HIGHER EDUCATION ENTRANCE STUDENT PLACEMENT PROCESSING AND RETRIEVAL SYSTEM

By

Getachew Feleke

A Project paper submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science

June, 2008

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

HIGHER EDUCATION ENTRANCE STUDENT PLACEMENT

PROCESSING AND RETRIEVAL SYSTEM

By

Getachew Feleke

Name and Signature of members of the Examining Board:

1. Dr. Dida Midekso, Advisor _____

2. Dr. Mulugeta Libsie _____

Acknowledgement

First I would like to thank my advisor Dr. Dida Midekso for his excellent advice and

passionate guidance throughout this project.

Thanks to my friends whose support and encouragement goes even beyond the

accomplishment of this project.

Finally, my thanks go to all my family, since it is difficult to mention their contribution to

my achievements in words it is better to say my heart has recorded it forever.

Table of Contents

List of Figures

Abbreviations

CPU: Central Processing Unit

CSS: Cascading Style Sheet

DLL: Dynamic Link Library

DOM: Document Object Model

HEI: Higher Education Institution

LAN: Local Area Network

MoE: Ministry of Education

UI: User Interface

UML: Unified Modeling Language

XML: eXtensible Markup Language

Abstract

This project is aimed to develop a system that handles higher education entrance placement processing and making available the placement information to the public by using the Ministry of Education website. Actually the Ministry has a system to manage this work, but it is almost impossible to view the placement information of students via Internet by using the Ministry's website due to problem on the existing system.

The project can be mainly divided in to three parts. The first part is the one that handles student personal and academic information registration and managing the placement process, and finally putting the processed data in an XML format. The second part is a website which gives placement information for individual students, schools, and universities and colleges. The third part is a Web Service that enables universities and colleges to get newly placed student information in an XML format so that they can incorporate it to their application by avoiding data reentry.

1. Introduction

1.1 Background

Even though it requires initial investment, expensive task of system analysis, design and implementation, continuous follow up and maintenance, and some risk; using computers is unquestionable so as to get benefits like speed, accuracy, flexibility, choice of configuration, storage capacity, data processing, reduction in paper work, reduced cost, facilitates report preparation, reduce the space requirements, reduces the manpower requirement, increased ability to perform computations, etc [8].

Due to its numerous benefits nowadays computers are involved in every sectors of life, in wide range of applications. And even in some areas mankind has been totally dependent on computers. In this project out of numerous areas of applications of computers we are going to apply for solving a challenge of a nation, developing higher education entrance student placement processing and retrieval system.

According to the Ethiopian Government Education and Training Policy of 1994, the curriculum is offering 10 years of general education consisting of 8 years of primary education and 2 years of general secondary education (9-10) with the second cycle of secondary education (11-12) which prepares students for continuing to higher education [9].

In Ethiopia there are a number of private and governmental higher education institutions. In private colleges since the expense of education is totally covered by students as long as a student fulfills the entrance requirement he/she can join any higher education institution he/she likes. However, in governmental higher education institutions as some percent of educational and living expenses are covered by the government placement is done by Ministry of Education based on the score of the student, the availability of fields of study, the number of students that it can accommodate, the government policy, proximity of the institution to student's home etc.

Since our interest focuses on placement and retrieval of placement information let us look at it in some detail. The Ministry of Education administers entrance examination at national level for college preparatory II (grade 12) students every year in the month of May. Then the national examination center corrects, grades, and distributes the results to students every year around the month of August. Based on the criteria set by the Ministry of Education those students who satisfy the entrance requirement will fill in a form about their choice of field of study and higher education institution they would like to join.

The results and chosen field of study of students will be encoded by the Ministry to software which stores its data in SQL server. Then after data entry is being completed, it will be processed by the system operators, then the result of execution i.e. the placement information will be available to students and schools through the web site of the Ministry of Education.

1.2 Statement of the Problem

The number of students being placed is dramatically increasing from year to year, for instance in the 2007/08 academic year more than 56,000 students were assigned to different government colleges and universities throughout Ethiopia. Currently student placement is done centrally by the Ministry of Education and this placement process has two major problems.

The first and the major problem is that it is almost impossible for students to access their placement information through the Ministry's website. Mainly, the cause for this problem is that, almost all students who have Internet access try to access the placement information at the same time which creates a bottleneck at the SQL server engine which is being used as data repository for the existing system. Managing more than 56,000 students using a single SQL server engine is unthinkable since most of them try to log on to the system at a time (just after the Ministry's announcement), and it will be even more difficult in the coming years since the Ministry is planning to accept more and more students. Actually, this placement information is released through plasma TV in schools, which requires listening the placement information of all students for very long hours since if some of its part is missed once there is no chance to listen it again, due to this it makes it very much inconvenient for students to get their placement information.

The second problem is misplacement. Basically, the Ministry has predefined and written criteria for placement of students that is based on their result, choice and proximity of the institution to their home, actually with some more detail information. Unlike the

availability of placement information criteria the existing application is not meeting the requirements. And since the system is developed by a commercial company and the company doesn't allow accessing the source code, it is impossible to modify and debug the software.

This project tries to answer the following leading questions

(1) What is the cause for the previously mentioned web application to be terribly slow when it is accessed by multiple users?

(2) What should have to be done to mitigate the performance problem of the MoE placement website?

(3) How can we develop an application that minimizes misplacement complaints?

(4) How can we make students' personal data available to the respective colleges and universities so that they can access and incorporate to their application in a safe way and get rid off reentry of data?

1.3   Objective of the Project

The general and specific objectives of the project are described as follows:

1.3.1 General Objective

To develop a student placement managing system that gives a reliable and timely response to each user requests.

1.3.2 Specific objectives

To attain the general objective the following specific objectives are set

- To develop a system that performs higher education student placement based on the placement criteria of the Ministry,

- To make retrieval of placement information easier by using the Ministry's web site, for both students and schools,

- To make student data available on the web for the respective universities and colleges so that they can access and use that information directly with their web site by avoiding repeated data entry.

1.4   Methodology

For the successful completion of this project, a number of data gathering, analysis, and programming tools and techniques are put into use. The data collected focuses on issues like, what is expected from the system? What are the inputs of the system? What are the outputs of the system? What are the rules of processing the input data? How are they processed? Who requires which data in what form? etc,.

1.4.1 Data Sources

Both primary and secondary sources of data are used to gather information.

Primary Data Source

- By interviewing respective personnel of the Ministry.

- Collect end users feedback about the existing system from the Ministry, students, schools, and government higher institutions,

Secondary Data Source

- Students' field of study and higher institution choosing forms of the Ministry,

- Placement criteria documents of the ministry,

- Relevant documents and techniques that helps to improve the performance of website

Just after data collection requirement analysis is done and the next stage of the project is system design, and object oriented design is used to design the system so that the output will enable to visualize the objects in the system in a simple and understandable form.

1.4.2 Programming Software and Tools

The next job that is done after system design is crosschecking gap between what has been done and what is needed. To do so prototype of the system was developed and refining missing requirements was done. The prototype has two parts

- The first part is the one which is used to demonstrate the placement process and it is developed using windows forms,

- The second part is used to demonstrate retrieval of placement information, and it is developed using web forms.

By making amendments from the feedback of prototyping, implementation of the system was the task in the project life cycle. The implementation is divided in to three different parts and done using a number of tools:

Since execution of placement process is always done by one person and only once (if necessary can be reprocessed), the part that manages the placement process is handled using windows application. For this purpose to store the pre-processed data, database is developed using SQL server. And the graphical user interface and the code are developed using C# since it is general-purpose, object-oriented programming language which enables us to develop all the three parts of the project i.e. windows, web, and web service application parts of the project with good features on manipulating XML.

The processed placement data is made available to students, schools and higher institutions using web application. But,

> "The min memory per query option lets the administrator specify the minimum amount of memory any single query receives. If min memory per query is set to 2,048 kilobytes (KB), the query is guaranteed to get at least that much total memory. You can set min memory per query to any value from 512 through 2,147,483,647 KB (2 gigabytes (GB)). The default is 1,024 KB. Increasing the value of min memory per query may improve performance for some small to

medium-sized queries, but doing so could lead to increased competition for memory resources." [10],

Which means using SQL server database as a backend for such very high number of concurrent users creates bottleneck. As a result the backend for the web application shall be free from such bottleneck problem. If the backend is any kind of database since it is controlled by the database engine when the number of query request increase there will be a long queue of queries on the database engine which is again bottleneck. And if we make the data store simple flat file since there is no central database engine their will not be bottleneck problem as we are using a database but the challenge here is on file reading performance of the server machine.

So the solution lies between database and flat files and the best solution is using XML as a data store since it is semi structured data. Using XML as a data store totally avoids the bottleneck problem since to read an XML file what we have to do is call the parser which is a DLL file in which the first call works independent of the second and the third call since there is no synchronization and transaction management which is the case of databases. After the placement information is being processed in an SQL server database it will be broken down and exported to a number of XML files so that the reading performance of the CPU will increase.

The other reason for using XML is that outputting XML is just as easy as reading it. XML is a very good choice for storing data in many cases. It's easy to parse and write. Parsers have mechanisms to verify syntax and completeness, so you can protect

your program from corrupted data. One of common application of XML is dictionary. A dictionary is a simple one-to-one mapping of properties to values. A property has a name, or key, which is a unique identifier. A dictionary is kind of like a table with two columns. It's a simple but very effective way to serialize data [1].

The graphical user interface and the code for the web application is being developed using C# and ASP.NET for the same reason described before.

The next part of the implementation tries to assure the availability of student personal data to universities and colleges through XML Web Services. An XML Web Service is chosen for transferring data due to:

> "XML Web Services are designed to interact directly with other applications over the Internet. As a result, XML web services do not have user interfaces (UIs); instead, XML Web Services provide standard defined interface called contracts that describe the services that they provide. An XML Web Service can be used internally by a single application, or it can be used externally by many applications that access it through the Internet. XML Web Services enable you to share programming logic and capabilities with numerous Web and Windows applications, with applications that are running on other platforms." [2]

The last but the most important part of the project is testing: to perform the testing pre-placement information of students, i.e., sample data we get from forms filled by

students is being encoded in to the system then unit, integration, and functionality tests being performed both on the windows and on the web application.

1.5 Justification of the Work

The current placement system of the Ministry is using SQL Server 2000 database as a back end and webpage for its front end, which is accessed by several thousands of users at a time which creates bottleneck at the database. The best solution to avoid the bottleneck problem due to one SQL Server database engine is to have server farm that contain a number of servers and each server storing only limited number of students, may be one server for storing students data from one region. But currently having server farm is not feasible solution for the Ministry since it requires having more than nine server machines for a system which will be functional only for not more than one month in a year.

Due to these reasons a system that can solve the current bottleneck problem and that do not incur much cost is a must for the Ministry. To address the problem on the current system, this project aims at developing an XML based system that brakes down the big data on the database into a number of smaller sized XML files which has no bottleneck problem due to a single processing engine is implemented. In addition since the web application is reading from a number of different XML files in the future if the Ministry is able to get server farms, the application can be used by only modifying the connection module of the system, which means the application is highly modifiable since modular approach of development is used.

1.6   Limitations

The project targets at a product which is accessible any where any time independent of software and hardware platform, and it is successfully addressed. And as it is discussed in the methodology section under Programming Software and Tools sub section the project is developed using C#, ASP.NET, XML Web Services using Microsoft .NET in which to run all this applications the server should have the .NET Framework installed on it and in addition the C# application uses SQL server as a back end, which means we have to run the system only on windows server. But if the Ministry wants to change its server from windows to some other like SUN, their will be problem of platform.

1.7   Organization of the Document

This report document contains six chapters including this chapter which mainly focuses on introduction. Chapter two gives us general overview of the placement process done by Ministry of Education step by step. Chapter three presents review of research works, on the root causes of performance of WebPages and their respective remedies. In chapters four and five, we presented the analysis and design of the developed system respectively. The conclusion and recommendations drawn from the project are briefly explained in the last chapter, chapter six.

2. Overview of Student Placement Process

As described in the background in Ethiopia a student will join higher education when he/she completes college preparatory education (Grade 11-12) and gets at least the minimum entrance result from the national entrance exam.

Those students who fulfill the entrance requirement of higher education can join both governmental and private higher education institution. But to join governmental higher education institution, the student should pass through the placement process of the Ministry of Education.

The Ministry of Education has placement policy to manage this placement process. According to the policy, students who qualify to join governmental higher education will list their choice of fields of study and their preferred colleges or universities in descending order. Then the placement process will be done as shown in the following steps:

Step 1: List fields of studies of all universities and colleges throughout the country;

 Step 2: Set the maximum number of students a university or a college can accept under each specific field of study;

Step 3: Encode students score and choice information in one file;

Step 4: Evaluate the cases of each student with disabilities and special cases, and if their case is satisfactory assign those students in their choice of field of study and institution;

Step 5: Select one field of study from the list, the total number of students who can be placed in that field is known;

Step 6: Sort only female students (government policy of affirmative action) who have chosen that specific field of study as their first choice in descending order based on their score;

Step 7: If the number of female students who have chosen that field as their first choice is greater than one forth (1/4) of the available capacity of institutions only the top students who are within the range of quarter of total capacity will get that specific field of study with the respective institution, the rest will compete with the whole students as indicated in step 8;

Step 8: Select one field of study from the list, the current capacity (at least ¾ of its capacity is free, since ¼ of its capacity could have been taken by activities from steps 6-7) of that field is known;

Step 9: Sort those students (without sex discrimination) who have chosen that specific field of study as their first choice in descending order based on their score;

Step 10: If the number of students who have chosen that field is greater than the available capacity of institutions only the top students who are within the range of the capacity will get that specific field of study, the rest will compete with their next choice as shown in Step 12,

Step 11: Repeat steps 8-10 for all students who are not placed yet and for available fields of study;

Step 12: Using the previous list of field of studies check the total capacity of a specific field of study and the number of students who are assigned to that field, if the two

numbers are equal change its status to full, but if the number of assigned students in that field is less than the capacity of the field of study go to the next step;

Step 13: List students who are not assigned to a certain field of study and whose second choice is that specific field of study in descending order of their score;

Step 14: Assign those students within the range of the current capacity of that field to that specific field;

Step 12: Repeat steps 12-14 until all fields of studies and student choices exhaust;

Step 15: If there are some students who are not assigned to a certain field of study, assign those students in the available fields of study based on their high school stream;

Step16: Up to now the students are assigned only to certain field of study but not to a specific institution. So those students whose field of study is known will be assigned to universities and colleges first by their score and second by proximity of their home to that specific institution.

At this stage all students are assigned to a certain field of study at a certain higher education institution, so the next activity is announcing the placement information to students, schools, and higher institutions. And even though it has chronic problems the Ministry is doing this job using the Ministry's placement website and plasma TV.

3. Literature Review

Even though the web has enabled us to get information easily and it enables people in the world to exchange information as if it is a small village managing such information on the net in a way that is understandable and easily accessibly to all is a challenging task.

For a service provider speed is one of the main factors that determine its relationship with its clients. When we consider specifically the speed of a service provider website, even though it provides all the necessary information to its clients if there is problem on its speed it will be one of the root causes for its dark image.

> "There's a time bomb on the web: user patience. It starts ticking each time someone opens one of your pages. You only have a few seconds to get compelling content onto the screen. Fail, and you can kiss your customers and profits goodbye.
>
> You can't count on fast connections either. Most of your customers are still sucking content through a 56K straw. You have to serve up greased lightning or they'll bail. " [5]

One of the main challenges of web-based applications is the loading speed or the performance of its pages.

> "It is estimated that if a page doesn't load within 5-8 seconds you will lose 1/3 of your visitors. So if your website speed is slower than that perhaps

you should revise your design, site structure or just move to a faster server." [6]

So for web application developers it is a must to identify and understand the root causes of performance problems of web pages before going deep into development.

What affects site loading speed of a web page?

- web server speed (server resources, Internet connection speed)
- database usage (none, one or many)
- HTML code page size (20kb or larger)
- CSS file size (5kb or larger)
- Total image size (50kb or larger)
- Script usage (none, 10kb or larger)
- amount of file requests (style sheets, Server Side Includes, images, scripts, etc) [7]

According to [7] the solutions for these problems are listed as follows:

1) Web server speed

Web server can affect web site speed in several cases: when there are many sites on the server and it is running slow or it is hosted on a narrow Internet connection. To solve this, one can either move to dedicated virtual or physical server or pick another hosting company.

2) Databases

If the site uses a lot of databases, we should consider whether we need the database solution at all, first. If not, switch to static pages. And if we do need to use databases, we can merge some of them to make them load faster.

3) HTML code size

If the site is using tables for layout, the developer should reduce the total HTML size by about 50% (verified by real site development experience) by converting the site to CSS for layout. This will not only increase the site speed, but will also make the site display nicely in liquid layout and very easy to edit (need to adjust a single line in a single file to change the look of the element - a heading, for instance - throughout the entire site).

4) CSS file size

If the developer is using style sheets (which are good), you may want to optimize them by using shortcuts, grouping selectors with the same properties and values together, actually use the cascading effect and so on. This may reduce the size of the CSS file by 10-20%, if not more.

If the site uses inline or internal styles, the developer would rather place them in an external CSS file. This will allow browsers cache them and they won't need to load them every time the page does.

5) Total image size

If the web site is using a lot of images, the developer may consider several things. One is to consider whether some images are needed to convey what you need at all. Useless images can be removed.

If some of the images have text, it is better to replace them with text, styled with CSS. With addition to increasing site speed, this will make the website readable without images and accessible (both to the humans and the search engines).

If still there are any images left, you may want to ensure they are created to fit the size they are displayed at. You don't want to use a 100*100 image in a 50*50 thumbnail, because a larger image would be shown as a four times smaller one.

Lastly, we have to optimize the image files themselves. We may want to save images as GIF or PNG files, if you don't need high quality images. Large product images should rather be displayed in the JPEG format anyway. We should try saving the images "for the Web", if we can.

Simply optimizing website images can reduce the total page size by half or more. This is one of the most powerful techniques to make the website load faster along with converting the site to CSS.

6) Script files size

If you use Javascript on your site, you need to consider if you need it at all. Some things, such as mouse-over, can be done by all browsers without any scripts.

At any rate, you'd rather put your Javascript in external files and attach them using the <script> tag. This way, the page will load faster and the search engines won't choke on Javascript.

7) HTTP file requests

Though it was advised to place styles in external stylesheets as well as Javascript in external script files, which has created a few more file requests, it is much better than have all users load all the styles and scripts on every page. So a couple of more file requests are worth the improvement [7].

If we optimize all the site speed factors, mentioned before, we can get a noticeable site speed increase in our website. But if we used all the developer skills available and have not noticed any site speed improvement, which means have optimized our site well.

According to Yahoo Developer Network there are a number of reasons that influences the performance of a website. A research group at Yahoo has investigated the main causes for this problem and they have pointed out the causes under the title "Best Practices for Speeding Up Your Web Site" [11] and according to their work best practices for making web pages fast are as follows.

1) Minimize HTTP Requests

80% of the end-user response time is spent on the front-end. Most of this time is tied up in downloading all the components in the page: images, stylesheets, scripts, Flash, etc.

Reducing the number of components in turn reduces the number of HTTP requests required to render the page. This is the key to faster pages. One way to reduce the number of components in the page is to simplify the page's design.

2) Put Stylesheets at the Top

Moving style sheets to the document HEAD makes pages appear to be loading faster. This is because putting stylesheets in the HEAD allows the page to render progressively. The problem with putting stylesheets near the bottom of the document is that it prohibits progressive rendering in many browsers, including Internet Explorer. These browsers block rendering to avoid having to redraw elements of the page if their styles change. The user is stuck viewing a blank white page.

3) Put Scripts at the Bottom

The problem caused by scripts is that they block parallel downloads. The HTTP/1.1 specification suggests that browsers download no more than two components in parallel per hostname. If you serve your images from multiple hostnames, you can get more than two downloads to occur in parallel. While a script is downloading, however, the browser won't start any other downloads, even on different hostnames.

4) Avoid CSS Expressions

CSS expressions are a powerful (and dangerous) way to set CSS properties dynamically. They're supported in Internet Explorer, starting with version 5. The problem with

expressions is that they are evaluated more frequently than most people expect. Not only are they evaluated when the page is rendered and resized, but also when the page is scrolled and even when the user moves the mouse over the page. Moving the mouse around the page can easily generate more than 10,000 evaluations.

5) Make JavaScript and CSS External

Using external files in the real world generally produces faster pages because the JavaScript and CSS files are cached by the browser. JavaScript and CSS that are inlined in HTML documents get downloaded every time the HTML document is requested. This reduces the number of HTTP requests that are needed, but increases the size of the HTML document. On the other hand, if the JavaScript and CSS are in external files cached by the browser, the size of the HTML document is reduced without increasing the number of HTTP requests.

6) Reduce DNS Lookups

The Domain Name System (DNS) maps hostnames to IP addresses, just as phonebooks map people's names to their phone numbers. When you type www.yahoo.com into your browser, a DNS resolver contacted by the browser returns that server's IP address. DNS has a cost. It typically takes 20-120 milliseconds for DNS to lookup the IP address for a given hostname. The browser can't download anything from this hostname until the DNS lookup is completed.

DNS lookups are cached for better performance. This caching can occur on a special caching server, maintained by the user's ISP or local area network, but there is also caching that occurs on the individual user's computer. The DNS information remains in the operating system's DNS cache (the "DNS Client service" on Microsoft Windows). Most browsers have their own caches, separate from the operating system's cache. As long as the browser keeps a DNS record in its own cache, it doesn't bother the operating system with a request for the record.

Internet Explorer caches DNS lookups for 30 minutes by default, as specified by the DnsCacheTimeout registry setting. Firefox caches DNS lookups for 1 minute, controlled by the network.dnsCacheExpiration configuration setting.

When the client's DNS cache is empty (for both the browser and the operating system), the number of DNS lookups is equal to the number of unique hostnames in the web page. This includes the hostnames used in the page's URL, images, script files, stylesheets, Flash objects, etc. Reducing the number of unique hostnames reduces the number of DNS lookups.

7) Minify JavaScript and CSS

Minification is the practice of removing unnecessary characters from code to reduce its size thereby improving load times. When code is minified all comments are removed, as well as unneeded white space characters (space, newline, and tab). In the case of JavaScript, this improves response time performance because the size of the downloaded

file is reduced. Two popular tools for minifying JavaScript code are JSMin and YUI Compressor. The YUI compressor can also minify CSS.

8) Remove Duplicate Scripts

It hurts performance to include the same JavaScript file twice in one page. This isn't as unusual as you might think. A review of the ten top U.S. web sites shows that two of them contain a duplicated script. When it does happen, duplicate scripts hurt performance by creating unnecessary HTTP requests and wasted JavaScript execution.

9) Reduce the Number of DOM Elements

A complex page means more bytes to download and it also means slower DOM access in JavaScript.

10) No 404s

HTTP requests are expensive so making an HTTP request and getting a useless response (i.e. 404 Not Found) is totally unnecessary and will slow down the user experience without any benefit. Some sites have helpful 404s "Did you mean X?", which is great for the user experience but also wastes server resources (like database, etc).

The other issue that the new placement managing system tries to emphasize is reuse of students' personal data. In the current academic year 2007/08 more than 56,000 students data is being encoded to the ministry's database by data entry clerks for processing the placement but just after the placement process is completed the personal data of all these

students is being encoded by higher institutions data entry clerks again which is wastage of resources.

"Software assets, or components, include all software products, from requirements and proposals, to specifications and designs, to user manuals and test suites. Anything that is produced from a software development effort can potentially be reused.

A good software reuse process facilitates the increase of productivity, quality, and reliability, and the decrease of costs and implementation time. An initial investment is required to start a software reuse process, but that investment pays for itself in a few reuses. In short, the development of a reuse process and repository produces a base of knowledge that improves in quality after every reuse, minimizing the amount of development work required for future projects, and ultimately reducing the risk of new projects that are based on repository knowledge." [4]

As we have seen in the previous sections, the Ministry of Education stores its data in SQL Server database but higher education institutions use different kinds of application both in hardware and software platform, and to make the data reusable independent of platform XML web services using .NET is used.

"A Web service is a unit of programmable logic that is accessible by using standard web protocols such as HTTP and XML. A Web service can be used locally by a single application, or published on the Internet for use by many different applications.

Web services allow applications to share data and functionality, and use XML-based messaging to communicate between systems that use different component models, operating systems, and programming languages. Developers can create applications that combine Web services from a variety of sources in much the same way that developers traditionally use components when creating a distributed application.

Using XML-based messaging to communicate between a Web service and a client application frees both the Web service client and the Web service provider from needing any knowledge of each other beyond inputs, outputs, and location. A Web service can provide reusable functionality that many clients can share." [3]

According to the Ministry, the number of students who will be joining higher institutions will increase each year, for example the number of students who will be joining higher institutions by the academic year 2008/09 is expected to be more than 90,000. So reusing student's personal data by avoiding data entry by higher institutions will have a significant contribution when we think at a national level.

Finally in this project to get the benefits of those issues raised by literatures the following measures are taken,

- Data bound controls like combo boxes shall read data from the server only when it is loaded for the first time, but when the page is refreshed they shouldn't read data again and again,

- We have used the minimum possible html code size,

- No images used,

- We have reduced the number of DOM elements so the page will be displayed easier,

- No 404 pages used, if a record doesn't exist only a text that describes the situation will be displayed on the current page itself,

- We have used XML web services, since to consume from such application it needs developing an application which will minimize the number of downloads as a result it will reduce the burden on the web server.

4. System Analysis

In chapter 2, we have seen the different steps, approaches and the main processes involved in the placement process by the Ministry of Education. In this chapter the functional and non-functional requirements of the proposed system will be described and modeled using UML models.

4.1 Functional Requirements

The developed system is expected to provide the following functionalities:

- Register student personal information,

- Register score information,

- Register student choice of field of study,

- Register student choice of university or college,

- Process the placement based on the policy of the Ministry,

- Break down the bulk data into smaller groups and exporting the processed data to XML format,

- Display students', schools' and universities' placement information online,

- Enable universities to download newly placed students information in an XML format

4.2 Non-Functional Requirements

The following list of non functional requirements is expected from the system:

The windows application part

- should allow login to only authorized users i.e. users that have user name and password,

- should work over a networked environment,

- should validate data during data entry,

- should be user friendly

The web based part

- Should respond to requests in a reasonable period of time, i.e. at least before the session expires,

- Should give only valid result, if no data is found with the specified criteria the system should not have to crash or give invalid response,

4.3 System Models

4.3.1 Use Case Model

In its simplest form, a use case can be described as a specific way of using the system from a user's (actor's) perspective. Use cases provide a means to capture system requirements, communicate with the end users and domain experts, and test the system. Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system.

4.3.1.1. Use cases

The following use cases have been identified from the system specification

- Register student

- Register score

- Register field of study

- Register institute

- Process placement

- Export to XML

- View placement information

- Download student data

4.3.1.2. Actors

The identified actors that will be participating in the system are:

- Student

- School Clerk

- Higher Education Institution Clerk

- MoE Clerk

4.3.1.3.   Use Case Diagram

Use case diagrams graphically depict system behavior (use cases).   These diagrams present a high level view of how the system is used as viewed from an outsider's (actor's) perspective. From the identified use cases and actors the use case diagram of the system is shown in Fig 4.1.
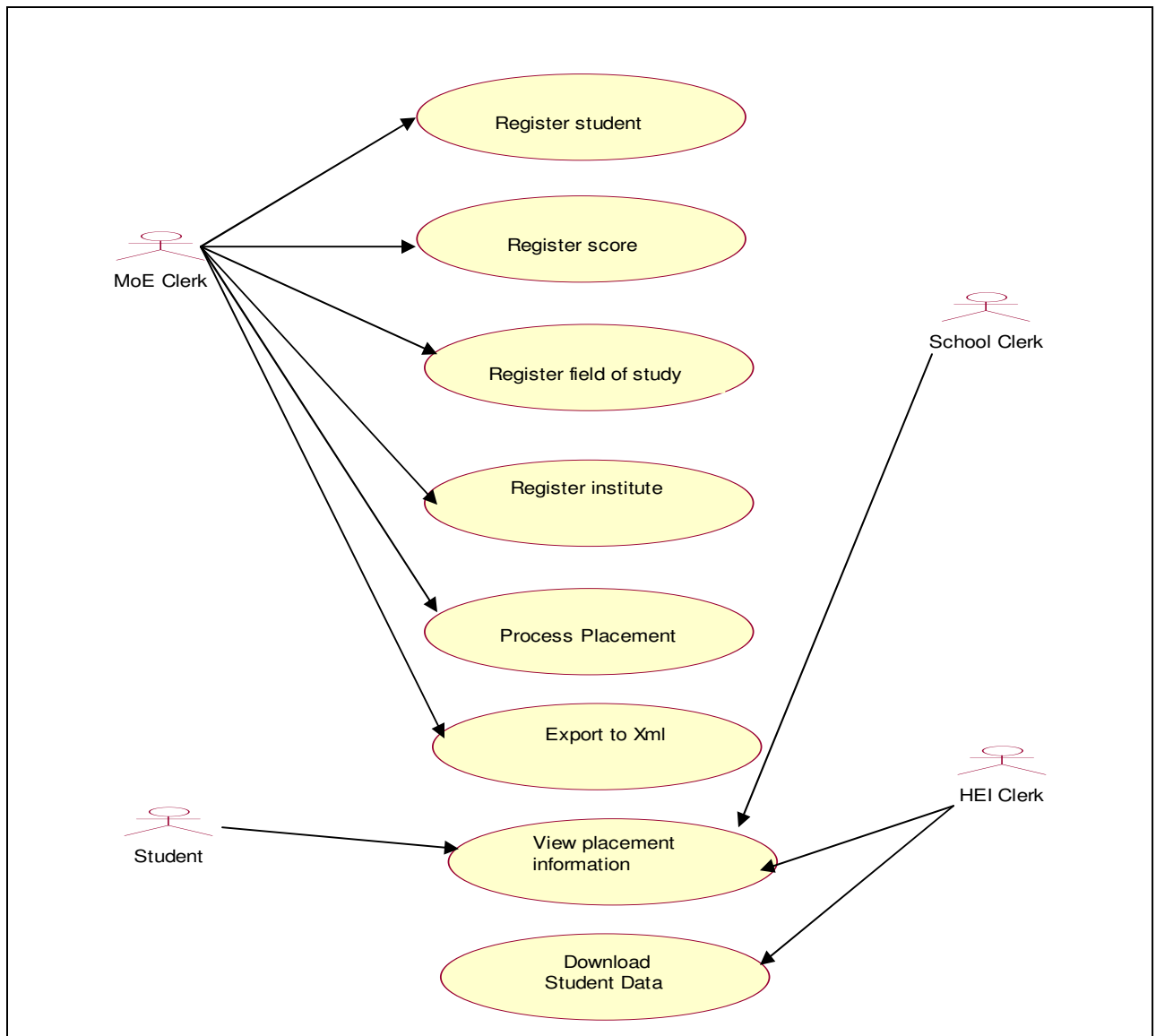

Fig 4.1 Use Case diagram of student placement processing & retrieval system

As we can see from the use case diagram data clerk of the Ministry can register students' personal information; register student's score, chosen field of study and chosen institution; process placement, and export the processed data to XML. A student is another user of the system and he/she can see his/her placement information. Representative of a school can get from the system placement report of all students from that school and similarly personnel of the respective higher institution can get list of newly assigned student together with students' personal data from the system.

4.3.2 Class Diagram

Class diagrams are used to describe the structure of the system in terms of classes. Class diagrams describe the system in terms of objects, classes, attributes, operations, and their association. Identified classes and relationship among them is shown in Fig 4.2.
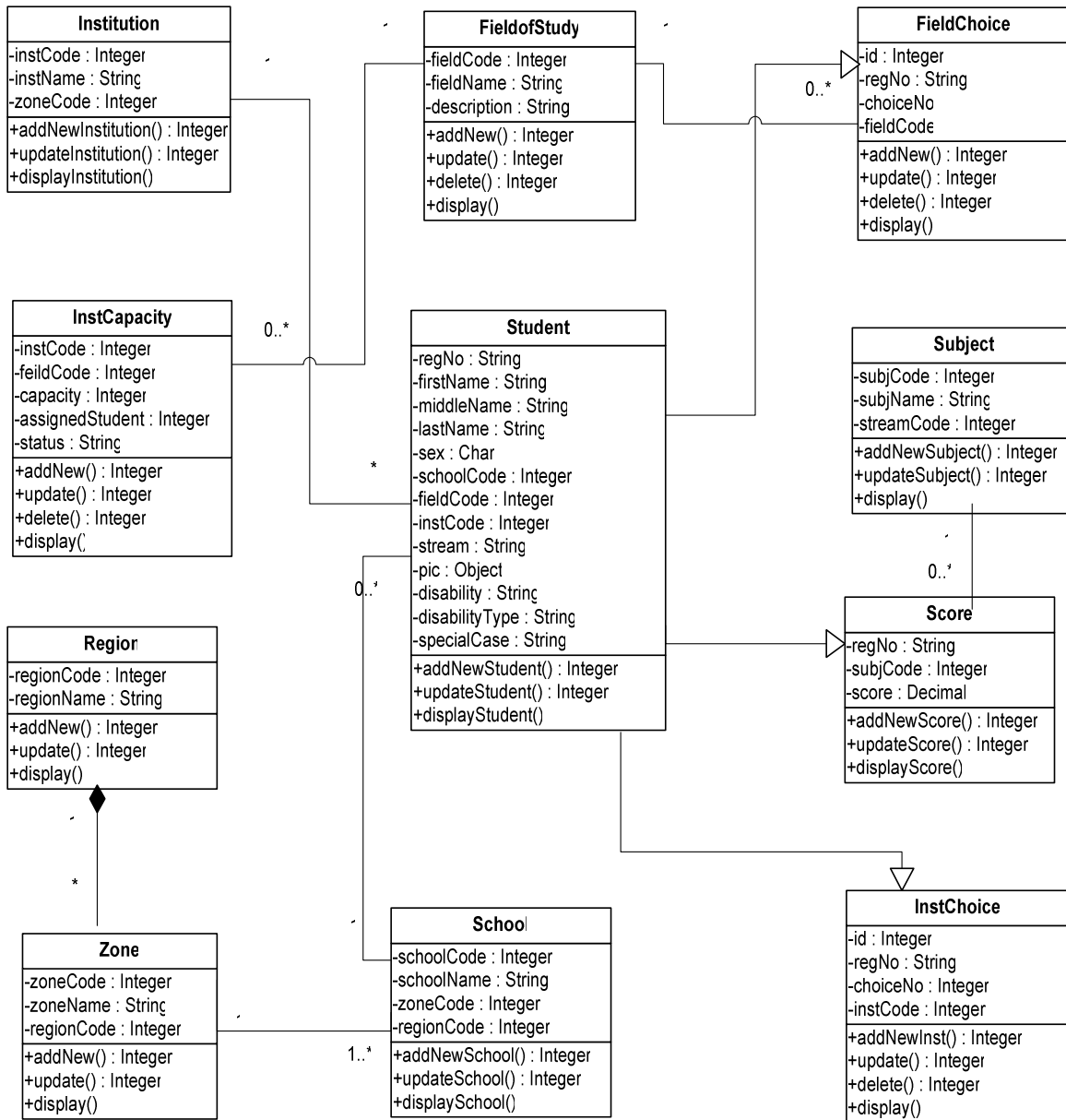
Institution

-instCode : Integer
-instName : String
-zoneCode : Integer

+addNewInstitution() : Integer
+updateInstitution() : Integer
+displayInstitution()

FieldofStudy

-fieldCode : Integer
-fieldName : String
-description : String

+addNew() : Integer
+update() : Integer
+delete() : Integer
+display()

FieldChoice

-id : Integer
-regNo : String
-choiceNo
-fieldCode

+addNew() : Integer
+update() : Integer
+delete() : Integer
+display()

InstCapacity

-instCode : Integer
-feildCode : Integer
-capacity : Integer
-assignedStudent : Integer
-status : String

+addNew() : Integer
+update() : Integer
+delete() : Integer
+display()

Student

-regNo : String
-firstName : String
-middleName : String
-lastName : String
-sex : Char
-schoolCode : Integer
-fieldCode : Integer
-instCode : Integer
-stream : String
-pic : Object
-disability : String
-disabilityType : String
-specialCase : String

+addNewStudent() : Integer
+updateStudent() : Integer
+displayStudent()

Subject

-subjCode : Integer
-subjName : String
-streamCode : Integer

+addNewSubject() : Integer
+updateSubject() : Integer
+display()

Score

-regNo : String
-subjCode : Integer
-score : Decimal

+addNewScore() : Integer
+updateScore() : Integer
+displayScore()

Region

-regionCode : Integer
-regionName : String

+addNew() : Integer
+update() : Integer
+display()

Zone

-zoneCode : Integer
-zoneName : String
-regionCode : Integer

+addNew() : Integer
+update() : Integer
+display()

School

-schoolCode : Integer
-schoolName : String
-zoneCode : Integer
-regionCode : Integer

+addNewSchool() : Integer
+updateSchool() : Integer
+displaySchool()

InstChoice

-id : Integer
-regNo : String
-choiceNo : Integer
-instCode : Integer

+addNewInst() : Integer
+update() : Integer
+delete() : Integer
+display()

0..*   0..*   *   0..*   1..*   0..*

Fig 4.2 Class Diagram of student placement processing & retrieval system

32

4.3.3 Sequence Diagram

A sequence diagram ties use case with objects. It shows the interaction between participating objects in a given use case. It is helpful to identify the missing objects that are not identified in the analysis object model. From the use case and the class diagrams shown in the previous section the sequence diagrams of the system is shown as follows.

Sequence Diagram of Student Registration Use Case

As it is shown in Fig 4.3 to register a student data clerk of the Ministry should click on '*NewRegButton*' which displays for him/her the registration form. Using the registration form the data clerk should fill all the required data and click *submit* button, then the format of the submitted data will be validated by registration control module. If the submitted data is in accordance with the requirement of the system student data will be stored in the database and an acknowledgement message will be displayed.
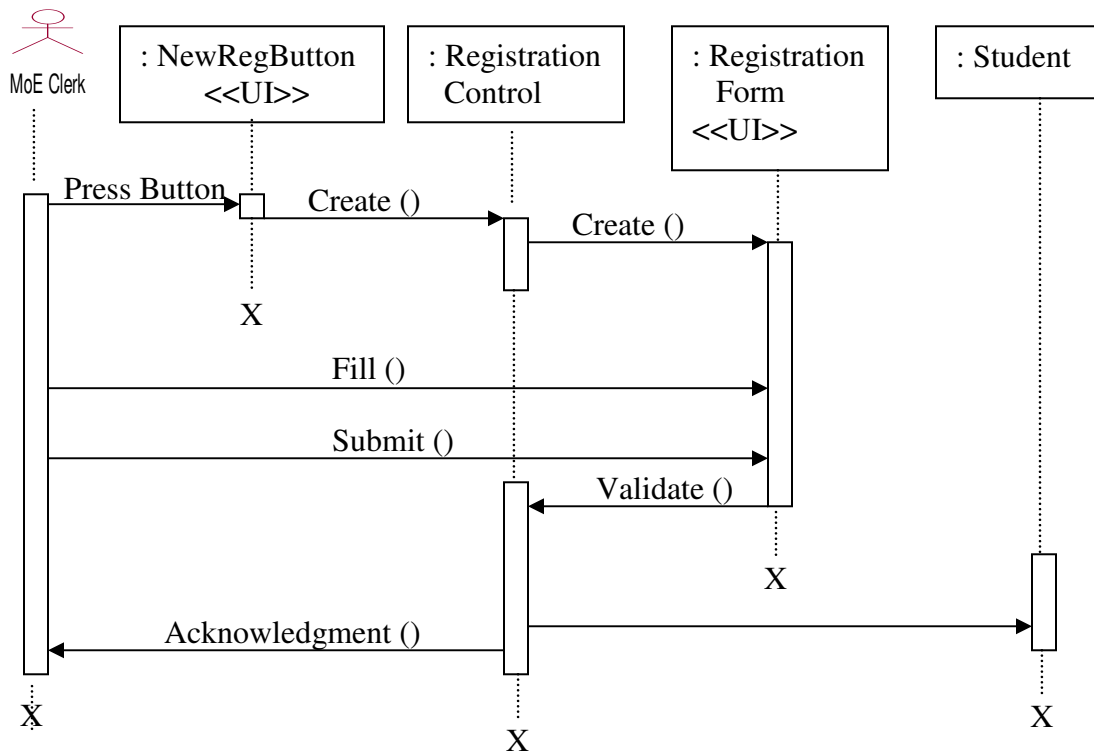
Fig 4.3 Sequence diagram for student registration use case

Sequence Diagram of Score Registration Use Case

Score registration sequence diagram which is shown in Fig 4.4 shows how the score of a student from national exam will be recorded into the system. The data clerk of the ministry should click 'NewScoreRegButton' and it will create score registration form and score registration control. Filling the respective data on score registration form and clicking submit button will activate the score registration controller to validate the submitted data. If there are missing values for fields, data type mismatch, or no student found by the given registration number message that describes the situation will be displayed. If the data is valid the score of a student will be stored in the database followed by an acknowledgement message to the clerk.
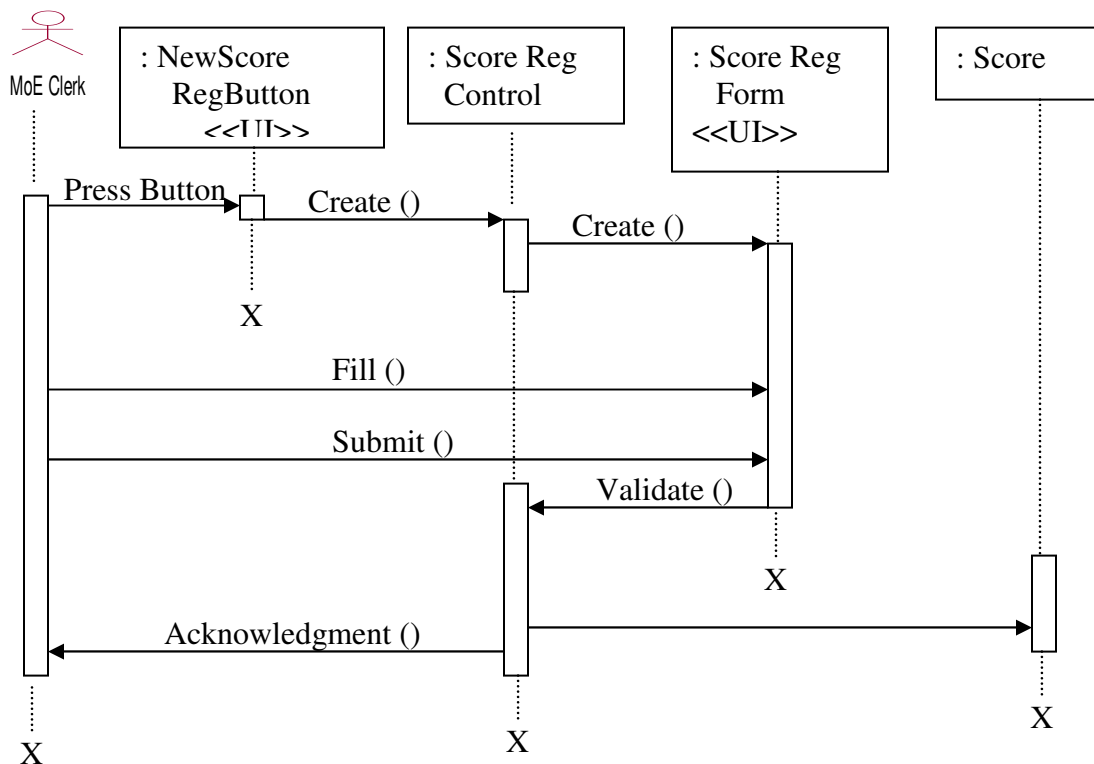
Fig 4.4 Sequence diagram for score registration use case

Sequence Diagram of Field Choice Registration Use Case

As shown in Fig 4.5 Field choice registration sequence diagram which illustrate to us that first the data clerk of the Ministry should initiate field of study registration control, fill the respective data on field choice registration form, and submit it. Then the 'Fieldchoicecontrol' will validate the completeness and correctness of data type of the submitted data and if the submitted data is valid it will store the data on the database and display an acknowledgement message to the clerk.
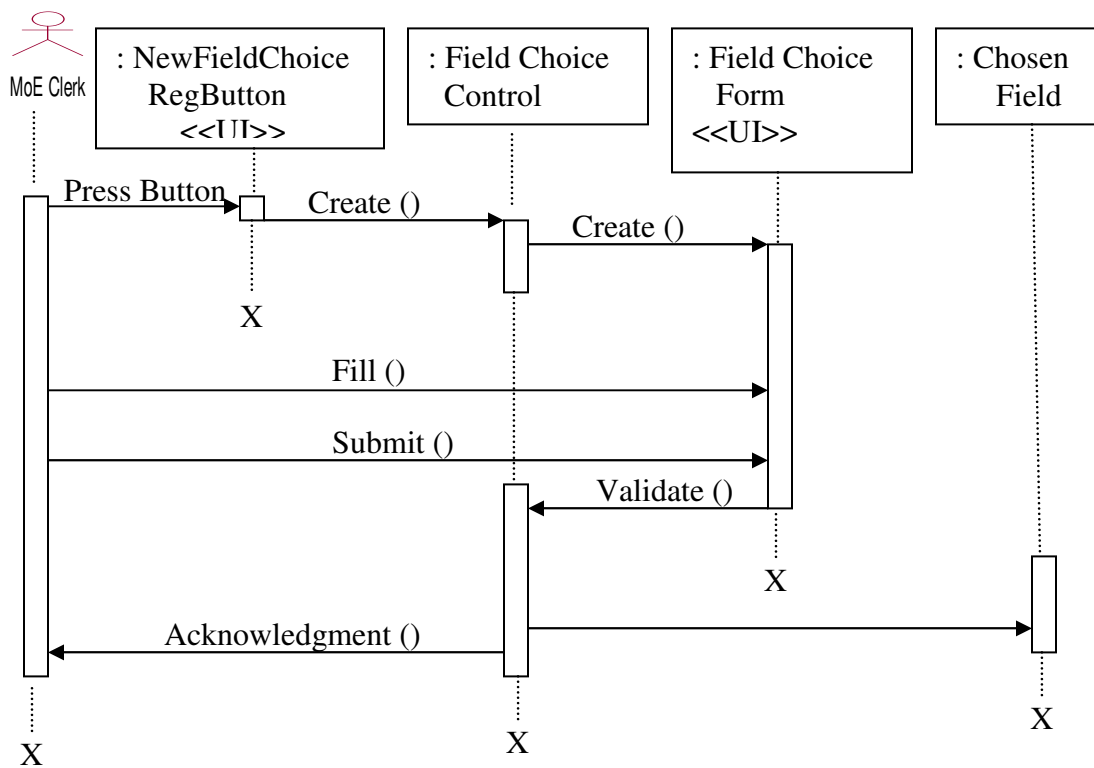
Fig 4.5 Sequence diagram for field choice registration use case

Sequence Diagram of Chosen Institution Registration Use Case

Fig 4.6 shows how an institution chosen by a student is recorded into the database by data clerk of the Ministry. The process begins when the data clerk of the Ministry click 'NewInstChoiceReg' button which automatically creates 'Institution choice control' and 'Institution Choice Form'. Then the data clerk should fill the form and submit it, in which followed by validation of the completeness and correctness of data type. If data is filled for all the available fields of the form and no data type mismatch the data will be stored to the database and an acknowledgement message will be displayed.
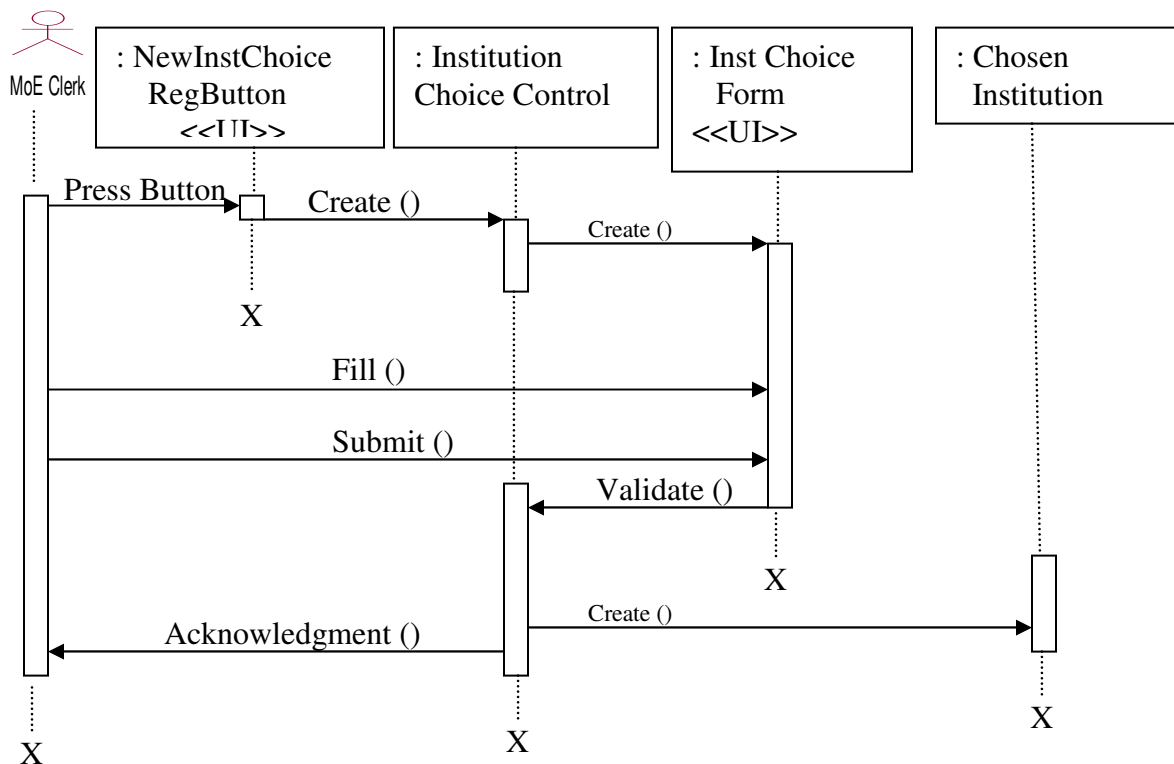
Fig 4.6 Sequence diagram for chosen institution registration use case

Sequence Diagram of Process Placement Use Case

After making sure that all the placement processing data is recorded in the database the next activity is to perform the placement process. As it is shown in Fig 4.7 the MoE data clerk should click 'process placement button' which creates placement processing control and display placement processing form. In this form clicking 'start' button will start the process of assigning students in different fields of study and institutions.
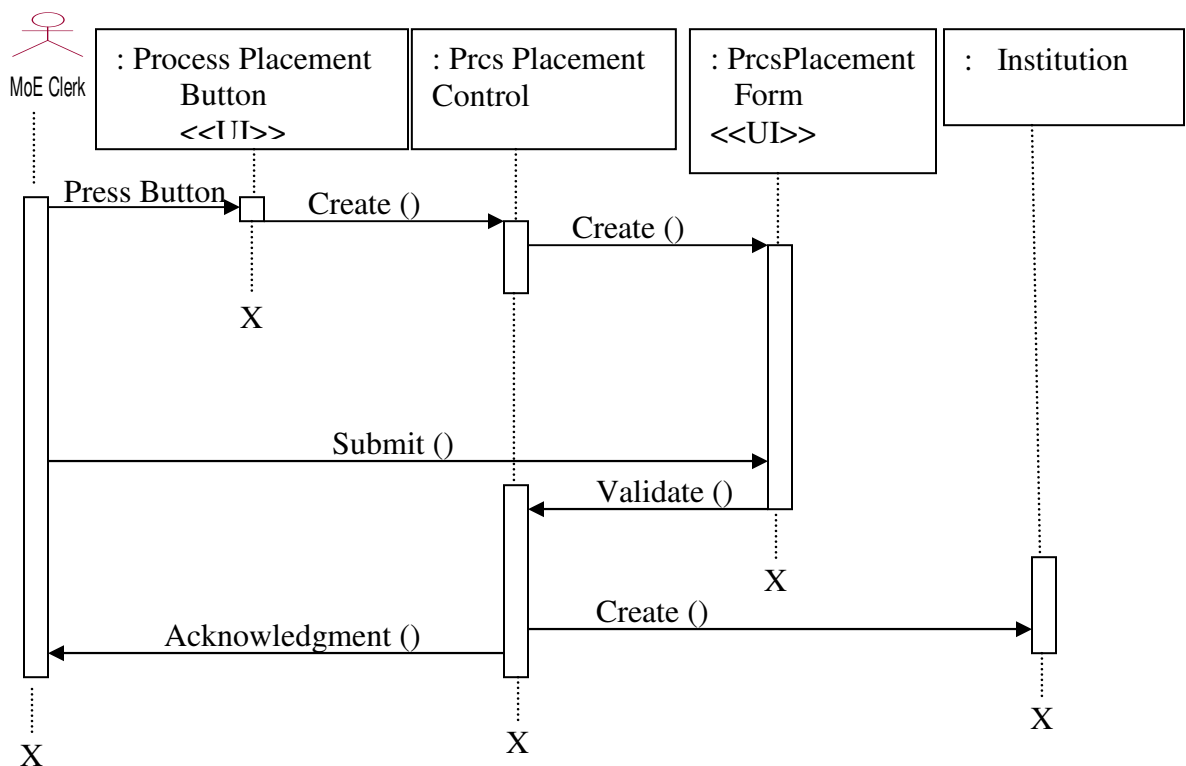
Fig 4.7 Sequence diagram for process placement use case

Sequence Diagram of Export to XML Use Case

As shown in Fig 4.8 export to XML use case is initiated by clicking 'NewExportXMLButton' which will create 'ExportXMLControl' which is responsible for exporting data from SQL server database to XML file. So when the data clerk click submit button the processed placement data will be broken down into groups and will be written in an XML file.
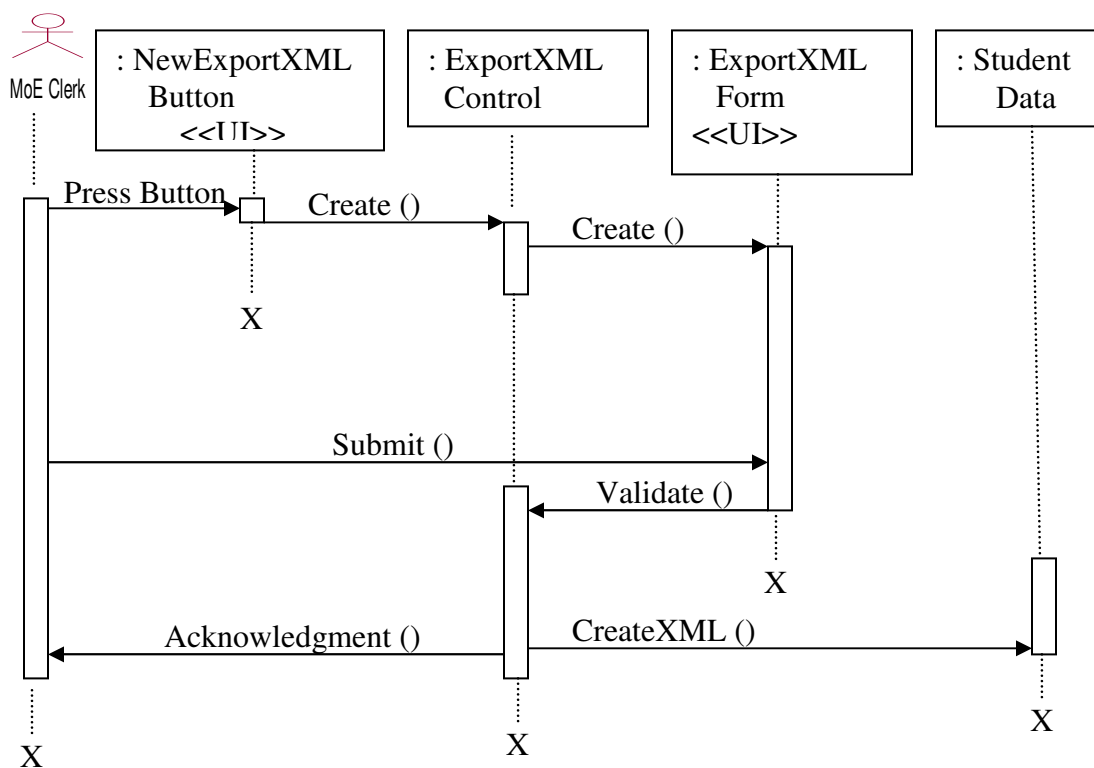
Fig 4.8 Sequence diagram for export to XML use case

Sequence Diagram of Display Placement Information Use Case

As it is illustrated in Fig 4.9 Display placement information sequence diagram shows how users such as students, schools, and higher education institutions can see their respective placement information. So when the user click display button the report controller will be initiated, after that if the user provides the appropriate input i.e.

- a student should select school name and provide his/her registration number

- personnel of higher education institution or school should select the name of their institution or school,

and then after clicking 'Display' button will enable the report controller to generate the placement report.
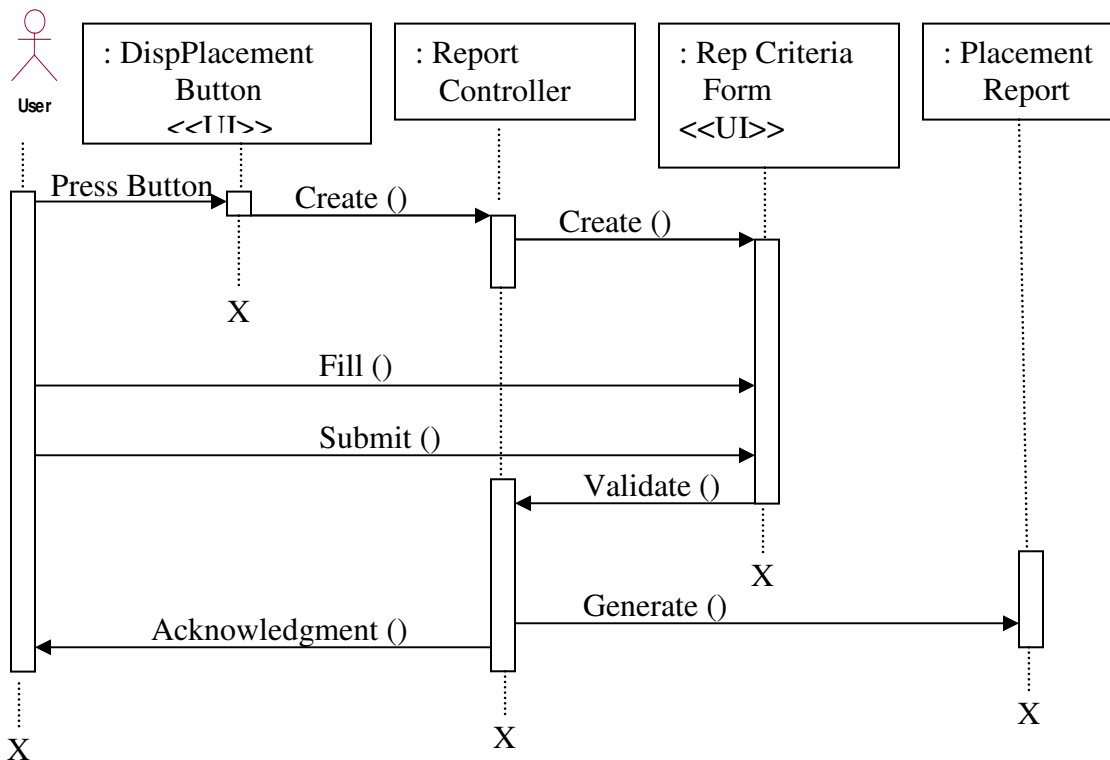
Fig 4.9 Sequence diagram for display placement information use case

Sequence Diagram of Download Student Data Use Case

The user of download student data use case is only personnel of higher education institutions. As it is discussed in 1.4.2 the data to be downloaded is not directly available to the public for privacy, and it is available in XML web service format so the user should have to develop an application that can consume from the available web service application. The application can be developed in variety of forms ranging from using Microsoft Word or Excel to open source development environments like Java.

After developing the appropriate application as shown in Fig 4.10 when the user starts his/her application it will activate the web service controller. The web service controller will authenticate and validate the parameters of the user, and if it is valid and correct it will let it go the download process.
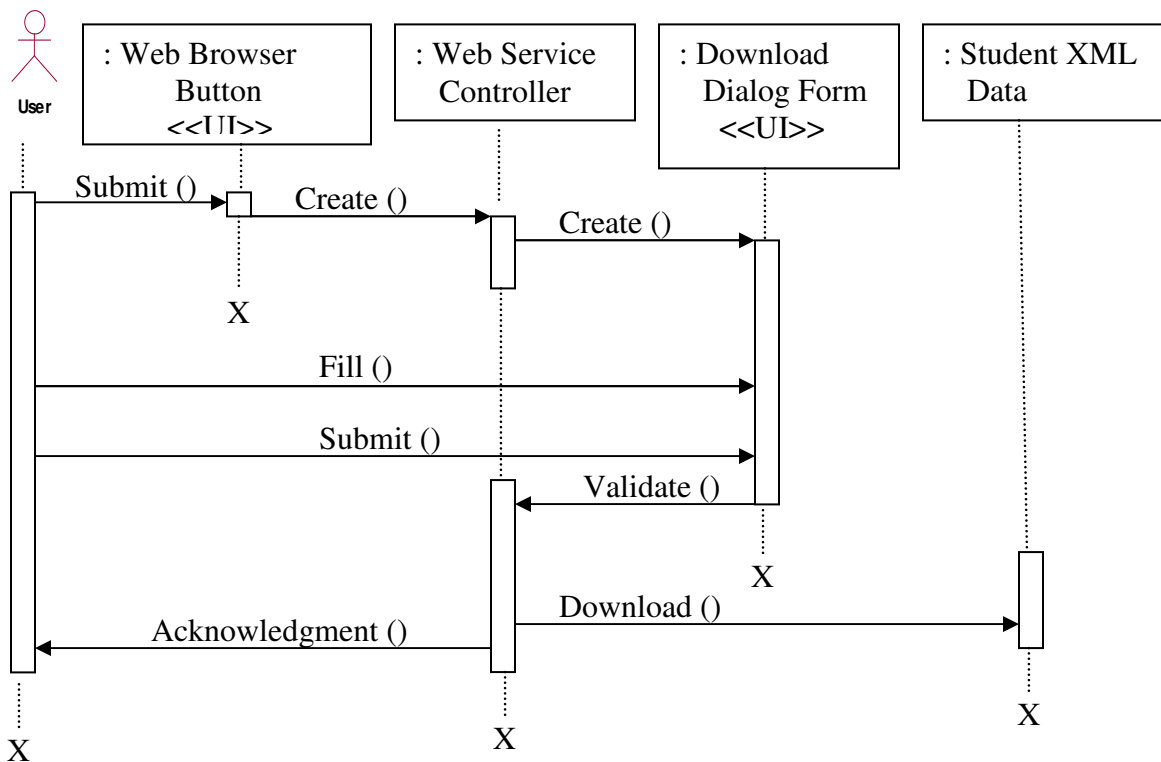
Fig 4.10 Sequence diagram for download student data use case

40

5. System Design

System design is the transformation of the analysis model into a system design model. Up to now we were in the problem domain. System deign is the first part to get into the solution domain in a software development. This chapter focuses on transforming the analysis model into the design model that takes into account the nonfunctional requirements and constraints described in the problem statement and requirement analysis sections discussed earlier.

5.1 Design Goals

Design goals describe the qualities of the system that the developers should consider. These goals can be inferred from the non functional requirements already discussed in section 4.2.

The design goals can be generally grouped into five categories. These are: **Performance Criteria, Dependability Criteria, Cost Criteria, Maintenance Criteria, End User Criteria.**

Performance: - The system should respond fast with high throughput, i.e. it should perform placement processing in a time less than 2 minutes and when users retrieve placement information online it should respond at least before the session expires.

Dependability: - The Ministry needs the system to be highly dependable. Even though placement information is released through plasma TV due to its inconvenience and lack

of users requirements, the Ministry wants the software system to be dependable to users. The system should be robust i.e. it should be able to survive invalid user inputs, fault tolerant, reliable and available. Since the system stores sensitive data, high emphasis is given to security to withstand malicious attacks. The system shouldn't allow non-authorized users to access students' personal data or modify.

Cost: - The system should be developed with minimum cost possible. In reality there is always trade-offs and therefore from it previous experience the Ministry prefers to invest more on development cost than maintenance cost to minimize bugs which may appear at the later stage.

Maintenance: - The system should be easily extensible to modify the government placement criteria, add new functionality, portable to different platforms. The code for the system should be easily readable, understandable and should be easily mapped to specific requirements.

End User Criteria: - The system should have simple and understandable graphical user interface such as forms and buttons which have descriptive names. It should give reliable response for each user request at least before the session expires.

5.2 Architecture of the System

The proposed subsystem will be implemented in Client/Server architecture. Wherever a user is as long as there is Internet connection he/she can browse the Ministry's placement page, fill the required inputs by the web page, and then submit it then the request of the user will be sent to the central server. The server will give response based on the user

request. From this description the architecture of the system is depicted diagrammatically as in Fig 5.1.
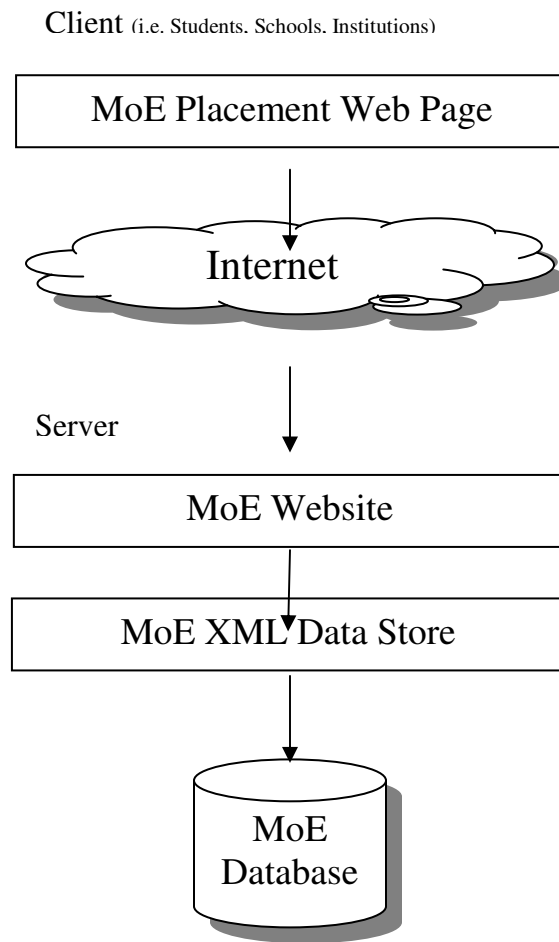
Client (i.e. Students. Schools. Institutions)

```
┌─────────────────────────────────┐
│     MoE Placement Web Page       │
└─────────────────────────────────┘
                 │
                 ▼
            ☁ Internet ☁

Server
                 │
                 ▼
┌─────────────────────────────────┐
│          MoE Website             │
└─────────────────────────────────┘
                 │
                 ▼
┌─────────────────────────────────┐
│       MoE XML Data Store         │
└─────────────────────────────────┘
                 │
                 ▼
             ⬭ MoE
             Database ⬭
```

Fig 5.1. The over all architecture of the system

5.2.1   Subsystem Decomposition

From the functional requirements we identified in the system analysis section, the envisaged system consists of the following subsystems:

- Student Registration Subsystem

- Field Choice Registration Subsystem

- Institution Choice Registration Subsystem

- Placement Processing Subsystem

- Export to XML Subsystem

- Display Placement Information Subsystem

- Download XML Data Subsystem

The description of these subsystems is as follows

Student Registration Subsystem

This subsystem is used to register student's personal information like registration number, full name, sex, school, and address.

Field Choice Registration Subsystem

This subsystem is used to register the chosen field of study by each student.

Institution Choice Registration Subsystem

This subsystem is used to register the chosen institution by each student.

Placement Processing Subsystem

This subsystem is used to process the placement of students based on the criteria set by the Ministry of education, the availability of the field of study, the number of students competing for that field, etc.

Export to XML Subsystem

This subsystem is used to divide the data in the SQL Server database and to export to XML format. As discussed in the previous sections breaking down the data into subgroups is for the purpose of avoiding the bottleneck problem in the SQL Server database engine. The data is broken down into groups based on the school of students, i.e. one XML file for students from one school, and based on higher institutions, i.e. one XML file for students placed in one higher institution.

Display Placement Information Subsystem

This subsystem is used to display placement information for individual students, school's placement information i.e. placement information of all students from one school, higher institution placement information i.e. placement information of all students into one higher institution.

Download XML Data Subsystem

This subsystem is used for higher institutions to download the newly placed students' data in an XML format so that they can upload the data into their database independent of their platform by avoiding data reentry.

From the nature of the system these subsystems perform their duties one after the other, i.e. the output of one subsystem is input for another subsystem, and this layered representation from bottom to top is shown in Fig 5.2.
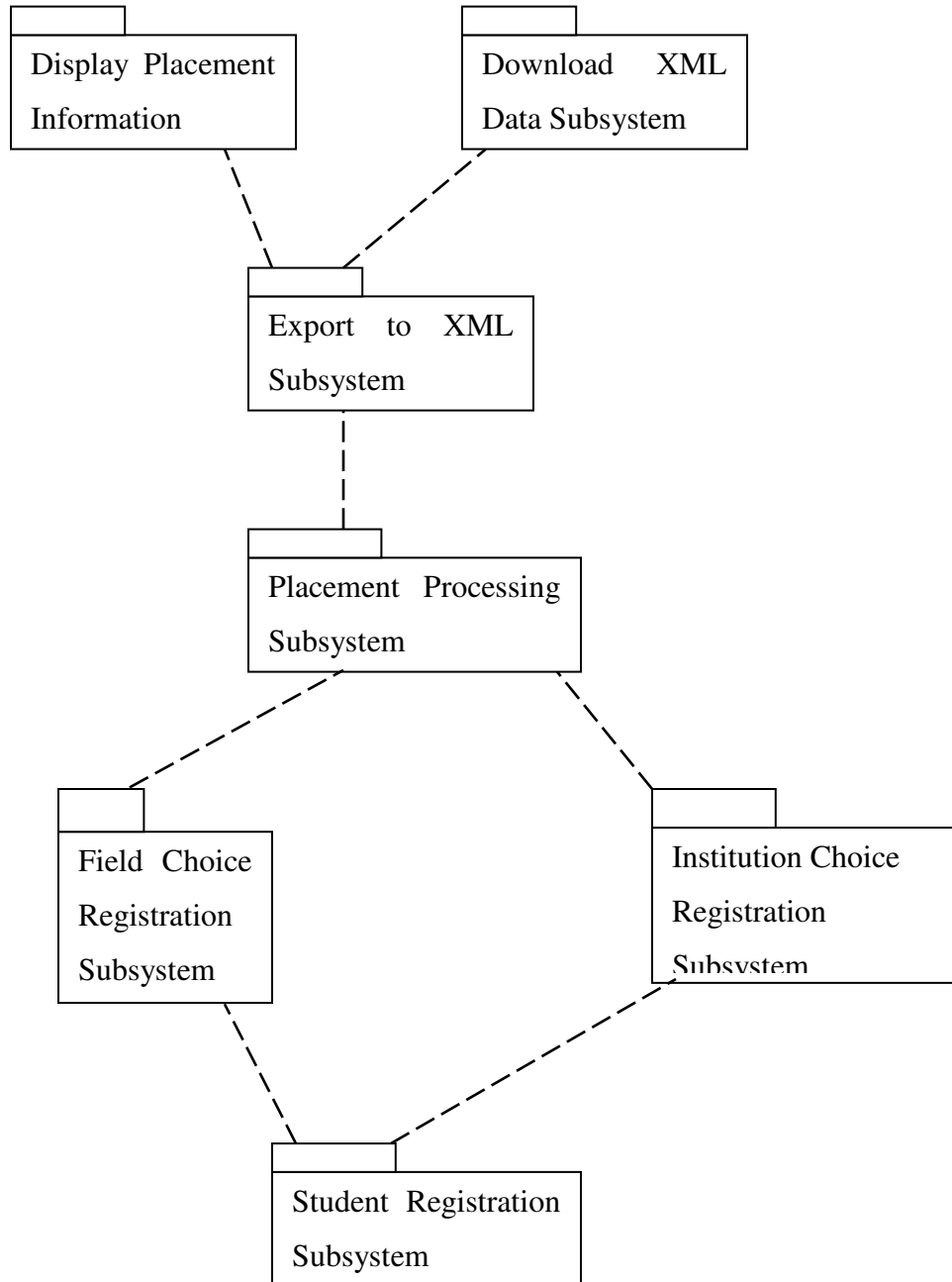
Fig 5.2. The Layered Representation of the sub-systems

5.2.2   Persistent Data Management

As it is discussed in the previous sections, the Ministry of Education placement processing and retrieval system consists of a number of sub systems. Student Registration Subsystem, Field Choice Registration Subsystem, Institution Choice Registration Subsystem and Placement Processing Subsystem are windows based subsystems that run only over a LAN, and the persistent data of these subsystems will be stored in an SQL Server database.

On the other hand, Display Placement Information Subsystem and Download XML Data Subsystem subsystems will use XML as a persistent data store in which the transformation from SQL Server database data to an XML format will be handled using Export to XML Subsystem. The data in the XML format will be used only for searching, display and downloading data, no insert, update or delete operations will be taking place on it.

Mapping

In order to store information persistently we map objects into tables and the attributes into fields to the specific table based on the objects found on the system. Therefore, we identified the major tables that will be implemented on the selected DBMS. For this reason, the mapping of objects to tables is shown as follows in Fig 5.3.
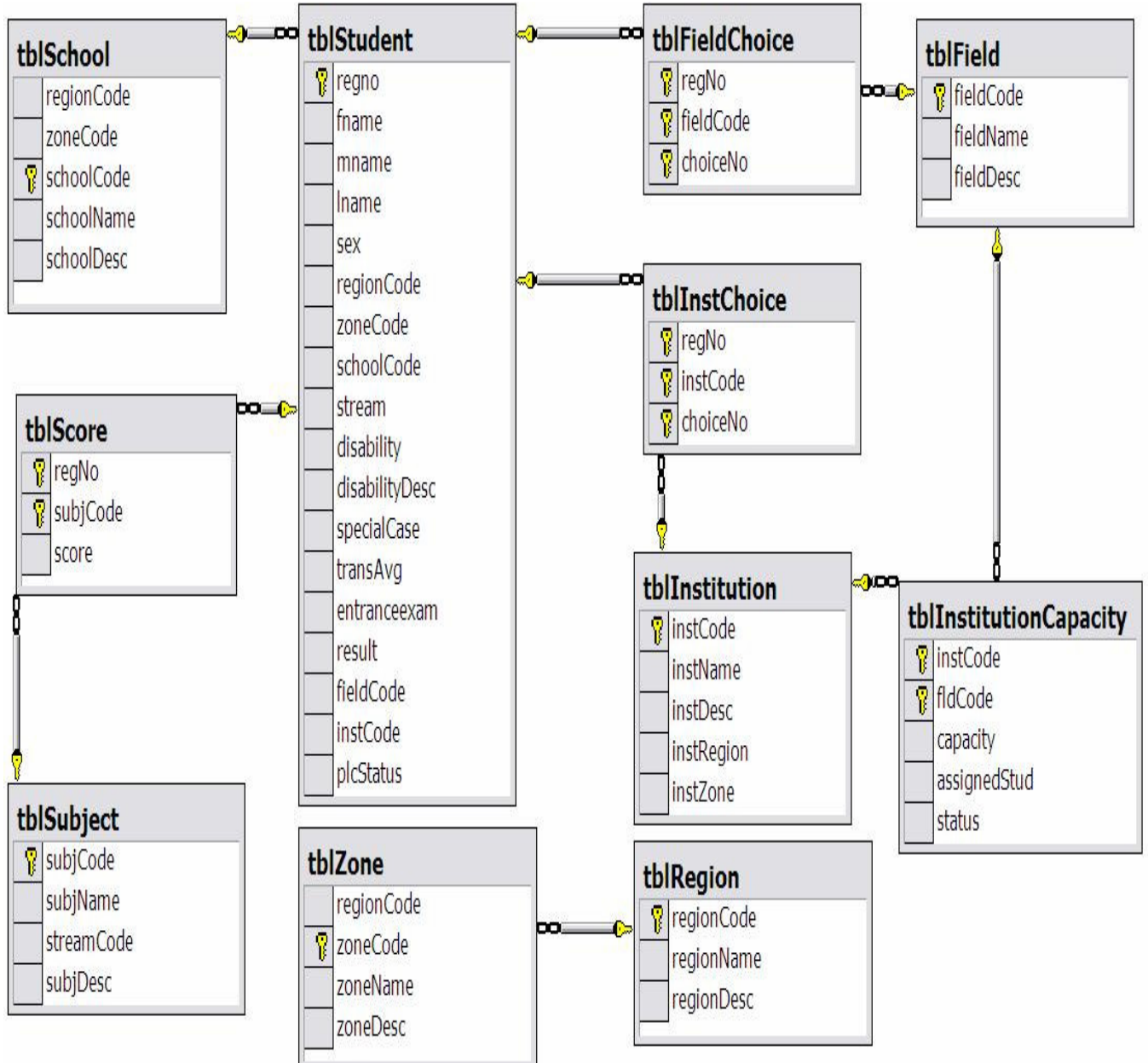
Fig 5.3 Identified tables and relationship among them

6. Implementation

As it is discussed in the previous sections for implementation we have used C#.NET for windows forms and ASP.NET for web forms, and implementation of functional requirements is discussed as follows.

Student Registration Form

It is used to register student personal information like registration number, full name, sex, school, address, disability conditions, stream (high school stream of the student), photograph, etc. To display the data of an already registered student first the user should enter registration number of that student and click search button as shown in Fig 6.1 which is next to registration number text box. So in addition of registration, this form enables us to edit and delete student's personal information.

In this form if the user tries to save duplicate registration number or if some required field is missing it will display message that describes the situation and terminate the saving process. But for some fields like disability or photograph if the user omits them since they are not required fields the saving process will proceed and finally it will display a message that confirms the submitted data is successfully saved in the database.

Once a student is registered using this form its information will be available for the next activities like score registration, chosen field of study registration, chosen institution registration, etc.

Fig 6.1 User Interface of Student Registration form

Score Registration Form

One of the criteria to perform student placement is done based on the score of students on

each subjects they get in college preparatory school and their score of entrance

examination. This form is used to register the score of students on each subject, entrance

exam result, and to display what is already registered, etc. As it is shown in Fig 6.2 to

record the score of a student the user should type registration number of the student and

click find button. If a student with that registration number is found his/her name, school

etc will be displayed, but if not message that describes so will be displayed. To record score the user should select the subject from the combo box, enter the score and click save button.

Fig 6.2 User Interface of Score Registration Form

Chosen Field Registration Form

Students who qualify to join governmental higher institutions will list his/her top 10 chosen field of study. And as its name indicates field choice registration form is used to register the chosen field of study of each student with its priority. As it is shown in Fig 6.3 and as in score registration form to perform field choice registration a student should be registered using student registration form.

Fig 6.3 User Interface of Chosen Field Registration Form

Institution Choice Registration Form

After selecting his/her choice of field of study he/she has to list his/her 15 chosen institutions in ascending order. The list of chosen institutions by students will be encoded in to the system using this form. As shown in Fig 6.4 the chosen institution and the respective choice number is chosen from the combo boxes. When ever we save new record it will be in the grid shown below.

Fig 6.4 User Interface of Institution Choice Registration Form

Placement Processing Form

After student personal data, score, field choice, institution choice is recorded the next step is processing the placement of students. Placement of students is done based on the procedures set by the Ministry of education, and it is done in batch i.e. as shown in Fig 6.5 once the '*Start'* button is clicked placement of all students will be done one after the other.
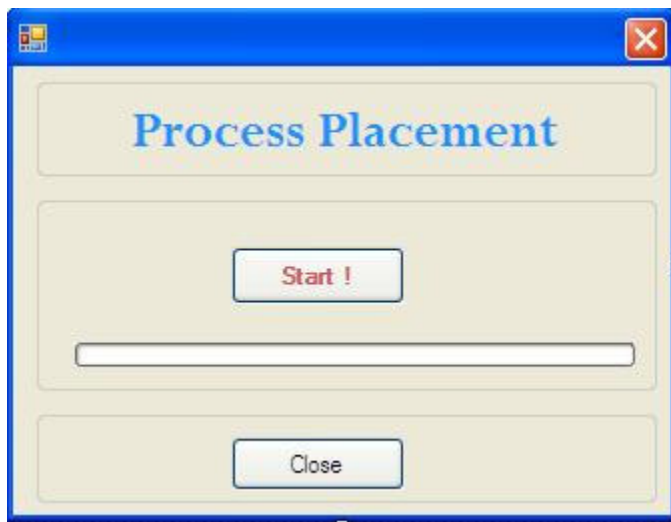

Fig 6.5 User Interface of Placement Processing Form

Export Data to XML Form

As it is discussed in the previous sections until placement is processed the data will be stored in SQL Server database and after that the processed data will be broken down into smaller groups i.e. group by schools, and institutions, and will be exported to XML format and it will be ready for displaying it using the WebPages. And this form is used to export the processed data from SQL Server to XML format. To export the data to XML first the user should make sure that placement is processed, and then clicking the 'Start' button which is shown in Fig 6.6 will do the exporting.

Fig 6.6 User Interface of Export Data to XML Form

Student's Placement Information Display Page

The website of the placement system consists of a number of WebPages. When a user open the website the main form will be displayed, and as shown in Fig 5.10 the main form consists of links to student's page, school's page, and institutions page.

Fig 6.7 (a) Home page of placement website

For a student to see his/her placement information he/she should select the school name from the combo box, enter registration number and click 'Search' button. If the information provided is correct the full name, name of institution, and field of study will be displayed.
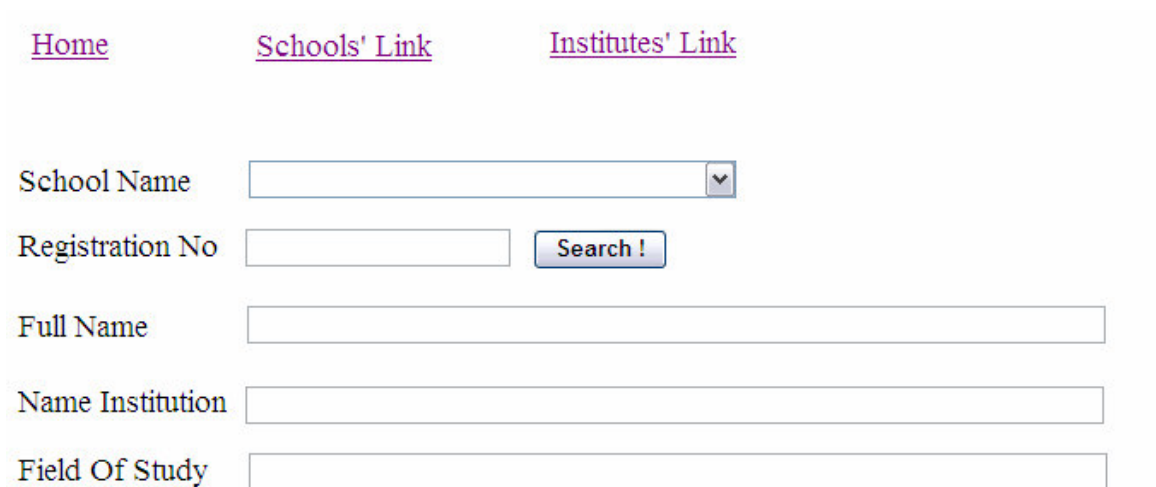

Fig 6.7 (b) Student's page of placement website

7. Conclusion and Recommendations

In developing web applications that are accessed by thousands of concurrent users we should consider the wide range of access infrastructure verities among users. Some of the users may use PC that has very high performance over broadband Internet connection and to the contrary some of the users may access the webpage using sluggish PC over very low bandwidth dial up connection. Especially if the web application is dynamic the number of request and reply between the user PC and the web server will be much more and managing such bidirectional communication over specific period of time (before the session expires) through high traffic network will be more challenging task.

When those users with the second type of specification access the web server the connection between their PC and the server will stay for relatively longer period of time than the first type of users, which results decline in the throughput of the web server which in turn result longer connection time even for the first type of users.

In this project so as to combat such challenges we have used two main strategies: the first is to minimize performance traits that could affect the throughput of the web server. And to address this issue a number of small pieces of XML data store is being used instead of one big SQL server database. The second strategy is to design the webpage that has the minimum possible loading speed. And to do so the web page is being built by considering the points rose in chapter 3.

The other issue that is being addressed in this project is transfer of structured student personal data from the web server to user's computer so that the data will be used in any

platform. To solve this XML web service is being used which is a very good choice since it uses XML messaging for communication and XML data is highly serializeable for data transfer which makes semi-structured data transfer easier.

Recommendations and Future work

- One of the issues raised in this project is data entry, specifically the third part of the project is aimed at avoiding data reentry by higher education institutions. And we have mentioned that even in 2007/2008 academic year the data of more than 56,000 students is recorded by the ministry through its data entry clerks. But if the Ministry can get student personal and score information in soft copy directly from National Examinations Office and for the newly developed placement system if an interface that can read and upload the data to the database is developed it is possible to avoid data reentry by around 60%.

- The main problem of the old system is performance of the website, and to improve this problem the new system implements XML as a backend of the website to avoid the bottleneck problem of the SQL Server. But from literatures discussed in Chapter 3 static web pages are much better in performance than dynamic pages. And once placement is processed there will no any other processing and data manipulation to be done on the web server, so if static pages can be generated dynamically and stored persistently the performance of the website can be even better than the newly developed.

References

[1] Erik, T. (2003), **Learning XML**, 2nd Edition, California: O'Reilly& Associates Inc.

[2] Microsoft Corporation, (2002), **Developing Microsoft ASP.NET Web Applications Using Visual Studio .NET**, Microsoft Press,

[3] Microsoft Corporation, (2002), **Programming with Microsoft ADO.NET**, Microsoft Press,

[4] Kimberly Jordan, MJY Team, (1997), **Software Reuse For The Mjy Team Software Risk Management Site**, George Mason University,

[5] Andrew B. King, **Speed Up Your Site**, http://www.amazon.com/ Speed-Up-Your-Site-Optimization/dp/0735713243, visited on, 29 March 2008

[6] "Free website speed checker", http://www.selfseo.com/website_speed_test.php, visited on, 29 March 2008

[7] "How to Optimize Site Speed ", http://www.improvetheweb.com/how-to-optimize-site-speed, visited on, 29 March 2008

[8] "What are the advantages and disadvantages of computerization of business activities (electronic data processing)?", http://www.careergears.com/advantages-disadvantages-of-computerization-of-business-activities-electronic-data-processing/, visited on May 13, 2008

[9] http://home.heroshima-u.ac.jp/cice/paper67.pdf, visited on, 07 March 2008

[10] http://msdn.microsoft.com/en-us/library/ms191303.aspx, visited on, 07 March 2008

[11] Yahoo, "Performance of WebPages", http://developer.yahoo.com/performance/ rules.html, visited on, 29 March 2008

Annex

Interview Questions

The following were interview questions that were prepared to raise discussion with personnel of the Ministry for the purpose of identifying requirements and to analyze the system.

1. What does the system is expected to do?

2. Who are the users of the system?

3. What are the roles of each user?

4. What are the activities that should be done so as to perform the placement of students?

5. What are the shortcomings of the existing system?

6. What additional new features do you expect from the new system?