



**שם הקורס: ניתוח ותכנון מערכות מידע א'**

**נושא הפרויקט: מערכת מידע לתכנון פיננסי**

**מרצה הקורס: ד"ר גבי פינטו**

**מגישים:**

**נועם אמרון 305568818**

**חן שליו 203205984**

**מיכאל יירמולייב 311461263**

תוכן

[תקציר 3](#_Toc3190082)

[מבוא 4](#_Toc3190083)

[מבנה העבודה: 4](#_Toc3190084)

[ייזום 5](#_Toc3190085)

[המצב הקיים: 5](#_Toc3190086)

[השיפור המוצע ע"י מ"מ 5](#_Toc3190087)

[המצב המשופר: 6](#_Toc3190088)

[אפיון 7](#_Toc3190089)

[מטרת מערכת המידע: 7](#_Toc3190090)

[גבולות המערכת: 7](#_Toc3190091)

[אילוצים: 8](#_Toc3190092)

[תהליכים: 8](#_Toc3190093)

[**תיאור לוגיקת התהליך:** 10](#_Toc3190094)

[ניתוח מערכת 12](#_Toc3190095)

[תרשימי DFD 12](#_Toc3190096)

[עיצוב מערכת 14](#_Toc3190097)

[טרנזקציות 14](#_Toc3190098)

[עץ תפריטים 16](#_Toc3190099)

[תרשים ERD 17](#_Toc3190100)

[תכנות – בדיקות – הטמעה 18](#_Toc3190101)

[תרשים מערכת: 18](#_Toc3190102)

[יישום ובדיקות 19](#_Toc3190103)

[מערכת המידע 21](#_Toc3190104)

[סיכום הפרויקט 25](#_Toc3190105)

[ביבליוגרפיה 26](#_Toc3190106)

[נספחים 27](#_Toc3190107)

[מילון מרכיבי DFD: 27](#_Toc3190108)

[מילון מאגרי נתונים 31](#_Toc3190109)

[מילון ישויות 33](#_Toc3190110)

[מילון זרמי נתונים 34](#_Toc3190111)

[נספח מס' 3: מערכת המידע 35](#_Toc3190112)

# תקציר

התחלנו בחיפוש רעיון אחר מערכת מידע אשר תסייע למשתמשים בה לשפר תהליכים בחייהם ולהפוך אותם ליעילים וקלים יותר עבורם.   
בתהליך החשיבה שעברנו יחד כל חברי הצוות, הגענו למסקנה שיש צורך במערכת שכר אשר לא רק מחשבת את סה"כ הכנסה חודשית אלא גם מחשבת את ההוצאות, החיסכון החודשי ואת תזרים מזומנים לפעילות יומיומית, תוך כדי התחשבות באילוצים שונים וביניהם: אילוצי תקציב, התחייבויות, הוצאות קבועות ועוד. כמו כן, היה חשוב לנו שהמערכת תיתן יעוץ והכוונה למשתמשים, ותיתן הערכה להכנסות אל מול הוצאות.

בכדי לדעת כיצד לאפיין את המערכת, ביצענו Benchmarking – ביקרנו במספר גדול של אפליקציות דומות בין אם באינטרנט ובין אם בחנויות האפליקציות וראינו מה הם מציעים ומה חסר לנו ובכך כיוונו למטרה מסוימת.

מאחר ומערכת מסוג זה דורשת שיתוף פעולה מצד המשתמשים, בחרנו להשתמש בשיטת Agile Scrum. שיטה זו אפשרה לנו לפתח את המערכת תוך קבלת משוב והתיקון הנדרש במהלך הפיתוח עד לכדי קבלת מערכת שתהיה ברורה למשתמש באופן מוחלט.

לקראת סוף תהליך הפיתוח, האצנו את השימוש במערכת ע"י משתמשים שונים. שימוש זה היווה לנו כפיילוט למערכת והמחיש את תפעול המערכת במצב של ריבוי משתמשים ונתונים.

שמחנו לגלות שיתוף פעולה מצד חברינו שבא לידיי ביטוי ברישום לאתר ושיתוף המידע הפרטי שלהם במערכת המידע וכן הפצתה לחברים נוספים.   
כמו כן, נהיה מעוניינים להרחיב את השימוש במערכת ולהוסיף לה פיצ'רים נוספים אשר יאפשרו שימוש נוח יותר למשתמש ותפעול מהיר של המערכת, תוך מתן מענה לדרישות הלקוחות (המשתמשים).

# מבוא

במסגרת הקורס, ניתוח ותכנון מערכות מידע א', נתבקשנו לבנות מערכת מידע שתסייע לבני אדם. בחרנו לבנות מערכת מידע שתסייע לאנשים לתכנן את הכנסותיהם וההוצאות ע"פ תקציב ו/או אופי המשתמש (אילוצים שונים\ התחייבויות) בקלות ובמהירות תוך עמידה במדדי ה-TCQ.

כיום, כאשר משתמש מעוניין לקנות מוצר \ לשכור שירות, עליו לדעת מהי הכנסתו החודשית ומהם ההתחייבויות הכספיות שלהם התחייב.

מטרת מערכת המידע שלנו הינה לא רק לאגד לאדם את התחייבויות חודשית והכנסותיו, אלא גם לתת הכוונה לפרק זמן שעליו לחסוך על מנת לרכוש מוצר \ שירותים שבהם המשתמש חפץ.

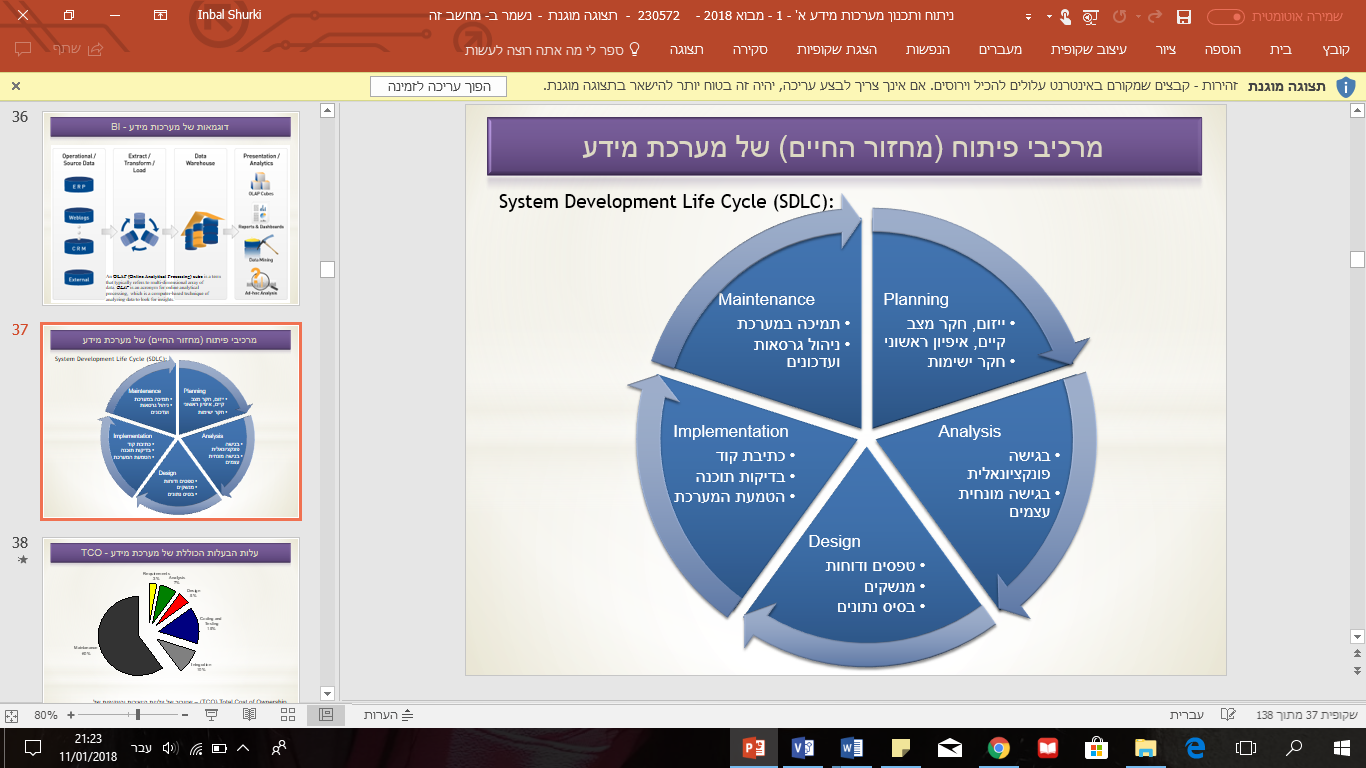
המוטיבציה לבחירת התחום ולביצוע הפרויקט הייתה פרקטית ומאוד קרובה לליבנו. אנו כסטודנטים\שכירים מבינים את חשיבות הדבר והייחודיות שבאפליקציה שלנו לעומת האפליקציות בשוק אשר רק מחשבות את השכר החודשי למשתמש ולא מעבר לכך.

הסיבה לכך שסטודנטים לוקח זמן רב לרכוש מוצר נובעת מעצם חוסר הודאות שבתכנון, הן מבחינת התקציב, הן מבחינת אופי המוצר והן מבחינת הזמן הדרוש לחסכון ובהתחשב באילוצים אחרים. מערכת זו מאפשרת למשתמש לבחור באופי החיסכון המתאים באופן פרטני בהתחשב בהכנסות החודשיות אל מול ההתחייבויות החודשיות והוצאות חד פעמיות\ משתנות.

המערכת תפעל באופן שבו כל משתמש יכול להירשם לאתר ולשתף את המידע שיבחר, כאשר כל משתמש יכול לבחור באיזה מידע לשתף ובאיזה לא תוך כדי הגנת פרטיות הלקוח וביטחון מידע חסוי לחלוטין. (יש לקחת בחשבון שהחסרת מידע מהמערכת לא תשקף תמונה הדומה למציאות והחישובים בה יהיו לא מדויקים).

## מבנה העבודה:

ביצענו את הפרויקט על-פי מחזור החיים של מערכת המידע הכולל:

* ייזום הכולל חקר מצב קיים ומצב משופר.
* אפיון.
* ניתוח המערכת בגישה הפונקציונאלית.
* עיצוב המערכת (מנשקים, דוחות, תפריטים, בסיס נתונים).
* יישום (תכנות) ובדיקות.
* הטמעת המערכת ותחזוקה.

# ייזום

## המצב הקיים:

כיום קיימים שירותים רבים שמאפשרים למשתמשים לחשב את שכרם החודשי ולעקוב אחר המשמרות שביצעו החודש, אומנם דבר חשוב שחסר באפליקציות הללו הוא כלי מובנה לעזור לנהל את הפיננסים של המשתמשים מעבר לחישוב השכר הבסיסי החודשי.

מדוע מתן מידע פיננסי מורחב למשתמש זו תכונה כה חשובה למשתמש?

ניתן להדגים זאת לפי שלושה מדדי ביצוע (TCQ):

* **זמן (Time)** – נדרש מאתנו זמן רב בכדי לתאם בין האפליקציות השונות שמספקות לנו מידע שיכול להיות מסופק במקום אחד מרוכז .
* **עלות (Cost)** – אנו נאלצים לשכור את שירותם יקרים של בנקים ושל יועצים פיננסים בכדי לעזור לנו לאזן את חשבון הבנק ומאגר המזומנים שלנו.
* **איכות (Quality)** – מכיוון שהידע הנחוץ לנו בכדי לבצע החלטות פיננסיות נכונות מפוזר במקומות שונים , לרוב ההחלטות שנבצע היו לא נכונות ולא ישקפו את המצב הפיננסי שלנו בצורה האיכותית ביותר.

תמונת מצב פיננסית מדויקת הוא אחד הדברים החשובים ביותר לאדם שכיר שצריך לבצע החלטות על כמה כסף נזיל יש ברשותנו וכמה אנחנו יכולים להרשות לעצמנו לבזבז.   
לפיכך, אנו מציעות מצב משופר.

### השיפור המוצע ע"י מ"מ

מערכת המידע שנקים עתידה לרכז בתוכה את כל המידע הפיננסי הרלוונטי למשתמש הקצה.

מ"מ מחשבת עבור המשתמש את הכנסתו החודשית , בהתחשב לשעות נוספות ,שכר גלובלי ושכר נטו וברוטו. בנוסף האפליקציה מספקת נתונים לגבי התחייבויות קודמות של המשתמש בהתחשב באילוצים שהוגדרו מראש. לסיכום המ"מ מאפשרת למשתמש לנהל במקום אחד מרוכז את ההוצאות וההכנסות שלו , הן קבועות והן משתנות .

בחרנו לפתח את המערכת על פי מתודולוגיית Agile Scrum.

כאשר מפתחים מערכת המתבססת על מתודולוגיה זו, מנהלים פיתוח זריז על מנת להנפיק ללקוח אב-טיפוס של המערכת בשלב מוקדם ובכך להספיק לבצע שינויים נצרכים עוד טרם שלב הבדיקות. מצב זה מותיר ללקוח אפשרות לשנות את דרישותיו לאורך כל הפרויקט. בנוסף, הלקוח רואה תוצרים בשלבים מוקדמים מאוד.

יתרה מזאת. שיטה זו מאפשרת לנו כמפתחים להתאים את המערכת לצורכי הלקוח בזמן אמת ולהפוך אותה לאפקטיבית יותר עבור הלקוח. כמו כן, שיטה זו מאפשרת להימנע מטעויות בשלב מוקדם יחסית של הפיתוח ובכך לחסוך זמן ומשאבים רבים.   
בפועל, בזמן פיתוח המערכת אפשרנו ללקוחות לעשות בה שימוש – להוסיף משמרות והתחייבויות במהלך היום ולקבל פידבק על מצבם הפיננסי בזמן אמת.

בכל שלב בפיתוח ביצענו משוב ושאלנו:

* מה חסר למשתמשים במערכת?
* האם המערכת ברורה למשתמשים?
* האם התפריטים ברורים למשתמשים?
* עם מה "לא נוח" למשתמש?
* אילו תכונות במערכת הם אוהבים/מועילים למשתמש?
* האם המשתמש יהיה מעוניין ביותר/פחות מידע ?

שיטת Agile Scrum מאופיינת בגמישות רבה אשר באה לידיי ביטוי בשינויים המהירים בהתאם לצורכי הלקוח והמשוב ממנו תוך זמן קצר. בכדי לאפשר גמישות זו, בחרנו לעבוד בצוות קטן שבוא כולם לקחו חלק בכתיבת הקוד עבור האפליקציה. עבודה בצוות של 3 מאפשרת לנו להתעדכן במהירות וביעילות אודות השינויים המהירים במערכת. עם זאת, עבודה בשיטה זו הקשתה עלינו לתכנן את הפעילות העתידית בפרויקט כיוון שידענו שצפויים בפנינו שינויים רבים בהתאם למשוב שנקבל.

## המצב המשופר:

כיום, עם השימוש במערכת, החלו משתמשים להשתמש באפליקציה באופן שותף ביום יום.

השימוש במערכת מאפשר למשתמש לנהל את האספקט הפיננסי של חייב ללא כאב ראש ולקבל במקום אחד את כל המידע הרלוונטי בכדי לבצע החלטות פיננסיות חכמות ונכונות.

בנוסף, הפורמט הנוכחי מאפשר קבלת מידע על הוצאות עתידיות שאולי המשתמש הממוצע לא מתחשב בהן כאשר הוא מבצע החלטה פיננסית.

ניתן לראות כי המצב המשופר נמדד בשלושה מדדי ביצוע (TCQ):

* **זמן (Time)** – חוסך זמן חיפוש של מידע פיננסי רלוונטי במקומות שונים.
* **עלות (Cost)** – המ"מ מסייעת בקבלת החלטות פיננסיות ללא צורך בשכירת יועצים ואנשי מקצוע יקרים.
* **איכות (Quality)** – נותנת תמונת מצב מדויקת ואיכותית של המצב הכספי של המשתמש דבר שמאפשר לבצע החלטות נכונות ואיכותיות.

# אפיון

## מטרת מערכת המידע:

מציאת תכנון למשתמש תוך שיפור מדדי ה- TCQ.   
מטרת המערכת הינה להוות כלי יעיל לתכנון תקציב ותזרים המזומנים.

|  |  |  |
| --- | --- | --- |
| **עדיפות** | **מטרה** | **יישום במערכת** |
| 1 | המערכת תאפשר זרימת מידע באופן שוטף, הנגיש למשתמש מכל מקום בעולם (צריך קישור אינטרנטי) | בחירה במערכת אינטרנטית |
| 4 | מערכת ידידותית למשתמש אשר תגרום לשימוש נוח ומהיר של המערכת. | פיתוח ממשק ידידותי |
| 5 | T-מציאת חיסכון בזמן באופן המהיר ביותר | אפשרות לבצע סינון במערכת על פי הקריטריונים הרצויים. דבר המאפשר לקבל תוצאות מתאימות במהירות וביעילות. |
| 2 | C- מציאת תוכנית כלכלית לפי אילוצים שונים: מחויבויות, הוצאות חודשית קבועות\משתנות ועוד. |
| 3 | Q- מציאת תוכנית כלכלית המתאימה באופן פרטני |

להלן המטרות שהגדרנו:

## גבולות המערכת:

**מערכת המידע כוללת:**

* DATA BASE – מסדי הנתונים פועלים בשלושה רבדים:

פרטי המשתמש- הכוללים שם, שם משתמש, סיסמה, אימייל ומאפשרים למשתמש להזדהות, להתחבר למערכת ולהוסיף משמרות, שכר שעתי, זמני עבודה, התחייבויות ועוד.

חישובי רקע- חישוב שעות נוספות, שכר שעתי נוסף, הוספה לשכר החודשי, עדכון תזרים מזומנים ועוד.

פלט- הוצאת תוכנית חסכון אשר נתונה למשתמש בצע בה שינויים כדי לבחור התאמה אישית.

ממשק ידידותי למשתמש, סביבת UI (User interface) הכוללת מסכי קלט, פלט ותפריטים ברורים.

למשתמש יש אפשרות לצפות בערכים שהוסיף ולערוך אותם בצורה פשוטה.

* במטרה לאפשר נגישות למידע בחרנו לאפשר גישה לנתונים רק למשתמשים שנרשמו לאתר ובכך לאפשר הגנת משתמש ויצור פרטיות למשתמש.
* ממשק בשפה אנגלית – מאפשר גלובליזציה.

**מערכת המידע אינה כוללת:**

* חישוב לעצמאים- האפליקציה מיודעת כרגע לשכירים בלבד.
* חיוב המשתמש לשתף את כל הפרטים (מתוך התחשבות, אך החישובים אינם ישקפו תמונה הקרובה למציאות כפי שמהערכת רוצה)
* ממשק בשפה העברית.
* תפריטי/מסכי עזרה, HELP.
* אפשרות הוספת\עריכה\צפייה למשתמש לא רשום.

## אילוצים:

* **אילוץ זמן –** מערכת המידע צריכה לעבוד ולעמוד במטרותיה לא יאוחר מיום ד' 14.2.19
* **אילוצים כספיים –** מערכת המידע אינה מתוקצבת. כלומר, אין מימון לשיפור ולתחזוקת המערכת ולכן גם תוכנתה ע"י חברי הצוות. במידה וימצא מימון מתאים, ניתן יהיה לשכור מתכנתים לשיפורה.
* **אילוצים טכנולוגיים –** המערכת מתבססת על תשתית אינטרנטית. כלומר יש צורך ביישומים אינטרנטיים (דפדפנים) כגון: Internet Explorer, Chrome, Edge, FireFox.  
  המערכת מיושמת בשרתי גוגל באמצעות Firebase כאשר ימצא מימון ותקציב נכון נוכל ליישם זה על שרת אחר (ענן) ולפתח את המערכת גם לעצמאים.
* **אילוצי כוח אדם –** נדרשת תחזוקה של המערכת שתבוצע על ידי המתכנתים (לאחר קבלת מימון לפרויקט).
* **הנחות יסוד –** המערכת מיודעת למשתמשים שכירים. ללא שיתוף פעולה מצד המשתמשים המערכת תאבד את מטרתה.
* **אילוצי סף –** המערכת תתבסס על כלים בסיסיים של סביבה אינטרנטית כגון:

HTML, Java Script, JQuery, CSS, PHP, SQL, AJAX

## תהליכים:

* **משתמשים:** שכירים, גלובלים.
* **אירועים:** הרשמה, כניסת משתמש למערכת, הוספת משמרת, הוספת התחייבויות, צפייה בנתונים שהוזנו ועריכתם.
* **תדירות:** משתנה ממשתמש למשתמש ולכן אינה קבועה ופתוחה למשתמש בכל מקום ובכל זמן.
* **הקלט לתהליך:**

**א.** מקור הנתונים וצורת מסירתם – נתוני הקלט מוזנים על ידי המשתמשים המשתפים. נתוני הקלט נמסרים באופן ממוחשב על ידי מילוי שדות מתאימים.

**ב.** אמצעי קליטת הנתונים – מסך, מקלדת המחשב ועכבר או על ידי טלפון בעל גישה לאינטרנט.

**ג.** פירוט הנתונים שיש לקלוט וכמותם –  
שם ההוצאה: במידה ויש לרשום רק את שם ההוצאה לזיהוי.

שם ההכנסה: במידה ויש לרשום רק את שם ההוצאה לזיהוי.

כמות הנתון: ערך להוצאה\הכנסה\משמרת.

תאריך המשמרת: תאריך ביצוע משמרת.

שעת כניסה ויציאה מהמשמרת: לפי שעון 24 שעות

שכר שעתי: אינדיבידואלי.

* **הפלט לתהליך:** קבלת תוכנית חסכון בדרישות המשתמש.

**א.** יעד הפלט – תזרים מזומנים שמתעדכן בכל פעילות \ עדכון המשתמש.

**ב.** אמצעי הפקת הפלט וצורת מסירתו ליעדו –גרפית (מסך)(בעתיד ברצוננו הדפסת הפלט ושליחתו במייל, הצגת גרף השוואתי ועוד).

* **הנתונים הנשמרים במאגרי הנתונים והנשלפים מהם:  
  א.** סוג הפעולה כתיבת נתונים למערכת נעשית באמצעות טפסים (Forms).  
  בכל פעם שמשתמש רוצה להוסיף משמרת למערכת או להוסיף הוצאה\הכנסה עליו למלא טפסים שפרטיהם נשמרים במערכת המידע לאחר שליחתם (לחיצה על כפתור).  
  נוסף לכך, במערכת ישנן פעולות קריאה ממאגרי הנתונים. כאשר מבצע סינון על פי הקריטריונים שבחר – מתבצעת קריאה ממאגרי המידע.  
  **ב.** מהם הנתונים שיש לקרוא או לכתוב ומה כמותם? הנתונים שיש לקרוא או לכתוב הם נתוני המשתמש(גיל, מין, סטטוס ועוד), סוג הכנסה\הוצאה, סכום כספי, שם ההוצאה\הכנסה, תאריך הפעולה, חישוב החזר מס, חישוב ביטוח לאומי ועוד.

# **תיאור לוגיקת התהליך:**

**רישום למערכת**

* **התחלה**
* הצג טופס התחברות למערכת.
* מילוי פרטי המשתמש: אימייל ,שם פרטי ומשפה, מין, תאריך לידה, סטטוס משפחתי, סוג משכורת , מטבע, שכר שעתי \ גלובלי
* קלוט נתוני רישום
* בדוק את הקלט
* אחרת, השלם רישום
* **סיום**

**התחברות למערכת**

* **התחלה**
* הצג טופס התחברות למערכת.
* קלוט שם משתמש וסיסמה.
* בדוק שם משתמש וסיסמה.
* אם שם המשתמש והסיסמה תקינים ותואמים - התחבר למערכת.
* אחרת, הצג "שם משתמש/סיסמה לא תקינים".
* **סיום**

**תהליך הוספת משמרת (משתמש רשום בלבד)**

* **התחלה**
* הצג טופס הוספת משמרת
* קלוט נתונים מהמשתמש.
* שליחת טופסובדיקתו.
* הכנס את הנתונים הנ"ל למסד הנתוניםבהתאמה**.**
* עדכון מסד הנתונים
* **סיום**

**תהליך הוספת התחייבויות\הוצאות**

* **התחלה**
* הצג טופס הוספת התחייבויות.
* קלוט נתונים מהטופס.
* הכנס את הנתונים למסד הנתונים (התחייבויות).
* עדכן נתונים במסד הנתונים.
* **סיום**

**חלוקת העבודה בין חברות הצוות**

חן שליו- אפיון ראשוני ודוח ייזום, בקרה מול המתכנת.

מיכאל ירמולייב- אפיון ודוח יזום, דוח פרויקט.

נועם אמרון-QA ועיצוב המערכת ובקרה מול המתכנת.

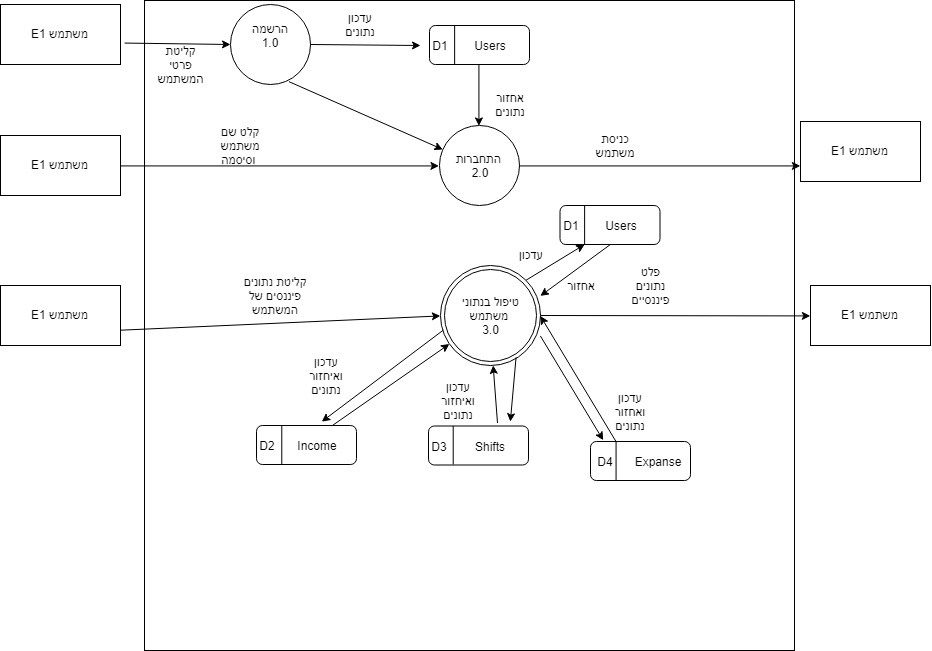
לאחר שבחרנו את הנושא לפרויקט, החלטנו להיפגש פעמיים בשבוע על מנת לאפיין יחד את המערכת באופן שישקף את המטרה שלשמה החלטנו להקימה.   
לאחר קבלת מסמכי האפיון, חילקנו בינינו את העבודה כמפורט לעיל ונפגשנו פעם ביומיים על מנת להתעדכן בשינויים המתרחשים ובכדי להתאים את המשך העבודה בהתאם לשינויים אלה. לקראת מועד ההגשה, נפגשנו בכל אחד מימות השבוע לשם ביצוע שיפורים ובדיקות אחרונות.

# ניתוח מערכת

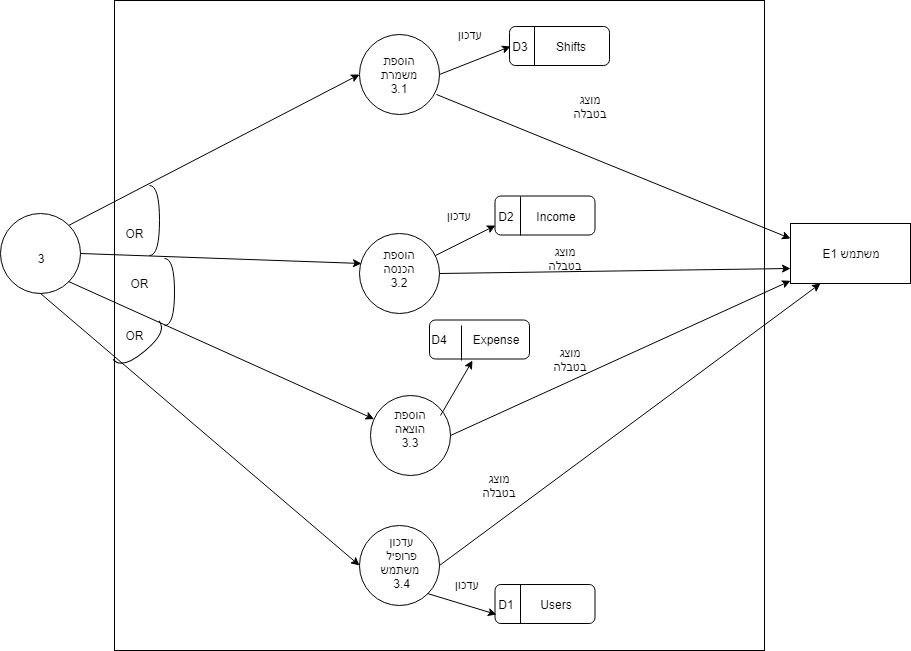
## תרשימי DFD

"תרשים DFD (Diagram Flow Data) הוא אמצעי גרפי לתיאור פעילויות וזרימת מידע ביניהן. מטרתנו בשימוש תרשימי DFD היא להציג את הפעילויות שהמערכת מבצעת ואת המידע הזורם בין הפעילויות, כלומר, הקלט הדרוש כדי לבצע כל פעילות והפלט הנוצר בעקבות ביצועה. בניגוד לתרשים זרימה של תוכנית, אין מטרתו של תרשים ה DFD-להציג לוגיקת ביצוע הפעילויות" (פרץ שובל, 2015.(

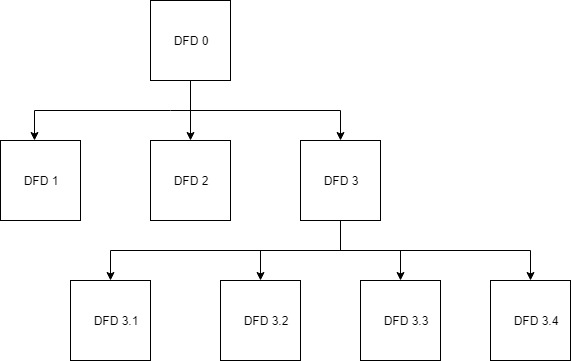
DFD-0



DFD-3



**עץ תרשימים:**

****

**מילון נתונים:**

"מילון הנתונים מתעדכן ומלווה את מערכת המידע בכל שלבי הפיתוח" (פרץ שובל,2015).

מילון הנתונים עוזר למנוע כפילויות, מקל בביצוע שינויים של נתונים ומסייע למפתחי ומנתחי המערכת למצוא נתונים.

מילון נתונים מורכב משני מילונים: מילון מרכיבי ה-DFD ומילון רכיבי הנתונים.  
באמצעות מילון מרכיבי ה-DFD, ניתן לדעת מה מצוי בתרשימים השונים, הקשרים בין כל אחד ומידע על התהליכים, מאגרים, ישויות וזרמי הנתונים.

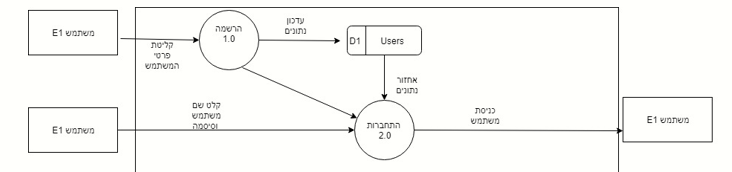
***ראה נספח מס' 1: מילון מרכיבי DFD***

# עיצוב מערכת

## טרנזקציות

"טרנזקציה היא רצף של פונקציות יסודיות הקשורות זו לזו על ידי זרמי מידע. הטרנזקציה כוללת גם את זרמי המידע, את מאגרי המידע ואת הישויות החיצוניות הקשורות אל הפונקציות שלה. טרנזקציה כוללת לפחות ישות חיצונית אחת שהיא "הדק" המאפשר את הפעלתה לפי הצורך". (פרץ שובל, 2015)

**טרנזקציה מס' 1:**



**התחל טרנזקציה**

קלוט פרטים אישיים מישות E1 - משתמש

בצע תהליך 1.0 - הרשמה

כתוב למאגר D1 – Users

בצע פונקציה 2.0 – התחברות

הפק פלט כניסה למערכת לישות E1 - משתמש

**סוף טרנזקציה**

**טרנזקציה מס' 2:**

**התחל טרנזקציה**

קלוט מישות E1 נתונים פיננסים- משתמש

אם הפונקציה היא הוספת משמרת אז עדכן את D3.

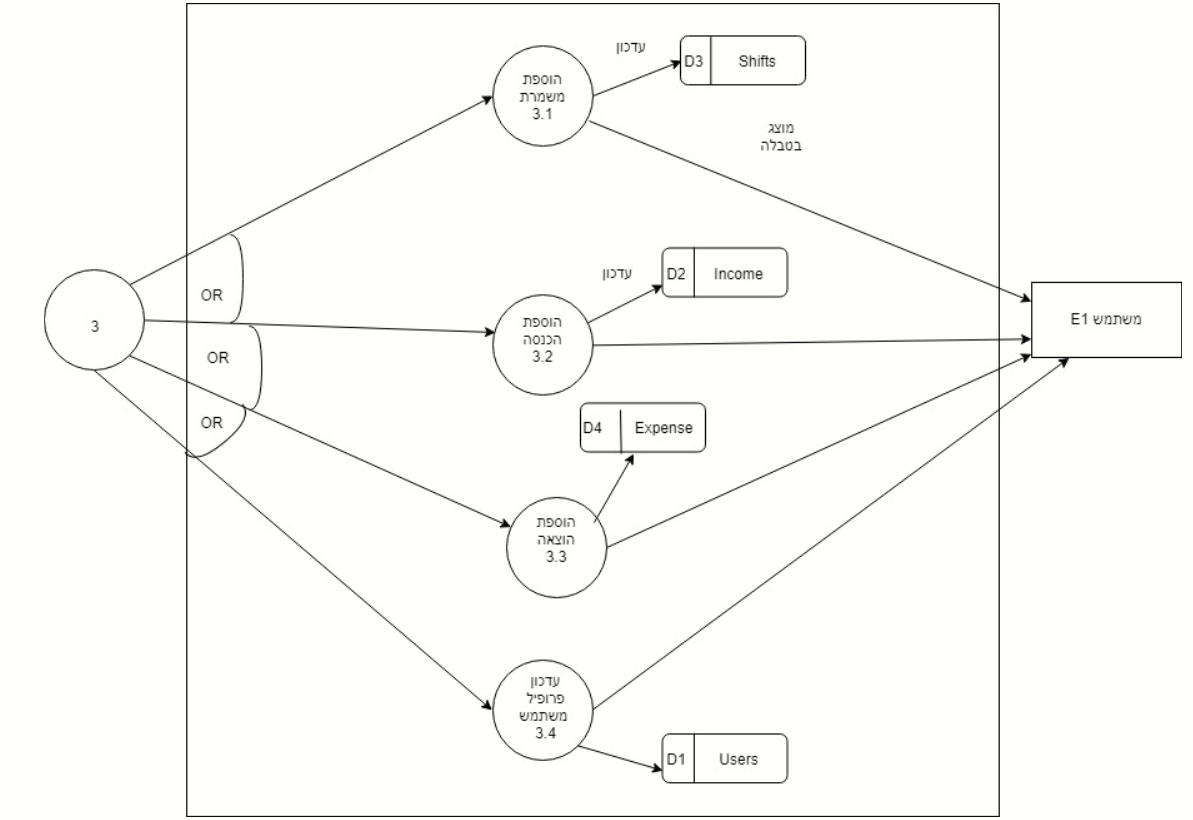
אם הפונקציה היא הוספת הוצאה אז עדכן את D4.

אם הפונקציה היא הוספת הכנסה אז עדכן את D2.

אם הפונקציה היא עדכון פרופיל אז עדכן את D1.

הפק פלט לישות E1.

**סוף טרנזקציה**



## עץ תפריטים

קיימים שני שלבים בבניית עץ תפריטים: השלב האלגוריתמי והשלב ההידברותי.

**1. השלב האלגוריתמי**

**צעד 1:** גזירת עץ תפריטים מתרשימי DFD

**T 1.0**

**T 2.0**

**T 3.1**

**T 3.2**

**T 3.3**

**T 3.4**

**צעד 2+3+4** – לא נעשה שינוי

**T 1.0**

**T 2.0**

**T 3.1**

**T 3.2**

**T 3.3**

**T 3.4**

**2. השלב ההידברותי**

מטרתו של שלב זה היא לשפר את עץ התפריטים שנוצר בשלב האלגוריתמי ולעצב את צורתו החיצונית.

הרשמה

התחברות

הוספת משמרת

הוספת הכנסה

הוספת הוצאה

עדכון פרופיל משתמש

## **תמונה שמכילה טקסט, מפה תיאור שנוצר ברמת מהימנות גבוהה מאודתרשים** **ERD**

# תכנות – בדיקות – הטמעה

## **תרשים מערכת:**

מפרט את התוכנה (מערכת הפעלה, שפת תוכנה ובסיס נתונים) והחומרה בשימוש בפרויקט.

מסד נתונים: NOSql

ממשק: FireBase

כלי פיתוח:

שפת תוכנה: HTML , JavaScript , Angular

שפות נוספות: PHP, SQL

עיצוב: CSS, Bootstrap

חומרה: השרת של Google

הוצאות\הכנסות

משמרות

פרטי משתמש

מחשב

מסך הכנסות והוצאות- קבלת נתוני הכנסות ושכר של המשתמש.

מסך משמרות - קבלת נתוני משמרות של המשתמש

FireBase

# יישום ובדיקות

**בניית המערכת:**

כיוון שרצינו שהמערכת תהיה נגישה לכל, החלטנו לפתח את המערכת בסביבת אינטרנט אשר מאפשרת גישה קלה ונוחה.

תחילה, בנינו את חלון הצגת הנתונים של המשתמש . בחלון זה ניתן לראות את כל הנתונים שהמשתמש הזין למערכת. בצד ימין למעלה ישנו Balance והוא למעשה תזרים המזומנים של המשתמש המתעדכן בזמן אמת , אם זה מצב ההכנסות ואם מצב ההוצאות , הכל מתאפשר בעזרת iFrame וAngular.

לאחר מכן הוספנו את חלון הוספת המשמרות וחלון להוצאות והכנסות, וכמובן את טבלאות הנתונים הרלוונטיות לפעילות תקינה של המערכת. בחלק זה שמנו דגש על הלוגיקה שמוטמעת בתוך הדף , לוגיקה המאפשרת למערכת לחשב את נתוני ההכנסות וההוצאות של המשתמש בצורה נכונה.

לאחר שהצלחנו להקים את הליבה של האתר, התחלנו להוסיף פונקציות ולוגיקה סטנדרטית עבור מערכת מידע. הוספנו אפשרויות סינון, התחברות, הגבלה למשתמשים לא רשומים, נתוני מס הכנסה ונתונים רלוונטיים נוספים תוך התחשבות בתוכן מסדי הנתונים, דרישות הלקוחות והפידבקים שקיבלנו.

במהלך בניית המערכת הושקעה מחשבה רבה בעיצוב הממשק כך שיהיה ידידותי ונוח למשתמש.

**בדיקות:**

בדיקות תוכנה (Software Testing) הינן חלק הכרחי בתהליך פיתוח מערכת המידע.   
מטרת בדיקות התוכנה היא לוודא שהמערכת עונה על דרישות הלקוח שצוינו בשלב האפיון הראשוני וכן לוודא את תקינותה.

הבדיקות כוללות בין היתר הרצת קוד כתוב והזנת קלטיים שונים (בעיקר לבחינת מקרי קיצון). הן בוחנות מתן משוב וחוקרות התנהגות בלתי צפויה של המערכת.

כחלק בלתי נפרד מהתהליך, נעשה במערכת שימוש ע"י משתמשים המאפשר גילוי שגיאות שייתכן שלא היו מתגלות על ידינו או ע"י המתכנת.

לפי מתודולוגית Agile, הבדיקות הן חלק אינטגרלי בפיתוח המערכת. לכן, ערכנו בדיקות תכופות על מנת לבדוק את תקינותה של המערכת הסופית ולמנוע בעיות העלולות להתפתח בהמשך.

מטרות הבדיקות:

* למצוא באגים ולגרום לך שיתוקנו על מנת שלא ליצור באגים גדולים יותר בהמשך.
* לוודא שהמערכת עונה על דרישות הלקוח בכל שלב.
* להעריך את איכות המוצר.

סוגי הבדיקות:

* בדיקות פונקציונאליות: מאמתות את פעילותה של המערכת. בבדיקות מסוג זה נוודא שהמערכת מציגה את התוכן הרצוי בצורה שמשקפת את האפיון ומביאה לידיי ביטוי את מטרות המערכת.
* בדיקת יחידה: בדיקות מסוג זה כוללות בדיקות תקינות קלטיים שונים אשר מוודאים את תקינות הפונקציות במערכת. השוואה בין תוצאות הפונקציות לבין תוצאות ידניות.
* בדיקות ממשק לקוח (GUI): בדיקות כי הפקודות והשדות תקינים בהתאם למה שהגדרנו.
* בדיקות שימושיות: בדיקות שבוחנות את מידת השימושיות והנוחות של המשתמש במערכת. בוצעו על ידי קבלת משוב מלקוחות שעשו שימוש במערכת בזמן הפיתוח (בהתאם לשיטת Agile Scrum).

הבדיקות שבוצעו:

בדיקות התקינות נעשו בצד הלקוח, לאחר לחיצה על הכפתורים השונים (שלח/כניסה). אם נמצאו שגיאות, הצגנו הודעות בהתאם ולאחר תיקון השגיאות – הועבר הטופס לשרת.

להלן פירוט הבדיקות:

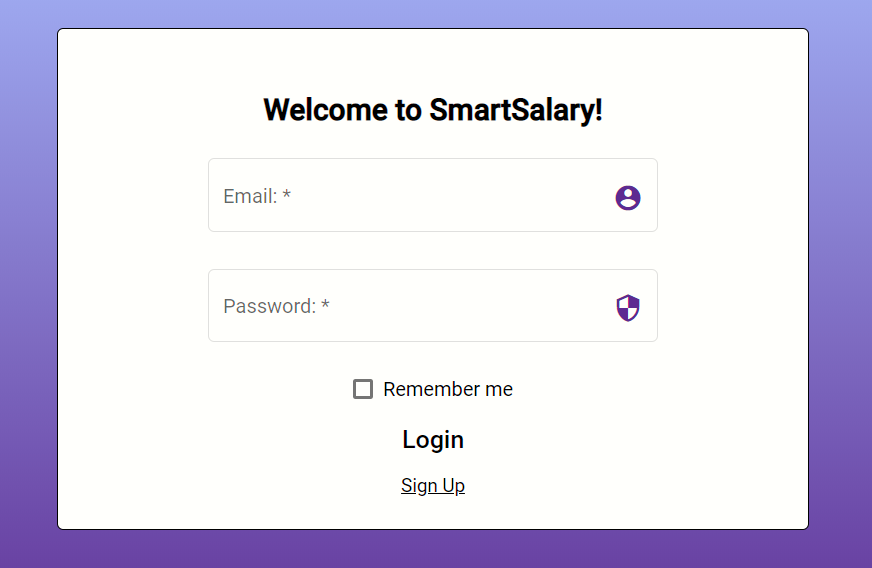
* בדיקת תקינות קלט: וידאנו כי הקלטיים הינם מהסוג הנדרש.
* כתובת אימייל חוקית - וידאנו כי סיומת הקלט הינה לפי התבנית: [\_\_\_\_@\_\_\_.com](mailto:____@___.com) זאת בעזרת פונקציה מובנת , דבר זה חשוב במיוחד כי כתובת האימייל של המשתמש משמשת כמפתח הראשי של המערכת מידע.
* שדות חובה – וידאנו כי לא ניתן לשלוח את טופס ההרשמה עד אשר ימולאו כל שדות החובה.
* סיסמאות זהות.
* בדיקת התחברות למערכת.
* בדיקת תקינות כפתורים, קישורים ופונקציות.
* בדיקה של תקינות הלוגיקה של הפונקציות הקריטיות:
* הוספת משמרת
* הוספת הוצאה קבועה/חד - פעמית
* חישוב נכון של שכר וחוב העתידי של המשתמש
* בדיקת תקינות פלט: וידאנו כי הפלטים נכונים.

למשל: כאשר המשתמש מזין את מצבו המשפחתי , המערכת תדע להתחשב בנתון זה כאשר היא מחשבת את השכר ברוטו ונטו של המשתמש.

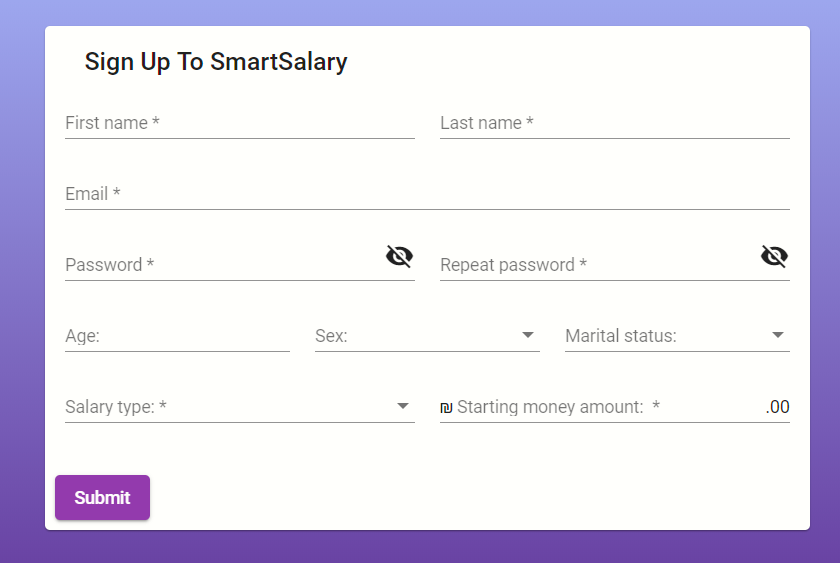
# מערכת המידע

***ראה נספח מס' 3: מערכת המידע***

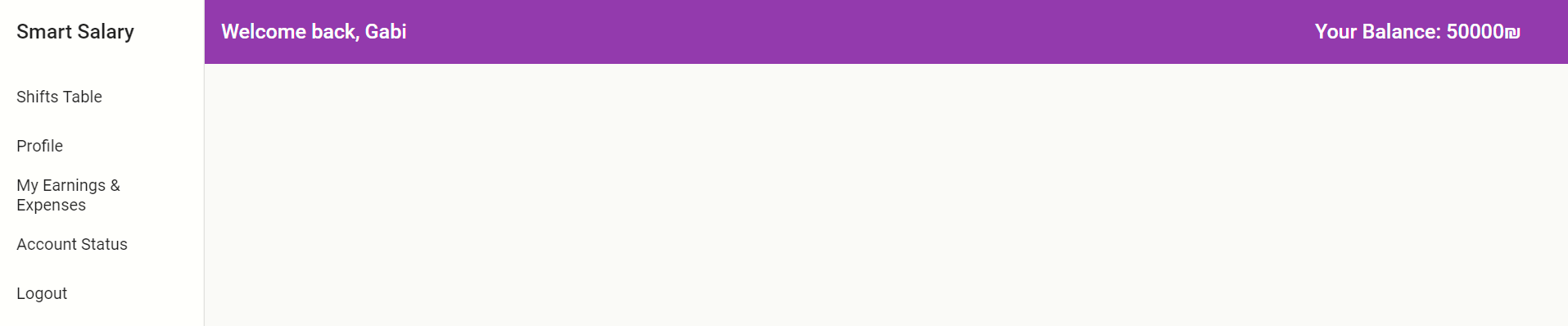
מסך כניסה למערכת:



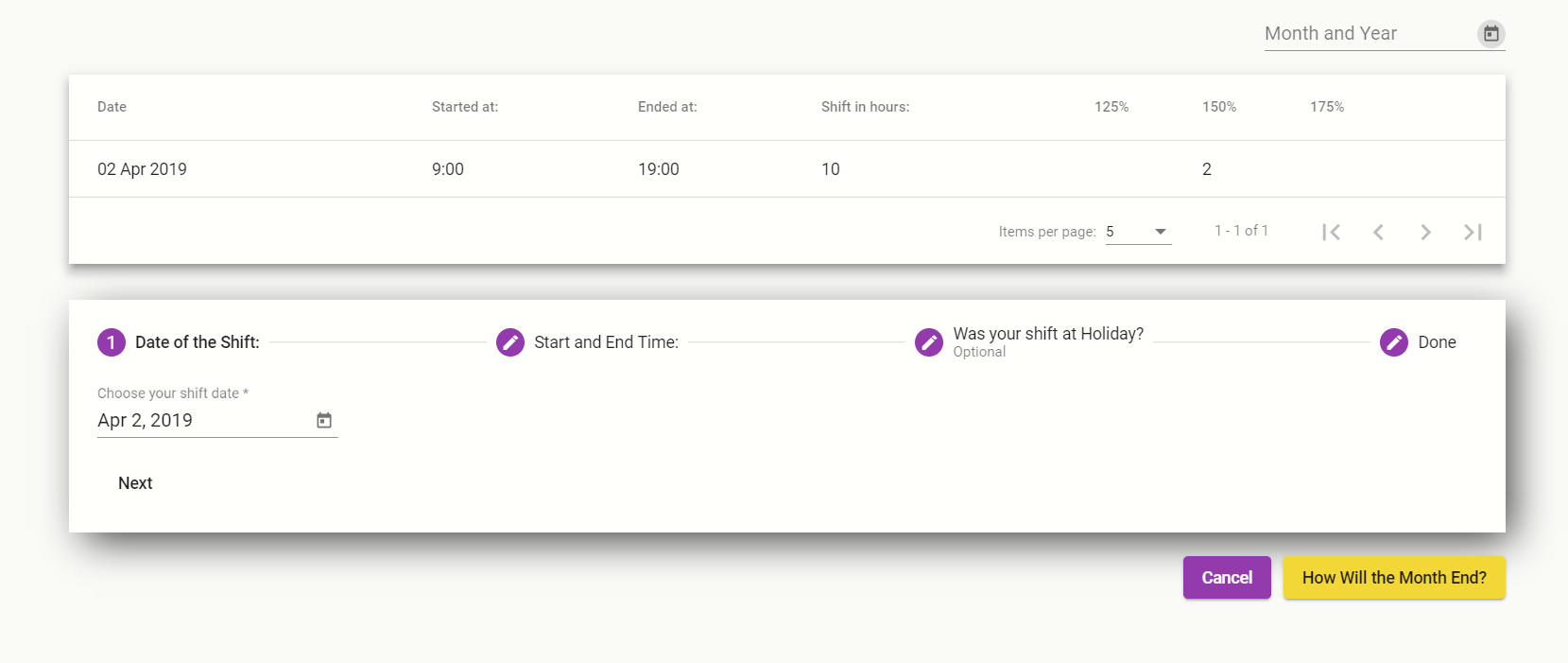
מסך רישום למערכת:



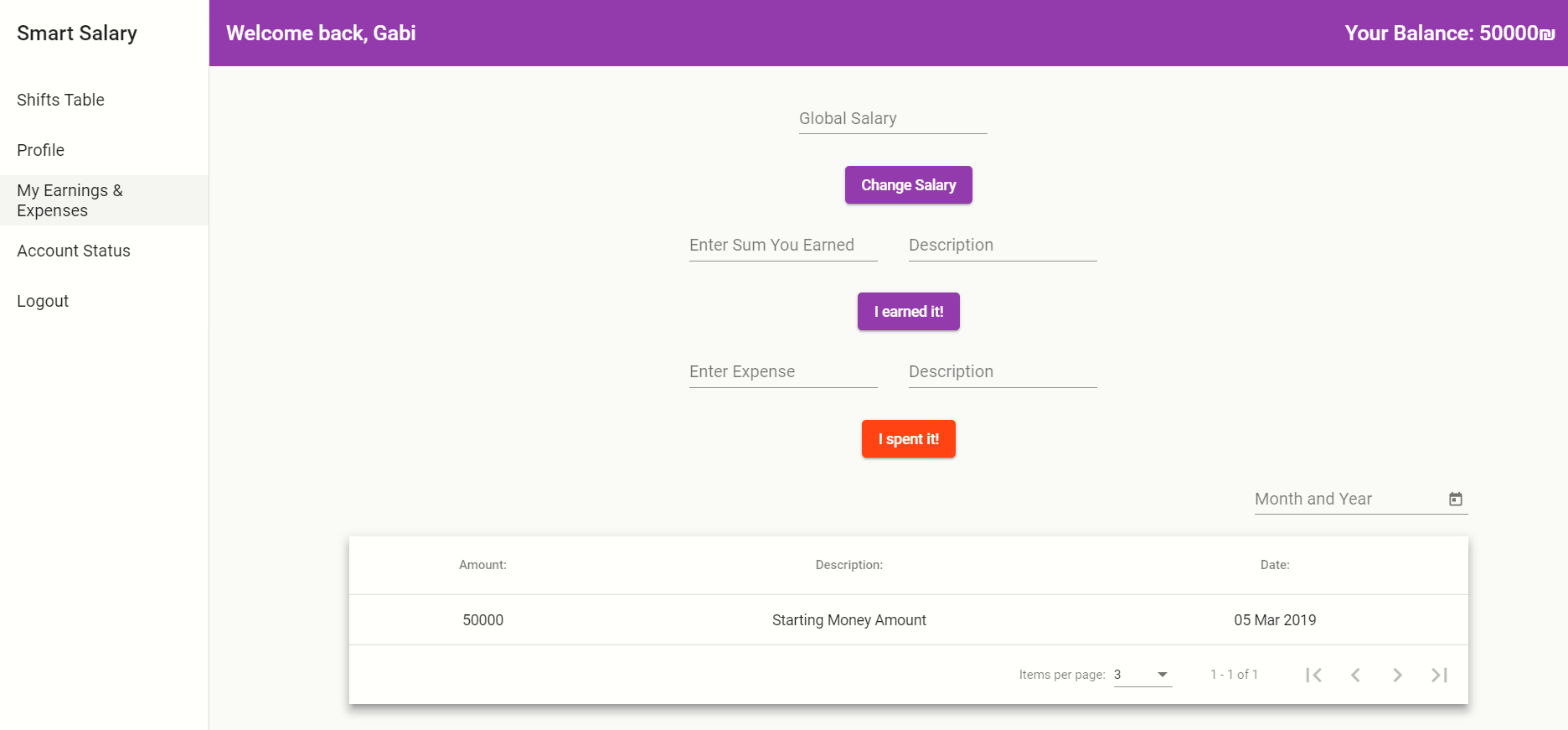
מסך לאחר התחברות ראשונית למערכת:



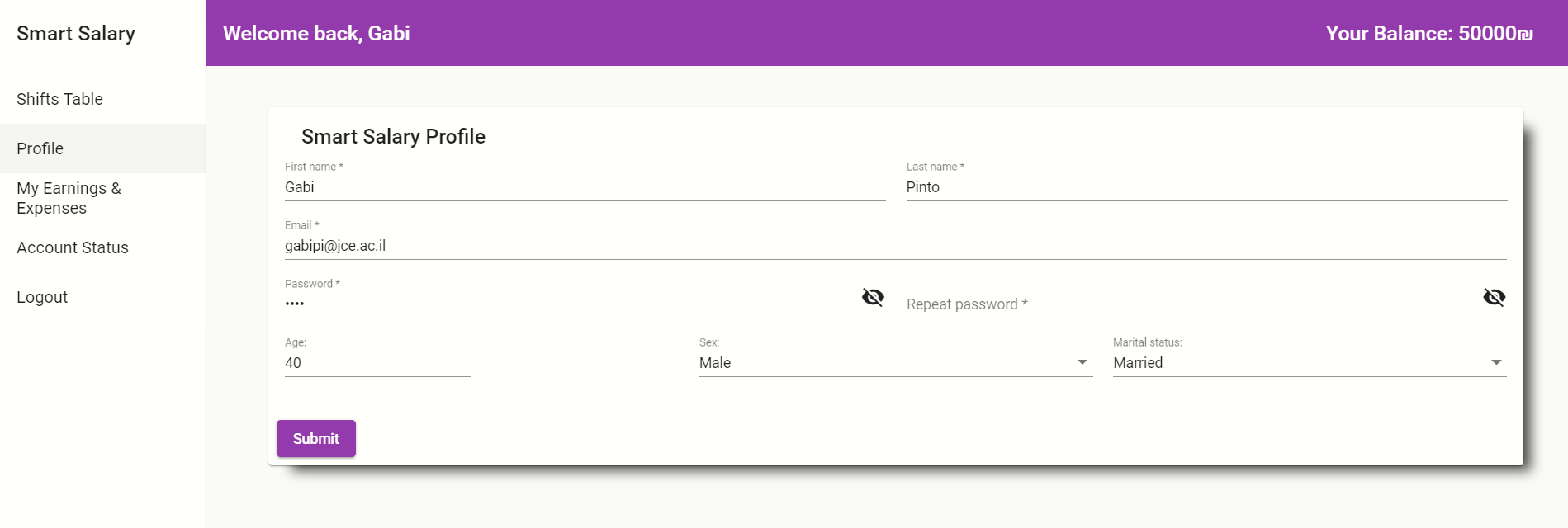
מסך הוספת משמרות:



מסך הוספת הוצאות/הכנסות:



מסך עדכון פרופיל:



**ד. הטמעה**

הטמעת המערכת התבצע בכמה אופנים:

* פרסום ושיווק ברשתות חברתיות ובאינטרנט
* שיתוף בין חברים

ניהול תצורה: נערוך בדיקות שוטפות כי המערכת פועלת כנדרש.

אחזקה: נכון להיום, המערכת מבוססת על גבי השרת של Google ולכן היא ללא עלות.   
בהמשך, נוכל להרחיב את פיתוח המוצר והוספת פיצ'רים לאתר.

**ה. תוצאות**

הצגת התוצאות במדדי הביצוע:

את איכות המערכת בחנו באמצעות מדד שביעות רצון המשתמש שנבדק באמצעות מתן ציון לגרסת הראשונית של הפיתוח.

מתוך הסקר עלה כי שביעות רצון המשתמשים עמד על 3.5 מתוך 5.

נראה כי היתרון האפשרות למקם את ההוצאות, הכנסות והמשמרות במקום אחד קיבל ציון מרבי.

תוך כדי השימוש באתר, קיבלנו עצות מהמשתמשים ודיווחים בזמן אמת על האם יש משהו שלא תקין.

נכון להיום ישנם עוד מספר באגים אשר עלינו לתקן וזאת בעקבות חוסר מימון.

כתוצאה מכך, שיפרנו את מדדי הביצוע הבאים:

* **זמן (Time)** – השימוש במערכת חוסך זמן חיפוש באתרי אינטרנט והצלבה ממקורות רבים( אתר הבנק, אפליקציית שכר, אפליקציית האשראי..)
* **עלות (Cost)** – מסייע בבחינת התקציב ותזרים המזומנים החודשי של האדם בכל רגע נתון.
* **איכות (Quality)** – שמירה והיבדלות משאר האפליקציות בשוק בהוספת חישובי מס הכנסה בהתאם לגיל ורמת ההכנסה.

בנוסף לסקר, ביקשנו מהמשתתפים הצעות לשיפור ולשימור המערכת ושינינו בהתאם. כתוצאה מכך, המערכת הפכה לשימושית יותר וידידותית למשתמש.

# סיכום הפרויקט

**א. מסקנות:**

* הבחירה במתודולוגיית פיתוח Agile הייתה נכונה במיוחד לאור העובדה שהכוח הטמון במערכת הוא משתמשי המערכת. בכדי שהמערכת תהפוך לשמישה ופעילה, יש צורך במשתמשים משתפים ובמשתמשים שיעשו שימוש שוטף במערכת ויפיצו אותה. לאור כך, השימוש בבדיקות וקבלת משוב תוך כדי בניית המערכת, סייע לנו באופן ניכר בהשגת מטרה זו.  
  במילים אחרות, הבחירה במתודולוגיית המאפשרת גמישות בבניית המערכת הייתה נכונה. ערכנו שינויים במערכת בהתאם לדרישות הלקוח בכל שלב.
* המערכת אינה מערכת גמישה המאפשרת שימוש גם למשתמשים לא רשומים.
* המערכת מציגה בצורה ברורה כל הוצאה או הכנסה שיש למשתמש.
* המערכת מציגה באופן ברור את תזרים המזומנים של המשתמש.
* מערכת המידע חוסכת למשתמש זמן יקר ומאפשרת קבלת מידע במקום אחד.
* לאחר סיום הפרויקט ביקשנו מקרובינו להשתמש במערכת ולתת עליה משוב.
* הבנו כי סיעור מוחין זוהי שיטה יעילה לשימוש בפרויקטים קבוצתיים, אשר גורמת ליישור קו בין כל חברי הקבוצה.

**ב. המלצות להמשך:**

* להוסיף פיצ'רים נוספים למערכת.
* שיפור איכות החישוב.
* המרת האתר לאפליקציה כדי לאפשר לכל משתמש להוריד אותה מחנות האפליקציות.
* חיפוש מימון המאפשר את הוספת ושדרוג המערכת.

# ביבליוגרפיה

* פרץ שובל, **תכנון ניתוח ועיצוב מערכות מידע**, כרך א' ,תל אביב: האוניברסיטה הפתוחה, מהדורה שנייה, 2015.
* פרץ שובל, **תכנון ניתוח ועיצוב מערכות מידע**, כרך ב' , תל אביב: האוניברסיטה הפתוחה, מהדורה שנייה, 2015.
* פרץ שובל, **תכנון ניתוח ועיצוב מערכות מידע**, כרך ג' , תל אביב: האוניברסיטה הפתוחה, מהדורה שנייה, 2015.
* מצגות הקורס "ניתוח ותכנון מערכות מידע א' 30037"
* מצגות הקורס "מסדי נתונים 30030"

# נספחים

## מילון מרכיבי DFD:

1. **מילון תהליכים:**

|  |  |
| --- | --- |
| זיהוי התהליך: **1.0** | |
| שם התהליך: **הרשמה** | |
| סוג: יסודי | |
| זרמי נתונים נכנסים: | |
| שם הזרם: | המקור: |
| קלט פרטים אישיים | E1 |
| זרמי נתונים יוצאים: | |
| שם הזרם: | היעד: |
| עדכון נתונים | D1 |
| המשך פעולה | 2.0 |
| תיאור כללי של התהליך: ביצוע הרשמה למערכת. | |

|  |  |
| --- | --- |
| זיהוי התהליך: **2.0** | |
| שם התהליך: **התחברות** | |
| סוג: יסודי | |
| זרמי נתונים נכנסים: | |
| שם הזרם: | המקור: |
|  | 1.0 |
| אחזור | D1 |
| קלט שם משתמש וסיסמא | E1 |
| זרמי נתונים יוצאים: | |
| שם הזרם: | היעד: |
| כניסת משתמש | E1 |
| תיאור כללי של התהליך: ביצוע התחברות משתמש למערכת. | |

|  |  |
| --- | --- |
| זיהוי התהליך: **3.0** | |
| שם התהליך: **טיפול בנתוני משתמש** | |
| סוג: מורכב | |
| זרמי נתונים נכנסים: | |
| שם הזרם: | המקור: |
| קלט פרטי משתמש | E1 |
| אחזור | D2 |
| אחזור | D3 |
| אחזור | D4 |
| זרמי נתונים יוצאים: | |
| שם הזרם: | היעד: |
| פלט תכנון פיננסי | E1 |
| עדכון | D2 |
| עדכון | D3 |
| עדכון | D4 |
| תיאור כללי של התהליך: ביצוע פונקציות על ידי המשתמש. | |

|  |  |  |
| --- | --- | --- |
| זיהוי התהליך: **3.1** | | |
| שם התהליך: **הוספת משמרת** | | |
| סוג: יסודי | | |
| זרמי נתונים נכנסים: | | |
| שם הזרם: | המקור: | |
|  | פונקציה 3 | |
| זרמי נתונים יוצאים: | | |
| שם הזרם: | היעד: | |
| פלט נתוני משמרת | E1 | |
| עדכון | D3 | |
| תיאור כללי של התהליך: הוספת משמרת. | | |
| זיהוי התהליך: **3.2** | | |
| שם התהליך: **הוספת הכנסה** | | |
| סוג: יסודי | | |
| זרמי נתונים נכנסים: | | |
| שם הזרם: | | המקור: |
| טיפול בנתוני משתמש | | פונקציה 3 |
| זרמי נתונים יוצאים: | | |
| שם הזרם: | | היעד: |
| פלט נתוני הכנסות | | E1 |
| עדכון | | D2 |
| תיאור כללי של התהליך: הוספת נתוני הכנסה | | |

|  |  |
| --- | --- |
| זיהוי התהליך: **3.3** | |
| שם התהליך: **הוספת הוצאה** | |
| סוג: יסודי | |
| זרמי נתונים נכנסים: | |
| שם הזרם: | המקור: |
| טיפול בנתוני משתמש | פונקציה 3 |
| זרמי נתונים יוצאים: | |
| שם הזרם: | היעד: |
| פלט נתוני הוצאות | E1 |
| עדכון | D4 |
| תיאור כללי של התהליך: הוספת נתוני הוצאה חדשה | |

|  |  |
| --- | --- |
| זיהוי התהליך: **3.4** | |
| שם התהליך: **עדכון פרופיל משתמש** | |
| סוג: יסודי | |
| זרמי נתונים נכנסים: | |
| שם הזרם: | המקור: |
| טיפול בנתוני משתמש | פונקציה 3 |
| אחזור | D1 |
| זרמי נתונים יוצאים: | |
| שם הזרם: | היעד: |
| עדכון | D1 |
| פלט נתוני משתמש | E1 |
| תיאור כללי של התהליך: מעדכן את נתוני פרופיל המשתמש. | |

## מילון מאגרי נתונים

|  |  |
| --- | --- |
| זיהוי מאגר הנתונים: **D1** | |
| שם המאגר: **Users** | |
| זרמי נתונים נכנסים מתהליכים (עדכונים): | |
| שם הזרם: | המקור: |
| עדכון | 1.0 |
| זרמי הנתונים יוצאים מתהליך (קריאות): | |
| שם הזרם: | היעד: |
| אחזור | 2.0 |
| אחזור | 3.0 |
| תיאור כללי של המאגר: מאגר משתמשים הכולל שם פרטי, שם משתמש, סיסמא. | |

|  |  |
| --- | --- |
| זיהוי מאגר הנתונים: **D3** | |
| שם המאגר: **Shifts** | |
| זרמי נתונים נכנסים מתהליכים (עדכונים): | |
| שם הזרם: | המקור: |
| עדכון | 3.0 |
| עדכון | 3.1 |
| זרמי הנתונים יוצאים מתהליך (קריאות): | |
| שם הזרם: | היעד: |
| אחזור | 3.0 |
| תיאור כללי של המאגר: מאגר נתוני אטרקציות הכולל: שם אטרקציה, שם טיול, תקציר, עלות אטרקציה, קישור למפה. | |

|  |  |
| --- | --- |
| זיהוי מאגר הנתונים: **D2** | |
| שם המאגר: **Income** | |
| זרמי נתונים נכנסים מתהליכים (עדכונים): | |
| שם הזרם: | המקור: |
| עדכון | 3.0 |
| עדכון | 3.2 |
| זרמי הנתונים יוצאים מתהליך (קריאות): | |
| שם הזרם: | היעד: |
| אחזור | 3.0 |
| תיאור כללי של המאגר: מאגר נתוני הטיולים הכולל: שם טיול, תקציר, עלות טיסה, עלות לינה, מס' ימים, תמונה, סוג טיול, מחיר כולל. | |

|  |  |
| --- | --- |
| זיהוי מאגר הנתונים: **D4** | |
| שם המאגר: **Expense** | |
| זרמי נתונים נכנסים מתהליכים (עדכונים): | |
| שם הזרם: | המקור: |
| עדכון | 3.0 |
| עדכון | 3.3 |
| זרמי הנתונים יוצאים מתהליך (קריאות): | |
| שם הזרם: | היעד: |
| אחזור | 3.0 |
| תיאור כללי של המאגר: מאגר נתוני אטרקציות הכולל: שם אטרקציה, שם טיול, תקציר, עלות אטרקציה, קישור למפה. | |

## מילון ישויות

|  |  |
| --- | --- |
| שם הישות: **E1 משתמש** | |
| זרמי נתונים יוצאים לתהליכים (קלטים): | |
| שם הזרם: | היעד: |
| קלט פרטים אישיים | 1.0 |
| קלט שם משתמש וסיסמה | 1.0 |
| קלט מין | 1.0 |
| קלט סוג משכורת | 1.0 |
| קלט סטטוס | 1.0 |
| קלט כמות שכר | 1.0 |
| זרמי נתונים נכנסים מתהליך (פלטים): | |
| שם הזרם: | המקור: |
| כניסת משתמש | 2.0 |
| קלט הוספת משמרת | 3.1 |
| קלט הוספת הכנסה | 3.2 |
| קלט עדכון הוצאה | 3.3 |
| קלט עדכון פרופיל | 3.4 |
| תיאור כללי של הישות: משתמש שמחובר למערכת ויש לו היכולת להוסיף ולעדכן נתונים פיננסים. | |

## מילון זרמי נתונים

|  |  |  |
| --- | --- | --- |
| **היעד** | **המקור** | **שם זרם נתונים** |
| 1.0 | E1 | קלט פרטים אישיים |
| D1 | 1.0 | עדכון |
| 2.0 | D1 | אחזור |
| 2.0 | E1 | קלט שם משתמש וסיסמא |
| E1 | 2.0 | כניסת משתמש |
| 3.0 | E1 | קלט טיפול בנתוני משתמש |
| D2 , D1,D3,D4 | 3.0 | עדכון |
| D2 , D1,D3,D4 | D2 | אחזור |  |  |
| E1 | 3.0 | פלט טיפול נתוני משתמש |  |  |
| 3.1 | 3 | הוספת משמרת |  |  |
| 3.2 | 3 | הוספת הכנסה |  |  |
| 3.3 | 3 | הוספת הוצאה |  |  |
| 3.4 | 3 | עדכון פרופיל משתמש |  |  |
| D3 | 3.1 | עדכון |  |  |
| D2 | 3.2 | עדכון |  |  |
| D4 | 3.3 | עדכון |  |  |
| D1 | 3.4 | עדכון |  |  |
| E1 | 3.1 | הפקת פלט | VPE, |  |
| E1 | 3.2 | הפקת פלט |  |  |  |  |
| E1 | 3.3 | הפקת פלט | הפקת פלט |  |  |  |
| E1 | 3.4 | הפקת פלט |  |  |  |  |

## נספח מס' 3: מערכת המידע

<https://smart-salary.firebaseapp.com/#/>