

**REPUBLIC OF IRAQ**

**MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH**

**AL-QADISIYAH UNIVERSITY**

**COLLEGE OF COMPUTER SCIENCES AND IT**

**MULTIMEDIA DEPARTMENT**

**SYSTEM TO FOLLOW-UP OF ISSUED AND RECEIVED MAIL**

**2018-2019**



**جمهورية العراق**

**وزارة التعليم العالي والبحث العلمي**

**جامعة القادسية**

**كلية علوم الحاسبات وتكنولوجيا المعلومات**

**قسم الوسائط المتعددة**

**نظام متابعة البريد الوارد والصادر**

**2018-2019**

**بسم الله الرحمن الرحيم**

((وَإِنِّي مُرْسِلَةٌ إِلَيْهِمْ بِهَدِيَّةٍ فَنَاظِرَةٌ بِمَ يَرْجِعُ الْمُرْسَلُونَ (35) فَلَمَّا جَاءَ سُلَيْمَانَ قَالَ أَتُمِدُّونَنِ بِمَالٍ فَمَا آتَانِيَ اللَّهُ خَيْرٌ مِمَّا آتَاكُمْ بَلْ أَنْتُمْ بِهَدِيَّتِكُمْ تَفْرَحُونَ (36) ارْجِعْ إِلَيْهِمْ فَلَنَأْتِيَنَّهُمْ بِجُنُودٍ لَا قِبَلَ لَهُمْ بِهَا وَلَنُخْرِجَنَّهُمْ مِنْهَا أَذِلَّةً وَهُمْ صَاغِرُونَ (37) ))

**REPUBLIC OF IRAQ**

**MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH**

**AL-QADISIYAH UNIVERSITY**

**COLLEGE OF COMPUTER SCIENCES AND IT**

**MULTIMEDIA DEPARTMENT**

**GRADUATION PROJECT REPORT**

**SYSTEM TO FOLLOW-UP OF ISSUED AND RECEIVED MAIL**

**by**

|  |  |
| --- | --- |
| **Abbas Khaled Hamza** | **Majed Hashim Serhan** |
| **Mohammed Rahman Na’ama** | **Ghadeer Jasim Kadim** |
| **Ammar Hameed Razaq** |

**ADVISER**

**ASSISTANT LECTURE. ALI HAKEM**

**AL-QADISIYAH, 2019**

**جمهورية العراق**

**وزارة التعليم العالي والبحث العلمي**

**جامعة القادسية**

**كلية علوم الحاسبات وتكنولوجيا المعلومات**

**قسم الوسائط المتعددة**

**تقرير مشروع التخرج**

**نظام متابعة البريد الوارد والصادر**

**اعداد**

|  |  |
| --- | --- |
| **عباس خالد حمزة** | **ماجد هاشم سرحان** |
| **محمد رحمن نعمة** | **غدير جاسم كاظم** |
| **عمار حميد رزاق** |

**اشراف**

**م.م. علي حاكم جبر**

**القادسية 2019**

**REPUBLIC OF IRAQ**

**AL-QADISIYAH UNIVERSITY**

**MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH**

**COLLEGE OF COMPUTER SCIENCES AND IT**

MULTIMEDIA DEPARTMENT

**SYSTEM TO FOLLOW-UP OF ISSUED AND RECEIVED MAIL**

A project submitted by Abbas Khaled Hamza, Majed Hashim Serhan, Mohammed Rahman Na’ama, Ghadeer Jasim Kadim, and Ammar Hameed Razaq, in partial fulfilment of the requirements for the degree of  BACHELOR'S OF COMPUTER SCIENCE, is approved by the committee in Department of Multimedia.

**Project Adviser**

Assistant Lecturer. Ali Hakem

Al-Qadisiyah University

**Approved By the Examining Committee**

ACKNOWLEDGEMENTS

First, I would like to thank God for giving me strength and patience to work on this thesis. It would not have been possible without God guidance and support.

I would like to thank my supervisor, Assist. Lecture. Ali Hakem, for the opportunity to work under his guidance, for the suggestions and advices on how to perform a master thesis; without his guidance and present support, this master thesis would not have been possible. Under his guidance, I could learn on working methods and improve my performance, which allows me to perform this thesis.

Also, I want to thank all my family mother, brothers, sisters, children, who were a great source of encouragement and motivation.

April 2019

 Authors

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS xi

TABLE OF CONTENTS xii

LIST OF FIGURES xiii

ABSTRACT xiv

الخلاصة xv

CHAPTER 1

INTRODUCTION 1

1.1 Introduction 1

1.3 Report Outline 3

CHAPTER 2

SYSTEM ANAYSIS 4

2.1 Method of Problem’s Solution 4

2.2 Programming Environments 5

2.3 Program’s Structure Analyzing and GUI Constructing 6

2.4 Database Connections and Code-Implementation 6

2.5 Order to database server 8

2.5.1 Retrieving data from the database 8

2.5.2 Saving Data Into The Database 8

2.5.3 Deleting Data From The Database 8

2.5.4 Update Data From The Database 9

CHAPTER 3

CONCLUSION AND FUTURE WORK 10

3.1 Conclusion and Future Works 10

APENDIX A

Graphical User Interface (GUI) 12

APENDIX B

Command of Email processing 16

REFERENCES 20

LIST OF FIGURES

Page

[Figure 2.1 Database information systems - principle scheme 2](#_Toc6680604)

[Figure 2.1 Preparing to get a connection established 7](#_Toc6680605)

[Figure 1: Main Interface 12](#_Toc6680606)

[Figure 2: Login 13](#_Toc6680607)

[Figure 3: Add new Email 13](#_Toc6680608)

[Figure 3: Search 14](#_Toc6680609)

[Figure 4: Edit Email 15](#_Toc6680610)

ABSTRACT

SYSTEM TO FOLLOW-UP OF ISSUED AND RECEIVED MAIL

|  |  |
| --- | --- |
| Abbas Khaled Hamza | Majed Hashim Serhan |
| Mohammed Rahman Na’ama | Ghadeer Jasim Kadim |
| Ammar Hameed Razaq |

Department of Multimedia

B.S.c Graduate Report

Adviser

Assistant Lecture. Ali Hakem

This report includes a development presentation of an information system for managing the issued and received mail data within a director or organization. The system as such as it has been developed is called system to follow-up of issued and received mail. It consists of functionally related GUI (application program) and database.

The choice of the programming tools is individual and particular.

**Keywords**: Information system, Database system, Database mangement System

AL-QADISIYAH UNIVERSITY

**COLLEGE OF COMPUTER SCIENCES AND IT**

**MULTIMEDIA DEPARTMENT**

الخلاصة

 نظام متابعة البريد الوارد والصادر

|  |  |
| --- | --- |
| **عباس خالد حمزة** | **ماجد هاشم سرحان** |
| **محمد رحمن نعمة** | **غدير جاسم كاظم** |
| **عمار حميد رزاق** |

**قسم الوسائط المتعددة**

**تقرير مشروع تخرج البكلوريوس**

**اشراف**

**م.م. علي حاكم جبر**

تضمن هذا التقرير عرضًا تقديميًا لتطوير نظام معلومات لإدارة بيانات البريد الصادرة والمستلمة داخل مديرية أو مؤسسة. يسمى النظام كما تم تطويره بالنظام لمتابعة البريد الصادر والمستلم. وهو يتألف من واجهة المستخدم الرسومية ذات الصلة وظيفيا (برنامج التطبيق) وقاعدة البيانات.

اختيار أدوات البرمجة الفردية وخاصة.

الكلمات المفتاحية: نظام المعلومات ، نظام قواعد البيانات ، نظام إدارة قواعد البيانات

جامعة القادسية

كلية علوم الحاسوب وتكنولوجيا المعلومات

قسم الوسائط المتعدد

CHAPTER 1

# INTRODUCTION

1. 1. **Introduction**

Most of the contemporary Information systems are based on the Database technology as a collection of logically related data, and DBMS as a software system allowing the users to define, create, maintain and control access to the database. The process of constructing such kind of systems is not so simple. It involves a mutual development of application program and database. The application program is actually the bridge between the users and the database, where the data is stored. Thus, the well-developed application program and database are very important for the reliability, flexibility and functionality of the system. The so defined systems differentiate to each other and their development comprises a great variety of tasks to be resolved and implemented. The basic idea can be depicted in Figure 1.1 below:



Figure 2.1 Database information systems - principle scheme

Information system suggests a computer technology to be used in order to provide information to users in an organization (for instance), as for the purposes of data transformation into useful information; computer hardware and software are designed and used [2]. A particular case is the Human Resources Information System development. This kind of systems are responsible for storing data of the email within an organization and generating reports upon request. Such kind of system could be integrated with other Information systems or modules: Accounting Information System (AIS) – designed to transform financial data into information, or Management Information System (MIS) that provides decision-oriented information to managers, and so on… “Organizations depend on Information Systems in order to stay competitive. Productivity, which is crucial to staying competitive, can be increased through better Information Systems.” [2].

* 1. **The objective of the Project**

This system should consist of an application program, on one hand, and a database (repository of data) on the other. The program should perform the basic operations upon the database as retrieving, inserting, updating and deleting data. Any additional functionality is a goal of further module development. It is a kind of strategy to start the development from designing and constructing the database, as this structure will determine the further structure of the application program. The logical database model (tables, their content and the relationships between them) should respond to the given task and cover the basic requirements. The Interface of the program should be user-friendly, and the program should be as easy for use as it is possible. Both controls and forms should logically and functionally be related within the program and fully respond to the structure of the database. Another problem is establishing the connections with the database, every time, when a query is needed to be performed upon it. Exception-handling should also be taken into an account during the system’s development due to eventual exceptions that may occur.

* 1. **Report Outline**

Chapter 1 : Introduction

Chapter 2 : System Analysis

Chapter 3 : Conclusion and Future Work

Appendix A : GUI

Appendix B : Commands of Email Processing

 CHAPTER 2

# SYSTEM ANALYSIS

##  Method of Problem’s Solution

At the very commencement, I proceeded to a decision to carry out the development of my task into the following steps:

1. **Exploring the available development environments and techniques**

There is a lot of programming environments available to be used for such kind of elaborations. The point is to choose such an environment that we will be able to operate with in a convenient and easy way. This is more or less optional and individual process, that depends on the developer’s experience as well.

1. **Database Analyzing**

It concerns all of the demands, put upon the database content and its functionality. The database should be designed and implemented in a way that the user would expect it to be.

1. **Database design and Implementation**

This step is tightly related to the previous one as it is completely etermined by the requirements, analyzed and discussed in step2.

1. **Program’s Structure Analyzing**

The application program as an interface between the users and the database should be an accurate “reflection” of the database on the screen; hence a well analyzed and defined structure is needed.

1. **GUI Constructing**

After analyzing the program’s structure and defining what it should consist of, a graphical representation of this stuff is needed in order to enable the user to interact with the data.

1. **Bringing all the stuff together**

The next step that should be taken is connecting the program with the database and performing the necessary functionality upon all of the controls.

1. **Tests**

To ensure that everything works properly and as it has been expected, test

performance has to be done upon the system’s functionality.

## 2.2 Programming Environments

The system divided into two part Database, and graphical user interface GUI. The SQL used to build database server and C# used to build GUI.

## Program’s Structure Analyzing and GUI Constructing

After getting the database prepared, application program should be constructed and implemented in some programming environment to enable the users to communicate with the database. Graphical User Interface (GUI) is intended to be built up as a basic structure of the program. The first general advice when constructing GUIs is to “know thy users” as

there is a large number of rules and requirements, concerning the whole process of GUI development. Every GUI consists of a certain number of controls (text-boxes, combo boxes, buttons…etc.). The list of all properties and methods for all controls is called Application Programming Interface (API). The Program’s GUI is shown in (Appendix B). A set of controls is used in order to reach the desired purpose, what concerns the functionality of the application, including Labels, Text boxes, Combo Boxes, Data Grid, Buttons, Group Boxes, Panels, Tab controls etc. All of these controls, available in the program, are fitted to the corresponding forms that are used in the application.

## Database Connections and Code-Implementation

This subsection represents the main approach that has been made up in order to establish the connections with the database:

|  |
| --- |
| class DataAcessLayer { SqlConnection sqlconnaction;  public DataAcessLayer() { sqlconnaction = new SqlConnection(@"Server = DESKTOP-UQ4QU19; Database = Email\_trace; Integrated Security= true"); } // open connection public void openconnaction() { if (sqlconnaction.State != ConnectionState.Open) { sqlconnaction.Open(); } } // close connection public void closeconnaction() { if (sqlconnaction.State == ConnectionState.Open) { sqlconnaction.Close(); } } // Select method -- Read public DataTable selectdata(string stored\_procedure, SqlParameter[] parm) { SqlCommand sqlcmd = new SqlCommand(); sqlcmd.CommandType = CommandType.StoredProcedure; sqlcmd.CommandText = stored\_procedure; sqlcmd.Connection = sqlconnaction; if (parm != null) { sqlcmd.Parameters.AddRange(parm); } SqlDataAdapter DA = new SqlDataAdapter(sqlcmd); DataTable DT = new DataTable(); DA.Fill(DT); return DT; } // update operations  public void excutecommand(string stored\_procedure, SqlParameter[] parm) { SqlCommand sqlcmd = new SqlCommand(); sqlcmd.CommandType = CommandType.StoredProcedure; sqlcmd.CommandText = stored\_procedure; sqlcmd.Connection = sqlconnaction; if (parm != null) { sqlcmd.Parameters.AddRange(parm); } sqlcmd.ExecuteNonQuery(); } }} |

Figure 2.1 Preparing to get a connection established

## 2.5 Order to database server

1.
2. 1.
	2.
	3.
	4.
	5.

### Retrieving data from the database

Retrieving data from a database is less or more tightly related to dealing with the SELECT query that should be applied to the database in order to extract the desirable result, which one should satisfy certain conditions. This SQL query has the following structure:

|  |
| --- |
| SELECT <column\_name>FROM <table\_name>WHERE [(condition\_1), (condition\_2), …..(condition\_n)]. |

Into the WHERE-statement, the following logical and arithmetical operators are included as well: [AND, OR, <, <=, >, >=, =].

### Saving Data Into The Database

This kind of operation upon the database is subdivided into two groups: Saving a new employee’s records (Populating all of the tables with data) and Add a record to an employee’s data records. This SQL query has the following structure:

|  |
| --- |
| INSERT INTO [table\_name] ([<column\_name>)] VALUES ([<column\_name>)] |

### Deleting Data From The Database

This kind of operation, performed upon the database, is subdivided into two parts: Single Records Deletion and All Records Deletion. Both parts concern only single employee’s data into the database. Deleting a single record from the database means moving to a certain child table, selecting the record we want to be deleted and press the “Delete a Record” button. The result is instantly reflected into the database and back into the program as well. This SQL query has the following structure:

|  |
| --- |
| DELETE FROM [table\_name] WHERE <Search Conditions,,> |

### Update Data From The Database

SQL query for update date from the database as the following structure

|  |
| --- |
| UPDATE [table\_name] SET  ([<column\_name>) = value  WHERE <Search Conditions> |

CHAPTER 3

# CONCLUSION AND FUTURE WORK

## 3.1 Conclusion and Future Works

In this report, an information system’s development has been presented. The report’s content comprises the whole task solution, starting from the programming environments have been selected, going through the database, the applications analyze and construction, and finishing with the code-implementation and test-samples, shown separately in Appendix chapters.

As future work, some additional stuff could be implemented and integrated into the application code making it much more reliable and flexible; especially what concerns a pay-roll module, for instance.

Apparently, the role of such systems is basic and essential within each company that wants to keep really good control and record concerning its personal data relative to email, functionality and performance on all levels in its structure. Every organization, in nowadays, has the necessity of managing its emails on a really good level as the document has definitely the greatest merit of building up a company as such as it is. The well-managed email means giving the appropriate financial award-ness and all kind of benefits as such as they have been deserved. That’s why the development of such systems is not just a programming business – a lot of people are ordinarily involved in such projects and one of the basic requirements is the reliability of the system, especially what concerns the storage of data and all of the operations that will be performed upon it.

APPENDIX A

# Graphical User Interface (GUI)

Figure 1: Main Interface



Figure 2: Login



Figure 3: Add new Email



Figure 3: Search

A



B



Figure 4: Edit Email



APPENDIX B

# Command of Email processing

using System.IO;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

namespace EmailTrace.DPL

{

 class Email\_process

 {

 public void ADDNEWEMAIL ( string ETF , string ETFDEP, string ETFEM, string ETFDATE, string ETFNO,

 string WHORE , string ETFNOT, byte [] ETFIMAGE)

 {

 DAL.DataAcessLayer DAL = new DAL.DataAcessLayer();

 DAL.openconnaction();

 SqlParameter [] pram = new SqlParameter[8];

 pram[0] = new SqlParameter("@ETF", SqlDbType.NVarChar);

 pram[0].Value = ETF;

 pram[1] = new SqlParameter("@ETFDEP", SqlDbType.NVarChar);

 pram[1].Value = ETFDEP;

 pram[2] = new SqlParameter("@ETFEM", SqlDbType.NVarChar);

 pram[2].Value = ETFEM;

 pram[3] = new SqlParameter("@ETFDATE", SqlDbType.NVarChar);

 pram[3].Value = ETFDATE;

 pram[4] = new SqlParameter("@ETFNO", SqlDbType.NVarChar);

 pram[4].Value = ETFNO;

 pram[5] = new SqlParameter("@WHORE", SqlDbType.NVarChar);

 pram[5].Value = WHORE;

 pram[6] = new SqlParameter("@ETFNOT", SqlDbType.NVarChar);

 pram[6].Value = ETFNOT;

 pram[7] = new SqlParameter("@ETFIMAGE", SqlDbType.Image);

 pram[7].Value = ETFIMAGE;

 DAL.excutecommand("AddNewItemToemailDatabase", pram);

 DAL.closeconnaction();

 }

 //IN

 public void UPDA\_OUT\_EWEMAIL(string ETF, string ETFDEP, string ETFEM, string ETFDATE, string ETFNO,

 string WHORE, string ETFNOT)

 {

 DAL.DataAcessLayer DAL = new DAL.DataAcessLayer();

 DAL.openconnaction();

 SqlParameter[] pram = new SqlParameter[7];

 pram[0] = new SqlParameter("@ETF", SqlDbType.NVarChar);

 pram[0].Value = ETF;

 pram[1] = new SqlParameter("@ETFDEP", SqlDbType.NVarChar);

 pram[1].Value = ETFDEP;

 pram[2] = new SqlParameter("@ETFEM", SqlDbType.NVarChar);

 pram[2].Value = ETFEM;

 pram[3] = new SqlParameter("@ETFDATE", SqlDbType.NVarChar);

 pram[3].Value = ETFDATE;

 pram[4] = new SqlParameter("@ETFNO", SqlDbType.NVarChar);

 pram[4].Value = ETFNO;

 pram[5] = new SqlParameter("@WHORE", SqlDbType.NVarChar);

 pram[5].Value = WHORE;

 pram[6] = new SqlParameter("@ETFNOT", SqlDbType.NVarChar);

 pram[6].Value = ETFNOT;

 DAL.excutecommand("updataemail", pram);

 DAL.closeconnaction();

 }

 public DataTable VerifyEmailNo(string EmailNo)

 {

 DAL.DataAcessLayer DAL = new DAL.DataAcessLayer();

 // in login need two parmeters username and password

 SqlParameter[] pram = new SqlParameter[1];

 pram[0] = new SqlParameter("@EmailNo", SqlDbType.NVarChar, 50);

 pram[0].Value = EmailNo;

 DAL.openconnaction();

 DataTable Dt = new DataTable();

 Dt = DAL.selectdata("VerifyEmailNo", pram);

 DAL.closeconnaction();

 return Dt;

 }

 //

 public DataTable VIN(string EmailNo)

 {

 DAL.DataAcessLayer DAL = new DAL.DataAcessLayer();

 // in login need two parmeters username and password

 SqlParameter[] pram = new SqlParameter[1];

 pram[0] = new SqlParameter("@EmailNo", SqlDbType.NVarChar, 50);

 pram[0].Value = EmailNo;

 DAL.openconnaction();

 DataTable Dt = new DataTable();

 Dt = DAL.selectdata("VEIN", pram);

 DAL.closeconnaction();

 return Dt;

 }

 public DataTable getallt(string PROCE)

 {

 DAL.DataAcessLayer DAL = new DAL.DataAcessLayer();

 DAL.openconnaction();

 DataTable Dt = new DataTable();

 Dt = DAL.selectdata(PROCE, null);

 DAL.closeconnaction();

 return Dt;

 }

 public DataTable SearchEmailT(string proc,string text)

 {

 DAL.DataAcessLayer DAL = new DAL.DataAcessLayer();

 // in login need two parmeters username and password

 SqlParameter[] pram = new SqlParameter[1];

 pram[0] = new SqlParameter("@text", SqlDbType.NVarChar, 50);

 pram[0].Value = text;

 DAL.openconnaction();

 DataTable Dt = new DataTable();

 Dt = DAL.selectdata(proc, pram);

 DAL.closeconnaction();

 return Dt;

 }

 public void deletemail(string ETF,string proc)

 {

 DAL.DataAcessLayer DAL = new DAL.DataAcessLayer();

 DAL.openconnaction();

 SqlParameter[] pram = new SqlParameter[1];

 pram[0] = new SqlParameter("@text", SqlDbType.NVarChar);

 pram[0].Value = ETF;

 DAL.excutecommand(proc, pram);

 DAL.closeconnaction();

 }

 // IN

 public void ADDNEWEMAIL\_IN(string ETF, string ETFDEP, string ETFEM, string ETFDATE, string ETFNO, string WHORE,

 string ETFNOT, string S, byte[] ETFIMAGE)

 {

 DAL.DataAcessLayer DAL = new DAL.DataAcessLayer();

 DAL.openconnaction();

 SqlParameter[] pram = new SqlParameter[9];

 pram[0] = new SqlParameter("@EF", SqlDbType.NVarChar);

 pram[0].Value = ETF;

 pram[1] = new SqlParameter("@EFDEP", SqlDbType.NVarChar);

 pram[1].Value = ETFDEP;

 pram[2] = new SqlParameter("@EFEM", SqlDbType.NVarChar);

 pram[2].Value = ETFEM;

 pram[3] = new SqlParameter("@EFNO", SqlDbType.NVarChar);

 pram[3].Value = ETFDATE;

 pram[4] = new SqlParameter("@ETD", SqlDbType.NVarChar);

 pram[4].Value = ETFNO;

 pram[5] = new SqlParameter("@EFT", SqlDbType.NVarChar);

 pram[5].Value = WHORE;

 pram[6] = new SqlParameter("@ETFNOT", SqlDbType.NVarChar);

 pram[6].Value = ETFNOT;

 pram[7] = new SqlParameter("@EFS", SqlDbType.NVarChar);

 pram[7].Value = S;

 pram[8] = new SqlParameter("@ETFIMAGE", SqlDbType.Image);

 pram[8].Value = ETFIMAGE;

 DAL.excutecommand("ADDNEWIN", pram);

 DAL.closeconnaction();

 }

 //IMAGE

 public DataTable GETIMAGE(string proc, string text)

 {

 DAL.DataAcessLayer DAL = new DAL.DataAcessLayer();

 // in login need two parmeters username and password

 SqlParameter[] pram = new SqlParameter[1];

 pram[0] = new SqlParameter("@ID", SqlDbType.NVarChar, 50);

 pram[0].Value = text;

 DAL.openconnaction();

 DataTable Dt = new DataTable();

 Dt = DAL.selectdata(proc, pram);

 DAL.closeconnaction();

 return Dt;

 }

 }

}

#

REFERENCES

[1] – Begg Carolyn, Connolly Thomas, *Database systems (a Practical approach to*

*Design, Implementation, and Management)*, Addison-Wesley, an imprint of

Pearson Education, University of Paisley (U.K.), Fourth edition 2005

[2] – Bodnar George /Duquesne University/, Hopwood William /Florida Atlantic

University/, *Accounting Information systems*, Eighth Edition, Prentice Hall,

Upper Saddle River, New Jersey **.**

[3] – Andersson Tobias, *[DAB744] C# Course Lectures*, School of Mathematics and

System Engineering, Växjö University.

[4] - http://msdn.microsoft.com/library/default.asp?url=/library/en**us/**

**vbcon/html/vboritextboxctltasks.asp** (2006-05-25).