**בית אבות הקשיש**

**מגישה:** רות מונסונגו

**שם בית הספר:** סמינר בית יעקוב הרב מאיר

**מורה מלווה:** המו' מ.שור

**תאריך הגשה:** סיון תשפ"ב

**תוכן עניינים:**

מסמך ייזום:

* מבוא --------------------------------------------------------------------------------------------------------------------------------- 3
* מבנה וארכיטקטורה ---------------------------------------------------------------------------------------------------------------- 5
* תיחום המערכת --------------------------------------------------------------------------------------------------------------------- 6
* אילוצים ומגבלות -------------------------------------------------------------------------------------------------------------------- 8

ניתוח מצב קיים:

* תרשים תוכן -------------------------------------------------------------------------------------------------------------------------- 9
* תהליכים ------------------------------------------------------------------------------------------------------------------------------ 10
* בעיות בתהליכים --------------------------------------------------------------------------------------------------------------------- 13

מדריך למפתח:

* שכבת הנתונים – DAL --------------------------------------------------------------------------------------------------------------- 14
* שכבת הלוגיקה העסקית – BLL ----------------------------------------------------------------------------------------------------- 15
  + מחלקות רבים
  + מחלקות יחיד
* שכבת היישום – GUI ----------------------------------------------------------------------------------------------------------------- 29

מסמך הגדרת דרישות:

* ניתוח נתונים --------------------------------------------------------------------------------------------------------------------------- 66
* עיצוב - מדריך למשתמש ------------------------------------------------------------------------------------------------------------- 69
* רפלקציה ------------------------------------------------------------------------------------------------------------------------------- 76

**מסמך יזום**

מבוא:

נאמר "כבד את אביך ואת אימך למען יאריכון ימיך"(דברים ה' ט"ז) רצון בסיסי של בן אנוש הוא כמובן לחיות, ורצון בסיס אף יותר הוא שאותם החיים ימשכו ככול שניתן. בני אדם מוכנים לשלם כל הון שבעולם עבור רגע אחד מין המצרך הבסיסי הזה-חיים. אנשים מוכנים להתנדב להסתכן ולהתאמץ ובלבד שיזכו להציל חיים- חיי אדם. כל בר דעת מבין את ערך המושג שנקרא "חיי אדם". פעילים מוכנים להסתכן ולטוס מעבר לעולם, לארצות העולם השלישי, ארצות נחשלות, על מנת להגיש סיוע הומניטרי לילדים במטרה להציל את חייהם. משלחות חירום מכל העולם נשלחות במקרים של אסונות טבע ופגיעות אקלים, ערך החיים הוא עיניין גלובלי ורצון אנושי ומובן לחלוטין.

התורה מייחסת ערך עליון לעיניין החיים, האדם מצווה לשמור על גופו ועל חיו, "ונשמרתם מאוד לנפשותיכם", נסתם ולא בכדי... כל רגע של חיי יהודי בעולם הזה נחשב ונספר.

ואם וכאשר זוכה האדם ומגיע בכבוד לגיל הזיקנה אזי ערך חיו- עם כל הניסיון שצבר, מרקיע פי כמה וכמה. "כל הנוטל עיצה מין הזקנים אינו נכשל" זו השקפתה של התורה וזה ההיגיון הפשוט והבסיסי.

למרבה הצער, ישנה דעה, ונכון יותר לקרוא לה תופעה, שלמען האמת די תופסת תאוצה בשנים האחרונות שלא כדאי כל כך להלחם על חיי הזקנים. כך ירד באפן דרסטי ערך חיי הזקנים בעולם, פרשיות של הפרות זכיות אדם, התעללויות, הזנחה וכדו מתפרסמות כל הזמן. והמסכנה הכואבת היא שעולם הצעירים רואה בזקנים נטל כבד ותו לא וזה מחלחל ומשפיע על תודעת הדור הצעיר. ומכיוון שכך לפני כ30 שנה נהגה הרעיון של דיור מוגן, אך גם הוא לא ממש הניב תוצאות והתגלו מחדלים מזעזעים.

לאור הצורך הזועק בשוק והמצב המביש שהגיענו אליו כאנושות, (הבהרה חשובה: תיאור המקרים מדבר על הציבור הכללי בלבד) הגעתי למסכנה שיש לפעול בזריזות ובנחישות, ולכן הוחלט על הקמת המיזם "הקשיש" שהוא בעצם רשת של בתי אב המיועדים לזקנים שמכל סיבה שהיא לא ניתן לטפל בהם בבית, סיעודיים, חולים וכדו בית אבות זה שם דגש עליון על השגחה פרטנית על כל זקן וזקן מבני בתי האבות מפוארים וכוללים שלוש קומות בלבד של חדרי אישפוז וכל זקן הנקלט במוסד מתקבל לרופא אישי האחראי על הטיפול פה עד לעזיבתו לביתו או חלילה לבית אחר...

עבורנו, כבית אבות, חשובה מאוד הדאגה לאיכות חיים גבוהה בעבור המבוגרים הדרים אצלינו. בתי האבות שלנו מאוד יוקרתיים ושווי ערך למלונות חמישה כוכבים, רק שכאן קהל היעד מורכב מאוכלוסיית גיל הזהב. אם אפשר לפרוש לפנסיה ברמת חיים גבוהה- רצוי לעשות זאת אצלינו.

חשוב שתדעו שכאשר מדובר בבתי האבות של "הקשיש", חשבנו על כל פרט ופרט שינעים את שהותכם אצלינו. אם יש ברשותכם רכב או שמא אתם מעוניינים שהמשפחה תבוא לבקר אתכם, בבית האבות יעמידו לרשותכם חנייה מרווחת לכל שעות היממה, ניקיון, תחזוקה, כביסה, בית כנסת וחדר אוכל עם תפריט דיאטטי המותאם לכל אחד מהאנשים השוהים בבית האבות.  
  
אם תרצו לשתות קפה או לארח אנשים שלא בחדר האישפוז המעוצב שלכם, תוכלו ליהנות מישיבה נוחה בלובי המפואר של בית האבות. אם אתם מפחדים שהנכדים ישתעממו, אל דאגה, ברוב בתי האבות היוקרתיים שלנו ישנה פינת משחק לילדים.  
  
בבתי האבות שלנו מודעים לכך שתעסוקה מעניינת ומגווונת חשובה והכרחית לחיים תקינים ולכן מעמידים אנו בפניכם אולם תרבות, מופעים, הרצאות וספריה המכילה ספרים רבים מתחומים שונים.  
  
בתי האבות שלנו מעודדים את הדיירים המבוגרים להקפיד על פעילות ספורטיבית סדירה, תוכלו למצוא בבתי אבות היוקרתיים: חדר כושר חדיש עם כל המכשירים המשוכללים, התעמלות אירובית, בריכה [ושירותי פיזיותרפיה](https://www.d.co.il/h-c7400-e0-p0-l0-t40055024.na1). כמו כן בתי האבות יספקו לכם גם שירותי שמירה במשך כל שעות היממה.

בבתי האבות החדרים מעוצבים בטעם ענין וביתי כדי למנוע אוירה מדכאת וגנרית של מוסדות סיעוד וטיפוח דומים (כדוגמת בתי חולים מסויימים), בתי האבות מכילים חדרים פרטיים מרווחים ומאובזרים בלי כל פשרה על איכות ורמת השהות בהם.

אך, מעל הכול, הכי חשוב לנו זאת מטרת העל שלשמה יצאנו לפעולה- לדאוג לבריאותם ולאורך חייהם של המבוגרים השוהים שם. לכן בבתי האבות שלנו תיהיו תחת פיקוח רפואי של עשרים וארבע שעות ביממה. דבר זה יכלול צוות רפואי המכיל רופא אישי לכל דייר, שירותי אחיות רצופים, והספקת תרופות סדירה.

כמו כן, בית האבות דואג כמובן, ללחצן מצוקה לכל דייר, מטעמים בטיחותיים. כך יתאפשר מענה מידי במקרה של בעייה זו או אחרת, חלילה.

עד לכתיבת תכנה זו התנהלו בתי האבות באופן ידני עם רישום ידני מסורבל מה שהביא באופן בלתי נמנע טעויות בלבולים ושכחות, כיוון שבית אבות תרופות ומשמרות הם לא דברים של מה בכך וטעיות אלו עלולות לעלות בחיי אדם שאת חשיבותם כבר פרטנו קודם הוחלט להגדיר את התכנה הזאת שמתעדת פרטים חשובים אלו באופן הברור והמסודר ביותר.

הכניסה לתוכנה מסוווגת ע"פ תפקידו של המשתש, כניסה למנהל לרופאים ולאחים. הכניסה מותנית בהקשת מספר זהות שעל פיו מזהה התוכנה באיזה רופא\ אח מדובר ובהמצעותו מוודאת התוכנה שאין מדובר במתחזה. כאשר לצורך הדוגמא – יש באפשרותו לפתוח את מאגר ההדיירים הרופאים ואחים וכו ולהוסיף להוסיף רופאים, אחיות לעדכן את פרטיהם ולצפות ברשימות. כאשר דייר חדש מתווסף הוא נרשם ע"י המנהל. לצורך מניעת טעיות (מביכות☹), המנהל יוכל לפקח על המשמרות ועל התרופות ולמעשה יוכל לשלוט על המערכת כולה. כאשר, לצורך הדוגמא רופא נכנס- כמובן גם הוא ע"פ הקשת תעודת זהות, מקב את רשימת הזקנים המטופלים על ידו, ובלחיצה על אחד מהם הוא נכנס לתיקו האישי של המטופל ויוכל לצפות בנתוני התרופות ולעדכן את רישומן ע"פ שיקול הדעת שהערך ממצב המטופל. הרופא יכול ג"כ להתעדכן בכל ההסטוריה הרפואית של המטופל. לאחר רישות התרופה יופיע הרישום בדף האח שנכנס למשמרת בקומה בה מאושפז מטופל זה ויוכל לאשר את נתינת התרופה באותו יום. כמו כן יכול האח לצפות ברשימת המשמרות שבהן הוא משובץ.

מבנה וארכיטקטורה:

תוכנת **Nursing Home** היא תוכנה לניהול תרופות לזקנים בבית אבות המבצעת את התהליכים:

תהליך עם הרופא: רישום התרופות לזקנים באופן מסודר.

תהליך עם האח: רשימת נטילות של תרופות שעליו לבצע.

תהליך עם המנהל: ניהול הצוות והדיירים בבית האבות, ניהול והזמנת תרופות חסרות וכן ניהול משמרות לאחים בבית האבות.

התוכנה מיועדת לשימוש כתוכנת Desktop ונכתבה בWindows Forms בספת C# בסביבת Visual Studio 2019.

התוכנית נבנתה בצורת חשיבה של DNA- מודל שלושת השכבות.

התוכנית חולקה לשלוש שכבות כאשר כל שכבה מטפלת בתחום שונה ומכילה כמה מחלקות ו/או טפסים שונים. בתוך כל מחלקה הוגדרו משתנים ואובייקטים מתאימים המיוחדים למחלקה. כל מחלקה מטפלת בתחום מסוים ומתקשרת לטופס/ המחלקה המתאימים לה. כתיבה זו נועדה להקל על המתכנת בכתיבת הקוד והן בתחזוקה השוטפת של הקוד. כמובן שבין המחלקות והטפסים שבשכבות השונות קיים קשר, וקיימת התייחסות מתוך מחלקה אחת לעצמים המוגדרים במחלקות האחרות, אך הכל נעשה תוך שמירה על ביטחון ואבטחת הנתונים והפונקציות.

פרוט השכבות:

DAL Data Access Layer - שכבה מטפלת בקישור התוכנה עם בסיס הנתונים.

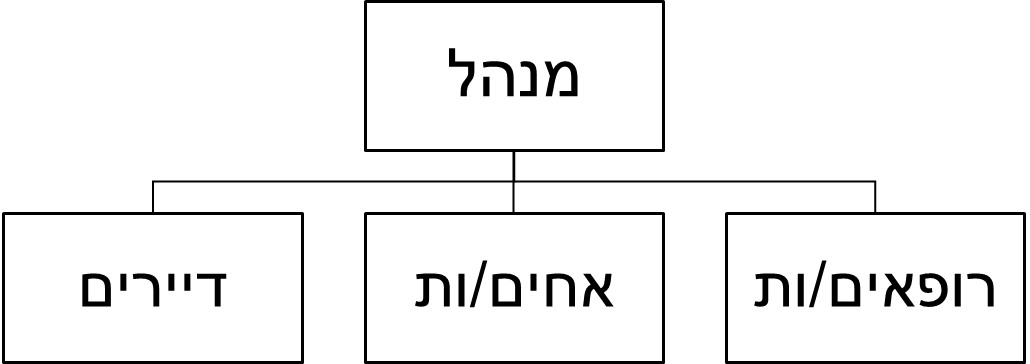
BLL Business Logic Layer – שכבה המטפלת בכל המחלקות המייצגות את המידע בפרויקט ופונקציות לוגיות שונות.

Graphic User Inerface GUI – שכבה המטפלת בממשק משתמש של התכנה (טפסים בWinForms ).

תיחום המערכת:

תרשים המבנה הארגוני

מנכ"ל בתי האבות



תיחום תהליכים:

* טיפול ברופאים
* טיפול באחים
* טיפול בדיירים
* טיפול בתרופות
* טיפול ברשימת תרופות חסירות
* טיפול במשמרות
* טיפול ברישום תרופות לזקנים
* טיפול בנטילת תרופה

גורמים מעורבים

|  |  |  |
| --- | --- | --- |
| תיאור: | רמת מעורבות: | שם הגורם: |
| טיפול ברופאים  טיפול באחים  טיפול בדיירים  טיפול בתרופות  טיפול ברשימת תרופות חסירות  טיפול במשמרות | משתמש ויווזם | מנהל |
| טיפול במשמרות  טיפול בנטילת תרופה | משתמש | אח |
| טיפול ברישום תרופות לזקנים | משתמש | רופא |
| טיפול ברישום תרופות לזקנים  טיפול בנטילת תרופה | לא משתמש | דיירים |

אילוצים ומגבלות:

**אילוצי כוח אדם:** המערכת מפותחת ע"י תלמידת תיכון המערכת צריכה להיות ידידותית ונוחה למשתמש.

עזרה ניתנת ע"י מורת בית הספר למחשבים המורה מ. שור

**אילוצי אופי המערכת:** המערכת צריכה להיות ידידותית, נוחה להפעלה, חסינה בפני שגיאות משתמש, בעלת מראה אסטטי ונקי וגמישה לשינויים עתידים.

**אילוצים טכנולוגים:** התוכנה מפותחת במערכת הפעלה windows 2010 ובסביבת עבודה . Microsoft Visual Studio 2019

**אילוצי אופי המערכת:** המערכת תפעל בכל ימות השבוע למעט שבתות וחגים שבהם תופעל ע"י נכרי.

**ניתוח מצב קיים**

תרשים תוכן:

פירוט הטיפולים הניתנים להם עכשיו

פרטי התרופות ורשימת התרופות החסרות

E3

**אח**

E2

**רופא**

**מנהל**

E1

**מנהל**

E1

**רופא**

E2

**אח**

E3

הוספה ועדכון של רופאים/ות אחים/יות דיירים ותרופות

מספר הקומה שבה הוא עובד היום

הוספת מחיקת ועידכון תרופות לזקנים

פרטי האחים/יות הרופאים/ות והדיירים/ות

התרופות שהוא צריך לתת

פרטי הזקנים שלו

עדכון הזמנת תרופות חסרות

משמרות לאחים

פרוט המשמרות שלו

פרטי הזקנים שלו

ההיסטוריה הרפואית שלהם

פרטי המשמרות

עץ תהליכים:

תיאור תהליכים:

**קוד תהליך:** 1P

**שם התהליך:** ניהול דיירים/ות

**תיאור התהליך:** טיפול בכל מה שקשור לרישום הדיירים

**התהליך כולל את הפעולות:**

* הוספת דייר
* מחיקת דייר
* עדכון פרטי הדיירים
* צפיה בנתוני הדיירים

**קוד תהליך:** 2P

**שם התהליך:** ניהול אחים/ות

**תיאור התהליך:** טיפול בכל מה שקשור לרישום האחים

**התהליך כולל את הפעולות:**

* הוספת אח/ות
* מחיקת אח/ות
* עדכון פרטי האחים
* צפיה בנתוני האחים

**קוד תהליך:** 3P

**שם התהליך:** ניהול רופאים/ות

**תיאור התהליך:** טיפול בכל מה שקשור לרישום הרופאים

**התהליך כולל את הפעולות:**

* הוספת רופא
* מחיקת רופא (יתאפשר רק אחרי שיוך של מטופליו לרופא אחר)
* עדכון פרטי הרופאים
* צפיה בנתוני הרופאים

**בעיות בתהליך:** מספר רישיון הרופא לא בהכרח תקין.

**קוד תהליך:** 4P

**שם התהליך:** ניהול משמרות

**תיאור התהליך:** טיפול בכל מה שקשור למשמרות

**התהליך כולל את הפעולות:**

* צפיה בדף המשמרות האישי של כל אח
* קביעת משמרות לאחים
* מחיקת משמרות אחים

**בעיות בתהליך:** המערכת אינה יכולה להבטיח שלא תיווצרנה משמרות ריקות.

**קוד תהליך:** 5P

**שם התהליך:** ניהול תרופות

**תיאור התהליך:** טיפול בכל מה שקשור לתרופות

**התהליך כולל את הפעולות:**

* הוספת תרופה
* מחיקת תרופה (יתאפשר רק אחרי הגדרת תרופה חלופית)
* צפיה בנתוני התרופות
* רישום תרופה לזקן +אפשרות לשינוי אחר כך
* צפיה ברשימת התרופות האוזלות
* עדכון כמות התרופות במלאי אחרי שהזמינו
* נטילת תרופה

**בעיות בתהליך:** אין התייחסות בתוכנה למצב הצבירה של התרופות כך שכמות של תרופה במאגר נשמרת במספרים של מיליליטרים/מיליגרמים ומשם יורדת הכמות של התרופה שניטלה ע"פ מרשם הרופא.

תיאור בעיות:

**מספר הבעיה:** 1

**שם הבעיה:** מספר רישיון הרופא לא בהכרח תקין.

**תיאור הבעיה:** כאשר המנהל רוצה להכניס רופא חדש לבית האבות שלו, המערכת דורשת ממנו להכניס מספר רישיון רופא. המערכת בודקת רק האם הוא הגיוני ולא בודקת האם הוא הונפק כבר והאם הוא בתוקף.

**תיאור סיבת הבעיה:** המערכת אינה מחוברת למאגר מספרי רישיון הרפואה המעודכנים.

**תיאור הנזק:** סיכון לכניסת רופא עם רישיון מזויף.

**כוון לפתרון:** להפוך את התוכנה לאינטרנטית ולחבר אותה למערכת של מספרי הרישיון החוקי.

**מספר הבעיה:** 2

**שם הבעיה:** המערכת אינה יכולה להבטיח שלא תיווצרנה משמרות ריקות.

**תיאור הבעיה:** ישנה אפשרות להגדיר משמרות של אחים במערכת, אך אין למערכת אפשרות לוודא שיהיו אחים בכל המשמרות .

**תיאור סיבת הבעיה:** אין למערכת אפשרות לכפות על המנהל להכניס אחים לבית האבות ולקבוע להם משמרות.

**תיאור הנזק:** עלול להיות מצב שבו לא יהיה אח בבית האבות לשמור על הזקנים ולטפל בהם.

**כוון לפתרון:** למנות מנהל אחראי שידאג לקביעת משמרות.

**מספר הבעיה:** 3

**שם הבעיה:** התכנה לא עוקבת אחרי מצב הצבירה של התרופות במאגר.

**תיאור הבעיה:** גם ברישום תרופה לזקן וגם בהוספת מחיקת ועדכון תרופות במאגר התרופות נמדדות במילימטרים, דבר שאינו נכון הכרח הכוונה בתוכנה היא מיליליטרים/מיליגרמים.

**תיאור סיבת הבעיה:** המערכת לא יכלה לכפות על המשתמש להכניס למאגר תרופות במצבי צבירה כאלה או אחרים.

**תיאור הנזק:** הצורה אינה תקינה ולא נכונה ועלולה לגרום חוסר הבנה מצד המשתמש.

**מדריך למפתח**

שכבת הנתונים DAL

שכבה זו מטפלת בקישור התכנה עם בסיס הנתונים ובה מחלקה לתקשורת עם הConnect.cs.DataBase

המחלקה Connect.cs היחידה ש"מדברת" עם הDataBaes כ שאר המחלקות פונות לDataBase דרך מחלקה זו.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

using System.Data;

namespace Nursing\_Home.DAL

{

public class Connect

{

private DataSet dataset;

private OleDbConnection con;

private OleDbDataAdapter adapter;

public Connect()

{

string path = System.IO.Directory.GetCurrentDirectory();

int x = path.IndexOf("\\bin");

path = path.Substring(0, x) + "\\DAL\\Nursing\_Home.accdb";

con = new OleDbConnection(@"Provider= Microsoft.ACE.OLEDB.12.0;Data Source='" + path + "';Persist Security Info=True");

dataset = new DataSet();

}

private void AddTable(string tableName)

{

adapter = new OleDbDataAdapter("select \* from " + tableName, con);

adapter.Fill(dataset, tableName);

}

public DataTable GetTable(string tableName)

{

if (!(dataset.Tables.Contains(tableName)))

{

AddTable(tableName);

}

return dataset.Tables[tableName];

}

public void UpdateTable(string tableName)

{

adapter = new OleDbDataAdapter("select \* from " + tableName, con);

OleDbCommandBuilder oleDbCommandBuilder = new OleDbCommandBuilder(adapter);

adapter.InsertCommand = oleDbCommandBuilder.GetInsertCommand();

adapter.DeleteCommand = oleDbCommandBuilder.GetDeleteCommand();

adapter.UpdateCommand = oleDbCommandBuilder.GetUpdateCommand();

adapter.Update(dataset, tableName);

}

}

}

שכבת הלוגיקה העיסקית BLL

שכבה המטפלת בכל הפונקציות והלוגיקה הקשורות לתוכנה עצמה.

המחלקות המוכלות בשכבה:

מחלקות בסיס לטיפול בטבלה אחת מתוך הGeneralDB – DataBase

מחלקה עבור כל טבלה מהDataBase היורשת מהמחלקה GenneralDB

מחלקותNameTableRow תפקידן הוא לקשר בן שכבת הנתונים לבין שכבת הממשק החיצוני.

מחלקות הרבים

המחלקה GeneralTable

היא היחידה המתקשרת עם שכבת הDAL מנהלת קשר עם טבלה בודדת מתוך הDataBase המחלקות היורשות ממנה מתייחסות כל אחת לטבלה אחרת במאגר. תפקידן הוא לקשר בין שכבת הנתונים לבין שכבת הממשק החיצוני.

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Nursing\_Home.BLL;

using Nursing\_Home.DAL;

namespace Nursing\_Home.BLL

{

public abstract class GeneralTable

{

private static Connect connect = new Connect();

protected DataTable table;

protected List<GeneralRow> list;

public GeneralTable(string NameTable)

{

table = connect.GetTable(NameTable);

list = new List<GeneralRow>();

}

public void AddItem(GeneralRow item)

{

list.Add(item);

item.row = table.NewRow();

item.FillDataRow();

table.Rows.Add(item.row);

}

public void UpDataItem(GeneralRow item)

{

item.FillDataRow();

}

public void DeleteItem(GeneralRow item)

{

item.row.Delete();

list.Remove(item);

}

public void SaveChanges()

{

connect.UpdateTable(this.table.TableName);

}

}

}

כל אחת מהמחלקות הבאות יורשת מהמחלה GeneralDB ומנהלת את הטיפול בטבלה בודדת מתוך הDataBase

מחלקת רבים- רופאים

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace Nursing\_Home.BLL

{

public class DoctorsTable:GeneralTable

{

public DoctorsTable() : base("Doctor")

{

foreach (DataRow item in base.table.Rows)

{

list.Add(new Doctor(item));

}

}

public List<Doctor> GetList()

{

return base.list.ConvertAll(x => (Doctor)x);

}

}

}

מחלקת רבים- דיירים

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace Nursing\_Home.BLL

{

public class EldersTable:GeneralTable

{

public EldersTable():base("Elder")

{

foreach (DataRow item in base.table.Rows)

{

list.Add(new Elder(item));

}

}

public List<Elder> GetList()

{

return base.list.ConvertAll(x => (Elder)x);

}

}

}

מחלקת רבים- אחים

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace Nursing\_Home.BLL

{

public class NursesTable:GeneralTable

{

public NursesTable():base("Nurse")

{

foreach (DataRow item in base.table.Rows)

{

list.Add(new Nurse(item));

}

}

public List<Nurse> GetList()

{

return base.list.ConvertAll(x => (Nurse)x);

}

}

}

מחלקת רבים- תרופות

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace Nursing\_Home.BLL

{

public class MedicinesTable:GeneralTable

{

public MedicinesTable() : base("Medicine")

{

foreach (DataRow item in base.table.Rows)

{

list.Add(new Medicine(item));

}

}

public List<Medicine> GetList()

{

return base.list.ConvertAll(x => (Medicine)x);

}

public int ThisCode()

{

return GetList().Last().CodeMedicine + 1;

}

}

}

מחלקת רבים- סוגי תרופות

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace Nursing\_Home.BLL

{

public class Types\_Of\_Medicines\_Table:GeneralTable

{

public Types\_Of\_Medicines\_Table():base("Types\_Of\_Medicines")

{

foreach (DataRow item in base.table.Rows)

{

list. Add(new Types\_Of\_Medicines(item));

}

}

public List<Types\_Of\_Medicines> GetList()

{

return base.list.ConvertAll(x => (Types\_Of\_Medicines)x);

}

public int ThisCode()

{

return GetList().Last().CodeMedicineType + 1;

}

}

}

מחלקת רבים- תרופות לזקנים

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Nursing\_Home.BLL

{

public class Medicines\_Of\_Elders\_Table:GeneralTable

{

public Medicines\_Of\_Elders\_Table():base("Medicines\_Of\_Elders")

{

foreach (DataRow item in base.table.Rows)

{

list.Add(new Medicines\_Of\_Elder(item));

}

}

public List<Medicines\_Of\_Elder> GetList()

{

return base.list.ConvertAll(x => (Medicines\_Of\_Elder)x);

}

public int ThisCode()

{

int n = this.GetList().Max(x=>x.AutomaticCode);

return n+1;

}

}

}

מחלקת רבים- נטילת תרופה

using Nursing\_Home.BLL;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Nursing\_Home

{

public class Medicines\_Take\_Table : GeneralTable

{

public Medicines\_Take\_Table() : base("Medicine\_Take")

{

foreach (DataRow item in base.table.Rows)

{

list.Add(new Medicine\_Take(item));

}

}

public List<Medicine\_Take> GetList()

{

return base.list.ConvertAll(x => (Medicine\_Take)x);

}

public int ThisCode()

{

return GetList().Last().Code + 1;

}

}

{

מחלקת רבים- משמרות

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace Nursing\_Home.BLL

{

public class ShiftsTable:GeneralTable

{

public ShiftsTable():base("Shift")

{

foreach (DataRow item in base.table.Rows)

{

list.Add(new Shift(item));

}

}

public List<Shift> GetList()

{

return base.list.ConvertAll(x => (Shift)x);

}

public int ThisCode()

{

return GetList().Last().CodeShift + 1;

}

}

}

מחלקת רבים- משמרות לאחים

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace Nursing\_Home.BLL

{

public class Employee\_Shifts\_Table:GeneralTable

{

public Employee\_Shifts\_Table():base("Employee\_Shifts")

{

foreach (DataRow item in base.table.Rows)

{

list.Add(new Employee\_Shifts(item));

}

}

public List<Employee\_Shifts> GetList()

{

return base.list.ConvertAll(x => (Employee\_Shifts)x);

}

public int EmployeeShiftCode()

{

return GetList().Last().CodeShiftEm + 1;

}

}

}

**מחלקות יחיד**

GeneralRow מחלקת

מחלקת אב אבסטרקטית המטפלת ברשומה בודדה מתוך הטבלה. המחלקה מאחדת את כל הפונקציונליות של כל סוגי רשימות היחיד. וממנה ירשו סוגים שונים של מחלקות יחיד. המחלקה כוללת אוביקט DataRow המכיל הפניה לרשומה המתאימה מתוך ה.DataTable

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

namespace Nursing\_Home.BLL

{

public abstract class GeneralRow

{

public DataRow row;

public GeneralRow()

{

}

public GeneralRow(DataRow row)

{

this.row = row;

FillFields();

}

public void UpdateItem(GeneralRow item)

{

item.FillDataRow();

}

// פונקציה שמקבלת נתונים מהאקסס ושומרת אותם בתוך משתני המחלקה

public abstract void FillFields();

// פונקציה שממלאה את הנתונים בטבלה באקסס

public abstract void FillDataRow();

}

}

כל אחת מהמחלקות הבאות מטפלת ברשומה בודדת מתוך טבלה מסויימת תכונותיה: משתנים המקבילים לשדות של אותה טבלה (גם בסוג הנתונים), אוביקט מסוג מחלקת DB השייכת לטבלה זו ואובייקט מסוג DataRoe .

מחלקת יחיד- רופא

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Nursing\_Home.BLL;

namespace Nursing\_Home.BLL

{

public class Doctor: GeneralRow

{

public string IdDoctor { get; set; }

public string NameDoctor { get; set; }

public string LicenseNumber { get; set; }

public string PoneNumber { get; set; }

public string Address { get; set; }

public DateTime DateOfBirth { get; set; }

public Doctor():base()

{

}

public Doctor(DataRow row):base(row)

{

}

public override void FillDataRow()

{

row["IdDoctor"] = this.IdDoctor;

row["NameDoctor"] = this.NameDoctor;

row["LicenseNumber"] = this.LicenseNumber;

row["PoneNumber"] = this.PoneNumber;

row["Address"] = this.Address;

row["DateOfBirth"] = this.DateOfBirth;

}

public override void FillFields()

{

this.IdDoctor = row["IdDoctor"].ToString();

this.NameDoctor = row["NameDoctor"].ToString();

this.LicenseNumber = row["LicenseNumber"].ToString();

this.PoneNumber = row["PoneNumber"].ToString();

this.Address = row["Address"].ToString();

this.DateOfBirth =Convert.ToDateTime( row["DateOfBirth"]);

}

public override string ToString()

{

return " ת.ז. של הרופא: " + IdDoctor + " שם הדוקטור: " + NameDoctor + " מספר רישיו רופא: " + LicenseNumber + " מספר הטלפון: " + PoneNumber + " כתובת:" + Address + " תאריך לידה: " + DateOfBirth.ToShortDateString();

}

}

}

מחלקת יחיד- דייר

using Nursing\_Home.BLL;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Nursing\_Home.BLL

{

public class Elder:GeneralRow

{

public string IdElder { get; set; }

public string NameElder { get; set; }

public DateTime DateOfBirth { get; set; }

public int HospitalizationRoom { get; set; }

public string IdDoctor { get; set; }

public string PhoneContact { get; set; }

public string NameContact { get; set; }

public string ProximityContact { get; set; }

public Elder():base()

{

}

public Elder(DataRow row):base(row)

{

}

public override void FillDataRow()

{

row["IdElder"] = this.IdElder;

row["NameElder"] = this.NameElder;

row["DateOfBirth"] = this.DateOfBirth;

row["HospitalizationRoom"] = this.HospitalizationRoom;

row["IdDoctor"] = this.IdDoctor;

row["PhoneContact"] = this.PhoneContact;

row["NameContact"] = this.NameContact;

row["ProximityContact"] = this.ProximityContact;

}

public override void FillFields()

{

this.IdElder = row["IdElder"].ToString();

this.NameElder = row["NameElder"].ToString();

this.DateOfBirth =Convert.ToDateTime( row["DateOfBirth"]);

this.HospitalizationRoom = Convert.ToInt32(row["HospitalizationRoom"]);

this.IdDoctor = row["IdDoctor"].ToString();

this.PhoneContact = row["PhoneContact"].ToString();

this.NameContact = row["NameContact"].ToString();

this.ProximityContact = row["ProximityContact"].ToString();

}

public override string ToString()

{

return " ת.ז. הזקן: " + IdElder + " שם הזקן: " + NameElder + " תאריך לידה: " + DateOfBirth.ToShortDateString() + " חדר אישפוז: " + HospitalizationRoom + " קוד הרופא המטפל: " + IdDoctor + " שם איש הקשר: " + NameContact + " מספרו: " + PhoneContact + " קרבה: " + ProximityContact;

}

}

}

מחלקת יחיד- אח

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Nursing\_Home.BLL

{

public class Nurse:GeneralRow

{

public string IdNurse { get; set; }

public string NameNurse { get; set; }

public string PhoneNumber { get; set; }

public string Address { get; set; }

public DateTime DateOfBirth { get; set; }

public Nurse() : base()

{

}

public Nurse(DataRow row):base(row)

{

}

public override void FillDataRow()

{

row["IdNurse"] = this.IdNurse;

row["NameNurse"] = this.NameNurse;

row["PhoneNumber"] = this.PhoneNumber;

row["Address"] = this.Address;

row["DateOfBirth"] = this.DateOfBirth;

}

public override void FillFields()

{

this.IdNurse = row["IdNurse"].ToString();

this.NameNurse = row["NameNurse"].ToString();

this.PhoneNumber = row["PhoneNumber"].ToString();

this.Address = row["Address"].ToString();

this.DateOfBirth = Convert.ToDateTime(row["DateOfBirth"]);

}

public override string ToString()

{

return " ת.ז. אחות:" + IdNurse + " שם האחות: " + " מספר טלפון: " + PhoneNumber + " כתובת: " + Address + "תאריך לידה: " + DateOfBirth.ToShortDateString();

}

}

}

מחלקת יחיד- תרופה

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Nursing\_Home.BLL

{

public class Medicine:GeneralRow

{

public int CodeMedicine { get; set; }

public string NameMedicine { get; set; }

public int CodeMedicineType { get; set; }

public int TheIdealAmauntInStock { get; set; }

public int UnitsInStock { get; set; }

public Medicine():base()

{

}

public Medicine(DataRow row):base(row)

{

}

public override void FillDataRow()

{

row["CodeMedicine"] = this.CodeMedicine;

row["NameMedicine"] = this.NameMedicine;

row["CodeMedicineType"] = this.CodeMedicineType;

row["TheIdealAmauntInStock"] = this.TheIdealAmauntInStock;

row["UnitsInStock"] = this.UnitsInStock;

}

public override void FillFields()

{

this.CodeMedicine = Convert.ToInt32(row["CodeMedicine"]);

this.NameMedicine = (row["NameMedicine"]).ToString();

this.CodeMedicineType = Convert.ToInt32(row["CodeMedicineType"]);

this.TheIdealAmauntInStock = Convert.ToInt32(row["TheIdealAmauntInStock"]);

this.UnitsInStock = Convert.ToInt32(row["UnitsInStock"]);

}

public override string ToString()

{

return "קוד התרופה: " + CodeMedicine + " שם התרופה: " + NameMedicine + " קוד סוג התרופה: " + CodeMedicineType + " הכמות האידיאלית במלאי: " + TheIdealAmauntInStock + " הכמות הנוכחית במלאי: " + UnitsInStock;

}

}

}

מחלקת יחיד- סוג תרופה

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Nursing\_Home.BLL

{

public class Types\_Of\_Medicines:GeneralRow

{

public int CodeMedicineType { get; set; }

public string NameMedicineType { get; set; }

public Types\_Of\_Medicines():base()

{

}

public Types\_Of\_Medicines(DataRow row):base(row)

{

}

public override void FillDataRow()

{

row["CodeMedicineType"] = this.CodeMedicineType;

row["NameMedicineType"] = this.NameMedicineType;

}

public override void FillFields()

{

this.CodeMedicineType = Convert.ToInt32(row["CodeMedicineType"]);

this.NameMedicineType = row["NameMedicineType"].ToString();

}

public override string ToString()

{

return "קוד סוג תרופה: " + CodeMedicineType + " הסוג: " + NameMedicineType;

}

}

}

מחלקת יחיד- תרופה לזקן

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Nursing\_Home.BLL

{

public class Medicines\_Of\_Elder:GeneralRow

{

public int AutomaticCode { get; set; }

public string IdElder { get; set; }

public string IdDoctor { get; set; }

public int CodeMedicine { get; set; }

public DateTime StartDate { get; set; }

public DateTime EndDate { get; set; }

public int TimesADay { get; set; }

public int NumberOfMilimetersAtADay { get; set; }

public Medicines\_Of\_Elder():base()

{

}

public Medicines\_Of\_Elder(DataRow row):base(row)

{

}

public override void FillDataRow()

{

row["AutomaticCode"] = this.AutomaticCode;

row["IdElder"] = this.IdElder;

row["IdDoctor"] = this.IdDoctor;

row["CodeMedicine"] = this.CodeMedicine;

row["StartDate"] = this.StartDate;

row["EndDate"] = this.EndDate;

row["TimesADay"] = this.TimesADay;

row["NumberOfMilimetersAtADay"] = this.NumberOfMilimetersAtADay;

}

public override void FillFields()

{

this.AutomaticCode = Convert.ToInt32(row["AutomaticCode"]);

this.IdElder= row["IdElder"].ToString();

this.IdDoctor = row["IdDoctor"].ToString();

this.CodeMedicine = Convert.ToInt32(row["CodeMedicine"]);

this.StartDate = Convert.ToDateTime(row["StartDate"]);

this.EndDate = Convert.ToDateTime(row["EndDate"]);

this.TimesADay = Convert.ToInt32(row["TimesADay"]);

this.NumberOfMilimetersAtADay = Convert.ToInt32(row["NumberOfMilimetersAtADay"]);

}

public string NameMedicine(int code)

{

Medicine m = MyDB.medicines.GetList().First(x => x.CodeMedicine == code);

return m.NameMedicine;

}

public string NameElder(string id)

{

Elder e = MyDB.elders.GetList().First(x => x.IdElder == id);

return e.NameElder;

}

public string NameType(int code)

{

Types\_Of\_Medicines t = MyDB.types.GetList().First(x => x.CodeMedicineType == code);

return t.NameMedicineType;

}

public override string ToString()

{

return "ת.ז. זקן: " + IdElder + " ת.ז. של הרופא הרושם: " + IdDoctor + " קוד התרופה: " + CodeMedicine + " תאריך תחילת נטילתה: " + StartDate.ToShortDateString() + " תאריך סיום נטילתה +" + EndDate.ToShortDateString() + " מספר הפעמים ביום: " + TimesADay + " מספר המילימטרים בכל פעם: " + NumberOfMilimetersAtADay;

}

}

}

מחקת יחיד- נטילת תרופה

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Nursing\_Home.BLL;

namespace Nursing\_Home

{

public class Medicine\_Take : GeneralRow

{

public int Code { get; set; }

public int AutomaticCode { get; set; }

public DateTime InDate { get; set; }

public int InShift { get; set; }

public bool Status { get; set; }

public Medicine\_Take() : base()

{

}

public Medicine\_Take(DataRow row) : base(row)

{

}

public override void FillDataRow()

{

row["Code"] = this.Code;

row["AutomaticCode"] = this.AutomaticCode;

row["InDate"] = this.InDate;

row["InShift"] = this.InShift;

row["Status"] = this.Status;

}

public override void FillFields()

{

this.Code = Convert.ToInt32(row["Code"]);

this.AutomaticCode =Convert.ToInt32(row["AutomaticCode"]);

this.InDate =Convert.ToDateTime(row["InDate"]);

this.InShift =Convert.ToInt32(row["InShift"]);

this.Status = Convert.ToBoolean(row["Status"]);

}

public override string ToString()

{

return "קוד נטילת תרופה: " + Code + "תאריך: " + InDate + "משמרת: " + InShift + "האם ניתן? " + Status;

}

}

}

מחלקת יחיד- משמרת

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Nursing\_Home.BLL

{

public class Shift:GeneralRow

{

public int CodeShift { get; set; }

public int NameShift { get; set; }

public int DayInWeek { get; set; }

public int PaymentForHour { get; set; }

public DateTime BeginningTime { get; set; }

public DateTime EndTime { get; set; }

public int Nnurses { get; set; }

public Shift():base()

{

}

public Shift(DataRow row):base(row)

{

}

public override void FillDataRow()

{

row["CodeShift"] = this.CodeShift;

row["NameShift"] = this.NameShift;

row["DayInWeek"] = this.DayInWeek;

row["PaymentForHour"] = this.PaymentForHour;

row["BeginningTime"] = this.BeginningTime;

row["EndTime"] = this.EndTime;

row["Nnurses"] = this.Nnurses;

}

public override void FillFields()

{

this.CodeShift = Convert.ToInt32(row["CodeShift"]);

this.NameShift =Convert.ToInt32(row["NameShift"]);

this.DayInWeek =Convert.ToInt32(row["DayInWeek"]);

this.PaymentForHour = Convert.ToInt32(row["PaymentForHour"]);

this.BeginningTime = Convert.ToDateTime(row["BeginningTime"]);

this.EndTime = Convert.ToDateTime(row["EndTime"]);

this.Nnurses = Convert.ToInt32(row["Nnurses"]);

}

public override string ToString()

{

return "משמרת: " + NameShift + " מתחילה בשעה: " + BeginningTime + " מסתיימת בשעה: " + EndTime+" מספר האחיות הנחוץ למשמרת:"+ Nnurses;

}

}

}

מחלקת יחיד- משמרת לאח

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Nursing\_Home.BLL

{

public class Employee\_Shifts:GeneralRow

{

public int CodeShiftEm { get; set; }

public string CodeEmployee { get; set; }

public int CodeShift { get; set; }

public Employee\_Shifts():base()

{

}

public Employee\_Shifts(DataRow row) : base(row)

{

}

public override void FillDataRow()

{

row["CodeShiftEm"] = this.CodeShiftEm;

row["CodeEmployee"] = this.CodeEmployee;

row["CodeShift"] = this.CodeShift;

}

public override void FillFields()

{

this.CodeShiftEm = Convert.ToInt32(row["CodeShiftEm"]);

this.CodeEmployee = row["CodeEmployee"].ToString();

this.CodeShift = Convert.ToInt32(row["CodeShift"]);

}

public override string ToString()

{

return "קוד עובד: " + CodeEmployee + " קוד משמרת: " + CodeShift;

}

}

}

שכבת הישום GUI

שכבה המטפלת בממשק החיצוני של התכנה:

עמוד הכניסה:

using Nursing\_Home.BLL;

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 Nursing\_Home.GUI

{

public partial class MainPage : Form

{

Nurse n;

Doctor d;

public MainPage()

{

InitializeComponent();

this.Text = "כניסה";

}

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

{

button4.Visible = true;

textBox1.Visible = true;

label1.Visible = true;

}

private void button2\_Click(object sender, EventArgs e)

{

button5.Visible = true;

label2.Visible = true;

textBox2.Visible = true;

}

private void button3\_Click(object sender, EventArgs e)

{

textBox3.Visible = true;

button6.Visible = true;

label3.Visible = true;

}

private void button4\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "2010")

{

DirectorPage x = new DirectorPage();

this.Hide();

x.ShowDialog();

this.Show();

}

else

{

textBox1.Text = "";

errorProvider1.SetError(textBox1, "סיסמאת המנהל שגויה");

}

}

private void button5\_Click(object sender, EventArgs e)

{

d = MyDB.doctors.GetList().FirstOrDefault(x => x.IdDoctor == textBox2.Text);

if (!Validation.CheckId(textBox2.Text))

{

textBox2.Text = "";

errorProvider1.SetError(textBox2, "תעודת הזהות שגויה");

}

else if (d == null)

{

textBox2.Text = "";

errorProvider1.SetError(textBox2, "תעודת זהות זו אינה מופיעה במאגר הרופאים, אנא פנה למנהל");

}

else

{

DoctorsPage dp = new DoctorsPage(d);

this.Hide();

dp.ShowDialog();

this.Show();

}

}

private void button6\_Click(object sender, EventArgs e)

{

n = MyDB.nurses.GetList().FirstOrDefault(x => x.IdNurse == textBox3.Text);

if (!Validation.CheckId(textBox3.Text))

{

textBox3.Text = "";

errorProvider1.SetError(textBox3, "תעודת הזהות שגויה");

}

else if (n == null)

{

textBox3.Text = "";

errorProvider1.SetError(textBox3, "תעודת זהות זו אינה מופיעה במאגר האחים, אנא פנה למנהל");

}

else

{

NursesPage np = new NursesPage(n);

this.Hide();

np.ShowDialog();

this.Show();

}

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Validation.IsNum(e.KeyChar.ToString()) && e.KeyChar != '\b')

{

e.Handled = true;

}

}

} }

עמוד המנהל:

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;

using Nursing\_Home.BLL;

namespace Nursing\_Home.GUI

{

public partial class DirectorPage : Form

{

int status,z;

Medicine medicine;

public DirectorPage()

{

InitializeComponent();

this.Text = "עמוד המנהל";

comboBox1.DataSource = MyDB.medicines.GetList().Select(x => x.NameMedicine).ToList();

comboBox1.SelectedItem = null;

medicine = null;

}

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

{

label1.Text = "ע'מ למחוק אחים מהמאגר יש לבחור את האח המבוקש וללחוץ על מחק";

dataGridView1.DataSource = MyDB.doctors.GetList().Select(x=> new { x.IdDoctor,x.NameDoctor,x.LicenseNumber,x.PoneNumber,x.Address,x.DateOfBirth.Date}).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הרופא";

dataGridView1.Columns[2].HeaderText = "מספר רישיון";

dataGridView1.Columns[3].HeaderText = "מספר פלאפון";

dataGridView1.Columns[4].HeaderText = "כתובת מגורים";

dataGridView1.Columns[5].HeaderText = "תאריך לידה";

button1.Visible = true;

status=1;

}

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

{

status=1;

AddDN a = new AddDN(true);

this.Hide();

a.ShowDialog();

this.Show();

dataGridView1.DataSource = MyDB.doctors.GetList().Select(x => new { x.IdDoctor, x.NameDoctor, x.LicenseNumber, x.PoneNumber, x.Address, x.DateOfBirth.Date }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הרופא";

dataGridView1.Columns[2].HeaderText = "מספר רישיון";

dataGridView1.Columns[3].HeaderText = "מספר פלאפון";

dataGridView1.Columns[4].HeaderText = "כתובת מגורים";

dataGridView1.Columns[5].HeaderText = "תאריך לידה";

}

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

{

label1.Text="ע'מ לעדכן את נתוני הרופאים יש ללחוץ פעמיים על הרופא/ה המבוקש/ת";

dataGridView1.DataSource = MyDB.doctors.GetList().Select(x => new { x.IdDoctor, x.NameDoctor, x.LicenseNumber, x.PoneNumber, x.Address, x.DateOfBirth.Date }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הרופא";

dataGridView1.Columns[2].HeaderText = "מספר רישיון";

dataGridView1.Columns[3].HeaderText = "מספר פלאפון";

dataGridView1.Columns[4].HeaderText = "כתובת מגורים";

dataGridView1.Columns[5].HeaderText = "תאריך לידה";

button1.Visible = false;

status=1;

}

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

{

label1.Text = "ע'מ לעדכן את נתוני האחים יש ללחוץ פעמיים על האח/ות המבוקש/ת";

dataGridView1.DataSource = MyDB.nurses.GetList().Select(x=> new { x.IdNurse,x.NameNurse,x.PhoneNumber,x.Address,x.DateOfBirth.Date}).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם האח";

dataGridView1.Columns[2].HeaderText = "מספר פלאפון";

dataGridView1.Columns[3].HeaderText = "כתובת מגורים";

dataGridView1.Columns[4].HeaderText = "תאריך לידה";

button1.Visible = false;

status=2;

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

if(status==1)

{

Doctor d = new Doctor();

foreach (var item in MyDB.doctors.GetList())

{

if (item.IdDoctor == dataGridView1.CurrentRow.Cells[0].Value.ToString())

d = item;

}

AddDN a = new AddDN(d);

this.Hide();

a.ShowDialog();

this.Show();

dataGridView1.DataSource = MyDB.doctors.GetList().Select(x => new { x.IdDoctor, x.NameDoctor, x.LicenseNumber, x.PoneNumber, x.Address, x.DateOfBirth.Date }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הרופא";

dataGridView1.Columns[2].HeaderText = "מספר רישיון";

dataGridView1.Columns[3].HeaderText = "מספר פלאפון";

dataGridView1.Columns[4].HeaderText = "כתובת מגורים";

dataGridView1.Columns[5].HeaderText = "תאריך לידה";

}

if(status==2)

{

Nurse n = new Nurse();

foreach (var item in MyDB.nurses.GetList())

{

if (item.IdNurse == dataGridView1.CurrentRow.Cells[0].Value.ToString())

n = item;

}

AddDN a = new AddDN(n);

this.Hide();

a.ShowDialog();

this.Show();

dataGridView1.DataSource = MyDB.nurses.GetList().Select(x => new { x.IdNurse, x.NameNurse, x.PhoneNumber, x.Address, x.DateOfBirth.Date }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם האח";

dataGridView1.Columns[2].HeaderText = "מספר פלאפון";

dataGridView1.Columns[3].HeaderText = "כתובת מגורים";

dataGridView1.Columns[4].HeaderText = "תאריך לידה";

}

if (status == 3)

{

Elder el = new Elder();

foreach (var item in MyDB.elders.GetList())

{

if (item.IdElder == dataGridView1.CurrentRow.Cells[0].Value.ToString())

el = item;

}

AddElder a = new AddElder(el);

this.Hide();

a.ShowDialog();

this.Show();

dataGridView1.DataSource = MyDB.elders.GetList().Select(x => new { x.IdDoctor, x.NameElder, x.DateOfBirth.Date, x.HospitalizationRoom, MyDB.doctors.GetList().FirstOrDefault(y => y.IdDoctor == x.IdDoctor).NameDoctor, x.PhoneContact, x.NameContact, x.ProximityContact }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הדייר";

dataGridView1.Columns[2].HeaderText = "תאריך לידה";

dataGridView1.Columns[3].HeaderText = "חדר אישפוז";

dataGridView1.Columns[4].HeaderText = "הרופא המטפל";

dataGridView1.Columns[5].HeaderText = "מספר איש הקשר";

dataGridView1.Columns[6].HeaderText = "שם איש הקשר";

dataGridView1.Columns[7].HeaderText = "קרבה משפחתית";

}

if (status == 4)

{

Medicine m = new Medicine();

foreach (var item in MyDB.medicines.GetList())

{

if (item.CodeMedicine == Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value))

m = item;

}

AddMedicine a = new AddMedicine(m);

this.Hide();

a.ShowDialog();

this.Show();

dataGridView1.DataSource = MyDB.medicines.GetList().Select(x => new { x.CodeMedicine, x.NameMedicine, MyDB.types.GetList().First(y => y.CodeMedicineType == x.CodeMedicineType).NameMedicineType, x.TheIdealAmauntInStock, x.UnitsInStock }).ToList();

dataGridView1.Columns[0].HeaderText = "קוד תרופה";

dataGridView1.Columns[1].HeaderText = "שם תרופה";

dataGridView1.Columns[2].HeaderText = "סוג תרופה";

dataGridView1.Columns[3].HeaderText = "כמות אידיאלית";

dataGridView1.Columns[4].HeaderText = "כמות במלאי כרגע";

}

if(status==44)

{

Medicine m = MyDB.medicines.GetList().First(x => x.CodeMedicine == Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value));

UpdateOrderingMedicine u = new UpdateOrderingMedicine(m);

this.Hide();

u.ShowDialog();

this.Show();

dataGridView1.DataSource = MyDB.medicines.GetList().Where(x => x.UnitsInStock < x.TheIdealAmauntInStock / 2).Select(x => new { x.CodeMedicine, x.NameMedicine, MyDB.types.GetList().First(y => y.CodeMedicineType == x.CodeMedicineType).NameMedicineType, x.TheIdealAmauntInStock, x.UnitsInStock }).ToList();

dataGridView1.Columns[0].HeaderText = "קוד תרופה";

dataGridView1.Columns[1].HeaderText = "שם תרופה";

dataGridView1.Columns[2].HeaderText = "סוג תרופה";

dataGridView1.Columns[3].HeaderText = "כמות אידיאלית";

dataGridView1.Columns[4].HeaderText = "כמות במלאי כרגע";

}

}

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

{

status=2;

AddDN a = new AddDN(false);

this.Hide();

a.ShowDialog();

this.Show();

dataGridView1.DataSource = MyDB.nurses.GetList().Select(x => new { x.IdNurse, x.NameNurse, x.PhoneNumber, x.Address, x.DateOfBirth.Date }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם האח";

dataGridView1.Columns[2].HeaderText = "מספר פלאפון";

dataGridView1.Columns[3].HeaderText = "כתובת מגורים";

dataGridView1.Columns[4].HeaderText = "תאריך לידה";

}

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

{

label1.Text = "רשימת האחים";

status =2;

dataGridView1.DataSource = MyDB.nurses.GetList().Select(x => new { x.IdNurse, x.NameNurse, x.PhoneNumber, x.Address, x.DateOfBirth.Date }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם האח";

dataGridView1.Columns[2].HeaderText = "מספר פלאפון";

dataGridView1.Columns[3].HeaderText = "כתובת מגורים";

dataGridView1.Columns[4].HeaderText = "תאריך לידה";

button1.Visible = false;

}

private void צפיהבנתוניהרופאיםToolStripMenuItem1\_Click(object sender, EventArgs e)

{

label1.Text = "רשימת הרופאים";

status =1;

dataGridView1.DataSource = MyDB.doctors.GetList().Select(x => new { x.IdDoctor, x.NameDoctor, x.LicenseNumber, x.PoneNumber, x.Address, x.DateOfBirth.Date }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הרופא";

dataGridView1.Columns[2].HeaderText = "מספר רישיון";

dataGridView1.Columns[3].HeaderText = "מספר פלאפון";

dataGridView1.Columns[4].HeaderText = "כתובת מגורים";

dataGridView1.Columns[5].HeaderText = "תאריך לידה";

button1.Visible = false;

}

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

{

label1.Text = "ע'מ למחוק אחים מהמאגר יש לבחור את האח המבוקש וללחוץ על מחק";

dataGridView1.DataSource = MyDB.nurses.GetList().Select(x => new { x.IdNurse, x.NameNurse, x.PhoneNumber, x.Address, x.DateOfBirth.Date }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם האח";

dataGridView1.Columns[2].HeaderText = "מספר פלאפון";

dataGridView1.Columns[3].HeaderText = "כתובת מגורים";

dataGridView1.Columns[4].HeaderText = "תאריך לידה";

button1.Visible = true;

status=2;

}

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

{

if (status==1)

{

Doctor d = MyDB.doctors.GetList().FirstOrDefault(x => x.IdDoctor == dataGridView1.CurrentRow.Cells[0].Value.ToString());

DialogResult r = MessageBox.Show("האם ברצונך למחוק את הרופא " + d.NameDoctor , "מחיקת רופא מהמאגר", MessageBoxButtons.YesNo, MessageBoxIcon.Question);//איך מסדרים את זה

if (r == DialogResult.Yes)

{

Elder elder = MyDB.elders.GetList().FirstOrDefault(x => x.IdDoctor == d.IdDoctor);

if (elder != null)

{

DeleteDoctor deleteDoctor = new DeleteDoctor(d);

this.Hide();

deleteDoctor.ShowDialog();

this.Show();

}

else

{

MyDB.doctors.DeleteItem(d);

MyDB.doctors.SaveChanges();

MessageBox.Show("מחיקת הרופא מהמאגר נשלמה");

}

//MyDB.doctors.DeleteItem(d);

//MyDB.doctors.SaveChanges();

dataGridView1.DataSource = MyDB.doctors.GetList().Select(x => new { x.IdDoctor, x.NameDoctor, x.LicenseNumber, x.PoneNumber, x.Address, x.DateOfBirth.Date }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הרופא";

dataGridView1.Columns[2].HeaderText = "מספר רישיון";

dataGridView1.Columns[3].HeaderText = "מספר פלאפון";

dataGridView1.Columns[4].HeaderText = "כתובת מגורים";

dataGridView1.Columns[5].HeaderText = "תאריך לידה";

}

}

if (status==2)

{

Nurse n = MyDB.nurses.GetList().FirstOrDefault(x => x.IdNurse == dataGridView1.CurrentRow.Cells[0].Value.ToString());

DialogResult r = MessageBox.Show("האם ברצונך למחוק את האח " + n.NameNurse, "מחיקת אח מהמאגר", MessageBoxButtons.YesNo, MessageBoxIcon.Question);//איך מסדרים את זה

if (r == DialogResult.Yes)

{

int i = 0;

foreach (var item in MyDB.shiftsemp.GetList())

{

if (item.CodeEmployee == n.IdNurse)

{

i++;

Shift shift = MyDB.shifts.GetList().First(q => q.CodeShift == item.CodeShift);

shift.Nnurses++;

MyDB.shifts.UpDataItem(shift);

MyDB.shifts.SaveChanges();

MyDB.shiftsemp.DeleteItem(item);

MyDB.shiftsemp.SaveChanges();

}

}

MessageBox.Show("לתשומת ליבך, במהלך הסרת האח מהמאגר נמחקו גם כן " + i + " משמרות שלו. יש לדאוג לקבוע בהקדם אחים אחרים למשמרות אלו");

MyDB.nurses.DeleteItem(n);

MyDB.nurses.SaveChanges();

dataGridView1.DataSource = MyDB.nurses.GetList().Select(x => new { x.IdNurse, x.NameNurse, x.PhoneNumber, x.Address, x.DateOfBirth.Date }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם האח";

dataGridView1.Columns[2].HeaderText = "מספר פלאפון";

dataGridView1.Columns[3].HeaderText = "כתובת מגורים";

dataGridView1.Columns[4].HeaderText = "תאריך לידה";

}

}

if(status==3)

{

Elder el = MyDB.elders.GetList().First(x => x.IdElder == dataGridView1.CurrentRow.Cells[0].Value.ToString());

DialogResult r = MessageBox.Show("האם ברצונך למחוק את הדייר " + el.NameElder, "מחיקת דייר מהמאגר", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (r == DialogResult.Yes) {

MyDB.elders.DeleteItem(el);

MyDB.elders.SaveChanges();

dataGridView1.DataSource = MyDB.elders.GetList().Select(x => new { x.IdDoctor, x.NameElder, x.DateOfBirth.Date, x.HospitalizationRoom, MyDB.doctors.GetList().FirstOrDefault(y => y.IdDoctor == x.IdDoctor).NameDoctor, x.PhoneContact, x.NameContact, x.ProximityContact }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הדייר";

dataGridView1.Columns[2].HeaderText = "תאריך לידה";

dataGridView1.Columns[3].HeaderText = "חדר אישפוז";

dataGridView1.Columns[4].HeaderText = "הרופא המטפל";

dataGridView1.Columns[5].HeaderText = "מספר איש הקשר";

dataGridView1.Columns[6].HeaderText = "שם איש הקשר";

dataGridView1.Columns[7].HeaderText = "קרבה משפחתית";

}

}

if (status == 4)

{

Medicine m = MyDB.medicines.GetList().FirstOrDefault(x => x.CodeMedicine ==Convert.ToInt32( dataGridView1.CurrentRow.Cells[0].Value));

medicine = m;

DialogResult r = MessageBox.Show("האם ברצונך למחוק מהמאגר את התרופה " + m.NameMedicine, "מחיקת תרופה מהמאגר", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (r == DialogResult.Yes)

{

comboBox1.Visible = true;

label2.Visible = true;

label1.Visible = false;

dataGridView1.Visible = false;

menuStrip1.Visible = false;

button1.Visible = false;

comboBox1.SelectedItem = null;

z = 5;

comboBox1.DataSource = MyDB.medicines.GetList().Where(x => x.NameMedicine != medicine.NameMedicine).Select(x => x.NameMedicine).ToList();

}

}

}

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

{

label1.Text = "ע'מ למחוק דיירים מהמאגר יש לבחור את הדייר המבוקש וללחוץ על מחק";

status = 3;

dataGridView1.DataSource = MyDB.elders.GetList().Select(x=> new {x.IdDoctor,x.NameElder,x.DateOfBirth.Date,x.HospitalizationRoom,MyDB.doctors.GetList().FirstOrDefault(y=>y.IdDoctor==x.IdDoctor).NameDoctor,x.PhoneContact,x.NameContact,x.ProximityContact }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הדייר";

dataGridView1.Columns[2].HeaderText = "תאריך לידה";

dataGridView1.Columns[3].HeaderText = "חדר אישפוז";

dataGridView1.Columns[4].HeaderText = "הרופא המטפל";

dataGridView1.Columns[5].HeaderText = "מספר איש הקשר";

dataGridView1.Columns[6].HeaderText = "שם איש הקשר";

dataGridView1.Columns[7].HeaderText = "קרבה משפחתית";

button1.Visible = true;

}

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

{

label1.Text = "רשימת הדיירים";

status = 3;

dataGridView1.DataSource = MyDB.elders.GetList().Select(x => new { x.IdDoctor, x.NameElder, x.DateOfBirth.Date, x.HospitalizationRoom, MyDB.doctors.GetList().FirstOrDefault(y => y.IdDoctor == x.IdDoctor).NameDoctor, x.PhoneContact, x.NameContact, x.ProximityContact }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הדייר";

dataGridView1.Columns[2].HeaderText = "תאריך לידה";

dataGridView1.Columns[3].HeaderText = "חדר אישפוז";

dataGridView1.Columns[4].HeaderText = "הרופא המטפל";

dataGridView1.Columns[5].HeaderText = "מספר איש הקשר";

dataGridView1.Columns[6].HeaderText = "שם איש הקשר";

dataGridView1.Columns[7].HeaderText = "קרבה משפחתית";

button1.Visible = false;

}

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

{

label1.Text = "ע'מ לעדכן את נתוני הדיירים יש ללחוץ פעמיים על הדייר/ת המבוקש/ת";

status = 3;

dataGridView1.DataSource = MyDB.elders.GetList().Select(x => new { x.IdDoctor, x.NameElder, x.DateOfBirth.Date, x.HospitalizationRoom, MyDB.doctors.GetList().FirstOrDefault(y => y.IdDoctor == x.IdDoctor).NameDoctor, x.PhoneContact, x.NameContact, x.ProximityContact }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הדייר";

dataGridView1.Columns[2].HeaderText = "תאריך לידה";

dataGridView1.Columns[3].HeaderText = "חדר אישפוז";

dataGridView1.Columns[4].HeaderText = "הרופא המטפל";

dataGridView1.Columns[5].HeaderText = "מספר איש הקשר";

dataGridView1.Columns[6].HeaderText = "שם איש הקשר";

dataGridView1.Columns[7].HeaderText = "קרבה משפחתית";

button1.Visible = false;

}

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

{

status = 3;

AddElder a = new AddElder();

this.Hide();

a.ShowDialog();

this.Show();

dataGridView1.DataSource = MyDB.elders.GetList().Select(x => new { x.IdDoctor, x.NameElder, x.DateOfBirth.Date, x.HospitalizationRoom, MyDB.doctors.GetList().FirstOrDefault(y => y.IdDoctor == x.IdDoctor).NameDoctor, x.PhoneContact, x.NameContact, x.ProximityContact }).ToList();

dataGridView1.Columns[0].HeaderText = "תעודת זהות";

dataGridView1.Columns[1].HeaderText = "שם הדייר";

dataGridView1.Columns[2].HeaderText = "תאריך לידה";

dataGridView1.Columns[3].HeaderText = "חדר אישפוז";

dataGridView1.Columns[4].HeaderText = "הרופא המטפל";

dataGridView1.Columns[5].HeaderText = "מספר איש הקשר";

dataGridView1.Columns[6].HeaderText = "שם איש הקשר";

dataGridView1.Columns[7].HeaderText = "קרבה משפחתית";

} private void הוספתתרופותלמאגרToolStripMenuItem\_Click(object sender, EventArgs e)

{

status = 4;

AddMedicine a = new AddMedicine();

this.Hide();

a.ShowDialog();

this.Show();

dataGridView1.DataSource = MyDB.medicines.GetList().Select(x => new { x.CodeMedicine, x.NameMedicine, MyDB.types.GetList().First(y => y.CodeMedicineType == x.CodeMedicineType).NameMedicineType, x.TheIdealAmauntInStock, x.UnitsInStock }).ToList();

dataGridView1.Columns[0].HeaderText = "קוד תרופה";

dataGridView1.Columns[1].HeaderText = "שם תרופה";

dataGridView1.Columns[2].HeaderText = "סוג תרופה";

dataGridView1.Columns[3].HeaderText = "כמות אידיאלית";

dataGridView1.Columns[4].HeaderText = "כמות במלאי כרגע";

}

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

{

label1.Text = "ע'מ למחוק תרופות מהמאגר יש לבחור את התרופה המבוקשת וללחוץ על מחק";

status = 4;

dataGridView1.DataSource = MyDB.medicines.GetList().Select(x => new { x.CodeMedicine, x.NameMedicine, MyDB.types.GetList().First(y => y.CodeMedicineType == x.CodeMedicineType).NameMedicineType, x.TheIdealAmauntInStock, x.UnitsInStock }).ToList();

dataGridView1.Columns[0].HeaderText = "קוד תרופה";

dataGridView1.Columns[1].HeaderText = "שם תרופה";

dataGridView1.Columns[2].HeaderText = "סוג תרופה";

dataGridView1.Columns[3].HeaderText = "כמות אידיאלית";

dataGridView1.Columns[4].HeaderText = "כמות במלאי כרגע";

button1.Visible = true;

}

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

{

label1.Text = "רשימת התרופות החסרות";

status = 44;

dataGridView1.DataSource = MyDB.medicines.GetList().Where(x => x.UnitsInStock < x.TheIdealAmauntInStock / 2).Select(x => new { x.CodeMedicine, x.NameMedicine, MyDB.types.GetList().First(y => y.CodeMedicineType == x.CodeMedicineType).NameMedicineType, x.TheIdealAmauntInStock, x.UnitsInStock }).ToList();

dataGridView1.Columns[0].HeaderText = "קוד תרופה";

dataGridView1.Columns[1].HeaderText = "שם תרופה";

dataGridView1.Columns[2].HeaderText = "סוג תרופה";

dataGridView1.Columns[3].HeaderText = "כמות אידיאלית";

dataGridView1.Columns[4].HeaderText = "כמות במלאי כרגע";

button1.Visible = false;

}

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

{

label1.Text = "ע'מ לעדכן הזמנת תרופה יש ללחוץ פעמיים על השם של התרופה המבוקשת";

status = 44;

dataGridView1.DataSource = MyDB.medicines.GetList().Where(x => x.UnitsInStock < x.TheIdealAmauntInStock / 2).Select(x => new { x.CodeMedicine, x.NameMedicine, MyDB.types.GetList().First(y => y.CodeMedicineType == x.CodeMedicineType).NameMedicineType, x.TheIdealAmauntInStock, x.UnitsInStock }).ToList();

dataGridView1.Columns[0].HeaderText = "קוד תרופה";

dataGridView1.Columns[1].HeaderText = "שם תרופה";

dataGridView1.Columns[2].HeaderText = "סוג תרופה";

dataGridView1.Columns[3].HeaderText = "כמות אידיאלית";

dataGridView1.Columns[4].HeaderText = "כמות במלאי כרגע";

button1.Visible = false;

}

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

{

label1.Text = "רשימת התרופות";

status = 4;

dataGridView1.DataSource = MyDB.medicines.GetList().Select(x => new { x.CodeMedicine, x.NameMedicine, MyDB.types.GetList().First(y => y.CodeMedicineType == x.CodeMedicineType).NameMedicineType, x.TheIdealAmauntInStock, x.UnitsInStock }).ToList();

dataGridView1.Columns[0].HeaderText = "קוד תרופה";

dataGridView1.Columns[1].HeaderText = "שם תרופה";

dataGridView1.Columns[2].HeaderText = "סוג תרופה";

dataGridView1.Columns[3].HeaderText = "כמות אידיאלית";

dataGridView1.Columns[4].HeaderText = "כמות במלאי כרגע";

button1.Visible = false;

}

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

{

MyShifts shifts = new MyShifts();

this.Hide();

shifts.ShowDialog();

this.Show();

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (medicine != null)

{

if (z == 5)

z++;

else

{

DialogResult r = MessageBox.Show("האם ברצונך להגדיר את התרופה " + comboBox1.SelectedItem + " כתחליף לתרופה " + medicine.NameMedicine, "הגדרת תרופה חלופית", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (r == DialogResult.Yes)

{

Medicine mm = MyDB.medicines.GetList().First(x => x.NameMedicine == comboBox1.SelectedItem.ToString());

foreach (var item in MyDB.medicineelder.GetList())

{

if (medicine.CodeMedicine == item.CodeMedicine)

item.CodeMedicine = mm.CodeMedicine;

}

MyDB.medicineelder.SaveChanges();

MyDB.medicines.DeleteItem(medicine);

MyDB.medicines.SaveChanges();

MessageBox.Show("המחיקה נגמרה בהצלחה");

comboBox1.Visible = false;

label2.Visible = false;

label1.Visible = true;

dataGridView1.Visible = true;

menuStrip1.Visible = true;

label1.Text = "רשימת התרופות";

status = 4;

dataGridView1.DataSource = MyDB.medicines.GetList().Select(x => new { x.CodeMedicine, x.NameMedicine, MyDB.types.GetList().First(y => y.CodeMedicineType == x.CodeMedicineType).NameMedicineType, x.TheIdealAmauntInStock, x.UnitsInStock }).ToList();

dataGridView1.Columns[0].HeaderText = "קוד תרופה";

dataGridView1.Columns[1].HeaderText = "שם תרופה";

dataGridView1.Columns[2].HeaderText = "סוג תרופה";

dataGridView1.Columns[3].HeaderText = "כמות אידיאלית";

dataGridView1.Columns[4].HeaderText = "כמות במלאי כרגע";

button1.Visible = false;

}

}

}

}

}

}

עמוד הוספת ועדכון פרטי הצוות:

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;

using Nursing\_Home.BLL;

namespace Nursing\_Home.GUI

{

public partial class AddDN : Form

{

FormStatus f;

bool isboctor;

Nurse n;

Doctor d;

bool flag = true;

public AddDN(Doctor d)

{

InitializeComponent();

f = FormStatus.update;

isboctor = true;

this.d = d;

label1.Text = "עדכון פרטי הרופא/ה " + d.NameDoctor;

this.Text= " עדכון פרטי הרופא/ה" + d.NameDoctor;

textBox1.Text = d.IdDoctor;

textBox2.Text = d.NameDoctor;

textBox3.Text = d.LicenseNumber;

textBox4.Text = d.PoneNumber;

textBox5.Text = d.Address;

dateTimePicker1.Value = d.DateOfBirth;

}

public AddDN(Nurse n)

{

InitializeComponent();

f = FormStatus.update;

isboctor = false;

this.n = n;

label1.Text = "עדכון פרטי האח/ות " + n.NameNurse;

this.Text = "עדכון פרטי האח/ות " + n.NameNurse;

textBox3.Visible = false;

label4.Visible = false;

btnAddShift.Visible = true;

textBox1.Text = n.IdNurse;

textBox2.Text = n.NameNurse;

textBox4.Text = n.PhoneNumber;

textBox5.Text = n.Address;

dateTimePicker1.Value = n.DateOfBirth;

}

public AddDN(bool isdoctor)

{

InitializeComponent();

f = FormStatus.add;

if (isdoctor)

{

label1.Text = "אנא הכנס את פרטי הרופא/ה";

this.Text = "הוספת רופא חדש למאגר ";

button1.Text = "לאישור פרטי הרופא/ה";

this.isboctor = true;

d = new Doctor();

}

else

{

label1.Text = "אנא הכנס את פרטי האח/ות";

this.Text = "הוספת אח חדש למאגר ";

button1.Text = "לאישור פרטי האח/ות";

label4.Visible = false;

textBox3.Visible = false;

this.isboctor = false;

n = new Nurse();

btnAddShift.Visible = false;

}

}

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

{

errorProvider1.Clear();

flag = true;

FillDitails();

if (flag)

{

if (f == FormStatus.add)

{

if (isboctor)

{

MyDB.doctors.AddItem(d);

MyDB.doctors.SaveChanges();

MessageBox.Show("הרופא/ה התקבל/ה בהצלחה!");

this.Close();

}

if (!isboctor)

{

if (f == FormStatus.add)

{

MyDB.nurses.AddItem(n);

MyDB.nurses.SaveChanges();

MessageBox.Show("האח/ות התקבל/ה בהצלחה");

}

DialogResult r = MessageBox.Show("האם ברצונך להגדיר כעת משמרות עבור אח זה?","הוספת אח למאגר", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if(r==DialogResult.Yes)

{

MyShifts frmshift = new MyShifts(n, true);

this.Hide();

frmshift.ShowDialog();

this.Show();

}

else

{

this.Close();

}

}

}

if (f == FormStatus.update)

{

if (isboctor)

{

MyDB.doctors.UpDataItem(d);

MyDB.doctors.SaveChanges();

MessageBox.Show("השינויים נשמרו בהצלחה!");

this.Close();

}

if (!isboctor)

{

MyDB.nurses.UpDataItem(n);

MyDB.nurses.SaveChanges();

MessageBox.Show("השינויים נשמרו בהצלחה!");

this.Close();

}

}

}

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Validation.IsHebrew(e.KeyChar.ToString()) && e.KeyChar != '\b')

{

e.Handled = true;

}

if(e.KeyChar.ToString()==" ")

{

e.Handled = false;

}

}

private void textBox3\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Validation.IsNum(e.KeyChar.ToString()) && e.KeyChar != '\b')

{

e.Handled = true;

}

}

private void FillDitails()

{

errorProvider1.Clear();

flag = true;

if (isboctor)

{

if (f == FormStatus.add)

{

Doctor doctor = MyDB.doctors.GetList().FirstOrDefault(x => x.IdDoctor == textBox1.Text);

if(doctor!=null)

{

flag = false;

errorProvider1.SetError(textBox1, ".ת.ז. זו קימת כבר במאגר, אין אפשרות להכניס רופא נוסף עם מספר זהות זה,");

}

else if (Validation.CheckId(textBox1.Text))

{

d.IdDoctor = textBox1.Text;

}

else

{

flag = false;

errorProvider1.SetError(textBox1, "תעודת הזהות אינה תקינה");

}

}

d.NameDoctor = textBox2.Text;

if (textBox3.Text.Length == 7)

{

d.LicenseNumber = textBox3.Text;

}

else

{

flag = false;

errorProvider1.SetError(textBox3, "מספר רישיון לא חוקי");

}

if (Validation.IsTel(textBox4.Text) || Validation.IsPelepon(textBox4.Text))

{

d.PoneNumber = textBox4.Text;

}

else

{

flag = false;

errorProvider1.SetError(textBox4, "מספר הטלפון לא תקין");

}

d.Address = textBox5.Text;

d.DateOfBirth = dateTimePicker1.Value;

}

if (!isboctor)

{

Nurse nurse = MyDB.nurses.GetList().FirstOrDefault(x => x.IdNurse == textBox1.Text);

if (f == FormStatus.add)

{

if (nurse != null)

{

flag = false;

errorProvider1.SetError(textBox1, ".ת.ז. זו קימת כבר במאגר, אין אפשרות להכניס אח נוסף עם מספר זהות זה,");

}

else if (Validation.CheckId(textBox1.Text))

{

n.IdNurse = textBox1.Text;

}

else

{

flag = false;

errorProvider1.SetError(textBox1, "תעודת הזהות אינה תקינה");

}

}

n.NameNurse = textBox2.Text;

if (Validation.IsTel(textBox4.Text) || Validation.IsPelepon(textBox4.Text))

{

n.PhoneNumber = textBox4.Text;

}

else

{

flag = false;

errorProvider1.SetError(textBox4, "מספר הטלפון לא תקין");

}

n.Address = textBox5.Text;

n.DateOfBirth = dateTimePicker1.Value;

}

}

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

{

MyShifts frmshift = new MyShifts(n, true);

this.Hide();

frmshift.ShowDialog();

this.Show();

}

}

}

עמוד הוספת ועדכון פרטי הדיירים:

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;

using Nursing\_Home.BLL;

namespace Nursing\_Home.GUI

{

public partial class AddElder : Form

{

FormStatus f;

Elder el;

bool flag = true;

public AddElder()

{

InitializeComponent();

f = FormStatus.add;

comboBox1.DataSource = MyDB.doctors.GetList().Select(x => x.NameDoctor).ToList();

el = new Elder();

this.Text = "הוספת דייר למאגר";

}

public AddElder(Elder el)

{

InitializeComponent();

f = FormStatus.update;

comboBox1.DataSource = MyDB.doctors.GetList().Select(x => x.NameDoctor).ToList();

button1.Text = "שמור שינויים";

this.el = el;

this.Text = "עדכון פרטי הדייר " + el.NameElder;

textBox1.Text = el.IdElder;

textBox2.Text = el.NameElder;

comboBox1.SelectedItem = MyDB.doctors.GetList().First(x => x.IdDoctor == el.IdDoctor).NameDoctor;

textBox3.Text = el.HospitalizationRoom.ToString();

dateTimePicker1.Value = el.DateOfBirth;

textBox4.Text = el.NameContact;

textBox5.Text = el.PhoneContact;

textBox6.Text = el.ProximityContact;

}

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

{

if(f==FormStatus.add)

{

FillDitailies();

if (flag)

{

MyDB.elders.AddItem(el);

MyDB.elders.SaveChanges();

MessageBox.Show("הדייר/ת התקבל/ה בהצלחה");

this.Close();

}

}

if(f==FormStatus.update)

{

FillDitailies();

if (flag)

{

MyDB.elders.UpDataItem(el);

MyDB.elders.SaveChanges();

MessageBox.Show("השינויים נשמרו בהצלחה!");

this.Close();

}

}

}

public void FillDitailies()

{

flag = true;

errorProvider1.Clear();

if (f == FormStatus.add)

{

Elder elder = MyDB.elders.GetList().FirstOrDefault(x => x.IdElder == textBox1.Text);

if (elder != null)

{

flag = false;

errorProvider1.SetError(textBox1, ".ת.ז. זו קימת כבר במאגר, אין אפשרות להכניס דייר נוסף עם מספר זהות זה,");

}

else if (Validation.CheckId(textBox1.Text))

{

el.IdElder = textBox1.Text;

}

else

{

errorProvider1.SetError(textBox1, "ת.ז. אינה תקינה");

flag = false;

}

}

el.NameElder = textBox2.Text;

el.DateOfBirth = dateTimePicker1.Value;

if(Convert.ToInt32(textBox3.Text)!=el.HospitalizationRoom)

if (Convert.ToInt32(textBox3.Text) < 99 || Convert.ToInt32(textBox3.Text) > 400)

{

errorProvider1.SetError(textBox3, "מספר החדר לא תקין");

flag = false;

el.HospitalizationRoom = Convert.ToInt32(textBox3.Text);

}

else

{

Elder e;

e = MyDB.elders.GetList().FirstOrDefault(x => x.HospitalizationRoom == Convert.ToInt32(textBox3.Text));

if (e != null)

{

errorProvider1.SetError(textBox3, "החדר מאוכלס כבר, אנא בחר חדר אישפוז אחר");

flag = false;

}

else

{

el.HospitalizationRoom = Convert.ToInt32(textBox3.Text);

}

}

el.IdDoctor = MyDB.doctors.GetList().First(x => x.NameDoctor == comboBox1.SelectedItem.ToString()).IdDoctor;

el.NameContact = textBox4.Text;

if(Validation.IsPelepon(textBox5.Text)||Validation.IsTel(textBox5.Text))

{

el.PhoneContact = textBox5.Text;

}

else

{

errorProvider1.SetError(textBox5, "מספר הטלפון אינו תקין");

flag = false;

}

el.ProximityContact = textBox6.Text;

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Validation.IsHebrew(e.KeyChar.ToString()) && e.KeyChar != '\b')

{

e.Handled = true;

}

if (e.KeyChar.ToString() == " ")

{

e.Handled = false;

}

}

private void textBox4\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Validation.IsHebrew(e.KeyChar.ToString()) && e.KeyChar != '\b')

{

e.Handled = true;

}

if (e.KeyChar.ToString() == " ")

{

e.Handled = false;

}

}

private void textBox6\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Validation.IsHebrew(e.KeyChar.ToString()) && e.KeyChar != '\b')

{

e.Handled = true;

}

if (e.KeyChar.ToString() == " ")

{

e.Handled = false;

}

}

}

}

עמוד מחיקת רופא:

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;

using Nursing\_Home.BLL;

namespace Nursing\_Home.GUI

{

public partial class DeleteDoctor : Form

{

List<Elder> elderlist = new List<Elder>();

Doctor d;

public DeleteDoctor(Doctor d)

{

InitializeComponent();

this.d = d;

label2.Text += d.NameDoctor + " מרשימת הרופאים ";

elderlist = MyDB.elders.GetList().Where(x => x.IdDoctor == d.IdDoctor).ToList();

comboBox1.DataSource = MyDB.doctors.GetList().Where(x => x.IdDoctor != d.IdDoctor).Select(x=>x.NameDoctor).ToList();

comboBox1.SelectedItem = null;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (comboBox1.SelectedItem != null)

{

if (elderlist.Count != 0)

{

label3.Text = elderlist.First().NameElder;

elderlist.First().IdDoctor = MyDB.doctors.GetList().FirstOrDefault(x => x.NameDoctor == comboBox1.SelectedItem.ToString()).IdDoctor;

MyDB.elders.UpDataItem(elderlist.First());

MyDB.elders.SaveChanges();

elderlist.Remove(elderlist.First());

}

else

{

MyDB.doctors.DeleteItem(d);

MyDB.doctors.SaveChanges();

MessageBox.Show("מחיקת הרופא מהמאגר נשלמה");

this.Close();

}

}

}

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

{

this.Close();

}

}

}

דף הוספת תרופות:

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;

using Nursing\_Home.BLL;

namespace Nursing\_Home.GUI

{

public partial class AddMedicine : Form

{

FormStatus f;

Medicine m;

public AddMedicine()

{

InitializeComponent();

f = FormStatus.add;

m = new Medicine();

this.Text = "הוספת תרופה למאגר";

comboBox1.DataSource = MyDB.types.GetList().Select(x => x.NameMedicineType).ToList();

}

public AddMedicine(Medicine m)

{

InitializeComponent();

f = FormStatus.update;

this.m = m;

button1.Text = "שמור שינויים";

this.Text = "עדכון פרטי התרופה";

textBox1.Text = m.NameMedicine;

comboBox1.SelectedItem = MyDB.types.GetList().First(d=>d.CodeMedicineType==m.CodeMedicineType).NameMedicineType.ToList();

numericUpDown1.Value = m.TheIdealAmauntInStock;

numericUpDown2.Value = m.UnitsInStock;

}

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

{

if (f == FormStatus.add)

{

m.CodeMedicine = MyDB.medicines.ThisCode();

m.NameMedicine = textBox1.Text;

m.CodeMedicineType = MyDB.types.GetList().First(x => x.NameMedicineType == comboBox1.SelectedItem.ToString()).CodeMedicineType;

m.TheIdealAmauntInStock = Convert.ToInt32(numericUpDown1.Value);

m.UnitsInStock = Convert.ToInt32(numericUpDown2.Value);

MyDB.medicines.AddItem(m);

MyDB.medicines.SaveChanges();

MessageBox.Show("התרופה נשמרה בהצלחה");

this.Close();

}

if (f == FormStatus.update)

{

m.NameMedicine = textBox1.Text;

m.CodeMedicineType = MyDB.types.GetList().First(x => x.NameMedicineType == comboBox1.SelectedItem.ToString()).CodeMedicineType;

m.TheIdealAmauntInStock = Convert.ToInt32(numericUpDown1.Value);

m.UnitsInStock = Convert.ToInt32(numericUpDown2.Value);

MyDB.medicines.UpDataItem(m);

MyDB.medicines.SaveChanges();

MessageBox.Show("השינויים נשמרו בהצלחה!");

this.Close();

}

}

}

}

דף עדכון כמות תרופות חסרות:

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;

using Nursing\_Home.BLL;

namespace Nursing\_Home.GUI

{

public partial class UpdateOrderingMedicine : Form

{

Medicine m;

public UpdateOrderingMedicine(Medicine m)

{

InitializeComponent();

this.m = m;

label1.Text += m.NameMedicine + " ואח'כ לחץ על אישור";

this.Text = "עדכון כמות התרופה " + m.NameMedicine + " במלאי";

}

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

{

m.UnitsInStock = m.UnitsInStock + Convert.ToInt32(textBox1.Text);

MyDB.medicines.UpDataItem(m);

MyDB.medicines.SaveChanges();

this.Close();

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Validation.IsNum(e.KeyChar.ToString()) && e.KeyChar != '\b')

{

e.Handled = true;

}

}

}

}

עמוד הרופא:

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;

using Nursing\_Home.BLL;

namespace Nursing\_Home.GUI

{

public partial class DoctorsPage : Form

{

Doctor d;

public DoctorsPage(Doctor d)

{

InitializeComponent();

this.d = new Doctor();

this.d = d;

label1.Text = d.NameDoctor;

dataGridView1.DataSource = MyDB.elders.GetList().Where(x => x.IdDoctor == d.IdDoctor).ToList();

dataGridView1.Columns[0].HeaderText = "מספר ת.ז. ";

dataGridView1.Columns[1].HeaderText = "שם המטופל";

dataGridView1.Columns[2].HeaderText = "תאריך לידה";

dataGridView1.Columns[3].HeaderText = "חדש איפוז";

dataGridView1.Columns[4].Visible = false;

dataGridView1.Columns[5].HeaderText = "מספר איש הקשר";

dataGridView1.Columns[6].HeaderText = "שם איש הקשר";

dataGridView1.Columns[7].HeaderText = "קרבה משפחתית";

this.Text = "עמוד הרופא " + d.NameDoctor;

}

private void CellClickElder(object sender, DataGridViewCellEventArgs e)

{

Elder eld = new Elder();

foreach (var item in MyDB.elders.GetList())

{

if (item.IdElder == dataGridView1.CurrentRow.Cells[0].Value.ToString())

eld = item;

}

ElderPage f = new ElderPage(eld,d);

this.Hide();

f.ShowDialog();

this.Show();

dataGridView1.DataSource = MyDB.elders.GetList().Where(x => x.IdDoctor == d.IdDoctor).ToList();

}

}

}

עמוד תיק רפואי:

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;

using Nursing\_Home.BLL;

namespace Nursing\_Home.GUI

{

public partial class ElderPage : Form

{

Elder el;

Doctor d;

public ElderPage(Elder e,Doctor d)

{

InitializeComponent();

label1.Text = e.NameElder;

label2.Text = "ת.ז.: " + e.IdElder;

label3.Text = "תאריך לידה: " + e.DateOfBirth.ToShortDateString();

label4.Text = "חדר אישפוז: " + e.HospitalizationRoom;

this.el = e;

this.d = d;

dataGridView1.DataSource = MyDB.medicineelder.GetList().Where(x => x.IdElder == e.IdElder&&x.EndDate.Date>=DateTime.Today).Select(x=> new {x.AutomaticCode,MyDB.medicines.GetList().First(y=>y.CodeMedicine==x.CodeMedicine).NameMedicine,x.StartDate.Date,x.EndDate,x.TimesADay,x.NumberOfMilimetersAtADay }).ToList();

dataGridView1.Columns[0].Visible = false;

dataGridView1.Columns[1].HeaderText = "שם התרופה";

dataGridView1.Columns[2].HeaderText = "תאריך הנטילה הראשונה";

dataGridView1.Columns[3].HeaderText = "תאריך הנטילה האחרונה";

dataGridView1.Columns[4].HeaderText = "מספר הפעמים ביום";

dataGridView1.Columns[5].HeaderText = "מספר המילימטרים בפעם";

this.Text = "תיק רפואי: " + e.NameElder;

}

private void button2\_Click(object sender, EventArgs e)

{

Medicines\_Of\_Elder m = MyDB.medicineelder.GetList().FirstOrDefault(x => x.AutomaticCode == Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value));

DialogResult r = MessageBox.Show("האם ברצונך למחוק את " + m.NameMedicine(m.CodeMedicine) + "לזקן: "+ m.NameElder(m.IdElder),"מחיקת תרופה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);//איך מסדרים את זה

if (r == DialogResult.Yes)

{

MyDB.medicineelder.DeleteItem(m);

MyDB.medicineelder.SaveChanges();

dataGridView1.DataSource = MyDB.medicineelder.GetList().Where(x => x.IdElder == el.IdElder && x.EndDate.Date >= DateTime.Today).Select(x => new { x.AutomaticCode, MyDB.medicines.GetList().First(y => y.CodeMedicine == x.CodeMedicine).NameMedicine, x.StartDate.Date, x.EndDate, x.TimesADay, x.NumberOfMilimetersAtADay }).ToList();

dataGridView1.Columns[0].Visible = false;

dataGridView1.Columns[1].HeaderText = "שם התרופה";

dataGridView1.Columns[2].HeaderText = "תאריך הנטילה הראשונה";

dataGridView1.Columns[3].HeaderText = "תאריך הנטילה האחרונה";

dataGridView1.Columns[4].HeaderText = "מספר הפעמים ביום";

dataGridView1.Columns[5].HeaderText = "מספר המילימטרים בפעם";

}

}

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

{

AddMedicineOfElder a = new AddMedicineOfElder(d,el);

this.Hide();

a.ShowDialog();

this.Show();

dataGridView1.DataSource = MyDB.medicineelder.GetList().Where(x => x.IdElder == el.IdElder && x.EndDate.Date >= DateTime.Today).Select(x => new { x.AutomaticCode, MyDB.medicines.GetList().First(y => y.CodeMedicine == x.CodeMedicine).NameMedicine, x.StartDate.Date, x.EndDate, x.TimesADay, x.NumberOfMilimetersAtADay }).ToList();

dataGridView1.Columns[0].Visible = false;

dataGridView1.Columns[1].HeaderText = "שם התרופה";

dataGridView1.Columns[2].HeaderText = "תאריך הנטילה הראשונה";

dataGridView1.Columns[3].HeaderText = "תאריך הנטילה האחרונה";

dataGridView1.Columns[4].HeaderText = "מספר הפעמים ביום";

dataGridView1.Columns[5].HeaderText = "מספר המילימטרים בפעם";

}

private void button3\_Click\_1(object sender, EventArgs e)

{

if (button3.Text == "לצפיה בכל ההיסטוריה הרפואית")

{

dataGridView1.DataSource = MyDB.medicineelder.GetList().Where(x => x.IdElder == el.IdElder).Select(x => new { x.AutomaticCode, MyDB.medicines.GetList().First(y => y.CodeMedicine == x.CodeMedicine).NameMedicine, x.StartDate.Date, x.EndDate, x.TimesADay, x.NumberOfMilimetersAtADay }).ToList();

dataGridView1.Columns[0].Visible = false;

dataGridView1.Columns[1].HeaderText = "שם התרופה";

dataGridView1.Columns[2].HeaderText = "תאריך הנטילה הראשונה";

dataGridView1.Columns[3].HeaderText = "תאריך הנטילה האחרונה";

dataGridView1.Columns[4].HeaderText = "מספר הפעמים ביום";

dataGridView1.Columns[5].HeaderText = "מספר המילימטרים בפעם";

button3.Text = "לחזרה";

}

else

{

dataGridView1.DataSource = MyDB.medicineelder.GetList().Where(x => x.IdElder == el.IdElder && x.EndDate.Date >= DateTime.Today).Select(x => new { x.AutomaticCode, MyDB.medicines.GetList().First(y => y.CodeMedicine == x.CodeMedicine).NameMedicine, x.StartDate.Date, x.EndDate, x.TimesADay, x.NumberOfMilimetersAtADay }).ToList();

dataGridView1.Columns[0].Visible = false;

dataGridView1.Columns[1].HeaderText = "שם התרופה";

dataGridView1.Columns[2].HeaderText = "תאריך הנטילה הראשונה";

dataGridView1.Columns[3].HeaderText = "תאריך הנטילה האחרונה";

dataGridView1.Columns[4].HeaderText = "מספר הפעמים ביום";

dataGridView1.Columns[5].HeaderText = "מספר המילימטרים בפעם";

button3.Text = "לצפיה בכל ההיסטוריה הרפואית";

}

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

Medicines\_Of\_Elder c = MyDB.medicineelder.GetList().FirstOrDefault(x => x.AutomaticCode == Convert.ToInt32(dataGridView1.CurrentRow.Cells[0].Value));

if (c != null)

{

AddMedicineOfElder f = new AddMedicineOfElder(c);

this.Hide();

f.ShowDialog();

this.Show();

}

dataGridView1.DataSource = MyDB.medicineelder.GetList().Where(x => x.IdElder == el.IdElder && x.EndDate.Date >= DateTime.Today).Select(x => new { x.AutomaticCode, MyDB.medicines.GetList().First(y => y.CodeMedicine == x.CodeMedicine).NameMedicine, x.StartDate.Date, x.EndDate, x.TimesADay, x.NumberOfMilimetersAtADay }).ToList();

dataGridView1.Columns[0].Visible = false;

dataGridView1.Columns[1].HeaderText = "שם התרופה";

dataGridView1.Columns[2].HeaderText = "תאריך הנטילה הראשונה";

dataGridView1.Columns[3].HeaderText = "תאריך הנטילה האחרונה";

dataGridView1.Columns[4].HeaderText = "מספר הפעמים ביום";

dataGridView1.Columns[5].HeaderText = "מספר המילימטרים בפעם";

}

}

}

עמוד רישום תרופה למטופל:

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;

using Nursing\_Home.BLL;

namespace Nursing\_Home.GUI

{

public partial class AddMedicineOfElder : Form

{

FormStatus f =new FormStatus();

Medicines\_Of\_Elder med;

bool flag = true;

string d;

string e;

public AddMedicineOfElder(Doctor d,Elder e)

{

InitializeComponent();

f = FormStatus.add;

comboBox1.DataSource = MyDB.types.GetList().Select(x => x.NameMedicineType).ToList();

label2.Visible = false;

comboBox2.Visible = false;

med = new Medicines\_Of\_Elder();

this.d = d.IdDoctor;

this.e = e.IdElder;

this.Text = "רישום תרופה למטופל " + e.NameElder;

}

public AddMedicineOfElder(Medicines\_Of\_Elder m)

{

InitializeComponent();

f = FormStatus.update;

med = m;

d = m.IdDoctor;

e = m.IdElder;

button1.Text = "עדכן";

comboBox1.DataSource = MyDB.types.GetList().Select(x => x.NameMedicineType).ToList();

numericUpDown1.Value = m.TimesADay;

textBox1.Text = m.NumberOfMilimetersAtADay.ToString();

dateTimePicker1.Value = m.StartDate;

string p= (med.EndDate.Date - med.StartDate.Date).ToString();

string q = "";

bool b = true;

for (int i = 0; i <p.Length&&b; i++)

{

if (p[i] != '.')

q += p[i];

else

b = false;

}

txtnumdays.Text =q;

label2.Visible = true;

comboBox2.Visible = true;

comboBox2.SelectedItem = MyDB.medicines.GetList().First(z => z.CodeMedicine == m.CodeMedicine).NameMedicine;

comboBox1.SelectedItem = MyDB.types.GetList().First(x => x.CodeMedicineType == (MyDB.medicines.GetList().First(y => y.CodeMedicine == m.CodeMedicine)).CodeMedicineType).NameMedicineType;

this.Text = "עדכון תרופה למטופל " + MyDB.elders.GetList().First(x => x.IdElder == m.IdElder).NameElder;

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!Validation.IsNum(e.KeyChar.ToString()) && e.KeyChar != '\b')

{

e.Handled = true;

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

Types\_Of\_Medicines t;

t = MyDB.types.GetList().First(x => x.NameMedicineType == comboBox1.SelectedItem.ToString());

comboBox2.Visible = true;

comboBox2.DataSource = MyDB.medicines.GetList().Where(x => x.CodeMedicineType == t.CodeMedicineType).Select(x => x.NameMedicine).ToList();

label2.Visible = true;

}

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

{

if (f == FormStatus.add)

{

FillMedicineOfElder();

if (flag)

{

MyDB.medicineelder.AddItem(med);

MyDB.medicineelder.SaveChanges();

AddDaysForMedicine();

MessageBox.Show("רישום התרופה התבצע בהצלחה!");

this.Close();

}

}

if(f==FormStatus.update)

{

FillMedicineOfElder();

if (flag)

{

foreach (var item in MyDB.medicinestake.GetList())

{

if(item.Status=false&&item.AutomaticCode==med.AutomaticCode)

{

MyDB.medicinestake.DeleteItem(item);

MyDB.medicinestake.SaveChanges();

}

}

MyDB.medicineelder.UpDataItem(med);

MyDB.medicineelder.SaveChanges();

AddDaysForMedicine();

MessageBox.Show("השינויים נשמרו בהצלחה!");

this.Close();

}

}

}

private void AddDaysForMedicine()

{

int numd =Convert.ToInt32(txtnumdays.Text);

DateTime dt = med.StartDate;

Medicine\_Take mt = new Medicine\_Take();

mt.AutomaticCode = med.AutomaticCode;

mt.Status = false;

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

{

mt.InDate = dt;

if (med.TimesADay==1)

{

mt.Code = MyDB.medicinestake.ThisCode();

mt.InShift = 1;

MyDB.medicinestake.AddItem(mt);

MyDB.medicinestake.SaveChanges();

}

if(med.TimesADay==2)

{

mt.Code = MyDB.medicinestake.ThisCode();

mt.InShift = 1;

MyDB.medicinestake.AddItem(mt);

MyDB.medicinestake.SaveChanges();

mt.Code = MyDB.medicinestake.ThisCode();

mt.InShift = 3;

MyDB.medicinestake.AddItem(mt);

MyDB.medicinestake.SaveChanges();

}

if(med.TimesADay==3)

{

mt.Code = MyDB.medicinestake.ThisCode();

mt.InShift = 1;

MyDB.medicinestake.AddItem(mt);

MyDB.medicinestake.SaveChanges();

mt.Code = MyDB.medicinestake.ThisCode();

mt.InShift = 2;

MyDB.medicinestake.AddItem(mt);

MyDB.medicinestake.SaveChanges();

mt.Code = MyDB.medicinestake.ThisCode();

mt.InShift = 3;

MyDB.medicinestake.AddItem(mt);

MyDB.medicinestake.SaveChanges();

}

dt=dt.AddDays(1.0);

}

}

public void FillMedicineOfElder()

{

flag = true;

errorProvider1.Clear();

if(f==FormStatus.add)

med.AutomaticCode = MyDB.medicineelder.ThisCode();

med.CodeMedicine = MyDB.medicines.GetList().First(x => x.NameMedicine == comboBox2.SelectedItem.ToString()).CodeMedicine;

med.IdElder = e;

med.IdDoctor = d;

med.StartDate = dateTimePicker1.Value;

if (txtnumdays.Text!="")

{

med.EndDate = dateTimePicker1.Value.AddDays(Convert.ToInt32(txtnumdays.Text));

}

else

{

errorProvider1.SetError(txtnumdays, "מספר הימים לנטילת התרופה לא הוגדר");

flag = false;

}

med.TimesADay = Convert.ToInt32(numericUpDown1.Value);

if (textBox1.Text != "")

{

med.NumberOfMilimetersAtADay = Convert.ToInt32(textBox1.Text);

}

else

{

errorProvider1.SetError(textBox1, "מספר המיליליטרים לא הוגדר");

flag = false;

}

}

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

{

if (!Validation.IsNum(e.KeyChar.ToString()) && e.KeyChar != '\b')

{

e.Handled = true;

}

}

}

}

עמוד אח:

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;

using Nursing\_Home.BLL;

namespace Nursing\_Home.GUI

{

public partial class NursesPage : Form

{

Nurse n;

int shift = 0;

List<Treat> t = new List<Treat>();

public NursesPage(Nurse n)

{

InitializeComponent();

this.n = n;

label1.Text = "שלום ל" + n.NameNurse;

this.Text = "כניסה למשמרת";

}

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

{

MyShifts msh = new MyShifts(n);

this.Hide();

msh.ShowDialog();

this.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

if(textBox1.Text!="1"&&textBox1.Text!="2"&&textBox1.Text!="3")

{

errorProvider1.SetError(textBox1, "מספר קומה שגוי");

textBox1.Text = "";

}

else

{

errorProvider1.Clear();

button1.Visible = true;

tub.Visible = true;

dataGridViewElders.DataSource = MyDB.elders.GetList().Select(x => new { x.IdElder,x.NameElder, x.DateOfBirth.Date, x.HospitalizationRoom, MyDB.doctors.GetList().First(y=>y.IdDoctor==x.IdDoctor).NameDoctor,x.PhoneContact,x.NameContact,x.ProximityContact}).Where(x => x.HospitalizationRoom.ToString()[0] == textBox1.Text[0]).ToList();

dataGridViewElders.Columns[0].HeaderText = "תעודת זהות";

dataGridViewElders.Columns[1].HeaderText = "שם";

dataGridViewElders.Columns[2].HeaderText = "תאריך לידה";

dataGridViewElders.Columns[3].HeaderText = "חדר אישפוז";

dataGridViewElders.Columns[4].HeaderText = "הרופא המטפל";

dataGridViewElders.Columns[5].HeaderText = "מספר הקשר";

dataGridViewElders.Columns[6].HeaderText = "שם איש קשר";

dataGridViewElders.Columns[7].HeaderText = "קרבה משפחתית";

List<Elder> el = MyDB.elders.GetList().Where(x => x.HospitalizationRoom.ToString()[0] == textBox1.Text[0]).ToList();

if (DateTime.Now.Hour>=7&&DateTime.Now.Hour<14)

shift = 1;

else if (DateTime.Now.Hour >= 14 && DateTime.Now.Hour < 21)

shift = 2;

else

shift = 3;

List<Medicine\_Take> mt =MyDB.medicinestake.GetList().Where(x=>x.Status==false&&x.InShift==shift&&x.InDate.Date==DateTime.Today&&MyDB.elders.GetList().First(z=>z.IdElder== MyDB.medicineelder.GetList().First(y=>y.AutomaticCode==x.AutomaticCode).IdElder).HospitalizationRoom.ToString()[0]==textBox1.Text[0]).ToList();

t.Clear();

Treat treat = new Treat();

foreach (var item in mt)

{

treat = new Treat();

treat.c = item.Code;

treat.code = MyDB.medicineelder.GetList().First(y => y.AutomaticCode == item.AutomaticCode).CodeMedicine;

treat.namem = MyDB.medicines.GetList().First(z => z.CodeMedicine == MyDB.medicineelder.GetList().First(y => y.AutomaticCode == item.AutomaticCode).CodeMedicine).NameMedicine;

treat.nummili = MyDB.medicineelder.GetList().First(y => y.AutomaticCode == item.AutomaticCode).NumberOfMilimetersAtADay;

treat.namee = MyDB.elders.GetList().First(a => a.IdElder == MyDB.medicineelder.GetList().First(y => y.AutomaticCode == item.AutomaticCode).IdElder).NameElder;

treat.id = MyDB.medicineelder.GetList().First(y => y.AutomaticCode == item.AutomaticCode).IdElder;

treat.room = MyDB.elders.GetList().First(a => a.IdElder == MyDB.medicineelder.GetList().First(y => y.AutomaticCode == item.AutomaticCode).IdElder).HospitalizationRoom;

t.Add(treat);

treat = null;

}

dataGridViewMedicines.DataSource = t;

dataGridViewMedicines.Columns[0].HeaderText = "קוד הטיפול";

dataGridViewMedicines.Columns[1].HeaderText = "קוד תרופה";

dataGridViewMedicines.Columns[2].HeaderText = "שם תרופה";

dataGridViewMedicines.Columns[3].HeaderText = "הכמות במילימטרים";

dataGridViewMedicines.Columns[4].HeaderText = "שם המטופל";

dataGridViewMedicines.Columns[5].HeaderText = "תעודת זהות";

dataGridViewMedicines.Columns[6].HeaderText = "חדר אישפוז";

button2.Visible = false;

textBox1.Visible = false;

label2.Visible = false;

label1.Text = n.NameNurse;

this.Text = "משמרת ";

if (shift == 1)

this.Text += "בוקר";

else if (shift == 2)

this.Text += "צהוריים";

else

this.Text += "ערב";

this.Text += " של: " + n.NameNurse;

}

}

private void dataGridViewMedicines\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

DialogResult r=MessageBox.Show("האם נתת את התרופה " + dataGridViewMedicines.CurrentRow.Cells[2].Value + " לזקן " + dataGridViewMedicines.CurrentRow.Cells[4].Value, "נטילת תרופה", MessageBoxButtons.YesNo);

if (r == DialogResult.Yes)

{

//שינוי הסטטוס והסרה מהרשימה

Medicine\_Take m = MyDB.medicinestake.GetList().First(x => x.Code == Convert.ToInt32(dataGridViewMedicines.CurrentRow.Cells[0].Value));

m.Status = true;

MyDB.medicinestake.UpDataItem(m);

MyDB.medicinestake.SaveChanges();

Treat treat = new Treat();

treat = t.First(x => x.c == m.Code);

t.Remove(treat);

if (t.Count != 0)

{

dataGridViewMedicines.DataSource = t;

}

else

dataGridViewMedicines.Visible = false;

//הפחתת התרופה מהמלאי

Medicine medicine = MyDB.medicines.GetList().First(x => x.CodeMedicine == treat.code);

medicine.UnitsInStock = medicine.UnitsInStock - treat.nummili;

MyDB.medicines.UpDataItem(medicine);

MyDB.medicines.SaveChanges();

}

}

}

}

דף משמרות אישי:

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;

using Nursing\_Home.BLL;

namespace Nursing\_Home.GUI

{

public partial class MyShifts : Form

{

Nurse n;

public MyShifts(int shift)

{

InitializeComponent();

label1.Text = "מפת משמרות האחים";

this.Text = "דף משמרות, מנהל";

label5.Visible = true;

label6.Visible = true;

flowLayoutPanel1.Controls.Clear();

for (int i = 0; i < 7; i++)

{

UCshift shiftControl = new UCshift(i + 1);

flowLayoutPanel1.Controls.Add(shiftControl);

}

groupBox1.Visible = true;

string ss;

if (shift % 10 == 1)

ss = "בוקר";

else if (shift % 10 == 2)

ss = "צהוריים";

else

ss = "לילה";

groupBox1.Text = "משמרת " + ss + " ביום " + MyDB.arrdays[shift / 10 - 1];

Shift s = MyDB.shifts.GetList().First(x => x.CodeShift == shift);

if (s.Nnurses != 0)

{

label7.Visible = true;

label7.Text += s.Nnurses.ToString() + " אחים/ות";

}

List<Employee\_Shifts> lst = new List<Employee\_Shifts>();

lst = MyDB.shiftsemp.GetList().Where(x => x.CodeShift == shift).ToList();

if(lst.Count!=0)

{

label9.Visible = true;

label9.Text = MyDB.nurses.GetList().First(x => x.IdNurse == lst[0].CodeEmployee).NameNurse;

lst.Remove(lst[0]);

}

if (lst.Count != 0)

{

label10.Visible = true;

label10.Text = MyDB.nurses.GetList().First(x => x.IdNurse == lst[0].CodeEmployee).NameNurse;

lst.Remove(lst[0]);

}

if (lst.Count != 0)

{

label11.Visible = true;

label11.Text = MyDB.nurses.GetList().First(x => x.IdNurse == lst[0].CodeEmployee).NameNurse;

lst.Remove(lst[0]);

}

if (lst.Count != 0)

{

label12.Visible = true;

label12.Text = MyDB.nurses.GetList().First(x => x.IdNurse == lst[0].CodeEmployee).NameNurse;

lst.Remove(lst[0]);

}

if (lst.Count != 0)

{

label13.Visible = true;

label13.Text = MyDB.nurses.GetList().First(x => x.IdNurse == lst[0].CodeEmployee).NameNurse;

lst.Remove(lst[0]);

}

if (lst.Count != 0)

{

label14.Visible = true;

label14.Text = MyDB.nurses.GetList().First(x => x.IdNurse == lst[0].CodeEmployee).NameNurse;

lst.Remove(lst[0]);

}

}

public MyShifts()

{

InitializeComponent();

label1.Text = "מפת משמרות האחים";

this.Text = "דף משמרות, מנהל";

label5.Visible = true;

label6.Visible = true;

flowLayoutPanel1.Controls.Clear();

label15.Visible = false;

for (int i = 0; i < 7; i++)

{

UCshift shiftControl = new UCshift(i+1);

flowLayoutPanel1.Controls.Add(shiftControl);

}

}

public MyShifts(Nurse n)

{

InitializeComponent();

this.n = new Nurse();

this.n = n;

label1.Text = "משמרות עבור: " + n.NameNurse;

this.Text="דף המשמרות של "+n.NameNurse;

ShowYoman();

double sum = 0;

foreach (var item in MyDB.shiftsemp.GetList().Where(x=>x.CodeEmployee==n.IdNurse))

{

sum += 8 \* MyDB.shifts.GetList().First(y => y.CodeShift == item.CodeShift).PaymentForHour;

}

label15.Text = "שכר לחודש(ארבעה שבועות) הוא " + sum+" שקלים";

}

public MyShifts(Nurse n, bool f)

{

InitializeComponent();

this.n = new Nurse();

button1.Visible = true;

this.n = n;

label1.Text = "משמרות אופציונליות עבור: " + n.NameNurse;

this.Text = "דף המשמרות החסירות";

AddShiftForNurs();

}

private void ShowYoman()

{

flowLayoutPanel1.Controls.Clear();

for (int i = 0; i < 7; i++)

{

//להצגת משמרות שולח לפעולה בונה

UCshift shiftControl = new UCshift(n,Convert.ToInt32(i+1), MyDB.arrdays[i]);

flowLayoutPanel1.Controls.Add(shiftControl);

}

}

private void AddShiftForNurs()

{

flowLayoutPanel1.Controls.Clear();

for (int i = 0; i < 7; i++)

{

//שולח לפעולה בונה להוספת משמרות

UCshift shiftControl = new UCshift(n, i+1, MyDB.arrdays[i], true);

flowLayoutPanel1.Controls.Add(shiftControl);

}

}

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

{

Close();

}

}

}

UCshift:

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;

using Nursing\_Home.BLL;

using Nursing\_Home.GUI;

namespace Nursing\_Home.GUI

{

public partial class UCshift : UserControl

{

Nurse n1;

List<Employee\_Shifts> lstempsh;

List<Shift> lstsh;

Shift s;

Employee\_Shifts emp;

int thisShift;

int shift;

int dayN;

public UCshift(int i)

{

InitializeComponent();

this.label2.Text = MyDB.arrdays[i-1];

lstsh = new List<Shift>();

lstsh = MyDB.shifts.GetList().Where(x => x.Nnurses > 0 && x.DayInWeek == i).ToList();

s = null;

while (lstsh.Count != 0)

{

foreach (var item in this.Controls)

{

if (s != null)

{

lstsh.Remove(s);

s = null;

}

if (item is TextBox)

{

TextBox tx = item as TextBox;

tx.Click += Messege\_Click;

//לחפש את התג של הטקסטבוקס הנוכחי- איזה מס' משמרת ביום

shift = 10 \* i + Convert.ToInt32(tx.Tag);

//בדוק אם האח עובד ביום זה ובשעה זו

s = lstsh.FirstOrDefault(x => x.CodeShift == shift);

if (s != null)

{

//לצבוע את המשבצת

tx.BackColor = Color.Yellow;

}

}

}

}

}

//פעולה בונה המקבלת יום בשבוע ואח ומציגה משמרות שלו

public UCshift(Nurse n, int d, string named)

{

InitializeComponent();

lstempsh = new List<Employee\_Shifts>();

n1 = n;

dayN = d;

this.label2.Text = named;

//רשימת משמרות לאח זה

lstempsh = MyDB.shiftsemp.GetList().Where(x => x.CodeEmployee == n.IdNurse&& x.CodeShift/10==d).ToList();

emp = null;

while(lstempsh.Count!=0)

{

foreach (var item in this.Controls)

{

if (emp != null)

{

lstempsh.Remove(emp);

emp = null;

}

if (item is TextBox)

{

TextBox tx = item as TextBox;

shift = d \* 10 + Convert.ToInt32(tx.Tag);

//בדוק אם האח עובד ביום זה ובשעה זו

emp = lstempsh.FirstOrDefault(x => x.CodeShift == shift);

if (emp != null)

{

//לצבוע את המשבצת

tx.BackColor = Color.Yellow;

}

}

}

}

}

//פעולה המציגה את המשמרות שניתן עדין לשבץ בהם אחים

//ומציגה כמה אחים חסרים בכל משמרת בשביל שיבוץ מקסימלי

public UCshift(Nurse n, int d, string named, bool t)

{

InitializeComponent();

n1 = n;

dayN = d;

this.label2.Text = named;

lstsh = new List<Shift>();

lstsh = MyDB.shifts.GetList().Where(x => x.Nnurses > 0 && x.DayInWeek == d).ToList();

s = null;

while (lstsh.Count!=0)

{

foreach (var item in this.Controls)

{

if(s!=null)

{

lstsh.Remove(s);

s = null;

}

if (item is TextBox)

{

TextBox tx = item as TextBox;

tx.Click += Tx\_Click;

//לחפש את התג של הטקסטבוקס הנוכחי- איזה מס' משמרת ביום

shift =10\*d+ Convert.ToInt32(tx.Tag);

//בדוק אם האח עובד ביום זה ובשעה זו

s = lstsh.FirstOrDefault(x => x.CodeShift == shift);

if (s != null)

{

//לצבוע את המשבצת

tx.BackColor = Color.Yellow;

tx.Text = s.Nnurses.ToString();

}

}

}

}

}

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

{

TextBox tx = sender as TextBox;

s = new Shift();

s = MyDB.shifts.GetList().First(x => x.CodeShift == 10 \* dayN + Convert.ToInt32(tx.Tag));

emp = new Employee\_Shifts();

thisShift = s.CodeShift;

emp = MyDB.shiftsemp.GetList().FirstOrDefault(x => x.CodeEmployee == n1.IdNurse && x.CodeShift == s.CodeShift);

if (emp != null)

{

DialogResult d = MessageBox.Show("אח זה רשום כבר במשמרת שנבחרה, האם ברצונך למחוק אותו משם", "שאלה", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (d == DialogResult.Yes)

{

MyDB.shiftsemp.DeleteItem(emp);

MyDB.shiftsemp.SaveChanges();

MessageBox.Show(n1.NameNurse + " הוסר/ה מהמשמרת שנבחרה");

s.Nnurses++;

MyDB.shifts.UpDataItem(s);

MyDB.shifts.SaveChanges();

tx.Text = s.Nnurses.ToString();

if (s.Nnurses!=0)

tx.BackColor = Color.Yellow;

}

}

else if (s.Nnurses > 0)

{

int shift1, shift2;

if (s.CodeShift % 10 == 2)

{

shift1 = s.CodeShift - 1;

shift2 = s.CodeShift + 1;

}

else if (s.CodeShift % 10 == 1)

{

if (s.CodeShift / 10 == 1)

shift1 = 73;

else

shift1 = s.CodeShift - 8;

shift2 = s.CodeShift + 1;

}

else

{

shift1 = s.CodeShift - 1;

if (s.CodeShift / 10 == 7)

shift2 = 11;

else

shift2 = s.CodeShift + 8;

}

if (MyDB.shiftsemp.GetList().FirstOrDefault(x => x.CodeEmployee == n1.IdNurse && x.CodeShift == shift1) != null)

{

MessageBox.Show("אח זה משובץ כבר במשמרת הקודמת לזו שנבחרה. שיבוץ בשתי משמרות רצופות אינו חוקי");

}

else if (MyDB.shiftsemp.GetList().FirstOrDefault(x => x.CodeEmployee == n1.IdNurse && x.CodeShift == shift2) != null)

{

MessageBox.Show("אח זה משובץ כבר במשמרת שאחרי המשמרת שנבחרה. שיבוץ בשתי משמרות רצופות אינו חוקי");

}

else

{

DialogResult d = MessageBox.Show("האם תרצה להוסיף את ?" + n1.NameNurse + " למשמרת שנבחרה", "שאלה", MessageBoxButtons.YesNo);

if (d == DialogResult.Yes)

{

AddShiftForNurs();

s.Nnurses--;

MyDB.shifts.UpDataItem(s);

MyDB.shifts.SaveChanges();

tx.Text = s.Nnurses.ToString();

if (s.Nnurses == 0)

tx.BackColor = Color.White;

}

}

}

else

{

emp = MyDB.shiftsemp.GetList().FirstOrDefault(x => x.CodeEmployee == n1.IdNurse && x.CodeShift == s.CodeShift);

if (emp == null)

MessageBox.Show("משמרת זו מלאה, אין אפשרות להוסיף אח נוסף למשמרת זו.");

}

}

private void AddShiftForNurs()

{

emp = new Employee\_Shifts();

emp.CodeShiftEm = MyDB.shiftsemp.EmployeeShiftCode();

emp.CodeEmployee = n1.IdNurse;

emp.CodeShift = thisShift;

MyDB.shiftsemp.AddItem(emp);

MyDB.shiftsemp.SaveChanges();

MessageBox.Show("המשמרת נוספה בהצלחה, תוכלו להוסיף עכשיו עוד משמרות או לחזור לטופס הקודם");

}

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

{

int day=0, shift;

for (int i=0; i<7; i++)

{

if (MyDB.arrdays[i] == label2.Text)

{

day = i + 1;

}

}

TextBox tx = sender as TextBox;

shift = Convert.ToInt32(tx.Tag);

MyShifts myShifts = new MyShifts(day \* 10 + shift);

myShifts.Show();

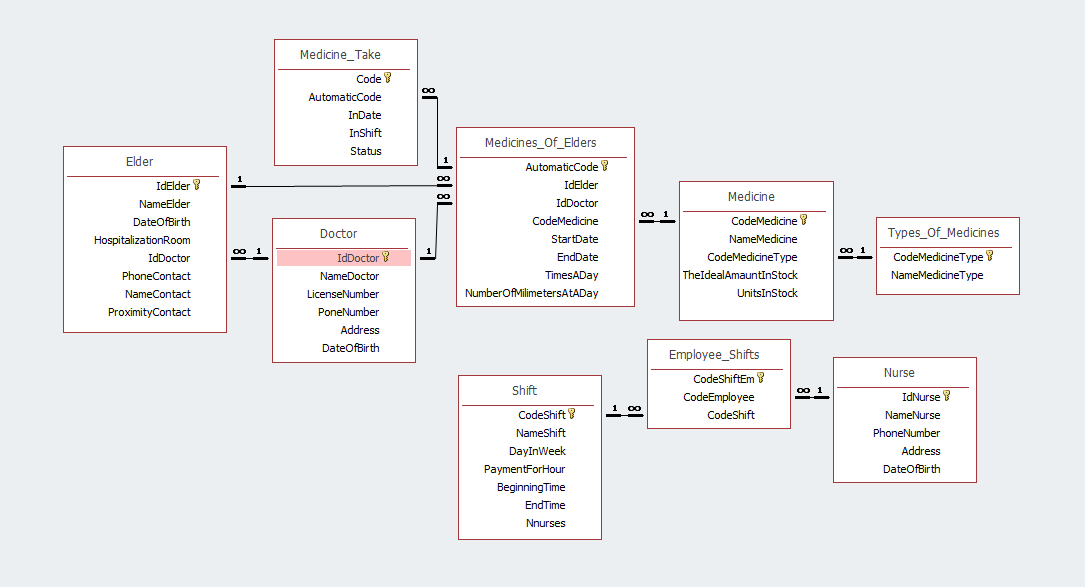
}

} }

**מסמך הגדרת דרישות**

ניתוח נתונים:

תיאור מסד נתונים

****

|  |  |  |
| --- | --- | --- |
| **טבלת רופאים/ות** | | |
| שם באנגלית | שם בעברית | סוג |
| IdDoctor | תעודת זהות | טקסט קצר |
| NameDoctor | שם | טקסט קצר |
| LicenseNumber | מספר רישיון | טקסט קצר |
| PoneNumber | מספר פלאפון | טקסט קצר |
| Address | כתובת | טקסט קצר |
| DateOfBirth | תאריך לידה | תאריך/שעה |

|  |  |  |
| --- | --- | --- |
| **טבלת דיירים/ות** | | |
| שם באנגלית | שם בעברית | סוג |
| IdElder | תעודת זהות | טקסט קצר |
| NameElder | שם | טקסט קצר |
| DateOfBirth | תאריך לידה | תאריך/שעה |
| HospitalizationRoom | חדר אשפוז | מספר |
| IdDoctor | ת.ז. רופא | טקסט קצר |
| PhoneContact | מספר איש קשר | טקסט קצר |
| NameContact | שם איש קשר | טקסט קצר |
| ProximityContact | קרבת איש קשר | טקסט קצר |

|  |  |  |
| --- | --- | --- |
| **טבלת אחים/ות** | | |
| שם באנגלית | שם בעברית | סוג |
| IdNurse | תעודת זהות | טקסט קצר |
| NameNurse | שם | טקסט קצר |
| PhoneNumber | מספר פלאפון | טקסט קצר |
| Address | כתובת | טקסט קצר |
| DateOfBirth | תאריך לידה | תאריך/שעה |

|  |  |  |
| --- | --- | --- |
| **טבלת תרופות** | | |
| שם באנגלית | שם בעברית | סוג |
| CodeMedicine | קוד תרופה | מספר |
| NameMedicine | שם תרופה | טקסט קצר |
| CodeMedicineType | סוג התרופה | מספר |
| TheIdealAmauntInStock | הכמות האידיאלית במלאי | מספר |
| UnitsInStock | הכמות הנוכחית במלאי | מספר |

|  |  |  |
| --- | --- | --- |
| **טבלת תרופות לדיירים** | | |
| שם באנגלית | שם בעברית | סוג |
| AutomaticCode | קוד אוטומטי | מספר |
| IdElder | ת.ז. דייר | טקסט קצר |
| IdDoctor | ת.ז. רופא | טקסט קצר |
| CodeMedicine | קוד תרופה | מספר |
| StartDate | תאריך ההתחלה | תאריך/שעה |
| EndDate | תאריך הסיום | תאריך/שעה |
| TimesADay | מספר הפעמים ביום | מספר |
| NumberOfMilimetersAtADay | מספר המיליליטרים בפעם | מספר |

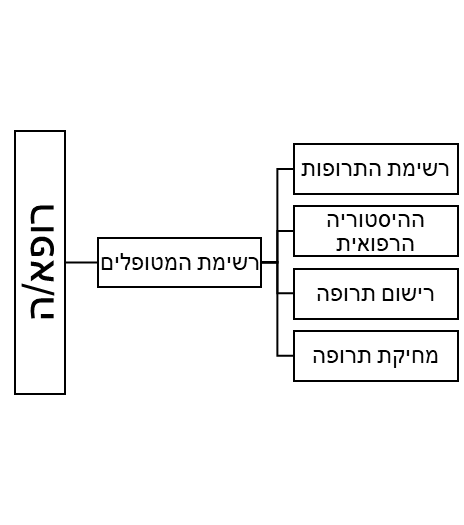
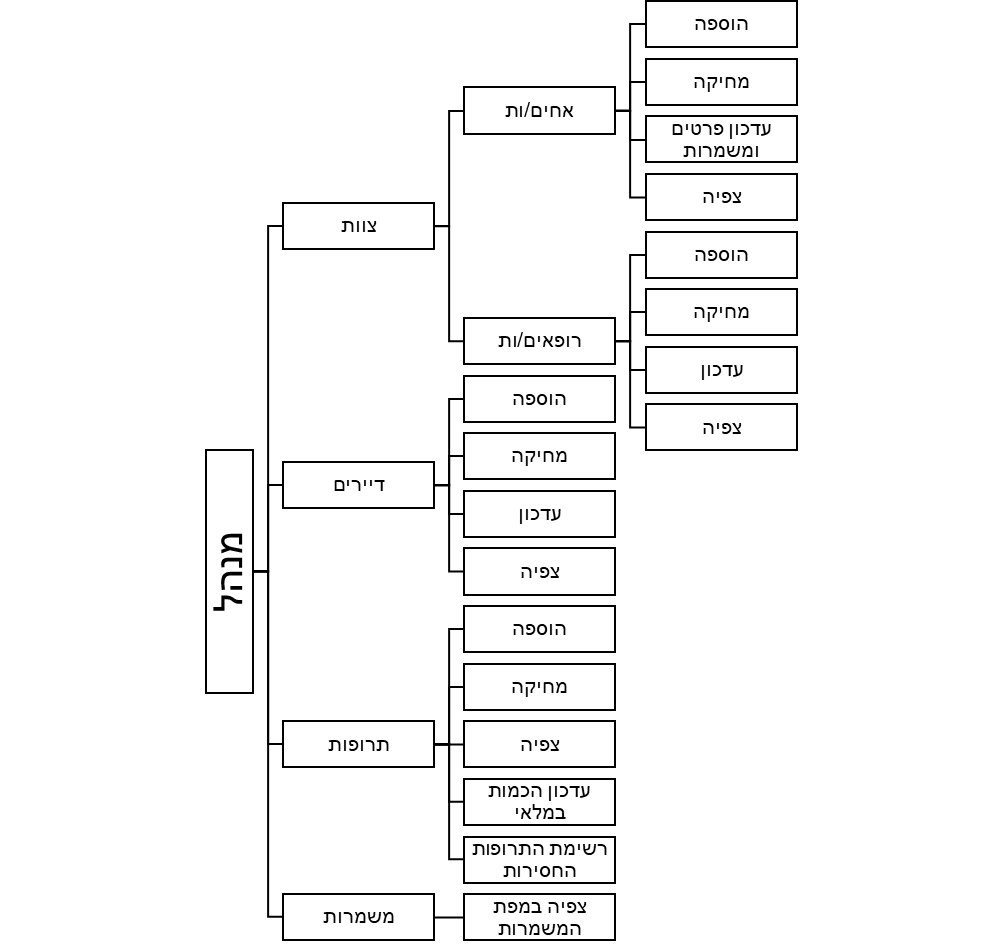
|  |  |  |
| --- | --- | --- |
| **טבלת נטילת תרופה** | | |
| שם באנגלית | שם בעברית | סוג |
| Code | קוד אוטומטי | מספר |
| AutomaticCode | קוד (מהתרופות לזקנים) | מספר |
| InDate | בתאריך | תאריך/שעה |
| InShift | במשמרת | מספר |
| Status | מצב | כן/לא |

|  |  |  |
| --- | --- | --- |
| **טבלת סוגי תרופות** | | |
| שם באנגלית | שם בעברית | סוג |
| CodeMedicineType | קוד סוג תרופה | מספר |
| NameMedicineType | שם סוג תרופה | טקסט קצר |

|  |  |  |
| --- | --- | --- |
| **טבלת משמרות** | | |
| שם באנגלית | שם בעברית | סוג |
| CodeShift | קוד משמרת | מספר |
| NameShift | שם משמרת(בוקר צהוריים...) | מספר |
| DayInWeek | יום בשבוע | מספר |
| PaymentForHour | תשלום לשעה | מספר |
| BeginningTime | שעת התחלה | תאריך/שעה |
| EndTime | שעת סיום | תאריך/שעה |
| Nnurses | מספר אחיות נחוץ | מספר |

|  |  |  |
| --- | --- | --- |
| **טבלת משמרות לאחים** | | |
| שם באנגלית | שם בעברית | סוג |
| CodeShiftEm | קוד | מספר |
| CodeEmployee | ת.ז. עובד | טקסט קצר |
| CodeShift | קוד משמרת | מספר |

עיצוב- מדריך למשתמש

****



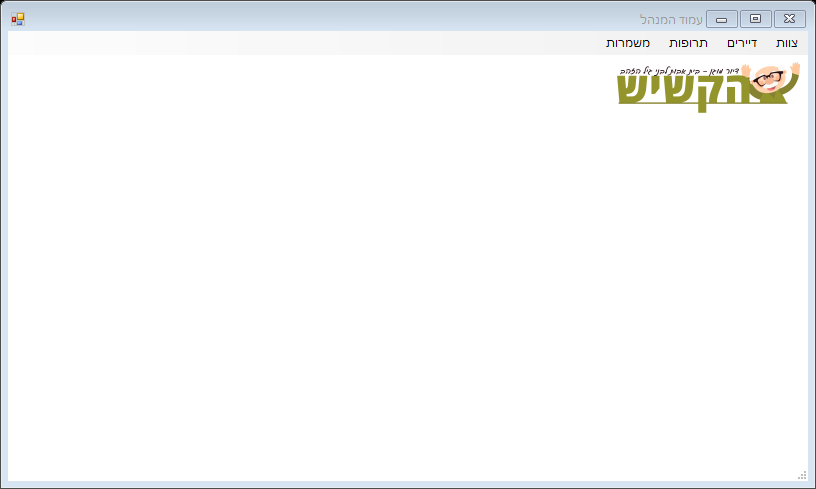
**עמוד כניסה**

כניסה לעמוד המנהל/האח/הרופא ע"י סיסמת מנהל/ תעודות זהות



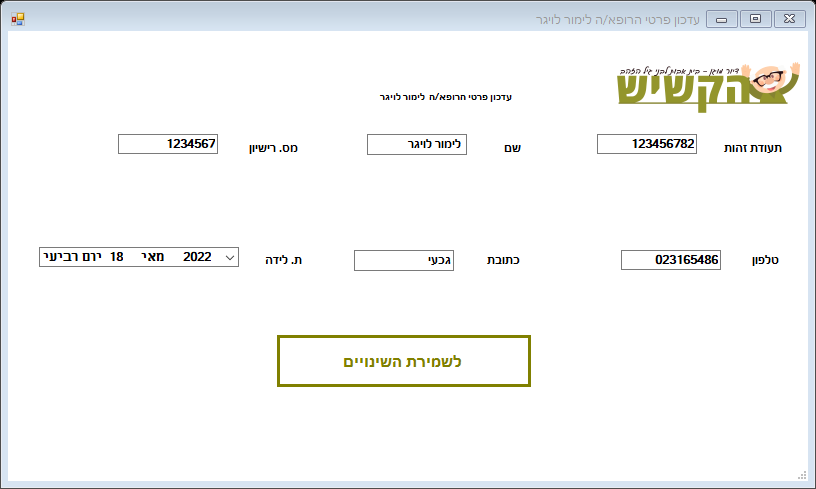
**עמוד המנהל**

ניהול הצוות הדיירים התרופות והמשמרות ע"י בחירה בתפריטים



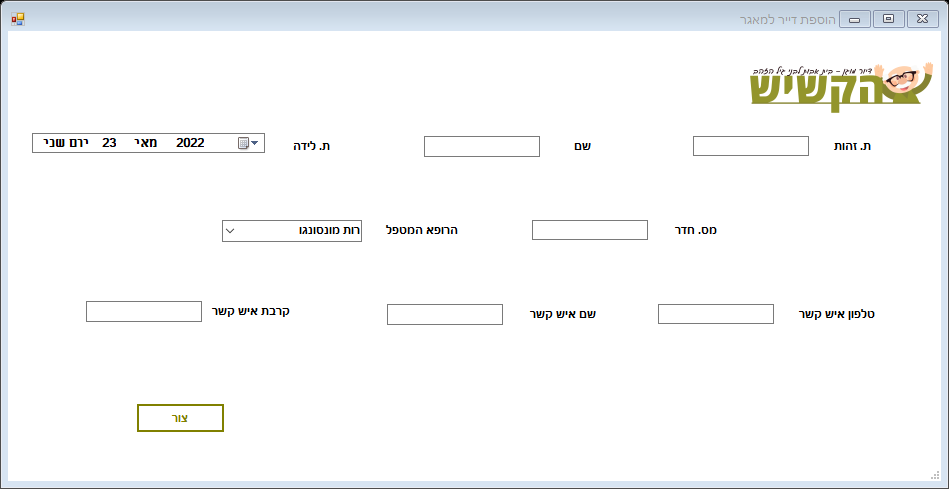
**עמוד הוספת ועדכון פרטי הצוות**

אפשרות להוסיף אחים ורופאים חדשים וכן לעדכן את פרטי הרופאים והאחים הקיימים



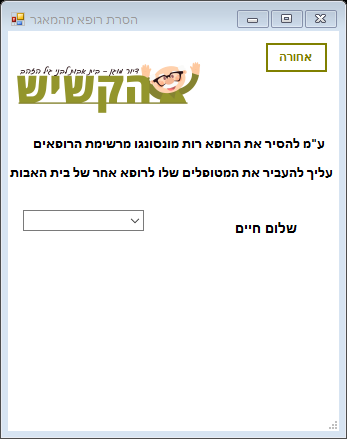
**עמוד הוספת ועדכון פרטי הדיירים**

אפשרות להוסיף דיירים חדשים וכן לעדכן את פרטי הדיירים הקיימים



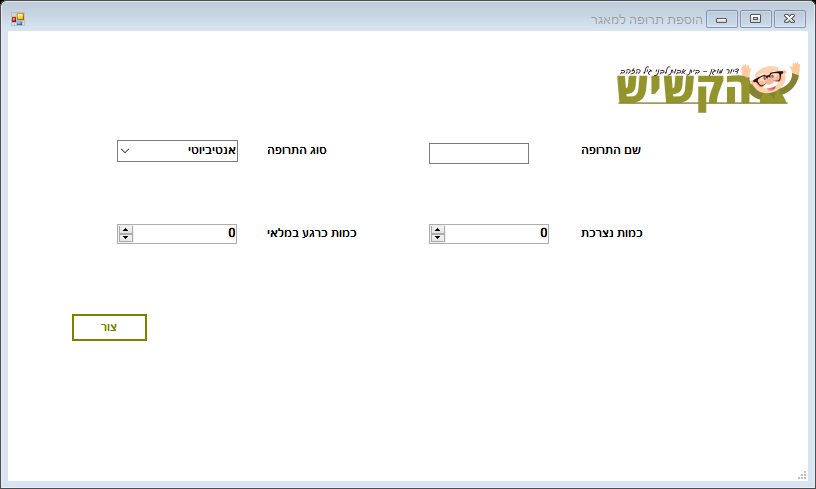
**עמוד מחיקת רופא**

כשמסירים רופא/ה מהמאגר יש צורך לעדכן קודם רופא חלופי למטופליו, התכנה מגיעה לדף זה בו צריך המשתמש להקצות רופא אחר עבור כל זקן שהיה בטיפולו של הרופא הנמחק.



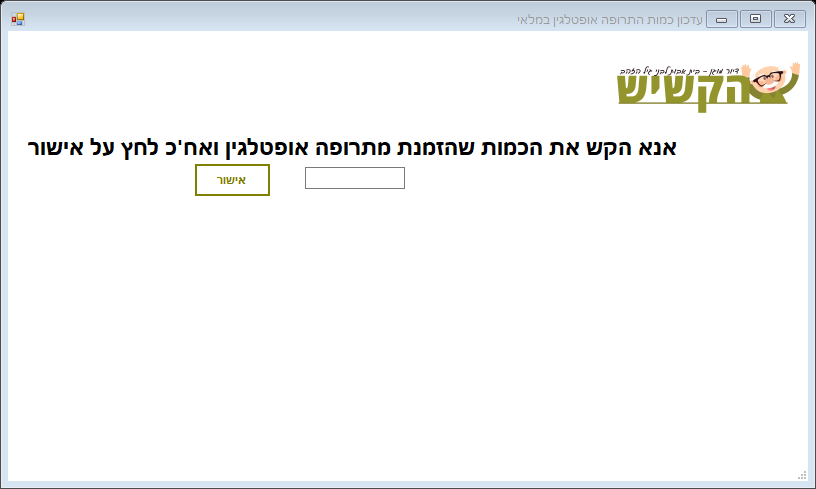
**דף הוספת תרופות**

בדף זה ישנה אפשרות להוסיף תרופות חדשות למאגר התרופות בבית האבות



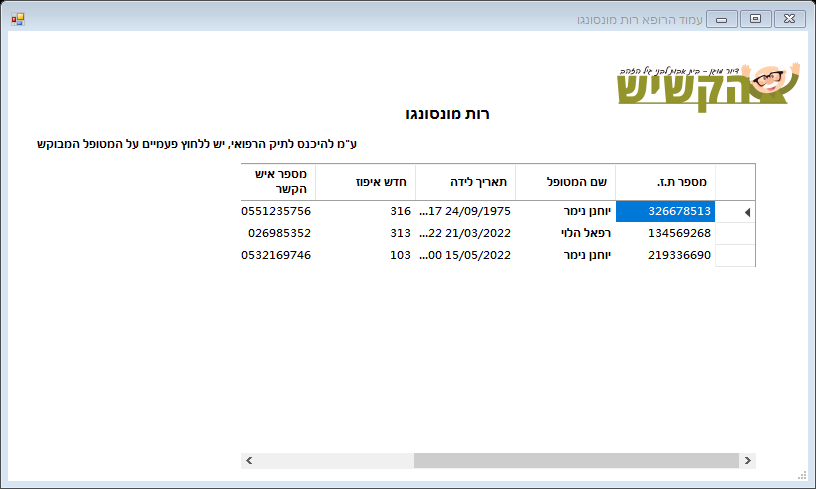
**דף עדכון כמות תרופות חסרות**

בדף זה ישנה אפשרות לעדכן במלאי תוספת של תרופות שנרכשו לאחר שאזלו או שכמותם פחתה.



**עמוד הרופא**

עמוד תצוגה של רשימת הדיירים המטופלים ע"י רופא זה, בבחירת דייר יוכלו להיכנס לתיק הרפואי שלו.



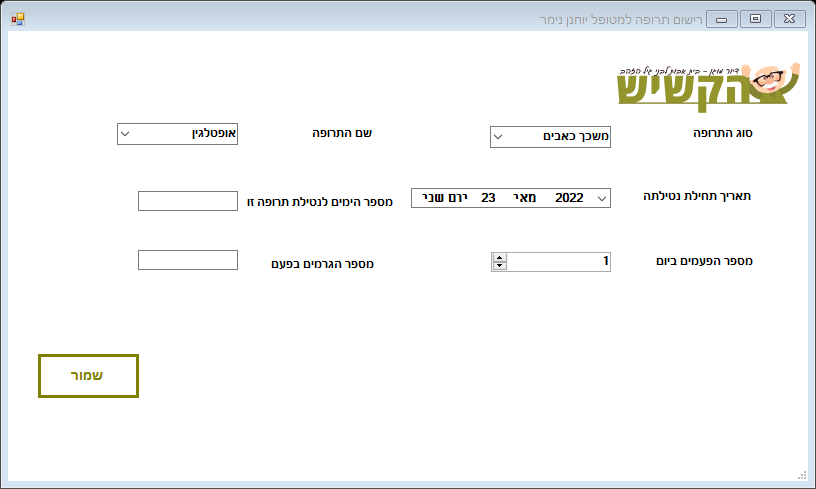
**עמוד תיק רפואי**

בעמוד תצוגה של התרופות הניטלות כרגע ע"י הזקן ואפשרות צפיה בהיסטוריה הרפואית שלו.



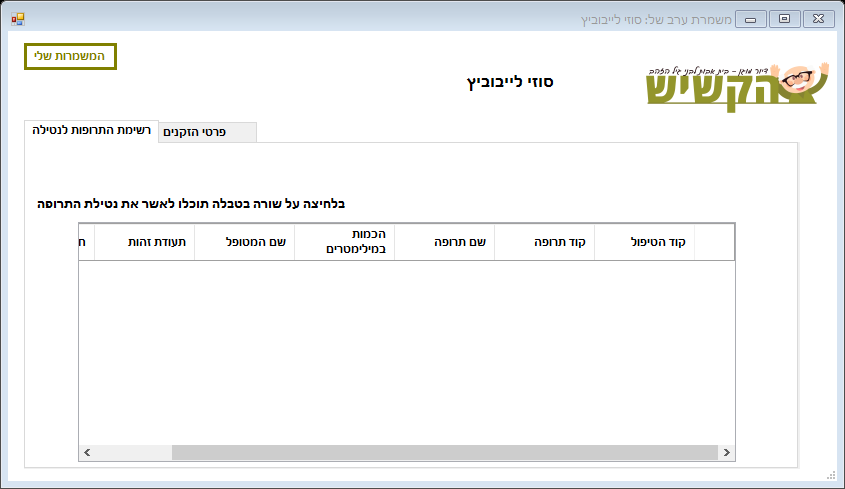
**עמוד רישום תרופה למטופל**

בעמוד אפשרות לרישום ומחיקת תרופה למטופלים השונים.



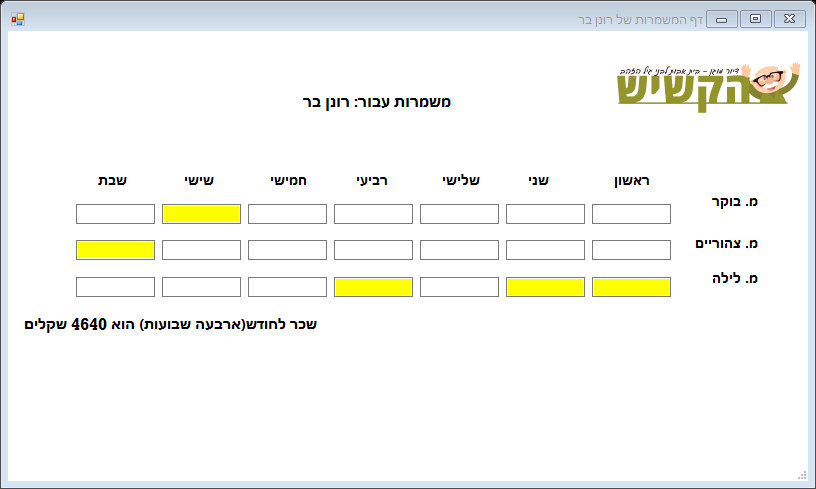
**עמוד האח**

העמוד כולל רשימת תרופות שעליו לתת במשמרת זו לזקנים הנמצאים בקומה שבה הוא עובד היום, אפשרות לאישור נתינת התרופה, וכן אפשרות של צפיה בפרטי הזקנים הדרים בקומה זו.



**דף המשמרות האישי**

בעמוד תצוגה מפורטת של מפת המשמרות כשהמשרות הרשומות על אותו אח מוזרחות וכן חישוב של משכורת חודשית ע"פ המשמרות שהוא ממלא.(הדף נפתח ג"כ אצל המנהל באפשרות של צפיה במפת המשמרות, היכן חסרים אחים ומי משובץ איפה כמו כן הדף נפתח ג"כ בהוספת אח עם אפשרות של הגדרת משמרות עבורו וכן בעדכון פרטי האחים עם אפשרות של הוספת ומחיקת משמרות עבורו)



רפלקציה

כתיבת הפרוייקט חיזקה בי את ההבנה האמיתית של חשיבות הדור המבוגר כמדריך לדור הצעיר, והחשיבות של קשר טוב עימם ודאגה אמיתית לבטחונם ובריאותם. כמו כן כתיבת התכנה ביטאה את הזהירות שאני חושבת שחייבת להנקט בטיפול בנתונים רפואיים המצריכים דיוק רב, כיוון שהטעויות שם הרות גורל.

בכל ההיבטים.WinForms במהלך כתיבת הפרוייקט העמקתי את הידיעות שלי בתוכנת הפיתוח ובניהל טפסי

העבודה עצמה הייתה ארוכה אך מעניינת ומלמדת.

כעת בסיום העבודה, אני יכולה לומר שהעבודה הזאת חידשה לי הרבה וחידדה לי כל מיני דברים שלמדנו ואני יחסית ממש מרוצה מהתוצר המוגמר.