Palestine Polytechnic University
College of Administrative Science and Informatics
Information Technology Department

Towards a Knowledgebase for Caspases Substrates

Prepared by:

Abdel Rahman Haddad
Hani Qudsi
Islam Itkaidek

Supervised by:

Dr. Yaquob Ashhab
Dr. Mahmoud Saheb

26/6/2011

23-6-2011

This project was submitted in partial fulfillment of the requirements for the degree B.Sc.

2011

DEDICATION

To who lit the light in my road and made the sun in the darkness, to best woman in the world ***my mother****. To who gave me hope to be myself, for you dear* ***my father****. To the spike of Grains and the fruits of the olive: Our* ***Home Palestine.***

ACKNOWLEDGEMENT

*We would like to take this opportunity to express our thanks to **ALLAH**, who innovates our soul, help us to complete the project ...*

*Then we would like to thanks the Dean of college of administrative science and informatics **Dr. Mohammad Al Desht**. Special thanks to the head of information technology department **Dr. Ismael Al Romi** ...*

*The team members advances deep thanks to our supervisor **Dr. Yaqoub Ashhab** and Co-supervisor **Dr. Mahmoud Al Saheb** who have granted us support, orientation, guidance, help, encourage and advices.*

Then our thanks and gratitude to everyone who contributed to the success of the project and we don't forget our dear teachers, lectures, colleagues, friends ...

ABSTRACT

The idea of the project is to develop a huge and user friendly knowledge database that contains all proteins which are cleaved by Caspases enzymes. The proteins are represented graphically to facilitate the analysis and studying of the Caspases substrates. Each protein file contains rich information such as the cleaving Caspase(s), the exact cleavage position, functional domains, biological implications of the cleavage process, and links for the pathway and interaction for each protein. The database offers easy to use searching tools that allow simple as well as advanced search. The database will be freely available via a web site through Palestine Polytechnic University's server.

The unique feature of this database is its ability to evolve through the contribution of browsers (biologists) from all around the world. The browsers can contribute to grow the database through adding more Caspases substrates or editing already available data. The contributions of browsers were designed in a way to ensure high security data entry as well as to minimize effort needed by the administrators to follow-up and validate the contributions. In addition, our database is capable to update its content by recent data that is available through such as Uniprot and PubMed

الملخص

تكمن فكره المشروع في تطوير قاعدة بيانات معرفيه تحتوي على جميع البروتينات التي تقتطع باستخدام الانزيمات القاطعه، بالاضافه الى تمثيل البروتينات بشكل جرافيكي بحيث يسهل تحليل ودراسة البروتينات و جميع بروتين يحتوي معلومات خاصه عن الانزيم القاطع، موقع القطع، و المناطق الفعاله، والفائده من هذا القطع، بالاضافة الى روابط الى مواقع اخرى توضح تفاعل البروتينات مع بعضها البعض ومعلومات اخرى .

يوفر النظام ايضا اليه للبحث، تسهل على الباحثين الوصول الى معلومات البروتينات، ويتوفر نوعين من اليات البحث وهي اليه البحث البسيط واليه البحث المتقدم، وسيكون النظام متوفر من خلال الشبكه العنكبوتيه، بحيث سيتم وضع النظام على الخادم في جامعه بوليتكنك فلسطين.

ان ما يميز هذا النظام هو قدرة قاعده البيانات المعرفيه على التوسع من خلال مشاركات البيولوجين من جميع انحاء العالم، الباحثون يستطيعون المشاركه من خلال اضافة معلومات عن البويتينات التي تقتطع من الانزيمات القاطعه او من خلال تعديل المعلومات الموجوده مسبقا. ان نظام المشاركه تم تطويره بحيث يراعي أعلى درجات الحماية خوفا من الارسالات الكاذبة، وايضا العمل على تقليل الجهد الذي يلزم المسؤول لادارة النظام والمشاركات، بالاضافه الى قدرة تحديث معلومات قاعدة البيانات المعرفية من خلال موقع موثوقه الا وهي Uniprot و Pubmed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Content:

1. Introduction
2. Problem statement
3. Objectives
4. Standards of biological databases
5. Project domain
6. Project importance
7. Project schedule

1.1 Introduction

Information Technology has had a great effect on the evolution in many fields. After the entering of IT technologies in many areas especially in biological science, biologists try to gain benefits from modern IT applications to make their studies and researches faster and easier. The introduction of such IT solutions has lead to a marked reduction in costs and efforts that are needed for almost all biomedical research projects.

This project aims to create a Knowledge Base that contains Caspases information and to show this information graphically by website. Moreover; the proposed Caspase-Knowledge Base (CKB) will also include a bioinformatic tool that can accurately predict cleavage sites in any protein sequence.

This chapter will be about the problem statement, which explains the problem of the routine searching for homes. Second, the system objectives, which are about the main goal of the system. Third, the project scope, which shows the homes, will cover in which domain. Fourth, the project importance, which explain that benefits that we get from the home finder system. Finally, we put table of tasks that contains each task we did in an expected time.

1.2 Problem statement

After studying other projects and alternatives that are related to our project, and studying biologist's needs and requirements, we found the following main problems:

1- There is no unified database that contains all proteins that is cleavage by Caspases.
2- Most of the websites do not provide professional user-friendly interface.
3- Most of the websites do not offer efficient way to search for proteins information.
4- Exchange information between biological, researchers and website administrators is not efficient.
5- Lack of databases that update its information periodically.

1.3 Objectives

The project aims to achieve the following main objectives:

1. Offer a unified reference for all information related to proteins cleaved by caspases.
2. Provide an efficient way to search for proteins information that cleaved by caspases.
3. Provide a graphical and user-friendly interface to present proteins information (cleavage site and protein domains).
4. Develop a safe mechanism that would allow users around the world to add contribution about proteins information and easy way to send feedbacks to the system administrators.
5. Offer special tools that help administrators to manage the system.

1.4 Standards of biological databases

With the progress of science and the increasing of data, the need to store and share data has increased. The most important data needed to be stored and shared globally is the biological data. Now, the biological data increases significantly, and becomes necessary to store it in electronic databases, but these databases have increased until it reached to dispersion, and the number of biological databases becomes more than one thousand [Pascale Gaudetl et al, 2010] [Taylor,C.F., 2008].

This spreading in biological databases has become a key obstacle in front of browsers which searches to reach a specific biological data, where they have to search in many scattered databases to access specific data.

To solve this problem, group of people proposed a basic standards for biological databases, and these basics are called "BIODBCORE "[Pascale Gaudetl et al, 2010], and it is necessary for all biological databases to apply these standards to support data exchange and integration of biological data.

And these standards [Taylor,C.F., 2008] will be list in the following points:

1. Database name.
2. Main resource URL.
3. Contact information (e-mail; postal mail).
4. Date resource established (year).
5. Conditions of use (free, or type of license).
6. Scope: data types captured, curation policy and standards used.
7. Standards: MIs, Data formats and Terminologies.
8. Taxonomic coverage.
9. Data accessibility/output options.
10. Data release frequency.
11. Versioning policy and access to historical files.
12. Documentation available.
13. User support options.
14. Data submission policy.
15. Relevant publications.
16. Resource's Wikipedia URL.
17. Tools available.

Despite the existence of these standards and applying them in biological databases, the problem is yet not solved completely. Databases have become a unified form, but there are still various dispersed databases.

As our study result of the standards required building a biological database from several scientific papers, we have decided to adopt these standards during the creation of the database for our site.

We find that there are several rating for biological websites, where the rating specifies the level of website and its database, and the commitment of the creator of the website in standards. As well as it determines the size of modern technology that used in creating the site, which facilitates the exchange of information and experiences and facilitates the access of information and present data in an easy and smooth way. Examples of those standards are accessibility, concurrency, data representation standards, quality and consistency.

We want to build a website and a knowledgebase to become in the highest levels of biological websites.

1.5 Project domain

The system will help biological searcher and scientist to find caspases substrate information in unified knowledgebase.

1.6 Project importance

The importance of the project appears in following points:

- Saving time and effort by provide the search mechanism for caspases substrate information.
- Presenting information graphically, that makes the understanding easier.
- Facilitate get up-to-date information and exchange the knowledge.

1.7 Project schedule

Every project needs a set of tasks shown in stages. We will display the stages using textual description, table and Gantt chart.

1. System conception and planning :

   To understanding the system conception, we need to read papers and books about the caspases, caspase substrates and knowledgebase concepts.

2. Requirements Definition:

   In this stage, we study the functional, non-functional requirements and some risks that may be face the system.

3. Requirements specification:

Make complete description of the functional requirements of the system by text and flow charts.

4. System design:

In this stage, we will design the system functions and system database.

5. System implementation:

After finishing system design, we will begin programming the system.

6. System testing:

In this stage, system testing will performed on the entire system or on specific unit code in the context of functional requirement specifications.

7. System documentation:

The documentation will be along the system development process.

The following table (1.1) shows tasks distribution through system development life:

Number of weeks	Tasks	Task number
5	System conception and planning	Task 1
4	Requirements Definition	Task 2
5	Requirements specification	Task 3
3	System design	Task 4
11	System implementation	Task 5
2	System testing	Task 6
30	System documentation	Task 7

Table 1. 1 Tasks distribution.

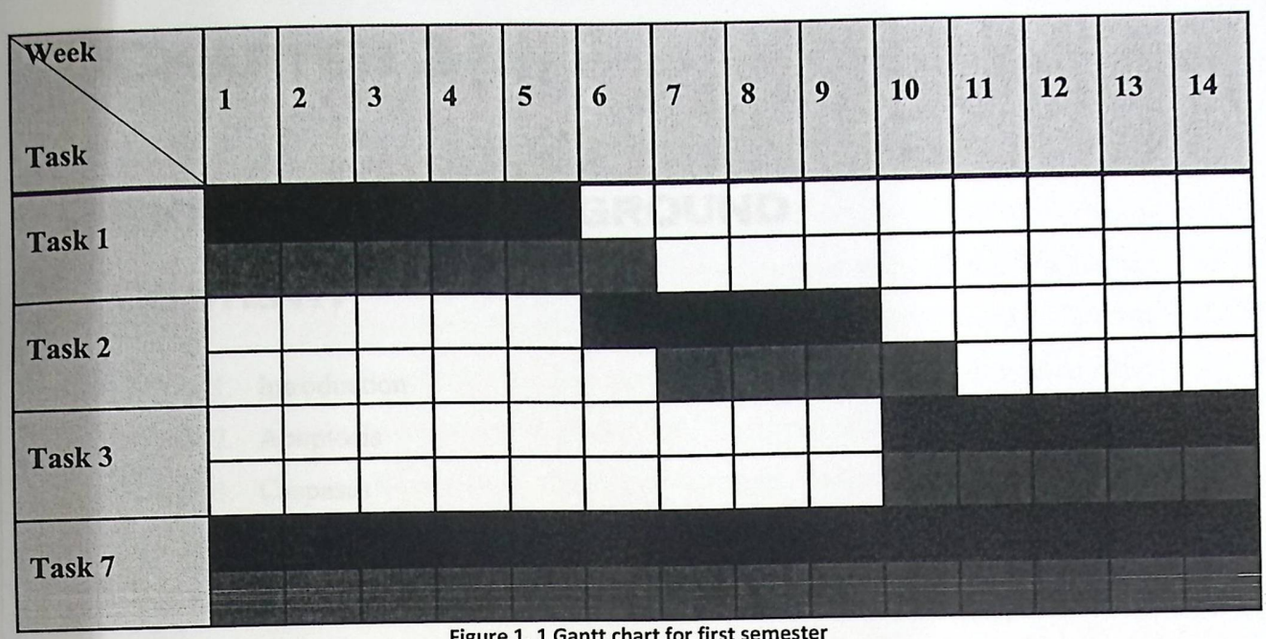


Figure 1. 1 Gantt chart for first semester

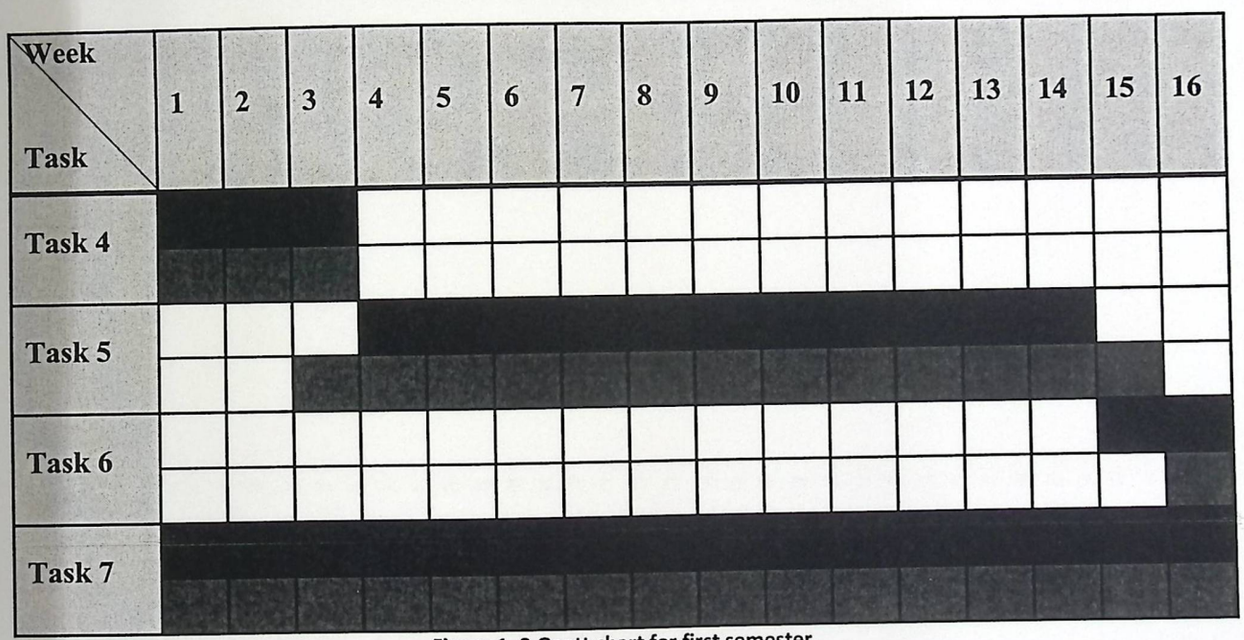


Figure 1. 2 Gantt chart for first semester.

Chapter 2

Biological Background

Content:

1. Introduction
2. Apoptosis
3. Caspases

2.1 Introduction

One of the biology branches is studying apoptosis "Programmed Cell Death" that is initiated and executed by a group of intra-cellular enzymes known as caspases. There are 14 different caspases types (Hammou Oubrahim et al, 2004). The protein targets of these enzymes have reached hundreds and the list is still growing. Caspases cleave their protein targets at a specific motif known as cleavage site. The data about caspases and their targets is growing very fast. Therefore, there is a need to create a Caspase-Knowledge Base (CKB) that would provide efficient management of data, providing the means for the computerized collection, organization, and retrieval of knowledge related to caspases and their targets (substrates).

This project aims to create a Knowledge Base that contains proteins information and show this information graphically by website.

This chapter contains a summary of some important biologically concepts that are necessary to gain a basic background about Caspases.

2.2 Apoptosis

Apoptosis is a normal and constant process in the organisms called programmed cell death. It involves an enzymatic biochemical cascade that leads to change cell morphology and finally it will end in cell death. The major factors that are controlling apoptosis initiation and execution are a group of enzymes known as Caspases [Molly Edmonds, 2010].

2.2.1 Importance of apoptosis

Apoptosis offers many benefits for organism such as:

1. Apoptosis is an essential mechanism for development and maintenance of organism. For example, fingers are connected together in a fetus by skin membrane but apoptosis make it disappear.

2. Apoptosis is the key mechanism to destroy and get rid of the cells that represent a threat to the integrity of the organism, such as cells infected with viruses and cancerous cells.

2.2.2 Apoptosis mechanism

The apoptosis initiated by one of two types of signals [Apoptosis, 2009], this signals are:

1- Internal signals: Signals come from inside of the cell, such as signals come from cell DNA.
2- External signals: Signals come from outside of the cell, such as signals come from immune system.

After apoptosis signal come to the cell, the cell will pass through several stages (Figure 2.1), and these stages are:

1- Shrinking.
2- Appear bubbles on cell surface.
3- The mitochondria start breaking down.
4- Nucleus starts degrading.
5- The cell breaks into small fragments.
6- Lyses of cell body.

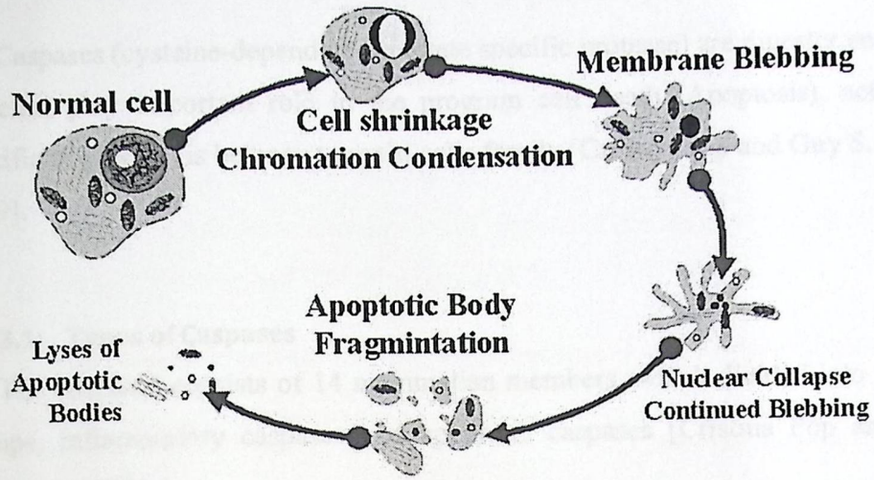


Figure 2. 1 Major steps that takes place when a cell receive an apoptotic signal [Baculovirus-Host Interactions, (2007)]

2.2.3 Apoptosis in health and disease

Apoptosis occurs in normal development and it is necessary to continue organism life. The apoptosis of cells and the creation of new cells cause formation tissues and organs in developing embryos, for example, apoptosis of cells between the toes make them separate from each other [Dash Phil, 2009].

Any problem in regulation of apoptosis can cause serious diseases, such as cancer, neurodegenerative and autoimmune diseases. Cancer is a disease occurs of too little apoptosis process in the body, cancer makes many mutations for the infected cells and makes it to ignore normal signals that come to the cell, so any apoptosis signal will not motivate the cell death, as done in normal cells. On the other hand too little apoptosis can cause Parkinson's and Alzheimer's diseases [Dash Phil, 2009].

As we notice the importance of apoptosis and the diseases that occur when there is any problem in the apoptosis signals, the need of understanding apoptosis and its environment increase to treat these intractable diseases.

2.3 Caspases

Caspases (cysteine-dependent aspartate specific protease) are digester enzymes for proteins, play important role in the program cell death (Apoptosis), activated by specific triggers. It is belongs to multi cells family [Cristina Pop and Guy S. Salvesen, 2009].

2.3.1 Types of Caspases

The caspases consists of 14 mammalian members, which divided into two major groups, inflammatory caspases and apoptotic caspases [Cristina Pop and Guy S. Salvesen, 2009].

- **Inflammatory caspases:**

Linking between cells and immune system enabling apoptosis for inflammatory diseases by activate caspase-1, caspase-4, and caspase-5.

- **Apoptotic caspases contains:**

1. Initiator (activator) caspases: caspases are activation cascade to cleavage the proteins, and this type contains caspase-2, caspase-8, caspase-9, and caspase-10. Moreover, this type of caspases includes more than 90 amino acids.

2. Executioner caspases: caspases are cleavage the proteins in specific positions, and this type contains caspase-3, caspase-6, and caspase-7. Moreover, this type of Caspases includes 20 – 30 amino acids.

When initiator caspases increased its number more than inhibitors in cell that lead to motivate apoptosis.

2.3.2 Substrates of Caspases

During apoptosis several proteins are cleaved, since apoptosis is similar in different cells, so we can expect most important substrate such as cleavage sites.

2.3 Caspases

Caspases (cysteine-dependent aspartate specific protease) are digester enzymes for proteins, play important role in the program cell death (Apoptosis), activated by specific triggers. It is belongs to multi cells family [Cristina Pop and Guy S. Salvesen, 2009].

2.3.1 Types of Caspases

The caspases consists of 14 mammalian members, which divided into two major groups, inflammatory caspases and apoptotic caspases [Cristina Pop and Guy S. Salvesen, 2009].

- **Inflammatory caspases:**

Linking between cells and immune system enabling apoptosis for inflammatory diseases by activate caspase-1, caspase-4, and caspase-5.

- **Apoptotic caspases contains:**

1. Initiator (activator) caspases: caspases are activation cascade to cleavage the proteins, and this type contains caspase-2, caspase-8, caspase-9, and caspase-10. Moreover, this type of caspases includes more than 90 amino acids.

2. Executioner caspases: caspases are cleavage the proteins in specific positions, and this type contains caspase-3, caspase-6, and caspase-7. Moreover, this type of Caspases includes 20 – 30 amino acids.

When initiator caspases increased its number more than inhibitors in cell that lead to motivate apoptosis.

2.3.2 Substrates of Caspases

During apoptosis several proteins are cleaved, since apoptosis is similar in different cells, so we can expect most important substrate such as cleavage sites.

The substrate specificities [JC Timmer and GS Salvesen, 2007] of Caspases is determined in laboratory, so the result has little percentage of accuracy in the real life, some caspases have overlapping in substrate specificities so that mean there's partially overlapping in its functions.

Studies show that caspase-3 and caspase-7 are very similar to each other in their cleavage site specificities, so caspase-3 and caspase-7 almost have the same substrates description [JC Timmer and GS Salvesen, 2007].

The following table will be showing some caspases substrate specificities:

Caspase	**Substrates**	**Cleavage motivation**
Caspase-1	Pro-IL-1ß, U1-70 kDa PARP Pro-Caspases-1, -3, -4	YVHD - A DGPD - G DEVD- G
Caspase-2	PARP	DEVD – G
Caspase-3	PARP U1-70 kDa DNA-PK (cs) Protein Kinase C delta Huntingtin SREBP-1 and SREBP-2 Rb DNA Frag. Factor (DFF)	DEVD - G DGPD - G DEVD - N DMQD - N DXXD DEPD - S DEAD - G DETD-S/DAVD-T
Caspase-4	Similar to Caspase-1 substrate specificity	
Caspase-6	Lamin A PARP U1-70 kDa	VEID - N DEVD - G DGPD – G
Caspase-7	PARP Substrate range like	DEVD - G

	Caspase-3	
Caspase-8	All other Pro-Caspases	
Caspase-9	PARP, And Pro-Caspase-3	DEVD – G
Caspase-10	All other Pro-Caspases	

Table 2. 1 Caspases substrates information [Caspase, (1997)]

Chapter 3

System Requirements

Content:

1. Introduction
2. Functional requirements
3. Nonfunctional requirements
4. Feasibility study
5. Risks analysis
6. Proposed solutions for risks

3.1 Introduction

After we mention problem statements and objectives of project in first chapter, we will describe system requirements (functional and non-functional requirements). Also we will explain the feasibility study. Finally, we will analyze the risks may face the project and propose solutions.

3.2 Functional Requirements

The following points will explain the main functional requirement of the project:

1- Creating and displaying dynamic proteins image (Visualization) depend on database that contains proteins information.
2- Performing basic as well as advanced search.
3- Developing a mechanism that would allow browsers to contribute in adding and correcting data.
4- Showing the location of the protein of interest in a protein interaction network and/or pathway.
5- Periodic update of the knowledgebase.

These functional requirements can be achieved by the system itself, and two types of users, these users are:

1- Administrator
2- Browser

- Administrator requirements:

1. Updating system database manually.
2. Uploading excel file contains proteins information.
3. Accepting or rejecting browsers contributions.
4. View browsers feedback.
5. Add new administrators.

6. Modifying or deleting protein information.

- Browser requirements:

1. Searching in the knowledgebase.
2. View proteins information.
3. Generate contribution and feedback.

3.3 Non-functional requirements

The main nonfunctional requirements are:

1- High security measurements:

The system will apply high security measures by several ways. At first, it is a server administrator responsibility to take periodic system backup to avoid any failure problems. Second, we use database procedures and parameters to avoid SQL injection, SQL injection is an attack in which malicious code is inserted into SQL statement that are later passed to an instance of SQL Server for parsing and execution in order to hack the system. Third, hide login page and access the login page by specific uniform resource location (URL) to avoid unauthorized attempts to access administrator account. Finally, we use the captcha box to avoid any fakes request.

2- Availability:

The system and the knowledgebase will be available for browsers through a web-based application.

3- User friendly interface:

All the system pages must look symmetric by colors, pictures, buttons, and overall look.

4- Scalability:

To keep on modernity for our knowledgebase, the system will implement periodically update for the knowledgebase.

3.4 Feasibility study

This section will describe the software and the hardware needed to develop the project.

3.4.1 Software development resources

The system needs development software resources as explained in table (3.1):

Item	Quantity	Price
SQL server 2008	1	299.95 $
Visual studio 2008	1	799 $
MS office Professional 2007	1	89.45$
MS Visio Standard 2007	1	59.95$
Total		1248.35$

Table 3. 1 Development software resources[software-cheapest, Microsoft and Zteach software].

3.4.2 Hardware development resources

To develop the systems we need to use computer with specific specifications, according to the programs that will be used to build the system.

3.4.3 Recommended hardware resources

The following table (3.2) shows the recommended hardware that need to build our project:

Item	specifications	Cost $
Personal Computer	CPU: 2 GHZ Ram: 2 GB H.D: 80 GB Monitor: 19 Inch	394 $

Table 3. 2 Recommended hardware resources

Source : Afaq co., 15/12/2010.

3.4.4 Human resources

The following table (3.3) show human resources cost

Name	No. of workers	Time in months	Total cost
System analysts	3	3	1500 $
Programmers	3	4	4800 $

Table 3. 3 Human resource cost

3.4.5 Hosting software resources

To access the website, it must be hosted on a web server, this server must have many specifications to run the website and make it available to the users, and this specification is divided into two main sections.

3.4.6 Software requirement

The server must have the following software to host our website and its database:

1- IIS
2- SQL server 2008

3.4.7 Hosting cost

The system will be uploaded on Palestine Polytechnic University server so the hosting cost will be free but to upload in public server its cost around 220$.

3.4.8 Total cost

Table (3.4) show overall project cost

Hardware cost	Software cost	Human resources	Total cost
394 $	1248.35 $	6300 $	7942.35 $

Table 3. 4 Overall project cost

3.4.9 Browsing requirement

When users want to brows our website, they have to consider specific specifications to view and interact with the website in efficient way, and these specifications are:

- Operating system: any.
- Screen resolution: 1024x768+ or above.
- Browser: any browser support Ajax and XML, like :

  1. Internet Explorer 6 and higher.
  2. Firefox 1.0 and higher.
  3. Safari 1.2 and higher (3.0 beta has problems).
  4. Netscape 7.0 and higher.
  5. Camino 1.0 and higher.
  6. Opera 8.5 and higher.

- JavaScript support: yes

3.5 Risks analysis

1- The time constraint, time is limited by the end of course, which is only 29 weeks.

2- Achieve high level of security and reliability.

3- Power failure during the development process and implementation of the system.

4- System failure.

5- We have to understand biological concepts to be able to work in our project effectively.

3.6 Proposed solutions for risks

In this section, we will provide solutions for the project risks:

1. Commitment to time schedule.
2. We will use parameters, captcha box and hide administrator login page to apply security measurements.
3. To avoid suddenly power failure we will use UPS power supply.
4. To avoid system failure and loss data and information we will make periodic system back up.
5. Collect information related to biological concept from trusted resources.

Chapter 4

System Specifications

Content:

1. Introduction
2. General system description
3. Requirements specifications
4. Context diagram
5. Data flow diagram

4.1 Introduction

This chapter will be about general system description, full detailed description of the system's functional requirements, which have been mentioned in the previous chapter. Also we give general database description. Finally, we illustrate some models, which present the functionality provided by a system in terms of data flow diagram.

4.2 General system description

The system consists of knowledgebase that contains all proteins information are cleaved by caspases, proteins information will present graphically in the website. Moreover, the knowledgebase will be update periodically.

The website offers hyperlinks to provide more information about secondary biological topics that are related to cleaved proteins, such as, protein-protein interaction and protein pathway. Moreover, the website has a mechanism to simplify searching by offering word correction and auto complete for basic and advanced search.

The website is supposed to provide a rich environment for researchers, they can exchange knowledge and experiences, by offers a mechanism to submit a feedback about the website and contribute about proteins information.

4.3 Requirements specifications

Functional requirements determined the specific behavior or functions of the system. The functional requirements done by two types of users:

1- Administrator

2- Browser

These requirements will be explain in following points:

4.3.1 Administrator functional requirements

1- An updating system database manually.

Function	An updating system database manually
Description	Enable the administrator to update the knowledgebase by manual request
Input	Proteins accession number from the knowledgebase
Source	Administrator
Output	Download proteins information from Uniprot as xml files then add this information to the knowledgebase.
Requirements	Logging on as administrator and request knowledgebase update

Table 4. 1 Updating system database manually

2- Uploading excel file contains new proteins information.

Function	Uploading excel file contains proteins information
Description	Upload excel file that contains new proteins information
Input	Excel file, contains: Proteins information like accession number, recommended name, cleavage site…
Source	Administrator
Output	Save proteins information in the knowledgebase
Requirements	Logging on as administrator and upload excel file

Table 4. 2 Uploading excel file contains proteins information

3- Accepting or rejecting browsers contributions.

Function	Accepting or rejecting browsers contributions
Description	The administrator able to accept or reject browsers contributions according to its biological correctness
Input	Accept or reject request
Source	Administrator
Output	Add contribution information to the knowledgebase if the contribution accepted or mark the contribution as rejected if the administrator rejects it.
Requirements	Logging on as administrator and view contribution information

Table 4. 3 Accepting or rejecting browsers contributions

4- View browsers feedback.

Function	View browsers feedback
Description	The administrator view browsers feedback about system interfaces and capabilities
Input	View feedbacks request
Source	Administrator
Output	Feedback information
Requirements	Logging on as administrator.

Table 4. 4 View browsers feedbacks

5- Add new administrator.

Function	Add new administrators
Description	The administrator can add new administrators
Input	Administrator information
Source	Administrator
Output	Save new administrator information in the database
Requirements	Logging on as administrator.

Table 4. 5 Add new administrator

6- Modifying or deleting protein information.

a. Modifying existing proteins information

Function	Modifying protein information
Description	The administrator can modify proteins information
Input	View current protein information request
Source	Administrator
Output	Save modified information in the knowledgebase
Requirements	Logging on as administrator and view existing protein information request.

Table 4.6 Modifying protein information

b. Deleting proteins information

Function	Delete protein information
Description	The administrator can delete protein from the knowledgebase
Input	Protein accession numbers
Source	Administrator
Output	Delete protein information from the knowledgebase
Requirements	Logging on as administrator and delete protein request.

Table 4. 6 Deleting protein information

7- View reports.

Function	View reports
Description	The administrator can view reports about the system and about the knowledgebase
Input	Select report to view request
Source	Administrator
Output	View report information
Requirements	Logging on as administrator and select desire report request.

Table 4. 7 View reports

4.3.2 Browser functional requirements

1- Searching in the knowledgebase.

a. Basic search

Function	Searching in the knowledgebase
Description	Browsers can search about proteins information.
Input	Keyword
Source	Browser
Output	Search result
Requirements	Keyword to search in the knowledgebase.

Table 4. 8 Searching in the knowledgebase

b. Advance search

Function	Searching in the knowledgebase
Description	Browsers can search about proteins information in details
Input	Select field to search in it, keyword, logical operations
Source	Browser
Output	Search result
Requirements	Select field to search in it, keyword, logical operations

Table 4. 9 Searching in the knowledgebase

2- View protein information.

Function	View protein information
Description	After browsers search about keywords, they are able to view proteins information
Input	Search result as protein accession number
Source	Browser
Output	Protein information
Requirements	Protein accession number

Table 4. 10 View protein information

3- Contribution and feedback.

a. Add contribution

Function	Contribution
Description	Browsers can add contributions to add new or modify proteins information
Input	Contribution form information
Source	Browser
Output	Save contribution in the database
Requirements	Enter contribution form information and security number

Table 4. 11 Contribution

b. Add feedback

Function	Feedback
Description	Browsers can add feedback about the system
Input	Feedback form information
Source	Browser
Output	Save feedback information in the database
Requirements	Enter Feedback form information and security number

Table 4. 12 Feedback

4.4 Context Diagram

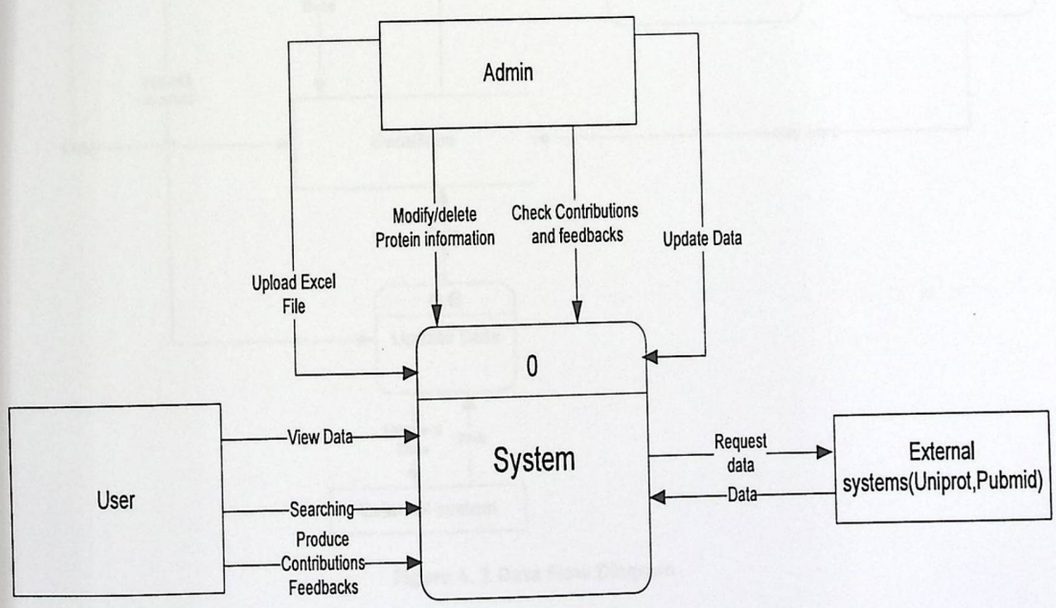


Figure 4. 1 Context Diagram

4.5 Data Flow Diagram

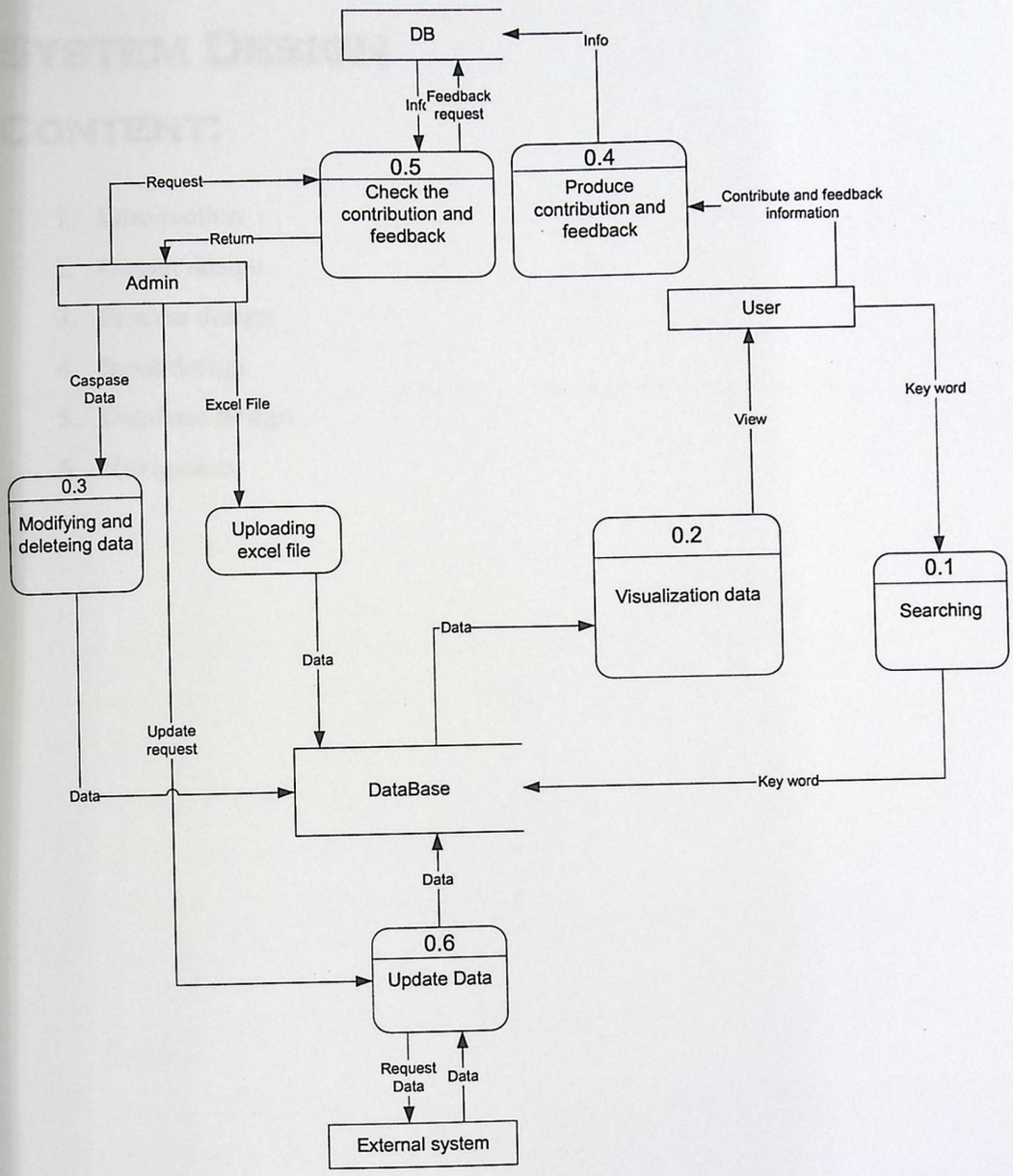


Figure 4. 2 Data Flow Diagram

CHAPTER 5

SYSTEM DESIGN

CONTENT:

1. Introduction
2. Output design
3. Process design
4. Input design
5. Database design
6. Navigation

5.1 Introduction

This chapter describes the system design of the project, which contains output, input, database and process design.

5.2 Output design

This section describes the screens through which the data can obtained, such as explicit data. Administrators and browsers can preview the output data.

5.2.1 Administrator

There are special screens to show data and settings to the administrators to help them in doing their jobs, and these screens are:

1- Reports
2- View all contributions
3- View modifying exist protein contribution
4- View add new protein contributions
5- View feedbacks

- **Reports**

The system offer reports for administrators to help them in made statistics and manage information. Show report example in figure (5.1):

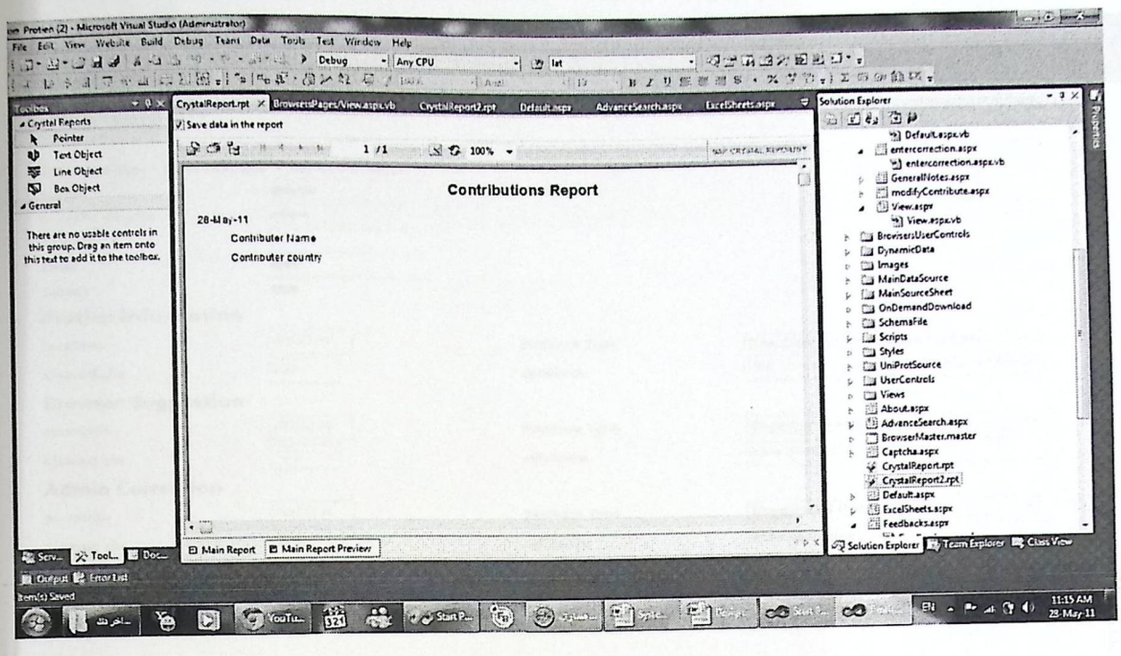


Figure 5. 1 Administrator report example

- Browse all contributions from browsers, as shown in figure (5.2).

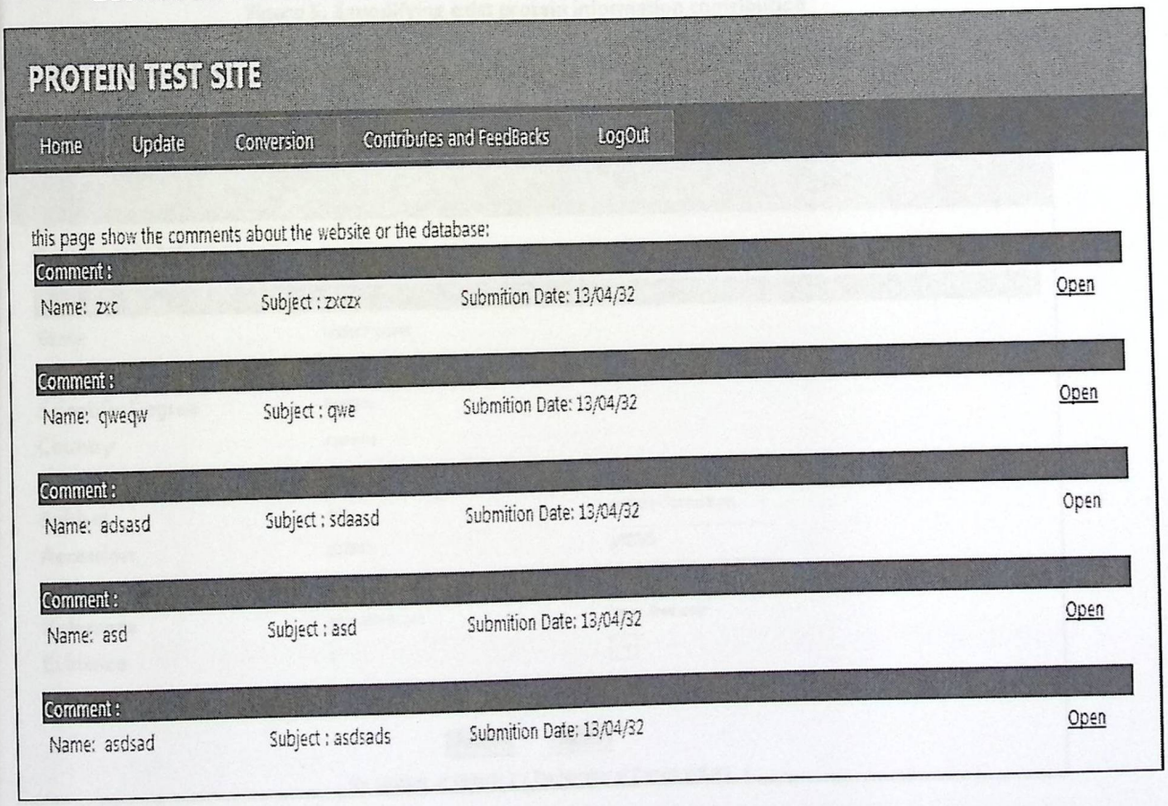


Figure 5. 2 View all browsers contributions

- Show modifying exist protein Contribution in order to accept or reject it, as shown in figure(5.3).

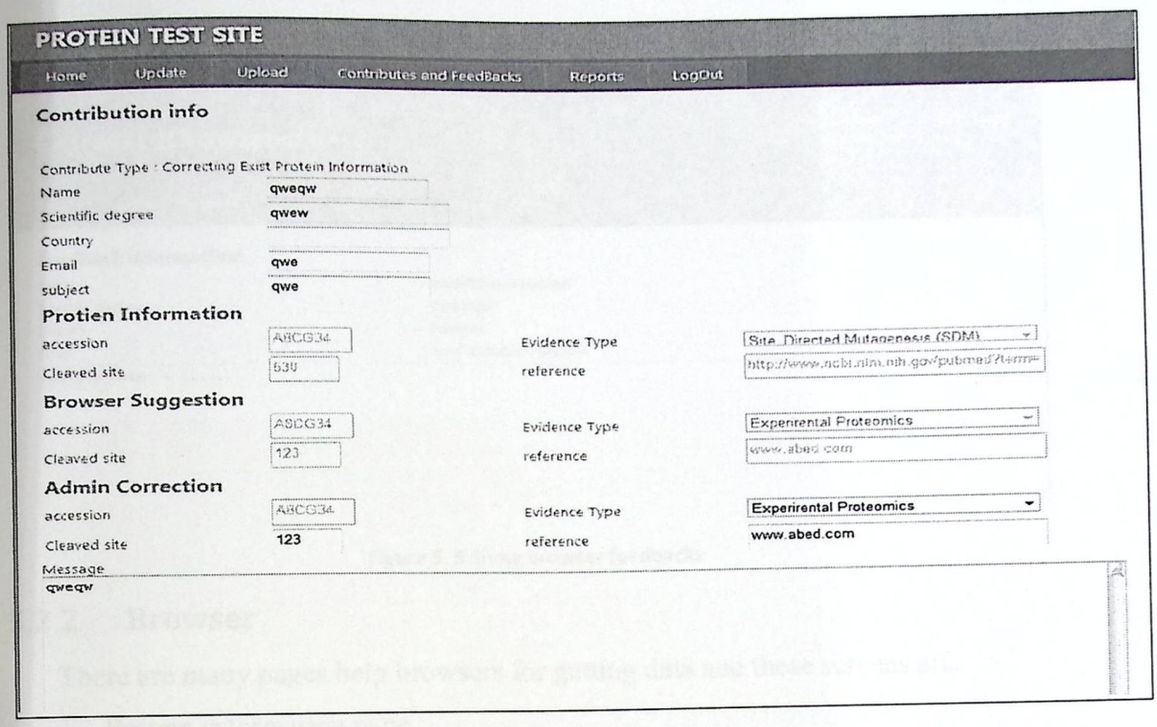


Figure 5. 3 modifying exist protein information contribution

- Show adding new protein Contributions, as shown in Figure (5.4).

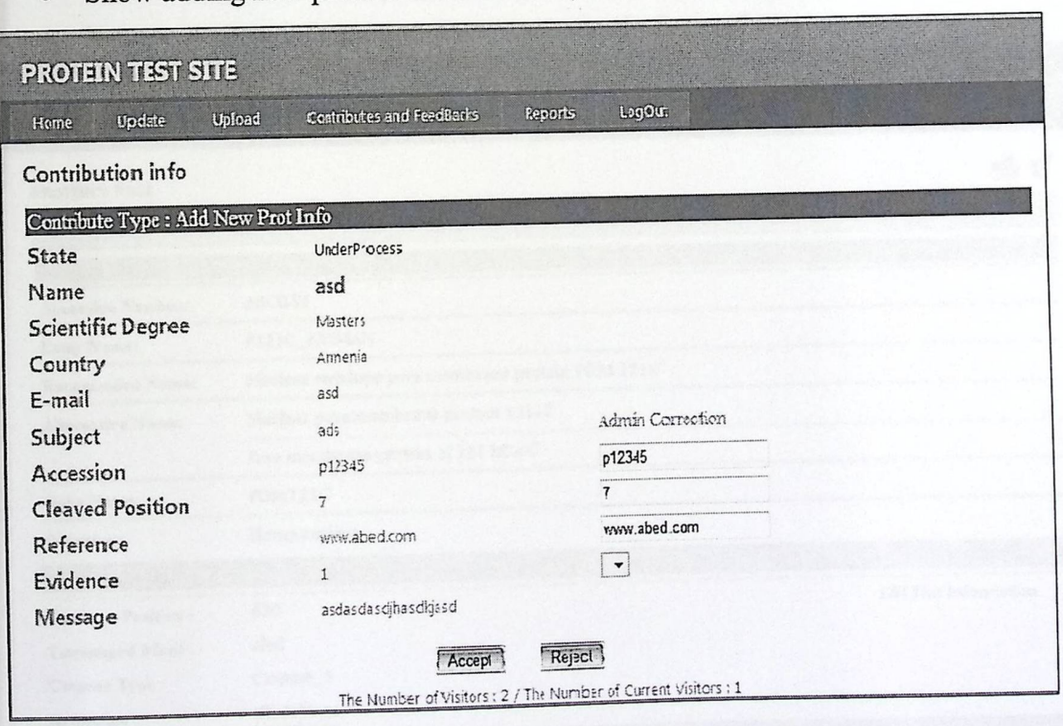


Figure 5. 4 Adding new Protein contribution

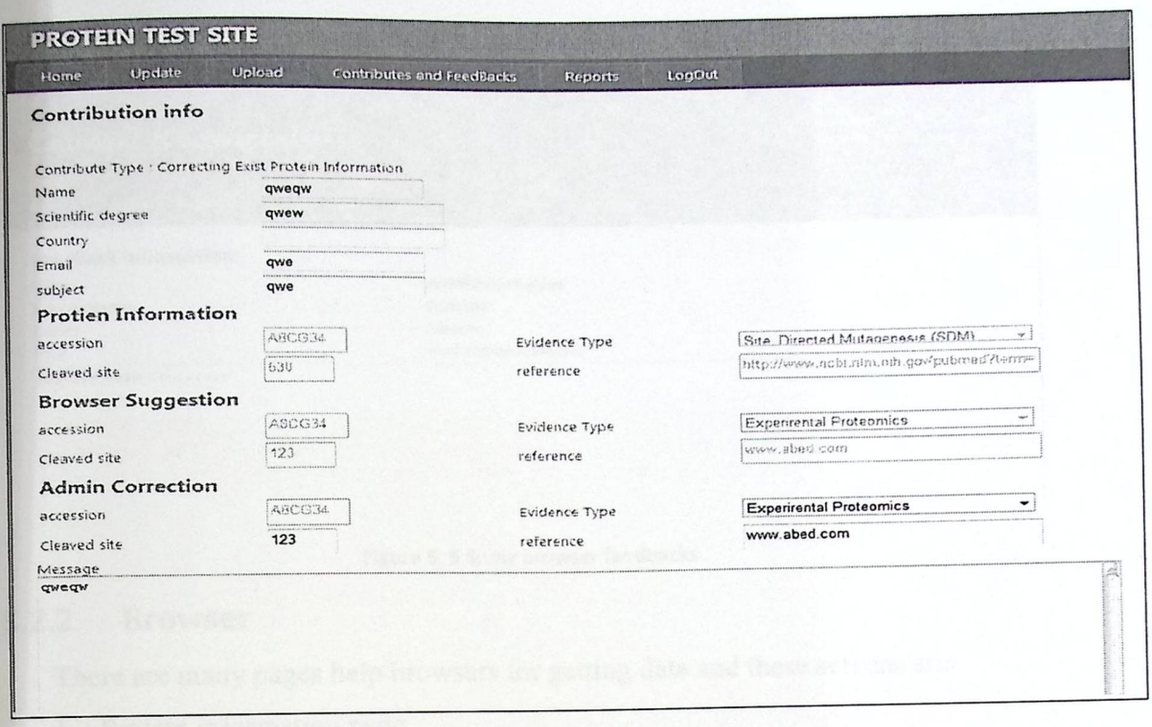


Figure 5. 3 modifying exist protein information contribution

- Show adding new protein Contributions, as shown in Figure (5.4).

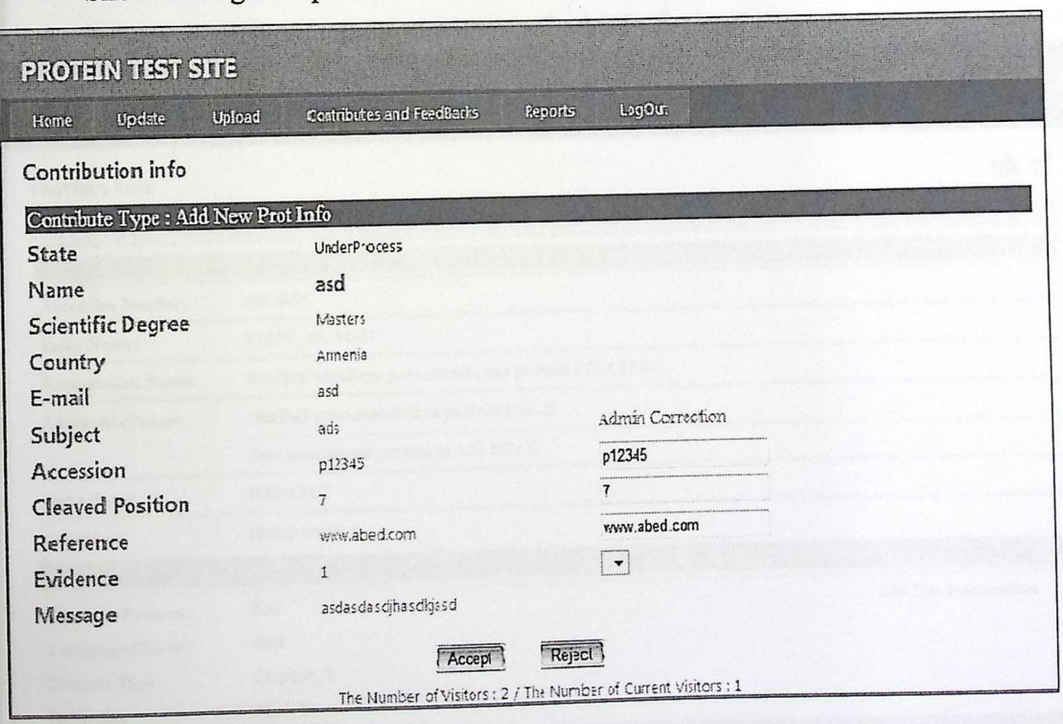


Figure 5. 4 Adding new Protein contribution

- Show browsers general feedbacks, as shown in Figure (5.5).

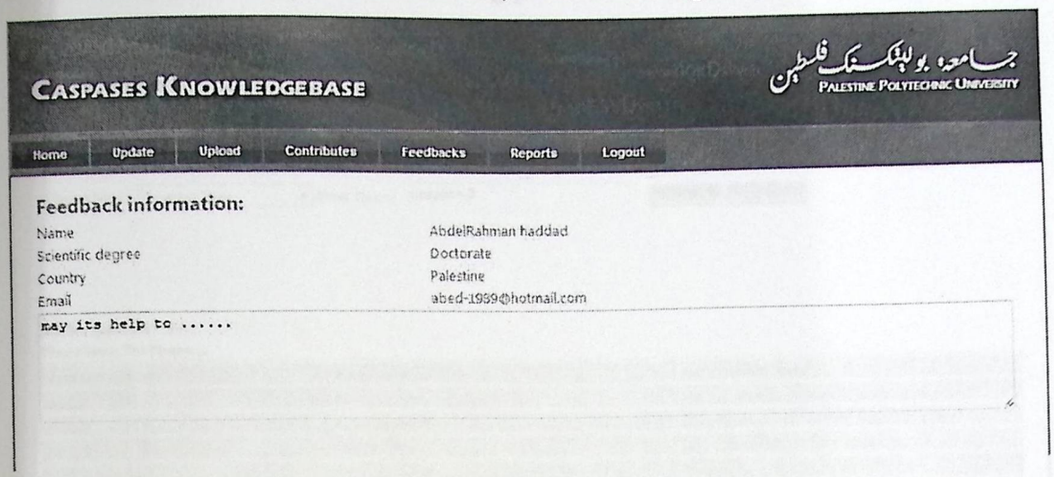


Figure 5. 5 Show browser feedbacks

5.2.2 Browser

There are many pages help browsers for getting data and these screens are:

1- Protein information page

2- Result of the searching process

- Browse protein information page, as shown in figure(5.6):

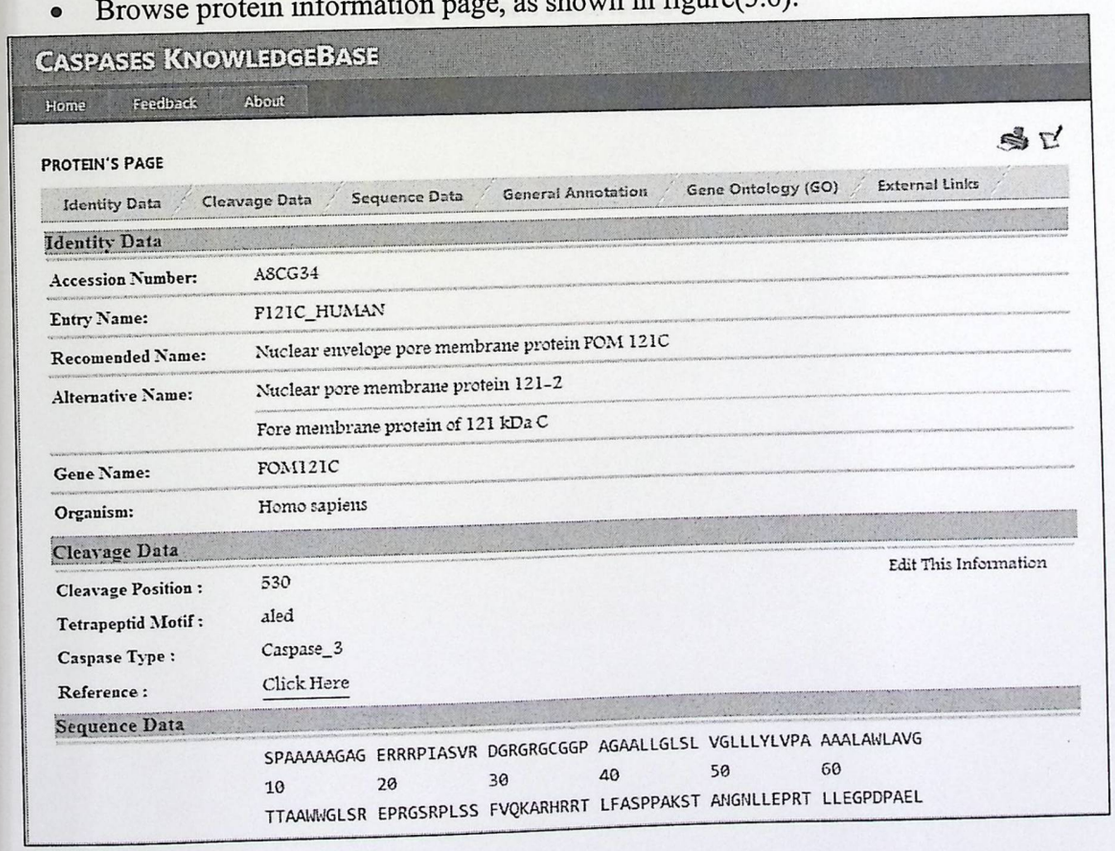


Figure 5. 6 View protein information

- Browse results from the searching process screen, as shown in Figure (5.7):

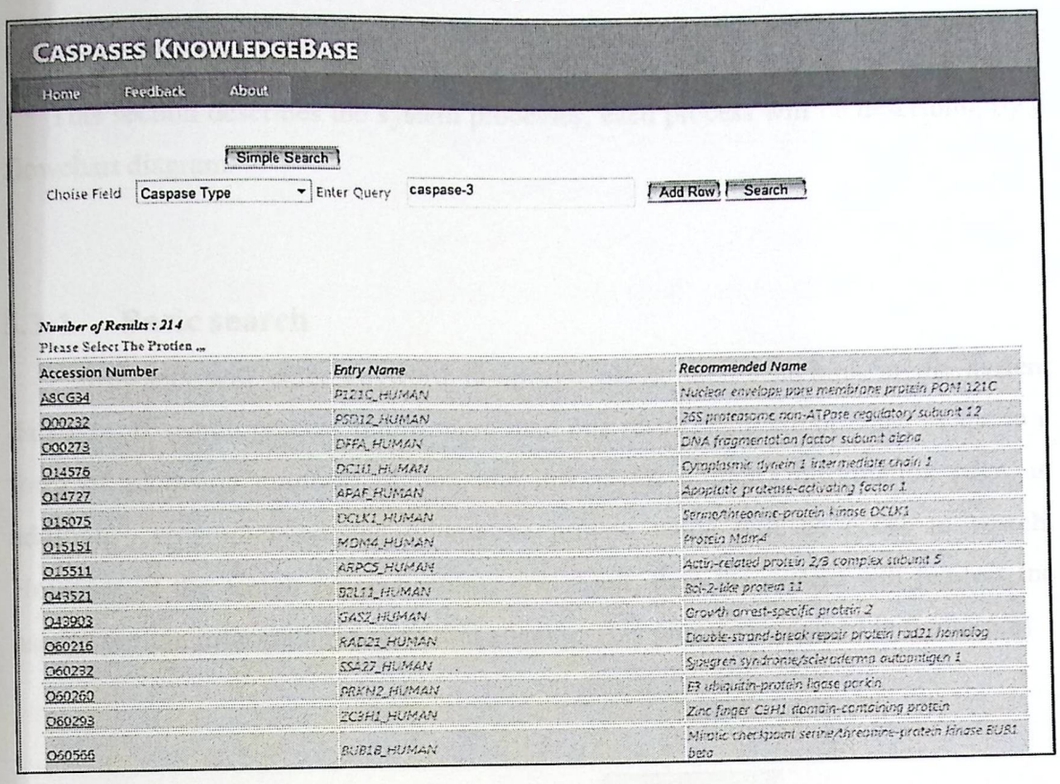


Figure 5. 7 View search results

5.3 Process Design

This section describes the system processes, each process will be describing by a flowchart diagram.

5.3.1 Basic search

Browsers will enter keyword in special place, then searching in the system database. If search process return more than one result, they are be view as list in a form of accession numbers as link, the link is for protein's information page, in addition, recommended and alternative protein name will be occur in search. However, if the search process returns one result, then the system will redirect the user to protein information page automatically. This process explained in figure(5.8).

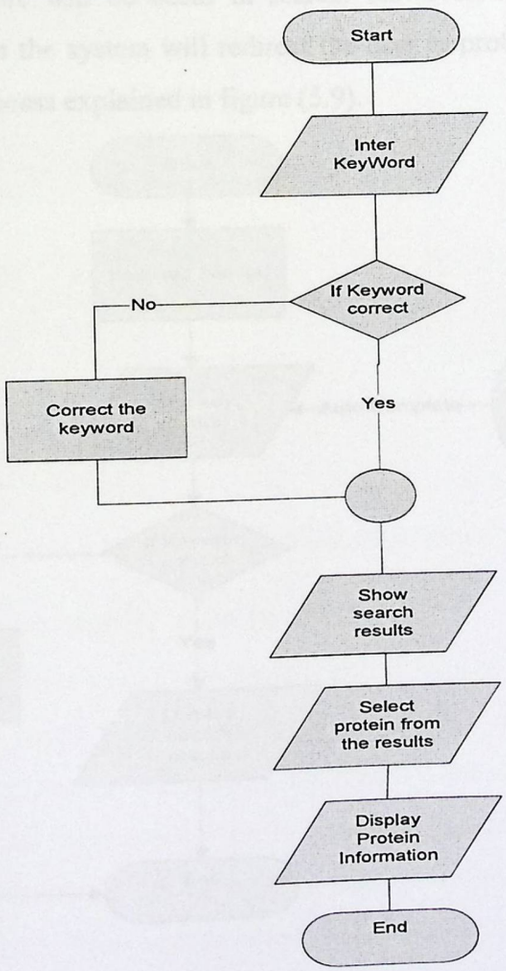


Figure 5. 8 Simple search process

5.3.2 Advance search

This technique able browser to search in details and specific fields. Browsers can be select field to search about it, and then enter the keyword that related with selected field. If the browser selected more than one fields, the user can connect between these fields by one of the logical relationship (And, Or, and Not).

The system offers auto complete for each field, when the browser enter a keyword, the system will suggest keyword completion to facilitate search process.

After selecting the fields and enter keyword, the browser click on search button, then the system will search in the database and showing the results. If search process return more than one result, they are be view as list in a form of accession numbers as link, the link is for protein's information page, in addition, recommended and alternative protein name will be occur in search. However, if the search process returns one result, then the system will redirect the user to protein information page automatically. This process explained in figure (5.9).

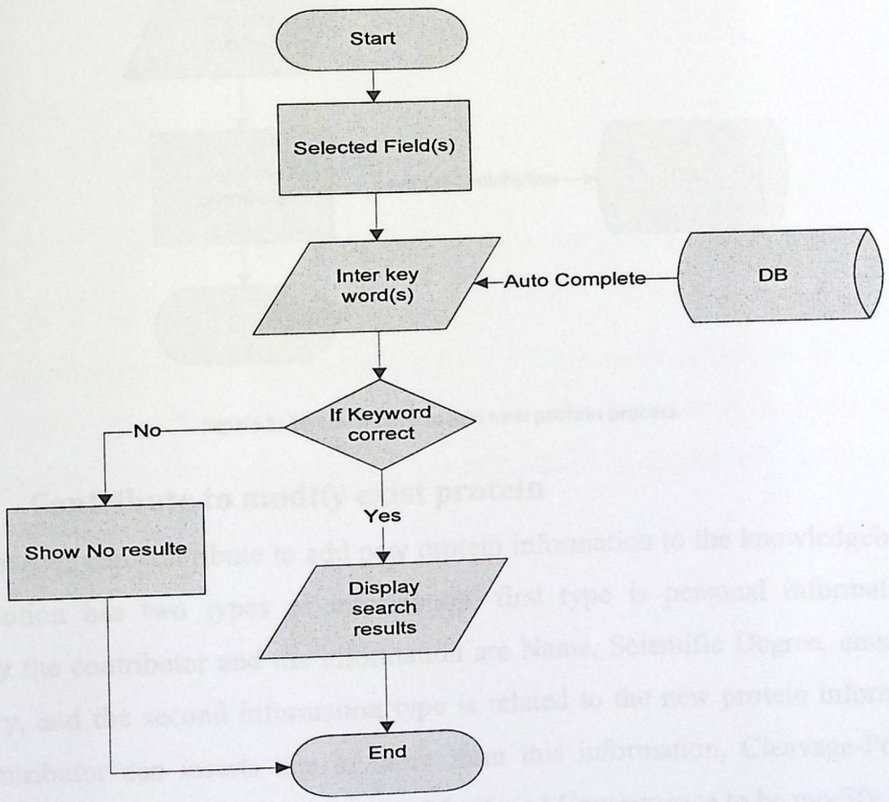


Figure 5. 9 Advance search process

5.3.3 Contribute to add new protein

Browsers can contribute to add new protein information to the knowledgebase. A contribution has two types of information, first type is personal information to identify the contributor and this information are Name, Scientific Degree, email, and Country, and the second information type is related to the new protein information and this information are Accession-Number, Cleavage-Position, Evidence, Cleavage assay, Tetra peptide Motif, and Consequence.

Finally the browser should enter security number (in captcha box), then save the contribution. This process explained in figure (5.10).

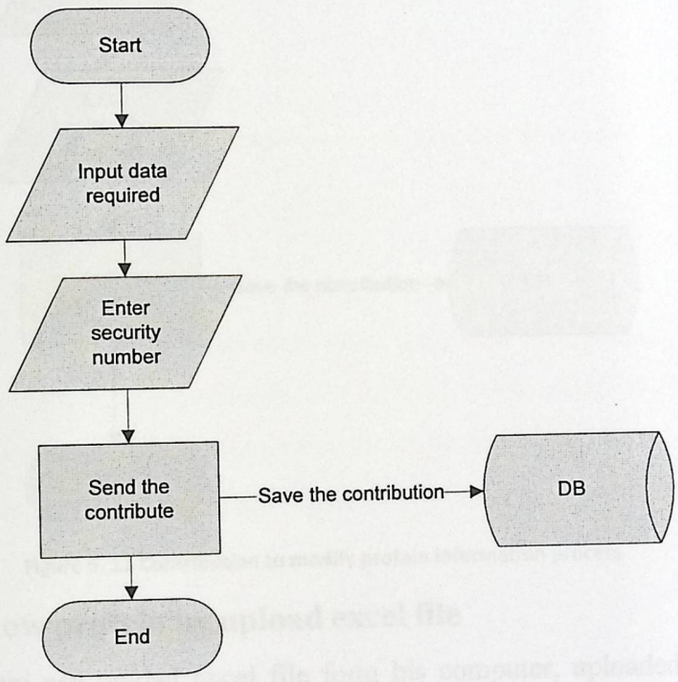


Figure 5. 10 Contribute to add new protein process

5.3.4 Contribute to modify exist protein

Browsers can contribute to add new protein information to the knowledgebase. A contribution has two types of information, first type is personal information to identify the contributor and the information are Name, Scientific Degree, email, and Country, and the second information type is related to the new protein information, the contributor can inserts one or more from this information, Cleavage-Position, Evidence, Cleavage assay, Tetra peptide Motif, and Consequence to be modify.

Finally the browser should insert security number (in captcha box), then save the contribution. This process explained in figure (5.11).

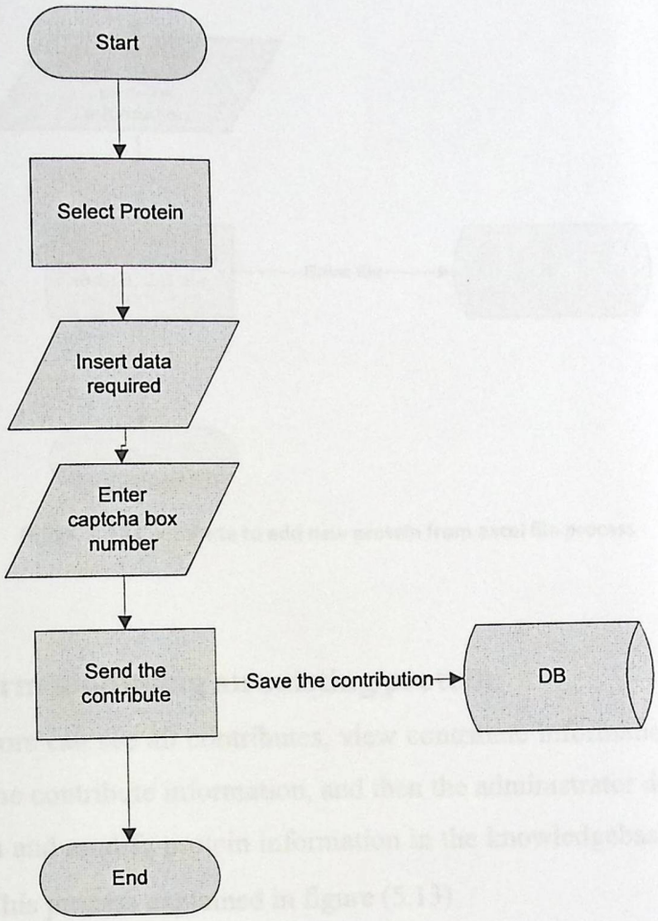


Figure 5. 11 Contribution to modify protein information process

5.3.5 Add new protein by upload excel file

Administrators can upload excel file form his computer, uploaded files contain new proteins, when the administrator upload the file, file information will transfer from excel to SQL DB, then save proteins information in the system database. This process explained in figure (5.12).

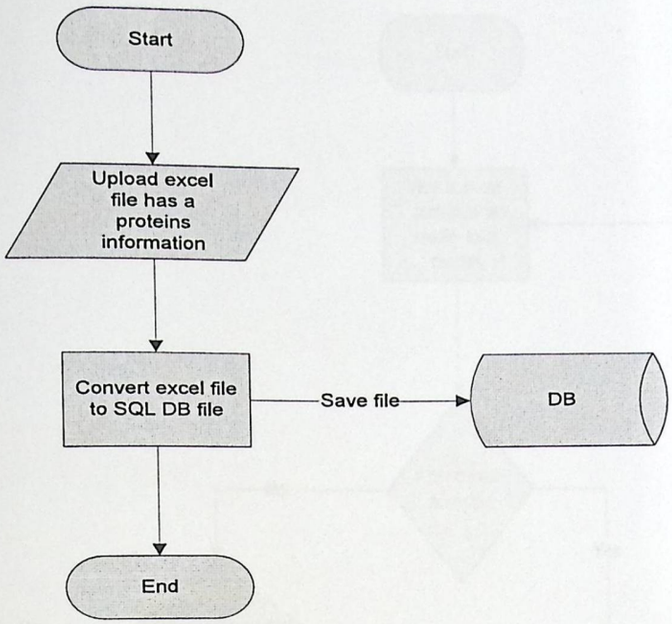


Figure 5. 12 Contribute to add new protein from excel file process

5.3.6 Confirm modifying an existing protein

Administrators can see all contributes, view contribute information, verifying the correctness of the contribute information, and then the administrator decide to confirm the contribution and modify protein information in the knowledgebase or rejected the modification. This process explained in figure (5.13).

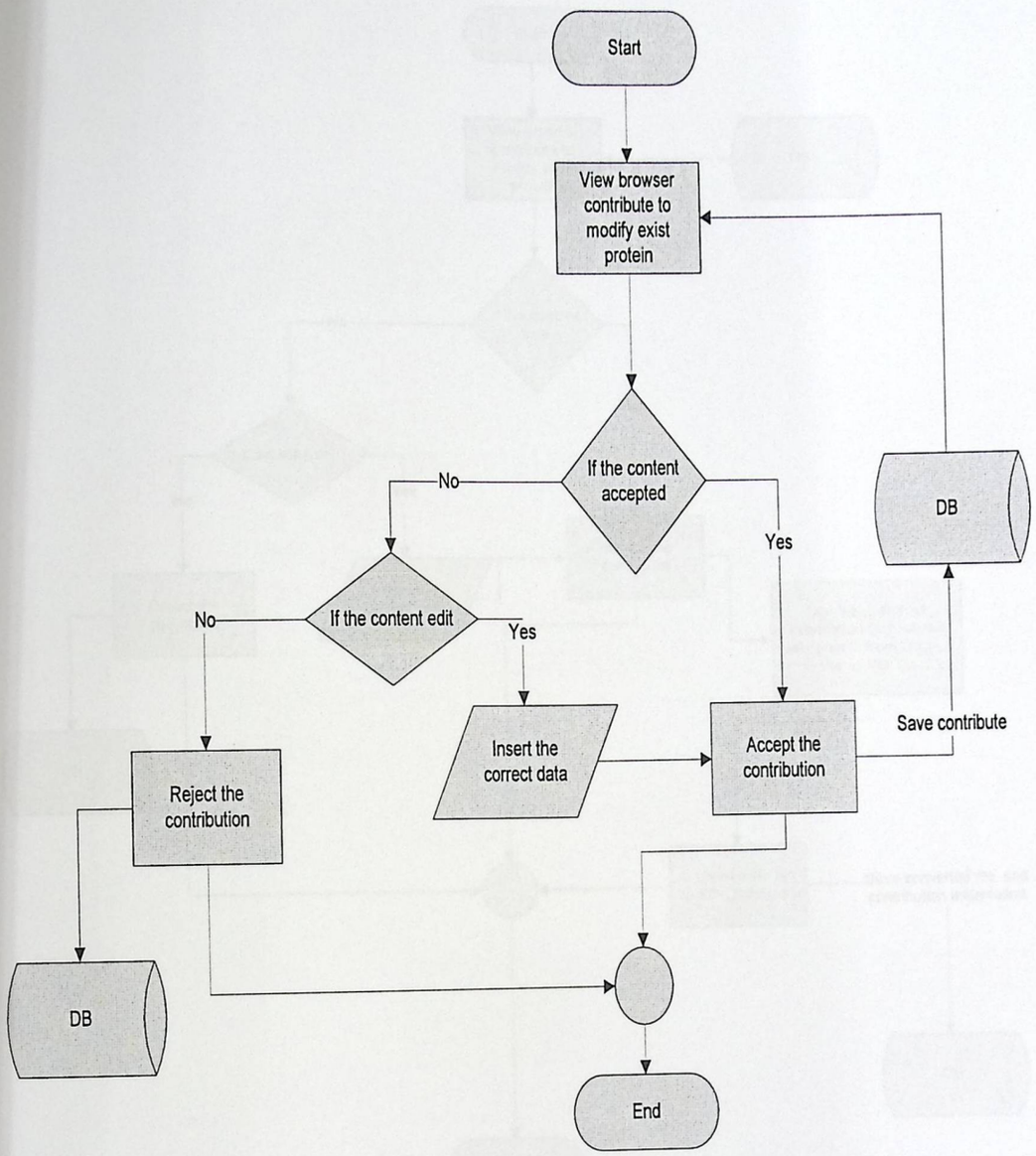


Figure 5. 13 The editing protein contribution process

5.3.7 Add new protein

Administrators can see all contributes, view contribute information, verifying the correctness of the contribute information, and then the administrator decide to confirm the contribution and add it to the knowledgebase or rejected it. This process explained in figure (5.14).

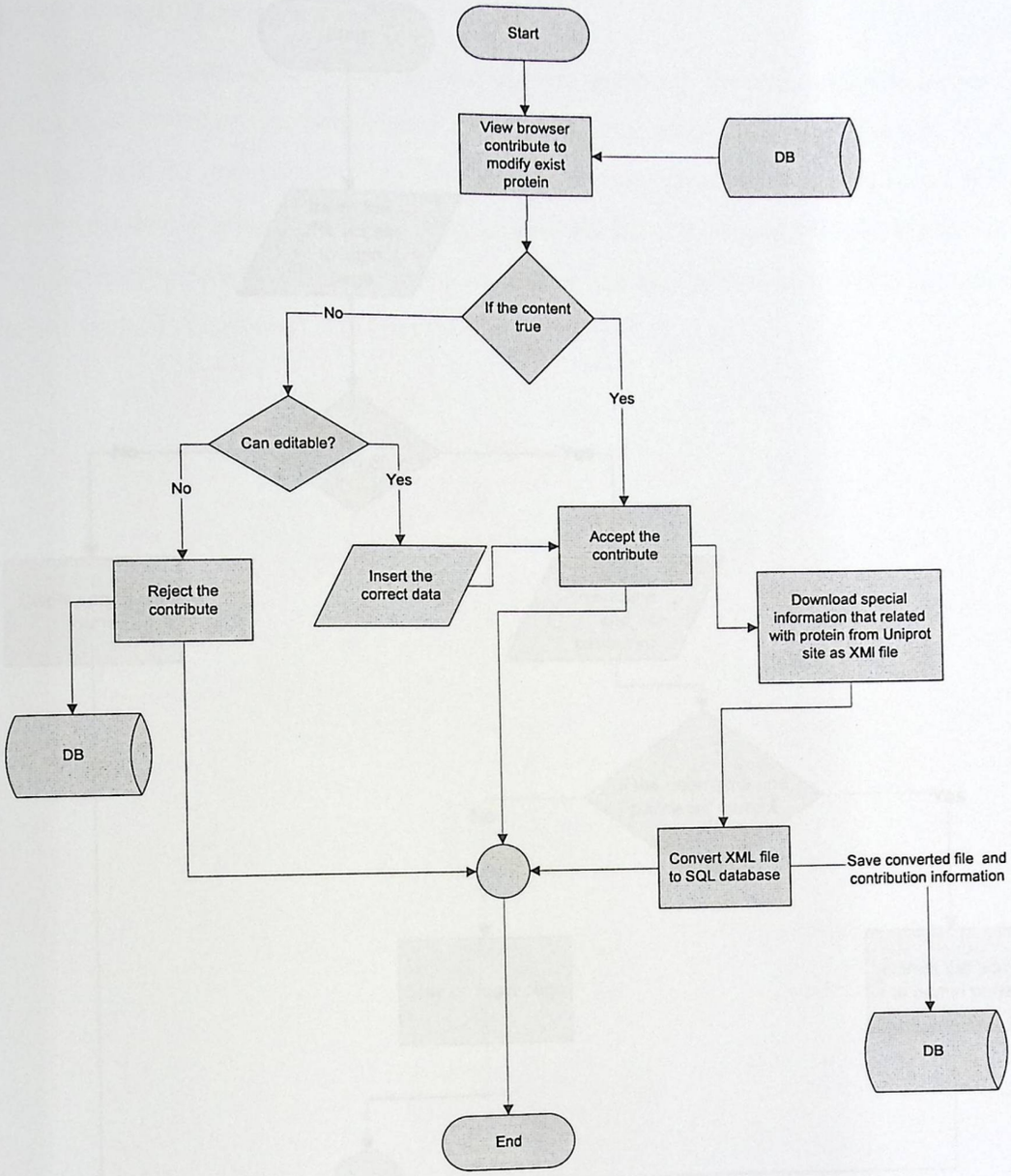


Figure 5. 14 Add new protein process

5.3.8 Login

The system offer hidden secure login page, to let administrators enter to the system and make administration process, the administrator can access hidden login page by enter specific URL in address bar. After that, the administrator inserts his username and password to login. If inserted data are correct, the administrator can access administrator pages, but if username or password does not correct, the system show error message and deny the access. This process explained in figure (5.15).

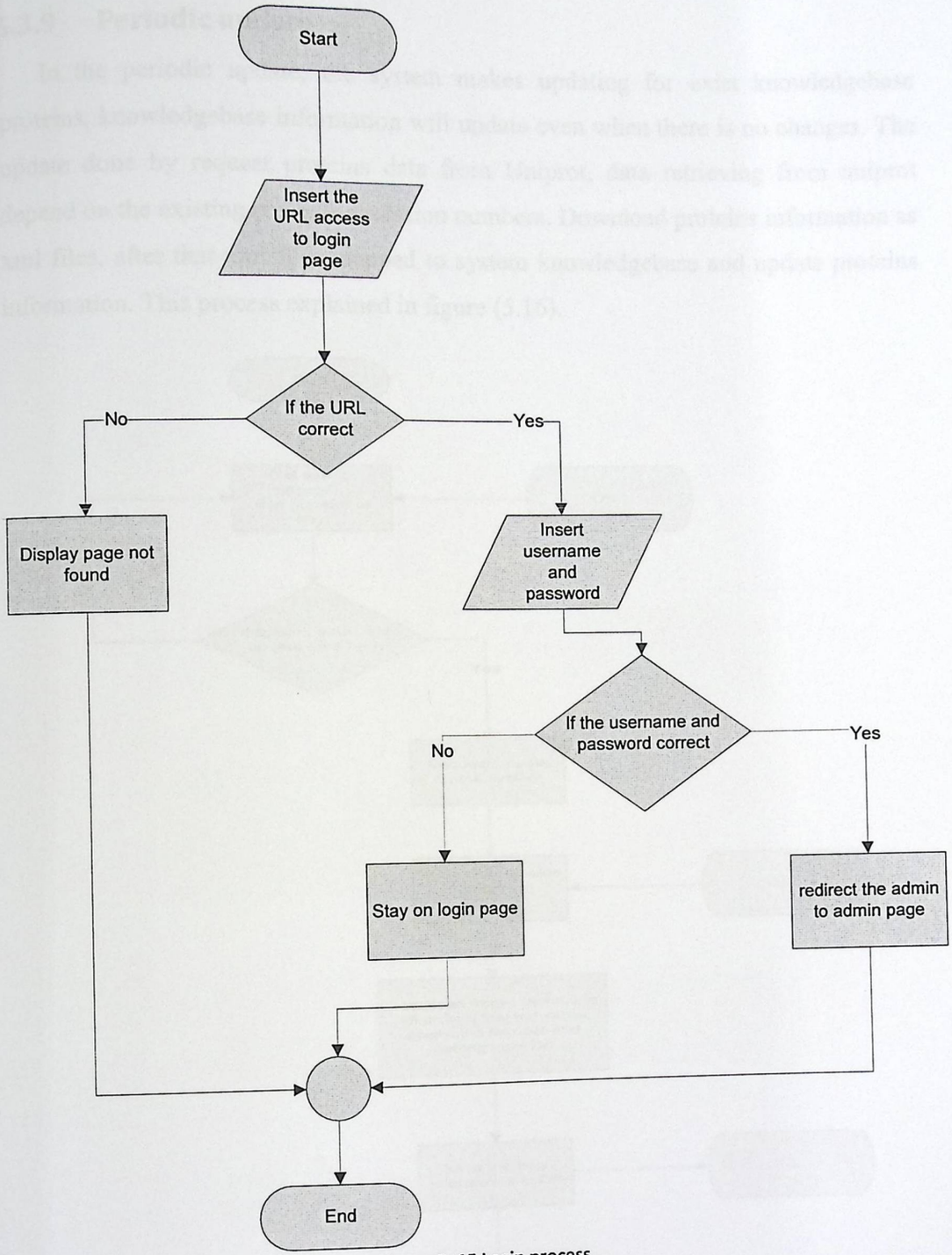


Figure 5. 15 Login process

5.3.9 Periodic update

In the periodic update, the system makes updating for exist knowledgebase proteins, knowledgebase information will update even when there is no changes. The update done by request proteins data from Uniprot, data retrieving from uniprot depend on the existing proteins accession numbers. Download proteins information as xml files, after that xml files mapped to system knowledgebase and update proteins information. This process explained in figure (5.16).

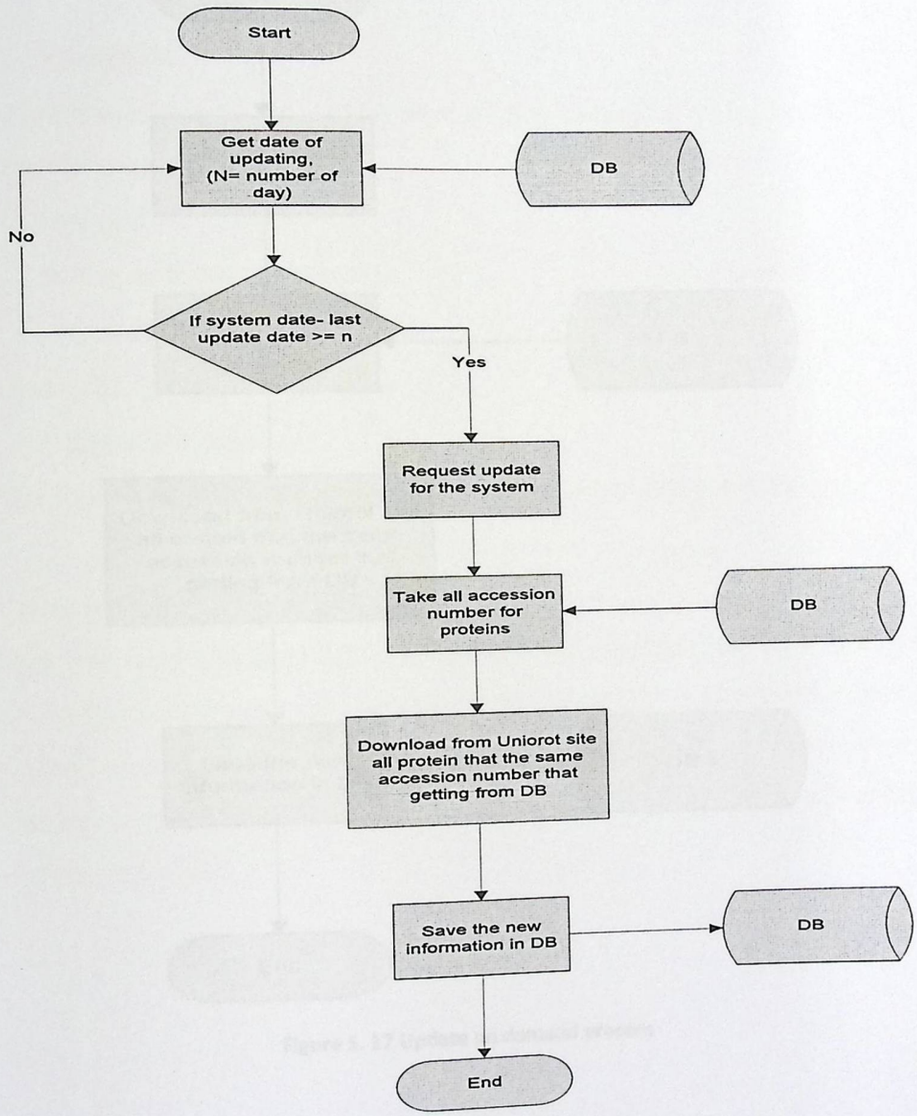


Figure 5. 16 periodic update process

5.3.10 Update on demand

This type of updating occur when the administrator click on search button. The update applying by request for other database (such as Uniprot), the system retrieve all proteins accession number from our database, then download all xml files from external database depend on proteins accession number, and save proteins information in the knowledge base. This process explained in figure (5.17).

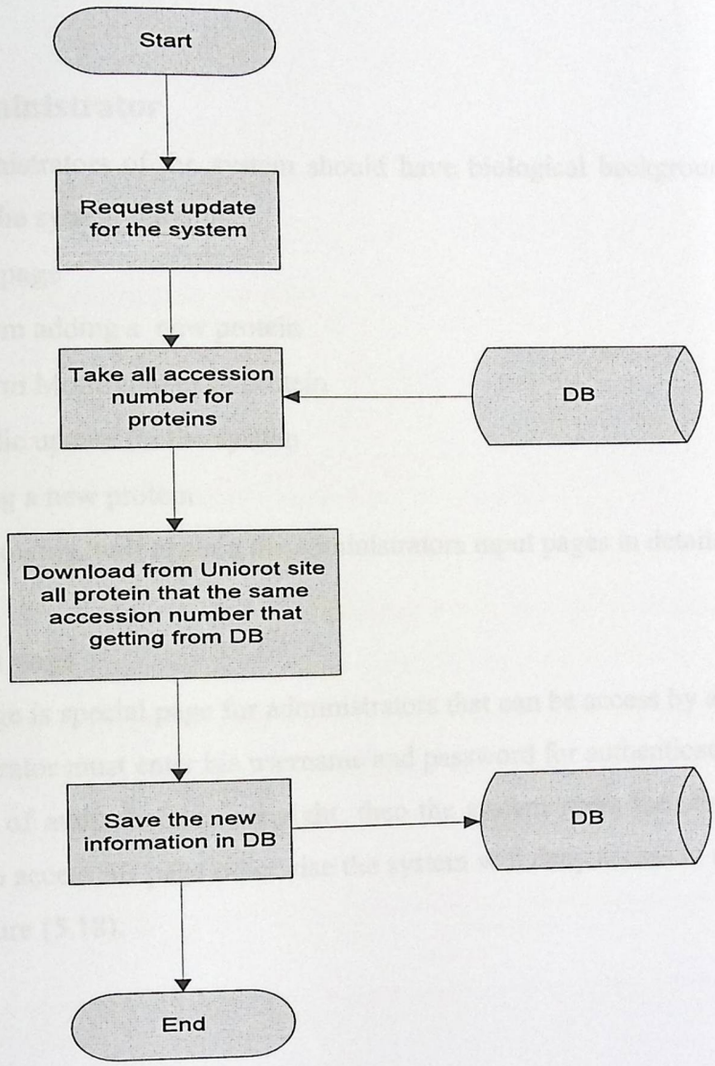


Figure 5. 17 Update on demand process

5.4 Input design

This section describes the screens through which the data can be entered to the system as explicit data or files.

Administrators and browsers can insert data into the system through input screens.

5.4.1 Administrator

The administrators of the system should have biological background. They can interact with the system through:

- Login page
- Confirm adding a new protein
- Confirm Modifying exist protein
- Periodic update for the system
- Adding a new protein

In the following, will explain the administrators input pages in details:

- **Login page**

Login page is special page for administrators that can be access by a special URL. The administrator must enter his username and password for authentication process. If the elements of authentication are right, then the system gives the administrator the permission to access his page otherwise the system will deny access to administrator's page, see figure (5.18).

Item	Type	Size	Source	Note
Username	Nvarchar	20	Administrator	Should be unique
Password	Nvarchar	20	Administrator	Must have complexity

Table 5. 1 Authentication elements

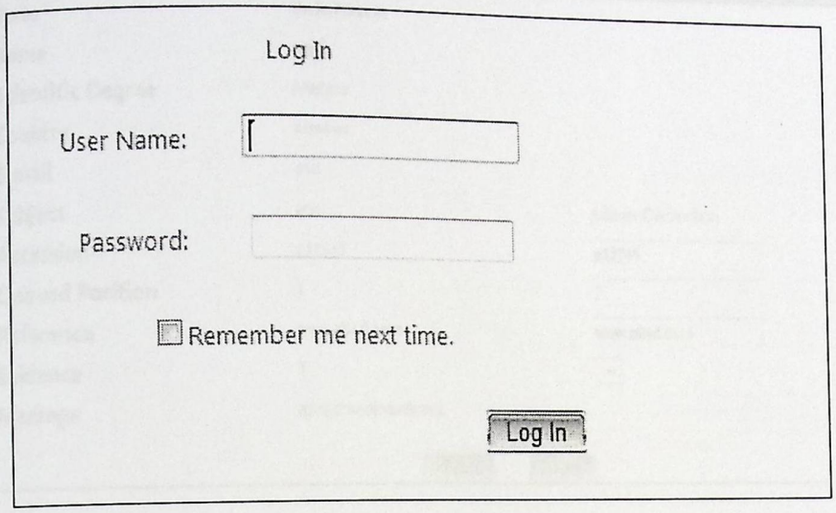


Figure 5. 18 Administrator login page

- **Confirm adding a new protein**

As shown in Figure (5.19), the administrator will be able to confirm the browser contributions for add new protein. If the contribution information is not acceptable, the administrator will make certain changes and then add the contribution information into the database as a new protein, or reject the contribution.

Note: size Max is according to SQL server specifications.

Item	Type	Size	Source
Accession Number	Int	9	Administrator
Cleaved Position	Int	20	Administrator
Reference	URL	MAX	Administrator
Evidence	Int	1	Administrator

Table 5. 2 Main contribution information

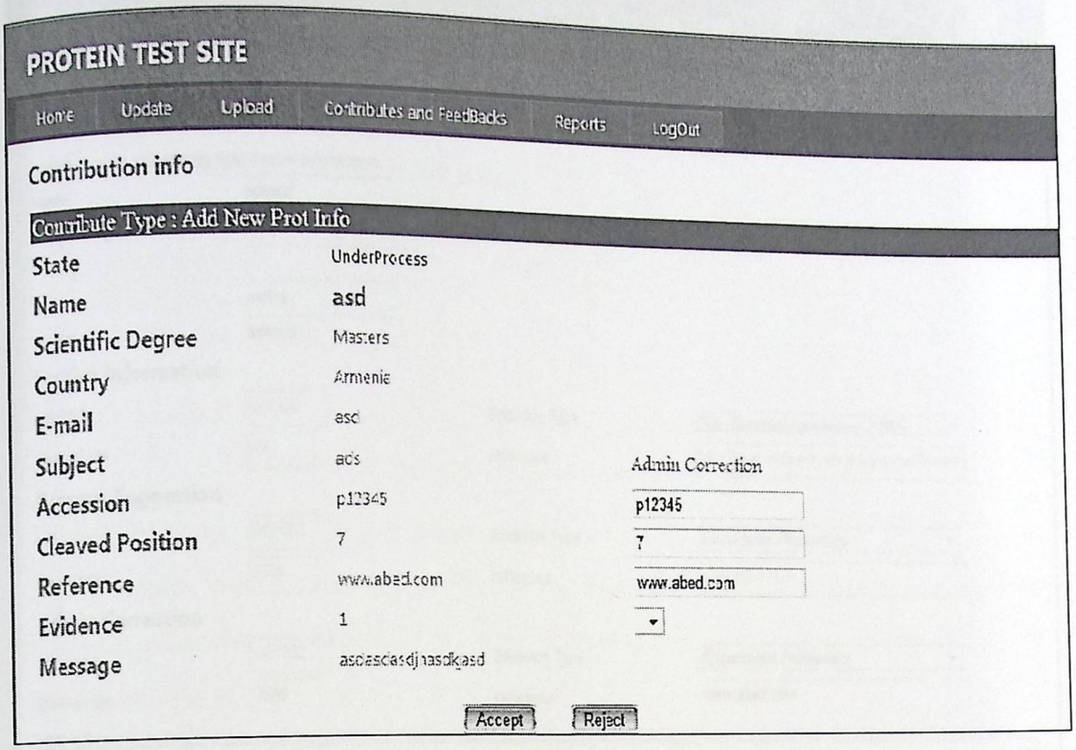


Figure 5. 19 Confirm adding a new protein page

- **Confirm modifying an existing protein**

As shown in Figure (5.20), the administrator will be able to confirm the browser contributions for existing protein. If the contribution information is not acceptable, the administrator can edit some of information in these contribute and then modified the protein information that is in the database, or reject the contribution.

Item	Type	Size	Source
Evidence Type	Text	30	Administrator
Cleaved Position	Int	20	Administrator
Reference	URL	Max	Administrator

Table 5. 3 Main contribution information

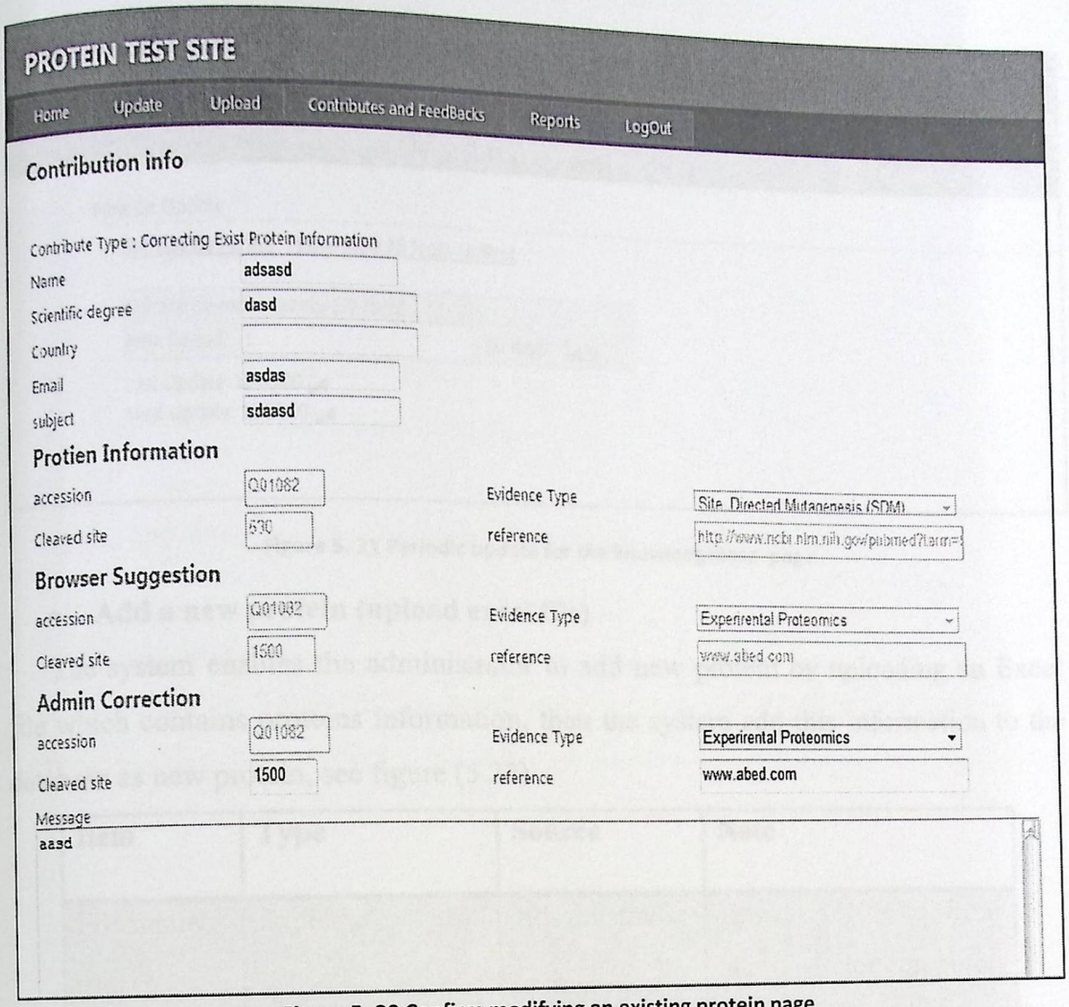


Figure 5. 20 Confirm modifying an existing protein page

- **Periodic update for the system's database**

The administrator can determine the time between two knowledgebase updates, as shown in figure (5.21).

Item	Type	Source
Number of Days	Int	Administrator

Table 5. 4 Update needed information

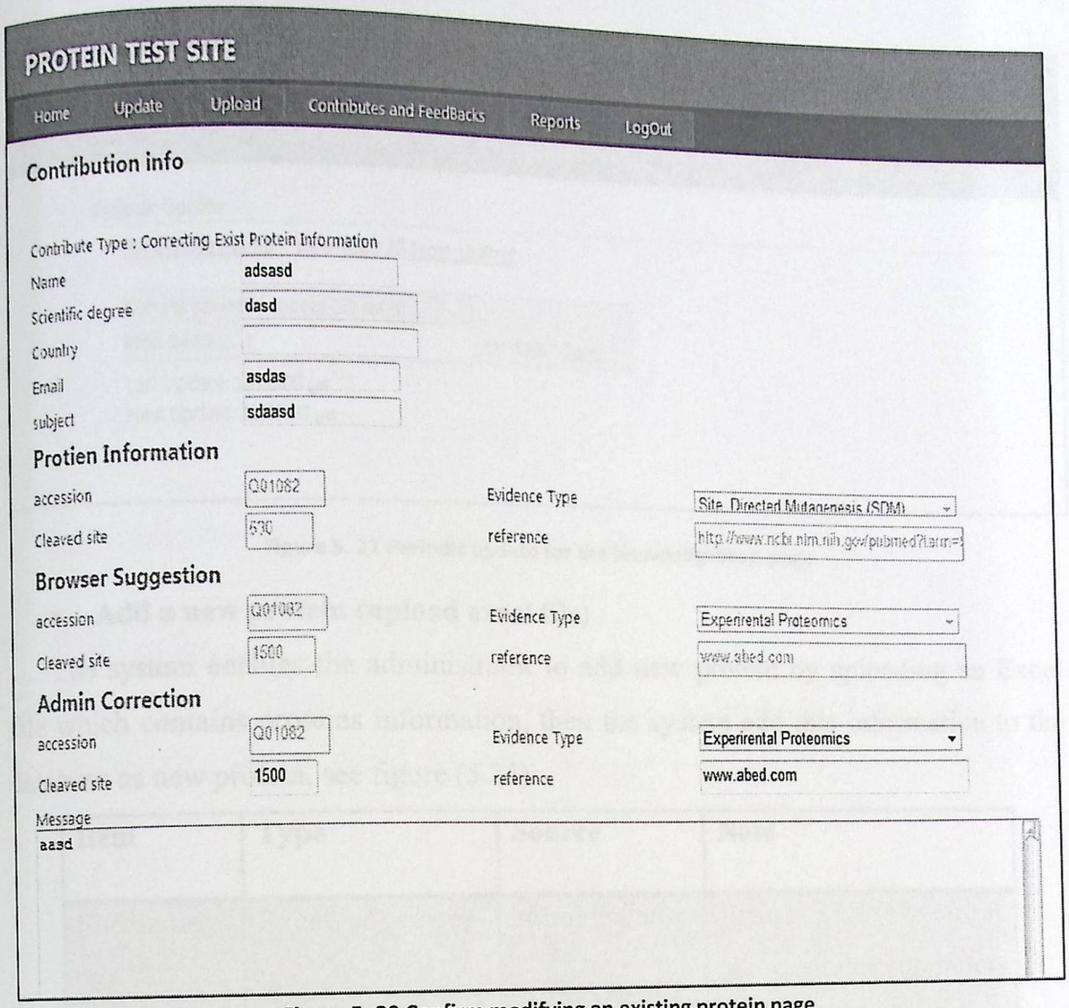


Figure 5. 20 Confirm modifying an existing protein page

- **Periodic update for the system's database**

The administrator can determine the time between two knowledgebase updates, as shown in figure (5.21).

Item	Type	Source
Number of Days	Int	Administrator

Table 5. 4 Update needed information

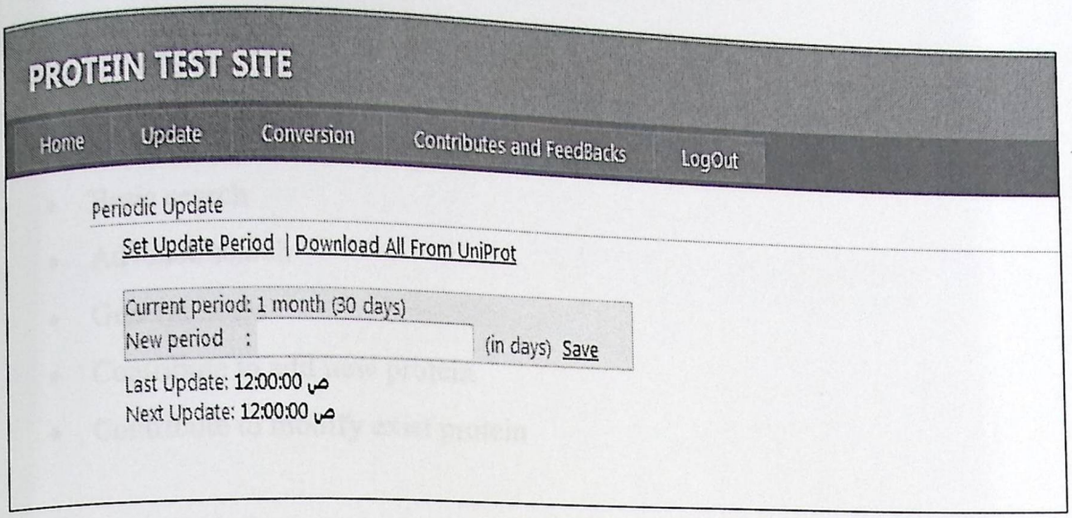


Figure 5. 21 Periodic update for the knowledgebase page

- **Add a new protein (upload excel file)**

The system enables the administrator to add new protein by uploading an Excel file which contains proteins information, then the system add this information to the database as new protein, see figure (5.22).

Item	Type	Source	Note
File name	Excel	Administrator	Brows from administrator computer

Table 5. 5 Add a new protein by upload excel file

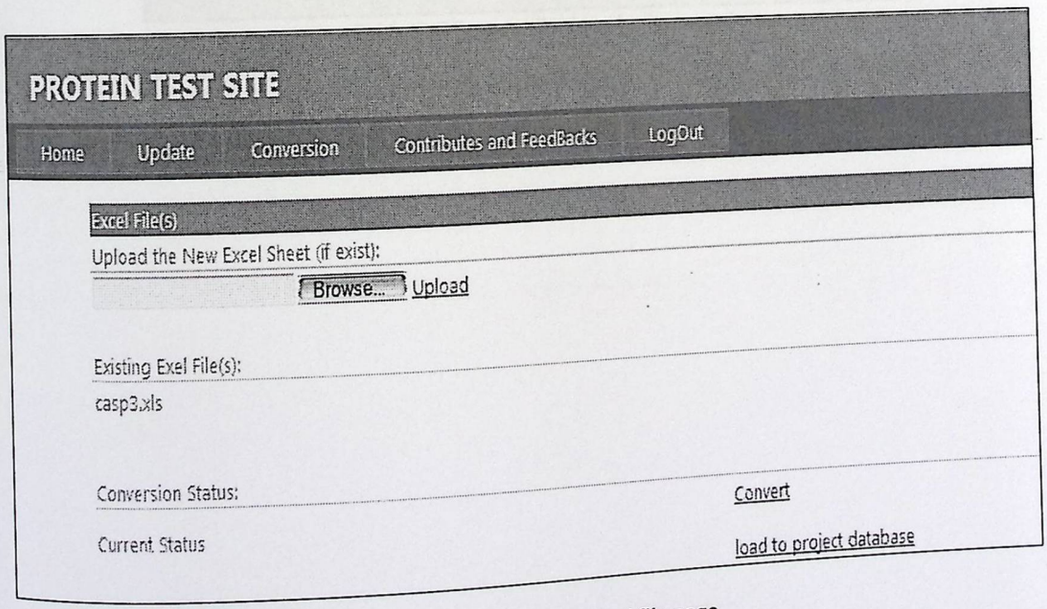


Figure 5. 22 Upload excel file page

5.4.2 Browsers

The browsers can make many the following processes through the input pages as the following:

- Basic search
- Advance search
- General notes
- Contribute to add new protein
- Contribute to modify exist protein

In the following, will be explaining the browsers input pages in details.

- **Basic search:**

As shown in figure (5.23), home page contains basic search mechanism, the system enables the browsers to enter any keyword to search in the whole knowledgebase.

Item	Type	Source
Keyword	Text	Browser

Table 5. 6 Search all tables for key word

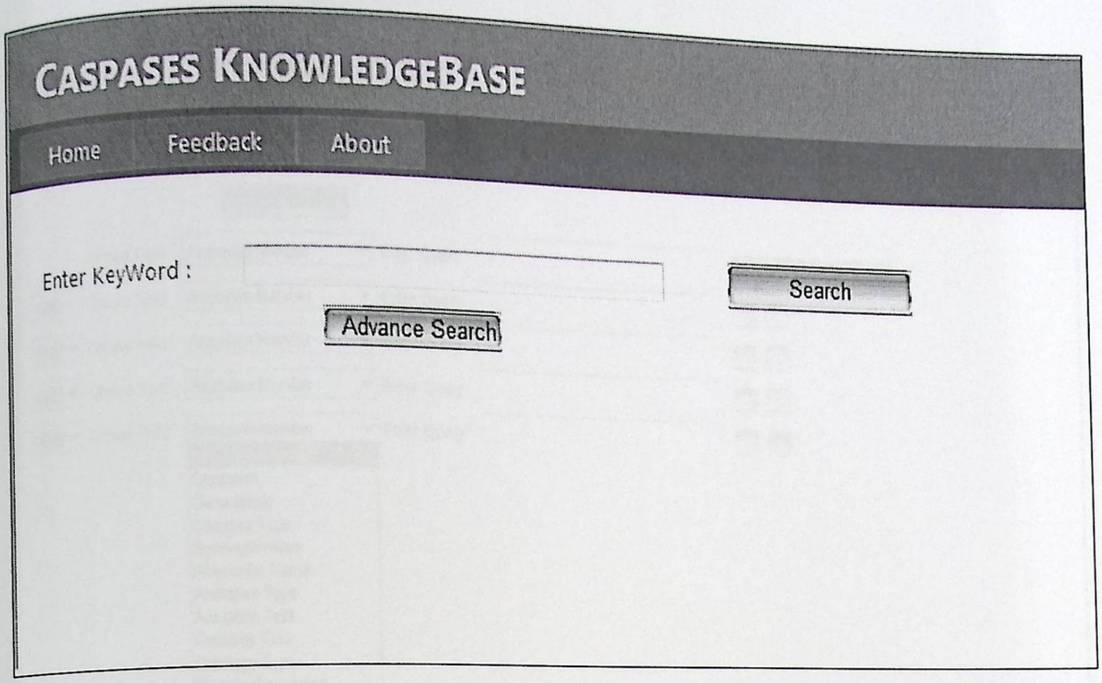


Figure 5. 23 Basic search page

- **Advance search:**

The system enables the browsers to search in the knowledgebase based on selecting specific protein properties and searching inside this property for specific information. The system also enables the browsers to make combination search about more than one property at the same time and use logical operation (and, or and not) to combine between properties, see figure (5.24).

Item	Type	Source
Filed	Text	Browser
Logical operation	Text	Browser

Table 5. 7 Input information

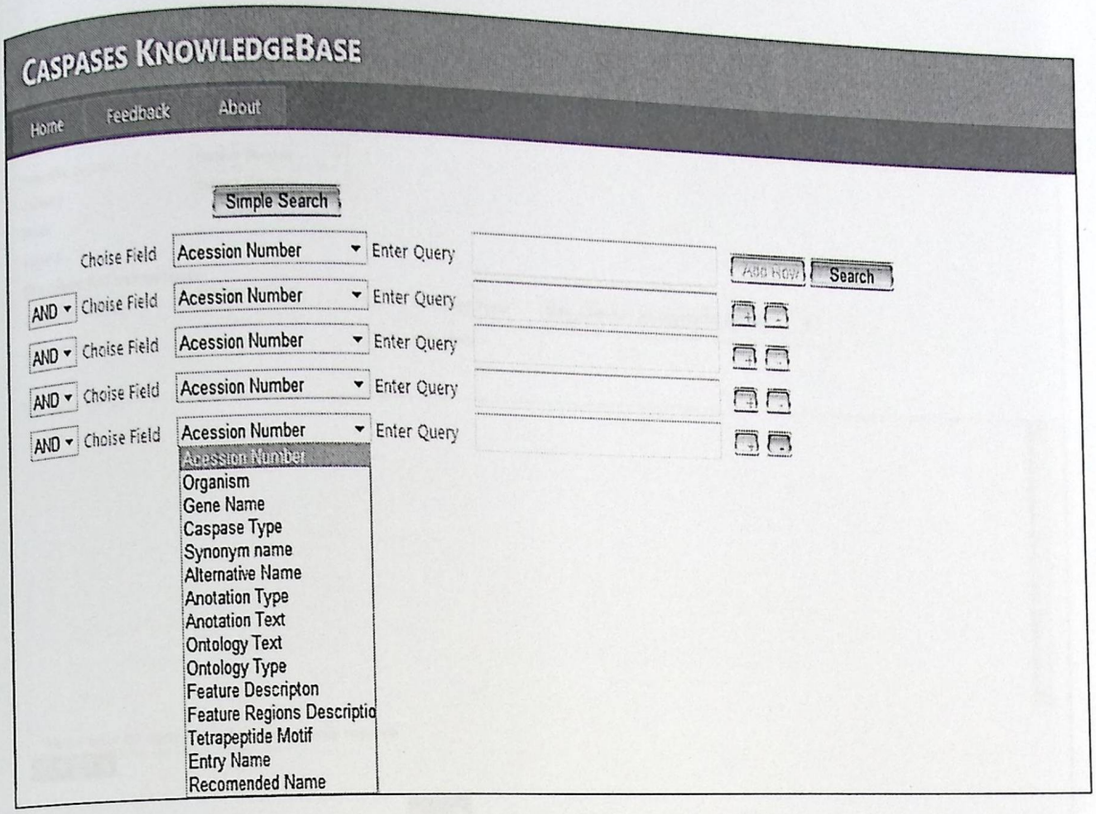


Figure 5. 24 Advance search page

- **General notes:**

The system offers a form to enable the browser to communicate with the administrators of the system by sending a note about any error or feature in the system. See figure (5.25).

Item	Type	Size	Source
Name	Text	30	Browser
Significant Degree	Text	30	Browser
Country	Text	12	Browser
Email	Text	30	Browser
Message	Text	200	Browser
Computing result	Int	2	System

Table 5. 8 Feedback information

Evidence Type	Text	1	Browser
References	URL	MAX	Browser
Computing result	Int	2	System

Table 5. 9 Contribution information

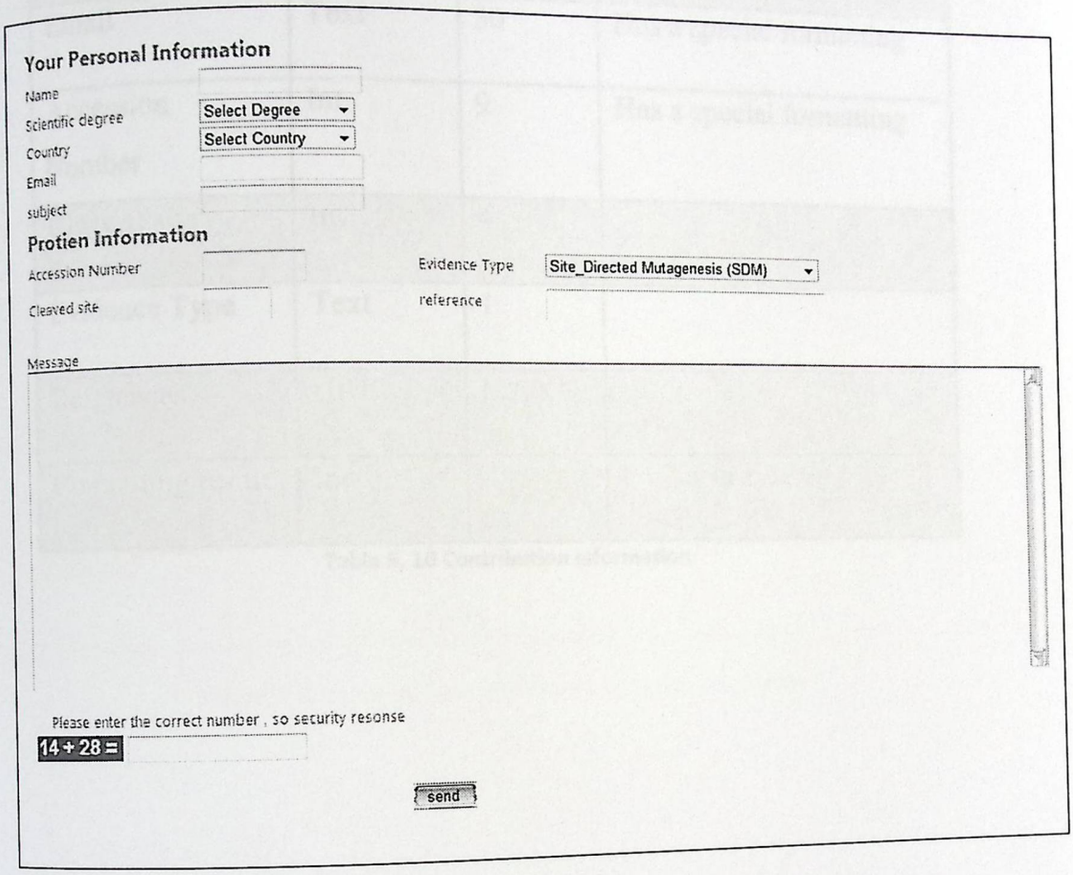


Figure 5. 26 Contribution to add new protein information

- **Contribute to modify an existing protein**

The system offers a form to enable the browsers to modify existing protein information and send it to the administrators to be accepted and modify the protein information. Figure (5.27) shows the modified protein information form.

Item	Type	Size	Source
Name	Text	30	Browser
Significant Degree	Text	30	Browser
Country	Text	12	Browser
Email	Text	30	Has a special formatting
Accession Number	Int	9	Has a special formatting
Cleaved site	Int	4	
Evidence Type	Text	1	
References	URL	MAX	
Computing result	Int	4	For security

Table 5. 10 Contribution information

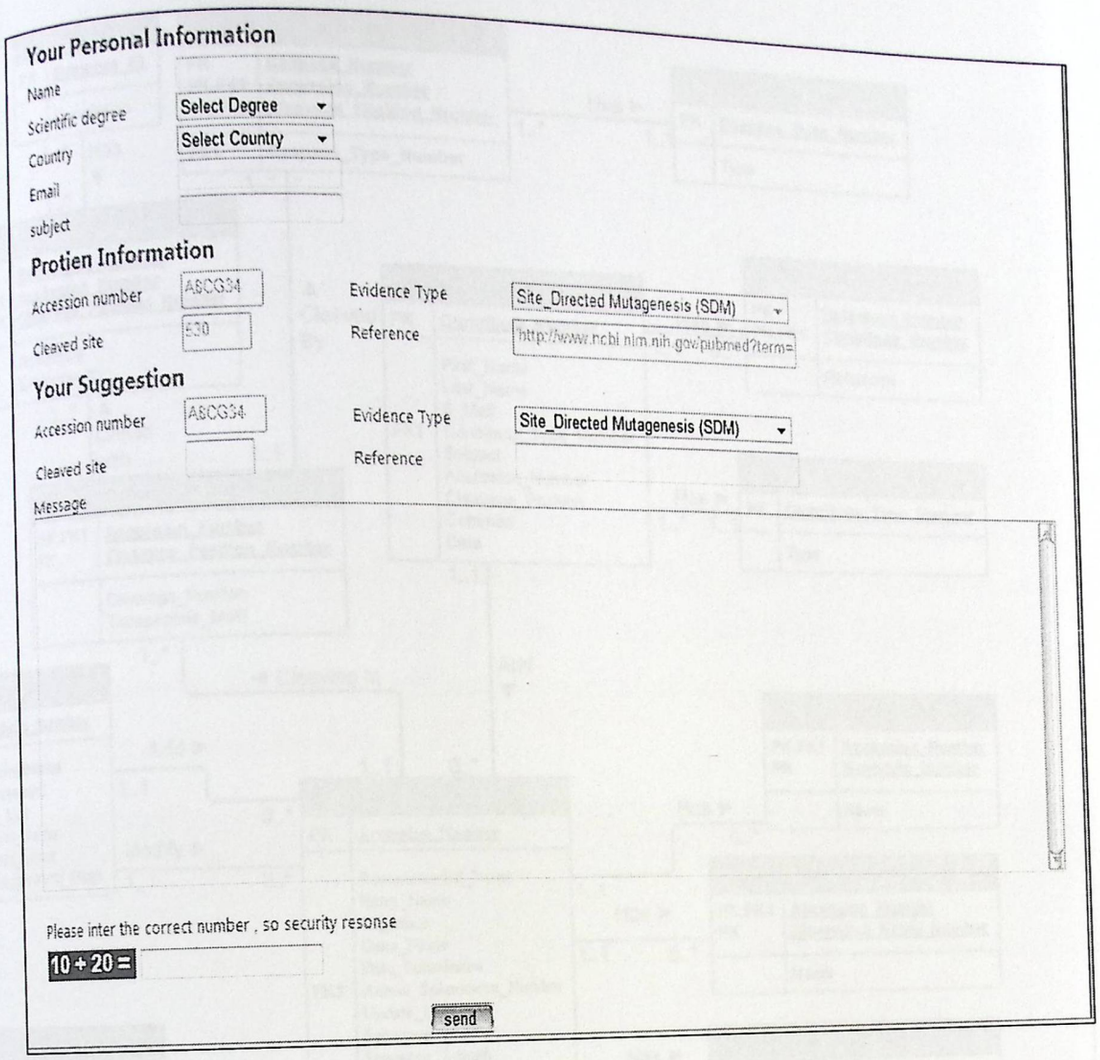


Figure 5. 27 Contribution to modify exist protein information

5.5 Database design

This section describes the system database in details.

5.5.1 ER Model

Figure (5.28) shows the ER model that describes all tables in the system database and the relationship between them.

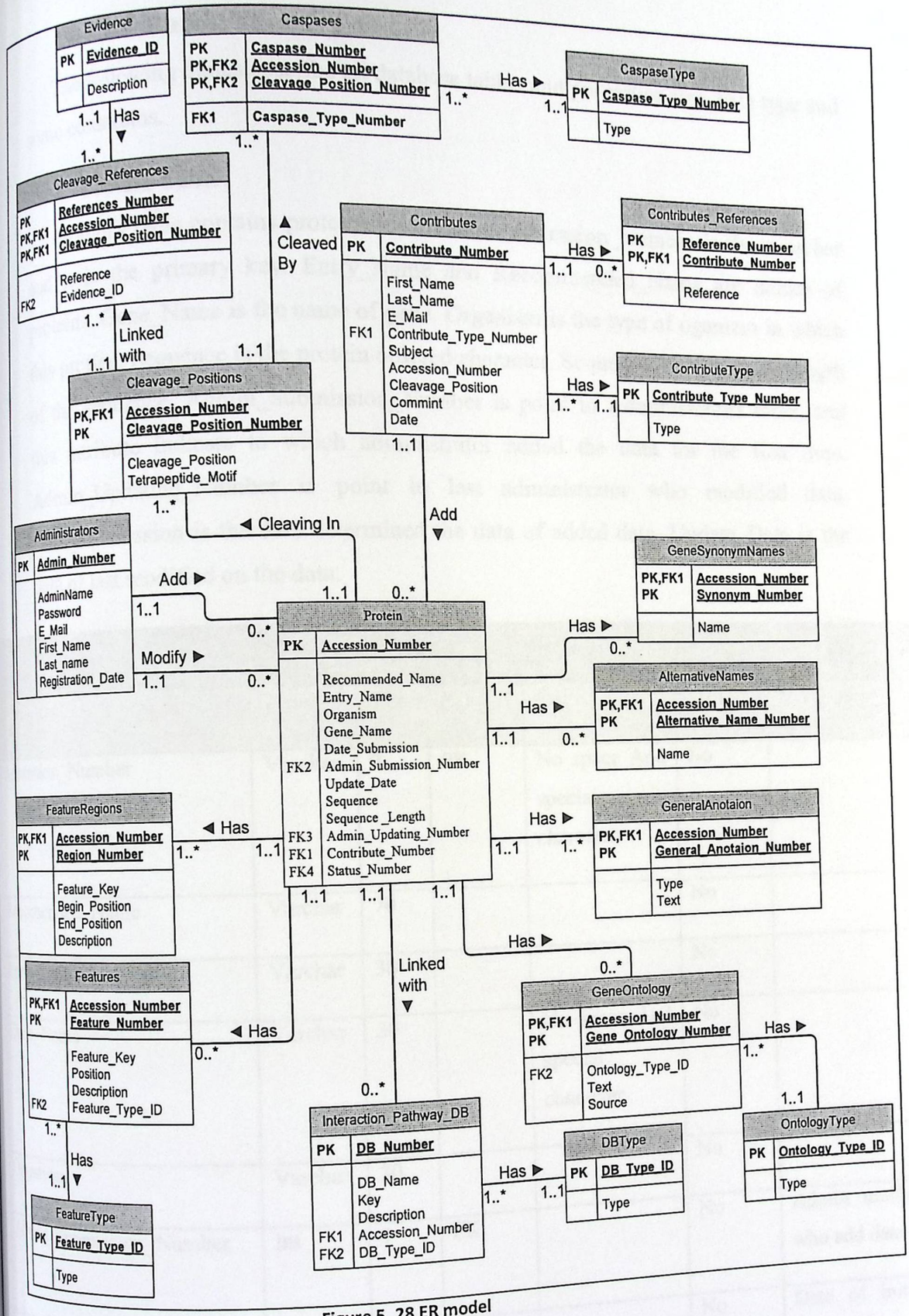


Figure 5. 28 ER model

5.5.2 Data dictionary

The following tables describe all database tables and properties size, data type and some constrains.

1- Protein Table:

Protein's table contains proteins information. Accession_Number is the nvarchar and it is the primary key. Entry_Name and Recommended_Name are names of proteins. Gene_Name is the name of gene. Organism is the type of oganizm in which this protien. Sequence is the protein ordered character. Sequence_Length is the length of the sequence. Admin_Submission_Number is point to Administrators table, and this attribute indicate to which administrator added the data for the first time. Admin_Updating_Number is point to last administrator who modified data. Date_Submission is the data determined the data of added data. Update_Date is the data of last modified on the data.

Protein table

Attribute	*Type*	*Size*	*Key*	*Constraint*	*Null?*	*Description*
Accession_Number	Varchar	9	PK	No space And special character	No	
Recomended_Name	Varchar	70			No	
Entry_Name	Varchar	30			No	
Gene_Name	Varchar	30		No space And special character	No	
Organism	Varchar	30			No	
Admin_Submission_Number	Int	1	FK		No	Admin number who add data
Date_Submission	Date	8			No	Date of initial submission

Contribute_Number	Int	6	FK		No	
Update_Date	Date	8			Yes	Date of update
Admin_Updating_Number	Int	1	FK		Yes	Admin Name how updating data
Sequence	Varchar	Max			No	
Sequence_Length	Int	5			No	Length of sequence

Table 5. 11 Protein table

2- Administrators Table:

A table that stores administrators credentials in order to authenticate and authorize them when they attempt to log in to the system. Admin_Number which is a numeric value indicates the uniquely identifier for each administrator. Admin_Name is a unique name for each administrator in the system. Password is the secret correction of character and numbers that administrator create and maintain to be recognized through it besides the Admin_Name. Regester_Date is the date in which the user had been register in the system. E_mail is the E_Mail for administrator. First_Name is the first name for administrator. Last_Name is the last name of administrator.

Attribute	*Type*	*Size*	*Key*	*Constraint*	*Null?*	*Description*
Admin_Number	Int	1	PK		No	
Admin_Name	Varchar	10		Unique, more than 6 Char	No	
Password	Varchar	20		More than 6 Char	No	
E_mail	Varchar	50		E_Mail	No	
First_Name	Varchar	10			No	

Last_Name	Varchar	10			No	
Regester_Date	Date	8			No	

Table 5. 12 Administrators table

3- GeneSynonymNames:

GeneSynonymNames table has the gene synonym names of each proteins. Accession_Number point to the Protein table and this property declare to witch protein this name follows. Synonym_Number is the unique number of the gene synonym name. Name is the gene synonym name. Both Accession_Number and Synonym_Number together are the primary key.

Attribute	*Type*	*Size*	*Key*	*Constraint*	*Null?*	*Description*
Accession_Number	Varchar	7	PK, FK		No	
Synonym_Number	Int	4	PK		No	
Name	Varchar	30			No	

Table 5. 13 Gene synonym names

4- AlternativeNames:

This table has the alternative names of the proteins. Accession_Number point to Protein's table. Alternative_Name_Number is the unique number of the alternative name. Name is the alternative proteins name. Both Accession_Number and Alternative_Name_Number together are the primary key.

Attribute	Type	Size	Key	Constraint	Null?	Description
Accession_Number	Varchar	7	PK,FK		No	
Alternative_Name_Number	Int	4	PK		No	
Name	Varchar	70			No	

Table 5. 14 Alternative names

5- GeneralAnnotation:

This table provides any useful information about the protein, mostly biological knowledge. Accession_Number point to Protein's table. General_Anotation_Number is the unique number of the comment. Type determines the type of comments. Text is the comment. Both Accession_Number and General_Anotation_Number together are the primary key.

Attribute	Type	Size	Key	Constraint	Null?	Description
Accession_Number	Varchar	6	PK,FK		No	
General_Anotation_Number	Int	4	PK		No	
Type	Varchar	20			No	
Text	Varchar	Max			No	

Table 5. 15 General anotaion

6- GeneOntology:

Accession_Number is point to the protein number. Gene_Ontology_Number is the unique number for ontology. Ontology_Type_Number is the point to the OntologyType table, and that identify the type of ontology (Biological process, Cellular component and Molecular function). Text is the description of the gene ontology. Source is the link (URL) declare from where the information taken.

Attribute	Type	Size	Key	Constraint	Null?	Description
Accession_Number	Varchar	7	PK,FK		No	
Gene_Ontology_Number	Int	4	PK		No	
Ontology_Type_Number	Int	1	FK		No	
Text	Varchar	200			No	
Source	Varchar	20		URL	No	

Table 5. 16 Gene ontology

7- Ontology Type:

OntologyType table has the types of gene ontology. Ontology_Type_Number is unique value of the ontology type, and it's the primary key. Type is the text value of the ontology type.

Attribute	Type	Size	Key	Constraint	Null?	Description
Ontology_Type_Number	Int	1	PK		No	
Type	Varchar	50			No	

Table 5. 17 Ontology type

8- Cleavage Positions:

This table contains the cleavage positions of the proteins. Accession_Number point to the Protein table. Cleavage_Position_Number is the unique number of the cleavage position. Cleavage_Posistion is the numeric value determined the position of the cleavage protein. Tetrapeptide_Motif is the four characters before the cleavage position. Both Accession_Number and Cleavage_Position_Number together are the primary key.

Attribute	Type	Size	Key	Constraint	Null?	Description
Accession_Number	Varchar	7	PK,FK		No	
Cleavage_Position_Number	Int	6	PK		No	
Cleavage_Posistion	Int	5			No	
Tetrapeptide_Motif	Varchar	4		Only characters and no space	No	

Table 5. 18 Cleavage positions

9- Cleavage References:

Cleavage References table contains the references of cleavage position papers. Reference_Number is unique reference number. Reference is the URL of scientific paper reference. Evidence_ID is point to the Evidence table. Accession_Number and Cleavage_Position_Number are point to the CleavagePositions table.

Reference_Number, Accession_Number and Cleavage_Position_Number are the primary key.

Attribute	Type	Size	Key	Constraint	Null?	Description
Reference_Number	Int	6	PK		No	
Accession_Number	Varchar	7	PK,FK		No	
Cleavage_Position_Number	Int	6	PK,FK		No	
Reference	Varchar	Max		URL	No	
Evidence_ID	Int	1	FK		No	

Table 5. 19 Cleavage references

10- Evidence:

This table describes the evidence Type of protein cleavage position. Evidence_ID is the unique number of the evidence, and this attribute is the primary key. Description is the description of the evidence.

Attribute	Type	Size	Key	Constraint	Null?	Description
Evidence_ID	Int	1	PK		No	
Description	Varcahr	50			No	

Table 5. 20 Evidence

11- Feature Regions:

FeatureRegions table has the description of the feature regions of the proteins. Accession_Number point to the protein table. Region_Number is the uniqe number of the feature. Feature_Key is the type of the feature region. Begin_Position is the begin position of the domain in the squence. End_Position is the end position of the domain in the sequence. Description is the description of protein's feature. Both Accession_Number and Region_Number together are the primary key.

Attribute	Type	Size	Key	Constraint	Null?	Description
Accession_Number	Varchar	7	PK,FK		No	
Region_Number	Int	6	PK		No	
Feature_Key	Varchar	50			No	
Begin_Position	Int	6			No	
End_Position	Int	6			No	
Description	Varchar	100			No	

Table 5. 21 Feature regions

12- Features:

This table contains the other feature. Accession_Number is point to the protein table. Feature_Number is the unique number of the feature. Feature_Key is the type of the feature. Position is the position of the feature in protein's sequence. Description is the description of the feature. Feature_Type_ID is point for the number of the main feature type. Both Accession_Number and Feature_Number together are the primary key.

Attribute	*Type*	*Size*	*Key*	*Constraint*	*Null?*	*Description*
Accession_Number	Varchar	7	PK,FK		No	
Feature_Number	Int	6	PK		No	
Feature_Key	Varchar	50			No	
Position	Int	6			No	
Description	Varchar	100			No	
Feature_Type_ID	Int	1	FK		No	

Table 5. 22 Feature

13- Feature Type:

FeatureType table has the feature types of the proteins. Feature_Type_ID is the unique number of the feature type, and it's the primary key. Type is the text value of the type.

Attribute	*Type*	*Size*	*Key*	*Constraint*	*Null?*	*Description*
Feature_Type_ID	Int	1	PK		No	
Type	Varchar	50			No	

Table 5. 23 Feature type

16- Contributes:

This table has the contributions of the browsers. Contribute_Number is unique number of each contribute, and it is the primary key. Name is the name of the contribute sender. E_mail is E_mail for the contributer. Contribute_Type_Number is the point to the ContributeType table, and determined the type of contribute (e.g. contribute add new protein). Subject is the title of the contribution. Accession_Number is the number of protein (is required when the contribution is add or modify protein). Cleaved_Position is the position of cleavage protein (is required if the contribution is add or modify protein). Comment is the message from the contributor about his/her contribution. Date is the date for the adding contribution.

Attribute	*Type*	*Size*	*Key*	*Constraint*	*Null?*	*Description*
Contribute_Number	Int	6	PK		No	
Name	Varchar	20			No	
Scientific_Degree	Varchar	20			No	
E_Mail	Varcahr	50		E_Mail	No	
Contribute_Type_Number	Int	1	FK		No	
Subject	Varchar	50			No	
Accession_Number	Varchar	7			No	
Cleaved_Position	Int	5			Yes	
Comment	Varchar	Max			Yes	
Date	Date	8			No	
State	Varchar	50			No	
EvidenceID	Int	5	FK		Yes	
CorrectionRow	Int	5			Yes	
Country	Varchar	50			Yes(No)	

Table 5. 26 Contributes

19- Interaction_Pathway_DB:

This table contains information about the external databases and websites, it is has URL for interaction and pathway for the proteins. Accession_Number is point to the proteins table, which determining the databases that has the protein interaction and pathway information. DB_Number is the unique number for each database. DB_Name is the name of the database. Key is the string value has the key to link to the external database. Description is the string value has description the data in the external database. DB_Type_ID is point to the DBType table, and determining the type of this external database. Both DB_Number and Accession_Number together are the primary key.

Attribute	*Type*	*Size*	*Key*	*Constraint*	*Null?*	*Description*
DB_Number	Int	6	PK		No	
DB_Name	Varchar	50			No	
Key	Varchar	50			No	
Description	Varchar	Max			Yes	
DB_Type_ID	Int	1	FK		No	
Accession_Number	Varchar	7	FK		No	

Table 5. 29 Interaction pathway DB

20- External_DB_Names:

External_DB_Names table has the names of the external databases. DB_Name_ID is the unique number for each database, and it is the primary key. Name is the string value for the name of the database.

- Administrator navigation

The following figure (5.30) shows administrator navigation:

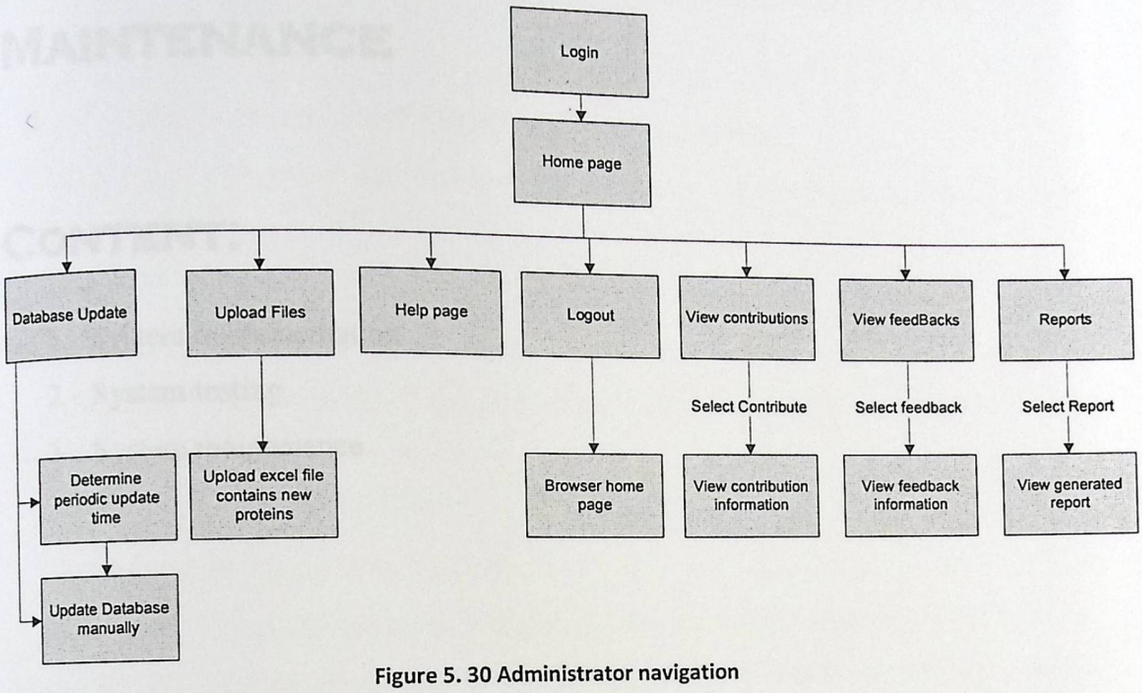


Figure 5. 30 Administrator navigation

CHAPTER 6

SYSTEM IMPLEMENTATION, TESTING AND MAINTENANCE

CONTENT:

1. System implementation
2. System testing
3. System maintenance

6.1 System Implementation

6.1.1 Introduction

This section will discuss the implementation phase that includes development environment requirements, physical equipment, tools, and required environment to implement the system.

6.1.2 Development Environment Requirements

This phase describes the software environment requirements to develop the project.

1- Microsoft Windows win7

Windows 7 this operating system contain requirement that is needed to develope this project.

2- Microsoft Visual Studio 2010

Microsoft Visual Studio is an Integrated Development Environment (IDE) from Microsoft, it's used as development environment, help in find the errors and correct them, design system interfaces, activate the Web.NET and help in execute instructions. Also, it is supports different programming languages by language service which allows for editor and debugger to support many programming language. These languages are C/C++, C#, F#, J#, VB.NET and others. Also, it include a built-in tools as Graphical User Interface (GUI) application, class designer, database schema designer, and web designer, Figure (6.1) shows this environment.

To develop the project we use ASP.NET technology with VB.NET language, which are included in visual studio environment.

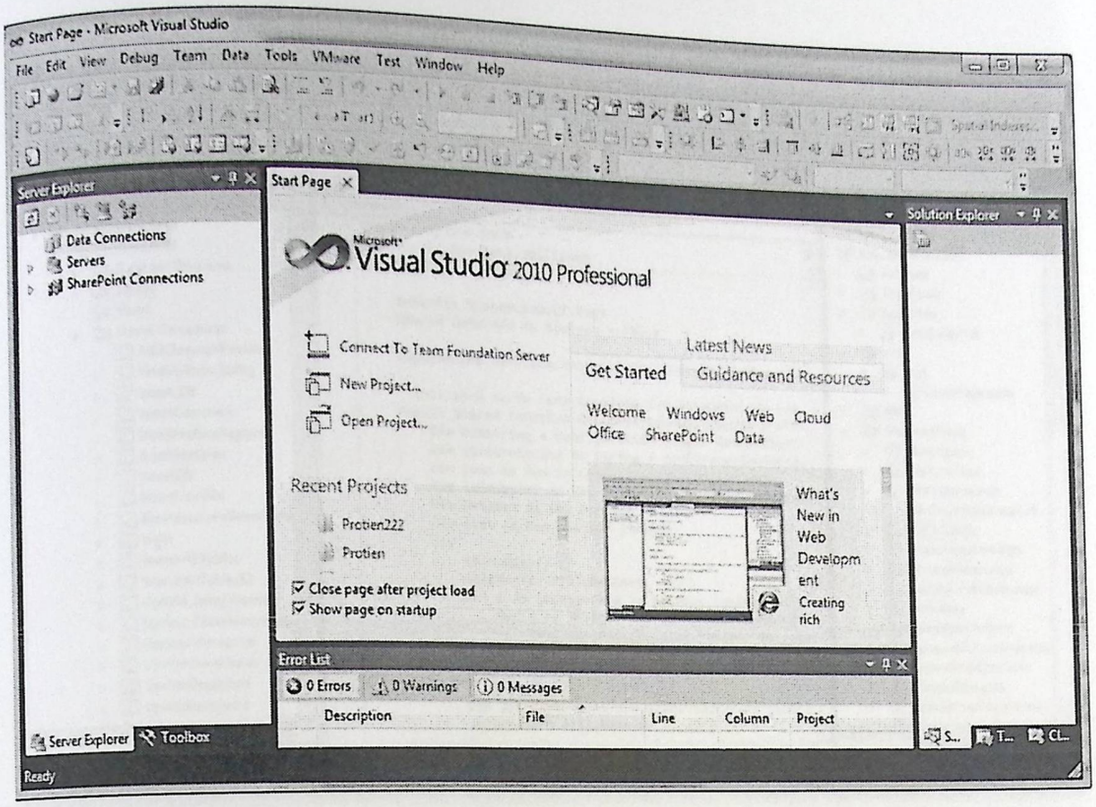


Figure 6. 1 Microso Visual Studio 2010

3- Microsoft ASP.NET

ASP.NET is a unified Web development model offered by Microsoft that includes all service necessary to create enterprise-class Web applications. It contain programs like Visual Studio.NET and Visual Web Developer that allow for developers to develop dynamic websites by a visual interface that write by code and scripts. ASP.NET is part of the .NET Framework. The developer can code the application in any language compatible with the common language runtime (CLR) such as Microsoft Visual Basic and C# that enable the developer to develop ASP.NET applications, Figure (6.2) shows screenshot for asp.net program.

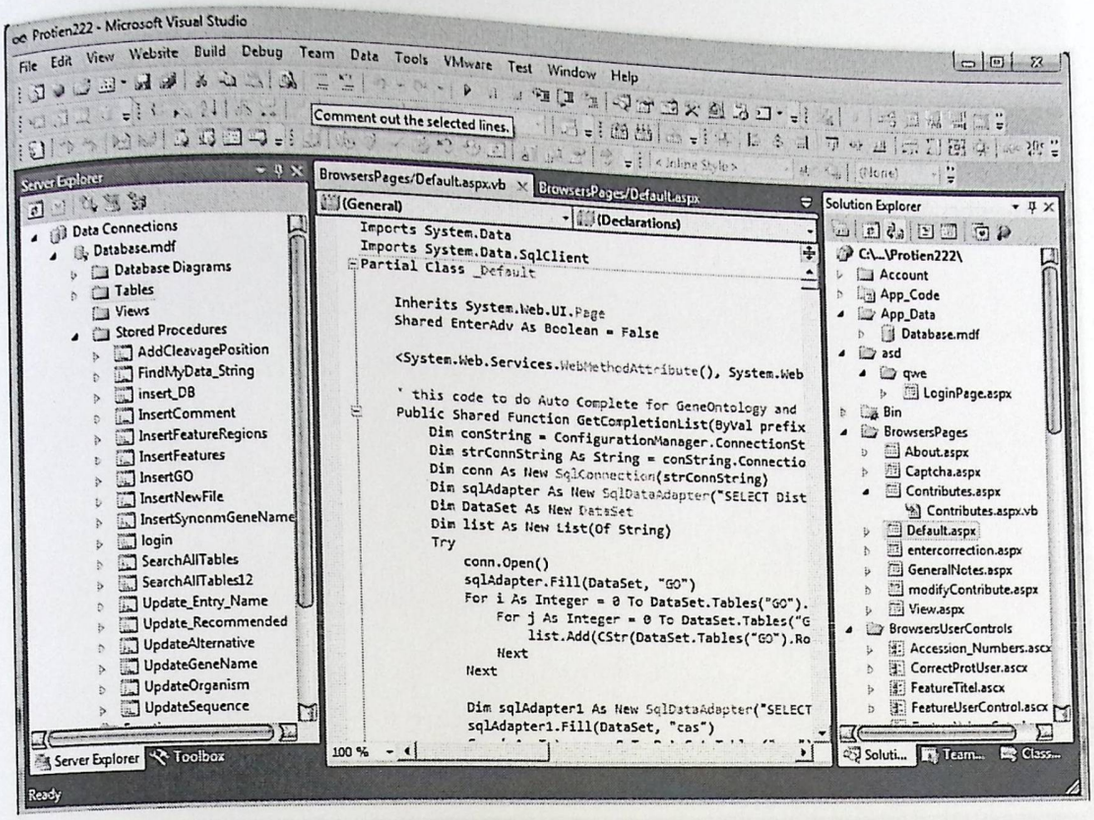


Figure6. 2: Microsoft ASP.NET Framework

4- Microsoft SQL Server 2008

In general, SQL Server provides a complete and easy platform for databases including database creation, controlling, and management. It is support three type of authentication: SQL Server authentication, Windows Authentication, and Mixed mode, which use previous both types. SQL Server has many features. Firstly, stored procedure is a mediator between program page and database that used to build and examine procedure for one time and use it at any time. Secondly, default value where can establish a number of a default value for desired fields. Thirdly, determine rules for fields so any field would not accept any data does not meet rules. Finally, support a large number of users at the same time.

SQL Server 2008 supports structured and semi-structured data include audio, video, and digital media formats for picture and other multimedia data. SQL Server 2008 can store different varieties of data as XML, calendar, document, email, file…etc and also offer many tools as analysis, sharing, search, and query…etc, Figure (6.3) show the main screen of SQL server .

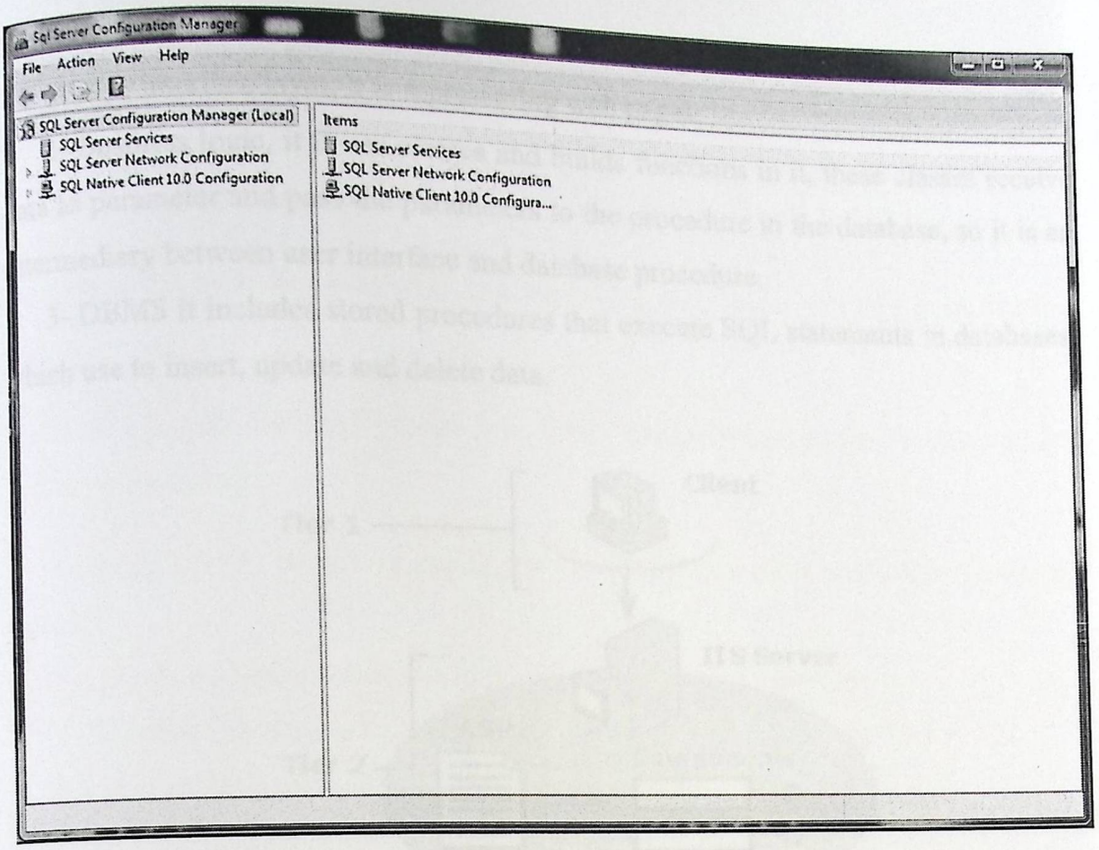


Figure 6. 3 Microso SQL server manager 2008

5- Microsoft Office 2003

It is a Microsoft product. It includes word processor (Microsoft word) to complete the documentation, Microsoft PowerPoint, and Microsoft Office Visio 2003 to accomplish the design, shapes, and drawing tasks.

6.1.3 Deployment diagram

To implement the system in real word, the system needs to be configured and to be uploaded on a web server to publish it. The system has to be browsed from any browser-controlled environment and ability to be accessed over internet.

To offer the ability for all clients to access the website without need to install any additional component on their desktop, the system designed as a Three-Tier Architecture as shown in figure (6.4).

Three-Tier Architecture has three layers as a following:

1- User interface layer, it is representing web pages.

2- Business logic, it is use classes and builds functions in it, these classes receive data as parameter and pass the parameters to the procedure in the database, so it is an intermediary between user interface and database procedure.

3- DBMS it includes stored procedures that execute SQL statements in databases, which use to insert, update and delete data.

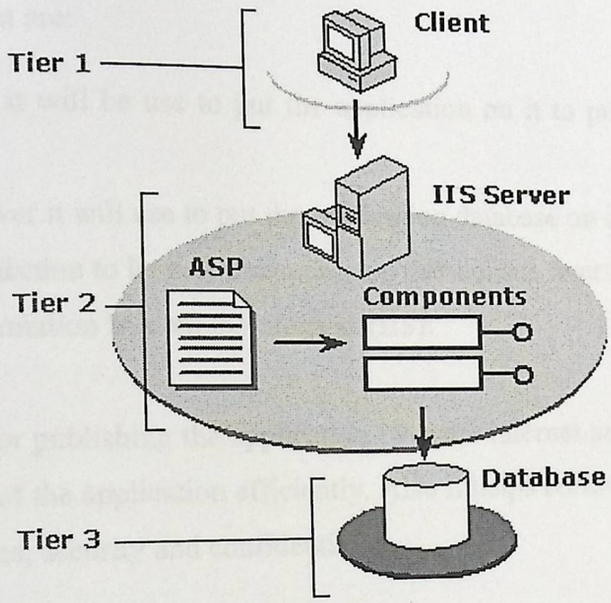


Figure 6. 4 Three-Tier structure

6.1.4 System installation

To make the system work effectively and efficiently in real life, the system must be deployed by certain steps, the following points describe the steps of the system deployment:

1. **Establish production environment:**

As explained in feasibility study in chapter four, the hardware and the software must meet the minimum requirements to deploy the system.

2. **Decision of deploying the system:**

At this step we make the decision to deploy the system, the decision will be made after answering the following questions:

- Dose the new system meet all the functional requirements that the system needed?

- Does the available working environment have the minimum requirements to run the new system and deploy it?

3. Running the system

This is the final step and after its completion, the system will be running in real life.

In this project the deployment need many hardware and software requirements and these requirement are:

1- Web Server, it will be use to put the application on it to publish it over the internet.

2- Database server it will use to put the application database on it.

3- Internet connection to let researchers access the website over the internet.

4- Internet Information System maintenance(IIS):

IIS is required for publishing the application over the internet and it is essentially for the deployment of the application efficiently. Also it helps controlling some of the system characteristics, security and confidentiality.

It is possible to run the database and the web service over one machine, but it is better to run database and web server on different machines for security reasons.

6.2 System Testing

6.2.1 Introduction

Testing is very important step before deliver the system to ensure that the system work as expected and satisfy the system requirements.

The system need testing phase to check the efficiency and the effectiveness of the system. The system will be test according to these levels:

1- Unit testing.
2- Sub-system testing.
3- Integration testing.
1- System testing.

6.2.2 Unit testing

This phase dividing the system into separate components, and then tests each component separately to ensure that each unit achieves the requirements.

The system contains many unit test cases, these cases are search testing, administrator upload excel file, update the system database.

In this section we will discuss "update the system database" as an example for unit testing. And we will use equivalent -partituning approach, in this case we will make the update for the system Database manually, so the system will get all proteins accession number from database and make a connection with Uniprot and download all proteins information as XML files, then XML files will convert to SQL server Database and will update existing proteins information.

Case	Test case	Expected Result	Actual Result
1	Uniprot server goes down	Destination Connection error	Error message
2	Lose the connection with the database	Connection establish error	Error message
3	All protein downloaded correctly	Download desired information from uniprot	successful message

Table 6. 1 Manual knowledgebase update testing

Case 1:

This test is about connection to uniprot to get desire protein information, so we will turn off internet connection, and then request manual knowledgebase update, figure (6.5) will show test result.

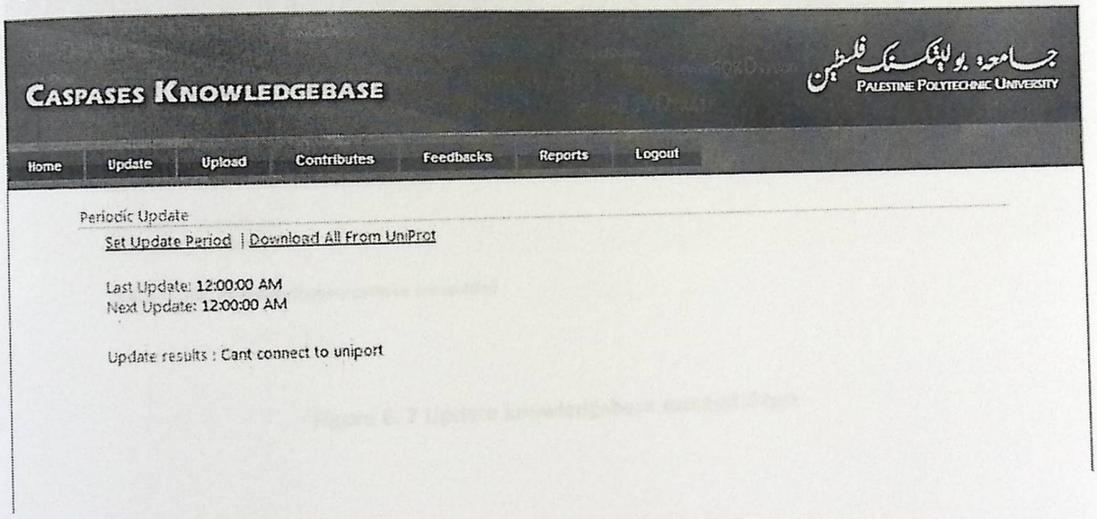


Figure 6. 5 Connection to uniprot test

Case 2:

This test is about connection to knowledgebase to get protein accessions, so we can update these proteins, figure (6.6) will show test result.

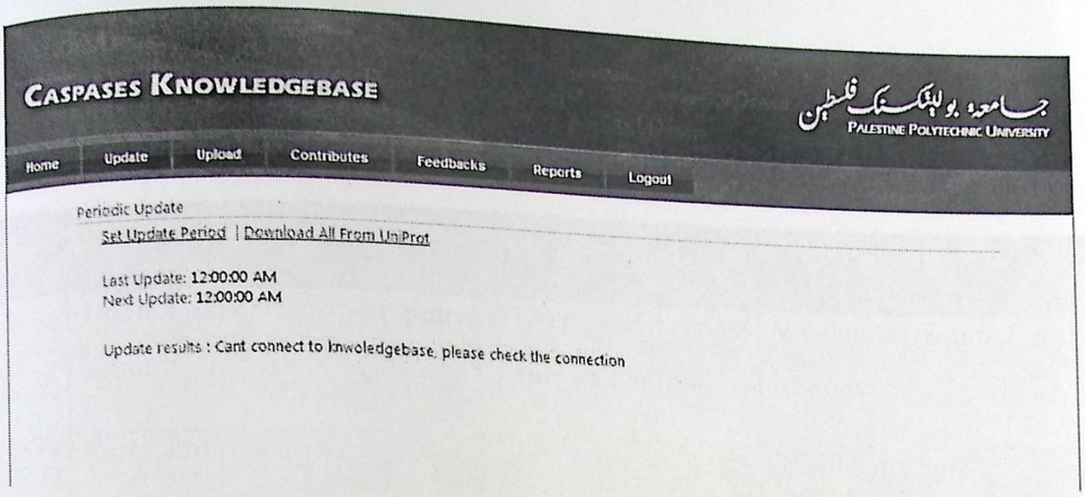


Figure 6. 6 Connection to the knowledgebase test

Case 3:

This test is to show successful knowledgebase manual update, figure (6.7) will show test result.

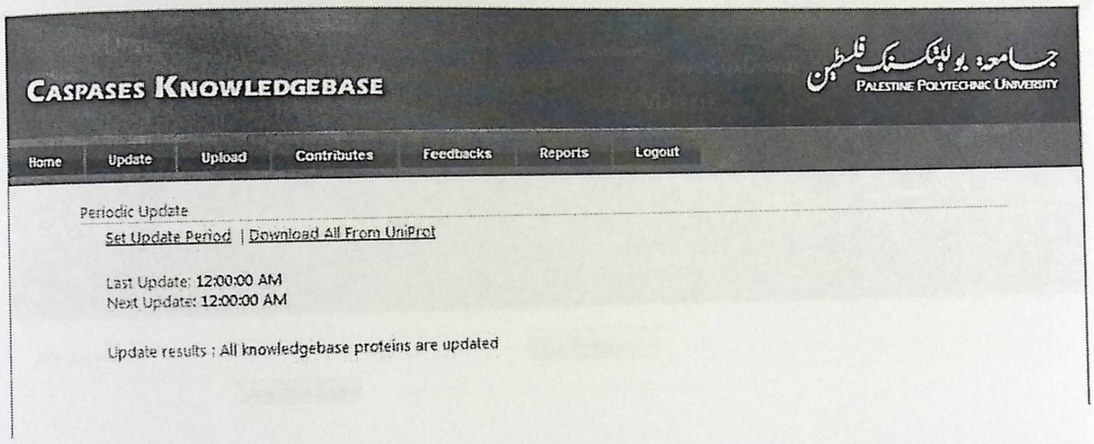


Figure 6. 7 Update knowledgebase successful test

6.2.3 Subsystem testing

Subsystem tests aid in fault isolation by testing specific functions within a subsystem to determine if they are generated correctly. In many cases, these tests check the transmission of data between the subsystem under test and associated subsystems. Computer programs are available that provide specific test capabilities suited to subsystem testing. Also this test includes testing data type and format.

The most important example about subsystem testing is searching mechanism testing, this subsystem work to search for specific data and pass it throw different system pages. In table (6.2) shows test cases.

case	Test case	Expected Result	Actual Result
1	Do not enter word to search about it	User and server side validation	Validation required will be araies
2	Enter wrong word	Suggest correct words	Word correction
3	Get more than one result	List results	List the results desired
4	Pass data to protein information page,	Selected protein information	Show proteins information

Table 6. 2 Searching test

We will take case1 as example case. Figure (6.8) will show test result.

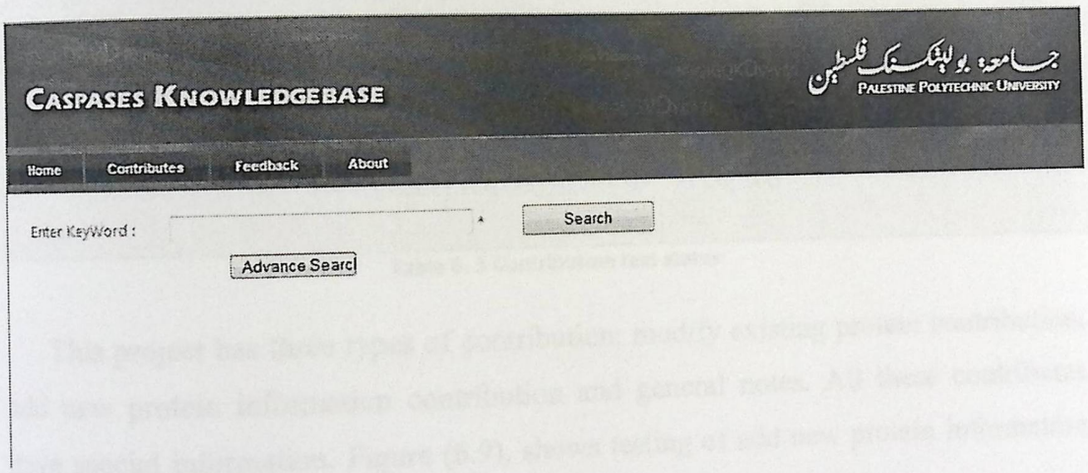


Figure 6. 8 Do not insert keyword

6.2.4 Integration testing

This phase test a integral functions as one unit to make sure that individual modules are combined and tested as a group completely compatible and matching the requirements.

The most important integral functions example in the system is contribution and contribution confirmation.

Here we test adding contribute functions and it has many states. In table(6.3) will show the states and the results.

Entered values	Expected Result	Actual Result
Do not enter required fields in contribution and feedback pages.	Validation controls and error summary	Validation required will arise
Do not enter captcha box result	Validation controls	Validation required will arise
enter required information correctly	Accept the contribution or the feedback	Send the contribution to the administrators.
Accept countribution	Its statuse will change to accepted	Get protein information from the contribution,uniprot and pubmid
Reject contribution	Its statuse will change to accepted	Its statuse will change to accepted

Table 6. 3 Contribution test states

This project has three types of contribution: modify existing protein contribution, add new protein information contribution and general notes. All these contributes have special information. Figure (6.9), shows testing of add new protein information contribute.

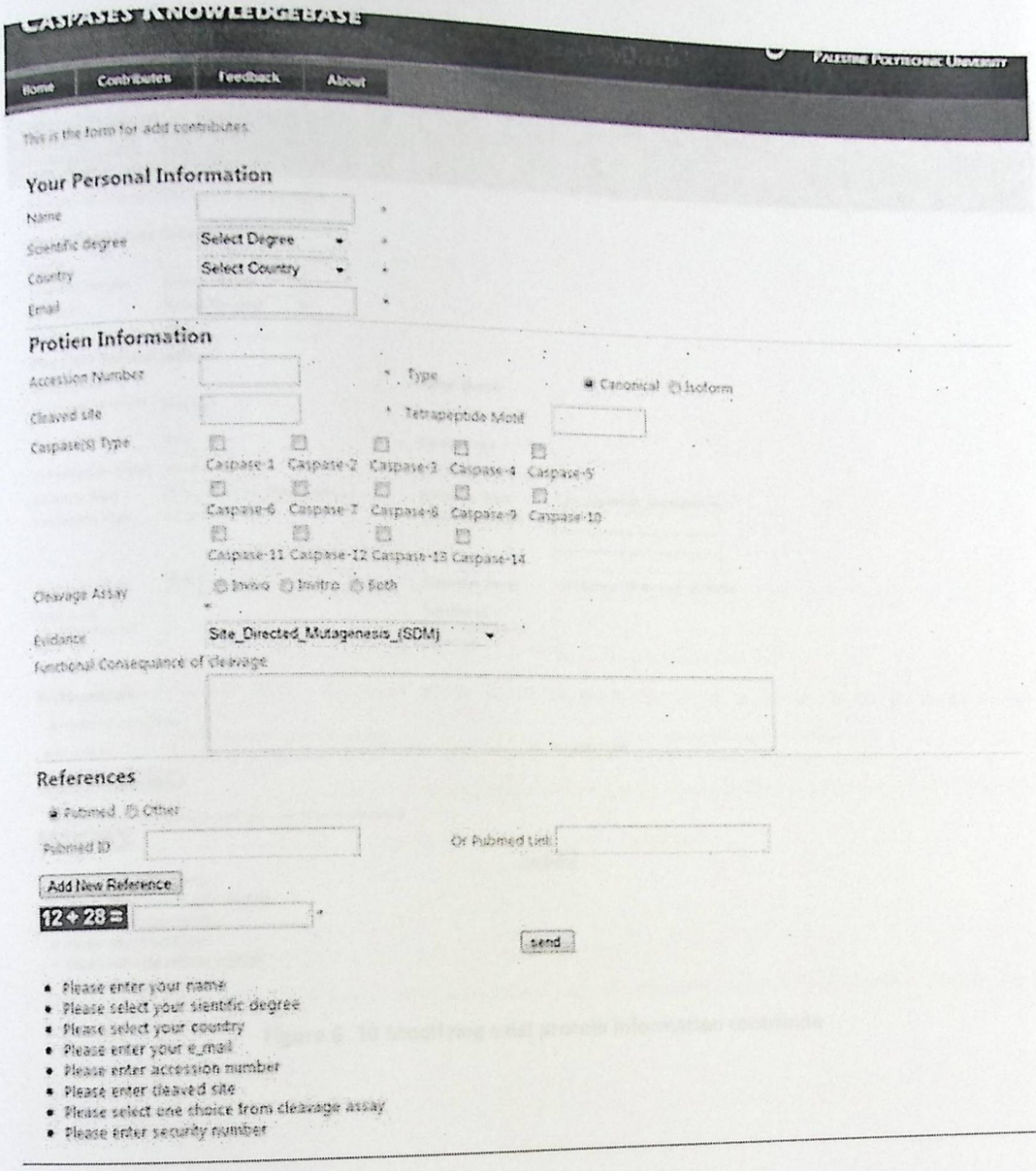


Figure 6. 9 Add new protein contribute

Figure (6.10) shows testing of modifying exist protein information contribute.

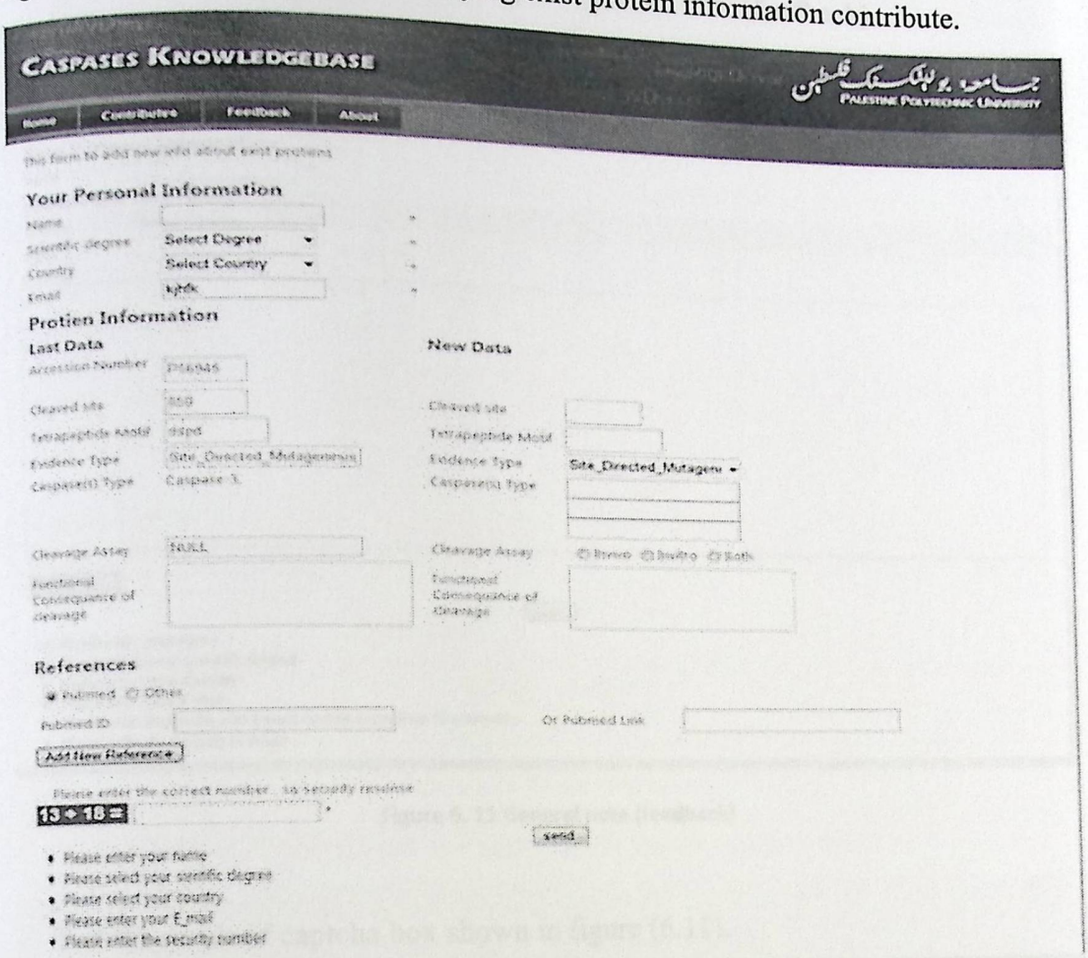


Figure 6. 10 Modifying exist protein information contribute

Figure (6.11) shows testing of adding new general note (feedback).

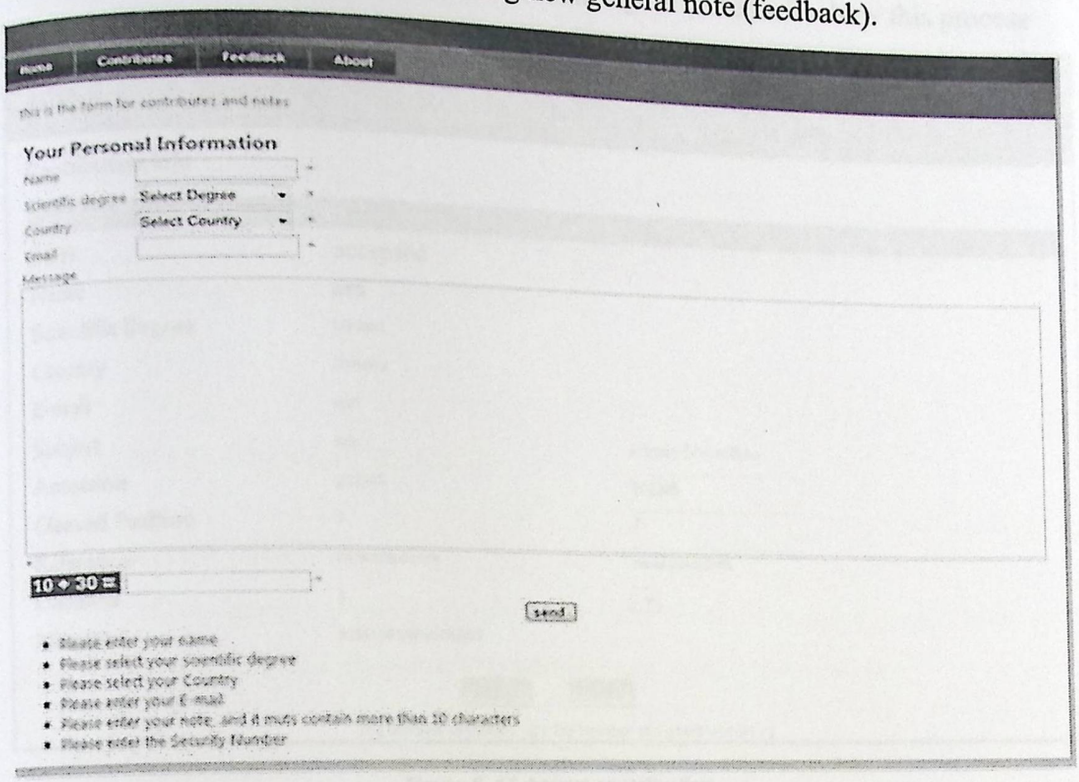


Figure 6. 11 General note (feedback)

Test the result of captcha box shown in figure (6.11).

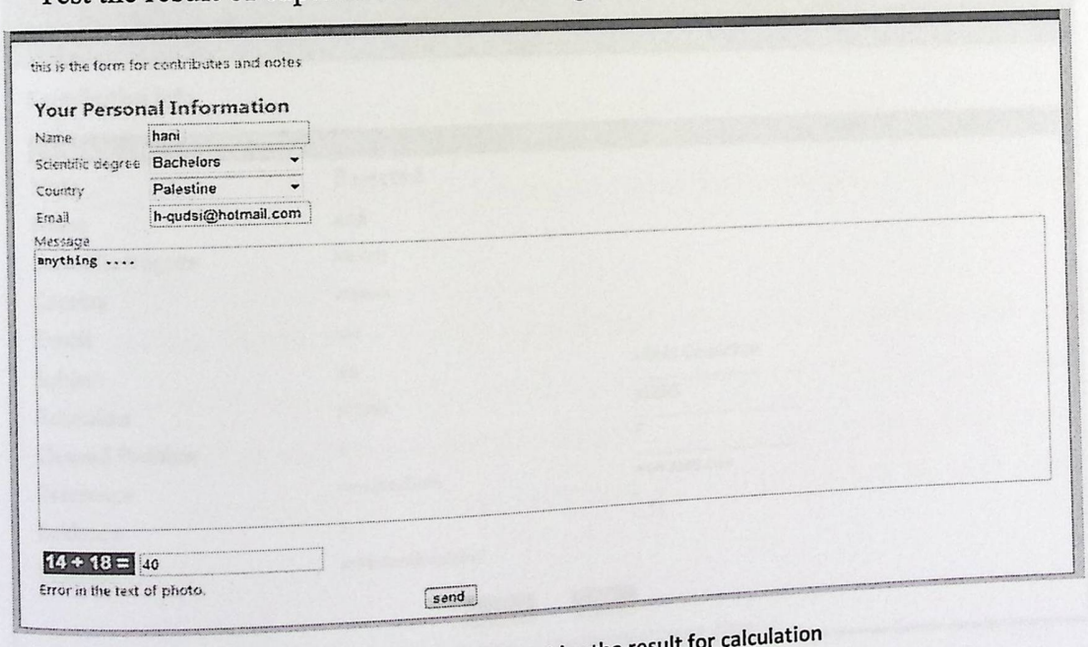


Figure 6. 12 Test and consider the result for calculation

Test the accept of contribution information, figure (6.13) will show this process

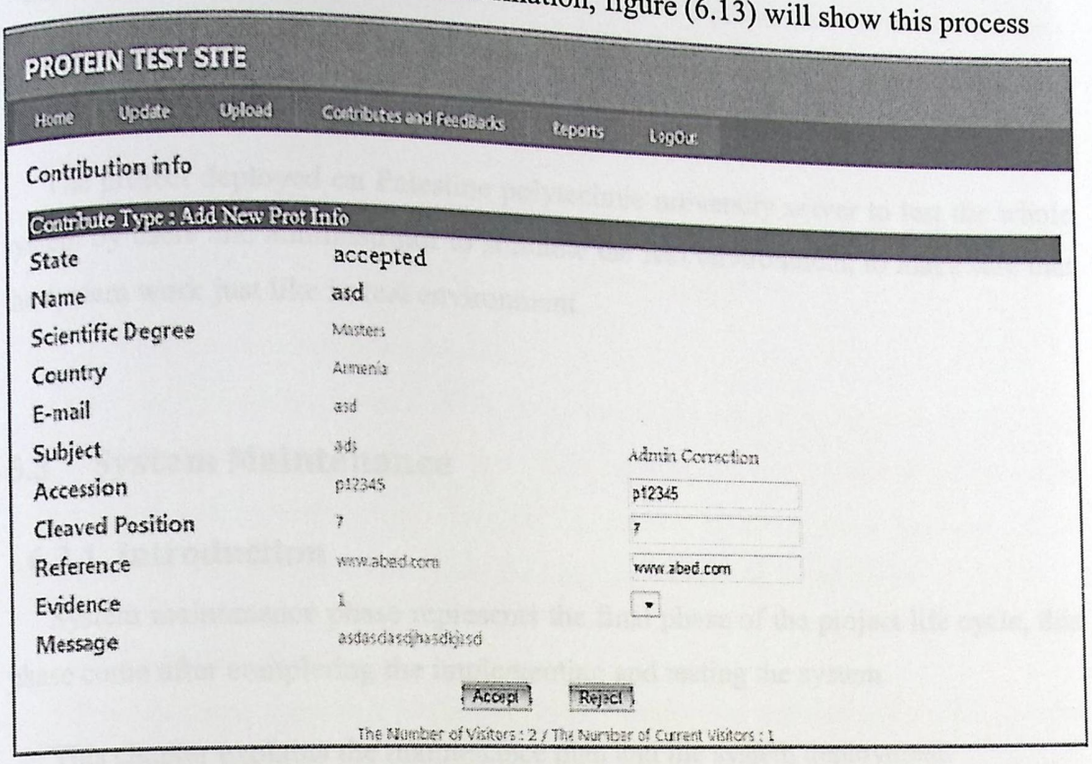


Figure 6. 13 Accept contribution

Test the reject of contribution. Figure (6.14) shows this process:

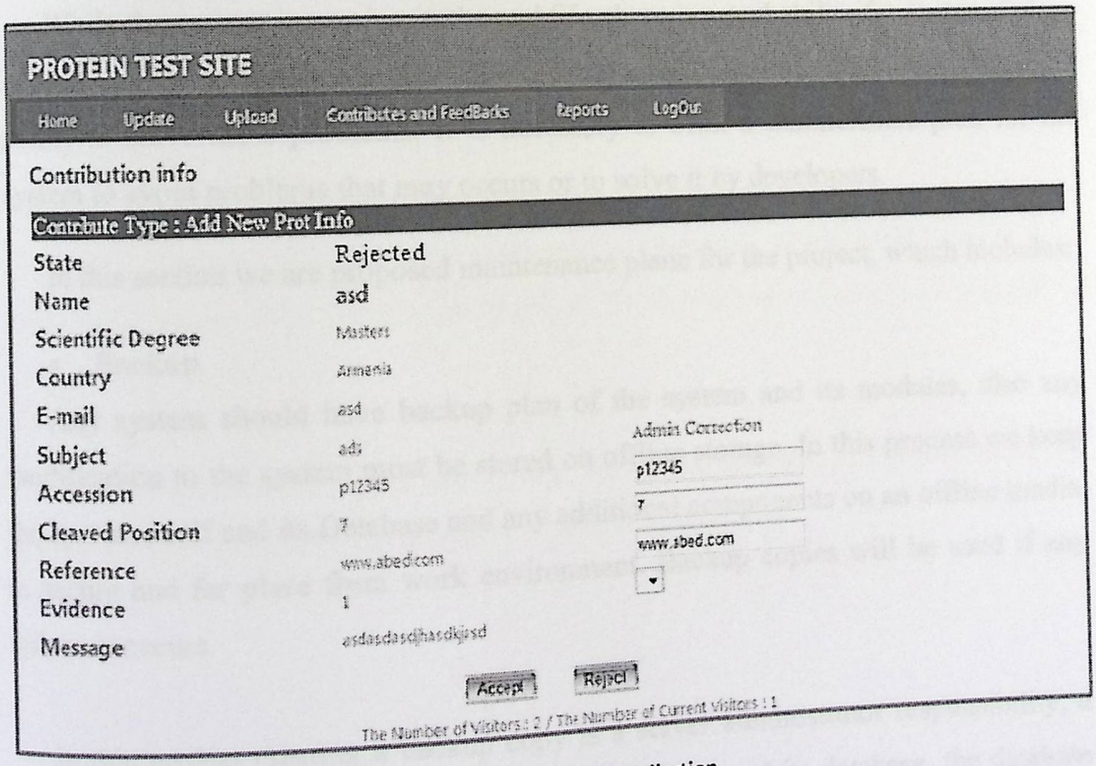


Figure 6. 14 Reject contribution

6.2.5 System testing

This phase of testing aims to validate the system accuracy, performance and efficiency, also this test simulates a real data under environment requirements.

The project deployed on Palestine polytechnic university server to test the whole system by users and administrator to simulate the real environment, to make sure that the system work just like in real environment.

6.3 System Maintenance

6.3.1 Introduction

System maintenance phase represents the final phase of the project life cycle, this phase come after completing the implementing and testing the system.

This chapter explains the maintenance plan and the system deployment.

6.3.2 Maintenance plan

While the system is running in the real life, there is a probability for system failure or some errors and problems that should be avoided. The end user doesn't have the ability to solve these problems. It is necessary to offer a maintenance plan for the system to avoid problems that may occurs or to solve it by developers.

In this section we are proposed maintenance plane for the project, which includes:

- **Backup**

Any system should have backup plan of the system and its modules, also any modification to the system must be stored on offline storage. In this process we keep the system itself and its Database and any additional components on an offline media, in secure and far place from work environment. Backup copies will be used if any failure is occurs.

In this project creating a backup copy is a server administrator responsibility, a backup copy will be create every week for the system and its database, the database backup will be token using SQL Server Tools and the system backup will be token by

copying system files, and then put both in one copy. Because backup process is very important process, each copy should be saved in several places.

- **Upgrade**

Upgrading the system by replace existing system with a newer version, newer version will contain new characteristics for the system, to facilitate and help the browsers and researchers to reach and do what they need easily.

6.3.3 NET Framework Maintenance

Using the Visual Studio and .NET Framework can amend to the rules at any time and make improvements to design user interfaces in the application to increase the performance and effectiveness. Through the screen in the solution explorer visual studio can see all the pages and select the page that needs an amendment.

6.3.4 IIS Maintenance

IIS is the basic approach to the process of deploying applications on the internet so that the successful deployment of applications, efficiency and effectiveness required. It depends primarily on the basic accuracy, confidentiality and stability enjoyed by the IIS. In order to maintain the confidentiality and security in the form required the user must choose the appropriate value for the option of confidentiality for the security.

6.3.5 The SQL Server 2008 Maintenance

In SQL Server 2008, database maintenance plans are almost entirely created by using SQL Server 2008 Integration Services. The main part needed is the sample of database that build.

By maintenance we finish the system life cycle, so after this phase when the number of errors increase the system will need reengineering.

CHAPTER 7

CONCLUSION & RECOMMENDATION

Content:

1. Introduction
2. Conclusions
3. Recommendation

7.1 Introduction

After the team work finish the system, they reach some conclusions will be explained in this chapter, and there are some of recommendations which is aim to improve the system in the future.

7.2 Conclusion

This web application has been build to achieve the objective of the project, Caspases Knowledgebase is a unique biological resource for researchers from various biomedical disciplines, which represents a reliable, sustainable and self evolving bioinformatic system.

7.3 Recommendations

The work team recommend to who work in this field the following:

1. Enhance protein presentation by generate 3D images.
2. Update proteins information from several trusted databases not just Uniprot.
3. Develop a protein prediction tool and integrate it in the system.
4. Present protein pathway and protein-protein interaction inside the system, not just provide links to other websites.

Appendix A: Administrators guide

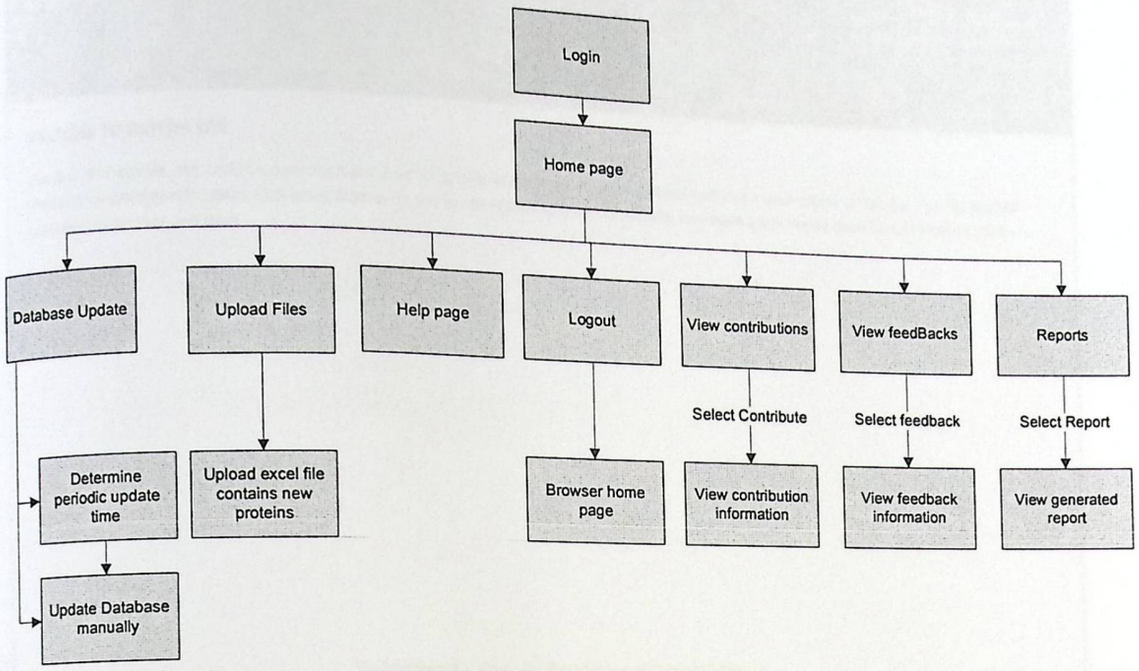
Introduction

This guide designed to help administrators, and to facilitate using this system efficiently, also it describes the screens for administrators and explains their functions.

The manual contains the following screens:

1- Navigation.
2- Home.
3- Update.
4- Upload.
5- Contributes.
6- Feedbacks.
7- Logout.

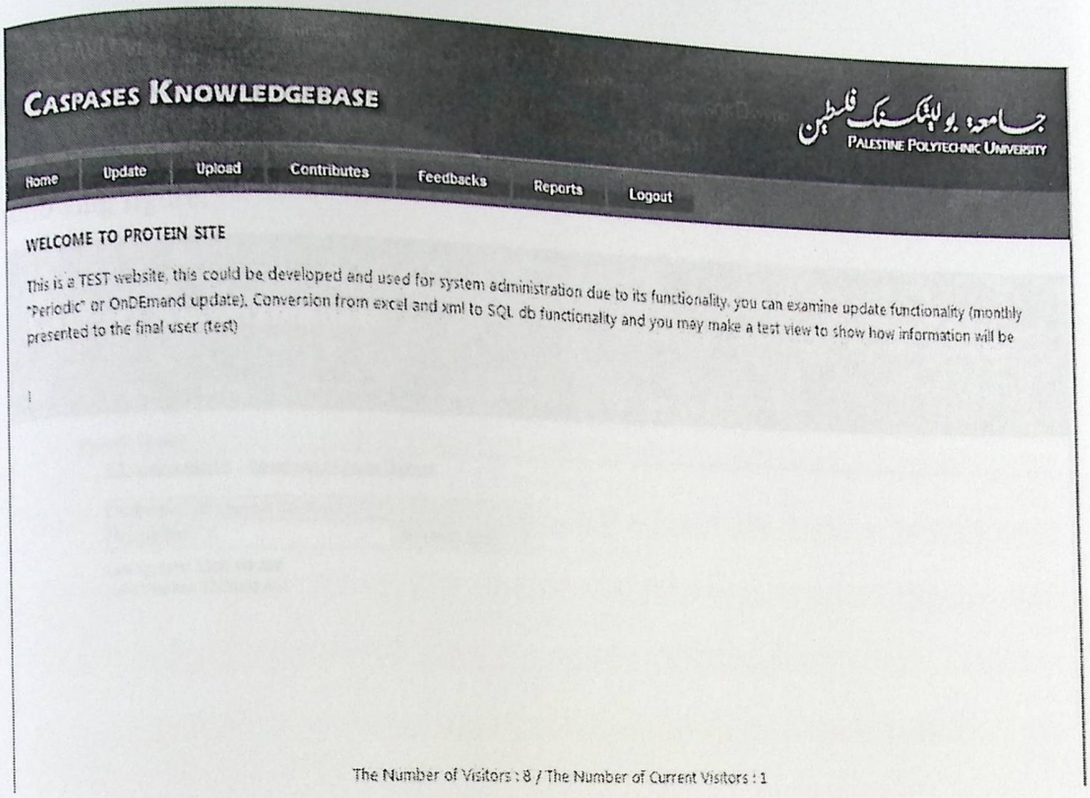
1- Navigation:



2- Home page:

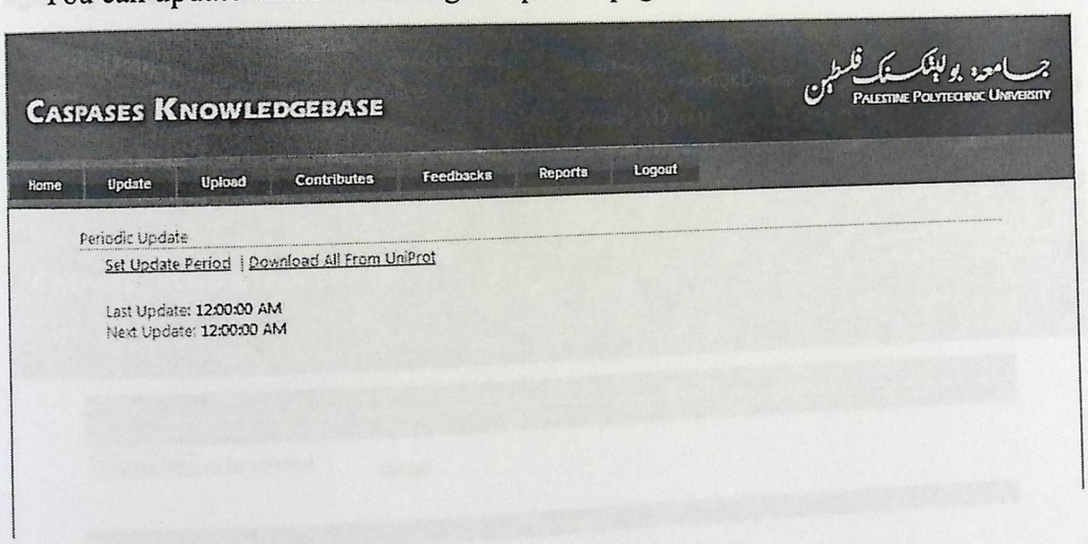
This page contains information that describes administrator's tasks. The user comes to this page once he login.

In addition, this page contains a main menu that includes all pages: Home, Update, Upload, Contributes, Feedbacks, Reports and Logout. As shown the following figure.



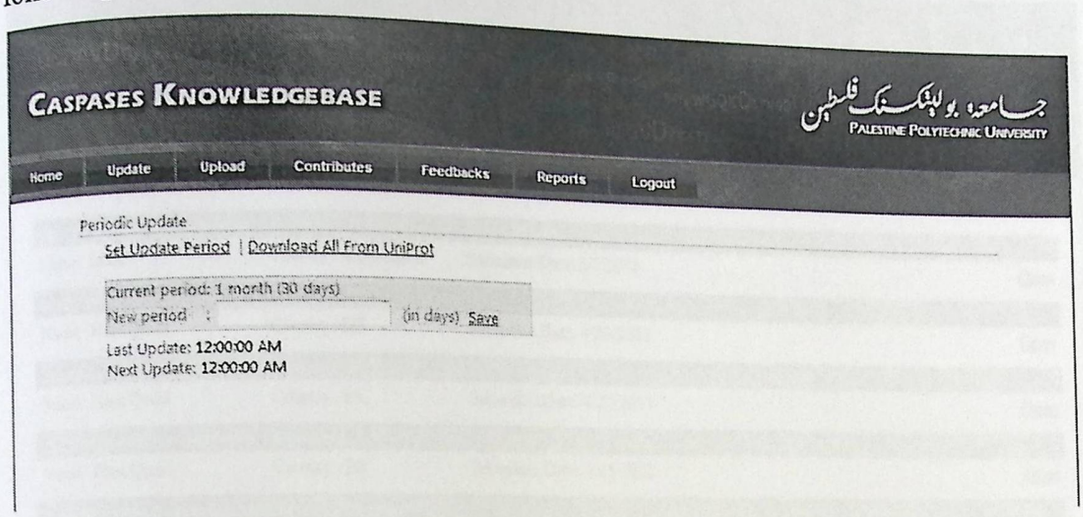
3- Update:

You can update database through "Update" page as shown in the following figure.



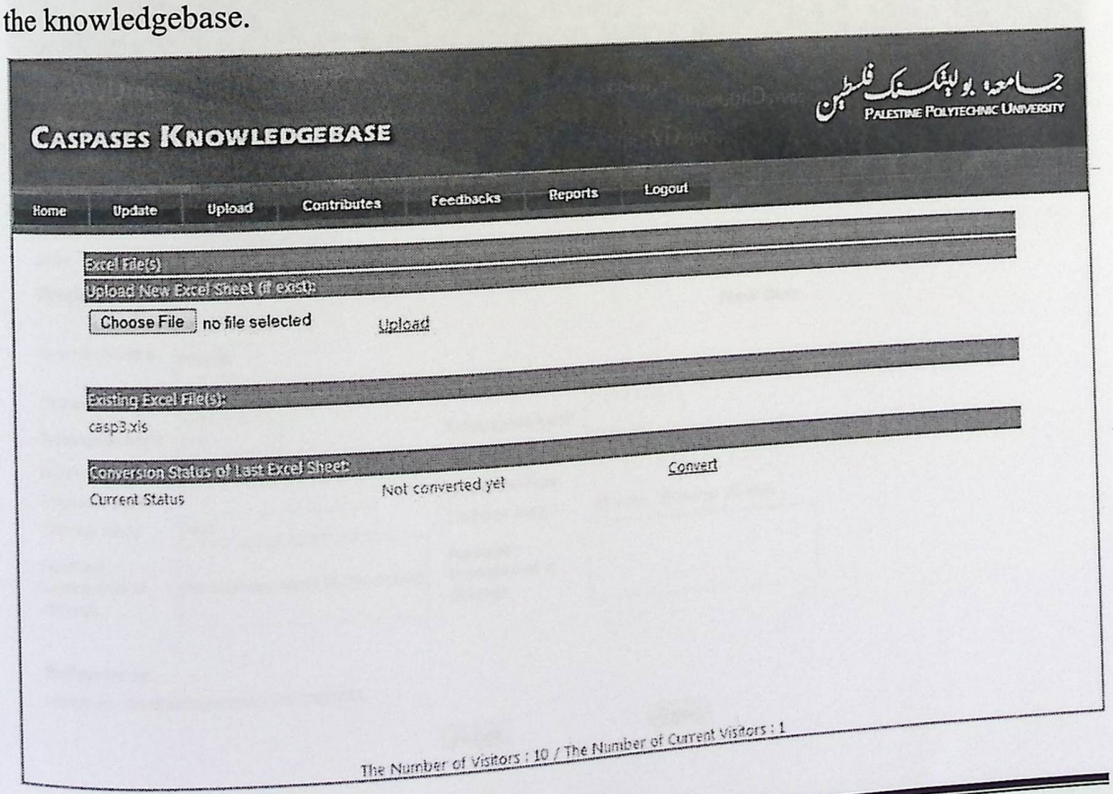
When you click on "Download All from UniProt" link, the system will display all accession numbers from the system knowledgebase and download all proteins that have the same accession numbers from Uniprot site as excel file, and then save its proteins in the system knowledgebase as SQL files.

Also, by clicking "Set Update Periodic" link it will allow you to make update on database periodically, and by inserting the number of days between two updates, the system will check the number of days and makes the update process, as shown the following figure.



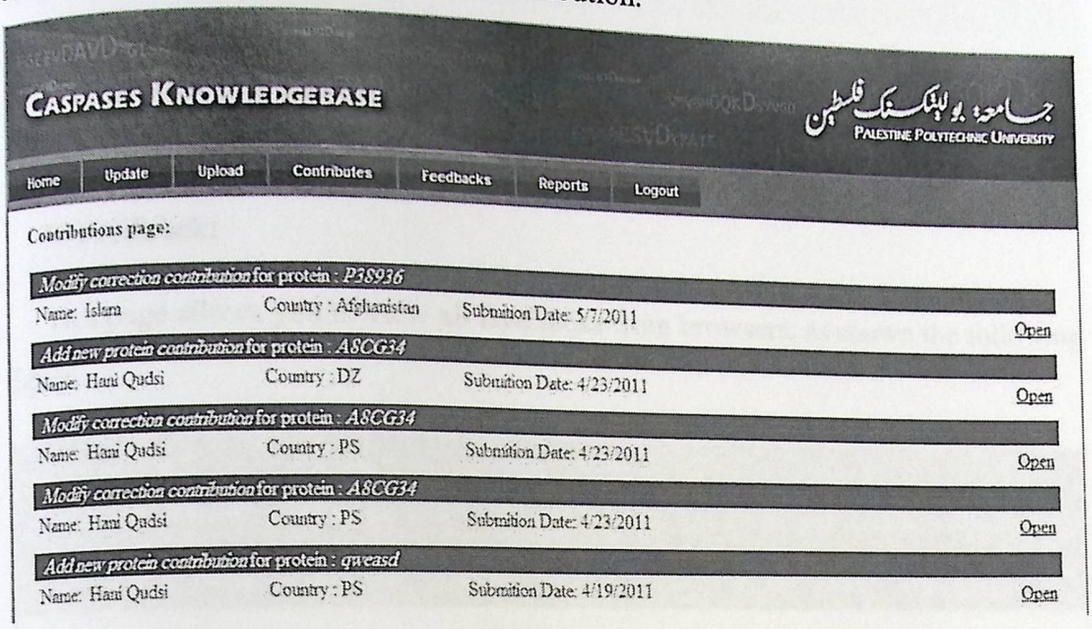
4- Upload:

This page allows administrators to add new protein(s), as shown in the following figure, you can browse the excel file that has the new protein through "Choose file" button and then upload this file to the system through "Upload" link, and by clicking on "Convert" link the system convert the data from excel to SQL file, and save it in the knowledgebase.

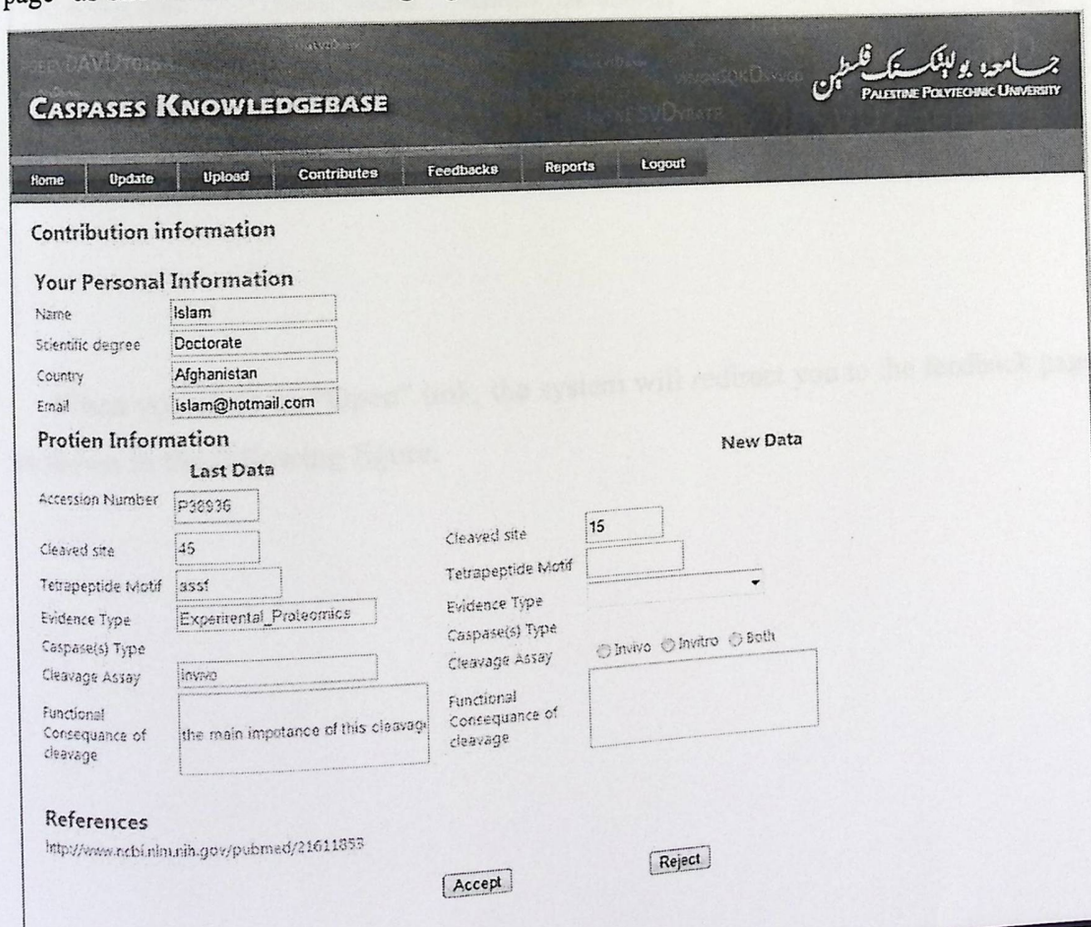


5- Contributes:

This page displays all browser contribution, which includes add new protein contribution and modify exist protein contribution.



When you click on "Open" link, the system will redirect you to the "contributes page" as shown in the following figure.

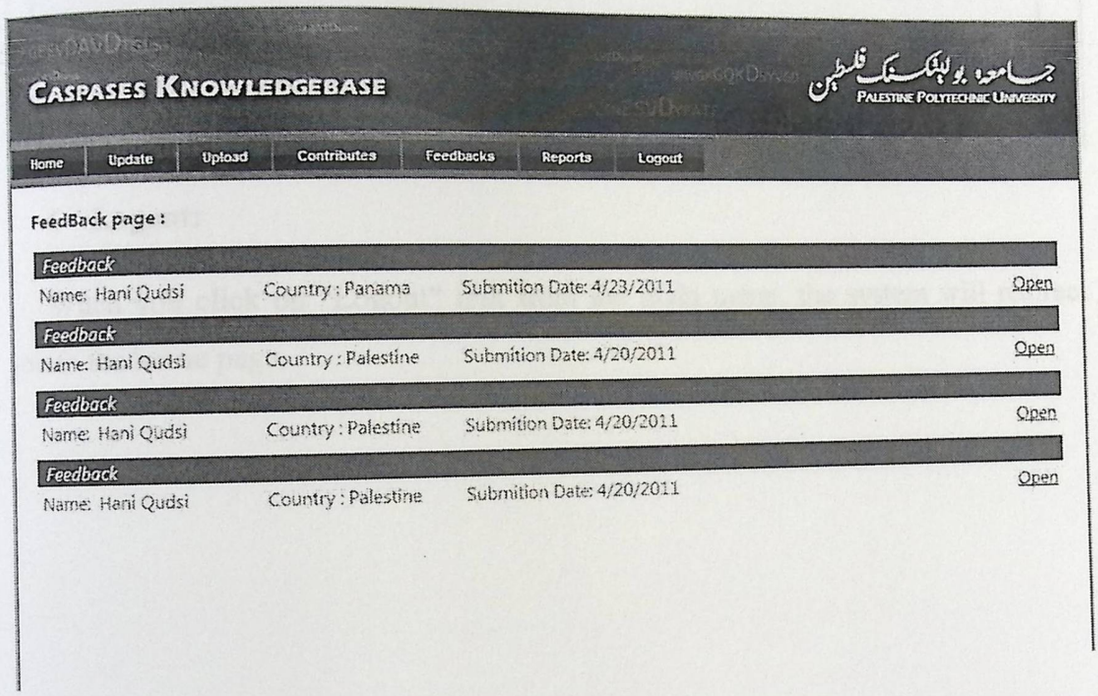


As you see, you can edit the contribute, and then accept it if it is necessary by clicking on "Accept" button, or reject it if it is not necessary by clicking on "Reject" button.

When you accept the contribution the system will download all information about the new protein from external database, and then save these information in the system.

5- Feedback:

This page allows you to view all feedbacks from browsers, as shown the following figure.



When you click on "Open" link, the system will redirect you to the feedback page as shown in the following figure.

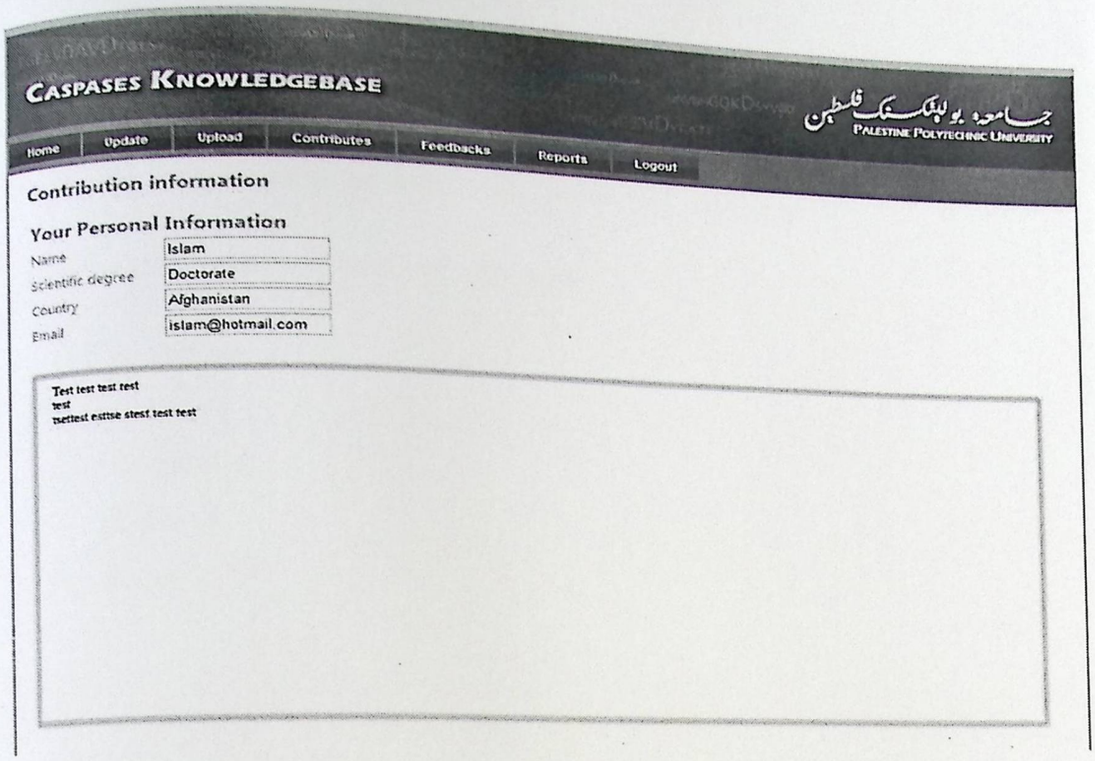
As you see, you can edit the contribute, and then accept it if it is necessary by clicking on "Accept" button, or reject it if it is not necesssary by clicking on "Reject" button.

When you accept the contribution the system will download all information about the new protein from external database, and then save these information in the system.

5- Feedback:

This page allows you to view all feedbacks from browsers, as shown the following figure.

When you click on "Open" link, the system will redirect you to the feedback page as shown in the following figure.



6- Logout:

When you click on "Logout" link from the main menu, the system will redirect you to the home page.

Appendix B: Browsers guide

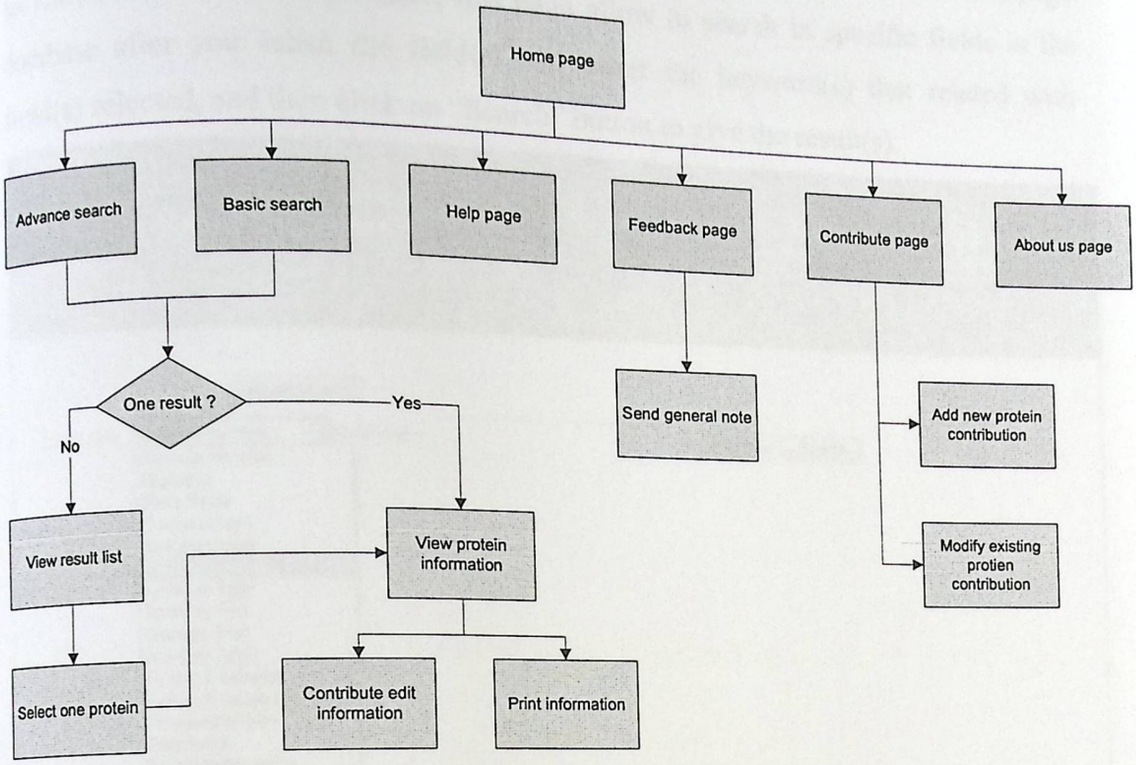
Introduction:

This guide designed to help users and facilitate using this system efficiently way, also describes the screens for users and explains their functions.

This guide contains the following screens:

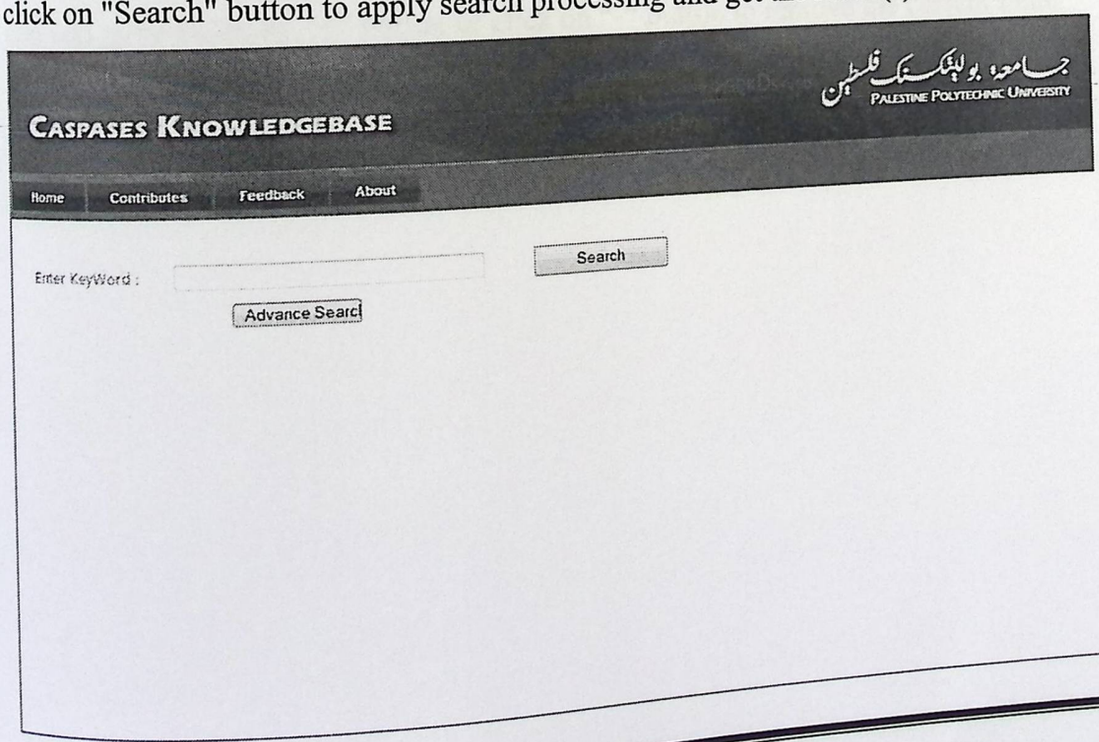
1- Navigation.
2- Home.
3- Contributes.
4- Feedback.
5- About.

1- Navigation:

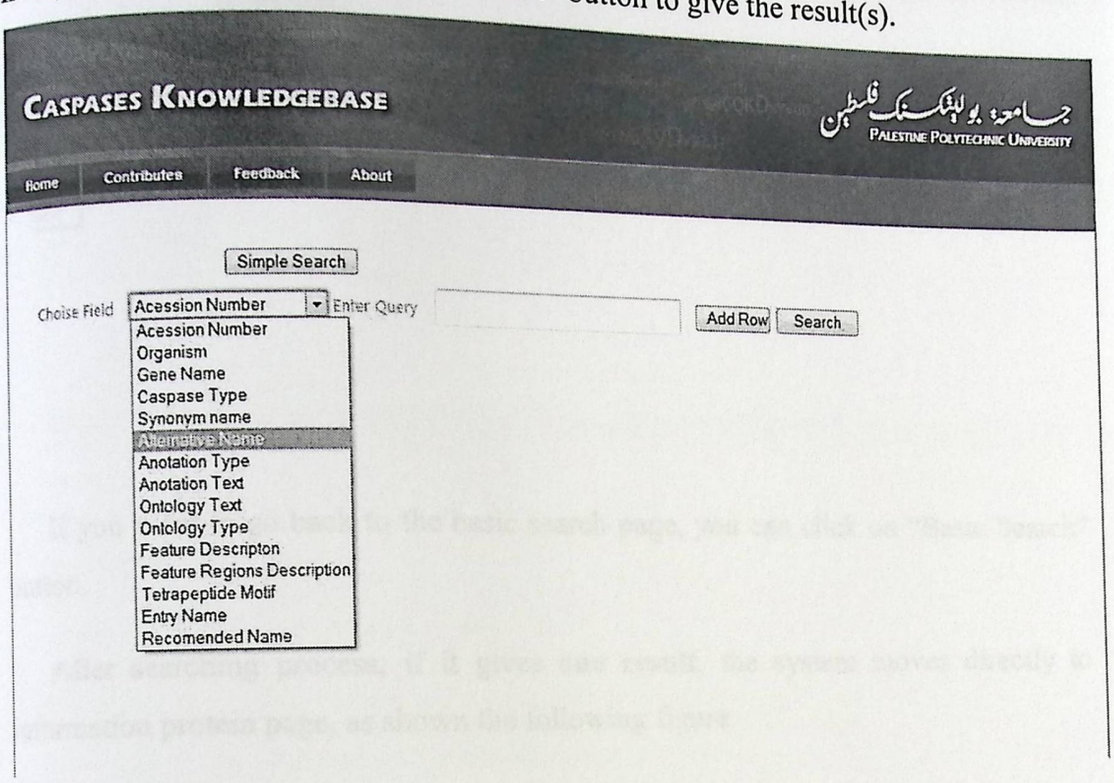


2- Home page:

This page contains a main menu, which includes Home, Contributes, Feedbacks and About page. Also contains basic and advance search tools. In the following figure, view the basic search automatically. You can inserting any keyword and then click on "Search" button to apply search processing and get the result(s).

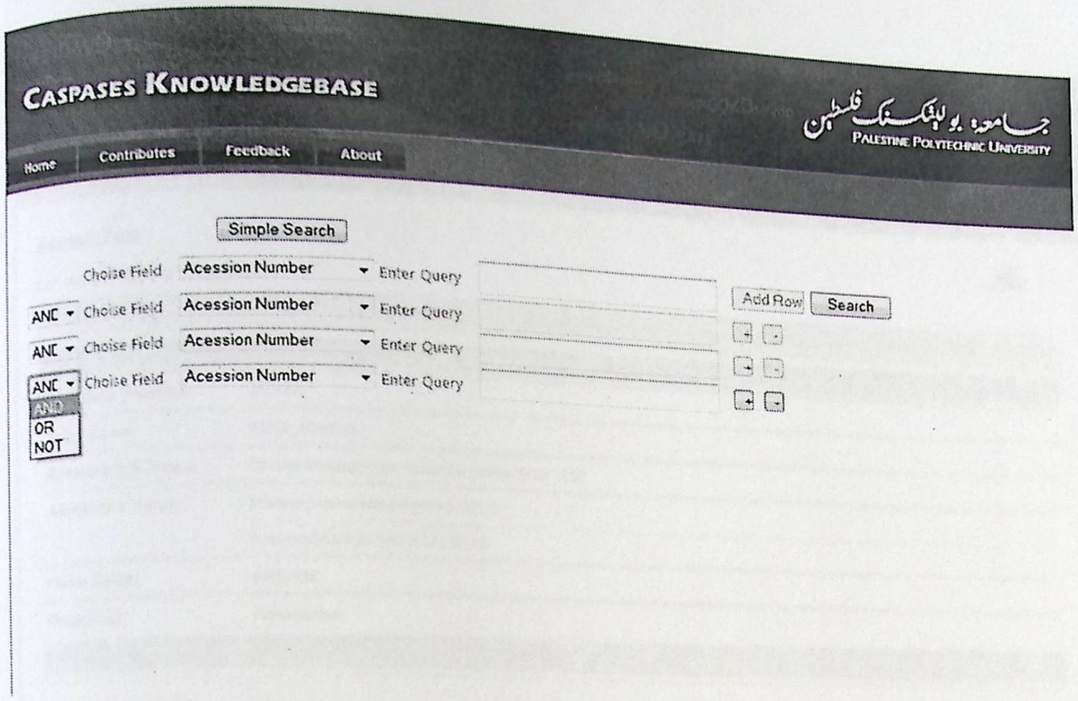


When you click on "advance search" button you move to the advance search page as shown in the following figure, this page allow to search in specific fields in the database after you select the field(s) and insert the keyword(s) that related with field(s) selected, and then click on "Search" button to give the result(s).



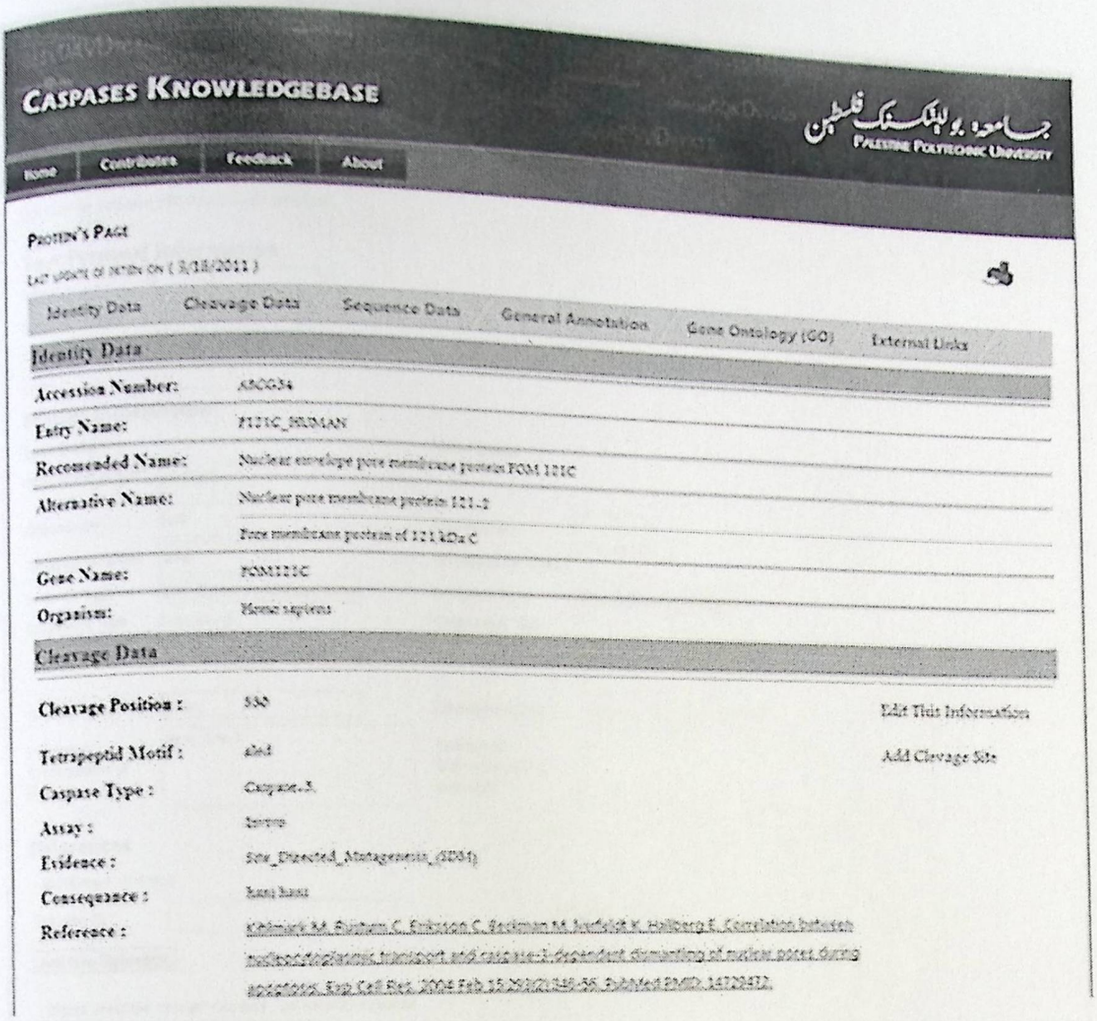
Also, in advance search technique you can add more fields to search for it by click on "Add Row" button to open another field as shown the following figure, and then can add more fields by "+" button or click on "-" button to remove the current field.

In addition, you can combine between fields by logical operations which contains (AND, OR, NOT).

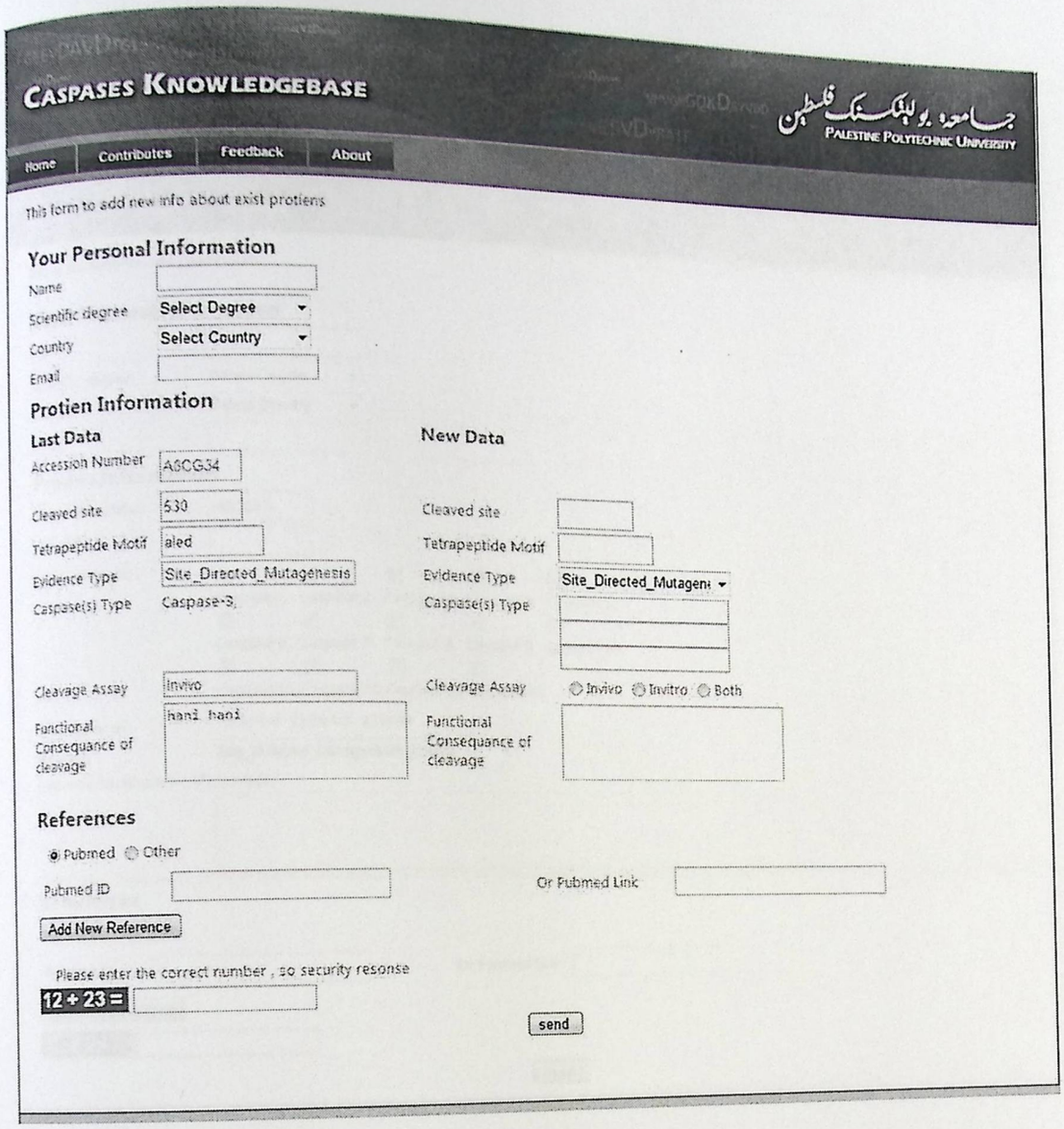


If you want to go back to the basic search page, you can click on "Basic Search" button.

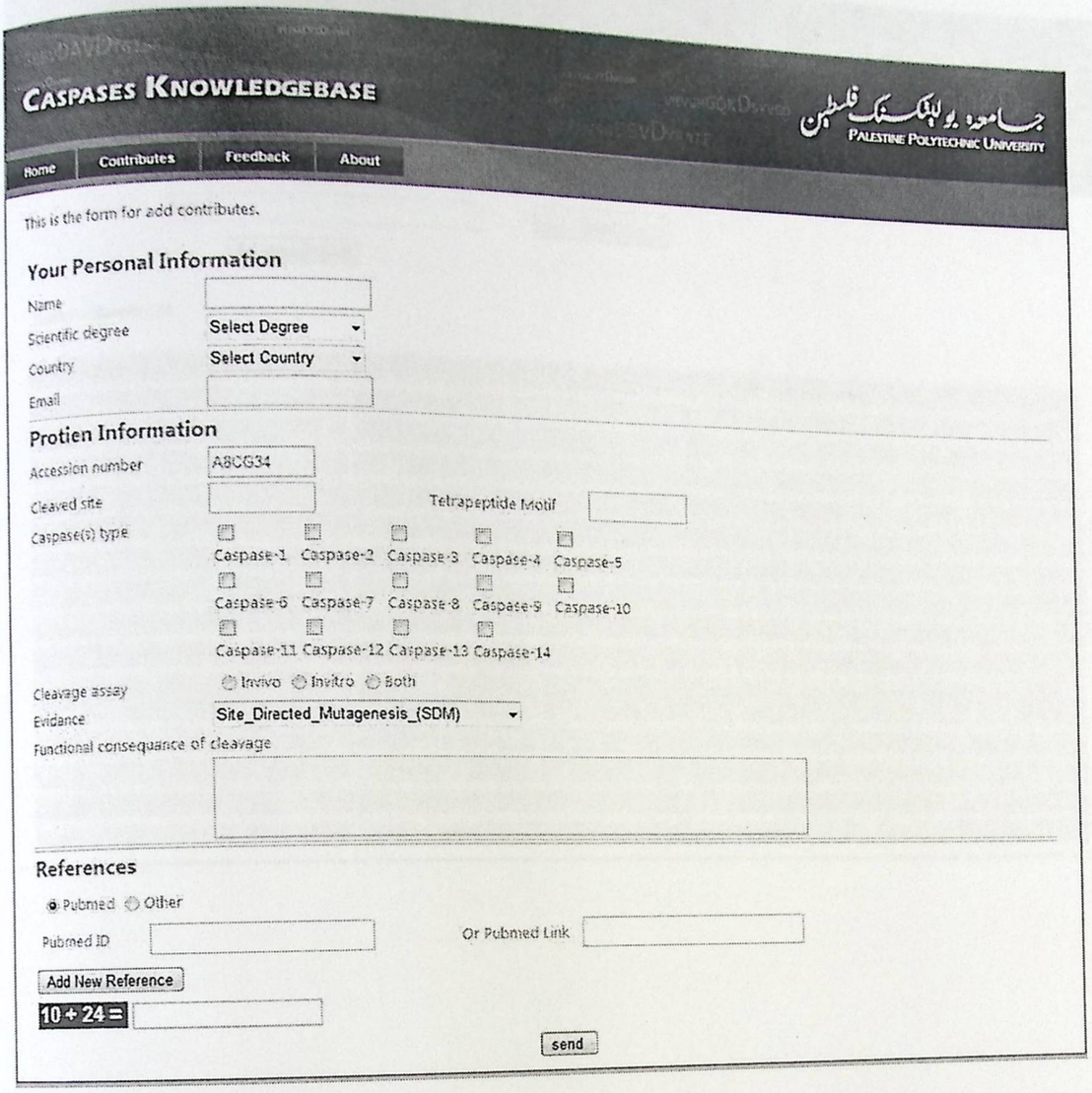
After searching process, if it gives **one result**, the system moves directly to information protein page, as shown the following figure.



In addition, the system offer technique to edit some information as "Cleavage Data" block, so when you click on "Edit This Information" link move to another page that let you to editing some information about the current protein, as shown in the following figure.



Moreover, you can add a new cleavages site for the same protein by click on "Add Cleavage Site" link, and the following figure view the screen that let you to add new cleavages site.

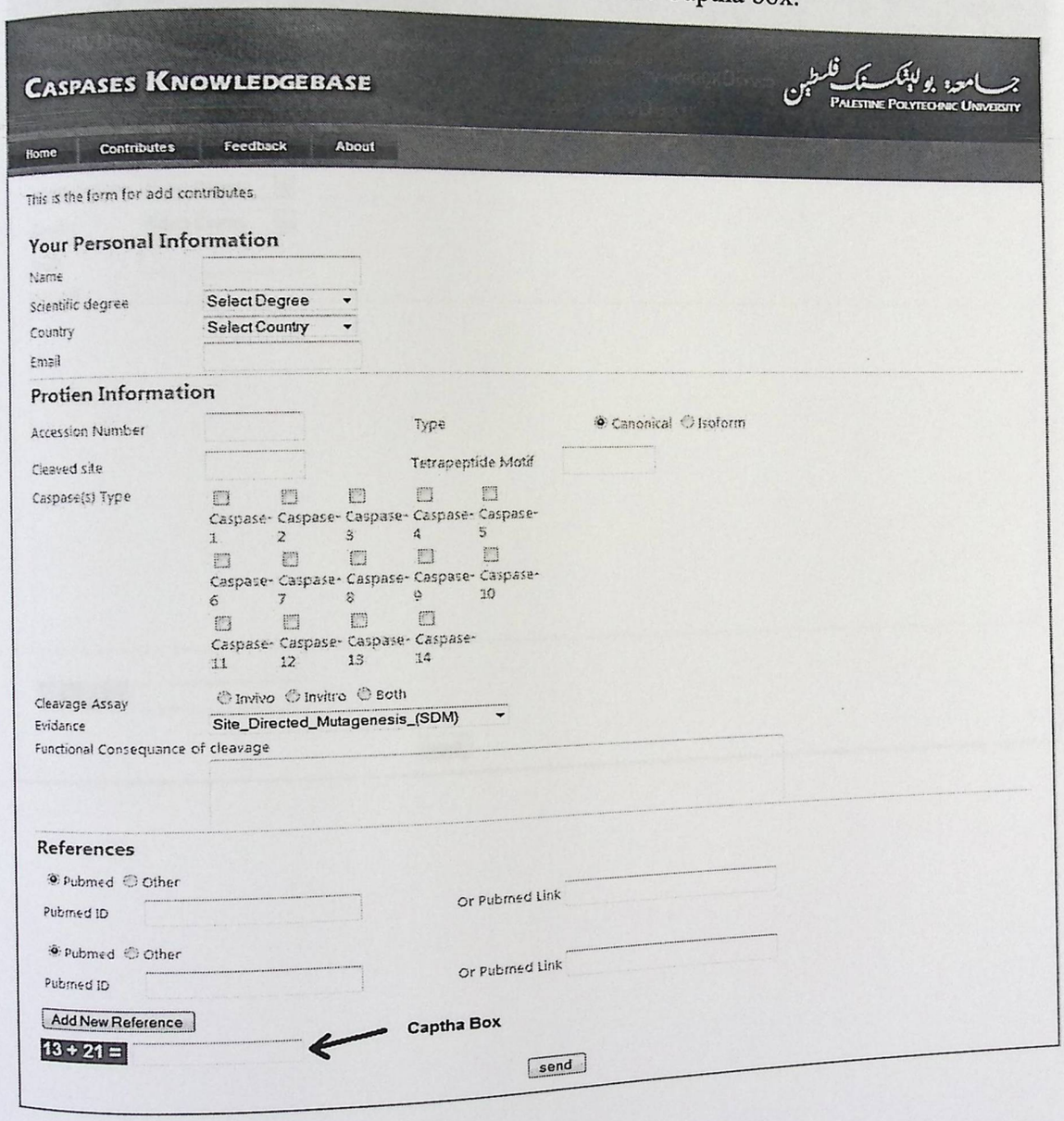


If the searching process **gives more than one result**, the system dispaly all results as the following figure and after that you can click on the Accession Number link for the chosen protein.

3- Contribute:

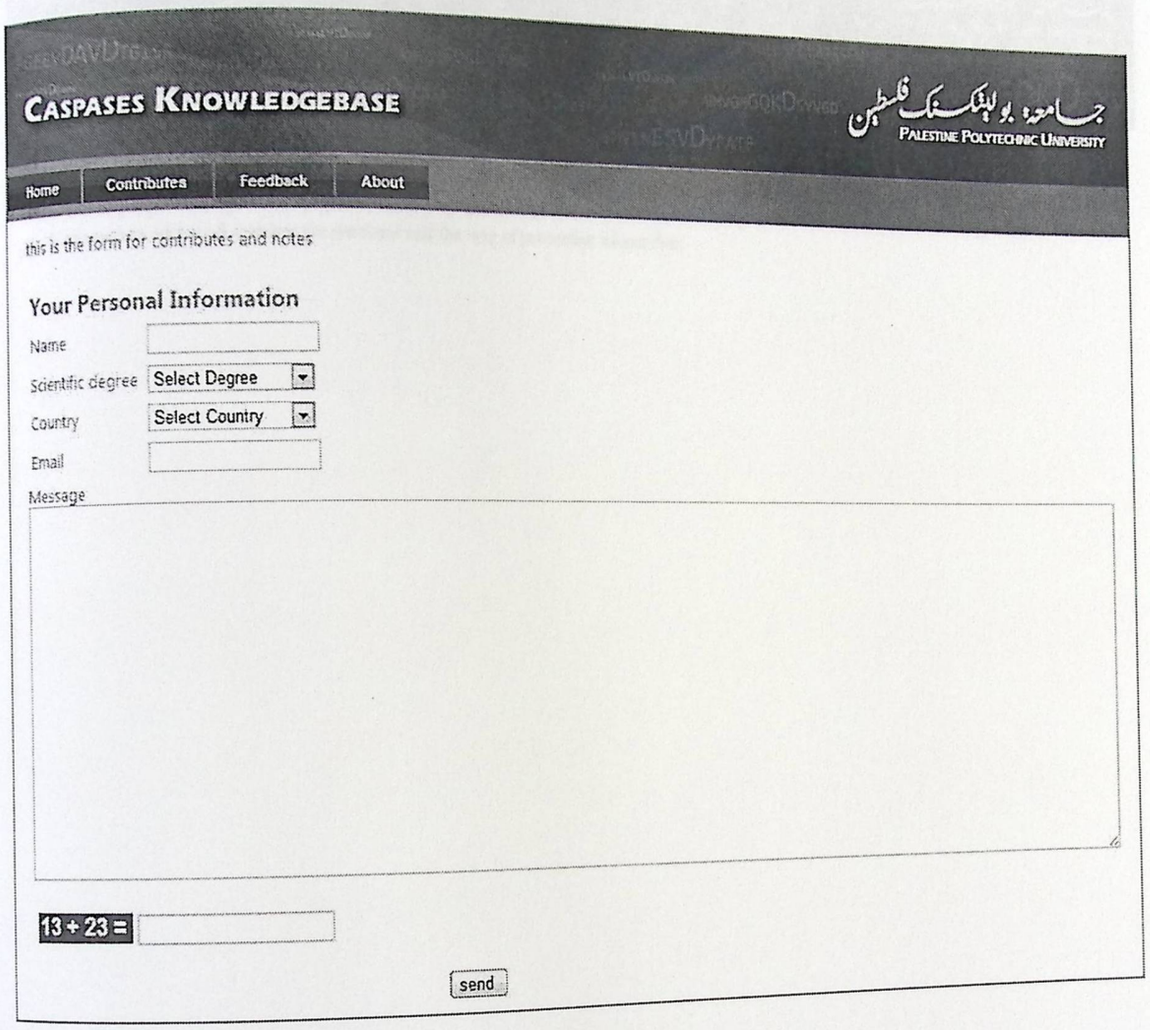
This page let you to contribute to add new protein as shown in the following figure the page dived to three type of information must insert, firstly information about contributor, secondly information about the new protein, and thirdly information about the source of your contribute and can you add more reference (source) for your contribute by click on "Add New Reference" button.

After that, you must click on "Send" button to sent your contribute for administrator after insert the result of calculation in the Captha box.



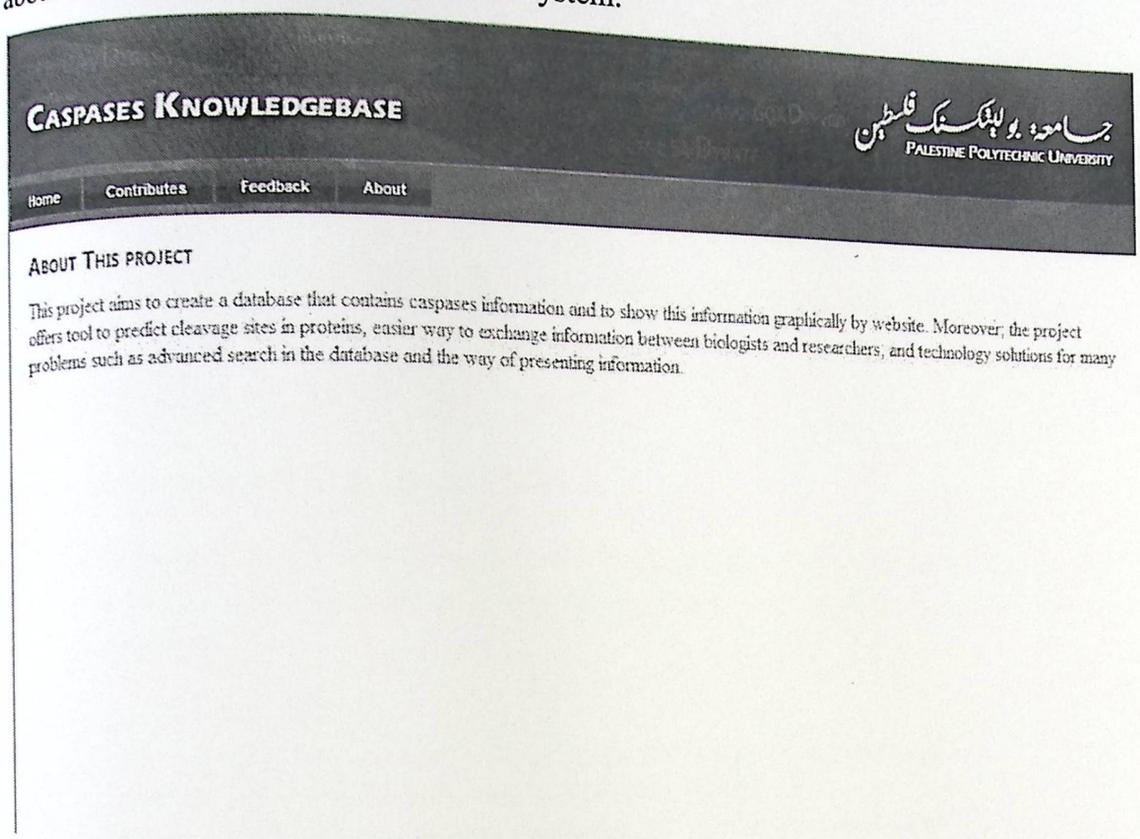
4- Feedback:

In this, page can you send feedback for administrators. Feedback may be about the site design, information, to ask questions or general notes. As the contribute page must be insert the result of calculation before send the feedback as show in the following figure.



5- About:

This page contains abstract information let you know about the system such as the idea, the objectives and short descriptions about the system. In addition, let you know about the work team that developed this system.



Appendix C: Deal with XML data

How we deal with xml files?

Asp.net technology provide some tools to deal and enter act with XML files, in this project we asp.net tools to parse XML files, we parse XML files manually by the system, and this is the code used to parse these files.

```
Public Sub ConductConversion()
    Dim XmlFiles = Directory.GetFiles(path)
    Dim Files() = XmlFiles.ToArray()
    Dim rd As XmlReader
    For i As Integer = 0 To XmlFiles.Count - 1
      check = True
      rd = XmlReader.Create(Files(i))
      While rd.Read ' Start Read The XML Elements ,,,,,,,,,,,,,,
        If rd.NodeType = XmlNodeType.Element And rd.Name = "accession" And
check Then
          rd.Read()
          If rd.NodeType = XmlNodeType.Text Then
            Accession_Number = rd.Value
            check_Name = True
            Insert(Accession_Number)
            check = False
          End If
        End If
        '------------------- STRART SEQUENCE ----------------------------------
        If rd.NodeType = XmlNodeType.Element And rd.Name = "sequence" Then
          Dim length As String = rd.GetAttribute("length")
          'MsgBox("length = " & length)
          While rd.Read
            If rd.NodeType = XmlNodeType.EndElement And rd.Name =
"sequence" Then
              Exit While
            End If
            If rd.NodeType = XmlNodeType.Text Then
              Dim seq As String = rd.Value
              'MsgBox("SEQ = " & seq)
              Dim comm As New SqlCommand
              comm.Connection = conn
              comm.CommandText = "UpdateSequence"
              comm.CommandType = CommandType.StoredProcedure
              comm.Parameters.Add(New SqlParameter("@Sequanice",
SqlDbType.NVarChar))
              comm.Parameters.Add(New SqlParameter("@Sequanice_Length",
SqlDbType.Int))
              comm.Parameters.Add(New SqlParameter("@Accession_Number",
SqlDbType.NVarChar))
```

How we deal with xml files?

Asp.net technology provide some tools to deal and enter xml files in our project we asp.net tools to parse XML files, we parse XML files manually by system, and this is the code used to parse these files.

```
Public Sub ConductsConversion()
    Dim XmlFiles = Directory.GetFiles(path)
    Dim Files() = XmlFiles.ToArray()
    Dim rd As XmlReader
    For i As Integer = 0 To XmlFiles.Count - 1
      check = True
      rd = XmlReader.Create(Files(i))
      While rd.Read 'Start Read NodeType.Text Then
        If rd.NodeType = XmlNodeType.Value)
check Then
          rd.Read()
          If rd.NodeType = CommandType.StoredProcedure
            comm.CommandText = "Update_Entry_Name"
            comm.Parameters.Add(New SqlParameter("@Entry_Name",
            SqlDbType.NChar))
            comm.Parameters.Add(New SqlParameter("@Accession_Number",
            SqlDbType.NChar))
        End If comm.Parameters("@Entry_Name").Value = CStr(rd.Value)
        '------comm.Parameters("@Accession_Number").Value =
        CStr(Accession_Number)
          Execute_NonQuery(comm, "Entry Name")
          End If
          check_Name = False
        End If 'end name
"sequence" Then ---------- E N D _ ENTRY NAME ----------
----------

        '---------- START _ R E C O M E N D E D ----------
----------
        If rd.NodeType = XmlNodeType.Element And rd.Name =
"recommendedName" Then 'start recomendedname

          Read_Next_Element(rd)
          'MsgBox("Read Recommended" & rd.NodeType)
          If rd.NodeType = XmlNodeType.Element And rd.Name = "fullName"
Then
            Read_Next_Text(rd)
            If rd.NodeType = XmlNodeType.Text Then
              Dim comm As New SqlCommand
              comm.Connection = conn
```

```
                comm.CommandType = CommandType.StoredProcedure
                comm.CommandText = "Update_Recommended"
                comm.Parameters.Add(New
SqlParameter("@Recommended_Name", SqlDbType.NVarChar))
                comm.Parameters.Add(New SqlParameter("@Accession_Number",
SqlDbType.NVarChar))
                comm.Parameters("@Recommended_Name").Value =
CStr(rd.Value)
                comm.Parameters("@Accession_Number").Value =
CStr(Accession_Number)
                Execute_NonQuery(comm, "Recommended name")
            End If

        End If
    End If 'end recomendedname
    '---------------------------- E N D _ R E C O M E N D E D ----------------
-----------------------------------

    '---------------------------- E N D  N A M E ----------------------------------
----------------
    If rd.NodeType = XmlNodeType.Element And rd.Name =
"alternativeName" Then 'start alternative name
        Read_Next_Element(rd)
        'MsgBox("alternative Recommended" & rd.NodeType)
        If rd.NodeType = XmlNodeType.Element And rd.Name = "fullName"
Then
            Read_Next_Text(rd)
            If rd.NodeType = XmlNodeType.Text Then
                Dim comm As New SqlCommand
                comm.Connection = conn

                comm.CommandType = CommandType.StoredProcedure
                comm.CommandText = "UpdateAlternative"
                comm.Parameters.Add(New SqlParameter("@Alternative_Name",
SqlDbType.NVarChar))
                comm.Parameters.Add(New SqlParameter("@Accession_Number",
SqlDbType.NVarChar))
                comm.Parameters("@Alternative_Name").Value = CStr(rd.Value)
                comm.Parameters("@Accession_Number").Value =
CStr(Accession_Number)
                Execute_NonQuery(comm, "Alternaive name")
            End If

        End If
    End If 'end alternative name
    '---------------------------- E N D  ALTERNATIVE NAME ----------------
-----------------------------------
    '---------------------------- START __ G E N E ----------------------------
----------------
    If rd.NodeType = XmlNodeType.Element And rd.Name = "gene" Then
```

```
            'MsgBox("Start Gene")
            While rd.Read
              If rd.NodeType = XmlNodeType.EndElement And rd.Name = "gene"
Then
                Exit While
              End If
              If rd.NodeType = XmlNodeType.Element And rd.Name = "name"
Then
                If rd.GetAttribute("type") = "primary" Then
                  Read_Next_Text(rd)
                  If rd.NodeType = XmlNodeType.Text Then
                    Dim comm As New SqlCommand
                    comm.Connection = conn
                    comm.CommandType = CommandType.StoredProcedure
                    comm.CommandText = "UpdateGeneName"
                    comm.Parameters.Add(New SqlParameter("@Gene_Name",
SqlDbType.NVarChar))
                    comm.Parameters.Add(New
SqlParameter("@Accession_Number", SqlDbType.NVarChar))
                    comm.Parameters("@Gene_Name").Value = CStr(rd.Value)
                    comm.Parameters("@Accession_Number").Value =
CStr(Accession_Number)
                    Execute_NonQuery(comm, " Gene")
                  End If
                End If
                '----------
                If rd.GetAttribute("type") = "synonym" Then
                  Read_Next_Text(rd)
                  If rd.NodeType = XmlNodeType.Text Then
                    Dim comm As New SqlCommand
                    comm.Connection = conn
                    comm.CommandType = CommandType.StoredProcedure
                    comm.CommandText = "InsertSynonmGeneName"
                    comm.Parameters.Add(New SqlParameter("@SynonmName",
SqlDbType.NVarChar))
                    comm.Parameters.Add(New
SqlParameter("@Accession_Number", SqlDbType.NVarChar))
                    comm.Parameters("@SynonmName").Value = CStr(rd.Value)
                    comm.Parameters("@Accession_Number").Value =
CStr(Accession_Number)
                    Execute_NonQuery(comm, "Gene")
                  End If
                End If


              End If
            End While
          End If 'end GENE
          '---------- END __ GENE ----------
----------
```

```
                comm.Parameters.Add(New SqlParameter("@Ontology_Type_ID",
    Type.Int))
                comm.Parameters.Add(New SqlParameter("@Text",
SqlDbType.NVarChar))
                comm.Parameters.Add(New SqlParameter("@Accession_Number",
SqlDbType.NVarChar))
                comm.Parameters.Add(New SqlParameter("@Source",
SqlDbType.NVarChar))
                comm.Parameters("@Text").Value = CStr(g_text)
                comm.Parameters("@Accession_Number").Value =
CStr(Accession_Number)
                comm.Parameters("@Source").Value = CStr("NULL")
                If g_type = "F" Then
                    comm.Parameters("@Ontology_Type_ID").Value = CInt(3)
```

```
                    comm.Parameters("@Begin_Position").Value = CInt(Beg_pos)
                    comm.Parameters("@End_Position").Value =
CInt(rd.GetAttribute("position"))
                    comm.Parameters("@Description").Value = CStr("feat_desc")
                    Execute_NonQuery(comm, "Feature")
                End If
            End If
        End If

    ElseIf feat_type.Contains("modified residue") Or
feat_type.Contains("lipid") Then
        Read_Next_Element(rd)
        If rd.NodeType = XmlNodeType.Element And rd.Name = "location"
Then
            Read_Next_Element(rd)
            If rd.NodeType = XmlNodeType.Element And rd.Name =
"position" Then
                'insert in feature table, type =2
                Dim comm As New SqlCommand
                comm.Connection = conn
                comm.CommandType = CommandType.StoredProcedure
                comm.CommandText = "InsertFeatures"
                comm.Parameters.Add(New SqlParameter("@Feature_Key",
SqlDbType.NVarChar))
                comm.Parameters.Add(New
SqlParameter("@Features_Type_ID", SqlDbType.Int))
                comm.Parameters.Add(New SqlParameter("@Position",
SqlDbType.Int))
                comm.Parameters.Add(New SqlParameter("@Description",
SqlDbType.NVarChar))
                comm.Parameters.Add(New
SqlParameter("@Accession_Number", SqlDbType.NVarChar))
                comm.Parameters("@Feature_Key").Value = CStr(feat_type)
                comm.Parameters("@Accession_Number").Value =
CStr(Accession_Number)
                comm.Parameters("@Position").Value =
CInt(rd.GetAttribute("position"))
                comm.Parameters("@Features_Type_ID").Value = CInt(2)
                comm.Parameters("@Description").Value = CStr(feat_desc)
                Execute_NonQuery(comm, "Feature")
            End If
        End If

    ElseIf feat_type.Contains("site") Then
        Read_Next_Element(rd)
        If rd.NodeType = XmlNodeType.Element And rd.Name = "location"
Then
            Read_Next_Element(rd)
            If rd.NodeType = XmlNodeType.Element And rd.Name =
"position" Then
```

```
                'insert in feature table, type =1
                Dim comm As New SqlCommand
                comm.Connection = conn
                comm.CommandType = CommandType.StoredProcedure
                comm.CommandText = "InsertFeatures"
                comm.Parameters.Add(New SqlParameter("@Feature_Key",
SqlDbType.NVarChar))
                comm.Parameters.Add(New
SqlParameter("@Features_Type_ID", SqlDbType.Int))
                comm.Parameters.Add(New SqlParameter("@Position",
SqlDbType.Int))
                comm.Parameters.Add(New SqlParameter("@Description",
SqlDbType.NVarChar))
                comm.Parameters.Add(New
SqlParameter("@Accession_Number", SqlDbType.NVarChar))
                comm.Parameters("@Feature_Key").Value = CStr(feat_type)
                comm.Parameters("@Accession_Number").Value =
CStr(Accession_Number)
                comm.Parameters("@Position").Value =
CInt(rd.GetAttribute("position"))
                comm.Parameters("@Features_Type_ID").Value = CInt(1)
                comm.Parameters("@Description").Value = CStr(feat_desc)
                Execute_NonQuery(comm, "Feature")
              End If
            End If


          End If
        End If
        '------------------ E N D Features -----------------------------

        '------------------ START DB Reactome ---------------------------------
        If rd.NodeType = XmlNodeType.Element And rd.Name = "dbReference"
And (rd.GetAttribute("type") = "Reactome" Or rd.GetAttribute("type") =
"Pathway_Interaction_DB" Or rd.GetAttribute("type") = "PhosphoSite" Or
rd.GetAttribute("type") = "STRING" Or rd.GetAttribute("type") = "IntAct") Then
          Dim comm As New SqlCommand
          comm.CommandType = CommandType.StoredProcedure
          comm.CommandText = "insert_DB"
          comm.Connection = conn
          comm.Parameters.Add(New SqlParameter("@Name",
SqlDbType.NVarChar))
          comm.Parameters.Add(New SqlParameter("@Key_",
SqlDbType.NVarChar))
          comm.Parameters.Add(New SqlParameter("@Desc",
SqlDbType.NVarChar))
          comm.Parameters.Add(New SqlParameter("@Type_id", SqlDbType.Int))
          comm.Parameters.Add(New SqlParameter("@Accession_Number",
SqlDbType.NVarChar))
          comm.Parameters("@Name").Value = CStr(rd.GetAttribute("type"))
```

```
        MsgBox(ex.Message)
      Finally
        CloseSqlConnection()
      End Try
    End Sub

    Public Sub Read_Next_Element(ByRef rd As XmlReader)
      While rd.Read
        If Not rd.Name = "" Then
          Exit While
        End If
      End While
    End Sub

    Public Sub Read_Next_Text(ByRef rd As XmlReader)
      While rd.Read
        If Not rd.Value = "" Then
          Exit While
        End If
      End While
    End Sub

    Public Sub Execute_NonQuery(ByRef command As SqlCommand, ByVal Type As
String)
      Try
        conn.Open()
        command.ExecuteNonQuery()
      Catch ex As Exception
        MsgBox(ex.Message & ", Error In : " & Type)
      Finally
        conn.Close()
      End Try
    End Sub
```

- Apoptosis, (21 September 2009). Retrieved from: http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/A/Apoptosis.html

- AU Luthi and SJ Martin, 2007, The CASBAH: a searchable database of caspase substrates. Retrieved from: http://www.nature.com/cdd/journal/v14/n4/pdf/4402103a.pdf

- Baculovirus-Host Interactions, (2007). Retrieved from: http://www.microbiologybytes.com/virology/kalmakoff/baculo/baculohostinteract.html

- Ben-dor shifra,orr erit, the applications of multiple alignment. Retrieved from: http://bioportal.weizmann.ac.il/course/introbioinfo/lecture8/motifs09.pdf

- Ben-Gal I, Shani A, Gohr A, Grau J, Arviv S, Shmilovici A, Posch S, Grosse I (2005), "Identification of Transcription Factor Binding Sites with Variable-order Bayesian Networks". Bioinformatics 21 (11): 2657-2666. doi:10.1093/bioinformatics/bti410.

- Caspase, 1997. Retrieved from: http://www.celldeath.de/encyclo/caspases/intro.htm#Table

- Cristina Pop and Guy S. Salvesen, 2009, Human Caspases: Activation, Specificity, and Regulation. Retrieved from: http://www.jbc.org/content/284/33/21777.long#corresp-1

- Dash Phil, apoptosis. Retrieved from: http://www.sgul.ac.uk/depts/immunology/~dash/apoptosis/disease.htm

- JC Timmer and GS Salvesen, 2007, Caspase substrates. Retrieved from: http://www.nature.com/cdd/journal/v14/n1/pdf/4402059a.pdf

- Jun Wang, Earl R. Stadtman, and P. Boon Chock. (2004) Molecular cloning and characterization of murine caspase-12 gene promoter; Hammou, Oubrahim. Retrieved from: http://www.ncbi.nlm.nih.gov/pubmed/15701691

- Kurt Brunner, 06-30-2007, 10:40 PM , What versions of internet explorer support Ajax. Retrieved from: http://forums.asp.net/p/1128039/1779978.aspx

- Microsoft, (20 November 2010) software price. Retrieved from: http://msdotnetsupport.blogspot.com/2008/06/what-is-price-of-visual-studio-2008-how.html

- Minimum system requirement. Retrieved from: http://www.esko.com/web/site.aspx?p=1291

- Molly Edmonds, 2010, what is apoptosis. Retrieved from: http://science.howstuffworks.com/environmental/life/cellular-microscopic/apoptosis.htm

- Pascale Gaudet1 et al, 2010, Towards BioDBcore: a community-defined information specification for biological databases. Retrieved from: http://nar.oxfordjournals.org/content/early/2010/11/17/nar.gkq1173.abstract

- Sepp Hochreiter, Martin Heusel and Klaus Obermayer2, (May 1, 2007) Fast model-based protein homology detection without alignment, Vol. 23 no. 14 2007, pages 1728–1736.

- Software cheapest, (20 November 2010) software price. Retrieved from: http://softwarecheapest.com/products/Microsoft-SQL-Server-2008-Standard-Edition.html

- Steven Henikoff1 and Jorja G. Henikoff, 1994, Position-based sequence weights. Retrieved from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.106.3439&rep=rep1&type=pdf

- Substrate of caspase. Retrieved from: http://www.celldeath.de/encyclo/caspases/substrat.htm

- Taylor,C.F., Field,D., Sansone,S.A., Aerts,J., Apweiler,R., Ashburner,M., Ball,C.A., Binz,P.A., Bogue,M., Booth,T. et al.(2008) Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project.Nat. Biotechnol., 26, 889–896. Retrieved from: http://www.ncbi.nlm.nih.gov/pubmed/18688244

- Zteach software, software price. Retrieved from: http://www.ztechsoftware.com/flash/65018738.html