iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 Issue: II Month of publication: February 2022

DOI: https://doi.org/10.22214/ijraset.2022.39619

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue II Feb 2022- Available at www.ijraset.com

Neonatal Screening Application

Adarsh Raj[1], Akshit Kaneriya[2], Atharva Gham[3], Gaurav Singh Dafauti[4]
[1, 2, 3, 4]*Department of Computer Engineering/ LY-B Vishwakarma University*

Abstract: Neonatal screening is a set of medical diagnostics that are performed on infants. Our project deals with the problem of a centralized management system for hospitals making it easier for doctors as well as patients to view reports and compare current reports with the previous reports. Our app would also help in reducing errors that are generally caused when this process of comparing and contrasting the report is done manually by doctors.

The main objective of this project is to develop a Client Server based environment, for transferring the medical report details from the bio medical lab. Also to improve data analysis technique, using previous data comparison methods.

This system works on accurate recording of all user report transactions lead to better data management and increases in various comparisons. This project consists of two types of interactions like admin and patient. Admin will be providing with a well secured login and patient details will be registered using admin. A user name and password will be provided to the user. The initially process is to enter the medical test requirement of the patient. The details contain prescribed doctor details, patient details and lab test details. Each disease will contain different CTA (Complete test analysis) for their test result. For example a general blood sugar test contains HBL cholesterol, LDL cholesterol, Glucose level and etc. These all said to be CTA test.

Once patient has been register successfully their blood will be collected and a temporary reference number will be provided. This temporary number is for admin and employee reference. Once their blood test has been done, the test result will be uploaded by the admin by selected the corresponding user name. The uploading details will be stored in a centralized server for data utility purpose. Patient can login with their user and password. Through selecting the date, the patient can view their current test result from their home itself. A test history option will be provided to the patient. So that patient can compare their test with their previous test results. This option makes the patient to manage and know about their health condition and their treatment. An interactive grid and chart will be generated for furnishing the result to the patient. Admin can maintain their patient detail, patient count, test results and etc. This makes more comfortable communication between admin and patients. Also entire treatment record will be computerized for future references.

I. INTRODUCTION

Neonatal Screening Application is important as this screening will test ,analyze and predict the health of infants. Neonatal screening is a system for identifying genetic and other health problems.This test look for serious developmental, genetic and metabolic disorders so that major actions can be taken during the critical time before symptoms develop in the newborns. As the detection is early ,it will reduce or eliminate the effects of the condition .Many of the conditions detectable in Newborn screening ,if left untreated ,have serious symptoms and effects for life longs nervous system damage, developmental ,and physical disabilities and even death. The Software will suggest the results of conditions detected by testing done.

II. MODULES

A. Admin Login and Patient Creation
B. Upload Test Reports
C. User View
D. Upload Contents
E. Charts and Grids

III. MODULES DESCRIPTION

A. Admin Login

The initial module of this project is the admin module. Here admin will be provided with a user name and password. There is also a possibility for a password changing option inside the admin login. The admin will be fully authorized person for this entire project. In addition he will be fully operational authority of this project. The admin is able to access all the options in this project and he can able to do all types of updating. And also admin is able to create patients and provide their username and password.

B. Upload Test Reports

This module is under the control of admin. A temporary reference number will be generated for patients. After the completion of lab test, the test report details will be uploaded in the patient zone. The uploading data will be centralized in server. The uploading details will be categorized into date wise, doctor wise, Patient wise, disease wise and etc. Each and every test contains sub test details. Full payment will be collected from the user side while patient giving their blood to the lab advisor.

C. User View

Using the user name and password, the patient can login anywhere at any time. All the uploaded contents can be viewed in the user's login. A change password option will be provided to the user to make their treatment history more secured**.**

D. Upload Content

A premium option will be provide to the patient in this module. In case the patient has been taking treatment in other place means, after their treatment, patient can upload their new lab report with the existing treatment report. Data merging technique has been implemented for merging the existing data with the current data. Not even other lab details, they can upload various medical records, which can be used in their future.

E. Chart and Grid

This is the data display module, various grid and charts can be generated from the user side. This gives the data more clarity to access. Patient can select from date and to date to view their treatment history. All data will be compared in charts and graphs for clarity results. Using this option patient can monitor their health. And they will get more confidence on their treatment.

IV. SOFTWARE CONFIGURATION

FRONTEND	:	HTML, CSS, JS
CODING LANGUAGE :		C#
BACK END	:	ASP.NET, SQL SERVER 2010
OPERATING SYSTEMS :		Microsoft Windows 10
DOCUMENTATION	:	Microsoft word 2007.
SCRIPTING LANGUAGE :		JavaScript

V. SYSTEM STUDY

A. Existing System

In the current Neonatal screening Bio Labs admin and patients are facing more problems due to manual works. In case the blood sample form the patient is received means, they need to wait for the medical result or they need to come again to receive their medical reports.

This makes the patient more inconvenient to travel and wait for the result. In some times the patient may miss their report, at that time they need to visit again to the lab for getting their report again. Also some times the patient may miss their previous report , which may need to track their medical history.

In case of verifying the patient history with the manual paper reports means, they need to compare their reports manually. Admin also facing more problems in the existing system like updating the test details and managing the customer details. Still doing more paper works with problems

Some of the bio labs are computerized like maintaining their customer details and proving the reports in a computer printout. But they still typing their result in a word file and proving printouts to the patients. They are not maintaining proper patient details with their previous medical reports. This is the most important problems facing by the bio lab now a day.

VI. DISADVANTAGES OF EXISTING SYSTEM:

A. Only Manual work
B. No computerized patient report
C. No customized login for patients to receive their medical report
D. No comparison for the records with the previous records
E. Patients may miss their reports
F. Admin facing more problem in adding new disease to the database
G. Problem in maintain customers

VII. PROPOSED SYSTEM

All the drawbacks in the existing system have been over came in the proposed system. The important of the proposed system in computerization of the manual work to an automated process. In the proposed system the patient need not visit the lab frequently. Once the blood sample has been collected from the patient, the lab will start their process. The lab admin will be directly entering the reports details into the patient corresponding username. This data will be transfer to the patient's login directly. Also all the patients will be provided with a user name and a password. Patient can directly login in the link and they can view their test report immediately.

For the admin part, no paper works or manual print outs will be done. All the details will be computerized in this application. Lab can provide both software copy and hardcopy to the patients. What even data has been provided will be computerized in the system. Even after some year a particular patient's details can be fetched out.

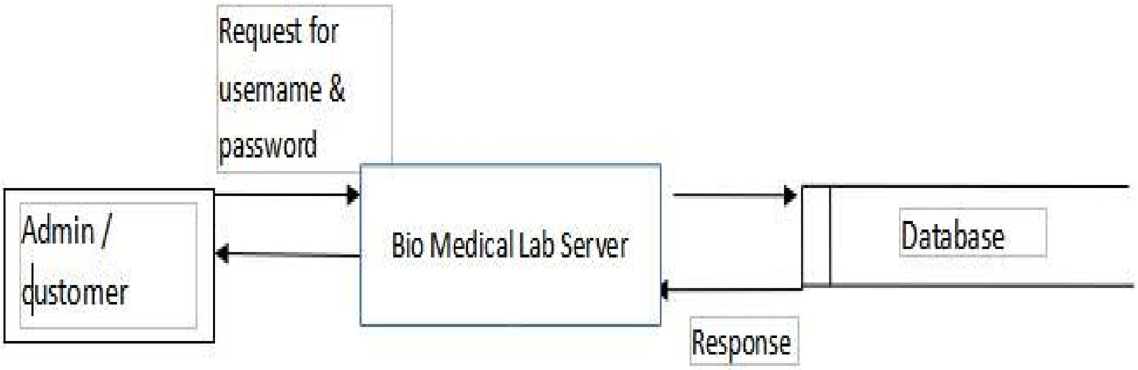
VIII. ADVANTAGES OF THE PROPOSED SYSTEM

A. Fully automated work.
B. All patient details and report details will be computerized.
C. Individual login will be provided to patients with user name and password.
D. Reports can be compared by the patient in their login itself. Patient will get aware about their treatment while comparing their records.
E. No possible way to miss their records, everything will be computerized.
F. In case of adding new disease details, admin can add or remove the disease details with admin login.
G. Any customers can be tracked easily and can easily maintain their details also.

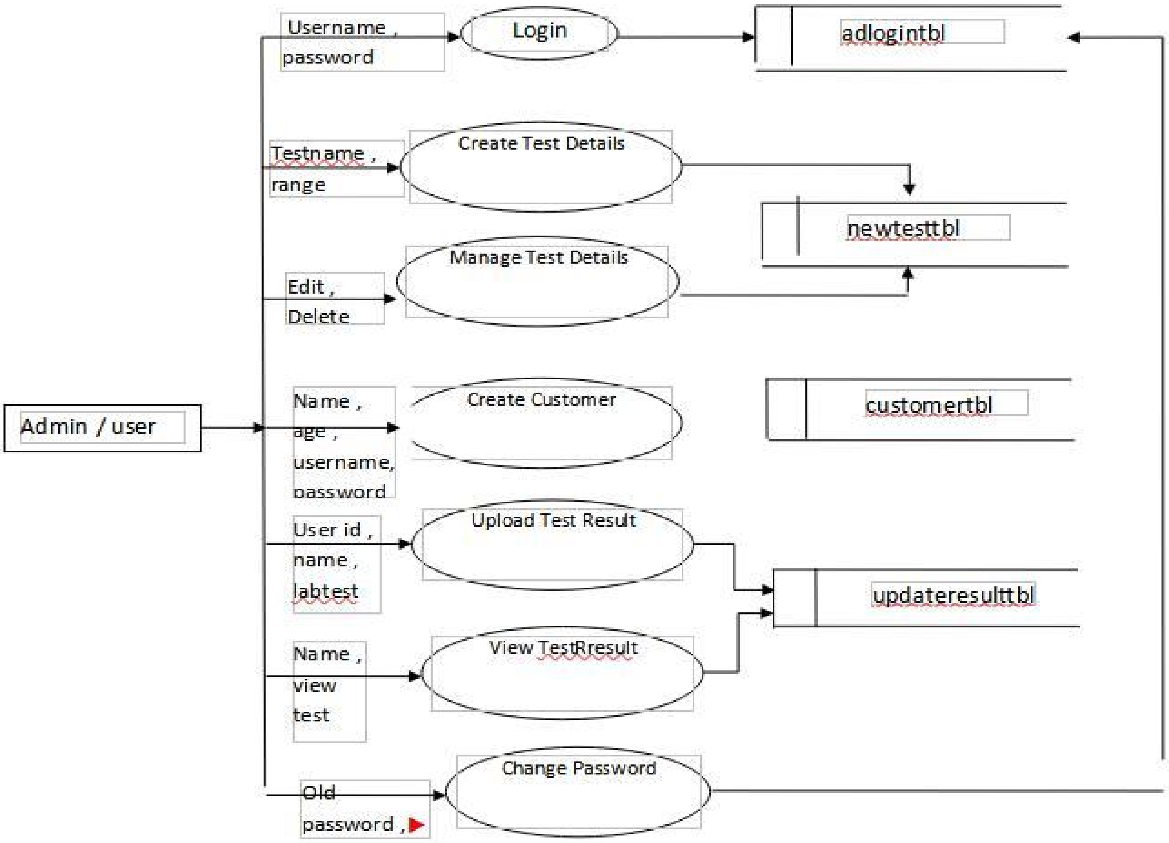
IX. SYSTEM DESIGN

A. Data Flow Diagram

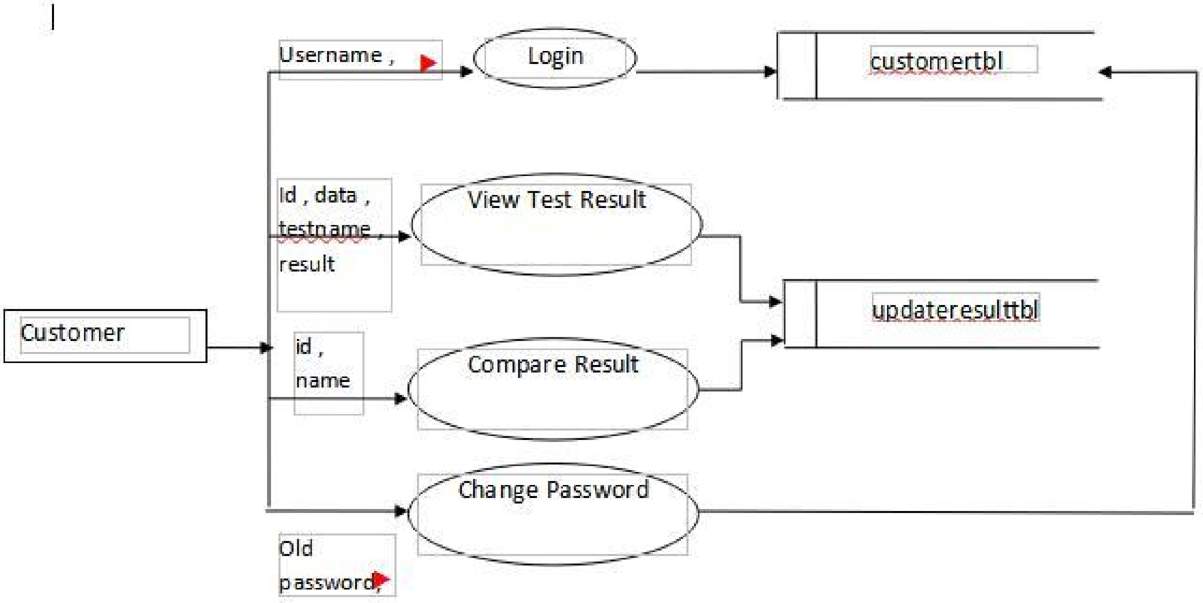
1) Level 0



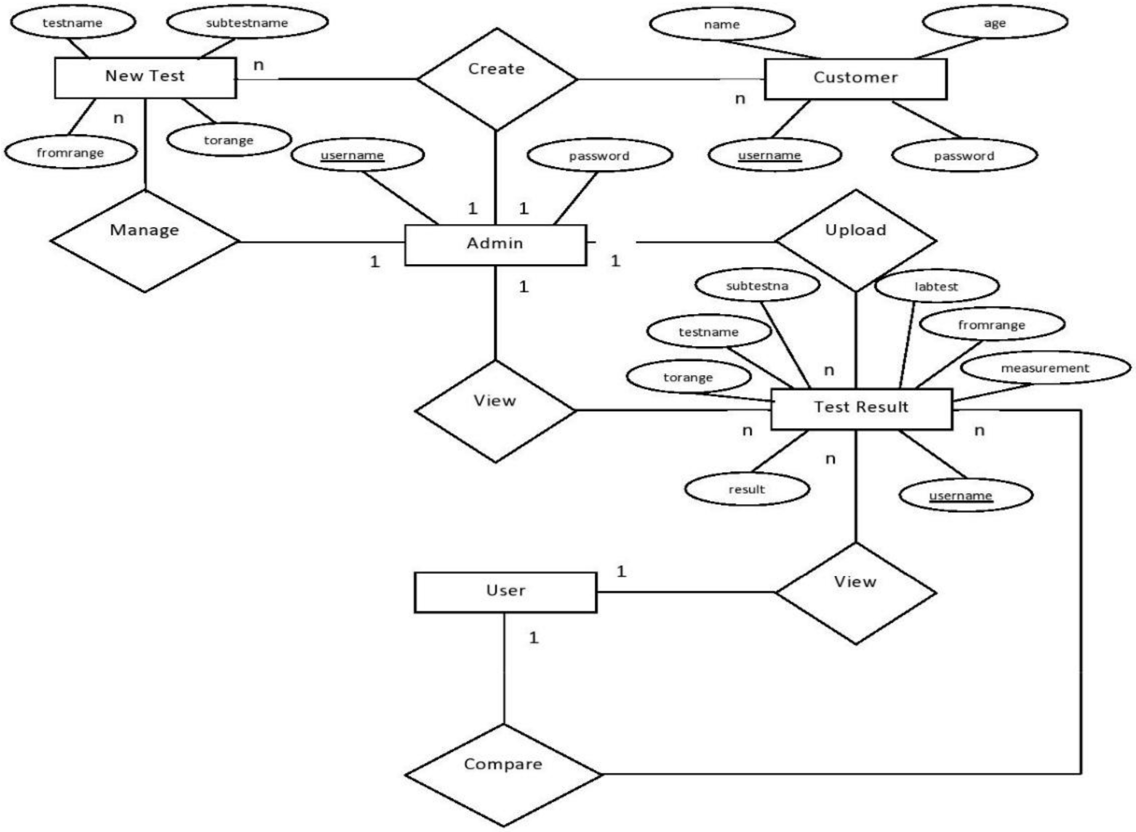
2) Level 1



3) Level 2



X. ER.DIAGRAM



XI. TABLE DESIGN

Table Name: adlogintbl
Primary key : username

Fieldname	Datatype	Size	Constrains	Description
Username	VARCHAR	50	Primary key	Admin Uesrname
Password	VARCHAR	20	NOT NULL	Admin Passowrd

Table Name: customertbl
Primary key : username

FieldName	Datatype	Size	Constrains	Description
Name	VARCHAR	50	PRIMARY KEY	name
gender	VARCHAR	20	NOT NULL	gender
age	INT	10	NOT NULL	age
bloodgroup	VARCHAR	50	NOT NULL	Blood group
street1	VARCHAR	100	NOT NULL	address1
street2	VARCHAR	100	NOT NULL	address2
city	VARCHAR	100	NOT NULL	city
state	VARCHAR	100	NOT NULL	state
pin	INT	10	NOT NULL	pin
phone	INT	10	NOT NULL	Phone number
mail	VARCHAR	100	NOT NULL	Email id
username	VARCHAR	20	NOT NULL	password

Table Name: newtesttbl
Primary key : testname

FieldName	Datatype	Size	Constrains	Description
testname	VARCHAR	50	PRIMARY	Create testname
subtestname	VARCHAR	50	NOT NULL	Create subtestname
frmrange	INT	10	NOT NULL	Create from range
fmeasurement	VARCHAR	50	NOT NULL	Create from measurement
torange	INT	10	NOT NULL	Create to range
tmeasurement	VARCHAR	50	NOT NULL	Create to measurement

Table Name: updateusertbl
Primary key: id
Foreign key: username

FieldName	Datatype	Size	Constraints	Description
ID	INT	10	PRIMARY KEY	Identification number
username	VARCHAR	50	FOREIGN KEY	Customer username
name	VARCHAR	50	NOT NULL	Customer name
testname	VARCHAR	50	NOT NULL	testname
subtestname	VARCHAR	50	NOT NULL	subtestname
labtest	VARCHAR	50	NOT NULL	Labtest name
labtestvalue	INT	10	NOT NULL	Labtest value
frmrange	INT	10	NOT NULL	From range
fmeasurement	VARCHAR	50	NOT NULL	From measurement
torange	INT	10	NOT NULL	To range
tmeasurement	VARCHAR	50	NOT NULL	To measurement
result	VARCHAR	50	NOT NULL	Test result
update	Datetime	-	NOT NULL	Update date
uptime	Datetime	-	NOT NULL	Update time

XII. TECHNICAL STACK DESCRIPTION

Asp.net is a server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. It was first released in January 2002 with version 1.0 of the.net framework, and is the successor to Microsoft's active server pages (asp) technology. Asp.net is built on the common language runtime (clr), allowing programmers to write asp.net code using any supported .net language. The asp.net soap extension framework allows asp.net components to process soap messages. After four years of development, and a series of beta releases in 2000 and 2001, asp.net 1.0 was released on January 5, 2002 as part of version 1.0 of the .net framework. Even prior to the release, dozens of books had been written about asp.net, and Microsoft promoted it heavily as part of its platform for web services. Scott Guthrie became the product unit manager for asp.net, and development continued apace, with version 1.1 being released on April 24, 2003 as a part of windows server 2003. This release focused on improving asp. Net's support for mobile devices.

A. Characteristics

Asp.net web pages, known officially as web forms, are the main building blocks for application development. web forms are contained in files with a ".aspx" extension; these files typically contain static (x)html markup, as well as markup defining server-side web controls and user controls where the developers place all the rc content[further explanation needed] for the web page. Additionally, dynamic code which runs on the server can be placed in a page within a block <% -- dynamic code -- %>, which is similar to other web development technologies such as php, jsp, and asp. With asp.net framework 2.0, Microsoft introduced a new code-behind model which allows static text to remain on the.aspx page, while dynamic code remains in an .aspx.vb or .aspx.cs or .aspx.fs file (depending on the programming language used).

B. Directives

A directive is a special instruction on how asp.net should process the page.the most common directive is <%@ page %> which can specify many attributes used by the asp.net page parser and compiler.

C. Source Language c#

```
Using system;
Namespace website
{
public partial class samplecodebehind : system.web.ui.page
{
protected void page_load(object sender, eventargs e)
{
response.write("hello, world");
}
}
}
```

D. Source language visual basic.net:

```
Imports system
Namespace website
public partial class samplecodebehind
inherits system.web.ui.page
protected sub page_load(byval sender as object, byval e as eventargs)
response.write("hello, world")
end sub
end class
```

E. End Namespace

In this case, the page_load() method is called every time the aspx page is requested. The programmer can implement event handlers at several stages of the page execution process to perform processing.

F. User Controls

User controls are encapsulations of sections of pages which are registered and used as controls in asp.net, etc.

G. Custom Controls

Programmers can also build custom controls for asp.net applications. Unlike user controls, these controls do not have an ascx markup file, having all their code compiled into a dynamic link library (dll) file. Such custom controls can be used across multiple web applications and visual studio projects.

H. Rendering Technique

Asp.net uses a visited composites rendering technique. During compilation, the template (.aspx) file is compiled into initialization code which builds a control tree (the composite) representing the original template. Literal text goes into instances of the literal control class, and server controls are represented by instances of a specific control class. The initialization code is combined with user-written code (usually by the assembly of multiple partial classes) and results in a class specific for the page. The page doubles as the root of the control tree. Actual requests for the page are processed through a number of steps. First, during the initialization steps, an instance of the page class is created and the initialization code is executed. This produces the initial control tree which is now typically manipulated by the methods of the page in the following steps. As each node in the tree is a control represented as an instance of a class, the code may change the tree structure as well as manipulate the properties/methods of the individual nodes. Finally, during the rendering step a visitor is used to visit every node in the tree, asking each node to render itself using the methods of the visitor. The resulting html output is sent to the client. After the request has been processed, the instance of the page class is discarded and with it the entire control tree. This is a source of confusion among novice asp.net programmers who rely on the class instance members that are lost with every page request/response cycle.

I. State Management

Asp.net applications are hosted by a web server and are accessed using the stateless http protocol. As such, if an application uses state full interaction, it has to implement state management on its own. Asp.net provides various functions for state management. Conceptually, Microsoft treats "state" as gui state. Problems may arise if an application needs to keep track of "data state"; for example, a finite-state machines which may be in a transient state between requests (lazy evaluation) or which takes a long time to initialize. State management in asp.net pages with authentication can make web scraping difficult or impossible.

J. Application

Application state is held by a collection of shared user-defined variables. These are set and initialized when the application_onstart event fires on the loading of the first instance of the application and are available until the last instance exits. Application state variables are accessed using the applications collection, which provides a wrapper for the application state. Application state variables are identified by name.

K. Session State

Server-side session state is held by a collection of user-defined session variables that are persistent during a user session. These variables, accessed using the session collection, are unique to each session instance. The variables can be set to be automatically destroyed after a defined time of inactivity even if the session does not end. Client-side user session is maintained by either a cookie or by encoding the session id in the url itself.

Asp.net supports three modes of persistence for server-side session variables:

L. In-process Mode

The session variables are maintained within the asp.net process. This is the fastest way; however, in this mode the variables are destroyed when the asp.net process is recycled or shut down.

M. Asp State Mode

Asp.net runs a separate windows service that maintains the state variables. Because state management happens outside, the asp.net process, and because the asp.net engine accesses data using .net removing, asp state is slower than in-process. This mode allows an asp.net application to be load-balanced and scaled across multiple servers.

Because the state management service runs independently of asp.net, the session variables can persist across asp.net process shutdowns. However, since session state server runs as one instance, it is still one point of failure for session state. The session-state service cannot be load-balanced, and there are restrictions on types that can be stored in a session variable.

N. Database

SQL Server is Microsoft's relational database management system (RDBMS). It is a full-featured database primarily designed to compete against competitors Oracle Database (DB) and MySQL.

Like all major RBDMS, SQL Server supports ANSI SQL, the standard SQL language. However, SQL Server also contains T-SQL, its own SQL implemention. SQL Server Management Studio (SSMS) (previously known as Enterprise Manager) is SQL Server’s main interface tool, and it supports 32-bit and 64-bit environments.

XIII. SYSTEM TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Development Testing is a software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the software development lifecycle. Rather than replace traditional QA focuses, it augments it. Development Testing aims to eliminate construction errors before code is promoted to QA; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development and QA process. Software testing can be stated as the process of validating and verifying that a computer program/application/product:

1) meets the requirements that guided its design and development,
2) works as expected,
3) can be implemented with the same characteristics,
4) and satisfies the needs of stakeholders.
5) White-box testing,
6) Black-box testing,
7) Unit testing,
8) Integration testing,
9) System testing,
10) Acceptance testing,
11) Validation testing

A. Test Case

A test case normally consists of a unique identifier, requirement references from a design specification, preconditions, events, a series of steps (also known as actions) to follow, input, output, expected result, and actual result. Clinically defined a test case is an input and an expected result.

This can be as pragmatic as 'for condition x your derived result is y', whereas other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps (but often steps are contained in a separate test procedure that can be exercised against multiple test cases, as a matter of economy) but with one expected result or expected outcome.

The optional fields are a test case ID, test step, or order of execution number, related requirement(s), depth, test category, author, and check boxes for whether the test is automatable and has been automated. Larger test cases may also contain prerequisite states or steps, and descriptions.

A test case should also contain a place for the actual result. These steps can be stored in a word processor document, spreadsheet, database, or other common repository.

In a database system, you may also be able to see past test results, who generated the results, and what system configuration was used to generate those results. These past results would usually be stored in a separate table.

XIV. IMPLEMENTATION

System implementation generally benefits from high levels of user involvement and management support. User participation in the design and operation of information systems has several positive results. First, if users are heavily involved in systems design, they move opportunities to mold the system according to their priorities and business requirements, and more opportunities to control the outcome. Second, they are more likely to react positively to the change process. Incorporating user knowledge and expertise leads to better solutions.

The relationship between users and information systems specialists has traditionally been a problem area for information systems implementation efforts. Users and information systems specialists tend to have different backgrounds, interests, and priorities. This is referred to as the user-designer communications gap. These differences lead to divergent organizational loyalties, approaches to problem solving, and vocabularies

Conversions to new systems often get off track because companies fail to plan the project realistically or they don’t execute or manage the project by the plan. Remember that major systems conversions are not just IT projects. Companies should maintain joint responsibility with the vendor in the project-planning process, maintenance of the project-plan status, as well as some degree of control over the implementation.

All key user departments should have representation on the project team, including the call center, website, fulfillment, management, merchandising, inventory control, marketing and finance. Team members should share responsibilities for conversion, training and successful completion of the project tasks.

The software vendor should have a time-tested project methodology and provide a high-level general plan. As the merchant client, your job is to develop the detailed plan with the vendor, backed up with detail tasks and estimates.

For example, a generalized plan may have a list of system modifications, but lack the details that need to be itemized. These may include research, specifications, sign-offs, program specs, programming, testing and sign-off, and the various levels of testing and program integration back into the base system.

Plan for contingencies, and try to keep disruptions to the business to a minimum. We have seen systems go live and with management initially unable to get their most frequently used reports this can be a big problem.

The systems project should have a senior manager who acts as the project sponsor. The project should be reviewed periodically by the steering committee to track its progress. This ensures that senior management on down to the department managers are committed to success.

Once you have a plan that makes sense, make sure you manage by the plan. This sounds elementary, but many companies and vendors stumble on it.

Early in the project publish a biweekly status report. Once you get within a few months, you may want to have weekly conference call meetings and status updates. Within 30 days of “go live,” hold daily meetings and list what needs to be achieved.

Project management is the discipline of planning, organizing, motivating, and controlling resources to achieve specific goals. A project is a temporary endeavor with a defined beginning and end (usually time-constrained, and often constrained by funding or deliverables), undertaken to meet unique goals and objectives, typically to bring about beneficial change or added value. The temporary nature of projects stands in contrast with business as usual (or operations), which are repetitive, permanent, or semi-permanent functional activities to produce products or services. In practice, the management of these two systems is often quite different, and as such requires the development of distinct technical skills and management strategies.

The primary challenge of project management is to achieve all of the project goals and objectives while honoring the preconceived constraints. The primary constraints are scope, time, quality and budget. The secondary —and more ambitious— challenge is to optimize the allocation of necessary inputs and integrate them to meet pre-defined objectives.