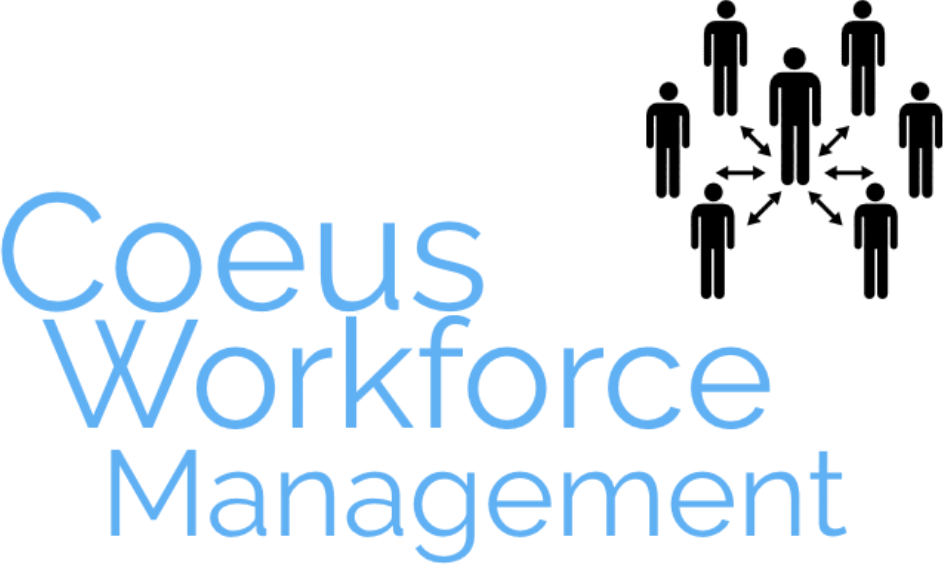
פרויקט גמר הנדסאי תוכנה



**מגישים**: פבל פיש מן

אלכסנדר בורלקוב

**קורס**:­­­­­­­­­­­­­­­הנדסאי תוכנה

**מנחה**: מיכאל חרסונסקי**­­­­­­­­­­­**

**נושא:** Coeus Workforce Management,

תוכנה לניהול כוח אדם וחישובי שכר.

**מוגש ע"י:** אלכסנדר בורלקוב

פבל פישמן

**קורס:** ­­­­­­­­­­­­­­ הנדסאי תוכנה

**מנחה:** מיכאל חרסונסקי**­­­­­­­­­­­**

**חתימת מנח :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ראש מגמה:** איגור ברגמן

**חתימת ראש המגמה :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

סימוכין : כלל 2

הצהרת סטודנט

שם הסטודנט: **פבל פישמן** ת.ז .: **307661652**

שם המכללה בה לומד הסטודנט : מכללה טכנולוגית באר – שבע קדימה מדע (ע"ר)

אני הח"מ, מצהיר בזאת כי פרויקט הגמר וספר הפרויקט המצ"ב נעשו על ידי בלבד.

פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.

פרויקט הגמר וספר הפרויקט נעשו על בסיס הנחייתו של המנחה האישי.

מקורות המידע בהם השתמשתי לביצוע פרויקט הגמר מצוינים ברשימת המקורות המצוינים בספר הפרויקט.

אני מודע לאחריות שהנני מקבל על עצמי על ידי חתימתי על הצהרה זו שכל הנאמר בה אמת ורק אמת.

חתימת הסטודנט: תאריך :

אישור המנחה האישי

הריני מאשר שהפרויקט בוצע בהנחייתי, קראתי את ספר הפרויקט ומצאתי כי הוא

מוכן לצורך הגשת הסטודנט להגנה על פרויקט הגמר.

שם המנחה: חתימה: תאריך

אישור ראש המגמה

הריני מאשר שספר הפרויקט מוכן לצורך הגשת הסטודנט להגנה על פרויקט הגמר

שם ראש המגמה: חתימה : תאריך:

סימוכין : כלל 2

הצהרת סטודנט

שם הסטודנט: **אלכסנדר בורלקוב** ת.ז .: **317538296**

שם המכללה בה לומד הסטודנט : מכללה טכנולוגית באר – שבע קדימה מדע (ע"ר)

אני הח"מ, מצהיר בזאת כי פרויקט הגמר וספר הפרויקט המצ"ב נעשו על ידי בלבד.

פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.

פרויקט הגמר וספר הפרויקט נעשו על בסיס הנחייתו של המנחה האישי.

מקורות המידע בהם השתמשתי לביצוע פרויקט הגמר מצוינים ברשימת המקורות המצוינים בספר הפרויקט.

אני מודע לאחריות שהנני מקבל על עצמי על ידי חתימתי על הצהרה זו שכל הנאמר בה אמת ורק אמת.

חתימת הסטודנט: תאריך :

אישור המנחה האישי

הריני מאשר שהפרויקט בוצע בהנחייתי, קראתי את ספר הפרויקט ומצאתי כי הוא

מוכן לצורך הגשת הסטודנט להגנה על פרויקט הגמר.

שם המנחה: חתימה: תאריך

אישור ראש המגמה

הריני מאשר שספר הפרויקט מוכן לצורך הגשת הסטודנט להגנה על פרויקט הגמר

שם ראש המגמה: חתימה : תאריך:

**תוכן עניינים**

[מבוא 6](#_Toc502244563)

[מדריך למתכנת 7](#_Toc502244564)

[מחקלה DataHandler 8](#_Toc502244565)

[מחלקה LoginForm 14](#_Toc502244566)

[מחלקה MainMenu 14](#_Toc502244567)

[מחלקה AddEmployee 18](#_Toc502244568)

[מחקלה EditEmployee 22](#_Toc502244569)

[מחלקה EditEmployeeSubForm 24](#_Toc502244570)

[מחלקה HourReport 29](#_Toc502244571)

[מחלקה HourReportSheet 30](#_Toc502244572)

[מחלקה Salary 32](#_Toc502244573)

[מחלקה SalarySheet 32](#_Toc502244574)

[מחלקה UserProfile 32](#_Toc502244575)

## מבוא

**מה זה Coues Workforce Management (CWM)?**

* CWM הינה תוכנה לניהול כוח אדם אשר תאפשר בקלות להוסיף עובדים לממסד נתונים, להדפיס דוחות, לחשב משכורות, לקבוע שעות עבודה, חופשות וכ"ו.
* **הסיבה לכתיבת פרויקט**

לאחר בדיקה מעמיקה של המצב הקיים כיום בשוק, מצאנו שלמרות שקיימות מספר תוכנות המציאות שירותים דומים, הן, בעיקרן ,מיועדות לעסקים גדולים ולא נותנות מענה לבעלי עסקים קטנים, שלא תמיד צריכים את כל האופציות שהתוכניות הנ"ל מציעות, אך נאלצים לשלם ממיטב כספם עבור אפשרויות שלעולם לא ישתמשו בהן.

בעיה נוספת שניסינו לפתור נעוצה בעובדה שכל התוכניות שמצאנו דרשו ידע רב בתחום עולם המחשוב, נושא שעלול להרתיע אנשים מבוגרים או כאלה שאין להם רקע במדעי מחשב. על מנת לעודד אותם לעבור לניהול ממוחשב ולא להסתמך על נייר ועט, ייצרנו ממשק משתמש פשוט וידידותי.

לסיכום, המטרה העיקרית שלנו היא ליצור תוכנה עבור מנהלי עסקים קטנים, עם ממשק משתמש ידידותי גם עבור אנשים ללא ידע במחשבים

**עבודה על הפרויקט**

התוכנה פותחה בשפת C# בסביבת NET.

במהלך העבודה על הפרויקט נתקלנו במספר בעיות:

* זמן תגובה איטי של שרת (המכיל את בסיס נתונים) בעקבות פניות רבות בעת ובעונה אחת.
* יצירת תוכנה קלה להבנה עבור משתמשים מתחילים מצד אחד ומתוחכמת מספיק כדי שתוכל להוות תחליף הולם עבור משתמשים מנוסים יותר.
* ALEX!!! WRITE SOME BS ABOUT SECURITY YOU IMPLEMENTED!!

## מדריך למתכנת

## מחקלה DataHandler

using MySql.Data.MySqlClient;//You need to have MySQL.Data installed via NuGet.

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GUI

{

class DataHandler

{

#region MySQL Connection String

private static string dbHost = "vps.vashbaldeus.pw";

private static string dbUsername = "hr";

private static string dbPassword = "3d1\*M7dS4&Gy5f";

/\*private static string dbHost = "localhost";//kept here just in case there's internet issues to easily switch to local mysql server;

private static string dbUsername = "root";

private static string dbPassword = "password";\*/

private static string dbName = "hr";

private static string dbString = $"Server={dbHost}; database={dbName}; UID={dbUsername}; password={dbPassword}; charset=hebrew;";

MySqlConnection connection = new MySqlConnection(dbString);

#endregion

DataTable dt = new DataTable();

#region MySQL Query Processing

private void ImportTable(string query)//function recieves a query string using that to pull data into a DataTable;

{

dt.PrimaryKey = null;

dt.Rows.Clear();

dt.Columns.Clear();

using (var da = new MySqlDataAdapter(new MySqlCommand(query, connection)))

{

da.FillSchema(dt, SchemaType.Source);

da.Fill(dt);

}

}

public DataTable GetTable(string query)//function returns a DataTable full of data that was pulled using the query;

{

ImportTable(query);

return dt.Copy();

}

public void ExecuteServerQuery(string query)//function recieves a string containing a query, and performs the query to the server;

{

using (MySqlConnection connection = new MySqlConnection(dbString))

{

MySqlCommand cmd = new MySqlCommand();

cmd = connection.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = query;

connection.Open();

cmd.ExecuteNonQuery();

connection.Close();

cmd.Dispose();

}

}

public void ExecuteServerQuery(string query, params object[] parameters)//function recieves a string containing a query, and performs the query to the server;

{

using (MySqlConnection connection = new MySqlConnection(dbString))

{

MySqlCommand cmd = new MySqlCommand();

cmd = connection.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = query;

for (var i = 0; i != parameters.Length; ++i)

cmd.Parameters.AddWithValue(i.ToString(), parameters[i]);

connection.Open();

cmd.ExecuteNonQuery();

connection.Close();

cmd.Dispose();

}

}

#endregion

#region DateTime

public string GetCurTime()//returns current time in the following format HH:mm:ss i.e. 14:13:45;

{

return DateTime.Now.ToString("HH:mm:ss", System.Globalization.DateTimeFormatInfo.InvariantInfo);

}

public string GetCurDate()//returns current date in the following format yyyy-MM-dd i.e. 2017-12-31;

{

return DateTime.Now.ToString("yyyy-MM-dd");

}

#endregion

public string Hash512(string password)//recieves a string, encrypts it in SHA512 and returns it back;

{

var bytes = new UTF8Encoding().GetBytes(password);

byte[] hashBytes;

using (var algorithm = new System.Security.Cryptography.SHA512Managed())

{

hashBytes = algorithm.ComputeHash(bytes);

}

return Convert.ToBase64String(hashBytes);

}

#region LoginAuth

public bool IsHRMember(string name)//checks whether a use is member of Human Resources;

{

foreach (DataRow dr in GetTable($"SELECT id, ishr FROM employees WHERE id='{name}'").Rows)

{

if (dr["id"].ToString() == name && dr["ishr"].ToString() == "1")

return true;

else return false;

}

return false;

}

public bool UserLoginPermissionChk(string username, string password)//recieves user data to check whether he has a permission to login into sytstem;

{

foreach (DataRow dr in GetTable($"SELECT id, userlogin FROM employees WHERE id='{username}'").Rows)

{

if (dr["id"].ToString() == username && dr["userlogin"].ToString() == "1")

return true;

else return false;

}

return false;

}

public bool LoginAuthentication(string username, string password)//checks user credentials in order to approve a login or deny it;

{

foreach (DataRow dr in GetTable($"SELECT id, passwd FROM employees WHERE id='{username}'").Rows)//LoadEmployees().Rows)

{

if (dr["id"].ToString() == username && dr["passwd"].ToString() == password)

return true;

else return false;

}

return false;

}

#endregion

#region TimeKeeper & Data Verification

public bool ChkEID(string id)//verifies employee id exists in the database;

{

foreach (DataRow dr in GetTable($"SELECT eid FROM employees WHERE eid='{id}'").Rows)

{

if (dr["eid"].ToString() == id)

return true;

else return false;

}

return false;

}

public bool ChkTKEnter(string id)//checks employee enter time, to prevent duplicates;

{

DateTime now = DateTime.Now;

foreach(DataRow dr in GetTable($"SELECT eid, enter\_time FROM reports WHERE eid='{id}'").Rows)

{

if (dr["eid"].ToString() == id)

{

return true;

}

else return false;

}

return false;

}

public bool ChkTKExit(string id)//checks empoloyee exit time to prevent duplicates;

{

return false;

}

public int GetDepartmentCode(string username)//pulls department code in order to allow a user who is not HR to view only his department employees;

{

foreach (DataRow dr in GetTable($"SELECT id, dcode FROM employees WHERE id={username}").Rows)

if (dr["id"].ToString() == username)

return int.Parse(dr["dcode"].ToString());

else return -1;

return -1;

}

public bool CheckDuplicateDepartment(string dname)//checks whether there is a duplicate department;

{

using (DataTable d = GetTable($"SELECT dname FROM departments WHERE dname='{dname}'"))

{

if (d.Rows != null)

foreach (DataRow dr in d.Rows)

{

if (dr["dname"].ToString() == dname)

return true;

}

else return false;

}

return false;

}

#endregion

#region Permissions

public bool GetPermAdd()//performs a check if user has AddEmployee permissions;

{

foreach(DataRow dr in GetTable($"SELECT id, addp FROM employees WHERE id={Properties.Settings.Default.username}").Rows)

{

if (dr["id"].ToString() == Properties.Settings.Default.username && dr["addp"].ToString() == "1")

return true;

else return false;

}

return false;

}

public bool GetPermAlter()//performs a check if user has EditEmployee permissions;

{

foreach (DataRow dr in GetTable($"SELECT id, alterp FROM employees WHERE id={Properties.Settings.Default.username}").Rows)

{

if (dr["alterp"].ToString() == "1")

return true;

else return false;

}

return false;

}

public bool GetPermHours()//performs a check if user has Hour Reports permissions;

{

foreach (DataRow dr in GetTable($"SELECT id, hours FROM employees WHERE id={Properties.Settings.Default.username}").Rows)

{

if (dr["id"].ToString() == Properties.Settings.Default.username && dr["hours"].ToString() == "1")

return true;

else return false;

}

return false;

}

public bool GetPermUserPrpfile()//performs a check if user has permissions to change his own password;

{

foreach (DataRow dr in GetTable($"SELECT id, userprofile FROM employees WHERE id={Properties.Settings.Default.username}").Rows)

{

if (dr["id"].ToString() == Properties.Settings.Default.username && dr["userprofile"].ToString() == "1")

return true;

else return false;

}

return false;

}

#endregion

}

}

## מחלקה LoginForm

## מחלקה MainMenu

using GUI.Menus;

using System;

using System.Data;

using System.Windows.Forms;

namespace GUI

{

public partial class MainMenu : Form

{

DataHandler dh = new DataHandler();

public DataTable cityCodes = new DataTable();

public DataTable countryCodes = new DataTable();

public DataTable bcountryCodes = new DataTable();

public DataTable deptCodes = new DataTable();

public MainMenu()

{

InitializeComponent();

#region Menu Tab Selection Check

tabControl1.TabPages.Remove(tabPage1);//removal of both tabs;

tabControl1.TabPages.Remove(tabPage2);

if (dh.IsHRMember(Properties.Settings.Default.username))//checks if user Human Resources member;

tabControl1.TabPages.Add(tabPage1);//makes tab 1 visible

else tabControl1.TabPages.Add(tabPage2);//makes tab 2 visible according to result;

#endregion

LoadDataTables();

}

private void MainMenu\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

#region DataTable Load and Get Functions

private void LoadDataTables()

{

//calls DataHandler class, pulling data according to query;

//using this data later in order to load menu forms faster;

cityCodes = dh.GetTable("SELECT city\_name FROM city\_codes");

countryCodes = dh.GetTable("SELECT country\_name FROM country\_codes");

bcountryCodes = dh.GetTable("SELECT country\_name FROM country\_codes");

deptCodes = dh.GetTable("SELECT dname FROM departments");

}

public DataTable GetCity()

{

return cityCodes;

}

public DataTable GetCountry()

{

return countryCodes;

}

public DataTable GetBCountry()

{

return bcountryCodes;

}

public DataTable GetDepartments()

{

return deptCodes;

}

#endregion

private void buttonAddEmployee\_Click(object sender, EventArgs e)

{

try

{

if (dh.GetPermAdd() != true)//checks if user has Add permission;

throw new Exception("אין לך הרשאה להוסיף עובדים");

using (var addemp = new AddEmployee())//opens AddEmployee form;

addemp.ShowDialog();

}

catch (Exception err)

{

MessageBox.Show(err.Message, "שגיאה", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonEditEmployee\_Click(object sender, EventArgs e)

{

try

{

if (dh.GetPermAlter() != true)//checks if user has Alter permission;

throw new Exception("אין לך הרשאה לערוך עובדים");

using (EditEmployee ee = new EditEmployee())//opens EditEmployee form;

ee.ShowDialog();

}

catch (Exception err)

{

MessageBox.Show(err.Message, "שגיאה", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonHourReport\_Click(object sender, EventArgs e)

{

try

{

/\*if (dh.GetPermHours() != true)//checks if user has Hours permission;

throw new Exception("אין לך הרשאה לערוך עובדים");\*/

using (HourReport hr = new HourReport())//opens HourReport form;

hr.ShowDialog();

}

catch (Exception err)

{

MessageBox.Show(err.Message, "שגיאה", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonDepartments\_Click(object sender, EventArgs e)

{

try

{

/\*if (dh.IsHRMember(Properties.Settings.Default.username) != true)//checks if user member of Human Resources;

throw new Exception("אין לך הרשאה לערוך עובדים");\*/

using (Departments dpt = new Departments())//opens Departments form;

dpt.ShowDialog();

deptCodes.Clear();

deptCodes = dh.GetTable("SELECT dname FROM departments");

}

catch (Exception err)

{

MessageBox.Show(err.Message, "שגיאה", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonSalary\_Click(object sender, EventArgs e)

{

try

{

using (Salary sal = new Salary())

sal.ShowDialog();

}

catch (Exception err)

{

MessageBox.Show(err.Message, "שגיאה", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonUserProfile\_Click(object sender, EventArgs e)

{

try

{

if (dh.GetPermUserPrpfile() != true)//check if user can change his own password;

throw new Exception("אין לך הרשאה לערוך את הפרופיל משתמש שלך");

using (UserProfile up = new UserProfile())//open UserProfile form;

up.ShowDialog();

}

catch (Exception err)

{

MessageBox.Show(err.Message, "שגיאה", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

## מחלקה AddEmployee

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GUI.Menus

{

public partial class AddEmployee : Form

{

MainMenu main = new MainMenu();

DataHandler dh = new DataHandler();

public AddEmployee()

{

InitializeComponent();

LoadComboBox();

}

private void LoadComboBox()//function loads data from database and fills the comboBox's;

{

comboBoxCityCode.DisplayMember = "city\_name";

comboBoxCityCode.ValueMember = "city\_name";

comboBoxCityCode.DataSource = main.GetCity();//pulls CityCode DataTable from MainMenu form;

comboBoxCityCode.SelectedIndex = -1;

comboBoxCountry.DisplayMember = "country\_name";

comboBoxCountry.ValueMember = "country\_name";

comboBoxCountry.DataSource = main.GetCountry();//pulls CountryCode DataTable from MainMenu form;

comboBoxCountry.SelectedIndex = -1;

comboBoxCOB.DisplayMember = "country\_name";

comboBoxCOB.ValueMember = "country\_name";

comboBoxCOB.DataSource = main.GetCountry();//pulls CountryCode DataTable from MainMenu form;

comboBoxCOB.SelectedIndex = -1;

comboBoxDept.DisplayMember = "dname";

comboBoxDept.ValueMember = "dname";

comboBoxDept.DataSource = main.GetDepartments();//pulls DepartmentCode DataTable from MainMenu form;

comboBoxDept.SelectedIndex = -1;

}

private void buttonAddEmployee\_Click(object sender, EventArgs e)

{

try

{

if (this.Controls.OfType<TextBox>().Any(t => string.IsNullOrEmpty(t.Text)) || this.Controls.OfType<ComboBox>().Any(t => string.IsNullOrEmpty(t.Text)))

throw new Exception("השארת שדות ריקים"); //condition checks whether the form fields are not left empty, if true sends a pop up message;

foreach (DataRow dr in dh.GetTable("select id,fname,lname from employees").Rows)

if (dr["id"].ToString() == textBoxID.Text && dr["fname"].ToString() == textBoxFName.Text && dr["lname"].ToString() == textBoxLName.Text)

throw new Exception("עובד תחת השם הזה קיים במסד נתונים");//function checks whether employee under said ID, Name, or Lastname exists in the database.

#region Permissions Settings

int[] perms = new int[6];

for (int i = 0; i < 6; i++)

perms[i] = 0;//initiates the perms array with 0 values;

foreach (int index in checkedListBoxUserPerms.CheckedIndices)

{//checks each selection, assigns proper digit on 'perms' array;

switch (index)

{

case 0://חלק מכח אדם

perms[0]++;

break;

case 1://יכול לשנות סיסמה

perms[1]++;

break;

case 2://יכול להתחבר למערכת

perms[2]++;

break;

case 3://יכול להוסיף עובדים

perms[3]++;

break;

case 4://יכול לעשות שינויים לעובדים קיימים

perms[4]++;

break;

case 5://יכול לצפות בדו"ח שעות

perms[5]++;

break;

}

}

#endregion

//dh.ExecuteServerQuery - pushes a query via DataHandler class;

dh.ExecuteServerQuery("INSERT INTO employees(id,passwd,userlogin,userprofile,ishr,addp,alterp,hours,fname,lname,gender,dob,address,zip,city,country,cob,mdate,married,children,sdate,job\_type,wage\_class,dcode) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)",

long.Parse(textBoxID.Text), dh.Hash512(textBoxPASSWD.Text), perms[2], perms[1], perms[0], perms[3], perms[4], perms[5],

textBoxFName.Text, textBoxLName.Text, comboBoxGender.Text, dateTimePickerDOB.Value.Date.ToString("yyyy-MM-dd"),

textBoxStreet.Text, textBoxZIP.Text, comboBoxCityCode.SelectedIndex + 1, comboBoxCountry.SelectedIndex + 1, comboBoxCOB.SelectedIndex + 1, dateTimePickerMigDate.Value.Date.ToString("yyyy-MM-dd"),

comboBoxMarital.SelectedIndex, comboBoxChildren.SelectedIndex, dateTimePickerJStart.Value.Date.ToString("yyyy-MM-dd"), comboBoxJType.SelectedIndex, comboBoxSalary.SelectedIndex, comboBoxDept.SelectedIndex+1);

DialogResult result = MessageBox.Show("הוספת עובד התבצעה בהצלחה, מעוניין להוסיף עוד עובד?", "אישור", MessageBoxButtons.YesNo);

if (result == DialogResult.Yes)

ResetForm();//resets the form if user selected Yes on MessageBox;

else this.Close();

}

catch (Exception err)

{

MessageBox.Show(err.Message, "שגיאה", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void buttonResetForm\_Click(object sender, EventArgs e)

{

ResetForm();//calls the function that clears the form fields;

}

private void textBoxID\_KeyPress(object sender, KeyPressEventArgs e)//function verifies that only digits are allowed in textBoxID field;

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && (e.KeyChar != '.'))

{//makes sure that textBoxID field can not receive nothing except digits;

e.Handled = true;

}

}

private void ResetForm()//function resets WinForm Text fields;

{

textBoxFName.Clear();

textBoxLName.Clear();

textBoxID.Clear();

textBoxStreet.Clear();

textBoxZIP.Clear();

comboBoxGender.ResetText();

dateTimePickerDOB.ResetText();

dateTimePickerJStart.ResetText();

dateTimePickerMigDate.ResetText();

comboBoxChildren.SelectedIndex = -1;

comboBoxCityCode.SelectedIndex = -1;

comboBoxCOB.SelectedIndex = -1;

comboBoxCountry.SelectedIndex = -1;

comboBoxDept.SelectedIndex = -1;

comboBoxGender.SelectedIndex = -1;

comboBoxJType.SelectedIndex = -1;

comboBoxMarital.SelectedIndex = -1;

comboBoxSalary.SelectedIndex = -1;

textBoxPASSWD.Clear();

for (int i = 0; i < checkedListBoxUserPerms.Items.Count; i++)

checkedListBoxUserPerms.SetItemChecked(i, false);

}

private void textBoxPASSWD\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && !char.IsLetter(e.KeyChar) && char.IsPunctuation(e.KeyChar))

{//function watches textBoxPASSWD field for non-english characters;

e.Handled = true;

return;

}

e.Handled = false;

return;

}

private void AddEmployee\_Load(object sender, EventArgs e)

{

}

}

}

## מחקלה EditEmployee

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GUI.Menus

{

public partial class EditEmployee : Form

{

DataHandler dh = new DataHandler();

public EditEmployee()

{

InitializeComponent();

}

private void buttonSearch\_Click(object sender, EventArgs e)

{

try

{

Properties.Settings.Default.temp = textBoxSearchField.Text;//saves textBoxSearchField to later use in EditEmployeeSubForm;

using (DataTable dtEditEmployee = dh.GetTable($"SELECT \* FROM employees WHERE eid={textBoxSearchField.Text} OR id={textBoxSearchField.Text}"))

{

if (dtEditEmployee != null)//makes sure DataTable is not empty;

{

if (dtEditEmployee.Rows.Count > 0)//makes sure DataTable has Rows;

{

using (EditEmployeeSubForm eesf = new EditEmployeeSubForm())

eesf.ShowDialog();//opens the EditEmployeeSubForm in order to load the employee data for editing;

}

}

else throw new Exception("מספר עובד או תעודת זהות שגויים");

}

}

catch (Exception err)

{

MessageBox.Show(err.Message, "שגיאה", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void textBoxSearchField\_KeyPress(object sender, KeyPressEventArgs e)

{//function keeps track that only digits are types into the search field;

if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && (e.KeyChar != '.'))

{

e.Handled = true;

}

}

}

}

## מחלקה EditEmployeeSubForm

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GUI.Menus

{

public partial class EditEmployeeSubForm : Form

{

MainMenu main = new MainMenu();

DataHandler dh = new DataHandler();

public EditEmployeeSubForm()

{

InitializeComponent();

LoadComboBox();

LoadFormData();

if (dh.IsHRMember(Properties.Settings.Default.username) != true)

{//checks whether user is part of Human Resources or not, if false then asigns controls to read-only mode;

checkedListBoxPermissions.SelectionMode = SelectionMode.None;

textBoxPASSWD.ReadOnly = true;

}

}

private void LoadComboBox()//calls for data from MainMenu WinForm in order to fill the comboBox fields;

{

comboBoxCity.DisplayMember = "city\_name";

comboBoxCity.ValueMember = "city\_name";

comboBoxCity.DataSource = main.GetCity();//pulls CityCode DataTable from MainMenu form;

comboBoxCountry.DisplayMember = "country\_name";

comboBoxCountry.ValueMember = "country\_name";

comboBoxCountry.DataSource = main.GetCountry();//pulls CountryCode DataTable from MainMenu form;

comboBoxCOB.DisplayMember = "country\_name";

comboBoxCOB.ValueMember = "country\_name";

comboBoxCOB.DataSource = main.GetBCountry();//pulls CountryCode DataTable from MainMenu form;

comboBoxDpt.DisplayMember = "dname";

comboBoxDpt.ValueMember = "dname";

comboBoxDpt.DataSource = main.GetDepartments();//pulls DepartmentCode DataTable from MainMenu form;

}

private void LoadFormData()

{

using(EditEmployee ee = new EditEmployee())

{

string query = "";

if (dh.IsHRMember(Properties.Settings.Default.username))//checks whether user is member of Human Resources, loading data accordingly;

query = $"SELECT \* FROM employees WHERE eid={Properties.Settings.Default.temp} OR id={Properties.Settings.Default.temp}";

else query = $"SELECT \* FROM employees WHERE (eid={Properties.Settings.Default.temp} OR id={Properties.Settings.Default.temp}) AND dcode={dh.GetDepartmentCode(Properties.Settings.Default.username)}";

using (DataTable dt = dh.GetTable(query))

{//calls on DataHandler class to execute the query and pull data from the SQL server.

foreach(DataRow dr in dt.Rows)

{

this.Text = $"עריכת עובד - {dr["fname"].ToString()} {dr["lname"].ToString()}";

textBoxEID.Text = dr["eid"].ToString();

textBoxID.Text = dr["id"].ToString();

textBoxFName.Text = dr["fname"].ToString();

textBoxLName.Text = dr["lname"].ToString();

comboBoxGender.Text = dr["gender"].ToString();

textBoxStreet.Text = dr["address"].ToString();

textBoxZIP.Text = dr["zip"].ToString();

comboBoxCity.SelectedIndex = int.Parse(dr["city"].ToString()) - 1;

comboBoxCountry.SelectedIndex = int.Parse(dr["country"].ToString()) - 1;

comboBoxCOB.SelectedIndex = int.Parse(dr["cob"].ToString()) - 1;

comboBoxDpt.SelectedIndex = int.Parse(dr["dcode"].ToString()) - 1;

comboBoxKids.SelectedIndex = int.Parse(dr["children"].ToString());

comboBoxMarital.SelectedIndex = int.Parse(dr["married"].ToString());

comboBoxJType.SelectedIndex = int.Parse(dr["job\_type"].ToString());

comboBoxSalary.SelectedIndex = int.Parse(dr["wage\_class"].ToString());

if (dr["ishr"].ToString() != "0")

checkedListBoxPermissions.SetItemChecked(0, true);

if(dr["userprofile"].ToString() != "0")

checkedListBoxPermissions.SetItemChecked(1, true);

if (dr["userlogin"].ToString() != "0")

checkedListBoxPermissions.SetItemChecked(2, true);

if (dr["addp"].ToString() != "0")

checkedListBoxPermissions.SetItemChecked(3, true);

if (dr["alterp"].ToString() != "0")

checkedListBoxPermissions.SetItemChecked(4, true);

if (dr["hours"].ToString() != "0")

checkedListBoxPermissions.SetItemChecked(5, true);

dateTimePickerDOB.Value = DateTime.Parse(dr["dob"].ToString());

dateTimePickerMigration.Value = DateTime.Parse(dr["mdate"].ToString());

dateTimePickerJStart.Value = DateTime.Parse(dr["sdate"].ToString());

if (dr["endate"].ToString() != "")

dateTimePickerJEnd.Value = DateTime.Parse(dr["endate"].ToString());

}

}

}

}

private void buttonUpdate\_Click(object sender, EventArgs e)

{

try

{

if (this.Controls.OfType<TextBox>().Where(t => t != textBoxPASSWD).Any(t => string.IsNullOrEmpty(t.Text)) || this.Controls.OfType<ComboBox>().Any(t => string.IsNullOrEmpty(t.Text)))

throw new Exception("השארת שדות ריקים"); //condition checks whether the form fields are not left empty, if true sends a pop up message;

foreach (DataRow dr in dh.GetTable("select eid,id,fname,lname from employees").Rows)

if (dr["id"].ToString() == textBoxID.Text && dr["fname"].ToString() == textBoxFName.Text && dr["lname"].ToString() == textBoxLName.Text && dr["eid"].ToString() != textBoxEID.Text)

throw new Exception("עובד תחת השם הזה קיים במסד נתונים");//function checks whether employee under said ID, Name, or Lastname exists in the database.

#region Permissions Settings

int[] perms = new int[6];

for (int i = 0; i < 6; i++)

perms[i] = 0;

foreach (int index in checkedListBoxPermissions.CheckedIndices)

{//checks each selection, assigns proper digit on 'perms' array;

switch (index)

{

case 0://חלק מכח אדם

perms[0]++;

break;

case 1://יכול לשנות סיסמה

perms[1]++;

break;

case 2://יכול להתחבר למערכת

perms[2]++;

break;

case 3://יכול להוסיף עובדים

perms[3]++;

break;

case 4://יכול לעשות שינויים לעובדים קיימים

perms[4]++;

break;

case 5://יכול לצפות בדו"ח שעות

perms[5]++;

break;

}

}

#endregion

if (textBoxPASSWD.Text != "")//checks if password field left empty or not and performs a query accordingly;

dh.ExecuteServerQuery("UPDATE employees SET id=?,passwd=?,userlogin=?,userprofile=?,ishr=?,addp=?,alterp=?,hours=?,fname=?,lname=?,gender=?,dob=?,address=?,zip=?,city=?,country=?,cob=?,mdate=?,married=?,children=?,sdate=?,job\_type=?,wage\_class=?,dcode=? WHERE eid=?",

long.Parse(textBoxID.Text), dh.Hash512(textBoxPASSWD.Text), perms[2], perms[1], perms[0], perms[3], perms[4], perms[5],

textBoxFName.Text, textBoxLName.Text, comboBoxGender.Text, dateTimePickerDOB.Value.Date.ToString("yyyy-MM-dd"),

textBoxStreet.Text, textBoxZIP.Text, comboBoxCity.SelectedIndex + 1, comboBoxCountry.SelectedIndex + 1, comboBoxCOB.SelectedIndex + 1,

dateTimePickerMigration.Value.Date.ToString("yyyy-MM-dd"), comboBoxMarital.SelectedIndex, comboBoxKids.SelectedIndex, dateTimePickerJStart.Value.Date.ToString("yyyy-MM-dd"),

comboBoxJType.SelectedIndex, comboBoxSalary.SelectedIndex, comboBoxDpt.SelectedIndex + 1, long.Parse(textBoxEID.Text));

else dh.ExecuteServerQuery("UPDATE employees SET id=?,userlogin=?,userprofile=?,ishr=?,addp=?,alterp=?,hours=?,fname=?,lname=?,gender=?,dob=?,address=?,zip=?,city=?,country=?,cob=?,mdate=?,married=?,children=?,sdate=?,job\_type=?,wage\_class=?,dcode=? WHERE eid=?",

long.Parse(textBoxID.Text), perms[2], perms[1], perms[0], perms[3], perms[4], perms[5],

textBoxFName.Text, textBoxLName.Text, comboBoxGender.Text, dateTimePickerDOB.Value.Date.ToString("yyyy-MM-dd"),

textBoxStreet.Text, textBoxZIP.Text, comboBoxCity.SelectedIndex + 1, comboBoxCountry.SelectedIndex + 1, comboBoxCOB.SelectedIndex + 1,

dateTimePickerMigration.Value.Date.ToString("yyyy-MM-dd"), comboBoxMarital.SelectedIndex, comboBoxKids.SelectedIndex, dateTimePickerJStart.Value.Date.ToString("yyyy-MM-dd"),

comboBoxJType.SelectedIndex, comboBoxSalary.SelectedIndex, comboBoxDpt.SelectedIndex + 1, long.Parse(textBoxEID.Text));

DialogResult result = MessageBox.Show("השינויים נשמרו בהצלחה.", "אישור", MessageBoxButtons.OK);

if (result == DialogResult.OK)

this.Close();

}

catch (Exception err)

{

MessageBox.Show(err.Message, "שגיאה", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

## מחלקה HourReport

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GUI.Menus

{

public partial class HourReport : Form

{

public HourReport()

{

InitializeComponent();

}

private void HourReport\_Load(object sender, EventArgs e)

{

}

private void buttonRunReport\_Click(object sender, EventArgs e)

{

try

{

Properties.Settings.Default.temp = $"eid={textBoxID.Text} OR id={textBoxID.Text}";//saves the search query parameters for the report;

using (HourReportSheet hrs = new HourReportSheet())

hrs.ShowDialog();

}

catch (Exception err)

{

MessageBox.Show(err.Message, "שגיאה", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

## מחלקה HourReportSheet

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GUI.Menus

{

public partial class HourReportSheet : Form

{

public HourReportSheet()

{

InitializeComponent();

}

private void HourReportSheet\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'hrDataSet.archived\_reports' table. You can move, or remove it, as needed.

this.archived\_reportsTableAdapter.Fill(this.hrDataSet.archived\_reports);

// TODO: This line of code loads data into the 'hrDataSet.employees' table. You can move, or remove it, as needed.

this.employeesTableAdapter.Fill(this.hrDataSet.employees);

// TODO: This line of code loads data into the 'hrDataSet.reports' table. You can move, or remove it, as needed.

this.reportsTableAdapter.Fill(this.hrDataSet.reports);

using (hrDataSetTableAdapters.reportsTableAdapter db = new hrDataSetTableAdapters.reportsTableAdapter())

{//pulls the data of the reports table from the database;

using (DataTable dt = db.GetData())

{

reportsBindingSource.DataSource = dt.Select(Properties.Settings.Default.temp).ToList();//filters the data according to previous sent query;

}

}

using (hrDataSetTableAdapters.employeesTableAdapter db = new hrDataSetTableAdapters.employeesTableAdapter())

{

using (DataTable dt = db.GetData())

{//pulls the data of the employees table from database;

employeesBindingSource.DataSource = dt.Select(Properties.Settings.Default.temp).ToList();//filters the data according to previous sent query;

}

}

this.reportViewer1.RefreshReport();//reload the report form with pulled data;

}

}

}

## מחלקה Salary

## מחלקה SalarySheet

## מחלקה UserProfile

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace GUI.Menus

{

public partial class UserProfile : Form

{

DataHandler dh = new DataHandler();

public UserProfile()

{

InitializeComponent();

textBoxCurPW.PasswordChar = '\*';//hides the text of the textBox;

textBoxNewPW.PasswordChar = '\*';

textBoxNewPWV.PasswordChar = '\*';

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

if (dh.LoginAuthentication(Properties.Settings.Default.username, dh.Hash512(textBoxCurPW.Text)) != true)

throw new Exception("סיסמה נוכחית שגוייה, נסה שוב.");//checks whether current password if correct before changing it;

if (textBoxNewPW.Text != textBoxNewPWV.Text)//checks new password fields are the same;

throw new Exception("סיסמאות אינן תואמות, נסה שוב.");

DialogResult result = MessageBox.Show("אתה בטוח שאתה רוצה לשנות סיסמה?", "אישור", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if(result == DialogResult.Yes)//uses a MessageBox to verify the user wants to replace their password before actually doing that;

{

dh.ExecuteServerQuery("UPDATE employees SET passwd=? WHERE id=?",dh.Hash512(textBoxNewPW.Text),Properties.Settings.Default.username);

textBoxCurPW.ResetText();

textBoxNewPW.ResetText();

textBoxNewPWV.ResetText();

MessageBox.Show("סיסמה הוחלפה בהצלחה", "אישור", MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Close();

}

}

catch (Exception err)

{

MessageBox.Show(err.Message, "שגיאה", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}