
English Learning Website

Zhichun Fu

A Major Report

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montreal, Quebec, Canada

August 2003

© Zhichun Fu, 2003

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitons et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

ISBN: 0-612-83900-1

Our file *Notre référence*

ISBN: 0-612-83900-1

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canada

ABSTRACT

English Learning Website

Zhichun Fu

This report presents a case study of implementing an English Learning Website (ELW) using Servlet and JSP technologies.

English Learning Website is a web application that runs on Tomcat web server. It provides with international students whose mother language is not English a friendly online environment in learning English vocabulary and increasing reading ability. The system administrator can update data from the database directly. The system consists of eight functions: Vocabulary Level Testing, Vocabulary Studying, Lesson Testing, Reading, Online Dictionary, User Comments, Update Vocabulary Database, and Update Article Database. A two-level index structure is designed for the vocabulary database. A two-level searching algorithm is developed for this structure in order to achieve fast searching in large database and multi-user environment.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, Dr. Peter Grogono. Without his patience and guidance, I could not finish my work.

I am also grateful to Dr. Leila Kosseim. The time and effort she dedicated to this report, as well as her advices, are valuable.

It is hard to express my grateful with words for the love and support from my husband Jun Zhou, my parents, and my sister. Their care, help and encouragement are the power to push me finishing this report and will be with me forever.

TABLE OF CONTENTS

<u>1. INTRODUCTION</u>	1
<u>1.1 BACKGROUND</u>	1
<u>1.2 MAIN TASK</u>	1
<u>1.3 END USER</u>	1
<u>1.4 ASSUMPTIONS AND CONSTRAINS</u>	2
<u>1.5 ABBREVIATIONS</u>	2
<u>2. SOFTWARE REQUIREMENTS AND ANALYSIS</u>	2
<u>2.1 FUNCTIONAL REQUIREMENTS</u>	2
<u>2.1.1 General Users</u>	2
<u>2.1.2 System administrators</u>	3
<u>2.2 USE CASE</u>	3
<u>2.2.1 Use Cases Description and Diagrams</u>	4
<u>3. NON-FUNCTIONAL REQUIREMENTS</u>	10
<u>3.1 PRODUCT REQUIREMENTS</u>	11
<u>3.1.1 User interface</u>	11
<u>3.1.2 Reliability</u>	11
<u>3.1.3 Efficiency</u>	12
<u>3.1.4 Safety</u>	12
<u>3.2 ORGANIZATIONAL & PROCESS REQUIREMENTS</u>	12
<u>3.2.1 Architecture</u>	12
<u>3.2.2 Software platforms</u>	12
<u>3.2.3 Development Tools</u>	12
<u>3.2.4 Software Maintenance</u>	14
<u>4. SYSTEM DESIGN</u>	15
<u>4.1 DESIGN RATIONALE</u>	15

4.1.1 <i>System Architecture</i>	15
4.2 <u>DATABASE SCHEMA AND DESCRIPTION</u>	16
4.2.1 <i>Database Schema</i>	16
4.2.2 <i>Database Table definition</i>	17
4.2.3 <i>E R diagram</i>	19
4.3 <u>FUNCTION DESIGN</u>	21
4.3.1 <i>Vocabulary Level Testing Function</i>	21
4.3.2 <i>Vocabulary Studying Function</i>	24
4.3.3 <i>Lesson testing Function</i>	27
4.3.4 <i>Reading Function</i>	30
4.3.5 <i>Online Dictionary Function</i>	30
4.3.6 <i>User comments Function</i>	36
5. <u>IMPLEMENTATION</u>	37
5.1 <u>SYSTEM ENVIRONMENT</u>	37
5.2 <u>APPLICATION ARCHITECTURE</u>	37
5.3 <u>DATA ACCESS AND JDBC</u>	39
5.4 <u>HTTPSERVLET CLASS</u>	43
5.5 <u>THE SESSION SERVLETS</u>	44
5.6 <u>FUNCTIONS IMPLEMENTATION</u>	45
5.6.1 <i>Vocabulary Level Testing</i>	45
5.6.2 <i>Vocabulary Studying</i>	47
5.6.3 <i>Lesson testing</i>	47
5.6.4 <i>Reading</i>	49
5.6.5 <i>Online Dictionary</i>	49
5.6.6 <i>User comments</i>	50
6. <u>COMPARISON WITH OTHER ENGLISH LEARNING WEBSITES</u>	52
6.1 <u>OTHER ENGLISH LEARNING WEBSITES</u>	52
6.1.1 <i>Learn English--- British Council</i>	52
6.1.1.2 <i>BBC World Service</i>	52
6.1.1.3 <i>Learning English</i>	53
6.2 <u>STRENGTHS AND WEAKNESSES OF ELW</u>	53

<u>7. CONCLUSION</u>	54
<u>7.1 SUMMARY</u>	54
<u>7.2 FUTURE WORK</u>	55
<u>REFERENCE</u>	57
<u>APPENDIX</u>	59
<u>A. USER'S GUIDE</u>	59

LIST OF FIGURES

<u>Figure 1: Use case of English Learning Website</u>	4
<u>Figure 2: Use case of Vocabulary Level Testing</u>	5
<u>Figure 3: Use case of Vocabulary Studying</u>	6
<u>Figure 4: Use case of Lesson Testing</u>	7
<u>Figure 5: Use case of Reading</u>	8
<u>Figure 6: Use case of Online Dictionary</u>	9
<u>Figure 7: Use case of User Comments</u>	9
<u>Figure 8: Use case of Update Vocabulary Database</u>	10
<u>Figure 9: Use case of Update Reading pages</u>	10
<u>Figure 10: System Architecture Diagram</u>	16
<u>Figure 11: E-R Diagram</u>	20
<u>Figure 12: Vocabulary Level Testing Class Diagram</u>	22
<u>Figure 13: Vocabulary Level Testing Sequence Diagram</u>	23
<u>Figure 14: Vocabulary Studying Class Diagram</u>	25
<u>Figure 15: Vocabulary Studying Sequence Diagram</u>	26
<u>Figure 16: Lesson Testing Class Diagram</u>	28
<u>Figure 17: Lesson Testing Sequence Diagram</u>	29
<u>Figure 18: Online Dictionary Database Structure</u>	31
<u>Figure 19: Online Dictionary Class Diagram</u>	34
<u>Figure 20: Online Dictionary Sequence Diagram</u>	35
<u>Figure 21: User Comments Class Diagram</u>	37
<u>Figure 22: Application Architecture Diagram</u>	38

1. Introduction

1.1 Background

The population of international students in Concordia University is increasing. For those whose first language is not English, the first problem they encounter is the language problem. They meet difficulties in understanding academic papers and technical reports. They have problem in communication with professors and classmates. Limited vocabulary as well as poor reading abilities are of the major weakness in their English, which hinder them from adapting well to the new environment and studying effectively. Students may take ESL classes to improve English comprehensively. Or they may read English books, newspapers and magazines to improve their reading abilities and vocabulary. However, many students are wondering how to quickly improve their English, especially vocabulary, besides taking ESL classes and other methods. This project tries to solve the problem by providing an intensive vocabulary-studying environment.

1.2 Main Task

This English Learning Website provides a friendly environment for the international students to improve their English. They can check their vocabulary levels, remember words and read articles. Online lessons are offered to the students so that they can learn words in an intensive way and quickly enlarge their vocabulary. An online dictionary is provided to give spelling, explanation and usage examples on all the words in the lessons.

1.3 End User

Based on the services provided by the website, there are two groups of end users in this ELW.

- General users

International students whose first language are not English and have the necessity to improve English in a short period. They can access to the Internet. They need not have any knowledge of ELW.

- System administrators

Users have basic knowledge and experience on operating system, database management, and Internet.

1.4 Assumptions And Constrains

Following are the assumptions and constraints before the design of this ELW:

- The website is developed in English.
- The users of this website should have Basic English reading ability.
- The default web browser of users is Internet Explorer or Netscape.

1.5 Abbreviations

ELW —— English Learning Website

JSP —— JavaServer Page

J2SDK —— Java2 Standard Edition

System —— the English Learning Website system

2. Software Requirements and analysis

2.1 Functional Requirements

2.1.1 General Users

General users are international students who want to increase vocabulary and improve reading ability quickly. Their main concerns are how to know their vocabulary levels and how to find the appropriate vocabulary lessons. Reading practice is also important.

Following are the basic functions to be implemented for general users:

- Test users' vocabulary level.
- Vocabulary studying in different levels
- Vocabulary testing after each lesson
- Reading Article/Novel/News
- Check words from online Dictionary
- User comments

2.1.2 System administrators

The main concern of the system administrators is to update the databases in ELW. Thus, the major functions to be implemented for system administrators are:

- Update vocabulary database
- Update article database

2.2 Use Case

Figure1 contains a list of the possible use cases and the actors that interact with them. General users and system administrators are the two groups of users that will interact with the systems.

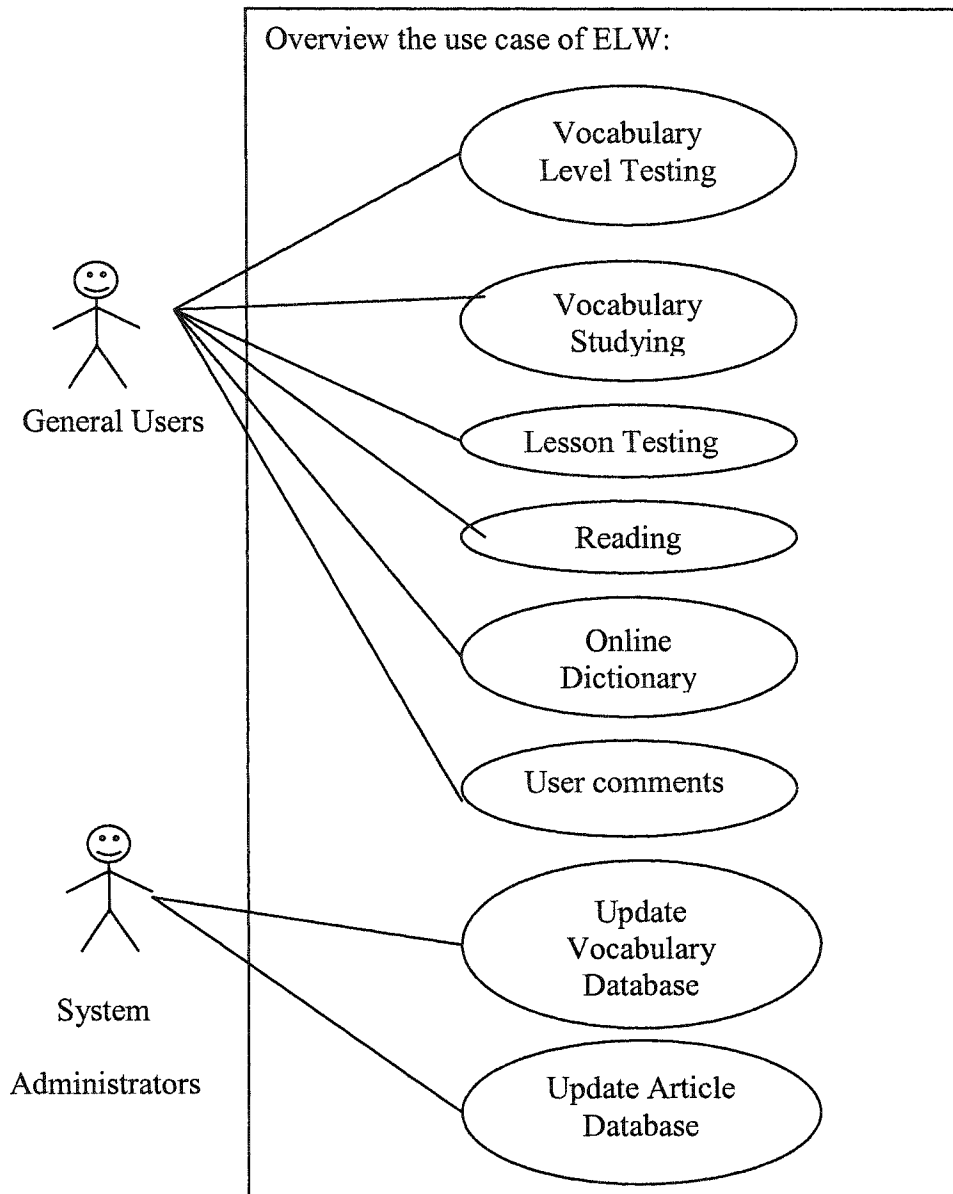


Figure 1: Use case of English Learning Website

2.2.1 Use Cases Description and Diagrams

--- Use Case 1: Vocabulary Level Testing

ELW is designed for general users with three different levels of English vocabulary, i.e. beginner level, intermediate level and advanced level. To help them know their current

level, we recommend new users to take Vocabulary Level Testing as the first step in their vocabulary learning. Users begin the vocabulary level test by entering the main page of ELW and clicking the “Vocabulary Level Testing” link. The system then randomly selects 25 questions from the database and displays them on the web page. Each question has four choices, but only one is correct. Users then select the radio buttons to the left of the choices that they consider to be correct for each question. Users click the “Submit” button to submit the answer after they have finished all or part of the questions. The system compares the submitted answers with the correct answers in the database and then gives a correct rate for each user. The current vocabulary levels of the users are decided according to the correct rates achieved. Then, users may select to begin vocabulary learning by clicking the “learning vocabulary” button or do another test by clicking the “new test” button.

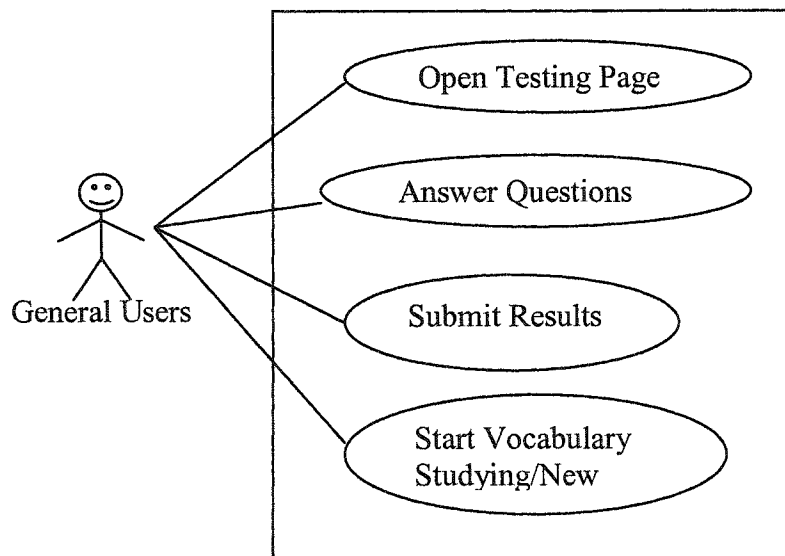


Figure 2: Use case of Vocabulary Level Testing

--- Use Case 2: Vocabulary Studying

Vocabulary studying is provided for each vocabulary level. Each level contains several lessons. Users can choose level and lesson according to their requirements. Users begin the vocabulary studying by entering the main page of ELW and clicking the “Vocabulary Studying” link. A level choice page will be displayed, which contains a short introduction and a drop-down menu for vocabulary level. Users choose the level and a lesson choice page will be displayed with a drop-down menu. Users choose a lesson and a lesson with 20 words will be displayed. The learning content includes spelling, pronunciation, meaning and usage examples of each word. Having finished a lesson, users may choose to begin another lesson, to do a test, or return to the lesson choice page.

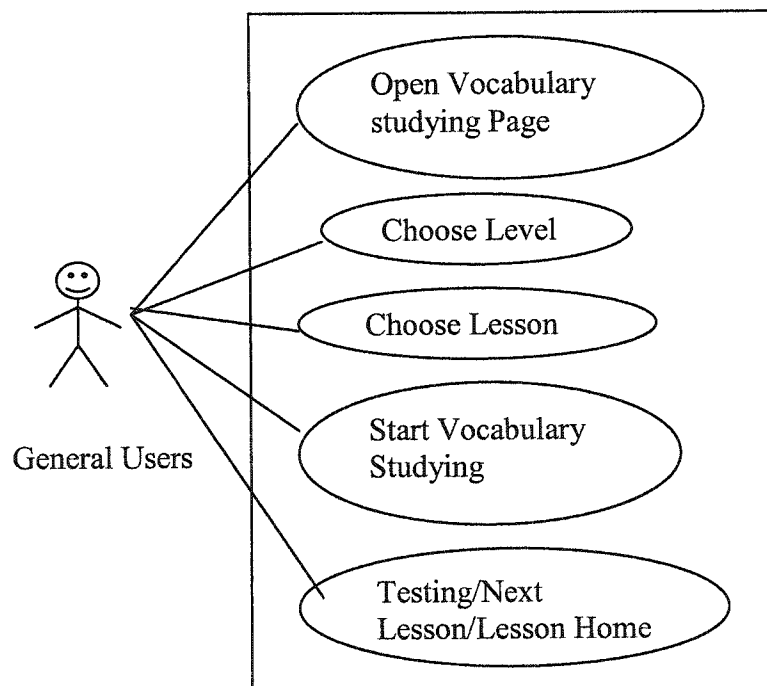


Figure 3: Use case of Vocabulary Studying

--- Use Case3: Lesson Testing

Having finished each lesson, users may want to know how much they've learned by taking a short test. We consider the test a good auxiliary way for vocabulary learning. Users begin the lesson testing by click the "Testing" button at the bottom of the lesson page. Five questions, each has four choices with only one correct, will be displayed in a new page. Users select the choices they consider to be correct by clicking the radio buttons to the left of the choices. After users have finished all the questions, they click the "Submit" button. The system will display the question, the correct answer, and the user's answer.

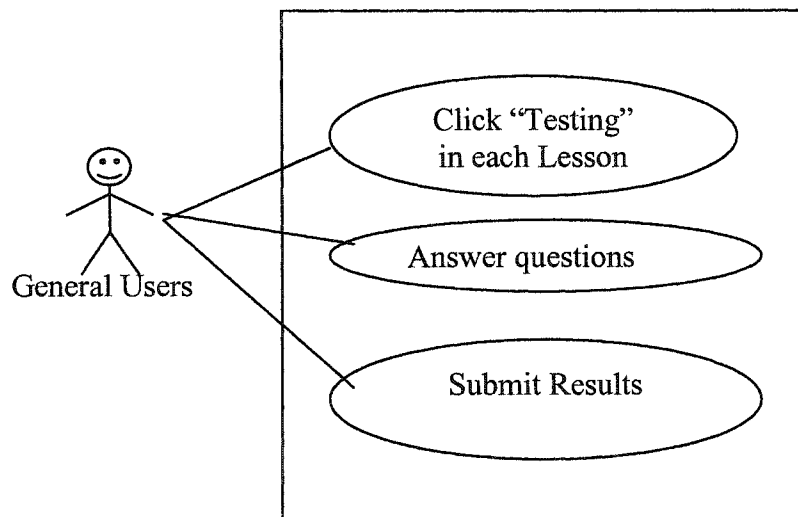


Figure 4: Use case of Lesson Testing

--- Use Case4: Reading

Reading is an excellent way of improving students' English. ELW provides users with several reading categories. They are academic papers, TOELF articles, stories, and news. The users begin the reading by entering the main page of ELW and clicking the

“Reading” link. Several categories of reading materials will be displayed. Users may select an article by clicking the link of the article in each category.

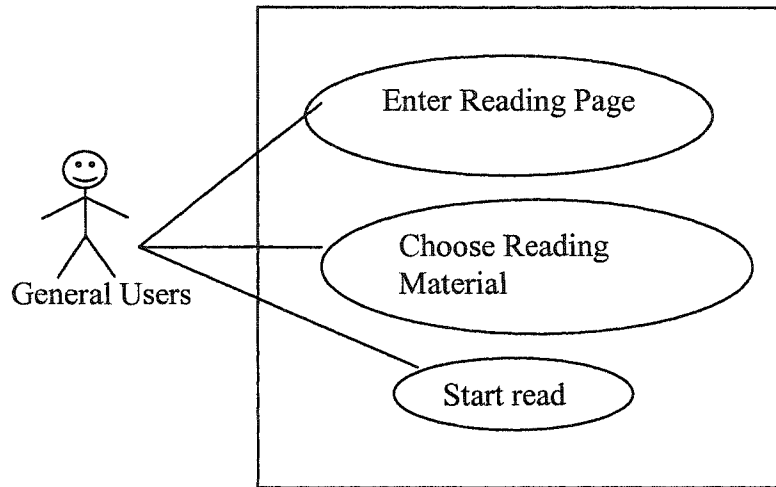


Figure 5: Use case of Reading

--- Use Case 5: Online Dictionary

This dictionary is based on the vocabulary database of the system. The users enter the main page of ELW and click the “Online Dictionary” link. A page will be displayed for users to enter the words. Users input an English word and click the “Submit” button. The system then searches the word in the database. If the system finds the word in the database, it displays the spelling, meaning and usage example. Otherwise, an error message will be display on the screen. In both cases, the user may use the imbedded Merriam Webster Online dictionary [13] to find the word.

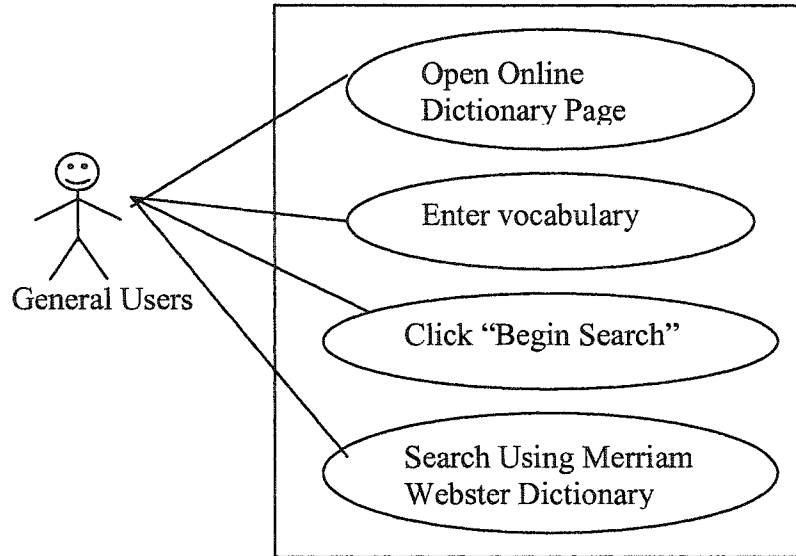


Figure 6: Use case of Online Dictionary

--- Use Case 6: User Comments

We welcome user comments for the website. To input comments, users click “Contact us” link. A page will be displayed for user to input their names, E-mail addresses and comments. Users click “Submit” button to submit their comments.

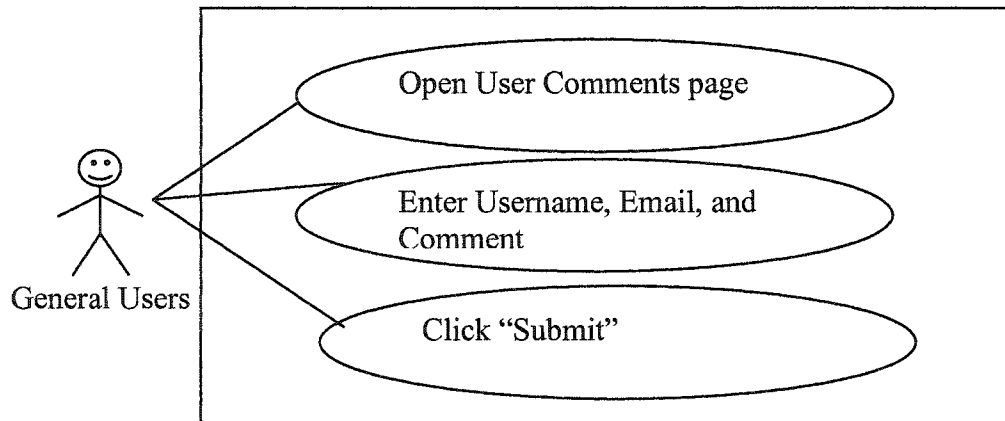


Figure 7: Use case of User Comments

--- Use Case 7: Update Vocabulary Database

To improve the ELW system, more and more vocabulary entries should be added into the databases. Thus, the most important job for system administrators is to update the vocabulary database. System administrators use Microsoft Access to open the vocabulary database file and then add, delete or edit the word entries.

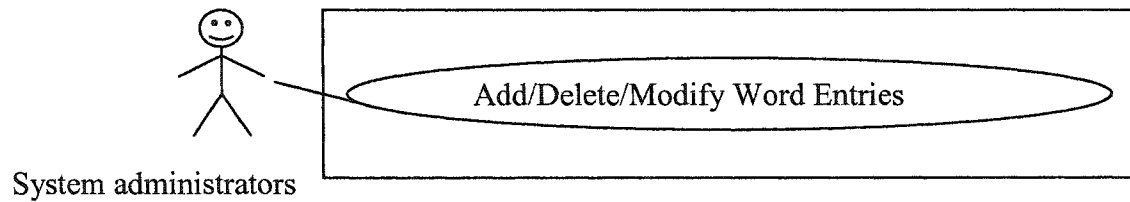


Figure 8: Use case of Update Vocabulary Database

--- Use Case 8: Update Reading pages

System administrators may change the reading materials time by time. They modify the reading html pages directly.

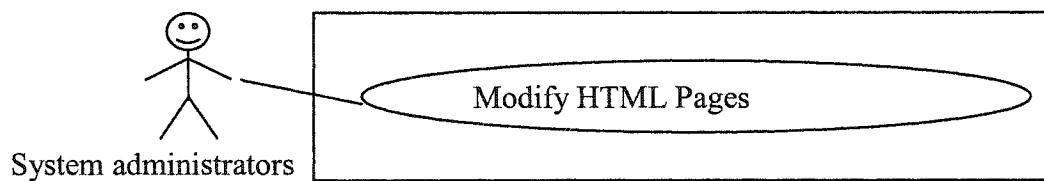


Figure 9: Use case of Update Reading pages

3. Non-functional Requirements

Unlike functional requirements, which are related directly to the functions or operations of the system, non-functional requirements are the requirements or restrictions imposed

by the client or the problems that do not relate directly to the functions or operations to be performed by the system. We have proposed the following non-functional requirements.

3.1 Product Requirements

Product requirements are the requirements that result from the need for the delivered product to behave in a particular way. For product requirements, ELW should meet the following prerequisites:

Server side:

- Microsoft Window 95/98/2000, or NT 4.0 as operating system
- Microsoft Access 2000 for DBMS
- Apache Tomcat as a server

Client side:

- Microsoft Window 95/98/2000/NT or UNIX as operating system
- Internet Explorer 5.0 or higher

3.1.1 User interface

Irrespective of the technical background, the user of ELW should be very comfortable with the system without any special training. User-Interface should be consistent and adherent to common-sense expectations.

3.1.2 Reliability

The system must be robust enough and handle multi-users gracefully and efficiently. Recovery from failures should be done easily and it is highly preferable that restoration of operation be accomplished automatically as regard to the most types of operational failures.

3.1.3 Efficiency

ELW is a web-based application. Everyone can enter into it online in a short time and even the remote users do not need longer response-times. But the speed will be affected by the controlling network-traffic over the Internet, which is an issue beyond the capabilities of any Internet user.

3.1.4 Safety

System safety is one of the most important things to be considered for ELW. To ensure the safety of the system, only database administrators have the privilege to directly access the system database.

3.2 Organizational & Process Requirements

These are the requirements that are a consequence of organizational policies and procedures.

3.2.1 Architecture

The system should comply with the client/server technology.

3.2.2 Software platforms

All web-users should access the English Learning Website using Internet Explorer (version 5.0 or higher) or Netscape (version 7.0 or higher) as their web-browser.

3.2.3 Development Tools

The developing software tools for developing this ELW are:

- Apache Tomcat

Tomcat is the Servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. [1]

- Java 2 Platform, Standard Edition (J2SE) 1.4.1.02SDK

“J2SE is the premier solution for rapidly developing and deploying mission-critical and enterprise applications. J2SE provides the essential compiler, tools, runtimes, and APIs for writing, deploying, and running applets and applications in the Java programming language.” [2]

- JAVA

Java is a programming language that’s well suited to designed software that works in conjunction with the Internet [5].

- Java Servlet / JavaServer Pages

“Servlets are the Java way to create web-enabled application. Each servlet extends the capability of a web server by providing additional custom functionality” [4].

A servlet receives an HTTP request, and returns an HTTP response.

JavaServer Pages perform the same task as Servlets, but use a different development paradigm [4]. While Servlets are created by writing Java classes, JavaServer Pages are created using syntax very similar to HTML. The JavaServer Pages are then converted into Servlets at run-time automatically.

- JavaScript

“JavaScript is a programmable API that allows cross-platform scripting of events, objects, and actions. It allows the page designer to access events such as startups, exits, and users' mouse clicks. JavaScript extends the programmatic capabilities of

most browsers to a wide range of authors, and is easy enough for anyone who can compose HTML.” [3]

- JDBC

“JDBC is Java’s solution to providing programmatic access for retrieving and storing relational data from within any Java program” [4]. ELW uses JDBC connect the Web Server with Database.

- Microsoft access

Microsoft access can be used to store the whole information used in the ELW operational database. It can receive the SQL queries and give the query results out.

- FrontPage

FrontPage is a powerful web design and development tool. With FrontPage, you can design the website you want and generate code more efficiently.

3.2.4 Software Maintenance

After the software is released to the clients, the main task becomes supervision of the software configuration as well as the physical modification of the software. The developers should be equipped with internal modules that keep cumulative data about the system performance and automatically tune themselves to respond to the changing needs of the system’s actor.

4. System Design

4.1 Design Rationale

The English Learning Website is designed to provide the following web-based services including: testing users' English level, vocabulary Lessons studying, lesson's vocabulary testing, reading Article/Novel/News, online Dictionary, and send comments. Therefore ELW should consist of the following subsystems:

- **User Interface:** The UI allows the general users to interact with the ELW system through the web using an Internet browser. System accepts user inputs in mouse clicks or keyboard inputs.
- **Tomcat Server:** To receive the information from Event Handle and connect the user interface with JDBC.
- **JDBC:** To connect the Web Server with Database because most web servers cannot access database directly.
- **Database:** To store information of the ELW system. To receive the SQL queries and to give the query results.

4.1.1 System Architecture

Figure 10 shows the system architecture diagram of ELW that encompasses four subsystems: User Interface Subsystem, Web Server Subsystem, JDBC Subsystem, and Database Subsystem.

General users interact with the system through the user interface. User's requests are received by web server through HTTP protocol. Web server connects with JDBC. JDBC

transferred the requests into SQL statements. The SQL statements are executed in the database. The database retrieve data back to the web server through JDBC. The web server gives an HTTP response to the general user through the User Interface.

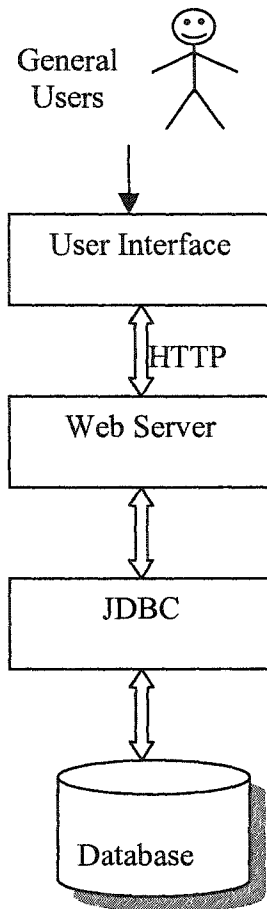


Figure 10: System Architecture Diagram

4.2 Database Schema and Description

4.2.1 Database Schema

The following is the definition of database schemes. The **Bold** words are the primary key.

The *Italic* and underlined words are primary foreign keys:

Index (**Word**, *Start ID*, *End ID*);

Dictionary (ID, Vocabulary, Level, Level_ID)

*_Voca (Level_ID, Vocabulary, pronounce, Meaning, Examples)

TestQuestion(Number, Question, Answer1, Answer2, Answer3, Answer4,
CorrectResult, Level, CourseID)

4.2.2 Database Table definition

Index Table

Field Name	JDBC Data Type	Description
Word	Text	From a to z, primary key
Start_ID	Number	
End_ID	Number	

Dictionary Table

Field Name	JDBC Data Type	Description
ID	Number	
Vocabulary	Text	Primary key
Level	Number	
Level_ID	Number	

Begi_Voca Table

Field Name	JDBC Data Type	Description
Level_ID	AutoNumber	Auto increment, Primary key

Vocabulary	Text	
Pronounce	Text	
Meaning	Memo	
Examples	Memo	

Inte_Voca Table

Field Name	JDBC Data Type	Description
Level_ID	AutoNumber	Auto increment, Primary key
Vocabulary	Text	
Pronounce	Text	
Meaning	Memo	
Examples	Memo	

Adva_Voca Table

Field Name	JDBC Data Type	Description
Level_ID	AutoNumber	Auto increment, Primary key
Vocabulary	Text	
Pronounce	Text	
Meaning	Memo	
Examples	Memo	

TestQuestion

Field Name	JDBC Data Type	Description
Number	AutoNumber	Auto increment, Primary key
Question	Memo	
Answer1	Text	
Answer2	Text	
Answer3	Text	
Answer4	Text	
CorrectResult	Text	
Level	Number	
CourseID	Number	

4.2.3 E_R diagram

Following is the E_R diagram of the system database that is used in this ELW project:

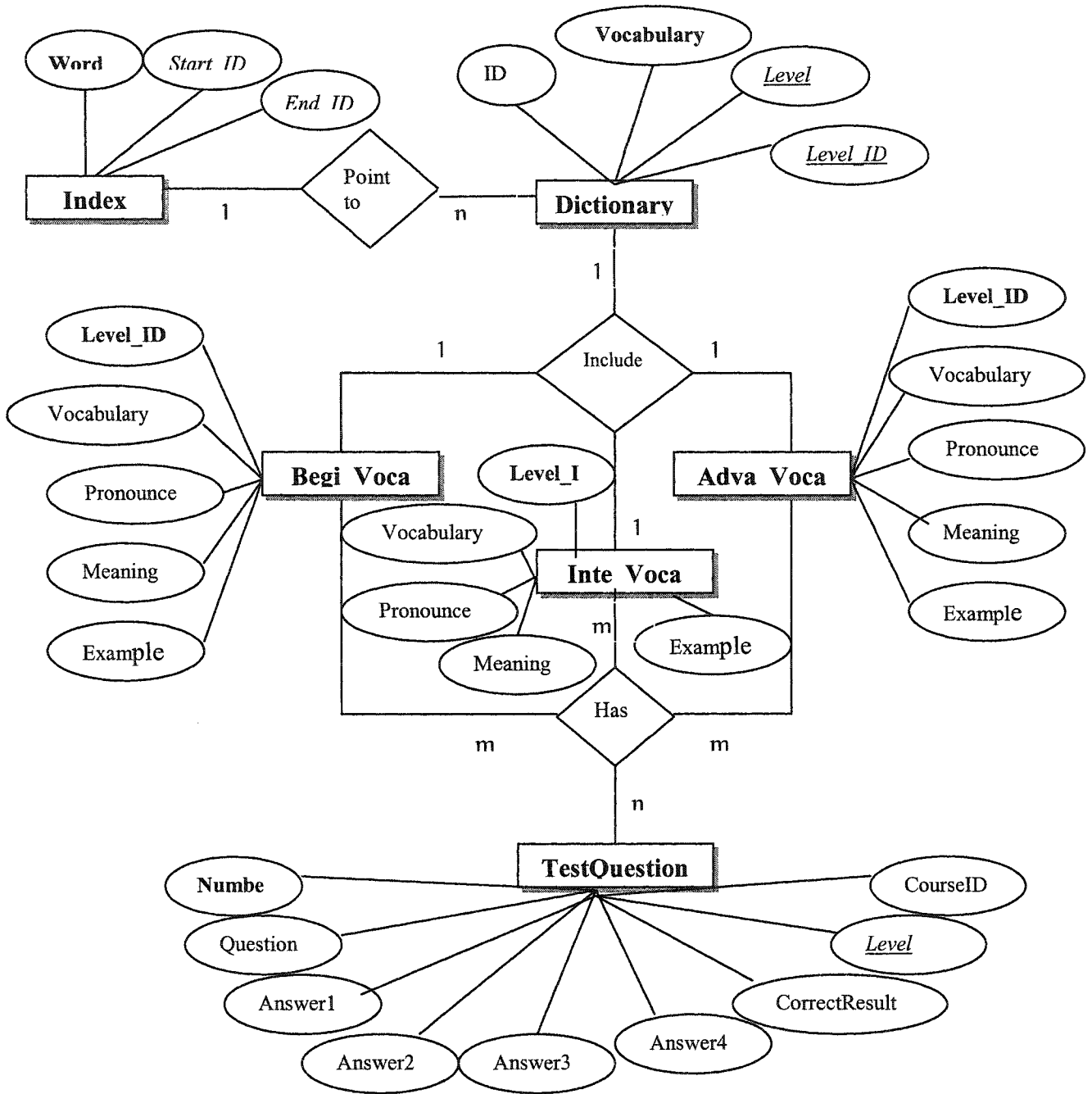


Figure 11: E-R Diagram

4.3 Function Design

According to the software requirements and analysis, ELW has five basic functions for general users. They are “Vocabulary Level Testing”, “Vocabulary Studying”, “Reading”, “Online Dictionary”, and “User Comments”. Following is the detail introduction of each function design.

4.3.1 Vocabulary Level Testing Function

4.3.1.1 Class Diagram

Vocabulary Level Testing consists of five classes: HttpServlet Class, TestingLevel Class, TestingLevelResult Class, JDBCconnection Class, and Question Class. The class diagram is shown in the Figure 12.

- TestingLevel Class and TestingLevelResult Class are extended from HttpServlet Class, which provides an abstract class to be subclassed to create an HTTP servlet suitable for a Web site [7].
- TestingLevel Class contains two other classes: JDBCconnection Class and Question Class. It's connected to the database through JDBCconnection Class variable. It uses the *PickNumber()* function to get several random numbers and retrieves data (Questions) by these random numbers from TestQuestion Table in database. Questions are then stored into Question class variable.
- TestingLevelResult class gets the user's answer from interface, and gets Question variable from TestingLevel class, respectively. The *getcorrectresult()* function gives the correct rate of user's test. The *GetUserLevel()* function gives the user's current vocabulary level.

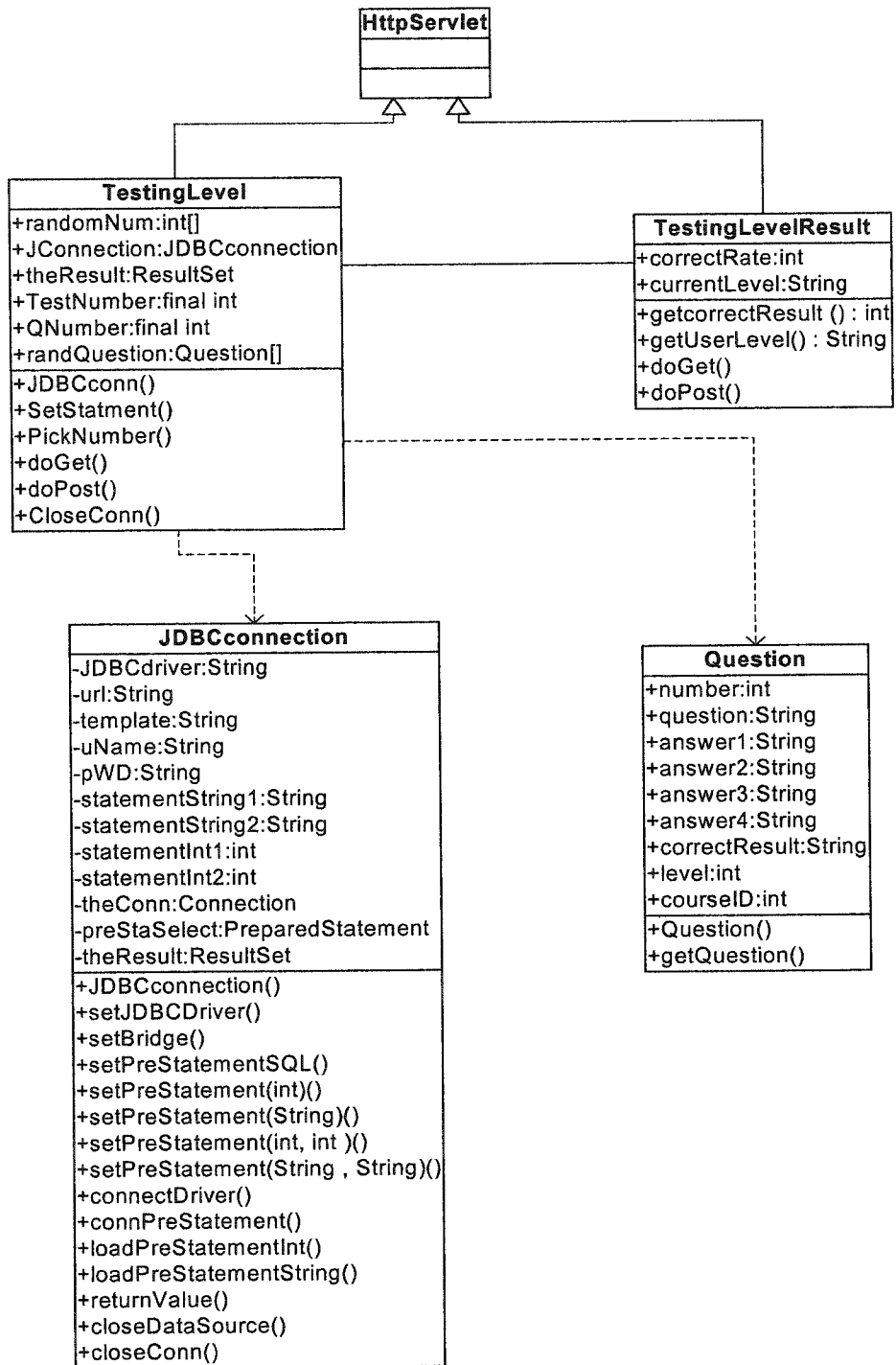


Figure 12: Vocabulary Level Testing Class Diagram

4.3.1.2 Sequence Diagram

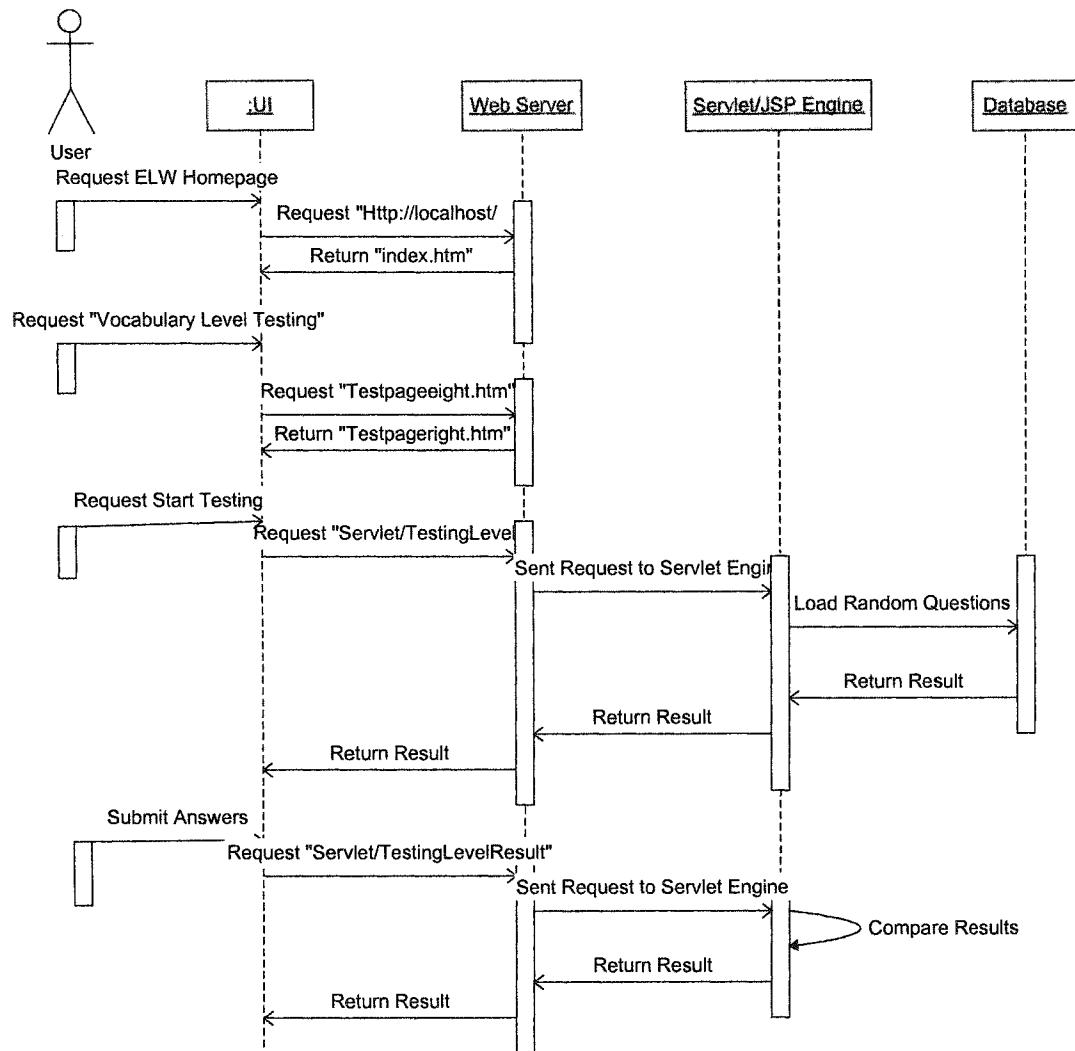


Figure 13: Vocabulary Level Testing Sequence Diagram

Figure 13 shows the sequence diagram of Vocabulary Level Testing, which is designed according to the Vocabulary Level Testing scenario. It begins when a user requests ELW

homepage through user interface. Web server receives the information and returns the “index.htm” page. User requests Vocabulary level Testing page, then the web server returns the “Testrightpage.htm” page. The user sends a request to start the vocabulary level testing. The web server transfers the request to Servlet/JSP engine. Servlet/JSP engine opens database and selects random questions. Then, it returns the questions. At last, the user gets questions and submits the answers after having finished the test. Web server transfers the answers to Servlet/JSP engine. The engine compares the answers with correct answers and returns the testing result.

4.3.2 Vocabulary Studying Function

4.3.2.1 Class Diagram

Vocabulary studying function consists of five classes: HttpServlet Class, VocabularyStudy1 Class, VocabularyStudy2 Class, VocabularyStudy3 Class, and JDBCconnection Class. The class diagram is shown Figure 14.

- The VocabularyStudy1 Class is a class for beginner vocabulary level.
- The VocabularyStudy2 Class is a class for intermediated vocabulary level.
- The VocabularyStudy3 Class is a class for advanced vocabulary level.

According to the diagram, VocabularyStudy1 Class, VocabularyStudy2 Class, and VocabularyStudy3 Class are extended from HttpServlet Class. They all use JDBCconnection Class to connect to the database.

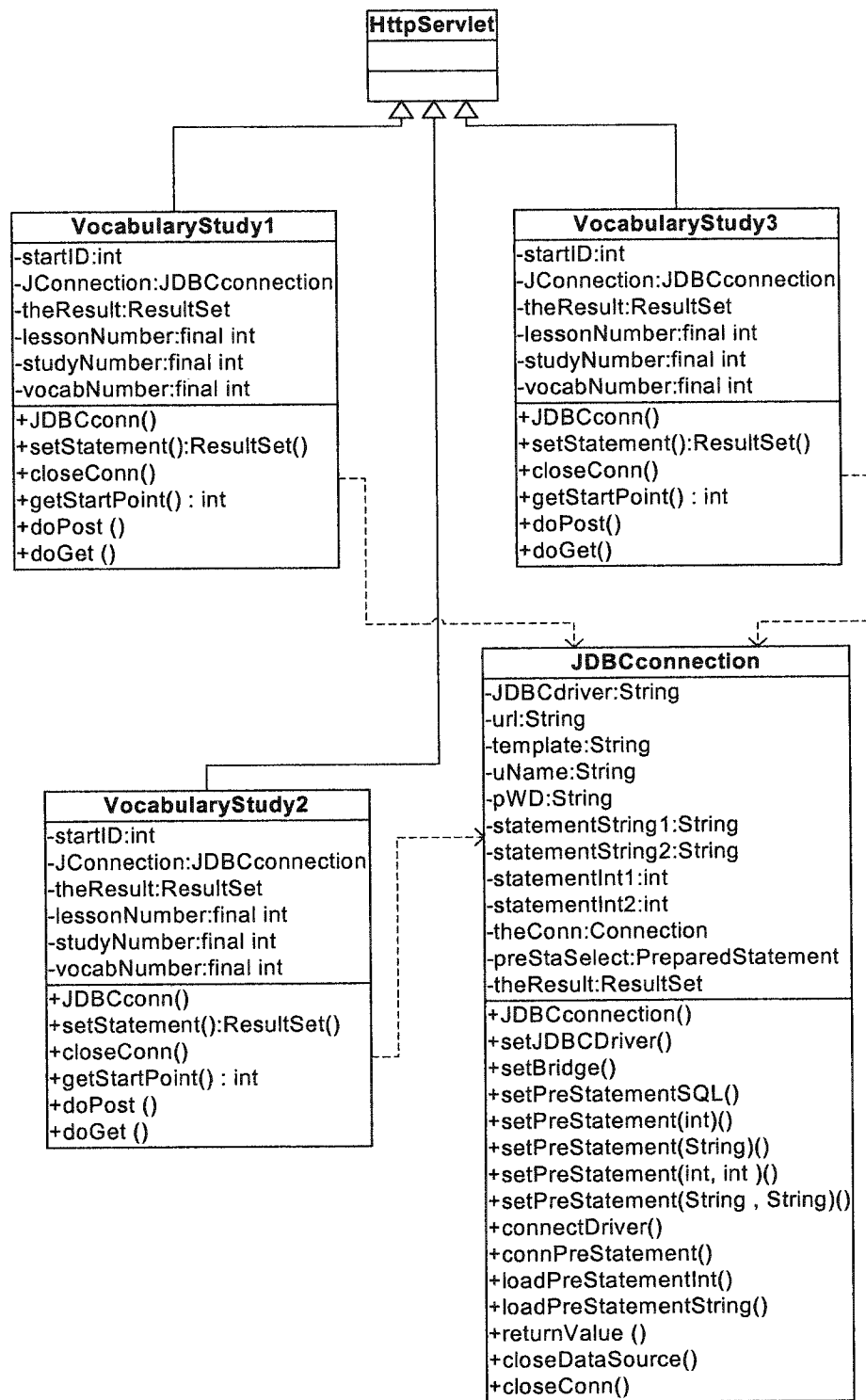


Figure 14: Vocabulary Studying Class Diagram

4.3.2.2 Sequence Diagram

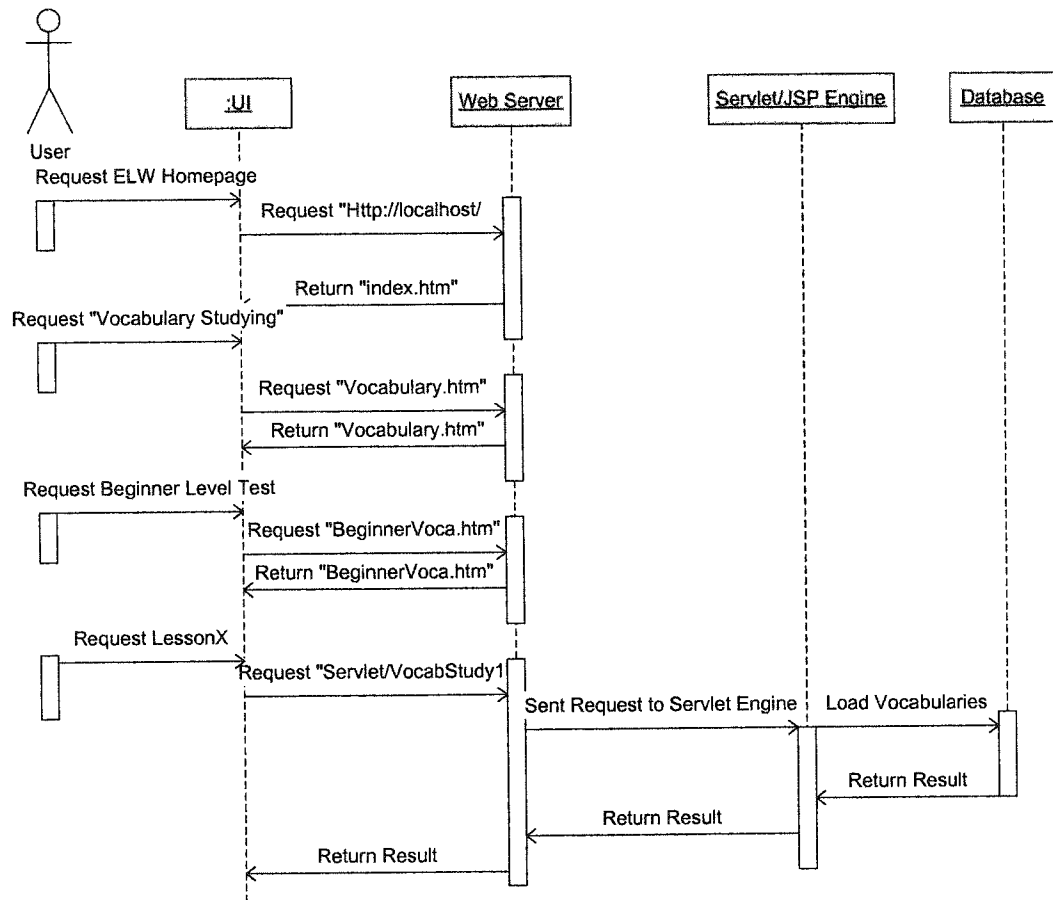


Figure 15: Vocabulary Studying Sequence Diagram

Figure 15 shows the sequence diagram of Vocabulary studying, which is designed according to the Vocabulary studying scenario. It begins when the user requests the ELW homepage through the interface. The web server receives the request and returns the homepage. The user then requests “Vocabulary” page. The web server returns

“Vocabulary.htm” page. The user selects which level they want to enter, for example, beginner level. The web server returns “BeginnerVoca.htm”. The user selects a lesson. The web server transfers the request to Servlet/JSP engine. The Servlet/JSP engine opens database, gets the words, and then, returns all the related data. At last, the user gets the requested vocabulary.

4.3.3 Lesson testing Function

4.3.3.1 Class Diagram

Lesson testing function consists of six classes: HttpServlet Class, VocabTest Class, VocabTestResult Class, JDBCconnection Class, Question Class, and AnswerArray Class. The class diagram is shown in Figure 16.

VocabTest Class and VocabTestResult Class are extends from HttpServlet Class.

- VocabTest Class uses two other classes: JDBCconnection Class and Question Class. It connects to database through JDBCconnection Class variable. It creates a new file for storing questions. Then, once the user submits the answers, it transfers it to VocabTestResult Class.
- VocabTestResult Class contains AnswerArray Class. This class gets the user’s answer and compares it with the file with Questions.

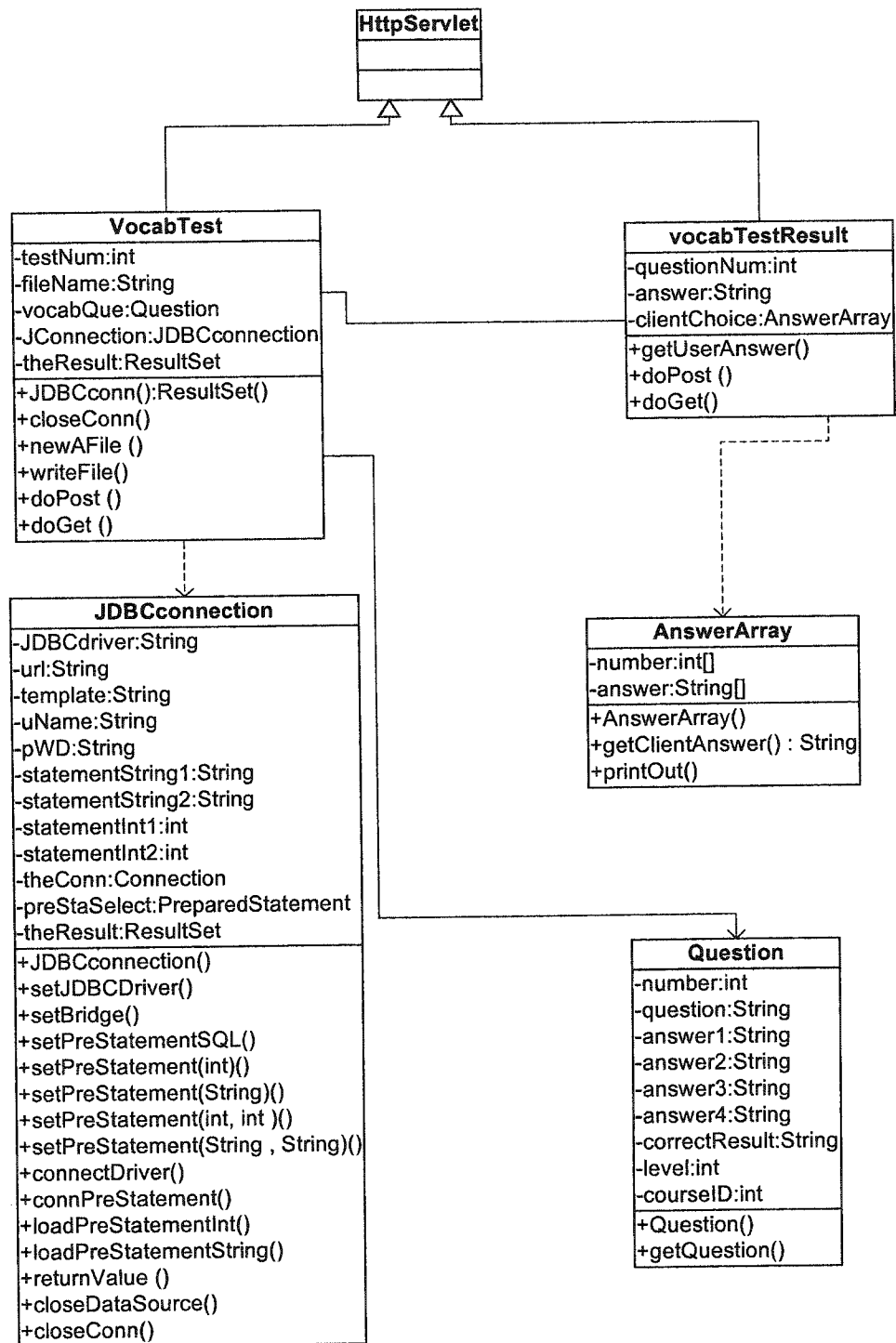


Figure 16: Lesson Testing Class Diagram

4.3.3.2 Sequence Diagram

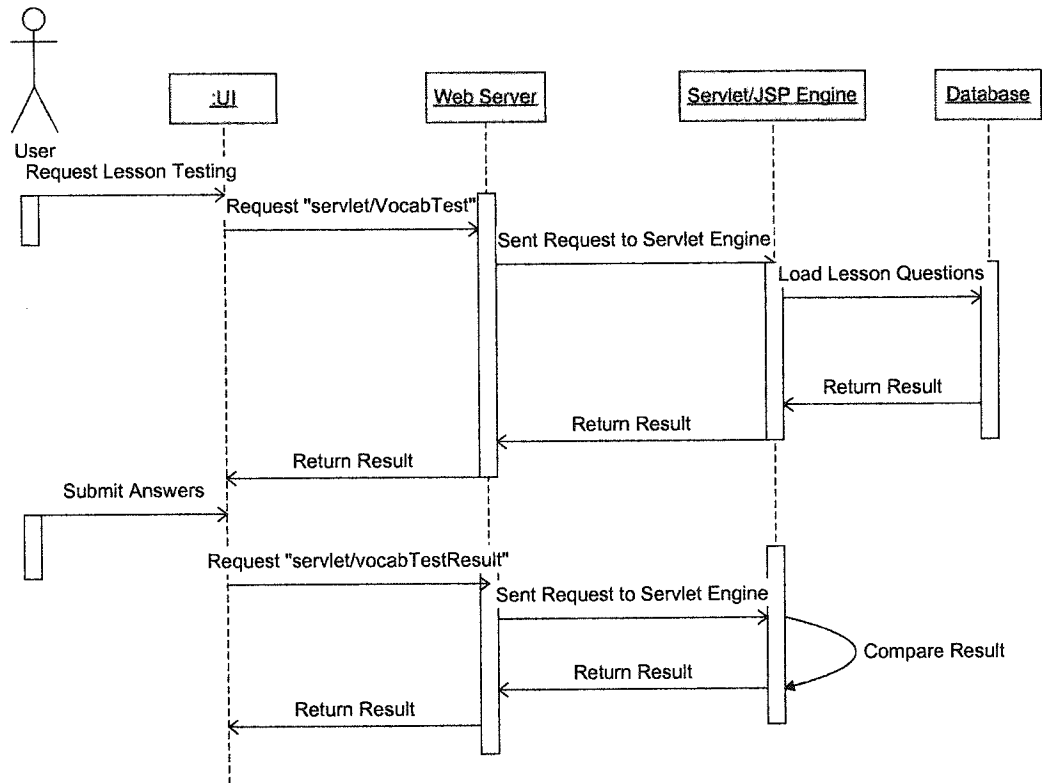


Figure 17: Lesson Testing Sequence Diagram

Figure 17 is the sequence diagram of Lesson testing, which is designed according to the Lesson Testing scenario. It begins when a user has finished a lesson studying and decides to do a test for the lesson. The user starts the test by requesting the VocabTest class. The web server transfers the request to Servlet/JSP engine. The Servlet/JSP engine opens the

database and selects questions. The user receives the questions, finishes them and submits the answers. The web Server receives the answer and transfers the request to Servlet/JSP engine. The system checks the answers and returns the questions and results to the user interface.

4.3.4 Reading Function

The reading part should be suitable for different level users to improve their reading skill.

For this reason, I divide it into three kinds of articles:

- **Stories:** These stories include basic vocabulary and it is suited for the users who are in the beginner level in ELW.
- **TOEFL:** These articles are selected from the each year's official TOEFL comprehensions. It is suited for the users who are in the intermediated level in ELW.
- **BBC News:** Reading and understand English news are the big challenge for anyone whose mother language is not English. These articles are suited for the users whose English level is in advanced in ELW.

4.3.5 Online Dictionary Function

4.3.5.1 Online Dictionary Database Structure Design

The online dictionary should retrieve data from the database fast and efficiently. To achieve this goal, I designed a two-level Index database access method to search and retrieve data from database. Indexing is a widely used concept in some retrieval system such as dictionary or card catalog systems for libraries [6].

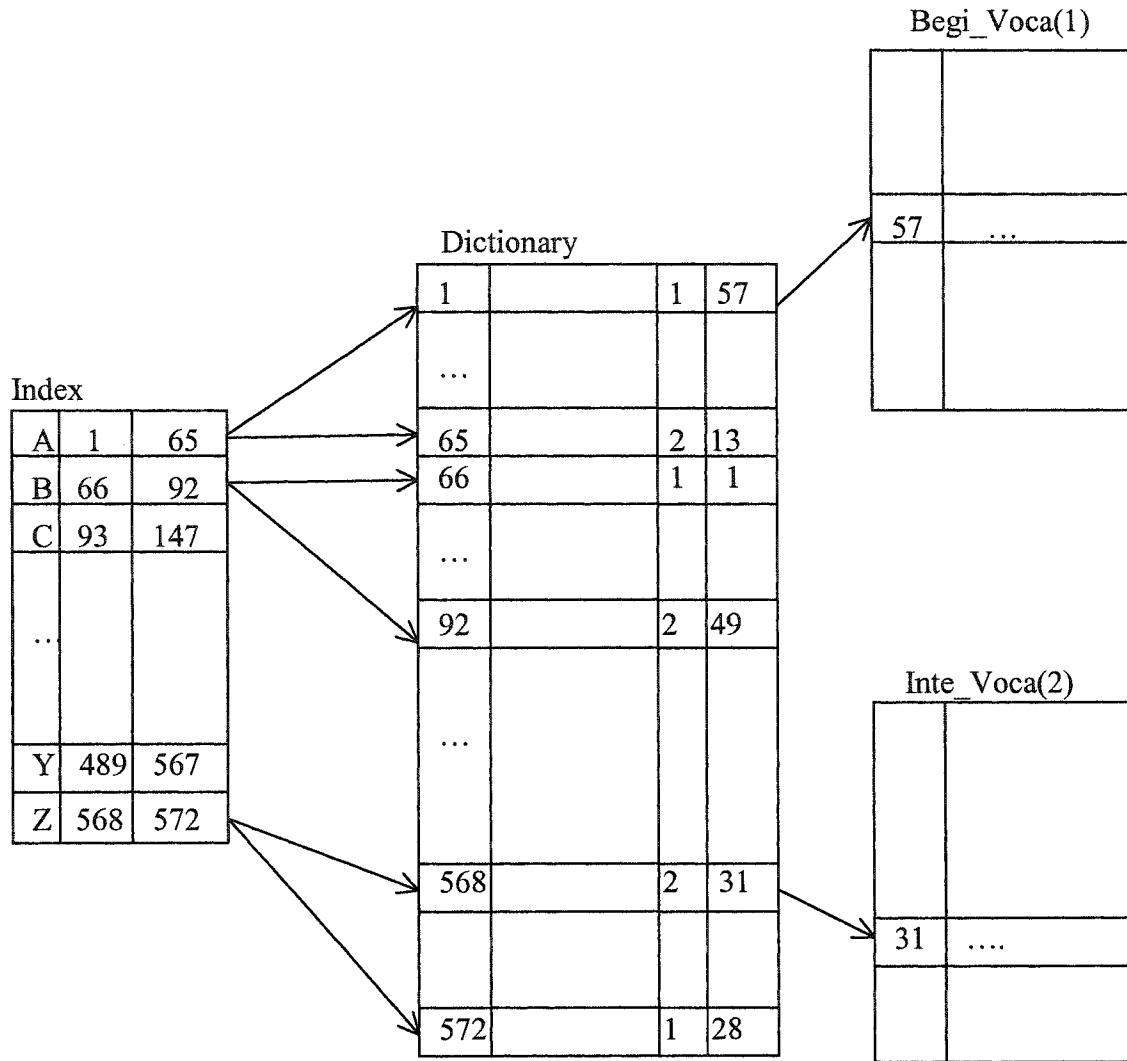


Figure 18: Online Dictionary Database Structure

Figure 18 shows how to search a word in the database. The two-level index searching algorithm is shown below:

Algorithm: ONLINEDICTIONARY

get request parameter (P) from user

trim the blanks before P

if (P.length == 0)

```

    Dispartcher to dictionaryW.jsp

else

    get first letter (F)
    if (F is not a alphabet)
        Dispartcher to dictionaryW.jsp

else

    search in index table
    get startpoint (S) and endpoint (E)
    if (S == 0)
        Dispartcher to dictionaryW.jsp
    else
        search (S,E) in Dictionary table
        if (!P)
            Dispartcher to dictionaryW.jsp
        else
            get level (L) and levelID(L1)
            search (L1) in level (L)
            Dispartcher to dictionaryR.jsp

```

After the system has retrieved the word that user wants to search, it extracts the first letter of the word and compares with the first column of Index Table. If it finds the same letter, it gets the start point number and end point number in the Dictionary Table. The words between the start point and the end point has the same first letter as the word user is

searching. The system then searches for the vocabulary from the start point number. Once it finds the vocabulary, the system gets the Level Number and LevelID Number from the Dictionary Table. Level has three numbers, they are 1: beginner level, 2: intermediate level, and 3: advanced level. LevelID gives the exactly position of the word.

Using this two-level indexed searching algorithm, the system finds a vocabulary in the database quickly. The algorithm will show great advantage when the vocabulary database is large, or when many users are visiting the website at the same time.

4.3.5.2 Class Diagram

The online dictionary function consists of three classes: HttpServlet Class, Dictionary Class, and JDBCconnection Class. The class diagram is shown in Figure 19.

Dictionary Class is extended from HttpServlet Class. And, it connects to database through JDBCconnection Class.

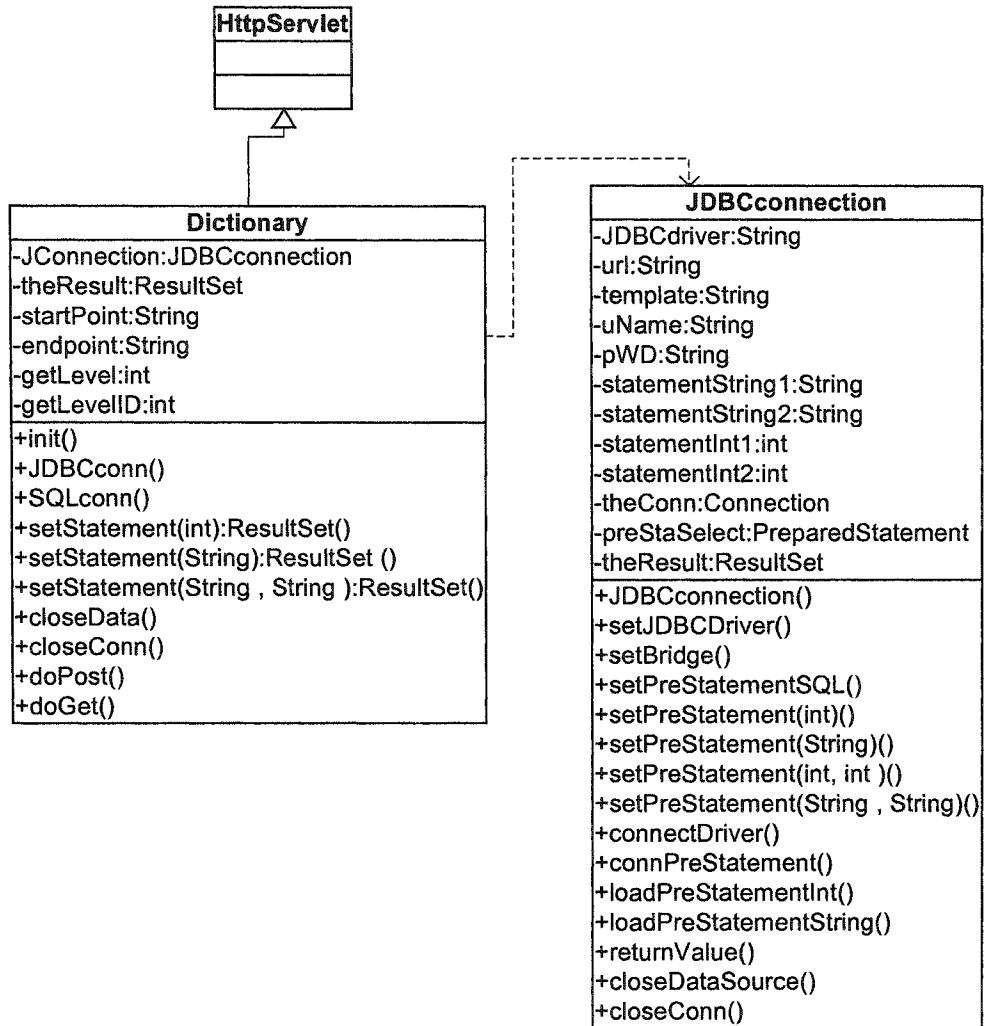


Figure 19: Online Dictionary Class Diagram

4.3.5.3 Sequence Diagram

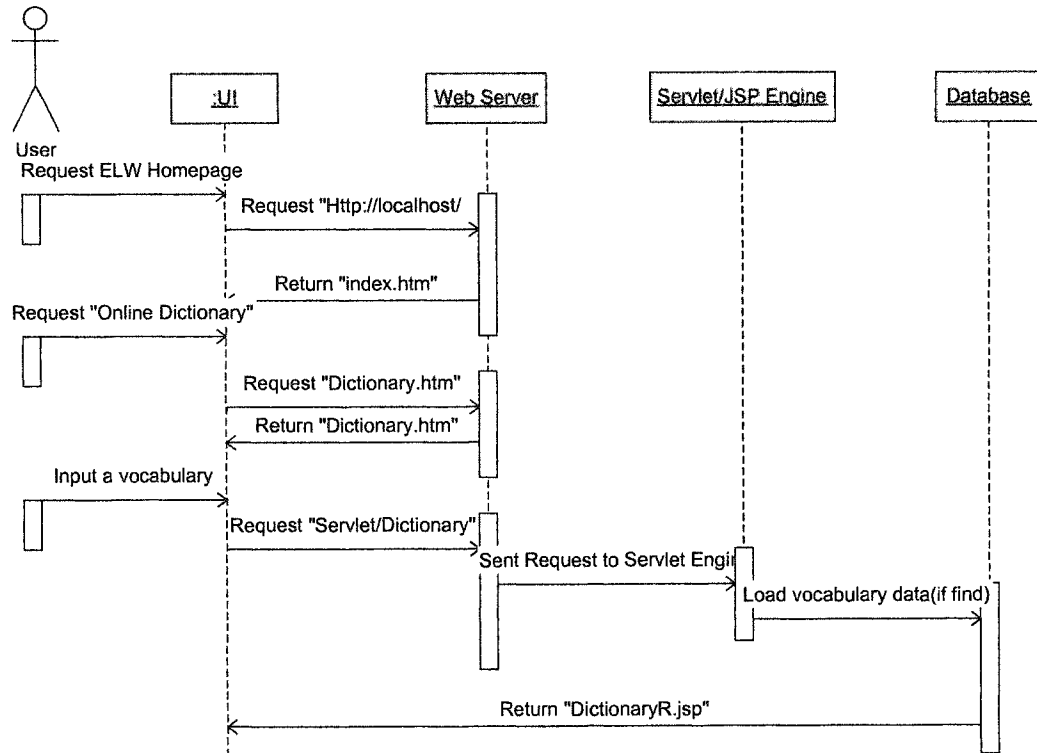


Figure 20: Online Dictionary Sequence Diagram

Figure 20 shows the sequence diagram of Online Dictionary, which is designed according to the online dictionary scenario. It begins when a user requests the ELW homepage. Web server receives the request and returns the homepage. The user then requests “Online Dictionary” page. Web server returns “Dictionary.htm” page. User inputs the word they want to search. Web server transfers the request to Servlet/JSP engine.

Servlet/JSP engine opens database and looks for the word. If the word is found in the database, it returns all of the data related to the word to user interface by “DictionaryR.jsp”. Otherwise, it enters error handling.

4.3.5.4 Error Handling

In some cases, the system cannot find the word that user wants to search. The situations include:

- Empty input
- Blanks before the input word
- Word begins with a non-alphabet letter
- Database does not contain the word

In these cases, the system finds the error class and passes it to “dictionaryW.jsp”. The JSP file performs different operation according to the error classes. The JSP file also provides connection to Merriam-Webster Online dictionary so that user may find the result they want.

4.3.6 User comments Function

4.3.6.1 Class Diagram

User comments function is extended from HttpServlet Class. The class diagram is shown in Figure 21.

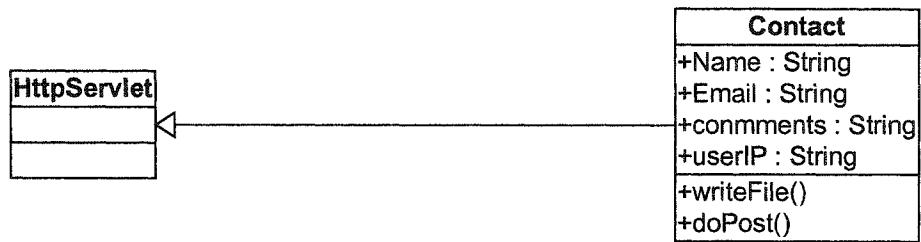


Figure 21: User Comments Class Diagram

5. Implementation

5.1 System Environment

“Apache Tomcat 4 is a standalone web server used for development that supports servlet and JSP. Tomcat is produced by the Apache Software Foundation’s Jakarta project, and is free to download from <http://jakarta.apache.org>. It provides a high-quality, robust web container with excellent performance” [4]. Tomcat and J2SDK are installed in the Windows XP system to support ELW development.

To run Tomcat, it is simple to execute the appropriate startup file in `%CATALINA_HOME%\bin\startup.bat`. To shutdown Tomcat, execute `%CATALINA_HOME%\bin\shutdown.bat`. `%CATALINA_HOME%` is a system environment variable that points to the directory when the Tomcat is installed.

5.2 Application Architecture

According to the System Architecture, we can get the application architecture of ELW: Users use web browser to send HTTP requests to the server and get HTTP responses from the server. Servlets and JSPs are a subset of the overall collection of Java server-

side APIs called the J2SE application server. The J2SE technologies are designed to provide the essential compiler, tools, runtimes, and APIs for writing, deploying, and running applets and applications in the Java programming language [2]. Servlets and JSP together form the presentation layer of J2SE web applications is used to form the logic and data layers [4].

Figure 22 shows the application architecture.

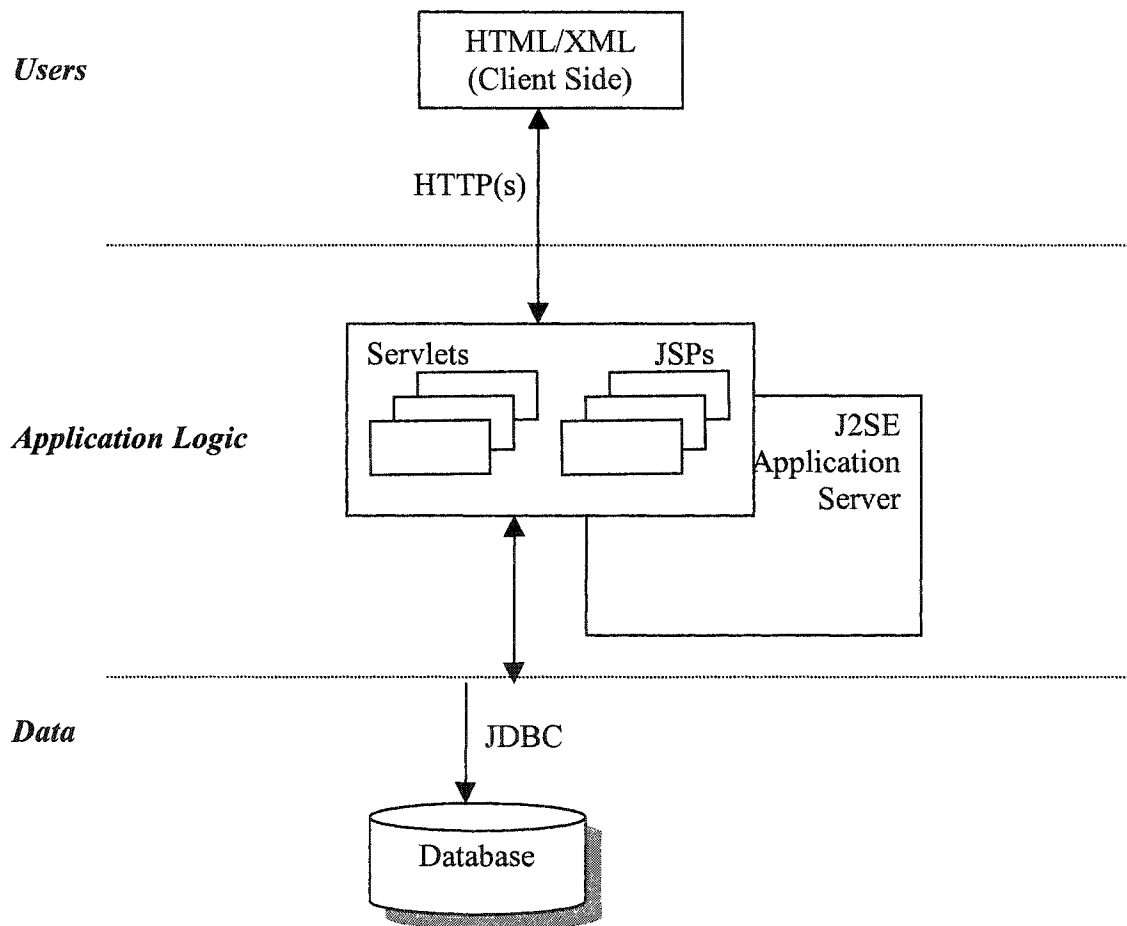


Figure 22: Application Architecture Diagram

5.3 Data Access and JDBC

Data Access in JDBC is accomplished via a driver, which implements the `java.sql.Driver` interface [4]. Generally speaking, the driver can be thought as a layer of interposition between Java application and the underlying database, accepting SQL commands from the application, forwarding them to a database, obtaining the results and returning them to the calling application. The exact details of how communication between application and database is conducted depend on the type of the driver.

JDBC driver implementations normally have four kinds of categories. They are

- JDBC-ODBC Bridge plus ODBC Driver
- Native-API, Partly-Java Driver
- JDBC-Net Pure Java Driver
- Native Protocol Java Driver

ELW uses JDBC-ODBC Bridge plus ODBC Driver to implement data access.

JDBC-ODBC Bridge plus ODBC Driver is JDBC over ODBC. ODBC is a Microsoft-touted technology to provide programmatic access to relational databases [4]. ODBC has good support from all of the leading database product companies. JDBC build on all of the concepts and features of ODBC, but different in that it provides a higher-level abstraction of interacting with a database in Java.

To using JDBC, the following steps should be implemented:

- Load the JDBC Driver
- Obtain a connection

- Obtain a statement or prepared statement to execute a query or update
- Close the connection

ELW uses JDBCconnection class to implement all of the functions. The detail of how to implement the four steps is:

1. Using *Class.forName()* to load the JDBC driver class into the application's Java virtual machine (JVM), which makes the driver available for opening a connection. If the class-loader fails to load the specified driver, a *ClassNotFoundException* is thrown.

In JDBCconnection class, I use the *setJDBCdriver()* method to set driver name and use *connectDriver()* to load the driver.

```
public void setJDBCdriver(String driverName) {
    this.JDBCdriver = driverName;
}
public void connectDriver() {
    try {
        Class.forName(this.JDBCdriver);
    } catch(java.lang.ClassNotFoundException e) {
        System.err.print("ClassNotFoundException: ");
        System.err.println(e.getMessage());
    }
}
```


2. When the driver is loaded into memory, it is registered as an available driver with the `java.sql.Drivermananger` class, which is a basic service for managing a set of JDBC drivers [4]. The next step is to ask the `DriverManager` for a connection to a database using `DriverManager.getConnection()`.

In the `JDBCconnection` class, I use the `setBridge()` method to set `bridgeURL`, `username` and `password`. Then, I use `connectDSource()` to connect database.

```
public void setBridge(String bridgeUrl, String userName, String passWord) {  
    this.url = bridgeUrl;  
    this.uName = userName;  
    this.pWD = passWord;  
}  
  
public void connectDSource() throws Exception {  
    this.theConn = DriverManager.getConnection(this.url, this.uName, this.pWD);  
}
```

3. After establishing the connection, one or several `Statement` is created to operate the underlying relation data. Since ELW `Statements` need to be executed multiple times, it uses prepared statement instead of statement. A `PreparedStatement` differs from a regular statement in that it specifies a parameterized, fill-in-the-blanks SQL template [4]. It is sent to the database before actually being used and the question marks in the `PreparedStatement` are used to placeholders for data supplied as run-time.

In the JDBCconnection class, I use the *setPreStatementSQL()* method to set SQL comments and use *connPreStatement()* to create a PreparedStatement object. Then, I use *loadPreStatementInt()* or *loadPreStatementString()* to set PreparedStatement question marks Value and execute query.

```
public void setPreStatementSQL(String temp) {  
        this.template = temp;  
}  
public void connPreStatement() throws Exception{  
        this.preStaSelect = theConn.prepareStatement(this.template);  
}  
public void loadPreStatementInt(int num) throws Exception{  
        if (num == 1) {  
                (this.preStaSelect).setInt(1,this.statementInt1);  
        }  
        else if(num == 2) {  
                this.preStaSelect.setInt(1,this.statementInt1);  
                this.preStaSelect.setInt(2,this.statementInt2);  
        }  
        this.theResult = (this.preStaSelect).executeQuery();  
}  
public void loadPreStatementString(int num) throws Exception{  
        if (num == 1) {  
                (this.preStaSelect).setString(1,this.statementString1);}  
        else if(num == 2) {
```

```

        this.preStaSelect.setString(1,this.statementString1);
        this.preStaSelect.setString(2,statementString2);
    }
    this.theResult = (this.preStaSelect).executeQuery();
}

```

4. Finally, it is important to close the database connection. It includes everything returned from the Connection, as well as Connection itself.

In JDBCconnection class, I use *closeDataSource()* and *closeConn()* method to close a the database connection.

```

    public void closeDataSource() throws Exception {
        if ( theResult!= null) theResult.close();
        if (preStaSelect != null) preStaSelect.close();
    }
    public void closeConn() throws Exception {
        if (theConn != null) theConn.close();
    }
}

```

5.4 HttpServlet Class

HttpServlet is a convenient abstract class for creating Servlets [4][7]. It extends the GenericServlet base class and provides a framework for handling the HTTP protocol. Because it is an abstract class, servlet writers must subclass it and override at least one method. The methods normally overridden are either the doGet or doPost methods (or

both). The server calls service. Service in its turn calls doGet, doPost, whatever, depending on the client's request.

ELW only overrides doGet method and uses doPost method to call doGet method because most situations will require the Get and Post HTTP methods to have the same response.

```
public void doPost(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {  
doGet(request, response);  
}
```

The Servlet defines that doGet and doPost methods take two parameters named request and response, and throws two exceptions as the doPost() example above.

5.5 The Session Servlets

There are a number of problems that arise from the fact that HTTP is a "stateless" protocol [8]. In ELW, these problems become more and more serious. When the users finish their Vocabulary Level Testing, click the "Submit" button to move to the result page, how does the server remember the correct answer for each question that has been gotten in the previous page?

The easiest way we can image is to use cookies. But, cookies can only be used if the browser supports them. The servlets session management provides an outstanding technical solution. "It goes beyond just identifying the connection" [4]. Each session is an object in the HttpSession class. The class gives the capability of associating arbitrary data

to it. The mechanism allows a number of name-value pairs to be saved into the session and it may be available in the future requests.

A session should be implemented with following steps:

1. Build an HttpSession object associated with the current request.

```
HttpSession session = request.getSession(true);
```

We pass “true” to getSession so that it can automatically create a new session.

2. Save data into the session

```
session.setAttribute("name", uName);
```

The first parameter in *setAttribute()* is the string value name that will be stored in session and it is must be unique for session itself. The second parameter is the value that is any Java object.

3. Get data out of the session

```
session.getAttribute("name");
```

getAttribute() method returns the object names that are stored in the session. This can be used in any other pages---Servlet page, JSP page, or even HTML page.

5.6 Functions implementation

According to the system design, six functions are to be implemented. They are Vocabulary Level Testing, Vocabulary Studying, Lesson Testing, Reading, Online Dictionary and User Comments. Following is the detail description.

5.6.1 Vocabulary Level Testing

In Vocabulary Level Testing, 25 questions are retrieved from the database. These 25 questions are retrieved randomly from the database. Each question is stored in the class

named Question first, then, it is stored in Java Session (refer to section 5.5) for future testing result page use.

Vocabulary Level Testing Uses *pickNumbers()* Function selects “count” numbers at random between “low” and “high”, in which “low” and “high” means the low bound and high bound of the range that the numbers are selected from, “count” means the number of the numbers to be selected. The random numbers are stored into an integer array as the return value.

```
public static int [] pickNumbers(int low, int high, int count) {  
    Random random = new Random();  
    int range = high - low + 1;  
    int [] num = new int[range];  
    for (int i=0, start=low; i < range; i++, start++)  
        num[i] = start;  
    for (int i=0; i < count; i++) {  
        int which = i+random.nextInt(range-i);  
        int tmp = num[i];  
        num[i] = num[which];  
        num[which] = tmp;  
    }  
    int [] num1 = new int[count];  
    System.arraycopy(num, 0, num1, 0, count);  
    return num1;  
}
```

5.6.2 Vocabulary Studying

Vocabulary studying is divided into three levels classes: Beginner level, Intermediate level, and Advanced level. All the vocabulary are taken from a series of books written by Mr. Yi Liu, in which Beginner Vocabulary get from book Vocabulary 5000[10], intermediated vocabulary get from book Vocabulary 10000[11], and Advanced vocabulary get from book Vocabulary 22000 [12].

For each level, there is a *getStartPoint()* function to get the start point of each lesson.

```
public int getStartPoint(String LessonNum) {  
  
    int returnNum;  
  
    returnNum = Integer.parseInt(LessonNum.substring(6));  
  
    returnNum = (returnNum-1)*20+1;  
  
    System.out.println(returnNum);  
  
    return returnNum;    //return a start point of each Lesson  
  
    }
```

There are 20 words in each lesson. Each vocabulary word has its pronunciation, meaning, and example for learning. After each lesson, the user can go to the next lesson or can do a lesson testing by click corresponding button.

5.6.3 Lesson testing

In each lesson, there is a test for the vocabulary the user just learned from this lesson. Each test has 5 questions corresponding to the vocabulary in this lesson. The system stores questions and correct answers in a file and compare the correct answer with the user's answers. Then, the system outputs them.

```

//create new file to store question and answer
public void newAFile (String theFileName) throws IOException{
    FileOutputStream file = new FileOutputStream(fileName);
file.close();
}

// write into file
public void writeFile(Question vQue) throws IOException {
    DataOutputStream writeData = new DataOutputStream(
        new BufferedOutputStream(
            new FileOutputStream(fileName, true)));
writeData.writeInt(vQue.number);
writeData.writeChar('\t');
writeData.writeChars(vQue.correctResult);
writeData.writeChar('\n');
writeData.writeChars(vQue.question);
writeData.writeChar('\n');
writeData.writeChars(vQue.answer1);
writeData.writeChar('\n');
writeData.writeChars(vQue.answer2);
writeData.writeChar('\n');
writeData.writeChars(vQue.answer3);
writeData.writeChar('\n');
writeData.writeChars(vQue.answer4);
writeData.writeChar('\n');
writeData.close();
}

```


5.6.4 Reading

The reading part is implemented using a static method. Stories and TOEFL articles are made into HTML pages, which is accessible using hyperlinks in Reading homepage. BBC News articles are from the BBS website. System administrators may change the reading materials according to necessity.

5.6.5 Online Dictionary

ELW online dictionary is based on the vocabulary database. The searching of a word in the database starts from the first character of the input word, which should be an English alphabet (A to Z. or a to z). If the first character is not in the English alphabet, it might be a blank, a digit or another characters. The system should be able to recognize these different input content and input format, and respond accordingly. The system uses *trim()* to remove front blank or blanks.

```
searchString = request.getParameter("Search");  
searchString = searchString.trim();
```

If no data has been entered, the system calls dictionaryW.jsp and prompts users to input a word or returns no result.

```
if (searchString.length() == 0) {  
    RequestDispatcher rd = getServletContext().getRequestDispatcher("/dictionaryW.jsp");  
    if (rd != null) {  
        rd.forward(request,response);  
    }  
}
```

If the input does not begin with an English letter, the system also calls dictionaryW.jsp to returns no result.

```
if (asciiValue<65 ||asciiValue>122 || (asciiValue>90 && asciiValue<97)) {  
    RequestDispatcher rd =  
    getServletContext().getRequestDispatcher("/dictionaryW.jsp");  
    if (rd != null) {  
        rd.forward(request,response);  
    }  
}
```

When the input word begins with a letter, the system starts to search the database until it finds the word or arrive the end of the database. If the word is found, the system calls dictionaryR.jsp to output the vocabulary and its corresponding pronunciation, meaning, and example. Otherwise, it returns no result.

5.6.6 User comments

To get the feedback from users, user comments page is designed. All comments are stored into a file named "UserInfo.txt". Each comment contains the user IP address, the user name, the user E-mail address, and the content of the comments. The implementation can be divided into three parts:

1. Get user's comment data from the web page. It uses *request.getParameter()* method.

```
Name = request.getParameter("name");  
Email = request.getParameter("Email");  
comments = request.getParameter("Comments");
```

```
userIP = request.getRemoteAddr();
```

2. Stored the data into session.

```
HttpSession session = request.getSession(true);
```

```
session.setAttribute("userip", userIP);
```

```
session.setAttribute("name", Name);
```

```
session.setAttribute("email", Email);
```

```
session.setAttribute("comments", comments);
```

3. Write the comment into a file in the server side.

```
public void writeFile(String fName) throws IOException{
```

```
    DataOutputStream writeData = new DataOutputStream(
```

```
        new BufferedOutputStream(
```

```
            new FileOutputStream(fName, true)));
```

```
    writeData.writeChars(userIP);
```

```
    writeData.writeChar('\n');
```

```
    writeData.writeChars(Name);
```

```
    writeData.writeChar('\n');
```

```
    writeData.writeChars(Email);
```

```
    writeData.writeChar('\n');
```

```
    writeData.writeChars(comments);
```

```
    writeData.writeChar('\n');
```

```
    writeData.writeChar('\n');
```

```
    writeData.close();
```

```
}
```

6. Comparison with other English learning websites

6.1 Other English learning websites

Many English leaning websites are available on the Internet, which make it difficult to do a complete review. Most of these websites provide materials on grammars, phrases, idioms, reading, and listening. I would like to compare my English Learning Website with the following three popular websites:

- Learn English--- British Council [14]
- BBC World Service [15]
- Learning English [16]

6.1.1 Learn English--- British Council

Learn English Website is provided by the British Council, the United Kingdom's international organization for educational and cultural relations. The main users of this site are kids or professional persons. English-in-game is the strategy of the website. Many grammar games and vocabulary games are provided which make the English learning a funny process. In the testing part, the designers divide English into three levels: elementary level, intermediate level, and advanced level. Reading materials are mainly articles. Vocabulary in the article is summarized at the end of each article. Learning English uses Java Applets and static connection to implement the functions.

6.1.1.2 BBC World Service

BBC Learning English is provided by BBC World Service. The users of this website are the learners who have intermediate English level. This website focuses on learning English through reading and listening to news. Several channels are designed to help

vocabulary learning from the text news, the audio program, the video program and the quizzes. A special business English program is also provided. BBC English is implemented by connecting the static html pages.

6.1.1.3 Learning English

Learning English are launched on the 1st April 2002. It has four major functions: grammar, vocabulary, phrase book & listening practice, and reading practice. Grammar part gives a basic explanation of tenses and other grammatical points. Vocabulary part divides English into three levels: elementary level, intermediate level, and advanced level. Each level includes vocabulary in different categories and is performed by finishing questions. Phrase book & listening practice gives the pronunciation of daily words and phrases. Reading part includes reading, watching and listening news. Learning English implements functions by connecting with static pages.

6.2 Strengths and Weaknesses of ELW

The above-mentioned websites, as well as most of the English learning websites available on Internet, are designed for more general users. These users could be anyone who wants to improve English. To make the learning process more interesting, games, audio and video tools as well as other multimedia methods are utilized. These websites adopt a more static structure, which links static pages. Although database is used in some functions in the websites, such as in testing, the entries in the database are very limited.

Comparing with these websites, the designed ELW has the following advantages:

- Ease-of-use user interface

- Specific user definition --- international students whose native language are not English.
- Specific mission --- help international students to improve English in the new environment, more academic-oriented.
- Dynamic generation of vocabulary and testing. The website is easy to maintain and update.
- Level-oriented --- vocabulary studying and reading are designed for three English levels
- Based on local database --- vocabulary learning, testing and online dictionary are based on local database. The database is easy to maintain and improvement.

The ELW is a good English learning tool for international students. However, it has some limitations. First, it lacks multimedia tools, such as flash, and other forms, such as games, of English learning. These tools and forms could make the English learning more interesting. Second, the vocabulary learning is not closely related with the reading practice. A summary of vocabulary could be provided at the end of each reading material.

7. Conclusion

7.1 Summary

English Learning Website is a web-based application, which provides a friendly interface and functions for user to improve their English. The functions include vocabulary level testing, vocabulary studying, lesson testing, reading, online dictionary, and user comments. The system administrator may update the database directly.

In this report, the software development document is provided, which include introduction, system requirements and analysis, system design, implementation and user manual. UML diagrams (Use Case diagram, Class diagram, and sequence diagram) are used to analyze what and how system should perform [9]. Java is used as the coding language. Tomcat, Servlet/JSP, JDBC, Microsoft Access are used to implement the system. A special database design is included in the project. The using of two-level index in database and searching algorithm has greatly increasing the speed of database operation.

7.2 Future work

Two aspect of ELW should be improved. First, there should be more words and testing questions in the database to make ELW an effective learning system. Second, the system administration could be done through the web page interface so that the system administrators need not work with the database directly.

- Database:

In this project, we only provide a prototype of the English learning website. The database of ELW only contains 600 words and 150 questions. Clearly, this is not enough to make ELW an effective learning system. Future work is to increase the number of words and questions in database.

- System Administration

Automatic system administration through web page interface is not implemented in this project. This is because the direct modification of data entries in database is quite easy. But, it is better that the system administrators could update database through web page so that the administration could be done anywhere. Some basic

function could be add, delete, and modify word and question entries through web page interface, modify index pointers, update reading articles through web page interface, etc.

Reference

- [1] *The Apache Tomcat Project*. Available at <http://jakarta.apache.org/tomcat/>, August 11, 2003.
- [2] *Java 2 Platform, Standard Edition (J2SE)*. Available at <http://java.sun.com/j2se/>, August 11, 2003.
- [3] *Authoring JavaScript*. <http://www.wdvl.com/Authoring/JavaScript>, August 11, 2003.
- [4] S. Brown, R. Burdick. *Professional JSP 2nd Edition*. Wrox Press Ltd, 2001.
- [5] L. Lemay and R. Cadenhead. *Teach Yourself Java 2 platform in 21 days*. Sams Publishing, 1999.
- [6] Toby J.T, James P.F. *Design of Database Structures*. Prentice-Hall inc., 1982.
- [7] `javax.servlet.http` Class `HttpServlet`. Available at <http://jakarta.apache.org/tomcat/tomcat-4.0-doc/servletapi/javax/servlet/http/HttpServlet.html>, August 11, 2003.
- [8] *Session Tracking*. Available at <http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/Servlet-Tutorial-Session-Tracking.html>, August 11, 2003.
- [9] G. Booch, J. Rumbaugh and I. Jacobson. *The Unified Modeling Language User Guide*. Addison Wesley, 1999.
- [10] Yi Liu. *For the beginning learners Vocabulary 5000*. San Huan, 1993
- [11] Yi Liu. *For the intermediate learners Vocabulary 10000*. San Huan, 1994
- [12] Yi Liu. *For the advanced learners Vocabulary 22000*. San Huan, 1994
- [13] *Merriam Webster Online*. Available at <http://www.m-w.com/home.htm>, August 11, 2003.

[14] *Learn English--- British Council.* <http://www.learnenglish.org.uk/index.asp>,
September 8, 2003.

[15] *BBC World Service.*<http://www.bbc.co.uk/worldservice/learningenglish/index.shtml>,
September 8, 2003

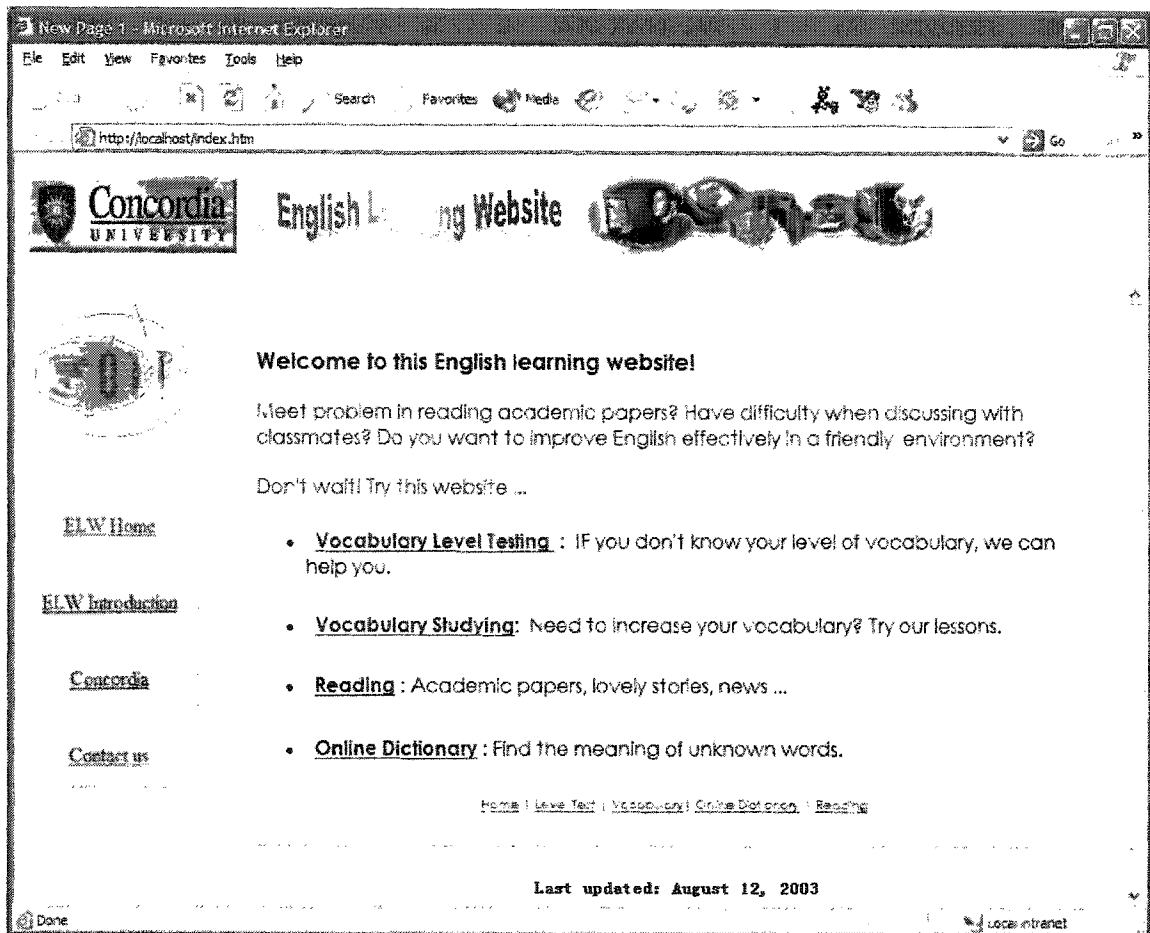
[16] *Learning English.* <http://www.learning-english.com/learnonline/index.htm>,
September 8, 2003

Appendix

A. User's Guide

Please use Internet Explorer 5.0 or higher to browse this website.

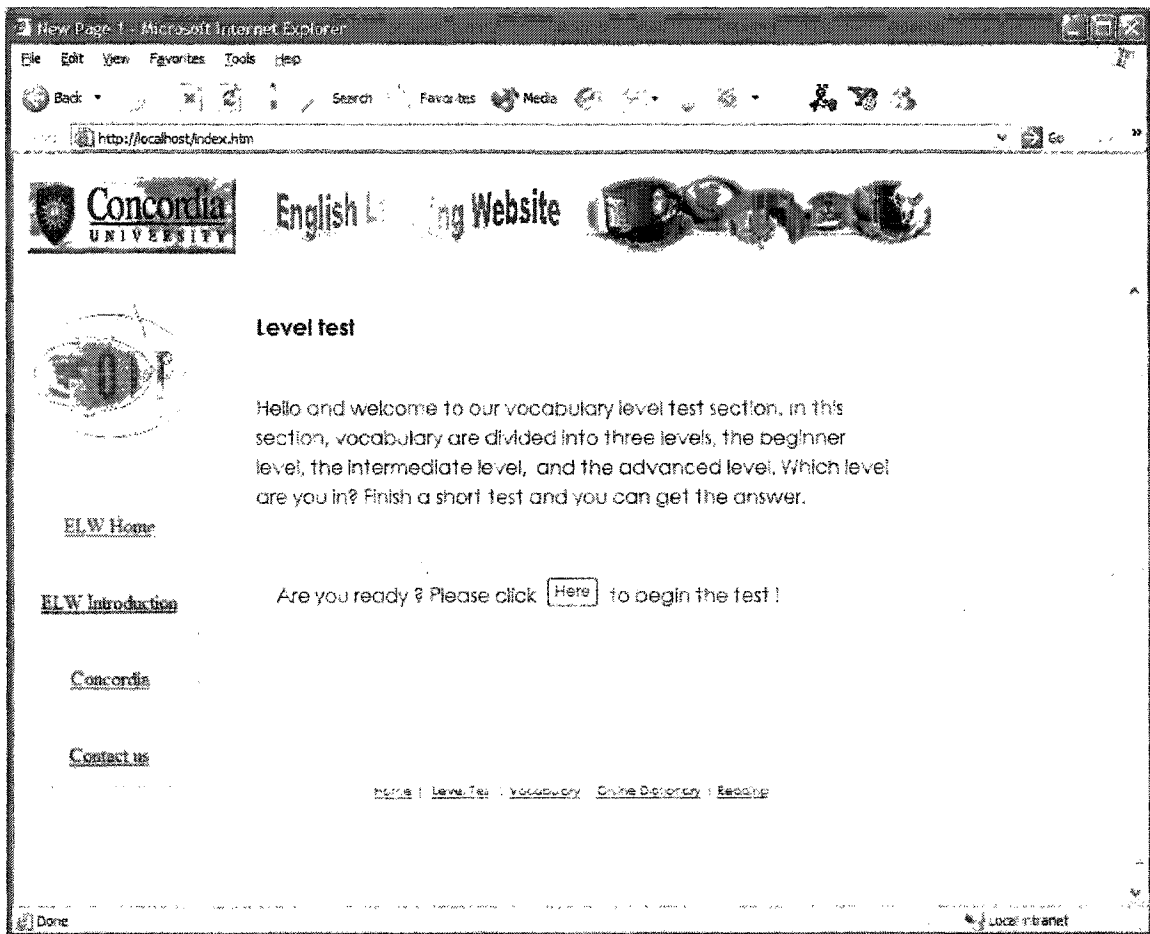
Homepage: [Index.htm](#)



Above is the homepage of ELW. In the left frame, there are five links. “ELW Home” links to the homepage of English Learning Website. “ELW Introduction” gives a simple introduction of the website. “Concordia” links to the homepage of Concordia University. “Contact us” links to the comment pages where user can send comments and ask questions. In the main frame, there are links to the four main functions of the system.

They are “Vocabulary Level Testing”, “Vocabulary”, “Reading”, and “Online Dictionary”. You may use mouse to click the links to enter the functions.

Level Test:



Click the “Vocabulary Level Testing” link in the homepage, you may enters the level test page. Clicks the “here” button to begin the test.

Start Testing:

New Page 1 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media Go

http://localhost/index.htm

Concordia UNIVERSITY English Learning Website

[ELW Home](#)

[ELW Introduction](#)

[Concordia](#)

[Contact us](#)

Level Testing

This test includes 25 single choice questions. Please read the questions carefully and choose an answer by clicking the radio button to the left of the answer you've chosen.

Question 1:

He doesn't seem to have a good _____ of the subject.

groove grab

grasp grind

Question 2:

Rain runs down a _____ from the roof to the ground.

dust spout

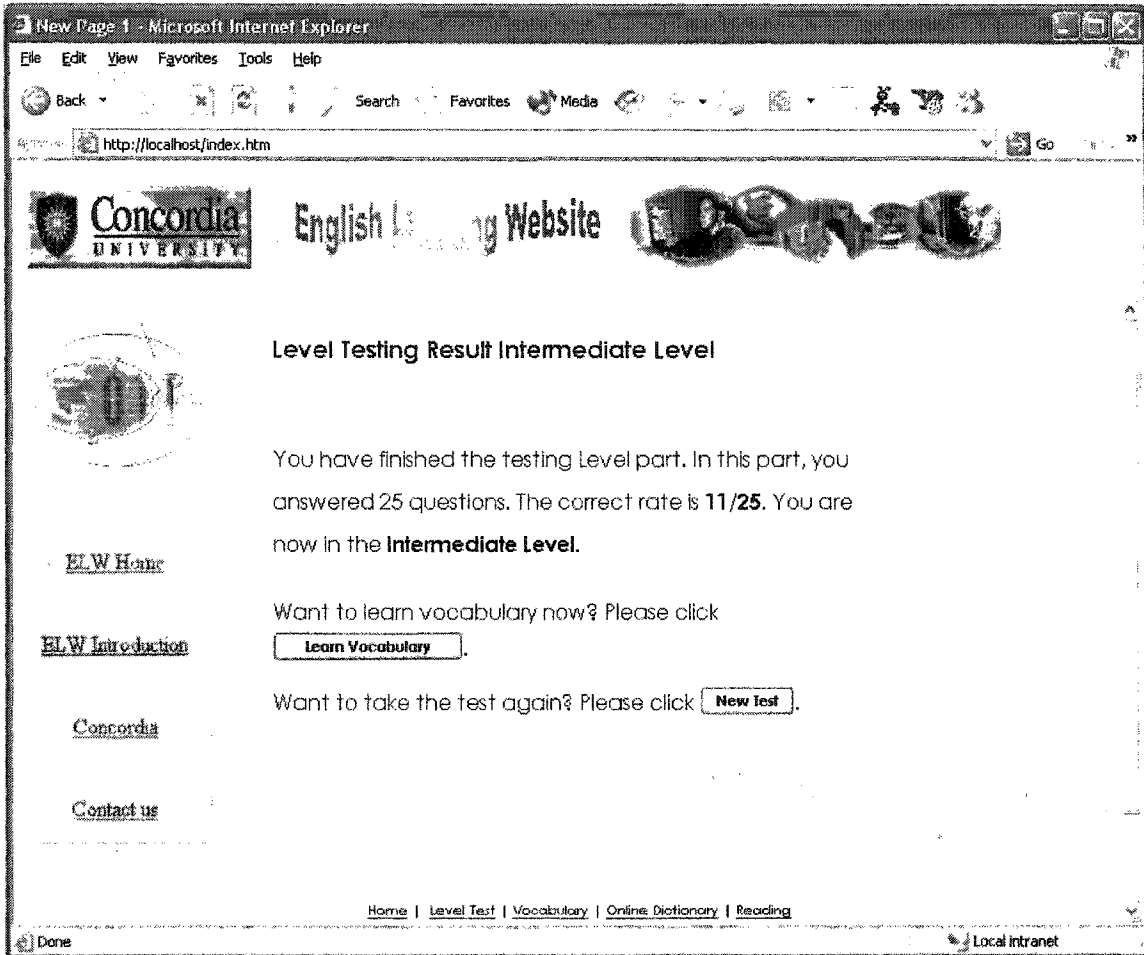
seed filth

Question 3:

Done Local Intranet

There are 25 questions in each test. All the questions are given in a fixed form: Question Number, question and four choices. Choose the correct choice by checking the button before the choice. After you have finished these questions, click "Submit" button to submit the answers.

Testing Result:





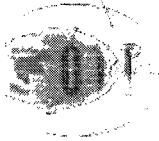
New Page 1 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media Go

http://localhost/index.htm

 **Concordia UNIVERSITY** English Learning Website 



Level Testing Result Intermediate Level

You have finished the testing Level part. In this part, you answered 25 questions. The correct rate is **11/25**. You are now in the **Intermediate Level**.

Want to learn vocabulary now? Please click

Want to take the test again? Please click

[ELW Home](#)

[ELW Introduction](#)

[Concordia](#)

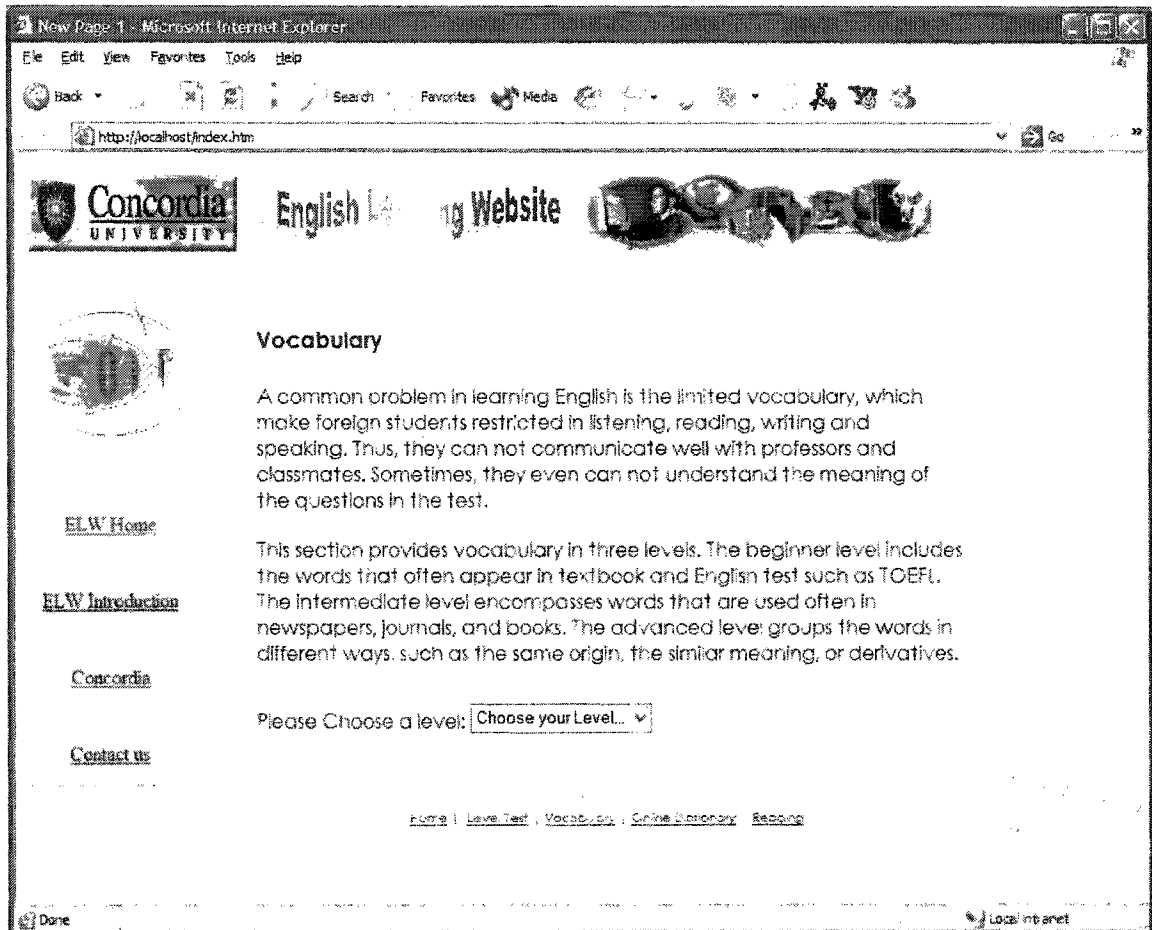
[Contact us](#)

Home | [Level Test](#) | [Vocabulary](#) | [Online Dictionary](#) | [Reading](#)

Done Local intranet

Your testing result will be displayed after you've submitted the answers. A correct rate and your vocabulary level will be displayed. You may select to begin learning vocabulary or do the test again by clicking the corresponding button.

Vocabulary Studying:



The screenshot shows a Microsoft Internet Explorer window displaying the 'Vocabulary' page of the Concordia University English Learning Website. The browser's address bar shows 'http://localhost/index.htm'. The page features a navigation menu on the left with links for 'ELW Home', 'ELW Introduction', 'Concordia', and 'Contact us'. The main content area is titled 'Vocabulary' and contains the following text:

Vocabulary

A common problem in learning English is the limited vocabulary, which make foreign students restricted in listening, reading, writing and speaking. Thus, they can not communicate well with professors and classmates. Sometimes, they even can not understand the meaning of the questions in the test.

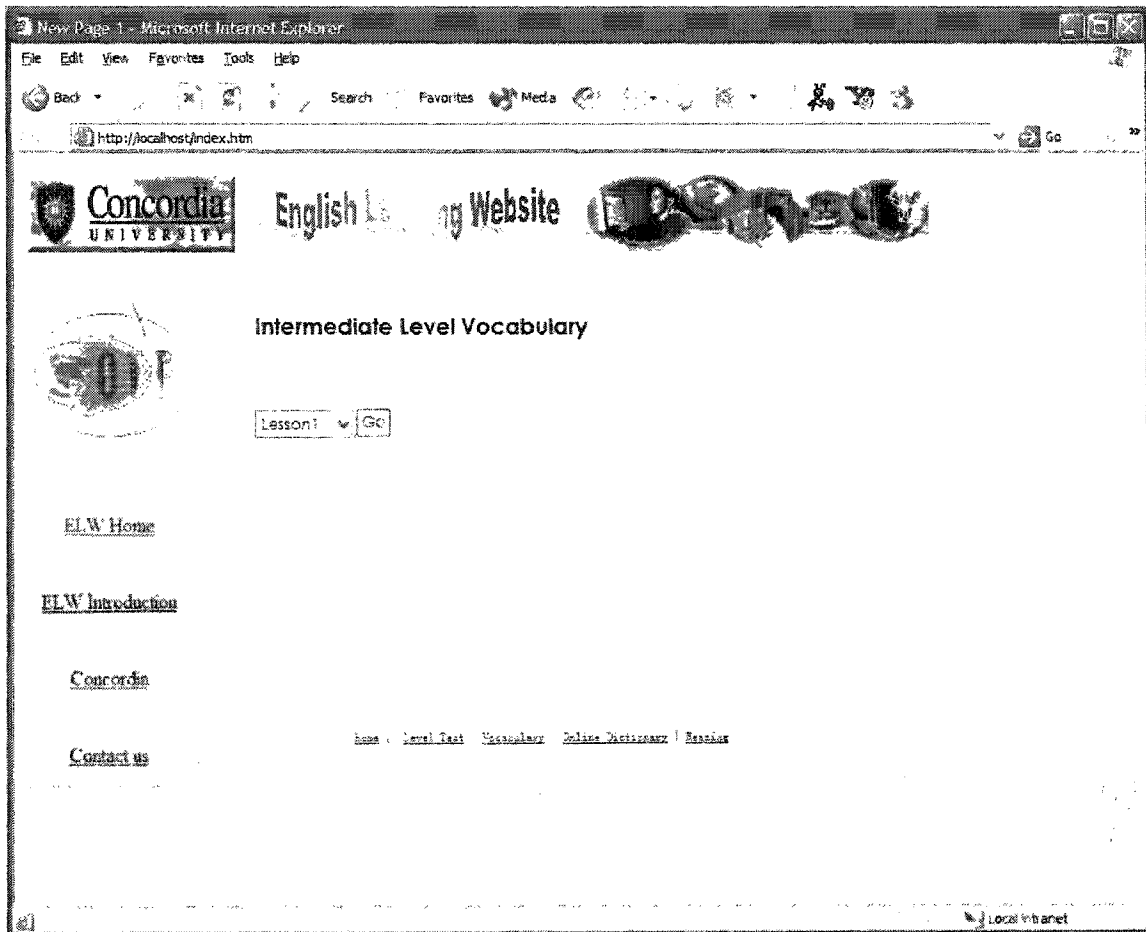
This section provides vocabulary in three levels. The beginner level includes the words that often appear in textbook and English test such as TOEFL. The intermediate level encompasses words that are used often in newspapers, journals, and books. The advanced level groups the words in different ways, such as the same origin, the similar meaning, or derivatives.

Please Choose a level:

At the bottom of the page, there is a navigation bar with links for 'Home', 'Level Test', 'Vocabulary', 'Online Dictionary', and 'Reading'. The browser's status bar at the bottom shows 'Done' and 'Local intranet'.

Press the arrow displayed right to the “Choose your level”, a drop-down menu for vocabulary level will be displayed. You may select the level that you want to start with by clicking the level name. We suggest you select the level according to your level testing results.

Choose Lesson:



For each level, there are ten lessons. Press the arrow right to the “Lesson1”, a drop-down menu will be displayed. You may select the lesson you want to learn by clicking the lesson name.



Lesson:


New Page 1 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media Go

http://localhost/index.htm

 English Learning Website 



[ELW Home](#)

[ELW Introduction](#)

[Concordia](#)

[Contact us](#)

1 **abdomen** [*'abd&men*] the middle part of the body containing the stomach and bowels

Example: The pregnant woman had an enlarged abdomen.

2 **bereave** [*bi'ri:v*] 1. deprive ruthlessly; rob; take away 2. leave desolate and along

Example: 1. The lost hikers were bereft of hope when the rescue plane did not see them. 2. The children were bereaved by the death of their parents.

3 **consecrate** [*'kɒnsikreɪt*] devote to some special use; make holy or sacred

Example: This battlefield is consecrated to the memory of the soldiers who died here.

4 **evoke** [*I'vɒk*] bring out; call forth; cause to appear

Example: A good joke does not necessarily evoke a hearty laugh.

5 **groove** [*gru:v*] a long narrow channel or furrow; corrugation; rut

Example: Wheels left grooves in a muddy dirt road.

6 **jolt** [*dɪɔlt*] shake roughly; shake with sudden jerks

Done Local intranet

In each lesson, a level and lesson number are displayed on the top of page. There are 20 words in each lesson. Each word has its spelling, pronunciation, meaning, and usage example.



After lesson:


New Page 1 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media Go

http://localhost/index.htm

 English Learning Website 



firmness of spirit

Example: She could bear the disappointments of other people with tolerable fortitude.

17 **inconsolable** [*ink&n'soul&b.l*] not to be comforted; broken hearted

Example: The little was inconsolable at the loss of her kitten.

18 **nibble** [*'nib1*] eat away with quick, small bites as a rabbit or a mouse does

Example: Aren't you hungry? You are only nibbling your food.

19 **pageant** [*'padj&nt*] an elaborate spectacle; exposition

Example: The coronation of the new king was a splendid pageant.

20 **scourge** [*sk&:dj*] something or someone which causes great trouble or misfortune

Example: After the scourge of flood usually comes the scourge of disease.

[ELW Home](#)

[ELW Introduction](#)

[Concordia](#)

[Contact us](#)

Lesson2 Testing Back to Intermediate Lessons

Done Local intranet

You may choose to begin the next lesson, begin a lesson test, or return to the level homepage by clicking the buttons at the bottom of the page.

Lesson Testing

New Page 1 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

Address http://localhost/index.htm Go

Concordia UNIVERSITY English Learning Website

Testing Vocabularies in Lesson 1

Question 1:

A doctor's life is _____ to curing poor and sick people.

limited subjected

consecrated depended

Question 2:

Roman emperors built arches to _____ their victories.

encourage record in history

commemorate publicize widely

Question 3:

Choose the word that means the same as: (the)scourge(of Heaven)

bliss delusion

[ELW Home](#)

[ELW Introduction](#)

[Concordia](#)

[Contact us](#)

Done Local Intranet

Lesson testing give 5 questions related to the words in the lesson that you've just learned. Check the correct answers for each question and submit the answer by clicking the "Submit" button, you will see how you've learned.

Lesson Testing Result:

New Page 1 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media Go

http://localhost/index.htm

Concordia UNIVERSITY English Learning Website

Testing Result

Question 1:

A doctor's life is _____ to curing poor and sick people.

A) limited
B) subjected
C) consecrated
D) depended

The correct answer is: C ; Your answer is: D

Question 2:

Roman emperors built arches to _____ their victories.

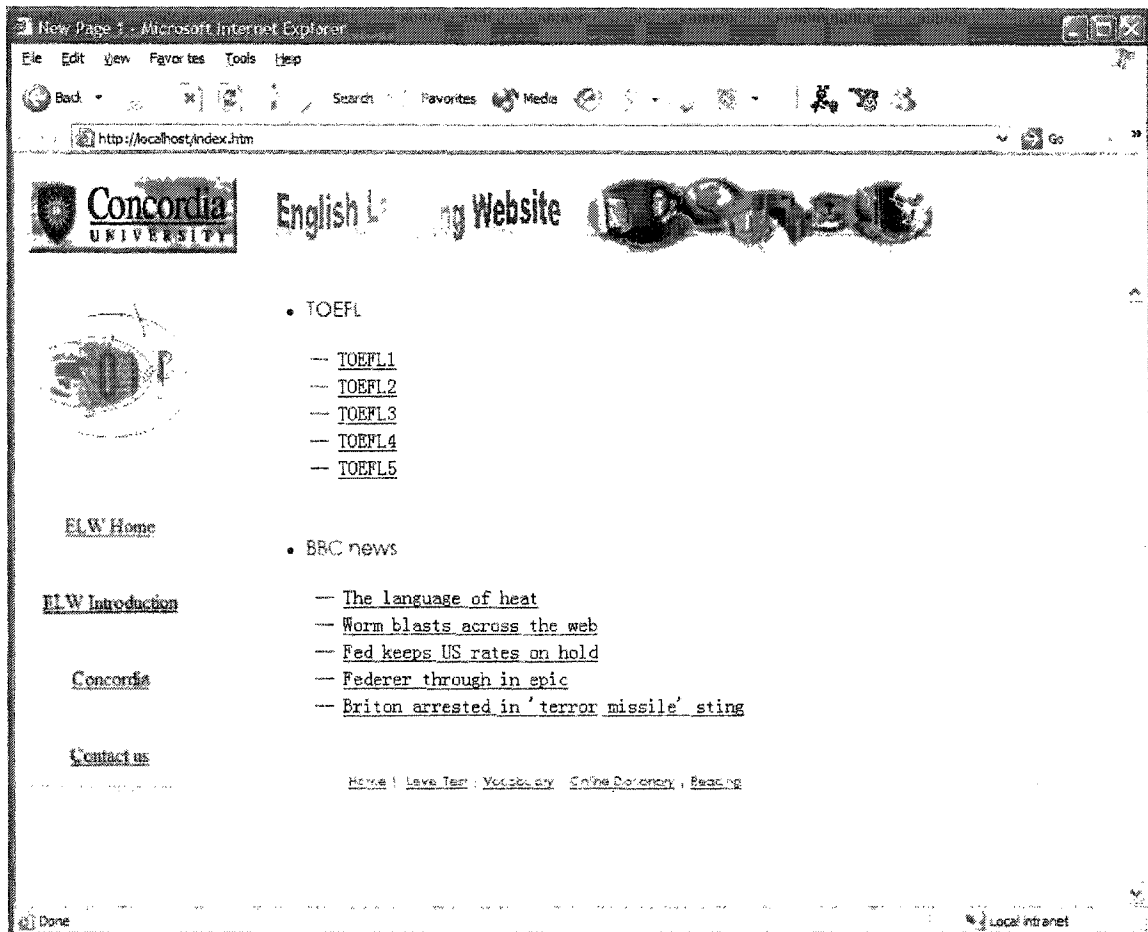
A) encourage
B) record in history
C) commemorate

[ELW Home](#)
[ELW Introduction](#)
[Concordia](#)
[Contact us](#)

Done Local intranet

After you've submitted your answers, correct answers and your answers will be displayed together so that you can see your performance.

Reading:

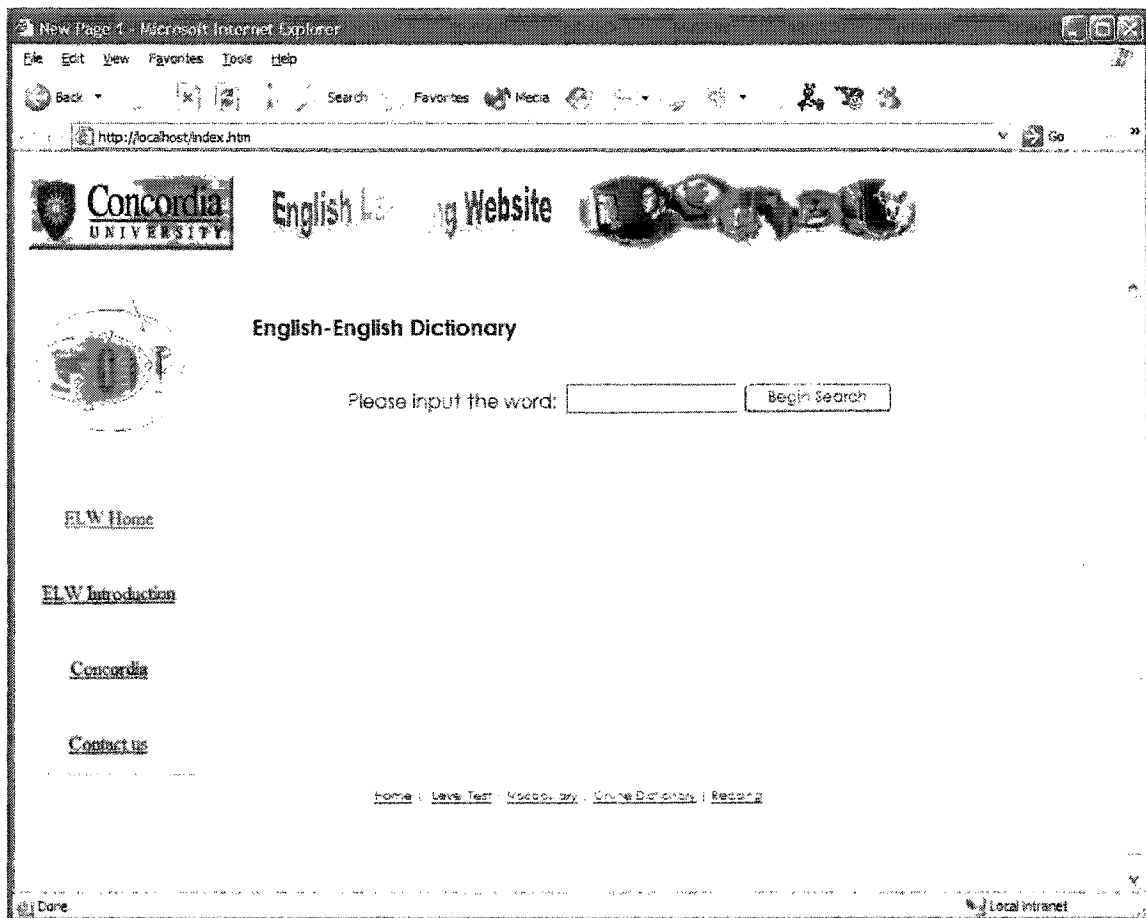


Three categories of reading materials are provided in ELW:

- Stories: These stories are selected from classic literatures, which contain lots of basic vocabulary. It is suitable for those users the beginner vocabulary level.
- TOEFL: These articles are selected from TOEFL reading materials. It is suitable for the users with the intermediated vocabulary level.
- News: Reading and understand English news are the big challenge for anyone whose mother language is not English. These articles are suitable for the users with different levels.

These reading materials are updated by website manager periodically.

Online Dictionary:



You may search a word in our online dictionary. Type the word in the space provided and click "Begin Search" button.

Find Word:

New Page 1 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media Go

http://localhost/index.htm

Concordia UNIVERSITY English Learning Website

English-English Dictionary

Please input word:

anew [ə'nju:] once more; in a new or different way

Example. He made so many mistakes that he had to begin his work anew.

[ELW Home](#)

[ELW Introduction](#)

[Concordia](#)

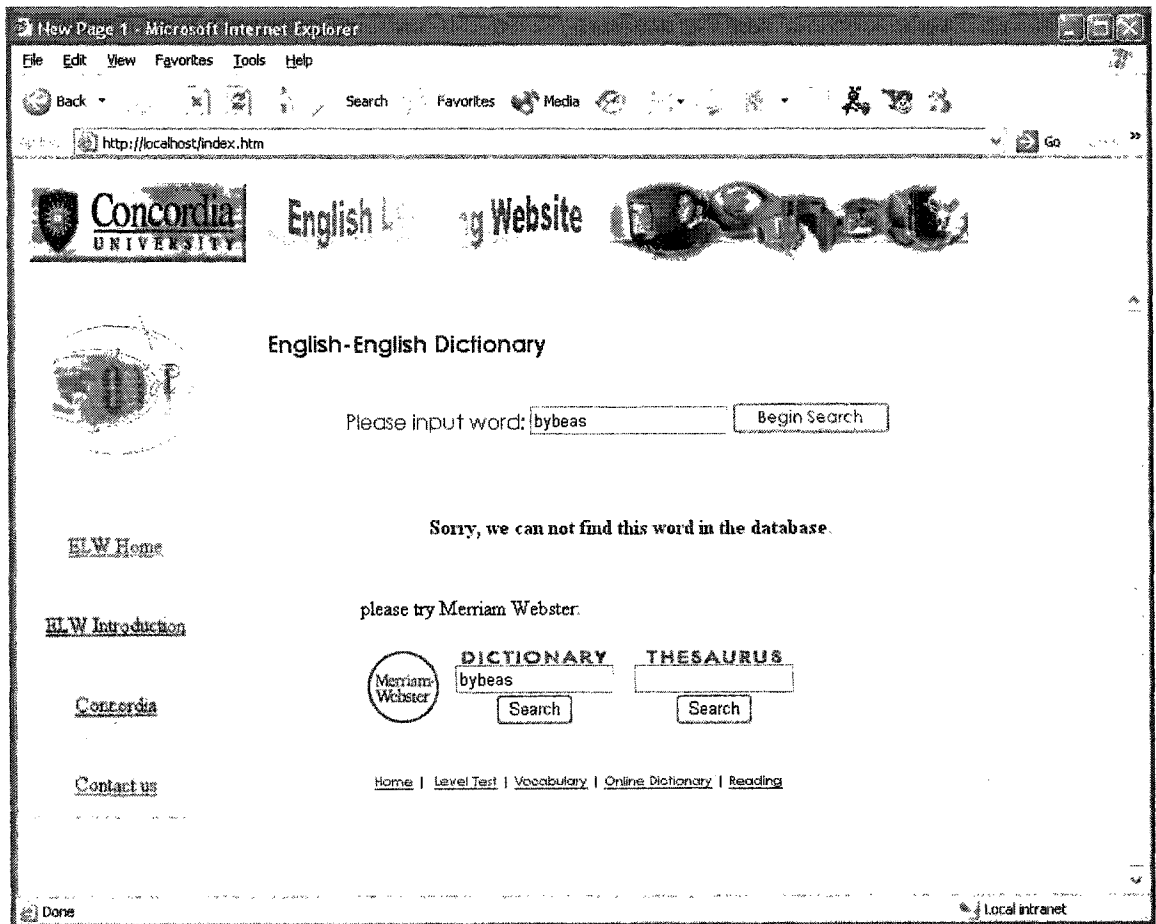
[Contact us](#)

Want to learn more, please check Merriam Webster.

Done Local intranet

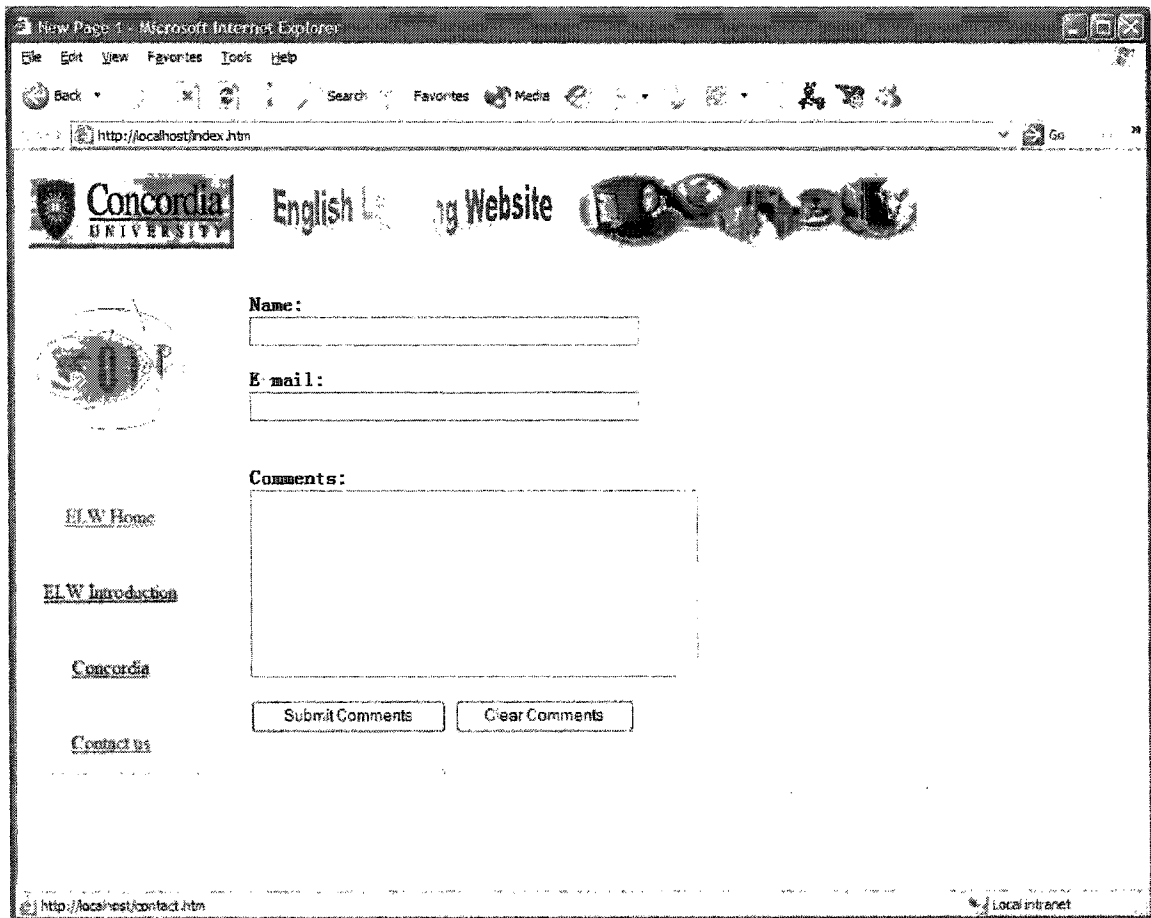
If a word is found, its spelling, pronunciation, meaning, and usage example will be displayed.

Word does not exist:



If the word is not found or you want to find more about the word, you may input the word into the search engine of the Merriam-Webster online dictionary.

User Comments:



The screenshot shows a Microsoft Internet Explorer browser window displaying a web page. The page header includes the Concordia University logo and the text "English Learning Website". The main content area features a comment form with the following elements:

- Name:** A text input field.
- E-mail:** A text input field.
- Comments:** A large text area for entering the comment.
- Submit Comments** and **Clear Comments** buttons.

On the left side of the page, there are several navigation links: [ELW Home](#), [ELW Introduction](#), [Concordia](#), and [Contact us](#). The browser's address bar shows the URL <http://localhost/index.htm>, and the status bar at the bottom shows <http://localhost/contact.htm>.

You may write a comment for this website. Type your name, Email address, and comments in the space provided and click “Submit Comments” button.