פרויקט גמר

מגישה:

chaya avichathira

סמל

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 תשפ''ב**

סמל מוסד: 560342

שם מכללה: סמינר תורני בית יעקב.

שם הסטודנט: חיה שבה אביחצירא.

ת.ז. הסטודנט: **322419276**   
שם הפרויקט: שיבוץ חדרי ניתוח.

תיאור הפרויקט:

המערכת תתכנן לוח זמנים לביצוע מקביל של מספר ניתוחים.   
המערכת תתחשב בחשיבות ובנחיצות הניתוחים, בלוחות הזמנים של הרופאים הנמצאים בבית החולים, בזמני החיטוי בין ניתוח לניתוח, ותבנה לוח זמנים אידיאלי להספק הרב ביותר בזמן הקצר ביותר.   
כדי להיכנס למערכת יש להזין תעודת זהות של אחות האחראית על חדרי ניתוחים, כאשר הכניסה התבצעה בהצלחה תינתן האפשרות ליצור לוח זמנים חדש, לצפות בלוח זמנים קיים או להכניס ניתוח חירום.  
במקרה של יצירת לו"ז על האחות להכניס את פרטי הניתוח:  
רופא, מחלקה, רמת עדיפות וסיכון והאם הוא זקוק למכשירים מיוחדים, כך לכל הניתוחים והמערכת תשבץ אותם בצורה האופטימלית הכי אפשרית.   
במקרה של ניתוח חירום על האחות להכניס את פרטי הניתוח כמו לעיל במקרה כזה יתכנו שתי אפשרויות:  
1.יש חדר פנוי - הניתוח ישתבץ בחדר הפנוי.  
2. אין חדר פנוי - המערכת תעדכן את לו"ז מחדש ותשבץ את כולם בהתאם.  
כמובן שתינתן גם אפשרות לצפות בלוח הזמנים הנוכחי של השבוע ולשנותו במידת הצורך.

הגדרת הבעיה האלגוריתמית:   
שיבוץ לוח זמנים האידיאלי ביותר תוך התחשבות באילוצים השונים (כגון: זמני הניתוח, חדרי ניתוח עם מכשירים מיוחדים שאינם יכולים להיות בשימוש מקביל, ניתוחים בחשיבות עליונה שנכנסים מחדרי מיון וחדרי טיפול נמרץ וכדו'). כמובן שבעיית שיבוץ היא בעיה בלתי פתירה, והמטרה היא להגיע לתוצאה הטובה בקירוב.  
כיוון שיש מספר חדרי ניתוח ומספר ניתוחים וכל ניתוח צריך חדר ניתוח כדי להתבצע, לכן אני אשתמש לפתרון הבעיה בשיבוץ הונגרי שתפקידו לשבץ את הדברים בצורה הנכונה ביותר כדי להגיע אל התוצאה הטובה ביותר המתואמת והמשובצת על הצד הכי טובה.  
הרעיון בשיבוץ הונגרי המקורי שיש כמה אפשרויות בחירה ומכל אחד מהם בוחרים את האפשרות הטובה ביותר כדי להגיע לפתרון הזול ביותר גם כאן מתוך כמה אפשרויות תבחר האפשרות בעלת הסיכון הגבוה ביותר קודם כל ולאחר מכן יסונכרן בין רמת הסיכון והעדיפות כדי להציל כמה שיותר חיים ולהועיל לכמה שיותר אנשים.

רקע תאורטי בתחום הפרויקט:   
כשמשבצים ניתוחים צריך להתחשב בכמות חדרי הניתוח שיש בבית החולים, בלוח הזמנים של הרופאים שנמצאים בבית החולים כרגע ויכולים לנתח, בחשיבות הניתוח וכן במקרים שחדרי ניתוח נלקחים לטובת טיפול נמרץ וחדרי מיון, חדרי הניתוח מלאים וצריך לשלוח לבית חולים אחר או שחדרי הניתוח ריקים ואפשר לשבץ ניתוחים של מטופלי חוץ ומטופלים מהמחלקות תוך כדי שימת לב לא ליצור מקרים שבהם ימתינו לניתוח זמן רב.  
בכל בית חולים קיימים מספר רב של חדרי ניתוחים. לכל מחלקה כמה חדרים שבהם יש מכונות ומכשירים המותאמים למחלקה. בכל בית חולים יש צורך לבצע מספר גדול של ניתוחים ואין מספיק חדרים, לכן תמיד יש צורך לתעדף את הניתוחים לפי רמת סיכון ורמת עדיפות, כדי שלא יקרה מצב שבו יכנסו ניתוחים בעלי סיכון ושאר הניתוחים יוזנחו או שיכניסו סתם ניתוחים בלי להתחשב ברמת הסיכון ויתכנו סיכוני חיים. כדי למנוע מצב שניתוח שלא מסכן חיים לעולם לא יקבל את תורו, כל פעם שניתוח יתעכב בשבוע רמת העדיפות תעלה (כדי לא ליצור הרעבה).  
כמובן כאשר נכנסים ניתוחי חירום או מקרי חירום אין זה משנה באיזה חדר הניתוח יתבצע (מבחינת מחלקה) והוא ישתבץ מידית תוך עדכון כל הלוח הזמנים בהתאם.

הפתרון לכך הוא תכנון ושילוב נכון של חדרי הניתוח תוך התחשבות בנתונים השונים.

תהליכים עיקריים בפרויקט:

1. המערכת תקבל את לוחות הזמנים של הרופאים, חדרי הניתוח, מטופלים וכו'.
2. המערכת תשבץ את סדר ביצוע שלבי הניתוח, החיטוי כך שיסתיימו בזמן הקצר האפשרי.
3. המערכת תתחשב באפשרות שיכנסו ניתוחים לא מתוכננים וידחו את כל הלוח זמנים המתוכנן ותעדכן אותו מחדש.

תיאור הטכנולוגיה:

צד שרת:

**שפת תכנות בצד השרת:** C#

צד לקוח:

**שפת תכנות בצד לקוח:** angular .

מסד נתונים**:** SQL Server .

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. בנית התוכנה – ינואר, פברואר
7. בדיקות – מרץ
8. הכנת תיק פרויקט – אפריל
9. הטמעת המערכת – מאי
10. הגשת פרויקט סופי - מאי

חתימת הסטודנט: חיה שבה אבוחצירא

חתימת הרכז המגמה:

אישור משרד החינוך:

# מבוא / תקציר

## הרקע לפרויקט

מהרגע שנכנסים למגמת הנדסת תוכנה ישנו דבר אחד שתמיד מרחף באוויר ומחכים לו פרויקט ההגשה, פרויקט שיתרום לנו כל כך הרבה התנסות ובעצם יעשה לנו הכנה לעבודה בחברות עם התמודדויות שלא נתקלים בהם במשך הלימודים ,באגים שצריך לפתור לבד, חיפוש מידע שנצרך לפרויקט, חשיבה עצמאית של איך לפתח את הרעיון מבחינה שיהיה יעיל ויעזור לאנשים וגם שיהיה עם אלגוריתם טוב וחזק שיביא למימוש ומיצוי הלימודים שלנו.  
המחשבה שקדמה לבחירת הרעיון שלי היתה שילוב של שני תחומים שמענינים אותי הנדסת תוכנה-המקצוע שלמדתי ורפואה-תחום שתמיד ענין אותי ואז עלה לי הרעיון על שיבוץ ניתוחים, במקום לשבץ ניתוחים באופן ידני ולבדוק שיש את כל המכשירים הנצרכים לניתוח ובמקרה חירום להזיז את כל הניתוחים, ולבדוק שכל השינויים שנעשו יתאימו לכולם ולא נשכח אף אחד דבר שמסובך יותר אם מדובר בכמה שינויים שנעשעים בטווח קרוב, ולא נוצר מקרה שבו יהיו כפילויות וטעויות אדם, חשבתי על שיבוץ הניתוחים על ידי אלגוריתם כך שיבדוק את ההתאמה הטובה ביותר בין חדרי הניתוח לניתוח המתבצע, המכשירים הנצרכים שיתן עדיפות לניתוחי חירום ומצד שני לא ירעיב את שאר הממתינים , שהשיבוץ יעשה במהירות ולא יצטרכו להשקיע זמן ומחשבה בכל שינוי .  
האלגוריתם שמתאים לי לפרויקט הוא צריך להיות אחד שבודק את כל האפשרויות שמתאימות ובוחר את הטובה מבין כולם- בדיקת התאמה של הניתוחים וחדרי הניתוח ציון שינתן לכל אחד מהם ובחירת ההתאמה הטובה ביותר לכן האלגוריתם שבחרתי הוא שיבוץ הונגרי.  
הפרויקט נקרא בשם ""scudge and surge משחק מילים על המילים שיבוץ וניתוח באנגלית.  
"- "scudge and surgeאפליקציה שתסייע לבתי חולים לשיבוץ ניתוחים בדרך הטובה ביותר ותסייע להצלת חיי אנשים בהמשך הספר אפשר יהיה לראות את השילוב הלוגי ביחד עם עיצוב נעים לעין ,מידע נרחב על האלגוריתם ועל דרך הפיתוח.

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

בתור התחלה שאלתי הרבה שאלות על הנושא כדי לפשט אותו כמה שיותר ושיהיה לי ברור על מ אני צריכה לעבוד   
איך קובעים באיזה חדר מנתחים?  
לפי מה קובעים איזה ניתוח מתבצע לפני?  
מה קורה במקרי חירום?  
כיצד כיום משבצים ניתוחים?  
ועוד שאלות שעלו לי במהלך המחקר ואז התחלתי לחפש להם תשובות אם זה היה בחיפוש באתרים שונים או בהתייעצות עם אנשי מקצוע שונים אחיות, מזכירות וכדו'  
לאחר שקיבלתי את השתובות לשאלות ידעתי מה אני צריכה לעשות אבל היה חסר לי איך לפעול מלבד ידע בסיסי שמדובר בשיבוץ.  
ערכתי חיפוש על אלגוריתמים של שיבוץ וקיבלתי כמה אפשרויות   
שיבוץ גנטי- שיבוץ שנעשה על ידי שלוקחים שני פרטים מתאימים בינהם ומהם יוצרים מוטציה ומתאימים שוב כך עד שמגיעים לתוצאה האפשרית המתאימה ביותר.

שיבוץ הונגרי- שיבוץ שנעשה על ידי כך שיש שני סוגי דברים שצריך להתאים בינהם ומתאימים כל פריט עם כל פריט לדוגמא ישנם שלושה אחים: אפרים, מאיר וישראל. אחד מהם צריך לנקות את חדר השינה, שני לטאטא את החצר ושלישי לשטוף את החלונות. כל אחד מהם דורש תשלום שונה עבור כל משימה. המטרה היא למצוא את השיבוץ שייתן את העלות הנמוכה ביותר לביצוע שלוש המשימות  
ואז הבנתי שהאלגוריתם המתאים לי ביותר הוא אלגוריתם הונגרי רק שבפרויקט הזה צריך ציון מקסימלי.

* מה הדבר הראשון שעשית כשרצית להתחיל לפתח את הפרויקט?
* באיזה כיוונים ניסית לברר לחפש? (חיפוש חומרים / אלגוריתמים / בירור מידע אצל אנשים / מפתחים שונים / אנשי עסק / ספרות )
* מה מצאת? (איזה אלגוריתם / טכנולוגיה / ספריה )
* תסבירי קצת על מה שמצאת. (הסבר על האלגוריתם או על הספריה כיצד היא עובדת)
* למה דווקא מה שבחרת מתאים לפרויקט?

## סקירת ספרות

מידע רב על הנושא והאלגוריתם מצאתי באתרים ונעזרתי בהם רבות  
חומר לימודי-המכלול  
ויקיפדיה  
אתרי קוופ"ח ובתי חולים שונים  
וכן באנשי מקצוע  
אלגוריתם-ויקיפדיה  
stackoverflwe  
github  
צד שרת ולקוח- angular io  
Microsoft  
stackoverflew  
github  
לעיצוב השתמשתי בbootstrep

באיזה אתרים נעזרת ל- (תיאור במשפטים)

* אלגוריתם
* שפות, שרת, לקוח
* עיצוב

# מטרות ויעדים

מטרת בחירת הפרויקט נבעה מכמה שיקולים כמו התנסות מקצועית- בפרויקט שבביצוע שלו דומה לעבודה אמיתית עם ההתמודדות עצמית של קשיים ובאגים.  
מטרת ידע- להרחיב את האופקים על ידי בחירת הנושא שעוסק ברפואה נושא שמאד מענין אותי אישית וחקירה עליו ואיך הוא פועל  
וכמובן שהמטרה הייתה לפתח לא סתם עוד אפליקציה אלא כזו שתתרום ותעזור לאנשים ואולי אף תציל חיים.

מטרת העל היא אפליקציה שתעשה שיבוץ באופן מהיר קל ומזמין למשתמש בלי להתפשר על מקצועיות ובחירת התוצאות הטובות ביותר.  
המערכת תקבל רשימת ניתוחים או אחד בכל פעם לפי הצורך ותשבץ אותם בהתאם באופן הטוב ביותר ותוציא למשתמש לוח זמנים מאורגן ונעים לעין.

חלקי למטרות ויעדים.

מה הביא אותך לרצות לעשות את הפרויקט ( מטרות אנושיות, מטרות של ידע, מטרות של התנסות מקצועית)?

מה מטרה המרכזית/מטרת העל?

מה המטרות הנוספות? (לתת מידע בנוחות, לאפשר גישה ל... ליצור עיצוב נח...לאפשר, להקל על..)

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

# אתגרים

תחילה מאד התפזררתי עם הרעיון של מה האפליקציה תעשה וכל הזמן זה לא היה לי עם קווים ברורים עד שישבתי עם עצמי לכמה שעות להיסגר בדיוק על מה הוא עושה,  
ואז היה צורך לנתח את האפליקציה לרמה של טבלאות-כלומר איז טבלאות יהיו בדאטה בייס שעליו היא תרוץ,ולא לשכוח לעבוד לפיהם.  
לעשות מראה אפליקציה- איך אני רוצה שהאפליקציהה שלי תראה באופן הסופי  
דברים אלו היו טיפה קשים בכללי כיוון שזהו סוג של חיזוי העתיד וקרו כמה וכמה מקרים שטבלאות מסוימות היו מיותרות או להפך היה צורך בלהוסיף טבלאות נוספות דר שגוורם שוב לחישו מסלול מחדש.  
בנוגע לאלגוריתם הייתי צריכה להתייעץ איזה מהם הוא הטוב לי ביותר עם מנחת הפרויקטים אבל המחשבה העיקרית שהושקעה זה איך לבצע בפועל את האלגוריתם זה בהחלט היה מאתגר איך לחשב ולהגיע לציון הסופי של כל אחד.  
תחילה חשבתי שהפרויקט יהיה קל אך נתקלתי בקשיים ובבאגים שלקחו כמה שעות עד לפתירתם הסופית ,בעיקר בבנית המטריצה עליה ירוץ האלגוריתם ומה הפרמטרים שלפיהם היא תבנה , הבנת האלגוריתם בו אני משתמשת עד למצב שהכל נאור וברור לי והתאמת האלגוריתם לפרויקט שלי.

(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 אכן בסופו של דבר השיבוץ יעבוד-כלומר כל הניתוחים משובצים בחדריםעם ההתאמה המקסימאלית הכי אפשרית.  
כמובן שיש כמהה השגים נוספים שארצה לאגיע אליהם אך הפם פחות עיקריים לתלות בהם את הצלחת האפלקציה:  
שיהיה גישה אך ורק למורששים,  
שבסופו של דבר יווצר לוח שנה מסודר עםם הניתוחים המובצים,  
שבניתוח חירום יהיה אפשרות כניסה מידית לניתוח והתאמת כל השינווים שנעשו  
ועוד כמה נקודות חשובות.

(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 לאחר בירורי שערכתי עם אנשים שעובדים בתחום נודע לי שיש מקומות שעובדים בשיבוץ ידני כלומר יש מישהו שמכניס את הניתוחים למערכת ומוודא ששאכן אין כפילויות והכל מתאים,  
בחיפשוש אחר החומר מצאתי אפליקצי שככל הנראה עושה בדיוק את אותו הרעיון כמוו שלי אך אי אפשר לקבל עליה שום מידע כיוון ששהיא מוצעת לבתי חולים בתשלום

כיום המצב הוא.....

לאחר בירור אצל אנשים שונים....

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

# רקע תאורטי

האלגוריתם המרכזי בפרויקט הוא שיבוץ הונגרי וכיצד הוא בעצם עובד?  
אלגוריתם ששיבוץ הונגרי עובד על מטריצה שה ישש ששני מאפיניםוכל ריבוע במטריצה מהווה ניקוד מסוים של התאמה בין שני המאפינים באופן שכל אחד מושווה עם כל אחד   
כדי להבין זזאת היטב הנה דוגמא פשוטה:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **ניתוח 1** | **ניתוח 2** | **ניתוח 3** |
| **חדר 1** | 2 | 3 | 3 |
| **חדר 2** | 3 | 2 | 3 |
| **חדר 3** | 3 | 3 | 2 |

כאשר נריץ אץת האלגוריתם ההונגרי על מטריצה זו הוא יתן לנו את התוצאה המינימאלית הטובה ביווותר או המקסימלית לפי הצורך  
במקרה שאנו מחפשים את העלות המקסימאלית הטובה יותר בממטריצה זו השלב הבא יהיה לחסר מכל שורה ראת הסכוםהמקסימאלי פחות מה שיש בעמודה כלומר:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **ניתוח 1** | **ניתוח 2** | **ניתוח 3** |
| **חדר 1** | 1 | 0 | 0 |
| **חדר 2** | 0 | 1 | 0 |
| **חדר 3** | 0 | 0 | 1 |

לאחר מכן מחסרים מכל עמודה את הסכום המקסימאלי במקרה לעיל אפשר למצוא כבר מהשלב השני שני פתרונות אופטימאליים אך במקרה שלא מוצאים :  
מסמניםאת השורות שללא פתרון ואת העמודות שנמצאו בהם פתרון שהוא גם בשורות, וושלב סופי את השורות עפ שיבוץ בעמודות מסומנות מסמנים בקו את השורות הלא מסומנות ואת העמודות המסומנות מוצאים מינימאלי/מקסימאלי שלא מכוסה ומפחיתים מכל הלא מכוסים ומוסיפים לצמתים ואזז ישנו שיבוץ   
בשתי האפשרות לעיל התוצאה המקסימאלית היא 9,  
כך בעצם האלגוריתם ההונגרי עובד הוא רץ על מטריצת ציונים של התאמה בין ניתוח לחדר שנקבעת ממספר פרמטרים כמו רמת סיכון, עדיפות, מחלקה, מכשירים וכו' ומשבצת ניתוחים בחדרים על פי ההתאמה הטובה ביותר.

(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

ישנן אפשרויות שונות לפתרון הבעיה כמו:  
שיבוץ ידני באופן שרירותי על ידי גורם מוסמך-לא מחייב התאמה בין חדר לניתוח וחוסר במכשירים מסוימים  
שיבוץ כל האפשרויות-הגיוני כאשר מדובר בכמה ניתוחים אך כשאר מדובר בעשרות אן זה הגיוני  
המצאת אלגוריתם פשוט שיעבור בלולאה על כל החדרים הרקים והניתוחים שיש ופשוט יבץ אחד לשני  
אני בחרתי באפשרות של מעבר בלולאה .

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

המערכת שפיתחתי מקבלת רשימת חדים ורשימת ניתוחים ומשבצת ניתוחים בחדרים על פי ההתאמה ביותר שנבדקת על ידיי מכשירים, מחלקה,ועוד תוך התחששבות במקרי חירום.

הרעיונות הקודמים שהוצעו לא רלוונטים כיוון שהאפשרות של שיבות באופן שרירותי לא מתחבת בהתאמהה ולכן עדיף לעות זאת כבר באופן ממוחשב כיוון שייקח הרבה פחות זמן

והאפשרות השניה של בדיקת כל האופציות לא הגיונית כייון שאם מדובר בכמה חדרעים וניתוחים הדבר מתאפשר אך כאשר מדובר בדאטה בייס של בית חולים שיש בו כמה עשרות ניתוחים אין זה הגיוני לעשות זאת באופםן ידני לכן האפשרות שבחרתי גם אינה טובה אך הכי מועילה מבין השאר מקבלים רשימת חדרים ריקים ורשימת ניתוחים עוברים ומאימים אחד לשני באופן שרירותי.

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

הכנסת ניתוחים ופרטיהם על ידי צד לקוח

חיבור לצד שרת והזנתהנתונים בדאטהבייס

קריאה למערכת לערוך שיבוץ  
מציאאת השיבוץ האופטימאלי על די המערכת

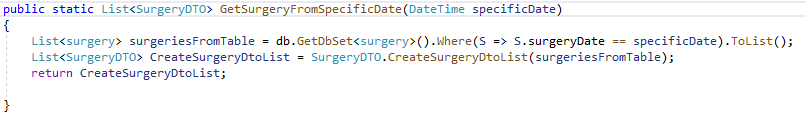
הזנת פרטי השיבוץ בדאטה בייס

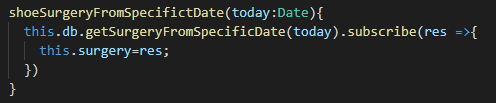
הצגת פרטי השיבוץ ללקוח

* 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

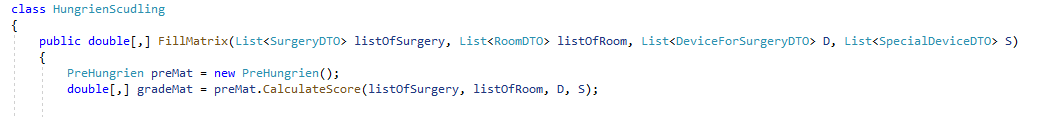
() GetSurgeryFromSpecificDate-

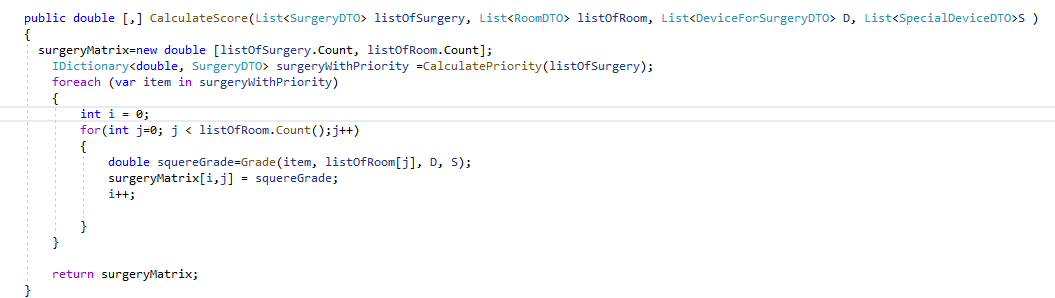
פונקציה זו מקבלת מפונקציה נוספת שנמצאת בצד לקוח שכתובה באנגולר:  
() shoeSurgeryFromSpecifictDate-



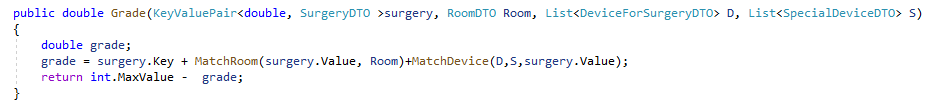
פונקציות אלו פועלות על ידי כך שמקבלים מהלקוח תאריך מסוים ששלפיו מוצגים כל הניתוחים שהוכנסו לדאטה בייס בתאריך זה ברשימה לפעמים הניתוחים מוצגים ללקוח ולפעמים נשלחים לפונקציה נוספת לפי הצורך.

בדומה לפונקציות למעלה נוצרות רשימות של חדרים, מכשירים הנצרכים לכל ניתוח ורשימת המכשירים הזמינים בבית החולים  
כל הרשימות לעיל נשלחות לפונקציה:  
() FillMatrix-

פונקציה זזו נמצאת ב class HungrienScudling שבו מתבצע השיבוץ בפועל אבל לפניהפונקציה בונה מטריצת התאמות כפי שהסברנו בסעיף 7 שעליה תרוץ ובעצם שולחת לפונקציה:

() CalculateScore-  
  


פונקציה זו עוברת רשימת מהניתוחים ונותנת התאמה של כל אחד מהניתוחים לכל אחד מהחדרים  
וכיצד היא נותנת ציון עפ"י הפונקציה:  
() Grade-

פונקציה זו מחשבת התאמה על פי רמת עדיפות וסיכון שנקבעה על ידי רופא, חדר תואם ומכשירים זמינים הנצרכים לניתוח כל אחד מהפונקציות האלו מחזירה ציון שמשתכלל לציון סופי ונכנס לריבוע המתאים של החדר והניתוח, לאחר שיש מטריצה מלאה היא חוזרת בחזרה לפונקציה הראשונה שקראה לה  
 () FillMatrix ואז האלגוריתם מתחיל לרוץ על המטריצה ולעשות שיבוץ בפועל

(פירוט פונקציות עיקריות ותפקידן)

MyFunc1 – הפונקציה מבצעת....

MyFunc2 – הפונקציה מסננת את...

## ביצועים עיקריים

אחות בכירה רשאית להוסיף את שאר המשתמשים למערכת

אחות בכירה רשאית לההוסיף ניתוחים למערכת

אחות רגילה יכולה לצפות בלו"ז השיבוץ בלבד

המשתמש יכול לקבל רשימת חדרים, ניתוחים, מכשירים עפ"י סינון

למשתמש מותג לוח שיבוץ עפ"י תאריכים מסודר

המנהל יכול להוסיף....

המשתמש מכניס את פרטיו ומקבל רשימת....

למשתמש מוצגת מפה של....

למשתמש נשלחת הודעה עם....

## אילוצים

המערכת פועלת על פי נתונים ואינה יכולה לערוך את השיבוץ בלעדיהם.  
יהיה מספר מוגבל של ניתוחים ביום בחדר לפי יכולת בית החולים

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 הניתוחים מתאריך מסוים

ברצוננו לקבל את כל הניתוחים בתאריךמסוים מ הDB ולכן יתבצעו השלבים הנ"ל:

* המשתמש יחפוץ לראות את כל הניתוחים של תאריךמסוים הוא ילחץ על כפתור מסוים בתצוגה) html( ישלח את התאריך שהוא חפץ ובקשתו תפנה לTypeScript.
* .servicesאשר תפנה ל shoeSurgeryFromSpecifictDate 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 GetSurgeryFromSpecificDate שנמצאת SurgeryManager. 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 הניתוחים לשכבת הBL בה מתבצעת פונקצית הסינון של הבאת ניתוחים מתאריך מסוים.
* .BLמה controllerמחזירה את הנתונים ל GetSurgeryFromSpecificDate 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

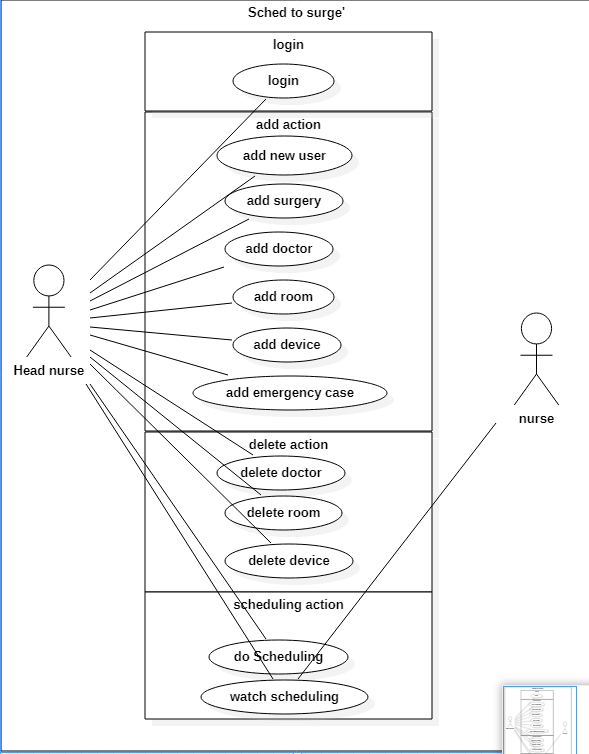
## שרת – לקוח

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



user

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

## רשימת use case

המשתמש מתחבר על ידי הזנת שם וסיסמא

במקרה שזוהי אחות רגילה:  
אחות רגילה רשאית לצפות בשיבוץ

במקרה שזוהי אחות ראשית:

מוסיפה משתמש חדש על ידי הכנסת פרטיו תעודת זהות, שם ,סיסמא, מה התפקיד....  
מוסיפה ניתוח חדש ומכניסה תאריך, מחלקה, מכשירים הנצרכים לניתוח ,רמת הסיכון והעדיפות שנקבעו על ידי הרופא, מי הרופא שקבע.

הוספת חדר איזה מחלקה ומה המכשירים הנמצאים בו

הוספת מכשיר שם, מחלקה, כמות, נייד

הוספת מקרה חירום

מחיקת משתמש על ידי הכנסת שם  
מחיקת חדר הכנסת ID של חדר ומחיקתו

מחיקת מכשיר הכנסת ID של מכשיר ומחיקתו  
הפעלת השיבוץ על ידי שליחת רשימת חדרים , רשימת ניתוחים, רשימת מכשירים, לפעולת השיבוץ

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- היה לי צורך במבנה נתונים שישמור לכל ניתוח את קוד הניתוח ואת רמת העדיפות שחישבו לו בתור הכנה למטריצת השיבוץ וזהו המבנה המתאים ביותר  
רשימה- יה צורך בקבלת מספר אוביקטים ,המרת מספר אוביקטים ושליחתם

(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

API-

שכבה זו מתקשרת עם צד הלקוח, באמצעות קונטרולרים – בקרים, המקבלים פניות מהלקוח ומאחזרים מידע בהתאם.

בשיטה זו קיימת הפרדת ישויות מוחלטת וכל שכבה עומדת באופן עצמאי לחלוטין ומתקשרת רק עם השכבה שמעליה.

בשכבה זו נמצאים הקונטרולים:  
RoomController-קונטרולר שמקבל פניות מהלקוח בנוגע לחדרים כמו קבלת רשימת חדרים פנויים

SurgeryController--קונטרולר שמקבל פניות מהלקוח בנוגע לניתוחים כמו שליפת ניתוחים מתאריך נוכחי,מתאריך ספציפי,שליפת כל הניתוחים והוספת ניתוח.

UserController- קונטרולר שמקבל פניות מהלקוח בנוגע למשתמשים כמו התחברות,הרשמה,עדכון

BL-שכבה זו מכילה את הלוגיקה של הפרויקט. כמו כן, משמשת כמתווכת בין שכבת הDAL לשכבת ה-WebApi, בשליפת ובעדכון הנתונים.

RoomManager-מחלקה שמכילה את כל הפעולות הקשורות לחדרים כמו הוספה,מחיקה,שליפה של רשימת חדרים פנויים,המחלקה מקבלת פניות מהלקוח על ידי ה RoomControllerועושה שינויים בהתאם בדאטה בייס

SurgeryManager- מחלקה שמכילה את כל הפעולות הקשורות לניתוחים כמו הוספה, שליפה של רשימת ניתוחים על פי סינונים,המחלקה מקבלת פניות מהלקוח על ידי ה SurgeryController ועושה שינויים בהתאם בדאטה בייס

UserManager- מחלקה שמכילה את כל הפעולות הקשורות לניתוחים כמו התחברות,הרשמה על ידי ה UserController ועושה שינויים בהתאם בדאטה בייס

PreHungrien-מחלקה המכילה את כל ההכנות המקדימות לאלגוריתם כמו חישוב עדיפות,חדר מתאים,מכשירים מתאימים ובעצם יצירת מטריצת התאמות עם ציונים לכל התאמה

HungrienScudling-מחלקה שבה נמצא האלגוריתם ההונגרי ומתבצע השיבוץ בפועל

Dal-

בשכבה זו קיים מודל שנבנה ע"י טכנולוגיית EntityFrameWork, ובו מחלקה מקבילה לכל טבלה במסד הנתונים

* **DTO-** ספריה זו מכילה את מחלקות מקבילות למבנה הנתונים כדי לקשר בין שכבת הDAL- וה-WebApi.

DeviceForSurgeryDTO-מחלקת מכשירים השמורים לניתוח משמש להמרה לטבלה וממנה

DoctorDTO- מחלקת רופאים משמש להמרה לטבלה וממנה

RoomDTO- מחלקת חדרים משמש להמרה לטבלה וממנה

SpecialDeviceDTO-- מחלקת מכשירים משמש להמרה לטבלה וממנה

SurgeryDTO- מחלקת ניתוחים משמש להמרה לטבלה וממנה  
UserDTO - -מחלקת משתמשים משמש להמרה לטבלה וממנה

ClassDTO- מחלקת סוגי מחלקות משמש להמרה לטבלה וממנה

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

פונקציית האלגוריתם העיקרי מקבלת רשימת חדים, ניתוחים, מכשירים, ומכשירים שצריך כל ניתוח אותם היא שולחת לפונקציה שבונה מטריצה ונותנת ציון לכל התאמה בין חדר לניתוח על פי רמת עדיפות וסיכון שנתן הרופא, מחלקה תואמת, ומכשירים מתאימים לניתוח את הציון הסופי משבצים בריבוע המתאים במטריצה ואת המטריצה שולחים בחזרה למחלקת השיבוץ

על המטריצה המוכנה מפעילים את פעולת השיבוץ שתמצא את ההתאמה הטובה ביותר עבור כל ניתוח היכן יתבצע, האלגוריתם עובר על המטריצה מוצא ציון מקסימאלי בכל שורה ולאחר מכן בכל עמודה ואז בודק אם יש ציון חד חד ערכי במקרה שלא ממשיך בפעולות נוספות כפי שאפשר לראות בקוד עד אשר ימצא הפיתרון הסופי

## חלק מהאלגוריתם... הפיכת התמונה לשחור לבן

## חלק אחר מהאלגוריתם... פירוק התמונה לאותיות

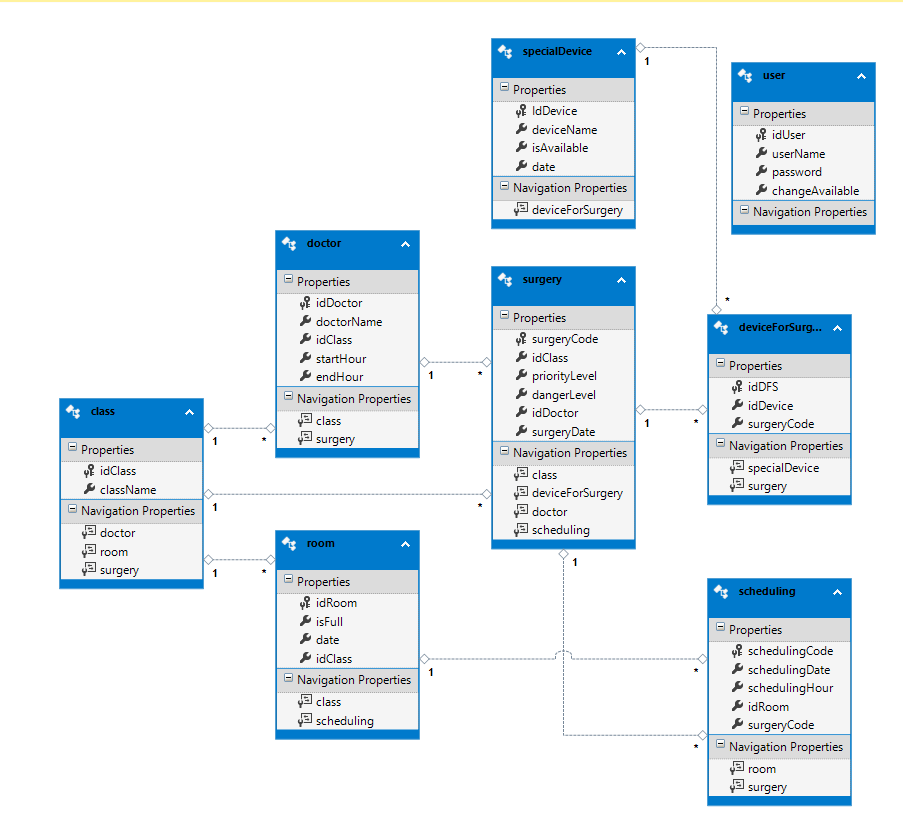
## החלק העיקרי באלגוריתם למידת מכונה – וזיהוי האות

# 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......

User-

|  |  |  |  |
| --- | --- | --- | --- |
| מפתח | שם שדה | תיאו ר | טיפו ס |
| pk | idUser | קוד משתמש | int |
|  | userName | שם משתמש | string |
|  | password | סיסמא | int |
|  | changeAvailable | מורשה/לא | bit |

מכילה את רשימת המשתתמשים הנמצאים בדאטה קוד משתמש

# 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.