# STUDENT PORTAL FOR AL KHAWARIZMI INTERNATIONAL COLLEGE, ABU DHABI, UAE

Noura Al Neyadi

ID: k-20961023

Semester: Fall 2016/2017

Type: Temporary Trainee

# Offers Management System for Federal Authority of Nuclear Regulation, ABU DHABI, UAE

By

Noura Al Neyadi

ID: k-20961023

A Research Project Presented in Partial Fulfillment
of the Requirements for the Bachelor Degree in Information Technology
AL KHAWARIZMI INTERNATIONAL COLLEGE
Semester: Fall 2016/2017

Academic Supervisor: Dr. Muath Alrammal

# Acknowledgment

First and foremost, I'd like to thank God, the Merciful, and the Mercy Giving for giving me the strength to trust in myself and follow my ambitions, and for making everything possible.

It is with great pleasure that I state my genuine and deep sense of gratefulness to my supervisors, Dr. Muath Alrammal, for his continuous inspiration and excellent support during the course of my internship.

Special thanks and appreciation to all staff members of Al Khawarizmi International College for providing me with valuable information and for making my time spent at KIC a memorable one.

I'm substantially thankful to my husband for his wholehearted love, support, patience & motivation.

Last but not the least, many thanks for my classmates, my friends and all of people I have met in the college for their contribution in my success and completing my bachelor degree in IT.

# Abstract

The Federal Authority of Nuclear Regulation (FANR) provides the facility of different discounts offers related to restaurants, travel, hotel, spa, gym leisure and many more. General Services Department is responsible for originating these offers and then publish them to the FANR employees through the email. GS department has problems in managing the increasing numbers of offers, sending offers through email make it difficult for the employees to search and find the required offer and most of the employees delete the email or archive the email and they cannot categorize the default email box, this was the main motivation to propose a solution for the General Services Department. A desktop based application Offer Management system is the better way to communicate with the employees. The General services department can add the new offers within the offer management system, Employees have the access to the system from where they can find the required offers, add offers to their favorite section and can request a ticket according to the offer. Offer Management System is more efficient than using email.

# LIST OF TABLES

# LIST OF FIGURES

## CHAPTER ONE

## 1.0 INTRODUCTION OF ORGANIZATIONAL RESEARCH

### 1.1 Organization Description

FANR was established to be the regulatory body for the nuclear sector in the UAE in accordance with Federal Law by Decree No 6 of 2009, Concerning the Peaceful Uses of Nuclear Energy, which was issued by the UAE President H.H. Sheikh Khalifa bin Zayed Al Nahyan.

FANR protects the UAE's public, its workers and the environment by conducting nuclear regulatory programs in safety, security, radiation protection and safeguards, which fulfill key objectives in licensing and inspection in accordance with best international practices. FANR also oversees the implementation of the UAE's obligations under the international treaties, conventions and agreements in the nuclear sector, and determines administrative standards, which support excellence in regulation.

FANR has achieved remarkable success in the UAE's peaceful nuclear programs through transparency in its operations and a dedication to sustainability through the capacity-building of Emiratis in the nuclear sector at FANR. Furthermore, FANR has also gained international recognition as a competent regulatory body and for its close cooperation with the International Atomic Energy Agency (IAEA).

### 1.1.1 Strategy

**Vision:**

To ensure long-term safety, security and sustainability in the peaceful uses of nuclear energy and ionizing radiation in the UAE by establishing world-class regulations and supervising their implementation.

**Mission:**

To protect the public, workers and the environment by conducting nuclear regulatory programs in safety, security, radiation protection and safeguards, which fulfil key objectives in licensing and inspection in accordance with best international practices, capacity-building to ensure sustainability, overseeing the UAE obligations under the international treaties, conventions and agreements in the nuclear sector, and administrative standards, which support excellence in regulation.

**Objectives:**

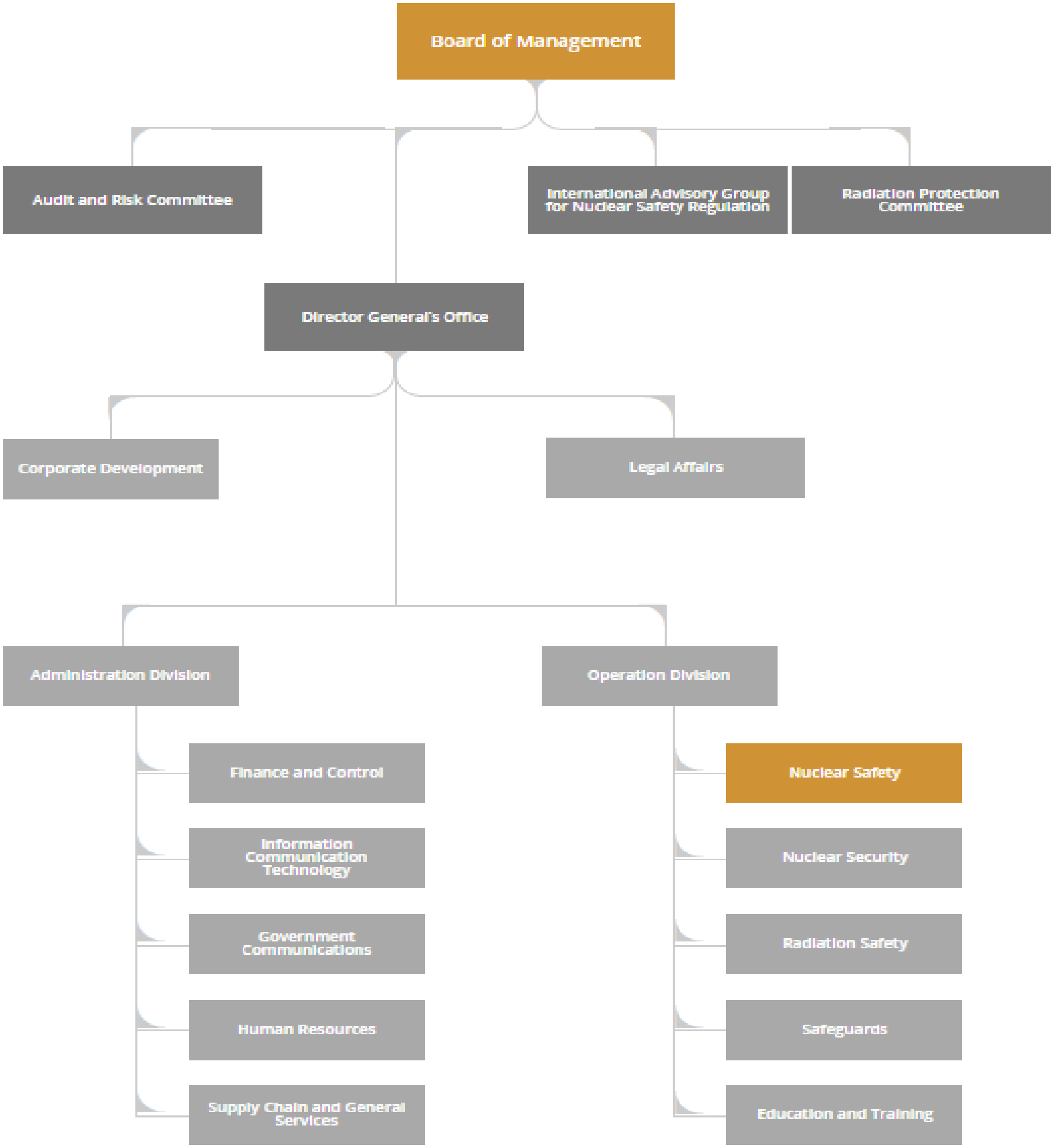
- Safety Awareness and Responsibility
- Independence
- Transparency
- Competency

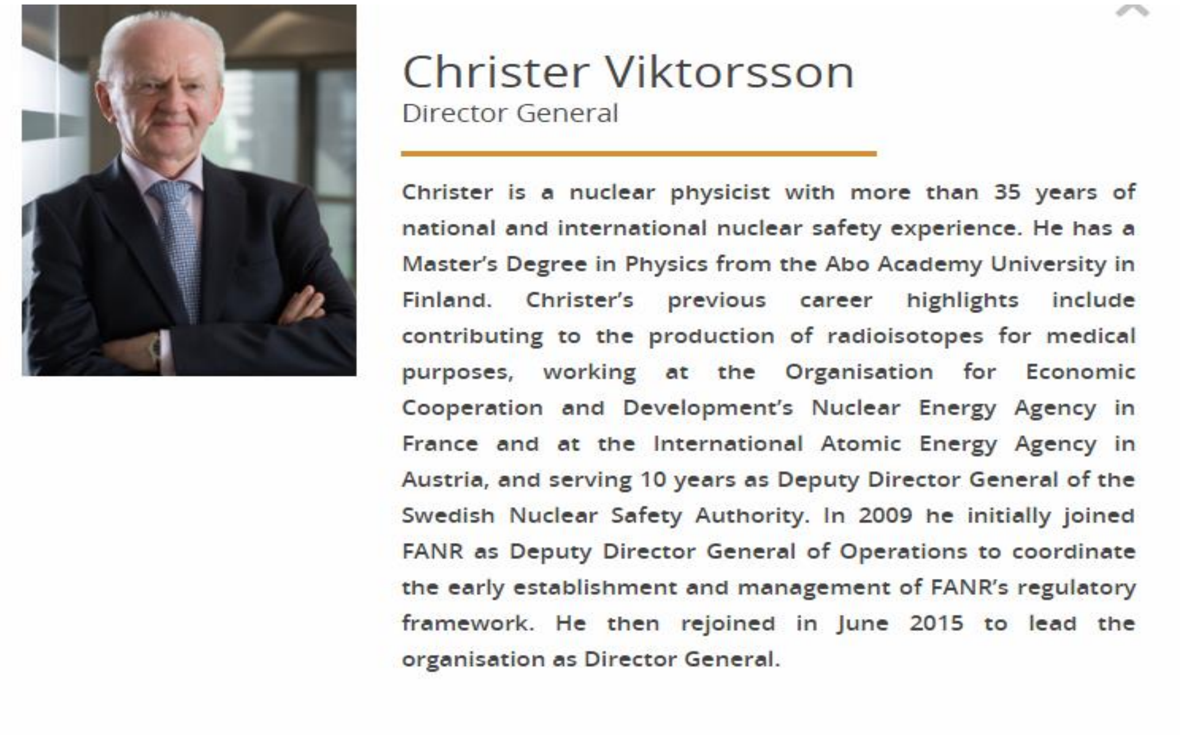
**Top management's/leadership's role:**

- The Board of Management is FANR's decision-making body and is appointed by a resolution of the UAE Cabinet.
- The Board of Management appoints the Director General and is responsible for making key decisions at FANR.
- The Board of Management consists of the following individuals:
- H.E. Abdulla Nasser Al Suwaidi - Chairman
- H.E. Ambassador Hamad Ali Al Kaabi - Deputy Chairman
- H.E. Mr. Ali Khalfan Al Dhaheri - Board Member
- H.E. Mrs. Razan Khalifa Al Mubarak - Board Member
- H.E. Dr. Abdul Qader Ebrahim Alkhayat - Board Member
- H.E. Dr. Ali Mohamed Shaheen Ahmed - Board Member
- H.E. Mr. Sultan Abdullah Sultan Al Hebsi - Board Member

**Organizational Chart- staff names & positions:**

**Organization Chart ( Figure 1 )**





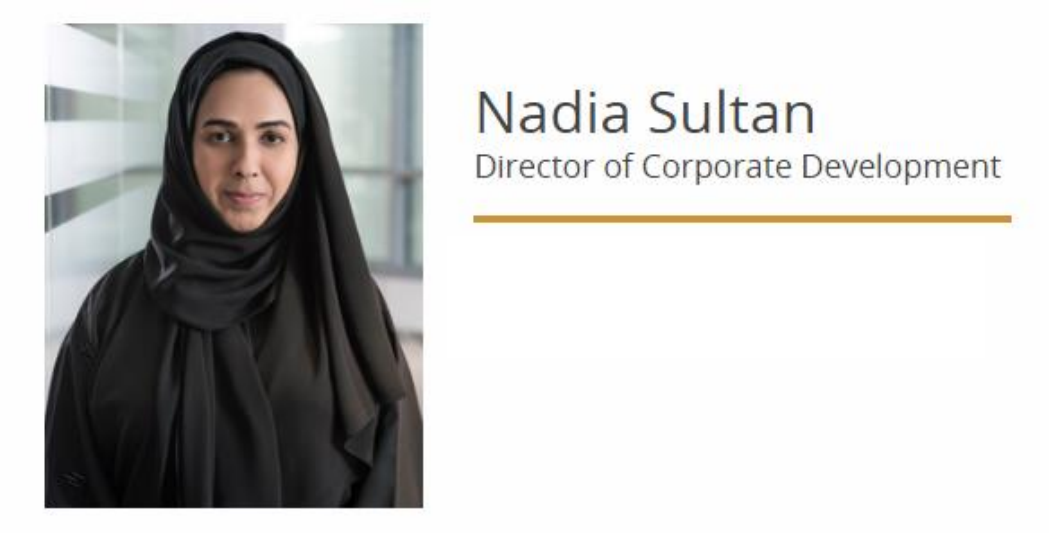
## Christer Viktorsson

Director General

Christer is a nuclear physicist with more than 35 years of national and international nuclear safety experience. He has a Master's Degree in Physics from the Abo Academy University in Finland. Christer's previous career highlights include contributing to the production of radioisotopes for medical purposes, working at the Organisation for Economic Cooperation and Development's Nuclear Energy Agency in France and at the International Atomic Energy Agency in Austria, and serving 10 years as Deputy Director General of the Swedish Nuclear Safety Authority. In 2009 he initially joined FANR as Deputy Director General of Operations to coordinate the early establishment and management of FANR's regulatory framework. He then rejoined in June 2015 to lead the organisation as Director General.

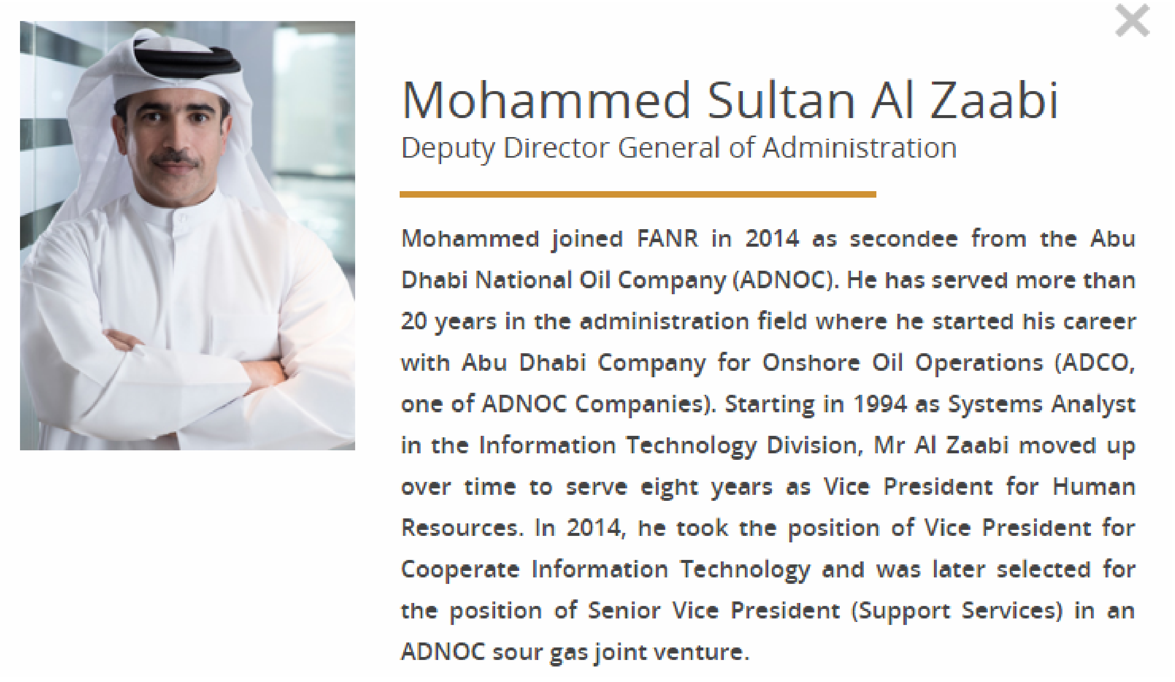
## Zoryana Vovchok

Acting Director, Legal Affairs



## Nadia Sultan

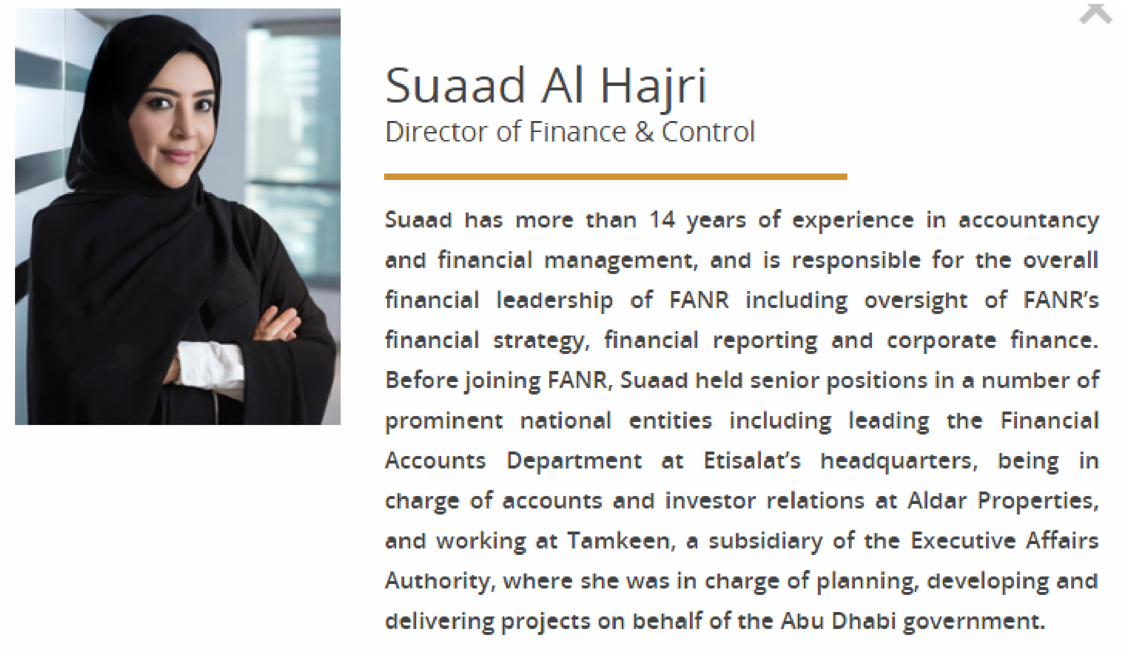
Director of Corporate Development



## Mohammed Sultan Al Zaabi

Deputy Director General of Administration

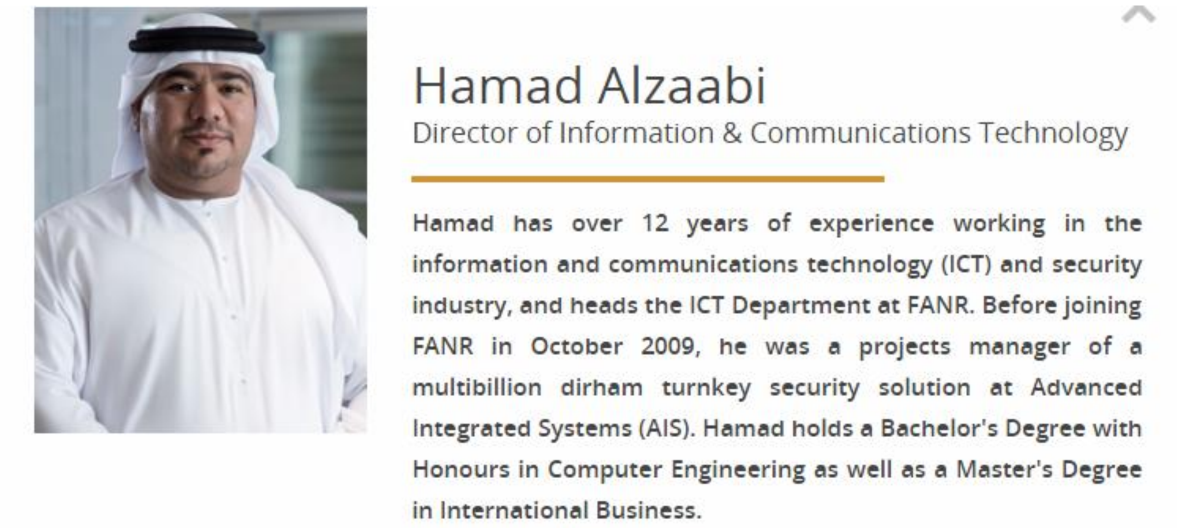
Mohammed joined FANR in 2014 as secondee from the Abu Dhabi National Oil Company (ADNOC). He has served more than 20 years in the administration field where he started his career with Abu Dhabi Company for Onshore Oil Operations (ADCO, one of ADNOC Companies). Starting in 1994 as Systems Analyst in the Information Technology Division, Mr Al Zaabi moved up over time to serve eight years as Vice President for Human Resources. In 2014, he took the position of Vice President for Cooperate Information Technology and was later selected for the position of Senior Vice President (Support Services) in an ADNOC sour gas joint venture.



## Suaad Al Hajri

Director of Finance & Control

Suaad has more than 14 years of experience in accountancy and financial management, and is responsible for the overall financial leadership of FANR including oversight of FANR's financial strategy, financial reporting and corporate finance. Before joining FANR, Suaad held senior positions in a number of prominent national entities including leading the Financial Accounts Department at Etisalat's headquarters, being in charge of accounts and investor relations at Aldar Properties, and working at Tamkeen, a subsidiary of the Executive Affairs Authority, where she was in charge of planning, developing and delivering projects on behalf of the Abu Dhabi government.



## Hamad Alzaabi

Director of Information & Communications Technology

Hamad has over 12 years of experience working in the information and communications technology (ICT) and security industry, and heads the ICT Department at FANR. Before joining FANR in October 2009, he was a projects manager of a multibillion dirham turnkey security solution at Advanced Integrated Systems (AIS). Hamad holds a Bachelor's Degree with Honours in Computer Engineering as well as a Master's Degree in International Business.

## Salah Hashimi

Director of Government Communications

Salah joined FANR in 2015 as the Director of the Government Communications Department and has more than 12 years' marketing and communications experience. His previous roles were that of Senior Corporate Manager for Majid Al Futtaim Holding, Assistant Vice President as well as Director of Marketing and Corporate Communications for the Community Development Authority (CDA) and Head of Corporate Communications for Dubai Islamic Bank (DIB). Other career highlights include developing and launching websites for the CDA, the DIB and Dubai Sports City, launching and marketing the Manchester United Soccer Schools brand in the UAE, and being selected by the UAE's Minister of State for Happiness to serve as one of the nation's 60 Chief Happiness and Positivity Officers.

## Saja Awad

Director of Human Resources

Saja is responsible for overseeing the Human Resources Department at FANR. She has a Master's Degree in Executive Business and comes with 13 years of experience working predominantly in the administrative and technical fields. Saja started her career with the UAE Army in the Administration and IT Department followed by the role she undertook as Communications and Document Management Officer at Dolphin Energy before finally moving to FANR in 2008 to work in the Human Resources Department. Saja was one of the first to have worked as a researcher with Mubadala on the Abu Dhabi Awards.

FANR
الهيئة الاتحادية للرقابة النووية
Federal Authority for Nuclear Regulation

## Ali Al Zaabi

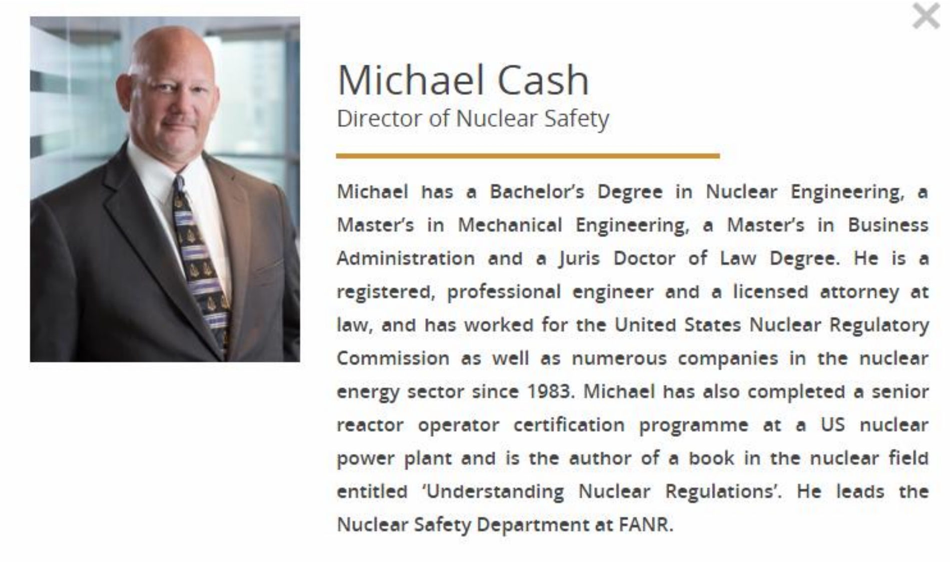
Director of Supply Chain & General Services

Ali holds a Master's Degree in Business Administration (MBA) as well as a Bachelor's Degree in Business and Management, Financial Services and Banking. He comes with over 15 years of experience in administration of which five years were specifically spent gaining in-depth knowledge in procurement. Ali has worked in diverse sectors such as energy, marine and business where he developed his managerial skills and applied his academic background before finally joining FANR and continuing to excel in his area of expertise.

## Ian Grant

Deputy Director General of Operations Division

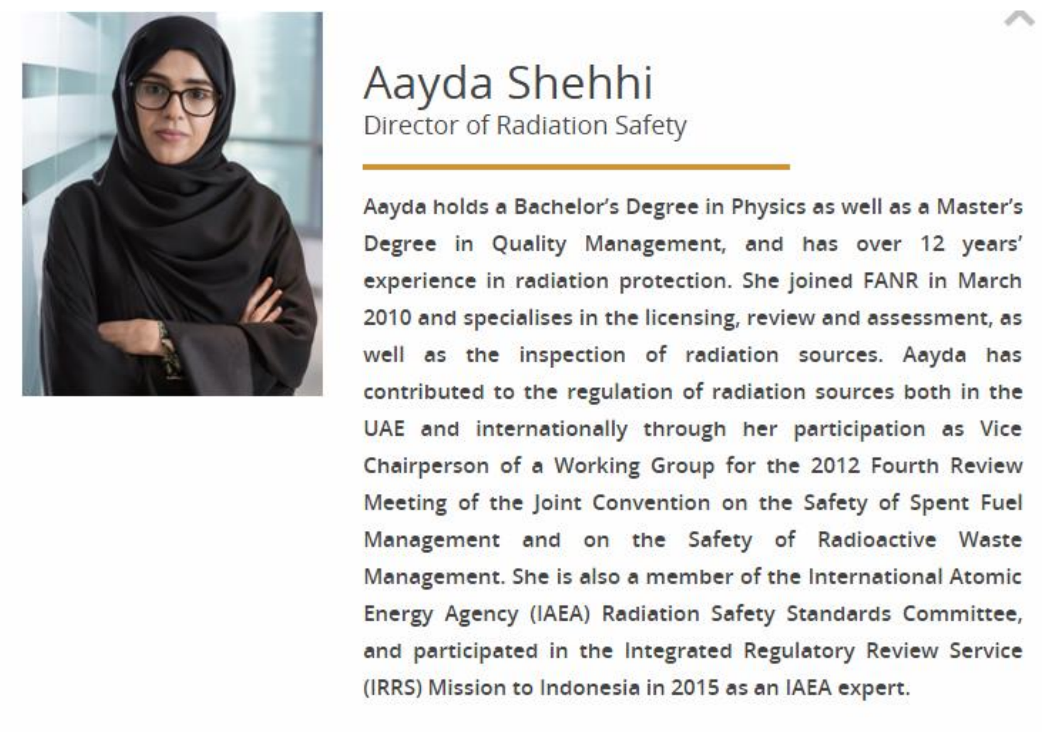
Ian heads the Operations Division having previously held the position of Director of Nuclear Safety at FANR. His career before joining FANR included various senior management roles at the Canadian Nuclear Safety Commission as well as senior management positions at Atomic Energy of Canada Limited and Babcock and Wilcox in the United Kingdom. Ian was also invited to contribute to several regulatory peer review missions organised by the International Atomic Energy Agency (IAEA), to participate in the Convention on Nuclear Safety, to be a national delegate on the Organisation for Economic Co-operation and Development's Nuclear Energy Agency committees for nuclear safety and regulation, and to serve as a UAE delegate on the IAEA Nuclear Safety Standards Committee.



## Michael Cash

Director of Nuclear Safety

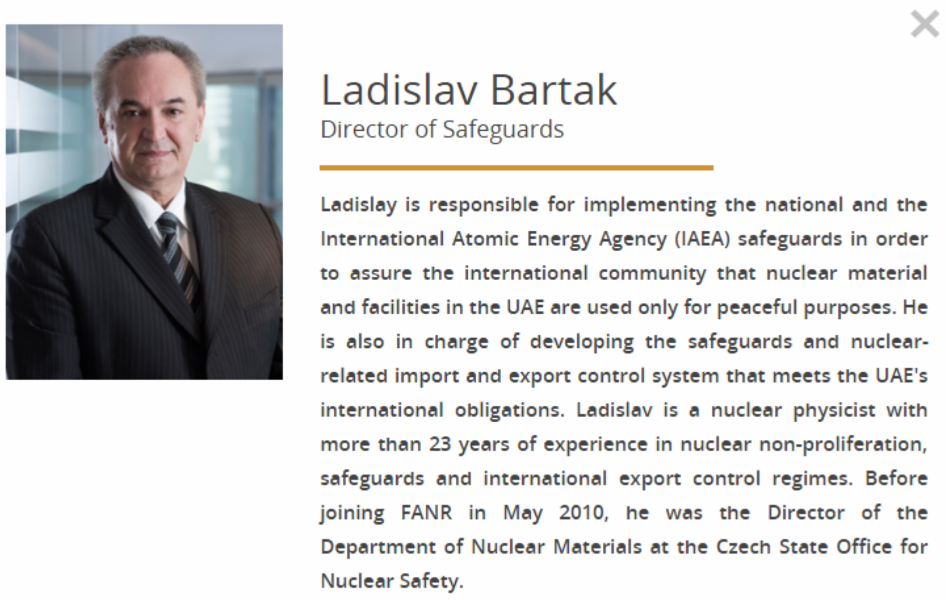
Michael has a Bachelor's Degree in Nuclear Engineering, a Master's in Mechanical Engineering, a Master's in Business Administration and a Juris Doctor of Law Degree. He is a registered, professional engineer and a licensed attorney at law, and has worked for the United States Nuclear Regulatory Commission as well as numerous companies in the nuclear energy sector since 1983. Michael has also completed a senior reactor operator certification programme at a US nuclear power plant and is the author of a book in the nuclear field entitled 'Understanding Nuclear Regulations'. He leads the Nuclear Safety Department at FANR.



## Aayda Shehhi

Director of Radiation Safety

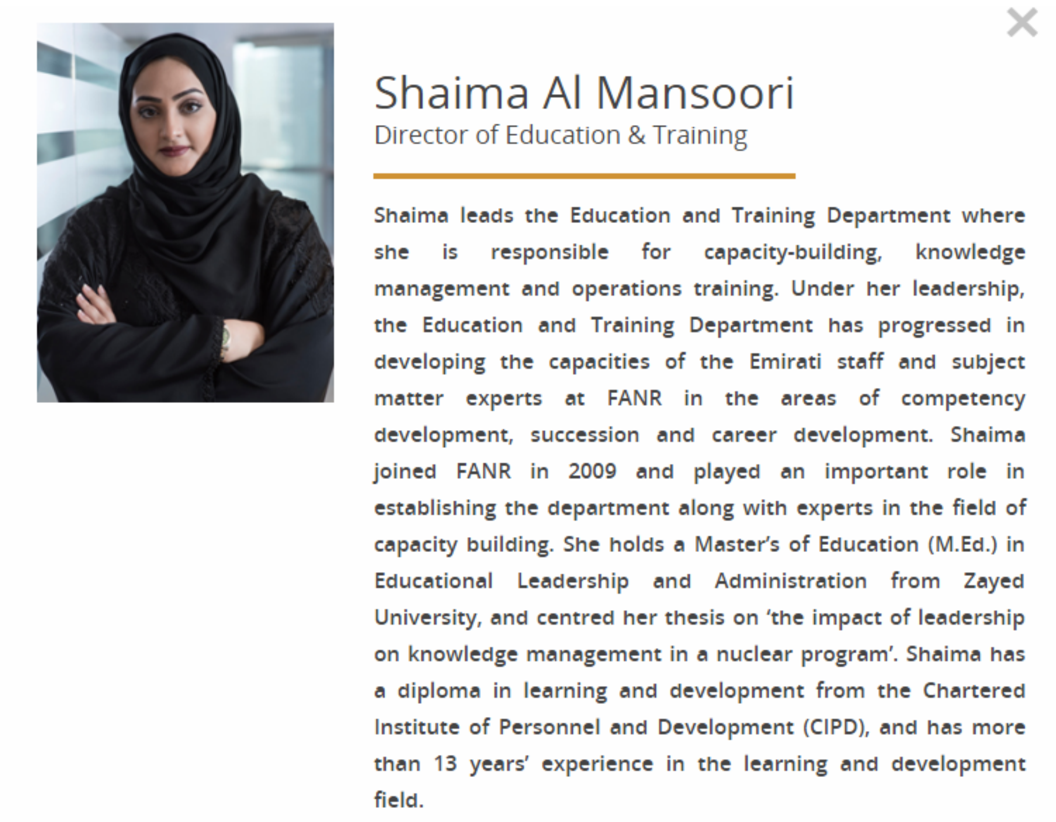
Aayda holds a Bachelor's Degree in Physics as well as a Master's Degree in Quality Management, and has over 12 years' experience in radiation protection. She joined FANR in March 2010 and specialises in the licensing, review and assessment, as well as the inspection of radiation sources. Aayda has contributed to the regulation of radiation sources both in the UAE and internationally through her participation as Vice Chairperson of a Working Group for the 2012 Fourth Review Meeting of the Joint Convention on the Safety of Spent Fuel Management and on the Safety of Radioactive Waste Management. She is also a member of the International Atomic Energy Agency (IAEA) Radiation Safety Standards Committee, and participated in the Integrated Regulatory Review Service (IRRS) Mission to Indonesia in 2015 as an IAEA expert.



## Ladislav Bartak

Director of Safeguards

Ladislay is responsible for implementing the national and the International Atomic Energy Agency (IAEA) safeguards in order to assure the international community that nuclear material and facilities in the UAE are used only for peaceful purposes. He is also in charge of developing the safeguards and nuclear-related import and export control system that meets the UAE's international obligations. Ladislav is a nuclear physicist with more than 23 years of experience in nuclear non-proliferation, safeguards and international export control regimes. Before joining FANR in May 2010, he was the Director of the Department of Nuclear Materials at the Czech State Office for Nuclear Safety.



## Shaima Al Mansoori

Director of Education & Training

Shaima leads the Education and Training Department where she is responsible for capacity-building, knowledge management and operations training. Under her leadership, the Education and Training Department has progressed in developing the capacities of the Emirati staff and subject matter experts at FANR in the areas of competency development, succession and career development. Shaima joined FANR in 2009 and played an important role in establishing the department along with experts in the field of capacity building. She holds a Master's of Education (M.Ed.) in Educational Leadership and Administration from Zayed University, and centred her thesis on 'the impact of leadership on knowledge management in a nuclear program'. Shaima has a diploma in learning and development from the Chartered Institute of Personnel and Development (CIPD), and has more than 13 years' experience in the learning and development field.

### 1.1.2 Organization Activities

The Authority had automated all its services online so it will be easy for the customers to access its services, where the user can login in the authority website to apply for FANR e-services .Also there is section in the website called FAQs that answer most of the questions that probably asked by customers .

Some of the Authority activities:

- Provide Licence to Conduct a Regulated Activity using Regulated Materials
- Renewal a Licence to Conduct Regulated Activity using Regulated Materials
- Regulated Material Inventory List
- Suppliers Registration
- Occupational Dose Records
- Notification of Transport of Regulated Materials
- Import/ Export Permit for Regulated Materials (Radiation sources).

### 1.1.3 Nature of the business

FANR's Mission stems from the UAE's decision to pursue a nuclear energy programme whilst adhering to the highest international standards. FANR has established regulations and carries out inspections to protect the nation's people and environment.

In order to implement FANR's regulatory framework, FANR licenses the construction and operation of nuclear power plants and all uses of radioactive material and radiation sources such as those used in the medical sector.

FANR's success in establishing itself as an effective nuclear regulator can be attributed to transparency in its operations, dedication to building the capacities of young Emiratis in the nuclear sector, and assuring the world of FANR's peaceful nuclear activities by meeting international nuclear safeguards obligations.

## 1.2 Investigation and Problem/Project

General Services Department provides employees with Offers and special discount from different hotels, spa, ..etc.

But once they got new offer they send email to all FANR employees.

First for the employee it's difficult to save all these email and it make their email box always with the message :" your email almost full " because of the size of the emails sent by THE GS department including all the pictures and details of the offers so most of the employee delete them or archive them and forgot about them.

Second, it's difficult for the employee to search among that large number of email to find a specific offer he needs.

The email send randomly there is no categories or specific criteria the employee can use to search through all these offers he completely depends on what outlook offers to search which is difficult among other emails he gets.

Thirds, the emails it's not attractive, they use formal template to send the offer.

I conducted interview with the GS department Director to propose my idea about developing a system called "Offers Management System" to avoid the manual and inefficient way to present the offers.

## 1.3 Statement of the Problem/Project

Based on the results of observation & interview, I have come to the conclusion that using Email as an E-communication tool to inform FANR employee about new offers is not very effective. When the employee misses out on useful discounts, it leads to confusion and complaints. This can cause dissatisfaction of employees, and could effects GS department's reputation.

## 1.4 Purpose/Objective of the Research Study

The objectives of the project are the following: -

- To investigate the problem in the company and collect requirements.
- To design reliable offers Management System.
- To develop a prototype for the expected system.

### 1.4.1 Contribution

This study is being done to improve GS department E-communication by developing Offers Management Systems which will offer the following services:

- The Employee can search the offers automatically by using keywords for what he is looking.
- The admin can easily search, add, edit, and delete new offers.

- Employee can register and login to add offers to his favorite and request a booking for ticket from GS department.

Without using the system, employees would have difficulties in finding offers by using outlook searching and his email box will always be full due to the size and amount of offers send by GS department. The Offers Management system will simplify the process of searching offers and categorize them. So the employee  will get offers without worrying about how to save them or where to find them and which offer related to the city he want to use the offer in its hotels , restaurants ….etc. .

## 1.5 Project Plan and Approach Study

The project will begin through observation and interview will be conducted to generate the problem statement. A research study will be done to find related issues in the area and a suitable solution will be chosen.

The methodology that will be used during the course of the project is the Waterfall Model which is the earliest SDLC approach. Requirements (functional and non-functional) for the system will be gathered by conducting interview with GS department‘s employees. Based on these requirements, the system will be designed by creating ER diagram, use case diagram, and sequence diagrams.

After finalizing the design, implementation of the system will take place. The system will be developed in Visual Studio 2015 using C# language, and the database using MS Access. All needed documentation will be presented in this report.

A Gantt chart has been developed using Microsoft Project to indicate the project timelines. (Refer to Appendix A for the Gantt chart).

**Chapter Two:**

# 2.0 BACKGROUND

## 2.1 Literature Reviews

General Services (GS) department is one of the most active department in FANR. It has direct connections with all FANR employees by offering them different types of services .one of these services is searching for commercial companies such as hotels  travel agents …etc. to conduct agreement with them in order to provide FANR employees with special offers .

Automating these services become necessary for the organization to be in line with other government organizations. GS department had automated many processes such as catering process, Meeting room reservation process and other main process in the department.

But the offers system has been not automated yet, since GS is always looking for new ways to present their services to the employee, automating the offers system will help the employee in being more motivated to their work through:

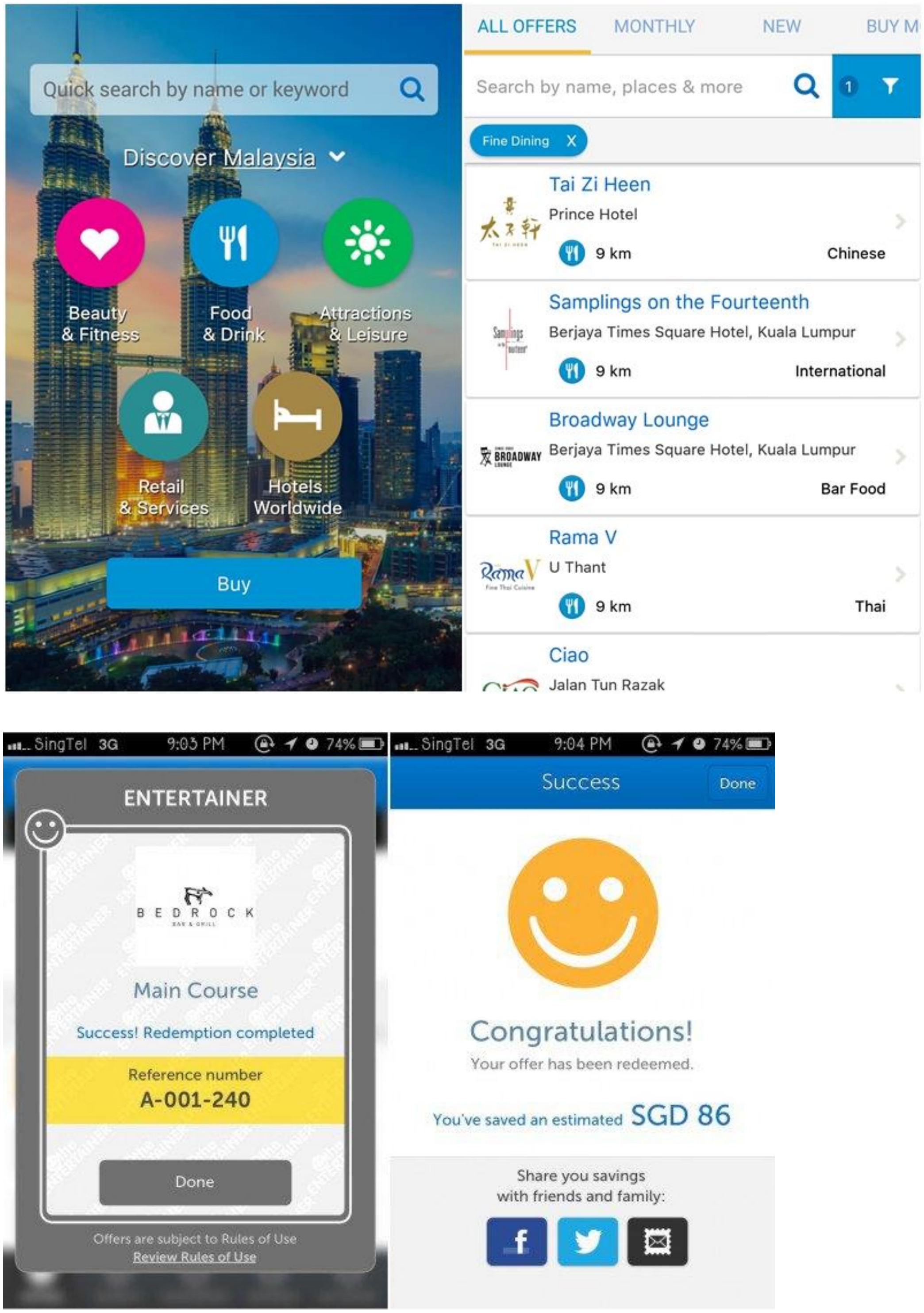
- Getting special discounts from most well-known brands.
- Easy to find and search the offers according on what they are looking for.
- Adding preferred offers in their favorites.

Having an effective system can benefit the company in several ways; paper-based work will be eliminated or significantly reduced, time taken to perform the routine tasks will be reduced, too, as a result productivity will be increased, and decisions taken will be more accurate as data will be analyzed automatically and reports will be generated automatically as well.

## 2.2 Review of similar system

### -The ENTERTAINER

### Snapshot of Entertainer application (Figure 2)



Tai Zi Heen
Prince Hotel
9 km Chinese

Samplings on the Fourteenth
Berjaya Times Square Hotel, Kuala Lumpur
9 km International

Broadway Lounge
Berjaya Times Square Hotel, Kuala Lumpur
9 km Bar Food

Rama V
U Thant
9 km Thai

Ciao
Jalan Tun Razak

Main Course

Success! Redemption completed

Reference number
A-001-240

Offers are subject to Rules of Use

Congratulations!
Your offer has been redeemed.

You've saved an estimated SGD 86

Share you savings
with friends and family:

## Description

The ENTERTAINER brings you unbeatable value with thousands of Buy One Get One Free offers for dining, attractions, leisure activities, health & fitness and hotel accommodation across 40 destinations worldwide. Discover new restaurants, soothe your senses at the best spa in town, treat the kids to a fun activity, and check out an attraction and even holiday more often for less – all with the ENTERTAINER. Our offers are valid all year round, 7 days a week, with virtually no restrictions as to how and when you use them.

This app is specifically designed and architected for iOS8 - please upgrade your device for the best experience.

With the ENTERTAINER you can:
• Browse all 2-for-1 offers across our destinations.
• Search offers and merchants by location and category.
• Redeem all your offers straight from your iPhone.
• View how much you've saved.
• Share your offers with family and friends.
• Try an offer before you buy.

- Offers are available for many destinations such as Abu Dhabi, Athens, Bahrain, Cape Town and much more, Plus 300+ hotels and resorts worldwide.

# Chapter Three: RESEARCH METHODOLOGY

## 3.1 Data Collection Techniques

- **Interview**

  Interviews are conducted with the General services Manager and hotel / travel officer to analysis the current used system and its weaknesses and to know the exact requirements they need for the new system as they are the end users.

- **Literature Review:**

  Literature review was done in order to gain knowledge about what other people are saying concerning areas related to the project topic. It was essential to review similar systems to see what those systems are offering, their advantages & drawbacks, and what features could be added to the new system to make it a better one.

## 3.2 SDLC - Waterfall Model

The methodology that will be used during the course of the project is the Waterfall Model which is the earliest SDLC approach. This model is very easy to understand & use.  In this approach, each phase is processed & completed one at a time. At the end of every phase, a review is done to verify if the project is on the correct path or not, and if it should be continued or discarded. This helps in easy management of the project. (Anonymous, 2016) The phases in Waterfall Model are Requirement, Design, Implementation, Verification, and Maintenance (Figure 3).

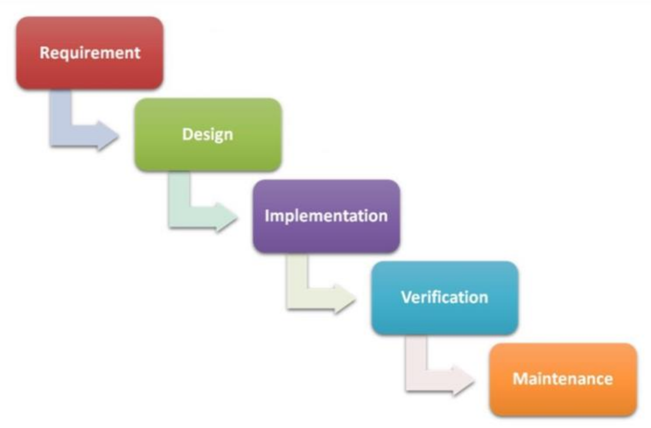


**Figure 3: Phases of Waterfall Model**

### 3.2.1 Requirement

In this phase we will gather all possible requirements of the system from the end users. These requirements will then be analyzed so see whether they're valid and can be incorporated into the system or not. Lastly, a requirements specification document will be generated to serve as an input for the next step of the model. (The Waterfall Model Explained, 2016)

### 3.2.2 Design

In this phase the requirements from the previous phase will be reviewed, and a system design will be made. This design will help in identifying system requirements and describing the complete architecture of the system. The system design specifications will act as the input for the next step. (The Waterfall Model Explained, 2016)

### 3.2.3 Implementation

In this phase the design specifications from the previous phase will be used to construct the system by coding. At the end of this phase, we will come out with one or more system components which will be integrated to meet the requirements of the system architecture. (Melonfire, 2006)

### 3.2.4 Verification

In this phase the system will undergo testing & debugging to make sure that it is free of errors and completely satisfies the requirements defined in the first phase. The system documentation will also be created, revised & published in this phase. (Melonfire, 2006)

# Chapter Four: ANALYSIS AND DESIGN

## 4.1 System Analysis

Based on the interviews conducted with related employees, the functional & non-functional requirements of the Offers Management System are as follows.

### 4.1.1 Functional Requirements

- **Admin:**
  - **Login** – Admin can login by providing username and password.
  - **Post offers**– Admin can post latest offers for FANR.
  - **View offers**– Admin can view offers.
  - **Add offers** – Admin can add new offers.
  - **Delete offers** – Admin can delete offers.
  - **Logout** – Admin can logout.

- **User:**
  - **Register** – User can register by providing name, username, password, department & email.
  - **Login** – User can login by using username & password.
  - **View offers**– User can view offers posted by Admin.
  - **Add offers to favorite** – User can add the offers he liked to his favorite so that later he can get back to it.
  - **Reserve tickets** – User can request for ticket booking from booking officer through filling a form and then the details will be sent to the officer.
  - **Logout** – User can logout.

### 4.1.2 Non-Functional Requirements

- **Usability:**

  The system’s structure should be simple, well-ordered and perceptive. Its functions should be easy to use and efficient.

- **Availability:**

  The system should be available all the time so that users can use it any time they need. In the event of a system crash, the system should still be accessible due to backup server.

- **Security:**

  The system should be safe from hackers by using authentication & authorization. Regularly backing up data will help in preventing losing data.

- **Recoverability:**

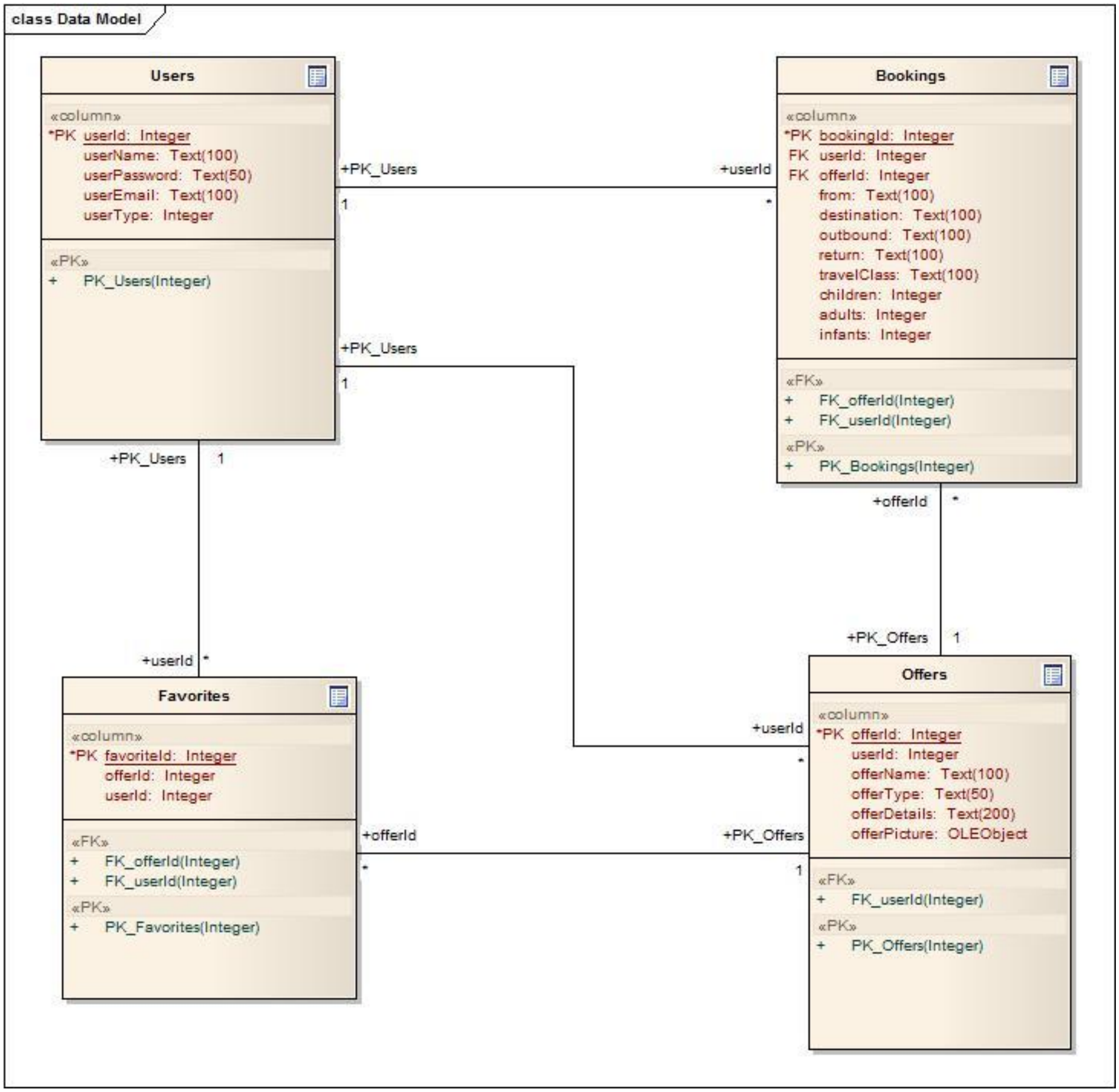
  If the system happens to crash, it should be able to recover by less than 1 hour. This can be made possible by making backup of databases, and giving training lessons and documentation.

- **Maintainability:**

  The system should be updated once a month . It should be flexible so that features can be added or modified according to changing requirements. When errors occur, they should be fixed.

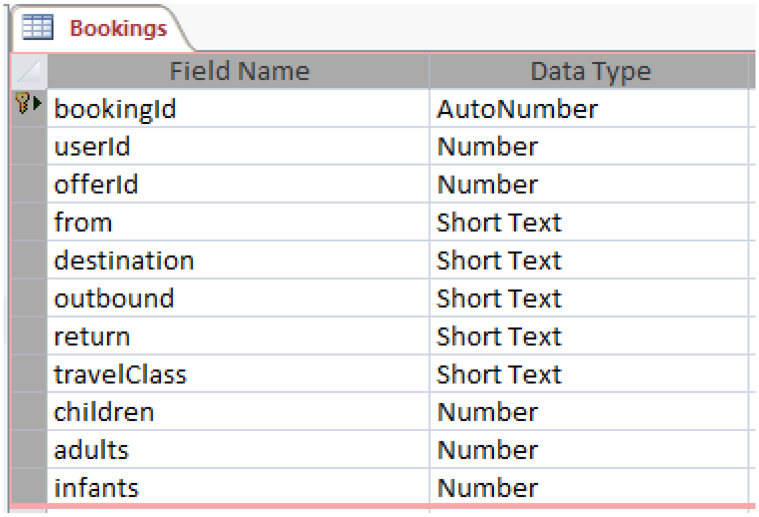
## 4.2 System Design

### 4.2.1 ER Diagram (figure 4)



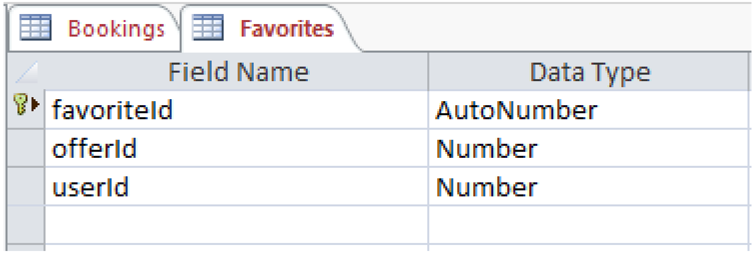
### 4.2.2 Data Dictionary

Booking Table (table 1)



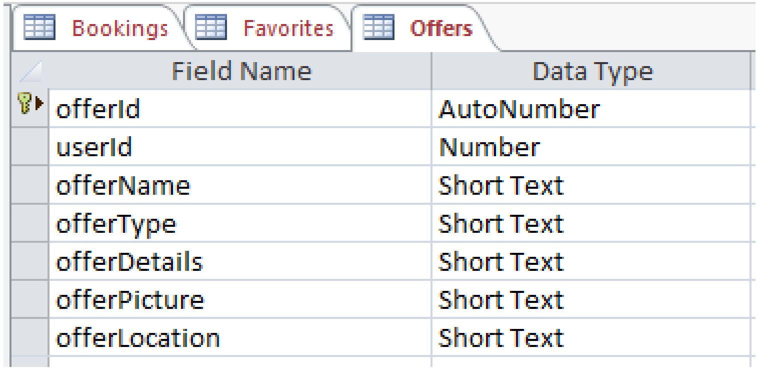
| Field Name | Data Type |
|---|---|
| bookingId | AutoNumber |
| userId | Number |
| offerId | Number |
| from | Short Text |
| destination | Short Text |
| outbound | Short Text |
| return | Short Text |
| travelClass | Short Text |
| children | Number |
| adults | Number |
| infants | Number |

Favorites Table (table 2)



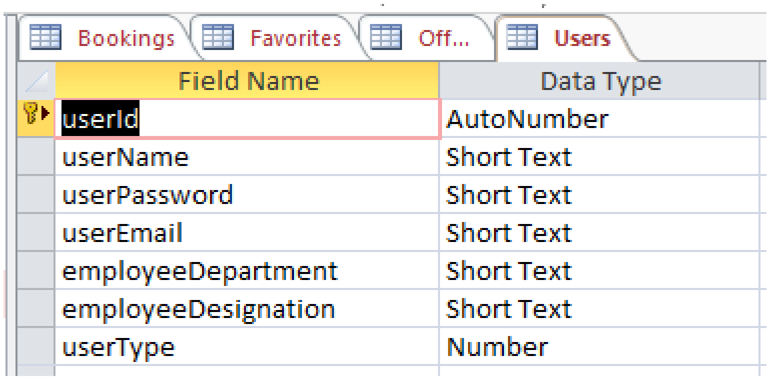
| Field Name | Data Type |
|---|---|
| favoriteId | AutoNumber |
| offerId | Number |
| userId | Number |

Offers Table (table 3)



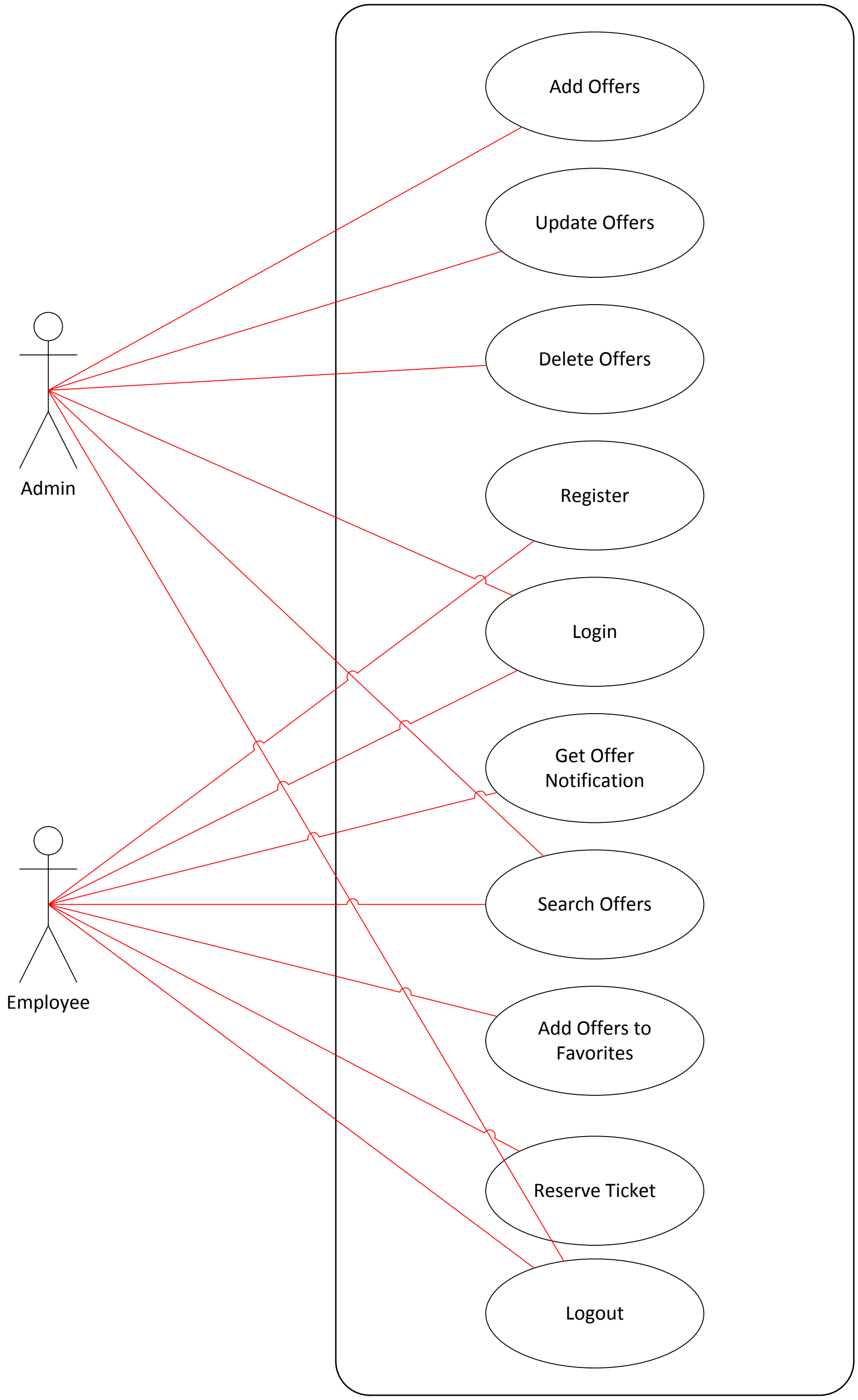
| Field Name | Data Type |
|---|---|
| offerId | AutoNumber |
| userId | Number |
| offerName | Short Text |
| offerType | Short Text |
| offerDetails | Short Text |
| offerPicture | Short Text |
| offerLocation | Short Text |

Users Table (table 4)



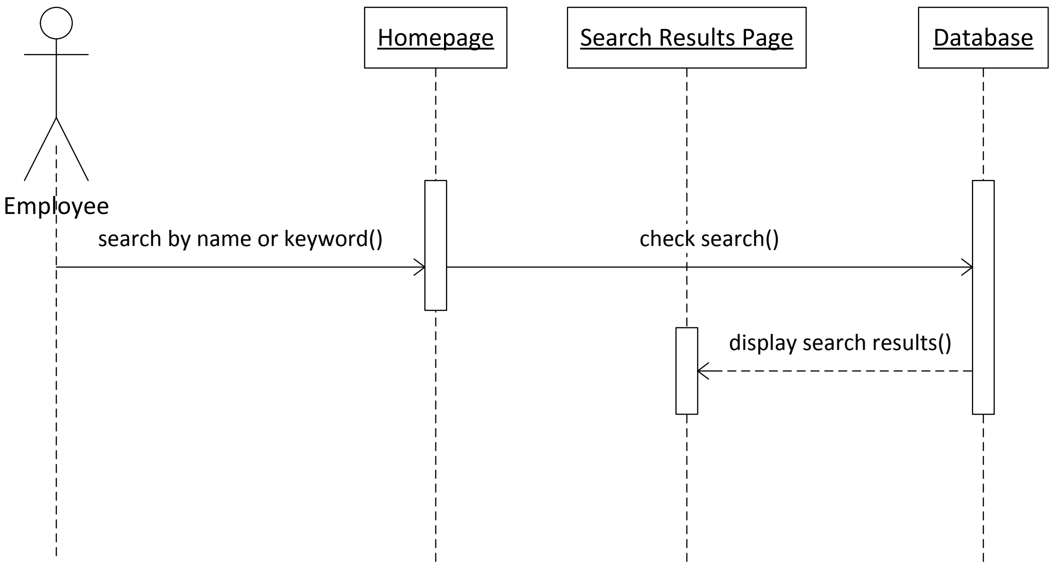
| Field Name | Data Type |
|---|---|
| userId | AutoNumber |
| userName | Short Text |
| userPassword | Short Text |
| userEmail | Short Text |
| employeeDepartment | Short Text |
| employeeDesignation | Short Text |
| userType | Number |

### 4.2.3 Use case diagram (figure 5)

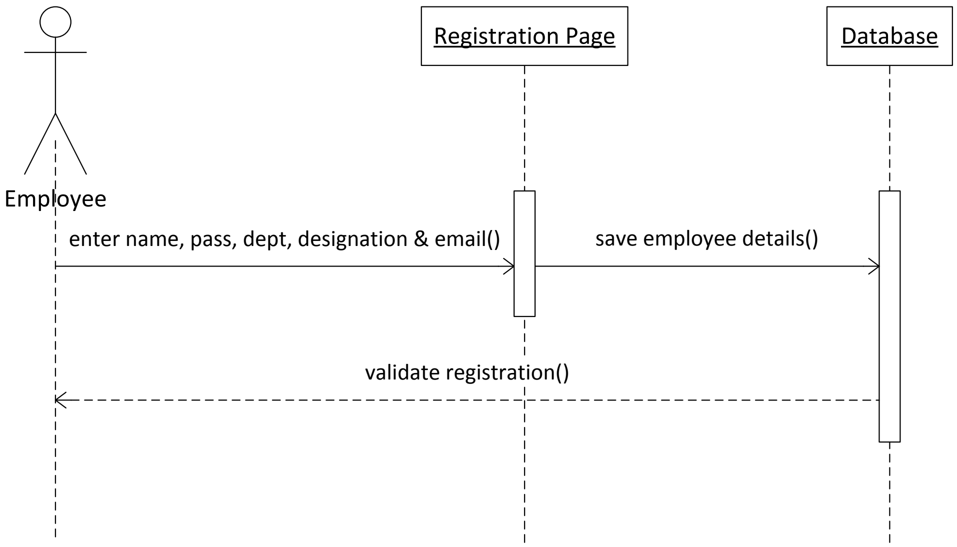


### 4.2.4 Sequence Diagrams

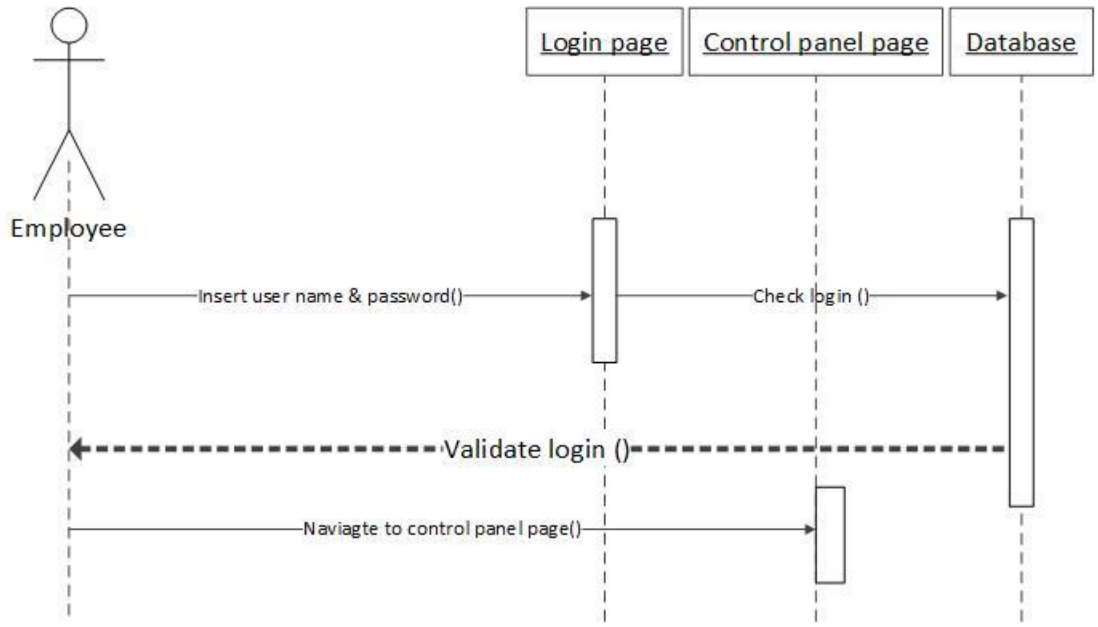
**Sequence diagram for Search Offers( Figure 6):**



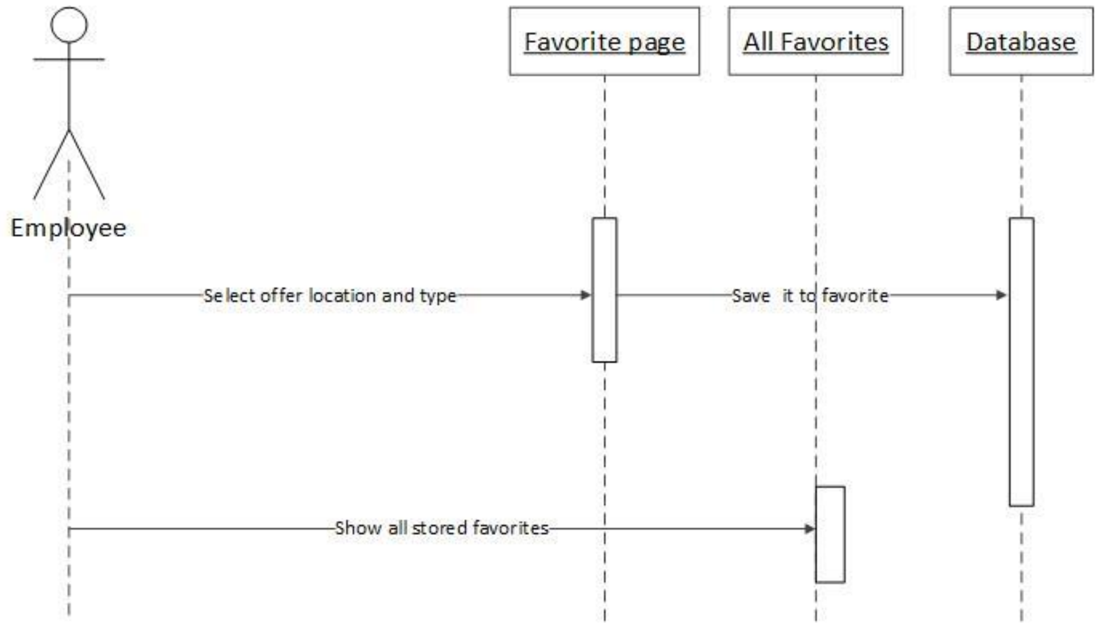
**Sequence diagram for employee registration (figure 7)**



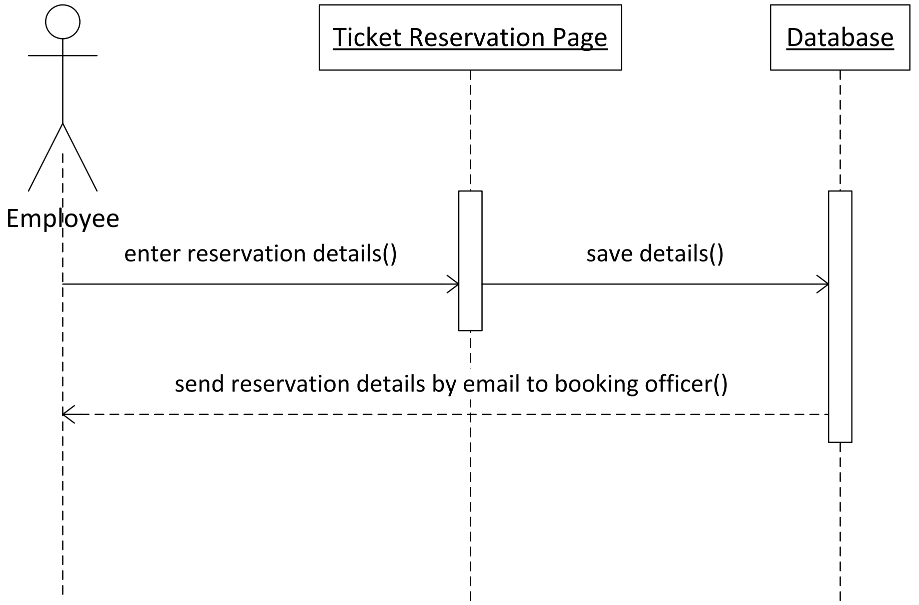
**Sequence diagram for user authentication (figure 8)**



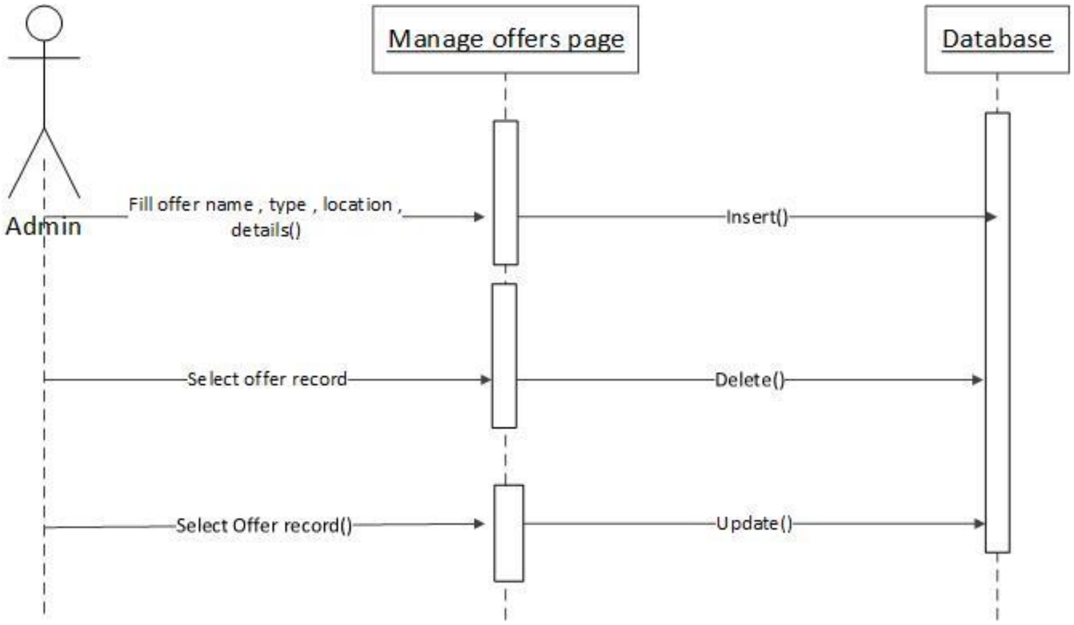
**Sequence diagram for add to favorites (figure 9)**



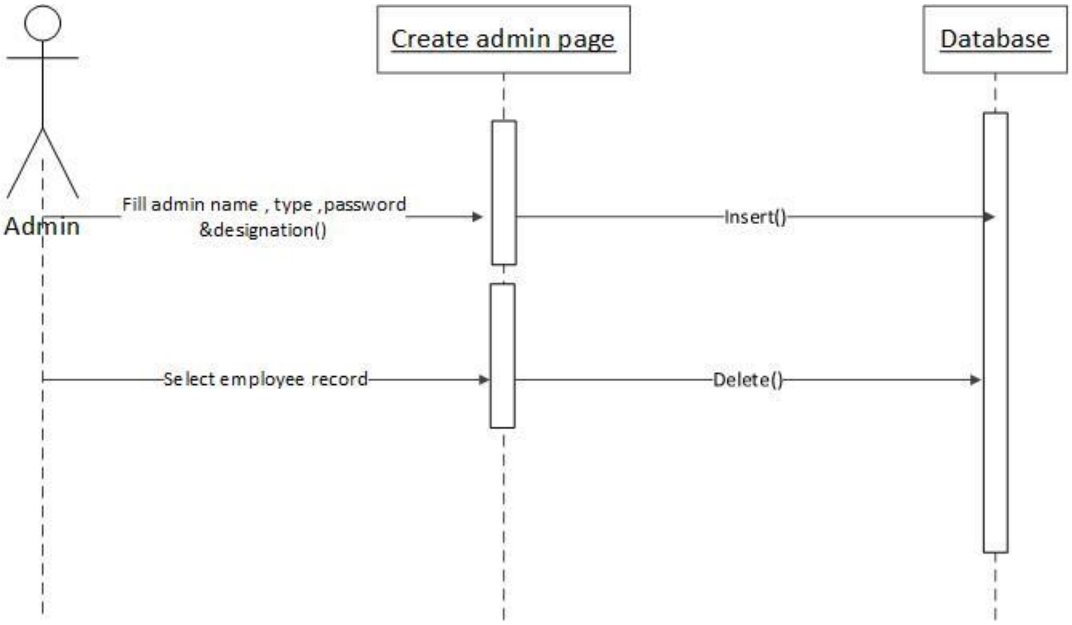
**Sequence diagram for reserve ticket (figure10)**


Ticket Reservation Page
Database
Employee
enter reservation details()
save details()
send reservation details by email to booking officer()

**Sequence diagram for add/delete/update offer (figure 11)**



**Sequence diagram for add admin (figure 12)**



# Chapter Five: IMPLEMENTATION

## 5.1 Used Frame work and programming language

**C#**

C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework.( (Wikipedia, 2016)

In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:

- Encapsulated method signatures called *delegates*, which enable type-safe event notifications.
- Properties, which serve as accessors for private member variables.
- Attributes, which provide declarative metadata about types at run time.
- Inline XML documentation comments.
- Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

(Wikipedia, 2016)

## 5.2 System Interfaces

- **System Home Page**

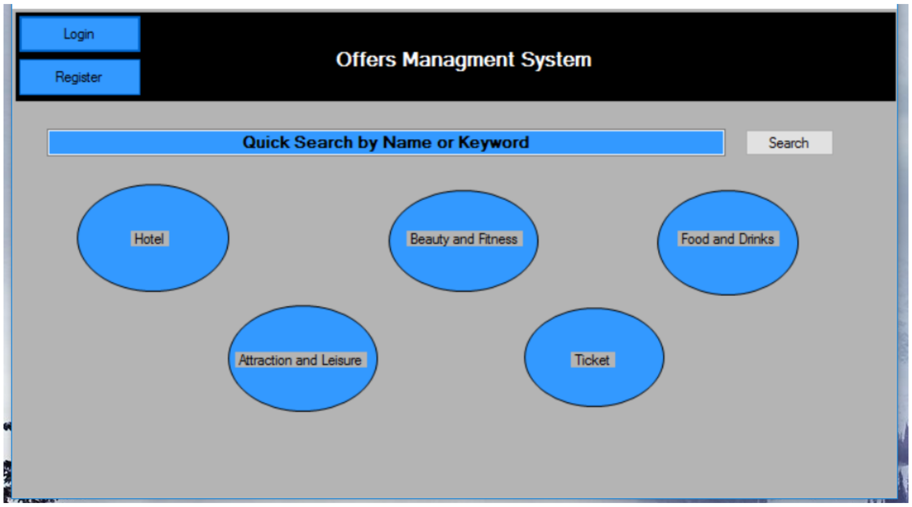


Figure 13 – System interface - Home page

Through the home page of the system the employee can either perform quick search from search bar by typing keyword or by clicking on one of the categories (Hotel, Food & Drinks …. etc)

There is login button to allow users who are already registered in the system to login and another button call register to allow users who still didn't register yet to register.

- **Location list screen**

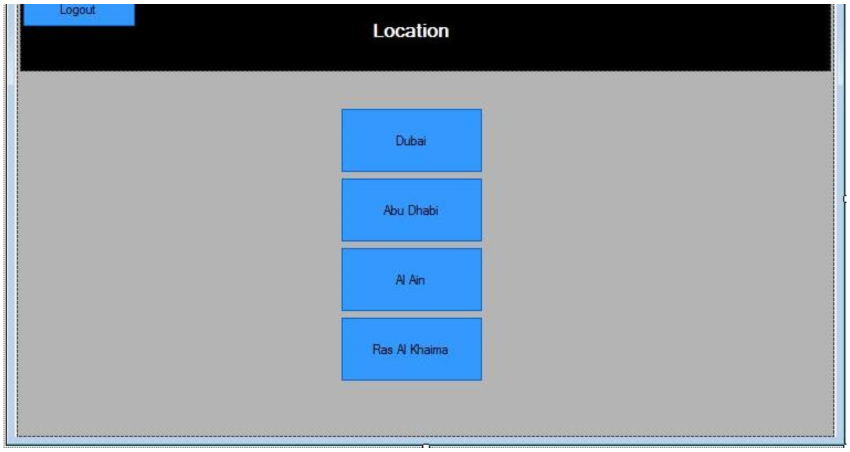


Figure 14 - System interface-location page

When the user choose one of the categories the system filter these categories according to the location so it will be easy for the user to reach what offers he looking for.

- **Registration form screen**

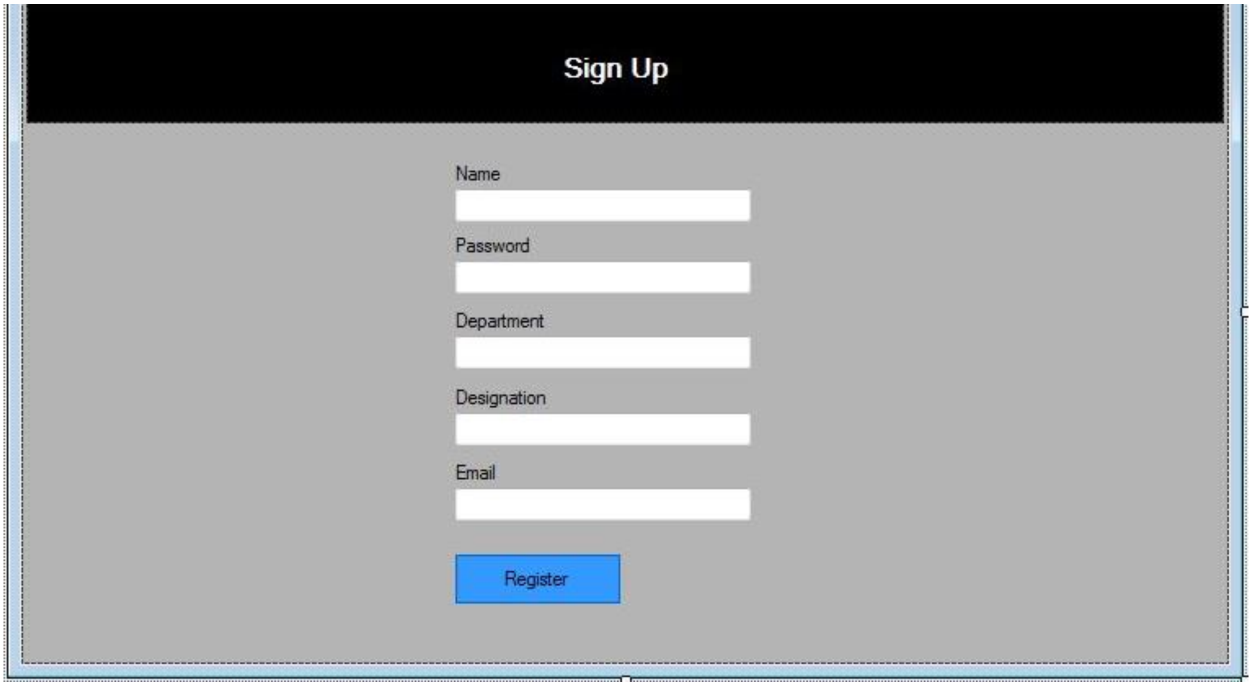


Figure 15 - System interface-registration page

To get more featuers the employee should register by filling this form .

- **Login form**

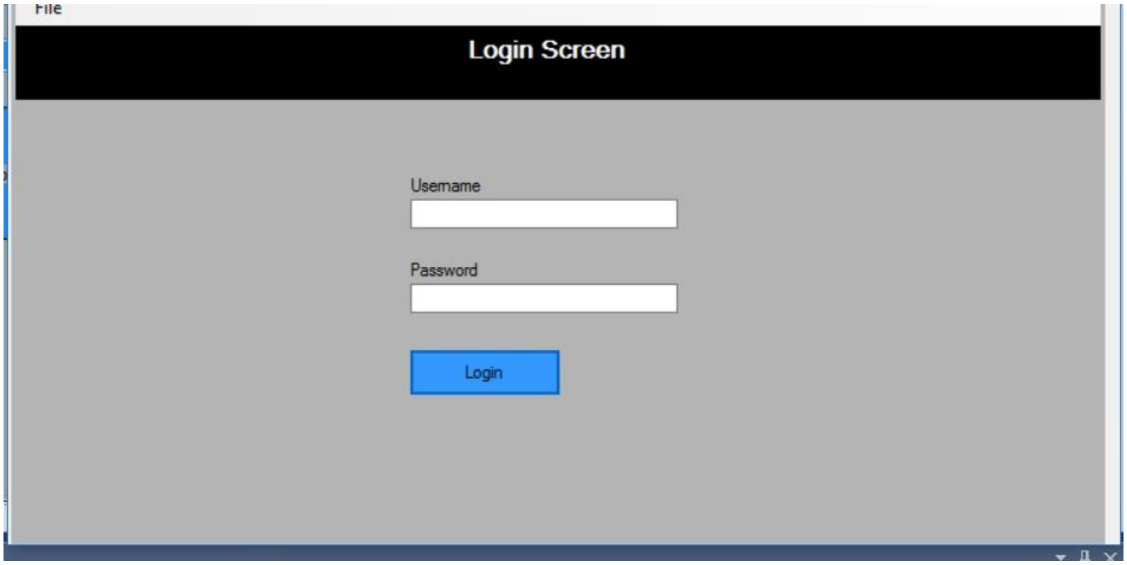


Figure 16– System interface – Login form

In the system we have two type of users, employee and admin of the system so when the user login if he is an employee the system will take him to employee control panel but if admin login the system will take him to admin control panel .

- **Admin Dashboard Screen**

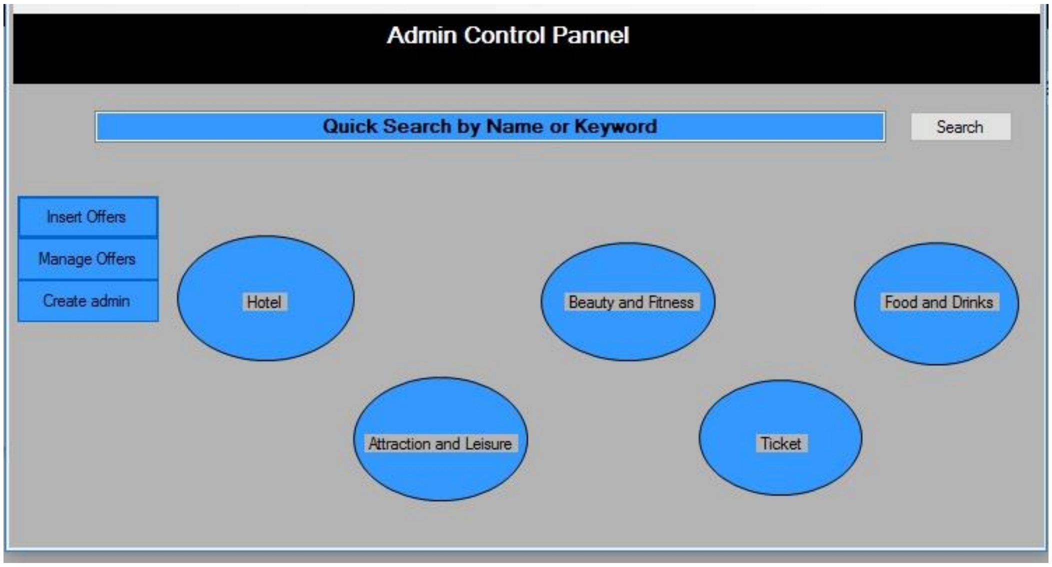


Figure 17 - System interface – admin control panel

This is the control panel for the admin where he can insert /delete / edit offers also he can add / delete admin .

**Add new offer screen**

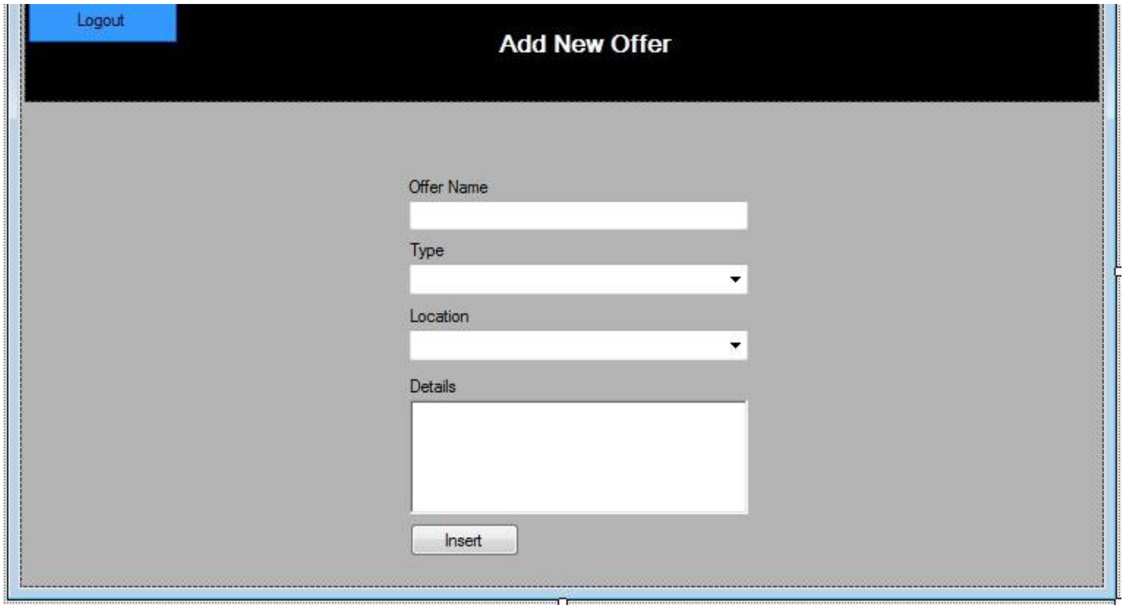
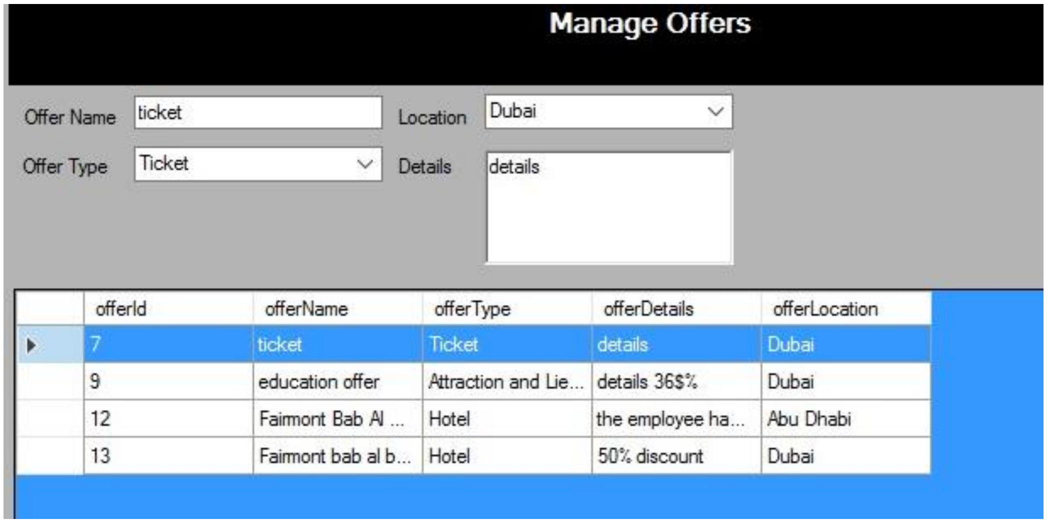


Figure 18 – System interface -Add New Offer page

- **Edit / Delete offer screen**



Manage Offers

Offer Name: ticket

Location: Dubai

Offer Type: Ticket

Details: details

| | offerId | offerName | offerType | offerDetails | offerLocation |
|---|---|---|---|---|---|
| ▶ | 7 | ticket | Ticket | details | Dubai |
| | 9 | education offer | Attraction and Lie... | details 36$% | Dubai |
| | 12 | Fairmont Bab Al ... | Hotel | the employee ha... | Abu Dhabi |
| | 13 | Fairmont bab al b... | Hotel | 50% discount | Dubai |

Figure 19 - System interface – delete / edit offers

- **Add/ delete admin**

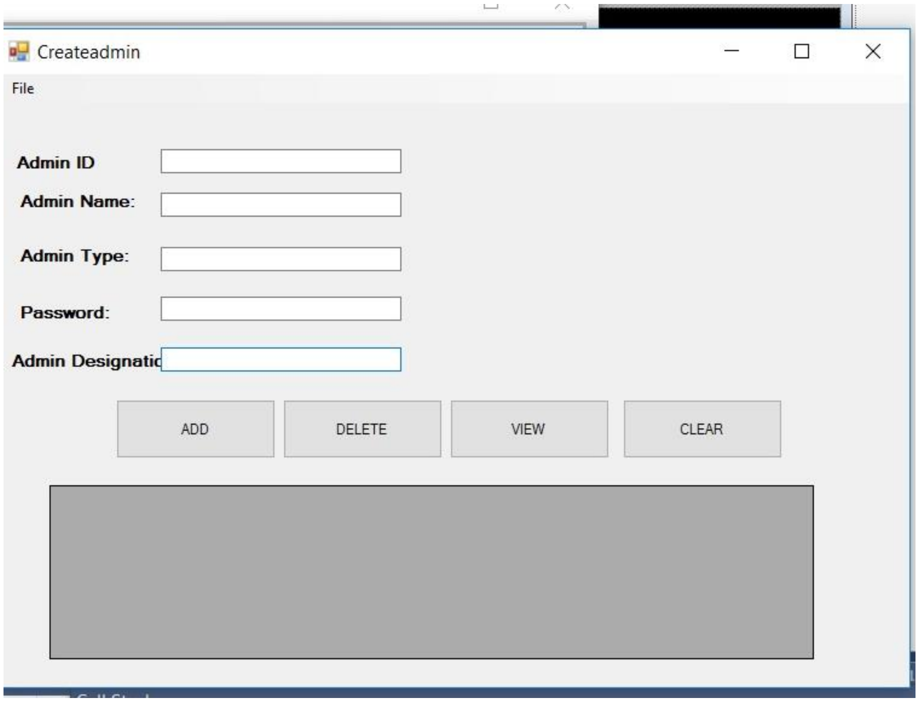


Figure 20 - System interface – add/delete admin

- **Employee control panel**

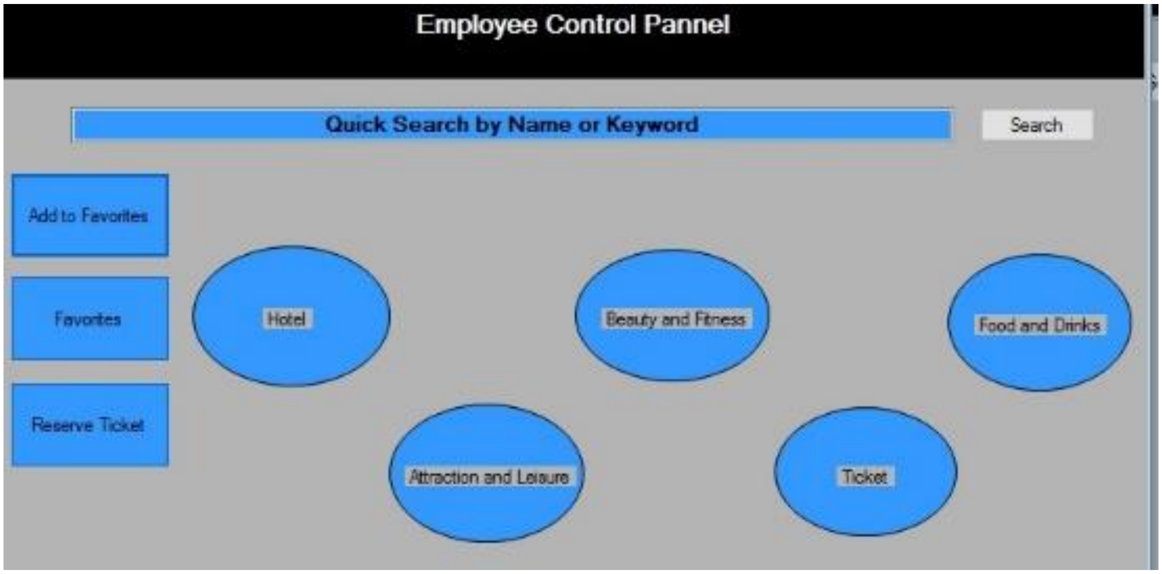
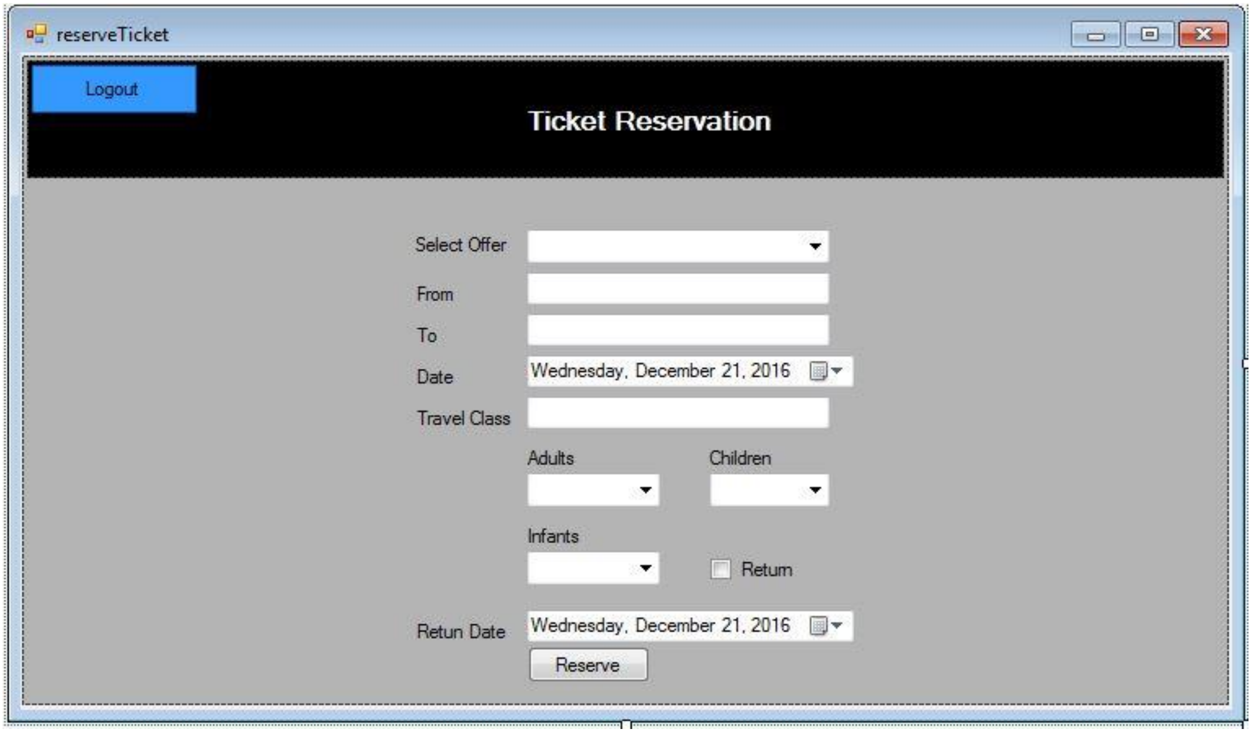


Figure 21 - System interface – employee control panel

Through this form the employee can add to his favorites the offers he liked , he can see the list of offers he added to his favorite before and he can fill form to request a ticket booking from booking officer.

**Ticket reservation screen**

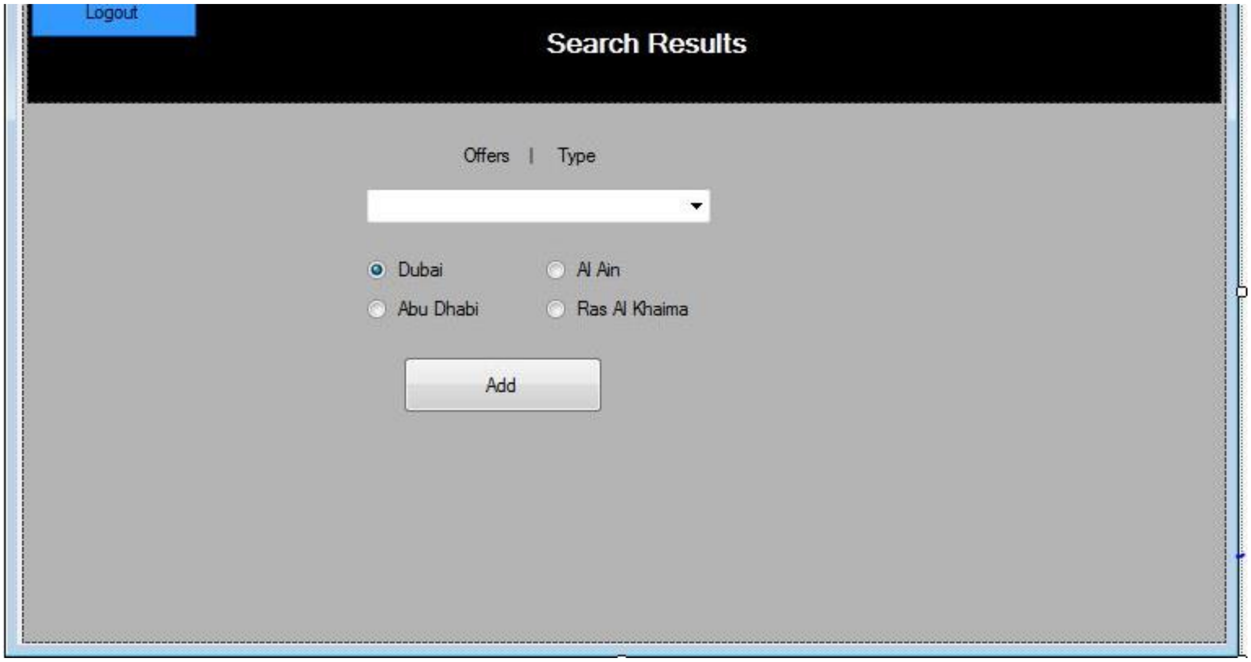


Date: Wednesday, December 21, 2016

Retun Date: Wednesday, December 21, 2016

Figure 22 – System interface -Ticket Reservation page

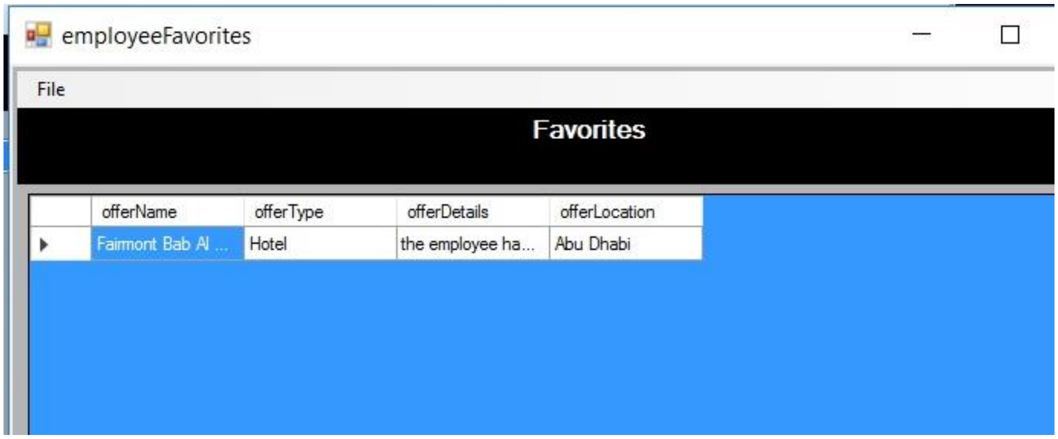
- **Adding offers to favorite**



● Dubai ○ Al Ain
○ Abu Dhabi ○ Ras Al Khaima

Figure 23 - System interface – Adding to favorite

- Report of favorite offers .



| | offerName | offerType | offerDetails | offerLocation |
|---|---|---|---|---|
| ▶ | Fairmont Bab Al ... | Hotel | the employee ha... | Abu Dhabi |

Figure 24 - System interface – favorite report

## 5.3 Testing

### Black box testing

- **Black-box Testing:**

  Black-box testing is used to make sure that the system functions as it's supposed to, and satisfies all user requirements. It verifies whether the system interface and user inputs & outputs all function properly and without any errors.

  In black-box testing, we test the system without worrying about the internal structure of the software program. We just focus on testing the input, and compare the actual result with the expected result. (Guru99, 2016)

  I have used black-box testing to make sure that my system works correctly. Some test cases are given below with their results:

Table 6- system test cases

| Test Case | 01 | 02 | 03 | 04 |
|---|---|---|---|---|
| Login to client end | Successfully get | Error | Successfully get | Successfully get |
| Get offer notification | Successfully get | Successfully get | Error | Successfully get |
| Search offers | Successfully result | Error | Error | Error |
| Apply for ticket | All information send | Information sending failed | Information sending failed | Error |
| Add to favorites | Successfully added to favorite | Error | Error | Error |
| Login to server end | All information must be send to server successfully | All information must be send to server successfully | All information must be send to server successfully | All information must be send to server successfully |
| Add new offers | Successfully update | Error Message | Error Message | Error Message |

| | | | | |
|---|---|---|---|---|
| Update offers | Successfully update | Error Message | Error Message | Error Message |
| Delete offers | Successfully deleted | Error Message | Error Message | Error Message |

## Chapter Six: CONCLUSION AND FUTURE WORK

### 6.1 Summary

The project of Offer Management System was done to fulfill the requirements of GS department of FANR Company who are utilizing email benefit for discount offer administration framework. The venture is finished by experiencing diverse stages beginning from characterizing the issue and arranging finishing with the usage and testing.

All through the venture I found that human asset exercises, for the most part, are critical to be mechanized as they can squander a great deal of representatives' chance and can be inclined to contain many oversights. Offer administration is one of GS department of FANR exercises that ought to be computerized, as well. Utilizing email for offer management is not a decent decision as it is uncertain and problematic.

### 6.2 Contribution of the project

By and large, the venture fulfills the goals. Issue is explored and necessities are gathered through perception and meetings. The discount offer administration framework is designed utilizing distinctive apparatuses, lastly a model for the normal framework is created. The created framework permits a worker to login to the framework to see official discount offers. The administrator can include/redesign and erase offers. Finally, the client can login to the framework to see inquiry and benefit offers.

This framework will make benefits for workers and the organization when all is said in done in a few ways. To begin with, it will spare the season of all gatherings required in offer administration prepare. The administrator won't be required to send manual discount offer messages to the worker. These procedures are mechanized through the proposed framework. Also, the right hand will never again be stressed over ascertaining discount offers for representatives since this will be done consequently by the framework. Having such framework will diminish mistakes that were conceivable to be done because of human errors.

## 6.3 Limitations

There were many obstacles and difficulties I have faced during working with the project. Below are some of them:

Coming out with the problem statement.

It was expected to mention objective fact and meeting with related representatives in the organization to turn out with issue proclamation. This procedure set aside a great deal of opportunity to have a last exact issue proclamation.

Literature review and searching for similar systems.

A lot of time was taken for doing the literature review and finding similar problems and similar systems.

Collecting requirements.

It was required to meet the GS department of FANR, (they are likewise the person who is in charge of discount offer administration), to gather prerequisites. Directors are constantly occupied, and discovering time to meet him a few circumstances was troublesome and additionally timing for discourse was restricted.

System implementation.

This was the greatest deterrent for me in view of the absence of experience for creating desktop applications. The created Offer Management System was my first desktop application, implying that investing a considerable measure of energy looking for how things can be actualized.

### 6.4 Recommendation for further development

The project can be improved and enhanced in the future to better meet the requirements and speed up the process.

The following are some areas that can be improved:

• A month to month timetable view can be added to the framework to show who is withdrawing in what days, this will permit the associate and the administrator to check who is not accessible in a particular day effectively.

• A notice email can be sent to the administrator (i.e. at regular intervals) to let him know what number of solicitations are pending.

• The framework can be incorporated with the GS department of FANR database as opposed to having a different database.

## 7.0 References

[1] Advantages and Disadvantages of SDLC. (2015, March 3). Retrieved 22 September 2015 from https://eternalsunshineoftheismind.wordpress.com/2013/03/03/advantages-anddisadvantages-of-sdlc/

[2] Benefits of SDLC. (2015, February 21). Retrieved 22 September 2015 from https://eternalsunshineoftheismind.wordpress.com/2013/02/21/benefits-of-sdlc/

[3] Microsoft: Why Use C#. Retrieved 12 October 2015 from https://msdn.microsoft.com/enus/library/aa664274(v=vs.71).aspx

[4] Laurie.W (2006). Testing Overview and Black-Box Testing Techniques. Retrieved 4 November 2015 from http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf

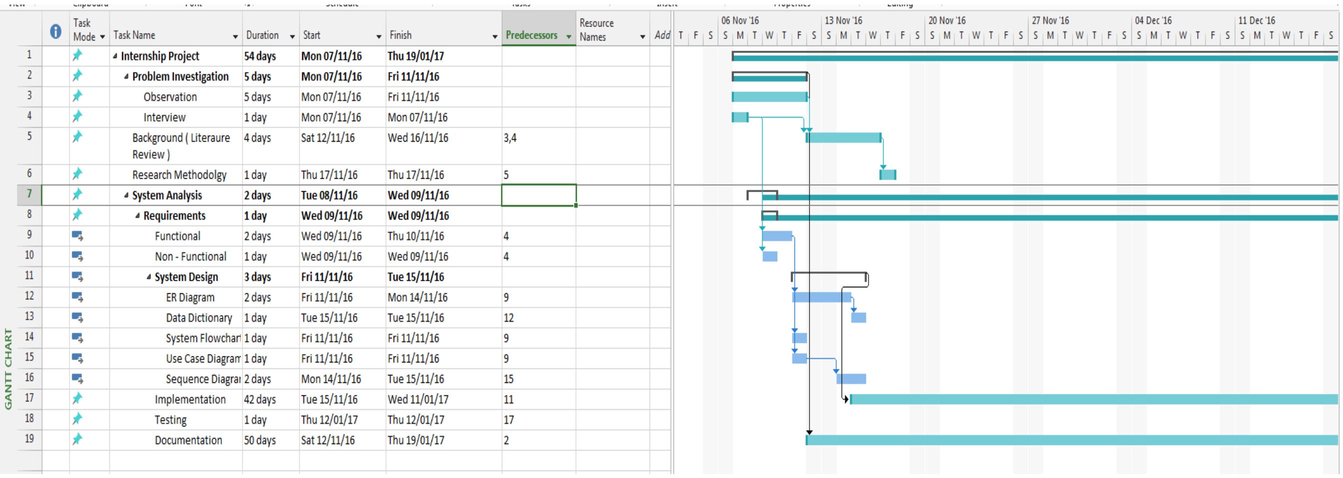
[5] Gaggioli.A. (2015, October 26). 4 Reasons to Stop Using Excel Spreadsheets. Retrieved 6 November 2015 from https://www.cloudbeds.com/articles/4-reasons-to-stop-using-excelspreadsheets/

[6] https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx

## 8.0 APPENDIX

## Appendix A

- **Gantt Chart:**



GANTT CHART
06 Nov '16
13 Nov '16
20 Nov '16
27 Nov '16
04 Dec '16
11 Dec '16
T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S

| | Task Mode | Task Name | Duration | Start | Finish | Predecessors | Resource Names | Add |
|---|---|---|---|---|---|---|---|---|
| 1 | | **Internship Project** | **54 days** | **Mon 07/11/16** | **Thu 19/01/17** | | | |
| 2 | | **Problem Investigation** | **5 days** | **Mon 07/11/16** | **Fri 11/11/16** | | | |
| 3 | | Observation | 5 days | Mon 07/11/16 | Fri 11/11/16 | | | |
| 4 | | Interview | 1 day | Mon 07/11/16 | Mon 07/11/16 | | | |
| 5 | | Background ( Literaure Review ) | 4 days | Sat 12/11/16 | Wed 16/11/16 | 3,4 | | |
| 6 | | Research Methodolgy | 1 day | Thu 17/11/16 | Thu 17/11/16 | 5 | | |
| 7 | | **System Analysis** | **2 days** | **Tue 08/11/16** | **Wed 09/11/16** | | | |
| 8 | | **Requirements** | **1 day** | **Wed 09/11/16** | **Wed 09/11/16** | | | |
| 9 | | Functional | 2 days | Wed 09/11/16 | Thu 10/11/16 | 4 | | |
| 10 | | Non - Functional | 1 day | Wed 09/11/16 | Wed 09/11/16 | 4 | | |
| 11 | | **System Design** | **3 days** | **Fri 11/11/16** | **Tue 15/11/16** | | | |
| 12 | | ER Diagram | 2 days | Fri 11/11/16 | Mon 14/11/16 | 9 | | |
| 13 | | Data Dictionary | 1 day | Tue 15/11/16 | Tue 15/11/16 | 12 | | |
| 14 | | System Flowchart | 1 day | Fri 11/11/16 | Fri 11/11/16 | 9 | | |
| 15 | | Use Case Diagram | 1 day | Fri 11/11/16 | Fri 11/11/16 | 9 | | |
| 16 | | Sequence Diagrai | 2 days | Mon 14/11/16 | Tue 15/11/16 | 15 | | |
| 17 | | Implementation | 42 days | Tue 15/11/16 | Wed 11/01/17 | 11 | | |
| 18 | | Testing | 1 day | Thu 12/01/17 | Thu 12/01/17 | 17 | | |
| 19 | | Documentation | 50 days | Sat 12/11/16 | Thu 19/01/17 | 2 | | |

## Appendix C-1

**Template Code:**

- **Methods created in one class**

```
namespace Offers_Management_System.Database
{    // SINGLETON CLASS
     class DataBaseHandler {
         private static DataBaseHandler dbHandler;

Functions  structure  for Offer Management system :
   private DataBaseHandler() { }
         public static DataBaseHandler DbHandler
         {
             get
             {
                 if (dbHandler == null)
                 {
                     dbHandler = new DataBaseHandler();
                 }
                 return dbHandler;
             }
         }
         //Crecte and Return a OleDbConnection obj.
         private OleDbConnection GetConnection() {
            OleDbConnection conn = new OleDbConnection();
            try {
                String connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data 
Source=C:\Users\nayou\Desktop\Database1.accdb";
                conn = new OleDbConnection(connectionString);
                conn.Open();
            }
            catch (Exception e) {
                Console.WriteLine(e.Message);
            }
            return conn;
        }
        //close the databse connection
          private void CloseConnection(OleDbConnection conn) {
            try {
                conn.Close();
            }
            catch (Exception e) {
                Console.WriteLine(e.Message);
            }
        }
        // used for user validation
        public OleDbDataReader ValidateUser(string userName, string password)
        {
            OleDbConnection conn = GetConnection();

            OleDbCommand cmd = new OleDbCommand("Select * from users where userName='" + 
userName + "' and userPassword='" + password + "'", conn);

            return cmd.ExecuteReader();

        }

        public int RegisterEmployee(string username, string pass, string dept, string 
designation, string email)

        {

            OleDbConnection conn = GetConnection();

            OleDbCommand cmd = new OleDbCommand("Insert into [Users] (userName, 
userPassword, userEmail, employeeDepartment, employeeDesignation, userType) VALUES 
(@userName, @pass, @email, @dept, @dsg, @type)", conn);

            cmd.Parameters.Add("@userName",username);

            cmd.Parameters.Add("@pass",pass);

            cmd.Parameters.Add("@email",email);

            cmd.Parameters.Add("@dept",dept);

            cmd.Parameters.Add("@dsg",designation);

            cmd.Parameters.Add("@type",1);

            return cmd.ExecuteNonQuery();

        }


        public int AddOffer(string offerName, string offerType, string 
offerDetails,string offerLocation)

        {

            OleDbConnection conn = GetConnection();

            OleDbCommand cmd = new OleDbCommand("Insert into [Offers] (userId, offerName, 
offerType, offerDetails, offerPicture, offerLocation) VALUES (@Id, @name, @typ, @details, 
@pic, @loc)", conn);

            //OleDbCommand cmd = new OleDbCommand("Insert into [Offers] (userId, 
offerName, offerType, offerDetails, offerPicture) VALUES (@Id, @name, @typ, @details, 
@pic)", conn);

            cmd.Parameters.Add("@Id", 2);

            cmd.Parameters.Add("@name", offerName);

            cmd.Parameters.Add("@typ", offerType);

            cmd.Parameters.Add("@details", offerDetails);
            cmd.Parameters.Add("@pic", "");
            cmd.Parameters.Add("@loc", offerLocation);
            return cmd.ExecuteNonQuery();

        }
        public DataTable GetOffersTypeLocation(string type, string location)



        {
            OleDbConnection conn = GetConnection();
            OleDbCommand cmd = new OleDbCommand("Select * from Offers where offerType='"
+ type + "' and offerLocation='" + location + "'", conn);
            OleDbDataAdapter da = new OleDbDataAdapter(cmd);
            DataTable dt = new DataTable();
            da.Fill(dt);
            return dt;
        }

        public DataTable GetOffersTypeLocation(string keyword)
        {
            OleDbConnection conn = GetConnection();
            OleDbCommand cmd = new OleDbCommand("Select * from Offers where offerName
LIKE '%" + keyword + "%'", conn);
            OleDbDataAdapter da = new OleDbDataAdapter(cmd);
            DataTable dt = new DataTable();
            da.Fill(dt);
            return dt;
        }
        public DataTable getOffers(string loc)
        {
            OleDbConnection conn = GetConnection();
            OleDbCommand cmd = new OleDbCommand("Select offerName,offerType from Offers
where offerLocation='" + loc + "'",conn);
            OleDbDataAdapter da = new OleDbDataAdapter(cmd);
            DataTable dt = new DataTable();
            da.Fill(dt);
            return dt;
        }
        public DataTable getofferid(string offerName, string offerType) {

            OleDbConnection conn = GetConnection();
            OleDbCommand cmd = new OleDbCommand("Select offerId from Offers where
offerType='" + offerType + "' and offerName='" + offerName + "'", conn);
            OleDbDataAdapter da = new OleDbDataAdapter(cmd);
            DataTable dt = new DataTable();
            da.Fill(dt);
            return dt;

        }
        public int AddtoFav(int offerid, int uid)
        {
            OleDbConnection conn = GetConnection();
            OleDbCommand cmd = new OleDbCommand("Insert into [Favorites] (offerId,
userId) VALUES (@offerid, @uid)", conn);
            cmd.Parameters.Add("@offerId", offerid);
            cmd.Parameters.Add("@userId", uid);
            return cmd.ExecuteNonQuery();

        }

        public DataTable GetFavorites(int userId)
        {
            OleDbConnection conn = GetConnection();
            OleDbCommand cmd = new OleDbCommand("Select 
offerName,offerType,offerDetails,offerLocation from Offers where offerId IN (Select 
offerId from Favorites where userId=" + userId + ")", conn);
            OleDbDataAdapter da = new OleDbDataAdapter(cmd);
            DataTable dt = new DataTable();
            da.Fill(dt);
            return dt;
        }


        public int reserveTickt(int userId,int offerId, string from,string to,bool 
ret,string travelClass,int adult,int child,int infant,string departureDate,string 
returnDate)
        {
            OleDbConnection conn = GetConnection();
            OleDbCommand cmd = new OleDbCommand("Insert into [Bookings] 
(userId,offerId,departure,destination,isReturn,travelClass,children,adults,infants,depart
ureDate,returnDate) VALUES (@uid, @oid, 
@frm,@dest,@rt,@tclas,@ch,@ad,@in,@dd,@rd)",conn);
            cmd.Parameters.Add("@uid", userId);
            cmd.Parameters.Add("@oid", offerId);
            cmd.Parameters.Add("@frm", from);
            cmd.Parameters.Add("@dest", to.ToString());
            cmd.Parameters.Add("@rt", ret.ToString());
            cmd.Parameters.Add("@tclas", travelClass);
            cmd.Parameters.Add("@ch", child);
            cmd.Parameters.Add("@ad", adult);
            cmd.Parameters.Add("@in", infant);
            cmd.Parameters.Add("@dd", departureDate);
            cmd.Parameters.Add("@rd", returnDate);
            return cmd.ExecuteNonQuery();


        }
        public DataTable GetTicketOffers()
        {
            OleDbConnection conn = GetConnection();

            OleDbCommand cmd = new OleDbCommand("Select * from Offers where offerType
='Ticket'", conn);

            OleDbDataAdapter da = new OleDbDataAdapter(cmd);

            DataTable dt = new DataTable();

            da.Fill(dt);

            return dt;

        }


        public DataTable GetUserDetails(int userId)

        {

            OleDbConnection conn = GetConnection();

            OleDbCommand cmd = new OleDbCommand("Select
userName,userEmail,employeeDepartment,employeeDesignation from users where userId=" +
userId + "", conn);

            OleDbDataAdapter da = new OleDbDataAdapter(cmd);

            DataTable dt = new DataTable();

            da.Fill(dt);

            return dt;

        }


        public int deleteOffer(int offerid)

        {

            OleDbConnection conn = GetConnection();

            OleDbCommand cmd = new OleDbCommand("Delete from [Offers] where
offerId=@oid", conn);

            cmd.Parameters.Add("@oid", offerid);

            return cmd.ExecuteNonQuery();

        }


        public int UpdateOffer(int offerId, string offerName, string offerType, string
offerDetails, string offerLocation)

        {
            OleDbConnection conn = GetConnection();
            OleDbCommand cmd = new OleDbCommand("Update Offers Set 
offerName=@name,offerType=@typ,offerDetails=@details,offerLocation=@loc where offerId=" + 
offerId, conn);


            cmd.Parameters.Add("@name", offerName);
            cmd.Parameters.Add("@typ", offerType);
            cmd.Parameters.Add("@details", offerDetails);
            cmd.Parameters.Add("@loc", offerLocation);
            return cmd.ExecuteNonQuery();


        }
        public DataTable getAllOffers()
        {
            OleDbConnection conn = GetConnection();
            OleDbCommand cmd = new OleDbCommand("Select 
offerId,offerName,offerType,offerDetails,offerLocation from Offers", conn);
            OleDbDataAdapter da = new OleDbDataAdapter(cmd);
            DataTable dt = new DataTable();
            da.Fill(dt);
            return dt;
        }


    }
}
```

- **Main Window Form :**

```
namespace Offers_Management_System
{
    public partial class mainWindow : Form
    {
        public mainWindow()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //this.Hide();
            Form loginForm = new login();
            loginForm.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            //this.Hide();
            Form registerationForm = new registration();
            registerationForm.Show();
        }

        private void ovalShape1_Click(object sender, EventArgs e)
        {
            //this.Hide();
            Form locSelction = new locationSelection("Beauty & Fitness");
            locSelction.Show();
        }

        private void ovalShape5_Click(object sender, EventArgs e)
        {
            //this.Hide();
            Form locSelction = new locationSelection("Food & Drinks");
            locSelction.Show();
        }

        private void ovalShape4_Click(object sender, EventArgs e)
        {
            //this.Hide();
            Form locSelction = new locationSelection("Ticket");
            locSelction.Show();
        }

        private void ovalShape3_Click(object sender, EventArgs e)
        {
            //this.Hide();
            Form locSelction = new locationSelection("Attraction and Leisure");
            locSelction.Show();
        }

        private void ovalShape2_Click(object sender, EventArgs e)
        {
            //this.Hide();
            Form locSelction = new locationSelection("Hotel");
            locSelction.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            string keyWord = textBox1.Text.Trim();
            searchResults form = new searchResults(keyWord);
            form.Show();
        }
```

- **Login form :**

```
amespace Offers_Management_System.FrontEnd
{
    public partial class login : Form
    {
        public login()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string userName = textBox1.Text.ToLower().Trim();
            string password = textBox2.Text;
            int[] usertpeId =
frontEndController.FeHandler.validateUser(userName,password);
            int type = usertpeId[0];
            int userid = usertpeId[1];

            if (type == 0)
            {
                adminHomeScreen homeScreen = new adminHomeScreen();
                   // this.Hide();
                    homeScreen.Show();
            }
            else if (type == 1)
            {
              employeeLoginForm homeScreen = new employeeLoginForm(userid);
              homeScreen.Show();
            }
            else
            {
                MessageBox.Show("Invalid Credentials, Please Re-Enter");
            }
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("You sure you want to Quit", "Exit",
MessageBoxButtons.YesNo) == DialogResult.Yes)
            {
                Application.Exit();
            }
        }

        private void homeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            mainWindow f1 = new mainWindow();
            f1.Show();
        }
    }
}
    }
}
```

- **Registration Form :**

```
namespace Offers_Management_System.FrontEnd
{
    public partial class registration : Form
    {
        public registration()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {

              if (ValidateChildren(ValidationConstraints.Enabled))

                {
                    MessageBox.Show("Registerd Successfully!!");


                }

            string userName = textBox1.Text.ToLower().Trim();
            string passWord = textBox2.Text;
            string dept = comboBox1.Items[comboBox1.SelectedIndex].ToString();
            string designation = textBox4.Text.Trim();
            string email = textBox5.Text.Trim();

            if (frontEndController.FeHandler.registerEmployee(userName, passWord, dept, 
designation, email))
            {
                this.Close();


            }
            else
            {
                this.Close();
                MessageBox.Show("Registerd Failed Due to Internal Error");
            }
        }
```

```
private void textBox1_Validating(object sender, CancelEventArgs e)
{
    if (string.IsNullOrEmpty(textBox1.Text))
    {
        e.Cancel = true;
        textBox1.Focus();
        errorProvider1.SetError(textBox1, "This field can't be empty ");
    }
    else
    {
        e.Cancel = false;
        errorProvider1.SetError(textBox1, null);
    }
    }

private void textBox2_Validating(object sender, CancelEventArgs e)
{
    if (string.IsNullOrEmpty(textBox2.Text))
    {
        e.Cancel = true;
        textBox2.Focus();
        errorProvider1.SetError(textBox2, "This field can't be empty ");
    }
    else
    {
        e.Cancel = false;
        errorProvider1.SetError(textBox2, null);
    }
}

private void textBox4_Validating_1(object sender, CancelEventArgs e)
```

```
        {
            if (string.IsNullOrEmpty(textBox4.Text))
            {
                e.Cancel = true;
                textBox4.Focus();
                errorProvider1.SetError(textBox4, "This field can't be empty ");
            }
            else
            {
                e.Cancel = false;
                errorProvider1.SetError(textBox4, null);
            }


        }

        private void textBox5_Validating_1(object sender, CancelEventArgs e)
        {
            if (string.IsNullOrEmpty(textBox5.Text))
            {
                e.Cancel = true;
                textBox5.Focus();
                errorProvider1.SetError(textBox5, "This field can't be empty ");
            }
            Regex BanndChar = new Regex(@"^[a-zA-Z][\w\.-]*[a-zA-Z0-9]@[a-zA-Z0-9][\w\.-
]*[a-zA-Z0-9]\.[a-zA-Z][a-zA-Z\.]*[a-zA-Z]$");


            if (textBox5.Text.Length > 0)
            {


                if (!BanndChar.IsMatch(textBox5.Text))
                {
                    MessageBox.Show("Invalid Email", "Error", MessageBoxButtons.OK);

                }
                else
                {
                    e.Cancel = false;
                    errorProvider1.SetError(textBox5, null);
                }
            }
        }

        private void comboBox1_Validating(object sender, CancelEventArgs e)
        {
            if (string.IsNullOrEmpty(comboBox1.Text))
            {
                e.Cancel = true;
                comboBox1.Focus();
                errorProvider1.SetError(comboBox1, "This field can't be empty ");
            }
            else
            {
                e.Cancel = false;
                errorProvider1.SetError(comboBox1, null);
            }
        }

        private void homeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            mainWindow f1 = new mainWindow();
            f1.Show();
        }


        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("You sure you want to Quit", "Exit",
MessageBoxButtons.YesNo) == DialogResult.Yes)
            {

                Application.Exit();
            }
        }
    }
    }
```

- **Admin Main home from**

```
namespace Offers_Management_System.FrontEnd
{
    public partial class adminHomeScreen : Form
    {
        public adminHomeScreen()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {

        }

        private void button2_Click(object sender, EventArgs e)
        {
            insertOffer insertOfferForm = new insertOffer();
            insertOfferForm.Show();
        }

        private void button7_Click(object sender, EventArgs e)
        {
            string keyWord = textBox1.Text.Trim();
            searchResults form = new searchResults(keyWord);
            form.Show();
        }

        private void ovalShape2_Click(object sender, EventArgs e)
        {
            Form locSelction = new locationSelection("Hotel");
            locSelction.Show();
        }

        private void ovalShape1_Click(object sender, EventArgs e)
        {
            //this.Hide();
            Form locSelction = new locationSelection("Beauty & Fitness");
            locSelction.Show();
        }

        private void ovalShape5_Click(object sender, EventArgs e)
        {
            //this.Hide();
            Form locSelction = new locationSelection("Food & Drinks");
            locSelction.Show();
        }
```

```
private void ovalShape4_Click(object sender, EventArgs e)

{

    //this.Hide();

    Form locSelction = new locationSelection("Ticket");

    locSelction.Show();

}


private void ovalShape3_Click(object sender, EventArgs e)

{

    //this.Hide();

    Form locSelction = new locationSelection("Attraction and Leisure");

    locSelction.Show();

}


private void button3_Click(object sender, EventArgs e)

{

    Form editDleteOffers = new editDeleteOffers();

    editDleteOffers.Show();

}


private void editToolStripMenuItem_Click(object sender, EventArgs e)

{


}


private void homeToolStripMenuItem_Click(object sender, EventArgs e)

{

   mainWindow f1 = new mainWindow();

    f1.Show();

}
```

```
        private void exitToolStripMenuItem_Click(object sender, EventArgs e)

        {

            if (MessageBox.Show("You sure you want to Quit", "Exit",
MessageBoxButtons.YesNo) == DialogResult.Yes)

            {


                Application.Exit();

            }

        }


        private void button5_Click(object sender, EventArgs e)

        {

            Createadmin C1 = new Createadmin();

            C1.Show();

            this.Hide();

        }

    }

}
```

- **Edit Delete Form :**

```
amespace Offers_Management_System.FrontEnd

{

    public partial class editDeleteOffers : Form

    {

        int ID = 0;

        public editDeleteOffers()

        {

            InitializeComponent();

            dataGridView1.DataSource = frontEndController.FeHandler.getAllOffers();


        }
        private void editDeleteOffers_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'database1DataSet1.Offers'
table. You can move, or remove it, as needed.
            dataGridView1.DataSource = frontEndController.FeHandler.getAllOffers();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            if (ID != 0)
            {
                // frontEndController.FeHandler.deleteOffer(ID);
                frontEndController.FeHandler.UpdateOffer(ID, textBox1.Text,
comboBox1.Text, richTextBox1.Text, comboBox2.Text);
                dataGridView1.DataSource = frontEndController.FeHandler.getAllOffers();
                ID = 0;
            }
            else
            {
                MessageBox.Show("Please Select Record to Update");
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (ID != 0)
            {
                frontEndController.FeHandler.deleteOffer(ID);
                dataGridView1.DataSource = frontEndController.FeHandler.getAllOffers();
                ID = 0;
            }
            else
            {
                MessageBox.Show("Please Select Record to Delete");
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void dataGridView1_RowHeaderMouseClick(object sender, 
DataGridViewCellMouseEventArgs e)
        {
            ID = 
Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            comboBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
            comboBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();
            richTextBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("You sure you want to Quit", "Exit", 
MessageBoxButtons.YesNo) == DialogResult.Yes)
            {

                Application.Exit();
            }
        }
        private void homeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            mainWindow f1 = new mainWindow();
            f1.Show();
        }
    }
}
```

- **Create admin form**

```
namespace Offers_Management_System.FrontEnd
{
    public partial class Createadmin : Form
    {
        int ID = 0;

        OleDbConnection conn = new
OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\nayou\Desktop\Database1.accdb");

        public Createadmin()
        {
            InitializeComponent();
            txtPassword.PasswordChar = '*';
            txtPassword.MaxLength = 10;
        }
        private void btnSave_Click(object sender, EventArgs e)
        {
            OleDbConnection conn = new OleDbConnection();
            conn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\nayou\Desktop\Database1.accdb";
String ADMINNAME = txtName.Text;
            String ADMINTYPE = txtType.Text;
            String PASSWORD = txtPassword.Text;

            String designation = txtDes.Text;

            String userId = TxTadmin.Text;


            OleDbCommand cmd = new OleDbCommand("INSERT into users ( userName, userType, 
employeeDesignation ,[userPassword]) 
Values(@AdminName,@AdminType,@designation,@Password)");

            cmd.Connection = conn;

            conn.Open();


            if (conn.State == ConnectionState.Open)

            {

                cmd.Parameters.Add("@AdminName", OleDbType.VarChar).Value = ADMINNAME;

                  cmd.Parameters.Add("@AdminType", OleDbType.VarChar).Value = ADMINTYPE;

                 cmd.Parameters.Add("@designation", OleDbType.VarChar).Value = 
designation;

                cmd.Parameters.Add("@Password", OleDbType.VarChar).Value = PASSWORD;


                try

                {

                    cmd.ExecuteNonQuery();

                    MessageBox.Show("Admin Added");

                    conn.Close();


                }

                catch (OleDbException ex)

                {


                    MessageBox.Show(ex.Source);

                    conn.Close();
                }



            }

            else { MessageBox.Show("Connection Failed"); }

        }

        private void Createadmin_Load(object sender, EventArgs e)
        {
            this.usersTableAdapter.Fill(this.database1DataSet1.Users);


        }

        private void btnView_Click(object sender, EventArgs e)
        {
            conn.Open();
            OleDbCommand cmd = conn.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "select userId ,userName ,userType,[userPassword],
employeeDesignation from users";
            cmd.ExecuteNonQuery();
            DataTable dt = new DataTable();
            OleDbDataAdapter da = new OleDbDataAdapter(cmd);
            da.Fill(dt);
            dataGridView1.DataSource = dt;
            conn.Close();
        }

        private void btnClear_Click(object sender, EventArgs e)
        {

            txtName.Text = String.Empty;
            txtType.Text = String.Empty;
            txtPassword.Text = String.Empty;
            txtDes.Text = String.Empty;

        }

        private void btnDelet_Click(object sender, EventArgs e)
        {

            conn.Open();
            OleDbCommand cmd = conn.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "delete from Users where userName='" + txtName.Text + "'"
            cmd.ExecuteNonQuery();
            conn.Close();

            if (MessageBox.Show("You sure you want to Delete", "Exit",
MessageBoxButtons.YesNo) == DialogResult.Yes)
            {

                MessageBox.Show("Record Deleted Sucessfully");
            }
        }

        private void dataGridView1_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
        {
            ID =
Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());
```

```
            txtName.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            txtType.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
           txtPassword.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
            txtDes.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();
        }

        private void homeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            mainWindow f1 = new mainWindow();
            f1.Show();
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {

            if (MessageBox.Show("You sure you want to Quit", "Exit", 
MessageBoxButtons.YesNo) == DialogResult.Yes)
            {

                Application.Exit();
            }
        }
    }
}
```

- **Insert offer form :**

```
namespace Offers_Management_System.FrontEnd
{
    public partial class insertOffer : Form
    {
        public insertOffer()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            string offerName = textBox1.Text.Trim();
            string type = comboBox1.SelectedItem.ToString();
            string loc = comboBox2.SelectedItem.ToString();
            string details = richTextBox1.Text.Trim();
            if (frontEndController.FeHandler.addOffer(offerName,type,details,loc))
            {
                this.Close();
                MessageBox.Show("Offer Added");
            }
            else
            {
                this.Close();
                MessageBox.Show("Offer Insertion Failed Due to Internal Error");
            }
        }

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        private void homeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            mainWindow f1 = new mainWindow();
            f1.Show();
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {

            if (MessageBox.Show("You sure you want to Quit", "Exit", 
MessageBoxButtons.YesNo) == DialogResult.Yes)
            {

                Application.Exit();
            }
        }
    }
}
```

- **Employee Login form**

```
namespace Offers_Management_System.FrontEnd
{
    public partial class employeeLoginForm : Form
    {
        int userID;
        public employeeLoginForm()
        {
            InitializeComponent();
        }
        public employeeLoginForm(int userId)
        {
            InitializeComponent();
            userID = userId;
        }

        private void ovalShape5_Click(object sender, EventArgs e)
        {
            Form locSelction = new locationSelection("Food & Drinks");
            locSelction.Show();
        }

        private void ovalShape4_Click(object sender, EventArgs e)
        {
            Form locSelction = new locationSelection("Ticket");
            locSelction.Show();
        }

        private void ovalShape3_Click(object sender, EventArgs e)
        {
            Form locSelction = new locationSelection("Attraction and Leisure");
            locSelction.Show();
        }

        private void ovalShape2_Click(object sender, EventArgs e)
        {
            Form locSelction = new locationSelection("Hotel");
            locSelction.Show();
        }

        private void ovalShape1_Click(object sender, EventArgs e)
        {
            Form locSelction = new locationSelection("Beauty & Fitness");
            locSelction.Show();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form adFav = new addToFav(userID);

            adFav.Show();

        }


        private void button3_Click(object sender, EventArgs e)

        {

            string keyWord = textBox1.Text.Trim();

            searchResults form = new searchResults(keyWord);

            form.Show();

        }


        private void button2_Click_1(object sender, EventArgs e)

        {

            Form fav = new employeeFavorites(this.userID);

            fav.Show();

        }


        private void button3_Click_1(object sender, EventArgs e)

        {

            Form resTic = new reserveTicket(this.userID);

            resTic.Show();

        }


        private void homeToolStripMenuItem_Click(object sender, EventArgs e)

        {

            mainWindow f1 = new mainWindow();

            f1.Show();

        }

    }

}
```

- **Employee Favorite form :**

```
namespace Offers_Management_System.FrontEnd
{
    public partial class employeeFavorites : Form
    {
        public employeeFavorites()
        {
            InitializeComponent();
        }
        public employeeFavorites(int uId)
        {
            InitializeComponent();
            dataGridView1.DataSource = frontEndController.FeHandler.getFavorites(uId);
        }

        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {

        }

        private void homeToolStripMenuItem_Click(object sender, EventArgs e)
        {

            mainWindow f1 = new mainWindow();
            f1.Show();
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("You sure you want to Quit", "Exit",
MessageBoxButtons.YesNo) == DialogResult.Yes)
            {

                Application.Exit();
            }
        }

    }
}
```

- **Add to favorite form :**

```
namespace Offers_Management_System.FrontEnd
{
    public partial class addToFav : Form
    {
        int userId;
        public addToFav()
        {
            InitializeComponent();
            if (radioButton1.Checked)
            {
                comboBox1.DataSource = frontEndController.FeHandler.getOffers("Dubai");
            }

        }
        public addToFav(int userId)
        {
            InitializeComponent();
            if (radioButton1.Checked)
            {
                comboBox1.DataSource = frontEndController.FeHandler.getOffers("Dubai");
            }
            this.userId = userId;

        }

        private void button2_Click(object sender, EventArgs e)
        {
            string slectedItem = comboBox1.SelectedItem.ToString();
            string[] tokens = slectedItem.Split('|');
            string offrName = tokens[0].Trim();
            string offerType = tokens[1].Trim();
            int offerId = frontEndController.FeHandler.getOfferId(offrName,offerType);
            int uid = this.userId;
            int status=frontEndController.FeHandler.insertToFav(offerId,uid);

            if (status > 0) {

                MessageBox.Show("Sucessfully Added");
            }
        }

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        private void radioButton1_CheckedChanged(object sender, EventArgs e)
        {
             comboBox1.DataSource =  frontEndController.FeHandler.getOffers("Dubai");
        }
        private void radioButton3_CheckedChanged(object sender, EventArgs e)
        {
            comboBox1.DataSource = frontEndController.FeHandler.getOffers("Al Ain");
        }

        private void radioButton2_CheckedChanged(object sender, EventArgs e)
        {
            comboBox1.DataSource = frontEndController.FeHandler.getOffers("Abu Dhabi");
        }

        private void radioButton4_CheckedChanged(object sender, EventArgs e)
        {
            comboBox1.DataSource = frontEndController.FeHandler.getOffers("Ras Al
Khaima");
        }

        private void panel2_Paint(object sender, PaintEventArgs e)
        {

        }
    }
}
```

- **Location Selecting form**

```
namespace Offers_Management_System.FrontEnd
{
    public partial class locationSelection : Form
    {
        string offerType;
        public locationSelection()
        {
            InitializeComponent();
        }
        public locationSelection(string type)
        {
            this.offerType = type;
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
           // this.Hide();
            Form searchResults = new searchResults(offerType,"Dubai");
            searchResults.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Form searchResults = new searchResults(offerType,"Abu Dhabi");
            searchResults.Show();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            Form searchResults = new searchResults(offerType, "Al Ain");
            searchResults.Show();
        }

        private void button5_Click(object sender, EventArgs e)
        {
            Form searchResults = new searchResults(offerType, "Ras Al Khaima");

            searchResults.Show();

        }


        private void exitToolStripMenuItem_Click(object sender, EventArgs e)

        {

            if (MessageBox.Show("You sure you want to Quit", "Exit",
MessageBoxButtons.YesNo) == DialogResult.Yes)

            {


                Application.Exit();

            }

        }


        private void homeToolStripMenuItem_Click(object sender, EventArgs e)

        {


            mainWindow f1 = new mainWindow();

            f1.Show();

        }

    }

}
```

- **Reserve Ticket form:**

```
namespace Offers_Management_System.FrontEnd

{

    public partial class reserveTicket : Form

    {

        int userId;

        public reserveTicket()

        {

            InitializeComponent();
```

```
}

public reserveTicket(int uid)
{
    InitializeComponent();
    this.userId = uid;
    comboBox4.DataSource = frontEndController.FeHandler.getTicketOffers();
}

private void button2_Click(object sender, EventArgs e)
{
    string from = textBox1.Text.Trim();
    string to = textBox2.Text.Trim();
    string travelClass = textBox3.Text.Trim();
    string DepartureDate = dateTimePicker1.Value.Date.ToShortDateString();
    string returnDate = dateTimePicker2.Value.Date.ToShortDateString();
    bool ret = checkBox1.Checked;
    int adult = 1;
    int child = 0;
    int infant = 0;
    if (comboBox3.SelectedItem != null)
    {
         child = Convert.ToInt32(comboBox3.SelectedItem.ToString());
    }
    if(comboBox2.SelectedItem != null)
    {
        infant = Convert.ToInt32(comboBox2.SelectedItem.ToString());
    }
    if(comboBox1.SelectedItem != null)
    {
        adult = Convert.ToInt32(comboBox1.SelectedItem.ToString());
```

```
            }
            int offerId =
frontEndController.FeHandler.getOfferId(comboBox4.SelectedItem.ToString(), "Ticket");


                int status = frontEndController.FeHandler.reserveTicket(userId, offerId,
from, to, ret, travelClass, adult, child, infant, DepartureDate, returnDate);
           if (status > 0)
           {
              string emailBody = frontEndController.FeHandler.createEmailBody(userId,
offerId, from, to, ret, travelClass, adult, child, infant, DepartureDate,
returnDate,comboBox4.SelectedItem.ToString());
   frontEndController.FeHandler.sendEmail("noura.m.alneyadi@gmail.com", "Ticket
Reservation Request", "offersmanagmentsystem@gmail.com", emailBody);
               MessageBox.Show("Sucessfully Email Request Sent to Department");
           }
        }
        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            if (checkBox1.Checked)
            {
                label5.Visible = true;
                dateTimePicker2.Visible = true;
            }
            else
            {
                label5.Visible = false;
                dateTimePicker2.Visible = false;
            }
        }
        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {

            if (MessageBox.Show("You sure you want to Quit", "Exit",
MessageBoxButtons.YesNo) == DialogResult.Yes)
            {
                Application.Exit();
            }
        }
        private void homeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("You sure you want to Quit", "Exit",
MessageBoxButtons.YesNo) == DialogResult.Yes)
            {
                Application.Exit();
            }
            mainWindow f1 = new mainWindow();
            f1.Show();
        }
    }
}
Search result Form : namespace Offers_Management_System.FrontEnd
{
    public partial class searchResults : Form
    {
        public searchResults()
        {
            InitializeComponent();
        }
        public searchResults(string type,string location)
        {
                InitializeComponent();
                DataTable dt = frontEndController.FeHandler.GetOfferTypeLoc(type,
location);
            // dataGridView
            dataGridView1.DataSource = dt;
        }
        public searchResults(string keyword)
        {
            InitializeComponent();
            DataTable dt = frontEndController.FeHandler.GetOfferTypeLoc(keyword);
            dataGridView1.DataSource = dt;
        }
        private void dataGridView1_CellContentClick(object sender, 
DataGridViewCellEventArgs e)
        {
        }
        private void homeToolStripMenuItem_Click(object sender, EventArgs e)
        {
            mainWindow f1 = new mainWindow();
            f1.Show();
        }
        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (MessageBox.Show("You sure you want to Quit", "Exit", 
MessageBoxButtons.YesNo) == DialogResult.Yes)
            {
                Application.Exit();
            }

        }
           }
}
```