# בס"ד

* תכלול את שם המכללה (לוגו), סמל מה"ט, המגמה, המסלול, שם הפרויקט, שם התלמיד/תלמידים, שם המנחה ותאריך ההגשה. (מצ"ב דוגמא לדף הכריכה)



**מותר ורצוי להוסיף לדף אלמנט גרפי או לוגו מתוך הפרויקט.**

# ברוכים הבאים

****

# הצעת פרויקט



תאריך: 3.6.19

לכבוד: יחידת הפרויקטים מה״ט

**הצעה לפרויקט גמר**

**א. פרטי הסטודנטים**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **שנת סיום הלימודים** | **טלפון נייד** | **כתובת** | **ת.ז. 9 ספרות** | **שם הסטודנט** |
| תשע"ט | 0556796880 | הרב דסלר 18 ב"ב | 207087297 | תמר הכהן עדני |
| תשע"ט | 0548524606 | ירושלים 23 ב"ב | 207818709 | אסתר קליין |

שם המכללה: סמינר וולף – שלוחת המכללה למנהל ראשל"צ

סמל המכללה:

מסלול ההכשרה: הנדסאים

מגמת לימוד: תכנות מחשבים

מקום ביצוע הפרויקט: בסמינר

**ב. פרטי המנחה האישי**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מקום עבודה/תפקיד | תואר | טלפון נייד | כתובת | שם המנחה |
| המכללה למנהל | B.ED הנדסאי מחשבים | 0527171295 | חזו"א 11 ב"ב | מ. שימונוביץ |

חתימת הסטודנט חתימת המנחה האישי חתימת הגורם המקצועי מטעם מה״ט

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## שם הפרויקט :

School Schedule

## רקע

### תיאור ורקע כללי

הפרויקט מציג תוכנה לניהול מערכת בית ספר .בתוכנה זו ניתנת אפשרות למשתמש לשנות

לערוך ולהזיז שעורים. כמו כן המערכת תשמור את המערכת עם השינויים האחרונים כל יום בהיסטוריה לפי תאריך.

### מטרות המערכת

#### לאפשר למזכירה לבצע שינויים במערכת בקלות ובנוחות.

#### לאפשר שליפת היסטוריה.

#### לאפשר סקירה של המערכת לפי מורות, כיתות וחדרים.

## סקירת מצב קיים בשוק

בשוק קימות מספר תוכנות המאפשרות בניה וניהול מערכות למוסדות לימודים.

ישנה תוכנה המפרידה בין בניה לניהול, ז"א מחולקת ל 2 תוכנות נפרדות,

בתוכנת הניהול יש כמה באגים ואפשרויות מוגבלות.

## מה הפרויקט אמור לחדש או לשפר

תוכנה המתממשקת לתוכנת הבניה ומנהלת את המערכת היומיומית עם שפורים, תיקונים,

תצוגה נוחה יותר וטכנולוגיה משופרת.

## דרישות מערכת ופונקציונאליות

### דרישות מערכת, סביבת הטמעה ושימוש.

המערכת תעבור קומפילציה והפצה בסביבת visual studio עם התקנת .Net Framework גרסה 4.0 ומעלה

והיא אמורה לרוץ בסביבת שרת אשר מריץ לקבלת בקשות IIS Express לתצוגת דפי אינטרנט.

המשתמש יוכל להריץ את התוכנה בכל מכשיר אשר מותקן עליו דפדפן אינטרנט.

### שרידות, ביצועים, התמודדות עם עומסים

לא צפויים עומסים, צפוי משתמש אחד.

### דרישות פונקציונאליות

#### המזכירה נדרשת להזדהות במערכת.

#### בכל חודש תטען המזכירה למערכת קבצי XML המכילים מערכת עדכנית.

#### המזכירה יכולה לצפות במערכת.

#### המזכירה יכולה להוסיף שעת לימוד למערכת.

#### בעת הוספת שעור המזכירה תבחר בכיתה הרצויה מרשימת הכיתות, ובמורה מרשימת המורות המלמדות בכיתה.

#### המזכירה תצפה ברשימת המקצועות אותם מלמדת המורה הנבחרת, בכיתה הנבחרת ותבחר במקצוע הרצוי.

#### בעת הצורך המזכירה מעדכנת נתוני שעת לימוד.

#### בעת עדכון שעור המזכירה תבחר בכיתה הרצויה מרשימת הכיתות, ובמורה מרשימת המורות המלמדות בכיתה.

#### המזכירה תצפה ברשימת המקצועות אותם מלמדת המורה הנבחרת, בכיתה הנבחרת ותבחר במקצוע הרצוי.

#### בעת ביטול שעת לימוד המזכירה תבחר בכיתה ברשימת הכיתות בתאריך ובשעה הרצויים.

#### בעת היעדרות מורה המזכירה תוסיף היעדרות למורה החסרה מתוך רשימת מורות.

#### המזכירה יכולה לבחור היעדרות לפי שעות.

#### המזכירה יכולה לבחור היעדרות ליום שלם.

#### המזכירה יכולה לעדכן מבחן.

#### המזכירה יכולה לבחור מורה משגיחה מרשימת מורות משוחררות או ממורות פנויות במאגר.

#### המזכירה יכולה לבחור כיתות נבחנות מרשימת הכיתות.

#### המזכירה יכולה לעדכן פעילות.

#### המזכירה תוכל לבחור בחדר לפעילות מרשימת חדרים.

#### המזכירה יכולה לבחור מורה מלווה מרשימת מורות משוחררות או ממורות פנויות במאגר.

#### המזכירה תוכל לבחור מחזור וכיתה.

#### המזכירה תוכל לעדכן פעילות לפי שעות או תאריך.

## בעיות צפויות במהלך הפיתוח

### הבעיות:

#### בעיה 1: שמירת שינויים חדשים לאחר שנעשו כבר שינויים לאותו היום (ללא התייחסות למערכת שעות המקורית)

#### בעיה 2: תקינות קבצי הXML

### פתרונות אפשריים:

#### לבעיה 1: פתרון 1: יצירת טבלה לכל יום שבו ישמרו השינויים לאותו היום. (אחד מהמאפיינים בטבלה הוא האם השיעור הזה התבטל.)

פתרון 2: הוספת שדה בטבלת שיעורים – מורה קודמת.

#### לבעיה 2: פתרון 1: נוודא את תקינות הנתונים לפני ההכנסה לDB

פתרון 2: נדרוש מהמשתמש לספק קבצים מהתוכנה המתממשקת בלבד.

### הפתרון הנבחר עבור כל אחת מהבעיות:

#### לבעיה 1: פתרון 2

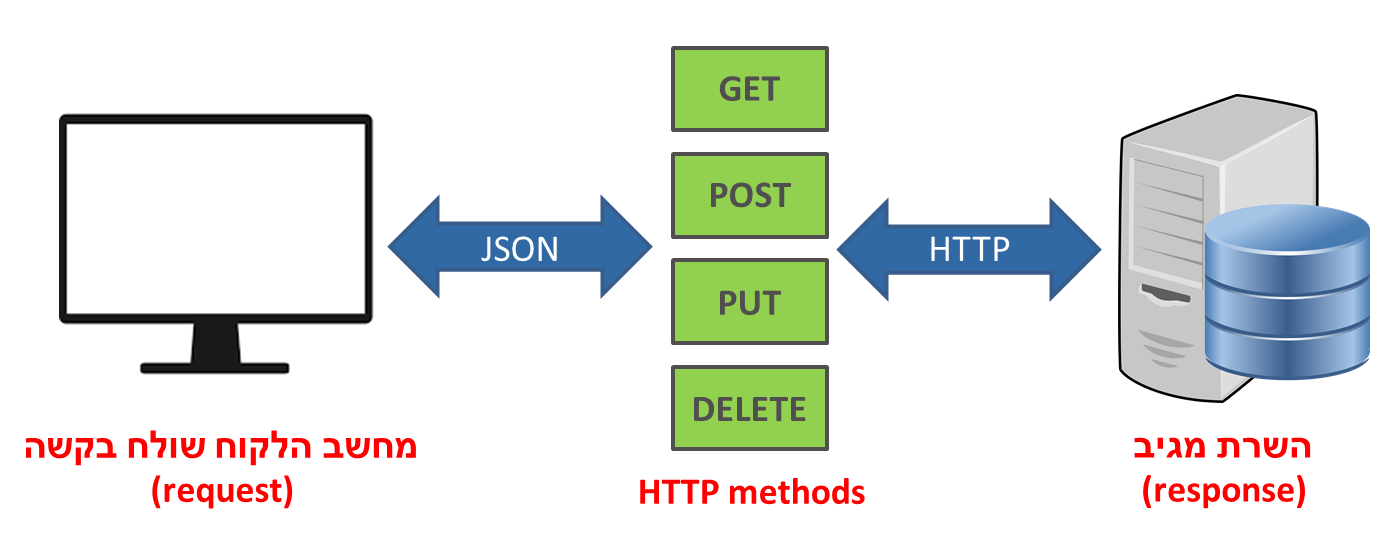
#### לבעיה 2: פתרון 2

## פתרון טכנולוגי נבחר

### טופולוגית הפתרון

ארכיטקטורה אשר מחלקת את היישום לשלש פונקציות עיקריות: ממשק המשתמש, החלק הלוגי/אלגוריתמי וניהול הנתונים. מימוש פונקציות אלה ייתכן במגוון אפשרויות של מחשבים ורשת תקשורת. אחת הצורות הנפוצות למימוש ארכיטקטורת שרת/לקוח היא חלוקת עבודה בין מחשב קצה (לקוח) לבין מחשב מרכזי (שרת).

### דיאגרמה



### טכנולוגיות בשימוש:

צד לקוח: Angular 7.

צד שרת: Web Api.   
מסד נתונים: SQL-SERVER

### שפות הפיתוח:

#### בצד השרת: C#

#### בצד הלקוח: TypeScript HTML5 CSS

### תיאור הארכיטקטורה הנבחרת:

צד לקוח:

אנגולר היא תשתית תוכנה ליישומי רשת בקוד פתוח המתוחזקת ע"י גוגל.  לאנגולר יש קהילה הולכת וגדלה של מפתחים שמפתחים את התשתית באופן שוטף והיא מתבססת על java script ומשמשת בעיקר לפיתוח של אתרים דינמיים. התשתית פותחה ע"י גוגל מתוך מטרה ליעל את האכסון של המידע של אפליקציות בשרתים ובכך לשפר ביצועים ולהוזיל עלויות.

[AngularJs](https://www.mouseux.co.il/ux-blog/angular-dev/) מאפשרת לפתח dynamic Single Page Application ע"י כתיבת קוד נקי וקריא. הרבה קוד נחסך ע"י מאפיינים טכניים שכבר מובנים באנגולר

* קל לבצע מניפולציות על הDOM
* ניתן לעשות שימוש חוזר ברכיבים בקלטת
* ניתן להריץ אנגולר על כל היישומים.

צד שרת:  
Web Api-שירות אינטרנט באמצעות C# שפה נוחה לשימוש.

### חלוקה לתכניות ומודולים:

#### בצד השרת: BL-Business Logic Layer- שכבה שתפקידה לעבד ולהגיב לאירועים שמתרחשים. DAL- - Data Access Layerשכבה שמטפלת בגישה לנתונים. DTO-Data Transfer Object- שכבה המכילה ישויות המקבילות לישויות בשכבת ה DAL על מנת לשנע נתונים לצד הלקוח בעזרת ה DTO עצמם.  ניתן לשמור על המידע שעובר בין ה- שכבות בפורמט יציב בלי להתייחס לשינויים שמתרחשים בכל אחד מהשכבות.

#### בצד הלקוח: Component Service Model Module

## סביבת השרת

### ממשק המשתמש/לקוח – GUI

ממשק הלקוח יתקבל כקובץ HTML ויהיה מוצג באמצעות דפדפן אינטרנט.

### ממשקים למערכות אחרות / API רכיב תוכנה קיים המתממשק למערכת ומזין עבורה נתונים.

### שימוש בחבילות תוכנה: Entity Framework Angular Core Bootstrap

## מבני נתונים וארגון קבצים

### שיטת האחסון : הנתונים ישמרו במסד נתונים של SQL-SERVER הנתונים המיובאים בקבצי XML הנתונים מיוצאים בקבצי Excel

### מבני הנתונים:

#### כיתות: קוד, מספר מזהה, שם, מספר כיתה במחזור, מספר מחזור, קוד סוג ביה"ס (חטיבת ביניים, תיכון וכדו').

#### כיתות לאירועים: קוד כיתה, קוד אירוע.

#### יומן: קוד, מספר מזהה, קוד מורה, נושא, סוג, קוד כיתה, קוד חדר, האם מקובץ, קוד אירוע, רפורמה, מתאריך, עד תאריך, שעה ביום, סכום שעות מחושב, סיבה.

#### יומן לכיתות: קוד כיתה, קוד יומן.

#### אירועים: קוד, מתאריך, עד תאריך, קוד סוג אירוע.

#### סוגי אירוע: קוד, שם.

#### תשלומים עבור חיסורים: קוד, שם, האם בתקן(מאושר).

#### תשלומי חיסורים למורות: קוד מורה, קוד תשלום.

#### חדרים: קוד, קומה, מספר חדר, קוד כיתה, תיאור השימוש.

#### כיתות לאירועים: קוד כיתה, קוד אירוע.

#### תצורת המערכת: קוד, יום תחילת שבוע, יום, שעה, קוד חדר.

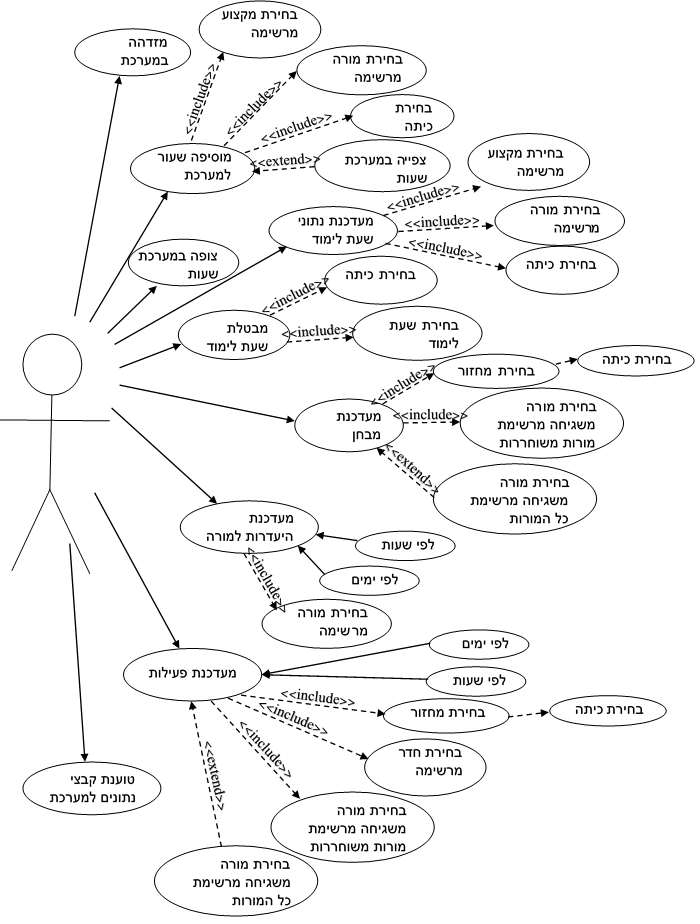
#### מורות: קוד, מספר מזהה, שם.

#### חיסורים למורה: קוד, קוד מורה, מתאריך, עד תאריך, קוד מורה ממלאת מקום.

#### מקצועות: קוד, שם.

#### מקצועות לכיתות: קוד מקצוע, קוד כיתה.

#### מורות לאירועים: קוד מורה, קוד אירוע.



## תרשימי מערכת מרכזיים

### Use Case

## תיאור המרכיב האלגוריתמי – חישובי

### איזה בעיה בא לפתור, איך יפתור?

הבעיה : הנתונים בתוכנה מגיעים ממקור חיצוני, בעת שינוי הנתונים יש צורך לטעון את הנתונים המעודכנים מהמקור החיצוני.   
הפתרון: הנתונים מגיעים בקבצי XML , התוכנה תשלוף את המידע ותעביר אותו למסד הנתונים.

### איסוף מידע וניתוחים סטטיסטיים (אנליטיקות)

איסוף מידע : בעת הטעינה יש לבדוק מה השתנה ולעדכנו, לבדוק עם נוספו נתונים ולהוסיפם.

## תיאור/התייחסות לנושאי אבטחת מידע

לא נדרשת אבטחת מידע מיוחדת, למעט כניסה למערכת.

## משאבים הנדרשים לפרויקט:

### מספר שעות המוקדש לפרויקט: 720

### חלוקת עבודה בין חברי הצוות: עבודה משותפת וחלוקה למשימות בהתאם לצורך.

### ציוד נדרש: מחשב הכולל חיבור לאינטרנט.

### תוכנות נדרשות: VISAL STUDIO 2017, VISUAL CODE, SSMS.

### ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט

### ספרות ומקורות מידע <https://github.com/topics/timetable> <http://www.yedatech.co.il> <https://reshetech.co.il> <http://webmaster.org.il> <https://dotnet.microsoft.com>

## תכנית עבודה ושלבים למימוש הפרויקט:

|  |  |  |
| --- | --- | --- |
| **שלב** | **משך זמן** | **תאריכים משוערים** |
| * + ייזום הרעיון | שבועיים | 16.4.19-3.5.19 |
| * + ניתוח מערכת | 3 שבועות | 2.5.19-20.5.19 |
| * + ניתוח מבנה נתונים | שבועיים | 10.6.19-24.6.19 |
| * + אפיון UX – UI | חודש | 25.6.19-25.7.19 |
| * + כתיבת הלוגיקה העסקית | חודשיים | 26.7.19-26.9.19 |
| * + כתיבת ממשק המשתמש | חודש | 27.9.19-27.10.19 |
| * + עיצוב | 2.5 שבועות | 27.10.19-15.11.19 |

## תכנון הבדיקות שיבוצעו

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **מספר בדיקה** | **מס` דרישה במסמך אפיון** | **מקרי הבדיקה** | **ידנית/ אוטומטית** | **חשיבות** | **הערות- תוצאת הבדיקה הרצויה** |
| 1. | 5.3.1 | כניסה למערכת תתאפשר ע"י הזדהות בלבד | ידנית | גבוהה | לא תתאפשר כניסה ללא הזדהות במערכת |
| 2. | 5.3.2 | בעת טעינת קבצי הXML יכנסו הנתונים לDB | ידנית | גבוהה | לאחר הטעינה יופיעו הנתונים מהקבצים בDB |
| 3. | 5.3.3 | תתאפשר צפייה במערכת | ידנית | בינונית | ניתן לצפות במערכת |
| 4. | 5.3.4 | תתאפשר הוספת שעת לימוד למערכת | ידנית | גבוהה | שעת הלימוד אכן התווספה למערכת |
| 5. | 5.3.4.1  5.3.5.1  5.3.8.1  5.3.9.2 | המזכירה תוכל לבחור מורה מרשימת המורות | ידנית | גבוהה | נתוני המורה הנבחרת ישמרו במערכת |
| 6. | 5.3.5 | המזכירה תוכל לעדכן נתוני שעת לימוד | ידנית | גבוהה | נתוני שעת הלימוד אכן יתעדכנו |
| 7. | 5.3.6 | המערכת תאפשר ביטול שעת לימוד | ידנית | גבוהה | שעת הלימוד אכן תרד מהמערכת |
| 8. | 5.3.7 | תתאפשר הוספת ההיעדרות למורה חסרה | ידנית | גבוהה | למורה הנבחרת תתווסף היעדרות |
| 9. | 5.3.7.1 | תתאפשר בחירת היעדרות מורה לפי שעות | ידנית | גבוהה | היעדרות למורה הנבחרת תוגדר בטווח השעות שהוזן למערכת |
| 10. | 5.3.7.2 | תתאפשר בחירת היעדרות מורה ליום שלם | ידנית | בינונית | היעדרות למורה הנבחרת תוגדר בטווח השעות של כל אותו היום |
| 11. | 5.3.8 | המערכת תאפשר עדכון מבחן | ידנית | גבוהה | המבחן יוגדר במערכת |
| 12. | 5.3.8.2 | תתאפשר בחירת כיתות מרשימת הכיתות | ידנית | גבוהה | פרטי הכיתה הנבחרת ישמרו במערכת |
| 13. | 5.3.9 | המערכת תאפשר עדכון פעילות | ידנית | גבוהה | הפעילות תוגדר במערכת |
| 14. | 5.3.9.1 | המערכת תאפשר לבחור חדר לפעילות מתוך רשימת חדרים. | ידנית | נמוכה | מקום הפעילות יוגדר בחדר הנבחר |
| 15. | 5.3.9.3 | המערכת תאפשר לבחור מחזור וכיתה. | ידנית | גבוהה | תתעדכן פעילות למחזור/הכיתה הנבחר/ת |
| 16. | 5.3.9.4 | יתאפשר עדכון פעילות לפי שעות או תאריך | ידנית | בינונית | הפעילות תתעדכן לפרק הזמן הנבחר. |

## בדיקות יחידה (unit test)

לא רלוונטי.

## בקרת גרסאות (version control)

לא רלוונטי

**חתימת הסטודנט חתימת המנחה האישי**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

הערות ראש המגמה במכללה:

אישור ראש המגמה במכללה:

שם:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

הערות הגורם המקצועי מטעם מה"ט:

אישור הגורם המקצועי מטעם מה"ט:

שם:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# הבעת תודה

ראשית תודה לבורא העולם.

תודה לרכזת המגמה בסמינר הרב וולף – המו' חנה ברגמן, שפועלת רבות כדי לאפשר לנו תנאים מיטביים ומתאימים.

כמו כן, תודה למנחה על הציוד בעקרונות ובכלים הטובים ביותר לפיתוח תוכנה שלוונו במהלך כתיבת הפרויקט, ועל הסיוע בתכנון הפרויקט בצורה המוצלחת ביותר.

תודה למזכירת המוסד ג. פישר על שיתוף הפעולה, ההכרות עם פעילות המוסד וצרכיו לפרטי פרטים.

תודה לאחראיות חדרי המחשבים על סיוע טכני בחפץ לב, ושהיו בעת הצורך.

תודה לחברות לספסל הלימודים על העצות הקטנות והרעיונות היעילים שנתנו בין לבין, שתמכו ושתפו.

ולכל מי שליווה תמך ועזר מסביב, אם ישירות ואם בעקיפין.

ולסיום גם אם נמלא כאן דפים מלאי תודות למי שהכל שלו והכל ממנו, אף פעם לא נגיע לקצה גודל התודה האמיתית.

# הצהרה

# תוכן עניינים

[בס"ד I](#_Toc18515017)

[ברוכים הבאים II](#_Toc18515018)

[הצעת פרויקט III](#_Toc18515019)

[הבעת תודה 2](#_Toc18515065)

[הצהרה 3](#_Toc18515066)

[תוכן עניינים 4](#_Toc18515067)

[תקציר 7](#_Toc18515068)

[מבוא 9](#_Toc18515069)

[תיאור הפרויקט 10](#_Toc18515070)

[אסטרטגיות וטכנולוגיות 10](#_Toc18515071)

[שפות 10](#_Toc18515072)

[טכנולוגיות 11](#_Toc18515073)

[סביבות עבודה: 12](#_Toc18515074)

[**תאור מבנה הפרויקט** 13](#_Toc18515075)

[**עקרונות התכנון הבניה והניתוח** 18](#_Toc18515076)

[הפרדת  שכבות 18](#_Toc18515077)

[OOP 18](#_Toc18515078)

[Binding 18](#_Toc18515079)

[xml DOM 18](#_Toc18515080)

[LINQ to SQL 18](#_Toc18515081)

[**תרשימים** 19](#_Toc18515082)

[תרשים זרימה המתאר את מסכי המערכת או לחילופין עץ תהליכים. 19](#_Toc18515083)

[תרשים מראה המחלקות 19](#_Toc18515084)

[מבנה  נתונים 22](#_Toc18515085)

[מאוכסנים SQL - כולל תיאור טבלאות, קשרי גומלין, שיקולים שנבעו תוך כדי בנית מסד הנתונים. 22](#_Toc18515086)

[תיאור מבנה קבצי XML, 22](#_Toc18515087)

[תוכן הפרויקט 23](#_Toc18515088)

[תיאור המחלקות 23](#_Toc18515089)

[פונקציות מעניינות 24](#_Toc18515090)

[מדריך למשתמש 24](#_Toc18515091)

[סיכום ומסקנות 25](#_Toc18515092)

[נספחים 26](#_Toc18515093)

[XML Classes 26](#_Toc18515094)

[שונות 31](#_Toc18515095)

[ביבליוגרפיה 32](#_Toc18515096)

**עד כאן מספור העמודים יופיע באותיות רומיות.**

**מכאן והלאה – גוף העבודה – יופיע מספור עמודים בספרות רגילות – 1,2,3.**

# תקציר

המערכת שלנו פותחה ותוכננה עבור מוסד לימודים גדול. היא מאפשרת לנהל היעדרויות של מורות, לשבץ מורות ממלאות מקום. לנהל שעורים- להוסיף שעור במקרה הצורך, להוריד שעור או לעדכן שעור, כלומר לשנות את מקצוע השיעור או את המורה המלמדת. לנהל אירועים-לשבץ אירועים במערכת, כמו מבחן, טיול, הרצאה, מחנה וכדו'.

יש בה 2 הרשאות :  
הרשאת מזכירה- בהכנסת סיסמה בפתיחת המערכת, יכולה לבצע שינויים במערכת לעיל, לבצע חיסורי מורות ולצפות בהם, לערוך אירועים.    
הרשאת אורח- כל משתמש אחר יכול לצפות במערכת ובאירועים השונים.

# מבוא

בניהול מוסד לימודים ישנם תחומים רבים, התחום העיקרי הוא ניהול מערכת השעות.  
כאשר מורה נעדרת יש צורך לטפל בשעורים שבהן היא מלמדת, לקבוע ממלאת/ות מקום או לבצע הזזת שעורים במקרה הצורך, כמו כן להוסיף לה היעדרות.  
כאשר יש צורך לאחד כיתות, לשחרר כיתות וכדו' יש לבצע את הפעולות הנדרשות.  
במקרה של אירוע, ז"א אין שעור/ים סדיר/ים (כמו הרצאה, טיול או מבחן) יש לדאוג לפרטים המתאימים.  
כאשר רוצים לצפות במערכת השעות, או לאפשר למורות/תלמידות צפייה במערכת המתאימה להן, יש לאפשר להן.

בבנייתנו את המערכת חקרנו את המערכות הנמצאות בשימוש כיום, למדנו את הצרכים והדרישות. מערכת זו פותחה ועוצבה כדי לעמוד בדרישות, להיות נוחה, יעילה וזמינה.

כמו שנכתב לעיל המערכת מאפשרת:   
- להוסיף חיסור למורה בתאריך המתאים, וכן לשבץ מורה ממלאת מקום מתוך רשימת המורות.  
- להוסיף שיעור, להוריד שיעור או לעדכן שיעור בפרטים הרלוונטיים.  
- להוסיף אירוע, להוריד אירוע או לעדכן אירוע בפרטים הרלוונטיים.  
- לצפות במערכת המתאימה, לפי כיתות, מורות או אירועים.

ההרשאות הקימות במערכת:  
הרשאת מנהל - המזכירה האחראית יכולה לנהל את השינויים והפרטים שקורים כאשר מורה חסרה. לשנות את המערכת לפי שינויים זמניים, לנהל את השיעורים, לנהל את האירועים.     
הרשאת אורח - כל אחד אחר יכול לצפות במערכת השעות העדכנית, לצפות באירועים שנקבעו וליצור קשר עם הצוות הטכני - צוות התמיכה.

המערכת חיבת כקלט קבצי xml תקניים שנבנו במערכת התואמת -תלמ"ש, המכילים נתונים בסיסיים למערכת השעות. בנתונים אלו כלולים מבנה מוסד הלימודים, הכיתות הקימות בו, המקצועות הנלמדים בו והמורות המלמדות בו.

בתקווה שהבנתם ונהניתם

צוות   School Schedule

# תיאור הפרויקט

## אסטרטגיות וטכנולוגיות

### שפות

בכתיבת הפרויקט השתמשנו בשפות הבאות:

• #C :

שפת #C היא שפת תכנות שפותחה ע"י מיקרוסופט ונחשבת לאחת

משפות התכנות הפופולריות בעולם. התחביר והעקרונות שלה הם עשירים ביכולות ועם זאת פשוטים ונוחים לעבודה.

היא מיועדת לפיתוח כללי של מגוון אפליקציות בכל התחומים כמו משחקים, אתרי Web, אפליקציות למכשירי מובייל וטאבלטים ועוד.

• HTML:

שפת HTML היא כלי פיתוח בסיסי להצגת מידע סטאטי בדפדפני האינטרנט הנפוצים. השיטה בה עובדת שפה זו היא הורדת המידע מהשרת בו מתארח האתר אל מחשב המשתמש.

שפת HTML היא השפה הבסיסית המרכיבה את מבנה ותוכן האתר. כאשר הדפדפן שלנו דוגמת chrome או explorer מעוניין להעלות את האתר הוא למעשה קורא את שפת ה-HTML ועל פיה יודע מה להציג לנו.

שפת ה-HTML היא שפת תגיות, מתקשרת עם שפת ה-CSS שהיא שפת העיצוב.

• : TypeScript

Typescript הינה שפת תכנות חינמית ובקוד פתוח, שפותחה במשך כשנתיים ע"י Microsoft וראתה אור לראשונה בשנת 2012.

היא מאוחסנת ב – GitHub ומטרתה להתמודד עם חסרונות קיימים ב – JavaScript ע"י הוספת סוגים, מחלקות, מודולים ועוד.

TypeScript הינה הרחבה ל – JavaScript והיא מכילה את כל הפקודות והתחביר של שפת המקור.

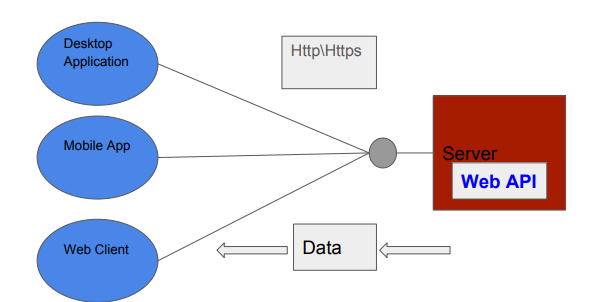
• : CSS

גיליונות סגנון מדורגים הם פורמט לעיצוב דפי אינטרנט. הגיליונות קובעים את עיצובם של תגים ב-HTML, ‏ XHTML וכל שפה דומה ל-XML לבניית אתרי אינטרנט. CSS נוצר במטרה להפריד בין תוכן ומבנה דפי האינטרנט לבין עיצובם.

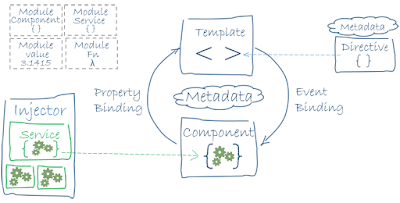
### טכנולוגיות

השתמשנו בטכנולוגיות הבאות לכתיבת הפרויקט:

* Entity Framework-צד שרת:  
   הספרייה Entity Framework דואגת למיפוי של אובייקטי המודל אל תוך מסד הנתונים באופן כמעט אוטומטי שאינו דורש כתיבת קוד. מיקרוסופט פיתחה ORM (Object Relational Mapping) אשר חוסך את כל העבודה המונוטונית הנדרשת לפיתוח שכבת מידע.  
   Entity Framework הוא כלי אשר, אוטומטית, ממפה טבלאות של מסדי נתונים שטוחים למחלקות, חוסך את הצורך בכתיבת קוד רב ומסורבל, ומאפשר למפתחים להתמקד ב-Business Logic של המערכת. Entity Frameworkהיא מערכת טכנולוגיות ב- ADO.NET התומכות בפיתוח יישומי תוכנה מוכווני נתונים. Entity Framework מאפשרת למפתחים לעבוד עם נתונים בצורה של אובייקטים ותכונות ספציפיים לתחום, כגון לקוחות וכתובות לקוחות, מבלי שהם צריכים לדאוג לעצמם לטבלאות ועמודות בסיס הנתונים בהן נשמרים נתונים. באמצעות Entity Framework, מפתחים יכולים לעבוד ברמה גבוהה יותר של הפשטה כאשר הם מתמודדים עם נתונים, ויכולים ליצור ולתחזק יישומים מונחי נתונים עם פחות קוד מאשר ביישומים בעלי טכנולוגיה ישנה יותר.
* Web Api-צד שרת:  
   Web Api הוא API אותו מממש השרת והוא מחזיר לקליינט Data בלבד ולא .View הבקשה והתשובה הן בפרוטוקול Https\Http הקליינטים של Web Api יכולים להיות רבים.



* Angular -צד לקוח:  
   Angular זה Framework לבניית אפליקציות תוך שימוש ב-HTML ו-JavaScript או TypeScript. Angular זה פרויקט קוד פתוח שמפותח ומתוחזק ע"י גוגל וקהילה רחבה של מפתחים. כתיבת אפליקציית Angular מתבצעת ע"י כתיבת HTML שבתוכו משולבים תוספות של Angular, כתיבת components כדי לשלוט על ה-templates, כתיבה של services כדי לשלוט על הלוגיקה, ושילוב של components ו- services ליצירת modules. הארכיטקטורה של Angular:



### סביבות עבודה

* Visual Studio 2017
* Visual Code
* Microsoft SQL Server Management Studio 17

## **תיאור מבנה הפרויקט**

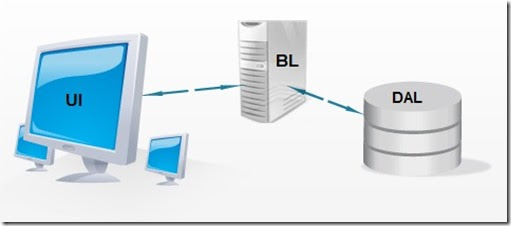
**חלוקה לשכבות** היא תבנית עיצוב בסיסית שמגדירה הפרדת האפליקציה ל שכבת נתונים, שכבת לוגיקה, ושכבת ממשק משתמש.

(כאמור בהגדרה הבסיסית, ככל שהתכנה מורכבת יותר ייתכנו שכבות נוספות)

למה זה טוב?

* תחזוקה: ניתן להחליף או לתקן מימוש פנימי של שכבה אחת בארכיטקטורה בלי לשנות שכבה אחרת.
* נוחות פיתוח: אדם אחד עובד על רכיב בתכנה, אדם אחר עובד על רכיב אחר, כל עוד שהחתימות זהות ניתן לשלב כוחות ולייעל זמני פיתוח.
* בדיקות: תקלה כלשהיא מבודדת בכל שכבה בנפרד, לדוגמא אם לא קיבלנו רשימת נתונים לתצוגה נבדוק קודם את שלב הביניים (שכבת הBL) אם הנתונים שם תקינים נדע בוודאות שגם ברמת שכבת הDAL הנתונים תקינים וכל שנותר הוא לפתור את התקלה ברמת הUI
* שימוש חוזר: נניח שהרכיבים BL + DAL אהובים עלינו ועובדים היטב, ניתן להחליף את פלטפורמת הUI לטכנולוגיה אחרת במינימום מאמץ.
* אבטחה: נוח יותר להגדיר Interface-API לשכבה מסוימת בלי לחשוף מבני נתונים, לוגיקה או נתונים ומידע שלא רלוונטיים למשתמש, כמו גם חסימה בפני האקרים (רלוונטי יותר בטכנולוגיות WEB) ברמות שונות.

ניתן להגדיר רמות שונות של חלוקה לשכבות ולקבל מבנה יעיל יותר; נקבל מבנה שנראה בערך כך:

[](http://blogs.microsoft.co.il/blogs/uriel99/3T_5B0030EA.jpg)

* API  
  עם שכבה זו מתקשר ה Client, כאן קימות מתודות הדואגות לקבל את הבקשות, לתקשר עם השכבות הנוספות ולהשיב תגובות.
* BL  
  השכבה שאמונה על הלוגיקה של המערכת, עוסקת בעיבוד המידע, בחישובים שונים ושליחתו לשכבת הAPI-.  
  בשכבה זו מימשנו את הפונקציונאליות של המערכת.
* DAL  
  שכבה זו מורכבת מקישור למסד הנתונים בטכנולוגית Entity Framework, וממערכת תוכנה אשר תפקידה לקרוא את המידע הנדרש למערכת, לשמור את העדכונים, ולהוסיף מידע חדש או למחוק פרטי מידע קיימים (נקרא גם CRUD- Create, Read, Update, Delete).
* DTO  
  שכבה זו מכילה טיפוסי נתונים (מחלקות) המקבילים לטיפוסים בשכבת ה DAL, ללא שדות הניווט וכו', ובנוסף מתודות הממירות את טיפוסי ה DAL לטיפוסי ה DTO ולהיפך.
* XmlFilesLoadingImplemention  
  שכבה זו קוראת את קבצי הנתונים לפני טעינתם למסד הנתונים, מבצעת את הפעולות המתאימות עד להכנסתם.

## **עקרונות התכנון הבניה והניתוח**

### הפרדת  שכבות

כפי שהרחבנו בתיאור הפרויקט הפרדנו את הפרויקט לשכבות, ובכל שכבה טפלנו בחלקי הקוד המתאימים.

### OOP

היא פיתוח מתודה בעלת שם מתודה ורמת גישה זהה למתודה קיימת אך עם פרמטרים שונים. כך ניתן לכתוב למשל 10 מתודות עם Overloading אותו שם ורמת גישה, אך שמקבלות פרמטרים שונים ומבצעות דברים שונים (ולרוב גם קיימים קשרים בין המתודות השונות).  
השתמשנו בפונקציות מועמסות, לדוגמה Get(int id) המחזירה אובייקט לפי ID מתאים, וGet() המחזירה את אוסף האובייקטים הקיים.

### Binding

בצד לקוח השתמשנו בAngular , רכיב חשוב בטכנולוגיה זו הואngModel אשר מאפשר Two Way Binding -קישור דו כיווני. קשירה דו-כיוונית נותנת לנו לעדכן את המידע המוצג למשתמש בקובץ התבנית וגם את ערכה של התכונה בקומפוננטה בו-זמנית

### xml DOM

DOM (Document Object Module) מרכיב את קובץ ה-XML בזיכרון המחשב מאובייקטים. בצורה כזו המידע הרבה יותר זמין, דרך הפעולה הזו יותר קלה ואינטואיטיבית והקוד נהיה אף מובן יותר. הנתונים למערכת נטענים באמצעות קבצי xml, בתבנית מסוימת, בטעינת הקבצים אנו אוספים את הנתונים ושופכים לתוך מסד הנתונים.

### LINQ to SQL

**Linq** - ראשי תיבות של Language Integrated Query, מספקת לנו מודל ותחביר אחידים לטיפול במידע ללא קשר למקור הנתונים ממנו הוא מגיע, היא מאפשרת להגדיר שאילתות כחלק אינטגראלי משפת התכנות בה אנו משתמשים. Linq זה כינוי לשפת שאילתות המובנית כחלק אינטגרלי משפת #C. Linq היא שפה שלא רק שהתחביר שלה מאוד דומה לשפת SQL אלא בעיקר המשמעות שלה, היא שפת שאילתות המאפשרת לבצע שאילתות על אוספים. ב-Linq ניתן להשתמש בשתי דרכים: הראשונה בתחביר ייעודי שהפך להיות חלק מתחביר השפה והשניה היא אוסף מתודות ייעודיות העושות שימוש בביטויי למבדה תוך שימוש ב- Func<T> delegate ומסתירות את התחביר הבסיסי ובכך להפחית את כמות הקוד הנדרש למספר שורות בודדות.

Linq מאפשרת ביצוע שאילתות חיפוש, סינון, סידור, צירוף, איחוד לקבוצות של נתונים וכו'.

השתמשנו רבות בסיוע לשליפה מהמסד נתונים, לסנן וכדו'.

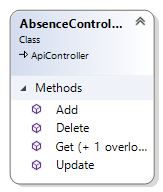
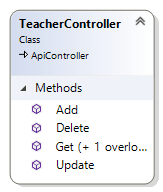
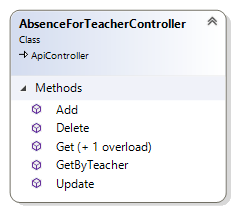
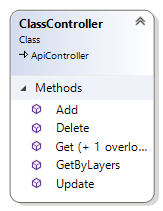
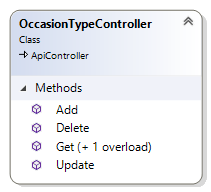
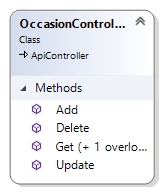
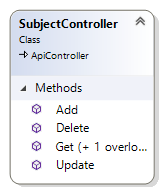
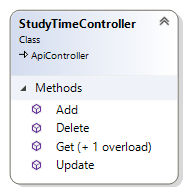
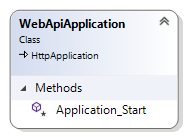
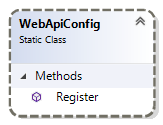
### Singleton

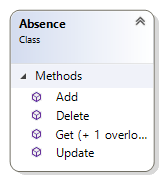
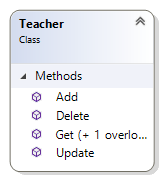
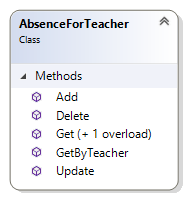
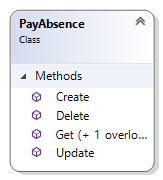
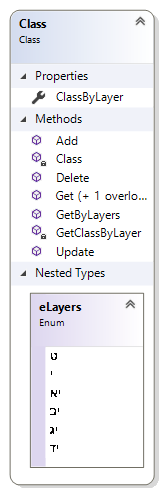
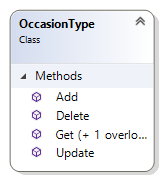
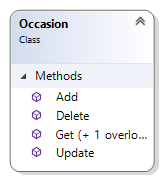
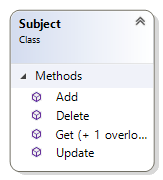
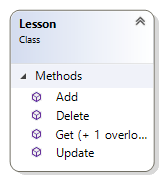
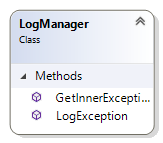
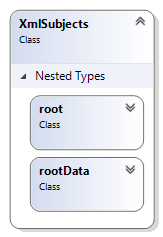
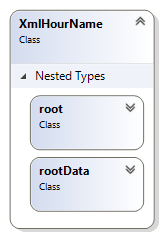
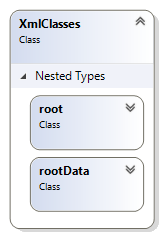
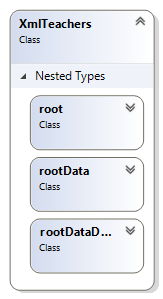
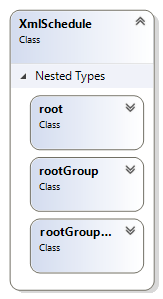
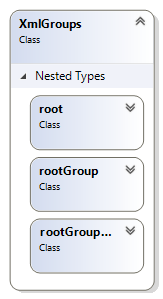
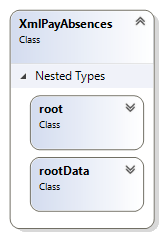
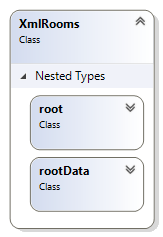
Singleton זו מחלקה שייתכן רק מופע אחד שלה. בצד לקוח השתמשנו בAngular ,השימוש בSingleton נפוץ בAngular לדוגמה בService וכן ב Route.

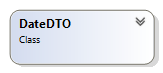
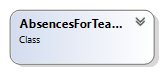
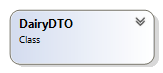
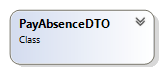
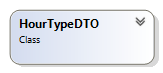
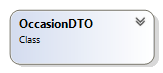
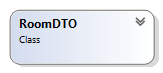
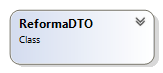
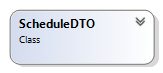
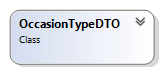
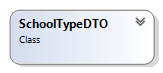
## **תרשימים**

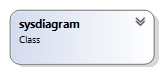
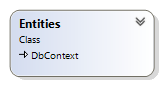
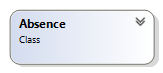
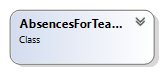
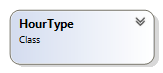
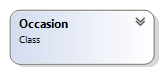
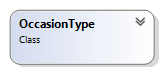
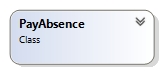
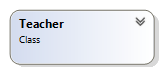
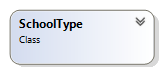
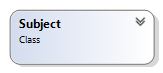
### תרשים זרימה המתאר את מסכי המערכת או לחילופין עץ תהליכים.

### תרשים מראה המחלקות

API  
  
מחלקה המטפלת בסוגי חיסורים (של מורות)   
מחלקה המטפלת במורות  
מחלקה המטפלת בחיסורים למורות  
מחלקה המטפלת בכיתות  
מחלקה המטפלת בסוגי אירועים (של כיתות)   
מחלקה המטפלת באירועים לכיתות  
מחלקה המטפלת במקצועות  
מחלקה המטפלת בשעת לימוד  
 מחלקה המטפלת בבקשות ה Webבכלליות  
מחלקה המטפלת במאפיינים והגדרות של הServer

BL  
מחלקה המנהלת את סוגי חיסורים  
מחלקה המנהלת את מורות  
מחלקה המנהלת את חיסורים למורות  
מחלקה המנהלת את גורמים משלמים לחיסורי מורות  
מחלקה המנהלת את כיתות  
מחלקה המנהלת את סוגי אירועים  
מחלקה המנהלת את אירועים לכיתות  
מחלקה המנהלת את מקצועות  
מחלקה המנהלת את שעות לימוד (משבצת במערכת)   
מחלקה המנהלת את הLog –ים   
  
המחלקות הבאות מקבילות לקובץ הXml המתאים  
  


DTO  
מחלקה המכילה מתודות המרה משכבת הDAL לשכבת הDTO ולהיפך  
   
מחלקות המכילות טיפוסי נתונים מקבילים לשכבת הDTO  
               

DAL  
מחלקה *המכילה את נתוני הדיאגרמות במסד הנתונים*  
  
מ*חלקה המכילה אוספים המאפשרים לקבל נתונים מאובייקטים עיקריים במסד הנתונים*  
*  
המחלקות הבאות מכילות את צורת האובייקטים במסד הנתונים  
*             

#### XmlFilesLoadingImplemention

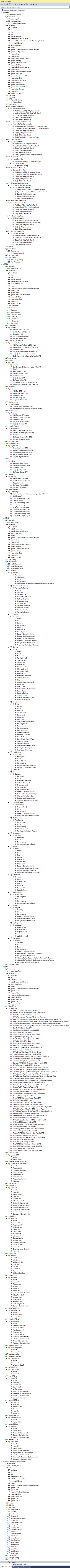
## מבנה  נתונים



### תיאור מבנה קבצי XML

## תוכן הפרויקט

### תיאור המחלקות



מייני את המחלקות שאת מביאה לפי פרויקטים, ולפי מרחבי שמות. בתתי סעיפים.

- לכל מחלקה הציגי ככותרת את שם המחלקה,

לאחר מכן הביאי תאור קצר על המחלקה, מה תפקידה ואלו חברי מחלקה עיקריים היא מכילה, (עם הסבר מה תפקיד כל חבר)

פונקציות מעניינות  
כאן את יכולה להציג בפרוט- פונקציות מעניינות ויפות שכתבת במחלקה, להביא את שמן ככותרת משנה, ולאחר מכן לתאר מה הן עושות, ולהביא את הקוד או קטעים יפים ממנו שאת בוחרת להציג.

במידה וזו פונקציה מורכבת ניתן להסביר בקצרה את האלגוריתם במילים.

אין צורך להדפיס כאן את כל הקוד ממש, אלא רק את כותרות הפונקציות, הקוד, מיקומו בנספחים. והוא **חייב** להיות מלווה בהערות.

=========

### פונקציות מעניינות

# מדריך למשתמש

הוראות כלליות לשימוש במערכת:

מסכים- כאן המקום להציג את צילומי המסכים של המערכת. והוראות והסברים איך משתמשים בהם.

# סיכום ומסקנות

לאחר תקופה שאנו עובדות ובונות נרצה להצביע על מספר הישגים ותוצאות וגם לחלוק  
במסקנות:  
נרצה לציין שפרויקט בסדר גודל כזה, נתן לנו המון ידע וניסיון בפיתוח הן בצד ה client והן  
בצד ה server. הפרויקט איחד את כל המושגים שלמדנו. וגרם לנו להבין את המושגים  
המופשטים יותר.  
הקמת פרויקט מרעיון כללי ועד לסיומו דורש המון זמן ידע והשקעה. בהחלט השקענו המון,  
ובזמן שניתן לנו הצלחנו לבנות ממשק נוח וברור המקיים את הפונקציונליות המורכבת של  
טיפול יעיל בפניות.  
כפי שניתן לראות, האתר נמצא בשלבי פיתוח מתקדמים, עדיין לא כל הקוד מופיע בספר, ולא  
כל הדפים מוכנים. סביר להניח שעם ההתקדמות והסיום, יעלו עוד שינויים, שיפורים ומסקנות  
חדשות.

# נספחים

## XML Classes

public class XmlClasses

{

// NOTE: Generated code may require at least .NET Framework 4.5 or .NET Core/Standard 2.0.

/// <remarks/>

[System.SerializableAttribute()]

[System.ComponentModel.DesignerCategoryAttribute("code")]

[System.Xml.Serialization.XmlTypeAttribute(AnonymousType = true)]

[System.Xml.Serialization.XmlRootAttribute(Namespace = "", IsNullable = false)]

public partial class root

{

public rootData[] dataField;

public uint school\_idField;

public string data\_typeField;

public string eG\_time2Field;

/// <remarks/>

[System.Xml.Serialization.XmlElementAttribute("data")]

public rootData[] data

{

get

{

return this.dataField;

}

set

{

this.dataField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public uint school\_id

{

get

{

return this.school\_idField;

}

set

{

this.school\_idField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public string data\_type

{

get

{

return this.data\_typeField;

}

set

{

this.data\_typeField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public string EG\_time2

{

get

{

return this.eG\_time2Field;

}

set

{

this.eG\_time2Field = value;

}

}

}

/// <remarks/>

[System.SerializableAttribute()]

[System.ComponentModel.DesignerCategoryAttribute("code")]

[System.Xml.Serialization.XmlTypeAttribute(AnonymousType = true)]

public partial class rootData

{

public string nameField;

public byte numField;

public byte mapField;

public ushort idField;

public byte class\_numberField;

public byte layerField;

public byte reformaField;

public byte schooltypeField;

public ushort backColorField;

public uint foreColorField;

/// <remarks/>

public string name

{

get

{

return this.nameField;

}

set

{

this.nameField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public byte num

{

get

{

return this.numField;

}

set

{

this.numField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public byte map

{

get

{

return this.mapField;

}

set

{

this.mapField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public ushort id

{

get

{

return this.idField;

}

set

{

this.idField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public byte class\_number

{

get

{

return this.class\_numberField;

}

set

{

this.class\_numberField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public byte layer

{

get

{

return this.layerField;

}

set

{

this.layerField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public byte reforma

{

get

{

return this.reformaField;

}

set

{

this.reformaField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public byte schooltype

{

get

{

return this.schooltypeField;

}

set

{

this.schooltypeField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public ushort BackColor

{

get

{

return this.backColorField;

}

set

{

this.backColorField = value;

}

}

/// <remarks/>

[System.Xml.Serialization.XmlAttributeAttribute()]

public uint ForeColor

{

get

{

return this.foreColorField;

}

set

{

this.foreColorField = value;

}

}

}

}

# ביבליוגרפיה

[https://docs.microsoft.com](https://docs.microsoft.com/)

[https://stackoverflow.com](https://stackoverflow.com/)

[https://angular.io](https://angular.io/)

[https://github.com](https://github.com/)

[https://www.w3.org](https://www.w3.org/)

[https://scotch.io](https://scotch.io/)

[https://codepen.io](https://codepen.io/)

[https://en.wikipedia.org](https://en.wikipedia.org/)

[https://confluence.atlassian.com](https://confluence.atlassian.com/)

[https://nodejs.org](https://nodejs.org/)

[https://update.angular.io](https://update.angular.io/)

[https://getbootstrap.com](https://getbootstrap.com/)

[https://ng-bootstrap.github.io](https://ng-bootstrap.github.io/)

[https://www.freecodecamp.org](https://www.freecodecamp.org/)

[https://love2dev.com](https://love2dev.com/)

[https://www.w3schools.com](https://www.w3schools.com/)

[https://www.regextester.com](https://www.regextester.com/)

[https://regex101.com](https://regex101.com/)

[https://www.techiediaries.com](https://www.techiediaries.com/)

[https://www.flaticon.com](https://www.flaticon.com/)

<https://www.hamichlol.org.il>

http://www.tzurel.co.il/tag

[https://www.geektime.co.il](https://www.geektime.co.il/microsoft-snip-will-upgrade-your-screenshots/)

<https://support.office.com>

[https://www.maketecheasier.com](https://www.maketecheasier.com/capture-scrolling-screenshot-windows/)

<https://blogs.msdn.microsoft.com>

<https://picpick.app/en/>