TOOLS FOR SOFTWARE PROJECT DATA COLLECTION AND INTEGRATION

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCE

OF

NEAR EAST UNIVERSITY

by

SALAR FAISAL NOORI ALBRIFCANI

In Partial Fulfillment of the Requirements for

the Degree of Master of Science

in

Software Engineering

NICOSIA, 2017

TOOLS FOR SOFTWARE PROJECT DATA COLLECTION AND INTEGRATION

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCE

OF

NEAR EAST UNIVERSITY

by

SALAR FAISAL NOORI ALBRIFCANI

In Partial Fulfillment of the Requirements for the Degree of Master of Science

in

Software Engineering

NICOSIA, 2017

I hereby declared that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Salar Faisal Noori Albrifcani Signature: Date:

ACKNOWLEDGEMENTS

This thesis would not have been possible without the help, support and encouragement of my supervisor that help me a lot is the Assist Prof. Dr. Yöney Kırsal Ever, for her constant encouragement and guidance. She has walked me through all the stage of the writing of my thesis. Without her consistent and illuminating instruction, this thesis could not have reached its present from.

I would like to thank Assist Prof. Dr. Boran Şekeroğlu who has been very helpful through the duration of my thesis.

My deepest thank to my father Shex Faisal Noori who open the road for me to study the master degree.

Above all, my unlimited thanks and heartfelt love would be to my gorgeous and beautiful wife Rojan, for her loyalty and her great confidence in me. She always encourage and support me through this thesis it can't be describe how she help me with this thesis and that's mean a lot to me, thank you my lovely wife.

ABSTRACT

In this thesis there are some data tools that I explain such as (CVS, SVN, GIT, Issue tracking system, Bugzilla, Hudson, Wikipedia and Twitter) that are created in this thesis. Originally this thesis is about the data collection and integration and to know how to collect and integrate the data that you have, especially when we store the data in to the database, for example we can store data in to the (Excel or Access).

This master's thesis aims to integrate the relational database management system and analyzing the relationships between each of them. In this thesis, unified data model for Ed-Fi were designed. Reverse engineering is used for designing the data model and unification of the data models. After, creating the unified data model, I design the data model for CVS and job history.

In order to know more about unified data model, we also explain the Unified Data Model for Relational and NoSQL Databases to supply a way for memory and recovery of information from the database system.

Keywords: CVS; SVN; GIT; Issue tracking system; bugzilla; hudson; wikipedia; twitter

ÖZET

Bu tezde, (CVS, SVN, GİT, Sayı izleme sistemi, Bugzilla, Hudson, Wikipedia ve Twitter) gibi oluşturulan bazı veri araçları detaylı bir şekilde açıklanacak. Başlangıçta bu tez veri toplama ve entegrasyon hakkında, sahip olduğumuz verilerin nasıl toplanacağını ve nasıl işleneceğini göstermek üzere hazırlanmıştır. Özellikle veriyi, veritabanında sakladığımızda, örneğin Excel veya Access gibi yazılımlarda nasıl depolanacağını göstermemizi sağlayacaktır.

Bu yüksek lisans tezi, ilişkisel veritabanı yönetim sistemini bütünleştirmeyi ve her biri arasındaki ilişkileri analiz etmeyi amaçlıyor. Bu tezde, Ed-Fi için birleşik veri modeli tasarlanmıştır. Tersine mühendislik yöntemleri, veri modelinin tasarımı ve veri modellerinin birleştirilmesi için kullanılmıştır. Birleştirilmiş veri modeli oluşturulduktan sonra, CVS ve iş geçmişi için veri modeli tasarlanmıştır.

Birleştirilmiş veri modeli hakkında daha fazla bilgi edinmek için, İlişkisel Birleştirilmiş Veri Modeli ve NoSQL Veritabanları açıklanmıştır. Bu modeller, veri tabanından bilgi kurtarmak ve veri hafizasına ulaşmak için bir yol sağlamaktadır.

Anahtar Kelimeler: CVS; SVN; GİT; sayı izleme sistemi; bugzilla; hudson; wikipedia; twitter

TABLE OF CONTENTS

ACKNOWLEDGMENTS	i
ABSTRACT	ii
ÖZET	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vii
LIST OF TABLES	viii
LIST OF ABBREVIATIONS	ix

CHAPTER 1: INTRODUCTION

1.1 Thesis Problem	1
1.2 Aim of the Thesis	3
1.3 Overview of the Thesis	3

CHAPTER 2: LITERATURE REVIEW

2.1	Data Collection	5
	2.1.1 The Purpose of Data Collecting	5
	2.1.2 Method of Data Collection	5
	2.1.3 Information Quality levels	7
	2.1.4 Data Characteristics and Properties	8
	2.1.5 The Rationale of Data Collection	9
2.2	The Integrating of Data	10
	2.2.1 Process of Integration Data	10
	2.2.2 Advantage of Integrating Data Tools	11
	2.2.3 Next Generation of Integrating Data	11
2.3	Defining Data Model	12
	2.3.1 Types of Data Model	12
	2.3.2 Data Model Contains	12
	2.3.3 Important of Data Model	13
	2.3.4 Data Model Usage	13

2.3.5 Data Model Process	13
2.4 The Model of Entity Relationship	
2.4.1 Data Model for Entity Relationship Diagram	
2.4.2 The Data Model Semantic	
2.5 What Is Unified Data Model (UDM)?	
2.5.1 Unified Data Model Process	
2.5.2 Comparison between Unified Data Mode & I	Data Model Semantic 21
2.6 The Ed-Fi Unifying Data Model	
2.6.1 Unified Data Model for Relational and NOS	QL Databases 23
2.7 Explaining the Data Flow Diagram	
2.7.1 Data Flow Diagram Types	
2.7.2 Data Flow Diagram Notations	
2.7.3 The Example of Data Flow Diagram	
2.8 Existing Model	
2.8.1 CVS	
2.8.1.1 The Use of CVS	
2.8.2 SVN	
2.8.2.1 Using the Subversion	
2.8.3 GIT	
2.8.3.1 GIT Distributed	
2.8.3.2 The Backup Multiple of GIT	
2.8.3.3 The Workflow of GIT	
2.8.3.4 The Workflow Subversion of GIT	
2.8.3.5 The Integrating Manager of GIT Workflow	35
2.8.4 The Issue of Tracking the System	
2.8.5 Bugzilla	
2.8.5.1 Using the Bugzilla	
2.8.5.2 The Bugzilla Life Cycle	
2.8.6 Defining of the Hudson	
2.8.6.1 Using the Hudson	
2.8.7 Definition of the Wikipedia	
2.8.8 Definition of the Twitter	

2.8.9 Definition of the Social Media	39
2.8.9.1 Social Media Analytics	39
2.9 Comparison	40
2.9.1 What Is the Difference between CVS and SVN?	40
2.9.2 What Is the Difference between GIT and SVN?	41
CHAPTER 3: DATA MODEL DESIGN FOR CVS AND JOB HISTORY	
3.1 Design Data Model for CVS and Job History	44
3.2 Apply the Data Model for CVS and Job History by Tortoise CVS Program	46
CHAPTER 4: ANALYSIS MODEL AND RESULTS	
4.1 Analysis Model and Results	54
CHAPTER 5: CONCLUSIONS AND FUTURE WORK	
5.1 Conclusion and Future Work	55
REFERENCES	56

LIST OF FIGURES

Figure 1: Information quality level concept	8
Figure 2: Agency data collection rationale	9
Figure 3: Integrating data from different departments or sectors	11
Figure 4: Data model process	14
Figure 5: One-to-one relationship	17
Figure 6: One-to-many relationship	17
Figure 7: Many-to-many relationship	17
Figure 8: The entity relationship diagram	18
Figure 9: Ed-Fi operational data store	22
Figure 10: One-to-many relationship design	25
Figure 11: External entity	26
Figure 12: Process of data	26
Figure 13: Flow of data	27
Figure 14: Store of data	27
Figure 15: DFD food ordering system	28
Figure 16: DFD level 0	29
Figure 17: DFD level 1	30
Figure 18: Workflow subversion of GIT	34
Figure 19: Relationship between integration manager and developer of GIT	35
Figure 20: The Bugzilla life cycle of	37
Figure 21: Primary and foreign key	43
Figure 22: Design of data model for CVS and job history	45
Figure 23: Data model for main process	46
Figure 24: Data model for policy process	47
Figure 25: Data model for edit process	48
Figure 26: Data model for tools process	49
Figure 27: Data model for advanced process	50
Figure 28: Data model for appearance process	51
Figure 29: Data model for cache process	52
Figure 30: Data model for ignored files process	53

LIST OF TABLES

Table 1: Comparison between unified data model and data model semantics	21
Table 2: The difference between RDBMS and NoSQL databases	24
Table 3: Differences between CVS and SVN	40

LIST OF ABBREVIATIONS

GPS:	Global Positioning System
IQL:	Information Quality Level
DBMS:	Database Management System
SSN:	Social Security Number
EDM:	Entity Data Model
ERM:	Entity Relational Model
ERD:	Entity Relational Diagram
UDM:	Unified Data Model
Ed-Fi:	European Development Finance Institutions
JSON:	JavaScript Object Notation
ODS:	Operational Data Store
API:	Application Programming Interface
XML:	Extensible Markup Language
UML:	Unified Modeling Language
NEDM:	National Education Data Model
SQL:	Structure Query Language
RDBMS:	Relational Database Management System
ACID:	Atomicity Consistency Isolation Durability
DFD:	Data Flow Diagram
CVS:	Concurrent Version System
SVN:	Subversion
SCM:	Software Configuration Management
ITS:	Issue Tracking System
PK:	Primary Key
FK:	Foreign Key
CK:	Candidate Key, Composite Key
AK:	Alternate Key

CHAPTER 1 INTRODUCTION

The tools that I used in this thesis are (CVS, SVN, GIT, Bugzilla, Issue tracking system, Hudson, Wikipedia and Twitter). Each of those tools mentioned has different system and different purpose in software engineering uses. Also issue tracking system known as bug trackers are basically repositories which can keep track of our issues such as refactoring issues, new features or production issues. These systems help our software development speed significantly and most importantly they provide internal communication for everyone who contributes to the project. One of the most famous issue tracker system especially is Bugzilla which is an open source issue tracker used in many projects. we used unified data models for NoSQL database to know more about database system and the relationship between each of the database and tables. Also here we design unified data model for CVS and job history that include seven table of entity and each of table consist several of attribute. The examples of data flow diagram of food ordering system is mention on chapter 2.

1.1 Thesis Problem

The thesis problem is not much recognized about the interest of using some of device or social media like Wikipedia or Twitter, this kind of the features social media must be software development integration (Storey, Deursen and Cheng, 2010).

To support and solve thesis problem of software engineering we can choose from tools that include:

1. Collaborative Development Environment

The main goal of Collaborative Development Environment (CDE) is to reduce friction in collaborative processes. Let's consider an example of CDE (Jazz). Jazz supports the types of integrating project. The information about issue tracking system allows accessing the collaborative development environment (Storey, Deursen and Cheng, 2010).

2. Integrated Development Environments

Integrated Development Environments (IDE) of software projects for example to solve the problem of any design, which I use in chapter 3 the design for CVS and job history and I did the solution to be sure that my design is working and have no problem of integrate development environments. These tools (CVS and SVN) are using different system and different purpose of software engineering is consists of demand the development of the management. Each of integrated development environments has data integration control technique to support the activities of project through concurrent version system (Storey, Deursen and Cheng, 2010).

The issue tracking system are the management of any project of collaboration tools, issue tracking system is defined in chapter 2.

3. Software Project and Plagiarism

Software project and plagiarism is an easy project that anyone can use for example any open source software that specifically designed for project. While others are for exchanging the information and searching from internet of the project and that will be the plagiarism. But source code project dependent on project. The applications of the software project consists the types or tools of integrating interfaces like issue tracking system and social media for interring to the website that save the project in the collaborative development environment for supporting the development (Storey, Deursen and Cheng, 2010).

The Social Media Using

The social media using anyone can use the social media whose connected to the internet, however the one has no connected to the internet cannot use social media for example (Wikipedia and Twitter). Most of the conventional kinds of the development software going on the projects and the integrating of the society are the range of process development tools from Integrated Development Environments (IDE) to Collaborative Development Environment (CDE) and Software Project Plagiarism.

The way for using the social media is to support and power the software engineering development from requirements to test the documentation (Storey, Deursen and Cheng, 2010).

Social media such as Wikipedia is open source software, if you need to access the Wikipedia you don't need any account, but in twitter you can't access you need an account to access the twitter. Wikipedia design with expansion of information about the software and the collaboration from 2005, the software engineering can create any project or system to understand easily of the issue tracking system (Storey, Deursen and Cheng, 2010).

1.2 Aim of the Thesis

This thesis aim is to collect data from multiple systems and integrate the relational database management system (RDBMS) and analyzing the relationships between them.

To also elaborate unified data model for Relational and NoSQL Databases to supply a way for memory and recovery of information from the database system.

It elaborates the data flow diagram for food ordering system and has 2 levels and that is level 0 and level 1.Moreover to design data model for CVS and job history with the use of tables which shows the relationship between each table and we also used the primary and foreign key.

Last but not least the aim is to develop adapters for extracting data from some of the systems those that include (CVS, SVN, GIT, Issue Tracking System, Bugzilla, Hudson, Wikipedia and Twitter).

1.3 Overview of the Thesis

The current thesis is classified into five chapters as follows: Chapter 1 Talking about the introduction, problem of thesis, aim of thesis and summary of thesis. Chapter 2 is a literature reviews that includes the information about data collection, data integration, data flow diagram, design, unified data model, some tools of the system. Chapter 3 is the data model design for CVS and job history, the keys of database and database management system (DBMS). Chapter 4 includes data analysis and result. Chapter 5 conclusion and ideas for the future work can be found. Summary as I have identified tools for software project of collecting data and integrating data.

The information produced in software project is often distributed upon several systems, that data I use in this thesis are (CVS, SVN, GIT, Issue Tracking System, Bug, Wikipedia, Twitter and Hudson). Those systems have different purpose and different ways of using data models as I explain each of them and the comparison between those systems. In order to analyze software project that need to collect and integrate data from multiple systems and collecting and integration of data is shown on Chapter 2.

In order to get data from issue tracking systems, I have created unified data models so that I can get the data and analyses it effectively. In addition, also I design an Ed-Fi unified data model for representing data about software development project extracted for the purpose of analysis, the foundation of data model of generally exchanged and shared data education. I was successful in combining the data for literature review and issue tracking systems. I took data from well-known social media such as Twitter and also well-known issue tracking systems and Bugzilla.

In this thesis I design data model for CVS and job history it's explain the entity and attribute and show the RDBMS, in the design i have seven table each table consist some of attribute and we know that database use tables to regular the information, to protect the data integrity should be correct and in a well shape, I use notion of keys, such as (foreign, primary, alternate, candidate and composite keys).

CHAPTER 2 LITERATURE REVIEW

2.1 The Collecting of Data

The collecting of data defines the process in collecting data, for example student data and employee data to be collected and the information from their collecting data and stores the data that we collect into the database. The reasoning of collecting data is for using the process that connected to computer system, the information that we input to the computer we can save it in any database (John, 2004).

2.1.1 The Purpose of Data Collecting

There are four purpose of data collection which includes the following:

- Data collection is better for obtain the information to the database.
- To keep data on record and avoid losing data.
- To make decisions about important problem that we have in our data and solving that problem.
- To pass the information of data to other and saving the data (Kadam, Rizwan and Parab, 2013).

2.1.2 Method of Data Collecting

The method of data collecting is the various methods technologies of system that used for infrastructure data collection, the method of data collecting used for the collection large data for example management data (Flintsch and Bryant, 2006).

Type of data collection method:

1. Manual Data Collecting

Manual collecting data is the method that collects more than one data and knowing the distance of measuring device. The manual collected data is a simplest data and you can store the data to database.

The collecting of data is moving from one site to another to record and checking data, manual collection data show us detailed of data collection but require some of time (Flintsch and Bryant, 2006).

2. Automated Data Collecting

Automated data collecting method is the method that collects data automatically which involves the use of a multipurpose of the computer hardware and Global Positioning System (GPS) to know and show our data that we store in any system to be capture. The advantage of the automated collected data is the GPS are used to capture or knowing the location data. Specifically developed software is generally used the data to be easy to see their data and also to show the location of the data. The collecting of data automation has high accuracy (Flintsch and Bryant, 2006).

3. Semi-Automated Data Collecting

Semi-automated data collecting is like and similar of automated data collection that involves the complete of semi-automated data collection method but with less of degree of automation. It's important of semi-automated data collection within transportation data that properly implement data collection (Flintsch and Bryant, 2006).

4. Remote Data Collecting

Remote data collecting is the last type of data collection method that is using for application remote, this kind of data collecting obtain the information across type of images and satellites technologies method. Remote data collection is collect data from satellite and stores that data (Flintsch and Bryant, 2006).

2.1.3 Information Quality Levels

Information Quality Levels, the abbreviation is (IQL) is the level that consists of five types of information quality levels; these levels use different kinds of information and process such as high level data and low level data as shown in Figure 1, the process of data collecting methods for the degree corresponding to support decisions types. The figure 1 is to explain the five levels of information quality levels (IQL) concept. The information quality levels demand several levels of feature detail of data collecting to support identical decision of process. The information quality levels have five different types of data collecting methods that use different performance of data collection, also in information quality levels have two special type like high level data and low level data, this two types is using for system performance monitoring, planning and performance evaluation, program analysis or detailed planning, project level or detailed program and last one is project detailed or research (Flintsch and Bryant, 2006).



Figure 1: Information Quality Level Concept (Flintsch and Bryant, 2006)

2.1.4 Data Characteristics and Properties

Data characteristics and properties recommended the elements types of data characteristics that are define some different type of properties that use according to the source WERD 2003 (Smith and Lytton, 1992).

Type of data characteristics and properties:

- Data collecting for specification.
- Data collecting that use frequency.
- The reliability of high quality on data collecting.
- The perfection of data collecting.

2.1.5 The Rationale of Data Collecting

The rationale of data collecting is define the rationale information of data collecting that provide the historical practice or staff experience, data collecting standard and system process. The most contractors of rationale data collecting have been standards and input the data into the management system to define the process of the system that the contractor is use for rationale data collecting. Figure 2 below is shown the summarized of the rationale data collecting (Flintsch and Bryant, 2006).



Figure 2: Agency data collection rationale (Flintsch and Bryant, 2006)

2.2 The Integrating of Data

The integrating of data is define a different system and fountain of visualizing not like the collecting of data, the integrating of data combine the act of conversion to effect the data information, then the integrated provides data format like a data warehouse. The integration of data process save in several system on different fragmented and format (Wiley, 2014).

2.2.1 Process of Integrating Data

Process of integrating data is the data from multiple sources and has a single view over all sources, and answering queries using the combined information.

Process of the integrating data includes two types:

- Physical of integrating data: Copings the integrating data to warehouse.
- Virtual of integrating data: Keep the integrating data only at the sources.

The integrating of data process is also valid within a single organization. Figure 3 is shown the integrating data from different departments or sectors (Eltabakh, 2012).



Figure 3: Integrating data from different departments or sectors

2.2.2 Advantage of Integrating Data Tools

Advantage of integrating data tools is the modern of integrating data that let us to combine data from several of different sources to create any data on obvious point of view and make the decision of integrating data tools. This kind of advantage tools is to progress the integrating of data process to provide a high user moderator and visual that enables a view of data activity process, for example the advantage of integrating data tools like data mapping this kind of data show for each to understand exactly where to produce each part of data and how the data pass into the system and modify the exact way where the data is going (Wiley, 2014).

2.2.3 Next Generation of Integrating Data

Next generation of integrating data will be powerful and even save more data of next generation integrating data. Integrating data define a set of manner that contain many of data like data management, the quality of data, the union data, data capture, data modify and more.

The integrating of data produce like collaboration of data integrating that related to a data warehouse and database management, the produce of integrating data have obtain precocity and will show many ideas for the next generation in integrating data (Russom, 2011).

2.3 Defining Data Model

The model of data is defining a gathering of conceptual tools like term of data, data relationship and semantic of data. The design of data model is to know the relationship between each of them. The model of data is the performance of data structures that require for database system is much power for the term of communication and easy to show how data will regular know in database system (Ravvi, 2013).

2.3.1 Types of Data Model

Data model have several type but we will explain two of the important type of data model to be accomplished during many kinds of projects and in multiple levels of projects. The data models should ideally be stored in repository so that they can be retrieved, expanded, and edited over time (Oludele, 2012).

(Jeffrey, 2004) specified two kinds of data model that includes:

- Data model pending system analysis: At analyzing system there are logical data models are made like piece of expanding new databases.
- Strategic model of data: This is a level of the making a strategy for information systems, which represents an architecture and a complete vision for information systems. Engineering Information is a way which can represent this method.

2.3.2 Data Model Contains

The model of data takes from the planning and analysis stage as its inputs. Components of data model are almost similar of type of data model of designing. Data model gathers information about storing data to database and the requirements. In the other hand there are two outputs.

The first one is the diagram of relationship that indicates the structures of data and defining the relationship among them. The second one is the document that explains the object of data to save any kind of document in database. The regulations asked by the dictionary of database to give the details asked by the developer of data to build the physical database (Mamcenko and Gediminas, 2004).

2.3.3 Importance of Data Model

The aim and main target of data is to be sure that the all objects of data needed by database and store the data successively and completely represented, due to the easily uses of the data model got records and local language. The data model is used by the developer of the database in order to use as a scheme for constructing the physical database. The information consist of data model that will be used to indicates the tables of keys and relational like foreign and primary key (Mamcenko and Gediminas, 2004).

2.3.4 Data Model Usage

Data model methodologies and techniques are used data in a criterion and harmonious for controlling a resource. The data model usage criteria are hardly bespoke for every schemas and analyzing data within a company (Oludele, 2012).

Using of data model includes the following:

- Use data model to make data looks like a resource.
- For integrating data of information systems.
- Use it also for designing the databases.

2.3.5 Data Model Process

Data model process is a database design of processing to explain briefly about the model of database, the rational of process data model include everything of the main logical and the design of physical. Also the data model process showing the keys of database such as (primary key, foreign key, composite key, alternative key and candidate key), and showing the relationship between each table that I have in database design.

System analysis use many techniques to describe the information of data model like the flowing of data diagram that are use character to appear the system convert and to input the data in beneficial information, and also the flowing of data diagram consist of some of level that level that I use in this thesis is (Level 0 and Level 1) in chapter 2 that each of level use different system. An entity model contains some of attribute in every existence. The idea of designing the database can be used to explain on several ways to design a database system (Brewington, 2012).

Data model process has the conceptual data model and physical data model and each of those two types contains some of tools, to explain more about data model process Figure 4 showing that how data model process create or update the data.



Figure 4: Data model process (Oludele, 2012)

The Figure above explains the way of the data model process that shows how to create the physical and logical of the data model. Each of them has different ways to create the data.

Concept of Attributes

The Attributes can be defined as a concept of representing entity and also can use in data model that contain the entity of the characteristics and attributes. Let's consider we have two table one of them named (Company) and the other one called (Employee) each of this table contains some of attribute and the information of company and employee, such as (SSN, Address, First name, Birthdate, CV and more). The current elements of the data must completely be presenting to show each attribute and characteristic to display of how the data must be regular (Oludele, 2012).

The attribute have three different types such as:

1. The attribute knows the benefit of the association.

2. The attribute are able to describe the actual real phrase.

3. Also can be the certain relevant of contexts environment.

The Attribute Definition: the attribute we use much of time in a database management system and also describe the way of the database, for example using the attribute in tables of database if we have two tables each of them have different name and different entity like (Color, Location, Address, ID, Name, Salary and more) (Oludele, 2012).

The Characteristic Definition: the characteristic is very specific way of attribute in general element of data which identify the recognizing of characteristic of group from one part to another.

Entity Concept

Entity Concept is the entity type of table design of abstraction that represents some of table in database design. For example entity can be (Person, Manager, Employee, Plant and etc.), and also each of entity table contains several of attribute, the type of entity point out the relationship to describe the attribute in every tables (Oludele, 2012).

There are three kinds of entity as showing below:

1. The Entity of Physical: is the entity that is very simple way and easy to understand.

2. The Entity of Conceptual: is the entity that is less to understand easily, also they are known to define the type of entity.

3. The Entity of Event: this kind of entity is a typical way of event to define the table of entity and also in associative entity (Oludele, 2012).

There are three ways of entity:

1. The Primary of Entity: This kind of entity is the entity that describes actual things like person, place and more.

2. The Associative of Entity: This is using for something that is produce for two ways of entity like the reception that complete the exist package.

3. The Attribute of Entity: The attribute of entity is using for the data that depend of the way of entity and also to describe the attributes, for example to define the specific results of the attribute and entity in each table that are using in database (Oludele, 2012).

2.4 The Model of Entity Relationship

The Model of Entity Relationship (ERM) defines as the relationship to explain entity, attribute and multiplicity of the relationship. The model of entity relationship is a relational database system model method that is using in a data model of software engineering department. I will explain some of example of entity relationship model (ERM) to understand of the entity, attribute and the multiplicity of the relationship. There are three ways to draw of entity relationship model; I will explain each of these ways.

This example is One-to-One relationships; this example below explains the results;

(1:1) Manger

Figure 5: One-to-one relationship

This design is One-to-One relationship, we have one company and one manager of the company and that will be One-to-One relationship.

This example is One-to-Many relationships; the example below explains the results;



Figure 6: One-to-many relationship

This design above is One-to-Many relationship, also we have one company but in the company there are many employees that works in that company, and that will be One-to-Many relationship.

The last one is Many-to-Many relationship; the example below explains the results;



Figure 7: Many-to-many relationship

This design above is Many-to-Many relationship; I have many students for example that want to register in Near East University that takes many of courses, and that will be Many-to-Many relationship.

2.4.1 Data Model for Entity Relationship Diagram

The entity relationship of data model is describing the design of the database system. Figure 8 is the entity relationship of data model or diagram (ERD) that shows the relational between entities and attribute to save the data about products and the product consist of three attribute as shown in figure 8 also the attribute product ID of the product entity will be the primary key of the product entity, then the supplier consist two attribute, then knowing about their relationship between of those two entities, about the saving data must be collected in both of entity supply and products. Figure 8 showing a relationship between products and suppliers (Oludele, 2012).



Figure 8: The entity relationship diagram

2.4.2 The Data Model Semantic

The data model semantic is a method to define the concept on the design of data model with all the relationship between them. The data model semantic is a way of building the data to represent the exact logical way, the data model semantic is also know an abstracting while defining how to save the character that related in the database (Oludele, 2012).

The data model semantic may have several of requirements that requirements is...

- Allow to define the data model schema.
- The data model semantic is an effective way by supporting the develop type of semantic information.
- Data model semantic make on the metadata.
- Standing the data on the manage application to provide the interchange between metadata.
- Easy way of file integrating to store metadata to the web.
- Semantic data model it's simple to understand the database for example, such as structure query languages (SQL).

2.5 What Is Unified Data Model?

The unified of the data model define process for designing model of data to be sure our data model is correct and complete; also this kind of data is initial new generation process to variable work practices, data model is an exercise for automated dimensions of optimizing data model and to show the solution technology of the process and the data model work (Ghosh, 2009).

2.5.1 Unified Data Model Process

Unified data model process that concentrate the requirement of data before doing the data modeling, each system project evolution that contains step of data must do the following steps:

- You must know your data that you are using it to define the concerned data with validate and requirement.
- Then you must regular you're saving data into the database system and using the traditional data model.

On the other hand the EDM that stand for unified data model, the system is to combining the two steps that I explain in unified data model process.

Ensure that your unified data model is complete of database design and store that data in database, all data elements have an entity into the data structure and also all identifying of data attribute of the entity concept required by the process (Ghosh, 2009).

2.5.2 Comparison between Unified Data Model & Data Model Semantic?

The comparison is to know each different way of unified data model and data model semantic about validation, fundamental, implementation and more. To know each different way I have compare them as shown below.

Unified Data Model	Data Model Semantic
1. Unified data model should focus on	1. Semantic data model a technique used to
identifying and validating data requirements	define the meaning of data within the
first, then defining the inter relationship	context of inter relationships with other
with other data.	data.
2. The use of the unified data model	2. Sub-unit defines fundamental concepts of
approach can provide benefits to an	database, this concept are described in
organization for the discovery.	Semantic Binary Model (SBM) of data.
3. Documentation and implementation of	3. The logical data structure of a database
process requirements as well as data	management system (DBMS) either
requirements.	hierarchical, network or relational.

Table 1: Comparison between unified data model and data model semantics

2.6 The Ed-Fi of Unifying Data Model

Ed-Fi: Stand for European Development Finance Institutions.

Ed-Fi stands for European Development Finance Institutions. The Ed-Fi of unified data model is to exchange data to express the unified modeling language (UML) and includes entities to be easily recognized the data by that one that use the unified data model. The Ed-Fi of unified data model use many tools such as (XML, API, ODS and etc.).

The role of the unified data model information is transaction of data like JSON that stand for JavaScript Object Notation, the Ed-Fi technology means that the data standard components discussed in this document share common models and data definitions with all other Ed-Fi technology components (Christopher, 2015).

The XML which is the extensible markup language describe the data exchange to support framing the sharing of any data, for example (student) data for designing the database source, the exchange data of the framing student that contains transcript, degree and more. The European Development Finance Institutions of data level can do the work of each software and hardware platform. This figure below represents the Ed-Fi operational data store and the dashboards of unified data model (Christopher, 2015).



Figure 9: Ed-Fi operational data store

JSON: as shown in figure above is using the transactional data of JSON to connect with operational data and then saving the operational data and at the end applying the unified data model. JSON represent data structures to support binary values, and JSON use in the European Development Finance Institutions figure above. This kind is stand for Java Script Object Notation (Vogel, 2011).

ODS: ODS is stand for Operational Data Store. Is a kind of a database with the integrating of data also used in the European Development Finance Institutions figure above that relate to the data structures with operational data store for exact functions process (Rogers, 2010).

API: this kind is use for transactional of data with Operational Data Store that connect together for applying the unified data model and also API is always looking for every part. API stands for Application Programming Interface (Patterson, 2015).

XML: XML is a language to define the record of data structure that contains a specific way of structure like picture, mark and more. The different part for the database table which have a different meaning and content, also XML use for the European Development Finance Institutions figure that XML do the bulk of data then applying unified data model. XML stand for Extensible Markup Language (Walsh, 1998).

UML: This type is to design and analyzing the software for international manufacture, and UML is combining the Java Script Object Notation (JSON), Operational Data Store (ODS), Application Programming Interface (API) and Extensible Markup Language (XML). UML stands for Unified Modeling Language (Williams, 2004).

2.6.1 Unified Data Model for Relational and NOSQL Databases

The relational database of NoSQL is to supply a technique for store and storage for each relational data model that we are using for database design, data model for relational and NoSQL database is to recover each model of data relational that means the classifies a relational for using (Strozzi, 1998).

What is SQL?

SQL is a design for computer language database for describing and manage the data in RDBMS; SQL is originally developed by IBM (International Business Machine) by integrated the computer language of data and that was used for query. SQL stands for Structured Query Language (Halvorsen, 2016).

Define the RDBMS

RDBMS is a program that we can do anything like updating and creating in the relational database system, the most trades in the relational database management system that are using SQL which means for Structured Query Language to inter the database system. Structured Query Language is not important to use when we create or update the database managing system (Rouse, 2005).

On the other hand the database for NoSQL is a project of architecture data that manage the achievement of application to support the special features for document of software engineering. The RDBMS use tables of database each data have entity and entity consists some or several of attribute to know the relationship between each table in a condition part of table, NoSQL also use for unified data model to support for existing database. RDBMS is stand for the Relational Database Management System (Wang, 2016).

RDBMS	NoSQL
1. Use entity relationship.	1. Have four major types: document,
2. Schema predefined.	object-oriented, key value and figures.
3. Scaling up.	2. Schema on read.
4. Strong consistency.	3. Scaling out.
	4. Eventual consistency.

Table 2: The difference between RDBMS and NoSQL databases (Wang, 2016)

The methodology of unified data model is using the database management system for NoSQL database to provide solution across the RDBMS and NoSQL databases. The notation for concept model is to describe the data and knowing the relationship between each data (Wang, 2016).

For example let's consider we have to entity of concept and the relationship between those two entities, for example (Company and Employee).

In the methodology of unified data model for NoSQL database after when defining the entity and the relationship between those two entities, I will put the tags between the (Company and Employee). This diagram below explain that I have two entity properties and each of them use tags of relationship that have two entity and may have several of attribute. The relationship between them is One-to-Many.



Figure 10: One-to-many relationship design

2.7 Explaining the Data Flow Diagram

Data flow diagram is the process in efficacy of diagram of showing the clarify way of how we can through the data flow into the system in one process to other process. DFD are the design to understand easy of the diagram and contain some of level. I will explain the two level of data flow diagram (Level 0 and Level 1) with their diagram. The bad system of data flow diagram will show a bad process and activity of subsystem. DFD stands for Data Flow Diagram (GenerisTeam, 2006).

2.7.1 Data Flow Diagram Types

There are two general type of DFD, such as:

1. The Data Flow Physical Diagram: the data flow physical diagram is the application of showing the exact system devices of the data.

2. The Data Flow Logical Diagram: the data flow logical diagram is the process of describing the system and explaining how the system implemented to represents in effect way for process and function (GenerisTeam, 2006).

2.7.2 Data Flow Diagram Notations

The data flow diagram notations have several types, such as:

1. The Entity External: the entity external is the exporter purpose of the system that does the outputs and inputs as shown below (GenerisTeam, 2006).



Figure 11: External entity

2. The Process: the process of data represents the transference of modify the system of data of the input and output (GenerisTeam, 2006).



Figure 12: Process of data

3. The Flow of Data: the flow of data is not a direct line that through the packets of information of data flow (GenerisTeam, 2006).



Figure 13: Flow of data

4. The Store of Data: the store of data shows either do a short time or constant place where the data come to store (GenerisTeam, 2006).

Figure 14: Store of data

2.7.3 The Example of Data Flow Diagram

To show an example of data flow diagram I discover the food ordering system as an example of data flow diagram (DFD) and that consists of an apparent representing of the information in system of data, according to the example of data flow diagram the food ordering system you can tell the information that provide a delivery form a person to another person who is the part in the process system. The need of the information must be complete the process when is complete it should be save to access the system easily. In figure 15 is the example of data flow diagram for food ordering system (Ndiaye, 2015).

The Example of Data Flow Diagram Food Ordering System

To show an example of data flow diagram for food ordering system, the diagram contains several way or process to complete the ordering and do the easy way to the system that containers in this figure are the manager, customer, kitchen and supplier, each part of them is the entity and do the exchange information between the system and entity. The figure below shows the example of data flow diagram for food ordering system (Ndiaye, 2015).



Figure 15: DFD food ordering system example

There are two important level of data flow diagram (DFD), which I will explain and draw those two levels...

1. Level 0 DFD diagram

The process of the system in all data should be appear easy for saving in this level, the diagram of flow data process is drawn from the split process in level 0 of data flow diagram and supposed to be the same level, every process in the data flow diagram in level 0 should be number like 1.0, 2.0, and 3.0. Figure 16 explain DFD level 0 (Dasgupta, 2005).



Figure 16: DFD level 0

2. DFD level 1 diagram

In this level 1 of data flow diagram explain the different ways in drawing and it's much similar from level 0 of data flow diagram. Level 1 must be the exact process from the level 0. Figure 17 show the details of data flow diagram level 1 and knowing the difference between them (Dasgupta, 2005).



Figure 17: DFD level 1

Data flow diagram has many examples, the example I explain is the data flow diagram for food ordering system and the two levels (Level 0 and Level 1).

2.8 Existing model

2.8.1 CVS

CVS an abbreviation that stand for Concurrent Version System, CVS is the process that allow us to get a source of coding that a designer can save the code on different system or process on any program; even it allows a designer can share the different system control of relation in a mutual on the data warehouse of several files. This type of process may also know as a version control system. Concurrent version system was creating in the UNIX system of operating on the available environment for a free using software basis on the versions commercial, however the concurrent version system a public tools for that any designer can work on UNIX systems (Rouse, 2011). The concurrent version system cannot work by keeping the multiple of track and copy the files of source code, but can do the preserve on one copy and to save of all the changing of the concurrent version system. The designer can do the exact of special version on the system; concurrent version system can rebuild that special version from changing of the record (Rouse, 2011).

The concurrent version system that each designer can use to keep their work or track on a different system, also this change of can be used or add to the members that responsible for this work through any command (Rouse, 2011).

The concurrent version system can be use another program, this program is called the (RCS) Revision Control System, this can do the effective management of the revision that is saving the record of any change that go with any file of source code (Rouse, 2011).

2.8.1.1 The Use of CVS

The use of CVS has four ways, such us:

1. The Checking of CVS

The checking out for the concurrent version system is to make a copy of our job or work and suppose to copy it only one time (Tichy, 2000).

2. The Updating of CVS

This kind of CVS has two types:

- The updating of the concurrent version system is to integrate our work to change for other designer that have obliged into the work of copy.
- Beneficial of time that has checkout since passing on the last update of the concurrent version system (Tichy, 2000).

3. The Editing of CVS

• It supposed to write your work clearly to understand and make a copy of your work.

- It's a choice for the designer to report his files while other is given or using his files.
- To edit your work always do the changes in your copy working (Tichy, 2000).

4. The commitment of CVS

This type of the commitment of concurrent version system that any designer or developers can change the same files that you change it during last updating, also the commitment of concurrent version system will ask you to integrate the changes that you did in your updating into the original copy before accessing you to commit your changes (Tichy, 2000).

2.8.2 SVN

SVN is an open source relation of system control and SVN stand for the subversion. The achievements of the subversion guide and files that change the subversion of system, during the time of the subversion that let us to recover the ancient version of the data to check of how our data is changing in the history. This kind of subversion is important for concurrent version system that any of developer can change their data easily by the subversion (Collins, Fitzpatrick and Pilato, 2002).

The subversion can work through the network which authorizing to everyone by using the different computer; in some scale the power of managing the same set that everyone can manage the their competent of their location in collaboration enhance. The advance of the subversion can happen so fast without a single channel through which all adjustment in the channel, because the work is completed that anyone need the best quality of the subversion and not to losing any of data (Collins, Fitzpatrick and Pilato, 2002).

Sometimes the concurrent version system is also known as the SCM Software Configuration Management. The software configuration management is includes on two thought, the first thought is the virtually that every product can extent any of user to show the multiple of the concurrent version system (Handfield, 2011).

The second thought is for exist the supplies of the software configuration management that SCM based on the system that is identifying to manage the source code and have much of features that are the exact way to the software development. The example of software configuration management such as the program languages or saving the software tools for subversion. On the other hand the subversion is a general system that can be used to collecting and managing any of files (Collins, Fitzpatrick and Pilato, 2002).

2.8.2.1 Using the Subversion

The subversion use several ways that includes the flowing:

- The using of subversion is very strong with interface command.
- The subversion can use in a free source.
- The using of subversion is available for software client.
- The using of subversion is much easy than concurrent version system.
- Using the subversion system is very fast.
- Using the attributes of the subversion to save file.
- The subversion takes every file of types without directive.
- It also supports any folder or file (Fouad, 2009).

2.8.3 GIT

This kind of software or tools that I define is a process of the concurrent version system and also is much similar for the concurrent version system and use a source code, this can change many code on very fast time and saving any files that the code are changed (Lee and Edwards, 2013).

2.8.3.1 GIT Distributed

This feature is the most distributed of any software configuration management. The GIT distributed always doing the checkout of the source code of the system (Chacon and Straub, 2014).

2.8.3.2 The Backup Multiple of GIT

The backup of multiple GIT means that even when using the workflow of GIT distributed and many of this backup and the copies should be save to the software configuration management. On the other hand there is only one single copy of the data warehouse (Chacon and Straub, 2014).

2.8.3.3 The Workflow of GIT

The definition of this type of GIT distributed is the system of data warehouse that are related to the concurrent version system (Chacon and Straub, 2014).

2.8.3.4 The Workflow Subversion of GIT

Can be defines as a GIT workflow and will not let us to push any data warehouse where the developer want to push or save the data in a specific common special from that person that are work on the concurrent version system. Figure 18 is explaining the workflow subversion of GIT (Chacon and Straub, 2014).



Figure 18: Workflow Subversion of GIT

2.8.3.5 The Integrating Manager of GIT Workflow

The integrating manager of GIT workflow is to collect and integrate the data based on the developer from the integrating. Figure 19 shows the relationship between integration manager and developer of GIT (Chacon and Straub, 2014).



Figure 19: The relationship between integration manager and developer of GIT

2.8.4 The Issue of Tracking the System

The system of issue tracking defines as an application of the software that let the project to fix each problem of the forward way in the system of issue tracking that the user of computer system cannot use the program till the problem of the system is fixed. When the problem is fixed it will show the technical report of the error that can use know the problem and not happen again, ITS stand for Issue Tracking System (Rouse, 2003).

2.8.5 Bugzilla

The definition of Bugzilla is related to the issue of tracking the system, and also Bugzilla know as a Bug tracker. The designer can easily save the track effectively of the product, if a problem show in Bug tracker of issue system the designer or developer can fix or solve it with their product that he use easily. We can use a programming language in Bug tracking to design the feature that we have, as I explain the feature of life cycle of Bug that explain the way of new bug from a user with can assure the product without unconfirmed (Barnson, 2006).

(Barnson, 2006) discover several types of Bugzilla which includes:

- Bugzilla is powerful for searching
- The users configurable email notification of Bugzilla changes
- Full change of history
- Dependency tracking and graphing
- Excellent attachments of management
- The integrate product based of granular security schema
- Bugzilla have full security
- Stable of relational database management system
- Using web and xml email of interface
- Completely customizable and localizable web user interface
- The extensive of configurability
- The Bugzilla is easy to upgrade

2.8.5.1 Using the Bugzilla

Using the Bugzilla is to know the each figure or life cycle that we can use of issue tracking system that most software has stay to domain the fundamentally high software of development. This step of using Bugzilla share the email and list of the product to the monitor of considerable Bugzilla by the designer to dropped it, the important of using the Bugzilla is much of organization or company that interest for collecting or integrating the issue tracking system of Bugzilla that contentment with the system. About the problem of Bugzilla is easy to manage the data and to resolve the data to keep it to the track, on the other hand according to the software system of issue tracking by using Bugzilla is to find or discover the trouble by help IT information technologies (Barnson, 2006).

2.8.5.2 The Bugzilla Life Cycle

The Bugzilla life cycle also can be defined as a workflow. Figure 20 explain the Bugzilla life cycle (Branson, 2006)



Figure 20: The Bugzilla life cycle

2.8.6 Defining of the Hudson

This type is defined by using free software to integrate the tools of the software, also the Hudson work with much of software like the subversion, software configuration management and concurrent version system (Krishnan, 2010).

2.8.6.1 Using the Hudson

For using Hudson (Krishnan, 2010) discovers several ways which includes:

- Easy to use web based user interface
- Lots of plugins
- Collected many user interface
- Change set of support
- Easy to setup, configure and administer
- Powerful of supporting various SCM system
- Comprehensive project reporting and dashboard
- Security to support user authentication and restriction
- Free open source

2.8.7 Definition of the Wikipedia

Everyone can use wiki or also know Wikipedia that is connected to the internet and wiki is an open system, each that have the internet can register by using the wiki to take the information about website, social media, finding any program and more.

2.8.8 Definition of the Twitter

This is a social media network that design for everyone to connected to each other and it's free software only you must have the internet to access the twitter and make your account. There are many characters letters of twitter social media; also you can post anything that you want on the twitter social media.

Twitter are so short that you might wonder why people bother posting them in the first place. But that's actually a big part of what makes twitter so popular: It's easy to quickly share what's happening in your world or follow along with a popular topic in real time.

2.8.9 Definition of the Social Media

Defining the social media is provide many features of communication and application that use in every society and have many application like Facebook, Twitter, Wiki and more, as I use the twitter and wiki in this thesis. To access in social media only you must have the internet because it is free open source system software.

2.8.9.1 Social Media Analytics

The analytical of the social media is the exercise of collecting the data from any websites and to uses the social media for collecting and integrating that data. The most of analytical social media is usually to make enough (Rouse, 2015).

2.9 Comparison

2.9.1 What Is the Difference Between CVS and SVN (Ismail, 2011)?

There are several types of differences between concurrent version system (CVS) and subversion (SVN) which discover by (Ismail, 2011). These differences are listed below table.

Subversion (SVN)
1. Maintain the current and preceding
versions of file.
2. Based on commits as true atomic
operations.
3. Each file is an ordinary for user of
information parable output.
4. SVN basis on relational database.
5. SVN works faster than CVS.
6. Allows attaching to a file of any number
of all possible attributes.
7. The storage of manipulates all the file
types of special information.
8. SVN does not allow rollback of any
commit.
9. Transaction support.

Table 3: Differences between CVS and SVN

2.9.2 What Is the Difference Between GIT and SVN?

Now I will explain the main difference between GIT distributed and subversion (SVN) in some different way which includes:

1. The Difference in the Workflow:

Workflow of distributed GIT:

The difference between of GIT distributed and subversion (SVN) in two kinds of ways, such as:

- The GIT distributed stores the full history of all its branches and tags within the GIT.
- The latest stable release is contained within the master.
- Active feature work is developed in separate branches.
- When a feature is finished the feature branch is merged into masters and deleted.

Workflow of subversion (SVN)

- The trunk directory represents the latest stable release of a project.
- Active feature work is developed within subdirectories under branches.
- When a feature is finished the feature directory is merged into trunk and removed.

2. The Difference in the Distribution:

GIT: The distributed revision control system which means every developer checking out code from central repository server will have their own cloned repository installed on the machine.

SVN: Have centralized revision control system or server.

CHAPTER 3 DATA MODEL DESIGN FOR CVS AND JOB HISTORY

In this chapter I design the data model for CVS and job history and also each of keys in relational database management system (RDBMS). In the design I have an entity and attribute each entity must have some of attribute and the relationship between them and also consist several of keys.

The management of database system provides a logical view of process and to understand the relationship between each data that I use and store in database management system (DBMS). As we know that database use tables and keys to know how we can use the database and relationship between each table. There are five keys of relational database management system (RDBMS), such as:

1. The primary key

- 2. The foreign key
- 3. The candidate key
- 4. The composite key
- 5. The alternate key

Define each of these keys:

1. The Primary Key

Primary key (PK) is a relational database management system that use in each table, for example if I have a table named manager and this table consists some of attribute such as (Id, name, address and phone), the id of manager table will be the primary key. Primary key must be unique values and column of the primary key can't be written with null values, each of table of the database can have only one primary key not more.

2. The Foreign Key

The foreign key (FK) is the key which is used to describe the relationship between two tables.

Example, knowing more about the primary key and foreign key, let's consider I have two tables, table one is (customer) and table two is (sales).

The figure 21 below will explain and knowing the primary and foreign key.



Figure 21: Primary and foreign key

In customer table, above I have the customer id is the primary key, and in sales table customer id would be the foreign key, because I have on customer table customer id is the primary key. The primary key and foreign key are the most important and use of keys in the relational database management system (RDBMS).

3. The Candidate Key

The candidate key is the key which is for primary key of a table; each table must have at least one candidate key but at the same time can have many of candidate key.

4. The Composite Key

The composite key uses many attributes to make the primary key then that primary key is known as composite key.

5. The Alternate Key

The alternate key is the key that can be each of candidate keys except for the primary key.

3.1 Design Data Model for CVS and Job History

To be sure that my design of data model for CVS and job history is working, I will use the TortoiseCVS software. I will define the TortoiseCVS software then I will apply that software to my design of data model for CVS and job history.

TortoiseCVS Software is the software that enables the developer to use code by concurrent version system (CVS) server. Developers download the TortoiseCVS to make some change and upload those changes to the CVS server

Figure 22 is explain the design of data model for CVS and job history and it will explain the relationship between each of the table and their keys.

In Figure 22 have seven entities and the relationship between them and each entity consist some of attribute, each entity have a primary and foreign key.

E.g. if people come and want to register for the job, they will take the information of that people e.g. that we have in the entity of people that include some of attribute (person-id, birthdate, address, phone number, and SSN).

Then each of that people will register their CVS as shown is CVS entity and their attribute (cv-id, person-id, cv-create-date and cv-details). We have in people entity a primary key (person-id) and the foreign key of that primary key will be the (person-id) in CVS entity. The relationship between these two entity people and CVS will be one to many (1: N).



Figure 22: Design data model for CVS and job history

3.2 Apply the Data Model for CVS and Job History by Tortoise CVS Program

In the main process as shown in window below that we should choose the application as of main process and choosing the language. The important application of Overlay icons should be the TortoiseCVS.

lain	Policy	Edit	Tools	Advanced	Appearance	Cache	Ignored files	
rogre	ess dialog	:		Close manu	ually			v
Progress messages:			Quiet				×	
)verla	ay icons:			TortoiseCV	S			v
				Changes wo	on't take effect	t until you	log off or reboot.	
older	overlay i	cons:		Change ico	on when folder	contains	changed files.	~
utolo	ad chang	ed folde	er icons:	Autoload	d folder icon ov	verlays		
angu	age:			English				~
					- (s			

Figure 23: Data model for main process

In the policy windows follow that information.

Main	Policy	Edit	Tools	Advanced	Appearance	Cache	Ignored files				
Empty	folders:		Pr	une empty fo	olders						
New fo	olders:		Øα	reate new fol	lders on updat	e					
Force	head revi	ision:		no matching	revision is fou	nd, use t	he most recent one				
Automatic commit:			Don	Do not automatically commit after Add/Remove							
Allow 1	Network [Drives:		ermit sandbo	xes and reposi	tories on	network (remote) drive	es			
Allow F	Removab	le Drives		ermit sandbox	xes on remova	ble drive	S				
Add Ig	nored Fil	es:	S	now .cvsignoi	red files in the	Add Con	tents dialog				
Explor	er Colum	ns:	⊠sł	now columns	only for CVS-c	ontrolled	folders				
Empty	comment	t check:	⊡Is	sue warning	if no comment	provided	in commit dialog				
				4							

Figure 24: Data model for policy process

If there is a problem on our model and we want to edit follow that windows below.

101	toisecvs	- Pre	erences (Jiobalj				,
Main	Policy	Edit	Tools	Advanced	Appearance	Cache	Ignored files	
Edit p	olicy:		Do not us	e Edit/Unedit	2			~
Edit o	ptions dia	og:	Show e	dit options di	alog			
Auton	natic uned	it:	Automa	atically unedit	after Commit			

Figure 25: Data model for edit process

Tools window is to know what our model consist of as shown below

1ain	Policy	Edit	Tools	Advanced	Appearance	Cache	Ignored files	
Diff ap	plication:			2				Browse
rwo-v	ay diff pa	arameter	rs:	"% 1" "%	2"			
Merge	applicatio	on:		?				Browse
Two-v	ay merge	e parame	eters:	"%mine"	"%yours"			
SSH a	oplication	(:ext: o	nly):	C: Progra	am Files\Tortoi	seCVS\To	or <mark>to</mark> isePlink.exe	Browse
SSH p	arameters	s (:ext: o	only):	-l "%u" "	%h"			
SSHO	s server	(:ext: a	nd :ssh:):	CVS				
Bug tr	acker URI							

Figure 26: Data model for tools process

Then the advance window choosing the custom home folder and other application

		- 14		Adupped	<u>.</u>			
Main	Policy	Edit	Tools	Advanced	Appearance	Cache	Ignored files	
lome	folder:			⊠ Alw	ays recalculate	home fo	lder	
Custor	m home fi	older:						Browse
Vetwo	rk compre	ession:		0 - No	ne			v
Exclud	led folder	s:						
Includ	ed folder	5:						
1enu i	icons:			Ena	ble menu icons	6		
Maxim	um numb	er of tag	s for gra	oh: -1				
Sandb	ox DOS/L	INIX:		Autod	etect (default	to DOS)		~

Figure 27: Data model for advanced process

The appearance is when we input our entity and attribute we can change the color by selection the bottom Select color.

Main	Policy	Edit	Tools	Advanced	Appearance	Cache	Ignored files		
Noise text colour:		Sample	text			[Select colour		
Warning text colour:		Sample	Sample text						
Update	ed files co	olour:	Sample	text			[Select colour	
Modified files colour:			Sample	[Select colour				
Conflic	t files col	our:	Sample	text			[Select colour	
Files n	ot in CVS	colour:	Sample	text			[Select colour	
Error t	ext colou	r:	Sample	text			[Select colour	
Tip tex	ct colour:		Sample	text			[Select colour	

Figure 28: Data model for appearance process

The cache is to know about the data model and list

1ain	Policy	Edit	Tools	Advanced	Appearance	Cache	Ignored files	
Server	version:	C	lear cache	•				
Module	e list:	C	lear cache	•				
Branch	/Tag list:	C	lear cache					
				OK			ř	

Figure 29: Data model for cache process

Ignored files the last one of TortoiseCVS

Main	Policy	Edit	Tools	Advanced	Appearance	Cache	Ignored files	
Per-us	er list of	cvsignor	e file nam	e patterns:				
		-						^
								~
lint: T	To recet t	he list ar	d include	all files ente	r a cincle evela	mation	ark ("I")	
mit. I	UTESEL U	ic list di		di nes, ente	a a single exua		ык (:).	

Figure 30: Data model for ignored files process

CHAPTER 4 ANALYSIS MODEL AND RESULTS

4.1 Analysis Model and Results

This chapter I explain analysis model and the results of model analysis. The analysis model is the research of the dynamic characteristic of structures and the domain of measuring and analyzing the dynamic response of structures, when data are collected as shown in details in chapter 2 the process of data collection is better to obtain the information to store that information to the database and to keep data on record. The model analysis of data for using the logical component of data, this data is the one of important step that should be completed. The data technology is the database systems used for managing information that use to access the database between entities and compare the data sets that use the database management system (DBMS).

The profit includes the database management system (DBMS) and database for CVS and job history. One of the profits of (DBMS) because the data is kept on more than one computer using database management software it becomes easy to handle information.

The result of model analysis is to know more information about the database management system (DBMS) and database for CVS and job history application that the data should be independent in details of data storage. The feature of database management system (DBMS) is important for techniques to store data on storage devices.

For improvement I am looking forward to be updating unified data model for design principles and patterns and also using latest updated for CVS and job history information to avoid bogus data.

CHAPTER 5 CONCLUSIONS AND FUTURE WORKS

5.1 Conclusions and Future Works

In this thesis is about tools for software project data collection and integration. Data collection is about how to gather or collect the information of any data that we can use and store it in the database management system (DBMS), for example the data that we can use is (Student data, employee data, manager data, teacher data and more). The integration of data is explaining how to complete those data when we store it in the database management system (DBMS) as I explain it in details in chapter 2 about the data collection and integration. This thesis have many information and examples of data, which I have data model, semantic data model, the Ed-Fi unifying data model, data model design for CVS and job history and more information of data model, this will help people to understand easily of data model and how it's work. I also use social media in this thesis because it's important that everyone if they connected to the internet can get more of the details or information of the data in future work.

For future work this thesis will help people and companies to exchange the information easily and faster since almost everyone is connected to the internet and social media. And also in the future work I will do much more example of database management system (DBMS) and how to use the structure query language (SQL) because maybe there will be a new version or program of database that I can get this thesis beggar of information that may help people a lot to understand everything in database management system (DBMS).

However for some of the people who are not connected to the internet or social media they will not easily understand the database management system (DBMS) or any data that I use in the thesis because it's will be difficult for them to know everything of the data model.

REFERENCES

- Barnson, M. (2006). The Bugzilla Team. *This is the documentation for Bugzilla, the mozilla.org bug-tracking system.* The Bugzilla Guide 2.16.11 Release.
- Chacon, S., & Straub, B. (2014). GIT Distributed, GIT Multiple Backups, GIT Workflow, Subversion-Style Workflow of GIT and GIT Integration Manager Workflow. *Is available to read online for free*. Dead tree versions are available on Amazon.com.
- Christopher, I. (2015). The Ed-Fi of Unifying Data Model, Tittle: Developers' Guide Overview, [Last update: April 21, 2015].
- Dasgupta, S. (2005). Level 0 and Level 1 DFD diagram. Available at: <u>http://www.marcoullis.com</u>/KNOWLEDGE/SYSTEMS/marcoullisp_systems_proc ess_modelling.html
- Eltabakh, M. (2012). Process of data integration. *Data integration is also valid within a single organization*.
- Flintsch, W., & Bryant, W. (2006). Information Quality Levels and Data Collection Methods, Asset Management Data Collection for Supporting Decision Processes, edited 2009. Virginia Department of Transportation, US Highway Transportation Press.
- Flintsch, W., & Bryant, W. (2006). Data Characteristics and Properties and Data Collection Rationale, Asset Management Data Collection for Supporting Decision Processes edited 2009. Virginia Department of Transportation, US Highway Transportation Press.
- Fouad, A. (2009). SVN Usage & Best Practices. Alternative Delivery Channels Architecture & Design Team Leader. Published in: Technology, Business, Saudi Arabia.

- Ghosh, N. (2009). The Unified Data Modeling (UDM), He has earned dual bachelor's degrees in Physics and Computer Science and has a Master of Science in Software Engineering. *Real-World Decision Support (RWDS)*.
- GenerisTeam, S. (2006). Data flow diagram, Data Flow Diagram Principles and Data Flow Diagram Types. Created/Modified by Sui Generis Team. Process # CMPE202-5-Sui2.
- GenerisTeam, S. (2006). Data Flow Diagram Components and Corresponding Notations and Data Flow Diagram Levels. *Created/Modified By Sui Generis Team.* Process # CMPE202-5-Sui2.
- GenerisTeam, S. (2006). Data Flow Diagram Levels and Data Flow Diagram Approach. *Created/Modified By Sui Generis Team*, Process # CMPE202-5-Sui2.
- Halvorsen, H. (2016). Tittle: Structured Query Language, University College of Southeast Norway.
- Ismail, A. (2011). Difference between CVS and SVN, Tittle: Subversion and bug tracking. Senior Android Software Enginner at Technivance, Published in: Business, Technology. Egypt.
- John, D. (2004). Introduction to the data collection. *Dictionary of computing*. Ox Ford University.
- Kadam, A., Rizwan, S., & Parab, P. (2013). The purpose of data collection. Published in: Marketing, Technology and Business.
- Krishnan, T. (2010). Thtroduction to Hudson. Continuous Integration and Hudson.
- Lee, J., & Edwards, A. (2013), Introduction to Git, Columbia University.
- Mamčenko, J., & Gediminas, V. (2004). Contains of a Data Model. Course Title Information Recourses Course volume 3, 0 cr. (4, 50 ECTS cr.). Technical University Lecture Notes on Information Resources Part I.

- Mamčenko, J., & Gediminas, V. (2004). Important Data Model. Course Title Information Recourses Course volume 3, 0 cr. (4, 50 ECTS cr.). Technical University Lecture Notes on Information Resources Part I.
- Ndiaye, M. (2015). Data Flow Diagram with Examples Food Ordering System, University of Washington, Seattle.
- Oludele, A. (2012). Types of Data Model. *Course Title: Semantic Data Modeling,* COURSE CODE: DAM 344. Available at: <u>http://wikieducator.org/images/b/b5/DAM_344_PAPER_1xx.pdf</u>
- Oludele, A. (2012). Use of Data Model and Data Model Process. *Course Title: Semantic Data Modeling*, COURSE CODE: DAM 344. Available at: <u>http://wikieducator.org/images/b/b5/DAM_344_PAPER_1xx.pdf</u>
- Oludele, A. (2012), Entity Relationship Model (ER) and Data Model for Entity Relationship Diagram. *Course Title: Semantic Data Modeling*, COURSE CODE: DAM 344. Available at: <u>http://wikieducator.org/images/b/b5/DAM_344_PAPER_1xx.pdf</u>
- Oludele, A. (2012). Introduction to Semantic Data Model. *Course Title: Semantic Data Modeling*, COURSE CODE: DAM 344. Available at: <u>http://wikieducator.org/images/b/b5/DAM_344_PAPER_1xx.pdf</u>
- Patterson, M. (2015). Introduction to API, Application Programming Interface. Tittle: What Is an API and Matter?
- Raavi, T. (2013). Business Development Executive at, Published in: Education, Technology.
- Rouse, M. (2011). Concurrent Versions System (CVS). Contributor(s): Erin Kelly [Last updated: March 2011].
- Rogers, D. (2010). Introduction to the ODS, Operational Data Store.

Rouse, M. (2003). Introduction to Issue Tracking System.

Russom, P. (2011). Next Generation Data Integration, Second Quarter.

- Storey, M., Deursen, A., & Cheng, L. (2010). Thesis problem. *Information Software Engineering Research Group*. Delft University of Technology.
- Strozzi, C. (1998). Unified Data Model for Relational and NoSQL Databases. Contributor(s): Johan Oskarsson. Available at: <u>https://en.wikipedia.org/wiki/NoSQL.</u>
- Sussman, B., Fitzpatrick, B., & Pilato, C. (2002). Version Control with Subversion.
- Tichy, W. (2000). Typical scenario for Concurrent Versions System (CVS). Formerly maintained by Cyclic Software.
- Vogel, L. (2011). Introduction to the JSON, JavaScript Object Notification.
- Walsh, N. (1998). Introduction to XML, Extensible Markup Language, available at: http://www.edb.utexas.edu/minliu/multimedia/XML.pdf
- Wiley, J. (2014). Data Integration for Dummies. *111 River St. Hoboken, NJ 07030-5774*.Hoboken, New Jersey.
- Zock, K. (2015). Differences between GIT and SVN. Senior Software Engineer at *Ebizproduction*. Between-git-svn-kareem-zock.