CHAPTER 1
PROJECT DESCRIPTION

1.1 INTRODUCTION

The best research method, according to Ann Hill Duin in her article on usability, involves methodological triangulation using multiple methods to collect data and arrive at a greater understanding of the topic. Accordingly, I investigated my research questions using a combination of methods drawn from qualitative case study methods and rhetorical analysis. The research for this historical case study of production practices for a particular college yearbook incorporated 1) a rich description of yearbook production from 2003-05, and 2) a rhetorical analysis of the spreads and images of the two college yearbooks produced during that period. The description of production relies on materials and documents used in generating the yearbooks as well as a personal interview with a publishing representative and retrospective description of personal experience. The rhetorical analysis of yearbook spreads and images is of two yearbooks from 2004 and 2005, applying the concepts of Kress and van Leuven. The design and implementation of a comprehensive Online College Management System and user interface is to replace the current paper records. College Staff are able to directly access all aspects of a student's academic progress through a secure, online interface embedded in the college's website. The system utilizes user authentication, displaying only information necessary for an individual's duties. Additionally, each sub-system has authentication allowing authorized users to create or update information in that subsystem. All data is thoroughly reviewed and validated on the server before actual record alteration occurs. In addition to a staff user interface, the system plans for student user interface, allowing users to access information and submit requests online thus reducing processing time. All data is stored securely on SQL servers managed by the college administrator and ensures highest possible level of security. The system features a complex logging system to track all users' access and ensure conformity to data acc ess guidelines and is expected to increase the efficiency of the college's record management thereby decreasing the work hours needed to access and deliver student records to users. Yearbooks are an important part of an individual's history. In the United States, high school and college yearbooks are extraordinarily well known as a genre, yet the publication of yearbooks is an area largely unstudied by researchers. Students are introduced to yearbooks at an early age. In elementary and middle school students learn what a yearbook is, and at many schools between

60-80 percent of students purchase a yearbook (Walsh “Preserving Student Memories a $500 Million Industry”). In high school, there is an entire day devoted to yearbook distribution. Students crowd in line to get their copy of the yearbook and find their picture in it. Then, they rush to get all their friends and acquaintances to sign their yearbook—personalizing it and creating a lasting memory of their school year.

1.2 EXISTING SYSTEM

.3.1 HARDWARE SPECIFICATION

Processor	: Dual core processor 2.6.0 GHz
RAM	: 1GB
Hard disk	: 160 GB
Compact Disk	: 650 MB
Keyboard	: Standard keyboard
Monitor	: 15 inch color monitor

1.3.2 SOFTWARE SPECIFICATION

Front End	: PHP
IDE	: dream weaver
Back End	: My SQL

1.3.3 PHP

PHP is server side back end programming language. It executes in server along with maximum all available web servers like Apache, IIS (Internet Information Server) etc.., and return the response as required MIME type. It is a Pre Process Hypertext, we could do many things on server by using PHP on server and co-ordinate with DB server for CURD (Create, Update, Read, and Delete) actions. Front end in the seance, UI which intact the users, it can done by HTML, or any others. And UI Behavior is defined in UI back end Languages (Scripting languages) via: Java script, VB script

PHP started out as a small open source project that evolved as more and more people found out how useful it was. RasmusLerdorf unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".

- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.

- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the UNIX side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.

- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.

- PHP is forgiving: PHP language tries to be as forgiving as possible.

- PHP Syntax is C-Like.

Common Uses of PHP

PHP performs system functions, i.e. from files on a system it can create, open, read, write and close them. The other uses of PHP are:

PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user. You add, delete and modify elements within your database thru PHP. Access cookies variables and set cookies. Using PHP, you can restrict users to access some pages of your website. It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible:

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

PHP Variables

The main way to store information in the middle of a PHP program is by using a variable. Here are the most important things to know about variables in PHP.

- All variables in PHP are denoted with a leading dollar sign ($).
- The value of a variable is the value of its most recent assignment.

- Variables are assigned with the = operator, with the variable on the left-hand side and the expression to be evaluated on the right.
- Variables can, but do not need, to be declared before assignment.
- Variables in PHP do not have intrinsic types - a variable does not know in advance whether it will be used to store a number or a string of characters.
- Variables used before they are assigned have default values.
- PHP does a good job of automatically converting types from one to another when necessary.

PHP variables are Perl-like. PHP has a total of eight data types which we use to construct our variables:

- **Integers:** are whole numbers, without a decimal point, like 4195.
- **Doubles:** are floating-point numbers, like 3.14159 or 49.1.
- **Booleans:** have only two possible values either true or false.
- **NULL:** is a special type that only has one value: NULL.
- **Strings:** are sequences of characters, like 'PHP supports string operations.'
- **Arrays:** are named and indexed collections of other values.
- **Objects:** are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- **Resources:** are special variables that hold references to resources external to PHP (such as database connections).

Back End (MySQL)

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, Word Press, phpBB, MyBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google (though not for searches), ImagebookTwitter, Flickr, Nokia.com, and YouTube.

Inter images

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

Graphical

The official MySQL Workbench is a free integrated environment developed by MySQL AB, which enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

MySQL ships with some command line tools. Third-parties have also developed tools to manage a MySQL server, some listed below. Maatkit - a cross-platform toolkit for MySQL, PostgreSQL and Memcached, developed in Perl Maatkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up servers. Maatkit is included with several GNU/Linux distributions such as CentOS and Debian and packages are available for Programming. MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Solaris, Symbian, SunOS, SCO Open Server, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists. MySQL is written in C and C++. Its SQL parser is written in yacc, and a home-brewed lexical analyzer. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBCinterimage called MyODBC allows additional programming languages that support the ODBC inter image to communicate with a MySQL database, such as ASP or ColdFusion. The HTSQL - URL-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

Features

As of April 2009, MySQL offered MySQL 5.1 in two different variants: the open source MySQL Community Server and the commercial Enterprise Server. MySQL 5.5 is offered under the same licenses. They have a common code base and include the following features:

A broad subset of ANSI SQL 99, as well as extensions

- Cross-platform support
- Stored procedures
- Triggers
- Cursors
- Updatable Views
- Information schema

Strict mode (ensures MySQL does not truncate or otherwise modify data to conform to an underlying data type, when an incompatible value is inserted into that type)

X/Open XAdistributed transaction processing (DTP) support; two phase commit as part of this, using Oracle's InnoDB engine

- Transactions with the InnoDB, and Cluster storage engines
- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Replication support (i.e. Master-Master Replication & Master-Slave Replication)
- Embedded database library
- Partitioned tables with pruning of partitions in optimizer
- Shared-nothing clustering through MySQL Cluster
- Hot backup (via `mysqlhotcopy`) under certain conditions

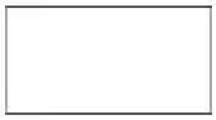
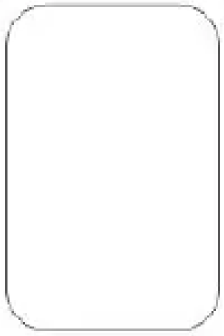
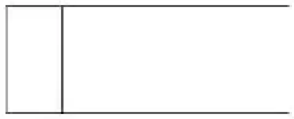
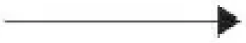
Multiple storage engines, allowing one to choose the one that is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at run time): Native storage engines (MyISAM, Falcon, Merge, Memory (heap), Federated, Archive, CSV, Black hole, Cluster, EXAMPLE, Maria, and InnoDB, which was made the default as of 5.5). Partner-developed storage engines (solidDB, NitroEDB, ScaleDB, TokuDB, Infobright (formerly Brighthouse), Kickfire, XtraDB, IBM DB2). InnoDB used to be a partner-developed storage engine, but with recent acquisitions, Oracle now owns both MySQL core and InnoDB.

CHAPTER 2
LOGICAL DEVELOPMENT

2.1 DATA FLOW DIAGRAM

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

Data flow Symbols:

Symbol	**Description**
	An **entity**. A source of data or a destination for data.
	A **process** or task that is performed by the system.
	A **data store**, a place where data is held between processes.
	A **data flow**.

LEVEL 0

DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers. A context diagram gives an overview and it is the highest level in a data flow diagram, containing only one process representing the entire system. It should be split into major processes which give greater detail and each major process may further split to give more detail. Level 0 DFD must balance with the context diagram it describes. Input going into a process is different from outputs leaving the process. Data stores are first shown at this level.

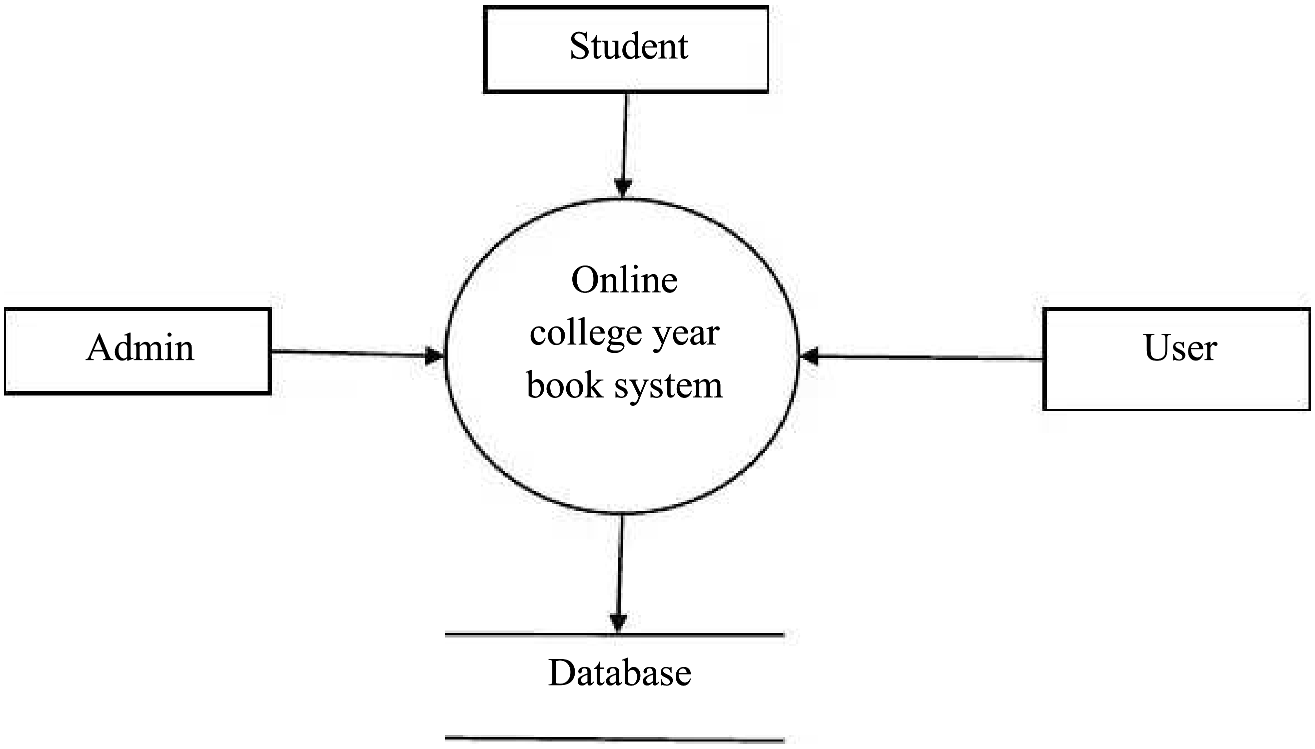


Fig 2.1.1 level 0-DFD

LEVEL 1

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub – processes. Level 1 - interaction between 2 different business applications. This is primarily used to explain the process to business and tech leads, QA leads. As described previously, context diagrams (level 0 DFDs) are diagrams where the whole system is represented as a single process. A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an “exploded view” of the context diagram.

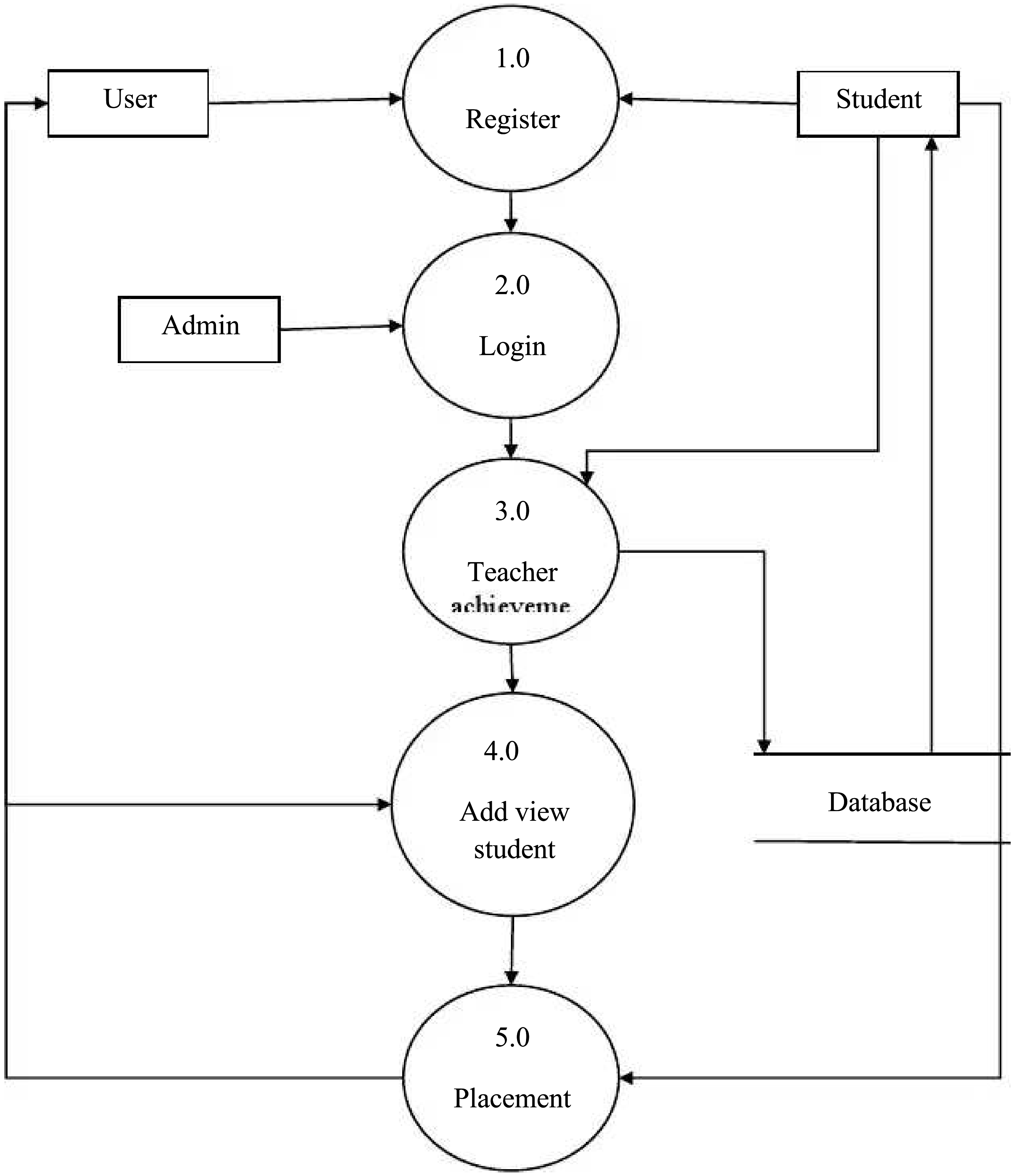


Fig 2.2.1 level 1 DFD

2.2 ARCHITECTURAL DIAGRAM

The term architecture is used to describe physical structures, such as buildings, however it also describes the technicalities of designing those buildings, as well as the method of construction. Studying an architecture degree is challenging but rewarding and can take between three and seven years to complete. Architectural education in Australia varies depending on the university offering the course. All Australian architecture schools and programs have recently moved to some variation on the 3 + 2 Bologna model, with a three-year undergraduate degree, followed by a two-year course-work based Master of Architecture professional degree. Some universities also require 6–12 months of practice work experience between the two degrees. Registration as an architect is via state-based boards and requires a minimum of two years practice experience under the supervision of a registered architect.

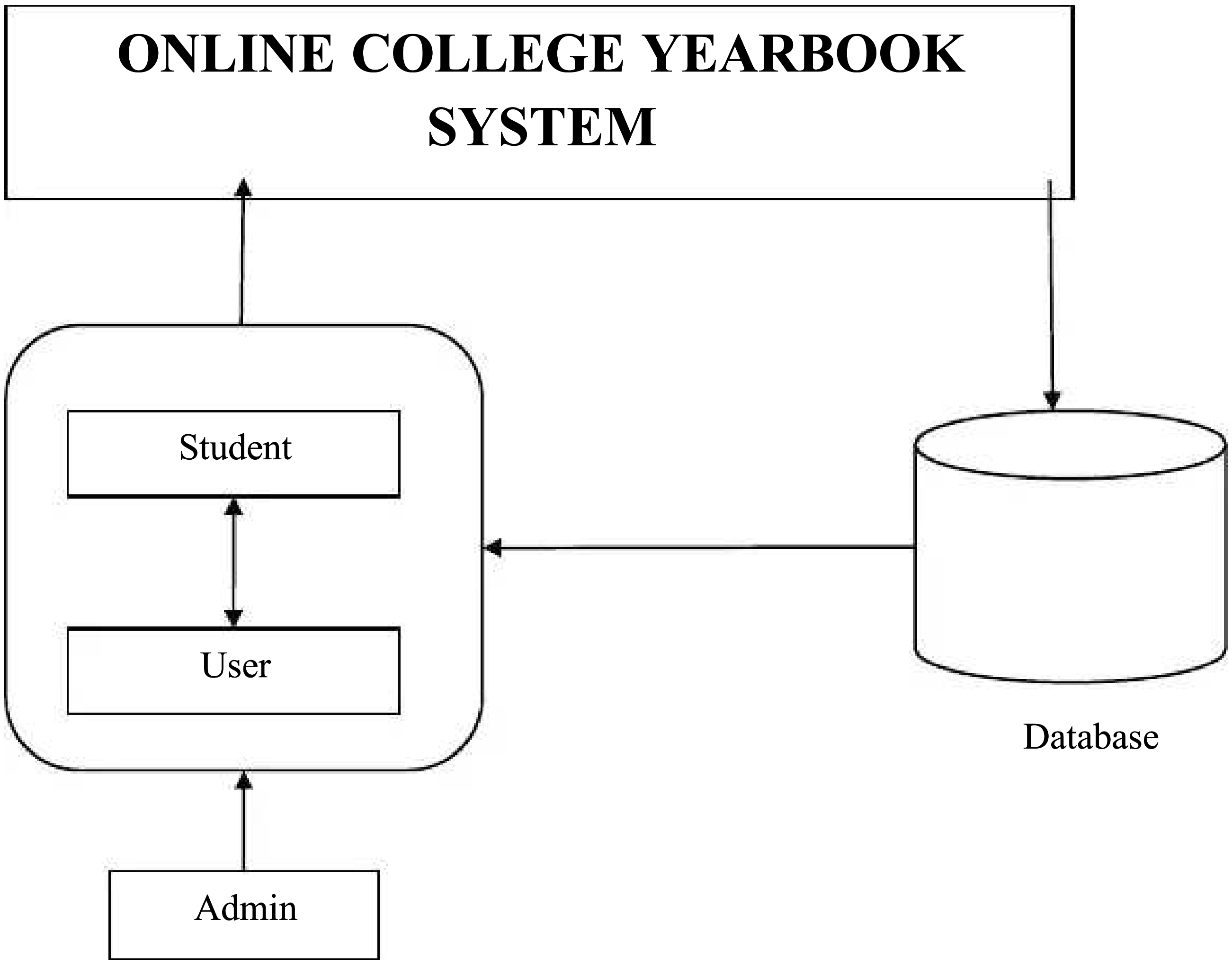


Fig 2.2.1system architecture

CHAPTER 3
DATABASE DESIGN

3.1 DATA DICTIONARY

3.3 RELATIONSHIP DIAGRAM

Entity Relationship Diagram, also known as ERD, ER Diagram or ER model, is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information: **The major entities within the system scope**, and the **inter-relationships among these entities**.And that's why it's called "Entity" "Relationship" diagram (ERD)!When we talk about entities in ERD, very often we are referring to business objects such as people/role (e.g. Student), tangible business objects (e.g. Product), intangible business objects (e.g. Log), etc. "Relationship" is about how these entities relate to each other within the system.

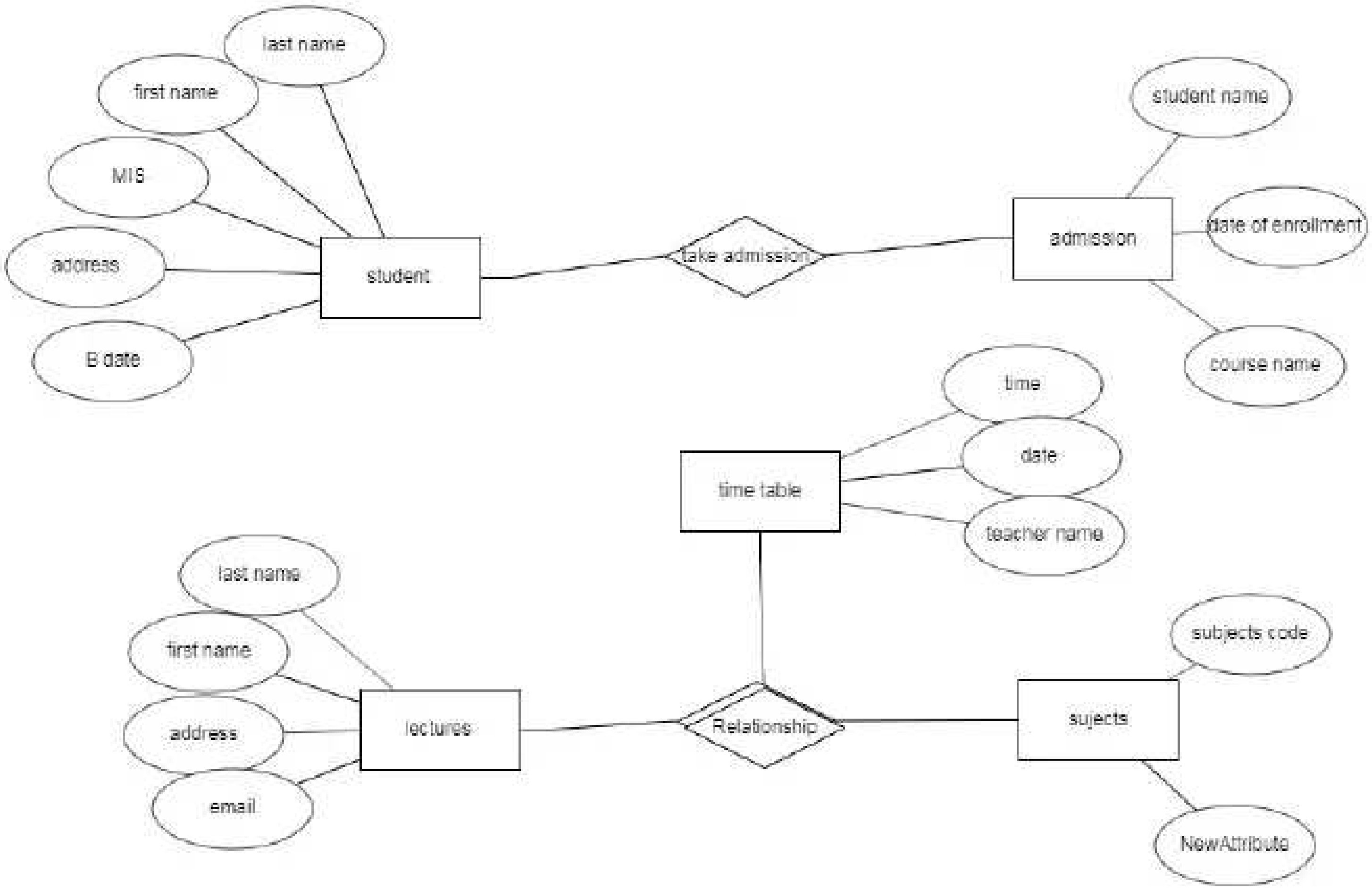


Fig 3.3.1 ER diagram

CHAPTER 4
PROGRAM DESIGN

4.1 MODULES

- **ADMIN MODULE**
- **REGISTER**
- **USER MODULE**

4.1.2ADMIN MODULE

The administrator is responsible for entering the new student and managing the student Accounts. The administrator also manages the faulty accounts like entering a new faculty assigning the faculty to the subjects. The Administrator also updates the college related information about events that occur in the college. The Administrator also updates placement related information and manages E-Library. The administrator will check all the updates i.e. student updates faculty, updates, exam updates etc. The administrator has the highest level of power in the college management system.

4.1.3 REGISTER

- **Registration:** the college will register their details which are needed by the company. When registering their details the company will enter each and every individual details about the company
- **Login**: college will be provided with a login session. The college will login with their username and password. This is provided for a login session where the company will login and view the details of the user or else add they details when they recruit the person when they are in need.
- **Add details:** the college year will add the details they have to hire for their company to recruit. They will add the designation, qualification and other details which are needed for the user to attend the interview.

4.1.4 USER MODULE

The student is of center focus, because in every college student plays the very important role. Student can access the information of the college, subject details, training and placement cell information and exam section information. The course details include information regarding branch he is studying, the academic curriculum of the college, year wise subject offered by the branch, the subject details include the syllabus of the subjects, information regarding the staff handling the subjects, the subjects he presently registered for the semester he is presently studying, attendance and internal marks of the subjects, he can also ask any queries to the staff

USER MODULE:

- **Registration:** The user has to register into the system providing his personal details. The user will register their details entering all details which the user can register the entire details.
- **Login:** The user has to login into the add their profile and the certificates to the system database. After login completed the user will view the post which are added by the company profile. The user will like the company details which they wish to like.
- **Upload document:** user will upload their college documents and the resume will be uploaded. After uploading the documents the company will view their resume and marks percentage in it. If they wish to hire the person the company will invite the person to attend their interview.

CHAPTER 5

SYSTEM TESTING

5.1 Testing

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system. The philosophy behind testing is to find the errors. A good test is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input.

5.1.1 Types of Testing:

➢ **System testing**

After a system has been verified, it needs to be thoroughly tested to ensure that every component of the system is performing in accordance with the specific requirements and that it is operating as it should including when the wrong functions are requested or the wrong data is introduced.

Testing measures consist of developing a set of test criteria either for the entire system or for specific hardware, software and communications components. For an important and sensitive system such as an electronic voting system, a structured system testing program may be established to ensure that all aspects of the system are thoroughly tested.

Testing measures that could be followed include:

- Applying functional tests to determine whether the test criteria have been met
- Applying qualitative assessments to determine whether the test criteria have been met.

- Conducting tests in “laboratory” conditions and conducting tests in a variety of “real life” conditions.
- Conducting tests over an extended period of time to ensure systems can perform consistently.
- Conducting “load tests”, simulating as close as possible likely conditions while using or exceeding the amounts of data that can be expected to be handled in an actual situation.

Test measures for hardware may include:

- Applying “non-operating” tests to ensure that equipment can stand up to expected levels of physical handling.
- Testing “hard wired” code in hardware (firmware) to ensure its logical correctness and that appropriate standards are followed.

Tests for software components also include:

▪ Testing all programs to ensure its logical correctness and that appropriate design, development and implementation standards have been followed.

▪ Conducting “load tests”, simulating as close as possible a variety of “real life” conditions using or exceeding the amounts of data that could be expected in an actual situation.

- Verifying that integrity of data is maintained throughout its required manipulation.

➢ Unit testing

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed.

Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results. Functional and reliability testing in an Engineering environment. Producing tests for the behavior of components (nodes and vertices) of a product to ensure their correct behavior prior to system integration.

- **System testing**

Several modules constitute a project. If the project is long-term project, several developers write the modules. Once all the modules are integrated, several errors may arise. The testing done at this stage is called system test. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points. Testing a specific hardware/software installation. This is typically performed on a COTS (commercial off the shelf) system or any other system comprised of disparate parts where custom configurations and/or unique installations are the norm.

- **Integration testing**

Testing is which modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is release.

CHAPTER 6

CONCLUSION