Republic of the Philippines

**University of the Philippines Cebu**

Gorordo Ave., Lahug, Cebu City 6000

**Digital Audit Trail: A Web Application on the**

**Automation of Transaction Audit Trail in**

**Accounting Information Systems (AIS)**

A Thesis Draft

Submitted to

Prof. Demelo M. Lao

Adviser

In partial fulfillment of the requirements for

CMSC 198.2 (Special Problem)

Submitted by

Kristian Jacob Abad Lora

April 02, 2013

**Table of Contents**

**Table of Contents i**

**List of Figures ii**

**List of Tables iii**

**Chapter I: Introduction 1**

Background 1

Statement of the Problem 2

Objectives of the Study 3

Definition of Terms 4

Scope and Limitations 6

**Chapter II: Review of Related Literature and Works 7**

The Audit Trail Process 7

The Digital Audit Trail 8

Sampling Techniques 10

Related Works 16

**Chapter III: Methodology 17**

Web Application Development 17

Audit Trail Process 18

Sampling Method 18

**Bibliography 23**

**List of Figures**

**Figure 1:** Tracing in Manual Audit Trail 5

**Figure 2:** Transaction Audit Trail in Computerized AIS 7

**Figure 3:** MVC Diagram 16

**List of Tables**

**Table 1:** Factors Influencing Sample Sizes for a Test of Details in Sample Planning 9

**Table 2:** Examples of Homogenous Groups 11

**Table 3:** Examples of Sample Types 12

**Table 4:** Recommended Sample Sizes Based on Sampling Units 13

CHAPTER I

**INTRODUCTION**

**Background**

For years, the Philippines is known to be one of the most corrupt countries in both Asia and the world. In a 2011 report of Political & Economic Risk Consultancy (PERC), a Hongkong-based institution, the Philippines ranked as the fourth most corrupt country in Southeast Asia while it ranked 134th least corrupt country among 180 countries under the 2010 Corruption Perception Index of Transparency International, a global movement against corruption [1-2].

It is, on the other hand, one role of accounting information system (AIS) to combat and prevent corrupt practices in public and private sectors by enhancing transparency. However, many AIS are vulnerable to fraud. For instance, a survey conducted by Asian Foundation resulted to revealing disturbing practices within the private sector such as the practice of double bookkeeping. Only 21% of them said “‘almost all’ companies in their line of business keep a single set of accounting books” [3].

Learning from this vulnerability, people thought of automating the procedures of accounting. In the late 1960s, various computerized AIS applications were released into the market and began to be used in business accounting [4]. These applications did not just automate the collection of accounting data and generation of accounting information but also automated the protection of the data and detection of fraud using internal system controls and other mechanisms. One of these is the audit trail.

An audit trail is a piece of evidence that provides history of activities in the accounting information system of an entity [5]. It provides information on who entered transaction entries into the system and when these are entered and who committed changes to the entries, what entries were modified and when these were modified. This, known as *system audit trail*, is just one type of audit trail in the context of accounting. The other type is *transaction audit trail*. This type focuses on tracing the economic event *per se*. It can show, step-by-step, the journey of an economic transaction from source document(s) to product documents (e.g. financial statements)*,* and vice-versa [6].

Because of these features, private establishments became patrons of the high-technology software but not the governments, public institutions and small-and-medium enterprises (SMEs) because it is costly for them to purchase such tools. It was at a later date when governments began computerizing their accounting information systems as one step to preventing corruption. The use of computerized AIS became more popular in governments and public institutions when, in the late 1980s, the World Bank “funded financial management information system(FMIS) projects as part of larger government reforms or as a stand-alone project in more than 27 countries totaling US $ 1.1 billion” [7]. In the Philippines, it was only in 2003 when the Commission on Audit (COA), the Philippine government’s auditing body, computerized their AIS, in-house [8].

With the introduction of computerized AIS, corruption was expected to have been lessened, if not eliminated. It was thought to have the ability to detect patterns of suspicious activity and provide security for transaction data, among others [9]. However, people later realized that the computerization only brought a dismal effect to the eradication of corruption. “The myth of computer omnipotence and assumption that ICT removes the opportunities for corruption” [9] brought complacency among the developers, overlooking the possibility of computer fraud. Even with the internal system controls such as audit trail, internal and external persons of an entity can still commit fraud without being detected or traced through hacking or bribing.

These occurrences imply that even with the sophistication of AIS, the need of external auditing has never been eliminated. Though most of the entities have internal auditors, “internal auditors can't effectively critique the company's internal processes because they are part of the company” [10]. Contrarily, external auditing is conducted by third-party accountants; thus, can provide, with high chances, unbiased audit opinion [11]. Thus, external auditing also plays a vital role in overseeing finances and in detecting financial irregularities and fraud in an entity.

**Statement of the Problem**

Just like AIS, in the advent of technology, recent developments also include computerization of auditing to aid external auditors in generating more accurate and timely audit opinion. Meanwhile, in the Philippines, one can rarely find computerized auditing software, “given the strength of the national software industry” [12]. Although there are auditing software that can be downloaded or bought abroad, these are not flexible enough to comply with the generally accepted auditing standards (GAAS) in the Philippines [13]. Hence, this study intends to automate the auditing process in accordance to Philippine standards. However, since building a computerized system for the entire process of auditing takes time, this study chooses to develop an automation of a major part of the auditing process – the transaction audit trail.

Along with the pursuit of automating this type of audit trail, this study also seeks for answers to the following questions:

* What are the Philippine auditing standards on transaction audit trail?
* How effective is an automated transaction audit trail compared to manual transaction audit trail in terms of accuracy and efficiency?
* Will accountants readily accept the system?

**Objectives of the Study**

In general, this study aims to develop a digital transaction audit trail in accordance to Philippine auditing standards. Specifically, it aims to:

* Develop a digital transaction audit trail that may become a stepping stone towards building a full-blown auditing software suitable to Philippine auditing process;
* Develop a digital transaction audit trail that is cross-platform;
* Measure the accuracy and efficiency of the system;
* Validate the acceptability of the system using the Technology Acceptance Model (TAM)

**Scope and Limitations**

This scope of the system to be developed is limited to conducting audit trail to small-scale retail stores having: a minimal number of transactions, that is, an average of 20 transactions per day; two to five employees; and, an average gross income of P450, 000.00 per annum. Since this is a development from scratch, the features of the system are constrained to the following limitations:

* Only vouching (backward tracing) is supported, mainly, due to time-constraints. Nonetheless, vouching is the most widely used approach in audit trail
* Only uses one sampling technique that falls under a statistical approach to identify the sample size since the non-statistical approach involves more judgment of the auditor; thus, the need to have an ample knowledge on Artificial Intelligence to capture the judgment, which the developer does not have enough

**Definition of Terms**

*Accounting Information System* – set of interrelated components that collects data on financial transactions and certain non-financial transactions that affect the processing of financial transactions and process them into information which are distributed to intended users to guide them in their next course of action

*Agile Software Development* – an approach in software development that uses iterative and incremental software development methodologies

*Audit Trail* – piece of evidence that provides history of activities in an accounting information system

*Digital Audit Trail* – computerized (automated) audit trail

*Fraud* – in the context of accounting, is an intentional deception, misappropriation of company’s assets, or manipulation of its financial data to the advantage of the perpetrator

*General Ledger* – compilation of all ledger accounts; shows activity for each account listed on the chart of accounts; also known as the “book of final entry”

*Journal­ –* a chronological record of transactions; also known as the “book of original entry”

*Product Document* – an output document of transaction processing; this includes financial statements such as the balance sheet

*Sampling* – process by which a sample of transactions is pulled from the population of transactions using sampling techniques

*Relational Database Management System* – a database management system that allows relationship among data

*Source Document* – used to capture and formalize transaction data needed for transaction processing; this includes cash receipts, vouchers, invoice, etc.

*Subsidiary Ledger* – a ledger that shows activity, by detail, for each account type

*System Audit Trail* – an audit trail that shows the activity of users in a system such as who committed changes to the system, what these changes are and when are these modified

*Tracing* – an approach of transaction audit trail that involves tracing from source document to product document

*Transaction* – an event that affects or is of interest to the organization and is processed by its information system as a unit of work

*Transaction Audit Trail* – an audit trail that traces the journey of a transaction from a source document to a product document, or vice-versa; for brevity, “transaction audit trail” and “audit trail” are used interchangeably in this study

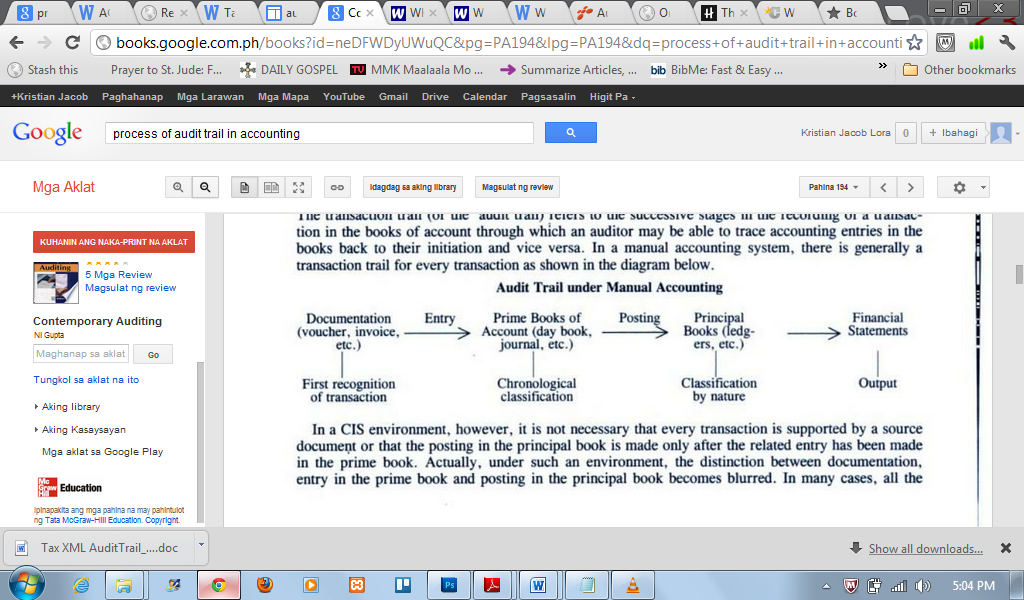
*Vouching* – an approach of transaction audit trail that involves tracing from product document down to source document

CHAPTER II

**REVIEW OF RELATED LITERATURE AND WORKS**

**The Audit Trail Process**

Auditors validate the entries reported on financial statements by four criteria: (1) existence, (2) ownership, (3) completeness and (4) valuation. Two of these criteria are the focus of an audit trail. An audit trail checks that all transactions are properly recorded (completeness) including the correctness of the amount and that all balances "are made up of things that actually exist (existence)" [14]. There are two primary tests used on financial data to test these criteria: (1) tracing for completeness by following the journey of a transaction from the source document (official receipt, vouchers, etc.) to product document (financial statements) and (2) vouching for existence by backtracking from product document to source document [14].



*Fig. 1. Tracing in Manual Audit Trail*

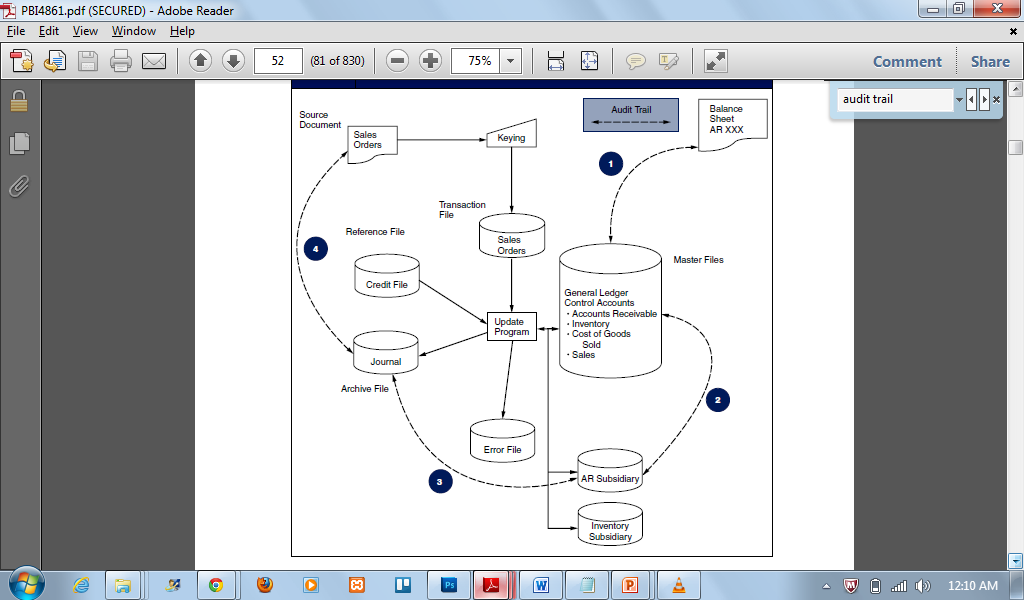
Under a manual audit trail, tracing starts from recognizing a transaction through source documents (Fig. 1). A transaction is then checked if it is posted on the book of original entry (its respective journal). Afterwards, the auditor checks if the journal entry of such transaction is posted on the book of final entry (general ledger or subsidiary ledger, if any). If such transaction has a subsidiary ledger, the auditor checks if the balance of that subsidiary ledger, along with the balances of subsidiary ledgers alike, is posted to the general ledger. Finally, the total of the balances of these subsidiary ledgers appearing in the general ledger is checked if it is posted on the financial statements [5].

Meanwhile, vouching works in an opposite fashion. Starting from the product document, say for example, a balance of an account in an income statement is compared with its corresponding balance in the general ledger. If subsidiary ledgers exist for such general ledger, the balance is compared with the total amount of balances of these subsidiary ledgers. A sample that is representative enough of the entries made to the accounts in each subsidiary ledger is then selected. Afterwards, these entries are checked if they have corresponding transactions in the journal in which they are posted. From these journal entries, respective source documents are identified. Some auditors contact the persons or entities transacted with whose names appear on the source documents to verify the existence of such transaction and the accuracy of the particulars including the amounts. This technique is known as "confirmation" [6].

Many of audit trail practices use vouching. “Often, beginning the audit process with the most recent completed phase and working backward is an efficient means of establishing the audit trail” [15]. Nonetheless, accountants should be able to “trace in both directions” using the two tests [6].

**Digital Audit Trail**

For computerized AIS, the audit trail traverses through four files: (1) the master file that contains the account data; (2) the transaction file which is a temporary file that contains transaction records used to change or update data in the master file; (3) the archive file that contains past transaction data that are retained for future reference as these transactions form an important part of the audit trail; and, (4) the reference file that contains data used as standards (lookup) for processing transactions [6].



*Fig. 2. Digital Transaction Audit Trail*

As shown in Figure 2 above, the digital audit trail begins by comparing a balance of an account appearing in a financial statement with the balance in the master file (general ledger or subsidiary ledger) to which such account falls under. If subsidiary ledgers exist, the balance of the account in question in the general ledger is compared with the total amount of the balances of the subsidiary ledgers concerned. A sample of entries is then selected from the entries of each subsidiary ledger and these entries are traced back to their corresponding journals (archive file). Specific source documents (orders, vouchers, official receipts, etc.) for these journal entries are retrieved from the transaction file. Upon reading the name of the person transacted with that appears in each source document, the audit trail looks up to the reference file to retrieve the information of that person, especially, his or her contact number. Afterwards, either the auditor or the computer itself calls that person, to verify the existence of the transaction and the accuracy of the particulars and their corresponding amount [6].

**Sampling Techniques**

For AIS having large volume of transactions, it is never necessary to go through all transactions in the audit trail but only a representative sample of them. “Standard scientific sampling techniques are employed, consistent with Generally Accepted Accounting Principles, to derive a sample population of the strata. Transactions are selected on a random basis throughout each stratum. The result is a sample that represents all the transactions within audit population” [16].

There have been a number of sampling techniques used for auditing and these techniques can be approached into two: statistical and non-statistical. These techniques, however, must be in accordance with generally accepted and recognized sampling technique standards such as SAS (Statement of Auditing Standards) No. 39 and the Audit Sampling of the American Institute of Certified Public Accountants (AICPA). Among the standards, SAS No. 39 is used primarily [17].

Under SAS No. 39, both statistical and non-statistical approaches require professional judgment in planning, performing and evaluating the sample. The size of the sample depends on the objectives and efficiency of the sample. “One sample is more efficient than another if it can achieve the same objectives with a smaller sample size” [17]. There is a sampling risk (the possibility that a sample is not representative of the whole population [18]), however, when the auditor’s conclusions are different from the conclusions he would arrive if same test is applied to other entries if a test of controls is restricted to a sample. Moreover, the risk is inversely proportional to the sample size. This means that the smaller the sample size, the higher the sampling risk is [17].

Table 1. Factors Influencing Sample Sizes for a Test of Details in Sample Planning

|  |  |  |  |
| --- | --- | --- | --- |
| **Factor** | **Conditions leading to** | | **Related factor**  **for substantive**  **sample planning** |
| **Smaller Sample Size** | **Larger Sample Size** |
| Assessment of inherent risk. | Low assessed level  of inherent risk. | High assessed  level of inherent  risk. | Allowable risk of  incorrect  acceptance. |
| Assessment of  control risk. | Low assessed level  of control risk. | High assessed  level of control  risk. | Allowable risk of  incorrect  acceptance. |
| Assessment of  risk for other  substantive  procedures  related to the  same assertion  (including  substantive  analytical  procedures and  other relevant  substantive  procedures). | Low assessment of  risk associated with  other relevant  substantive  procedures. | High assessment  of risk associated  with other  relevant  substantive  procedures. | Allowable risk of  incorrect  acceptance. |
| Measure of  tolerable  misstatement for  a specific account. | Larger measure of  tolerable  misstatement. | Smaller measure  of tolerable  misstatement. | Tolerable  misstatement. |
| Expected size  and frequency of  misstatements. | Smaller  misstatements or  lower frequency. | Larger  misstatements or higher frequency. | Assessment of  population  characteristics. |
| Number of  items in the  population. | Virtually no effect on sample size unless  population is very small. | | |
| Choice between  statistical and  non-statistical  sampling | Ordinarily, sample sizes are comparable. | | |

Statistical approach uses tables or statistical formulas to compute the sample size based on factors such as “the tolerable misstatement and the expected misstatement (of transactions), the audit risk, the characteristics of the population, the assessed risk of material misstatement (inherent risk and control risk), and the assessed risk for other substantive procedures related to the same assertion” (see Table 1) [19]. The sampling risk is measured using the law of probability; hence, the methods in this approach are variants of probability sampling like equal probability sampling such as simple random number and systematic sampling, and sampling with probability proportional to size such as monetary unit sampling [20]. It helps in designing an efficient sample, measuring the sufficiency of the audit evidence produced by the trail, and in evaluating the results. It also helps in measuring the sampling risk for an auditor to know how to limit it into a level that is acceptable [19]. Moreover, it is appropriate in drawing a conclusion about a population without “performing an examination of all the items composing that population”, especially, when an auditor has no “prior knowledge as to which specific items in a population are misstated” [20]. Nonetheless, it must be noted that even though statistical approach uses statistical methods, it does not necessarily eliminate the need to exercise judgment of an auditor [17].

Meanwhile, in non-statistical approach, the sample risk is not measured [20], the sample size and sample evaluation are determined and done, respectively, by professional judgment, and the sample is selected haphazardly [18], that is, “by selecting, without any conscious bias, items regardless of their size, source, or other distinguishing characteristics” [21]. In other words, pure yet careful judgment is the sole basis of this approach; thus, commonly used methods in this approach are haphazard, block and judgment sampling [17] [21]. However, block number is rarely used because it does not satisfy the requirements for a representative sample [21]. Non-statistical approach is mostly preferred by auditors due to “the cost of training, the cost of sample selection, the cost of sample evaluation” that will be incurred when using the statistical approach [20].

Table 2. Examples of Homogenous Groups

|  |  |
| --- | --- |
| **Major Categories** | **Specific Categories** |
| **Outlets or locations** | Metro vs. rural  Opened for the entire audit period vs. opened for only part of the audit period  Mall store vs. free-standing store |
| **Divisions** |  |
| **Departments** | Production vs. non-production departments  Corporate office vs. sales branch |
| **Product Lines** |  |
| **Type of Customer** | Reseller vs. end user  Cash vs. credit  Manufacturer  Contractor  Farmer  Prices  High dollar items vs. low dollar items |
| **Time periods** | Law changes  Personnel or management changes  Accounting changes |
| **General Ledger Accounts** | Expenses vs. assets  Inventory vs. expense |
| **Accounting Treatment** | Cash vs. accrual  Taxable vs. non-Taxable  Capitalized vs. expensed  Tax accrued vs. tax paid  Journal entry vs. detailed entry  Computerized entry vs. manual entry |

Despite the differences of the two approaches, both statistical approach and non-statistical approach follow general guidelines in sampling. The guidelines are classified into three major phases: (1) planning; (2) performance; and (3) evaluation and may include documentation [18] [22]. The planning phase constitutes: (1) defining the audit objective like whether to educate the client or determine errors; (2) testing data integrity, that is, assessing if all data needed for auditing are present; (3) defining the population and stratifying it into sub-populations (homogenous groups), if needed (See Table 2) SOURCE; (4) identifying the sampling unit (is it a voucher, a check or block of transactions); (5) identifying type of sample based on the sampling unit (see Table 3) [23]; and (6) determining the sample size based on the type of sample (see Table 4) [17]. Meanwhile, the performance phase involves the (7) sampling proper which is the selection of sampling unit using the commonly used random, systematic or haphazard (judgmental) selection methods. Evaluation phase evaluates the sampling (8) if the selected items are representative of the whole population, (9) examines for errors and inconsistency and (10) draws conclusion. The findings are then documented, if necessary [18].

Table 3. Examples of Sample Types

|  |  |
| --- | --- |
| **Sample Type** | **When to Use** |
| **Stratified Random Sampling** | When the data is provided in a manner that will facilitate stratifying the population. Usually, the records are available electronically and the auditor will obtain assistance from the Computer Audit Analyst. The more variable the population being sampled, the greater is the need for stratified random sampling. |
| **Transaction Sampling** | When the sampling unit is a line item or an individual invoice. This method allows for selected transactions to be dispersed throughout the audit period, which allows for a more representative sample. This is the preferred method of sampling because it is usually the smallest usable unit but is not always the most practical. For example, when invoices are filed in vendor order, locating the selected transactions requires more work. |
| **Cluster Sampling** | When a contiguous group of transactions are present such as vouchers, boxes, and ledger pages. The clusters contain equal numbers of transactions in each group and are not overlapping. Cluster sampling may be feasible when there is a high cost of retrieving items randomly selected across the entire population or when there is a sample size of more than 1000 units. This method can save time because the sample is pulled from fewer locations within the population. To illustrate, a sample of 1000 transactions requires that the items be pulled from 1000 physical locations. A sample of 40 clusters of 25 items each requires the auditor or taxpayer pull records from only 40 different locations (boxes, files, etc.). Cluster samples are more efficient than block sampling (time period) because the auditor controls the size of each sample unit and all clusters contain the same number of individual transactions. Instead of reviewing all transactions falling in a given time period, the auditor may randomly select between 30 and 80 even clusters. |
| **Block Sampling (Time-Period Sampling)** | When transactions that fall within a unit (days, weeks, months, or years). While the block method is similar to the cluster method, block samples are normally not equal in size. Therefore, the number of blocks necessary for a reliable sample is difficult to know. This method produces a sample larger than necessary to produce reliable results because the blocks may vary in size by a considerable amount. In block sampling, the auditor assumes the sample time period is representative of the entire population. By not taking sample transactions over the entire audit period, block samples increase sampling risk. If the tax error ratio in the sample time period differs significantly from the time periods not sampled, the block sample will produce results that are not valid. |

Table 4. Recommended Sample Sizes Based on Sampling Units

|  |  |  |
| --- | --- | --- |
| **Sampling Unit** | **Recommended Sample Size** | |
| **Transaction – Dollar Stratified** | Minimum of 100 items per stratum | |
| **Transaction – Manual** | Minimum of 250 “items of interest” (i.e., non-taxed Texas sales, or purchases charged to accounts with potential tax adjustments) | |
| **Cluster** | Minimum of 30 to 80 clusters approximating 250 “items of interest” for all clusters. Each cluster should be equal in size (units or dollars). | |
| **Time Period** | ***Regular Time Period*** | ***Outlet Time Period \**** |
| Days/Batches | Minimum of 30 for each outlet | Minimum of 80 for all combined outlets |
| Weeks | Minimum of 20 for each outlet | Minimum of 50 for all combined outlets |
| Months | Minimum of 6 for each outlet | Minimum of 30 – 80 for all combined outlets depending on the number of outlets |

**Related Works**

Many of the existing computerized Accounting Information Systems support transaction audit trail like the Sage Simply Accounting Premium software of the Digital Steer IT Solutions (http://www.digitalsteer.ph), the Front Accounting online AIS (http://www.frontacounting.com), and the accounting software of Balmori Software, Inc. (http://www.balmorisoftware.com), to name a few. However, the audit trail that these systems employ merely works by showing all transactions for a general ledger account in a period of time, both specified by the user. Their audit trails do not satisfy the audit trail defined in this study that involves tracing transactions, sampling or confirmation.

CHAPTER III

**METHODOLOGY**

This study used the Software Development Life Cycle (SDLC) in developing the system. This systematic model is comprised of five critical phases elaborated in this section that were taken to ensure that the system is delivered on time with its quality being maintained. Moreover, the system was developed as web-based because web applications are more portable compared to desktop applications, that is, it can be used in any operating system (OS).

**System Analysis**

Researches were done to seek for elaborated steps of transaction audit trail but all of the literatures sought give the same result – an overview of the transaction audit trail process. It is for this reason that the researcher interviewed some accountants to get detailed steps of the audit trail. With the information gathered, the major requirements of the system are the following:

* The system must be able to accept an electronic file in CSV (comma-delimited file) format
* The system must be able to trace transactions from financial statements to source documents
* The system must be able to perform the three major tasks of the audit trail process:
  + Footing – checks if the total amount displayed in a financial document matches the total amount displayed in the next financial document (e.g. the Sales amount displayed in an income statement matches the total amount displayed in its corresponding ledger, so on and so forth)
  + Matching – checks if the total amount displayed in a financial document matches the calculated total amount in the next financial document
  + Vouching – checks the existence of the transaction in question from financial statements down to source document(s)
* The system must be able to generate a summary of the trail results
* The system must be able to give the user an option to contact the person transacted with, if possible, for further validation and confirmation

*Use Cases*

The actor of the following use cases is basically an accountant, specifically, an auditor.

Use Case #1: System Setup

This use case describes how the system is set up before a user can utilize the features of the system. The Setup page records information about the user and about the files needed by the system. It also sets up an account for a user that he/she will use for user authentication.

1. Preconditions
   1. The system has not been used yet and the user has still no account in the system.
   2. The database of the system is non-existent
2. Interaction Scenarios
   1. The use case begins when the user accesses the system.
   2. Upon first access, the Setup pages – one page for user information and another page for the location of the files needed for the trail – appears.
   3. Use Case: Setup is performed.
   4. User and files information are stored in the database.
   5. The use case ends successfully.
3. Alternative Scenario
   1. Invalid Input – If the input is invalid such as empty input and invalid file/folder path, then the system prevents the user from progressing to the next page but, instead, identifies the fields having invalid input and alerts the users on what kind of error has been committed. The user can then correct the errors with valid input.
4. Post-conditions
   1. The user is successfully registered into the system.
   2. The information about the user and the file/folder paths are stored into the system’s database.
   3. The user is redirected to the Login page.

Use Case #2: User Authentication

This use case describes how a user logs into the system. The Login page either authenticates or denies the user depending on the credentials he/she entered.

1. Preconditions
   1. The system has already been set up and the database of the system exists.
   2. User is not authenticated in the system.
2. Interaction Scenarios

2.1. The use case begins when the user has accessed the Login page.

2.2. The user inputs a valid username and password.

2.3. Use Case: User Authentication is performed.

2.4. The user gains access to the system.

2.5. The use case ends successfully.

1. Alternative Scenarios

3.1. Invalid Username – If the username entered is invalid, then an alert message appears prompting the user to enter valid username and password.

3.2. Invalid Password – If the password entered is invalid, then an alert message appears prompting the user to enter valid username and password.

3.3. Invalid Username and Password – If the username and password entered are invalid, then an alert message appears prompting the user to enter valid username and password.

3.4. Empty Username and/or Password – If either or both username and password fields are empty, then an alert message appears prompting the user to enter credentials.

4. Post-condition

4.1. User is successfully authenticated and logged into the system.

Use Case #3: User Profile and Account Update

This use case describes how a user updates his/her profile and account.

1. Preconditions
   1. The user has been authenticated and logged into the system.
   2. The user’s profile and account already exist in the database.
2. Interaction Scenarios
   1. The use case begins as the user has accessed his/her account.
   2. Upon login, the homepage appears with a menu at the left side. In this case, the user selects ‘Profile’ to edit his/her profile and account.
   3. The user inputs data onto the fields that he/she wants to edit or change.
   4. Use Case: User Profile Update is performed
   5. The user’s profile and account information are updated in the database.
   6. The use case ends successfully.
3. Alternative Scenario
   1. Invalid Input – If the input is invalid such as empty input in this case, then the system prevents the user from updating his/her profile and account but, instead, identifies the fields having invalid input and alerts the users on what kind of error has been committed. The user can then correct the errors with valid input.
4. Post-condition
   1. User Profile is successfully updated.

Use Case #4: Preferences Update

This use case describes how a user updates the system’s preferences, i.e. the paths of the files or folders containing the files needed for the trail.

1. Preconditions
   1. The user has been authenticated and logged into the system.
   2. The files or folders exist in the computer.
   3. The paths of the files or folders needed exist in the database.
2. Interaction Scenarios
   1. The use case begins as the user has accessed his/her account.
   2. Upon login, the homepage appears with a menu at the left side. In this case, the user selects ‘Preferences’ to edit his/her profile and account.
   3. The user inputs data onto the fields that he/she wants to edit or change.
   4. Use Case: Preferences Update is performed
   5. Paths of the files or folders are updated in the database.
   6. The use case ends successfully.
3. Alternative Scenarios
   1. Invalid Input – If the input is invalid such as empty input and non-existent file or folder path in this case, then the system prevents the user from updating the preferences but, instead, identifies the fields having invalid input and alerts the users on what kind of error has been committed. The user can then correct the errors with valid input.
4. Post-condition
   1. Paths of the files or folders intended to be edited are successfully updated.

Use Case #5: Audit Trail

This use case describes how the user performs transaction audit trail using the system.

1. Preconditions
   1. The user has been authenticated and logged into the system.
   2. Necessary files such as financial statements, ledgers, journals, transaction files and log files exist in the computer and their paths exist in the system’s database.
2. Interaction Scenarios
   1. The use case begins as the user has accessed his/her account.
   2. Upon login, the homepage appears with a menu at the left side. In this case, the user selects ‘Start a Trail’ to begin performing audit trail
   3. Upon clicking the menu item, a dialog pops up asking the user to choose a file and select what kind of financial statement that he/she will be inspecting.
   4. The user inputs data and the system loads the financial statement chosen by the user. The user traverses from a transaction document to another via hyperlinks attached by the system until he/she arrives at the output page – summary of the trail.
   5. Use Case: Audit Trail is performed
   6. The use case ends successfully.
3. Alternative Scenarios
   1. Invalid Input – If the user fails to choose a file and/or select the kind of financial statement that he/she will be inspecting, then the system prevents the user from loading the file but, instead, alerts the user of the error committed. The user can correct the error by selecting the kind of financial statement to be inspected.
   2. File Not Found – If the files related to the financial statement being inspected do not exist, the system displays ‘File Not Found’ error and ends the trail. The user is then given the link to the summary of the trail.
4. Post-condition
   1. Audit Trail has been successfully performed.
   2. The system generates summary report of the trail performed.

Use Case #6: Audit Trail – Transaction Validation

This use case describes how the user sends confirmation message to the person transacted with by the company in question. This is one of the last parts of the entire process of the audit trail.

1. Preconditions
   1. The user has been authenticated and logged into the system.
   2. The person in question has a contact number stored in the system’s database,
2. Interaction Scenarios
   1. The use case begins as the user has arrived at the Transaction File page.
   2. Upon clicking the OR (Official Receipt) Number, a dialog pops up displaying the details of the transaction in question with the button ‘Send SMS’ at the bottom of the dialog.
   3. Use Case: Transaction Validation is performed, sending a validation message to the person in question.
   4. The use case ends successfully.
3. Alternative Scenarios

There is no alternative scenario for this use case as the button is disabled if the person in question has no contact number stored in the system’s database.

1. Post-condition
   1. Transaction Validation has been successfully performed.

**Design**

*Data Design*

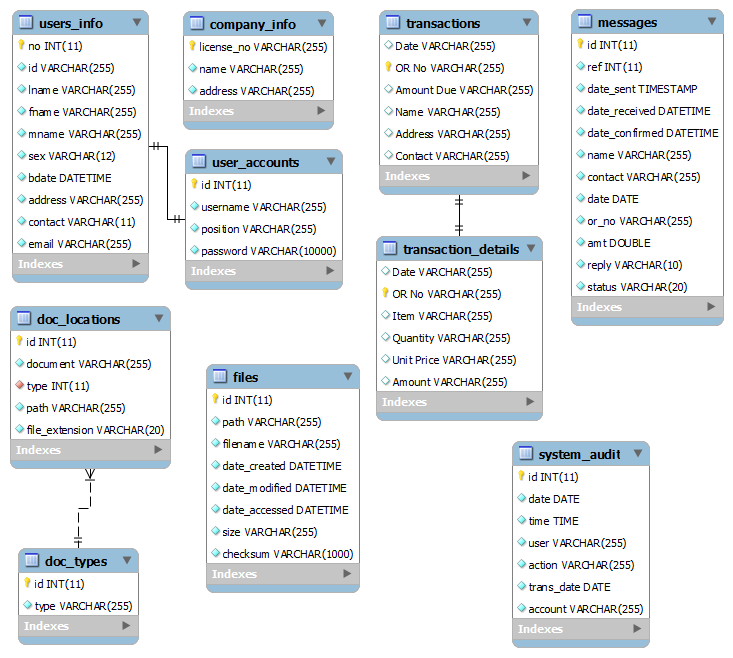
Data Objects

The following are the data objects (tables) that will be managed and manipulated by the system.

* Users\_Info – is an entity that contains all information about the user
  + no – an attribute that is unique to all users
  + id – ID of a user, be it the license number of the accountant or his/her company ID (for internal auditors)
  + lname – the last name of a user
  + fname – the first name of a user
  + mname – the middle name of a user
  + sex – the sex of a user
  + bdate – the birthdate of a user
  + address – the address of a user
  + contact – the mobile or phone number of a user
  + email – the email address of a user
* User\_Accounts – is an entity that contains user’s credentials to be able to log into the system
  + id – an attribute that is unique to all users
  + username – the username of a user in the system
  + position – the position of a user in his/her company
  + password – the password of a user to gain access to the system
* Company\_Info – is an entity that contains all information about the user’s company
  + license\_no – the license number of the company of a user
  + name – the name of the company of a user
  + address- the address of the company of a user
* Doc\_Locations – is an entity that contains all information of the accounting documents needed for the audit trail, especially their locations in the host computer
  + id – an attribute that is unique to all accounting documents
  + document – the name of an accounting document (e.g. cash receipts journal, general ledger, etc.)
  + type – the type/category of an accounting document (e.g. journals, ledgers, etc.)
  + path – the absolute path of an accounting document in the computer
  + file\_extension – the extension of the files associated to the document
* Doc\_Types – is an entity that contains the types or categories of accounting documents needed for the audit trail
  + id – an attribute that is unique to all accounting documents
  + type – the type/category of an accounting document (e.g. journals, ledgers, etc.)
* Files – is an entity that contains all information of all accounting files automatically retrieved by the system through the document locations identified by the user
  + id – an attribute that is unique to all accounting files
  + path – the absolute path of a file
  + filename – the filename of a file
  + date\_created – the date and time that a file was created
  + date\_modified – the date and time that a file was last modified
  + date\_accessed – the date and time that a file was last accessed
  + size – the size of the file
  + checksum – the fingerprint of the data contained in a file represented in a big hexadecimal number that is mainly used by the system for data verification (e.g. has a file been modified?)
* Transactions – is an entity that contains all information of all transactions made by the company whose transactions are to be audited by the user
  + date – the date a transaction was made
  + or\_no – the official receipt number of a transaction
  + amount\_due – the total amount involved in a transaction between a company (client of the system user) and a person/entity (e.g. customer, supplier, etc.)
  + name – the name of a person/entity whom/that a company has transacted with
  + address – the address of a person/entity whom/that a company has transacted with
  + contact – the contact of a person/entity whom/that a company has transacted with
* Transaction\_Details – is an entity that contains all the details of the transactions
  + date – the date a transaction was made
  + or\_no – the official receipt number of a transaction
  + item – the name of an item involved in a transaction
  + quantity – the quantity of an item involved in a transaction
  + unit\_price – the unit price of an item involved in a transaction
  + amount – the total amount of an item involved in a transaction derived from multiplying its unit price to its quantity
* Transactions\_Sample – is an entity similar to Transactions but is only used for the audit sampling
  + date – the date a transaction was made
  + or\_no – the official receipt number of a transaction
  + amount\_due – the total amount involved in a transaction between a company (client of the system user) and a person/entity (e.g. customer, supplier, etc.)
  + name – the name of a person/entity whom/that a company has transacted with
  + address – the address of a person/entity whom/that a company has transacted with
  + contact – the contact of a person/entity whom/that a company has transacted with
* Messages – is an entity that contains all information of all transaction validation messages sent and received by the user
  + id – an attribute that is unique to all validation messages
  + ref – the reference number of a message that is automatically generated by the system
  + date\_sent – the date that a validation message is sent to a person/entity whom/that a company has transacted with
  + date\_received – the date that the system has received a reply from a person/entity to a validation message sent to him/her/it
  + date\_confirmed – the date the user has confirmed a reply of person/entity to a validation message sent to him/her/it
  + name – the name of a person/quantity whom/that a company has transacted with
  + contact – the contact of a person/quantity whom/that a company has transacted with
  + date – the date of a transaction
  + or\_no – the official receipt number of a transaction
  + amt – the total amount involved in a transaction
  + reply – the response of a person/entity contacted by the system
  + status – the status of the message (e.g. sent, new, ignored, and confirmed)
* System\_Audit – is an entity that contains all information about all general ledger accounts
  + id – an attribute unique to all accounts (general ledger chart of accounts)
  + date – the date an action is taken on an account
  + time – the time an action is taken on an account
  + user – the user who takes an action on an account
  + trans\_date – the date a transaction made under an account
  + account – the name of an account involved in a transaction (e.g. sales, cash disbursements, etc.)

Entity-Relationship Diagram (ERD)

The relationships among the abovementioned activities are shown in the Entity-Relationship Diagram (ERD) in Fig. \_\_ below.



*Fig.*

*Component Design*

The following are the components that build up the system:

* Homepage

This page functions as the main navigational page of the system after the user has logged in.

* Setup Module

This module is responsible for setting up the system when it is used for the first time. This is comprised of two sub-components: user profile and preferences (accounting document locations).

Input:

* + - User information
    - User credentials (username, password, etc.)
    - Accounting document locations/paths

Output:

This module has no output; it will only store the data entered and redirect the user to the Login page afterwards.

* Audit Trail Module

This module is responsible for performing an audit trail which is the core module of the system.

Input:

* + - A financial statement and other accounting files (ledgers, journals, transaction files, etc.) in CSV format (Refer to Appendix \_\_)

Output:

* + - Summary of the Audit Trail

This output consists of the diagram that shows the trail of a transaction from the financial statement down to the source document and the results of the sub-processes performed during the entire audit trail process discussed in the ‘Audit Trail Process’ section of this chapter.

* Transactions Sampling Module

This is a sub-module of the Audit Trail module since this is performed during the audit trail process according to the will of the user, that is, a button for the sampling is provided and it is up to the user if he/she opts to have the system perform the sampling. The sampling method used by the system is discussed in the ‘Sampling Method’ section of this chapter.

Input:

* + - Sample size

Output:

* + - A sample of the entire population of transactions displayed in random order
* Transaction Validation Module

This is another sub-module of the audit trail module. This module is responsible for contacting the person/entity involved in a transaction to further validate its existence. However, just like the sampling, this is also optional. The process of transaction validation is discussed in the last part of the ‘Audit Trail Process’ of this chapter.

Input:

* + - Contact number (mobile) of the person/entity whom/that the company has transacted with. If there is no contact number available, this module is disabled.

Output:

This module has no output; it will only send a validation message to the contact number specified.

* System Integrity Module

The quality of the output of the audit trail is largely dependent to the quality of the accounting files. Therefore, this module is responsible for monitoring and protecting the integrity of the necessary accounting files when there are changes in their content as the audit starts. These changes can be due to accident, intentional or due to virus intrusion.

After the user enters the paths of the documents during the setup phase or when editing the preferences, the system automatically searches for files in those paths and copy them into a temporary folder. These cloned files will be the basis in monitoring for file changes and will be used when the user opts to restore the files or will be updated when the user opts to adopt the changes.

Input:

* + - Changes in any of the accounting files

Output:

* + - Alert dialog prompting the user about the changes and giving the user options such as restore the file(s) to its/their last saved state or adopt the changes.

*Major Functions*

The following are the major functions of the system categorized by module:

* Setup Module
  + Save Profile

This function saves the newly entered information about the user and his/her account credentials.

* + Save Preferences

This function saves all paths of the documents needed for the audit that are entered by the user.

* + Edit Profile

This function allows the user to edit his/her profile and account any time after the setup.

* + Edit Preferences

This function allows the user to edit the paths of the documents he/she entered during the setup phase. This also allows him/her to add more paths when needed.

* Audit Trail Module
  + Load File

This function allows the user to load the financial statement that he/she is going to audit

* + View Transaction Details

This function, accessible through a link attached to each Official Receipt (OR) Number, allows the user to view the details of a transaction at the end of the audit trail.

* + View Audit Trail Results

This function allows the user to view the summary of the audit trail performed.

* + Save Audit Trail Results

This function allows the user to save the results. Only saving into an image file is supported by the system.

* + Print Audit Trail Results

This function allows the user to print the results.

* Sampling Module
  + Select a Sample

This function allows the user to command the system to select a sample of the entire population of transactions, if deemed necessary according to his/her judgment.

* Transaction Validation Module
  + Send SMS to Confirm

This function allows the user to command the system to send a validation message via SMS to the person/entity involved in a transaction if a contact number is available.

* System Integrity Module
  + Update the File(s)

This function allows incorporating the changes in an accounting file into its equivalent cloned file that is stored in a temporary folder within the system.

* + Restore the File(s)

This function allows restoration of a modified file into its last saved state prior to the changes in its content.

* + Update Later

This function allows the user to think first before taking action on a modified file.

*Other Functions*

* Authentication

This function is a security measure of the system that prohibits unauthorized access. It prompts the user to log into the system in order to gain access. There are two instances that the user is prompted to log in: (1) when the user begins to use the system and (2) when the user accesses the system after it has become idle for \_\_\_ minutes, that is, there is no interaction between the user and the system.

* Validation Messages Management

This function basically allows the user to view the information of transaction validation messages sent that are stored in an archive. It also allows the user to delete the message one-by-one or to clear the entire archive.

* System Audit

This function allows the user to view the actions (edit, delete, etc.) done towards general ledger accounts (e.g. sales, taxes, etc.) and other information such as who initiated the action and when the action was initiated, among others.

*Graphical User Interface Design*

The graphical user interface (GUI) design was implemented first before all other components of the system. In SDLC, specifically in agile development model, this is called prototyping. The user interfaces must be easy to navigate and use to increase the efficiency in using the system. Refer to Appendix \_\_\_ for the flow of the GUIs and Appendix \_\_\_ for the screenshots of the GUIs.

**Development**

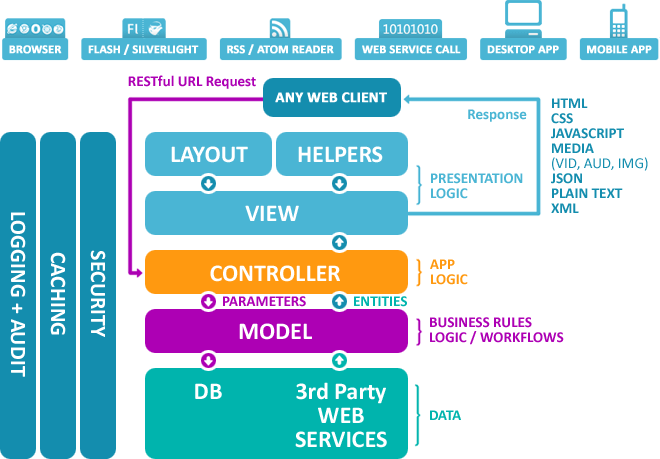
*Programming Tools*

Since the system is web-based, the programming language used is PHP 5 for the backend while HTML 5 is used for the GUI. PHP is a widely-used scripting language for web development that can be easily embedded into HTML [24]. As for the programming environment, Notepad++ v.5.9.2 was used since the researcher-developer is already comfortable and well-acquainted with it in developing web applications.

The framework used is CodeIgniter 2 that, aside from being compatible with PHP, uses the MVC (Model-View-Controller) architecture. This open-source architecture, developed by Trygve Reenskaug in the late 1970s, separates the application logic from presentation [25]. It allows HTML pages to contain minimal number of code lines since the PHP scripts for managing the data are separated from the HTML scripts. Simply, the model handles data management, that is, it represents the data structures that aid in creating, retrieving, updating, and deleting information in the database; the view handles the GUI, that is, how the data will be presented to the user; and, the controller acts as the liaison between the model and the view (see Figure 3) [26].

Meanwhile, this study used the open-source MySQL as the Relational Database Management System (RDBMS) tool to handle the database that stores the transactions, user information and all other data objects of the system [27].

On the other hand, Agile software development is used in the development of the system because of the one-year timeframe of this study. It is a model of SDLC that uses iterative and incremental software development methodologies [28]; thus, it is suitable for rapid development. For example, in order not to start from scratch in developing the GUI, Twitter Bootstrap 2 is used. Twitter Bootrstap is a front-end toolkit which is a collection of HTML, CSS (Cascading Style Sheets) and JS (Javascript) conventions [29].



*Fig. 3. MVC Diagram*

Lastly, the system was developed in Windows 7 operating system (OS).

*Audit Trail Process*

The system only employs vouching, that is, tracing a transaction from a financial statement down to source document(s). During the vouching, hyperlinks are automatically attached to the references displayed in each accounting document to aid the user in tracing transactions (See Fig. \_\_\_). For example, when the system loads the financial statement file, it automatically attaches a link to each account name that appears in the financial statement. The link will lead the user to the corresponding general ledger account. When such general ledger account is loaded, the system attaches a link to the reference of each ledger entry that will lead the user to the journal that corresponds to the ledger entry he/she has chosen to trace. Same process is applied to the remaining accounting document. As for the transaction file, the link is attached to the OR No. of each transaction (See Fig.\_\_) that will lead the user to the details of the transaction chosen in a form of a modal dialog. It is in this modal dialog that the ‘Send SMS to Confirm’ button that supports transaction validation resides (See Fig. \_\_)

When the user clicks the ‘Send SMS to Confirm’ button, the system fetches the contact number available and sends a validation message to that number. To support this feature, Frontline SMS is used for SMS handling. Frontline SMS is an open-source software that ‘turns an offline laptop into a communication hub’. It was developed by Ken Banks in 2005 especially to ‘enable effective communications channels for communities in the developing world’, specifically those communities that have lesser access to Internet [30].

The system uses the following Frontline SMS URI (Uniform Resource Identifier) to send a message via its HTTP Trigger function [31]:

**http://localhost:<portNumber>/send/sms/<destinationPhoneNumber>/<messageContent>/**

Meanwhile, Frontline SMS supports two-way messaging; thus, is suitable for the system because the system does not only sends a validation message but also receives a confirmation reply with the following case-insensitive syntax:

**TRAIL<space><Ref.#><space><Yes or No>**

wherein, ‘TRAIL’ is a keyword set by the researcher-developer to be recognized by Frontline SMS and the Ref. # (reference number) is automatically generated by the system and is included in the validation message. However, when the confirmation reply sent is not in the correct syntax, the system automatically sends back an error message suggesting the correct syntax.

As soon as the confirmation reply (in correct syntax) has been received by the system, an alert dialog pops up prompting the user that the person/entity involved has replied. The dialog gives the user two options: (1) ignore the message but the alert dialog keeps on popping up everytime the user loads the page or (2) confirm the reply; thereby, recording the ‘yes’ or ‘no’ reply of the person/entity involved in the transaction in question (See Fig. \_\_\_\_).

At the end of each trail, the system automatically generates the summary of the trail composed of a diagram that shows the trail itself and the results of the sub-processes performed during the trail such as footing and matching (See Fig. \_\_\_). This summary can be viewed by clicking the ‘View’ link beside each transaction in the transaction file displayed by the system (See Fig.).

*Sampling Method*

Since the scope of the system is limited to auditing transactions of retail stores having an average number of transactions of 20 transactions per day and whose transactions are mostly covered by receipts, the system uses ‘Pareto’s Principle of Distribution’ as the sampling method.

Meanwhile, at the Transaction File page, a button is provided that gives the user an option to select a sample of the entire population of the transactions displayed before viewing the proceeding to the final step of the audit trail (See Fig \_\_). When the button is clicked, a dialog pops up asking the user how the sample size is determined (See Fig. \_\_). If the user chooses ‘Use the standard’, the system applies the Pareto method. First, the system arranges the transactions from highest to lowest in terms of their amount. From there, the system automatically chooses the top 20% of the arranged transactions and add them into the *sample[]* array. The remaining 80% is copied to the *transactions\_sample* table of the database and will be subjected to random sampling. In this case, the Pareto method uses the following formula in determining the sample size [32]:

where,

n – sample size

N – population which is now just the remaining 80% of the entire population

d – desired reliability factor = 0.05

p – 0.50

z – 1.95

After the sample size is determined, the sample is selected randomly using the RAND() method of MySQL and the sample transactions are then added to *sample[]* array. For the random sampling in MySQL, the following SQL command is used [33]:

**SELECT \* FROM ... WHERE ... ORDER BY RAND() LIMIT sample\_size;**

However, if the user chooses ‘Set manually’, the system asks him/her to identify his desired sample size based on his/her judgment. The Pareto method is still applied except in determining the sample size.

**Deployment**

*Installation Requirements*

* Operating System

Since web-based applications are portable, it can be used in Windows or Linux OS.

* Hardware
  + 1.86 GHz processor, at the minimum
  + 1 GB RAM
  + Mouse
  + Keyboard
  + Printer (optional)
* Software
  + XAMPP for Windows v1.7.7 or higher or XAMPP for Linux 1.8.1 or higher (both already contain PHP and MySQL; hence, there is no need to download and install them, separately)
  + Frontline SMS v1.6.16.3

*Installation Package*

All the files and software needed in order for the system to function properly are compressed into a ZIP package. Installers for Windows and for Linux, in the form of a batch file and bash file, respectively, are included in the packaged. These installers will automate the installation of the system for user convenience by installing all necessary files and software into the user’s computer through single click.

**Testing**

**Bibliography**

[1] “Corruption by Country,” *Transparency International*. [Online]. Available: http://www.transparency.org/country. [Accessed: 30-Sep-2012].

[2] “Why Corruption,” *Cibac Partylist*. [Online]. Available: http://www.cibac.org/index.php?option=com\_content&view=article&id=100:why-corruption&catid=38:our-advocacy&Itemid=56. [Accessed: 12-Oct-2012].

[3] K. D. Johnson, “Corruption in the Philippines: Public Sector Improves But Private Sector Lags,” 2012. [Online]. Available: http://asiafoundation.org/in-asia/2012/09/26/corruption-in-the-philippines-public-sector-improves-but-private-sector-lags/. [Accessed: 30-Sep-2012].

[4] K. Swain, “The Effects of Technological Development on Accounting Systems,” *eHow.com*. [Online]. Available: http://www.ehow.com/list\_7708020\_effects-technological-development-accounting-systems.html. [Accessed: 30-Sep-2012].

[5] K. Gupta, *Contemporary Auditing*, 6th ed. New Delhi: Tata McGraw-Hill Publishing Company Limited, 2005, p. 1530.

[6] J. A. Hall, *Accounting Information Systems SEVENTH EDITION*, 7th ed. Mason: Cengage Learning, 2011, p. 830.

[7] “Integrated Financial Management System: An important but limited anti-corruption tool,” in *e-Transparency Conference*.

[8] G. Carague, “The Philippine’s New Government Accounting System,” *Public Sector Technology & Management Magazine*, Manila, 2006.

[9] Kenya Anti-Corruption Commission, “Corruption Prevention Guidelines on ICT Systems in the Public Sector,” no. March, 2008.

[10] J. Ori, “Importance of an External Audit,” *Small Business Chron*. [Online]. Available: http://smallbusiness.chron.com/importance-external-audit-17630.html. [Accessed: 02-Apr-2013].

[11] B. (eHow) Brown, “What Is an External Audit?,” *eHow.com*. [Online]. Available: http://www.ehow.com/about\_4894466\_what-external-audit.html. [Accessed: 02-Apr-2013].

[12] Asian Development Bank, “Accounting and Auditing in the Philippines,” in *Diagnostic Study of Accounting and Auditing Practices in the Philippines*, Mandaluyong: Asian Development Bank, 2003, pp. 29–43.

[13] Auditing and Assurance Standards Council, “Philippine Standards on Auditing (PSAs).” [Online]. Available: http://www.aasc.org.ph/downloads/PSA/PSA.php.

[14] E. Cohen, “The Need for and Issues Surrounding the Seamless Audit Trail.” p. 15, 2006.

[15] Conjecture Corporation, “What is an Audit Trail?,” *wiseGeek.com*. [Online]. Available: http://www.wisegeek.com/what-is-an-audit-trail.htm. [Accessed: 13-Oct-2012].

[16] Computer Assisted Audit Group, “A Guide to Computer Assisted Audit Techniques.” p. 9.

[17] Texas Comptroller of Public Accountants, “Sampling Manual,” *Muscle & Nerve*, vol. 46, no. 5, p. fmiii–fmiv, Nov. 2012.

[18] J. H. Yamamura, “Introduction to Nonstatistical Sampling for Auditors.” 2010.

[19] American Institute of Certified Public Accountants, “AU Section 350,” no. 1, pp. 511–524, 2006.

[20] N. Hitzig, “Statistical Sampling Revisited,” *New York State Society of CPAs*, vol. 2004, no. 5, 2004.

[21] Putra, “Non-statistical Audit Sampling Techniques [SAS 39],” *Accounting-Financial-Tax.com*, 2012. [Online]. Available: http://accounting-financial-tax.com/2010/03/non-statistical-audit-sampling-techniques-sas-39/. [Accessed: 13-Oct-2012].

[22] “Audit Sampling.” .

[23] Department of Revenue | State of Louisiana, “Non-statistical Sampling Guidelines.” 2009.

[24] The PHP Group, “PHP: Hypertext Processor.” [Online]. Available: http://www.php.net. [Accessed: 14-Oct-2012].

[25] M. Fowler, “Model View Controller.” [Online]. Available: http://martinfowler.com/eaaCatalog/modelViewController.html. [Accessed: 14-Oct-2012].

[26] EllisLab Inc., “Model-View-Controller,” *CodeIgniter.com*. [Online]. Available: http://codeigniter.com/user\_guide/overview/mvc.html. [Accessed: 14-Oct-2012].

[27] C. Janssen, “What is a Relational Database Management System (RDBMS)?,” *Janalta Interactive Inc.* [Online]. Available: http://www.techopedia.com/definition/1235/relational-database-management-system-rdbms. [Accessed: 14-Oct-2012].

[28] VersionOne Inc, “What Is Agile Software Development.” [Online]. Available: http://www.versionone.com/Agile101/Agile-Development-Overview/.

[29] Twitter Developers, “Bootstrap from Twitter,” *Twitter Blog*. [Online]. Available: https://dev.twitter.com/blog/bootstrap-twitter. [Accessed: 14-Oct-2012].

[30] “Frontline SMS,” *kiwanja.net*, 2003. [Online]. Available: http://www.kiwanja.net/frontlinesms.htm. [Accessed: 06-Apr-2013].

[31] “Triggering an Outgoing SMS in FrontlineSMS: Using the HTTP Trigger,” *Int’l Extension and Public Alert Systems Program*, 2010. [Online]. Available: http://iepas.net/iepas-blog/2012/7/29/lujuplxx8kj4fv0omknp4re8yezcrc. [Accessed: 06-Apr-2013].

[32] R. V. Salvanera and F. B. Cabalay, “Sampling History - The Philippines Experience,” *Asian Journal of Government Audit*, 1996.

[33] Oracle Corporation, “MySQL 5.0 Reference Manual :: 12.6.2 Mathematical Functions,” *MySQL.com*. [Online]. Available: https://dev.mysql.com/doc/refman/5.0/en/mathematical-functions.html#function\_rand. [Accessed: 14-Oct-2012].