TEACHER CREDENTIALING IN THE STATE OF CALIFORNIA

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Hitesh Chaudhary
Summer 2013
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Hitesh Chaudhary:

Teacher Credentialing in the State of California

Joseph Lewis, Chair
Department of Computer Science

Carl Eckberg
Department of Computer Science

Mary Ann Lyman-Hagar
Department of European Studies

5/12/2013
Approval Date
Copyright © 2013
by
Hitesh Chaudhary
All Rights Reserved
DEDICATION

I dedicate this thesis work to my parents and elder brothers for their continuous encouragement. This would not have been possible without their unconditional love and support.
ABSTRACT OF THE THESIS

Teacher Credentialing in the State of California

by

Hitesh Chaudhary

Master of Science in Computer Science

San Diego State University, 2013

In last five decades schools and universities in the United States have attracted thousands of students from across the globe; as the inflow of students has increased tremendously, there is an increased demand for teachers who can teach subjects such as math and science, as well as foreign languages like Arabic, French, Spanish, Punjabi, Farsi, etc.

To teach these subjects, schools and universities require credentialed. Unfortunately, the system of teacher credentialing is over complicated in the United States; each state has different requirements for credentialing teachers for teaching a particular subject. Moreover there is hardly any single source over the internet that can help these prospective teachers with their diverse questions on requirements, exams, fees, jobs etc.

This thesis project is dedicated to providing all the necessary information to these prospective teachers for the state of California, through a full-featured web application consisting of blogs, a job portal, resources and social networking features. Microsoft .Net technology is used for development and implementation of this project.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>.................................................................</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>.................................................................</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>.................................................................</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>.................................................................</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>.................................................................</td>
</tr>
<tr>
<td>2 BACKGROUND AND LITERATURE</td>
<td>.................................................................</td>
</tr>
<tr>
<td>2.1 Teacher Credentialing</td>
<td>.................................................................</td>
</tr>
<tr>
<td>2.1.1 CTC</td>
<td>.................................................................</td>
</tr>
<tr>
<td>2.1.2 NCSSFL</td>
<td>.................................................................</td>
</tr>
<tr>
<td>2.1.3 Certification Map</td>
<td>.................................................................</td>
</tr>
<tr>
<td>2.2 Challenges and Approach</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3 TECHNOLOGIES</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.1 Microsoft ASP.NET 4.0</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.1.1 Comparisons of ASP.NET 4.0 with Previous Versions</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.1.2 Benefits of Using ASP.NET Framework</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.2 C#</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.2.1 Advantages of C# Over Other Languages</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.2.2 Advantages Over Java</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.3 MySQL</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.3.1 Features of MySQL</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.3.2 Advantages of MySQL</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.4 Html and CSS</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.4.1 HTML</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.4.2 CSS</td>
<td>.................................................................</td>
</tr>
<tr>
<td>3.5 Teleriks</td>
<td>.................................................................</td>
</tr>
<tr>
<td>4 APPLICATION ARCHITECTURE</td>
<td>.................................................................</td>
</tr>
</tbody>
</table>
4.1 User Roles ........................................................................................................20
  4.1.1 Guest User ...............................................................................................20
  4.1.2 Administrator ..........................................................................................20
  4.1.3 Employer .................................................................................................21
  4.1.4 Teacher ....................................................................................................22
  4.1.5 Student ....................................................................................................22

4.2 Basic Architecture ............................................................................................22
  4.2.1 Presentation Layer ..................................................................................22
  4.2.2 Business Layer ........................................................................................22
  4.2.3 Data Layer ...............................................................................................23
  4.2.4 MVC – 3 Tier Architecture .....................................................................23

4.3 Application Modules ........................................................................................24
  4.3.1 Login and Registration Module ..............................................................24
  4.3.2 Blog Module ...........................................................................................24
  4.3.3 Job Portal Module ...................................................................................25

4.4 Social Networking ...........................................................................................25

4.5 Google Translation...........................................................................................25

5 APPLICATION DESIGN IMPLEMENTATION AND STYLING .........................26
  5.1 System Architecture .........................................................................................26
  5.2 Database Design ...............................................................................................26
    5.2.1 User Data ................................................................................................27
    5.2.2 Blogs Data ...............................................................................................27
    5.2.3 Jobs Data .................................................................................................27
    5.2.4 Credentialing Data ..................................................................................28

  5.3 Design Flow .....................................................................................................29
    5.3.1 Create Blog .............................................................................................30
    5.3.2 Manage Blog ...........................................................................................31
    5.3.3 Create Job ................................................................................................33
    5.3.4 Manage Job .............................................................................................33

  5.4 Application Logic ............................................................................................37
    5.4.1 Database Logic ........................................................................................37
    5.4.2 Blogging Logic .......................................................................................38
5.4.3 Job Portal Logic .................................................................38
5.4.4 Facebook API .................................................................39
5.4.5 Google Translation API ..................................................39
5.5 User Interface Implementation ...........................................40
  5.5.1 Blogs .....................................................................40
  5.5.2 Job Portal ...............................................................42
  5.5.3 Resources ...............................................................45

6 CONFIGURATION TESTING AND CONCLUSION .........................46
  6.1 Configuration ..................................................................46
  6.2 Deployment ...................................................................46
  6.3 Testing .........................................................................46
  6.4 Summary and Conclusions .............................................47

7 FUTURE ENHANCEMENT AND SCOPE .....................................48

REFERENCES ............................................................................49

APPENDIX
  GLOSSARY .............................................................................51
LIST OF TABLES

Table A.1. Abbreviations.........................................................................................................52
LIST OF FIGURES

Figure 2.1. CTC. ........................................................................................................................6
Figure 2.2. NCSSFL. ..................................................................................................................7
Figure 2.3. Certification map ....................................................................................................8
Figure 3.1. Architecture of ASP.NET. ......................................................................................11
Figure 3.2. Architecture of MySQL. .........................................................................................16
Figure 4.1. UML Diagram of user roles and their privilidges ..................................................21
Figure 4.2. A typical collaboration of the MVC components. ..................................................23
Figure 5.1 User data EER-1. ....................................................................................................28
Figure 5.2. Blog data EER-2. .................................................................................................29
Figure 5.3. Jobs data EER-3. .................................................................................................30
Figure 5.4. Credentialing data EER-4. ...................................................................................31
Figure 5.5. Create blog flowchart. ..........................................................................................32
Figure 5.6. Manage blog flowchart. ........................................................................................34
Figure 5.7. Create jobs flowchart. ...........................................................................................35
Figure 5.8. Manage job flowchart. .........................................................................................36
Figure 5.9. Blog - View all blogs. ..........................................................................................41
Figure 5.10. Blog - Blog details. ............................................................................................42
Figure 5.11. Job portal - View all jobs. ...................................................................................43
Figure 5.12. Job portal - Job details. .......................................................................................44
Figure 5.13. Job portal - View profile. ...................................................................................45
Figure 5.14. Job portal - View applicants. ............................................................................44
Figure 5.15. Resources. ............................................................................................................45
ACKNOWLEDGEMENTS

I would like to thank Dr. Mary-Ann Lyman Hager, Director of the National Language Resource Center for taking her precious time out of her busy schedule and guiding me throughout this project. I would also like to take this opportunity to thank Dr. Carl Eckberg and Dr. Joseph Lewis of SDSU for becoming part of the thesis committee and for providing support and encouragement to achieve the results in this thesis. A special thanks to Mr. Norman Leonard of LARC for introducing this thesis topic.
CHAPTER 1

INTRODUCTION

“Teaching is a profession that can yield amazing results when the right ideas and techniques are used in the classroom. I believe that the purpose of teaching is not to teach students how to memorize facts, or how to know all the correct answers. The purpose of teaching lies in getting students to truly understand the concepts being examined. I view teaching as a process of encouraging students to make connections between their real world experiences and the subject being studied” [1].

What makes a great teacher? Teaching is one of the most complicated jobs today. It demands broad knowledge of subject matter, curriculum, and standards; enthusiasm, a caring attitude, and a love of learning; knowledge of discipline and classroom management techniques; and a desire to make a difference in the lives of young people. With all these qualities required, it's no wonder that it's hard to find great teachers [2].

What is a Teaching Credential? A teaching credential is a certification which indicates that someone is allowed to teach a particular subject or grade level. There are numerous types of teaching credentials, ranging from those designed for school nurses to special education credentials. Using a credentialing system for teachers ensures that all students have access to qualified teachers who are familiar with the subject and teaching techniques. The path to obtaining a teaching credential can be quite lengthy, depending on the credential being sought [3].

A United States teaching credential is a basic multiple or single subject credential obtained upon completion of a bachelor's degree and prescribed professional education requirements. Teaching credentials are required in the United States in order to qualify to teach in public school, as well as many other types of institutions. Requirements vary from state to state. Teachers in California must also pass the California Basic Educational Skills Test in order to teach in a school [4].

The requirements for teaching certification are set by each state in the United States. All states require a teacher to have, at minimum, a bachelor’s degree from an accredited
college or university, but other certification requirements vary greatly from state to state. Most states require completion of a teacher education program and a state certification exam or the Praxis exam.

Teacher certification reciprocity is an agreement that allows an educator who is certified to teach in one state to transfer his or her certification and teach in a different state. More than 40 states have some type of reciprocity policy in place. In many cases, the certification transfer is temporary. Once a time limit has been reached, a teacher will need to complete additional requirements, such as classroom experience or coursework, in order to be officially certified in the new state [5].

Some states offer alternative licensure programs for teachers who do not have the experience required for a traditional license. Teachers in these programs typically have their Bachelor’s degree, but may have not taken the necessary education courses to pursue standard teacher certification. Teachers in these programs work for about one or two years before they receive their teaching license. Alternative licensure programs are in place to compensate for shortages of teachers in certain subjects or to qualify more people to teach in high needs areas that have difficulty attracting and retaining teachers [6].

In spite of all these measures taken by states the number of qualified teachers is not increasing rapidly. There can be many reasons for this, but by far I see the following three major reasons to be the cause of this slow growth in number of qualified teachers in the whole of the United States.

First, the current system is over complicated for a teacher to understand what routes, subjects, and exams he/she needs to take in order to teach a particular subject in a particular state. This is further complicated by some states which do not accept the teaching credentials of other states. Also different states have different categories like single subject, multiple subjects, in-state, out of state, Peace Corps and out of the United States. Further all States have different requirements for each of the categories mentioned above.

Second, although the process of becoming a teacher and teaching a particular subject in a given state is over complicated in the United States, this can be made simpler by providing most of the information regarding teaching in a particular state online through a web application. There are close to seventy websites that provide information on teaching credentials requirements in a particular state, but unfortunately none of the websites have
complete information on requirements, scholarships, fees, courses, routes etc. So lack of information that is available online forces these teachers to personally visit or call the school where they want to teach or sometime even drop their plan to teach at that school.

Third, lack of knowledge about future prospects and jobs is another reason why many prospective teachers do not want to go for this discipline; rather, they want to go for degrees in business and computer science in which they can easily get a job, which also affects the number of teachers getting credentialed to teach in a particular state.

This project will serve as one of the solutions for this problem; in fact this project will provide a twofold solution to the problem.

One, through the medium of the latest Web 2.0 application and robust tools like social networking, blogging, Wiki, Job Portals and Information brochures, we would be able to bring more awareness among prospective teachers who want to pursue their careers in this field by providing them enough information through our website about the various doubts that they may have regarding a career in this field. At the same time they will be getting very useful and authentic information from current teachers through the medium of wiki and social networking which is one of the many other features of our website. Another important feature of the website is a job portal, which will provide a common platform to these teachers and their prospective employers.

Second, this website will also be very beneficial for attracting prospective students as well as already trained teachers who are looking to teach in schools and colleges in different states in the United States. They will be able to access the vital information on prerequisites for teaching in a particular state i.e. teaching credentials requirements, scholarships, bridge courses, etc. Other than that there will be ample information from experts and current teachers on various faq’s through wiki and blogs on such topics. Furthermore their interest will be stimulated by showing some of highly paid jobs available for them in public and private sectors companies.

To sum up, this website and its powerful tools will help to attract more qualified teachers to get credentialed and teach at various school, colleges and other educational institutions across the United States.
CHAPTER 2

BACKGROUND AND LITERATURE

A teaching credential is a certification which indicates that someone is allowed to teach a particular subject or grade level. There are numerous types of teaching credentials, ranging from those designed for school nurses to special education credentials. Using a credentialing system for teachers ensures that all students have access to qualified teachers who are familiar with the subject and teaching techniques. The path to obtaining a teaching credential can be quite lengthy, depending on the credential being sought.

Each state of the United States has varying teacher credentialing requirements. Someone who is interested in becoming a teacher should contact their local department of education to find out what the requirements are. Typically, a teaching credential is issued by the state in which the teacher lives and works and some states may have agreements with each other so that teachers can transfer existing teaching credentials [3].

As mentioned in the previous chapter, the education system in the United States is being affected by two major factors, the first being a declining number of qualified teachers who are credentialed to teach in a particular state. Second, the over complicated system of getting credentialed to teach in a state is not helpful. These two factors are residues of several problems that teachers are facing today. One of them is finding jobs; most of the jobs are filled using traditional methods like the walk in interview, referrals, or advertisement in local newspapers, which result in fewer applicants applying for that job and making it less selective. Another problem is finding resources or information to get credentialed. This problem has caused lot of prospective teachers to quit this profession, since they think the process of getting credentialed is too complex and their only source to get information is a state educational board.

In this chapter we will discuss how this thesis project will provides a solution to the problems mentioned above and we will also compare the features of this thesis project with existing websites that provide information on teacher credentialing.
2.1 TEACHER CREDENTIALING

In the process of developing this project and researching on the topic of teacher credentialing, I came across at least a dozen websites that provide teaching credential information for all states in the United States. Some of these websites have links that just redirect to a state education board website and some do not have any social networking features, which make them less popular or visited among prospective teachers.

I have tried to include all good features of the existing websites and added new features that I felt were missing. With a robust and scalable database design, fluid user interface and other features including blogs, job portal, and language translation, my thesis project provides a fairly complete solution to the problems mentioned earlier in this chapter.

The following sections discuss a few websites that provide some useful features that provide a partial solution to the problems discussed earlier in this chapter. We will briefly discuss the features of these websites and evaluate what they are lacking to provide a complete solution.

2.1.1 CTC

CTC stands for Commission on Teacher Credentialing. This website primarily provides teacher credentialing information only for the state of California. Figure 2.1 [7] shows a snapshot of the websites resource page for prospective teachers. This website has, overall, very good resources for teachers in a systematic pdf format, which minimizes the hassle to search multiple resources for a single thing. For example out of state applicants can find all information on requirements, fees, scholarships exams etc., in a single pdf document. But there are a few features that are missing on this website. This website does not have any social networking features, which makes its less popular among internet audience, plus it does not have a blog and job portal feature which limits its interaction with the user for just a resource. For my thesis project, I have adopted a similar idea to show resources in a pdf. One improvement that I made on this part is that instead of showing all resources on one page and causing confusion among users, I made a full text search on a state and a search term, which will yield only relevant results.
2.1.2 NCSSFL

NCSSFL stands for National Council for State Supervisors for Languages. Figure 2.2 [8] shows the resources link page on the website. Although the website has a good design and usability, but the resource links actually redirect to each state education board website. This website also allows users to view state reports, and a user can select a question and see its answer for all states. This is one of the few websites that provide teacher credentialing information nationwide. This website also has a member section, where in which you can register and download states reports or view credentialing information for any state. But this website does still not provide a complete package, since it is missing blogging and social networking features. Another important functionality missing here is a job portal that will provide a common platform for employers and teachers to communicate with each other. Clearly, another limitation is the restriction to language education.
2.1.3 Certification Map

The Certification Map is one of the websites that provide teacher credentialing information nationwide. The Certification Map is a unique website that provides map based teacher credentialing, which is very interactive and easy to use. A user just need to select a state from the map and all relevant teachers credentialing information for the selected state will be shown on the page. Figure 2.3 [9] shows an example where the user has selected the state as California and all relevant teachers credentialing information is shown on the page. Again, this website has same shortcomings similar to NCSSFL. This website does not provide any language translation on the website. This website has no job portal to help thousands of teachers looking for jobs. The Certification Map has no feature to register on the website, so it is just a resource website providing teacher credentialing information. I really liked the map based credentialing and have incorporated a similar feature in my thesis project.
2.2 CHALLENGES AND APPROACH

Teacher credentialing is a very wide and complex topic; as mentioned earlier, there is an increased demand for qualified and credentialed teachers in schools and other educational institutions across all states in the United States. There are several factors that contribute to the slow growth in the number of qualified teachers. One of the major reasons is the complexity of the system; in order for a teacher to teach in the United States he/she must have a bachelor’s degree, and additionally they have to get teaching credential for the subject that they want to teach. Furthermore the teaching credential for a subject is different for approximately all states in the United States. This means that a well-qualified, credentialed teacher teaching in a particular subject in a state may not meet the requirement to teach the same subject in a different state in the United States. Currently there are very few states that follow reciprocity [6], which means they recognize the teaching credential acquired in other state for teaching a particular subject.
The goal of this project is to provide all the information and resources that these prospective teachers need and also to provide them with other tools, such as social networking, blogging, and job portal. While developing this project I faced a number of challenges, some of which were technical relating to the technology that I am using for this project and others were non-technical relating to information gathering for teacher credentialing. I also had to change the database of my application from MySQL server 2008 to MySQL 5 when it was half completed, in order to meet LARC’s technology requirements.

A website that simply provides information and resources for teacher credentialing would not be enough to attract all the teachers, so I also wanted to add few more tools such as blogging, job portal, and social networking features such as Facebook. I also felt the need to have language translation on the website to help teachers from foreign countries, to help them better understand the content of the website in their own language. To add all these useful features to the website was obviously a big challenge.

To approach the challenges, I started with developing the database, since for blogs, job portal, and user personal and job, profile information had to be stored in an efficient way. I designed the database to be highly normalized and scalable by storing data in different tables and then accessing the data using primary and foreign key relationship. Once the initial database development was complete, I started developing the user Interface. I made sure that the user interface is functional, attractive, and easy to use.

I added the blogging feature to the website realizing the importance of mutual collaboration between current and prospective teachers. Job portal was another feature that I added to the website after doing lot of research on top job portal websites. This feature would provide teachers and employers a common platform in which to meet each other. Language translation was on the website was accomplished using Google translation API.

The most important part was to gather information and resources for teacher credentialing in the state of California. I got the best information and resources by researching on other teacher credentialing and state education board websites. This would have not been possible without the help of talented LARC staff. I received all the necessary help and support from Dr. Mary Ann Lyman-Hager, Mr. Norman Leonard, and Miss Ashley Wynne to develop this project.
CHAPTER 3

TECHNOLOGIES

This chapter describes the technologies used to implement this application. A brief comparison of these technologies with other well known technology platforms will be shown in this chapter. We will also discuss the reasons for making a choice of these technologies over others.

3.1 MICROSOFT ASP.NET 4.0

The ASP.NET Framework is a software framework that runs primarily on Microsoft Windows. It includes a large library and supports several programming languages like C#, VB, C++, J# etc. which allows language interoperability (each language can use code written in other languages) [5].

The .NET library is available to all the programming languages that .NET supports. Programs written for the .NET Framework execute in a software environment (as contrasted to hardware environment), known as the Common Language Runtime (CLR), an application virtual machine that provides important services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework [5].

The .NET Framework's Base Class Library provides user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. Programmers produce software by combining their own source code with the .NET Framework and other libraries [5].

The .NET Framework Architecture as shown in Figure 3.1 [10], is intended to be used by most new applications created for the Windows platform. Microsoft also produces a popular integrated development environment largely for .NET software called Visual Studio [5].

The .Net Frameworks allows developers to create rich, powerful, database driven Web and Windows applications.
3.1.1 Comparisons of ASP.NET 4.0 with Previous Versions

ASP.NET 1.0: This was the first version of ASP.NET released together with Visual Studio .NET on 16 January, 2002. Following features were new to this version [6]:

- Object-oriented Web application development supporting inheritance, polymorphism and other standard OOP features
- Developers are no longer forced to use Server.CreateObject (...), so early-binding and type safety are possible.
- Based on Windows programming, the developer can make use of DLL class libraries and other features of the Web server to build more robust applications that do more than simply rendering HTML (e.g. exception handling).

ASP.NET 1.1: This version was released together with Windows Server 2003 and Visual Studio .Net 2003 on April 24, 2003. Following features were new to this version:

- Mobile controls
- Automatic input validation

ASP.NET 2.0: This version was released together with Visual Studio 2005 and SQL Server 2005 on November 7, 2005. Following features were new to this version [6]:

- New data controls (Grid View, Form View, Details View)
- New technique for declarative data access (SqlDataSource, ObjectDataSource, XmlDataSource controls)
- Navigation controls
- Master pages, Themes, Skins
- Login controls
- Web parts
- Personalization services
- Full pre-compilation
- Provider class model

**ASP.NET 3.0:** This version was released on November 21, 2006. Following features were new to this version [6]:

- Windows Presentation Foundation (WPF)
- Windows Workflow Foundation (WF)
- Windows Communication Foundation which can use ASP.NET to host services.
- Windows Card Space which uses ASP.NET for login roles.

**ASP.NET 3.5:** This version was released with Visual Studio 2008 and Windows Server 2008 on November 19, 2007. Following features were new to this version [6]:

- New data controls (List View, Data Pager)
- ASP.NET AJAX included as part of the framework
- Support for HTTP pipelining and syndication feeds.
- WCF support for RSS, JSON, POX and Partial Trust
- All the .NET Framework 3.5 changes, like LINQ etc.

**ASP.NET 4.0:** This version was released together with Visual Studio 2010 on April 12, 2010. Following features were new to this version [6]:

- Routing was introduced.
- Chart control was introduced.
- Better validation model was supported.

**ASP.NET 4.5:** This is still under development [6].

### 3.1.2 Benefits of Using ASP.NET Framework

According to Microsoft, ASP.NET offers following advantages over other Web development models [11]:

- ASP.NET reduces the amount of code required to build large applications.
• With built-in Windows authentication and per-application configuration, applications are safe and secured.

• It provides better performance by taking advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box.

• The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.

• Provides simplicity as ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration.

• The source code and HTML are together therefore ASP.NET pages are easy to maintain and write. Also the source code is executed on the server. This provides a lot of power and flexibility to the WebPages.

• All the processes are closely monitored and managed by the ASP.NET runtime, so that if process is dead, a new process can be created in its place, which helps keep your application constantly available to handle requests.

• It is purely server-side technology so, ASP.NET code executes on the server before it is sent to the browser.

• Being language-independent, it allows one to choose the language that best applies to a specific application or partition an application across many languages.

• ASP.NET makes for easy deployment. There is no need to register components because the configuration information is built-in.

• The Web server continuously monitors the pages, components and applications running on it. If it notices any memory leaks, infinite loops, other illegal activities, it immediately destroys those activities and restarts itself.

• Easily works with ADO.NET using data-binding and page formatting features. It is an application which runs faster and counters large volumes of users without having performance problems [11].

3.2 C#

In the words of Microsoft, “C# (pronounced as C sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object (class-based), and component-oriented programming disciplines”. It was developed by Microsoft within its .NET initiative and later approved as a standard by (ECMA). C# is one of the programming languages designed for the Common Language Infrastructure.
C# is intended to be a simple, modern, general-purpose, object-oriented programming language. Its development team is led by Anders Hejlsberg. The most recent version is C# 5.0, which was released on August 15, 2012 [12].

### 3.2.1 Advantages of C# Over Other Languages

According to Microsoft, following are advantages of C# over other known languages.

Some advantages over C and C++ are

- It is compiled to an intermediate language (CIL) independently of the language it was developed or the target architecture and operating system [13].
- Automatic garbage collection.
- Pointers no longer needed (but optional).
- Reflection capabilities.
- Don't need to worry about header files ".h".
- Definition of classes and functions can be done in any order.
- Declaration of functions and classes not needed.
- Non existing circular dependencies.
- Classes can be defined within classes.
- There are no global functions or variables. Everything belongs to a class.
- All the variables are initialized to their default values before being used (this is automatic by default but can be done manually using static constructors).
- You can not use non-Boolean variables (integers, floats...) as conditions. This is much cleaner and less error prone.
- Apps can be executed within a restricted sandbox [13].

### 3.2.2 Advantages Over Java

- Usually it is much more efficient than java and runs faster [13].
- CIL (Common (.NET) Intermediate Language) is a standard language, while java byte codes are not.
- It has more primitive types (value types), including unsigned numeric types.
- Indexers let you access objects as if they were arrays.
- Conditional compilation.
- Simplified multithreading.
- Operator overloading. It can make development a bit trickier but they are optional and sometimes very useful [13].

### 3.3 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 [14].

It is named after co-founder Michael Widenius' daughter, “My” The SQL phrase stands for Structured Query Language [14].

The MySQL development project has made its source code available under the terms of the GNU General Public License, The Swedish company, MySQL AB, is now owned by Oracle Corporation [14].

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 (and other 'AMP' stacks). 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 [14]. Figure 3.2 [15] below shows the general architecture of MySQL database.

#### 3.3.1 Features of MySQL

Here are some of the features of MySQL that make it a widely used open source database system [16].

Relational Database System: Like almost all other database systems on the market, MySQL is a relational database system.

Client/Server Architecture: MySQL is a client/server system. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc.
SQL Compatibility: MySQL supports as its database language -- as its name suggests -- SQL (Structured Query Language). SQL is a standardized language for querying and updating data and for the administration of a database.

Views: Put simply, views relate to an SQL query that is viewed as a distinct database object and makes possible a particular view of the database. MySQL has supported views since version 5.0.

Stored Procedures: (SPs for short) are generally used to simplify certain steps, such as inserting or deleting a data record. For client programmers this has the advantage that they do not have to process the tables directly, but can rely on SPs. Like views, SPs help in the administration of large database projects. SPs can also increase efficiency. MySQL has supported SPs since version 5.0.

Triggers: Triggers are SQL commands that are automatically executed by the server in certain database operations (INSERT, UPDATE, and DELETE). MySQL has supported triggers in a limited form from version 5.0, and additional functionality is promised for version 5.1.

Unicode: MySQL has supported all conceivable character sets since version 4.1, including Latin-1, Latin-2, and Unicode (either in the variant UTF8 or UCS2).

User interface: There are a number of convenient user interfaces for administering a MySQL server.
Full-text search: Full-text search simplifies and accelerates the search for words that are located within a text field. If one employ MySQL for storing text (such as in an Internet discussion group), one can use full-text search to implement simply an efficient search function.

Replication: Replication allows the contents of a database to be copied (replicated) onto a number of computers. In practice, this is done for two reasons: to increase protection against system failure (so that if one computer goes down, another can be put into service) and to improve the speed of database queries.

Transactions: In the context of a database system, a transaction means the execution of several database operations as a block. The database system ensures that either all of the operations are correctly executed or none of them. This holds even if in the middle of a transaction there is a power failure, the computer crashes, or some other disaster occurs.

Foreign key constraints: These are rules that ensure that there are no cross references in linked tables that lead to nowhere. MySQL supports foreign key constraints for InnoDB tables.

GIS functions: Since version 4.1, MySQL has supported the storing and processing of two-dimensional geographical data. Thus MySQL is well suited for GIS (geographic information systems) applications.

Programming languages: There are quite a number of APIs (application programming interfaces) and libraries for the development of MySQL applications. For client programming one can use, among others, the languages C, C++, Java, Perl, PHP, Python, and Tcl.

ODBC: MySQL supports the ODBC interface Connector/ODBC. This allows MySQL to be addressed by all the usual programming languages that run under Microsoft Windows (Delphi, Visual Basic, etc.). The ODBC interface can also be implemented under UNIX, though that is seldom necessary. Windows programmers who have migrated to Microsoft's new .NET platform can, if they wish, use the ODBC provider or the .NET interface Connector/.NET.

Platform independence: It is not only client applications that run under a variety of operating systems; MySQL itself (that is, the server) can be executed under a number of operating systems. The most important are Apple Macintosh OS X, Linux, Microsoft Windows, and the countless Unix variants, such as AIX, BSDI, FreeBSD, HP-UX, OpenBSD, Net BSD, SGI Iris, and Sun Solaris.

Speed: MySQL is considered a very fast database program. This speed has been backed up by a large number of benchmark tests [16].

### 3.3.2 Advantages of MySQL

Here are some of the advantages of using MySQL over other known Database Systems.

Open Source: MYSQL is an open source database system which means that anyone can use it for free. Developers can amend its code to suit their requirements which
means that MYSQL is highly customizable as well. Another edge of using MYSQL over other database systems is that; it is available widely in the market with no ownership cost [17].

Fast Development: A lot of people around the globe are continuously developing new modules for integration with MYSQL. This means that it has a wider and faster development circle. Patches, upgrades and fixes are developed fast and become available in forums, blogs and developer sites on the internet.

Better for Small Businesses: This relational database system is free so it reduces the cost of overall database solution for small businesses and companies. This database is relatively easy to learn and operate, so operational cost is reduced substantially which is again an important factor in classifying MYSQL as an applicable and practical tool for small businesses.

Cross Platform Operability: MYSQL is easily installable and operable on different platforms including Windows, Linux, OS2 and Solaris. Cross platform operability makes it a favorable choice for development companies. MYSQL database system also contains APIs for integration with C, C++, PHP, Java, Perl, Python, Tcl, and Ruby etc. You can connect it easily with different development platforms so you can actually integrate applications developed in different OS and development platforms.

Security: MYSQL as a relational database is secure as all access passwords are stored in an encrypted format restricting any unauthorized access to the system. It also encrypts the transactions so eavesdroppers and data harvest tools cannot replicate or regenerate the database transactions once they are processed.

Connectivity: MYSQL clients can access this relational database through standard TCP/IP sockets, named pipes, UNIX sockets and many more. Standard ODBC 2.5 and above functions and commands are also supported in MYSQL.

In short, MYSQL is a free and favored relational database system which is serving the web and application development communities [17].

3.4 HTML AND CSS

In Asp.Net HTML (Hyper Text Markup Language) is used to define the structure of a Webpage and CSS (Cascading Style Sheets) is used to define the styling or appearance of a Webpage.

3.4.1 HTML

Hyper Text Markup Language (HTML): is the main markup language for displaying web pages and other information that can be displayed in a web browser [18].

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in
pairs like `<h1>` and `</h1>`, although some tags, known as empty elements, are unpaired, for example `<img>`. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content [18].

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page [18].

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behavior of HTML WebPages [18].

### 3.4.2 CSS

*Cascading Style Sheets (CSS)* is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL [19].

CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts [20]. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design) [19].

### 3.5 Teleriks

Telerik is a third party .NET component used to make development rapid and easy. Some of the popular tools include Reporting, Charts, Ajax, RadGrids etc. Teleriks is used with .NET application to provide a nice look to the user interface as well [21].
CHAPTER 4

APPLICATION ARCHITECTURE

This chapter explains the basic architecture of the application, various user roles, and their access to different modules. In this chapter we will briefly discuss all the modules used in the application and their interaction with each other.

4.1 USER ROLES

This section will focus on different roles that a user can have for this application and privileges associated with each and every role. The mapping of these roles and privileges are explained using a UML diagram in Figure 4.1.

4.1.1 Guest User

- Guest or non registered user can register on the website.
- Guest must be able to navigate on the website and see open content.
- Guest must be able to access resource section.
- Guest must not be able to create or view blogs.
- Guest must not be able to create, apply or view job listings.

4.1.2 Administrator

- Administrator must not be able to register on the website.
- Administrator must be able to modify their account password.
- Administrator must be able to modify their profile information after logging into the account.
- Administrator must be able to create new administrator only.
- Administrator must be able to create an employer or a teacher role user.
- Administrator must be able to view, create, update and delete job listings.
- Administrator must be able to view, create, update and delete blogs.
- Administrator must be able to delete any objectionable comments from any user on a particular blog.
4.1.3 Employer

- Employer must be able to register on the website.
- Employer must be able to modify his password and update his profile information after logging into the account.
- Employer must be able to view, create, and update his own jobs listings on the website.
- Employer must be able to see the applicants and their resume who applied for his job listings.
- Employer must be able to view, create, and update his own blogs on the website.
- Employer must be able to comment on blogs owned by him or others.
Employer must be able to retrieve his login information in his registered email in case if he/she forgets username and/or password.

4.1.4 Teacher

- Teacher must be able to register on the website.
- Teacher must be able to modify his password and update his profile information after logging into the account.
- Teacher must be able to apply for jobs from any employer.
- Teacher must be able to create multiple job profiles and assign a resume and cover letter to each profile.
- Teacher must be able to retrieve his login information in his registered email in case if he/she forgets username and/or password.

4.1.5 Student

This role is currently not implemented, but should be created in future for students to interact with teachers from around the world.

4.2 Basic Architecture

In this section we will discuss the architecture of the application and interaction between layers in three –tier architecture namely Presentation layer, Business layer, and Data layer. Later in this section I will compare the application architecture with a more commonly known architecture Model-View-Controller (MVC) [20].

4.2.1 Presentation Layer

User Interface or Presentation layer contains pages like .aspx or windows form where data is presented to the user or input is taken from the user. The ASP.NET website or windows forms application (the UI for the project) is called the presentation layer. The presentation layer is the most important layer simply because it’s the one that everyone sees and uses. Even with a well structured business and data layer, if the presentation layer is designed poorly, this gives the users a poor view of the system [20].

4.2.2 Business Layer

Business Access Layer contains business logic, validations, or calculations related with the data. Though a website could talk to the data access layer directly, it usually goes
through another layer called the business layer. The business layer is vital in that it validates the input conditions before calling a method from the data layer. This ensures the data input is correct before proceeding and can often ensure that the outputs are correct as well. This validation of input is called business rules, meaning the rules that the business layer uses to make “judgments” about the data [20].

4.2.3 Data Layer

Data Access Layer contains methods that helps business layer to connect the data and perform required action, might be returning data or manipulating data (insert, update, delete etc) [20].

4.2.4 MVC – 3 Tier Architecture

Model–View–Controller (MVC) as shown in Figure 4.2 [22], is an architecture that separates the representation of information from the user's interaction with it. The model consists of application data and business rules, and the controller mediates input, converting it to commands for the model or view. A view can be any output representation of data, such as a chart or a diagram. Multiple views of the same data are possible, such as a pie chart for management and a tabular view for accountants. The central idea behind MVC is code reusability and separation of concerns.

In addition to dividing the application into three kinds of components, the MVC design defines the interactions between them.

A controller can send commands to its associated view to change the view's presentation of the model (e.g., by scrolling through a document). It can send commands to the model to update the model's state (e.g., editing a document).

A model notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. A passive implementation of MVC omits these notifications, because the application does not require them or the software platform does not support them.

A view requests from the model the information that it needs to generate an output representation [23].

4.3 APPLICATION MODULES

In this section we will discuss all the modules that are part of this application. A brief explanation of these modules is presented below:

4.3.1 Login and Registration Module

- User can register on the website with a role of Employer, Teacher, or Student.
- User can modify their password and change profile information under account settings, once they are logged in.
- User can retrieve their Username and Password by clicking Forgot Username/password link.
- Registered users will be able to see the menu tabs based on their roles and privileges.
- Users are signet out automatically in case of no activity on website for thirty minutes.
- An Administrator is able to create a new user with administrator role.

4.3.2 Blog Module

- Users have different access levels for the blog interface.
- User can click the add button to create a blog and provide a title and content of the blog.
- Teacher role users can only view blogs and post a comment on these blogs.
- Employer role users can Add/Update/Delete their own blogs and can comment on all the blogs.
• Administrator role users can Add/Update/Delete any blogs in the system, remove any objectionable comment form a user and post any comment on a blog.
• User can share or like a blog via facebook widget added on top of the blog.

4.3.3 Job Portal Module
• Users have different access levels for job portal interface.
• Employer role users can Add/Update/Delete their own jobs.
• Employers can view applicant’s profile who applied to the job posted by them.
• Employer can add a Job by providing job title, responsibilities, and expiration date.
• Teachers can create multiple profiles and apply for multiple jobs.
• Administrator can view all jobs posted by various employers and can Add/Update/Delete a particular job.

4.4 Social Networking
A social networking service is an online service, platform, or site that focuses on facilitating the building of social networks or social relations among people who, for example, share interests, activities, backgrounds, or real-life connections. A social network service consists of a representation of each user (often a profile), his/her social links, and a variety of additional services. Most social network services are web-based and provide means for users to interact over the Internet [24].
• Users should be logged in to face book to like a blog
• User can share the blog link on their wall with other friends

4.5 Google Translation
Google Translate is a free statistical multilingual machine-translation service provided by Google Inc. to translate written text from one language into another [25].
• User can select a language from the language dropdown on top of the page, and the website will be translated to that specific language.
• User can choose “show original” to go back to original language content.
CHAPTER 5
APPLICATION DESIGN IMPLEMENTATION AND STYLING

In this chapter we will discuss the architecture of the application and interaction of various system modules with each other. This chapter will explain in detail the implementation of three different layers namely User Interface, Business Objects, and Data Access that form the 3-tier architecture for this application. We will also discuss the actual implementation of various system modules with the flow charts and application screenshots. Later in the chapter we will discuss the security of the application.

5.1 SYSTEM ARCHITECTURE

The system is built on a 3-tier architecture design model, which comprises of presentation layer, business layer, and data layer [26]. Presentation layer has aspx pages, style sheets, JavaScript, jQuery, and master pages. This layer is most important because the end user will be interacting with this layer directly. Business layer has all the application logic defined in it. Users are redirected to certain pages and sections based on the logic defined in this layer. This layer also defines the logic to pull data from the database and render to user interface. Data layer describes the database design of the system; it has details about various tables, stored procedures, functions, indexes, triggers, and interaction between various data objects.

5.2 DATABASE DESIGN

In this section we will discuss the data layer of the 3-tier architectural model on which the application is based. As the application is data oriented, all data is stored and retrieved by database. The system has fifteen tables to store information, twenty five stored procedures to carry out CRUD (Create, Update and Delete) operations on this data, and two functions to do repetitive work. The following information is stored in the database:

- User Account Information: This contains a user’s personal information, his credentials for the website, and his user role is also defined here.
• Jobs Portal Data: This contains data related to jobs posted by employers, profile information of teachers consisting of their resumes, and cover letters and their entity relationship.

• Blogs Data: This contains data related to blogs, blogs created by members and comments posted by members, and their entity relationship.

• Examination Data: This contains data for teacher credentialing in California. Some of the sections are examinations, requirements, fees, and courses.

5.2.1 User Data

In this section we will discuss the user data module. User data is stored and normalized in three tables as shown in Figure 5.1. Normalization also helps in faster data retrieval:

Country: This table stores the member’s country information.

mem_tbl_userrole: This table stores the member’s user role information (i.e. Administrator, Employer, Teacher, and Student).

mem_tbl_members: This table stores personal information of the member and also contains his credentials for the website. Using the email provided in this table, a member can recover his username and password.

5.2.2 Blogs Data

In this section we will discuss blog data module.

Blog related data is normalized and stored in three tables as shown in Figure 5.2. Stored procedures are used to store and retrieve data from the following tables:

*mem_tbl_blog*: This table stores the actual content of the blog and uses iMemberID as foreign key to establish relationship with the user.

*mem_tbl_blogcomments*: This table stores the comments posted by a members on different blogs. This stores the entity relationship between a member, blog, and a comment.

*mem_tbl_members*: This table stores member personal information and his user role.

5.2.3 Jobs Data

In this section we will discuss jobs data module.
Job related data is normalized and stored in four tables as shown in Figure 5.3. These tables have entity relationship among each other for faster data retrieval. Following are the brief descriptions of each table:

mem_tbl_job: This table stores the actual job posting, and it also contains entity relationship between a job and the member who posted it.

mem_map_job: This mapping table contains entity relationship between a member and a job.

mem_tbl_jobapplications: This table contains job profile information of members, i.e., resumes and cover letters. This also contains entity relationship between a profile and a member.

5.2.4 Credentialing Data

Credentialing related data is normalized and stored in five tables as shown in Figure 5.4. Following are the brief descriptions of each table:

glb_tbl_state: This table stores the states for which credential data are stored.
Figure 5.2. Blog data EER-2.

glb_tbl_routes: This table stores different routes for teachers to become credentialized to teach in school.

glb_tbl_examinations: This table stores details of various examinations that a teacher has to take before he/she can start teaching in a school.

glb_tbl_searchterms: This table stores commonly searched terms for prospective teachers.

glb_tbl_requirements: This table stores various requirements that a teacher has to meet before he/she can start teaching in a school.

5.3 DESIGN FLOW

In this section we will discuss the workflow of Blog and Job module through a series of flowcharts. We will also discuss the accessibility of certain features that are based on specific user roles.
5.3.1 Create Blog

The application allows for creating a blog for only registered users who have a user role of admin or employer. Once the user registers himself in one of these user roles, the user has to sign in to the member section to create his or her own blog or to view blogs by another user.

Once the user is logged in to the member section, he/she can click on the blogs tab to get to an interface, where he/she can create new blog. Figure 5.5 shows a step by step process to create a blog for different user roles. For creating a blog, the user has to supply the following information.

- Title of the Blog
- Content of the Blog

Once the above information is supplied, user can click the create blog button to initiate a database transaction. After submission a record in created in mem_tbl_blog, this blog entry is linked to the member id. Upon processing the data, a grid is refreshed to show this blog on user interface. Now the user can manage this blog and make any modifications to it.
5.3.2 Manage Blog

The Blog feature is available to registered users only. The user has to register first in order to access this feature. After registration, the user has to log in to the member section and click on blogs tab. If the user has already created some blogs, he/she can manage it and do following operations on an existing blog:

- Modify the blog content
- Modify the blog title
- Delete the blog
- Delete all comments

Edit Blog: Once the user has made all the modifications to the blog, he/she can click the update button, and blog details are updated in mem_tbl_blog using iBlogID as the key identifier. Upon processing the data, the grid is refreshed with new information.
Figure 5.5. Create blog flowchart.

1. Start
2. Login
3. Is Registered? [Yes/No]
   - Yes: Is Administrator? [Yes/No]
     - Yes: Go To Blog Tab
     - No: Add New Blog
     - Fill Out Blog Details
     - Submit
     - Database Transaction
     - Refresh Datagrid
   - No: Go To Blog Tab
4. Is Employer? [Yes/No]
   - Yes: Add New Blog
   - Fill Out Blog Details
   - Submit
   - Database Transaction
   - Refresh Datagrid
   - END
5. Is Teacher? [Yes/No]
   - Yes: Go To Blog Tab
   - No: View All Blogs
   - Like/Share on Facebook
   - Post Comment
   - Delete Own Comment

Delete Blog: The user can delete a particular blog, by clicking on the delete button to delete the blog, and upon confirmation, the blog is deleted from mem_tbl_blog using iBlogID as the key identifier. Upon processing, the data grid is refreshed and that particular blog no longer exists.

Delete Comments: User can delete his or her own comments, plus comments posted by other members, on his blog. The user can click on the delete button to delete the comment, upon clicking, the comment is removed from mem_tbl_blogcomments using iCommentID and iBlogID as key identifiers. Upon Processing the page is refreshed and that particular comment no longer exists. Figure 5.6 gives step by step process for managing a blog.

5.3.3 Create Job

Job portal is a feature that is available to all the registered users on the website. Based on the user role, a user has more or fewer privileges over the other users. Only users with the employer role can create new job postings. Once a user with this role is logged in to member section, he/she has to click the Job Portal menu tab. Figure 5.7 explains the step by step process to create a job posting, which includes the following:

- The user has to provide job title.
- The user has to fill in job description.
- There needs to be an expiration date for the job.
- The user can enable or disable the job at any time.

Once the user has provided all the above information, the user can click on the create job button to initiate a database transaction. Upon clicking a new entry is created in mem_tbl_job, and entity relationship is established between this job and the user. Once the database processing is complete, the datagrid is refreshed to display this newly created job. Now the user can manage this job.

5.3.4 Manage Job

The Job Portal feature is available for registered users only. The user has to register first in order to access this feature. After registration, the user has to log in to the member section and click on Job Portal tab. If the user has already created some jobs, he/she can manage it and do the following operations on an existing job. Figure 5.8 shows the step by step process to manage a job:
Figure 5.6. Manage blog flowchart.
Figure 5.7. Create jobs flowchart.
Figure 5.8. Manage job flowchart.

- Modify the job title
- Modify the job description
- Modify expiration date
- Delete the job posting
• View applicants profile who applied for the job

  Edit Job: Once the user has made all modifications to the job, he/she can click the
  update button and job details are updated in mem_tbl_job using iJobID as the key
  identifier. Upon processing the data grid is refreshed with new information.

  Delete Job: A user can delete his or her own job posting by clicking the delete button
  to initiate a database transaction, and upon processing, the data grid is refreshed, and
  that particular job no longer exists.

  View Applicants: A user with the employer role can see the list of all the applicants
  who applied to a job posted by him or her; he/she can see the applicants resume and
  cover letter.

5.4 APPLICATION LOGIC

In this section, we will discuss how the three layers interact logically, i.e., how the
data is pulled from database, processed, or formatted by a business logic rule and finally
displayed on the user interface.

5.4.1 Database Logic

The application is data driven, meaning every action (except browsing) that a user
takes triggers a database event such as creating a new record, modifying an existing one,
deleting an existing record, or just fetching a record on display in the user interface.

In the application the layer responsible for fetching data from the database and
displaying them on user interface is called the Data Access Layer (DAL). So primarily, DAL
holds the connection class for the database and encapsulates other properties and methods in
different classes that are invoked to cause database transactions.

DAL is used in almost every operation that is performed by the user on the website.
In general all the values that are passed from the user interface will be grabbed in code
behind or aspx.cs page and then these will be supplied to a corresponding DAL function. In
turn, that function will fire a SQL query or call a stored procedure with parameters as the
values that were passed from the code behind page. Once the query is complete data is
returned in the form of a database objects (datatable or dataset) to the DAL function. In turn
DAL returns the data in the form expected by the invoking function in the code behind.
Hence, DAL is responsible for data transmission back and forth between the database and the
user interface.
5.4.2 Blogging Logic

The Blogging feature is available to registered users only. Using this feature a user can create blogs, update his/her blogs anytime, and delete them.

In this section we will explain how the above operations are carried out. After logging in, a registered user fills in all the details for the blog and clicks on the create button. A record is created in mem_tbl_blog associating this blog with the user. A time stamp is also recorded as creation date.

After the blog is created, a grid is shown with blog name, author name, and the creation date, plus there are buttons to add, edit, view, and delete the blog. The user can click on the view blog button to view the detailed blog with comments (if any) from him/her or other users. User can delete any comments posted on his blog. Comments are stored in mem_tbl_BlogComments maintaining association with the blog and the user, plus a timestamp for marking the creation date.

Additionally, user can edit his/her own blog using the edit button, which will open the form in edit mode so that user can make any modifications to the content or title of the blog. After making changes, the user has to click the update button and using DAL, the record will be updated in the database and a time stamp for update will be recorded. Similarly a user can also delete his/her own blog(s).

5.4.3 Job Portal Logic

This website also provides a fully functional job portal. Using this feature an employer can post jobs and prospective teacher can view and apply for jobs posted by various employers.

An employer can fill out the job details, title, and expiry date. Once done he/she has to click the create button, which in turn will call the DAL to create a new record in the mem_tbl_job with a time stamp as creation date. Similar to the blogging feature, an employer can edit or delete the job posted by him/her anytime. The Employer has also the privilege to view profiles of candidates who applied for the job posted by him/her.

A teacher can go to the jobs tab and view all jobs posted by employers. Only active jobs sorted by creation date are shown inside a grid. Teachers can view and apply for any job, but before that, it is mandatory that they create a job profile. Multiple job profiles can be
A job profile consists of a resume and a cover letter at this point. After a job profile has been created, teacher can apply for a job by clicking apply now button, he/she has the option to select the profile they created in case if they have multiple job profiles.

A record will be created in mem_tbl_jobapplications having association between job, profile, and user, plus a time stamp as date of applying for job.

### 5.4.4 Facebook API

The website provides a special social media feature to the registered users through which they can like any blog or resources on the website. They can share the same on Facebook too.

The Like button lets a user share this website content with friends on Facebook. When the user clicks the Like button on this website, a story appears in the user's friends' News Feed with a link back to this website.

When the Web page represents a real-world entity, things like movies, sports teams, celebrities, and restaurants, use the Open Graph protocol to specify information about the entity. If the website includes Open Graph tags on website’s page, that web page becomes equivalent to a Facebook page. This means when a user clicks on a Like button on this web page, a connection is made between this web page and the user. This page will appear in the "Likes and Interests" section of the user's profile, and you have the ability to publish updates to the user. This web page will show up in same places that Facebook pages show up around the site (e.g. search), and the website can target ads to people who like website content. **Note:** The count on the Like button will include all likes and shares whereas the like connection on the Graph API includes only the number of likes for the web page [27].

### 5.4.5 Google Translation API

The application also provides a language translation tool that can be used to translate English to more than fifty other languages by click one button only. For this feature I am using Google’s API. Google API provides an iframe component that is placed in the master page of the application so that every page can be translated to other languages. This feature will help users from across the world to understand the site content in a better way [25].
5.5 USER INTERFACE IMPLEMENTATION

In this section, we will discuss the User Interface also known as the presentation layer of the application. For this application I have made a very fluid design with good choice of web colors to allow better user experience. Most of the web pages are using a single master page, where the page layout, the header, and the footer are defined. All the styles are placed in two cascading style sheets (css). I have used jQuery for adding visual effects and animation on some panels and for AJAX postbacks. Next we have screenshots of features on the website. I will briefly discuss each all of them.

5.5.1 Blogs

The Blog feature is available to registered users of the website. Once the user logs in, he/she can click on blogs tab that will show him/her the list of all the blogs created by him/her or other registered users on the website. Figure 5.9 shows an administrator view of the blog section. An administrator can add, view, delete, edit, or delete any comments from a blog. A non-admin user can however do the following operations:

View Blogs: A registered user can view all blogs created by him/her or other registered users on the website. He/she can like or share the blog on Facebook.

Edit Blog: The user can edit his/her own blogs only, by clicking on the edit button represented by a pencil icon. A form will open in edit mode with the current content prefilled in the respective sections. The user can make the modifications and click on the update button, which will update the database and refresh the User interface.

Delete Blog: The user can delete his/her own blog only. The user can click on the delete button represented by a cross icon to delete the blog. Upon confirmation, the blog will be deleted permanently from the database.

Delete Comments: The user can delete any comments posted on his/her own blog. For deleting a comment, the user has to click on view blog and then on blog page, delete comment by clicking delete button. Figure 5.10 shows an example of a live blog page.

Create Blog: A registered user with a user role of administrator or teacher can create a blog. After logging into the system, the user has to click the blogs tab, where he/she can see his/her own blogs plus blogs from other registered users. The user can create a new blog by clicking the “Add new record” button from the grid shown in Figure 5.9. The maximum characters limit for the blog content is 12000 characters, and for comment, the limit is 500 characters.
After clicking on this button, a form will open in edit mode with empty values. The user has to provide the title and the content for this blog and click on create button. Upon clicking, a new blog will be added to the existing grid. The grid is fully functional with many features such as filtering, sorting, and paging. Ajax is used to do partial post backs and to rebind the grid.

Blog Details Page: Once the user clicks on the View Blog button in the grid shown in Figure 5.9, he/she is redirected to the blog details page as shown in Figure 5.10. This page shows all the content that user used while creating the blog. Now any registered user can post comments on this blog or they can like or share this blog with their friends on Facebook. Comments can be deleted by both, the blog owner and the user who posted the comment. The character limit for a comment is 400 characters. Once a comment is posted, the user’s name and the date stamp are shown below the comment. The blog details page has a different master page that I have created just for blogs. For easy navigation, returning to the blogs menu will take the user to the blogs menu shown in Figure 5.9.
5.5.2 Job Portal

Create Job: A registered user with a user role of either Administrator or Employer can create a job posting. After logging in, the user has to click on the job portal tab, where he/she can see any jobs created by him/her. To create a new job, the user has to click on Add new record button. Upon clicking on it, a form will open in edit mode, and user can fill in the job title, the job description, etc. On clicking the create button, a job posting will be added to the grid as shown in Figure 5.11.

Edit Job: A user can edit the job posted by him/her by clicking the edit (pencil icon) button. The user can modify any information and click update button, and a job posting will be modified instantly, as shown in Figure 5.11.

Delete Job: User can delete jobs posted by him/her by clicking delete button shown in Figure 5.11.
Figure 5.11. Job portal - View all jobs.

Job Details Page: A user with a user role of teacher can log in to the website, click the job portal tab, and view all active jobs posted by all the employers. The user can click the view job button to see the job details as shown in Figure 5.12. Here, the user can find the requirements of the job, its title, and the last day to apply. The user can choose from his/her multiple job profiles and apply for a job.

View Applicants: An employer can view all the applicants that applied to their job posting by clicking the applicants link in the grid as shown in Figure 5.11. Once they click on that link, they will be redirected to a page as shown in Figure 5.13. On this page, the employer can see the job title, the names of the applicants who applied for this particular job, and a link to view/download their profile information.

View Profile: An employer can view the profile of applicants, who applied to their job postings by clicking the view profile button as shown in Figure 5.13, upon clicking on this link, user will be redirected to a page, as shown in Figure 5.14. Here the employer can download the profile information of the applicant. Originally there was an option to view profile information too. I have removed that feature because of the compatibility issues on various operating systems. The navigation links on this page provide easy access to other pages.
Figure 5.12. Job portal - Job details.

Figure 5.13. Job portal - View applicants.
5.5.3 Resources

The website provides very useful information on teacher credentialing in a particular state. Any user can take advantage of this feature by clicking on the resources tab; here the user has to select the state for which he/she need credentialing information and a search term.

Based on this some PDF links are provided to the user which has all the information regarding subjects, fees, exams, requirements etc. as shown in Figure 5.15.

Currently, data is just available for the state of California, and there are seven search terms that the user can select and get information.

Figure 5.14. Job portal - View profile.

Figure 5.15. Resources.
CHAPTER 6

CONFIGURATION TESTING AND CONCLUSION

The project is a web-based application developed using Microsoft Visual Studio 2010 and MySQL database. In this chapter we will discuss the deployment of the application on the server, the configuration of the application, and finally the conclusions.

6.1 CONFIGURATION

The configuration needed to deploy this application is following:

- Windows Server 2003 or above
- .NET Framework 4.0 or above
- IIS 7.0 or above
- MySQL 5.5 or above
- Teleriks Tools for ASP.NET

6.2 DEPLOYMENT

Deployment is the most crucial part in software development life cycle. Following steps should be followed to deploy the application on server with above configuration:

- Copy the application folder and paste it in ‘\intepub\wwwroot’ directory on the server.
- Go to IIS manager, and under ‘Sites’, create a new site, and browse to the application path in ‘\intepub\wwwroot’ directory.
- Purchase a domain and point DNS host name record of the domain to an IP address of the server.
- Navigate to the public domain or local host to see the application in action.

6.3 TESTING

The application has been thoroughly tested, and all known bugs have been fixed. Below are the steps followed to test the application:

- Tested all the hyperlinks to see if they redirect to correct page.
- Tested browser compatibility on all major browsers.
- Tested database connectivity and porting onto other machine.
- Tested validation on all user inputs to avoid incorrect information.
Tested all stored procedure and their return codes.

6.4 SUMMARY AND CONCLUSIONS

This thesis is developed to help teachers, employers and students find a common platform where they can find resources, communicate with each other, and share information through social media. I have tried to make this web application very interactive and useful for all users whether it is an employer or a teacher. The website has a blogging feature and fully developed job portal for the users. The database for the application is scalable to accommodate any future enhancement. Currently the web application is designed to show data for California, but more states can be added anytime, since the data is highly normalized in database.

I faced many challenges, from gathering the requirements to changing the database framework from Sql Server 2008 to MySQL when the application was almost sixty percent complete. A further challenge, barely addressed, is the vast amount of information required to cover this topic. Furthermore, gathering credentialing information for a particular subject and reciprocity agreements between various states was challenging.

Nevertheless the website has a fluid design and works on all major browsers, plus it has a robust database design to accommodate all future enhancements.
CHAPTER 7

FUTURE ENHANCEMENT AND SCOPE

Software development is dependent on new evolving technologies. My application is packaged to provide many features, mentioned below:

- Resources for teachers.
- Blogging feature.
- Job portal.
- Language translation using google api.

Some of the future enhancements that can be made for this application are:

- Implement a chat feature to help users to do live chat on the website.
- Implement a new user role for student.
- Implement the website on android and IOS based devices.
- Implement this project for other states in the United States.
- Add the other 49 states, including a dynamic map.
REFERENCES


APPENDIX

GLOSSARY
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LARC</td>
<td>Language Acquisition Resource Center</td>
</tr>
<tr>
<td>NCSSFL</td>
<td>National Council for State Supervisors for Languages</td>
</tr>
<tr>
<td>CTC</td>
<td>Commission on Teacher Credentialing</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>CLR</td>
<td>Common Language Runtime</td>
</tr>
<tr>
<td>BCL</td>
<td>Base Class Library</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>AJAX</td>
<td>Asynchronous Java Script and XML</td>
</tr>
<tr>
<td>Admin</td>
<td>Administrator</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>VS</td>
<td>Visual Studio</td>
</tr>
<tr>
<td>ASP</td>
<td>Active Server Pages</td>
</tr>
<tr>
<td>C#</td>
<td>C Sharp</td>
</tr>
<tr>
<td>VB</td>
<td>Visual Basic</td>
</tr>
<tr>
<td>T-SQL</td>
<td>Transact-SQL</td>
</tr>
<tr>
<td>IIS</td>
<td>Internet Information Server</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SP</td>
<td>Stored Procedure</td>
</tr>
</tbody>
</table>