PicChat

שם התלמידה: הדר אשר
בית ספר : הרצוג כפר סבא
ת.ז : 207767005
שם המורה : יוסי זהבי

עבודת גמר

מסלול 14.50

הגנת סייבר

שנת הלימודים תשע"ז

תוכן עניינים

תקציר מנהלים

הפרויקט "PicChat" הוא צ'אט מסרים טקסטואליים כאשר העברת המסרים תעשה על ידי הסתרתם בתוך תמונה, כך שבהסנפה של הפקטות העוברות לא ניתן לדעת במבט ראשון שיש מסר שעובר מעבר לתמונה. רק מי שיודע על הסתרת המסר בתמונה יכול לעלות עליו.
הפרויקט הוא בצורת שרת-לקוח ותומך במספר לקוחות. כל משתמש הוא לקוח שנכנס למערכת כמשתמש עם שם משתמש וסיסמא. הוא כותב את המסר שלו ושולח אותו אל השרת אשר שולח אותו לנמען- לקוח אשר נכנס למערכת כמשתמש אליו ההודעה מיועדת. המסר יוסתר בתוך התמונה באופן הבא- בכל פיקסל בתמונה יוחלף הביט האחרון בבית האחראי על שקיפות הפיקסל בביט מהמחרוזת של המסר. הצד השני יקרא את הביטים האלו ויהפוך אותם חזרה למסר טקסטואלי.

הצעת פרויקט

הפרויקט יהיה צ'אט המעביר מסר טקסטואלי בצורה מוצפנת בתוך קוד של תמונה, כך שבמעקב אחר הפקטות העוברות לא יהיה ניתן לראות את המסר המועבר.

בתוך פרויקט של שרת- לקוח המשתמש בכל צד יכתוב את המסר שאותו הוא רוצה להעביר ויעלה איתו תמונה מסוג מסוים שאקבע. לאחר מכן לפני השליחה לצד השני אצפין את המחרוזת ואכניס אותה לתוך הביטים האחרונים שבבית הרביעי של כל פיקסל (האחראי על שקיפות הפיקסל) שבקוד התמונה. אחר כך אשלח את התמונה. בצד השני אוציא את המסר המוצפן מתוך התמונה, אפענח בחזרה ואציג את המסר.

ממשק המשתמש יהיה צ'אט אליו יהיה צריך להכניס מחרוזת לשליחה. תמונה לשליחה תבחר באופן שרירותי על ידי. כדי לשלוח המשתמש ילחץ על כפתור ה"שלח". כאשר הצד השני ישלח מסר הוא יעלה אוטומטית בפני המשתמש. ההודעות ישמרו עבור כל משתמש בשרת ויגיעו אל המשתמש בעת כניסתו אל הצ'אט והתקשרות עם השרת.

בתוך הפרויקט יהיו תקשורת של שרת-לקוח(פרוטוקול TCP), אבטחת מידע (הצפנה) וממשק משתמש גרפי של צ'אט בשפת פייתון.

אפיון פרויקט

הגדרת הפרויקט:

הפרויקט הוא צ'אט תמונות המעביר מסרים טקסטואליים מוצפנים המוסתרים בתוך קוד של תמונה כך שאם מסניפים את הפקטות המועברות נראה כי מועברות תמונות רגילות בלבד (סטגנוגרפיה). סטגנוגרפיה היא האמנות והמדע של הסתרת מסרים באופן שאף אחד זולת המקבל לא יוכל לראותם או לדעת על קיומם. בניגוד לקריפטוגרפיה, שבה קיום המידע עצמו אינו מוסתר, אלא רק תוכנו. כדי להשתמש בצ'אט המצפין יש להיכנס עם שם וסיסמא השמורה במערכת כמורשית. הפרויקט פותר בעיה של "ציתות" לתקשורת העוברת על ידי אדם באמצע (man in the middle) באמצעות הצפנת המסר ובנוסף הכנסתנו לתוך קוד של תמונה שנראה כביכול תמים ולא מעורר חשד.

אוכלוסיית היעד היא גופים, חברות ואנשים המתקשרים מכמה מחשבים ומעוניינים לשמור על התקשורת סודית ולהבטיח שלא יהיה ניתן להבין את המסרים המועברים על ידי אדם חיצוני שלא מכיר את המערכת.

חקר מוצרים:

כיום ישנן כמה אפליקציות המשתמשות בהצפנת מידע בהעברת מסרים בין משתמשים בצורת צ'אט:

- **Snapchat**

זוהי אפליקציה שמאפשרת לשלוח תמונות שנמחקות עד 10 שניות לאחר שהמקבל רואה אותן. כדי להצפין הודעה, Snapchat משתמשת במפתח הצפנה סימטרי, מכיוון שההודעות הן יחסית גדולות, וניתן להצפין הודעות בעזרת מפתח אסימטרי רק הודעות הקטנות מגודל המפתח עצמו.
הצפנה סימטרית והצפנה אסימטרית הן שתי שיטות הצפנה המשתמשות במפתח ציבורי. ההבדלים העיקריים הם: הצפנה סימטרית טובה בעיקר להצפנת בלוקים גדולים של מידע, והיא מהירה פי כמה מהצפנה אסימטרית בכל קנה מידה ואף עמידה מאוד בפני התקפת טקסט מוצפן נבחר. לעומתה הצפנה אסימטרית מסוגלת לבצע דברים שהצפנה סימטרית אינה מסוגלת, והיא טובה במיוחד לניהול והעברת מפתחות הצפנה ולחתימה דיגיטלית.

<u>בעיות וחסרונות:</u>
השיטה של Snapchat היא שהאפליקציה משתמשת באותו מפתח **סימטרי** בכל הודעה, והמפתח מוטבע בכל אפליקציה בנייד. לכן, כדי לגלות את המפתח, תוקף צריך רק לקמפל את החבילה של האפליקציה. בנוסף, האפליקציה אמנם מצפינה את המידע שהיא מעבירה, אך עדיין במעקב אחר הפקטות העוברות במהלך התקשורת ניתן לזהות היכן מוצפנים המסרים.
בפרויקט זה, אני אסתיר את המסרים בתוך תמונות כך שרק אנשים המודעים ליכולות אלה ובעלי הרשאות מיוחדות יוכלו לנסות ולמצוא את המסר, ומשתמש שאינו מכיר את היכולת הזו ומנסה לחפש את המסר, יתקשה יותר למצוא אותו.

- **WhatsApp**

אפליקציה סלולרית להעברת מסרים מידיים, תמונות, קטעי וידאו וקול.
באפליקציה המסרים מוצפנים באמצעות **הצפנה מקצה לקצה**. הצפנה מקצה לקצה היא מערכת תקשורת שבה רק הצדדים המשתתפים בשיחה יכולים לקרוא את ההודעות. למעשה, נמנעת האזנה של צד שלישי וגורמים שאינם מהווים חלק מן השיחה - כגון: ספקי אינטרנט, ספקי תקשורת סלולרית, ואפילו מספק התקשורת עצמו, מלגשת למפתחות ההצפנה הדרושים כדי לפענח את השיחה. מערכות אלו נועדו למנוע כל ניסיון של מעקב או שינוי של המידע המועבר משום שאף צד שלישי לא יכול לפענח את הנתונים שמועברים או מאוחסנים. במערכת הצפנה מקצה לקצה מפתחות ההצפנה חייבים להיות ידועים רק לצדדים המתקשרים. כדי להשיג מטרה זו, מערכות הצפנה מקצה לקצה יכולות להצפין מידע באמצעות דרכים מגוונות כגון: שימוש ב**פרוטוקול דיפי-הלמן** (Diffie-Hellman).
פרוטוקול דיפי-הלמן הוא פרוטוקול הצפנה אסימטרית המשמש להצפנה הנותן פתרון לבעיית העברת מפתח הצפנה. הפרוטוקול מאפשר לשני משתתפים שלא נפגשו מעולם לשתף ביניהם בעזרת תקשורת ציבורית מפתח סודי לצורך העברת מסרים מוצפנים. הפרוטוקול פוטר מהצורך לשמור מפתח הצפנה סודי לאורך זמן. תחת זאת, המצפין יכול להכין מפתח הצפנה סודי לפי הצורך, להצפין באמצעותו את המסר, ואז להצפין את מפתח ההצפנה עצמו ולשלוח אותו אל המקבל כשהוא מוצפן. כך אין חשש מפני חשיפה.
היתרון של השימוש באפליקציה הוא שהתקשורת מאובטחת באמצעות הצפנת המסרים, ובנוסף, כך ניתן להשתמש באפליקציה זו להעברת מסרים פרטיים.

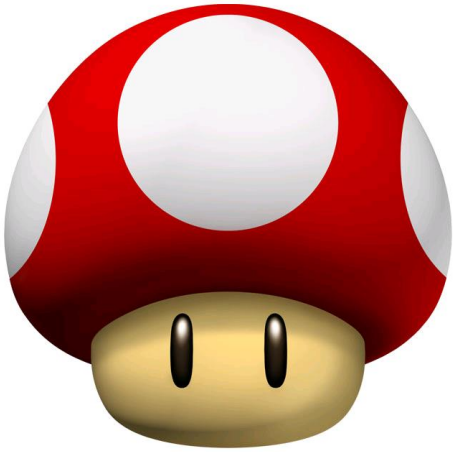
<u>בעיות וחסרונות:</u>
בשיטה של Whatsapp המסרים נשלחים באמצעות מפתח הצפנה **אסימטרי**, בדומה לפרויקט שלי. אולם באמצעות WireShark או אמצעי הסנפה אחר, ניתן לעלות על הפקטות המכילות את המסרים המוצפנים ולנסות לפצח אותם.
בפרויקט זה, המסרים המוצפנים נכנסים לתוך תמונות שהעברתן הן מטרת האפליקציה, וכך כל תקשורת המסרים מוסווית וקרוב לוודאי שרק מי שמורשה ויודע על התעבורה המוצפנת יוכל לעלות עליה.

חקר פיתוחי

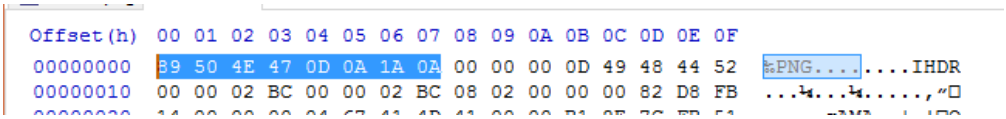
הפרויקט הוא הסוואת מידע טקסטואלי מוצפן על ידי מיזוג המידע עם כמות רבה של מידע לא מוצפן וניצול הביטים הריקים שבקובץ התמונה. טכניקה סטגנוגרפית זו נקראת Null cipher.

פורמט התמונה איתו בחרתי לעבוד בפרויקט זה הוא פורמט PNG מכיוון שהוא פורמט מוכר ונמצא בשימוש על ידי הרבה מהמשתמשים בתמונות שרשת. PNG הוא פורמט תמונה דיגיטלית המשתמש בדחיסה ללא איבוד נתונים.

כדי להדגים כיצד המידע מוצג בתוך קובץ PNG, אשתמש בצילומי מסך מHexEditor של התמונה הבאה:



כל קובץ PNG מתחיל בחתימה באורך 8 בית. ערכיהם של הבתים בהקסדצימלית הם 89 50 4E 47 0D 0A 1A 0A, כל אחד לצורך מטרה מפורשת לדוגמא: 50 4E 47 הם האותיות PNG ב-ASCII, לצורך זיהוי בכלי טקסט.



אחרי הפתיח באים חלקים של הקובץ בגושים .הגושים מכילים את התמונה עצמה ומידע עליה, שחלק ממנו הכרחי לתצוגת התמונה וחלק נועד רק לשפר אותה. הראשונים הם הגושים ההכרחיים והאחרונים גושי העזר. לפני כל גוש יש ארבעה בתים המציינים את אורך הנתונים אשר בו . אחרי שם הגוש באים הנתונים של הגוש עצמו. אחרי כל גוש בא CRC (סכום בדיקה של סיביות) של

שם הגוש והמידע (ללא האורך) באורך 4 בתים, שתפקידו לוודא שלא היו שגיאות בהורדת הקובץ.

גושים הכרחיים

את הגושים ההכרחיים חייבת כל תוכנת פריסה להכיר כדי לקרוא ולהציג את קובץ ה:PNG-

- IHDR- הגוש הראשון. זהו פתיח התמונה, והוא מכיל את אורך התמונה ורוחבה, עומק הסיביות, סוג הצבע, שיטת הדחיסה, שיטת הסינון ושיטת הסירוג.

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000  89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52  ‰PNG........IHDR
00000010  00 00 02 BC 00 00 02 BC 08 02 00 00 00 82 D8 FB  ...¼...¼.....,ˆ□
00000020  14 00 00 00 04 67 41 4D 41 00 00 B1 8E 7C FB 51  .....gAMA..±.|□Q
```

- PLTE- כולל את מפתח הצבעים(PALLETE) בתמונות שיש להן 256 צבעים או פחות.
- IDAT- כולל את מפת הסיביות של התמונה עצמה. לפעמים יש כמה גושי IDAT .

```
00000050  60 00 00 3A 97 00 00 17 6F 97 A9 99 D4 00 01 3C  `..:—...o—©™Ô..<
00000060  A5 49 44 41 54 78 DA EC 9D 3D 68 1B 4B D7 C7 55  ¥IDATxÚì.=h.K ¿U
00000070  6C 31 85 8B 2D 5C A8 48 11 43 8A 18 52 C4 90 22  l1…‹-\¨H.C..R ,"
00000080  86 34 31 A4 88 21 45 0C 29 6E E0 16 C1 A4 08 26  †41¤ˆ!E.)n à.Á¤.&
```

- IEND- חותם את הקובץ. גוש זה ריק מכל נתונים.

גושי עזר

גושי העזר אינם הכרחיים לתצוגה הנכונה של התמונה, אך יש בהם כדי לשפר אותה מאוד. בנוסף יש גושי טקסט להוספת מידע על התמונה בשפת אנוש.

דוגמאות:

- tRNS- ערכי שקיפות לצבעים. בתמונות גוני אפור וצבע RGB יש בגוש זה ערך אחד המציין צבע אחד בתמונה שהוא שקוף לגמרי. בתמונות עם מפתח צבעים (סוג אחר של צבעים) יכולים להיות ערכים למפתחות רבים, וב-255 דרגות של שקיפות.
- pHYs- הגודל הפיזי של הפיקסלים. גוש זה מציין את מספר הפיקסלים למטר בתמונה, אופקית ואנכית. למשל ערך של 3780 3780 שווה בקירוב ל-96 פיקסלים לאינץ'. הוא גם יכול לציין את היחס בין אורך כל פיקסל לרוחבו.
- cHRM- פרמטרים לניהול צבעים. גוש זה מסייע לקביעת מרחב הצבעים של התמונה.

```
00000020  14 00 00 00 04 67 41 4D 41 00 00 B1 8E 7C FB 51  .....gAMA..±.|□Q
00000030  93 00 00 00 20 63 48 52 4D 00 00 7A 25 00 00 80  "... cHRM..z%..€
```

- gAMA- הגאמה של התמונה, כלומר היחס בין הדגימות שבתמונה לבין עוצמת פלט התצוגה. בתוצאה הנראית לעין: בהירות התמונה.

```
00000010  00 00 02 BC 00 00 02 BC 08 02 00 00 00 82 D8 FB  ...¼...¼.....,"□
00000020  14 00 00 00 04 67 41 4D 41 00 00 B1 8E 7C FB 51  .....gAMA..±.|□Q
00000030  93 00 00 00 20 63 48 52 4D 00 00 7A 25 00 00 80  "... cHRM..z%..€
```

- iCCP- פרופיל צבע המוטמע בתוך התמונה. זהו עזר יותר מתקדם לניהול צבעים מאשר גוש.cHRM
- sRGB- גוש המציין שהצבעים שבתמונה הם לפי מרחב הצבעים הנפוץ sRGB. הוא מכיל, בנוסף ,בית אחד המציין את ייעוד תצוגת התמונה (תצלום, לוגו, דיאגרמה וכו').
- sBIT- מספר הסיביות לערוץ שבתמונה המקורית. למשל גוש sBIT עם ערכים של 6 עבור שלושת צבעי היסוד מציין שהתמונה המקורית קודדה במרחב עוצמת צבעים של 0 עד 63 לכל ערוץ, במקום המרחב 0 עד 255 הרגיל. מכיוון שפורמט PNG תומך רק בערכים של חזקות של 2 למספר הסיביות לערוץ, גוש sBIT מאפשר לשחזר עומקי סיביות לא שגרתיים ללא איבוד נתונים.

סוג הצבע

סוגי הצבע שפורמט PNG מסוגל לאחסן הם אלה:

- תמונת גוני אפור (סוג צבע 0)
- תמונת RGB או truecolor (סוג צבע 2)
- תמונות גוני אפור עם ערוץ אלפא (סוג צבע 4)
- תמונת RGB עם ערוץ אלפא (סוג צבע 6)
- תמונה עם מפתח צבעים (סוג צבע 3)

שקיפות

פורמט PNG מאפשר הן שקיפות בינארית, כלומר צבע אחד שקוף לגמרי, או שקיפות משתנה (צבעים רבים בדרגות שקיפות שונות), בכל סוגי הצבעים. השקיפות המשתנה, הקרויה גם אלפא, מצוינת בתמונות גוני אפור ו - RGB- באמצעות ערוץ נפרד.

סביבת פיתוח

שפת התכנות בה אפתח את הפרויקט- Python
מערכת ההפעלה עליה ירוץ הפרויקט- WINDOWS 7
פלטפורמת הפרויקט- Personal Computer

ממשק גרפי

צד לקוח-

תפריט ראשי

בתפריט ישנן שתי תיבות טקסט, הראשונה לשם משתמש והשנייה לסיסמא. לאחר שהמשתמש מכניס את הנתונים שלו הוא לוחץ על כפתור ה- Sign in כדי להיכנס לצ'אט. לאחר מכן, במידה והמשתמש רשום (שם משתמש וסיסמא נכונים) המשתמשים עוברים למסך אנשי הקשר.

אנשי קשר

במסך זה יש שני כפתורים- האחד יציאה מהתוכנית והשני בשם אנשי קשר אשר פותח חלון עם רשימת אנשי הקשר. בעת לחיצה על כפתור זה החלון הנוכחי אינו נסגר.

רשימת אנשי קשר

במסך זה על המשתמש לבחור אל מי הוא רוצה לדבר, לצ'וטט. על המשתמש לבחור בלחיצת כפתור את אחד מאנשי הקשר מהרשימה וכדי להתחיל את הצ'אט הוא צריך ללחוץ על הכפתור start chat. לאחר בחירת איש הקשר יפתח חלון נוסף עם הצ'אט עם איש הקשר הנבחר.

צ'אט

על המשתמש להכניס את המסר שאותו הוא רוצה לשלוח לאיש הקשר איתו הוא מדבר. לאחר ההודעה המשתמש צריך ללחוץ על כפתור השליחה בשביל לשלוח את המסר.

צד שרת-
בצד השרת לא יהיה ממשק גרפי אלא המשתמשים הפעילים ורשימות המשתמשים הרשומים (רגילים ומיוחדים) יהיו שמורים במסמכי טקסט.

ספריות מיוחדות

- os- מספק ממשק אחיד למספר פעולות של מערכת ההפעלה. ספרייה זו טוענת אוטומטית את מודול היישום הנכון כאשר היא מיובאת.

- Thread- ספרייה המאפשרת הרצת מספר תהליכי משנה דרך אותה תכנית. היתרון של שיטה זו הוא שתהליכי המשנה חולקים אותו מרחב זיכרון עם התהליך הראשי (תהליך האב). בנוסף, מכיוון שהם תהליכי משנה הם לא מצריכים הקצאת מרחב זיכרון חדש אלא משתמשים בזה הקיים (של תהליך האב) וכך יותר "זולים" למעבד מבחינת הקצאת משאבים.
בעזרת ספרייה זו ניתן לבצע מימוש של שרת לקוח multi thread. שרת לקוח multi thread זה שרת מקבילי כלומר, כזה שמאפשר טיפול של כמה לקוחות במקביל כאשר כל פעם שלקוח מבקש להתחבר לשרת, תהליך האב בשרת (התהליך הראשי) מקצה עבורו תהליך בן (thread) שמטפל בו באופן ספציפי, במקביל ללקוחות אחרים שמטופלים בידי תהליכי בן אחרים.

- Socket- מודול לממשק עם תכנות רשת ברמת-בסיס.
הממשק של socket בפייתון הוא תעתיק של ממשק קריאות המערכת (system calls) של יוניקס, באופן שיתאים לסביבה מוכוונת העצמים של פייתון: הפונקציה ()socket מחזירה אובייקט socket, והמתודות של אובייקט זה מיישמות את קריאות המערכת השונות. הפרמטרים שמקבלות המתודות השונות הם ב"רמה גבוהה" יותר של שימוש מאשר אלו של קריאות המערכת ב-C, כך שלמשל בפקודות הכתיבה והקריאה הקצאת המאגר הזמני (buffer) מתבצעת אוטומטית.

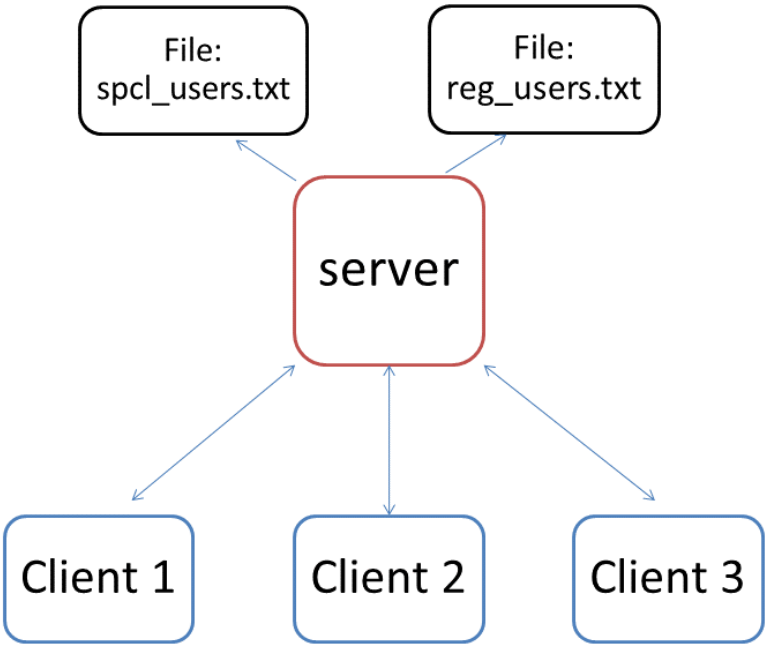
- wxPython- ערכת כלים גרפית עבור שפת התכנות python. היא מאפשרת למתכנתי python ליצור תכניות עם ממשק משתמש גרפי חזק ופונקציונלי בפשטות ובקלות. הוא מיושם כמודול הרחבה של פייתון (native code) שעוטף את ספריית ה- wxWidgets GUI, שכתובה ב- ++C.wxPython הוא

קוד פתוח, כלומר זה חינם לשימוש וקוד המקור זמין עבור כל אדם שיכול להסתכל עליו ולשנות אותו. כל אחד יכול לתרום תיקונים או שיפורים לפרויקט.wxPython היא ערכת כלים חוצה פלטפורמות. משמעות הדבר היא שאותה תכנית תרוץ על מספר פלטפורמות ללא שינוי. נכון לעכשיו, הפלטפורמות הנתמכות הן 32 סיביות של Microsoft Windows, רוב מערכות יוניקס או דמוי יוניקס ו- Macintosh OS X.

- Time- מודול זה מספק מספר פונקציות להתמודד עם תאריכים בפרק זמן של יום. זוהי שכבה דקה על גבי ספריית זמן הריצה של C.תאריך ושעה נתונים יכולים להיות מיוצגים כ- tuple של זמן.

- PIL (Python Image Library)-ספרייה המוסיפה יכולות עיבוד תמונה. ספריה זו תומכת בפורמטים רבים של קבצים רבים, ומספקת יכולות עיבוד גרפיקת תמונה חזקות. הספרייה קוראת ועורכת קבצי PNG בהם אני משתמשת בפרויקט.

חלוקה פונקציונלית



בין לקוח לשרת- הלקוח מבקש להתחבר לצ'אט עם לקוח המחובר כעת . בנוסף, הלקוח מבקש מהשרת להעביר מסר אל אותו הלקוח.

בין השרת ללקוחות- השרת והלקוח מתחברים זה לזה והלקוח נכנס לרשימת המשתמשים על פי סוג המשתמש. השרת מקבל את הבקשות של הלקוחות להעברת המסרים ומעביר אותם לפי לקוח היעד של המסר.

בין השרת לקבצים- השרת ניגש אל הקבצים המכילים את רשימת הלקוחות המיוחדים והרגילים המחוברים אליו באותו הרגע. השרת יכול גם לעדכן את הרשימות ברגע שמתחבר לקוח חדש או להסיר לקוח שהתנתק מהרשימה. בנוסף השרת ניגש את הקבצים המכילים את רשימות המשתמשים השמורים במערכת כרגילים ומיוחדים. לשרת אין הרשאות לשנות רשימות אלו.

פרוטוקול תקשורת

המערכת תשתמש בפרוטוקול TCP/IP לשליחת ההודעות בין חלקי המערכת.
התקשורת בין בלקוחות תתבצע דרך השרת והוא ינתב את ההודעות.
יוגדרו סוגים שונים של הודעות במערכת, והם:

- הודעת פקודה – שליחת פעולה לביצוע
- הודעת תגובה – תגובה לפקודה – החזרת מידע בהתאם לפקודה שבוקשה.

<u>צד לקוח:</u>

1. **הודעת חיבור לשרת**- ההודעה תכלול מילת מפתח המסמלת התחלת חיבור ובנוסף שם משתמש וסיסמא שהלקוח רוצה להתחבר איתם.
2. **שליחת הודעה למשתמש**- הודעה הכוללת מילת מפתח ואת שם המשתמש הנמען, ותמונה שבתוכה מוסתרת ההודעה אותה הלקוח מבקש לשלוח. התמונה בפורמט PNG.

<u>צד שרת:</u>

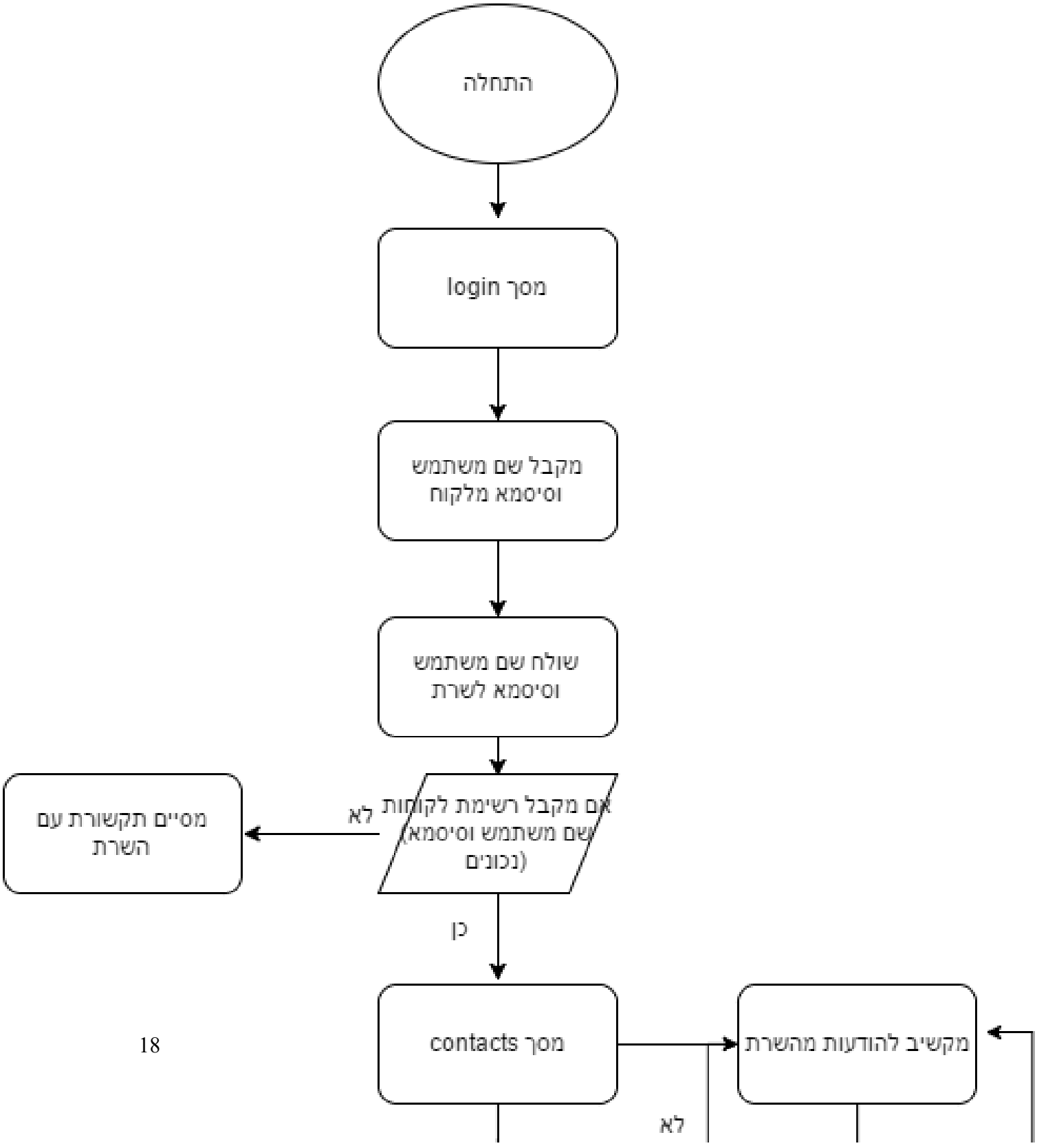
1. **הודעת login שגוי**- הודעה המסיימת את התקשורת. במקרה ששם המשתמש והסיסמא לא שמורים בשרת.
2. **הודעת רשימת משתמשים מחוברים**- הודעה שכביכול מאשרת את החיבור לצ'אט. היא מכילה את שמות המשתמשים המחוברים כעת לשרת איתם הלקוח יכול להתחיל לדבר.
3. **שליחת הודעה למשתמש**- הודעה הכוללת מילת מפתח ואת שם המשתמש הנמען, ותמונה שבתוכה מוסתרת ההודעה אותה הלקוח מבקש לשלוח. התמונה בפורמט PNG. הודעה זו מתקבלת מלקוח המוען ונשמרת אצל השרת עד אשר הנמען מתחבר.

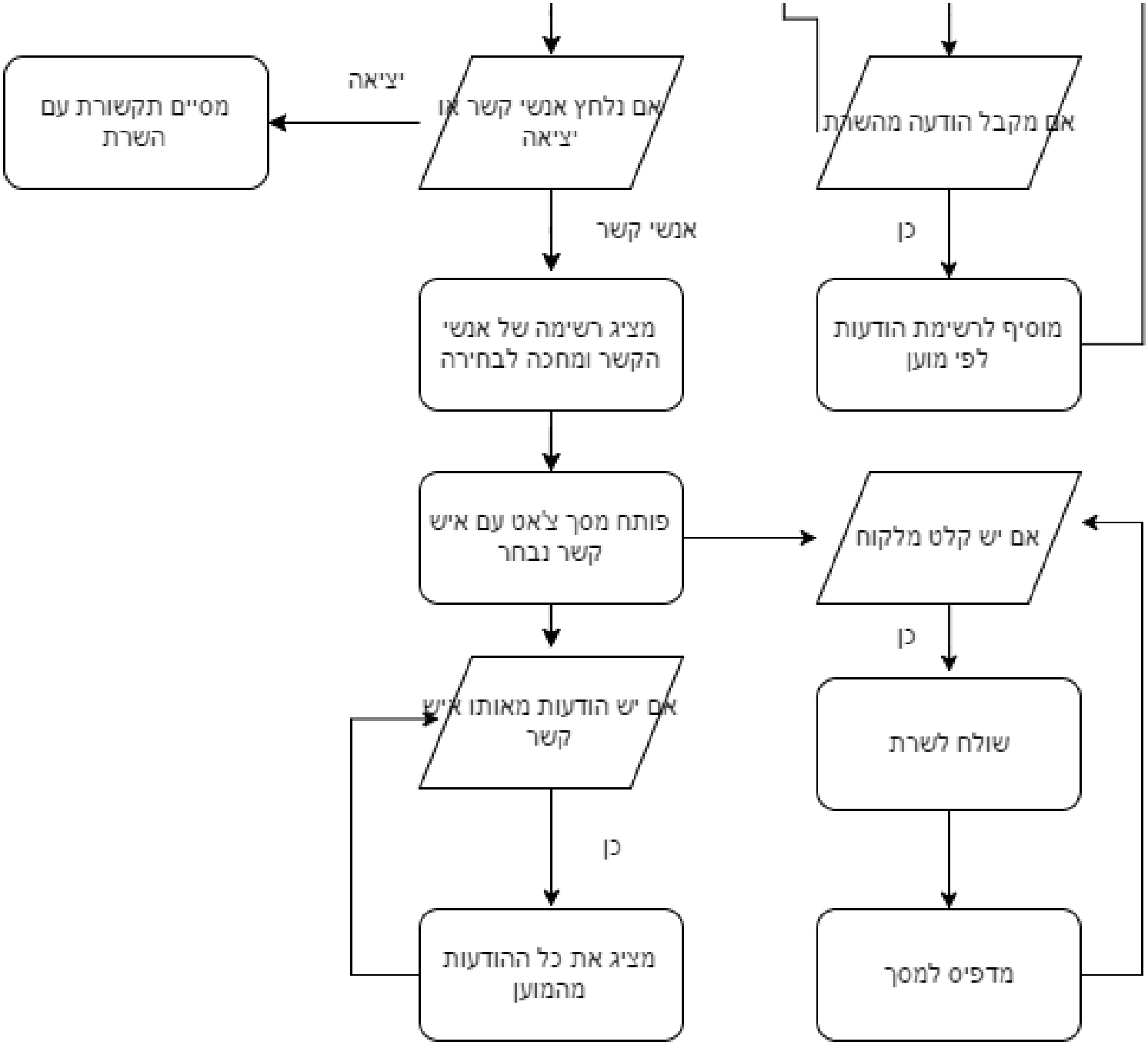
הגבלות הפרויקט

- הפרויקט מוגבל לעבודה רק בסביבת windows.
- את הפרויקט לא ניתן להוריד מהרשת או בדרכים אחרות, אלא רק דרך ה-DOK והעברת הקוד למחשב באופן ידני.
- לפרויקט אין בסיס נתונים בו הוא מאחסן את הרשימות, אלא הן נשמרות בקבצי טקסט.
- בצד השרת של הפרויקט לא יהיה ממשק משתמש גרפי הידידותי למשתמש.

תרחישים עיקריים

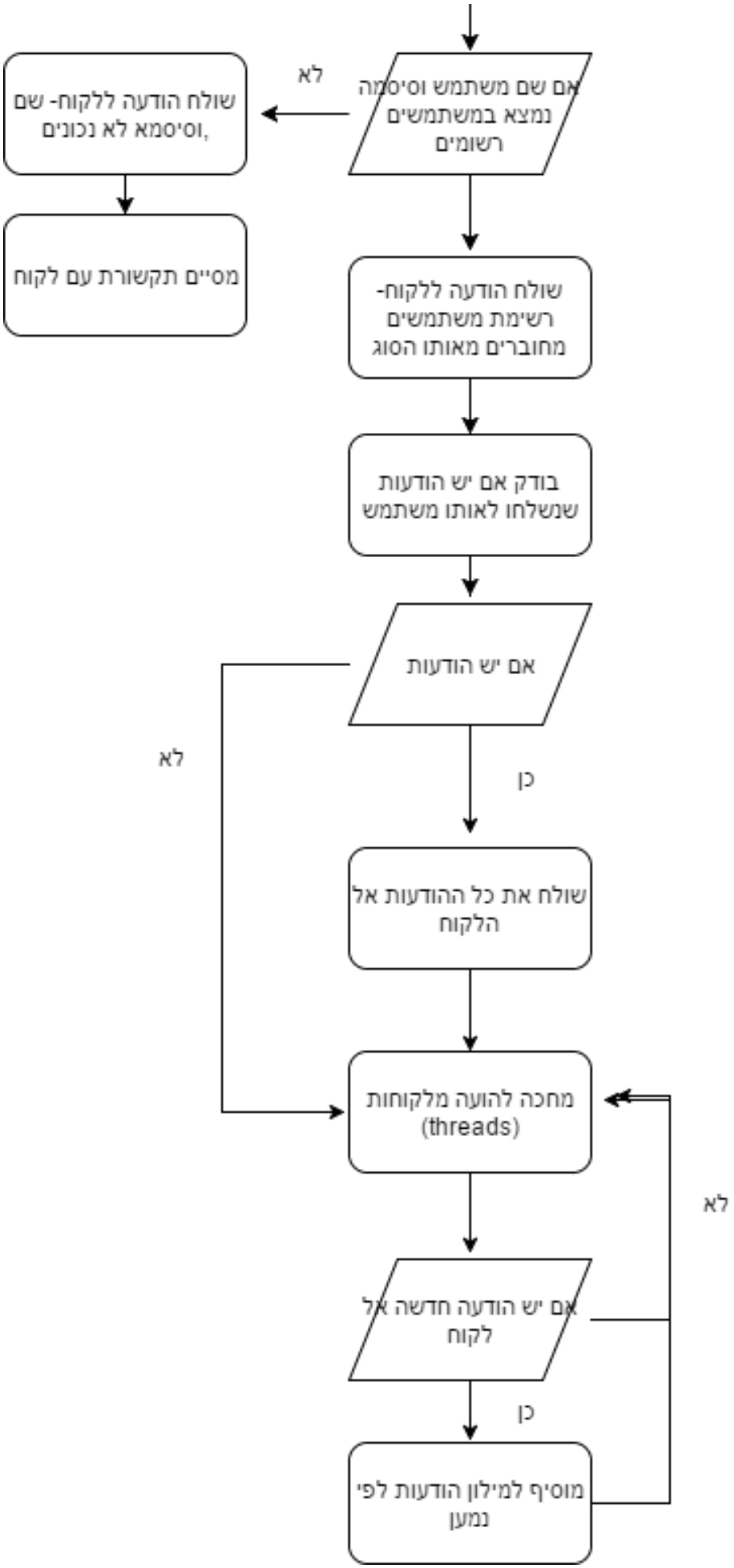
צד לקוח:




אם מקבל הודעה מהשרת
כן
מוסיף לרשימת הודעות לפי מוען
אם נלחץ אנשי קשר או יציאה
יציאה
מסיים תקשורת עם השרת
אנשי קשר
מציג רשימה של אנשי הקשר ומחכה לבחירה
פותח מסך צ'אט עם איש קשר נבחר
אם יש קלט מלקוח
כן
שולח לשרת
מדפיס למסך
אם יש הודעות מאותו איש קשר
כן
מציג את כל ההודעות מהמוען

צד שרת:


התחלה
קריאת משתמשים רשומים מקבצי משתמשים רגילים ומיוחדים
תחילת תקשורת (socket,bind)
מחכה לחיבור מלקוחות (listen)
אם מתחבר לקוח חדש
לא
כן
מקבל שם משתמש וסיסמה מלקוח


אם שם משתמש וסיסמה נמצא במשתמשים רשומים
לא
שולח הודעה ללקוח- שם וסיסמא לא נכונים,
מסיים תקשורת עם לקוח
שולח הודעה ללקוח- רשימת משתמשים מחוברים מאותו הסוג
בודק אם יש הודעות שנשלחו לאותו משתמש
אם יש הודעות
לא
כן
שולח את כל ההודעות אל הלקוח
מחכה להועה מלקוחות (threads)
לא
אם יש הודעה חדשה אל לקוח
כן
מוסיף למילון הודעות לפי נמען

עיצוב תוכנה

רקע

מטרת מסמך זה היא הבנה מעמיקה יותר של הפרויקט ותכנונו מבחינה טכנית יותר. מסמך זה בא להנחות איך לכתוב את הפרויקט בשיטת עיצוב מטה-מעלה, כלומר פירוק הפרויקט לחלקים קטנים ובנייתם בנפרד תחילה, ורק לאחר מכן חיבורם לפרויקט אחד. שיטה זו מאפשרת שימוש חוזר בקוד ומקלה על בדיקתו. בנוסף מסמך זה מאפשר למתכנתים אחרים להבין את הפרויקט.

פרויקט זה הוא צ'אט המעביר מסרים טקסטואליים המוסתרים בתוך קוד של תמונה כך שאם מסניפים את הפקטות המועברות נראה כי מועברות תמונות רגילות בלבד (סטגנוגרפיה). סטגנוגרפיה היא האמנות והמדע של הסתרת מסרים באופן שאף אחד זולת המקבל לא יוכל לראותם או לדעת על קיומם. בניגוד לקריפטוגרפיה, שבה קיום המידע עצמו אינו מוסתר, אלא רק תוכנו.

שימוש בסטגנוגרפיה בקובץ PNG תתבצע על ידי ביטול השקיפות של התמונה ושימוש בבית האחראי על כך. כל פיקסל מיוצג על ידי ארבעה בתים- RGBובית של שקיפות. אשתמש בבית האחרון בשביל המידע המוצפן.

כדי להשתמש בצ'אט יש להיכנס עם שם וסיסמא השמורה במערכת כמורשית. רשימות המורשים תשמר בקובץ טקסט חיצוני לקוד. הפרויקט פותר בעיה של "ציתות" לתקשורת העוברת על ידי אדם באמצע (man in the middle) באמצעות הכנסת המסר לתוך קוד של תמונה שנראה כביכול תמים ולא מעורר חשד. כלומר, כאשר מסניפים את הפקטות המכילות את התמונה לא ניתן לראות ישר את המסר המוצפן, אלא רק אחרי סריקה של קוד התמונה, בידיעה שיש בתוך הקוד מסר מוסתר.

סביבה

את התכנית אכתוב בשפת Python מכיוון שזו שפה בעלת תחביר ללמוד ולהבין ונוחה לתכנות. קל לכתוב בה פונקציות יעילות המקלות על כתיבת הפרויקט.
בנוסף זו שפה נפוצה שתהפוך את הפרויקט למובן ע"י תכניתנים.
הפלטפורמה בה אשתמש היא PC(Private Computer),
ומערכת הפעלה היא windows 7.

מודולים חיצוניים

- wxPython- ערכת כלים גרפית עבור שפת התכנות python. היא מאפשרת למתכנתי python ליצור תכניות עם ממשק משתמש גרפי חזק ופונקציונלי בפשטות ובקלות. הוא מיושם כמודול הרחבה של פייתון (native code) שעוטף את ספריית ה- GUI wxWidgets, שכתובה ב- ++C.wxPython הוא קוד פתוח, כלומר זה חינם לשימוש וקוד המקור זמין עבור כל אדם שיכול להסתכל עליו ולשנות אותו. כל אחד יכול לתרום תיקונים או שיפורים לפרויקט.wxPython היא ערכת כלים חוצה פלטפורמות. משמעות הדבר היא שאותה תכנית תרוץ על מספר פלטפורמות ללא שינוי. נכון לעכשיו, הפלטפורמות הנתמכות הן 32 סיביות של Microsoft Windows, רוב מערכות יוניקס או דמוי יוניקס ו- Macintosh OS X.

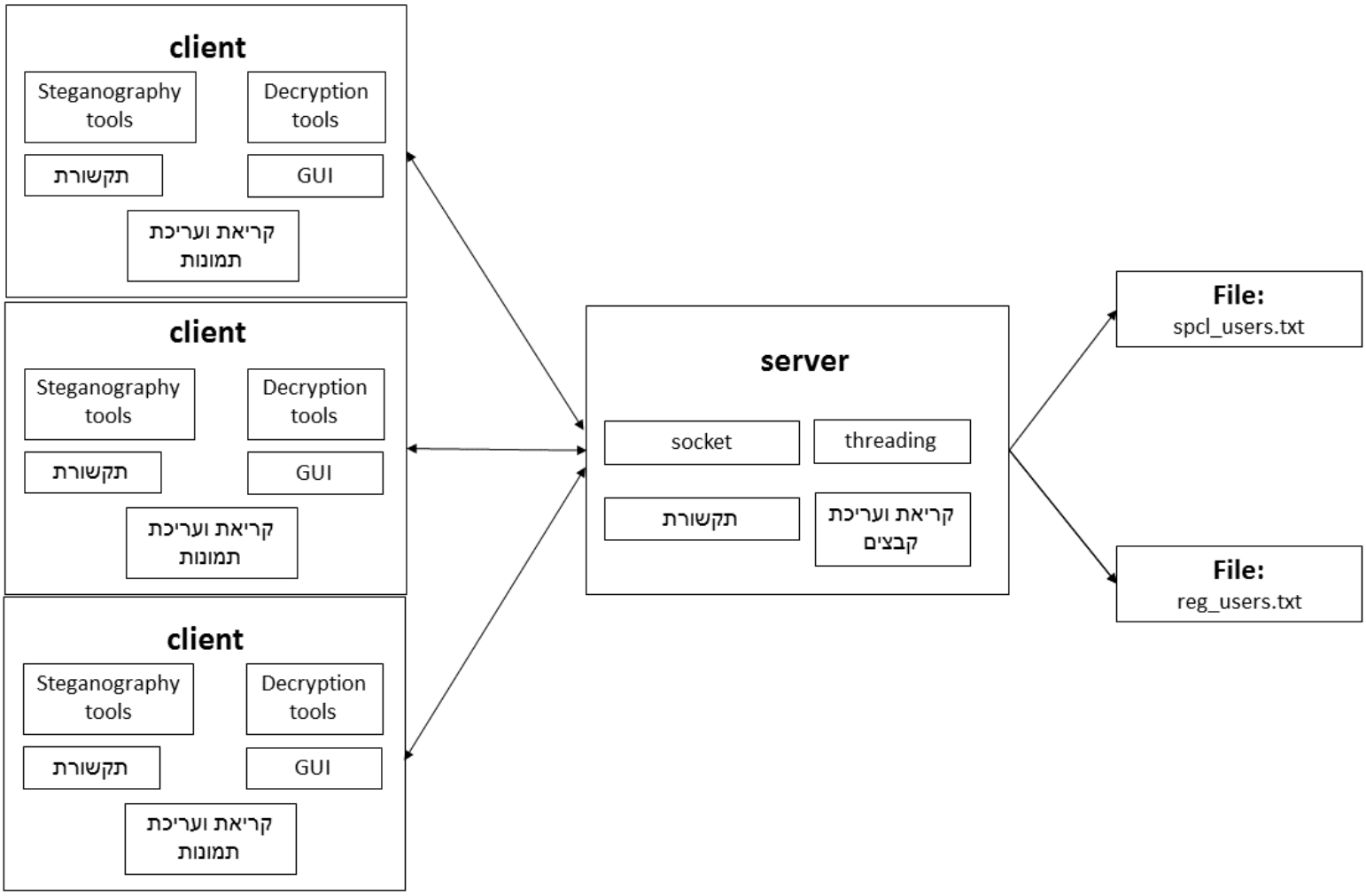
- PIL (Python Image Library)-ספרייה המוסיפה יכולות עיבוד תמונה. ספריה זו תומכת בפורמטים רבים של קבצים רבים, ומספקת יכולות עיבוד גרפיקת תמונה חזקות. הספרייה קוראת ועורכת קבצי JPEG בהם אני משתמשת בפרויקט. ייתכן ואשתמש גם בספריית Pillow המבוססת על ספריית PIL ומתעסקת גם כן בעיבוד תמונות מסוגים שונים, אך היא ידידותית יותר למשתמש.

- Socket- מודול לממשק עם תכנות רשת ברמת-בסיס.
  הממשק של socket בפייתון הוא תעתיק של ממשק קריאות המערכת (system calls) של יוניקס, באופן שיתאים לסביבה מוכוונת העצמים של פייתון: הפונקציה ()socket מחזירה אובייקט socket, והמתודות של אובייקט זה מיישמות את קריאות המערכת השונות. הפרמטרים שמקבלות המתודות השונות הם ב"רמה גבוהה" יותר של שימוש מאשר אלו של קריאות המערכת ב-C, כך שלמשל בפקודות הכתיבה והקריאה הקצאת המאגר הזמני (buffer) מתבצעת אוטומטית.

- Thread- ספרייה המאפשרת הרצת מספר תהליכי משנה דרך אותה תכנית. היתרון של שיטה זו הוא שתהליכי המשנה חולקים אותו מרחב זיכרון עם התהליך הראשי (תהליך האב). בנוסף, מכיוון שהם תהליכי משנה הם לא מצריכים הקצאת מרחב זיכרון חדש אלא משתמשים בזה הקיים (של תהליך האב) וכך יותר "זולים" למעבד מבחינת הקצאת משאבים.

בעזרת ספרייה זו ניתן לבצע מימוש של שרת לקוח multi thread. שרת לקוח multi thread זה שרת מקבילי כלומר, כזה שמאפשר טיפול של כמה לקוחות במקביל כאשר כל פעם שלקוח מבקש להתחבר לשרת, תהליך האב בשרת (התהליך הראשי) מקצה עבורו תהליך בן (thread) שמטפל בו באופן ספציפי, במקביל ללקוחות אחרים שמטופלים בידי תהליכי בן אחרים.

ארכיטקטורת מערכת



מודולים

- **steg.py**

המודול יעסוק בהסתרה ובקריאה של מסרים טקסטואליים בתמונות מסוג PNG. בעזרת מודול זה יהיה אפשר לקרוא תמונות PNG, לכתוב אל קובץ תמונות PNG, להסתיר מסרים טקסטואליים ולקרוא אותם חזרה.

- **server.py**

השרת בו אשתמש בפרויקט זה יכלול תקשורת מקבילה ואסינכרונית. הוא יהיה multi-client וישתמש ב threads לשם כך. השרת יקבל הודעות מלקוחות וישלח הודעות מלקוחות ללקוחות אחרים. בנוסף הוא יקשר בין הלקוחות.

- **client.py**

הלקוח יכלול בתוכו ממש גרפי למשתמש ותקשורת עם השרת. הלקוח ישלח אל הלקוח הודעות לפי קלט משתמש ויציג הודעות למשתמש שקיבל מהלקוח. בנוסף הלקוח יקרא תמונות מהמחשב ויצפין ויסתיר בתמונה מסרים טקסטואליים בעזרת המודולים שלמעלה.

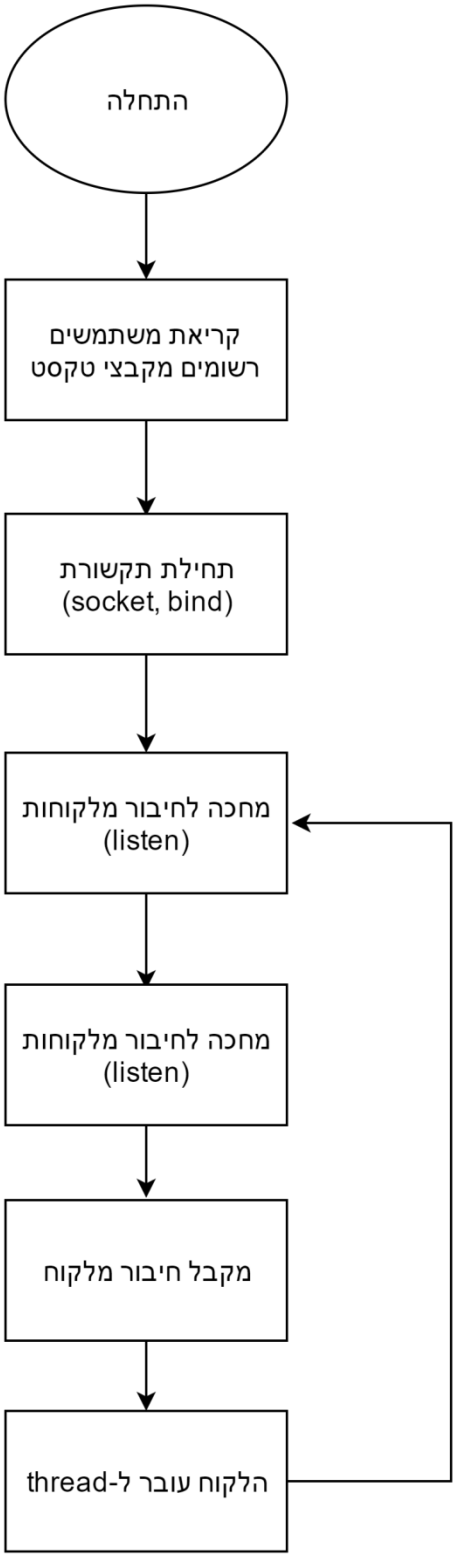
- **PicChat.py**

זהו מודול הגרפיקה בו משתמש הלקוח. דרך מודול זה המשתמש מריץ את התקשורת עם השרת (באמצעות מחלקת client), קולט מידע של המשתמש (כפתורים ותיבות טקסט) ומציג את ההודעות והכפתורים בצורה גרפית הנוחה למשתמש, מסתיר את המסרים הטקסטואליים בתמונות באמצעות המודול steg.py .

פעולות ראשיות

- **main of server**

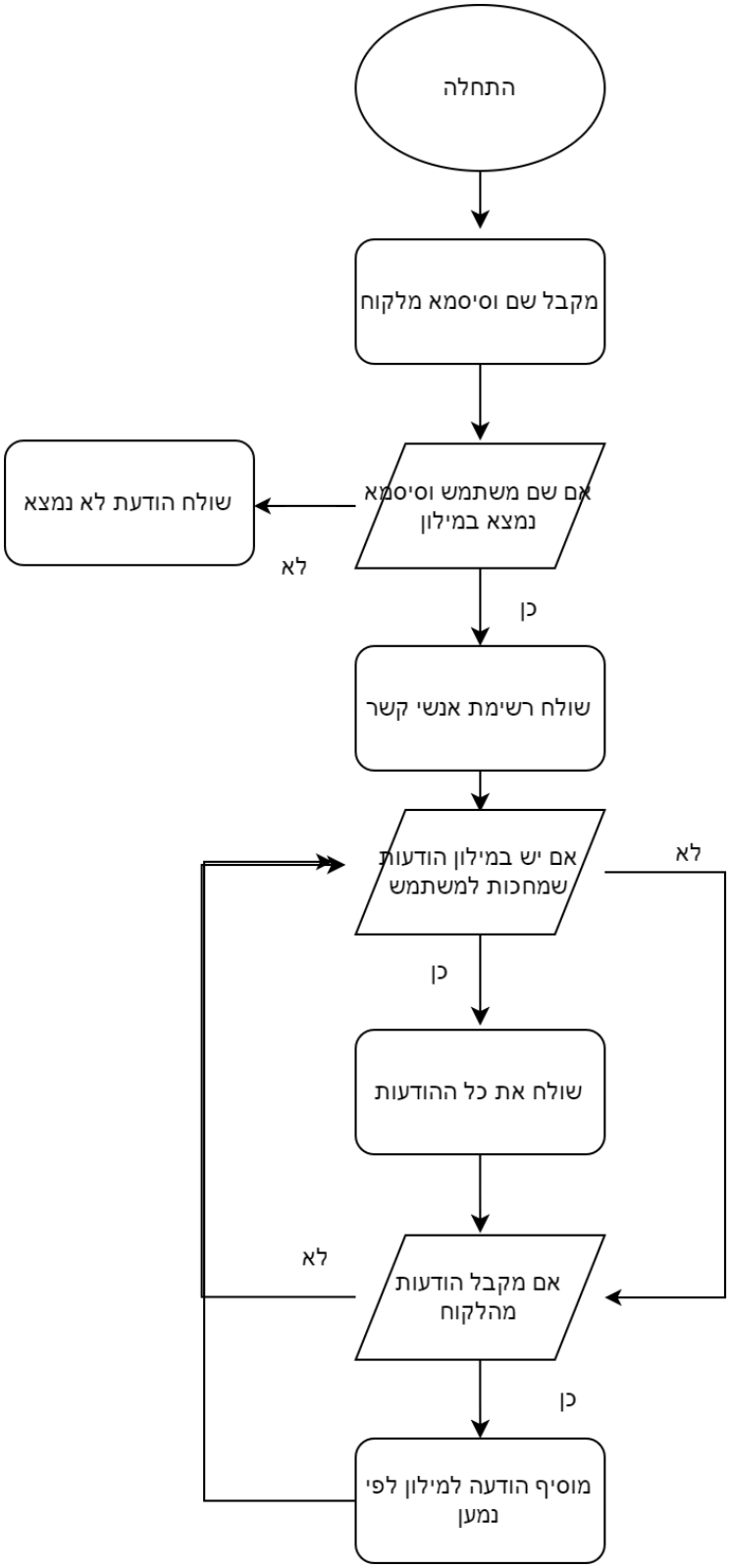
בפעולה הראשית של השרת מתחיל את התקשורת ומאזין ללקוחות שרוצים להתחבר אליו. לכל לקוח הוא פותח thread והמשך הטיפול בכל לקוח



נעשה בצורה מקבילית באמצעות threads.

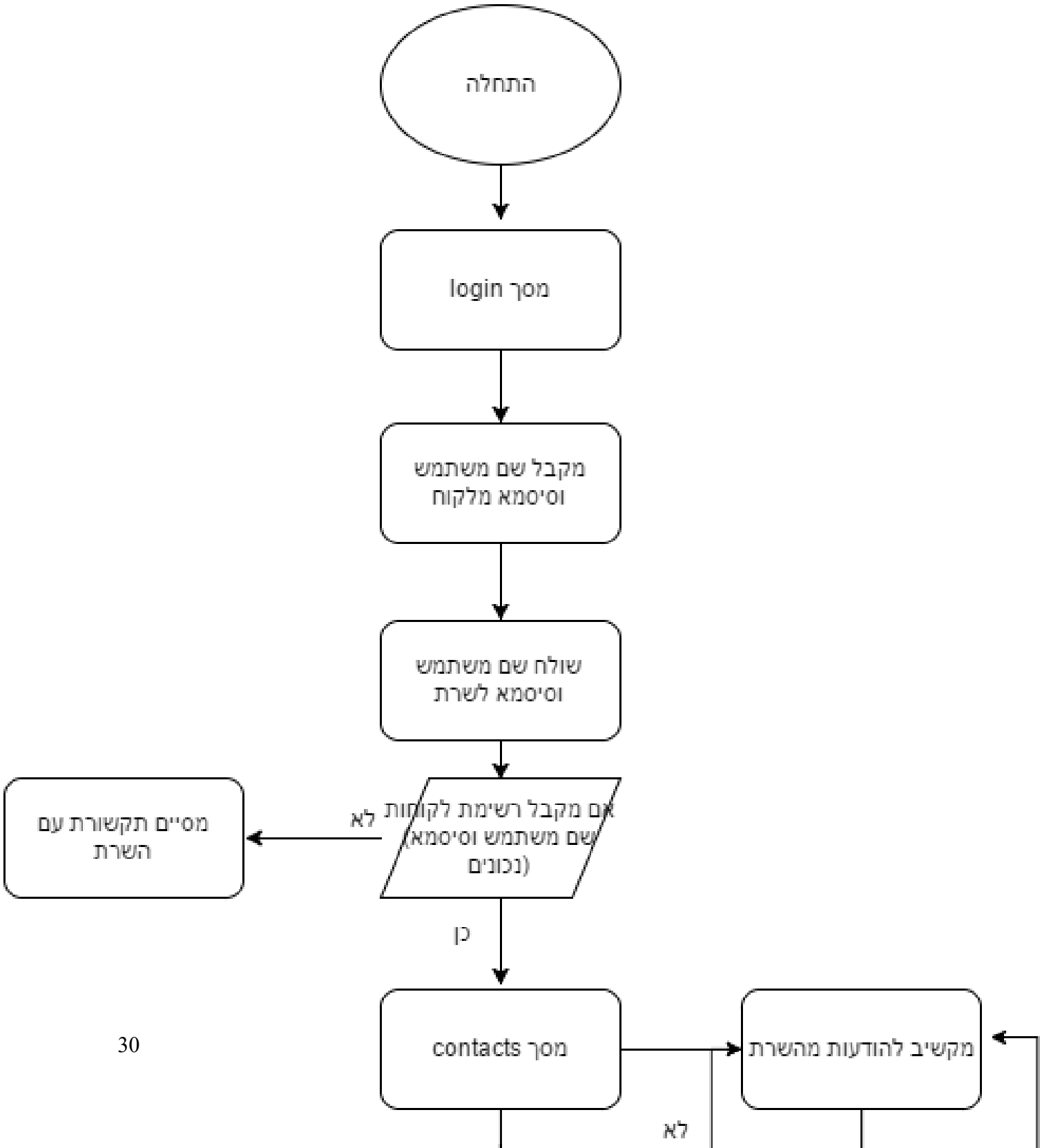
- **main of thread**

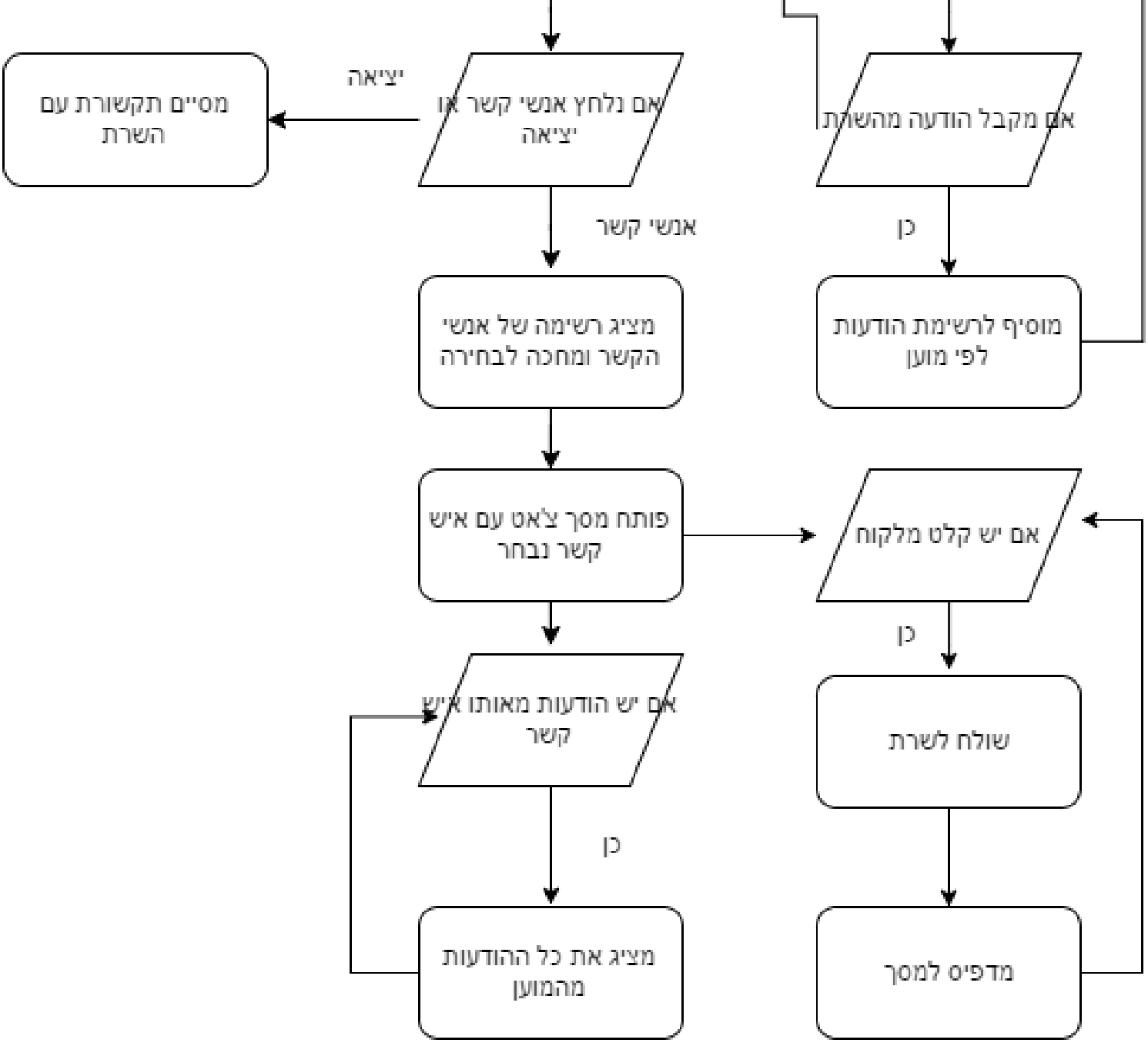
thread מטפל בלקוח אחד בלבד. הוא בודק האם הוא מורשה להיכנס למערכת באמצעות שם משתמש וסיסמא שמורים, מקבל ממנו אל משתמשים אחרים ושולח אליו הודעות ממשתמשים אחרים.



- **main client**

הפעולה הראשית של הלקוח כוללת ממשק משתמש גרפי ותקשורת עם השרת. הלקוח מתחבר לשרת, שולח בקשת התחברות למערכת ע"י שם משתמש וסיסמא איתה המשתמש ביקש להתחבר, שולח ומקבל הודעות מהלקוח איתו הוא מדבר. הלקוח מקבל קלט מהמשתמש ופועל לפיו.




אם מקבל הודעה מהשרת
כן
מוסיף לרשימת הודעות לפי מוען
אם יש קלט מלקוח
כן
שולח לשרת
מדפיס למסך
אם נלחץ אנשי קשר או יציאה
יציאה
מסיים תקשורת עם השרת
אנשי קשר
מציג רשימה של אנשי הקשר ומחכה לבחירה
פותח מסך צ'אט עם איש קשר נבחר
אם יש הודעות מאותו איש קשר
כן
מציג את כל ההודעות מהמוען

מחלקות

steg	
hide_text(img_name,txt)	פעולה המקבלת שם תמונה וטקסט, פותחת את התמונה ומסתירה את הטקסט בתוכה
unhide_text(img_name)	פעולה המקבלת שם של תמונה ומוציאה את הטקסט המוחבא בה ומחזירה אותו
get_img_data(img_name)	פעולה המקבלת שם של תמונה, פותחת אותה ומחזירה את התמונה בצורה טקסטואלית
make_img_file(data)	פעולה המקבלת מידע טקסטואלי של תמונה, פותחת קובץ חדש מסוג PNG וכותבת לתוכו את המידע של התמונה (מכין תמונה חדשה)
encode_image(img,msg)	מקבל משתנה מסוג תמונה ומחרוזת ומסתיר בתוך התמונה את המחרוזת. מחזיר את התמונה החדשה
decode_image(img)	מקבל תמונה עם מסר מוסתר בתוכה ומחזיר את המחרוזת המוצפנת בתוכה
String2bin(msg)	הפעולה מקבלת מסר טקסטואלי ומחזירה מחרוזת של ערכים בינאריים
Bin2string(bin_msg)	הפעולה מקבלת מחרוזת של ערכים בינאריים ומחזירה מחרוזת טקסטואלית
Hex_to_rgba(hexcode)	פעולה המקבלת מספר הקסדצימלי באורך 4 בתים והופכת אותו לtuple של RGBA
Rgba_to_hex(r,g,b,a)	מקבלת ארבעה בתים RGBA(כל אחד בית, ומחזירה מחרוזת הקסדצימלית

קבצים

- **reg_users.txt-** הקובץ יכיל את המידע על המשתמשים הרשומים הרגילים של הצ'אט. בקובץ ישמר עבור כל משתמש רשום שם המשתמש והסיסמא. בין השדות תהיה הפרדה בעזרת הסימן ~. בין משתמש למשתמש תהיה הפרדה של הסימן | וירידת שורה, כלומר \n|.
- **spcl_users.txt-** הקובץ יכיל את המידע על המשתמשים הרשומים המיוחדים של הצ'אט. בקובץ ישמר עבור כל משתמש רשום שם המשתמש והסיסמא. בין השדות תהיה הפרדה בעזרת הסימן ~. בין משתמש למשתמש תהיה הפרדה של הסימן | וירידת שורה, כלומר \n|.

פעולות עזר

- **put_msg_in_async_msgs(data,socket)**
הפעולה תהיה פעולת עזר לשרת. השרת צריך לקבל ולשלוח הודעות מלקוחות ללקוחות אחרים. לשם כך הוא צריך לשמור את ההודעות במילון לפי socket שהוא ייחודי לכל תקשורת מול כל לקוח. פעולה זו תכניס את ההודעות למילון ותשמור את ההודעות לכל לקוח במערך.

- **send_with_size(socket,data)**
הפעולה תקבל socket לשלוח אליו את ההודעה ואת ההודעה לשליחה. היא תשלח את ההודעה ותדפיס אותה למסך.
- **recieve_by_size(socket)**
הפעולה תקבל socket לקבל ממנו ההודעה. היא תקבל את ההודעה ותדפיס אותה למסך.

תקשורת

הפרויקט יעבוד במוד של שרת-לקוח. השרת יהיה מקבילי ויתמוך בריבוי לקוחות. אעשה זאת באמצעות שימוש במודול thread של python המאפשר הרצת תת תהליכים.

צורת ההתקשרות תהיה אסינכרונית מכיוון שעל כל אחד מהצדדים להאזין כל הזמן לתקשורת ולהיות זמין לקבלת הודעות או לקלט משתמש (בצד הלקוח). כל אחד מהצדדים יכול לקבל או לשלוח קלט בכל רגע שהוא ללא תלות בהודעה מקדימה או שליחת הודעות בצורה סינכרונית. הלקוח יכול לקבל קלט משתמש או הודעה מהשרת בכל רגע שהוא והשרת יכול לקבל הודעה מכל לקוח בכל רגע שהוא והוא שולח הודעה אל לקוח אחר לאחר קבלת ההודעה.

הודעות העוברות בתקשורת

כל ההודעות יכללו בשדה האחרון את גודל ההודעה והמפריד בין השדות יהיה הסימן |.

- בקשת התחברות מלקוח לשרת- ההודעה תכלול מילת מפתח המסמנת תחילת התקשרות, שם משתמש וסיסמא איתה הלקוח רוצה להתחבר למערכת הצ'אט.

SIZE|HELLO|USERNAME|PASSWORD

- הודעת התחברות שגויה מהשרת ללקוח- ההודעה תכלול מילת מפתח המסמנת ששם המשתמש או הסיסמא שגויים, ובנוסף את סיום התקשורת של השרת עם הלקוח.

SIZE|INCORRECT

- רשימת משתמשים מחוברים משרת ללקוח- ההודעה אומרת שההתחברות למערכת הצליחה (שם המשתמש והסיסמא נמצאים ונכונים), וכוללת מילת מפתח המסמלת שליחת רשימת אנשי קשר, סוג משתמש(1 לרגיל, 2 למיוחד) ורשימה של כל המשתמשים המחוברים באותו הרגע ומאותו הסוג (רגיל או מיוחד). בין כל משתמש יהיה הסימן ~.

SIZE|CONTACTS|TYPE|USER1~USER2~USER3

- הודעה של צ'אט מלקוח אל שרת- כאשר שני לקוחות נמצאים בצ'אט אחד עם השני ואחד הלקוחות רוצה לשלוח הודעה למשתמש השני. ההודעה תכלול את מילת המפתח המסמלת שליחת הודעה חדשה, את שם המשתמש אליו נשלחת ההודעה ואת ההודעה עצמה (תמונה).

MSG|USER2|PICTURE_CODE|SIZE

- הודעה של צ'אט מהשרת ללקוח-כאשר הודעה מגיעה מלקוח אחד אל לקוח אחר השרת יעביר את ההודעה ללקוח בו פועל המשתמש הרצוי. ההודעה תהיה זהה להודעה שנשלחה לשרת מלבד שם המשתמש שיהיה שם המוען.

MSG|USER2|PICTURE_CODE|SIZE

שגיאות אפשריות בתקשורת

- השרת יכול להתנתק בשל בעיות תקשורת. במקרה זה כל הלקוחות לא יוכלו להמשיך לשלוח הודעות אך השרת או לקבל ממנו הודעות.
- הלקוח יכול לשלוח הודעה שאינה תואמת את מבנה ההודעה שנקבע מראש. במקרה זה השרת יתעלם מההודעה.
- השרת יכול לשלוח הודעה שאינה תואמת את מבנה ההודעה שנקבע מראש. במקרה זה הלקוח יתעלם מההודעה.
- הודעה יכולה להגיע לא בשלמותה לכל אחד מהצדדים. במקרה זה אותו הצד יתעלם לחלוטים מההודעה.

ממשקים

-WxPython

ממשק המאפשר לבנות ממשק גרפי נוח למשתמש במקום להשתמש בתצוגה הטקסטואלית הבסיסית. הממשק מאפשר למתכנת אפשרות הלצגת כפתורים, תיבות טקסט, זיהוי לחיצת עכבר ועוד.

תכנון קריאות:

- יצירת כפתור- button=wx.Button
- זיהוי לחיצה על כפתור- self.Bind(wx.EVT_BUTTON)
- כתיבת טקסט למסך- text=wx.StaticText
- סגירה של פריים- self.Destroy()
- יצירת פריים- wx.Dialog.__init__

התקנות

- **WxPython**

נכנסים לכתובת https://wxpython.org/download.php. שם ישנה רשימה של גרסאות של wxPython. יש לבחור בגרסת wxPython3.0-win64-py27 המתאימה לPython בגרסה 2.7 בה נכתב הפרויקט. יש לפתוח את קובץ ה-EXE שירד להריץ אותו וכאשר הוא מבקש לשמור יש להכניס C:\Heights\PortableApps\PortablePython2.7.6.1\App\Lib\site-packages.

מסמך בדיקות

במהלך העבודה על הפרויקט ישנו מספר שגיאות אפשריות.

- **ניסיון כניסה למערכת עם שם משתמש או סיסמא שגויים או חוסר התאמה בין שם משתמש וסיסמא-**
  המשתמש יכול להכניס קלט שאינו תואם את שם המשתמש או הסיסמא שנמצאים בצד השרת. במקרה זה הלקוח יקבל הודעה על כך מהשרת שלא נמצא משתמש התואם את הקלט שנשלח ויציג את ההודעה כקלט למשתמש.

- **שליחת הודעה שאינה עונה על המבנה שנקבע-**
  הלקוח או השרת עלולים לשלוח הודעה שאינה עונה על מבנה הפרוטוקול. במקרה כזה ההודעה לא תיענה על ידי הצד השני.

- **העלאת תמונה גדולה מדי שלא תצליח לעבור בתקשורת-**
  כדי למנוע מקרה כזה גודל התמונה ייבדק לפני שליחתה ובמקרה שהיא תעלה על הגודל המקסימלי שאפשרי לשליחה המשתמש יקבל הודעת שגיאה על כך. בכל מקרה התמונות נבחרות ממאגר תמונות שהוכן לפני ותואמות את הגודל האפשרי.

- **העלאת תמונה קטנה מדי שלא ניתן להעביר בה את המידע-**
  כדי למנוע מקרה כזה לפני הסתרת המסר בתמונה ייבדקו גודל התמונה וגודל המסר ובמקרה שאינם עומדים ביחסים המשתמש יקבל הודעת שגיאה על כך.

- **הכנסת PATHשלא קיים לתמונה -**
  במקרה בו התמונה לא תמצא במיקום שהוכנס המשתמש יקבל על כך הודעת שגיאה. הכנסת התמונות לא אמורה להיות על ידי המשתמש ולכן שגיאה זו לא אמורה לקרות.

מבני קובצי הפרוייקט

PROJECT.zip

הוראות שימוש- התקנה והפעלה

כדי להפעיל את הפרויקט צריך לשים את תיקיית השרת על מחשב אחד ואת תיקיית הלקוח על מחשב או מחשבים. ניתן לעשות זאת על ידי הכנסת DOK המכיל את תיקיות הפרויקט הנחוצות או על ידי העתקת הפרויקט למחשב עליו תרצה להריץ אותו.

תיקיית השרת כוללת:

- קובץ פייתון המכיל את הקוד של השרת אותו צריך להריץ- PicChat_server.py
- קובץ טקסט המכיל את שמות המשתמשים המורשים כמשתמשים והסיסמאות שלהם – spcl_users.txt

תיקיית הלקוח כוללת:

- קובץ פייתון עם המודול client.py
- קובץ פייתון עם המודול steg.py
- קובץ פייתון עם הגרפיקה ופעולות הלקוח אותו יש להריץ- PicChat.py
- תמונת רקע של מסך הכניסה- login_frame.png
- תמונת רקע של מסך אנשי הקשר- contacts_frame.png
- תמונת רקע של מסך הצ'אט- chat_frame.png
- תמונות בהן משתמשים בשביל מעבר המסרים

Server

כדי להריץ את השרת יש לפתוח CMD בתיקייה בה נמצא קובץ הפייתון עם השרת וקובץ הטקסט ולהוסיף פורט בו אנו רוצים להשתמש לדוג':

F:\cyber\project\2017 PicChat_server.py 5555

Client

כדי להריץ לקוח יש לפתוח CMD בתיקייה בה נמצאים כל הקבצים הקשורים ללקוח, להריץ את הקובץ PicChat.py ולהוסיף אליו את הIP בו נמצא השרת ופורט התואם לפורט שהוכנס בצד השרת לדוג':

F:\cyber\project\2017 PicChat.py 10.0.0.8 5555

כדי להכנס למערכת ניתן להשתמש בשם "Enter Username" ובסיסמא "Password" (הכתובים כברירת מחדל בכניסה)

משתמש נוסף בו ניתן להשתמש הוא "hadar" עם הסיסמא "12345".
כדי לבדוק את ההתקשרות אפשר לשלוח הודעות מEnter Username אל hadar.

משוב

תהליך העבודה על הפרויקט השנה היה עניין ומרתק אך מצד שני לחוץ ועמוס מבחינתי מכיוון שהשנה הייתי עמוסה בלימודים אחרים ובהתחייבויות מחוץ לבית הספר. כאשר עבדתי על הפרויקט יצא לי להתעניין בדברים רבים ולחקור את חלקם, אך בגלל חוסר הזמן נמנעתי מלקרוא נושאים נוספים שיכולתי להוסיף לפרויקט. לצערי לא הספקתי להוסיף את כל הפונקציות שרציתי להוסיף לפרויקט.

הפרויקט נתן לי לחקור לעומק מספר פורמטים של תמונות ולחשוב מחוץ לקופסא על דרכים בהן אפשר להסתיר מידע בתוך התמונות. נהניתי לחשוב על בדרך של הצד הפרוץ כדי להגן על המידע שאני רוצה להעביר ולהגיע לתוצאות המקסימליות של הגנה על המידע שברצוני להעביר.

הצעות להמשך

יש לי מספר רעיונות כיצד אפשר להפוך את המידי למוגן יותר וגם כיצד להוסיף פונקציות לנוחיות המשתמש.

- ניתן להצפין את המידע לפני הסתרתו בתוך התמונות. הצפנה תקשה עוד יותר על קריאת המסרים כך שגם אם האקר מודע למידע המוסתר בתמונה עדיין יהיה לו קושי בלפענח את משמעותו.
- ניתן להוסיף אופציה למשתמש בצד הלקוח לבחור את התמונה אליה הוא רוצה להכניס את ההודעה שהוא רוצה לשלוח. כך בהסנפת הפקטות ומעקב אחרי התמונות הגיוון יהיה גדול יותר ופחות מחשיד שיש משהו בתוכו.
- ניתן להוסיף כניסה למשתמשים תמימים בנוסף למשתמשים הנוכחיים כדי שגם משתמשי התוכנית לא ידעו בוודאות שקיימת האופציה למידע העובר בתוך תמונות.

ביבליוגרפיה

- https://he.wikipedia.org/wiki/PNG