תוכן

[1. הצעת פרויקט 3](#_Toc102417030)

[2. מבוא / תקציר 3](#_Toc102417031)

[2.1. הרקע לפרויקט 3](#_Toc102417032)

[2.2 תהליך המחקר 3](#_Toc102417033)

[2.3 סקירת ספרות 3](#_Toc102417034)

[3. מטרות ויעדים 4](#_Toc102417035)

[4. אתגרים 4](#_Toc102417036)

[5. מדדי הצלחה 4](#_Toc102417037)

[6. תיאור המצב הקיים 4](#_Toc102417038)

[7. רקע תאורטי 4](#_Toc102417039)

[8. ניתוח חלופות מערכתי 4](#_Toc102417040)

[9. תיאור החלופה הנבחרת והנימוקים לבחירה 4](#_Toc102417041)

[10. אפיון המעכת 4](#_Toc102417042)

[10.1. ניתוח דרישות המערכת 4](#_Toc102417043)

[10.2. מודול המערכת 4](#_Toc102417044)

[10.3. אפיון פונקציונאלי 4](#_Toc102417045)

[10.4. ביצועים עיקריים 4](#_Toc102417046)

[10.5. אילוצים 4](#_Toc102417047)

[11. תיאור הארכיטקטורה 4](#_Toc102417048)

[11.1. הארכיטקטורה של הפיתרון המוצע בפורמט של Design level Down-Top 4](#_Toc102417049)

[11.2. תיאור הרכיבים בפיתרון 4](#_Toc102417050)

[11.3. ארכיטקטורת רשת (לא רלוונטי ) 4](#_Toc102417051)

[11.4. תיאור פרוטוקולי התקשורת (HTTP) 4](#_Toc102417052)

[11.5. שרת – לקוח 4](#_Toc102417053)

[11.6. תיאור הצפנות (לא רלוונטי) 4](#_Toc102417054)

[12. ניתוח ותרשים use case של המערכת המוצעת 4](#_Toc102417055)

[12.1. רשימת use case 4](#_Toc102417056)

[12.2. תיאור ה-use case העיקריים של המערכת 4](#_Toc102417057)

[12.3. מבני נתונים בהם משתמשים בפרויקט 4](#_Toc102417058)

[12.4. תרשים מחלקות 4](#_Toc102417059)

[12.5. תיאור המחלקות 4](#_Toc102417060)

[13. תיאור התוכנה 4](#_Toc102417061)

[14. אלגוריתמים מרכזיים 4](#_Toc102417062)

[14.1. חלק מהאלגוריתם... 4](#_Toc102417063)

[14.2. חלק אחר מהאלגוריתם... 5](#_Toc102417064)

[14.3. החלק העיקרי באלגוריתם 5](#_Toc102417065)

[15. קוד האלגוריתם 5](#_Toc102417066)

[16. תיאור מסד הנתונים 5](#_Toc102417067)

[16.1. פירוט הטבלאות ב- Data Base 5](#_Toc102417068)

[17. מדריך למשתמש 5](#_Toc102417069)

[17.1. תיאור המסכים 5](#_Toc102417070)

[17.2. מדריך למשתמש 5](#_Toc102417071)

[17.3. צילומי מסכים 5](#_Toc102417072)

[18. בדיקות והערכה 5](#_Toc102417073)

[19. ניתוח יעילות 5](#_Toc102417074)

[20. אבטחת מידע 5](#_Toc102417075)

[21. מסקנות 5](#_Toc102417076)

[22. פיתוח עתידי 5](#_Toc102417077)

[23. ביבליוגרפיה 5](#_Toc102417078)

[24. מטרות ויעדים 5](#_Toc102417079)

**1. הצעת פרויקט:**

**1.2 מבוא / תקציר:**

**רעיונות שהיו לי :**

חיפשתי פרויקט עם דרכים ומסלולים ,אך היו דברים שהם פחות מוצלחים כמו ניווט של מכוניות בכביש, ניווט בבית קברות ,ולסיום חשבתי על מערכת שתעזור לאנשים לבחור בגדים על פי סגנון מסוים שהם אוהבים אבל זה היה פחות פרקטי, ולבסוף עלה לי רעיון של וויז בקניון רננים ברעננה .

**איך בחרתי את הרעיון שלי :**

חשבתי המון על משהו שיכול להיות יעיל באמת, ולתרום לסביבה ,ונסתי לחשוב ממש לעומק מה לא טוב בהתנהלות מסוימת בחיי היום יום ואיך מאידך ניתן לתקן היו לי המון רעיונות בעקבות המחשבה הנ"ל כמו :פינוי מתואם של זבל במקומות מגורים צפופים , למקם תחנות משטרה ברחובות נטושים ובקרבת פאבים ומועדונים , אבל לא ממש התחברתי לנושאים והרגשתי שרוב הדברים (הבעיות ) לא הכי גרועים ודי יש שליטה על המצב ,לא ברמה מקסימאלית אבל לגמרי בסדר, תוך כדי חשיבות ומציאת רעיון טוב הסתובבתי בקניונים ואחזתי שהבעיה ממש טובה ונוגעת להמון אנשים שבאמת צריך ניווט בקניון גדול ושמחתי שסוף סוף קיימת בעיה .

**למה אני חושבת שהוא טוב בתור רעיון לפרויקט גמר?**

אני חושבת שלחסוך בזמן במקסימום הנאה וסיפוק זה דבר שכל אחד רוצה ,וזה מתבטא בכך שנותן מענה לאנשים שרוצים משהו מסוים או חניות מסוימות או אחד שהזמן שלו יחסית יקר ואין לו כוח להסתובב ,למרות שקניון זה מקום בילוי וזה פחות מתוכנן ומוגבל אף אחד לא מעוניין להסתבך ולחזור על עצמו אז פה הפרויקט שלי נכנס לסיטואציה ופותר את הבעיה-לא במאה אחוז כי אין מושלם אבל זה רוב אחוז ורוב פתרון 😊

**למה אני חושבת שיש בו תועלת?**

התועלת היא מועילה מאוד אין בזמן חיפוש והן בהתמקדות במה לחפש ,ובנוסף קיימת אפשרות לבחור את המסלול בשלושה אופציות :

1.מסלול לחנות מסוימת(יחידה).

2.מסלול לפי קטגוריות (רבות)/לפי חניות(רבות).

3.מסלול חכם - משכלל לפי סטטוס האם המשתמש מעוניין החניות עם מבצעים בתחילה וכן מאפשר לשלב לפי הסטטוס חניות מועדפות בתחילה .

**למה מומלץ לפתח את מה שאת מפתחת?**

מומלץ מאוד לפתח את הרעיון שלי משום שזה מאוד נפוץ בכלל מרכזי הקנייה הגדולים ונותן אופציה למקסימום הספק ותועלת במינימום זמן דבר שכל אחד רוצה (במיוחד ישראלים שאין סבלנות).

**מה קורה כיום – לפני הפיתוח?**

כיום לפני הפיתוח קיים בזבוז מטורף של זמן וסיבובים מיותרים , ואנשים מאבדים את הסבלנות ואת עצמם בדרך ..

**מה יהיה לאחר שתפתחי?**

לאחר הפיתוח הנ"ל יחסך המון זמן מיותר וחווית הקניה תהיה מהנה ולפי סטטוס- מיועד לכל משתמש שבוחר האם הוא מעונין בחניות מסוימות שהוא אוהב ,והאם יש עניין במבצעים בתחילת המסלול, וכך מצמצם האופן משמעותי את הבעיה ונותן מענה לכל אדם ונותן חוויית קניה מושלמת .

**איזה אלגוריתם עזר לי בפיתוח הרעיון?**

האלגוריתם שעזר לי לפתח את הפרויקט הוא הדיאקסטרה חמדני .

**עוד על האפליקציה :**

השם של האפליקציה הוא speed Mall-.

לאחר עבודה ממושכת וחשיבה עמוקה ועבודה על האלגוריתם

והסתבכות במשקל הקשתות (רציתי שזה יצא הכי מדויק ולכן הייתה לי המון עבודה ) -יצאה האפליקציה המדהימה הזאת שבאמת נותנת מענה ומהווה פתרון על ,ובקיצור זה היה שווה את זה כי זה ממש אתגר אותי וחידד את החשיבה -מבחירת הרעיון ועד לסיום פיתוח האפליקציה שתורמת לסביבה כולה.

**2.2 תהליך המחקר :**

הדבר הראשון שעשיתי חקרתי המון על הנושא-קראתי מלא על האלגוריתם פתרתי דוגמאות וריאתי קודים דומים בנושא ניווט ,חיפשתי מפות של קניונים גדולים וכו' .

ניסיתי המון בכיוון של חיפוש מיידע וחומרים באינטרנט יותר מיידע תאורתי יחד עם קוד לדוג'. ולבסוף מצאתי מפה שתתאים מיישום שבה הקניון ממש גדול ויהווה תחילת בעיה ממש כפי שרציתי, נעזרתי במנהלת שיווק של הקניון ששמה אורית היא נתנה לי כיווני חשיבה נוספים ,תוך כדי הבירורים מאיפה אני מתחילה התייעצתי עם המורות לפרויקט והם כיוונו אותי מאוד. בגלל המפה שאותה מצאתי – זאת מפה של קניון רננים ברעננה וכעיקרון הייתי צריכה ממש לעבוד עי'' ומשקלתי את כל המרחקים כי זה הבסיס . עבדתי המון על הדאטה בייס שיהיה אמיתי ונכון ,האלגוריתם כעיקרון עובד מצומת מקור ליעד, וכן מצומת מקור ליעד מתוך אחת ליותר משני צמתים , ועכשיו פה נפלתי לבאג ולמה ?! משום שבבסיס אני מקבלת רשימת חניות -וממנה אני אמורה לחשב את המסלול הקצר ביותר , כי כביכול מהמון צמתים להמון צמתים ואת זה האלגוריתם לבדו אינו מבצעת ולכן על ידי חשיבה עמוקה בניסיון בלחפש פתרון נמצאה דרך מתוחכמת , שבאפשרותה לעבור בשני לולאות ובקוד דייאסטרה (פירוט בהמשך ..).

ובחרתי דווקא דייאסטרה משום שהוא הכי מתאים למציאת מסלול קצר.

משום שהסוכן הנוסע זו סיבוכיות מטורפת !n.

וגם בלמן פורד לא מתאים בגלל שהוא כולל משקלות שלילים ובמרחקים אי אפשר משקל שלילי.

ולכן המסקנה היא דייאסטרה הכי מתאים 😊

**3.2 סקירה ספרותית :**

* https://stackoverflow.com
* bootstrap, css
* dijkstras
* C#, SQL, angular

1. **מטרות ויעדים :**

**מטרות :**

על שימוש באפליקציה שלי חוסכים המון ריצות וסיבובים מיותרים, ובנוסף זה חוסך גם כסף כי אתה יותר ממוקד בקניה שלך בלי להתפזר , והנוחות היא מעל ומעבר שמאשפרת הגדרות ואופציות למסלול חכם (פורט מקודם).

וכן המטרה המרכזית היא לאפשר מענה לפתרון על של מינימום זמן במקסימום עשייה.

**היעדים :**

המשתמש בוחר באיזה אופציה להשתמש (אחד מהשלושה פורט מקודם)ועל פי בחירתו נשלחות החניות המבוקשות מהSERVIS לAPI ומשם לקונטרולר המתאים ולבסוף

מה היעדים שאליהם שאפת להגיע? (המערכת תעשה, המשתמש יעשה, החולים יוכלו, הפרטים ישמרו)

# אתגרים

(מקומות שבהן הסתכבת, היית צריכה להשקיע מחשבה, להתייעץ, לשנות כיוון.)

במהלך הפרויקט נתקלתי באתגרים....  
במהלך עבודתי נהניתי לראות שהדברים אינם פשוטים כמו שנראו בתחילה. אפרט כמה נקודות....

# מדדי הצלחה

(האפליקציה שלי הצליחה אם...... 1, 2, 3  
השיבוץ אכן עובד / יש זיהוי של לפחות 70 אחוז מהמקרים / המחשב מזהה את האותיות / כניסה לבעלי הרשאה בלבד)

# תיאור המצב הקיים

כיום המצב הוא.....

לאחר בירור אצל אנשים שונים....

אחרי חיפוש של תוכנות המספקות את השירות הנ''ל גיליתי ש...

# רקע תאורטי

(חלק חשוב מאד!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! פירוט של האלגוריתם)

האלגוריתם המרכזי בפרוייקט הוא...... האלגוריתם שמאפשר את ביצוע ה..... הוא  
האלגוריתם שבו השתמשתי לצורך ה..... הוא. אפרט את האלגוריתם העיקרי בפרויקט.

1. מה האלגוריתם (החלק הכי מורכב) העיקרי בפרויקט? (3 שורות)
2. תזכירי, מה הייתה הבעיה שהאלגוריתם פותר? (6 שורות)
3. איזה סוג של אלגוריתם הכי מתאים לפירוט הבעיה שתיארת?  
   אל תתנו שם של אלגוריתם אלא תסבירו איזה סוג של אלגוריתם צריך.

# ניתוח חלופות מערכתי

1. אפשרויות שונות לפתרון הבעיה (לדוגמא, שיבוץ – באופן ידני, לנסות את כל האפשרויות, להמציא אלגוריתם, לבחור אלגוריתם הונגרי, לבחור אלגוריתם גנטי)
2. מה בחרת? (משפט 1)

# תיאור החלופה הנבחרת והנימוקים לבחירה

1. שוב, מה המערכת שפתחת?
2. מה עם הרעיונות האחרים שהצעת לגבי המערכת שלך?
3. לגבי האלגוריתם שנבחר. כיצד הוא עובד? (מהם הם השלבים שלו, החלקים שלו, ממה מורכב?) (תוכלי לתת דוגמאות והמחשה ע''י תמונות)

# אפיון המערכת

**סביבת פיתוח :**

חומרה: מעבד RAM 32GB i7 עמדת פיתוח: מחשב Lenovo מערכת ההפעל ה: 10 Windows

שפות תוכנה: #C , תוך שימוש בטכנולוגי ת WebApi, אנגולר .

כלי תוכנה לפיתוח המערכת: .Microsoft Visual Studio2019, vs code:

מסד נתונים:.SQL Server עמדת משתמש מינימאלי ת :

* + חומרה: מעבדRAM 4GB i5 .

o מערכת ההפעל ה: Windows 7ומעלה.

* + חיבור לרש ת: נדרש .
  + . chrome :תוכנות

## ניתוח דרישות המערכת

**דרישות בהן המערכת צריכה לעמוד:**

* + כתיבה בסטנדרטים מקצועיים.
  + מחשוב השרות ללקוח.
  + כתיבת הקוד בסיבוכיות היעילה ביותר.
  + ממשק נוח וידידותי למשתמש.
  + תגובה מהירה ככל שניתן למשתמש.

## מודול המערכת

* העלאת קובץ Excel המכיל את רשימת התלמידים לשיבוץ עם פרטים טכני ים עליהם.
* קריאת הקובץ שהועלה והזנת הנתונים בדאטה בייס, וכן הצגתם ללקוח.
* הזנת נתונים נוספים לתלמידים ע"י המשתמש.
* קריאה למערכת לביצוע שיבוץ.
* מציאת השיבוץ האופטימלי ככל שניתן ע"י האלגוריתם.
* הזנת פרטי השיבוץ בדאטה ביי ס
* הצגת פרטי השיבוץ ללקוח.
* שליחת מייל ללקוח עם השיבוץ הסופי

## אפיון פונקציונאלי

(פירוט פונקציות עיקריות ותפקידן)

MyFunc1 – הפונקציה מבצעת....

MyFunc2 – הפונקציה מסננת את...

## ביצועים עיקריים

המנהל יכול להוסיף....

המשתמש מכניס את פרטיו ומקבל רשימת....

למשתמש מוצגת מפה של....

למשתמש נשלחת הודעה עם....

## אילוצים

המערכת יכולה להכיל עד 300 חנויות

המערכת מתבססת על קובץ EXCELL לקבלת המידע על

המערכת פועלת ע''פ נתונםי ... ואינה יכולה לנתח את ה... בלעדיהם.

# תיאור הארכיטקטורה

## הארכיטקטורה של הפיתרון המוצע בפורמט של Design level Down-Top

צד השרת - server side פותח במודל 3 השכבות ומתחלק ל-4 פרויקטי ם

החלוקה לשכבות 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.

API – שכבת ה Controller – חיבור בין צד השרת והלקוח.

BL – הלוגיקה של המערכת.

DAL – מכיל את הפונקציונאליות הנדרשת לכל התקשורת עם הData Base.

Models – מכילה מחלקות המתארות את הנתונים ובמבנה זה מעבירים את הנתונים בין השכבות.

מטרת שכבה זו היא למנוע תלות של שכבת הBL במבנה בסיס הנתונים. שכבת הBL מכילה פונקציות המרה מטיפוס הנתונים של בסיס הנתונים לטיפוס הנתונים של שכבת הModels ולהיפך, וכך מיוצגים הנתונים בכל הפרויקט.

## תיאור הרכיבים בפיתרון

הפרויקט מחולק ל-2 חלקים:

* + - צד שרת - הנכתב בשפת #C ובטכנולוגיית WebApi.
    - צד לקוח - נכתב בשפת Angular ובטכנולוגיית Html, TypeScript.

בחרתי לכתוב צד לקוח ב - אנגולר שהינה שפה מתקדמת ועדכנית בעלת מאפייניAngular8 חדשניים ופונקציונאלית ביות ר.

אנגולר הינה סביבת עבודה שפותחה על ידי גוגל. מאפשרת לפתח אפליקציותFramework אינטרנט בקלות ומהירות. במקור היא באה לתת מענה לבניי תApplications Page Single בצורה מושלמת ומהירה. מהיתרונות הבולטים והעיקריים של אנגולר אפשר למנות: חיסכון במשאבים, מהירות ביצוע, קוד קצר יותר, רוב העבודה מתבצעת בצד הלקוח ופחות בשרת ויכולת התמודדות טובה )סינון מהיר ופשוט לביצוע( של תוכן המתקבל מהשרת לפי מספר רב של פרמטרים.

צד שרת בחרתי לכתוב בC# .C# היא שפת תכנות עילית מרוב ת-פרדיגמות, מונחית עצמים בעיקרה המשלבת רעיונו ת כמו טיפוסיות חזקה, אימפרטיביות, הצהרתיו ת, פונקציונאליו ת פרוצדוראליות וגנריות .

C# היא שפה מעניינ ת, נוחה ומלאה פונקציונאליות למתכנת. שימוש בשפה זו נפוץ כיום , וכתוצאה מכך, ניתן היה למצוא בה קודים שונים שנדרשו לפיתו ח.

בנוסף ,בחרתי להשתמש ב - EntityFramework טכנולוגית עבודה מתקדמת של מיקרוסופט.

הEntityFramework מאפשר לטעון את הנתונים מהDB -ולעשות להם השמה בצורה ישירה ואוטומטית לתוך אובייקטים בקוד הממפים את מאגר הנתונים בצורה מידי ת.

.Sql Server שנכתב בשפת DataBaseקורא נתונים מ ה EntityFrameworkה

למסד הנתונים של ה-SQL Server יש כלים נרחבים לגיבוי כל המידע של המערכת ,כולל מערכת ההפעלה, חשבונות המשתמשים והרשאותיהם, הגדרות ההתקנים, תוכניות וכן של שאר הרכיבים המסופקים עם השרת ואובייקטי המשתמש.

*דוגמא לזרימת מידע במערכת*

שליפת כל התלמידי ם

ברצוננו לקבל את כל התלמידים של משתמש מסוים מ הDB ולכן יתבצעו השלבים הנ"ל:

* המשתמש יחפוץ לראות את כל התלמידים שלו, הוא ילחץ על כפתור מסוים בתצוגה) html( ובקשתו תפנה לTypeScript.
* .servicesאשר תפנה ל GetStudentByiD תתבצע קריאה לפונקציה- script Type ב .לשרת url תתבצע בקשת services -ב o
* השרת מקבל את הבקשה ומנווט ל Controller שנמצא בAPI.
* הController יזמן את הפונקציה GetStudentByiD שנמצאת studentBL. o BL מעוניין לקבל נתונים מהDB ולכן הוא פונה לDAL- דרך הframework Entity o ה-DAL שואב את הנתונים הרצויים ממסד הנתונים וכעת מתבצע שלב החזרה.
* ה DAL מחזיר את רשימת התלמידים לשכבת הBL בה מתבצעת פונקצית הסינון של הבאת התלמידים של משתמש מסוים.
* .BLמה controllerמחזירה את הנתונים ל GetStudentByiD הפונקציה
* .serviceהנתונים מוחזרים ל controller -מה
* מהservice חוזרת הרשימה לtypeScript. o הרשימה מוצגת בHTML.

*איור:*

DB

Sql Server

Dal

Entity Framework

Bl

L

ayer

Controllers

Web Api

Client

Models

1. מסד הנתונים הבנוי מטבלאות וקשרי גומלין ביניה ם.
2. שכבת הגישה לנתונים באמצעות Entity Framework.
3. שכבת הישויות.
4. שכבת ה - BL בה כתובים האלגוריתמים.
5. Web Api פרוטוקול התקשורת בי ן צד הלקוח וצד השר ת.
6. .angular, TypeScript צד לקו ח

## ארכיטקטורת רשת (לא רלוונטי )

## תיאור פרוטוקולי התקשורת

http – תמצאו משפט הסבר

## שרת – לקוח

צד השרת נכת ב בטכנולוגית WebApi ובשפת c#.

. angular בטכנולוגית -Html, css ,typescript צד הלקוח נכתב בשפות

## תיאור הצפנות (לא רלוונטי)

# ניתוח ותרשים use case של המערכת המוצעת

תרשים כפי הנלמד בשנה שעברה. (איש לכל סוג משתמש וחץ עם הפעולות שהוא מבצע)

## רשימת use case

רשימת הפעולות המבוצעות ע''י המשתמשים.  
(המשתמש מתחבר ע''י הקלדת שם משתמש וסיסמא,  
 המנהל צופה בהודעות שהתקבלו,  
המשתמש מקבל הודעה לתיבת המייל...)

## תיאור ה-use case העיקריים של המערכת

Identifier , Name, Description , Actors, Frequency , pre-condition ,   
תחפשו כיצד מנסחים תיאור של use case.

## מבני נתונים בהם משתמשים בפרויקט

(מילון / רשימה / תור / עץ / מחסנית / ערימה והסבר איפה ולמה השתמשת במבנה זה)

## תרשים מחלקות

לכל שכבה תיאור המחלקות שלה.

שכבת ה-DAL  
צילום של פירוט המחלקות:  


שכבת ה-BL ...... וכו'.

## תיאור המחלקות

לכל שכבה בפרויקט– הסבר מילולי על השכבה. רשימת כל שמות המחלקות ולכל מחלקה הסבר מילולי. (לפי הסדר – DAL, BL, DTO, WEB-API , צד הלקוח הוא השכבה הגבוהה ביותר)

**DAL – השכבה ................**

מחלקת User – המחלקה משמשת ל....

# תיאור התוכנה

* + סביבת עבודה:

Visual Studio Codeו Visual Studio

* + שפות תכנות:

צד השרת נכתב בטכנולוגית WebApi ובשפת c#.

**.** angular בטכנולוגית -Html, css ,typescript צד הלקוח נכתב בשפות

# אלגוריתמים מרכזיים

כאן תפרטי את הפעולות העיקריות בפרויקט.

## חלק מהאלגוריתם... הפיכת התמונה לשחור לבן

## חלק אחר מהאלגוריתם... פירוק התמונה לאותיות

## החלק העיקרי באלגוריתם למידת מכונה – וזיהוי האות

# קוד האלגוריתם

לכאן תעתיקי את הפונצקיות העיקריות בפרויקט

# תיאור מסד הנתונים

תצלמי את הדיאגרמה מה-SQL

## פירוט הטבלאות ב- Data Base

יש לפרט שם כל טבלה, הסבר על תפקידה ועל עמודות חשובות (מפתח זר או קוד מסוים ) ולמלא:

טבלת ??????? – משמשת ל......

U\_userid – תפקידו הוא.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מפתח | שם שדה | תיאו ר | טיפו ס | שדה שאינו חובה |
| pk | u\_UserId | קוד משתמש | int |  |
|  | u\_FirstName | שם פרט י | string |  |
|  | u\_LastName | שם משפח ה | String |  |

# מדריך למשתמש

## תיאור המסכים

תרשים זרימה של (כל)המסכים:  


## מדריך למשתמש

תכתבי הדרכה למישהו שלא מכיר את האתר שלך ( בכניסה לאתר יש.... בפרופיל האישי ניתן לבחור...)

## צילומי מסכים

לכל מסך: שם המסך ותמונה

# בדיקות והערכה

(דוגמא: כל אחת צריכה לכתוב בניסוח משלה!!!  
לאחר הרצת האלגוריתם נבחנו כל האילוצים שדרושים כדי להביא לשיבוץ אפשרי ואופטימל י. כאשר הופיעו טעויות ובאגים בביצוע של האלגוריתם נבדק הקוד שוב עד שתוקנו הבעיות . לאח ר בדיקות רבות אחר כל מקרי הקצה שעלו בדעתי ,והרצת האלגוריתם מספר פעמים על נתוני ם שונים, האלגוריתם הגיע לקירוב האפשרי ביותר בכלים העומדים לרשות י. )

# ניתוח יעילות

(תסבירי מה זה יעילות למה היא הייתה חשובה לך וכמה הסיבוכיות של האלגוריתם שלך)

# אבטחת מידע

(תכתבתי על כניסה באמצעות סיסמא – אם יש)

# מסקנות

(מסקנות חיוביות ....... ; ) בנוגע לצורת העבודה, לידע שצברתן, איך לכתוב קוד, איך לבדוק, מה חשוב לעשות, מה חשוב לא לעשות, מסקנות לעתיד, מלא אופטימיות וסיפוק.....ופרגון לפרויקט)

# פיתוח עתידי

מה תוסיפי לפרויקט אם יהיה זמן ומשאבים..........

# ביבליוגרפיה

רשימת האתרים שבהם נעזרת במהלך העבודה.