*ינואר 2019*

מיני פרויקט תשעט- ניהול מבחני נהיגה

הערה: המסמך אינו מפרט את דרישות הבסיס של הפרויקט אלא רק את התוספות הלא טריוויאליות.

הפרויקט הינו בסדר גודל של כ 10000 שורות, על מנת לראות את מלוא היכולת אנו ממליצים להתנסות בשימוש בתוכנה!

**מבנה כללי**

התוכנה בנויה עבור שלושה סוגים של משתמשים:

1. תלמידים הנבחנים לקראת קבלת רישיון נהיגה

2. בוחנים

3. מנהל מערכת (administrator)

בנוסף, ישנו מסך log-in שמזהה את המשתמש לפי ת.ז ומעביר אותו למסך המתאים (טסטר /נבחן /מנהל)

**פירוט המסכים**

***באפשרות התלמיד:***

* לצפות בטסטים שעליו לעבור (לכל סוגי הרישיונות)
* לצפות בסטטוס הטסטים שעבר (כולל מרכיבי הציון)
* לקבוע לעצמו טסט חדש

***באפשרות הטסטר:***

* *לצפות בטסטים שעליו לבצע*
* *לצפות בטסטים שבוצעו כבר על ידו*
* *לעדכן את מצב הטסט (עבר/לא עבר)*

***באפשרות המנהל:***

* לצפות, להוסיף, לעדכן ולמחוק טסטרים
* לצפות, להוסיף, לעדכן ולמחוק תלמידים
* לצפות, להוסיף, לעדכן ולמחוק את כל הטסטים שבוצעו/ לא בוצעו , עברו/ לא עברו
* לייצא את רשימת התלמידים לקובץ EXCEL
* לייצא את רשימת הטסטרים לקובץ EXCEL
* לייצא את רשימת הטסטים לקובץ EXCEL
* לשלוח בקליק תזכורת בדוא"ל לכל התלמידים שקבעו לאותו היום
* לשנות הגדרות מערכת (פירוט בהמשך\*)
* לחפש באופן חופשי בכל השדות
* לחפש לפי שם פרטי, משפחה ותעודת זהות
* לסנן לפי סוגי רישיון שהמורה מתמחה
* לסנן לפי סוגי רישיון שהתלמיד לומד
* לסנן טסטים לפי עבר/ לא עבר, לפי סוג רשיון, ולפי טווח תאריכים

**\*פירוט מסך ההגדרות**

באפשרות המנהל לשנות את ההגדרות הבאות:

* מספר שיעורים מינימלי שעל התלמיד לבצע לפני שניגש לטסט
* גיל מינימלי לטסטר
* גיל מינימלי לתלמיד
* זמן מינימלי בין שני טסטים לאותו תלמיד
* אחוז של מרכיבי הציון שעל התלמיד להצליח במהלך הטסט על מנת לעבור אותו
* הגדרות תצוגה של המערכת
* שם משתמש וסיסמת מנהל
* מהם מרכיבי הציון לבדיקת התלמיד בטסט

Login - MainWindow

Administrator

TraineeWin

TesterWin

ShowTest

UpdateTest

Settings

ShowTestResults

EditTest

AddTrainee

AddTester

EditTest

**דיאגרמת החלונות**

**בדיקות מערכת**

על מנת לשמור על יציבות ואמינות המערכת נעשות בדיקות רבות באופן אוטומטי (בעיקר במחלקה BlImpl בפרויקט BL)

**אפשרויות מערכת**

* המערכת שולחת מייל לתלמיד עם סיום הטסט
* **במידה ועבר מצורף במייל הרישיון הזמני שלו (בקובץ PDF)**
* במידה ולא עבר נשלח דוא"ל עם הודעה מתאימה
* באפשרות המערכת להראות את מסלול הטסט באתר Google maps
* מחלקה שקובעת מסלול לטסט (פירוט בהמשך\*\*)
* המערכת שומרת את המידע שיש בקובצי הXML ב"cache" ובכך נחסכות גישות לקובצי הXML, מה שמשפר את היעילות במידה ניכרת
* שימוש בfactory , singleton , clone במידת הצורך
* על מנת לשלוח לכל תלמיד את קובץ הPDF של הרישיון הזמני שלו המערכת מנהלת את הקבצים בצורה שתתאים לכל מחשב.

**Routes Class\*\***

מחלקה סטטית שמכילה את כל הפונקציות שמטפלות במסלולים ומרחקים. המחלקה משתמשת בAPI של Google Maps [להרחבה לגבי הAPI](https://developers.google.com/places/web-service/intro)

**GetDistanceGoogleMapsApi**

הפונקציה מקבלת שתי כתובות ומחשבת את המרחק ביניהם. הפונקציה מחזירה את המרחק במטרים.

הפונקציה פונה ל- Google Maps. אם היא מקבלת תשובה אז היא מחזירה את המרחק מיד. אם יש חריגה היא מחכה 2 שניות וקוראת לעצמה בצורה רקורסיבית עוד פעם. אם נזרק עוד פעם חריגה מאתר גוגל ,אז היא זורקת חריגה ולא מנסה עוד פעם על מנת שלא להטריד את השרת ,ובמיוחד שככל הנראה פשוט המחשב לא מחובר לרשת.

בפרויקט משתמשים בפונקציה בשביל לקבל רשימה של טסטרים מסודרים בסדר עולה לפי מרחק מהתלמיד.

**GetAddressSuggestionsGoogle**

הפונקציה מקבלת string שאמור להיות חלק מכתובת. המחלקה מחזירה מערך של הצעות של כתובות אפשריות. הפונקציה מקבלת משתנה token שמשמש את גוגל לדעת האם מדובר בחיפוש אחד או שתי חיפושים. [להרחבה לגבי הAPI](https://developers.google.com/places/web-service/autocomplete).

בפרויקט משתמשים בפונקציה בשביל הUser Control AddressPicker.

**SetRouteAndAddressToTest**

הפונקציה היא פונקציית הרחבה לטסט. לכל טסט יש שדה שאפשר לשמור בו URL של מסלול שאפשר להציג על המפה. כאשר קוראים לפונקציה ,היא תחפש מיקומים רנדומליים באזור הטסט. הפונקציה תיקח כמה מהם ותנסה לבנות מסלול באורך בין 20 דקות ל- 60 (מוגדר בהגדרות). אם הפונקציה תצליח אז היא תכניס את המערך לטסט. במקרה שהיא לא תמצא מסלול מתאים היא תזרוק חריגה.

ישנם 2 סוגי חריגות :

1. בעיה עם החיבור לרשת.
2. בעיה עם הכתובת.

הפונקציה משתמשת בפונקציות הבאות:

1. GetLocationsInRadius פונקציה שמקבלת מיקום גאוגרפי ורדיוס ומחזירה מיקום בשטח הנתון.
2. GetLocationLatLog פונקציה שמקבלת כתובת ומחזירה מיקום גאוגרפי.
3. GetRouteDuration פונקציה שמקבלת מערך של כתובות ובודק מה יהיה זמן המסלול ביניהם.
4. GetGoogleUrl פונקציה שמקבלת מערך של כתובות ומחזירה URL שמציג את המסלול על מפה.
5. DownloadDataIntoXml פונקציה שמקבלת URL ומחזירה XML של התשובה. אם היתה בעיה היא תזרוק חריגה.

**ShowUrlInChromeWindow**

הפונקציה מקבלת URL ופותחת חלון של chrome שמציג את המסלול באתר Google Maps.