



המכון הממשלתי להכשרה
בטכנולוגיה ומדע



SCH School Schedule

מוגש ע"י: אסתר קליין

תמר הבהן עדני

מגמת הנדסת תוכנה

מנחה: מ.שמונוביץ



המכון הממשלתי להכשרה
בטכנולוגיה ומדע



School Schedule

שמות המגישות: אסתר קליין

תמר הכהן עדני

חתימה: _____

חתימה: _____

מגמת הנדסת תוכנה

מנחה: מ.שמונוביץ

חתימה: _____

מרכזת המגמה: ח.ברגמן

חתימה: _____

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.

הצהרת סטודנט

שם הסטודנט: ראר הכנן-רז ת.ז. 207087297

שם הסטודנט: סלמא קל ת.ז. 207818709

שם המכללה בה לומד הסטודנט: סמינר הוג וול

אני הח"מ, מצהיר בזאת כי פרויקט הגמר וספר הפרויקט המצ"ב נעשו על ידי בלבד.
פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.
פרויקט הגמר וספר הפרויקט נעשו על בסיס הנחייתו של המנחה האישי.
מקורות המידע בהם השתמשתי לביצוע פרויקט הגמר מצוינים ברשימת המקורות המצוינים בספר הפרויקט.
אני מודע לאחריות שהנני מקבל על עצמי על ידי חתימתי על הצהרה זו שכל הנאמר בה אמת ורק אמת.

תאריך: 04/09/19

תאריך: 04/09/19

חתימת הסטודנט: ראר הכנן-רז

חתימת הסטודנט: סלמא קל

אישור המנחה האישי

הריני מאשר שהפרויקט בוצע בהנחייתי, קראתי את ספר הפרויקט ומצאתי כי הוא מוכן לצורך הגשת הסטודנט להגנה על פרויקט גמר.

שם המנחה: ד"ר ע'מאן חתימה: _____ תאריך: _____

אישור ראש המגמה

הריני מאשר שספר הפרויקט מוכן לצורך הגשת הסטודנט להגנה על פרויקט הגמר.

שם ראש המגמה: _____ חתימה: _____ תאריך: _____

תוכן עניינים

3.....	הבעת תודה
4.....	הצהרה
5.....	תוכן עניינים
6.....	תקציר
7.....	מבוא
8.....	תיאור הפרויקט
8.....	אסטרטגיות וטכנולוגיות
8.....	שפות
9.....	טכנולוגיות
10.....	סביבות עבודה
12.....	תיאור מבנה הפרויקט
14.....	עקרונות התכנון הבניה והניתוח
14.....	הפרדת שכבות
14.....	OOP
14.....	Binding
14.....	XML DOM
14.....	LINQ to SQL
15.....	Singleton
16.....	תרשימים
16.....	תרשים מראה המחלקות
36.....	מבנה נתונים
39.....	תיאור מבנה קבצי XML
42.....	תוכן הפרויקט
42.....	תיאור המחלקות
65.....	פונקציות מעניינות
71.....	מדריך למשתמש
82.....	סיכום ומסקנות
83.....	נספחים
83.....	מנגנון אתחול מערכת
88.....	מנגנון טיפול בשגיאות
88.....	XML Classes
93.....	ביבליוגרפיה

תקציר

המערכת שלנו פותחה ותוכננה עבור מוסד לימודים גדול. היא מאפשרת לנהל היעדרויות של מורות, לשבץ מורות ממלאות מקום. לנהל שעורים- להוסיף שיעור במקרה הצורך, להוריד שיעור או לעדכן שיעור, 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 חייבת בקלט קבצי xml תקינים 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

בבתיבת 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 חינומית 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

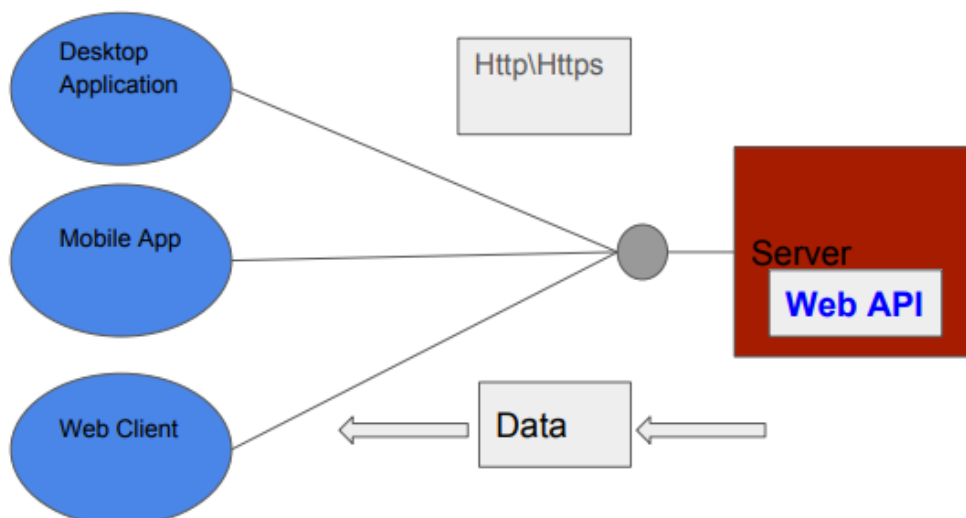
השתמשנו בטכנולוגיות הבאות לכתיבת הפרויקט:

- 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 (Object Relational Mapping) ORM 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 Framework היא 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:

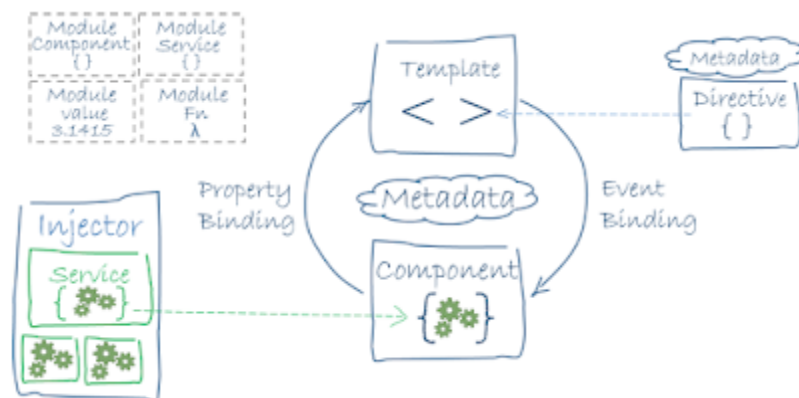
Web Api הוא API אותו מממש השרת והוא מחזיר לקליינט Data בלבד ולא View. הבקשה והתשובה הן בפרוטוקול Http\Https הקליינטים של Web Api יכולים להיות רבים.



- 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 Studio היא סביבת פיתוח מובילה מבית חברת מיקרוסופט, המאפשרת למתכנתים לפתח תוכניות מחשב ואתרי אינטרנט אינטראקטיביים.
Visual Studio .NET מאפשרת פיתוח, מתוך תוכנה אחת וסביבה אחת, אפליקציות מסוגים שונים (חלונאיות, WEB, פרויקטי התקנה ועוד) ובשפות שונות הנתמכות ב-.NET. ולמעבדים שונים (32 ו-64 סיביות) ואף להתקנים ניידים כמו מחשבי כף יד וטלפונים סלולריים. הרכיב העיקרי של הסביבה הוא עורך טקסט שצובע את המילים לפי ההקשר התכנותי שלהם. לעורך תכונת "השלמה אוטומטית" המציגה חלונית עם שמות העצמים הקשורים לכתיבה ומאפשרת כתיבה מהירה ומניעת טעויות ושגיאות.
בין היתר מוטמעים כלי פיתוח חזותיים שונים שמחוללים קוד בצורה אוטומטית, כמו מעצבים חזותיים לממשקי משתמש Winform ו-WPF, וכלים שמאפשרים נגישות והתמצאות במרחבי הפרויקט כמו סייר הפרויקט לעיון בקובצי הפרויקט, סייר בסיסי הנתונים לעבודה עם בסיסי נתונים. כלי ניפוי שגיאות שונים שמאפשרים עצירה בנקודות מסוימות ועיון בערכי המשתנים.

- **Visual Code**
Visual Studio Code הוא שמה של פלטפורמה מבוססת קוד פתוח

לעריכת קוד ופיתוח תוכנה הפועלת על מערכות ההפעלה Windows, לינוקס ו-OS X. העורך תומך בניפוי שגיאות, בקרת תצורה של גיט, המחשה סינטקטית של קטעי קוד, השלמת קוד חכמה (intelligent code completion), קטעי קוד אוטומטיים (snippets), ושינויי קוד רוחביים (code refactoring). ויז'ואל סטודיו קוד היא עורך קוד. ויז'ואל סטודיו קוד תומכת במספר שפות תכנות.

- Microsoft SQL Server Management Studio 17
Microsoft SQL Server היא מערכת לניהול בסיס נתונים במודל היחסי (RDBMS) של חברת מיקרוסופט. שפת הפיתוח שבאמצעותה מועברות הפקודות למערכת (על ידי כתיבת פקודות או שימוש בממשק גרפי ששולח פקודות באופן סמוי) היא Transact-SQL (TSQL) שהיא מימוש תקן ANSI של שפת SQL משמשת לתשאול וטיפול בנתונים (DML), יצירת טבלאות והיחסים ביניהן (DDL) ותחזוקת המערכת תוך שימוש בתוכניות שירות שונות.

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 ה-BL) אם הנתונים שם תקינים נדע בוודאות שגם ברמת שכבת ה-DAL הנתונים תקינים וכל שנותר הוא לפתור את התקלה ברמת ה-UI
- שימוש חוזר: נניח שהרכיבים BL + DAL אהובים עלינו ועובדים היטב, ניתן להחליף את פלטפורמת ה-UI 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:



- 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 ה-API.
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 טיפוסים נתונים (מחלקות) המקבילים לטיפוסים בשכבת ה DAL, ללא שדות הניווט וכו', ובנוסף מתודות הממירות את טיפוס ה DAL לטיפוס ה DTO ולהיפך.
- XmlFilesLoadingImplementation
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 כמה מתודות עם Overloading אותו שם ורמת גישה, אך שמקבלות פרמטרים שונים ומבצעות דברים שונים (ולרוב גם קיימים קשרים בין המתודות השונות).

ניתן לראות בקלות כי השתמשנו בפונקציות מועמסות בפרויקט, לדוגמה `Get(int id)` המחזירה אובייקט לפי ID מתאים, ו `Get()` המחזירה את אוסף האובייקטים הקיים.

Binding

בצד לקוח השתמשנו ב Angular, רכיב חשוב בטכנולוגיה זו הוא ngModel 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 בזמן 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 אינטגרלי 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 בביטוי למבדה תוך שימוש ב- `delegate Func<T>` 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'.

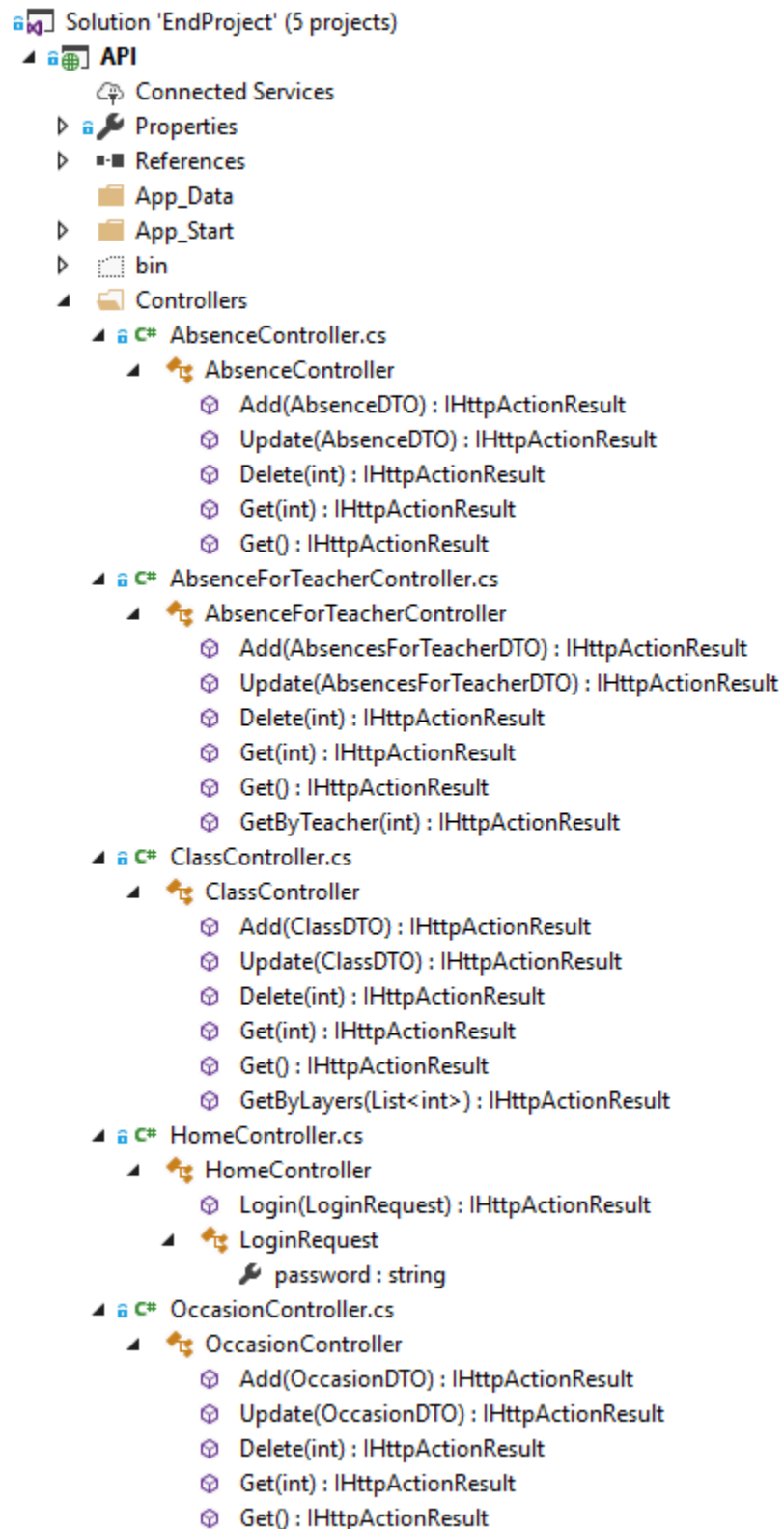
השתמשנו רבות בסיוע לשליפה מהמסד נתונים, לסנן וכדו'.

Singleton

Singleton זו מחלקה שיייתכן רק מופע אחד שלה. בצד לקוח השתמשנו ב Angular, השימוש ב Singleton נפוץ ב Angular לדוגמה ב Service וכן ב Route.













תרשימים

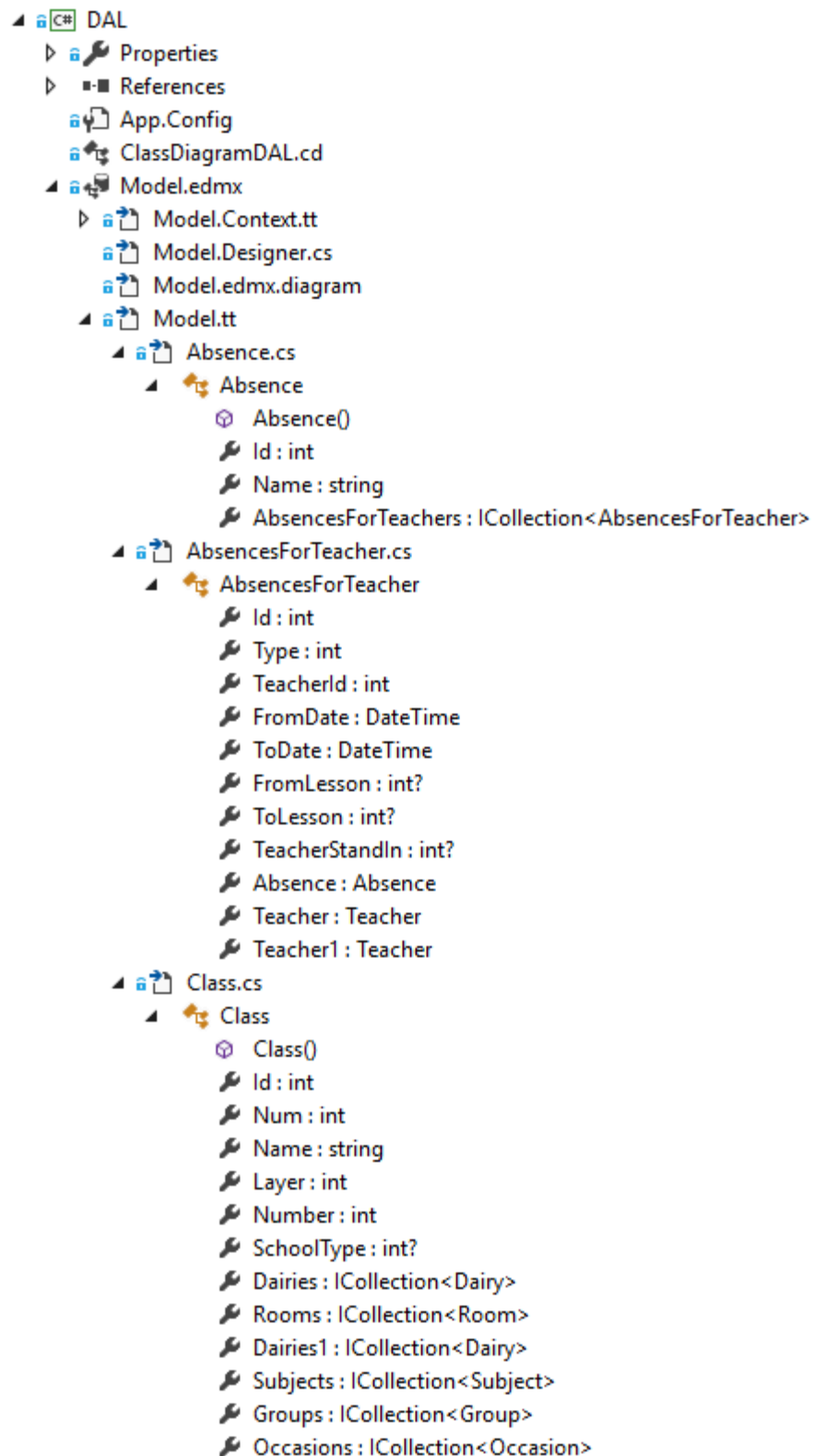
תרשים מראה המחלקות

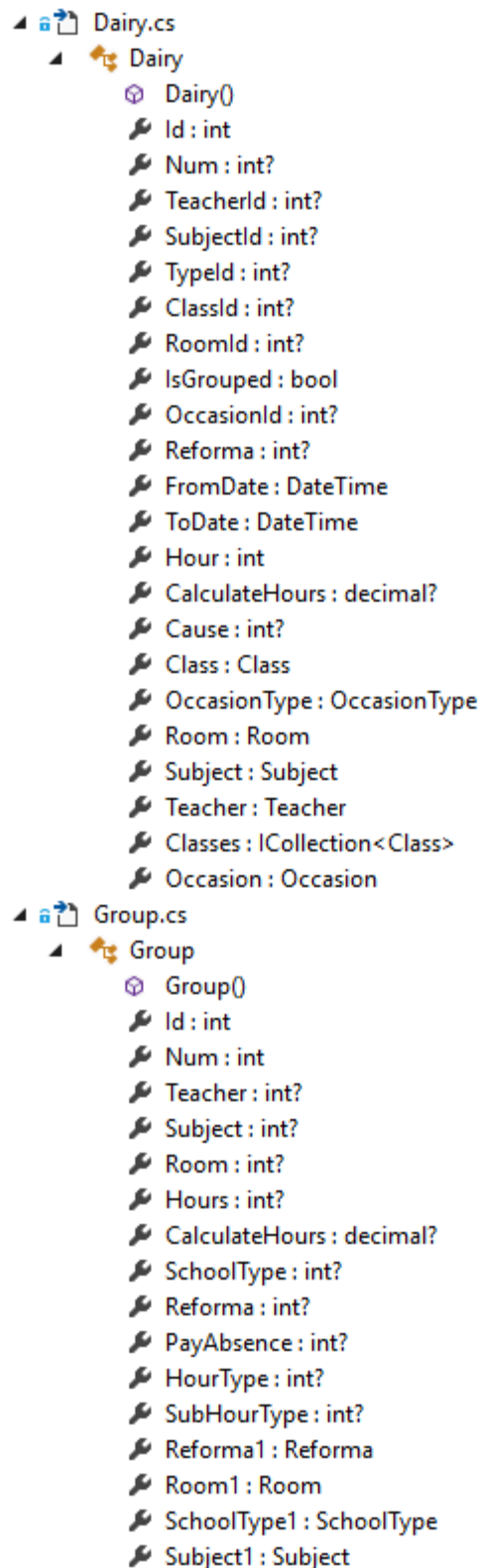


- OccasionTypeController
 - Add(OccasionTypeDTO) : IHttpActionResult
 - Update(OccasionTypeDTO) : IHttpActionResult
 - Delete(int) : IHttpActionResult
 - Get(int) : IHttpActionResult
 - Get() : IHttpActionResult
 - StudyTimeController.cs
 - StudyTimeController
 - Add(DairyDTO) : IHttpActionResult
 - Update(DairyDTO) : IHttpActionResult
 - Delete(int) : IHttpActionResult
 - Get(int) : IHttpActionResult
 - Get() : IHttpActionResult
 - SubjectController.cs
 - SubjectController
 - Add(SubjectDTO) : IHttpActionResult
 - Update(SubjectDTO) : IHttpActionResult
 - Delete(int) : IHttpActionResult
 - Get(int) : IHttpActionResult
 - Get() : IHttpActionResult
 - TeacherController.cs
 - TeacherController
 - Add(TeacherDTO) : IHttpActionResult
 - Update(TeacherDTO) : IHttpActionResult
 - Delete(int) : IHttpActionResult
 - Get(int) : IHttpActionResult
 - Get() : IHttpActionResult
 - Models
 - obj
 - ClassDiagramAPI.cd
 - Global.asax
 - packages.config
 - Web.config
 - BL
 - Properties
 - References
 - XmlClassesFiles
 - Absence.cs
 - Absence
 - Add(AbsenceDTO) : void
 - Update(AbsenceDTO) : void
 - Delete(int) : void
 - Get() : List<AbsenceDTO>
 - Get(int) : AbsenceDTO

- └─ AbsencesForTeacher
 - ⊞ Add(AbsencesForTeacherDTO) : void
 - ⊞ Update(AbsencesForTeacherDTO) : void
 - ⊞ Delete(int) : void
 - ⊞ Get() : List<AbsencesForTeacherDTO>
 - ⊞ Get(int) : AbsencesForTeacherDTO
 - ⊞ GetByTeacher(int) : AbsencesForTeacherDTO
- App.config
- Class.cs
 - Class
 - eLayers
 - ClassByLayer : Dictionary<int, List<ClassDTO>>
 - Class()
 - ⊞ Add(ClassDTO) : void
 - ⊞ Update(ClassDTO) : void
 - ⊞ Delete(int) : void
 - ⊞ Get(int) : ClassDTO
 - ⊞ Get() : List<ClassDTO>
 - ⊞ GetClassByLayer(int) : List<ClassDTO>
 - ⊞ GetByLayers(List<int>) : List<ClassDTO>
- ClassDiagramBL.cd
- Lesson.cs
 - Lesson
 - ⊞ Add(DairyDTO) : void
 - ⊞ Update(DairyDTO) : void
 - ⊞ Delete(int) : void
 - ⊞ Get(int) : DairyDTO
 - ⊞ Get() : List<DairyDTO>
- LogManager.cs
 - LogManager
 - ⊞ LogException(Exception) : void
 - ⊞ GetInnerExceptionMessage(Exception) : string
- Occasion.cs
 - Occasion
 - ⊞ Add(OccasionDTO) : void
 - ⊞ Update(OccasionDTO) : void
 - ⊞ Delete(int) : void
 - ⊞ Get() : List<OccasionDTO>
 - ⊞ Get(int) : OccasionDTO

- ▲  OccasionType.cs
 - ▲  OccasionType
 - ⊗ Add(OccasionTypeDTO) : void
 - ⊗ Update(OccasionTypeDTO) : void
 - ⊗ Delete(int) : void
 - ⊗ Get() : List<OccasionTypeDTO>
 - ⊗ Get(int) : OccasionTypeDTO
 -  packages.config
- ▲  PayAbsence.cs
 - ▲  PayAbsence
 - ⊗ Create(PayAbsenceDTO) : void
 - ⊗ Update(PayAbsenceDTO) : void
 - ⊗ Delete(PayAbsenceDTO) : void
 - ⊗ Get() : List<PayAbsenceDTO>
 - ⊗ Get(int) : PayAbsenceDTO
- ▲  Subject.cs
 - ▲  Subject
 - ⊗ Add(SubjectDTO) : void
 - ⊗ Update(SubjectDTO) : void
 - ⊗ Delete(int) : void
 - ⊗ Get(int) : SubjectDTO
 - ⊗ Get() : List<SubjectDTO>
- ▲  Teacher.cs
 - ▲  Teacher
 - ⊗ Add(TeacherDTO) : void
 - ⊗ Update(TeacherDTO) : void
 - ⊗ Delete(int) : void
 - ⊗ Get() : List<TeacherDTO>
 - ⊗ Get(int) : TeacherDTO
- ▲  XmlManager.cs
 - ▲  XmlManager
 -  fileNameToAction : Dictionary<string, Action<string>>
 - ⊗ XmlManager()
 - ⊗ LoadDirectory(string) : void
 - ⊗ ClearDB() : void
 - ⊗ LoadClasses(string) : void
 - ⊗ LoadGroups(string) : void
 - ⊗ LoadRooms(string) : void
 - ⊗ LoadSchedule(string) : void
 - ⊗ LoadSubjects(string) : void
 - ⊗ LoadTeachers(string) : void
 - ⊗ GetXmlData<T>(string) : T

















- Teacher1 : Teacher
- Classes : ICollection<Class>
- HourType1 : HourType
- HourType2 : HourType
- HourType.cs
 - HourType
 - HourType()
 - Id : int
 - ParentId : int?
 - Name : string
 - Groups : ICollection<Group>
 - Groups1 : ICollection<Group>
- Model.cs
- Occasion.cs
 - Occasion
 - Occasion()
 - Id : int
 - FromDate : DateTime?
 - ToDate : DateTime?
 - FromLesson : int?
 - ToLesson : int?
 - OccasionType : int?
 - Dairies : ICollection<Dairy>
 - OccasionType1 : OccasionType
 - Classes : ICollection<Class>
 - Rooms : ICollection<Room>
 - Teachers : ICollection<Teacher>
- OccasionType.cs
 - OccasionType
 - OccasionType()
 - Id : int
 - Name : string
 - Dairies : ICollection<Dairy>
 - Occasions : ICollection<Occasion>
- PayAbsence.cs
 - PayAbsence
 - PayAbsence()
 - Id : int
 - Name : string
 - IsStandart : bool?
 - Teachers : ICollection<Teacher>

- Reforma.cs
 - Reforma
 - Reforma()
 - Id : int
 - Name : string
 - Groups : ICollection<Group>
 - Room.cs
 - Room
 - Room()
 - Id : int
 - Floor : int
 - Number : int
 - ClassId : int?
 - UseFor : string
 - Class : Class
 - Dairies : ICollection<Dairy>
 - Schedules : ICollection<Schedule>
 - Groups : ICollection<Group>
 - Occasions : ICollection<Occasion>
 - Schedule.cs
 - Schedule
 - Num : int
 - WeekDay : int
 - Day : int
 - Hour : int
 - Room : int?
 - Room1 : Room
 - SchoolType.cs
 - SchoolType
 - SchoolType()
 - Id : int
 - Name : string
 - Groups : ICollection<Group>
 - Subject.cs
 - Subject
 - Subject()
 - Id : int
 - Name : string
 - Dairies : ICollection<Dairy>
 - Classes : ICollection<Class>
 - Groups : ICollection<Group>
- sysdiagram.cs

- Teacher.cs
 - Teacher
 - Teacher()
 - Id : int
 - Num : int
 - Name : string
 - Dairies : ICollection<Dairy>
 - PayAbsences : ICollection<PayAbsence>
 - Groups : ICollection<Group>
 - AbsencesForTeachers : ICollection<AbsencesForTeacher>
 - AbsencesForTeachers1 : ICollection<AbsencesForTeacher>
 - Occasions : ICollection<Occasion>
 - packages.config
- DTO
 - Properties
 - References
 - _CastDTO.cs
 - _CastDTO
 - AbsenceToDTO(Absence) : AbsenceDTO
 - AbsenceToDTO(List<Absence>) : List<AbsenceDTO>
 - DTOToAbsence(AbsenceDTO) : Absence
 - DTOToAbsence(List<AbsenceDTO>) : List<Absence>
 - AbsencesForTeacherToDTO(AbsencesForTeacher) : AbsencesForTeacherDTO
 - AbsencesForTeacherToDTO(List<AbsencesForTeacher>) : List<AbsencesForTeacherDTO>
 - DTOToAbsencesForTeacher(AbsencesForTeacherDTO) : AbsencesForTeacher
 - DTOToAbsencesForTeacher(List<AbsencesForTeacherDTO>) : List<AbsencesForTeacher>
 - ClassToDTO(Class) : ClassDTO
 - ClassToDTO(List<Class>) : List<ClassDTO>
 - DTOToClass(ClassDTO) : Class
 - DTOToClass(List<ClassDTO>) : List<Class>
 - DairyToDTO(Dairy) : DairyDTO
 - DairyToDTO(List<Dairy>) : List<DairyDTO>
 - DTOToDairy(DairyDTO) : Dairy
 - DTOToDairy(List<DairyDTO>) : List<Dairy>
 - GroupToDTO(Group) : GroupDTO
 - GroupToDTO(List<Group>) : List<GroupDTO>
 - DTOToGroup(GroupDTO) : Group
 - DTOToGroup(List<GroupDTO>) : List<Group>
 - HourTypeToDTO(HourType) : HourTypeDTO
 - HourTypeToDTO(List<HourType>) : List<HourTypeDTO>
 - DTOToHourType(HourTypeDTO) : HourType
 - OccasionToDTO(Occasion) : OccasionDTO
 - OccasionToDTO(List<Occasion>) : List<OccasionDTO>
 - DTOToOccasion(OccasionDTO) : Occasion
 - DTOToOccasion(List<OccasionDTO>) : List<Occasion>
 - OccasionTypeToDTO(OccasionType) : OccasionTypeDTO

- ⊗ OccasionTypeToDTO(List<OccasionType>) : List<OccasionTypeDTO>
- ⊗ DTOTToOccasionType(OccasionTypeDTO) : OccasionType
- ⊗ DTOTToOccasionType(List<OccasionTypeDTO>) : List<OccasionType>
- ⊗ PayAbsenceToDTO(PayAbsence) : PayAbsenceDTO
- ⊗ PayAbsenceToDTO(List<PayAbsence>) : List<PayAbsenceDTO>
- ⊗ DTOTToPayAbsence(PayAbsenceDTO) : PayAbsence
- ⊗ DTOTToPayAbsence(List<PayAbsenceDTO>) : List<PayAbsence>
- ⊗ ReformaToDTO(Reforma) : ReformaDTO
- ⊗ ReformaToDTO(List<Reforma>) : List<ReformaDTO>
- ⊗ DTOTToReforma(ReformaDTO) : Reforma
- ⊗ DTOTToReforma(List<ReformaDTO>) : List<Reforma>
- ⊗ RoomToDTO(Room) : RoomDTO
- ⊗ RoomToDTO(List<Room>) : List<RoomDTO>
- ⊗ DTOTToRoom(RoomDTO) : Room
- ⊗ DTOTToRoom(List<RoomDTO>) : List<Room>
- ⊗ SchoolTypeToDTO(SchoolType) : SchoolTypeDTO
- ⊗ SchoolTypeToDTO(List<SchoolType>) : List<SchoolTypeDTO>
- ⊗ DTOTToSchoolType(SchoolTypeDTO) : SchoolType
- ⊗ DTOTToSchoolType(List<SchoolTypeDTO>) : List<SchoolType>
- ⊗ ScheduleToDTO(Schedule) : ScheduleDTO
- ⊗ ScheduleToDTO(List<Schedule>) : List<ScheduleDTO>
- ⊗ DTOTToSchedule(ScheduleDTO) : Schedule
- ⊗ DTOTToSchedule(List<ScheduleDTO>) : List<Schedule>
- ⊗ SubjectToDTO(Subject) : SubjectDTO
- ⊗ SubjectToDTO(List<Subject>) : List<SubjectDTO>
- ⊗ DTOTToSubject(SubjectDTO) : Subject
- ⊗ DTOTToSubject(List<SubjectDTO>) : List<Subject>
- ⊗ TeacherToDTO(Teacher) : TeacherDTO
- ⊗ TeacherToDTO(List<Teacher>) : List<TeacherDTO>
- ⊗ DTOTToTeacher(TeacherDTO) : Teacher
- ⊗ DTOTToTeacher(List<TeacherDTO>) : List<Teacher>
- ⊗ DateToDTO(DateTime) : DateDTO
- ⊗ DateToDTO(DateTime?) : DateDTO
- ⊗ DateToDTO(List<DateTime>) : List<DateDTO>
- ⊗ DateToDTO(List<DateTime?>) : List<DateDTO>
- ⊗ DTOTToDate(DateDTO) : DateTime
- ⊗ DTOTToDate(List<DateDTO>) : List<DateTime>

- ▲  AbsenceDTO.cs
 - ▲  AbsenceDTO
 -  Id : int
 -  Name : string

- ▲  AbsencesForTeacherDTO.cs
 - ▲  AbsencesForTeacherDTO
 - 🔗 Id : int
 - 🔗 TeacherId : int
 - 🔗 FromDate : DateDTO
 - 🔗 ToDate : DateDTO
 - 🔗 FromLesson : int?
 - 🔗 ToLesson : int?
 - 🔗 TeacherStandIn : int?
 - 🔗 Type : int
 -  App.config
 -  ClassDiagramDTO.cd
- ▲  ClassDTO.cs
 - ▲  ClassDTO
 - 🔗 Id : int
 - 🔗 Num : int
 - 🔗 Name : string
 - 🔗 Layer : int
 - 🔗 Number : int
 - 🔗 SchoolType : int?
 - 🔗 Dairies : ICollection<int>
 - 🔗 Rooms : ICollection<int>
 - 🔗 Dairies1 : ICollection<int>
 - 🔗 Subjects : ICollection<int>
 - 🔗 Groups : ICollection<int>
 - 🔗 Occasions : ICollection<int>
- ▲  DairyDTO.cs
 - ▲  DairyDTO
 - 🔗 Id : int
 - 🔗 Num : int?
 - 🔗 TeacherId : int?
 - 🔗 SubjectId : int?
 - 🔗 Typeld : int?
 - 🔗 ClassId : int?
 - 🔗 RoomId : int?
 - 🔗 IsGrouped : bool
 - 🔗 OccasionId : int?
 - 🔗 Reforma : int?
 - 🔗 FromDate : DateDTO
 - 🔗 ToDate : DateDTO
 - 🔗 Hour : int
 - 🔗 CalculateHours : decimal?
 - 🔗 Cause : int?

```

└─ C# DTO
    └─ DateDTO.cs
        └─ DateDTO
            ├── Month : int
            ├── Minute : int
            ├── Millisecond : int
            ├── Hour : int
            ├── Day : int
            ├── Second : int
            ├── Year : int
            └── thisDate : DateTime

    └─ GroupDTO.cs
        └─ GroupDTO
            ├── Id : int
            ├── Num : int
            ├── Teacher : int?
            ├── Subject : int?
            ├── Room : int?
            ├── Hours : int?
            ├── CalculateHours : decimal?
            ├── SchoolType : int?
            ├── Reforma : int?
            ├── PayAbsence : int?
            ├── HourType : int?
            ├── SubHourType : int?
            └── Classes : ICollection<int>

    └─ HourTypeDTO.cs
        └─ HourTypeDTO
            ├── Id : int
            ├── ParentId : int?
            └── Name : string

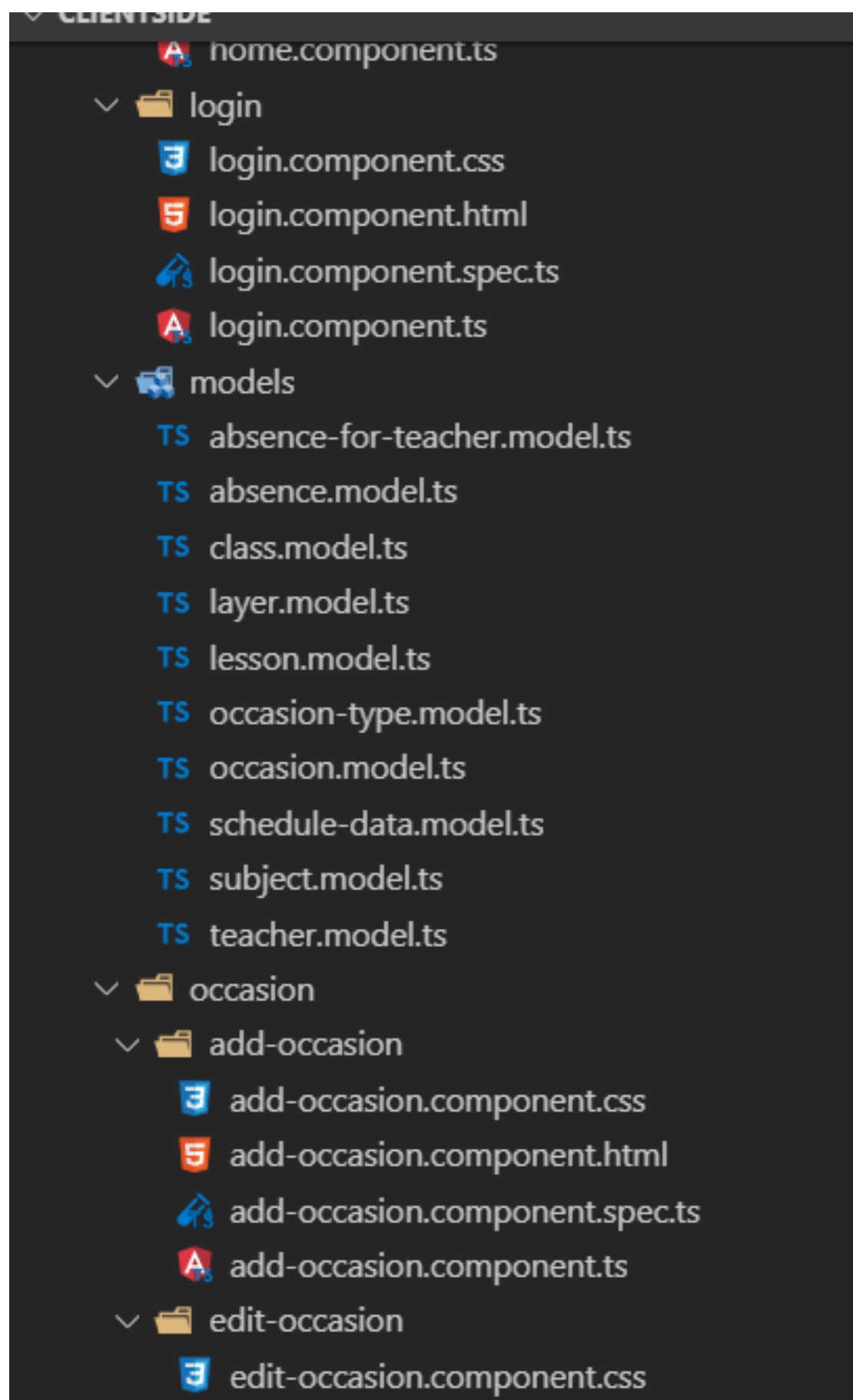
    └─ OccasionDTO.cs
        └─ OccasionDTO
            ├── Id : int
            ├── FromDate : DateDTO
            ├── ToDate : DateDTO
            ├── FromLesson : int?
            ├── ToLesson : int?
            ├── OccasionType : int?
            ├── Dairies : ICollection<int>
            ├── Classes : ICollection<int>
            ├── Rooms : ICollection<int>
            └── Teachers : ICollection<int>

```

























- OccasionTypeDTO.cs
 - OccasionTypeDTO
 - Id : int
 - Name : string
 - packages.config
 - PayAbsenceDTO.cs
 - PayAbsenceDTO
 - Id : int
 - Name : string
 - IsStandart : bool?
 - ReformaDTO.cs
 - ReformaDTO
 - Id : int
 - Name : string
 - RoomDTO.cs
 - RoomDTO
 - Id : int
 - Floor : int
 - Number : int
 - ClassId : int?
 - UseFor : string
 - Dairies : ICollection<int>
 - Schedules : ICollection<int>
 - Groups : ICollection<int>
 - Occasions : ICollection<int>
 - ScheduleDTO.cs
 - ScheduleDTO
 - Num : int
 - WeekDay : int
 - Day : int
 - Hour : int
 - Room : int?
 - SchoolTypeDTO.cs
 - SchoolTypeDTO
 - Id : int
 - Name : string
 - SubjectDTO.cs
 - SubjectDTO
 - Id : int
 - Name : string
 - Dairies : ICollection<int>
 - Classes : ICollection<int>
 - Groups : ICollection<int>

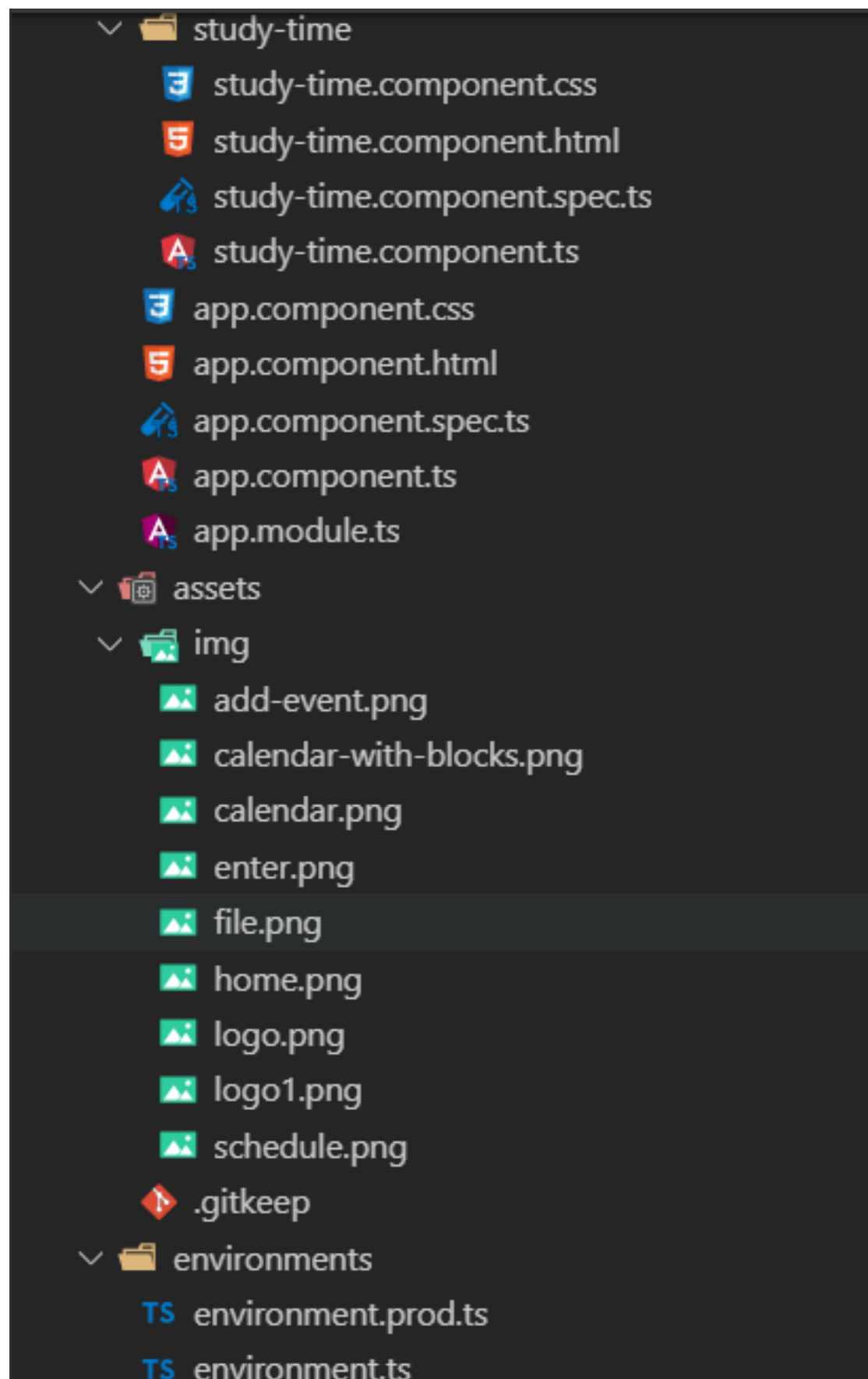
- TeacherDTO.cs
 - TeacherDTO
 - Id : int
 - Num : int
 - Name : string
 - Dairies : ICollection<int>
 - PayAbsences : ICollection<int>
 - Groups : ICollection<int>
 - AbsencesForTeachers : ICollection<int>
 - Occasions : ICollection<int>
 - XmlFilesLoadingImplementation
 - Properties
 - References
 - General
 - SQL_files
 - XML_files
 - App.config
 - ClassDiagramXML.cd
 - packages.config
 - Program.cs
 - Program
 - Main(string[]) : void

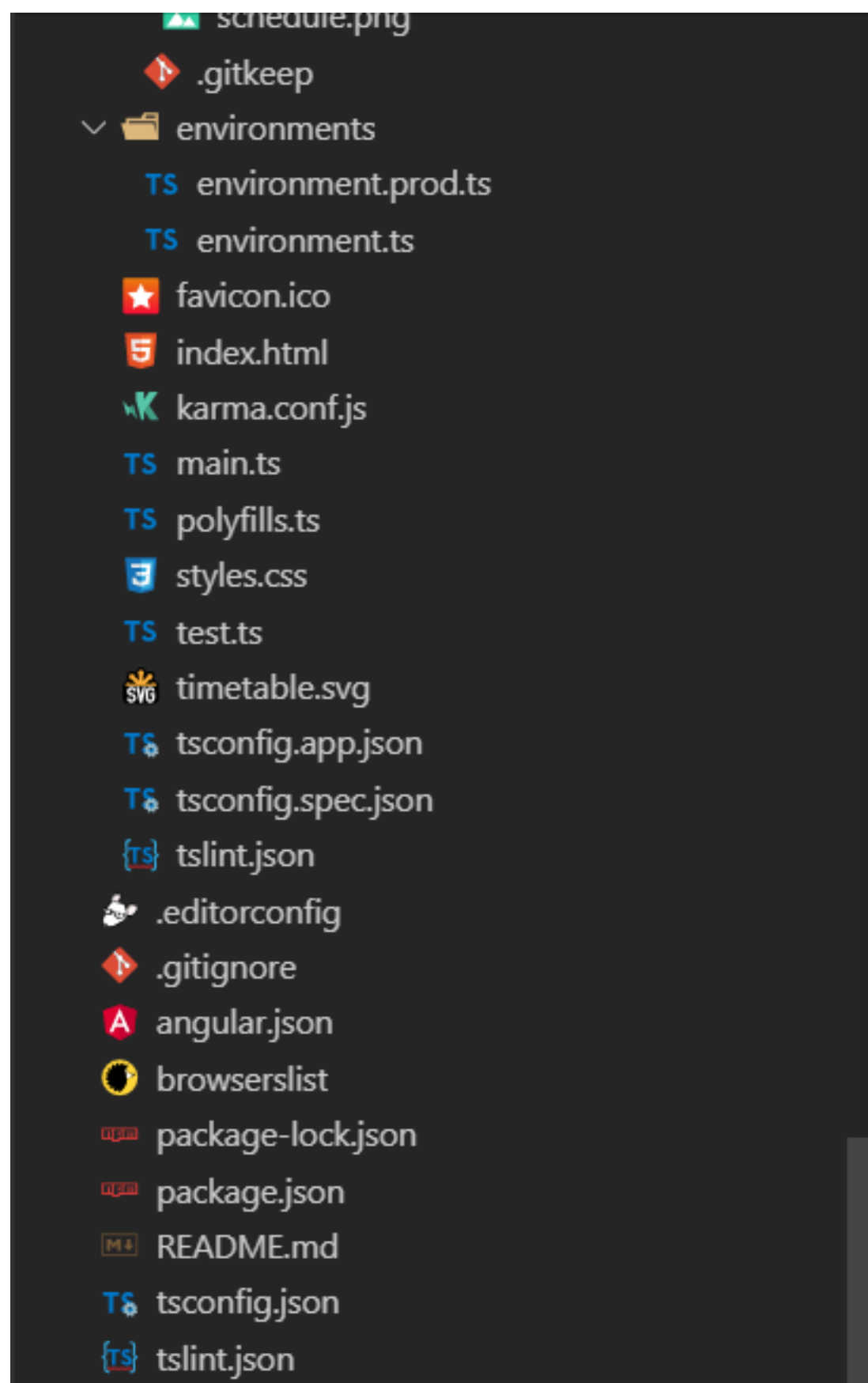
```
src
├── app
│   ├── Absences
│   │   ├── add-absence
│   │   │   ├── add-absence.component.css
│   │   │   ├── add-absence.component.html
│   │   │   ├── add-absence.component.spec.ts
│   │   │   ├── add-absence.component.ts
│   │   ├── view-absence
│   │   │   ├── view-absence.component.css
│   │   │   ├── view-absence.component.html
│   │   │   ├── view-absence.component.spec.ts
│   │   │   ├── view-absence.component.ts
│   ├── guard
│   │   ├── admin.guard.spec.ts
│   │   ├── admin.guard.ts
│   ├── home
│   │   ├── home.component.css
│   │   ├── home.component.html
│   │   ├── home.component.spec.ts
│   │   ├── home.component.ts
```



- edit-occasion
 - edit-occasion.component.css
 - edit-occasion.component.html
 - edit-occasion.component.spec.ts
 - edit-occasion.component.ts
- occasion
 - occasion.component.css
 - occasion.component.html
 - occasion.component.spec.ts
 - occasion.component.ts
- view-occasion
 - view-occasion.component.css
 - view-occasion.component.html
 - view-occasion.component.spec.ts
 - view-occasion.component.ts
- occation
 - add-occasion
 - add-occasion.component.css
 - add-occasion.component.html
 - add-occasion.component.spec.ts
 - add-occasion.component.ts
 - schedule
 - schedule.component.css
 - schedule.component.html

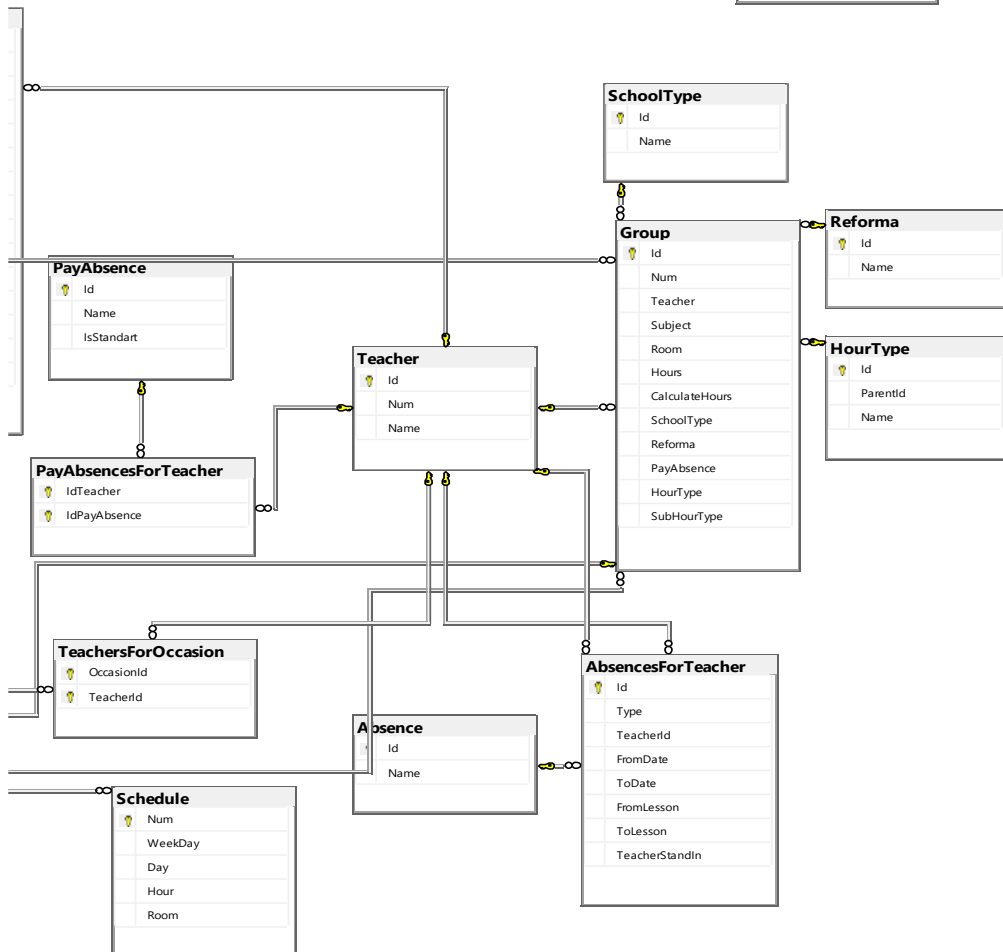
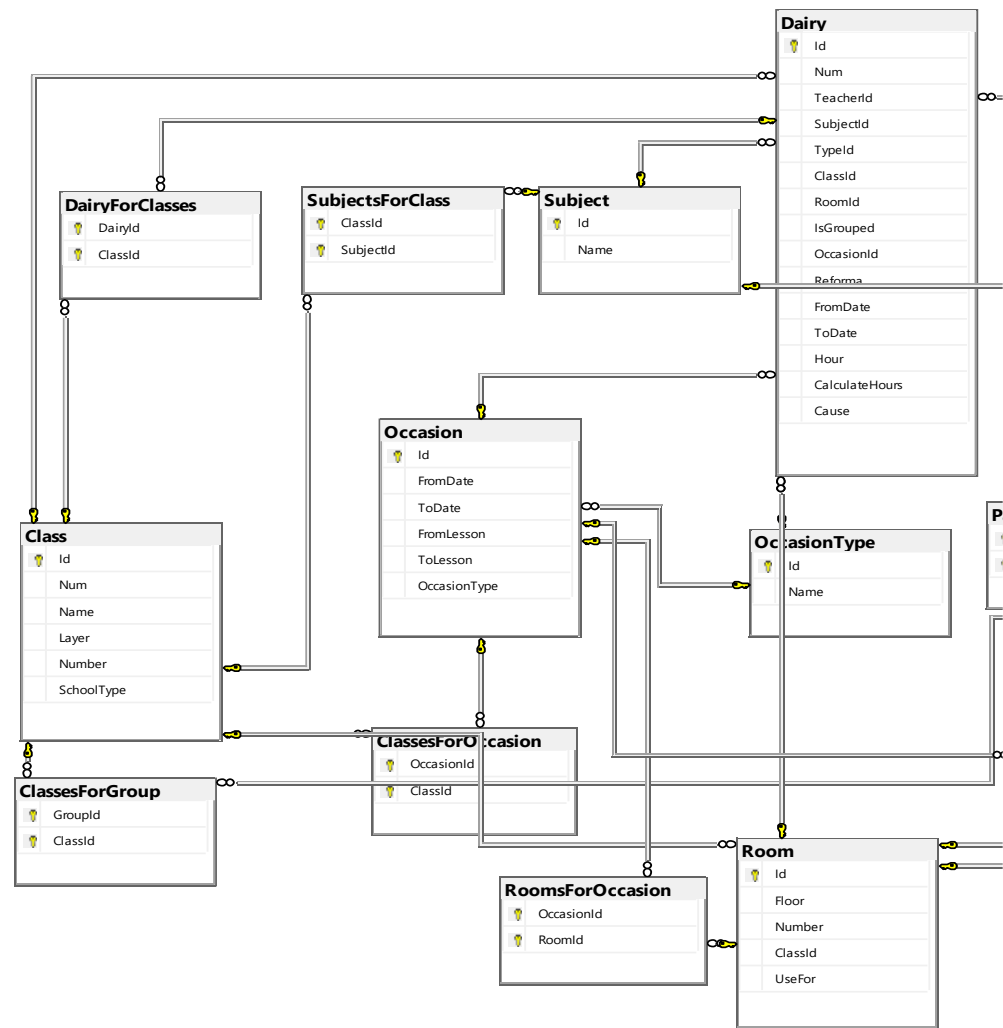
- ▼  schedule
 -  schedule.component.css
 -  schedule.component.html
 -  schedule.component.spec.ts
 -  schedule.component.ts
- ▼  services
 -  absence-for-teacher.service.spec.ts
 -  absence-for-teacher.service.ts
 -  absence.service.spec.ts
 -  absence.service.ts
 -  class.service.spec.ts
 -  class.service.ts
 -  occasion-type.service.spec.ts
 -  occasion-type.service.ts
 -  occasion.service.spec.ts
 -  occasion.service.ts
 -  study-time.service.spec.ts
 -  study-time.service.ts
 -  subject.service.spec.ts
 -  subject.service.ts
 -  teacher.service.spec.ts
 -  teacher.service.ts
 -  user.service.spec.ts
 -  user.service.ts

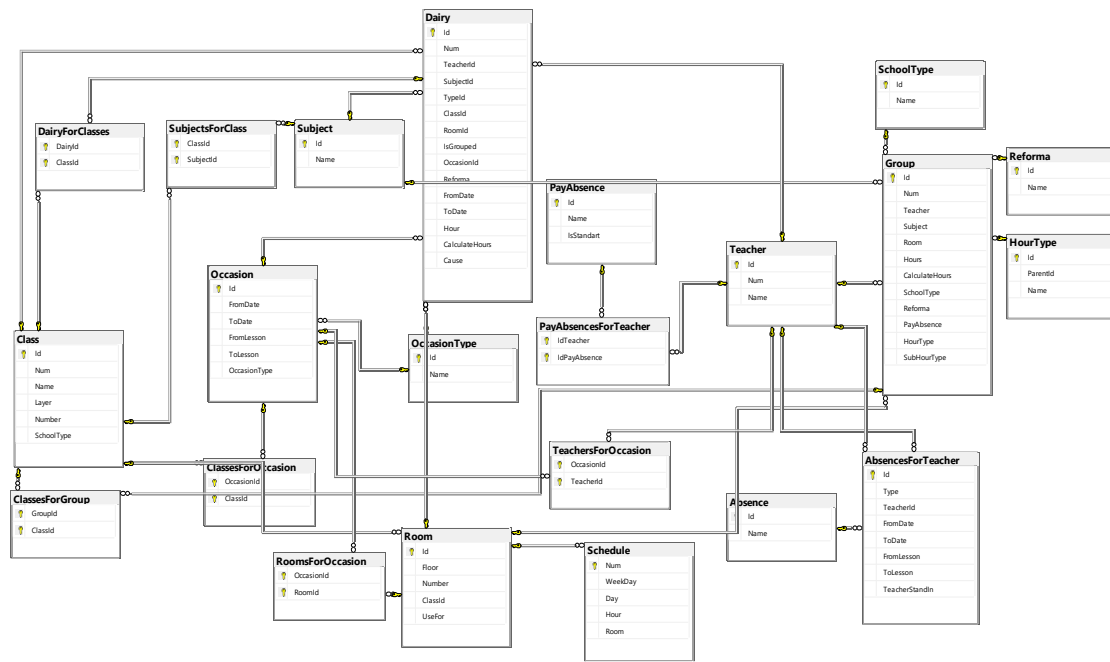




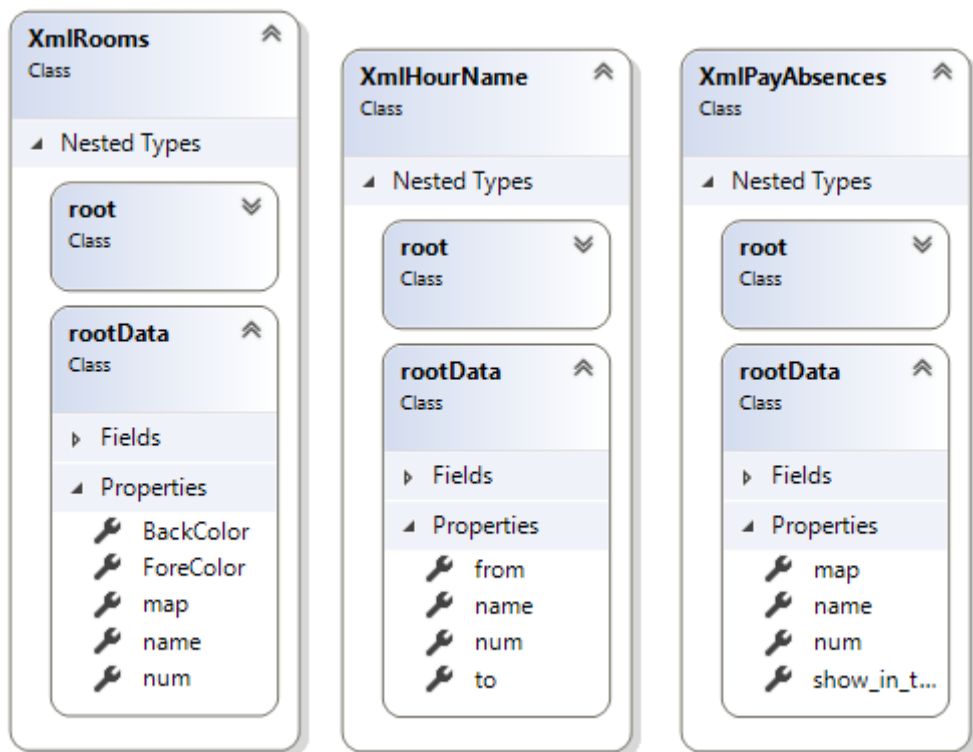
מבנה נתונים

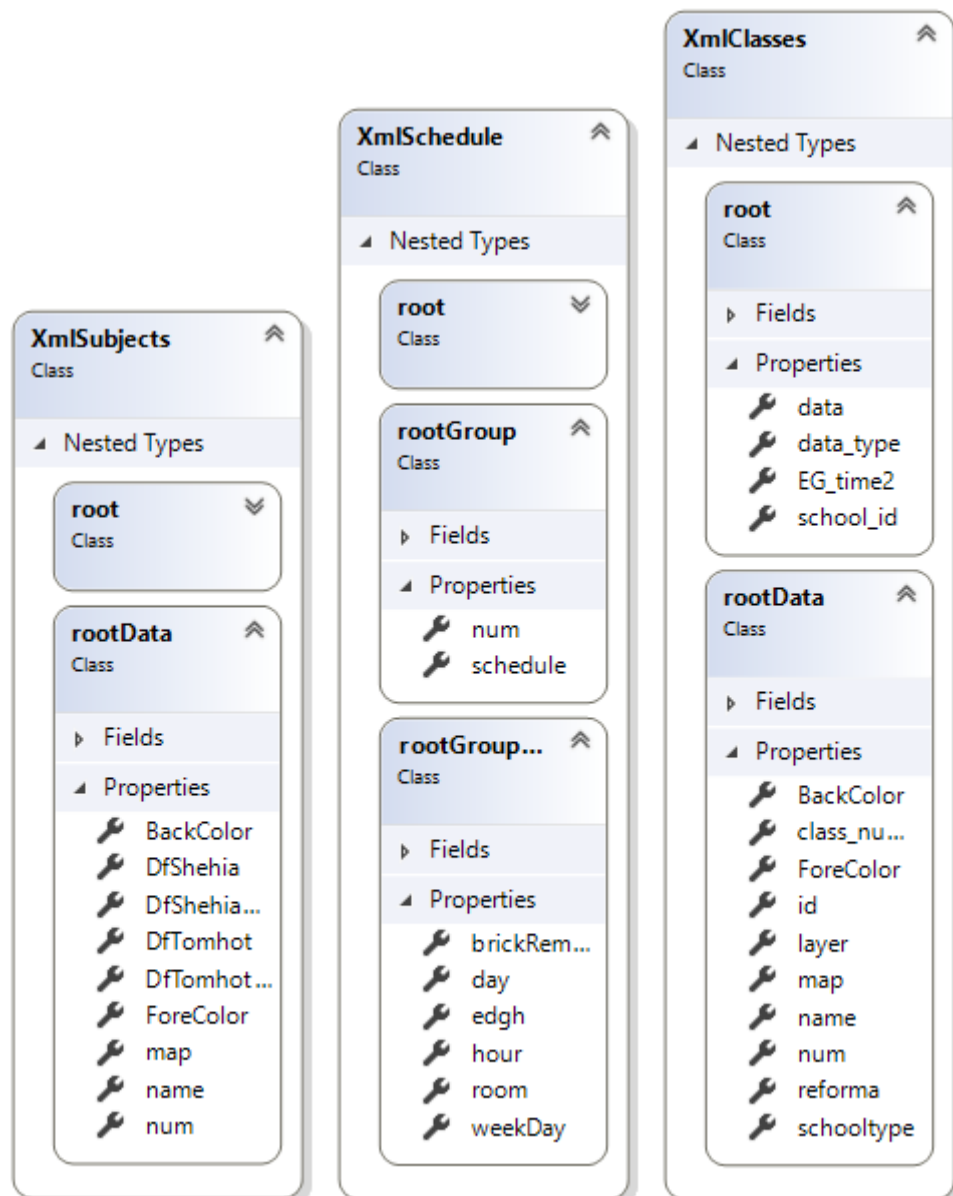
sql-server.!ESTY&TAMAR.dbo		
Tables		
▶ Absence	Group	Schedule
▶ AbsencesForTeacher	Id	Num
Id	Num	WeekDay
Type	Teacher	Day
TeacherId	Subject	Hour
FromDate	Room	Room
ToDate	Hours	▶ SchoolType
FromLesson	CalculateHours	▶ Subject
ToLesson	SchoolType	▶ SubjectsForClass
TeacherStandIn	Reforma	▶ Teacher
Class	PayAbsence	▶ TeachersForOccasion
Id	HourType	
Num	Id	
Name	ParentId	
Layer	Name	
Number	Occasion	
SchoolType	Id	
ClassesForGroup	FromDate	
ClassesForOccasion	ToDate	
Dairy	FromLesson	
Id	ToLesson	
Num	OccasionType	
TeacherId	OccasionType	
SubjectId	PayAbsence	
TypeId	Id	
ClassId	Name	
RoomId	IsStandart	
IsGrouped	PayAbsencesForTeacher	
OccasionId	Reforma	
Reforma	Room	
FromDate	Id	
ToDate	Floor	
Hour	Number	
CalculateHours	ClassId	
Cause	UseFor	
DairyForClasses	RoomsForOccasion	





תיאור מבנה קבצי XML





XmlTeachers
 Class

Nested Types

root
 Class

rootData
 Class

Fields

Properties

BackColor
 details
 ForeColor
 id
 map
 name
 num
 WantedD...

rootDataD...
 Class

Fields

Properties

customRe...
 goremMe...
 goremMe...
 phone1
 phone1S...
 phone2
 phone2S...
 remark
 status_mi...

XmlGroups
 Class

Nested Types

root
 Class

rootGroup
 Class

Fields

Properties

A
 ASpecified
 blockid
 calculate_...
 calculate_...
 classes
 edgh_flag
 from_date
 goremMe...
 grouped
 groupNa...
 map
 mispar_ra...
 mispar_sh...
 no_of_ho...
 num
 reforma
 schooltype
 schooltyp...
 siba
 sroom
 sub_sug
 subj
 subjlevel
 sug
 tea
 tkufa
 tkufaSpec ...
 to_date

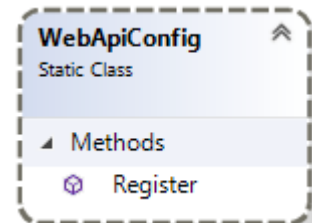
rootGroup...
 Class

תוכן הפרויקט תיאור המחלקות

API

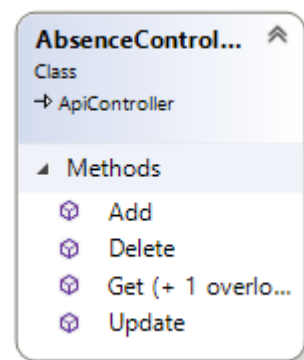
WebApiConfig

מחלקה המטפלת במאפיינים והגדרות של הApplication



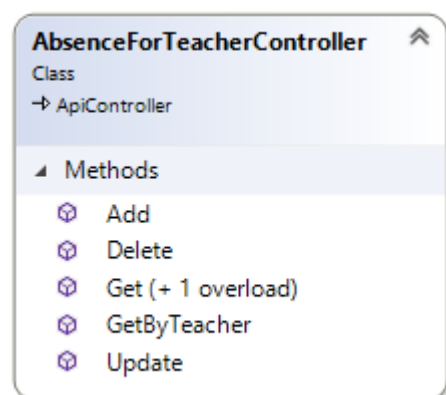
AbsenceController

מחלקה המטפלת בסוגי חיסורים (של מורות)



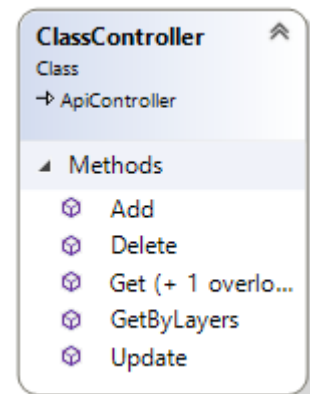
AbsenceForTeacherController

מחלקה המטפלת בחיסורים למורות



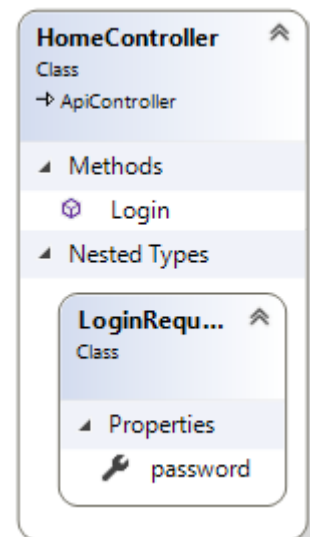
[ClassController](#)

מחלקה המטפלת בביתות



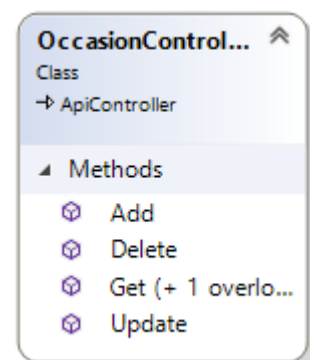
[HomeController](#)

מחלקה המטפלת בהרשאות למערכת



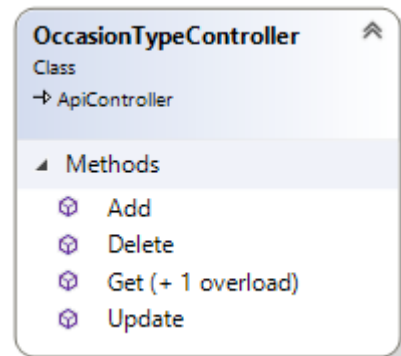
[OccasionController](#)

מחלקה המטפלת באירועים לביתות



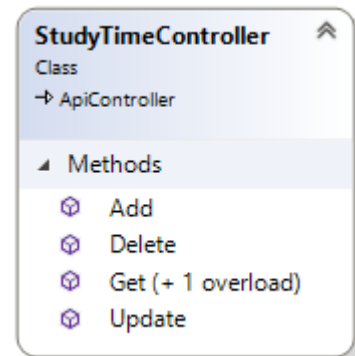
OccasionTypeController

מחלקה המטפלת בסוגי אירועים (של ביתות)



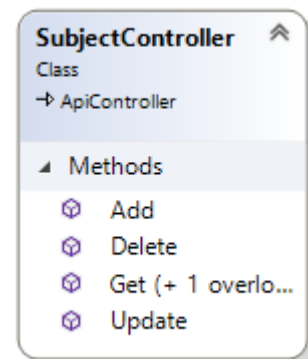
StudyTimeController

מחלקה המטפלת בשעת לימוד



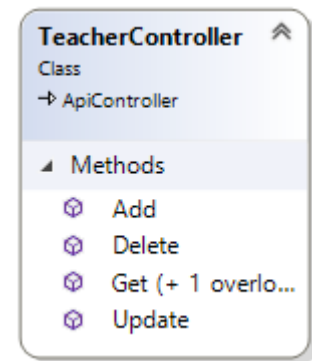
SubjectController

מחלקה המטפלת במקצועות



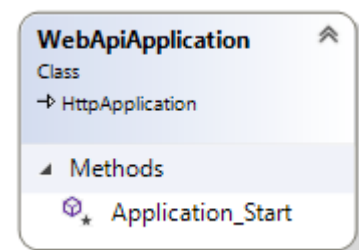
TeacherController

מחלקה המטפלת במורות



[WebApiApplication](#)

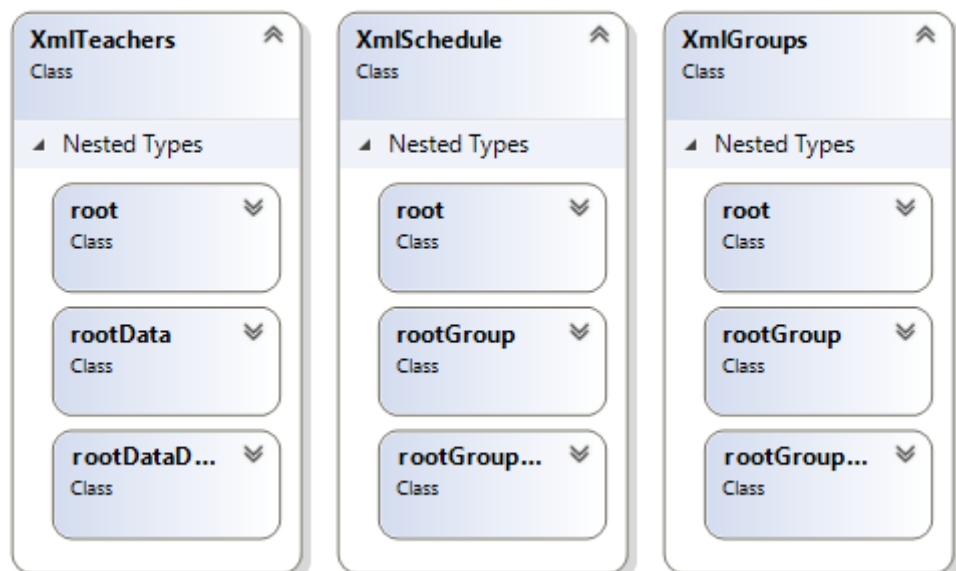
מחלקה המטפלת בבקשות Web בבליות

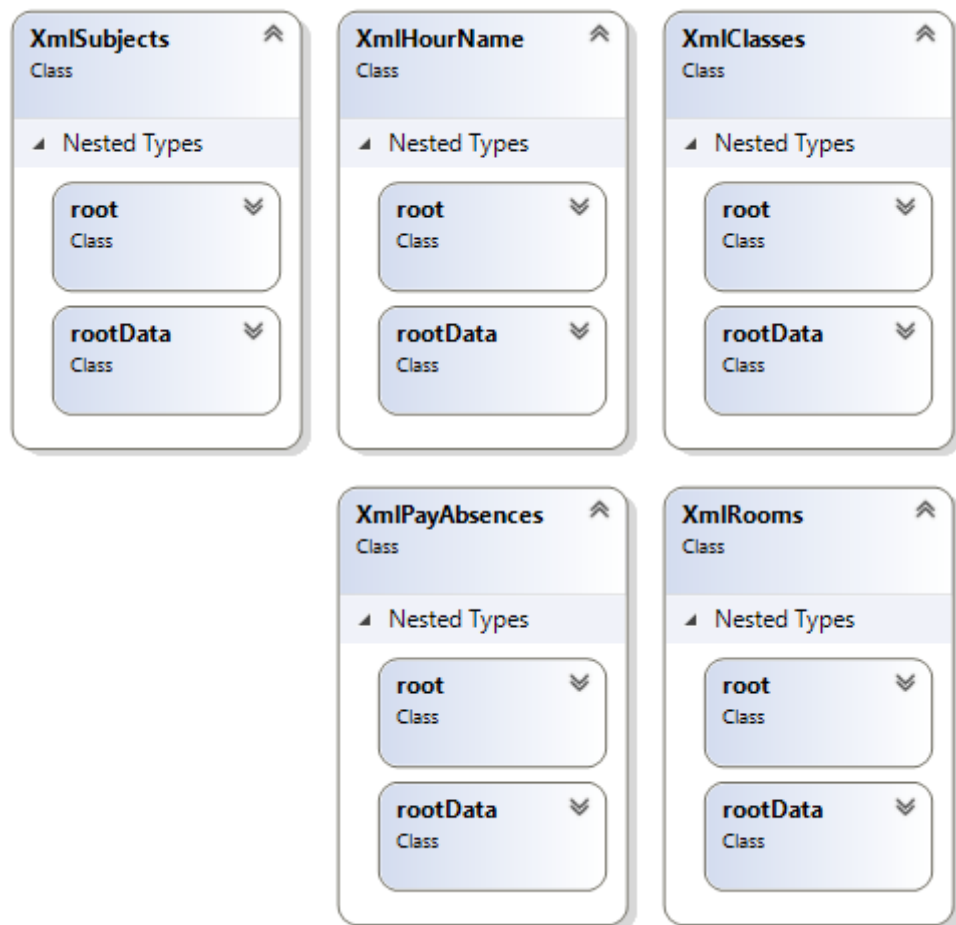


BL

[XmlClassesFiles](#)

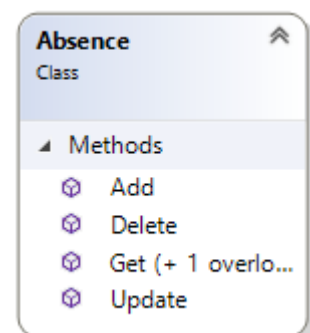
המחלקות הבאות מקבילות לקובץ הxml המתאים





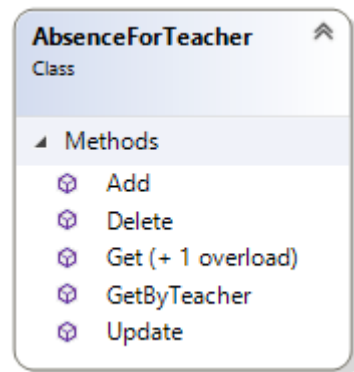
[Absence](#)

מחלקה המנהלת את סוגי חיסורים



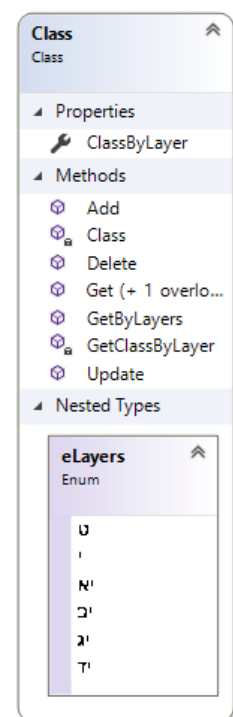
[AbsenceForTeacher](#)

מחלקה המנהלת את חיסורים למורות



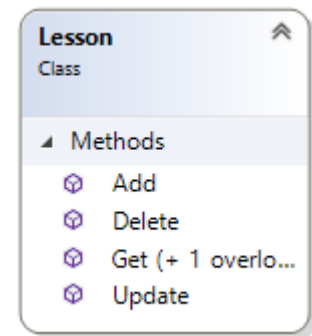
Class

מחלקה המנהלת את ביתות



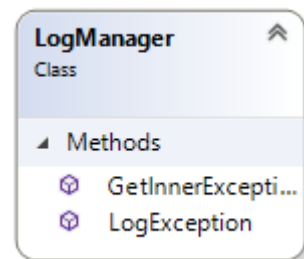
Lesson

מחלקה המנהלת את שעות לימוד (משבצת במערכת)



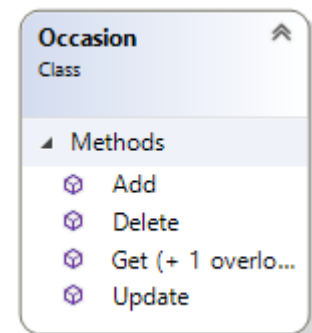
[LogManager](#)

מחלקה המנהלת את Log –ים



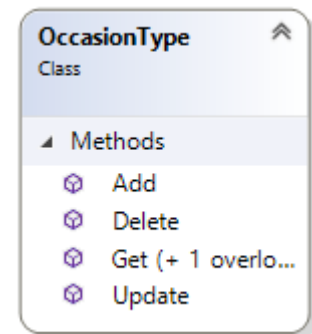
[Occasion](#)

מחלקה המנהלת את אירועים לכיתות



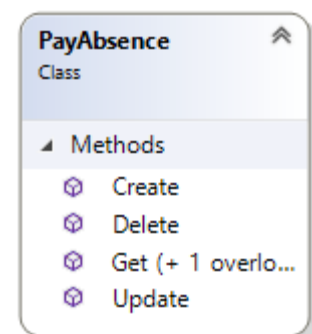
[OccasionType](#)

מחלקה המנהלת את סוגי אירועים



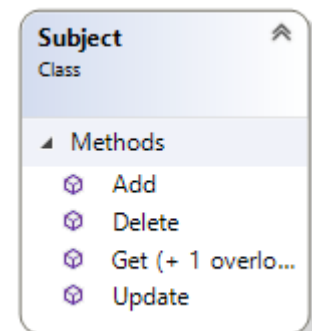
PayAbsence

מחלקה המנהלת את גורמים משלמים לחיסורי מורות



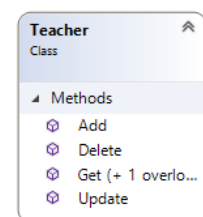
Subject

מחלקה המנהלת את מקצועות

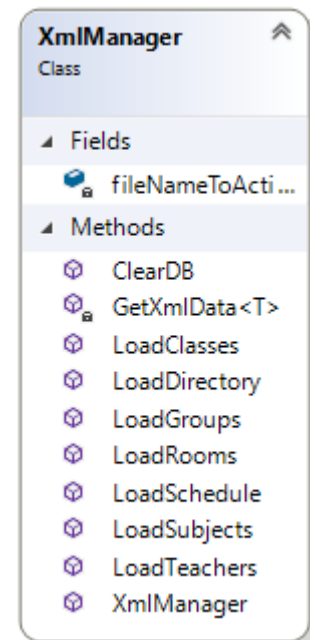


Teacher

מחלקה המנהלת את מורות



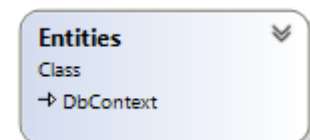
מחלקה המנהלת את קבצי הxml



DAL

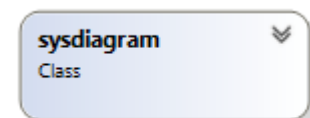
Entities

מחלקה המכילה אוספים המאפשרים לקבל נתונים מאובייקטים עיקריים במסד הנתונים



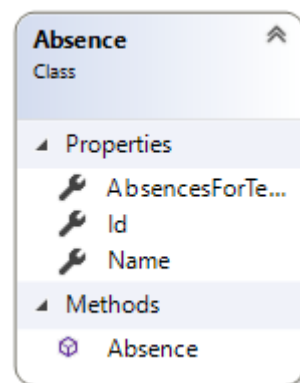
Sysdiagram

מחלקה המכילה את נתוני הדיאגרמות במסד הנתונים



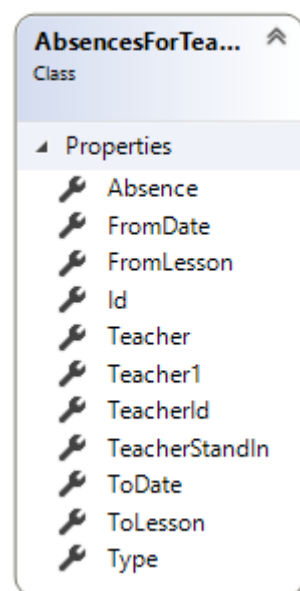
[Absence](#)

המחלקה מכילה את צורת אובייקט סוג חיסור של מורה במסד הנתונים



[AbsencesForTeacher](#)

המחלקה מכילה את צורת אובייקט חיסור למורה במסד הנתונים

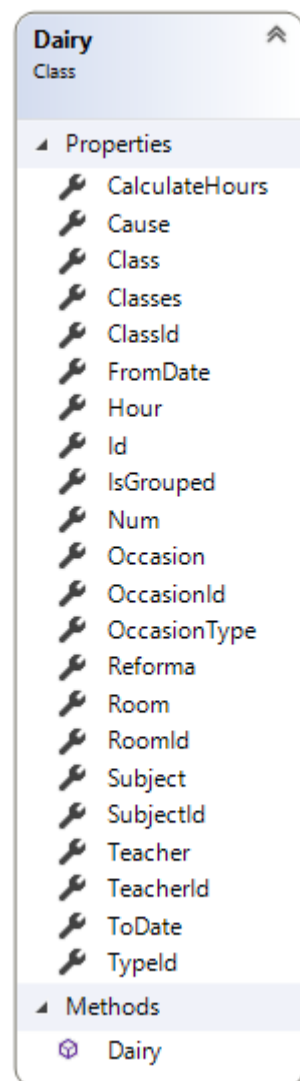


המחלקה מכילה את צורת אובייקט ביתה במסד הנתונים



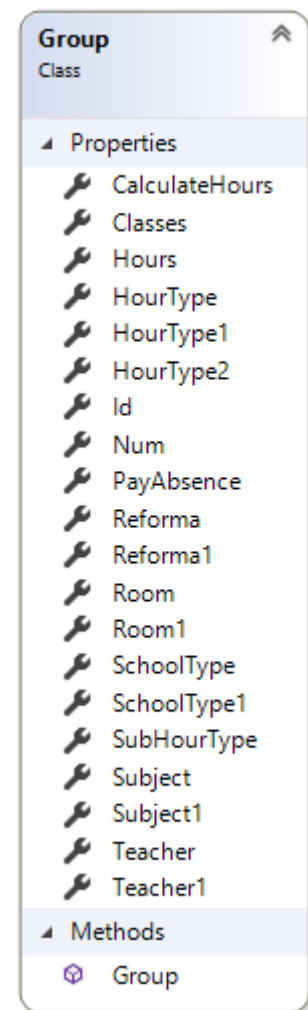
Dairy

המחלקה מכילה את צורת אובייקט יומן לימודים במסד הנתונים



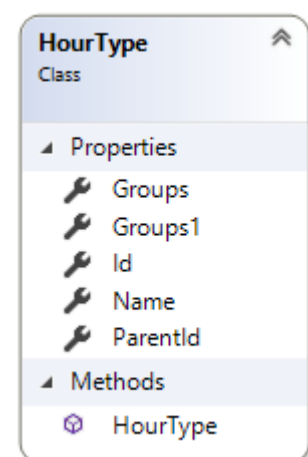
Group

המחלקה מכילה את צורת אובייקט שיעור הקבצה או איחוד במסד הנתונים

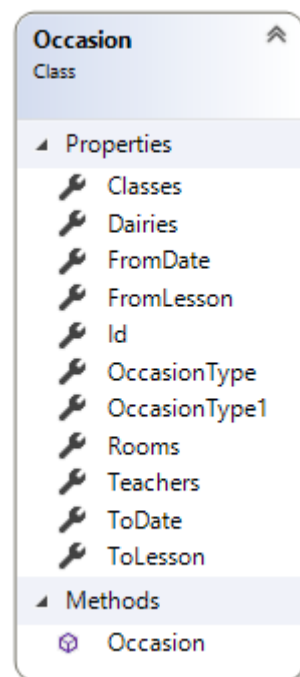


HourType

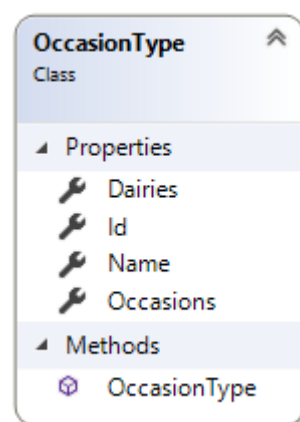
המחלקה מכילה את צורת אובייקט סוג שעת לימוד במסד הנתונים



המחלקה מכילה את צורת אובייקט אירוע לכיתות במסד הנתונים

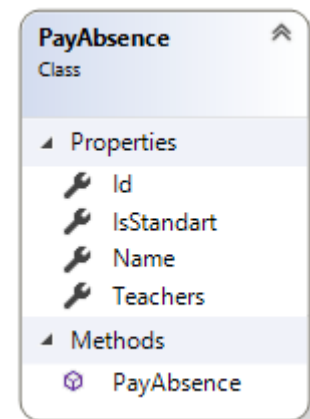


המחלקה מכילה את צורת אובייקט סוג ארוע במסד הנתונים



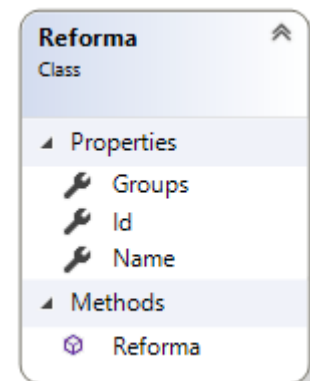
PayAbsence

המחלקה מכילה את צורת אובייקט גורם משלם במסד הנתונים



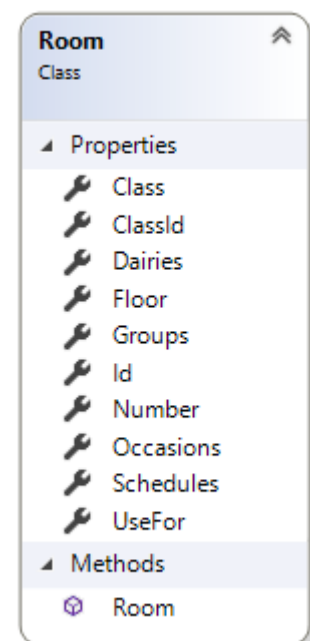
Reforma

המחלקה מכילה את צורת אובייקט רפורמה במסד הנתונים



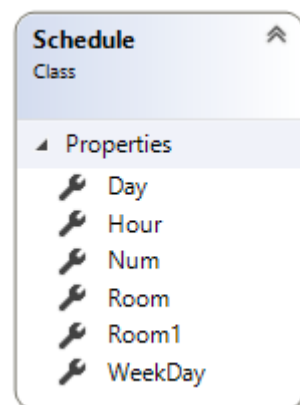
Room

המחלקה מכילה את צורת אובייקט חדר במסד הנתונים



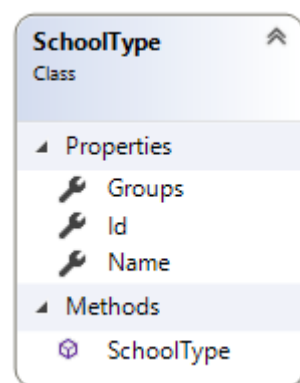
[Schedule](#)

המחלקה מכילה את צורת אובייקט מסגרת מערכת השעות הקבועה במסד הנתונים



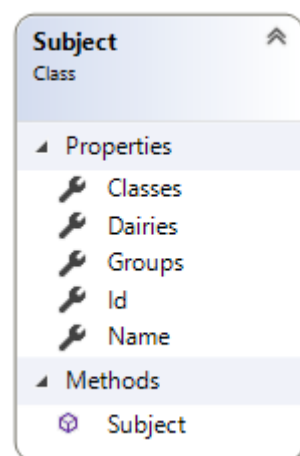
[SchoolType](#)

המחלקה מכילה את צורת אובייקט סוג ביה"ס במסד הנתונים

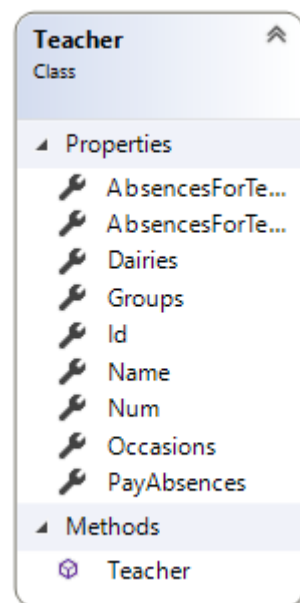


[Subject](#)

המחלקה מכילה את צורת אובייקט מקצוע במסד הנתונים



המחלקה מכילה את צורת אובייקט מורה במסד הנתונים

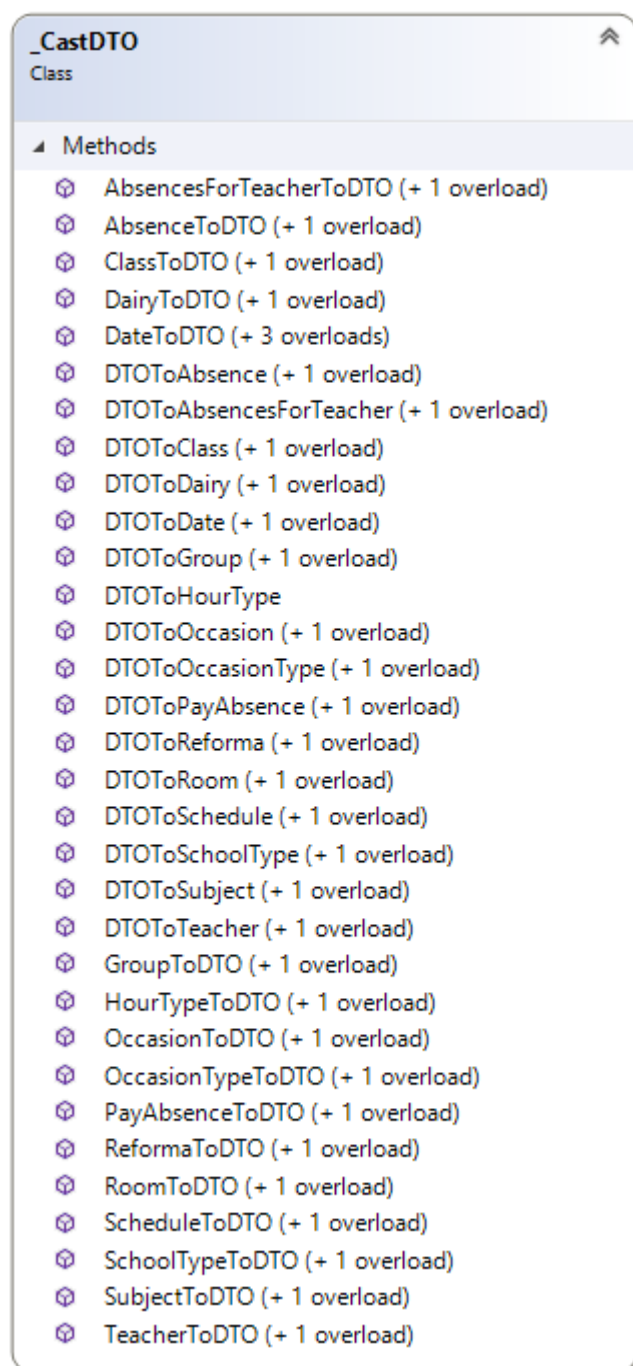


DTO

מחלקות המכילות טיפוס נתונים מקבילים לשכבת הDTO



מחלקה המכילה מתודות המרה משכבת הDAL לשכבת הDTO ולהיפך



מחלקות מקבילות למחלקות בשכבת Daln
ללא שדות ניווט

AbsenceDTO
Class

Properties

Id

Name

AbsencesForTea...
Class

Properties

FromDate

FromLesson

Id

TeacherId

TeacherStandIn

ToDate

ToLesson

Type

ClassDTO
Class

Properties

Dairies

Dairies1

Groups

Id

Layer

Name

Num

Number

Occasions
















Rooms

SchoolType

Subjects














DairyDTO
Class

▲ Properties

 CalculateHours
 Cause
 ClassId
 FromDate
 Hour
 Id
 IsGrouped
 Num
 OccasionId
 Reforma
 RoomId
 SubjectId
 TeacherId
 ToDate
 TypeId




GroupDTO
Class


▲ Properties

 CalculateHours
 Classes
 Hours
 HourType
 Id
 Num
 PayAbsence
 Reforma
 Room
 SchoolType
 SubHourType
 Subject
 Teacher


HourTypeDTO
Class


▲ Properties


 Id
 Name
 ParentId


OccasionDTO


Class


 Properties


 Classes


 Dairies


 FromDate


 FromLesson


 Id


 OccasionType

 Rooms


 Teachers


 ToDate


 ToLesson


OccasionTypeDTO

Class


 Properties


 Id


 Name


PayAbsenceDTO


Class

 Properties


 Id


 IsStandart


 Name

ReformaDTO

Class










 Properties

 Id

 Name






RoomDTO
Class

▲ Properties

 ClassId
 Dairies
 Floor
 Groups
 Id
 Number
 Occasions
 Schedules
 UseFor



ScheduleDTO
Class

▲ Properties

 Day
 Hour
 Num
 Room
 WeekDay






SchoolTypeDTO
Class

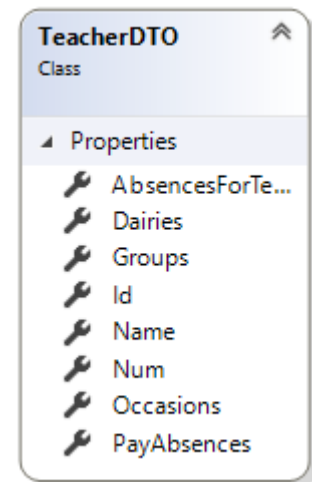
▲ Properties

 Id
 Name

SubjectDTO
Class

▲ Properties

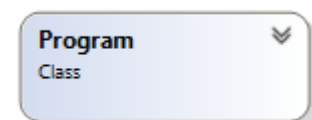
 Classes
 Dairies
 Groups
 Id
 Name



XmlFilesLoadingImplementation

Program

מחלקה המכילה קוד לטעינת הנתונים מקבצי הxml



פונקציות מעניינות

API

Login

```
[HttpPost]//define the data transfer method
[Route("login")]//define the url
public IActionResult Login([FromBody]LoginRequest loginRequest)
{
    if (loginRequest.password.Equals(WebConfigurationManager.AppSettings["password"])) {
        return StatusCode(HttpStatusCode.NoContent);
    }
    return Unauthorized();
}
public class LoginRequest
{
    public string password { get; set; }
}
```

בודקת האם הסיסמה נכונה, ז"א נגשת לקובץ הקונפיגורציה ובודקת בהגדרות מה ה value של הסיסמה, אם הסיסמאות שוות- מחזירה תגובה (ריקה אמנם, כי אין מה להחזיר), אחרת מחזירה תגובה מסוג "לא מאופשר".

לצורך קבלת הסיסמה, לא קבלנו במתודה מחרוזת רגילה, אלא יצרנו מחלקה שעוטפת (Wrapper), מכיוון ש Angularש יכול רק לקבל או/ו להחזיר אובייקטים ולא פרימיטיביים (Value Type).

BL

GetsByLayers

רצינו להחזיר אוסף כיתות לפי מחזור/ים.

```
public static Dictionary<int, List<ClassDTO>> ClassByLayer { get; } = new Dictionary<int, List<ClassDTO>>();
```

יצרנו Dictionary שהמפתח הוא מספר מחזור והערך הוא אוסף הכיתות של המחזור.

```
static Class()
{
    for (int i = 9; i < 15; i++)
    {
        ClassByLayer.Add(i, GetClassByLayer(i));
    }
}
```

בבונה הסטטית של מחלקת כיתה ה Dictionary מתאחל בנתונים.

```
private static List<ClassDTO> GetClassByLayer(int layer)
{
    using (Entities db = new Entities())
    {
        return _CastDTO.ClassToDTO(db.Classes.Where(c => c.Layer == layer).ToList());
    }
}
```

מחזירה אוסף כיתות לפי מספר מחזור. כיון שהעניין דורש גישה למבנה הנתונים, נמצאה דרך יעילה להוריד את הגישות למסד הנתונים כמה שפחות, לכן מתודה זו פרטית.

```
public static List<ClassDTO> GetByLayers(List<int> layers)
{
    List<ClassDTO> classes = new List<ClassDTO>();
    if (layers.Contains(0))
    {
        for (int i = 9; i < 15; i++)
        {
            classes.AddRange(ClassByLayer[i]);
        }
        return classes;
    }
    foreach (int i in layers)
    {
        classes.AddRange(ClassByLayer[i]);
    }
    return classes;
}
```

באן המתודה מקבלת אוסף מחזורים, ומחזירה כיתות לפי האוסף. אם קיים מספר 0 באוסף הכונה לכל המחזורים, אחרת עוברים לפי האוסף ומוסיפים את הכיתות המתאימות מה Dictionary.

LoadDirectory
LoadClasses
GetXmlData

DTO

DateToDTO

```
public static DateDTO DateToDTO(DateTime date)
{
    return new DateDTO()
    {
        Day = date.Day,
        Hour = date.Hour,
        Millisecond = date.Millisecond,
        Minute = date.Minute,
        Month = date.Month,
```

```

        Second = date.Second,
        Year = date.Year
    };
}

```

ממירה מאובייקט `DateTime` של C#, בחלק מתהליך שליפת נתון, לאובייקט מסוג `DateTime` (מחלקה שיצרנו, מכיוון שאובייקט תאריך ב Angular הוא `Date` בלבד, והפרמטרים לא הצליחו להיבסס כראוי)

```

public static DateTime DTOToDateTime(DateTime? date)
{
    DateTime thisDate = new DateTime();
    return new DateTime()
    {
        thisDate = date.GetValueOrDefault(),
        Hour = thisDate.Hour,
        Millisecond = thisDate.Millisecond,
        Minute = thisDate.Minute,
        Month = thisDate.Month,
        Second = thisDate.Second,
        Year = thisDate.Year
    };
}

```

ממירה מאובייקט `DateTime` של C#, בחלק מתהליך שליפת נתון, לאובייקט מסוג `DateTime`. פונקציה מועמסת-נוספת כדי להימנע מנפילה במקרה שהתאריך ריק.

```

public static List<DateTime> DTOToDateTime(List<DateTime> dateTimeList)
{
    return dateTimeList.Select(d => DTOToDateTime(d)).ToList();
}

```

ממירה מאוסף אובייקטים `<DateTime>` של C#, בחלק מתהליך שליפת נתון, לאוסף אובייקטים מסוג `<DateTime>`.

DTOToDate

```

public static DateTime DTOToDate(DateTime date)
{
    return new DateTime(date.Year, date.Month, date.Day, date.Hour, date.Minute,
        date.Second, date.Millisecond);
}

```

ממירה מאובייקט `DateTime`, בחלק מתהליך שמירת נתון, לאובייקט מסוג `DateTime` של C#.

```

public static List<DateTime> DTOToDate(List<DateTime> dateTimeList)
{
    return dateTimeList.Select(d => DTOToDate(d)).ToList();
}

```

ממירה מאוסף אובייקטים `<DateTime>`, בחלק מתהליך שמירת נתון, לאוסף אובייקטים מסוג `<DateTime>` של C#.

מנגנון טיפול בשגיאות-*LogManager*

פונקציה זו כותבת את הודעת השגיאה (המלאה) ל – Event Viewer:

```
public static void LogException(Exception e)
{
    using (EventLog eventLog = new EventLog())
    {
        eventLog.Source = "SchoolSchedule";
        eventLog.WriteEntry(GetInnerExceptionMessage(e), EventLogEntryType.Error);
    } /**/
}
```

פונקציה זו מחזירה את כל הודעות השגיאות הפנימיות משורשרות זו לזו:

```
public static string GetInnerExceptionMessage(Exception e)
{
    if (e.InnerException != null)
    {
        return e.Message + "\n" + GetInnerExceptionMessage(e.InnerException);
    }

    return e.Message + "\n" + e.StackTrace +
    "\n*****\n\n";
}
```

מנגנון טעינת קבצים ואתחול מערכת-XmlManager

```
public XmlManager()
{
    fileNameToAction = new Dictionary<string, Action<string>>();
    fileNameToAction["classes.xml"] = LoadClasses;
    fileNameToAction["groups.xml"] = LoadGroups;
    fileNameToAction["rooms.xml"] = LoadRooms;
    fileNameToAction["schedule.xml"] = LoadSchedule;
    fileNameToAction["subjects.xml"] = LoadSubjects;
    fileNameToAction["teachers.xml"] = LoadTeachers;
}
```

פונקציה זו טוענת את תקיית קבצי הXML:

```
public void LoadDirectory(string path)
{
}
```

```

//reset the DB

ClearDB();

var directoryInfo = new DirectoryInfo(path);

var files = directoryInfo.GetFiles();

foreach (var file in files)
{
    Action<string> action;

    if (fileNameToAction.TryGetValue(file.Name, out action))
    {
        action?.Invoke(file.FullName);

        Console.WriteLine("{0} Loading....", file.FullName);
    }
}
}
}

```

פונקציה זו מאתחלת את ה DB מחדש:

```

public void ClearDB()
{
    var conn = new SqlConnection("data source=DESKTOP-7A0S24C; initial
catalog=!ESTY&TAMAR; integrated security=True");

    SqlCommand cmd;

    conn.Open();

    cmd = new SqlCommand("sp_MSforeachtable 'DELETE FROM ?'", conn);
    cmd.ExecuteNonQuery();

    cmd = new SqlCommand("sp_MSforeachtable 'truncate table ?'", conn);
    cmd.ExecuteNonQuery();

    conn.Close();
}

```

פונקציה זו טוענת קבצי כיתות (XML):

```
public void LoadClasses(string path)
```

פונקציה זו טוענת קבצי קבוצות (XML):

```
public void LoadGroups(string path)
```

פונקציה זו טוענת קבצי חדרים (XML):

```
public void LoadRooms(string path)
```

פונקציה זו טוענת קבצי מערכת קבועה (XML):

```
public void LoadSchedule(string path)
```

פונקציה זו טוענת קבצי מקצועות (XML):

```
public void LoadSubjects(string path)
```

פונקציה זו טוענת קבצי מורות (XML):

```
public void LoadTeachers(string path)
```

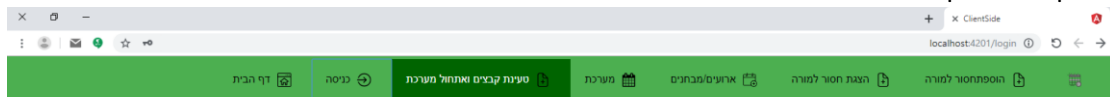
פונקציה זו ממירה את הטקסט של הקובץ לאובייקטים:

```
private static T GetXmlData<T>(string path)
{
    XmlSerializer serializer = new XmlSerializer(typeof(T));
    StringReader stringReader = new StringReader(File.ReadAllText(path,
Encoding.GetEncoding(1255)));
    T root = (T)serializer.Deserialize(stringReader);
    return root;
}
```

מדריך למשתמש

הוראות כלליות לשימוש במערכת:

כל חודש יש לטעון את קבצי הנתונים.
להכניס את הניתוב של הקבצים המתאימים.
ללחוץ "טעינת קבצים ואתחול מערכת".



School Scedule

סיסמה

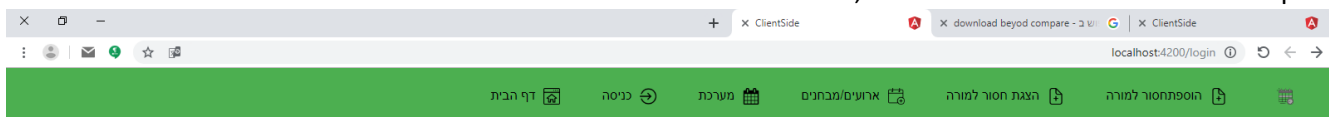
...

כניסה

כניסה כאורח

שכחת סיסמה?

לצורך כניסה למערכת נדרשת הכנסת סיסמה, יש אפשרות להיכנס כאורח עם הרשאות מוגבלות.



School Scedule

סיסמה

הזיני קוד כניסה למערכת

כניסה

כניסה כאורח

שכחת סיסמה?

כאשר מנסים ללחוץ על כפתור "כניסה" ללא הכנסת סיסמה, הודעה מתאימה מוצגת.

ClientSide

download beyod compare - שיב

ClientSide

localhost:4200/login

דף הבית כניסה משרת ארועיס/מבחרים הצגת חסור למורה הספתחוסור למורה

School Scedule

סיסמה

הזיני קוד כניסה למערכת

הסיסמה הכרחית

כניסה

כניסה כאורח שכחת סיסמה?

אם הסיסמה שגויה הדף מתרענן, והמשתמש יכול להכניס סיסמה שוב.

כאשר מורה נעדרת, יש למלא טופס הכולל את פרטי ההיעדרות.

העדרות למורה

מורה חסרה

yyyy-mm-dd

yyyy-mm-dd

סוג חסור

משיעור

עד שיעור

טיפול יומי גורף

אישור

לבחור את המורה הנעדרת מרשימת המורות, זהו שדה חובה.

העדרות למורה

מורה חסרה

אדלר ב

אדלר ט

אדלר מ

אדלר ש

אדלשטיין ה

אולמן ד

אולמן ה

איונשטיין ח

איונשטיין ש

איתן י

איתן מ

אלחרר ש"צ

אלטוסקיח

אלפא ש

אלפר ג

אנגל פ

אריאלי ש

בביון.ט

בוגץ ש

בומרינד ב

אישור

לבחור את תאריך תחילת ההיעדרות מתוך לוח השנה, זהו שדה חובה.
 לבחור את תאריך סיום ההיעדרות כנ"ל, זהו שדה רשות, אם לא ממלאים אותו הוא מתמלא בתאריך תחילת ההיעדרות.

לבחור את סוג החיסור המתאים מתוך הרשימה.

לבחור את שעור תחילת ההיעדרות מתוך הרשימה, זהו שדה חובה.
 לבחור את שעור סיום ההיעדרות כנ"ל, זהו שדה רשות, אם לא ממלאים אותו הוא מתמלא בשעור תחילת ההיעדרות.

כאשר מסמנים את תיבת "טיפול יומי גורף", הכוונה שיש ממלאת מקום לכל השעות של המורה הנעדרת.
 ואז מתוסף שדה "ממלאת מקום" לבחירת מורה מרשימת המורות.

לסיום, ללחוץ על אישור.

ClientSide download beyod compare - ט ב - ClientSide localhost:4200/absence-for-teacher/add

דף הבית | כניסה | מערכת | ארועים/מבחנים | הצגת חסור למורה | הוספתחסור למורה

School Scedule

העדרות למורה

מורה חסרה | אדלר ב

2019-09-09

2019-09-09

סוג חסור | שמחה משפחתית

משיעור | ראשון

עד שיעור: | שמיני

☒ טיפול יומי גורף

מחלאת מקום | בוגץ ש

אישור

טופס עדכון היעדרות למורה, יש לעדכן את השדות הרצויים. וללחוץ "עדכון".

ClientSide download beyod compare - ט ב - ClientSide localhost:4200/absence-for-teacher/view

דף הבית | כניסה | מערכת | ארועים/מבחנים | הצגת חסור למורה | הוספתחסור למורה

School Scedule

עדכון העדרות למורה

מורה חסרה | אדלר ט

2019-09-09

2019-09-09

סוג חסור | שמחה משפחתית

☐ משיעור

עד שיעור: |

☒ טיפול יומי גורף

מחלאת מקום | בוגץ ש

עדכון

כאשר מוסיפים אירוע למערכת, יש למלא טופס הכולל את פרטי האירוע.

דף הבית

כניסה

מערכת

אירועים/מבחנים

הצגת חסור למורה

הוספת חסור למורה

הוספת אירוע

School Scedule

הוספת אירוע

סוג אירוע

yyyy-mm-dd

yyyy-mm-dd

עד שעות

מלוח

בחירת מלוח

בחירת מחזור

כל המחזורים

ט

י

יא

יב

יג

יד

בחירת כיתות

הטופס אינו מושלם

אישור

כל עוד הטופס אינו תקין (שדות חובה לא מולאו וכדו'), תוצג האזהרה הבאה "הטופס אינו מושלם".
ואזהרות ליד השדות הבעייתיים.
לבחור את סוג האירוע המתאים מתוך הרשימה.

download beyond compare - יו 3

ClientSide

localhost:4200/occasion/add

דף הבית

כניסה

מערכת

אירועים/מבחנים

הצגת חסור למורה

הוספת חסור למורה

הוספת אירוע

School Scedule

הוספת אירוע

סוג אירוע

טויל

תוכנית

הרצאה

מבחן

מחנה

מחזור

ט

י

יא

יב

יג

יד

בחירת מלוח

בחירת מחזור

כל המחזורים

ט

י

יא

יב

יג

יד

בחירת כיתות

הטופס אינו מושלם

אישור

לבחור את תאריך תחילת האירוע מתוך לוח השנה, זהו שדה חובה.
 לבחור את תאריך סיום האירוע כנ"ל, זהו שדה רשות, אם לא ממלאים אותו הוא מתמלא בתאריך תחילת האירוע.

לבחור את שעות תחילת ההיעדרות מתוך הרשימה, זהו שדה חובה.
 לבחור את שעות סיום ההיעדרות כנ"ל, זהו שדה רשות, אם לא ממלאים אותו הוא מתמלא בשעות תחילת ההיעדרות.

לסמן את המלווה/ות מתוך הרשימה.
 ניתן לחפש בתיבת החיפוש המתאימה.
 ישנה אפשרות לסמן את כל המורות.

לסמן את הכיתות המתאימות.
בעת הטופס מוכן לשליחה.

כל המחזורים

ט ☒

י ☐

יא ☐

יב ☐

יג ☐

יד ☐

בחירת כיתות

טו ☐

טז ☐

טז ☐

טז ☐

טז ☐

טז ☐

טז ☐

טז ☐

טז ☐

טז ☐

טז ☒

טז ☐

טז ☐

טז ☒

הטופס מוכן

תצוגת מערכת השעות.

School Scedule

המערכת

ראשון	שני	שלישי	רביעי	חמישי	ששי	
מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	שעור 1
מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	שעור 2
מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	שעור 3
מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	שעור 4
מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	מורה שרה מקצוע מתמטיקה	שעור 5

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

מנגנון אתחול מערכת

```
/// <summary>
/// this class take care of all the loadings of the XML files
/// </summary>
public class XmlManager
{
    private Dictionary<string, Action<string>>> fileNameToAction;
    /// <summary>
    /// each XML file has is his loading function.
    /// </summary>
    public XmlManager()
    {
        fileNameToAction = new Dictionary<string, Action<string>>>();
        fileNameToAction["classes.xml"] = LoadClasses;
        fileNameToAction["groups.xml"] = LoadGroups;
        fileNameToAction["rooms.xml"] = LoadRooms;
        fileNameToAction["schedule.xml"] = LoadSchedule;
        fileNameToAction["subjects.xml"] = LoadSubjects;
        fileNameToAction["teachers.xml"] = LoadTeachers;
    }
    /// <summary>
    /// this function load the directory of the xml files and for each file, invoke his loading function.
    /// </summary>
    /// <param name="path">the location of the XML files</param>
    public void LoadDirectory(string path)
    {
        //reset the DB
        ClearDB();
        var directoryInfo = new DirectoryInfo(path);
        var files = directoryInfo.GetFiles();
        foreach (var file in files)
        {
            Action<string> action;
            if (fileNameToAction.TryGetValue(file.Name, out action))
            {
                action?.Invoke(file.FullName);
                Console.WriteLine("{0} Loading....", file.FullName);
            }
        }
    }
    /// <summary>
    /// this function reset the DB
    /// </summary>
    public void ClearDB()
    {
        // var conn = new SqlConnection("data source=SQL-SERVER; initial catalog=!ESTY&TAMAR; integrated
        security=True");
        var conn = new SqlConnection("data source=DESKTOP-7A0S24C; initial catalog=!ESTY&TAMAR; integrated
        security=True");
        SqlCommand cmd;
        conn.Open();

        cmd = new SqlCommand("sp_MSforeachtable 'DELETE FROM ?'", conn);
        cmd.ExecuteNonQuery();
        cmd = new SqlCommand("sp_MSforeachtable 'truncate table ?'", conn);
        cmd.ExecuteNonQuery();
    }
}
```

```

        conn.Close();
    }
    /// <summary>
    /// classes loading
    /// </summary>
    /// <param name="path">xml file location</param>
    public void LoadClasses(string path)
    {
        XmlClasses.root root = GetXmlData<XmlClasses.root>(path);
        try
        {
            using (Entities db = new Entities())
            {
                foreach (XmlClasses.rootData rootData in root.data)
                {
                    db.Classes.Add(_CastDTO.DTOToClass(new ClassDTO()
                    {
                        Layer = rootData.layer,
                        Name = rootData.name,
                        Num = rootData.num,
                        Number = rootData.class_number,
                        SchoolType = rootData.schooltype
                    }));
                }
                db.SaveChanges();
            }
        }
        catch (Exception e)
        {
            LogManager.LogException(e);
        }
    }
    /// <summary>
    /// groups loading
    /// </summary>
    /// <param name="path">xml file location</param>
    public void LoadGroups(string path)
    {
        XmlGroups.root root = GetXmlData<XmlGroups.root>(path);
        try
        {
            using (Entities db = new Entities())
            {
                foreach (XmlGroups.rootGroup rootData in root.group)
                {
                    db.Groups.Add(_CastDTO.DTOToGroup(new GroupDTO()
                    {
                        Num = rootData.num,
                        Teacher = rootData.tea,
                        Subject = rootData.subj,
                        Room = rootData.sroom,
                        Hours = rootData.no_of_hours,
                        CalculateHours = rootData.calculate_hours,
                        SchoolType = rootData.schooltype,
                        Reforma = rootData.reforma,
                        PayAbsence = rootData.goremMeshalem,
                        HourType = rootData.sug,
                        SubHourType = rootData.sub_sug
                    }));
                }
                db.SaveChanges();
            }
        }
    }

```

```

    }
}
catch (Exception e)
{
    LogManager.LogException(e);
}
}
/// <summary>
/// rooms loading
/// </summary>
/// <param name="path">xml file location</param>
public void LoadRooms(string path)
{
    XmlRooms.root root = GetXmlData<XmlRooms.root>(path);

    try
    {
        using (Entities db = new Entities())
        {
            foreach (XmlRooms.rootData rootData in root.data)
            {
                var roomDataNum = rootData.num.ToString();
                db.Rooms.Add(_CastDTO.DTOTORoom(new RoomDTO()
                {
                    ClassId = db.Classes.Where(c => c.Name == rootData.name).FirstOrDefault()?.Id,
                    Floor = int.Parse(roomDataNum[2].ToString()),
                    Number = int.Parse(roomDataNum[2].ToString()) * 100 + int.Parse(roomDataNum[3].ToString()),
                    UseFor = rootData.name
                }));
            }
            db.SaveChanges();
        }
    }
    catch (Exception e)
    {
        LogManager.LogException(e);
    }
}
/// <summary>
/// schedule loading
/// </summary>
/// <param name="path">xml file location</param>
public void LoadSchedule(string path)
{
    XmlSchedule.root root = GetXmlData<XmlSchedule.root>(path);

    try
    {
        using (Entities db = new Entities())
        {
            foreach (XmlSchedule.rootGroup rootData in root.group)
            {
                //TODO
                foreach (XmlSchedule.rootGroupSchedule rootGroupSchedule in rootData.schedule)
                {
                    db.Schedules.Add(_CastDTO.DTOTOSchedule(new ScheduleDTO()
                    {
                        Day = rootGroupSchedule.day,
                        Hour = rootGroupSchedule.hour,
                        WeekDay = rootGroupSchedule.weekDay,
                        Room = rootGroupSchedule.weekDay
                    }));
                }
            }
        }
    }
}

```

```

        }
        db.SaveChanges();
    }
}
catch (Exception e)
{
    LogManager.LogException(e);
}
}

/// <summary>
///subjects loading
/// </summary>
/// <param name="path">xml file location</param>
public void LoadSubjects(string path)
{
    XmlSubjects.root root = GetXmlData<XmlSubjects.root>(path);

    try
    {
        using (Entities db = new Entities())
        {
            foreach (XmlSubjects.rootData rootData in root.data)
            {
                db.Subjects.Add(_CastDTO.DTOTOSubject(new SubjectDTO()
                {
                    Id = rootData.num,
                    Name = rootData.name
                }));
            }
            db.SaveChanges();
        }
    }
    catch (Exception e)
    {
        LogManager.LogException(e);
    }
}

/// <summary>
///teacher loading
/// </summary>
/// <param name="path">xml file location</param>
public void LoadTeachers(string path)
{
    XmlTeachers.root root = GetXmlData<XmlTeachers.root>(path);

    try
    {
        using (Entities db = new Entities())
        {
            foreach (XmlTeachers.rootData rootData in root.data)
            {
                db.Teachers.Add(_CastDTO.DTOTOTeacher(new TeacherDTO()
                {
                    Num = rootData.num,
                    Name = rootData.name,
                }));
            }
            db.SaveChanges();
        }
    }
    catch (Exception e)
    {

```

```

        LogManager.LogException(e);
    }

}
/// <summary>
/// read xml file
/// and convert the text to object
/// </summary>
/// <typeparam name="T"></typeparam>
/// <param name="path"></param>
/// <returns>root xml object</returns>
private static T GetXmlData<T>(string path)
{
    XmlSerializer serializer = new XmlSerializer(typeof(T));
    StringReader stringReader = new StringReader(File.ReadAllText(path, Encoding.GetEncoding(1255)));
    T root = (T)serializer.Deserialize(stringReader);
    return root;
}
}

```

מנגנון טיפול בשגיאות

```
/// <summary>
/// this class take care of exceptions
/// </summary>
public class LogManager
{
    /// <summary>
    /// write logs
    /// </summary>
    /// <param name="e"></param>
    public static void LogException(Exception e)
    {
        // File.AppendAllText("C:\\workspace\\logs.txt", GetInnerExceptionMessage(e)); // our file in bin/debug
        folder
        using (EventLog eventLog = new EventLog())
        {
            eventLog.Source = "SchoolSchedule";
            eventLog.WriteEntry(GetInnerExceptionMessage(e), EventLogEntryType.Error);
        } /**/
    }
    /// <summary>
    /// find all the inner exception messages
    /// </summary>
    /// <param name="e"></param>
    /// <returns>all the inner exception messages</returns>
    public static string GetInnerExceptionMessage(Exception e)
    {
        if (e.InnerException != null)
        {
            return e.Message + "\n" + GetInnerExceptionMessage(e.InnerException);
        }
        return e.Message + "\n" + e.StackTrace + "\n*****\n\n";
    }
}
```

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://stackoverflow.com>
<https://angular.io>
<https://github.com>
<https://www.w3.org>
<https://scotch.io>
<https://codepen.io>
<https://en.wikipedia.org>
<https://confluence.atlassian.com>
<https://nodejs.org>
<https://update.angular.io>
<https://getbootstrap.com>
<https://ng-bootstrap.github.io>
<https://www.freecodecamp.org>
<https://love2dev.com>
<https://www.w3schools.com>
<https://www.regextester.com>
<https://regex101.com>
<https://www.techiediaries.com>
<https://www.flaticon.com>
<https://www.hamichlol.org.il>
<http://www.tzurel.co.il/tag>
<https://www.geektime.co.il>
<https://support.office.com>
<https://www.maketecheasier.com>
<https://blogs.msdn.microsoft.com>
<https://picpick.app/en/>