**עמוד השער**

תמונה

**יכלול:**

1. לוגו של בית הספר
2. לוגו של האפליקציה/יישום/מערכת
3. נושא העבודה
4. שם התלמיד
5. ת"ז של התלמיד
6. שם בית הספר ועיר
7. שמות המנחים + המורים המלווים
8. מועד הגשת המסמך (חודש ושנה)

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

פרק א' – מערכת שמירת קבצים מבוזרת – ייזום

**1. תיאור ראשוני:**

פעמים רבות כאשר תלמידים מגיעים למעבדה נוצרת בעיה בכך שמחשב שתלמיד עבד עליו בשיעור הקודם ונשמרו בו קבצים הכרחיים , נלקח ע"י תלמיד אחר. בשביל לפתור זאת נשתמש במערכת שמירת קבצים מבוזרת. מערכת זאת תשמור כל קובץ שתלמיד שינה או יצר ובכך בכל פעם שהתלמיד יתחבר ממחשב אחר כאשר המחשב שעבד עליו בשיעור הקודם פתוח יהיה לו את האפשרות לגשת ולשנות את הקבצים.

בחרתי בפרויקט זה משום שרציתי פרויקט שיתעסק בעבודה עם הרשת מכיוון שנושא זה מאוד מעניין אותי, בנוסף נתקלתי בבעיה זו של שמירת הקבצים במחשב אחד שלא יכולתי לגשת אליהם מכיוון שתלמיד אחר התיישב במקום של המחשב שעבדתי עליו בשיעור הקודם.

האתגר המרכזי של הפרויקט יהיה לגשת אל הקבצים ללא שרת. תחילה אבנה את הפרויקט בצורה של שרת שיודיע ללקוח (המיוצג כמחשב) איפה נמצאים הקבצים שלו, לשם הלקוח יצטרך לגשת ולקחת אותם. כדי לגשת אל הקבצים ללא שרת הלקוח יצטרך לשלוח broadcast לכל המחשבים ברשת ולשאול אצל מי נמצא הקבצים שלו. לבנות את בקשת הbroadcast ושליחת הקבצים יהיו האתגר המרכזי בפרויקט. בנוסף שמירת הקבצים בצורה בטוחה שרק הלקוח המתאים יוכל לגשת אליהם היא אתגר נוסף.

**2. הגדרת הלקוח:**

הפרויקט מיועד למעבדות מחשבים באוניברסיטאות ובבתי ספר בהם אין מחשב אישי מוגדר לכל תלמיד.

**3. הגדרת יעדים/מטרות:**

יש לפרויקט 2 מטרות מרכזיות:

1. לשמור את הקבצים בצורה בטוחה, רק התלמיד הספציפי יוכל לגשת לקבצים שלו, אין אנשים אחרים שיקבלו הרשאה לגשת אל הקבצים של אותו תלמיד
2. התלמיד יוכל לגשת אל הקבצים שלו מכל מחשב במעבדת המחשבים כל עוד המחשבים שעבד עליהם בפעמים הקודמות דלוקים.

**4.** **בעיות, תועלות וחסכונות:**

הבעיה שנוצרה היא שתלמיד שעובד על מחשב מסוים בשיעור אחד אך בשיעור אחר נלקח לו המחשב, אין לתלמיד אפשרות לגשת אל הקבצים שעבד עליו בפעם הקודמת ובכך הוא יצטרך ליצור את הקבצים מחדש ובכך לבזבז המון זמן.

התועלת של הפרויקט היא עזרה לתלמידים לשמור את הקבצים שלהם בצורה יותר נוחה ללא שימוש באביזרים חיצוניים כגון דיסק און קי או אפילו במייל ובמתן אפשרות לתלמידים לגשת לקבצים שלהם ששמורים על מחשבים שונים ברחבי הרשת מכל מחשב אחר ברשת ללא צורך בהעלאתם לאחסון מרכזי.

פתרון קיים שעונה על אותו הצורך של המערכת הוא דיסק און קי. דיסק און קי גם כן מטרתו לשמור את הקבצים שעליו עבד התלמיד אך בשונה מהמערכת הדיסק און קי הוא כלי פיזי לכן אפשר לשכוח אותו ובנוסף צריך לשמור את הקבצים בצורה ידנית ברגע שהתלמיד מתנתק מהמחשב.

פתרון קיים נוסף שעונה על אותו הצורך של המערכת הוא תיקייה משותפת ברשת. בדומה למערכת תיקייה משותפת ברשת שומרת את הקבצים של המשתמש והתלמיד יוכל לגשת אל הקבצים מכל מקום במעבדה לעומת זאת הקבצים של התלמיד אינם יהיו מוגנים לכן כל אדם שמתחבר לרשת יוכל למחוק את הקבצים של התלמיד.

פתרון קיים נוסף שעונה על אותו הצורך של המערכת הוא מקום אחסון ברשת לדוגמא ענן, אבל פתרון זה מצריך את התלמיד בביצוע פעולות של העלאת והורדת קבצים לענן וכן ברישום לענן כלשהו.

הפתרון שלי לא מצריך מהמשתמש לשנות את הרגליו אלא דואג לאפשר לו לגשת לקבצים עליהם עבד במקומות שונים ברשת מכל מחשב אחר ברשת.

**5. מגבלות בהגדרת המערכת:**

הפרויקט לא מתעסק עם טכנולוגיה חדשה ובלתי מוכרת.

מכיוון שהרשתות של בתי הספר הם רשתות סגורות אין אפשרות לתלמיד לגשת אל הקבצים שלו מביתו לכן מערכת שמירת הקבצים תעבוד רק בבית הספר.

**6. תיחום הפרויקט:**

הפרויקט עובד במיוחד עם רשתות מכיוון שהעברת הקבצים ובקשת הקבצים תעבור בתוך רשת ספציפית. בנוסף הפרויקט יעבוד עם מערכת הפעלה ווינדוס משום שבפרויקט אנחנו עובדים עם העברת קבצים וכן יעשה שימוש ב thread-ים למיקבול תהליכים ועוד. הפרויקט יעבוד גם עם קריפטוגרפיה מכיוון שנצטרך להצפין את המידע שמועבר בתוך הרשת של הקבצים.

**7. קיימות כדור הארץ:**

כרגע הפתרון העיקרי לבעיית שמירת הקבצים הוא דיסק און קי. הייצור של דיסק און קי עולה כסף ומשאבים מזהמים לכן מערכת שמירת הקבצים עדיפה בהרבה ושומרת על כדור הארץ.

**פרק ב' - 'שם המערכת' - אפיון**

**1. פרוט המערכת:**

פעמים רבות כאשר תלמידים מגיעים למעבדה נוצרת בעיה בכך שמחשב שתלמיד עבד עליו בשיעור הקודם ונשמרו בו קבצים הכרחיים , נלקח ע"י תלמיד אחר. בשביל לפתור זאת נשתמש במערכת שמירת קבצים מבוזרת. מערכת זאת תשמור כל קובץ שתלמיד שינה או יצר ובכך בכל פעם שהתלמיד יתחבר ממחשב אחר כאשר המחשב שעבד עליו בשיעור הקודם פתוח ומחובר לרשת יהיה לו את האפשרות לגשת ולשנות את הקבצים.

תחילה אבנה את הפרויקט בצורה של שרת שיודיע ללקוח (המיוצג כמחשב) איפה נמצאים הקבצים שלו, לשם הלקוח יצטרך לגשת ולקחת אותם. כדי לגשת אל הקבצים ללא שרת הלקוח יצטרך לשלוח broadcast לכל המחשבים ברשת ולשאול אצל מי נמצא הקבצים שלו.

**3. יכולות המשתמש:**

התחברות למערכת – כאשר משתמש פותח את המחשב יופיע לו 3 אפשרויות. או להתחבר למערכת עם שם משתמש וסיסמא, ליצור משתמש חדש או להיכנס כאורח (לבחור באופציה זו לא ייתן למשתמש את האפשרות להשתמש ביכולות הבאות).

שמירת קבצים - כמו בכל מחשב רגיל, המשתמש יוכל לשמור את הקבצים שהוא יוצר ולשמור אותם בדיסק D ובכך הקבצים שלו ישמרו לפעם הבאה שיתחבר למערכת.

שינוי קבצים שמורים - בדומה לשמירת הקבצים שהמשתמש יצר, המשתמש יוכל לשנות את תוכן, שמות הקבצים ששמר והשינויים ישמרו לפעם הבאה שיתחבר למערכת.

קבלת קבצים שמורים - כאשר משתמש מתחבר למערכת, יופיע למשתמש כל הקבצים אשר עבד עליהם בפעמים קודמות בכל המחשבים הפזורים ברשת.

**4. פירוט הבדיקות ('קופסא שחורה')**

| **מספר** | **שם הבדיקה (שם שיעיד על תוכן הבדיקה)** | **מה אמורה לבדוק** | **איך מתכננים לבדוק**  **(לתאר בפירוט את שלבי הבדיקה)** |
| --- | --- | --- | --- |
| 1 | יצירת קבצים | לאחר התחברות למערכת, האם כאשר יוצרים במחשב א' קובץ, הקובץ יימצא במראה הקבצים במחשב ב'. | מתחברים למערכת במחשב א', יוצרים קובץ. לאחר מכן מתחברים למערכת במחשב ב' ובודקים האם הקובץ נמצא במראה הקבצים במחשב ב' |
| 2 | מחיקת קבצים | לאחר התחברות למערכת, האם כאשר מוחקים במחשב א' קובץ, הקובץ לא יימצא במראה הקבצים במחשב ב'. | עושים זאת לאחר בדיקה מס' 1.  מתחברים למערכת במחשב א', מוחקים את הקובץ שיצרנו. לאחר מכן מתחברים למערכת במחשב ב' ובודקים האם הקובץ נעלם ממראה הקבצים. |
| 3 | שינוי שם קובץ | לאחר התחברות למערכת, האם כאשר משנים של קובץ במחשב א', שם הקובץ ישתנה במראה הקבצים במחשב ב'. | עושים זאת לאחר בדיקה מס' 1.  מתחברים למערכת במחשב א', משנים את שם הקובץ שיצרנו. לאחר מכן מתחברים למערכת במחשב ב' ובודקים האם שם הקובץ ישתנה במראה הקבצים. |
| 4 | שינוי תוכן קובץ | לאחר התחברות למערכת, האם כאשר משנים תוכן של קובץ במחשב א', תוכן הקובץ ישתנה במחשב ב'. | עושים זאת לאחר בדיקה מס' 1.  מתחברים למערכת במחשב א', כותבים בתוך הקובץ (txt) כמה מילים. לאחר מכן מתחברים למערכת במחשב ב' ובודקים האם מה שכתבנו בקובץ במחשב א' נמצא בקובץ במחשב ב'. |
| 5 | סנכרון פעולות | נבדוק האם כל הפעולות שכתבנו לעיל עובדות כאשר עושים אותם באותו זמן ושום דבר אינו מתבלבל | ניצור 2 קבצים במחשב א' נתחבר למחשב ב' ונבדוק כי שניהם נוצרו. לאחר מכן שוב נתחבר למחשב א' ניצור עוד קובץ, נשנה את שם הקובץ של אחד משני הקבצים ואת תוכן הקובץ ונמחק את הקובץ הנוסף. נתחבר למחשב ב' ונבדוק שכל הפעולות שעשינו התבצעו כראוי |
| 6 | זמני תגובה | האם בעבודה עם קבצים גדולים יש פגיעה בזמני התגובה של המערכת | נשמור 5 קבצים גדולים במחשב ב' דרך מחשב א' ונבדוק האם במחשבים ב' בו מחובר משתמש אחר ברשת יש עיכוב בפעולות רגילות |
| 7 | עומס | האם כשיש משתמשים רבים במערכת, הביצועים עדיין טובים | נחבר 4 משתמשים למערכת מ8 מחשבים שונים ונבצע כמה פעולות שמתוארות לעיל ונבדוק האם הפעולות מתבצעות כראוי |
| 8 | יציבות | האם המערכת נשארה יציבה אחרי התחברויות והתנתקויות של משתמשים | נחזור על פעולה מס' 7 ומדי פעם נחבר וננתק משתמשים לבדוק האם המערכת אינה קורסת |
| 9 | אבטחה | האם בעת העבודה על הקבצים ממחשב אחר המידע לא זולג החוצה | נפתח wireshark נחזור על פעולה 1 ונבדוק האם המידע שהועבר ברשת מוצפן או לא |
| 10 | העלאת קובץ | האם כאשר מעלים קובץ אפשרי לפתוח אותו דרך המערכת | מתחברים למחשב מס 1 מעלים קובץ מן המחשב, מתחברים למחשב מס 2 ובודקים אם הקובץ קיים והתוכן שלו נכון |
| 11 | הורדת קובץ | האם כאשר מורידים קובץ הוא מופיע במראה הקבצים של ווינדוס | מתחברים למחשב מס 1 מורידים קובץ ובודקים האם הוא נפתח במראה הקבצים של ווינדוס |
| 12 | העברת קובץ | האם כאשר מעבירים קובץ ממחשב אחד למחשב אחר הקובץ מופיע במחשב האחר | מתחברים למחשב מס 1 מעבירים קובץ מהמחשב למחשב מס 2. בודקים האם הקובץ נמחק מהמחשב. מתחברים למחשב מס 2 ובודקים האם הקובץ נמצא |
| 13 | שכפול קובץ | האם כאשר מעתיקים קובץ ממחשב אחד למחשב אחר הקובץ מופיע במחשב האחר | מתחברים למחשב מס 1 מעתיקים קובץ מהמחשב למחשב מס 2. מתחברים למחשב מס 2 ובודקים אם הקובץ נמצא |
| 14 | הרשמה למערכת | האם כאשר נרשמים למערכת עם שם משתמש שלא קיים מכניס למערכת הקבצים | נרשמים למערכת עם שם משתמש שלא קיים ובודקים אם נכנס למערכת קבצים. מריצים את המערכת מחדש ומנסים להירשם עם אותו משתמש ובודקים אם מוצגת הודעה ראויה שהמשתמש כבר קיים |
| 15 | התחברות למערכת | האם כאשר מתחברים למערכת עם שם משתמש קיים מכניס למערכת קבצים | מתחברים למערכת עם שם משתמש קיים ובודקים אם נכנס למערכת הקבצים. מריצים את המערכת מחדש ומנסים להיכנס עם שם משתמש שלא קיים ובודקים שמוצגת הודעה ראויה שהמשתמש לא קיים. |

**5. תכנון לוח זמנים לפרויקט**:

| **פעילות** | **זמן התחלה מתוכנן** | **זמן סיום מתוכנן** | **זמן התחלה בפועל** | **זמן סיום בפועל** | **הערות** |
| --- | --- | --- | --- | --- | --- |
| יזום | 6.11.2023 | 25.11.2023 | 6.11.2023 | 9.11.2023 |  |
| אפיון | 8.11.2023 | 1.12.2023 | 10.11.2023 |  |  |
| ניתוח | 11.11.2023 | 15.12.2023 |  |  |  |
| עיצוב | 1.12.2023 | 15.1.2023 |  |  |  |
| קוד (לפרוט לאבנים גדולות) | 1) בניית שרת לקוח בסיסי, עבודה בסיסית עם קבצים  2) בניית בסיס נתונים  3) ממשק לקוח 1, עבודה עם מחשב אחד, מראה את הקבצים ששמר פעמים קודמות  4) עבודה עם 2 לקוחות. התחברות עם מחשב אחר ומראה קבצים שעבד פעמים קודמות  5) עבודה עם פעולות של קבצים  6) שרת מרובה לקוחות  7) הצפנה  8) החלפת שרת בבקשת ברודקאסט  9) יצירת מערכת קבצים חיה (live) שכאשר משנים\יוצרים\מוחקים קובץ תוצאת הפעולה תופיע ישירות בכל המחשבים בהם קבצים אלה פתוחים. | 1) 20.1.2023  2) 24.1.2023  3) 1.2.2023  4) 7.2.2023  5) 12.2.2023  6) 15.2.2023  7) 17.2.2023  8) 25.2.2023  9) 1.3.2023 |  |  |  |
| גרסה ראשונית | 15.1.2023 | 1.3.2023 |  |  |  |
| מסמך בדיקות | 1.4.2023 | 10.4.2023 |  |  |  |
| מדריך למשתמש | 10.4.2023 | 1.5.2023 |  |  |  |
| גרסה סופית | 1.5.2023 | 10.5.2023 |  |  |  |
| סגירת תיק פרויקט | 1.5.2023 | 10.5.2023 |  |  |  |
| מצגת הסבר | 1.5.2023 | 10.5.2023 |  |  |  |

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

**6. 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 |  |  |
| אי הצלחת עבודה עם מערכת הקבצים של ווינדוס | לא אצליח להראות בדרך הנחוצה את השינויים בקבצים ועוד. | בינוני | * לנסות את העבודה עם הקבצים כמה שיותר מוקדם * במקרה הגרוע לבנות gui נפרד. |  |  |
| אי הצלחת מימוש הצפנה | כל המידע שמועבר ברשת לא יצליח לעבור מוצפן | קשה | * ניסיונות של כמה סוגי הצפנה אם ההצפנה הרצויה לא עובדת * לחקור יותר לעומק על סוגי הצפנה וההצפנה הנכונה ביותר למקרה הספציפי שלי |  |  |
| ניהול קוד לא מאורגן | ניהול הקוד יהיה מבולגן ויכול לגרום לאי עמידה בזמנים | בינוני | * שמירה על קוד מאורגן מתחילת הפרויקט ועד סופו, הערות. * יצירת מחלקות ושמירה כללית מאורגנת של איפה נמצא הקוד |  |  |
| מהימנות מראה הקבצים | לא ניתן ליצור מראה קבצים ללקוח שתואם למציאות | קשה | * התנהלות עם מראה הקבצים יהיה עדיפות ראשונית בפרויקט * נעשה במהלך כל הפרויקט בדיקות רבות שיבדקו את מהימנות מראה הקבצים |  |  |
| תקשורת | לא ניתן לגשת לקבצים שלי ברחבי הרשת | קשה | * עבודה עם tcp כדי שלא יאבד מידע ברשת והמידע יעבור בצורה מהימנה * נעבוד עם wirkeshark במהלך הפרויקט כדי לבדוק את התקשורת |  |  |

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

**פרק ג' - 'שם המערכת' - מסמך ניתוח**

צד לקוח\שרת:

שם היכולת: **שליחת נתונים – צד לקוח\שרת**

מהות היכולת: שליחת נתונים כללית לפי פרוטוקול והצפנה

אוסף יכולות:

* בניית בקשה לפי פרוטוקול
* הצפנה
* שליחת נתונים לשרת\לקוח

רשימת אובייקטים: פרוטוקול, הצפנה, תקשורת

צד לקוח\שרת:

שם היכולת: **פירוק מידע שהתקבל – צד לקוח\שרת**

מהות היכולת: פירוק נתונים כללי לפי פרוטוקול והצפנה

אוסף יכולות:

* קבלת נתונים מלקוח\שרת
* פענוח
* פירוק לפי פרוטוקול

רשימת אובייקטים: פרוטוקול, פיענוח, תקשורת

צד לקוח:

שם היכולת: **הרשמה למערכת – צד לקוח**

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

אוסף יכולות:

* ממשק משתמש – מסך הרשמה
* קליטת נתונים
* בדיקת תקינות
* שליחת נתונים
* פירוק מידע שהתקבל
* הצגת התשובה למשתמש

רשימת אובייקטים: ממשק משתמש

צד שרת:

שם היכולת: **הרשמה למערכת – צד שרת­­**

מהות היכולת: רישום משתמש חדש במערכת

אוסף יכולות:

* פירוק מידע שהתקבל
* בדיקה מול בסיס הנתונים
* גיבוב סיסמא
* הוספה לבסיס נתונים
* שליחת נתונים

רשימת אובייקטים: בסיס נתונים

**צד לקוח:**

שם היכולת: **התחברות למערכת – צד לקוח**

מהות היכולת: התחברות משתמש קיים

אוסף יכולות:

* הרשמה למערכת – צד לקוח
* ממשק משתמש – מסך התחברות
* קליטת נתונים
* בדיקת תקינות
* שליחת נתונים
* פירוק מידע שהתקבל
* הצגת התשובה למשתמש

רשימת אובייקטים: ממשק משתמש

צד שרת:

שם היכולת: **התחברות למערכת – צד שרת**

מהות היכולת: התחברות משתמש קיים במערכת

אוסף יכולות:

* פירוק מידע שהתקבל
* גיבוב סיסמא
* בדיקה מול בסיס הנתונים
* שליחת נתונים

רשימת אובייקטים: בסיס נתונים

**צד לקוח:**

שם היכולת: **הצגת רשימת קבצים – צד לקוח**

מהות היכולת: להציג למשתמש את כל הקבצים שלו במערכת

אוסף יכולות:

* התחברות למערכת – צד לקוח
* פירוק מידע שהתקבל
* ממשק משתמש – מראה קבצים

רשימת אובייקטים: ממשק משתמש

**צד שרת:**

שם היכולת: **בניית רשימת קבצים – צד שרת**

מהות היכולת: בניית הודעה הכוללת את שמות קבצי המשתמש ומיקומם

אוסף יכולות:

* התחברות למערכת – צד שרת
* שליפת נתונים מבסיס הנתונים לפי שם משתמש
* בדיקת MAC קיים ברשת ועדכון רשימה בהתאם
* שליחת נתונים

רשימת אובייקטים: בסיס נתונים, MAC, תקשורת, רשימת קבצים

**צד לקוח :**

שם היכולת: **ניטור קובץ**

מהות היכולת: תפיסת אירועים בתיקייה המנוטרת ושליחתם לשרת

אוסף יכולות:

* האזנה
* תפיסת אירוע רלוונטי
* שליחת נתונים לשרת

**צד לקוח – כללי:**

שם היכולת: **יצירת קובץ – צד לקוח כללי**

מהות היכולת: קבלת קובץ

אוסף יכולות:

* הצגת רשימת קבצים – צד לקוח

**צד לקוח – בעל קובץ:**

שם היכולת: **יצירת קובץ – צד לקוח יוצר קובץ**

מהות היכולת: יצירת קובץ

אוסף יכולות:

* התחברות למערכת – צד לקוח
* הצגת רשימת קבצים – צד לקוח
* בחירת מיקום קובץ – ממשק משתמש
* בחירת שם קובץ – ממשק משתמש
* יצירת קובץ
* שליחת נתונים

רשימת אובייקטים: ממשק משתמש, קובץ

**צד שרת:**

שם היכולת: **יצירת קובץ – צד שרת**

מהות היכולת: יצירת קובץ והצגת הקובץ אצל כל הלקוחות

אוסף יכולות:

* התחברות למערכת – צד שרת
* בניית רשימת קבצים – צד שרת
* בדיקת תקינות שם ומיקום הקובץ
* פירוק מידע שהתקבל
* שליחת נתונים (לקוח כללי)

רשימת אובייקטים: קובץ

**צד לקוח - כללי:**

שם היכולת: **הצגת תוכן קובץ – צד לקוח כללי**

מהות היכולת: להוריד ולפתוח קובץ למשתמש

אוסף יכולות:

* התחברות למערכת – צד לקוח
* הצגת רשימת קבצים – צד לקוח
* בחירת קובץ – ממשק משתמש
* שליחת נתונים
* פירוק מידע שהתקבל
* הצגת תוכן הקובץ

**צד לקוח – בעל קובץ:**

שם היכולת: **הצגת תוכן קובץ – צד לקוח בעל קובץ**

מהות היכולת: לשלוח קובץ לשרת

אוסף יכולות:

* פירוק מידע שהתקבל
* שליחת נתונים (קובץ)

רשימת אובייקטים לקוח: ממשק משתמש, קובץ

**צד שרת:**

שם היכולת: **הצגת תוכן קובץ – צד שרת**

מהות היכולת: לשלוח למשתמש תוכן קובץ

אוסף יכולות:

* התחברות למערכת – צד שרת
* בניית רשימת קבצים – צד שרת
* פירוק מידע שהתקבל
* שליחת נתונים (לבעל קובץ)
* פירוק מידע שהתקבל
* שליחת נתונים (למבקש הקובץ)

רשימת אובייקטים: קובץ

**צד לקוח - כללי:**

שם היכולת: **שינוי שם קובץ – צד לקוח כללי**

מהות היכולת: שינוי שם קובץ אצל כל המחשבים

אוסף יכולות:

* התחברות למערכת – צד לקוח
* הצגת רשימת קבצים – צד לקוח
* בחירת קובץ – ממשק משתמש
* קליטת שם הקובץ ובדיקת תקינות
* שליחת נתונים
* הצגת רשימת קבצים – צד לקוח

**צד לקוח – בעל קובץ:**

שם היכולת: **הצגת תוכן קובץ – צד לקוח בעל קובץ**

מהות היכולת: שינוי שם קובץ שקובל

אוסף יכולות:

* פירוק מידע שהתקבל
* שינוי שם קובץ
* ניטור תיקייה

רשימת אובייקטים לקוח: ממשק משתמש, קובץ

**צד שרת:**

שם היכולת: **שינוי שם קובץ – צד שרת**

מהות היכולת: שינוי שם קובץ אצל כל הלקוחות

אוסף יכולות:

* התחברות למערכת – צד שרת
* בניית רשימת קבצים – צד שרת
* פירוק מידע שהתקבל
* שליחת נתונים (בקשה לשנות שם הקובץ אצל הלקוח בעל הקובץ)
* פירוק מידע שהתקבל (ניטור)
* בניית רשימת קבצים – צד שרת

רשימת אובייקטים: קובץ

**צד לקוח - כללי:**

שם היכולת: **מחיקת קובץ – צד לקוח - כללי**

מהות היכולת: מחיקת קובץ אצל כל המחשבים

אוסף יכולות:

* התחברות למערכת – צד לקוח
* הצגת רשימת קבצים – צד לקוח
* בחירת קובץ – ממשק משתמש
* שליחת נתונים
* הצגת רשמית קבצים – צד לקוח

**צד לקוח – בעל קובץ:**

שם היכולת: **מחיקת קובץ – צד לקוח בעל קובץ**

מהות היכולת: מחיקת קובץ שקובל

אוסף יכולות:

* פירוק מידע שהתקבל
* מחיקת קובץ
* ניטור תיקייה

רשימת אובייקטים לקוח: ממשק משתמש, קובץ

**צד שרת:**

שם היכולת: **מחיקת קובץ – צד שרת**

מהות היכולת: מחיקת קובץ אצל כל המשתמשים

אוסף יכולות:

* התחברות למערכת – צד שרת
* בניית רשימת קבצים – צד שרת
* פירוק מידע שהתקבל
* שליחת נתונים (בעל קובץ)
* פירוק מידע שהתקבל
* בניית רשימת קבצים – צד שרת

רשימת אובייקטים: קובץ

**צד לקוח - כללי:**

שם היכולת: **שינוי תוכן קובץ – צד לקוח**

מהות היכולת: שינוי תוכן קובץ אצל מחשב מקורי

אוסף יכולות:

* התחברות למערכת – צד לקוח
* הצגת רשימת קבצים – צד לקוח
* הצגת תוכן קובץ - לקוח
* כתיבת תוכן לשינוי
* שליחת נתונים

**צד לקוח – בעל קובץ:**

שם היכולת: **שינוי תוכן קובץ – צד לקוח בעל קובץ**

מהות היכולת: שינוי תוכן קובץ מקומי

אוסף יכולות:

* פירוק מידע שהתקבל
* שינוי תוכן קובץ
* שליחת נתונים (אישור)

רשימת אובייקטים לקוח: ממשק משתמש, קובץ

**צד שרת:**

שם היכולת: **שינוי תוכן קובץ – צד שרת**

מהות היכולת: שינוי תוכן קובץ אצל כל הלקוחות

אוסף יכולות:

* התחברות למערכת – צד שרת
* בניית רשימת קבצים – צד שרת
* הצגת תוכן קובץ – צד שרת
* פירוק מידע שהתקבל
* שליחת נתונים (לבעל קובץ)
* פירוק מידע שהתקבל

רשימת אובייקטים: קובץ

**צד לקוח - כללי:**

שם היכולת: **העברת קובץ – צד לקוח כללי**

מהות היכולת: העברת קובץ ממיקום אחד לאחר

אוסף יכולות:

* התחברות למערכת – צד לקוח
* הצגת רשימת קבצים – צד לקוח
* בחירת קובץ – ממשק משתמש
* בחירת מיקום חדש לקובץ – ממשק משתמש
* שליחת נתונים
* הצגת רשימת קבצים – צד לקוח

**צד לקוח – בעל קובץ:**

שם היכולת: **העברת קובץ – צד לקוח בעל קובץ**

מהות היכולת: העברת מיקום הקובץ

אוסף יכולות:

* פירוק מידע שהתקבל
* העברת מיקום
* ניטור תיקייה

רשימת אובייקטים לקוח: ממשק משתמש, קובץ

**צד שרת:**

שם היכולת: **העברת קובץ – צד שרת**

מהות היכולת: העברת קובץ ממיקום אחד לאחר

אוסף יכולות:

* התחברות למערכת – צד שרת
* בניית רשימת קבצים – צד שרת
* פירוק מידע שהתקבל
* בדיקת תקינות מקום שהועבר אליו
* שליחת נתונים (לבעל קובץ)
* פירוק מידע שהתקבל
* בניית רשימת קבצים – צד שרת

רשימת אובייקטים: אין

**צד לקוח - כללי:**

שם היכולת: **שכפול קובץ – צד לקוח כללי**

מהות היכולת: שכפול קובץ למקום מבוקש

אוסף יכולות:

* התחברות למערכת – צד לקוח
* הצגת רשימת קבצים – צד לקוח
* בחירת קובץ – ממשק משתמש
* בחירת מיקום חדש לקובץ משוכפל – ממשק משתמש
* בחירת שם קובץ לקובץ משוכפל – ממשק משתמש
* שליחת נתונים
* הצגת רשימת קבצים צד לקוח

**צד לקוח – בעל קובץ:**

שם היכולת: **שכפול קובץ – צד לקוח בעל קובץ**

מהות היכולת: שכפול קובץ מבוקש

אוסף יכולות:

* פירוק מידע שהתקבל
* שכפול הקובץ למיקום מבוקש
* ניטור תיקייה

רשימת אובייקטים לקוח: ממשק משתמש, קובץ

**צד שרת:**

שם היכולת: **שכפול קובץ – צד שרת**

מהות היכולת: שכפול קובץ למקום מבוקש

אוסף יכולות:

* התחברות למערכת – צד שרת
* בניית רשימת קבצים – צד שרת
* פירוק מידע שהתקבל
* בדיקת תקינות מקום שהועבר אליו
* בדיקת תקינות שם הקובץ
* שליחת נתונים (בעל קובץ)
* פירוק מידע שהתקבל
* בניית רשימת קבצים - צד שרת

רשימת אובייקטים: אין

**צד לקוח – כללי:**

שם היכולת: **יצירת תיקייה – צד לקוח כללי**

מהות היכולת: הצגת תיקייה שנוצרה

אוסף יכולות:

* הצגת רשימת קבצים – צד לקוח

**צד לקוח – בעל תיקייה:**

שם היכולת: **יצירת תיקייה – צד לקוח יוצר תיקיה**

מהות היכולת: יצירת תיקייה

אוסף יכולות:

* התחברות למערכת – צד לקוח
* בחירת מיקום התיקייה – ממשק משתמש
* בחירת שם תיקייה – ממשק משתמש
* יצירת תיקייה
* ניטור תיקייה

רשימת אובייקטים: ממשק משתמש

**צד שרת:**

שם היכולת: **יצירת תיקייה – צד שרת**

מהות היכולת: יצירת תיקייה

אוסף יכולות:

* התחברות למערכת – צד שרת
* בניית רשימת קבצים – צד שרת
* בדיקת תקינות שם ומיקום התיקייה
* פירוק מידע שהתקבל
* בניית רשימת קבצים – צד שרת

רשימת אובייקטים: אין

**צד לקוח - כללי:**

שם היכולת: **שינוי שם תיקייה – צד לקוח**

מהות היכולת: שינוי שם תיקייה אצל כל המחשבים

אוסף יכולות:

* התחברות למערכת – צד לקוח
* הצגת רשימת קבצים – צד לקוח
* בחירת תיקייה – ממשק משתמש
* קליטת שם התיקייה ובדיקת תקינות
* שליחת נתונים
* הצגת רשימת קבצים – צד לקוח

רשימת אובייקטים: ממשק משתמש

**צד לקוח – בעל תיקייה:**

שם היכולת: **שינוי שם תיקייה – צד לקוח בעל תיקייה**

מהות היכולת: שינוי שם תיקייה

אוסף יכולות:

* פירוק מידע שהתקבל
* שינוי שם תיקייה
* ניטור תיקייה

**צד שרת:**

שם היכולת: **שינוי שם תיקייה – צד שרת**

מהות היכולת: שינוי שם תיקייה אצל כל הלקוחות

אוסף יכולות:

* התחברות למערכת – צד שרת
* בניית רשימת קבצים – צד שרת
* פירוק מידע שהתקבל
* שליחת נתונים (בעל תיקייה)
* פירוק מידע שהתקבל
* בניית רשימת קבצים

רשימת אובייקטים: אין

**צד לקוח - כללי:**

שם היכולת: **מחיקת תיקייה – צד לקוח כללי**

מהות היכולת: מחיקת תיקייה אצל כל המחשבים

אוסף יכולות:

* התחברות למערכת – צד לקוח
* הצגת רשימת קבצים – צד לקוח
* בחירת תיקייה – ממשק משתמש
* שליחת נתונים
* הצגת רשימת קבצים – צד לקוח

**צד לקוח – בעל תיקייה:**

שם היכולת: **מחיקת תיקייה – צד לקוח בעל תיקייה**

מהות היכולת: שינוי שם תיקייה

אוסף יכולות:

* פירוק מידע שהתקבל
* מחיקת תיקייה
* ניטור תיקייה

רשימת אובייקטים: ממשק משתמש

**צד שרת:**

שם היכולת: **מחיקת תיקייה – צד שרת**

מהות היכולת: מחיקת תיקייה אצל כל המשתמשים

אוסף יכולות:

* התחברות למערכת – צד שרת
* בניית רשימת קבצים – צד שרת
* פירוק מידע שהתקבל
* שליחת נתונים (בעל תיקייה)
* פירוק מידע שהתקבל
* בניית רשמית קבצים – צד שרת

רשימת אובייקטים: אין

**פרק ד' -** reflection **- העיצוב**

1. תיאור הארכיטקטורה של המערכת המוצעת

* החומרה בה הפרויקט משתמש היא מחשבי windows עם דיסק
* הלקוחות הם מחשבי windows עם דיסק, השרת הוא מחשב windows אשר מחובר לdatabase הלקוחות מתקשרים עם השרת אך גם מתקשרים עם הלקוחות האחרים בצורה של pear to pear. כל זה נמצא ברשת מקומית אחת.



1. תיאור הטכנולוגיה הרלוונטית

שפות התכנות שאשתמש בהם הם python wxpython,.וsql מערכת ההפעלה שנבחרה היא windows מכיוון שהמחשבים במעבדת המחשבים בבית ספר הם של windows, התקשורת מבוססת TCP מכיוון שאנחנו צריכים תקשורת אמינה בפרויקט שלי (אסור שיעבד מידע על קבצים).

תחומי עניין:

1. תקשורת – מימוש שרת מרובה לקוחות, תקשורת pear to pear, שימוש בסוקטים מעל פרוטוקול TCP
2. אבטחה – כל המידע עובר בתקשורת מוצפנת, בבסיס הנתונים מידע רגיש שמור כ hash.
3. מערכות הפעלה – שימוש בתהליכונים למיקבול, עבודה עם מערכת קבצים, שימוש בwinapi לטובת הניטור.
4. גרפיקה – שימוש בwxpython למימוש ממשק המשתמש
5. בסיס נתונים – שימוש בבסיס נתונים טבלאי sql
6. תיאור מודולים בהם נעשה שימוש:



סקירת כל המודולים המרכיבים את המערכת וקשרי הגומלין ביניהם:

**מודלים שאני מייבא:**

|  |  |
| --- | --- |
| מודול | לטובת |
| Socket | עבודה עם סוקטים בתקשורת |
| select | מימוש שרת מרובה לקוחות |
| Ast | לייצוג משתנים כמחרוזות בפרוטוקל |
| Ctypes | ליצירת תיקייה מוחבאת |
| Os | עבודה עם מחרוזות העוסקות במיקום של קבצים ותיקיות |
| Shutil | עבודה ישירה עם קבצים במחשב |
| Popen | בשביל פתיחת קבצים |
| Psutil | להתנהלות עם תהליכים |
| Queue | לשימוש בתור |
| Threading | לניהול תהליכונים |
| Sqlite3 | לניהול בסיס הנתונים |
| Pkcs1\_v1\_5 | להצפנה אסימטרית |
| rsa | ליצירת מפתחות אסימטריים |
| Base64 | להמרת בסיסים בהצפנה |
| Hashlib | למען גיבוב |
| Crypto | למען הצפנה |
| Sys | לסגירת התוכנית |
| time | לעיכוב התוכנית במקרים הנדרשים |
| getnode | לקבלת הmac |
| wx | גרפיקת הפרויקט |
| Pubsub | תקשורת בין גרפיקה ללוגיקה |
| Win32file | ניטור הקבצים |
| Functools | עבודה עם דברים בסיסיים בפייתון |

**מודלים שאני מייצר:**

| serverProtocol | | |
| --- | --- | --- |
| מחלקה עם אוסף פעולות לטיפול בפרוטוקול בצד שרת | | |
| פעולה | טענת כניסה | טענת יציאה |
| Unpack | data – הודעה שהתקבלה בתקשורת | מפרקת לפי הפרוטוקול ומחזירה tuple המכיל:  (opcode, [params]) |
| String\_to\_dict | Data – המחרוזת שמומרת למילון | מחזירה מילון המיוצג ע"י המחרוזת שהתקבלה |
| Dict\_to\_string | Data – המילון שמומר למחרוזת | מחזירה מחרוזת המיוצגת ע"י המילון שהתקבל |
| pack\_status\_register | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_login | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_ask\_file\_tree | Folder – השם של התיקייה שהשרת מבקש לקבל | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_send\_file\_tree | File\_tree - עץ הקבצים | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_create | Status – הצליח/לא הצליח (Boolean)  Path – המיקום של הקובץ  Typ – סוג הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_rename | Status – הצליח/לא הצליח (Boolean)  Path – המיקום של הקובץ  New\_name - שם הקובץ החדש | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_delete | Status – הצליח/לא הצליח (Boolean)  Location – מיקום הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_move | Status – הצליח/לא הצליח (Boolean)  Old\_path – המיקום הישן של הקובץ  New\_path – המיקום החדש של הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_clone | Status – הצליח/לא הצליח (Boolean)  Copy\_from – המיקום ממנו מעתיקים את הקובץ  Copy\_to – המיקום להעתיק אליו את הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_status\_mac | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_do\_rename | Location – מיקום (str)  new\_name – שם חדש (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_do\_delete | Location – מיקום (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_do\_move | old\_location – מיקום ישן (str)  new\_location – מיקום חדש (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_do\_clone | old\_location – מיקום ישן (str)  new\_location – מיקום חדש (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_do\_create | Location – מיקום ליצור (str)  type – סוג יצירה (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_do\_remove | Folder – תיקייה להוריד ממראה הקבצים | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_public\_key | public\_key – מפתח ציבורי (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |

| userClientProtocol | | |
| --- | --- | --- |
| מחלקה עם אוסף פעולות לטיפול בפרוטוקול בצד לקוח משתמש | | |
| פעולה | טענת כניסה | טענת יציאה |
| unpack | data – הודעה שהתקבלה בתקשורת | מפרקת לפי הפרוטוקול ומחזירה tuple המכיל:  (opcode, [params]) |
| String\_to\_dict | Data – המחרוזת שמומרת למילון | מחזירה מילון המיוצג ע"י המחרוזת שהתקבלה |
| Pack\_register | Username – שם משתמש pass – סיסמה של המשתמש | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_sign\_in | Username – שם משתמש pass – סיסמה של המשתמש  Mac – הmac של המשתמש ברגע זה | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_do\_create | Location – מיקום ליצור (str)  type – סוג יצירה (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_do\_rename | Location – מיקום (str)  new\_name – שם חדש (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_do\_delete | Location – מיקום (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_do\_move | old\_location – מיקום ישן (str)  new\_location – מיקום חדש (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_do\_clone | old\_location – מיקום ישן (str)  new\_location – מיקום חדש (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_do\_open\_file | Location – מיקום קובץ (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_change\_file | Location – מיקום קובץ (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |

| generalClientProtocol | | |
| --- | --- | --- |
| מחלקה עם אוסף פעולות לטיפול בפרוטוקול בצד לקוח משתמש | | |
| פעולה | טענת כניסה | טענת יציאה |
| unpack | data – הודעה שהתקבלה בתקשורת | מפרקת לפי הפרוטוקול ומחזירה tuple המכיל:  (opcode, [params]) |
| Pack\_client\_type | Client\_type – סוג הלקוח | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_mac | Mac – הMAC של המחשב | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_rename | Status – הצליח/לא הצליח (Boolean)  Location – מיקום הקובץ  New\_name – השם החדש של הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_delete | Status – הצליח/לא הצליח (Boolean)  Location – מיקום הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_do\_move | New\_path – המיקום החדש של הקובץ  Old\_path – המיקום הישן של הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_status\_move\_to\_client | Status – הצליח/לא הצליח (Boolean)  Moved\_to – המיקום אליו הועבר הקובץ  Moved\_from – המיקום ממנו הועבר הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_move\_to\_server | Status – הצליח/לא הצליח (Boolean)  Old\_path – המיקום הישן של הקובץ  New\_path – המיקום החדש של הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_do\_clone | New\_path – המיקום החדש של הקובץ  Old\_path – המיקום שממנו יש להעתיק את הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_clone\_to\_client | Status – הצליח/לא הצליח (Boolean)  Cloned\_to – המיקום אליו שוכפל הקובץ  Cloned\_from – המיקום ממנו שכפלו את הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_status\_clone\_to\_server | Status – הצליח/לא הצליח (Boolean)  copy\_to – המיקום אליו שוכפל הקובץ  copy\_from – המיקום ממנו שכפלו את הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_create | Status – הצליח/לא הצליח (Boolean)  Location – מיקום הקובץ  Typ – סוג הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_open\_file | Status – הצליח/לא הצליח (Boolean)  Path – מיקום הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_change\_file | Status – הצליח/לא הצליח (Boolean)  Path – מיקום הקובץ | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_file\_tree | Path – המיקום של התיקייה | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_key | Key – מפתח סימטרי | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |

| fileHandler | | |
| --- | --- | --- |
| מחלקה המקבלת עץ קבצים ומפעילה עליו פעולות | | |
| שם משתנה | תפקיד | |
| My\_ip | הIP של המחשב המריץ את התוכנית | |
| username | שם המשתמש המייצג את האובייקט | |
| User\_path | הנתיב למיקום של הקבצים של המשתמש במחשב | |
| פעולה | טענת כניסה | טענת יציאה |
| Init | Username – שם המשתמש | שומרת משתנים נדרשים לפי שם משתמש ומשיגה את הip של המחשב הנוכחי (Settings.get\_ip()) |
| Create\_root | none | יוצרת את התיקייה בה הפרויקט יעבוד משתמשת  ב\_create hidden\_dir, create |
| Is\_local | Path – נתיב מסויים לקובץ | בודק האם הקובץ מקומי או נמצא במחשב אחר ומחזיר אמת או שקר |
| פעולות סטטיות | | |
| Split\_name\_typ | Name – שם קובץ | מחזיר את הקובץ מחולק ל2 שם הקובץ והtyp שלו |
| Build\_name\_for\_file | File\_tree – המילון של הקבצים  New\_dir\_path – שם התיקייה אליו מועבר הקובץ  Path\_of\_file – הנתיב של הקובץ במיקומו הנוכחי  Adding – ההוספה לשם של הקובץ אם כבר שמו תפוס | מחזיר את השם החדש של הקובץ המתאים לתיקייה |
| Direct\_copy\_file | From\_path – המיקום שממנו מעתיקים את הקובץ  To – המיקום אליו מעתיקים את הקובץ | משכפלת את הקובץ ומחזירה האם השכפול עבד או לא |
| rename | Location – מיקום (str)  new\_name – שם חדש (str) | משנה שם קובץ\תיקייה |
| delete | Location – מיקום (str) | מוחק קובץ\תיקייה |
| move | old\_path – מיקום ישן (str)  new\_path – מיקום חדש (str) | מעביר את הקובץ ממקום ישן למקום חדש |
| Split\_path\_last\_part | Path – הנתיב לפרק | מחזירה בצורה של tuple את הנתיב ללא החלק את האחרון ואת החלק האחרון |
| Insert\_ip | Path – הנתיב  Username – שם המשתמש הנמצא בתוך הנתיב  Ip – הip שעל הפונקציה להכניס לנתיב | מחזירה את הנתיב עם הip בתוכו |
| Extract\_ip | Username – השם משתמש שנמצא בנתיב  Path – הנתיב שבו נמצא הIP | מחזירה את הIP שנמצא בתוך הנתיב |
| Remove\_ip | Username – השם משתמש שנמצא בנתיב  Path – הנתיב שנמצא בו הIP | מחזירה את הנתיב ללא הIP |
| Create | path – מיקום ליצור (str)  typ – סוג יצירה (str) | יוצר קובץ\תיקייה במיקום הנתון |
| open\_file | Location – מיקום קובץ (str) | פותחת קובץ |
| \_create\_hidden\_dir | Path – הנתיב לתיקייה | יוצרת תיקייה מוחבאת |
| Get\_user | Path – הנתיב שנמצא בו שם המשתמש | מחזירה את שם המשתמש |
| change\_content | Location – מיקום קובץ (str)  Data – תוכן קובץ חדש (binary) | משנה את תוכן הקובץ הנתון |
| Get\_path\_tree | Path – המיקום של התיקייה שרוצים ליצור ממנה עץ | מחזירה את תוכן התיקייה בצורה של עץ המיוצג כמחרוזת |

| Process\_handler | | |
| --- | --- | --- |
| מחלקה בעלת פעולות המנתחות תהליכים | | |
| שם משתנה | תפקיד | |
| Image\_types | מחזיק את כל הסיומות האפשריות לתמונות | |
| Default\_for\_type | מילון המחבר בין סיומת לשם התהליך המריץ אותו | |
| פעולה | טענת כניסה | טענת יציאה |
| Get\_all\_pid | Process\_name – השם של התהליך | מחזיר רשימה עם כל התהליכים של הסיומת אשר רצים במחשב |
| Get\_process\_name | File\_path – הנתיב של הקובץ | מחזיר את השם של התהליך המתאים לקובץ |
| Wait\_for\_process\_to\_close | Pid – המספר המייצג את התהליך | מחכה שהתהליך ייסגר |

| Db | | |
| --- | --- | --- |
| המחלקה שאחראית על טיפול במבנה הנתונים | | |
| שם משתנה | תפקיד | |
| Conn | מצביע לבסיס הנתונים | |
| DB\_name | שם בסיס הנתונים | |
| Users\_Tbl | טבלת המשתמשים user\_pass | |
| Mac\_tbl | טבלת הmac user\_mac | |
| Curr | המצביע בתוך בסיס הנתונים על מנת לעשות פעולות | |
| Encryption | מטפל בגיבוב הסיסמאות | |
| פעולה | טענת כניסה | טענת יציאה |
| \_\_init\_\_ | none | יוצרת את עצם בסיס הנתונים וקוראת ל \_create\_table |
| \_create\_db | None | יוצר את בסיס הנתונים |
| \_create\_table | none | מתחברת לבסיס הנתונים ויוצרת את טבלת user\_pass אם לא קיימת ואת user\_mac אם לא קיימת |
| \_username\_exists | Username – שם המשתמש | בודק האם שם המשתמש קיים בטבלת user\_pass מחזיר boolean לפי אמת או שקר |
| \_username\_mac\_exists | Username – שם המשתמש  mac – כתובת MAC | בודק האם שם המשתמש קיים בטבלת user\_mac מחזיר boolean לפי אמת או שקר |
| add\_user | Username – שם המשתמש password – סיסמא | מוסיף משתמש לטבלאות אם הוא לא קיים באמצעות \_username\_exists ומחזיר boolean |
| add\_user\_mac | Username – שם המשתמש Mac – כתובת MAC | מוסיף לטבלת user\_mac משתמש וmac ובודק אם קיים באמצעות \_username\_mac\_exists ומחזיר boolean |
| Change\_password | Username – שם המשתמש password – סיסמא | משנה את הסיסמא של המשתמש אם הוא קיים ומחזיר boolean |
| Get\_password | Username – שם המשתמש | מחזיר את הסיסמא של המשתמש אם הוא קיים |
| Get\_macs | Username – שם המשתמש | מחזירה את כל הmacs המיוחסים למשתמש |
| Get\_users | Mac – כתובת mac | מחזירה את כל שמות המשתמשים שמיוחס להם mac כזה |

| asymmetricEncrpytion | | |
| --- | --- | --- |
| המחלקה האחראית על הצפנה א-סימטרית | | |
| שם משתנה | תפקיד | |
| public\_key | מפתח ציבורי | |
| private\_key | מפתח פרטי | |
| פעולה | טענת כניסה | טענת יציאה |
| \_\_init\_\_ | Private\_key – מפתח פרטי (מאותחל כnone)  public\_key – מפתח ציבורי (מאותחל כnone) | יוצרת את עצם ההצפנה הא-סימטרית ויוצרת מפתח ציבורי ומפתח פרטי אם לא התקבלו כפרמטרים |
| encrypt\_msg | Msg - הודעה (str)  receiver\_key – המפתח הציבורי של המקבל | מחזירה את ההודעה מוצפנת |
| decrypt\_msg | Msg - הודעה מוצפנת (str) | מחזירה את ההודעה מפוענחת |
| get\_public\_key | none | מחזירה את המפתח הציבורי של העצם |

| symmetricEncrpytion | | |
| --- | --- | --- |
| המחלקה האחראית על הצפנה סימטרית | | |
| שם משתנה | תפקיד | |
| key | המפתח הסימטרי | |
| encryption | עצם של AES המשתמש בkey בשביל הצפנה ופענוח | |
| פעולה | טענת כניסה | טענת יציאה |
| \_\_init\_\_ | Key – מפתח סימטרי (מאותחל כnone) | יוצרת את עצם ההצפנה הסימטרית ויוצרת מפתח אם לא התקבל כפרמטר |
| encrypt | data - הודעה (str) | מחזירה את ההודעה מוצפנת |
| decrypt | data - הודעה מוצפנת (str) | מחזירה את ההודעה מפוענחת |
| \_pad | Data – מידע לרפד | מחזירה את המידע מרופד |
| \_\_unpad | Data – מידע להוריד ריפוד | מורידה את הריפוד מהמידע |
| hash | Data – מידע לגבב | פעולה חיצונית - מחזירה את המידע מגובב |

| Server\_Comm | | |
| --- | --- | --- |
| המחלקה האחראית על התקשורת של השרת | | |
| שם משתנה | תפקיד | |
| port | הפורט שהשרת משתמש בו | |
| rcv\_q | תור שבו יועברו ההודעות מserverComm לserverLogic | |
| socket | הsocket בוא יעבור המידע בתקשורת | |
| open\_clients | מילון בו מאוכסנים הsocket של הלקוח ובvalue הip המיוחס לsocket ואובייקט ההצפנה הסימטרית  Open\_clients[socket]=[ip, cryptObj] | |
| a\_encrypt | האובייקט של המפתח הא-סימטרי של השרת | |
| is\_running | האם הmain\_loop של המחלקה רץ | |
| Send\_len | כמה בתים יהיה אורך ההודעות המועברות בתקשורת | |
| Receiving\_files | רשימה של ips האמורים לשלוח קובץ | |
| My\_ip | הIP של המחשב המריץ תהליך זה | |
| פעולה | טענת כניסה | טענת יציאה |
| \_\_init\_\_ | Port – הפורט בו עובד השרת  rcv\_q – תור שבו יועברו ההודעות מserverComm לlogic  Send\_len – כמה בתים יהיה אורך ההודעות המועברות בתקשורת | מריצה את \_main\_loop  ומאתחלת משתנים |
| disconnect\_client | Ip – הכתובת של הלקוח אשר צריך לנתק אותו  From\_logic – האם הפעולה נקראה מהלוגיקה | משתמשת ב \_find\_socket\_by\_ip  קוראת ל  \_disconnect\_client  ומודיעה לmainServer באמצעות הrcv\_q  ("close", ip) |
| \_disconnect\_client | client – הסוקט של הלקוח אשר צריך לנתק אותו | מוחקת את הלקוח מכלל הרשימות וסוגרות את הסוקט |
| \_find\_socket\_by\_ip | ip – הכתובת של הלקוח אשר מומר לsocket | מחזירה את הsocket המיוחס לip |
| send | Ip – למי לשלוח את ההודעה  Data – מה לשלוח  Encypt – האם להצפין את ההודעה או לא (Boolean) | שולחת את ההודעה את ההודעה לip בהתאם לפרוקטול (אורך ואז הודעה) |
| Send\_file | Ip – הIP אליו לשלוח את הקובץ  Header – ההקדמה לפני קבלת הקובץ  Data – המידע של הקובץ | שולח את הקובץ ללקוח. משתמש בsend |
| \_key\_exchange \*thread | Client – הסוקט של הלקוח  Ip – הכתובת של הלקוח | שולחת את המפתח הציבורי  מקבלת את המפתח הסימטרי  יוצרת עצם הצפנה  מעדכנת רשימה  Open\_clients[client]=[ip,obj]  קוראת גם ל-\_get\_client\_type() |
| \_Get\_client\_type | Client – הסוקט של הלקוח  Ip – הip של הלקוח  Encryption – עצם ההצפנה של הלקוח | מקבלת מהלקוח את הסוג לקוח ושומרת זאת בopenclients |
| \_main\_loop \*thread | None | מקבלת את כל הלקוחות ומקבלת את כל ההודעות מן הלקוחות ולאחר פענוח ההודעות מעבירה את ההודעות לlogic דרך rcv\_q, קוראת ל\_key\_exchange כאשר לקוח חדש מתחבר |
| close\_server | None | משנה את is\_running לfalse |
| \_Receive\_file \*thread | Client – הסוקט של הלקוח  Header – מה שנדרש לדעת על הקובץ | עושה receive לקובץ, שומרת אותו על המחשב ולאחר מכן שמה את הדברים הנדרשים בrcv\_q לפי פרוטוקול |

| Client\_Comm | | |
| --- | --- | --- |
| המחלקה האחראית על התקשורת של הלקוחות | | |
| שם משתנה | תפקיד | |
| port | הפורט שהשרת משתמש בו | |
| rcv\_q | תור שבו יועברו ההודעות מserverComm לserverLogic | |
| server\_ip | הip של השרת | |
| server | הsocket בוא יעבור המידע בתקשורת | |
| symmetric | אובייקט של ההצפנה סימטרית | |
| Send\_len | כמה בתים יהיה אורך ההודעות המועברות בתקשורת | |
| Client\_type | האם הלקוח הוא כללי או לקוח משתמש | |
| Running | האם הmain\_loop רץ או לא | |
| פעולה | טענת כניסה | טענת יציאה |
| \_\_init\_\_ | Server\_ip – הIP של השרת  Port – הפורט בו עובד השרת  rcv\_q – תור שבו יועברו ההודעת clientComm לlogic  Send\_len – כמה בתים יהיה אורך ההודעות המועברות בתקשורת  Client\_type – סוג הלקוח (כללי או משתמש) | מריצה את \_main\_loop ומאתחלת משתנים |
| send | Data – מה לשלוח  key– מפתח ציבורי של השרת (מאותחל כnone) | שולחת את ההודעה את ההודעה לשרת, אם קיבלה מפתח שולחת באמצעות המפתח שקיבלה אחרת שולחת עם ההצפנה הסימטרית |
| Send\_file | Header – ההקדמה לפני קבלת הקובץ  Data – המידע של הקובץ | שולח את הקובץ לשרת. משתמש בsend |
| \_key\_exchange \*thread | none | מקבלת מהשרת את המפתח הציבורי שלו  יוצרת מפתח סימטרי  ושולחת אותו לשרת קוראת ל\_send\_client\_type() ומשתמשת בAsymmetricEncryption |
| \_send\_client\_type | None | שולחת לשרת את סוג הלקוח |
| \_main\_loop \*thread | None | מתחברת לשרת, בהודעה הראשונה קוראת ל\_key\_exchnage ולאחר מכן מקבלת הודעות מהשרת, מפענחת אותן ומעבירה לlogic דרך rcv\_q |
| \_Receive\_file \*thread | Header – מה שנדרש לדעת על הקובץ | עושה receive לקובץ שומרת אותו על המחשב ולאחר מכן שמה את הדברים הנדרשים בrcv\_q לפי פרוטוקול |
| Close | None | סוגרת את הסוקט ומודיעה ללוגיקה |

| mainServer | | |
| --- | --- | --- |
| המחלקה של השרת הראשית האחראית על הלוגיקה | | |
| שם משתנה | תפקיד | |
| rcv\_q | תור שבו יועברו ההודעות מComm לLogic | |
| Server\_comm | אובייקט מסוג serverComm האחראי על התקשורת של השרת | |
| Commands | מילון ששומר במפתחות שלו opcode של פעולות ובvalue את הפונקציה המתאימה לopcode | |
| Ip\_mac | מילון ששומר במפתחות שלו ip של לקוחות המחוברים למערכת ובvalue את הmac שלהם | |
| Db | עצם של בסיס הנתונים | |
| User\_comps | מילון שבו ישמר כמפתח הip של משתמש מחובר ובvalue רשימה של כל הip של המחשבים הדולקים שעבד עליהם בעבר | |
| Username\_ip | מילון שבו שמור שם המשתמש שהתחבר והip הנוכחי שלו | |
| פעולה | טענת כניסה | טענת יציאה |
| Main | None | מוציאה מrcv\_q את ההודעות שקיבלה מלקוחות וקוראת לפעולות מתאימות |
| Handle\_got\_mac | Client\_ip – הip ממנו נשלח הmac  mac – כתובת MAC | שומרת בip\_mac את הip ואת הmac. לאחר מכן קורא לget\_users ומוסיפה לuser\_comps את הip במקומות הנדרשים, משתמשת גם בusername\_ip, בנוסף אם מחובר משתמש אשר היה מחובר לmac שנשלח נבקש מclient\_ip את עץ הקבצים שלו |
| Handle\_register | Ip – הip אליו ישלח התשובה  User – שם משתמש  Password – סיסמה | קוראת לadd\_user מחזירה תשובה בהתאם לip, באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_sign\_in | Ip – הip אליו ישלח התשובה  Username – שם משתמש  Password – סיסמה  Mac – כתובת | קוראת לget\_password ומחזירה תשובה בהתאם ל .ipאם התשובה חיובית קוראת לadd\_user\_mac ולאחר מכן לget\_macs ונבקש מכל הmacs המיוחסים למשתמש את עץ הקבצים שלהם ונוסיף לuser\_comps את הip של המשתמש ובvalue את הip של כל המחשבים הדולקים שעבד עליהם בעבר באמצעות ip\_mac ומה שקיבל מget\_macs. מוסיף גם לusername\_ip את שם המשתמש והIp |
| Handle\_got\_file\_tree | Ip - הip ממנו נשלח עץ הקבצים  File\_tree – עץ הקבצים | שולחת באמצעות עטיפה בפרוטוקול ושליחה עם send למשתמש הרצוי את עץ הקבצים לאחר שהכנסנו את הip במקומות הנדרשים בעץ |
| got\_rename | Ip – הip ממנו נשלח המידע  Location – מיקום הקובץ\תיקייה  New\_name – שם חדש | שולחת לכתובת שמיקומה ניתן בLocation את הlocation לשנות ואת השם החדש באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_rename | Got\_ip – הip שהשרת ביקש ממנו למחוק  Status – האם שינוי השם עבד או לא  Location – המיקום של הקובץ ששונה שמו  New\_name – השם החדש של הקובץ | מודיעה ללקוח שביקש את שינוי השם האם שינוי השם עבד או לא באמצעות עטיפה בפרוטוקול ושליחה עם send |
| got\_delete | Got\_Ip - הip ממנו ביקשו למחוק  Location – מיקום קובץ\תיקייה | שולחת לכתובת שמיקומה ניתן בLocation את הlocation למחוק את הקובץ\תיקייה באמצעות עטיפה בפרוטוקול ושליחה עם send לאחר מכן |
| Handle\_status\_delete | Got\_ip – הip שהשרת ביקש ממנו למחוק  Status – האם המחיקה עבדה או לא  Location – המיקום של הקובץ שנמחק | מודיעה ללקוח שביקש את המחיקה האם המחיקה עבדה או לא באמצעות עטיפה בפרוטוקול ושליחה עם send |
| got\_move | Ip – הip ממנו נשלח המידע  Move\_from – מיקום הקובץ  Move\_to– מיקום חדש | שולחת לכתובת שמיקומה ניתן בmove\_from Location את move\_from להעביר לmove\_to באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_move | Got\_ip – הIP שיצר את הקובץ  Status – האם היצירה עבדה  Move\_from – המיקום שממנו העבירו את הקובץ  move\_to – המיקום אליו העבירו את הקובץ | מחזירה ללקוח שביקש את ההעברה האם ההעברה עבדה או לא באמצעות עטיפה בפרוטוקול ושליחה עם send |
| got\_clone | Ip – הip ממנו נשלח המידע  Copy\_from – המיקום ממנו לשכפל  Copy\_to – המיקום לשכפל אליו | שולחת לכתובת שמיקומה ניתן בcopy\_from את copy\_from לשכפל לcopy\_to באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_clone | Got\_ip – הIP שיצר את הקובץ  Status – האם היצירה עבדה  Copy\_from – המיקום שממנו שכפלו את הקובץ  Copy\_to – המיקום אליו שיכפלו את הקובץ | מחזירה ללקוח שביקש את השכפול האם השכפול עבד או לא באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Got\_create | Ip – הip ממנו נשלח המידע  Location – מיקום יצירה  Typ – סוג היצירה | שולחת לכתובת שמיקומה ניתן בlocation את הlocation ליצור עם הtyp שהתקבל באמצעות עטיפה בפרוטוקול ושליחה עם send לאחר מכן נשלח לip את סטטוס הפעולה |
| Handle\_status\_create | Got\_ip – הIP שיצר את הקובץ  Status – האם היצירה עבדה  Location – המיקום שבו נוצר הקובץ  Typ – סוג הקובץ\תיקייה | מחזירה ללקוח שביקש את היצירה האם היצירה עבדה או לא באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_disconnect | Ip - הip שהתנתק  Called\_by\_server\_comm – האם נקרא על ידי הcomm | שולחת לכל המשתמשים אשר משתמשים בקבצים של IP למחוק אותו מעץ הקבצים ומוחקת את הip מכל המילונים הנדרשים אם נדרש מיידעת את server\_comm |
| Get\_key\_by\_value | Dic – המילון להשיג ממנו את המפתח  Value – הערך שממנו נשיג את המפתח | מחזיר את המפתח לפי הערך במילון |

| mainGeneralClient | | |
| --- | --- | --- |
| המחלקה של הלקוח שאינו בהכרח משתמש האחראית על הלוגיקה | | |
| שם משתנה | תפקיד | |
| Client\_rcv\_q | תור שבו יועברו ההודעות מclient\_Comm לLogic | |
| Server\_rcv\_q | תור שבו יועברו ההודעות מserver\_Comm לLogic | |
| client | אובייקט מסוג client\_comm האחראי על התקשורת עם השרת | |
| File\_server | אובייקט מסוג server\_comm האחראי על התקשורת עם לקוחות המשתמש | |
| Commands | מילון ששומר במפתחות שלו opcode של פעולות ובvalue את הפונקציה המתאימה לopcode | |
| פעולה | טענת כניסה | טענת יציאה |
| Main | None | מאפסת משתנים ומפעילה פעולות ראשוניות כמו rcv\_comm() |
| Rcv\_comm \*thread | Comm – האובייקט של התקשורת  Q - התור שבו נקבל הודעות מהתקשורת | מקבלת מהתור הודעות ומפנה לפעולות הרצויות |
| Send\_mac | None | שולחת את הmac של הלקוח לשרת באמצעות עטיפה בפרוטוקל ושליחה עם send |
| Handle\_status\_mac | Client\_got – הcomm של הלקוח  Status – האם הפעולה הצליחה | אם הפעולה נכשלה סוגרת את הלקוח |
| Handle\_asked\_file\_tree | Client\_got – האובייקט של הcomm  Folder\_name – שם התיקייה שרוצה לקבל השרת | שולחת לשרת את עץ הקבצים באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_rename | Client\_got – האובייקט של הclient\_comm  Location – מיקום הקובץ\תיקייה שרוצים לשנות  New\_name – שם הקובץ\תיקייה החדש | קוראת לrename בfilehandler ומחזירה תשובה לשרת אם הפעולה הצליחה או לא באמצעות עטיפה בפרוטוקל וsend |
| Handle\_delete | Client\_got – האובייקט של הclient\_comm  Location – מיקום הקובץ\תיקייה שרוצים למחוק | קוראת לdelete בfilehandler ומחזירה תשובה לשרת אם הפעולה הצליחה או לא באמצעות עטיפה בפרוטוקל וsend |
| Move\_from\_server | Client\_got – האובייקט של הclient\_comm  Move\_from – מיקום הקובץ שרוצים להעביר  Move\_to – מיקום הקובץ החדש | אם מיקום הקובץ החדש הוא מקומי קוראת לmove בfilehandler ומחזירה תשובה לשרת אחרת מעבירה את תוכן הקובץ באמצעות pear to pear ובפעולה send\_file אל המחשב הרצוי |
| Move\_from\_client | Got\_ip – הIP ממנו נשלח הקובץ  Server – האובייקט של הcomm  Status – האם ההעברה הצליחה  Move\_to – לאיפה הועבר הקובץ  Move\_from – מאיפה הועבר הקובץ | מחזירה ללקוח האם ההעברה של הקובץ עבדה או לא באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_move | Client\_got – האובייקט של הcomm  Status – האם הפעולה עבדה או לא  Moved\_to – המיקום שהועבר אליו הקובץ  Moved\_from – המיקום ממנו הועבר הקובץ | מעבירה לשרת האם העברת הקובץ הצליחה באמצעות עטיפה בפרוטוקול ושליחה עם send |
| clone\_from\_server | Client\_got – האובייקט של הclient\_comm  copy\_from – מיקום הקובץ שרוצים לשכפל  copy\_to – מיקום הקובץ החדש | אם מיקום הקובץ החדש הוא מקומי קוראת לdirect\_copy\_file בfilehandler ומחזירה תשובה לשרת אחרת מעבירה את תוכן הקובץ באמצעות pear to pear ובפעולה send\_file אל המחשב הרצוי |
| Clone\_from\_client | Got\_ip – הIP ממנו נשלח הקובץ  Server – האובייקט של הcomm  Status – האם ההעברה הצליחה  copy\_to – לאיפה הועבר הקובץ  copy\_from – מאיפה הועבר הקובץ | מחזירה ללקוח האם השכפול של הקובץ עבד או לא באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_clone | Client\_got – האובייקט של הcomm  Status – האם הפעולה עבדה או לא  copied\_to – המיקום ששוכפל אליו הקובץ  copied\_from – המיקום ממנו שוכפל הקובץ | מעבירה לשרת האם שפכול הקובץ הצליח באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_create | Client\_got – האובייקט של הcomm  Location – המיקום של היצירה  Typ – סוג הקובץ | יוצר את הקובץ באמצעות Filehandler.create() ומודיע לשרת על הפעולה באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_open\_file | Got\_ip – הip שביקש את תוכן הקובץ  Server – האובייקט של הserver\_comm  Location – מיקום הקובץ | משיגה את תוכן הקובץ ושולחת ללקוח משתמש את תוכן הקובץ באמצעות עטיפה בפרוטוקל ושליחה בsend |
| Handle\_changed\_file | Got\_ip – הIP ששינה את תוכן הקובץ  Server – האובייקט של הserver\_comm  Location – מיקום הקובץ | בודקת אם שמירת הקובץ הייתה תקינה ושולחת ללקוח משתמש אם הפעולה הייתה תקינה באמצעות עטיפה בפרוטוקל ושליחה בsend |
| Handle\_status\_mac | Server – האובייקט של הcomm  Status – הצליח או לא | אם status אז נמשיך בריצת התהליך אחרת נעצור אותו |

| mainUserClient | | |
| --- | --- | --- |
| המחלקה של הלקוח שהוא גם משתמש | | |
| שם משתנה | תפקיד | |
| graphic\_q | תור שבו יועברו ההודעות מהגרפיקה ללוגיקה | |
| File\_tree | מילון המייצג את כל עץ הקבצים | |
| Server\_rcv\_q | תור שבו יועברו הודעות מהלוגיקה אל הComm | |
| Client | הcomm שמתקשר עם השרת | |
| User\_name | שם המשתמש | |
| Handle\_tree | תור שאליו יגיעו כל עצי הקבצים לקראת הצגתם בגרפיקה והוספתם לfile\_tree | |
| Ip\_comm | מחזיק בkey IP של המחשב המתוקשר באמצעות pear to pear ואת האובייקט של התקשורת שמדבר עם המחשב | |
| Downloads | מילון של מיקומים של קבצים המחכים להורדה והמיקום שלהם האמיתי | |
| Frame | האובייקט של הגראפיקה | |
| פעולה | טענת כניסה | טענת יציאה |
| \_\_init\_\_ | None | מאפסת משתנים מפעילה את הגראפיקה ואת התהליכונים שמדברים עם הגראפיקה ועם התקשורת |
| Monitor\_file \*thread | File\_path – מיקום הקובץ לנתר  End\_q – תור שבו יועבר משהו כאשר יש לסיים את הניתור | מנתרת את הקובץ וקוראת לsave\_file() כאשר המשתמש שומר את הקובץ |
| Folders\_remove | Path – המיקום של האובייקט | מוחקת את האובייקט מfile\_tree ומודיעה לגרפיקה להוריד אותו |
| Folders\_add | Path – מיקום התייקה של האובייקט  Name – השם של הקובץ  Typ – סוג הקובץ | מוסיפה את האובייקט לfile\_tree ואומרת לגרפיקה להוסיף אותו |
| Folders\_rename | Old\_path – המיקום של האובייקט  New\_name – השם החדש של האובייקט | משנה את שם האובייקט בfile\_tree ומודעה לגרפיקה לשנות את שמו |
| Get\_file\_data | Path – מיקום הקובץ | אם הקובץ מקומי פותח את הקובץ או מעביר אותו למקום הרצוי אם המשתמש הוריד את הקובץ (נמצא בdownloads).  אם הקובץ לא מקומי מבקש באמצעות pear to pear את הקובץ באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_open | Comm – האובייקט של התקשורת  Status – האם העברת המידע של הקובץ עבדה או לא  Path – מיקום הקובץ | פותחת את הקובץ למשתמש אחרת קוראת לcall\_error() |
| Save\_file | From\_path – המיקום של הקובץ עם המידע החדש  To\_path – המיקום של הקובץ שבו שומרים את המידע החדש | אם שמירת הקובץ יכולה להתבצע מקומית עושה זאת אחרת שולחת את המידע של from\_path לip שנמצא בto\_path באמצעות עטיפה בפרוטוקול ושליחה עם send\_file() |
| Handle\_status\_save\_file | Status – אם הפעולה עבדה או לא  Path – מיקום הקובץ | אם היה שגיאה כלשהי בשמירת הקובץ קורא לcall\_error() |
| Visualize\_open\_file | File\_path – המיקום של הקובץ לפתיחה  Comm – התקשורת עם המחשב שעליו נמצא הקובץ המקורי | מתחילה ניתור על הקובץ באמצעות monitor\_file() ובאמצעות המחלקה process\_handler מחכה שהקובץ יסגר ע"י המשתמש וסוגרת את התקשורת עם המחשב שנמצא עליו הקובץ |
| Rcv\_graphic \*thread | Q – התור שממנו מועברות הודעות מהגרפיקה | מקבלת הודעות מהגרפיקה עם משתנים ומפנה לפעולות המתאימות |
| Rcv\_comm \*thread | comm – הcomm שמחובר לשרת | מקבל מהשרת הודעות וקורא לפעולות בהתאם |
| Create | Path – מיקום תיקיית האובייקט  Name – שם האובייקט  Typ – סוג האובייקט | אם הpath מקומי יוצרת את האובייקט אחרת מודיעה לשרת ליצור אותו באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_create | Status – אם הפעולה הצליחה  Location – מיקום היצירה  Typ – סוג היצירה | אם הפעולה עבדה קוראת לfolders\_add אחרת קוראת לcall\_error |
| Move\_file | File\_to\_move – המיקום של הקובץ להעביר  Move\_to – לאן להעביר את הקובץ | אם אפשרי להעביר את הקובץ בצורה מקומית מעבירה אחרת מודיעה לשרת על ההעברה באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_move | Status – האם הפעולה עבדה  move\_from – מאיפה הקובץ הועבר  move\_to – לאן הקובץ הועבר | אם הסטטוס חיובי קוראת לfolders\_add() ולאחר מכן לfolders\_remove() אחרת קוראת לcall\_error() |
| Clone\_file | File\_to\_copy – המיקום של הקובץ לשכפל  copy\_to – מיקום השכפול | אם אפשרי לשכפל את הקובץ בצורה מקומית עושה זאת, אחרת מודיעה לשרת על ההעברה באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_clone | Status – האם הפעולה עבדה  Copy\_from – מאיפה הקובץ הועתק  Copy\_to – לאן הקובץ הועתק | אם הסטטוס חיובי קוראת לfolders\_add() אחרת קוראת לcall\_error() |
| Rename | Path – מיקום הקובץ  New\_name – השם החדש של הקובץ | אם אפשרי לשנות את שם הקובץ בצורה מקומית עושה זאת אחרת מודיעה לשרת על שינוי השם באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_rename | Status – האם הפעולה הצליחה  Location – מיקום הקובץ  New\_name – השם החדש של הקובץ | קוראת לfolders\_rename אם הסטטוס חיובי אחרת קוראת לcall\_error |
| Delete | Path – המיקום של הקובץ | אם אפשרי למחוק את הקובץ בצורה מקומית עושה זאת אחרת מודיעה לשרת על המחיקה באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_delete | Status – אם המחיקה עבדה או לא  Location – מיקום האובייקט שנמחק | אם הפעולה עבדה קוראת לfolders\_remove() אחרת קוראת לcall\_error() |
| Get\_file\_tree \*thread | None | מקבל עץ קבצים מהתור handle\_tree מעדכן את file\_tree ומעביר לגרפיקה את העץ שקיבל לעדכון |
| Do\_register | Password – סיסמה | שולחת לשרת נסיון להתחברות באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_status\_register | Success – אם הצליח או לא | מראה ללקוח בצורה גראפית אם הצליח או לא |
| Do\_connect | Password - סיסמה | מוציאה את כתובת הMAC ושולחת לשרת את נסיון ההתחברות (אם פעולה זו לא תתבצע נסגור את התהליך של mainUserClinet) |
| Handle\_status\_login | Success – הצליח או לא | מציגה למשתמש בצורה גראפית אם הצליח להתחבר או לא, מאפסת משתנים עם שם המשתמש, מפעילה את get\_file\_tree() |
| Handle\_got\_file\_tree | Data – עץ הקבצים שקיבל מהשרת | מעביר לתור handle\_tree את העץ שקיבל |
| Call\_error | Error – מה לכתוב בהודעת השגיאה | קוראת לsend\_to\_grpahics עם ההודעה. |
| Refresh\_cursor | None | אומרת לגרפיקה לשנות את הסמן לרגיל |
| Send\_to\_graphics | Flag – הדגל של הפעולה בגרפיקה  Params – הפרמטרים להעביר לגרפיקה | קוראת לrefresh\_cursor וקוראת לפעולה בגרפיקה עם הדגל והמשתנים |

| settings | | |
| --- | --- | --- |
| המחלקה המחזיקה פעולות בסיסיות ומשתנים אשר יש לשנות לפני הרצת הפרויקט | | |
| שם משתנה | תפקיד | |
| Server\_ip | הIP של השרת | |
| Server\_port | הפורט של השרת | |
| Root | המיקום בו הקבצים נשמרים מקומית על המחשב | |
| Pear\_port | הפורט בו מתקשרים מחשבים ללא עזרת השרת | |
| Dir\_working\_on | המיקום בו נמצא הפרויקט כרגע | |
| Tmp\_directory\_path | המיקום בו נשמרים קבצי tmp | |
| Pic\_path | המיקום של התמונות בהם אנחנו משתמשים בפרויקט | |
| פעולה | טענת כניסה | טענת יציאה |
| Get\_ip | none | מחזירה את הIP של המחשב שקרא לפעולה |
| Get\_mac\_address | None | מחזירה את הmac של המחשב שקרא לפעולה |

| My\_frame |
| --- |
| המחלקה האחראית על החלפת הפאנלים (הבסיס של הגרפיקה) |

| MainPanel |
| --- |
| המחלקה האחראית על המסך שבו המשתמש בוחר האם להירשם או להתחבר בגרפיקה |

| RegisterPanel |
| --- |
| המחלקה האחראית על ההרשמות של המשתמש בגרפיקה |

| LoginPanel |
| --- |
| המחלקה האחראית על ההתחברות של המשתמש בגרפיקה |

| CreateFileDialog |
| --- |
| המחלקה האחראית על הכנסת קלטים של שמות מהמשתמש בגרפיקה |

| TreeFrame | | |
| --- | --- | --- |
| המחלקה האחראית על הצגת עץ הקבצים בגרפיקה והעברת נתונים ללוגיקה מהמשתמש | | |
| שם משתנה | תפקיד | |
| Tree | האובייקט של העץ | |
| File\_commands | כל הפקודות שאפשר לעשות על קבצים | |
| Folder\_commands | כל הפקודות שאפשר לעשות על תיקיות | |
| Root | האובייקט שמייצג את ראש העץ | |
| Command\_q | התור שמתקשר עם הלוגיקה | |
| Folders | רשימה של המיקומים של התיקיות בעץ | |
| Image\_size | המשתנה האחראי על גודל האייקונים | |
| Item\_image\_path | מילון שהמפתח שלו הוא איבר בעץ והערך שלו הוא המיקום של התמונה המתאימה לו | |
| Image\_list | רשימה שבה שמורים כל התמונות של האייקונים הנוכחיים | |
| Path\_item | מילון שהמפתח שלו הוא מיקום של אובייקט והאיבר המתאים לו | |
| Forbidden | אוסף של כל התווים האסורים בשמות של אובייקטים | |
| פעולה | טענת כניסה | טענת יציאה |
| Refresh\_cursor | none | משנה את העכבר למצבו הרגיל |
| Send\_to\_logic | Command – הפקודה ללוגיקה  Data – איזה משתנים מועברים עם הפקודה | מעבירה את העכבר לטעינה ומעבירה בcommand\_q ללוגיקה |
| Change\_size | Evt – האירוע שהפעיל את הפעולה | משנה את גודל העץ |
| Add\_pic | Item – אובייקט בעץ  Name – השם הבסיסי של האובייקט  Is\_folder – האם האובייקט הוא תיקייה | משיגה לאובייקט שם של תמונה להצמיד וקוראת לput\_pic\_for\_item() |
| Put\_pic\_for\_item | Pic\_path – המיקום של התמונה  Item – האובייקט להצמיד לו את התמונה | מצמידה לאובייקט את התמונה ומוסיפה לitem\_image\_path |
| On\_expanded | Evt – האירוע שהפעיל את הפעולה | מראה למשתמש את מה שיש בתיקייה ומשנה את תמונת התיקייה באמצעות put\_pic\_for\_item |
| On\_collapsed | Evt – האירוע שהפעיל את הפעולה | מחביא את תוכן התיקייה ומשנה את תמונת התיקייה באמצעות put\_pic\_for\_item |
| Copy\_paste | Evt – האירוע שהפעיל את הפעולה | בודקת האם המשתמש העתיק או הדביק (באמצעות ctrl + C או ctrl + V) אם הדביק מודיעה ללוגיקה אם העתיק שומרת את ההעתקה |
| On\_drag | Evt – האירוע שהפעיל את הפעולה | משנה את העכבר לתמונה של הקובץ ושומרת את את האובייקט שהמשתמש מעביר |
| On\_drop | Evt – האירוע שהפעיל את הפעולה | מודיעה ללוגיקה מאיפה לאיפה המשתמש העביר קובץ |
| Valid\_input | File\_name – שם הקובץ  Is\_folder – אם האובייקט תיקייה | מחזירה אם המשתמש הכניס קלט תקין או לא לשם של האובייקט |
| Create\_file\_dialog | Command – איזה פעולה מחכה לקלט מהמשתמש  Path – המיקום של האובייקט עליו לחץ המשתמש | בונה מקום בו המשתמש יכול להכניס את הקלט באמצעות CreateFileDialog ושולחת ללוגיקה את מה שהמשתמש הכניס |
| Rename\_object | Old\_path – המיקום הקודם של האובייקט  New\_name – השם החדש של האובייקט | משנה את שם האובייקט בכל הרשימות והמילונים ובעץ |
| Delete\_object | Path – מיקום הקובץ | מוחקת את האובייקט ומוציאה אותו מpath\_item |
| Add\_object | Path – מיקום האובייקט  Name – שם האובייקט  Typ – סוג האובייקט | מוסיפה לרשימות והמילונים המתאימים את האובייקט ומוסיפה לעץ. קוראת לadd\_pic |
| On\_right\_click | Evt – האירוע שהפעיל את הפעולה | מראה למשתמש את אפשרויות הבחירה ולאחר בחירתו קוראת לcommand\_selected() |
| Command\_selected | Evt – האירוע שהפעיל את הפעולה  Item – האיבר שנחברה עליו הפקודה | מעבירה ללוגיקה באמצעות send\_to\_logic את הערכים המתאימים לפי הפקודה שנחברה ע"י המשתמש |
| Convert\_to\_tree | Dic – המילון המייצג עץ  Father – האבא הנוכחי של האיברים בעץ | בצורה רקורסיבית עובר על המילון ומכניס לעץ את העץ שמייצג המילון. קורא לadd\_pic() |
| On\_double\_clicked | Evt – האירוע שהפעיל את הפעולה | אם האובייקט שנלחץ עליו הוא תיקייה מחביא או מראה את תוכן התיקייה, אם האובייקט הוא קובץ פותח אותו |

1. תיאור סביבת הפיתוח –

* שפת התכנות שנבחרה
* לכתיבת הפרויקט – פייתון,
* Wxpython - למימוש הגרפיקה
* - sql בשביל עבודה מול בסיס נתונים
* כלי פיתוח וכלי בדיקה – pycharm: פיתוח בסביבת עבודה שתומכת בפייתון, dbbrowser: בדיקת בסיס הנתונים; לראות שהשינויים שקורים בבסיס הנתונים רצויים, wireshark: בדיקה של התקשורת ברשת לדוגמה לבדוק שהתקשורת מוצפנת, fileexplorer: לבדוק שהקבצים נשמרים כראוי

1. תיאור האלגוריתמים המרכזיים בפרויקט:

**הבעיה: איך לבנות את רשימת הקבצים הקיימים במערכת כדי להציגה למשתמש שהתחבר.**

| מס' | חלופה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | עם התחברות משתמש למערכת השרת פונה לכלל המחשבים ומבקש את רשימת הקבצים שנמצאים אצלו | יעילות – רק בעת הדרישה בונים את הרשימה | זמן תגובה יכול להתארך - מרגע שהמשתמש התחבר עד שהוא רואה את הרשימה |
| 2 | כל מחשב שמתחבר למערכת (בלי קשר להימצאות משתמש) מעביר לשרת את רשימת הקבצים שנמצאים אצלו | זמן תגובה קצר – בשרת תמיד רשימת קבצים עדכנית מה שמקצר את זמן התגובה למשתמש | יעילות – כל הזמן מתבצע תהליך ברקע של העברת רשימה גם אם אין משתמש  עומס על הרשת – העברת רשימת קבצים גם ללא בקשה אפילו אם אין משתמש מעמיסה את הרשת.  בנוסף השרת אוסף את המידע של כלל המשתמשים ללא צורך ומעמיס על התקשורת |

חלופה 1 היא החלופה הרצויה – אומנם היא תיקח בטעינת המערכת אך לא תפגע בביצועי השרת.

**הבעיה: איך לשלוח למשתמש את מבנה רשימת הקבצים בתקשורת?**

|  |  |  |  |
| --- | --- | --- | --- |
| מס' | חלופה | יתרונות | חסרונות |
| 1 | כאשר המשתמש מתחבר השרת ישלח לו כל פעם בחלקים את הקבצים, כל פעם שהמשתמש ייכנס לתיקייה השרת ישלח לו את תוכן התיקייה | מונע סרבול – הפרוטוקול תקשורת הופך להרבה יותר קצר ופשוט | זמן תגובה ארוך – כל פעם שילחץ על תיקייה ייקח זמן עד שיופיע לו את תוכן התיקייה.  עומס על הרשת – נוצר יותר תקשורת ברשת |
| 2 | כאשר המשתמש מתחבר השרת ישלח לו את כל מערכת הקבצים בשליחה אחת | זמן תגובה קצר – השרת ישלח למשתמש בשליחה אחת ולא יצטרך לשלוח כל פעם שהמשתמש מבקש תוכן תיקייה. | פרוטוקול מסובך – כדי שנוכל לשלוח מערכת שלמה של קבצים במחרוזת אחת נצטרך לבנות פרוטוקול מסובך. |

חלופה 2 היא החלופה הרצויה – בפרויקט של יש עדיפות לנוחות המשתמש בדגש על רשימת הקבצים שהיא מהות המערכת.

**הבעיה: איזו הצפנה להשתמש?**

| מס' | חלופה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | הצפנה סימטרית  הצפנה בה משתמשים באותו המפתח להצפנה ולפענוח | פשוט מהיר ויעיל | פרוצה לחלוטין, לכל אדם יש את המפתח |
| 2 | הצפנה א-סימרית  לפרט | מאובטח | ההצפנה עצמה לוקחת הרבה זמן מכיוון שהמפתחות יותר ארוכים |
| 3 | הצפנה היברידית לפרט | מאובטח ויחסית מהיר | הצפנת המפתחות ארוכה |

חלופה 3 היא החלופה הרצויה – נרצה בפרויקט זה תקשורת בטוחה אך מהירה.

**הבעיה: איך מעבירים את הקבצים ברשת? מהמחשבים בהם המשתמש עבד בעבר למחשב הנוכחי שלו**

|  |  |  |  |
| --- | --- | --- | --- |
| מס' | חלופה | יתרונות | חסרונות |
| 1 | העברה דרך שרת | נוחיות – נשתמש בסוקט אחד במהלך כל התקשורת ברשת | איטיות ועומס על השרת – בסוקט יועבר הרבה מאוד מידע לכן העברת דברים ברשת תהיה תהליך איטי |
| 2 | תקשורת pear to pear באמצעות פורט ספציפי | מהירות – כל פעם שנעביר את המידע הוא לא יעבור דרך השרת אלא ישר בpear to pear | יכול להיווצר עומס על לקוח כאשר מבוקשים ממנו קבצים גדולים |

חלופה 2 היא החלופה הרצויה – בפרויקט זה אנו מתייחסים לעבודה במעבדה של תלמידים לכן לא יבוקשו קבצים גדולים (כמו סרטים) אלא קבצים קטנים (כמו קבצי py, java) לכן חלופה מספר 2 הרבה יותר מהירה.

**הבעיה: איך מציגים למשתמש את מראה הקבצים?**

|  |  |  |  |
| --- | --- | --- | --- |
| מס' | חלופה | יתרונות | חסרונות |
| 1 | Windows explorer  (כדי שהקבצים לא יפתחו בצורה ישירה ניצור קבצי דמה, קבצי bat שיריצו את התוכנית הרצויה שינסו לפתוח אותם) | אין צורך בהכנת GUI חדש, אלא השתמשות בwindows explorer שכבר מוכן. | קושי לעבוד עם המערכת של הקבצים בדרך הרצויה.  נראות גראפית לא נוחה למשתמש מכיוון שהicons של הקצים יהיו לא של הקבצים האמיתיים |
| 2 | GUI שאני אצור | קל מאוד לעבוד עם המערכת של הקבצים בדרך הרצויה | יש צורך להכין GUI חדש |

חלופה 2 היא החלופה הרצויה מכיוון שלעבוד עם הwindows explorer יהיה קשה מאוד ובנוסף נראות המערכת היא חשובה מאוד בפרויקט שלנו מכיוון שהוא מיועד לנוחות המשתמש.

1. תיאור מסכי הפרויקט:

**מסך ראשוני** – בחירה אם התחברות או הרשמה.

**מסכי הרשמה\התחברות** - הכנסת שם משתמש וסיסמה אם השם משתמש או סיסמה לא תקינים תוצג הודעה מותאמת.

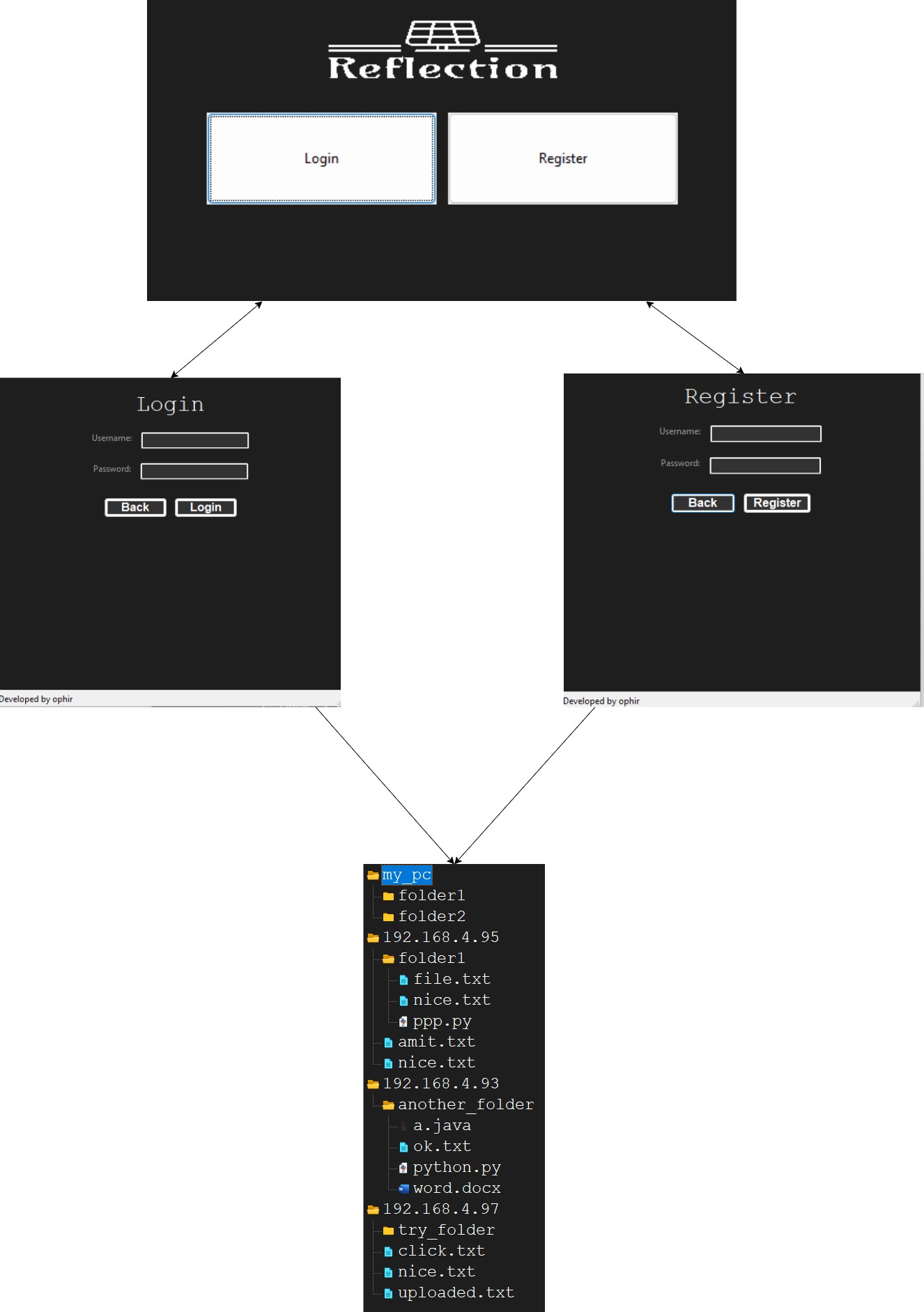
**מסך בית** – לאחר הרשמה או התחברות יפתח מסך בית

**מראה קבצים** - יהיה באפשרות מסך הבית להיכנס למראה הקבצים

תכנון:



בפועל:



**מסכים נוספים:**

1. תיאור פרוטוקול התקשורת

**כל ההודעות בתקשורת מוצפנות באמצעות מפתח סימטרי שמועבר בתחילת התקשורת (אפשר לראות בפרוטוקול החלפת מפתחות)**

פרוטוקול לקוח משתמש לשרת

אורך כל הודעה ישלח לפני כל ההודעה, אורכו 6 בתים, לאחר מכן ישלח Opcode שאורכו 2 בתים ראשונים לאחר מכן כל משתנה יופרד ע"י פסיק

| הודעות | מ- | אל | שדות-גודל | opcode | דוגמה |
| --- | --- | --- | --- | --- | --- |
| הרשמה | לקוח | שרת | שם משתמש,סיסמה | 01 | 01ophir,12345 |
| סטטוס הרשמה | שרת | לקוח | סטטוס | 02 | 02ok  02no |
| התחברות | לקוח | שרת | שם משתמש\_סיסמה  mac | 03 | 03ophir,12345, AC:BD:9D:F8:35:B2 |
| סטטוס התחברות | שרת | לקוח | סטטוס | 04 | 04ok  04no |
| רשימת קבצים | שרת | לקוח | יפורט בהמשך | 05 | יפורט בהמשך |
| יצירת קובץ\תיקייה | לקוח | שרת | מיקום, סוג | 06 | 062.2.2.2/new\_file,txt  062.2.2.2/new\_folder,fld |
| סטטוס יצירה | שרת | לקוח | מצב,מיקום,סוג | 07 | 07ok,1.1.1.1/file,txt |
| שינוי שם | לקוח | שרת | מיקום, שם חדש | 08 | 081.1.1.1/folder/file.txt,text.txt |
| סטטוס שינוי שם | שרת | לקוח | מצב,מיקום,שם | 09 | 09ok,1.1.1.1/file,new\_file |
| מחיקה | לקוח | שרת | מיקום | 10 | 102.2.2.2/file.txt |
| סטטוס מחיקה | שרת | לקוח | מצב,מיקום | 11 | 11ok,1.1.1.1/file |
| העברה | לקוח | שרת | מיקום,מיקום | 12 | 124.4.4.4/file.txt,4.4.4.4/folder |
| סטטוס העברה | שרת | לקוח | מצב,מיקום,מיקום | 13 | 13ok,1.1.1.1/fold/file,1.1.1.1/fold1 |
| שכפול | לקוח | שרת | מיקום,מיקום | 14 | 142.2.2.2/file.txt,4.4.4.4/folder |
| סטטוס שכפול | שרת | לקוח | מצב,מיקום,מיקום | 15 | 15ok,1.1.1.1/fold/file,1.1.1.1/fold |
| הורדת תיקיית מחשב | שרת | לקוח | שם תיקייה | 30 | 302.2.2.2 |
| אם משתמש או כללי | לקוח | שרת | כללי או משתמש - 1 | 35 | 35U |

פרוטוקול לקוח משתמש ללקוח בעל קובץ

אורך כל הודעה ישלח לפני כל ההודעה, אורכו 8 בתים, לאחר מכן ישלח Opcode שאורכו 2 בתים ראשונים לאחר מכן כל משתנה יופרד ע"י פסיק

| הודעות | מ- | אל | שדות-גודל | opcode | דוגמה |
| --- | --- | --- | --- | --- | --- |
| פתיחת קובץ | משתמש | בעל קובץ | path | 16 | 161.1.1.1/file.txt |
| תשובה לפתיחת קובץ | בעל קובץ | משתמש | אישור-2, מיקום קובץ, אורך מידע, מידע  (אם יש) | 17 | 17ok,1.1.1.1/file.txt ,len,data  17no |
| שינוי תוכן קובץ | משתמש | בעל קובץ | מיקום קובץ,אורך קובץ, מידע | 18 | 184.4.4.4/file.txt,len,data |
| סטטוס שינוי תוכן קובץ | בעל קובץ | משתמש | סטטוס | 19 | 19ok |

פרוטוקול לקוח שאינו בהכרח משתמש לשרת

אורך כל הודעה ישלח לפני כל ההודעה, אורכו 6 בתים, לאחר מכן ישלח Opcode שאורכו 2 בתים ראשונים לאחר מכן כל משתנה יופרד ע"י פסיק

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| הודעות | מ- | אל | שדות-גודל | opcode | דוגמה |
| בקשת רשימת קבצים | שרת | לקוח | שם התיקייה שהשרת מבקש | 31 | 31ophir |
| רשימת קבצים | לקוח | שרת | יפורט בהמשך | 20 | יפורט בהמשך |
| שינוי שם | שרת | לקוח | מיקום, שם חדש | 21 | 211.1.1.1/folder/file.txt,text.txt |
| סטטוס שינוי שם | לקוח | שרת | סטטוס,מיקום,שם חדש | 22 | 22ok,1.1.1.1/file,1.1.1.1/file2 |
| מחיקה | שרת | לקוח | מיקום | 23 | 232.2.2.2/file.txt |
| סטטוס מחיקה | לקוח | שרת | סטטוס,מיקום | 24 | 24ok,1.1.1.1/file |
| העברה | שרת | לקוח | מיקום,מיקום | 25 | 254.4.4.4/file.txt,4.4.4.4/folder |
| סטטוס העברה | לקוח | שרת | סטטוס,קובץ,מיקום | 26 | 26ok,1.1.1.1/file,1.1.1.1/folder |
| שכפול | שרת | לקוח | מיקום,מיקום | 27 | 272.2.2.2/file.txt,4.4.4.4/folder |
| סטטוס שכפול | לקוח | שרת | סטטוס,מיקום,מיקום | 28 | 28no,1.1.1.1/file,1.1.1.1/folder |
| שליחת mac | לקוח | שרת | Mac | 29 | 29AC:BD:9D:F8:35:B2 |
| סטטוס שליחת mac | שרת | לקוח | סטטוס | 34 | 34no |
| יצירה | שרת | לקוח | מיקום,סוג | 32 | 321.1.1.1/folder,fld |
| סטטוס יצירה | לקוח | שרת | סטטוס, מיקום, סוג | 33 | 33no,1.1.1.1/folder,fld |
| אם משתמש או כללי | לקוח | שרת | כללי או משתמש - 1 | 35 | 35G |

פרוטוקול לקוח כללי ללקוח כללי

אורך כל הודעה ישלח לפני כל ההודעה, אורכו 8 בתים, לאחר מכן ישלח Opcode שאורכו 2 בתים ראשונים לאחר מכן כל משתנה יופרד ע"י פסיק

| הודעות | מ- | אל | שדות-גודל | opcode | דוגמה |
| --- | --- | --- | --- | --- | --- |
| העתקת קובץ | שולח קובץ | שומר קובץ | New\_path,  Old\_path | 36 | 361.1.1.1/file.txt,  2.2.2.2/file.txt |
| תשובה להעתקת קובץ | שומר קובץ | שולח קובץ | Status,new\_path,old\_path | 37 | 37ok1.1.1.1/file.txt,  2.2.2.2/file.txt |
| העברת קובץ | שולח קובץ | שומר קובץ | New\_path,  Old\_path | 38 | 381.1.1.1/file.txt,  2.2.2.2/file.txt |
| תשובה להעברת קובץ | שומר קובץ | שולח קובץ | Status,new\_path,old\_path | 39 | 39ok1.1.1.1/file.txt,  2.2.2.2/file.txt |

פרוטוקול החלפת מפתחות

אורך כל הודעה ישלח לפני כל ההודעה, אורכו 3 בתים, לאחר מכן ישלח Opcode שאורכו 2 בתים ראשונים לאחר מכן כל משתנה יופרד ע"י פסיק

| הודעות | מ- | אל | שדות-גודל | opcode | דוגמה |
| --- | --- | --- | --- | --- | --- |
| מפתח א-סימטרי  (לא מוצפן) | שרת | לקוח | מפתח א-סימטרי פומבי | 98 | 98key |
| מפתח סימטרי  (מוצפן באמצעות המפתח הציבורי) | לקוח | שרת | מפתח סימטרי | 99 | 99key |

הסבר על העברת מערכת קבצים:

**Opcode: 05**

הסבר: יישלח מחרוזת המתארת את מערכת הקבצים בצורה הזו:

Full path של התיקייה לאחר מכן סימן שאלה לאחר מכן בסוגריים מרובעות בצורה של רשימה כל התיקיות שנמצאות בתוך הfull path לאחר מכן סימן שאלה ולבסוף בסוגריים מרובעות בצורה של רשימה כל הקבצים שנמצאים בתוך full path. כל זה יבוצע בצורה רקורסיבית עבור כל התיקיות במערכת הקבצים

מ – שרת

אל – לקוח

שדות: full\_path?[folders]?[files]

דוגמה:

protocol\_example?['dir1', 'dir2', 'dir3']?[] protocol\_example\dir1?['folder\_in\_1']?['another\_text.txt', 'text.txt']

protocol\_example\dir1\folder\_in\_1?[]?[]

protocol\_example\dir2?['hello\_in\_2']?[]

protocol\_example\dir2\hello\_in\_2?[]?['hello.txt', 'ok.txt']

protocol\_example\dir3?[]?['archive.rar', 'image.bmp'],,1.1.1.1

**Opcode: 20**

הסבר: יישלח מחרוזת המתארת את מערכת הקבצים שנמצאת אצל הלקוח בצורה הזו:

Full path של התיקייה לאחר מכן סימן שאלה לאחר מכן בסוגריים מרובעות בצורה של רשימה כל התיקיות שנמצאות בתוך הfull path לאחר מכן סימן שאלה ולבסוף בסוגריים מרובעות בצורה של רשימה כל הקבצים שנמצאים בתוך full path. כל זה יבוצע בצורה רקורסיבית עבור כל התיקיות במערכת הקבצים

מ – לקוח

אל – שרת

שדות: full\_path?[folders]?[files]

דוגמה:

protocol\_example?['dir1', 'dir2', 'dir3']?[] protocol\_example\dir1?['folder\_in\_1']?['another\_text.txt', 'text.txt']

protocol\_example\dir1\folder\_in\_1?[]?[]

protocol\_example\dir2?['hello\_in\_2']?[]

protocol\_example\dir2\hello\_in\_2?[]?['hello.txt', 'ok.txt']

protocol\_example\dir3?[]?['archive.rar', 'image.bmp'],,1.1.1.1

1. תיאור מבני הנתונים

נשתמש בפרויקט בבסיס נתונים שיקרא reflection\_db

נמצא בבסיס נתונים זה 2 טבלאות:

* User\_pass – שומרת סיסמאות לפי משתמש סיסמאות נשמרות כhash (user = primary key)
* User\_mac – שומרת mac של מחשבים שהמשתמש עבד עליהם לפי משתמש (user+mac = primary key)

User\_pass

|  |  |
| --- | --- |
| Password – varchar(30) | User – varchar(30)(PK) |
| 1234 | Imriking123 |
| 555555 | ophir |
| Abc\_123 | yotam |

User\_mac

|  |  |
| --- | --- |
| Mac – varchar(17)(PK) | User – varchar(30)(PK) |
| 56:ab:56:ac:58:b1 | Imriking123 |
| 57:ab:56:ac:58:b1 | ophir |
| 58:ab:56:ac:58:b1 | yotam |
| 59:ab:56:ac:58:b1 | ophir |

בנוסף תהיה מחיצה עם השם של הuser בכל מחשב שבה ישמרו כל הקבצים שהמשמש ירצה לשמור לפעמים הבאות שהוא מגיע למעבדה.

1. סקירת חולשות והאיומים

שכבת האפליקציה:

* sql injection – מתקפה זו קורת בפרויקט שבו יש שימוש בסיס הנתונים SQL, כאשר מקבלים קלט מסוים המקושר לבסיס הנתונים לדוגמה שם משתמש וסיסמה יש באפשרות המשתמש להכניס קלט מסוים שישבש את בסיס הנתונים. נטפל בהתקפה זו בכך שנכתוב שאילתות נכונות במחלקה db שלא מאפשרות הכנסה של קוד זדוני, בנוסף גם כאשר נשמור את הסיסמאות בסיס הנתונים נשמור אותם לאחר שגיבבנו אותם ובכך אם לאדם לא רצוי יש גישה לבסיס הנתונים הוא לא יוכל לדעת את סיסמאות המשתמשים.
* תהליך ה login אימות ווידוא – כדי שאנשים שאין להם גישה לקבצים מסוימים לא יקבלו את הגישה, נבצע תהליך login לכל משתמש שכאשר כל משתמש יהיה בעל שם משתמש שונה וסיסמה שרק הוא יודע.
* MITM : התקפת MITM (man in the middle) היא התקפה בה אדם שמאזין בתקשורת רואה מה מועבר בה. נטפל בהתקפה זו בכך שנעבוד עם הצפנה היברידית שתחסום מהאדם הזדוני להבין מה מועבר בתקשורת
* DOS/DDOS : התקפת dos (denial of service) היא התקפה שלקוח משתמש בקצב מאוד מהיר במשאבי השרת ובכך מתיש את השרת ומפריע לו לנהל את כל שאר הלקוחות שלו. נמנע התקפה זו בכך שנראה שיש פעילות לא נורמלית של לקוח אחד נחסום את הip שלו ולא ניתן לו להתחבר לשרת יותר. התקפת ddos (distributed denial of service) היא התקפה בה מבוצע dos מהרבה מחשבים אך התקפה זו מטופלת כאשר טיפלנו בdos מכיוון שהפרויקט שלי נמצא במעבדה קטנה ואין הרבה מחשבים שיכולים להכביד על השרת.

שכבת התעבורה:

* נשתמש בפרוטוקל tcp ובהצפנה היברידית

הפעלת המערכת:

* הפרויקט שלי מושתת על כך שכאשר המחשב נדלק יופעל הקוד של הgeneralClient לכן אדם (בזדון או לא בזדון) יכול למחוק את הקובץ הספציפי הזה מהמחשב. נטפל בבעיה זו בכך שהמחשבים במעבדה יהיו נעולים ומאותחלים. הכוונה היא שכאשר מפעילים את המחשב תמיד הוא יחזור למצבו הקבוע שקבענו, והקבצים שלנו יהיו נעולים ולא יהיה אפשר לשנות אותם.

**פרק ה' - 'שם המערכת' - 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):

**הבעיה: איך לבנות את רשימת הקבצים הקיימים במערכת כדי להציגה למשתמש שהתחבר.**

| מס' | חלופה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | עם התחברות משתמש למערכת השרת פונה לכלל המחשבים ומבקש את רשימת הקבצים שנמצאים אצלו | יעילות – רק בעת הדרישה בונים את הרשימה | זמן תגובה יכול להתארך - מרגע שהמשתמש התחבר עד שהוא רואה את הרשימה |
| 2 | כל מחשב שמתחבר למערכת (בלי קשר להימצאות משתמש) מעביר לשרת את רשימת הקבצים שנמצאים אצלו | זמן תגובה קצר – בשרת תמיד רשימת קבצים עדכנית מה שמקצר את זמן התגובה למשתמש | יעילות – כל הזמן מתבצע תהליך ברקע של העברת רשימה גם אם אין משתמש  עומס על הרשת – העברת רשימת קבצים גם ללא בקשה אפילו אם אין משתמש מעמיסה את הרשת.  בנוסף השרת אוסף את המידע של כלל המשתמשים ללא צורך ומעמיס על התקשורת |

חלופה 1 היא החלופה הרצויה – אומנם היא תיקח בטעינת המערכת אך לא תפגע בביצועי השרת.

**בקטע קוד זה השרת שולח לכל המחשבים שהמשתמש התחבר אליהם בקשה שישלחו לו את רשימת הקבצים במחשב שלהם.**

**def** handle\_sign\_in(user\_ip: str, vars: list):  
 *"""  
 signs in the user and returns response accordingly* **:param** *user\_ip: user's ip* **:param** *vars: username, password, mac address* **:return***: None  
 """* print(**'got to sign in'**)  
 **if** len(vars) != 3:  
 handle\_disconnect(user\_ip, **False**)  
 **return** user, password, user\_mac = vars  
 db\_pass = db.get\_password(user)  
 hashed\_pass = symmetrical\_encryption.SymmetricalEncryption.hash(password)  
 **if** db\_pass == hashed\_pass:  
  
 db.add\_user\_mac(user, user\_mac)  
 server\_comm.send(user\_ip, protocol.pack\_status\_login(**True**))  
 macs\_worked\_on = db.get\_macs(user)  
 print(**'macs\_worked\_on:'**, macs\_worked\_on)  
 user\_comps[user\_ip] = [(user\_ip[0], **'G'**)]  
 username\_ip[user] = user\_ip  
  
 *# adding to user comps* **for** mac **in** macs\_worked\_on:  
 **for** ip **in** ip\_mac:  
 **if** ip\_mac[ip] == mac:  
 user\_comps[user\_ip].append(ip)  
 server\_comm.send(ip, protocol.pack\_ask\_file\_Tree(**f'{**user**}'**))  
  
 print(**'user\_comps:'**, user\_comps)  
 *# asking file tree from own user mac* **if** user\_mac **not in** macs\_worked\_on:  
 server\_comm.send((user\_ip[0], **'G'**), protocol.pack\_ask\_file\_Tree(**f'{**user**}'**))  
  
 **else**:  
 server\_comm.send(user\_ip, protocol.pack\_status\_login(**False**))

**הבעיה: איך לשלוח למשתמש את מבנה רשימת הקבצים בתקשורת?**

|  |  |  |  |
| --- | --- | --- | --- |
| מס' | חלופה | יתרונות | חסרונות |
| 1 | כאשר המשתמש מתחבר השרת ישלח לו כל פעם בחלקים את הקבצים, כל פעם שהמשתמש ייכנס לתיקייה השרת ישלח לו את תוכן התיקייה | מונע סרבול – הפרוטוקול תקשורת הופך להרבה יותר קצר ופשוט | זמן תגובה ארוך – כל פעם שילחץ על תיקייה ייקח זמן עד שיופיע לו את תוכן התיקייה.  עומס על הרשת – נוצר יותר תקשורת ברשת |
| 2 | כאשר המשתמש מתחבר השרת ישלח לו את כל מערכת הקבצים בשליחה אחת | זמן תגובה קצר – השרת ישלח למשתמש בשליחה אחת ולא יצטרך לשלוח כל פעם שהמשתמש מבקש תוכן תיקייה. | פרוטוקול מסובך – כדי שנוכל לשלוח מערכת שלמה של קבצים במחרוזת אחת נצטרך לבנות פרוטוקול מסובך. |

חלופה 2 היא החלופה הרצויה – בפרויקט של יש עדיפות לנוחות המשתמש בדגש על רשימת הקבצים שהיא מהות המערכת.

**בקטע קוד זה נבנית המחרוזת לפי המיקום של התיקייה.**

**def** get\_path\_tree(path: str):  
 *"""  
 returns path to dir returns tree of dir* **:param** *path: path to dir* **:return***: file tree of path  
 """* file\_tree = **''  
 for** part **in** os.walk(path):  
 part = **f'{**part[0]**}?{**part[1]**}?{**part[2]**}\n'** file\_tree += part  
  
 **return** file\_tree

**בקטע קוד זה השרת מעביר למילון את המחרוזת של עץ הקבצים, לאחר מכן מוסיף ip של הלקוח למקומות הנדרשים, מחזיר למחרוזת שמדמה עץ ושולח ללקוח משתמש.**

**def** handle\_got\_file\_tree(got\_ip, vars: list):  
 *"""* **:param** *got\_ip: the ip got file tree from* **:param** *vars: file tree got from got\_ip* **:return***: None  
 """* file\_tree = vars[0]  
 got\_ok = **True** ip\_to\_send = **None** folders\_with\_ip = {}  
 **if** len(vars) == 1:  
 user\_to\_send = FileHandler.get\_user(list(file\_tree.keys())[0])  
 print(**'user\_to\_send:'**, user\_to\_send)  
 **if** user\_to\_send **in** username\_ip:  
 ip\_to\_send = username\_ip[user\_to\_send]  
 print(**'to\_send:'**, ip\_to\_send)  
  
 **else**:  
 got\_ok = **False** ip\_to\_send = **None  
  
 if** got\_ok **and** ip\_to\_send **in** user\_comps **and** got\_ip **in** user\_comps[ip\_to\_send]:  
  
 folders\_with\_ip = {}  
  
 *# inserting ip to each path in dictionary* **for** folder\_path **in** file\_tree.keys():  
 user\_plus\_ip = os.path.join(user\_to\_send, got\_ip[0])  
 **if** user\_to\_send **in** folder\_path:  
 new\_folder\_path = folder\_path.replace(user\_to\_send, user\_plus\_ip, 1)  
 folders\_with\_ip[new\_folder\_path] = file\_tree[folder\_path]  
 **else**:  
 got\_ok = **False  
 break  
  
 else**:  
 got\_ok = **False  
 else**:  
 got\_ok = **False  
  
 if** got\_ok **and** ip\_to\_send:  
 *# sending file tree to user* server\_comm.send(ip\_to\_send, protocol.pack\_send\_file\_tree(folders\_with\_ip))  
  
 **else**:  
 print(**'error in getting file tree'**)  
 handle\_disconnect(got\_ip, **False**)

**הבעיה: איזו הצפנה להשתמש?**

| מס' | חלופה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | הצפנה סימטרית  הצפנה בה משתמשים באותו המפתח להצפנה ולפענוח | פשוט מהיר ויעיל | פרוצה לחלוטין, לכל אדם יש את המפתח |
| 2 | הצפנה א-סימרית  לפרט | מאובטח | ההצפנה עצמה לוקחת הרבה זמן מכיוון שהמפתחות יותר ארוכים |
| 3 | הצפנה היברידית לפרט | מאובטח ויחסית מהיר | הצפנת המפתחות ארוכה |

חלופה 3 היא החלופה הרצויה – נרצה בפרויקט זה תקשורת בטוחה אך מהירה.

**החלפת מפתחות צד שרת:**

**def** \_key\_exchange(self, client, ip):  
 *"""  
 exchanging keys with client, puts client in open clients at the end* **:param** *client: client socket* **:param** *ip: ip address* **:return***:  
 """* self.send(client, server\_protocol.pack\_public\_key(self.a\_encrypt.get\_public\_key()), **False**)  
 **try**:  
 length = int(client.recv(self.send\_len).decode())  
 data = client.recv(length)  
 data = self.a\_encrypt.decrypt\_msg(data)  
 **except** Exception **as** e:  
 print(**"error in key\_exchange: "** + str(e))  
 self.disconnect\_client(client)  
 **else**:  
  
 encryption = symmetrical\_encryption.SymmetricalEncryption(data[2:])  
  
 self.\_get\_client\_type(client, ip, encryption)

**החלפת מפתחות צד לקוח:**

**def** \_key\_exchange(self):  
 *"""  
 gets server's public key and sends server symmetric key* **:return***: None  
 """* **try**:  
 length = int(self.server.recv(self.send\_len).decode())  
 data = self.server.recv(length)  
  
 **except** Exception **as** e:  
 self.close()  
 sys.exit(**"server is down, try again later"**)  
  
 public\_key = data[2:]  
 print(**'pulic key:'**, public\_key)  
 self.symmetric = symmetrical\_encryption.SymmetricalEncryption()  
 key\_to\_send = general\_client\_protocol.pack\_key(self.symmetric.key)  
 key\_to\_send = AsymmetricEncryption.encrypt\_msg(key\_to\_send, public\_key)  
 **try**:  
 self.server.send(str(len(key\_to\_send)).zfill(self.send\_len).encode())  
 self.server.send(key\_to\_send)  
 **except** Exception **as** e:  
 self.close()  
 sys.exit(**'could not send server key encrypted'**)  
  
 **else**:  
 self.\_send\_client\_type()

**הבעיה: איך מעבירים את הקבצים ברשת? מהמחשבים בהם המשתמש עבד בעבר למחשב הנוכחי שלו**

|  |  |  |  |
| --- | --- | --- | --- |
| מס' | חלופה | יתרונות | חסרונות |
| 1 | העברה דרך שרת | נוחיות – נשתמש בסוקט אחד במהלך כל התקשורת ברשת | איטיות ועומס על השרת – בסוקט יועבר הרבה מאוד מידע לכן העברת דברים ברשת תהיה תהליך איטי |
| 2 | תקשורת pear to pear באמצעות פורט ספציפי | מהירות – כל פעם שנעביר את המידע הוא לא יעבור דרך השרת אלא ישר בpear to pear | יכול להיווצר עומס על לקוח כאשר מבוקשים ממנו קבצים גדולים |

חלופה 2 היא החלופה הרצויה – בפרויקט זה אנו מתייחסים לעבודה במעבדה של תלמידים לכן לא יבוקשו קבצים גדולים (כמו סרטים) אלא קבצים קטנים (כמו קבצי py, java) לכן חלופה מספר 2 הרבה יותר מהירה.

**כדי לגשת לתוכן קובץ שנמצא במחשב אחר יוצר מופע של clientComm ומפעיל תהליכון שמחכה למידע שישלח מהמחשב שהתחבר אליו.**

*# if file is on another computer***else**:  
 ip\_to\_connect = self.file\_handler.extract\_ip(self.user\_name, path)  
 *# setting up pear to pear* **if** ip\_to\_connect **not in** self.ip\_comm:  
 rcv\_q = Queue()  
  
 comm = ClientComm(ip\_to\_connect, Settings.pear\_port, rcv\_q, 8, **'U'**)  
 self.ip\_comm[ip\_to\_connect] = comm  
 threading.Thread(target=self.rcv\_comm, args=(comm,), daemon=**True**).start()  
  
 **else**:  
 comm = self.ip\_comm[ip\_to\_connect]  
  
 comm.send(protocol.pack\_do\_open\_file(path))

**הבעיה: איך מציגים למשתמש את מראה הקבצים?**

|  |  |  |  |
| --- | --- | --- | --- |
| מס' | חלופה | יתרונות | חסרונות |
| 1 | Windows explorer  (כדי שהקבצים לא יפתחו בצורה ישירה ניצור קבצי דמה, קבצי bat שיריצו את התוכנית הרצויה שינסו לפתוח אותם) | אין צורך בהכנת GUI חדש, אלא השתמשות בwindows explorer שכבר מוכן. | קושי לעבוד עם המערכת של הקבצים בדרך הרצויה.  נראות גראפית לא נוחה למשתמש מכיוון שהicons של הקצים יהיו לא של הקבצים האמיתיים |
| 2 | GUI שאני אצור | קל מאוד לעבוד עם המערכת של הקבצים בדרך הרצויה | יש צורך להכין GUI חדש |

חלופה 2 היא החלופה הרצויה מכיוון שלעבוד עם הwindows explorer יהיה קשה מאוד ובנוסף נראות המערכת היא חשובה מאוד בפרויקט שלנו מכיוון שהוא מיועד לנוחות המשתמש.

**קטע קוד זה ממיר מילון לעץ הקבצים בגרפיקה בצורה רקורסיבית.**

**def** convert\_to\_tree(self, dic: dict, father=**None**):  
 *"""  
 gets dictionary and converts it to tree* **:param** *dic: dictionary* **:param** *father: a parent* **:return***: None  
 """* **if not** father:  
 print(**'dictionary working with:'**, dic)  
 father\_path = next(iter(dic.keys()))  
 father = self.root  
 **else**:  
 father\_path = self.tree.GetItemData(father)  
  
 folder = **True  
 if** father\_path **not in** dic:  
 **return** *# runs on every object in this folder adds it to tree and if is folder then* **for** element **in** dic[father\_path]:  
 **if** element == **','**:  
 folder = **False  
 continue** path = os.path.join(father\_path, element)  
 **if** element == self.my\_ip:  
 element = **'my\_pc'** new\_item = self.tree.AppendItem(father, element, data=path)  
  
 self.path\_item[path] = new\_item  
 self.add\_pic(new\_item, element, folder)  
 **if** folder:  
 self.folders.append(path)  
 self.convert\_to\_tree(dic, new\_item)

**פרק ו' - 'שם המערכת' – בדיקות ('קופסא לבנה')**

1. פירוט הבדיקות

יש לעשות את הבדיקות לאחר התחברות מ2 המחשבים לפחות פעם אחת חוץ מבדיקות 14 ו15.

| **מספר** | **שם הבדיקה (שם שיעיד על תוכן הבדיקה)** | **מה אמורה לבדוק** | **איך מתכננים לבדוק**  **(לתאר בפירוט את שלבי הבדיקה)** | **מתי** | **מה בוצע בפועל** |
| --- | --- | --- | --- | --- | --- |
| 1 | יצירת קבצים | לאחר התחברות למערכת, האם כאשר יוצרים במחשב א' קובץ, הקובץ יימצא במראה הקבצים במחשב ב'. | מתחברים למערכת במחשב א', יוצרים קובץ. לאחר מכן מתחברים למערכת במחשב ב' ובודקים האם הקובץ נמצא במראה הקבצים במחשב ב' | 16.5 | נוצר הקובץ במחשב א' והופיע במחשב ב' |
| 2 | מחיקת קבצים | לאחר התחברות למערכת, האם כאשר מוחקים במחשב א' קובץ, הקובץ לא יימצא במראה הקבצים במחשב ב'. | עושים זאת לאחר בדיקה מס' 1.  מתחברים למערכת במחשב א', מוחקים את הקובץ שיצרנו. לאחר מכן מתחברים למערכת במחשב ב' ובודקים האם הקובץ נעלם ממראה הקבצים. | 16.5 | נמחק הקובץ במחשב א' והשינוי הופיע במחשב ב' |
| 3 | שינוי שם קובץ | לאחר התחברות למערכת, האם כאשר משנים של קובץ במחשב א', שם הקובץ ישתנה במראה הקבצים במחשב ב'. | עושים זאת לאחר בדיקה מס' 1.  מתחברים למערכת במחשב א', משנים את שם הקובץ שיצרנו. לאחר מכן מתחברים למערכת במחשב ב' ובודקים האם שם הקובץ ישתנה במראה הקבצים. | 16.5 | שם הקובץ השתנה במחשב א' והשינוי הופיע במחשב ב' |
| 4 | שינוי תוכן קובץ | לאחר התחברות למערכת, האם כאשר משנים תוכן של קובץ במחשב א', תוכן הקובץ ישתנה במחשב ב'. | עושים זאת לאחר בדיקה מס' 1.  מתחברים למערכת במחשב א', כותבים בתוך הקובץ (txt) כמה מילים. לאחר מכן מתחברים למערכת במחשב ב' ובודקים האם מה שכתבנו בקובץ במחשב א' נמצא בקובץ במחשב ב'. | 16.5 | נפתח הקובץ במחשב א' שונה התוכן שלו. פתחנו את הקובץ במחשב ב' וראינו כי השתנה התוכן |
| 5 | סנכרון פעולות | נבדוק האם כל הפעולות שכתבנו לעיל עובדות כאשר עושים אותם באותו זמן ושום דבר אינו מתבלבל | ניצור 2 קבצים במחשב א' נתחבר למחשב ב' ונבדוק כי שניהם נוצרו. לאחר מכן שוב נתחבר למחשב א' ניצור עוד קובץ, נשנה את שם הקובץ של אחד משני הקבצים ואת תוכן הקובץ ונמחק את הקובץ הנוסף. נתחבר למחשב ב' ונבדוק שכל הפעולות שעשינו התבצעו כראוי | 16.5 | במחשב א' נוצרו 2 הקבצים שם הקובץ הראשון שונה והקובץ השני נמחק. במחשב ב' פתחנו את מראה הקבצים ונראה כי הכל השתנה כראוי |
| 6 | זמני תגובה | האם בעבודה עם קבצים גדולים יש פגיעה בזמני התגובה של המערכת | נשמור 5 קבצים גדולים במחשב ב' דרך מחשב א' ונבדוק האם במחשבים ב' בו מחובר משתמש אחר ברשת יש עיכוב בפעולות רגילות | 16.5 | הקבצים נשמרו במחשב ב' ממחשב א' ולמרות זאת לא היו בעיות בזמני התגובה במחשב ב' |
| 7 | עומס | האם כשיש משתמשים רבים במערכת, הביצועים עדיין טובים | נחבר 4 משתמשים למערכת מ8 מחשבים שונים ונבצע כמה פעולות שמתוארות לעיל ונבדוק האם הפעולות מתבצעות כראוי | 16.5 | חיברנו 4 משתמשים ל8 מחשבים שונים והפעולות התבצעו כראוי אם דיליי קטן מאוד מדי פעם |
| 8 | יציבות | האם המערכת נשארה יציבה אחרי התחברויות והתנתקויות של משתמשים | נחזור על פעולה מס' 7 ומדי פעם נחבר וננתק משתמשים לבדוק האם המערכת אינה קורסת | 16.5 | כאשר מחשבים התנתקו והתחברו המערכת לא קרסה |
| 9 | אבטחה | האם בעת העבודה על הקבצים ממחשב אחר המידע לא זולג החוצה | נפתח wireshark נפתח קובץ של מחשב ב' ממחשב א' ונבדוק האם התוכן שלו מוצג בwireshark | 16.5 | התוכן של הקובץ היה מוצפן כאשר מסתכלים דרך הwireshark |
| 10 | העלאת קובץ | האם כאשר מעלים קובץ אפשרי לפתוח אותו דרך המערכת | מתחברים למחשב מס 1 מעלים קובץ מן המחשב, מתחברים למחשב מס 2 ובודקים אם הקובץ קיים והתוכן שלו נכון | 16.5 | הקובץ עלה והתוכן שלו היה מתאים |
| 11 | הורדת קובץ | האם כאשר מורידים קובץ הוא מופיע במראה הקבצים של ווינדוס | מתחברים למחשב מס 1 מורידים קובץ ובודקים האם הוא נפתח במראה הקבצים של ווינדוס | 16.5 | הקובץ ירד וכאשר הוא נפתח מהwindows file explorer התוכן שלו היה מתאים |
| 12 | העברת קובץ | האם כאשר מעבירים קובץ ממחשב אחד למחשב אחר הקובץ מופיע במחשב האחר | מתחברים למחשב מס 1 מעבירים קובץ מהמחשב למחשב מס 2. בודקים האם הקובץ נמחק מהמחשב. מתחברים למחשב מס 2 ובודקים האם הקובץ נמצא | 16.5 | הקובץ הועבר ממחשב א' למחשב ב' דרך מחשב א'. נראה כי הקובץ נמחק ממחשב א'. התחברנו דרך מחשב ב' וראינו כי הקובץ נמצא. |
| 13 | שכפול קובץ | האם כאשר מעתיקים קובץ ממחשב אחד למחשב אחר הקובץ מופיע במחשב האחר | מתחברים למחשב מס 1 מעתיקים קובץ מהמחשב למחשב מס 2. מתחברים למחשב מס 2 ובודקים אם הקובץ נמצא | 16.5 | הקובץ הועתק ממחשב א' למחשב ב' דרך מחשב א'. התחברנו דרך מחשב ב' וראינו כי הקובץ נמצא |
| 14 | הרשמה למערכת | האם כאשר נרשמים למערכת עם שם משתמש שלא קיים מכניס למערכת הקבצים | נרשמים למערכת עם שם משתמש שלא קיים ובודקים אם נכנס למערכת קבצים. מריצים את המערכת מחדש ומנסים להירשם עם אותו משתמש ובודקים אם מוצגת הודעה ראויה שהמשתמש כבר קיים | 16.5 | נרשמנו למערכת עם שם משתמש שלא קיים ונכנסנו למערכת. יצאנו מהמערכת ניסינו להירשם עם אותו משתמש וקפצה לנו הודעת שגיאה כנדרש |
| 15 | התחברות למערכת | האם כאשר מתחברים למערכת עם שם משתמש קיים מכניס למערכת קבצים | מתחברים למערכת עם שם משתמש קיים ובודקים אם נכנס למערכת הקבצים. מריצים את המערכת מחדש ומנסים להיכנס עם שם משתמש שלא קיים ובודקים שמוצגת הודעה ראויה שהמשתמש לא קיים. | 16.5 | התחברנו למערכת עם שם משתמש קיים והופיע לנו מראה הקבצים. לאחר מכן התנתקנו מהמערכת וניסינו להתחבר עם שם משתמש לא קיים וקפץ הודעת שגיאה כנדרש. |

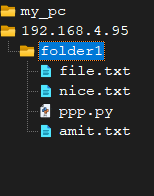
**פרק ז' - 'שם המערכת' – מדריך למשתמש**

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

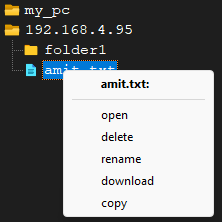
* על מנת להשתמש במערכת, ראשית יש להירשם. לאחר ההירשמות המשתמש יכול להשתמש במערכת.



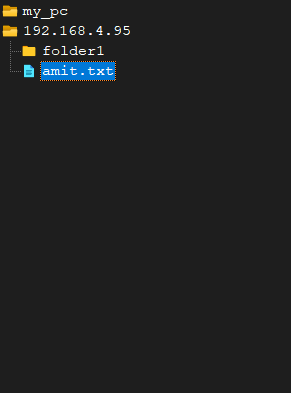
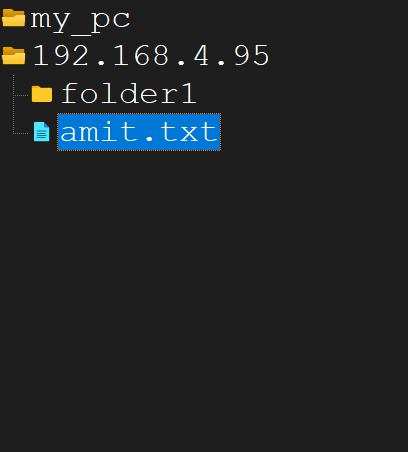
* לאחר הכניסה יוצג עץ הקבצים של המשתמש – כל הקבצים בכל המחשבים: my\_pc המחשב הנוכחי והאחרים מזוהים ע"י ip



* כדי לבצע פעולות בסיסיות על קובץ יש ללחוץ עליו במקש ימני בעכבר ולבחור את הפעולה הרצויה. אותו דבר גם לגבי תיקייה.



כדי להגדיל ולהקטין את המסך יש ללחוץ על CTRL ובאותו זמן לגלגל את הגלגלת.

* העברת קובץ נעשית ע"י גרירת הקובץ ע"י המקש השמאלי בעכבר (במצב לחוץ)

**פרק ח' - 'שם המערכת' – 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' - 'שם המערכת' – 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**