**סיכום הספרינט שעבר**

**ספרינט ראשון – 8-29 לפברואר:**

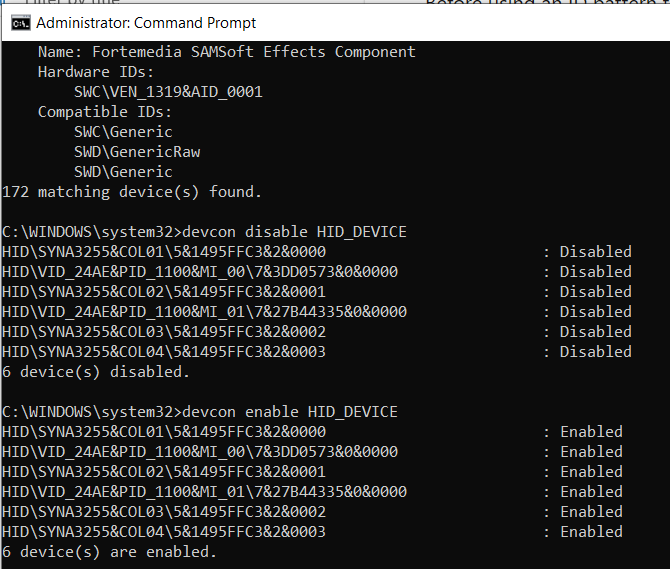
1. **הפרדת תהליכים של תקשורת וממשק המשתמש, כך שממשק המשתמש רץ במקביל לתקשורת:**

**באמצעות הספרייה queue שקיימת בפייתון 3.7 הצלחתי להריץ תהליכונים במקביל זה לזה כך שהתוכנה של ממשק המשתמש רצה במקביל לפעולות התקשורת של השרת עם הלקוחות. Queue זהו סוג של טיפוס נתונים, יצרתי בקובץ של המשתנים הגלובליים משתנה בשם conn\_q שאחראי על ניתוב מידע רצוי למיקום הרלוונטי בתהליכון אחר. בכל פעם שהמורה לוחץ על פעולה מסוימת, למשל על פעולת שליחת המסך, נוסף אל conn\_q שם הפעולה ובעקבות זאת בפעולה האחראית על שליחת הודעות בשרת יודעת להפנות את השרת לפעולה הנכונה ( למשל לפעולת צילומי המסך) במקביל לתרד של ממשק המשתמש. אמנם לא כל התקשורת מופרדת ממשק המשתמש (בעיקר בקובץ הלקוח), אך עדיין הצלחתי לפתור את הבעיה של הרצת התהליכים במקביל.**

1. **נעילת העכבר, המקלדת ומסכי התלמידים בצורה שגם לחיצה על control+alt+delete יהיה חסום בפני המשתמש. לעבוד גם על שחרור נעילה שיעבוד בצורה חלקה:**

**כיוון שספריית pywinhook חוסמת במהלך הנעילה את המשך ההרצה של הקוד גם כאשר זה רץ בץרד נפרד, לכן עברתי לרעיון של שימוש ב-devcon. זהו כלי cmd שמציג מידע מפורט על התקנים במחשבים שבהם פועל windows. באמצעות devcon ניתן להפעיל, להשבית, להתקין, להגדיר ולהסיר התקנים של המחשב.**

**דוגמא לפעולה הנועלת את העכבר באמצעות devcon:**



**באמצעות הקוד הבא אני ומתן מצאנו דרך לנעול את כל התקני הusb שקיימים במחשב ולאחר מכן להחזיר אותם, אך על מחשב לפטופ אנחנו צריכים למצוא פיתרון לנעול בדרך אחרת:**

**import subprocess**

**from time import sleep**

**def lock():**

**subprocess.call("devcon remove usb\*")**

**def open():**

**subprocess.call("devcon rescan")**

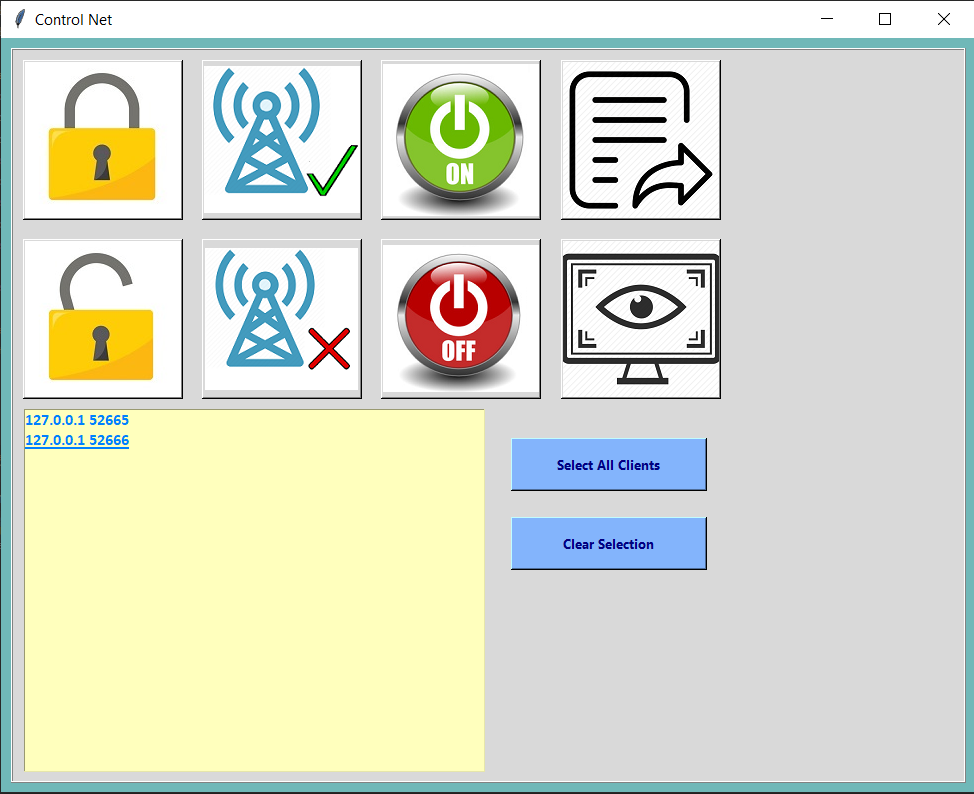
**lock()**

**sleep(10)**

**open()**

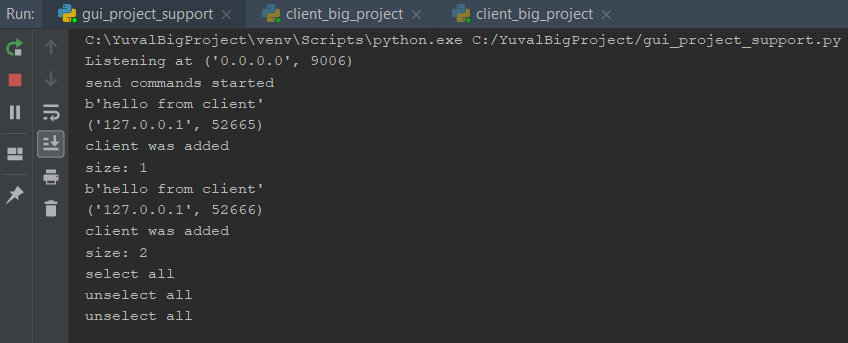
1. **הצגת המחשבים המחוברים למערכת על ממשק המשתמש של המורה:**

**יצרתי בקובץ של המשתנים הגלובליים אובייקט בשם gui\_queue מטיפוס queue, הוא אחראי על הצגת המידע על גבי ממשק המשתמש. בכל פעם שלקוח חדש מתחבר נוסף לlistbox שנמצא בממשק המורה כתובת הip והפורט של הלקוח.**



Clients address

בדיקות של תקינות -



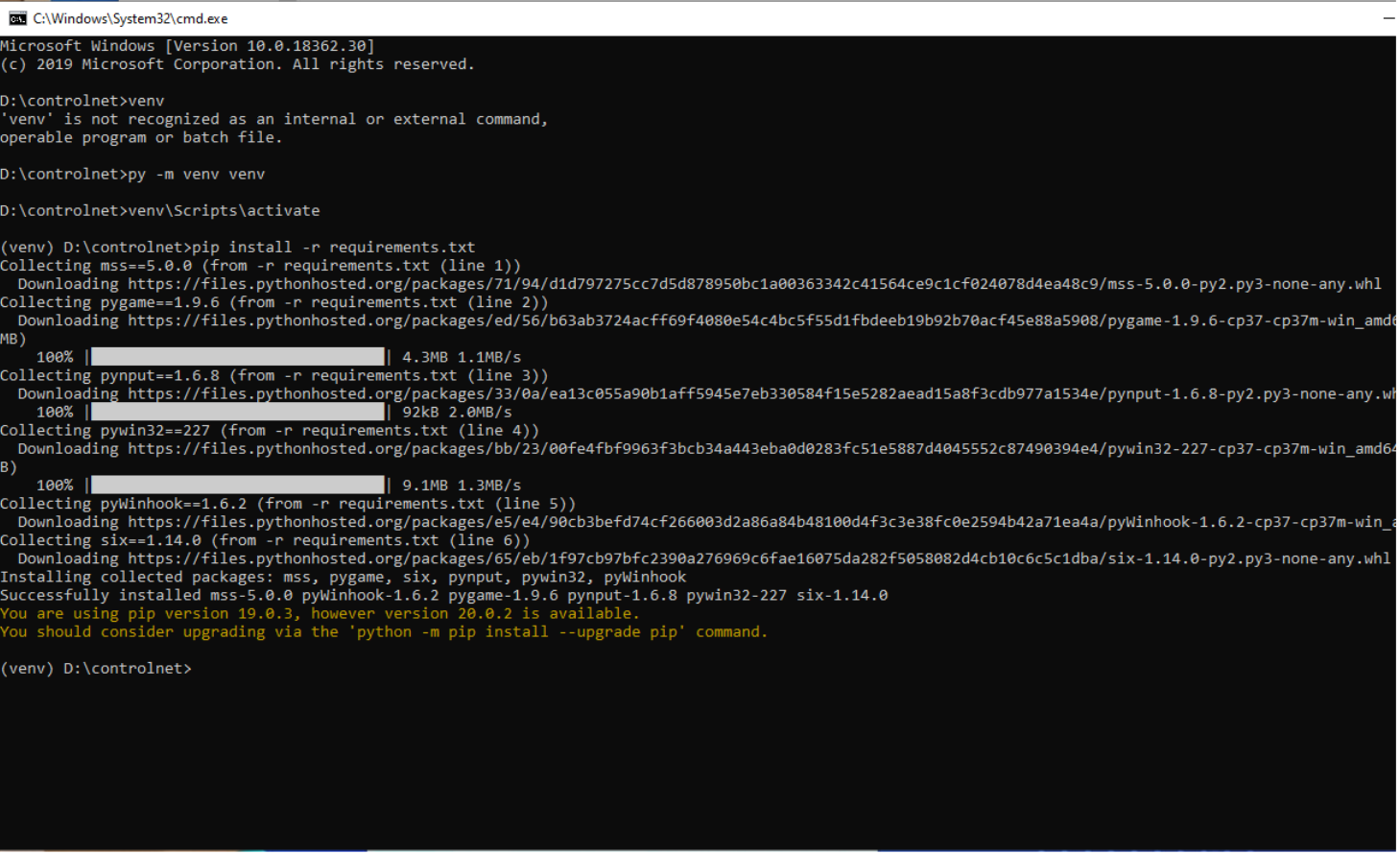
1. **הוספת בחירת המורה האם לבצע את הפעולות על כתובת ה-broadcast, או ביצוע הפעולות על מחשבי התלמידים שהוא בוחר: הוספתי 2 כפתורים של בחירת כל המשתמשים שנמצאים ב-listbox ושל מחיקת הבחירה. המורה בוחר את התלמידים עליהם רוצה לבצע את הפעולה וכאשר לוחץ על פעולה מסויימת הם נכנסים למערך selected\_clients. לא הצלחתי להתמש בכתובת הbroadcast אז כרגע אני עדיין חוקר את זה.**
2. **כיבוי מחשבים מרחוק: הוספתי פעולה זו, השורה האחראית על כך- os.system(‘’shutdown /p /f’’)**
3. **הוספתי לשידור המסך של המורה לתלמידים האזנה למיקומי העכבר באמצעות יצירת socket חדש האחראי לטיפול בעכבר ומשתמש במודל pynput. העכבר במחשבי התלמידים זז לפי מיקומי עכבר המורה.**

**ספרינט שני – 29 לפברואר עד 21 במרץ:**

1. **צפייה במסך תלמיד מסוים שהמורה בוחר מתוך התלמידים הקיימים במערכת -** *צילום מסך התלמיד הנבחר ע"י המורה והצגתו על מסך המורה – הוספתי את הפ'יצאר הזה באמצעות כך שיצרתי* **socket** *חדש שאחראי רק על שליחת המסך של התלמיד הנבחר וקבלת המסך אצל השרת. סוקט זה בשם* **watch\_server\_socket** *פועל במקביל לתקשורת הראשית.*
2. **עצירת שידור מסך המורה- הצלחתי לפתור את הבעיה בעזרת הוספת print במקומות נחוצים וגם הוספתי timeout לsocket באמצעות הפקודה settimeout()כך שברגע שמפסיקים את השידור חלון הpygame נסגר.**
3. **נעילת העכבר, המקלדת ומסכי התלמידים בצורה שגם לחיצה על control+alt+delete יהיה חסום בפני המשתמש. לעבוד גם על שחרור נעילה שיעבוד בצורה חלקה – יצרתי קובץ טקסט באופן אוטומטי בעזרת הספרייה הבנויה shelve. קובץ זה שומר משתנים גלובליים שאני צריך שגם כשהקוד לא רץ יהיו מעודכנים (למשל בשביל מצב בו התלמיד מכבה את המחשב, אז אני חייב כשהקובץ הנ"ל יהיה מעודכן). כאשר משתנה ששמרתי בקובץ בשם activation מכיל ערך true אני נועל למחשב את כל אופציות הusb באמצעות תוכנת devcon וכך גם אפשרות control+alt+delete לא קיימת לתלמיד. בנוסף, באמצעות קובץ batch אני מריץ מהתוכנה הראשית את הקובץ נעילה ובמקביל חלון tkinter שמוצג על המסך ומעדכן את התלמיד כי המחשב נעול.**
4. **udp broadcast- באמצעות הקוד הבא החזרתי מערך שמכיל את כתובת המחשב/המחשבים שהמורה בוחר לבצע עליהם את הפעולה ואליהם בלבד שולח את מה שצריך. כך הוספתי אפשרות גם לבצע על מחשב ספציפי פעולה וגם לבצע על כל המחשבים המחוברים למערכת, ולכן broadcast לפי כתובת udp broadcast פחות חשוב לי.**

**def** selection\_list():  
 **global** w, top\_level  
 cl = w.Clients\_List  
 selected = cl.curselection()  
 selected\_list = []  
 **if** selected: *# only do stuff if user made a selection* print(selected)  
 **for** index **in** selected:  
 *# selected\_list.append(cl.get(index).split(" ")[1])* selected\_list.append(cl.get(index)) *# how you get the value of the selection from a listbox* print(selected\_list)  
 **return** selected\_list

1. **הוספת database של התלמידים – יצרתי מחלקה בשם clients שיוצרת db באמצעות sql ושקיימים בה פעולות הוספת תלמיד וכו', בשלב הזה אני עובד על הוספת התלמידים לטבלה.**
2. **במעבדה בדקתי גם שהקוד עובד על מספר מחשבים כפי שהפרוייקט מיועד להתנהל – במעבדת מחשבים כאשר מורה שולט על התלמידים. באמצעות העזרה שלך, הורדתי סביבת עבודה venv לכמה מחשבים.**

****