בס"ד

**מפרט ארכיטקטוני:**

הארכיטקטורה של הפתרון המוצע בפורמט של level Down-Top :

הפרויקט מחולק לשני חלקים- לצד השרת וצד הלקוח:

1.צד שרת שנכתב בשפת #C בטכנולוגית WebApi .

2.צד לקוח שנכתב בשפת js, Html .

**תיאור צד שרת(server):**

צד השרת מחולק כמקובל לשכבות:

שכבת DAL.

שכבת DTO .

שכבת BLL .

החלוקה לשכבות נועדה להפריד באופן מוחלט בין הלוגיקה של הפרויקט לבין הנתונים עצמם.

הפרדה זו מאפשרת לבצע שינויים בכל אחת מהשכבות בלי תלות ובלי זעזועים בשכבות האחרות.

פירוט:

**: DAL**שכבת ה DAL היא השכבה דרכה ניגשים לנתונים היושבים ב. Data Base היא מכילה מחלקות המייצגות את בסיס הנתונים וכן פונקציות נחוצות לתפעולו. פעולות ההתקשרות עם בסיס הנתונים נעשו בטכנולוגית Entityframwork, טכנולוגיה זו היא המקובלת לטיפול בבסיס הנתונים כאשר היא מנתחת את בסיס הנתונים, בונה מחלקות לייצוג הטבלאות ומספקת רשימות מלאות בנתוני בסיס הנתונים.

**BLL:** שכבה שבה כתובה כל הלוגיקה של הפרויקט.

**DTO**: שכבה זו מכילה מחלקות המתארות את הנתונים ובמבנה זה מעבירים את הנתונים בין השכבות. מטרת שכבה זו היא למנוע תלות של שכבת ה bll במבנה בסיס הנתונים.

שכבת Dal מכילה פונקציות המרה מטיפוס הנתונים של בסיס הנתונים לטיפוס הנתונים של ה dto ולהיפך, וכך מיוצגים הנתונים בכל הפרויקט.

מודל זה אמנם גורם טרחה טכנית לא מעטה בכתיבת הקוד ובתכנון הפונקציות אבל מספק קוד נקי, קל להבנה ונוח לשינויים ולשדרוגים.

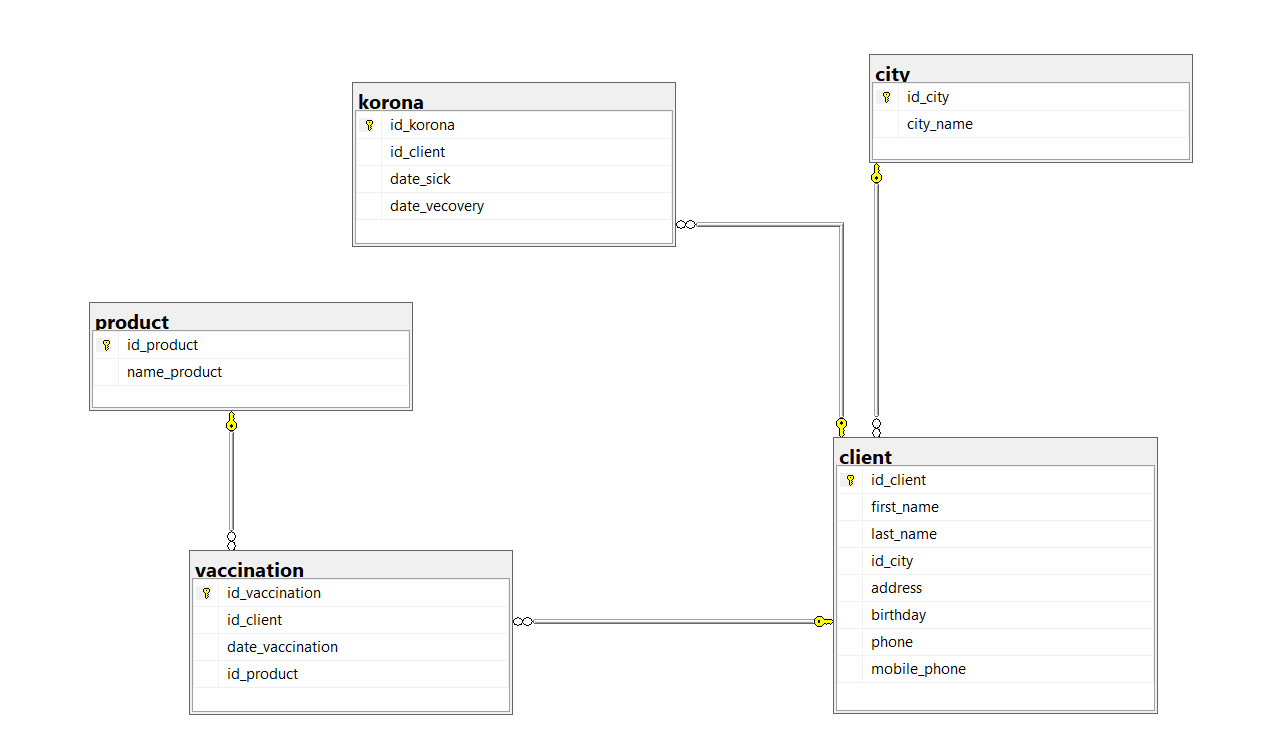
תמונה שמכילה תרשים

התיאור נוצר באופן אוטומטי

Dto

Bll

**תיאור קישרי גומלין:**

****