**פרויקט גמר 2022**

תהילה אברהמי

213289754

**תוכן**

[**1.** **הצעת פרויקט** 4](#_Toc103039593)

[**2.** **מבוא / תקציר** 7](#_Toc103039594)

[**2.1.** **הרקע לפרויקט** 7](#_Toc103039595)

[2.2 תהליך המחקר 9](#_Toc103039596)

[2.3 סקירת ספרות 10](#_Toc103039597)

[3. מטרות ויעדים 11](#_Toc103039598)

[4. אתגרים 12](#_Toc103039599)

[5. מדדי הצלחה 13](#_Toc103039600)

[6. תיאור המצב הקיים 14](#_Toc103039601)

[7. רקע תאורטי 15](#_Toc103039602)

[8. ניתוח חלופות מערכתי 16](#_Toc103039603)

[9. תיאור החלופה הנבחרת והנימוקים לבחירה 17](#_Toc103039604)

[10. אפיון המערכת 18](#_Toc103039605)

[10.1. ניתוח דרישות המערכת 18](#_Toc103039606)

[10.2. מודול המערכת 18](#_Toc103039607)

[10.3. אפיון פונקציונאלי 19](#_Toc103039608)

[10.4. ביצועים עיקריים 19](#_Toc103039609)

[10.5. אילוצים 19](#_Toc103039610)

[11. תיאור הארכיטקטורה 20](#_Toc103039611)

[11.1. הארכיטקטורה של הפתרון המוצע בפורמט של Design level Down-Top 20](#_Toc103039612)

[11.2. תיאור הרכיבים בפתרון: 20](#_Toc103039613)

[11.3. ארכיטקטורת רשת (לא רלוונטי ) 22](#_Toc103039614)

[11.4. תיאור פרוטוקולי התקשורת 22](#_Toc103039615)

[11.5. שרת – לקוח 22](#_Toc103039616)

[11.6. תיאור הצפנות (לא רלוונטי) 22](#_Toc103039617)

[12. ניתוח ותרשים use case של המערכת המוצעת 23](#_Toc103039618)

[12.1. רשימת use case 23](#_Toc103039619)

[12.2. תיאור ה-use case העיקריים של המערכת 23](#_Toc103039620)

[12.3. מבני נתונים בהם משתמשים בפרויקט 23](#_Toc103039621)

[12.4. תרשים מחלקות 23](#_Toc103039622)

[12.5. תיאור המחלקות 24](#_Toc103039623)

[13. תיאור התוכנה 26](#_Toc103039624)

[14. אלגוריתמים מרכזיים 27](#_Toc103039625)

[14.1. חלק מהאלגוריתם... הפיכת התמונה לשחור לבן 27](#_Toc103039626)

[14.2. חלק אחר מהאלגוריתם... פירוק התמונה לאותיות 27](#_Toc103039627)

[14.3. החלק העיקרי באלגוריתם למידת מכונה – וזיהוי האות 27](#_Toc103039628)

[15. קוד האלגוריתם 28](#_Toc103039629)

[16. תיאור מסד הנתונים 29](#_Toc103039630)

[16.1. פירוט הטבלאות ב- Data Base 29](#_Toc103039631)

[17. מדריך למשתמש 30](#_Toc103039632)

[17.1. תיאור המסכים 30](#_Toc103039633)

[17.2. מדריך למשתמש 31](#_Toc103039634)

[17.3. צילומי מסכים 32](#_Toc103039635)

[18. בדיקות והערכה 33](#_Toc103039636)

[19. ניתוח יעילות 34](#_Toc103039637)

[20. אבטחת מידע 35](#_Toc103039638)

[21. מסקנות 36](#_Toc103039639)

[22. פיתוח עתידי 37](#_Toc103039640)

[23. ביבליוגרפיה 38](#_Toc103039641)

# **הצעת פרויקט**

**סמל מוסד:**

**שם מכללה:**

**שם הסטודנט:** תהילה אברהמי

**ת.ז הסטודנט:** 213289754

**שם הפרויקט:**

**תיאור הפרויקט:**

האפליקציה מזהה פנים אנושיות עפ"י תמונה קודמת שהוא מכניס למערכת או עפ"י תמונות שלו שאחרים מעלה לאפליקציה בזמן אמת כדי למוצאו.

משתמש חדש יכול להירשם לאפליקציה. לצורך כך עליו להכניס את פרטיו האישיים: שם, שם משפחה, ת.ז ומספרי טלפון וכן תמונת פנים. כל אלה ישמרו במאגר המשתמשים.

ישנה אופציה נוספת של העלאת תמונות בזמן אמת. במצב של היעדרות אדם ישנה אופציה שהוא נמצא בבית החולים ללא זיהוי במיוחד במקרים של אירוע המוני כמו תאונה או פיגוע וכו'. במצב כזה קרוביו של אותו אדם יכולים להעלות לאפליקציה תמונות עדכניות של הנעדר וכך כאשר בבית החולים ינסו לזהותו ע"י צילום פניו הוא ימצא במאגר וכך יתאפשר הזיהוי.

בבית החולים כאשר הגיע חולה לא מזוהה על הצוות להכניס את תמונתו ולבדוק אם הוא נמצא במאגר וע"י כך ליצור קשר עם משפחתו וקרוביו.

**הגדרת הבעיה האלגוריתמית:**

* זיהוי פנים אנושיות מתמונה.
* חילוץ הפנים מתמונה.
* השוואת הפנים לפנים מתמונה אחרת מהמאגר ובכך לזהות את האדם המבוקש.

לצורך כך עלי ללמוד לעומק על כל הנושא של זיהוי פנים מתמונה ועל השוואת פנים מתמונות שונות.

בהשוואת הפנים ישנה בעיה שלעיתים ישנם פערים בין הפנים בתמונה המקורית לבין התמונה העכשווית. ישנו פער גילאים של האדם בין התמונות השונות כמו כן בצילום חולה יתכן שפניו יהיו שונות מהמקור – עיניו עצומות או פרצופו פצוע, לפעמים יש עליו מכשירים שונים וכד'. עלי לנסות להתגבר על בעיה זו כדי להגיע לתוצאה הטובה ביותר.

בכתיבת האלגוריתם אני אמורה להשתמש בספריה OpenCV וכן בספריות נוספות לשם כך עלי ללמוד על הספרייה כיצד להשתמש בה בצורה יעילה.

נוסף על כך אני צריכה לחפש אלגוריתמים בתחומים של זיהוי והשוואת פנים שיעזרו לי בכתיבת האלגוריתם וללמוד עליהם כיצד עובדים ואיך אני אוכל להיעזר בהם. מתוכם עלי למצוא את האלגוריתם היעיל ביותר לצורך ביצוע המשימה.

לצורך שמירת המידע במסד נתונים עלי להשתמש ב- mongoDB  שהוא בסיס נתונים הנמצא בקטגוריית NoSQL  ועובד במבנה של מסמכים (בשונה מבסיסי נתונים טבלאיים כמו SQL ועוד). לשם כך עלי ללמוד על בסיס נתונים זה כיצד עובד וכיצד אוכל להשתמש בו.

**רקע תאורטי בתחום הפרויקט:**

האפליקציה יעילה לבתי חולים. במקרה של חולה חסר הכרה שמגיע לבית החולים ללא זיהוי ניתן לצלם את תמונתו ואם הוא נמצא במאגר יהיה ניתן לזהותו בקלות.

יתר על כן ישנה אפשרות של זיהוי בזמן אמת. במקרים שבהם נעדר אדם ובני משפחתו חוששים אולי הוא נמצא בבית החולים ניתן להעלות לאפליקציה תמונות עדכניות שלו בתוספת פרטים מזהים וכך כשבבית החולים ירצו לזהותו זה יתאפשר בקלות רבה.

**תהליכים עיקריים בפרויקט:**

1. משתמש רוצה להירשם לאפליקציה. מזין את פרטיו האישיים ותמונת פרופיל שלו.
2. צוות בית החולים מעלה תמונות למאגר במקרה של מטופל אלמוני.
3. משפחה המחפשת נעדר מעלה את תמונתו כדי לבדוק האם הוא הגיע לבית החולים.
4. האפליקציה מזהה את תמונת האדם במקרה שמגיע לבית החולים ללא זיהוי.
5. האפליקציה מודיעה לבית החולים לגבי ההתאמה. באחריות בית החולים ליצור קשר עם משפחת החולה.

**תיאור הטכנולוגיה:** שרת REST, SPA / WINFORMS

**צד שרת:**

שפת תכנות בצד השרת: python

**צד לקוח:**

שפת תכנות בצד לקוח: ionic angular

**מסד נתונים:** mongoDB

**פרוטוקלי תקשורת:** אין**.**

**לוחות זמנים:**

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

**חתימת הסטודנט:**

**חתימת רכז המגמה:**

**אישור משרד החינוך:**

# **מבוא / תקציר**

## **הרקע לפרויקט**

כשהתחלנו לחשוב על רעיון לפרויקט רציתי רעיון שיהיה לי מעניין לעשות לא רציתי לעשות פרויקט שלא יעניין אותי, העדפתי להשקיע בפרויקט שיכול להיות שיהיה קשה יותר אבל גם מהנה.

אחת ממטרות הפרויקט היא פיתוח מיומנויות חשיבה, להביא אותנו להתנסות עצמית. כשאני יושבת על פרויקט משלי שאני אמורה לעשות לבד מתחילתו ועד סופו אני אמורה ללמוד לבד חומרים שלא למדנו בכיתה, לנתח פרויקט ולהבין מה אני צריכה לבצע, להכיר קודים ואלגוריתמים שונים, לנסות ולהתנסות. וזה בעצם חלק ממטרות הפרויקט – להביא אותי בסופו של דבר למוכנות לעבודה כי הרי לימודים ועבודה זה עולמות שונים.

רציתי לנצל את הפרויקט כדי ללמוד עוד על תחומים שלא יצא לי להכיר ואכן זה מה שגם גרם לבחור דווקא את התחום של הזיהוי פנים – התחום נשמע לי מעניין, מרתק ושונה. מעולם לא יצא לי להתעסק בתחום הזה לפני כן וניצלתי את ההזדמנות ללמוד ולהכיר דברים חדשים. ואכן היה לי הרבה מה ללמוד. אני מרגישה שהפרויקט הרחיב לי אופקים התחום הצריך הרבה למידה עצמית, התנסות וכישלון וזה הביא אותי לרמה הרבה יותר גבוהה.

לאחר שהחלטתי על נושא לפרויקט וחקרתי אותו קצת הבנתי עד כמה הוא חשוב ונצרך. כל העניין הזה של איתור נעדרים הוא דבר שכמעט ולא קיים ויכול לתרום הרבה לאנושות ברמה שיכול ממש להציל חיי אדם במקרי היעדרות של מבוגרים או ילדים קטנים. החלטתי שאני מאוד רוצה לקחת דווקא את הנושא הזה כי אני רואה חשיבות הרבה ונחיצותו.

הדבר שהכי חיזק בי את ההחלטה לבחור ברעיון של איתור נעדרים כרעיון לפרויקט גמר הוא הסיפור הטרגי שארע במירון שנה שעברה. אני חושבת שאם הייתה קיימת כזאת אפליקציה הייתה נחסכת המון עוגמת נפש ולחץ של אימהות דואגות וקרובי משפחה שחיפשו את בני משפחתם ואפילו הגיע מצב של אימהות שהיו בטוחות שילדיהם בין ההרוגים ואפליקציה מעין זו יכלה להוריד במעט את הדאגה העצומה של אלפי אנשים.

גם ביום יום לאו דווקא באירועים המוניים וחריגים אפליקציה זו יכולה להציל חיים. סיפורים על מבוגרים חסרי ישע נעדרים וילדים קטנים שהולכים לאיבוד מתרחשים לצערנו בכל יום ולכן אפליקציה כזאת היא דבר נחוץ והכרחי. לאחר שתפותח אפליקציה כזאת מקרים רבים של היעדרות יפתרו במהירות ולא יצטרכו להמתין ימים שלמים עד שהמשטרה תעזור בעניין וימצא הנעדר.

השם שבחרתי לפרויקט הינו Relatiive- השם מורכב מצירוף המילים relate שמשמעותה לייחס, לשייך וrelative- שמשמעותה קרוב משפחה.

לסיכום אני מאוד מקווה שהפרויקט יועיל ויעזור לשיפור האנושות, אני חולמת שאנשים באמת יוכלו להשתמש באפליקציה והיא באמת תהיה יעילה ושימושית. בנוסף הקוד יהיה קוד פתוח ואנשים רבים יוכלו להיעזר בקוד ולהשתמש בו לצרכיהם הפרטיים.

כאן יהיה כתוב כל מה שהביא אותך לפתח את הפרויקט.

* איזה סוג של רעיון חיפשת לפרויקט?
* איך בחרת את הרעיון שלך?
* למה את חושבת שהוא טוב בתור רעיון לפרויקט גמר?
* למה את חושבת שיש בו תועלת?
* למה מומלץ לפתח את מה שאת מפתחת?
* מה קורה כיום – לפני הפיתוח?
* מה יהיה לאחר שתפתחי?
* איזה אלגוריתם עוזר לך לפתח את הרעיון שלך?
* מה השם לפרויקט שלך?
* סיום יפה ואופטימי כהקדמה לספר.

## תהליך המחקר

כשהתחלתי לפתח את הפרויקט דבר ראשון הלכתי לחפש חומרים קראתי הרבה מאמרים בנושאים שקשורים (יותר או פחות) לפרויקט. חקרתי והכרתי לעומק את כל הנושא של זיהוי והשוואת פנים – איך זה מתבצע, באיזה ספריות משתמשים, איזה אלגוריתמים קיימים בתחום וכו'. בהמשך שראיתי שאני יודעת הרבה על הנושא התחלתי לעבוד על הפרויקט בפועל וליישם את כל מה שקראתי ולמדתי.

את החומר הרב שקראתי מצאתי בגוגל באתרי למידה ומידע שונים כמו: Internet Israel, Stack Overflow וכד'. כמו כן ראיתי סרטונים רבים בנושאים של זיהוי פנים. לאחר החומר הרב שקראתי החלטתי להשתמש בספריה .OpenCV

בשלב מסוים התייעצתי עם מתכנת המתמחה בתחום של עיבוד תמונה ומכיר את הספרייה OpenCV , הוא הסביר לי קצת על הספרייה וכיצד להשתמש בה ונתן לי רעיונות נוספים לפיתוח מערכת לזיהוי פנים.

OpenCV הינה ספריה המספקת למתכנתים פלטפורמת פיתוח ובה יכולות מתחום הראיה הממוחשבת. ייחודיותה הוא יכולתה לעבודה גם בזמן אמת. הספרייה פותחה במקור ע"י אינטל. היא מאופיינת בכך שהיא cross platform – חוצה פלטפורמות. השימוש בספרייה הוא חינמי לחלוטין תחת רישיון Apachi2.

יכולות עיקריות בספרייה: יכולת עיבוד תמונה, זיהוי אובייקטים, מעקב תנועה וזיהוי חיוויים אנושיים.

בספריית OpenCV השתמשתי ב- EnguCV שהיא פלטפורמת עטיפה ל- OpenCV המאפשרת גישה לפונקציות של OpenCV מתוך פרויקט .net

הסיבה שבשלה החלטתי להשתמש בספריית OpenCV בפרויקט היא עקב הצורך בשימוש בטכנולוגיות מתחום עיבוד התמונה וככזה OpenCV הוא כלי מהותי והכרחי לשימוש.

## סקירת ספרות

האתרים בהם השתמשתי לביצוע הפרויקט הם:

חקר הפרויקט –

* ויקיפדיה
* Microsoft
* Internet Israel
* OpenCV
* DataCamp

אלגוריתם –

* GeeksForGeeks
* Stack OverFlow
* GitHub

שפות –

* MongoDB
* Medium
* Dwh.co.il

עיצוב –

* Bootstrap
* W3School

# מטרות ויעדים

כ-250 בני אדם נעדרים בכל יום בישראל, בין אם אלו קשישים עם דמנציה, אנשים הסובלים מאוטיזם ונחשבים בתפקוד נמוך או ילדים שהלכו לאיבוד.

למטרה זו נועדה Relatiive - לעזור לאנשים שנראים חסרי ישע או איבדו את דרכם וכן ילדים קטנים המשוטטים לבדם, כדי לנסות לחבר אותם למשפחה המודאגת.

בבניית מערכת ובמיוחד מערכת כזו שמטרתה איתור נעדרים - אחד הדברים החשובים הוא להקפיד על מערכת ברורה ונוחה למשתמש. לרוב האנשים אין את הזמן והכוח לנסות להבין איך המערכת עובדת וכיצד להשתמש בה ולכן על המתכנת לבנות את המערכת בצורה שיהיה למשתמש קל ונוח להתעסק בה כמובן שחשוב שהמערכת תעבוד בצורה יעילה ונכונה שהרי היא נועדה לזיהוי נכון של נעדרים.

לשם הקמת המערכת וכדי שהיא תתפקד בצורה המיטבית הצבתי לעצמי כמה מטרות ובשביל כך כמה יעדים:

**מטרות המערכת:**

1. אפשרות לאיתור נעדרים בצורה מהירה ויעילה.
2. נוחות וידידותיות למשתמש.
3. לחסוך התערבות משטרתית בכל אירוע של היעדרות.
4. לאפשר לבתי חולים לזהות חולים לא מזוהים המגיעים לבית החולים חסרי הכרה.

**יעדי המערכת:**

1. בניית מערכת חכמה לזיהוי פנים אנושיות.
2. הוספת אפשרות ליצירת חשבון אישי ובו העלאת תמונה שתאפשר הגנה על המשתמש במקרה היעדרות.
3. יצירת המערכת כאפליקציה המפותחת בצורה נוחה וקלה למשתמש.
4. בית החולים יוכל להעלות תמונה של החולה (כאשר לא מדובר במצב של שינוי דרסטי בפנים כתוצאה מפגיעה או פציעה), וכך לנסות לגלות את זהותו.

# אתגרים

כשניגשתי להתחיל את הפרויקט ידעתי שהוא לא הולך להיות קל אבל לקחתי את זה בחשבון. ידעתי שאם אני רוצה להצליח אני חייבת להתאמץ ולהשקיע אפילו שיהיה קשה.

בתחילת תכנון הפרויקט ושלבי העבודה לא ידעתי בכלל מאיפה להתחיל, הפרויקט היה נראה לי אתגר ענק ובלתי אפשרי, היו המון נושאים שלא הכרתי ומעולם התעסקתי בהם. עכשיו לאחר בניית הפרויקט אני מבינה שהכול אפשרי וגם דברים שנראים בלתי אפשריים בסוף עבירים.

לצורך שמירת התמונות הייתי צריכה להשתמש במסד נתונים השונה מ SQL – שאותו למדנו בכיתה ולכן הייתי צריכה ללמוד על mongoDB. הדבר היווה לי מעט קושי מכיוון שהיה עלי לחפש חומרים שיעזרו לי ללמוד על הנושא, להכיר אותו ולהצליח בסופו של דבר לעבוד עם זה ולשמור בו את הנתונים.

כשהתחלתי בכתיבת האלגוריתם לזיהוי פנים מצאתי קוד שהיה אמור לעזור לי בכך אך לאחר עבודה קשה ומאומצת התברר שהקוד לא הועיל במאום והיה עלי לחפש קוד חלופי אחר.

בהמשך באימון המודל להשוואת פנים

(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 אכן הצליחה והוכיחה את נכונותה במקרה שבו יש זיהוי של לפחות 60% הצלחה וכן ההשוואה הצליחה לפחות ב - 60% מהמקרים.

# 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 ההפעלה: 10 Windows

שפות תוכנה: #C , תוך שימוש בטכנולוגית WebApi, אנגולר .

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

מסד נתונים:.mongoDעמדת משתמש מינימאלית :

* + חומרה: מעבדRAM 4GB i5 .

o מערכת ההפעלה: Windows 7 ומעלה.

* + חיבור לרש ת: נדרש .
  + Chrom: 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 טכניים 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

משתמש יכול ליצור לעצמו חשבון ובו הוא יוצר הגנה על עצמו במקרה של היעדרות חלילה.

המשתמש מזין את פרטיו האישיים כנדרש ומעלה תמונה עדכנית שלו למקרה הצורך.

כאשר נמצא ילד קטן לבדו, אדם מבוגר ללא זהות או חולה בבית החולים חסר הכרה יש לצלמו ולהעלות את התמונה לRelatiive - ויש סיכוי שאם הוא מוגן באפליקציה תמצא התאמה ותתברר זהותו.

ישנה אפשרות נוספת למציאת אבודים גם אם אין להם חשבון באפליקציה ואין עליהם הגנה: במקרה של היעדרות בני המשפחה או קרובים רוצים לאתר את הנעדר, הם יכולים להעלות תמונות של הנעדר בזמן אמת ולנסות לבדוק אולי מישהו מצא את הנעדר וצילם אותו וכך תימצא התאמה.

## אילוצים

המערכת מבצעת את ההתאמה במקרה שהתמונה עדכנית וברורה. לא תתבצע התאמה במקרים שבהם ישנו שוני בפנים ולא ניתן לזהותו כמו פצוע שמגיע לבית החולים והוא חבוש או פצוע בפנים. האפליקציה לא באה לגבור על היכולות האנושיות ולכן במקרה שאפילו אדם לא יכול לזהות כמובן שהאפליקציה לא תוכל לבצע את הזיהוי.

# תיאור הארכיטקטורה

## 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 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 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 לבנייתApplications 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 רעיונות 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 מידית.

* EntityFramework - ה

קורא נתונים מה DataBase שנכתב בשפת mongoDB. למסדר מתונים של mongoDB 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 התלמידים

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 ממסמכים מבנה המסמכים עובד מעל מימוש של JSON הנקרא על ידי mongoDB – BSON/
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 השרת.
6. angular, TypeScript צד לקוח

## ארכיטקטורת רשת (לא רלוונטי )

## תיאור פרוטוקולי התקשורת

http – ראשי תיבות של המילים - HypertextTransferProtocol הוא [פרוטוקול תקשורת](https://he.wikipedia.org/wiki/%D7%A4%D7%A8%D7%95%D7%98%D7%95%D7%A7%D7%95%D7%9C_%D7%AA%D7%A7%D7%A9%D7%95%D7%A8%D7%AA) שנועד להעברת דפי [HTML](https://he.wikipedia.org/wiki/HTML)  ואובייקטים ברשת ה[אינטרנט](https://he.wikipedia.org/wiki/%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98) וברשתות האינטראנט. הפרוטוקול פועל ב[שכבת היישום של מודל ה-OSI](https://he.wikipedia.org/wiki/%D7%A9%D7%9B%D7%91%D7%AA_%D7%94%D7%99%D7%99%D7%A9%D7%95%D7%9D_%D7%A9%D7%9C_%D7%9E%D7%95%D7%93%D7%9C_%D7%94-OSI)  וב[שכבת היישום של מודל TCP/IP](https://he.wikipedia.org/wiki/%D7%A9%D7%9B%D7%91%D7%AA_%D7%94%D7%99%D7%99%D7%A9%D7%95%D7%9D_%D7%A9%D7%9C_%D7%9E%D7%95%D7%93%D7%9C_TCP/IP) ..

## שרת – לקוח

צד השרת נכתב בטכנולוגית WebApi ובשפת c#.

צד הלקוח נכתב בשפות: HTML, scc, TypeScript בטכנולוגית Angular.

## 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

בפרויקט השתמשתי במבנה הנתונים BitMap שהוא מטריצה. השימוש בו היה בזיהוי הפנים...

## תרשים מחלקות

**שכבת ה-DAL**:  
צילום של פירוט המחלקות:  


**שכבת ה-BL**:

**שכבת ה-API**:

## תיאור המחלקות

הפרויקט מחולק ל- 3 שכבות. כל שכבה אחראית על תחום מסוים בפרויקט.

**DAL –** השכבה הנמוכה ביותר. שכבה זו אחראית על התקשורת עם ה-Data Base. בשכבה זו פונקציות שונות המפעילות את ההתקשרות.

**המחלקות הקיימות בשכבה:**

* **DBSetting –** מחלקה זו משמשת להתקשרות בפועל ל – Data Base. המחלקה מכילה את הניתוב ל - Data Base, שם ה – Data Base, ושם ה – Collection.
* **User –** מחלקה זו מייצגת משתמש במערכת. מכילה את תכונות המשתמש: שם פרטי, שם משפחה, מספר זהות, מייל, פלאפון וסיסמה – כולם מסוג String.
* **UserService –** במחלקה זו קיימות פונקציות האחראיות על פעולות המשתמש:

GetCollection **–** החזרת רשימת כל המשתמשים**.**

Insert– הוספת משתמש חדש למערכת.

Update – עדכון נתוני משתמש.

Delete – הסרת משתמש מהמערכת.

**BL –** השכבה שמעל ה – DAL היא מקשרת בין ה - DAL ל -API . השכבה אחראית על כל החלק הלוגי של המערכת.

**המחלקות הקיימות בשכבה:**

* **userManager –** במחלקה קיימות פונקציות שמשמשות לניהול המשתמשים במערכת:

GetUsers–הפונקציה מחזירה משתמש מסוג List.

SignUp – פונקציית הרשמה למערכת.

SignIn – פונקציית התחברות למערכת. הפונקציה מוודאת שהמספר זהות והסיסמה אכן תקינים.

**API –** שכבה זו אחראית על החיבור בין צד השרת והלקוח. בשכבה זו קיימים קבצי מערכת רבים, קבצי התקנות, סקריפטים וכו'. בנוסף בשכבה זו קיימים Controlers – בקרים האחראים על ניתוב התקשורת בין השרת והלקוח:

* **UserController –** במחלקה זו קיימות הפעולות:GET, POST, הרשמה, התחברות, הוספה, מחיקה, LoadPictures– מטרתה לטעון תמונה המתקבלת מהלקוח.

# תיאור התוכנה

* + סביבת עבודה:

Visual Studio Codeו Visual Studio

* + שפות תכנות:

צד השרת נכתב בטכנולוגית WebApi ובשפת c#.

צד הלקוח נכתב בשפות: HTML, Css, TypeScript בטכנולוגית Angular.

# 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 ואופטימלי. 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 נתונים שונים, האלגוריתם הגיע לקירוב האפשרי ביותר בכלים העומדים לרשותי. )

# ניתוח יעילות

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

# אבטחת מידע

הכניסה לחשבון האישי חייבת להתבצע באמצעות הזנת שם משתמש וסיסמא. כך משתמש יכול להיות רגוע שהמידע אודותיו שמור במערכת וחסוי.

באופן כללי כניסה למערכת לא מחייבת התחברות באמצעות סיסמא אך כל פעילות במערכת של העלאת תמונה או עדכון פרטים אישיים מחייבת התחברות או הרשמה.

# 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

בהמשך כשיהיה לי את הזמן והאפשרות הייתי מאוד רוצה לשכלל את הפרויקט. שהזיהוי פנים יתבצע בצורה יותר מדויקת ומרבית, שההתאמה תהיה יותר מלאה והמערכת תעבוד בצורה יותר מהירה ומדוייקת.

# ביבליוגרפיה

* GitHub
* StacK Overflow
* Bootstrap
* Internet Israel
* ZetaCode
* MongoDB
* W3School
* GeeksForGeeks
* ויקיפדיה