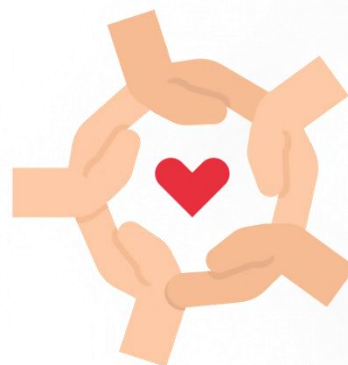


יד ליד
מכל הלב



מגישה: תמר שם טוב

ת.ז: 325112332

ביה"ס: סמינר בית יעקב צפת

מנחה: המ' יעל עמר

תאריך הגשה: יולי 2022

תוכן

3	הצעת פרויקט מאושרת
5	(תודות :)
5	תקציר / מבוא:
5	הרקע לפרויקט:
6	תהליך המחקר:
7	סקירת ספרות:
8	מטרות – יעדים:
8	אתגרים:
9	מדדי הצלחה למערכת:
9	רקע תיאורי:
9	תיאור מצב קיים:
9	ניתוח חלופות מערכת:
10	תיאור החלופה הנבחרת, כולל נימוקים לבחירתה:
10	אפיון המערכת שהוגדרה:
10	ניתוח דרישות המערכת:
11	מודול המערכת:
11	אפיון פונקציונלי:
11	ביצועים עיקריים:
12	תיאור הארכיטקטורה:
12	הארכיטקטורה של הפתרון המוצעת בפורמט של Top-Down Level Design:
12	תיאור הרכיבים בפתרון:
13	ארכיטקטורת רשת:
13	תיאור פרוטוקולי התקשורת:
13	שרת – לקוח:
13	תיאור הצפנות אם יש:
13	ניתוח ותרשים use-case של המערכת המוצעת:
13	תרשים use-case:
15	רשימת use-cases:
15	תיאור ה UC העיקריים של המערכת:
19	מבני נתונים בהם השתמשתי:
19	עץ מודולים:
20	תרשים מחלקות:

22	תיאור המחלקות המוצעות
38	אלגוריתמים מרכזיים:
38	חישוב מרחק בין כתובות המתקבלות כטקסט:
40	אלגוריתם קריאה מקובץ excel במבנה לא קבוע.
45	אלגוריתם שיבוץ מערכת בארגון:
59	תיאור התוכנה:
59	סביבת עבודה
59	שפות תכנות
59	תיכון המערכת
60	תיאור מסד נתונים
60	תרשים טבלאות + קשרי גומלין
61	תיאור הטבלאות ב data base:
64	תיאור מסכים:
64	תרשים מסכים:
65	פירוט מסכים:
79	מדריך למשתמש:
81	ניתוח יעילות
82	אבטחת מידע:
82	פיתוחים עתידיים:
82	מסקנות:
82	ביבליוגרפיה:

הצעת פרויקט מאושרת

הצעת פרויקט – יד הנדסת תוכנה

סמל מוסד: 240721

שם מכללה: סמינר בית יעקב צפת

שם הסטודנט: תמר שם טוב

תז הסטודנט: 325112332

שם הפרויקט: שיבוץ מערכת לפרויקט תרומה בקהילה

תיאור הפרויקט:

מחלקת הרווחה בעיריית צפת מקדמת את תחום ההתנדבות, הערבות ההדדית ותרומה בקהילה, במסגרתו מתנדבים. השיבוץ נעשה ע"י פרמטרים שונים: קבוצות גיל ואפיונים של קהל היעד – תחום ההתנדבות, ימים ושעות המתאימים למתנדב ולמשפחה, אזור מגורים נוח למתנדב וכו'. האתר נועד לחסוך את העבודה הידנית והמסורבלת ולשבץ את המתנדבים באופן אוטומטי עפ"י תחומי עניין וכן בזמנים המתאימים למתנדב, דבר אשר יתרום למס' גדול יותר של מתנדבים בקהילה.

ההתנדבות תבוצע במסגרת ארגוני ההתנדבות, מוסדות ושירותים בקהילה. כאשר רכזת ההתנדבות באגף הרווחה בעיריית צפת שותפה לתהליך. המערכת תשבץ לכל ארגון את המתנדבים המתאימים עפ"י פרטי מתנדב, אפשרויות, בקשות ושאלונים שימולאו ע"י המתנדבים בעת רישומם למערכת ההתנדבות.

האתר כולל כניסת מתנדב, כניסת מנהל, וכניסת רכז/מנהל ארגון/מוסד. מתנדב נכנס לאיזור האישי באתר ע"י תז וסיסמה, שם הוא יכול לעדכן את פרטיו האישיים – כתובת, מס' טלפון וכו', ולהגדיר את העדפותיו ואילוציו: תחומי התנדבות רצויים (יכנסו לתוקף רק לאחר אישור הרכז), מס' שעות התנדבות שבועי וכולל, ימים ושעות בהם הוא יכול / לא יכול להתנדב באופן קבוע / בתאריכים מסוימים. ההעדפה נקבעת עפ"י בחירת הערך המתאים מתוך רשימה – יכול / מעדיף שלא / לא יכול.

בכניסת מנהל לאתר יוצגו לו המתנדבים, אלו שמתנדבים אצלם ופרטיהם עפ"י דרישה, וכן הוא יכול לשבץ את המערכת בלחיצת כפתור לשנה הקרובה. באם יש אילוצים זמניים שהוכנסו לאחרונה- המערכת תתריע לו והוא יבחר אפ להתעלם מהם או לשבץ מערכת חדשה לפרק הזמן הדרוש.

הגדרת הבעיה האלגוריתמית:

לכל מתנדב יש סך שעות התנדבות מסוים אותו הוא תורם לקהילה -קשישים, בתי חולים, משפחות נזקקות, הדרכת תנועות נוער וכו'. ישנם ארגונים / מוסדות רבים הניזונים בקבלת מתנדבים ממחלקת רווחה. רכזת ההתנדבות מסייעת ומעודדת שיתופי פעולה בין הארגונים והשירותים ומקשרת בין המתנדבים לארגונים.

יש לשבץ את המערכת באופן אופטימלי כך שכל מתנדב יבצע את שעות ההתנדבות בתחום בו הוא עוסק. כל מתנדב יסמן בקטגוריות ההתנדבות עד 3 תחומי התנדבות בהם הוא מעוניין.

יש 2 אלגוריתמים מרכזיים לשיבוץ – גנטי והונגרי. מצאתי שלשיבוץ כמו שלי בו כל מתנדב צריך לקבל תפקיד בארגון / מוסד – מתאים יותר ההונגרי כי הוא משבץ לכל אחד, כלומר לכל מתנדב יהיה תפקיד, לא יישאר מתנדב בלי תפקיד.

רקע תיאורטי בתחום הפרויקט:

מחלקת רווחה בעיריית צפת מנהלת מערך התנדבותי שלם של מעורבות הדדית ותרומה לקהילה. במערכת יש מצד אחד ארגוני חסד כדוגמת ביקור חולים, יד שרה ועוד כמו גם מוסדות כגון בי"ח, בית אבות איזורי וכדומה. ומצד שני יש את ההון האנושי של מתנדבים אותם יש לשבץ בארגון / מוסד המתאים עפ"י בקשותיהם, כישוריהם, יכולותיהם, שאלון התאמה וכן תנאי סף / בקשות של הארגונים.

תהליכים עיקריים בפרוייקט:

- * מתנדב נרשם למערך תרומה בקהילה ומעורבות הדדית לאתר ע"י הקלדת תז וכתובת מייל
- * המערכת שולחת סיסמה זמנית לכתובת המייל שהוכנסה
- * המתנדב מקליד את הסיסמה שנשלחה אליו, במידה והסיסמה נכונה – נפתח לו טופס באתר בו הוא מאשר את פרטיו האישיים: שם, כתובת טלפון וכו'.
- * מתנדב נכנס לאזור האישי באתר ע"י תז וסיסמה ומעדכן את העדפותיו: ימים ושעות רצויים – ע"י בחירה מרשימה: יכול / מעדיף שלא / לא יכול, אזורי התנדבות רצויים, עד 3 תחומי התנדבות מועדפים
- * המתנדב עונה על שאלון התאמה
- * המערכת מנתחת את הנתונים שנאספו ומפעילה עליהם אלגוריתם שיבוץ
- * כל מנהל/רכז ארגון/מוסד מקבל את רשימת המתנדבים ששובצו אצלו כולל פרטיהם
- * מנהל/רכז נכנס לאזור האישי של הארגון שלו ובוחר אם לבקש מהמערכת שיבוץ בתוך הארגון עפ"י צרכי הארגון כפי שהכניסם
- * במידה ומבוקש שיבוץ בתוך הארגון – המערכת מפעילה שיבוץ ע"פ פרטי הארגון כפי שהוזנו ופרטי המתנדבים כפי שכבר קיימים אצלה
- * המערכת מתריעה למנהל על אילוצים חדשים וזמניים שהוכנסו לאחרונה ע"י מתנדב, המנהל יכול להתעלם או לבחור לשבת מערכת חדשה לתקופה הקרובה
- * המערכת שולחת הודעה לכתובת המייל של כל מתנדב על פרטי ההתנדבות שלו וכן אם יש שינויים זמניים – מעדכנת אותו

תיאור הטכנולוגיה:

צד שרת: שפת תכנות בצד השרת: c#

צד לקוח: שפת תכנות בצד הלקוח: angular

מסד נתונים: sqlServer

פרוטוקולי תקשורת:

לוחות זמנים:

- חקר המצב הקיים – ספטמבר
- הגדרת הדרישות – ספטמבר
- אפיון המערכת – אוקטובר
- אפיון בסיס הנתונים – נובמבר
- עיצוב המערכת – דצמבר
- בניית התוכנה – ינואר, פברואר
- בדיקות – מרץ
- הכנת תיק פרויקט – אפריל
- הטמעת המערכת – מאי
- הגשת פרויקט סופי – מאי

מנחות פרויקט: שירה בר-און, יעל עמר

חתימת הסטודנט:

תמר

חתימת רכז המגמה:

תודות (:

בראש ובראשונה, בהתחלה ובסוף, תודה לבורא העולם, שממנו התובנות והרעיונות. בנוסף, לעונג הוא לי להודות לאנשים הנפלאים שסייעו רבות בכל שלבי בניית הפרויקט עד הוצאתו לאוויר העולם.

בלעדיכם לא יכל הפרויקט להיבנות!

למנחה היקרה, הגב' יעל עמר, שסיעה לי לאורך כל הדרך בהכנת הפרויקט, ייעצה, בדקה ונתנה חוות דעת וביקורת בונה החל משלבי התכנון והביצוע ועד לשלב הסופי של הגשת הפרויקט המוגמר. לרכזת המגמה הגב' מרים קאסמן היקרה על הסבלנות, הייעוץ, ההקשבה והעזרה מעל ומעבר בכל שלבי הפרויקט.

למורות היקרות והמסורות שליוו אותנו במהלך הלימודים והעבודות, על החומרים, ההכוונה והאכפתיות: המ' רחלי לוי, המ' אסתי קירשנבויים, המ' אלישבע כהן, והמ' שירה בר און.

תקציר / מבוא:

בבואי לחשוב על נושא הפרויקט שלי – שאפתי לבחור בנושא שיקדם אותי ויביא לידי ביטוי את שעות הלימוד הרבות וההשקעה העצומה לאורך כל תקופת הלימודים. בנוסף לרצון של הפיתוח המקצועי והלמידה העצמית - היה לי חשוב לבנות פרויקט שימושי שיוכל לעזור לאנשים ויחסוך להם מזמנם.

לכן בחרתי בפרויקט זה, המשלב צד שרת הדורש לוגיקה ומחשבה רבה לצד צד לקוח נח וידידותי למשתמש.

נהייתי לעבוד על הפרויקט ולראות את התוצאות שלב אחרי שלב. אני מקווה שהאתר יביא תועלת ונוחות בעז"ה.

הרקע לפרויקט:

"יד ליד" הינו מערך התנדבות שבועי המופעל ע"י מחלקת הרווחה בעיריית צפת.

"יד ליד" שם לעצמו מטרה – להיות לעזר למשפחות הזקוקות לסיוע יומיומי בעקבות התמודדות עם קושי רפואי ל"ע, אובדן, משבר או עומס. מלבד עצם ההתמודדות הכרוכה בניסיון שחוות משפחות אלו בפן הרפואי, הכלכלי והרגשי – ישנה חזית נוספת – עליהם להמשיך לנהל אורח חיים שגרתי למען בני המשפחה הנוספים שבבית. יש צורך בבגדים נקיים, אוכל מוכן, בית נעים לחיות בו ושמירה על ילדים קטנים אם ישנם.

וכאן נכנס לתמונה "יד ליד", התומך ומעניק כוח למשפחות אלו. מדי שבוע נשלחים מתנדבים לבתי המשפחות וזאת ע"מ לאפשר הפוגה קלה מהפעילות האינטנסיבית בה הם נתונים.

מערך ההתנדבות פועל בתיאום של מספר גופי חסד הפועלים בתחומים שונים: ביקור חולים, שמירה על תפקוד תקין של הבית מבחינה טכנית – ניקיון, הכנת מזון, שמרטפות וכדומה.

מתנדב הרוצה לקחת חלק בפעילות הברוכה – יוצר קשר עם מחלקת הרווחה, ושם דואגים לקשר אותו לארגון הרצוי שישבץ אותו במערכת. כך, בשיתוף פעולה עילאי - נוצר עולם שלם של חסד.

ב"יד ליד" מודעים לחשיבות הארגון ומשתדלים להגדיל עוד ועוד את היקף הפעילות כדי לתמוך באנשים נוספים ובתחומים נוספים. אך כל עוד מנוהלים הארגונים באופן ידני – קלסרי רישום / טבלאות אקסל, ובראש כל ארגון עומד אדם שאמור לנהל את גיוס המתנדבים, רישום הנזקקים, ושיבוץ לוח זמנים בהתאמה לכל הצדדים ובסנכרון מלא עם שאר הארגונים – נעשה הדבר מסובך מאוד עד בלתי אפשרי.

לכן, הגתה מחלקת הרווחה את הרעיון שבעז"ה יביא תועלת לכל הנוגעים בדבר: הקמת אתר אינטרנטי שיתמוך בפעילות הארגון ובהגדלתו. האתר יכלול רישום מתנדבים, מקבלי סיוע ומנהלי ארגונים, ואזור אישי לכל אחד מהצדדים - בו יוכל להכניס אילוצים, לצפות במערכת המעודכנת הרלוונטית עבורו, לעדכן על שינויים פתאומיים בלוח הזמנים, ובנוסף – אפשרות יעילה במיוחד עבור מנהלי הארגונים – בניית מערכת זמנים חכמה עבור הארגון, תוך התחשבות אופטימלית באילוצים השונים ועדכון על שינויים חדשים בזמן אמת. כדי להבטיח את בטיחות הנתונים הנכנסים לאתר – נקבע שרק מתנדב יוכל להרשם לאתר באופן עצמאי. שאר המשתמשים יוכלו להרשם רק ע"י המנהל הרלוונטי עבורם – מקבלי סיוע נרשמים ע"י מנהל ארגון, ומנהל ארגון חדש נרשם ע"י מנהל "יד ליד". לאחר הרישום ניתן להכנס לאזור האישי ע"י תעודת זהות וסיסמה שנקבעה ע"י המערכת בעת ההרשמה.

תהליך המחקר:

כדי להגיע לדיוק מקסימלי שישרת את מטרות האתר – השקעתי שעות רבות בחקירת הנושא, הן בתחום התיאורטי והן בתחום התכנותי.

בתחום התיאורטי ערכתי ראיונות עם העוסקים בתחום – מנהלי מחלקת הרווחה, מנהלי ארגונים שונים וכן מתנדבים ומקבלי סיוע. חקרתי לעומק מהן הציפיות של כל צד מהאתר, מה יתרום להתנדבות רבה, מה יקל על המנהלים את המלאכה וכדומה. התשובות שקיבלתי עזרו לי להבין לעומק איך עובד מערך ההתנדבות ומהם הצרכים החשובים שעלי להתייחס אליהם במהלך העבודה. כך ניגשתי לבניית מסד הנתונים בצורה הנכונה והמתאימה ביותר, וכן לבניית צד הלקוח כך שיהיה נעים ונוח למשתמש.

בתחום התכנותי – במקביל לעבודת המחקר התיאורטי – חקרתי את סוגי האלגוריתמים הניתנים לפתרון בעיית השיבוץ. מתוך המחקר הגעתי לכך שלא ניתן למצוא פתרון טוב לבעיה זו בסיבוכיות זמן פולינומינאלית, אלא, אם נרצה למצוא פתרון שיענה על כל הקריטריונים - נצטרך לעבור על כל אפשרויות השיבוץ שקיימות ולבדוק האם השיבוץ תואם לכלל הקריטריונים. אלגוריתם כזה הינו אלגוריתם בסיבוכיות זמן מעריכית ולא ניתן למחשב זאת. כלומר, בשיבוץ רחב של כמות נתונים סבירה ומעלה ייקח כמה שנים עד שהאלגוריתם יגמור לעבוד.

לכן, החלטתי לממש אלגוריתם מתחום הבינה המלאכותית – אלגוריתם ממשפחת האלגוריתמים הגנטיים.

השם "אלגוריתמים גנטיים" מתאר משפחה של אלגוריתמים, בהם משלבים זה בזה פתרונות אפשריים לבעיה, ומפעילים הליכים של ברירה טבעית כדי לבחור את המועמדים שיעברו לשלבים הבאים. במודל תכנותי זה יוצרים קהילה של פרטים אשר מאופיינים בכרומוזומים, ומעבירים אותם "תהליך אבולוציוני". לאחר יצירת קהילת הפרטים הראשונה, מדרגים כל פרט על מנת למצוא את הפרטים הטובים ביותר. לאחר מכן, עורכים מיזוג בין פרטים אלו על מנת ליצור קהילה חדשה, טובה במקצת מקודמתה, ומעבירים אותה את אותו התהליך. ההנחה כאן היא שמיזוג בין שני פתרונות טובים לבעיה יניב בדרך-כלל פתרון טוב יותר. כמו בגנטיקה ביולוגית, שני הורים בעלי סט גנים טובים יוצרים צאצא שגם הוא עליון מבחינה גנטית. לאחר מספר חזרות על הפעולה, הפרטים ישתפרו דרמטית ויציגו את הפתרון הטוב ביותר, או פתרון טוב באופן יחסי לבעיה הנתונה.

מחקר נוסף שערכתי בתחום התכנותי הוא האופן של קריאת נתונים מקובץ excel במבנה לא ידוע, לצורך רישום מהיר של מתנדבים ומקבלי סיוע עבור ארגונים בהם המידע כבר ממוחשב, כדי לחסוך להם את הצורך בעבודה ידנית וכפולה.

במהלך המחקר התוודעתי למספר שינויים שעלי לערוך בתוכנית המקורית המוצגת בהצעת הפרויקט: לא נכון לשייך מתנדב לארגון באופן שרירותי ככתוב בהצעה, אלא על המתנדב לבחור בעצמו את הארגונים שברצונו להתנדב בהם. דבר נוסף שלא הודגש הוא הצורך בסנכרון לוחות הזמנים בין כל הארגונים, כלומר בעת שיבוץ מערכת לארגון מסוים - יש לוודא שלא קיימות חפיפות במערכת עבור מתנדבים ומקבלי סיוע בארגון זה הרשומים ביותר מארגון אחד, ולא ישובצו התנדבויות עבורם בזמן זהה ע"י ארגונים שונים. בנוסף, כאשר משתמש מכניס אילוצים חדשים לעצמו בארגון מסוים לאחר שהמערכת בארגון זה כבר משובצת – תיערך בדיקה האם המערכת הקבועה שלו מתאימה לאילוצים החדשים שהכניס, ובמידה והיא לא תואמת – הוא יוכל לבחור בין האפשרויות להמשיך במערכת למרות הקונפליקטים עד שמנהל הארגון ישבץ מחדש, או לבטל את כל האירועים הקבועים לו מהיום ועד לשיבוץ חדש. מנהל הארגון יתעדכן במייל על ההתפתחויות.

התוצאה שעומדת לפניכם מביאה לניהול מערך התנדבותי שלם באופן יעיל, מקצועי ונח ובהתאמה מקסימלית לצרכי כל הצדדים. למנהל "יד ליד" ולמנהלי הארגונים לא נותר אלא להכנס לאתר ולצפות בפרטים. החלק הלוגי וניהול המערכת עבור כל ארגון נעשה באופן אוטומטי.

סקירת ספרות:

האתרים העיקריים בהם נעזרתי במהלך החקירה וכתובת הפרויקט:

הסבר על האלגוריתם הגנטי:

<https://medium.com/analytics-vidhya/introduction-to-genetic-algorithms-in-c-with-a-real-application-72c3fc42bdf9>

בניית צד השרת – #C:

[/https://www.w3schools.com](https://www.w3schools.com)
[/https://developers.google.com](https://developers.google.com)

בניית צד הלקוח - ANGULAR:

<https://angular.io>

<https://material.angular.io>

<https://www.w3schools.com>

<https://developers.google.com>

פתרון באגים ושגיאות והסבר על קודים נוספים:

<https://stackoverflow.com>

<https://github.com>

codeproject.com

getbootstrap.com

<https://www.codegrepper.com>

מטרות – יעדים:

המטרה הראשית אליה חתרת היא להקל על כל הצדדים הקשורים למערך ההתנדבות – מתנדבים, מקבלי סיוע ומנהלים, ולחסוך להם זמן, אנרגיות וכוח אדם באופן היעיל ביותר.

מטרות נוספות:

- ✓ תכנון המערכת תוך שימת דגש על כתיבה נכונה, מאורגנת ומקצועית של הקוד.
- ✓ הבאת מענה יעיל לצרכי כל אחד מהמשתמשים.
- ✓ בניית מסד נתונים מתוכנן היטב השומר את הנתונים בצורה מסודרת ויעילה.
- ✓ אחזור מהיר ונוח של הנתונים.
- ✓ ממשק קל ונוח לתפעול, וכן מעוצב ויזווי למשתמש.
- ✓ כתיבת אלגוריתם יעיל ומדויק לשיבוץ מערכת שעות אופטימלית עבור כל ארגון.

בפן המקצועי – שאפתי לרכוש ידע מקיף ונרחב בשימוש בטכנולוגיות כגון angular, web api ועוד, וכן יכולת למידה עצמית שאזדקק לה רבות בשטח העבודה בעז"ה.

אתגרים:

האתגרים שניצבו בפני במהלך פיתוח האתר היו הן בשטח התכנותי - כתיבת הקוד עצמו, והן בתחום המחקר - איסוף מידע ותכנון שלבי הפעולה.

האתגר העיקרי בתחום התכנותי היה הטמעת האלגוריתם הגנטי והתאמתו לנתוני האתר. בתחום המחקר נדרשתי ללמידה מעמיקה של אופן התנהלות מערכי התנדבות כאלו.

גם במהלך כתיבת הפרויקט נתקלתי באתגרים רבים, שהקנו לי ניסיון ומיומנויות חשובות כמו לימוד חומר חדש, התמצאות בשפות, פתירת באגים, לימוד קודים קיימים ולמידה עצמית.

מדדי הצלחה למערכת:

שיבוץ מערכת שעות אופטימלית, הגיונית ברמה של 100% ומתאימה לאילוצי כל הצדדים ברמה של מעל 80% - יהווה הצלחה של המערכת.
שיבוץ ברמה פחותה מכך יורה על חוסר הצלחה.

רקע תיאורטי:

אופן שיבוץ המערכת נעשה באמצעות אילוצי זמן ומקום – זמן אפשרי ופנוי עבור המתנדבים, עבור מקבלי הסיוע ועבור תאריכי הפעילות של הארגון, ומקום – חיפוש של נזקק הגר קרוב ככל האפשר למתנדב.
השיבוץ הנבחר יהיה השיבוץ המקיים את מקסימום ההתאמה האפשרית לאילוצים.

תיאור מצב קיים:

נכון להיום, מנוהל מערך ההתנדבות באופן ממוחשב חלקית, הכולל מס' גיליונות excel לפרטי הארגונים, המתנדבים והנזקקים.
רוב הארגונים מתנהלים בצורה דומה, כאשר גיוס ורישום המתנדבים נעשה באמצעות שיחת טלפון למנהל הארגון / למזכירה, וחלקם אף באופן ידני – קלסרי רישום וטבלאות בדפים.
שיבוץ מערכת עבור ארגון נעשה ע"י טבלאות ודפים, בצורה מסורבלת וארוכה, שיכולה להיות שגויה לפעמים וכמעט בלתי אפשרי להתחשב בהעדפות של כל הצדדים, מה שמוריד את אחוז המתנדבים.
כמו כן, כל פעולה שרוצה מתנדב / מקבל סיוע לבצע – כגון הוספת אילוצים, הודעה על ביטול וכו' – מסורבלת למדי ונעשית ע"י שיחות טלפון לכל הנוגעים בדבר.

ניתוח חלופות מערכת:

בבואי ליעל את ההתנהלות הנוכחית של מערך ההתנדבות, היה ברור האופן של קבלת הפרטים מהמשתמש.
כדי לשבץ את מערכת הארגון באופן אוטומטי עמדו לפני מס' אפשרויות:

1. שיבוץ ללא אלגוריתם מתקדם, אלא באופן "ידני", כמו לדוג' לבצע סדרת פעולות של `else if` שלבסוף יביאו שיבוץ כלשהו.

דרך זו אינה טובה מספיק, מכמה סיבות:

ראשית, בשיטה זו לא נוכל להגיע באמת לשיבוץ הטוב ביותר האפשרי באמצעות קוד, משום שקשה מאוד "לחזור אחורה" ולשנות שיבוץ שכבר בוצע. שנית, כתיבת הקוד תהיה מייגעת ולא יעילה בכלל, ושלישית - הסיבוכיות תהיה גרועה.

2. שימוש באלגוריתם ההונגרי - אלגוריתם המאפשר למזער עלויות בבעיית אופטימיזציה על בסיס תכנות ליניארי.
מטרת השיטה ההונגרית היא למצוא את העלות המינימלית של מכלול משימות שיש לבצע על ידי האנשים המתאימים ביותר.

מבחינת סיבוכיות הזמן – האלגוריתם ההונגרי מהווה פתרון מצוין, שכן הוא פותר את בעיית ההשמה בזמן פולינומי. אך לאחר המחקר שביצעתי על אופן התנהלות הארגונים – הגעתי למסקנה שהוא לא מתאים לבעיה זו, מכיוון שהאלגוריתם מספק פתרון מסוג "אחד על אחד", כלומר ישובץ מתנדב אחד בלבד לכל נזקק ונזקק אחד לכל מתנדב, מה שלא מספק את המטרה הדרושה – לאפשר להתנדב ולקבל סיוע כמה שיותר. לכן, לאחר חשיבה ומחקר מעמיק – החלטתי להשתמש באלגוריתם הבא – אלגוריתם גנטי:

תיאור החלופה הנבחרת, כולל נימוקים לבחירתה:

האלגוריתם הגנטי הוא אלגוריתם מתחום הבינה המלאכותית. אלגוריתם זה עוסק בבעיות פולינומיאליות, כלומר – הוא לא סדרת פעולות שבוודאות תיתן את התוצאה הרצויה, אלא רצף פעולות כך שכל פעולה אמורה לשפר את המצב הקודם. האלגוריתם נעצר באופן יזום כאשר הגענו לתוצאה משביעת רצון, גם אם לא תוצאה מושלמת, או כאשר הוא עובד זמן רב מדי.

רובם המוחלט של האלגוריתמים המודרניים מתחום הבינה המלאכותית נותנים תוצאה על פי דוגמאות. כלומר, על מנת שהאלגוריתם יעבוד כמו שצריך – יש לתת לו סט רחב מאוד של דוגמאות ושיבוצים, וכך האלגוריתם "לומד" את הטכניקה כיצד לשבץ בצורה יעילה.

במסגרת הפרויקט שלי כסטודנטית אין לי אפשרות לתת סט כזה של שיבוצים, ולכן אלגוריתם גנטי נותן את המענה: הוא גם מתחום הבינה המלאכותית, וגם לא דורש מגוון דוגמאות ומידע.

העיקרון של אלגוריתם גנטי הוא כמו פעילות של השבחה גנטית שבה נוטלים מכל יצור את הגן הטוב יותר ומעבירים אותו לדור הבא, וכך משתבח היצור מדור לדור.

מהלך האלגוריתם כולל:

שילוב אלמנטים של פתרונות אפשריים לבעיה (על מנת לקבל פתרון מוצלח יותר), והפעלת הליכים של ברירה מלאכותית כדי לבחור את המועמדים שיעברו לשלבים הבאים. לכן, אם ניקח אוכלוסייה של פתרונות ונבחר מתוכם רק את המתאימים ביותר לפתרון הבעיה, נמזג אותם אחד עם השני ונוסיף מעט שינויים – נקבל דור חדש של פתרונות הקרוב צעד נוסף לפתרון הבעיה הנתונה. נחזור על התהליך מספר רב של פעמים (דורות) ולבסוף נגיע לפתרון הטוב ביותר.

אפיון המערכת שהוגדרה:

ניתוח דרישות המערכת:

על המערכת לעמוד בדרישות הבאות:

- ← מחשוב השירות ללקוח – מתנדב / נזקק / מנהל
- ← כתיבה בסטנדרטים מקצועיים
- ← ממשק ידידותי ונוח למשתמש
- ← תגובה מהירה ככל הניתן למשתמש
- ← כתיבת הקודים והאלגוריתם בצורה מדויקת ומותאמת לצרכים ובסיבוכיות היעילה ביותר

מודול המערכת:

- ← קליטת פרטי הארגונים והמנהלים
- ← קליטת פרטי המתנדבים ומקבלי הסיוע – באופן ידני או באמצעות העלאת קובץ excel
- ← קליטת אילוצים מכל אחד מהמשתמשים הנ"ל
- ← בדיקה האם אילוצים חדשים מהווים קונפליקט ללוח הקיים, ואפשרות לביטול השיבוץ הרלוונטי לפי העדפת המשתמש.
- ← קריאה למערכת לבצע שיבוץ בארגון מסוים
- ← מציאת השיבוץ האופטימלי ככל שניתן ע"י אלגוריתם גנטי
- ← הזנת פרטי השיבוץ במסד הנתונים
- ← צפיה בפרטי השיבוץ עם אפשרות לשינוי

אפיון פונקציונלי:

- TimeTableChromosome** – יצירת רשימה של משבצות זמן שירכיבו שיבוץ למשך שבוע ימים.
- RandomIndex** – פונקציה המקבלת מספר – limit ומחזירה מספר אקראי כלשהו בין 0 ל limit .
- Generate** - אחראית על יצירת לוח זמנים אקראי.
- Clone** – משכפלת את השיבוץ של לוח זמנים (לצורך שילוב עם לוח זמנים נוסף).
- Mutate** – מגרילה מספר המציין את סוג השינוי שיש לבצע ברשימה – שינוי הנתונים במשבצת זמן מסוימת, מחיקת משבצת או הוספת משבצת חדשה. מבצעת את השינוי.
- Crossover** – ממזגת 2 לוחות זמנים בכמות אקראית בלוח הזמנים הנוכחי.
- Evaluate** – הערכת לוח זמנים ע"י בדיקת מספר הקונפליקטים בלוח.
- EliteSelection** – בחירת לוחות הזמנים עם הציון הגבוה ביותר לצורך מיזוג.
- createNew** – יצירת מערכת זמנים חדשה.
- Gentic** – יצירת לוח זמנים ע"י שימוש באלגוריתם הגנטי.
- ReadFromExcel** – מזמנת את אלגוריתם הקריאה מ excel.

ביצועים עיקריים:

- מנהל "יד ליד" רושם למערכת את פרטי הארגונים המשתתפים בפעילות וכן את פרטי המנהלים.
- המערכת מזינה את הנתונים לאחר בדיקות תקינות בטבלאות המתאימות במסד הנתונים. לאחר מכן מתאפשרת הכניסה למנהל ארגון.
- מנהל ארגון נכנס למערכת ורושם את פרטי המתנדבים ומקבלי הסיוע בארגון – באופן ידני או ע"י העלאת קובץ אקסל עם פרטיהם. המערכת קוראת את הקובץ, מזינה את הנתונים בטבלה המתאימה במסד הנתונים ויוצרת קובץ אקסל חדש המכיל את הסיסמאות האקראיות שנוצרו עבור כל אחד מהנרשמים. הקובץ זמין להורדה למנהל הרלוונטי.
- מתנדב נרשם / נכנס למערכת ובוחר את הארגונים המתאימים לו בהם הוא מעוניין להתנדב.
- מתנדב / מקבל סיוע נכנסים למערכת ומזינים את האילוצים שלהם – ימים ושעות בהם נח להם להתנדב.

המערכת מתבקשת ליצור שיבוץ מערכת לארגון מסוים. המערכת משבצת שיבוץ אופטימלי אפשרי ע"י אלגוריתם גנטי ומזינה את פרטי השיבוץ במסד הנתונים. המתנדבים ומקבלי הסיוע יכולים לצפות בכל רגע נתון במערכת הרלוונטית עבורם, וכן מנהל הארגון יכול לצפות בכל המערכת של ארגון זה. מתנדב / מקבל סיוע / מנהל ארגון - יכולים לבטל התנדבות ביום מסוים, כאשר פעולה זו מבטלת את השיבוץ המסוים ביום הרלוונטי ממערכת הארגון וכן מעדכנת במייל את מנהל הארגון. מנהל ארגון יכול לשבץ התנדבות חדשה באופן ידני בלוח הזמנים של הארגון. מתנדב / מקבל סיוע מכניס אילוצים חדשים לארגון מסוים לאחר שהמערכת כבר משובצת בו, ובמידה ואילוצים אלו מהווים קונפליקט ללוח הקיים – יכול לבחור באפשרות לבטל את השיבוץ שלו עד שיערך שיבוץ חדש, או להמשיך כך.

תיאור הארכיטקטורה:

הארכיטקטורה של הפתרון המוצעת בפורמט של Top-Down Level Design:

החלוקה ל 3 שכבות נועדה להפריד באופן מוחלט בין הלוגיקה של הפרויקט לבין הנתונים עצמם. הפרדה זו מאפשרת לבצע שינויים בכל אחת מהשכבות בלי תלות ובלי זעזועים בשכבות האחרות, וכן לגלות כשלים בקלות ולטפל בהם בשכבה אחת בלבד.

טכנולוגיות כתיבה:

צד שרת- נכתב בשפת #C בטכנולוגיית WebApi.
צד הלקוח נכתב בשפות SCSS, HTML ו Type script – בטכנולוגיית Angular.

תיאור צד השרת:

צד השרת מחולק כמקובל לשכבות: שכבת ה DAL, שכבת ה BL, שכבת ה API ושכבת ה MODELS. פירוט:

שכבת ה DAL היא השכבה דרכה ניגשים לנתונים היושבים במסד הנתונים. היא מכילה מחלקות המייצגות את בסיס הנתונים וכן פונקציות נחוצות לתפעול. פעולות ההתקשרות עם בסיס הנתונים נעשו בטכנולוגיית EntityFramework, טכנולוגיה זו היא המקובלת לטיפול בבסיס הנתונים, כאשר היא מנתחת את בסיס הנתונים, בונה מחלקות לייצוג הטבלאות ומספקת רשימות מלאות בנתוני בסיס הנתונים.

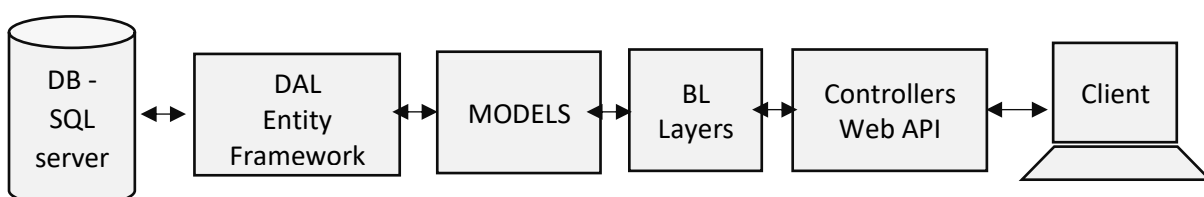
שכבת ה BL היא השכבה בה כתובה כל הלוגיקה של הפרויקט.

שכבת ה API – בשכבה זו פונקציות controllers דרכן עובר המידע מהפונקציות ב BL לצד הלקוח ולהיפך. **שכבת ה MODELS** – שכבה זו מכילה מחלקות המתארות את הנתונים ובמבנה זה מעבירים את הנתונים בין השכבות. מטרת שכבה זו היא למנוע תלות של שכבת ה BL במבנה בסיס הנתונים לטיפול הנתונים של שכבת ה MODELS ולהיפך.

בנוסף, קיימות מתודות השרת המחצינות את הפעולות שניתן לבצע בשרת. מתודות אלו משתמשות ב BL ומופעלות ע"י שכבת ה GUI בצד הלקוח.

מודל זה אמנם גורם טרחה טכנית לא מעטה בכתיבת הקוד ובתכנון הפונקציות אבל מספק קוד נקי, קל להבנה ונוח לשינויים ולשדרוגים.

תיאור הרכיבים בפתרון:



1. מסד הנתונים הבנוי מטבלאות וקשרי גומלין ביניהן
2. שכבת הגישה לנתונים באמצעות Entity Framework
3. שכבת היישויות
4. שכבת ה BL – בה כתובים האלגוריתמים
5. Web Api - פרוטוקול התקשורת בין צד הלקוח וצד השרת
6. צד לקוח – Angular, Type Script

ארכיטקטורת רשת:

לא רלוונטי

תיאור פרוטוקולי התקשורת:

HTTPS

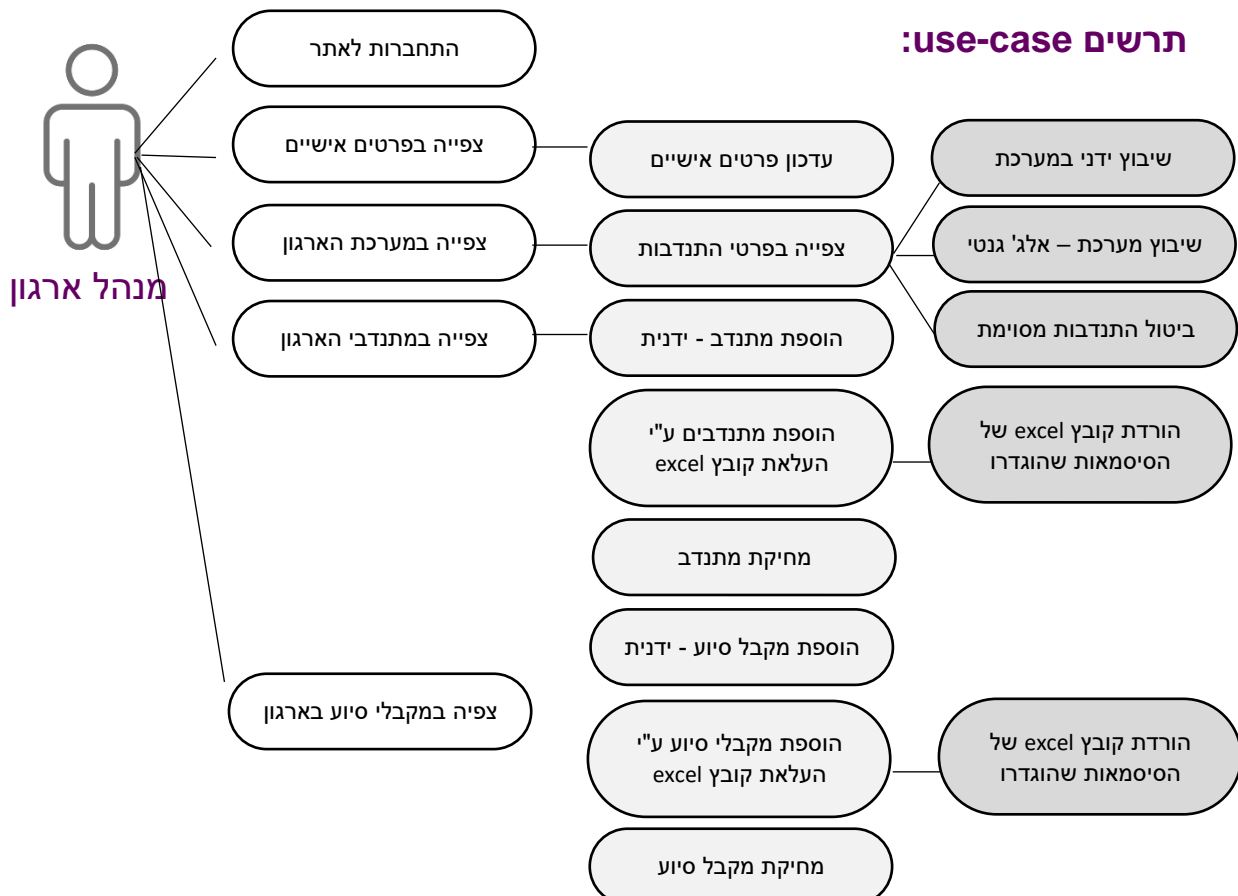
שרת – לקוח:

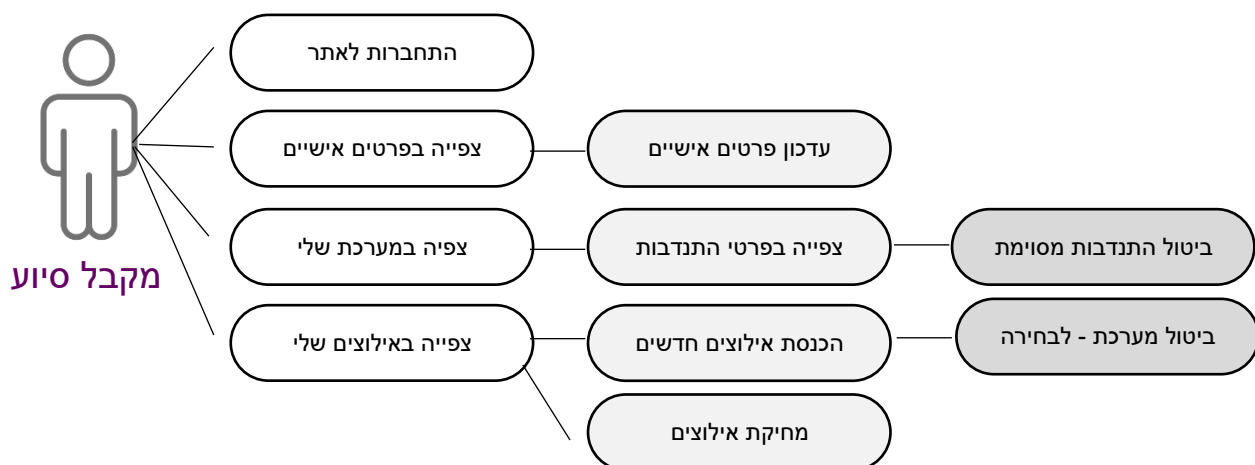
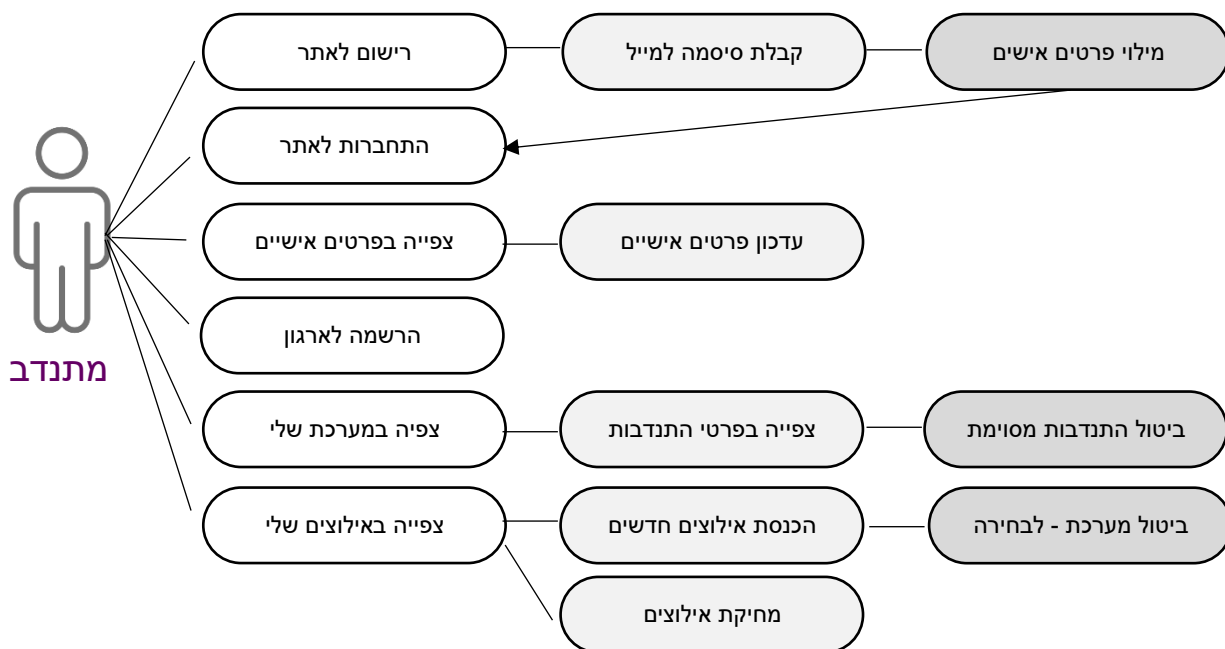
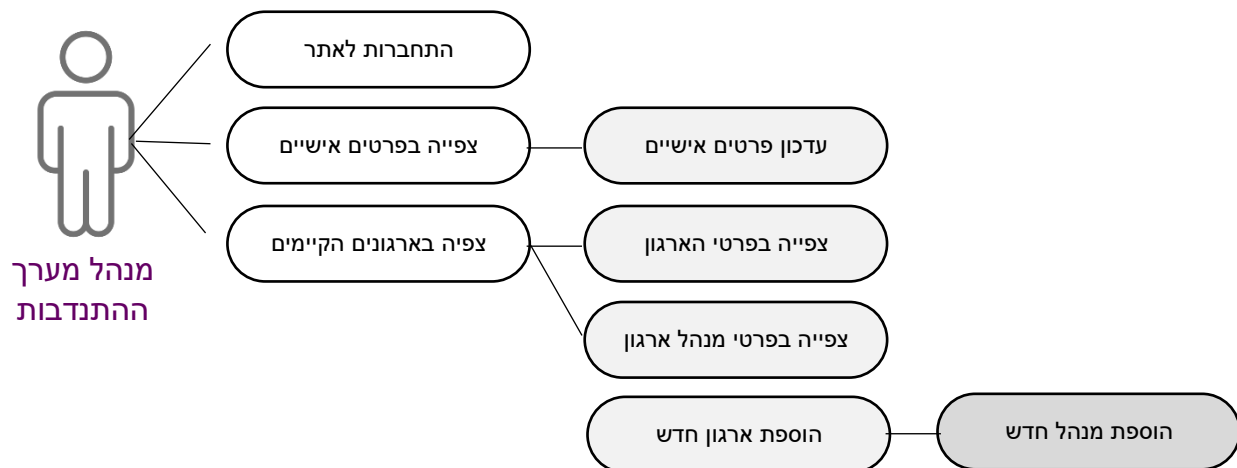
צד השרת נכתב בטכנולוגיית Api Web ובשפת C# .
צד הלקוח נכתב בשפות SCSS, HTML, Type script – בטכנולוגיית Angular.

תיאור הצפנות אם יש:

לא רלוונטי.

ניתוח ותרשים use-case של המערכת המוצעת:





רשימת use-cases

- A.** כניסה לאתר כמנהל כללי
- a. צפייה ועדכון פרטים אישיים
 - b. הכנסת פרטי ארגון חדש
 - c. הכנסת פרטי מנהל ארגון חדש
- B.** רישום לאתר כמתנדב חדש
- C.** כניסה לאתר כמתנדב קיים
- a. צפייה ועדכון פרטים אישיים
 - b. רישום כמתנדב לארגון
 - c. צפייה והוספת אילוצים
 - d. ביטול מערכת – לבחירה, במקרה שהאילוצים החדשים מהווים קונפליקט ללוח הקיים
 - e. צפייה במערכת הרלוונטית
 - f. ביטול התנדבות ביום מסוים – מעדכן את המנהל במייל אוטומטי
- D.** כניסה לאתר כמקבל סיוע קיים
- a. צפייה ועדכון פרטים אישיים
 - b. צפייה והוספת אילוצים
 - c. ביטול מערכת – לבחירה, במקרה שהאילוצים החדשים מהווים קונפליקט ללוח הקיים
 - d. צפייה במערכת הרלוונטית
 - e. ביטול התנדבות ביום מסוים – מעדכן את המנהל במייל אוטומטי
- E.** כניסה לאתר כמנהל ארגון
- a. הכנסת פרטי מקבלי הסיוע בארגון – ידנית או באמצעות העלאת קובץ excel
 - b. הכנסת פרטי מתנדבים בארגון – ידנית או באמצעות העלאת קובץ excel
 - c. צפייה במערכת הארגון
 - d. הפעלת שיבוץ מערכת בארגון
 - e. שיבוץ מערכת ידנית

תיאור ה UC העיקריים של המערכת

שם	התחברות לאתר ומועבר לאזור האישי המתאים	1
תיאור	משתמש מקליד תעודת זהות וסיסמה ומועבר לאזור האישי המתאים	
שחקנים	משתמש הרשום במערכת	
תנאי קדם	פרטי המשתמש שמורים במערכת	
היזנק	משתמש מבקש להיכנס לאזור האישי המתאים באתר	
תיאור מהלך התרחיש	נתוני המשתמש נקלטים והמערכת בודקת האם הוא קיים, במידה וכן הוא מועבר לאזור האישי המתאים	
תרחישים אלטרנטיביים	במידה והמשתמש לא נמצא במערכת מוצגת לו הודעה מתאימה	
תנאי סופי	המשתמש מועבר לאזור האישי המתאים באתר	

2	שם	הוספת ארגונים חדשים למערך ההתנדבות
	תיאור	מנהל כללי מכניס ארגון חדש למערכת
	שחקנים	מנהל המזוהה במערכת כמנהל כללי (ולא מנהל ארגון)
	תנאי קדם	פרטי המנהל שמורים במערכת, והמנהל הוא מנהל כללי
	היזנק	מנהל כללי מבקש לרשום ארגון חדש למערכת
	תיאור מהלך התרחיש	המשתמש מקליד את פרטי הארגון החדש במערכת, וכן את פרטי מנהל הארגון החדש – אם רוצה
	תרחישים אלטרנטיביים	פרטי הארגון החדש ופרטי המנהל החדש – אם הוכנסו – נשמרים במערכת.

3	שם	רישום מתנדבים חדשים לארגון מסוים
	תיאור	מנהל ארגון רוצה להכניס מתנדבים חדשים לארגונו
	שחקנים	מנהל ארגון המזוהה כמנהל ארגון (ולא כמנהל כללי)
	תנאי קדם	פרטי המנהל שמורים במערכת, והוא מנהל ארגון
	היזנק	מנהל ארגון מבקש לרשום מתנדבים חדשים לארגון אותו הוא מנהל
	תיאור מהלך התרחיש	המשתמש בוחר באפשרות הרצויה – הכנסת מתנדבים באופן ידני ע"י מעבר לטופס המתאים, או העלאת קובץ excel (אין חובה על תבנית קבועה של הקובץ) המכיל את פרטי המתנדבים במידה ובחר באפשרות העלאת קובץ: המערכת מנתחת את הקובץ ושומרת את המתנדבים החדשים, תוך הוספה אוטומטית של הנתונים הרלוונטים, כדי שיזוהו כמתנדבים השייכים לארגון זה
	תרחישים אלטרנטיביים	אם נמצאה תעודת זהות הקיימת כבר במערכת – המערכת לא מוסיפה אותה שוב אלא מוסיפה עבודה רק רשומה בטבלה המתאימה כדי שיזוהה כרשום בארגון זה אם פענח הקובץ נכשל - מוצגת הודעה מתאימה
	תנאי סופי	המתנדבים החדשים נרשמים במערכת וכן מזוהים כשייכים לארגון זה

4	שם	רישום מקבלי סיוע חדשים לארגון מסוים
	תיאור	מנהל ארגון רוצה להכניס מקבלי סיוע חדשים לארגונו
	שחקנים	מנהל ארגון המזוהה כמנהל ארגון (ולא כמנהל כללי)
	תנאי קדם	פרטי המנהל שמורים במערכת, והוא מנהל ארגון
	היזנק	מנהל ארגון מבקש לרשום מקבלי סיוע חדשים לארגון אותו הוא מנהל
	תיאור מהלך התרחיש	המשתמש בוחר באפשרות הרצויה – הכנסת מקבלי סיוע באופן ידני ע"י מעבר לטופס המתאים, או העלאת קובץ excel (אין חובה על תבנית קבועה של הקובץ) המכיל את פרטי מקבלי הסיוע.

	תרחישים אלטרנטיביים	אם נמצאה תעודת זהות הקיימת כבר במערכת – המערכת לא מוסיפה אותה שוב אלא מוסיפה עבודה רק רשומה בטבלה המתאימה כדי שיזוהה כרשום בארגון זה אם פענח הקובץ נכשל - מוצגת הודעה מתאימה
	תנאי סופי	מקבלי הסיוע החדשים נרשמים במערכת וכן מזוהים כשייכים לארגון זה
		במידה ובחר באפשרות העלאת קובץ: המערכת מנתחת את הקובץ ושומרת את מקבלי הסיוע החדשים, תוך הוספה אוטומטית של הנתונים הרלוונטים, כדי שיזוהו כמקבלי סיוע השייכים לארגון זה

5	שם	הרשמה לאתר כמתנדב חדש
	תיאור	מתנדב חדש הרוצה להירשם למערכת
	שחקנים	מתנדב חדש שלא רשום במערכת
	תנאי קדם	המתנדב לא רשום במערכת
	היזנק	מתנדב מבקש להירשם למערכת
	תיאור מהלך התרחיש	המתנדב מקליד תעודת זהות וכתובת מייל, במידה ותעודת הזהות לא נמצאה במערכת – נשלחת לכתובת המייל שהוזנה סיסמה זמנית אותה הוא מתבקש להקליד במקום המתאים. אם הסיסמה נכונה – הוא מועבר לטופס השלמת פרטים אישיים ונרשם במערכת
	תרחישים אלטרנטיביים	אם תעודת הזהות שהוקלדה נמצאה שייכת למתנדב רשום – מוצגת הודעה מתאימה אם הסיסמה שהקליד המשתמש לא תואמת לסיסמה שנשלחה לו – לא מתאפשר תהליך השלמת הרישום
	תנאי סופי	המשתמש נרשם במערכת כמתנדב חדש ומועבר לאזור אישי מתאים

6	שם	הרשמה כמתנדב בארגון מסוים
	תיאור	מתנדב קיים הרוצה להתנדב בארגון מסוים
	שחקנים	מתנדב קיים
	תנאי קדם	פרטי המתנדב שמורים במערכת
	היזנק	מתנדב מבקש להתנדב בארגון מסוים
	תיאור מהלך התרחיש	המתנדב עובר לכרטיסיית "רישום לארגון" באזור האישי שלו, שם מוצגים לו פרטי כל הארגונים והוא יכול לבחור בארגונים בהם הוא רוצה להתנדב והמתאימים לנתונים שלו. לאחר שבחר ארגון מסוים הוא מקליד את מס' השעות השבועיות בהם הוא מעוניין בארגון זה, ונרשם כמתנדב בארגונים שבחר.
	תרחישים אלטרנטיביים	
	תנאי סופי	פרטי ההתנדבות החדשים נרשמים במערכת : תעודת הזהות של המתנדב, קוד הארגון/ים אותם בחר ומס' השעות עבור כל ארגון.

7	שם	הכנסת אילוצי זמן עבור התנדבות בארגון מסוים
	תיאור	מתנדב / מקבל סיוע רוצה להכניס אילוצי זמן – זמנים בהם הוא יכול להתנדב / לקבל סיוע בארגון מסוים
	שחקנים	מתנדב / מקבל סיוע שפרטיו שמורים במערכת
	תנאי קדם	פרטי המשתמש שמורים במערכת, והוא משובץ בארגון מסוים כמתנדב / כמקבל סיוע
	היזנק	מתנדב / מקבל סיוע מבקש להכניס אילוצים – זמנים בהם הוא יכול להתנדב / לקבל סיוע בארגון מסוים
	תיאור מהלך התרחיש	המשתמש בוחר את הארגון אליו הוא מכניס את האילוצים, בוחר שעות וימים בהם נוחה לו ההתנדבות ושומר. המערכת שומרת את האילוצים ובודקת האם הם מהווים קונפליקט ללוח קיים (אם יש) עבור התנדבויות שהמשתמש רשום בהם. במידה והם מהווים קונפליקט תוצג הודעה למשתמש והוא יבחר האם לבטל את השיבוצים המביאים לקונפליקט או להמשיך במערכת הקיימת.
	תרחישים אלטרנטיביים	משתמש שלא רשום באף ארגון – לא יכול להכניס אילוצים
	תנאי סופי	הפרטים שהוכנסו נשמרים במערכת ומוצגים למשתמשים הרלוונטים

8	שם	שיבוץ מערכת באופן ידני בארגון מסוים
	תיאור	מנהל ארגון רוצה לשבץ מערכת באופן ידני בארגון שלו
	שחקנים	מנהל ארגון המזוהה כמנהל ארגון הצריך שיבוץ (ולא כמנהל כללי)
	תנאי קדם	פרטי המנהל שמורים במערכת, והוא מנהל ארגון הצריך שיבוץ
	היזנק	מנהל ארגון מבקש לשבץ מערכת באופן ידני בארגון שלו
	תיאור מהלך התרחיש	מנהל הארגון בוחר (מתוך אלו הרשומים בארגון) את מקבל הסיוע, את המתנדב, את טווח התאריכים, את היום / ימים בשבוע ואת שעות ההתחלה והסיום, והפרטים נשמרים במערכת.
	תרחישים אלטרנטיביים	
	תנאי סופי	הפרטים שהוכנסו נשמרים במערכת ומוצגים למשתמשים הרלוונטים

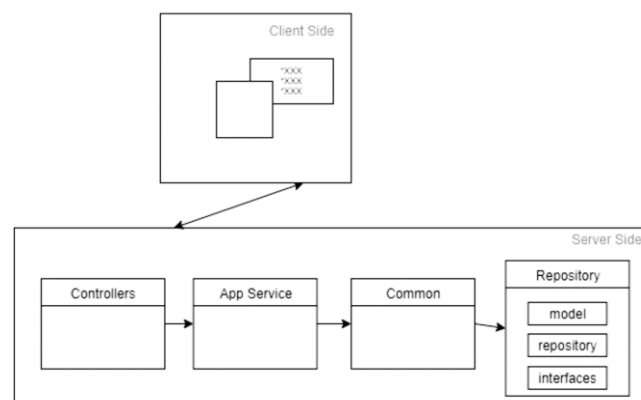
9	שם	שיבוץ מערכת חכמה בארגון מסוים
	תיאור	מנהל ארגון רוצה לשבץ מערכת חכמה בארגון שלו
	שחקנים	מנהל ארגון המזוהה כמנהל ארגון הצריך שיבוץ (ולא כמנהל כללי)
	תנאי קדם	פרטי המנהל שמורים במערכת, והוא מנהל ארגון הצריך שיבוץ
	היזנק	מנהל ארגון מבקש לשבץ מערכת חכמה בארגון שלו
	תיאור מהלך התרחיש	המערכת משבצת מערכת זמנים חכמה בארגון שנבחר ומציגה למשתמשים את פרטי השיבוץ הרלוונטים עבורם
	תרחישים אלטרנטיביים	
	תנאי סופי	פרטי מערכת הזמנים נשמרים במערכת ומוצגים למשתמשים הרלוונטים

10	שם	עדכון מערכת זמנים בזמן אמת
	תיאור	משתמש צופה במערכת הזמנים שלו ומעונין לבטל התנדבות ביום מסוים
	שחקנים	מנהל ארגון המזוהה כמנהל ארגון, מקבל סיוע או נזקק
	תנאי קדם	פרטי המשתמש שמורים במערכת וקיימת עבורו מערכת זמנים
	היזנק	משתמש מבקש לבטל התנדבות ביום מסוים
	תיאור מהלך התרחיש	המשתמש בוחר את ההתנדבות המסוימת אותה הוא מעונין לבטל ובוחר באפשרות של ביטול. מיד נשלחת הודעה למנהל הארגון המתאים עם פרטי הפעולה, ההתנדבות המסוימת מתבטלת והפרטים מעודכנים במערכת
	תרחישים אלטרנטיביים	
	תנאי סופי	הפרטים המתאימים מתעדכנים במערכת, ומנהל הארגון הרלוונטי מעודכן במייל מיידי.

מבני נתונים בהם השתמשתי

- רשימת עצמים עבור כל מחלקה
- מערכי עזר בפונקציות
- מילון ערכי שמות עמודות של קובץ לצורך קריאה מקובץ excel של מתנדבים / מקבלי סיוע חדשים שהועלו ע"י מנהל ארגון.
- מטריצה להחזרת סיסמאות אקראיות למתנדבים / מקבלי סיוע חדשים שהוכנסו ע"י מנהל ארגון בהעלאת קובץ excel. המטריצה מומרת ב angular לקובץ excel זמין להורדה למנהל הרלוונטי.

עץ מודולים

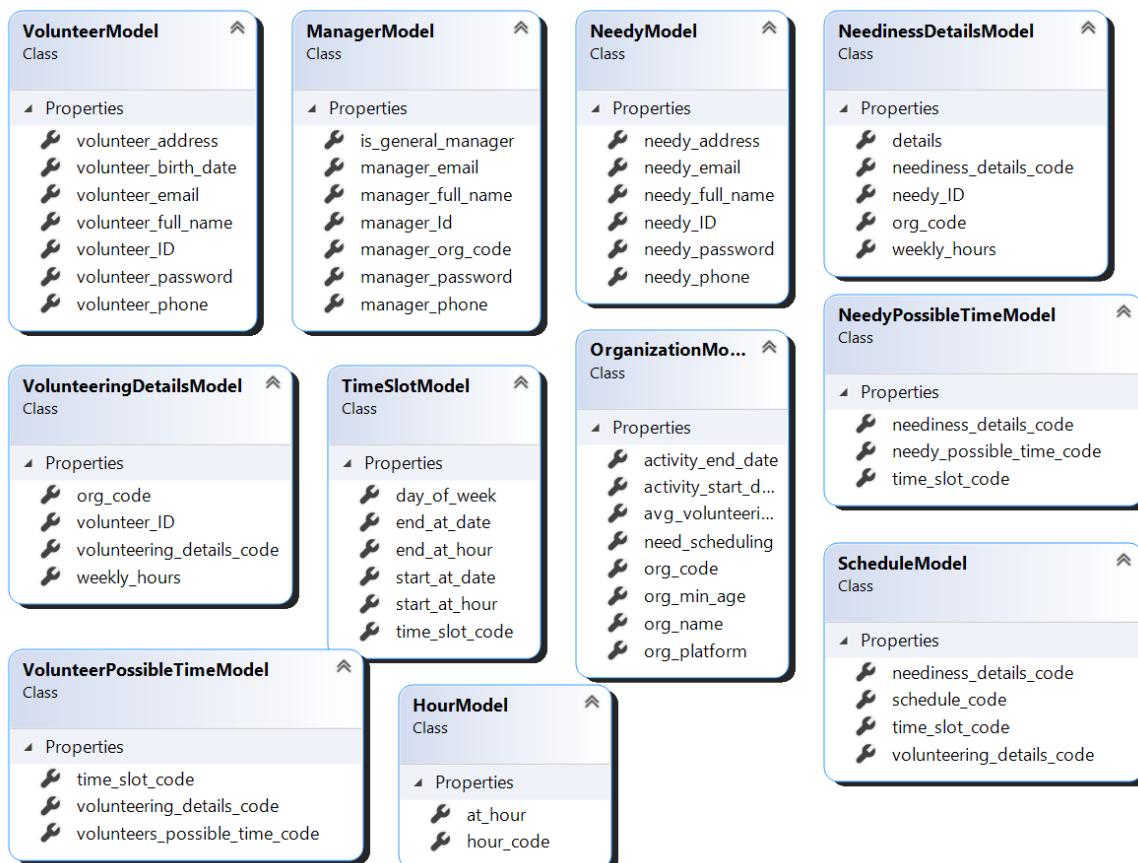


תרשים מחלקות:

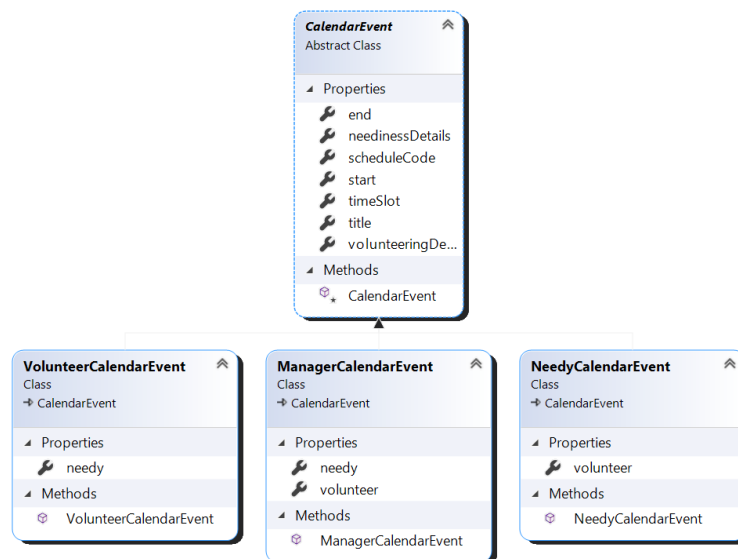
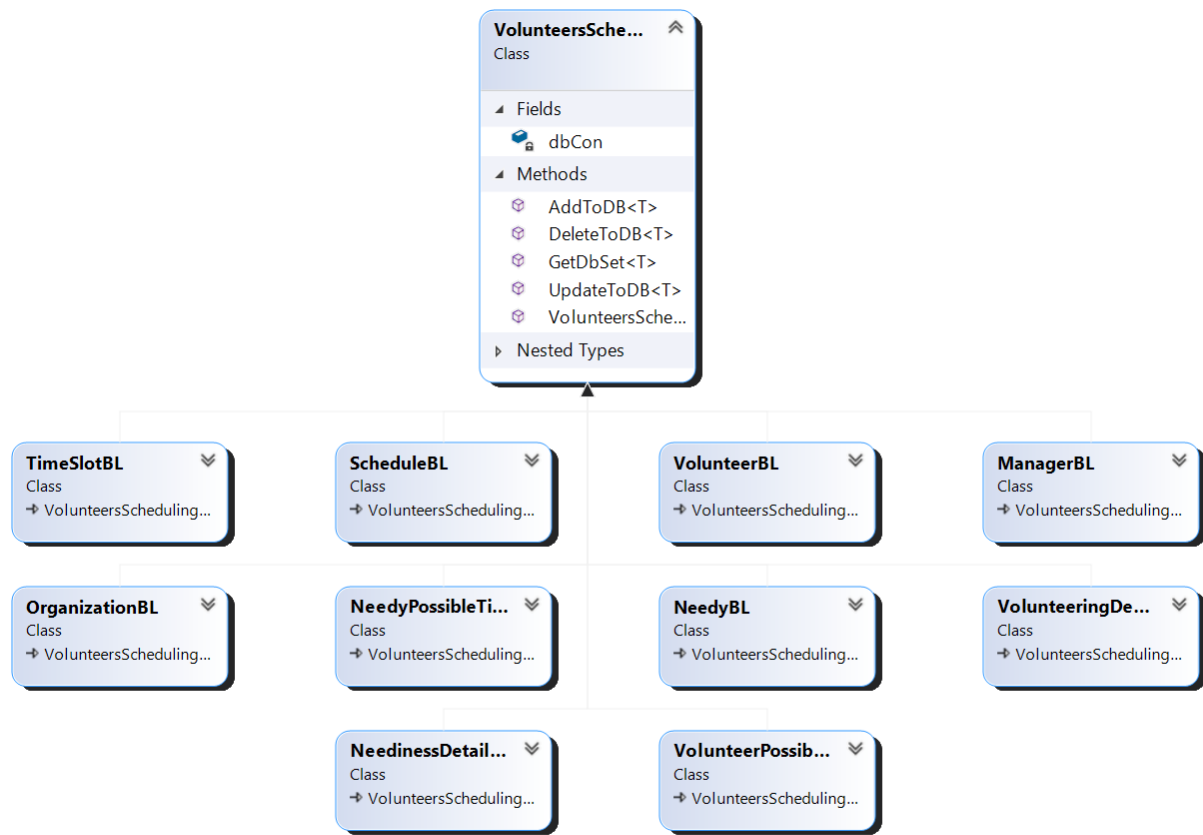
שכבת הגישה למסד הנתונים - DAL:

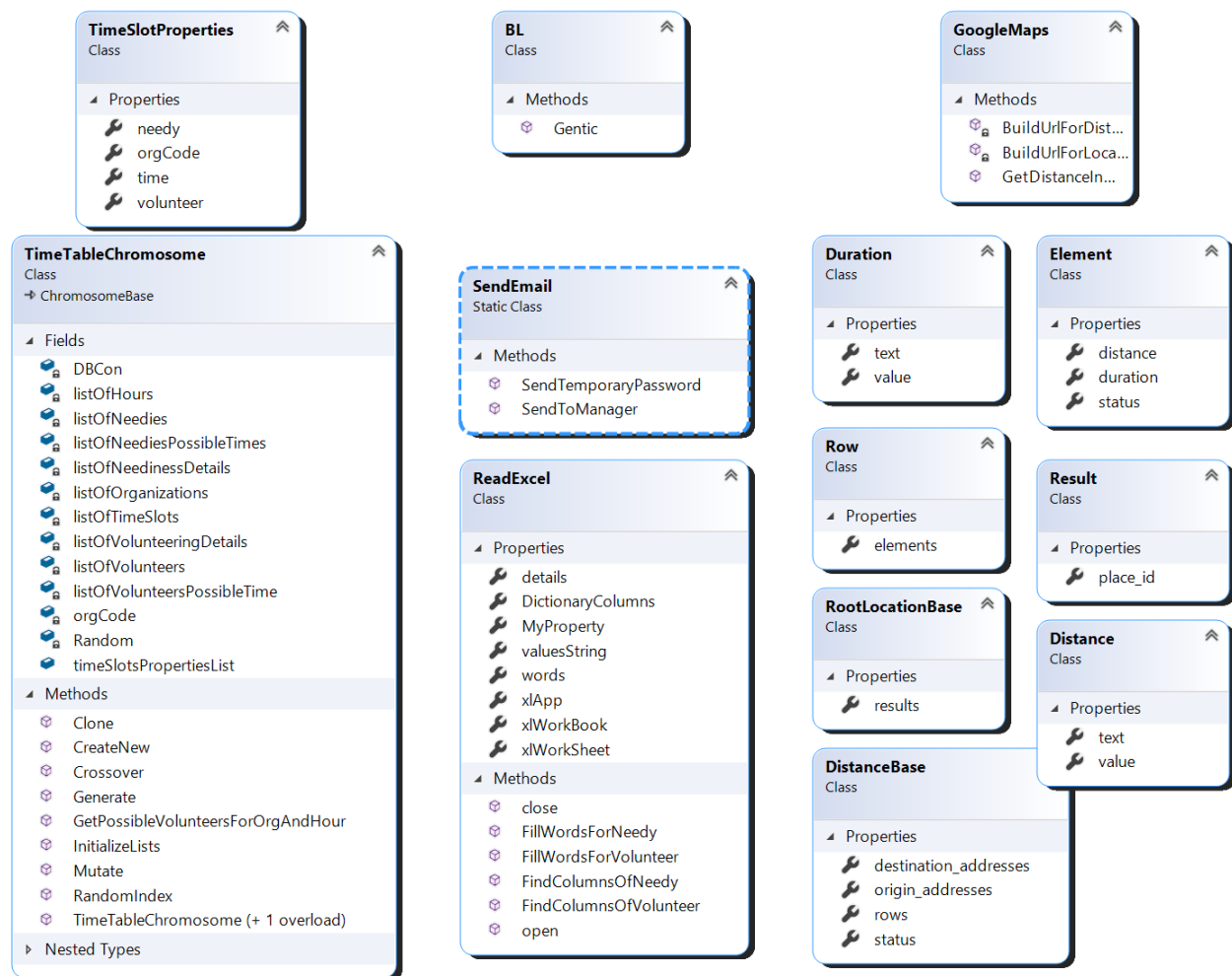


שכבת MODELS:



שכבת האלגוריתמים – BL:





תיאור המחלקות המוצעות

1. שכבת DataBase:

שכבה זו היא בסיס נתונים שישמור את הנתונים עבור הפרויקט, בנוי כקובץ של DATABASE SQL ומכיל את כל הטבלאות הנחוצות עבור הנתונים שישמשו עבור השיבוץ.

בBD נשמרים הנתונים הבאים:

- **טבלת ארגונים:** בה נשמרים פרטי הארגונים הקיימים במערכת: קוד, שם, מצע הארגון - פירוט על פעילותו, גיל מינימלי להתנדבות בו, תאריך תחילת הפעילות, תאריך סיום הפעילות, האם הארגון צריך שיבוץ או לא (ארגונים כגון איחוד הצלה לא צריכים שיבוץ אלא רק ניהול שוטף של רישום), וזמן ממוצע בדקות למשמרת של התנדבות.
- **טבלת מנהלים:** בה נשמרים פרטי המנהלים – תעודת זהות, שם, טלפון, כתובת מייל, סיסמה לאתר, קוד הארגון אותו הוא מנהל, וביט המציין האם המנהל הוא מנהל ארגון (0) או מנהל כללי של המערכת (1).
- **טבלת מתנדבים:** בה נשמרים פרטי המתנדבים: תעודת זהות, שם, כתובת, טלפון, כתובת מייל, סיסמה לאתר, תאריך לידה.

- **טבלת מקבלי סיוע:** בה נשמרים פרטי מקבלי הסיוע: תעודת זהות, שם, כתובת, טלפון, כתובת מייל, סיסמה לאתר.
- **טבלת פרטי התנדבות:** טבלה המקשרת בין מתנדב לארגון בו הוא מתנדב. בניית אותה כטבלה בפני עצמה ולא כשדה בטבלת מתנדבים כדי לאפשר התנדבות בכמה ארגונים לכל מתנדב. מכילה: קוד פרטי התנדבות – מספור רץ, תעודת זהות מתנדב, קוד ארגון, ומספר שעות התנדבות שבועיות שהמתנדב מעונין בהם בהתנדבות זו.
- **טבלת פרטי הזדקקות:** טבלה המקשרת בין מקבל סיוע לארגון בו הוא רשום. כנ"ל - בניית אותה כטבלה בפני עצמה ולא כשדה בטבלת מקבלי סיוע כדי לאפשר קבלת סיוע מכמה ארגונים לכל מקבל סיוע. מכילה: קוד פרטי הזדקקות – מספור רץ, תעודת זהות מקבל סיוע, קוד ארגון, מספר שעות התנדבות שבועיות שמקבל הסיוע זכאי להן מארגון זה, ופירוט מילולי מה כוללת ההתנדבות.
- **טבלת שעות:** בה נשמרות כל השעות הקיימות ביממה בעיגולים של רבעי שעה. עבור כל שעה נשמר קוד ושעה.
- **טבלת משבצת זמן:** טבלה הכוללת מאפיינים של משבצת זמן – קוד – מספור רץ, קוד שעת התחלה – מפתח זר מטבלת שעות, קוד שעת סיום – מפתח זר מטבלת שעות, יום בשבוע – מספר מ' עד 7 המציין את היום בשבוע, תאריך התחלה ותאריך סיום.
- **טבלת זמן אפשרי למתנדב:** טבלה המכילה אילוצי מתנדב – זמנים בהם הוא יכול להתנדב בארגון מסוים: קוד – מספור רץ, קוד פרטי התנדבות – מפתח זר מטבלת פרטי התנדבות, קוד משבצת זמן – מפתח זר מטבלת משבצת זמן.
- **טבלת זמן אפשרי למקבל סיוע:** טבלה המכילה אילוצי מקבל סיוע – זמנים בהם הוא יכול לקבל סיוע מארגון מסוים: קוד – מספור רץ, קוד פרטי הזדקקות – מפתח זר מטבלת פרטי הזדקקות, קוד משבצת זמן – מפתח זר מטבלת משבצת זמן.
- **טבלת לוח זמנים:** טבלה המכילה את פרטי מערכות הזמנים של הארגונים. כוללת: קוד – מספור רץ, קוד פרטי הזדקקות – מפתח זר מטבלת פרטי הזדקקות, קוד פרטי התנדבות – מפתח זר מטבלת פרטי התנדבות, קוד משבצת זמן – מפתח זר מטבלת משבצת זמן.

2. שכבת ה DAL – שכבת הגישה למסד הנתונים:

שכבה זו היא השכבה האחראית על הגישה לנתונים השמורים במסד הנתונים ולהעבירם לשכבה המבצעת את האלגוריתם. ע"י הפקודה Scaffold-Db נוצרות ישויות – מחלקות עבור כל אחת מהטבלאות שדרכן, עם טכנולוגית Entity Framework ניתן לגשת לנתוני מסד הנתונים.

המחלקות:

DBConnection:

תכונות המחלקה:

• **public enum ExecuteActions**

enum המכיל את 3 אופציות הגישה למסד הנתונים – insert / update / delete.

פונקציות המחלקה:

- **public DBConnection()**
קונסטרקטור היוצר עצם מסוג המחלקה.
 - **public void Execute<T>(T entity, ExecuteActions exAction) where T : class**
פונקציה גנרית המקבלת entity ופרמטר נוסף מסוג execute options המציין את הפעולה שיש לבצע ומבצעת אותה על מחלקה מסוימת. לאחר הפעולה הפונקציה מרעננת את הDB.
 - **public List<T> GetDbSet<T>() where T : class**
פונקציה גנרית השולפת נתוני טבלה מסוימת מה DB.
פונקציה זו מופעלת על שם המחלקה שממנה רוצים לשלוף ומחזירה את רשימת העצמים של המחלקה.
 - **public List<T> GetDbSetWithIncludes<T>(string[] includes) where T : class**
פונקציה גנרית השולפת נתוני טבלה מסוימת מה DB כולל INCLUDES רצויים המתקבלים כמערך מחרוזות.
כדי לממש את הפונקציה הזו – הוספתי פונקצייה ב DB.Context :

```

public IQueryable<T> IncludeMultiple<T>(IQueryable<T> query, string[] includes)
where T : class
{
    if (includes != null)
    {
        query = includes.Aggregate(query, (current, include) =>
            current.Include(include));
    }
    return query;
}

```
- ```

public class DBConnection
{
 public DBConnection() { }
 public List<T> GetDbSet<T>() where T : class
 {
 using (volunteers_scheduling_DBEntities volunteers_scheduling_DBEntities = new
volunteers_scheduling_DBEntities())
 {
 volunteers_scheduling_DBEntities.Configuration.LazyLoadingEnabled = false;
 volunteers_scheduling_DBEntities.Configuration.LazyLoadingEnabled = false;
 return volunteers_scheduling_DBEntities.GetDbSet<T>().ToList();
 }
 }

 public List<T> GetDbSetWithIncludes<T>(string[] includes) where T : class
 {
 using (volunteers_scheduling_DBEntities volunteers_scheduling_DBEntities = new
volunteers_scheduling_DBEntities())
 {
 volunteers_scheduling_DBEntities.Configuration.LazyLoadingEnabled = false;

 return
volunteers_scheduling_DBEntities.IncludeMultiple<T>(volunteers_scheduling_DBEntities.GetDbSet
<T>(),includes).ToList();
 }
 }
}

```

```

 }
}

public enum ExecuteActions
{
 Insert,
 Update,
 Delete
}

public void Execute<T>(T entity, ExecuteActions exAction) where T : class
{
 using (volunteers_scheduling_DBEntities volunteers_scheduling_DBEntities = new
volunteers_scheduling_DBEntities())
 {
 var model = volunteers_scheduling_DBEntities.Set<T>();
 switch (exAction)
 {
 case ExecuteActions.Insert:
 model.Add(entity);
 break;
 case ExecuteActions.Update:
 model.Attach(entity);
 volunteers_scheduling_DBEntities.Entry(entity).State =
System.Data.Entity.EntityState.Modified;
 break;
 case ExecuteActions.Delete:
 model.Attach(entity);
 volunteers_scheduling_DBEntities.Entry(entity).State =
System.Data.Entity.EntityState.Deleted;
 break;
 default:
 break;
 }
 volunteers_scheduling_DBEntities.SaveChanges();
 }
}
}

```

### 3. שכבת ה MODELS

בשכבה זו ישנן מחלקות כנגד כל מחלקה שנוצרה ע"י ה Entity Framework (כנגד כל טבלה שבמסד הנתונים). הסיבה ליצירת מחלקות אלו היא: החילול האוטומטי – entity framework - יוצר מחלקות כנגד כל טבלה שבמסד הנתונים, אך הוא מוסיף למחלקות אלו פרטים נוספים שמפריעים לריצת הפרויקט. ולכן יצרתי את שכבת הישויות בה יש מחלקות כנגד כל טבלה. במחלקות המקוריות משתמשים לאורך כל הפרויקט וכאשר רוצים לבוא במגע עם צד הלקוח לצורך קבלת או שליחת נתונים- ממירים בין המחלקות.

#### 4. שכבת האלגוריתמים – BL

בשכבה זו מתבצעים האלגוריתמים וקטעי הקוד המשמעותיים בפרויקט.

##### המחלקות:

##### מחלקת VolunteersSchedulingBL:

##### תכונות המחלקה:

##### • **DBConnection dbCon**

מופע ממחלקת DBConnection המאפשר לתקשר עם מסד הנתונים.

##### פונקציות המחלקה:

##### • **public List<T> GetDbSet<T>() where T : class**

פונקציה גנרית המחזירה רשימה המכילה את כל העצמים של הטבלה המבוקשת ע"י זימון הפונקציה הגנרית שב dbCon.

##### • **public void AddToDB<T>(T entity) where T : class**

פונקציה גנרית המקבלת עצם ומוסיפה אותו ל DB ע"י זימון הפונקציה הגנרית שב dbCon. היא מופעלת על שם המחלקה אליה רוצים להוסיף.

##### • **public void DeleteToDB<T>(T entity) where T : class**

פונקציה גנרית המקבלת עצם ומוחקת אותו מה DB ע"י זימון הפונקציה הגנרית שב dbCon. היא מופעלת על שם המחלקה ממנה רוצים למחוק.

##### • **public void UpdateToDB<T>(T entity) where T : class**

פונקציה גנרית המקבלת עצם ומעדכנת אותו ב DB ע"י זימון הפונקציה הגנרית שב dbCon. היא מופעלת על שם המחלקה בה רוצים לעדכן.

```
public class VolunteersSchedulingBL
{
 DBConnection dbCon;
 public VolunteersSchedulingBL()
 {
 dbCon = new DBConnection();
 }
 public enum Result
 {
 IncorrectDetails,
 NotFound,
 Found
 }
 public List<T> GetDbSet<T>() where T : class
 {
 return dbCon.GetDbSet<T>();
 }
 public void AddToDB<T>(T entity) where T : class
 {
 dbCon.Execute<T>(entity, DBConnection.ExecuteActions.Insert);
 }
}
```

```
public void DeleteToDB<T>(T entity) where T : class
{
 dbCon.Execute<T>(entity, DBConnection.ExecuteActions.Delete);
}
public void UpdateToDB<T>(T entity) where T : class
{
 dbCon.Execute<T>(entity, DBConnection.ExecuteActions.Update);
}
}
```

שכבה זו כוללת תיקיה בשם classes המכילה מחלקה עבור כל טבלה במסד הנתונים. מחלקות אלו דואגות להמיר מישויות של entity framework לטיפוס models, ומטפלות בהוספה, עדכון ומחיקה. מחלקות אלו יורשות מהמחלקה הנ"ל – VolunteersSchedulingBL.

### :VolunteerBL:VolunteersSchedulingBL

#### תכונות המחלקה:

- **DBConnection dbCon**  
מופע למחלקת החיבור למסד הנתונים, לצורך תקשורת איתו.
- **List<VolunteerModel> listOfVolunteerss**  
רשימת עצמים מטיפוס המחלקה ב models. מכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

#### פונקציות המחלקה:

- **Public VolunteerBL()**  
קונסטרוקטור המאתחל את הערך של dbCon ואת הרשימה לערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **public List<VolunteerModel> GetAllvolunteers()**  
פונקציה המחזירה את הרשימה listOfVolunteerss המכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **public string InsertVolunteer(VolunteerModel volunteer1)**  
פונקציה המקבלת מתנדב חדש מסוג models ומוסיפה אותו לטבלת מתנדבים במסד הנתונים. במידה והפעולה הצליחה או שהמתנדב קיים כבר במערכת – הפונקציה מחזירה את תעודת הזהות שלו. במקרה של שגיאה הפונקציה מחזירה מחרוזת ריקה.
- **public string UpdateVolunteer(VolunteerModel volunteer1)**  
פונקציה המקבלת מתנדב מסוג models ומעדכנת אותו בטבלת מתנדבים במסד הנתונים. במקרה של הצלחה הפונקציה מחזירה את תעודת הזהות שלו, במקרה של שגיאה הפונקציה מחזירה מחרוזת ריקה.
- **public bool DeleteVolunteer(string volunteerID)**  
פונקציה המקבלת תעודת זהות של מתנדב ומוחקת אותו מטבלת מתנדבים. הפונקציה מחזירה ערך בוליאני המציין האם הפעולה הצליחה או נכשלה.
- **public static volunteer ConvertVolunteerToEF(MODELS.VolunteerModel v)**  
פונקציה סטטית המקבלת מתנדב מסוג models ומחזירה אותו בתבנית של מתנדב מסוג entity framework.
- **public static MODELS.VolunteerModel ConvertVolunteerToModel(volunteer v)**  
פונקציה סטטית המקבלת מתנדב מסוג entity framework ומחזירה אותו בתבנית של models.

- **public static List<MODELS.VolunteerModel> ConvertListToModel(List<volunteer> li)**  
פונקציה המקבלת רשימת מתנדבים מסוג entity framework ומחזירה את הרשימה מסוג models.
- **public string[,] InsertVolunteersFromExcelFile(string path,int orgCode)**  
פונקציה המקבלת נתיב לקובץ אקסל של מתנדבים שהועלה ע"י מנהל ארגון, מזמנת את אלגוריתם הקריאה מאקסל במבנה לא קבוע, קוראת את המתנדבים שמופיעים בקובץ ומכניסה אותם לdb.

### :NeedyBL:VolunteersSchedulingBL

#### תכונות המחלקה:

- **DBConnection dbCon**  
מופע למחלקת החיבור למסד הנתונים, לצורך תקשורת איתו.
- **List<NeedyModel> listOfNeedies**  
רשימת עצמים מטיפוס המחלקה ב models. מכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

#### פונקציות המחלקה:

- **Public NeedyBL()**  
קונסטרוקטור המאתחל את הערך של dbCon ואת הרשימה לערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **public List<NeedyModel> GetAllNeedies()**  
פונקציה המחזירה את הרשימה listOfNeedies המכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **public string InsertNeedy(MODELS.NeedyModel needy1)**  
פונקציה המקבלת מקבל סיוע חדש מסוג models ומוסיפה אותו לטבלת מקבלי סיוע במסד הנתונים. במידה והפעולה הצליחה או שמקבל הסיוע קיים כבר במערכת – הפונקציה מחזירה את תעודת הזהות שלו. במקרה של שגיאה הפונקציה מחזירה מחרוזת ריקה.
- **public string UpdateNeedy(MODELS.NeedyModel needy1)**  
פונקציה המקבלת מקבל סיוע מסוג models ומעדכנת אותו בטבלת מקבלי סיוע במסד הנתונים. במקרה של הצלחה הפונקציה מחזירה את תעודת הזהות שלו, במקרה של שגיאה הפונקציה מחזירה מחרוזת ריקה.
- **public bool DeleteNeedy(string needy\_ID)**  
פונקציה המקבלת תעודת זהות של מקבל סיוע ומוחקת אותו מטבלת מקבלי סיוע. הפונקציה מחזירה ערך בוליאני המציין האם הפעולה הצליחה או נכשלה.
- **public static needy ConvertNeedyToEF(MODELS.NeedyModel n)**  
פונקציה סטטית המקבלת מקבל סיוע מסוג models ומחזירה אותו בתבנית של מקבל סיוע מסוג entity framework.
- **public static MODELS.NeedyModel ConvertNeedyToModel(needy n)**  
פונקציה סטטית המקבלת מקבל סיוע מסוג entity framework ומחזירה אותו בתבנית של models.
- **public static List<MODELS.NeedyModel> ConvertListToModel(List<needy> li)**  
פונקציה המקבלת רשימת מתנדבים מסוג entity framework ומחזירה את הרשימה מסוג models.

• **public string[,] InsertNeediesFromExcelFile(string path,int orgCode)**

פונקציה המקבלת נתיב לקובץ אקסל של מקבלי סיוע שהועלה ע"י מנהל ארגון, מזמנת את אלגוריתם הקריאה מאקסל במבנה לא קבוע, קוראת את מקבלי הסיוע שמופיעים בקובץ ומכניסה אותם לdb.

**ManagerBL:VolunteersSchedulingBL**

**תכונות המחלקה:**

• **DBConnection dbCon**

מופע למחלקת החיבור למסד הנתונים, לצורך תקשורת איתו.

• **List<MODELS.ManagerModel> listOfManagers**

רשימת עצמים מטיפוס המחלקה ב models. מכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

**פונקציות המחלקה:**

• **Public ManagerBL ()**

קונסטרוקטור המאתחל את הערך של dbCon ואת הרשימה לערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

• **public List<MODELS.ManagerModel> GetAllManagers()**

פונקציה המחזירה את הרשימה listOfManagers המכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

• **public string InsertManager(MODELS.ManagerModel manager1)**

פונקציה המקבלת מנהל חדש מסוג models ומוסיפה אותו לטבלת מנהלים במסד הנתונים. במידה והפעולה הצליחה או שמנהל זה קיים כבר במערכת – הפונקציה מחזירה את תעודת הזהות שלו. במקרה של שגיאה הפונקציה מחזירה מחרוזת ריקה.

• **public string UpdateManager(MODELS.ManagerModel manager1)**

פונקציה המקבלת מנהל מסוג models ומעדכנת אותו בטבלת מנהלים במסד הנתונים. במקרה של הצלחה הפונקציה מחזירה את תעודת הזהות שלו, במקרה של שגיאה הפונקציה מחזירה מחרוזת ריקה.

• **public bool DeleteManager(string managerID)**

פונקציה המקבלת תעודת זהות של מנהל ומוחקת אותו מטבלת מקבלי סיוע. הפונקציה מחזירה ערך בוליאני המציין האם הפעולה הצליחה או נכשלה.

• **public static manager ConvertManagerToEF(MODELS.ManagerModel m)**

פונקציה סטטית המקבלת מנהל מסוג models ומחזירה אותו בתבנית של מנהל מסוג entity framework.

• **public static MODELS.ManagerModel ConvertManagerToModel(manager m)**

פונקציה סטטית המקבלת מנהל מסוג entity framework ומחזירה אותו בתבנית של models.

• **public static List<MODELS.ManagerModel> ConvertListToModel(List<manager> li)**

פונקציה המקבלת רשימת מנהלים מסוג entity framework ומחזירה את הרשימה מסוג models.

**OrganizationBL:VolunteersSchedulingBL**

**תכונות המחלקה:**

- **DBConnection dbCon**  
מופע למחלקת החיבור למסד הנתונים, לצורך תקשורת איתו.
- **List<MODELS.OrganizationModel> listOfOrganizations**  
רשימת עצמים מטיפוס המחלקה ב models. מכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

#### פונקציות המחלקה:

- **(public OrganizationBL**  
קונסטרקטור המאתחל את הערך של dbCon ואת הרשימה לערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **(public List<MODELS.OrganizationModel> GetAllOrganizations**  
פונקציה המחזירה את הרשימה listOfNeedies המכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **public int InsertOrganization(MODELS.OrganizationModel Organization1)**  
פונקציה המקבלת ארגון חדש מסוג models ומוסיפה אותו לטבלת ארגונים במסד הנתונים. במידה והפעולה הצליחה או שהארגון קיים כבר במערכת – הפונקציה מחזירה את הקוד שלו. במקרה של שגיאה הפונקציה מחזירה 0.
- **public int UpdateOrganization(MODELS.OrganizationModel Organization1)**  
פונקציה המקבלת ארגון מסוג models ומעדכנת אותו בטבלת ארגונים במסד הנתונים. במקרה של הצלחה הפונקציה מחזירה את קוד הארגון. במקרה של שגיאה הפונקציה מחזירה 0.
- **public bool DeleteOrganization(MODELS.OrganizationModel Organization1)**  
פונקציה המקבלת ארגון ומוחקת אותו מטבלת מקבלי סיוע. הפונקציה מחזירה ערך בוליאני המציין האם הפעולה הצליחה או נכשלה.
- **public static organization ConvertOrganizationToEF(MODELS.OrganizationModel o)**  
פונקציה סטטית המקבלת ארגון מסוג models ומחזירה אותו בתבנית של ארגון מסוג entity framework.
- **public static MODELS.OrganizationModel ConvertOrganizationToModel(organization o)**  
פונקציה סטטית המקבלת ארגון מסוג entity framework ומחזירה אותו בתבנית של models.
- **public static List<MODELS.OrganizationModel> ConvertListToModel(List<organization> li)**  
פונקציה המקבלת רשימת ארגונים מסוג entity framework ומחזירה את הרשימה מסוג models.

#### VolunteeringDetailsBL:VolunteersSchedulingBL:

#### תכונות המחלקה:

- **DBConnection dbCon**  
מופע למחלקת החיבור למסד הנתונים, לצורך תקשורת איתו.
- **List<MODELS.VolunteeringDetailsModel> listOfVolunteeringDetails**  
רשימת עצמים מטיפוס המחלקה ב models. מכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

### פונקציות המחלקה:

- **(public VolunteeringDetailsBL)**  
קונסטרקטור המאתחל את הערך של dbCon ואת הרשימה לערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **(public List<MODELS.VolunteeringDetailsModel> GetAllVolunteeringDetails)**  
פונקציה המחזירה את הרשימה listOfVolunteeringDetails המכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **public int InsertVolunteeringDetails(MODELS.VolunteeringDetailsModel VolunteeringDetails1)**  
פונקציה המקבלת פרטי התנדבות חדשים מסוג models ומוסיפה אותם לטבלת פרטי התנדבות במסד הנתונים. במידה והפעולה הצליחה – הפונקציה מחזירה את הקוד של השורה בטבלה. במקרה של שגיאה הפונקציה מחזירה 0.
- **public int UpdateVolunteeringDetails(MODELS.VolunteeringDetailsModel VolunteeringDetails1)**  
פונקציה המקבלת פרטי התנדבות מסוג models ומעדכנת אותם בטבלת פרטי התנדבות במסד הנתונים. במקרה של הצלחה הפונקציה מחזירה את קוד השורה, במקרה של שגיאה הפונקציה מחזירה 0.
- **public bool DeleteVolunteeringDetails(int VolunteeringDetailsCode)**  
פונקציה המקבלת קוד פרטי התנדבות ומוחקת אותם מטבלת פרטי התנדבות. הפונקציה מחזירה ערך בוליאני המציין האם הפעולה הצליחה או נכשלה.
- **public static volunteering\_details ConvertVolunteeringDetailsToEF(MODELS.VolunteeringDetailsModel v)**  
פונקציה סטטית המקבלת פרטי התנדבות מסוג models ומחזירה אותם בתבנית של פרטי התנדבות מסוג entity framework.
- **public static MODELS.VolunteeringDetailsModel ConvertVolunteeringDetailsToModel(volunteering\_details v)**  
פונקציה סטטית המקבלת פרטי התנדבות מסוג entity framework ומחזירה אותם בתבנית של models.
- **public static List<MODELS.VolunteeringDetailsModel> ConvertListToModel(List<volunteering\_details> li)**  
פונקציה המקבלת רשימת פרטי התנדבות מסוג entity framework ומחזירה את הרשימה מסוג models.

### NeedinessDetailsBL:VolunteersSchedulingBL:

#### תכונות המחלקה:

- **DBConnection dbCon**  
מופע למחלקת החיבור למסד הנתונים, לצורך תקשורת איתו.
- **List<MODELS.NeedinessDetailsModel> listOfNeedinessDetails**  
רשימת עצמים מטיפוס המחלקה ב models. מכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

### פונקציות המחלקה:



- **(public NeedinessDetailsBL**  
קונסטרקטור המאתחל את הערך של dbCon ואת הרשימה לערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **(public List<MODELS. NeedinessDetailsModel> GetAllNeedinessDetails**  
פונקציה המחזירה את הרשימה listOfNeedinessDetails המכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **public int InsertNeedinessDetails(MODELS. NeedinessDetailsModel VolunteeringDetails1)**  
פונקציה המקבלת פרטי הזדקקות חדשים מסוג models ומוסיפה אותם לטבלת פרטי הזדקקות במסד הנתונים. במידה והפעולה הצליחה – הפונקציה מחזירה את הקוד של השורה בטבלה. במקרה של שגיאה הפונקציה מחזירה 0.
- **public int UpdateNeedinessDetails(MODELS. NeedinessDetailsModel VolunteeringDetails1)**  
פונקציה המקבלת פרטי הזדקקות מסוג models ומעדכנת אותם בטבלת פרטי הזדקקות במסד הנתונים. במקרה של הצלחה הפונקציה מחזירה את קוד השורה, במקרה של שגיאה הפונקציה מחזירה 0.
- **public bool DeleteNeedinessDetails(int NeedinessDetailsCode)**  
פונקציה המקבלת קוד פרטי הזדקקות ומוחקת אותם מטבלת פרטי הזדקקות. הפונקציה מחזירה ערך בוליאני המציין האם הפעולה הצליחה או נכשלה.
- **public static neediness\_details ConvertNeedinessDetailsToEF(MODELS. NeedinessDetailsModel v)**  
פונקציה סטטית המקבלת פרטי הזדקקות מסוג models ומחזירה אותם בתבנית של פרטי הזדקקות מסוג entity framework.
- **public static MODELS. NeedinessDetailsModel ConvertNeedinessDetailsToModel(neediness\_details v)**  
פונקציה סטטית המקבלת פרטי הזדקקות מסוג entity framework ומחזירה אותם בתבנית של models.
- **public static List<MODELS. NeedinessDetailsModel> ConvertListToModel(List<neediness\_details> li)**  
פונקציה המקבלת רשימת פרטי הזדקקות מסוג entity framework ומחזירה את הרשימה מסוג models.

### HoursBL:VolunteersSchedulingBL

#### תכונות המחלקה:

- **DBConnection dbCon**  
מופע למחלקת החיבור למסד הנתונים, לצורך תקשורת איתו.
- **List<MODELS.HourModel> listOfHours**  
רשימת עצמים מטיפוס המחלקה ב models. מכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

#### פונקציות המחלקה:

- **Public HourBL ()**

קונסטרקטור המאתחל את הערך של dbCon ואת הרשימה לערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

- **public List<MODELS.HourModel> GetAllManagers()**

פונקציה המחזירה את הרשימה listOfHours המכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

- **public static MODELS.HourModel ConvertHoursToModel(hour h)**

פונקציה סטטית המקבלת שעה מסוג entity framework ומחזירה אותו בתבנית של models.

- **public static List<MODELS.HourModel> ConvertListToModel(List<hour> li)**

פונקציה סטטית המקבלת רשימת שעות מסוג entity framework ומחזירה אותה כmodels.

### TimeSlotBL:VolunteersSchedulingBL

#### תכונות המחלקה:

- **DBConnection dbCon**

מופע למחלקת החיבור למסד הנתונים, לצורך תקשורת איתו.

- **;List<MODELS.TimeSlotModel> listOfTimeSlot**

רשימת עצמים מטיפוס המחלקה ב models. מכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

#### פונקציות המחלקה:

- **Public TimeSlotBL ()**

קונסטרקטור המאתחל את הערך של dbCon ואת הרשימה לערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

- **public List<MODELS.TimeSlotModel> GetAllTimeSlots()**

פונקציה המחזירה את הרשימה listOfTimeSlot המכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

- **public int InsertTimeSlot(MODELS.TimeSlotModel TimeSlot1)**

פונקציה המקבלת משבצת זמן חדשה מסוג models ומוסיפה אותה לטבלת משבצות זמן במסד הנתונים. במידה והפעולה הצליחה - הפונקציה מחזירה את קוד משבצת הזמן. במקרה של שגיאה הפונקציה מחזירה 0.

- **public int UpdateTimeSlot(MODELS.TimeSlotModel TimeSlot1)**

פונקציה המקבלת משבצת זמן מסוג models ומעדכנת אותה בטבלת משבצות זמן במסד הנתונים. במקרה של הצלחה הפונקציה מחזירה את קוד משבצת הזמן, במקרה של שגיאה הפונקציה מחזירה 0.

- **public bool DeleteTimeSlot(int code)**

פונקציה המקבלת קוד של משבצת זמן ומוחקת אותה מטבלת משבצות זמן. הפונקציה מחזירה ערך בוליאני המציין האם הפעולה הצליחה או נכשלה.

- **public static time\_slot ConvertTimeSlotToEF(MODELS.TimeSlotModel t)**

פונקציה סטטית המקבלת משבצת זמן מסוג models ומחזירה אותו בתבנית של משבצות זמן מסוג entity framework.

- **public static MODELS.TimeSlotModel ConvertTimeSlotToModel(time\_slot t)**

פונקציה סטטית המקבלת משבצת זמן מסוג entity framework ומחזירה אותה בתבנית של models.

- **public static List<MODELS.TimeSlotModel> ConvertListToModel(List<time\_slot> li)**  
פונקציה המקבלת רשימת משבצות זמן מסוג entity framework ומחזירה את הרשימה מסוג models.

### NeedyPossibleTimeBL:VolunteersSchedulingBL

תכונות המחלקה:

- **DBConnection dbCon**  
מופע למחלקת החיבור למסד הנתונים, לצורך תקשורת איתו.
- **List<MODELS.NeedyPossibleTimeModel> listOfNeedyPossibleTime**  
רשימת עצמים מטיפוס המחלקה ב models. מכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

פונקציות המחלקה:

- **(public NeedyPossibleTimeBL**  
קונסטרוקטור המאתחל את הערך של dbCon ואת הרשימה לערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **(public List<NeedyPossibleTimeModel> GetAllNeedyPossibleTime**  
פונקציה המחזירה את הרשימה listOfNeedyPossibleTime המכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **public int InsertNeedyPossibleTime(NeedyPossibleTimeModel needyPossibleTime1)**  
פונקציה המקבלת זמן אפשרי חדש למקבל סיוע מסוג models ומוסיפה אותו לטבלת זמן אפשרי למקבלי סיוע במסד הנתונים. במידה והפעולה הצליחה - הפונקציה מחזירה את קוד הזמן האפשרי. במקרה של שגיאה הפונקציה מחזירה 0.
- **public int UpdateNeedyPossibleTime(NeedyPossibleTimeModel needyPossibleTime1)**  
פונקציה המקבלת זמן אפשרי למקבל סיוע מסוג models ומעדכנת אותו בטבלת זמן אפשרי למקבלי סיוע במסד הנתונים. במקרה של הצלחה הפונקציה מחזירה את קוד הזמן האפשרי, במקרה של שגיאה הפונקציה מחזירה 0.
- **public bool DeleteNeedyPossibleTime(NeedyPossibleTimeModel needyPossibleTime1)**  
פונקציה המקבלת זמן אפשרי למקבל סיוע ומוחקת אותו מטבלת זמן אפשרי למקבלי סיוע. הפונקציה מחזירה ערך בוליאני המציין האם הפעולה הצליחה או נכשלה.
- **public static needy\_possible\_time**  
**ConvertNeedyPossibleTimeToEF(MODELS.NeedyPossibleTimeModel n)**  
פונקציה סטטית המקבלת זמן אפשרי למקבל סיוע מסוג models ומחזירה אותו בתבנית של זמן אפשרי למקבל סיוע מסוג entity framework.
- **public static MODELS.NeedyPossibleTimeModel**  
**ConvertNeedyPossibleTimeToModel(needy\_possible\_time n)**  
פונקציה סטטית המקבלת זמן אפשרי למקבל סיוע מסוג entity framework ומחזירה אותו בתבנית של models.
- **public static List<MODELS.NeedyPossibleTimeModel>**  
**ConvertListToModel(List<needy\_possible\_time> li)**

פונקציה המקבלת רשימת זמנים אפשריים למקבלי סיוע מסוג entity framework ומחזירה את הרשימה מסוג models.

### VolunteerPossibleHoursBL:VolunteersSchedulingBL

#### תכונות המחלקה:

- **DBConnection dbCon**  
מופע למחלקת החיבור למסד הנתונים, לצורך תקשורת איתו.
- **List<MODELS.VolunteerPossibleTimeModel> listOfVolunteerPossibleTime**  
רשימת עצמים מטיפוס המחלקה ב models. מכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

#### פונקציות המחלקה:

- **(public VolunteerPossibleHoursBL**  
קונסטרקטור המאתחל את הערך של dbCon ואת הרשימה לערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **(public List<MODELS.VolunteerPossibleTimeModel> GetAllVolunteerPossibleTime**  
פונקציה המחזירה את הרשימה listOfVolunteerPossibleTime המכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **public int InsertVolunteerPossibleTime(MODELS.VolunteerPossibleTimeModel volunteerPossibleTime1)**  
פונקציה המקבלת זמן אפשרי חדש למתנדב מסוג models ומוסיפה אותו לטבלת זמן אפשרי למתנדב במסד הנתונים. במידה והפעולה הצליחה - הפונקציה מחזירה את קוד הזמן האפשרי. במקרה של שגיאה הפונקציה מחזירה 0.
- **public int UpdateVolunteerPossibleTime(MODELS.VolunteerPossibleTimeModel volunteerPossibleTime1)**  
פונקציה המקבלת זמן אפשרי למתנדב מסוג models ומעדכנת אותו בטבלת זמן אפשרי למתנדב במסד הנתונים. במקרה של הצלחה הפונקציה מחזירה את קוד הזמן האפשרי, במקרה של שגיאה הפונקציה מחזירה 0.
- **public bool DeleteVolunteerPossibleTime(int code)**  
פונקציה המקבלת זמן אפשרי למתנדב ומוחקת אותו מטבלת זמן אפשרי למתנדב. הפונקציה מחזירה ערך בוליאני המציין האם הפעולה הצליחה או נכשלה.
- **public static volunteer\_possible\_time ConvertVolunteerPossibleTimeToEF(MODELS.VolunteerPossibleTimeModel v)**  
פונקציה סטטית המקבלת זמן אפשרי למתנדב מסוג models ומחזירה אותו בתבנית של זמן אפשרי למתנדב מסוג entity framework.
- **public static MODELS.VolunteerPossibleTimeModel ConvertVolunteeringDetailsToModel(volunteer\_possible\_time v)**  
פונקציה סטטית המקבלת זמן אפשרי למתנדב מסוג entity framework ומחזירה אותו בתבנית של models.

• **public static List<MODELS.VolunteerPossibleTimeModel>  
ConvertListToModel(List<volunteer\_possible\_time> li)**

פונקציה המקבלת רשימת זמנים אפשריים למתנדב מסוג entity framework ומחזירה את הרשימה מסוג .models

### :CalendarEvents

בצד הלקוח, מוצגת למשתמש המערכת הרלוונטית עבורו. הצורה בה מוצגת לו המערכת כוללת נתונים מרוב הטבלאות, כך שהיה נוח ויעיל יותר ליצור מחלקות חדשות של אירועי לוח שנה למתנדב, למקבל סיוע ולמנהל ארגון, המכילות את כל הפרטים הרלוונטיים באופן ישיר, וכך לארגן את המערכת בצד השרת ולשלוח את הלוח כשהוא מושלם לאנגולר. לצורך כך יצרתי מחלקה אבסטרקטית של אירוע בלוח שנה עם המאפיינים הדרושים: משבצת הזמן הרלוונטית, קוד פרטי התנדבות, קוד פרטי הזדקקות, קוד מערכת, תאריך התחלה, תאריך סיום, וכו'. המחלקה מכילה קונסטרקטור לאיתחול המשתנים.

```
Public abstract class CalendarEvent
{
 public TimeSlotModel 36imeslot { get; set; }
 public VolunteeringDetailsModel volunteeringDetails { get; set; }
 public NeedinessDetailsModel needinessDetails { get; set; }
 public int scheduleCode { get; set; }

 public DateTime start { get; set; }
 public DateTime end { get; set; }
 public string title { get; set; }

 protected CalendarEvent(TimeSlotModel 36imeslot, VolunteeringDetailsModel
volunteeringDetailsModel, NeedinessDetailsModel needinessDetailsModel, int scheduleCode,
DateTime start, DateTime end, string title)
 {
 this.timeSlot = 36imeslot;
 this.volunteeringDetails = volunteeringDetailsModel;
 this.needinessDetails = needinessDetailsModel;
 this.scheduleCode = scheduleCode;
 this.start = start;
 this.end = end;
 this.title = title;
 }
}
```

ממחלקה זו יורשות 3 מחלקות חדשות: אירועי לו"ז למתנדב, אירועי לו"ז למקבל סיוע, ואירועי לו"ז למנהל ארגון. בכל אחת מהמחלקות, מלבד התכונות שעוברות בירושה התווספה תכונות נוספות: בלו"ז של מנהל – מתנדב ומקבל סיוע הרלוונטי להתנדבות זו בלו"ז של מתנדב – מקבל הסיוע הרלוונטי להתנדבות זו בלו"ז של מקבל סיוע – המתנדב הרלוונטי להתנדבות זו

## ScheduleBL:VolunteersSchedulingBL

### תכונות המחלקה:

- **DBConnection dbCon**  
מופע למחלקת החיבור למסד הנתונים, לצורך תקשורת איתו.
- **List<MODELS.ScheduleModel> listOfSchedule**  
רשימת עצמים מטיפוס המחלקה ב models. מכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.

### פונקציות המחלקה:

- **Public ScheduleBL ()**  
קונסטרקטור המאתחל את הערך של dbCon ואת הרשימה לערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **(public List<MODELS.ScheduleModel> GetAllSchedule**  
פונקציה המחזירה את הרשימה listOfSchedule המכילה את הערכים הנוכחיים בטבלה המתאימה במסד הנתונים.
- **public int InsertSchedule(MODELS.ScheduleModel Schedule1)**  
פונקציה המקבלת מערכת חדשה מסוג models ומוסיפה אותו לטבלת מערכת במסד הנתונים. במידה והפעולה הצליחה - הפונקציה מחזירה את הקוד שלה. במקרה של שגיאה הפונקציה מחזירה 0.
- **public int UpdateSchedule(MODELS.ScheduleModel Schedule1)**  
פונקציה המקבלת מערכת מסוג models ומעדכנת אותה בטבלת מערכת במסד הנתונים. במקרה של הצלחה הפונקציה מחזירה את הקוד, במקרה של שגיאה הפונקציה מחזירה 0.
- **public bool DeleteSchedule(int scheduleCode)**  
פונקציה המקבלת קוד מערכת ומוחקת אותה מטבלת מערכת. הפונקציה מחזירה ערך בוליאני המציין האם הפעולה הצליחה או נכשלה.
- **public static schedule ConvertScheduleToEF(MODELS.ScheduleModel s)**  
פונקציה סטטית המקבלת מערכת מסוג models ומחזירה אותה בתבנית של מערכת מסוג entity framework.
- **public static MODELS.ScheduleModel ConvertScheduleToModel(schedule s)**  
פונקציה סטטית המקבלת מערכת מסוג entity framework ומחזירה אותה בתבנית של models.
- **public static List<MODELS.ScheduleModel> ConvertListToModel(List<schedule> li)**  
פונקציה המקבלת רשימת מערכות מסוג entity framework ומחזירה את הרשימה מסוג models.
- **public List<ManagerCalendarEvent> GetScheduleForMnager(string managerID)**  
פונקציה המקבלת תעודת זהות של מנהל ומחזירה את המערכת הרלוונטית עבורו בפורמט של ManagerCalendarEvent.
- **public List<NeedyCalendarEvent> GetScheduleForNeedy(string needyID)**  
פונקציה המקבלת תעודת זהות של מקבל סיוע ומחזירה את המערכת הרלוונטית עבורו בפורמט של NeedyCalendarEvent.
- **public List<VolunteerCalendarEvent> GetScheduleForVolunteer(string volunteerID)**  
פונקציה המקבלת תעודת זהות של מתנדב ומחזירה את המערכת הרלוונטית עבורו בפורמט של VolunteerCalendarEvent.

## אלגוריתמים מרכזיים:

### חישוב מרחק בין כתובות המתקבלות כטקסט:

באלגוריתם זה השתמשתי באלגוריתם השיבוץ, לצורך התאמה מקסימלית בין מתנדב למקבל הסיוע, שבין היתר דורש מרחק מינימלי ביניהם.

יצרתי מחלקות נחוצות לצורך עבודה עם google maps ועם matrix api כדי להמיר את קבצי ה json שחוזרים מהם לצורך עבודה עם הנתונים:

```
public class DistanceBase
{
 public string[] destination_addresses { get; set; }
 public string[] origin_addresses { get; set; }
 public Row[] rows { get; set; }
 public string status { get; set; }
}

public class Row
{
 public Element[] elements { get; set; }
}

public class Element
{
 public Distance distance { get; set; }
 public Duration duration { get; set; }
 public string status { get; set; }
}

public class Distance
{
 public string text { get; set; }
 public int value { get; set; }
}

public class Duration
{
 public string text { get; set; }
 public int value { get; set; }
}

public class RootLocationBase
{
 public Result[] results { get; set; }
}

public class Result
{
 public string place_id { get; set; }
}
```

### מחלקת googleMaps:

#### פונקציות המחלקה:

• static string BuildUrlForLocationId(string address)

פונקציה המקבלת כתובת במילים, ומחזירה קישור ל api הנדרש משורשר בכתובת שקיבלה, הקישור מגיע לדף במבנה json המתאר את המיקום שהתקבל - את הקואורדינטות שלו, id שלו ועוד.

• **static string BuildUrlForDistance(string path1, string path2)**

פונקציה המקבלת id של 2 כתובות ומחזירה קישור ל api הנדרש כדי לקבל את המרחק בין הכתובות.

• **public static async Task<double> GetDistanceInMinutes(string origin, string destination)**

פונקציה אסינכרונית המקבלת כתובות מקור ויעד במילים, ומחזירה את משך הזמן שלוקח לעבור ביניהם ע"י שימוש בפונקציות הנ"ל.

```
public class GoogleMaps
{
 public static async Task<double> GetDistanceInMinutes(string origin, string
destination)
 {
 string[] locationUrls = { BuildUrlForLocationId(origin),
BuildUrlForLocationId(destination) },

 idLocation = new string[2];
 HttpClient http = new HttpClient();
 for (int i = 0; i < idLocation.Length; i++)
 {
 var responseId = Task.Run(() => http.GetAsync(locationUrls[i]));
 if (responseId.Result != null)
 {
 var result = await responseId.Result.Content.ReadAsStringAsync();
 RootLocationBase root =
JsonConvert.DeserializeObject<RootLocationBase>(result);
 idLocation[i] = root.results[0].place_id;
 }
 }
 var responseDistance = Task.Run(() =>
http.GetAsync(BuildUrlForDistance(idLocation[0], idLocation[1])));
 if (responseDistance.Result != null)
 {
 var result = await responseDistance.Result.Content.ReadAsStringAsync();
 DistanceBase root = JsonConvert.DeserializeObject<DistanceBase>(result);
 double f = root.rows[0].elements[0].duration.value;
 double minutes = f / 60;
 return minutes;
 }
 return 0.0;
 }

 // בניית כתובת עבור הגוגל מפת
 static string BuildUrlForLocationId(string address)
 {
 return
"https://maps.googleapis.com/maps/api/place/textsearch/json?key=AIzaSyAyEYq8aMTFp3eNcRxWj3z4r
FPLx7BamYo&query=" + address;
 }

 // בניית פונקציית המרחק
 static string BuildUrlForDistance(string path1, string path2)
 {
 string url =
"https://maps.googleapis.com/maps/api/distancematrix/json?destinations=place_id:";
 url += path1 + "&origins=place_id:" + path2 +
"&key=AIzaSyAyEYq8aMTFp3eNcRxWj3z4rFPLx7BamYo";
 return url;
 }
}
```



```
}
}
```

## אלגוריתם קריאה מקובץ excel במבנה לא קבוע

כדי להקל על מנהלי הארגונים, שברובם ניהלו את הרישומים עד היום בטבלאות אקסל - יצרתי את האפשרות של הוספת מתנדבים ומקבלי סיוע לארגון ע"י העלאת קובץ אקסל המכיל את פרטיהם. ע"מ שיהיה למשתמש נח וקל להשתמש באתר וכמה שפחות דברים הקבועים לו מראש - אפשרתי לו להעלות לאתר כל קובץ אקסל שברשותו המכיל את הפרטים הנצרכים של המתנדבים / מקבלי הסיוע, בטבלה שהוא מעלה יכולים להיות שדות נוספים מלבד השדות ההכרחיים ושמות הכותרות ניתנות לו לבחירה, בניית אלגוריתם אשר יודע לזהות את כותרת השדה הרצוי ולשמור את נתוניו בטבלה המתאימה ב DB וכן לא להתייחס לשדות שאין לי צורך בהם. לצורך כך – יצרתי קבצי txt בתיקיית ה BL הבנויים בצורה של קובץ json ומכילים את כל האפשרויות שעלו בדעתי לכותרות עבור הנתונים שאמורים להיקרא מהקובץ.

דוגמה לקובץ טקסט של פרטי מתנדב:

```
}
"volunteer_ID":["תעודת זהות","תז","ת.ז."],
"volunteer_full_name":["שם","שם מלא","שם נזקק","שם חבר","שמות"],
"volunteer_address":["כתובת"],
"volunteer_phone":["טלפון","פלאפון","פלא","טל","מספר פלאפון","מספר טלפון","מס' פלאפון","מס' טלפון"],
"volunteer_email":["כתובת מייל","מייל","אימייל","דוא"ל","כתובת דוא"ל"],
"volunteer_birth_date":["תאריך לידה","יום הולדת"]
{
```

אלגוריתם הקריאה מאקסל מזומן עבור קריאת קבצים של מתנדבים / מקבלי סיוע שמעלה מנהל ארגון. לכן יצרתי 4 קבצי טקסט: פרטי מתנדבים, פרטי מקבלי סיוע – חובה שימצאו בקובץ, פרטי התנדבות ופרטי הזדקקות (הכוונה למספר שעות בשבוע) – פרטים אלו לא חייבים להימצא בקובץ, במידה ולא נמצאו יכנסו ערכי ברירת מחדל.

אופן פעילות האלגוריתם:

האלגוריתם מקבל מצד הלקוח נתיב לקובץ שהועלה, ובדוק אם הנתיב חוקי, במידה וכן: מזהה את כותרות הטבלה ושומר את הנתונים. ואם חסרות עמודות נחוצות בטבלה או שהכותרות נכתבו בצורה שגויה תוצג הודעה למשתמש על כך.

### מחלקת ReadExcel:

תכונות המחלקה:

- public Excel.Application xlApp
- public Excel.Workbook xlWorkbook

- **public Excel.Worksheet xlWorksheet**
- **public Dictionary<string, List<string>> words**  
מילון המשמש כעזר בקריאת כותרות העמודות שהן חובה. המפתח הוא שם השדה כפי שמופיע ב DB והערך הוא רשימה המכילה את כל הכותרות האפשריות לשדה זה, כפי שנקראים מקובץ הטקסט.
- **public Dictionary<string, List<string>> details**  
מילון המשמש כעזר בקריאת כותרות העמודות שאינן חובה. המפתח הוא שם השדה כפי שמופיע ב DB והערך הוא רשימה המכילה את כל הכותרות האפשריות לשדה זה, כפי שנקראים מקובץ הטקסט.
- **public Dictionary<string, int> DictionaryColumns**  
מילון המכיל שם שדה ואת האינדקס שלו בטבלה שבקובץ האקסל כפי שנמצא לאחר הפעלת אלגוריתם הקריאה מהקובץ.

#### פונקציות המחלקה:

- **public void open(string path)**  
הפונקציה מקבלת נתיב לקובץ האקסל שהועלה ופותחת את הקובץ.
- **public void close()**  
הפונקציה סוגרת את קובץ האקסל
- **public void FillWordsForVolunteer()**  
הפונקציה ממלאה את מילוני שמות העמודות עבור קריאה מקובץ של מתנדבים – הן העמודות שהן חובה והן העמודות שאינן חובה.
- **public void FillWordsForNeedy()**  
הפונקציה ממלאה את מילוני שמות העמודות עבור קריאה מקובץ של מקבלי סיוע – הן העמודות שהן חובה והן העמודות שאינן חובה.
- **public bool FindColumnsOfVolunteer()**  
פונקציה הממלאה את מילון העמודות שנמצאו בקובץ הנוכחי – עבור מקבלי סיוע. לכל שדה בטבלה ב DB נשמר האינדקס שלו בקובץ שהועלה.  
הפונקציה מחזירה תשובה לשאלה האם הקובץ מכיל את כל העמודות שהן חובה.
- **public bool FindColumnsOfNeedy()**  
פונקציה הממלאה את מילון העמודות שנמצאו בקובץ הנוכחי – עבור מקבלי סיוע. לכל שדה בטבלה ב DB נשמר האינדקס שלו בקובץ שהועלה.  
הפונקציה מחזירה תשובה לשאלה האם הקובץ מכיל את כל העמודות שהן חובה.

```
public class ReadExcel
{
 public int MyProperty { get; set; }
 public Excel.Application xlApp { get; set; }
 public Excel.Workbook xlWorkBook { get; set; }
 public Excel.Worksheet xlWorkSheet { get; set; }
 public Dictionary<string, List<string>> words { get; set; }
 public Dictionary<string, List<string>> details { get; set; }
 public Dictionary<string, int> DictionaryColumns { get; set; }
 public Dictionary<string, string> valuesString { get; set; }

 // פתיחת קובץ האקסל
 public void open(string path)
```

```
{
 Microsoft.Office.Interop.Excel.Application excel = new
Microsoft.Office.Interop.Excel.Application();

 xlApp = new Excel.Application();
 xlWorkbook = xlApp.Workbooks.Open(path, 0, true, 5, "", "", true,
Excel.XlPlatform.xlWindows, "\t", false, false, 0, true, 1, 0);
 xlWorkSheet = (Excel.Worksheet)xlWorkbook.Worksheets.getItem(1);

}

//גירת קובץ האקסל
public void close()
{
 xlWorkbook.Close(true, null, null);
 xlApp.Quit();
 Marshal.ReleaseComObject(xlWorkSheet);
 Marshal.ReleaseComObject(xlWorkbook);
 Marshal.ReleaseComObject(xlApp);
}

public void FillWordsForVolunteer()
{
 using (StreamReader sr = File.OpenText(@"D:\תומס\טז אדר
א\VolunteersScheduling\BL\Volunteers.txt"))
 {
 words = JsonConvert.DeserializeObject<Dictionary<string,
List<string>>>(sr.ReadToEnd());
 }
 using (StreamReader sr = File.OpenText(@"D:\תומס\טז אדר
א\VolunteersScheduling\BL\VolunteeringDetails.txt"))
 {
 details=JsonConvert.DeserializeObject<Dictionary<string,
List<string>>>(sr.ReadToEnd());
 }
}

public void FillWordsForNeedy()
{
 using (StreamReader sr = File.OpenText(@"D:\תומס\טז אדר
א\VolunteersScheduling\BL\Needies.txt"))
 {
 words = JsonConvert.DeserializeObject<Dictionary<string,
List<string>>>(sr.ReadToEnd());
 }
 using (StreamReader sr = File.OpenText(@"D:\תומס\טז אדר
א\VolunteersScheduling\BL\NeedinessDetails.txt"))
 {
 details = JsonConvert.DeserializeObject<Dictionary<string,
List<string>>>(sr.ReadToEnd());
 }
}

//מילוי מילון שמות העמודות
//הפונקציה מוצאת עבור כל מפתח במילון המבטא שדה בטבלה במסד הנתונים מהי מספר העמודה המתאימה לה
//הפונקציה מחזירה תשובה לשאלה האם לכל השדות בטבלה זו במסד הנתונים יש עמודה מתאימה בטבלה
public bool FindColumnsOfNeedy()
{
 FillWordsForNeedy();
}
```

```

DictionaryColumns = new Dictionary<string, int>();
for (int cCnt = 1; cCnt <= xlWorkSheet.UsedRange.Columns.Count; cCnt++)
{
 string st1 = (string)(xlWorkSheet.UsedRange.Cells[1/*שורה*/, cCnt/*עמודה*/] as
Excel.Range).Value2;
 foreach (var item in words)
 {
 if (item.Value.Contains(st1))
 {
 DictionaryColumns[item.Key] = cCnt;
 break;
 }
 }
 foreach (var item in details)
 {
 if (item.Value.Contains(st1))
 {
 DictionaryColumns[item.Key] = cCnt;
 break;
 }
 }
}
if (DictionaryColumns.Count < words.Count)
{
 return false;
}
else
{
 return true;
}
}

public bool FindColumnsOfVolunteer()
{
 FillWordsForVolunteer();
 DictionaryColumns = new Dictionary<string, int>();
 for (int cCnt = 1; cCnt <= xlWorkSheet.UsedRange.Columns.Count; cCnt++)
 {
 string st1 = (string)(xlWorkSheet.UsedRange.Cells[1/*שורה*/, cCnt/*עמודה*/] as
Excel.Range).Value2;
 foreach (var item in words)
 {
 if (item.Value.Contains(st1))
 {
 DictionaryColumns[item.Key] = cCnt;
 break;
 }
 }
 foreach (var item in details)
 {
 if (item.Value.Contains(st1))
 {
 DictionaryColumns[item.Key] = cCnt;
 break;
 }
 }
 }
 if (DictionaryColumns.Count < words.Count)
 {
 return false;
 }
 else

```

```

 {
 return true;
 }
}

```

דוגמה לזימון האלגוריתם – עבור הוספת מתנדבים ע"י מנהל ארגון:  
פונקציה זו מוסיפה מתנדבים חדשים מקובץ אקסל, משייכת אותם לארגון הנדרש, ומחזירה מטריצה  
המתארת את הסיסמאות שהוגרלו עבור על אחד מהמתנדבים.  
המטריצה מומרת באנגולר לקובץ אקסל חדש הזמין להורדה למנהל זה.

```
Random random = new Random();
```

```

public string[,] InsertVolunteersFromExcelFile(string path, int orgCode)
{
 Dictionary<string, List<string>> words;
 ReadExcel excel = new ReadExcel();
 int mone = 0;
 excel.open(path);
 VolunteerModel newVolunteer;
 VolunteeringDetailsModel volunteeringDetailsModel;
 VolunteeringDetailsBL volunteeringDetailsBL = new VolunteeringDetailsBL();
 int randomPassword;
 string[,] passwords = new string[0, 2];
 List<string[]> list = new List<string[]>();

 if (excel.FindColumnsOfVolunteer())
 {
 for (int rCnt = 2; rCnt <= excel.xlWorkSheet.UsedRange.Rows.Count; rCnt++)
 {
 randomPassword = random.Next(10000, 99999);
 newVolunteer = new VolunteerModel();
 newVolunteer.volunteer_ID =
 (excel.xlWorkSheet.UsedRange.Cells[rCnt/*שורה*/, excel.DictionaryColumns["volunteer_ID"]] as
 Microsoft.Office.Interop.Excel.Range).Value2.ToString();
 newVolunteer.volunteer_full_name =
 (excel.xlWorkSheet.UsedRange.Cells[rCnt/*שורה*/,
 excel.DictionaryColumns["volunteer_full_name"]] as
 Microsoft.Office.Interop.Excel.Range).Value2.ToString();
 newVolunteer.volunteer_address =
 (excel.xlWorkSheet.UsedRange.Cells[rCnt/*שורה*/,
 excel.DictionaryColumns["volunteer_address"]] as
 Microsoft.Office.Interop.Excel.Range).Value2.ToString();
 newVolunteer.volunteer_email =
 (excel.xlWorkSheet.UsedRange.Cells[rCnt/*שורה*/, excel.DictionaryColumns["volunteer_email"]]
 as Microsoft.Office.Interop.Excel.Range).Value2.ToString();
 newVolunteer.volunteer_phone =
 (excel.xlWorkSheet.UsedRange.Cells[rCnt/*שורה*/, excel.DictionaryColumns["volunteer_phone"]]
 as Microsoft.Office.Interop.Excel.Range).Value2.ToString();
 newVolunteer.volunteer_birth_date = new DateTime();

 // הגרלת סיסמה אקראית למתנדב
 randomPassword = random.Next(10000, 99999);

 newVolunteer.volunteer_password = randomPassword.ToString();

 if (this.InsertVolunteer(newVolunteer) != "") mone++;
 }
 }
}

```

```
// הוספת חוליה לרשימה המסכמת את המתנדבים שהוכנסו והסיסמאות שהוגרלו עבורם
string[] row = new string[2];
row[0] = newVolunteer.volunteer_ID;
row[1] = newVolunteer.volunteer_password;
list.Add(row);

volunteeringDetailsModel = new VolunteeringDetailsModel();
volunteeringDetailsModel.volunteer_ID = newVolunteer.volunteer_ID;
volunteeringDetailsModel.org_code = orgCode;

// בדיקה האם הקובץ מכיל את הנתונים שאינם חובה, ואם לא – הכנסת ערכי ברירת מחדל לשדות אלו
if (excel.DictionaryColumns.ContainsKey("monthly_hours"))
{
 volunteeringDetailsModel.weekly_hours =
Convert.ToDouble((excel.xlWorkSheet.UsedRange.Cells[rCnt/*שורה*/,
excel.DictionaryColumns["monthly_hours"]] as
Microsoft.Office.Interop.Excel.Range).Value2.ToString());
}
else
{
 volunteeringDetailsModel.weekly_hours = 0;
}

volunteeringDetailsBL.InsertVolunteeringDetails(volunteeringDetailsModel);
}
excel.close();
}

// המרת הרשימה של התעודות זהות והסיסמאות המוגרלות – למטריצה שתומר באנגולר לקובץ אקסל זמין להורדה למנהל הארגון
string[,] passwordsMatrix = new string[mone, 2];
for (int i = 0; i < mone; i++)
{
 passwordsMatrix[i, 0] = list[i][0].ToString();
 passwordsMatrix[i, 1] = list[i][1].ToString();
}
return passwordsMatrix;
}
```

## אלגוריתם שיבוץ מערכת בארגון:

השיבוץ נעשה בעזרת אלגוריתם גנטי כאשר הוא מופעל על רשימת משבצות זמן של מערכת בארגון מסוים, והוא משבץ בהן מתנדבים ומקבלי סיוע מתאימים, ומחזיר שיבוץ אפשרי אופטימלי, אם כי לא תמיד השיבוץ הוא האופטימלי ביותר אלא בקירוב.

העיקרון של אלגוריתם גנטי הוא כמו פעילות של השבחה גנטית שבה נוטלים מכל יצור את הגן הטוב יותר ומעבירים אותו לדור הבא וכך משתבח היצור מדור לדור.

מהלך האלגוריתם כולל:

שילוב אלמנטים של פתרונות אפשריים לבעיה (על מנת לקבל פתרון מוצלח יותר), והפעלת הליכים של ברירה מלאכותית כדי לבחור את המועמדים שיעברו לשלבים הבאים.

לכן, אם ניקח אוכלוסייה של פתרונות ונבחר מתוכם רק את המתאימים ביותר לפתרון הבעיה, נמזג אותם אחד עם השני ונוסיף מעט שינויים - נקבל דור חדש של פתרונות הקרוב צעד נוסף לפתרון הבעיה הנתונה. נחזור על התהליך מספר רב של פעמים (דורות) ולבסוף נגיע לפתרון הטוב ביותר. כדי לעבוד בצורה מסודרת ויעילה יותר, יצרתי מחלקה חדשה - TimeSlotProperties, המכילה סיכום של כל המאפיינים הנדרשים למשבצת זמן של מערכת בארגון. כך במהלך האלגוריתם עבדתי עם אובייקטים ממחלקה זו. המחלקה מכילה 4 תכונות - מתנדב, מקבל סיוע ומשבצת זמן - מסוג entity framework, וקוד ארגון - int.

```
public class TimeSlotProperties
{
 public volunteer volunteer { get; set; }
 public needy needy { get; set; }
 public int orgCode { get; set; }
 public time_slot time { get; set; }
}
```

### האלגוריתם המרכזי:

שלב זה הינו לאחר שלב קליטת הנתונים, כאשר נתונים בידינו רשימות המתנדבים, מקבלי הסיוע, האילוצים עבור כל אחד - הימים המועדפים בשבוע, שעות רצויות. עלינו לשבץ את המתנדבים אצל מקבלי הסיוע במערכת בצורה חכמה ויעילה כך שהמערכת תתאים לאילוצי המתנדבים ומקבלי הסיוע, בלי קונפליקטים פנימיים או בין מערכות של ארגונים אחרים. לשם כך נשתמש באלגוריתם גנטי. כמו שהוסבר לעיל - באלגוריתם גנטי אנו יוצרים דורות של פתרונות כך שכל דור חדש נוצר ע"י שילוב של פתרונות מהדור הקודם. בכל דור נוצרים מספר שיבוצים, פתרונות אפשריים. כל פתרון מיוצג ע"י TimeTableChromosome המכיל value - רשימה של איברים מסוג TimeSlotProperties - המייצג שיבוץ של מתנדבים למקבלי סיוע בזמנים מסוימים בארגון מסוים. הדור הראשון של הפתרונות ייוצר באופן רנדומלי: לכל מקבל סיוע בארגון זה יוגרל קוד משבצת זמן מהזמנים האפשריים לו, כל עוד לא עבר את מכסת השעות השבועיות שהוא זכאי להן בארגון זה. לאחר שהוגרלה משבצת זמן מסוימת - יוגרל מתנדב מהמתנדבים היכולים להתנדב בשעה זו. מהדור השני ואילך, ייוצר הדור ע"י בחירת 10% מלוחות משמרות עם הציונים הגבוהים ביותר (הקרובים ביותר לשיבוץ האופטימלי) מהדור הקודם והעברתם לדור החדש בשלמותם. את ה- 50% האחרונים נזרוק ואילו מה- 40% הנותרים נבצע מיזוג בין 2 לוחות ע"י לקיחת הגנים באופן רנדומלי משני לוחות, כלומר שילוב של השיבוץ של שני ההורים לאותו לוח זמנים - ע"י פונקציית Crossover שתוסבר בהמשך. כך יוצרים דורות נוספים כשכל דור מתקרב יותר לפתרון עד שמגיעים לדור שבו יש מערכת עם ציון הקרוב ביותר ל 0 - שיבוץ הקרוב ביותר מושלם.

## הסבר מורחב על פונקציות נבחרות:

### Evaluate:

פונקציה המחשבת ציון לכל כרומוזום – שיבוץ.

הפונקציה תעבור על כל קוביות השיבוץ של לוח הזמנים ותחשב ציון עבורו בהתאם לנתונים הבאים:

- לכל משבצת תיצור רשימה של המשבצות ששובצו באותו יום ושעה וע"י אותו מתנדב / אותו מקבל סיוע, ותוריד נקודה על כל עובד ששובץ באותו הזמן.
- לכל משבצת תחשב את משך זמן ההליכה מהמתנדב למקבל הסיוע, ותוריד נקודות עבור זמן גדול מדי.
- לכל מתנדב תבדוק האם יש לו יותר מ 3 התנדבויות ליום. עבור כל התנדבות שמעבר ל 3 תוריד נקודה
- לכל מתנדב תבדוק את הפרשים בין ההתנדבויות שיש לו באותו יום, ועבור התנדבויות צפופות מדי תוריד נקודה.
- לכל מתנדב תבדוק שמס' שעות ששובצו לו תואם למספר השעות שהוא מוכן להתנדב בארגון זה.

הפונקציה מחזירה את משתנה הציון אבסולוטי בחזקת -1. ציון גבוה יותר מראה על שיבוץ טוב יותר.

### Mutate:

פונקציה היוצרת שינוי אקראי בכרומוזום.

הפונקציה מקבלת לוח זמנים ומגרילה מספר בין 0 ל 2 (כולל) המציין את סוג השינוי שיש לבצע: מחיקת משבצת, הוספת משבצת או עדכון משבצת, ומבצעת את השינוי.

### Crossover:

פונקציה זו מופעלת על לוח זמנים, מקבלת לוח זמנים נוסף וממזגת אותם.

הפונקציה מגרילה מיקום אקראי בלוח הזמנים שעליו היא מופעלת, וממקום זה והלאה משנה את משבצות הזמן לאלו במיקום המקביל בלוח הזמנים שהתקבל.

### Generate:

פונקציה זו מגרילה את השיבוץ הראשוני.

הפונקציה יוצרת רשימה של כל מקבלי הסיוע בארגון זה, ועוברת על הרשימה בלולאה.

עבור כל מקבל סיוע יוצרת רשימה של כל המשבצות זמן שהוא יכול בהם, מערבת את סדר הרשימה ועוברת

עליה בלולאה, כל עוד הרשימה לא נגמרה וגם מקבל הסיוע לא עבר את מכסת השעות שהוא זכאי להן:

בתוך הלולאה הפונקציה מגרילה אינדקס רנדומלי מתוך רשימת הזמנים האפשריים לו, מוצאת את כל

המתנדבים היכולים (מבחינת השעות הרצויות להם) בזמן זה, ואם נמצא לפחות מתנדב אחד - מגרילה

מתנדב רנדומלי מתוך הרשימה, מכניסה את הנתונים ל value ומורידה מסך השעות ששובצו למקבל הסיוע

את משך הזמן ששובץ.

השיבוץ לכל דור ייעשה כמספר שנקבע שיהיה גודל הדור.



## קוד האלגוריתם:

אתחול משתנים הנצרכים לשימוש בפונקציות:

```
private readonly DBConnection DBCon = new DBConnection();
public List<TimeSlotProperties> timeSlotsPropertiesList = new List<TimeSlotProperties>();
private organization currentOrg;
private static Random Random = new Random();
```

כדי לחסוך בסיבוכיות של הבאת הנתונים מה DB בכל פעם מחדש – יציתי רשימות מסוג entity framework בתחילת המחלקה, המאותחלות בקונסטרקטורים, ואיתן עבדתי במהלך האלגוריתם.

```
private List<organization> listOfOrganizations;
private List<volunteer> listOfVolunteers;
private List<needy> listOfNeedies;
private List<volunteering_details> listOfVolunteeringDetails;
private List<neediness_details> listOfNeedinessDetails;
private List<volunteer_possible_time> listOfVolunteersPossibleTime;
private List<needy_possible_time> listOfNeediesPossibleTimes;
private List<time_slot> listOfTimeSlots;
private List<hour> listOfHours;
```

פונקציה שמאתחלת את הרשימות הנ"ל לערכים שב DB:

```
public void InitializeLists()
{
 listOfOrganizations = DBCon.GetDbSet<organization>().ToList();
 listOfVolunteers = DBCon.GetDbSet<volunteer>().ToList();
 listOfNeedies = DBCon.GetDbSet<needy>().ToList();
 listOfVolunteeringDetails = DBCon.GetDbSetWithIncludes<volunteering_details>(new
string[] { "volunteer_possible_time.time_slot" }).ToList();
 listOfNeedinessDetails = DBCon.GetDbSetWithIncludes<neediness_details>(new
string[] { "needy_possible_time.time_slot" }).ToList();
 listOfVolunteersPossibleTime =
DBCon.GetDbSetWithIncludes<volunteer_possible_time>(new string[] { "volunteering_details",
"time_slot" }).ToList();
 listOfNeediesPossibleTimes = DBCon.GetDbSetWithIncludes<needy_possible_time>(new
string[] { "neediness_details", "time_slot" }).ToList();
 listOfTimeSlots = DBCon.GetDbSet<time_slot>().ToList();
 listOfHours = DBCon.GetDbSet<hour>().ToList();
}
```

קונסטרקטור 1:

```
public TimeTableChromosome(DBConnection db, int orgCode)
{
 DBCon = db;
 InitializeLists();
 this.currentOrg = listOfOrganizations.Find(org=>org.org_code==orgCode);
 Generate();
}
```

קונסטרקטור 2:

```
public TimeTableChromosome(DBConnection db, List<TimeSlotProperties> slots)
{
 this.DBCon = db;
 this.timeSlotsPropertiesList = slots;
 InitializeLists();
}
```

פונקציה המקבלת מספר limit – ומגרילה מספר כלשהו בין 0 ל limit:

```
public int RandomIndex(int limit)
{
 return Random.Next(0, limit);
}
```

פונקציה המקבלת שעת התחלה ויום בשבוע, ומחזירה רשימה של כל המתנדבים שיכולים (מבחינת העדפות הזמן שלהם) בשעה זו וביום זה בארגון זה:

```
public List<volunteering_details> GetPossibleVolunteersForOrgAndHour(int
startHourCode, int dayOfWeek)
{
 var listOfVolunteeringDetailsInCurrentOrg = listOfVolunteeringDetails.FindAll(vd
=> vd.org_code == currentOrg.org_code).ToList();
 var listOfPossibleVolunteersInTheCurrentHour =
listOfVolunteeringDetailsInCurrentOrg.FindAll(
 vd => vd.volunteer_possible_time.ToList().FindAll(
 vpt => vpt.time_slot.start_at_hour<= startHourCode
 &&
vpt.time_slot.end_at_hour>=startHourCode+(currentOrg.avg_volunteering_time/15)
 && vpt.time_slot.day_of_week==dayOfWeek)
 .Count > 0).ToList();
 return listOfPossibleVolunteersInTheCurrentHour;
}
```

## פונקציית אתחול הנתונים:

```
public override void Generate()
{
 var list = new List<TimeSlotProperties>();
 var timeSlotProperties = new TimeSlotProperties();

 //רשימות פרטי הנזקקים שקשורים לארגון הזה
 var listOfNeedinessDetailsInCurrentOrg = listOfNeedinessDetails.FindAll(nd =>
nd.org_code == currentOrg.org_code).ToList();

 //רשימת כל הזמנים האפשריים למתנדב הרלוונטים לארגון//
 var listOfPossibleVolunteersInTheCurrentHour = new List<volunteering_details>();

 var currentNeedy = new needy();
 var requiredWeeklyHoursOfNeedy = 0.0;

 //רשימת המשבצות זמן האפשריות בארגון הזה לנזקק הנוכחי//
 var listOfPossibleTimeSlotsOfCurrentNeedy = new List<time_slot>();

 foreach (var currentNeedinessDetails in listOfNeedinessDetailsInCurrentOrg)
 {
 //אתחול המשתנים שיתאימו לנזקק הנוכחי//
 currentNeedy = listOfNeedies.First(needy => needy.needy_ID ==
currentNeedinessDetails.needy_ID);

 //בונה את רשימת כל המשבצות זמן שהנזקק הנוכחי זקוק להם בארגון הזה//
 listOfPossibleTimeSlotsOfCurrentNeedy =
listOfNeediesPossibleTimes.FindAll(npt => npt.needy_details_code ==
currentNeedinessDetails.neediness_details_code).ToList().Select(npt =>
npt.time_slot).ToList();

 //כל עוד לא עבר את המכסת שעות שהוא זכאי להן//
 while(requiredWeeklyHoursOfNeedy>currentNeedinessDetails.weekly_hours)
 {
 var randomIndex =
RandomIndex(listOfPossibleTimeSlotsOfCurrentNeedy.Count);

 //השעת התחלה מוגרלת מבין השעות האפשריות בטווח שהוגרל לפי זמן התנדבות ממוצע בארגון//
 var startHour =
Random.Next(listOfPossibleTimeSlotsOfCurrentNeedy[randomIndex].start_at_hour,
```

```
(listOfPossibleTimeSlotsOfCurrentNeedy[randomIndex].end_at_hour -
(currentOrg.avg_volunteering_time/15))+1);

//שימת המתנדבים האפשריים בשעה הזאת וביום הזה
listOfPossibleVolunteersInTheCurrentHour =
GetPossibleVolunteersForOrgAndHour(startHour,
listOfPossibleTimeSlotsOfCurrentNeedy[randomIndex].day_of_week);

//אם נמצאו מתנדבים מתאימים לשעה הזאת
if (listOfPossibleVolunteersInTheCurrentHour.Count > 0)
{
 var randomVolunteerIndex =
RandomIndex(listOfPossibleVolunteersInTheCurrentHour.Count);

 //איתחול משבצת זמן חדשה למאפיינים המתאימים
 timeSlotProperties = new TimeSlotProperties();
 timeSlotProperties.needy = currentNeedy;
 timeSlotProperties.volunteer =
listOfPossibleVolunteersInTheCurrentHour[randomVolunteerIndex].volunteer;
 timeSlotProperties.orgCode = currentOrg.org_code;

 //תאריך התחלה שווה לתאריך המוקדם יותר - תחילת פעילות הארגון או היום, אם זה שיבוץ באמצע שנה
 timeSlotProperties.time.start_at_date =
DateTime.Compare(currentOrg.activity_start_date, DateTime.Today) <= 0 ?
currentOrg.activity_start_date : DateTime.Today;
 timeSlotProperties.time.end_at_date = currentOrg.activity_end_date;

 //שאר המאפיינים לפי המשבצת זמן הנוכחית שמתאימה גם למתנדב וגם לנוקד
 timeSlotProperties.time.start_at_hour = startHour;
 timeSlotProperties.time.end_at_hour =
startHour+(currentOrg.avg_volunteering_time/15);
 timeSlotProperties.time.day_of_week =
listOfPossibleTimeSlotsOfCurrentNeedy[randomIndex].day_of_week;

 list.Add(timeSlotProperties);

 //הורדת סך השעות ששובצו מסך השעות של הנוקד הנוכחי
 requiredWeeklyHoursOfNeedy -= (currentOrg.avg_volunteering_time/15);
}
}
}
```

```
}
```

יצירת לוח זמנים חדש:

```
public override IChromosome CreateNew()
{
 var timeTableChromosome = new TimeTableChromosome(DBCon, currentOrg.org_code);
 timeTableChromosome.Generate();
 return timeTableChromosome;
}
```

שכפול לוח הזמנים:

```
public override IChromosome Clone()
{
 return new TimeTableChromosome(DBCon, timeSlotsPropertiesList);
}
```

שינוי אקראי בלוח הזמנים:

```
public override void Mutate()
{
 var indexToReplace = RandomIndex(timeSlotsPropertiesList.Count);

 //גרלת מספר שמציין את פעולת השינוי שמבצעים ברשימה - מחיקת משבצת עדכון משבצת או הוספה
 var action = RandomIndex(3);

 //פרטי הזדקקות נוכחיים - אם זה מצב הוספה הם מוגרלים ואם שינוי - הם שווים לאלמנט שבאינדקס שהוגרל
 var currentNeedinessDetails = new neediness_details();

 switch (action)
 {
 //מחיקת המשבצת
 case 0:
 timeSlotsPropertiesList.RemoveAt(indexToReplace);
 return;

 //עדכון משבצת
 case 1:
 var currentNeedy =
timeSlotsPropertiesList.ElementAt(indexToReplace).needy;
 currentNeedinessDetails =
listOfNeedinessDetails.Find(n=>n.needy_ID==currentNeedy.needy_ID &&
n.org_code==currentOrg.org_code);
 break;
 }
}
```

```
//הוספת משבצת
case 2:
 //כאן מגדילים פרטי הזדקקות רלוונטים לארגון, שעליהם מבצעים את ההוספה
 timeSlotsPropertiesList.Add(new TimeSlotProperties());
 indexToReplace = timeSlotsPropertiesList.Count-1;
 var relevantNeedinessDetails = listOfNeedinessDetails.FindAll(n =>
n.org_code == currentOrg.org_code);
 var randomNeedinessDetailsIndex =
RandomIndex(relevantNeedinessDetails.Count);
 currentNeedinessDetails =
relevantNeedinessDetails.ElementAt(randomNeedinessDetailsIndex);
 break;
}

//מכאן הקוד מתאים גם להוספה וגם לשינוי

//בונה את רשימת כל המשבצות זמן שהנוקק הנוכחי יכול בהם בארגון הזה
var listOfPossibleTimeSlotsOfCurrentNeedy =
listOfNeediesPossibleTimes.FindAll(npt => npt.needy_details_code ==
currentNeedinessDetails.neediness_details_code).ToList().Select(npt =>
npt.time_slot).ToList();

//משבצת זמן אקראית של הנוקק
var randomTimeSlotIndex =
RandomIndex(listOfPossibleTimeSlotsOfCurrentNeedy.Count);
var randomTimeSlot = listOfPossibleTimeSlotsOfCurrentNeedy[randomTimeSlotIndex];

//הגרלת מתנדב אקראי שמתאים לזמן ולארגון
var listOfPossibleVolunteersInTheCurrentHour =
GetPossibleVolunteersForOrgAndHour(randomTimeSlot.start_at_hour,randomTimeSlot.day_of_week);
var randomVolunteerIndex =
RandomIndex(listOfPossibleVolunteersInTheCurrentHour.Count);
var randomVolunteer =
listOfPossibleVolunteersInTheCurrentHour[randomVolunteerIndex].volunteer;

//אתחול משבצת חדשה
var newTimeSlot = new TimeSlotProperties();
newTimeSlot.needy = currentNeedinessDetails.needy;
newTimeSlot.volunteer = randomVolunteer;
newTimeSlot.orgCode = currentNeedinessDetails.org_code;
newTimeSlot.time = randomTimeSlot;
```

```
timeSlotsPropertiesList[indexToReplace] = newTimeSlot;
}
```

מיזוג 2 לוחות זמנים:

```
public override void Crossover(ICHromosome pair)
{
 var randomVal = Random.Next(0, timeSlotsPropertiesList.Count - 2);
 var otherChromosome = pair as TimeTableChromosome;
 for (int index = randomVal; index <
otherChromosome.timeSlotsPropertiesList.Count; index++)
 {
 timeSlotsPropertiesList[index] =
otherChromosome.timeSlotsPropertiesList[index];
 }
}
```

מחלקה המכילה פונקציה הנותנת ציון ללוח הזמנים:

```
public class FitnessFunction : IFitnessFunction
{
 DBConnection dbCon = new DBConnection();
 List<neediness_details> needinessDetailsInOrg = new List<neediness_details>();
 List<volunteering_details> volunteeringDetailsInOrg = new
List<volunteering_details>();

 public double Evaluate(ICHromosome chromosome)
 {
 double score = 1;
 var values = (chromosome as TimeTableChromosome).timeSlotsPropertiesList;
 var needinessDetailsInOrg =
dbCon.GetDbSet<neediness_details>().ToList().FindAll(nd => nd.org_code ==
values[0].orgCode).ToList();
 var volunteeringDetailsInOrg =
dbCon.GetDbSet<volunteering_details>().ToList().FindAll(vd => vd.org_code ==
values[0].orgCode).ToList();

 #region בדיקה של מספר שעות מתאים לנוקד ומרחקים בין כתובות
 var volunteersOfNeedy = new List<volunteer>();
 var currentNeedyHours = 0;
```

```

var currentVolunteerHours = 0;

foreach (var item in needinessDetailsInOrg)
{
 volunteersOfNeedy = values.FindAll(slot => slot.needy.needy_ID ==
item.needy_ID).Select(slot => slot.volunteer).ToList();
 currentNeedyHours = volunteersOfNeedy.Count;
 volunteersOfNeedy = volunteersOfNeedy.Distinct().ToList();
 score -= (item.weekly_hours - currentNeedyHours);

 //עבור מרחק גדול מעשר דקות יורד ציון
 foreach (var volunteer in volunteersOfNeedy)
 {
 score -=
(GoogleMaps.GetDistanceInMinutes(volunteer.volunteer_address,
item.needy.needy_address).Result-10);
 }
}

#endregion

#region בדיקה של מספר שעות מתאים למתנדב
foreach (var item in volunteeringDetailsInOrg)
{
 currentVolunteerHours = (values.FindAll(slot =>
slot.volunteer.volunteer_ID ==
item.volunteer_ID).ToList().Count()*(item.organization.avg_volunteering_time))/60;
 score -= (item.weekly_hours - currentVolunteerHours);
}
#endregion

#region בדיקה של חפיפות במערכת החדשה כולל הלוח הקיים, וכולל הפרש הגיוני בין ההתנדבויות של אותו
מתנדב

var allSchedule = dbCon.GetDbSet<schedule>().ToList();

//איחוד המערכת הכללית למערכת הנוכחית
var timeSlotProperties = new TimeSlotProperties();
var scheduleAsTimeSlotPropertiesList = new List<TimeSlotProperties>();
var listOfAllOverLaps = new List<TimeSlotProperties>();

foreach (var scheduleSlot in allSchedule)
{

```



```

 timeSlotProperties = new TimeSlotProperties();
 timeSlotProperties.needy = scheduleSlot.neediness_details.needy;
 timeSlotProperties.volunteer =
scheduleSlot.volunteering_details.volunteer;
 timeSlotProperties.orgCode = scheduleSlot.neediness_details.org_code;
 timeSlotProperties.time = scheduleSlot.time_slot;

 scheduleAsTimeSlotPropertiesList.Add(timeSlotProperties);
 }
 values.AddRange(scheduleAsTimeSlotPropertiesList);

 var GetOverLaps = new Func<TimeSlotProperties,
List<TimeSlotProperties>>(current =>
 values.Except(new[] { current })
 .Where(slot => slot.volunteer.volunteer_ID == current.volunteer.volunteer_ID
 || slot.needy.needy_ID == current.needy.needy_ID)
 .Where(slot => slot.time.end_at_date > DateTime.Today)
 .Where(slot => slot.time.day_of_week == current.time.day_of_week)
 // בדיקות של שעות התחלה וסיום חופפות - זהות או שאחד מתחיל באמצע השני או שההפרש ביניהם קטן משעה
 .Where(slot => slot.time.start_at_hour <= current.time.start_at_hour
 && slot.time.end_at_hour >= current.time.end_at_hour
 || current.time.start_at_hour <= slot.time.start_at_hour
 && current.time.end_at_hour >= slot.time.end_at_hour
).ToList());

 foreach (var item in values)
 {
 var overLaps = GetOverLaps(item);
 listOfAllOverLaps.AddRange(overLaps);
 }

 // רשימת כל החפיפות בלוח, כולל השיבוץ החדש ובלי כפילויות
 listOfAllOverLaps = listOfAllOverLaps.Distinct().ToList();

 score -= listOfAllOverLaps.Count() - 1;
 #endregion

 #region
 בדיקה שאין למתנדב יותר מ 3 התנדבויות ליום
 volunteeringDetailsInOrg = dbCon.GetDbSet<volunteering_details>();
 var dailyVolunteeringsCounter = 0;
 var slotsOfVolunteer = new List<TimeSlotProperties>();

```

```

foreach (var item in volunteeringDetailsInOrg)
{
 slotsOfVolunteer = values.Where(slot => slot.volunteer.volunteer_ID ==
item.volunteer_ID).Where(slot => slot.time.end_at_date >= DateTime.Today).ToList();

 for (int i = 0; i <= 7; i++)
 {
 dailyVolunteeringsCounter = slotsOfVolunteer.Where(slot =>
slot.time.day_of_week == i).Count();
 }

 if (dailyVolunteeringsCounter > 3)
 {
 score -= (dailyVolunteeringsCounter-3);
 }
}
#endregion

return Math.Pow(Math.Abs(score), -1);
}
}
}

```

הפעלת האלגוריתם עבור ארגון מסוים והכנסת נתוני השיבוץ ל DB:

```

public void Gentic(int orgCode)
{
 DBConnection dBConnection = new DBConnection();

 //דכון הלוח הקיים של הארגון שכל מה שיש עד עכשיו יגמר בתאריך של היום כי מעכשיו תהיה מערכת חדשה
 List<schedule> scheduleToDelete = dBConnection.GetDbSet<schedule>().FindAll(s =>
s.neediness_details.organization.org_code == orgCode).ToList();
 foreach (var item in scheduleToDelete)
 {
 item.time_slot.end_at_date = DateTime.Today;
 dBConnection.Execute<schedule>(item, dBConnection.ExecuteActions.Update);
 }

 Population population = new Population(100, new TimeTableChromosome(dBConnection,
orgCode),
new
TimeTableChromosome.FitnessFunction(), new EliteSelection());
}

```

```
int i = 0;
while (true)
{
 population.RunEpoch();
 i++;
 if (population.FitnessMax >= 0.99 || i >= 100)
 {
 break;
 }
}
```

```
List<TimeSlotProperties> value = (population.BestChromosome as
TimeTableChromosome).timeSlotsPropertiesList.ToList();
```

*//הכנסת פרטי השיבוץ למסד הנתונים*

```
schedule newScheduleSlot = new schedule();
time_slot newTimeSlot;
TimeSlotBL timeSlotBL = new TimeSlotBL();
```

```
foreach (var item in value)
{
```

```
 newTimeSlot = new time_slot();
 newTimeSlot.start_at_date = item.time.start_at_date;
 newTimeSlot.end_at_date = item.time.end_at_date;
 newTimeSlot.start_at_hour = item.time.start_at_hour;
 newTimeSlot.end_at_hour = item.time.end_at_hour;
 newTimeSlot.day_of_week = item.time.day_of_week;
 newTimeSlot.time_slot_code =
```

```
timeSlotBL.InsertTimeSlot(TimeSlotBL.ConvertTimeSlotToModel(newTimeSlot));
```

```
 newScheduleSlot.time_slot_code = item.time.time_slot_code;
 newScheduleSlot.volunteering_details_code =
item.volunteer.volunteering_details.First(v => v.org_code ==
item.orgCode).volunteering_details_code;
 newScheduleSlot.neediness_details_code = item.neediness_details.First(n
=> n.org_code == item.orgCode).neediness_details_code;
```

```
dbConnection.Execute<schedule>((newScheduleSlot),dbConnection.ExecuteActions.Insert);
```

```
 }
}
```

## תיאור התוכנה:

### סביבת עבודה

Visual studio 2019

Visual studio code 1.67.2

### שפות תכנות

צד שרת: טכנולוגיית web api, שפת c#

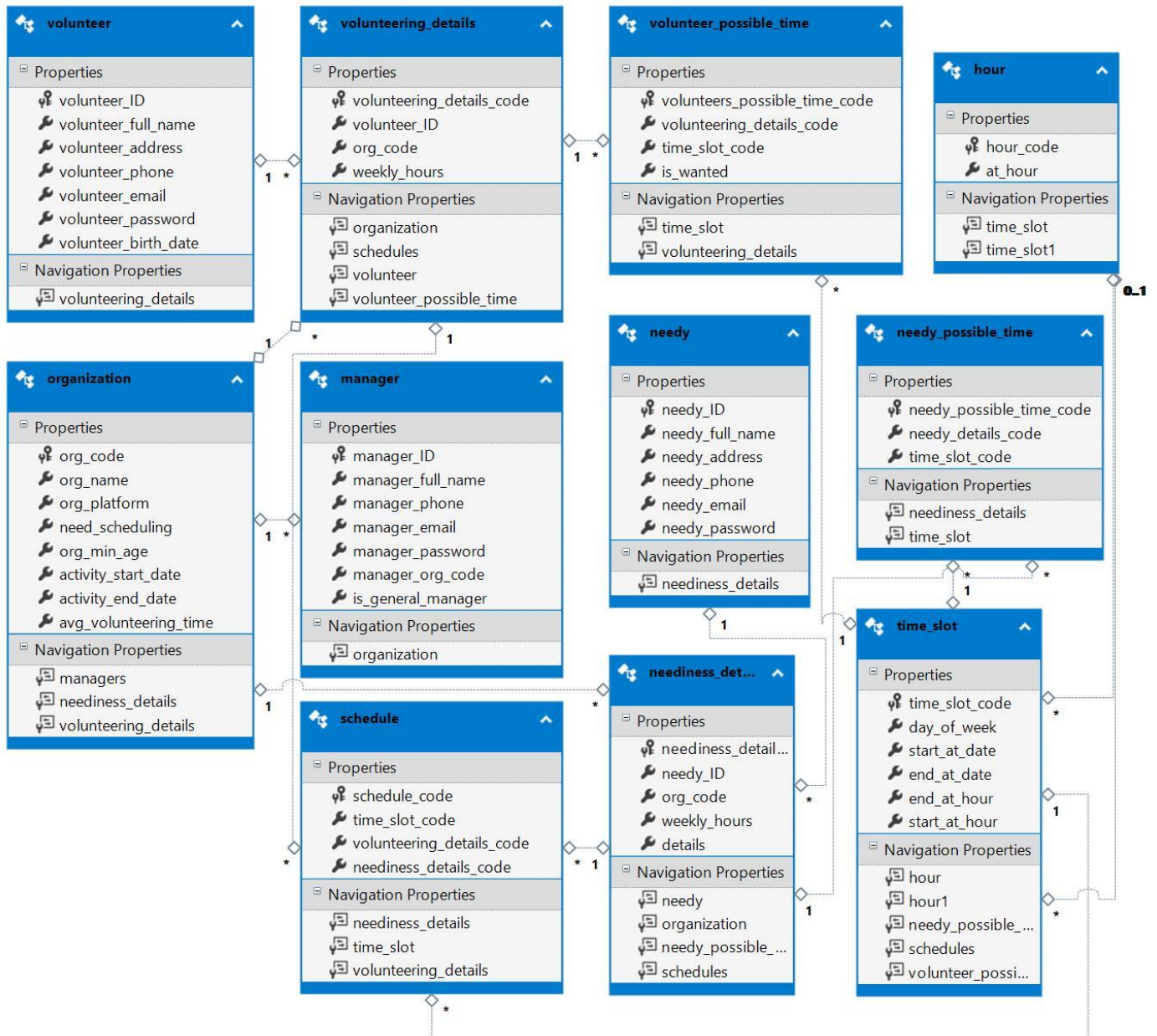
צד לקוח: טכנולוגיית angular, שפות SCSS, HTML ו Type script.

## תיכון המערכת

השתמשתי בטכנולוגיית API שהוא ערכה של ספריות קוד, פקודות, פונקציות ופרוצדורות מן המוכן, בהן יכולים המתכנתים לעשות שימוש פשוט, בלי להידרש לכתוב אותן בעצמם כדי שיוכלו להשתמש במידע של היישום ממנו הם רוצים להשתמש לטובת היישום שלהם.

## תיאור מסד נתונים

### תרשים טבלאות + קשרי גומלין



## תיאור הטבלאות ב data base:

### • Volunteer

מכילה את פרטי המתנדבים.

| מפתח | שם שדה               | טיפוס נתונים | תיאור               | שדה חובה? |
|------|----------------------|--------------|---------------------|-----------|
| PK   | volunteer_ID         | String       | ת.ז מתנדב           | V         |
|      | volunteer_full_name  | String       | שם מתנדב            | V         |
|      | Volunteer_address    | String       | כתובת מתנדב         | V         |
|      | Volunteer_birth_date | dateTime     | תאריך לידה של מתנדב | V         |
|      | Volunteer_email      | String       | כתובת מייל של מתנדב | V         |
|      | Volunteer_phone      | string       | מס' טלפון של מתנדב  |           |
|      | Volunteer_password   | string       | סיסמה לאתר          | V         |

### • needy

מכילה את פרטי מקבלי העזרה

| מפתח | שם שדה          | טיפוס נתונים | תיאור                   | שדה חובה? |
|------|-----------------|--------------|-------------------------|-----------|
| PK   | needy_ID        | String       | ת.ז מקבל עזרה           | V         |
|      | needy_full_name | String       | שם מקבל עזרה            | V         |
|      | needy_address   | String       | כתובת מקבל עזרה         | V         |
|      | needy_email     | String       | כתובת מייל של מקבל עזרה | V         |
|      | needy_phone     | string       | מס' טלפון של מקבל עזרה  |           |
|      | needy_password  | String       | סיסמה לאתר              | V         |

### • organization

מכילה את פרטי הארגונים

| מפתח | שם שדה                | טיפוס נתונים  | תיאור                        | שדה חובה? |
|------|-----------------------|---------------|------------------------------|-----------|
| PK   | Org_code              | Int, מספור רץ | קוד ארגון, מספור רץ          | V         |
|      | org_name              | String        | שם ארגון                     | V         |
|      | Org_platform          | String        | מצע הארגון (הסבר על פעילותו) |           |
|      | Need_scheduling       | Bool          | האם צריך שיבוץ               | V         |
|      | Org_min_age           | Int           | גיל מינימלי להתנדבות         | V         |
|      | Activity_start_date   | DateTime      | תאריך תחילת פעילות הארגון    | V         |
|      | Activity_end_date     | DAtetime      | תאריך סיום פעילות הארגון     | V         |
|      | avg_volunteering_time | int           | זמן ממוצע (בשעות) להתנדבות   | V         |

• **manager**

מכילה את פרטי מנהלי הארגונים

| מפתח | שם שדה             | טיפוס נתונים | תיאור                                                                               | שדה חובה? |
|------|--------------------|--------------|-------------------------------------------------------------------------------------|-----------|
| PK   | manager_ID         | String       | ת.ז מנהל                                                                            | ✓         |
|      | manager_full_name  | String       | שם מנהל                                                                             | ✓         |
|      | Volunteer_email    | String       | כתובת מייל של מנהל                                                                  | ✓         |
|      | Volunteer_phone    | string       | מס' טלפון של מנהל                                                                   |           |
| FK   | Manager_org_code   | int          | קוד הארגון אותו הוא מנהל                                                            | ✓         |
|      | Manager_password   | String       | סיסמה לאתר                                                                          | ✓         |
|      | is_general_manager | Bool         | מציין האם המנהל הוא מנהל כללי (ממנהלי יד ליד) או רק מנהל ארגון (לצורך הרשאות הגישה) | ✓         |

• **volunteering\_details**

מכילה פרטי התנדבויות עבור כל מתנדב (מקשרת בין מתנדב לבין ארגון, כדי לאפשר התנדבות בכמה ארגונים לכל מתנדב)

| מפתח | שם שדה                    | טיפוס נתונים  | תיאור                         | שדה חובה? |
|------|---------------------------|---------------|-------------------------------|-----------|
| PK   | Volunteering_details_code | Int, מספור רץ | קוד פרטי התנדבות, מספור רץ    | ✓         |
| FK   | Volunteer_ID              | String        | ת.ז מתנדב                     | ✓         |
| FK   | Org_code                  | int           | קוד ארגון                     | ✓         |
|      | weekly_hours              | double        | מס' שעות התנדבות רצויות בשבוע | ✓         |

• **neediness\_details**

מכילה פרטי סיוע עבור מקבל עזרה (מקשרת בין מקבל עזרה לבין ארגון, כדי לאפשר קבלת סיוע מכמה ארגונים לכל מקבל סיוע)

| מפתח | שם שדה                 | טיפוס נתונים  | תיאור                          | שדה חובה? |
|------|------------------------|---------------|--------------------------------|-----------|
| PK   | neediness_details_code | Int, מספור רץ | קוד פרטי סיוע, מספור רץ        | ✓         |
| FK   | needy_ID               | String        | ת.ז מקבל עזרה                  | ✓         |
| FK   | Org_code               | int           | קוד ארגון                      | ✓         |
|      | weekly_hours           | double        | מס' שעות עזרה שזכאי להן בשבוע  | ✓         |
|      | Details                | String        | פרטים טכניים אודות הסיוע הנדרש |           |

• hours

מכילה את כל השעות ביממה בעיגול לרבעי שעה

| מפתח | שם שדה    | טיפוס נתונים  | תיאור             | שדה חובה? |
|------|-----------|---------------|-------------------|-----------|
| PK   | Hour_code | int, מספור רץ | קוד שעה, מספור רץ | V         |
|      | At_hour   | time          | שעה מסוימת ביממה  | V         |

• time\_slot

מכילה פרטי זמן עבור משבצת זמן, לשימוש באילוצים ובמערכת סופית

| מפתח | שם שדה         | טיפוס נתונים  | תיאור                     | שדה חובה? |
|------|----------------|---------------|---------------------------|-----------|
| PK   | Time_slot_code | int, מספור רץ | קוד משבצת זמן, מספור רץ   | V         |
| FK   | Start_at_date  | DateTime      | תאריך התחלה               | V         |
| FK   | End_at_date    | DateTime      | תאריך סיום                | V         |
| FK   | Start_at_hour  | int           | קוד שעת התחלה מטבלת hours | V         |
| FK   | End_at_hour    | int           | קוד שעת סיום מטבלת hours  | V         |
|      | Day_of_week    | Int           | יום בשבוע                 | V         |

• volunteer\_possible\_time

מכילה אילוצי מתנדבים – תאריכים, ימים ושעות בהם נוח להם להתנדב – עבור כל סוג התנדבות בנפרד.

| מפתח | שם שדה                       | טיפוס נתונים  | תיאור                          | שדה חובה? |
|------|------------------------------|---------------|--------------------------------|-----------|
| PK   | Volunteer_possible_time_code | int, מספור רץ | קוד זמן אפשרי למתנדב, מספור רץ | V         |
| FK   | Volunteering_details_code    | int           | קוד פרטי התנדבות               | V         |
| FK   | Time_slot_code               | int           | קוד משבצת זמן                  | V         |
|      | Is_wanted                    | bool          | האם יכול או רוצה               | V         |

• needy\_possible\_time

מכילה אילוצי מקבלי עזרה – תאריכים, ימים ושעות בהם הוא זקוק לעזרה – עבור כל סוג סיוע בנפרד.

| מפתח | שם שדה                   | טיפוס נתונים  | תיאור                              | שדה חובה? |
|------|--------------------------|---------------|------------------------------------|-----------|
| PK   | needy_possible_time_code | int, מספור רץ | קוד זמן אפשרי למקבל עזרה, מספור רץ | V         |
| FK   | neediness_details_code   | int           | קוד פרטי סיוע                      | V         |
| FK   | Time_slot_code           | int           | קוד משבצת זמן                      | V         |



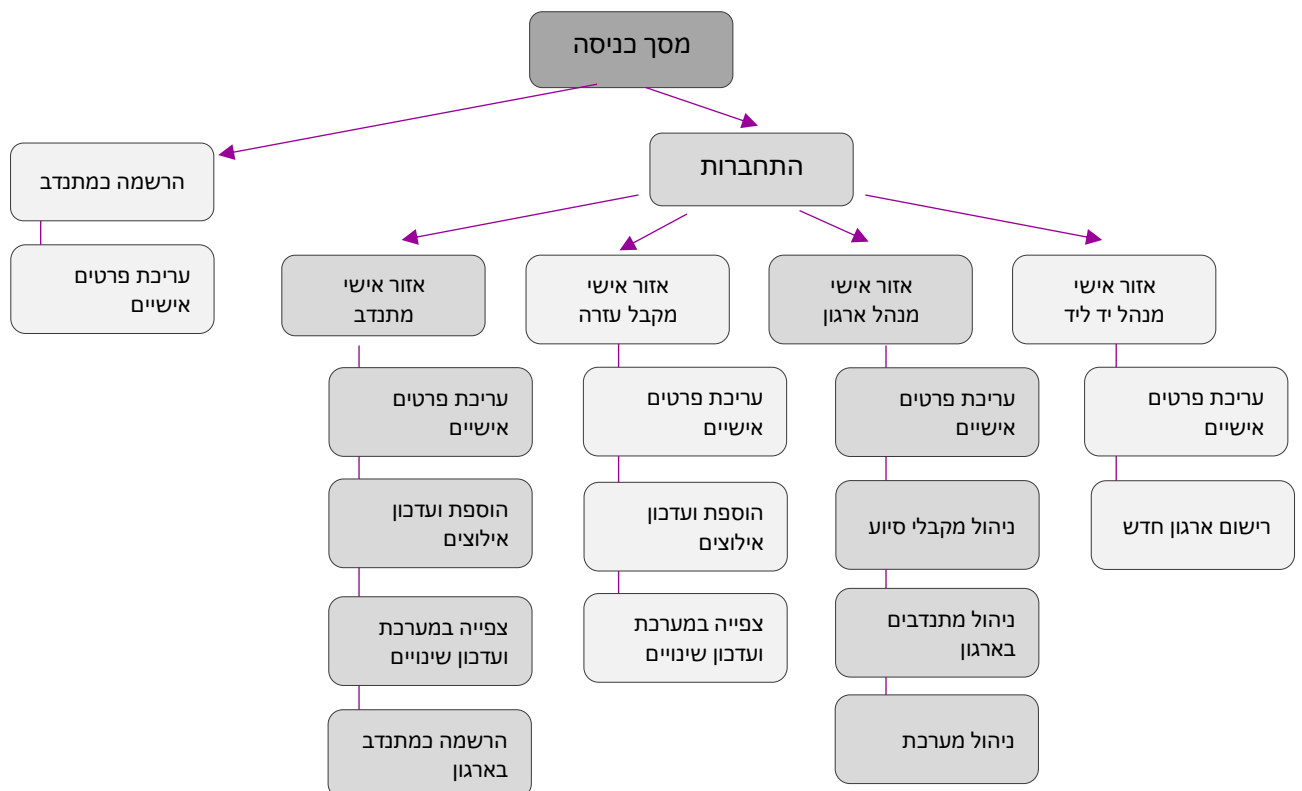
• schedule

מכילה פרטים עבור המערכת המשובצת

| מפתח | שם שדה                    | טיפוס נתונים  | תיאור               | שדה חובה? |
|------|---------------------------|---------------|---------------------|-----------|
| PK   | Schedule_code             | Int, מספור רץ | קוד מערכת, מספור רץ | V         |
| FK   | Time_slot_code            | int           | קוד משבצת זמן       | V         |
| FK   | Volunteering_details_code | int           | קוד פרטי התנדבות    | V         |
| FK   | Neediness_details_code    | int           | קוד פרטי סיוע       | V         |

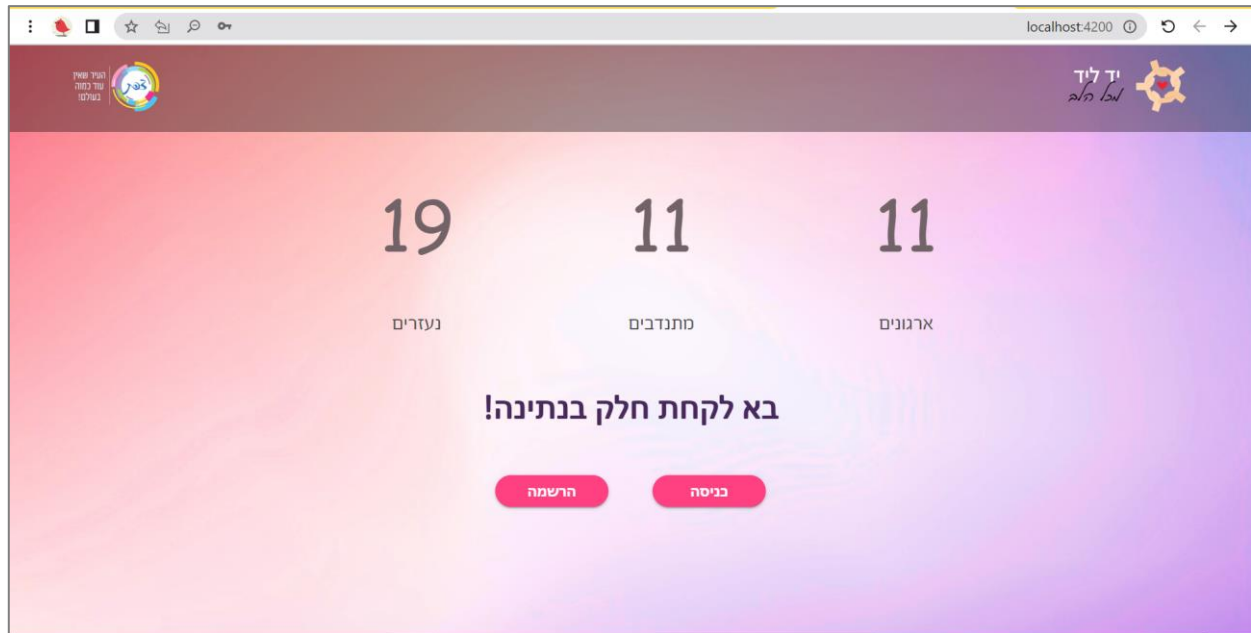
תיאור מסכים:

תרשימים מסכים:



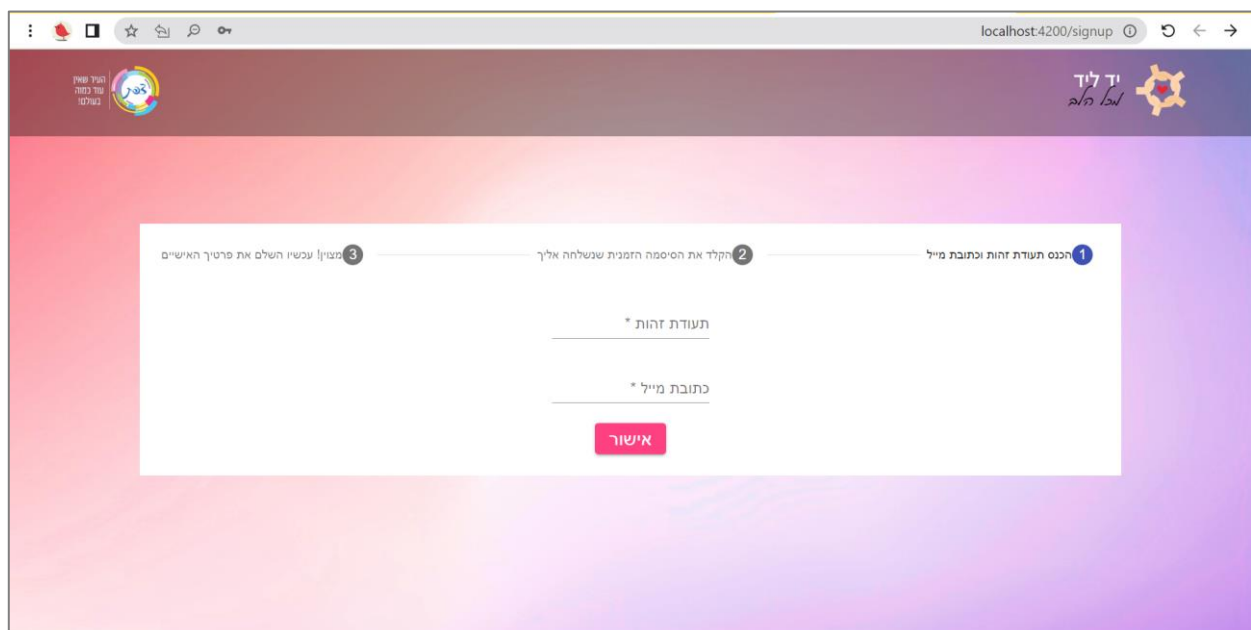
## פירוט מסכים:

### דף הבית:



מסך הרשמה – המשתמש מתבקש להקליד תעודת זהות וכתובת מייל, המערכת שולחת לכתובת המייל סיסמה לאימות, והמשתמש מקליד אותה במסך המתאים. במידה והסיסמאות תואמות – נפתח לו טופס בו הוא משלים את פרטיו האישיים: הרשמה עצמית כזו אפשרית רק כמתנדב. שאר המשתמשים מוכנסים ע"י מי שמעליהם.

### מסך הרשמה שלב 1 – הכנסת תעודת זהות וסיסמה:



## מסך הרשמה שלב 2 – הקלדת הסיסמה שנשלחה לאימות:

The screenshot shows a web browser window with the URL `localhost:4200/signup`. The page has a header with the 'יד ליד' logo and navigation links. A progress bar at the top indicates three steps: 1. 'הכנס תעודת זהות וכתובת מייל' (Completed), 2. 'הקלד את הסיסמה הזמנית שנשלחה אליך' (Current step), and 3. 'מצוין! עכשיו השלם את פרטיך האישיים' (Upcoming).

The main form area contains a label 'סיסמה \*' and an input field with the placeholder text 'סיסמה זמנית'. Below the input field is a red 'אישור' (Confirm) button.

## מסך הרשמה שלב 3 – לאחר אימות הסיסמה יש להקליד את הפרטים האישיים:

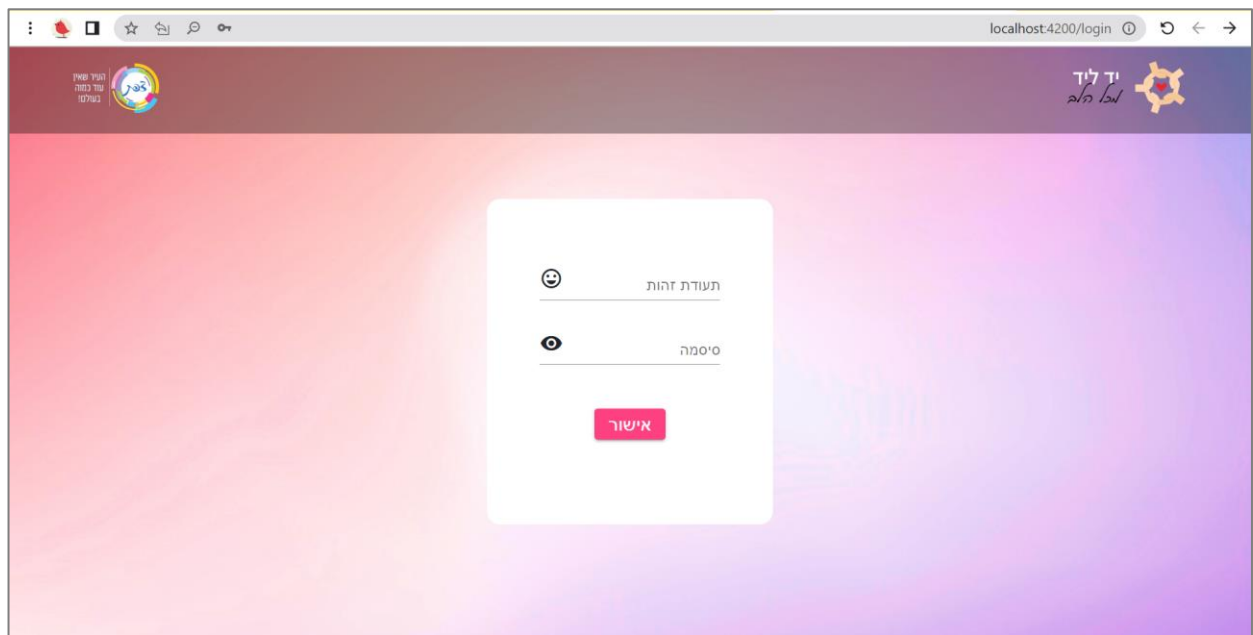
The screenshot shows the same web browser window, but the URL is now `localhost:4200/signup/personal-details/add/325641111`. The progress bar shows step 2 as completed and step 3 as the current step.

The form contains several input fields for personal details:

- שם מלא** (Full Name): Input field.
- תעודת זהות** (ID Card): Input field with the value `111146523`.
- כתובת** (Address): Input field.
- טלפון** (Phone): Input field.
- סיסמה לאתר** (Site Password): Input field with a red error message 'חובה להכניס סיסמה' (Password is required).
- כתובת מייל** (Email): Input field with a red error message 'חובה להכניס כתובת מייל תקינה' (Valid email is required).
- תאריך לידה** (Date of Birth): Input field with a calendar icon.

At the bottom of the form is a red 'אישור' (Confirm) button.

## מסך כניסה – המשתמש מקליד תעודת זהות וסיסמה ולפיהן מועבר לאזור האישי המתאים:



## מסך אזור אישי של מנהל כללי – מנהל מערך ההתנדבות:

כאן מוצגים לו כל הארגונים הרשומים במערכת, הוא יכול לצפות בפרטיהם, לערוך אותם ולהוסיף ארגונים ומנהלים חדשים.

בטבלה המוצגת כאן ניתן לראות את הסיכום – וכן – עבור ארגונים שלא משויך אליהם מנהל מוצג אייקון של מנהל +, ועבור ארגונים שמשויך להם מנהל מוצג אייקון של מנהל בלבד.

בלחיצה על מנהל + : יפתח טופס הוספת מנהל חדש, ובלחיצה על מנהל יפתח טופס פרטי המנהל הרלוונטי.

## בלחיצה על עריכת פרטי ארגון (העמודה הראשונה בטבלה הנ"ל):

בלחיצה על כפתור "הוסף ארגון" – יפתח הטופס הנ"ל כשהוא ריק, בסטטוס הוספה.

## בלחיצה על אייקון של מנהל ארגון, עבור ארגון עם מנהל – מסך פרטי מנהל ארגון:

בלחיצה על אייקון של מנהל +, עבור ארגון שלא משויך אליו מנהל, או בלחיצה על "הוסף מנהל" - יפתח הטופס הנ"ל כשהוא ריק, בסטטוס הוספה, ובשדה של ארגון יופיע הארגון הנבחר.

## אזור אישי של מנהל ארגון:

למנהל ארגון מוצגות 4 כרטיסיות: פרטים אישיים, המערכת שלי, ניהול החברים בארגון וניהול המתנדבים בארגון.

### מסך פרטים אישיים:

localhost:4200/login/org-manager-area/324447687/manager-personal-details/update

ניהול המתנדבים בארגון | ניהול החברים בארגון | המערכת שלי | **פרטים אישיים**

שם מלא: מאיר זכאי | תעודת זהות: 786744423

כתובת מיל: tamar96115@gmail.com | טלפון: 0527137777

ארגון: עזרה וחסד | סיסמה לאתר: 12345

**אישור**

### מסך המערכת שלי – מערכת הארגון:

localhost:4200/login/org-manager-area/324447687/schedule

ניהול המתנדבים בארגון | ניהול החברים בארגון | **המערכת שלי** | פרטים אישיים

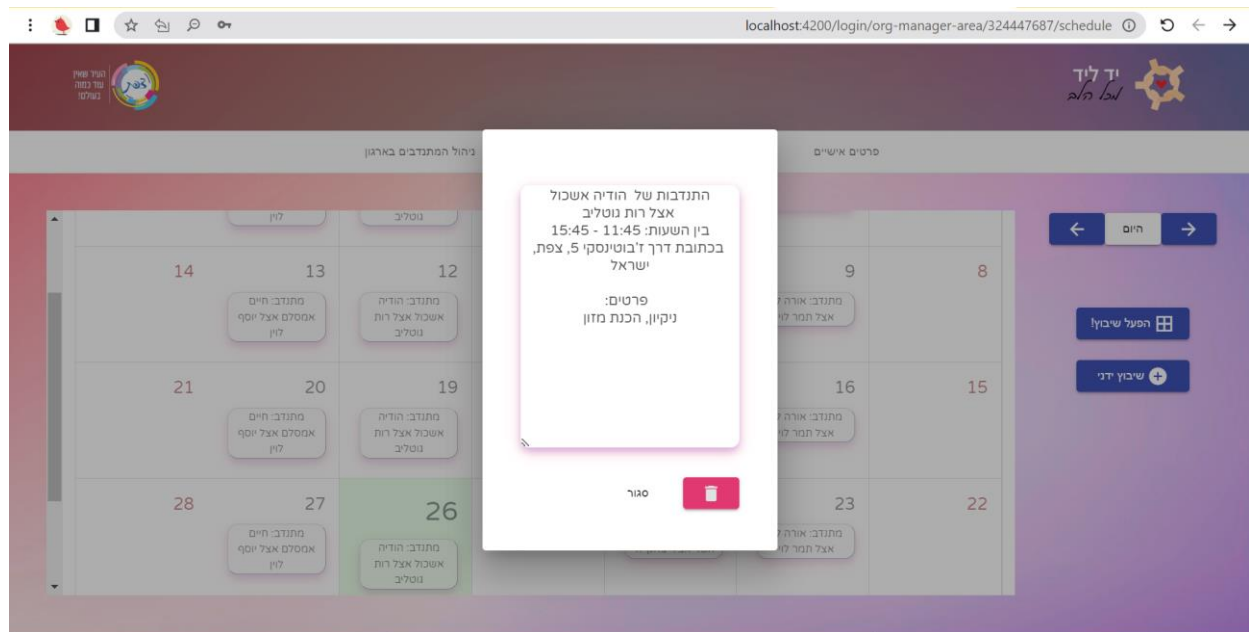
| יום ראשון                               | יום שני                               | יום שלישי                                | יום רביעי                                 | יום חמישי                               | יום שישי                              | יום שבת                                  |
|-----------------------------------------|---------------------------------------|------------------------------------------|-------------------------------------------|-----------------------------------------|---------------------------------------|------------------------------------------|
| 1<br>מתנדב: אורח לוי<br>אצל תמר לוי     | 2<br>מתנדב: מרדכי<br>אשר אצל יצחק לוי | 3<br>מתנדב: הודיה<br>אשכול אצל רות נטליב | 4<br>מתנדב: חיים<br>אמסלם אצל יוסף לוי    | 5<br>מתנדב: אורח לוי<br>אצל תמר לוי     | 6<br>מתנדב: מרדכי<br>אשר אצל יצחק לוי | 7<br>מתנדב: הודיה<br>אשכול אצל רות נטליב |
| 8<br>מתנדב: חיים<br>אמסלם אצל יוסף לוי  | 9<br>מתנדב: אורח לוי<br>אצל תמר לוי   | 10<br>מתנדב: מרדכי<br>אשר אצל יצחק לוי   | 11<br>מתנדב: הודיה<br>אשכול אצל רות נטליב | 12<br>מתנדב: חיים<br>אמסלם אצל יוסף לוי | 13<br>מתנדב: אורח לוי<br>אצל תמר לוי  | 14<br>מתנדב: מרדכי<br>אשר אצל יצחק לוי   |
| 15<br>מתנדב: חיים<br>אמסלם אצל יוסף לוי | 16<br>מתנדב: אורח לוי<br>אצל תמר לוי  | 17<br>מתנדב: מרדכי<br>אשר אצל יצחק לוי   | 18<br>מתנדב: הודיה<br>אשכול אצל רות נטליב | 19<br>מתנדב: חיים<br>אמסלם אצל יוסף לוי | 20<br>מתנדב: אורח לוי<br>אצל תמר לוי  | 21<br>מתנדב: מרדכי<br>אשר אצל יצחק לוי   |

הפעל שיבוץ | שיבוץ ידני

שיבוץ ידני –  
הסבר בעמוד  
הבא

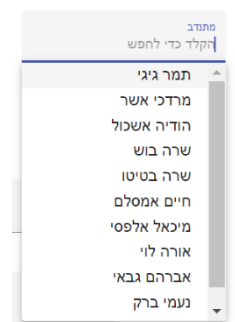
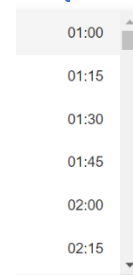
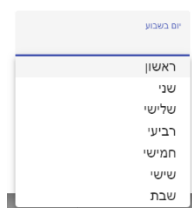
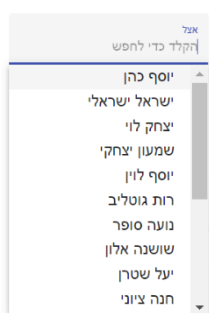
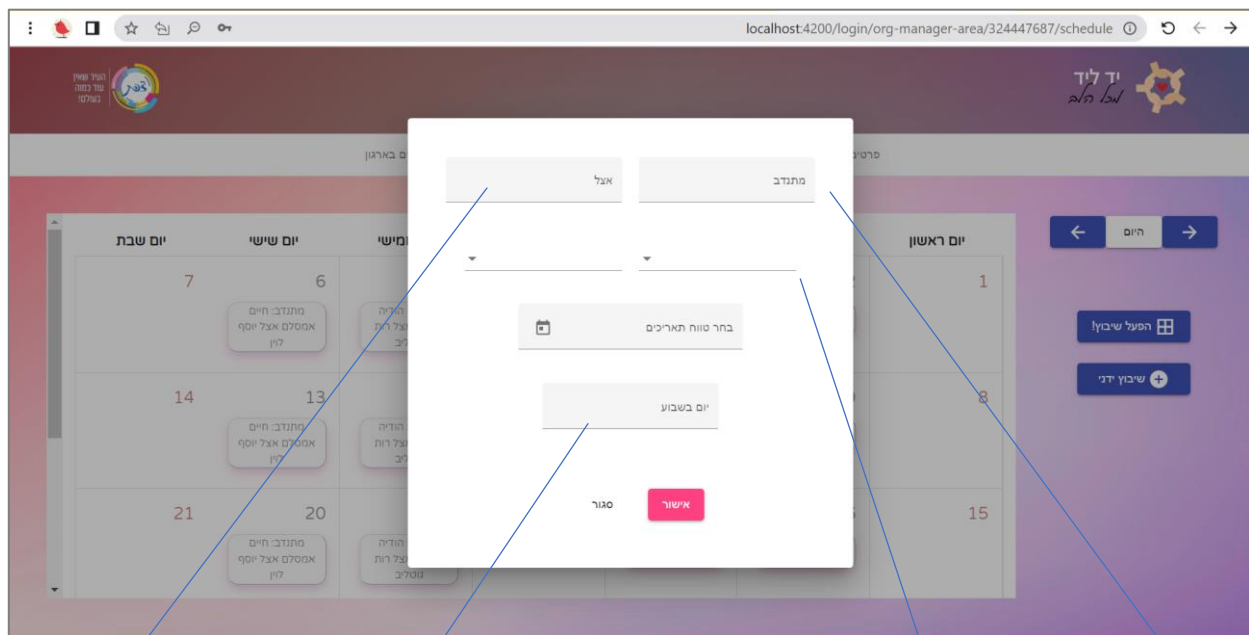
כפתור להפעלת  
אלגוריתם השיבוץ  
בארגון זה

## בלחיצה על התנדבות כלשהי בלוח יופיעו פרטי ההתנדבות:



לחיצה על אייקון המחיקה תבטל את ההתנדבות המסוימת ביום שנבחר.

## בלחיצה על שיבוץ ידני:



## מסך ניהול חברים (מקבלי סיוע) בארגון:

במסך זה מוצגים למנהל כל מקבלי הסיוע שבארגונו, עם אפשרויות מתקדמות לניהולם.

| פרטים             | זמן התנדבות לשבוע, בדקות | מייל                 | טלפון      | כתובת                        | שם           | מחק | ערוך |
|-------------------|--------------------------|----------------------|------------|------------------------------|--------------|-----|------|
| ניקיון ובייביסיטר | 6                        | tamar96115@gmail.com | 0583456545 | אליהו פרומצ'קו 5, צפת, ישראל | ישראל ישראלי |     |      |
|                   | 8                        | tamar96115@gmail.com | 0528767876 | זלמן שז"ר 7, צפת, ישראל      | שמעון יצחקי  |     |      |
|                   | 3                        | tamar96115@gmail.com | 0533123654 | לוחמי הנטאות 18, צפת, ישראל  | יוסף לוי     |     |      |
|                   | 7                        | tamar96115@gmail.com | 0552345654 | צה"ל 1, צפת, ישראל           | יצחק לוי     |     |      |
|                   | 4                        | tamar96115@gmail.com | 0548798888 | הרצל צפת ישראל               | יוסף כהן     |     |      |
| ניקיון, הכנת מזון | 6                        | tamar96115@gmail.com | 0587654345 | דרך ד'בוטנסקי 5, צפת, ישראל  | רות נטליב    |     |      |

בלחיצה על כפתור "הוספה ידנית" – יפתח טופס רישום של מקבל סיוע חדש, בו יכניס המנהל את הפרטים הדרושים:

## מסך הוספה ידנית:

שם מלא

תעודת זהות

כתובת

טלפון

סיסמה לאתר

כתובת מייל

**אישור**

סגור



בלחיצה על כפתור "העלאת קובץ" – יפתח חלון העלאת קובץ והמנהל יתבקש להעלות קובץ אקסל עם פרטי מקבלי הסיוע החדשים שברצונו להוסיף. לאחר ההעלאה – יוצר קובץ המכיל את הסיסמאות שהוגרלו עבור כל אחד ממקבלי הסיוע החדשים, אותו יוכל המנהל להוריד, וכן תוצג הודעה על כמות מקבלי הסיוע החדשים שהועלו.

**מסך זה כפי שנראה לאחר העלאת קובץ:**

| עורר | מחק | שם          | כתובת                         | טלפון      | מייל                 | זמן התנדבות לשבוע, בדקות | פרטים |
|------|-----|-------------|-------------------------------|------------|----------------------|--------------------------|-------|
| 1    |     | ישראל       | אליהו פרומצ'נקו 5, צפת, ישראל | 0583456545 | tamar96115@gmail.com | 6                        |       |
| 2    |     | שמעון יצחקי | זלמן שז"ר 7, צפת, ישראל       | 0528767876 | tamar96115@gmail.com | 8                        |       |
| 3    |     | יצחק לוי    | צה"ל 1, צפת, ישראל            | 0552345654 | tamar96115@gmail.com | 7                        |       |
| 4    |     | יוסף כהן    | הרצל 55 צפת ישראל             | 0548798888 | tamar96115@gmail.com | 4                        |       |
| 5    |     | יעל שטרן    | מ"נ יורדי הסירה 7, צפת, ישראל | 0547651222 | tamar96115@gmail.com | 9                        |       |
| 6    |     | חנה ציוני   | צה"ל 11 צפת, ישראל            | 0589784444 | tamar96115@gmail.com | 10                       |       |

רענן טבלה

הוספת חברים לארגון

העלאת קובץ

הוספת ידנית

קובץ הסיסמאות מוכן!

הורדה

חברים נוספו בהצלחה 19

|   | A          | B     |
|---|------------|-------|
| 1 | תעודת זהות | סיסמה |
| 2 | 123565456  | 76333 |
| 3 | 456876568  | 81634 |

מבנה הקובץ:

**מסך ניהול מתנדבים בארגון – עובד בדיוק כמו ניהול החברים בארגון:**

| עורר | מחק | שם          | כתובת                 | טלפון      | מייל                 | תאריך לידה   | מספר שעות התנדבות שבועי רצוי |
|------|-----|-------------|-----------------------|------------|----------------------|--------------|------------------------------|
| 1    |     | הודיה אשכול | התבור 77, צפת, ישראל  | 0549876123 | tamar96115@gmail.com | Jan 7, 1999  | 6                            |
| 2    |     | מרדכי אשר   | האר"י 7, צפת, ישראל   | 0589842169 | tamar96115@gmail.com | May 19, 2002 | 4                            |
| 3    |     | שרה בוש     | צה"ל 10 צפת, ישראל    | 0548769876 | tamar96115@gmail.com | Oct 2, 1991  | 7                            |
| 4    |     | מיכאל אלפסי | צה"ל 10 צפת, ישראל    | 0559872345 | tamar96115@gmail.com | Dec 12, 1995 | 2                            |
| 5    |     | חיים אמסלם  | יוני 5, צפת           | 0553988098 | tamar96115@gmail.com | Jan 4, 2000  | 3                            |
| 6    |     | אורה לוי    | דוד רמז 1, צפת, ישראל | 0521565409 | tamar96115@gmail.com | Oct 10, 2001 | 6                            |

רענן טבלה

הוספת מתנדבים לארגון

העלאת קובץ

הוספת ידנית

## אזור אישי של מתנדב:

למתנדב מוצגות 4 כרטיסיות: פרטים אישיים, המערכת שלי, העדפות והרשמה לארגון.

## מסך פרטים אישיים:

localhost:4200/login/volunteer-area/562431243/personal-details/update

יד ליד

פרטים אישיים

שם מלא: אורה לוי

תעודת זהות: 342134265

כתובת: דוד רמז 1, צפת, ישראל

טלפון: 0521565409

כתובת מיל: tamar96115@gmail.com

סימני לאתר: 91875

תאריך לידה: 10/10/2001

אישור

כתובת: דוד רמז 1

- דוד רמז 1 אשקלון, ישראל
- דוד רמז 1 ירושלים, ישראל
- דוד רמז 1 ראשון לציון, ישראל
- דוד רמז 1 נתניה, ישראל
- דוד רמז 1 גבעתיים, ישראל

powered by Google

## מסך המערכת שלי:

localhost:4200/login/volunteer-area/562431243/volunteer-schedule

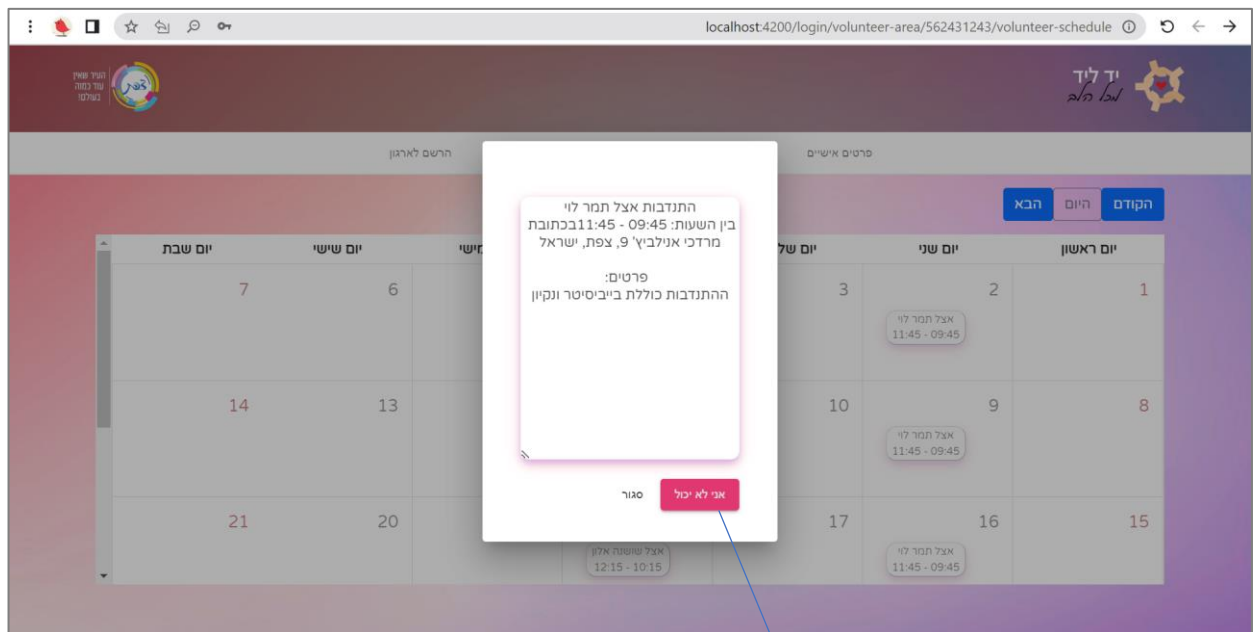
יד ליד

המערכת שלי

הקודם | היום | הבא

| יום ראשון                          | יום שני                            | יום שלישי                          | יום רביעי                             | יום חמישי | יום שישי | יום שבת |
|------------------------------------|------------------------------------|------------------------------------|---------------------------------------|-----------|----------|---------|
| 1<br>אצל תמר לוי<br>11:45 - 09:45  | 2<br>אצל תמר לוי<br>11:45 - 09:45  | 3<br>אצל תמר לוי<br>11:45 - 09:45  | 4<br>אצל שושנה אלון<br>12:15 - 10:15  | 5         | 6        | 7       |
| 8<br>אצל תמר לוי<br>11:45 - 09:45  | 9<br>אצל תמר לוי<br>11:45 - 09:45  | 10<br>אצל תמר לוי<br>11:45 - 09:45 | 11<br>אצל שושנה אלון<br>12:15 - 10:15 | 12        | 13       | 14      |
| 15<br>אצל תמר לוי<br>11:45 - 09:45 | 16<br>אצל תמר לוי<br>11:45 - 09:45 | 17<br>אצל תמר לוי<br>11:45 - 09:45 | 18<br>אצל שושנה אלון<br>12:15 - 10:15 | 19        | 20       | 21      |

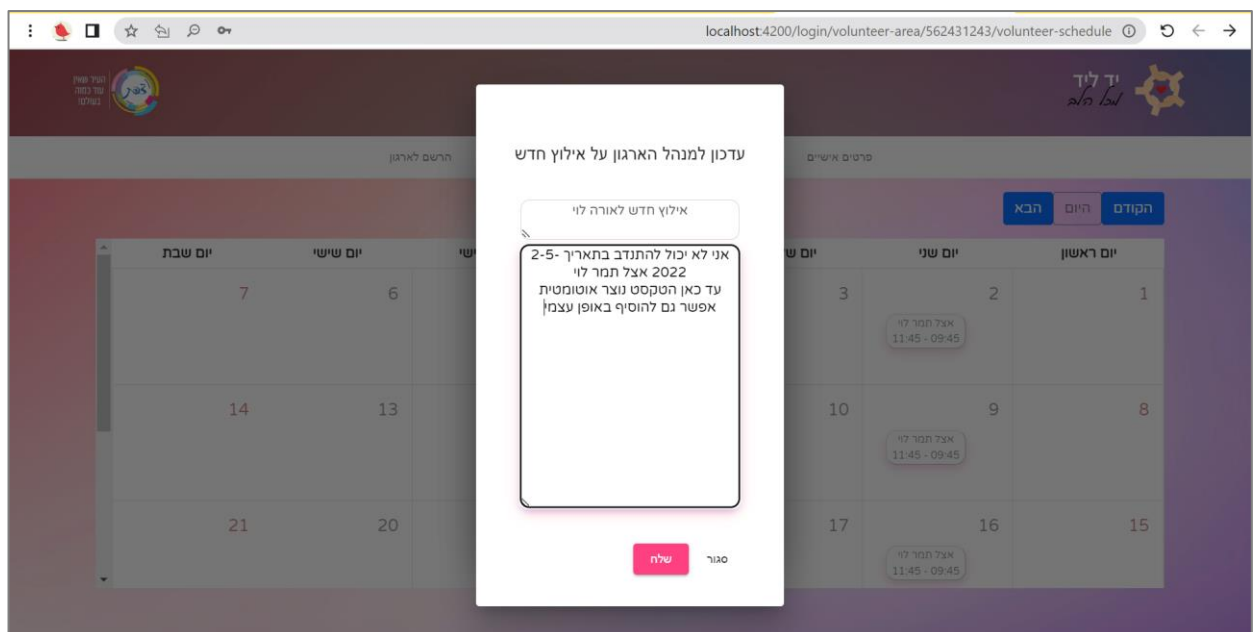
## בלחיצה על התנדבות כלשהי בלוח – יפתח חלון עם פרטי ההתנדבות ואפשרות לביטולה:



בלחיצה על "אני לא יכול" יפתח חלון עם פרטי הודעה שתישלח מיידית למנהל הארגון.

הטקסט של ההודעה נוצר באופן אוטומטי והמתנדב יכול להוסיף.

הודעה למנהל הארגון על ביטול התנדבות מסוימת:



בלחיצה על "שלח" – תישלח ההודעה למייל של מנהל הארגון, וההתנדבות ביום הספציפי שנבחר תתבטל.

## מסך הכנסת אילוצים – העדפות:

במסך זה מתבקש המתנדב לבחור את הארגון אליו הוא מכניס את העדפותיו, ואז לבחור שעות התחלה וסיום נוחות וימים מתאימים בשבוע.

עבור כל ארגון שיבחר – מוצגת לו בצד שמאל טבלה של סיכום האילוצים שלו בארגון זה עד כה, והוא יכול לבחור שורות למחיקה מהטבלה ע"י כפתור המחיקה.

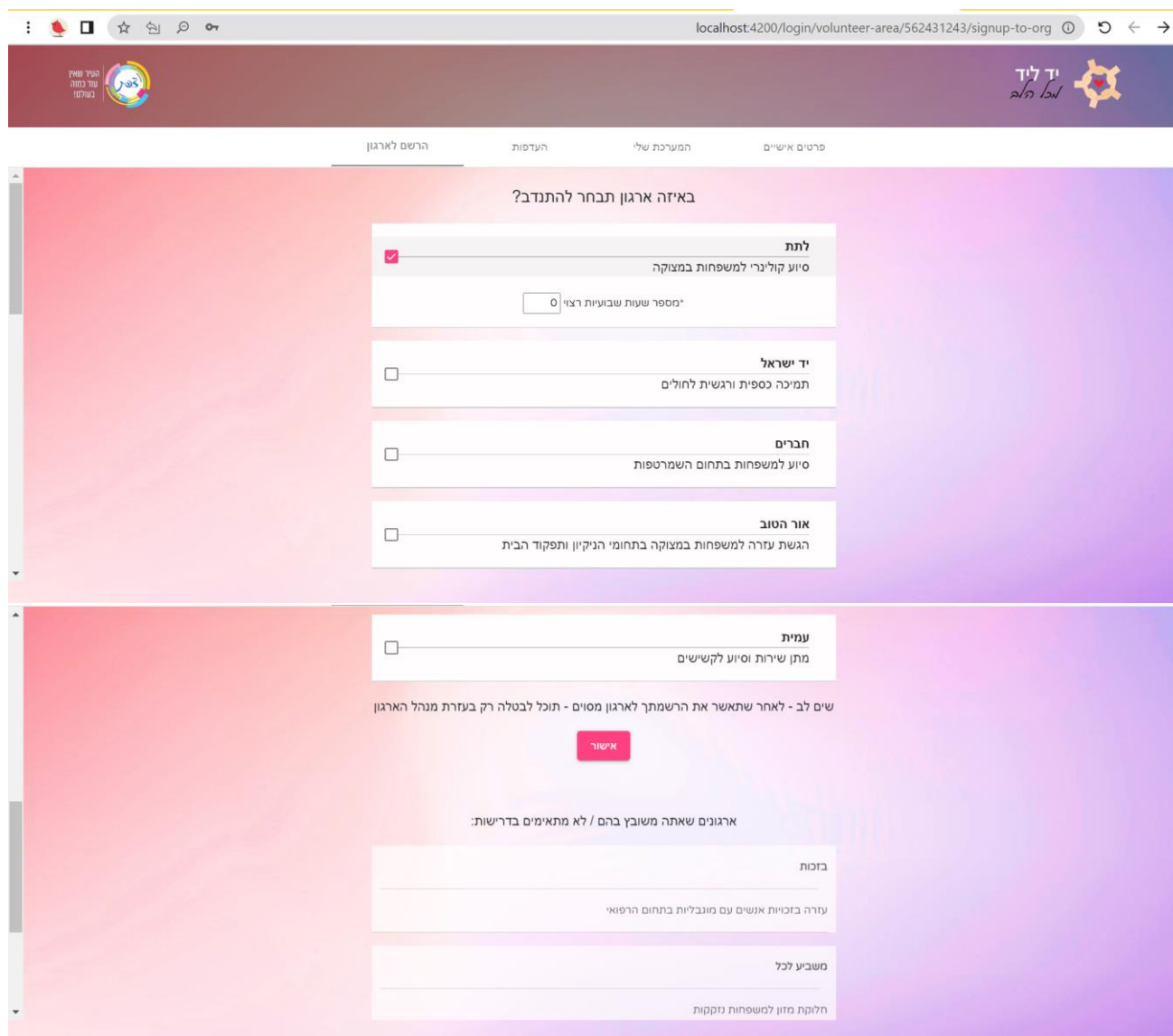
לאחר הכנסת אילוצים חדשים – המערכת בודקת האם הם מהווים קונפליקט למערכת הקיימת בארגון זה (אם ישנה), ובמידה ונמצאו קונפליקטים – מציגה הודעה למשתמש על כך והוא יכול לבחור – האם לבטל את השיבוצים שאינם מתאימים מעכשיו, או להמשיך במערכת הקיימת עד לשיבוץ חדש ע"י המנהל.

מחיקת זמן אפשרי

הוספת שורה נוספת של טווח שעות

## מסך הרשמה לארגון:

במסך זה בוחר המתנדב את הארגונים בהם הוא מעונין להתנדב ומציין את מס' השעות השבועיות שהוא רוצה לתרום. כדי להבטיח את בטיחות הנתונים – המערכת מאפשרת לו לבחור רק בארגונים הרלוונטים עבורו – ארגונים שהוא לא רשום בהם עדיין, וארגונים המתאימים לו מבחינת גיל.



localhost:4200/login/volunteer-area/562431243/signup-to-org

הרשם לארגון העדפות המערכת שלי פרטים אישיים

באיזה ארגון תבחר להתנדב?

☒ **לתת**  
סיוע קולני למשפחות במצוקה

מספר שעות שבועיות רצוי

☐ **יד ישראל**  
תמיכה כספית ורגשית לחולים

☐ **חברים**  
סיוע למשפחות בתחום השמרטפות

☐ **אור הטוב**  
הגשת עזרה למשפחות במצוקה בתחומי הניקיון ותפקוד הבית

☐ **עמית**  
מתן שירות וסיוע לקשישים

שים לב - לאחר שתאשר את הרשמתך לארגון מסוים - תוכל לבטלה רק בעזרת מנהל הארגון

**אישור**

ארגונים שאתה משובץ בהם / לא מתאימים בדרישות:

**בזכות**  
עזרה בזכויות אנשים עם מוגבלויות בתחום הרפואי

**משובע לכל**  
חלוקת מזון למשפחות נזקקות

## אזור אישי של מקבל סיוע:

אזור אישי של מקבל סיוע זהה לאזור אישי של מתנדב, מלבד מסך ההרשמה לארגון שלא קיים אצל מקבל סיוע.

## מסך פרטים אישיים:

localhost:4200/login/needy-area/987562343/needy-personal-details/update

העדפות המערכת שלי פרטים אישיים

שם מלא: יוסף לוי

תעודת זהות: 343265789

כתובת: לוחמי הגטאות 81, צפת, ישראל

טלפון: 0533123654

כתובת מייל: tamar96115@gmail.com

סימנה לאתר: 35042

אישור

כתובת: דוד רמז 1

דוד רמז 1, אשקלון, ישראל

דוד רמז 1, ירושלים, ישראל

דוד רמז 1, ראשון לציון, ישראל

דוד רמז 1, נתניה, ישראל

דוד רמז 1, גבעתיים, ישראל

powered by Google

## מסך המערכת שלי:

localhost:4200/login/needy-area/987562343/needy-schedule

העדפות המערכת שלי פרטים אישיים

הבא היום הקודם

| יום ראשון | יום שני | יום שלישי | יום רביעי | יום חמישי | יום שישי | יום שבת |
|-----------|---------|-----------|-----------|-----------|----------|---------|
| 1         | 2       | 3         | 4         | 5         | 6        | 7       |
| 8         | 9       | 10        | 11        | 12        | 13       | 14      |
| 15        | 16      | 17        | 18        | 19        | 20       | 21      |

על ידי חיים אמסלם  
14:30 - 12:30

על ידי חיים אמסלם  
14:30 - 12:30

על ידי חיים אמסלם  
14:30 - 12:30

## צפיה בפרטי התנדבות ואפשרות לביטולה:

The screenshot shows a web browser window with the URL `localhost:4200/login/needy-area/987562343/needy-schedule`. The page displays a calendar for the month of May 2022, with columns for 'יום ראשון' (Sunday), 'יום שני' (Monday), and 'יום שבת' (Saturday). Each day contains a list of volunteer shifts, each with a time slot (14:30 - 12:30) and a status (על ידי חיים אמסלם). A modal dialog is open in the center of the screen, with the text: 'התנדבות על ידי חיים אמסלם בין השעות: 14:30 - 12:30'. Below the text are two buttons: 'סגור' (Close) and 'אני לא יכול' (I can't).

## ביטול התנדבות:

The screenshot shows the same web browser window as the previous one. The modal dialog is now open, displaying the text: 'עדכון למנהל הארגון על אילוץ חדש' (Update to the organization manager about a new constraint). Below this text is a text input field containing 'אילוץ חדש ליוסף לוי' (New constraint for Yosef Levi). Below the input field is a text area containing the text: 'אני לא יכול שיתנדב אצלי בתאריך 06-05-2022' (I can't volunteer with me on 06-05-2022). Below the text area are two buttons: 'שליח' (Send) and 'סגור' (Close).

## מסך הכנסת העדפות – אילוצים:

הוספת שורה נוספת של טווח שעות

## מדריך למשתמש:

בכניסה לאתר בחר באפשרות המתאימה:

אם הנך מתנדב שלא רשום במערכת – בחר "הרשמה", הקלד ת.ז וכתובת מייל במקום המתאים, הזן את הסיסמה שנשלחה אליך, והשלם את פרטיך האישיים.

אם הנך משתמש שכבר רשום במערכת – בחר "כניסה", הקלד את ת.ז והסיסמה שלך ותגיע לאזור האישי.



### מדריך למנהל המערכ:

מוצגת לך טבלת הארגונים הקיימים במערכת. כדי לצפות ולעדכן פרטי מנהל קיים, או לשייך מנהל חדש לארגון ללא מנהל – לחץ על האייקון של מנהל בשורה המתאימה בטבלה. תוכל להכניס ארגונים חדשים ומנהלים חדשים בלחיצה על "הוסף ארגון" או "הוסף מנהל".

### מדריך למנהל ארגון:

באזור האישי תוכל לנווט בין 4 הכרטיסיות המוצגות: פרטים אישיים, המערכת שלי, ניהול החברים בארגון וניהול המתנדבים בארגון.

כדי לצפות ולערוך את פרטיך האישיים בחר "פרטים אישיים" ולאחר העריכה "שמור".

כדי לצפות ולנהל את מערכת הארגון בחר "המערכת שלי", יוצג לך לוח חודשי עם אפשרות למעבר לחודשים אחרים, ובו סיכום קצר של כל ההתנדבויות הפעילות כרגע בארגון. תוכל לצפות בפרטי התנדבות מסוימת ע"י לחיצה על התקציר שלה, וכן למחוק התנדבות ביום מסוים ע"י לחיצה על אייקון המחיקה.

תוכל לשבץ מערכת חכמה בארגון בקלות ובנוחות ע"י לחיצה על "הפעל שיבוץ מערכת", ואם ברצונך לשבץ מערכת באופן ידני – בחר "שיבוץ ידני", ובחלון שנפתח בחר את מאפייני השיבוץ החדש – מתנדב ומקבל סיוע – מתוך רשימת הרלוונטים לארגון, וכן תאריכים, שעות ויום בשבוע.

כרטיסיית ניהול החברים בארגון: מוצגת לך טבלה עם פרטי החברים בארגון, כדי לערוך או למחוק מקבל סיוע מסוים – לחץ על האייקון המתאים בטבלה.

קימות 2 אפשרויות להוספת חברים לארגון:

1. הוספה באופן ידני – לחץ על כפתור "הוספה ידנית" ומלא את הפרטים המתאימים.
2. הוספה באמצעות העלאת קובץ אקסל המכיל את פרטי המתנדבים. שם לב לקרוא לעמודות בשם משמעותי ע"מ שהמערכת תדע לנתח את הקובץ בצורה טובה. לאחר העלאת הקובץ – המתן להודעת האישור שמידעת כמה מקבלי סיוע חדשים התווספו למערכת.

כרטיסיית ניהול המתנדבים בארגון זהה לניהול חברים בארגון.

### מדריך למתנדב ולמקבל סיוע:

באזור האישי תוכל לנווט בין 4 הכרטיסיות המוצגות: פרטים אישיים, המערכת שלי, העדפות והרשמה לארגון.

כדי לצפות ולערוך את פרטיך האישיים בחר "פרטים אישיים" ולאחר העריכה "שמור".

כדי לצפות במערכת שלך בחר "המערכת שלי", יוצג לך לוח חודשי עם אפשרות למעבר לחודשים אחרים, ובו סיכום קצר של כל ההתנדבויות הפעילות שלך. תוכל לצפות בפרטי התנדבות מסוימת ע"י לחיצה על התקציר שלה, וכן למחוק

התנדבות ביום מסוים ע"י לחיצה על אייקון המחיקה. בחירה באפשרות של מחיקה תיידע במייל מיידי את מנהל הארגון הרלוונטי, לצורך כך יפתח לך חלון ובו פרטי המייל שישלח למנהל. תוכל לערוך את הטקסט שישלח אליו.

כדי לנהל את הזמנים שנחים לך להתנדבות – בחר "העדפות", ושם בחר בארגון המתאים. בצד שמאל תופיע טבלה עם סיכום ההעדפות הפעילות שלך לארגון זה, תוכל למחוק שורות ממנה על ידי לחיצה על כפתור המחיקה. כדי להגדיר העדפות חדשות – בחר יום / ימים רצויים בשבוע, ושעות התחלה וסיום. תוכל להוסיף שעות התחלה וסיום ע"י לחיצה על הכפתור המתאים.

שים לב – אם הגדרת העדפות חדשות עבור ארגון מסוים, בזמן שהמערכת בו כבר משובצת עבורך – תוצג לך הודעה מתאימה ותוכל לבחור האם לבטל את כל ההתנדבויות שמהוות קונפליקט לאילוצים החדשים, או להמשיך במערכת זו עד לשיבוץ מערכת חדשה.

### המשך מדריך למתנדב:

כדי להרשם כמתנדב בארגון מסוים בחר "הרשמה לארגון" ושם בחר מבין הארגונים המוצגים כאפשריים את הארגון הרצוי, לאחר שתסמן אותו – ציין את מספר השעות השבועיות הרצויות להתנדבות.

## ניתוח יעילות

נתקלתי בנושא היעילות פעמים רבות בפרויקט, לדוג':

- כשבניתי את מסד הנתונים – עמדו לפני 2 אפשרויות לשיוך מנהל לארגון: האם בטבלת ארגונים לשמור קוד מנהל, או בטבלת מנהלים לשמור קוד ארגון. לבסוף הגעתי למסקנה שיעיל יותר לשמור עבור כל מנהל את קוד הארגון שלו, משום שהצורה הנפוצה בפרויקט שלי לגישה כזו היא גישה לארגון דרך מנהל ופחות למנהל דרך הארגון.
- באלגוריתם השיבוץ הדעפתי להביא את כל רשימות העצמים ממחלקת ה DAL בראש המחלקה כדי לחסוך את הקריאות למסד הנתונים עבור כל פקודה רלוונטית. בכך העדפתי שיפור ביצועים על פני זיכרון.
- בבניית אלגוריתם השיבוץ – בפונקציית generaten המגרילה ערכים ראשוניים למערכת – יכלתי להגריל שעת התחלה וסיום בלי קשר להעדפות המתנדב ומקבל הסיוע, ואז ב fitness להוריד מהציון אם אכן השעות לא כלולות בשעות הנוחות לכל אחד מהם. אך רציתי לחסוך כמה שיותר בדיקות כדי ליעל את האלגוריתם. חשבתי הרבה על הדרך היעילה ביותר לממש את הפונקציה ולבסוף מימשתי אותה בצורה שחוסכת את הבדיקה של האם השעות מועדפות – הן מצד המתנדב והן מצד מקבל הסיוע.

## אבטחת מידע:

המידע של המשתמשים מוגן באמצעות סיסמאות השמורות במערכת. הכניסה היא רק באמצעות תעודת זהות וסיסמה, כך ששילוב שניהם יוצר הגנה כפולה.

## פיתוחים עתידיים:

בשלב זה – המערכת מטפלת בנושא הרישום עבור כל סוגי הארגונים, ובנושא ניהול המערכת – רק עבור ארגונים הרשומים כצרכים שיבוץ. בעתיד אשמח לשכלל יותר את האתר כך שיכיל ניהול שוטף גם עבור ארגונים שאינם צרכים שיבוץ, כגון השאלת ציוד רפואי, איחוד הצלה וכדומה – הדורשים התאמת מתנדבים להתנדבות בזמן אמת, לפי פרמטרים שיוכנסו. כמו כן – אשמח לפתח את המערכת כך שתקבל משוב מכל אחד מהמשתמשים על המערכת בה הוא שובץ – האם הוא שבע רצון ממנה, ולהסיק מסקנות לשיבוצים הבאים.

## מסקנות:

ביצוע הפרויקט הועיל לי יותר מכל מבחן. ניתן לומר כי הוא תרם לי רבות – את היכולת לבצע עבודת מחקר מעמיקה, למצוא חומרים, לעיין בהם, ללמוד את הנושא בצורה מקיפה ובאופן עצמאי, ולדעת ליישם אותו. למדתי לתמצת מחקר רב למספר שורות קוד קריא וקצר.

במהלך הפיתוח עלו שאלות רבות לגבי ביצוע טכני ותכנותי בפרויקט, היוו חלקים בקוד שנראו בלתי אפשריים, ונאלצתי למצוא להם פתרונות בדרכים יצירתיות ובאמצעות שימוש בספריות וחומרים בנושא.

היקפו הגדול של הפרויקט מול מגבלות הזמן לימד אותי לחלק את הזמן בצורה יעילה בין הדרישות, כך שההשקעה תניב פירות, נוכחתי לדעת כמה חשובה תוכנית עבודה מסודרת, אך גם כשהיא משתנה – למצוא מוצא בעז"ה.

גם כתיבת הספר תרמה רבות לביסוס הידע שצברתי במהלך הפרויקט ונתנה לי מבט מקיף ורחב עליו ועל אופן ניתוח מערכות.

לסיכום – ניתן לומר שתקופת פיתוח הפרויקט הביאה אותי לרמה גבוהה יותר וקידמה אותי רבות – הן ברמה מקצועית והן מבחינה אישית.

## ביבליוגרפיה:

