**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 א' ונבדוק האם במחשבים האחרים ברשת בהם מחובר המשתמש יש עיכוב בהגעת הקבצים |
| 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 |

**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 תיקייה**

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 |
|  |  |

**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]) |
| 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\_system | None | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| Pack\_send\_file\_system | None | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_create | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_rename | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_delete | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_move | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_clone | 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]) |
| 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\_open\_file | Location – מיקום קובץ (str) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_change\_file | Location – מיקום קובץ (str)  Data\_len – אורך מידע (str)  data – מידע (binary) | 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\_mac | Mac – הMAC של המחשב | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_rename | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_delete | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_move | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_clone | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_create | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_open\_file | Status – הצליח/לא הצליח (Boolean)  Data\_len – אורך מידע  Data - מידע | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_status\_change\_file | Status – הצליח/לא הצליח (Boolean) | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |
| pack\_key | Key – מפתח סימטרי | בונה הודעה ע"פ הפרוטוקול ומחזירה packed str |

| fileHandler | | |
| --- | --- | --- |
| מחלקה עם אוסף פעולות לטיפול במערכת הקבצים הפיזית | | |
| פעולה | טענת כניסה | טענת יציאה |
| rename | Location – מיקום (str)  new\_name – שם חדש (str) | משנה שם קובץ\תיקייה |
| delete | Location – מיקום (str) | מוחק קובץ\תיקייה |
| move | old\_location – מיקום ישן (str)  new\_location – מיקום חדש (str) | מעביר את הקובץ ממקום ישן למקום חדש |
| clone | old\_location – מיקום ישן (str)  new\_location – מיקום חדש (str) | משכפל את הקובץ למקום החדש |
| create | Location – מיקום ליצור (str)  type – סוג יצירה (str) | יוצר קובץ\תיקייה במיקום הנתון |
| open\_file | Location – מיקום קובץ (str) | פותחת קובץ |
| change\_content | Location – מיקום קובץ (str)  Data – תוכן קובץ חדש (binary) | משנה את תוכן הקובץ הנתון |
| get\_data | Location – מיקום קובץ (str) | מחזיר את תוכן הקובץ (binary) |

| 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 | |
| Macs\_reTbl | טבלת המחשבים user\_mac | |
| Cursor | 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\_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\_msg | data - הודעה (str) | מחזירה את ההודעה מוצפנת |
| decrypt\_msg | 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 |

| monitor | | |
| --- | --- | --- |
| המחלקה האחראית על ניטור הקבצים | | |
| שם משתנה | תפקיד | |
| path | התיקייה עליה יופעל הניטור | |
| Mon\_q | תור שיעביר ללוגיקה את הפעולות שקרו בתיקייה | |
| פעולה | טענת כניסה | טענת יציאה |
| \_\_init\_\_ | Mon\_q – התור  Path – התיקייה עליה יופעל הניתור | יוצרת את עצם הניתור וקוראת לmonitor\_ |
| \_monitor \*thread | none | מנטרת את התיקיה ומכניס לmon\_q את הפעולות שנעשו בתיקייה ומשתמש בפעולות של הפרוטוקל לפני ההכנסה לq |

| 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 | |
| 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 | 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) |
| \_key\_exchange | 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] |
| \_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 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 | |
| socket | הsocket בוא יעבור המידע בתקשורת | |
| symm\_encrypt | אובייקט של ההצפנה סימטרית | |
| asymm\_encrypt | האובייקט של ההצפנה הא-סימטרית | |
| Send\_len | כמה בתים יהיה אורך ההודעות המועברות בתקשורת | |
| פעולה | טענת כניסה | טענת יציאה |
| \_\_init\_\_ | Port – הפורט בו עובד השרת  rcv\_q – תור שבו יועברו ההודעת clientComm לlogic  Send\_len – כמה בתים יהיה אורך ההודעות המועברות בתקשורת | 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 |
| \_key\_exchange | none | מקבלת מהשרת את המפתח הציבורי שלו  יוצרת מפתח סימטרי  ושולחת אותו לשרת |
| \_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 ולאחר מכן שמה את הדברים הנדרשים בrcv\_q לפי פרוטוקול |

| 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 משתמשים המחוברים למערכת ובvalue את הmac שהמשתמשים כרגע נמצאים עליו | |
| Db | עצם של בסיס הנתונים | |
| User\_comps | מילון שבו ישמר כמפתח הip של משתמש מחובר ובמפתח 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 | מנהלת את הלוגיקה |
| Handle\_got\_mac | server – האובייקט של הcomm  Ip – הip ממנו נשלח הmac  mac – כתובת MAC | יוצרת עצם של db ושומרת בip\_mac את הip ואת הmac. לאחר מכן קורא לget\_users ומוסיפה לuser\_comps את הip במקומות הנדרשים, משתמשת גם בusername\_ip, לאחר מכן קוראת לask\_file\_tree עם הmac |
| Handle\_register | server – האובייקט של הcomm  Ip – הip אליו ישלח התשובה  Username – שם משתמש  Password – סיסמה | יוצרת עצם של db וקוראת לadd\_user מחזירה תשובה בהתאם לip, באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_sign\_in | server – האובייקט של הcomm  Ip – הip אליו ישלח התשובה  Username – שם משתמש  Password – סיסמה  Mac – כתובת | יוצרת עצם של db וקוראת לget\_password ומחזירה תשובה בהתאם ל .ipאם התשובה חיובית קוראת לadd\_user\_mac ולאחר מכן לget\_macs ונשלח לכל הmacs המיוחסים למשתמש באמצעות ask\_file\_tree ומוסיף 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 |
| ask\_file\_tree (thread) | server – האובייקט של הcomm  Mac – הmac שאליו נשלח את הבקשה | מבקש מהmac את מערכת הקבצים שלו ומחכה לתשובה לאחר קבלה של עץ הקבצים השרת שולח ללקוח משתמש את המידע על הקבצים שקיבל באמצעות send\_file\_tree |
| Send\_file\_tree | server – האובייקט של הcomm  Ip - הip אליו ישלח התשובה | שולחת באמצעות עטיפה בפרוטוקול ושליחה עם send לip את מערכת הקבצים |
| got\_rename | server – האובייקט של הcomm  Ip – הip ממנו נשלח המידע  Location – מיקום הקובץ\תיקייה  New\_name – שם חדש | שולחת לכתובת שמיקומה ניתן בLocation אלא אם הlocation הוא הip של המחשב שנשלח ממנו ההודעה, את הlocation לשנות ואת השם החדש באמצעות עטיפה בפרוטוקול ושליחה עם send לאחר מכן נשלח לip את סטטוס הפעולה |
| got\_delete | server – האובייקט של הcomm  Ip - הip אליו ישלח התשובה  Location – מיקום קובץ\תיקייה | שולחת לכתובת שמיקומה ניתן בLocation אלא אם הlocation הוא הip של המחשב שנשלח ממנו ההודעה את הlocation למחוק את הקובץ\תיקייה באמצעות עטיפה בפרוטוקול ושליחה עם send לאחר מכן נשלח לip את סטטוס הפעולה |
| got\_move | server – האובייקט של הcomm  Ip – הip ממנו נשלח המידע  Location – מיקום הקובץ\תיקייה  New\_location– מיקום חדש | שולחת לכתובת שמיקומה ניתן בlocation אלא אם הlocation הוא הip של המחשב שנשלח ממנו ההודעה Location את הlocation להעביר לnew\_location באמצעות עטיפה בפרוטוקול ושליחה עם send לאחר מכן נשלח לip את סטטוס הפעולה |
| got\_clone | server – האובייקט של הcomm  Ip – הip ממנו נשלח המידע  Location – מיקום הקובץ\תיקייה  New\_location– שם חדש | שולחת לכתובת שמיקומה ניתן בLocation אלא אם הlocation הוא הip של המחשב שנשלח ממנו ההודעה את הlocation לשכפל לnew\_location באמצעות עטיפה בפרוטוקול ושליחה עם send לאחר מכן נשלח לip את סטטוס הפעולה |
| Got\_create | server – האובייקט של הcomm  Ip – הip ממנו נשלח המידע  Location – מיקום יצירה  Type – סוג היצירה | שולחת לכתובת שמיקומה ניתן בlocation אלא אם הlocation הוא הip של המחשב שנשלח ממנו ההודעה את הlocation ליצור עם הtype שהתקבל באמצעות עטיפה בפרוטוקול ושליחה עם send לאחר מכן נשלח לip את סטטוס הפעולה |
| Handle\_disconnect | server – האובייקט של הcomm  Ip - הip שהתנתק | קוראת לsend\_remvoe לכל הusers שמיוחסים לip לפי user\_comps ומוחקת את הip מכל המילונים הנדרשים |
| Send\_remove | Server – האובייקט של הcomm  Ip – למי לשלוח  Fold – התיקייה להוריד ממראה הקבצים | שולחת לip את הfold שעליו להוריד ממראה הקבצים באמצעות עטיפה בפרוטוקול ושליחה עם send |

| mainGeneralClient | | |
| --- | --- | --- |
| המחלקה של הלקוח שאינו בהכרח משתמש האחראית על הלוגיקה | | |
| שם משתנה | תפקיד | |
| Client\_rcv\_q | תור שבו יועברו ההודעות מclient\_Comm לLogic | |
| file\_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 | מנהלת את הלוגיקה |
| Handle\_ask\_file\_tree | client – האובייקט של הcomm | שולחת לשרת את עץ הקבצים באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Handle\_rename | Client – 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 – 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 |
| Handle\_move | Client – האובייקט של הclient\_comm  Location – מיקום הקובץ שרוצים לשנות  New\_location – מיקום הקובץ החדש | קוראת לmove בfilehandler ומחזירה תשובה לשרת אם הפעולה הצליחה או לא באמצעות עטיפה בפרוטוקל וsend |
| Handle\_clone | Client – האובייקט של הclient\_comm  Location – מיקום הקובץ שרוצים לשכפל  New\_location – שם הקובץ\תיקייה החדש | קוראת לclone בfilehandler ומחזירה תשובה לשרת אם הפעולה הצליחה או לא באמצעות עטיפה בפרוטוקל וsend |
| Handle\_open\_file | Server – האובייקט של הserver\_comm  Location – מיקום הקובץ | קוראת לget\_data בfilehandler ושולחת ללקוח משתמש את תוכן הקובץ באמצעות עטיפה בפרוטוקל ושליחה בsend |
| Handle\_change\_content | Server – האובייקט של הserver\_comm  Location – מיקום הקובץ  Data – תוכן קובץ חדש | קוראת לchange\_content בfilehandler ושולחת ללקוח משתמש אם הפעולה הייתה תקינה באמצעות עטיפה בפרוטוקל ושליחה בsend |

| mainUserClient | | |
| --- | --- | --- |
| המחלקה של הלקוח שהוא גם משתמש | | |
| שם משתנה | תפקיד | |
| rcv\_q | תור שבו יועברו ההודעות מclient\_Comm לLogic | |
| client | אובייקט מסוג client\_comm האחראי על התקשורת עם השרת | |
| Commands | מילון ששומר במפתחות שלו opcode של פעולות ובvalue את הפונקציה המתאימה לopcode | |
| monitor | עצם מסוג monitor שאחראי על ניטור התיקייה | |
| File\_q | תור שבו יועברו ההודעות מהניטור ללוגיקה | |
| פעולה | טענת כניסה | טענת יציאה |
| Main | None | מנהלת את הלוגיקה |
| Do\_register | Username – שם משתמש  Pass – סיסמה | שולחת לשרת נסיון להתחברות באמצעות עטיפה בפרוטוקול ושליחה עם send |
| Do\_connect | Username – שם משתמש  Pass - סיסמה | מוציאה את כתובת הMAC ושולחת לשרת את נסיון ההתחברות (אם פעולה זו לא תתבצע נסגור את התהליך של mainUserClinet) |
| Do\_remove | Fold – תיקייה המקושרת למחשב שהתנתק | מורידה את fold ממראה הקבצים |
| Handle\_got\_file\_tree | Data – עץ הקבצים שקיבל מהשרת | מסדר את עץ הקבצים במראה הקבצים |
| Handle\_file\_q \*thread | File\_q – התור שמיוחס לניטור | מוציא הודעות מהתור, אם ההודעה היא בעלת opcode שצריך בו לבצע pear to pear יוצר אובייקט של client\_comm ומתחבר אל השרת קבצים אשר נמצא אצל הip שמיוחס לdirectory ושולח לו את הפעולה שצריך לעשות ומחכה לתשובה. אחרת ישלח את הפעולה לשרת ויחכה לתשובה. בנוסף אם הפעולה מקומית יקרא לפעולה המתאימה בfilehandler |

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



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 | הצליח או לא – 2 | 02 | 02ok  02no |
| התחברות | לקוח | שרת | שם משתמש\_סיסמה  mac | 03 | 03ophir,12345, AC:BD:9D:F8:35:B2 |
| סטטוס התחברות | שרת | לקוח | הצליח או לא – 2 | 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 | הצליח או לא – 2 | 07 | 07ok  07no |
| שינוי שם | לקוח | שרת | מיקום, שם חדש | 08 | 081.1.1.1/folder/file.txt,text.txt |
| סטטוס שינוי שם | שרת | לקוח | הצליח או לא – 2 | 09 | 09ok  09no |
| מחיקה | לקוח | שרת | מיקום | 10 | 102.2.2.2/file.txt |
| סטטוס מחיקה | שרת | לקוח | הצליח או לא – 2 | 11 | 11ok  11no |
| העברה | לקוח | שרת | מיקום,מיקום | 12 | 124.4.4.4/file.txt,4.4.4.4/folder |
| סטטוס העברה | שרת | לקוח | הצליח או לא – 2 | 13 | 13ok  13no |
| שכפול | לקוח | שרת | מיקום,מיקום | 14 | 142.2.2.2/file.txt,4.4.4.4/folder |
| סטטוס שכפול | שרת | לקוח | הצליח או לא – 2 | 15 | 15ok  15no |
| הורדת תיקיית מחשב | שרת | לקוח | שם תיקייה | 30 | 302.2.2.2 |

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,אורך מידע, מידע  (אם יש) | 17 | 17ok,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 | הצליח או לא – 2 | 19 | 19ok  19no |

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 | אין | 31 | 31 |
| 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 | הצליח או לא – 2 | 22 | 22ok  22no |
| מחיקה | שרת | לקוח | מיקום | 23 | 232.2.2.2/file.txt |
| סטטוס מחיקה | לקוח | שרת | הצליח או לא – 2 | 24 | 24ok  24no |
| העברה | שרת | לקוח | מיקום,מיקום | 25 | 254.4.4.4/file.txt,4.4.4.4/folder |
| סטטוס העברה | לקוח | שרת | הצליח או לא – 2 | 26 | 26ok  26no |
| שכפול | שרת | לקוח | מיקום,מיקום | 27 | 272.2.2.2/file.txt,4.4.4.4/folder |
| סטטוס שכפול | לקוח | שרת | הצליח או לא – 2 | 28 | 28ok  28no |
| שליחת mac | לקוח | שרת | Mac | 29 | 29AC:BD:9D:F8:35:B2 |
| יצירה | שרת | לקוח | מיקום,סוג | 32 | 321.1.1.1/folder,fld |
| סטטוס יצירה | לקוח | שרת | הצליח או לא – 2 | 33 | 33ok  33no |

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']

**Opcode: 20**

הסבר: יישלח מחרוזת המתארת את מערכת הקבצים שנמצאת אצל הלוקח 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. 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):

* 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' - 'שם המערכת' – 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' - 'שם המערכת' – 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' - '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**