

# Representation of Object-Oriented Database for the Development of Web Based Application Using Db4o

Vipin Saxena, Ajay Pratap

Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow, India.  
Email: vsax1@rediffmail.com, pratap\_aj@yahoo.co.in

Received June 23<sup>rd</sup>, 2012; revised July 26<sup>th</sup>, 2012; accepted August 10<sup>th</sup>, 2012

## ABSTRACT

Impedance mismatch of data flow is the major problem in the relational database when one is using the web based application development on the computer system. It effects the development of scalable and reusable web application. Due to this, the research cost of development and maintenance increases. Therefore the present work is an attempt towards an object-oriented database system by using Db4o to overcome the impedance mismatch problem. In the current work, the development of web application for Indian Postal Services by using Db4o is demonstrated on dot NET platform. Before developing the application, the authors used a Unified Modeling Language (UML) model in the form of UML class, sequence and use-case diagrams. The Db4o is used to store the object-oriented database and the performance of Db4o is observed through several object-oriented queries and corresponding results are demonstrated.

**Keywords:** RDBMS; OODBMS; Db4o; Web Development; UML

## 1. Review of Research Work

The relational databases are the most common database management system (DBMS) used as a backend for the web based application. In the current trend of software development, many of the software companies are still using the relational databases but due to impedance mismatch of data flow and evolution of graphical user interface (GUI), software industries are switching from relational databases to object-oriented databases. Let us first describe the important references related to databases. Elmasri and Navathe [1] have presented a clear explanation of theory and design, broad coverage of models and real systems, and an up-to-date introduction to modern database technologies results with a leading introduction to database systems. Silberschatz *et al.* [2] have discussed the inner working of database system and also discussed about few commercial databases as well as open source database systems. Paterson *et al.* [3] have described to configure the Db4o, querying and managing objects, performing transactions, and replicating data. Db4o is highly reliable and scalable, which cuts down on development time by smoothly integrating into the system, cutting out the costly object-relational mapping typical for larger applications. Db4o developers community [4] present various blogs and forums related to object-oriented database system such as Db4o, Versant Object Database, Versant JPA, ODBMS.ORG, which helps to get deep understanding on various topics. In [5],

Rosenberger made effort to store all objects manually in SQL and has took up half time of total development time and today it is the most well known brand in the niche category “object databases”. A big RDBMS machine is oversized for a cell phone. In this case, a tight integration between the application and the database are helpful to achieve top performance. Db4o is also used for caching in near real-time environments when application models are complex, such as in the financial sector, for trading applications.

Rădulescu [6] has given a brief look about SQL technology and demonstrated its limitations on persistence feature of an object-oriented model and also discussed the principles and concepts of Db4o and its implementation on dot NET platform. Danturthi [7] has described a comparative study between the three different methods of data access: SQL Server 2005 with stored procedure, LINQ to SQL and Db4o. Db4o solves the problem of impedance mismatch and making the development of database model much simpler and similar to the application domain model. Liaw *et al.* [8] have discussed a three tier web based application which can be developed for taking advantage of an object-oriented database instead of using ADO.NET or traditional relational database. Bernardi *et al.* [9] have presented the validation and the performance evaluation of systems with the help of UML sequence diagrams and state charts. They proposed an automatic translation of state charts and sequence dia-

Grams into generalized stochastic petri nets based on the abstract syntax of the UML collaborations and of the state machines packages.

Rumbaugh *et al.* [10] completely described UML concepts, including major revisions to sequence diagrams, activity models, state machines, components, internal structure of classes and components, and profiles. Umoh *et al.* [11] have focused on creating a UML structure by specifying the use-case, classes, and activities in the client-server application to design a web-based object-oriented database management system (OODBMS). A working prototype of the system on three-tier client server architecture is discussed in this paper. Chaurasia and Saxena [12] have also introduced a model design through UML for a mobile based electricity bill deposit system. A real case study of Uttar Pradesh Electricity Bill deposit System is considered. In [13], Tagliati and Caloro have discussed the application of UML for the analysis of dynamic words. Mangala [14] also described about the need of ASP.NET and designed the programming model through ASP.NET, which is a completely re-engineered and enhanced technology that offers much, much more than traditional ASP and can increase productivity significantly.

## 2. Background

### 2.1. Object-Oriented Database

Evolution of internet and extranet has increased the usage of web-based technology and the companies have shown their interest in object-oriented database management system (OODBMS) to handle the complex data. OODBMS is fusion of two technologies: object-oriented system and database management system. New applications require data persistence, transaction, authorization, distribution, buffering and data scalability associated with the database system, which is fulfilled by the OODBMS. The term object-oriented database system first appeared in 1985 and 2004 was second growth period when open source database system such as Db4o and DTS/SI emerged, which was affordable and easy to use.

Db4o (database for objects) is an open sourced bi-licensed software (General Public License and Commercial), written in both java and .NET and can run on any operating system that supports java or .NET. It stores the objects directly without changing their characteristics or crops them to fit in a table of relational data model. The db4o project was started in 2000 by Carl Rosenberger and it was commercially launched in year 2004 by Db4objects Inc but in 2008 it was brought by Versant Corporation.

### 2.2. Unified Modeling Language

The Unified Modeling Language (UML) which is a very

dominant modeling graphical language for specifying, constructing, presenting and documenting the artifacts of software system. It is a collection of best engineering practices that have successful in the modeling for a design of a huge and complex systems. Modeling through Unified Modeling language has three categories: Class Model, State Model and Interaction Model. The class modeling is done with the help of object diagram and class diagram. The state model is done, using state diagram and the Interaction modeling is done with the help of activity diagram, use-case diagram and sequence diagram. The main task of the UML is to create a simple, well documented and easy to understand software model for the people.

### 2.3. Web Based Applications

Web based applications are classified into two major categories: static web applications and dynamic web applications. Static web applications are those applications, where the elements comprising the web application are static HTML pages. In this environment, the end user can not interact or modify the application behaviour. On the other hand the dynamic web applications have capability to interact or modify with the application behaviour. The end user gives input and it can be parsed both at the client end or server end. Client side scripts the language by using VB Script and Java Script which are required to parse and interact according to the client input. Using the client side scripting language reduces the traffic of network and also increases the throughput of web based applications.

Server-side scripting is a web server technology and the programs that are executed on the server to generate dynamic web pages. The languages used for these tasks are normal programming languages with special libraries/packages for server side scripting like ASP, C via CGI (\*.c, \*.csp), PHP (\*.php), ASP.NET (\*.aspx), Java via java server pages (\*.jsp).

The clients use web browsing software to interact with the web server. In ASP.NET when any \*.aspx file is requested then the request is send to the Internet Information Server (IIS), which is send to .NET engine. The NET engine parses the client request and converts it to HTML, which is returned to IIS. Finally, IIS returns the HTML to the web browser. Hyper text transfer protocol (HTTP) and file transfer protocol (FTP) are used to deal with the client request and server responses.

### 3. Architecture of Web Based Application

The structure of the web based application can be single-tiered, two-tiered or three-tiered. Here, authors are using the three-tiered architecture as shown below in **Figure 1**, which contains the presentation layer, application layer

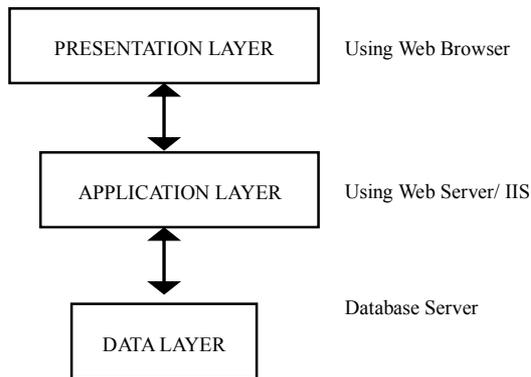


Figure 1. Three tiered web architecture.

and data layer.

Presentation of data and GUI environments are provided to the user with the help of presentation layer. All the rules for the accessing of data and business logic is presented in the application layer. The data layer handles the requests from the application layer. **Figure 1** represents the three tier application architecture of web based applications.

## 4. Implementation

The project has been developed in Microsoft visual studio 2008 IDE with ASP.NET using C# on Microsoft

Windows XP (SP1) operating system. An object-oriented database management system Db4o 7.14 is used as a back-end. Db4o is an open source OODBMS, which is freely available and it can be downloaded from: <http://community.versant.com/Downloads.aspx>.

The software and hardware requirements are described above in the **Table 1**.

Installation of Db4o database is done with the help of an object browser known as Object Manager Enterprise (OME). After installing it as a plug-in of Ms-Visual Studio 2008, one can get a sub option of object Manager Enterprise in tools option of menu bar. After the completion of installation, new tools are added in the window and it is shown in **Figure 2**.

To create a new web application choose “New Project” from the file menu, in the pop up window select “web” from the visual c# project type. Then choose “ASP.NET Web Application” from the template sub window as shown in **Figure 3**. We have to add Db4-objects.Db4o.dll files to the references of project, which is freely available and the process is shown in the **Figures 3 and 4**.

### 4.1. Sample Queries for Indian Post Office Database

After completing these tasks, one can store new objects

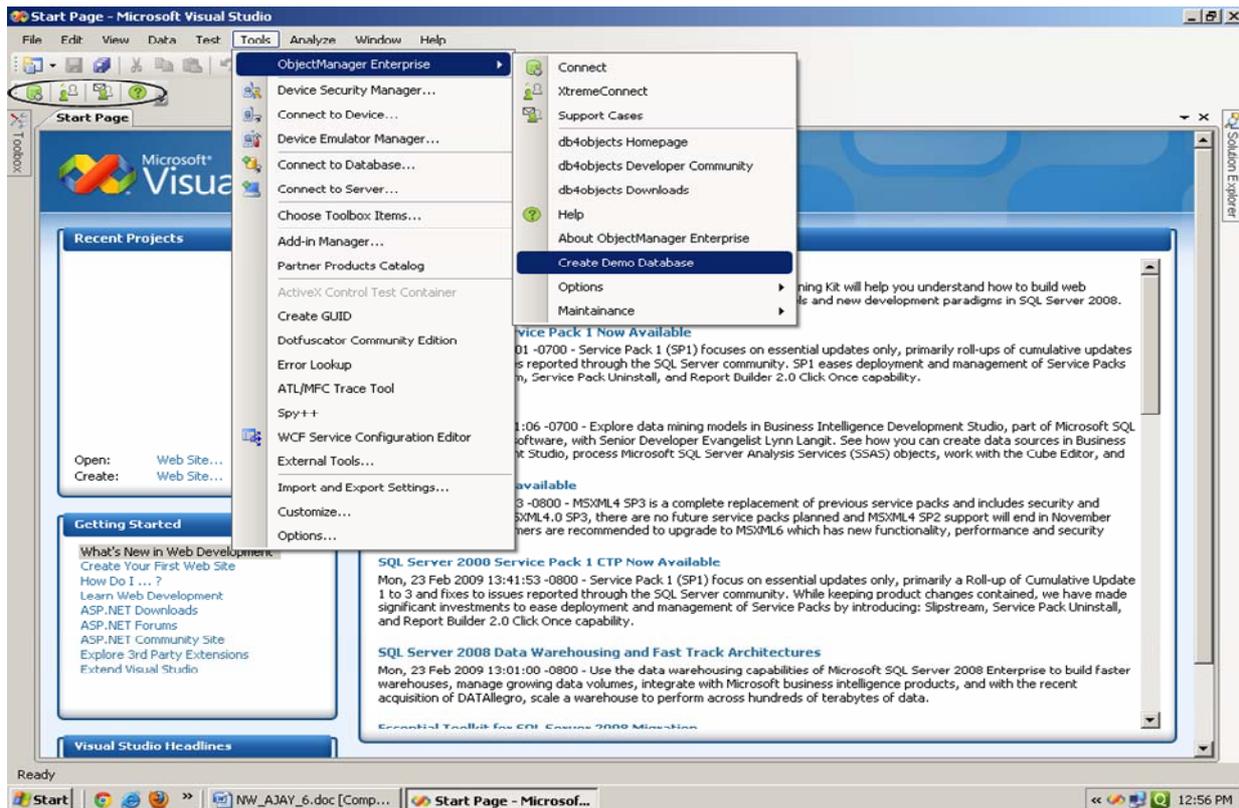


Figure 2. Installation of Db4o.

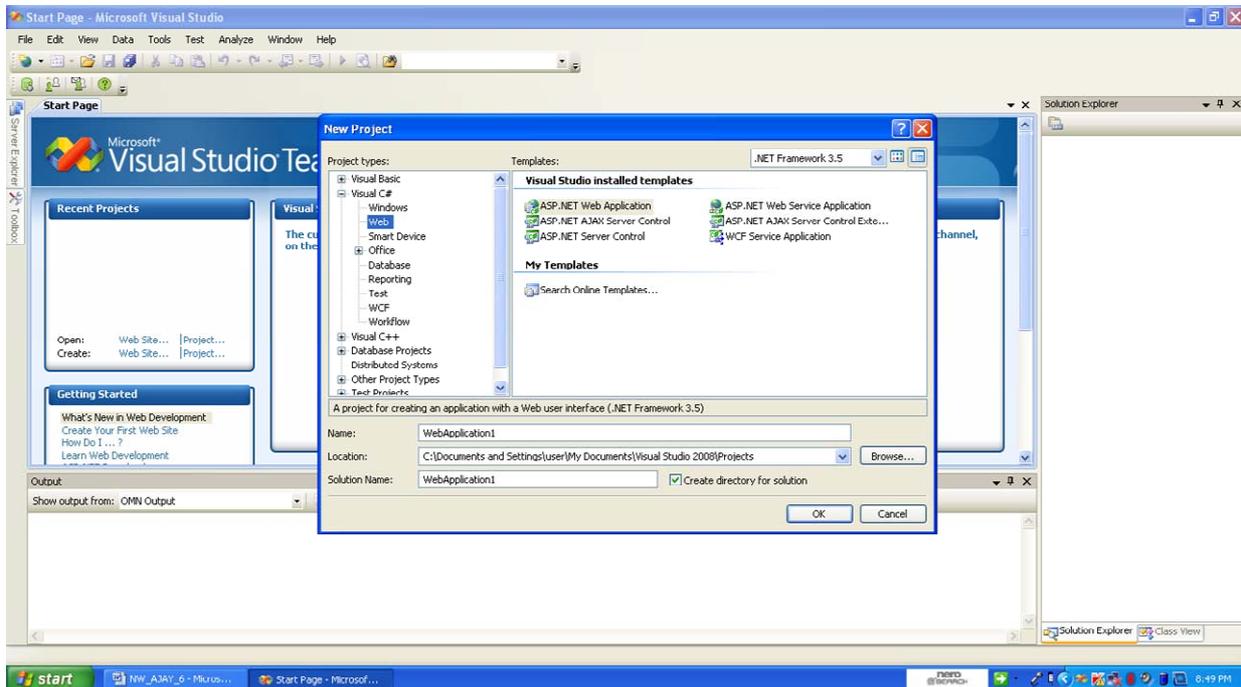


Figure 3. Starting Db4o web project.

Table 1. Software and hardware requirement table.

Hardware Requirements			
	Min	Recommended	Used
Processor	400 MHz	1 GHz	Intel Pentium Dual CPU 2.20 GHz
RAM	96 MB	256 MB	0.99 GB
Software Requirements			
Operating System	Microsoft Windows XP Professional (Version 2002) SP1 Microsoft Visual Studio 2008 (Version 9.0.210222)		
Application Softwares	Microsoft .NET Framework (Version 3.5) Db4o version 7.12		

in Db4o. Object files are the class objects with properties. Here we have four major classes, which are User, Scheme Database, Office Database and SalesRecord. The User class has three sub-classes, which are Admin, Employee and Customer. The User object file is created by creating a class in C# with all its properties. These properties are converted as a database object file for Db4o. The code for object storage mechanism in Db4o is given below and also shown in the **Figure 5**.

Code for storing User Class in C# for Db4o:

```
class User
{
    private int UID;
    private string Name;
    private string Address;
    private string MobNumber;
    public User (int UID, string Name, string Address,
string MobNumber)
```

```
{
    this.UID=UID;
    this.Name=Name;
    this.Address=Address;
    this.MobNumber=MobNumber;
}
public User()
{
    User user=new User(1001, "ajay", "Lucknow",
"9936189152");
try{
    IObjectContainer
db=Db4oFactory.OpenFile("User.yap");
db.Store(User);
db.Commit();
}
}
```

Three methods used for querying the objects in Db4o

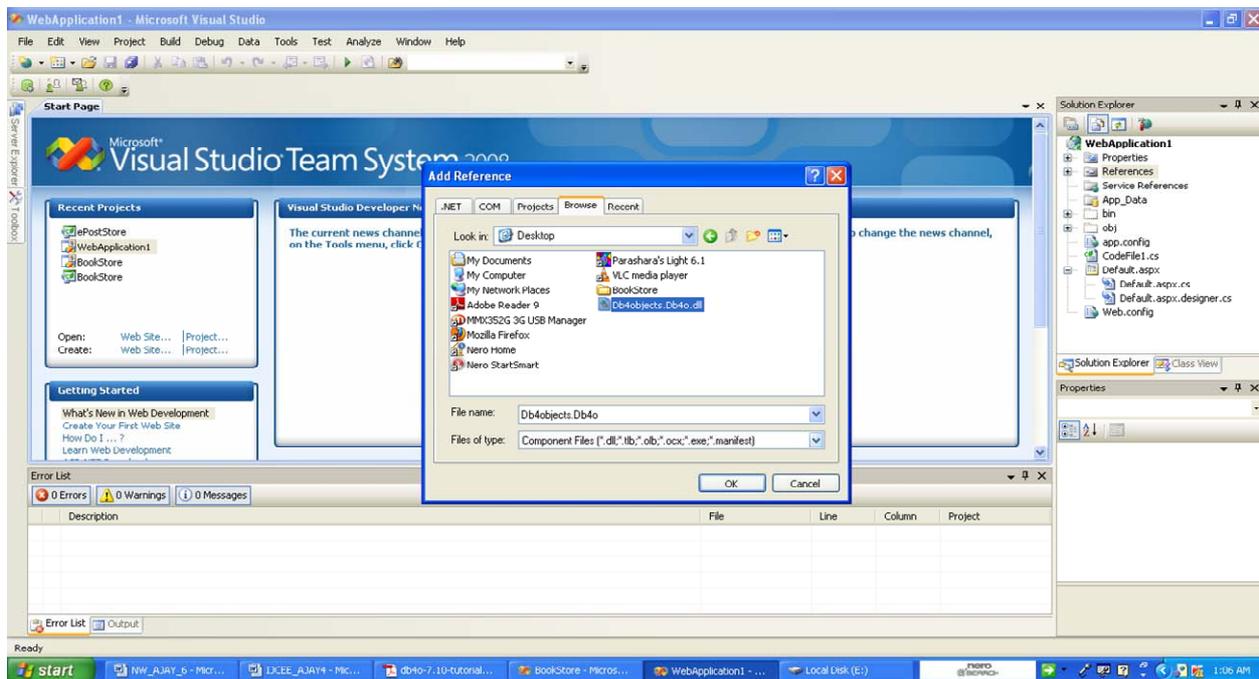


Figure 4. Adding Db4objects.Db4o.dll file.

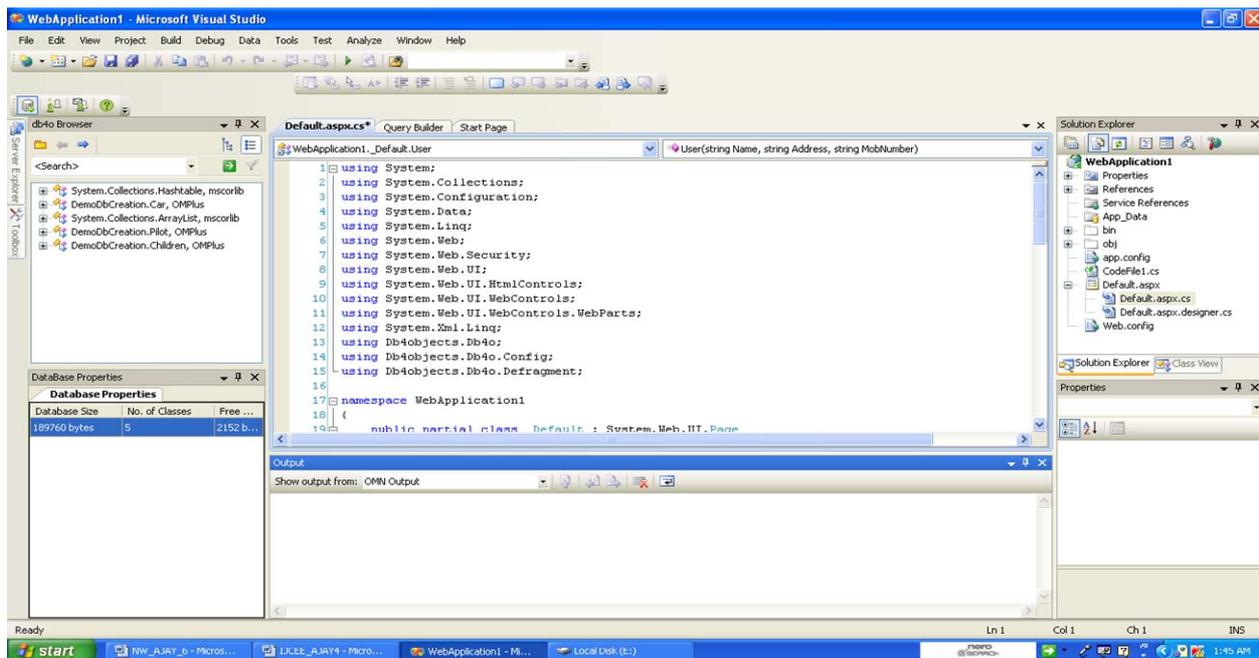


Figure 5. Storing objects in C# for Db4o.

are: Query By Example (QBE), Simple Object Data Access (SODA) and Native Queries (NQ). Authors have used native query for the querying purpose in this paper. The query is expressed as predicate in .NET language as the object template. The problem of impedance mismatch is completely removed when one can use native query in Db4o. Code for native query is given below:

```
try {
```

```
IObjContainer
db=Db4oFactory.OpenFile("User.yap");
IList<Class1>result= db.Query<Class1>(delegate(Class1
obj){
return obj.Name = "ajay" && obj.Address
=="Saharanpur";}); PrintResults(userList);
}
```

Figure 6 represents a use case diagram for the Cus-

tomter, which has three use-cases: Search Schemes, Validate Password and Make Payment. Specilization of general use-case “Search Scheme” is done into two child use-cases “Search by Category” and “Search by Amount Limit”. Similarly, specilization of general use case “Make Payment” is done in two specialized child use cases “Credit Card Mode” and “Net Banking Mode”.

**Figure 7** represents a use case diagram for the Admin, which has five use cases: Search Scheme, Add Scheme, Validate Password, Make Trade, Add Emp & Branches. Generalization of use-case “Add Scheme” is done into two child use-cases “Add Banking Services” and “Add Non-Banking Services”. Similarly, use case “Make Trade” has two specialized child use-cases “Credit Card Mode” and “Net Banking Mode”.

**4.2. UML Sequence Diagram**

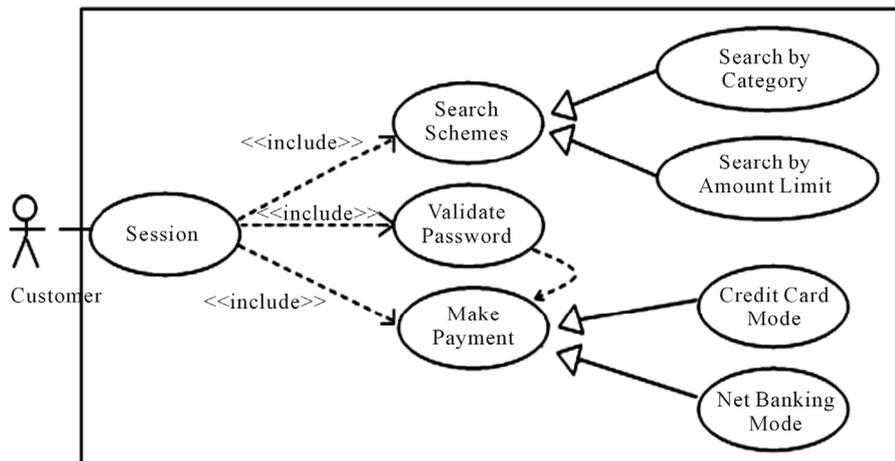
Sequence diagram shown in **Figure 8** has five objects Admin, Scheme\_Database, Office\_Database, Business\_

Layer and DataAccess\_Layer. In this diagram, the Admin adds a new scheme at Scheme\_Database.

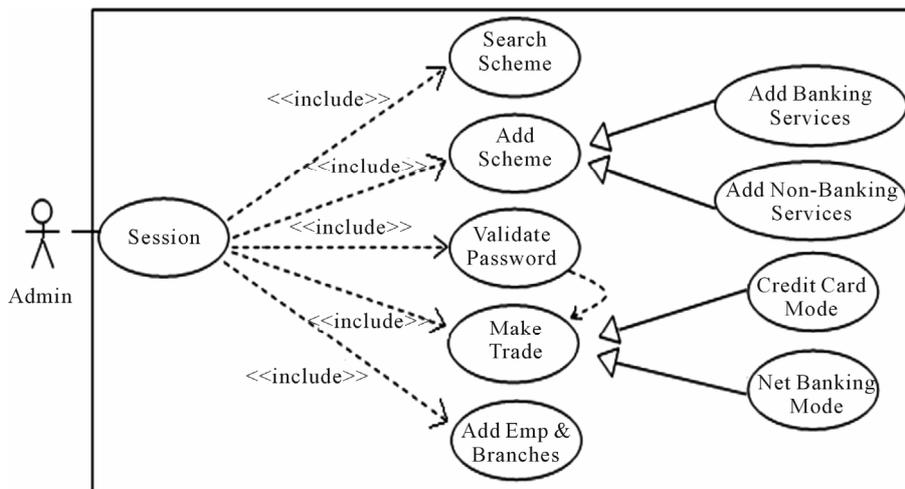
The request is being sent at Business Layer and PerformUpdate (SID, Qty) operation is performed at DataAccess Layer. The Admin can also add a new employees and branches in office database. After that the request is being sent at Business Layer and Perform Update( ) operation is performed at DataAccess Layer. The Admin can perform queries for the available schemes in the Scheme\_Database. The ComputeQuery( ) operation is performed at Business\_Layer and the RetrieveResult( ) operation is performed at DataAccess\_Layer. Finally, the result is passed to the Admin.

**4.3. UML Class Diagram**

The UML class diagram shows the static structural behaviour of the system, in which attributes and operations are designed for the complete system. **Figure 9** represents the class diagram of the online post office



**Figure 6. UML use-case diagram for administrator.**



**Figure 7. UML use-case diagram for Customer.**

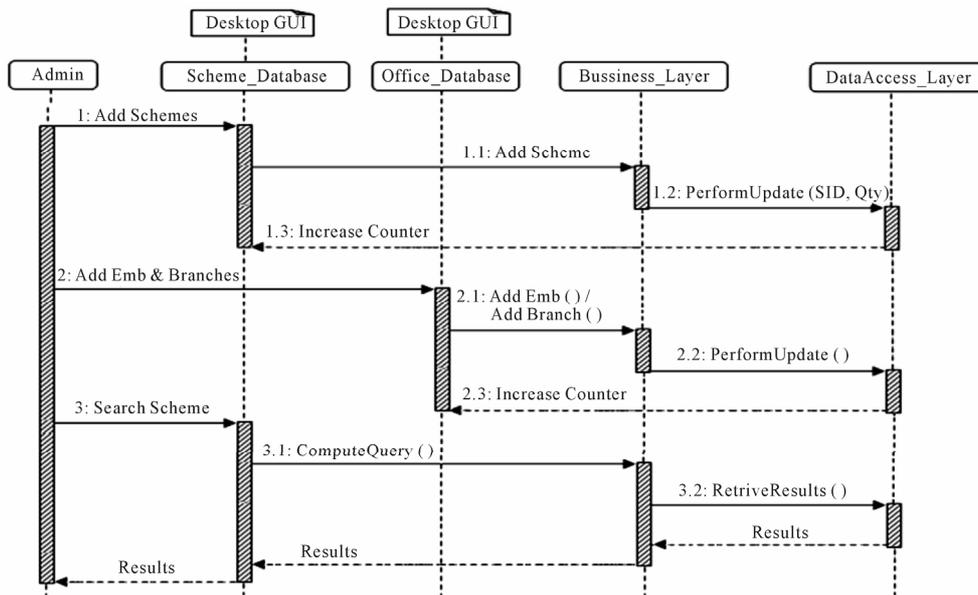


Figure 8. UML sequence diagram for administrator.

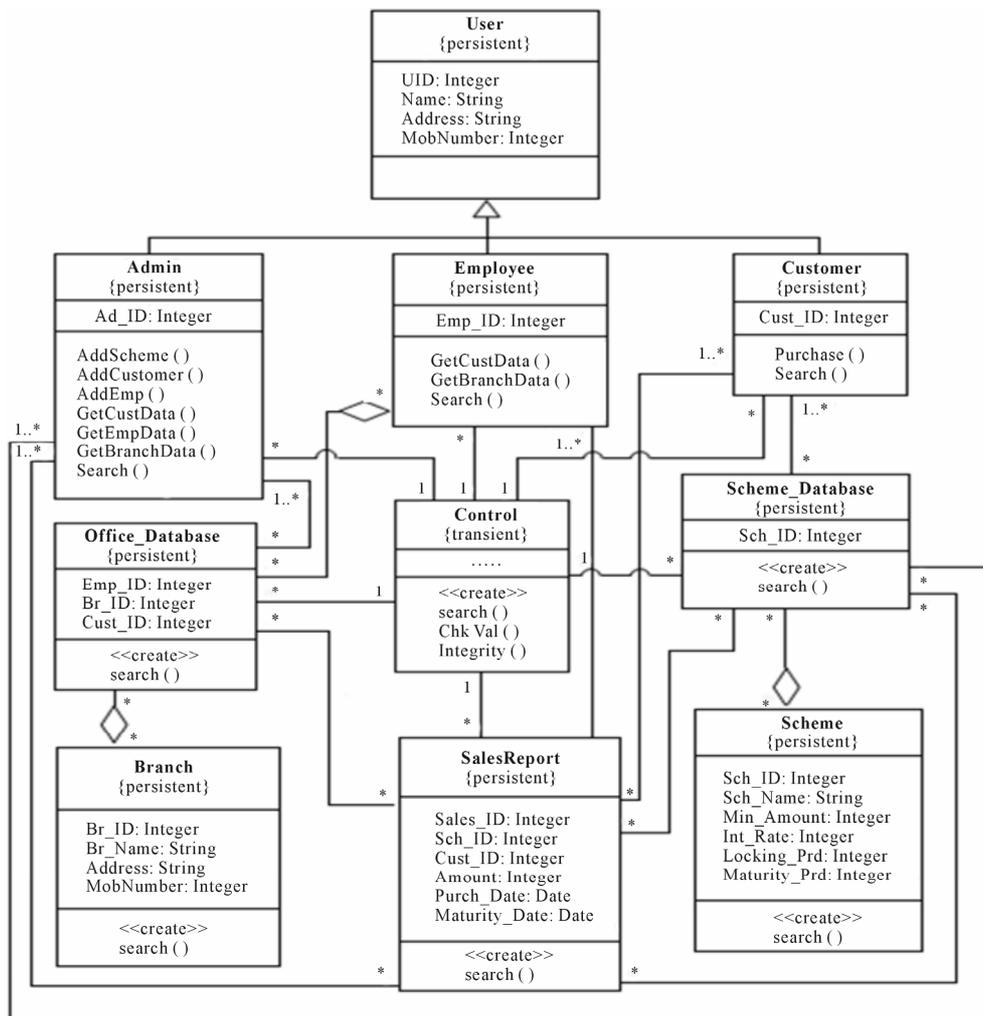


Figure 9. UML class diagram for the application.

system. The class diagram has nine persistent classes, which are User, Admin, Employee, Customer, Office\_Database, Scheme\_Database, Branch, Scheme and Sales Report and one transient class Control. Classes are connected to each other by various relationships with their multiplicities as shown in the diagram and the transient class Control is used to manage all persistent classes. Admin, Employee and Customer are inherited from non-abstract class User. Class Office\_Database consists of many branches and many employees. Aggregation between the classes Office\_Database, Branches and Employees are shown in the class diagram. Scheme\_Database also consists various types of schemes. So, aggregation is also shown between class Scheme\_Database and Scheme.

## 5. Conclusion

Major web applications are being developed using relational database management system (RDBMS) but enough time is consumed for relating the object model of .NET and database model. Object-oriented database system (OODBMS) is an alternative of RDBMS, which can be verified efficiently and easy to use. The development of web application for the post office is developed using Microsoft Studio IDE that takes the advantage of object-oriented database system Db4o. Further studies can be done to explore new capabilities and potentials of Db4o like data mining, clustering of object-oriented database, etc.

## REFERENCES

- [1] R. Elmasri and S. Navathe, "Fundamentals of Database Systems," 4th Edition, Addison Wesley, New York, 2003.
- [2] A. Silberschatz, H. Korth and S. Sudarshan, "Database System Concepts," 4th Edition, 2005.
- [3] J. Paterson, S. Edlich, H. Hoerning and R. Hoerning, "The Definitive Guide to Db4o," Apress, Berkeley, 2006.
- [4] Db4o Developer Community, 2011, <http://developer.db4o.com/>
- [5] Carl Rosenberger on Db4o, "Db4o Documentation," 2011. <http://developer.db4o.com/Blogs/Community/tabid/166/entryid/204/Default.aspx>
- [6] M. Rădulescu, "Object-Oriented Database Development Using Db4o," Master of Science Thesis, Babes-Bolyai University, Cluj-Napoca, 2009.
- [7] S. Danturthi, "Comparative Study of Web Application Development with SQL Server and Db4o," Master's Thesis in Computer Science, Mälardalen University, Västerås, 2011.
- [8] M. M. Liaw, V. Durbhakula and M. Molakalapalli, "Developing a Three Tier Application Using ASP.NET and Open Source Object-Oriented Database Db4Objects," *Proceedings of 46th Annual Conference of ACET*, League city, 7-9 October 2010, p. 15.
- [9] S. Bernardi, S. Donatelli and J. Merseguer, "From UML Sequence Diagrams and Statecharts to Analysable Petri Net Models," *Proceedings of the 3rd International Workshop on Software and Performance*, ACM Press, New York, 2002, pp. 35-45. [doi:10.1145/584369.584376](https://doi.org/10.1145/584369.584376)
- [10] J. Rumbaugh, I. Jacobson and G. Booch., "The Unified Modeling Language Reference Manual," 2nd Edition, Addison-Wesley, Boston, 2005.
- [11] U. A. Umoh, E. O. Nwachukwu, I. J. Eyoh and A. A. Umoh, "Object Oriented Database Management System: A UML Design Approach," *The Pacific Journal of Science and Technology*, Vol. 10, No. 2, 2009, pp. 355- 365.
- [12] P. K. Chaurasia and V. Saxena, "Mobile Based Electricity Bill Deposit System through UML," *Journal of Software Engineering and Applications*, Vol. 4, No. 3, 2011, pp. 187-190. [doi:10.4236/jsea.2011.43021](https://doi.org/10.4236/jsea.2011.43021)
- [13] L. V. Tagliati and C. Caloro, "UML and Object Oriented Drama," *Journal of Object Technology*, Vol. 7, No. 1, 2008, pp. 85-101. [doi:10.5381/jot.2008.7.1.a2](https://doi.org/10.5381/jot.2008.7.1.a2)
- [14] S. Mangla, "CodeProject, Beginners Introduction to ASP.NET," 2008. <http://www.codeproject.com/KB/aspnet/aspnetintro.aspx>