

**ניתוח ובקרת מחשבים**

**שם: עדן פודוקסיק**

**תעודת זהות: 209230093**

**בית ספר: אורט גוטמן**

**כיתה: יב'7**

**עיר: נתניה**

**שמות המנחים ומורים מלווים: מיכאל צ'רנובילסקי,**

**גד רוזנטל**

מבוא ומוטיבציה לפיתוח........................................................................3

**תוכן עניינים**

תיאור מוצר.........................................................................................4

אילוצים ודרישות..................................................................................5

שפת תכנות וסביבת עבודה...................................................................6

בעיות אלגוריתמיות ומחקר....................................................................7

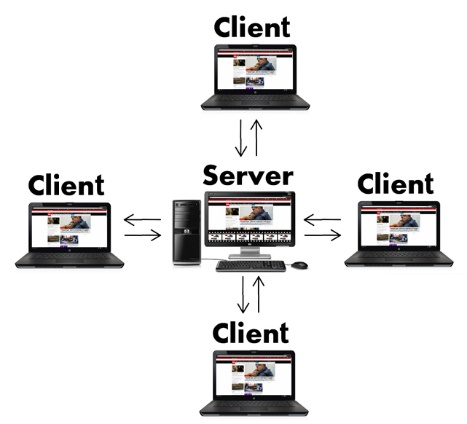
**מבוא ומוטיבציה לפיתוח**

במהלך המאה ה-21, עם הקמת חברות גדולות, החלו להיווצר בעיות של שימוש במחשב החברה לצורך פעולות לא רצויות. הפתרון לבעיה זו הוא להפעיל תוכנה בכל מחשב בחברה שבאפשרותה להקליט את המסך של כל עובד (בהסכמת המשתמש) לפי רצון השרת (המנהל) לצורך בקרה. לדוגמה, מנהל החברה יכול להחליט שבכל פעם שאחד מהמשתמשים נכנס לדפדפן, המחשב אוטומטית מתחיל לצלם את המסך של אותו משתמש.

בחרתי בנושא זה מכיוון שעובדים רבים מנצלים את סביבת עבודתם לצורכיהם האישיים ובכך פוגעים בארגון בו הם עובדים. לכן, מנהל החברה רשאי ואף מחויב לבקר את פעולות העובדים במחשבים.

הפרויקט אמור להוות תוכנת בקרה והקלטה על מחשבים לפי צורכי השרת. בתחילת תהליך הבקרה, מתבקש המשתמש להגדיר חריגות מסוימות, כמו כניסה ל- YouTube, שימוש ב- Windows Media Player , שימוש מופרז ב-CPU וכו', שבאפשרותן יהיה ניתן לבדוק האם הלקוחות המחוברים לשרת אכן לא פועלים כפי שעליהם לפעול (צפייה בסרט במקום לעבוד ועוד).

כאשר המערכת מזהה לפחות חריגה אחת באחד המחשבים המחוברים לשרת, היא מודיעה על כך למשתמש בתוכנה ושואלת אותו האם ברצונו "להקליט" את המחשב המסוים.

במידה והמשתמש מעוניין להקליט, הוא מתבקש להחליט האם הוא רוצה להקליט את כל מה שקורה במחשב או להקליט רק תהליך מסוים כגון דפדפן. לבסוף, התוכנה מצלמת את מסך המחשב כל כמה שניות ולאחר מכן שולחת את התמונות לשרת ועושה מתמונות אלה סרטון קצר שמראה את פעולתו של המחשב בתהליך זה.

**תיאור המוצר**

המטרה המרכזית של הפרויקט הוא פיקוח ובקרה על המתרחש במחשבים המחוברים לשרת.

כאשר התוכנה מזהה שלאחד המחשבים המחוברים לשרת יש לפחות חריגה אחת, היא מקליטה את המתרחש במחשב זה (על ידי צילום המסך כל כמה שניות ולבסוף יצירת סרטון מתמונות אלה) לפי קריטריונים הנקבעים בצד השרת.

התוכנה תבוא לידי ביוטי בעיקר בחברות גדולות כאשר מנהל החברה רוצה לפקח על הנעשה במחשבי החברה. השרת צריך להתמודד עם כמות מסוימת של חיבורים בו זמנית, למנוע שיקרה מצב שיהיו יותר מדי תמונות ושיהיו ממש מעט תמונות, לסדר את התמונות מכל מחשב בתיקיה נפרדת, לרכז את התמונות של כל מחשב בנפרד לסרטון אחד, למנוע בלבול במסד הנתונים בצד השרת ועוד.

במהלך העבודה על הפרויקט, השתמשתי במספר רכיבים עיקריים ולהלן סקירה לגביהם:

1. מרכיב עיקרי בפרויקט ועליו מתבסס הדיאלוג בין השרת ללקוחות הוא תקשורת. התקשורת בעולם המחשבים מבוססת על Network Socket , שהוא נקודות קצה עבור זרם נתונים בתקשורת בין תהליכים על גבי קשת מחשבים. רוב ה-Sockets כיום מבוססים פרוטוקול IP.

בפרויקט שלי התקשורת בין השרת לבין הלקוחות ובין השרת לבין ה-GUI מתבצעת באמצעות Internet Socket.

1. מרכיב חשוב נוסף בפרויקט הוא קבלת מידע על תהליכים הרצים במחשב. המודול האחראי על פעולה זו הוא WMI.

ה-WMI הוא ממשק של מערכת ההפעלה שמאפשר לקבל מידע לגבי רכיביה, ובעזרתו ניתן לקבל מידע לגבי רכיבי המערכת, החומרה והתוכנה. קבלת המידע על התהליכים שרצים במחשב בא לידי ביטוי בפרויקט בחיפוש חריגות, זיהוי התהליך שיש להקליטו ועוד.

1. בנוסף לכך, יש צורך לשלב בפרויקט מסד נתונים.

במסד הנתונים ישמרו רשימת החריגות מכל מחשב, תאריך הקלטה, מספר תמונות וכו'.

הנתונים ישמרו בתוכנת ה- SQL שהיא השפה המקובלת לטיפול ועיבוד מידע בבסיסי נתונים).

1. החלק הוויזואלי של הפרויקט יבוצע בשפת C# בסביבת עבודה Visual Studio. על הGUI - להיות נוח וברור למשתמש כדי שיוכל לבצע את כל פעולות התוכנה בקלות.

עם פיתוח הפרויקט, התגלו מספר אילוצים שעליהם המערכת נדרשת לענות:

ראשית, המערכת צריכה להתמודד עם נפח גדול של מידע מה- Client – במידה והמערכת צילמה מספר רב של תמונות, עליה לדאוג שכל התמונות יועברו באופן מלא.

בעיה נוספת היא שלפני פעולת ההקלטה יש להתקין את התוכנה על כל המחשבים הרצויים, ונצטרך את אישור המשתמש כדי שפעולה זו לא תחשב לריגול.

בנוסף לכך, לאחר בדיקת החריגות המערכת מבצעת תהליך של "אסיפת ראיות" ומקליטה את פעולתו של מחשב מסוים. דבר נוסף שניתן להוסיף לקטגוריית הראיות הוא צפייה במה שמצלמת מצלמת הרשת המחוברת באותו מחשב אך החלטתי שלא לעשות זאת מבחינה אתית בהסיבה שדבר זה הוא פלישה לפרטיות באופן מוחלט.

כמו כן, אפשרות לעתיד היא סטטיסטיקה ודירוג משתמשים לפי היסטוריית ההקלטות וחריגות שתשמר במסד הנתונים גם כן.

יתרה מזאת, המערכת עונה על מספר נוסף של דרישות:

1. בכדי לעבוד בו זמנית עם כל המחשבים המתחברים לשרת המערכת מפתחת Thread נפרד לכל לקוח המסוגל לתקשר במקביל עם כולם, לקבל מידע ולשלוח להם פקודות.
2. בכדי לקבוע קריטריונים מסוימים כמו איזה חריגות יש לבדוק במחשבים ואיזה תהליך יש להקליט, המערכת מאפשרת למנהל לקבוע את כל ההגדרות שקובעות את אופן פעולת התוכנה.
3. כאשר מחשב מפסיק לפעול ולתקשר עם השרת, המערכת מנתקת את התקשורת איתו ושולחת הודעה על ההתנקות לשרת.

**אילוצים ודרישות**

1. כאשר מחשב מסוים יוצא מהתהליך המוקלט צילום המסך מסתיים והתמונות נשלחות לשרת ומוצגות למנהל.
2. במקרה שלא זוהתה אף חריגה במחשב מסוים, המנהל רשאי להחליט האם להמשיך למחשב אחר אם שמא להקליט את המחשב בכל זאת.

הפרויקט נכתב ב- Python וב- C#. כל הליבה של הפרויקט נכתבה ב- Python בעקבות הלמידה של השפה בכיתות יא' ויב', וה- GUI של הפרויקט נכתב ב- C# בשל הנוחות ב- C# בעזרת ה- Visual Studio ואז לחבר אותו לליבת הפרויקט שנכתבה ב- Python בעזרת Socket.

**שפת תכנות וסביבת עבודה**

סביבת העבודה עם Python הייתה בתוכנה PyCharm בעקבות ההרגל שנרכש בכיתות יא' ויב' שהיא תוכנה הרבה יותר מקצועית ונוחה לעבודה על פרויקטים מרובי מודולים וקבצים. היא שימשה גם לפיתוח הקוד ועבודה במקביל עם כל המודולים השונים, וגם להרצתו בשביל בדיקות.

סביבת העבודה עם C# הייתה 15Microsoft Visual Studio 20 ששימשה לכתיבת ה- GUI באמצעות גרירת פקדים מה- toolbox, וה- debugger המובנה שימש לבדיקות. הבחירה בסביבת עבודה זו הייתה משום שה- Visual Designer המובנה היה פתרון מאוד נוח לכתיבת ה- GUI של הפרויקט.

סביבת העבודה עם מסד הנתונים הייתה SQLite, שהיא ספרייה מוטמעת (כחלק בלתי נפרד) בתוך התהליך של התוכנה, שמאפשרת את ניהול מסדי הנתונים המקומיים בצד השרת.

**בעיות אלגוריתמיות ומחקר**

במהלך כתיבת הפרויקט נתקלתי במספר בעיות אלגוריתמיות שדורשות מחקר עבורן.

ראשית, הדבר הראשון שצריך כדי ליצור קשר בין השרת ללקוחות המחוברים ובין השרת ל- GUI הוא יצירת שרת Multi-Client. הקשר בין השרת ללקוחות התקיים באמצעות Multi-Threading כך שכל לקוח יטופל בהתאם במקביל לשאר הלקוחות, והקשר בין השרת ל- GUI התקיים באמצעות Socket.

בהמשך הכתיבה הייתי צריך לבדוק חריגות מסוימות במחשבים המחוברים לשרת.

אחת מהחריגות היא שהמשתמש במחשב מסוים נכנס ל- Windows Media Player.

כדי לבדוק זאת הייתי צריך לעבור על כל התהליכים הרצים במחשב זה באמצעות פונקציית Win32process של WIN32 שמחזירה על כל תהליך שרץ במחשב את שמו, את מספר ה-ID שלו ואת ה-Handle שלו.

חריגה נוספת היא בדיקה האם האתר YouTube רץ במחשב מסוים.

כדי לבדוק זאת הייתי צריך לעבור על כל הכותרות של התהליכים שרצים במחשב זה באמצעות פעולות של מודול WIN32 גם כן.

הפעולה עוברת על כל הכותרות של התהליכים שרצים וכדי למצוא האם המחשב נכנס לאתר YouTube, אמורה להיות כותרת של התהליך של הדפדפן ובשם הכותרת אמורה להופיע המילה YouTube.

דבר נוסף שבודקים מקטגוריית החריגות הוא ה- CPU.

את אחוזי ה- CPU בדקתי באמצעות מודל שנקרא psutil.

במהלך בדיקה זו בדקתי כמה ליבות יש למחשב והצבתי גבולות שיכולות להיות חריגות בהתאם למספר הליבות.

לדוגמה, בדקתי שכאשר במחשב בעל 4 ליבות תוכניות כבדות כמו משחק, יכולות להוריד את אחוזי ה- CPU לרמה מינימלית או להעלות אותם למעל 80 אחוזים, דבר שנחשב כחריגה.

לאחר מכן, בכדי לבדוק האם התהליך שאותו ביקש המשתמש להקליט באחד המחשבים רץ השתמשתי שוב בפונקציית ה- Win32process שהסברתי עליה קודם, ובנוסף לכך בדקתי האם היא לא ממוזערת על ידי פונקציית שנקראת WIN32gui אשר גם היא נמצאת בספריית WIN32.

בדקתי על ידי פעולה בספרייה זו את ה-ID של התהליך שנמצא בתחילת המסך והשוויתי אותו ל-ID של התהליך המבוקש. במידה והתהליך ממוזר או סגור ההקלטה לא מתבצעת.

לאחר הבדיקה האם התהליך רץ ולא ממוזער המערכת מבצעת צילום מסך של המחשב ומעבירה את התמונה לשרת ומשם ל-GUI. צילום המסך נעשה על ידי מודל שנקרא PIL וההעברה מתבצעת באמצעות מודל שנקרא Pickle.