****

**המחלקה להנדסת תוכנה**

שם הפרויקט: אוטו-דסקטופ

Project Name: AutoDesktop

הגדרת הפרויקט – Statement Of Work

|  |  |
| --- | --- |
| שם הסטודנט: | יקיר גלעדי |
| שם המנחה: | בוריס מורוז |
| חתימת המנחה: | \_\_ בוריס מורוז \_\_\_ |
| תאריך ההגשה: | 19.6.2017 |

תוכן עניינים:

**מבוא3**

**מטרת הפרויקט4**

סקירת ספרות ראשונית5-6

**סקר שוק ראשוני7-9**

**מסמך דרישות ראשוני של המערכת.10-11**

תרשים בלוקים/מודולים כללי של המערכת12

ניתוח פונקציונאלי ראשוני13-15

הצגה ראשונית של חלופות מערכתיות. 16

האמצעים/הכלים הנדרשים16

תוצרי הפרויקט 16

לוח זמנים ראשוני לפרויקט17

רשימת מקורות17

**מבוא:**

אירגונים בעלי מוצר אשר יש לו UI (ממשק משתמש), צריכות לבדוק את הפונקציונליות של ה UI.

לכן, ישנם עובדים באירגונים אלה שנקראים "בודקים ידניים" אשר הם עושים בדיקות ידניות של ה UI שהן בדיקות תוכנה הנעשות על ידי עובד שהוכשר לכך, בודקים אלה מקבלים הנחיות לבדיקת הפונקציונליות של ה UI.

הנחיות אלה הן תרחישי משתמש אשר משתמש בתוכנה עצמה, במידה והבודקים מוצאים בעיה בפונקציונליות הם פותחים דו"ח על כך וצוות הפיתוח של ה UI מתקן את הבעיה.

AutoDesktop היא תוכנה אשר תחסוך זמן וכוח אדם באירגונים אלה,

היא תהפוך את הבדיקות הידניות לבדיקות אוטומטיות (קוד תוכנה המבצע סדרת פעולות, ברצף מתוכנן, ללא מגע יד אדם).

AutoDesktop היא ספריית פייטון או/ו תוכנה ידידות אשר תהיה ברורה למפתחי תוכנה וגם לאנשים בלי רקע בתחום.

בעזרתה יהיה אפשר ליצור תרחישי UI באופן אוטומטי המדמים את פעולותיו של המשתמש ובכך AutoDesktop תסייע לבנות סקריפטים[[1]](#endnote-1) של תרחישי UI לפעולות שחוזרות על עצמן ובכך לחסוך זמן וכח אדם.

בעזרת AutoDesktop יהיה אפשר לבדוק את הפונקציונליות של הUI של התוכנות אשר יריצו עליהן את התחרישים.

AutoDesktop תהיה עמידה לשינויי UI ללא שינוי רב של קוד התרחיש.

**מטרות הפרויקט:**

**מטרה:** בניית ספריית פייטון המאפשרת .....שמטרתה היא לאפשר פיתוח פשוט ומובן וידידותי למפתחים.

**יעד:** הכלים יהיו ידידותיים וקלים להבנה כך שלמשתמש יהיה את המון מידע שיעזור לו בפיתוח מהיר תוך כדי שליטה מוחלטת בפרמטרים הפנימיים של הספרייה.

**מדד:** ידע רחב - 80% מהמשתמשים יבינו תוך שעתיים איך להשתמש ב90% מהספרייה ו- 100% גישה לשינוי פרמטרים.

**מטרה:** ספריית פייטון שמטרתה פיתוח בדיקות אשר יבדקו פעולות המדמות תרחישי משתמש באופן אוטומטי.

**יעד:** כלים שהמשתמש צריך כדי שיוכלו לסייע לו לפיתוח מהיר.

**מדד-1:** חיפוש מהירות של אובייקטים במסך - 1-3 שניות לכל אובייקט, בכל ניסיון.

**מדד-2:** בכל שיטות לחיפוש אובייקטים במסך לא יהיו יותר מ10% אי הצלחות.

**מטרה:** למשתמש לא יהיה צורך בידע מוקדם בתכנות כדי לדעת להשתמש בכלי הספרייה.

**יעד:** הסבר מפורט וידידותי איך משתמשים בכל הפונקציות.

**מדד:** 90% מהאנשים יצליחו להשתמש בכלים של הספרייה בלי ידע מוקדם בתכנות.

**יעד סופי - מדד להצלחת הפרויקט:**

היעד הסופי של AutoDesktop הוא תוכנה אשר תעזור לייצור תרחישי UI שכאשר תוך שעתיים יהיה ל 90% מהמשתמשים שלה מובן וקל לייצר תרחישי UI בצורה יעילה מבלי שהיו חסרים להם כלים לכך.

**סקירת ספרות ראשונית:**

Challenges in GUI Test Automation[1]:

1. במאמר מופיע איבחון מעמיק של מספר שיטות שונות של זיהוי והכרות עם האובקייטים והפקדים בבדיקות אוטמטיות של ממשק המשתמש הגראפי, המאמר כולל את התרונותיהן וחסרונותיהן של אותן שיטות.

**לדוגמא:**

* איתור אובייקטים ע"פ ID – שיטה מדוייקת אך דורשת הכרה של כל הIDs
* שם האובייקט (class) – שיטה טובה אך לא תמיד עובדת ודורשת הכרה של כל שמות האובייטקים.
* סידור היררכי של חלונות – לא עמידה בשינויי UI.

1. פתרון ה GUIRobo:

תוכנה אשר מאפשרת בדיקות דחק, עשן וביצועים ע"י כתיבת סקריפטים בפסיאודו קוד.

The First Decade of GUI Ripping: Extensions, Applications, and Broader Impacts[2]:

פירוק של ממשק המשתמש על ידי reverse engineering שמאפשר לאבחן המון use cases ע"י כך נבין את הכישלונות הGUI ואת התוכנות של מקרי הבדיקה ולאתר אותם.

ברמה הגבוהה, בניית מודלים אשר יקבלו קלט ויציגו למשתמש פלט ולאחר מכן יתוארו האלגוריתמים אשר השתמשו בהם.

OHT: Open and Cross-platform GUI Testing[3]:

**Open HMI Tester (OHT) – תוכנת בדיקות GUI:**

**תוכנה לייצור תרחישי UI הכוללת:**

* סביבת עבודה המאפשרת עבודת פיתוח ממושכת ונוחה שבעזרתה ניתן ליצור תרחישי משתמש.
* אפשרות לתמיכה בכל מערכות הפעלה – מה ששונה מהערכת שלי.
* עמידה בשינויי ה GUI בלי שינוי ניכר של הקוד.
* ליצור סימולציות המדמות פעולות של משתמש שאפשרות ליצור ברמה גבוהה בדיקות GUI, ברמה שאין תלות במיקום האובייקטים או ברזולוציית המסך.
* המשתמש בOHT יכול לשנות את עיצוב התוכנה לפי איך שנוח למשתמש – Open Design.
* OHT היא מערכת קוד פתוח - תוכנה ש[קוד המקור](https://he.wikipedia.org/wiki/%D7%A7%D7%95%D7%93_%D7%9E%D7%A7%D7%95%D7%A8) שלה פתוח ונגיש לכל מי שחפץ בו והוא חופשי לשימוש, לצפייה, לעריכת שינויים ולהפצה מחודשת.
* נתמכת במערכות הפעלה Linux ו- Windows.

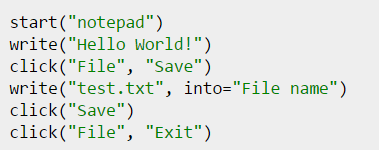
**סקר שוק ראשוני:**

Automa – אוטומה[4]:

ספריה הכתובה בשפת פייטון שבעזרתה ניתן ליצור תרחישי UI:

ספריה זו ידידותית וקלה ללמידה, שבעזרתה ניתן לייצר תרחישי UI מהר מאוד ללא כל ניסיון בפיתוח וניתן לפנות לכל כפתור או טקסט ב UI באופן ישיר ומהיר על ידי הטקסט הכתוב על אותו אובייקט (במידה וקיים), בנוסף ניתן לקרוא טקסט מהמסך.

שמות המטודות קלות להבנה.

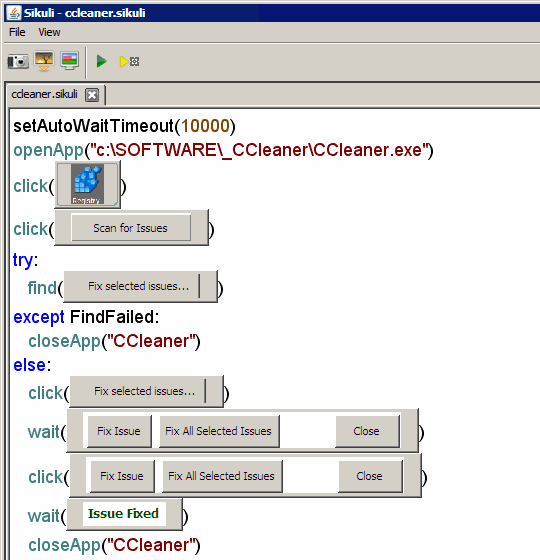


Sikuli – סיקולי[5]:

יישום זה הוא סביבת עבודה (IDE) בעזרתה ניתן לייצר תרחישי UI, סביבת עבודה זו מאפשרת גישה לכל רכיב GUI אשר קיים במסך.

ניתן להכניס תמונות של אובייקטים וקומפוננטות בתוך הקוד (כמו שניתן לראות בתמונה).

מאוד מובנת אך **מיועד אך ורק למפתחים**, אבל מובנת מאוד ואינה דורשת הבנה של מבנה הUI של התוכנה הנבדקת.

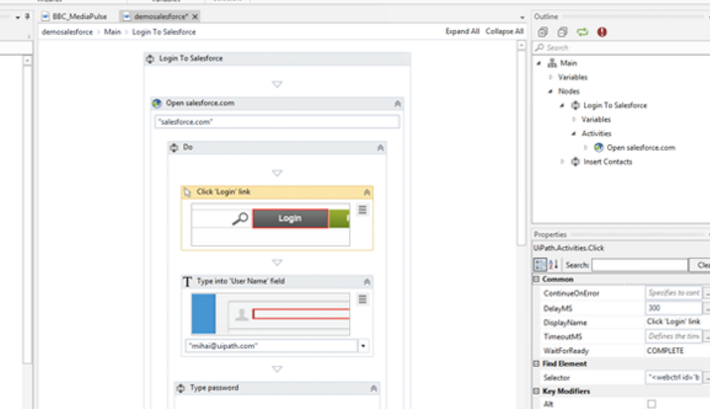


UiPath[6]:

שיטה זו היא סביבת עבודה (IDE) בעזרתה ניתן לייצר תרחישי UI המיועדת למפתחים, סביבת עבודה זו מבינה את הGUI של מערכת ההפעלה ויכולה לשלוט בלוגיקה שלו בלי לדעת היכן כל קומפוננטה נמצאת על המסך.

סביבה עבודה זו מאוד ויזואלית (תמונות בתוך הקוד). מאפשרת שינויים קלים בקוד במהירות.

שיטה זו איזה תלויה בשינויי רזולוציית המסך – **מה שמאפשר להריץ את אותו התרחיש במחשבים שונים**.



[Ranorex – רנורקס[7]:](https://technet.microsoft.com/en-us/library/8fb08f1b-0e08-472d-83db-313e2e56e400)

שיטה זו מספקת 3 דברים:

* כתיבת תרחישים בעזרת קוד – סביבית עבודה (IDE) אפשר מספקת מחלקות לכל אובייקטים וקומפוננטות שנמצאות בUI.
* יצירת תרחישים בעזרת הקלטה – כאשר המשתמש מבצע את התרחיש בעצמו, התוכנה מקליטה את ההתנהגות שלו ושומרת את זה בתור תרחיש UI.
* דו"ח תוצאה על כל תרחיש – דו"ח המכיל את השגיאות כאשר התוכנה אינה מצאה אובייקטים או שהתרחיש נכשל, הזהרות והסבר על כל צעד בתרחיש כדי שתהיה אפשרות להבין ולתקן באגים.

השוואות:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AutoDesktop | Automa | Sikuli | [UiPath](http://m.wikihow.com/Block-an-Application-or-.EXE-from-Running-in-Windows) | Ranorex |  |
|  |  |  |  |  | User friendly |
|  |  |  |  |  | Independent resolution  change |
|  |  |  |  |  | Supply IDE |
|  |  |  |  |  | Designate to Non-Developers |
|  |  |  |  |  | Record |
|  |  |  |  |  | Report |
|  |  |  |  |  | Intuitive method names |
|  |  |  |  |  | Easy to maintain small changes in the code. |



לא נתמך.

נתמך.

נתמך אך לא ממולץ.

**מסקנות:**

* כל המערכות המדוברות הם ידידותיות, כלומר ידידותיות המערכת מאוד חשובה.
* AutoDesktop היא היחידה אשר מיועדת למשתמשים בלי רקע של תכנות.
* AutoDesktop תספק אפשרות להקלטה אך אינה ממליצה שימוש בפונקציה הזאת בגלל ירידה בביצועי המערכת.
* AutoDesktop תספק דו"ח סיום, אך הוא יהיה הסבר של זרימת התחריש.
* רוב המערכות מספקות IDE – מערכת ה AutoDesktop לא תספק IDE מכיוון שהיא מספקת תוכנה ידידותית שאינה מיועדת למתשמשים עם רקע של תכנות.
* רוב המערכות עמידות בשינויי UI של המערכת הנבדקת.

**מסמך דרישות ראשוני של המערכת:**

**דרישות פונקציונליות:**

1. **יצירת תרחיש חדש** – אפשרות לייצר תרחיש חדש.
2. **שמירת תרחיש** – אפשרות לשמור את התרחיש עליו המשתמש עובד.
3. **עריכת תרחיש קיים** – אפשרות לערוך תרחיש שמור.
4. **מחיקת תרחיש קיים** – אפשרות למחוק תרחיש שמור.
5. **הוספת פעולת UI[[2]](#footnote-1) לרשימת הפעולות** – אפשרות להוסיף פעולת UI לתוך רשימת הפעולות המוצגת לפניו בזמן עבודתו על התוכנה.
6. **חיפוש אוביקט** – אפשרות לחפש אובייקטים על המסך (תמונה, כפתור, סמנים ועוד).
7. **שינוי מספר הניסיונות לחיפוש אובייקטים** – אפשרות לשנות את מספר ניסיונות חיפוש האובייקטים כאשר החיפוש נכשל.
8. **שינוי זמן חיפוש האוביקט** – אפשרות לשנות את זמן חיפוש האובייקט (מעל 2 שניות לכל אובייקט).
9. **יצירת השהיית זמן** – כאשר בזמן התרחיש יש מצב שצריך להמתין לאובייקט על המסך, קיימת אפשרות לייצר השהיות של זמן.
10. **למשתמש יהיה כלי לבחירה/מציאת אובייטקים על המסך**.
11. **הקלקת העכבר** – פעם אחת או מספר פעמים על המסך.
12. **יצירת תנאים (IF-ELSE) בזמן התרחיש** – אפשרות לייצר מספר מצבים שונים כאשר קיימות מספר אפשרויות שונות בזמן התרחיש.
13. **יצירת קוד פיטון** – פיתוח. האפליקציה מאפשרת למשתמשים עם רקע של תכנות לייצר את התרחישים בקוד פייטון.

**דרישות לא פונקציונליות:**

דרישות עיצוב, שימוש וממשק משתמש:

1. שפת התוכנה תהיה באנגלית מובנת וידידותית.
2. רשימת פעולות הUI תהיה מוצגת בכל מסך בתוכנה.
3. כאשר המשתמש יבחר בכתיבת קוד פייטון יפתח חלון של עריכת טקסט לכתיבת הקוד.
4. שמות המטודות בספרייה יהיו מובנות וברורות.
5. כאשר המשתמש יבחר בפעולת UI יוצג לו הסבר מפורט על הפעולה.
6. הסבר הפעולות יהיה ניתן להסרה לאחר פעם שימוש ראשוני בפעולה.

דרישות ביצועים:

1. ניסיון חיפוש אוביקט התבצע 3 פעמיים כברירת מחדל.
2. חיפוש האוביקטים יהיה 2 שניות לכל היותר (יהיה ניתן לשנות).

דרישות הפעלה:

1. Windows XP / 7 / 8 / 10 (32/64 Bits).
2. Python 3.6.0 or later releases.

דרישות תחזוקה ותמיכה:

1. שמירת היסטוריה של תרחישים.
2. יהיה הסבר מפורט על כל הפעולות האפשריות.

**תצלום ראשוני של AutoDesktop:**

Actions list

Sleep(2sec)

Search object(Name.png)

Sleep(3sec)

Click (Name.png)

DoubleClick(Name.png)

Sleep(3sec)

**Sleep**

3

Sec

**Search Object**

Path to image

**…**

**Click**

3

Times

**New Scenario**

**Delete Scenario**

**Clear Action**

**AutoDesktop**

**Coding**

**Save Scenario**

**Run**

**000**

**Move Mouse**

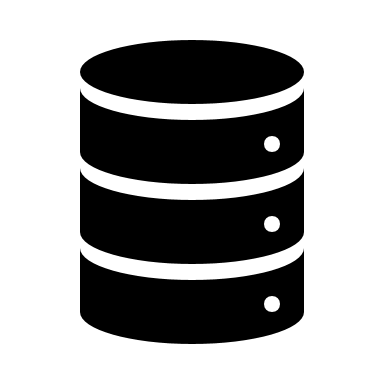
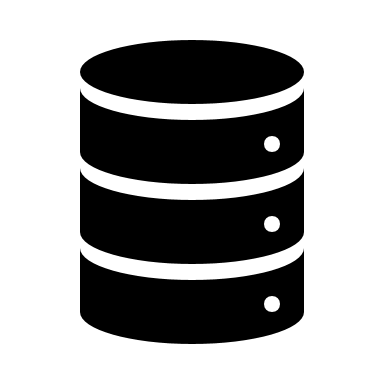
785

X

300

Y

**תרשים בלוקים/מודולים כללי של המערכת:**



Converting Pseudo Code to Python Code

Supported

UI Actions

User’s UI Actions List

Insert a UI Actions

Compare

Python Code Generator

User

**Insert a UI Action:**

הכנסת פעולת UI למערכת ע"י המשתמש.

**User’s UI Action List:**

רשימת פעולות הUI שהכניס המשתמש למערכת שיוצרות את התרחיש כולו.

**Supported UI Actions:**

רשימת פעולות UI שמורות אשר נתמכות ע"י AutoDesktopשאיתם ניתן לבצע תרחישים.

**Converting Pseudo Code to Python Code:**

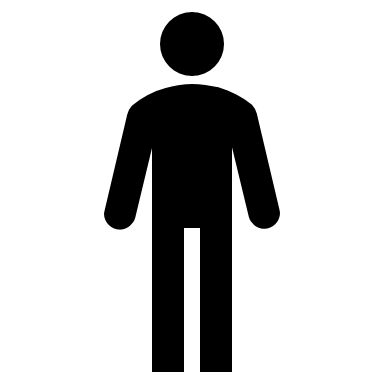
המרה של פעולות הUI השמורות יחד עם רשימת פעולות הUI שמשתמש הכניס למערכת לקוד פייטון.

**Python Code Generator:**

מחולל קוד פייטון – קוד פייטון המוכן לריצה.

**ניתוח פונקציונאלי ראשוני:**

**AutoDesktop**



Automatic

User

**שחקן: User**

**מערכת: AutoDesktop**

**Create a Scenario (1 – יצירת תרחיש:**

**זרימה עיקרית:**

1. המשתמש בוחר ביצירת תרחיש חדש - התוכנה מאפשרת לו בעזרת הקלה על הכפתור "New Scenario”.
2. המשתמש בוחר ממבחר הפעולות המוצגות מולו פעולת UI אשר אותה הוא רוצה שהתוכנה תבצע.
3. הפעולת הUI נכנסת לרשימת הפעולות אשר תומר בעת השמירה לקוד פייטון.

**זרימה חלופית:**

1. המשתמש בוחר ביצירת תרחיש חדש - התוכנה מאפשרת לו בעזרת הקלה על הכפתור "New Scenario”.
2. המשתמש בוחר ממבחר הפעולות המוצגות מולו פעולת UI אשר אותה הוא רוצה שהתוכנה תבצע.
3. **הפעולה לא נכנסת לרשימת הפעולות אשר תומר בעת השמירה לקוד.**
4. תוצג הודעת שגיאה על המסך שהפעולה אינה נכנסה לרשימת הפעולות מסיבה כלשהי.

**Delete a Scenario (2 – מחיקת תרחיש:**

**זרימה עיקרית:**

1. המשתמש בוחר בתרחיש קיים.
2. בוחר למחוק אותו.
3. התרחיש נמחק.
4. מוצגת הודעה על המסך – "התרחיש <שם התרחיש> נמחק".

**זרימה חלופית:**

1. המשתמש בוחר בתרחיש קיים.
2. בוחר למחוק אותו.
3. **התרחיש אינו נמחק.**
4. מוצגת הודעת שגיאה על המסך – "התרחיש <שם התרחיש> אינו נמחק מסיבה כלשהי".

**Save a Scenario (3 – שמירת תרחיש:**

**זרימה עיקרית:**

1. המשתמש בוחר לשמור את התרחיש אשר יצר.
2. התרחיש נשמר כקובץ טקסט מקומי על המחשב.
3. מתבצעת המרה אוטומטית מקובץ הטקסט לקוד פייטון.
4. מוצגת הודעה על המסך – "התרחיש <שם התרחיש> נשמר".

**זרימה חלופית:**

1. המשתמש בוחר לשמור את התרחיש אשר יצר.
2. התרחיש נשמר כקובץ טקסט מקומי על המחשב.
3. **לא** **מתבצעת המרה אוטומטית מקובץ הטקסט לפייטון.**
4. מוצגת הודעת שגיאה על המסך – "התרחיש <שם התרחיש> נשמר ככקובץ טקסט אך לא הומר לקוד מסיבה כלשהי".

**Run a Scenario (4 – הרצת תרחיש:**

**זרימה עיקרית:**

1. המשתמש בוחר בתרחיש קיים.
2. בוחר להריץ את התרחיש.
3. מוצגת בחירה של מספר השניות לספירה לאחור לתחילת התרחיש.
4. המשתמש בוחר ב5 שניות.
5. ספירה לאחור של 5 שניות והמשתמש עוזב את העכבר והמקלדת.
6. התרחיש רץ.

**זרימה חלופית:**

1. המשתמש בוחר בתרחיש קיים.
2. בוחר להריץ את התרחיש.
3. מוצגת הודעת שגיאה על המסך "התרחיש <שם התרחיש> אינו ניתן להרצה מסיבה כלשהי".

**Create a Code (5 – יצירת קוד:**

**זרימה עיקרית:**

1. המשתמש יכול לבחור באופציה ליצור קוד פייטון בעצמו.
2. נפתח חלון עריכת טקסט.
3. המשתמש כותב קוד פייטון.

**זרימה חלופית:**

1. המשתמש יכול לבחור באופציה ליצור קוד פייטון בעצמו.
2. **חלון עריכת הטקסט לא נפתח.**
3. מוצגת הודעת שגיאה על המסך "חלון עריכת הטקסט אינו יכול להפתח מסיבה כלשהי".

**הצגה ראשונית של חלופות מערכתיות:**

* מערכת/תוכנה כלשהי אשר המפעיל שלה הוא **בודק ידני** שמריץ סקריפטים של תרחישי UI.
* בדיקות ידניות – בדיקות תוכנה הנעשות על ידי עובד שהוכשר לכך בדרך כלל ע"פ תוכנית בדיקות מסודרת ומוסכמת, כלומר בנאדם הבודק את המערכת באופן ידני אשר מדמה תרחישי משתמש על ה UI.
* מערכת שהיא בוט שמבצע תרחישי UI על המחשב באופן אוטומטי עם שליטה מרחוק.

**האמצעים/הכלים הנדרשים:**

* Windows XP / 7 / 8 / 10 (32/64 Bits).
* Python 3.6.0 or later releases.

**תוצרי הפרויקט:**

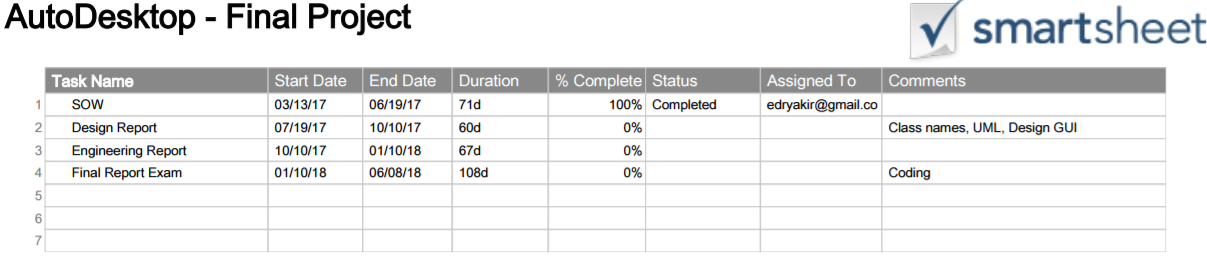
תוכנה או/ו ספרייה קלה לתפעול אשר תתן מענה לארגונים ולאנשים פרטיים.

התוכנה תאפשר לממש תרחישי משתמש לבדיקות UI וכתוצאה מכך תחסוך בזמן ובכח אדם לאירגונים ואנשים פרטיים.

עם זאת שימוש בתוכנה או/ו בספריה יגלה בעיות UI לאירגונים ויפנה משאבים נחוצים במקומות אחרים.

**בדיקות פונקציונליות:**

* + 1. בדיקה שאפשר ליצור תרחיש חדש.
    2. בדיקה שאפשר לשמור תרחיש.
    3. בדיקה שאפשר למחוק תרחיש קיים.
    4. בדיקה שאפשר לייצר תרחישים בעזרת קוד פייטון.
    5. בדיקת הרצת תרחיש קיים וביצועיו.
    6. בדיקה האם כאשר הוספתי פעולת UI לתרחיש היא הוספה.
    7. בדיקה האם לאחר הוספתי פעולת UI לתרחיש היא מוצגת שהוספה.

**תוכנית עבודה ראשונית לביצוע הפרויקט:**

**רשימת מקורות (ראשונית):**

[1] G. Mohan Doss Gandhi and Anitha S. Pillai. “Challenges in GUI Test Automation”. International Journal of Computer Theory and Engineering, Vol. 6, No. 2, April 2014. PP 194-195.

[2] Atif Memon, Ishan Banerjee, Bao N. Nguyen, Bryan Robbins. “The First Decade of GUI Ripping: Extensions, Applications, and Broader Impacts”. Department of Computer Science, University of Maryland, College Park, MD 20742, USA, 2013. PP 11-12.

[3] Pedro Luis Mateo Navarro, Diego Sevilla Ruiz and Gregorio Martínez Pérez. “OHT: Open and Cross-platform GUI Testing“. University of Murcia, 30.071 Murcia, Spain, 2016. PP 1-4.

[4] <https://www.getautoma.com/> [last seen 19th June 2017]

[5] <https://www.uipath.com/> [last seen 19th June 2017]

[6] <http://www.sikuli.org/> [last seen 19th June 2017]

[7] <https://www.ranorex.com/> [last seen 19th June 2017]

[8] Industrial Automation: A Brief History of Manufacturing Application & The Current State and Future Outlook. available online: <http://cerasis.com/2014/10/22/industrial-automation/> [last seen 19th June 2017]

1. **תסריט (Script)** - תוכנית מחשב הנכתבת על מנת למכן ביצוע משימות, שאחרת היו אולי מבוצעות באופן ידני על ידי משתמש בסביבת תוכנה. [↑](#endnote-ref-1)
2. פעולת UI הינה פעולה אשר מדמה את המשתמש (לדוגמא: להזיז את העכבר, להמתין כמה שניות וכו') [↑](#footnote-ref-1)