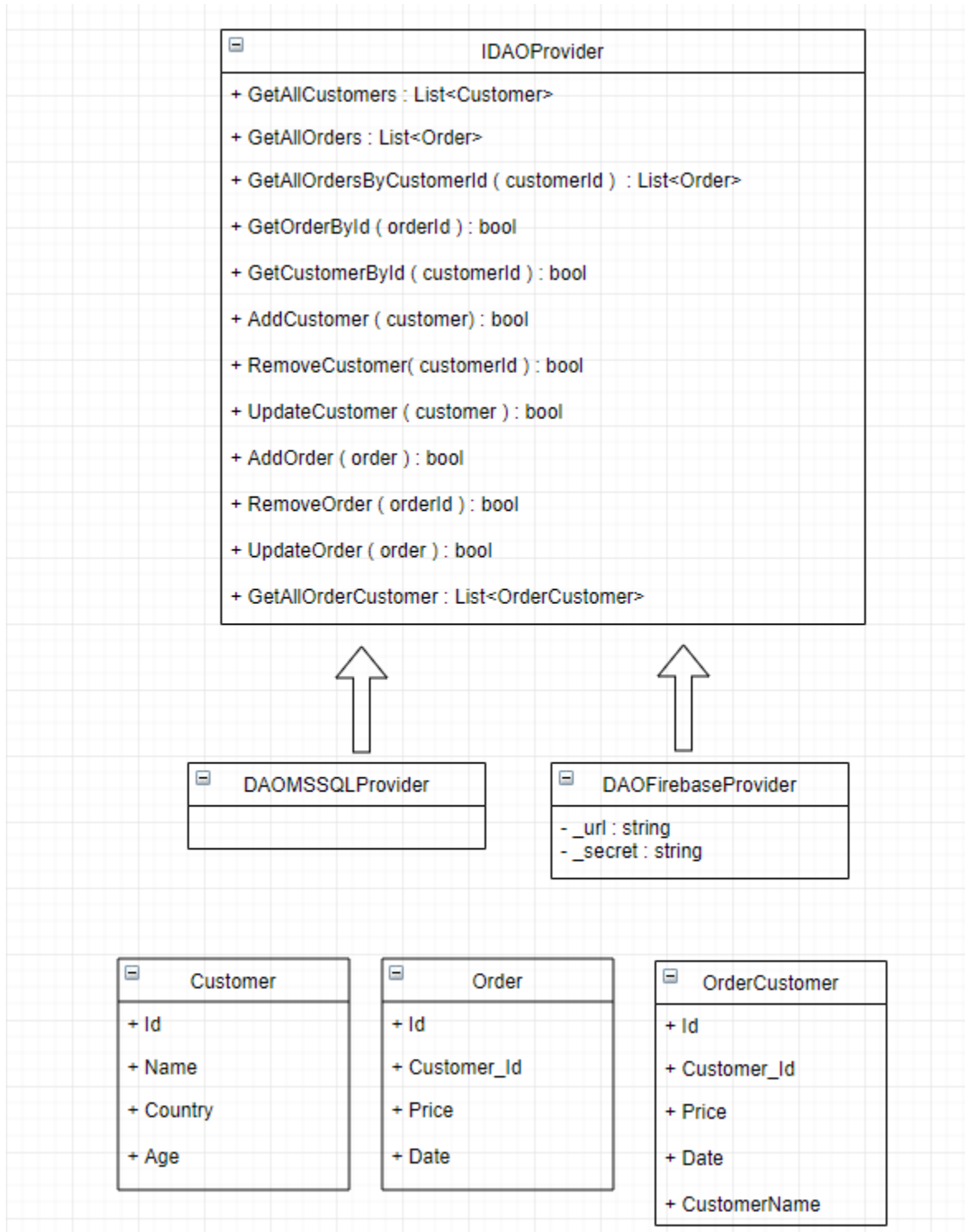


תרגיל סיכום Entity FW + Firebase



- מה מטרת הפרוייקט?

- לתרגל entity framework ושאליות LINQ
- לתרגל NOSQL ושימוש ב Firebase
- לחוש בהבדל בין עבודה עם SQL לבין עבודה עם NOSQL
- **אין להשתמש בשאליות SQL בתרגיל** – אלא רק ב EntityFramework + LINQ עבור ה MSSQL וב- FireSharp עבור ה- Firebase

- בדיאגרמה נתונה תצורת טבלת לקוחות והזמנות.

1. צור את טבלת **Customer** ואת טבלת **Order** ב MSSQL
 - a. שים לב ש ID בטבלת לקוח הוא מפתח ראשי בטבלת לקוחות
 - b. לקוח ID מהווה מפתח משני בטבלת הזמנות
 - c. הזמנה ID בטבלת הזמנות היא מפתח ראשי
2. צור את טבלת **Customer** ואת טבלת **Order** ב Firestore
3. צור פרוייקט ב VS2017
 - a. צור מחלקה בשם **Customer** (POCO) ושים בתוכה את כל השדות הקיימים בטבלה
 - b. צור מחלקה בשם **Order** (POCO) ושים בתוכה את כל השדות הקיימים בטבלה
 - c. צור מחלקה בשם **OrderCustomer** (POCO) . מחלקה זו תשמש רק להחזרת תשובה ממתודה **GetAllOrderCustomer**
 - d. צור את האינטרפייס **IDAOProvider**
 - e. צור שתי מחלקות אשר ממשות את האינטרפייס **IDAOProvider**:
DAOFireBaseProvider ו- **DAOMSSQLProvider**
הרעיון הוא שמחלקת DAOFireBase תממש את כל המתודות עבור בסיס הנתונים ב Firebase ומחלקת DAOMSSQL תממש את כל המתודות עבור בסיס הנתונים MSSQL. כמו כן יש להתמש רק במחלקה שקיבלנו מה- EntityFramework כדי לתקשר עם בסיס הנתונים (ולא להשתמש בשאליות SQL)
 - f. עבור ה- **DAOFireBaseProvider**: ה- url וה- _secret יוגדרו בקובץ App.config וייקראו בבנאי. ולאחר מכן יש להשתמש בהם כדי להגדיר את החיבור ל- Firebase
 - g. יש להגדיר Main אשר ממנו תבצע קריאה לכל אחת מהמתודות בשני ה- DAO, כדי לוודא את תיקנות ההרצה
 - h. במקום לממש ToString , בכדי להדפיס את האובייקטים, השתמש ב JSON CONVERT (Newtonsoft)

המשך בדף הבא

הסבר הפונקציות:

+ GetAllCustomers : List<Customer>

החזר את רשימת כל הלקוחות מטבלת לקוחות

+ GetAllOrders : List<Order>

החזר את רשימת כל ההזמנות מטבלת הזמנות

+ GetAllOrdersByCustomerId (customerId) : List<Order>

החזר רשימת הזמנות עבור לקוח מסוים (עפ"י ה-ID שלו)

+ GetOrderById (orderId) : bool

מצא הזמנה ספציפית (עפ"י ה-ID של ההזמנה)

+ GetCustomerById (customerId) : bool

מצא לקוח ספציפי (עפ"י ה-ID של הלקוח)

+ AddCustomer (customer) : bool

הוסף לקוח לטבלה

שים לב במימוש ב-Firebase: יש למנוע הוספה של לקוח אשר קיים לקוח עם אותו ID

+ RemoveCustomer(customerId) : bool

הסר לקוח מהטבלה (עפ"י ה-ID של הלקוח)

שים לב במימוש ב-Firebase: יש למנוע מחיקה של לקוח אשר קיימת הזמנה אחת (או יותר) עבורו

+ UpdateCustomer (customer) : bool

עדכן לקוח

אין לעדכן ID של לקוח

+ AddOrder (order) : bool

הוסף הזמנה לטבלה

שים לב במימוש ב-Firebase: לא תהיה אפשרות להוסיף הזמנה עם ID של לקוח שאינו קיים

+ RemoveOrder (orderId) : bool

הסר הזמנה ספציפית (עפ"י ה-ID של ההזמנה)

+ UpdateOrder (order) : bool

עדכן הזמנה

שים לב במימוש ב-Firebase: לא תהיה אפשרות לעדכן בהזמנה את הלקוח ID ל-ID של לקוח שאינו קיים בטבלת לקוחות

+ GetAllOrderCustomer : List<OrderCustomer>

החזר רשימה של טיפוס OrderCustomer המכילה את רשימת כל ההזמנות בטבלה.

שים לב שבכל רשומה יופיע שמו של הלקוח אשר ביצע את ההזמנה.

יש לעשות JOIN כדי לקבל את רשימת התשובות...

המתודות המחזירות בוליאני: יחזירו אמת אם הפעולה הצליחה ושקר אם נכשלה

בהצלחה