

**#E53F00**

סמל מוסד: 560342

שם מכללה: סמינר תורני בית יעקב

שם הסטודנט: הודיה חיה אוחנה

ת.ז. הסטודנט:213741572  
שם הפרויקט: Digivert

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)

# הצעת פרויקט

תיאור הפרויקט:

התוכנה מיועדת לאדריכלים, מהנדסי בניה, ארכיטקטים ומעצבי פנים.

משתמש חדש נכנס למערכת. תחילה הוא מכניס שם משתמש וסיסמה שנשמרים במאגר וכך יוצר לעצמו תיק עבודות.

בפני המשתמש עומדת אפשרות של פתיחת פרויקט. לתיקיית הפרויקט הוא יכול להוסיף תמונות של תכניות, שרטוטים וכד׳. בשלב זה תהיה לו אפשרות לערוך את הקובץ מבחינה גרפית בסיסית לדוגמא:

* הוספת מידות מנקודה לנקודה.
* הוספת קווים לשינוי השרטוט.
* עריכת השרטוט ועוד.

אפשרות נוספת היא המרת סוג קובץ.

המערכת מקבלת תמונה או קובץ PDF מהמשתמש וממירה אותו לקובץ DWG קבצי CAD.

התוכנה תאפשר למעצבים לסרוק תכניות בניה מודפסות או מצולמות. כמו כן היא תאפשר להמיר את הקובץ לקובץ שיאפשר להם לערוך בו שינויים בצורה מדויקת ובכך להשתמש בו בתוכנות המיועדות לשרטוט תכניות.

התוכנה חוסכת למעצבים העתקה פיזית מתוכנית מודפסת לתוכנה.

הגדרת הבעיה האלגוריתמית :

* זיהוי שרטוט מתמונה.
* המרת תמונה לקובץ DWG CAD.

לצורך כך עלי ללמוד לעומק על כל הנושא של אדריכלות ועיצוב פנים. עלי לקרוא חומרים שונים, להכיר את הנושא מבפנים ולקבל חוות דעת מקצועית מאנשי מקצוע.

נוסף על כך עלי לחפש אלגוריתמים שונים שאוכל להשתמש בהם לצורך ביצוע התוכנית בצורה האופטימלית ביותר. אלגוריתמים של עיבוד תמונה וניתוח תמונה כדי להפוך את הקובץ המועלה לתוכנה לקובץ בר עריכה.

לעיתים התמונות השונות באיכות ירודה וכתוצאה מכך יכולים להיווצר אי דיוקים ועלי לנסות להתגבר על כך.

רקע תיאורטי בתחום הפרויקט:

מהנדסים רבים מבזבזים זמן וכסף רב בהעתקת תוכניות מודפסות לתוכניות ממוחשבות.

התוכנית תביא לחיסכון רב בזמנם של אנשי מקצוע ותביא לתוצאות מדויקות בהרבה מהעתקה ידנית.

תהליכים עיקריים בפרויקט:

1. משתמש נרשם למערכת ונפתח לו תיק פרויקטים אישי .
2. המשתמש יבחר קובץ שהוא רוצה להמיר.
3. המערכת תקבל את קובץ התמונה.
4. המערכת תזהה את קווי השרטוט בקובץ.
5. המערכת תמיר את הקובץ לקבציי CAD(dwg) .
6. המערכת תשמור את הקובץ בתיק הפרויקטים האישי שלו.

תיאור הטכנולוגיה:

צד שרת:

**שפת תכנות בצד השרת:** c#

צד לקוח:

**שפת תכנות בצד לקוח:** angular

מסד נתונים**:** SQL

פרוטוקולי תקשורת:

לוחות זמנים:

1. חקר המצב הקיים – ספטמבר
2. הגדרת הדרישות – ספטמבר
3. אפיון המערכת – אוקטובר
4. אפיון בסיס נתונים – נובמבר
5. עיצוב המערכת – דצמבר
6. בניית התוכנה – ינואר, פברואר
7. בדיקות – מרץ
8. הכנת תיק פרויקט – אפריל
9. הטמעת המערכת – מאי
10. הגשת פרויקט סופי - מאי

חתימת הסטודנט: הודיה חיה אוחנה

חתימת רכז המגמה:

אישור משרד החינוך:

# מבוא / תקציר

## הרקע לפרויקט

חיפשתי רעיון לאפליקציה שתוכל לחסוך לאנשים הרבה כאב ראש ,ותהיה יעילה בצורה המרבית.

בתור אחת שבאה ממשפחה שכולם מתעסקים בענף הבניה והתכנון, תמיד ראיתי אותם משלמים הרבה מאוד כסף לאנשים כדי שיעשו להם שינויים ועריכה לתוכניות/שרטוטים קיימים ,לפעמים אפילו שינויים קטנים .

או כשהם צריכים להעתיק תכניות שלמות לאוטוקאד(תוכנה שירטוט לאדריכלים ומעצבי פנים)מה שלוקח המון זמן, ואז עלה לי רעיון לפתח אפליקציה/תוכנה שפשוט תעשה להם את העבודה לקלה יותר ותחסוך להם הרבה זמן וכסף ומאמץ לא מבוטל.

וכאחת שבאה ממשפחה שמתעסקת בענף הבניה, אני רואה את הצורך הרב שיש להם באמצעי עזר נגיש וממוחשב ,שיעזור להם לחסוך זמן, כסף ושיעזור להם לארגן את כל הקבצים בצורה מסודרת.

הפרויקט כולל בתוכו עיבוד תמונות, התפתחות טכנולוגית

שם הפרויקט הוא (digital convert- )Digivert-המרה דיגיטלית,

וכולי תקווה שאכן פתרון זה יהיה שימושי ויקל מעל כולם את הקושי ואת המורכבות שבדבר.

האלגוריתם שעזר לי לפתח זה אלגוריתם לעיבוד תמונה .

## תהליך המחקר

כאשר רציתי להתחיל לפתח את הפרויקט התחלתי לחקור כמה דברים על קבצים מהסוג הזה לדוג:

1. מהם סוגי הקבצים שמשמשים לשרטוט תכניות
2. ממה מורכב קבצים אלו
3. איך אוכל להמיר קובץ תמונה לקובץ CAD,DWG,DFX(קבצים לשרטוט)
4. אילו ספריות יכולות לעזור לי לעשות את זה
5. חיפוש אלגוריתמים לביצוע המרב
6. חיפוש Api להמרת קבצי תמונה לקבציי CAD

ואכן מצאתי ספריות שמשמשות ליצירת קבצי CAD-ספרית AutoCad.Net,CADlib-הספריות מאפשרות לערוך ,ליצור קבצי CAD.

חקרתי איך להשתמש בספריות הללו ומצאתי את ספריית AutoCad.Netכמתאימה ביותר משום שהפרויקט שלי מיועד לקבציי CAD ובעיקר קבציי DXF ולשימוש בתוכנת AutoCAD אשר מיועדת גם ובעיקר לקבציי DXF .

## סקירת ספרות

לשם פיתוח הפרויקט נעזרתי באתרי מידע רבים ,הן באתרי קוד והן באתרי חקר ומידע על קבצים המיועדים לשרטוטי בנייה.  
קראתי חומרים רבים על אלגוריתמי עיבוד תמונה ,אלגוריתמי איור ווקטורי ועל אלגוריתמים ליצירת קבצי DXF.  
בנוסף נעזרתי באתרים לתיקוני וביאורי שגיאות וכדומה.

* האפליקציה פותחה בשפת c# משום שהיא נוחה וגמישה לעבודה.
* צד לקוח פותח באנגולר משום שהוא נוח למשתמש ויעיל למתכנת.  
   כמו כן אנגולר היא שפה מוכרת ונפוצה בשוק העבודה והיה חשוב לי להכיר אותה לעומק ולהתמקצע בה ואכן , הפרויקט סייע לי בכך רבות.

לשם חקר אחר האלגוריתם לפרויקט שלי נעזרתי בעיקר באתר AUTODESK למציאת ,מכלול, ויקיפדיה.

* שפות, שרת- C# stack overflow, לקוח-Angular
* עיצוב-bootstrap,w3school, stackoverflow.

# מטרות ויעדים

כאחת שנפגשת עם קהל היעד של הפרויקט שלי , ראיתי אותם עמלים רבות, על כתיבת תכניות במשך שעות רבות, וכשהיה מדובר בפרויקטים גדולים הם היו צריכים לשכור בעלי מקצוע בסכום גבוה כדי שיעשו עבורם את העבודה ה"שחורה" כביכול.

בנוסף אחד מתחומי העניין שלי זה עיצוב פנים ואני גם מתמצאת בתחום כך שביצוע הפרויקט וגם המחקר על הנושא הוסיפו לי רבות הן ברמה המקצועית והן ברמה האישית .מה גם שתמיד עניין אותי לדעת איך תוכנת AutoCad עובדת ממה הקבצים שלה מורכבים ואיך הכול פועל "מאחורי הקלעים" מה שנקרא .

המטרה המרכזית שלי בפרויקט הזה זה ליצור מענה אמין ויעיל לצורך של מהנדסים, ארכיטקטים ומעצבי פנים ולכל מי שמגיע מענף הבנייה

ולפתור אחת ולתמיד את הבעיה שנקראת העתקת שרטוטים למחשב או כתיבת תכניות מחדש לשם עריכת שינויים פעוטים.

אחד המטרות שלי הוא ליצור ממשק נוח למשתמש שבו הוא יוכל לארגן את הקבצים שלו בתיקיות מסודרות לפי פרויקטים כך שהוא יוכל למקם כל תכנית לפי הנושא שלה והפרויקט שאליה היא שייכת.

היעד שאליו אני שואפת להגיע הוא :

שהמשתמש יוכל לבחור תמונה של תכנית בניה ולערוך אותה בתוכנה עם ממשק קליל ונוח ללא סיבוכים או/ובנוסף להמיר אותה לקובץ CAD כך שיוכל להשתמש בקובץ בתוכניות המותאמות לעריכת קבצים אלו באיזה צורה שיבחר.

המערכת תשמור את נתוני התמונה שהומרה שזהו בעצם בקובץ החדש שנוצר בתוך תיק העבודות של המשתמש בתיקיית הפרויקט שאותה המשתמש יבחר לנכון כמתאימה .

# אתגרים

בתחילת הפרויקט נחשפתי לחומרים ומידע שכלל לא הכרתי מעולם די זר של עיבודי תמונה והבנה איך אלגוריתמים של עיבוד תמונה עובדים,  
כל העניין של הכרת רכיבי התמונה,  
החשיבה על איך אני אמורה לחפש צורה מסוימת/ אובייקט מסוים מתוך התמונה הכללית ולהגדירה כאובייקט/גרופ.

וגם הכרת עולם נוסף של קבצי DXF /המרה לקבציי DXF :  
ממה מורכב קובץ הDXF ?

במהלך הפרויקט נתקלתי באתגרים:

\*ההבנה שעלי להמיר את השרטוט בתמונה לאיור וקטורי  
\*איך אני יוצרת קובץ DXF?  
\*איך אני כותבת את האיור הווקטורי שחילצתי מהתמונה לקובץ הDXF?  
\*איך אני מזהה אובייקטים מתוך תמונה/שרטוט?  
\*הרבה זמן חיפשתי ספריה שתוכל לממשק אותי עם AutoCAD עד שמצאתי ספרייה מתאימה לביצוע הפרויקט שלי.

במהלך העבודה נהניתי לראות שגם אם בסופו של יום של חיפושים ,מחקר וכתיבה לא יצא מזה משהו ממשי ,עדיין החכמתי מאוד הן בנושא הפיתוח שלי והן בידע כללי שלאו דווקא הכרחי לפרויקט שלי.

גם הבנתי שהדברים אינם פשוטים כמו שנראו בתחילה. אפרט כמה נקודות: בתחילה היה נראה שזה בסך הכול למצוא את כל הנקודות של השרטוט מהתמונה וכתיבתם היישר לקובץ אוטוקאד , אך אחרי שחקרתי את דרכי הפעולה הבנתי שזה הרבה יותר מורכב מזה.

אך יש לציין שזה היה מרתק לראות מה עומד מאחורי כל דבר ,ואיך זה עובד .

וגם איך באמת כותבים תוכנה /אפליקציה .

יש לציין שפרויקט זה וכל הכלול בו ,ההצלחות והקשיים, האתגריות שבו, גרמו לי להבין באמת איך ליישם את כל החומר שלמדתי בפועל בצורה מופלאה.

# 5.מדדי הצלחה

האפליקציה שלי הצליחה אם:

1.התוכנה אכן ממירה קובץ של תמונה לקובץ CAD.

2.התוכנה אכן מזהה אובייקטים בשרטוט.

3.אם ואכן הממשק למשתמש יהיה נוח ומונגש לכולם

4.כאשר כל הDATA יישמר בצורה מושלמת בDB

# 6.תיאור המצב הקיים

לאחר בירור אצל בעלי מקצוע וחיפוש באינטרנט אכן ישנם תוכנות שנותנות את השירות שהפרויקט שלי מציע אך התוכנות/אפליקציות האלו הם בעלות חודשית גבוהה ,מה שגורם לכך שהרבה מעדיפים לעבוד קשה ולא לשלם על תוכנות כאלו ואחרות, במיוחד אם מדובר בבעלי מקצוע פרטיים ולא שכירים.

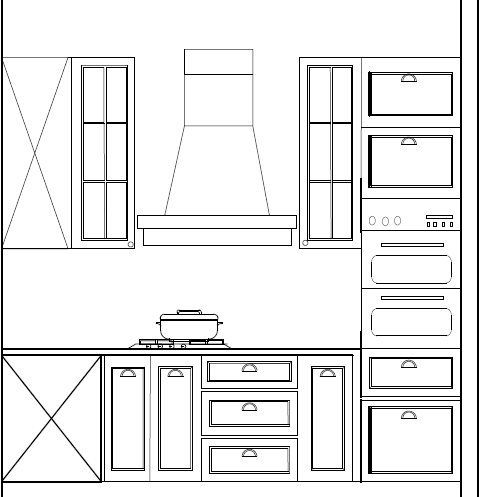
מה גם שאחד מהבעיות שרציתי לפתור זה לחסוך את העלות הכספית .

# 7.רקע תאורטי

הפרויקט אמור להפוך תרשים שמשרטט מעצב פנים / אדריכל לקובץ ממוחשב שלDFX ממשפחת קבצי CAD

וכן מאפשר לו לערוך אותו בצורה נוחה וקלה.

להלן דוגמא של קובץ CAD



קובץ CAD מורכב **בצורה וקטורית**.

**וקטור:** הוא ישות מתמטית בעלת גודל וכיוון.

**צורה וקטורית:**

השיטה הבסיסית ביותר לייצוג תמונה באופן ממוחשב ע"י bitmap -מפת סיביות  שהיא למעשה  מערך דו-ממדי, שכל איבר בו מייצג נקודה במישור. בכל איבר במערך שמורים נתונים אודות הנקודה :מיקומה וצבע הנקודה.

בשיטה זו, ייצוג של קטעים  נעשה מיותר, במקום לייצג את הקטע מייצגים מספר סופי של נקודות עליו. ממילא, מתיחה של התמונה לשם הגדלה תיצור טשטוש, כי רק הנקודות יוגדלו ולא כל הקטע.

גרפיקה וקטורית עושה שימוש בווקטורים כדי לייצג את הקטע עצמו. שמירת נקודות המוצא והסיום מספיקות כדי לייצג קטע.

איורים בגרפיקה וקטורית מורכבים צורות בסיסיות, אותן המחשב מסוגל לעבד בקלות, כמו נקודה, קו, אליפסה וקשת.

יתרונה העיקרי של גרפיקה וקטורית הוא ביכולת לשנות את ממדי התמונה (להרחיב אותה או לכווץ אותה) בלא כל פגיעה באיכות התמונה, שלא כמו במקרה של תמונת מפת סיביות. כמו כן, כאשר מדובר באיורים פשוטים יחסית נפח האחסון יחסית הנדרש בייצוג וקטורי קטן משמעותית לעומת מפת סיביות.

**קבצי CAD:**

קבצי CAD (Computer Aided Design) זה משפחה של קבצים דיגיטליים המכילים עיצובים דו ממדיים ותלת ממדיים וגם מכילים מידע לגבי חומרים ,תהליכים ונתונים אחרים.

בתוך המשפחה הזו קיימים בין השאר סוגי הקבצים: DFX DWG

**קובץ DXF:**

פותח עבור תוכנות CAD ,מהווה פורמט מוביל בתחום הגרפיקה הממוחשבת הווקטורית.

**האלגוריתם:**

**האלגוריתם שלי מתחלק שלושה שלבים:**

לשם יצירת קובץ DXF מתמונה ,עלי לעבד את התמונה כך שיהיה אפשר לקרוא ממנה נתונים מדויקים לשם כך אבצע את השלבים הבאים:

**1.עיבוד תמונה:**

-לקבל תמונה

-להפוך לשחור לבן

**2. האלגוריתם העיקרי והמורכב זה קריאת הנתונים המתאימים מהתמונה והפיכתם לנתונים ווקטוריים:**

-זיהוי האלמנטים הגרפיים (קו, קשת, אליפסות, נקודה)

-לזהות צורה: קו. עיגול, ריבוע, קשת.

האלגוריתם מקבל תמונה של שרטוט ומזיר רשימת ווקטורים.

שמהם יורכב קובץ ה. Dxf

**3.יצירת קובץ dxf:**

לשם יצירת קובץ DXF אני משתמשת בספרייה של Autocad.net API

של תוכנת אוטוקאד (תוכנת שרטוט. התוכנה משמשת לתכנון וייצור בעזרת מחשב לתחומים של הנדסה אזרחית ואדריכלות).

אובייקטים הם אבני הבניין העיקריות של AutoCAD .NET API,

כל אובייקט חשוף מייצג חלק מדויק של AutoCAD, והם מקובצים למכלולים ומרחבי שמות שונים.

לדוג:

\* אובייקטים גרפיים: קווים, קשתות, טקסט וממדים.

\*עיצוב: סגנונות טקסט ומימד.

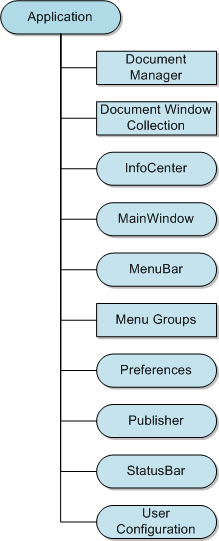
\*מבנים ארגוניים : שכבות, קבוצות ובלוקים

\*תצוגה : אובייקט תצוגה.

קובץ אוטוקאד בנוי מכל האובייקט שמקושרים ביניהם בהיררכיה כאשר האובייקט של AutoCAD Application נמצא בשורש.

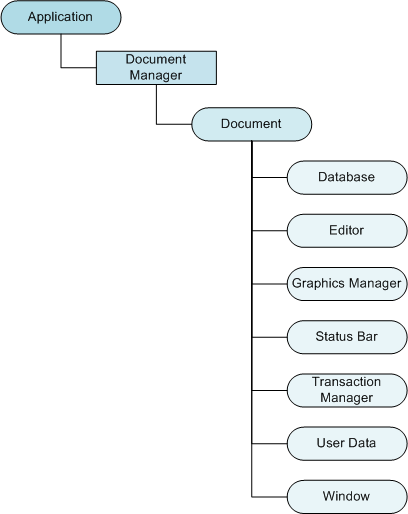
**אובייקט האפליקציה:**

אובייקט היישום הוא אובייקט הבסיס של ה-API של AutoCAD .NET. מאובייקט היישום, אתה יכול לגשת לחלון הראשי וכן לכל ציור פתוח. ברגע שיש לך ציור, תוכל לגשת לאובייקטים בציור .



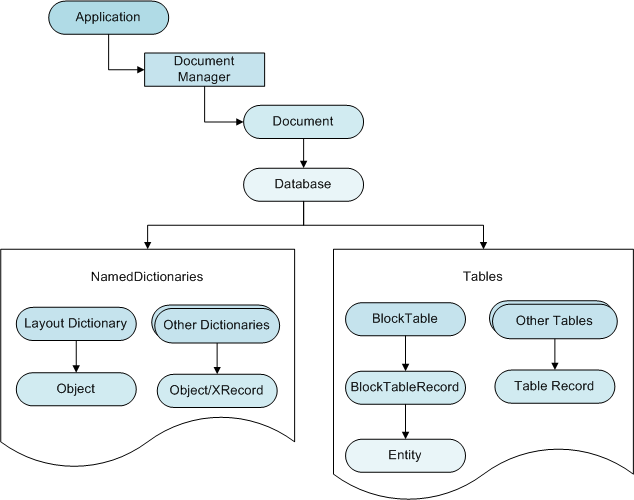
**אובייקט מסמך:**

אובייקט ה-Document, שהוא למעשה ציור AutoCAD, הוא חלק מהאובייקט DocumentCollection ומספק גישה לאובייקט ה-Database המשויך לאובייקט ה-Document.



**דאטה בייס:**

אובייקט מסד הנתונים מכיל את כל האובייקטים הגרפיים ורוב האובייקטים הלא-גרפיים של AutoCAD. .



**אובייקטים גרפיים:**

אובייקטים גרפיים, הידועים גם בשם ישויות, הם האובייקטים הגלויים (קווים, עיגולים, תמונות רסטר וכדומה) המרכיבים ציור.

**אוספים:**

AutoCAD מקבץ את רוב האובייקטים הגרפיים והלא-גרפיים לאוספים או אובייקטים של מיכל.

לכל אוסף יש שיטה להוספת חפץ או קבלת פריט מאוסף. רוב האוספים משתמשים בשיטות Add או SetAt כדי להוסיף אובייקט לאוסף.

**אובייקטים גרפיים ולא- גרפיים שאינם מקוריים:**

כאשר ניגשים דרך AutoCAD .NET APIניגשים לאובייקט ה-COM המקביל ממאפיין.

# 8.ניתוח חלופות מערכתי

**objects detection from images with OpenCV . 1 –** אימוןהסורק עם כמה מאות אפשרויות של אובייקט מסוים המותאמות לאותו קנה מידה, ודוגמאות שליליות –תמונות שרירותיות באותו גודל, לאחר אימון הסורק, ניתן להחיל אותו על אזור עניין בתמונת קלט.

הסורק מוציא "1" אם סביר שבאזור יוצג את האובייקט –"0" אחרת.  
כדי לחפש את האובייקט בתמונה כולה אפשר להעביר את החיפוש על כל התמונה ולבדוק כל מיקום באמצעות הסורק.  
הסורק מתוכן כך שניתן "לשנות את גודלו" בקלות על מנת שניתן יהיה למצוא את האובייקטים בגדלים שונים, וזה יעיל יותר משינוי גודל התמונה עצמה .לכן כדי למצוא אובייקט בגודל לא ידוע בתמונה, הליך הסריקה צריך להיעשות מספר פעמים בקנה מידה שונה.  
הסורק שנוצר מורכב ממספר סורקים (שלבים)פשוטים יותר המוחלים אחר כך על אזור עניין עד שבשלב מסוים האובייקט המועמד נדחה או שעובר את כל השלבים.

2**.using AutoCAD.Net API**- מאפשר לך להפוך משימות לאוטומטיות כגון יצירה ושינוי של אובייקטים המאוחסנים במסד הנתונים של קובץ ציור או לשנות את התוכן של קובץ התאמה אישית.

בחרתי בחלופה השנייה: **using AutoCAD.Net API**

# 9.תיאור החלופה הנבחרת והנימוקים לבחירה

**objects detection from images with OpenCV 1.**

יתרון: הOpenCV הוא מבנה מודולרי שכולל מספר ספריות משותפות או סטטיות.  
\*הספרייה כוללת מודלים רבים ופונקציונליות רבה.  
\* מטפל בכל הזיכרון באופן אוטומטי.

חסרון :לשימוש בשפת C++.

2. ישנם מספר יתרונות להטמעת .NET API AutoCAD:

* גישה פרוגרמתית לשרטוטי AutoCAD נפתחת לסביבות תכנות נוספות. לפני ה-.NET API, מפתחים היו מוגבלים לאוטומציה של ActiveX® ולשפות שתמכו ב-COM, AutoLISP® ו-C++ עם ObjectARX.
* שילוב עם יישומים אחרים מבוססי Windows®, כגון Microsoft Excel ו-Word, נעשה קל יותר באופן דרמטי על ידי שימוש ב-.NET API המקורי של יישום או בספריית ActiveX/COM חשופה.
* ה-.NET Framework מיועד למערכות הפעלה של 32 סיביות ו-64 סיביות. Visual Basic for Applications תוכנן רק עבור מערכות הפעלה של 32 סיביות.
* מאפשר גישה לממשקי תכנות מתקדמים עם עקומת למידה נמוכה יותר מאלו של שפות תכנות מסורתיות יותר כמו C++.

1. לגבי האלגוריתם שנבחר. כיצד הוא עובד? (מהם הם השלבים שלו, החלקים שלו, ממה מורכב?) (תוכלי לתת דוגמאות והמחשה ע''י תמונות)

# 10.אפיון המערכת

**סביבת פיתוח :**

חומרה: מעבד

עמדת פיתוח: מחשבLenovo

מערכת ההפעלה:

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 מינימאלית :

* + חומרה: מעבדRAM 4GB i5 .

o מערכת ההפעלה: Windows 7ומעלה.

* + חיבור לרשת: נדרש .
  + תוכנות: . chrome

## .10.1ניתוח 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.

## 10.2.מודול המערכת

* הרשמת הלקוח למערכת והזנת הנתונים שלו DB.
* העלאת קובץ png jpg/ או כל קובץ תמונה.
* קריאת הקובץ שהועלה והזנת הנתונים בדאטה בייס.
* עיבוד התמונה וזיהוי האובייקטים מהתמונה.
* המרת התמונה לאיור וקטורי.
* עריכת האיור ע"י המשתמש.
* יצירת קובץ DXF.
* הזנת נתוני האיור לקובץ הDXF.
* הצגת קובץ הDXF ללקוח +אפשרות להורדה.

## 10.3אפיון פונקציונאלי

(פירוט פונקציות עיקריות ותפקידן)

MyFunc1 – הפונקציה מבצעת....

MyFunc2 – הפונקציה מסננת את...

## 10.4ביצועים עיקריים

המשתמש מכניס את פרטיו ונכנס למערכת בנוסף נפתח לו תיק עבודות ,  
המשתמש בוחר קובץ מתוך הגלריה שלו ,  
למשתמש ניתנת האפשרויות :  
1.לערוך את הקובץ.  
2.להמיר את הקובץ לקובץ DXF.  
לאחר המרת הקובץ ,הקובץ יתווסף לתיק העבודות של המשתמש.

## 10.5אילוצים

המערכת מתבססת על קובץ ,DXF.

המערכת תוכל לפעול רק וכאשר תוכנת AutoCAD מותקנת על המחשב

המערכת פועלת ע''פי נתוני התוכנה ואינה יכולה לנתח ולהמיר את הקובץ בלעדיהם.

# 11.תיאור הארכיטקטורה

## 11.1.הארכיטקטורה של הפתרון המוצע בפורמט של Design level Down-Top

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

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 ל 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.

## 11.2.תיאור הרכיבים בפתרון.

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, הצהרתיות, פונקציונאליות פרוצדוראליות וגנריות .

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

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, תכניות וכן של שאר הרכיבים המסופקים עם השרת ואובייקטי המשתמש.

***זרימת מידע במערכת***

שליפת כל הפרויקטים.

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

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

## 11.3.ארכיטקטורת רשת (לא רלוונטי )

## 11.4.תיאור פרוטוקולי התקשורת

## **שרת) HTTP** נקרא גם **שרת דפי אינטרנט**, **שרת web** או **שרת אינטרנט**) הוא תוכנת שרת המתקשרת בפרוטוקול **.HTTP**

## השם משמש לעיתים גם ככינוי למחשב עליו פועלת תוכנת השרת, אם כי לאותו מחשב יכולים להיות גם תפקידים נוספים. שרתי HTTP הם שרתי התוכן המרכזיים ברשת האינטרנט וברשתות מחשבים אחרות, ומשמשים להפצת דפי HTML, תמונות, וקבצים אחרים.

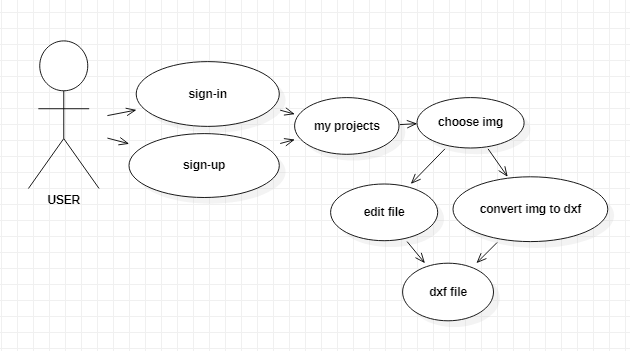
## 11.5. שרת – לקוח

צד השרת נכת ב בטכנולוגית WebApi ובשפת c#.

צד הלקוח נכתב בשפות -Html, css ,typescript בטכנולוגיתAngular

## 11.6.תיאור הצפנות (לא רלוונטי)

# 12.ניתוח ותרשים use case של המערכת המוצעת



## 12.1.רשימת use case

רשימת הפעולות המבוצעות ע''י המשתמשים.

* המשתמש מתחבר /נרשם למערכת.
* המשתמש פותח תיקיית פרויקט חדשה .
* המשתמש בוחר תמונה מהגלריה .
* למשתמש ניתנת האפשרות לערוך את השרטוט.
* למשתמש ניתנת האפשרות להמיר את השרטוט לקובץ DXF.
* מוצג קובץ DXF

## 12.2.תיאור ה-use case העיקריים של המערכת

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

## 12.3מבני נתונים בהם משתמשים בפרויקט

רשימה-  
1.לצורך טעינה והוספה של קבצים לתיק העבודות של המשתמש.  
השתמשתי משום שהוא יעיל מבחינת הקצאה דינאמית .  
2.לצורך שמירת האובייקטים והמיקום שלהם.

## 12.4.תרשים מחלקות

לכל שכבה תיאור המחלקות שלה.

שכבת ה-DAL  
צילום של פירוט המחלקות:

שכבת ה-BL ...... וכו'.

## 12.5.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 קוראת את המידע הנדרש למערכת ,שומרת נתונים ,מוסיפה מידע חדש או מוחקת פרטי מידע קיימים**

מחלקת User – המחלקה משמשת לשמירת נתוני המשתמש בDB.

מחלקת File-משמשת לשמירת נתוני הקובץ.  
מחלקת Savedblock-משמשת לשמירת נתוני הבלוקים /אובייקטים.

**שכבת BL–**השכבה אחראית על הלוגיקה של המערכת, עוסקת בעיבוד המידע, בחישובים שונים ושליחתו לשכבת התצוגה.  
בשכבה זו נממש את הפונקציונאליות של המערכת.

שכבת DTO- זהו אובייקט העברת נתונים המעביר נתונים בין תהליכים.

# 13.תיאור התוכנה

* + סביבת עבודה:

Visual Studio Codeו- Visual Studio

* + שפות תכנות:

צד השרת נכתב בטכנולוגיתWebApi ובשפת .c#

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

# 14.אלגוריתמים מרכזיים

כאן תפרטי את הפעולות העיקריות בפרויקט.

## 14.1 חלק מהאלגוריתם הוא הפיכת התמונה לשחור לבן .

## 14.2. החלק העיקרי באלגוריתם הוא פירוק התמונה לאובייקטים והחזרת רשימת ווקטורים של האובייקטים.

14.3.חלק נוסף מהאלגוריתם הוא יצירת קובץ DXF שמורכב מרשימת הווקטורים של האובייקטים.

# 15.קוד האלגוריתם

לכאן תעתיקי את הפונצקיות העיקריות בפרויקט

# 16.תיאור מסד הנתונים

תצלמי את הדיאגרמה מה-SQL

## פירוט הטבלאות ב- Data Base

טבלת User– משמשת לשמירת נתוני המשתמש.

Iduser- משמש כקוד זיהוי למשתמש.

-firstName שם פרטי של המשתמש.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מפתח | שם שדה | תיאור | טיפוס | שדה שאינו חובה |
| pk | Userid | קוד משתמש | int |  |
|  | firstName | שם פרטי | varchar | + |
|  | lastName | שם משפחה | varchar | + |
|  | e-mail | שם המשתמש | varchar | + |
|  | password | סיסמא | varchar | + |

טבלת file– משמשת לשמירת נתוני הקובץ.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מפתח | שם שדה | תיאור | טיפוס | שדה שאינו חובה |
| pk | idfile | קוד קובץ | bigint |  |
|  | idproject | קוד פרויקט | varchar |  |
|  | srcimagetarget | מקור הקובץ | varchar |  |
|  | newimagetarget | ניתוב של הקובץ החדש | varchar |  |

טבלת saved block– משמשת לשמירת נתוני הבלוקים /אובייקטים.

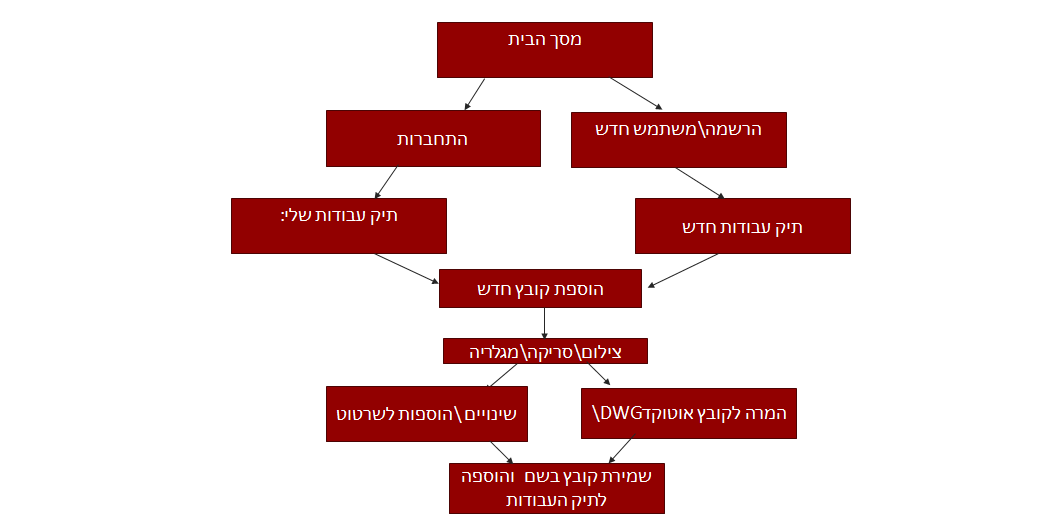
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מפתח | שם שדה | תיאו ר | טיפו ס | שדה שאינו חובה |
| pk | idblock | קוד בלוק | bigint |  |
|  | imagetarget | ניתוב של תמונת הבלוק | varchar |  |

טבלת project– משמשת לשמירת נתוני הפרויקט.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מפתח | שם שדה | תיאור | טיפוס | שדה שאינו חובה |
| pk | idproject | קוד פרויקט | int |  |
|  | iduser | קוד המשתמש של בעלי הפרויקט | varchar |  |

# 17.מדריך למשתמש

# 17.1.תיאור המסכים

תרשים זרימה של (כל)המסכים:  


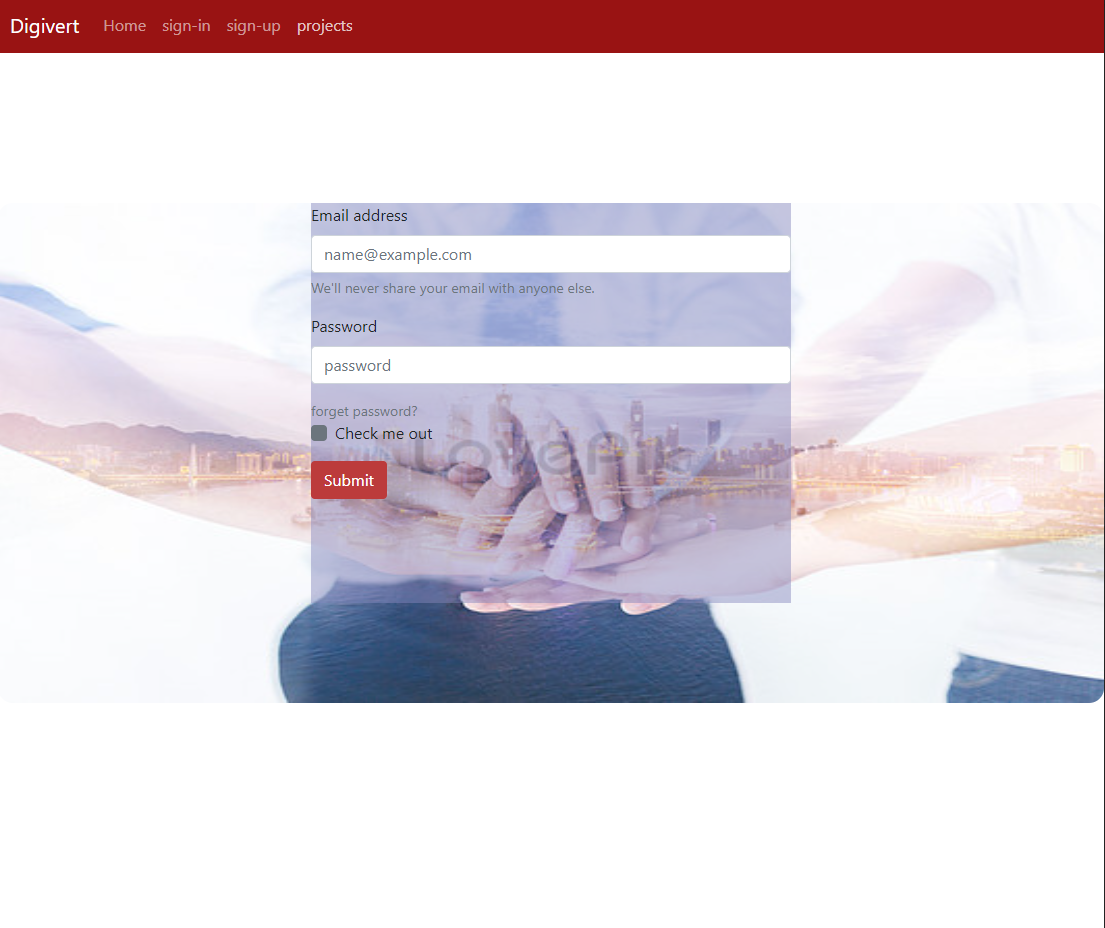
## 17.2.מדריך למשתמש

היום, נסקור כיצד Digivert יכול להמיר בקלות image לקבציי DXF.

1.

## 17.3.צילומי מסכים

1.



ן05DD

# 18.בדיקות והערכה

לאחר הרצת האלגוריתם נבחנה התוצאה של המרת הקובץ והאם התוצאה היא התוצאה האופטימלית.

כאשר הופיעו טעויות ובאגים בביצוע של האלגוריתם ,בדקתי את הקוד שוב עד שנמצאה הבעיה ,והבעיה תוקנה.  
לאחר בדיקות רבות עם כל מקרי הקצה האפשריים שעלו בדעתי ,והרצת האלגוריתם מספר פעמים עם נתונים שונים, האלגוריתם הגיע לקירוב האפשרי ביותר למצב האופטימלי בכלים העומדים לרשותי.

# 19.ניתוח יעילות  **יעילות אלגוריתמית** מתייחסת לכמות צריכת משאבי מערכת של אלגוריתם, ובפרט משאבי זמן וזיכרון, יעלות אלגוריתמית חשובה אף יותר ממחשב מהיר.

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

# 20.אבטחת מידע

כניסה למערכת מאופשרת באמצעות סיסמא ושם משתמש.

# 21.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)

# 22.פיתוח עתידי

אשמח להוסיף לתוכנה :

* + - 1. אפשרות לזהות אובייקטים של תלת ממד
      2. עורך שרטוטים מתקדם יותר
      3. הרחבת האפשרויות של המרת הקבצים

# 23.ביבליוגרפיה

* מכלול
* ויקיפדיה
* stack overflow
* Autodesk forum
* W3school
* Bootstrap