**Chapter Three**

**PROPOSED UMIS USING SOA**

1. **Introduction**

LMS can not achieve UMIS functionalities, and vice versa. University needs both LMS and UMIS, so, proposed service based architecture can not ignore neither of them. One of thesis’s objectives is to present major components of a service based UMIS and LMS. This chapter will present proposed architecture of selected UMIS components based on both the surveys, and university requirements.

1. **Proposed Architecture Characteristics**

**2.1 Layered Architecture**

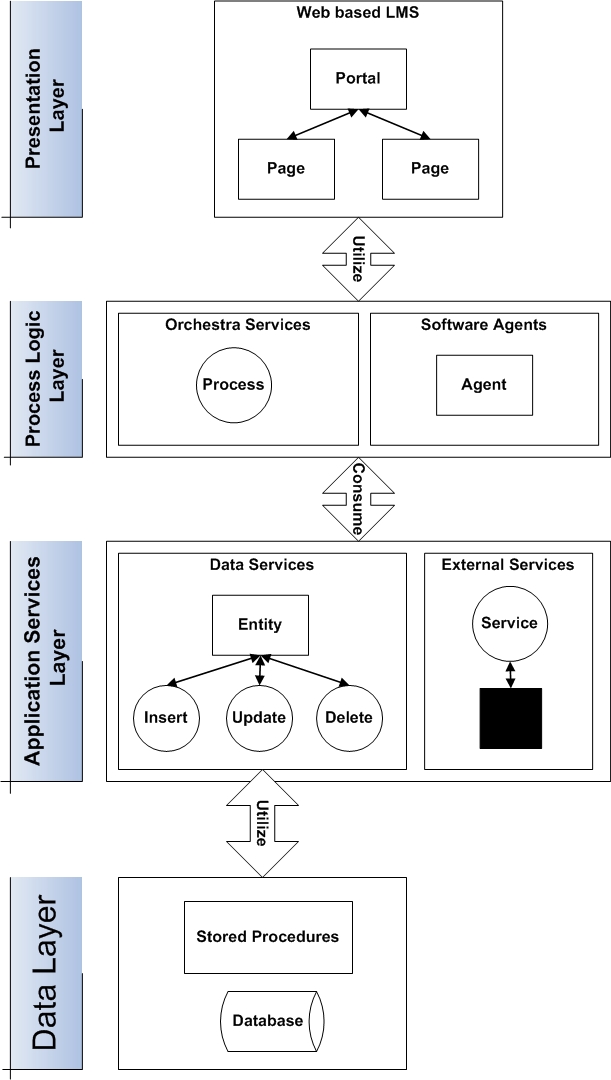
Figure 3.1 depicts guidelines of proposed service based architecture which is divided into four layers: data, application services, process logic, and interface layer. Interface layer utilizes process logic layer, which consumes application services layer that utilizes data layer.

**2.1.1 Data Layer**

Implementation architecture consumes one Microsoft SQL Server 2005 database server that holds the data files and relevant stored procedures that reside beside data files on the same database server. Usage of stored procedures to act as a data layer provides many advantages on regarding performance and security. Though usage of stored procedures might be an overload on the database server, caching is the solution that enhances performance. From a security perspective, it is not a recommended practice to implicitly implement SQL statements in the Web services. Beside, stored procedures allow reusability on data level.

**2.1.2 Application Services Layer**

A Microsoft Windows 2003 Server that has .net framework 2.0 installed satisfies the software requirements. Application services layer contains data services, which are Web services that directly utilize stored procedures for data access. This layer holds external consumed services. Internet is the connection medium between LMS and external services. Proposed LMS implement direct connection between services.



**Figure 3.1: Proposed Learning Management System Layered Architecture**

**2.1.3 Process Logic Layer**

Process logic layer holds orchestra Web services and software agents that reflect processes supported by the LMS. Orchestras and software agents consume application services layer and external Web services. Orchestras are implemented as Web services, and software agents are implemented as windows services that is always running and executing.

**2.1.4 Presentation Layer**

Portals are presented for human interaction with LMS. As depicted separately later in interface design section, portals present the capability to perform main functionalities in easy direct manner that reflects LMS services composability.

**2.2 Development Framework**

.Net framework is an emerging Microsoft technology that enables execution of any .net compiled code on any machine that has .net framework installed. Machines include: PDA; personal digital assistants, Tablet PCs, Smart Phones, and Computers. .Net framework is not a software developing tool, neither a programming language. .Net framework supports more than 30 programming languages, including Delphi, Visual Java, and Visual C++. Choosing .net framework was affected by two other factors: efficiency of Microsoft SQL Server 2005, and the seamless integration between SQL Server and .net framework based programming languages. .Net framework deals with standards efficiently.

1. **UMIS Proposed Components**

Selected UMIS components to be analyzed, designed, implemented, and evaluated are: Student Information System (SIS), Library Information System (LIS), and Faculty Information System. SIS will hold students data, registration, and enrollment details. LIS will present the automation of the library traditional system without the availability of digital contents. Faculty Information System is the information system that holds faculties, and staff members’ data.

1. **Faculty Information System**

Faculty Information System can be thought of as a collection of information systems that manage entities depicted in figure 3.2. Instructors and Employees can be further analyzed into categories as presented in table 3.1. Automating and managing the different activities of mentioned departments is a must for integrating different components of faculty within learning activities.

Figure 3.2: Faculty Information System Entities

Table 3.1: Instructors and Employees Categories

|  |  |
| --- | --- |
| Instructors | Employees |
| Dean | e-Learning Center |
| Vice Dean | Youth Activities |
| Heads of Departments | Dean Secretary |
| Doctors | Department Secretary |
| Assistant Teachers | Legal Issues Department |
| Teaching Assistants | Head of Employees Secretary |
|  | Stuff Members Secretary |
|  | Financial Department |
|  | Head of Employees |
|  | Import and Export |
|  | Post-Graduate Studies |
|  | Student Affairs |
|  | Inventory |
|  | Employee Affairs |
|  | Library |

1. **Student Affairs Management System (SIS)**

SIS is the information system responsible for managing student data and all student related activities. Student Affairs is one of the departments that exist in all faculties in all universities. Student Information System reflects the processes initiated, managed, and maintained at that department.

* 1. **Analysis of SIS**

Figure 3.3 presents the main activities at Student Affairs Department.

**Figure 3.3: SIS Use Case**

There are eight major processes at student affairs department:

* **Registration:** Students register in the faculty by fulfilling the required forms.
* **Join University Residence:** Students can join the university residence if they meet the predetermined requirements.
* **Recruiting:** Student Affairs is responsible for managing the army forces papers of student till they reach 28 years old or finish studying.
* **Time Tabling:** Student Affairs manages the term time table trying to satisfy almost all stuff members and students requirements.
* **Departure:** Students can submit a departure form incase they want to join another Faculty / University. This activity requires integration among different Student Affairs Management Systems to make this process globally available.
* **Tracking Students Attendance:** in practical faculties, students must attend lectures and labs to achieve educational goals. Student Affairs must keep track of student attendance and present attending report status available upon request by dean and / or instructor. Decisions regarding the way to punish the student that exceeds the maximum limit of absence are taken by dean and / or instructor.
* **Managing Exams:** involve the routine activities of preparing the disks, seat numbers, matching unique student IDs with seat numbers, preparing the answer sheets, announcing the exam schedule, delivering the seat numbers, and arranging the exam benches.
* **Preparing Forms:** Instructors and Students can request several different kinds of forms available at every Student Affairs department. Forms need to be automated, and composing data need to be stored in the database.

**Registration Analysis and Join University Residence**

Analysis of both processes shows that they share the same steps so; they are abstracted to registration operation as depicted in figure 3.4.



Figure 3.4: Registration Analysis

**Recruiting**

Recruiting is one of the processes that can be split into two others: the first one is invoked by Student, and ends where the Student Affairs Employee starts the second one; the process that is achieved by him. Figure 3.5 presents the Student-Recruiting process, and figure 3.6 shows the Student Affairs Employee-Recruiting one.



Figure 3.5: Student – Recruiting Analysis



Figure 3.6: Student Affairs Employee-Recruiting

**Time Tabling**

Time Tabling process is initiated by any Student Affairs Employee and it is the process that tends to make the faculty time table ready. It consumes the activities that are depicted in figure 3.7.



Figure 3.7: Time-Tabling Analysis

**Departure**

Students can sign out from the faculty and head to another faculty. According to the status of the student, the place s/he is heading to, different situations can take place. Figure 3.8 shows the analysis of the student departure process.



Figure 3.8: Departure Analysis

**Tracking Student Attendance**

Faculties should keep track of student attendance in order to guarantee a healthy educational process. Tracking Student Attendance can be done by Student Affair Employee as shown in figure 3.9.



Figure 3.9: Track Student Attendance Analysis

**Managing Exams**

From managerial point of view, exam involves activities that should be good taken care of as depicted in figure 3.10.



Figure 3.10: Manage Exam Analysis

**Preparing Different Forms**

Students and / or Instructors can request certain forms to be able to present them to other organizations. Student Affairs Employee to prepare the form as shown in figure 3.11.



Figure 3.11: Prepare Different Forms

* 1. **Design of SIS**

Figure 3.12 presents the required Student Affairs entities based on analyzed presented system processes. Figure 3.13 depicts the Entity-Relationship (E-R) diagram of the system.

* + 1. **Proposed SIS Architecture**

Figure 3.14 shows the proposed Services Based SIS on two parts, part (a) presents services related to employee, student, CV, instructor, and course managers. Part (b) addresses certification, e, reference, department, faculty, and questions managers. Entity Centric based Services are designed to reflect system entities.

Figure 3.12: SIS Entities

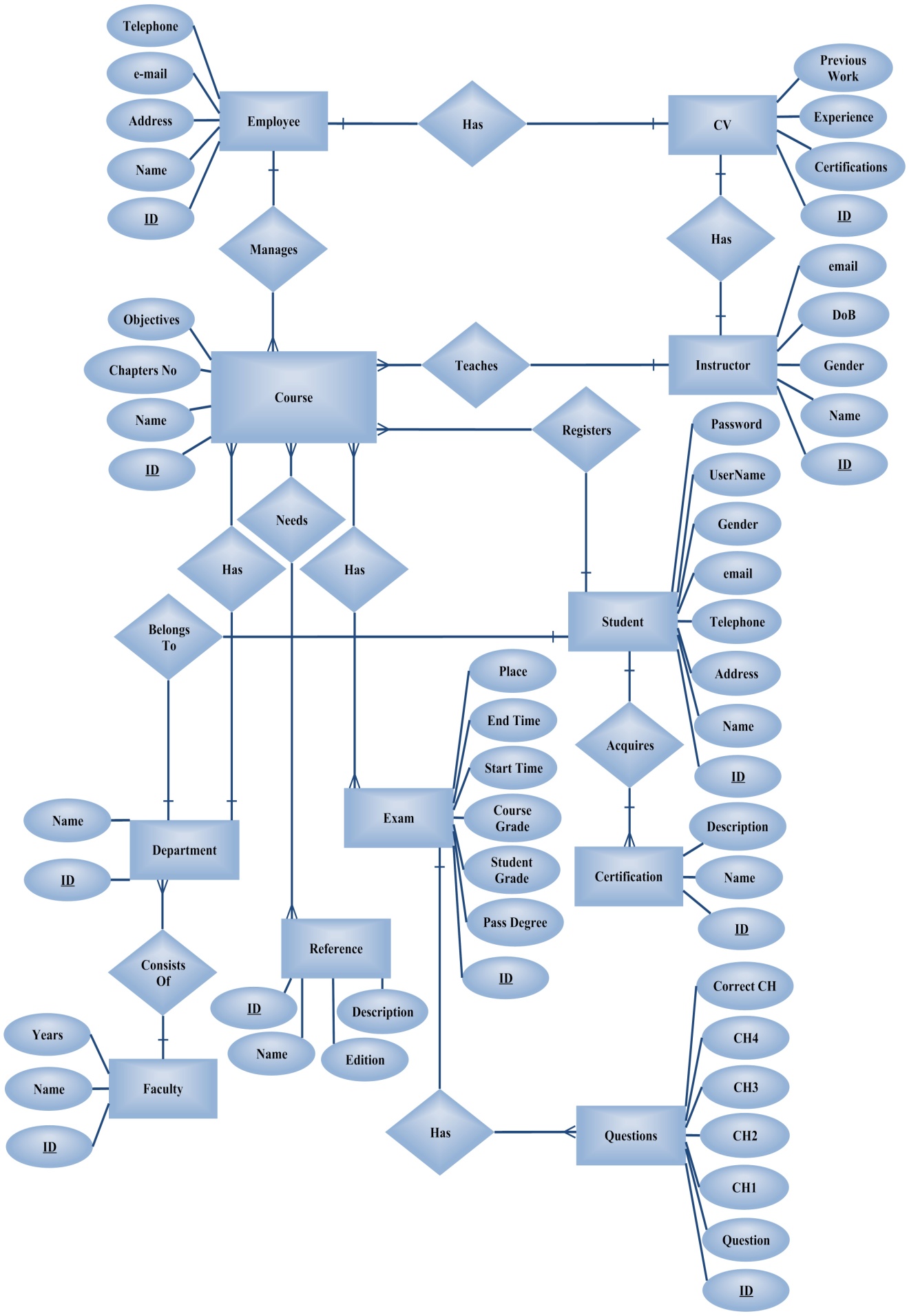


Figure 3.13: SIS Entity-Relationship (E-R) Diagram

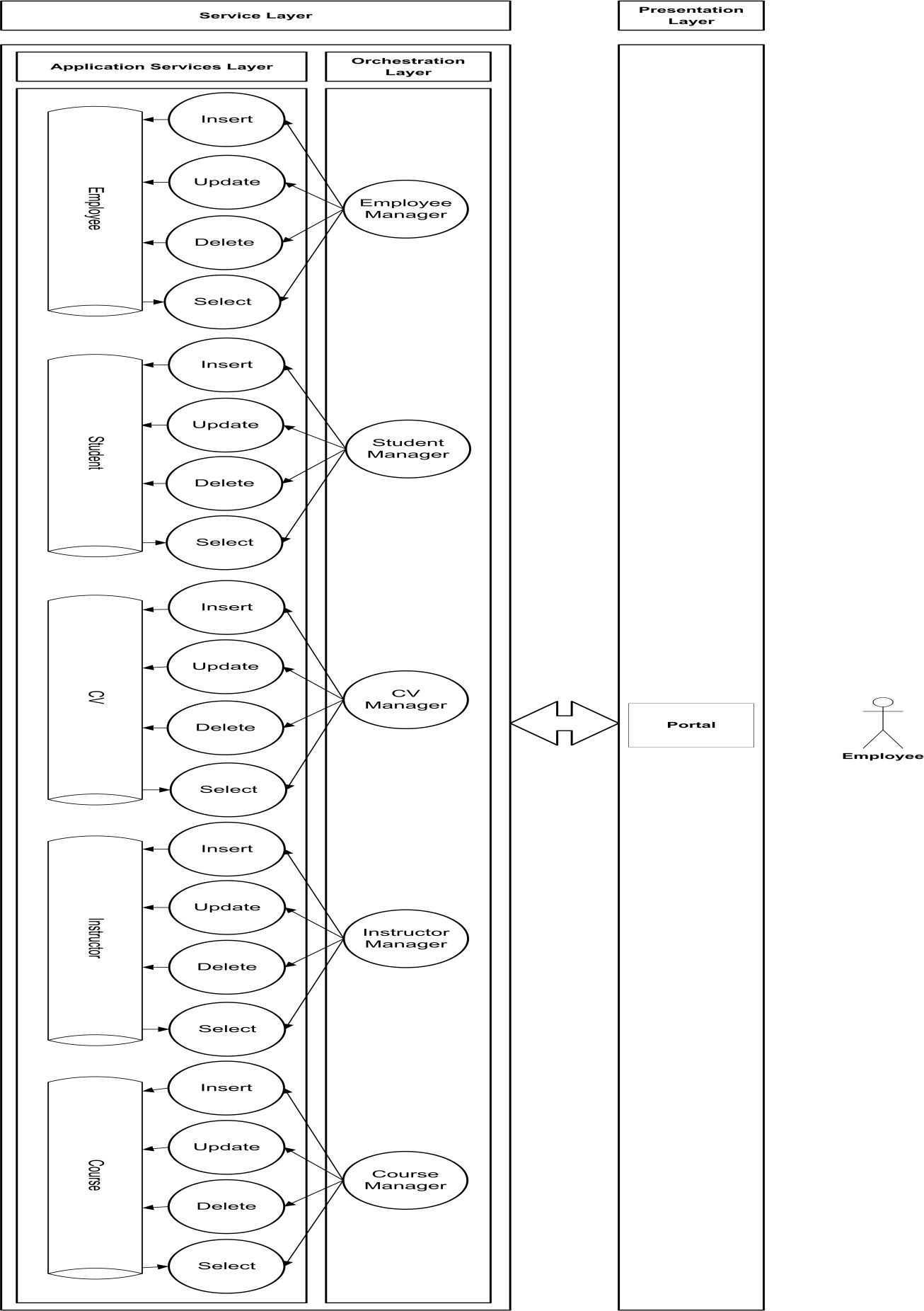
****

Figure 3.14 - Part a: SIS Architecture

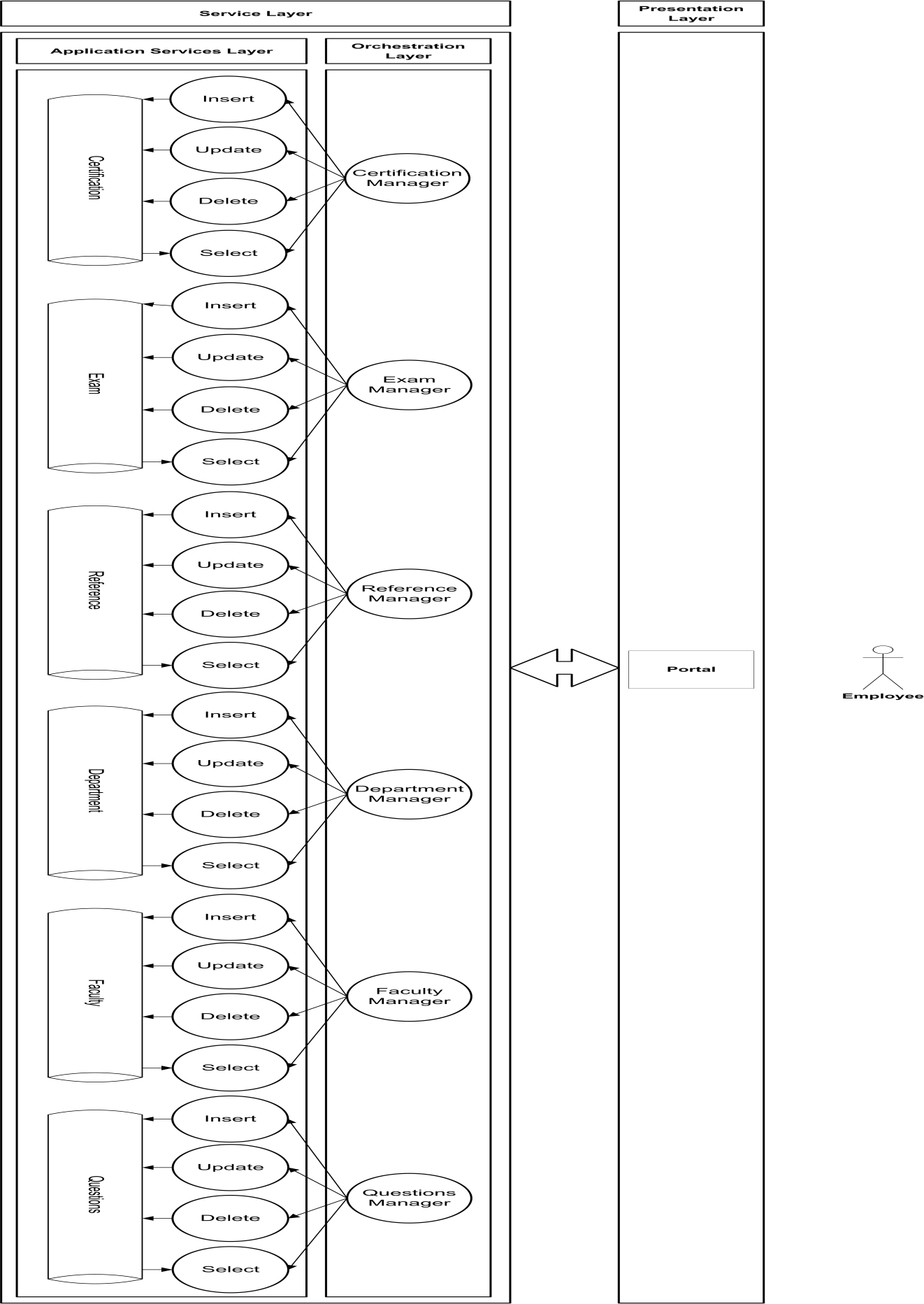
****

Figure 3.14 - Part b: SIS Architecture

* + 1. **Implement Database Tables**

Figure 3.15 presents the implemented Student Affairs Management System database tables. Database tables include: instructor, course, administrator, administrator CV, exam, reference, questions, student, student courses, faculty, department, certificates, and student certificates.

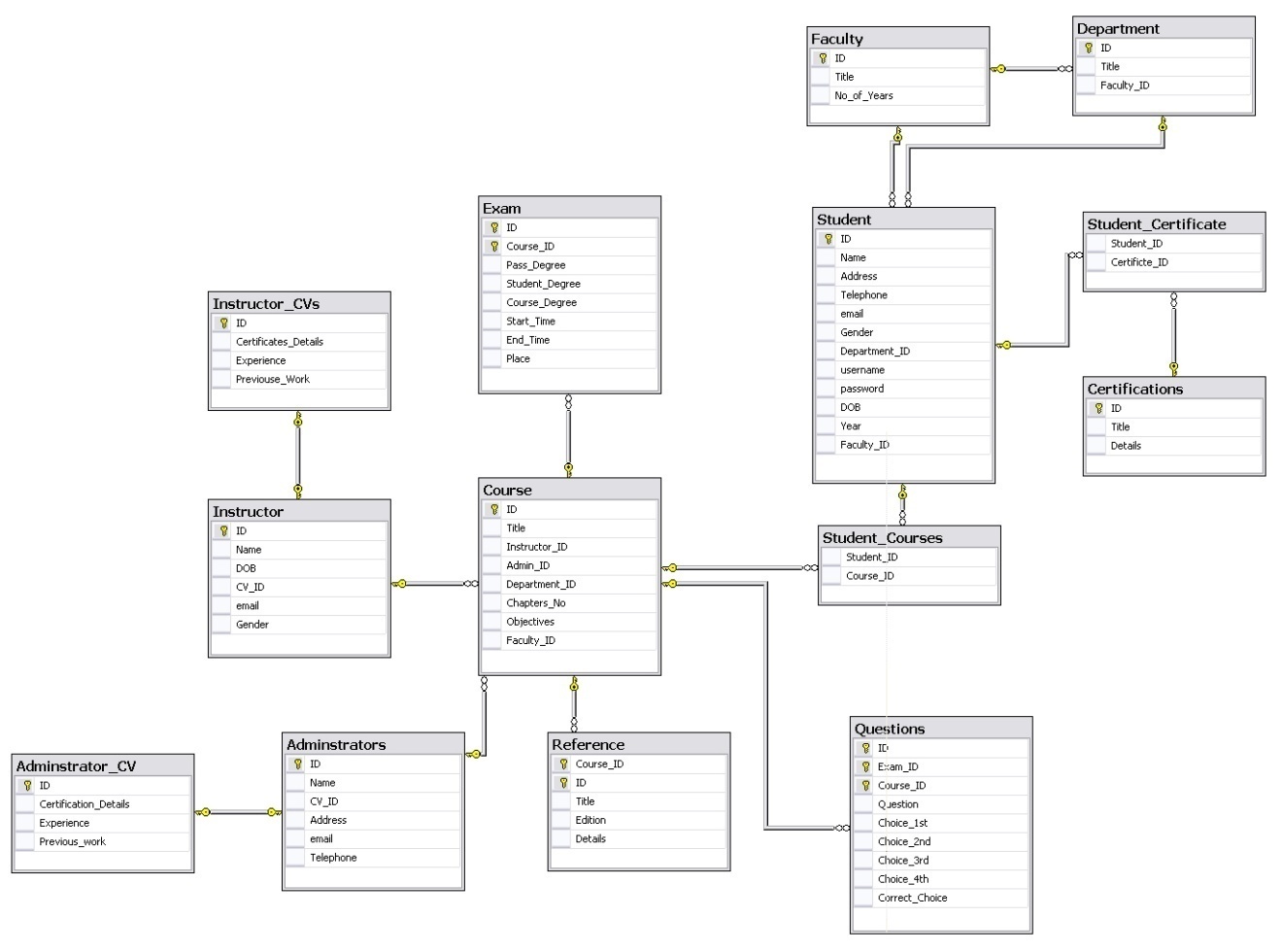


Figure 3.15: Student Affairs Management System Implemented Database Tables

* + 1. **Implement Stored Procedures**

Table 3.2 presents a list of implemented stored procedures and parameters.

Table 3.2: Student Information System Stored Procedures

|  |  |
| --- | --- |
| Student Information System Stored Procedures | |
| **Stored Procedure Name** | **Input** |
| **Employee** |  |
| Delete Employee | Employee ID [int] |
| Insert Employee | Employee ID [int]  Name [Varchar (30)]  CV ID [int]  Address [Varchar (30)]  email [Varchar (30)]  Telephone [Varchar (20)] |
| Exact Search By Name | Employee Name [Varchar (30)] |
| Partial Search By Name | Employee Name [Varchar (30)] |
| Select By ID | Employee ID [int] |
| Select All |  |
| Update Employee | Employee ID [int]  Name [Varchar (30)]  CV ID [int]  Address [Varchar (30)]  email [Varchar (30)]  Telephone [Varchar (20)] |
| **Employee CV** |  |
| Delete Employee CV | Employee CV ID [int] |
| Insert Employee CV | CV ID [int]  Certification Details [Varchar (50)]  Experiences [Varchar (50)]  Previous Work [Varchar (30)] |
| Select By ID | Employee CV ID [int] |
| Select All |  |
| Update Employee CV | CV ID [int]  Certification Details [Varchar (50)]  Experiences [Varchar (50)]  Previous Work [Varchar (30)] |
| **Certification** |  |
| Delete Certification | Certification ID [int] |
| Insert Certification | Certificate ID [int]  Name [Varchar (50)]  Details [Varchar (50)] |
| Select By ID | Certification ID [int] |
| Select All |  |
| Update Certification | Certificate ID [int]  Name [Varchar (50)]  Details [Varchar (50)] |
| **Course** |  |
| Delete Course | Course ID [int] |
| Insert Course | Course ID [int]  Name [Varchar (30)]  Instructor ID [int]  Employee ID [int]  Department ID [int]  Chapters NO [int]  Objectives [Varchar (50)]  Faculty ID [int] |
| Exact Search By Name | Course Name [Varchar (30)] |
| Partial Search By Name | Course Name [Varchar (30)] |
| Select By ID | Course ID [int] |
| Select All |  |
| Update Course | Course ID [int]  Name [Varchar (30)]  Instructor ID [int]  Employee ID [int]  Department ID [int]  Chapters NO [int]  Objectives [Varchar (50)]  Faculty ID [int] |
| **Department** |  |
| Delete Department | Faculty ID [int]  Department ID [int] |
| Insert Department | Department ID [int]  Name [Varchar (30)]  Faculty ID [int] |
| Select By ID | Faculty ID [int]  Department ID [int] |
| Select All |  |
| Update Department | Department ID [int]  Name [Varchar (30)]  Faculty ID [int] |
| **Exam** |  |
| Delete Exam | Exam ID [int]  Course ID [int] |
| Insert Exam | Exam ID [int]  Course ID [int]  Pass Degree [int]  Student Degree [int]  Course Degree [int]  Start Time [Varchar (20)]  End Time [Varchar (20)]  Place [Varchar (30)] |
| Select By ID | Exam ID [int]  Course ID [int] |
| Select All |  |
| Update Exam | Exam ID [int]  Course ID [int]  Pass Degree [int]  Student Degree [int]  Course Degree [int]  Start Time [Varchar (20)]  End Time [Varchar (20)]  Place [Varchar (30)] |
| **Faculty** |  |
| Delete Faculty | Faculty ID [int] |
| Insert Faculty | Faculty ID [int]  Name [Varchar (50)]  Years [int] |
| Select By ID | Faculty ID [int] |
| Select All |  |
| Update Faculty | Faculty ID [int]  Name [Varchar (50)]  Years [int] |
| **Instructor** |  |
| Delete Instructor | Instructor ID [int] |
| Insert Instructor | Instructor ID [int]  Name [Varchar(50)]  DOB [varchar (20)]  Instructor CV ID [int]  email [Varchar (50)]  Gender [Char (10)] |
| Exact Search By Name | Instructor Name [Varchar (30)] |
| Partial Search By Name | Instructor Name [Varchar (30)] |
| Select By ID | Instructor ID [int] |
| Select All |  |
| Update Instructor | Instructor ID [int]  Name [Varchar(50)]  DOB [varchar (20)]  Instructor CV ID [int]  email [Varchar (50)]  Gender [Char (10)] |
| **Instructor CV** |  |
| Delete Instructor CV | Instructor CV ID [int] |
| Insert Instructor CV | Instructor CV ID [int]  Certification Details [Varchar (50)]  Experience [Varchar (50)]  Previouse Work [Varchar (50)] |
| Select By ID | Instructor CV ID [int] |
| Select All |  |
| Update Instructor CV | Instructor CV ID [int]  Certification Details [Varchar (50)]  Experience [Varchar (50)]  Previouse Work [Varchar (50)] |
| **Questions** |  |
| Delete Question | Course ID [int]  Exam ID [int]  Question ID [int] |
| Insert Question | Question ID [int]  Exam ID [int]  Course ID [int]  Question [Varchar (50)]  Choice1 [Varchar (30)]  Choice2 [Varchar (30)]  Choice3 [Varchar (30)]  Choice4 [Varchar (30)]  Correct Choice [Varchar (30)] |
| Select By ID | Question ID [int]  Course ID [int]  Exam ID [int] |
| Select All |  |
| Update Question | Question ID [int]  Exam ID [int]  Course ID [int]  Question [Varchar (50)]  Choice1 [Varchar (30)]  Choice2 [Varchar (30)]  Choice3 [Varchar (30)]  Choice4 [Varchar (30)]  Correct Choice [Varchar (30)] |
| **Reference** |  |
| Delete Reference | Reference ID [int]  Course ID [int] |
| Insert Reference | Course ID [int]  Reference ID [int]  Name [Varchar (30)]  Edition [Varchar (30)]  Description [Varchar (50)] |
| Select All | Course ID [int] |
| Select By ID | Reference ID [int]  Course ID [int] |
| Update Reference | Course ID [int]  Reference ID [int]  Name [Varchar (30)]  Edition [Varchar (30)]  Description [Varchar (50)] |
| **Student** |  |
| Delete Student | Student ID [int] |
| Insert Student | Student ID [int]  Name [Varchar (30)]  Address [Varchar (30)]  Telephone [Varchar (30)]  email [Varchar (30)]  Gender [Char (10)]  Department ID [int]  Username [Varchar (30)]  Password [Varchar (30)]  DOB [Varchar (30)]  Year [Char (10)]  Faculty ID [int] |
| Exact Search By Name | Student Name [Varchar (30)] |
| Partial Search By Name | Student Name [Varchar (30)] |
| Select All |  |
| Select By ID | Faculty ID [int]  Department ID [int]  Student ID [int] |
| Update Student | Student ID [int]  Name [Varchar (30)]  Address [Varchar (30)]  Telephone [Varchar (30)]  email [Varchar (30)]  Gender [Char (10)]  Department ID [int]  Username [Varchar (30)]  Password [Varchar (30)]  DOB [Varchar (30)]  Year [Char (10)]  Faculty ID [int] |
| **Student Certificates** |  |
| Insert Student Certificate | Student ID [int]  Certificate ID [int] |
| Select All |  |
| **Student Courses** |  |
| Insert Student Course | Student ID [int]  Course ID [int] |
| Select All |  |

* + 1. **Implement Classes**

SIS class diagram include definition for the main classes and methods that form the system. Class diagram includes instructor, course, administrator, administrator CV, exam, reference, questions, student, student courses, faculty, department, certificates, and student certificates with the attributes highlighted in the E-R diagram and at least the four methods (insert, update, delete, select) presented in the proposed architecture.

1. **Library Information System**

LIS is responsible for automating library activities and has been widely known and accepted for more than forty years. LISs represents the main library automation functionality and almost share the same features and functionalities among them as presented in proposed LIS.

* 1. **Analysis of LIS**

Figure 3.16 presents LIS Use Case.



Figure 3.16: LIS Use Case

LIS Processes include:

* **Registration:** students, instructors, and outsider faculty members (registrars) can register for the library to have the ability to borrow and / or buy books.
* **Purchase Books:** invoked either to satisfy internal library requirements of newly released books or to satisfy clients’ requests. Purchase Books stores data about Purchase Orders from the library to Suppliers, and Suppliers details.
* **Lend Books:** Instructors, Students, Employees, and Registrants can borrow one or more of the books available in the library. A record of Library Clients and Borrow activity details needs to be recorded.
* **Sell Books:** Library Clients can buy borrowed books, or request non-available books via special order.
* **Manage Library Employees Data:** involves the main operations of: Insert, Update, and Delete employees.
* **Manage Books Data:** considers managing and recording data about existing books in the library.

LIS automates the common activities of educational institutions libraries.

**Registration**

Registration to the library requires the steps shown in figure 3.17.



**Figure 3.17: Registration Analysis**

**Sell Books**

Figure 3.18 shows the steps that instructor, student, or ordinary registrar has to go through to buy a book from the library.



**Figure 3.18: Sell Books Analysis**

**Purchase Books**

Purchase book activities is depicted in figure 3.19 where Library Employee purchases books from suppliers either to be available for loan by library or to be sold.



**Figure 3.19: Purchase Book Analysis**

**Borrow Books**

Figure 3.20 shows the steps the borrower has to go through to borrow a book from the library.



**Figure 3.20: Borrow Book**

* 1. **Design of LIS**

Further analysis of proposed processes implies system entities presented in figure 3.21. LIS Entity-Relationship diagram is depicted in figure 3.22.

Figure 3.21: LIS Entities

* + 1. **Proposed LIS Architecture**

Figure 3.23 presents the proposed services based LIS architecture. The architecture consists of two layers: presentation, and services layer. Services layer holds two sub layers: orchestration and application services layer. Orchestration layer addresses entity centric services like employee, client, author, publisher, book, order, purchase order, and supplier managers that are responsible for consuming services of Insert, Update, Delete, and Retrieve the system entities presented in the application services layer. Presentation layer portal consumes the exposed Web services presented at services layer.

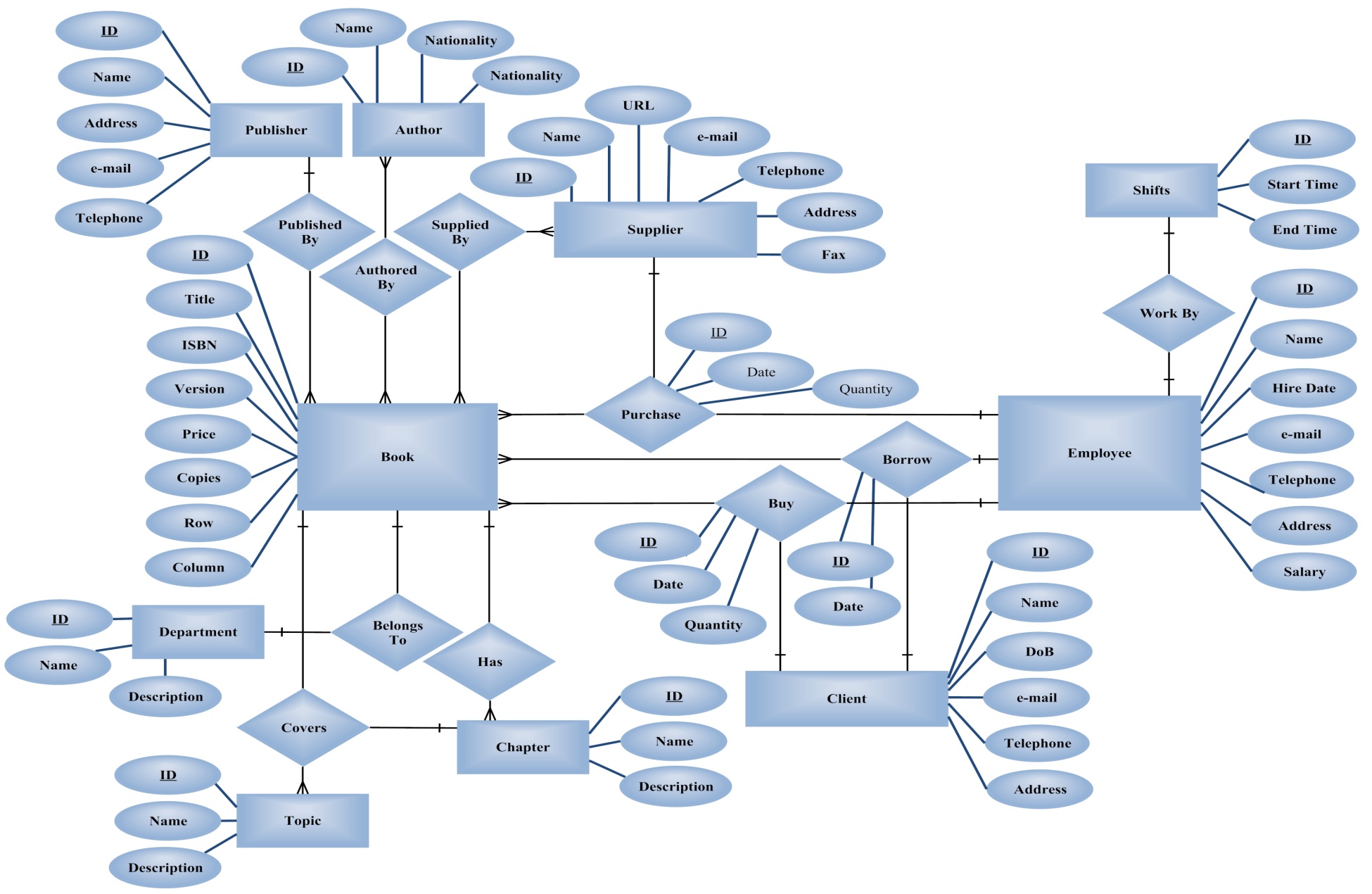


Figure 3.22: LIS E-R Diagram

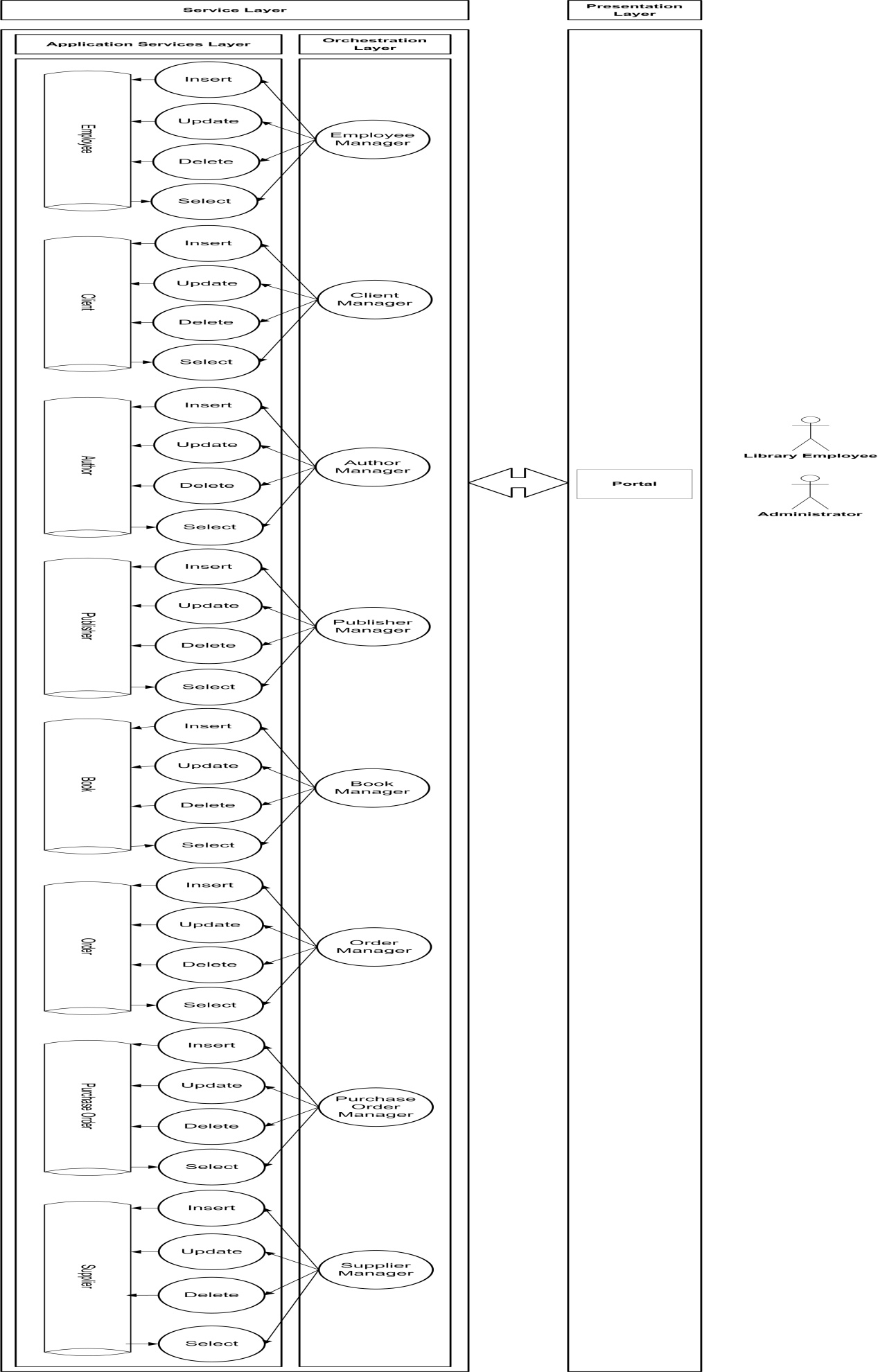


Figure 3.23: LIS Proposed Architecture

70

72

* + 1. **LIS Database Tables**

Figure 3.24 presents the implemented LIS database tables. Database tables include: employee, shift, author, borrow, chapter, book, client, order, order details, publisher, topics, department, purchase order, purchase order details, and supplier.

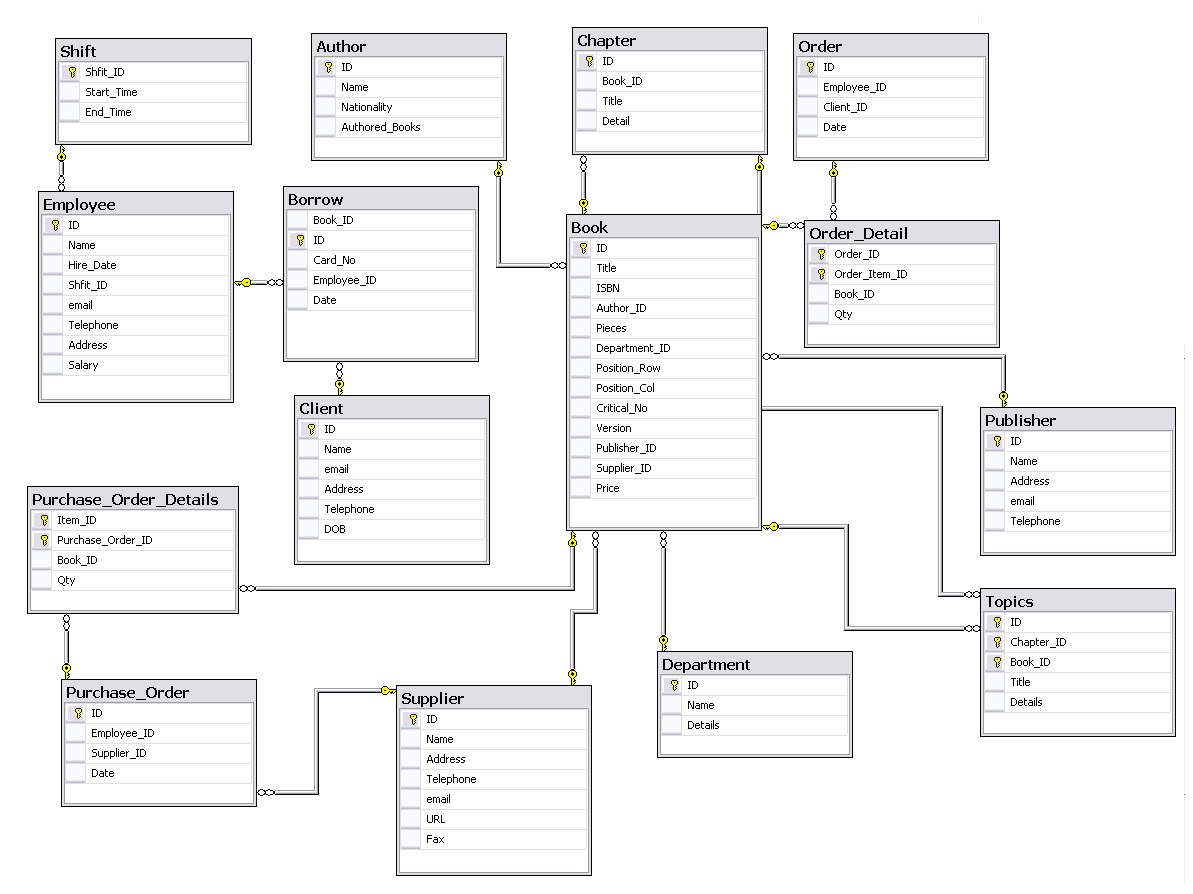


Figure 3.24: LIS Database Tables

* + 1. **LIS Stored Procedures**

Table 3.3 lists the implemented LIS stored procedures and parameters of each one.

Table 3.3: LIS Stored Procedures

|  |  |
| --- | --- |
| LIS Stored Procedures | |
| **Stored Procedure Name** | **Input** |
| **Author** |  |
| Delete Author | Author ID [int] |
| Insert Author | Author ID [int]  Author Name [Varchar (30)]  Nationality [Varchar (30)]  No. of Authored Books [int] |
| Exact Search Author Name | Author Name [Varchar (30)] |
| Partial Search Author Name | Author Name [Varchar (20)] |
| Display All Authors  71 |  |
| Display Author | Author ID [int] |
| Update Author | Author ID [int]  Author Name [Varchar (30)]  Nationality [Varchar (30)]  No. of Authored Books [int] |
| **Book** |  |
| Delete Book | Book ID [int] |
| Insert Book | Book ID [int]  Book Name [Varchar (40)]  ISBN [Varchar(20)]  Author ID [int]  Price [int]  Department ID [int]  Row [int]  Column[int]  Critical No.[int]  Version [Varchar (8)]  Publisher ID [int]  Supplier ID [int] |
| Exact Search Book Name | Book Name [Varchar (40)] |
| Partial Search Book Name | Book Name [Varchar (40)] |
| Display All Books |  |
| Display Book | Book ID [int] |
| Update Book | Book ID [int]  Book Name [Varchar (40)]  ISBN [Varchar(20)]  Author ID [int]  Price [int]  Department ID [int]  Row [int]  Column[int]  Critical No.[int]  Version [Varchar (8)]  Publisher ID [int]  Supplier ID [int] |
| **Borrow** |  |
| Delete Borrow | Borrow ID [int] |
| Insert Borrow | Book ID [int]  Borrow ID [int]  Client ID [int]  Employee ID [int]  Date [Varchar (30)] |
| Display All Borrows |  |
| Display Borrow | Borrow ID [int] |
| Update Borrow  72 | Book ID [int]  Borrow ID [int]  Client ID [int]  Employee ID [int]  Date [Varchar (30)] |
| **Chapter** |  |
| Delete Chapter | Book ID [int]  Chapter ID [int] |
| Insert Chapter | Chapter ID [int]  Book ID [int]  Name [Varchar (30)]  Description [varchar (40)] |
| Exact Search Chapter Name | Chapter Name [Varchar (30)] |
| Partial Search Chapter Name | Chapter Name [Varchar (30)] |
| Display Book Chapters | Book ID [int] |
| Display Chapter | Book ID [int]  Chapter ID [int] |
| Update Chapter | Chapter ID [int]  Book ID [int]  Name [Varchar (30)]  Description [varchar (40)] |
| **Client** |  |
| Delete Client | Client ID [int] |
| Insert Client | Client ID [int]  Name [Varchar (50)]  e-mail [Varchar (30)]  Address [Varchar (30)]  Telephone [Varchar (50)]  DOB [DateTime] |
| Display All Clients |  |
| Display Client | Client ID [int] |
| Update Customer | Client ID [int]  Name [Varchar (50)]  e-mail [Varchar (30)]  Address [Varchar (30)]  Telephone [Varchar (50)]  DOB [DateTime] |
| **Department** |  |
| Delete Department | Department ID [int] |
| Insert Department | Department ID [int]  Name [char (10)]  Description [Varchar (50)] |
| Display All Departments |  |
| Display Department | Department ID [int] |
| Update Department | Department ID [int]  Name [char (10)]  Description [Varchar (50)] |
| **Employee** |  |
| Delete Employee | Employee ID [int] |
| Insert Employee  73  75 | Employee ID [int]  Name [Varchar (20)]  Hire Date [DateTime]  Shift ID [int]  e-mail [Varchar (20)]  Telephone [nvarchar (20)]  Address [Varchar (30)]  Salary [int] |
| Display All Employees |  |
| Display Employee | Employee ID [int] |
| Update Employee | Employee ID [int]  Name [Varchar (20)]  Hire Date [DateTime]  Shift ID [int]  e-mail [Varchar (20)]  Telephone [nvarchar (20)]  Address [Varchar (30)]  Salary [int] |
| **Order** |  |
| Delete Order | Order ID [int] |
| Insert Order | Order ID [int]  Employee ID [int]  Client ID [int]  Date [Varchar (16)] |
| Display All Orders |  |
| Display Order | Order ID [int] |
| Update Order | Order ID [int]  Employee ID [int]  Client ID [int]  Date [Varchar (16)] |
| **Order Details** |  |
| Delete Order Detail | Order ID [int]  Order Detail ID [int] |
| Insert Order Detail | Order ID [int]  Order Item ID [int]  Book ID [int]  Quantity [int] |
| Display All Order Details | Order ID [int] |
| Display Order Detail | Order ID [int]  Order Item ID [int] |
| Update Order Detail | Order ID [int]  Order Item ID [int]  Book ID [int]  Quantity [int] |
| **Publisher** |  |
| Delete Publisher | Publisher ID [int] |
| Insert Publisher | Publisher ID [int]  Name [Varchar (50)]  Address [Varchar (50)]  e-mail [Varchar (50)]  Telephone [Varchar (15)] |
| Exact Search Publisher Name | Publisher Name [Varchar (50)] |
| Partial Search Publisher Name | Publisher Name [Varchar (50)] |
| Display All Publishers  74 |  |
| Display Publisher | Publisher ID [int] |
| Update Publisher | Publisher ID [int]  Name [Varchar (50)]  Address [Varchar (50)]  e-mail [Varchar (50)]  Telephone [Varchar (15)] |
| **Purchase Order** |  |
| Delete Purchase Order | Purchase Order ID [int] |
| Insert Purchase Order | Purchase Order ID [int]  Employee ID [int]  Supplier ID [int]  Date [Varchar (16)] |
| Display All Purchase Orders |  |
| Display Purchase Order | Purchase Order ID [int] |
| Update Purchase Order | Purchase Order ID [int]  Employee ID [int]  Supplier ID [int]  Date [Varchar (16)] |
| **Purchase Order Details** |  |
| Delete Purchase Order Detail | Purchase Order ID [int]  Purchase Order Detail ID [int] |
| Insert Purchase Order Detail | Purchase Order Item ID [int]  Purchase Order ID [int]  Book ID [int]  Quantity [int] |
| Display All Purchase Order Details | Purchase Order ID [int] |
| Display Purchase Order Detail | Purchase Order ID [int]  Purchase Order Item ID [int] |
| Update Purchase Order Detail | Purchase Order Item ID [int]  Purchase Order ID [int]  Book ID [int]  Quantity [int] |
| **Shift** |  |
| Delete Shift | Shift ID (int) |
| Insert Shift | Shift ID [int]  Start Time [Varchar (20)]  End Time [Varchar (20)] |
| Display All Shifts |  |
| Display Shift | Shift ID [int] |
| Update Shift | Shift ID [int]  Start Time [Varchar (20)]  End Time [Varchar (20)] |
| **Supplier** |  |
| Delete Supplier | Supplier ID (int) |
| Insert Supplier  75 | Supplier ID [int]  Name [Varchar (50)]  Address [Varchar (50)]  Telephone [Varchar(50)]  e-mail [Varchar(50)]  URL [Varchar(50)]  Fax [Varchar(50)] |
| Exact Search Supplier Name | Supplier Name [Varchar (50)] |
| Partial Search Supplier Name | Supplier Name [Varchar (50)] |
| Display All Suppliers |  |
| Display Supplier | Supplier ID [int] |
| Update Supplier | Supplier ID [int]  Name [Varchar (50)]  Address [Varchar (50)]  Telephone [Varchar(50)]  e-mail [Varchar(50)]  URL [Varchar(50)]  Fax [Varchar(50)] |
| **Topic** |  |
| Delete Topic | Book ID (int)  Chapter ID (int)  Topic ID (int) |
| Insert Topic | Topic ID [int]  Chapter ID [int]  Book ID [int]  Name [Varchar (30)]  Details [Varchar (40)] |
| Exact Search Topic Name | Topic Name [Varchar (30)] |
| Partial Search Topic Name | Topic Name [Varchar (30)] |
| Display Chapter Topics | Book ID [int]  Chapter ID [int] |
| Display Topic | Book ID [int]  Chapter ID [int]  Topic ID [int] |
| Update Topic | Topic ID [int]  Chapter ID [int]  Book ID [int]  Name [Varchar (30)]  Details [Varchar (40)] |

* + 1. **LIS Classes**

LIS class diagram include classes that address employee, client, author, publisher, book, order, purchase order, and supplier. Classes address at least the four main database operations: insert, update, delete, and select.

76

78

1. **Summary**

This chapter highlighted the proposed architecture characteristics and layered nature. SIS, and LIS are presented. Use cases, system processes, analysis and design details like entities, E-R diagrams, class diagrams, database tables, and proposed service based architectures. Faculty Information System provides the required functionalities to store data about faculties, but SIS addresses the internal faculty requirements. LIS automates all library activities and exposes library components via service layer that can be utilized within different external systems.

77