

Online Recruitment System for BINUS Career (June 2011)

Gusyanto, Hendry Lie, and Efendi

Abstract— The application that we made is the improvement from previous BINUS Career's website that has more features. It needs improvement because the previous one didn't have online recruitment system. The previous one only Received Application which application that job seeker is the only one that has to apply his/her CV to the company he/she wants to work. In this new system that we made, we add several new features. This online recruitment system's goal is to make employers can choose which job seeker they want to recruit.

I. INTRODUCTION

THIS project is made to fulfill the requirement graduation for Internship course subject on our sixth semester in Bina Nusantara University. The internship took place at IT Directorate Bina Nusantara. Project that we made here is an ONLINE RECRUITMENT SYSTEM FOR BINUS CAREER. This project is to make BINUS Career's website implement online recruitment system, but still has the same user interface like the previous one. Like the previous BINUS Career's website, we only add some menu. Candidate Match, Request CV Mining, and CV Mining Candidate are some new menus that we add to BINUS Career's new website. We built this website by using ASP.Net and C# programming language.

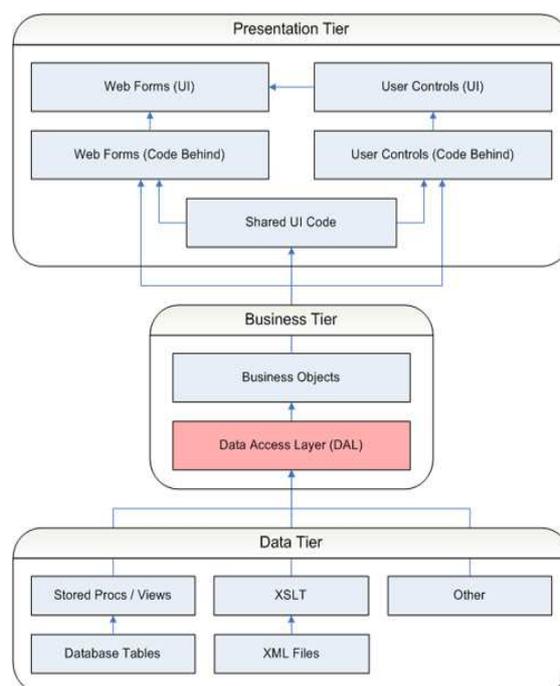
We also add Telemarketing site for BINUS Career's Administrator. This website is to provide Administrator to approve or reject the CV that employer requested. If the request is rejected, administrators have to fill the notes. Besides that, we also can see the CV Mining history, employer profile, and audit activity.

Gusyanto is with IT Directorate of Bina Nusantara, Jakarta (corresponding author to provide phone: 085959593792; e-mail: gusyanto@binus.edu).

Hendry Lie is with IT Directorate of Bina Nusantara, Jakarta (corresponding author to provide phone: 081806127017; e-mail: hendrylie1989@yahoo.com).

Efendi is with IT Directorate of Bina Nusantara, Jakarta (corresponding author to provide phone: 08988252488; e-mail: j.fndlee@gmail.com).

II. ARCHITECTURE AND DESIGN



Layered application designs are extremely popular because they increase application performance, scalability, flexibility, code reuse, and have a myriad of other benefits that I could rattle off if I had all of the architectural buzzwords memorized [1]. In the classic three tier design, applications break down into three major areas of functionality:

- The data layer manages the physical storage and retrieval of data
- The business layer maintains business rules and logic
- The presentation layer houses the user interface and related presentation code.

The presentation tier

In the presentation layer, the code-behind mechanism for ASP.NET pages and user controls is a prominent example of a layered design [2]. The markup file defines the look and layout of the web form and the code behind file contains the

presentation logic. It's a clean separation because both the markup and the code-behind layers house specific sets of functionality that benefit from being apart. Designers don't have to worry about messing up code to make user interface changes, and developers don't have to worry about sifting through the user-interface to update code.

The data tier

You also see sub-layers in the data tier with database systems [2]. Tables define the physical storage of data in a database, but stored procedures and views allow you to manipulate data as it goes into and out of those tables. Say, for example, you need to denormalize a table and therefore have to change its physical storage structure. If you access tables directly in the business layer, then you are forced to update your business tier to account for the changes to the table. If you use a layer of stored procedures and views to access the data, then you can expose the same logical structure by updating a view or stored procedure to account for the physical change without having to touch any code in your business layer. When used appropriately, a layered design can lessen the overall impact of changes to the application.

The business tier

And of course, this brings us to the topic of business objects and the Data Access Layer (also known as the DAL), two sub-layers within the business tier. A business object is a component that encapsulates the data and business processing logic for a particular business entity. It is not, however, a persistent storage mechanism. Since business objects cannot store data indefinitely, the business tier relies on the data tier for long term data storage and retrieval. Thus, your business tier contains logic for retrieving persistent data from the data-tier and placing it into business objects and, conversely, logic that persists data from business objects into the data tier. This is called data access logic.

This system uses Microsoft Visual Studio 2003, and based on ASP.NET and C# programming language, using Microsoft SQL Server 2005 for database.

IV. CONCLUSION

According to analysis results above, we can conclude:

1. The new BINUS Career's website can provide feature which allow employer to

choose any job seeker from BINUS Career's database.

2. The new BINUS Career's website will use a different system of membership.
3. Administrator can approve and reject CV which requested by employer

We use database SQL Server 2005 because in previous BINUS Career's website is also using SQL Server, so it can be easier for us to integrate this system.

REFERENCES

- [1] Stehling, Brennan. *Pro ASP.NET for SQL Server: High Performance Data Access For Web Developers*. New York: Apress, 2007.
- [2] Armstrong, Damon. .NET Application Architecture: the Data Access Layer [Online]. Available: <http://www.simple-talk.com/dotnet/.net-framework/.net-application-architecture-the-data-access-layer/>