



פרויקט גמר במחשבים

בית ספר: הרצוג כפר סבא

שם העבודה: MicroStore

שם התלמיד: תומר וולוך

ת.ז. התלמיד: 215304304

שם המנחה: אופיר שביט

שם החלופה: הגנת סייבר ומערכות הפעלה

תאריך הגשה: 20.5.23

תוכן עניינים

5.....	מבוא.....	1.
6.....	ייזום.....	1.1
6.....	תיאור ראשוני ופונקציונלי של המוצר (המערכת).....	1.1.1
8.....	הגדרת הלקוח.....	1.1.2
9.....	יעדים פונקציונליים למערכת.....	1.1.3
9.....	הבעיה אותה אנו מנסים לפתור: אחסון בענן מקומי ולא בענן של חברה חיצונית.....	1.1.4
10.....	פתרונות קיימים בשוק (התחרות).....	1.1.5
12.....	סקירת הטכנולוגיה בפרויקט.....	1.1.6
14.....	תיחום הפרויקט.....	1.1.7
15.....	פירוט תיאור המערכת (איפיון).....	1.2
15.....	תיאור מפורט של המוצר (המערכת).....	1.2.1
16.....	פירוט היכולות לכל משתמש.....	1.2.2
18.....	פרוט הבדיקות במהלך הפיתוח.....	1.2.3
19.....	לו"ז לביצוע של הפרויקט.....	1.2.4
19.....	רשימת סיכונים בפרויקט ודרכי הפעולה לצמצום הסיכון.....	1.2.5
20.....	תיאור תחום הידע.....	2.
20.....	רשימת התכונות של המערכת עבור משתמש הקצה.....	2.1
20.....	רשימת התכונות של המערכת עבור צד השרת.....	2.2
21.....	יכולות כלליות של המערכת.....	2.3
22.....	מבנה \ ארכיטקטורה של הפרויקט.....	3.
22.....	תיאור הארכיטקטורה של המערכת.....	3.1
22.....	תאור החומרה.....	3.1.1
22.....	שרטוט המערכת.....	3.1.2
23.....	תיאור הטכנולוגיה.....	3.2
24.....	תיאור זרימת המידע במערכת.....	3.3
24.....	תיאור האלגוריתמים המרכזיים בפרויקט.....	3.4
25.....	תיאור סביבת הפיתוח.....	3.5
26.....	פרוטוקול התקשורת.....	3.6
27.....	תיאור מסכי המערכת.....	3.7
27.....	מסכי משתמש קצה בטלפון IOS.....	3.7.1
30.....	מסכי משתמש קצה במחשב אישי WINDOWS.....	3.7.2
32.....	תיאור מבנה הנתונים.....	3.8
33.....	סקירת חולשות ואיומים.....	3.9
34.....	מימוש הפרויקט.....	4.

34.....	סקירת כל המודלים \ מחלקות מיובאות של המערכת	4.1
34.....	מרכיבי המערכת בשרת	4.1.1
34.....	מרכיבי המערכת ביחידת משתמש קצה מסוג מחשב אישי	4.1.2
34.....	מרכיבי המערכת ביחידת משתמש קצה מסוג טלפון IOS	4.1.3
35.....	סקירת כל המודלים \ מחלקות שאותם פיתחתי של המערכת	4.2
35.....	מרכיבי המערכת בשרת	4.2.1
36.....	מרכיבי המערכת ביחידת משתמש קצה מסוג מחשב אישי	4.2.2
37.....	מרכיבי המערכת ביחידת משתמש קצה מסוג טלפון IOS	4.2.3
39.....	תיאור מפורט של האלגוריתמים "מעניינים" בפרויקט	4.3
39.....	פיתוח ומימוש פרוטוקול תקשורת שרת-לקוח	4.3.1
39.....	מימוש הצפנה היברידית באמצעות אלגוריתם דיפי-הלמן ו AES	4.3.2
39.....	מימוש אלגוריתם לניהול הקבצים והרשאות המשתמשים השונים	4.3.3
39.....	מימוש מערך נתונים להקמת משתמש חדש, יצירת סיסמה עבורו והכנסתו למערכת	4.3.4
40.....	מערך\מסמך הבדיקות של הפרוייקט	4.4
40.....	בדיקת יצירת חשבון ע"י שם משתמש וסיסמה	4.4.1
40.....	בדיקת תקשורת שרת-לקוח מוצפנת	4.4.2
40.....	בדיקת גישה מרחוק לקבצים מכל מחשב windows או iPhone	4.4.3
41.....	בדיקת יצירת תיקיות	4.4.4
41.....	בדיקת העלאת קבצים	4.4.5
41.....	בדיקת הורדת קבצים	4.4.6
42.....	בדיקת העברת קבצים בין תיקיות	4.4.7
42.....	בדיקת מחיקת תיקיות וקבצים	4.4.8
42.....	בדיקת העתקת קבצים	4.4.9
43.....	בדיקת אפשרות לכבות ולהדליק את השרת תוך שמירה על כלל המידע של המשתמשים	4.4.10
44.....	מדריך למשתמש	5.
44.....	עץ הקבצים	5.1
45.....	התקנת המערכת	5.2
45.....	הפעלת המערכת	5.3
46.....	סיכום אישי	6.
47.....	ביבליוגרפיה	7.

שם התלמיד: תומר וולוך / שם החלופה: הגנת סייבר ומערכות הפעלה / שם הפרויקט: MicroStore

דף זה הוא ריק

1. מבוא

שורשיו של מחשוב הענן החלו בשנות ה-60, אז הוצעה לראשונה "רשת מחשבים אינטר-גלקטית". בשנת 1963, DARPA, הסוכנות לפרויקטי מחקר מתקדמים של משרד ההגנה האמריקאי, שילמה 2 מיליון דולר עבור הפרויקט. הסבסוד כלל תנאי מוקדם ופשוט, ליצור טכנולוגיות חדשות המאפשרות שימוש ב"מחשב שישמש לפחות שני אנשים".

עבור המצב הזה, אחד מאותם מחשבים עתיקי יומין שהשתמשו בסלילי קלטת לזיכרון הפך לאב טיפוס למה שנודע כיום כמחשוב ענן.

בשנת 1969, ג'יי סי אר ליקלידר הקים את ARPANET גרסה בסיסית ביותר של האינטרנט, וכך קידם חלום שנקרא "רשת המחשבים הבין-גלקטית", שבה כולם על פני כדור הארץ יהיו מחוברים זה לזה באמצעות מחשבים אישיים, ויוכלו לגשת למסדי נתונים מכל מקום בגלובוס. [7] Error!

Reference source not found.



בשנת 2023, 60 שנים לאחר ההצעה הראשונה להקמה של אחסון ענן, בחרתי לממש בעצמי שירות אחסון קבצים ענני-מקומי (ממוקם פיזית אצל המשתמש) שנגיש למשתמשים פרטיים שמקנה מקסימום פרטיות ומאפשר לשתף קבצים.

1.1 ייזום

1.1.1 תיאור ראשוני ופונקציונלי של המוצר (המערכת)

הקדמה

כיום קיים צורך ברור לגשת לקבצים שלנו מכל המכשירים הדיגיטליים שלנו.



בנוסף אנשים מעוניינים לשתף את קרוביהם, עמיתיהם לעבודה וחבריהם בתמונות, סרטונים, מסמכים וקבצים נוספים.

לכן בחרתי לפתח שירות אחסון ושיתוף קבצים ענני-מקומי שנמצא בבית\משרד הלקוח כדי לענות על צרכים אלו.

הפרויקט שאפתח הינו שירות אחסון ושיתוף קבצים ענני-מקומי (שרת מקומי שנמצא אצל הלקוח פיזית) למספר משתמשים.

המערכת מאפשרת, לכל משתמש, יצירת חשבון מאובטח הדורש שם משתמש וסיסמה. לאחר יצירת החשבון, יכול המשתמש להעלות, ולהוריד קבצים באמצעות מחשב או טלפון (iPhone) שיפנה לשרת.

בנוסף השירות יתמוך בתיקיות משותפות, שישותפו על ידי המשתמש למשתמשים אחרים, המשתמש ששיתף יוכל להגדיר הרשאות לכל משתמש ששותף.

MicroStore שירות אחסון ושיתוף קבצים בבית הלקוח

שירות שיתוף הקבצים MicroStore הינו שירות חדשני ויעיל לאחסון ושיתוף קבצים בענן עבור מספר משתמשים באותו ארגון.

השירות תוכנן לאחר לימוד של שוק האחסון בענן (ראה סעיפים בהמשך על השוק והתחרות) ואמור לתת מענה לפלח שוק צומח - מערכת עם מספר משתמשים מוגבל (ולא אינסופי כמו בענן) שרוצה להחזיק את המידע אצלו "בתוך הבית/המפעל".

שירות MicroStore מיועד למשפחות, חברות קטנות ואנשים פרטיים שרוצים לשמור ולחלוק קבצים דיגיטליים "בענן מקומי" בצורה מאובטחת תוך שמירה מקסימלית על פרטיות המידע שלהם.

תיאור השירות

- מערכת MicroStore מספקת שיתוף ושמירת קבצים, כאשר השרת נמצא אצל הלקוח (שירות ענני-מקומי).
- כל משתמש יוצר שם משתמש וסיסמא
- כל משתמש יכול להעלות כל סוג של קובץ דיגיטלי לשרת המקומי.
- משתמש שהעלה קובץ ישמש כADMIN של הקובץ ויכול לתת הרשאות שימוש למשתמשים נוספים
- משתמש יכול למחוק קובץ שהעלה לשרת
- משתמש יכול להעתיק (לשכפל) קובץ שהעלה בתוך השרת
- משתמש יכול להזיז קובץ שהעלה בשרת
- משתמש יכול לשנות שם של קובץ שנמצא בשרת
- כל משתמש יכול ליצור תיקיות ולהכניס אליהם תכנים דיגיטליים מכל סוג שהוא בשרת.
- משתמש יכול להוריד קובץ מהשרת למכשיר הקצה.
- משתמש שיצר קובץ יכול לתת את ההרשאות הבאות לשיתוף קובץ למשתמשים אחרים:

○ הרשאה נמוכה

אפשרות להוריד את הקובץ

אפשרות להעתיק את הקובץ

○ הרשאה גבוהה

אפשרות להוריד את הקובץ

אפשרות להעתיק את הקובץ

אפשרות למחוק את הקובץ

אפשרות להזיז את הקובץ

אפשרות לשנות שם לקובץ

אפשרות שיתוף הקובץ

- השירות יהיה נגיש ממכשירי קצה מהסוגים:
 - מחשבים מבוססי מערכת הפעלה WINDOWS
 - טלפונים סלולריים מבוססי מערכת הפעלה IOS

1.1.2 הגדרת הלקוח

הפרויקט אותו הגדרתי אמור לשרת מגוון רחב של לקוחות ויישומים ובכך לצבור פופולריות רבה בקרב מספר משתמשים רב (מטרת העל).

להלן קהל היעד והשימושים אותם אני מייעד למערכת.

- משתמשים בחברה, כגון משפחה – משתמשים שמנהלים את הקבצים שלהם כגון תמונות, סרטונים, מסמכים וכו' בענן מקומי. הורים יוכלו לשתף עם ילדיהם הקטנים תמונות וסרטונים בהרשאות נמוכות וכך לוודא שהילדים לא בטעות מוחקים את אותם קבצים.
- משתמשים בחברה מסחרית – החברה יכולה לנהל, על שרת מקומי, תיקיות לפי מחלקות כאשר לכל מחלקה יש את התיקיות שלה ויש תיקיות משותפות בין מחלקות. יתרון נוסף הוא שמשתמשים בחברה יכולים לשתף ביניהם קבצים גדולים שלא יכולים לעבור במייל.
- משתמשים עצמאיים – משתמשים שלא מעוניינים לשתף אנשים אחרים בקבצים אלא רק שיהיה להם שרת מאובטח לאחסון קבצים שיוכלו לגשת אליו.



1.1.3 יעדים פונקציונליים למערכת

- שימוש מקבילי במערכת בידי מספר משתמשים בו-זמנית
- אחסון קבצים דיגיטליים בשרת ענן מקומי
- גישה לקבצים דיגיטליים בשרת בצורה מאובטחת, מוצפנת ופרטית באמצעות שם משתמש וסיסמא



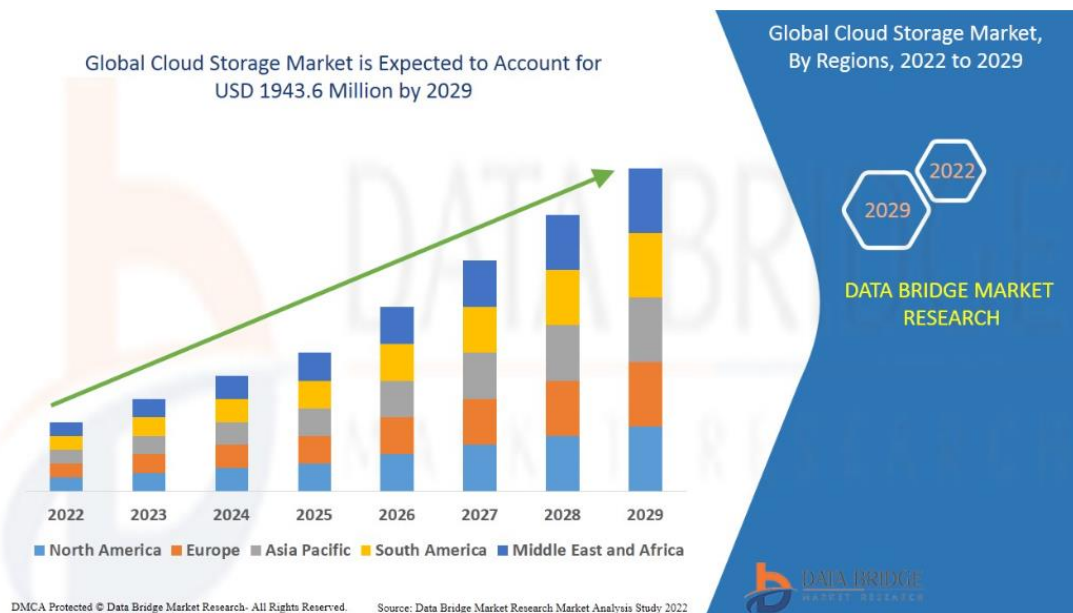
- יכולת שיתוף קבצים באופן סלקטיבי בין משתמשים ע"י הרשאות.
- ממשק משתמש נוח וידידותי למשתמש שינגיש את השירות למשתמשי הקצה מכל מכשיר דיגיטאלי.

1.1.4 הבעיה אותה אנו מנסים לפתור: אחסון בענן מקומי ולא בענן של חברה חיצונית

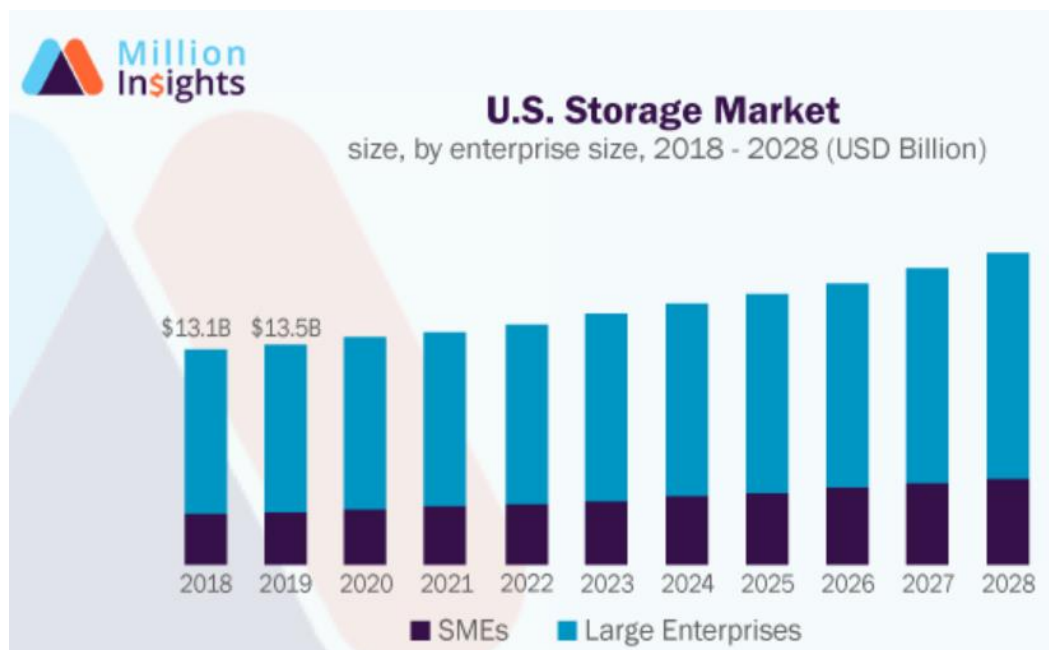
בשנים האחרונות, ישנה דרישה גדלה והולכת לשמירת המידע בצורה מקומית מאובטחת. חברות קטנות ואנשים פרטיים רוצים לשמור את המידע שלהם נגיש ובמקביל לשמור על פרטיות המידע ולוודא מעל לכל ספק שרק הם יכולים לגשת למידע זה. שוק אחסון הענן-המקומי צומח בשנים האחרונות, אבל קטן משמעותית ולא מהווה עניין עבור החברות הגדולות.

1.1.5 פתרונות קיימים בשוק (התחרות)

שוק האחסון בענן הינו שוק גדול ואינסופי. על פי מחקר שנעשה ע"י חברת DATA BRIDGE שוק זה אמור להגיע דמיוני של כמעט 2 טריליון דולר בשנת 2029. [7]



כיום רובו המוחלט של אחסון הענן נועד עבור צרכים מסחריים (שימוש ע"י חברות) ומיעוטו ע"י צרכנים פרטיים. גם בסקטור המסחרי, השימוש הנרחב נעשה ע"י חברות גדולות, כפי שמראה המחקר שנעשה ביבשת אמריקה ע"י חברת MILLION INSIGHTS. [7]



שוק האחסון בענן מספק עניין רב ליצרנים וספקי אחסון רבים, שכוללים חברות ענקיות כגון אמזון ומיקרוסופט, שגם עוסקות בתחומים אחרים, וכן ספקים ייעודיים, כגון דרופבוקס, שמתמחים בשירותי אחסון ענן בלבד.

רבים מהשחקנים השולטים בשוק מאפשרים אחסון חינמי של עד כ-200 GB מה שמיצר חסם כניסה גבוהה לשחקנים חדשים. למרות זאת, ובגלל ההזדמנות הכלכלית של שוק ענק זה, החלטתי לנסות ולהתחרות בשוק כאשר אני פונה לנישה מצומצמת שאינה מעניינת את השחקנים הגדולים בשוק.

הנישה אותה בחרתי היא: אנשים פרטיים וחברות קטנות עם מספר עובדים קטן, שמעוניינות להחזיק את המידע הדיגיטלי אצלן "בבית" ולא אצל ספקיות אחסון הענן הגדולות. שוק זה לא מכוסה היום ע"י החברות הגדולות, שמתרכזות בשוק המסחרי ובעיקר בחברות הגדולות שבו.

להלן רשימת השחקניות הגדולות בשוק, שפונות בעיקר לשווקים המסחריים גדולים:

AMAZON DRIVE -

GOOGLE DRIVE -

MICROSOFT ONE DRIVE -

DROPBOX -

BLOMP -

- ועוד רבים אחרים [7]



1.1.6 סקירת הטכנולוגיה בפרויקט

בפרויקט שאממש ישנה תקשורת רבה בין השרת ללקוח. לכן אממש את השרת והלקוח (PC) בpython. מאוד נוח לממש תקשורת באמצעות python. את הלקוח בטלפון (ios) אממש בשפת swift וזאת בגלל שזו השפה המרכזית למימוש בios. את הממשק בטלפון אממש באמצעות UIKit.

הספריות שבהן אשתמש

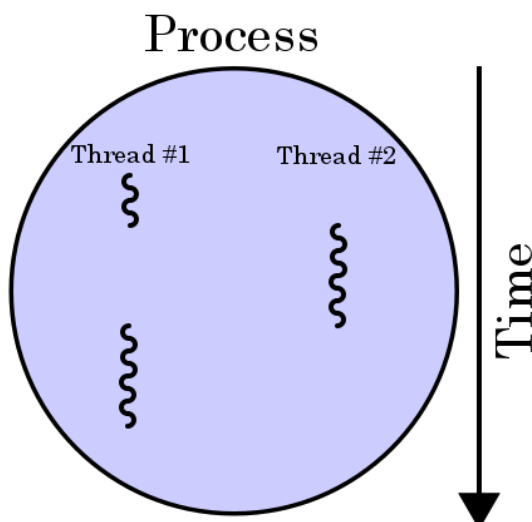
- socket – הוא ממשק תכנות תקשורת המסופק בדרך כלל על ידי מערכת ההפעלה, ומאפשר ליישומי התוכנה לשלוט ולהשתמש בנקודות קצה שהם עבור זרם נתונים בתקשורת בין תהליכים על גבי רשת מחשבים. אשתמש בספריה זו בכדי לפתח את



התקשורת בפרויקט בעזרת פרוטוקול שאותו אני אפתח. יכולה להיווצר בעיה בפיתוח באמצעות socket בswift לכן יכול להיות שאאלץ לנהל תקשורת על גבי

https בתקשורת מול הטלפון. [7]

- Thread - מערכות הפעלה מאפשרות לנהל במסגרת ריצה של תהליך (Process) מספר תהליכונים הרצים במקביל במרחב כתובות אחד. במערכות אלו כל תהליך חדש מתחיל את ביצועו באמצעות 'תהליכון ראשי' אשר עשוי בהמשך ליצור תהליכונים נוספים.



מנגנון הריצה באמצעות תהליכונים מאפשר לספק למשתמש במערכת ההפעלה מהירות תגובה ורציפות פעולה כאשר התהליך (יישום) מבצע כמה משימות במקביל. תחושת המקביליות מתקבלת באמצעות קפיצה מריצה בהקשר של תהליכון אחד לריצה בהקשר של תהליכון שני. קפיצה זו, המכונה

החלפת הקשר (Context switch), מבצעת מעין צילום מצב של המעבד (אוגרים וכדומה) לפני נטישה של תהליכון מסוים כדי לאפשר מעבר לתהליכון אחר וחזרה לתהליכון הקודם בהמשך. [7]

שם התלמיד: תומר וולוך / שם החלופה: הגנת סייבר ומערכות הפעלה / שם הפרויקט: MicroStore

- Pygame/wxpython/tkinter – עבור הממשק הגרפי הן בצד השרת והן בצד הלקוח (pc) אפתח באמצעות אחת או יותר מהספריות הבאות. לכל ספריה יש את היתרונות שלה ואני אצטרך להשוות ולמצוא איזו מתאימה לפרויקט שלי.



- UIKit - מספק תצוגות, פקדים ומבני פריסה בתוך ממשק המשתמש של האפליקציה. framework מספקת מטפלי אירועים להעברת הקשות, מחוות וסוגים אחרים של קלט



לאפליקציה, וכלים
לניהול זרימת הנתונים
מהקלט של המשתמש
אל האפליקציה. לפיתוח
בUIKit יש ממשק נוח
המאפשר עיצוב של
ממשק המשתמש ללא
שימוש בקוד. [7]

כמו כן אצטרך למצוא ממשקים על מנת לנהל את הקבצים בצד השרת (ככל הנראה ממשקים
osn וכן ממשק לנהל את מסדי הנתונים ככל הנראה sqlite).

1.1.7 תיחום הפרויקט

הפרויקט יעסוק בתחומים הבאים:

- תקשורת - התכונה העיקרית של הפרויקט היא "שהוא ענן" כלומר המידע לא נשמר לוקאלית על המכשיר לכן אני אצטרך לבנות פרוטוקול תקשורת ולמצוא דרכים לנהל תקשורת עם כמות גדולה של מידע (לדוגמא קבצים גדולים)
 - הצפנות – כל התקשורת תהיה מוצפנת בהצפנה היברידית (AES, דיפי-הלמן). במידה שבSwift לא אוכל לממש זאת אשתמש בhttps ולשם כך אשתמש בcertificate ודברים נוספים. כמו כן הסיסמאות נשמרות בצד השרת לאחר שעברו salted hash.
 - פיתוח אפליקציות – אלמנט נוסף חשוב בפרויקט הוא ממשק המשתמש. אני אפתח אפליקציה לטלפון באמצעות פלטפורמות לפיתוח אפליקציות וכן אפליקציה עבור השרת והלקוח במחשב.
 - מסדי נתונים – בצד השרת אצטרך לנהל מסד נתונים שישמור את הפרטים של המשתמש. כמו כן אצטרך לנהל מסד נתונים עבור ההרשאות לתיקיות.
 - מערכות הפעלה – בצד השרת אצטרך לנהל את כל הקבצים והתיקיות של כל המשתמשים. כמו כן אצטרך לנהל תהליכונים בצד השרת עבור כל התחברות של לקוח. ובצד הלקוח עבור שילוב בין התקשורת לGUI.
- הפרויקט לא יעסוק בתוכן הקבצים. כלומר הפרויקט לא יוודא שהקבצים אינם זדוניים, כמו כן הוא לא יאפשר עריכה של הקבצים מתוך השרת אלא יחייב להוריד את הקובץ ולהעלותו שוב. נוסף לכך הפרויקט לא יתמוך באנדרואיד ולינוקס בשלב זה.

1.2 פירוט תיאור המערכת (איפיון)

1.2.1 תיאור מפורט של המוצר (המערכת)

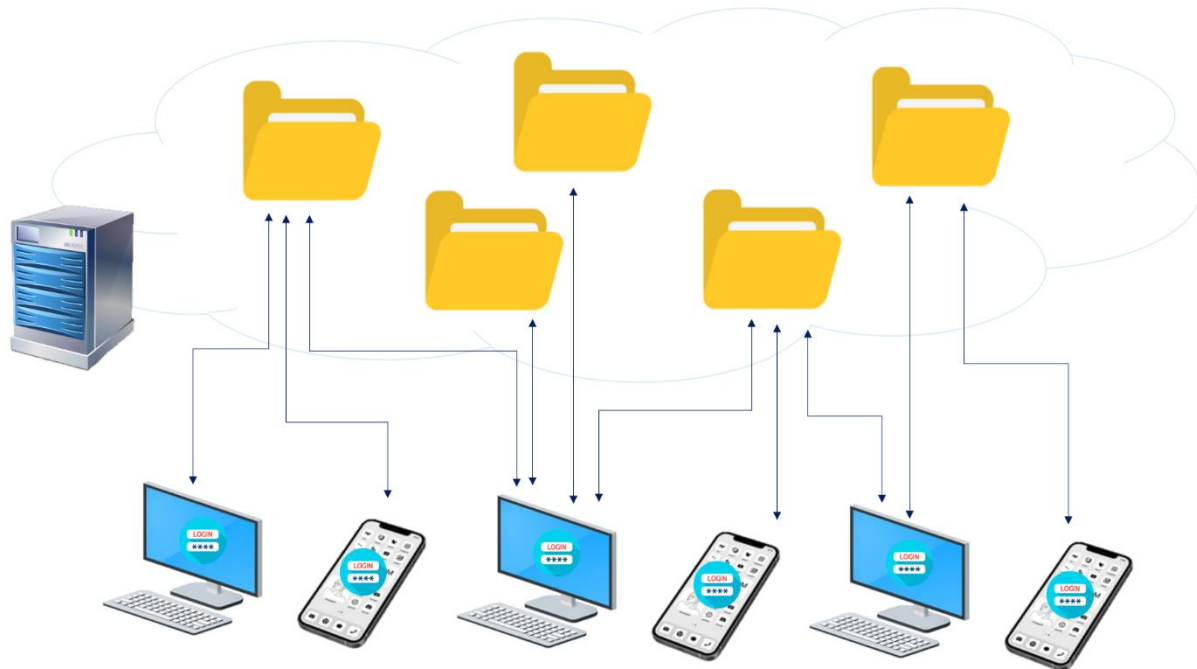
המוצר הנה מערכת אחסון ענן מקומי היושבת פיזית אצל הלקוח המכילה את האלמנטים הבאים:

- שרת אחסון לכל סוגי הקבצים הדיגיטליים, כגון מסמכים, תמונות, סרטונים וכו'.
- אפליקציית קצה למכשירים ניידים מבוססי IOS אשר מתקשרת עם שרת האחסון
- אפליקציית קצה למחשבים (נייחים וניידים) מבוססי Windows אשר מתקשרת עם שרת האחסון
- להלן תיאור פונקציונלי של האפליקציה (מעל iOS ומעל windows) ושל השרת אחסון:
 - יצירת חשבון ע"י שם משתמש וסיסמה (פעולה זו תיעשה מטעם האחראי על השרת בצד השרת)
 - שמירת הסיסמה מוצפנת בצד השרת
 - תקשורת שרת-לקוח מוצפנת
 - יצירת תיקיות
 - העלאת קבצים
 - הורדת קבצים
 - העברת קבצים בין תיקיות
 - מחיקת תיקיות וקבצים
 - העתקת תיקיות וקבצים
 - שיתוף קבצים בין משתמשים שונים ברמות הרשאה שונות.

1.2.2 פירוט היכולות לכל משתמש

לאחר שהפרק הקודם תיאר את המאפיינים של המערכת ברמה הכללית, נפרט בפרק זה את מכלול השירותים והפונקציונליות של המערכת.

שירטוט המערכת ותיאורה



יכולות כלל המערכת

- לכל משתמש יש תיקיית root ובה הוא יכול ליצור עוד תיקיות ולהוסיף קבצים.
- המערכת מאפשרת למספר רב של מכשירי קצה (מחשבים וטלפונים ניידים) לגשת לשרת מרוחק.
- כל משתמש יכול ליצור קבצים ולשתף אותם עם משתמשים אחרים.
- לכל משתמש יש את אחת מההרשאות הבאות:
 - ADMIN – יוצר הקובץ ובעל כלל ההרשאות
 - VIEWER – משתמש אשר יכול אך ורק לראות, להוריד ולהעתיק קבצים מהתיקיה
 - EDITOR – אותן הרשאות כמו VIEWER ויכול גם למחוק, לשנות שם, להזיז ולשתף את הקובץ בו הוא שותף
- כל משתמש צריך ליצור שם משתמש וסיסמא בכדי להתחבר לשירות

רשימת התכונות של המערכת עבור משתמש הקצה

לכל משתמש יש את השירותים הבאים:

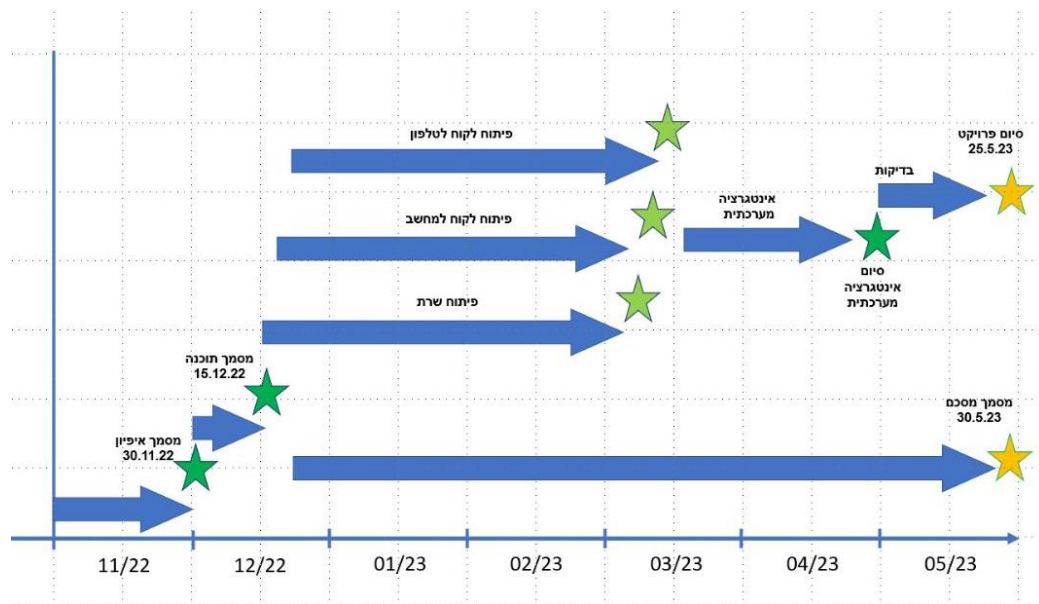
- יצירת חשבון ע"י שם משתמש וסיסמה
- שמירת הסיסמא מוצפנת בצד השרת
- תקשורת שרת-לקוח מוצפנת
- גישה מרחוק לקבצים מכל מחשב windows או iPhone
- יצירת תיקיות
- העלאת קבצים
- הורדת קבצים
- העברת קבצים בין תיקיות
- מחיקת תיקיות וקבצים
- העתקת תיקיות וקבצים

1.2.3 פרוט הבדיקות במהלך הפיתוח

לאחר סיום תהליך הקידוד והאינטגרציה מתוכנן להקדיש שלושה שבועות לבדיקת המערכת. בפועל בגלל עיכובים בלוחות זמנים התקצר תהליך הבדיקות ויארך מספר ימים. להלן הבדיקות אותם אבצע ברמת המערכת:

- בדיקת יצירת חשבון ע"י שם משתמש וסיסמה
 - בדיקת תקשורת שרת-לקוח מוצפנת
 - בדיקת גישה מרחוק לקבצים מכל מחשב windows או iPhone
 - בדיקת יצירת תיקיות
 - בדיקת העלאת קבצים
 - בדיקת הורדת קבצים
 - בדיקת העברת קבצים בין תיקיות
 - בדיקת מחיקת תיקיות וקבצים
 - בדיקת העתקת תיקיות וקבצים
 - בדיקת ממשק צד שרת נוח וידידותי בוא יוכל לעשות את כל הפעולות מצד השרת
 - בדיקת יצירת חשבון עבור משתמש חדש בשרת והוספתו לרשימת המשתמשים המורשים
 - בדיקת הסרת משתמשים מרשימת המורשים בשרת
 - בדיקת אפשרות לכבות ולהדליק את השרת תוך שמירה על כלל המידע של המשתמשים
- לאחר סיום הבדיקות הנ"ל אבצע בדיקות של רצף פעולות משורשרות אחת לשניה מהרשימה הנ"ל.

1.2.4 לו"ז לביצוע של הפרויקט



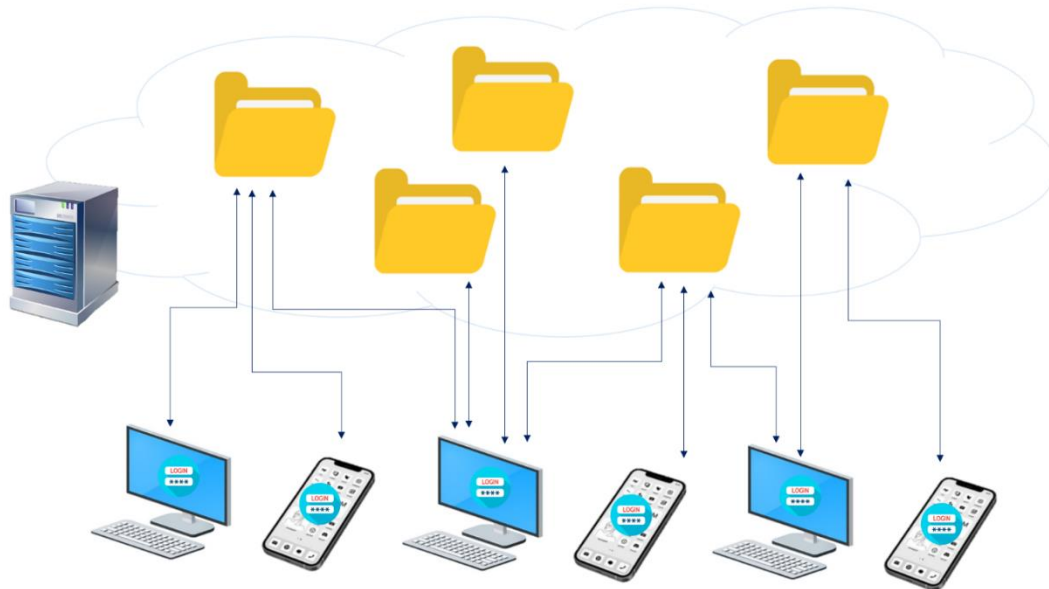
1.2.5 רשימת סיכונים בפרויקט ודרכי הפעולה לצמצום הסיכון

פרק זה סוקר את הסיכונים הפוטנציאליים בפרויקט ואת הדרכים לצמצום הסיכון שלהם.

- **הסיכון:** סיום הפרויקט במסגרת זמן שהוקצב. פרויקט תוכנה מורכב ומסובך שיכול להסתבך ולהגרר מעל למסגרת הזמן
צמצום הסיכון: תכנון ואפיון מפורטים לפני תחילת מימוש. בניית לו"ז מפורט, ומעקב צמוד אחרי התקדמות הפיתוח
- **הסיכון:** סיום פיתוח מוצלח של האפליקציה מעל מערכת ההפעלה iOS. אצטרך ללמוד פיתוח אפליקציה ב-SWIFT יתכן ויש צורך בפרוצדורה אדמיניסטרטיבית בכדי ליצור אפליקציה מעל iOS
צמצום הסיכון: אתחיל לטפל בנושא בשלב הראשון של הפרויקט בכדי לצמצם למינימום את הסיכון הנ"ל.
- **הסיכון:** עמידה ביעד של יצירת "חווית משתמש" טובה. יצירת חווית משתמש טובה היא משימה טכנולוגית קשה מאוד.
צמצום הסיכון: מעבר על ממשקים של אפליקציות קימות ופופולאריות מעולם האחסון (כגון דרופבוקס) ליצירת "חווית משתמש" טובה.

2. תיאור תחום הידע

לאחר שהפרק הקודם תיאר את המאפיינים של המערכת ברמה הכללית, נפרט בפרק זה את מכלול השירותים והפונקציונליות של המערכת בפירוט.



2.1 רשימת התכונות של המערכת עבור משתמש הקצה

לכל משתמש יש את השירותים הבאים:

- יצירת חשבון ע"י שם משתמש וסיסמה
- שמירת הסיסמא מוצפנת בצד השרת
- תקשורת שרת-לקוח מוצפנת
- גישה מרחוק לקבצים מכל מחשב windows או iPhone
- יצירת תיקיות
- העלאת קבצים
- הורדת קבצים
- העברת קבצים בין תיקיות
- מחיקת תיקיות וקבצים
- העתקת תיקיות וקבצים

2.2 רשימת התכונות של המערכת עבור צד השרת

- אפשרות לכבות ולהדליק את השרת תוך שמירה על כלל המידע של המשתמשים

2.3 יכולות כלליות של המערכת

- לכל משתמש יש תיקיית root ובה הוא יכול ליצור עוד תיקיות ולהוסיף קבצים.
- המערכת מאפשרת למספר רב של מכשירי קצה (מחשבים וטלפונים ניידים) לגשת לשרת מרוחק.
- כל משתמש יכול ליצור קבצים ולשתף אותם עם משתמשים אחרים.
- תיקיה יכולה להכיל תיקיות פנימיות, אשר יורשות את ההרשאות של תיקיה בה הם נמצאים
- לכל משתמש יש את אחת מההרשאות הבאות:
 - VIEWER – משתמש אשר יכול אך ורק לראות, להוריד ולהעתיק קבצים מהתיקייה
 - EDITOR – אותן הרשאות כמו VIEWER ויכול גם למחוק, לשנות שם, להזיז ולשתף את הקובץ בו הוא שותף
- כל משתמש צריך ליצור שם משתמש וסיסמא בכדי להתחבר לשירות

3. מבנה \ ארכיטקטורה של הפרויקט

פרק זה יתאר את המימוש החומרתי והתוכנתי של המערכת

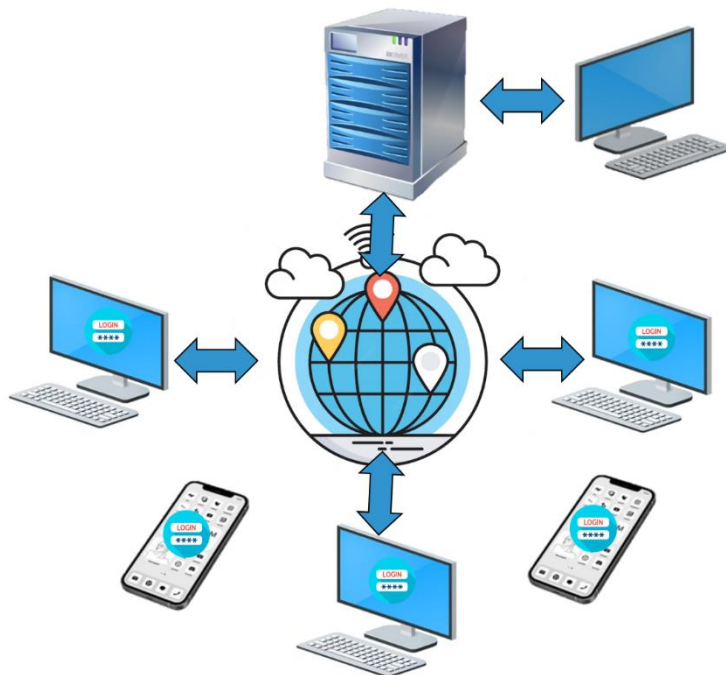
3.1 תיאור הארכיטקטורה של המערכת

3.1.1 תאור החומרה

המערכת מורכבת ממספר אלמנטים:

- צד השרת: שרת פיזי (מחשב אישי) בעל מערכת הפעלה מסוג WINDOWS וכן ממשק מנהל שרת
 - משתמשי קצה: יחידות קצה מסוג טלפון בעלי מערכת הפעלה IOS וכן מחשבים אישיים בעלי מערכת הפעלה מסוג WINDOWS
- היחידות מחוברות בניהן ע"י פרוטוקול אינטרנט מעל כל תשתית פיזית (קוי ו\או אלחוטי)

3.1.2 שרטוט המערכת



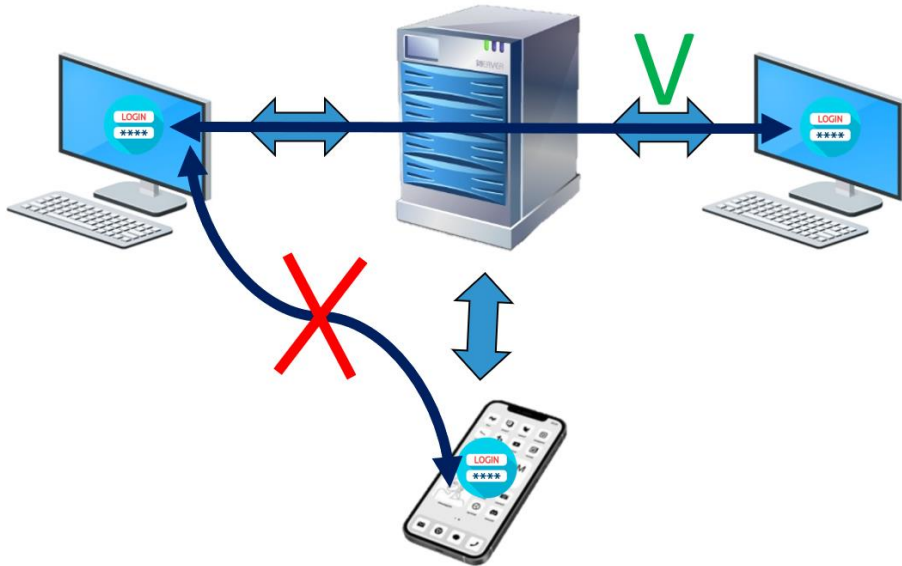
3.2 תיאור הטכנולוגיה

להלן רשימת שפות התכנות, מערכות ההפעלה וכו' שהשתמשתי למימוש הפרויקט:

- מימוש השרת על בסיס מחשב אישי
 - מערכת הפעלה מסוג WINDOWS
 - שפות תכנות Python
 - ניהול נתונים על בסיס SQL
 - שימוש בפרוטוקלי הצפנות מסוג דיפי-הלמן AESI
 - פיתוח יכולות השרת ופרוטוקול בשכבת האפליקציה כפי שמתואר במסמך זה על בסיס הסעיפים הקודמים
- מימוש משתמש קצה מסוג מחשב אישי
 - מערכת הפעלה מסוג WINDOWS
 - שפות תכנות Python
 - שימוש בפרוטוקלי הצפנות מסוג דיפי-הלמן AESI
 - פיתוח יכולות משתמש הקצה ופרוטוקול בשכבת האפליקציה כפי שמתואר במסמך זה על בסיס הסעיפים הקודמים
- משתמש קצה מסוג טלפון
- מערכת הפעלה מסוג IOS
- שפות תכנות SWIFT
- שימוש בפרוטוקלי הצפנות מסוג דיפי-הלמן AESI
- פיתוח יכולות משתמש הקצה ופרוטוקול בשכבת האפליקציה כפי שמתואר במסמך זה על בסיס הסעיפים הקודמים

3.3 תיאור זרימת המידע במערכת

זרימת המידע במערכת הנה בטופולוגיית STAR. כלומר, כל יחידת קצה מעבירה ומקבלת מידע מהשרת המרכזי. יחידות הקצה אינן מעבירות מידע ישירות בניהן וכל העברת מידע נעשת דרך השרת בלבד, כפי שמתואר בשרטוט המצורף.



3.4 תיאור האלגוריתמים המרכזיים בפרויקט

להלן רשימת האלגוריתמים המרכזיים שפתחתי בפרויקט:

- פיתוח ומימוש פרוטוקול תקשורת שרת-לקוח
- מימוש הצפנה היברידית באמצעות אלגוריתם דיפי-הלמן ו AES
- מימוש אלגוריתם לניהול הקבצים והרשאות המשתמשים השונים
- מימוש מערך נתונים להקמת משתמש חדש, יצירת סיסמה עבורו והכנסתו למערכת
- שמירת כלל הנתונים במערכת, קבצים והרשאות, גם לאחר כיבוי השרת

3.5 תיאור סביבת הפיתוח

להלן סביבות הפיתוח שאיתן עבדתי למימוש הפרויקט:

- מימוש השרת נעשה בסביבת הפיתוח Visual Studio Code של חברת מיקרוסופט. כמו כן השתמשתי בסביבת Python Virtual Environment
- מימוש משתמש הקצה מעל מחשב אישי בסביבת הפיתוח Visual Studio Code של חברת מיקרוסופט. כמו כן השתמשתי בסביבת Python Virtual Environment
- מימוש משתמש הקצה מעל טלפון IOS בסביבת הפיתוח Xcode של חברת אפל

3.6 פרוטוקול התקשורת

Start of comms HELLO~	Server to client	HELLO
Msg for login LOGIN~username~password	Client to server	LOGIN
Msg for signup: SGNUP ~username~password	Client to server	SGNUP
Acknowledge msg for SGNUP and LOGIN	Server to client	LOGAK
Requesting dir content: CHGDR~dirname	Client to server	CHGDR
Returning dir content: CHGDR~dirname~filelist	Server to client	RECHD
Requesting to delete file/folder: REMOV~name	Client to server	REMOV
Acknowledge msg for REMOV: REMAK~name	Server to client	REMAK
File upload: UPLD~filename~data	Client to server	UPLD
Acknowledge msg for UPLD: UPLAK~filename	Server to client	UPLAK
Requesting file data: DWNLD~filename	Client to server	DWNLD
Return for DWNLD: REDWN~filename~data	Server to client	REDWN
Requesting to move file/folder: MOVFL~src~dst	Client to server	MOVFL
Acknowledge msg for MOVFL: MOVAK~src~dst	Server to client	MOVAK
Requesting to copy file/folder: MOVFL~src~dst	Client to server	COPYP
Acknowledge msg for COPYP: CPYAK~src~dst	Server to client	CPYAK
Request to rename file/folder MOVFL~old~new	Client to server	CHGNM
Acknowledge msg for CHGNM: CHGAK~old~new	Server to client	CHGAK
Request to create a new dir: MKDIR~dirname	Client to server	MKDIR
Acknowledge msg for MKDIR: MKDAK~dirname	Server to client	MKDAK
Request to share a file: SHARE~user~path~perm	Client to server	SHARE
Acknowledge msg for SHARE: SHRAK~path	Server to Client	SHRAK
Request to exit: EXITT~goodbye	Both ways	EXITT
Acknowledge msg for EXITT: EXTAK~Goodbye	Both ways	EXTAK
Error msg: ERROR~errorcode~description	Server to client	ERROR

השרת עובד בצורה סינכרונית

אורך קוד ההודעה הוא 5 תווים

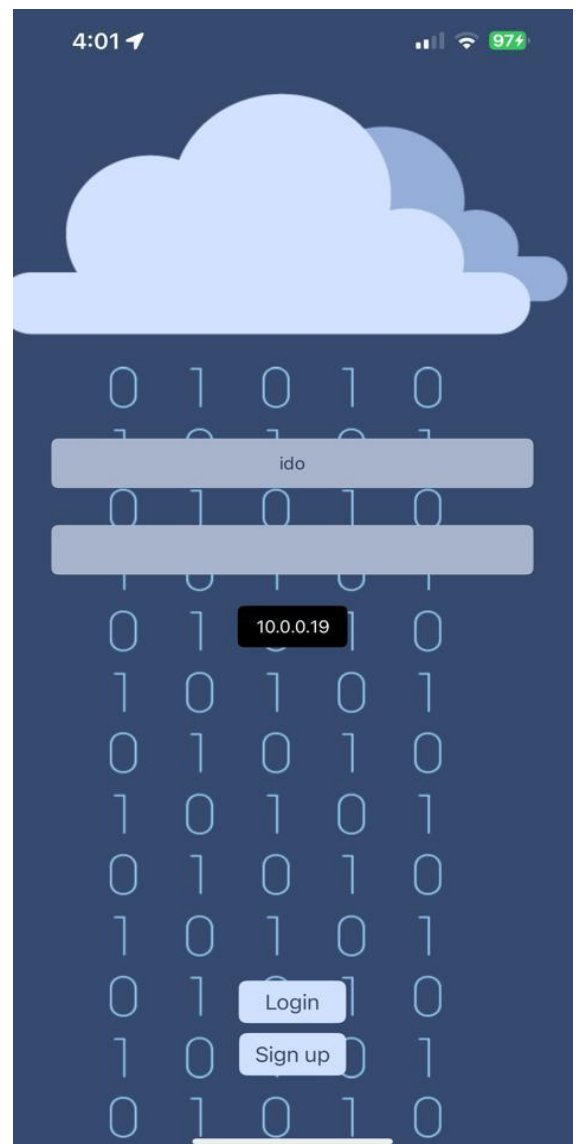
השדות מופרדים על ידי תו ~

3.7 תיאור מסכי המערכת

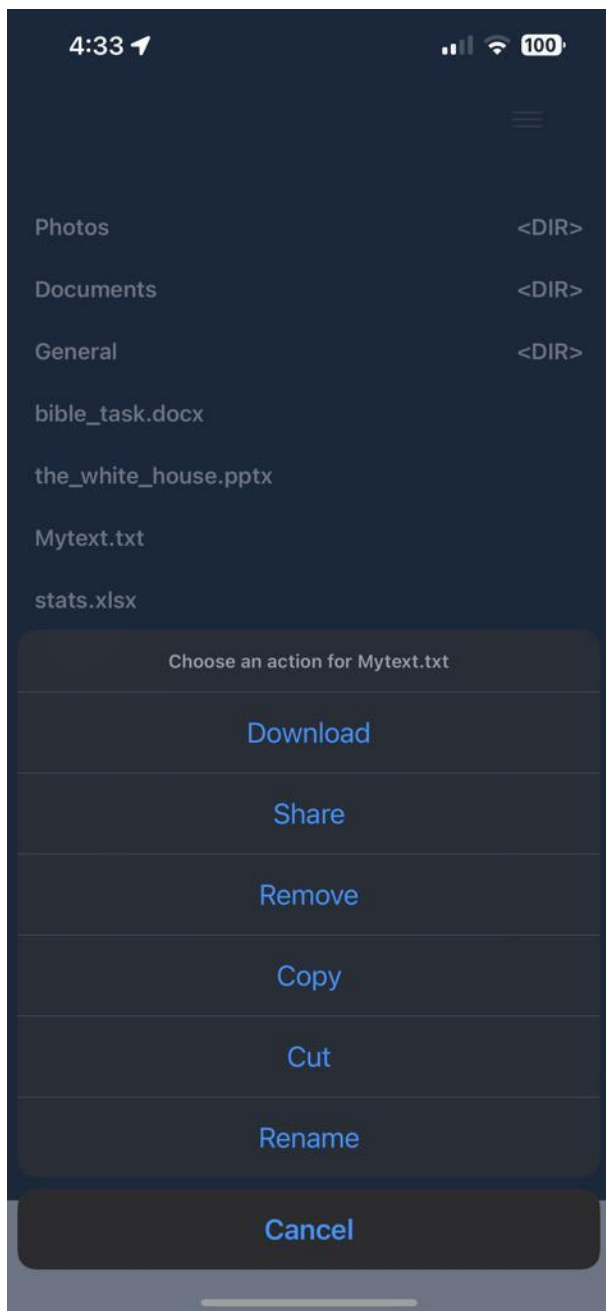
לאור השימוש הגובר במכשירים ניידים פיתחתי אפליקציה עבור מערכת הפעלה IOS וכן אפשרות גישה לקבצים ממחשבים. בפרק זה אציג את המסכים אותם פיתחתי לטובת הפרויקט

3.7.1 מסכי משתמש קצה בטלפון IOS

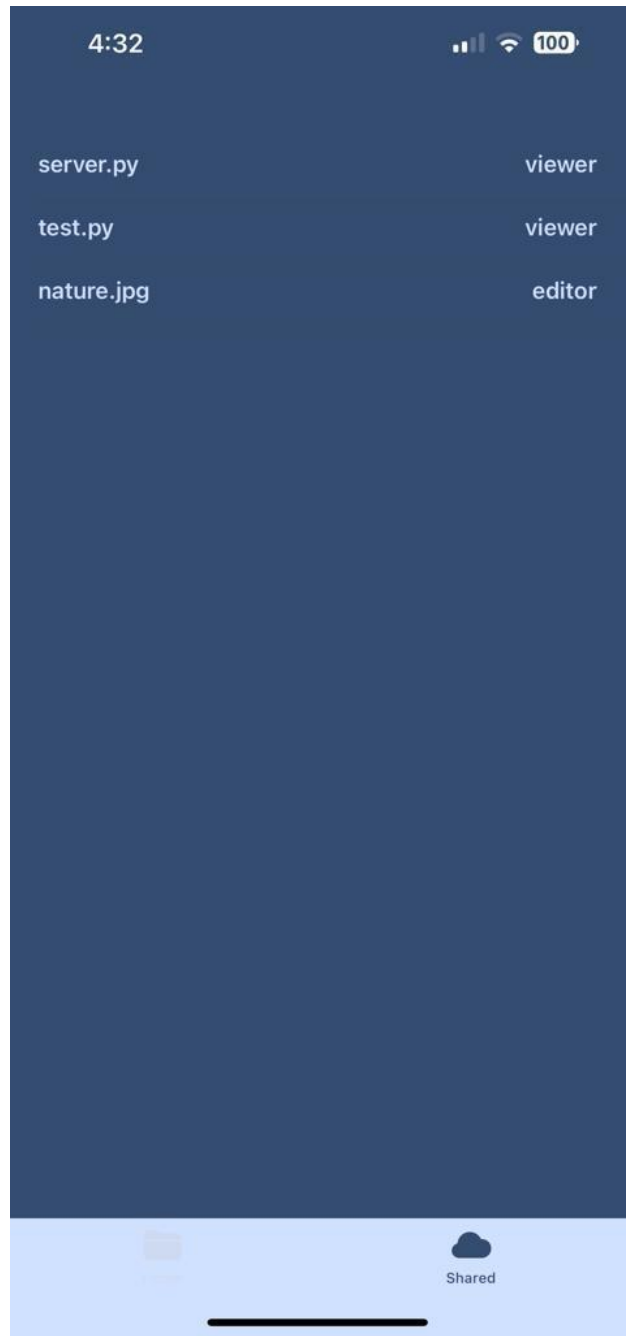
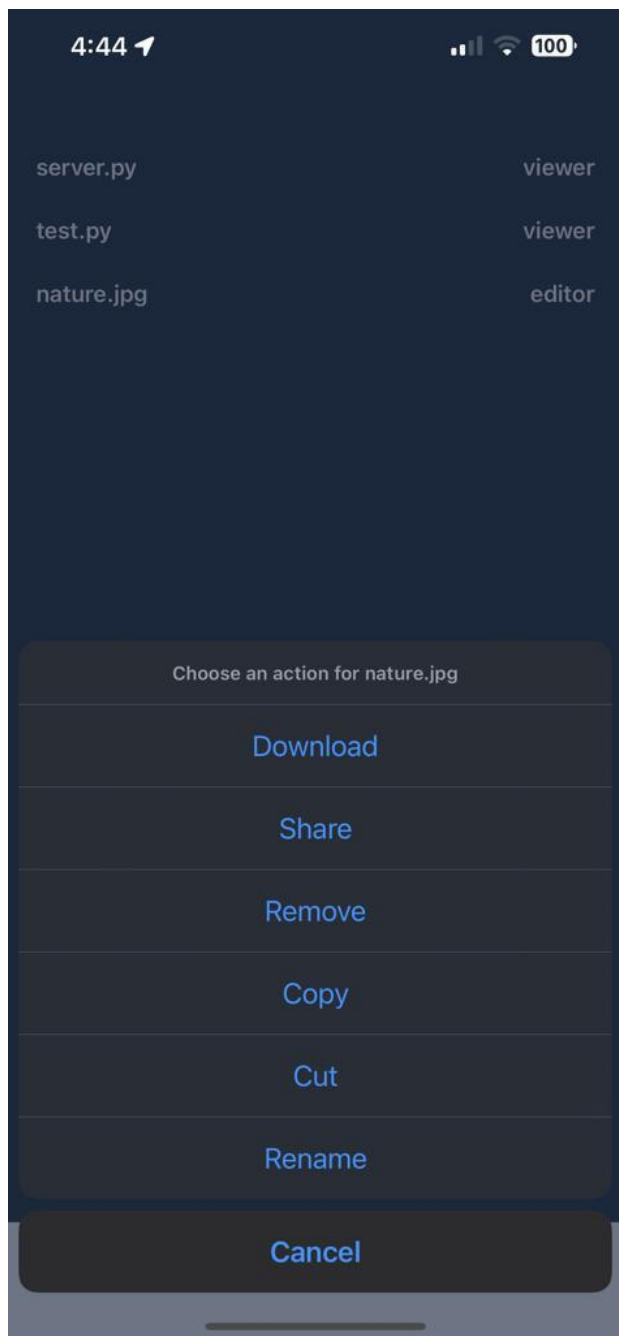
מסך Login & Sign up ואיקון האפליקציה



מסך קבצים אישיים

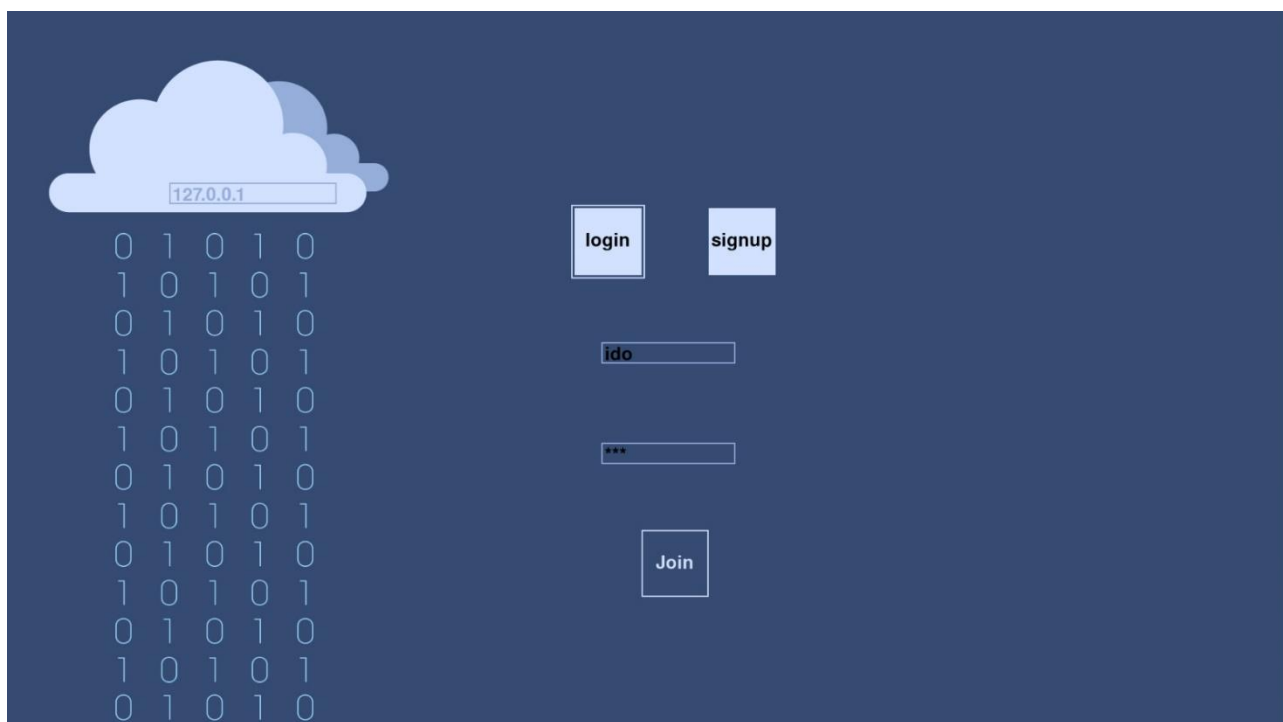


מסך קבצים משותפים

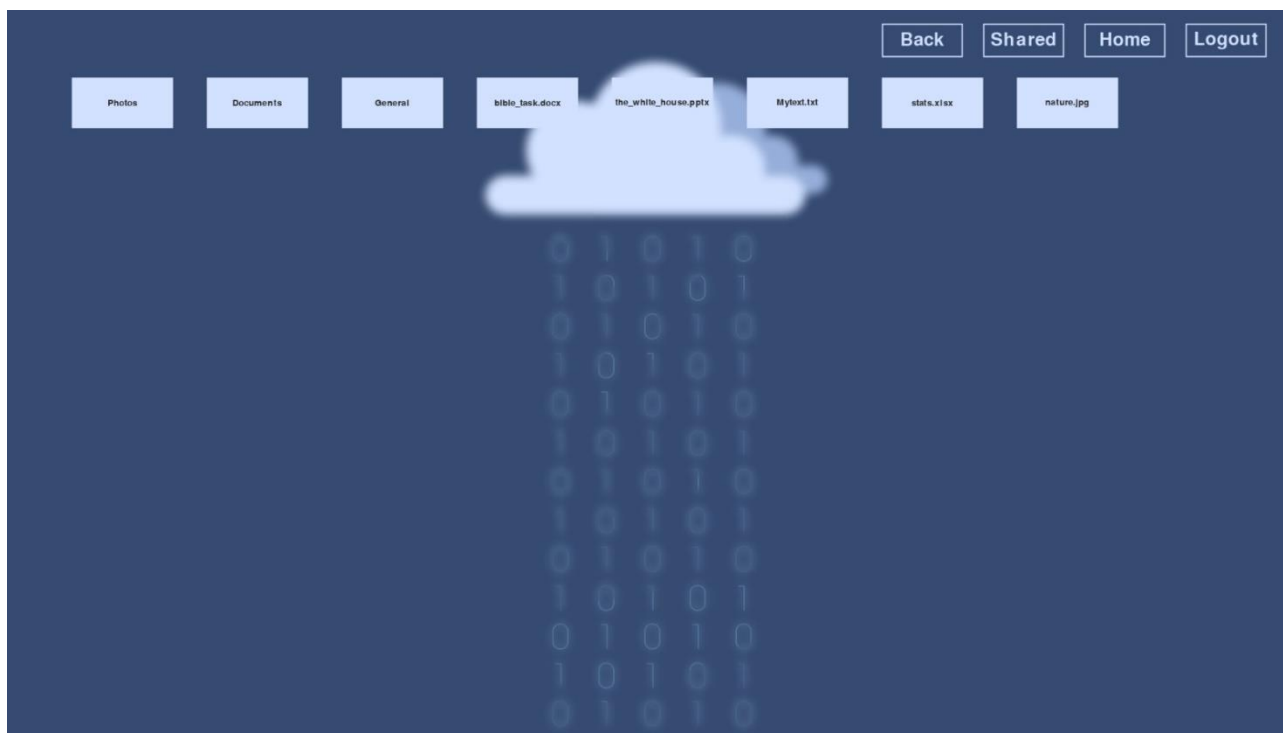


3.7.2 מסכי משתמש קצה במחשב אישי WINDOWS

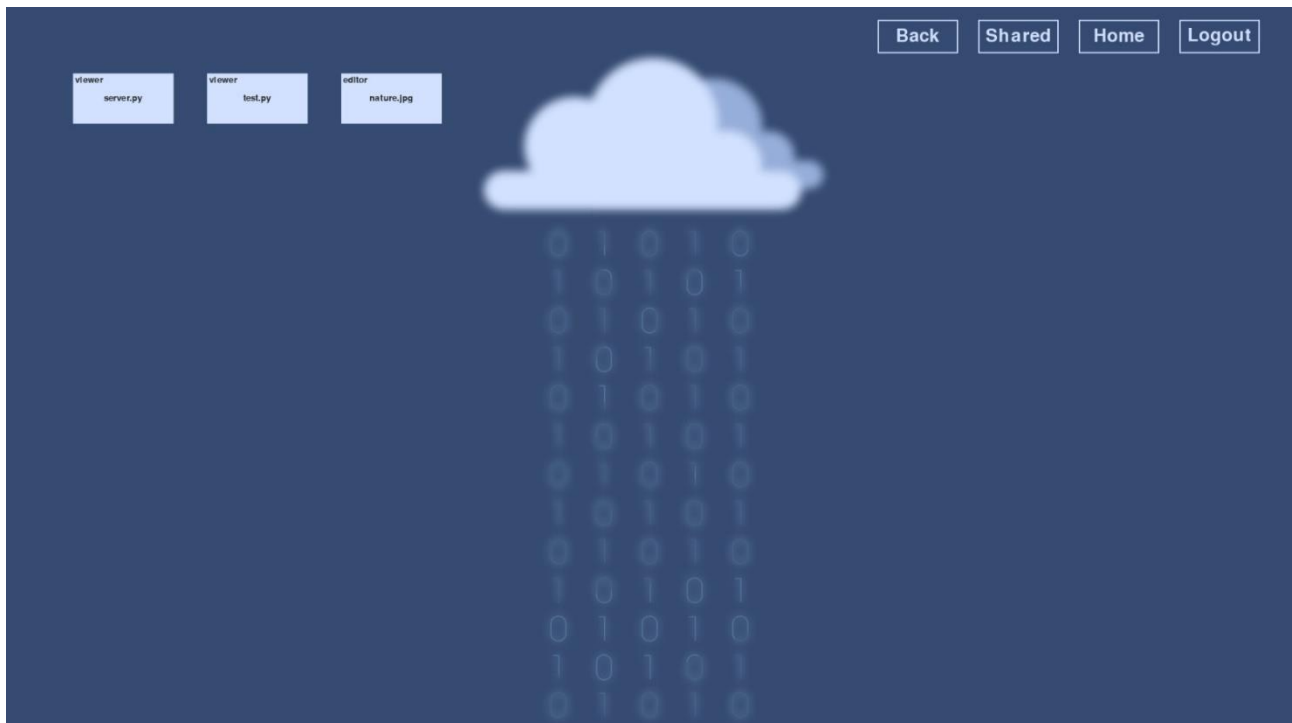
מסך Login & Sign up



מסך קבצים אישיים



מסך קבצים משותפים



3.8 תיאור מבנה הנתונים

בפרויקט יש מספר מבני נתונים. בהם נתוני המשתמשים, הקבצים המשותפים וכן הקבצים והתיקיות עצמם.

את נתוני המשתמשים אני שומר במילון users כאשר המפתח הוא השם של המשתמש והערך הוא טיפוס מסוג user עליו אפרט בהמשך. כאשר מכבים את השרת אני שומר את המילון לpickle כדי שיהיה אפשר להעלות את המשתמשים בפעם הבאה שמדליקים את השרת.

את הקבצים המשותפים אני שומר בשתי טבלאות sql

טבלה אחת שומרת את uuid של הקבצים ואת path שלהם כאשר id הוא המפתח הראשי. טבלה אחרת שומרת את המשתמשים ואיזה קבצים משותפים אליהם ובאיזו הרשאה. כל השדות הם מטיפוס טקסט

id	path
5b0e4182-247a-481b-a3b3-b9644e6664f5	E: \\.\USB\Tomer\school\Networking\MicroStore\server\users\ido\idos_dir\hello2.txt

user	id	perm
tomor	5b0e4182-247a-481b-a3b3-b9644e6664f5	editor
tomor	dddf50ea-34bb-4177-8a1e-1e1fd1c63c1f	viewer

את הקבצים והתיקיות אני שומר עבור כל משתמש בתיקיה משל עצמו כאשר הוא לא יכול לצאת מהתיקיה שלו אלא רק ליצור תתי תיקיות ולהיכנס אליהן.

3.9 סקירת חולשות ואיומים

חלק מרכזי בפרויקט הוא הגנת סייבר. בכדי להגן על הפרוייקט מימשתי מספר אמצעים:

- הצפנה היברידית באמצעות דיפי הלמן aes
 - הגנה מפני "טיול" בקבצים ויציאה מהתיקיה של כל משתמש על ידי הגבלת המשתמש משליחה של path עם ..
 - התחברות באמצעות שם משתמש וסיסמא
 - שמירת סיסמאות באמצעות salted hash
- באמצעות הגנות אלו אני מונע התקפות של גניבת מידע והתקפות MITM. יחד עם זאת המערכת עדיין חשופה לאיומים כמו ddos ו sql injection ניתן לתקן את פרצות אלו בעדכוני גרסה.

4. מימוש הפרויקט

4.1 סקירת כל המודלים \ מחלקות מיובאות של המערכת

4.1.1 מרכיבי המערכת בשרת

Socket – חבילה זו אחראית על ניהול התקשורת

Threading – חבילה זו מאפשרת לנהל מספר תהליכים במקביל וכך מספר לקוחות במקביל

Pickle – חבילה זו מאפשרת שמירה של נתוני המשתמשים לקובץ והעלתם לאחר הדלקת השרת

random, string – חבילה זו אחראית לייצור salt של הסיסמא וייצור dh

Hashlib – חבילה זו משמשת לייצור hash עבור שמירת הסיסמאות והצפנת התקשורת

Crypto – חבילה זו משמשת להצפנת התקשורת

os, shutil – חבילה זו אחראית על ניהול הקבצים והתיקיות

Json – חבילה זו מאפשרת שליחה של מבני נתונים

Sqlite3 – חבילה זו אחראית על כל ממשק הsql

Uuid – חבילה זו מייצרת id בלעדי עבור הקבצים

4.1.2 מרכיבי המערכת ביחידת משתמש קצה מסוג מחשב אישי

Socket – חבילה זו אחראית על ניהול התקשורת

Threading – חבילה זו מאפשרת לנהל מספר תהליכים במקביל וכך לאפשר תקשורת וממשק בויז

random - - חבילה זו אחראית לייצור dh

Crypto, hashlib – חבילה זו אחראית על הצפנת התקשורת

os – חבילה זו אחראית על ניהול path-like strings

Json – חבילה זו מאפשרת שליחה של מבני נתונים

Pygame, tkinter, time – חבילות אלו אחראיות על הממשק הגרפי

4.1.3 מרכיבי המערכת ביחידת משתמש קצה מסוג טלפון IOS

UIKit - חבילה זו אחראית על כל ממשק המשתמש

Foundation – חבילה זו מכילה דברים בסיסיים כמו טיפוס Data

Darwin – חבילה זו מאפשרת גישה לרכיבים במערכת ההפעלה ביניהם socket

CryptoSwift, CommonCrypto – חבילות אלו משמשות להצפנת התקשורת

UniformTypeIdentifiers – חבילה זו מאפשרת קריאה של קבצים מקומיים על מנת להעלותם

4.2 סקירת כל המודלים \ מחלקות שאותם פיתחתי של המערכת

4.2.1 מרכיבי המערכת בשרת

מחלקה	תכונה/פעולה	תיאור
User	username	שם המשתמש
User	password	סיסמת המשתמש אחרי שעברה salted hash
User	salt	רצף התווים עבור salted hash
User	path	מיקום תיקיית המשתמש
User	curr_dir	התיקיה הנוכחית בה המשתמש נמצא
User	connected	האם המשתמש מחובר או לא
File	owner	הבעלים המקוריים של הקובץ
File	path	ה path של הקובץ
File	perm	רמת ההרשאה לקובץ
AESCipher	key	מפתח ההצפנה
AESCipher	encrypt	הפעולה מקבלת מידע ומחזירה אותו מוצפן
AESCipher	decrypt	הפעולה מקבלת מידע מוצפן ומחזירה את המידע המפוענח
UserFileORM	create	הפעולה יוצרת את הטבלאות db
UserFileORM	get_all	הפעולה מקבלת user ומחזירה את כל הקבצים המשותפים אליו
UserFileORM	add_file	הפעולה מקבלת משתמש, קובץ והרשאה ומשתפת את המשתמש בקובץ הזה
UserFileORM	Delete_file	הפעולה מקבלת קובץ ומוחקת את כל המקרים של הקובץ בטבלאות

הפעולה מקבלת שם חדש ושם ישן ומחליפה את השם הישן של הקובץ בשם החדש	change_name	UserFileORM
הפעולה מקבלת id ומחזירה את הקובץ המתאים	get_path_by_id	UserFileORM
הפעולה מקבלת id וuser ומחזירה את ההרשאה שיש למשתמש לקובץ עם id	get_perm	UserFileORM

4.2.2 מרכיבי המערכת ביחידת משתמש קצה מסוג מחשב אישי

מפתח ההצפנה	key	AESCipher
הפעולה מקבלת מידע ומחזירה אותו מוצפן	encrypt	AESCipher
הפעולה מקבלת מידע מוצפן ומחזירה את המידע המפוענח	decrypt	AESCipher
הצבע של הכפתור	color	Button
הצבע של הטקסט	text_color	Button
הטקסט עצמו	text	Button
מלבן של pygame המייצג את הכפתור	rect	Button
האם העכבר נמצא על הכפתור	hover	Button
האם הכפתור נלחץ	clicked	Button
מצייר את הכפתור למסך	Draw	Button
מעדכן את הכפתור במידה והעכבר נמצא עליו	Isover	Button
מעדכן את הכפתור במידה ונלחץ	Click	Button
מקבל טקסט ומדפיס את הטקסט שהתקבל על הכפתור	Draw_text	Button
מקבל x, y ומעדכן את מיקום הכפתור	update	Button

מחלקות אלה יורשות ממחלקת Button אך בעלות תכונות של path ו name		File_But, Dir_But
המחלקה יורשת מ File_Path אך בעלת תכונות של UUID, perm כמו כן כאשר מציירים את הכפתור מדפיסים גם את ההרשאה		Shared_File_But
הפונט של הטקסט	FONT	InputBox
המלבן של הטקסט בוקס pygame()	Rect	InputBox
צבע הטקסט בוקס	Color	InputBox
המחרוזת של הטקסט עצמו	Text	InputBox
הטקסט להדפסה	Txt_surface	InputBox
האם הטקסט בוקס פעיל	active	InputBox
יורשת מ InputBox אך מדפיסה טקסט מצונזר		PassBox

4.2.3 מרכיבי המערכת ביחידת משתמש קצה מסוג טלפון IOS

מפתח ההצפנה	key	AESCipher
הפעולה מקבלת מידע ומחזירה אותו מוצפן	encrypt	AESCipher
הפעולה מקבלת מידע מוצפן ומחזירה את המידע המפוענח	decrypt	AESCipher
המקרה היחיד של המחלקה (סינגלטון)	shared	SocketManager
File descriptor של הסוקט דרכו ניגשים אליה	socketFD	SocketManager
הפעולה מקבלת ip, port ויוצרת חיבור	Connect	SocketManager
הפעולה מקבלת מידע ושולחת אותו דרך הסוקט	Send	SocketManager

SocketManager	Recv	הפעולה מחזירה מידע המתקבל מהסוקט
SocketManager	close	הפעולה סוגרת את החיבור
SecureSocketManager	shared	המקרה היחיד של המחלקה (סינגלטון)
SecureSocketManager	Cipher	מצפין של מידע
SecureSocketManager	socket	סוקט מהמחלקה socketmanager
SecureSocketManager	Connect	הפעולה יוצרת את חיבור עם סוקט בשרת ומחליפה מפתח באמצעות DH
SecureSocketManager	Send	הפעולה מצפינה ושולחת את המידע
SecureSocketManager	Recv	הפעולה מחזירה מידע מפוענח שהתקבל
SecureSocketManager	close	הפעולה סוגרת את החיבור
SecureSocketManager	dh	הפעולה אחראית על ההחלפה עצמה של המפתח מול השרת
SecureSocketManager	Modpow	הפעולה מוצאת את החזקה המודולרית של שלושה מספרים
SecureSocketManager		למחלקה יש "הרחבה" המכילה את כל הפעולות שהן חלק מהפרוטוקול שאחראיות על שכבת האפליקציה
ExplorerViewController SharedViewController		מחלקות אלה אחראיות על הממשק הגרפי של האפליקציה ומשמשות כמעין שכבת ביניים בין בקשות המשתמש לביצוען

4.3 תיאור מפורט של האלגוריתמים "מעניינים" בפרויקט

4.3.1 פיתוח ומימוש פרוטוקול תקשורת שרת-לקוח

החלק המרכזי בפרויקט הוא התקשורת. בכדי לייצר את התקשורת הייתי צריך לתכנן את הפרוטוקול של התקשורת (ניתן לראות את הפרוטוקול בפרק פרוטוקול התקשורת). במהלך השנתיים האחרונות יצא לי הרבה ליצור הרבה אפליקציות שרת-לקוח בפייתון המבוססות על סוקט. אך כשהתחלתי לפתח בסוויפט הבנתי שהדבר שונה מפייתון ומסובך יותר. הייתי צריך לחפש חבילות שיעשו את העבודה של סוקט אך הדבר לא היה פשוט כל כך מכיוון שרוב החבילות מיועדות לשכבות גבוהות יותר של תקשורת לדוגמא http. לכן הייתי צריך לפתח את המחלקה socket manager שמזכירה את socket בפייתון. בתוך המחלקה כבר שילבתי את האלגוריתם של tcp_by_size אותו למדנו בתחילת כיתה יא. האלגוריתם מאפשר שליחה של מספר פקטות על מנת לשלוח רצפים גדולים של מידע.

4.3.2 מימוש הצפנה היברידית באמצעות אלגוריתם דיפי-הלמן ו AES

חלק מרכזי נוסף בפרויקט הוא הגנת סייבר. על מנת להגן על הפרויקט מפני התקפות MITM החלטתי להצפין את התקשורת. את הצפנת התקשורת אני עושה באמצעות הצפנה היברידית כאשר את החלפת המפתח אני עושה באמצעות דיפי הלמן ואת הצפנת המידע אני עושה באמצעות AES. בכדי לממש את האלגוריתם של דיפי הלמן קראתי באינטרנט על האלגוריתם וכתבתי אותו בפייתון ובסוויפט. כאשר מימשתי אותו בסוויפט נתקלתי בבעיה, כאשר רוצים לעשות שארית למספר גדול מאוד הדבר לוקח זמן רב. בפייתון יש פעולה built-in של חזקה מודולרית אך בסוויפט אין לכן הייתי צריך לממש את הפעולה הזאת גם כן.

4.3.3 מימוש אלגוריתם לניהול הקבצים והרשאות המשתמשים השונים

בכדי לאפשר את שיתוף הקבצים השתמשתי בsql. יש לי שתי טבלאות אחת שומרת את הpath של הקובץ id ואת השניה שומרת לכל יוזר את כל הקבצים המשותפים אליו ובאיזו הרשאה. כך כאשר משתמש מבקש את כל הקבצים המשותפים אליו אני מבקש מהטבלה את id של כל הקבצים שלו בצורה כזאת לא עובדים ישירות עם מסלולים וכך מגנים על המידע מפני התקפות ומסתירים את התיקיה הספציפית בה הוא נמצא.

4.3.4 מימוש מערך נתונים להקמת משתמש חדש, יצירת סיסמה עבורו והכנסתו למערכת

כאשר משתמש יוצר משתמש חדש הוא מתבקש להזין שם משתמש חדש וסיסמא. במידה והמשתמש כבר קיים הוא מקבל הודעה על כך. כאשר שומרים את הסיסמא מערבלים אותה באמצעות salted hash. כך גם אם משהו משיג גישה למאגר המשתמשים הוא אינו יכול לפענח את הסיסמאות של כל המשתמשים אפילו אם הוא משתמש ב rainbow table מכיוון שלסיסמא התווסף רצף של תווים ששינה לחלוטין את הערבול ולכן היא לא נמצאת בשום מאגר של סיסמאות נפוצות.

4.4 מערך\מסמך הבדיקות של הפרויקט

פרק זה מתאר את הבדיקות הפונקציונליות שיש לבצע לאחר סיום הקידוד ולפני שיחרורו האפליקציה ללקוח. תנאי מקדים לכל פעולה היא מערכת רצה המכילה: שרת, יחידת קצה מסוג מחשב אישי, יחידת קצה מסוג טלפון מבוסס IOS.

כל בדיקה אמורה להתבצע כ-10 פעמים, ובצורה אקראית (כלומר לא לחזור על הבדיקה באופן חזרתי אחד אחרי השני, אלא לבצע "שזירה" של הבדיקות אחת בשניה)

4.4.1 בדיקת יצירת חשבון ע"י שם משתמש וסיסמה

בדיקת יחידת קצה מחשב אישי

- הזן שם משתמש חדש וסיסמה חדשה דרך כפתור Sign Up
- צא מהחשבון שהקמת
- כנס לחשבון מחדש דרך המחשב וצא
- כנס לחשבון דרך מכשיר הטלפון מבוסס IOS
- צא מהחשבון

בדיקת יחידת קצה טלפון מבוסס IOS

- הזן שם משתמש חדש וסיסמה חדשה דרך כפתור Sign Up
- צא מהחשבון שהקמת
- כנס לחשבון מחדש דרך המחשב וצא
- כנס לחשבון דרך מכשיר הטלפון מבוסס IOS
- צא מהחשבון

4.4.2 בדיקת תקשורת שרת-לקוח מוצפנת

בדיקת יחידת קצה מחשב אישי

- הזן שם משתמש חדש וסיסמה חדשה דרך כפתור Sign Up

בדיקת יחידת קצה טלפון מבוסס IOS

- הזן שם משתמש חדש וסיסמה חדשה דרך כפתור Sign Up

4.4.3 בדיקת גישה מרחוק לקבצים מכל מחשב windows או iPhone

בדיקת יחידת קצה מחשב אישי

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור Log In
- העלה קובץ לשרת מהמחשב האישי
- גש לקובץ מהטלפון ובדוק שהוא קיים

בדיקת יחידת קצה טלפון מבוסס IOS

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור Log In

- העלה קובץ לשרת מהטלפון האישי
- גש לקובץ מהמחשב ובדוק שהוא קיים

4.4.4 בדיקת יצירת תיקיות

בדיקת יחידת קצה מחשב אישי

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור ה Log In
- צור סיפריה בשרת
- גש לסיפריה מהטלפון ובדוק שהיא קיימת

בדיקת יחידת טלפון מבוסס IOS

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור ה Log In
- צור סיפריה בשרת
- גש לסיפריה מהמחשב האישי ובדוק שהיא קיימת

4.4.5 בדיקת העלאת קבצים

בדיקת יחידת קצה מחשב אישי

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור ה Log In
- העלה קובץ לשרת מהמחשב האישי
- גש לקובץ מהטלפון ובדוק אם הוא קיים בשרת

בדיקת יחידת טלפון מבוסס IOS

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור ה Log In
- העלה קובץ לשרת מהטלפון
- גש לקובץ מהמחשב ובדוק אם הוא קיים בשרת

4.4.6 בדיקת הורדת קבצים

בדיקת יחידת קצה מחשב אישי

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור ה Log In
- העלה קובץ לשרת מהמחשב האישי
- גש לקובץ מהטלפון והורד אותו מקומית
- ודא שהקובץ שירד תקין וזהה למקורי

בדיקת יחידת טלפון מבוסס IOS

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור ה Log In
- העלה קובץ לשרת מהטלפון
- גש לקובץ מהמחשב והורד אותו מקומית
- ודא שהקובץ שירד תקין וזהה למקורי

4.4.7 בדיקת העברת קבצים בין תיקיות

בדיקת יחידת קצה מחשב אישי

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור ה Log In
- העלה קובץ לשרת מהמחשב לסיפריה A וצור במקביל סיפריה B
- גש לקובץ מהטלפון והעתק את הקובץ מסיפריה A לסיפריה B
- הורד את הקובץ בסיפריה B למחשב ובדוק תקינותו
- ודא שהקובץ לא קיים בסיפריה A

בדיקת יחידת קצה טלפון מבוסס IOS

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור ה Log In
- העלה קובץ לשרת מהטלפון לסיפריה A וצור במקביל סיפריה B
- גש לקובץ מהמחשב והעתק את הקובץ מסיפריה A לסיפריה B
- הורד את הקובץ בסיפריה B לטלפון ובדוק תקינותו
- ודא שהקובץ לא קיים בסיפריה A

4.4.8 בדיקת מחיקת תיקיות וקבצים

בדיקת יחידת קצה מחשב אישי

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור ה Log In
- העלה קובץ לשרת מהמחשב
- גש לקובץ מהטלפון ומחוק אותו
- ודא שהקובץ לא קיים בשרת דרך המחשב

בדיקת יחידת קצה טלפון מבוסס IOS

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור ה Log In
- העלה קובץ לשרת מהטלפון
- גש לקובץ מהמחשב ומחוק אותו
- ודא שהקובץ לא קיים בשרת דרך הטלפון

4.4.9 בדיקת העתקת קבצים

בדיקת יחידת קצה מחשב אישי

- הזן שם משתמש וסיסמה של חשבון קיים דרך כפתור ה Log In
- העלה קובץ לשרת מהמחשב לסיפריה A וצור במקביל סיפריה B
- גש לקובץ מהטלפון והעתק את הקובץ מסיפריה A לסיפריה B
- הורד את הקובץ בסיפריה B למחשב ובדוק תקינותו
- ודא שהקובץ קיים בסיפריה A

בדיקת יחידת קצה טלפון מבוסס IOS

- הזן שם משתמש עוסיסמה של חשבון קיים דרך כפתור ה Log In
 - העלה קובץ לשרת מהטלפון לסיפריה A וצור במקביל סיפריה B
 - גש לקובץ מהמחשב והעתק את הקובץ מסיפריה A לסיפריה B
 - הורד את הקובץ בסיפריה B לטלפון ובדוק תקינותו
 - ודא שהקובץ קיים בסיפריה A
- 4.4.10 בדיקת אפשרות לכבות ולהדליק את השרת תוך שמירה על כלל המידע של המשתמשים

5. מדריך למשתמש

פרק זה מכיל מדריך למשתמש וכולל את ההסברים הבאים:

- התקנת המערכת – שרת, יחידת קצה מחשב אישי מבוסס מערכת הפעלה WINDOWS, יחידת קצה טלפון מבוסס מערכת הפעלה IOS.
- אופן השימוש במערכת

5.1 עץ הקבצים

Server:

classes.py

file_manage_server.py

server.py

sql_orm.py

tcp_by_size.py

users.pickle

userfile.db

users - בתיקה זו שמורים הקבצים והתיקיות של המשתמשים.

ComputerApp:

asests - בתיקה זו שמורים התמונות עבור ממשק המשתמש

client.py

file_manage_client.py

glbl.py

gui.py

gui_classes.py

tcp_by_size.py

MicroStore(ios app):
AppDelegate.swift
FileTableViewCell.swift
LoginViewController.swift
SceneDelegate.swift
ViewController.swift
ExplorerViewController.swift
Info.plist
SharedViewController.swift
SocketManager.swift

*ישנם עוד תיקיות וקבצים האחראים על הגדרת הסביבה

5.2 התקנת המערכת

השרת צריך לרוץ על מערכת הפעלה windows
הלקוח צריך לרוץ על מערכת הפעלה windows או iOS
כדי להשתמש במערכת צריך להיות מותקן python 3.10 וצריך להיות מותקנות הספריות
Pycryptodome, pygame. הבאות.
כמו כן אם רוצים לגשת לשרת מחוץ לnat יש לעשות port forward בראוטר.

5.3 הפעלת המערכת

על מנת להתחיל את האפליקציה יש להפעיל את השרת ולאחר מכן להפעיל את הלקוחות.
לאחר שמפעילים את הלקוח יש להזין את האיי פי של השרת ולבחור האם ליצור משתמש חדש
או להתחבר לקיים, לאחר מכן להזין שם משתמש וסיסמא ולבסוף ללחוץ על כפתור
ההתחברות.

כעת ניתן למחוק, להוריד, לשנות שם, להעתיק, לשתף או להעביר מקום של קבצים ותיקיות וכן
להיכנס לתיקיות על ידי לחיצה עליהם. כמו כן ניתן להדביק, להעלות קובץ או ליצור תיקיה
חדשה יש ללחוץ על הכפתור הימני בעכבר או על כפתור העוד בטלפון.

6. סיכום אישי

את פרוייקט MicroStore התחלתי לבצע החל מאמצע שנת 2022 ועד למאי 2023. המטרה העיקרית של הפרויקט שלי הייתה ליצור שרת שמספק פתרון אחסון מאובטח, כאשר כל התקשורת מוצפנת וניתן יהיה לגשת לשרת דרך אפליקציה על המחשב או על הטלפון. ההשראה לפרויקט הגיעה מdropbox אך שמתי דגש על שמירה על הפרטיות על ידי כך שהמידע לא נמצא אצל החברות הגדולות אלא אצל הלקוח.

במהלך כל הפרויקט עבדתי בו זמנית על חלקים שונים של הפרויקט, אך היו מספר חלקים מרכזיים בהם ניהול הקבצים בצד השרת, שיתוף הקבצים, אפליקציה לIOS ואפליקציה למחשב.

אחד האתגרים המרכזיים שעמדו בפניי היו העדר הניסיון בפיתוח אפליקציות. הייתי צריך ללמוד מאפס את שפת swift ולהשתמש בסביבת storyboard וxcode. כדי לעשות זאת נעזרתי בעיקר באינטרנט. חיפשתי ביוטיוב ובאינטרנט על הבסיס של swift וגם על מקרים ספציפיים לדוגמה פיתוח של טבלה כמו שעשיתי עבור הקבצים. במהלך הפרויקט למדתי הרבה בפיתוח בswift בsql ובעבודה עם קבצים. כמו כן שיפרתי את יכולותיי בgit.

דבר נוסף שאני אקח מהפרויקט הזה הוא היכולת ללימוד עצמי, אמנם במהלך השנים שלי במגמה לימדו אותנו ללמוד לבד אך בפרויקט זה הרגשתי את זה יותר מבכל שלב אחר מכיוון שעסקתי במשהו שבכלל לא נגענו בו במגמה.

בהסתכלות לאחור הייתי מפתח את הלקוח של המחשב באופן שונה. הייתי משתמש בספריה שתתאים יותר לצרכים של הפרויקט מאשר pygame.

אם היו לי משאבים נוספים הייתי מוסיף שיתוף תיקיות ואפשרות לעריכת קבצים בתוך הפרויקט עצמו שמאפשר לכמה משתמשים לערוך את אותו קובץ בו זמנית כמו בgoogle drive. כמו כן הייתי מוסיף עוד הגנות כמו הגנה מפני ddos וsql injection.

אני רוצה להודות למורי בשנתיים האחרונות יוסי ואופיר על ההשקעה והיחס האישי שהם נתנו לתלמידים. כמו כן אני רוצה להודות לחבריי לכיתה.

7. ביבליוגרפיה

GeeksforGeeks. (2023b). Socket Programming in Python. *GeeksforGeeks*.

<https://www.geeksforgeeks.org/socket-programming-python/>

SQL Tutorial. (n.d.-b). <https://www.w3schools.com/sql/>

UIKit | Apple Developer Documentation. (n.d.-b). Apple Developer Documentation.

<https://developer.apple.com/documentation/uikit>

Paul Hudson @twostraws. (n.d.). *Hacking with Swift – learn to code iPhone and iPad apps with free Swift tutorials*. Hacking With Swift. <https://www.hackingwithswift.com/>

Careerera. (2021b, June 22). History of Cloud Computing. *Careerera.com*.

<https://www.careerera.com/blog/history-of-cloud-computing>

Newest “python” Questions. (n.d.). Stack Overflow.

<https://stackoverflow.com/questions/tagged/python>