דו"ח הפרוייקט של אריאל יוגב 207917501 וגיא בן משה 318363397:

1. המערכת היא חברה לניהול וארגון דייטים. כלומר, היא מורכבת מדייטים, לקוחות אשר מעוניינים לצאת לפגישה (הם יצאו לפגישה רק עם המגדר בו הם מעוניינים), שדכנים (עובדי החברה), מקומות אפשריים לדייט איתם החברה משתפת פעולה ומקבלת תשלום עבור שיתוף פעולה זה.  
בנוסף, עבור כל דייט ישלם הלקוח תשלום בסיס שקובע מנהל המערכת.

2. **הסבר על כל מבנה: (לא כולל: fileHelper, CompanyFile, General, List, Def, Macros)**

***\*הערה: בכל מבנה לפי הצורך קיימות פונצקיות init, הדפסה, שחרור, קריאה מקובץ וכתיבה לקובץ, וכן בדיקות תקינות רלוונטיות למבנה.***

**Company** ( 2.1 :מבנה זה הוא הישות הראשית של המערכת.

* Meet\*\* meetsArr - מערך הפגישות. **(מערך של מצביעים בגודל לא ידוע)**
* Char\* name - שם החברה. **(מערך תווים בגודל לא ידוע)**
* Place\* placesArr - מקומות אפשריים לדייטים. **(מערך של מקומות בגודל לא ידוע)**
* Int meetsCount, placesCount - כמות הפגישות המעודכנת וכמות המקומות שהחברה משתפת איתם פעולה.
* esortType type - מיון הפגישות על פי שלושה קריטריונים : תאריך/מיקום/שדכן הפגישה(לקסיקוגרפי)
* FinancialManagement pFM – הנהלת החשבונות של החברה. **(הכלה של מבנה ללא הצבעה)**

מבנה זה הוא הישות הראשית של המערכת. הוא כולל בתוכו את כל הפגישות, את כל המקומות שבהם ניתן להיפגש, מספר פגישות ומספר מקומות, אופן מיון הפגישות ואת הנהלת החשבונות.

פונקציות ראשיות:

* addMeet - הוספת פגישה.
* addPlace – הוספת מקום.
* sortMeet – מיון פגישה לפי שלושה קטגוריות לבחירה (לפי מקום(כתובת), לפי שם השדכן ולפי תאריך).
* findMeet – חיפוש פגישה לפי האופן בו מערך הפגישות ממוין, אם הוא ממוין.

2.2 ) **PeopleManager** : ישות שאחראית על האנשים בחברה - הן לקוחות והן שדכנים

* LIST Clients - רשימת לקוחות. **(רשימה)**
* LIST MatchMaker - רשימת שדכנים. **(רשימה)**
* Int countClient, countMatchMaker - כמות עדכנית של לקוחות ושדכנים בחברה.

מבנה זה הוא הישות שאחראית על האנשים בחברה, הן לקוחות והן שדכנים.   
הוא כולל בתוכו רשימה של לקוחות ורשימה של שדכנים(עובדי החברה), מספר לקוחות ומספר שדכנים.

לכל שדכן שנוסף יש מספר סידורי ייחודי, ולכל לקוח תעודת זהות ייחודית.

פונקציות ראשיות:

* addClient – הוספת לקוח, **הלקוחות מתווספים לרשימה לפי סדר אלפבית**.
* addMatchMaker – הוספת שדכן, **השדכנים מתווספים לרשימה לפי סדר אלפבית**.
* atLeastOneValidMatchByGender – בדיקה האם יש לפחות שני לקוחות אפשריים לשידוך. אחרת, לא ניתן להוסיף פגישה.

2.3 ) **Meet** : מבנה המכיל את כל פרטי הדייט(פגישה)

* Place\* thePlace - מיקום הדייט. **(מצביע למקום)**
* Char clientId1[ID\_LEN +1]– תעודות זהות של לקוח אחד בפגישה. **(משתנה char בגודל ידוע מראש)**
* Char clientId2[ID\_LEN +1]– תעודות זהות של לקוח שני בפגישה. **(משתנה char בגודל ידוע מראש)**
* Int matchMakerSN - מספר סידורי של העובד בחברה.
* Date theDate – תאריך קיום הפגישה.

מבנה זה מתאר פגישה. הוא כולל בתוכו את כתובת מקום הפגישה, תעודות זהות של שני הלקוחות המשתתפים בפגישה, המספר הסידורי של השדכן שאחראי על הפגישה ותאריך הפגישה.

פונקציות ראשיות:

* setClientToMeet – השמת תעודת הזהות של הלקוח בפגישה.
* setMM\_ToMeet – השמת המספר הסידורי של השדכן של הפגישה.
* pickTwoClients – בחירת שני לקוחות לפגישה, **(שני לקוחות שונים שהמגדר בו הם מעוניינים תואם)**.
* compareMeetByDate – השוואת שתי פגישות לפי התאריך.
* compareMeetByPlace – השוואת שתי פגישות לפי כתובת מיקום הפגישה.
* compareMeetByMatchMaker – השוואת שתי פגישות לפי מסםפר סידורי של השדכן.

2.4 ) **Place** : מבנה המכיל מקום ונתוניו

* Char\* placename – שם המקום. **(מערך תווים בגודל לא ידוע)**
* eType placeType - סוג המקום: מסעדה/בית קפה/מסיבה.
* Address pAddress - כתובת המקום : עיר, רחוב, מספר.

מבנה זה מתאר מקום. הכולל בתוכו את שם המקום, סוג המקום, ואת כתובת המקום כמבנה. שם המקום יישמר כך: לדוגמא – “Coffe Place” (הוספת גרשיים ואות גדולה בתחילת מילה).

פונקציות ראשיות:

* setAddress – הגדרת כתובת המקום ובדיקת תקינות וייחודיות.
* comparePlacesByAddress – השוואת שני מקומות לפי כתובת.

2.5 ) **MatchMaker** : שדכן שאחראי על יצירת הדייט

* char\* name – שם השדכן. (**מערך תווים בגודל לא ידוע)**
* Int sn(sequence number) - מספר סידורי של העובד.
* int matchCounter – כמות הדייטים שהוציא לפועל.
* int yearsInCompany - מספר שנים שעובד השדכן בחברה.
* int salary – משכורת פר דייט שהוציא לפועל, נקבע לפי מספר השנים בחברה.

מבנה המתאר שדכן בחברה. הכולל בתוכו את שם השדכן, המספר הסידורי של השדכן בחברה, כמות השידוכים שביצע השדכן, כמה שנים נמצא בחברה, תשלום פר פגישה ששידך(salary) ובנוסף, שדכן יישמר לקובץ בצורה דחוסה.

שם השדכן יישמר כך: לדוגמא – Yossi Cohen (אות גדולה בתחילת מילה).

כמות השידוכים המקסימליים שיכול לבצע היא: 60

כמות השנים המקסימלית של עובד בחברה היא: 7

משכורת מקסימלית של עובד היא: 120

המספר הסידורי המקסימלי של עובד בחברה הוא: 999

פונקציות ראשיות:

* getYearsInCompany – השמה של שנות עבודה בחברה.
* initMM\_SN – השמה של מספר סידורי ייחודי.
* findMMBySn – מציאת שדכן לפי מספר סידורי.
* matchMakeWithMaxMeets – פונקציה שמחזירה את השדכן עם מספר השידוכים הגבוהה ביותר. (לשימוש הנהלת החשבונות)
* setSalary – השמה של המשכורת פר פגישה.
* getMatchMakerName – השמה של שם השדכן.
* compareMatchMakerByName – השוואת שני שדכנים לפי השם שלהם.
* write\_Compressed\_MatchMakerToBinary – כתיבה של שדכן לקובץ בינארי באופן דחוס.
* read\_Compressed\_MatchMakerFromBinary – קריאה של שדכן בקובץ בינארי באופן דחוס.
* כאמור קיימות פונקציות כתיבה וקריאה מקובץ טקסט.

2.6 ) **Client** : לקוח בחברה.

* char\* name - שם הלקוח. (**מערך תווים בגודל לא ידוע)**
* char id[ID\_LEN + 1] - תעודת הזהות של הלקוח. **(משתנה char בגודל ידוע מראש)**
* eGender gender - סוגי מגדר : נקבה/זכר.
* eGender interestingIn – המגדר אליו נמשך.

מבנה המתאר לקוח בחברה. הוא כולל בתוכו את שם הלקוח, תעודת זהות ייחודית, מגדר הלקוח, והמגדר בו הוא מעוניין.

שם הלקוח יישמר כך: לדוגמא - Yossi Cohen (אות גדולה בתחילת כל מילה).

מספר תעודת הזהות הוא בעל 9 ספרות.

פונקציות ראשיות:

* getGender – השמה של מגדר.
* getClientName – השמה של שם הלקוח.
* initClientID – השמה של תעודת זהות תקינה וייחודית ללקוח.
* compareClientByName – השוואת שני לקוחות לפי השם.
* isClientsSuitable – בדיקה האם שני לקוחות מתאימים לשידוך לפי מגדר והמגדר בו הם מתעניינים.
* findClientById – חיפוש לקוח בעזרת תעודת זהות.
* calcMatchPrecent – חישוב אחוזי התאמה של שני לקוחות באופן נומרולוגי\* (לפי שם פרטי).

\*האלגוריתם לחישוב אחוזי התאמה הוא מבוסס נומרולוגיה עם תוספת אישית שלנו.

2.7 ) **FinancialManagement**: הנהלת החשבונות של החברה.

* Int outcome, income -הכנסות והוצאות של החברה.
* Int paymentPerMeet - תשלום בסיס עבור כל פגישה. נקבע על ידי מנהל המערכת.

מבנה המתאר את הנהלת החשבונות של החברה, כלומר מנהל את כל מה שקשור לכספים.  
הוא כולל בתוכו את הוצאות החברה, הכנסות החברה ותשלום של לקוח עבור פגישה שנקבעה לו.

תשלום לקוח הוא מינימום: 20 ומקסימום: 80.

בנוסף, במבנה מוגדרים התשלומים שמתקבלים מהמקומות בהם נקבעות פגישות לפי סוג המקום: ממסעדה החברה תקבל תשלום על סך 100, ממסיבה על סך 70, ומבית קפה על סך 20.

פונקציות ראשיות:

* calcCompanyProfit – חישוב רווח החברה.
* initPayment – השמה של התשלום של כל לקוח פר פגישה.
* getRaise – לתת העלאה במשכורת לעובד עם מספר השידוכים הגבוהה ביותר(מספר הפגישות שהוציא לפועל הגדול ביותר).

2.8) **Address** : כתובת.

* Char city[MAX\_STR\_LEN] – עיר
* Char street[MAX\_STR\_LEN] - רחוב
* Int number – מספר

מבנה המתאר כתובת. במערכת שלנו מבנה זה משמש ככתובת של מקום הפגישה, הכולל בתוכו את שם העיר, שם הרחוב והמספר.

שם של עיר או רחוב יישמר כך: לדוגמא- Tel\_Aviv (הוספה של התו '\_' בין מילה למילה ואות גדולה בתחילת כל מילה).

פונקציות ראשיות:

* getCorrectAddress – השמה של כתובת תקינה.
* compareAddresses – השוואת שתי כתובות.
* getCityNStreet – השמה של שם העיר או שם הרחוב.

2.9) **Date** : תאריך.

* Int day – יום.
* Int month – חודש.
* Int year – שנה.

מבנה המתאר תאריך. במערכת שלנו מבנה זה משמש כתאריך הפגישה, הכולל בתוכו את היום, חודש, ושנה.

פונקציות ראשיות:

* getCorrectDate – השמה של תאריך תקין.
* compareDates– השוואת שני תאריכים.

***\*\* void generalArrayFunction(void\*, int, int, void(\*func)(void\*))***

***פונקציה גנרית זו נמצאת במחלקת ה General ומשתמשים בה לצורך הפעולות הבאות:***

***freePlacesArr , printPlaceArr , freeMeetsArr .***

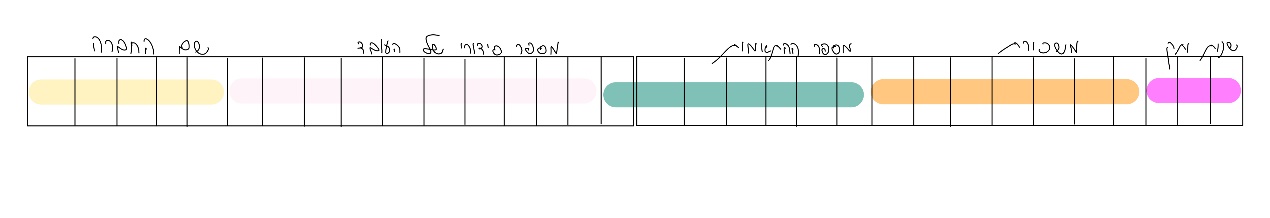
**3 ) נדחוס את מבנה Matchmaker**  
נגדיר מערך בגודל - BYTES 4

את אורך השם נדחוס ראשון לחמישה ביטים – נצא מנקודת הנחה שהשם לא יהיה ארוך יותר מ32 סיביות.

את המספר הסידורי של העובד נדחוס לעשרה ביטים – נצא מנקודת הנחה שהוא לא יהיה גדול מ999.

את מספר ההתאמות נדחוס לשבעה ביטים. – נגדיר שהמספר המקסימלי של התאמות עד לאיפוס הספירה הוא 60.

את המשכורת פר דייט נדחוס לשבעה ביטים – נגדיר שהמשכורת פר דייט לא יכולה להיות גבוהה מ120.

את מספר השנים שעובד בחברה נדחוס לשלושה ביטים – נגדיר שהמספר המקסימלי של שנים בהן אדם עובד בחברה הוא שבע שנים ואז יצטרך לחדש חוזה.

Byte 3

Byte 2

Byte 1

Byte 0

**4) הפעולות אותן המשתמש יכול לבצע במערכת :**

1) הוספת לקוח

2) הוספת שדכן

3) הוספת מקום

4) הוספת פגישה

5) הדפסת נתונים: פותח תפריט חדש של:

- הדפסת נתוני החברה: פגישות ומקומות

- הדפסת כל הלקוחות והשדכנים.

- הדפסת נתוני הנהלת החשבונות: תשלום של כל לקוח פר פגישה, הוצאות, הכנסות, רווח.

6) מיון מערך הפגישות. פותח תפריט חדש של:  
  
 - מיון לפי תאריך

- מיון לפי מספר סידורי של השדכן

- מיון לפי כתובת מיקום הפגישה

7) חיפוש פגישה באופן בה המערך ממוין.

8) הענקת העלאה בשכר(פר פגישה) לשדכן עם הכי הרבה שידוכים עד כה.

9) עדכון תשלום הבסיס של כל לקוח פר פגישה.

10) חישוב אחוזי התאמה בין שני לקוחות באופן נומרולוגי (לפי שם פרטי).

**5)**  ***\*הערה: בכל מבנה לפי הצורך קיימות פונצקיות init, הדפסה, שחרור, קריאה מקובץ וכתיבה לקובץ, וכן בדיקות תקינות רלוונטיות למבנה.***



**6 ) חלוקת תפקידים :**

את הפרויקט כולו עשינו יחד, בחשיבה משותפת.