Kingdom of Saudi Arabia
Ministry of Education
Majmaah University
College of Science Al Zulfi

المملكة العربية السعودية
وزارة التعليم
جامعة المجمعة
كلية العلوم بالزلفى
قسم علوم الحاسب والمعلومات

HELP DESK

Student Affairs System
For College of science Al Zulfi
Department of Computer Science and Information

Graduation Project
Submitted in partial fulfillment of the requirements for the award of
Bachelor degree of the Majmaah University
(Semester 2, 2020)

Submitted by:
Ahmed AlQarni,
And
Abdullah AlShahrani

Under the supervision of:
Dr. Hassan Aly

Abstract

This project is to design and implement an android mobile application, Called "Help Desk", to answer any questions and queries, easily and rapidly, for students at college of science. Queries on timetable, student's affairs, activities, and exams will be provided by the system. This application finds links between students and their colleage.

Acknowledgement

I would like to thank my supervisor of this project Dr. Hassan Aly, special thanks for his supervision. When we needed any help, he was there for us. He gave us all the information and guidance to achieve this project.

**MAJMAAH UNIVERSITY,
COLLEGE OF SCIENCE AL ZULFI,
DEPARTMENT OF COMPUTER SCIENCE AND
INFORMATION**

(CERTIFICATE BY STUDENT)

This is to certify that the project titled **"Help Desk"** submitted by (**Ahmed**

Al-Qarni 342101190 and **Abdullah Al-Shahrani 342106750**) under the supervision

of **Dr. Hassan Aly** for award of Bachelor degree of the Majmaah University carried out during

the second Semester, 2019-2020 embodies our original work.

Signature in full: _____

Name in block letters:

Student ID:

Date

Table of Content

Chapter 1: Introduction

1.1.1.1 Problem Definition

This project is to design and implement an android mobile application, Called "Help Desk", to answer any questions and queries, easily and rapidly, for students at college of science. Queries on timetable, student's affairs, activities, and exams will be provided by the system. This application finds links between students and their college. The current system of the university has problem like limitation and unavailability of information to answer rapidly, any queries of student. Our system tries to be a faster
and accurate alternative for students.

1.2 The beneficiary of the program

The student is the first beneficiary of the program, where he can inquire about the questions of registration in the college and specializations available at the college, and will provide them with all the necessary answers to their inquiries and problems

1.2.1 Objectives

- Help Desk is a program that provides assistance to all students' queries and complaints.
- It is an android-based software which helps students to search and find anything at the campus in an easy way.
- The program implements the solutions of the registered student inside or outside the university.
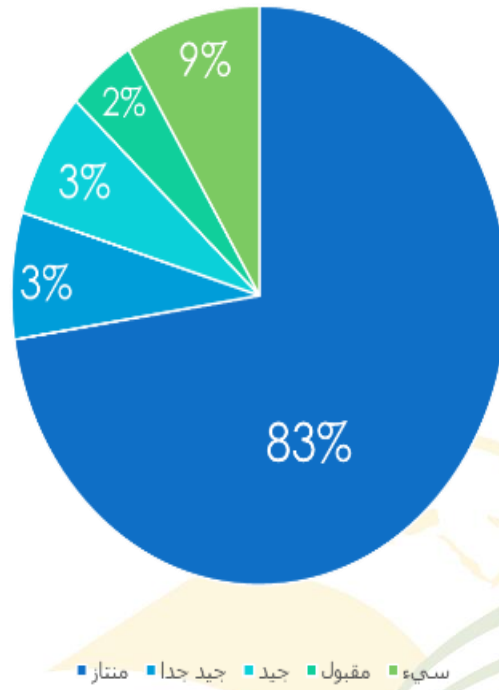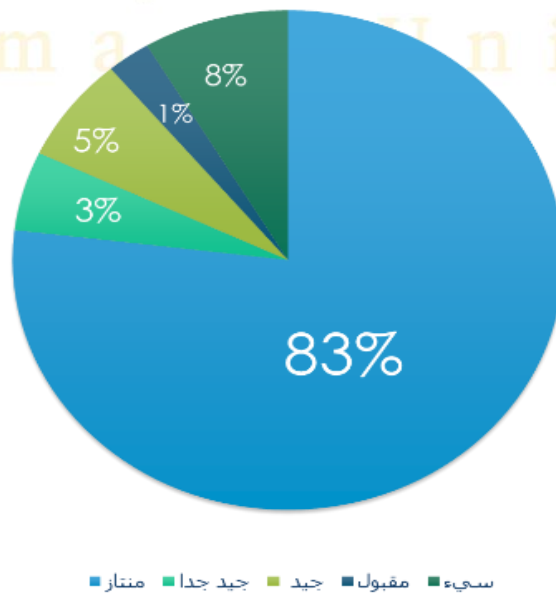
1.2.2 Feasibility study

Survey

- 1- ما هو تقييمك لمدى جودة الفكرة قبل التنفيذ؟

- 2- ما هو تقييمك لمقدار الفائدة التي تعود على المستخدم من التعامل مع التطبيق؟

- 3- هل يساعد التطبيق الطلاب و الموظفين في العمل؟

- 4- هل تستحق الفكرة التنفيذ أم لا؟
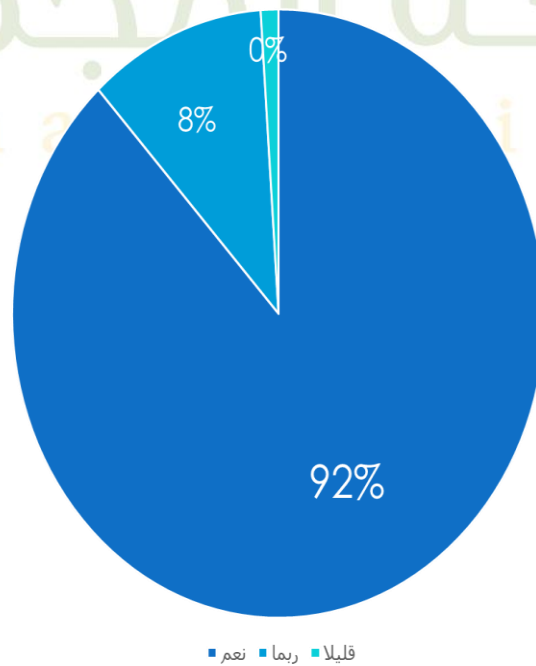
- 5- هل سبق شاهدت فكرة مشابهه لها من قبل؟
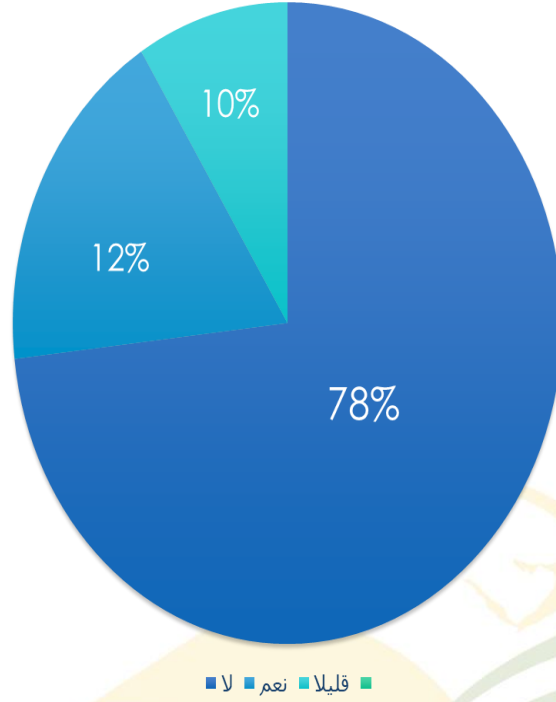
- 6- هل تنصح الطلاب الجدد في استخدام التطبيق لسهولة الحصول على المعلومة؟

- 7- هل يحتاج التطبيق إلى التطوير في المستقبل؟

- 8- هل يستهدف التطبيق شريحة معينة من المستخدمين؟

1- ما هو تقييمك لمدى جودة الفكرة قبل التنفيذ؟
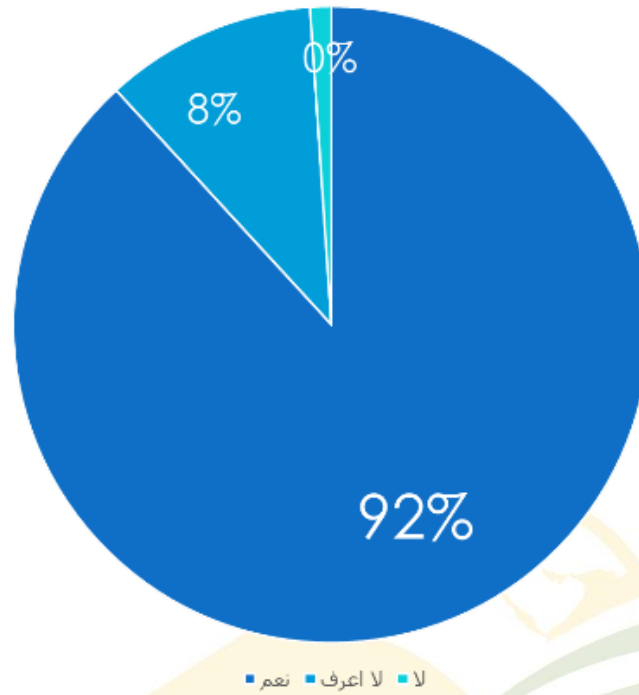

83%

9%

2%

3%

3%

سيء ■ مقبول ■ جيد جدا ■ جيد ■ منتاز ■

2- ما هو تقييمك لمقدار الفائدة التي تعود على المستخدم من التعامل مع التطبيق؟

8%

1%

5%

3%

83%

سيء ■ مقبول ■ جيد ■ جيد جدا ■ منتاز ■

3- هل يساعد التطبيق الطلاب و الموظفين في العمل؟

0%

8%

92%

لا ■   لا اعرف ■   نعم ■

4- هل تستحق الفكرة التنفيذ أم لا؟

0%

8%

92%

قليلا ■   ربما ■   نعم ■

5- هل سبق شاهدت فكرة مشابهه لها من قبل؟

78%

12%

10%

قليلا ■ نعم ■ لا ■

6- هل تنصح الطلاب الجدد في استخدام التطبيق لسهولة الحصول على المعلومة؟

95%

3%

2%

لا اعرف ■ لا ■ نعم ■

7- هل يحتاج التطبيق الى تطوير في المستقبل؟

56%

37%

7%

لا   ■ لست متخصص   ■ نعم

8- هل يستهدف التطبيق شريحة معينة من المستخدمين؟

22%

7%

71%
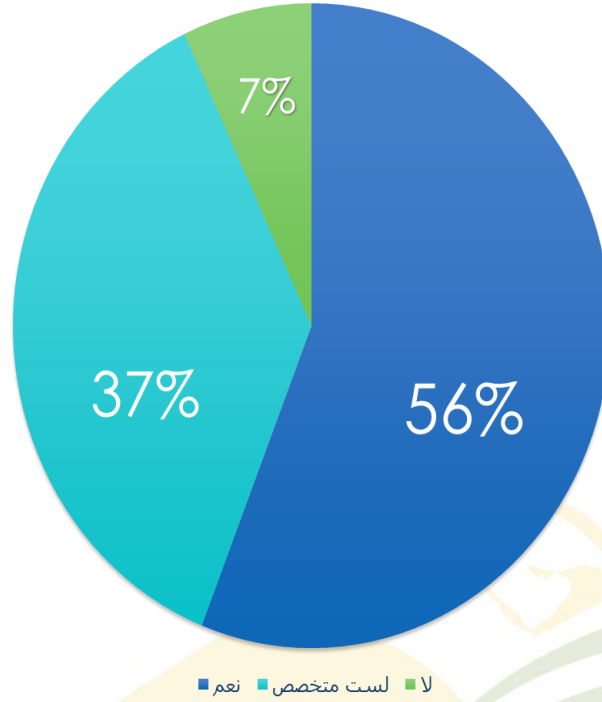
نعم هو متاح للطلبة فقط   ■ لا اعرف   ■ نعم هو متاح للجميع المستخدمين

1.2.3 Gantt Chart

ID	Activity	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
		2019-2020							
1	Requirements	▬▬							
2	Analysis		▬▬						
3	Design			▬▬					
4	Coding				▬▬				
5	Implementation					▬▬			
6	Testing						▬▬		
7	Maintenance							▬▬	

CHAPTER 2:

SYSTEM ANALYSIS AND SPECIFICATION

2.1 Introduction

The proposed system will contain three sections:

1. Admin: create departments, assign query to Helpers and create users.
2. Helper: reply query and answer live chat.
3. Students: start live chat, create query and receive answer.

Use Case

In software and systems engineering, a use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language (UML) as an actor) and a system to achieve a goal. The actor can be a human or other external system. In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals. The detailed requirements may then be captured in the Systems Modeling Language (SysML) or as contractual statements.

❖ Use Case for Student

The student can create a new query, modify the previously created query, and delete it if he wants as well as can communicate directly with the live chat.
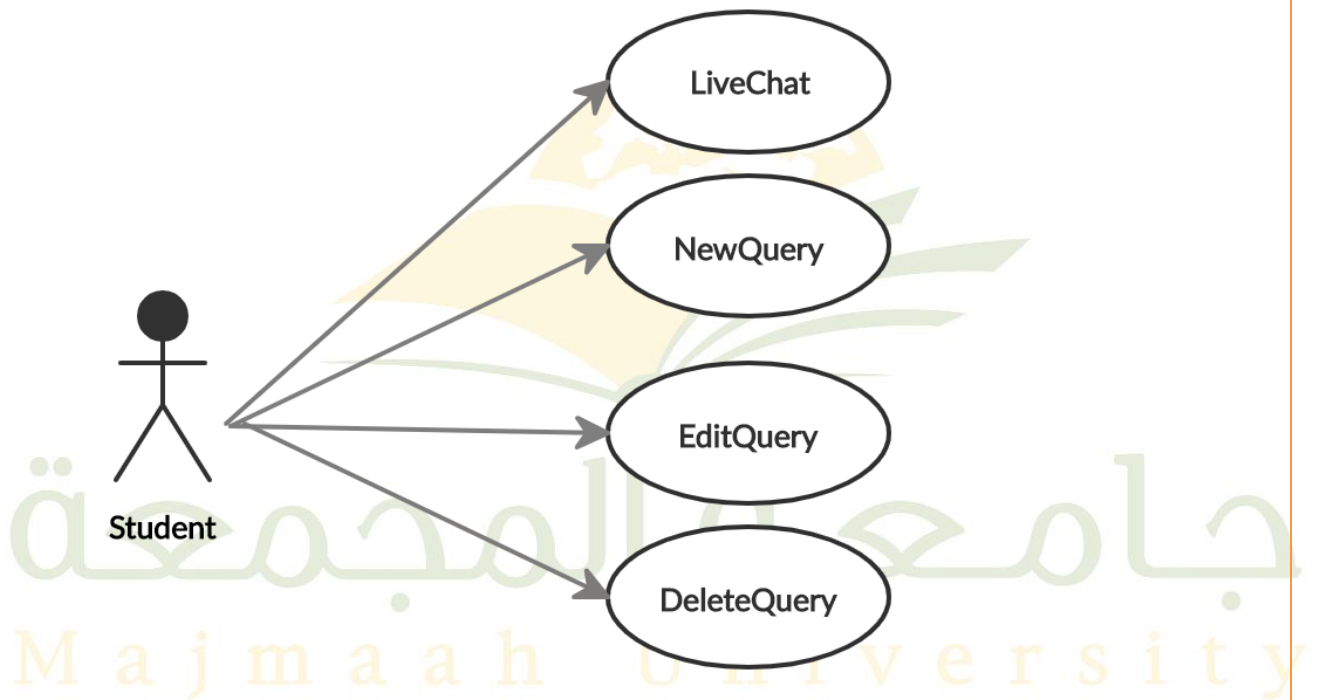

Fig 3.1: Use case diagram for student

Use Case for Helper

Helper can create a query for the admin and can modify the queries and delete and move the queries to the appropriate section and respond to queries as he can reply to live chat.
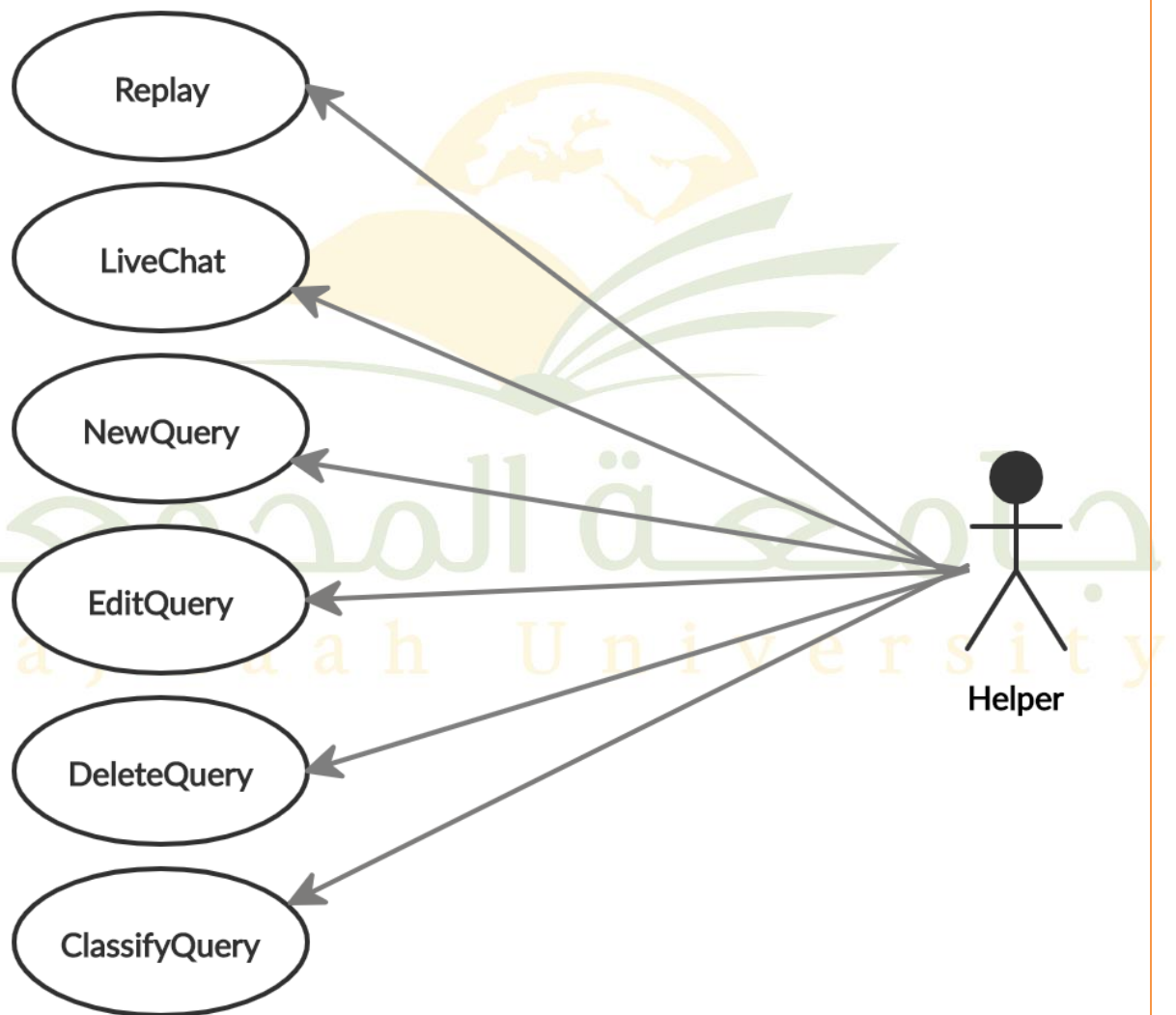

Fig 3.2: Use case diagram for student

❖ **Use Case for Admin**

The admin can use live chat, delete queries, create department , add a new helper and add new students.
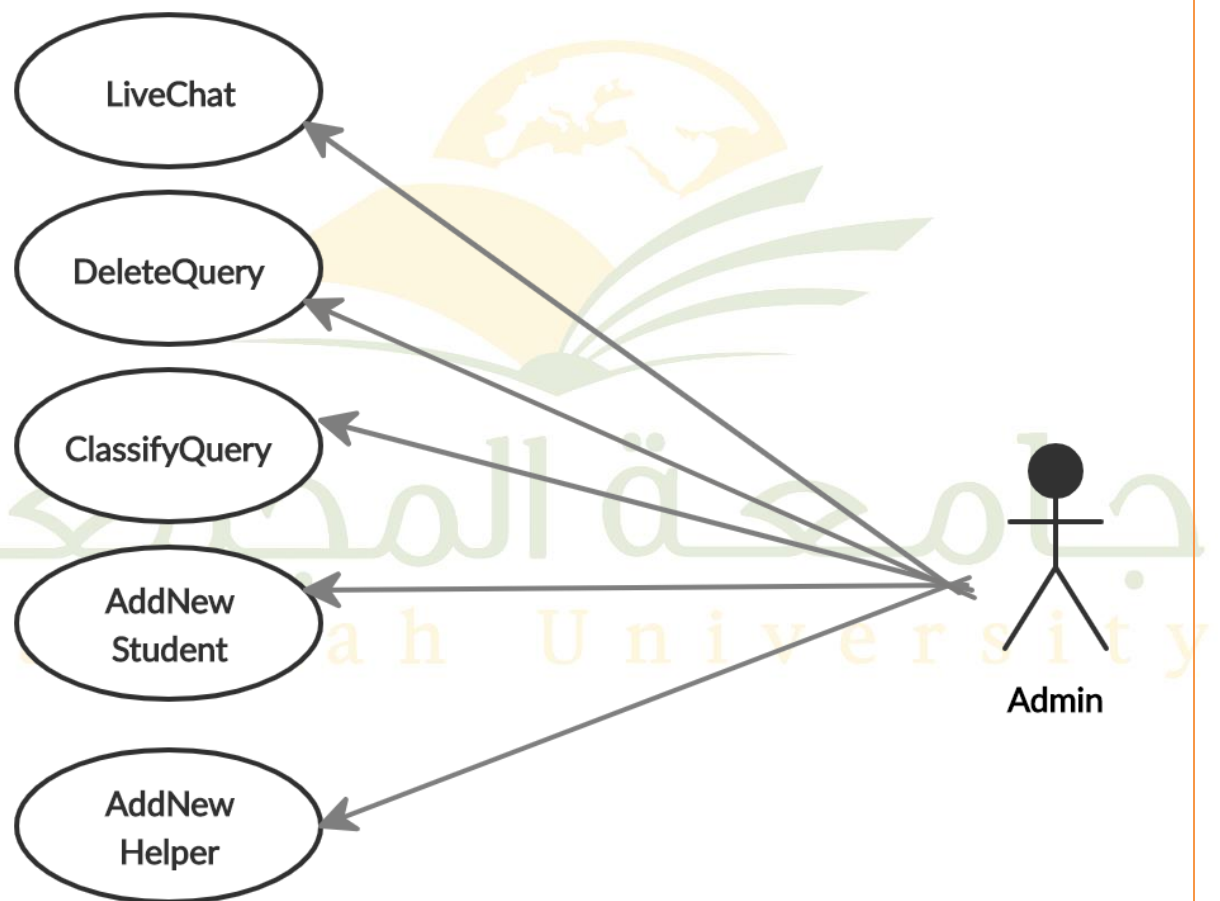
Fig 3.3: Use case diagram for student
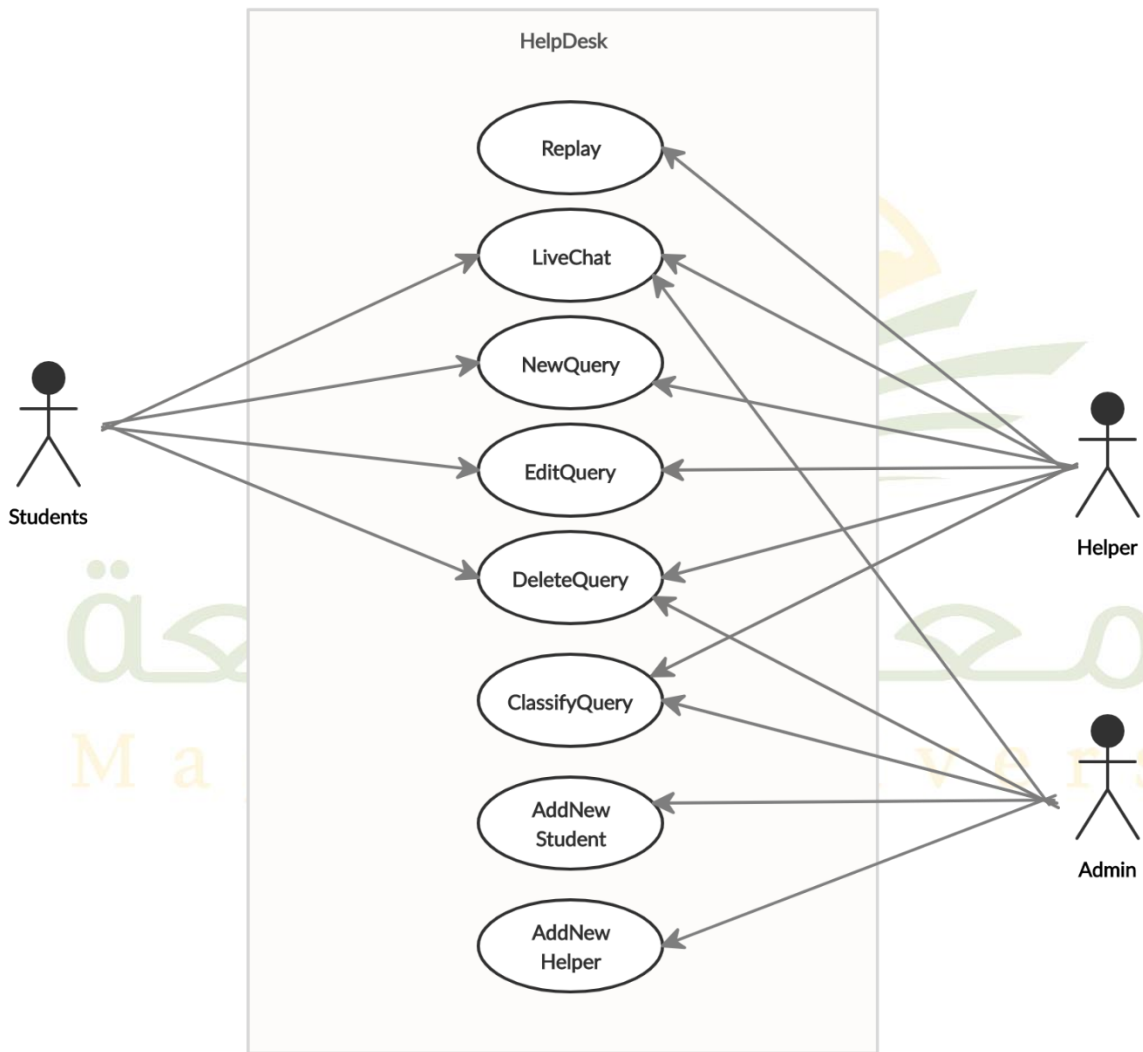
Fig 3.4: Use case diagram for student

Sequences Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessman and software developers to document and understand requirements for new and existing systems.
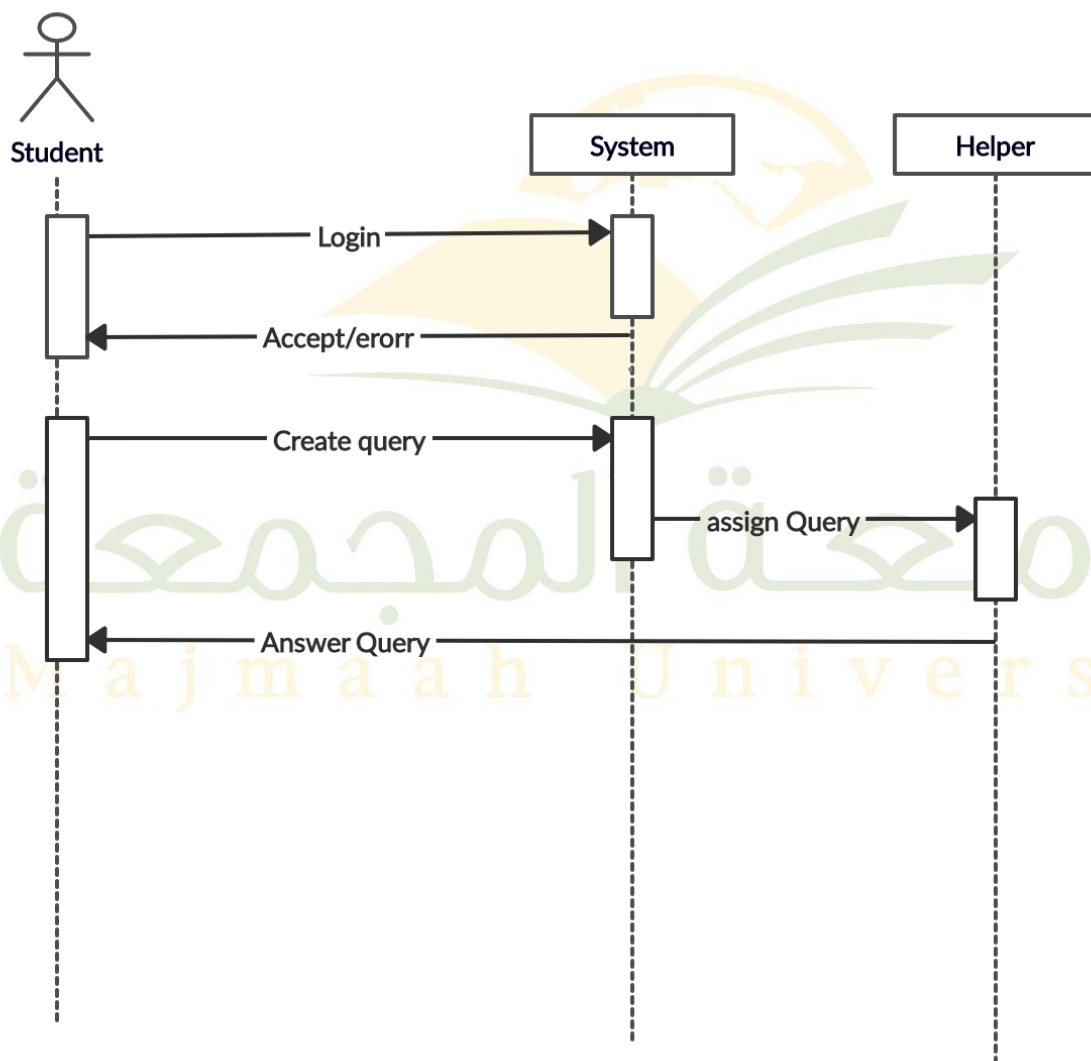
Fig 4.1: Sequence diagram for Student
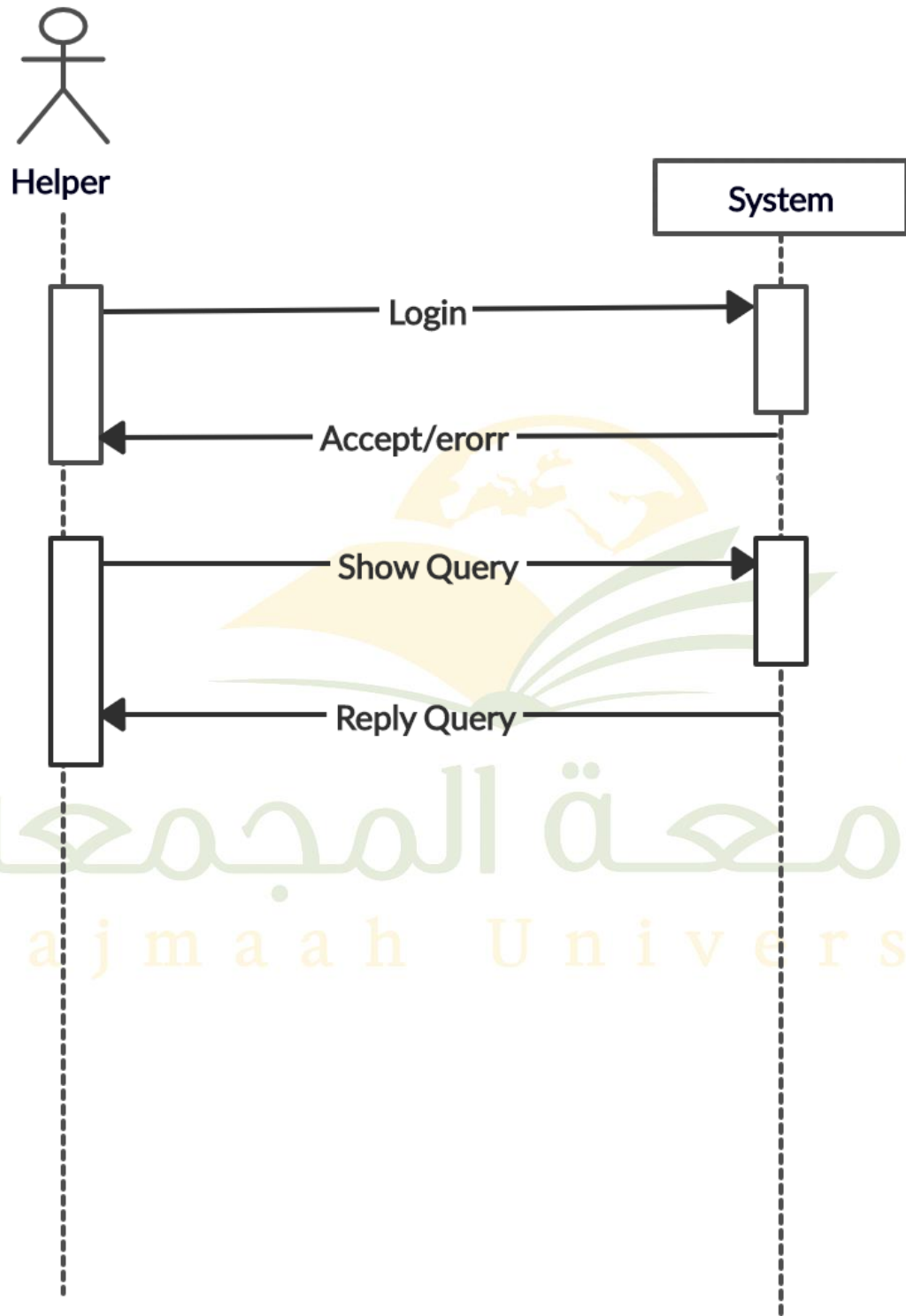
3.1 Fig 4.2: Sequence diagram for Helper
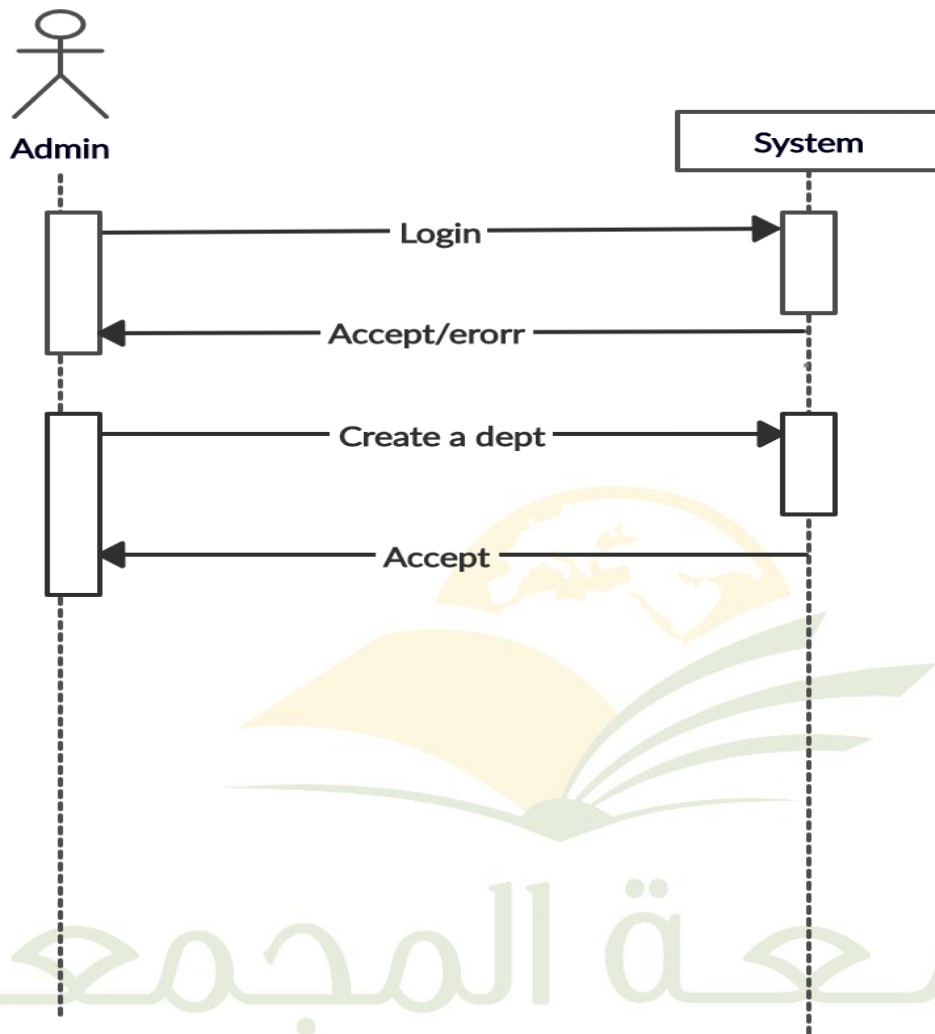
3.1   Fig 4.3: Sequence diagram for Admin

ER Diagram for Database Design

Entity Relationship Diagram, also known as ERD, ER Diagram or ER model, is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information: the major entities within the system scope, and the inter-relationships among these entities.
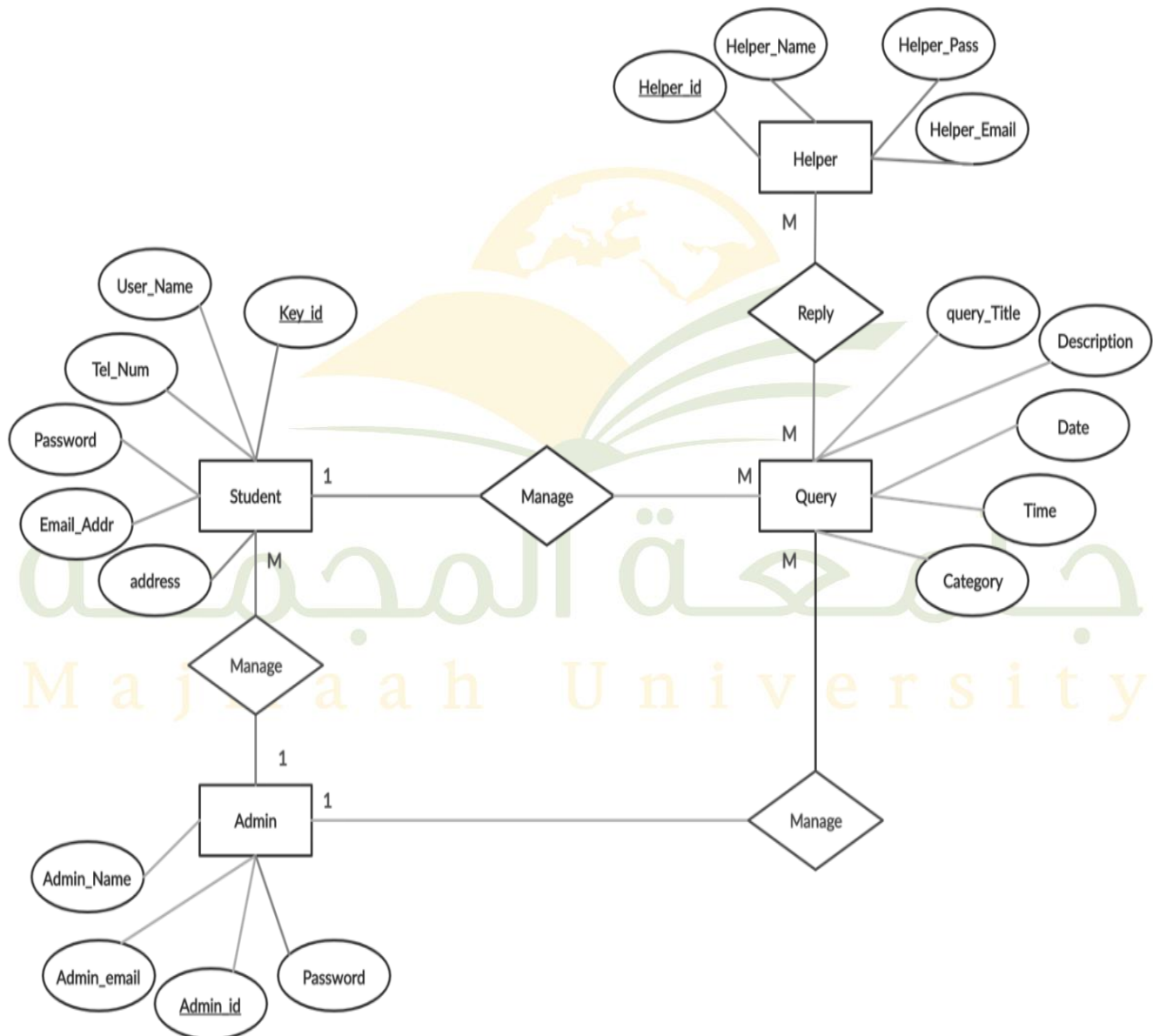
Fig 5: E-R diagram

❖ **ER for Student**
   The first entity is the student where he has the primary key, "key_id", user name, phone number, password, email and address where he has a "1to M" , manage relationship with query where the student can create more than one query and has another relationship call manage with an admin.

❖ **ER for Query**
   The second entity is a query that has more than one attribute, which is the title of the query, description, date, time and category.
   It has a relationship with the helper "M to M' where the helper can reply to more than one query and also has a manage relationship with the admin "M to 1".

❖ **ER for Admin**
   The third entity admin for more than one attribute is the name of the admin and email and ID and password.
   He has two relationships the call a manage, the first with the student is "1 to M", where the admin can manage more than the student accounts.
   The second relationship the call a manage, which is "1 to M", is with the query and can manage more than one query.

❖ **ER for Helper**
   The fourth entity a helper has more than one attribute, which is the primary key, password and email address.
   It has to do with the name of the reply with a query "M to M".
   More than one Helper can reply to queries.
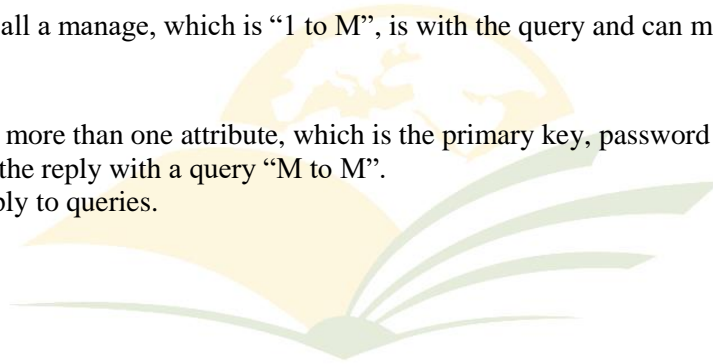
CHAPTER 3: SYSTEM DESIGN

3.1.1   Model Interfaces

❖ **The main interface**

The main interface is located in the left menus of Home, query and about and live chat and logout of the program.

On the right is Home containing all the queries.
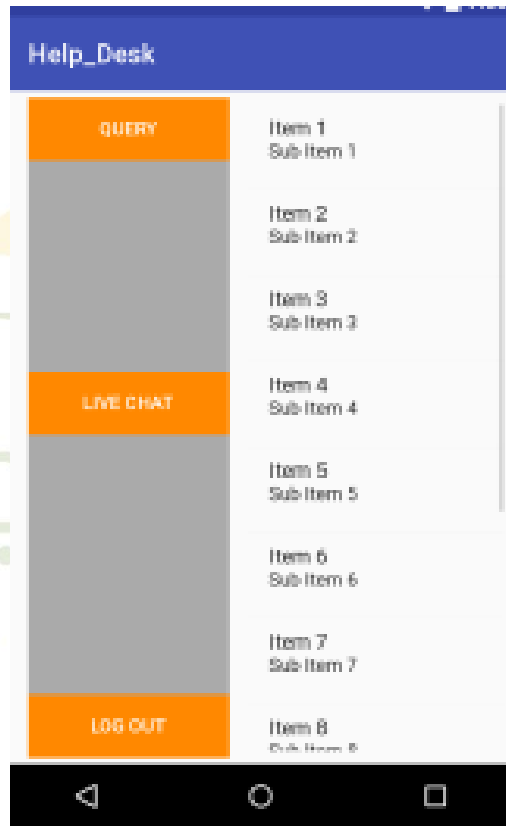
Figure 0-1 Main interface

Login Interface

This is the login interface where you can log in as a student, helper or administrator.
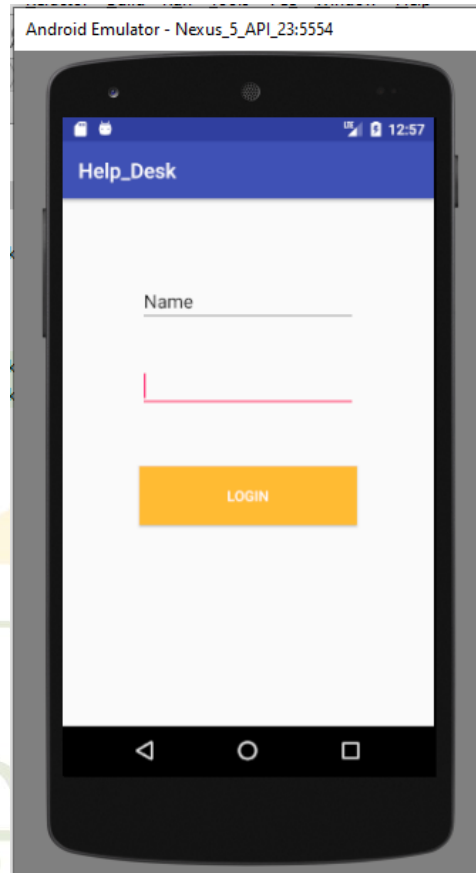
Figure 0-2 Login interface

Register Interface

The register interface in the program has several requirements where it is necessary to enter the full name,

e-mail address, mobile number, user name and password in order to be able to register in the program.
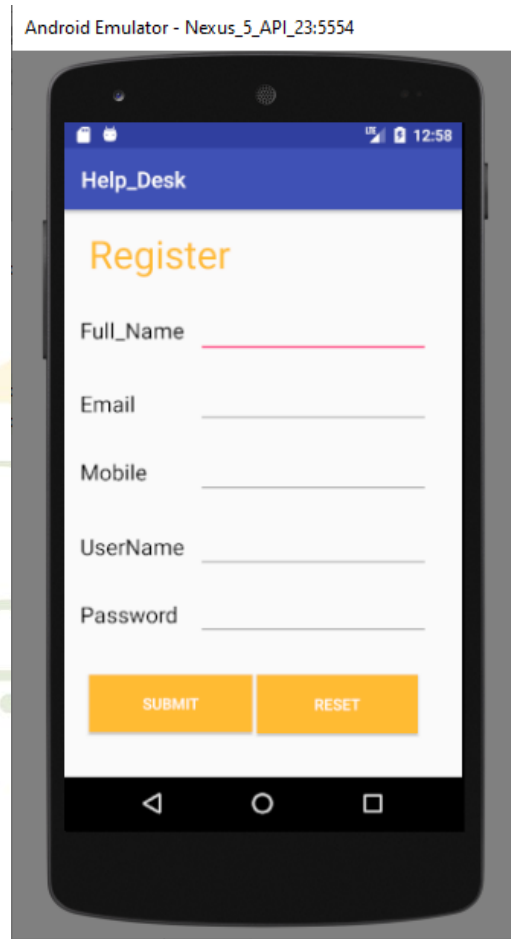
Figure 0-3 *Register* interface

❖ **Query Interface**

Here is the query interface where the queries are created and the query title is placed, description, date,

time and category.
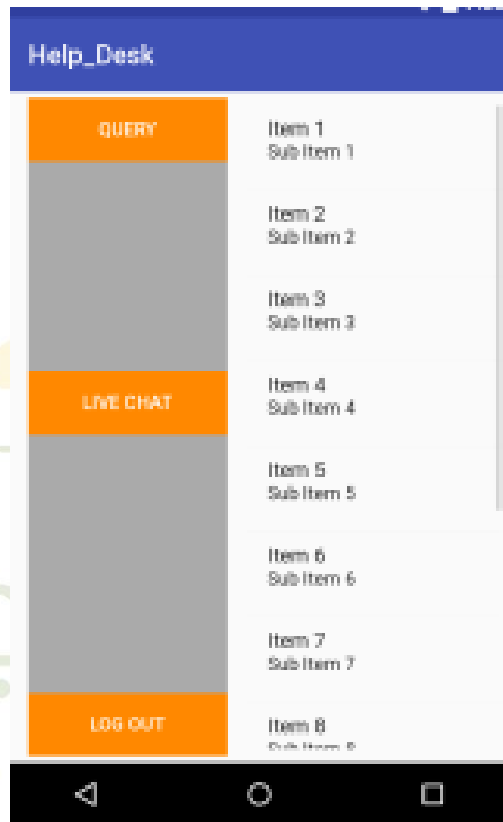
Figure 0-4 *Query* interface

Chapter 4:
Implementation and Testing

Introduction

This project is to design and implement an android mobile application, Called "Help Desk", to answer any questions and queries, easily and rapidly, for students at college of science. Queries on timetable, student's affairs, activities, and exams will be provided by the system. This application finds links between students and their colleage.

Procedures

we developed the application using android studio and used Xampp to make a local server that help us to upload our database
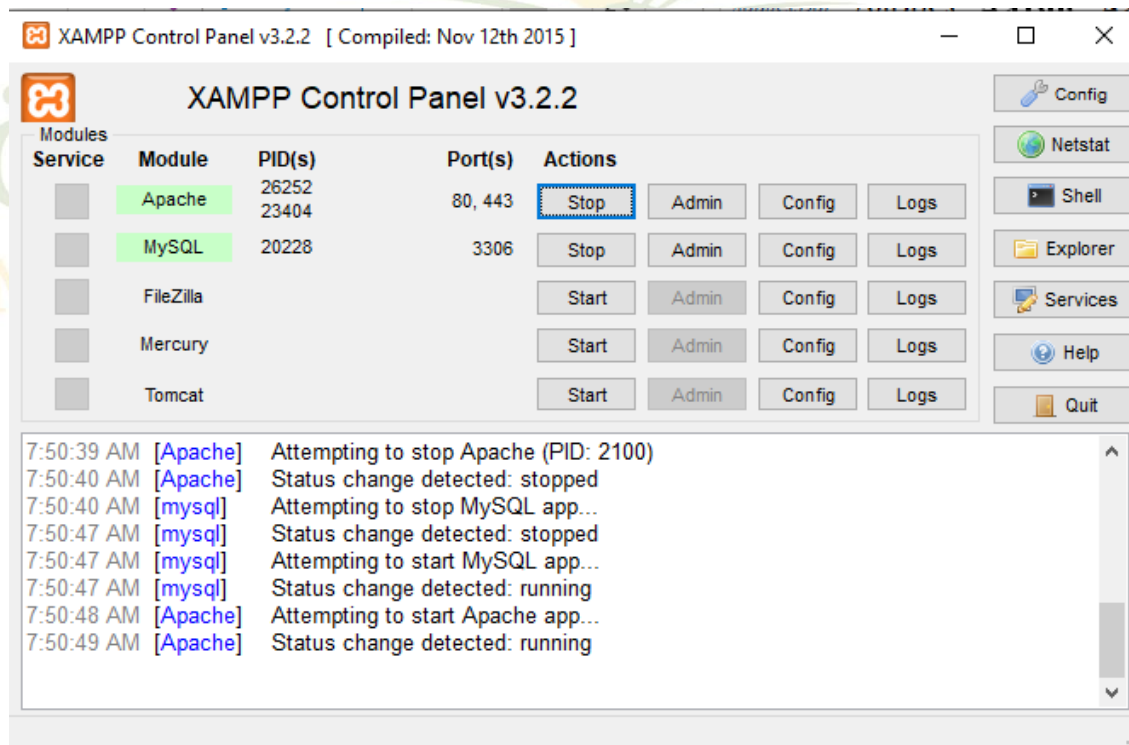
And then we upload our database as shown :
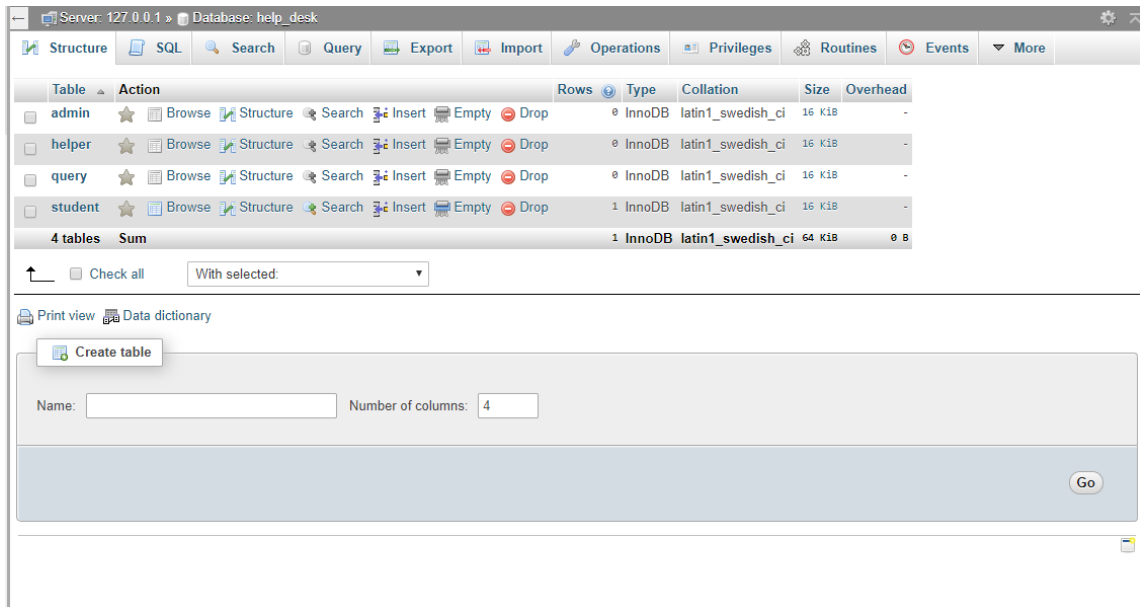
And we can open and use our application successfully as shown.
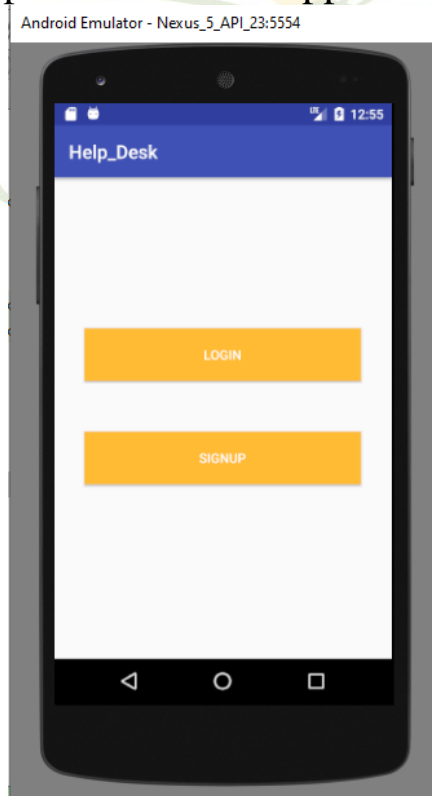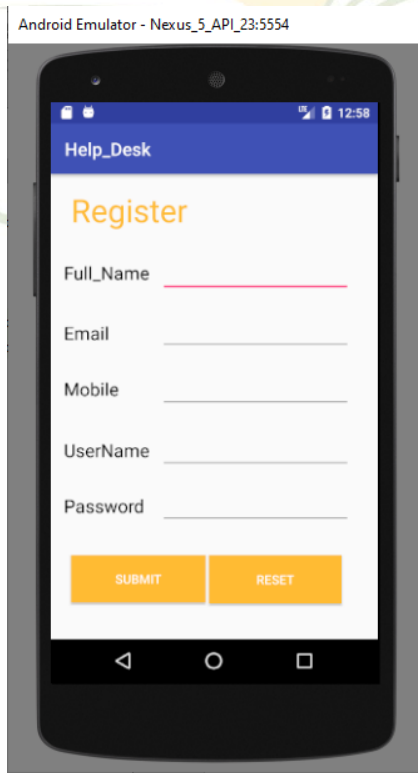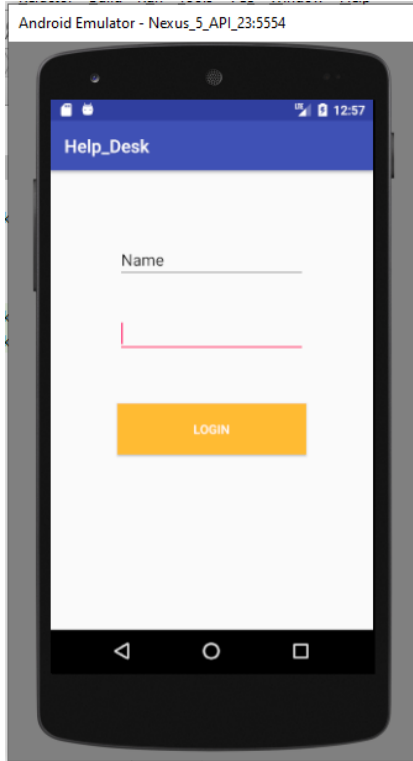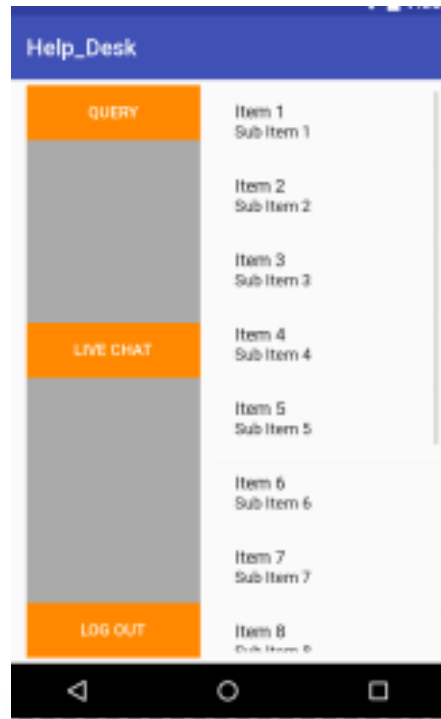
SQL database code :

```
-- phpMyAdmin SQL Dump
-- version 4.5.1
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: Apr 11, 2020 at 02:44 PM
-- Server version: 10.1.16-MariaDB
-- PHP Version: 5.6.24

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;


--
-- Database: `help_desk`
--

-- --------------------------------------------------------

--
-- Table structure for table `admin`
--

CREATE TABLE `admin` (
  `Admin_Name` text NOT NULL,
  `Admin_email` text NOT NULL,
  `Admin_id` int(11) NOT NULL,
  `Password` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `admin`
--

INSERT INTO `admin` (`Admin_Name`, `Admin_email`, `Admin_id`, `Password`) VALUES
('admin', 'admin@admin.com', 1, '123');

-- --------------------------------------------------------

--
```

```sql
-- Table structure for table `helper`
--

CREATE TABLE `helper` (
  `Helper_id` int(11) NOT NULL,
  `Helper_Name` text NOT NULL,
  `Helper_Pass` text NOT NULL,
  `Helper_Email` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `query`
--

CREATE TABLE `query` (
  `query_Title` text NOT NULL,
  `Description` text NOT NULL,
  `Date` date NOT NULL,
  `Time` time NOT NULL,
  `Category` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `query`
--

INSERT INTO `query` (`query_Title`, `Description`, `Date`, `Time`, `Category`) VALUES
('help to study', 'the best way to study is watching th lectures.', '2020-04-15', '06:17:00', 'general');

-- --------------------------------------------------------

--
-- Table structure for table `student`
--

CREATE TABLE `student` (
  `Key_id` int(11) NOT NULL,
  `User_Name` text NOT NULL,
  `Tel_Num` text NOT NULL,
  `Password` text NOT NULL,
  `Email_Addr` text NOT NULL,
  `address` text NOT NULL,
  `name` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `student`
--
```

33

```
INSERT INTO `student` (`Key_id`, `User_Name`, `Tel_Num`, `Password`, `Email_Addr`, `address`,
`name`) VALUES
(1, 'user1', '000000000000', '123', 'user@user.com', '1 st on', ''),
(2, 'kokpl', '222563', '456z', 'plpl@gmail.com', '', 'lplp'),
(3, 'kokpl', '222563', '456z', 'plpl@gmail.com', '', 'lplp');


--
-- Indexes for dumped tables
--


--
-- Indexes for table `admin`
--
ALTER TABLE `admin`
  ADD PRIMARY KEY (`Admin_id`);


--
-- Indexes for table `helper`
--
ALTER TABLE `helper`
  ADD PRIMARY KEY (`Helper_id`);


--
-- Indexes for table `student`
--
ALTER TABLE `student`
  ADD PRIMARY KEY (`Key_id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `admin`
--
ALTER TABLE `admin`
  MODIFY `Admin_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
--
-- AUTO_INCREMENT for table `helper`
--
ALTER TABLE `helper`
  MODIFY `Helper_id` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `student`
--
ALTER TABLE `student`
  MODIFY `Key_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Php code to connect the android application to the database :

```php
<?php

  include 'config.inc.php';

    // Check whether username or password is set from android
        // Innitialize Variable
        $result='';


        // Query database for row exist or not
    $sql = 'SELECT query_Title, Description FROM query';
    $stmt = $conn->prepare($sql);
    $stmt->execute();
    if($stmt->rowCount())
    {
            while ($row = $stmt->fetch()) {
 $result = $row['query_Title'].",".$row['Description'].",";
        }
    }

    elseif(!$stmt->rowCount())
    {
                $result="false";
    }

        // send result back to android
        echo $result;
```

```php
?>


<?php

   include 'config.inc.php';

      // Check whether username or password is set from android

         // Innitialize Variable
         $result='';
         $username = $_POST['name'];
      $password = $_POST['password'];
      $mobile = $_POST['mobile'];
      $email = $_POST['email'];
      $user = $_POST['username'];

         // Query database for row exist or not
      $sql = "INSERT INTO student (name, Password, Email_Addr,
      Tel_Num, User_Name) VALUES (?,?,?,?,?)";
      $stmt= $conn->prepare($sql);
      $stmt->execute([$username, $password, $email, $mobile,
$user]);

         // send result back to android
         echo "done";


   ?>


   <?php
```

```php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "help_desk";

try {
    $conn = new
    PDO("mysql:host=$servername;dbname=$dbname",
    $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE,
    PDO::ERRMODE_EXCEPTION);
}
catch(PDOException $e)
{
    die("OOPs something went wrong");
}

?>
```

Example of xml code to design an interface :

```xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.app.help_desk.Main3Activity">

    <TextView
        android:id="@+id/textView6"
        android:layout_width="98dp"
        android:layout_height="45dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:text="Password"
        android:textColor="@android:color/background_dark"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.029"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
```

```xml
                app:layout_constraintVertical_bias="0.762" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="98dp"
        android:layout_height="45dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:text="UserName"
        android:textColor="@android:color/background_dark"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.029"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.626" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="98dp"
        android:layout_height="45dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:text="Mobile"
        android:textColor="@android:color/background_dark"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.029"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.477" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="98dp"
        android:layout_height="45dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:text="Email"
        android:textColor="@android:color/background_dark"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.029"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.34" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="151dp"
        android:layout_height="53dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
```

```xml
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:text="Register"
        android:textColor="@android:color/holo_orange_light"
        android:textSize="35sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.078"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.018" />

    <EditText
        android:id="@+id/editText3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.843"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.174" />

    <EditText
        android:id="@+id/editText4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.843"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.316" />

    <EditText
        android:id="@+id/editText5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.843"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.457" />
```

```xml
<EditText
    android:id="@+id/editText6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.843"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.606" />

<EditText
    android:id="@+id/editText7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPassword"
    android:layout_marginRight="8dp"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginLeft="8dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    android:layout_marginBottom="8dp"
    app:layout_constraintHorizontal_bias="0.843"
    app:layout_constraintVertical_bias="0.741" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="98dp"
    android:layout_height="45dp"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:text="Full_Name"
    android:textColor="@android:color/background_dark"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.029"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.193" />

<Button
    android:id="@+id/button4"
    android:layout_width="148dp"
    android:layout_height="52dp"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"
    android:background="@android:color/holo_orange_light"
```

```xml
        android:onClick="signup"
        android:text="Submit"
        android:textColor="@android:color/background_light"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.077"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.927" />

    <Button
        android:id="@+id/button5"
        android:layout_width="146dp"
        android:layout_height="53dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="8dp"
        android:background="@android:color/holo_orange_light"
        android:text="Reset"
        android:textColor="@android:color/background_light"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.842"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.929" />
</android.support.constraint.ConstraintLayout>
```

Java sample of codes :

```java
package com.example.app.help_desk;

import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.Arrays;
```

```java
import java.util.List;


public class Main3Activity extends AppCompatActivity {
    public static final int CONNECTION_TIMEOUT=10000;
    public static final int READ_TIMEOUT=15000;
    List<String> items;
    List<String> items2;
    String myname;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main3);
    }

    void signup(View view)
    {
        Button buttonname = (Button)findViewById(R.id.button);
        EditText name = (EditText)findViewById(R.id.editText3);
        EditText email = (EditText)findViewById(R.id.editText4);
        EditText mobile = (EditText)findViewById(R.id.editText5);
        EditText user = (EditText)findViewById(R.id.editText6);
        EditText pass = (EditText)findViewById(R.id.editText7);

        myname = name.getText().toString();

        String first = name.getText().toString() + "," + pass.getText().toString()
                + "," + mobile.getText().toString();
        String second = email.getText().toString() +","+ user.getText().toString();
        Toast.makeText(Main3Activity.this, first, Toast.LENGTH_LONG).show();
        new Login().execute(first,second);

    }

    private class Login extends AsyncTask<String, String, String>
    {
        ProgressDialog pdLoading = new ProgressDialog(Main3Activity.this);
        HttpURLConnection conn;
        URL url = null;

        @Override
        protected void onPreExecute() {
            super.onPreExecute();

            pdLoading.setMessage("\tLoading...");
            pdLoading.setCancelable(false);
            pdLoading.show();

        }
        @Override
        protected String doInBackground(String... params) {
            try {

                url = new URL("http://localhost/app/insert.inc.php");

            } catch (MalformedURLException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
                return "exception";
            }
```

```java
            try {
                conn = (HttpURLConnection)url.openConnection();
                conn.setReadTimeout(READ_TIMEOUT);
                conn.setConnectTimeout(CONNECTION_TIMEOUT);
                conn.setRequestMethod("POST");

                conn.setDoInput(true);
                conn.setDoOutput(true);

                items = Arrays.asList(params[0].split(","));
                items2 = Arrays.asList(params[1].split(","));


                Uri.Builder builder = new Uri.Builder()
                        .appendQueryParameter("name", items.get(0))
                        .appendQueryParameter("password", items.get(1))
                        .appendQueryParameter("mobile", items.get(2))
                        .appendQueryParameter("email", items2.get(0))
                        .appendQueryParameter("username", items2.get(1));
                String query = builder.build().getEncodedQuery();

                OutputStream os = conn.getOutputStream();
                BufferedWriter writer = new BufferedWriter(
                        new OutputStreamWriter(os, "UTF-8"));
                writer.write(query);
                writer.flush();
                writer.close();
                os.close();
                conn.connect();

            } catch (IOException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
                return "exception";
            }

            try {

                int response_code = conn.getResponseCode();

                if (response_code == HttpURLConnection.HTTP_OK) {

                    InputStream input = conn.getInputStream();
                    BufferedReader reader = new BufferedReader(new InputStreamReader(input));
                    StringBuilder result = new StringBuilder();
                    String line;

                    while ((line = reader.readLine()) != null) {
                        result.append(line);
                    }

                    return(result.toString());


                }else{

                    return("unsuccessful");
                }

            } catch (IOException e) {
                e.printStackTrace();
                return "exception";
```

```java
            } finally {
                conn.disconnect();
            }


        }

        @Override
        protected void onPostExecute(String result) {

            Intent intent = new Intent(getBaseContext(), Main4Activity.class);
            intent.putExtra("name", myname);
            startActivity(intent);

            pdLoading.dismiss();

            Toast.makeText(Main3Activity.this, items.get(0), Toast.LENGTH_LONG).show();

        }

    }
}




package com.example.app.help_desk;

import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

public class Main2Activity extends AppCompatActivity {

    IOException es;
    public static final int CONNECTION_TIMEOUT=10000;
    public static final int READ_TIMEOUT=15000;
    private EditText username;
    private EditText Password;
    String name;

    @Override
```

```java
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);

    username = (EditText) findViewById(R.id.email);
    Password = (EditText) findViewById(R.id.password);
}

public void checkLogin(View arg0) {

    final String name = username.getText().toString();
    final String password = Password.getText().toString();

    new Login().execute(name,password);

}

private class Login extends AsyncTask<String, String, String>
{
    ProgressDialog Loading = new ProgressDialog(Main2Activity.this);
    HttpURLConnection conn;
    URL url = null;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();

        Loading.setMessage("\tLoading...");
        Loading.setCancelable(false);
        Loading.show();

    }
    @Override
    protected String doInBackground(String... params) {
        try {

            url = new URL("http://localhost/app/login.inc.php");

        } catch (MalformedURLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            return "exception";
        }
        try {
            conn = (HttpURLConnection)url.openConnection();
            conn.setReadTimeout(READ_TIMEOUT);
            conn.setConnectTimeout(CONNECTION_TIMEOUT);
            conn.setRequestMethod("POST");

            conn.setDoInput(true);
            conn.setDoOutput(true);

            name = params[0];

            Uri.Builder builder = new Uri.Builder()
                    .appendQueryParameter("username", params[0])
                    .appendQueryParameter("password", params[1]);
            String query = builder.build().getEncodedQuery();

            OutputStream os = conn.getOutputStream();
            BufferedWriter writer = new BufferedWriter(
                    new OutputStreamWriter(os, "UTF-8"));
```

45

```java
            writer.write(query);
            writer.flush();
            writer.close();
            os.close();
            conn.connect();

        } catch (IOException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
            es = e1;
            return "exception1";
        }

        try {

            int response_code = conn.getResponseCode();

            if (response_code == HttpURLConnection.HTTP_OK) {

                InputStream input = conn.getInputStream();
                BufferedReader reader = new BufferedReader(new InputStreamReader(input));
                StringBuilder result = new StringBuilder();
                String line;

                while ((line = reader.readLine()) != null) {
                    result.append(line);
                }

                return(result.toString());

            }else{

                return("unsuccessful");
            }

        } catch (IOException e) {
            e.printStackTrace();
            return "exception";

        } finally {
            conn.disconnect();
        }


    }

    @Override
    protected void onPostExecute(String result) {


        Loading.dismiss();

        if(result.equalsIgnoreCase("true"))
        {
            Intent intent = new Intent(getBaseContext(), Main4Activity.class);
            intent.putExtra("name", name);
            startActivity(intent);
            Main2Activity.this.finish();

        }else if (result.equalsIgnoreCase("false")){

            Toast.makeText(Main2Activity.this, "Invalid email or password",
```

46

```java
Toast.LENGTH_LONG).show();

        } else if (result.equalsIgnoreCase("exception")) {

            Toast.makeText(Main2Activity.this, "Connection Problem.", Toast.LENGTH_LONG).show();

        }
        else if (result.equalsIgnoreCase("exception1")) {

            Toast.makeText(Main2Activity.this, es.toString(), Toast.LENGTH_LONG).show();

        }
    }

  }
}
```

Chapter 5:

Conclusion and Future Work

In this report, we discuss the idea of Application for helpdesk. We explained the problems of the project and determined the solution. We draw diagrams to illustrate the stages of the work. Our feature work is to implement our application using suitable tools and in the future our application is need to be upgrade by more powerful functions and more features.

References

https://en.wikipedia.org/wiki/Sequence_diagram
https://en.wikipedia.org/wiki/Use_case
https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/
https://creately.com/