**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)**

# Phoenix U.E.S

**פרק א' - 'שם המערכת' - ייזום**

## תיאור ראשוני של המערכת

המערכת היא מערכת להצפנת דיסק־און־קי ( DOK).   
ה דיסק־און־קי – אינו מאובטח וברגע שהוא נופל לידיים זרות, כל המידע שבו גלוי לעיני כל.  
המערכת שלי תדאג לאבטחת כלל המידע ב דיסק־און־קי בצורה שתאפשר אך ורק לבעל ה דיסק־און־קי לפענח אותו. במקרה שה דיסק־און־קי יפול לידיים זרות והמערכת תזהה זאת, היא תדאג להשמיד את כל תוכנו.  
במקביל לעבודה עם ה דיסק־און־קי הוא יגובה לשרת מרוחק, מה שיאפשר לבעל ה דיסק־און־קי לשחזר מידע לאחר אובדן ה דיסק־און־קי ו/או לאחר השמדת ה דיסק־און־קי.

**המוצר המוגמר:**

מערכת דיסק־און־קי מאובטחת הכוללת הצפנה מלאה של כל המידע ב דיסק־און־קי , בדיקת משתמשים, שמירה עותק של כל הקבצים בשרת מרוחק למען שיחזור, מחיקה מלאה של הדיסק־און־קי בכישלון של כניסה אליו, אפשרות שיחזור דרך השרת, הפעלת ההצפנה בכל פעם מחדש שה דיסק־און־קי מופעל, אחרי הכנסה של הסיסמא ושם משתמש נכונים להחזיר את כל הקבצים למצב הראשוני שלהם.

**אז למה בחרתי בפרוייקט הזה?**

בחרתי בפרוייקט הזה כי אני מאוד מתעניין באבטחת מידע ואיך כל זה פועל גם שהשרת אינו מחובר ועובד כל הזמן האתגר בפרוייקט עניין אותי. בנוסף לזה כל שהותי במגמה הייתי עם דיסק־און־קי והוא באמת אף פעם לא היה מוצפן וכל בן אדם היה יכול לגשת לי לפרטים ולדברים שלי מבלי שום בעיה והמחשבה הזאת מטרידה אותי ומונעת ממני פרטיות והפרוייקט הזה יעזור לי לפרטיות שלי ולהגנה שלי.

**אתגרים שאני צופה להם בפרוייקט:**

יצירת קובץ מערכת פנימי שמסוגל לאמת גישה גם ללא חיבור לשרת, הצפנה ברמת קובץ בודד ובמקביל גם ברמת כלל דיסק־און־קי, יצירת גיבוי מאובטח לכל הקבצים המוצפנים, התאמה בין השרת לדיסקו דיסק־און־קי נקי כך שהמערכת לא תישבר גם אם המשתמש מעתיק או מזיז קבצים.

## הגדרת הלקוח

הלקוח הוא כל בן אדם אשר נחזיק בבעלותו דיסק־און־קי ורוצה שהוא יהיה מאובטח ולא יוכלו לפרוץ לו פנימה + שמירה על הקבצים במקום אחר שיוכל לשחזר אותם במקרה הצורך.

## למי מיועדת המערכת ומי הולך להשתמש בה?

המערכת מיועדת לכל אדם שמחזיק דיסק־און־קי – תלמידים, אנשי מחשבים, בעלי עסקים, עובדים שמחזיקים קבצים מסווגים, ואפילו אנשים פרטיים שרוצים לשמור תמונות/מסמכים חשובים.

### מטרות לכל לקוח:

**אבטחת מידע מלאה:** הצפנת הקבצים כך שלא ייפתחו ללא הרשאות  
**גיבוי ה דיסק־און־קי**: שמירת כל המידע ב דיסק־און־קי במקום נוסף  
**נוחות שימוש:** כל ההצפנה מתבצעת אוטומטית בעת חיבור לשרת.  
**הגנת מידע מקסימלית:** מחיקה אוטומטית לאחר כישלון בהזדהות.  
**שחזור דרך השרת:** יכולת לשחזר מידע באמצעות הגיבוי המוצפן.

## בעיות, תועלות וחסכונות

### הבעיה

רוב ה דיסק־און־קי -ים אינם מוצפנים, ומי שיגנוב או ימצא אותם – יכול לקרוא את כל התוכן בקלות וגם אם איבדת את ה דיסק־און־קי איבדת את כל מה שהיה בתוכו

### מה אנחנו מנסים להשיג?

מניעת גישה לא מורשית.  
שמירה על פרטיות המשתמש.  
העלאת רמת האבטחה בלי שימוש בחומרה מיוחדת.  
גיבוי כלל המידע ב דיסק־און־קי

### תועלות צפויות:

רמת אבטחה גבוהה ל דיסק־און־קי.  
קושי משמעותי לפריצת ההצפנה.  
אמינות גבוהה בהגנה על נתונים. שמירה מוגנת על הנתונים.

### שירותים שהמערכת תיתן:

הצפנה  
אימות משתמש  
מחיקה אוטומטית למניעת גניבת מידע  
גיבוי מאובטח  
שיחזור קבצים שנמחקו\ נאבדו

## השוואת העבודה עם פתרונות קיימים:

### 1. VeraCrypt

VeraCrypt היא תוכנה חינמית וקוד פתוח להצפנת DOK מלאים או יצירת "מכולות" מוצפנות. המערכת משתמשת באלגוריתמי הצפנה כמו AES, Serpent ו‑Twofish, ולעיתים שילובים שלהם. הצפנת DOK מתבצעת באמצעות יצירת שכבת קידוד מעל הקבצים, והגישה אליהם אפשרית רק לאחר הזנת סיסמה נכונה.

VeraCrypt מבצעת הצפנה בזמן אמת (on‑the‑fly), כלומר כל קובץ שנפתח או נשמר עובר פענוח/הצפנה מיידית בלי שהמשתמש מרגיש בכך. כדי לפתוח את הכונן המוצפן נדרש להזין סיסמה או להשתמש בקובץ‑מפתח מיוחד. אם הסיסמה שגויה – פשוט לא ניתן לפתוח את הכונן, אך התוכנה **אינה מוחקת את הקבצים** לאחר ניסיונות כושלים, אלא רק חוסמת גישה עד להזנת סיסמה נכונה.

### 2. BitLocker (Windows)

כלי מובנה ב-Windows להצפנת התקנים. מספק הצפנה, אך אינו כולל מערכת ניסיונות כושלים ומחיקה.

שניהם נותנים הצפנה, אבל המערכת שלי מוסיפה מנגנון הגנה ייחודי – מחיקה מלאה לאחר כישלון + שיחזור הקבצים לבעל ה דיסק־און־קי בעזרת השרת.

## האם צפויים קשיים או מגבלות?

המנגנון חייב להיות אמין גם בלי חיבור לשרת.  
שמירה על כל הקבצים של כל המשתמשים במקום מאובטח ואמין. שיחזור הקבצים במקרה הצורך.

## האם מדובר בטכנולוגיה חדשה?

הצפנה עצמה אינה חדשה, אך בניית מנגנון מחיקה אוטומטי ומערכת אימות מקומית ללא שרת – זה יחסית מתקדם. גם שיחזור הקבצים אינם מתקדמים מאוד אך הפרוייקט יוצר שילוב מעניין בין השנים.

## מגבלות בהגדרת המערכת

ייתכן שמערכות הפעלה מסוימות יחסמו הפעלה אוטומטית מה דיסק־און־קי.  
ייתכן צורך בדרייברים או הרשאות מיוחדות.  
לטובת הגיבוי ה דיסק־און־קי ושרת הגיבוי צריכים להימצא על אותה הרשת (מגבלות בה"ס)

## 

## 

## 

## תיחום הפרויקט

### תחומים שבהם הפרויקט עוסק:

**רשתות:** תקשורת בין השרת ל דיסק־און־קי.  
**מערכות הפעלה:** גישה לקבצים, הרשאות מערכת, הצפנה.  
**אבטחת מידע:** הצפנה, אימות, מחיקה יזומה.

### תחומים שבהם הפרויקט לא מטפל:

לא מטפל בניהול משתמשים מתקדם.  
לא מגן מפני שיבוש פיזי או גניבה של החומרה עצמה.

**פרק ב' - 'שם המערכת' - אפיון**

1. פרוט המערכת:

המערכת היא מערכת הצפנה של דיסק און קי (DOK)

המערכת עצמה לוקחת DOK שהוא ל מאובטח בכלל, כל אחד יכול להיכנס לדיסק און קי ולקחת מה שהוא רוצה ממנו

המערכת שלי מצפינה ומגבה את המידע שקיים ב DOK, (שמירה על נתונים במקרה וצריך שחזור). המערכת נותנת הרשאת גישה ופיענוח של ההצפנה רק לבעל ה DOK מה שיוצר אבטחה והגנה מפני פורצים שרוצים את המידע.

כאשר אתה מתחבר ל- DOK הוא יבקש ממך סיסמא וקוד ואם אתה טועה שלוש פעמים ימחק לך כל המידע מה DOK ובעל ה DOK יקבל הודעה (מייל) האומרת לו שנמחק לו הכל ואם הוא רוצה לשחזר הוא צריך לפנות לשרת שישחזר לו הכל מכיוון שהכל מגובה אז קיימת האפשרות.

* + מה היכולות שהיא תעניק למשתמש:
* הצפנת DOK – הצפנה מלאה של ה DOK
* גיבוי – גיבוי הקבצים במקום מאובטח
* שמירה על המידע – לאחר שלושה ניסיונות כניסה לא מורשים ל DOK תהיה מחיקה מלאה של ה DOK
* שיחזור – אם אתה מגיע למצב שנמחק לך ה DOK תהיה יכולת לשחזר את ה DOK
* התרעות – במקרה של מחיקת ה DOK המשתמש מקבל התרעות מהמערכת ומוזמן לשחזור

1. פירוט הבדיקות ('קופסא שחורה')

| **מספר** | **שם הבדיקה (שם שיעיד על תוכן הבדיקה)** | **מה אמורה לבדוק** | **איך מתכננים לבדוק**  **(לתאר בפירוט את שלבי הבדיקה)** |
| --- | --- | --- | --- |
| 1. | DOK מוצפן | האם כל ה DOK מוצפן בכל פתיחה שלו | פותחים את ה DOK ומנסים להיכנס לקובץ ורואים אם הוא מוצפן ועושים זאת שוב אחרי שפתחת את ההצפנה ורואים שניתן לגשת לקובץ |
| 2. | שגיאת סיסמא | אם אחרי שלוש שגיאות ה DOK נמחק | מכניסים שלושה שגיאות ל DOK ורואים אם אחרי שלושה שגיאות נמחק ה DOK |
| 3. | התראת מחיקה | לבדוק שבעל ה DOK מקבל התראה כשה DOK נמחק | לוודא שבסיום הבדיקה הקודמת נשלח מייל לבעל ה DOK |
| 4. | שיחזור | אם אחרי מחיקה מצליח לשחזר את הכל | אחרי קבלת הודעה מהשרת לעשות את מה שהוא מבקש ממך ולראות שהכל חוזר לקדמותו |
| 5. | עבודה ללא שרת | שניתן לעבוד עם ה DOK גם כשהשרת לא מחובר | מנסים להיכנס ל DOK ומוודאים שאחרי הכנסת סיסמא נכונה הכל זמין |
| 6. | גיבוי שלם | שהקבצים שאיתם עבדנו כשלא היה שרת, ברגע שיש שרת מגובים | לעבוד ללא שרת על אחד הקבצים, לבצע בו שינוי. לסגור את המערכת. להפעיל את השרת, להפעיל את המערכת. בסיום לגשת לשרת ולראות שהקובץ מהפעם הקודמת גובה |

1. תכנון לוח זמנים לפרויקט

| **פעילות** | **זמן התחלה מתוכנן** | **זמן סיום מתוכנן** | **זמן התחלה בפועל** | **זמן סיום בפועל** | **הערות** |
| --- | --- | --- | --- | --- | --- |
| יזום | 13.11.25 | 16.11.25 | 15.11.25 | 16.11.25 |  |
| אפיון | 16.11.25 | 30.11.25 | 26.11.25 | 30.11.25 |  |
| ניתוח | 30.11.25 | 21.12.25 | 30.11.25 | 4.12.25 |  |
| עיצוב | 21.12.25 | 22.2.26 |  |  |  |
| הצפנה ופענוח של DOK (בלי קשר לשרת) | 22.2.26 | 24.2.26 |  |  |  |
| מימוש תקשורת לשרת | 24.2.26 | 26.2.26 |  |  |  |
| מימוש מערכת גיבויים | 26.2.26 | 28.2.26 |  |  |  |
| שרת יוצר משתמשים ומשתיל שרתים קטנים שלו באחרים | 28.2.26 | 1.3.26 |  |  |  |
| תפעול המיני שרתים | 1.3.26 | 4.3.26 |  |  |  |
| שמירת נתונים "בענן" | 4.6.26 | 6.3.26 |  |  |  |
| שיחזור DOK | 6.3.26 | 13.3.26 |  |  |  |
| פיצירים נוספים | 13.3.26 | 15.3.26 |  |  |  |
| גרסה ראשונית | 22.02.26 | 15.03.26 |  |  |  |
| מסמך בדיקות | 15.03.26 | 1.4.26 |  |  |  |
| מדריך למשתמש | 15.03.26 | 1.4.26 |  |  |  |
| גרסה סופית | 1.4.26 | 1.5.26 |  |  |  |
| סגירת תיק פרויקט | 1.4.26 | 1.5.26 |  |  |  |

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

1. ניהול סיכונים בפרויקט

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |  |  |
| הצפנה לא עובדת | DOK אינו מאובטח | קשה | * לחקור על הצפנות שונות * מעבר על ההצפנה ולראות איך אפשר לשדרג אותה |  |  |
| "שרת" שתול לא עובד | DOK אינו יפענח את עצמו + יצפין | קשה | * החלפת דרך ההתקנה של השרת * ניטור רציף כדי לזהות תקלות בזמן אמת |  |  |
| שיחזור קבצים | אין שמירת נתונים ל-DOK ויכול לאבד הכל | קשה | * שימוש בשיטת שיחזור שונה * בדיקה אוטומטית מקיפה שבודקת האם הכל שוחזר |  |  |
| שמירת משתמש + נתונים | אי סדר בשיחזור ואיבוד נתונים | קל | * שמירה במסמך צד שהשרת יצור * שמירה בתוך DOK של השרת |  |  |
| מחיקת הפרויקט | איבוד כל הנתונים וכל השרת | קל | * שמירה בDRIVE * שמירה בGITHUB * שמירה בתוך DOK של השרת |  |  |

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

**פרק ג' - 'שם המערכת' - מסמך ניתוח**

**יכולות בצד שרת:**

שם היכולת: הרשמה למערכת

מהות היכולת: רישום משתמש חדש במערכת

אוסף יכולות:

* קבלת נתונים מהלקוח
* פענוח
* בדיקה מול בסיס הנתונים
* גיבוב סיסמא
* הוספה לבסיס נתונים
* החזרת תשובה מוצפנת

רשימת אובייקטים: הצפנה/פענוח, תקשורת, בסיס נתונים

שם היכולת: התחברות למערכת

מהות היכולת: התחברות למערכת למשתמש קיים

אוסף יכולות:

* קבלת נתונים מהלקוח
* פענוח
* בדיקה מול בסיס הנתונים
* בדיקת בקשה
* החזרת תשובה מוצפנת

רשימת אובייקטים: הצפנה/פענוח, תקשורת, בסיס נתונים

שם היכולת: יצירת קבצי מערכת

מהות היכולת: יצירת קבצי מערכת על מנת לטעון ל DOK של המשתמש

אוסף יכולות:

* קבלת נתונים מהלקוח
* קבלת נתוני הלקוח מבסיס הנתונים
* יצירה של קבצי המערכת
* הצפנה
* שליחה

רשימת אובייקטים: בסיס נתונים, ניהול קבצים, הצפנה/פענוח, תקשורת,

שם היכולת: גיבויים

מהות היכולת: מגבה את הDOK על מנת מתן שיחזור

אוסף יכולות:

* קבלת נתונים מהלקוח
* פענוח שם הקובץ
* שמירה ב"ענן" של השרת שם – קובץ לפי משתמש

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

שם היכולת: שיחזור DOK

מהות היכולת: משחזר את הDOK במקרי הצורך

אוסף יכולות:

* קבלת נתונים מהלקוח
* שליפת קבצי המשתמש מה 'ענן'
* שליחת קבצים

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

שם היכולת: שליחת הודעות למשתמש

מהות היכולת: שולח התרעה למשתמש במקרה הצורך

אוסף יכולות:

* קבלת נתוני משתמש (התראה)
* שליפת נתונים מבסיס נתונים
* שליחת הודעת מייל למשתמש

רשימת אובייקטים: תקשורת, בסיס נתונים, מייל

**יכולות בצד לקוח – עמדת התקנה/שחזור:**

שם היכולת: הרשמה

מהות היכולת: רישום משתמש חדש

אוסף יכולות:

* ממשק משתמש - הכנסת שם משתמש, סיסמא, מייל
* שליחה מוצפנת לשרת
* קבלת תשובה של הרשמה
* הצגתה למשתמש

רשימת אובייקטים: ממשק משתמש, הצפנה/פענוח, תקשורת

שם היכולת: התחברות

מהות היכולת: מתחבר שרת

אוסף יכולות:

* ממשק משתמש - הכנסת שם משתמש וסיסמא
* שליחה מוצפנת לשרת
* קבלת תשובה שהתחברת
* הצגת התשובה למשתמש

רשימת אובייקטים: ממשק משתמש, הצפנה/פענוח, תקשורת

שם היכולת: התקנת DOK

מהות היכולת: מתקין את קבצי המערכת לתוך הDOK

אוסף יכולות:

* שליחת בקשה מוצפנת של התקנה לשרת
* קבלת קבצי מערכת
* הטענה קבצי המערכת ל DOK

רשימת אובייקטים: ממשק משתמש, הצפנה/פענוח, תקשורת, DOK

שם היכולת: שחזור DOK

מהות היכולת: משחזר את הDOK בעת הצורך

אוסף יכולות:

* התחברות
* שליחת בקשת שיחזור DOK מוצפן
* קבלת קבצי המשתמש
* העתקת קבצים המשתמש ל DOK

רשימת אובייקטים: ממשק משתמש, הצפנה/פענוח, תקשורת, DOK

**יכולות בצד לקוח:**

שם היכולת: כניסה לDOK

מהות היכולת: כניסה לתוך הDOK

אוסף יכולות:

* חיבור הDOK למחשב
* ממשק משתמש - הכנסת סיסמא
* פענוח רשימת הקבצים והצגתה

רשימת אובייקטים: DOK**,** ממשק משתמש, סיסמא, רשימת קבצים, פענוח

שם היכולת: יציאה DOK

מהות היכולת: יציאה מהDOK

אוסף יכולות:

* כניסה לDOK
* ממשק משתמש - בקשת יציאה
* הצפנת כל הקבצים הפתוחים
* שליחה לגיבוי של כל הקבצים שטרם נשלחו
* קבלת אישור ו הוצאת הDOK

רשימת אובייקטים: DOK, ממשק משתמש, הצפנה, גיבוי, תקשורת

שם היכולת: עבודה עם הDOK

מהות היכולת: עבודה עם הDOK עצמו

אוסף יכולות:

* כניסה לDOK
* ממשק משתמש – הצגת רשימת הקבצים
* פעולות לפי הצורך שלך
* ניטור קבצים פתוחים ושליחת שינויים לגיבוי לשרת אם קיים או הוספת השם ל'רשימת גיבויים'

רשימת אובייקטים: DOK, ממשק משתמש, גיבויים, ניטור קבצים פתוחים, תקשורת

שם היכולת: פיענוח קובץ DOK

מהות היכולת: מפענח קובץ DOK ספציפי

אוסף יכולות:

* כניסה לDOK
* פענוח רשימת קבצי DOK
* ממשק משתמש - בחירת קובץ
* פענוח הקובץ הספציפי
* ניטור שינויים בקובץ הפתוח
* שמירה של הקובץ שפענח
* שליחת שינוים לשרת אם קיים / לקובץ 'רשימת גיבויים' את שם הקובץ

רשימת אובייקטים: DOK, פענוח, ממשק משתמש, קובץ, תקשורת, ניטור

שם היכולת: הצפנת קובץ DOK

מהות היכולת: מצפין את הקובץ DOK

אוסף יכולות:

* פיענוח קובץ DOK
* ניטור הקובץ
* זיהוי 'סגירת קובץ'
* הצפנה
* שליחת גיבוי אם שרת קיים / לקובץ 'גיבוי' את שם הקובץ

רשימת אובייקטים: DOK, הצפנה, ניטור, גיבוי, תקשורת

שם היכולת: שליחת קבצים לגיבוי בשרת

מהות היכולת: שולח קבצים על מנת לגבות אותם בשרת

אוסף יכולות:

* כניסה לDOK
* זיהוי תקשורת עם השרת – שליחת קבצים שטרם גובו ע"פ הרשימה

רשימת אובייקטים: DOK, הצפנה, תקשורת, FILE XEPLORER, ניטור, רשימת קבצים לגיבוי

שם היכולת: מחיקת הDOK

מהות היכולת: מחיקת כל הDOK

אוסף יכולות:

* כניסה לDOK
* כשל בכניסה
* מחיקה של הDOK
* שליחת הודעת מחיקה לשרת

רשימת אובייקטים: DOK, מחיקה, תקשורת

**פרק ד' - 'שם המערכת' – העיצוב**

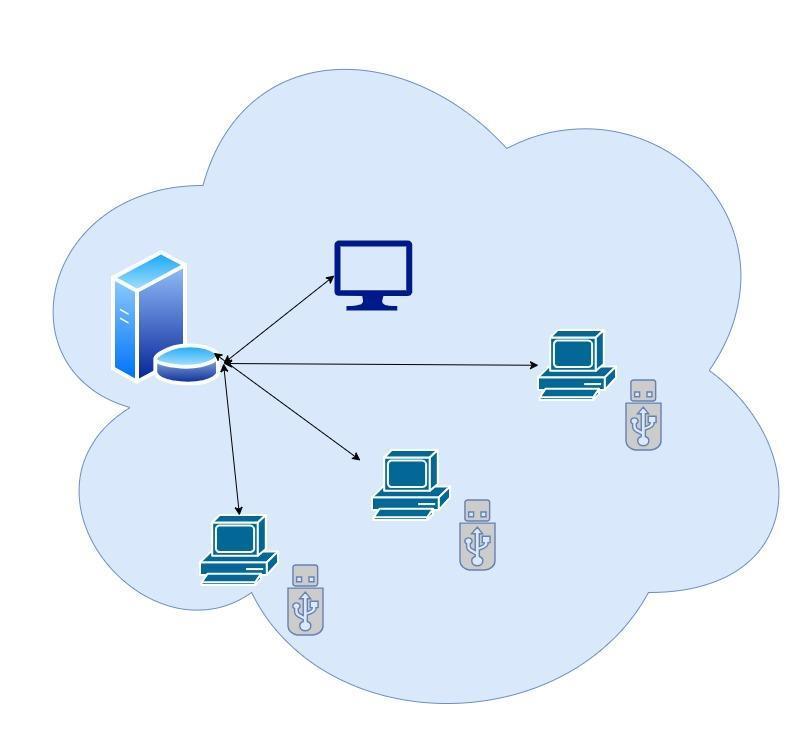
1. תיאור הארכיטקטורה של המערכת המוצעת

שרת מרובה לקוחות מבוסס מערכת הפעלה windows

עמדת התקנה מבוססת מערכת הפעלה windows

לקוחות – מחשבים מבוססי מערכת הפעלה windows

שימוש ב USB בעל מערכת קבצים FAT32  
לטובת גיבוי של ה USB הלקוחות והשרת צריכים לשבת על אותה הרשת.



1. הטכנולוגיה הרלוונטית

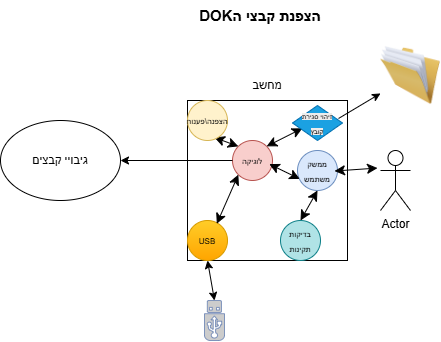
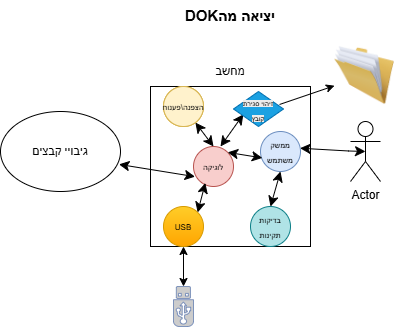
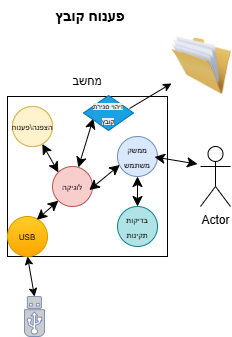
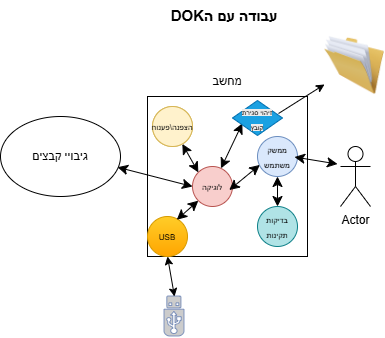
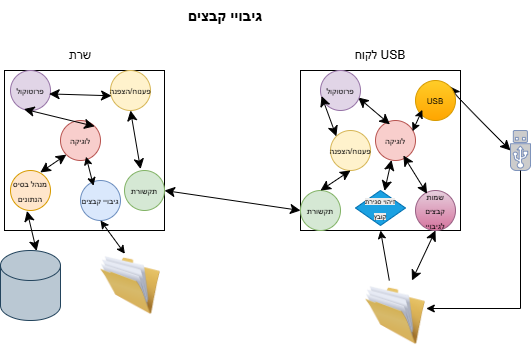
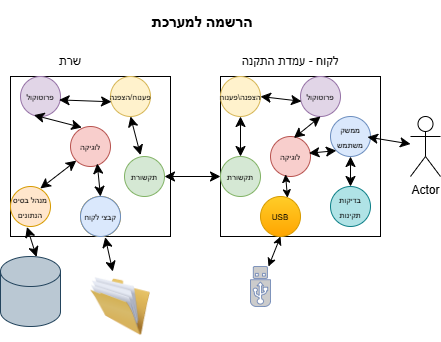
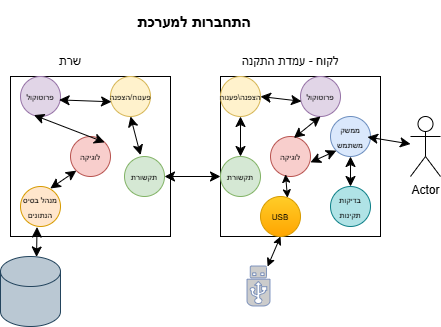
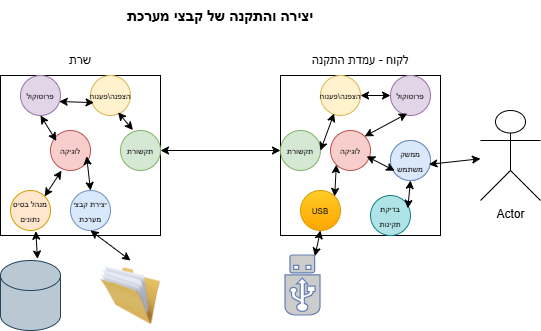
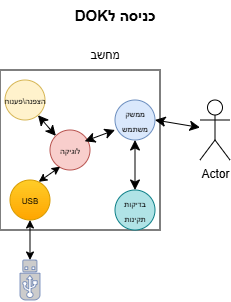
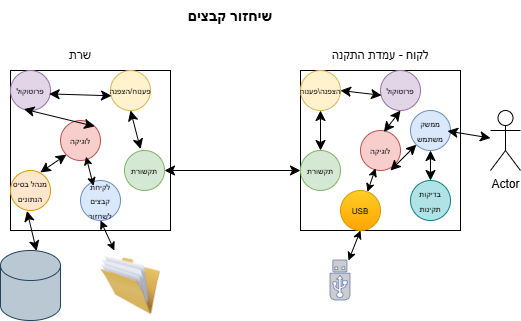
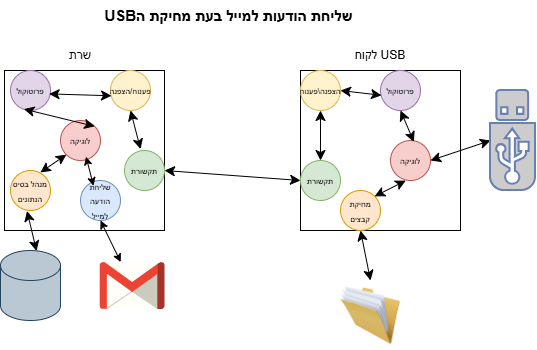
שפת התכנות שאני אשתמש בה היא פייתון וSQL מכיוון שבהם אני הכי מתורגל ואני צריך בסיס נתונים בשביל הפרוייקט.

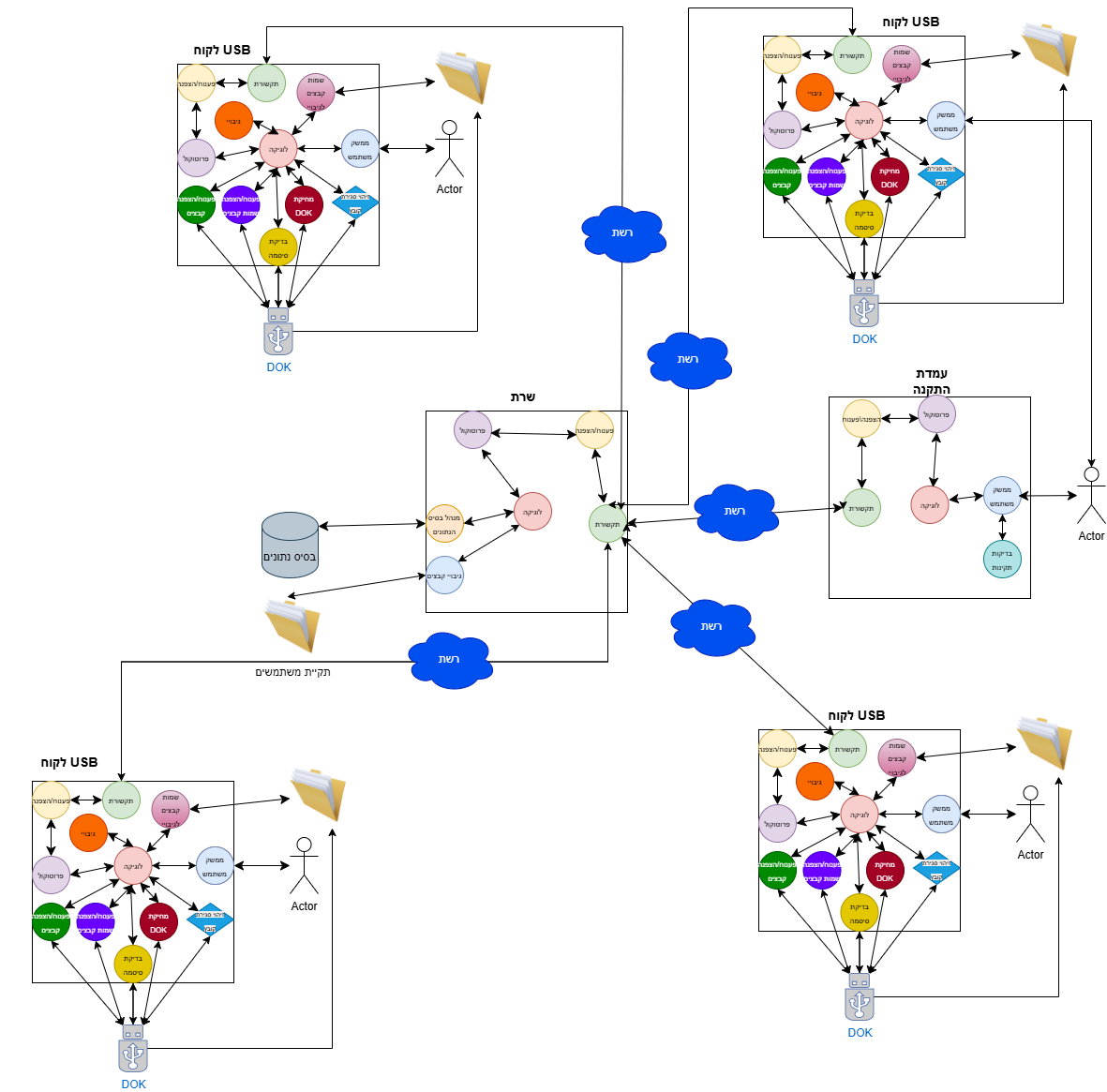
התקשורת שתהיה לי תהיה תקשורת מבוססת socket – ים המעבירים מידע בעזרת פרוטוקול TCP מכיוון שכל הרעיון של הפרוייקט הוא הגנה ואמינות ואני רוצה שכל המידע יעזור כמו שצריך

אני משתמש בפרויקט בהרבה הצפנות וההצפנות שאני אשתמש בהם יהיו AES לשמות קבצים ולקבצים עצמם ולשמות משתמש אני אשתמש בHASH כדי להגן על המשתמש – המפתחות שלי יהיו אי סימטריות (RSA) אז ההצפנה תקרא הצפנה היברידית.

שימוש במערכת ההפעלה של הDOK ובמערכת הקבצים מסוג FAT32

1. תיאור מודולים בהם נעשה שימוש





3, א.

**מודלים בהם אני עושה שימוש:**

| לטובת | שם המודל |
| --- | --- |
| לשימוש בתקשורת בעצמאות סוקטים | Socket |
| ניהול קבצים ומערכת הפעלה | os |
| מימוש שרת מרובה לקוחות | select |
| הצפנת מידע סימטרית מהירה | Aes |
| לטובת המפתחות | Rsa |
| יצירת חתימה דיגיטלית מאובטחת של הסיסמאות | SHA256 |
| השלמת מידע לאורך תקין - הצפנות | padding |
| לטובת מימוש הגרפיקה בפרוייקט | wxPython |
| נותן גישה לכל הקבצים בFILE EXPLORER המקורי | psutil |
| לטובת עזרה של שילוב המפתחות עם ההצפנה | fernt |
| יצירת מספרים וערכים אקראיים | random |
| ניהול זמן והשהיית תוכנית | time |
| ניהול תורים של נתונים לטובת התקשורת | queue |
| הרצת מספר משימות במקביל | threading |
| הרצת תוכניות ופקודות חיצוניות | subprocess |
| שימוש בספריות הצפנה מתקדמות | cryptodome |
| ניהול בסיס נתונים מקומי | sqlite3 |

3, ב.

**מחלקות:**

**לקוח:**

* תקשורת:

| Client\_com | | |
| --- | --- | --- |
| משמש לתקשורת של הלקוח עם השרת | | |
| **שם משתנה** | **תפקיד** | |
| my\_socket | הסוקט של הלקוח דרכו יעבור תקשורת | |
| server\_ip | כתובת השרת | |
| port | הפורט שהשרת רץ עליו | |
| recvQ | תור דרכו יעבור מידע מהתקשורת ללוגיקה | |
| Cipher | עצם הצפנה מול שרת | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| \_\_init\_\_ (בנאי) | * Server\_ip * Port * recvQ | מאתחל את המשתנים ומפעיל את \_main\_loop כ thread |
| \_main\_loop | ללא פרמטרים | הלולאה הכללית של התקשורת: יוצר את סוקט הלקוח וההתחברות לשרת מחליף מפתחות ויוצר עצם הצפנה באמצעות \_change\_key  ובלולאה מטפל בתקשורת מול השרת |
| \_change\_key | ללא פרמטרים | מבצע החלפת מפתחות עם השרת ויוצר את עצם ההצפנה Cipher |
| \_close\_client | ללא פרמטרים | סוגר את הסוקט |
| send\_msg | * msg | מצפין ושולח את ההודעה לשרת |
| send\_file | * file\_data | מצפין ושולח את הקובץ לשרת |

* לוגיקה ב DOK:

| Client\_logic | | |
| --- | --- | --- |
| משמש ללוגיקה של הלקוח כל הפעולות שיקרו בפועל יהיו כאן | | |
| **שם משתנה** | **תפקיד** | |
| backup\_files\_list | רשימת קבצים לגיבוי | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| main |  | יוצר client\_com מפעיל את send\_backup כ thread  יוצר תור להודעות מהגרפיקה מפעיל את handle\_msg כ thread מפעיל את הגרפיקה (עם client\_com ועם ה recvQ) |
| handle\_msg | * recvQ * client\_com | בלולאה אינסופית – מוציא שמות קבצים מהתור (recvQ) מפעיל את monitor\_file כ- thread |
| monitor\_file (thread) | * file\_name | מחכה לזיהוי של 'סגירת קובץ' ואז מצפין ומוסיף את הקובץ לרשימת קבצים לגיבוי |
| Send\_backup (thread) | * עצם תקשורת | פותח את קובץ הגיבוי ומכניס את השמות ל backup\_files\_list  ואז בלולאה:  במידה ו- backup\_files\_list לא ריקה: מוציא שם קובץ מצפין ושלוח לשרת אם אין תקשורת מוסיף את השם לקובץ - backup |

* הצפנה/פענוח:

|  |  |  |
| --- | --- | --- |
| Private\_cypher | | |
| משמש להצפנה ופיענוח של הקבצים עצמם של הDOK | | |
| **שם משתנה** | **תפקיד** | |
| user\_name | שם משתמש | |
| Hashed\_passward | סיסמה בHASH | |
| key | המפתח להצפנות והפענוח | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| \_\_init\_\_ (בנאי) | * User\_name * password | מאתחל את USER\_NAMEומפעיל את ה\_create\_key |
| create\_key | * password | הופך את הpassword לhash ומאתחל אותה.  יוצר את מפתח ההצפנה. |
| encrypt\_file | * file\_input | מצפין את קובץ המקור ומחליף אותו בקובץ מוצפן |
| decrypt\_file | * file\_enc\_input | מפענח את קובץ המקור ומחליף אותו בקובץ מפוענח |
| encrypt\_msg | * msg | מצפין את ההודעה |
| decrypt\_msg | * Msg | מפענח את ההודעה |

**עמדת התקנה:**

* תקשורת:

| stand\_com | | |
| --- | --- | --- |
| משמש לתקשורת של העמדת התקנה עם השרת | | |
| **שם משתנה** | **תפקיד** | |
| stand\_socket | הסוקט של העמדת התקנה דרכה יעבור תקשורת | |
| server\_ip | כתובת שרת | |
| port | הפורט שהשרת רץ עליו | |
| recvQ | תור דרכו יעבור מידע מהתקשורת ללוגיקה | |
| Cipher | עצם הצפנה מול השרת | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| \_\_init\_\_ (בנאי) | * Server\_ip * Port * recvQ | מאתחל את המשתנים ומפעיל את \_main\_loop כthread |
| \_main\_loop (thread) | ללא פרמטרים | הלולאה הכללית של התקשורת: יוצר את הסוקט של עמדת ההתקנה וההתחברות לשרת. מחליף מפתחות ויוצר עצם הצפנה באמצעות \_change\_key  ובלולאה מטפל בתקשורת מול השרת |
| \_change\_key  מופעל כthread | ללא פרמטרים | מבצע החלפת מפתחות עם השרת ויוצר את עצם ההצפנה Cipher |
| \_close\_client | ללא פרמטרים | סוגר את הסוקט |
| send\_msg | * msg | מצפין ושולח את ההודעה לשרת |
| send\_file | * file\_data | מצפין ושולח את הקובץ לשרת |
| recv\_msg | ללא פרמטרים | קבלת הודעה מהשרת |
| recv\_file | ללא פרמטרים | מקבל קובץ מהשרת |

* לוגיקה – לעבור מאפס:

| stand\_logic | | |
| --- | --- | --- |
| משמש ללוגיקה של העמדת התקנה כל הפעולות שיקרו בפועל יהיו כאן | | |
| **שם משתנה** | **תפקיד** | |
| stand\_com | עצם של התקשורת של עמדת ההתקנה | |
| Design\_stand | עצם של הגרפיקה | |
| msgQ | תור שדרכו מגיע ההודעות מהתקשורת ללוגיקה | |
| clientQ | תור שדרכו מגיע מידע מהמשתמש | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| \_\_init\_\_ (בנאי) | * msgQ * clientQ | שמורים אצלו עכשיו במחלקה |
| \_handle\_msg | * msgQ * clientQ | עושה את הפעולה לפי האופקוד שקיבל מהשרת או משלקוח המחובר לעמדה |
| Sigh\_in | * User\_name * password * mail | נרשם למערכת |
| Log\_in | * User\_name * password | מתחבר למשתמש |
| Add\_dok | * dok\_path * user\_name | מטעין לתוך הDOK את קבצי המערכת |
| restore | * dok\_path * user\_name | משחזר לפי השם משתמש והDOK את נתונים בDOK |
| Change\_settings | * User\_name * password * mail | משנה את הנתונים של המשתמש |

**שרת:**

* תקשורת:

| server\_com | | |
| --- | --- | --- |
| משמש לתקשורת של השרת עם הלקוח וגם עם העמדת התקנה | | |
| **שם משתנה** | **תפקיד** | |
| server\_socket | סוקט השרת | |
| recvQ | תור המידע מהתקשרות ללוגיקה | |
| Open\_clients\_list | הלקוחות המחוברים | |
| Stands\_list | העמדות התקנה המחוברות | |
| port | פורט המערכת | |
| Private\_key | המפתח הפרטי של השרת | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| \_\_init\_\_ (בנאי) | * port * recvQ | שמורים אצלו עכשיו במחלקה |
| \_main\_loop (thread) | ללא פרמטרים | הלולאה הכללית של התקשורת:  מקבל בקשות חיבור של לקוחות. מחבר אותם ומחליף איתם מפתחות באמצעות \_change\_key.  בלולאה מטפל בתקשורת מול הלקוחות שלו. |
| \_change\_key (thread) | ללא פרמטרים | מבצע החלפת מפתחות עם הלקוח ויוצר את המפתח בינהם |
| \_close\_client | * client\_socket | סוגר את הלקוח |
| close\_client | * client\_ip | מפעיל את הפעולה הפנימית של \_close\_client |
| send\_msg | * msg * client\_ip | מצפין ושולח את ההודעה ללקוח לפי הIP שלו |
| send\_file | * File\_data * client\_ip | מצפין ושולח את הקובץ ללקוח לפי הIP שלו |
| recv\_msg | ללא פרמטרים | קבלת הודעה מהלקוח או עמדת ההתקנה |
| recv\_file | ללא פרמטרים | מקבל קובץ מהלקוח |
| get\_socket\_by\_ip | * client\_ip | מחזיר את הסוקט של client\_ip |
| check\_valid\_ip | * client\_ip | מחזיר valid\_client\_ip |

* לוגיקה – לעבור על הלוגיקה:

| server\_logic | | |
| --- | --- | --- |
| משמש ללוגיקה של השרת, כל הפעולות שיקרו בפועל יהיו כאן | | |
| **שם משתנה** | **תפקיד** | |
| server\_com | עצם של התקשורת של השרת | |
| recvQ | תור שדרכו מגיע ההודעות מהתקשורת ללוגיקה | |
| Dok\_DB | בסיס הנתונים | |
| commends | פעולות | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| \_\_init\_\_ (בנאי) | * recvQ * Dok\_DB | מאתחל את המשתנים ומפעיל את \_handle\_msg כthread |
| \_handle\_msg | * recvQ | בלולאה אינסופית – מוציאה מידע מהתור(recvQ) ומפרק אותו בעזרת הפרוטוקול.  מפעיל את הפעולה לפי הopcode שהתקבל |
| Restore\_clients\_dok | * dok\_path * user\_name | לוקח את הקבצים של הDOK הנמען ושולח אותם לעמדת ההתקנה |
| Storage | * dok\_path * user\_name * file\_name * file | שומר את הקובץ שהתקבל בתקיה המתאימה |
| send\_del\_msg | * user\_name | שולח הודעת מחיקת DOK למייל |
| Add\_user | * User\_name * Hashed\_password * mail | מוסיף משתמש לטבלה של הuser – ים |
| Add\_dok | * dok * user\_name | מוסיף DOK לטבלה של הDOK -ים |
| Update\_mail | * mail * user\_name | מעדכן את המייל של הuser |

בסיס נתונים:

| Dok\_DB | | |
| --- | --- | --- |
| בסיס הנתונים בפרוייקט | | |
| **שם משתנה** | **תפקיד** | |
| Db\_name | שם הבסיס נתונים | |
| conn | חיבור לבסיס נתונים | |
| cursor | המידע העובר בין החיבור לבסיס נתונים | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| Init (בנאי) | * Dok\_DB | מאתחל את DB\_name ומפעיל את \_create |
| create\_ |  | פעולה פנימית היוצרת/מתחברת ממסד נתונים  מתאחל את CONN ואת CURSOR |
| user\_exsit | * user\_name | מחזיר האם הuser\_name קיים בטבלה |
| get\_mail | * user\_name | מחזיר את מייל הלקוח |

**פרוטוקול:**

* שרת:

| Server\_protocol | | |
| --- | --- | --- |
| פרוטוקול השרת | | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| Unpack\_stand | * msg | מחזיר הודעה מפורקת בהתאם לפרוטוקול עמדת התקנה |
| Unpack\_client | * msg | מחזיר הודעה מפורקת בהתאם לפרוטוקול הלקוח |
| pack\_stand | * opcode * file\_name * file | מחזיר הודעה בנויה בהתאם לפרוטוקול |

* לקוח:

| client\_protocol | | |
| --- | --- | --- |
| פרוטוקול הלקוח | | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| pack | * opcode * File\_name * file * dok\_path * user\_name * folder | מחזיר הודעה בנויה בהתאם לפרוטוקול |

* עמדת התקנה:

| stand\_protocol | | |
| --- | --- | --- |
| פרוטוקול עמדת התקנה | | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| Unpack | * msg | מחזיר הודעה מפורקת בהתאם לפרוטוקול |
| pack | * opcode * user\_name * password * mail * dok\_path | מחזיר הודעה בנויה בהתאם לפרוטוקול |

הצפנה/פענוח:

| cypher | | |
| --- | --- | --- |
| ההצפנה של כל התקשורת בין השרת ללקוחות שלה | | |
| **שם משתנה** | **תפקיד** | |
| Key | המפתח שלך (לא משנה צד לקוח או שרת) | |
| **שם פעולה** | **טענת כניסה** | **טענת יציאה** |
| Init (בנאי) | * Key | מאתחל את key |
| Encryption | * data | מחזיר את המידע מוצפן |
| Decryption | * data | מחזיר את המידע מפוענח |

תיאור סביבת הפיתוח

**שפות תכנות:**

PYTHON – שפת הבסיס של הפרויקט היא תשמש לרוב הפרויקט  
 SQL – קיים רק בצד שרת לעבודה מול בסיס הנתונים

HTML - רק אם ישאר זמן אני ארצה ליצור אתר הרשמה במקום עמדת התקנה שתעשה בדיוק מה שעמדת ההתקנה אמורה לעשות(מסמן את זה כי לא בטוח יקרה)

**כלי פיתוח וכלים הנדרשים לבדיקות:**

PYCHARM – כלי בו ייכתב כל הקוד של המערכת צד שרת צד לקוח

USB – שם מתקיים כל ההצפנה והפיענוח עצמו ועל זה מבוסס כל הפרויקט

WIRSHARK – ישמש לבדיקות תקשורת

FILE EXPLORER – ישמש לבדיקה של מידע מוצפן בתוך הDOK ולבדיקת הקבצים

DB BROWSER – צפיה בבסיס נתונים

1. תיאור האלגוריתמים המרכזיים בפרויקט

סוגיה - פענוח הקבצים  
תיאור – האם בעת הזדהות נכונה של המשתמש מפענחים לו את כל הקבצים

| מספר | האופציה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | פענוח מידי של כלל הקבצים | כל הקבצים זמינים ללקוח | תהליך ארוך – יכול להתיש את המשתמש וגם לעכב את כניסתו ויציאתו מהמערכת  תהליך מיותר – יתכן ובכלל לא ייגש לקבצים  מסוכן – כל הקבצים חשופים ופגיעים |
| 2 | פענוח שמות הקבצים בלבד ופענוח קובץ רק לאחר בחירתו | תהליך קצר לא חושף קבצים שלא משתמשים בהם | פתיחת קובץ יכולה להתעכב |

החלופה שנבחרה היא **פענוח שמות הקבצים בלבד ופענוח קובץ רק לאחר בחירתו** מכיוון שבאופציה הזאת אני לא חושף קבצים לחינם שזה מהות המערכת

סוגיה - שמירת שם משתמש וסיסמה  
תיאור – איך שומרים את המידע של זיהוי המשתמש שכן הוא צריך יכולת לעבוד גם ללא שרת?

| מספר | האופציה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | שמירת הפרטים בתוך הקוד | קל למימוש | לא מאובטח פריץ מאוד למשתמשים בIDA – REVERS ENGENIRING |
| 2 | שמירת הפרטים בתוך קובץ חיצוני מוסווה בתוך מלא מידע | קל למימוש | חשוף מאוד  גכל בן אדם הנמצא ליד הUSB יכול לגשת לזה בקלות  אפשר להצפין את זה בקלות ואז לא נוכל לדעת איך לפענח פעם הבאה |
| 3. | שמירה פרטי משתמש בתוך הקוד אך בעזרת הHASH שלהם | מוגן מאנשים  בן אדם שעושה REVERS ENGENIRING לא יוכל לעשות כלום עם המידע הזה | יותר עבודה לממשק משתמש – צריך לעשות HASH בעצמו כל פעם על מנת לבדוק אם הקוד תקין |

החלופה שנבחרה היא **שמירה פרטי משתמש בתוך הקוד אך בעזרת הHASH שלהם** מכיוון שזה הכי בטוח והחשיבות הכי גדולה של הפרויקט זה האבטחה שלו ואיני רוצה שבן אדם מהצד פשוט יוכל לבוא ולהיכנס לקוד שלי

סוגיה - העברת המידע לגיבוי בשרת  
תיאור – איך המידע עובר לשרת וכל כמה זמן? יתכנו מצבים שהשרת לא זמין

| מספר | האופציה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | העברת כלל המידע לשרת (אם קיים) ביציאה מהDOK | פשוט למימוש העברת כלל המידע בפעם אחת | פגיעה בחווית המשתמש - בעת סגירת הDOK ייקח למשתמש הרבה זמן לצאת  אם המשתמש מוציא את ה DOK לפני הזמן המידע פשוט לא יעבור ושום דבר לא ישמר בשרת |
| 2. | ניטור הקבצים ובעת סגירת קובץ שליחה לשרת, אם אין שרת שמירת השמות בקובץ מסודר ומוסתר ושליחה בפעם הבאה | אפשרות עבודה בזמן השליחה, זמן היציאה מתקצר כי רוב הקבצים ישלחו מייד עם סגירתם | תהליך יותר מורכב – צריך לנטר כל הזמן את העבודה עם הקבצים  תיתכן גם שליחה כפולה של אותו הקובץ אם הוא נפתח ונסגר כמה פעמים |

החלופה שנבחרה היא **ניטור הקבצים ובסגירתם שליחה לשרת, אם אין שרת שמירת השמות בקובץ מסודר ומוסתר ושליחה בפעם הבאה** בדרך זו אין פגיעה בחוויית המשתמש

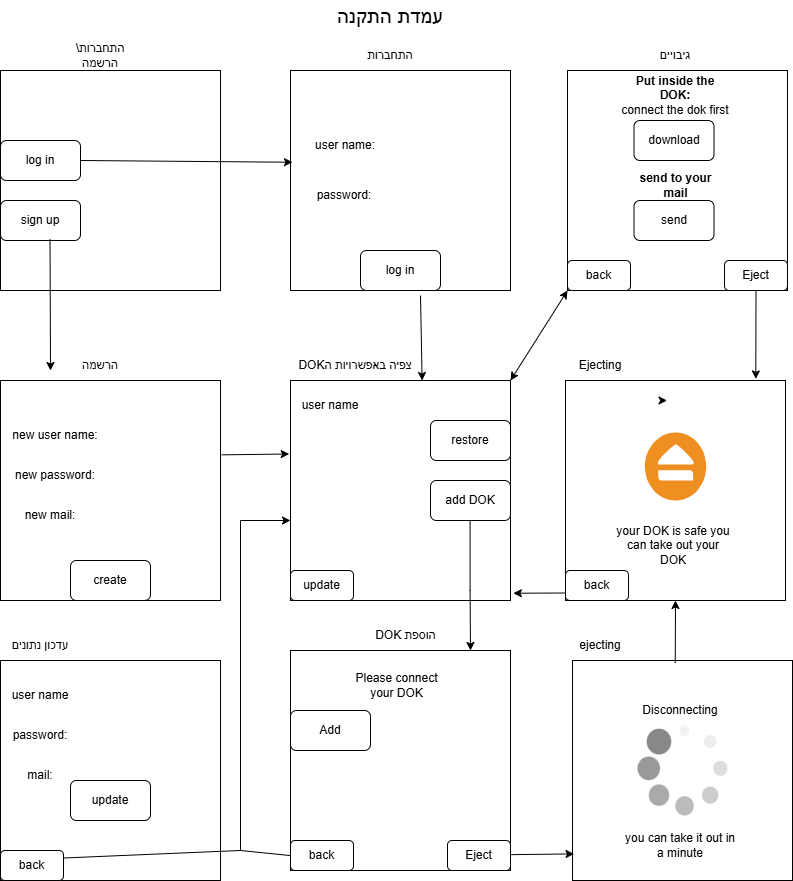
סוגיה - מתי להצפין את הקבצים חזרה  
תיאור – מתי הכי נכון להצפין את הקבצים חזרה בשביל שמירה עליהם

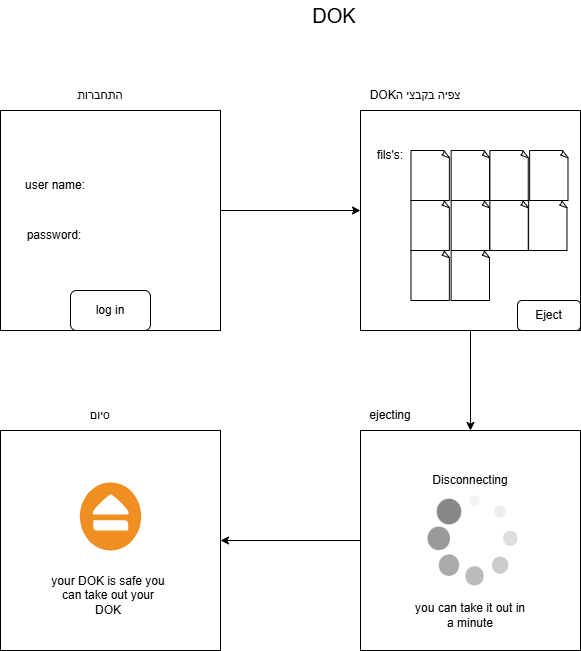
| מספר | האופציה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | הצפנה בניתוק הDOK | הצפנה מסודרת פעם אחת בסיום העבודה עם הקבצים  עבודה חוזרת עם אותו קובץ – יותר מהירה (הצפנה ופענוח פעם אחת) | מעכב את היציאה מהDOK בעת סגירה  במידה ונשכח הDOK הקבצים חשופים לעיני כל (כי הוא לא התנתק)  אם המשתמש מנתק את הDOK בלי יציאה מסודרת - לחכות להצפנה הכל / חלק יישאר מפוענח |
| 2. | הצפנה בעת סגירה של קובץ | היציאה מהירה - אין פגיעה בחוויית המשתמש (אין המתנה להצפנה)  מגן ממשתמש שמוציא סתם את הDOK בלי לעשות יציאה מסודרת | מורכב יותר - צריך לממש מנגנון שמזהה סגירת קובץ  במידה והמשתמש לא סוגר את הקובץ הוא יישאר פרוץ  ההצפנה תחזור כל פעם שיפתח וייסגר הקובץ  אם המשתמש מנתק את הDOK בלי יציאה מסודרת - לחכות להצפנה הכל / חלק יישאר מפוענח |
| 3. | הצפנה תמידית – כל זמן שהDOK בתוך המחשב, מגבים את הקבצים הפתוחים | אם שכחת את הDOK או הלכת והשארת אותו על המחשב הDOK יהיה מוצפן | מורכב יותר - צריך לממש מנגנון שמזהה קובץ פתוח  יכול לפגוע בחוויית המשתמש מבחינת ביצועים (תהליך כבד ברקע) ההצפנה תחזור כל פעם שיפתח וייסגר הקובץ  אם המשתמש מנתק את הDOK בלי יציאה מסודרת - לחכות להצפנה הכל / חלק יישאר מפוענח |

החלופה שנבחרה היא **הצפנה בעת שמירת קובץ** בדרך הזו חווית המשתמש לא תפגע בעת שימוש הDOK ותשמור על הפרטיות שלו גם במקרה שהלך או שכח את הDOK מחובר למחשב – כן זה מחייב אותו **באחריות מינימלית** של סגירת הקבצים שאיתם עבד. אולי אממש בנוסף גם מנגנון של נעילת ה DOK במידה ולא נעשה בו שימוש לאורך זמן.

1. תיאור מסכי הפרויקט:

למחוק את המייל ולשנות את הסדר של הלחכות



****

1. תיאור פרוטוקול התקשורת

מבנה כללי:

פרוטוקול:   
עמדת התקנה 🡨 שרת : ישלח אורך בבית בודד לא מוצפן ואז ההודעה מוצפנת, @#2 מפריד בין שדות ההודעה

שרת 🡨 עמדת התקנה : ישלח אורך בשני בתים לא מוצפן ואז ההודעה מוצפנת, @#2 מפריד בין שדות ההודעה

לקוח 🡨 שרת : ישלח אורך בשני בתים לא מוצפן ואז ההודעה מוצפנת, @#2 מפריד בין שדות ההודעה

| שם | נשלח מ | מתקבל אצל | פקודה | שדות בהודעה | דוגמה |
| --- | --- | --- | --- | --- | --- |
| רישום | עמדת התקנה | שרת | 00 | 1. שם משתמש  2. סיסמה  3. מייל  4 קוד פקודה | **00**@#2**name**@#2**password**@#2**eyal@gmail.com** |
| רישום - תגובה | שרת | עמדת התקנה | 00 | 1. סטטוס  2. קוד פקודה | **00**@#2**status** |
| התחברות | עמדת לקוח | שרת | 01 | 1. שם משתמש  2. סיסמה  3. קוד פקודה | **01**@#2**name**@#2**password** |
| התחברות – תגובה | שרת | עמדת התקנה | 01 | 1. סטטוס  2. קוד פקודה | **01**@#2**status** |
| עדכון פרטים | עמדת לקוח | שרת | 02 | 1. שם משתמש  2. סיסמה  3. מייל  4. קוד פקודה | **02**@#2**name**@#2**password**@#2**eyal@gmail.com** |
| עדכון פרטים תגובה | שרת | עמדת התקנה | 02 | 1. סטטוס  2. קוד פקודה | **02**@#2**status** |
| גיבויי | DOK | שרת | 03 | 1. קובץ  2. שם קובץ  3. קוד פקודה  4. שם משתמש  5. DOK  6. תקיית קבצים לגיבוי | **03**@#2**file\_name**@#2**file**@#2**name**@#2**dok1**@#2**needToSave** |
| שחזור | עמדת התקנה | שרת | 04 | 1. קוד פקודה 2. שם משתמש 3. DOK | **04**@#2**name**@#2**dok1** |
| שחזור - תגובה | שרת | עמדת התקנה | 04 | 1. שמות קבצים  2. קבצים  3. סטטוס | **04**@#2**files\_name**@#2**files** |
| התרעה | DOK | שרת | 05 | 1. קוד פקודה | **05**@#2 |
| ניתוק הDOK | עמדת התקנה | DOK | 06 | 1. קוד פקודה | **06**@#2 |
| הוספת DOK | עמדת התקנה | שרת | 07 |  |  |
| הוספת DOK - תגובה | שרת | עמדת התקנה | 07 |  |  |

1. תיאור מבני הנתונים

בסיס נתונים – SQL

שם בסיס נתונים – dok\_DB

שם הטבלה – **users** , שמירת פרטי המשתמשים

טבלה: המפתח הוא ה user

|  |  |  |  |
| --- | --- | --- | --- |
| שדה | mail | Password\_hash | User(key) |
| אורך | 0-50 | 64 | 0-10 |
| סוג | <string> | <string> | <string> |
| דוגמה | Eyalen22221@gmail.com | Adljfoinadof | Eyal en |
| דוגמה | idovz@gmail.com | ldisfionadofn | Ido vz |
| דוגמה | MerryHamalca@gmail.com | iosduhaadfg | Merry |

שם הטבלה – **doks** , שמירת פרטי ה DOK -ים של המשתמשים

טבלה: המפתח הוא צירוף של user+DOK\_name

|  |  |  |
| --- | --- | --- |
| שדה | DOK\_Name | User |
| אורך | 20 | 0-10 |
| סוג | <string> | <string> |
| דוגמה | DOK1 | Eyal en |
| דוגמה | DOK2 | Eyal en |
| דוגמה | DOK1 | Merry |

קבצי המשתמשים ישמרו בשרת בתיקיית files שעבור כל משתמש תהייה תיקייה על שמו ושם ימצאו כל קבציו, עבור כל DOK תיקייה שהיא בשם ה DOK ומכילה את הקבצים ששייכים לו

1. סקירת חולשות והאיומים

**שכבת האפליקציה:**

Sql injection – הלקוח מכניס לתוך הבסיס נתונים קוד זדוני אשר יכול לגרום לנזק לקוד. בעקבות שדות קלט המהווים גישה לבסיס נתונים קיים הסיכון הזה, הפתרון הוא בדיקה טובה של המתקבל מן הלקוח, קוד נכון ופיצול הבדיקה בין הנתונים לפעולה עצמה.

DDOS – שליחה מרובה זדונית לשרת בכוונה להשבית אותו, פתרון - בדיקת חיבורי IP בזמן קצוב

גניבת הDOK – כל המידע מוצפן - לגנב אין גישה למידע.   
בנוסף אחרי שלושה כישלונות בכניסה למערכת – המידע בדיסק יושמד.  
מערכת ההתקנה מאפשרת למשתמש לשחזר את ה DOK בכל אחד מהמצבים הנ"ל.

**שכבת התעבורה:**

MITM – בן אדם מקשיב לתקשורת ברקע מבלי שיש לנו אפשרות לדעת והוא יכול לקחת המידע שלנו ו/או לשנות אותו , פתרון העברה מידע אך ורק מוצפן כדי שגם אם יהיה לו את המידע הוא לא יוכל לעשות איתו כלום

סוג הצפנה – קיימת הצפנה סימטרית שהיא הצפנה מאוד חזקה מהירה ובלתי פריצה אך קיימת אצלה בעיה - העברת המפתח חייבת להתבצע והמפתח חייב להיות אותו הדבר ואין אפשרות פשוט להעביר את המפתח ככה באינטרנט כי זה בכלל לא בטוח, יש גם את האופציה של הצפנה אסימטרית אבל לה יש בעיה משלה אומנם היא עונה על הפתרון של העברת המפתח אבל ההצפנה עצמה כבדה ולוקחת יותר זמן.

קיים פתרון למצב והוא הצפנה היברידית המשלבת בין שניהם לוקחת את הטוב בכל דבר את העברת המפתח של הצפנה אסימטרית ואת ההצפנה של ההצפנה הסימטרית עם שימוש בAES וככה פותר את הבעיות של שתי ההצפנות

פרוטוקול תקשורת – הפרוטוקול תקשורת חייב לשמור על כל המידע שעובר בפרויקט כי המידע הוא אישי ורגיש של כל בן אדם ואני לא רוצה שמידע יאבד בדרך, הפתרון הוא פרוטוקול TCP הנחשב יותר אמין ובטוח לשימוש.

**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**