

ספר פרויקט

EzDelivery



מגישים:

תומר עפרוני - 205711682

ליאור סברי - 313137937

רועי אספורטס - 205718083

תאמר מסארוה - 207746744

עומרי קלנר - 206156663

**תקציר מנהלים**

EzDelivery היא פלטפורמת משלוחים חברתית המאפשרת לכל אדם פרטי להזמין שליח בלחיצת כפתור בשיטת המכרז.

מהצד השני, EzDelivery נותנת הזדמנות לכל אדם להיות שליח ולתת הצעה עבור חבילה הנמצאת באזור שלו וכך השליח יוכל גם להרוויח את דמי המשלוח.

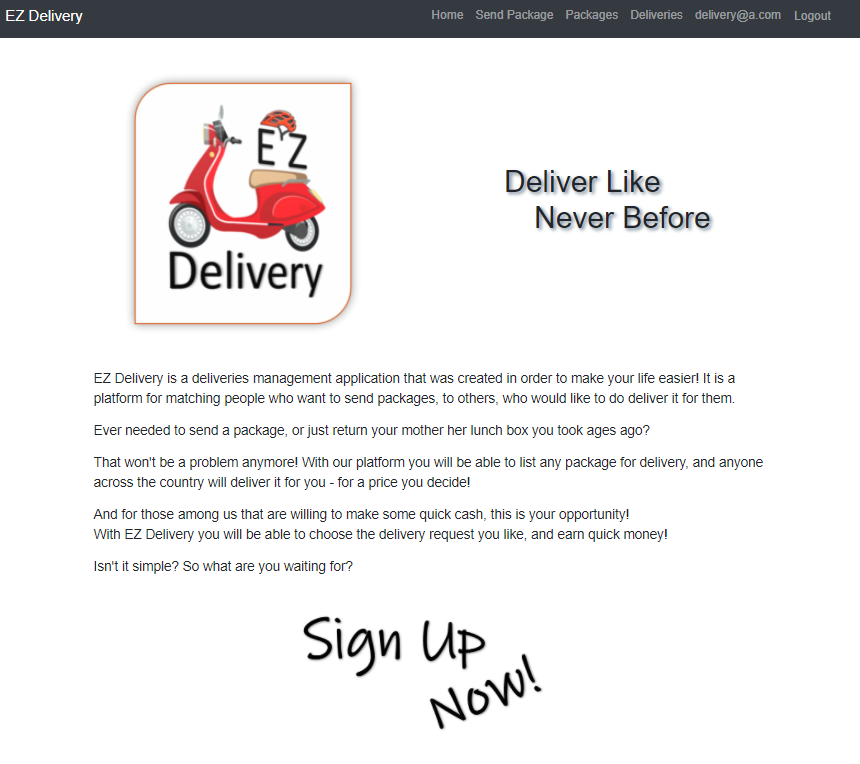
הפלטפורמה מאפשרת:

* הרשמה לאתר כמשתמש סטנדרטי / שליח.
* הגדרה והוספה של חבילה למאגר החבילות הפוטנציאליות למשלוח.
* הצעת מחיר לחבילה שנתונה במאגר החבילות מצד השליח.
* המשתמש יוכל לעקוב אחרי כלל החבילות שלו.
* צפייה בהיסטוריית החבילות עבור השליח שטיפל בהן.
* דירוג השליח בגמר התהליך.

**תצלומי מסך**

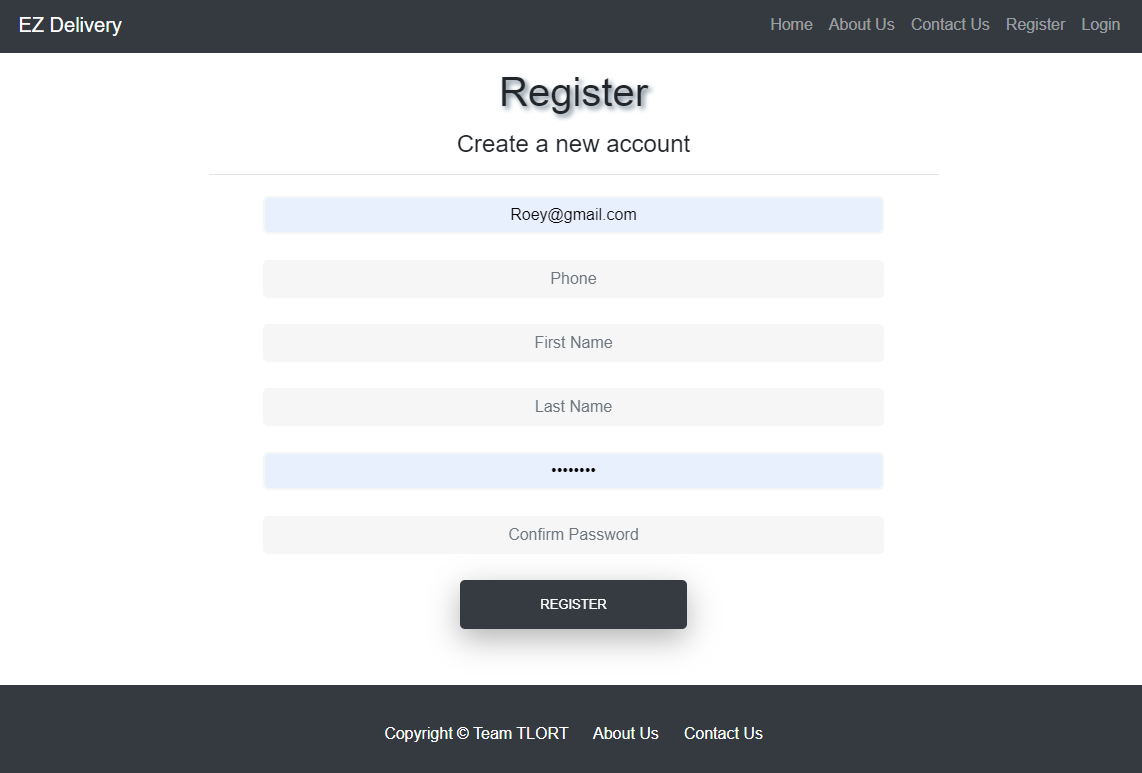
דף הבית – Home

* בדף הבית נוכל לראות את מהות הפלטפורמה החברתית ואת ה-"אני מאמין" של החברה.



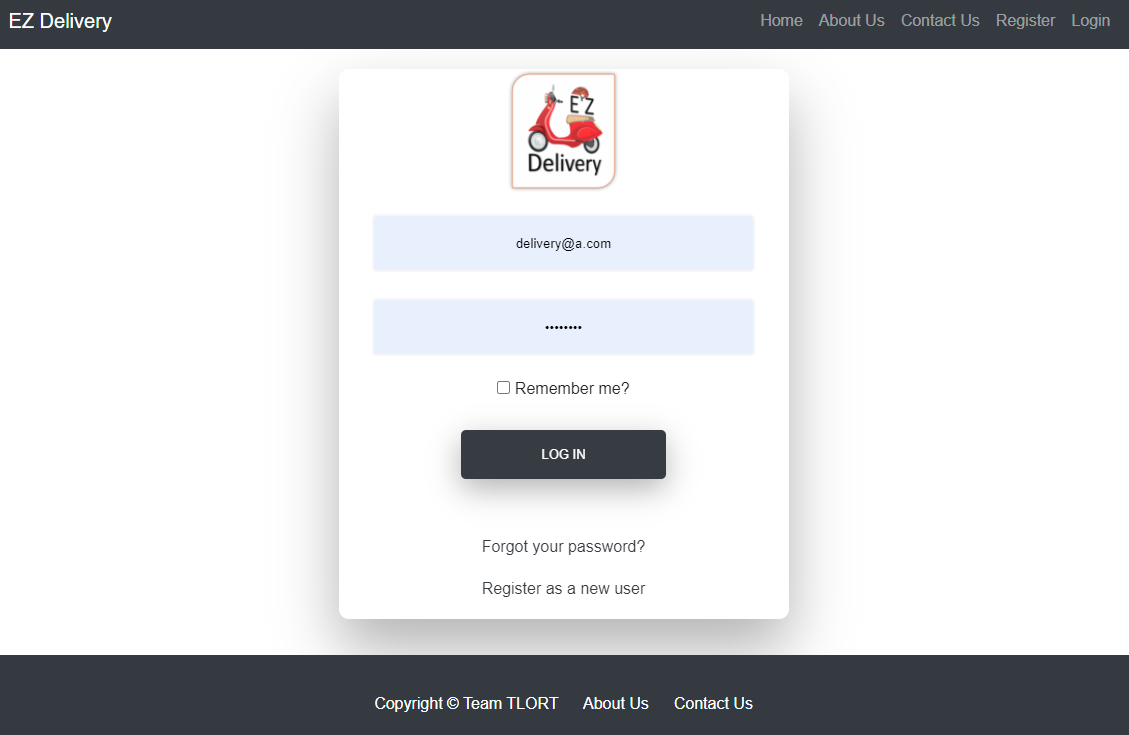
דף הרשמה - Register

* הרשמה עבור משתמשים חדשים.
* במידה והרישום הצליח, המשתמש יועבר לדף הבית של האתר.

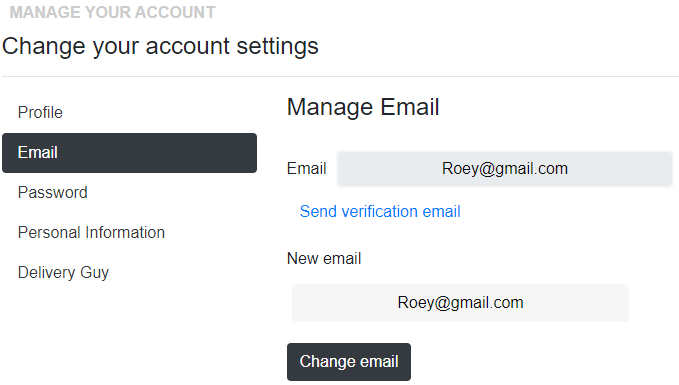
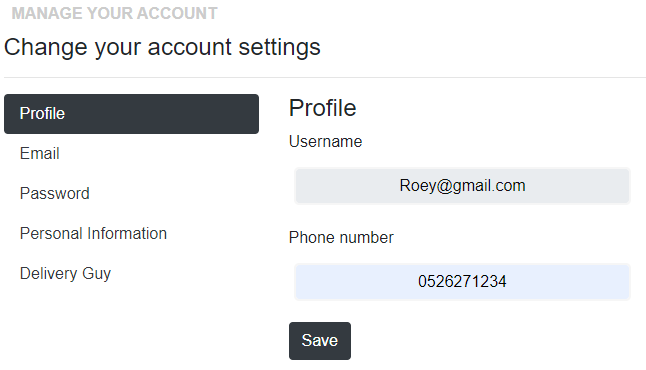


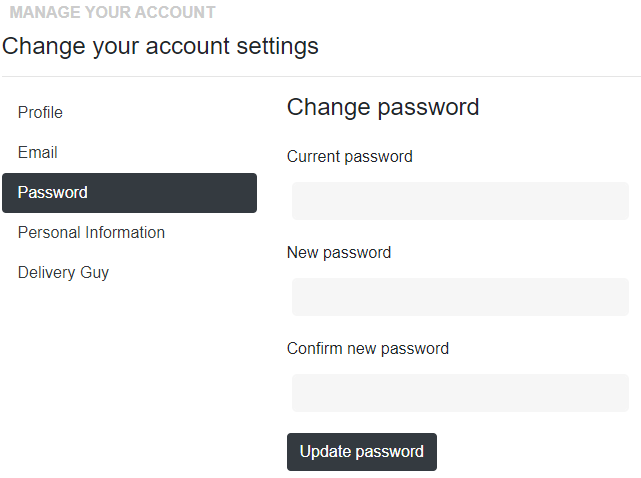
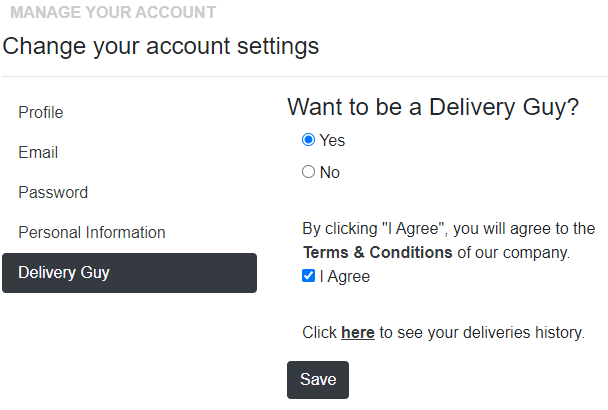
דף התחברות - Login

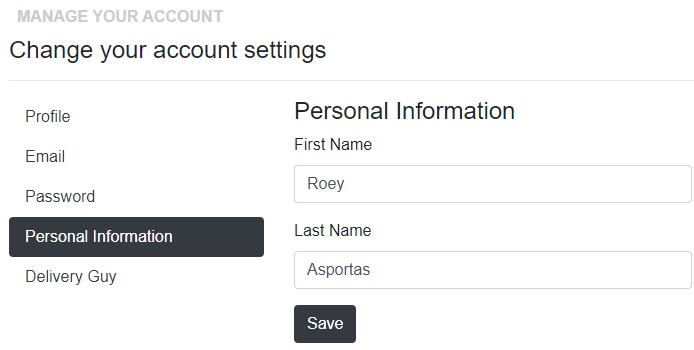
* התחברות לחשבון קיים. במידה ולא קיים חשבון, ניתן לגשת לדף ההרשמה.



דף הפרופיל - My Profile

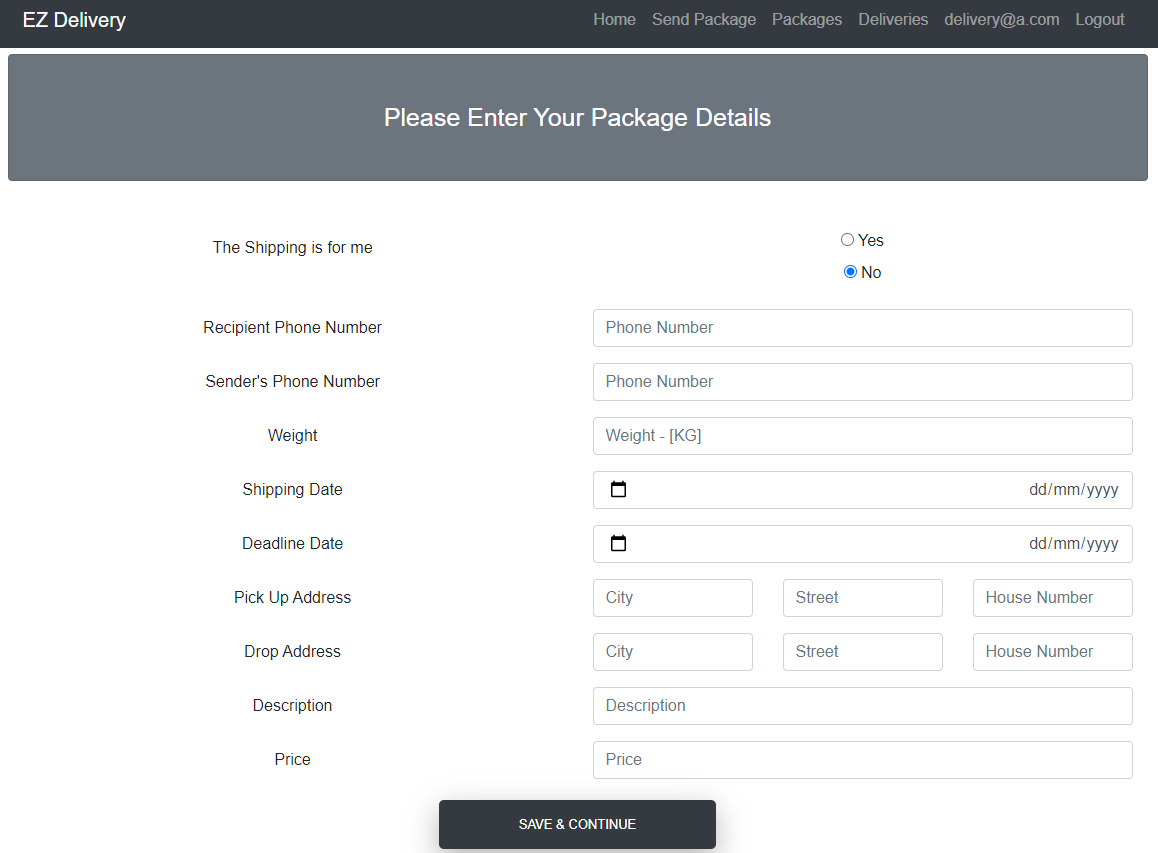
* בדף זה, ניתן לשנות את הפרטיים האישיים של המשתמש כמו: שם מלא, סיסמה, מספר טלפון ודואר אלקטרוני.
*  בנוסף, משתמש יכול להגדיר את עצמו כשליח במידה והוא מסכים למדיניות החברה עבור משתמש-שליח.



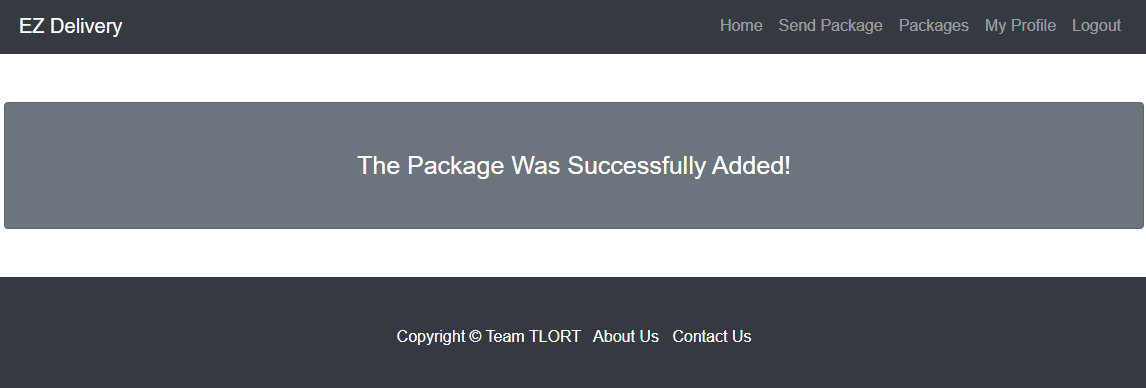


שליחת חבילה - Send Package

* הגדרה והוספה של חבילה חדשה למאגר החבילות העתידות למשלוח.
* בדף זה ניתן להגדיר אם המשלוח הוא עבור שולח החבילה או עבור אדם אחר.
* ניתן להגדיר ציר זמן עבור שליחת החבילה ושדות נוספים כמו: מחיר, כתובת לאיסוף ומשלוח, תיאור וטלפון.

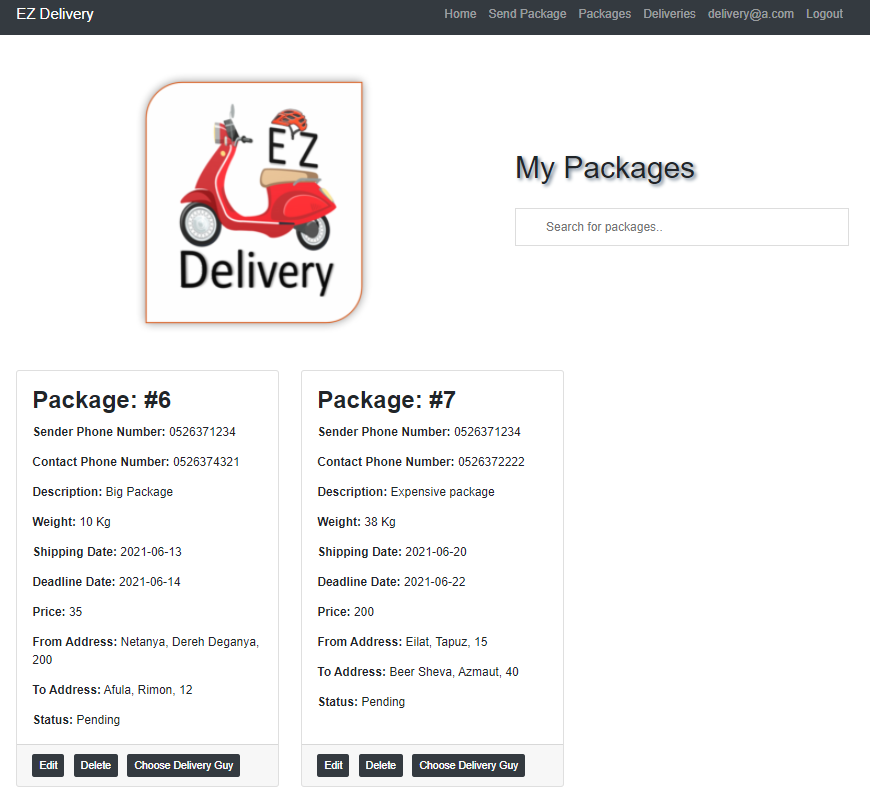


חיווי עבור חבילה שנוספה בהצלחה.



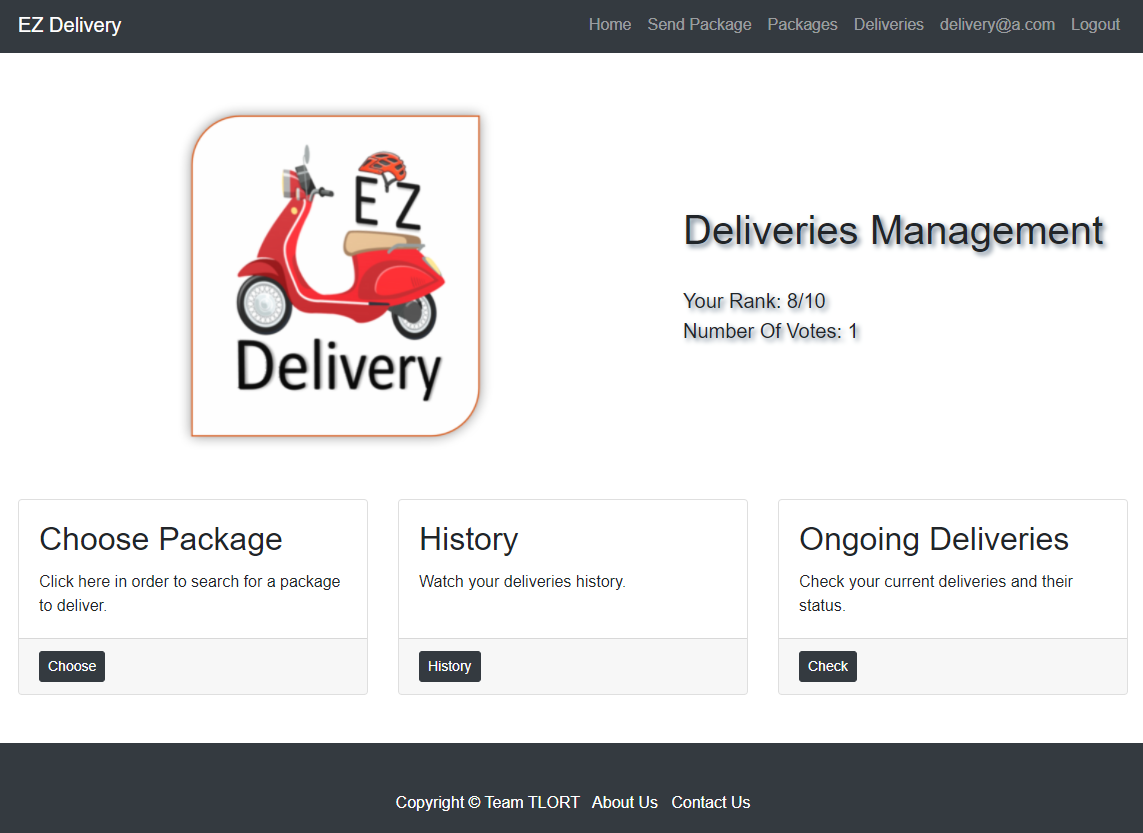
החבילות שלי - My Packages

* בדף זה ניתן לראות את כלל החבילות שהמשתמש הוסיף ואת סטטוס החבילות, לאחר בחירת השליח הסטטוס ישתנה בהתאם.
* קיימת פונקציית חיפוש עבור המשתמש אם ברצונו לחפש לפי ערים מסוימות, שליח ספציפי או כל פרמטר אחר.



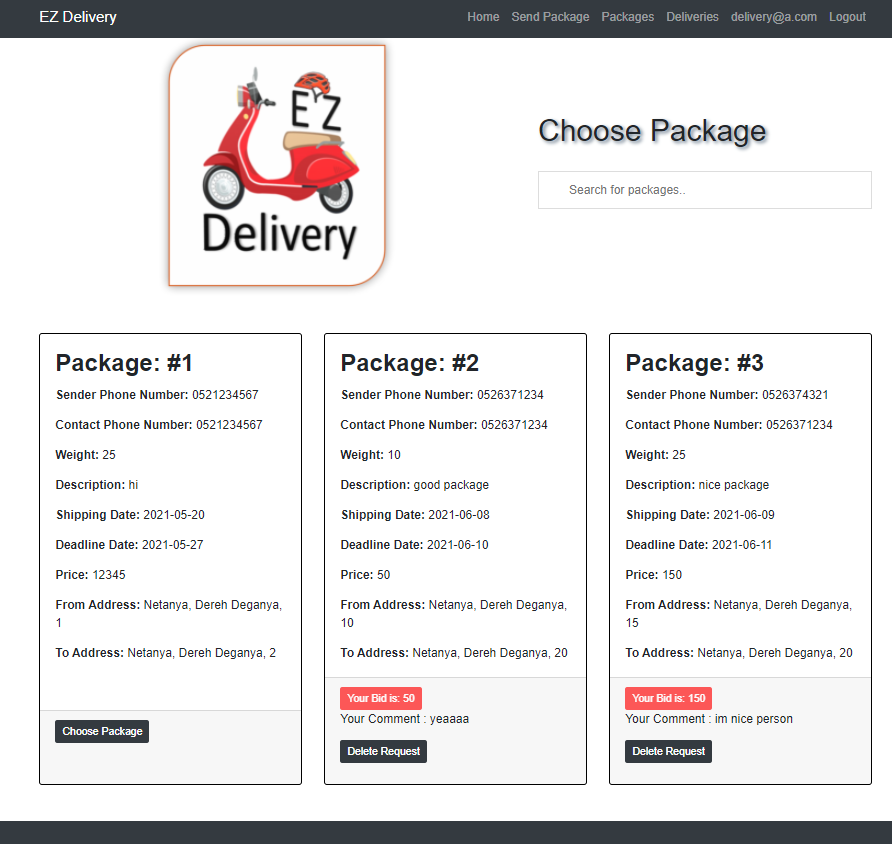
משלוחים - Deliveries

* בדף זה יש גישה לבחירת החבילות הזמינות, היסטוריית המשלוחים של השליח המחובר לאתר.
* ניתן לראות את דירוג השליח הממוצע אותו מבעלי החבילות שהשתמשו בשירותיו.



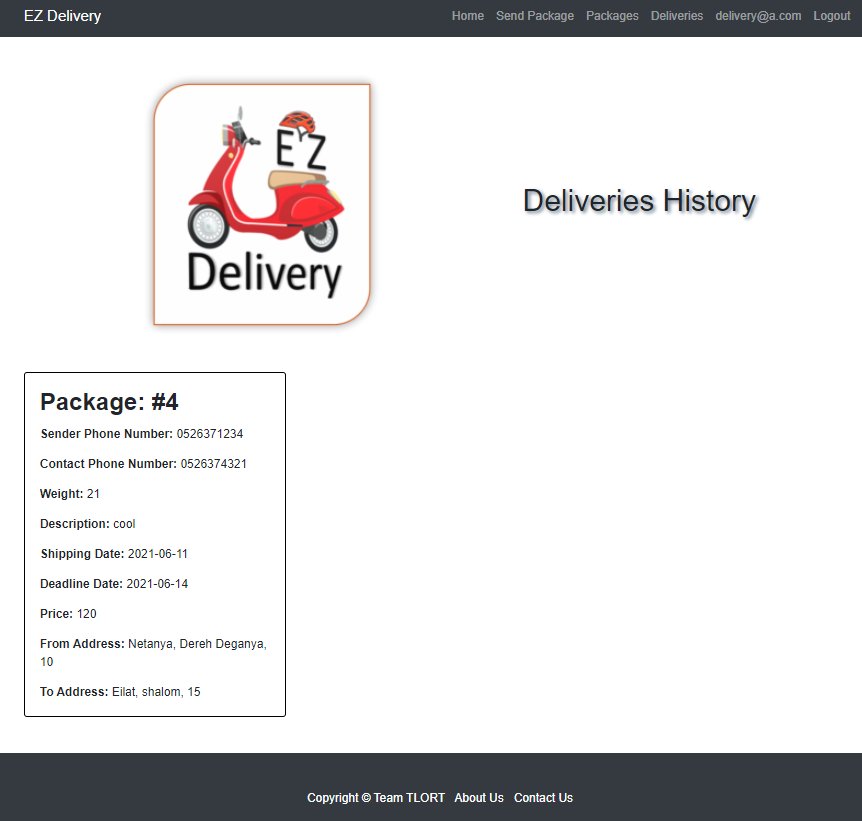
בחירת חבילה – Choose Package

* גישה לדף זה ניתן דרך דף המשלוחים.
* קיימת פונקציית חיפוש עבור השליח אם ברצונו לחפש לפי ערים מסוימות, שליח ספציפי או כל פרמטר אחר.
* לאחר בחירת החבילה בעמוד זה, השליח יוכל לתת הצעת מחיר לבעל החבילה ולהוסיף הערה.
* אם נדרש, יוכל השליח למחוק את הבקשה.



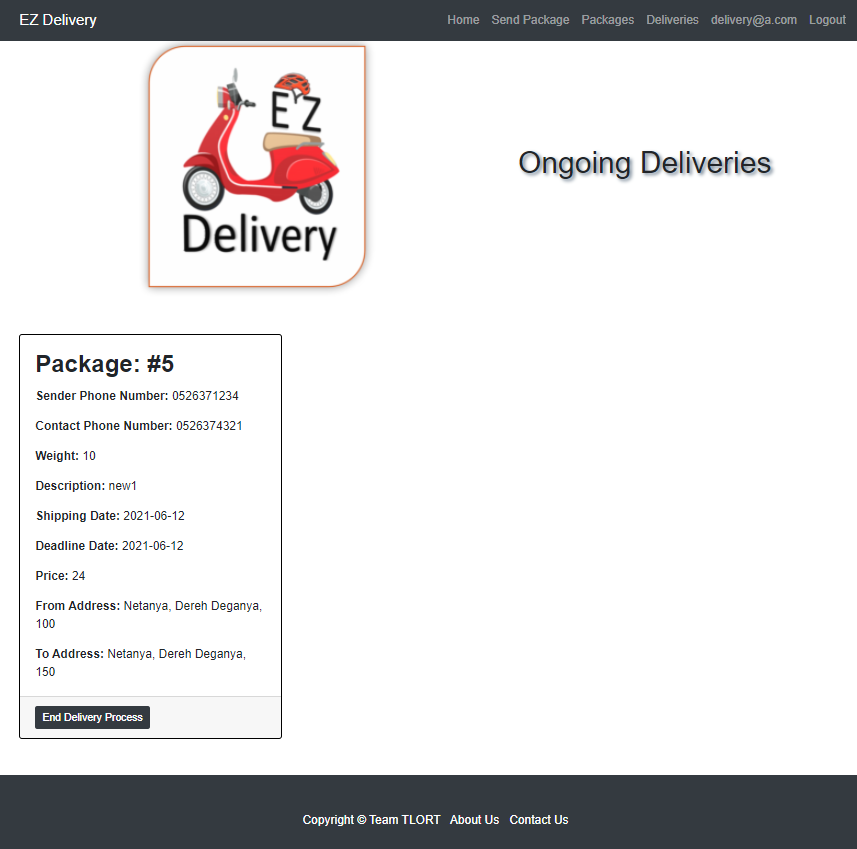
היסטורית משלוחים - Deliveries History

* גישה לדף זה ניתן דרך דף המשלוחים.
* ניתן לראות את ארכיון המשלוחים עבור אותו שליח.



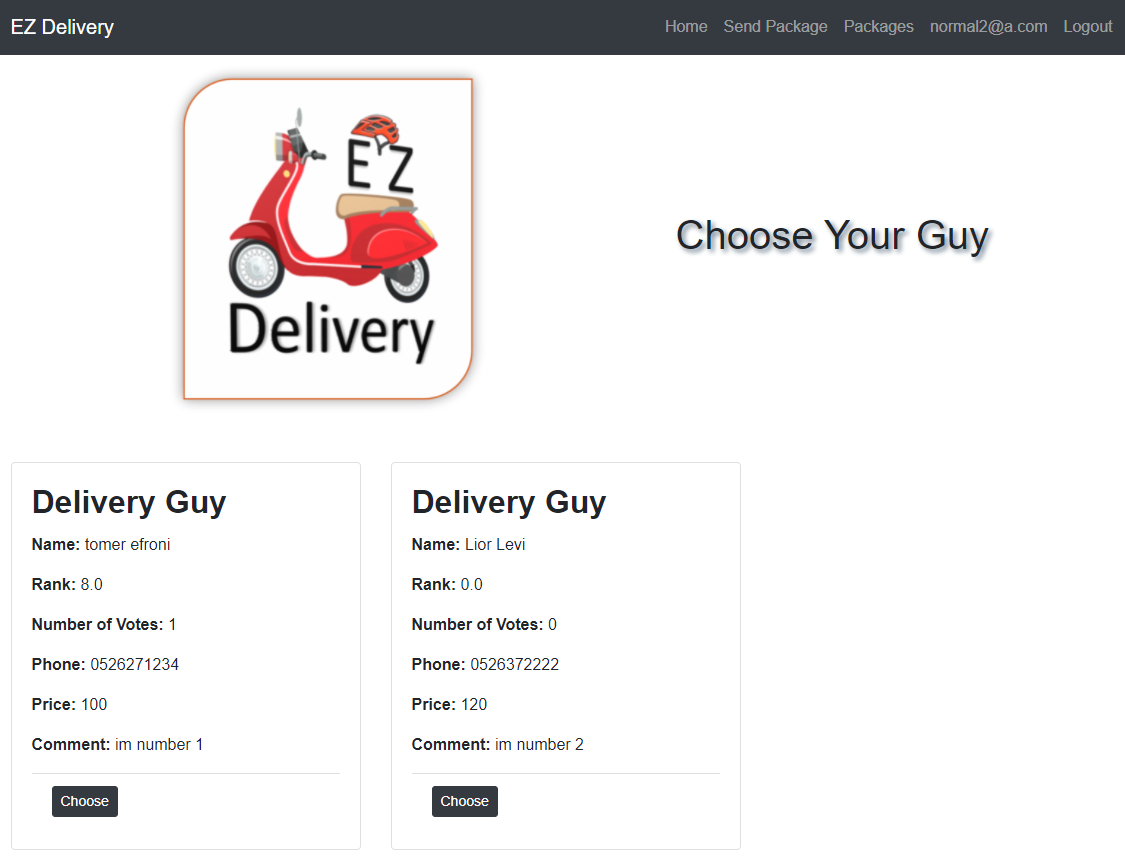
משלוחים בתהליך - Ongoing Deliveries

* ניתן לגשת לדף זה דרך דף המשלוחים.
* ניתן לעקוב אחר המשלוחים בזמן אמת.



בחר איש משלוח – Choose Your Guy

* ניתן לגשת לדף זה דרך דף "החבילה שלי".
* בעמוד זה ניתן לראות את כלל השליחים שהציעו הצעת מחיר עבור החבילה שנבחרה ולאחר מכן, לבחור שליח בהתאם לדירוג והצעת המחיר שלו.



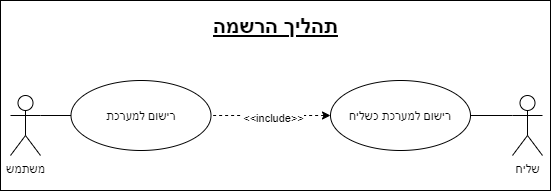
טבלת שחקנים

|  |  |
| --- | --- |
| **שחקן** | בעל החבילה |
| **תיאור** | אדם המוגדר כמשתמש סטנדרטי במערכת. |
| **מסלול עיקרי** | משתמש ממלא את פרטי החבילה אותה הוא רוצה לשלוח, מאשר את הצעת השליח ובגמר מדרג את השליח עבור המשלוח אותו ביצע. |
| **תנאי התחלה** | משתמש רושם את החבילה בהצלחה. כעת מחכה להצעת מחיר מהשליחים הזמינים. |
| **תנאי סוף** | המשלוח יאושר ויוכל לצאת לדרך לאחר שבעל החבילה אישור את הצעת המחיר מצד השליח. |

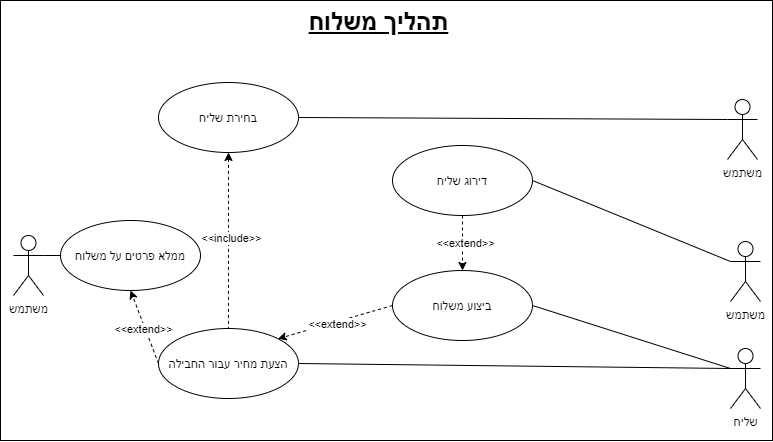
|  |  |
| --- | --- |
| **שחקן** | שליח |
| **תיאור** | אדם המוגדר כמשתמש-שליח במערכת. |
| **מסלול עיקרי** | מציע הצעת מחיר לבעלי החבילות הזמינים בשביל לבצע משלוח ומבצע את המשלוח עבור בעל החבילה. |
| **תנאי התחלה** | נותן הצעת מחיר לבעל החבילה ומחכה לאישור. |
| **תנאי סוף** | לאחר אישור הצעת המחיר מצד השליח, השליח יבצע את המשלוח. |

תרשים Use case –

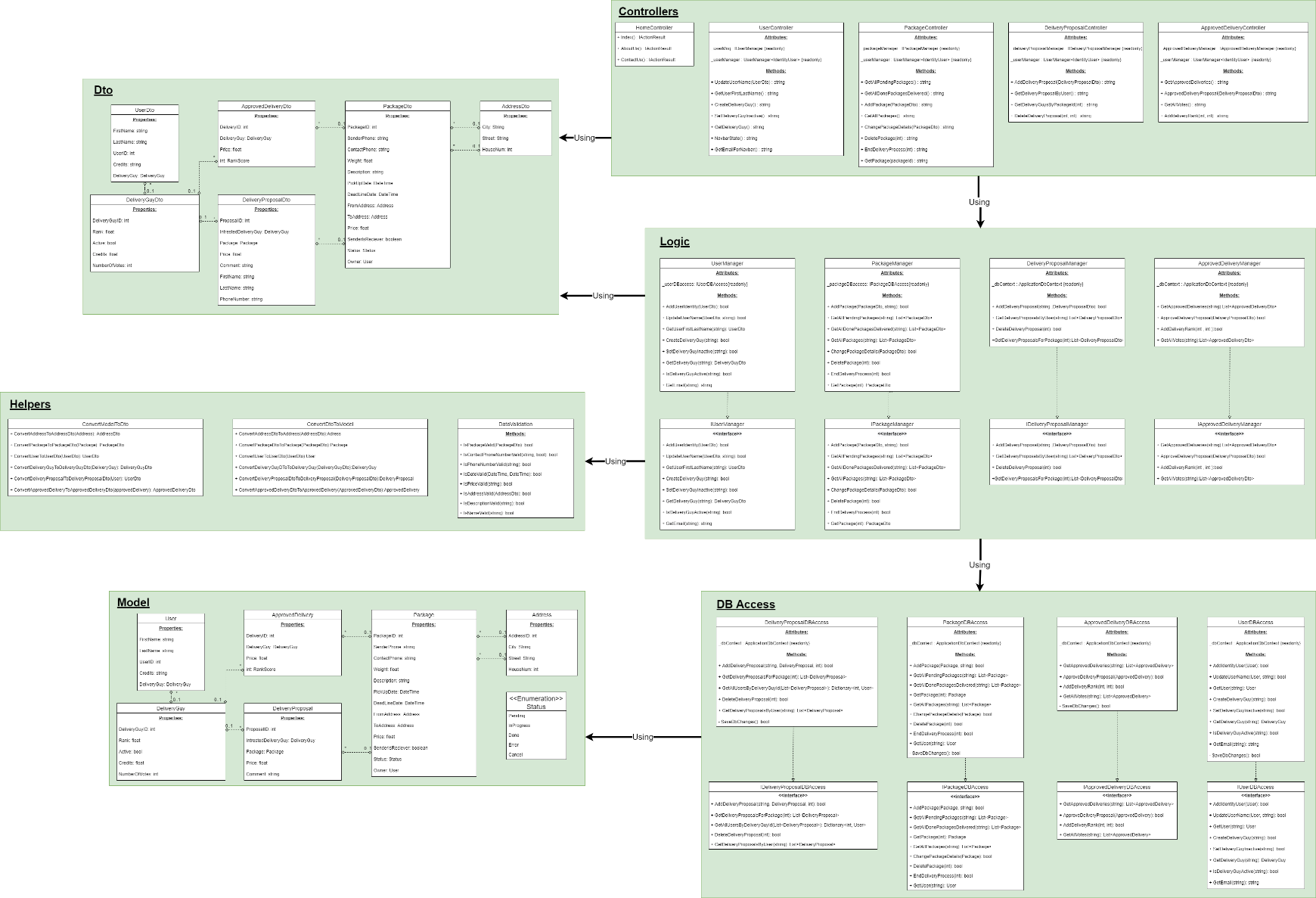
**הרשמה:**

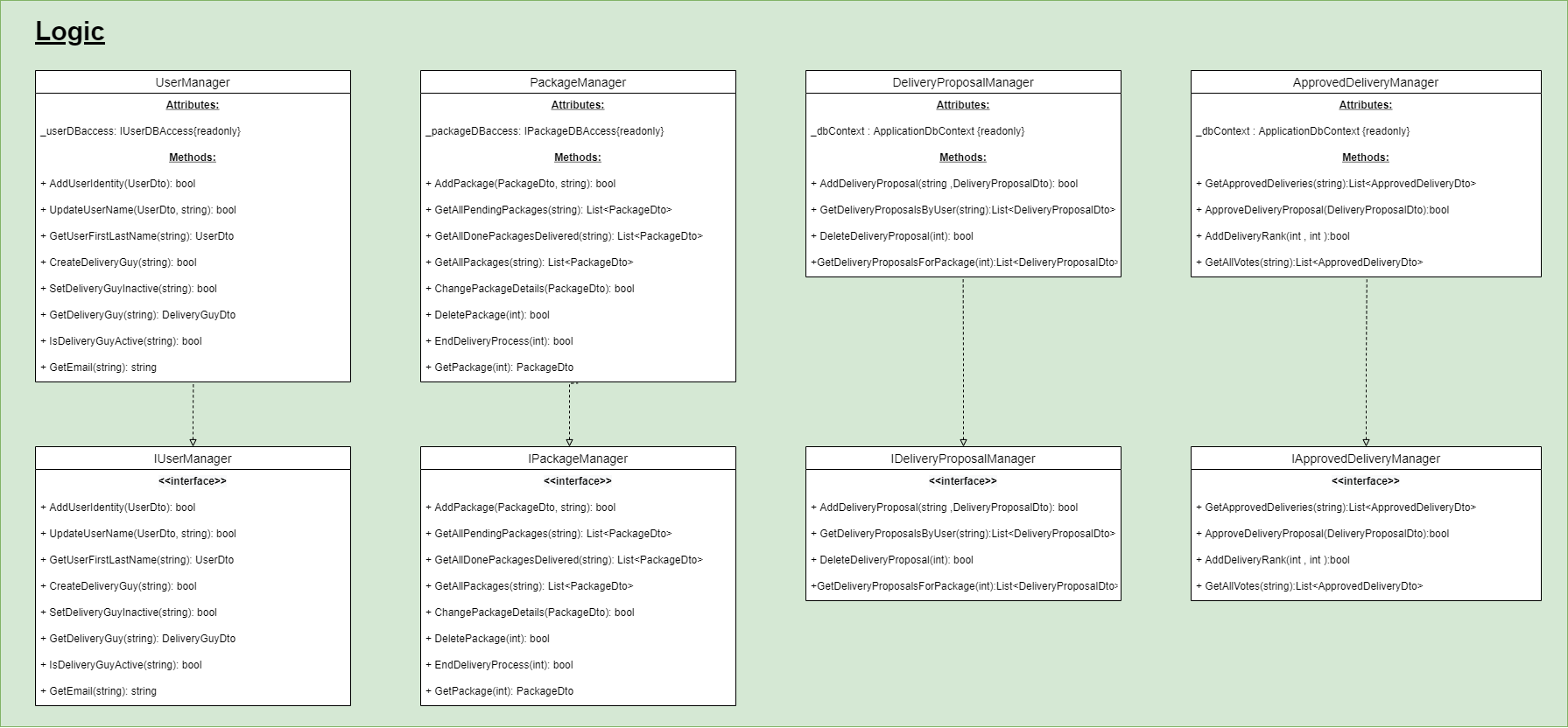
****

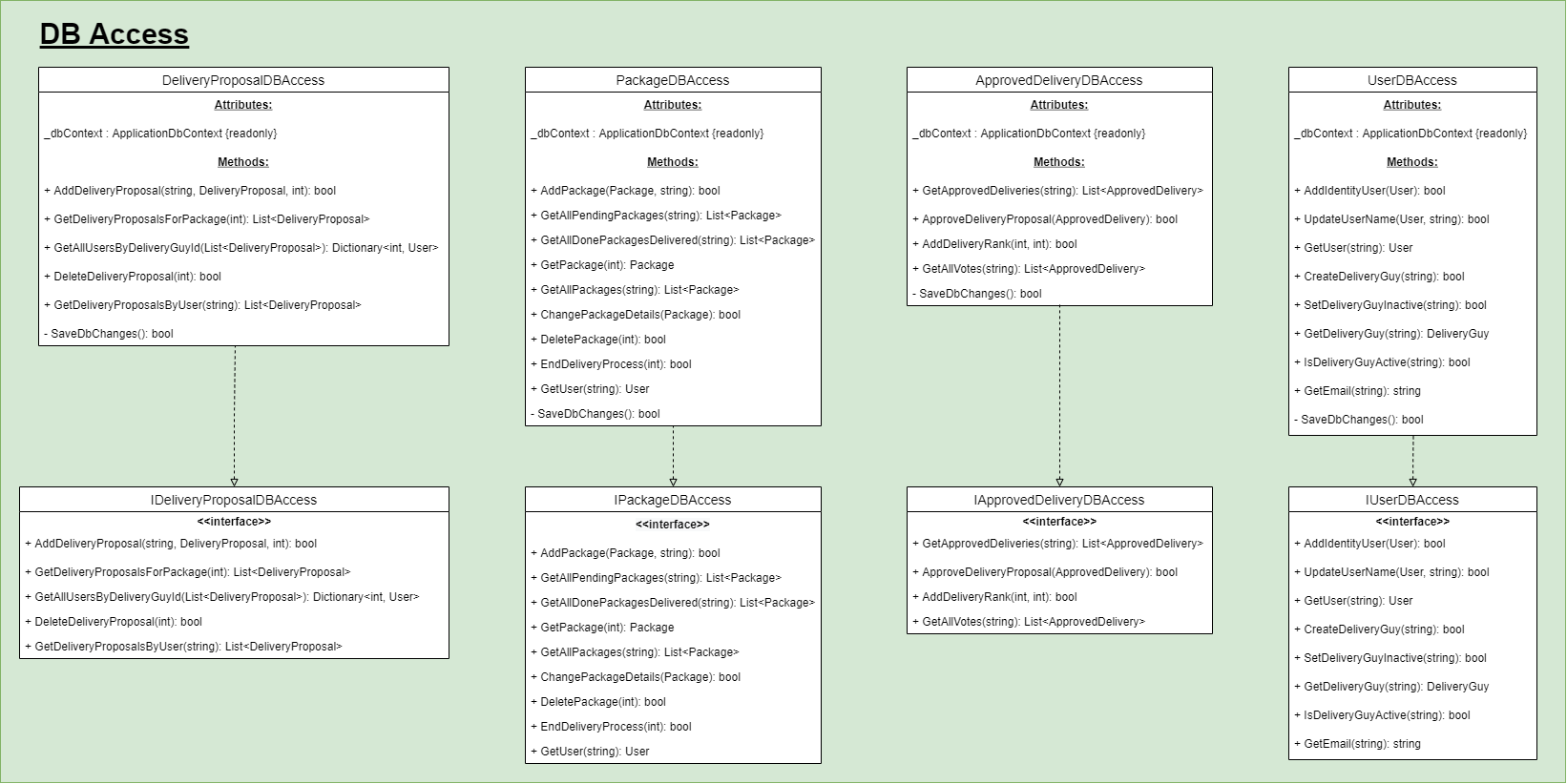
**משלוח:**

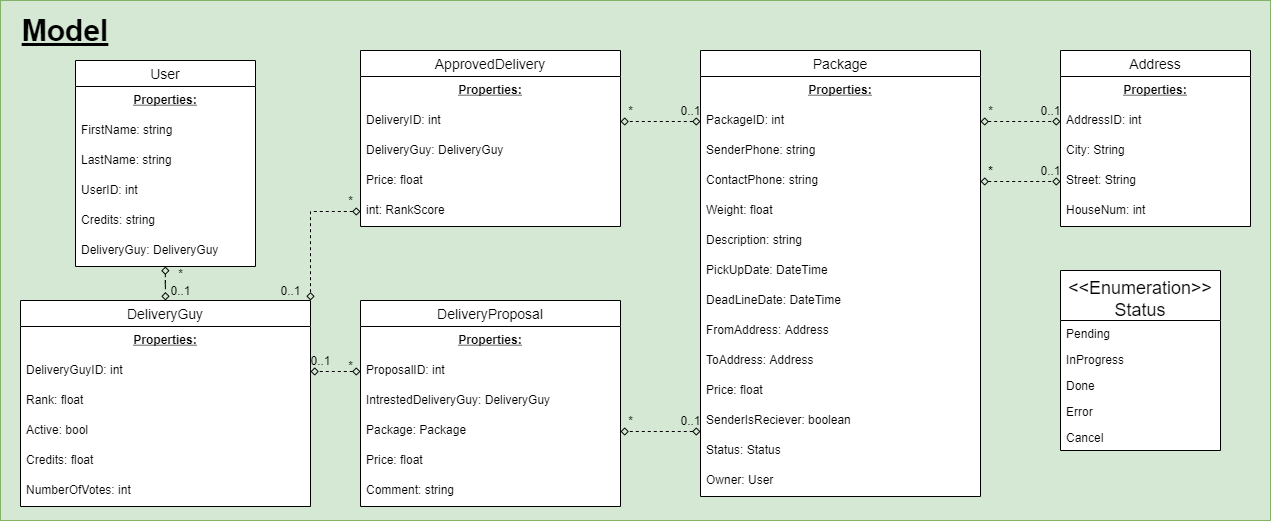


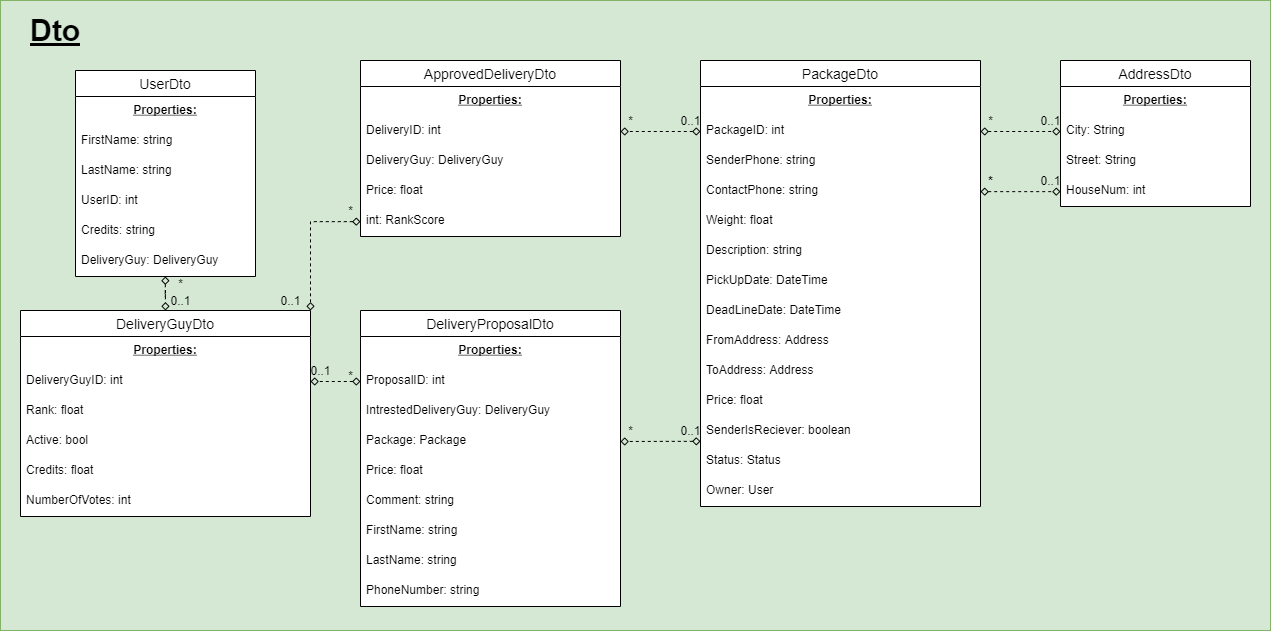
תרשים מחלקות - Class Diagram

מכיוון שכמות המחלקות במערכת שלנו היא גדולה, העדפנו לשרטט את ה- Class Diagram בצורה קצת שונה. שרטוט מלא מצורף בעמוד זה, ותצלום של כל רכיב בהמשך.

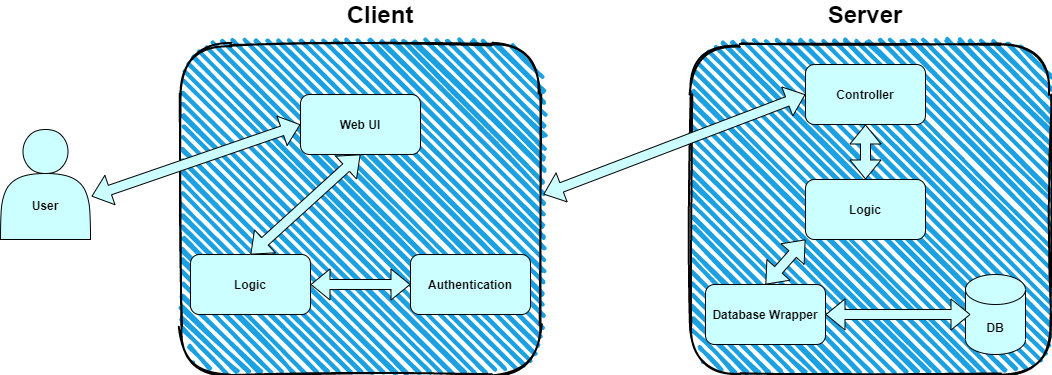










ארכיטקטורה פיזית

**User:**

* המשתמש אשר משתמש ב-Web Application שלנו.

**Client:**

* Web UI: פלטפורמת ה-Web דרכה המשתמש ניגש למערכת. רכיב זה משמש למעשה כשכבת האפליקציה. רכיב זה ייכתב ב-html, css, js תוך כדי שימוש בספריות Bootstrap & JQuery.
* Logic: אחראי על עיבוד הנתונים והבקשות המתקבלים בתקשורת ועיבודם במידת הצורך כך שיהיה ניתן להציג אותם למשתמש בפלטפורמת ה-Web.
* Authentication: שכבת אוטנתיקציה, מטרתה לוודא הרשמה והתחברות נכונה של משתמשים למערכת. ימומש באמצעות AspNetCore.Identity.זהומנגנון מובנה אשר נמצא ב-.Net 5שבעזרתו ניתן לממש את הפעולות הללו בצורה יעילה ופשוטה.

**Server:**

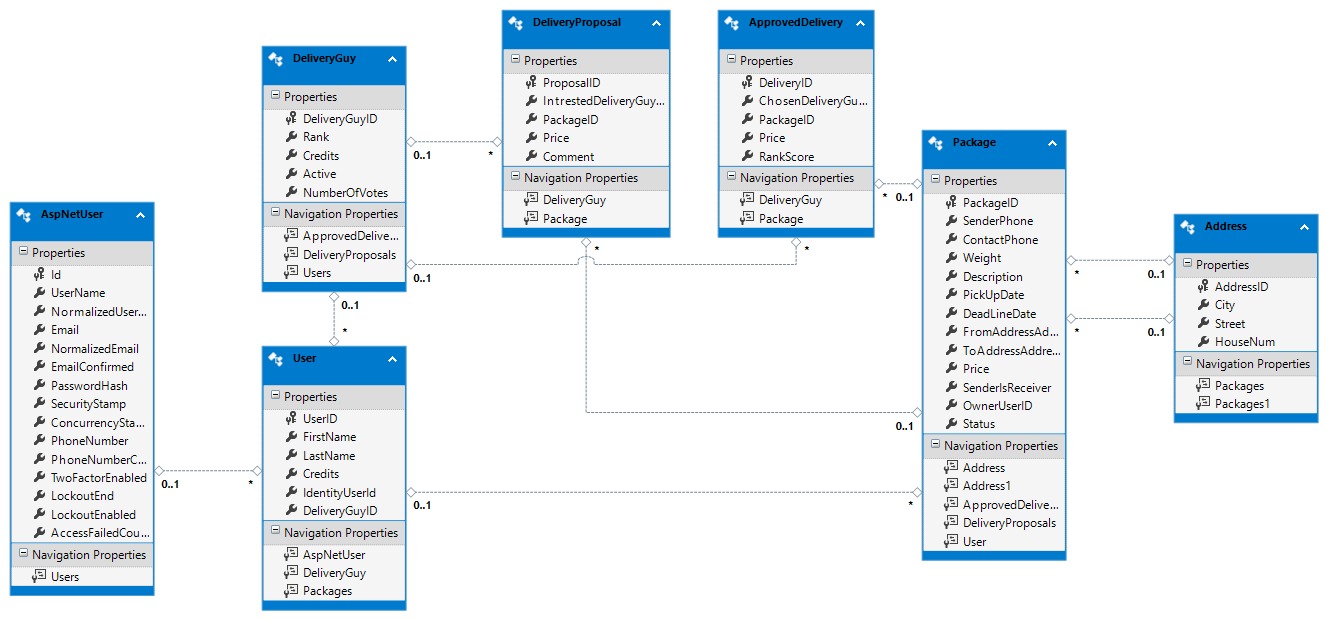
* Controller: רכיב המקשר בין צד הלקוח לצד השרת. מטרתו לטפל בבקשות המגיעות מהלקוח אל השרת ולנווט אותן בהתאם.
* Logic: אחרי על עיבוד הנתונים המתקבלים בצד השרת. תפקידו לבצע וריפיקציה למידע, המרת משתנים (בעיקר בין המודל של ה-db לישויות ה-Dto שלנו) והתקשרות לרכיב ה-Database Wrapper.
* Database Wrapper: רכיב בסיס הנתונים – אחראי לשלוף, להוסיף ולעדכן נתונים מול בסיס הנתונים שלנו. רכיבים אחרים (כמו רכיב הלוגיקה) מתקשרים מולו על מנת לעבוד באופן מסודר, בטוח ויעיל מול בסיס הנתונים של המערכת.
* DB: רכיב האחראי על שמירת הנתונים בנוגע למשתמשים במערכת. רכיב זה יבנה בעזרת Microsoft SQL Server, ואת הטבלאות בו ניצור באמצעות שימוש ב-EF Core בשיטת Code First.

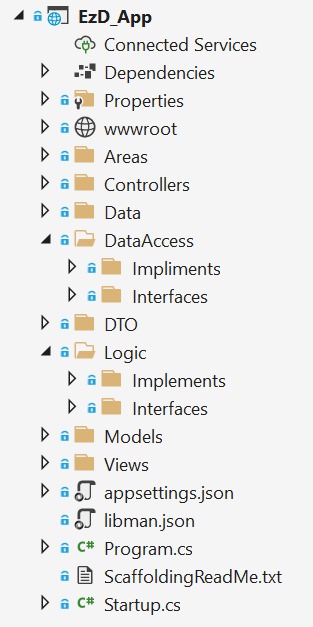
ארכיטקטורה לוגית

עבור מימוש המערכת שלנו השתמשנו ב-.Net 5 עבור צד השרת, וב-html, JS & Bootstrap עבור צד הלקוח, ויצרנו אותה לפי מודל שלושת השכבות:

* שכבת ה-UI שלנו, שהיא בעצם הדרך בה המשתמש מתקשר עם המערכת.
* שכבת הלוגיקה, יושבת בצד השרת ומבצעת פעולות ומניפולציות רלוונטיות.
* שכבת ה-DB / DB Wrapper, אחראית על שליפה/הוספה/עדכון נתונים עבור המערכת.

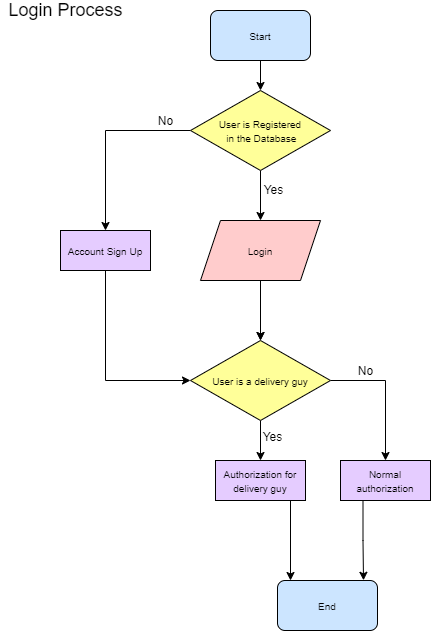
בנוסף, עבור בניית מסד הנתונים, השתמשנו ב-EF Core, בשיטת Code First, על מנת לייצר את מסד הנתונים עבור SQL Server. לשם כך יצרנו מודלים רלוונטיים אשר יהוו טבלאות במסד הנתונים לאחר יצירתו.

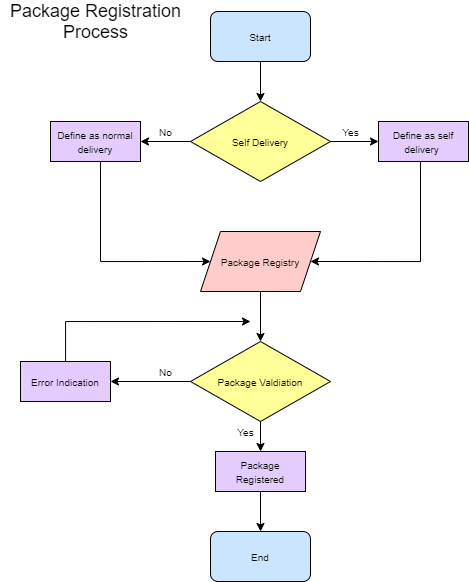
להלן מבנה מסד נתונים שלנו (ERD):

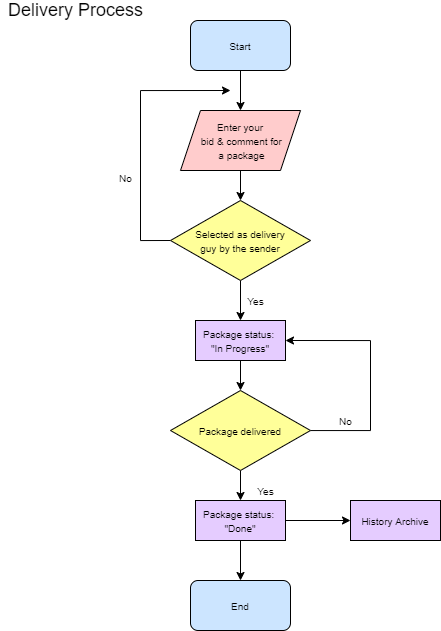
עבור קליטת/העברת נתונים מצד הלקוח/השרת יצרנו מחלקות Dto רלוונטיות. הישויות הללו משמשות אותנו עבור העברה או קבלת מידע מצד הלקוח אל צד השרת (ולהיפך). בנוסף, יצרנו מחלקות סטטיות אשר תפקידן לבצע המרות נתונים – Dto 🡪 Model או Dto 🡨 Model.

את כלל המחלקות במערכת שלנו עטפנו ב-Interfaces שונים שמטרתם לשמור על מבנה אחיד ומימוש יעיל של הפונקציונליות שלה, וזאת במטרה לנסות לקיים את עקרונות ה-SOLID השונים.

תהליכים מרכזיים







בדיקות

אנו מבינים את החשיבות והיעילות אשר קיימים בביצוע Unit Tests, אך עקב חוסר זמן העדפנו שלא לממש אותן במערכת שלנו (מבחינת עיצוב המערכת, היא אכן בנויה בצורה כזו כך שיהיה ניתן ליצור Unit Tests בצורה יעילה).

עבור בדיקות המערכת, ביצענו בדיקות ידניות הבודקות את הפונקציונליות שלה תוך כדי התחשבות במקרי הקצה השונים.

דאגנו לבדוק ולוודא שזרימת המערכת אכן תקינה ושהמידע נשמר/נשלף בצורה נכונה ב/ממסד הנתונים.

תהליך העבודה

בתחילת העבודה על הפרויקט בחרנו את הרעיון בזמן די קצר ובהסכמה משותפת של חברי הצוות.

בחירת הטכנולוגיות עבור הפרויקט לקחה זמן, ובהתחלה לא ידענו איזה טכנולוגיה כדאי להתחיל ללמוד ובעזרתה לכתוב את הפרויקט. לאחר דיון מעמיק עם תמיר לגבי הטכנולוגיות שכדאי לעבוד איתם בפרויקט כמו שלנו בחרנו לכתוב את EZ Delivery ב- #C (בסביבת העבודה .NET 5). לאחר כשבוע מבחירת נושא הפרויקט, התחלנו ללמוד את הטכנולוגיות ומה הצורה הנכונה לבניית הפרויקט שגם אם יהיו שינויים בעתיד, נבנה את הפלטפורמה בצורה שהיא מורכבת מחלקים שאינם תלויים זה בזה.

השלב השני בעבודה היה ללמוד לעבוד כצוות של חמישה אנשים. גם חלק זה היה מאתגר, גם מבחינת כמות האנשים שצריך ליישר קו כל פעם ולהישאר מעודכנים לגבי שינויים בפרויקט וגם מבחינת חלוקת העבודה. לכן, התשתמשו ב – Azure DevOps בשביל להיצמד למתודולוגית העבודה Agile, שם הקצנו משימות ((Stories ושיתפנו את חלקי הקוד שכל אחד עשה בזמנו לתוכנית אחת גנרית שעובדת על כל חלקיה.

לאחר מכן במהלך הדרך הקפדנו להיפגש בכל שבוע לפחות פעם אחת (בנוסף למפגש השבועי) בשביל להתעדכן ולחלק משימות.

התחלנו בתרשימי מחלקות, UseCase ופתחנו קובץ של Google Docs שכולם יכולים לערוך אותו על מנת להגדיר את אבני הבניין של הפרויקט.

בשלב השני חילקנו את העבודה ביננו, כל אחד היה אחראי על מס' תהליכים במערכת ולאחר שכולם סיימו, העלנו את קטעי הקוד ל – Azure DevOps וכל אחד לקח את התוכנית השלמה ובדק שהיא רצה באופן תקין לאחר חיבור קטעי הקוד.

סיכום

לאחר דיונים מרובים וחשיבה משותפת על רעיונות עבור פרויקט זה, הגענו להסכמה שאנחנו רוצים לבנות פלטפורמת משלוחים חברתית שתיתן מענה לכלל האוכלוסייה עבור בעיית המשלוחים ולאו דווקא עבור חבילה ספציפית אלא עבור מגוון רב של חבילות.

מצד אחד, לבנות את הפלטפורמה הזו שתיתן מענה למשלוחים שונים, ומצד שני השליחים יקבלו השלמת הכנסה עבור המשלוחים אותם ביצעו.

בפרויקט זה עבדנו לפי מתודולוגית Agile, בסביבת עבודה .NET 5 וכתבנו ב – C#.

את מסד הנתונים שלנו בנינו בשילוב של EF-Core (code first) ו- Microsoft SQL Server.

נעזרנו בשירות הענן של Azure DevOps על מנת לעבוד כצוות אחד באופן היעיל והמתואם ביותר, גם מבחינת כתיבת קוד התוכנית וגם מבחינה של תכנון נכון ומודולרי של הפרויקט עצמו.

למדנו בפרויקט זה המון על עבודת צוות, עבודה וארכיטקטורה המקבילה לתעשייה הנוכחית. פרויקט שבנוי בצורה נכונה, מורכב מכמה שכבות וחלקים המחוברים אך גם מופרדים בצורה גנרית שיוכלו לקבל שינויים בצורה טבעית שלא משפיעה על שאר מרכיבי הפרויקט ועל המערכת.

במהלך הפרויקט נפגשנו פעמים רבות על מנת לדון בהמשך בניית הפרויקט וחילוק משימות נכון עבור כל חברי הקבוצה. כמו בכל פרויקט משותף היו וויכוחים לגבי הטכנולוגיות שאיתן נעבוד, פיצ'רים שונים בפלטפורמת המשלוחים ועוד.. אך לבסוף הצלחנו להגיע להסכמה משותפת עם כלל חברי הקבוצה ולמדנו מטעויות שעשינו במהלך הדרך.