previously.

PRANJAL SAIKIA

4th Semester, M.Sc (IT)

Kaziranga University
Table of Contents

• INTRODUCTION
• OBJECTIVES OF PROJECT
• TOOLS, PLATEFORM, HARDWARE AND S/W REQUIREMENT
• SYSTEM ANALYSIS
  1. STRUCTURED ANALYSIS
  2. OBJECT ORIENTED ANALYSIS
• IDENTIFICATION OF NEED
• PRELIMINARY INVESTIGATION
  1. REVIEWING DOCUMENTS OF ORGANISATION
  2. ONSITE OBSERVATION
  3. CONDUCTING INTERVIEW
• FEASIBILITY STUDY
  1. Technical Feasibility
  2. Economical Feasibility
  3. Operational Feasibility
• Project planning
  1. Gantt chart
  2. PERT chart
• Quality Assurance Plan
• Project Scheduling
• Planning Tools
• SOFTWARE REQUIREMENT AND SPECIFICATION
• SURVEY OF TECHNOLOGY
  1. What is ASP.Net?
  2. ABOUT HTML
  3. About Internet Information Services (IIS):
  4. ABOUT SQL SERVER
• ENTITY RELATIONSHIP – DIAGRAM
• DATA FLOW DIAGRAM
• Activity Diagram
• Sequence Diagram
Online Examination System

• Class Diagram
• Use Case Diagram
• MODULARIZATION DETAILS
• Database Design
• PROCESS LOGIC
  1. USER INTERFACE DESIGN
• SYSTEM TESTING
• LIMITATION OF PROJECT
• FUTURE SCOPE OF THE PROJECT
• References
INTRODUCTION

Online Examination System forms the lifeline of the Educational Institutes to the functioning of the Examination. It is very essential for an Institute to handle the Examinations and their results. It is very useful for an Institute to test its students continuously for their mutual development. This system is helpful for conducting (M.C) Multiple Choice Examinations which can be conducted regularly as well as for surprise tests and provides immediate results saving the precious time of faculties to check the papers and prepare mark sheets. The IT initiatives have encouraged various Organizations to develop systems to facilitate their day to day operations. The Online Examination System will include various Courses and subjects for conducting examinations. This system helps in conducting examinations quickly and can thus help in saving time and the operations will be carried out efficiently. With the effective use, any Institute can apply the Online Examination System for conducting quick examinations and getting better results in less time.

OBJECTIVES OF THE PROJECT

Online Examination System is designed for Educational Institutes like Schools, Colleges, and Private Institutes to conduct logic tests of their students on a regular basis. The system handles all the operations and generates reports as soon as the test is completed which saves the precious time of faculties spent on reviewing answer sheets. The existing system is weak when it comes to surprise test organizations whereas this system can make it possible very easily. The objective of on-line examination system is to take online test in an efficient manner and no time wasting for checking the paper. The main objective of online test simulator is to efficiently evaluate the candidate thoroughly through a fully automated system that not only saves lot of time but also gives fast results. For students they give papers according to their convenience and time and there is no need of using extra thing like paper, pen etc.

Some of the main objectives of this project are as follows

- This can be used in educational institutions as well as in corporate world.
- Can be used anywhere any time as it is a web based application (user Location doesn’t matter).
- No restriction that examiner has to be present when the candidate takes the test.
TOOLS, PLATFORM, HARDWARE AND SOFTWARE REQUIREMENTS SPECIFICATION

Hardware Specification

- INTEL CORE 2 DUO 2.0 GHz
- 2 GB RAM
- 160 GB HDD

Software Specification

- OPERATING SYSTEM: WINDOWS XP/WINDOWS 7 BASIC
- FRONT END: PHP
- DATABASE: SQL Server 2005

SYSTEM ANALYSIS

System Analysis by definition is a process of systematic investigation for the purpose of gathering data, interpreting the facts, diagnosing the problem and using this information to either build a completely new system or to recommend the improvements to the existing system.

A satisfactory system analysis involves the process of examining a business situation with the intent of improving it through better methods and procedures. In its core sense, the analysis phase defines the requirements of the system and the problems which user is trying to solve irrespective of how the requirements would be accomplished.

There are 2 methods to perform System Requirement Analysis:

Structured Analysis

Structured Analysis is an analysis method that provides a basis for developing a model of software to be developed. The objective of structured analysis is to identify the customer requirements and establish a basis to create a software model.
The components of a Structured Analysis are:

- Data Dictionary
- Entity Relationship Diagram
- Data Flow Diagram
- Process Specification
- Control Specification

Object Oriented Analysis

It refers to a detailed study of the various objects involved in a system and the relationship of these objects with each other. While performing an object oriented analysis, the focus of the system analyst is on the availability of the objects that are relevant to software development.

Identification of Need

The heart of the system analysis is aimed at having a detailed understanding of all the important facets of the project under consideration. The key questions are:

- How is it being done?
- How frequently does it occur?
- How great is the volume of transactions or decisions?
- Does a problem exist?
- If a problem exists, how serious is it?
- What is the underlying cause?
To answer the above questions, we discuss with different book store to collect the facts about the current manual system and their opinions of why things happen as they do and their views of changing the existing process. After observation of the system we found that there is a need to computerize the working of the book store because of following reasons

**Preliminary Investigation**

The basic purpose behind Preliminary Investigation is to first clarify, understand and evaluate the Project Request. Preliminary Investigation basically refers to the collection of information that guides the management of an organization to evaluate the merits and demerits of the project request and make an informed judgment about the feasibility of the proposed system. This sort of investigation provides us with a through picture of the kind of software and hardware requirements which are most feasible for the system, plus the environment in which the entire project has to be installed and made operational.

1) **Reviewing the Documents provided by the Organization**

They were quite effective in guiding us towards visualizing the features that were needed to be put together in the system and the required output which had to be generated once the system became functional. These specifications provided to us by the organization showed how the new system should look like; it helped us in understanding the basic structure of the application which we were supposed to develop.

2) **On site Observation:**

Another technique utilized by us to gain information about the project was to visit the client site where the system had to be installed. Here a detailed system study was carried out, checking the existing system to replicate it with our system. We also observed the activities of the system directly. During the on-site observation, we saw the office environment, work load of the system and users, method of work, and the facilities provided by the organization. This information helped us to understand how the system should operate. But after interviewing the persons, who is affected by the system, we got more details that further explain the project and shown whether assistance is merited economically, operationally and technically.
3) **Conducting Interviews:**

This method of investigation conducted by us involved questioning the concerned personnel to get the user’s (client) view about the system and the features they desired it to have.

Some of the Questions put forward by our team were:

- The amount of data needed to be stored.
- The number of customers using the system and number of which the application needed to be installed.
- The issue of our application with existing system was widely discussed.
- The level of access given to the customer would depend on his department.

**FEASIBILITY STUDY**

Feasibility study is the process of determination of whether or not a project is worth doing. Feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report. The contents and recommendations of this feasibility study helped us as a sound basis for deciding how to precede the project. It helped in taking decisions such as which software to use, hardware combinations, etc.

The following is the process diagram for feasibility analysis. In the diagram, the feasibility analysis starts with the user set of requirements. With this, the existing system is also observed. The next step is to check for the deficiencies in the existing system. By evaluating the above points a fresh idea is conceived to define and quantify the required goals. The user consent is very important for the new plan. Along with, for implementing the new system, the ability of the organization is also checked. Besides that, a set of alternatives and their feasibility is also considered in case of any failure in the proposed system. Thus, feasibility study is an important part in software development.
In the SDLC (Systems Development Life Cycle) of our project we maintained a number of feasibility checkpoints between the two phases of the SDLC.

These checkpoints indicate that the management decision to be made after a phase is complete.

The feasibility checkpoints in our project were as follows:

- Survey phase checkpoint
- Study phase checkpoint
- Selection phase checkpoint
- Acquisition phase checkpoint
- Design phase checkpoint

We conducted three tests for Project feasibility namely, Technical, Economical, and Operational feasibilities.
Technical Feasibility

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology and available personnel. Technical feasibility is concerned with specifying equipment and software that will satisfy the user requirement.

This project is feasible on technical remarks also, as the proposed system is more beneficiary in terms of having a sound proof system with new technical components installed on the system. The proposed system can run on any machines supporting Windows and Internet services and works on the best software and hardware that had been used while designing the system so it would be feasible in all technical terms of feasibility.

Technical Feasibility addresses three major issues:

(a) Is the proposed Technology or Solution Practical?

The technologies used are matured enough so that they can be applied to our problems. The practicality of the solution we have developed is proved with the use of the technologies we have chosen. The technologies such as ASP, IIS, Vb script and the compatible H/Ws are so familiar with the today’s knowledge based industry that anyone can easily be compatible to the proposed environment.

(b) Do we currently possess the necessary technology?

We first make sure that whether the required technologies are available to us or not. If they are available then we must ask if we have the capacity. For instance, “Will our current Printer be able to handle the new reports and forms required of a new system?”

(c) Do we possess the necessary Technical Expertise and is the schedule reasonable?

This consideration of technical feasibility is often forgotten during feasibility analysis. We may have the technology, but that doesn’t mean we have the skills required to properly apply that technology. As far as our project is concerned we have the necessary expertise so that the proposed solution can be made feasible. Some projects are initiated with specific deadlines. In our case first we have given three months time but due to some problems regarding time and the constraints of expertise it has been extended to six months. Now there are some organizational constraints that have not yet given us the opportunity to install the system.
Economical Feasibility

Economical feasibility determines whether there are sufficient benefits in sufficient benefits in creating to make the cost acceptable, or is the cost of the system too high. As this signifies cost-benefit analysis and savings. On the behalf of the cost-benefit analysis, the proposed system is feasible and is economical regarding its pre-assumed cost for making a system. Economical feasibility has great importance as it can outweigh other feasibilities because costs affect organization decisions. The concept of Economic Feasibility deals with the fact that a system that can be developed and will be used on installation must be profitable for the Organization. The cost to conduct a full system investigation, the cost of hardware and software, the benefits in the form of reduced expenditure are all discussed during the economic feasibility. During the economical feasibility test we maintained the balance between the Operational and Economical feasibilities, as the two were the conflicting. For example the solution that provides the best operational impact for the end-users may also be the most expensive and, therefore, the least economically feasible.

We classified the costs of our Social Networking site according to the phase in which they occur. As we know that the system development costs are usually one-time costs that will not recur after the project has been completed. For calculating the Development costs we evaluated certain cost categories.

(i) Personnel costs  
(ii) Computer usage  
(iii) Training  
(iv) Supply and equipments costs  
(v) Cost of any new computer equipments and software.

In order to test whether the Proposed System is cost-effective or not we evaluated it through three techniques viz.

- Payback analysis  
- Return on Investment:  
- Net Present value

Operational Feasibility

Operation feasibility is a measure of how people feel about the system. Operational Feasibility criteria measure the urgency of the problem or the acceptability of a solution. Operational Feasibility is dependent upon determining human resources for the project. It refers to projecting whether the system will operate and be used once it is installed.
If the ultimate users are comfortable with the present system and they see no problem with its continuance, then resistance to its operation will be zero. Behaviorally also the proposed system is feasible. A particular application may be technically and but may fail to produce the forecasted benefits, because the company is not able to get it to work. For the system, it is not necessary that the user must be a computer expert, but any computer operator given a little bit of knowledge and training can easily operate.

Our Project is operationally feasible since there is no need for special training of staff member and whatever little instructing on this system is required can be done so quite easily and quickly as it is essentially. This project is being developed keeping in mind the general people who one have very little knowledge of computer operation, but can easily access their required database and other related information. The redundancies can be decreased to a large extent as the system will be fully automated.

**Project planning**

A software project development is a highly labor-intensive activity. Large software may involve hundreds of people and span of time. A project in it is dimension can easily turn into chaos if proper management is not done. Proper management controls and checkpoints are required for effective project monitoring. Controlling the development, ensuring quality, satisfying the constraints of the selective process model are require careful management of the project.

The major issue the project plan addresses are:

- Cost estimation
- Schedule and milestone
- Personal plan
- Software quality assurance
- Configuration management plan
- Project monitoring plans
- Risk management

**Quality Assurance Plan:**

To ensure that the final product is of high quality, some quality control activities must be performed throughout the development. As we saw earlier, if this is not done, correcting errors in the final stages can be vary expansive, especially if they originated in the early phases. The purpose of the software quality assurance plans (SQAP) is to specify all the work products that need to be produced during the project, activities that need to be methods that may be used for the SQA activities.
Note that SQAP takes a board view quality. It is interested in the quality of not only the final product, but also of the intermediate products, even through in a project we are ultimately interested in the quality of the delivered product. This is due to the fact that in a project it is very unlikely that the intermediate work products are of poor quality, but the final product is of high quality. So, to ensure that the delivered software is of good quality it is essential to make sure that the requirements and design are also of good quality.

**Project Scheduling**

It is important, right at the start of the design process, for the designer or design team, to set clear objective. The primary objectives will always be to design a system that delivers the functions required by the client to support the business objective of their organization. For example the system may be required to speed up the production of accurate invoices, or to provide up to date, detailed management information to improve the managing director’s control over the business; or to help senior managers to make strategic decisions. In the other words, to be a quality product – the system must conform to the customer’s requirements and be delivered in a way, which meets their expectations in terms of service. There are many ways in which these requirements might be met by a physical design solution, but there are a number of other objectives that must be considered if a god design is to be produced. The design objective is:

**1. Flexible:**

The design should enable future requirements of the business to be incorporated without too much difficulty. Often during the analysis phase, users may not be clear about exactly what they will require from the new system, for example which reports will be most useful to them. However during the evaluation period after the new system becomes operational, the real needs often emerge and flexible design will be able to accommodate these new requirements. In addition, business change over time and a good design enable the system to reflect these changes.

**2. Maintainable:**

This is closely linked to the previous objective because it is about change. A good design is easy to maintain ands this reduces the client’s maintenance costs, which usually represent a high proportion of the total lifetime cost of the system.
3. **Portable:**

Still on the subject of change, a client who has bought a software system may wish to change the hardware on which the system runs. A good design is portable—in other words it is capable of being transferred from one machine environment to another with the minimum amount of effort to convert it.

4. **Easy to use:**

With the increasing exposure of people to computer applications in the home as well as in the office, expectations of computer systems in terms of their ease of use are also increasing. A good design will result in a system which is ‘user friendly’—easy to understand, not difficult to learn how to use and straightforward to operate.

5. **Reliable:**

This objective is about designing systems which are secure against human error, deliberate misuse of machine failure, and in which data will be stored without corruption. While this is desirable in any computer system, for certain systems in the areas of defence, process control or banking, it will be a key design objective.

6. **Secure:**

Security is another objective that must be considered by the designer. In order to protect the confidentiality of the data, particularly if it is commercially sensitive, it may be important to build in methods to restrict access to authorized users only, for example by introducing passwords.

7. **Programmer-friendly:**

While the other objective are mainly about delivering benefits to the client, the designer must also consider how easy it will be for the programmers to produce the code from the program specifications. By producing a programmer-friendly design, both the costs of production and the risk of building in errors are reduced.

**Cost-effective:**

This includes a number of the other objectives, and is about designing a system that delivers the required functionally, ease (simplicity) of use, reliability, security, etc. to the client in the most cost-objective way.
A two part-design process:

The two design documents describe the same system, but in different ways because of the different audiences for the documents. The conceptual design answers the following questions.

- Where will the data come from?
- What will happen to the data in the system?
- How will the system look to users?
- What choices will be offered to users?
- What is the timing of events?
- How will the reports and screens look like?

The conceptual design describes the system in language understandable to the customer. It does not contain any technical jargons and is independent of implementation.

By contrast, the technical design describes the hardware configuration, the software needs, the communication interfaces, the input and output of the system, the network architecture, and anything else that translates the requirement into the solution to the customer problem.

Sometimes customers are very sophisticated and they can understand the “what” and “how” together. This can happen when customers are themselves software developers and may not require conceptual design. In such cases, a comprehensive design document may be produced.

Planning Tools

Without planning it is difficult to measure progress. As phases are crystallized, crises should begin to disappear. A project manager must plan the life cycle to the project and delegate authority for its implementation.

Project planning involves plotting project activities against a time frame. One of the first steps in planning is developing a road map structure or a network based on analysis of the tasks that must be performed to complete the project. In the early 1900s, formal planning used a Gantt chart or a milestone chart. By plotting activities on the Y-axis and time on the X-axis, the analyst laid out an overall network specifying interrelationships among actions. Later on, formal planning techniques such as the program evaluation and review technique (PERT) was introduced. Other operations research techniques such as linear programming and queuing theory have also been introduced in allocating resources. In the early 1980s, software packages became available for project planning.
Gantt chart:

Basic planning uses bar charts that show project activities and the amount of time they will take. This activity scheduling method was first introduced in 1914 by Henry L. Gantt as a rudimentary aid to plot individual tasks against time. The Gantt chart uses horizontal bars to show the durations of actions or tasks. The left end marks the beginning of the tasks; the right ends its finish. Earlier tasks appear in the upper left and later ones in the lower right.

In planning this project, several steps are undertaken:

1. Identify the activities and tasks in the stage. Each activity must be identified to plan the completion date and allocate responsibilities among members of the project team. In our project, there are seven activities:
   i. Understanding Project Requirement
   ii. Designing Tables
   iii. Designing Forms
   iv. Coding
   v. Report Designing
   vi. Testing
   vii. Implementation

2. Determine the tasks for each activity and the estimated completion times. Each activity is broken down into several tasks.

3. Determine the total estimated time for each activity and obtain an agreement to proceed.

4. Plot activities on a Gantt chart. All activities, tasks, and milestones are drawn on the Gantt chart, with emphasis on simplicity and accuracy.

5. Review and record progress periodically. The actual amount of time spent on each activity is recorded and compared with the budgeted times.
Program Evaluation and Review Technique (PERT)

Like the Gantt chart, PERT makes use of tasks. Like milestone charts, it shows achievements. These achievements however are not task achievements. They are terminal achievement, called events. Arrows are used to represent tasks and circles represent the beginning or completion of a task. The PERT chart uses these paths and events to show the interrelationships of project activities.

The events in my project can be categorized as:

1. Meeting to the Employees of company to understand the project.
2. Table Designing
3. Form Designing
4. Writing Codes
5. Designing Reports
6. Testing the project
7. Implementation of project

Each task is limited by an identifiable event. An event has no duration; it simply tells you that the activity has ended or begun. Each task must have a beginning and an ending event. A task can start only after the tasks depends on have been completed. PERT does not allow “looping back” because a routing that goes back to a task does not end.

A PERT chart is valuable when a project is being planned. When the network is finished, the next step is to determine the critical path. It is the longest path through the network. No task on the critical path can be held up without delaying the start of the next tasks and, ultimately, the completion of the project. So the critical path determines the project completion date.

In addition to showing the interrelationships among project activities, PERT chart shows the following:
1. The activities that must be completed before initiating a specific activity.
2. Interdependencies of the tasks.
3. Other activities that can be completed while a given activity is in progress.
4. The activities that cannot be initiated until after other specific activities are completed. This is called a precedence relationship.

SOFTWARE REQUIREMENT AND SPECIFICATION

The quality of a software product is only as good as the process that creates it. Requirements engineering one of the most crucial steps in this creation process. Without well-written requirements specification, developers do not know what to build, customers do not know what to expect, and there is no way to validate that the built system satisfies the requirements. Requirements engineering includes all activities related to the following:
• Identification and documentation of customer and user’s needs
• Creation of a document that describes the external behavior and the associated constraints that will satisfy those needs
• Analysis and validation of the requirements documents to ensure consistency, and feasibility
• Evolution of needs

The primary output of requirements engineering is requirements specification. If it describes both hardware and software, it is a system requirement specification. If it describes only software, it is a software requirement specification. Requirement stage end with creating a document called the Software Requirement Specification (SRS), which contains a complete description of the external behavior of the software system.

**Nature of the SRS:**

The basic issues that SRS writers shall address are the following:

1. **Functionality:** What the software is supposed to do?
2. **External interfaces:** How does the software interact with people, the system’s hardware, other hardware, and other software?
3. **Performance:** What is the speed, availability, response time, recovery time, etc. of various software functions?
4. **Attributes:** What is the consideration for portability, correctness, maintainability, security, reliability etc.?
5. **Design constraints imposed on an implementation:** Are there any required standards on effect, implementation language, policies for database integrity etc.?

Since the SRS has specific role to play in the software development process, SRS writers should be careful not to go beyond the bounds of that role. This means that SRS

1. Should correctly define all the software requirements. A software requirement may exist because of the nature or the task to be solved or because of a special characteristic of the project.
2. Should not describe any design or implementation details. These should be described in the design stage of the project.
3. Should not impose additional constraints on the software. These are properly specified in other documents such as a software quality assurance plan.

Therefore, a properly written SRS limits the range of valid designs, but does not specify any particular design.

**CHARACTERISTICS OF A GOOD SRS:**

1. Correct: An SRS is correct if; every requirement stated therein one that the software shall meet. There is no tool or procedure that assures correctness.
2. Unambiguous: An SRS is unambiguous if, and only if; every requirement stated therein has only one interpretation.

SOFTWARE ENGINEERING PARADIGM APPLIED

The development strategy that encompasses the process, methods, and tools and the generic phases is called Software Engineering Paradigm. The s/w paradigm for software is chosen based on the nature of the project and application, the method and tools to be used, and the controls and deliverables that are required. All software development can be characterized as a problem-solving loops (fig. 2) in which four distinct stages are encountered: status quo, problem definition, technical development, and solution integration.

Status quo represents the current state of affairs, Problem definition identifies the specific problem to be solved, technical development solves the problem through the application of some technology, and solution integration delivers the results to those who requested the solution in the first place. There are various software paradigms but we used Waterfall model (the linear sequential model), which states that the phases are organized in a linear order. The Waterfall model suggests a systematic, sequential approach to s/w development that begins at the system level and progresses through analysis, design, coding, testing, and maintenance and support as shown in below fig.3.
The sequence of activities performed in a software development project with the Waterfall model is: system analysis, system design, coding, testing & integration, installation, and maintenance. For a successful project resulting in a successful product, all phases listed in the waterfall model must be performed. Any different ordering of the phases will result in a less successful software product. There are a number of project outputs in waterfall model that is produced to produce a successful product:

- Requirement documents and project plan
- System and detailed design
- Programs (code)
- Test plan, test reports and manuals
- Installation reports

LIMITATIONS OF WATERFALL MODEL:

1. The waterfall model assumes that the requirements of a system can be baseline before the design begins. This is possible for system designed to automate an existing manual system. For our system, (Twin Job Portal) this is a new system, determining the requirement is difficult, as the user does not even know the requirements.
2. Freezing the requirements usually requires choosing the hardware
3. The waterfall model stipulates that the requirements be completely specified before the rest of the development can proceed.
4. It is a document driven process that requires formal documents at the end of each phase. This approach tends to make the process documentation-heavy and is not suitable for many applications (interactive applications).

The waterfall model is the most widely used process model. It is suited for routine types of projects where the requirements are well known, i.e., if the developing organization is quite familiar with the problem domain and the requirements for the software are quite clear, the waterfall model works well. This applies on our project.

Since my co-developer and I am the only person who is working on this project, hence, only closed paradigm of software engineering is applicable in this scenario. A Closed paradigm structures a team, along a traditional hierarchy of authorities. Such teams can work well when producing software that is quite similar to past efforts, but they will be less likely to be innovative when working within the closed paradigm. Only two persons, I and my co-developer are concerned with this project, so, I and he are playing the role of system analyst, programmer as well as testing engineer interchangeably.

**SURVEY OF TECHNOLOGY**

**ABOUT ACTIVE SERVER PAGE (PHP):**

**PHP:**

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.

PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

PHP code can be simply mixed with HTML code, or it can be used in combination with various tinplating engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends resulting output to its client, usually in form of a part of the generated web page; for example, PHP code can
generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a command-line interface (CLI) capability and can be used in standalone graphical applications.

ABOUT HTML:

Hypertext Markup Language (HTML) is a language for describing how pages of text, graphics, and other information are organized. Hypertext means text stored in electronic form with cross-reference links between pages. An HTML page contains HTML tags which are embedded commands that supply information about page structure, appearance and contents. Web browser uses the information about how to show the page. HTML pages are standard interface to the Internet. A web browser just retrieves a file and put it on the screen. It actually assembles the component parts of a page and arranges those parts according to commands hidden in the text by the author of the file. Those commands are written in the HTML.

About Internet Information Services (IIS):

Internet Information Services (IIS) is the Windows component that makes it easy to publish information and bring business applications to the Web. IIS makes it easy for you to create a strong platform for network applications and communications Internet Information Services 5.1 has many features to help Web administrators to create scalable, flexible Web applications.

- Security
- Administration
- Programmability
- Internet Standards

Microsoft Internet Information Services 5.0 and 5.1 comply with the HTTP 1.1 standard, including features such as PUT and DELETE, the ability to customize HTTP error messages, and support for custom HTTP headers.

IIS 5.1 offers greater protection and increased reliability for your Web applications. By default, IIS runs all of your applications in a common or pooled process that is separate from core IIS processes.

In IIS 5.1, administrators and application developers have the ability to add custom objects, properties, and methods to the existing ADSI provider, giving administrators even more flexibility in configuring their sites.

Internet Information Services (IIS) makes it easy for you to publish information on the Internet or your intranet. IIS includes a broad range of administrative features for managing Web sites and your Web Server. With programmatic features like Active Server Pages (ASP), you can create and deploy scalable, flexible Web applications.

ABOUT SQL SERVER 2005

SQL Server is one of the most popular RDBMS of today.
Microsoft makes SQL Server available in multiple editions, with different feature sets and targeting different users. These editions are:

SQL Server Compact Edition (SQL CE)

The compact edition is an embedded database engine. Unlike the other editions of SQL Server, the SQL CE engine is based on SQL Mobile (initially designed for use with hand-held devices) and does not share the same binaries. Due to its small size (1 MB DLL footprint), it has a markedly reduced feature set compared to the other editions. For example, it supports a subset of the standard data types, does not support stored procedures or Views or multiple-statement batches (among other limitations). It is limited to 4 GB maximum database size and cannot be run as a Windows service. Compact Edition must be hosted by the application using it. The 3.5 version includes Considerable work that supports ADO.NET Synchronization Services.

FEATURES OF SQL 2005

- **Security Management**: SQL Server provides a controlled access to data to users by providing a combination of privileges.
- **Backup and Recovery**: SQL Server provided sophisticated security backup and recovery routines.
- **Open connectivity**: SQL Server provides open connectivity to and from other vendor’s software such as Microsoft. Also SQL Server database can be access by various front-end software’s such as Microsoft Visual Basic®, Power Builder etc.
- **Space Management**: In SQL Server once can flexibly allocate disk spaces for data storage and can control them subsequently. SQL Server 5 is designed with special feature of data warehousing

ENTITY RELATIONSHIP - DIAGRAM

An E R diagram is a model that identifies the concept or entities that exist in a system and the relationships between those entities. An ERD is often used as a way to visualize a relational database: each entity represents a database table and the relationship lines represents the key in one table that point to specific records in related tables.

Advantages of ER diagram

- Professional and faster Development.
- Productivity Improvement.
-Fewer Faults in Development.
- Maintenance becomes easy.
Online Examination System

ER diagram
A Data flow diagram (DFD) is used to express system requirements in a graphical form. It is also known as a bubble chart. A DFD depicts information flow and transform that are applied as data moves from input. Use of DFD helps to clarify the system requirements. DFD are drawn surging analysis and are the starting point for the design phase. A DFD could be used represent a physical system, at the beginning of analysis, or a logical system at alter point in the system development life cycle. Being graphical, it is easy to understand.

A DFD is used to describe what data flows rather how if flows. The concern is understanding the transforms that are required that are required to convert the input to output. It is independent of hardware, software, data structure and file organization. Symbols used in DFD are as follows:

- A process
- : Defines a source(originator) or destination of system data.
- : The Arrow head shows the flow of information.
- : Data Store
Level 0 DFD

ONLINE EXAMINATION SYSTEM

Administrator:
- Add Subject / Class / Questions
- Create Exam
- Security Management

Student:
- Register For Exam
- Select & Appear Exam
- Request for Result

User Report
Student Report
Inquiry Report

Certificate
Get Rank
Online Examination System

Level 2 DFD

Create Exam Master
- Create New Exam
- Open / Close Exam
- Subject Record
- Exam Report

Subject Master
- Add Subject
- Edit Subject
- Question Record
- Class Details
- Add Class
- Edit Class

Question Master
- Add New Question
- Edit Question

View / Edit Question
- Subject
- Question Report
Online Examination System

Sequence Diagram

User: User

Start interface

Enter ID

Enter Password

Validate at client

Validate at client

Send details

Valid Login

Invalid Login

Display Page

Display Error Page

On Success
Exam Instruction

Login Failure
Class Diagram

Online Examination System
MODULAR DESIGN CONCEPTS

Functional Independence: The concept of functional independence is a direct outgrowth of modularity and the concepts of abstraction and information hiding. The principle of information hiding suggests that modules be “characterized by design decisions that (each) hides from all others”. In other words modules should be specified and designed so that information (procedure and data) contained within a module is inaccessible to other modules that have no need for such information. Hiding implies that effective modularity can be achieved by defining a set of independent modules that communicate with one another only that information necessary to achieve software function. Abstraction helps to define the procedure entities that make up the software. As data and procedure are hidden from other parts of the software inadvertent errors introduced during modification are less likely to propagate to other locations within the software. Functional independence is achieved by developing modules with “single-minded” function and an “aversion” to excessive interaction with other modules.

Advantages: Independent modules are easier to maintain (and test) because secondary effects caused by design or code modification are limited, error propagation is reduced, and reusable modules are possible. Thus with taking utmost care of this concept we have maintained functional independence in our project Twin Job Portal JOBS at some extent that required interaction among different modules is maintained.

(Cohesion: Cohesion of a module represents how tightly bound the internal elements of the module are to one another. Cohesion of a module gives the designer an idea about whether the different elements of a module belong together in the same module.

Coupling: Coupling is a measure of interconnection among modules in a software structure. Coupling depends on the interface complexity between modules, the point at which entry or reference is made to a module, and what data pass across the interface. In software design, we strive for lowest possible coupling. Simple connectivity among modules results in software that is easier to understand and less prone to a “ripple effect” when errors occur at one location and propagate through a system.

Data coupling: Data coupling means simple argument list (data) is passed and a one to one correspondence exists. A variation of data coupling is found when a portion of a data structure rather than simple arguments is passed via a module interface.

Control coupling: When a “control flag” (a variable that controls decisions in a subordinate or super ordinate module) is passed between modules.
External coupling: It is a relatively high level of coupling occurs when modules are tied to an environment external to software. 

Common coupling: When a number of modules reference a global data area. In Twin Job Portal JOBS we have maintained the use of global data but restricted ourselves against the common consequences of this coupling.

Content coupling: The highest degree of coupling, content coupling occurs when one module makes use of data or control information maintained within the boundary of another module. Secondly, content coupling occurs when branches are made into the middle of a module. As this type of coupling makes software complex so in Twin Job Portal JOBS we have tried our best to avoid such coupling.

As the cohesion and coupling are clearly related. Usually the greater the cohesion of each module in a system, the lower the coupling between modules is. So we have maintained a balance between these two engineering concepts.

SPECIFICATION OF MODULES:

Module specification is the major part of system design specification. All modules in the system should be identified when the system design is complete, and these modules should be specified in the document. To specify a module, the design document must specify,

(i) The abstract behavior of the module: specifying the module’s functionality or its input/output.

(ii) The interface of the module: All data items, their types, and whether they are for input and/or output.

(iii) All other modules used by the module being specified: This information is quiet useful in maintaining and understanding the design.

Database Design

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Field name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>User name</td>
<td>Nvarchar</td>
<td>Store user name for checking correct Username</td>
</tr>
<tr>
<td>2</td>
<td>Password</td>
<td>Nvarchar</td>
<td>Store password corresponding to username</td>
</tr>
<tr>
<td>3.</td>
<td>User Type</td>
<td>Nvarchar</td>
<td>User Type Administrator or User</td>
</tr>
</tbody>
</table>

Admin
## Online Examination System

### Online Members

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data type</th>
<th>Field size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login name</td>
<td>Nvarchar</td>
<td>10</td>
</tr>
<tr>
<td>Password</td>
<td>Nvarchar</td>
<td>20</td>
</tr>
<tr>
<td>Confirm Password</td>
<td>Nvarchar</td>
<td>20</td>
</tr>
<tr>
<td>Full name</td>
<td>Nvarchar</td>
<td>20</td>
</tr>
<tr>
<td>Qualification</td>
<td>Nvarchar</td>
<td>10</td>
</tr>
<tr>
<td>Address</td>
<td>Nvarchar</td>
<td>30</td>
</tr>
<tr>
<td>Pin Code</td>
<td>Number</td>
<td>8</td>
</tr>
<tr>
<td>E-mail</td>
<td>Nvarchar</td>
<td>20</td>
</tr>
</tbody>
</table>

### ClassMaster

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data type</th>
<th>Field size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cid</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Nvarchar</td>
<td>20</td>
</tr>
</tbody>
</table>

### Enquiry

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data type</th>
<th>Field size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eid</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Nvarchar</td>
<td>20</td>
</tr>
<tr>
<td>Address</td>
<td>Nvarchar</td>
<td>30</td>
</tr>
<tr>
<td>Pin</td>
<td>Nvarchar</td>
<td>8</td>
</tr>
<tr>
<td>Mobile</td>
<td>Nvarchar</td>
<td>10</td>
</tr>
<tr>
<td>E-mail</td>
<td>Nvarchar</td>
<td>20</td>
</tr>
<tr>
<td>Remarks</td>
<td>Nvarchar</td>
<td>30</td>
</tr>
<tr>
<td>EnquiryDate</td>
<td>Smalldatetime</td>
<td></td>
</tr>
</tbody>
</table>
### ExamMaster

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data type</th>
<th>Field size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExamId</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>ExamName</td>
<td>Nvarchar</td>
<td>20</td>
</tr>
<tr>
<td>Subject</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>TotalTime</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>NoOf Question</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>MaxMarks</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>Passing Marks</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>Nvarchar</td>
<td>1</td>
</tr>
</tbody>
</table>

### Online Exams

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data type</th>
<th>Field size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExamId</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>ExamMasterId</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>Mid</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>Sid</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>NoQ</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>NoCans</td>
<td>Int</td>
<td></td>
</tr>
<tr>
<td>StDate</td>
<td>Smalldatetime</td>
<td></td>
</tr>
<tr>
<td>EndDate</td>
<td>Smalldatetime</td>
<td></td>
</tr>
</tbody>
</table>
USER INTERFACE DESIGN

Admin Login page

User Login Page
Online Examination System

Member view page

Question Master
Exam master page

Start Exam page

Note:
1. This exam contains 5 questions.
2. Use Next and previous buttons to navigate between questions.
3. Results are displayed after the last question is answered.
4. CANCEL button can be used to cancel the exam.
5. No time limitation. However, the time taken is stored in database.
Online Examination System

Question Paper Page

Subject: c

1. What is c?
2. Subject
3. Function
4. Menu

Correct Answers:
1. 
2. 
3. 
4. 

Subject: c

2. What is function?
1. Sub program
2. Sub-sub-sub-sub
3. What is
4. What

Correct Answers:
1. 
2. 
3. 
4. 

Result page

Examination Result

<table>
<thead>
<tr>
<th>Subject</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting Time</td>
<td>00:00:00</td>
</tr>
<tr>
<td>No. of questions</td>
<td>5</td>
</tr>
<tr>
<td>No. of correct answers</td>
<td>3</td>
</tr>
<tr>
<td>Grade</td>
<td>B</td>
</tr>
</tbody>
</table>
Online Examination System

PHP CODING :-

Login Page

```php
<?php
session_start();
$hostname = "localhost";
$username = "root";
$password = "";
$dbhandle = mysql_connect($hostname, $username, $password)
or die("unable to connect to MYSQL");
$select = mysql_select_db("onles", $dbhandle)
or die("Could not select example");

if(isset($_GET['submit']))
{
    $login_name = $_GET['login_name'];
    $password = $_GET['password'];
    $user = $_GET['user'];

    if($user == "admin")
    {
        $table_name = "admin";
        $goto_page = "index.php";
    }
    else
    {
        $table_name = "user";
        $goto_page = "user.php";
    }
    $query = "SELECT * FROM `".$table_name."` WHERE `login_name` = ".$login_name." and `password` = ".$password."";
    $result = mysql_query($query);
    $row = mysql_fetch_array($result);
    $login_name = $row['login_name'];
    $password = $row['password'];
    $user = $row['user'];

    if ($row['login_name'] != NULL && $row ['login_name'] != '')
    {
        $_SESSION['login_name'] = $row['login_name'];
        header("Location: $goto_page");
    }
    else
    {
        header("Location: login.php");
    }
}
?>
```
Error Handling

An Exception occurs when a program encounter any unexpected problems. Such as running out of memory or attempting to read from a file that no longer exists. These problems are not necessarily caused by a programming error but they mainly occur because of violation of assumption that you might have made about the execution environment. When a program encounters an exception the default behavior is to throw the exception which generally translates to abruptly, terminating the program after displaying an error message. But this is not a characteristic of a robust application. But the best way is to bindle the exception situations if possible, gracefully recover from them. This is called “exception handling”. I used try, catch, finally and throw in my project to handle the exception.

The Try Block:
Place the code that might cause exception in a try block. A typical try block looks like this
Try
{
//Code that may cause exception
}
A try block can have another try block inside when an exception occurs at any point rather than executing any further lines of code, the CLR (Common Language Runtime) Secures for the nearest try block that enclosure this code. . The control is then passed to a matching catch block if any and then to the kindly block associated with this try block.

Catch Block:
There can be no of catch blocks immediately following a try block. Each catch block handles an exception of a particular type. When an exception occurs in a statement placed inside the try block the CLR looks for a mainly block that is capable of handling the type of exception.

Throw block:
A throw statement explicitly generates an exception in code. You can throw when a particular path in code results in an anomalous situation.

Finally Block:
The finally block contains the code that always executes whenever or not any exception occurs.

Parameter Passing
Passing parameters from one page to another is a very common task in Web development. There are still many situations in which you need to pass data from one Web page to another. One of the simplest and most efficient ways of passing parameters among pages is to use the query string. Unfortunately, packing data into the query string via string manipulations can quickly lead to cumbersome and often difficult to maintain code, especially as
the parameter list grows. To overcome this problem, I’ve used Session in my project.

**Query String some of them are described below**

- Query String is client side. But Session is server side.
- The information or data stored in Query String is visible to everyone. But in Session it is hidden and can’t be viewed easily.
- Query String can store only a piece of information but in Session we can store the more and more data.
- The Query String speed never falls as the load increase because it stores a piece of information. But on the other hand Session increase congestion as the loads increase.

**SYSTEM TESTING**

System testing is the expensive and time-consuming process. There are two Strategies for testing software that we use for testing our system: Code testing and Specification Testing. In Code testing, we developed those cases to execute every instructions and path in the program. In specification testing, we examined the program specification and then wrote test data to determine how the program operates under specified condition. The different levels of testing are used in the testing process. The basic levels are unit testing, integration testing, system testing, and acceptance testing. These different levels of testing detect different types of faults. The different levels of testing are shown in figure in next page.

**TESTING:**

**Test Case Execution:** - The workflow diagram below depicts the high level steps necessary to follow in order to set up and execute test based on the Test Case Template.

**Legend**

- *BII WG4 Test Case Template.doc.* The **Test Case template** used to define and set up the **Test Case Description**.
- The **test object specification** provides a reference to the object subject to test or if required, enters a copy of the object description excerpted from the object description for the test object. When referenced, the reference should include at least:

**SYSTEM TESTING:**

System testing is the expensive and time-consuming process. There are two strategies for testing software that we use for testing our system: Code Testing and Specification Testing. In Code testing, we developed those cases to execute every instructions and path in the program. In specification testing,
we examined the program specification and then wrote test data to determine how the program operates under specified condition.

OVERVIEW OF TESTING: -

**Testing:** - Testing involves executing the program (or part of it) using sample data and inferring from the output whether the software performs correctly or not. This can be done either during module development (unit testing) or when several modules are combined (system testing).

**Defect Testing:** - Defect testing is testing for situation where the program does not meet its fictional specification. Performance testing tests a system's performance or reliability under realistic loads. This may go some way to ensuring that the program meets its non-functional requirements.

**Debugging:** - Debugging is a cycle of detection, location, repair and test. Debugging is a hypothesis testing process. When a bug is detected, the tester must form a hypothesis about the cause and location of the bug. Further examination of the execution of the program (possible including many returns of it) will usually take place to confirm the hypothesis. If the hypothesis is demonstrated to be incorrect, a new hypothesis must be formed. Debugging tools that show the state of the program are useful for this, but inserting print statements is often the only approach. Experienced debuggers use their knowledge of common and/or obscure bugs to facilitate the hypothesis testing process. After fixing a bug, the system must be reset to ensure that the fix has worked and that no other bugs have been introduced. This is called regression testing. In principle, all tests should be performed again but this is often too expensive to do.

TEST PLANNING: -

Testing needs to be planned to be cost and time effective. Planning is setting out standards for tests. Test plans set out the context in which individual engineers can place their own work. Typical test plan contains:

Overview of testing process

- Requirements trace ability (to ensure that all requirements are tested)
- List of item to be tested
- Schedule
- Recording procedures so that test results can be audited
- Hardware and software requirement

OVERVIEW OF TESTING STRATEGIES: -
Large system usually tested using a mixture of strategies. Different strategies may be needed for different parts of the system or at a stage of the process.
Top-down testing: -
This approach tests high levels of system before detailed components. This is an appropriate when developing the system top-down likely to show up structural design errors early (and therefore cheaply) has advantage that a limited, working system available early on. Validation (as distinct from verification) can begin early. Its disadvantage is that stubs needs to be generated (extra effort) and might be impracticable if component is complex (e.g. converting an array into a linked list; unrealistic to generate random list; therefore end up implementing unit anyway). Test output may be difficult to observe (needs creation of artificial environment). This is not appropriate for OO systems (except within a class).

Bottom-up testing: -
This is opposite of top-down testing. This testing test low-level unit then works up hierarchy. Its advantages and disadvantages of bottom-up mirror those of top-down. In this testing there is need to write test drivers for each unit. These are as reusable as the unit itself. Combining top-down development with bottom-up testing means that all parts of system must be implemented before testing can begin, therefore does not accord with incremental approach discussed above. Bottom-up testing less likely to reveal architectural faults early on. However, bottom-up testing of critical low-level components is almost always necessary. Appropriate for OO systems.

Stress testing: -
Test system’s ability to cope with a specified load (e.g. transactions per second). Plan tests to increase load incrementally. Go beyond design limit until system fails (this test particularly important for distributed systems (check degradation as network exchange data).

Back-to-back testing: -
Comparison of test results from different versions of the system (e.g. compare with prototype, previous version or different configuration). Process - Run first system, saving test case results. Run second system, also saving its results. Compare results files. Note that no differences don’t imply no bugs. Both systems may have made the same mistake.

Defect testing: - A successful defect test is a test that causes the system to behave incorrectly. Defect testing is not intended to show that a program meets its specification. If tests don’t show up defects it may mean that the tests are not exhaustive enough.

Exhaustive testing is not always practicable. Subset has to be defined (this should be part of the test plan, not left to the individual programmer). Possible methods: Test capabilities rather than components (e.g. concentrate on tests for data loss over ones for screen layout).

- Test old in preference to new (users less effected by failure of new capabilities).
- Test typical cases rather than boundary ones (ensure normal operation works properly).
Three approaches to defect testing. Each is most appropriate to different types of component. Studies show that black box testing is more effective in discovering faults than white-box testing. However, the rate of fault detection (faults detected per unit time) was similar for each approach. Also showed that static code reviewing was more effective and less expensive than defect testing.

**Black-box (Functional) Testing:**

Testing against specification of system or component. Study it by examining its inputs and related outputs. Key is to devise inputs that have a higher likelihood of causing outputs that reveal the presence of defects. Use experience and knowledge of domain to identify such test cases. Failing this a systematic approach may be necessary. Equivalence partitioning is where the input to a program falls into a number of classes. E.g. positive numbers vs. negative numbers. Programs normally behave the same way for each member of a class. Partitions exist for both input and output. Partitions may be discrete or overlap. Invalid data (i.e. outside the normal partitions) is one or more partitions that should be tested. Test cases are chosen to exercise each portion. Also test boundary cases (atypical, extreme, zero) since these frequently show up defects. For completeness, test all combinations of partitions. Black box testing is rarely exhaustive (because one doesn't test every value in an equivalence partition) and sometimes fails to reveal corruption defects caused by "weird" combination of inputs. Black box testing should not be used to try and reveal corruption defects caused, for example, by assigning a pointer to point to an object of the wrong type. Static inspection (or using a better programming language!) is preferable for this.

**White-box (structural) Testing:**

Testing based on knowledge of structure of component (e.g. by looking at source code). Advantage is that structure of code can be used to find out how many test case need to be performed. Knowledge of the algorithm (examination of the code) can be used to identify the equivalence partitions. Path testing is where the tester aims to exercise every independent execution path through the component. All conditional statements tested for both true and false cases. If a unit has n control statements, there will be up to 2n possible paths through it. This demonstrates that it is much easier to test small program units than large ones. Flow graphs are a pictorial representation of the paths of control through a program (ignoring assignments, procedure calls and I/O statements). Use flow graph to design test cases that execute each path. Static tools may be used to make this easier in programs that have a complex branching structure. Tools support. Dynamic program analyzers instrument a program with additional code. Typically this will count how many times each statement is executed. At end, print out report showing which statements have and have not been executed.
Debugging and Code Improvement

In ideal worlds, all programmers would be so skilled and attentive to detail that they would write bug-free code. Unfortunately, we do not live in an ideal world. As such, debugging, or tracking down the source of errors and erroneous result, is an important task that all developers need to perform before they allow end-user to use their applications. We will discuss some techniques for reducing the number of bugs in code up front.

There are three categories of bugs

Syntax error:
These errors occur when code breaks the rule of the language, such as visual Basic sub statement without a closing End sub, or a forgotten closing curly braces (}) in php. Theses error the easiest to locate. The language complier or integrated development environment (IDE) will alert you to them and will not allow you to compile your program until you correct them.

Semantic error
These errors occur in code that is correct according to rules of the compiler, but that causes unexpected problems such as crashes or hanging on execution. A good example is code that execute in a loop but never exists the loop, either because the loop depends on the variable whose values was expected to be something different than it actually was or because the programmer forget to increment the loop counter.

Logic error
Logic errors are like semantic errors, logic errors are runtime error. That is, they occur while the program is running. But unlike semantic errors, logic errors do not cause the application to crash or hang. Logic error results in unexpected values or output. This can be a result of something as simple as a mistyped variables name that happens to match another declared variable in the program. This type of error can be extremely difficult to track down to eliminate.

Preventing Debug

Write readable code:
Develop and make consistent use of naming and coding standards. It not that Important which standard we use, such as Hungarian notation or ascal Casing (First Name) or another naming conventions, as long as we use one. We should also strive for consistency in our comments and encourage liberal commenting code.

Create effective test plan
The only effective way to eliminate logic error is to test very path of your application with every possible data values that a user could enter. This is difficult to manage without effective planning. We should create our test plan at the same time we are designing the application, and we should update these plans as you modify the application design.
Use a rich IDE

We should consider developing in an IDE that provides syntax checking as we type. If we develop with notepad, it is too easy to a mass number of syntax error that go unnoticed until you try to run the page. Then we got to spend the next half hour or more eliminating the errors, until you finally get the code to run. This not efficient way to write code.

Get another pair of eyes

Whether we working on a team or building an application on your own, it is important to have someone else review our code. Developers are simply to close to their own code to catch every bug before testing.

Code Improvement

We make the function “Data Utility” in our project which is useful for reducing the code redundancy and make code consistency. “Data Utility” function improves the code, when by using the function “Data Utility” we can create a connection with database, open the database, close the database, dispose the database, through the “DataUtility” function can access the datatable. In our project we can create the object of the “DataUtility” then after creating the object we do not need to write the functions, such as database connectivity, open connection close connection, and dispose connection, getdatatable connection. “DataUtility” functions make the code improvement in our project. Through the “DataUtility” function we can insert, delete, update the records and show the datatables and check the database so can say that the “DataUtility” function is more useful and make code improve.

LIMITATION OF PROJECT

- Only registered users can use this website.
- User have to login before using this website.
- Users can not use the others information.
- User Id or Email–ID should be unique or must for sing in.

FUTURE SCOPE OF THE PROJECT

The project has a very vast scope in future. The project can be implemented on internet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of Web Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner. The following are the future scope for the project.

The number of levels that the software is handling can be made unlimited in future from the current status of handling up to N levels as currently laid down by the software. Efficiency can be further enhanced and boosted up to a
great extent by normalizing and de-normalizing the database tables used in the project as well as taking the kind of the alternative set of data structures and advanced calculation algorithms available.

We can in future generalize the application from its current customized status wherein other vendors developing and working on similar applications can utilize this software and make changes to it according to their business needs.

- Faster processing of information as compared to the current system with high accuracy and reliability.
- Automatic and error free report generation as per the specified format with ease.
- Automatic calculation and generation of correct and precise Bills thus reducing much of the workload on the accounting staff and the errors arising due to manual calculations. With a fully automated solution, lesser staff, better space utilization and peaceful work environment, the company is bound to experience high turnover.

References:
Complete Reference of php
Programming in php - Deitel & Deitel
www.w3schools.com
http://en.wikipedia.org
The principles of Software Engineering – Roger S.Pressman
Software Engineering – Hudson
MSDN help provided by Microsoft .NET