# בית ספר הרצוג כפר סבא | הזמנת חולצות ביה"ס**שער הפרויקט**

בית ספר: הרצוג

שם העבודה: multi-control

שם התלמיד: איתי שפרנט

ת.ז:325751386

שם המנחה: יוסי זהבי

שם החלופה: multi-control

תאריך הגשה: 9 למאי 2021.

תוכן עניינים

[**שער הפרויקט** 1](#_Toc72681150)

[**מבוא** 3](#_Toc72681151)

[הרקע לפרויקט 3](#_Toc72681152)

[תהליך המחקר 3](#_Toc72681153)

[סקירת המצב הקיים בשוק 3](#_Toc72681154)

[החידושים בפרויקט 4](#_Toc72681155)

[אתגרים מרכזיים 4](#_Toc72681156)

[בעיות מרכזיות 4](#_Toc72681157)

[הסיבה לבחירה בנושא 4](#_Toc72681158)

[מוטיבציה לעבודה 4](#_Toc72681159)

[על איזה צורך עונה הפרויקט. 4](#_Toc72681160)

[הצגת פתרונות 4](#_Toc72681161)

[**מבנה הפרויקט** 5](#_Toc72681162)

[הצגת פתרון מוצע – הסבר טכני 5](#_Toc72681163)

[מחלקות הפרויקט 7](#_Toc72681164)

[מודולים 9](#_Toc72681165)

[פעולות חיצוניות 10](#_Toc72681166)

[מבני נתונים ומשתנים גלובליים 11](#_Toc72681167)

[זרימת מידע 11](#_Toc72681168)

[משתנים גלובליים 11](#_Toc72681169)

[לולאות רשאיות 12](#_Toc72681170)

[תיאור תפריטים 13](#_Toc72681171)

[טכנולוגיות בהם נעשה שימוש 13](#_Toc72681172)

[קבועים חשובים 14](#_Toc72681173)

[פרוטוקול תקשורת 15](#_Toc72681174)

[הצפנות 18](#_Toc72681175)

[אלגוריתמים מרכזיים 19](#_Toc72681176)

[**מדריך למשתמש** 20](#_Toc72681177)

[**בסיס נתונים** 24](#_Toc72681178)

[רשימת העמודות 24](#_Toc72681179)

[שאילתות SQL 25](#_Toc72681180)

[**מדריך למפתח** 26](#_Toc72681181)

[הוראות הפעלה 27](#_Toc72681182)

[רפלקציה 28](#_Toc72681183)

# **מבוא**

## הרקע לפרויקט

כאשר באתי לבחור פרויקט אותו אממש בתור פרויקט סיום לא היה לי רעיון שידעתי אותו ארצה לממש. בזמן שעברנו על בשיעור ביחד עם המורה על פרויקטים משנים קודמות הוא הציע בתור רעיון לפרויקט תוכנה שתאפשר הורדה של תוכנה מסוימת במספר מחשבים בו זמנית. די התלהבתי מהרעיון והחלטתי לפתח אותו יותר לכך שתהיה אפליקציה שתאפשר למשתמש לבצע את כל הפעולות שהוא עושה על המחשב שלו על עוד כמה מחשבים זהים לו במקביל. אני חושב שלאפליקציה מסוג זה יכול להיות שימוש אמיתי במקומות בהם יש חדרי מחשבים ענקיים ורוצים לבצע איזושהי פעולה על כולם. לדוגמה בהינתן מצב בו אנו רוצים להשתיק את הרמקולים בכל המחשבים בחדר המחשבים במקום לעשות את הפעולה על כל מחשב יהיה ניתן להתחבר לאפליקציה דרך מחשב אחד שהוא השולט ודרך כך לבצע את הפעולה רק פעם אחת.

## תהליך המחקר

את תהליך המחקר שלי התחלתי תחילה מלחפש אחר תוכנה דומה המאפשרת להעביר את הקלט מהעכבר והמקלדת ממחשב אחד אל מחשבים אחרים אך לא מצאתי אחת כזאת. מצאתי מספר תוכנות דומות המאפשרות לשלוט על שני מסכי מחשב באמצעות עכבר אחד ומקלדת אחת. לאחר שלא מצאתי תוכנה שמבצעת משהו דומה למה שאני רוצה לממש התחלתי לחקור על דרכים בהם אוכל להעביר קלט ממחשב אחד למחשבים אחרים. מצאתי מספר דרכים שבאמצעותם יהיה ניתן להעביר את הקלט בצורה טובה, על ידי שימוש בספריות שעוזרות לעקוב אחר הקלט מהמקלדת והעכבר.

### סקירת המצב הקיים בשוק

כאמור בשוק כיום לא מצאתי אפליקציה שיש בה את אותם הפונקציות שיש בפרויקט זה. אומנם מצאתי כמה תוכנות שמאפשרות שליטה על כמה מחשבים באמצעות סט אחד של עכבר ומקלדת. אומנם הפונקציונליות של התוכנות שונה מין הפרויקט שלי אך הבסיס סובב סביב אותו רעיון של שליטה על מספר מחשבים בו זמנית. דוגמה לתוכנה כזו היא “Mouse With Out Borders” של חברת Microsoft מאפשר לחבר מספר מחשבים עם מקלדת אחת ועכבר אחת ולהעביר את העכבר ממסך מחשב אחד למסך מחשב שני מבלי להעביר את החיבור הפיזי של העכבר ממחשב אחד לשני. תוכנה זו טובה במקרה שיש לכם שני מחשבים שאתם רוצים לעבוד על שניהם במקביל ולבצע עליהם פעולות שונות. בנוסף ישנם בשוק מספר תוכנות שמאפשרות התקנת מערכת הפעלה על מספר רב של מחשבים במקביל. בנוסף ניתן להשוות את הפרויקט לתוכנת TeamViwer שמאפשרת שליטה מרחוק על מחשב אחד.

<https://www.teamviewer.com/en/>

### החידושים בפרויקט

החדשנות בפרויקט היא הפונקציה המאפשרת לבצע את אותה פעולה במספר רב של מחשבים. לא מצאתי אפליקציה שמאפשרת את הפונקציה הזו. האפשרות לבצע פעולות על מספר רב של מחשבים היא בעלת ערך רב עבור אנשים רבים. הפונקציה הזו יכולה לחסוך המון זמן לאנשים שעובדים בחדרי מחשבים גדולים ורוצים להתקין תוכנה חדשה למחשב או לשנות הגדרות בכל המחשבים.

## אתגרים מרכזיים

### בעיות מרכזיות

במהלך כתיבת הפרויקט התמודדתי עם מספר קשיים. אחד המרכזיים הוא הרצה של הפרויקט מכיוון שעל מנת לבחון את הביצוע של הפרויקט יש להשתמש ביותר ממחשב אחד לכן כל הרצה לפעמים גזלה קצת יותר זמן מכיוון ומדובר בעבודה על שני מחשבים או יותר. בעקבות השנה הלא רגילה שחווינו יצא לי להריץ את הפרויקט בבית הספר פעמים מעטות, דבר זה היווה קושי, מכיוון שחדר מחשבים הוא המקום שאליו מיעודת התוכנה ובבית לעיתים היה יותר קשה לבחון את ביצועיה.

### הסיבה לבחירה בנושא

בחרתי לעשות פרויקט בתחום שעוסק בתקשורת בין מספר רב של מחשבים מכיוון שהתחום מאוד מעניין אותי אני חושב שיש הרבה מאוד אופציות ונושאים שונים שאפשר לבנות סביבם פרויקטים. אני חושב שיש מקום גדול לשפר את הנוחות בעבודה על כמה מחשבים והפרויקט הזה זו אחת הדרכים בהם אפשר להפוך את העבודה על מספר מחשבים לנוחה הרבה יותר

### מוטיבציה לעבודה

המוטיבציה שלי לעבוד על הפרויקט נובעת מכך שאני חושב שהפרויקט יכול להיות שימושי להרבה אנשים, התחום שבו עוסק הפרויקט מעניין אותי מאוד. לכן יש לי הרבה מוטיבציה לעבוד על הפרויקט ולגרום לו לעבוד על הצד הטוב ביותר.

### על איזה צורך עונה הפרויקט.

הפרויקט עונה על הצורך של אנשים שעובדים עם מספר רב של מחשבים לבצע פעולות על כל המחשבים בו זמנית בלי צורך לבזבז זמן על לעשות את אותה הפעולה על כל מחשב בנפרד. הפתרון שהפרויקט מציע הוא לחבר את כל המחשבים דרך אפליקציה שתאפשר למחשב אחד לשלוט על כל האחרים וכל הפעולות שיתבצעו על המחשב השולט יתבצעו בכל המחשבים האחרים.

## הצגת פתרונות

הפתרון שמומש יוצא מתוך נקודת הנחה שהמחשבים פועלים במערכות הפעלה זהות ופועל באופן אידיאלי כאשר שולחן העבודה זהה בכל המחשבים. על המחשב שמתחבר כשולט מופעל key logger שמתעד כל לחיצה על המקלדת. כל הקשה מועברת לכל המחשבים האחרים ומבוצעת בהם. כך גם ההקלקות על העכבר מועברות לכל המחשבים. באמצעות העברת הקלט של העכבר והמקלדת ניתן למעשה לשלוט על המחשבים ולבצע בהם מגוון רחב של פעולות מהורדת תוכנות ועד לשינוי הגדרות ולכל דבר שנרצה לקבוע בכל המחשבים עליהם אנו שולטים.

# **מבנה הפרויקט**

## הצגת פתרון מוצע – הסבר טכני

כאמור מטרת הפרויקט היא לאפשר ביצוע פעולות שונות על מספר מחשבים בו זמנית. הפתרון שנבחר הוא העברת הקלט ממחשב שולט דרך שרת מרכזי אל עבר הלקוחות הנשלטים.

על מנת לבצע את הפתרון נחלק את הפרויקט לשלוש יחידות שונות.

יחידה ראשונה זוהי השרת המרכזי. השרת ינהל רשימה של לקוחות שולטים שמהם הלקוחות שמתחברים בתור נשלטים יוכלו לבחור את המחשב שאמור לשלוט עליהם על פי השם שלהם. השרת יראה ללקוח שולט כמה לקוחות התחברו אליו. כאשר לקוח שולט מתחיל את השליטה השרת המרכזי יעביר את הקלט אל כל המחשבים שהתחברו אליו ספציפית. השרת גם אומר לכל לקוח שולט מתי עליו לשתף מסך. הלקוח השולט מקבל כל פעם מסך של user אחר ויכול להחליף מתי שהוא רוצה. תפקיד השרת הוא להגיד לכל user מתי עליו לשתף מסך. השרת מקבל גם לקוחות מסוג waiting לקוחות אלו שייכים למשתמשים מסוג user ומתחברים לשרת אוטומטית כאשר user מתנתק. בשרת קיימת רשימה גלובלית שבה נמצאים כתובות הIP של כל הusers שadmins מבקשים מהם להתחבר כאשר נמצאת התאמה השרת שולח לwaiting client שעליו לחבר את הuser. וכך למעשה ניתן לקרוא ללקוחות שיחזרו.

יחידה שנייה היא הלקוח השולט (admin) כאמור הלקוח השולט יתחבר לחשבון שלו באמצעות שם וסיסמה שאותה יצטרך להכניס user על מנת להישלט על ידי admin מסויים. השרת יוסיף את הלקוח לרשימת הadmins. הadmin מחליט מתי להתחיל את השליטה על המחשבים ומתי להפסיק אותה. במהלך השליטה הלקוח ישלח את כל הקלטים שבוצעו על ידי המשתמש באמצעות המקלדת והעכבר. בפני הadmin יוצג ממשק גרפי אשר יאפשר לו: לראות כמה לקוחות התחברו אליו, להתחיל את השליטה עליהם ולהפסיק אותה. זיהוי הקלט יתבצע על ידי תוכנת keylogger שתרוץ כאשר הadmin יחליט להתחיל שליטה התוכנה תתעד את הקשות המקלדת ואת מיקומם של לחיצות העכבר ביחס לגודל המסך. בנוסף על כך בצד הadmin ירוץ שרת שיקבל תמונת מסך מuser אחד כאשר ירצה הadmin להחליף הוא יקבל תמונה ממשתמש אחר. בנוסף לפני שהadmin מתחיל שליטה הוא יכול לקרוא לכל הלקוחות שהתחברו אליו בפעם האחרונה.

יחידה שלישית היא הלקוח הנשלט. כאמור הלקוח הנשלט יבחר מתוך רשימת המחשבים השולטים את המחשב שאמור לשלוט עליו ויכניס את הסיסמא אותה קבע הadmin. לאחר ההתחברות הלקוח יחכה לתחילת שליחת הקלט של הadmin. הקלט יועבר אליו דרך השרת המרכזי. תפקידו של הלקוח הוא לבצע את הפעולות שביצע הadmin. הוא מקבל מהשרת את הפעולות שביצע הadmin ומבצע אותם על המחשב עצמו. בנוסף תפקידו הוא בעת שהשרת אומר לו שעכשיו תורו לשלוח תמונות מסך לadmin, לפתוח thread חדש שיתחבר לשרת שמחזיק הadmin שלו וישלח לו תמונות מסך. כאשר הuser מתנתק מהשרת, כלומר כשמסתיים סשן הלקוח מפעיל לקוח אחר waitin\_client שמתחבר לשרת ומחכה שהשרת יגיד לו שהוא צריך לחבר את הuser שוב, כלומר הadmin שאליו הוא התחבר בפעם האחרונה רוצה שהוא יתחבר שוב. התוכנה של הwaiting client נרשמת בRegistey אוטומטית כך שכאשר המחשב נדלק הwaiting client יתחבר לשרת והadmin האחרון שהוא התחבר אליו יוכל לקרוא לו.



## מחלקות הפרויקט

|  |  |  |  |
| --- | --- | --- | --- |
| **שם המחלקה** | **תכונות** | **פעולות** | **יחידה** |
| admin | * Name- שם האדמין * Password- סיסמא שצריך להכניס על מנת להיכנס לשיטה על ידי האדמין * Client\_ connected (list)- רשימה שמכילה את הobjects של הלקוחות שעליהם שולט המחשב * Buffer- מכיל את רשימת הפעולות שביצע הadmin * Status- האם ניתן לחבר עוד משתמשים * sock * Session\_id * Admins\_lock- משמש להעברה בטוחה של מידע בין הbuffer של האדמין לזה של היוזר. * all\_actions (list)- שומר את כל הפעולות שביצע האדמין (לחיצות עכבר ומקלדת) | * Add\_to\_buffer(string)- מקבלת רשימת פעולות שביצע הלקוח ומוסיפה אותה לbuffer * Start\_session()- מתחיל את הסשן * Update\_clients\_buffer()- מוסיף לbuffer של כל הלקוחות את הbuffer הנוכחי. * End\_session()- מסיים את הסשן. * Show\_client\_connected()- תראה לadmin את מספר הלקוחות שהתחברו * disconnect()- תנתק את הadmin ואת כל הלקוחות שהיו מחוברים אליו. * Share\_up/share\_down- כאשר הadmin רוצה לשנות את המסך של המשתמש שהוא צופה אחת מן הפעולות תיקרא בהתאם לחץ עליו לחץ * Save\_to\_ip\_sql- שומר את כתובות הIP של הusers שהתחברו * Get\_last\_users- מחזיר את כתובות ה IP של הusers שהתחברו בפעם האחרונה | שרת |
| User | * Name- שם הuser * Address- כתובת IP של הuser * Admin- object מסוג admin שאליו משויך הuser. יאותחל לnone. * Sock-socket של הuser. * Buffer- יכיל את הפקודות שהuser צריך לבצע. * User\_lock- נעילה שתפקידה להבטיח ששהמחלקה תהיה thread safe | * Send\_buffer()- ישלח לuser את הbuffer שמכיל את הפקודות שעליו * לבצעAdd\_admin(name,password)- יוסיף את הuser לרשימת הנשלטים על ידי הadmin. * Reset\_buffer- מאפס את הbuffer של הuser. | שרת |
| ManageSQL | * Db\_file- שם קובץ הנתונים * Table\_name-שם הטבלה * Conn- החיבור לטבלת הנתונים * Cruser- אובייקט שעוזר לנו לבצע פעולות על הטבלה | * open\_db- יוצר קשר עם בסיס הנתנונים * Close\_db- סוגר את החיבור עם בסיס הנתונים * Commit- מעדכן את בסיס הנתונים * Create\_table- מקבל sql command ויוצר טבלה חדשה. * Drop\_table- מוחק טבלה מבסיס הנתונים | שרת |
| Action\_class | מכילה שלוש תת מחלקות  KeyPress   * Key- המקש שנלחץ * Is\_press- האם המקש נלחץ או שוחרר   MouseClick   * X- המיקום האופקי של הלחיצה על המסך * Y- המיקום האנכי של הלחיצה על המסך * Is\_right- האם ללחוץ על הלחצן הימני של העכבר או השמאלי. * Is\_release- האם זוהי לחיצה או עזיבה של העכבר   MouseClick   * X- המיקום האופקי של העכבר בעת הגלילה * Y- המיקום האנכי של העכבר בעת הגלילה * Dx- כמה יש לגלול בציר האופקי * Dy- כמה יש לגלול בציר האנכי | לכל שלוש תת המחלקות קיימת פעולה אחת.   * Commit- מבצע את הפעולה שדרושה מהמחלקה. לדוגמא KeyPress.commit()- יקיש על המקש המתאים בהתאם לתכונות העצם   MouseClick.commit()- יקליק במקום המתאים בהתאם לתכונות של העצם. | user |
| Recorder | * Mouse\_listener-אובייקט מספריית pynput, שאחראי להקליט את לחיצות העכבר והגלילות * Keyboard\_listener- אובייקט מספריית pynput, שאחראי להקליט את הקשות המקלדת. * Lock- נעילה שנועדה למנוע התנגשות בין הthread של העכבר וthread של המקלדת בהוספה לbuffer/ * Buffer (list)- שומר את כל הפעולות שהוקלטו. * Screen\_x- הגודל האופקי של המסך * Screen\_y- הגודל האנכי של המסך | * Start\_recored- מאתחל את הlisteners * Stop\_recored- מפסיק את הlisteners * Read\_buffer- מרוקן את הbuffer ומחזיר את התכולה שלו. | admin |
| AESCipher | * Bs (block\_size)- המידע המוצפן מתחלק לבלוקים. מספר זה קובע את גודלם * Key- המפתח המשמש להצפנה ופיענוח של המידע באמצעות AES. | * Encrypt- מקבל מידע. על מנת שיהיה ניתן להצפין את המידע עם AES מוסיפים pad (ריפוד) על מנת שיהיה ניתן לחלק את המידע לבלוקים שווים. מיצרים מידע רנדומלי בגודל של בלוק וקובעים אותו כ iv (initial vector) שנחות על מנת לבצע הצפנת בלוקים. מצפינים את המידע עם הריפוד האמצעות המפתח והiv. למידע המוצפן מוסיפים את הiv את כל המידע הזה מקודדים עם base64 ומחזירים. * Decrypt- מקבל מידע מוצפן שקודד באמצעות base64. ראשית מבצעים decode לbase64. לאחר מכן מוצאים מהמידע שהתקבל את הiv שאורכו הוא כאורך בלוק (bs). לאחר מכן מפענחים את המידע המוצפן באמצעות הiv והמפתח מפענחים את המידע המוצפן מוציאים את הריפוד ומחזירים מחרוזת עם מידע מפוענח. * Pad- מחשבים כמה בייטים חסרים על מנת שיהיה ניתן לבצע הצפנת בלוקים. לאחר מכן מוספים למחרוזת את מספר הביטיים שחסרים וכל בייט רשום המספר הזה בASCII. כלומר אם חסר בייט אחד התו האחרון המחרוזת יראה כך ‘0x1’. * Unpad- קוראים את התו האחרון ומורידם מהמחרוזת את מספר הבייטים שרשום שם מהסוף. | User/ admin/ server. |

## מודולים

1. Main\_Server.py- הקובץ המרכזי של השרת
2. Admins\_sql.py- משמש את השרת בתפעול טבלת הSQL שמכילה את השמות והסיסמאות של הadmin.
3. Users\_ip\_sql.py- משמש את השרת בתפעול טבלת הSQL ששומרת את כתובות הIP של היוזרים האחרונים של כל אדמין.
4. Admin\_class.py- מכיל את המחלקה admin (פירוט בטבלה).
5. User\_class.py- מכיל את המחלקה user (פירוט בטבלה)
6. sql\_manage.py- מכיל את המחלקה ManageSQL (פירוט בטבלה)
7. Admin\_client.py- הקובץ המרכזי של לקוח מסוג admin.
8. Admin\_gui.py- יוצר את התפריט הראשי של משתמש מסוג admin.
9. Key\_mouse\_logger.py- מכיל את המחלקה Recorder (פירוט למעלה).
10. Screen\_share\_admin.py- זהו השרת אשר מרים הלקוח כל מנת לקבל תמונות מסך מהיוזרים
11. User\_client.py- הקובץ המרכזי של יחידת הuser.
12. Action\_class.py- מכיל את המחלקות האחראיות על ביצוע הקלטים במחשבים הנשלטים
13. Screen\_share\_user.py- משמש על מנת להתחבר לשרת שמחזיק הadmin ושולח תמונות מסך
14. Waiting\_client.py- זהו הלקוח שמופעל אוטומטית כאשר לקוח מסיים את תפקידו.
15. AES\_class.py- מכיל את המחלקה AESCipher (פירוט למעלה) שתפקידה להצפין מידע. משמשת את כל שלושת היחידות
16. Tcp\_by\_size- משמש את כל היחידות, בכדי לבצע תקשורת בין מחשבים על פי פרוטוקול התקשורת (פירוט למטה).

## פעולות חיצוניות

פירוט הפעולות החיצוניות על פי יחידות הפרויקט השונות.

Admin

1. Log\_in()- מראה את התפריט של הadmin להתחברות. מסיים את תפקידו כאשר הadmin יכול להתחיל סשן. נמצא בקובץ admin\_gui.py
2. Start\_server()- מרים את הסרבר שמקבל את תמונות המסך מהיוזרים המחוברים. כאשר יש יותר מלקוח אחד כלומר תורו של לקוח אחר לשתף מסך השרת מנתק את הלקוח שמחובר כעת ומציג את תמונת המסך של הלקוח שהתחבר. נמצא בקובץ screen\_share\_admin.py
3. Start\_recored()- פעולה זו תיקרא כאשר המשתמש יחליט להתחיל שליטה. בפעולה זו נשתמש בפונקציה listener() של המודול pyinput. פונקציה זו מאפשרת לנו לקשר בין אירוע של לחיצת עכבר או הקשת מקלדת לפונקציה אחרת באופן א-סינכרוני. הפונקציות שיקראו במקרה של לחיצה או הקשה יוסיפו את האירוע לbuffer.מיקום של לחיצת עכבר תיקבע באופן יחסי לגודל המסך. על מנת למצוא את גודל המסך יש להשתמש במודול pyautogui. פעולה זו נמצאת בקובץ key\_mouse\_logger.py
4. Send\_buffer()- כל פעם שהbuffer לא ריק הפונקציה תשלח אותו ותרוקן אותו. נמצא בקובץ Admin\_client.py
5. Stop\_control()- פעולה זו תשלח הודעה להפסקת השליטה ותפסיק את ההקלטה של הקלטים. נמצא בקובץ Admin\_client.py
6. Generate\_aes\_key()- מייצר מפתח AES סימטרי בשרת ובלקוח

User

1. Get\_admin\_nameיראה את התפריט של הuser ויחזיר את בחירת המשתמש. תראה את רשימת הadmins המחוברים ותחזיר את השם של הadmin
2. Connect\_to\_host\_(name)- תשלח בקשת התחברות לadmin עם סיסמא שיכניס המשתמש.
3. Recv\_operations()- מקבלת operations מהשרת באמצעות socket ומוסיפה אותן לbuffer.
4. Commit\_operations()- מבצעת את הפעולות שנמצאות בbuffer נשתמש במודול pyautogui על מנת לבצע את הפעולות יש לזכור שהמיקומים של לחיצות העכבר הם ביחס לגודל המסך
5. Register\_waiting\_client- רושם את הwaiting client לRegistery וגורם לכך שירוץ אוטומטית כל פעם שהמחשב עושה boot.
6. Share\_screen()- מתחבר לשרת ושולח תמונות מסך
7. Waiting\_client\_main()- מתחבר לשרת ומחכה שהשרת יגיד לו להפעיל את הUser client.

שרת

1. Handle\_admin()- מחבר אדמין ומטפל בבקשות שלו מעביר את הקלטים שהוא שולח לכל הusers שהתחברו אליו ואחראי על התקשורת איתו
2. Handle\_user- מחבר לקוח מסוג user ומטפל בבקשות שלו מעביר אליו את הקלטים בעת הצורך ואחראי על התקשורת איתו
3. Add\_to\_list(name)- מוסיף את האדמין לרשימה מחזיר האם הצליח, לא יכול להיות שם שכבר נמצא ברשימה.
4. Remove\_from\_table(name)- מקבל שם של admin ומוציא אותו מהרשימה.
5. Send\_list()-שולחת את רשימת הadmins לuser.
6. Connect\_user(admin,user)- מוסיפה את הuser לרשימת הנשלטים של הadmin

## מבני נתונים ומשתנים גלובליים

Admins\_list- מבנה נתונים חשוב בפרויקט הוא רשימת הadmins.רשימה זו מכילה objects מסוג admin שכעת מחוברים. רשימה זו משמשת אותנו על מנת לחבר users לadmins שמתאימים להם. הרשימה היא גלובלית ומתעדכנת בהתאם להתחברות והתנתקות של משתמשים

Buffers- בפרויקט יש כמה bufferים שונים. כל buffer הינו מבנה נתונים מסוג queue. נבחר מבנה נתונים מסוג זה מכיוון שהbufferים הינם מאוד דינמיים וחשוב שהפעולות שנכנסות אליהם יישארו בסדר מתאים. תור יאפשר עבודה נוחה עם התעסקות מינמלית. בפרויקט ישנם כמה bufferים שונים ויש להבדיל ביניהם

Admin\_side\_buffer- לתוך התור הזה יכנסו הקשות המקלדת ולחיצות העכבר שתועדו אצל הadmin. הbuffer נשלח לשרת כל עוד איננו ריק. מאוכסן בצד לקוח.

User\_side\_buffer- לתוך הbuffer הזה נכנסים הפעולות שהגיעו מהשרת וממתינות לביצוע של הuser. מאוחסן בצד הלקוח

Server\_admin\_buffer- הbuffer של הadmin בצד של השרת הפעולות שתועדו ונכנסו לתור בצד של הלקוח כעת נכנסים לתור בצד של השרת מאוחסן בצד של השרת במחלקת admin.

Server\_user\_buffer- הbuffer של הadmin בצד של השרת מעביר את הפעולות לbufferים של כל אחד מהusers שמחוברים אליו. הbuffer הזה נשלח לuser כל עוד אינו ריק. מאוחסן בצד השרת בתוך מחלקת user.

על מנת לקבוע מתי admin צריך לשלוח תמונת מסך נשמור את כמות הפעולות שעשה מאז שליחת תמונת מסך האחרונה בתוך משתנה amount\_of\_actions. ואת הזמן שבה צולמה תמונת המסך הקודמת בתוך המשתנה last\_screenshot\_time.

CLIENTS\_TO\_CONNECT- רשימה של כתובות IP שadmins ביקשו שיתחברו.

REALESE\_CLIENTS- רשימה של כתובות IP שלא צריכים לחכות שהשרת יקרא להם.

Connected\_list- כל admin מחזיק ברשימה כזו שיש בה אובייקטים מסוג יוזר כל אוביירט שייך ליוזר שהתחבר

## זרימת מידע

המידע המועבר בפרויקט הוא הקלטים (הקשות עכבר ומקלדת) שמקליט האדמין ושם אותם בbuffer שלו במחלקת Recorder. את המידע הזה שולחים לשרת ששם את זה בbuffer של העצם של הadmin בthread שלו בשרת שמעדכן את הthreads של הusers שמחוברים לאותו admin. לאחר שהbuffer של הusers בשרת קיבל את המידע, הוא נשלח לuserים עצמם.

## משתנים גלובליים

שרת

* נעילות: נעילה לרשימת הadmins ונעילה לרשימות של הIP

Admin

WINDOW\_CLOSED- האם נסגר החלון שמציג את התפריט של הadmin

RECORED\_ENDED- האם נגמרה ההקלטה של המקלדת והעכבר

PRESSED\_START- האם התחיל סשן.

FINISHED\_LOGIN- האם המשתמש עבר את המסך של הLog in.

ERROR\_LABEL- איזה סוג שגיאה התרחשה בזמן ההתחברות לשרת

OLD\_USER- האם המשתמש ישן או חדש

נעילות בשרת של שיתוף המסך running\_lock- רק משתמש אחד יכול לשתף מסך בכל פעם. num\_lock למנוע התנגשות של threads בעדכון מספר הלקוחות. close\_lock נועד למנוע התנגשות בבדיקה האם לסגור את השרת.

NUM\_CLIENTS- בקובץ של הadmin\_client.py המשתנה מייצג את מספר הusers שהתחברו לadmin בקובץ של שיתוף המסך משמעותו כמה לקוחות מחוברים כעת לשרת.

CLOSE\_SERVER- האם צריך לסגור את השרת

Gui\_lock- נועד להפוך את העבודה עם הGUI לthread safe.

User

WINDOW\_CLOSED- האם נסגר החלון שמציג את התפריט של הadmin

FINISHED\_CONNECTING- האם המשתמש עבר את המסך של ההתחברות.

SCREEN\_SHARE- האם על הuser לשתף מסך.

Name,admins\_name\_password- פרטי ההתחברות של המשתמש

WIDTH, HIGHT- הגודל של המסך עליו רץ הuser

## לולאות רשאיות

בפרויקט ישנם מספר לולאות רשאיות. בשרת ישנן שלוש, לולאה אחת שמטפלת בלקוח מסוג admin, לולאה שנייה לטיפול בלקוח מסוג user ולולאה שלישית לניהול הSQL.

לולאת admin

* connect- מקבל את פרטי ההתחברות של האדמין ומאמת אותם
* Wait for start- מחכים לאדמין שיתחיל, כל פעם שמתחבר user חדש מעדכן את הadmin על מספר הclients שהתחברו. אם האדמין מבקש אז ניתן לקרוא ללקוחות שהתחברו בפעם הקודמת
* On\_session- מעביר את המידע (קלטים) שנשלחים מהאדמין לuserים, שולח הודעה כל פעם לuser שתורו לשתף מסך בהתאם למתי שהאדמין רוצה להחליף משתמש שממנו הוא מקבל תמונת מסך
* Check stop- בודק האם הadmin רוצה לסיים את השליטה

לולאת user

* Connect to host- מחברים את הuser לאדמין שבחר לפי שם ושיסמא.
* Waiting for start- מחכים שהאדמין יתחיל את הסשן
* Send\_buffer- כאשר הbuffer של הuser אינו ריק שולחים אותו
* Check\_stop- בדוק האם נגמרה השליטה של הadmin

לולאת waiting\_client

* מתחבר לשרת
* בודק האם ברשימה של לקוחות שצריכים להתחבר- אם כן מחבר user ויוצא
* בודק האם ברשימה של לקוחות שלא צריכים לחכות- אם כן יוצא

לולאת client admin

* מתחבר לשרת
* מחכה שהמשתמש ירצה להתחיל. יכול באותו זמן לקרוא למשתמשים קודמים
* Record keyboard mouse- אם המשתמש בחר להתחיל שליטה נתחיל הקלטה של המקשים והעכבר ונרים server שיקבל תמונות מסך מהusers בthread נוסף
* Send actions- שולחים את הפעולות שתועדו
* Stop- אם המשתמש בחר להפסיק בתפריט נפסיק את ההקלטה

לולאת client user

* Connect- מתחברים לadmin על פי בחירה מתוך רשימה שלadmins וסיסמא.
* Wait for start- מחכה שהשרת יגיד לו שהסשן התחיל
* Receive actions- קבלת פעולות מהשרת והכנסתם לbuffer.
* Commit action- ביצוע הפעולות הנמצאות בbuffer
* אם השרת מבקש מתחיל בthread נפרד לקוח ששולח לadmin תמונות מסך מהמסך שלו

## תיאור תפריטים

בפרויקט ישנם שני תפריטים. תפריט אחד הוא תפריט לadmin. תפריט זה מאפשר ללקוח להתחיל שליטה על מחשבים להפסיק אותה ולראות על כמה מחשבים הוא שולט בכל זמן נתון. התפריט יהיה תפריט גרפי שיהיה נוח למשתמש.

תפריט שני הוא תפריט שמוצג לuser. כאשר הוא אינו מופעל אוטומטית. התפריט מאפשר לו לבחור admin להתחבר אליו ולהכניס את פרטי ההתחברות: שמו, שם האדמין וסיסמא.

## טכנולוגיות בהם נעשה שימוש

1. Socket- מודול זה משמש אותנו כדי להעביר הודעות.
2. Threading- גם בשרת וגם בלקוח נשתמש בכמה threadים
3. Pynput- ישמש אותנו על מנת לתעד את הקשות המקלדת ולחיצות העכבר.
4. Pyautogui- ישמש אותנו על מנת לבצע את הקלט שהגיע מהadmin
5. Tkinter- נשתמש במודל זה על מנת ליצור ממשק גרפי למשתמש.
6. Sqlite3- ישמש אותנו על מנת לנהל את בסיס הנתונים.
7. Time- במודול זה נשתמש על מנת לשמור זמנים וליצור השהיות.
8. Pickle- נשתמש במודול זה כדי לשלוח python objects.
9. Crypto- ספרייה שמשמת להצפנות AES וRSA.
10. Base64- ספריה זו משמשת לקידוד בbase64.
11. Json- נשתמש בספרייה זו על מנת לשמור קובץ ובו פרטי ההתחברות האחרונים של user.
12. Sys- קבלת argv
13. Os- יצירת רצץ רנדומלי בעבור מפתח AES והרצת הwaiting\_client באופן אוטומתי.
14. Functools- נשתמש בפונקציה partial שעוזרת בקריאה לפעולות באופן אסינכרוני
15. Winreg- לרשום אוטומטית את הקובץ waiting\_client.py לRegistery.
16. Subprocess- משמש לפתיחת process נפרד
17. Mss- משמש ללקיחת תמונות מסך
18. Zlib- משמש לדחיסת תמונות המסך שנלקחו

## קבועים חשובים

SIZE\_OF\_LEN- קבוע זה נמצא גם בשרת וגם בלקוח וקובע מהו אורך המספר שמייצג את גודל ההודעה בפרוטוקול התקשורת חשוב לשים לב שהקבוע שווה גם בצד השרת וגם בצד הלקוח

IP,PORT- קבועים אלו מייצגים את כתובת הIP של השרת ואת הפורט עליו הוא מאזין

DATABASE\_FILE- השם של קובץ הdb.

TABLE\_NANE- שם הטבלה שבבסיס הנתונים

RSA\_KEY\_SIZE- הגודל שמשמש את הRSA על מנת ליצור מפתחות להצפנה.

## פרוטוקול תקשורת

**תקשורת שרת-אדמין-יוזר**

מטרה: להעביר את הקלט מהעכבר ומהקלדת של המחשב השולט (שרת) אל שאר המחשבים (לקוחות)

סוגי הודעות

|  |  |  |
| --- | --- | --- |
| **הודעה** | **פרוט** | **כיוון** |
| ISOLD | מבקשת פרטים מהלקוח | שרת אל אדמין |
| LOGED | מודיעה ללקוח שהוא נכנס לחשבון בהצלחה | שרת אל אדמין/ שרת אל יוזר |
| READY | מודיע ללקוח שהשליטה שלו התחילה | שרת אל אדמין |
| NUMCO | מודיע ללקוח כמה יוזרים התחברו | שרת אל אדמין |
| ADLST | שולח את הרשימה של כל השמות של האדמינים שיכולים לקבל יוזרים | שרת אל יוזר |
| CONNE | מודיע ללקוח שעליו לחבר את היוזר | שרת אל יוזר |
| STOPP | מודיע ללקוח שהוא יכול להפסיק לחכות שהשרת יגיד לו להפעיל את היוזר | שרת אל יוזר |
| INPUT | 1. הודעה שבה יש קלטים שהלקוח צריך לבצע 2. הודעה שבה יש קלטים שהאדמין הקליט | 1. שרת אל יוזר 2. אדמין אל שרת |
| START | 1. מודיע לusers שסשן התחיל. 2. admin מודיע לשרת שהוא רוצה להתחיל סשן | 1. שרת אל יוזר 2. אדמין אל שרת |
| ENDSE | מודיע לusers שסשן נגמר | שרת אל יוזר |
| SHARE | מודיע לuser שעליו לשתף מסך | שרת אל יוזר |
| CALST | מבקש מהשרת לקרוא לusers האחרונים שמתבקש | אדמין אל שרת |
| LOGIN | הודעה ובה פרטי ההתחברות של הadmin | אדמין אל שרת |
| CLNUM | אדמין מבקש מהשרת שישלח לו את כמות הלקוחות שמחוברים | אדמין אל שרת |
| SHRUP | Admin מודיע לשרת שברצונו לשנות את הuser שממנו | אדמין אל שרת |
| ENDSE | Admin מודיע לשרת לסיים את הסשן | אדמין אל שרת |
| CONEC | מידע ובו פרטי החיבור של הuser (שם, שם אדמין, סיסמא של אדמין) | יוזר אל שרת |
| BEGIN | מודיע לשרת שהוא מוכן לקבל קלטים | יוזר אל שרת |

שדות

שדה ראשון- 7 תווים מציין את גודל ההודעה

שדה שני – 5 תווים מציין את סוג

שדה שלישי תוכן ההודעה

דוגמה להודעה KEYBOHELLO0000010 הודעה זו נשלחת מהשרת ללקוח ומודיעה שנלחץ במחשב השולט HELLO

צורת מענה

צורת המענה תהיה סינכרונית ישנה חשיבות לכך שההודעות יגיעו בסדר מתאים מכיוון שכל הודעה מייצגת קלט שונה ועל מנת ליצור תוצאה זהה בכל המחשבים עלינו לוודא שהקלט יגיע אליהם באותו הסדר.



שגיאות

ERROR1- לקוח התנתק

ERROR2- לקוח לא יכול לבצע קלט

ERROR3- קלט לא זוהה.

**תקשורת שיתוף מסך**

מטרה: להעביר תמונות מסך ממחשב אחד לאחר

סוגי הודעות

|  |  |  |
| --- | --- | --- |
| **הודעה** | **פרוט** | **כיוון** |
| PIXEL | הודעה ובה תמונת המסך | לקוח של שרת |
| SSIZE | הודעה ובה נמצאים הגדלים של המסך | לקוח אל שרת |
| ENDST | מודיע ללקוח שהוא צריך להפסיק | שרת אל לקוח |

שדות

שדה ראשון- 7 תווים מציין את גודל ההודעה

שדה שני – 5 תווים מציין את סוג

שדה שלישי תוכן ההודעה

דוגמה להודעה KEYBOHELLO0000010 הודעה זו נשלחת מהשרת ללקוח ומודיעה שנלחץ במחשב השולט HELLO

צורת מענה

צורת המענה תהיה סינכרונית ישנה חשיבות לכך שההודעות יגיעו בסדר מתאים מכיוון שכל הודעה מייצגת קלט שונה ועל מנת ליצור תוצאה זהה בכל המחשבים עלינו לוודא שהקלט יגיע אליהם באותו הסדר.

שגיאות: אין שגיאות בפרוטוקול זה

## הצפנות

הצפנת מידע אישי

את תהליך ההתחברות של הadmin והuser נבצע באופן מוצפן. הרעיון הוא להצפין את ההודעות בהצפנה סימטרית על מנת להגיע להצפנה סימטרית נצטרך קודם להעביר מפתח שיהיה אחיד בשתי הצדדים על מנת לעשות זאת נשתמש בהצפנה א-סימטרית, RSA. נייצר בשרת מפתח ציבורי וממנו נייצר מפתח פרטי נשלח את המפתח הציבורי ללקוחות הם ייצרו ממנו מפתח פרטי באמצעות המפתח הפרטי הם יצפינו את מפתח הAES שהם ייצרו וישלחו בחזרה לשרת. לאחר שדבר זה יבוצע בשרת ובלקוח יהיה את אותו מפתח AES ויהיה ניתן להתחיל בתקשורת מוצפנת. הודעת ההתחברות שתכלול את פרטיהם האישיים של הלקוחות תוצפן בAES וכך תובטח פרטיותם של המשתמשים. הפעולה generate\_aes\_key

קידוד בשמירה של רשימות של כתובות IP

על מנת לשמור רשימות של כתובות IP נבצע על הרשימות serialization באמצעות pickle לפלט נבצע קידוד עם base64. כאשר נרצה לקבל את הרשימה בחזרה קודם נבצע decode לbase64 לאחר מכן בעזרת pickle נבצע de-serialization וכך נקבל את הרשימה בחזרה.

## אלגוריתמים מרכזיים

Keylogger- האלגוריתם של הkeylogger הוא אחד המרכזיים בפרויקט, מכיוון שאם ברצוננו להעביר את הקלט ממחשב אחד אל כמה מחשבים אחרים עלינו קודם כל לזהות את הקלט שנכנס במחשב הראשי, הkeylogger עוזר לנו את הקלט שמגיע מהמקלדת. האלגוריתם הוא די פשוט כל מקש שנכנס לbuffer של המקלדת מתועד בקובץ שבו אפשר לראות את כל הקשות המקלדת.

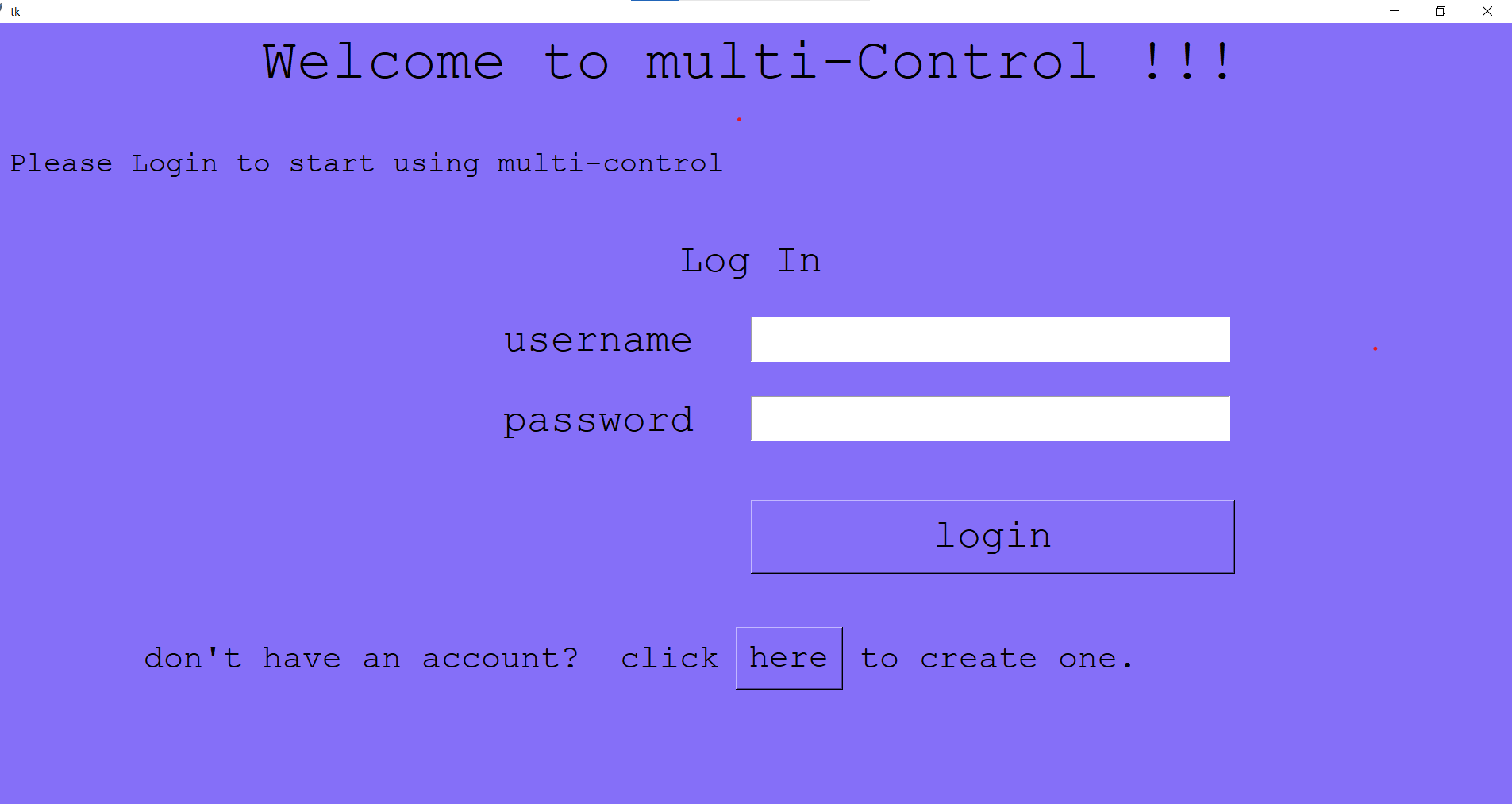


האלגוריתם המרכזי- הרעיון שעומד בבסיס הפרויקט הוא שבאמצעות העברת קלט מהעכבר והמקלדת ממחשב אחד למחשבים אחרים ניתן יהיה לבצע את אותם פעולות בכמה ממחשבים. לכן בכל פעם שנלחץ מקש או עכבר במחשב השולט יועבר הקלט לשרת המרכזי השרת המרכזי יעביר את הקלט לכל המחשבים עליהם שולט השרת.

# **מדריך למשתמש**

המשתמש הינו admin אשר רוצה לשלוט על כמה מחשבים. על מנת לבצע דבר זה עליו דבר ראשון להתחבר לחשבון שלו בmultu-control. לשם כך הוא יפעיל את תוכנת הלקוח של הadmin ויפתח בפניו המסך הבא:

מסך מספר1:

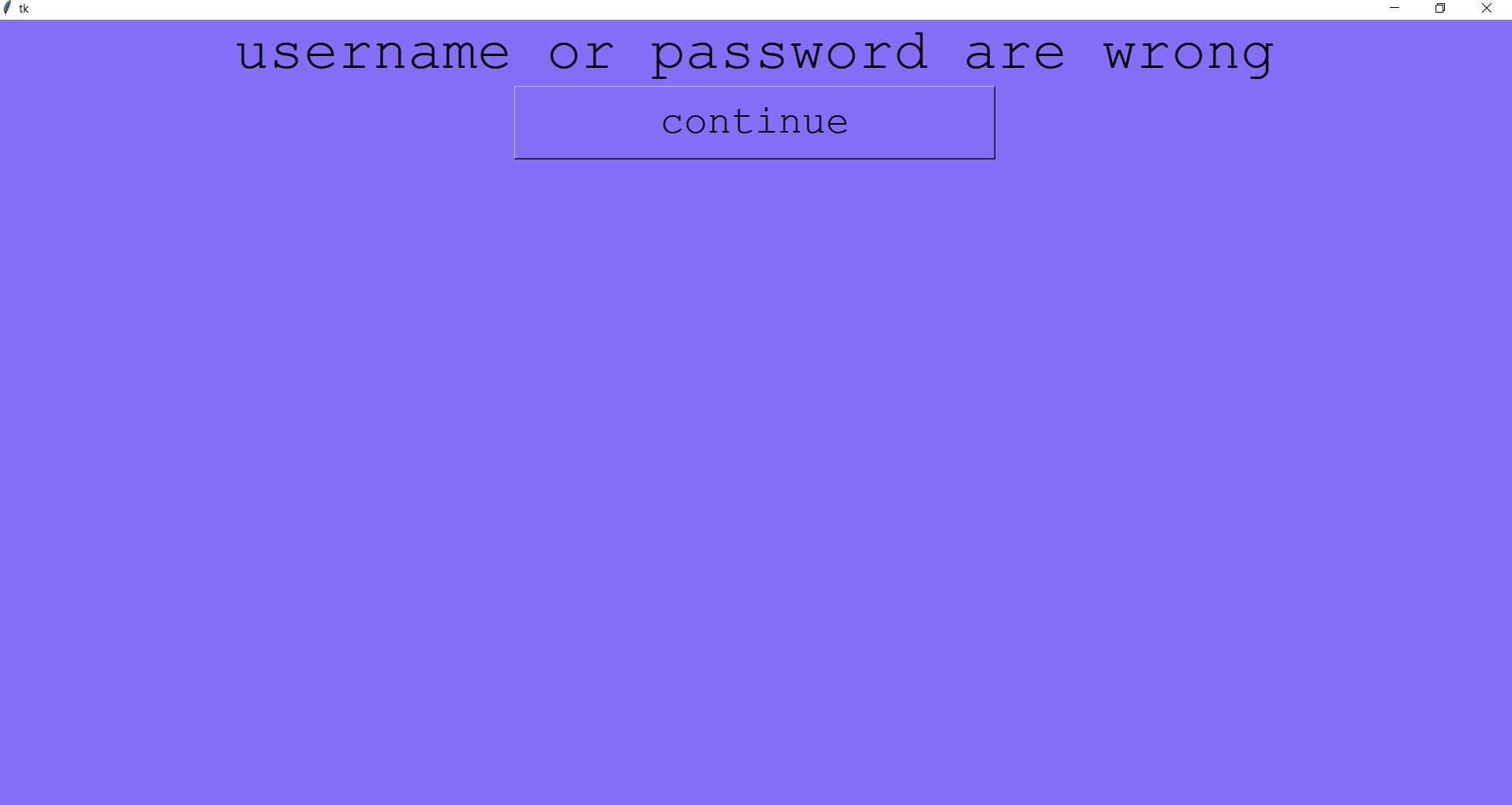


אם כבר יש לו חשבון הוא ימלא בתיבת הטקסט הראשונה את שם המשתמש ובתיבת הטקסט השנייה את הסיסמא שלו אם היא נכונה יעבור למסך הבית ממנו יוכל להתחיל את השליטה.

אם הסיסמא או שם המשתמש אינם תקינים תופיע הודעה בחלק התחתון של המסך על כך. כלומר אינם מכילים צירוף של אותיות ומספרים.

אם הסיסמא או שם המשתמש שגויים יועבר המשתמש למסך הבא:

מסך מספר 2:



אם המשתמש חדש לmulti- control עליו ללחוץ על הכפתור שרשום עליו here שראינו המסך הראשון ולאחר מכן יועבר למסך הבא:

מסך 3:



במסך זה הוא יכניס את השם משתמש והסיסמא הרצויים אם הם אינם תקינים תקפוץ הודעה לתחתית המסך שתיידע אותו על כך

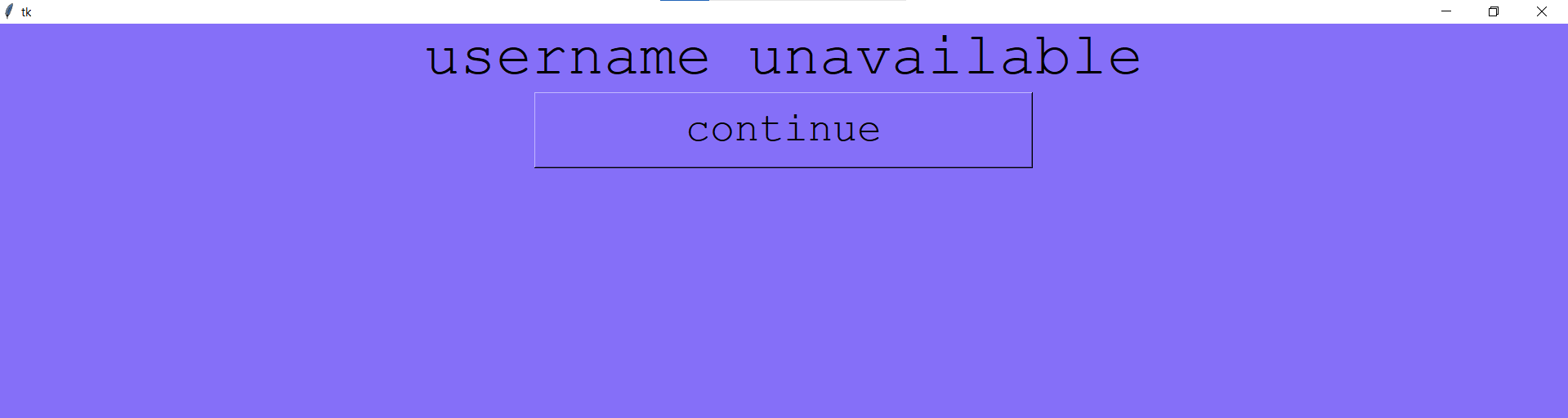
אם שם המשתמש כבר תפוס הוא יראה את המסך הבא:

מסך 3:

במידה והשם משתמש או הסיסמא אינם תקינים או שהסיסמא שהוכנסה בתיבה השלישית שונה מהתיבה השנייה תוצג הודעה מתאימה בתחתית המסך.

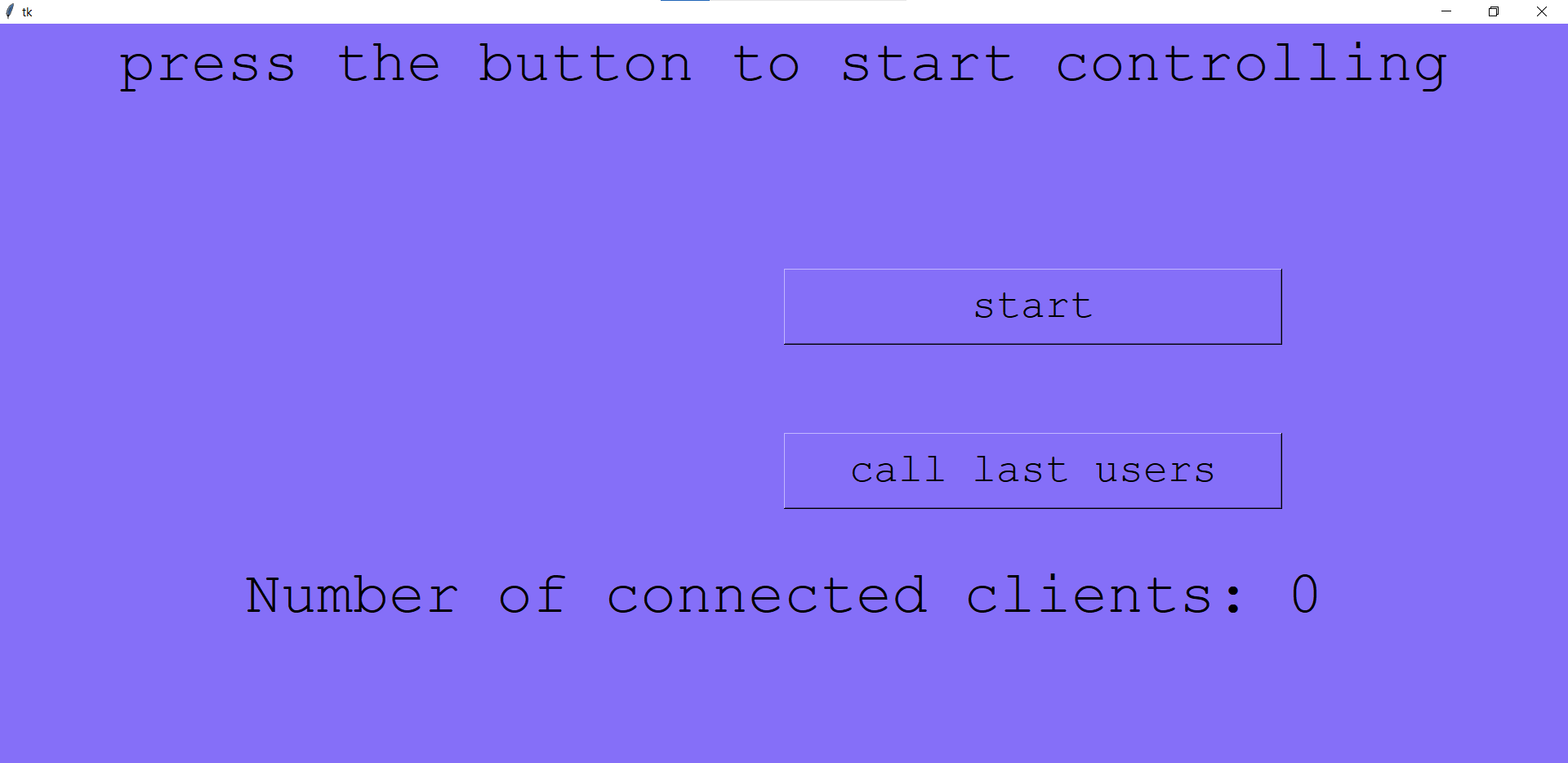
אם שם המשתמש תפוס נעבור למסך הבא

מסך 4:



אם הסיסמא ושם המשתמש תקינים ושם המשתמש אינו תפוס הוא יעבור למסך הבית אותו מסך שאליו יעבור משתמש ישן שהכניס נכונה את הסיסמא ושם המשתמש הישן שלו:

מסך 5:



במסך זה יש שני אפשרויות להתחיל שליטה ולקרוא לusers ישנים. בנוסף למעלה מוצג מספר הusers שחוברו ומוכנים לשליטה

לחיצה על כפתור call last connected יקרא לכל המחשבים שהתחברו בפעם האחרונה לadmin זה ויחבר אותם. ניתן לראות בטקסט למעלה אם הusers באמת התחברו אם מספר הלקוחות המחוברים גדל.

לחיצה על כפתור start session יגרום לכך שמאותו רגע כל הקשה על המקלדת ועכבר תבוצע גם במחשבים שהתחברו. לחיצה על כפתור זה תעביר אותנו לחלון הבא:

מסך 6:



בנוסף למסך זה ייפתח גם חלון pygame בו תוצג תמונת נמסך של אחד המשתמשים. ניתן להחליף בין המסכים של המחשבים באמצעות לחיצה על מקש SHIFT וחץ ימינה.

לחיצה על כפתור end session תעצור את העברת הקלט מהעכבר והמקלדת לשאר המחשבים ותסגור את תוכנת הmulti-control.

אם רוצים לחבר מחשב חדש לשליטה יש להריץ עליו את תוכנת הלקוח של הuser.

תחילה ייפתח המסך הבא:

מסך 7:



יש להכניס בתיבה טקסט הראשונה את השם של מי שמשתמש במחשב. בתיבת טקסט השנייה את השם של הadmin שאליו נרצה להתחבר. ובתיבת הטקסט השלישית את הסיסמא של אותו admin. במידע והשם משתמש או הסיסמא אינם תואמים יפתח חלון pop up אם תצוגה תואמת. לאחר לחיצה על login מסך זה ייסגר וזה סימן שהמחשב חובר בהצלחה לadmin והוא יכול להתחיל לשלוט עליו בכל רגע.

# **בסיס נתונים**

בפרויקט ישנם שני בסיסי נתונים עיקריים. בסיס נתונים ראשון הוא שומר את הסיסמאות ושמות המשתמשים של הadmins נקרא admins\_sql. מטרתו היא לאפשר התחברות של admin עם שם משתמש וסיסמא. השני הוא שומר את כתובות הIP של הusers האחרונים שהיו מחוברים לכל אחד מהadmins הוא נקרא users\_ip\_sql. מטרתו היא לאפשר לadmin לחבר אליו אוטומטית את כל המחשבים שהתחברו אליו בפעם האחרונה.

## רשימת העמודות

Admins\_sql

|  |  |  |  |
| --- | --- | --- | --- |
| **עמודה** | **משמעות** | **סוג משתנה** | **האם חובה** |
| id | מפתח ראשי(PRIMARY KEY) | INTEGER | כן |
| username | שם המשתמש של הadmin | TEXT | כן |
| password | סיסמא מוצפנת של הadmin | TEXT | כן |

Users\_ip\_sql

|  |  |  |  |
| --- | --- | --- | --- |
| **עמודה** | **משמעות** | **סוג משתנה** | **האם חובה** |
| id | מפתח ראשי(PRIMARY KEY) | INTEGER | כן |
| AdminIP | כתובת IP של הadmin | TEXT | כן |
| IpList | רשימה ובה כתובות הIP של כל הusers שהתחברו לadmin בפעם האחרונה | TEXT | כן |

## שאילתות SQL

**Admins\_sql**

* INSERT INTO admins\_sql (id, username, password) VALUES (1,’itai’,’is12’) מכניס שורה חדשה לטבלה.
* SELECET password FROM admin\_sql WHERE username == ‘itai’ מחזיר סיסמא של admin ספציפי.
* SELECT username FROM admin\_sql מחזיר את כל שמות המשתמשים משמש כדי לבדוק ששם משתמש כלשהו אינו כבר תפוס

**Users\_ip\_sql**

* SELECET AdminIP FROM users\_ip WHERE AdminIP == ‘1.2.3.4’ בודק האם כבר קיימת רשומה בבסיס הנתונים עבור IP מסויים.
* UPDATE users\_ip SET IpList=[‘2.3.5.3’]\* WHERE AdminIP == ‘1.2.3.4’ מעדכן את רשימת הכתובות של הלקוחות בעבור admin שכבר קיימת לו רשומה בבסיס הנתונים
* INSERT INTO users\_ip (id, AdminIP, IpList) VALUES (1,‘4.3.2.1’,[‘1.1.1.1’]\*) מכניס רשומה חדשה לבסיס הנתונים
* SELECT IpList FROM users\_ip WHERE AdminIP == ‘1.2.3.4’ מחזיר את הרשימה של כתובות הIP של admin ספציפי.

\*על מנת לשמור את רשימת הכתובות בבסיס הנתונים נעשה serialization לרשימה באמצעות pickle את הפלט נקדד עם base64 ואותו נשמור בבסיס הנתונים. לכן במקומות בהם מופיע רשימה מופיעה למעשה מחרוזת של base64.

# **מדריך למפתח**

רשימת קבצים ותפקידם

|  |  |  |
| --- | --- | --- |
| שם הקובץ | מטרתו | מיקומו בספריות הפרויקט |
| Admin\_class | בעבור כל admin שמתחבר לשרת נוצר אובייקט ממחלקה זו מטרת המחלקה היא לבצע פעולות שקשורות לadmin בצד השרת | Server side |
| Admins\_sql | מתפעל את בסיס הנתונים ששומר סיסמאות ושמות משתמשים | Server side |
| AES\_class | משמש ליצירת תקשורת מוצפנת עם AES | Server side, admin side, user side |
| Main\_server | הקוד של השרת | Server side |
| Sql\_manage | עוזר לנהל טבלאות sql | Server side |
| Tcp\_by\_size | שולח הודעות על פי פרוטוקול התקשורת | Server side, admin side, user side |
| User\_class | בעבור כל user שמתחבר נוצר אובייקט ממחקלה זו. מטרת המחלקה היא לבצע פעולות שקשורות בuser. | Server side |
| Users\_ip\_sql | מנהל את בסיס הנתונים ששומר את כתובת הIP של הusers | Server side |
| Action class | בקובץ זה נמצאות המחלקות שמבצעות קלטים מהמקלדת ומהעכבר | User side |
| Screen\_share\_user | שולח תמונות מסך לadmin | User side |
| User\_client | הקוד המרכזי של הלקוח מסוג user | User side |
| Waiting\_client | הקוד המרכזי של הלקוח שמתחבר אוטומטית לשרת ומחכה שהשרת יגיד לuser להתחבר | User side |
| Admin client | הקוד המרכזי של הלקוח מסוג admin | Admin side |
| Admin\_gui | אחראי לגרפיקה שמוצגת בפני לקוח מסוג admin | Admin side |
| Key\_mouse\_logger | מקליט את העכבר והמקלדת של הadmin | Admin side |
| Screen\_share\_admin | אחראי להציג את תמונות המסך שמתקבלות מה | Admin side |

## הוראות הפעלה

יש להוריד לpython 3.7 את הספריות הבאות

* Crypto
* Tkinter
* Pyautogui
* Pynput
* PIL
* Sqlite3
* Pygame
* Zlib
* Mss

יש לוודא שהשרת פועל לאחר מכן ניתן להפעיל את הלקוחות דרך Pycharm או cmd ניתן להעביר את הIP של השרת כargv לשני סוגי הלקוחות

use case

השימושים העיקרים של הפרויקט הם כאמור בחדרים מחשבים גדולים בהם רוצים לבצע פעולה או כמה פעולות על כמה מחשבים. לדוגמה רוצים להתקין פייטון 3 על כל המחשבים. כדי לעשות דבר זה נצטרך להתחבר לשרת המרכזי עם client מסוג admin שאר המחשבים יחוברו גם הם לשרת אבל בתור clients מסוג users. לאחר מכן המחשב שהתחבר בתור admin יתחיל שליטה על המחשבים בהנחה ובמחשבים יש שולחן עבודה זהה (דבר נפוץ בחדרי מחשבים) כאשר המחשב שיתחבר כadmin יקליק על מנת לפתוח דפדפן, הדפדפן יפתח בכל המחשבים המחוברים.

# רפלקציה

העבודה על הפרויקט הייתה מלמדת מאוד וחווייתית. חוויתי עליות ומורדות בזמן העבודה על הפרויקט. היו רגעים של תסכול ועצבים אך לצידם גם רגעים של שמחה והצלחה. היה לא קל להביא את הפרויקט לרמה שרציתי שהוא יהיה אך לבסוף הגעתי לתוצר שאני מסופק לגמרי מאיך שהוא עובד. מן העבודה על הפרויקט קיבלתי שיעור בנחישות ודבקות במטרה על איך לא לוותר גם שקשה וגם שמתוסכלים תמיד להמשיך לנסות ולא לוותר. הקושי העיקרי שלי היה לעמוד בדרישות של עצמי, דרשתי מעצמי הרבה לעיתים אף יותר מידי הצבתי לעצמי סטנדרטים גבוהים לאיך שאני רוצה שהפרויקט יעבוד, ולעיתים בזבזתי זמן על דברים פחות חשובים. בנוסף שנה זו לא הייתה שנה רגילה בשבילי ובעבור כולנו הלמידה מרחוק הקשתה על העבודה על הפרויקט מכיוון שלא יכולנו לעבוד במעבדת המחשבים, דבר זה גרם לי להתקשות בהרצת הפרויקט מכיוון והוא דורש מעורבות של מספר מחשבים, ובכללי מיועד לחדרי מחשבים כמו זה בבית ספר, בנוסף בעקבות הלמידה מרחוק לא יצא לנו ככיתה לחוות עבודה משותפת על הפרויקטים דבר שהוא מאוד חוויתי ולדעתי אחד מהדברים היותר כיפים שהיו אמורים להיות לנו במהלך הלימודים במגמה. כאמור העבודה על הפרויקט נתנה לי המון כלים הן בצד האישי והן בחלק המקצועי. למדתי שבעזרת תכנון מסודר של הדברים העבודה הופכת להרבה יותר קלה. אחד הדברים שאני מתחרט עליהם הוא התוכנה שהשתמשתי בה לממשק GUI אני חושב שבסופו של דבר אם הייתי משתמש בספרייה אחרת הייתי יכול להגיע לתוצר יותר טוב. אני חושב שאחת הבעיות שהיו לי בפרויקט היא שלא היה לי בראש מההתחלה מה יהיה התוצר הסופי, ותוך כדי עבודה רציתי להוסיף עוד ועוד דברים אילו ידעתי מראש מהו התוצר אליו אני מכוון יכולתי לעבוד בצורה הרבה יותר יעילה.