בס"ד

מגישה: תמר שם טוב

ת.ז: 325112332

ביה"ס: סמינר בית יעקב צפת

מנחה: יעל עמר

תאריך הגשה: יולי 2022

תוכן

[1.הצעת פרויקט מאושרת......................................................................................................................................................2](#_Toc103204846)

[2.תקציר / מבוא:....................................... 2](#_Toc103204847)

[2.1 הרקע לפרויקט:...................................... 2](#_Toc103204848)

[2.2 תהליך המחקר: 2](#_Toc103204849)

[2.3 סקירת ספרות: 3](#_Toc103204850)

[3.מטרות – יעדים:............................................. 3](#_Toc103204857)

[4.אתגרים:....................................................... 4](#_Toc103204858)

[5.מדדי הצלחה למערכת:.................................... 4](#_Toc103204859)

[6.רקע תיאורטי:................................................. 4](#_Toc103204860)

[7.תיאור מצב קיים:............................................ 4](#_Toc103204861)

[8.ניתוח חלופות מערכתי:.................................. 4](#_Toc103204862)

[9.תיאור החלופה הנבחרת, כולל נימוקים לבחירתה: 5](#_Toc103204863)

[10. אפיון המערכת שהוגדרה: ................................................................................................................................5](#_Toc103204864)

[10.1 ניתוח דרישות המערכת: 5](#_Toc103204865)

[10.2 מודול המערכת: 5](#_Toc103204866)

[10.3 אפיון פונקציונלי: 5](#_Toc103204867)

[10.4 ביצועים עיקריים: 6](#_Toc103204868)

[10.5 אילוצים: 6](#_Toc103204869)

[11. תיאור הארכיטקטורה: 6](#_Toc103204870)

[11.1 הארכיטקטורה של הפתרון המוצעת בפורמט של Top-Down Level Design: 6](#_Toc103204871)

[11.2 תיאור הרכיבים בפתרון: 6](#_Toc103204872)

[11.3 ארכיטקטורת רשת אם יש: 6](#_Toc103204873)

[11.4 תיאור פרוטוקולי התקשורת: 6](#_Toc103204874)

[11.5 שרת – לקוח: 6](#_Toc103204875)

[11.6 תיאור הצפנות אם יש: 7](#_Toc103204876)

[12. ניתוח ותרשים use-case של המערכת המוצעת: 7](#_Toc103204877)

[12.1 תיאור הUC העיקריים של המערכת 7](#_Toc103204878)

[12.2 הצגת מקרה עבור כל הפונקציות העיקריות בפרויקט: 8](#_Toc103204879)

[12.3 מבני נתונים בהם השתמשתי: 8](#_Toc103204880)

[12.4 עץ מודולים 8](#_Toc103204881)

[12.5 תרשים מחלקות 8](#_Toc103204882)

[12.6 תיאור המחלקות המוצעות 8](#_Toc103204883)

[13. תיאור התוכנה 8](#_Toc103204884)

[14.אלגוריתמים מרכזיים....................................... 9](#_Toc103204885)

[15. תיאור מסד נתונים 9](#_Toc103204886)

[15.1 תרשים טבלאות + קשרי גומלין................... 9](#_Toc103204887)

[15.2 תיאור הטבלאות ב data base: 9](#_Toc103204888)

[16.תיאור מסכים:.............................................. 11](#_Toc103204889)

# הצעת פרויקט מאושרת

# 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.

נהניתי 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 שינוים 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.

## 2.2 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 הפריטים הראשונה, מדרגים כל פריט על מנת למצוא את הפריטים הטובים ביותר. לאחר מכן, עורכים מיזוג בין פריטים 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, הפריטים 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 שינוים 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 צוין 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 האפשרות 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.

## 2.3 סקירת ספרות:

האתרים העיקריים בהם נעזרתי במהלך החקירה וכתיבת 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://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 חתרתי 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.
* בנית מסד נתונים מתוכנן היטב השומר את הנתונים בצורה מסודרת ויעילה.
* אחזור מהיר ונוח של הנתונים.
* ממשק קל ונוח לתפעול, וכן מעוצב וידידותי 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 מערכתי:

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 if וelse 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.

1. 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 פולינומינאליות, 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:

## 10.1 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

## 10.2 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

## 10.3 אפיון פונקציונלי:

**RandomVolunteer –** הגרלת מתנדב אפשרי עבור מקבל סיוע במשבצת זמן מסוימת.

**RandomVolunteeringDuration** – הגרלת מספר שעות הגיוני להתנדבות מסוימת.

**RandomStartHour** – הגרלת שעת התנדבות אקראית מתוך השעות הנתונות, שתתאים למשך זמן ההתנדבות.

**TimeTableChromosome** – יצירת רשימה של משבצות זמן שירכיבו שיבוץ למשך שבוע ימים.

**Generate** - אחראית על יצירת לוח זמנים אקראי.

**Clone** – משכפלת את השיבוץ של לוח זמנים (לצורך שילוב עם לוח זמנים נוסף).

**Mutate** – בוחרת משבצת זמן אקראית, ומשנה בה את היום ואת השעות ליום ושעות אקראיים.

**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.

**Gentic** – יצירת לוח זמנים ע"י שימוש באלגוריתם הגנטי.

**ReadFromExcel** – מזמנת את אלגוריתם הקריאה מ excel.

## 10.4 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 בארגונו – 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.

## 10.5 אילוצים:

## 

## 11. תיאור הארכיטקטורה:

## 11.1 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 HTML ,SCSS ו– Type script בטכנולוגית Angular.

תיאור צד השרת:

צד השרת מחולק כמקובל לשכבות: שכבת הDAL, 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 לתפעולו. פעולות ההתקשרות עם בסיס הנתונים נעשו בטכנולוגית 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 לטיפוס 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.

## 11.2 תיאור הרכיבים בפתרון:

DB -

SQL

server

DAL

Entity

Framework

MODELS

BL

Layers

Controllers

Web API

Client

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

## 11.3 ארכיטקטורת רשת אם יש:

לא רלוונטי

## 11.4 תיאור פרוטוקולי התקשורת:

HTTPS

## 11.5 שרת – לקוח:

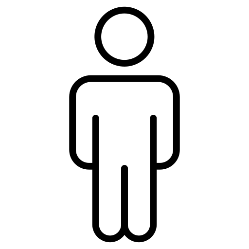
צד השרת נכתב בטכנולוגית Api Webובשפת C# .

צד הלקוח נכתב בשפות HTML ,SCSS ו– Type script בטכנולוגית Angular.

## 11.6 תיאור הצפנות אם יש:

לא רלוונטי.

## ניתוח ותרשים use-case של המערכת המוצעת:



**מתנדב**

צפייה בפרטים אישיים

הרשמה לארגון

הכנסת אילוצים חדשים

התחברות לאתר

צפיה במערכת שלי

צפייה באילוצים שלי

מחיקת אילוצים

עדכון פרטים אישיים

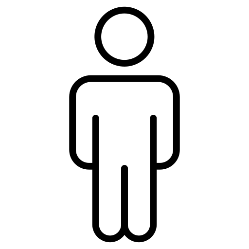
רישום לאתר

קבלת סיסמה למייל

מילוי פרטים אישים

צפייה בפרטי התנדבות

ביטול התנדבות מסוימת



**מקבל סיוע**

צפייה בפרטים אישיים

הכנסת אילוצים חדשים

התחברות לאתר

צפיה במערכת שלי

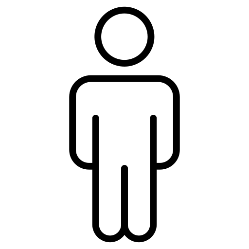
צפייה באילוצים שלי

מחיקת אילוצים

עדכון פרטים אישיים

צפייה בפרטי התנדבות

ביטול התנדבות מסוימת



**מנהל ארגון**

צפייה בפרטים אישיים

הוספת מתנדב - ידנית

התחברות לאתר

צפייה במערכת הארגון

צפייה במתנדבי הארגון

הוספת מתנדבים ע"י העלאת קובץ excel

עדכון פרטים אישיים

צפייה בפרטי התנדבות

שיבוץ ידני במערכת

מחיקת מתנדב

ביטול התנדבות מסוימת

הורדת קובץ excel של הסיסמאות שהוגדרו

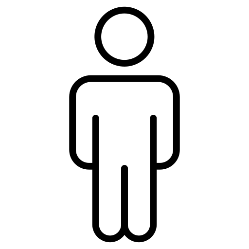
הוספת מקבל סיוע - ידנית

צפיה במקבלי סיוע בארגון

הוספת מקבלי סיוע ע"י העלאת קובץ excel

מחיקת מקבל סיוע

הורדת קובץ excel של הסיסמאות שהוגדרו



**מנהל מערך ההתנדבות**

צפייה בפרטים אישיים

צפייה בפרטי מנהל ארגון

התחברות לאתר

צפיה בארגונים הקיימים

הוספת ארגון חדש

עדכון פרטים אישיים

צפייה בפרטי הארגון

הוספת מנהל חדש

# 12.1 רשימת use-cases

1. כניסה לאתר כמנהל כללי
   1. צפייה ועדכון פרטים אישיים
   2. הכנסת פרטי ארגון חדש
   3. הכנסת פרטי מנהל ארגון חדש
2. רישום לאתר כמתנדב חדש
3. כניסה לאתר כמתנדב קיים
   1. צפייה ועדכון פרטים אישיים
   2. רישום כמתנדב לארגון
   3. צפייה והוספת אילוצים
   4. צפייה במערכת הרלוונטית
   5. ביטול התנדבות ביום מסוים – מעדכן את המנהל במייל אוטומטי
4. כניסה לאתר כמקבל סיוע קיים
   1. צפייה ועדכון פרטים אישיים
   2. צפייה והוספת אילוצים
   3. צפייה במערכת הרלוונטית
   4. ביטול התנדבות ביום מסוים – מעדכן את המנהל במייל אוטומטי
5. כניסה לאתר כמנהל ארגון
   1. הכנסת פרטי מקבלי הסיוע בארגון – ידנית או באמצעות העלאת קובץ excel
   2. הכנסת פרטי מתנדבים בארגון – ידנית או באמצעות העלאת קובץ excel
   3. צפייה במערכת הארגון
   4. הפעלת שיבוץ מערכת בארגון
   5. שיבוץ מערכת ידנית

# 12.2 תיאור ה UC העיקריים של המערכת

# הצגת UC עבור כל הפונקציות העיקריות בפרויקט

# 12.4 מבני נתונים בהם השתמשתי

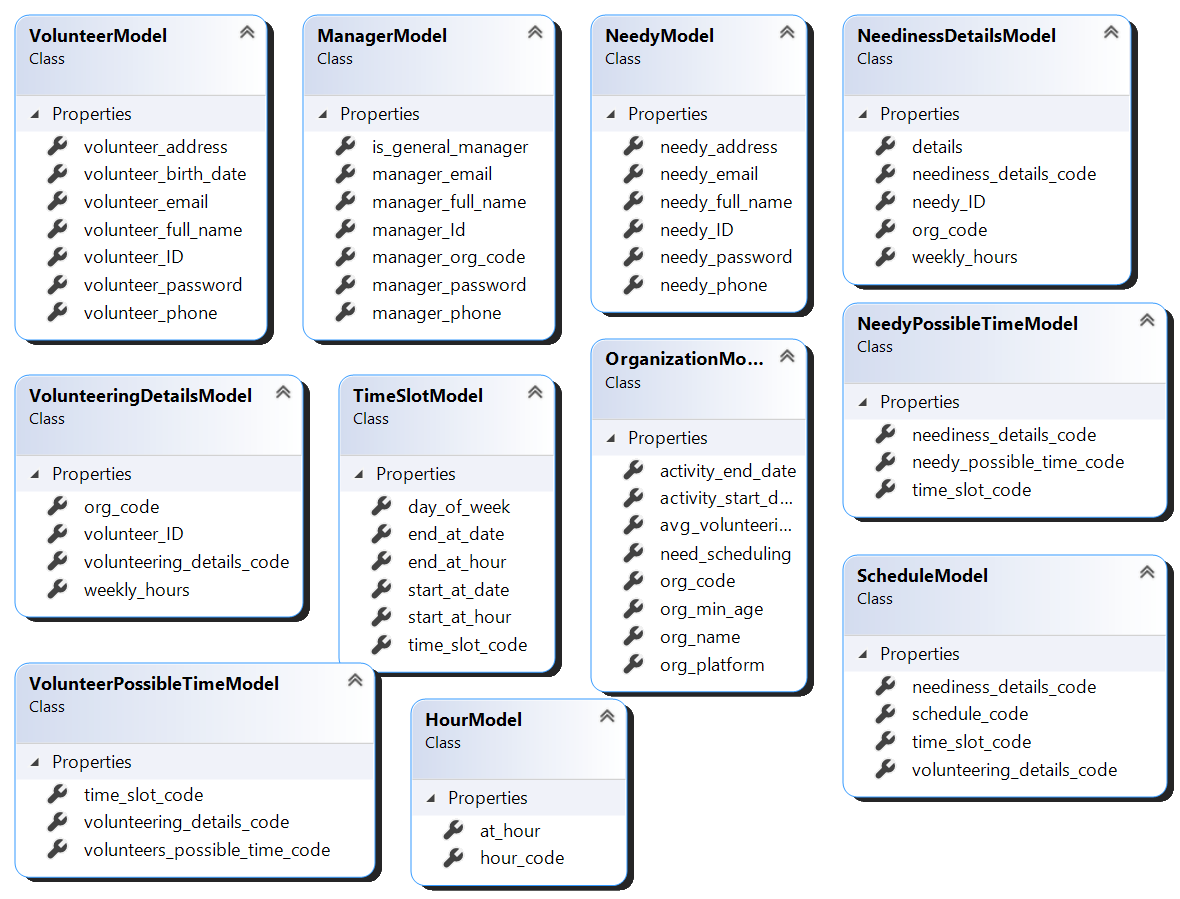
* רשימת עצמים עבור כל מחלקה
* מערכי עזר בפונקציות
* מילון ערכי שמות עמודות של קובץ לצורך קריאה מקובץ excel של מתנדבים / מקבלי סיוע חדשים שהועלו ע"י מנהל ארגון.
* מטריצה להחזרת סיסמאות אקראיות למתנדבים / מקבלי סיוע חדשים שהוכנסו ע"י מנהל ארגון בהעלאת קובץ excel. המטריצה מומרת ב angular לקובץ excel זמין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

# הקשרים בין היחידות השונות

# 12.6 עץ מודולים

# 12.7 תרשים מחלקות:

מחלקת DAL:

מחלקת MODELS:

## 12.8 תיאור המחלקות המוצעות

## תיאור התוכנה

# סביבת עבודה

Visual studio

Visual studio code

# שפות תכנות

צד שרת: טכנולוגיית web api, שפת c#

סביבת עבודה: Visual Studio ו Visual Studio Code שפות תכנות: צד השרת נכתב בטכנולוגית Api Web ובשפת #C. צד הלקוח נכתב בשפות html ,Css ו– script Type בטכנולוגית Angular.

## אלגוריתמים מרכזיים

## 15. תיאור מסד נתונים

* 1. תמונה שמכילה שולחן

     התיאור נוצר באופן אוטומטיתרשים טבלאות + קשרי גומלין

## 15.2 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 | V |
|  | manager\_full\_name | String | שם מנהל | V |
|  | Volunteer\_email | String | כתובת מייל של מנהל | V |
|  | Volunteer\_phone | string | מס' טלפון של מנהל |  |
| FK | Manager\_org\_code | int | קוד הארגון אותו הוא מנהל | V |
|  | Manager\_password | String | סיסמה לאתר | V |
|  | is\_general\_manager | Bool | מציין האם המנהל הוא מנהל כללי (ממנהלי יד ליד) או רק מנהל ארגון (לצורך הרשאות הגישה | V |

* volunteering\_details

מכילה פרטי התנדבויות עבור כל מתנדב (מקשרת בין מתנדב לבין ארגון)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **מפתח** | **שם שדה** | **טיפוס נתונים** | **תיאור** | **שדה חובה?** |
| PK | Volunteering\_details\_code | Int, מספור רץ | קוד פרטי התנדבות, מספור רץ | V |
| FK | Volunteer\_ID | String | ת.ז מתנדב | V |
| FK | Org\_code | int | קוד ארגון | V |
|  | weekly\_hours | double | מס' שעות התנדבות רצויות בשבוע | V |

* neediness\_details

מכילה פרטי סיוע עבור מקבל עזרה (מקשרת בין מקבל עזרה לבין ארגון)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **מפתח** | **שם שדה** | **טיפוס נתונים** | **תיאור** | **שדה חובה?** |
| PK | neediness\_details\_code | Int, מספור רץ | קוד פרטי סיוע, מספור רץ | V |
| FK | needy\_ID | String | ת.ז מקבל עזרה | V |
| FK | Org\_code | int | קוד ארגון | V |
|  | weekly\_hours | double | מס' שעות עזרה שזכאי להן בשבוע | V |
|  | Details | String | פרטים טכניים אודות הסיוע הנדרש |  |

* 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 |
|  | Start\_at\_hour | timespan | שעת התחלה | V |
|  | End\_at\_hour | timespan | שעת סיום | 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:

תרשים מסכים:

מסך כניסה

התחברות

עריכת פרטים אישיים

הרשמה כמתנדב

(הרשמה באופן עצמאי אפשרית רק כמתנדב. מנהל ארגון יכול להרשם רק ע"י המנהל הכללי, ומקבל עזרה יכול להרשם רק ע"י מנהל

הרשמה כמתנדב בארגון

צפייה במערכת ועדכון שינויים

הוספת ועדכון אילוצים

צפייה במערכת ועדכון שינויים

הוספת ועדכון אילוצים

ניהול מערכת

ניהול מתנדבים בארגון

ניהול מקבלי סיוע

עריכת פרטים אישיים

עריכת פרטים אישיים

עריכת פרטים אישיים

רישום ארגון חדש

רישום ארגון חדש

עריכת פרטים אישיים

אזור אישי

מנהל יד ליד

אזור אישי

מתנדב

אזור אישי

מקבל עזרה

אזור אישי

מנהל ארגון