# Admission Seeker

## A web based educational seeking service

## [Information Retrieval and Presentation Module]


By

**Mohammad Ali**

This report is submitted as part of the requirements for the degree of M.Sc. (Computer Sciences) to the Department of Computer Science, Quaid-i-Azam University, Islamabad

**February 2009**

QUAID-I-AZAM UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Dated: May 30, 2009

## FINAL APPROVAL

This is to certify that we have read the project report submitted by Mr. Muhammad Ali and it is our judgment that this report is of sufficient standard to warrant its acceptance by the Quaid-i-Azam University, Islambad for the degree of the Master of Science in Computer Science.

**COMMITTEE:**

1. **External Examiner**

   Prof. Dr. Abdul Qadir
   Dean, Faculty of Engineering & Science Applied Sciences
   Muhammad Ali Jinnah University (MAJU)
   74-E, Blue Area, Jinnah Avenue
   Islamabad.

2. **Supervisor**

   Dr. Muhammad Shuaib Karim
   Assistant Professor
   Deptt. of Computer Science
   Quaid-i-Azam University
   Islamabad.

3. **Chairman**

   Dr. Muhammad Afzal Bhatti
   Professor
   Deptt. of Computer Science
   Quaid-i-Azam University
   Islamabad.

# Dedicated to...

*ISO Pakistan*

*All My Teachers*

*My Parents*

*and All My Friends*

# Acknowledgement

I would like to extend my humble gratitude to Almighty Allah who endowed me potential and ability to complete this project. During my master program I received help, support and encouragement from several people. It is really a pleasure for me to acknowledge this cooperation.

First and foremost my **Parents** and my other **Family Members** and **Relatives** who practically freed me from all responsibilities and who constantly prayed for me through out my academic career.

I am greatly thankful to **Dr. Afzal Bhatti,** Chairman Department of Computer Science Quaid-i-Azam University whose guidelines during the final semester are really great and supplement for my future life.

I am greatly thankful to internal supervisor **Dr. M. Shuaib Karim** for his valuable suggestions and constant encouragement to complete this Project. He was really there to help me whatever he can do. I greatly appreciate his caring attitude towards me. Whenever I went to him for some help, he gave me the time even when he was busy.

**Once again I will say Sir Thank You for your sincere cooperation.**

I am thankful to all my Department Lecturers specially **Dr. Onaiza Maqbool, Mr. Mohammad Naqi, Ms. Memoona Afsheen and Ms. Muddassira Arshad** who had taught me many subjects in my degree.

Last but not the least I am thankful to my group member **Muhammad Anis Ahmed** for his valuable help and support during the project. Also I am thankful to all my friends especially **Saleem Ullah, Syed Imran Shah, Muhammad Asghar Khan, Muhammad Farooq** and **Sherzad Ali** and all my colleagues who remained kind throughout my M.Sc Career.

**Mohammad Ali.**

## Abstract

Admission Seeking on the web takes a lot of time of end users who are surfing here and there for their information of interest such as admission criteria, degree and major information etc. Different Universities represent information of interest in different forms and also at different levels of their hyperspace hierarchy. End users sometimes get frustrated because they may not find their information of interest where they expect it to be. Each University presents the same set of information in different forms and shapes.

There is a need to gather those diverse set information onto a single platform and present to end users with easy search options so as to keep the cognitive stress of end user low. This service would be very beneficial for the students seeking admission in various universities, as well as for their parents who are remotely located in different parts of the country.

# Table of Contents

# List of figures

# List of Code

Chapter 1

# Introduction



# 1. Introduction

## 1.1. System Introduction

**Admission Seeker** is a web based admission seeking system that assists its end users (students, their guardians etc) on the web in finding their university of choice by knowing about their information of interests (Major and degrees, fee structures etc).

Admission Seeker performs the following tasks (in general) diagrammatic explanation of which is shown in system's workflow diagram given below:

- Crawls local web for Pakistani Universities
- Brings some or all web pages of their websites
- Extracts information of interests out of those web pages regarding each university
- Stores extracted information
- Satisfying end user queries on stored information
- Presenting end user's required (queried) information in meaningful form and with their URLs for the sake of confirmation and to know more details

Admission Seeker has two interfaces, details of which are given below:

- **Admission Seeker Server Interface**

  **Server Interface** is basically a desktop system with which administrator of Admission Seeker communicates to set crawling attributes such as crawl method, no of pages per crawl etc and also sets options of what to extract such as links, majors and degrees etc.

- **Admission Seeker Client Interface**

  **Client Interface** is a website where end user visits and submits his/her fixed and self constructed queries through certain HTML GUI constructs and gets back queries' results.

Diagram below shows flow of activities going on in the whole Admission Seeker System.

**Figure: 1.1. Admission Seeker: System Workflow Diagram**

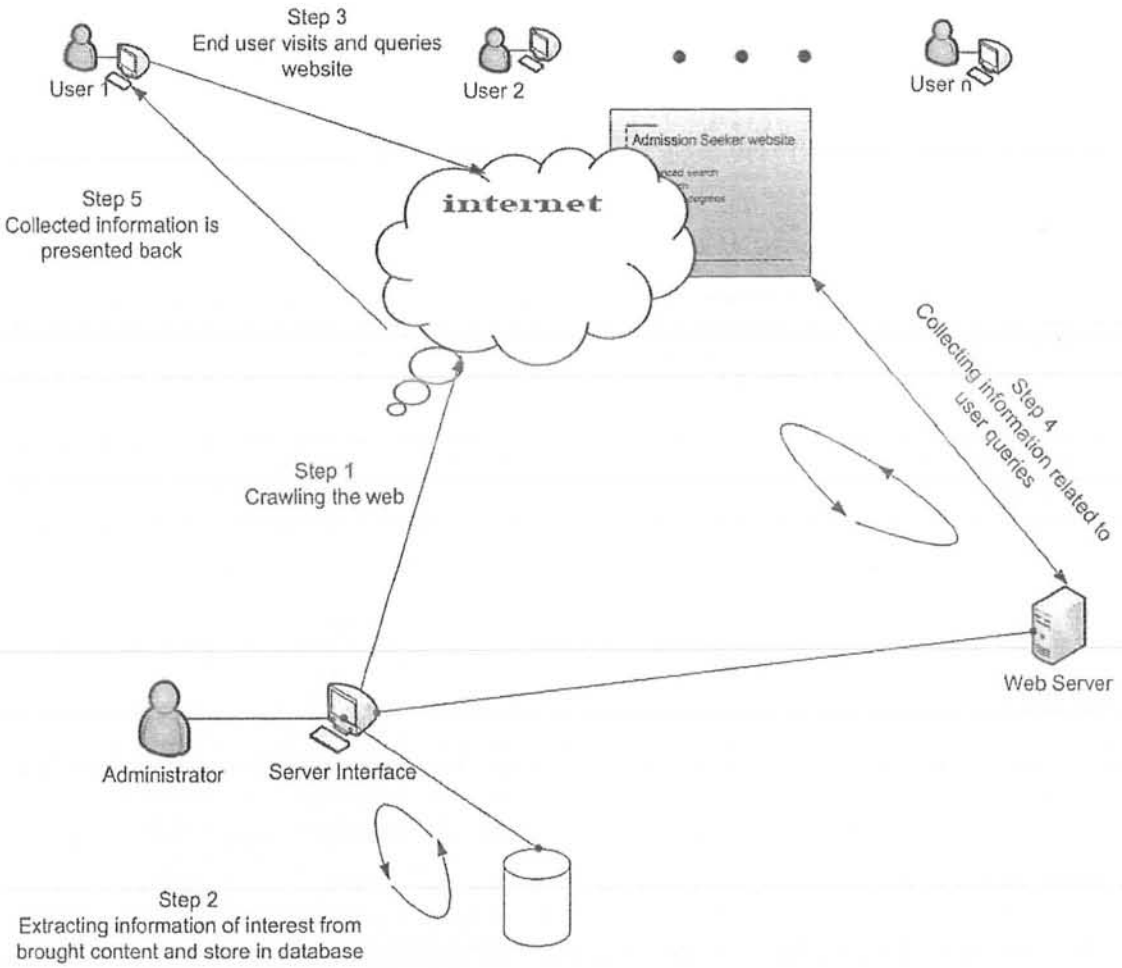



**Figure 1: Admission Seeker: System Workflow Diagram**

## 1.2. Problem Description

Seeking admissions on the web is a time consuming activity in fact, especially when it comes to credibility (university recognition from HEC) issues. End user has to browse a number of universities to find admission on the web. They normally look for majors taught there, degrees offered there, admission dates, fee structures, hostel facilities etc.

End users sometimes become frustrated while browsing certain universities websites to get information about certain university attributes in their mind. This thing increases user's cognitive stress and at the end users end up with no information in hand. Also end user may belong to a certain class of web surfers. He or she may belong to inexperienced user class. This class of user may suffer more than ordinary or advanced class of users. Search facilities on these websites are also unsuccessful to fill the needs of end users.

Each university puts information which is of interest to end users in a random or an ad hoc manner. Each university opens a new world for end users to explore while presenting the same set of information (normally degree, major, hostel etc) in different manners and sometimes with different shapes too.

There is a need to generalize this admission seeking process in case of **local web** (Pakistan Universities) and provide a general way of seeking admissions on the local web. The solution should end up with making admission seeking process easier from the perspective of different end users.

Chapter 2

# Project and Product

# 2. Project and Product

## 2.1. The Project

### 2.1.1 Scope

- Developing working software and submitting its source code
- Submitting thesis report

### 2.1.2. Objective

- To decrease admission seeking time by providing all extractable information from different universities' websites

## 2.2. The Product

### 2.2.1. Scope [My area of work is in bold letters]

- **Server Interface**
  - **Crawling Module**
    - Crawl multiple web pages of provided list of one or multiple universities
    - Store each university's web pages in its particular directory
    - Re crawl again on a pre set date to bring updated content
  - **Extraction Module**
    - Extract information of interests:
      - Majors and degrees
      - **Fee structure**
      - Hostel information
      - Important dates
      - All links

      from stored web pages of each university
  - **Storage Module**
    - **Associate each university's information with extracted information**
    - **Store above extracted information**

- Client Interface
  - Query and Aggregation Module
    - Construct raw queries from user requests through certain GUI constructs
    - Manipulate each given raw query and transform it into SQL Statement
    - Pass SQL to statements to database engine
    - Collect results from database engine and stores it into lists for presentation purposes

  - Presentation Module
    - Provide end users with the ease of querying admission seeker client interface by providing certain GUI constructs
    - Allow end users to construct their own queries
    - Provide simple navigation
    - Show results in a way so that subsequent queries could be launched on the result set

### 2.2.2. Objective

- To save end users' time by providing information from about 124 Pakistani Universities on a single platform (Admission Seeker)
- To allow end users view information in certain ways based on their queries
- To allow end users to construct their own queries with less efforts
- To extract maximum information out of stored web pages of universities to satisfy end user's information needs

## 2.3. Risks involved during the project

- Unexpected Project Demonstrations

  **Cause:** Timing and number of occurrences of **project demonstrations** are not planned since the start of the project session. It is normally done at runtime.

  **Impact:** Grades suffer

  **Plan for it:** Need to plan for it before date by keeping progress visible.

  **Occurrence:** It normally occurs 2 or 3 times during the whole project

- **Course assignments**

  **Cause:** Timing and number of occurrences of **course assignments** are not planned since the start of the project session. It is normally done at runtime.

  **Impact:** Project suffers

  **Plan for it:** Little compromise on both project and assignment

  **Occurrence:** It normally occurs 4 or 5 times (on the average) during the whole project

- Heavy load shedding

  **Cause:** Timing and duration of **load shedding** are not known in advance. It may go off at any instance of time in a day and may even go off for the whole 2 or 3 hours.

  **Impact:** Project suffers, courses suffer.

  **Plan for it:** We can't even plan for it.

  **Occurrence:** It is a continuous problem.

- **Miscalculated schedule**

  **Cause:** We often miscalculate project schedule and task division and also in assigning each task a particular duration. Miscalculations also occur in calculating task dependency

  **Impact:** Project schedule suffers and project gets late normally

  **Plan for it:** Need to plan for it before date by keeping progress visible.

  **Occurrence:** It normally occurs 2 or 3 times during the whole project

- **Misunderstood requirements**

**Cause:** Lack of expertise in understanding client's (supervisor here) requirements.

**Impact:** Project suffers, schedule suffers sometimes.

**Plan for it:** Constant visits.

**Occurrence:** 2 or 3 times in our case (as we'll do it in iterations)

- **Web nature**

  **Cause:** Rapid changes in web technologies and end user's requirements (query requirements and information needs here).

  **Impact:** Project scope changes at runtime.

  **Plan for it:** Using mature technologies (reusable software libraries that need small changes to be made in order to be customized according to our needs) and weekly planning.

  **Occurrence:** Occurs rapidly.

- **Network problem in laboratory**

  **Cause:** Virus attacks on laboratory computers on the network.

  **Impact:** Schedule overruns.

  **Plan for it:** Workaround: We bring our own PCs, crawl data locally for testing purposes.

  **Occurrence:** Unexpected.

## 2.4. Required Resources

### 2.4.1. Hardware Resources

A Personal Computer with:

- 3.06 GHz Clock Rate Microprocessor
- 512 MB RAM
- Network Interface Card, and a cross cable to connect PCs for testing purposes
- Web server for the sake of deployment on the web

### 2.4.2. Software Resources

- GATE (General Architecture for Text Engineering)

  **Explanation:** An open source java library for crawling and information extraction (explained in detail further in the text)

  **Purpose of use:** Crawling (Websphinx) and Extraction (ANNIE)

  **Reason of use:** Open source and java compatible

- **Netbeans 5.5.1 with Apache 5 built-in**

  **Explanation:** Java Program Development Environment

  **Purpose of use:** Coding

  **Reason of use:** Familiarity

- **MySQL 5**

  **Explanation:** Database Management Server

  **Purpose of use:** Storage and query handling on stored data

  **Reason of use:** Familiarity

- Java 5 (jdk 1.5)

  **Explanation:** Java Development Programming Kit having standard java libraries

  **Purpose of use:** Java Application development for Admission Seeker Server Interface Development

  **Reason of use:** Built-in in Netbeans 5.5.1

- JSP

  **Explanation:** Java Server Side Scripting Language

  **Purpose of use:** Writing server side scripts

  **Reason of use:** Java compatible

- **JavaScript**

  **Explanation:** Client side scripting language

  **Purpose of use:** Client side interactivity and form validation

  **Reason of use:** Almost every browser support

- **HTML**

  **Explanation:** Presentation Language for web page development

  **Purpose of use:** Information Presentation

  **Reason of use:** The only way to develop web pages

- **Microsoft Visio 2007**

  **Explanation:** Diagramming Tool

  **Purpose of use:** Drawing use case diagrams, sequence diagrams, class diagrams, data model diagrams etc

  **Reason of use:** Less restrictive and familiarity

- **Microsoft Word 2003**

  **Explanation:** Word processing tool

  **Purpose of use:** Report writing

  **Reason of use:** Familiarity

- **The internet**

  **Explanation:** Network of networks

  **Purpose of use:** Crawling web pages of Pakistani Universities, Literature study and reference help

  **Reason of use:** Literature Review, open source library downloading etc

### 2.4.3. Human Resources

- A team made up of two persons

## 2.5. Process Model Selection

Software Process Model is very much important when it comes to Software Development. It manages the flow of activities involved during software development. In our case it helps us a lot especially in case of above mentioned risks.

We have come across a phase of deciding which process model help us in this situation. Here some results are mentioned to justify how we reached a conclusion by comparing some process models against our requirements and some of the above mentioned risks.

| Our Requirements | Waterfall | RAD | Spiral | XP |
|---|---|---|---|---|
| Multiple Iterations | no | yes | yes | Yes |
| Less documentation | no | no | no | Yes |
| Small time frame | no | yes | no | Yes |
| Web nature | No | May be | can | Can |
| Team of two | No | No | No | Yes |
| Risk Analysis | Start only | Start of each iteration | Start of each iteration | Start of each iteration |
| Weekly planning | no | Based on duration of iteration | Based on duration of iteration | Based on duration of iteration |
| Random hours in a week | No | No | No | - 40 hours week so YES |

Scale

| |
|---|
| No = Rejected |
| Yes= Recommended |
| May be=not much surety |
| Can be= possibility |

Conclusion

Based on above table most of our requirements are satisfied by eXtreme Programming (XP) but a **'mix and match'** [SOM 2005] approach is always better and helps in certain situations so we have decided to put eXtreme Programming inline with Spiral Model.

## 2.6. Coarse Grained Plan

Admission Seeker Project goes through four phases. Each of these four phases involves a **Fine Grained Plan** which involves schedule and resource details and which is discussed in their dedicated chapters later in the text. Brief introduction of phases are the following:

- **Initiation**

  During initiation a number of activities happen that involve project ground work and concept development. Also a thorough analysis of Pakistani Universities is part of our initiation phase. A whole chapter is dedicated to initiation phase which comes later in the text. Each upcoming iteration involves plan, design, code, test and integration of a particular set of tasks.

- **First Iteration**

  First iteration involves plan, design, code, component tests and integration into modules

- **Second Iteration**

  Second iteration involves plan, design, code, component tests and integration into modules

- **Final Iteration**

  Final iteration involves plan, design, code, component tests, integration test and system test. Final iteration also produces a working product which is Admission Seeker Server and Client Part.

Figure: 2.1. Development Workflow Diagram

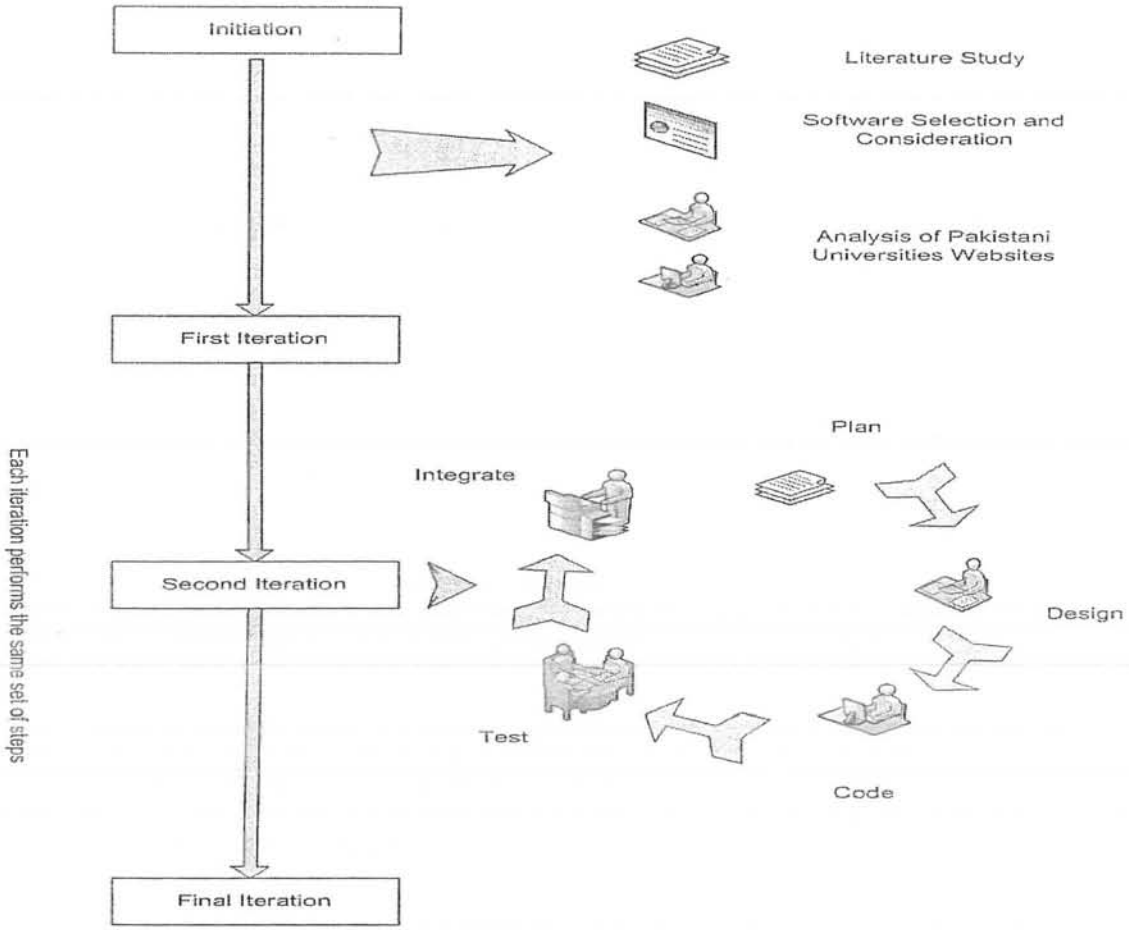



Figure 2: Development Workflow Diagram

Chapter 3

# Initiation

# 3. Initiation

## 3.1. Literature Study

We have studied literatures regarding crawling, information retrieval and process models. A thorough reference is provided in references chapter.

## 3.2. Software selection and customization

Software selection is an important decision we need to take before we start coding or even designing sometimes. We have considered a number of factors while selecting software for the sake of Admission Seeker Development out which some important factors are reusability, ease of use and familiarity. Bit details of each of these important factors are the following under the heading of software factors considerations.

### 3.2.1. Software factors considerations

- Reusability

  Software reusability helps a lot during development especially when it comes to standard code and quality code. A lot of efforts are required to write and test newly developed code so why not use reusable code and apply those efforts in other activities to save time and cost.

  Customizing reusable software requires some time to understand what is going on in the code and also some time in making it fit for the situation and purpose we are trying to use it for.

  We have made use of open source libraries and made them fit in to our environment and for our purpose and cause. I will explain how, further in the text.

- **Ease of use**

  Anything that is easy to use obviously saves time and effort required to learn it. We have tried to make use of those software that we thought are easy to use for us. Examples of such software were discussed in software resources in the previous chapter.

- **Familiarity**

  Familiarity is another factor we have considered during software selection especially in case of report writing tool and diagramming tool.

### 3.2.2. Open source libraries used

- **General Architecture for Text Engineering (GATE)** [GATE 2007]

  GATE is a comprehensive system in its own providing certain features through its resource plug-ins which are used to develop and deploy natural language processing software. Below are some features of GATE used for the purpose of Information Retrieval:

  - **Serial Data Store**

    GATE's Serial Data Store basically stores corpus document along with their associated document (s). Each serial data store has a unique ID given at the time when data store is created. All the corpus documents and their contained document inherit that particular data store's ID. Without that serial data stores ID corpus documents and documents cannot be decoded.

  - **Indexed Corpus Document**

    Indexed Corpus Document is a special kind of corpus document that holds documents need to be indexed and has some built-in features to create Index Manager.

- **Document**

In simplest term a GATE Document is a general entity that any document can be converted to such as PDF, DOC, RTF, HTML, XML etc. Contents in GATE Document is stored in encoded form, so decoding can only be done by following the same path such as

Serial Data Store → Corpus Document → Document

- **Apache Lucene 2.4.1** [MAN 2005]

Lucene is an Open Source API that provides the features of a full fledged search engine. Lucene is included in Apache's Jakarta Project and is currently being used by a number online service such as AOL Sports, Wikipedia etc. The purpose of using Lucene in Admission Seeker Project is to satisfy end user queries (keyword or phrase) on crawled web pages of certain universities' websites. In this project Lucene's Indexing and Searching features are used.

Lucene 1.3 comes with GATE providing indexing and searching facilities. Reasons of using Lucene 2.4.1 against Lucene 1.3 are:

- Lucene 2.4.1's Highlighter facility that highlights (formats) texts found during searching called 'Hits'
- As newer versions of Lucene come, their indexing feature gets better and better and also gets efficient too

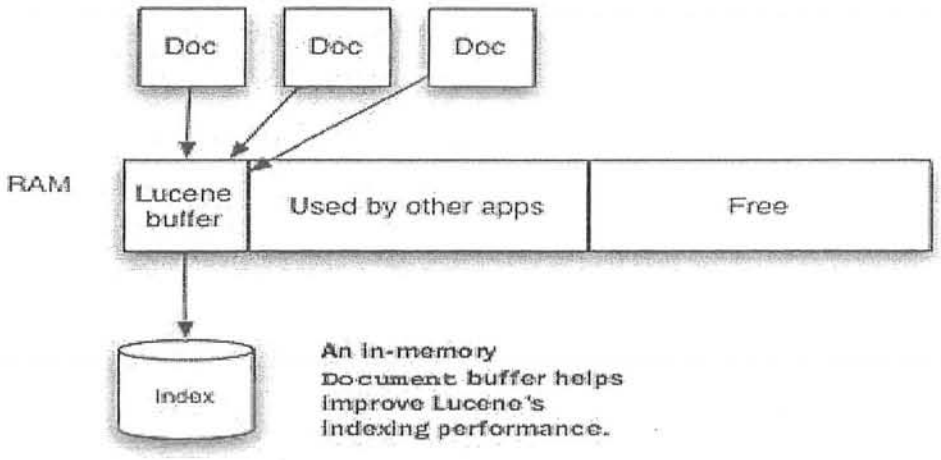
In Admission Seeker Project following Lucene 2.4.1 features are used to embed Information Retrieval requirements details of which come latter in the text:

- **Lucene Indexing**

Lucene Indexer indexes documents with respect of its certain features such as content, title etc. Features are user defined. Feature definition help user to query on each feature individually. Lucene

Indexer creates inverted index and generates 3 files _0.cfs, segments.gen and segments_2 with no extension.

**Figure: 3.1. An In- memory Document buffer helps improve Lucene's Indexing Performance**




- **Lucene Searching**

Lucene Searcher searches the index created by Lucene Indexer using different kind of queries, simplest of which is Term Query. Other kind queries are Phrase Query, Multi Term Query etc. In this project Term Query and Phrase Query are used.

### 3.3. Analysis of Pakistan Universities' websites

We have analyzed a number of Pakistan Universities' websites exploring how they represent certain information that we were interested in to extract out of their websites. Information representation on their websites is random, ad hoc and difficult to find out because in some cases information is at different levels of their hyperspace hierarchy. In some cases same information is found in different shapes too. There are more than 100 well reputed universities in Pakistan. They represent information in certain diverse manners.

Chapter# 3

**Initiation**

Before extraction we need to crawl and bring the whole website or some pages from the website. After that we extract information out of those web pages or the whole website.

### 3.4. User stories

There are two classes of Admission Seeker users. One sits on Admission Seeker Server Interface side called Administrator and other class is very much diverse, we call it End Users. End Users involve students, their guardians and many other people. The reason why I am discussing this here is to explain in detail their **user stories** (how they have intended to use the system or interact with the system).

Therefore I am explaining it from each class's perspectives:

[My area of work is in bold letters]

#### 3.4.1. Administrator: User Stories

- Set crawling attributes
- Add, delete degrees, majors to be extracted
- Extract degree, major, important dates and hostel information
- **Extract free structures**
- **Store extracted information**

#### 3.4.2. End User: User Stories

- **Give an advanced search query**
- **Get all links of one or multiple universities based on his/her entered keyword**
- **Search for major and degree separately and combined**
- **Get all universities' list**
- **Get university's details**
- **Get universities' list of one or more cities, one or more regions**

### 3.5. Refined set of tasks handled in upcoming iterations

Table below shows list of tasks handled by me during this project. Each iteration takes some tasks from this table and plans for them, designs them, codes them, tests them and if possible integrates them into a bigger one.

| Task Id | Task Name | Module |
|---|---|---|
| T1 | Get University list | Query and Presentation |
| T2 | Get raw query | Query |
| T3 | Transform raw query | Query |
| T4 | Search university by city | Query and Presentation |
| T5 | Search university by region | Query and Presentation |
| T6 | Search university by sector | Query and Presentation |
| T7 | Search degree or major | Query and Presentation |
| T8 | Search Major and Degree | Query and Presentation |
| T9 | Get University details | Query and Presentation |
| T10 | Link search | Query and Presentation |
| T11 | Advanced search | Query and Presentation |
| T12 | Write Index | Information Retrieval |
| T13 | Search Index | Information Retrieval, Query and Presentation |

Chapter 4

# First Iteration

# 4. First Iteration

## 4.1. Plan

| Task Id | Task Name | Module | Duration (days) [3 hours per day] |
|---|---|---|---|
| T1 | Get University list | Query and Presentation | 2 |
| T2 | Get raw query | Query | 4 |
| T3 | Transform raw query | Query | 5 |

## 4.2. Design

### 4.2.1. Use Case Description [LARMAN]

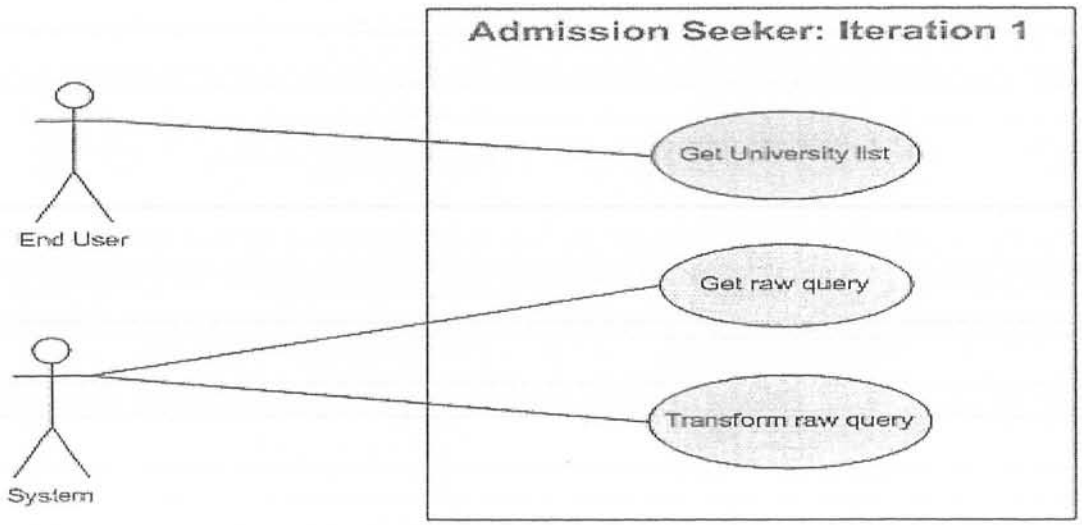
| Use Case ASUC1 : Get University list | |
|---|---|
| Primary Actor: | End User |
| Stakeholders and interests: | End User:<br>Wants list of all universities in Pakistan |
| Pre-Condition : | End user selects to view university list |
| Success Guarantee: | Universities list displayed |
| Main Success Scenario: | 1. System searches for list of all universities in Pakistan<br>2. System gets list of all universities in Pakistan<br>3. System display found results |
| Extensions(Alternative Flow): | 1*a. University list not found<br>(i) No information is displayed |
| Frequency of Occurrence: | Multiple times, as End user wants |

| Use Case ASUC2 : Get raw query | |
|---|---|
| Primary Actor: | System |
| Stakeholders and interests: | **End User:**<br>1. Wants to specify what to search<br>2. Wants to specify multiple search options like multiple university names, multiple cities, multiple regions etc |
| Pre-Condition : | End user visits website and select any kind of search available |
| Success Guarantee: | Raw query generated by the system |
| Main Success Scenario: | 1. End user specify one or more options<br>2. End user submits his/her options related to particular kind of search<br>3. System notes down options<br>4. System associates entity names with options<br>5. System generates result part of raw query<br>6. System generates condition part of raw query |
| Extensions(Alternative Flow): | 1*a. **End user specify no option**<br>(i) No raw query is generated |
| Frequency of Occurrence: | Multiple times, as End user wants |

| Use Case ASUC3 : Transform raw query | |
|---|---|
| Primary Actor: | System |
| Stakeholders and interests: | End User:<br>1. Wants results of his/her query |
| Pre-Condition : | Raw query is generated |
| Success Guarantee: | Raw query is transformed successfully |
| Main Success Scenario: | 1. System normalize result part by separating options and associated entity names<br>2. System attaches some pre-defined keywords with the normalized result part<br>3. System appends condition part |
| Extensions(Alternative Flow): | |
| Frequency of Occurrence: | Multiple times, as raw query is available |

### 4.2.2. Use Case Diagram [LARMAN]

Figure: 4.1. Use Case Diagram. Admission Seeker: Iteration 1



### 4.2.3. Test Case Description [PRAG 2009]

According to user needs and nature of the System I am using Scenario Based Testing strategy to test above mentioned user based and system based scenarios.

| Test case attributes | Details |
|---|---|
| Test case # | TC1 |
| Test case title | Get University list |
| User requirement | **Get University list** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'University List' |
| Expected results | 1. List of all Pakistani Universities |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-1 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Advanced Search'<br>3. End user selects one or multiple cities<br>4. End user selects a region<br>5. End user selects a sector |

| | |
|---|---|
| | 6. End user selects one or more degree+major<br>7. End user submits advanced query |
| Expected results | 1. University name + URL list satisfying above mentioned cities, region, offering selected degree+major list having hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-2 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Advanced Search'<br>3. End user selects one or multiple cities<br>4. End user submits advanced query |
| Expected results | 1. University name + URL list in selected city or cities of Pakistan having hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-3 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Advanced Search' |

| | |
|---|---|
| | 3. End user selects no city<br>4. End user submits advanced query |
| Expected results | 1. University name + URL list in all stored cities of Pakistan having hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-4 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Advanced Search'<br>3. End user selects no city<br>4. End user selects one or multiple regions<br>5. End user submits advanced query |
| Expected results | 1. University name + URL list in selected region (s) of Pakistan having hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-5 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website |

| | |
|---|---|
| | 2. End user clicks link 'Advanced Search'<br>3. End user selects no city<br>4. End user selects no region<br>5. End user submits advanced query |
| Expected results | 1. University name + URL list in all cities and all regions of Pakistan having hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-6 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Advanced Search'<br>3. End user deselects hostel option<br>4. End user submits advanced query |
| Expected results | 1. University name + URL list all universities of Pakistan without considering hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-7 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |

| | |
|---|---|
| Steps | 1. End user visits website<br>2. End user clicks link 'Advanced Search'<br>3. End user selects one or multiple cities<br>4. End user selects a region<br>5. End user submits advanced query |
| Expected results | 1. University name + URL list of universities in those selected cities of selected region having hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-8 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Advanced Search'<br>3. End user selects one or multiple cities<br>4. End user selects a region<br>5. End user deselects hostel option<br>6. End user submits advanced query |
| Expected results | 1. University name + URL list of universities in those selected cities of selected region without considering hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-9 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Advanced Search'<br>3. End user selects one or multiple cities<br>4. End user selects a region<br>5. End user selects a sector<br>6. End user submits advanced query |
| Expected results | 1. University name + URL list of universities in selected list cities and selected region and sector having hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-10 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Advanced Search'<br>3. End user selects one or multiple cities<br>4. End user selects a region |

| | |
|---|---|
| | 5. End user selects a sector<br>6. End user deselects hostel option<br>7. End user submits advanced query |
| Expected results | 1. University name + URL list of universities in selected list cities and selected region and sector without considering hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-11 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Advanced Search'<br>3. End user selects one or multiple cities<br>4. End user selects a region<br>5. End user selects a sector<br>6. End user selects one or more degree+major<br>7. End user submits advanced query |
| Expected results | 1. University name + URL list of universities in selected cities, region, sector and offering selected dergree+major list having hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-12 |
| Test case title | Get raw query for advanced search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Advanced Search'<br>3. End user selects one or multiple cities<br>4. End user selects a region<br>5. End user selects a sector<br>6. End user selects one or more degree+major<br>7. End user deselects hostel option<br>7. End user submits advanced query |
| Expected results | 1. University name + URL list of universities in selected cities, region, sector and offering selected dergree+major list without considering hostel |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-13 |
| Test case title | Get raw query for Link Search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Link Search' |

| | |
|---|---|
| | 3. End user selects one or multiple universities<br>4. End user types in a keyword<br>5. End user submits Link search query |
| Expected results | 1. List of Selected universities with links matching the entered keyword |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-14 |
| Test case title | Get raw query for Link Search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Link Search'<br>3. End user selects no university<br>4. End user types in a keyword<br>5. End user submits Link search query |
| Expected results | 1. List of all universities with links matching the entered keyword |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-15 |
| Test case title | Get raw query for Link Search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |

| | |
|---|---|
| Steps | 1. End user visits website<br>2. End user clicks link 'Link Search'<br>3. End user selects no university<br>4. End user enters no keyword<br>5. End user submits Link search query |
| Expected results | 1. List of all universities with all links |

| Test case attributes | Details |
|---|---|
| Test case # | TC2-16 |
| Test case title | Get raw query for Link Search |
| User requirement | **Get raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |
| Steps | 1. End user visits website<br>2. End user clicks link 'Link Search'<br>3. End user selects one or multiple universities<br>4. End user types no keyword<br>5. End user submits Link search query |
| Expected results | 1. List of Selected universities with all links |

| Test case attributes | Details |
|---|---|
| Test case # | TC3 |
| Test case title | Transform Raw query |
| User requirement | **Transform Raw query** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Query |

| | |
|---|---|
| Steps | 1. Raw query is passed to Query Manipulator<br>2. Query Manipulator's construct method is called |
| Expected results | 1. Valid SQL Query according to result and condition part of raw query passed to it |

### 4.2.4. Sequence Diagram [LARMAN]

**Figure: 4.2. Admission Seeker: Client Interface: Get University List**

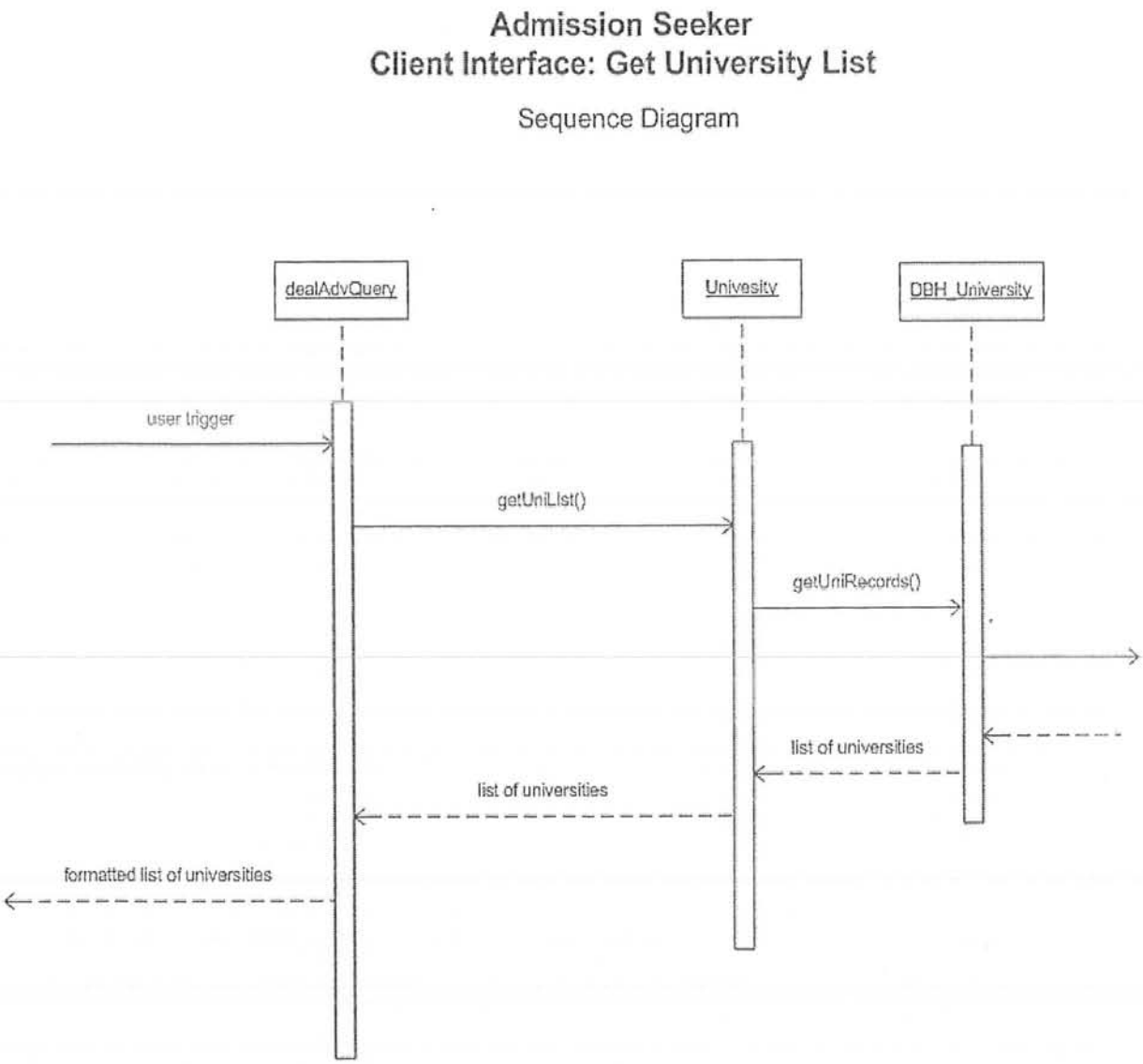
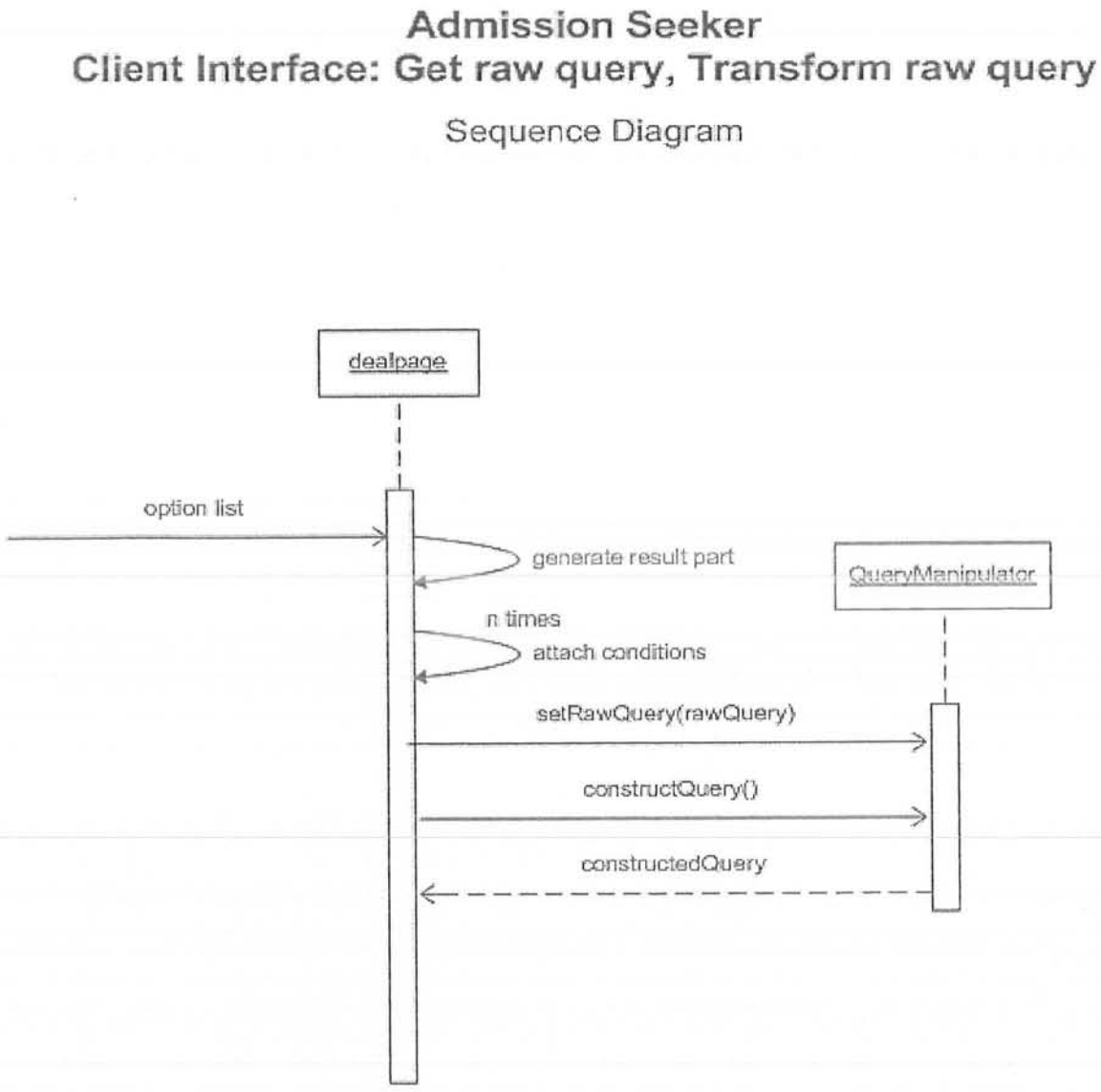


Figure: 4.3. Admission Seeker: Client Interface: Get raw query, Transform raw query



### 4.2.5. Class Diagram

Integrated class diagram is shown in Final Iteration Chapter.

## 4.3. Code

### 4.3.1. Examples of raw and transformed query

**Raw Query:** Raw Query has two parts:

- Result part may contain multiple table name-field name pair
- Condition part may contain multiple conditions

**Transformed Query:** Transformed is a SQL statement

**Example:**

*Unversity~uni_name,uni_url&ulink~lnk_literal,lnk_url@* is the result part of raw query and *university.uni_id = ulink_uni_id* is the condition part of raw query. Now after transformation, above example raw query is changed to *select university.uni_name,university.uni_url,ulink.lnk_literal,ulink.lnk_url where university.uni_id = ulink.uni_id* SQL Statement.

### 4.3.2. Code that transforms raw query to SQL Statement

**Code: 4.1. Code that transforms raw query to SQL Statement**

/* Code below is of method in **QueryManipulator** Class */

```
public String constructQuery(){
        constructedQuery = "select distinct ";

        String twoPart[] = rawQuery.split("[@]");

        //////////////  Dealing result part ////////////////////////

        String results[] = twoPart[0].split("[&]");
        String result[] = null;
        String fields[] = null;
        Vector<String> tableNames = new Vector();
        Vector<String> fieldNames = new Vector();
        System.out.println(rawQuery);
        for(int a=0;a<results.length;a++){
            result = results[a].split("[~]");
            tableNames.add(result[0]);
```

**First Iteration**

```
        //System.out.println(result[0]);
        if(result.length>1){
            fields = result[1].split("[,]");
            for(int b=0;b<fields.length;b++){
                fieldNames.add(result[0]+"."+fields[b]);
            }
        }
    }
    for(int a=0;a<fieldNames.size()-1;a++){
        constructedQuery += fieldNames.get(a) + ",";
    }
    constructedQuery += fieldNames.get(fieldNames.size()-1) + "from";

    for(int a=0;a<tableNames.size()-1;a++){
        constructedQuery += tableNames.get(a) + ",";
    }
    constructedQuery += tableNames.get(tableNames.size()-1) + " ";

    ///////////////// Dealing condition part //////////////////////

    if(twoPart.length>1){
        constructedQuery  += " where " + twoPart[1];
    }

    return constructedQuery;
}
```

## 4.4. Testing

### 4.4.1. Test execution and evaluation

| Test case Id | Name of Executer | Number of executions | Result |
|---|---|---|---|
| TC1 | M.Ali | 2 | Successful |
| TC2-1 | M.Ali | 2 | Successful |
| TC2-2 | M.Ali | 2 | Successful |
| TC2-3 | M.Ali | 2 | Successful |
| TC2-4 | M.Ali | 2 | Successful |
| TC2-5 | M.Ali | 2 | Successful |
| TC2-6 | M.Ali | 2 | Successful |
| TC2-7 | M.Ali | 2 | Successful |
| TC2-8 | M.Ali | 2 | Successful |
| TC2-9 | M.Ali | 2 | Successful |
| TC2-10 | M.Ali | 2 | Successful |
| TC2-11 | M.Ali | 2 | Successful |
| TC2-12 | M.Ali | 2 | Successful |
| TC2-13 | M.Ali | 2 | Successful |
| TC2-14 | M.Ali | 2 | Successful |
| TC2-15 | M.Ali | 2 | Successful |
| TC2-16 | M.Ali | 2 | Successful |
| TC3 | M.Ali | 2 | Successful |

## 4.5. Integration

Components produced during this iteration are made part of Admission Seeker Client Interface. Integration details are mentioned in Final Iteration Chapter.

Chapter 5

# Second Iteration

# 5. Second Iteration

## 5.1. Plan

| Task Id | Task Name | Module | Duration (days) [3 hours per day] |
|---|---|---|---|
| T4 | Search university by city | Query and Presentation | 2 |
| T5 | Search university by region | Query and Presentation | 2 |
| T6 | Search university by sector | Query and Presentation | 2 |
| T7 | Search major or degree | Query and Presentation | 3 |
| T8 | Search major+degree | Query and Presentation | 2 |
| T9 | Get University details | Query and Presentation | 5 |

## 5.2. Design

### 5.2.1. Use Case Description [LARMAN]

| Use Case ASUC4 : Search university by city | |
|---|---|
| Primary Actor: | End User |
| Stakeholders and interests: | End User:<br>Wants to search for university city wise |
| Pre-Condition : | End user visits city search page |
| Success Guarantee: | City search results displayed |
| Main Success Scenario: | 1. End user selects one or multiple cities<br>2. End user submits information<br>3. System generates a raw query based on user criteria<br>4. System refines raw query<br>5. System collects information related to user criteria<br>6. System display found results |

| | |
|---|---|
| Extensions(Alternative Flow): | 1*a. **End user required information not found**<br>(i) No information is displayed |
| **Frequency of Occurrence:** | Multiple times, as End user wants |

| **Use Case ASUC5 : Search university by region** | |
|---|---|
| Primary Actor: | End User |
| Stakeholders and interests: | **End User:**<br>Wants to search for university region wise |
| Pre-Condition : | End user visits region search page |
| Success Guarantee: | Region search results displayed |
| Main Success Scenario: | 1. End user selects a region<br>2. End user submits information<br>3. System generates a raw query based on user criteria<br>4. System refines raw query<br>5. System collects information related to user criteria<br>6. System display found results |
| Extensions(Alternative Flow): | 1*a. **End user required information not found**<br>(ii) No information is displayed |
| **Frequency of Occurrence:** | Multiple times, as End user wants |

| Use Case ASUC6 : Search university by sector | |
|---|---|
| Primary Actor: | End User |
| Stakeholders and interests: | End User:<br>Wants to search for university sector wise |
| Pre-Condition : | End user visits sector search page |
| Success Guarantee: | Sector search results displayed |
| Main Success Scenario: | 1. End user selects a sector<br>2. End user submits information<br>3. System generates a raw query based on user criteria<br>4. System refines raw query<br>5. System collects information related to user criteria<br>6. System display found results |
| Extensions(Alternative Flow): | 1*a. End user required information not found<br>(iii) No information is displayed |
| Frequency of Occurrence: | Multiple times, as End user wants |

| Use Case ASUC7 : Search major or degree | |
|---|---|
| Primary Actor: | End User |
| Stakeholders and interests: | End User:<br>Wants to search about major or degree offered in a single or multiple universities |
| Pre-Condition : | End user visits major or degree search page |
| Success Guarantee: | Major or degree Search results displayed |
| Main Success Scenario: | 1. End user provides major or degree to be searched<br>2. End user selects one or multiple major or degree |

| | |
|---|---|
| | 3. End user submits information<br>4. System generates a raw query based on user criteria<br>5. System refines raw query<br>6. System collects information related to user criteria<br>7. System display found results |
| Extensions(Alternative Flow): | 1*a. **End user required information not found**<br>(iv) No information is displayed |
| **Frequency of Occurrence:** | Multiple times, as End user wants |

| **Use Case ASUC8 : Search major+degree** | |
|---|---|
| Primary Actor: | End User |
| Stakeholders and interests: | **End User:**<br>Wants to search about major and degree offered in a single or multiple universities |
| Pre-Condition : | End user visits major and degree search page |
| Success Guarantee: | Major and degree Search results displayed |
| Main Success Scenario: | 1. End user provides major and degree to be searched<br>2. End user selects one or multiple major and degree<br>3. End user submits information<br>4. System generates a raw query based on user criteria<br>5. System refines raw query<br>6. System collects information related to user criteria<br>7. System display found results |
| Extensions(Alternative Flow): | 1*a. **End user required information not found**<br>(v) No information is displayed |
| **Frequency of Occurrence:** | Multiple times, as End user wants |

| Use Case ASUC9 : Get University Details | |
|---|---|
| Primary Actor: | End User |
| Stakeholders and interests: | **End User:**<br>Wants:<br>1. City, region and sector information of selected university<br>2. To know important dates of selected university<br>3. Fee Structure of selected university<br>4. Hostel information of selected university |
| Pre-Condition : | End user selects a university |
| Success Guarantee: | University details shown |
| Main Success Scenario: | 1. System generates a raw query to get selected university's city, region and sector information<br>2. System refines raw query<br>3. System collects selected university's city, region and sector information<br>4. System generates a raw query to get selected university's important dates<br>5. System refines raw query<br>6. System collects selected university's important dates<br>7. System generates a raw query to get selected university's fee structure<br>8. System refines raw query<br>9. System collects selected university's fee structure<br>10. System generates a raw query to get selected university's |

| | |
|---|---|
| | hostel information<br>11. System refines raw query<br>12. System collects selected university's hostel information<br>13. System display found results |
| Extensions(Alternative Flow): | 3*a. **Selected university's city, region, sector information not found**<br>(i) No information is displayed<br>6*a. **Selected university's important dates not found**<br>(i) No information is displayed<br>9*a. **Selected university's fee structure not found**<br>(i) No information is displayed<br>12*a. **Selected university's hostel information not found**<br>(i) No information is displayed |
| Frequency of Occurrence: | Multiple times, as End user wants |

### 5.2.2. Use Case Diagram [LARMAN]

Figure: 5.1 Use Case Diagram: Admission Seeker: Iteration2

### 5.2.3. Test Case Description [PRAG 2009]

According to user needs and nature of the System I am using Scenario Based Testing strategy to test above mentioned user based and system based scenarios.

| Test case attributes | Details |
|---|---|
| Test case # | TC4-1 |
| Test case title | Search university by city |
| User requirement | **Search university by city** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'City'<br>3. End user selects one or more cities<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities in selected cities |

| Test case attributes | Details |
|---|---|
| Test case # | TC4-2 |
| Test case title | Search university by city |
| User requirement | **Search university by city** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'City'<br>3. End user selects no city<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities |

**Second Iteration**

| Test case attributes | Details |
|---|---|
| Test case # | TC5-1 |
| Test case title | Search university by region |
| User requirement | **Search university by region** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'Region'<br>3. End user selects a region<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities in selected region |

| Test case attributes | Details |
|---|---|
| Test case # | TC5-2 |
| Test case title | Search university by region |
| User requirement | **Search university by region** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'Region'<br>3. End user selects no region<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities |

| Test case attributes | Details |
|---|---|
| Test case # | TC6-1 |
| Test case title | Search university by sector |
| User requirement | **Search university by sector** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'Sector'<br>3. End user selects a sector<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities in selected sector |

| Test case attributes | Details |
|---|---|
| Test case # | TC6-2 |
| Test case title | Search university by sector |
| User requirement | **Search university by sector** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'Sector'<br>3. End user selects no sector<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities |

**Second Iteration**

| Test case attributes | Details |
|---|---|
| Test case # | TC7-1 |
| Test case title | Search university by major |
| User requirement | **Search university by major or degree** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'Major'<br>3. End user selects one or multiple majors<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities offering selected major |

| Test case attributes | Details |
|---|---|
| Test case # | TC7-2 |
| Test case title | Search university by major |
| User requirement | **Search university by major or degree** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'Major'<br>3. End user selects no major<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities offering all stored majors |

| Test case attributes | Details |
|---|---|
| Test case # | TC7-3 |
| Test case title | Search university by degree |
| User requirement | **Search university by major or degree** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'Degree'<br>3. End user selects one or multiple degree<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities offering selected degree |

| Test case attributes | Details |
|---|---|
| Test case # | TC7-4 |
| Test case title | Search university by degree |
| User requirement | **Search university by major or degree** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'Degree'<br>3. End user selects no degree<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities offering all stored degrees |

| Test case attributes | Details |
|---|---|
| Test case # | TC8-1 |
| Test case title | Search university by major+degree |
| User requirement | **Search university by major+degree** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'Major and Degree'<br>3. End user selects one or multiple major+degree<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities offering selected major+degree |

| Test case attributes | Details |
|---|---|
| Test case # | TC8-2 |
| Test case title | Search university by major+degree |
| User requirement | **Search university by major+degree** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. End user clicks link 'Major and Degree'<br>3. End user selects no major+degree<br>4. End user submits query |
| Expected results | 1. List of all Pakistani Universities offering all stored major+degree |

| Test case attributes | Details |
|---|---|
| Test case # | TC9 |
| Test case title | Get University details |
| User requirement | **Get University details** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Search |
| Steps | 1. End user visits website<br>2. After getting results in the form of university name and URL end user clicks on university name |
| Expected results | 1. Details of selected university is listed |

### 5.2.4. Sequence Diagram [LARMAN]

Figure: 5.2. Sequence Diagram: Admission Seeker: Client Interface Search University by City

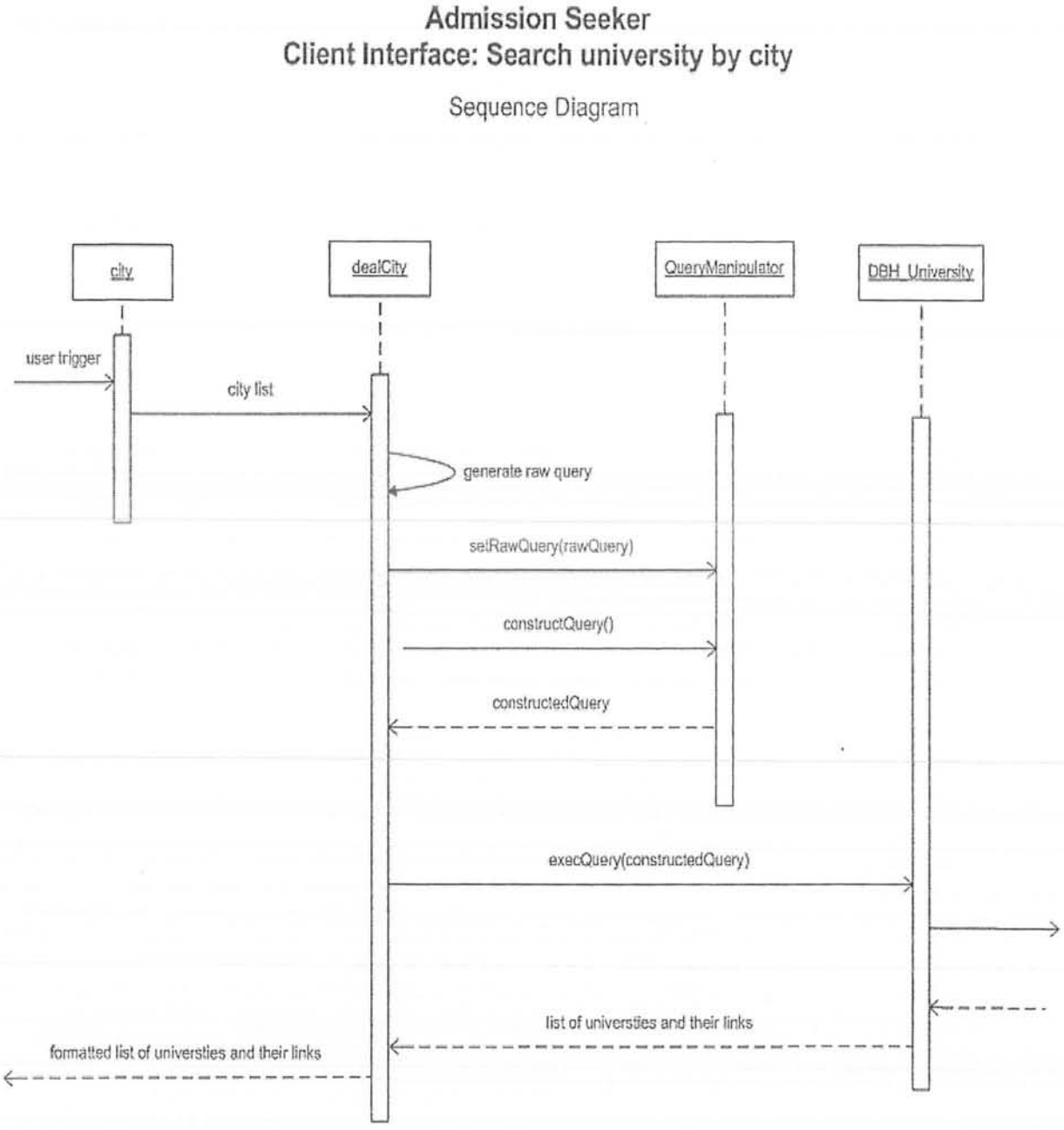


Figure: 5.3. Sequence Diagram: Admission Seeker: Client Interface Search University by Region

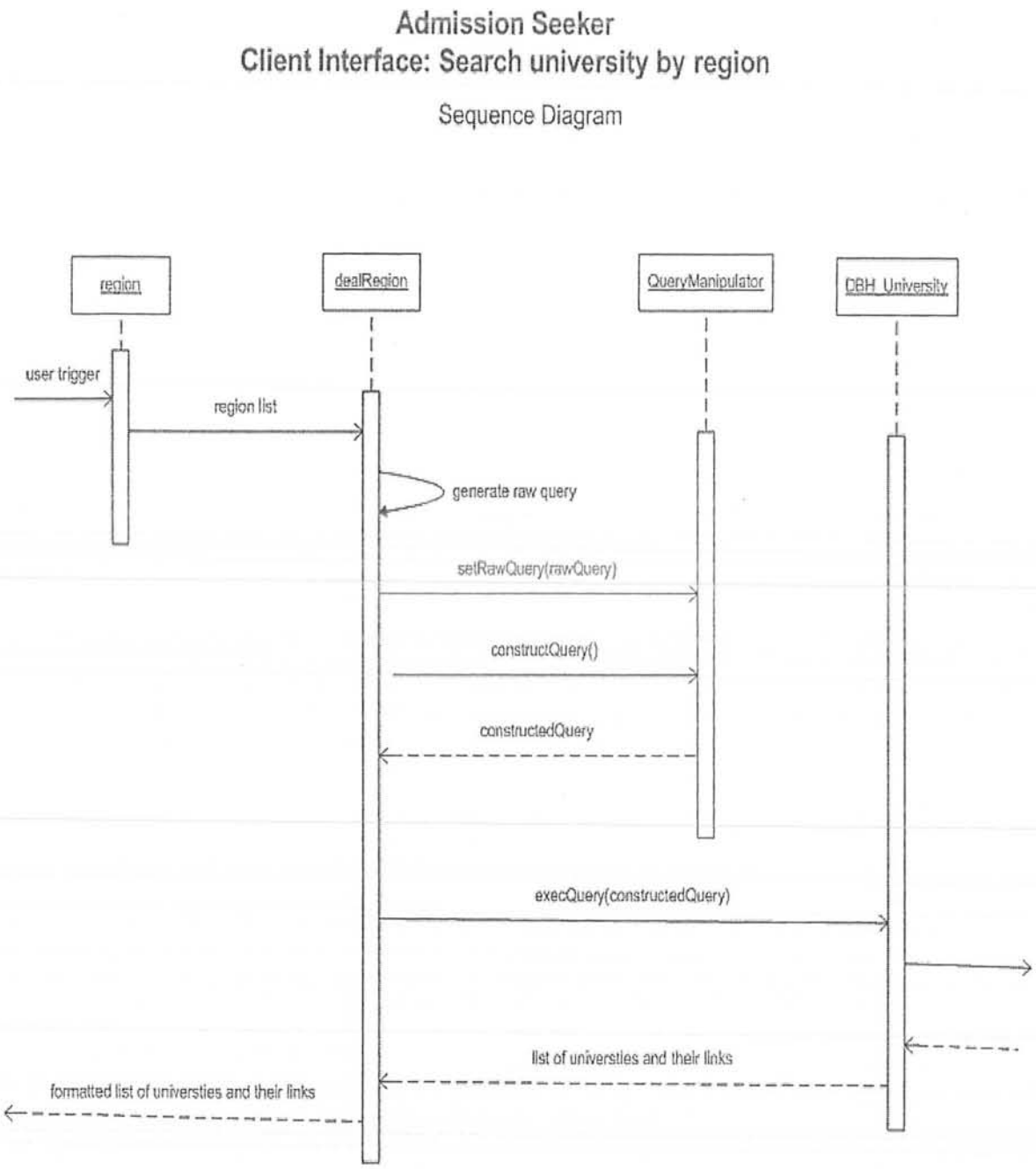


Figure: 5.4.Sequence Diagram: Admission Seeker: Client Interface Search University by sector

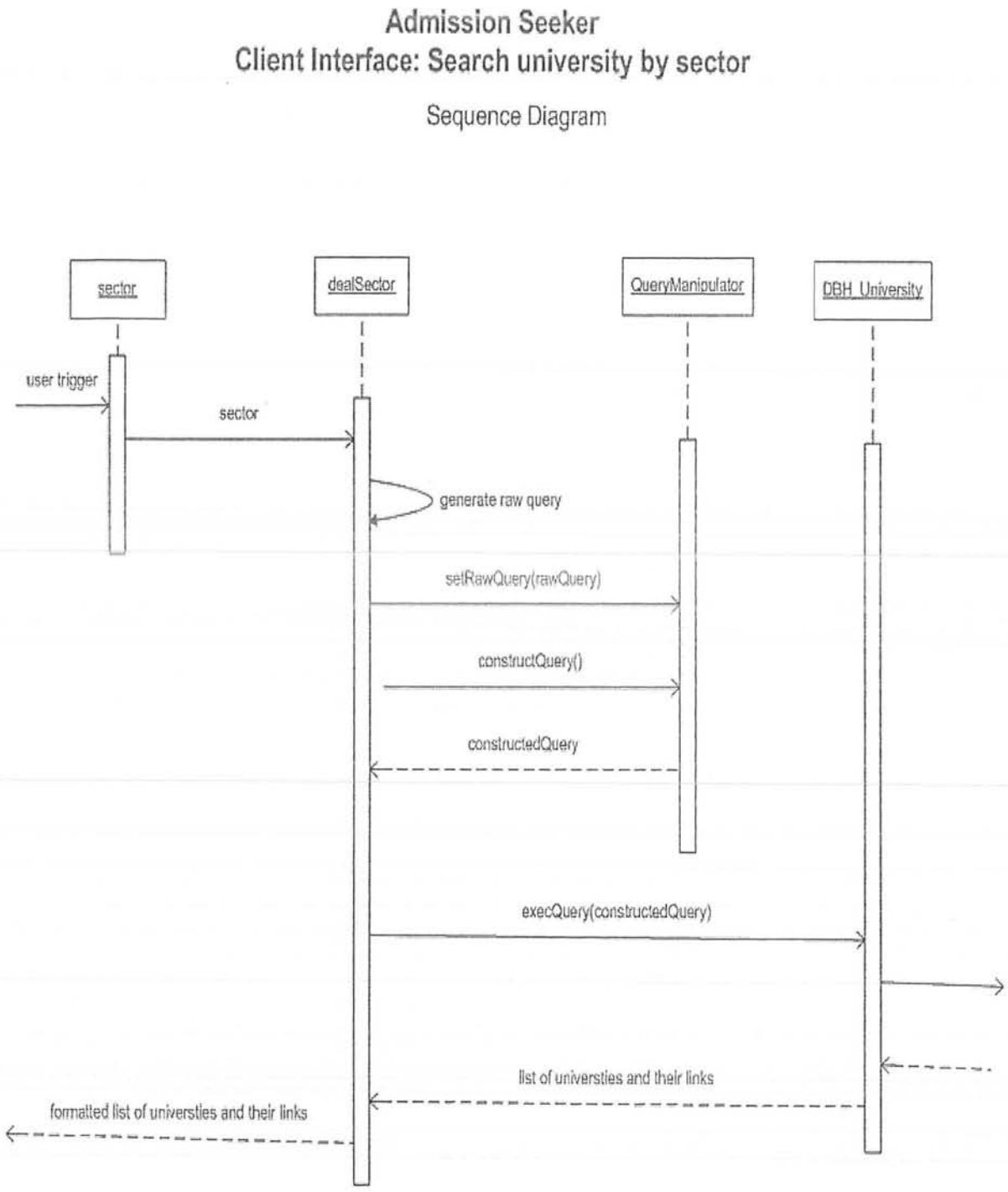


Figure: 5.5.Sequence Diagram: Admission Seeker: Client Interface Search major or Degree

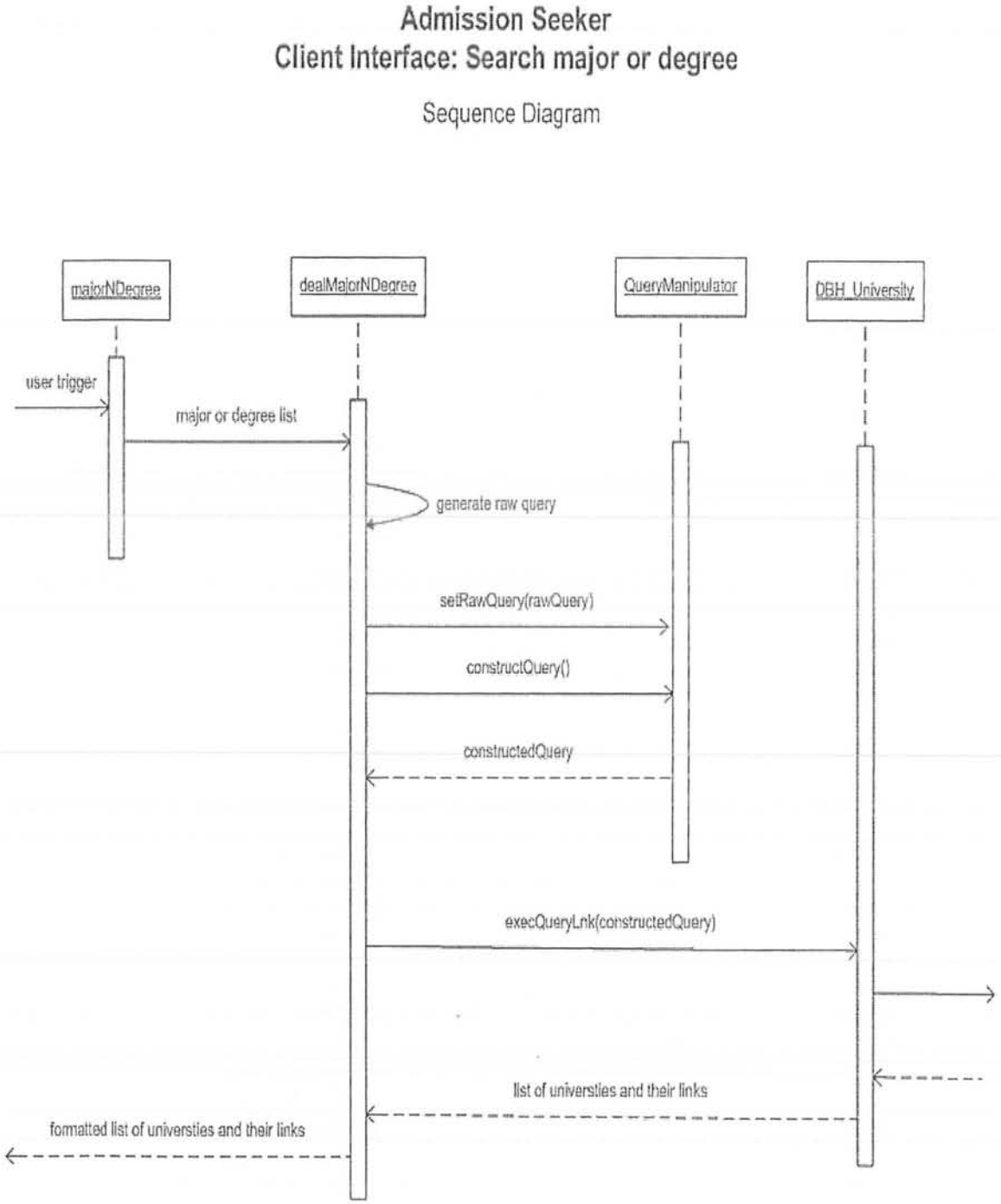


Figure: 5.6.Sequence Diagram: Admission Seeker: Client Interface Search major and Degree

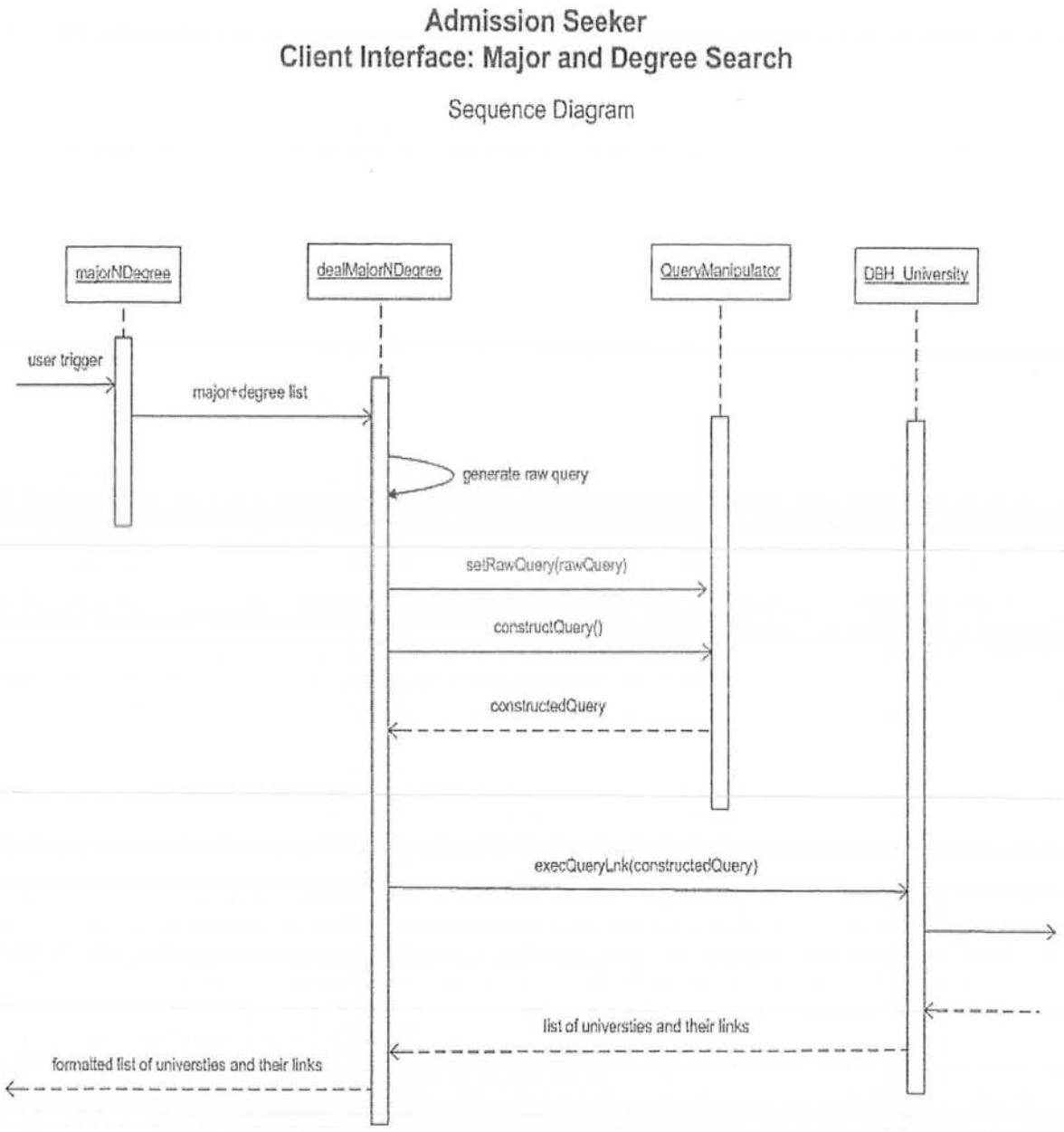
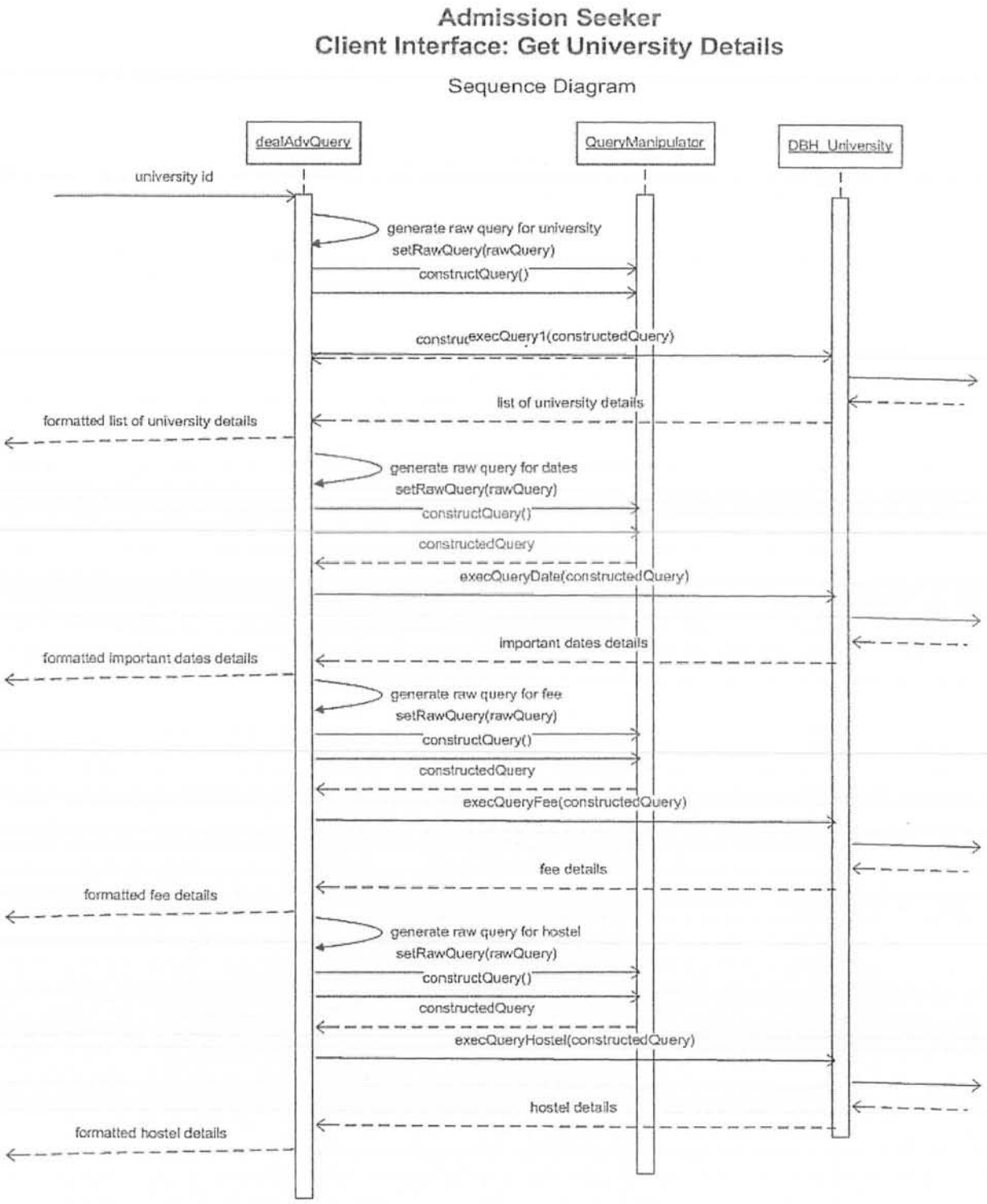


Figure: 5.7.Sequence Diagram: Admission Seeker: Client Interface: Get Universities Details



### 5.2.5. Class Diagram

Integrated class diagram is shown in Final Iteration Chapter.

## 5.3. Code

Tasks handled during this iteration have been made part of the system through certain JSP pages. Details of code written in these pages increase the size of this document, so it is avoided.

## 5.4. Testing

### 5.4.1. Test execution and evaluation

| Test case Id | Name of Executer | Number of executions | Result |
|---|---|---|---|
| TC4-1 | M.Ali | 2 | Successful |
| TC4-2 | M.Ali | 2 | Successful |
| TC5-1 | M.Ali | 2 | Successful |
| TC5-2 | M.Ali | 2 | Successful |
| TC6-1 | M.Ali | 2 | Successful |
| TC6-2 | M.Ali | 2 | Successful |
| TC7-1 | M.Ali | 2 | Successful |
| TC7-2 | M.Ali | 2 | Successful |
| TC7-3 | M.Ali | 2 | Successful |
| TC7-4 | M.Ali | 2 | Successful |
| TC8-1 | M.Ali | 2 | Successful |
| TC8-2 | M.Ali | 2 | Successful |
| TC9 | M.Ali | 2 | Successful |

## 5.5. Integration

Components produced during this iteration are made part of Admission Seeker Client Interface. Integration details are mentioned in Final Iteration Chapter.

Chapter 6

# Final Iteration

# 6. Final Iteration

## 6.1. Plan

| Task Id | Task Name | Module | Duration (days) [3 hours per day] |
|---|---|---|---|
| T10 | Link search | Query and Presentation | 6 |
| T11 | Advanced search | Query and Presentation | 9 |
| T12 | Write Index | Information Retrieval | 10 |
| T13 | Search Index | Information Retrieval, Query and Presentation | 9 |

## 6.2. Design

### 6.2.1. Use Case Description [LARMAN]

| Use Case ASUC10 : Link Search | |
|---|---|
| Primary Actor: | End User |
| Stakeholders and interests: | End User:<br>Wants to search a link on a single university's pages or multiple universities' pages |
| Pre-Condition : | End user visits link search page |
| Success Guarantee: | Link Search results displayed |
| Main Success Scenario: | 1. End user provides link literal to be searched<br>2. End user selects one or multiple universities<br>3. End user submits information<br>4. System generates a raw query based on user criteria<br>5. System refines raw query<br>6. System collects information related to user criteria<br>7. System display found results |

| | |
|---|---|
| Extensions(Alternative Flow): | 1*a. **End user required information not found**<br>(i) No information is displayed |
| Frequency of Occurrence: | Multiple times, as End user wants |

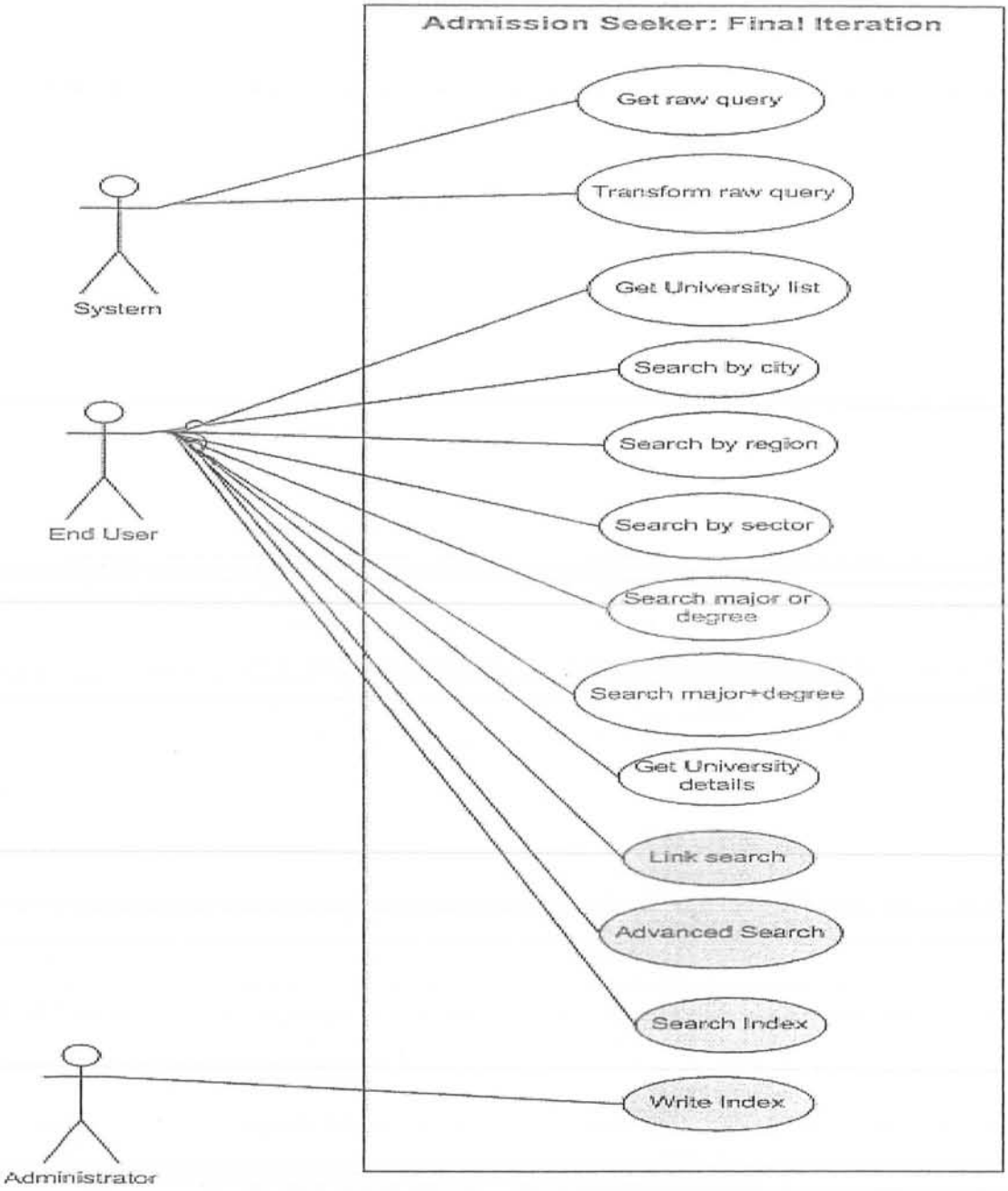
| **Use Case ASUC11 : Advanced Search** | |
|---|---|
| Primary Actor: | End User |
| Stakeholders and interests: | **End User:**<br>Wants to restrict searched information by applying a number of user selected criteria |
| Pre-Condition : | End user visits advanced search page |
| Success Guarantee: | Advanced Search results displayed |
| Main Success Scenario: | 1. End user selects one or multiple cities<br>2. End user selects one or multiple regions<br>3. End user selects one or multiple sectors<br>4. End user selects one or multiple degree+major<br>5. End user selects whether he or she wants hostel<br>6. End user submits search criteria<br>7. System generates a raw query based on user criteria<br>8. System refines raw query<br>9. System collects information related to user criteria<br>10. System display found results |
| Extensions(Alternative Flow): | 1*a. **End user required information not found**<br>(i) No information is displayed |
| Frequency of Occurrence: | Multiple times, as End user wants |

| Use Case ASUC12 : Write Index | |
|---|---|
| Primary Actor: | Administrator |
| Stakeholders and interests: | End User:<br>Wants keyword or phrase query searching |
| Pre-Condition : | Administrator logs on to the system |
| Success Guarantee: | Index created and written |
| Main Success Scenario: | 1. Administrator selects directory where crawled data is placed<br>2. Administrator selects directory where index needs to be created<br>3. Administrator selects to create index<br>4. System prepares all present documents for indexing<br>5. System adds fields to documents<br>6. System writes index into the index location |
| Extensions(Alternative Flow): | 1*a. **Crawled data is not present**<br>(i) Index is not created |
| Frequency of Occurrence: | Multiple times, as updated pages are crawled |

| Use Case ASUC13 : Search Index | |
|---|---|
| Primary Actor: | End User |
| Stakeholders and interests: | **End User:**<br>Wants keyword or phrase query searching |
| Pre-Condition : | End user visits keyword search page |
| Success Guarantee: | Keyword Search results displayed |
| Main Success Scenario: | 1. End user type in his/her query<br>2. End user selects to start search<br>3. System searches through index for matched terms<br>4. System ranks found results on the basis of frequency of terms matched<br>5. System lists found/matched results<br>6. System highlights found keywords in paragraphs/sentences<br>7. System displays the results |
| Extensions(Alternative Flow): | 1*a. **End user required information not found**<br>(ii) No information is displayed |
| Frequency of Occurrence: | Multiple times, as End user wants |

### 6.2.2. Use Case Diagram [LARMAN]

Figure: 6.1. Use Case Diagram: Admission Seeker: Final Iteration



### 6.2.3. Test Case Description [PRAG 2005]

According to user needs and nature of the System I am using Scenario Based Testing strategy to test above mentioned user based and system based scenarios.

| Test case attributes | Details |
|---|---|
| Test case # | TC12 |
| Test case title | Write index |
| User requirement | **Write index** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Information Retrieval |
| Steps | 1. Administrator selects crawled pages directory<br>2. Administrator selects index directory<br>3. Administrator selects to create index |
| Expected results | 1. Number of crawled pages matches number of documents in indexed corpus<br>2. Index created successfully in index directorys |

| Test case attributes | Details |
|---|---|
| Test case # | TC13 |
| Test case title | Search index |
| User requirement | **Search index** |
| Owner | M. Ali |
| Assignee | M. Ali |
| Group/ subgroup | Information Retrieval, Query and Presentation |
| Steps | 1. End user types in his/her keyword<br>2. End user selects to search |
| Expected results | 1. Number of results of found during index searching is equal to the number of results displayed |

Test cases of Tasks T10 and T11 are the same as Test cases of Task2 set.

### 6.2.4. Sequence Diagram [LARMAN]

**Figure: 6.2. Sequence Diagram: Admission Seeker: End User side: Link Search**

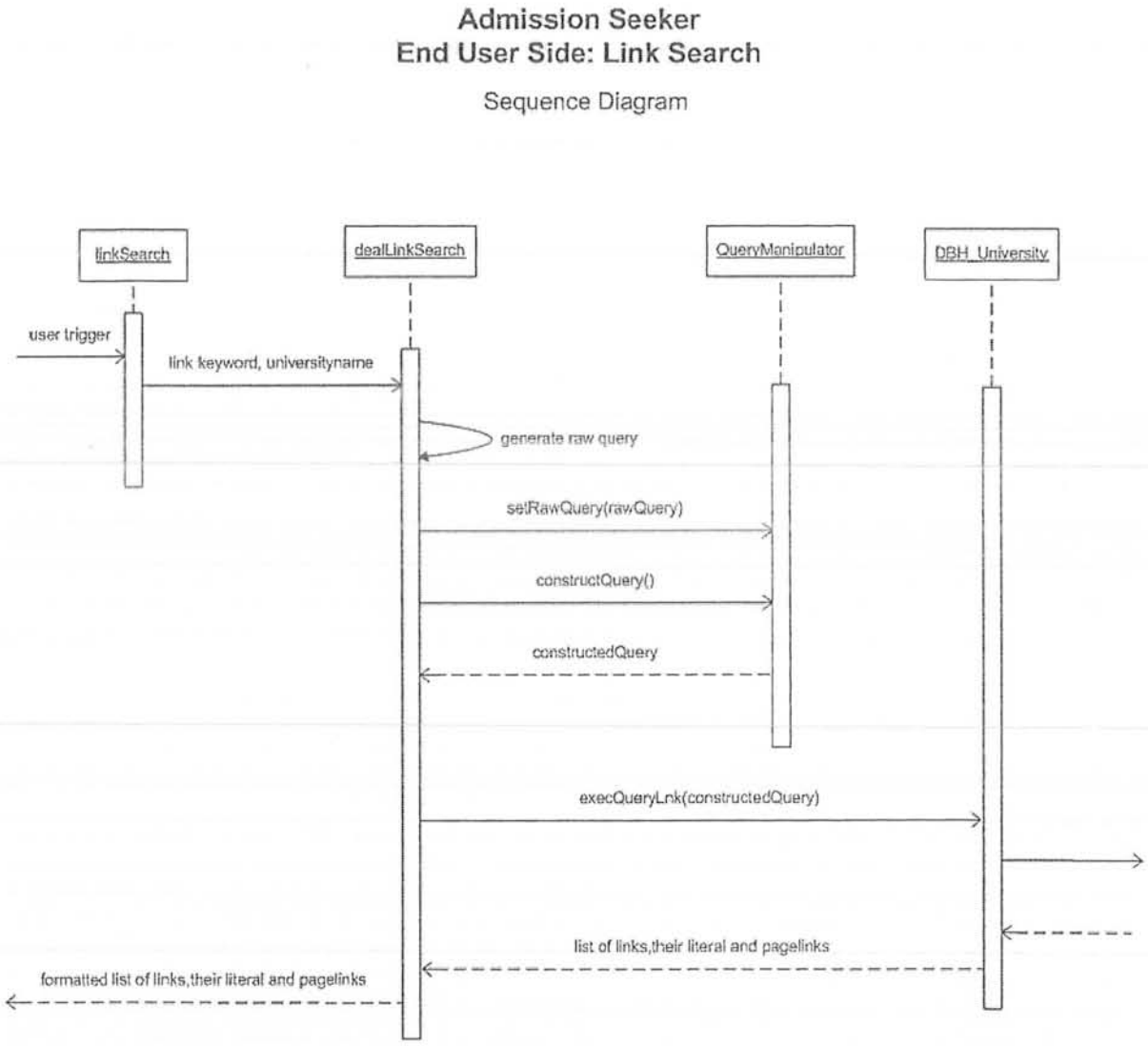
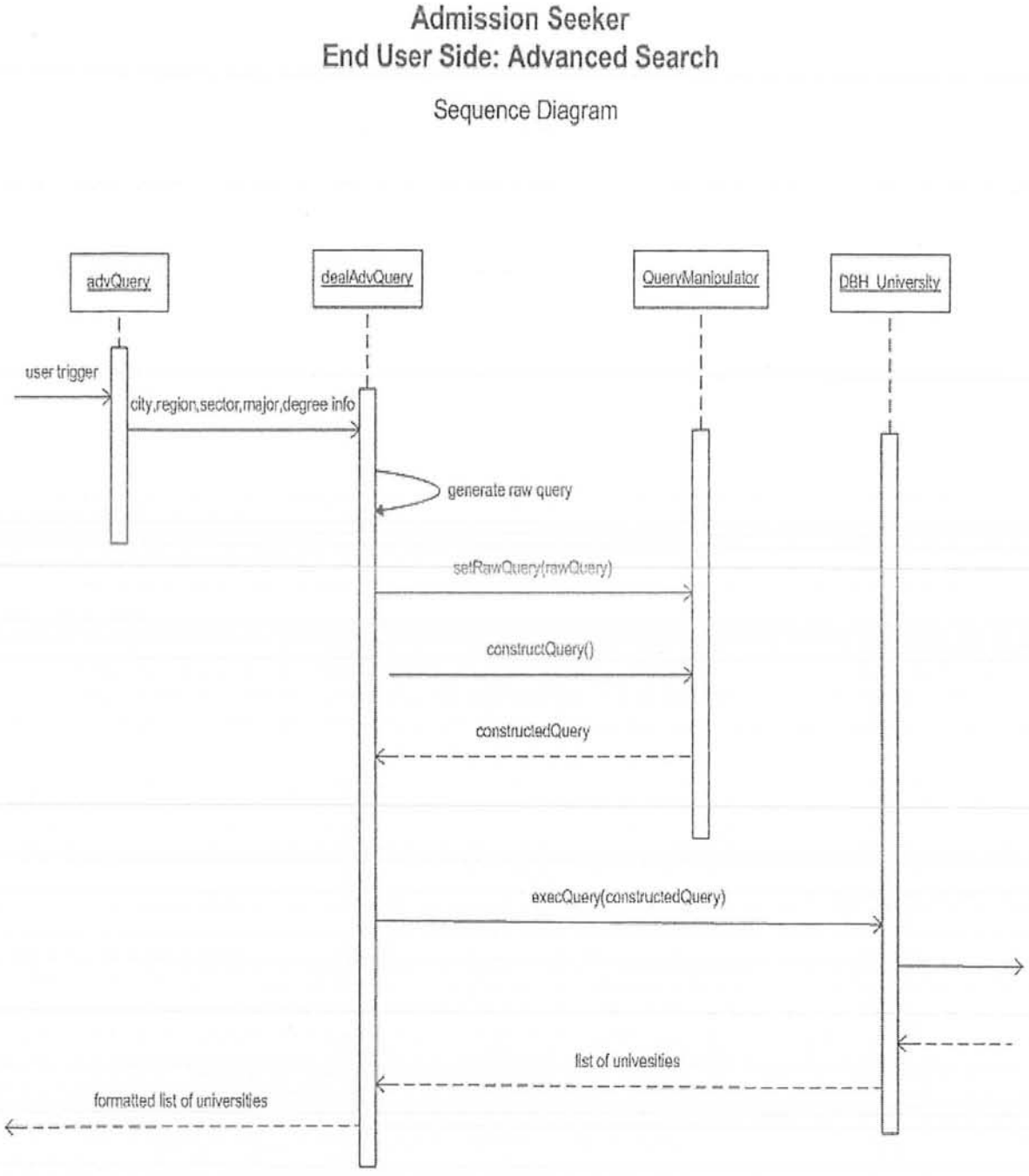
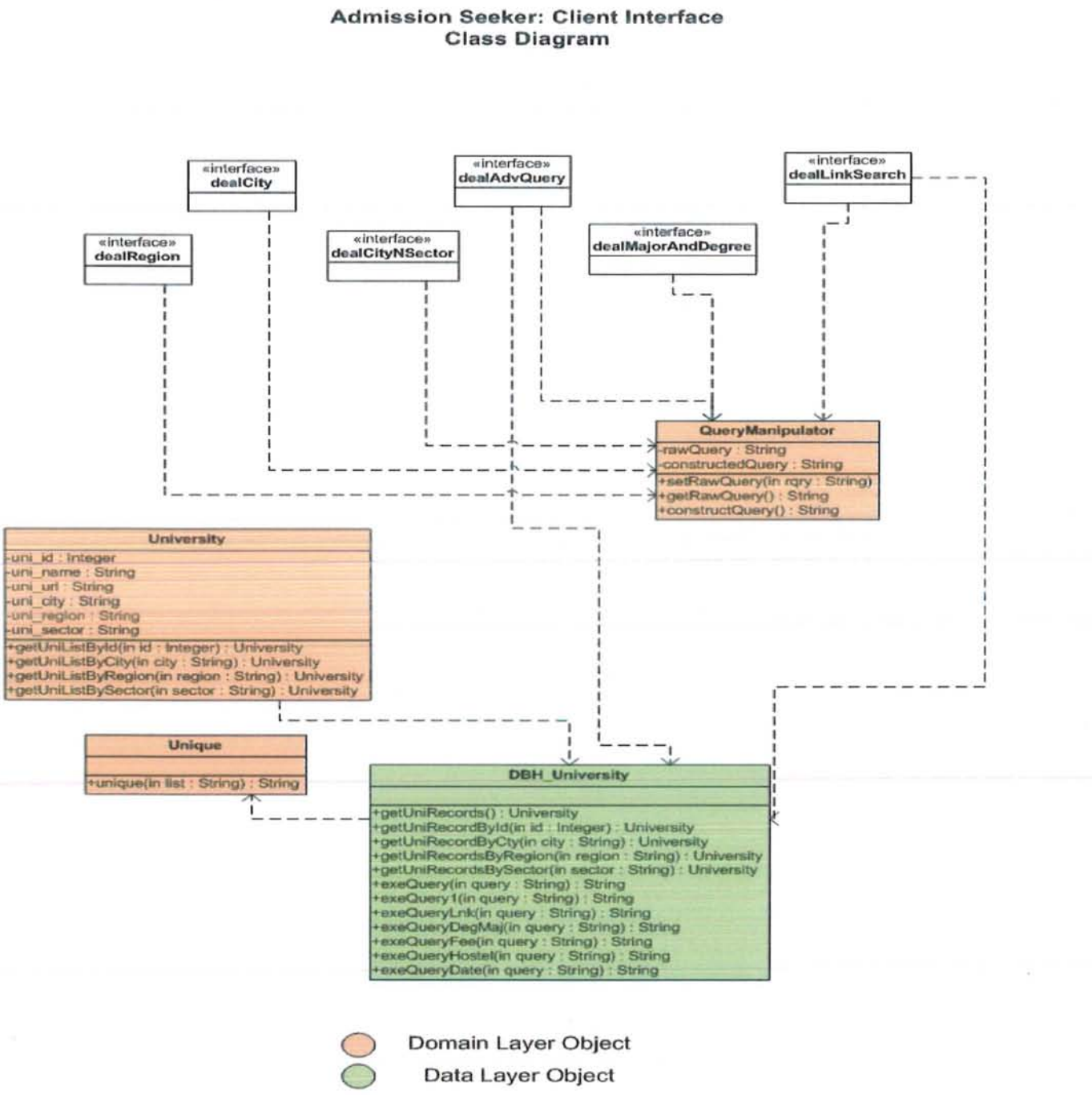


Figure: 6.3. Sequence Diagram: Admission Seeker: End User side: Advance Search



### 6.2.5. Class Diagram

Figure: 6.4. Admission Seeker: Client Interface Class Diagram



## 6.3. Code

Tasks handled during this iteration have been made part of the system through certain JSP pages. Details of code written in these pages increase the size of this document, so it is avoided.

### 6.3.1. AdseekIndexer Class ('createIndex' code)

**Code: 6.1. AdseekIndexer Class**

```
public void createIndex(){

        linkDatabase();
        createUniversityContext();
        try {
            initGATE();

            int corpusSize=uninames.size();
            File uniMainDir=new File(dataDirectory);
            File[] allUniDir=uniMainDir.listFiles();

            Directory directory =
FSDirectory.getDirectory(indexDirectory,false);
            IndexWriter writer = new IndexWriter(directory, new
WhitespaceAnalyzer(),IndexWriter.MaxFieldLength.UNLIMITED);

            for(int c=0;c<allUniDir.length;c++) {
                int uniId=-1;
                URL url = new URL("File:/" +
allUniDir[c].getAbsolutePath());

//System.out.println("\n+++++\n"+allUniDir[c].getAbsolutePath()+"\n++++
+");

                SerialDataStore ds = (SerialDataStore)
Factory.openDataStore("gate.persist.SerialDataStore", url.toString());
                ds.open();
                List<String> list = ds.getLrTypes();

                uniId=getUniId(uninames,uniId,allUniDir,c);

                Corpus corpuss = (Corpus)
ds.getLr("gate.corpora.SerialCorpusImpl",ds.getLrIds("gate.corpora.Seri
alCorpusImpl").get(0));
                String[] arrsplit = null;

                for (int b = 0; b < corpuss.size(); b++) {

                    Document mydoc = null;
                    mydoc = (Document) corpuss.get(b);

                    org.apache.lucene.document.Document orgDoc = new
org.apache.lucene.document.Document();
```

```
                    orgDoc.add(new Field("contents",
mydoc.getContent().toString().replaceAll("[\n]+","
"),Field.Store.YES,Field.Index.ANALYZED));
                    //orgDoc.add(new Field("contents",
mydoc.getContent().toString(),Field.Store.YES,Field.Index.ANALYZED));
                    orgDoc.add(new Field("uniId",
uniIds.get(uniId),Field.Store.YES,Field.Index.NOT_ANALYZED));
                    orgDoc.add(new Field("uniName",
uninames.get(uniId),Field.Store.YES,Field.Index.NOT_ANALYZED));
                    orgDoc.add(new Field("uniURL",
uniurls.get(uniId),Field.Store.YES,Field.Index.NOT_ANALYZED));
                    orgDoc.add(new Field("pageURL",
mydoc.getSourceUrl().toString(),Field.Store.YES,Field.Index.NOT_ANALYZE
D));

                    writer.addDocument(orgDoc);
                }
            }
            writer.close();

        } catch (Exception ex) {
            ex.printStackTrace();
        }
        unlinkDatabase();
    }
```

### 6.3.2. AdseekSearcher Class ('doSearch' method)

```
public AdseekResult[] doSearch(){

         AdseekResult[] ar = null;
        try{
            Directory directory =
FSDirectory.getDirectory(searchDirectory);
            IndexSearcher searcher = new IndexSearcher(directory);
            QueryParser qp = new QueryParser("contents",new
StandardAnalyzer());
            //Term t = new Term("contents", userQuery);
            //Query query = new TermQuery(t);
            Query query = qp.parse(userQuery);
            Hits hit = searcher.search(query);
            ar = new AdseekResult[hit.length()];
            System.out.println(hit.length());
            float maxRel = 0.0f;
            for(int a=0;a<hit.length();a++){
                //query = new TermQuery(new
Term("contents",userQuery));
                query = qp.parse(userQuery);
                QueryScorer scorer = new QueryScorer(query);
                Highlighter highlighter = new Highlighter(scorer);
                TokenStream tokenStream = new
SimpleAnalyzer().tokenStream("contents",new
StringReader(hit.doc(a).getField("contents").stringValue()));
                ar[a] = new AdseekResult();
                ar[a].setLiteral(new QueryParser("contents",new
StandardAnalyzer()).parse(userQuery).toString("contents"));
```

```
ar[a].setDescription(highlighter.getBestFragments(tokenStream,
hit.doc(a).getField("contents").stringValue(),1)[0]);
                ar[a].setUniName(hit.doc(a).get("uniName"));
                ar[a].setUniUrl(hit.doc(a).get("uniURL"));
                ar[a].setPageUrl(hit.doc(a).get("pageURL"));
                ar[a].setUniId(hit.doc(a).get("uniId"));
                if(maxRel < hit.score(a)){
                    maxRel = hit.score(a);
                }

ar[a].setRelevance(Math.round((hit.score(a)/maxRel)*100));
            }
            searcher.close();
        }catch(Exception e){
            e.printStackTrace();
        }
        return ar;
    }
```

## 6.4. Testing

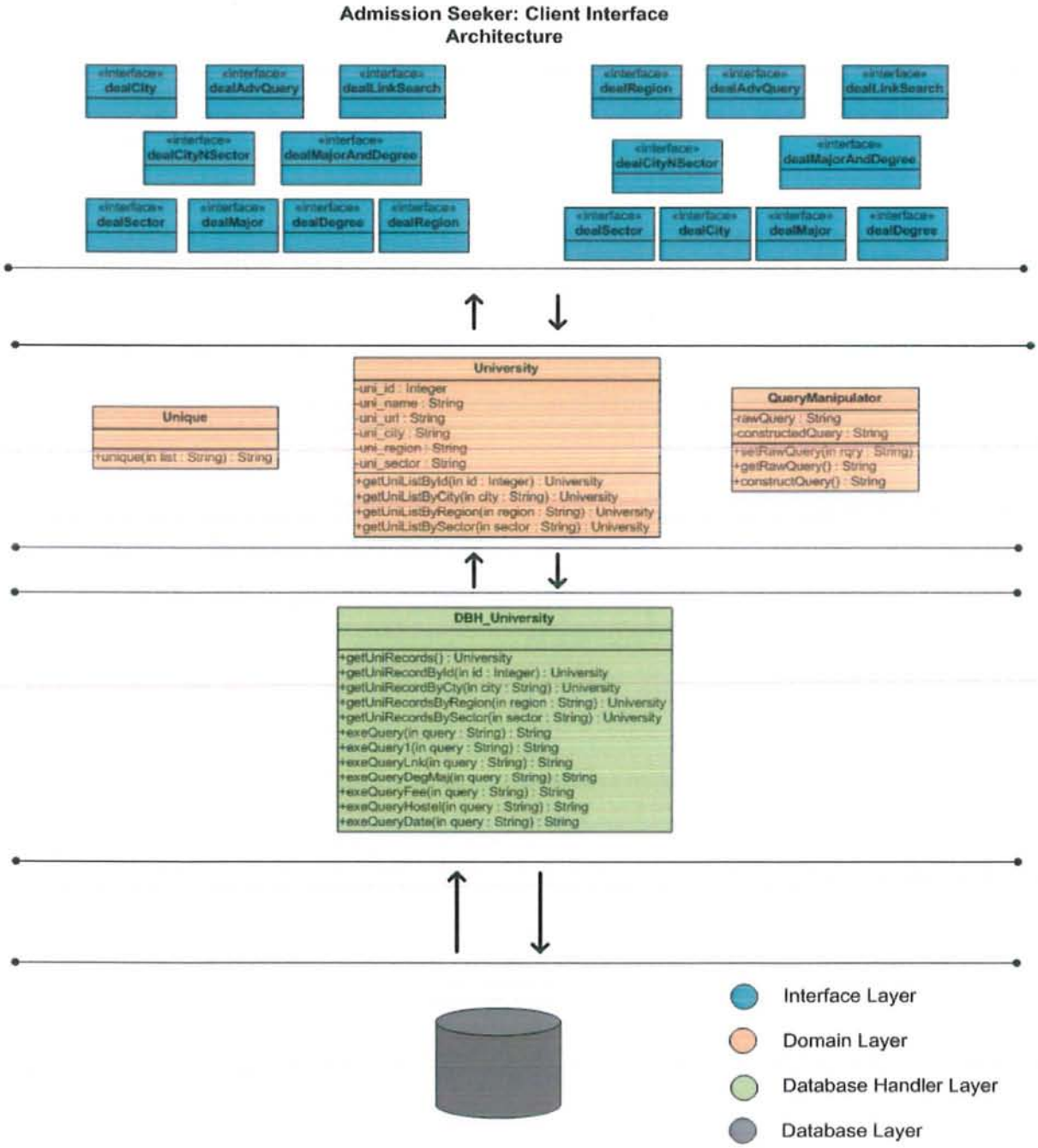
### 6.4.1. Test execution and evaluation

| Test case Id | Name of Executer | Number of executions | Result |
|---|---|---|---|
| TC10 | M.Ali | 2 | Successful |
| TC11 | M.Ali | 2 | Successful |
| TC12 | M.Ali | 2 | Successful |
| TC13 | M.Ali | 2 | Successful |

## 6.5. Integration

### 6.5.1. Web Architecture

**Figure: 6.5. Admission Seeker: Client Interface Architecture**



### 6.5.2. Deployment Diagram

Figure: 6.6. Admission Seeker Web site Deployment Diagram

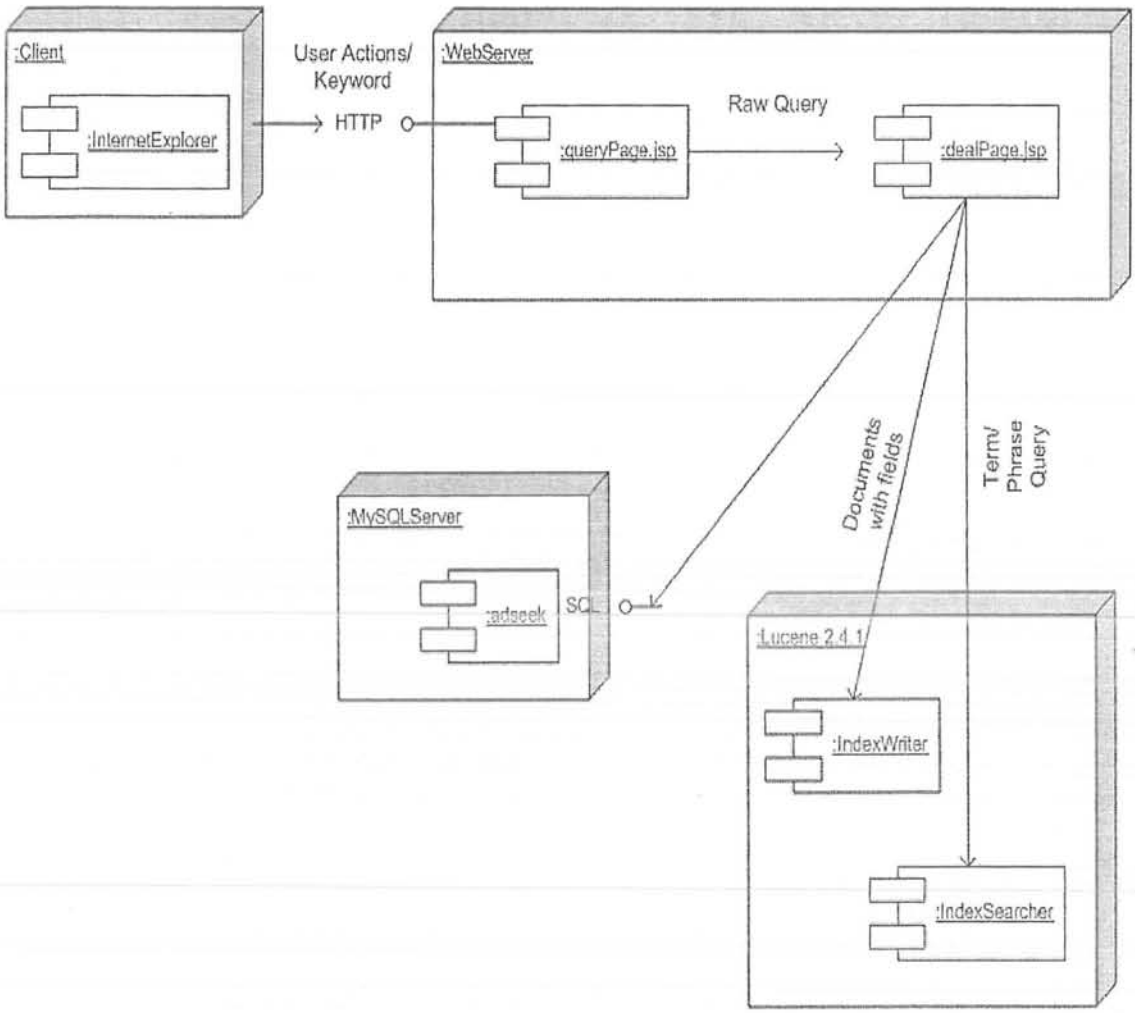
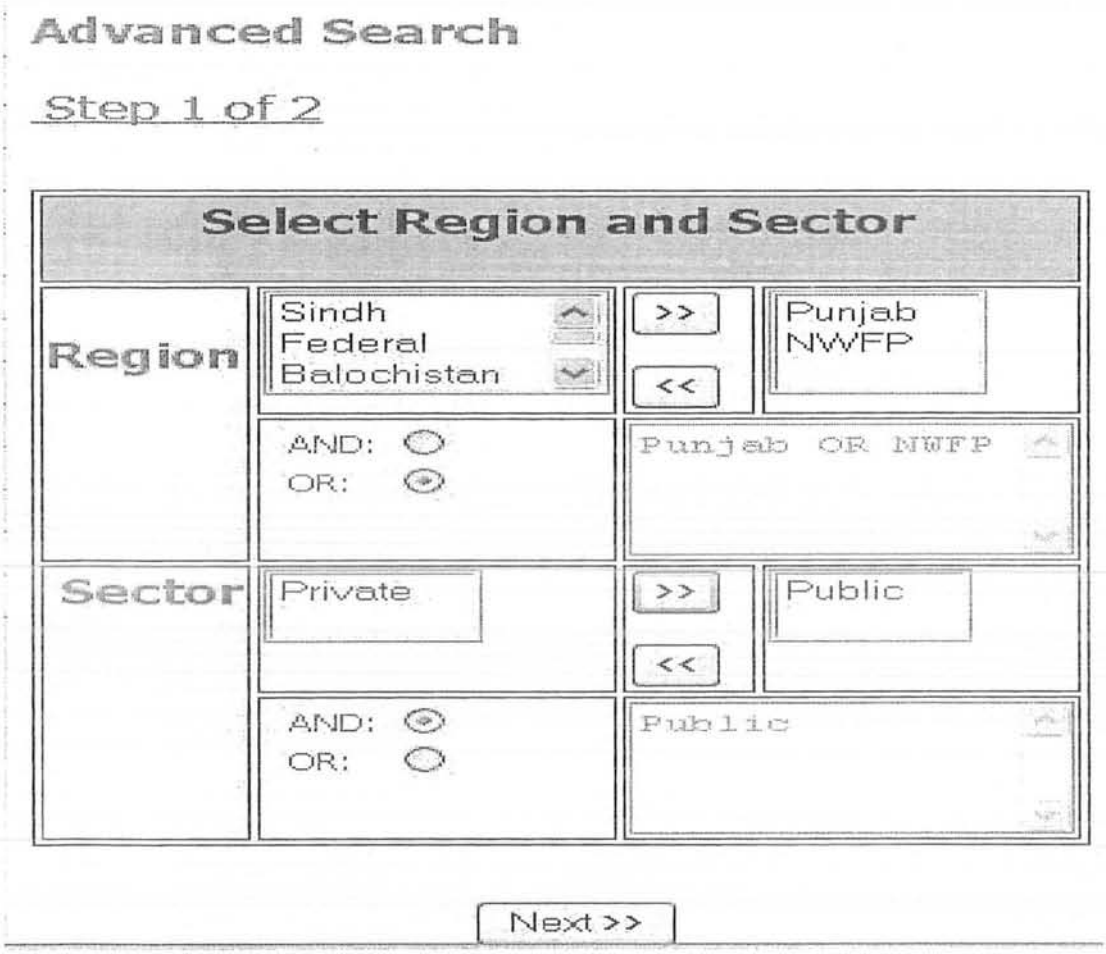



Figure: Admission Seeker Web Site
Deployment Diagram

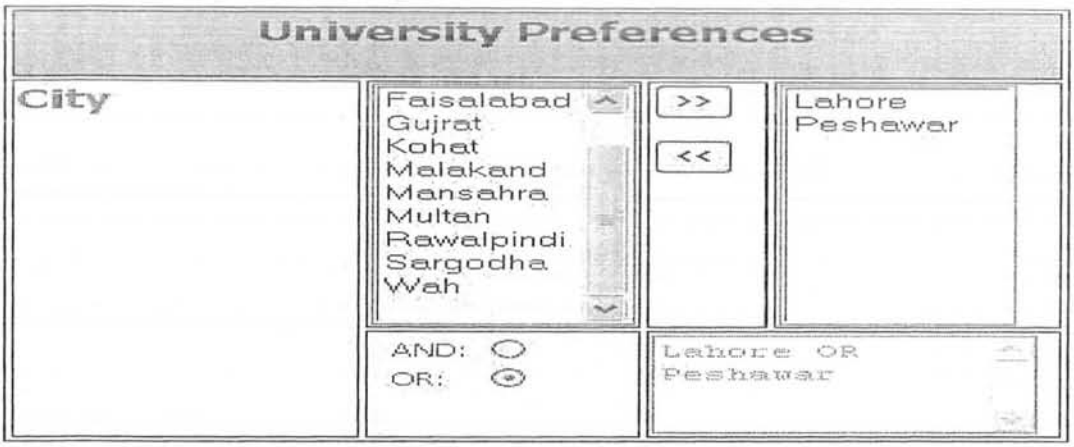
## 6.6 User Interfaces

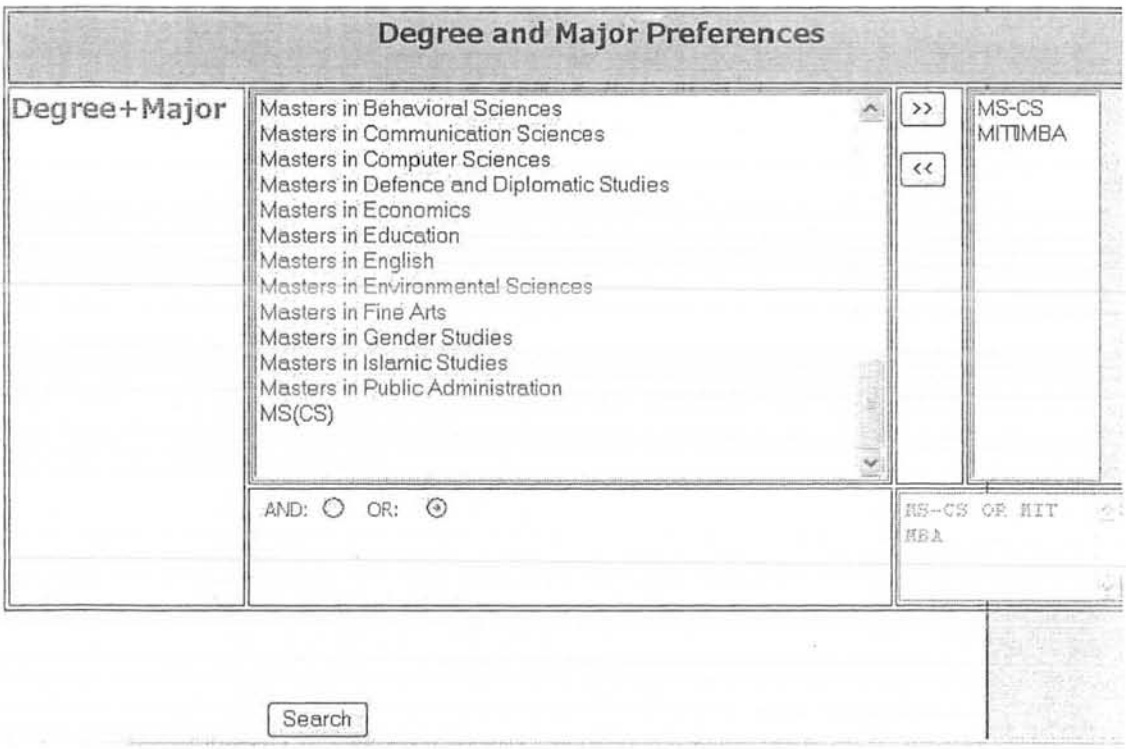
Figure: 6.7. User Interface: Advance Search



**Final Iteration**

Step 2 of 2



## Advanced Search Result (s)

Number of Universities Found: 1

| University Name | University URL |
|---|---|
| Lahore College for Women University | http://www.lcwu.edu.pk/ |
| MS-CS | http://www.lcwu.edu.pk/admission_postgraduate.php |
| MS-CS | http://www.lcwu.edu.pk/programs.php |

## Keyword Search Results

Enter Your Keyword: tuition fee Search

### Admission Seeker Search Results Showing 1 to 5 of 8

**tuition fee** Relevance: 100%

1,500 Rs 1,500 per semester **Tuition Fee** (2008-09) Rs 65,000 Rs 51,000 per semester The fees are

Page URL: http://www.nu.edu.pk/FeeStruct.aspx

University: **National University of Computer and Emerging Sciences**
University URL: http://www.nu.edu.pk/

**[View Details from Admission Seeker]**

---

**tuition fee** Relevance: 53%

the Master's program of study shall be required to pay **tuition** fee and such other dues as may be

Page URL: http://www.qau.edu.pk/admission/Fees.php

University: **Quaid-i-Azam University**
University URL: http://www.qau.edu.pk/

**[View Details from Admission Seeker]**

Chapter# 6

**Final Iteration**

## Keyword Search Results

Enter Your Keyword: hostel Search

**hostel** Relevance: 71%

Department, the University Press, a student **hostel** and some other departments of the University

Page URL: http://www.pu.edu.pk/campuses/default.asp?campid=2

University: **University of the Punjab**
University URL: http://www.pu.edu.pk/

**[View Details from Admission Seeker]**

---

**hostel** Relevance: 40%

400-seat auditorium, four **hostel** blocks, cafeteria, indoor and outdoor sports facilities, a rest

Page URL: http://www.pieas.edu.pk/about/profile.php

University: **Pakistan Institute of Engineering Applied Sciences**
University URL: http://www.pieas.edu.pk/

**[View Details from Admission Seeker]**

---

<<< Previous Next >>>

Chapter 7

# Benefits and Future Enhancements

# 7. Benefits and Future Enhancements

## 7.1. Benefits

As discussed in section 1.2 of this document there are some problems that need to be solved by Admission Seeker. There are some benefits attached to Admission Seeker's existence on the web. Here I would like to throw some light on those benefits:

- end user's time is saved; End user time would not be wasted in visiting different universities' websites
- End user would not be frustrated by browsing every university's website for his/her information of interests
- Information of interests collected from certain sources (universities' websites) are presented in a simplified and general way (as each university has its own way of representing information of interests)
- End user would not face credibility problems with universities as Admission Seeker crawls only those universities' website available on HEC Listing
- Admission Seeker provides 'Link Search' for those users who want direct links to their information of interests by just providing a keyword and list of universities they are interested in taking admission
- Admission Seeker provides 'Advanced Search' for end users to put multiple options together and get results satisfying all of their options
- Admission Seeker provides 'Keyword Search' for those end users who want keyword based search option like **google** search

## 7.2. Future Enhancements

There could be some enhancements that may become part of Admission Seeker in the future. List of some future enhancements are the following:

- Commercial use of Admission Seeker
- Embedding user feedback and request in extraction policies
- End user may add Universities
- International Universities

Chapter 8

# References

## 8. References

[SOM 2005] Sommerville, Ian (2005) *Software Engineering,* 7th Edition, Pearson Education, Singapore.

[LARMAN] Larman, Craig. *Applying UML and Patterns,* 2nd Edition, Prentice Hall, India.

[PRAG 2009] Pragmatic Software Co., Inc (www.PragmaticSW.com) (January 2009).

[GATE 2007] Cunningham, Hamish., Maynard, Diana., Bontcheva, Kalina., Tablan, Valentin., Ursu, Cristian., Dimitrov, Marin., Dowman, Mike., Aswani, Niraj., Roberts, Ian., Li, Yaoyong., Shafirin, Andrey. (July 2007) 'Developing Language Processing Components with GATE Version 4 (a User Guide)', The University of Sheffield 2001-2007, (http://gate.ac.uk/).

[MAN 2005] Gospodnetic, Otis., Hatcher, Erik. (2005) *Lucene In Action,* 2nd Edition, Manning, Greenwich.

[CAST 2004] Castillo, Carlos. (2004) 'Effective Web Crawling', University of Chile