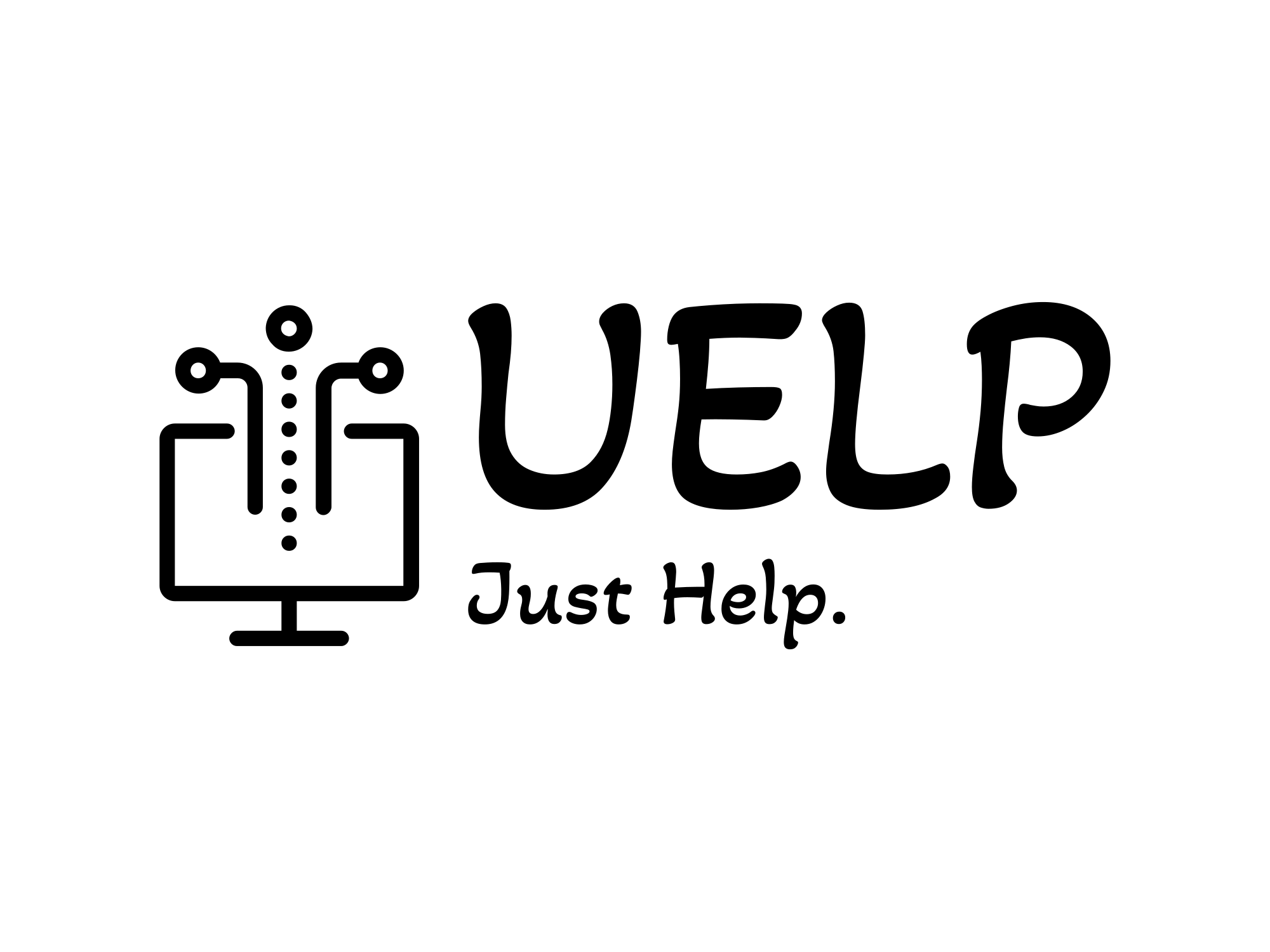
**תיק פרויקט- UELP**

**מערכת לבקרה ושליטה על מחשבים מרחוק**

**בית ספר: כרמים בנימינה**

**תלמיד: איתמר נחום**

**ת"ז תלמיד: 215871443**

**המנחה: מרי גבע**

**שם החלופה: מערכות הגנת סייבר**

**תאריך הגשה: 24/05/18**

**תוכן עניינים וראשי פרקים (אוטומטי לא ידני)**

**פרק א –UELP - ייזום**

**תיאור ראשוני של המערכת-**

1. תיאור ראשוני של המערכת

**מערכת בקרת מחשב מרחוק היא תוכנה המאפשרת לאדם לסייע לאדם אחר ממחשב מרוחק.  
המערכת תחבר בין** Helper **(אנשים שמבינים במחשבים ומעוניינים לסייע לאחרים) לבין** AssistanceSeeker **(אנשים המעוניינים לקבל סיוע מאחרים)**

**המערכת תבטיח קבלת סיוע אך ורק מ-** Helper**-ים 'מאושרים' כדי שלא יגרם נזק ל-** AssistanceSeeker.

**המוצר הסופי אמור לאפשר** Tech Support **לכל אדם ממישהו אחר בכל מקום בעולם.**

**בחרתי בפרויקט זה מכיוון שמאוד התחברתי לרעיון אני חושב שהוא יאתגר אותי וילמד אותי להתעסק בדברים שונים בתחום הסייבר שעדיין לא נגעתי בהם. בנוסף אני רואה שימוש אמיתי לפרויקט שלי בחיי היום יום של כל בן אדם.**

**אתגרים שצפויים לי בפרויקט הם מימוש שליטה על מחשב מרחוק, לממש יכולת לראות מסך של מישהו אחר בזמן אמת ולתקשר עם רכיבים במחשב כמו מקלדת ועכבר בלי** delay**. בנוסף היכולת של התוכנה לבצע כמה** sessions **של תקשורת שעוברת דרך השרת בשביל לאפשר לכמה מחשבים לבצע בקרה בו זמנית ושכל** sessions **יעבדו באופן רציף בלי** delay**.**

1. הגדרת הלקוח

המערכת מיועדת לכל מי שצריך תמיכה מרחוק ומוכן לסמוך על מישהו אחר לעזור לו בנוסף המערכת יכולה להיות בשימוש רב בחברה בינלאומית עם הרבה עובדים בכל העולם. הלקוחות של המערכת יהיו כל אחד שמבקש תמיכה או מציע תמיכה.

1. הגדרת יעדים/מטרות

* להוות מרכז תמיכה שמחבר בין AssistanceSeeker לבין Helper בצורה בטוחה.
* ליצור תשתית שתאפשר ל AssistanceSeeker לתמוך מרחוק במחשב של Helper בצורה נוחה, יעילה ומהירה.

1. בעיות, תועלות וחסכונות

הבעיה שאותה אנו מנסים לפתור עם מערכת זו היא היכולת לקבל תמיכה למחשב מרחוק, כלומר בלי צורך לצאת מהבית בשביל לקבל תמיכה. אנחנו מנסים להשיג דרך אמינה, מהירה וישירה לייצר תמיכה מרחוק לאנשים מכל העולם בלי הגבלה.

תועלתמרכזית אחת שיש למערכת זו היא היכולת לכמה אנשים לבצע תמיכה בו זמנית, התוכנה עובדת כך שאנחנו לא מוגבלים בכמות אנשים ואפשר שכמה אנשים יעזרו למחשבים אחרים בו זמנית.

תועלת נוספת שיש לפרויקט זה הוא שהשליטה של המחשב תלויה ב AssistanceSeeker בעיקר, כלומר אין מצב בו Helper יכול להשיג שליטה על מחשב ולא לנתק אותו כך שהוא יכול לבצע דברים שהמבקש תמיכה לא בסדר איתם, למבקש תמיכה תמיד יש את האופציה לנתק את הקשר. בנוסף כל קשר תלוי בזמן כלומר אם נראה לנו כי קשר מת (כלומר עבר כמות זמן מסוימת עד שה Helper לא ביצע משהו, הקשר מתנתק).

שירותים שהמערכת תיתן הם:

* מרכז תמיכה שמאפשר ל- Helper ול- AssistanceSeeker להתחבר אליו במטרה להבטיח מהימנות ה- Helper -ים.
* חיבור בין Helper ל- AssistanceSeeker ע"פ בקשת ה Helper כלומר ל Helper יש אפשרות לבחור מי תומך בו מבין ה- AssistanceSeeker הקיימים במערכת.
* תשתית ל- Helper שתאפשר לו לתמוך מרחוק ע"י שליטה במחשב ה – Helper.
* בקרה על ה"חיבור" כדי להבטיח את מהימנות התמיכה.

פתרונות ויישומים הקיימים:

TeamViewer- תוכנה זו מאפשרת לפתוח סאשנים של תמיכה מרחוק הנשלחים למבקש תמיכה או התומך על ידי קוד או אימייל. מול התוכנה שלי, ההבדל העיקרי הוא שהמערכת הזאת לא סגורה כלומר אפשר לעבוד איתה גם בלי להוריד את התוכנה למחשב.

<https://www.teamviewer.com/en-mea/>

AnyDesk- תוכנה זו מאפשרת תמיכה ושליטה בין שני מחשבים באופן מהיר ואמין כך שכל אחד יכול לקבל תמיכה מכל מקום בעולם.

<https://anydesk.com/en/solutions/remote-support>

1. האם צפויים קשיים או מגבלות בהגדרת המערכת

לא מדובר בטכנולוגיה חדשה. מאחר שהפרויקט מפותח ברשת הבית ספרית שהינה רשת סגורה כל משתמשי המערכת יהיו חייבים להיות משתמשים ברשת זאת לכן אין מגבלות הצפויות.

1. תיחום הפרויקט

הפרויקט עוסק בעיקר ברשתות ומערכות הפעלה.

רשתות – שרת מרובה לקוחות העברת כל האינפורמציה דרך הרשת.

מערכות הפעלה - מערכת הפעלה תצטרך לתת שליטה מלאה למחשב האחר לעבוד עליה ולתקשר עם הרכיבים הפנימיים של המחשב דרך הרכיבים החיצוניים של המחשב השני כלומר עכבר, מקלדת, מסך כמו כן שימוש ב thread-ים וב – Process-ים למקבול תהליכים.  
שימוש בהצפנה להבטחת קישורים ובבסיס נתונים לשמירת מידע של משתמשים.

המערכת לא מטפלת בתיקון בעיות פנימיות של המחשב, היא מטפלת רק ביצירת הקשר לבצע תמיכה מרחוק בין מחשבים.

1. קיימות

בעידן שלנו הרבה פעמים מחשבים נתקלים בבעיות כמו בעיות בתוכנה או ברכיבים הפיזיים של המחשב בדרך כלל שדברים כאלו קורים אנחנו קונים מחשב חדש או צריכים לקחת את המחשב לתיקון שמצריך הרבה כסף, זמן ומשאבים שפוגעים בכדור הארץ, כמו נסיעה במכונית. בנוסף, בהמון מצבים אפשר לפתור את הבעיה בלי אפילו לפתוח את המחשב ולחשוף את הרכיבים שלו. עם התוכנה שלי אפשר לפתור את כל הבעיות האלו בלי לבזבז כסף או לפגוע בכדור הארץ. כאשר אנו פועלים מהבית ולא שורפים דלק אנחנו עוזרים לכדור הארץ.

**פרק ב' - UELP - אפיון**

1. פרוט המערכת:

**מערכת בקרת מחשב מרחוק היא תוכנה המאפשרת לאדם לסייע לאדם אחר ממחשב מרוחק.  
המערכת תחבר בין Helper (אנשים שמבינים במחשבים ומעוניינים לסייע לאחרים) לבין AssistanceSeeker (אנשים המעוניינים לקבל סיוע מאחרים)**

**המערכת תבטיח קבלת סיוע אך ורק מ- Helper-ים 'מאושרים' כדי שלא יגרם נזק ל- AssistanceSeeker.**

**המוצר הסופי אמור לאפשר Tech Support לכל אדם ממישהו אחר בכל מקום בעולם.**

**בחרתי בפרויקט זה מכיוון שמאוד התחברתי לרעיון אני חושב שהוא יאתגר אותי וילמד אותי להתעסק בדברים שונים בתחום הסייבר שעדיין לא נגעתי בהם. בנוסף אני רואה שימוש אמיתי לפרויקט שלי בחיי היום יום של כל בן אדם.**

**אתגרים שצפויים לי בפרויקט הם מימוש שליטה על מחשב מרחוק, לממש יכולת לראות מסך של מישהו אחר בזמן אמת ולתקשר עם רכיבים במחשב כמו מקלדת ועכבר בלי delay. בנוסף היכולת של התוכנה לבצע כמה** sessions **של תקשורת שעוברת דרך השרת בשביל לאפשר לכמה מחשבים לבצע בקרה בו זמנית ושכל** sessions **יעבדו באופן רציף בלי** delay**.**

1. מה היכולות שהיא תעניק למשתמש , פירוט היכולות

Helper-

* התחברות למערכת- מתחבר למערכת בעזרת שם משתמש סיסמא ואימייל.
* קבלת קוד זמני- מקבל קוד זמני מהמערכת להעביר למשתמש המבקש עזרה בשביל לפתוח את התמיכה ביניהם. (הקוד זמני)
* עצירת שיתוף- כאשר אנו רוצים לסגור את התמיכה לוחצים על הכפתור וחוזרים למצב של יכולת לפתוח סאשן תמיכה.
* שליטה על העכבר של ה AssistanceSeeker
* שליטה על המקלדת של ה AssistanceSeeker
* צפייה רציפה במסך ה- AssistanceSeeker

AssistanceSeeker-

* התחברות למערכת- מתחבר למערכת בעזרת שם משתמש סיסמא ואימייל.
* הכנסת קוד התחברות זמני- הכנסת הקוד של התומך בשביל להתחיל את התמיכה.
* עצירת שיתוף- כאשר אנו רוצים לסגור את התמיכה לוחצים על הכפתור וחוזרים למצב "מבקש תמיכה" שבו רואים את כל המשתמשים שמציעים תמיכה.

1. פירוט הבדיקות ('קופסא שחורה')

(בדיקות לביצוע אך ורק עם קובץ ההרצה, ללא הקוד, בדגש על כך שכל אחד לפי הכתוב בטבלה יצליח לבצע את הבדיקה)

| **מספר** | **שם הבדיקה (שם שיעיד על תוכן הבדיקה)** | **מה אמורה לבדוק** | **איך מתכננים לבדוק**  **(לתאר בפירוט את שלבי הבדיקה)** |
| --- | --- | --- | --- |
| 1 | פתיחת תמיכה בין שני משתמשים | יצירת תמיכה חדשה עם קוד מהתומך | התחברות למערכת על ידי שני מחשבים, אחד של מבקש תמיכה ואחד של תומך. התומך יוצר קוד והמבקש תמיכה מכניס את הקוד שנוצר לפני סיום הזמן המוגבל, בודקים שכל השליטה עברה לתומך והקוד באמת עבד |
| 2 | תמיכה עובדת עם מסך | בדיקה שרואים את המסך של המבקש תמיכה במסך של התומך | התחברות למערכת על ידי שני מחשבים, אחד של מבקש תמיכה ואחד של תומך. התומך יוצר קוד והמבקש תמיכה מכניס את הקוד שנוצר התומך בודק שכאשר הוא מתחבר הוא רואה את המסך של המבקש תמיכה כפי שהוא נראה באופן רציף גם כאשר יש שינויים |
| 3 | תמיכה עובדת עם עכבר | בדיקת תזוזת העכבר של התומך כעכבר של המבקש עזרה | התחברות למערכת על ידי שני מחשבים, אחד של מבקש תמיכה ואחד של תומך. התומך יוצר קוד והמבקש תמיכה מכניס את הקוד שנוצר. התומך בודק שכאשר הוא מזיז את העכבר שלו זה גם מזיז את העכבר במחשב של המבקש תמיכה ומגיב באופן רציף לתזוזות של העכבר |
| 4 | תמיכה עובדת עם מקלדת | בדיקת המקלדת של התומך כמקלדת של המבקש עזרה | התחברות למערכת על ידי שני מחשבים, אחד של מבקש תמיכה ואחד של תומך. התומך יוצר קוד והמבקש תמיכה מכניס את הקוד שנוצר. התומך בודק שכאשר הוא מקיש מקשים במקלדת שלו זה נקלט כהמקלדת של המבקש תמיכה ומבצע את הפעולה על מחשבו. |
| 5 | קיום תמיכה בין יותר מ2 מחשבים | בדיקה שאפשר לבצע תמיכה בין כמה מחשבים בו זמנית בלי הפרעה | התחברות למערכת על ידי 4 מחשבים, שניים של מבקשי תמיכה ושניים של תומכים. התומכים יוצרים קוד ומבקשי התמיכה מכניסים את הקודים הזמניים שנוצרו. אנו בודקים שכל הבדיקות 2-4 מתקיימות באופן רציף ומהיר על שני הסאשנים בלי בעיה או delay. |
| 6 | בדיקת פג תוקף של "קוד תמיכה" | בדיקה שהקוד תמיכה שיוצר התומך הוא זמני ולא עובד יותר מפרק זמן מסוים. | התחברות למערכת על ידי שני מחשבים, אחד תומך ואחד מבקש תמיכה, התומך יוצר קוד ומחכים עד שפרק הזמן שהקוד תקף בו ייגמר, רואים שהקוד לא עובד והוא כבר לא תקף בשביל המבקש תמיכה. |
| 7 | סיום תמיכה ע"י הנתמך | בדיקה שמבקש התמיכה מצליח לסיים במיידית תמיכה | התחברות למערכת על ידי שני מחשבים, אחד של מבקש תמיכה ואחד של תומך. התומך יוצר קוד והמבקש תמיכה מכניס את הקוד שנוצר. המבקש תמיכה בודק שכאשר הוא לוחץ על הכפתור לסיים את הבקרה, אז הבקרה באמת מסתיימת גם מהצד שלו וגם מהצד של התומך, כלומר כל השליטה חוזרת אליו. |
| 8 | סיום תמיכה ע"י התומך | בדיקה שנותן התמיכה מצליח לסיים במיידית תמיכה | התחברות למערכת על ידי שני מחשבים, אחד של מבקש תמיכה ואחד של תומך. התומך יוצר קוד והמבקש תמיכה מכניס את הקוד שנוצר. התומך בודק שכאשר הוא לוחץ על הכפתור לסיים את הבקרה, אז הבקרה באמת מסתיימת גם מהצד שלו וגם מהצד של המבקש תמיכה, כלומר כל השליטה חוזרת למבקש תמיכה. |
| 9 | ביצועים | בדיקה שרמת הביצועים לא מושפעת מכמות המשתמשים | התחברות למערכת על ידי 7+ מחשבים, יצירת כמה סאשנים תמיכה ובדיקה שלא נפגעת היכולות של המערכת עכב כמות האנשים. |
| 10 | יציבות | בדיקה שהמערכת לא קורסת בעת שימוש ביכולות שלה | בדיקת היכולות של המערכת כולל, התחברות, יצירת קוד, להיכנס לסאשן ולבצע שימוש ברכיבים החיצוניים של המחשב, הפסקת סאשן והתנתקות מהמערכת בעת כל נקודה בשימושה. |

1. תכנון לוח זמנים לפרויקט

| **פעילות** | **זמן התחלה מתוכנן** | **זמן סיום מתוכנן** | **זמן התחלה בפועל** | **זמן סיום בפועל** | **הערות** |
| --- | --- | --- | --- | --- | --- |
| יזום | 6.11.23 | 23.11.23 | 6.11.23 | 8.11.23 | סיום מוקדם |
| אפיון | 9.11.23 | 1.12.23 | 9.11.23 | 12.11.23 |  |
| ניתוח | 13.11.23 | 15.12.23 | 13.11.23 | 23.11.23 |  |
| עיצוב | 29.11.23 | 15.1.23 | 24.11.23 | 14.01.24 |  |
| קוד (לפרוט לאבנים גדולות) | בניית יכולת לשלוט על עכבר של מחשב אחד ממחשב אחר-  15.1.24  בניית יכולת לשלוט על מקלדת של מחשב אחד ממחשב אחר-  22.1.24  בניית יכולת לראות מסך של מחשב אחד ממחשב אחר-  30.1.24  העברת השליטה לבסיס שרת מרובה לקוחות (שרת מתקשר ומחבר בין מחשב אחד למחשב אחר לפי קוד מהנותן תמיכה)-  12.2.24  הוספת אימות נתונים (שם משתמש, סיסמא ואימייל) דרך database-  3.3.24  הצפנה עיצוב, תיקון באגים ושהמערכת עובדת תמיד-  28.3.24 | בניית יכולת לשלוט על עכבר של מחשב אחד ממחשב אחר-  22.1.24  בניית יכולת לשלוט על מקלדת של מחשב אחד ממחשב אחר-  30.1.24  בניית יכולת לראות מסך של מחשב אחד ממחשב אחר-  12.2.24  העברת השליטה לבסיס שרת מרובה לקוחות (שרת מתקשר ומחבר בין מחשב אחד למחשב אחר לפי קוד מהנותן תמיכה)-  3.3.24  הוספת אימות נתונים (שם משתמש, סיסמא ואימייל) דרך database-  28.3.24  הצפנה עיצוב, תיקון באגים ושהמערכת עובדת תמיד-  20.4.24 | בניית יכולת לשלוט על עכבר של מחשב אחד ממחשב אחר-  15.1.24  בניית יכולת לשלוט על מקלדת של מחשב אחד ממחשב אחר-  22.1.24  בניית יכולת לראות מסך של מחשב אחד ממחשב אחר-  05.02.24  העברת השליטה לבסיס שרת מרובה לקוחות (שרת מתקשר ומחבר בין מחשב אחד למחשב אחר לפי קוד מהנותן תמיכה)-  20.02.24  הוספת אימות נתונים (שם משתמש, סיסמא ואימייל) דרך database-  5.03.23  הצפנה עיצוב, תיקון באגים ושהמערכת עובדת תמיד-  26.03.24 | בניית יכולת לשלוט על עכבר של מחשב אחד ממחשב אחר-  22.1.24  בניית יכולת לשלוט על מקלדת של מחשב אחד ממחשב אחר-  04.02.24  בניית יכולת לראות מסך של מחשב אחד ממחשב אחר-  20.02.24  העברת השליטה לבסיס שרת מרובה לקוחות (שרת מתקשר ומחבר בין מחשב אחד למחשב אחר לפי קוד מהנותן תמיכה)-  3.3.24  הוספת אימות נתונים (שם משתמש, סיסמא ואימייל) דרך database-  25.03.24  הצפנה עיצוב, תיקון באגים ושהמערכת עובדת תמיד-  7.05.24 |  |
| גרסה ראשונית | 15.1.24 | 1.3.24 | 15.1.24 | 3.03.24 |  |
| מסמך בדיקות | 1.4.24 | 10.4.24 | 16.05.24 | 18.05.24 |  |
| מדריך למשתמש | 20.4.24 | 1.5.24 | 16.05.24 | 18.05.24 |  |
| גרסה סופית | 1.5.24 | 10.5.24 | 7.5.24 | 18.05.24 |  |
| סגירת תיק פרויקט | 1.5.24 | 10.5.24 | 7.5.24 | 18.05.24 |  |
| מצגת הסבר | 10.5.24 | 15.5.24 | 18.05.24 | 23.05.24 |  |

* **לוח הזמנים – הינו מסמך חי ויש לעדכנו לאורך כל הפרויקט**

1. ניהול סיכונים בפרויקט

* זיהוי סיכוני הפרויקט, ניתוחם ומענה עליהם, הטבלה תמוין מסיכון גבוה לנמוך (הגבוהים באדום, בינוני –כתום , קל – צהוב) סיכון שיוסר יצבע בירוק ויעבור לתחתית המסמך

| **הסיכון** | **פירוט הסיכון** | **רמת הסיכון**  **(קל/בינוני/קשה)** | **תיאור דרכים (לפחות 2 ) להתמודדות עם הסיכון ולהקטין אותו** | **מה בוצע בפועל** | **תאריך** |
| --- | --- | --- | --- | --- | --- |
| אי עמידה בזמנים | פרויקט לא יושלם | קשה | * הקדמת לו"זים משימות * ארגון הזמן בצורה יעילה * להתחיל בדברים היותר קשים | עמדתי בזמנים שהצבתי לעצמי והצלחתי לבצע ולסיים את הפרויקט בהצלחה | 16.05.24 |
| חוסר הצלחה לממש קליטה של המקלדת של המבקש תמיכה | פרויקט לא יושלם | קשה | * בדיקה באינטרנט * בדיקה לפני אם המימוש בכלל אפשרי * הסתייעות בחברים/מורה | יחסית התחלתי עם סוגייה זו אז טילפתי בה ומצאתי לה פתרון מהר כך שלא יצור בעיות בהמשך הפרויקט | 16.05.24 |
| חוסר הצלחה לממש קליטה של העכבר של המבקש תמיכה | פרויקט לא יושלם | קשה | * בדיקה באינטרנט * בדיקה לפני אם המימוש בכלל אפשרי * הסתייעות בחברים/מורה | יחסית התחלתי עם סוגייה זו אז טילפתי בה ומצאתי לה פתרון מהר כך שלא יצור בעיות בהמשך הפרויקט | 16.05.24 |
| חוסר הצלחה לממש שהתומך רואה את המסך של המבקש תמיכה | פרויקט לא יושלם | קשה | * בדיקה באינטרנט * בדיקה לפני אם המימוש בכלל אפשרי * הסתייעות בחברים/מורה | יחסית התחלתי עם סוגייה זו אז טילפתי בה ומצאתי לה פתרון מהר כך שלא יצור בעיות בהמשך הפרויקט | 16.05.24 |
| הבנת wxpython | עיכוב פרויקט | קל | * הבנה דרך גורמים מהאינטרנט * התייעצות עם חברים אחרים מהמגמה | כשהגעתי לגרפיקה חקרתי לעומק את המודול ולמדתי בעזרת חברים האינטרנט והמצגות | 16.05.24 |
| חוסר הצלחה לממש שרת מרובה לקחות | פרויקט לא שלם | קל | * התייעצות עם חברים מהמגמה * לא לעכב מימוש | מימשתי שרת מרובה לקוחות ובניתי את כל הפרויקט עליו כך שפתרתי את בעיה זו בהתחלה | 16.05.24 |
| בקרה איטית לתומך | פרויקט פחות שימושי | קשה | * חשיבה קדימה לממש את כל השלבים לפני בצורה הכי יעילה ככה הבקרה תהיה מהירה בסופו של דבר * התייעצות עם חברים מהמגמה | ביצעתי מספר הכי מינימלי של שליחות שעוברות ברשת כך שהתמיכה תהיה מהירה ויעילה | 16.05.24 |
| ביצועים | פרויקט פחות יעיל | קשה | * חשיבה קדימה לממש את כל השלבים בצורה הכי יעילה ונכונה מבחינת קוד, ככה שהביצועים של המערכת לא יפגעו * התייעצות עם חברים המגמה | ביצעתי מספר הכי מינימלי של שליחות שעוברות ברשת כך שהתמיכה תהיה מהירה ויעילה | 16.05.24 |

* **ניהול סיכונים – הינו מסמך חי ויש לעדכנו לאורך כל הפרויקט**

**פרק ג' - UELP - ניתוח**

פרוט יכולות המערכת

**יכולות בצד שרת:**

שם היכולת: הרשמה למערכת

מהות היכולת: רישום משתמש חדש במערכת

אוסף יכולות:

* קבלת נתונים מהלקוח
* פענוח
* פירק לפי פרוטוקול
* בדיקה מול בסיס הנתונים
* גיבוב סיסמא
* הוספה לבסיס נתונים
* בנה תשובה לפי פרוטוקול ובדיקה
* הצפנה
* שליחה

רשימת אובייקטים: הצפנה/פענוח, תקשורת, בסיס נתונים, פרוטוקול

שם היכולת: התחברות למערכת

מהות היכולת: התחברות עם משתמש קיים למערכת

אוסף יכולות:

* קבלת נתונים מהלקוח
* פענוח
* פירק לפי פרוטוקול
* בדיקה מול בסיס הנתונים
* בנה תשובה לפי פרוטוקול ובדיקה
* הצפנה
* שליחה

רשימת אובייקטים: הצפנה/פענוח, תקשורת, בסיס נתונים, פרוטוקול

שם היכולת: בחירת סוג משתמש

מהות היכולת: בחירת סוג משתמש (עוזר או מבקש תמיכה)

אוסף יכולות:

* קבלת נתונים מהלקוח
* פענוח
* פירק לפי פרוטוקול
* בדיקה מול דרישות כל סוג משתמש
* בנה תשובה לפי פרוטוקול ודרישות סוג המשתמש
* הצפנה
* שליחה

רשימת אובייקטים: הצפנה/פענוח, תקשורת, דרישות סוג משתמש, פרוטוקול

שם היכולת: בקשת פתיחת סאשן

מהות היכולת: יצירת קוד חדש לשני משתמשים במערכת להתחבר

אוסף יכולות:

* קבלת נתונים מהלקוח
* פענוח
* פירק לפי פרוטוקול
* בדיקת סוג המשתמש ותקינות קבלת קוד
* בניית קוד משותף
* התחלת זמן תוקף הקוד ושמירת הקוד
* בנה תשובה לפי פרוטוקול ובדיקה עם קוד וזמן תוקף
* הצפנה
* שליחה

רשימת אובייקטים: הצפנה/פענוח, בניית קוד והגבלת זמנו, תקשורת, פרוטוקול

שם היכולת: בקשת התחברות לסאשן

מהות היכולת: התחברות לסאשן קיים בעזרת קוד זמני קיים

אוסף יכולות:

* קבלת נתונים מהלקוח
* פענוח
* פירק לפי פרוטוקול
* בדיקת סוג המשתמש ותקינות הקוד
* בנה תשובה לפי פרוטוקול והבדיקה
* אם הקוד תקין - **חיבור שני משתמשים ביחד לקיום תמיכה**
* אם הקוד לא תקין, נשלחת הודעה אך ורק למזין הקוד:
  + בניית הודעת ע"פ פרוטוקול למבקש העזרה
* הצפנה
* שליחה

רשימת אובייקטים: הצפנה/פענוח, בדיקת קוד לפי קודים קיימים ותוקפם, תקשורת, פרוטוקול

שם היכולת: חיבור שני משתמשים ביחד

מהות היכולת: חיבור שני משתמשים ביחד לקיום תמיכה

אוסף יכולות:

* **בקשת פתיחת סאשן**
* **בקשת התחברות לסאשן**
* בניית הודעה לשני המשתמשים לפי פרוטוקולים עם מידע להתחברות אחד לשני
* הצפנה
* שליחה

רשימת אובייקטים: הצפנה/פענוח, תקשורת, פרוטוקול עם מידע לחיבור משתמש

**יכולות בצד לקוח:**

שם היכולת: הרשמה למערכת

מהות היכולת: רישום משתמש חדש במערכת (קליטת פרטיים אישיים נדרשים)

אוסף יכולות:

* ממשק משתמש – מסך הרשמה
* קליטת נתונים
* בדיקת תקינות
* בניית הודעה לפי פרוטוקול
* הצפנה
* שליחה לשרת
* קבלת תשובה מהשרת
* פענוח
* מפרק לפי פרוטוקול
* הצגת התשובה למשתמש

רשימת אובייקטים: ממשק משתמש, הצפנה/פיענוח, תקשורת, פרוטוקול

שם היכולת: התחברות למערכת

מהות היכולת: התחברות למערכת עם משתמש קיים (קליטת פרטיים אישיים נדרשים)

אוסף יכולות:

* ממשק משתמש – מסך התחברות
* קליטת נתונים
* בדיקת תקינות
* בניית הודעה לפי פרוטוקול
* הצפנה
* שליחה לשרת
* קבלת תשובה מהשרת
* פענוח
* מפרק לפי פרוטוקול
* הצגת התשובה למשתמש

רשימת אובייקטים: ממשק משתמש, הצפנה/פיענוח, תקשורת, פרוטוקול

שם היכולת: בחירת סוג משתמש (עוזר או מבקש תמיכה)

מהות היכולת: בחירת סוג משתמש (קליטת בחירה לפי כפתור)

אוסף יכולות:

* ממשק משתמש – מסך בחירה (כפתור למבקש תמיכה או תומך)
* קליטת לחיצת כפתור
* בניית הודעה לפי פרוטוקול
* הצפנה
* שליחה לשרת
* קבלת תשובה מהשרת
* פענוח
* מפרק לפי פרוטוקול
* הצגת התשובה למשתמש

רשימת אובייקטים: ממשק משתמש, הצפנה/פיענוח, תקשורת, פרוטוקול

שם היכולת: בקשת פתיחת סאשן עם קוד

מהות היכולת: בקשה מהשרת לקבל קוד זמני לפתיחת סאשן

אוסף יכולות:

* ממשק משתמש – מסך פתיחת סאשן
* קליטת לחיצה
* בניית הודעה לפי פרוטוקול
* הצפנה
* שליחה לשרת
* קבלת תשובה וקוד זמני מהשרת
* פענוח
* מפרק לפי פרוטוקול
* הצגת ממשק משתמש עם הקוד והזמן שיש למבקש תמיכה להתחבר

רשימת אובייקטים: ממשק משתמש- פתיחת סאשן, הצפנה/פיענוח, תקשורת, פרוטוקול, ממשק משתמש- הצגת קוד

שם היכולת: התחברות לסאשן עם קוד

מהות היכולת: בקשה מהשרת להתחבר לסאשן פתוח עם קוד מהתומך

אוסף יכולות:

* ממשק משתמש – מסך הכנסת קוד
* בדיקת קלט
* בניית הודעה לפי פרוטוקול
* הצפנה
* שליחה לשרת
* קבלת תשובה
* פענוח
* מפרק לפי פרוטוקול
* הצגת תשובה למשתמש

רשימת אובייקטים: ממשק משתמש, הצפנה/פיענוח, תקשורת, פרוטוקול

שם היכולת: יצירת חיבור חדש מול עוד משתמש

מהות היכולת: חיבור למשתמש אחר במערכת בשביל התחלת תמיכה

אוסף יכולות:

* קבלת נתונים משרת
* פענוח
* מפרק לפי פרוטוקול
* יצירת חיבור חדש עם המשתמש

רשימת אובייקטים: הצפנה/פיענוח, תקשורת, פרוטוקול, יצירת חיבור חדש

יכולות בצד מבקש תמיכה:

שם היכולת: העברת תמונת מסך נוכחי

מהות היכולת: להעביר תמונה של מסך נוכחי לתומך

אוסף יכולות:

* לקחת תמונה של המסך הנוכחי
* להצפין את המידע של התמונה
* בניית מידע של התמונה לפי פרוטוקול
* שליחה לתומך את המידע

רשימת אובייקטים: הצפנה/פיענוח, תקשורת, פרוטוקול

שם היכולת: העברת מיקום עכבר חדש

שם היכולת: קבלת מיקום עכבר חדש מהתומך והזזת עכבר

מהות היכולת: לקבל את מיקום העכבר שהתומך הזיז אליו את העכבר ולהזיז בהתאם

אוסף יכולות:

* לקבל מידע מהתומך
* פענוח
* פירוק הפרוטוקול והוצאת מיקום עכבר חדש
* בדיקת מיקום
* הזזת עכבר למיקום שהתקבל

רשימת אובייקטים: הצפנה/פיענוח, תקשורת, פרוטוקול, בדיקת מיקום, הזזת עכבר למיקום שהתקבל

שם היכולת: קבלת תו נלחץ מהתומך ולחיצה עליו

מהות היכולת: לקבל תו שהתומך לחץ עליו

אוסף יכולות:

* לקבל מידע מהתומך
* פענוח
* פירוק הפרוטוקול והוצאת ייצוג התו
* בדיקת ייצוג התו
* לחיצה על התו אשר התקבל מהתומך

רשימת אובייקטים: הצפנה/פיענוח, תקשורת, פרוטוקול, בדיקת ייצוג התו, לחיצה על התו שהתקבל מהתומך

שם היכולת: ניתוק קשר מצד מבקש תמיכה

מהות היכולת: לנתק את הקשר במידה ומשתמש ביקש

אוסף יכולות:

* לקבל מהמשתמש לחיצה אשר אומר שהוא מנתק את הקשר
* בניית הודעה שאומרת למבקש תמיכה שמתנתק הקשר לפי פרוטוקול
* הצפנה
* שליחה למבקש תמיכה את ההודעה
* בניית ממשק משתמש בהתאם לניתוק

רשימת אובייקטים: הצפנה/פיענוח , ממשק משתמש , תקשורת, פרוטוקול, קליטת לחיצה לניתוק מהמשתמש

שם היכולת: ניתוק קשר מצד תומך

מהות היכולת: לנתק את הקשר במידה וקיבלנו כך מהתומך

אוסף יכולות:

* קבלת מידע מהתומך
* פענוח
* פירוק לפי פרוטוקול
* בניית ממשק משתמש בהתאם לניתוק

רשימת אובייקטים: הצפנה/פיענוח , ממשק משתמש , תקשורת, פרוטוקול

יכולות בצד תומך:

שם היכולת: קבלת תמונת מסך מהמבקש תמיכה

מהות היכולת: לקבל את תמונת המסך הנוכחית להצגה למשתמש

אוסף יכולות:

* לקבל את מידע של התמונה מהמבקש תמיכה
* פענוח
* פירוק הפרוטוקול
* בניית התמונה
* הצגת התמונה למשתמש

רשימת אובייקטים: הצפנה/פיענוח, תקשורת, פרוטוקול, הצגת תמונה

שם היכולת: להעביר מיקום עכבר חדש למבקש תמיכה במידה והשתנה

מהות היכולת: להעביר מיקום עכבר חדש במידה והוא השתנה

אוסף יכולות:

* לקבל מיקום של עכבר נוכחי במידה והשתנה
* בדיקה מיקום
* בניית הודעה עם מיקום עכבר לפי פרוטוקול
* הצפנה
* שליחה למבקש תמיכה את ההודעה

רשימת אובייקטים: הצפנה/פיענוח ,בדיקת מיקום, קבלת מיקום עכבר במידה והשתנה, תקשורת, פרוטוקול

שם היכולת: העברת לחיצת מקלדת/עכבר למבקש תמיכה

מהות היכולת: להעביר לחיצת מקלדת/עכבר למבקש תמיכה במידה ונלחצה

אוסף יכולות:

* לקבל תו שנלחץ על מקלדת/עכבר במידה ונלחץ
* בניית הודעה עם ייצוג התו לפי פרוטוקול
* הצפנה
* שליחה למבקש תמיכה את ההודעה

רשימת אובייקטים: הצפנה/פיענוח , קבלת תו במקלדת\עכבר במידה ונלחץ , תקשורת, פרוטוקול

**פרק ד' - UELP - העיצוב**

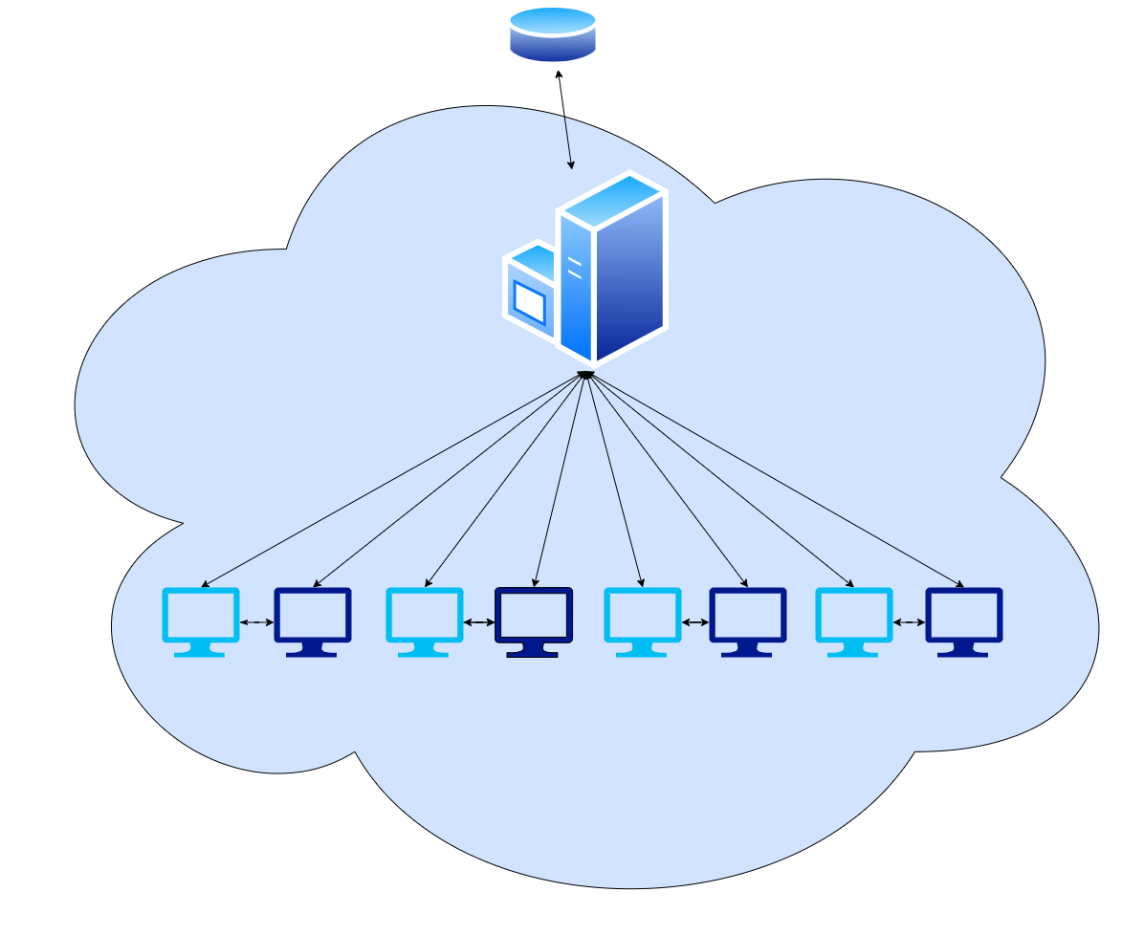
1. תיאור הארכיטקטורה של המערכת המוצעת

תיאור החומרה:

* מחשבים עם מערכת הפעלה windows
* רשת אינטרנט
* בסיס נתונים מבוסס SQL

רכיבים שונים והקשרים ביניהם:

* שרת מרובה לקוחות (מחבר בין תומכים ונתמכים במערכת בעזרת קוד זמני)
* לקוחות (**נתמכים ותומכים**, המתחברים למערכת ומתחברים אחד לשני בעזרת השרת ואז מעבירים מידע ביניהם.)
* כל הלקוחות מעבירים מידע דרך רשת האינטרנט:
* שרת - לקוח
* שרת – בסיס נתונים
* תומך – נתמך



1. תיאור הטכנולוגיה הרלוונטית

שפות תכנות: Python 3.7 ו-SQL לעבודה מול בסיס הנתונים

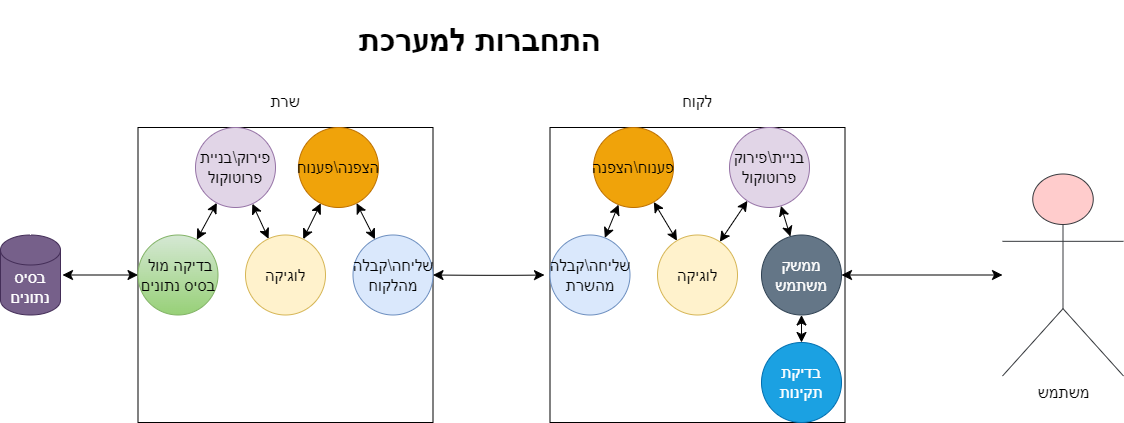
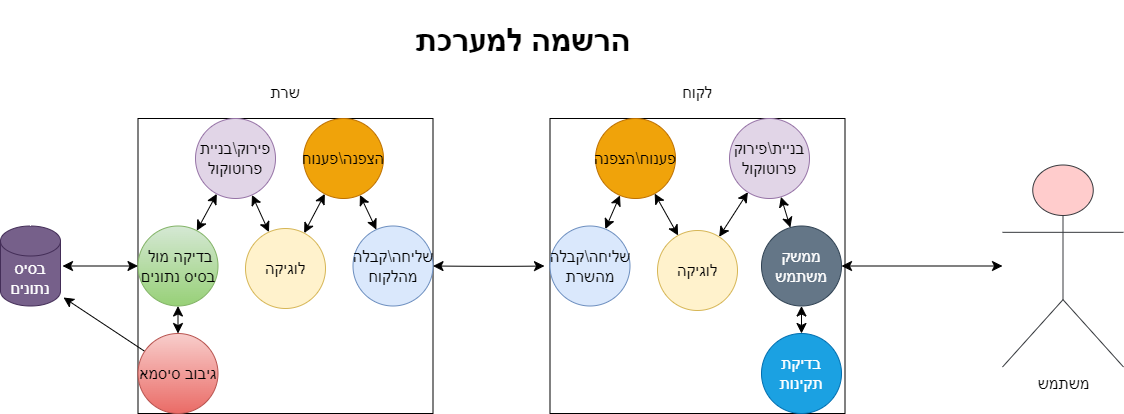
מערכת הפעלה: windows

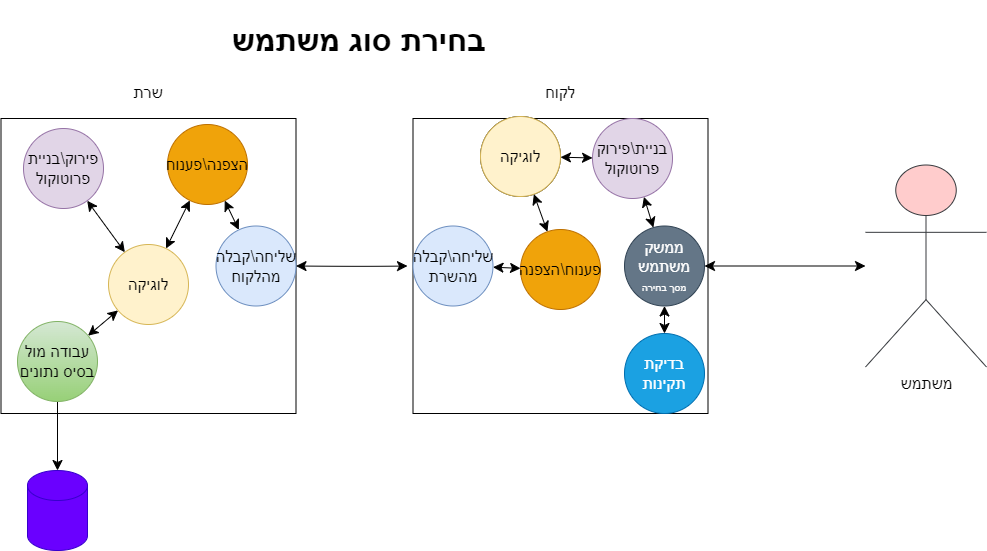
תקשורת: השרת הוא שרת מרובה לקוחות שמשתמש ב- sockets לתקשר עם הלקוחות, בנוסף ממומשת תקשורת peer2peer בין תומך לנתמך מבוססת סוקטים.

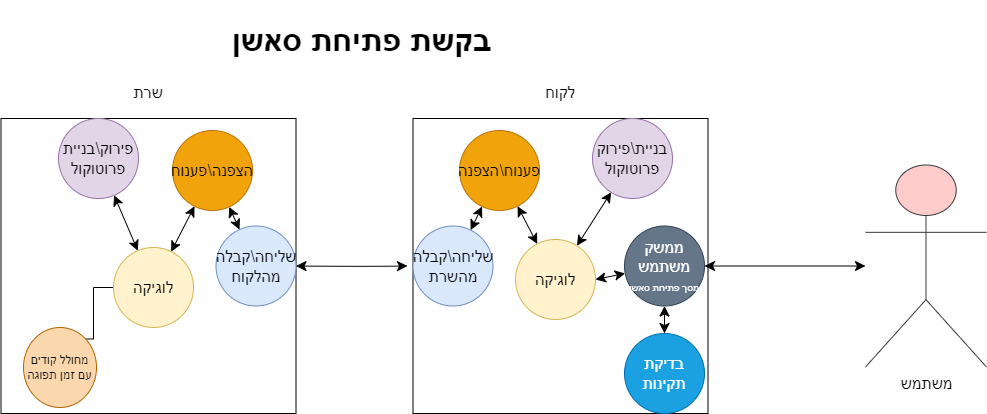
המידע יועבר ברשת בעזרת פרוטוקול TCP בשביל שהמידע יגיע מסודר, בנוסף המידע יהיה מוצפן לגמרי.

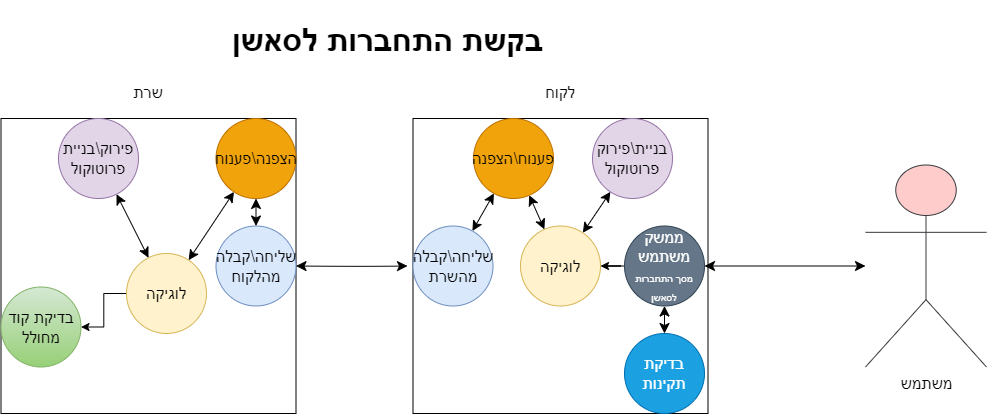
תחומי עניין: הצפנה, גרפיקה (באמצעות WXPython), עבודה עם בסיס נתונים (באמצעות SQL), שימוש ב thread-ים למיקבול פעולות במערכת, שימוש ב Process-ים למקבול תהליכים במערכת.

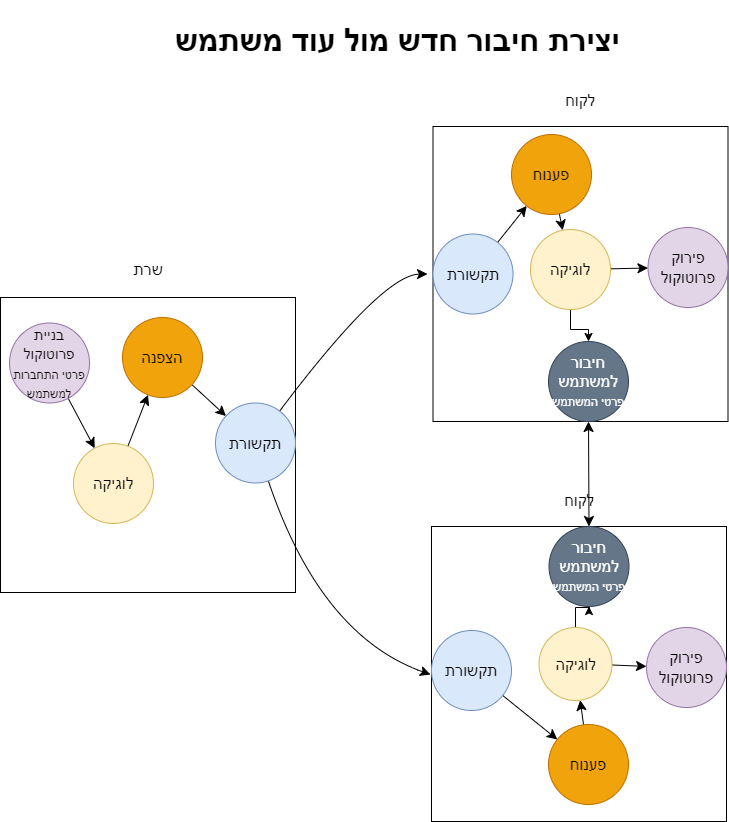
1. תיאור מודולים בהם נעשה שימוש

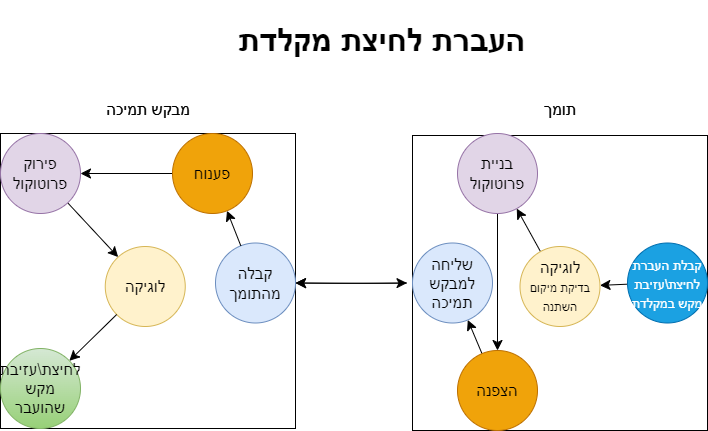


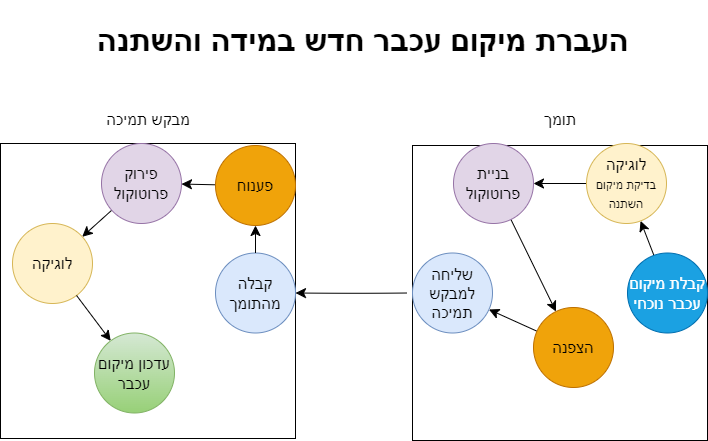
****

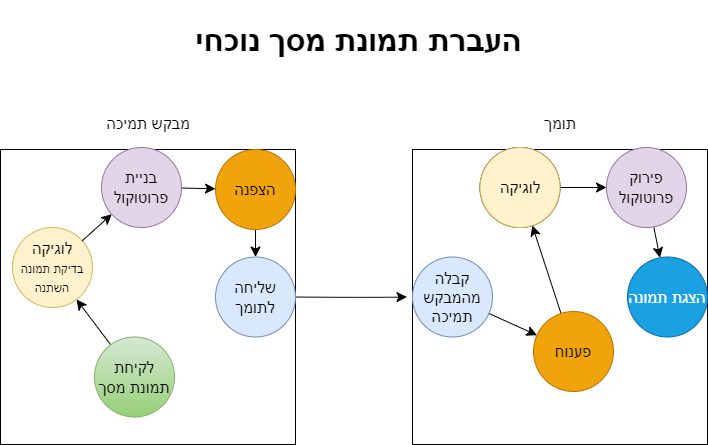
****

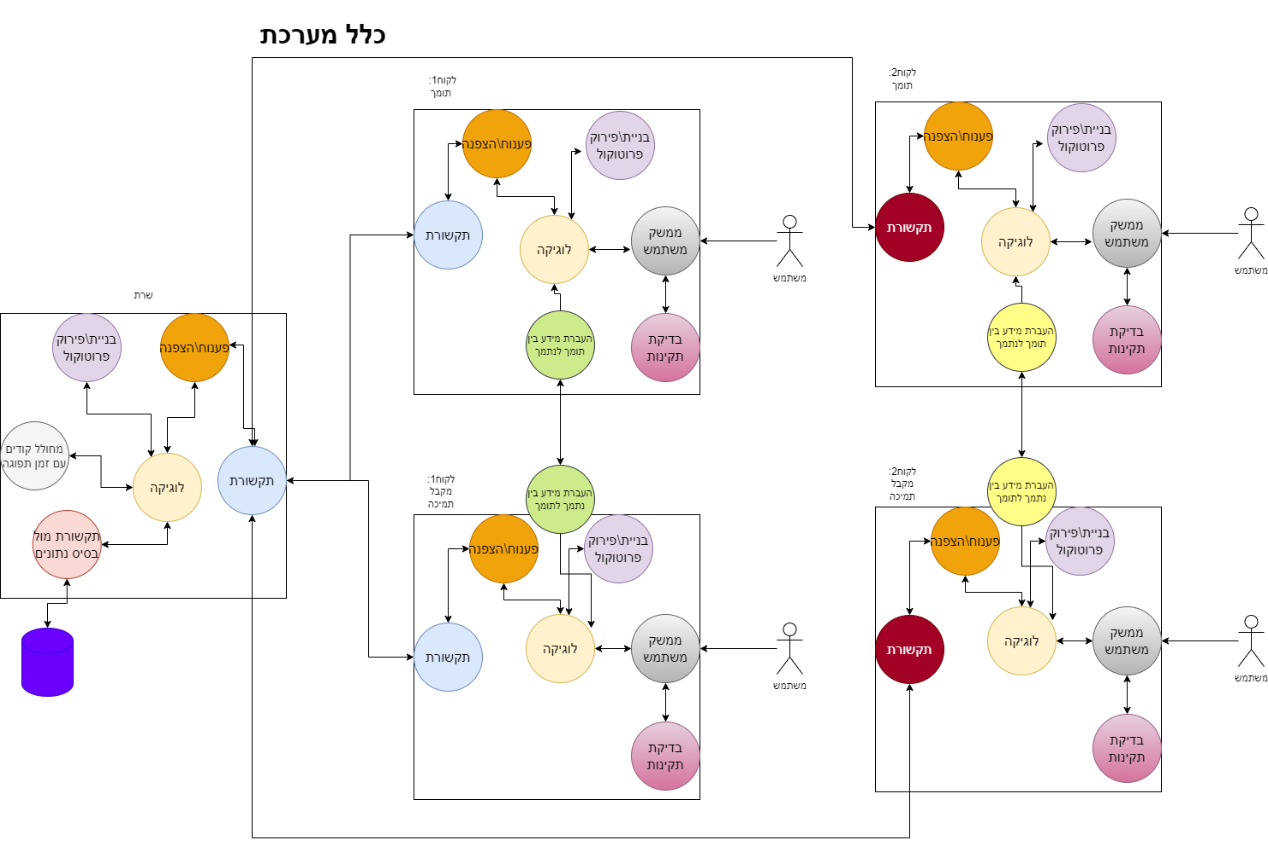
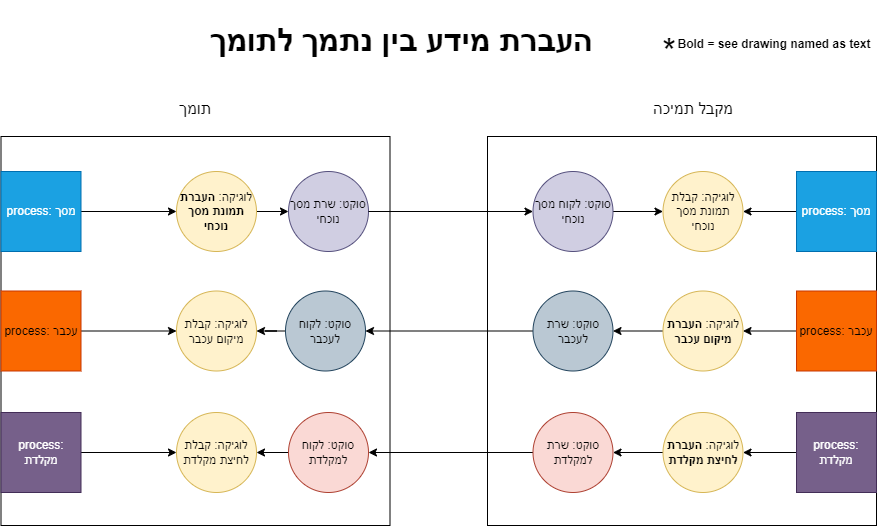
****

****

****



****

* ****

3.1 מודולים שאני מייבא:

| **שם המודול** | **לטובת** |
| --- | --- |
| queue | למימוש תור כמבנה נתונים |
| threading | לעבודה עם יותר מthread אחד במקביל |
| pynput | עבודה עם מעקב ושליטה על מקלדת ועכבר של מחשב (לחיצה על מקשים, הזזת ולחיצת עכבר ובדיקת מקשים ומיקומים של עכבר ומקלדת) |
| zlib | ביצוע פעולות compress וdecompress על מידע מועבר |
| PIL(ImageChops, ImageGrab, Image) | עבודה ועריכה עם תמונות, קבלת תמונות מהמחשב והדבקה על תמונות |
| pygame | להצגת מסך של הנתמך לתומך |
| cryptography | למימוש הצפנת rsa |
| multiprocessing | לעבודה והפעלת כמה פרוססים מפרוסס רץ |
| time | לעצירת פרוסס רץ לזמן מסוים |
| getnode (from UUid) | להשגת כתובת mac של המחשב |
| wx | יצירת גרפיקה של הפרויקט ועבודה עם גרפיקה ויצירת מסכים |
| pub (from pubsub) | עבודה לתקשורת בין threads |
| base64 | לencoding לbase64 ומימוש הצפנות |
| secrets | ליצירת רצף סטרינג רנדומלי למפתח וקוד סאשן |
| socket | עבודה של תקשורת בין שרת ללקוחות |
| sys | לעבודה עם פעולות של מערכת ההפעלה כמו יציאה מתוכנית |
| sqlite3 | לעבודה עם בסיס נתונים SQL |
| re | לחיפוש סוג מסוים של ערכים בתוך סטרינג למשל uppercase letters |
| pyperclip | להכניס ערך מסוים לclipboard של המחשב |
| wx.adv | לעבודה עם פעולות מתקדמות של גרפיקה |
| getmacbyip (from Scapy.layers.l2) | לשליחת בקשת arp לכתובת ip מסויימת |
| select | למימוש שרת מרובה לקוחות |
| hashlib | ליצירת ערך hash sha256 של סטרינגים. בשביל לשמור סיסמאות וכתובות mac בבסיס נתונים והשוואה עליהם |
| Random, AES, pad, unpad (from Cryptodome) | ליצירת הצפנת AES. מפתח הצפנה, הצפנה ופיענוח. |

3.2 מודולים שאני מייצר:

|  |
| --- |
| Server\_protocol |
| אוסף הפעולות הנדרשות בצד שרת לבנייה ופירוק הודעות ע"פ הפרוטוקול |

**אין למחלקה זו תכונות**

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| unpackData | הפעולה מקבלת מידע  מהתקשורת | הפעולה מפרקת את המידע שהתקבל לפרמטרים וה- opcode שלו.  מחזירה את הקוד הפעולה ואת השדות השייכים לקוד הפעולה.  (opcode, [parameters]) |  |
| pack\_login\_ans | הפעולה מקבלת סטרינג המייצג את סטטוס הצלחת ההתחברות למערכת | הפעולה מחזירה הודעה לפי פרוטוקול.    000 – הצליח  001 – נכשל  002- משתמש כבר מחובר  (000) |  |
| pack\_signup\_ans | הפעולה מקבלת סטרינג המייצג את סטטוס הצלחת הרשמה למערכת | הפעולה מחזירה הודעה לפי פרוטוקול.  010 – הצליח  011 – נכשל  (010) |  |
| pack\_typeuser\_ans | הפעולה מקבלת משתנה בוליאני המייצג אם בחירת המשתמש אושר או לא אושר | הפעולה מחזירה הודעה לפי פרוטוקול.  020 – הצליח  021 – נכשל  (020) |  |
| pack\_getcode\_ans | הפעולה מקבלת סטרינג המייצג את הקוד שמבקש המשתמש | הפעולה מחזירה הודעה לפי פרוטוקול.    0378634 – הצליח + קוד  (0378634) |  |
| pack\_expired\_code | הפעולה לא מקבלת כלום | הפעולה מחזירה הודעה לפי הפרוטוקול הודעת קוד לא תקף.  032- קוד לא תקף יותר |  |
| pack\_code\_ans | הפעולה מקבלת משתנה בוליאני המייצג את סטטוס אישור הקוד שהגיע מהלקוח | הפעולה מחזירה הודעה לפי פרוטוקול.    040 – עבר  041 – נכשל  (040) |  |
| pack\_con\_data | הפעולה מקבלת את הIP של משתמש והסוג משתמש  IP-  '127.0.0.1'  userType- “A” | הפעולה מחזירה הודעה לפי פרוטוקול.  A05127.0.0.1 – מידע על משתמש אחר  (A05127.0.0.1) |  |

|  |
| --- |
| Client\_protocol |
| אוסף הפעולות הנדרשות בצד שרת לבנייה ופירוק הודעות ע"פ הפרוטוקול |

**אין למחלקה זו תכונות**

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| unpackData | הפעולה מקבלת מידע מהתקשורת | הפעולה מפרקת את המידע שהתקבל לפרמטרים וה- opcode שלו.  מחזירה את הקוד הפעולה ואת השדות השייכים לקוד הפעולה.  (opcode, [parameters]) |  |
| pack\_mac\_addr | הפעולה מקבלת את הכתובתmac של המשתמש  macAddr- 00-B0-D0-63-C2-26 | הפעולה מחזירה הודעה לפי הפרוטוקול.  Opcode + mac address-  00-B0-D0-63-C2-2606 |  |
| pack\_login\_info | הפעולה מקבלת את פרטי ההתחברות של המשתמש: שם משתמש וסיסמא  loginInfo-  שם משתמש: itamar  סיסמא:  Itamar6 | הפעולה מחזירה הודעה לפי הפרוטוקול.  Opcode + login info-  006ItamarItamar6 |  |
| pack\_signup\_info | הפעולה מקבלת את פרטי ההרשמה של המשתמש.  signupinfo-ItamarItamar6 | הפעולה מחזירה הודעה לפי הפרוטוקול.  Opcode + signup info-  016ItamarItamar6 |  |
| pack\_type\_user | הפעולה מקבלת את סוג המשתמש שנבחר  typeUser-A/H | הפעולה מחזירה הודעה לפי הפרוטוקול.  Opcode + user type-  020 |  |
| pack\_code | הפעולה מקבלת קוד שהמשתמש נתמך הכניס  Code-a63h2j23 | הפעולה מחזירה הודעה לפי הפרוטוקול.  Opcode + code-  a63h2j2304 |  |

|  |
| --- |
| Server\_ comm |
| מחלקה ליצירת עצם לעבודה עם שרת מרובה לקוחות |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| recv\_q | Queue שמכניס לתוכו את כל ההודעות המתקבלות מהלקוחות ללוגיקה של השרת. |
| bindIP | הכתובת שממנה מקשיב השרת |
| Port | ה port שבו השרת רץ במחשב |
| socket | הsocket שממנו רץ השרת  (לא מתקבל מהמשתמש) |
| open\_clients | מילון של כל הsocket, ip שמחוברים לשרת וencryption object  Open\_clients[socket]=[ip, AESEncryption\_object]  (לא מתקבל מהמשתמש) |
| is\_running | משתנה בוליאני שאומר אם השרת כרגע רץ או לא  (לא מתקבל מהמשתמש) |
| RSAobject | עצם למימוש הצפנה א-סימטרית |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| \_\_init\_\_ | הפעולה מקבלת מידע ליצור עצם מסוג Server\_comm  Receive:  recv\_q: תור של הודעות  שדרכו יעבור ללוגיקה של השרת.  port: מספר פורט שעליו ירוץ השרת,  bindIP: כתובת ה-ip שהשרת מקשיב לו | הפעולה מאתחלת את תכונות המחלקה ומפעילה את \_main\_loop כ-thread שבו ירוץ השרת מרובה לקוחות |  |
| \_main\_loop | הפעולה לא מקבלת כלום | הפעולה יוצרת את השרת ומחכה שלקוחות יתחברו, (במידה והפורט הוא לא פורט ראשי מוודאים שהלקוח הוא p2pIP )כאשר לקוח חדש מתחבר לשרת מחליפים איתו מפתח באמצעות הפעולה \_get\_shared\_key (כthread)  ולאחר מכן מכל מידע שהלקוח מעביר לשרת היא מפענחת ומכניסה לrecv\_q עם הip של הלקוח במידה ומדובר בport המסך היא קוראת ל \_recvImage |  |
| \_get\_shared\_key | הפעולה מקבלת את ה ip של הלקוח החדש  Receive:  clientIP- הכתובת ip של הלקוח החדש שצריך לבנות איתו מפתח הצפנה משותף  curSocket-הסוקט של הלקוח שיש ליצור איתו מפתח | הפעולה שולחת ללקוח את המפתח הציבורי של ההצפנת RSA של השרת ובעזרת המפתח הלקוח שולח לשרת את המפתח המשותף שלהם מוצפן, הפעולה מכניסה לתוך מילון keys עצם הצפנה של AES של המפתח המשותף של הלקוח והשרת עם הip של הלקוח והip של הלקוח כהמפתח של המילון |  |
| \_disconnect\_client | הפעולה מקבלת socket של לקוח שמחובר לשרת ומנתקת אותו  Receive:  clientSocket- הסוקט של הלקוח שיש לנתק מהמערכת | הפעולה מוציאה את הלקוח משני המילונים שלה (open\_clients ו-keys)  וסוגרת ללקוח את הsocket  אם הלקוח הוא לקוח מסך מקלדת או עכבר סוגרים גם את השרת מכיוון שיש רק לקוח אחד. |  |
| \_find\_socket\_by\_ip | הפעולה מקבלת ip של לקוח במערכת ומחזירה את הsocket שלו  Receive:  findip- כתובת של לקוח במערכת | הפעולה מחזירה את הסוקט של הלקוח שהוכנס הip שלו  Return:  findSocket- הסוקט שחופש על ידי הכתובת שהתקבלה |  |
| send | הפעולה מקבלת ip של לקוח והודעה ושולחת את ההודעה המוצפנת ללקוח  Receive:  clientIP- כתובת של לקוח במערכת  msg- הודעה לא מוצפנת | הפעולה מצפינה את ההודעה שהתקבלה לפי מפתח ההצפנה (AES) ושולחת את האורך של ההודעה וההודעה שניהם דרך הsocket של הלקוח הזה |  |
| isRunning | הפעולה לא מקבלת כלום | הפעולה מחזירה את משתנה is\_running לבדוק אם השרת פועל  Return:  is\_running- משתנה במחלקה שאומר אם השרת שלה רץ או לא  (True) |  |
| recvImage\_ | הפעולה מקבלת socket ו-data | הפעולה מפענחת את המידע  מפרקת לפי פרוטוקול:  left, top, right, bottom, imgLen == server\_protocol.unpack(data)  בלולאה קולטת את המידע ע"פ imgLen  ומכניסה לתור ההודעות באופן הבא:  (ip, [top, left, bottom, right, bytearray]) | רלוונטי אך ורק לסוקט של התמונה |
| Close\_server | הפעולה לא מקבלת כלום | הפעולה מפסיקה את השרת הנוכחי, והופכת את משתנה is\_running לFalse |  |
| closeClient | הפעולה מקבלת כתובת ip של לקוח שהשרת רוצה לסגור  Receive:  clientIP- כתובת של לקוח במערכת | הפעולה מנתקת את הסוקט של המשתמש עם הip שהתקבל ומוציאה אותו מרשימות השרת בעזרת \_handle\_disconnect\_client |  |

|  |
| --- |
| Client\_ comm |
| מחלקה לעבודה ותקשורת עם שרת המערכת |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| serverIP | הכתובת ip של השרת |
| port | הport שעליו עובד השרת |
| recv\_q | תור של הודעות המתקבלות מהשרת |
| socket | הsocket של הלקוח |
| is\_running | משתנה בוליאני שאומר אם הלקוח רץ או לא |
| sharedKey | מפתח הצפנה משותף (AES) של הלקוח והשרת |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| \_\_init\_\_ | הפעולה מקבלת מידע ליצור עצם מסוג Client\_comm  Receive:  serverIP- כתובת הip של השרת  port- הפורט שעליו עובד השרת  recv\_q- תור של הודעות שדרכו יעבור ללוגיקה של השרת.  port: מספר פורט שעליו ירוץ השרת | הפעולה מפעילה את \_main\_loop כ-thread שבו ירוץ הלקוח |  |
| \_main\_loop | הפעולה לא מקבלת כלום | הפעולה מתחברת לשרת, ובונה מפתח הצפנה משותף לשרת וללקוח בעזרת \_get\_shared\_key  ולאחר מכן כל מידע שהשרת מעביר ללקוח היא מפענחת ומכניסה לrecv\_q |  |
| \_get\_shared\_key | הפעולה לא מקבלת כלום | הפעולה מקבלת מהשרת את המפתח הפומבי שלו ומשתמשת בו להצפין מפתח רנדומלי הלקוח משתמש במפתח זה ליצור עצם AES להצפנה ופיענוח |  |
| send | הפעולה מקבלת הודעה שהלקוח רוצה לשלוח לשרת  Receive:  msg– הודעה שהלקוח שולח לשרת ולא מוצפנת | הפעולה מצפינה את ההודעה בעזרת המפתח המשותף ושולחת את אורכה של ההודעה המוצפנת ואת ההודעה המוצפנת לשרת |  |
| sendImage | הפעולה מקבלת מידע על התמונה (data) ואת המידע של התמונה (imageData) | הפעולה מצפינה את הdata ושולחת לפי פרוטוקול ושולחת את הimageData מיד אחר כך (לא מוצפן) |  |
| close | הפעולה לא מקבלת כלום | הפעולה סוגרת ומנתקת את הsocket של הלקוח והופכת את משתנה is\_running לFalse |  |
| exchangeStatus | הפעולה לא מקבלת כלום | הפעולה מחזירה אמת אם נוצר כבר מפתח הצפנה משותף בין לקוח ושרת ושקר אם עדיין לא נוצר |  |

|  |
| --- |
| Database\_comm |
| מחלקה לעבודה ותקשורת עם בסיס נתונים |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| dbName | השם של בסיס הנתונים (Uelp\_db) שאנו רוצים להתחבר אליו |
| conn | חיבור לבסיס נתונים SQL |
| curr | מצביע לטבלה מסוימת |
| loginTableName | השם של הטבלה users שבה מאוחסן כל הפרטי כניסה של המשתמשים של המערכת |
| blacklistTableName | השם של הטבלה macs שבה מאוחסן כל כתובות ה mac של משתמשים שחסומים מלהיות תומכים |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| \_\_init\_\_ | הפעולה לא מקבלת כלום | הפעולה מאתחלת את תכונות המחלקה וקוראת לפעולה \_create\_db |  |
| \_create\_db | הפעולה לא מקבלת כלום | הפעולה מתחברת לבסיס נתונים ויוצרת את הטבלאות המבוקשות במידה והן כבר לא קיימות |  |
| checkUser­­\_ | הפעולה מקבלת  username- שם משתמש | הפעולה בודקת אם קיים משתמש כזה בטבלתusers  הפעולה מחזירה בהתאמה אמת או שקר אם המשתמש קיים במערכת או לא |  |
| addUser | הפעולה מקבלת  username- שם משתמש שרוצה להירשם למערכת  password- סיסמא של המשתמש שרוצה להרשם עם לחשבון חדש | במידה והשם משתמש לא קיים (\_checkUser) במערכת הפעולה מוסיפה אותו עם ה hash של הסיסמא שלו לתוך הטבלהusers ומחזירה אמת אם הצליחה להוסיף אותו אחרת שקר. |  |
| checkPassword | הפעולה מקבלת שם משתמש וסיסמא | במידה והשם משתמש קיים (\_checkUser) נבדקת ה hash של הסיסמא שהתקבלה מול ה hash של הבסיס נתונים.  הפעולה מחזירה אמת במידה והם שווים אחרת שקר |  |
| macExists | הפעולה מקבלת כתובת mac | הפעולה מחזירה אמת במידה והhash של הmac קיים בטבלה macs אחרת שקר |  |
| addBlackMac | הפעולה מקבלת כתובת mac | הפעולה מוסיפה את ה hash של ה mac לטבלה macs אם הוא לא קיים (\_macExists) |  |

|  |
| --- |
| AES\_hash\_cipher |
| מחלקה להצפנה ופענוח סימטרי ועבודה עם hash |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| bs | Block size שהמחלקה תצפין ותפענח בו |
| key | מפתח הצפנה שבעזרתו נצפין ונפענח מידע |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| \_\_init\_\_ | הפעולה מקבלת מידע ליצור עצם חדש של קריפטוגרפיה  Receive:  Key- מפתח להצפנה סימטרית | הפעולה יוצרת עצם חדש להצפנה פיענוח |  |
| encrypt | הפעולה מקבלת מידע בטיפוס בינארי ומצפינה אותו  Receive:  raw- מידע לא מוצפן | הפעולה מצפינה את המידע בעזרת הblock size ומפתח ההצפנה שהתקבל ומחזירה את המידע המוצפן בטיפוס בינארי  Return:  encrypted- מידע מוצפן |  |
| decrypt | הפעולה מקבלת מידע מוצפן מטיפוס בינארי ומפענחת אותו  Recieve:  enc- מידע מוצפן | הפעולה מפענחת את המידע בעזרת ה block size והמפתח ההצפנה בהתקבל ומחזירה את המידע המפוענח בטיפוס בינארי.  Return:  data- המידע המפוענח |  |
| hash | הפעולה מקבלת מידע  Recieve:  Data- מידע שרוצים להפעיל עליו פונקציית hash | הפעולה מפעילה על המידע פונקציית hash ומחזירה את ה hash של מידע זה בטיפוס בינארי.  Return-  hashedData- המידע שקיבלנו לאחר שהפעילו עליו פונקציית hash. | פעולה סטטית |

|  |
| --- |
| RSA\_cipher |
| מחלקה לעבודה עם הצפנה א-סימטרית |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| publicKey | מפתח הצפנה ציבורי (שמור בבינארי) |
| privateKey | מפתח פיענוח פרטי (שמור בבינארי) |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| \_\_init\_\_ | הפעולה לא מקבלת כלום | הפעולה מאתחלת את שדות המחלקה ויוצרת בהם מפתח ציבורי ופרטי של rsa ויוצרת עצם RSA חדש להצפנת מידע ופיענוח |  |
| encrypt | הפעולה מקבלת מידע ומפתח הצפנה ציבורי כString  Receive:  data- מידע להצפין  publicKey- מפתח הצפנה ציבורי שבעזרתו מצפינים מידע | הפעולה מצפינה את המידע שהתקבל בעזרת המפתח הציבורי ומחזירה את המידע המוצפן.  Return:  enc- מידע מוצפן | הפעולה היא פעולה סטטית |
| getStringKey | הפעולה אל מקבלת כלום | הפעולה מחזירה את מפתח ההצפנה הציבורי של העצם כstring  return:  stringKey- מפתח הצפנה ציבורי כstring |  |
| decrypt | הפעולה מקבלת מידע מוצפן מטיפוס בינארי  Receive:  data- מידע לפענח | הפעולה מפענחת את המידע שהוצפן בעזרת המפתח הציבורי ומחזירה את המידע המפוענח בטיפוס בינארי.  Return:  decData- המידע המפוענח |  |

|  |
| --- |
| Session\_codes |
| מחלקה לעבודה עם קודים של סאשנים |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| codes | מילון של כל הקודים שנפתחו, זמן הוספת הקוד והIP של מי שיצר אותם.  {'192.3.4.203', (123456, 17:56:59.23} |
| expired\_q | תור שמכניסים לתוכו ip של לקוח שקודו כבר לא תקף |
| cleanseTime | כל כמה זמן קוד פג תוקף |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| \_\_init\_\_ | הפעולה מקבלת את התור להודעות ליצור עצם חדש | הפעולה מאתחלת את כל תכונות המחלקה לפי הצורך ומריצה את \_remove\_expiredכthread |  |
| \_remove\_expired | הפעולה לא מקבלת כלום | הפעולה עוברת על מילון הקודים כל 2 שניות ובודקת אם יש קודים שזמנם לא תקף, במידה וכן מוציאה אותם מהמילון ומכניסה לexpired\_q את הip של הקוד |  |
| createCode | הפעולה מקבלת ip של לקוח | הפעולה יוצרת קוד רנדומלי באורך קבוע חדש מחזירה אותו ומוסיפה אותו למילון |  |
| checkCode | הפעולה מקבלת קוד באורך קבוע | הפעולה מחזירה אמת במידה והקוד קיים במילון ותקף ושקר אחרת |  |

|  |
| --- |
| Server\_logic |
| מחלקה לבניית הלוגיקה של השרת |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| server | שרת מרובה לקוחות לתקשורת |
| recv\_q | תור לקבלת הודעות מהלקוחות |
| expired\_q | תור לקבלת כתובות של משתמש שהקודים שלהם פג תוקפם |
| codes | עצם לעבודה עם קודים זמניים של לקוחות (גלובלי) |
| port | הפורט שעליו ירוץ השרת |
| commands | מילון לכל הפעולות הכלליות שבו לכל opcode יש שם של פעולה מתאימה |
| users | מילון שמצמיד בין כתובות ip כתובות mac של לקוחות הסוג של הלקוח והשם משתמש |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| main | הפעולה לא מקבלת כלום | הפעולה יוצרת תור recv\_q וexpired\_q, יוצרת שרת server ועצם קודים clientCodes  מפעילה את handleMsgs כthread  מפעילה handleExpired כthread |  |
| handleMsgs | הפעולה מקבלת את עצם השרת והתור הודעות.  receive:  server- עצם השרת  recv\_q- תור ההודעות | הפעולה יוצרת מופע של DB\_comm (אי אפשר להעביר בסיס נתונים ל thread) ועוברת בלולאה אינסופית מוציאה הודעות מהתור recv\_q במבנה של tuple הכולל (ip,data)  מפרקת לפי פרוטוקול  (opcode, [params]) ומפעילה פעולה לפי הopcode :  commands[opcode](clientIP, DB, server, params) |  |
| handleExpired | הפעולה מקבלת את עצם השרת ותור הפג תוקף  server- עצם השרת  expired\_q- תור פג תוקף | הפעולה עוברת בלולאה אינסופית על תור הפג תוקף expired\_q ומוציאה כתובות ip  ושולחת הודעות לפי פרוטוקול ללקוחות  server.send(ip, server\_protocol.pack\_expired\_code()) |  |
| handle\_mac\_addr | הפעולה מקבלת את הip של הלקוח, העצם של הבסיס נתונים, עצם השרת ורשימה של פרמטרים שבתוכה יש את כתובת הmac של המשתמש.  clientIP- הכתובת של הלקוח  DB- עצם הבסיס נתונים  server- עצם השרת  params- רשימת פרמטרים | הפעולה מוציאה את כתובת הmac מהפרמטרים ומוסיפה את כתובת הip של הלקוח וכתובת הmac שלו למילון users  (users[clientIP] = (mac\_addr, None, None) |  |
| handle\_login | הפעולה מקבלת את הip של הלקוח, העצם של הבסיס נתונים, עצם השרת ורשימה של פרמטרים שבתוכה יש את שם משתמש והסיסמא  clientIP- הכתובת של הלקוח  DB- עצם הבסיס נתונים  server- עצם השרת  params- רשימת פרמטרים | הפעולה מוציאה את השם משתמש וסיסמא מהרשימה ובודקת אותם לפי הבסיס נתונים במידה והם תואמים מחזיר אמת אחרת שקר ושולחת הודעה בחזרה למשתמש לפי פרוטוקול ומוסיפה למילון משתמש את השם משתמש אם תואם  userOk = DB.checkPassword(username, password)  server.send(clientIP, pack\_login\_ans(userOk)) |  |
| handle\_signup | הפעולה מקבלת את הip של הלקוח, העצם של הבסיס נתונים, עצם השרת ורשימה של פרמטרים שבתוכה יש את שם משתמש והסיסמא  clientIP- הכתובת של הלקוח  DB- עצם הבסיס נתונים  server- עצם השרת  params- רשימת פרמטרים | הפעולה מוציאה את השם משתמש וסיסמא מהרשימה ובודקת אותם לפי הבסיס נתונים ומחזיר הודעה למשתמש ומוסיף למילון משתמשים בהתאם  newuserOk = DB.addUser(username, password)  server.send(clientIP, pack\_signup\_ans(newuserOk)) |  |
| check\_logged\_in | הפעולה מקבלת שם משתמש של משתמש | הפעולה מחזירה הפעולה בודקת אם השם משתמש נמצא בתוך מילון המשתמשים, ומחזירה אמת או שקר בהתאם. |  |
| handle\_typeUser | הפעולה מקבלת את הip של הלקוח, העצם של הבסיס נתונים, עצם השרת ורשימה של פרמטרים שבתוכה יש את סוג המשתמש (0/1(  clientIP- הכתובת של הלקוח  DB- עצם הבסיס נתונים  server- עצם השרת  params- רשימת פרמטרים | הפעולה מוציאה את בחירת סוג המשתמש מהפרמטרים בודקים אם הכתובת mac שהתקבל מהלקוח תואמת לכתובת הmac שמגיע מבקשת ה arp ששלחנו ללקוח. אם המשתמש רוצה להיות תומך, בודקים שכתובת הmac שלו לא בטבלת הblacklist בבסיס נתונים  במידה ולא נשמור את סוג המשתמש מוצמד עם הmac שלו,  שולחים הודעה מתאימה  If mac\_from\_arp == users[clientIP][0]:  If typeUser == “0”:  If not db.macExists(users[clientIP[0]):  Users[clientIP][1] = “H”  Typeuser\_ans = True  Else:  typeUser\_ans = False  elif typeUser == “1”:  users[clientIP][1] = “A”  typeUser\_ans = True |  |
| handle\_getCode | הפעולה מקבלת את הip של הלקוח, העצם של הבסיס נתונים, עצם השרת ורשימה של פרמטרים  clientIP- הכתובת של הלקוח  DB- עצם הבסיס נתונים  server- עצם השרת  params- רשימת פרמטרים | הפעולה בודקת את רשות המשתמש לקבלת קוד במידה ויכול שולחת לו קוד חדש לפי הפרוטוקול  if users[clientIP][1] == "H":  server.send(clientIP, pack\_getCode\_ans(True, codes.createCode(clientIP)) |  |
| handle\_codeCheck | הפעולה מקבלת את הip של הלקוח, העצם של הבסיס נתונים, עצם השרת ורשימה של פרמטרים עם הקוד  clientIP- הכתובת של הלקוח  DB- עצם הבסיס נתונים  server- עצם השרת  params- רשימת פרמטרים | הפעולה מוציאה את הקוד שהתקבל מרשימת הפרמטרים ובודקת אותו ושולחת הודעה חזרה למשתמש אם הקוד תואם. אם הקוד תואם תשלח לשני המשתמשים שהתחברו לסאשן את םרטי ההתחברות.  userCode = params[0]  if users[clientIP][1] == "A":  server.send(clientIP, pack\_code\_ans(codes.checkCode(userCode)) |  |
| disconnectClient | הפעולה מקבלת את הip של הלקוח שרוצים לנתק ועצם השרת | הפעולה מוציאה את המשתמש מרשימת users וקוראת ל closeClient לנתק אותו מהשרת |  |

|  |
| --- |
| Client\_logic |
| מחלקה לבניית הלוגיקה של הלקוח |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| client | לקוח לתקשורת עם שרת (client\_comm) |
| ip | כתובת שעליה רץ השרת |
| recv\_q | תור לקבלת הודעות מהשרת |
| port | הפורט שעליו רץ השרת |
| commands | מילון לכל הפעולות הכלליות שבו לכל opcode יש שם של פעולה מתאימה |
| allGraphics | עצם גרפיקה ראשי לעבודה עם גרפיקה |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| main | הפעולה לא מקבלת כלום | הפעולה יוצרת תור recv\_q יוצרת לקוח client עם הip והport  מפעילה את handleMsgs כthread  ויוצרת עצם גרפיקה ומעפילה אותו בthread הראשי |  |
| handleMsgs | הפעולה מקבלת את עצם הלקוח והתור הודעות.  receive:  client- עצם הלקוח  recv\_q- תור ההודעות | הפעולה מוציאה מהתור בלולאה אינסופית על כל הtuple שהתקבלו.  מפרקת לפי פרוטוקול  (opcode, [params]) ומפעילה פעולה לפי הopcode :  commands[opcode](client, params) |  |
| send\_mac | הפעולה מקבלת את עצם הלקוח | הפעולה משיגה את הmac של המחשב ושולחת אותו לשרת |  |
| handle\_login\_ans | הפעולה מקבלת רשימה של פרמטרים  שבהם את אישור ההתחברות  params- רשימת פרמטרים | הפעולה מוציאה מהרשימה את אישור ההתחברות ומעבירה אותו לגרפיקה  wx.CallAfter(pub.sendMessage, "login\_ans", ans=login\_ans) |  |
| handle\_signup\_ans | רשימה של פרמטרים  שבהם את אישור ההרשמה  params- רשימת פרמטרים | הפעולה מוציאה מהרשימה את אישור ההרשמה ומעבירה אותו לגרפיקה  wx.CallAfter(pub.sendMessage, "signup\_ans", ans=signup\_ans) |  |
| handle\_typeUser\_ans | מקבלת רשימה של פרמטרים  שבהם את אישור בחירת המשתמש  params- רשימת פרמטרים | הפעולה מוציאה מהרשימה את אישור בחירת המשתמש ומעבירה אותו לגרפיקה  wx.CallAfter(pub.sendMessage, "typeUser\_ans", ans=typeUser\_ans) |  |
| handle\_getCode\_ans | מקבלת רשימה של פרמטרים  שבהם את הקוד  params- רשימת פרמטרים | הפעולה מוציאה מהרשימה את תשובת הבקשת קוד ומעבירה אותו לגרפיקה  wx.CallAfter(pub.sendMessage, "gotten\_code", ans=getCode\_ans) |  |
| handle\_code\_ans | מקבלת רשימה של פרמטרים בהם האם הקוד שנשלח תקין או לא  params- רשימת פרמטרים | הפעולה מוציאה מרשימת הפרמטרים את אישור הקוד שנשלח ומעבירה אותו לגרפיקה  wx.CallAfter(pub.sendMessage, "code\_ans", ans=code\_ans) |  |
| handle\_conData | מקבלת רשימה של פרמטרים בהם הכתובת של הלקוח השני  params- רשימת פרמטרים | הפעולה מוציאה מרשימת הפרמטרים את הip של הלקוח האחר והסוג משתמש של המשתמש הפעולה יוצרת תור הודעות להעברת הודעות בין הלוגיקה לפרוסס המקלדת, process לעכבר, מקלדת ומסך בהתאמה לפי סוג המשתמש. הפעולה מחכה 5 שניות לסיום מסך ההתחברות בגרפיקה ומפעילה את הפרוססים.  הפעולה גם תקרא לפעולה check\_closed לאחר הפעלת הפרוססים.  if my\_user\_Type == "H":  mouse = multiprocessing.Process(target=Helper\_logic.main\_Helper, args=(otherIP, 2001, None,))  keyboard = multiprocessing.Process(target=Helper\_logic.main\_Helper, args=(otherIP, 2002, close\_queue,))  screen = multiprocessing.Process(target=Helper\_screen\_logic.main\_Helper\_screen, args=(otherIP,)) wx.CallAfter(pub.sendMessage, "connecting\_session") |  |
| check\_closed | הפעולה מקבלת את תור ההודעות בין הלוגיקה ופרוסס המקלדת, פרוסס העכבר מקלדת ומסך.  (close\_queue, mouse, screen, keyboard) | הפעולה מוציאה מידע מתור ההודעות, אם מקבלת הודעה close (ההודעה היחידה שעוברת בתור) היא סוגרת את כל הפרוססים של השליטה ויוצאת מהפעולה |  |

|  |
| --- |
| Helper\_protocol |
| אוסף הפעולות הנדרשות בצד תומך לבנייה ופירוק הודעות ע"פ הפרוטוקול |

**אין למחלקה זו תכונות**

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| unpackData | הפעולה מקבלת מידע  מהתקשורת | הפעולה מפרקת את המידע שהתקבל לפרמטרים וה- opcode שלו לפי הפרוטוקול.  מחזירה את הקוד הפעולה ואת השדות השייכים לקוד הפעולה.  (opcode, [parameters]) | בשימוש רק להעברת תמונה |
| pack\_key\_click | הפעולה מקבלת unicode value של מקש שנלחץ | הפעולה מחזירה הודעה לפי הפרוטוקול  opcode + Unicode key-  01 unicode\_key.zfill(7) |  |
| pack\_key\_release | הפעולה מקבלת unicode value של מקש שנעזב | הפעולה מחזירה הודעה לפי הפרוטוקול  opcode + Unicode key-  02 unicode\_key.zfill(7) |  |

|  |
| --- |
| AssistanceSeeker\_protocol |
| אוסף הפעולות הנדרשות בצד תומך לבנייה ופירוק הודעות ע"פ הפרוטוקול |

**אין למחלקה זו תכונות**

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| unpackData | הפעולה מקבלת מידע  מהתקשורת | הפעולה מפרקת את המידע שהתקבל לפרמטרים וה- opcode שלו לפי הפרוטוקול.  מחזירה את הקוד הפעולה ואת השדות השייכים לקוד הפעולה.  (opcode, [parameters]) | בשימוש רק להעברת מקלדת |
| pack\_part\_image | הפעולה מקבלת את ערך המיקום של השמאל, למעלה, ימין ולמטה של התמונה אורך המידע של התמונה | הפעולה מחזירה הודעה לפי הפרוטוקול  opcode + top + left + bottom + right + image data size -  0103000400120010001234 |  |
| pack\_full\_image | הפעולה מקבלת את אורך המידע של התמונה | הפעולה מחזירה הודעה לפי הפרוטוקול  opcode + image data size -  02 img\_data\_size.zfill(10) |  |

|  |
| --- |
| Helper \_logic |
| מחלקה לבניית הלוגיקה של השרת של התומך |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| server | שרת מרובה לקוחות לתקשורת |
| recv\_q | תור לקבלת הודעות מהלקוחות |
| port | הפורט שעליו ירוץ השרת |
| otherIP | הip של הלקוח שאמור להתחבר לשרת |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| main\_Helper | הפעולה מקבלת את כתובת הip של המבקש תמיכה הפורט שעלו עובדים והתור הודעות שעובר בין הלוגיקה למקלדת הודעות (None אם מופעל כפרוסס עכבר)  (otherIP, port, close\_queue) | הפעולה יוצרת תור ויוצרת בעזרתו שרת server שמקשיב רק לotherIP בפורט שהתקבל הפעולה יוצרת עצם מסוג Keyboard\_monitor/Mouse\_monitor ומאתחלת את תכונותיו בהתאמה (מעבירה לkeyboard\_monitor את close\_queue)  ומפעילה את פעולה check כthread בשביל לעבור על הודעות המתקבלות מלקוח המתחבר לשרתים של העכבר/מקלדת |  |
| check | הפעולה מקבלת את תור ההודעות ואת עצם ההקשה למקלדת או מסך שעובד על המחשב  (check\_queue, listener) | הפעולה מקבלת הודעות מהתור, אם התקבלה הודעת disconnect הלקוח התנתק מהשרת ומהסאשן לכן נעצור את עצם ההקשבה שעובד על המחשב ונצא מהפעולה |  |

|  |
| --- |
| Mouse\_monitor |
| מחלקה למעכב אחרי פעולות העכבר ועדכון עליהם |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| server | שרת מרובה לקוחות לתקשורת |
| ClientIP | כתובת לשלוח לה את מידע העכבר |
| listener | עצם של ההקשבה על העכבר |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| \_\_init\_\_ | הפעולה מקבלת עצם שרת וכתובת ip | הפעולה מאתחלת את תכונותיה וקוראת לפעולה \_monitor\_mouse |  |
| \_monitor\_mouse | הפעולה לא מקבלת כלום | הפעולה יוצרת listener חדש להקשיב לפעולות העכבר ומפעילה פעולות מתאימות בהתאמה |  |
| stop\_listening | הפעולה לא מקבלת כלום | הפעולה עוצרת את העצם הקשבה במידה והוא פועל |  |
| \_on\_click | הפעולה מקבלת את הx והy של הלחיצה ומידע על סוג ואופן הלחיצה | הפעולה נקראת כאשר מזוהה שנלחץ על העכבר הפעולה בודקת איזה סוג לחיצה זו ושולחת הודעה מתאימה ללקוח לפי פרוטוקול |  |
| \_on\_scroll | הפעולה מקבלת את הx והy של הגלגל והdx וdy של אופן שינוי הגלגל | הפעולה נקראת כאשר מזוהה שהגלגל של העכבר בפעולה, הפעולה בודקת איזה סוג תזוזה זו ושולחת הודעה מתאימה ללקוח לפי הפרוטוקול |  |
| \_on\_move | הפעולה מקבלת את הx והy של מיקום העכבר החדש | הפעולה נקראת כאשר מזוהה שהעכבר זז הפעולה בודקת איזה סוג תזוזה זו ושולחת הודעה מתאימה לפי הפרוטוקול |  |

|  |
| --- |
| AssistanceSeeker \_mouse\_logic |
| מחלקה לבניית הלוגיקה של הלקוח לעכבר |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| client | לקוח לתקשורת עם שרת (client\_comm) |
| otherIP | כתובת שעליה רץ השרת |
| recv\_q | תור לקבלת הודעות מהשרת |
| port | הפורט שעליו רץ השרת |
| mouse\_cont | עצם ששולט על העכבר |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| main\_AS\_mouse | הפעולה מקבלת את כתובת הip של התומך | הפעולה יוצרת תור ויוצרת בעזרתו לקוח client עם otherIP וport  הפעולה עוברת על תור ההודעות בלולאה אינסופית ומקבלת מידע על העכבר ומבצעת את הפעולה של העכבר שנשלחה לפי הפרוטוקול |  |

|  |
| --- |
| Keyboard\_monitor |
| מחלקה למעכב אחרי פעולות המקלדת ועדכון עליהם |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| server | שרת מרובה לקוחות לתקשורת |
| ClientIP | כתובת לשלוח לה את מידע העכבר |
| special\_keys\_mapping | מילון שמצמיד בין מקשים מיוחדים לערכים חדשים בשבילהם |
| close\_queue | תור multiprocessing להעביר הודעות בין הלוגיקה לעצם הקשבה עכבר |
| listener | עצם ההקשבה למקלדת |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| \_\_init\_\_ | הפעולה מקבלת עצם שרת כתובת ip ותור ההודעות בין הלוגיקה למחלקה | הפעולה מאתחלת את תכונותיה וקוראת לפעולה  \_monitor\_keyboard |  |
| \_monitor\_keyboard | הפעולה לא מקבלת כלום | הפעולה יוצרת listener חדש להקשיב לפעולות המקלדת ומפעילה פעולות מתאימות בהתאמה |  |
| pressed\_end | הפעולה לא מקבלת כלום | הפעולה נקראת כאשר נלחץ על המקלדת של המחשב Shift + ctrl + d  קומבינציה זאת אומרת שהמשתמש רוצה להתנתק מהסאשן. הפעולה הפעולה שולחת לתומך הודעה לעזוב את המקשים האלו ועוצרת את עצם ההקשבה שלה. בנוסף הפעולה מכינסה close לתוך תור ההודעות close\_queue בשביל להודיע ללוגיקה לעצור את כל הפרוססים של השליטה. |  |
| stop\_listening | הפעולה לא מקבלת כלטם | הפעולה עוצרת את עצם ההקשבה למקלדת אם הוא פועל |  |
| \_on\_press | הפעולה מקבלת את הkey שנלחץ | הפעולה נקראת כאשר מזוהה שנלחץ מקש על המקלדת הפעולה ושולחת הודעה מתאימה ללקוח לפי פרוטוקול עם ייצוג המקש |  |
| \_on\_release | הפעולה מקבלת את הkey | הפעולה נקראת כאשר מזוהה ששוחרר מקש על המקלדת הפעולה ושולחת הודעה מתאימה ללקוח לפי פרוטוקול עם ייצוג המקש |  |

|  |
| --- |
| AssistanceSeeker \_keyboard\_logic |
| מחלקה לבניית הלוגיקה של הלקוח למקלדת |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| client | לקוח לתקשורת עם שרת (client\_comm) |
| otherIP | כתובת שעליה רץ השרת |
| recv\_q | תור לקבלת הודעות מהשרת |
| port | הפורט שעליו רץ השרת |
| Keyboard\_cont | עצם מסוג controller של pynput לעבודה עם מקלדת |
| close\_queue | תור multiprocessing להעביר הודעות בין הלוגיקה לפרוסס שמקבל את מידע המקלדת מהתומך |
| special\_keys\_mapping | מילון שמצמיד בין ערכים למקשים מיוחדים שלהם |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| main\_AS\_keyboard | הפעולה מקבלת את כתובת הip של התומך ותור ההודעות להודעות בין הלוגיקה של הלקוח לפרוסס שמקבל את ההודעות מקלדת מהתומך | הפעולה יוצרת תור ויוצרת בעזרתו לקוח client עם otherIP וport  הפעולה מפעילה כ thread את פעולה check\_close לבדיקת לחיצה על קומבינצית ההתנתקות.  הפעולה עוברת על תור ההודעות בלולאה אינסופית ומקבלת מידע מפרקת אותו (AssistanceSeeker\_protocol) lומבצעת את הפעולה הנדרשת |  |
| check\_close | הפעולה מקבלת את תור ההודעות close\_queue ועצם הלקוח | הפעולה בודקת בלולאה אינסופית אם נלחץ קומבינצית הסגירת סאשן על המקלדת, אם נלחץ תסגור את הלקוח ותכניס לתוך תור ההודעות close בשביל להודיע ללוגיקה לסגור את כל הפרוססים של השליטה. |  |

|  |
| --- |
| AssistanceSeeker\_screen\_logic |
| מחלקה לבניית הלוגיקה של הלקוח של הנתמך להעברת המסך |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| client | עצם לקוח לתקשורת לשרת(client\_comm) |
| recv\_q | תור לקבלת הודעות מהשרת |
| port | הפורט שעליו רץ השרת |
| otherIP | הip של השרת |
| currScreen | עצם מסוג תמונה (PIL) של המסך כרגע |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| main\_AS\_screen | הפעולה מקבלת את כתובת ה ip של התומך | הפעולה יוצרת תור ויוצרת בעזרתו לקוח client שמתחבר לשרת שרץ על otherIP בפורט שהתקבל ושולחת תמונת מסך ראשונית  בלולאה אינסופית הפעולה מבצעת תמונת מסך הפעולה קוראת לgetChanged במידה ויש שינוי שולחת את הריבוע המייצג את השינוי במידה והוא גדול מ75 אחוז שולחת את תמונת המסך החדשה  ושומרת אותה במשתנה התמונה |  |
| getChanged | הפעולה מקבלת תמונת מסך חדשה  ותמונת מסך הנוכחית | הפעולה מחזירה את מיקום ריבוע השינוי בתמונה ואת תמונת של השינוי עצמה |  |
| getChangedPrecentage | הפעולה מקבלת את הbounding box של תמונת השינוי, כלומר גודל תמונת השינוי ומספר הפיקסלים שיש על המסך | הפעולה מחשבת את אחוז השינוי מהמסך כולו לפי גודל תמונת השינוי וכמות הפיקסלים במסך |  |

|  |
| --- |
| Helper\_screen\_logic |
| מחלקה לבניית הלוגיקה של השרת למסך |

| תכונות | |
| --- | --- |
| שם תכונה | פירוט |
| server | לקוח לתקשורת עם לקוח(server\_comm) |
| otherIP | כתובת שעליה רץ השרת |
| recv\_q | תור לקבלת הודעות מהלקוח |
| port | הפורט שעליו רץ הלקוח |
| currScreen | עצם מסוג תמונה (PIL) של המסך כרגע (משתנה גלובלי) |

| פעולות | | | |
| --- | --- | --- | --- |
| שם פעולה | טענת כניסה | טענת יציאה | הערות |
| main\_Helper\_screen | הפעולה מקבלת את כתובת ה ip של התומך | הפעולה יוצרת תור ויוצרת בעזרתו שרת server עם otherIP וport ומשתנה למסך הנוכחי  הפעולה עוברת על תור ההודעות בלולאה אינסופית ומקבלת מידע מפרקת אותו בונה את התמונה וקוראת לbuildScreen במידה והתמונה היא לא תמונה מלאה אלא שינוי.  הפעולה מראה את התמונה בעזרת pygame ומראה אותה מחדש במידה וקיבלנו תמונה חדשה |  |
| build\_Screen | הפעולה מקבלת את הtop left right bottom ערכים של תמונת השינוי ואת תמונת השינוי עצמה בבתים | הפעולה בונה את המסך החדש בעזרת התמונה הקודמת של המסך והתמונת שינוי שהתקבלה ושומרת אותה בcurrScreen |  |

1. תיאור סביבת הפיתוח

שפת תכנות: שפת התכנות שבחרתי לעבוד איתה בפרויקט זה היא Python 3.7, בנוסף אעבוד עם SQL בשביל לתקשר עם הבסיס נתונים.

כלים לפיתוח ובדיקה:

* PyCharm: לסביבת עבודה נוחה לכתיבת קוד ובשביל Debugging
* Wireshark- לבדיקת תקשורת ומידע מוצפן שעובר ברשת
* DB Browser for SQLite: לבנות בסיס נתונים שאפשר להכניס לתוכו מידע שלנו
* CMD- להרצת פקודות ותכניות על המחשב
* File Explorer: לעבודה עם קבצים תמונות וכל דבר אחר שמאוכסן על המערכת קבצים במחשב

1. תיאור האלגוריתמים המרכזיים בפרויקט:

סוגיה: כיצד לממש את החיבור peer2peer?

| מס' | חלופה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | חיבור הלקוחות דרך השרת | המידע עובר דרך "גורם רשמי" , מבוקר, אמין  השרת אחראי על החיבור ומוודא את הצדדים | איטי – מידע עובר דרך שלישי  MITM – גורם נוסף מסתכל על המידע שעובר |
| 2 | חיבור ישיר בין הלקוחות | הרבה יותר מהיר  אין MITM | אין בקרה על המידע דרך גורם רשמי |

החלופה הנבחרת היא 2 – חיבור ישיר בין הלקוחות, מכיוון שבמערכת כזו נדרש חיבור מהיר וגם הגנה מפני MITM

סוגיה: איך יעבור המידע החיבור peer2peer כלל המידע: מהמקלדת, מהעכבר ותמונת המסך?

| מס' | חלופה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | סוקט יחיד | ניהול פשוט | איטי – כל המידע עובר דרך stream יחיד  תקורה – נדרש למיין את המידע |
| 2 | סוקט לכל סוג מידע | מהירות – כל מידע לא תלוי במידע האחר  פחות תקורות – אין צורך למיין (מימוש ריבוב) | שימוש במספר רב של סוקטים  קשה יותר למימוש |
| 3 | Thread לכל סוקט שמהווה סוג מידע אחר | מהירות – כל מידע לא תלוי במידע האחר  פחות תקורות – אין צורך למיין (מימוש ריבוב)  אין עיכובים אם תהליכון 'תקוע' מאיזו סיבה | שימוש במספר רב של סוקטים  קשה יותר למימוש  קשה לניהול משאבים משותפים |
| 4 | Process לכל סוקט שמהווה סוג מידע אחר | הרבה יותר מהיר – כל תהליך רץ בנפרד לחלוטין, אין חלוקת משאבים | שימוש במספר רב של סוקטים  מימוש משאבים משופים (אם יידרש) הרבה יותר קשה |

החלופה הנבחרת היא 4 – Process לכל סוקט שמהווה סוג מידע אחר – במערכת שלי חשובה המהירות יותר מכל דבר אחר (שיתוף מסך) ואין צורך בשיתוף בין המידע של התמונה, עכבר והמקלדת

סוגיה: באיזה צורה מעבירים את המידע שהשתנה בתמונה?

| מס' | חלופה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | העברת תמונה שלמה כל פרק זמן | הרבה יותר קל למימוש  קל לפירוק ובניית פרוטוקול לשליחות | כבד מאוד- לשלוח תמונה כל פרק זמן מסוים לפענח אותה ולהציג אותה זו פעולה מאוד כבדה שיכולה לאט את כל המערכת |
| 2 | העברת תמונה שלמה כל פעם שמזהים שינוי בה | פחות כבד- מתבצעת שליחות בתדירות יותר נמוכה (תלוי שינוי המסך) | עדיין שליחה של תמונה שלמה שזו פעולה מאוד כבדה על התומך והמבקש תמיכה |
| 3 | העברת רק את השינוי של המסך כל פעם שמזהים בו שינוי | הרבה פחות כבד- לא רק שתדירות השליחות של התמונה מתבצעת בקצב יותר איטי גם לא נשלחת תמונה שלמה ורק השינוי. | קשה למימוש- זיהוי שינוי במסך+פירוק השינוי מהתמונה ושליחה של השינוי היא פעולה קשה למימוש |

החלופה הנבחרת היא 3 – העברת רק את השינוי של המסך כל פעם שמזהים בו שינוי, מכיוון שבמערכת כזו נדרשת מהירות מעל הכל, אפילו אם יותר קשה לממש את העברת שינוי במסך.

סוגיה: באיזה פרוטוקול תקשורת להעביר את התמונה בין המקבל תמיכה והתומך?

| מס' | חלופה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | שימוש בפרוטוקול UDP להעביר תמונה | המידע עובר הרבה יותר מהיר מפרוטוקול TCP | לא אמין- המידע שנשלח דרך פרוטוקול UDP נשלח לרשת ואין בקרה ובדיקה אם הוא מגיע או לא מגיע |
| 2 | שימוש בפרוטוקול TCP להעביר מידע תמונה | הקשר שנוצר בין השרת והלקוח אמין מוצפן ונבדק כאשר מגיע | איטי מאוד-המידע עובר הרבה יותר לאט מUDP מכיוון שצריך לעשות לחיצת יד משולשת וכל המידע נבדק. |

החלופה הרצויה היא 1 – שימוש בפרוטוקול UDP להעביר תמונה, מכיוון שבמערכת כזו נדרשת מהירות מעל הכל, אך החלופה הנבחרת היא 2 - שימוש בפרוטוקול TCP להעביר מידע תמונה. נבחר ב TCP ולא UDP מכיוון שבחרנו להעביר רק את השינוי של התמונה אם הוא נאבד בגלל UDP נוצר בעיה. מעבר של רק השינוי יהיה יותר יעיל מלהעביר תמונה שלמה רק בUDP לכן נבחר TCP

סוגיה: איך לחבר משתמש תומך ונתמך?

| מס' | חלופה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | השרת בוחר לחבר את התומך ונתמך הראשונים אשר מתחברים למערכת (לפי סדר) | קל למימוש- כל פעם שיש לשרת תומך ונתמך במערכת אז הוא מחבר אותם  אמין- השרת בוחר איזה תומך מתחבר לנתמך, לכן הקשר שנוצר הוא אמין יותר | לא מתאים לצרכי הנתמך- מבקשי תמיכה העובדים במערכת כמעט תמיד מקבלים תמיכה ממישהו ספציפי שגם בתוך המערכת, לכן אם השרת מחבר את הנתמך אוטומטית לתומך פנוי זה לא עוזר לנתמך. |
| 2 | תומך מקבל קוד רנדומלי שתקף תמיד מהשרת והנתמך מכניס אותו בשביל להתחבר | מתאים לצרכי הנתמך- הנתמך יכול להכניס קוד (אשר קיבל מהתומך) בשביל להתחיל את הקשר לכן הוא בוחר למי להתחבר | לא אמין- מכיוון שהקוד ממשיך להיות תקף גם כאשר הקשר מתחיל\מתנתק, MITM יכול תמיד לנסות להתחבר במקום נתמך |
| 3 | תומך מקבל קוד רנדומלי שמוגבל בזמן מהשרת והנתמך מכניס אותו בשביל להתחבר | מתאים לצרכי הנתמך- הנתמך יכול להכניס קוד (אשר קיבל מהתומך) בשביל להתחיל את הקשר לכן הוא בוחר למי להתחבר    אמין – בגלל שהקוד התקבל מהשרת ומוגבל בזמן האמינות שלו תלויה רק בתומך ולמי הוא שולח את הקוד. לאחר שמתחבר נתמך אין אפשרות להמשיך להשתמש בקוד | קשה למימוש- בשרת צריך לממש מחולל קודים אשר ישמור כמה זמן מוגבל לכל קוד. דבר זה מסובך למימוש בצד השרת מבחינת שמירת ובדיקת הזמן והקוד שהוכנס. |

החלופה הנבחרת היא 3 – תומך מקבל קוד רנדומלי שמוגבל בזמן מהשרת והנתמך מכניס אותו בשביל להתחבר, אנחנו רוצים שהמערכת שלנו גם תתאים לצרכי המשתמשים וגם תהיה אמינה. חלופה זו מבצעת זאת לכן נבחר בה.

סוגיה: באיזה סוג הצפנה להעביר מידע ברשת?

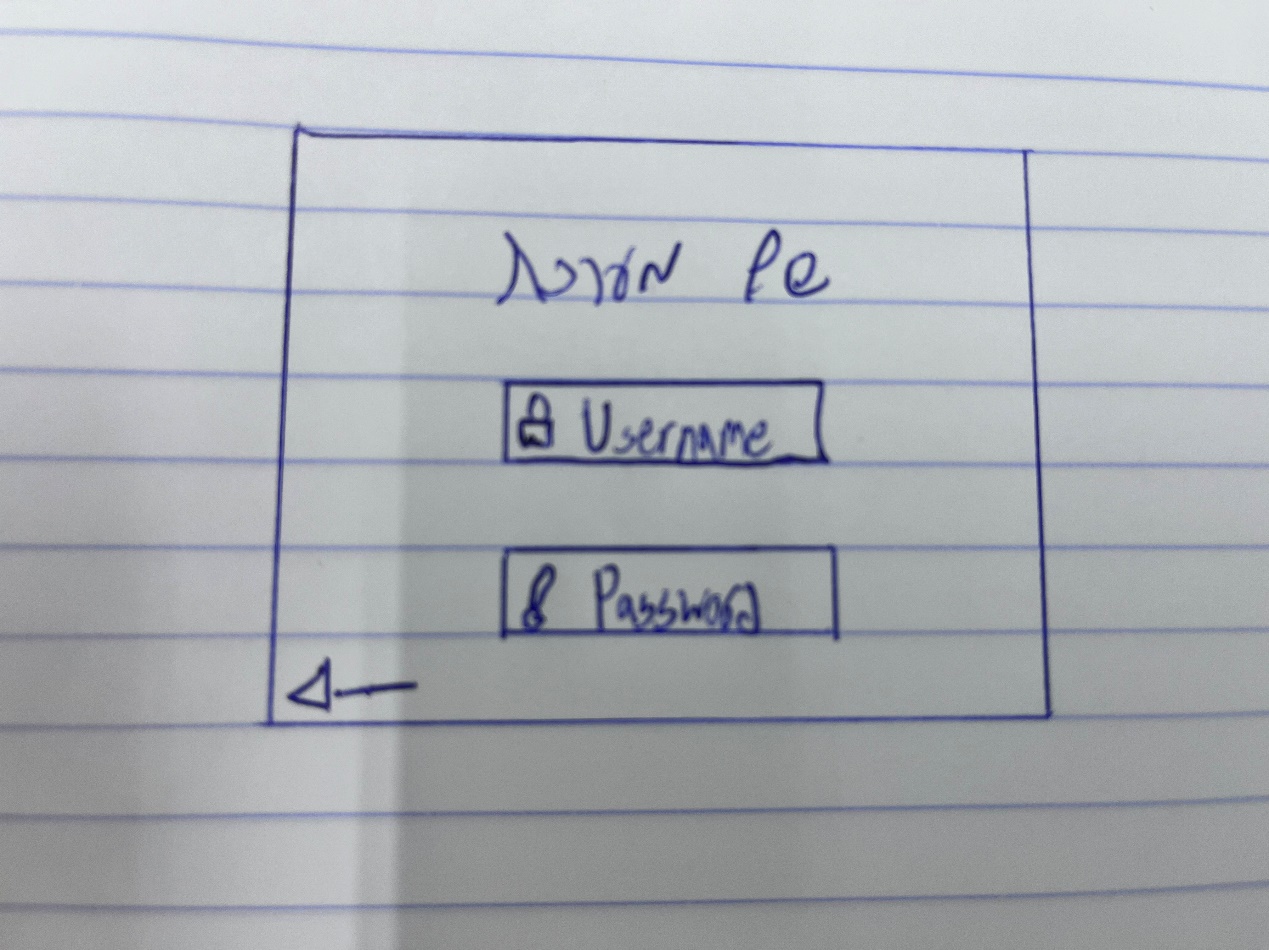
| מס' | חלופה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | הצפנה סימטרית | קל למימוש- הצפנה סימטרית לא קשה למימוש  מהירה- תהליך ההצפנה מהיר ולא מצריך הרבה זמן בשביל להצפין ולפענח מידע | בעיית המפתחות – הצפנה סימטרית תלויה במפתח משותף לנתמך ולתומך, בתוך הרשת אין איך להסכים על מפתח משותף בין המשתמשים בלי שיעבור באופן גלוי ברשת |
| 2 | הצפנה א-סימטרית | אין בעיית המפתחות-  הצפנה א-סימטרית עובדת על מפתח ציבורי ומפתח אישי, כך כל משתמש יכול להצפין למשתמש אחר מידע בעזרת המפתח הציבורי שלו ורק המשתמש שהמפתח ציבורי שלו יכול לפענח את המידע בעזרת המפתח האישי שלו.  בעזרת הצפנה זו אין בעיית מפתחות ואפשר להעביר מידע מוצפן בלי לעבוד עם מפתח משותף בין משתמשים. | איטי-  הצפנה א-סימטרית בדרך כלל משתמשת במפתחות ציבוריים ופרטיים מאוד מאוד ארוכים. בנוסף תהליך ההצפנה מאוד כבד לכן להצפין הרבה מידע יאט את המערכת |
| 3 | הצפנה היברידית | אין בעיית מפתחות – בהצפנה היברידית משתמשים בהצפנה א-סימטרית להעביר מפתח משותף בין התומך והנתמך בעזרת מפתח ציבורי ומפתח פרטי. לאחר מכן המשתמשים עוברים לתקשר עם הצפנה סימטרית שיותר מהיר ומשתמש במפתח משותף שגלוי רק להם | קשה יותר למימוש-  שימוש בשני שיטות הצפנה יכול לסבך את הקוד ויהיה קשה לממש |

החלופה הנבחרת היא 3 – הצפנה היברידית, מכיוון שהשימוש המשותף של שני שיטות ההצפנה עושה הבדל ענק במהירות ומביא את הטוב משני העולמות- הצפנה א-סימטרית פותר את בעיית המפתחות ויותר קל לעבוד עם הצפנה סימטרית כאשר יש לנו מפתח משותף הגלוי רק לתומך והנתמך.

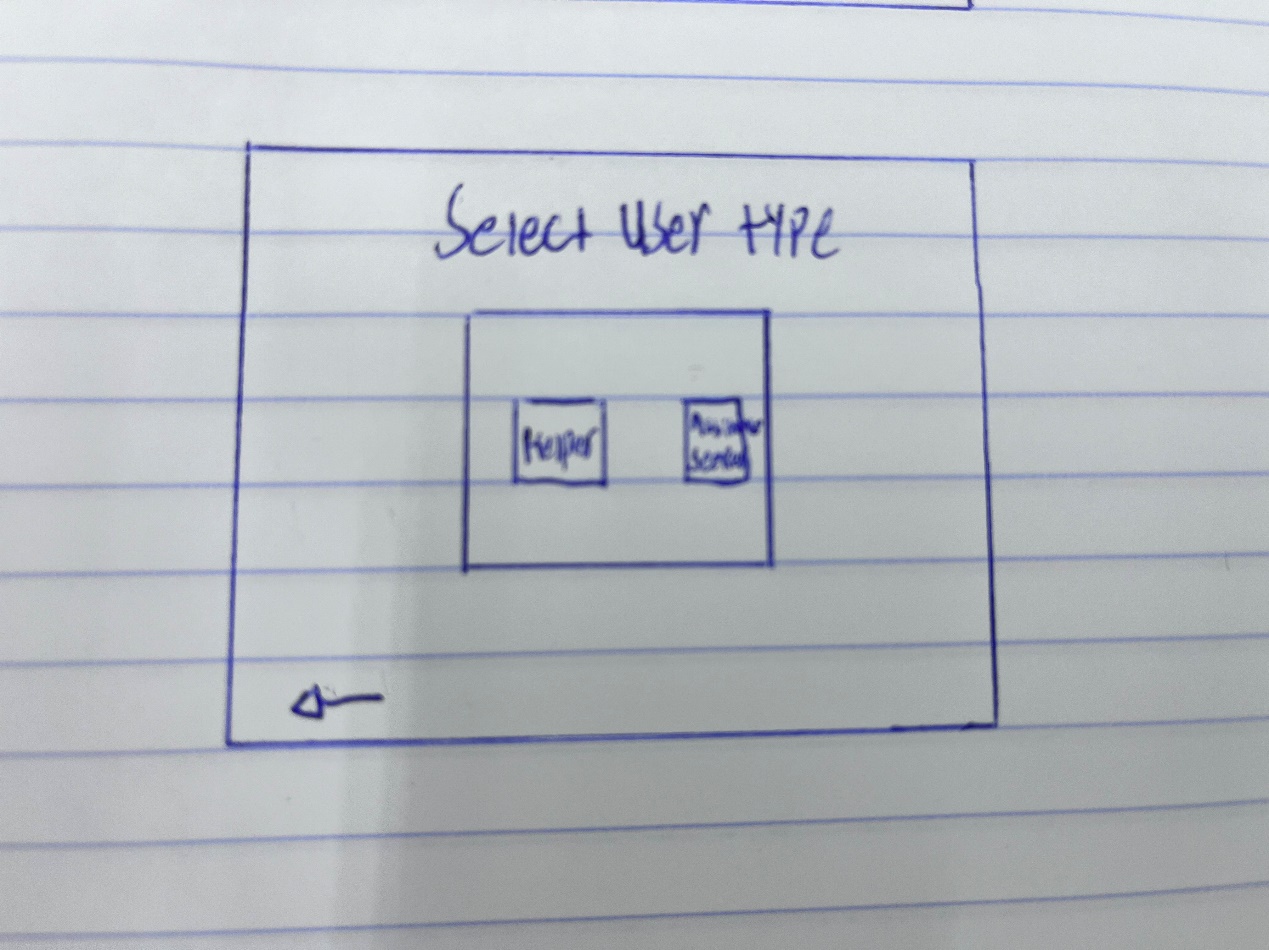
1. תיאור מסכי הפרויקט:

מסך 1: התחברות למערכת

מסך זה נועד בשביל להתחבר למערכת עם משתמש קיים או ליצור משתמש חדש. מסך זה יהיה המסך הראשון שיופיע כשנכנסים למערכת. כאשר מכניסים שם משתמש וסיסמא השרת בודק אם הם קיימים במערכת במידה ונכנסים למשתמש קיים ובודק אם הקלט תקין במידה והוא יוצר משתמש חדש. לאחר שמשתמש מתחבר למערכת הוא עובר למסך 2: בחירת סוג משתמש ששם הוא בוחר אם הוא תומך או מבקש תמיכה.

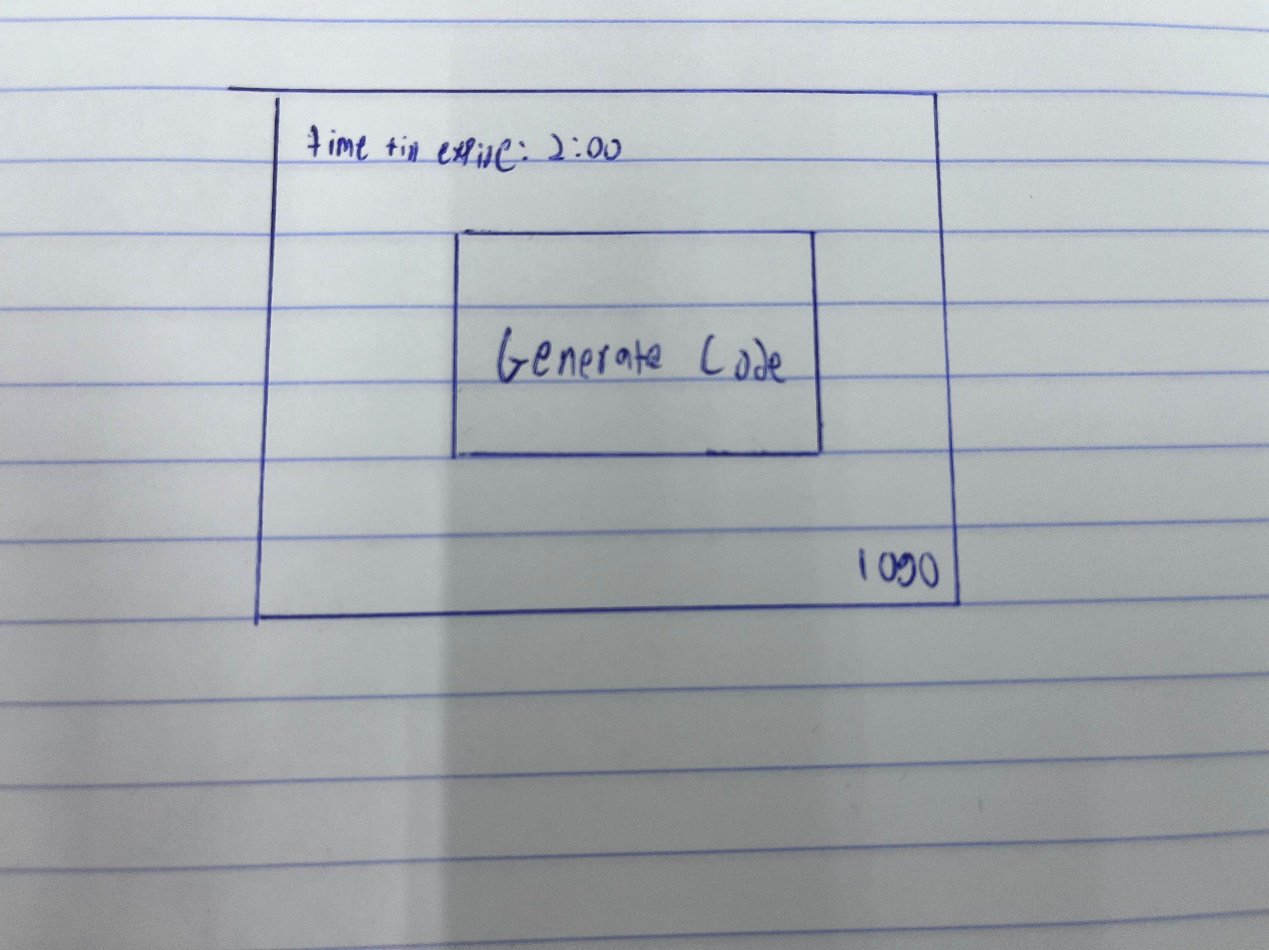


מסך 2: בחירת סוג משתמש

מסך זה יופיע למשתמש אחרי שהוא מתחבר למערכת. מטרת מסך זה היא שהמשתמש בוחר אם הוא רוצה להביא תמיכה למשתמשים אחרים או רוצה לקבל תמיכה ממשתמשים אחרים. השרת מאשר את המשתמש אם הוא רוצה להיות תומך לפי הBlacklist שלו בבסיס הנתונים. לאחר שהמשתמש בוחר סוג משתמש והשרת מאשר אותו, משתמשי תומכים יעברו למסך 3: קבלת קוד חדש ומשתמשי מבקשי תמיכה יעברו למסך 4: הכנסת קוד לפתיחת סאשן. 

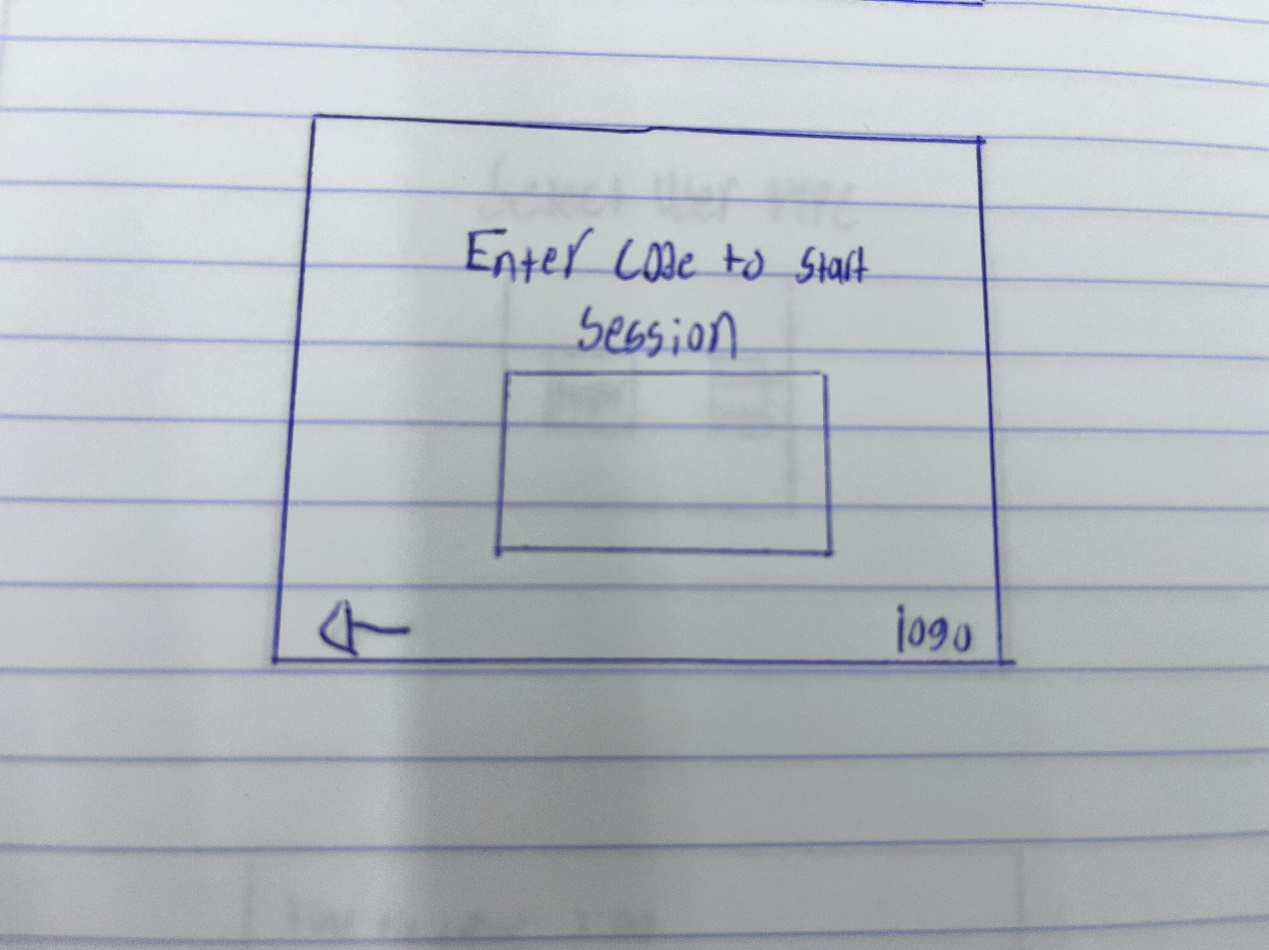
מסך 3: קבלת קוד חדש

מסך זה יופיע לאחר שמשתמש מאושר כתומך במערכת. מטרת מסך זה היא לפתוח קוד חדש על מנת שמבקש תמיכה יכניס אותו וכך יפתח סאשן. כאשר התומך מבקש קוד חדש, השרת יוצר קוד רנדומלי אשר מוגבל בזמן. לאחר קבלת הקוד המשתמש עובר למסך 5: הקוד המשותף.

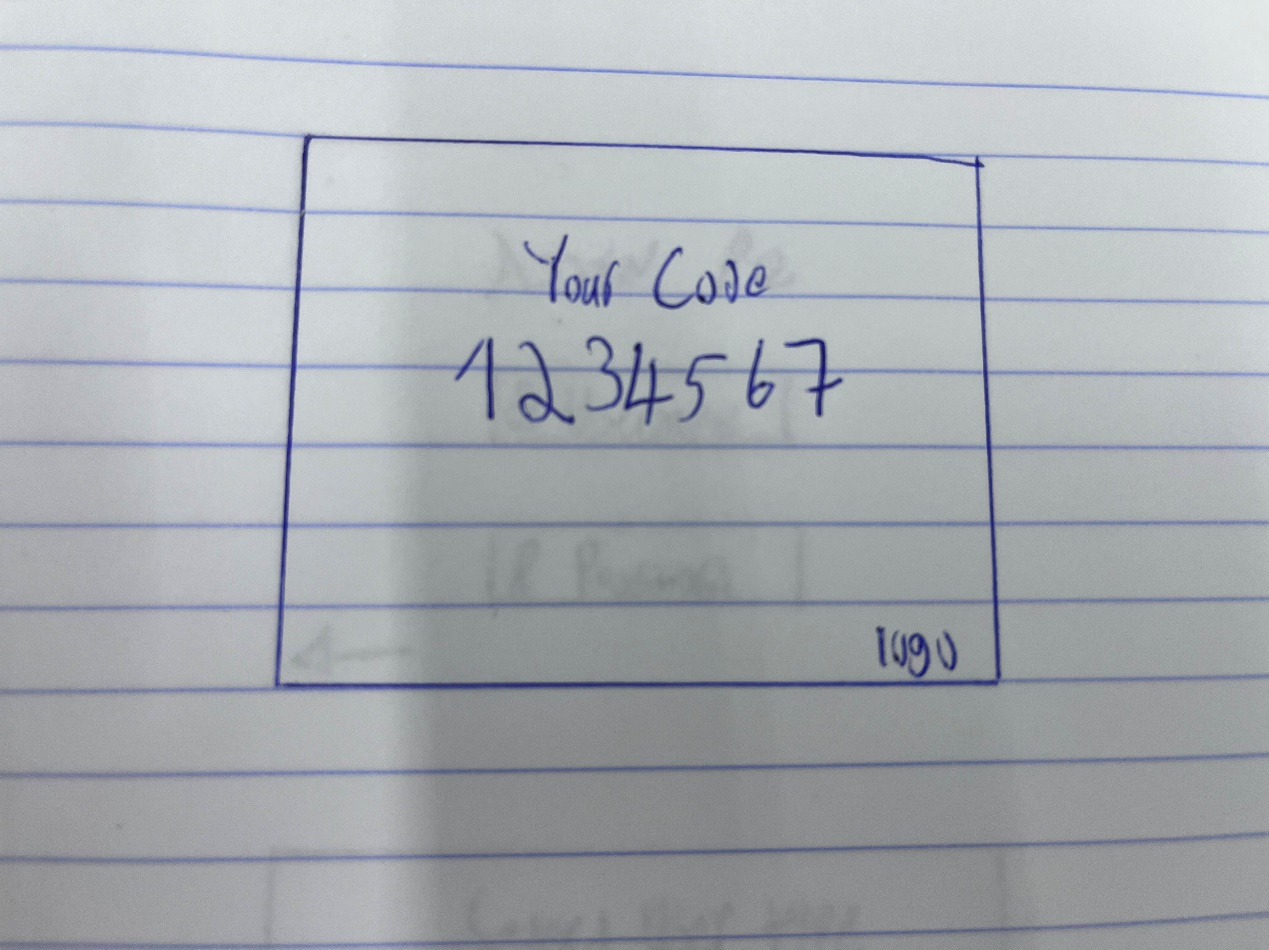


מסך 4: הכנסת קוד לפתיחת סאשן

מסך זה יופיע לאחר שמשתמש בחר ואושר כמבקש תמיכה. מטרת מסך זה היא להכניס קוד שהמשתמש קיבל מתומך אחר וכך להתחבר לסאשן שיצר התומך. הקוד שהכניס המבקש תמיכה נשלח לשרת והשרת בודק אם הקוד קיים והוא תקף, במידה וכן המשתמשים מחוברים וסאשן התמיכה התחיל. ממסך זה המבקש תמיכה לא עובר לאף מסך אחר, הוא יכול להפסיק את השיתוף על ידי רצף פעולות מסוים שמכניס במקלדת.

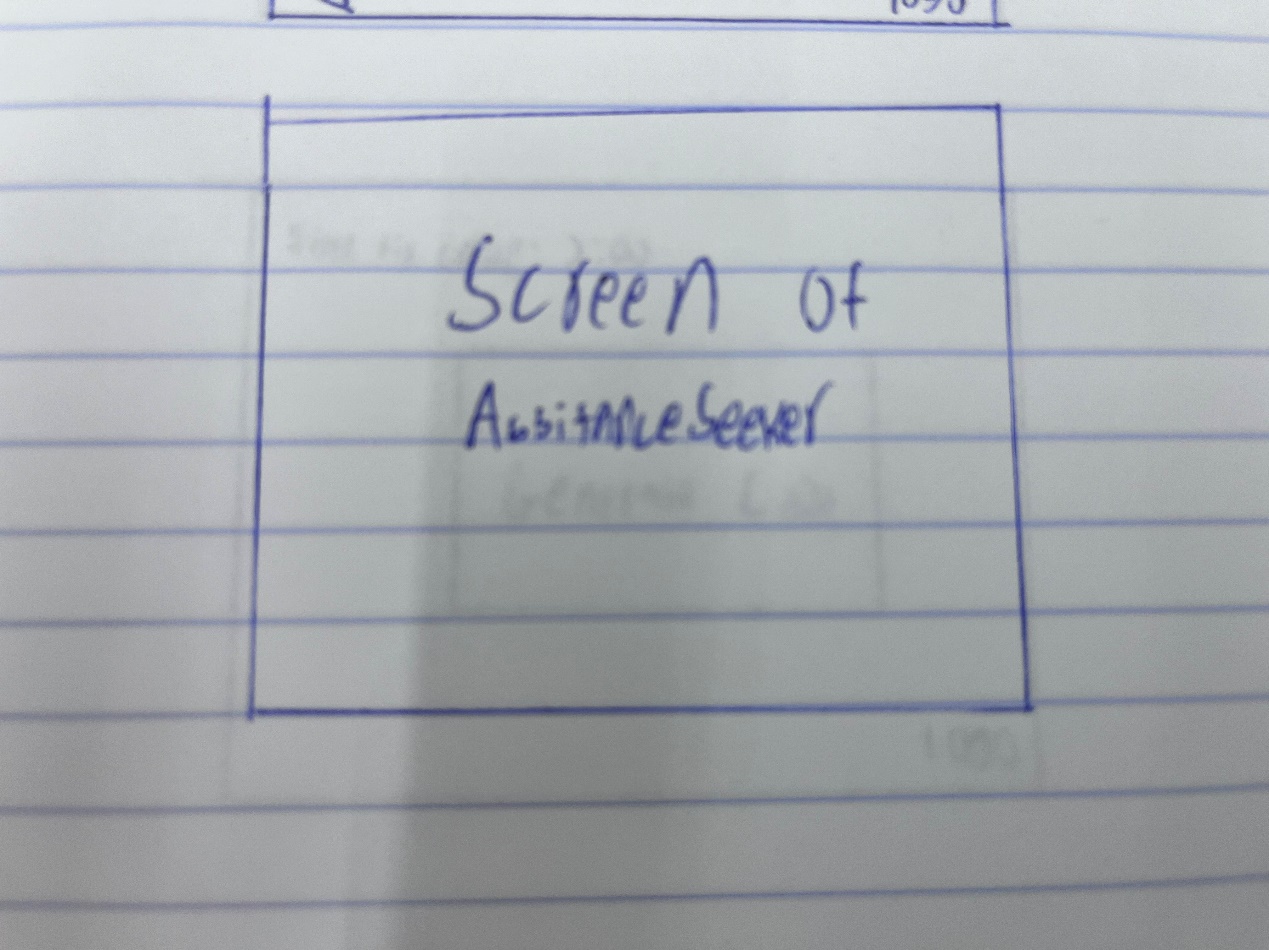


מסך 5: הקוד המשותף

תומך מגיע למסך זה לאחר שהוא פותח סאשן חדש. מטרת מסך זה היא להראות לתומך את הקוד הזמני שצריך להכניס המבקש תמיכה בשביל להתחבר ולהתחיל את סאשן התמיכה. לאחר שהשרת מקבל ממבקש תמיכה אחר את הקוד הנוכחי והוא מאושר ותקף התומך יעבור למסך 6: מסך השיתוף.

מסך 6: מסך השיתוף

תומך מגיע למסך זה כאשר המבקש תמיכה מצטרף לסאשן שלו לאחר שהכניס את הקוד שלו. מטרת מסך זה הוא שהתומך יכול לראות את המסך מחשב של המבקש תמיכה ויכול לשלוט על העכבר והמקלדת של המבקש תמיכה. ממסך זה התומך מסיים את התמיכה לאחר שהמבקש תמיכה מבקש לעשות זאת או שהוא מתנתק מהמערכת.

1. תיאור פרוטוקול התקשורת

שרת <-> לקוח

**\*כל ההודעות עוברות עם שדה אורך XXX בהתחלה**

|  | סוג הודעה | ממי למי? | שדות | דוגמא |
| --- | --- | --- | --- | --- |
| 1 | בניית הצפנה א-סימטרית: שליחת מפתח גלוי של שרת | ש -> ל | מפתח גלוי - 1234 | 1234 |
| 2 | בניית מפתח משותף להצפנה סימטרית: העברת מפתח פרטי מוצפן | ל -> ש | מפתח משותף (מוצפן) - 56894 | 56894 |

**מנקודה זו כל המידע העובר בהנחה מוצפן אך רשום כאילו הוא לא מוצפן**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | mac address | ל -> ש | mac address-  00-B0-D0-63-C2-26 | 0600-B0-D0-63-C2-26 |
| 4 | login/sign up | ל -> ש | opcode- 01/00  username\_len- 6  Itamar - username  Itamar6 - password | 016ItamarItamar6 |
| 5 | login/ sign up | ש -> ל | opcode- 01/00  0/1 – success/fail | 010 |
| 66 | user type | ל -> ש | 02 -opcode  0/1 – helper/assistance seeker | 020 |
| 7 | user type | ש -> ל | 02 -opcode  0/1 – success/fail | 020 |
| 8 | קבלת קוד חדש | ל -> ש | 03 -opcode | 03 |
| 9 | קבלת קוד חדש | ש -> ל | 03 -opcode  78634 – code | 0378634 |
| 10 | תוקף קוד תם | ש -> ל | 03 -opcode  expired-2 | 032 |
| 11 | הכנסת קוד | ל -> ש | 04 -opcode  43892 -code | 0443892 |
| 12 | הכנסת קוד | ש -> ל | 04 -opcode  0/1 – success/code invalid | 040 |
| 13 | חיבור חדש מול משתמש | ש -> ל | 05 -opcode  127.0.0.1 – ip of other user  userType- A | 05127.0.0.1A |

נתמך <-> תומך: עכבר (port:2001)

**\*כל ההודעות עוברות עם שדה אורך XXX בהתחלה כהנחה**

|  | סוג הודעה | ממי למי? | שדות | דוגמא |
| --- | --- | --- | --- | --- |
| 1 | בניית הצפנה א-סימטרית: שליחת מפתח גלוי של תומך | ת -> נ | מפתח גלוי - 1234 | 1234 |
| 2 | בניית מפתח משותף להצפנה סימטרית: העברת מפתח פרטי מוצפן | נ -> ת | מפתח משותף (מוצפן) - 56894 | 56894 |

**מנקודה זו כל המידע העובר בהנחה מוצפן אך רשום כאילו הוא לא מוצפן**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | העברת מיקום ולחיצת עכבר | ת -> נ | 1432 – x position of mouse  854 – y position of mouse  0/1/2/3/4/5/6 -left click/right click/ scroll up/ scroll down/ nothing/left click release/ right click release | 143208540 |

נתמך <-> תומך: מקלדת (port:2002)

**\*כל ההודעות עוברות עם שדה אורך XXX בהתחלה כהנחה**

|  | סוג הודעה | ממי למי? | שדות | דוגמא |
| --- | --- | --- | --- | --- |
| 1 | בניית הצפנה א-סימטרית: שליחת מפתח גלוי של תומך | ת -> נ | מפתח גלוי - 1234 | 1234 |
| 2 | בניית מפתח משותף להצפנה סימטרית: העברת מפתח פרטי מוצפן | נ -> ת | מפתח משותף (מוצפן) - 56894 | 56894 |

**מנקודה זו כל המידע העובר בהנחה מוצפן אך רשום כאילו הוא לא מוצפן**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | העברת לחיצת מקלדת | ת -> נ | opcode- 01  'a'-key | 01ord('a') |
| 4 | העברת עזיבת לחיצה על מקלדת | ת -> נ | 02 -opcode  'a' -key | 02ord('a') |
| 5 | העברת עזיבת מקשי התנתקות מסאשן | ת -> נ | 02 -opcode  9999999 | 029999999 |

נתמך <-> תומך: מסך (port:2003)

**\*כל ההודעות עוברות עם שדה אורך XXX בהתחלה כהנחה**

|  | סוג הודעה | ממי למי? | שדות | דוגמא |
| --- | --- | --- | --- | --- |
| 1 | בניית הצפנה א-סימטרית: שליחת מפתח גלוי של תומך | ת -> נ | מפתח גלוי - 1234 | 1234 |
| 2 | בניית מפתח משותף להצפנה סימטרית: העברת מפתח פרטי מוצפן | נ -> ת | מפתח משותף (מוצפן) - 56894 | 56894 |

**מנקודה זו כל המידע העובר בהנחה מוצפן אך רשום כאילו הוא לא מוצפן**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | העברת אזור שינוי ושינוי במסך | נ -> ת | 01-opcode  0300 -left  0400 -top  1200 -right  1000 -bottom  1234- image data size  Image data | 0103000400120010001234data |
| 4 | העברת תמונת מסך שלמה | נ -> ת | 02 -opcode  1234- image data size  Image data | 021234data |

1. תיאור מבני הנתונים

משתנים מקומיים:

| שם | טיפוס | שימוש | דוגמה |
| --- | --- | --- | --- |
| open\_client | dictionary(sockets: (IP, encryption\_object) | קישור בין sockets פתוחים לבין ה IP של ה socket והעצם הצפנה שבעזרתו מצפינים את המידע  נמצא בשרת | (socket object: '192.3.4.203: encryption\_object') |
| users | dictionary(IP:(mac address, type\_user) | קישור בין כתובת ip כתובת mac של לקוח וסוג משתמש הלקוח | ('192.3.4.203': '00-B0-D0-63-C2-26': 'H') |
| commands | dictionary(opcode:function) | קישור בין אופקוד לפונקציה שקוראים לה מהאופקוד הזה | (("00": handle\_login\_ans |
| currScreen | עצם PIL של תמונת מסך נוכחי | לשמור את המסך הנוכחי להדבקה עליו ותצוגה |  |

**מסד נתונים:**

לשרת יהיה גישה למסד נתונים אחד בשם Uelp\_db

במסד נתונים זה תהיה טבלה בשם users עם המידע:

עמודה 1 (username): שם משתמש של משתמש. ((primary key

עמודה 2 (:(password hash של הסיסמא של השם משתמש

|  |  |
| --- | --- |
| Password | Username |
| 4b4b0f17da04db1de6d24fade424adb13c41a802bae8fdb723df0fc43391b3cb | Itamarnahum234 |
| ffd8bc653cf25ed962af9af60aaea2d703213316feea067e40b15c847cb73fec | Moshbenaro27272 |
| a92f49d0461deffcf4b8b02149e591fa663d6052cec4fd714064d2beadd37ca7 | Merrygeva2024 |
| b183af27552242fc4addf8fa6498ff9d62d7ba6d82a4d7f22b7c848fc4f88043 | Kerenkuper023 |

במסד נתונים תהיה טבלה נוספת בשם macs עם הhash של הכתובת הפיזית של כל המשתמשים שחסומים מלהיות תומך במערכת:

עמודה 1 ( hash :(mac addressשל הכתובת הפיזית של user (primary key)

|  |
| --- |
| Mac addresses |
| 8c523a89ff59046ef5c827f3f3575212944f55a99992fe14824a085ea775d098 |
| d90e45e4258d7fae3e49a24b940a7ea46860621fecd5b948bb48971ae9f52152 |
| b2ca5ab19a63d223c36c9d81c36ea16d34a461259b8becf1c56ca834284aa5f8 |

1. סקירת חולשות והאיומים

|  | חולשה | סכנות | פתרון |
| --- | --- | --- | --- |
| 1 | SQL Injection | sql injection היא מתקפה נפוצה מאוד שמשתמשים בה להשיג מידע רגיש מבסיסי נתונים ולגשת למידע שלא אמורים לגשת.  עושים זאת על ידי הכנסת קוד sql בתוך מקומות שהמשתמש חושב שמריצים את הinput שלו כחיפוש בבסיס נתונים | כתיבת שאילתות נכונות כשניגשים לטבלאות בבסיס נתונים.  כך שלא מתאפשר הכנסת קוד זדוני והשגת טבלאות שלא אמורות להיות גישה אליהם |
| 2 | גניבת מידע שמור | מידע ששמור לאורך זמן יכול להיגנב אם נשמר באופן פומבי | שמירת מידע כhash מגן עליו כך שאי אפשר להפוך פונקציית hash ורק להשוות אליה.  אצלי בפרויקט כל מידע רגיש וחשוב שמור עם hash |
| 3 | Man in the middle | מתקפה זו פועלת כך שמישהו ברשת מאזין לשיחה אשר מתקיימת בין שתי ישויות וכך יכול להיחשף למידע רגיש או אפילו לשנות את אופן השיחה ולהשפיע על הישויות בעזרת התחזות לאחד מהן | אצלי בפרויקט כל מידע שמועבר ברשת מוצפן. תחילה בונים מפתח סימטרי משותף בעזרת הצפנה א-סימטרית ולאחר מכן כל מידע שעובר ברשת מוצפן בעזרת המפתח המשותף. |
| 4 | פרוטוקולי תקשורת | רשת אינטרנט נחשבת לא בטוחה ובשביל שמידע יגיע ליעדו צריך להשתמש בפרוטוקולי תקשורת בטוחים וידועים | בפרויקט שלי אני משתמש בפרוטוקול תקשורת tcp שמבטיח שההודעות יגיע בסדר הנכון ולישות הנכונה ברשת. דבר זה נעשה על ידי ה Three way Handshake שהפרוטוקול מבצע בעת חיבור ועל ידי הchecksum ובעזרת הצפנה |

**פרק ה' - UELP - הקוד**

1. עבור (שעושים קטעי קוד מיוחדים משהו מיוחד, משהו מסובך, משהו בדרך שונה, משהו יפה בעיניכם ויש לכלול בתוכם את קטעי הקוד הרלוונטים לאלגוריתמים המרכזיים שציינתם בפרק הקודם):

* הסבר על היכולת
* הקוד עצמו (כתוב ע"פ כללי התכנות הנכון ומלווה בהערות כנדרש)
* (למצוא לפחות 7 קטעי קוד מעניינים ...)
* רצוי תמיד לבחור את אלא שהסברתם קודם לכן את האלגוריתם שלהם

1. בנוסף יש לצרף את **שאר קוד המערכת** , לא נדרש הסבר לצד כל קוד אבל כן את שם הקובץ.  
   לבצע 'שמירה' של הקוד לקבצי PDF אן העתקה לתוך קבצי word **ולא לצרף כתמונה**

**פרק ו' - UELP – בדיקות ('קופסא לבנה')**

1. פירוט הבדיקות

* פירוט הבדיקות שהופיעו בפרק האפיון ברמת הביטים – לתאר בפירוט מה נדרש לבצע ומה בוצע בפועל. וכמובן על בדיקה צריכה להתבצע יותר מפעם אחת ועד שהיא עוברת

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **שם הבדיקה** | **מטרת הבדיקה** | **מה נדרש לבצע** | **מתי** | **מה בוצע בפועל** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* יש לתעד כל בדיקה שמבוצעת, חייבים לחזור לפחות פעמיים על כל בדיקה – כל חזרה תתועד בשורה נפרדת עם תוצאות הבדיקה

**פרק ז' - UELP – מדריך למשתמש**

1. מדריך למשתמש הכולל עבור כל תהליך/יכולת במערכת:

* הסבר על התהליך/יכולת
* תרשים זרימה (במקרה של תהליך)
* צילומי מסך מלווים בהסברים בשפה פשוטה

במידה ויש יותר ממשתמש אחד למערכת , לבנות פרקים שונים לדוגמא: מדריך למשתמש קצה, מדריך למנהל מערכת וכו'

**פרק ח' - UELP – רפלקציה**

1. מבט אישי על העבודה ועל תהליך פיתוחה:

* אתגרים שהיו לי בדרך
* אירועים מעניינים שקרו במהלך הפיתוח
* התמודדות עם קשיים, איך מה עשיתי
* הערכת הפתרון לעומת התכנון והמלצות לשיפורו
* תודות חשוב מאוד לחברים, למשפה, למורים...

יש למלא עד עמוד ולא להתבייש בכתיבה – כאן לא המקום לצניעות.

**פרק ט' - UELP – ביבליוגרפיה**

1. רקע תיאורטי
2. ספרות מקצועית ספציפית לנושא העבודה (רשימת ספרים, מאמרים והפנייה לכתובות אתרים המכילים חומר רקע ששימש לפיתוח העבודה)
3. הערה: לא מספיק להתייחס לספרים המיועדים ללימוד שפה או כלים יישומיים ומדריכים למיניהם.
4. קישורים לאתרים באינטנרט – קישורים ל stackOverflow עם הסבר על הנושא שנחקר (לא תחביר...)

**נספח א – החקר שבוצע בפרויקט**