

**<CommanDominion>**

מסמך עיצוב

<אייל אמדור>

<גרסה 1.6>

<15/05/2017>

**היסטוריית גרסאות המסמך**

|  |  |  |
| --- | --- | --- |
| תאריך | גרסה | תקציר השינויים |
| *30/10/2016* | 1.1 | זאת הגרסה הראשונה |
| *16/11/2016* | 1.2 | נוצר לוגו + הוספת מטרה – קדמה טכנולוגית |
| 07/02/2017 | 1.3 | שינוי שם, לוגו ובניית השתלטות בין שתי לקוחות |
| 10/03/2017 | 1.4 | בניית GUI למשתמש ,חיבור לשרת מרכזי והוספת סמל בדסקטופ (קובץ exe) |
| 12/04/2017 | 1.5 | שיפור התשתית |
| 15.05.2017 | 1.6 | חיבור בין GUI לבין התכנית שמאפשרת השתלטות ושיפורים אחרונים |

לוגו עמיאסף.jpg

**1. הקדמה**

1.1 מטרה

מטרתו העיקרית של ה-CommanDominion היא תקשורת קלה ונוחה בין מחשבים, בעזרת ה-CommanDominion ניתן לשלוט על כל מחשב בעולם (שעליו מותקנת התוכנה) מכל מחשב אחר.

1.2 המוצר

*שם המוצר:* CommanDominion*.*

ה-CommanDominion הוא כלי חזק, חינמי לכל וקל לשימוש למטרת בין מחשבים ושליטה של מחשב א' על מחשב ב'.

התוכנה מאפשרת הצגת המסך של המחשב האחר על גבי המסך של המסך המתחבר. התוכנה מספקת יכולת להשתמש בעכבר ובמקלדת המקומיים כדי לשלוט ולנהל את המחשב המרוחק. המוצר מאפשר שליטה מרחוק במחשב האישי של שותף עסקי\לקוח\חבר כאילו ישבת מולו.  באמצעות ה-CommanDominion תוכל להתחבר אל הלקוח שלך בתוך שניות מעטות ובקלות רבה.

מטרתו העיקרית של המוצר היא תקשורת קלה ונוחה בין מחשבים והייחוד בו הוא שהוא ידידותי למשתמש, נוח לשימוש, חינמי וקל להתקנה.

**1.3 הגדרות**

* *לקוח (Client) - משתמש אשר מוריד את החבילה של* ה-CommanDominion והוא יכול או להשתלט או לשלוט על מחשב אחר.
* *שרת (Server) - שרת מרכזי אשר מנתב ומקשר בין הלקוחות.*
* Graphics - חבילה שהורדתי מהאינטרנט ושיפרתי בעצמי אשר מאפשרת תקשורת נוחה וקלה עם ה-TKinter (ממשק GUI).
* מחשב שולט (Command) - בפרויקט התייחסתי למחשב השולט כ- Command וקיימת תוכנית שזוהי מטרתה (להשתלט על תכנית אחרת ששמה Dominion).
* מחשב נשלט (Dominion) - בפרויקט התייחסתי למחשב הנשלט כ- Dominion וקיימת תוכנית שזוהי מטרתה (להיות נשלטת על ידי התכנית Command).
* שגיאה (Error) - מצב בו התכנית לא יכולה לפעול מכיוון שיש תקלה.

**1.4 תקציר**

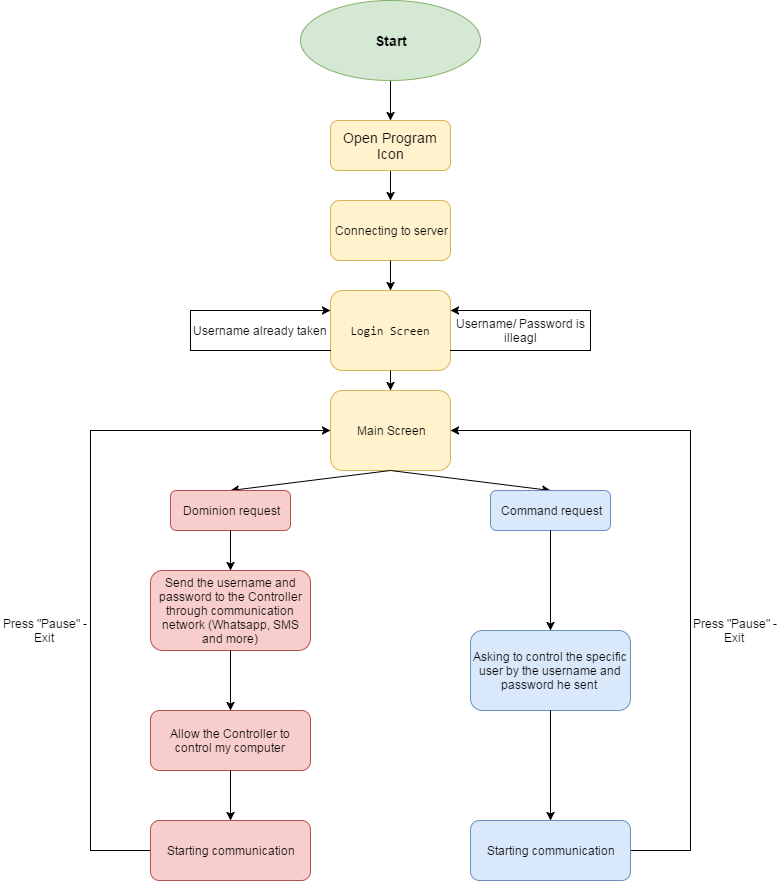
*בהמשך המסמך אפרט כיצד פועלת התוכנה.*

1. ארכיטקטורת המערכת

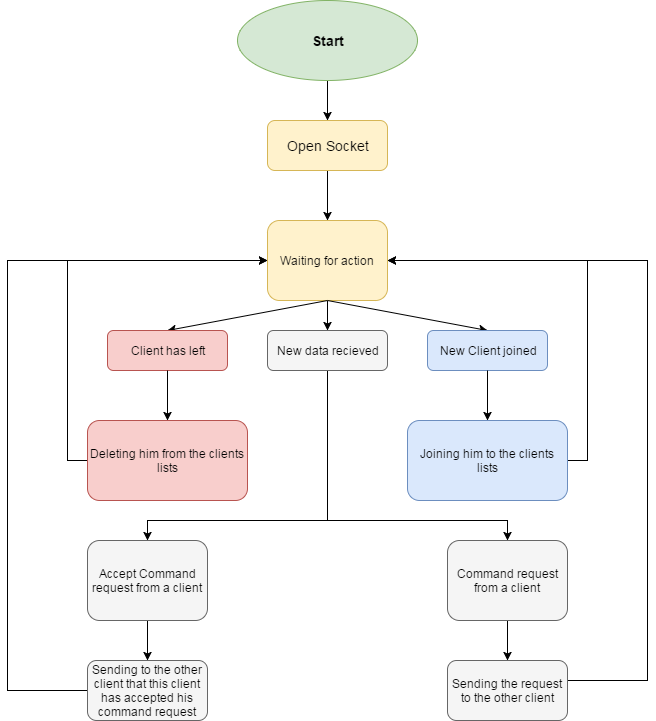
תיאור מבנה המערכת ופירוט המודולים השונים בה

2.1 מבט על

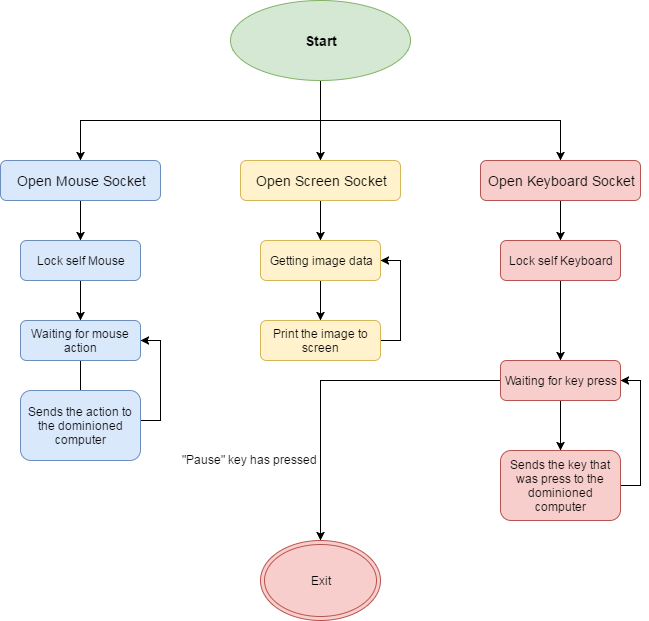
**צד לקוח:**



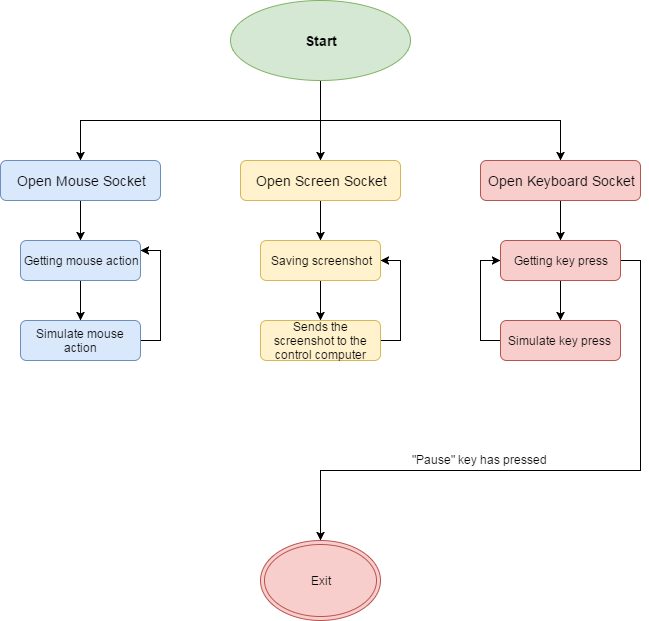
צד שרת:



מחשב שולט (Command):



מחשב נשלט (Dominion):



2.2 פירוט רכיבי המערכת

רכיבים עיקריים - צד שרת

* open\_client (server\_socket, open\_client\_sockets, nickpass\_socket, name\_list)

**פותח חיבור עם לקוח חדש ומצרף אותו לכל הרשימות של הלקוחות המחוברים.**

* close\_client(socket\_to\_delete, open\_client\_sockets, nickpass\_socket, name\_list)

**סוגר חיבור עם לקוח קיים ומוחק אותו מכל הרשימות של הלקוחות המחוברים.**

רכיבים עיקריים - צד לקוח

* is\_valid(name, password)

**מקבל שם וסיסמה ובודק אם האורך שלהם מתאים**

* resize\_image(name, width, height)

**מקבל תמונה ומשנה את גודל התמונה לאורך ולרוחב שהוא מקבל**

* login\_window(window, my\_socket)

**פותח את מסך הפתיחה וקולט שם וסיסמה**

* print\_connected\_list(data, us\_nickname, nickname\_lines, number\_of\_lines)

**מדפיס את רשימת הלקוחות המחוברים למסך**

* main\_screen(window, us\_nickname, us\_password, my\_socket)

**פותח את המסך הראשי ומאפשר השתלטות בין מחשבים**

רכיבים עיקריים - מחשב שולט (Command) ומחשב נשלט (Dominion)

* keyboard(procees\_list)

**פותח סוקט שאחראי על התקשורת של המקלדת**

* mouse()

**פותח סוקט שאחראי על התקשורת של העכבר**

* screen()

**פותח סוקט שאחראי על התקשורת של המסך**

2.3 דיון בנושא העיצוב הנבחר

**חלוקת רכיבים**

**הפרויקט מורכב מ 4 תכניות: שרת, לקוח, תכנית משתלטת (**Command**) ותכנית נשלטת (**Dominion**). המודל הוא מודל של שרת מרובה לקוחות כאשר כל לקוח שמתחבר לשרת נכנס לרשת המשותפת בין הלקוחות. כאשר לקוח מסוים מבקש להשתלט על לקוח אחר והלקוח השני מסכים, כל אחד מהם, בהתאם, קורא לתכנית המתאימה:**Command **או** Dominion**. כל אחת מהן היא תכנית בפני עצמה אשר מקבלת את ה-**IP **של המחשב השני ומתחילה בהשתלטות.**

מודולים עיקריים:

צד שרת

* Socket - מודול המאפשר פתיחת קשר בין שני מכשירים שונים דרך הרשת.
* Select - מודול המאפשר סנכרון בין שרת למספר לקוחות בו זמנית.

צד לקוח

* + Graphic - מודול המאפשר יצירת ממשק גרפי למשתמש (GUI).
* msvcrt - מודול המאפשר קבלת מידע שהמשתמש מזין למקלדת (לחיצת מקש).
* Socket - מודול המאפשר פתיחת קשר בין שני מכשירים שונים דרך הרשת.
* Select - מודול המאפשר סנכרון בין שרת למספר לקוחות בו זמנית.

מחשב שולט (Command) ומחשב נשלט (Dominion)

* pyHook - מודול המאפשר שליטה ובקרה בעכבר ובמקלדת.
* subProcess - מודול המאפשר פתיחת תהליך חדש (Process).
* PIL - מודול המאפשר עיסוק בתמונות בפייטון.
* Socket - מודול המאפשר פתיחת קשר בין שני מכשירים שונים דרך הרשת.
* Sys - מודול המאפשר ניהול של פרמטרים בתכנית.
* win32api - מודול המאפשר שליטה בעכבר ובמקלדת.
* Cv2 - מודול המאפשר עיסוק ושליטה בתמונות והצגתם למסך.
* Python

**בחרתי בשפת התכנות** Python **מכיוון שהיא נוחה לשימוש, ומוכרת לי מכיוון שאני כבר שנתיים מתכנת בשפה זו. בנוסף ישנו מודול** TKinter **שהוא מודול של גרפיקה (**GUI**) בשפת פייטון, ובעזרת מודול נוסף (**Graphics.py**) ניתן לעבוד עם המודול** TKinter **בקלות ובנוחות.**

* CV2

**בחרתי במודול** CV2 **מכיוון שהוא פשוט ונוח לשימוש, בנוסף דרכו הצלחתי להציג תמונה בדרך המתאימה ביותר לפרויקט שלי. המודול מובן ופשוט יחסית ללמידה, לכן בחרתי בו על פני מודלים אחרים.**

* Pyhook

**המודול** Pyhook **הינו מודול המשמש לתקשורת ושליטה בעכבר ובמקלדת, הוא המודול שרוב המקורות המליצו עליו ולכן בחרתי בו.**

1. עיצוב נתונים ופרוטוקולים

תיעוד של מבני נתונים שונים / פרוטוקולים המשמשים אותנו במערכת

**טיפוסי נתונים:**

* **חלון (GraphWin) - אובייקט מהמחלקה grapichs.py אשר מציג למסך חלון (Window).**
* **תמונה - אובייקט מהמחלקה grapichs.py אשר מאפשר להציג תמונה לאובייקט מסוג GraphWin (מציג את התמונה לחלון).**
* **קו - אובייקט מהמחלקה grapichs.py אשר מאפשר להציג קו לאובייקט מסוג GraphWin (מציג את הקו לחלון).**
* **תיבת טקסט - אובייקט מהמחלקה grapichs.py אשר מאפשר להציג תיבת טקסט לאובייקט מסוג GraphWin (מציג את תיבת הטקסט לחלון).**
* **טקסט רגיל - אובייקט מהמחלקה grapichs.py אשר מאפשר להציג טקסט לאובייקט מסוג GraphWin (מציג את הטקסט לחלון).**

פרוטוקולים**:**

* TCP/IP - בעלת 2 שכבות. השכבה העליונה, פרוטוקול בקרת שליחה (TCP) , מנהלת את חלוקת הקובץ שהוא ההודעה ל-packets קטנים שנשלחים למחשב השני ואוספת מחדש את ה-packets עד לקבלת ההודעה המקורית. השכבה התחתונה, פרוטוקול אינטרנט (IP) , מטפלת בכתובת אליו נשלח כל packet כדי שיגיע ליעד הנכון.  **מאפשר העברת נתונים אמינה יותר אך איטית יותר.**

השתמשתי בפרוטוקול זה כדי להעביר את תמונת המסך מכיוון שבחלק זה עדיפה אמינות ואיכות ההעברה על פני המהירות. (המהירות גם היא חשובה אך אם תגיע תמונה פגומה אז המחשב לא יוכל להציג אותה כנדרש).

* UDP/IP - בעלת 2 שכבות. השכבה העליונה, ה-UDP אינו מספק אמינות או שימור סדר כפי שקורה ב .[TCP](https://he.wikipedia.org/wiki/TCP)

[חבילות מידע](https://he.wikipedia.org/wiki/%D7%97%D7%91%D7%99%D7%9C%D7%AA_%D7%9E%D7%99%D7%93%D7%A2) עשויות להגיע בסדר שונה מזה שבו הן נשלחו,  להגיע מספר פעמים או ללכת לאיבוד ולא להגיע כלל. השכבה התחתונה, פרוטוקול אינטרנט (IP), מטפלת בכתובת אליו נשלח כל packet כדי שיגיע ליעד הנכון.

השתמשתי בפרוטוקול זה כדי להעביר את מצב העכבר והמקלדת מכיוון שבחלקים אלו עדיפה מהירות ההעברה ולא האיכות. (בכל תזוזה של קטנה של העכבר מועבר מידע חדש לגבי מיקומו ולכן לא קריטי אם אחד המיקומים לא יעבור כמו שצריך).

4. ממשק משתמש

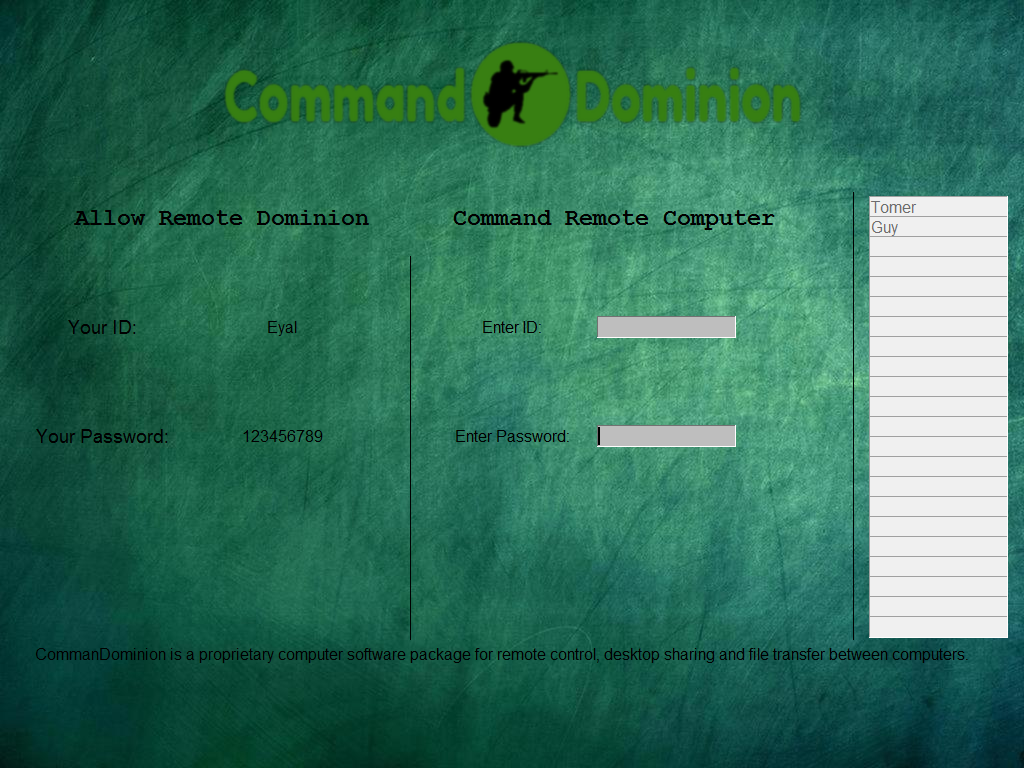
פירוט ממשקי המשתמשים וכן תופיע הסקיצה עבורם.

מסך הפתיחה - אחרי מסך זה (במידה והשם והסיסמה תקינים) המשתמש עובר למסך הבא.



תיבת הסיסמה - פה המשתמש מכניס את הסיסמה שלו

תיבת השם - פה המשתמש מכניס את השם שלו



רשימת הלקוחות המחוברים

השם שלי

תיבת השם - פה המשתמש מכניס את השם של הלקוח שהוא רוצה להשתלט עליו

הסיסמא שלי

תיבת הסיסמה - פה המשתמש מכניס את הסיסמה של הלקוח שהוא רוצה להשתלט עליו