

שם בית הספר: ישיבת בני עקיבא ראשון לציון

שם העבודה: "מלאכים בדרכים"

שם התלמיד: נריה דיינזדה

ת.ז. התלמיד: 323098616

שם המנחה: אסף לוק

שם החלופה: הנדסת תוכנה

תאריך ההגשה: 2020\06\03

# תוכן עניינים

1. מבוא ------------------------------------------------------------- 3 - 6
2. מבנה / ארכיטקטורה ----------------------------------------- 7 - 17
   1. תרשימי DFD ------------------------------------------- 7 – 11
   2. תרשימי USECASE ----------------------------------- 13 – 14
   3. רשימת שאילתות ------------------------------------------- 15
   4. תרשים UML ------------------------------------------------ 17
3. מדריך למשתמש -------------------------------------------- 18 - 21
4. בסיס הנתונים ----------------------------------------------------- 22
5. מדריך למפתח ----------------------------------------------- 23 - 24

5.1 הורדה ושימוש במערכת ------------------------------------- 24

1. רפלקציה ----------------------------------------------------------- 25
2. ביבליוגרפיה ------------------------------------------------------- 26
3. נספחים ------------------------------------------------------ 27 - 31

# מבוא

## **•תיאור תכולת הספר:**

### ספר זה יעסוק בתיאור פרויקט הגמר במסגרת מגמת הנדסת תוכנה 10 יח"ל תחת הנושא ניהול מערכות מידע. במסגרת הקורס בחרתי לייצר מערכת מידע אשר תנהל ארגון ציבורי אשר מספק שירותי דרך לרווחת הציבור בחינם, לארגון זה קוראים "מלאכים בדרכים". הספר יתאר את התהליך מרגע האיפיון והחקר שרך חשיבה על האלגוריתם ועל עיצוב המערכת ועד הבנייה של המערכת.

## **• הרקע לפרויקט:**

### הפרויקט יעסוק בשיפור מערכת המידע של ארגון "ידידים בדרכים". ארגון "ידידים בדרכים" הוא ארגון התנדבותי ללא מטרות רווח. מטרתו לספק שירותי דרכים חינם עבור אנשים אשר רכבם לא תקין. מתנדבי ארגון זה מפוזרים ברחבי הארץ, ומוכנים כל העת לבצע תיקונים פשוטים כגון החלפת גלגל, הוספת דלק וכו'. הארגון מספק עזרה שאינה רפואית 24 שעות ביממה, למעט שבתות וחגים. אסביר על קבלת ההחלטה לעשיית הפרוייקט על "ארגון ידידים" באמצעות סיפור מקרה אשר העניק מוטיבציה והבנה לביצוע הפרויקט: פעם אחת בשעה מאוחרת המכונית המשפחתית שלנו נתקעה באמצע הדרך. ידענו מיד לפנות לידידים, ענתה לנו המוקדנית ואחרי בירור מיקום קצר נפתחה קריאה למיקום שלנו. כעבור זמן קצר הגיע מתנדב ואבחן את הבעיה כחוסר שמן. אלא שאז התברר שאין לו שמן אצלו, אמרנו תודה והוא עזב. התקשרנו שוב למוקד של "ידידים", ושוב ענתה מוקדנית וכבר ידענו להגיד מראש שחסר שמן, היא פתחה קריאה ושוב הגיע מתנדב שלמרבה הצער לא היה לו את סוג השמן המתאים. לאחר שיחה שלישית, הגיע הכונן המתאים עם השמן הנכון. כל זה היה נפטר באמצעות קשר ישיר עם הכונן ובעזרת אבחון תקלה מראש. באותו יום החלטתי לשדרג ולתקן את המערכת כך שתיתן את המענה הנכון במהירות המירבית.

**• תהליך המחקר:**

### **המצב הקיים:**

### מדי שנה מטפל הארגון בלמעלה מ 140,000 פניות, והוא מונה נכון להיום (2019) כ-16000 אלף מתנדבים בפריסה כלל ארצית. חברת "נתיבי ישראל" מספקת גם כן שירותים לרכב ללא תשלום אך זו לא עיקרה של החברה, החברה עוסקת בקידום כללי של בטיחות ותרבות הנהיגה בישראל ומתן שירותים זהו חלק קטן מחברה זו. בנוסף שירותי הדרך אינם זמינים כל הזמן אלא רק 16 שעות ביממה. לאחר קריאה אודות הארגון באינטרנט ושיחה עם ארגון ידידים הבנתי את דרך פעולת הקריאות: הלקוח מתקשר למוקדנית, אשר פותחת קריאה למיקום. כל כונן שרואה את הקריאה יכול לאשר אותה ולגשת למיקום. הבעיה: לעיתים נשלח כונן לא מתאים לביצוע העבודה, ודבר זה מעכב את תיקון התקלה. לאחרונה הוציא הארגון שלוש אפליקציות חדשות: אחת לכוננים, אחת למוקדניות, ואחת ללקוחות. המוקדניות ליכולות מכל מקום לענות לשיחות של לקוחות ולפתוח קריאות. הקריאות מועברות לכוננים דרך האפליקציה לכוננים. הלקוחות יכולים לפנות הן דרך האפליקציה והן דרך המוקדניות. דרך האפליקציה יש יכולת ללקוחות לבחור את סוג הבעיה, או אחר במקרה ולא יודעים. למרות הייעול והשיפור, הבעיה נשארת בעינה: לעיתים נשלח הכונן הלא נכון לבצע את התיקון, ורק כאשר הוא מגיע למיקום מתברר לו כי אין ביכולתו לתקן.

### **חידושים בפרוייקט:**

### בפרויקט יש כמה חידושים: ראשית, הלקוח ימלא טופס בקשת טיפול בו יצטרך לענות על כמה שאלות פשוטות. המערכת תדע לזהות ולשלוח את הכונן המתאים ללא קשר למוקדניות כפי שהיה בעבר. לפי הפרוייקט שלי המוקדניות לא נצרכות.

## כמה יתרונות בדבר:

## חיסכון בכוח אדם.

## חיסכון בזמן יקר ללקוח

## התאמה מיטבית של הכונן המתאים ללקוח

## **סקירת ספרות:**

### חקרתי בחברות אחרות העוסקות בתחום:

#### אתר ארגון "ידידים בדרכים" - איך מבנה החברה אמור לפעול.

#### אתר חברת "נתיבי ישראל" - לגבי מתן שירותי דרך

#### אתר חברת "שגריר" העוסקות בנושא החילוץ בדרכים - לגבי מתן שירותי דרך

#### התייעצתי עם אחד מהמתנדבים של ארגון "ידידים בדרכים"

#### התייעצתי עם אנשים שקיבלו שירות בעבר מארגון "ידידים" – לגבי ייעול ותלונות מהשטח

#### מוסך הקרוב לביתי – לגבי התקלות הנפוצות ומתן שאלות לטפסים

#### אתר AUTO – למציאת בעיות במכונית

#### אתר ENGINE – למציאת בעיות במכונית

## **• אתגרים מרכזיים:**

### לארגון "ידידים בדרכים" היו כמה בעיות איתם התמודדתי במהלך כתיבת הפרויקט:

* 1. **שימוש יתר בכוח אדם** - כל קריאה מתחילה בקשר של הלקוח אל מוקדנית אשר נמצאת במרכזייה, כלומר רק בשנה האחרונה מוקדניות הארגון עלו ללמעלה מ141,000 שיחות.
  2. **חוסר תיאום ראשוני בין סיבת התקלה אל המתקן הנחוץ –** ישנם מקרים רבים בהם מתגלה רק בעת הגעתו של המתקן אל הדורש, שהוא לא יכול לעזור בגלל שסיבת התקלה לא תואמת את יכולותיו.
  3. **חוסר בציוד נחוץ אצל המתקן –** ישנם מקרים רבים בהם מתגלה רק בעת הגעתו של המתקן אל דורש התיקון, שחסר הציוד הנדרש לבצע תיקון**.**
  4. **אין קשר ישיר בין הלקוח למתקן** - ישנם מקרים רבים בהם דורש התיקון הסתדר בכוחות עצמו או על ידי עזרה מאחרים או מקרים רבים בהם המתקן לא מגיע אל הלקוח מסיבות שונות.

### הפרויקט מספק פתרון לבעיות הנ"ל:

#### המוקדניות ניתנות להחלפה על ידי מערכת ממוחשבת.

#### המערכת מתאימה בעצמה את הכונן על ידי חיזוי מוקדם של התקלה בעזרת שאלון אותו מילא הלקוח.

1. *קשר ישיר בין הכונן ללקוח*

## מבנה / ארכיטקטורה של הפרויקט

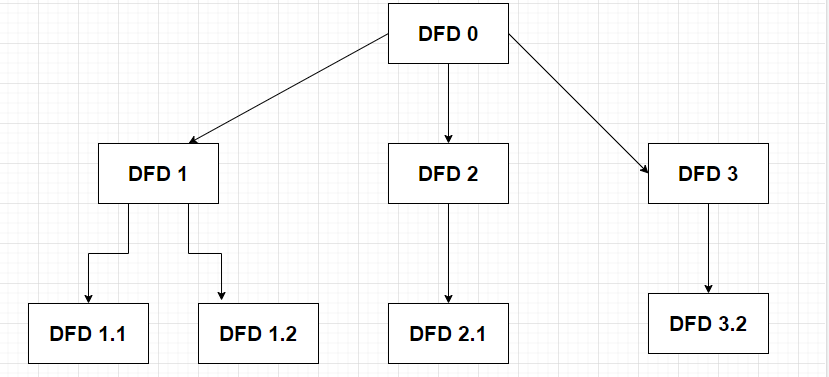
* הארכיטקטורה בפורמט ניהול מערכות מידע. הצרכים של המערכת הם: המערכת נצרכת למערכת הפעלה מסוג WINDOWS, סביבת עבודה – VISUAL STUDIO , שרת SQL ע"י תוכנת Microsoft SQL Server Management Studio 18 .

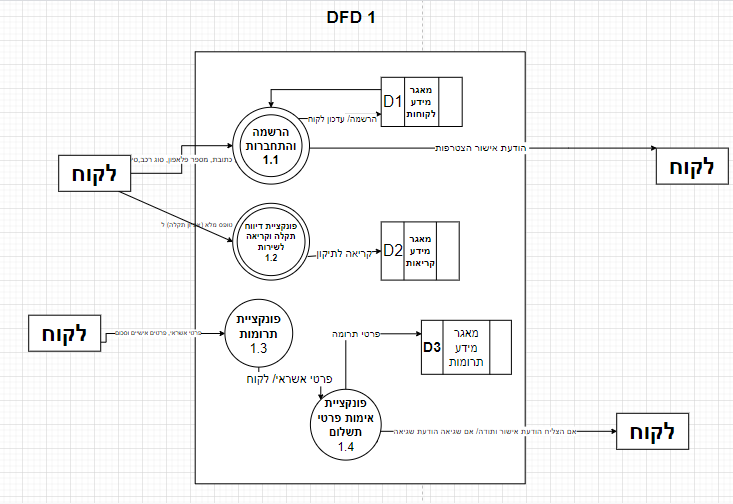
קוד המערכת כתובה בשפת # C(NET.) . מערכת המידע כתובה בSERVER SQL.

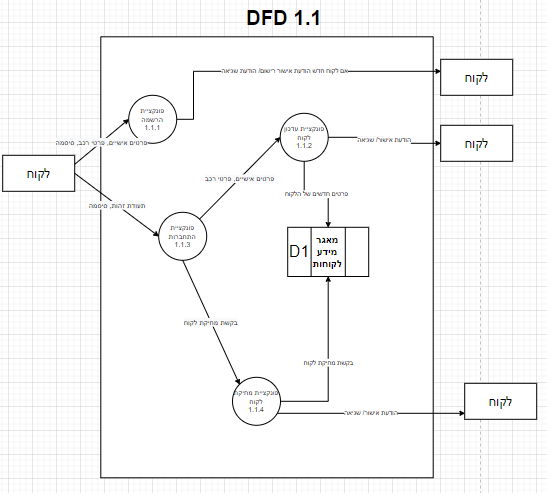
המערכת פועלת בתיאום מול הכנסת נתונים ממשתמש. כדי להגיע לתיאום הכי טוב בין הלקוח לכונן המתאים, יש צורך באפיון התקלה מראש. האלגוריתם החכם עונה לצורך זה. צורך של האלגוריתם החכם הוא טפסים קודמים של כוננים וטופס שממולא ע"י הלקוח בעת פתיחת אירוע. המערכת מסתמכת על העובדה שמספר התקלות קיים ותמידי, ועדכון תמידי של הכוננים.

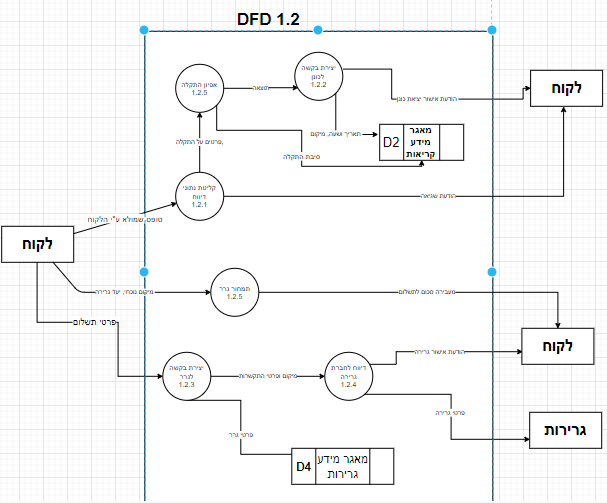
המהלך בניית המערכת השתמשתי במגוון רחב של מבני נתונים: Dictionary, factory, arrays, tables. מבני נתונים אלו מסייעים לשמור בצורה המיטבית על המידע אותו מכניס המשתמש.

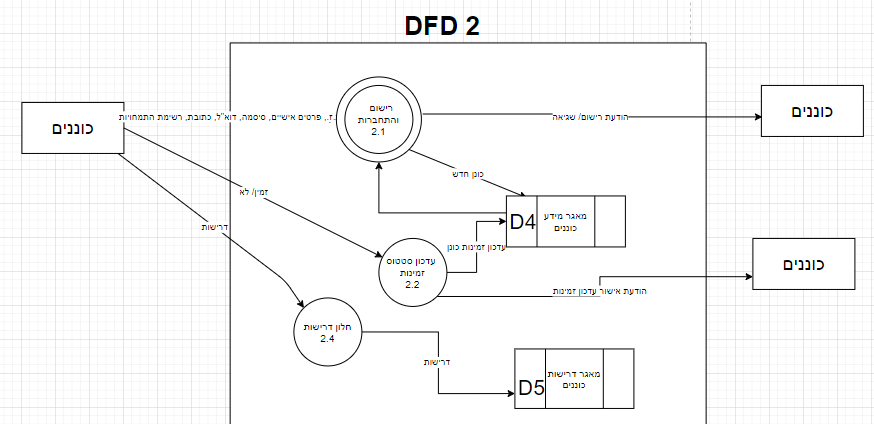
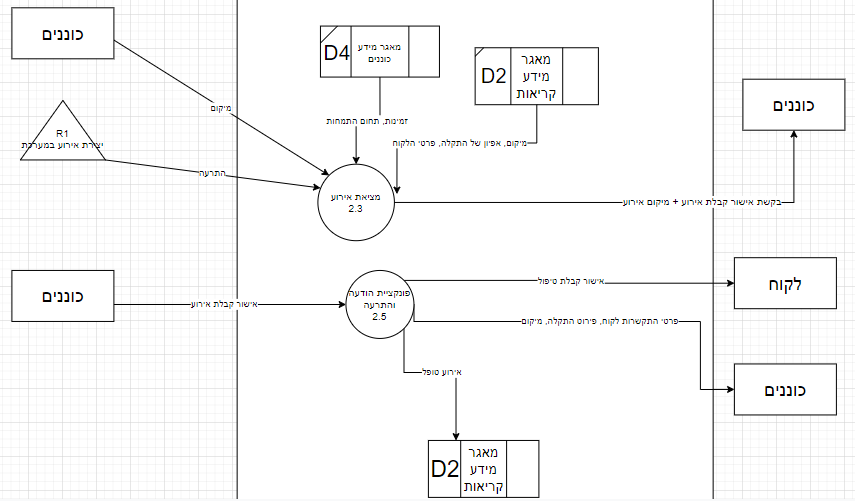
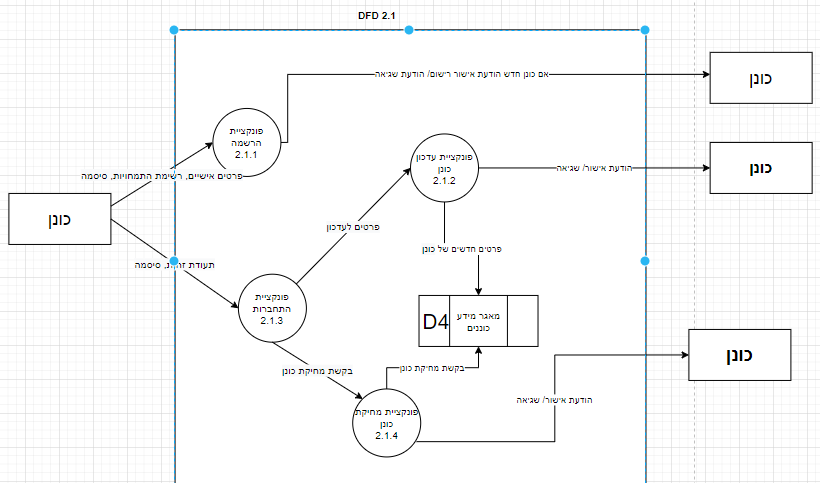
* המערכת בנויה משלוש יחידות: ניהול לקוחות, ניהול כוננים ודרג ניהולי (אלגוריתם חכם נמצא בו) .

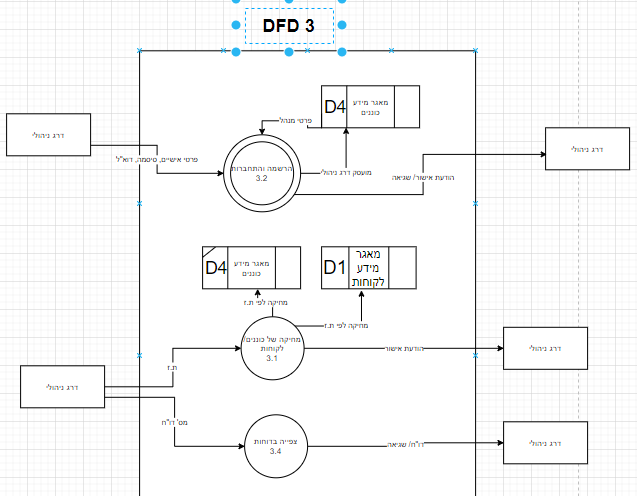
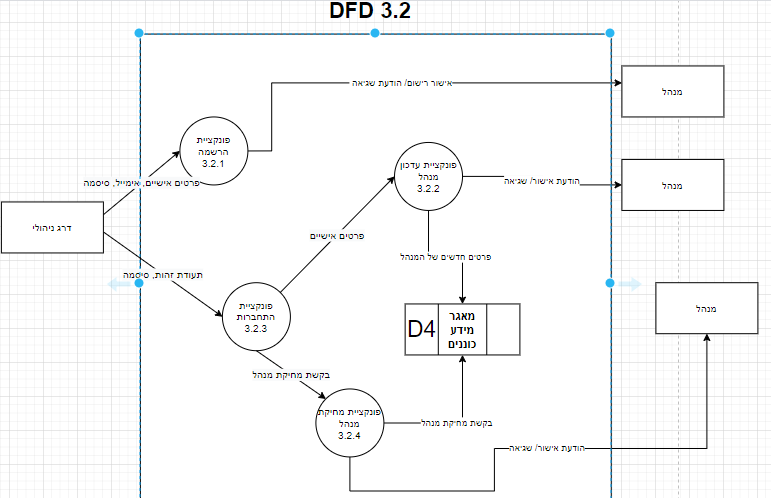










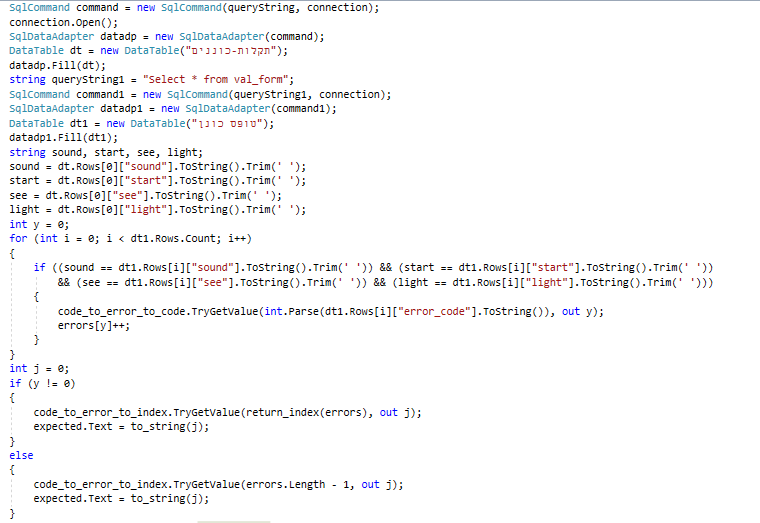


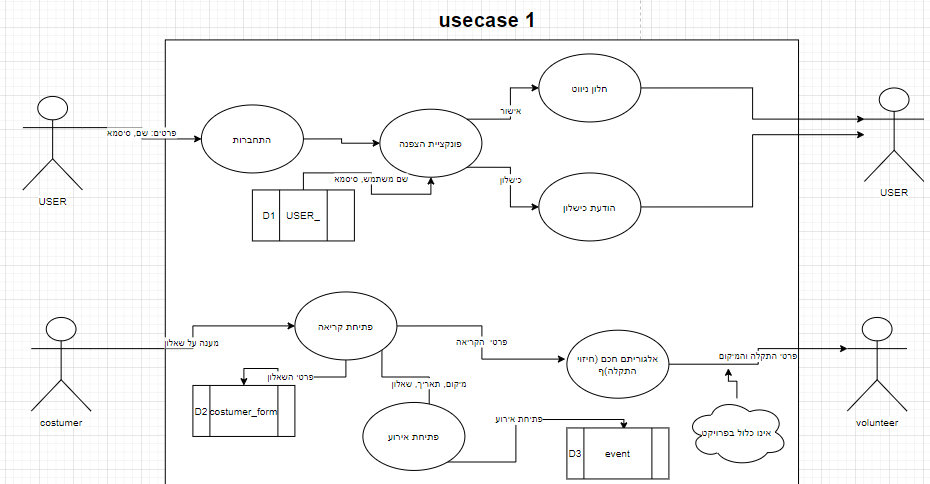
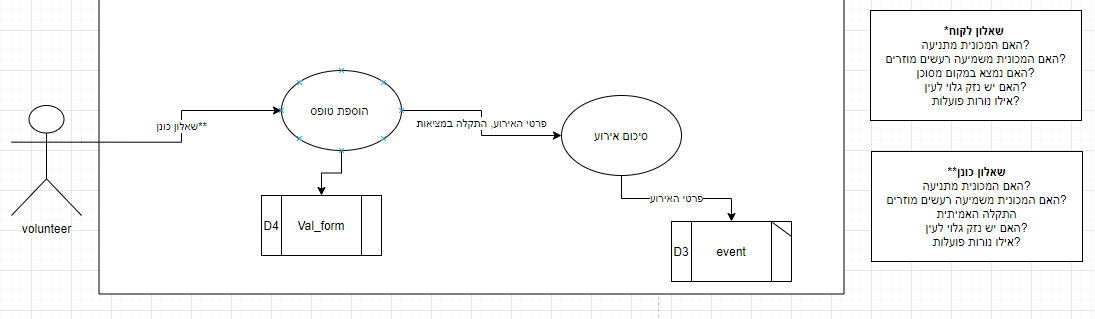
בפרויקט זה אין שימוש בשירותים אינטרנטיים. פרוטוקול התקשורת היחיד בו משתמש הפרוייקט מוגדר באופן אוטומטי ע"י תוכנת הMicrosoft SQL Server Management Studio 18.

* בפרוייקט זה רק לאנשים מורשים בעלי שם משתמש וסיסמה קבועים יכולים להיכנס אליה, רק למורשי תוכנה יש יכולת לעדכן שם משתמש ולצרף מורשים חדשים רק דרך המערכת. הסיסמאות מוגנות ע"י הצפנה. ובנוסף הסיסמה מושווית פעם בSQL ופעם בתוכנה כדי להבטיח ביטחון כפול וכדי להגן מסוגי תקיפות שונות. הן שם המשתמש והן הסיסמה מוגבלים רק לשימוש באותיות אנגלית(גדולה וקטנה) וכן ספרות, ההצפנה מחלקת את הסיסמה לקבוצות של חמש תווים ומחליפה את הסדר של כל חמישייה, מעלה את הערך האסקי של כל אות ב5, במחזוריות כלומר אם האות היא Z תישמר במקומה האות E.

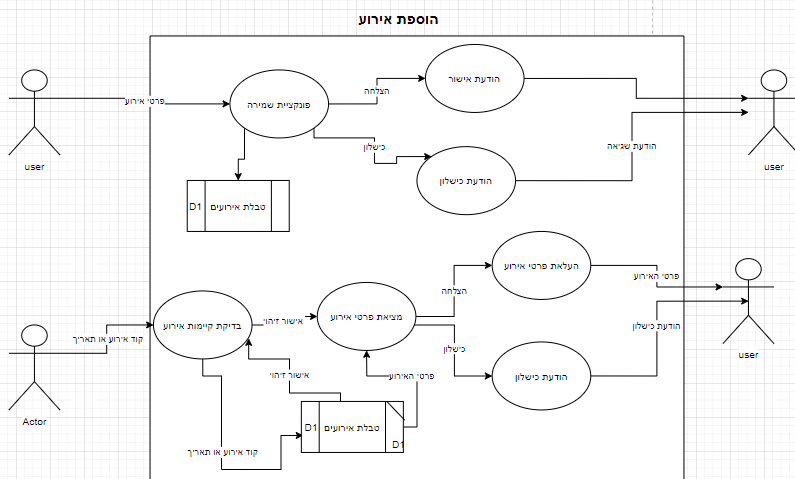
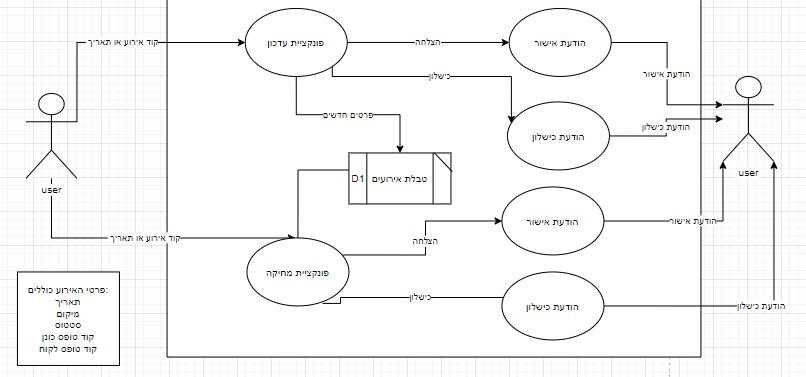
* האלגוריתם החכם מקבל רק קוד טופס של לקוח. האלגוריתם משווה את כל ההגדרות של הטופס לכל הגדרות טפסי הכוננים שהיו עד כה, כל טופס כונן שהאלגוריתם מצא לנכון שדומה לטופס הלקוח הוא לוקח את התקלה שהייתה שם ומכניס למערך אם אין כזה המערכת תסווג את התקלה כלא ידועה. יצרתי מערך של טיפוסים מסוג ERROR אותם יצרתי כדי למצוא סבירויות של תקלות. במערך זה נמצא כמה פעמים האלגוריתם החכם טעה ואיזה תקלה הופיעה במקום התקלה הצפוייה, כלומר איזה עוד תקלה יש סבירות גבוהה שתופיע במקום התקלה שנמצאה על ידי האלגוריתם. לכול טופס יש שמונה סעיפים. לכן, יש הרבה מאוד סבירויות (2 כפול 2 כפול 3 כפול 4 )בסוף בודקים איזה תקלה הופיעה הכי הרבה פעמים ולפי זה שולח כונן מתאים לשתי התקלות או שתי האפשרויות שזה תפקידו.

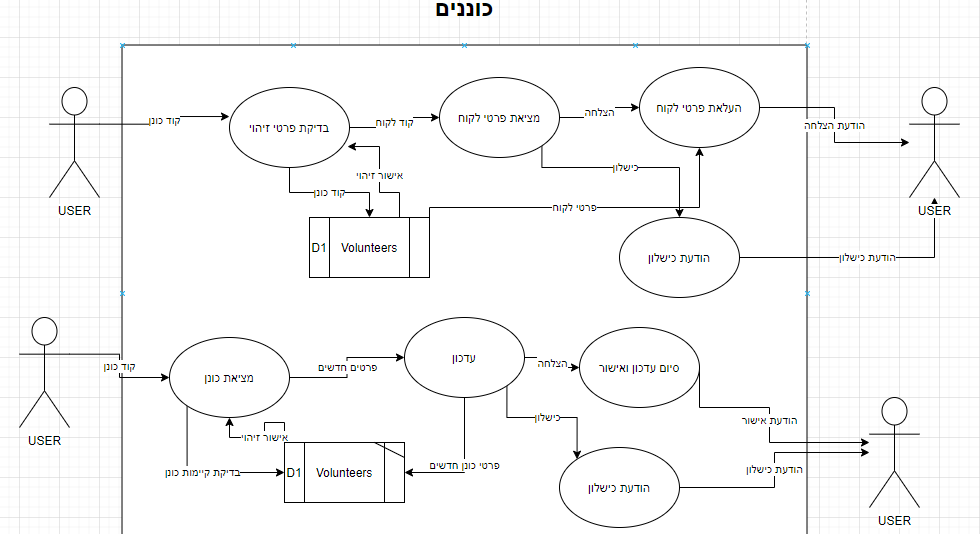
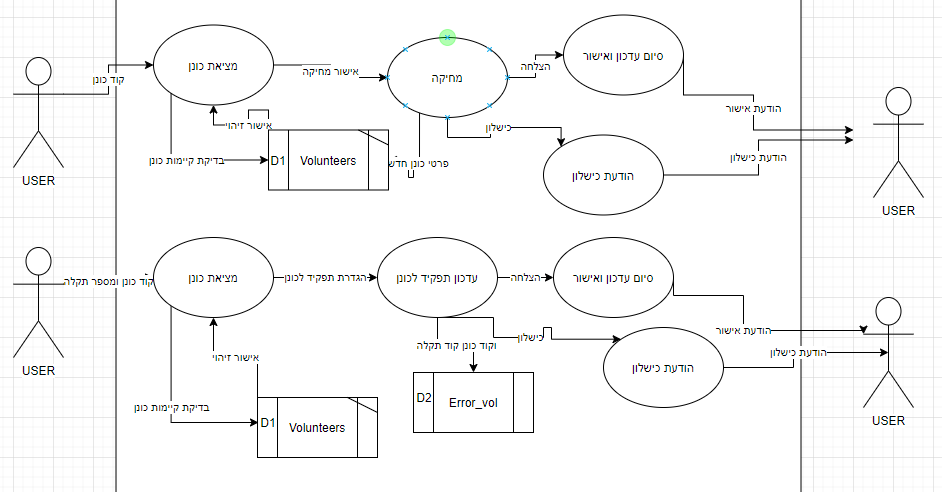
להלן חלק מקטע הקוד המרכיב את האלגוריתם החכם:

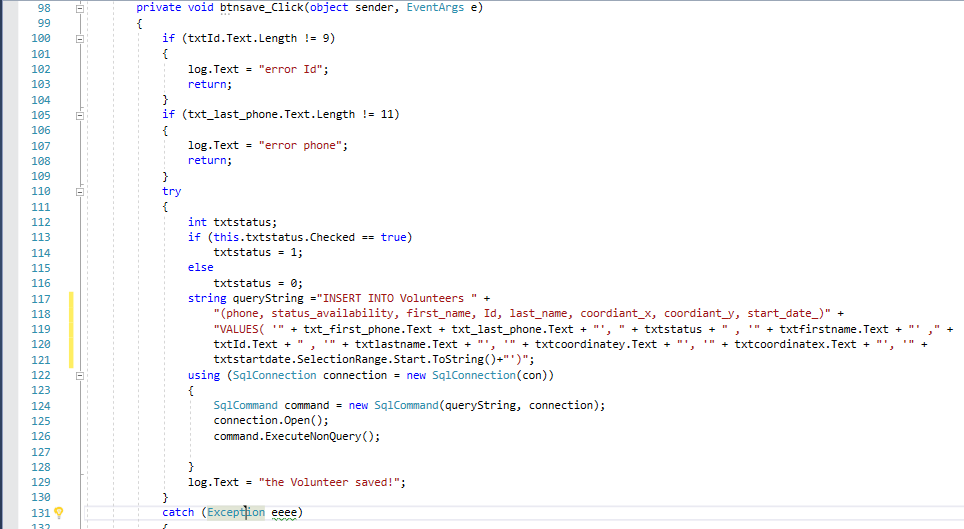
בתמונה מובאת חלק מהאלגוריתם החכם, העוסק בהשוואה ובמציאת התקלה הצפויה. בקוד רואים את ההשווה עצמה לכל טפסי הכוננים הקודמים



* במסך אירועים יש אפשרות לערוך אירוע, לצפות באירועים קיימים, ושמור ולמחוק אירועים מבסיס הנתונים:



* במסך כוננים יש אפשרות לשמור כונן חדש, לערוך פרטי כונן קיים, מחיקת כונן קיים והצגת כל הכוננים. כמו כן יש אפשרות להגדיר תפקיד לכונן דרך מסך זה:

לפנינו קוד המדגים את שמירת כוננים:

להלן רשימת שאילתות:

"select code\_Volunteers,COUNT(code\_Volunteers) as num from val\_form group by code\_Volunteers order by num DESC"; // שאילתה המציגה את קודי הכוננים הכי פעילים

"select error\_code, COUNT(error\_code) as num from val\_form group by error\_code order by num desc"; // שאילתה המציגה את התקלות הנפוצות

"select code\_costumer,COUNT(code\_costumer) as num from costumer\_form group by code\_costumer order by num desc"; // קודי הלקוחות שקראו הכי הרבה פעמים שאילתה המציגה את

select \* from events\_ where date > CURRENT\_TIMESTAMP - 30" ;// שאילתה המציגה את אירועי החודש

"SELECT \*FROM events\_ INNER JOIN drag ON drag.event\_code = events\_.event\_code where date > CURRENT\_TIMESTAMP - 30"; // שאילתה המציגה את גרירות החודש האחרון

"select sum(profit) as profit from drag INNER JOIN events\_ ON drag.event\_code = events\_.event\_code where date > CURRENT\_TIMESTAMP - 365"; // שאילתה המציגה את הרווח מגרירות בשנה האחרונה

= "select COUNT(\*) from error\_Prediction inner join events\_ on error\_Prediction.Num\_of\_costumer\_form = events\_.Num\_of\_costumer\_form inner join val\_form on events\_.Num\_of\_val\_form = val\_form.Num\_of\_val\_form where error\_Prediction.error\_code = val\_form.error\_code"; // שאילתה המציגה כמה פעמים האלגוריתם החכם פעל טוב

"select COUNT(\*) from error\_Prediction inner join events\_ on error\_Prediction.Num\_of\_costumer\_form = events\_.Num\_of\_costumer\_form inner join val\_form on events\_.Num\_of\_val\_form = val\_form.Num\_of\_val\_form where error\_Prediction.error\_code != val\_form.error\_code"; // שאילתה המציגה כמה פעמים האלגוריתם החכם לא פעל טוב

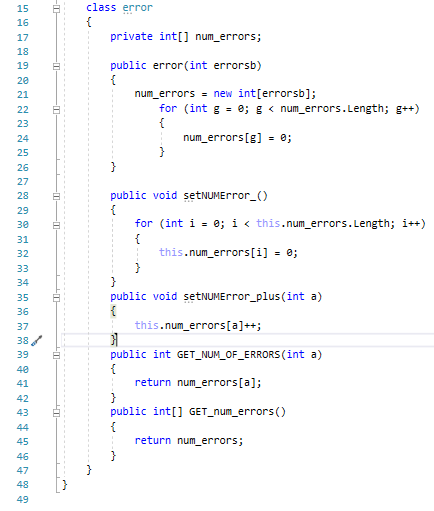
"select \* from Volunteers where status\_availability = 1"; // שאילתה המציגה את הכוננים הזמינים

"SELECT COUNT(CASE WHEN val\_form.code\_Volunteers > 5 THEN 1 ELSE NULL END) as c ,first\_name, last\_name, Volunteers.code\_Volunteers, coordiant\_x, coordiant\_y, status\_availability FROM Volunteers INNER JOIN val\_form ON Volunteers.code\_Volunteers = val\_form.code\_Volunteers group by first\_name, last\_name, Volunteers.code\_Volunteers, coordiant\_x, coordiant\_y, status\_availability"; // שאילתה המציגה את איזה כוננים עשו יותר מחמש קריאות

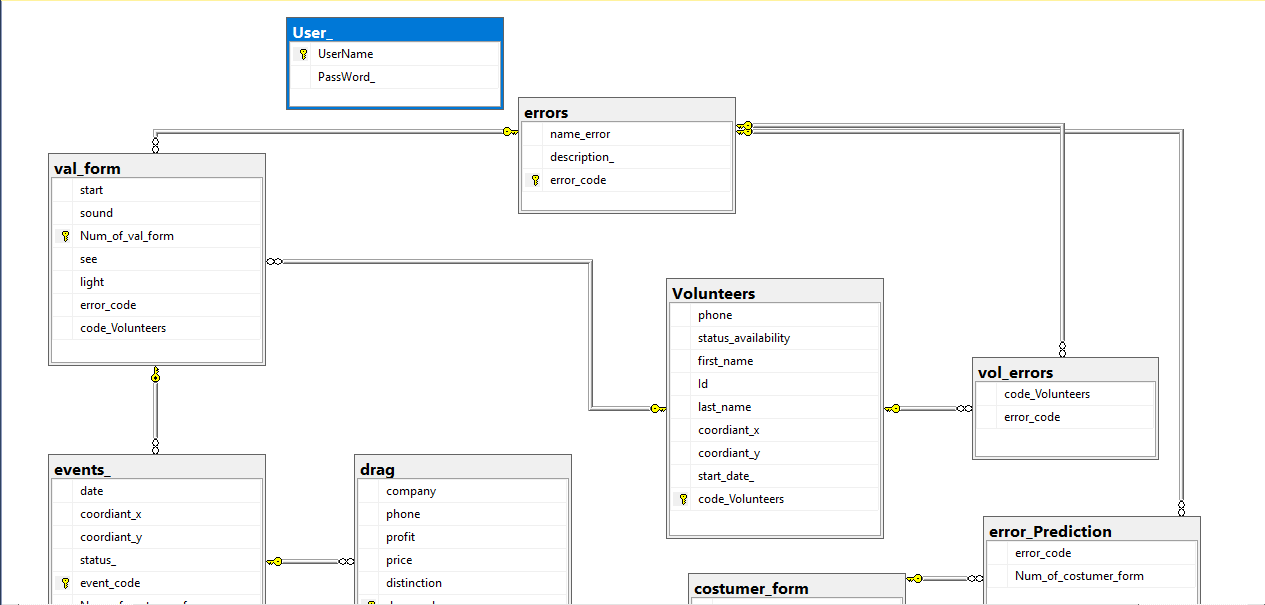
"select COUNT(costumer\_form.Num\_of\_costumer\_form) as times, type\_car from costumer\_form inner join costumer on costumer.code\_costumer = costumer\_form.code\_costumer group by(type\_car) order by(times) desc"; // שאילתה המציגה את הרכבים התקולים ביותר

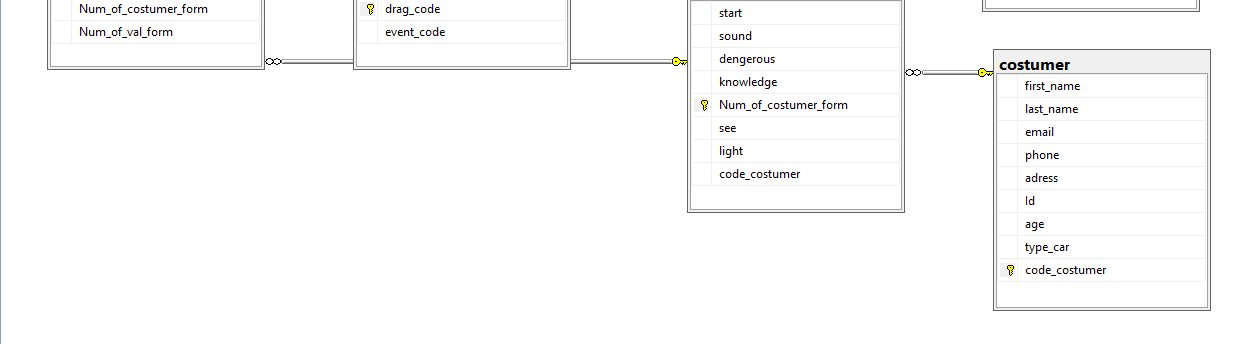
"select COUNT(company) as times, company from drag group by company order by times"; // שאילתה המציגה את החברה שעשתה הכי הרבה גרירות

"select events\_.coordiant\_x,events\_.coordiant\_y,events\_.date, events\_.event\_code, val\_form.Num\_of\_val\_form, Volunteers.first\_name,Volunteers.last\_name,Volunteers.phone, costumer\_form.Num\_of\_costumer\_form, costumer.first\_name,costumer.last\_name,costumer.phone, costumer.type\_car from events\_ inner join val\_form on events\_.Num\_of\_val\_form = val\_form.Num\_of\_val\_form inner join Volunteers on val\_form.code\_Volunteers = Volunteers.code\_Volunteers inner join costumer\_form on events\_.Num\_of\_costumer\_form = costumer\_form.Num\_of\_costumer\_form inner join costumer on costumer\_form.code\_costumer = costumer.code\_costumer where events\_.status\_ = 0 and events\_.date = '" +DateTime.Today.ToString(format)+"' " // שאילתה המציגה את החמ"ל – האירועים המתקיימים כרגע עם פרטי הכונן והלקוח

להלן קטע קוד של מחלקת מערכי הסבירויות:

בקטע הקוד מפורט מחלקת מערכי הסבירויות שבעזרתם האלגוריתם החכם פועל. בעזרת מחלקה זאת האלגוריתם יוכל לזהות את התקלה שסבירה שיכולה להיות לפי התקלה הצפויה. כלומר כאשר האלגוריתם החכם מאפיין תקלה מסוג אחד והתקלה בפועל הייתה אחרת היא תירשם במערך זה. והתקלה שקרתה הכי הרבה פעמים במקום אותה סוג תקלה צפויה על ידי האלגוריתם החכם תיבחר גם היא

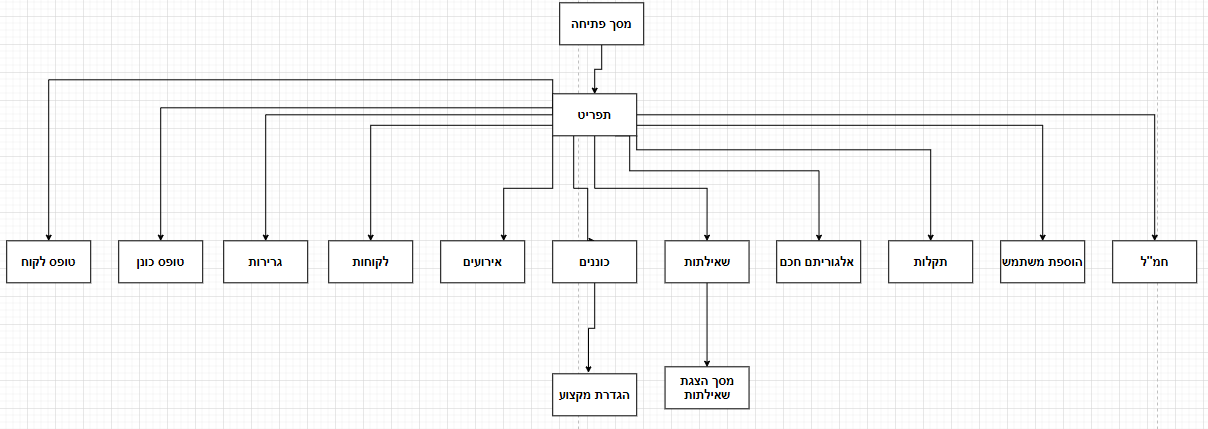
* לפנינו תרשים UML וקשרי הגומלין בין הטבלאות:



* בפרוייקט השתמשתי בפיצ'רים רבים: button, checkbox, checklistbox, ComboBox, DataGridView, DataTimePicker, Label, MenuStrip, Panel, PictureBox, TextBox,

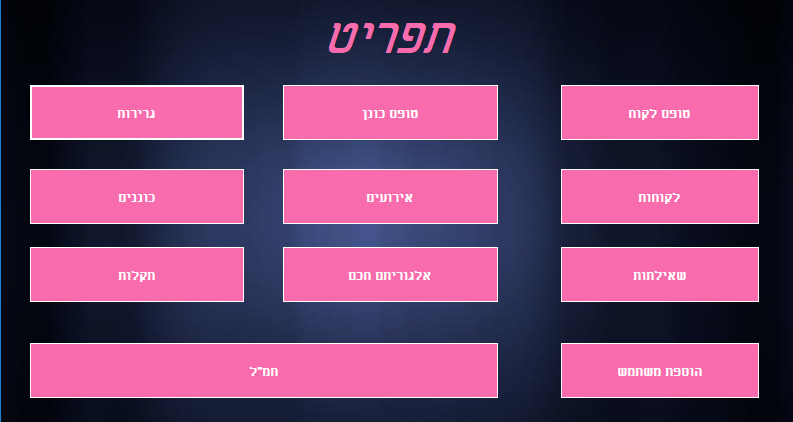
# **מדריך למשתמש:**

* תחילה המערכת נפתחת במסך התחברות בו על המשתמש להכניס סיסמה המורכבת מאותיות אנגלית(הקוד רגיש לאותיות גדולות) או מספרים או שילוב שלהם. אין הגבלה על אורך שם המשתמש או הסיסמה, אך אסור ששם המשתמש יהיה קיים במערכת. לאחר ההתחברות נפתח חלון ניווט. דרך חלון זה ניתן לעשות כל מה שהמערכת יודעת להציע.

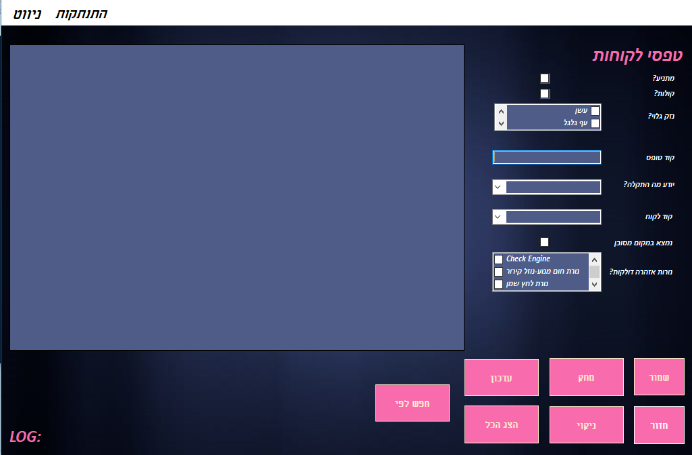
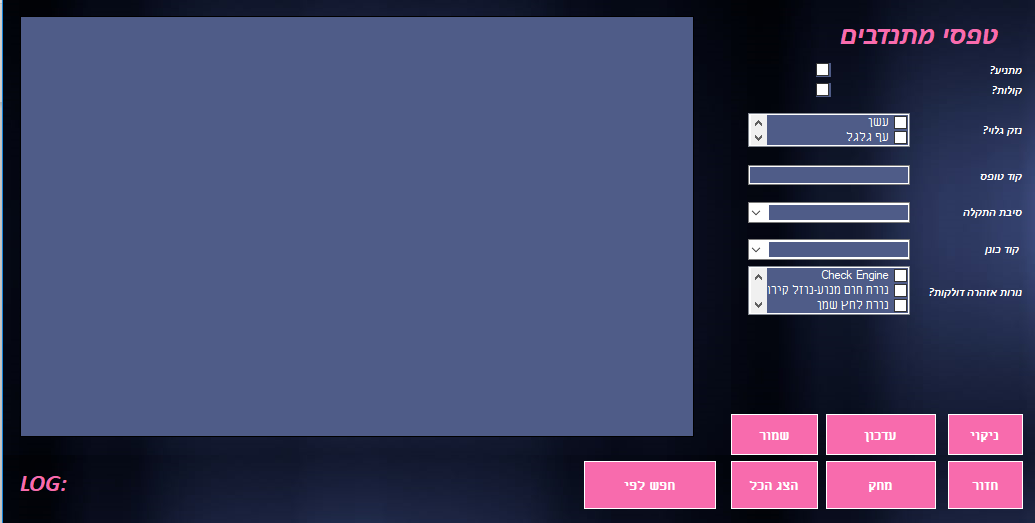




מסך התחברות: מתחברים ע"י שם משתמש וסיסמה נתונים מראש

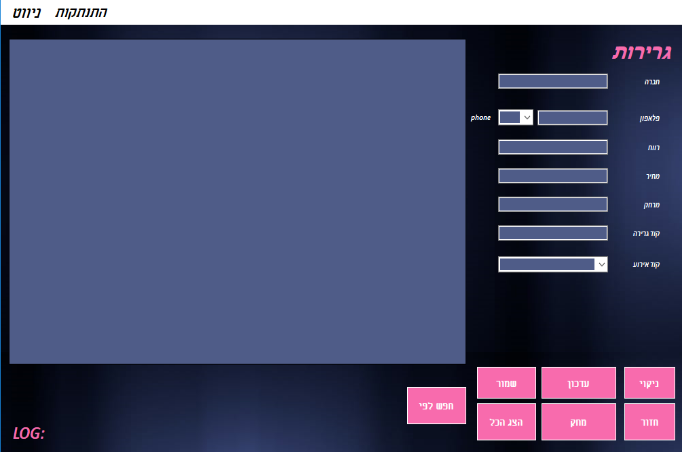


מסך תפריט: מסך ניווט בין הטפסים

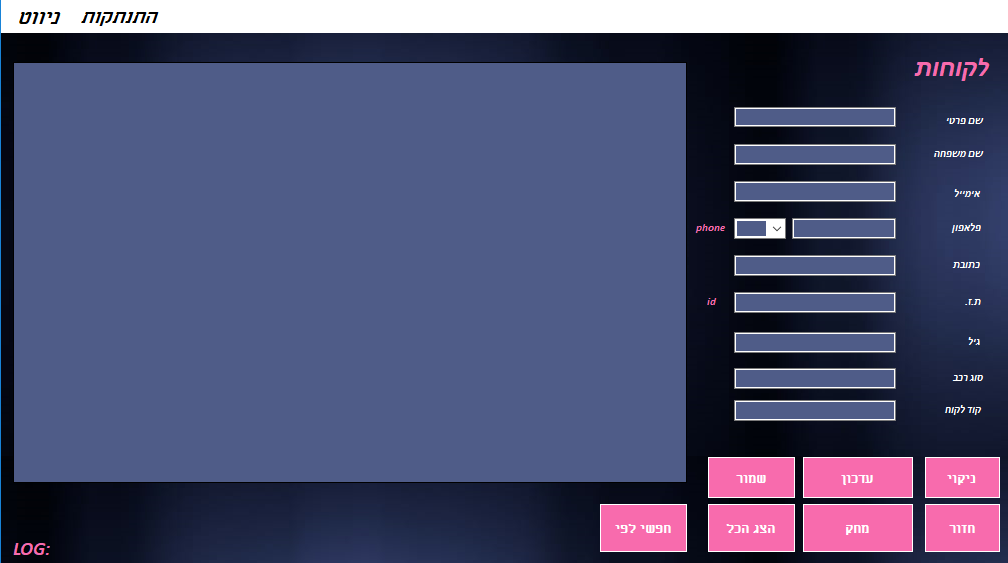


טפסי מתנדבים: במסך זה ניתן לערוך למחוק ולשמור טפסי כוננים קודמים, וכן לחפש טופס לפי קטגוריה לבחירה

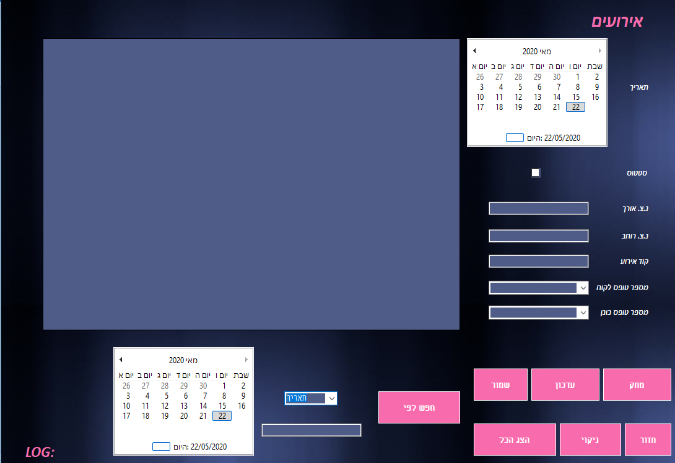
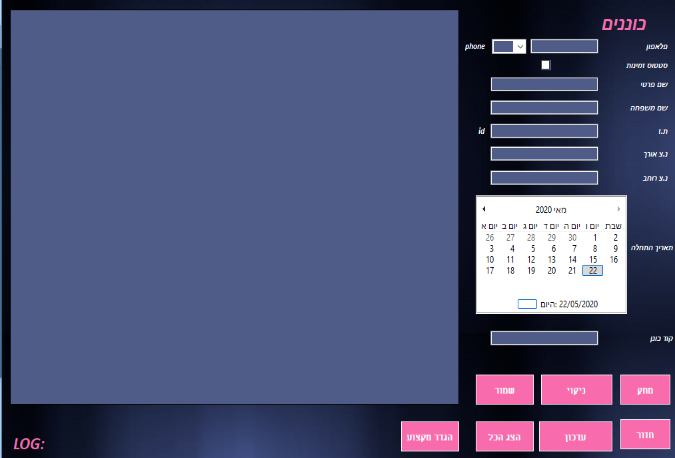
טפסי לקוחות: במסך זה ניתן לערוך למחוק ולשמור טפסי לקוחות קודמים, וכן לחפש טופס לפי קטגוריה לבחירה



מסך גרירות: במסך זה ניתן לערוך למחוק ולשמור אירועי גרירות קודמים, וכן לחפש אירוע גרירה לפי קטגוריה לבחירה

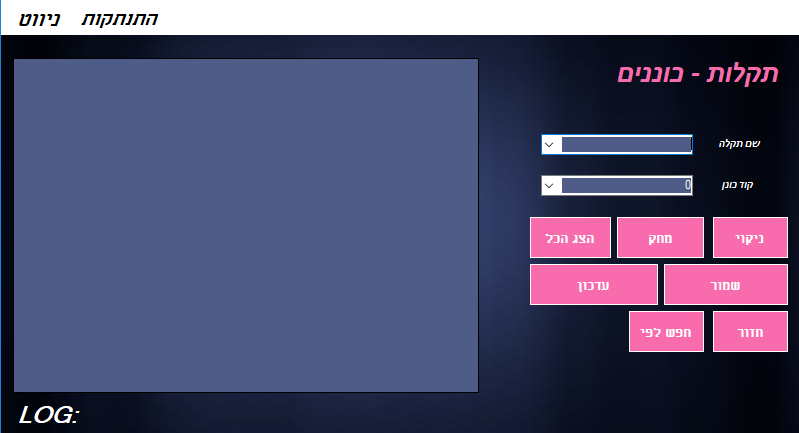


מסך לקוחות: במסך זה ניתן לערוך למחוק ולשמור לקוחות, וכן לחפש לקוחות לפי קטגוריה לבחירה



מסך כוננים: במסך זה ניתן לערוך למחוק ולשמור כוננים, וכן לחפש כוננים לפי קטגוריה לבחירה

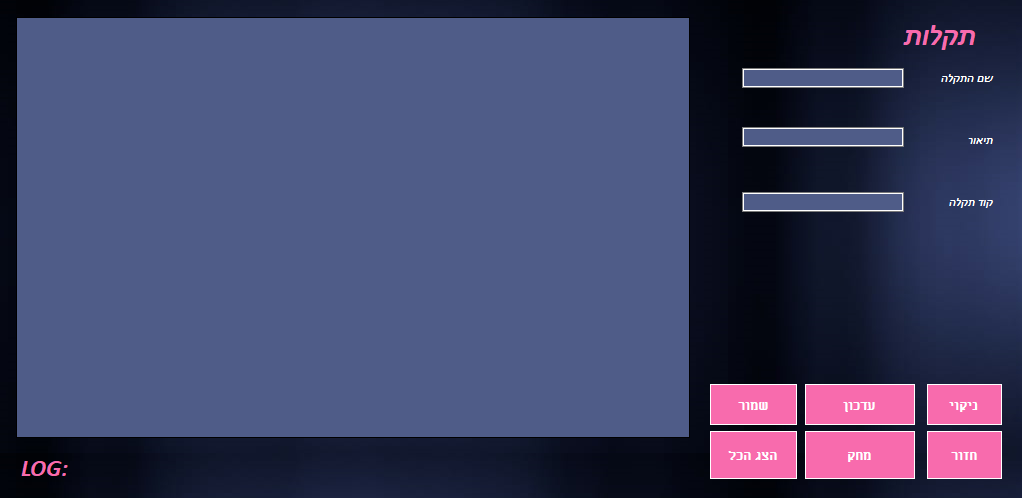
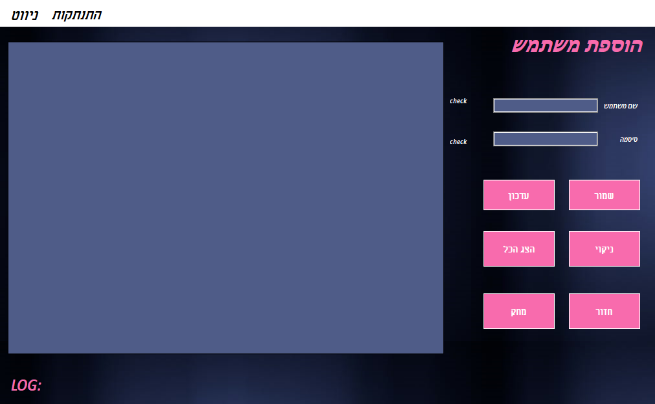
מסך אירועים: במסך זה ניתן לערוך למחוק ולשמור אירועים, וכן לחפש אירועים לפי קטגוריה לבחירה



מסך תקלות - כוננים: במסך זה ניתן לערוך למחוק ולשמור תפקידים לכוננים, וכן לחפש כוננים לפי קטגוריה לבחירה



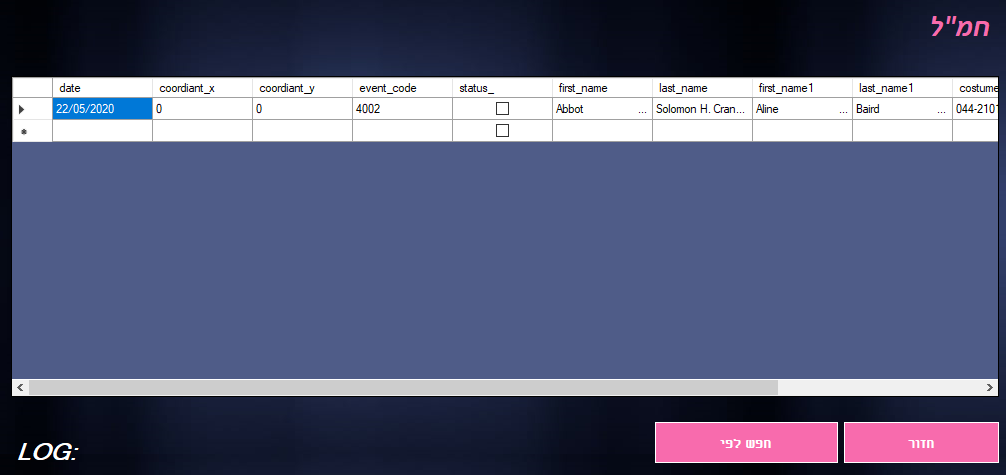
מסך שאילתות: במסך זה המשתמש יוכל למצוא מידע שימושי ונחוץ בלחיצת כפתור על כל אחת מהכפתורים.



מסך הוספת משתמש: במסך זה ניתן לערוך למחוק ולשמור משתמשים חדשים, וכן לחפש משתמש לפי שם משתמש

מסך תקלות: במסך זה ניתן לערוך למחוק ולשמור תקלות, וכן לחפש תקלות לפי קוד תקלה

מסך אלגוריתם חכם: במסך זה המשתמש יוכל ע"י הכנסת מספר טופס לקוח לדעת את התקלה הצפויה לפתוח אירוע חדש וטופס כונן חדש, וכל זה בלחיצת כפתור.



מסך חמ"ל: במסך זה ניתן לראות את כל האירועים של אותו יום שהסטטוס שלהם לא בוצע

**תצלום בגודל רגיל של כל המסכים נמצא בנספח**

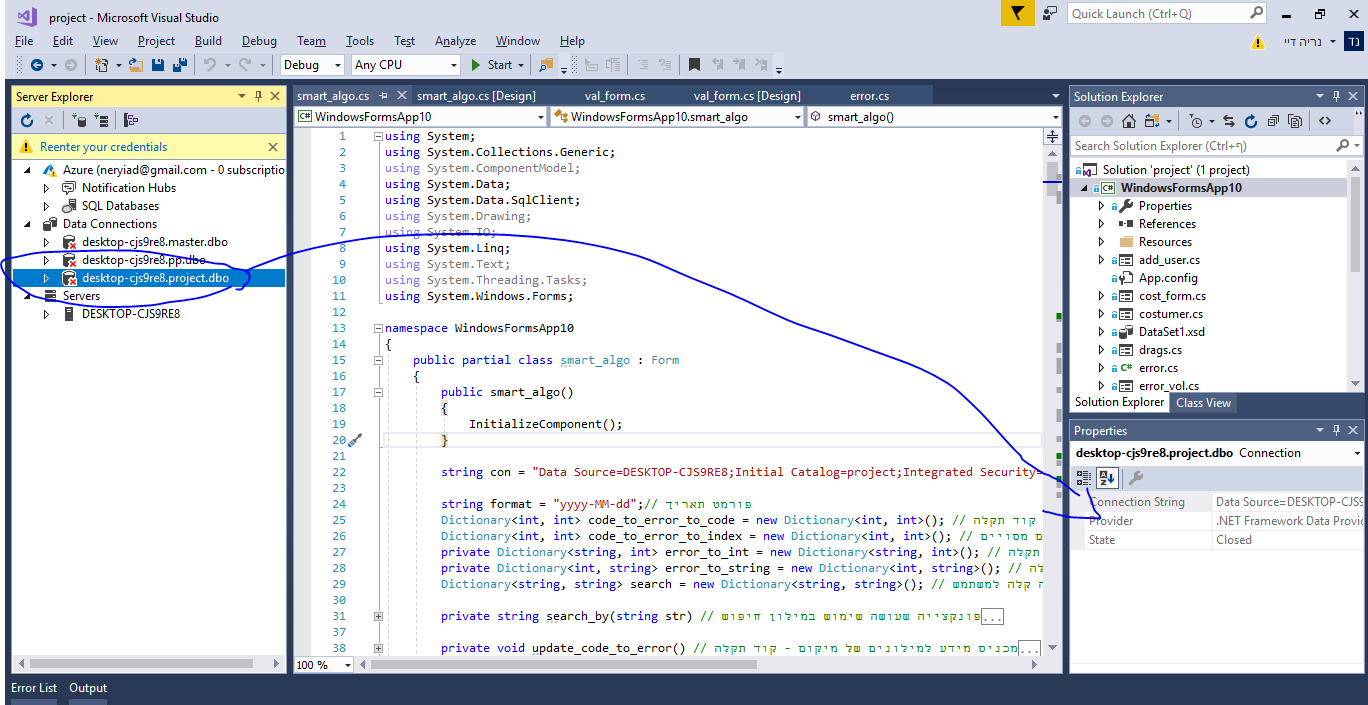
# **בסיס הנתנונים:**

# **מדריך למפתח**:

|  |  |
| --- | --- |
| add\_user.cs | קוד שמפעיל את מסך הוספת משתמש |
| add\_user.Designer.cs | קוד שמחזיק את עיצוב מסך הוספת משתמש |
| cost\_form.cs | קוד שמפעיל את מסך טפסי לקוחות |
| cost\_form.Designer.cs | קוד שמחזיק את עיצוב מסך טפסי לקוחות |
| costumer.cs | קוד שמפעיל את מסך לקוחות |
| costumer.Designer.cs | קוד שמחזיק את עיצוב מסך לקוחות |
| drags.cs | קוד שמפעיל את מסך גרירות |
| drags.Designer.cs | קוד שמחזיק את עיצוב גרירות |
| error\_vol.cs | קוד שמפעיל את מסך הגדרת תפקידים לכוננים |
| error\_vol.Designer.cs | קוד שמחזיק את עיצוב הגדרת תפקידים לכוננים |
| errors.cs | קוד שמפעיל את מסך תקלה |
| errors.Designer.cs | קוד שמחזיק את עיצוב תקלה |
| events.cs | קוד שמפעיל את מסך אירועים |
| events.Designer.cs | קוד שמחזיק את עיצוב אירועים |
| menu.cs | קוד שמפעיל את מסך תפריט |
| menu.Designer.cs | קוד שמחזיק את עיצוב תפריט |
| op\_room.cs | קוד שמפעיל את מסך חמ"ל |
| op\_room.Designer.cs | קוד שמחזיק את עיצוב חמ"ל |
| opening\_screen.cs | קוד שמפעיל את מסך ההתחברות |
| opening\_screen.Designer.cs | קוד שמחזיק את עיצוב ההתחברות |
| Program.cs | קוד שמפעיל את התוכנית |
| query.cs | קוד שמציג את השאילתות |
| query.Designer.cs | קוד שמחזיק את עיצוב מסך הופעת השאילתות |
| querybtn.cs | קוד שמפעיל את מסך לחצני שאילתות |
| querybtn.Designer.cs | קוד שמחזיק את עיצוב לחצני שאילתות |
| querytxt.cs | קוד שמחזיק את כל השאילתות |
| smart\_algo.cs | קוד שמפעיל את מסך האלגוריתם החכם |
| smart\_algo.Designer.cs | קוד שמחזיק את עיצוב האלגוריתם החכם |
| val\_form.cs | קוד שמפעיל את מסך טפסי כוננים |
| val\_form.Designer.cs | קוד שמחזיק את עיצוב טפסי כוננים |
| Volunteers.cs | קוד שמפעיל את מסך כוננים |
| Volunteers.Designer.cs | קוד שמחזיק את עיצוב מסך כוננים |

## הורדה ושימוש במערכת:

בכדי להוריד ולפתוח את העבודה, צריך להוריד את כל הקבצים בתיקייה שנשלחה. המערכת פועלת רק במערכת הפעלה מסוג WINDOWS. התוכנות הדרושות הם: VISUAL STUDIO ו Microsoft SQL Server Management Studio 18 . בכדי להפעיל את התוכנה צריךלפתוח את SQL Server Management Studio להריץ את קובץ הDDL, בשם script.sql ובכך לחבר אותו אל המערכת. בתיקייה המצורפת יש את קובץ הפרוייקט הכתוב כקוד, ויש את הקוד להרצת הDATABASE. בכדי להשיג את הconnection string , צריך לפתוח את הפרוייקט, בvisual, בלשונית View לבחור בServer explorer. . בdata connection קליק ימני ולבחור properties . ושם לפי התמונה הבאה תמצא אותה:



להלן כמה קישורים לעזרתך:

מייקרוסופט: <https://docs.microsoft.com/en-us/sql/integration-services/import-export-data/start-the-sql-server-import-and-export-wizard?view=sql-server-ver15>

Stackoverflow - <https://stackoverflow.com/questions/10479763/how-to-get-the-connection-string-from-a-database>

וכן כרטון הדרכה למקרה ולא מובן - <https://www.sqlshack.com/importexport-data-sql-server-using-sql-server-import-export-wizard/>

## **רפלקציה:**

את הפרוייקט התחלתי בחשיבה על הנושא. רציתי לבחור בנושא שהכי משמעותי בשבילי דבר שבאמת הרגשתי שיש בו צורך, ונזכרתי בארגון ידידים אליו פנינו לעזרה. המורה לימד אותי את החומר הנצרך, בהתחלה חשבתי כי מדובר בכמות קטנה של חומר, אך מהר מאוד התבדיתי. למדתי כמות חומר גבוהה ברמה גבוהה כמו שאני חושב שבחיים לא למדתי! העברנו את השיעורים בצורה כיפית וחוייתית. אבל הרווחתי מאוד מעצם עשיית הפרוייקט. למדתי על עצמי איך אני עובד במצבי לחץ. למדתי לראות ראייה כללית מלמעלה. לתכנת מסודר לעבוד עם תוכנית עבודה, וכמובן למדתי לקודד ברמה. לפעמים הייתי צריך לכתוב קטעי קוד מסובכים במיוחד שחשבתי עליהם רבות, וכמובן נתקלתי בקשיים טכניים בתוכנה . בעזרת ריכוז ובעזרת המורה הצלחתי לעבור אותם. והמלאכה הושלמה על הצד הטוב ביותר. לא יכולתי לבקש תנאיים יותר טובים לעשות וללמוד, אם כי פחות לחץ משאר המקצועות :)

פעם אחת המהלך כתיבת הקוד. נמחקה לי כל העבודה שעשיתי. נלחצתי, לא ידעתי מה לעשות פניתי למורה וביחד חשבו על פיתרון. לבסוף בדרך יצירתית במיוחד הכוללת גיבוי שאני תמיד עושה הצלחנו לשחזר את הפרוייקט כולו מלבד כמה קטעי קוד קטנים. מזה למדתי אודות עצמי, על יכולותי לעמוד בלחץ ולא להישבר. במבט לאחור העבודה שעשיתי מושלמת. תמיד כיף להסתכל על עבודה טובה ועובדת שאתה יצרת

## **בביליוגרפיה:**

#### אתר ארגון "ידידים בדרכים" - <https://yedidim-il.org/>

#### אתר חברת "נתיבי ישראל" - <https://www.iroads.co.il/%D7%A9%D7%99%D7%A8%D7%95%D7%AA-%D7%9C%D7%A7%D7%95%D7%97%D7%95%D7%AA/%D7%A2%D7%9E%D7%95%D7%93-%D7%9B%D7%AA%D7%91%D7%94-%D7%A0%D7%99%D7%99%D7%93%D7%95%D7%AA-%D7%A9%D7%99%D7%A8%D7%95%D7%AA/>

#### אתר חברת "שגריר" העוסקות בנושא החילוץ בדרכים - <https://www.shagrir.co.il/?lang=il&category=shiroutdereh>

#### אתר AUTO – למציאת בעיות במכונית - <https://www.auto.co.il/article/131189>

#### אתר ENGINE – למציאת בעיות במכונית - <https://engieapp.com/he/%d7%a1%d7%95%d7%92%d7%99-%d7%a0%d7%95%d7%a8%d7%95%d7%aa-%d7%90%d7%96%d7%94%d7%a8%d7%94-%d7%91%d7%a8%d7%9b%d7%91/>

מייקרוסופט: <https://docs.microsoft.com/en-us/sql/integration-services/import-export-data/start-the-sql-server-import-and-export-wizard?view=sql-server-ver15>

Stackoverflow - <https://stackoverflow.com/questions/10479763/how-to-get-the-connection-string-from-a-database>

## **נספח:**

