ISSN No. 0976-5697

Volume 3, No. 5, Sept-Oct 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Data Validation System For A Relational Database

BAAH, Barida
Computer Science Department
University of Port Harcourt
Port Harcourt, Nigeria
baridakara1@yahoo.com

Kabari, Ledisi Giok*
(Member, IEEE)
Computer Science department
Rivers State Polytechnic, Bori, Nigeria
ledisigiokkabari@yahoo.com

Abstract: One of the most neglected areas in systems implementation is data validation. Most application systems fail to achieve its set goals because there is no adequate and efficient data validation system to filter out input data that is not correct. Improper input data is responsible for a large number of system failures; and such failures could have been avoided or reduced if proper input data had been fed into the system. In this paper we focused on enhancing and elaborating on existing data validation methods like *limits checks, character checks, range checks, presence checks, consistency checks, format or picture checks and data type checks,* to improve the performance of our application systems. We also develop a sample program that integrates these validation techniques. A report was finally generated based on validated customer records created for Fractal Construction Nigeria Limited. The system was developed using PHP, MYSQL and XAMPP web server**.**

Keywords: Data validation, database, data base management system, database specification, relational model, normalization

I. INTRODUCTION

Data which is seen as an integral part of any information system, must be handle with much more care in order to ensure that the right information needed for any decision making processes are actually achievable.

To be able to achieve this, there is the need to validate such data before processing, in order to ensure that the right data are entered correctly in their respective field(s) after the program implementation of data validation system for a relational database, which is the purpose for this research work.

Most of the business application, data validation can be defined through declarative, data integrity rules or procedural based rules. Data that does not conform to these rules will negatively affect business process execution.

Therefore, data validity should start with business process definition and set of business rules with this process. It checks to ensure that data are valid, sensible, reasonable and secure before they are processed.

Data validation involves the process of ensuring that a program that is developed operates on a clean, correct and useful data. The validation of this data is applied to a relational database[1].

Today, many business organizations and companies still process their data manually, while others now use automated or computerized systems. The major drawback to manual approach is that it often result in the loss of customers records.

Often, customers may not provide a clean, correct, and useful data while entering data into forms that are given to them by the organization. This is because whatever that is done manually do not have any form of data validation to check data input that is not correct and useful.

On the other hand, organizations that have computerized systems have not integrated properly to allow for good data validation system that can enhance filtering of those data that are not correct in order to ensure clean, correct and useful data.

The important of the study is to ensure that correct data-item are entered in required field(s) before processing of the data, in order to actualize any business or organizational goals in good decision making processes.

II. DATA VALIDATION

Data validation is seen as the process of ensuring that a program operates on clean, correct and useful data. It often uses routines, usually called “Validation Rules” or “check routines”, which does the checking of data that are input to the system. The rules may be implemented through the automated facilities of a ***data dictionary***, or by the inclusion of explicit application program validation logic.

A validation rule is a criterion used in the process of data validation, carried out after the data has been encoded onto an input medium and involves a data vet or validation program. This is distinct from formal verification, where the operation of a program is determined to be that which was intended, and that meets the purpose. The method is to check that data fall the appropriate parameters defined by the systems analyst. A judgment as to whether data is valid is made possible by the validation program, but it cannot ensure complete accuracy.

Most credit cards contain a check digit, which is the digit at the end of the credit card number. The first part of the credit-card number identifies the type of credit card (Visa, MasterCard, American Express, etc.), and the middle digits identify the bank and customer.

There are five most recent approaches that attempt to address the lack of data validation problems, which are application-level gateway approach, client-side encryption approach, WAVES approach and Pixy, and Saner.

A gateway model which is an application-level firewall on a server for checking invalid inputs and detecting malicious script (e.g SQL injection attack and cross-site scripting attack). Have been proposed by Scott and Sharp[2]. They have developed a security policy description language (SPDL) base on XML to describe a set of validation constraints and transformation rules. This language is translated into code by a policy compiler, which sent to a security gateway on a server. The gateway analyzed the request and augments it with a Message Authentication Code (MAC1). Mac is used to protect the data integrity. For example, the MAC is used to secure session information in a cookie at the client-side. There are many ways to compute a MAC such as one-way hash algorithms (MD5,SHA-1) to create a unique fingerprint for data within the cookie.

However, this approach has a number of limitations. One of these is that data tampering is still a potential problem because dynamic data that is generated on fly is not validated. It is also difficult to define all the policies and rules of a legacy web application for every single data entry point.

A client-side encryption system to protect confidential, data integrity, and user trust was also proposed [3]. They encrypt data inputs using a client encryption key before submitting the content of an HTML form. The client encryption key is stored on the server and transferred over an HTTP connection. It uses a one way hash function. The message validation includes finding a new hash value from the decrypted message and comparing it to the hash value which is received with the message. If they are the same, the data is accepted, otherwise, the data is deemed to have been altered and the validation will fail.

Attempt to address data validation issue in the context of PHP applications was made[4]. They used a lattice-based analysis algorithm derived from type systems and type-state systems. The type of analysis in this algorithm is static analysis. WAVES approach is targeted for mitigating threats to web application at the server-side.

Pixy, was the first open source tool for statically detecting XSS vulnerabilities in PHP 4 code by means of data follow analysis which is based on a static analysis technique[5]. They adopted PHP as target language since it is commonly used for developing web applications and substantial number of security advisories refer to PHP programs.

Furthermore, for any business applications, data validation can be defined through declarative data integrity rules, or procedure-based business rules. Data that does not conform to these rules must negatively affect definition and set of business rules within this process. Rules can be collected through the requirements capture exercise.

The simplest data validation verifies that the characters provided come from a valid set. For example, Phone numbers should include the digits and possibly the characters +, - (, and) (plus, minus and parenthesis). A more sophisticated data validation would check to see the user had entered a valid Country code, i.e., that the number of digits entered matched the convention for the Country or area specified.

An incorrect data validation can lead to ***data corruption*** or security vulnerability. Data validation checks to ensure that data are valid, sensible, reasonable, and secure before they are processed.

A. Metods Of Data Validation:

There are several methods of validating data, some of which includes: Allow Character Checks, Batch Total Checks, Cardinality Checks, Others are digit check, consistency check, control total check, cross-system consistency check, data type check, file existence check, format or picture check, hash total check, limit check, logic check, presence check, range check and referential integrity check.

B. Data Validation Techniques:

A general rule is to accept only “Known Good” characters, i.e. the characters that are to be expected. If this cannot be done the next strongest technique is “Known bad”, where we reject all known bad characters. The issue with this is that today’s known bad list may expand tomorrow as new technologies are added to the enterprise infrastructure.

Basically, there are number of models to think about when designing a data validation technique, which are listed from the strongest to the weakest as follows:

a. Exact Match (Constrain)
b. Known Good (Accept)
c. Known Bad (Reject)
d. Encode Known bad (Sanitize)

In addition, there must be a check for maximum length of any input received from an external source, such as a downstream service/computer or a user at a web browser.

III. RELATIONAL DATABASE

A relational database can be defines as a set of tables containing data fitted into predefined categories. Each table (which is sometimes called relation) contains one or more data categories in columns. Each row contains a unique instance of data for the categories defined by the columns. For example, a typical business order entry database would include a table that described a customer with columns for name, address, phone number, and so forth. Another table would describe an order: Product, Customer, date, Sales price, and so forth. A user of a database could obtain a view of the database that fitted the user’s needs. For example, a branch office manager might like a view or report on all customers that had bought products after a certain date. A financial service manager could from the same tables, obtain a report on accounts that needed to be paid.

When creating a relational database, you can define the domain of possible values in a data column and further constraint that may apply to that data value. For example, a domain of possible customers could allow up to ten possible customer names but be constrained in one table to allowing only three of these customer names to be specified..

The standard user and application program interface to a relational database is the structured query language (SQL). SQL statements are used both for interactive queries for information from a relational database and for gathering data for reports.

In addition, to being relatively easy to create and access, a relational database has the important advantage of being easy to extend. After the original database creation, a new data category can be added without requiring that all existing applications be modified.

Relational database matches data using common characteristics found within the data set. The resulting groups of data are organized and are much easier for many people to understand. For example, a data set containing all the real-estate transactions in a town can be grouped by the year the transaction occurred; or it can be grouped by the sale price of the transaction; or it can be grouped by the buyer's last name; and so on.

Such grouping uses the relational model (a technical term for this is Schema). Hence, such a database is called a "relational database". The software that is use in grouping is called a relational database management system(RDMS). The term "relational database" often refers to this type of software.

The relational database is currently the predominant choice in storing financial records, manufacturing and logistical information, personnel data and much more.

Strictly, a relational database is a collection of relations (frequently called tables). Other items are considered part of the database, as they help to organize and structure the data, in addition to forcing the database to conform to a set of requirement.

A. Relational Model:

Relational model is today the primary data model for commercial data-processing applications. It has attained its primary position of its simplicity, which eases the job of the programmer, compared to earlier data models such as network model or the hierarchical model.

The relational model specifies that the tuples of a relation have no specific order and that the tuples, in turn, impose no order on the attributes. Applications access data by specifying queries, which use operations such as select to identify tuples, project to identify attributes, and join to combine relations. Relations can be modified using the inset, delete, and update operators. New tuples can supply explicit values or be derived from a query. Similarly, queries identify tuples for updating or deleting. It is necessary for each tuples of a relation to be uniquely identifiable by some combination (one or more) of its attribute values. This combination is referred to as the primary key.

B. Base and Derived Relations:

In a relational database, all data are stored and accessed via relations. Relations that stored data are called "Base Relation", and in implementations are called "tables". Other relations do not store data, but are computed by applying relational operations to other relations. These relations are sometimes called "derived relations". In implementations these are called "view" or "queries". Derived relations are convenient in that though they may grab information from several relations, they art as single relation. Also, derived relations can be used as an abstraction layer

C. Relational Operators:

Queries made against the relational database, and the derived relvars in the database are expressed in a relational calculus or a relational algebra, Codd introduced eight relational operators in two groups of four operators each[6]. The first four operators were based on the traditional mathematical set operators:

a. Union Operator: This is use to combine two relations and remove all duplicate tuples from the result. The relational union operator is equivalent to the SQL UNION operator.
b. Intersection Operator: This operator produces the set of tuples that two relations share in common. Intersection is implemented in SQL in the form of the INTERSECT operator.
c. Difference Operator: The difference operator acts on two relations and produces the set of tuples from the first relation that do not exist in the second relation. Difference is implemented in SQL in the form of EXCERT or MINUS operator.
d. Cartesian Product: The Cartesian product of two relations is a join that is not restricted by any criteria, resulting in every tuple of the first relation being matched with every tuple of the second relation. It is a Cartesian product is implemented in SQL as the CROSS JOIN join operator.
e. Selection or Restriction Operation: The selection or restriction operation retrieves tuples from a relation, limiting the results to only those that meet a specific criteria, i.e. a subset in terms of set theory. The SQL equivalent selection is the SELECT query statement with a WHERE clause.
f. Projection Operation: This operation retrieves tuples containing only the specified attributes.
g. Join Operation: The join operation defined for relational databases is often referred to as a natural join. In this type of join, two relations are connected by their common attributes. SQL's approximation of a natural join is the INNER JOIN operator.

Relational Division Operation: The relation division operation is a slightly more complex operation, which involves essentially using the tuples of one relation (the dividend) to partition a second relation (the divisor). The relational division operator is effectively the opposite of the Cartesian product operator

D. Normalization:

Normalization was first proposed by Codd[6] as an integral part of the relational model. It encompasses a set of best practices designed to eliminate the duplication of data, which in turn prevents data manipulation anomalies and loss of data integrity. The most common forms of normalization applied to databases are called the normal forms. Normalization is criticized because it increases complexity and processing overhead required to join multiple tables representing what are conceptually a single item.

E. Relational Database Management System:

A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a table represents a relationship among a set of values. A relational database as implemented in relational database management system, have become a predominant choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data and much more. Relational databases have often replaced legacy hierarchical databases and network databases because they are easier to understand and use, even though they are much more less efficient as computer power has increased, the inefficiencies of relational database, which made them impractical in earlier times, have been outweighed by their ease of use.

However, relational databases have been challenged by object databases, which were introduced in an attempt to address the object-relational impedance mismatch in relational database, and XML databases.

The three leading commercial relational database vendors are Oracle, Microsoft and IBM. The three leading open source implementation are MYSQL, PostgreSQL and SQLite.

IV. RESEARCH METHODOLOGY

The methodology that is beseech in this research work is called the structural system analysis and design method which is a waterfall method for the production of an information system design. SSADM can be thought to represent a pinnacle of the rigorous document-led approach to system design.

Structural System Analysis and Design Method (SSADM) is a systems approach to the analysis and design of information systems. one particular implementation of structural system analysis and design method which is builds on the work of different schools of structured analysis and development methods, such as Peter Checkland's, software system methodology, Larry Constantine's Structured Design, Edward Yourdon's Structured Method, Michael A Jackson's, Jackson Structured Programming and Tom DeMarco's Structured Analysis[7].

The reasons of chosen this research methodology is due to the following advantages:

a. The SSADM is mature
b. SSADM provide a clear separation of logical and physical aspects of the system.
c. It is well-defined techniques and also well documented.
d. It also provides an environment for the user involvement also.

A. Database Architecture:

Database architecture consists of three levels, *external*, *conceptual* and *internal*. Clearly separating the three levels was a major feature of the relational database model that dominates 21st century databases.

The external level defines how users understand the organization of the data. A single database can have any number of views at the external level. The internal level defines how the data is physically stored and processed by the computing system. Internal architecture is concerned with cost, performance, scalability and other operational matters. The conceptual is a level of indirection between internal and external. It provides a common view of the database that is uncomplicated by details of how the data is stored or managed, and that can unify the various external views into a coherent whole.

B. Relational Database Characteristics:

The fundamental of a relational database characteristics are:

a. The internal structure of an operating database is basically fixed in the row direction and the user will interact with a logical view of the data and need not know anything about the actual internal structure.

C. Models of the System:

Given in figure1 is a simple data flow diagram that depicts the process of procurement system.

D. Main Menu:

Given in figure2 is the initial view of the proposed home page of the web application (i.e. control centre).

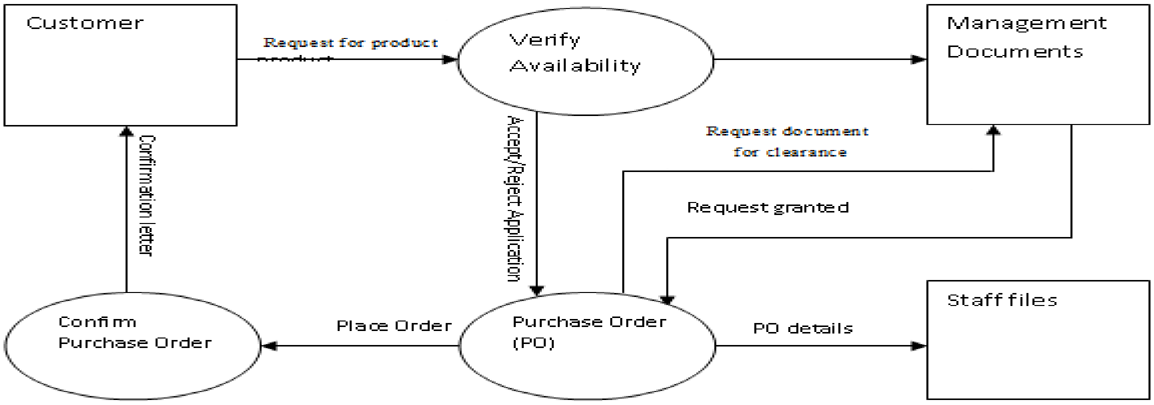
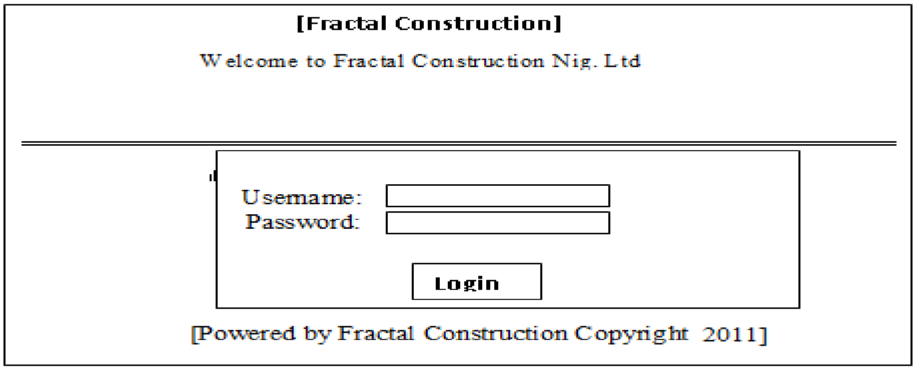


Figure1: A simple data flow diagram of procurement process



[Fractal Construction]

Welcome to Fractal Construction Nig. Ltd

Username:

Password:

Login

[Powered by Fractal Construction Copyright 2011]

Figure2: Sample main menu design

E. *Specifications:*

In this paper, we used a relational DBMS called MySQL. The name of the database is called fractalConstruction.sql. It contains one major table-customers. Other tables are supportive in nature. The basic table structure is shown below in table1.

It's representation in SQL is given as:

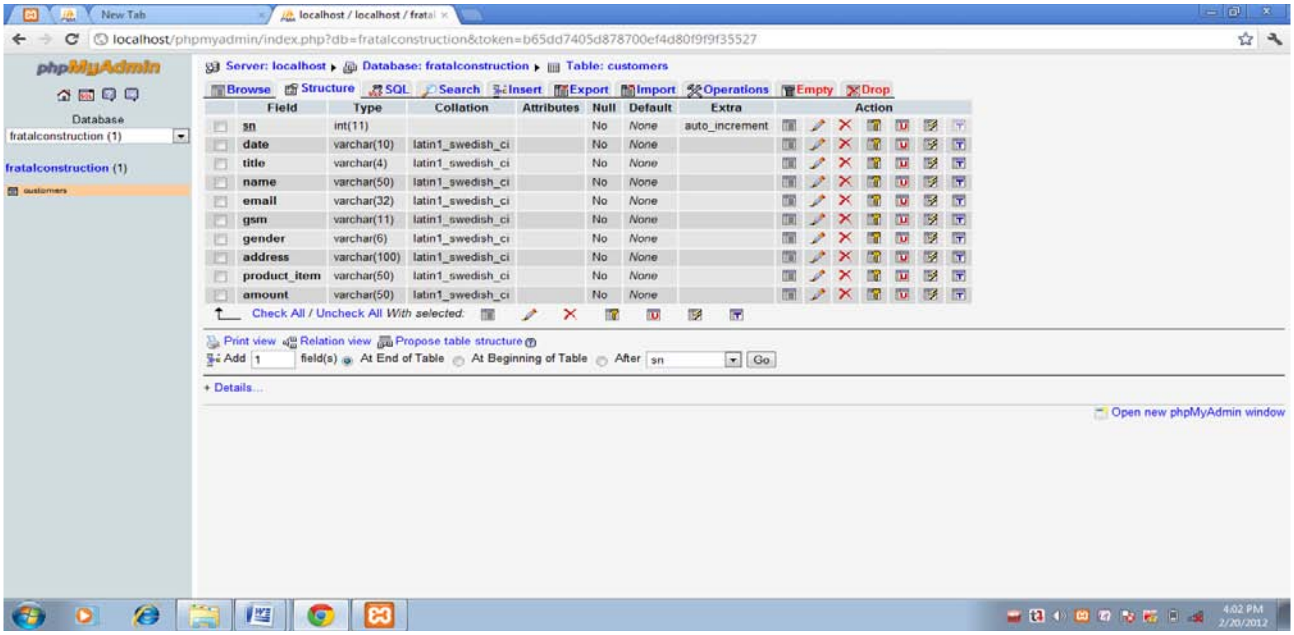
```
CREATE TABLE IF NOT EXISTS `customers` (
  `sn` int(11) NOT NULL AUTO_INCREMENT,
  `date` varchar(10) NOT NULL,
  `title` varchar(4) NOT NULL,
  `name` varchar(50) NOT NULL,
  `email` varchar(32) NOT NULL,
  `gsm` varchar(11) NOT NULL,
  `gender` varchar(6) NOT NULL,
  `address` varchar(100) NOT NULL,
  `product_item` varchar(50) NOT NULL,
  `amount` varchar(50) NOT NULL,
  PRIMARY KEY (`sn`)
)   ENGINE=MyISAM    DEFAULT    CHARSET=latin1
AUTO_INCREMENT=1 ;
```

The application will force users to login on the top left-corner of the application with their username and password. Once the login is successful, the user will be presented with a page containing links to perform several operations available for the user.

Table1 : Customer Table

Field	Type	Null
sn	int(11)	No
date	varchar(10)	No
title	varchar(4)	No
name	varchar(50)	No
email	varchar(32)	No
gsm	varchar(11)	No
gender	varchar(6)	No
address	varchar(100)	No
product_item	varchar(50)	No
amount	varchar(50)	No

The structure of the table as captured from XAMPP, the DBMS used in designing the database is shown in figure3.



New Tab
localhost / localhost / fratal
localhost/phpmyadmin/index.php?db=fratalconstruction&token=b65dd7405d878700ef4d80f9f9f35527
phpMyAdmin
Database
fratalconstruction (1)
fratalconstruction (1)
customers
Server: localhost ▸ Database: fratalconstruction ▸ Table: customers
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
□	sn	int(11)			No	None	auto_increment	
□	date	varchar(10)	latin1_swedish_ci		No	None		
□	title	varchar(4)	latin1_swedish_ci		No	None		
□	name	varchar(50)	latin1_swedish_ci		No	None		
□	email	varchar(32)	latin1_swedish_ci		No	None		
□	gsm	varchar(11)	latin1_swedish_ci		No	None		
□	gender	varchar(6)	latin1_swedish_ci		No	None		
□	address	varchar(100)	latin1_swedish_ci		No	None		
□	product_item	varchar(50)	latin1_swedish_ci		No	None		
□	amount	varchar(50)	latin1_swedish_ci		No	None		

Check All / Uncheck All With selected:
Print view
Relation view
Propose table structure
Add 1 field(s) ● At End of Table ○ At Beginning of Table ○ After sn
Go
+ Details...
Open new phpMyAdmin window
4:02 PM
2/20/2012

Figure3: Screenshot of Database created

F. Input Specification:

Input design addresses the design of the input forms which would be used to collect information from the user for input into the computer. In this system, data can be collected online directly and entered into the database or from source documents. Given below is a sample input form for capturing user input.

G. Output Specification:

Outputs present information to system users. Outputs are the most visible component of a working information system, and are the basis for the user's and management's final assessment of the systems value. The output can be seen directly on the computer screen or printed out on paper.

H. Enhance Data Validation Method:

We were able to design an improved data validation system for better performance through combining or merging two or more existing validation techniques to ensure the consistency of data, particularly we combines exact match techniques with the presence check techniques to test the title field. Assuming a user has selected Male (which satisfies the exact match technique), then the user must also select "Mr." to be consistent with the gender information that is required.

Also, the allow-combine characters check is the new method of data validation that we developed from the existing allow character check which allow for a check of just a single character like "@", dot (.) etc. but in the newly developed allow combined character checks tries to allow for the checks of more than one characters for example in the GSM field which is a numeric field will test to allow combine set of characters for a network, particularly the first 3 digits, if it is correctly entered or not e.g. 080,070,081 etc.

I. System Requirement:

I recommend the following computer system requirements:
- Pentium 4, 2.0 GHz Processor or higher
- Windows XP, Vista or Windows 7
- 1 GB RAM or higher
- High-Speed Internet connection (Dial-up is NOT recommended)
- Sound card and speakers

A monitor capable of at least 800x600 resolutions

V. CONCLUSION

Data validation is actually an important aspect when it comes to the design of a relational database, since it has to do with validation of client side or end user to ensure that only clean, correct and useful data are accepted while those data that are not useful to the relational database system are rejected by the display of an error message to alert the user or client while entering data into the database system.

This application was created using PHP and MYSQL web technology model.

In this paper we have been able to establish that every system should incorporate a data validation system to filter improper input data.

We have developed and also enhanced existing data validation techniques. These techniques are a product of combining two or more existing validation techniques to ensure the consistency of data. We have also proposed that certain checks can also be made to be dependent on others.

VI. RECOMMENDATION

The data validation system that is implored here in this paper is purely the client side of validation which is at the end-user input. Future research could focus on validation at the database level.

It is recommended that the local machine used to development web application should have the same general features and capabilities of the server(i.e. the remote machine) on which the final solution will finally be deployed. However, if the fractal construction limited that uses the system plans to outsource the maintenance of the application to another person or company, then proper considerations for mimicking the setup of the development server on the staging server can save a great amount of time.

VII. REFERENCES

[1]. M. Arkady "Data Quality Assessment", Technics Publication, LLC(2007), pp.1-2

[2]. D. Scott and R. Sharp "Specifying and enforcing application-level web security policies", IEEE knowledge Data Engineering, vol. 15, no. 4(2003), pp. 771-783.

[3]. M. Hassinen and P. Mussalo "Client controlled security for web applications", Proceedings of the IEEE Conference on Local Computer on World Wide Web, New York, NY, USA, ACM Press(2005) pp. 215-224.

[4]. Y. Huang, S. Huang, T. Lin and Tsai "Web application security assessment by fault injection and behavior monitoring, Proceedings of the 12th International Conference on World Wide Web, New York, NY,USA, ACM Press(2003), pp. 148-149.

[5]. N. Jovanovic, C. Kruegel and E. Kirda E. "Pixy, A Static Analysis Tool for Detecting Web Application Vulnerabilities(Short Paper)", Proceedings of the 2006 IEEE symposium on Security and Privacy, Washington, DC, IEEE Computer Society, pp. 258-263.

[6]. E. F. Codd "A Relational Model of Data for large Shared Data Bank", Communications of the ACM 13(6)(1970): pp. 377-387.

[7]. G. Mike and R. Karel"History of SSADM, SSADM an Introduction"(1999). pp. 2-5.