Palestine Polytechnic University

College of Information Technology and Computer Engineering

Department Of Information Technology

Data Sharing Management System

Graduation project submitted to fulfill the requirements of the B.Sc. in Information Technology

By:

Aya Muhtaseb Aya Rjoub Sanaa Sarahneh

Supervisor:

Ezdehar Jawabreh

Jan.2013

الإهداء

إلى نور القلوب وقرة العيون وشفاء الصدور

سيدنا رسول الله (صلى الله عليه وسلم)

إلى من رضاهم من رضا ربي

أمي ، أبي

إلى كل اسم يبدأ بالألف، وينتهي بالياء
أمي، أبي، أختي، أخي ...

إلى الشموع التي احترقت لتضئ لي دربي... إلى العيون التي سهرت معي الليالي

إلى القلوب التي غمرتني بالمحبة... إلى الشفاه التي تمنت لي النجاح

إلى كل الاحبه والأصدقاء

إليهم جميعا نهدي هذا الجهد المتواضع

فريق البحث

الشكر والتقدير

لا يسعنا بعد إتمام هذه الدراسة إلا شكر الله تعالى الذي أعاننا على إتمام هذا العمل .

ونتوجه بالشكر الى الشمعتان اللتان أضاءتا لنا الدرب ، والدانا الأعزاء

وانه لمن دواعي السرور و عظيم امتنان فريق البحث أن نتقدم بجزيل الشكر و التقدير وفائق الاحترام لكل من ساهم و ساعدنا في إتمام هذا العمل و نخص بالذكر المدرسة الفاضلة ازدهار جوابرة مشرفة المشروع.

والأستاذين الفاضلين أ. محمد أبو طه ، و أ. طارق العجلوني اللذان كانا عوناً لنا في مشروعنا هذا.

و الطواقم الإدارية والأكاديمية والفنية في دائرة تكنولوجيا المعلومات.

والسادة موظفي دائرة القبول والتسجيل في جامعة بوليتكنيك فلسطين الذين تعاونوا معنا في انجاز هذا العمل مع حفظ الألقاب والأسماء وعلى رأسهم الأستاذ وائل عواد مدير دائرة القبول والتسجيل ، وموظف التسجيل السيد زياد شاور.

فريق البحث

Abstract

Nowadays, most organizations expanded their work in the form of extranet to facilitate exchanging data among each other. To achieve this, organizations implements the inter-organizational system strategy (IOS), to achieve this strategy, this requires Electronic Data Interchange (EDI). This current study will focus on how to implement data sharing and EDI between organizations. Therefore, the main objective of this study will be to implement data sharing and practice the electronic data interchange between organizations. To pursue the study questions and objectives, the current study will use the qualitative research methodology in conjunction with the System life cycle. The expected result of this project will be Electronic Data Interchange, Generating and mapping XML documents, and Converting data into presentational formats (i.e. Reports, statistics) using XML standards, based on web application.

ملخص

في الوقت الحاضر، قامت معظم المنظمات بتوسيع عملها في نطاق الشبكة الخارجية لتسهيل تبادل البيانات بين بعضها البعض. ولتحقيق ذلك، يجب على المنظمات تنفيذ إستراتيجية نظام مشترك بين المنظمات(IOS) ، يتطلب تحقيق هذه الإستراتيجية إتاحة تبادل البيانات الإلكتروني (EDI). والدراسة الحالية تركز على كيفية تنفيذ تشارك البيانات والتبادل الالكتروني للبيانات ما بين المنظمات. ولذلك فإن الهدف الرئيسي من هذه الدراسة هو تنفيذ تشارك البيانات و التبادل الالكتروني للبيانات ما بين المنظمات. ولتحقيق أسئلة الدراسة وأهدافها، تقوم الدراسة الحالية على استخدام منهجية البحث النوعي جنبا إلى جنب مع نظام دورة الحياة. والنتيجة المتوقعة من هذا المشروع هي إيجاد تبادل إلكتروني للبيانات، وإنشاء وخلق وثائقXML ، وتحويل البيانات إلى أشكال قابلة للعرض(مثل: تقارير، إحصائيات) باستخدام معايير XML , وذلك من خلال تطبيق ويب.

Table of Contents

List of Tables

List of Figures

Chapter One

Introduction

Contents:

1.1 Introduction

Nowadays, most organizations expanded their work in the form of extranet to facilitate exchanging data among each other. This study will focus on critical part of the inter-organizational system strategy (IOS), which achieved using Electronic Data Interchange (EDI), therefore, some questions arise in this area: what is interoperability and how can be achieved, what is electronic data interchange and what are the models that allow electronic data interchange, how to implement data sharing between organizations, and how to allow an integrated interchange between these organizations. This project aims to practice the EDI in/between organizations, in order to achieve the IOS strategy. To do this the project team will study different models and choose the most suitable one.

1.2 Study Questions:

To achieve the study objectives, this study will express the following questions:

- What is interoperability and how can be achieved?
- What is EDI and what are the models that enable EDI?
- How to implement data sharing between organizations?
- How to allow an integrated interchange between these organizations?

1.3 Project Objective:

This project aims to practice the Electronic data interchange between organizations, in order to achieve the inter-organizational system strategy. To achieve this objective, this requires to achieve the following sub objectives:

- Implement data sharing between organizations.
- To practice the EDI between organizations.
- To harmonize the various authentications in organizations in a way that facilitates the EDI.
- To enable an integrated interchange between these organizations.

1.4 Project Importance:

The importance of the project can be formulated from this crucial area of information technology which is interoperability and data interchange, therefore, the importance of this project can be subdivided in three dimensions:

- Organization:

– Each organization shares its collected data with other organizations but without breaking the original data usage commitment to its clients.

– Data integration: which combine data from autonomous and heterogeneous sources, and provide a unified view of data.

– Provide secure sharing through authorized views.

- The project team: this project enables the project team to practice the theoretical part of our study and to go further insight in this new area, which is important for the future of cloud computing.
- The university: this project opens wide prospects of conducting researches in this important area.

1.5 Methodology:

In this project, the project team will follow the qualitative research methodology in conjunction with system life cycle (SLC), qualitative methodology through making surveys for previous work in this area, and using understanding and interpretation of this work. The project team will continue with SLC in developing and designing Data sharing management system.

1.6 System Scope:

This system must be implemented on organizations that needs to share and interchange their data between each other, such as educational organizations, therefore, the project team have selected a case that includes two databases from two educational organizations with a centralized organization, which is the Palestinian Ministry of Education and Higher Education in Ramallah, and the two databases are

Palestine Polytechnic University database in Hebron, and Birzeit University database in Ramallah.

1.7 Feasibility Study:

This section represent the feasibility study for development and operational recourses and their costs.

- **Development Requirement:**

This system requires software, hardware and human resources, as follows:

The following table describes the costs of the human resources required, that represented in a team of developers and programmers:

Table 1.1 Human Recourses Costs

Role	Number	Hours Number per Day	Price per Hour
System analyst	3	1.5	20$
Programmers	3	1.5	20$
Total per Day	180$		
Total per month	5400$		

The project team needs a set of software summarized in the following table:

Table1.2 Software Resources Costs
Source: http://www.amazon.com/: amazon

Software resource	Quantity	Price per Unit

XAMPP X (any of four different operating systems), Apache, MySQL, PHP and Perl.	2	Free
Appserv Application Server	1	Free
Oracle Database 10g Express Edition	1	329.99$
Microsoft Office Access 2007	1	105.00$
Microsoft Windows 7	3	100$
Eclipse	1	30.95$
NetBeans IDE 6.9	1	33$
Total	798.94$	

The project team needs a set of hardware summarized in the following table:

Table1. 3 Hardware Resources Costs
Source: http://www.amazon.com/: amazon

Hardware resource	Description	Quantity	Price per Unit
Computer	CPU speed: 2.20GHz	3	$800

Total	2400$

The following table shows the total costs of human, software and hardware recourses:

Table1. 4 Total Costs

Human Resources	Hardware Resources	Software Resources	Total Costs
5400$	2400$	798.94$	8598.94$

- **Operational Requirement:**

Operational Hardware Recourses:

The following table list hardware resource required to operate this project and its cost.

Table1.5 Operational Hardware Resources Costs
Source: http://www.amazon.com/: amazon

Hardware resource	Description	Quantity	Price per Unit
Computer	CPU speed:3GHz	1	$800
Total	800$		

Operational Software Recourses:

The following table list software resources required to operate this project and their cost.

Table1.6 Operational Software Resources Costs
Source: http://www.amazon.com/: amazon

Software resource	Quantity	Price per Unit
Microsoft Windows 7	1	100$
XAMPP	1	Free
Hosting	1	150$ in one year
Total	250$	

Operational Human Recourses:

The following table list human resources needed to operate this project.

Table1.7 Operational Human Resources Costs
Source: http://www.amazon.com/: amazon

Role	Number	Cost/month	Total human cost/year
Data sharing system administrator	1	800$	9600$

Total Operational Costs:

Table1.8 Total Operational Costs

Hardware Operational Costs	Software Operational Costs	Human Operational Costs	Total

800$	250$	9600$	10650$

Browsing Requirements: The system needs at least one of the following browsers that support XML, JavaScript, Ajax: Internet explorer 6 and higher, Firefox 1.0 and higher, Safari 1.2 and higher, Netscape 7.0 and higher, Camino 1.0 and higher, and Opera 8.5 and higher.

1.8 Limitations and Constrains:

1. Lack of existence of a real extranet, between these two universities and the Palestinian Ministry of Education and Higher Education there must be extranet to allow communication among them.
2. Different technology and data provided by database for each university.
3. Authentication may limit the access process between the databases.

1.9 Risks:

1. Sudden changes that do not enable the team to continue to work as a team: in such situations the load on the remaining team member will be increased in order to finish in exact time.
2. The project team may be unable to complete project tasks on time.
3. Time-quality-cost: Significant changes in system requirements.
4. Volume of change requests following testing extending work on each phase.
5. Failure to achieve benefits of project.
6. Lack of commitment or ability to change current business processes .
7. Poor capture of full system requirements.
8. Poor data quality.
9. Poor of communication between universities.

1.10 Project Schedule:

This section shows the expected time needed for every task in the Project. The Table Below represents the work schedule:

Table2.9 Expected Duration Time for Project Tasks

Task Number	Task Name	Expected Time Needed in week
1	Planning	2
2	Literature Review	8
3	Models and Algorithms Comparison	2
4	Functional Requirements Specifications	2
5	Data Modeling	2
6	System Design	4
7	Development and Programming	8
8	System Testing	2
9	System Maintenance	2
10	Documentation	All over the Project Duration

The chart below represents the expected time needed to accomplish every task.

Table1.10 Gant Chart

Week / Task	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
Planning																
Literature Review																
Models and Algorithms Comparison																
Functional Requirements Specifications																
Data Modeling																

System Design																
Development and Programming																
System Testing																
System Maintenance																
Documentation																

1.11 Project Outline:

The organization of this project can be summarized as follows:

1. **Chapter two:** In this chapter, the project team will provide literature review and background, lists three models of data sharing and compare between them, the selected model to be implemented in the project and analysis of the selected model.

2. **Chapter three:** In this chapter, the project team will clarify what the system will do, the system functional requirements in details, the system non-functional requirements, depict the context diagram and data flow diagram of the system, and provide an explanation for the case study.

3. **Chapter four:** In this chapter, the project team will present Architecture Design, Process design for the system, data design including database design and interface design for the system.

4. **Chapter five:** In this chapter the project team will discuss the implementation phase at which more technical details of this project, software necessary to implement the system, how to build database, and selected code for necessary algorithms in the system.

5. **Chapter six:** in this chapter the project team will go through the four stages of testing: Unit and form testing, Integration testing, System testing, and Acceptance testing.

6. **Chapter seven:** in this chapter project team will clarify the maintenance process and the proposed plan to maintain the system which include: Backup, System Upgrading, System Upgrading.

Chapter Two

Background and Literature Review

Contents:

2.1 Introduction:

Electronic data interchange can be classified as one of the important areas of information technology, where the need for data sharing increasingly required in almost every field. This chapter provides a literature review and background, list three models of data sharing, a comparison between them, the selected model to be implemented in the project.

2.2 Data Sharing Concepts:

Data sharing concept emerges to introduce a new era of cloud computing processes, e-commerce, e-government, e-operations, e-everything. This term was coined since 1970s as Bakis et. al., (2006) indicates, there have been a number of well-coordinated national and international efforts to develop standard data models for the exchange of geometrical and graphical information, they add, from the early 1980s, the use of IT in the construction industry and broader engineering sector began to increase and find application in many different areas. The exchange of many different types of information was required at that time. As a result, a series of generic product data models were developed.

Greif and Sarin (1987) defines Data Sharing as a fundamental to computer-supported cooperative work; people share information through explicit communication channels and through their coordinated use of shared database. Where Sarathy and Muralidhar, (2004) describes data sharing as a fundamental enabler of coordination among supply chain partners. Therefore, data sharing can be defined as the process of interchanging, analyzing, retrieving and integrating data among multiple data sources in a controlled access manner. The data source is considered a destination in case of obtaining data from other data sources.

2.3 The need for data sharing:

Sarathy and Muralidhar (2004) find out that data sharing is an important feature of modern organizations due to the increase in the use of communication networks, changes in architectures of enterprise information systems, as well as the increasing availability of data in computerized form, and perhaps the biggest impact on data sharing can be attributed to the widespread use of the Internet and Internet-

related technologies for e-government and e-commerce. They clarify; e-government involves sharing data for transactions with citizens, other agencies and outside vendors and businesses.

Then Sarathy and Muralidhar (2004) adds, in e-commerce, data can be shared for transactions, operations, and analysis. Conducting business transactions is a basic reason for sharing data in e-commerce it is mainly used in Electronic Data Interchange (EDI), business to-business marketplaces, as well as consumer purchases over the web. They empathize that the focus of data sharing for operational purposes leads to the optimization of business processes over the entire chain to benefit all participants in the chain. Information that shared among supply chain partners may include inventory sales, demand forecasts, order status, and production schedules. Analysis, business intelligence, and decision-support represents the third purpose for data sharing in e-commerce, information available for analysis is increased through the sharing of data, they provides an example of banks data sharing with affiliates and telemarketers, another example about retailers who allow suppliers to access their inventory data for analysis purposes.

Where Mannai and Bugrara, (1993) indicates that it is highly desirable to share data among the members of the medical community; because data is a very valuable, hard to produce, and in some cases irreproducible resource. Data sharing reduces the cost of reproducing redundant data collections as much as minimizing the efforts paid in performing this.

Although data sharing facilitates the way that data can be exchanged, security concerns arises a challenge for the conducting of data sharing, confidentiality and privacy must be taken into consideration, this means, a controlled access is required to authorize authenticated users or roles to access data. Each data source represents a database, each database may use an application -for example- to access another database, this application is assigned specific permissions to access specific view of a specific database, permissions that identifies what kind of access must be granted to this application, (e.g. to read, or write, or even to have full access), for this purpose, a database of databases is needed to allow the sharing of data among the different databases as Mannai and Bugrara, (1993) indicates. This increases the need for the

data sharing management and data integration. To conduct the data sharing management, a specific model of the data sharing must be followed.

2.4 Management:

Several storage solutions were found to help data interaction between distributed and heterogeneous information sources, such solutions able to specify both data and control flow among these sources .In the following sections the project team introduce data integration, the environment of data sharing, importance issue in data sharing security .

2.4.1 Data integration:

Since data sharing coined, emerging data from heterogeneous sources into a single common to make data compatible with each other becomes critical issue. Harris et al,(2006) Data integration has been attempted for about 20 years. Hu and Yang, (2011) define data integration as the problem of combining the data from autonomous and heterogeneous sources, and providing users with a unified view of these data through so called global schema. They add, the design of data integration system is a very complex task it's include: heterogeneity of the data source, relation between the global schema and data sources limitation of accessing the sources.

Bakis et al, (2006) Classified data integration methods into three subtypes information linkage, query translation and data translation, Information linkage uses a Uniform Resource Locater (URL) to access data in an Hypertext Markup Language (HTML) form presented by other computation platforms through the Internet. Query translation is meant to convert source data on the fly and present data via a virtual data organization structure and usually is a part of data federation solution. This approach does not require physically storing an extra copy of data. Data translation is often associated with a data warehouse method. They add, end product of this approach is a physical copy of data that may not be presented or organized the same way as they were in their original storage systems.

Bakis et al, (2006) say past design analysis integration efforts have addressed the need to achieve seamless data integration between design and analysis software applications. They add, the main effort is on achieving integration via produce data sharing and exchange by considering the integration of design applications taking place at three levels : (1) the conceptual level : the basis for integrating two or more autonomous applications is to enable the applications to exchange information. The information must be first translated from one model to another, (2) the physical level: the aim of integration to minimize the need for building translation software, (3) the data management level: In order to maintain the consistency of the design data, the integrated environment needs to provide transaction management mechanisms that would enable its recovery after system crashes and control the concurrent access to the data.

2.4.2 Environment:

The goal of data design environment is to help data interaction between distributed and heterogeneous information sources. In the manufacturing sector, the design work is more centralized storing all the design data in a central repository maintained or owned by a single organization is viable option, storing all the design data in a single repository raises serious maintenance concerns. Bakis et al, (2006) add, in the construction industry, a distributed architecture that enables the project partners to share their data logically rather than physically represents a more viable option. These mechanisms provide access to the remote data and ensure that any multiple copies of the data in the environment remain totally synchronized such platform like distributed object technology, like Common Object Request Broker Architecture (CORBA), or Component Object Model (COM).

Bakis et al, (2006) also add, collaborative computing environments the access rights of each user depend on his/her role (e.g. designer, cost estimator, team leader, etc) and are valid only for the duration of the role. These special requirements have led to the development of a number of role-based and task-based access control models with role-based access control. Mannai et. al. ,(1993) in their multi-database environment which is tailored for the interoperability of medical information systems

, with the ability to specify multi-database applications users will be able to specify both data and control flow among the different systems accessed for the purposes of executing the application.

2.4.3 Security:

Security considerations apply not only to data held in database, as Connolly and Begg, (1996) indicate, database security encompasses hardware, software; people. They add, loss or damage of data results as a reason of intentional or unintentional acts ,intentional acts such theft and fraud result in either loss of confidentiality or loss of privacy, they also add, unintentional acts such viewing of unauthorized data cause loss of privacy and loss of confidentiality. Hardware frailer either intentional or accidental , program alteration ,data corruption due to power loss , also unintended sharing is a which cause data race — when two or more threads access the same memory location without synchronization, and at least one access is a write, all considered threats leads to the loss of security. Harris et. al. ,(2006) say that Cyber crime as well as threats to national security is costing organizations billions of dollars each year, it is equally certain that unrestricted data sharing will reduce the privacy and/or confidentiality of individuals.

Connolly and Begg, (1996) assert that one of the methods to struggle threats is make sure that Sensitive data must always be encrypted when stored or transmitted to force all access through a Policy Enforcement Point (PEP), another is to provide backup facilities to assist with the recovery of the database, views mechanism provides a powerful and flexible security mechanism by hiding parts of the database from users of data share. Harris et al, (2006) add share data under the data sharing agreement which provide secure data sharing.

Harris et. al., (2006) The challenge is to enforce appropriate administration and security policies that facilitate data sharing as needed .These policies include policies for confidentiality, privacy, and trust.

During normal operations, it is important to maintain confidentiality and privacy. In addition, trust policies ensure that data is shared between trusted individuals. The standards efforts in this area include Role-based access control (RBAC) as well as Platform for Privacy Preferences (P3P).

2.5 Interoperability:

Nowadays most web and internets use HyperText Markup Language (HTML) as a format for publishing information. HTML is easy to understand language for display and present data but also HTML suffers from shortcomings including: presentation orientation, tight coupling of content, no extensibility, and no data validation.

Seligman and Rosenthal, (2000) add, that the HTML gives no information about the meaning of the data, it just tell the browser how to display data, HTML has a fixed set of markup tags and HTML dose not validate data when it's entered or imported, because of these shortcomings the world wide web consortium (W3C) develop a new standard that eliminate these shortcomings, the result of the developing was called eXtensible Markup Language (XML) which is more extensible and capable of clearly separating content and presentation, the W3C defined XML as simplified subset of an earlier document-structuring language called Standardized Generalized Markup Language (SGML), SGML used to create large information collection, Sarathy and Muralidhar, (2004) add, technologies such as XML provide the ability to link and query heterogeneous sources of data across the internet.

Seligman and Rosenthal, (2000) indicates that XML has the potential to benefit databases and data sharing by providing a common format to express data structure and content, where Mandreoli et. al. , (2008) clarifies that the semantic support for data representation as well as the flexible machine-readable format have made XML the factual standard for internet application semantic interoperability.

Seligman and Rosenthal, (2000) lists many advantages for XML including: (1) independence of content and presentation, (2) extensibility, and (3) validation, also XML offers several benefits such as supporting multiple views of the same content for different user groups and media, selective query over internet and intranets, a standard infrastructure for data and document interchange this includes freely available parsers that can validate conformance with a Document Type Definition (DTD).

To support and describe the schema of XML documents and validating their contents, Document Type Definition (DTD) and XML schema is used, it's a schema that describes the structure of an XML document, also It indicates which elements appear in the document and which sub elements, attributes, and relations are allowed within each element. The schema can be used to validate the structure of the XML document automatically.

The use of XML Schema has many advantages to the system: (1) it offers a strict way of defining the structure of interchanged information; (2) It can be easily produced from the analysis of the physical model of information, it can be mapped into an XML-Schema and consequently into a database schema. (3) In opposition to DTDs, it has an object-oriented nature. It is easier for object oriented analysis tools to export the information structure to an XML-Schema file.

Varlamis and Vazirgiannis, (2001) add that many organizations and enterprises establish distributed working environment, where different users need to exchange information based on a common model, XML is used to facilitate this information exchange. The extensibility of XML allows the creation of generic models that integrate data from different sources and XML is becoming the standard format for data exchange among distributed applications components. The use of XML for information interchange among different enterprises and organizations evokes the need for common schema that the information must follow.

Seligman and Rosenthal, (2000) lists several standards will increase the utility of XML for data sharing and management, includes: (1) eXtensible Stylesheet Language (XSL), (2) Document Object Model (DOM), (3) XML Query Language and (4) XML schema.

The Extensible Stylesheet Language (XSL): expresses rules that indicate how to transform an XML document to a presentation format such as HTML or PDF, or to an alternate representation of the content such as an XML document with a different DTD. Document Object Model(DOM): provides a tree-based application programming interface to XML with methods for traversing the tree, such as getParentNode() and getChildNodes().XML query language: developed by the World Wide Web Consortium (W3C) working group for extracting data from XML

document collections as well as encapsulating non-XML data via mappings. Then they add, XML and XSL together will bring complete interoperability of both content and style across applications and platforms.

Seligman and Rosenthal, (2000) also add, XML can help to improve data sharing in: (1) Distribution: XML provides some assistance with distribution by supporting mechanisms for remote function invocation across the web,(2) Heterogeneous data structures and language: XML provides a neutral syntax for describing graph-structured data as nested, tagged elements with links,(3) Heterogeneous attribute representations and semantics*:* develop standards within a community of interest,(4) Heterogeneous schemas : XML DTDs defined by various communities provide a neutral model for describing their data structures. Communities developing standard DTDs include electronic commerce, healthcare, and data warehousing vendors. Such DTDs will reduce diversity of interfaces and ease data sharing, (5) Object identification: Improvements in identifying data elements can remove one source of misidentification of objects, and (6) Data value reconciliation*:* Many strategies for data value reconciliation depend on having metadata attached to the data. XML helps in attaching such annotations. XML also makes it easy to return a set of alternative elements for an uncertain value.

2.6 Models:

Different models have been introduced to apply data sharing each may be the same or different structure of other, some models introduced to extend previous models, others developed to overcome limitations and challenges of previous models. In this section the project team presents three models: Collaborative data sharing system model, capability-based access control model and Privacy-preserving model, and then, compare between them.

2.6.1 Collaborative Data Sharing System Model:

Green et. al., (2007) defines a model for a declarative, flexible, approach for data sharing, called the **Collaborative Data Sharing System (CDSS):**

1-The CDSS specify a local database's relationship to other sites. The CDSS organize conflicts in a custom way for each participant, based on whom and what it trusts, in that it allows the end user complete control over the contents of the local data instance.

2- a single entity may in turn be represented logically as a collection of a number of tuples in different relations, a set of updates are applied together or none are applied.

3- To enforce transactional atomicity between different participants, the CDSS considers transactions as the basic unit of operation, and it propagates, translates, and considers conflicts among such units. Furthermore, Green et. al., (2007) observe that data dependencies between operations in different transactions induce a dependency graph on the transactions themselves that must be respected when considering which transactions to accept or reject.

4-The CDSS consists of a network of collaborators participants or peers at independent sites; each has a local database instance and may be intermittently connected.

5-Each site spends the majority of its time operating in a locally autonomous mode, with users posing queries and making modifications directly over a local database instance, upon an administrator's request.

6- The CDSS performs an update exchange; this allows data to flow between participants in the system. The two basic operations of update exchange are publication and reconciliation (to accept the situation).

A-When a participant publishes its new transactions, the CDSS archives them which are needed in case participants are only intermittently connected and makes them available to the other participants in the system.

b- When a peer reconciles, the CDSS translates newly published transactions into that peer's schema, and then chooses a consistent subset of the translated candidate transactions to apply to its local instance, based on a set of user preferences, Figure (2.1) present an overview of the architecture of CDSS.

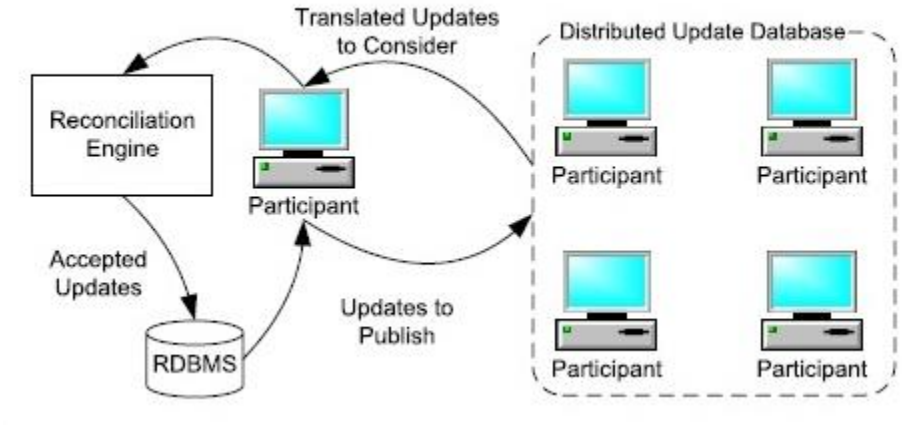


Figure2.1 An overview of the architecture of a CDSS

Source: Green et. al., (2007), P.1132.

The CDSS has three Features: (1) Any participant may make updates, including deletions, and this changes how data must be mapped and propagated in a peer-to-peer environment, (2) Each participant can ignore or even override updates it gets from elsewhere, using its own local updates, and (3) The CDSS must translate tuple-level updates in: Keeping track of their associated transactions, for purposes of conflict detection and resolution, Tracing their provenance, for purposes of trust assignment.

2.6.1.1 Collaborative Data Sharing System Benefits:

Green et. al., (2007) add that the CDSS model support Flexibility, functionality, and extensibility; it also reduces the barriers to sharing by allowing loosely coupled confederations of sites, each of which maintains a local schema and a fully autonomous, editable local data instance, in addition it considers transactions as the basic unit of operation.

2.6.1.2 Collaborative Data Sharing System Drawbacks

Despite of these features there still some difficulties for this model, the CDSS model relies on propagation of updates rather than data through the system, thus, there must be a method to translate updates over one schema to updates over a different schema, also because this model is based on distributed database, may create some shortcomings, first integrity control is more difficult, in distributed database enforcing integrity over a network may require too much of the network's resources to be

feasible, second distributed database is more complex than a centralized DBMS that's because of the fact that data can be replicated and if the software does not handle data replication adequately, there will be degradation in availability, reliability, and performance compared with the centralized system ,third Database Design in distributed database More Complex due to fragmentation, allocation of fragments to a specific site as Connolly and Begg, (1996) said.

2.6.2 Capability-based Access Control Model:

Geambasu et. al., (2007) use a model for data sharing called **Capability-based Access Control**, conceptually; each capability consists of a Name, which identifies a single object in the internet, and group of access rights for that object. In this model, the system sits between applications and the underlying file system. It presents applications a view-based interface to the file system. It executes queries over the local file system and communicates with other peers to evaluate distributed queries. The model is depicted through the following steps:

- The system registers each new view and capability in a local catalog, this capability has three parts (Figure 2):

1. A 128-bit global view Identification ID: this ID created by concatenating a hash of the local node's Media Access Control address (MAC address) with a locally unique-for-all-time view ID, this view ID uniquely identifies an individual view in the Internet.
2. A 128-bit random password: associated with each capability a 128-bit random password that ensures the capability's authenticity.
3. A 32-bit IP hint field: that contains the IP address of the node that likely contains or can locate the object addressed by the capability in the Peer to Peer Network (P2P), in general, they expect that objects will not move in their network, and the IP hint will be the address of the node that created the capability and still holds its definition. If the hint fails, then it must fall back on a conventional distributed hash-table scheme for location.

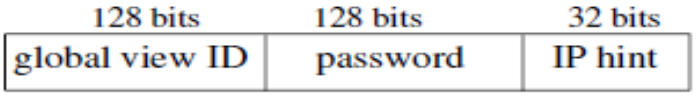


Figure2.2 Capability for a view

Source: Geambasu et. al., (2007), P.238.

- The per-node catalog table generated by the system holds view and capability information. It contains two tables *ViewTable* and *CapTable*(Figure 2.2). The ViewTable entry contains the global view ID, the view definition, and other attributes (such as the human-readable view name). For each view created on a node, there is one entry in a local view table (*ViewTable*). The CapTable entry stores the global view ID of the named view, the password, and the access rights. A node's capability table (CapTable) contains one entry for each capability minted to a locally known view.

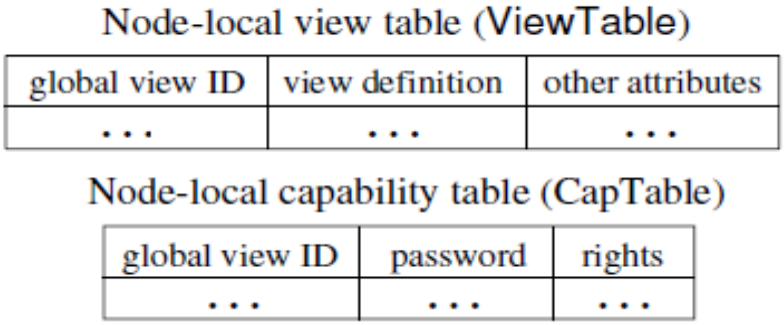


Figure2.3 Capability and view Catalog tables.

Source: Geambasu et. al., (2007), P.238.

- Users grant each other access to their data simply by exchanging capabilities to their views, much like users share access to private web pages by exchanging URLs.
- When the system receives a capability, it uses the IP hint to determine whether the capability is for a local view. If the capability is local, the system checks whether the <global view ID, password> pair in the capability matches a <global view ID, password> pair in CapTable. If so, the capability is valid, and the system then examines the access rights in CapTable to see if the requested operation is permitted. If the capability is not found in CapTable or the operation is not permitted, the request fails. If the capability is for a remote view, the system forwards the request to the

appropriate node in the peer-to-peer network, which then performs the validation itself.

- To revoke a capability, the system simply removes an entry from the CapTable. Once a capability is revoked, all queries issued on that capability will fail.

2.6.2.1 Capability-based Access Control Benefits:

Geambasu et. al., (2007) add that Capability Based Access control model is a flexible protection mechanism for controlling access to shared views. Capabilities also enable rewriting and optimization of distributed queries, leading to good query execution performance. They also add, because capability is independent of the person using it, the systems' access control scheme requires no user identities. Thus, sharing in a capability-based model requires no user accounts, no user authentication, and no centralized protection structure. Capabilities facilitate data sharing because it can easily passed from user to user as a way to grant access.

2.6.2.2 Capability-based Access Control Drawbacks:

After revoking a capability, all queries issued on that capability will fail. But if a user with a capability has made a local copy of the shared data, revoking the capability cannot prevent him from distributing that copy. However, it prevents the holder from executing a query and seeing new or modified files that would result from that query.

2.6.3 Semantic Privacy-Preserving Model:

Hu and Yang (2011) present Semantics privacy preserving model:

A semantic privacy-preserving model provides authorized view-based query answering over a widespread multiple servers for data sharing and integration. For that reason model consider a large number of servers. Therefore a unified global data sharing and protection service can be achieved at the virtual platform (VP).

1- The combined semantics-enabled privacy protection policies are used to empower the data integration and access control services at the (VP). Privacy protection policies represent a long-term promise made by an enterprise to its users and is determined by business practice and legal concerns, which is expressed as combination ontology and rule:

- A privacy protection policy is a type of formal policy (FP) used for specifying a data usage constraint from a data owner. FP is a declarative expression corresponding to a human legal norm that can be executed in a computer system without causing any semantic ambiguity.
- An FP is created from a policy language (PL), and this PL is shown as a combination of ontology language and rule language.
- A formal protection policy (FPP) is an FP that aims at representing and enforcing resource protection principles, where the structure of resources is modeled as ontology's O but the resources protection is shown as rules R.(It is combination of ontology's and rules O+R)
- Semantic Web Rule Language (SWRL) Tab development tools and Semantic Query-Enhanced Web Rule Language (SQWRL), Web Ontology Language (OWL-DL) query language to model and enforce semantic privacy protection policies.

2- Three approaches have been proposed to model a set of source descriptions that specify the semantic mapping between the source schema and the global schema:

- Global-as-view (GAV) requires that the each concept in the global schema is expressed in terms of query over the data sources.
- Local-as-view (LAV) requires the global schema to be specified independently from the sources, and the source descriptions between the stable global schemas.
- Global-local-as-view (GLAV), a source description that combines the expressive power of both GAV and LAV, allowing flexible schema definitions independent of the particular details of the data sources.

3- This model is proposed with three layers, where the bottom layer provides data sources from the relational databases .The middle layer provides a semantics- enabled local schema for each independent service domain. The top layer is served at the VP, which provides a unified global view of privacy-preserving data sharing and integration services.

4-The ontology mapping and merging algorithm with a local-as-view (LAV) source description that creates a global ontology schema (mediated), which is a reconciled view of the information that provides query services to end users ,at the VP by

integrating multiple local ontology schemas for data sharing. Model merged global ontology schema that mentioned above in the middle layer.

5- Using description logic (DL) to model the local and global schemas is to empower the ontology's abstract Concept representation and reasoning capabilities.

6- A query is defined as an SQWRL data log rule in the SWRL-based policy to access to a global ontology, and each SQWRL data service query for a global ontology at the VP is mapped to multiple queries as SQWRL data log rules for each local schema.

7-The challenge of designing a semantic privacy protection model is to ensure soundness and a completeness of data sharing and protection in multiple servers:

- For the soundness criterion, this model does not allow unintended data being released to the data users through the global policy schema (GPS) at the VP.
- As for the completeness criterion, the model does not miss any eligible shared data when a user asks for a data request service at the VP. Therefore, shareable data obtained the VP should equal data obtained directly from each server.

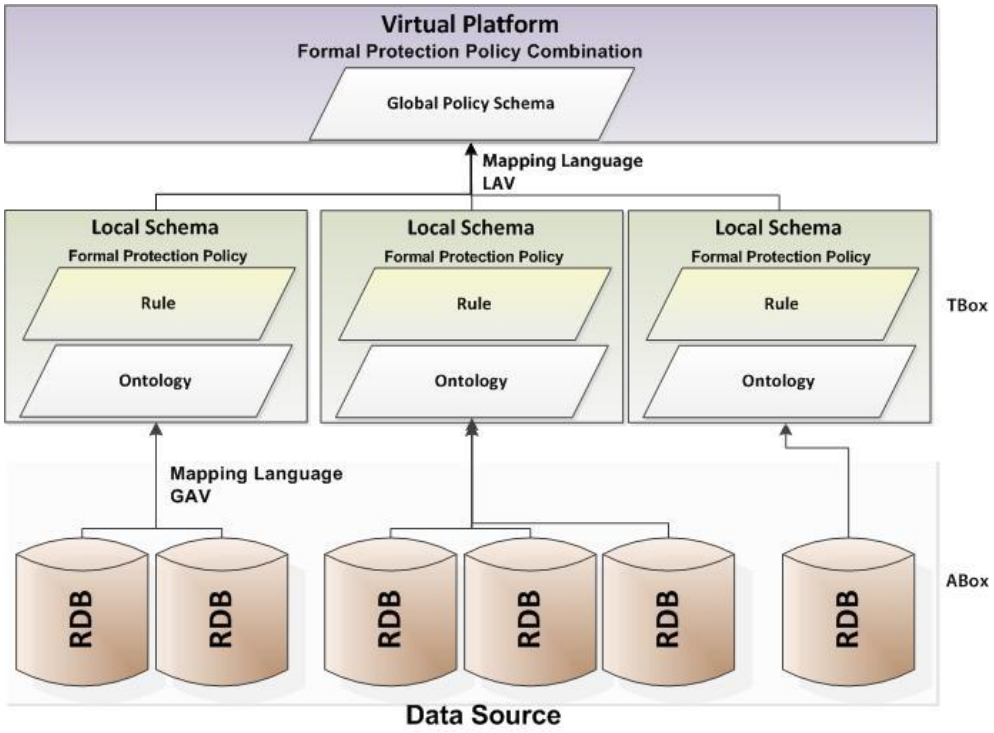


Figure2.4 A semantic privacy protection model

Source: Hu and Yang, (2011), P.4.

(Figure 2.4) is proposed with three layers, where the bottom layer provides data sources from the relational databases (RDB), the middle layer provides a semantics-enabled local schema for each independent service domain. The top layer is served at the VP, which provides a unified global view of privacy-preserving data sharing and integration services. In the top layer at the VP, we have a global policy schema (GPS), including a global ontology schema (GS) aligned and merged from several local schemas (LS), e.g. TBox and a set of rule integration at the middle layer. The VP provides conceptual data access and protection services that give users a unified conceptual\global view" with access control power for each data request. Ontology-based data sources are external, independent, and heterogeneous, and each local ontology was combined with logic program (LP)-based rules for each server in the middle layer. Mapping language (ML), which semantically links a GS and integrated rule, set in the top layer to each server's ontology LS and privacy protection rules in the middle layer. Ontology and the dynamic data sources are established by defining each concept in the data sources as a view over the global schema.

2.6.3.1 Semantic Privacy-Preserving Benefits:

Hu and Yang, (2011) list some features in semantics privacy preserving model, First ,each server shares its collected data with other servers but without breaking the original data usage commitment to its clients ,therefore a unified global data sharing and protection service can be achieved at the virtual platform (VP). Second ,the model solve the soundness and completeness of query rewriting problem using a perfect ontology merging and a perfect rule integration from the local formal protection policies, For the soundness criterion, we do not allow unintended data being released to the data users. As for the completeness criterion, we do not miss any eligible shared data when a user asks for a data request service at the VP, Third, the model develop a privacy management framework and a formal semantics language to empower agents to enforce privacy protection policies. These formal policy using ontology for privacy protection concept descriptions and rule for data query and access control services. Ontology-based data integration in DL is to provide a uniform access mechanism to a set of heterogeneous relational database sources, freeing the user from having the

knowledge about where the data are, what they are stored, and how they can be accessed.

2.6.3.2 Semantic Privacy-Preserving Drawbacks

In spite of these features ,this model still have a drawback , it face a background policy inconsistency problem when default policy assumptions vary between different servers (one server uses open policy assumption, where no explicit option-out for data usage mean option-in, but the other server uses closed policy assumption, where no explicit option-in for data usage means option-out) and to avoid this kind of policy inconsistency by requesting all sites to use a uniform policy assumption, and to collect option-in data usage choices from users whenever multiple policies are integrated.

2.7 Comparison between CDSS model, Capability-based Access Control model, and Semantic Privacy-Preserving model:

The following table provides a comparison between the three models in terms of their advantages and disadvantages:

Table 2.1 Models Comparison

Model Name	Advantages	Disadvantages
CDSS model	1. Support Flexibility, functionality, and extensibility. 2. Reduces the barriers to sharing. 3. Basic unit of operation is the transaction.	1. No fixed method for translation. 2. Difficulties in integrity control.
Capability-based Access Control model	1. Provide flexible protection mechanism for controlling access to shared views. 2. Reuse of queries.	1. Cannot prevent the user from keeping and distributing the shared data.

	3. Independent of the user.	2. Decentralized control.
Semantic Privacy-Preserving model	1. Each organization enables data sharing without affecting its clients. 2. Data integration. 3. Provide secure sharing through authorized views.	1. Inconsistency problems.

2.8 Selection of best model (Alternative):

Based on the previous comparison between the three models, the first two models have disadvantages, mainly: there is no fixed method for translation, difficulties in integrity control, Cannot prevent the user from keeping and distributing the shared data, and Decentralized control, and these disadvantages make the implementation of the models very hard, and don't achieve the minimum requirements for EDI, concerning semantic privacy preserving model overcomes the previous disadvantages. And provides data integration, secure sharing through authorized view, in addition, each organization enable data sharing without affecting its clients.

2.9 Analysis of the model:

Data Sharing Management System: is a system that enables and manages data sharing and interchange between two or more organizations employing XML standard. This system will be implemented depending on the selected model (semantic privacy-preserving model) and an extranet for communication.

Therefore, the functional requirement of the proposed system will be: Electronic Data Interchange (EDI): each organization is accept to share their data with other organizations by specifying access control criteria, Generating and mapping XML documents and Converting data into a presentational formats (i.e. reports, statistics).

Where the non-functional requirements provided by the model are: security; since the system will provide authorized requests in the virtual platform.

2.10 Summary:

In this chapter, the project team provided a literature review and background, listed three models of data sharing, a comparison between them, and selected a model to implement in the project. In the next chapter the project team will clarify what the system will do, the system functional requirements in details**,** the system non-functional requirements, depict the context diagram and data flow diagram of the system and provide an explanation for the case study.

Chapter Three

System Specifications

Content:

3.1 Introduction:

Determining the systems requirements is the crucial phase of the SDLC that depicts the systems specifications. Functional requirements are considered the building blocks for every project to be implemented. Where the non-functional requirements: is any requirement, that cannot be categorized as functional. In this chapter, the project team will clarify what the system will do, the system functional requirements in details**,** the system non-functional requirements, depict the context diagram and data flow diagram of the system and provide an explanation for the case study.

3.2 General idea of the system:

The Data sharing management system consist of two universities, which are Palestine Polytechnic University, Birzeit University. The Project team use the Palestinian Ministry of Education and Higher Education as the Virtual Platform(VP), which contains a centralized database that contain authorized requests and authenticated people with all privileges to direct the operations that may occur between universities. In other word we can say the Palestinian Ministry of Education and Higher Education is the authorized entity to coordinate between universities.

When one university needs to request data from the other, this request must match one of the specified categories that exists in the VP, these categories is specified based on an agreement between the two universities. Both universities cannot add a new request that is not within the specified categories. After adding a request, the requested university will be responded with XML Document that contain requested data, XML Document generated through PHP file, then the VP notified with the request, the generated XML document have two way to pass, it may be mapped into the database with the same technology of the destination university for example (transfer a student or a course), or converted into reports ,statistics that presented in the browser using web application for example (student marks in his registered course), after the request is completed, the other university will be notified, the VP itself can also request for some kinds of reports from the two university databases which is called integration

for example (all students who has average greater than a specified value from PPU and BZU).

3.3 Functional Requirements in Details:

3.3.1 Electronic Data Interchange (EDI): each organization accept to share their data with other organizations by specifying request categories in the virtual platform**.**

Table3.1 Electronic Data Interchange

Function	Data interchange between two database with different technologies.
Description	Data request from one database send to the VP, and the other university, then the other university send a response as XML document.
Input	Data requests
Output	Response (XML document, Row in database)
Requirements	Authorizes request

3.3.2 Generating XML documents: using Java, Mapping XML into a relational database.

Table3.2 Generating XML documents

Function	Generating XML documents, mapping XML documents into a relational database.
Description	Generating XML documents in the response side , then mapping these XML documents into a relational database matching technology used in the other database and vice versa
Input	Table in the database, XML Document
Output	XML Doc, Table in the database.
Requirements	Request Categories in the Virtual Platform

3.3.3 Converting data into a presentational format (Visualizing):

Table3.3 converting data into a presentational format (visualizing)

Function	Converting Xml documents into a presentational format.
Description	Using PHP to convert documents into presentational formats such as presenting data in browsers.
Input	XML Document.
Output	Reports and statistics in PDF format .
Requirements	XML Doc in the Virtual Platform

3.4 Non-functional requirements:

The project team had been choose the semantic privacy preserving model for the system implementation, which provides the system with two non-functional requirements: Security and Integration, in addition to those two, the system will support usability, which is achieved through five design guidelines: Consistency, Efficiency, Ease, Format, Flexibility as Hoffer et. al. indicates. Described below:

Consistency: the system will employ a consistent use of terminology, abbreviations, formatting, titles, and navigation within and across outputs, besides, the system will have a consistent response time each time a function is performed.

Efficiency: formatting will be designed with an understanding of the task being performed and the intended user. Text and data will be aligned and stored for efficient navigation and entry, entry of data will be avoided in most processes, because the system depending on data existed in databases.

Ease: output will be self-explanatory and won't require users to remember information from prior outputs in order to complete tasks, labels should be extensively used.

Format: information format will be consistent. Format will distinguish each piece of data and highlight, not bury, important data.

Flexibility: information will be viewed and retrieved in a manner most convenient to the user. Users will be given options for the sequence in which to enter data and for use of shortcut keystrokes.

3.5 Context Diagram:

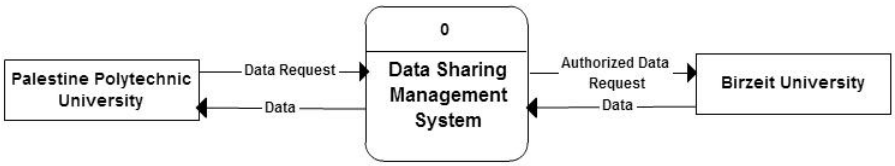


Figure 3.1 Context Diagram

3.6 Data Flow Diagram:

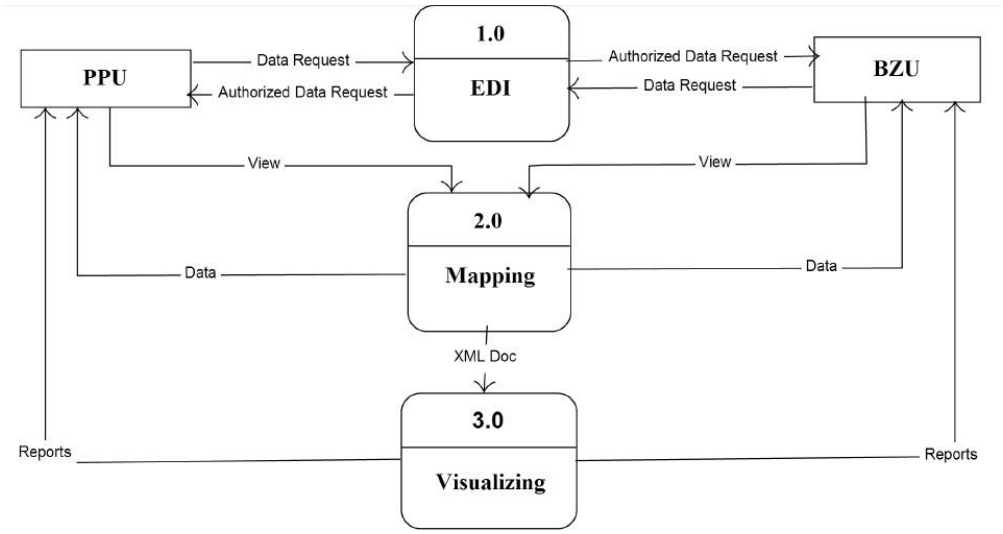


Figure 3.2 Data Flow Diagram

3.7 Case Study Explanation:

3.7.1 Transfer Student

This operation available for the both universities, it provides the ability to transfer student information from one of them to the other, including personal and educational information by entering student number and choosing what information to be transferred. At this operation, the origin university would still keeping the

information about the transferred student and changing the academic status for the student to transferred student.

3.7.2 Transfer Course

This operation represent the ability to transfer course information from one university to the other, including the course information such as course credit, level and type, by entering course number and choosing what information to be transferred. This operation would be useful when one of these universities want to offer a course that exists in the other university.

3.7.3 Request for a Report

This operation allows the VP and the both universities to get information from the both universities in one report, it also allows the two universities to get information from each other as a report, and the VP have the ability to get information from one of them as a report.

3.8 Summary:

In this chapter we describe the general idea of the project , analysis the functional and non-functional requirement and provide an explanation for the case study. In the next chapter the project team will clarify the design of the system.

Chapter Four

System Design

Content:

4.1 Introduction:

In this chapter, the project team will present Architecture Design, Process design for the system, data design including database design and interface design for the system.

4.2 Architecture Design:

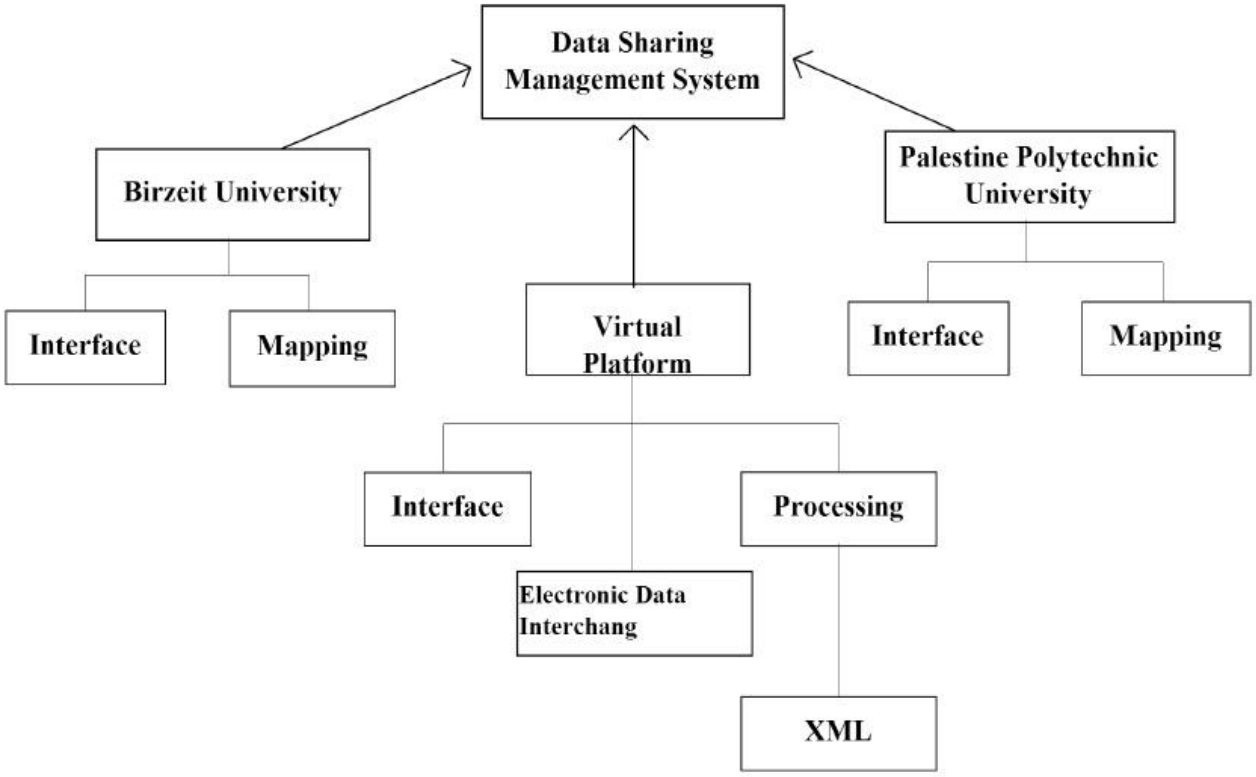


Figure4.1 Architecture Diagram

This figure depicts the three parties of data sharing management system: Palestine Polytechnic university, birzeit university, and virtual platform. Each of the two universities contain two components, interface and mapping processing, while the VP contains another two components, which are interface and XML processing (reports using extendable Stylesheet Language Transformation XSLT).

4.3 Processes Design:

4.3.1 Electronic Data Interchange (EDI):

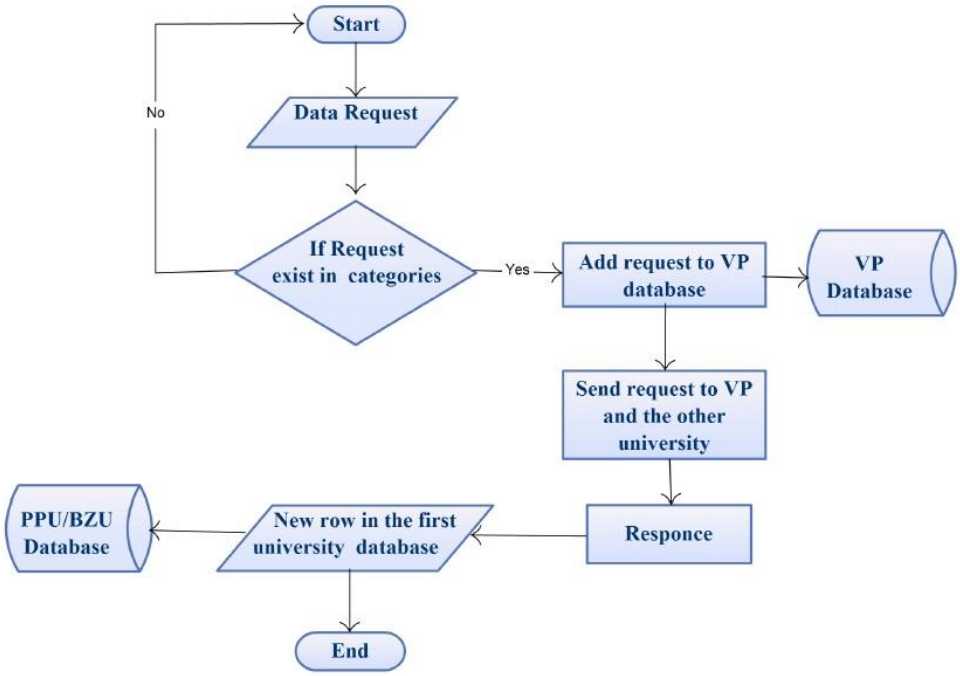


Figure 4.2 : EDI Flowchart.

4.3.2 Mapping:

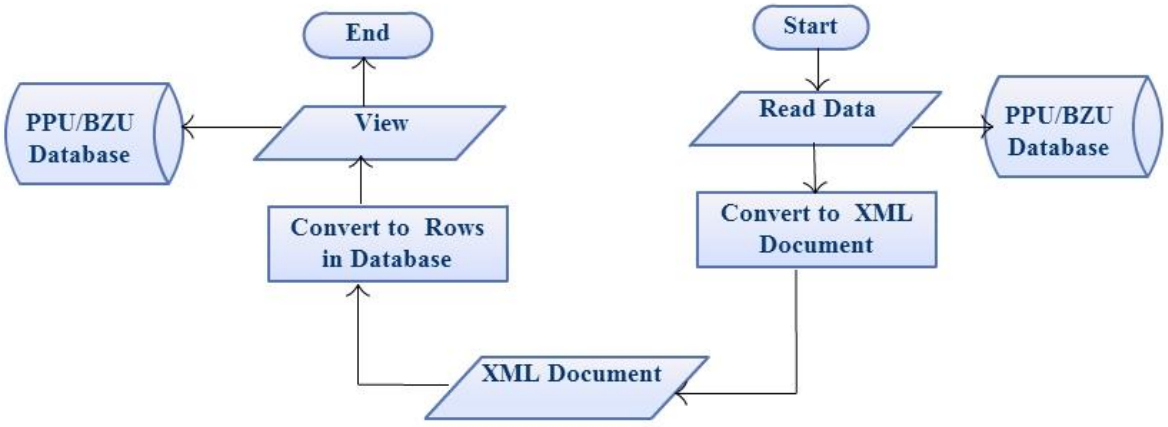


Figure 4.3 : Mapping Flowchart.

4.3.3 Visualizing:

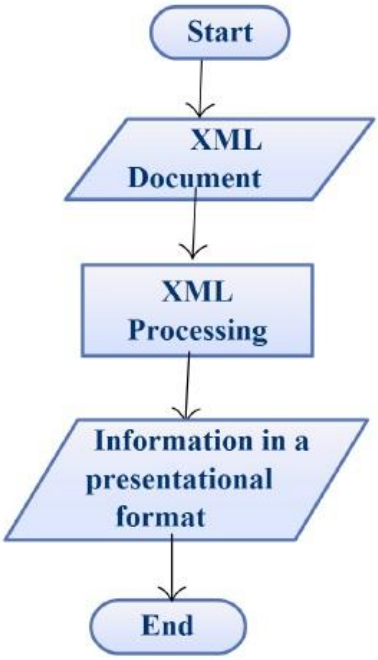


Figure 4.4 : Visualizing Flowchart.

4.4 Data Design:

Database Design:

In this section the project team will clarify system databases which are part of PPU database and BZU database and the Virtual Platform database, and present database tables and their field's details and Database Diagram.

- **Palestine Polytechnic University Database:**

Which is built on Microsoft Office Access.

Database Diagram :

This Diagram clarify the relationship between database tables.

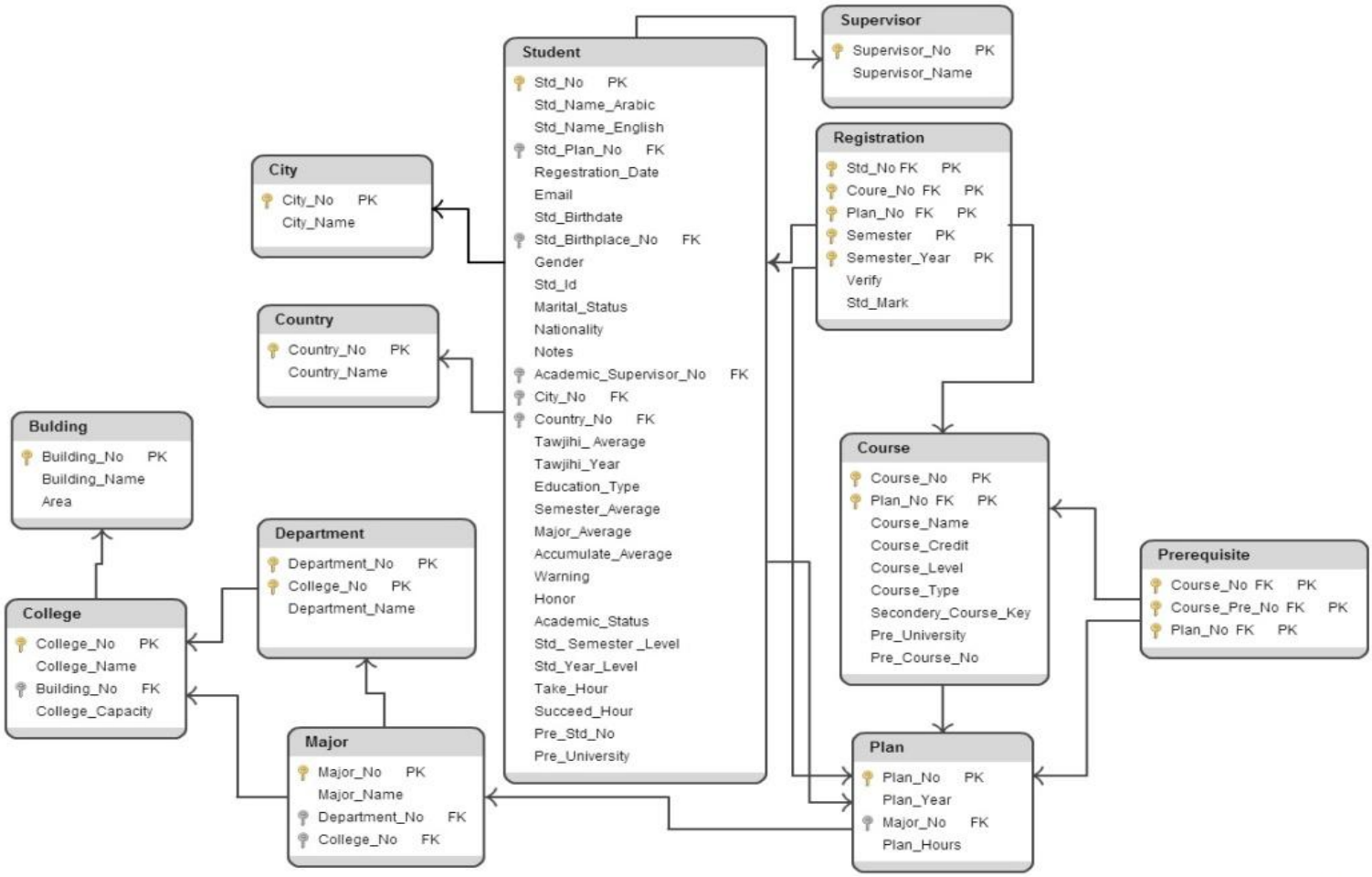


Figure4.5 Database Diagram(PPU).

Student Table:

Table4. 25 Student Table

Field name	Field type	Null able	Keys	References	Field length	description
Std_No	Number	No	PK	-	Long integer	A unique number for the student
Std_Name_Arabic	Text	No	-	-	50	Full student name in Arabic
Std_Name_English	Text	No	-	-	60	Full student name in English
Std_Plan_No	Number	Yes	FK	Plan Table	Long Integer	Student plan number
Regestration_Date	Date/time	Yes	-	-	Date	Registration

						date
Email	Text	No	-	-	50	Student email
Std_Birthdate	Date/time	Yes	-	-	Date	Student birth date
Std_Birthplace_No	Number	Yes	FK	City Table	Long Integer	Number of Student birth place
Gender	Number	No	-	-	Byte	Student gender: takes two values(0 for Male, 1 for Female).
Std_Id	Number	No	-	-	Long Integer	Student ID Number
Marital_Status	Number	Yes	-	-	Byte	Marital Status: takes for values: (0 for single, 1 for Married, 2 for Divorced, 3 for Widowed).
Nationality	Number	Yes	-	-	Byte	Student Nationality: takes four values (0 for Palestinian, 1 for Jordanian, 2 for Arabic, 3 for Other)
Notes	Text	Yes	-	-	255	Notes about student
Academic_Supervisor_No	Number	Yes	FK	Supervisor Table	Long Integer	Number of Academic supervisor
City_No	Number	Yes	FK	City Table	Long Integer	Area number for the student
Country_No	Number	Yes	FK	Country Table	Long Integer	Country Number
Tawjihi_ Average	Number	No	-	-	Double	Student Tawjihi Average
Tawjihi_Year	Number	No	-	-	Integer	Year when the student graduated from tawjihi

Education_Type	Number	Yes	-	-	Byte	Type of education: takes two values (0 for normal , 1 for parallel)
Semester_Average	Number	Yes	-	-	Double	Average of the semester
Major_Average	Number	Yes	-	-	Double	Average of the major courses
Accumulate_Average	Number	Yes	-	-	Double	Average cumulative
Warning	Number	Yes	-	-	Byte	Number of warnings
Honor	Number	Yes	-	-	Byte	Honors
Academic_Status	Number	Yes	-	-	Byte	Academic Status: takes three values (0 for regular, 1 for postponed, 2 for dismissed, 3 for transferred)
Std_ Semester _Level	Number	Yes	-	-	Byte	Student level in semesters
Std_Year_Level	Number	Yes	-	-	Byte	Student level in years
Take_Hour	Number	Yes	-	-	Byte	Total Hours registered by student.
Succeed_Hour	Number	Yes	-	-	Byte	Total Hours successfully passed by student.
Pre_Std_No	Number	Yes	-	-	Long Integer	Student Number in the previous university if he was.
Pre_University	Text	Yes	-	-	8	Character that indicates for previous university P for PPU and B for BZU.

Course Table:

Table4. 26 Course Table

Field name	Field type	Null able	Keys	References	Field length	Description
Course_No	Number	No	PPK	-	Long Integer	Course number Sequential number
Plan_No	Number	No	FK,PPK	Plan Table	Long Integer	Plan Number
Course_Name	Text	No	-	-	30	Course Name
Course_Credit	Number	No	-		Byte	Course credit
Course_Level	Number	No	-	-	Byte	Level of course
Course_Type	Number	No	-	-	Byte	Type of course: takes six values(0 for Department Requirement, 1 for college Requirement, 2 for University Requirement, 3 for Compulsory major requirement, 4 for optional major Requirement, 5 for Optional University Requirements).
Secondery_Course_Key	Text	No	-	-	10	A key that describes the course, and the major that the course follows.
Pre_University	Text	Yes	-	-	8	Character that indicates for previous university P for PPU and

						B for BZU.
Pre_Course_No	Number	Yes	-	-	Long Integer	Course Number in the previous university if it was transfer .

Registration Table:

Table4. 27 Registration Table

Field name	Field type	Null able	Keys	References	Field length	Description
Std_No	Number	No	PPK,FK	Student Table	Long Integer	A unique number for the student
Coure_No	Number	No	PPK,FK	Course Table	Integer	Course number
Plan_No	Number	No	PPK,FK	Plan Table	Long Integer	The plan where course belong
Semester	Number	No	PPK		Byte	Semester Number takes two values (1 for the first semester, 2 for the second)
Semester_Year	Number	No	PPK		Integer	Date of semester
Verify	Number	Yes	-	-	Byte	Confirmation : takes two values(0 for Yes, 1 for NO)
Std_Mark	Number	Yes	-	-	Double	Student mark in the course

Prerequisite Table:

Table4. 28 Prerequisite Table

Field name	Field type	Null able	Keys	References	Field length	Description
Course_No	Number	No	PPK,FK	Course Table	Long Integer	Course number
Course_Pre_No	Number	No	PPK,FK	Course Table	Long Integer	Prerequisite course number
Plan_No	Number	No	PPK,FK	Plan Table	Long Integer	The plan

						where course belong

Plan Table:

Table4. 29 Plan Table

Field name	Field type	Null able	Keys	References	Field length	Description
Plan_No	Number	No	PK	-	Long Integer	Plan number, Sequential number
Plan_Year	Number	No		-	Integer	Plan Date
Major_No	Number	No	FK	Major Table	Long Integer	Major number
Plan_Hours	Number	No	-	-	Byte	Number of hours in the plan

Major Table:

Table4. 30 Major Table

Field name	Field type	Null able	Keys	References	Field length	Description
Major_No	Number	No	PK	-	Long Integer	A unique number for the major
Major_Name	Text	No	-	-	30	Major name
Department_No	Number	No	FK	Department Table	Byte	Department Number
College_No	Number	No	FK	College Table	Byte	College Number

Department Table:

Table4. 31 Department Table

Field name	Field type	Null able	Keys	References	Field length	description
Department_No	Number	No	PPK	-	Byte	Sequential Number
College_No	Number	No	PPK,FK	College Table	Byte	College Number
Department_Name	Text	No	-	-	50	Department Name

College Table:

Table4. 32 College Table

Field name	Field type	Null able	Keys	References	Field length	Description
College_No	Number	No	PK	-	Byte	A unique number for the college
College_Name	Text	No	-	-	50	College name
Building_No	Number	No	FK	Building Table	Long Integer	College Building
College_Capacity	Number	Yes	-	-	Integer	College Capacity.

Supervisor Table :

Table4. 33 Supervisor Table

Field name	Field type	Null able	Keys	References	Field length	Description
Supervisor_No	Number	No	PK	-	Long Integer	A unique number for the Supervisor, Sequential number
Supervisor_Name	Text	No	-	-	40	Supervisor name

Building Table:

Table4. 34 Bulding Table

Field name	Field type	Null able	Keys	References	Field length	Description
Building_No	Number	No	PK	-	Long Integer	Sequential Number
Building_Name	Text	No	-	-	10	Building Name
Area	Text	Yes	-	-	15	Name of the Area where the Building exist

City Table:

Table4. 35 City Table

Field name	Field type	Null able	Keys	References	Field length	Description
City_No	Number	No	PK	-	Long Integer	Sequential Number
City_Name	Text	No	-	-	15	City Name

Country Table:

Table4. 36 Country Table

Field name	Field type	Null able	Keys	References	Field length	Description
Country_No	Number	No	PK	-	Long Integer	Sequential Number
Country_Name	Text	No	-	-	20	-

- **Birzeit University Database :**

  Which is a simulation for PPU database but built on Oracle Database Management System.

Database Diagram:

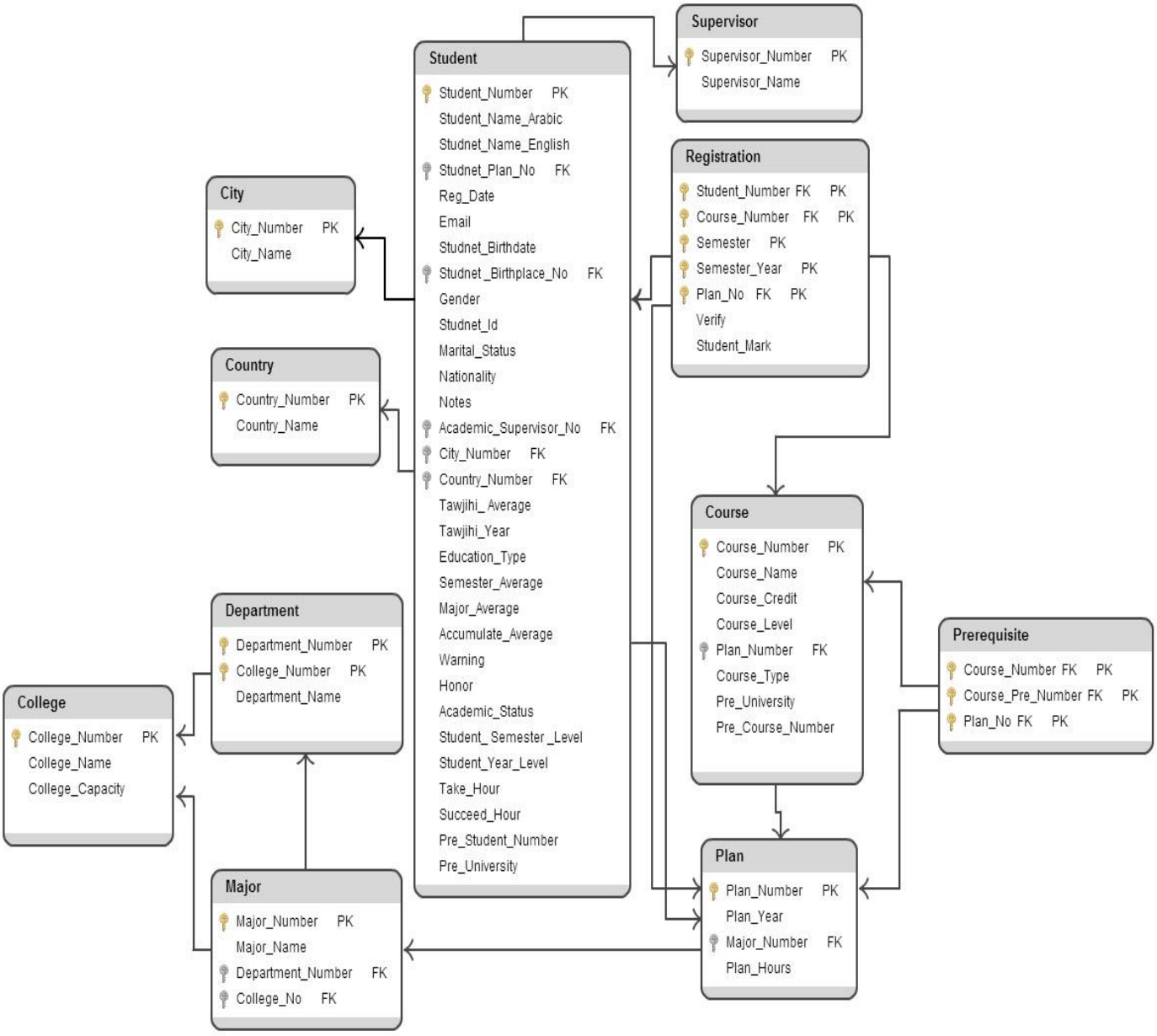


Figure 4.6 Database Diagram(BZU)

Student Table:

Table4. 37 Student Table

Field name	Field type	Null able	Keys	References	Field length	Description
Student_Number	Number	No	PK	-	9	A unique number for the student
Student_Name_Arabic	Varchar2	No	-	-	50	Full student name in Arabic
Studnet_Name_English	Varchar2	No	-	-	60	Full student name in English
Studnet_Plan_No	Number	Yes	FK	Plan Table	2	Student plan number
Reg_Date	Date	Yes	-	-	-	Registration date
Email	Varchar2	No	-	-	50	Student email
Studnet_Birthdate	Date	Yes	-	-	-	Student birth date
Studnet _Birthplace_No	Number	Yes	FK	City Table	3	Number of Student birth place
Gender	Number	No	-	-	1	Student gender: takes two values(0 for Male, 1 for Female).
Studnet_Id	Number	No	-	-	10	Student ID Number
Marital_Status	Number	Yes	-	-	1	Marital Status: takes four values: (0 for single, 1 for Married, 2 for Divorced, 3 for Widowed).
Nationality	Number	Yes	-	-	1	Student Nationality: takes four values (0 for Palestinian, 1 for Jordanian, 2 for Arabic, 3 for Other)

Notes	Varchar2	Yes	-	-	255	Notes about student
Academic_Supervisor_No	Number	Yes	FK	Supervisor Table	3	Number of Academic supervisor
City_Number	Number	Yes	FK	City	3	Area number for the student
Country_Number	Number	Yes	FK	Country	3	Country Number
Tawjihi_ Average	Number	No	-	-	2.2	Student Tawjihi Average
Tawjihi_Year	Number	No	-	-	4	Year when the student graduated from tawjihi
Education_Type	Number	Yes	-	-	1	Type of education: takes two values (0 for normal , 1 for parallel)
Semester_Average	Number	Yes	-	-	2.2	Average of the semester
Major_Average	Number	Yes	-	-	2.2	Average of the major courses
Accumulate_Average	Number	Yes	-	-	2.2	Average cumulative
Warning	Number	Yes	-	-	1	Number of warnings
Honor	Number	Yes	-	-	1	Honors (1 for Yes, 0 for No)
Academic_Status	Number	Yes	-	-	1	Academic Status: takes three values (0 for regular, 1 for postponed, 2 for dismissed, 3 for transferred)
Student_ Semester _Level	Number	Yes	-	-	2	Student level in semesters
Student_Year_Level	Number	Yes	-	-	1	Student level in years
Take_Hour	Number	Yes	-	-	3	Total Hours registered by

						student.
Succeed_Hour	Number	Yes	-	-	3	Total Hours successfully passed by student.
Pre_Student_Number	Number	Yes	-	-	10	Student Number in the previous university if it was transfer .
Pre_University	Varchar2	Yes	-	-	8	Character that indicates for previous university P for PPU and B for BZU.

Course Table:

Table4. 38 Course Table

Field name	Field type	Null able	Keys	References	Field length	Description
Course_Number	Number	No	PK	-	15	A key that describes the course, and the major that the course follows.
Course_Name	Varchar2	No	-		50	Course Name
Course_Credit	Number	No	-		1	Course credit
Course_Level	Number	No	-	-	2	Level of course
Plan_Number	Number	No	PK	Plan Table	2	Plan Number
Course_Type	Number	No	-	-	1	Type of course: takes six values(0 for Department Requirement, 1 for college Requirement, 2 for University Requirement, 3 for Compulsory major requirement, 4 for optional

						major Requirement, 5 for Optional University Requirements).
Pre_University	Varchar2	Yes	-	-	8	Character that indicates for previous university P for PPU and B for BZU.
Pre_Course_Number	Number	Yes	-	-	8	Course Number in the previous university if it was transfer .

Registration Table:

Table4. 39 Registration Table

Field name	Field type	Null able	Keys	References	Field length	Description
Student_Number	Number	No	PPK,FK	Student	9	A unique number for the student
Course_Number	Number	No	PPK,FK	Course	5	Course number
Semester	Number	No	PPK		2	Semester Number takes two values (1 for the first semester, 2 for the second)
Semester_Year	Number	No	PPK		4	Date of semester
Plan_No	Number	No	PPK,FK	Plan Table	2	
Verify	Number	Yes	-	-	1	Confirmation :

						takes two values(0 for Yes, 1 for NO)
Student_Mark	Number	Yes	-	-	2	Student mark in the course

Plan Table:

Table4. 40 Plan Table

Field name	Field type	Null able	Keys	References	Field length	Description
Plan_Number	Number	No	PK	-	2	Plan number
Plan_Year	Number	No		-	4	Plan Date
Major_Number	Number	No	FK	Major	5	Major number
Plan_Hours	Number	No	-	-	3	Number of hours in the plan

Major Table:

Table4. 41 Major Table

Field name	Field type	Null able	Keys	References	Field length	Description
Major_Number	Number	No	PK	-	5	A unique number for the major
Major_Name	Varchar2	No	-	-	50	Major name
Department_Number	Number	No	FK	College	5	Department Number
College_No	Number	No	FK	College	5	

Prerequisite Table:

Table4. 42 Prerequisite Table

Field name	Field type	Null able	Keys	References	Field length	Description
Course_Number	Number	No	PPK,FK	Course	5	Course number
Course_Pre_Number	Number	No	PPK,FK	Course	5	Prerequisite course number
Plan_No	Number	No	PPK,FK	Plan	2	

Department Table:

Table4. 43 Department Table

Field name	Field type	Null able	Keys	References	Field length	Description
Department_Number	Number	No	PK	-	5	Sequential Number
College_Number	Number	No	FK	College Table	5	College Number
Department_Name	Varchar2	No	-	-	30	Department Name

City Table:

Table4. 44 City Table

Field name	Field type	Null able	Keys	References	Field length	Description
City_Number	Number	No	PK	-	3	Sequential Number
City_Name	Varchar2	No	-	-	15	City Name

Country Table:

Table4. 45 Country Table

Field name	Field type	Null able	Keys	References	Field length	Description
Country_Number	Number	No	PK	-	3	Sequential Number
Country_Name	Varchar2	No	-	-	20	-

College Table:

Table4. 46 College Table

Field name	Field type	Null able	Keys	References	Field length	Description
College_Number	Number	No	PK	-	5	A unique number for the college
College_Name	Vaarchar2	No	-	-	20	College name
College_Capacity	Number	No	-	-	4	College Capacity.

Supervisor Table :

Table4. 47 Supervisor Table

Field name	Field type	Null able	Keys	References	Field length	description
Supervisor_Number	Number	No	PK	-	3	A unique number for the Supervisor
Supervisor_Name	Varchar2	No	-	-	40	Supervisor name

- # Virtual Platform Database:

Which is built on MYSQL database management system.

Database Diagram:

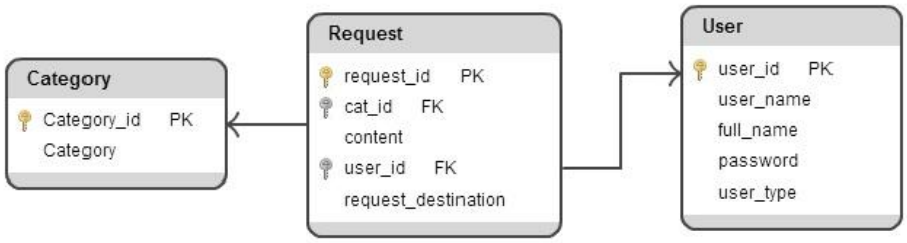


Figure4.7 Database Diagram(VP).

User Table:

Table4. 48 User Table

Field name	Field type	Null able	Keys	References	Field length	description
user_id	Int	No	PK	-	11	sequential number
user_name	Varchar	No	-	-	20	User name to use it for login
full_name	Varchar	No	-	-	30	Full name for the user
Password	Varchar	No	-	-	30	-
user_type	tinyint	No	-	-	4	User type (0 for Admin, 1 for PPU user, 2 for BZU user)

Category Table:

Table4. 25 Category Table

Field name	Field type	Null able	Keys	References	Field length	description
Cat_id	Tinyint	No	PK	-	4	sequential number
Category	Varchar	No	-	-	50	Field express the Category

Request Table:

Table4. 26 Request Table

Field name	Field type	Null able	Keys	References	Field length	description
request_id	int	No	PK	-	11	sequential number
cat_id	tinyint	No	FK	-	4	Reference number to Cat_id
content	Text	No	-	-	-	Content of the Request
user_id	int	No	FK	-	11	Reference number to user_id
request_destination	tinyint	No	-	-	4	Destination to where the request sent (1 to PPU, 2 to BZU)
Request_date	DateTime	No	-	-	-	The request date and time

4.5 User Interface Design:

- Palestine Polytechnic University And Birziet University Interface:
  - ✓ Login Interface: used to log the user into the system as shown in (figure 4.8).

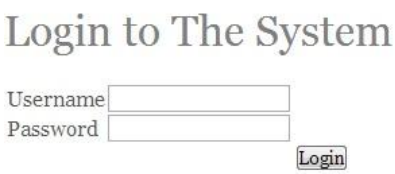


Figure4.8 Login Interface

- ✓ Select if you want to transfer a student or a course and describe the request specifications in the field request content as shown in (figure 4.9).

Figure4.9 Selection the type of the request.

- ✓ Student Information : after selecting transfer a student, the interface in (figure 4.10) will appear, check the information that you need to insert into the table in your database as shown in

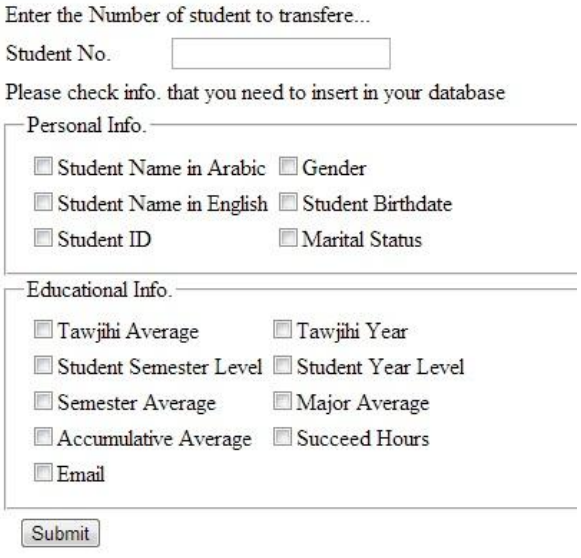


Figure4.10 Student Information

- ✓ Course Information : after selecting transfer a course this interface will appear, check the information that you need to insert into your database as shown in (figure 4.11).

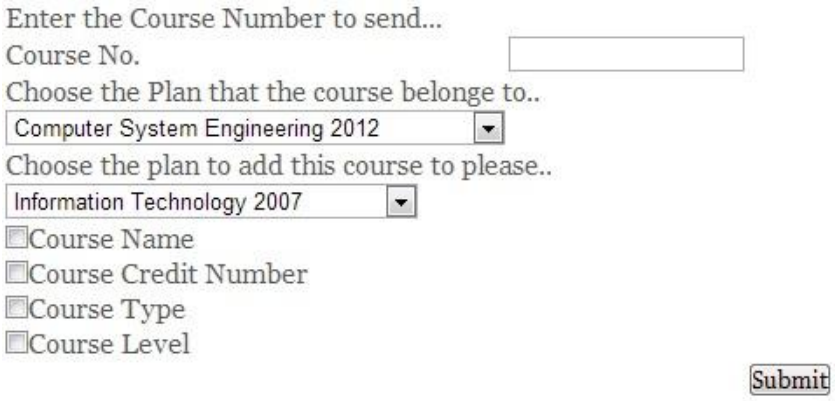


Figure4.11 Course Information

- Virtual Platform Interface:
  - ✓ Login Interface: used to log the VP user into the system as shown in (figure 4.12).

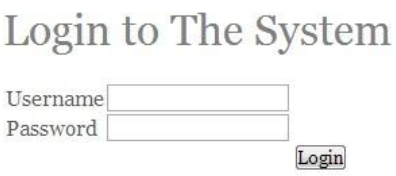


Figure4.12 Login Interface

- ✓ Add new user: (figure 4.13) shows interface of adding new user to the system which is one of the privileges granted to VP user (website Admin).

Figure4.13 Add new user

- ✓ Request for a report: (figure 4.14) shows interface of requesting new report this interface is for VP and the both universities.

Figure4.14 request for a report

4.6 Summary

In this chapter the project team present the design of the system, Architecture Design, Process design, data design and interface design. In the next chapter the project team will discuss the implementation phase, software necessary to implement the system, how to build database, and selected code for necessary functions of the system.

Chapter Five

System Implementation

Content:

5.1 Introduction:

In this chapter the project team will discuss the implementation phase at which more technical details of this project, software necessary to implement the system, how to build database, and selected code for necessary functions of the system.

5.2 Software needed to implement the system:

In this section, the project team will describe programming tools that the project team need through the process of system implementation and development, these include the following:

5.2.1 Microsoft Windows 7 :

This operating system is the most used now, and considered strong and described as high quality and high performance, it supports many applications, requirements needed in this project, and many protection programs where the user can browse through the Internet without fear of access to personal files.

5.2.2 Eclipse for PHP developers:

It is an Integrated Development Environment (IDE), and a general purpose open platform, it Provides tools for coding, building, running and debugging applications. Runs on almost any Web server on several operating systems. it support databases such as MySQL, ODBC, Oracle. Eclipse comes bundled as a zip file, unwrap the zip file to C:\eclipse\ .workspace is a location where the source code store. To begin work you have to :

- **Choose a location where you want to store your project, then click OK**

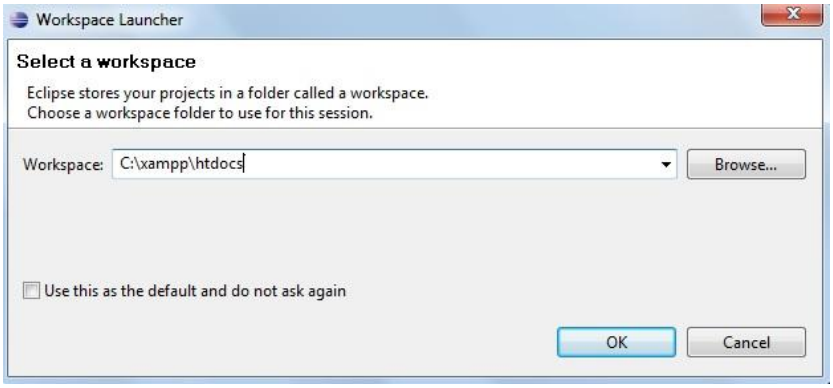


Figure 5.1 workspace selection in Eclipse

- **To create a new project: File → New → PHP Project**

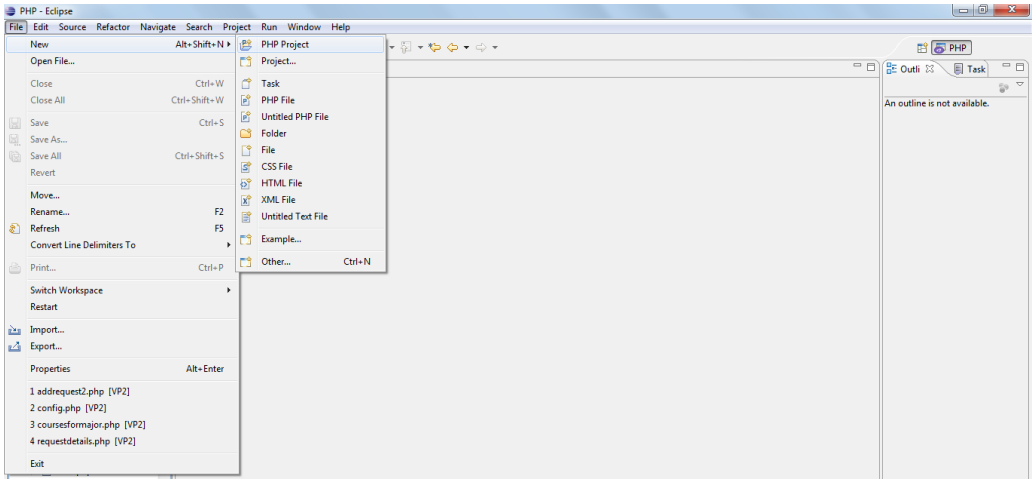


Figure5.2 creation of a new project in Eclipse

5.2.3 XAMPP:

It stands for X (any of four different operating systems(Microsoft Windows, Linux, Sun Solaris and Mac OS)), Apache, MySQL, PHP and Perl. XAMPP is a free, cross-platform web server, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages. To begin work with XAMPP you have to:

- **use the XAMPP Control Panel to start/stop all service:**

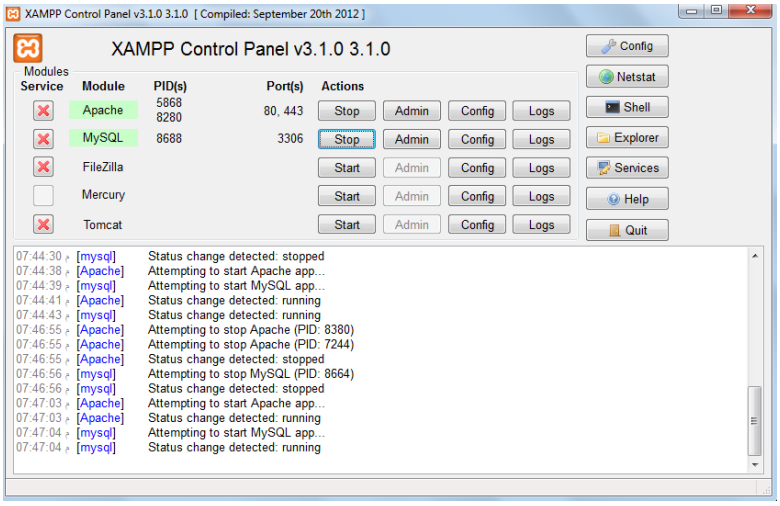


Figure5.3 service activation in XAMPP

- **To build new database choose Admin from XAMPP control Panel then choose PhpMyAdmin:**

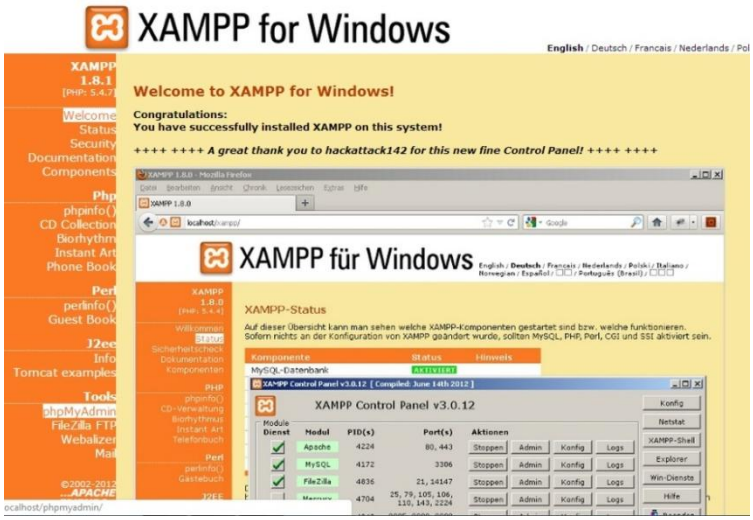


Figure5.4 Building a new database in PHPMyAdmin.

- **To create new database write the name of the database then click on create button:**

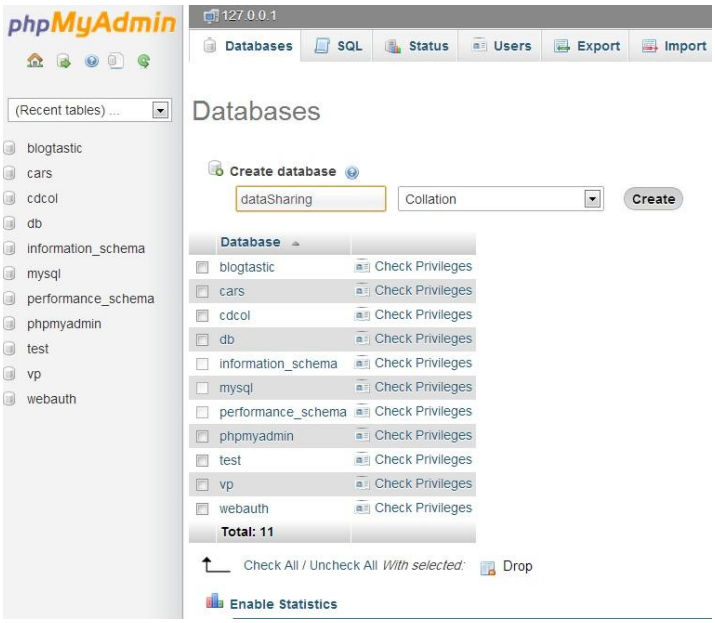


Figure5.5 naming the database

- **The following message will appear after creating the database:**

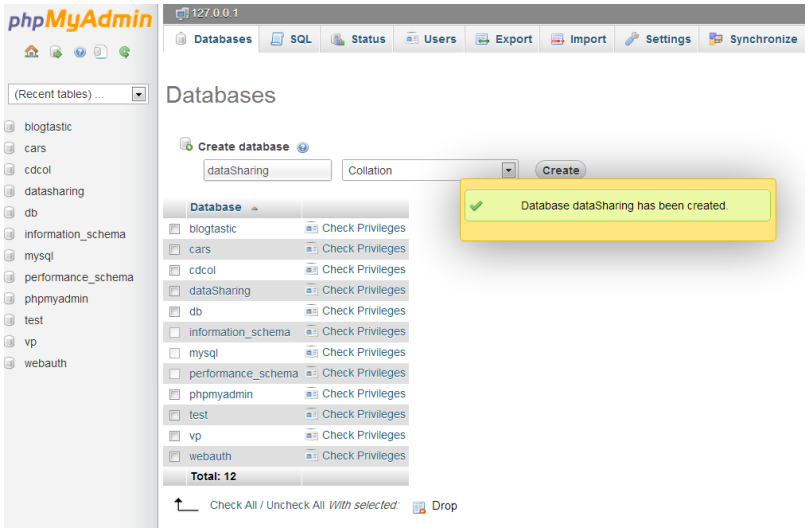


Figure 5.6 notification of database creation

5.2.4 Appserv:

It Stands for application server, it Rapidly create, easy to implement, Support dynamic scripting languages: PHP Hypertext Preprocessor. To begin work with Appserv, you have to:

- **Go to Start → All Programs → Appserv → start/stop all services:**

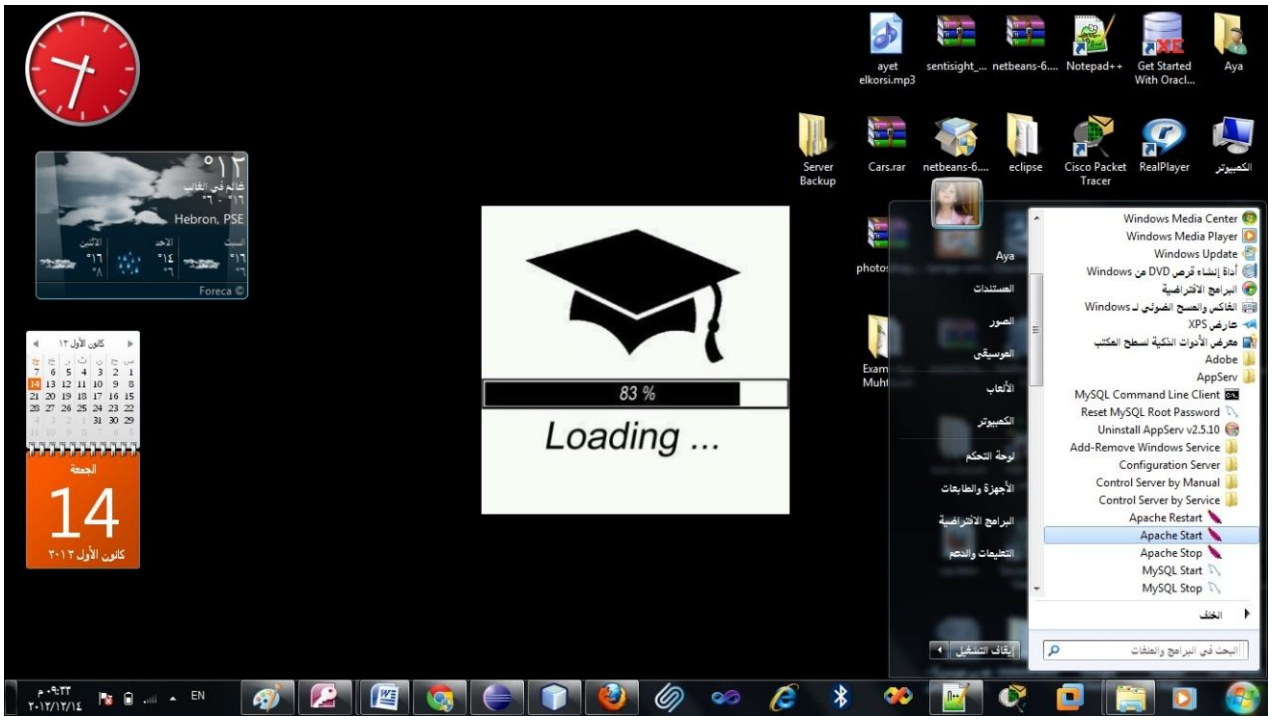


Figure5.7 service activation in Appserv

5.3Establishing Databases Connection

This section clarifies the connection of the Access database with PHP, the project team used Open Database Connectivity(ODBC) the following steps show how to make an ODBC:

- Control Panel -> Administrative Tools-> Data Sources(ODBC) as in (figure 5.8 and figure 5.9)

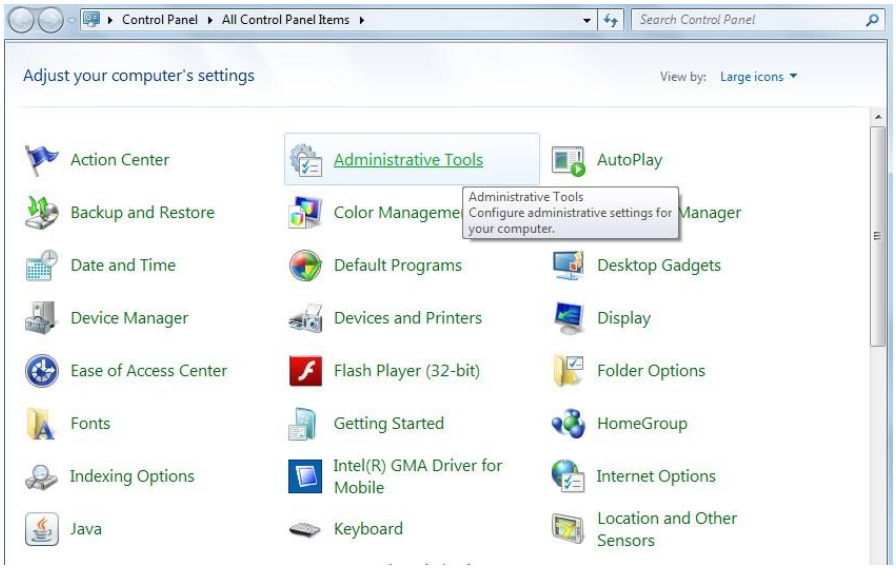


Figure5.8 Control Panel-Administrative Tools

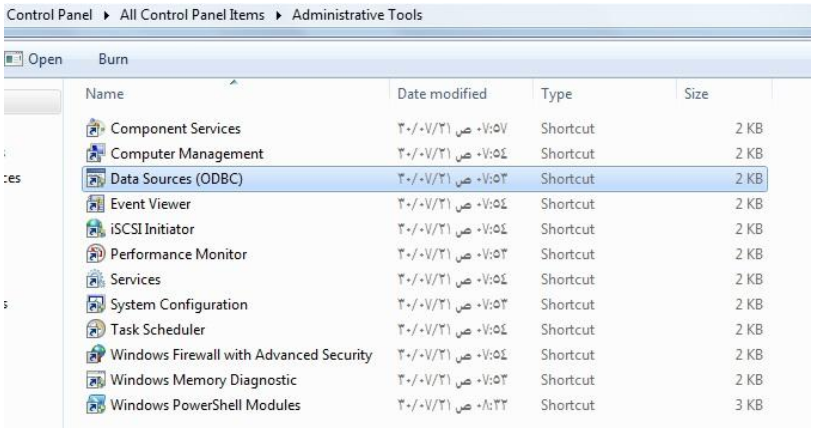


Figure5.9 Data Sources (ODBC)

Then the screen in (figure 5.10) will appear, click on add then a screen to select the type of database will appear, choose Microsoft Access Driver (*.mdb, *.accdb) as in (figure 5.11).

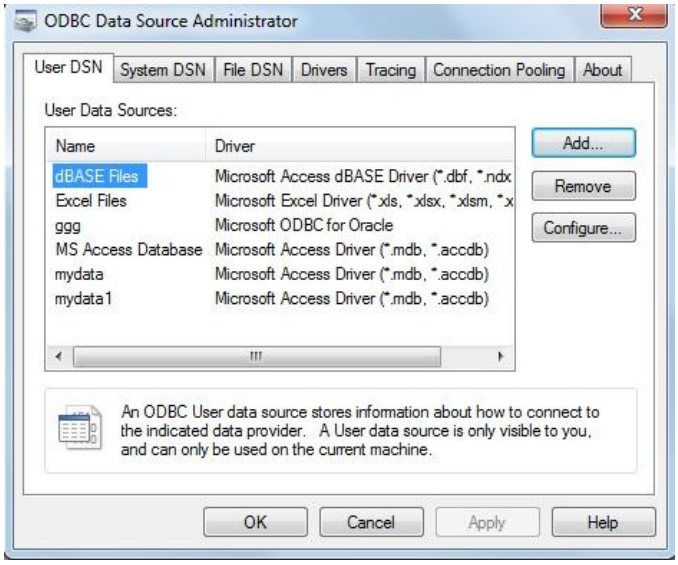


Figure5.10 ODBC Data Source Administrator

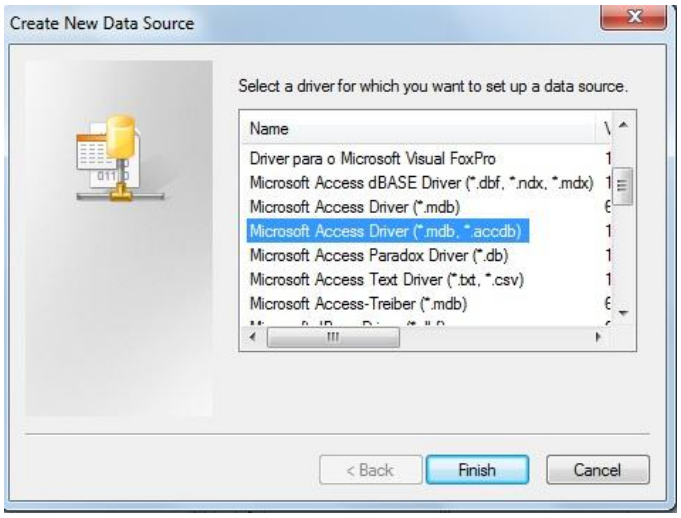


Figure5.11 Create New Data Source

Then name the data source and click on select to select the Access database as in (figure 5.12).

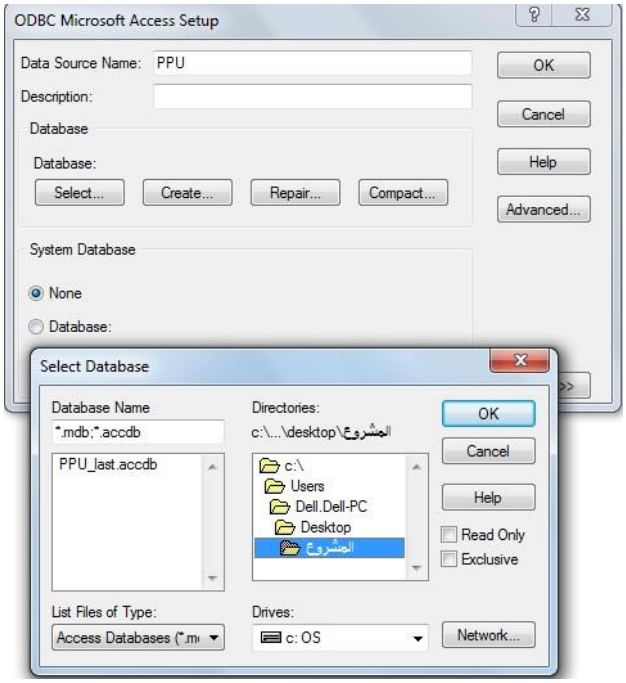


Figure5.12 naming the data source, selecting the Access Database

5.4 Coding For Necessary Functions of the System

5.4.1Access Database Connection:

```
$conn=odbc_connect($ODBCConnection,$username,$password);
$sql="SQL Query;";
$QueryResult =odbc_exec($conn,$sql);
$XMLDoc = ODBC_XML($QueryResult,$RootElement,$ChildElement);
```

The first statement include the function `odbc_connect()` that establish connection with Microsoft Access Database, this function takes three parameters `$ODBCConnection`: ODBC Connection associated with Microsoft Access Database, `$username`, `$password`: username and the password for the access database, the second statement contains the Query for the table or view needed to be converted to XML Document, the third statement executes the SQL Query into the database which we connect with in the first statement, the last statement calls the function `ODBC_XML`( ) that generate XML Document from the database based on the first parameter `$QueryResult` that executed in the third statement, also this function takes another

two parameters: `$RootElement` is the root element for the XML Document and `$ChildElement` for each child element within the root element.

5.4.2Oracle Database Connection:

```
$oconn = oci_connect($username,$password,$databasename 'xe');
$sql = "SQL Query";
$QueryResult = oci_parse($oconn, $sql);
oci_execute($QueryResult);
$XMLDoc = ORACLE_XML($QueryResult,$RootElement,$ChildElement);
oci_free_statement($QueryResult);
```

The first statement include the function `oci_connect()` that establish connection with Oracle Database Management System, this function takes three parameters, `$username, $password:` username and the password for the Oracle Database Management System, the second statement contains the Query for the table or view needed to be converted to XML Document, the third statement prepares an Oracle statement for execution, the forth statement executes the SQL Query into the database which we connect with in the first statement, the fifth statement calls the function `ORACLE_XML` ( ) that generate XML Document from the database based on the first parameter `$QueryResult` that executed in the fourth statement, also this function takes another two parameters: `$RootElement` is the root element for the XML Document and `$ChildElement` for each child element within the root element, the last statement can free the identifiers by setting the variable to `NULL`.

5.4.3 Loading XML file:

```
$url = 'XML Document Path';
$xml = simplexml_load_file($url);
```

These two lines embedded in every PHP file needs to retrieve data from an XML Document and insert into Oracle/ Access Database or presented in report, the first statement contain the path of XML Document, the second statement loads the XML Document from the specified `$url`.

5.4.4 Visualizing XML Document using XSLT:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="RootElement">
  <html>
  <body>
  ...
    <xsl:for-each select="RootElement/ChildElement">
    <xsl:value-of select="SubChildElement"/>
        .
        .
    </xsl:for-each>
  ...
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

Above an XSLT code, which is a declarative programming language that transform an xml document into another preferred xml document. The first statement in the code above is the declaration of XSL document including version and encoding , the second statement shows the root of XSL document which is **<xsl:stylesheet>**. XSL stylesheet is a collection of templates that are applied to source nodes , and each template has a match attribute that specifies to which source nodes the template can be applied as in the fourth statement **<xsl:template match="RootElement">**.

<xsl:for-each select="RootElement/ChildElement"> this statement loops over all the nodes that appears as the value of select attribute, the value of each node can be extracted by **<xsl:value-of select="SubChildElement"/>** it used to extract the value of an element that is selected from source xml document.

The following statement is the link that connect XSL document with XML document, by adding this link the browser can do the transformation.

```
<?xml-stylesheet type=\"text/xsl\" href=\"Student.xsl\"?>
```

5.5 Summary

In this chapter the project team presented the implementation phase at which more technical details of this project, software necessary to implement the system, how to build database, and selected code for necessary functions of the system. The next chapter will present the system testing through its three stages: Unit testing, Integration testing, and System testing.

Chapter Six

System Testing

Content:

6.1 Introduction

After ending the stage of application and programming the system, the system will be put under testing to ensure that it meets the functional requirement, in this chapter the project team will go through the three stages of testing: Unit testing, Integration testing, and System testing.

6.2 Unit Testing:

In this section the project team will divide the system into separate components, and then tests each component separately to ensure that each unit achieves the requirements.

The system include many units test cases, the project team will present two of these cases: Login testing, adding new user to the system.

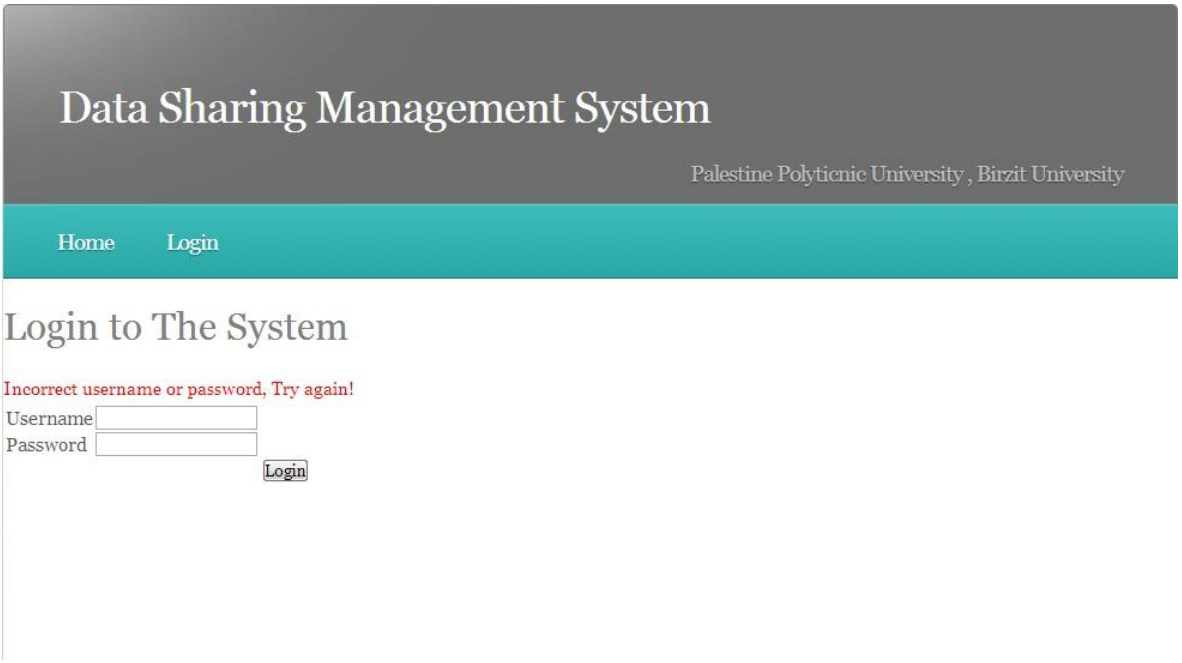


Figure6.1 login with incorrect Password

Figure6.2 add new user successfully

Figure6.3 Miss any of the required fields.

Table6.1 unit testing.

State	Test Case	Expected Result	Actual Result
Login testing	Administrator Login: user name:	Valid login.	Redirect to VP home page.

	vp , password: vp.		
	Administrator Login: user name: vp , password: wrong.	Invalid login.	Redirect to the login page and get warning message, as in (Figure 6.1)
	PPU/BZU Login: User name ppu, password: ppu. Or User name bzu, password: bzu.	Valid login.	Redirect to PPU/ BZU home page.
	PPU/BZU Login: User name ppu, password: wrong. Or User name bzu, password: wrong.	Invalid login.	Redirect to the login page and get warning message, as in (Figure 6.1)
Adding new user to the system	Filling all the required fields.	New user added to the system.	Inform the administrator that a new user has been added, as in figure (6.2).
	Miss any of the required fields.	Adding new user failed.	Redirect to the add new user page and get warning message to fill all the required fields. as in (Figure 6.3)

6.3 Integration Testing:

This stage tests an integral functions as one unit to make sure that individual modules are combined and tested as a group completely compatible and matching the requirements, the project team will present a case which is transferring a student data from BZU to PPU.

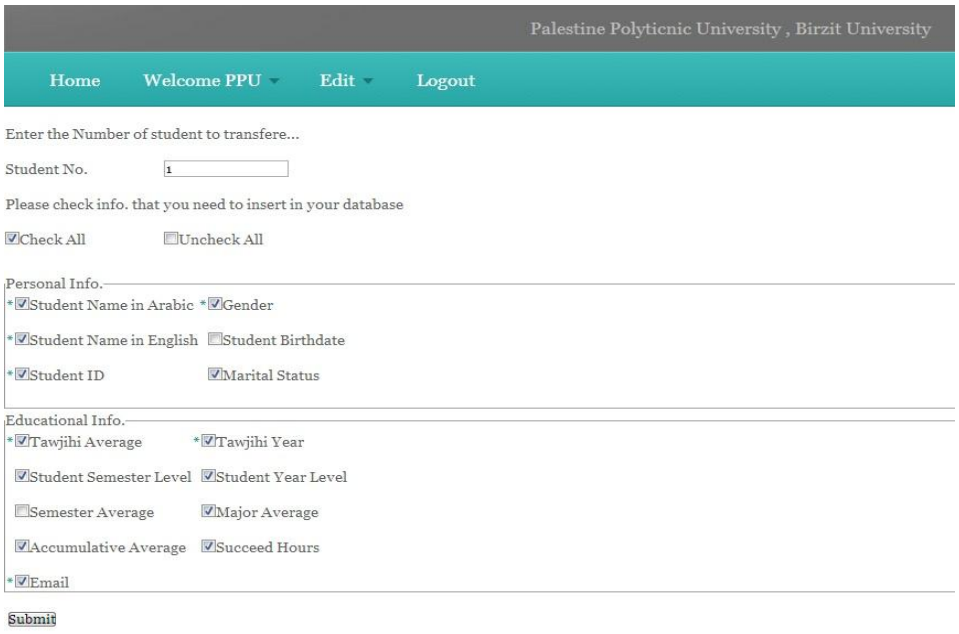


Figure6.4 transfer a student from BZU to PPU.

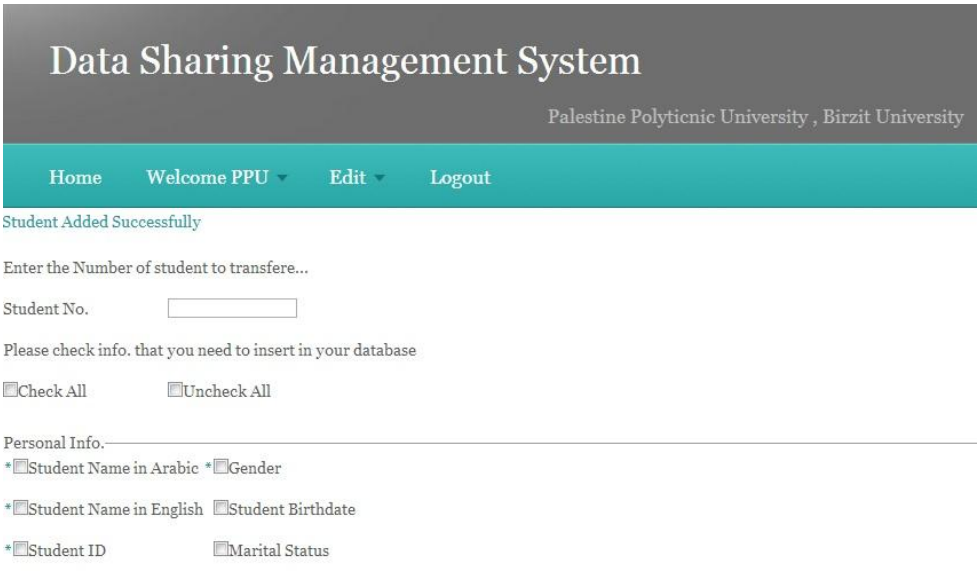


Figure6.5 Student added to PPU database successfully.

Figure6.6 the Student number you enter to transfer does not exist in BZU database.

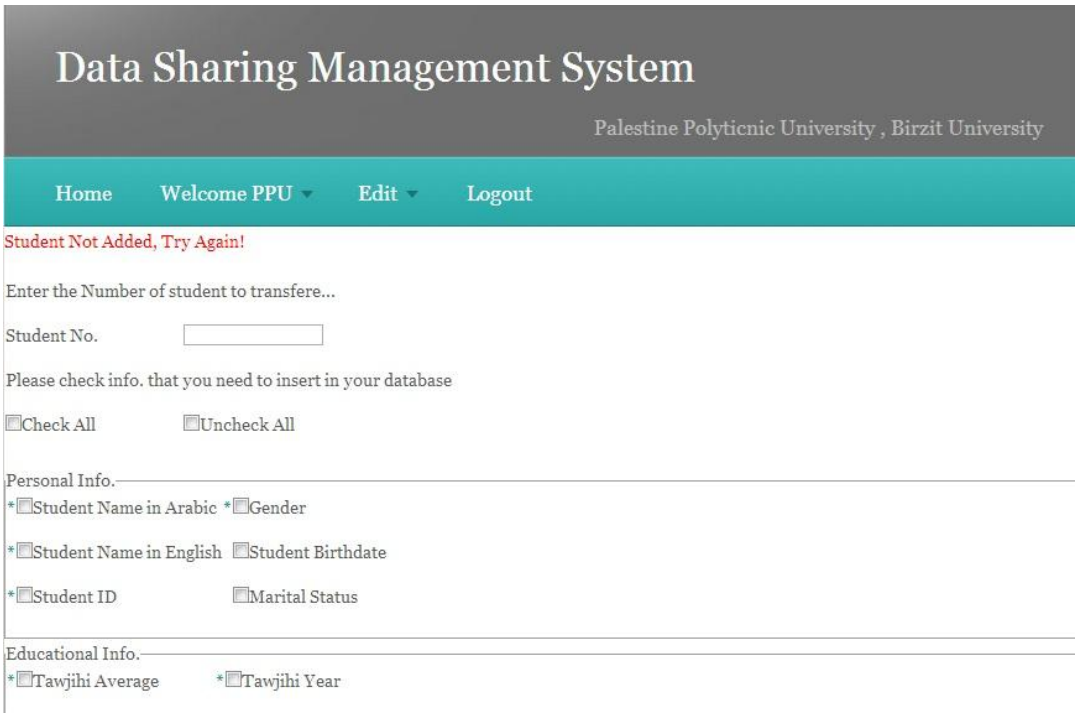


Figure6.7 Student not added to the PPU database, there's something wrong.

6.4 System Testing:

In this section the project team will test the system as one independent unit to ensure that it works correctly and error free, for example if the VP user request for a report that include data from the two universities for example all student in the two university who have accumulated average greater than 89 as shown in (figure 6.8-figure 6.11).

Figure6.8 Request for an Integration Report

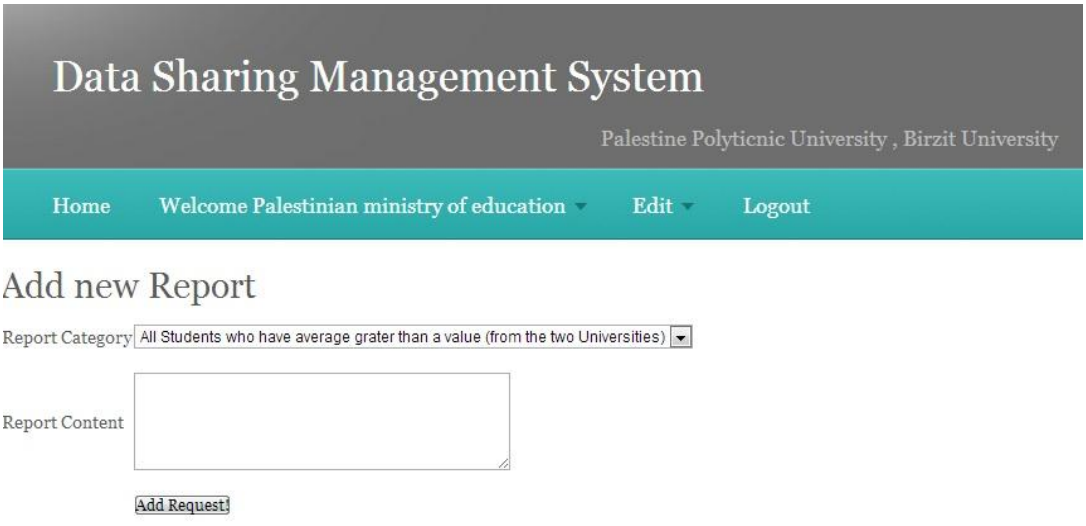
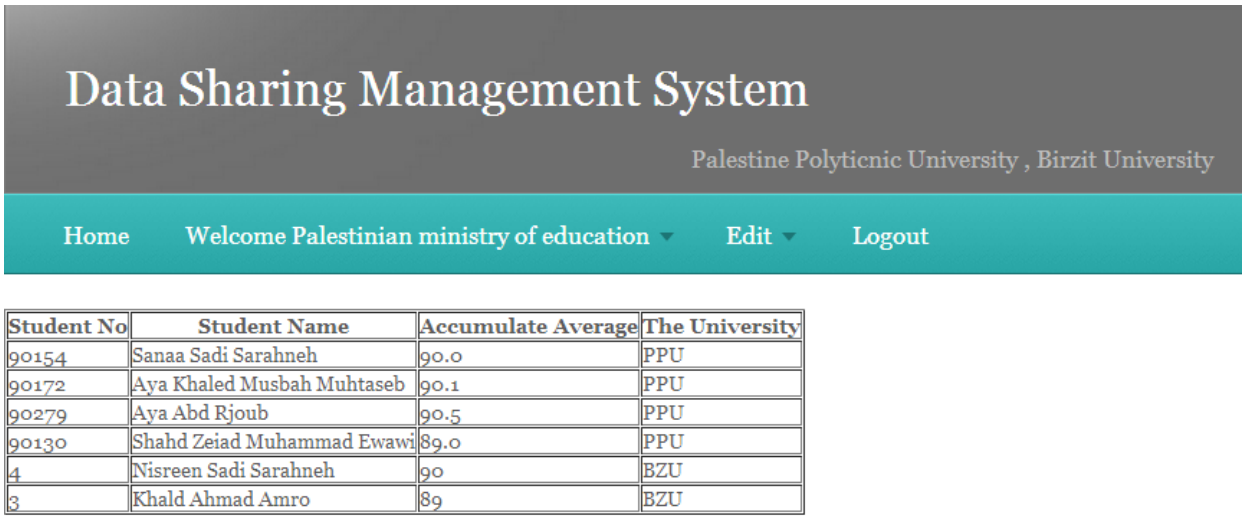


Figure6.9 choosing the report category.

Figure6.10 entering the Accumulated average.



Student No	Student Name	Accumulate Average	The University
90154	Sanaa Sadi Sarahneh	90.0	PPU
90172	Aya Khaled Musbah Muhtaseb	90.1	PPU
90279	Aya Abd Rjoub	90.5	PPU
90130	Shahd Zeiad Muhammad Ewawi	89.0	PPU
4	Nisreen Sadi Sarahneh	90	BZU
3	Khald Ahmad Amro	89	BZU

Figure6.11 The report result

6.5 Summary

In this chapter, the project team have put the system under testing to ensure that it meets the functional requirement. In the next chapter the maintenance stage of the system will be presented.

Chapter Seven

System Maintenance

Content:

7.1 Introduction

The maintenance stage is the final stage of the system life cycle ,in this stage system will put under full maintenance and modification to meet the requirements of the system environment, at this stage system will be ready to work in real environment , to do so and to prevent problems that can be occur during installation many things must be considered.

In the running of system in the real environment , the possibility of system failure or incident errors ,must be managed . Also during running of the system in the real environment users may be unfamiliar or have not enough information or knowledge about the errors that appears in the running mode ,so in this chapter project team will clarify the maintenance process and the proposed plan to maintain the system which include: Backup , System Upgrading , System Upgrading .

7.2 Backup

Any system would have backup plan of the system and it's component, also any modification to the system must be stored on an offline storage. In this process the administrator keep the system itself and its database and any additional component on an offline media, in secure and far place from work environment, backup copies will be used for recovery if any failure occurs, In this project creating a backup copy is administrator responsibility, a backup copy is will be created every month for the system and its database, database backup can be taken through exporting database from the PHPMyAdmin using export tool, system files will be taken by coping the system folder that exist in C:\xampp\htdocs.

7.3 System Upgrading

In the process of using the system a new requirements will be appear, so the upgrading to newer version is necessary, at this stage new requirements will be clarified and work to meet these requirements.

7.4 Apache server Maintenance

Apache server maintenance have many steps including: Monitoring Apache, Log Monitoring, Log Management, Configuration Management, Security and Passwords, and Keeping Apache Up to Date, Scheduling Maintenance steps is important for keeping the apache server in safe mode, this can implemented by applying this steps weekly, monthly, or annually depending on the system criticality.

7.5 Summary

This chapter presented the maintenance stage of the system during system running in the real environment, backup, system upgrading and apache server maintenance.

Bibliography:

[1] GREIF, I., SARIN, S., (1987), "Data Sharing in Group Work", *ACM, 5*, (2):187-211

[2] Mannai, D., Bugrara, K., (1993), "Enhancing Inter-Operability and Data Sharing In Medical Information Systems", *ACM*, 22, (2): 495-498

[3] Rodríguez-Gianolli, P., Garzetti, M., Jiang, L., Kementsietsidis, A., Kiringa, I., Masud, M., Miller, R., Mylopoulos, J., (2005), "Data Sharing in the Hyperion Peer Database System":1291-1294

[4] Hoffer, J., George, J., Valacich, J., (2007), "Modern Systems Analysis and Design", *Pearson*: USA

[5] Geambasu, R., Balazinska, M., Gribble, S., Levy, H., (2007), "HomeViews: Peer-to-Peer Middleware for Personal Data Sharing Applications", *ACM*, 235-246

[6] Green, t., Karvounarakis, G., Taylor, N., Biton, O., lves, Z., Tannen, V., (2007), "Orchestra: facilitating collaborative data sharing", *Department of computer and information science University of Pennsylvania*, 1131-1133

[7] Geambasu, R., Cheung, R., Moshchuk, A., Gribble, S., Levy, H., (2008), "Organizing and Sharing Distributed Personal Web-Service Data", *ACM*: 755-764

[8] Lves, Z., Green, T., Karvounarakis, G., Taylor, N., Tannen, V., Talukdar, P., Jacob, M., Pereira, F., (2008), "The Orchestra Collaborative Data Sharing System", *Department of computer and information science University of Pennsylvania*, 37 ,(3): 26-32

[9] Yu, S., Wang, C., Ren, K., Lou, W., (2010), "Attribute Based Data Sharing with Attribute Revocation", *ACM*: 261-270

[10] Hu, Y., Yang, J., (2011), "A Semantic Privacy-Preserving Model for Data Sharing and Integration", *ACM*

[11] Bakis, N., Aouad, G., Kagioglou, M., (2006), "Towards distributed product data sharing environments — Progress so far and future challenges", *Elsevier* : 586-595

[12] Sarathy, R., Muralidhar, K., (2004), "Secure and useful data sharing", *Elsevier*, 42, (1): 204– 220

[13] Bakis, N., Aouad, G., Kagioglou, M., (2007),"Towards distributed product data sharing environments — Progress so far and future challenges", *Automation in Construction*, 16: 586–595.

[14] Connolly, T., Begg, (1996),"DATABASE SYSTEM", *Addison Wiley*

[15]Harris, D., Khan, L., Paul, R., Thuraisingham, B., (2006), "Standards for secure data sharing across organizations", *Elsevier*.

[16] Nicola Asuni, (2011), TCPDF tool, Retrieved from Website http://www.tcpdf.org/