**Media Engineering and Technology Faculty**
**German University in Cairo**

# Dynamic website for German University in Cairo

**Bachelor Thesis**

Author: Omar Ahmed Alfar
Supervisor: Prof. Dr. Slim Abdennadher
Submission Date: 1 June, 2008

**Media Engineering and Technology Faculty**
**German University in Cairo**

# Dynamic website for German University in Cairo

**Bachelor Thesis**

Author: Omar Ahmed Alfar
Supervisor: Prof. Dr. Slim Abdennadher
Submission Date: 1 June, 2008

This is to certify that:

(i) the thesis comprimises only my original work toward the Bachelor Degree

(ii) due acknowlegement has been made in the text to all other material used

---

Omar Ahmed Alfar
1 June, 2008

# Abstract

Currently GUC is considered as one of the most leading universities in Egypt and the region; therefore its own website needs to play a positive role both internally and externally. The website shouldn't only concern about GUC educational and academic side but it must also reflect the image of the ongoing campus activities to the external world. The website should also provide online interactive services to the GUC internal community including staff and students and on the other hand services for visitors and interested parties. GUC website must be presented in creative elegant design and at the same time the GUC website must be implemented with the latest web technologies.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

Nowadays, Internet became one of the most important suppliers of information in the daily life. It is acknowledged as one of the most cost effective forms of advertising available. With the continuous revolution of the Internet applications and features that it delivers, Internet has found its way on the top of communication domain. Currently, Internet is not only bearing information transmission but also it delivers essential services for instance e-government paying bills and online shopping. E-government paying bills had allowed people by just few keystrokes to order birth certificates, review court records, water bills or taxes profiles. Online shopping is also another innovative service for the Internet it grabbed a considerable attention of the two contrasts of the trading field the consumers and the service providers.
Moreover, Internet is an extremely vital medium of interaction between wide ranges of people. The huge recognition it received is derived from the easiness of accessibility along with its availability anywhere worldwide anytime. Internet serves individuals as well as organizations to publish breakthroughs and information. It satisfied its functionality in spreading the knowledge among a very large audience. Anyone can find an easy way to publish a web page with a little initial cost. The great thing about the Internet is that it can reach the whole world. Internet is available all the 24 hours daily Thus, everyone can get to know about as nearly every individual have access to the Internet, sometimes more than one computer with an access to the Internet can be found in a single home. Using the Web, it is also easier than ever before for individuals and organizations to publish ideas and information to an extremely large audience. Anyone can find ways to publish a web page or build a website for very little initial cost. Needless to say, having a web site to promote your business or ideas is imperative nowadays. Having a web site is not just a commodity for businesses nowadays, but a must. A company or a professional without a web site is like a salesman without his business card. The web site needs to be your profile and what's good about it is that there is limitless information you can put on it. Another good thing is that static content on your site can be changed very easily. Even better, you can opt to go for a dynamic site which will automatically change your content based on changes to your business, like stock items, prices, articles, services and more.
The webs site It needs to be eye catching and transmit your message in a few seconds. It needs to have enough content that explains your message and what you do. It needs

to load fast, be reliable and look professional. The impression your web site should give is that of a company or professional who knows what they want to sell, or the message they want to send.

## 1.2 Objective

This project is intended to deliver a complete solution for the GUC website, from a software perspective, aside from any interface design issues. The solution must ensure the satisfaction and better experience of all stockholders involve that might use the website including GUC staff ,enrolled student, prospective students, press ,parents and visitors. This solution must apply to the latest web technologies and standards.

## 1.3 Current GUC website

This section presents an in depth analysis of the current GUC website. The analysis will focus on the following issues

- Implementation
- Information offered
- Services and Functionalities offered
- Design

The current GUC website is implemented using basic HTML with the help of java scripts for some parts; by this we can conclude that modifying and updating the content of the website requires technical background and it's not a task anyone can do. This also may results undesirable conflicting since the probability of human errors is high while editing and modifying in the HTML code.

Current GUC website offers good amount information however the amount of information is not sufficient meaning that some information that website visitors would like to know doesn't exist for example the current GUC website doesn't offer any information about the steps and procedures that students who would like to transfer from other universities to the GUC have to do. Another drawback on the information presented in the current GUC website is that it's out dated.

One of the major drawbacks of the current GUC website that it's easy to get lost while navigating the website contents this is due to the unstructured distribution of the information. Current GUC website main navigation menu is unstable meaning that the titles included in the navigation menus location change with every page and event sometime some titles are disappeared by this the user navigating the website will get confused and lost. Another drawback of the current GUC website is that a lot of hyperlinks are broken and leads to no page in addition some hyperlinks can take the user into a loop to the

same page while it's supposed to redirect to another section or page.

The only service offered by the current GUC website is the online admission for the perspective and MBA applicant likely this service is offered in stable good manner.

The design of the any website is an important issue in view of the fact that good design leaves good impression. Current GUC website interface design offers irrelevant colors along with very low quality images.

To summarize the major drawbacks of the current GUC website are

- It's Offered in native HTML.
- The information offered is insufficient and unstructured.
- It offers few amount of services for users.
- The website is presented with poor design is.

## 1.4 A Look on related work

For better insight about what functionalities, features and services the new GUC website should include a survey was made on twelve top universities from different regions , the universities that were involved in the survey are

- Stanford University (www.stanford.edu)
- University Pennsylvania (www.upenn.edu)
- Massachusetts Institute of Technology (www.mit.edu)
- Oxford University (www.oxford.edu)
- Cape Breton University (www.cbu.ca)
- Ludwig-Maximilians University (www.en.uni-muenchen.de)
- American University in Cairo (www.aucegypt.edu)
- Cairo University (www.cu.edu.eg)

## 1.5 Thesis Overview

Chapter 1 provides an introduction to the project and review of the current GUC website drawbacks. Chapter 2 discusses in details the make over solution for the GUC website. Chapter 3 discusses the implementation of the make over solution in details this includes the system design, architecture and database design. Chapter 4 lists the conclusion and future proposed work. Appendix A provides a list of the system database tables.

# Chapter 2

# The Make Over

After discussing the drawbacks of the current GUC website and investigating other universities websites design,features and services a solution was drawn to fulfill the GUC needs. This chapter presents the make-over solution of the GUC website.

## 2.1 Web Content Management System

The content of large scale website as GUC website are quite challenging since the content grow rapidly over time and is subject of change and update. For example the admission deadlines change every year. Another challenge on managing the website content is that it requires knowledge of basic and sometime complex web technologies and standards. Even with the existence of this knowledge a new challenge will arise which is in order to manage the content it cannot be from any where, it has to be done on the server machine hosting the website or through remote access to the website host machine which requires high speed Internet connection.

The corner stone of the make-over solution is to provide a **web content management system** (WCMS) to gain full control over website contents. The WCMS covers the complete life cycle of pages on the website starting from creation to the presentation of the page.

The idea behind the WCMS is that it provides an easy-to-use WYSIWYG web-based authoring environment to manage the web site content. This easy-to-use environment is designed to work like Microsoft Word or any other popular authoring tools, by this it provides a non-technical way of creating new pages or updating content, without having to know any HTML or other web technologies.Also since its web-based environment, it allows content managing and updating to be done remotely from any where.

GUC website WCMS offers the ultimate balance between power and robustness combined with an incredible ease of administration, and publishing. Even the most complex actions are done with simple clicking.The WCMS user don't need to be technical user or know a single line of HTML, CSS or any other web technology to be able to manage

the contents of the website. The architecture of the web content management system is described in details in section 3.2.

## 2.2 System Features

In this section the features offered by the implemented WCMS for the GUC website is presented.

### 2.2.1 Ticketing System

Managing GUC website content is impractical work load for one person, even with the existence of the WCMS. This is due the fact that the website provide huge amount of information distributed over website sections. This information comes from different sources and covers different areas of interests. To ensure web site contents consistency and integrity, the possibility for editing and managing web contents will be given only to the website administrator, nevertheless Ticketing System functionality will ease the task of the website administrator to manage a large scale website as GUC website. Ticketing System enables website administrator to issue customizable account tickets with privileges to edit website contents ranging from very limited to a clone of the web administrator privileges.

Ticketing System offers advanced account customizing which grant the web administrator the power to specify exactly the tasks and privileges of every issued account ticket. Every ticket has specific edit privileges, where the website sections that can be managed and edited by this ticket holder is defined. Also the website administrator can grant a validation period for the ticket, where the ticket is automatically deactivated along with all its privileges as soon as the specified time period is passed, Figure 2.1 illustrate how an account ticket can be customized.

After creating the ticket and customizing its privileges and validation period, the website administrator can invite a user to use this account by sending an electronic invitation to the user, and then the user can activate and start using his given privileges.
Ticketing System helps to:

1. Know who is editing where, and for how long.

2. Keep the website up-to-date with accurate information since managing the contents work load is divided among Competent users.

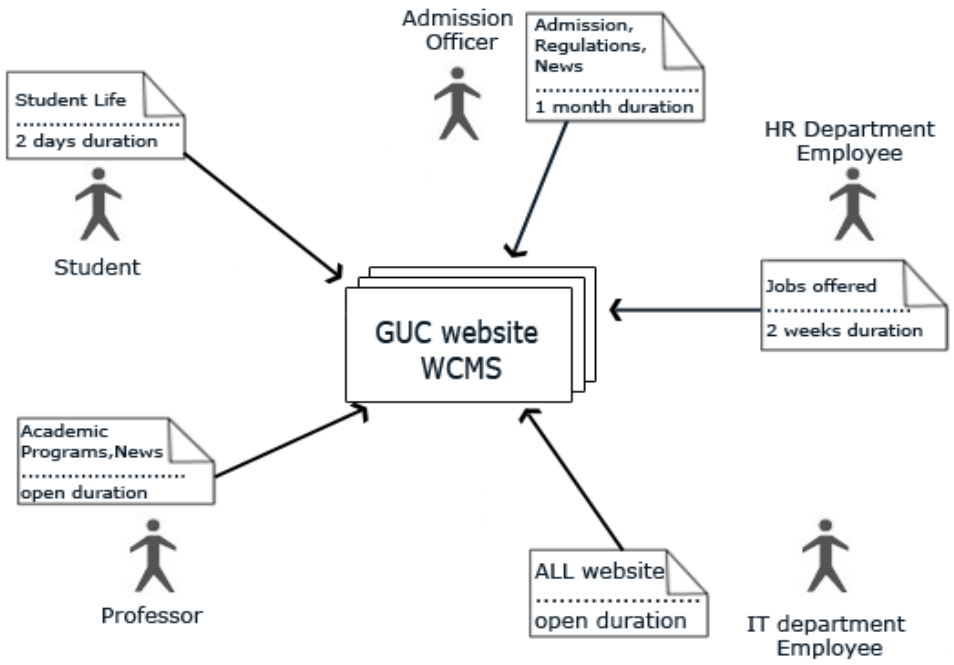


Figure 2.1: *Illustrating how account tickets can be customized by website sections and validation duration*

### 2.2.2 News Engine

Now that the GUC has developed to be one of the most prominent universities in Egypt, and one of the leading educational organizations in the region. It's important that news about GUC events, achievements, activities and all related GUC issues to be delivered up to date to website visitors. This up to date news will help students, prospective students, GUC staff, parents, visitors and press to know the latest about GUC and will also help to gain more reputation through publishing its latest remarkable achievements. Not only the latest but also news archive will be available to help users to know about past news and dates.
The features that will be offered by the news engine are :

- **Headline generator wizard**
  This functionality offers an easy step by step walk through wizard for Adding new news headlines. The generator wizard is easy to follow and offers extended control while adding.

- **Rich Media Support for headlines**
  This functionality will give the ability to support news headlines with images that can be easily uploaded and automatically viewed as an album with the headline.

- **Request news headline**
  Many times users that don't have an editing account ticket in GUC website WCMS would like to publish news headlines, for example some students would like to publish a headline about their active working group achievement. These kinds of

headlines are beneficial to be added to the GUC news repository but to give every single user who would like to add a headline an editing account ticket is not a convenient task and may result inconsistency in the information represented in the website. This feature is implemented to overcome this problem by giving the ability to any user surfing the GUC website to request adding a news headline. When the user request a news headline to be added he or she go through a step by step wizard where he or she define the headline title and other necessary information along with the news headline contents and description. After feeding the required information about the headline the user submits the headline at that time the headline is automatically transported to pending headlines list. The website administrator can view the pending headlines list and view the requested headlines contents .The website administrator can then accept or reject the requested headline to be posted in GUC website. When website administrator accepts a headline it's automatically moved from the pending headlines list to the news repository viewed normally along with other GUC news. Rejecting a headline will results in removing it permeably from the pending news headlines list.

- **Categorize news headlines**
- **Mark headlines as coverage of events**
- **Archiving of news headlines**
- **Headlines search engine**
  Headlines can be searched and filtered by their, categories, dates and keywords this will simplify the task for the user when trying to find the desired headline among the large amount of headlines available.
- **Related contents aggregation**
  For better website visitors experience and to ensure flexibility during navigating the website contents, a list of related contents from other website sections and articles can be defined for every headline and automatically viewed when the headline is viewed.
- **News highlight**
  This functionality allows marking a news headline as a highlight where it will be showed in news section in different styling and in more details. Highlight is the headline that GUC would like to scope on and attract the visitors to read it.
- **Notification functionalities for interested parties**
  GUC news is frequently updated, by notification functionalities the user can get automatic feeds of the newly added headlines with out the need to check on the website; all the user has to do is to register to the desired notification method from different notification methods available (Email notifications, RSS feeds).

### 2.2.3 Events Engine

Events happen periodically in GUC; these events cover wide diverse areas as academic events, ceremonies, competitions, sport activities, open days, conventions, parties and

many other areas. Furthermore events can be categorized as off campus and on campus events. Covering these events and offering correlated information for website visitors including GUC staff, students, prospective students and all kind of visitors is an essential issue since it helps GUC to advertise for its events and get interested participants. The implemented GUC website WCMS offers smart events engine that facilitates the publishing of the events and their related information and present powerful features that enhance the website administrator and user experience.

GUC website WCMS will offer the following functionalities for the events engine as implemented in the news engine discussed in 2.2.2:

- **Event generator wizard**
- **Rich Media Support for events**
- **Categorize events**
- **Archiving of events**
- **Events search engine**
- **Events highlight**
- **Notification functionalities for interested parties**

Another important feature that event engine offers is the possibility to **augment the events with agenda** which describe the day by day event activities, each entry to the event agenda shows the time the activity start , the end time of the activity and the description of the activity. The event agenda is showed with its related event in sorted friendly manner.

### 2.2.4 GUC Calendar

GUC calendar functionality will offer an easy way to show GUC important dates and deadlines. The calendar is not going to be as an ordinary calendar just showing marked dates but it will offer more extended futures. First the calendar is divided into three sub calendars:

1. **Academic calendar:** where everything related to the academic year will be marked, this includes academic year start and end dates, exams dates and vacations. Not only current academic year calendar can be viewed but also previous academic years' academic calendars are archived and are viewable.
2. **Deadlines calendar:** where the admission and financial affairs deadline are shown and highlighted.
3. **Miscellaneous calendar:** this calendar will show miscellaneous dates that are marked and highlighted.

Adding a date entry to GUC calendar is very easy task, all the user using the WCMS has to do is to pick up a date, write a description about the date and specify to which calendar of the three sub calendars this date entry belongs.

Another feature of the GUC Calendar is that it's automatically feed, meaning that it will be automatically feed by dates from existing events in the events repository, where events will be added to the calendar by their start and end dates, also links to the corresponding event page will be displayed.

When the GUC calendar is viewed by website visitor, it's shown with the three sub calendars grouped into one calendar and upon the visitor choice the visitor can group and filter the calendar by selecting from the three sub calendars.

Since the GUC calendar will archived all the entered dates a search and quick reach mechanism is offered to website visitor to smooth the process of finding the required calendar dates. The website visitor can search the calendar by keywords or can quickly jump to specific date or even date range and view the calendar entries associated with the provided date\s.

### 2.2.5 Course Catalog

Course catalog offers an organized, detailed, descriptive list of course offerings at GUC. The catalog will offer detailed information about GUC faculties and academic departments, the academic programs offered by the faculty or department and the offered courses. The corner stone of the course catalog is that it overcome the hardness of viewing the links between the faculties, programs and courses as it exactly shows which faculties offers what programs and also it shows exactly the courses correlated with each individual program and in which semester is offered. Additional information covering course prerequisite, credit hours and the faculty or department that offers the course will also be presented. Course catalog will help visitors especially prospective students to learn more about GUC programs and to decide which program they would like to join.

The main features course catalog will offer are :

- **Step by step managing techniques**
  Since enhancing the education quality is GUC main concern, its curriculum is subject of change and update, for that managing the course catalog is done in easy step by step procedures. New entries for example courses, programs or even faculties can be added easily and linked with the exciting entries in the course catalog. Managing techniques is also extended to provide the ability to manage small details as course perquisites, course credit hours.

- **Courses search engine**
  The user can quickly find desired courses by using the featured quick reach search techniques. The user can directly find courses by keywords or can filter the course according to certain programs , faculties and semesters.

### 2.2.6 Career

Recruitment is the key to a growing and successful of any organization. Finding the right person is not always easy, and successful organizations can't be built without great people. For GUC as large educational organization, the quality of recruitment can affect the intellectual shape of graduate generations from GUC.

The process of recruitment for GUC is challenging since GUC requiting focuses on two distinct areas:

1. Academic related recruitment, for example professors , lectures and teaching assistants.

2. Non academic recruitment, for example human resources , financial affairs and other administration positions.

There are three main problems that may face the staff management and recruiting which are:

- **Step down or resignation of GUC staff member**
  This problem is very critical for the case of academic staff member since unexpected step down or resignation of the academic staff member during the academic year may affect the flow of courses and program plans. This problem is also having great affect on non academic staff members as well.

- **Delay of recruiting due to lake in market**
  Sometimes the recruiting of a job vacancy takes more than the expected time due to lake of highly qualified candidates in the market , this problem is also very critical in the academic staff requiting since academic year may start with missing vacancies still open.

- **Coast of advertising for job vacancies**

The implemented WCMS for GUC website will offer e-recruitment engine to overcome the problems faced in recruitment process and to facilitate GUC human resources process to find candidates that are very actively looking for work and to help interested candidates to find appropriate job positions in GUC. The e-recruitment engine is evolved to encompass end to end recruitment.

- **Easy managing and posting for jobs**
  This feature facilitates the posting of new job vacancies since its also done through easy step by step wizard , the job position is defined by title , reference code , description and job qualifications and responsibilities which can be added and incremented dynamically. Also this feature will offer extended editing facilities to edit information about already posted jobs.

- **Viewing and applying online for job vacancies**
  Interested candidate can view and see information about current job vacancies. If a job vacancy meets the candidate's desires the candidate can create and online profile and apply for the job vacancy online.

- **Create candidate profile**
  This feature is offered for job candidates. Interested candidate can create an online profile where this profile will hold all candidate information, education levels and previous experience. The online candidate profile is meant to be as virtual curriculum vita for the candidate and also it saves candidate time when applying to more than one job Creation of the candidate profile is done in easy step by step interactive wizard to enhance the user experience.

- **View and search for candidates from candidate poll**
  In staff shortage cases GUC Human resources can look in the registered candidate repository to find applicable candidates, this feature helps to facilitate this task.

- **Monitoring of job applying history**
  By this feature each job will have a history log with candidates who applied for the job. This log will help GUC human resource to know all candidates interested in the job by this it will ease the requiting process.

- **Search job vacancies**
  This feature will help the user to find the appreciate job. The user can search for jobs by keywords, postdate and categories.

- **Notifications for job responsible with new applies**

```
Dear Sir,
Candidate with the following Profile has applied to job titled:
Academic Coordinator

Candidate ID :  11
Name :          Ahmed Badawy
Birthdate:      1986/9/16
Email :         ahmedbadawy1986@gmail.com
Mobile :        0163862082
Address :       Sherouk City , May Fair compund.


Education Levels :

* Bachelor Degree, from German University in Cairo in the field of Material
Engineering, aquired on June,2007.


Work Experinces :

* Testing Engineer In Beeco electrics from January,2008 To June,2008.

for more information please contact webmaster

Regards Career @GUC
```

Figure 2.2: Automatic notification email sent to job responsible

- **Thanks notification for candidates when applying**

```
Dear Ahmed Badawy

Thanks for Applying for
Position :Academic Coordinator

your application will be processed..

Regards,
Career @GUC
```

Figure 2.3: Thanks email sent after applying

### 2.2.7 Media Gallery

### 2.2.8 Interactive Map

Interactive map will provide virtual tours for the entire GUC campus which will help drawing the real image of the modern-designed GUC campus for anyone who would like to know the real GUC structure. It will also help the students, staff and parents to get directions inside the vast GUC campus by the 360 degrees tours for the campus landmarks.
Interactive map for the GUC can also be helpful showing the latest construction achievements on campus which will keep people aware of GUC's what's new. This might be involved in student's decision for joining this university as this virtual tour will increase the students awareness for the GUC campus and facilities which usually reflect on the student's decision of joining the university. To summarize, the features that will be offered by the interactive map are:

- Virtual tours for the GUC campus.
- Show the latest construction achievements on campus.
- 360 tours for the campus landmarks.

### 2.2.9 Online Admission

Online admission feature introduced in the GUC website mainly facilitates applying to the GUC educational system either as an undergraduate student or postgraduate student. Online admission has been developed for easier applying steps as it enhances the experience of the user interaction with the GUC educational system and the services offered by the university. The implementation of this feature is based on the latest web technologies.

### 2.2.10 Multiple language support

GUC website will be offered in two different languages which are the English and the German language since the GUC University applies the German educational system in Egypt. Presenting the GUC website by these two languages will facilitate publishing all the news and the information related to the GUC University all over the world, once the user opens the web page he is able to choose the language either English or German.

# Chapter 3

# Implementation

In this chapter the implementation of GUC website is discussed in details.

## 3.1 Constraints Applied

The only implementation constraints that were applied are to use ASP.NET as the developing framework for the GUC website WCMS and to use Microsoft SQL as the database management system.These constraint were applied due to the fact that the website is going to be integrated with GUC existing system that‘s using the previous mentioned applications.

## 3.2 System Architecture

This section discusses the implemented system's architecture models, the fundamental architecture design decisions made when designing the system and why these decisions were made. The advantages and disadvantages of the used architectural models are presented in this section also. The architecture of the system was based on two different architecture organizational models where each will be discussed briefly in this section.

### 3.2.1 System Model

Client-Server model was used as the system model for the implemented System. Client-Server model is a software architecture model where the system is organized as a set of separated clients and servers and it is almost used over computer networks. A client is characterized to be an active requester that requests services from a server and waits for the replies from the server and the server is characterized as a passive provider of services that waits for requests from clients and upon receipt of requests, processes them and replies with services to requesting clients. Usually, clients are connected to a small number of servers while servers are connected to a large number of clients.
Advantages of using the client-server model include:

1. Ease of maintenance of an application since it is possible to upgrade, modify or even relocate a server while its clients are unaffected and unaware of the changes.

2. Greater security for data stored in the servers because it has higher security than that at the client's side.

3. It allows for roles and responsibilities in computing to be distributed among different independent computers that are known to each other only through the network.

### 3.2.2 System Structure

The System was structured in Three-tier architecture.Three-tier architecture is a layered architecture model based on the client-server architecture and heavily used in web-applications structuring.
Three-tier architecture consists of three layers(tiers)

1. **Presentation Tier**
   This tier is responsible for outputting other tears processes and for inputting data coming from user interactions with the web application through browsers. Also this tier is mainly Concerned about the user interface of the web-application.

2. **Application Tier (Functional process logic)**
   This tier is responsible for controlling the functionalities by performing detailed processing.

3. **Data Tier**
   Database Servers are managed by this tier. The services include information storing and retrieval. This tier keeps data neutral and independent from other tiers.

Previous mentioned tiers are developed and maintained as independent modules.The three-tier architecture is used when an effective distributed client-server design is needed that provides increased performance, flexibility, maintainability, reusability,and scalability, while hiding the complexity of distributed processing from the user which is the case in the proposed system and therefore, this architecture is used in the implemented system. A potential problem in using three-tier architecture is that the separation of the functions of the three tiers is not always obvious. The placement of a particular function on one of the tiers should be based on some criteria like ease of maintaining or ease of administration. The implemented system follows the three-tier architecture in a major part of it.

How each Tier of the Three-tier architecture is used the system is described below.

#### Presentation Tier

This tier contains the ASP.NET code which is converted into HTML code that will be viewed in web browsers , also it contains C-Sharp code files that are associated with every individual ASP.NET page that receive data entered by the user and response to the user actions.The C-Sharp files also may be involved in generating HTML code dynamically ,this code is to be appended automatically to the generated HTML code from associated ASP.NET where The combined HTML file resulting from ASP.NET page code and C-Sharp code is viewed for the user in the browser.

#### Application Tier

Application Tier Provides group of C-Sharp code files that are responsible for providing services, processes and calculations needed for the upper presentation tear and the lower data tear.The application tear C-Sharp code files also help to encapsulate data coming from the presentation tear into business entities that are processed by the Data Access Layer in the data tear.Also the is responsible for receiving business entities coming form the Data Access Layer , perform necessary processes and then transport the results to the presentation tear.

#### Data Tier

First this tears contains the data access layer to the Database storage , also in contains the main system DataBase. The Database design is discussed in details in section 3.3 , and the Data Access Layer is described in details in section **??**.

## 3.3 Database Design

The database is the heart of the system since it stores all system information and data. Database where designed in Relational model. Before starting the database implementation some steps where taken to ensure database integrity and to make sure that it's sufficiently normalized The first step that was taken before database implementation where to design an entity-relationship model is which is an abstract conceptual representation of structured data. Entity-relationship modeling a heavily used Relational Database modeling technique. Entity relation ship modeling is presented in diagram which is called Entity Relationship Diagrams (ERD). The ERDs that was used to model the system database are show in figures.

By modeling the database in ERD diagrams it became easy to start with designing the database schema since with the help of ERDs a clearer look of how that database should like is available.

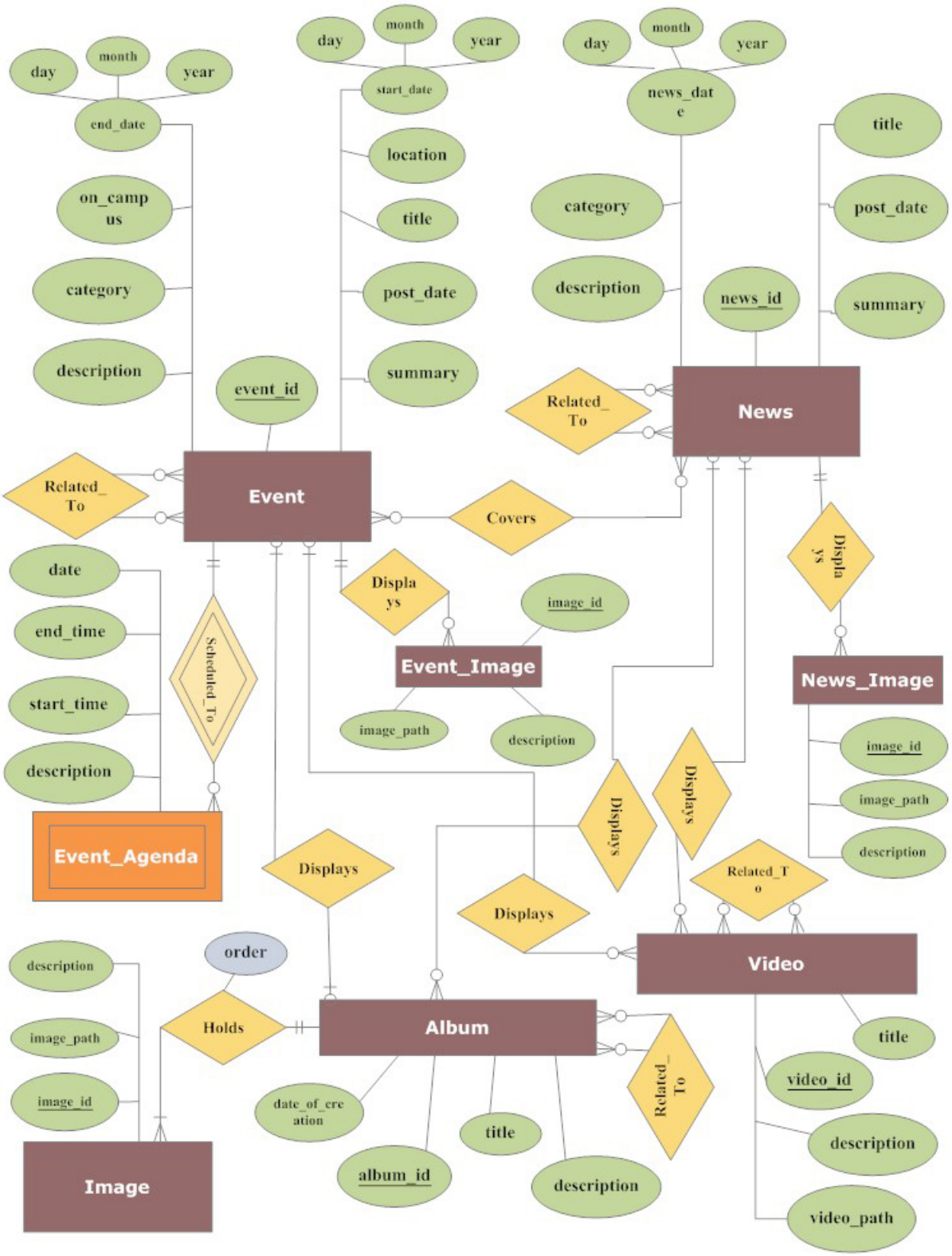


Figure 3.1: Database ERD part 1

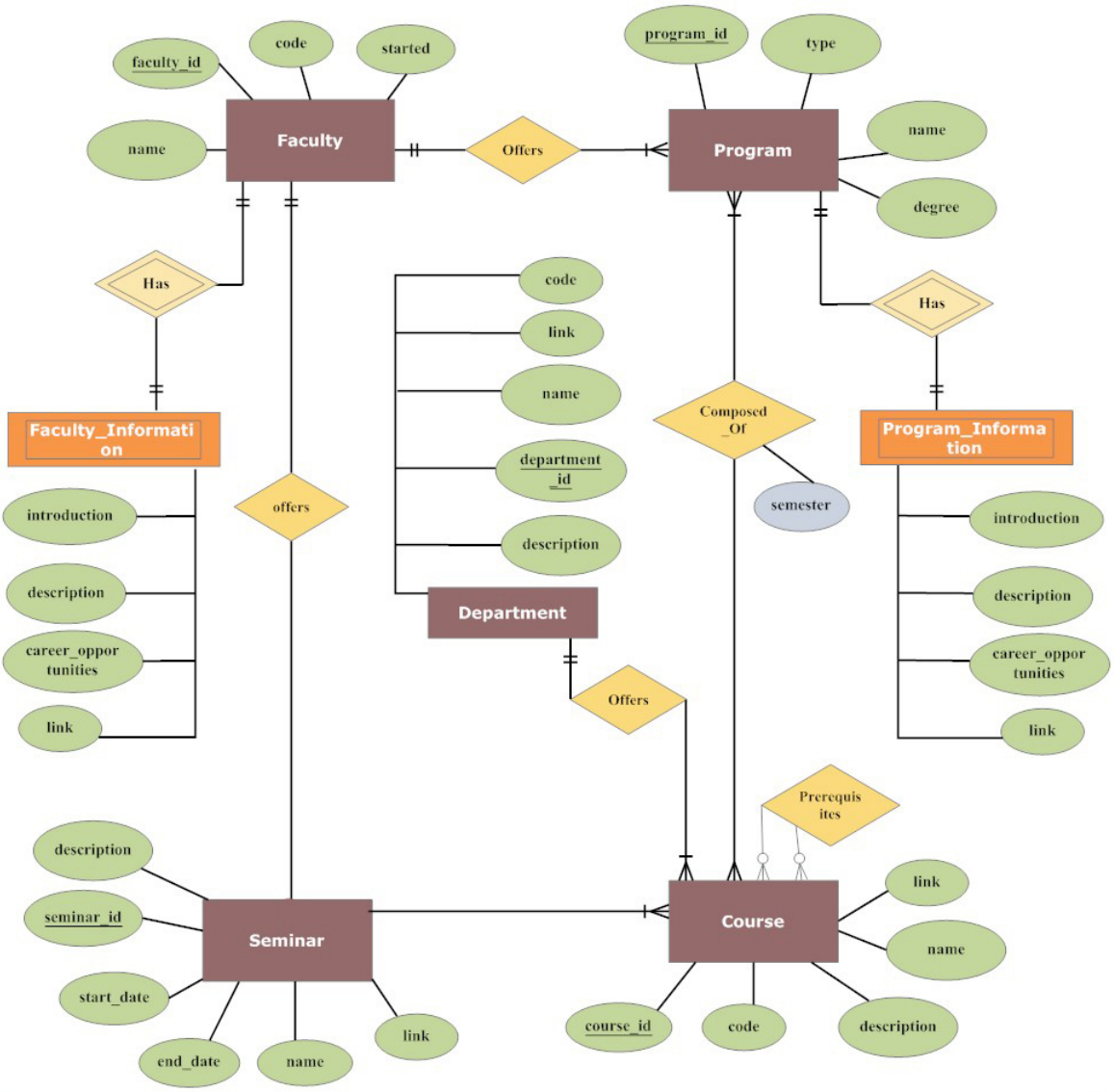


Figure 3.2: Database ERD part 2

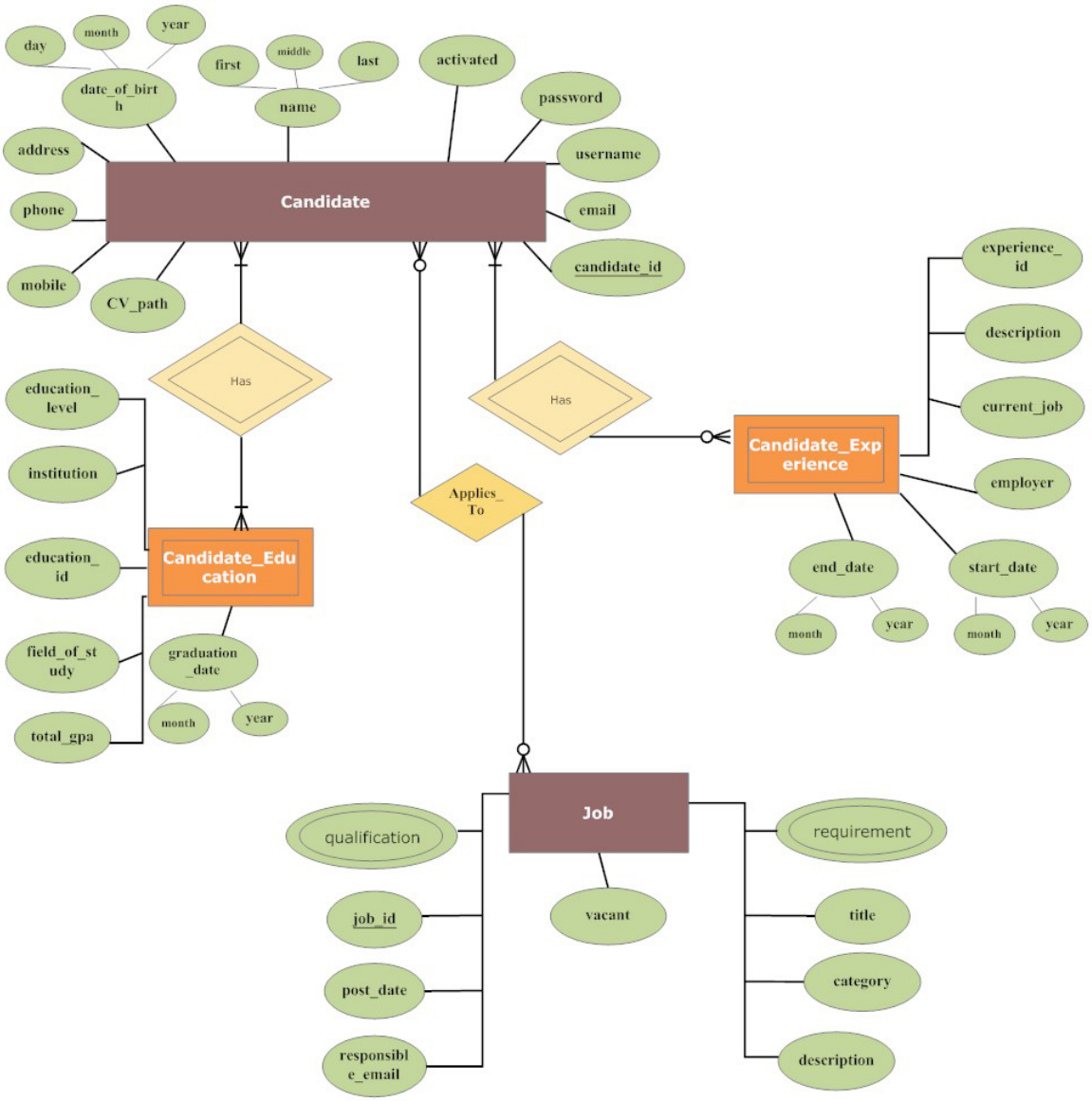


Figure 3.3: Database ERD part 3

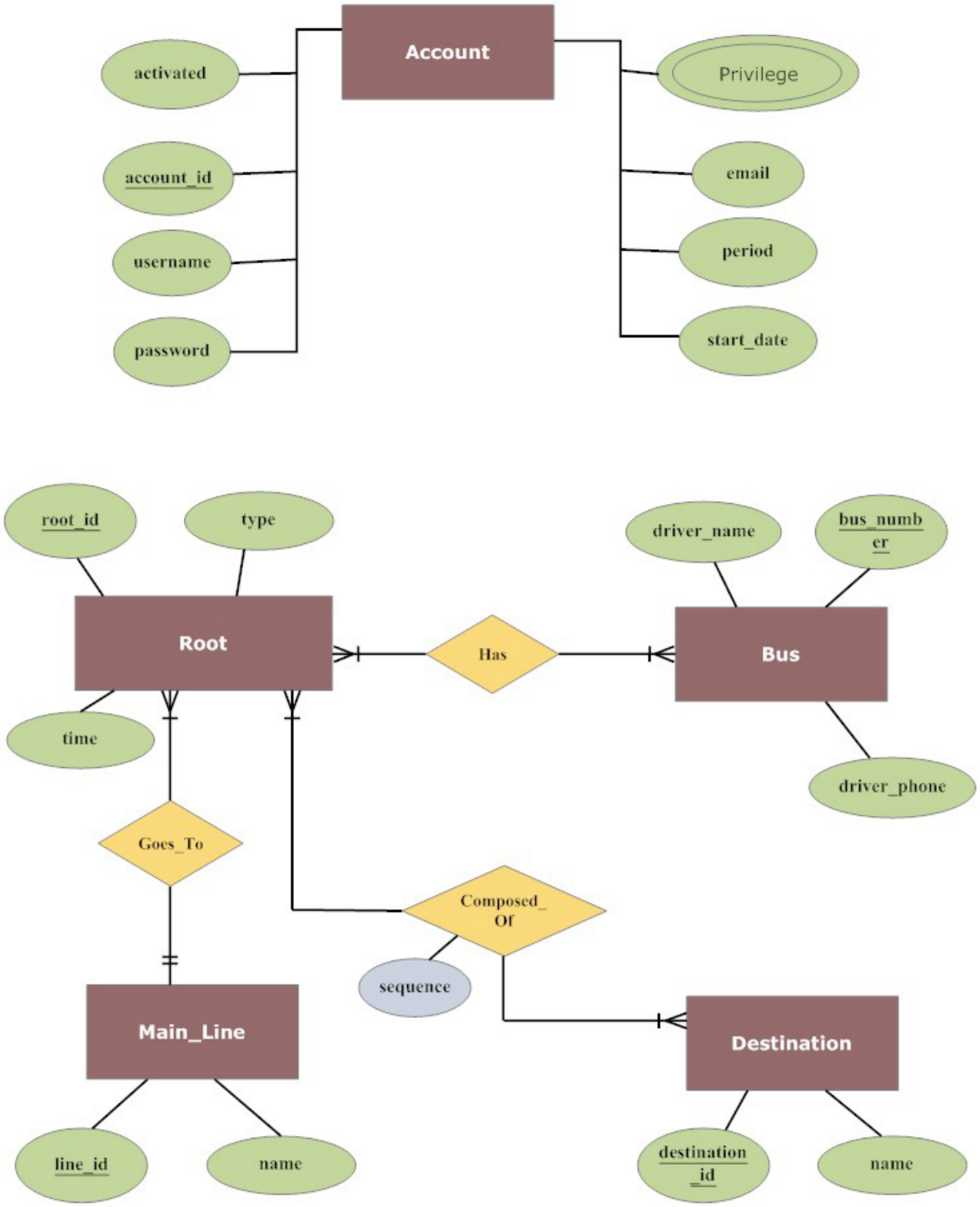


Figure 3.4: Database ERD part 4

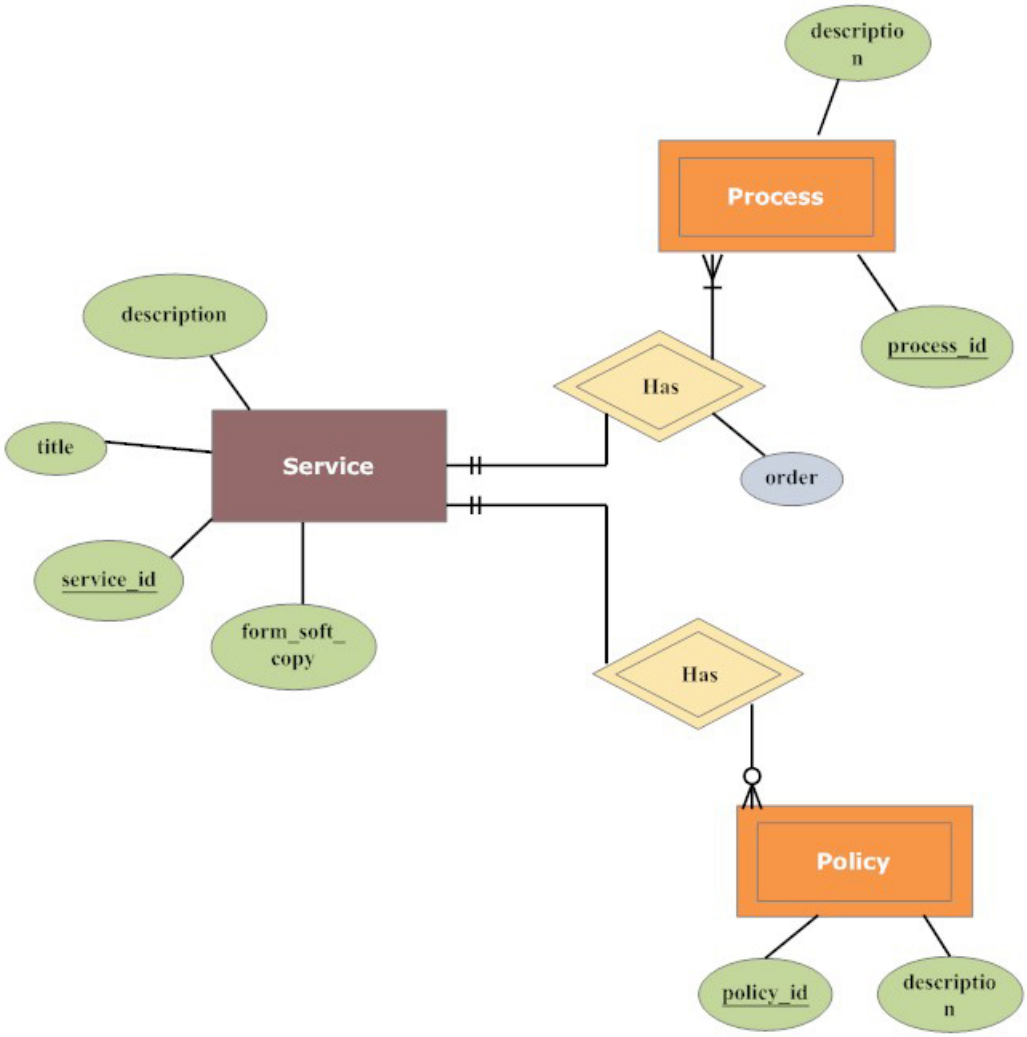


Figure 3.5: Database ERD part 5

After designing the schema all information about the database tables is available also the relationships of the database entities are resolved since now all the keys and foreign keys of each table and of the relations table are known.

As discussed in section 3.1 it‘s constrained to use Microsoft SQL as the RDBMS for that Microsoft SQL SERVER 2005 was chosen for the database implementation. Implemented Database tables are viewed in Appendix A.

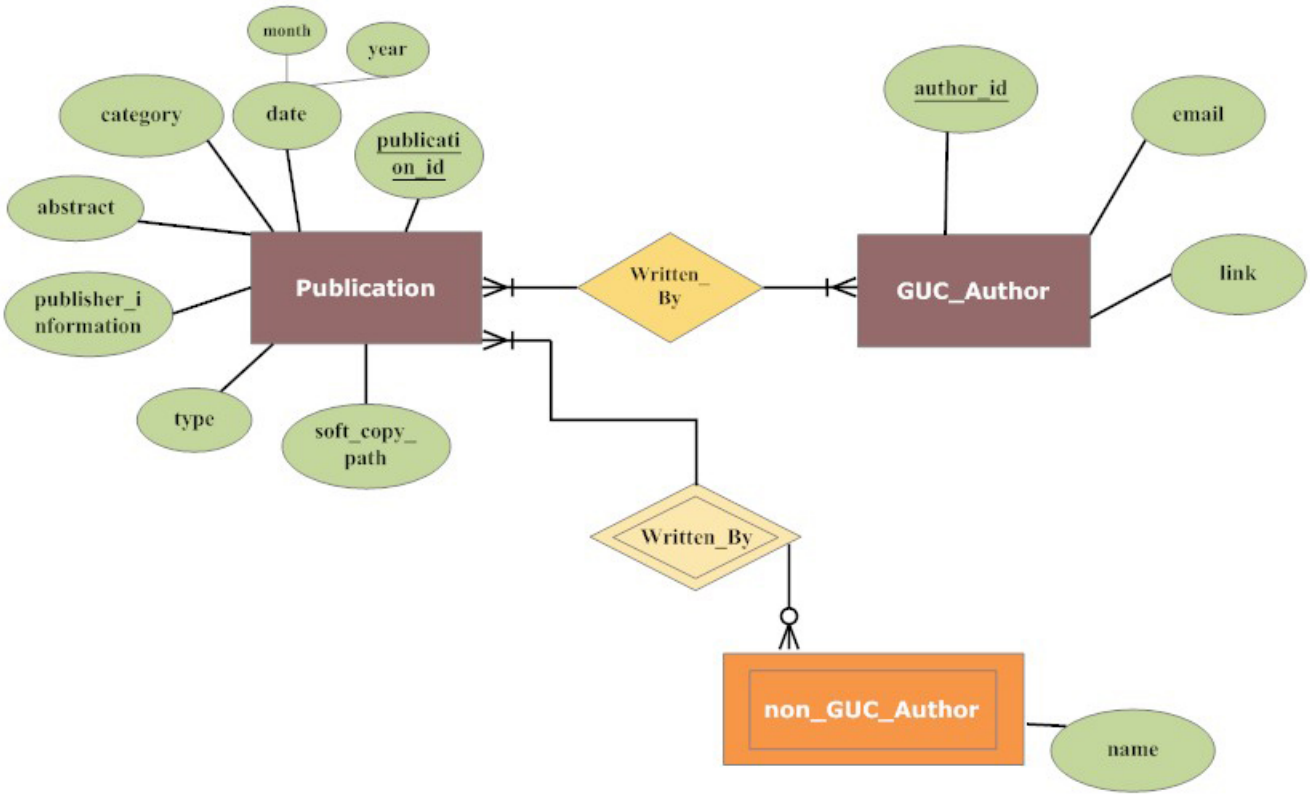


Figure 3.6: Database ERD part 6

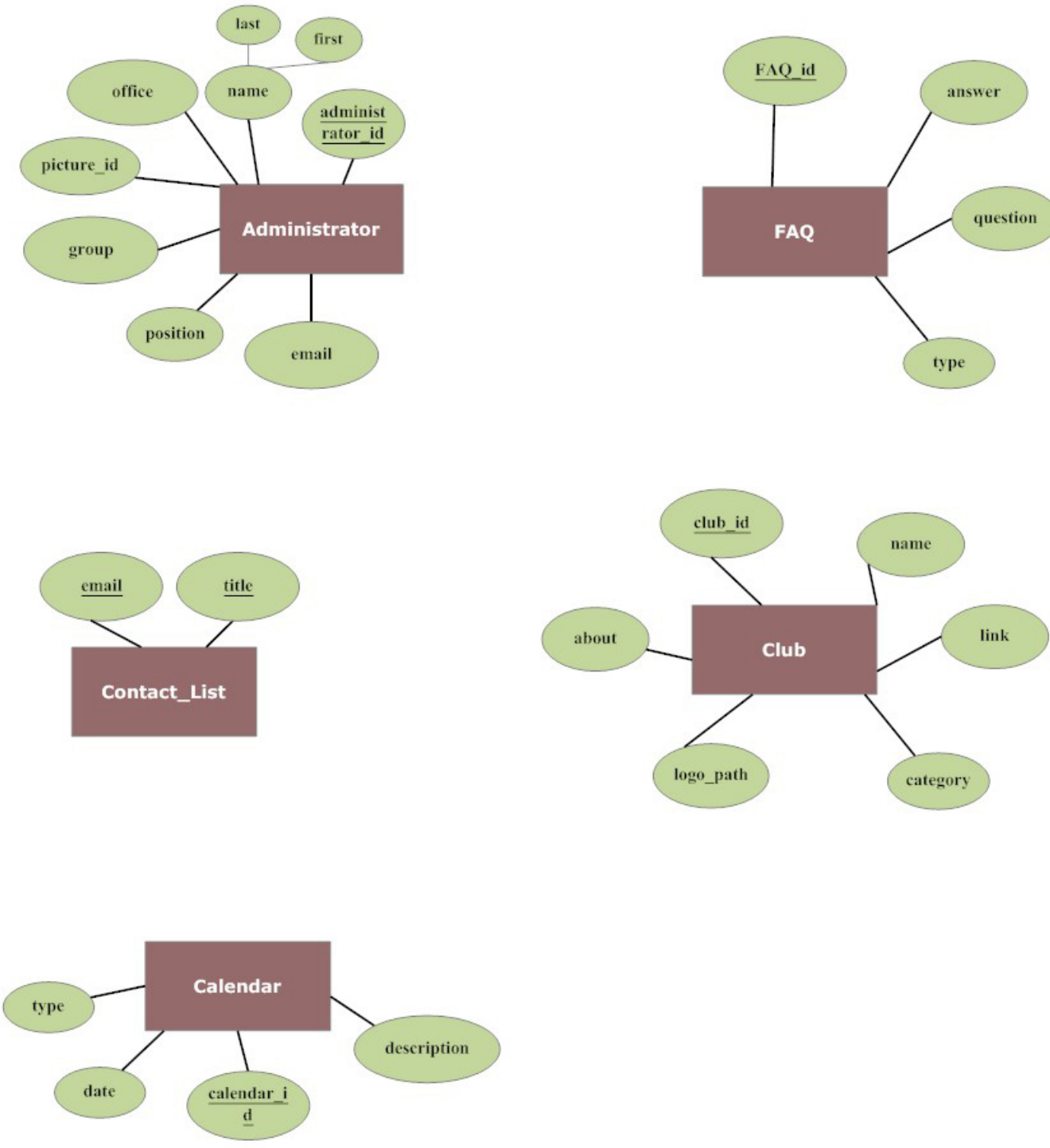


Figure 3.7: Database ERD part 7

# Chapter 4

# Conclusion and Future Work

Developing a website is a challenging task since many technologies are used, and many stockholders are involved.
Managing the contents of a large scale website as GUC website requires much effort and work thats way the coming GUC website offers an easy to use web content management system to manage the website contents. The web content management system implemented offers wide variety of features and services that well help both the visitors and GUC webmaster to have extraordinary experience.
The new GUC website will offer creative design supported with dynamic and interactive components this will guarantees visitors satisfaction.
Finally since the new website offers a clean valid markup it can be viewed safely on different browser and platform and more important it will be search engines friendly.

As for future work the provided web content management system for the GUC webiste can be enhanced in many ways and many more features can be added to it. A number of features and add-ons are considered currently as future developing directions and are described below.

- Extend the web content management system capabilities to offer page layout editing, and automatic generating of website sections.
- Enhance the ticketing system with tracking and visioning functionalities.
- Create publishing repository where publications authored under the GUC title can be asserted and viewed.

One more future work is to work on integrating the GUC website with other GUC faculties and departments website.

# Appendix A

# Database Tables

## Faculty

| Field | Data type | Role | Description |
|---|---|---|---|
| faculty_id | smallint | Primary Key | • Unique identifier for faculty<br>• Auto generated |
| code | nvarchar(20) | | Code of faculty |
| name | nvarchar(100) | | Faculty name |

# FacultyInfo

| Field | Data type | Role | Description |
|---|---|---|---|
| faculty_id | smallint | Primary Key , foreign key from table Faculty | Unique identifier of faculty this information is about |
| intro | nvarchar(MAX) | | • Introduction text about faculty<br>• Allow null |
| description | nvarchar(MAX) | | Description text about faculty |
| link | nvarchar(100) | | • Web link for faculty<br>• Allow null |
| career_opportunities | nvarchar(MAX) | | • Text about career opportunities of this faculty<br>• Allow null |

# Department

| Field | Data type | Role | Description |
|---|---|---|---|
| department_id | smallint | Primary Key | • Unique identifier for department<br>• Auto generated |
| description | nvarchar(MAX) | | • Text Description about department<br>• Allow null |
| name | nvarchar(100) | | Name of department |
| link | nvarchar(100) | | • Web link for department<br>• Allow null |
| code | nvarchar(50) | | • Code of department<br>• Allow null |

# Course

| Field | Data type | Role | Description |
|---|---|---|---|
| course_id | int | Primary Key | • Unique identifier for course<br>• Auto generated |
| description | nvarchar(MAX) | | • Text description about course<br>• Allow null |
| code | nvarchar(50) | | code of course |
| name | nvarchar(100) | | Name of course |
| link | nvarchar(100) | | • Web link for course<br>• Allow null |
| department_id | smallint | Foreign key form table Department | • Unique identifier for department offering the course<br>• Allow null |
| faculty_id | smallint | Foreign key form table Faculty | • Unique identifier for faculty offering the course<br>• Allow null |

# Program

| Field | Data type | Role | Description |
|---|---|---|---|
| program_id | smallint | Primary Key | • Unique identifier for program<br>• Auto generated |
| name | nvarchar(100) | | Name of program |
| degree | nvarchar(100) | | Degree of the program (ie. bachelor) |
| faculty_id | smallint | Foreign key from table Faculty | Unique identifier of faculty offering the program |
| type | nvarchar(50) | | type of program(ie.summer program) |

# ProgramInfo

| Field | Data type | Role | Description |
|---|---|---|---|
| program_id | smallint | Primary Key,foreign key from table Program | Unique identifier of program |
| intro | nvarchar(MAX) | | • Unique identifier for program<br>• Allow null |
| description | nvarchar(MAX) | | Description text about program |
| link | nvarchar(100) | | • Web link for program<br>• Allow null |
| requirements | nvarchar(MAX) | | • Requirements of to join the program<br>• Allow null |

## ProgramCourse

| Field | Data type | Role | Description |
|---|---|---|---|
| course_id | int | Primary Key , Foreign Key from table Course | Unique identifier of the associated course |
| program_id | smallint | Primary Key , Foreign Key from table Program | Unique identifier of the associated program |
| semester | tinyint | | Semester in which the course is taken in this program |

## CoursePrerequisites

| Field | Data type | Role | Description |
|---|---|---|---|
| course_id | int | Primary Key , Foreign Key from table Course | Unique identifier of the associated course |
| pre_course_id | int | Primary Key , Foreign Key from table Course | Unique identifier of the associated course |

## Seminar

| Field | Data type | Role | Description |
|---|---|---|---|
| seminar_id | int | Primary Key | • Unique identifier for seminar<br>• Auto generated |
| name | nvarchar(100) | | Name\Title of seminar |
| description | nvarchar(MAX) | | Description text of seminar |
| start_date | smalldatetime | | • Start date of the seminar<br>• Allow null |
| end_date | smalldatetime | | • End date of the seminar<br>• Allow null |

# Administrator

| Field | Data type | Role | Description |
|---|---|---|---|
| administrator_id | smallint | Primary Key | • Unique identifier of administrator<br>• Auto generated |
| first_name | nvarchar(50) | | First name of administrator |
| last_name | nvarchar(50) | | Last name of administrator |
| email | nvarchar(100) | | • Email address of administrator<br>• Allow null |
| group | nvarchar(100) | | Department or Group the administrator belongs to |
| position | nvarchar(100) | | Position of administrator in the group |
| office | nvarchar(100) | | • Office location of administrator<br>• Allow null |
| picture_name | varchar(32) | | • Picture id of the administrator<br>• Allow null |

# Dean

| Field | Data type | Role | Description |
|---|---|---|---|
| dean_id | smallint | Primary Key | • Unique identifier of dean<br>• Auto generated |
| first_name | nvarchar(50) | | First name of dean |
| last_name | nvarchar(50) | | Last name of dean |
| email | nvarchar(100) | | • Email address of dean<br>• Allow null |
| faculty_id | nvarchar(100) | Foreign key from table Faculty | Unique identifier of Dean's faculty |
| office | nvarchar(100) | | • Office location of dean<br>• Allow null |
| picture_id | nvarchar(100) | | • Picture id of the dean<br>• Allow null |

# FAQType

| Field | Data type | Role | Description |
|---|---|---|---|
| type_id | tinyint | Primary Key | • Unique identifier of FAQ type<br>• Auto generated |
| name | nvarchar(50) | | Name\Title of type which FAQ my belong to (ie. admission FAQ) |

# FAQ

| Field | Data type | Role | Description |
|---|---|---|---|
| FAQ_id | int | Primary Key | • Unique identifier of FAQ entry<br>• Auto generated |
| question | nvarchar(MAX) | | Question of the FAQ entry |
| answer | nvarchar(MAX) | | Answer associated with the question |
| type_id | tinyint | Foreign Key from table FAQType | Unique identifier of type which FAQ belongs to |

# ClubCategory

| Field | Data type | Role | Description |
|---|---|---|---|
| category_id | tinyint | Primary Key | • Unique identifier of club category.<br>• Auto generated |
| name | nvarchar(50) | | Name of the club category (ie. Academic) |

# Club

| Field | Data type | Role | Description |
|---|---|---|---|
| club_id | int | Primary Key | • Unique identifier of club<br>• Auto generated |
| name | nvarchar(50) | | Name of club |
| category_id | tinyint | Primary Key , Foreign Key from table ClubCategory | Unique identifier of the category which club belongs to |
| about | nvarchar(MAX) | | • Text about the club<br>• Allow null |
| link | nvarchar(100) | | • Web link for the club<br>• Allow null |
| logo_name | varchar(32) | | • Logo of the club<br>• Allow null<br>• Auto generated |

## Service

| Field | Data type | Role | Description |
|---|---|---|---|
| service_id | smallint | Primary Key | • Unique identifier of service<br>• Auto generated |
| title | nvarchar(100) | | Title of the service offered |
| description | nvarchar(MAX) | | • Text description of service<br>• Allow null |
| form_soft_copy | nvarchar(100) | | • Related form that can be downloaded with the service<br>• Allow null |

## ServiceProcess

| Field | Data type | Role | Description |
|---|---|---|---|
| process_id | int | Primary Key | • Unique identifier of process<br>• Auto generated |
| service_id | smallint | Foreign key from table Service | Unique identifier for the service that takes this process |
| description | nvarchar(MAX) | | Text description of the process |
| order | tinyint | | • Order of this process with respect to other processes for the service<br>• Allow null |

## ServicePolicy

| Field | Data type | Role | Description |
|---|---|---|---|
| policy_id | int | Primary Key | • Unique identifier of policy<br>• Auto generated |
| service_id | smallint | Foreign key from table Service | Unique identifier for the service that have this policy |
| description | nvarchar(MAX) | | Text description of the policy |

## Contact

| Field | Data type | Role | Description |
|---|---|---|---|
| title | nvarchar(50) | Primary Key | Name\Title of the contact person or department |
| email | nvarchar(100) | Primary Key | Link of the contact email |

## PublicationCategory

| Field | Data type | Role | Description |
|---|---|---|---|
| category_id | tinyint | Primary Key | • Unique identifier of publication category.<br>• Auto generated |
| name | nvarchar(50) | | Name of the publication category (ie. Computer Theory) |

## PublicationType

| Field | Data type | Role | Description |
|---|---|---|---|
| type_id | tinyint | Primary Key | • Unique identifier of publication type.<br>• Auto generated |
| name | nvarchar(50) | | Name of the publication type (ie. Academic Journal) |

# GUCAuthor

| Field | Data type | Role | Description |
|---|---|---|---|
| author_id | int | Primary Key | • Unique identifier of the auhtor<br>• Auto generated |
| first_name | nvarchar(100) | | First Name of the author |
| middle_name | nvarchar(100) | | • Middle Name of the author<br>• Allow null |
| last_name | nvarchar(100) | | Last Name of the author |
| email | nvarchar(100) | | • Email of Author<br>• Allow null |
| link | nvarchar(100) | | • Link to the author's page<br>• Allow null |

# Publication

| Field | Data type | Role | Description |
|---|---|---|---|
| publication_id | int | Primary Key | • Unique identifier of the publication<br>• Auto generated |
| title | nvarchar(200) | | Name Title of the publication |
| abstract | nvarchar(MAX) | | • Abstract of the publication<br>• Allow null |
| publisher_info | nvarchar(200) | | • Information about of the event's publisher( ie : conference name , journal title and page)<br>• Allow null |
| year | tinyint | | Year of the publication |
| month | tinyint | | Month of the publication |
| category_id | tinyint | Foreign Key from table PublicationCategory | Unique identifier of the publication category |
| type_id | tinyint | Foreign Key from table PublicationType | Unique identifier of the publication type |

# nonGUCAuthor

| Field | Data type | Role | Description |
|---|---|---|---|
| publication_id | int | Primary Key , Foreign Key from table Publication | Unique identifier of the associated publication |
| author_name | nvarchar(100) | Primary Key | Unique identifier of the associated external author |

# PublicationAuthor

| Field | Data type | Role | Description |
|---|---|---|---|
| publication_id | int | Primary Key , Foreign Key from table Publication | Unique identifier of the associated publication |
| author_id | int | Primary Key , Foreign Key from table GUCAuthor | Unique identifier of the associated GUC author |

# NewsEventCategory

| Field | Data type | Role | Description |
|---|---|---|---|
| category_id | tinyint | Primary Key | • Unique identifier of news\event category.<br>• Auto generated |
| name | nvarchar(50) | | Title\Name of category (ie. Academic) |

# News

| Field | Data type | Role | Description |
|---|---|---|---|
| news_id | int | Primary Key | • Unique identifier for News headline<br>• Auto generated |
| title | nvarchar(200) | | Title of News to be displayed |
| summary | nvarchar(MAX) | | • Summary of the news headline.<br>• Allow null |
| description | nvarchar(MAX) | | Full text of the news headline |
| post_date | smalldatetime | | • Date news posted on website.<br>• Automatically generated |
| category_id | tinyint | Foreign Key from table NewsEventCategory | Unique identifier of the category type of the news |
| news_year | tinyint | | • Year news headline happened.<br>• Allow null |
| news_month | tinyint | | • Month news headline happened.<br>• Allow null |
| news_day | tinyint | | • Day news headline happened.<br>• Allow null |
| thumbnail | varchar(32) | | • Thumbnail image of news<br>• Allow null |
| spotlight | bit | | • Used to define news headline as Spotlight,1 is spotlight 0 its not<br>• Default=0 |

## NewsImage

| Field | Data type | Role | Description |
|---|---|---|---|
| image_id | int | Primary Key | • Unique identifier for image<br>• Auto generated |
| description | nvarchar(MAX) | | Text Description associated with the image |
| image_name | varchar(32) | | Path where image stored |
| news_id | int | Foreign Key from table News | Unique identifier for news headline thats display the image |

# RequestedNews

| Field | Data type | Role | Description |
|---|---|---|---|
| requested_news_id | int | Primary Key | • Unique identifier for News headline<br>• Auto generated |
| title | nvarchar(200) | | Title of News to be displayed |
| summary | nvarchar(MAX) | | • Summary of the news headline.<br>• Allow null |
| description | nvarchar(MAX) | | Full text of the news headline |
| category_id | tinyint | Foreign Key from table NewsEventCategory | Unique identifier of the category type of the news |
| post_date | smalldatetime | | • Date news requested.<br>• Automatically generated |
| news_year | tinyint | | • Year news headline happened.<br>• Allow null |
| news_month | tinyint | | • Month news headline happened.<br>• Allow null |
| news_day | tinyint | | • Day news headline happened.<br>• Allow null |
| thumbnail | varchar(32) | | • Thumbnail image of news<br>• Allow null |

# RequestedNewsImage

| Field | Data type | Role | Description |
|---|---|---|---|
| image_id | int | Primary Key | • Unique identifier for image<br>• Auto generated |
| description | nvarchar(MAX) | | Text Description associated with the image |
| image_name | varchar(32) | | Path where image stored |
| requested_news_id | int | Foreign Key from table RequestedNews | Unique identifier for news headline thats display the image |

# RelatedNews

| Field | Data type | Role | Description |
|---|---|---|---|
| news_id | int | Primary Key , Foreign Key from table News | Unique identifier of the associated news |
| related_news_id | int | Primary Key , Foreign Key from table News | Unique identifier of the associated news |

# Event

| Field | Data type | Role | Description |
|---|---|---|---|
| event_id | int | Primary Key | • Unique identifier of the event<br>• Auto generated |
| title | nvarchar(200) | | Name title of the event |
| summary | nvarchar(MAX) | | • Summary of information about the event<br>• Allow null |
| description | nvarchar(MAX) | | Full information about the event |
| location | nvarchar(50) | | • Location the event is held<br>• Allow null |
| on_campus | bit | | 1 if event on campus 0 if not |
| post_date | smalldatetime | | • Date event posted on website.<br>• Automatically generated |
| category_id | tinyint | Foreign Key from table NewsEventCategory | identifier of the category type of the event |
| start_year | tinyint | | • Year event starts<br>• Allow null |
| start_month | tinyint | | • Month event starts<br>• Allow null |
| start_day | tinyint | | • Month event starts<br>• Allow null |
| end_year | tinyint | | • Year event Ends<br>• Allow null |
| end_month | tinyint | | • Month event ends<br>• Allow null |

## EventImage

| Field | Data type | Role | Description |
|---|---|---|---|
| image_id | int | Primary Key | • Unique identifier for image<br>• Auto generated |
| description | nvarchar(MAX) | | • Description text associated with the image<br>• Allow null |
| image_name | varchar(32) | | Path where image stored |
| event_id | int | Foreign Key from table Event | Unique identifier of Event that images are related to |

## RelatedVideo

| Field | Data type | Role | Description |
|---|---|---|---|
| video_id | int | Primary Key , Foreign Key from table Video | Unique identifier of the associated Video |
| related_video_id | int | Primary Key , Foreign Key from table Video | Unique identifier of the associated Video |

## RelatedEvent

| Field | Data type | Role | Description |
|---|---|---|---|
| event_id | int | Primary Key , Foreign Key from table Event | Unique identifier of the associated event |
| related_event_id | int | Primary Key , Foreign Key from table Event | Unique identifier of the associated event |

## RelatedNewsEvent

| Field | Data type | Role | Description |
|---|---|---|---|
| news_id | int | Primary Key , Foreign Key from table News | Unique identifier of the associated news |
| event_id | int | Primary Key , Foreign Key from table Event | Unique identifier of the associated event |

## Video

| Field | Data type | Role | Description |
|---|---|---|---|
| video_id | int | Primary Key | • Unique identifier of Video<br>• Auto generated |
| title | nvarchar(200) | | Title of the Video item |
| description | nvarchar(MAX) | | • Text description of the video.<br>• Allow null |
| name | varchar(32) | | Path where video stored |
| category_id | tinyint | Foreign Key from table AlbumCategory | Unique identifier of the category which video belongs to |

## NewsVideo

| Field | Data type | Role | Description |
|---|---|---|---|
| news_id | int | Primary Key , Foreign Key from table News | Unique identifier of the associated news |
| video_id | int | Primary Key , Foreign Key from table Video | Unique identifier of the associated video |

## EventVideo

| Field | Data type | Role | Description |
|---|---|---|---|
| event_id | int | Primary Key , Foreign Key from table Event | Unique identifier of the associated event |
| video_id | int | Primary Key , Foreign Key from table Video | Unique identifier of the associated video |

# EventAgenda

| Field | Data type | Role | Description |
|---|---|---|---|
| agenda_id | int | Primary Key ,Primary Key | • Unique identifier of agenda item<br>• Auto generated |
| event_id | int | Foreign Key from table Event | Unique identifier of the associated event |
| description | nvarchar(200) | | Description of the happening in event agenda |
| date | smalldatetime | | date to be scheduled |
| start_time | nvarchar(50) | | • Start time of the event schedule on the giving date<br>• Allow null |
| end_time | nvarchar(50) | | • End time of the event schedule on the giving date<br>• Allow null |

# AlbumCategory

| Field | Data type | Role | Description |
|---|---|---|---|
| category_id | tinyint | Primary Key | • Unique identifier of album category.<br>• Auto generated |
| name | nvarchar(50) | | Name of the album category (ie. event) |

# Album

| Field | Data type | Role | Description |
|---|---|---|---|
| album_id | int | Primary Key | • Unique identifier of the album<br>• Auto generated |
| title | nvarchar(200) | | Title\Name of the album |
| description | nvarchar(MAX) | | • Text description about the album<br>• Allow Null |
| creation_date | smalldatetime | | • Date of posting the album on website<br>• Auto generated |
| category_id | tinyint | Foreign Key from table AlbumCategory | Unique identifier of the album category |

# RelatedAlbum

| Field | Data type | Role | Description |
|---|---|---|---|
| album_id | int | Primary Key , Foreign Key from table Album | Unique identifier of the associated album |
| related_album_id | int | Primary Key , Foreign Key from table Album | Unique identifier of the associated album |

# NewsAlbum

| Field | Data type | Role | Description |
|---|---|---|---|
| album_id | int | Primary Key , Foreign Key from table Album | Unique identifier of the associated album |
| news_id | int | Primary Key , Foreign Key from table News | Unique identifier of the associated news |

# EventAlbum

| Field | Data type | Role | Description |
|---|---|---|---|
| album_id | int | Primary Key , Foreign Key from table Album | Unique identifier of the associated album |
| event_id | int | Primary Key , Foreign Key from Event | Unique identifier of the associated event |

# Picture

| Field | Data type | Role | Description |
|---|---|---|---|
| picture_id | int | Primary Key | • Unique identifier of image<br>• Auto generated |
| description | nvarchar(MAX) | | • Description text of the image<br>• Allow null |
| name | varchar(32) | | Path where image stored |
| order | smallint | | the order of the image in the album |
| album_id | int | Foreign Key from table Album | Unique identifier of album which image belongs to |

# CalendarType

| Field | Data type | Role | Description |
|---|---|---|---|
| type_id | tinyint | Primary Key | • Unique identifier of calendar date types<br>• Auto generated |
| name | nvarchar(50) | | Name\Title of possible category of date on calendar (ie. Academic) |

# Calendar

| Field | Data type | Role | Description |
|---|---|---|---|
| calendar_id | int | Primary Key | • Unique identifier of calendar date entry<br>• Auto generated |
| date | smalldatetime | | Date of calendar entry |
| description | nvarchar(MAX) | | Description of calendar entry |
| type_id | tinyint | Foreign Key from table CalendarType | Unique identifier of the type of date entry |

# Candidate

| Field | Data type | Role | Description |
|---|---|---|---|
| candidate_id | int | Primary Key | • Unique identifier for candidate<br>• Auto generated |
| first_name | nvarchar(50) | | First name of candidate |
| middle_name | nvarchar(50) | | Middle name of candidate |
| last_name | nvarchar(50) | | Last name of candidate |
| address | nvarchar(512) | | Address of candidate |
| phone | nvarchar(50) | | • Phone number of candidate<br>• Allow null |
| mobile | nvarchar(50) | | Mobile number of candidate |
| email | nvarchar(100) | | Email of candidate |
| birth_year | tinyint | | Birth year of candidate |
| birth_month | tinyint | | Birth month of candidate |
| birth_day | tinyint | | Birth day of candidate |
| CV_Path | varchar(32) | | • Path of CV pdf file of candidate<br>• Allow null |
| username | nvarchar(20) | | Username of candidate account |
| password | nvarchar(20) | | • Password of candidate account.<br>• Encrypted |
| activated | bit | | • 1 account activated 0 not activated<br>• Default = 0 |

## EducationLevel

| Field | Data type | Role | Description |
|---|---|---|---|
| level_id | tinyint | Primary Key | • Unique identifier of Education_Level<br>• Auto generated |
| description | nvarchar(50) | | Text description of the level of education (ie. PhD) |

## FieldOfStudy

| Field | Data type | Role | Description |
|---|---|---|---|
| field_id | tinyint | Primary Key | • Unique identifier of field of study<br>• Auto generated |
| description | nvarchar(50) | | Name\Description of the field (ie.Computer Science) |

## CandidateEducation

| Field | Data type | Role | Description |
|---|---|---|---|
| candidate_id | int | Primary Key , Foreign Key from table Candidate | Unique identifier of candidate |
| order | tinyint | Primary Key | Order of this level according to candidate preferences |
| institution | nvarchar(100) | | Institution where candidate acquired this level |
| field_id | tinyint | Foreign key from table FieldOfStudy | Unique identifier for the field in which the education level where acquired |
| level_id | tinyint | Foreign key from table EducationLevel | Unique identifier for the level of education level |
| GPA | real | | GPA acquired by candidate in the level of education |
| graduation_year | tinyint | | Year where this level of education is acquired |
| graduation_month | tinyint | | Month where this level of education is acquired |

# JobCategory

| Field | Data type | Role | Description |
|---|---|---|---|
| category_id | tinyint | Primary Key | • Unique identifier of job category.<br>• Auto generated |
| title | nvarchar(50) | | Name\Title of the job category (ie.Human Resources) |

# CandidateExperience

| Field | Data type | Role | Description |
|---|---|---|---|
| candidate_id | int | Primary Key , Foreign Key from table Candidate | Unique identifier of work experience |
| order | tinyint | Primary Key | Order of this work experience according to candidate preferences |
| employer | nvarchar(100) | | Name of Company\Employer where the candidate had this experience |
| current_job | bit | | 1 if current job 0 if not |
| start_year | tinyint | | Year experience started |
| start_month | tinyint | | Month experience started |
| end_year | tinyint | | Year experience ended |
| end_month | tinyint | | Month experience ended |
| category_id | tinyint | Foreign Key from table JobCategory | Unique identifier of experience job category |

# Job

| Field | Data type | Role | Description |
|---|---|---|---|
| job_id | int | Primary Key | • Unique identifier for Job<br>• Auto generated |
| title | nvarchar(100) | | Tile of job |
| description | nvarchar(MAX) | | Text description of job |
| post_date | smalldatetime | | • Date job vacancy posted on the website<br>• Auto generated |
| code | nvarchar(50) | | • Reference code of job vacancy<br>• Allow null |
| vacant | bit | 1 if job currently vacent 0 if not | |
| responsible_email | nvarchar(100) | | Email address of responsible of this job |
| category_id | tinyint | Foreign Key from table JobCategory | Unique identifier of job category |

# CandidateJob

| Field | Data type | Role | Description |
|---|---|---|---|
| job_id | int | Primary Key , Foreign Key from table Job | Unique identifier of associated job |
| candidate_id | int | Primary Key , Foreign Key from table Candidate | Unique identifier of associated candidate |

# JobRequirement

| Field | Data type | Role | Description |
|---|---|---|---|
| requirement_id | int | Primary Key | • Unique identifier for job requirement<br>• Auto generated |
| job_id | int | Foreign Key from table Job | Unique identifier of job |
| description | nvarchar(200) | | Text description for requirement of the job |

# JobQualification

| Field | Data type | Role | Description |
|---|---|---|---|
| qualification_id | int | Primary Key | • Unique identifier for job qualification<br>• Auto generated |
| job_id | int | Foreign Key from table Job | Unique identifier of job |
| description | nvarchar(200) | | Text description for qualification of the job |

# Privilege

| Field | Data type | Role | Description |
|---|---|---|---|
| privilege_id | tinyint | Primary Key | • Unique identifier for account privilege<br>• Auto generated |
| description | nvarchar(50) | | Name\Title of privilege (ie. Administrator) |

# Account

| Field | Data type | Role | Description |
|---|---|---|---|
| account_id | int | Primary Key | • Unique identifier for account<br>• Auto generated |
| username | nvarchar(20) | | username for the account |
| password | nvarchar(20) | | • password for the account<br>• Encrypted |
| first_name | nvarchar(50) | | First name of dean |
| last_name | nvarchar(50) | | Last name of dean |
| email | nvarchar(50) | | email of account holder |
| activated | bit | | • 1 account activated 0 not activated<br>• Default = 0 |
| start_date | smalldatetime | | date where account activated |
| period | smallint | | • Number of days account will be active for from date started<br>• Default =0 |

# Account_Privilege

| Field | Data type | Role | Description |
|---|---|---|---|
| account_id | int | Primary Key , Foreign Key from table Account | Unique identifier for associated account |
| privilege_id | tinyint | Primary Key , Foreign Key from table Privilege | Unique identifier for associated privilege |

# Bus

| Field | Data type | Role | Description |
|---|---|---|---|
| bus_number | smallint | Primary Key | Unique identifier of Bus |
| driver_name | nvarchar(50) | | • Name of bus driver<br>• Allow null |
| driver_phone | nvarchar(50) | | • Phone number of bus driver<br>• Allow null |

# Line

| Field | Data type | Role | Description |
|---|---|---|---|
| line_id | smallint | Primary Key | • Unique identifier of line<br>• Auto generated |
| name | nvarchar(50) | | Name of main line (ie. Haram) |

# Root

| Field | Data type | Role | Description |
|---|---|---|---|
| root_id | smallint | Primary Key | • Unique identifier of Root<br>• Auto generated |
| time | nvarchar(50) | | Time where this root is taken |
| type | nvarchar(20) | | type of root (ie. to campus) |
| line_id | smallint | Foreign Key from table Line | |

# BusRoot

| Field | Data type | Role | Description |
|---|---|---|---|
| bus_number | smallint | Primary Key , Foreign Key from table Bus | Unique identifier of the associated bus |
| root_id | smallint | Primary Key , Foreign Key from table Root | Unique identifier of the associated root |

# Destination

| Field | Data type | Role | Description |
|---|---|---|---|
| destination_id | smallint | Primary Key | • Unique identifier of destination<br>• Auto generated |
| name | nvarchar(50) | | name of destination the bus stops at (ie. Makram Abeid) |

# RootDestination

| Field | Data type | Role | Description |
|---|---|---|---|
| root_id | smallint | Primary Key , Foreign Key from table Root | Unique identifier of the associated root |
| destination_id | smallint | Primary Key,Foreign Key from table Destination | Unique identifier of the associated destination |
| sequence | tinyint | | • The sequence of destinations stops in the root<br>• Allow null |