

## **Developing a Three Tier Web Application Using ASP.NET and Open Source Object-Oriented Database db4objects**

Morris M. Liaw Ph. D.  
Univ. of Houston Clear Lake  
liaw@uhcl.edu

Venkata Durbhakula  
Univ. of Houston Clear Lake  
dvijaykrishna@gmail.com

Mahesh Molakalapalli  
Univ. of Houston Clear Lake  
molakal9@gmail.com

### **Abstract**

This paper details the use of object-oriented database db4objects (db4o), ASP.NET 3.5, MVC architecture, and the Visual Studio IDE (Integrated Development Environment) to develop a three tier web application. The major contribution of this paper is to show how a three tier web application can be easily developed and deployed by taking advantage of an object-oriented database instead of using ADO.NET or Datasets that access traditional relational databases. A sample web application called the “Fee Management Tool” is developed to illustrate how a db4o database can be used as a back-end database and how the Visual Studio 2008 IDE can be used to run the db4o plug-in. The benefits and some concerns about db4o are also discussed at the end of the paper.

### **Keywords**

Object-Oriented Database, db4objects, ASP.NET, Visual Studio, Web Application

### **Introduction**

For developing a web application, the essential tools are the Integrated Development Environment (IDE) and Application Programming Interface (API). API libraries are used in abstracting common programming tasks (making database connections) required in developing a web application. In this paper, we show how Microsoft’s Visual Studio IDE can be used to develop a three tier web application that uses C#.NET classes to access a db4o back-end database. The paper includes discussions about connecting to db4o, writing and retrieving data, and configuring Microsoft Visual Studio to run a db4o toolkit.

Visual Studio is an integrated development environment (IDE) for the .NET environment and is developed by Microsoft<sup>1</sup>. It is capable of installing different kinds of plug-ins into its environment. Object Manager Enterprise (OME), a db4o plug-in for Visual Studio can

---

<sup>1</sup> [www.microsoft.com](http://www.microsoft.com)

be installed for browsing database objects as business entities. Visual Studio supports graphical user interface applications, windows applications, and many other applications. It has an integrated server called the Internet Information Services (IIS) which runs and builds the applications. Visual Studio supports development of web applications in both C#.NET and VB.NET and the latest F#. For this study, we develop our project using C#.NET.

db4o is an open source object database management system for .NET and Java developers, which was acquired by Versant Corporation<sup>2</sup> in December 2008. db4o has a strong “One-line-of-code” database feature where a single command is used to store an entire object into a database. For example, a single “*store*” command alone can insert a single record without any additional statements. Along with this, there are many other features like dynamic scheme evolution, native queries, LINQ, etc.

In this paper, we develop a web application for an existing manual system to illustrate the use of db4o. The “Fee Management Tool” is an online web application developed for schools and colleges to manage their fee processing. This project uses Microsoft Visual Studio to facilitate the building and deployment of a server-based application. Visual Studio is also configured with OME to run db4o. In this application the client and server classes were eliminated so that the application connects directly to the database, which is a more portable solution. A web application was partially implemented using this approach.

## **Preliminaries**

Most existing systems are developed with a relational database as a back-end database. Though relational databases, especially those developed by Oracle, are ruling the market, there are several reasons for selecting a db4o database as a back-end database (Grehan, 2005). In some respects, object databases are much better than relational databases. One reason to select a db4o database is that db4o databases are fully featured and can be embedded into devices like mobile phones and personal computers. Since database administration also plays a key role in most relational databases, a db4o database is especially advantageous because db4o databases do not require any kind of special administration. Additionally, db4o databases can also simplify refactoring in data warehouse applications. Lastly, most object-databases, especially those using db4o, can efficiently work on different platforms, and db4o specifically is available for .NET, Java, and Mono (Paterson et.al, 2006).

The Fee Management Tool is a comprehensive web based system that helps institutions and schools to manage funds. The Fee Management Tool handles all key fee management functions, including the collection of student information, handling of funds by the administration, and management of student relations.

The tool is a system designed primarily to alleviate financial losses for institutions by reducing risk of loss. The aim of this project was to develop an online system for an

---

<sup>2</sup> [www.versant.com](http://www.versant.com)

institution or school. Generally, an institution manages the use of funds collected from students, and an accountant for the institution manages the information about fees collected from the students.

The figures below show sample screen shots of the system. Figure 1 shows the Home page of the Fee Management Tool while Figure 2 shows the “Add Student” page, which is used by administrators to enter in details about new students. A sample flow chart of the project is depicted in Figure 3. The online project is available at

<http://sce.uhcl.edu/db4o/Main.aspx>.



Figure 1: Home page of the System

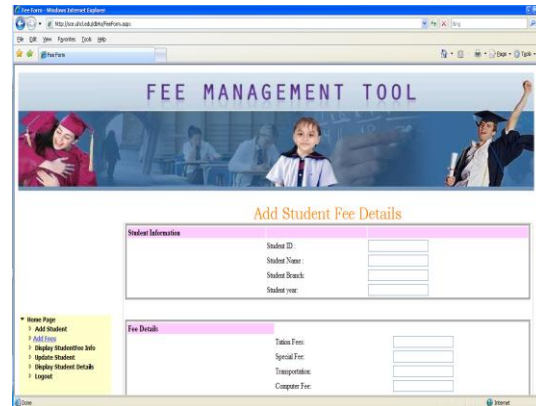


Figure 2: Add Student page

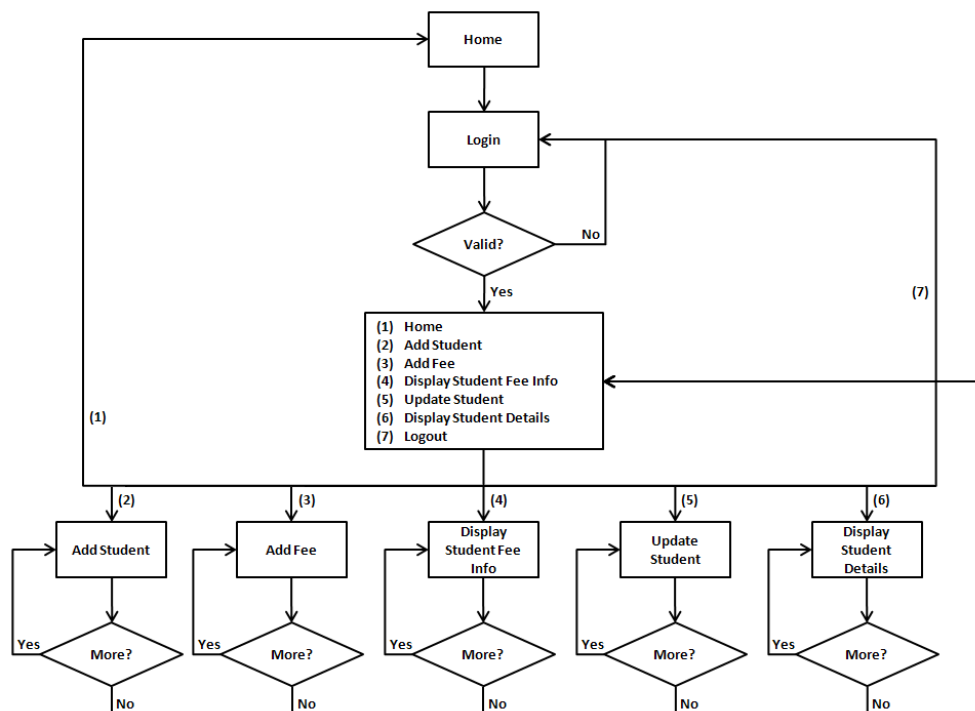


Figure 3: Flow-Chart showing actions available in the System

## Using db4o

db4o is an object oriented database system that can use simple commands to store/retrieve objects in a database. Commands like SET, STORE, and GET make the task of database transactions easy when compared with traditional relational databases (Leone and Chen, 2007). These commands are under the namespace of db4objects.db4o.

To insert an object into the Fee Management Tool's database we use a simple STORE command with a parameter as the required object. The following is the syntax for inserting a Student object into the database using the STORE command.

```
Student studentobject = new Student();  
db.Store(studentobject);
```

To retrieve an object from the database we use a simple GET command with a parameter as the required object. The following is the syntax for retrieving a Fee object from the database using the GET command. The result from the GET command is stored in IObjectSet and can be fetched to a table when needed.

```
Fee feeobject = new Fee();  
IObjectSet reslut = db.Get(feeobject);
```

To update an object, the object needs to be retrieved before the required update can be made by a db.Set( ) command. The following is the syntax for updating a Student object using the SET command.

```
Student studentobject = new Student();  
db.Set(studentobject);
```

The db.Close( ) command is used to close the object container. db.Close( ) finishes all database operations to be written to the database before closing the database.

```
db.Close();
```

The db.Delete( ) command is used to delete a retrieved object. As was the case for updating, the database must know which object is intended for deletion so the object to be deleted must either be set, or more likely retrieved, within the same transaction. The following is the syntax for deleting a Student object using the DELETE command.

```
Student studentobject = new Student();  
db.Delete(studentobject);
```

## **Benefits of db4o**

db4o is not only a zero-administration database but a full-featured, embeddable database engine for devices; for mobile, desktop, and server platforms in object-oriented environments; and for applications with multi-media content or complex data structures. It eliminates the need for tools and code that are required for object-relational mapping and offers a high level of cross platform support, including support for mobile platforms (Paterson et.al, 2006).

## **Concerns on db4o**

There are some concerns with db4o. If an older system is already relational-based, then the system may need to be redeveloped using an object-oriented approach with db4o. Also, db4o cannot support database clustering, data warehousing, data mining, or data tuning, and reporting tools have yet to be designed for it. Moreover, transaction support and recovery are not as comprehensive for db4o as they may be for other relational database management systems (Paterson et.al, 2006).

## **Conclusion**

The development of the Fee Management Tool application shows that the Microsoft Visual Studio IDE can be used to develop standard, portable web applications that take advantage of an object-oriented database. We have further shown that communication with a db4o database can be accomplished using .NET classes instead of routine libraries which are used to access traditional relational databases. Future studies will explore the deep potential that object-oriented databases have, especially with db4o.

## **References**

Grehan, R. "Complex Object Structures, Persistence and db4o." *Series of db4o whitepaper, db4objects Inc* (2005). Print

Leone, A, and Chen D. "Implementation of an Object Oriented Data Model in an Information System for Water Catchment Management: Java JDO and db4o Object Database." *Environmental Modeling & Software* 22 12 (2007): 1805-10. Print.

Paterson, J, et al. "*The Definitive Guide to db4o.*" Apress, Berkeley, CA. (2006). Print.