

**עבודת גמר בתכנון ותכנות מערכות**

**התמחות הגנת סייבר**

**נושא העבודה:** כפתור מצוקה לחרשים

**שם**: \*\*\*\*\*\*\*\*\*

**ת.ז**: \*\*\*\*\*\*\*\*

**עיר ובית ספר:** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**שמות המנחים:** \*\*\*\*\*\*\*\*

**תאריך ההגשה:** 30.4.2018

**תוכן עניינים**

**מסמך עיצוב**

מבוא ..................................................................................................................4

תיאור המוצר....................................................................................................... 4

אילוצים ודרישות................................................................................................... 4

תיחום הפרויקט.................................................................................................... 5

מצב השוק כיום.................................................................................................... 5

**מסמך איפיון**

פונקציונליות המערכת........................................................................................... 6

אילוצים עיקריים................................................................................................... 6

הנחות יסוד ודרישות המערכת................................................................................ 6

תרשים ארכיטקטורת המערכת................................................................................ 7

פירוט והסבר על פונקציונליות המערכת................................................................. 6-7

**מסמך עיצוב**

הגדרות........................................................................................................... 8-9

נושאים שנדרשתי ללימוד עצמי וחקר........................................................................ 9

תהליך החקר....................................................................................................... 9

תיעוד נושא החקר.............................................................................................. 10

תרשים מודולים.................................................................................................. 11

פרוט המודולים והפעולות העיקריות.................................................................. 12-13

עיצוב נתונים ופרוטוקולים................................................................................ 13-14

ממשק משתמש והוראות הפעלה...................................................................... 14-18

הקודים העיקריים של המודולים והפעולות העיקריים............................................ 19-25

רפלקציה........................................................................................................... 25

ביבליוגרפיה................................................................................................. 26-27

**נספחים**

נספח א' – תיעוד נושא החקר.......................................................................... 27-28

נספח ב' פירוט תהליך הבדיקות שבוצעו על המערכת................................................ 28

נספח ג' – תיעוד גרסאות המסמכים....................................................................... 28

**מבוא**

כיום קיימות מערכות של כפתורי מצוקה בעיקר לבני הגיל השלישי שכוללות שיחת טלפון במקרה בו קיימת סכנה מסויימת. אך כאשר אדם הוא חרש הוא איננו יכול לקיים שיחה זו.

לכן עלה ברצוני לפתח תוכנה שיכולה לתקן או לעזור במצב זה ולגרום לכך שגם חרשים יוכלו להשתמש בכפתורים אלו ואולי אפילו להציל חיים.

התוכנה תבצע קישור בין מרכז העזרה והלקוח בצורה ויזואלית (צ'אט) וכך יוכל הלקוח לקיים שיחה עם המרכז לעזרה. ניתן גם להתחיל שיחת וידאו עם המרכז כך ניתן, במקרה של שריפה לדוגמה, להראות את בית למרכז כדי שיטפלו בבעיה בצורה הטובה ביותר.

קהל היעד של המערכת הוא בעיקר חברות המספקות את כפתורי המצוקה או חרשים שמשתמשים בכפתורים אלו.

**תיאור המוצר**

המטרה המרכזית של העבודה היא לכתוב תוכנה שתעזור לאנשים חרשים לתקשר עם עזרה במקרה של צרה כל שלא תהיה.

תרחיש אחד שהמערכת עונה עליו הוא תרחיש בו יש שריפה בביתו של המשתמש, הוא מתקשר בעזרת הצ'אט עם חברת ההצלה ואפילו בוחר להראות למוקד את המצב בבית בעזרת שיחת הוידאו.

תרחיש נוסף שהמערכת עונה עליו הוא מקרה בו המשתמש חווה התקף רפואי מסוג כזה או אחר, הוא לוחץ על כפתור המצוקה ונשלחת אליו עזרה מתאימה לבעייתו.

**אילוצים ודרישות**

דוגמה לאירוע שהמערכת עונה עליו היא שריפה בבית של המשתמש רצון שלו לקרוא לעזרה. המערכת תקשר אותו בעזרת שיחוח (צ'אט) ובשיחת וידאו לבחירתו.

יש צורך שהמערכת תגיב בצורה מהירה כדי לספק את העזרה המהירה ביותר למשתמש. בנוסף, יש צורך שהמערכת תהיה מובנת וקלה לשימוש כדי לייעל ביקוש עזרה והגשתה.

**תיחום הפרויקט:**

תקשורת- רשתות, הצפנה. תקשורת בין המוקד לבין המשתמש בצורה טקסטואלית וויזואלית. הצפנה של קבצים רגישים כמו טפסים רפואיים, כמו כן, הצפנה של סיסמאות משתמשים.

קיימים מספר מודולים ותתינושאים במערכת. ראשית, ממשק לקוח הכולל הרשמה למערכת והתחברות אליה. שנית, שליחת הודעות בין המוקד ללקוח. בנוסף, שמירה של נתונים כמו קבצים רפואיים וסיסמאות בצורה מוצפנת. כמו כן, המערכת תומכת בשיחת וידאו בין הלקוח למוקד.

**מצב השוק כיום:**

כיום קיימות חברות ומוקדים שמציעים כפתורי מצוקה אך הן לא נוגעות לחרשים משום שאין הם יכולים לדבר בטלפון כשהמוקד מתקשר בלחיצה של כפתור המצוקה על ידי הלקוח. הפרויקט שלי שונה מזה בכך שהוא נותן מענה במיוחד לחרשים.

**מסמך אפיון**

**פונקציונליות המערכת:**

* הרשמה למאגר משתמשים.
* לחיצה על כפתור מצוקה.
* חיבור משתמש למוקד עזרה (צ'אט).
* ארגון מידע בdatabase.
* ממשק לקוח.
* שיחת וידאו.
* שליחת קריאה לעזרה ספציפית.
* אחסון ושליחה של קבצים רגישים.

**אילוצים עיקריים:**

* התוכנה רצה על מערכת הפעלה Windows בגרסאות 7,10
* התוכנה דורשת חיבור לאינטרנט
* התוכנה פועל בין מחשבים הפועלים באותה רשת.
* התוכנה תהיה מוכנה לשימוש עד קיץ 2018.

**הנחות יסוד ודרישות המערכת:**

המערכת דורשת חיבור למצלמה, חיבור לאינטרנט.

**פרוט והסבר על פונקציונליות המערכת:**

**הרשמה למאגר משתמשים:** כל לקוח בשימוש ראשון יצטרך להירשם, כלומר ליצור משתמש בעל שם משתמש וסיסמה, יוכל גם להזין פרטים רפואיים שישמרו בצורה מאובטחת בשרת.

**התחברות למערכת:** בתחילת שימוש, על הלקוח להתחבר למשתמש הפרטי שלו בעזרת שם המשתמש והסיסמה שהוכנסו בזמן יצירת המשתמש.

**לחיצה על כפתור מצוקה:** משתמשים יוכלו ללחוץ על כפתור וירטואלי כדי לקרוא לעזרה כללית. בנוסף יוכלו ללחוץ על כפתורים ספציפיים יותר כמו כפתור שמסמן שיש שריפה או בעיה רפואית ועוד...

**חיבור משתמש למוקד עזרה בעזרת צ'אט:** משתמשים יוכלו להתחבר לנציג שירות בצורת צ'אט, להתכתב עם הנציג ולהזעיק עזרה בצורה כזו.

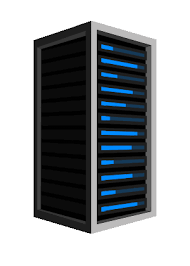
**אחסון מידע בDatabase :** מידע של משתמשים כמו סיסמאות ושמות משתמש יאוכסנו בDatabase בסרבר. בנוסף, מידע רגיש כמו תיקים רפואיים יאוכסן בצורה בטוחה.

**ממשק לקוח:** ממשק הלקוח כולל כפתור מצוקה כללי וכפתורי מצוקה ספציפיים יותר. בנוסף, הוא כולל גם אפשרות לדיבור בצ'אט עם נציג.כמו גם אפשרות לשיחת וידאו**.**

**שיחת וידאו:** ניתן גם למשתמש ליצור שיחת וידאו עם המוקד.

**תרשים ארכיטקטורת המערכת:**

****איור 1 מציג את ארכיטקטורת המערכת:

****

****

****

לקוח

סרבר

מוקד

**מסמך עיצוב**

**סביבת הפיתוח**

הקוד בפרויקט זה כתוב בשפת Python בגרסת 2.7. ונכתב בסביבת העבודה PyCharm . הקוד מתאים ויעבוד על מערכת הפעלה של Windows בגרסאות 7,10

הכלים והספריות שבהם השתמשתי הם:

1. Socket – מודול שמשמש לבניית תקשורת בסיסית בין מחשבים שונים בתוך הרשת.
2. - Select ספרייה שמאפשרת ביצוע שרת עם כמה משתמשים שונים.
3. Treading – ספרייה שמאפשרת יצירת Thread.
4. Cv2 – ספרייה שמכילה כלים לעיבוד ויצירה של תמונות.
5. Pickle, cPickle – ספרייה שמשמשת לניהול קבצי pickle שמהם בנוי מבנה הנתונים בפרויקט זה.
6. Numpy- עיבוד מערך עבור מספרים, מחרוזות, רשומות ואובייקטים בצורה יעילה.
7. Os- ספרייה שמספקת דרך נוחה לעבודה עם קבצים.
8. Time- ספרייה בה כלים לעבודה עם זמנים, כמו יצירת השהייה.
9. Tkinter- ספרייה המשמשת לבניית ממשק המשתמש.
10. Hashlib- ספרייה בה כלים ליצירת Hash.
11. tkFileDialog- ספרייה המשמשת שימוש ויזואלי בקבצים.
12. Struct- ספרייה ש"אורזת" קבצים למבנה בינארי.

**הגדרות**

* שרת (Server)- שרת הוא מחשב המריץ תוכנה שמטרתה לתת שירותים למחשבים אחרים, לקוחות (Clients) .
* לקוח (Client) – הלקוח מופעל על ידי המשתמש ופונה לשרת לקבלת שירותים כאלו ואחרים.
* קובץ Pickle – קובץ המכיל מבנה נותנים מסוים כמו רשימה או מילון, ניתן לגשת אליו בקלות.
* פיקסל (Pixel) – יחידת מידע גרפית בסיסית במחשב המתארת נקודה בתמונות דיגטליות.
* תמונה (Frame ) – כל תמונה בנויה ממאות אלפי על מיליוני פיקסלים. מספר תמונות ברצף ייצרו סרט וידאו.
* וידאו (Video) וידאו מורכב מרצף של תמונות המתחלפות במהירות גבוהה וכך יוצרות תחושה של תנועה.
* בסיס נתונים (Database) – אמצעי המשמש לאכסון מסודר של נתונים.
* Thread (תהליכון) – במערכות הפעלה מודרנית קיימת אפשרות להפעיל תהליכונים מתוך תהליך ראשי בכדי להריץ כמה דברים בו זמנית.
* Hash (גיבוב) – פונקציה שממירה קלט חופשי באורך משתנה לפלט באורך קבוע. קשה מאוד לדעת מה היה הקלט של הפונקציה לפי הפלט ולכן יש לה שימושים רבים באחסון מידע בטוחה.
* חדר(Room) – בפרויקט נעשה שימוש בחדרים שונים של צ'אט ווידאו צאט, חדר בפרויקט זה מוגדר כשיחה בין שני אנשים, בין לקוח לבין עובד מוקד.

**הנושאים שנדרשתי ללימוד עצמי וחקר**

בביצוע הפרויקט נדרשתי ללמוד נושאים שונים:

* שימוש בספריות לא מוכרות כמו cv2. בספריית cv2 יש שימוש בפרויקט בצילום של התמונה מהמצלמה.
* ניהול כמה חדרים ולקוחות בשרת העברה של הודעות רק למשתמשים המתאימים.
* עבודה עם קבצי pickle, כתיבה וקריאה מתוכם.
* יצירת ממשק משתמש.
* צילום וידאו ושליחתו בזמן אמת – שיחת וידאו.
* שימוש בגיבוב (Hash).
* שליחת קבצים ושמירתם בצורה מסודרת.

**תהליך החקר**

**שיחת וידאו**- תחילה תכננתי לבצע את השיחה בעזרת פרוטוקול רשת UDP אך כאשר ניסיתי להריץ את התוכנית כל הפיקסלים בתמונה היו מבולגנים משום שהפרוטוקול לא בודק שהפאקטות מגיעות בסדר הנכון. לאחר מכן, הפתרון שלי היה לשנות את הפרוטוקול ל TCP ובכך לסדר את החבילות כשהן מגיעות והווידאו נראה נכון ומסודר. אך הבעיה בפרוטוקול זה היא שהוא יחסית איטי ולוקח הרבה זמן למחשב לבדוק שהחבילות מסודרות כמו גם עוד בדיקות כאלו ואחרות ולכן הדיליי בין השליחה להצגה של הווידאו היה גדול מאוד. אחרי כמה שינויים בקוד והפיכתו ליעיל ומהיר יותר הדיליי קטן יותר ויותר.

**צ'אט שמחולק לחדרים** – בהתחלת העבודה על הפרויקט הצלחתי להכין צ'אט בעל חדר אחד בו כל דבר שנכתב יהיה פתוח לצפייה של שאר האנשים שמחוברים לצ'אט. אחרי זה רציתי שהצ'אט יהיה צ'אט של אחד על אחד, לכן חשבתי על הרעיון של ליצור רשימה עם חדרים, בעיה נוספת שנתקלתי בה הייתה לוודא שהחדרים מורכבים ממשתמש רגיל ומשתמש שהוא עובד שמטרתו לעזור למשתמש הרגיל לכן הייתי צריך לשמור רשימה של כל העובדים.

**קבצי pickle** – בתחילת הפרויקט ידעתי שאני רוצה לשמור משתנים ונתונים שונים ב database מסוים. אחרי חיפושים באינטרנט מצאתי ששמירה של מבני נתונים כאלו ואחרים בקבצי pickle היא דרך קלה ונוחה מאוד לבצע שמירה של נתונים. העדפתי את האפשרות הזו מאשר לבנות database מסובך יותר ב sql לדוגמה משום שהצורך שלי היה במשהו פשוט ונוח, וקבצי ה pickle נתנו לי את זה. הם קלים מאוד לגישה וקלים ליצירה.

**תיעוד נושא החקר**

נושא החקר שביצעתי הוא תקשורת ושליחה של מידע שונה.

* בהתחלה למדתי איך לכתוב שרת שתומך בכמה משתמשים בעזרת select. הבעיה שהייתה בשרת זה היא שכל הודעה שנשלחה אליו מאחד הלקוחות נשלחה לכל מי שמחובר אליו.
* אחרי זה למדתי איך ליצור חדרים שבהם נמצאים שני משתתפים, לקוח ועובד המרכז לפניות, ולשלוח מידע רק למשתמשים שנמצאים בתוך חדר מסוים בצורה ממוקדת.
* בהמשך, אחרי חיפוש ברשת ביצעתי שליחה של קובץ שנבחר על ידי המשתמש ושמירתו בשרת בתיקייה מסודרת לפי שם המשתמש של השולח.
* לאחר מכן, למדתי כיצד להשתמש במצלמה כדי לקחת תמונות באמצעות ספריית cv2 ,התחלתי לחקור שליחה של תמונות במהירות בכדי שיראו ביחד כמו סרט, כלומר שליחת סרטון בזמן אמת. לשם כך השתמשתי בספריית struct ואפילו pickle זאת בכדי לשלוח את הקבצים במהירות גבוהה יותר ולפענחם מקובץ numpy בצורה טובה.

**תרשים מודולים**

**Worker client**

Login()

Pgot()

Video\_recv()

Mamain()

Operate\_chat()

ChatPage

Trial()

LogIn

Whenlogin()

App

\_\_init\_\_

Show\_frame()

Destroy\_frame()

**Client**

Videochat()

Chatcon()

Operate\_chat()

On\_closing()

App

\_\_init\_\_()

Show\_frame()

Destroy\_frame()

Client

\_\_init\_\_()

Get\_socket()

Close\_socket()

Login()

\_\_init\_\_()

Whenlogin()

MainMenu

\_\_init\_\_()

ChatPage()

\_\_init\_\_()

Trial()

RegisterPage

\_\_init\_\_()

Opensurf()

Whenreg()

**Server**

Sendme()

Recving()

Video\_chat()

Register()

Savefiles()

Logout()

Is\_user\_valid()

Make\_Pickle

PickFile

\_\_init\_\_

Create\_file()

Get\_value()

Update\_file()

Open\_file()

**פירוט המודולים והפעולות העיקריות**

שרת (Project server)

אחראי על קישור בין שני סוגי המשתמשים, עובדים ומשתמשים רגילים. כמו גם אחראי על שמירת מידעעל משתמשים.

Recving() – פעולה שאחראית על קבלת מידע מהמשתמשים השונים וניתוב לפעולות שונות לפי הצורך של המשתמש.

Register() – פעולה שאחראית על מתן מענה בקבלה של בקשה ליצירת משתמש חדש, קבלה של קובץ ושמירה של סיסמאות במקרה בו הפונקציה בודקת שהמשתמש לא קיים ב database.

Savefiles() – פעולה שמטרתה לשמור את הקובץ שהתקבל בזמן יצירת משתמש חדש.

Logout() – פעולה שמטפלת בבקשה של המשתמש להתנתק מהשרת.

Is\_user\_valid() – פעולה שבודקת אם הפרטים שהוכנסו בזמן התחברות ומאשרת התחברות של משתמש.

Videochat() – פעולה שמעבירה את השידור של הוידאו מהמשתמש הרגיל אל המוקד.

Send\_me() – פעולה ששולחת לפי חדרים את ההודעות שמחכות לשליחה מלקוח אחד לשני.

לקוח (project)

App – מחלקת בסיס של חלון ui בפרויקט בו פעולות כמו showframe() שמפעילה מחלקה אחרת מתוך רשימה מוגדרת.

Client – מחלקה שמגדירה את החיבור של המודול לשרת. היא יוצרת חיבור אליו ומציעה פעולות לניהול החיבור.

LogIn – מחלקה שמגדירה את החלון בו המשתמש מתחבר אל המערכת, כולל קבלת שם משתמש וסיסמה. בתוך המחלקה פעולה whenlogin() שמופעלת אחרי שהמשתמש מנסה להתחבר ומטרתה לשאול את השרת אם שם המשתמש והסיסמה נכונים.

MainMenu – מחלקה שמגדירה את החלון המרכזי בו בוחרים אם לעבור לצ'אט כתוב או וידאו

ChatPage – מחלקה שמגדירה את החלון בו ניתן להתכתב בכתב. במחלקה מצויה הפעולה trial() שמטרתה לשלוח את המידע הרצוי לשליחה ועדכון החלון בו מוצג היסטוריית ההתכתבות.

Videochat – פעולה שמצלמת בעזרת המצלמה ושולחת את הוידאו בזמן אמת אל נציג ממרכז התמיכה של התוכנה.

RegisterPage – מחלקה שמגדירה את החלון בו המשתמש נרשם למערכת בהכנסת שם משתמש, סיסמה וקובץ רפואי. בתוך המחלקה יש את הפעולה whenreg() שמופעלת כאשר המשתמש מנסה להירשם, שמטרתה לבדוק שהשם משתמש וסיסמה שהקיש המשתמש פנויים ולשלוח את הקובץ הרפואי שלו לשמירה בשרת.

לקוח – עובד(worker client - project)

Login() – פעולת התחברות לשרת.

Mamain() – פעולה ששולחת את פרתי ההתחברות של המשתמש אל השרת.

Pgot() – פעולה שמופעלת ב thread שקולטת מידע מהשרת.

Video\_recv() – פעולה שמקבלת את הוידאו בזמן אמת מן המשתמש, מעבדת אותו ומציגה אותו.

Make\_Pickle

Pickfile – מחלקה לניהול קבצי ה pickle בה פעולה שיוצרת קובץ – Create\_file() פעולה שמעדכנת את התוכן של הקובץ – update\_file() ופעולה שמחזירה את הערך בתוך הקובץ – get\_value().

עיצוב נתונים ופרוטוקולים

הפרוטוקולים:

בתוכנית הלקוחות מתחברים לשרת בעזרת socket אחד שדרכו נשלח כל המידע כל שליחת המידע ב socket מתבצע בפרוטוקול TCP.

**פרוטוקול התחברות למערכת** – כאשר המשתמש מבקש להתחבר למערכת, התוכנית תשלח לשרת string שמבנהו הוא: '@'שם משתמש'#'סיסמה. כאשר הסיסמה היא hash256 של הסיסמה.

|  |  |  |  |
| --- | --- | --- | --- |
| Password | # | username | @ |

**פרוטוקול הרשמה למערכת** – כאשר המשתמש מבקש להירשם למערכת, התוכנית תשלח אל השרת string שמבנהו הוא: '`'שם משתמש'#'סיסמה. כאשר הסיסמה היא hash256 של הסיסמה המקורית. אחר כך שולח את הקובץ הנבחר על ידי המשתמש בפרוטוקול שמתנהל בכך שנשלח אורך של הstring שישלח ואחר כך נשלח המידע עצמו במקבצים של 1024.

|  |  |  |  |
| --- | --- | --- | --- |
| Password | # | Username | **`** |

**פרוטוקול ניתוק מהמערכת** – כאשר המשתמש מבקש להתנתק מהמערכת נשלחת הודעה לשרת עם None.

**פרוטוקול חיבור לצ'אט** – כאשר התוכנה מנסה להתחבר אל הצ'אט הכתוב נשלחת הודעה אל השרת שמכילה את התו '^' וכך השרת יודע שהוא צריך ליצור חדר לשיחה.

מבני הנתונים:

ה database שהשתמשתי בו בפרויקט מורכב מקבצי pickle.

קובץ אחד מכיל מילון בו שמות משתמשים וסיסמאות של עובדים וקובץ שני מכיל מילון ובו שמות משתמש וסיסמאות של משתמשים רשומים.

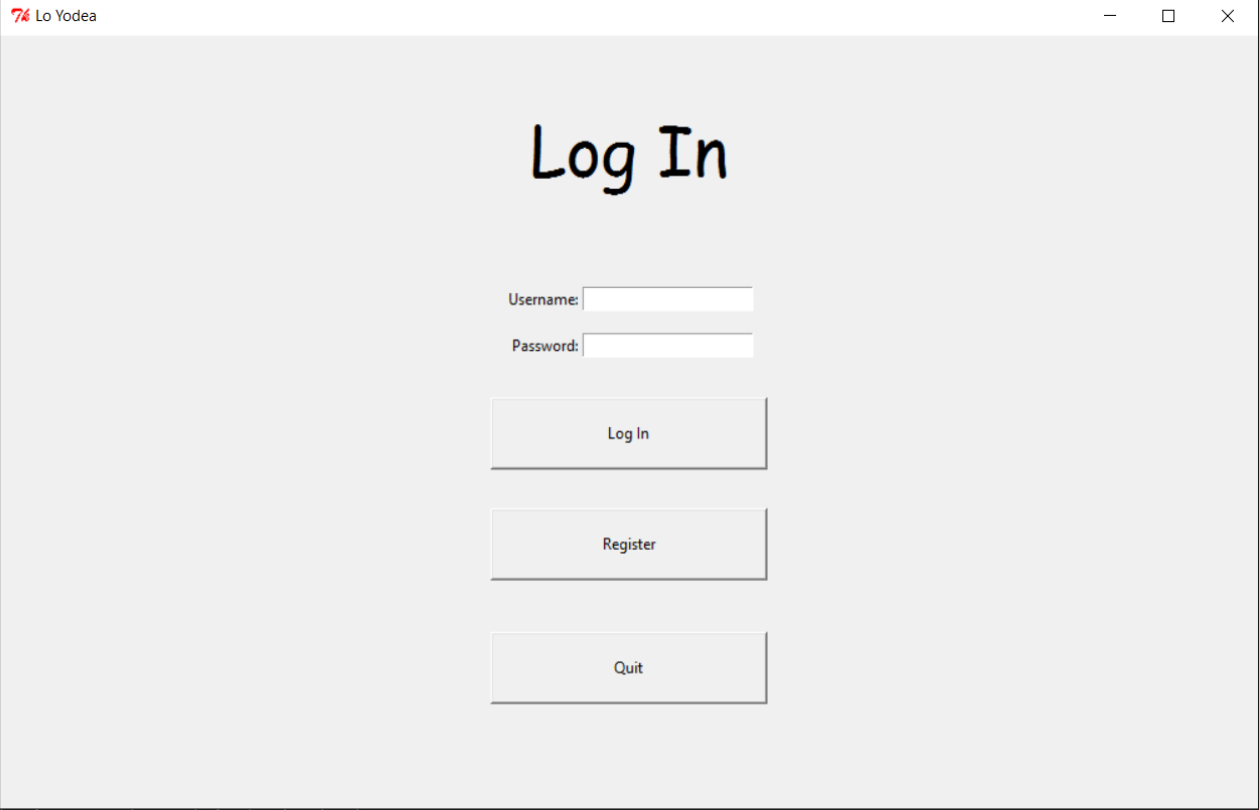
בתוך קובץ ה pickle ששמו הוא ourip נמצא מילון המורכב בצורה הזו:

{user\_username : user\_password}

בתוך קובץ ה pickle ששמו הוא users מאוחסן מילון המורכב בצורה הבאה:

{user\_username : user\_password}

ממשק משתמש והוראות הפעלה

המסך הראשון שנפתח כאשר מפעילים את התוכנה (עם הלקוח הרגיל) הוא המסך של ההתחברות למערכת. 

הוראות הפעלה למסך התחברות למערכת:

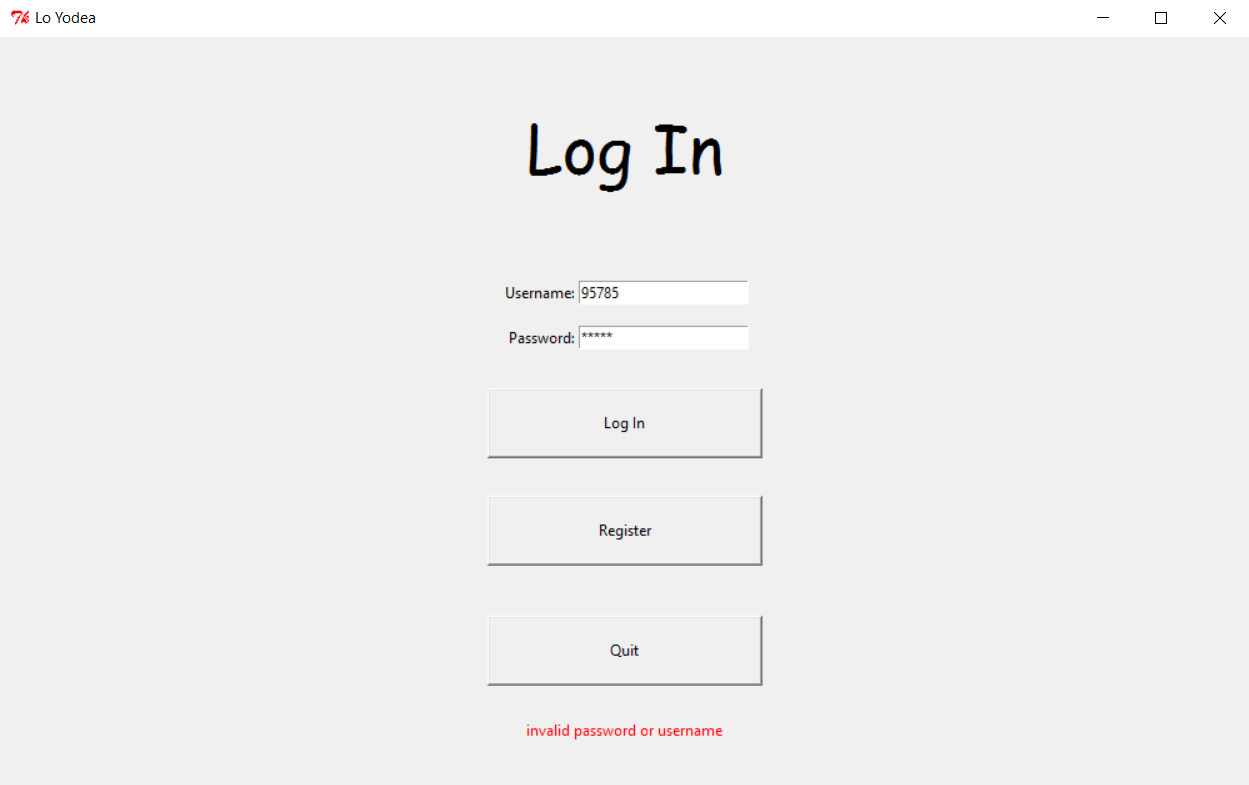
בכדי להתחבר אל המערכת יש להקליד את שם המשתמש והסיסמה שהוגדרו בזמן ההרשמה וללחוץ על כפתור ה log in. בכדי לצאת מן התוכנית יש ללחוץ על כפתור ה Quit.

בכדי לעבור למסך ההרשמה למערכת יש ללחוץ על כפתור Register .

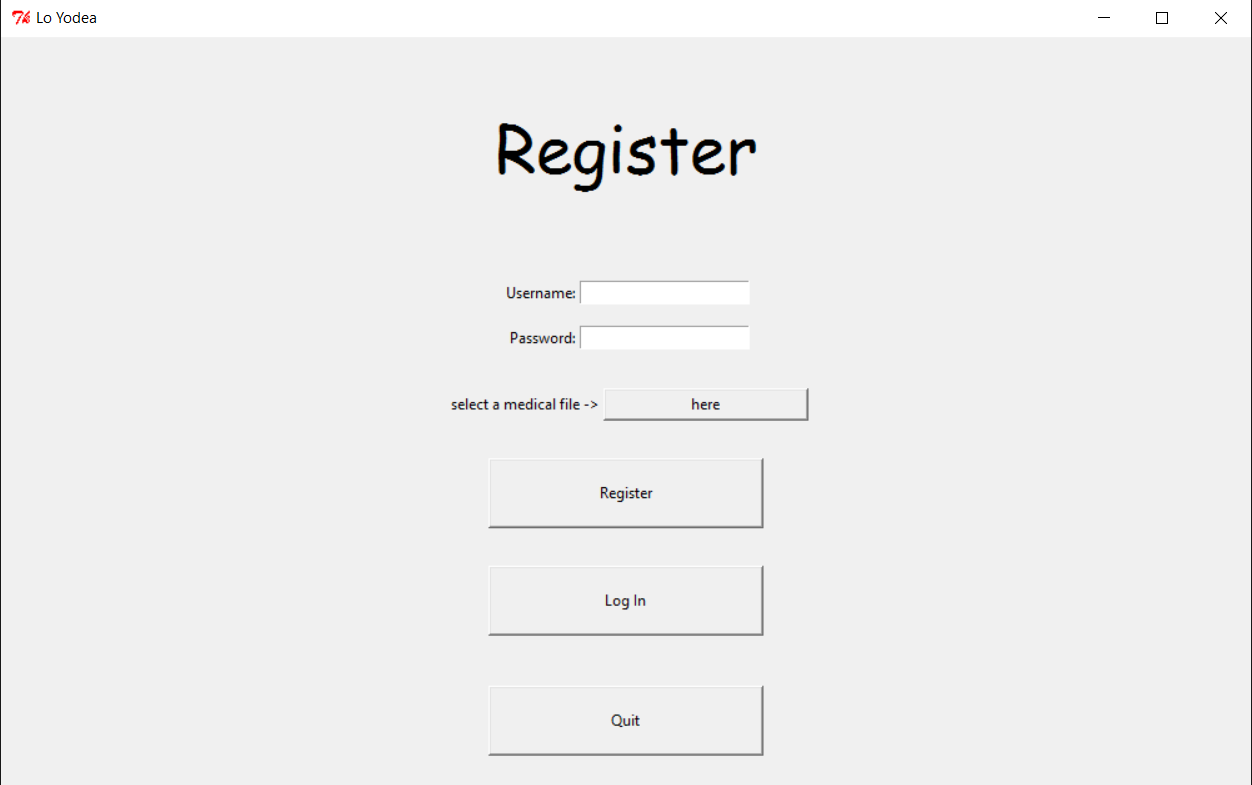
הודעות שגיאה והערות למשתמש:

אם שם המשתמש והסיסמה שהוכנסו לא מתאימים מסיבות שונות (כמו ששני השדות ריקים), לא יהיה ניתן להתחבר אל המערכת.

אם שם המשתמש או הסיסמה שהוקלדו אינם נכונים תופיע הודעה אדומה שאומרת שאחד מהם לא נכון:



בעת לחיצה על כפתור ה Register יופי המסך השני :



הוראות הפעלה למסך הרשמה למערכת:

בכדי להירשם למערכת יש להקליד את השם משתמש וסיסמה הרצויים במקומות הנכונים וללחוץ על כפתור הוספת קובץ רפואי, לבחור קובץ וללחוץ על כפתור Register.

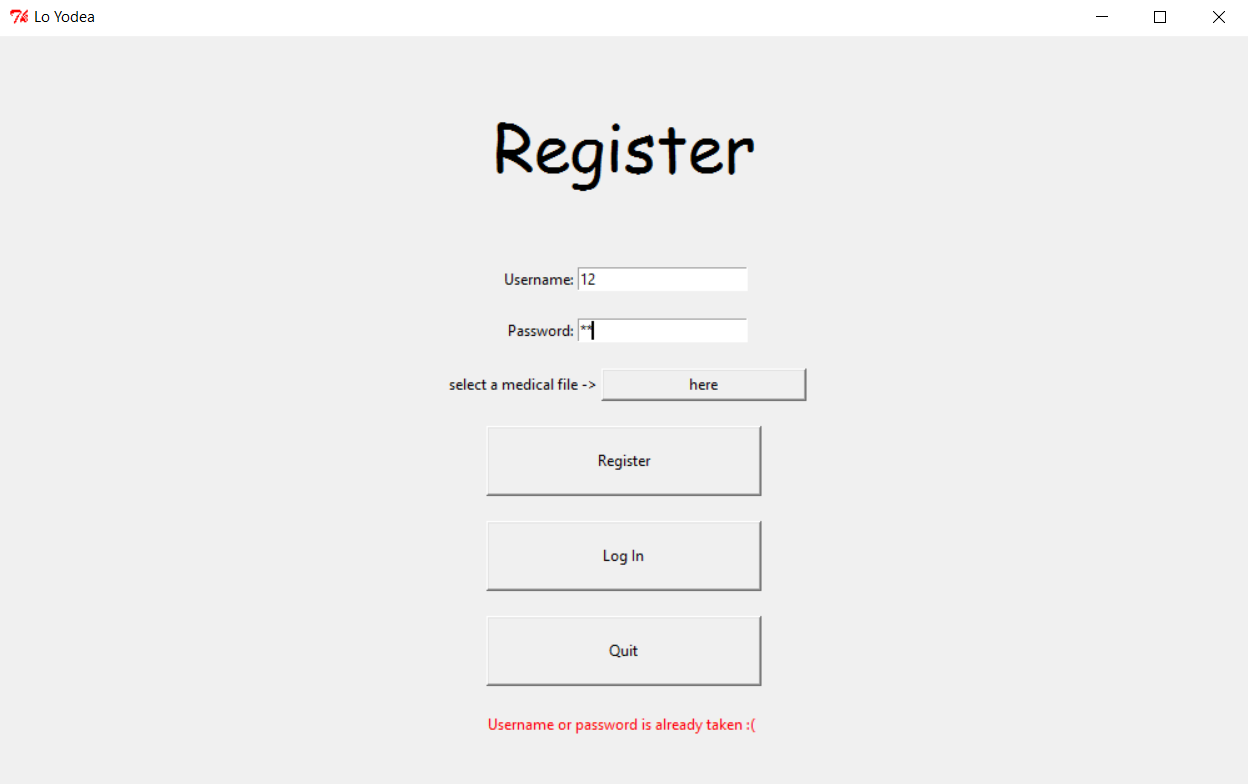
אם ברצון המשתמש לצאת מהתוכנית הוא צריך ללחוץ על כפתור ה Quit.

כדי לעבור למסך ההתחברות למערכת יש ללחוץ על כפתור ה Log In.

הודעות שגיאה והערות למשתמש:

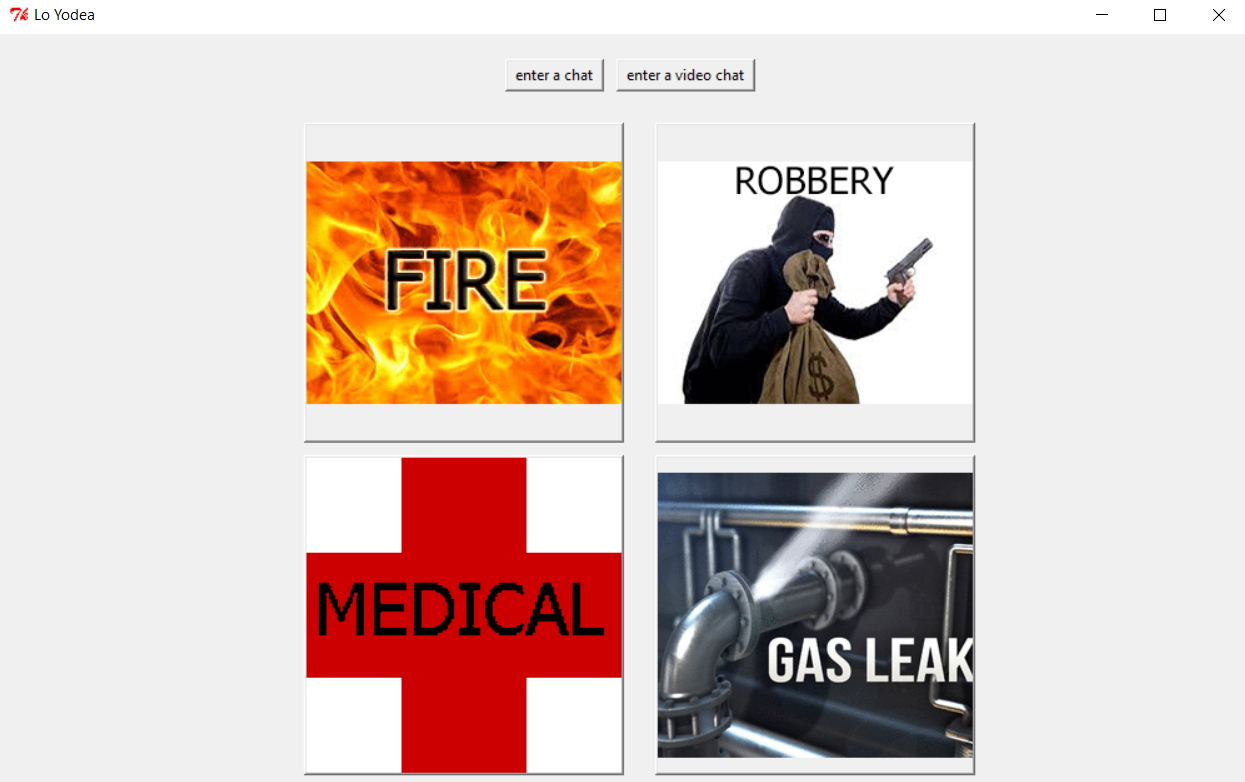
אם הנתונים שהוכנסו לשני השדות לא טובים לדוגמה אם השדות ריקים ההרשמה לא תתבצע.

אם שם המשתמש תפוס תופיע הודעה שאומרת זאת אחרי הלחיצה על כפתור ה Register.



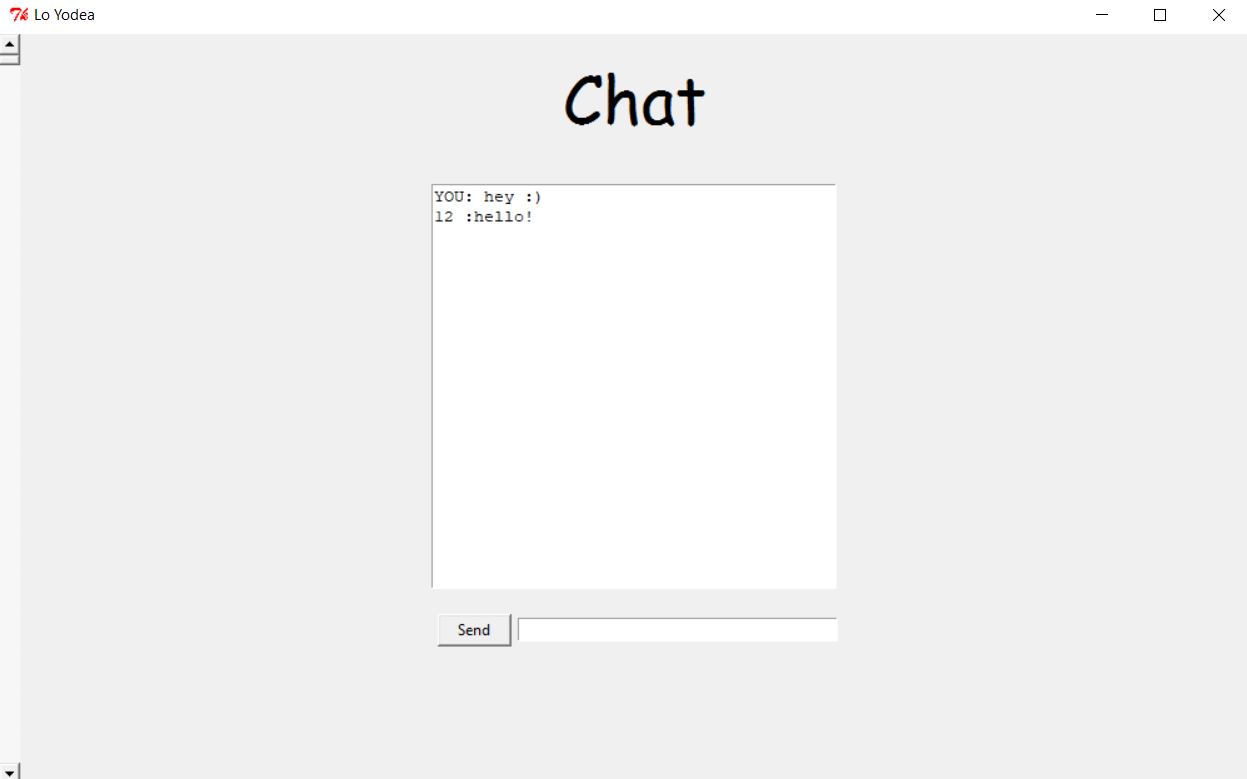
הוראות הפעלה למסך ראשי:

אחרי התחברות למערכת המסך שאליו מגיעים הוא המסך הראשי בו ניתן לבחור בין כמה מקרי חירום מהירים וניתן לבחור לדבר בצ'אט עם נציג. כמו כן, גם לבחור לדבר בוידאו צ'אט חד צדדי עם נציג. בלחיצה על אחד מהכפתורים המהירים (הכפתורים בעלי התמונות) יהיה מעבר אל שיחת צ'אט כתוב.



הוראות הפעלה למסך צ'אט:

בכדי לכתוב הודעה לנציג יש להקליד את ההודעה הרצויה במקום המתאים וללחוץ על כפתור ה Send.

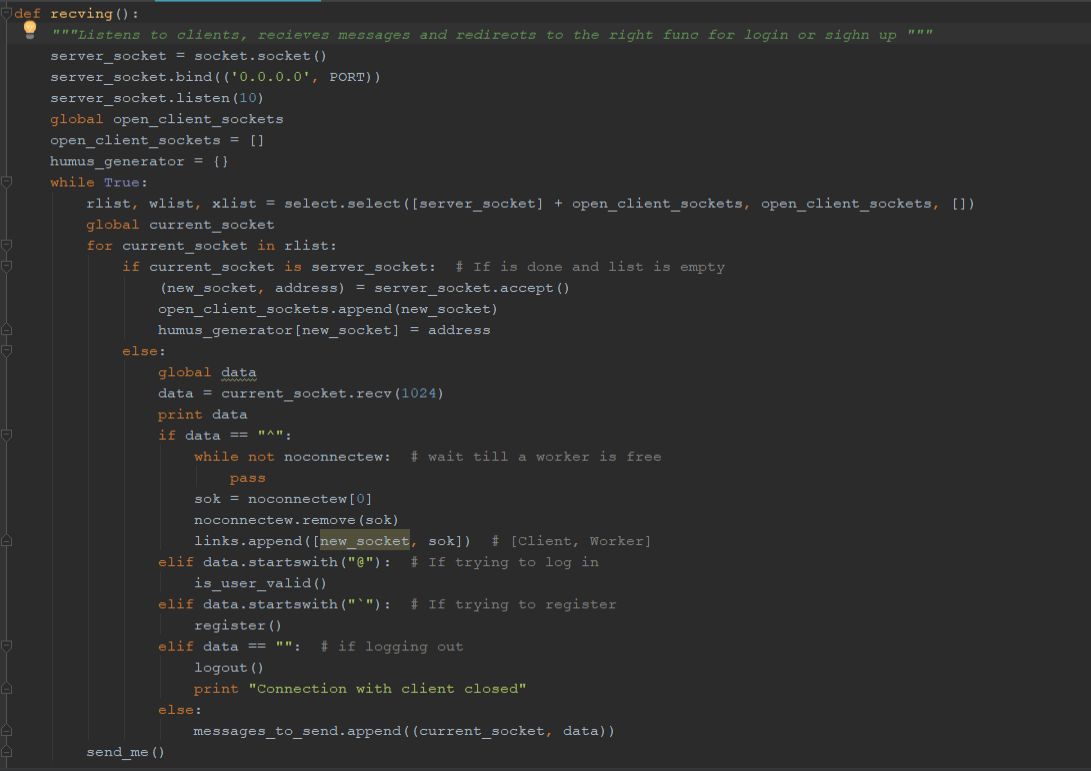
לאחר שהמסך מתמלא בהודעות ניתן לגלול בכדי לראות את ההודעות הקודמות יותר בעזרת מד הגלילה בצד שמאל.****

הוראות הפעלה למסך שיחת וידאו:

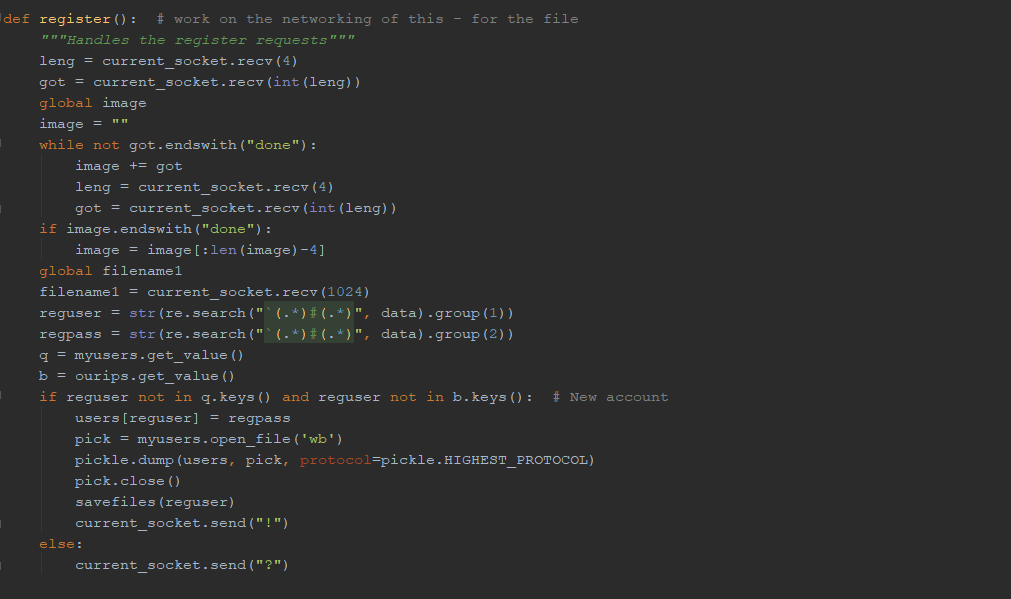
בכדי לצאת מהמסך יש ללחוץ על המקש "Q" במקלדת.

**הקודים העיקריים של המודלים והפונקציות העיקריות**

בפונקציית recving() מתוך מודול project server הסרבר מקבל הודעות מהלקוחות וממיין אותם לפי מטרתם. אם להרשמה למערכת, התחברות אליה ועוד...

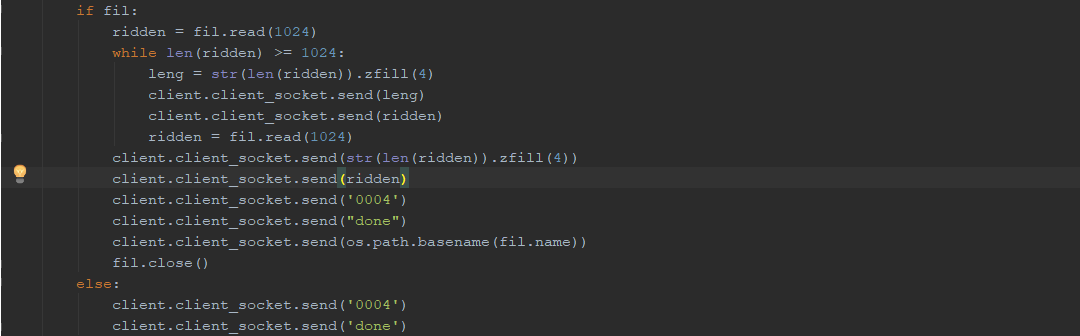


הרשמה למערכת:

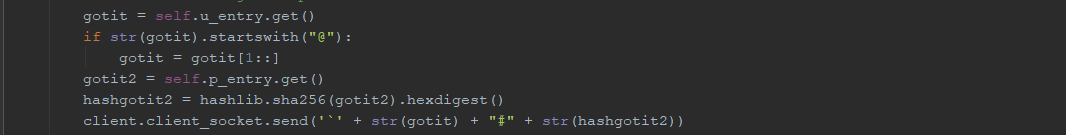
בסרבר, הפונקציה register() בודקת אם שם המשתמש והסיסמה לא תפוסים ושומרת את הקובץ שהמשתמש הכניס בזמן ההרשמה:

בצד הלקוח, התוכנית שולחת לשרת שם משתמש וגיבוב (Hash256) של הסיסמה שהמשתמש הכניס כמו גם את הקובץ שהכניס, אם הכניס. הוא גם מבקש לקבל תשובה מהשרת שהתהליך של ההרשמה עבר חלק גם בשרת.

קוד של שליחת הקובץ:

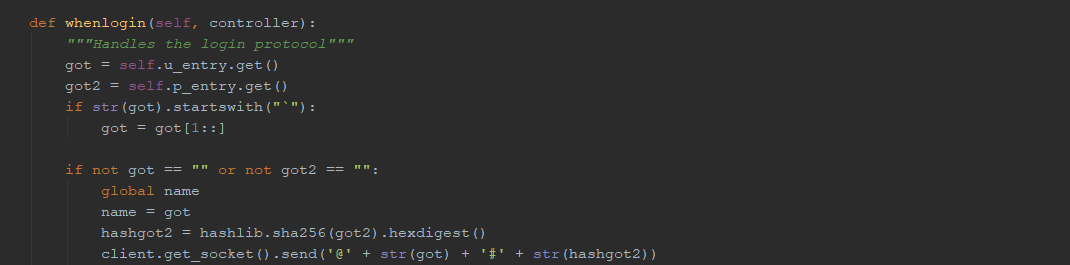


קוד של שליחת שם המשתמש והסיסמה:

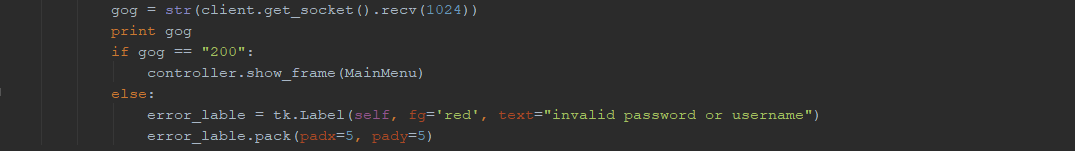


התחברות למערכת:

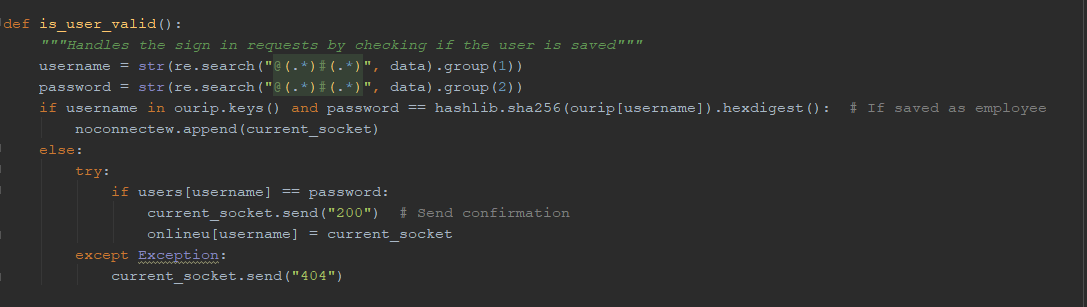
הלקוח, בלחיצה על כפתור ההתחברות, שולח את שם המשתמש שהוקלד ואת הסיסמה אחרי גיבוב (hash256):



ומחכה לאישור מהשרת:

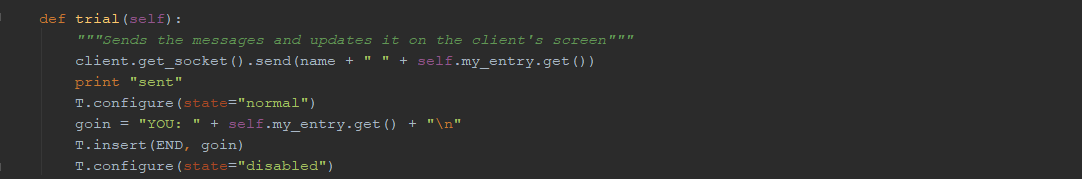


השרת בודק אם שם המשתמש והסיסמה נכונים, אחר כך, שולח הודעה אל הלקוח עם התשובה.

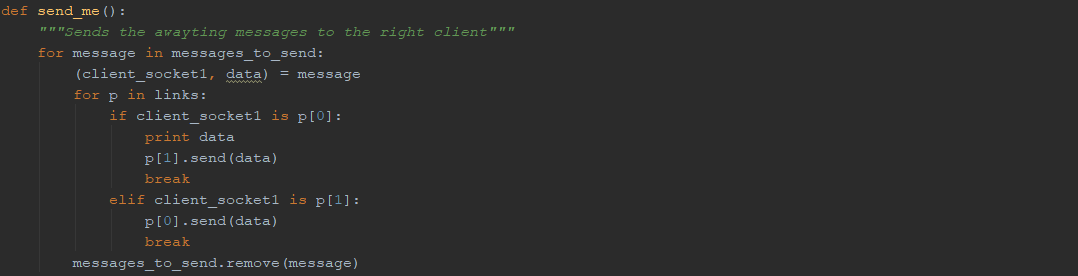


צ'אט כתוב

במתן פקודה לשליחה של מידע, לחיצה על כפתור השליחה, חלון ההצגה של המידע שנשלח משני הצדדים נפתח לכתיבה, נכתב בו מה שהמשתמש רוצה לשלוח, ההודעה נשלחת והחלון נסגר לכתיבה:

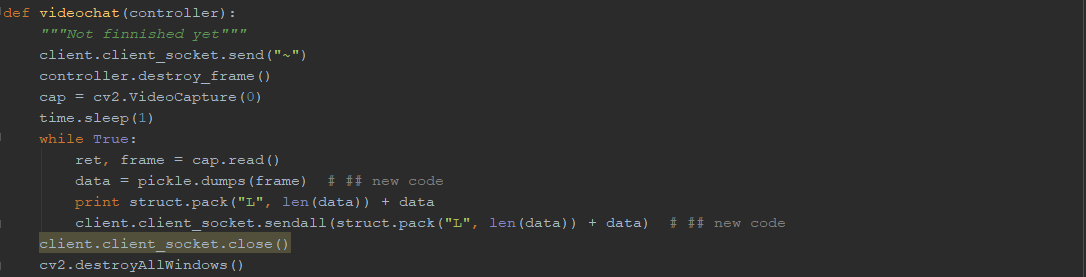


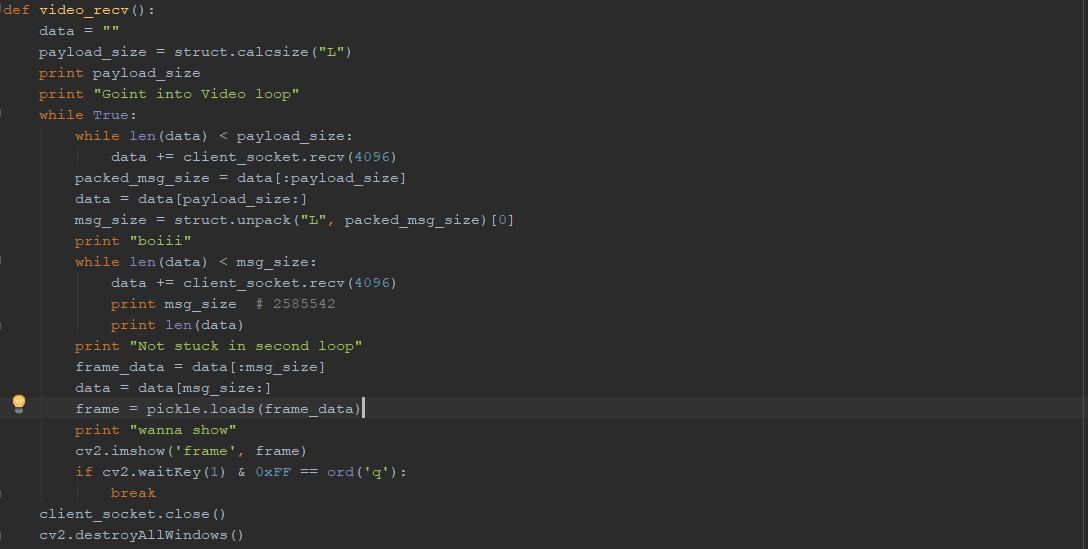
אם השרת מקבל מידע שאינו מוגדר בפרוטוקולים של התוכנה ומקבל אותו ממשתמש שברגע זה בשיחה, הוא מעביר את ההודעה לאדם השני בחדר:



**שיחת וידאו:**

קוד של לכידת תמונה מן המצלמה ושליחתו:



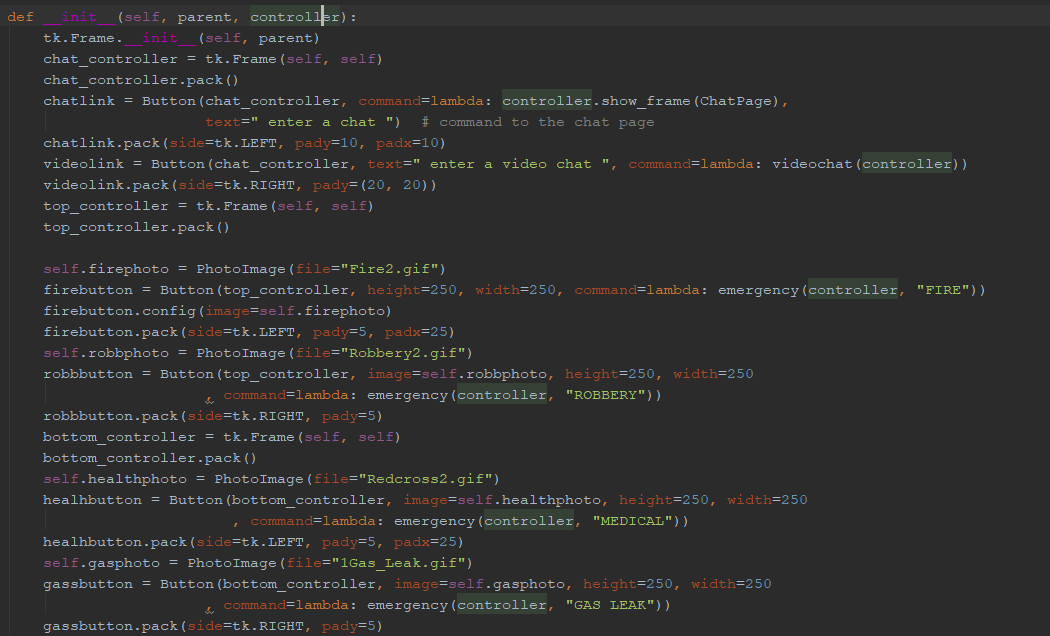
קוד של קבלה של frames , עיבודם והצגתם:

**ממשק משתמש**

מחלקה שמגדירה את כל ממשק המשתמש בתוכנית



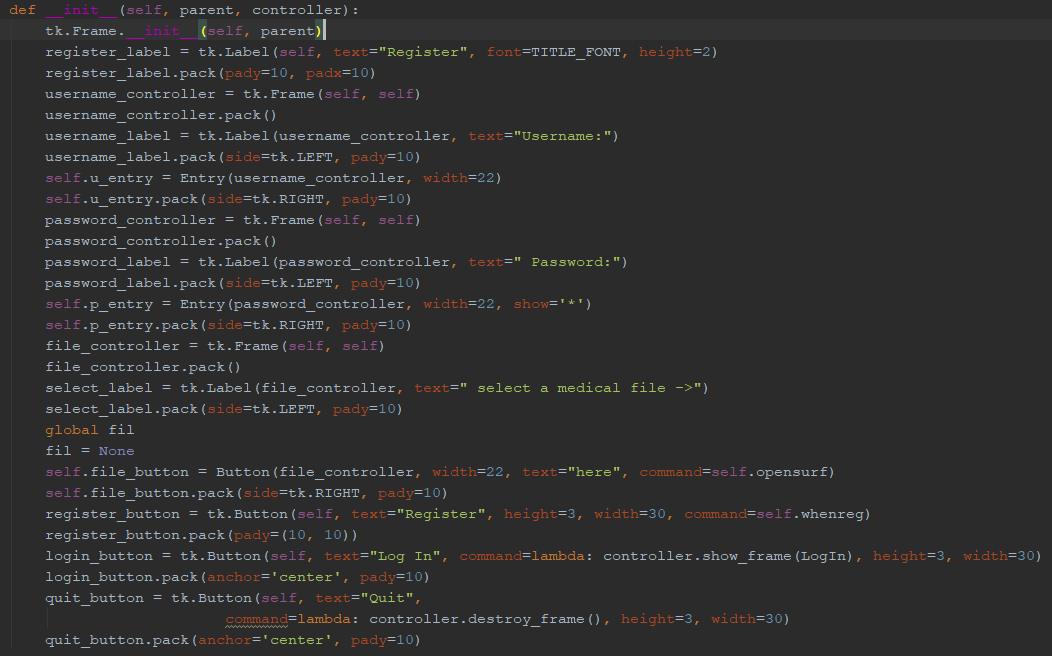
מחלקה שמגדירה את המסך המרכזי בממשק:



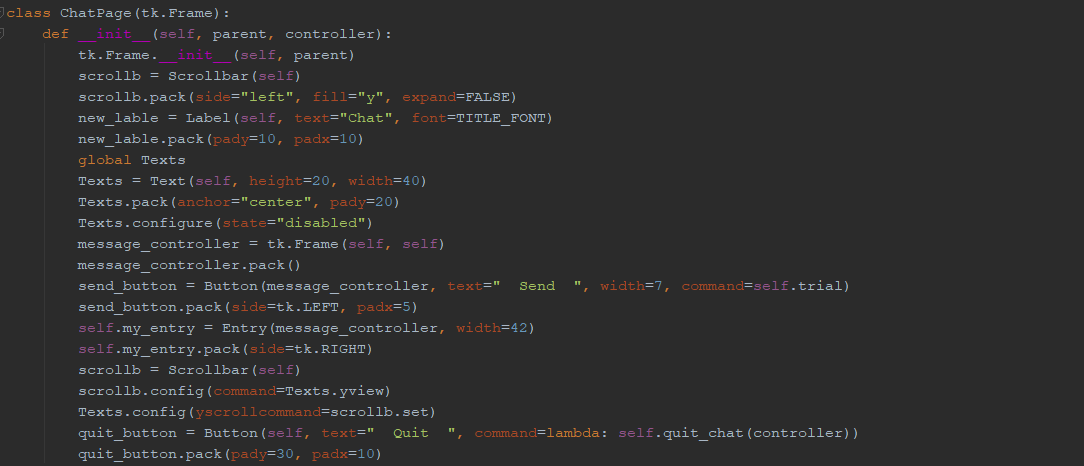
מחלקה שמגדירה את המסך בו מתבצעת התחברות למערכת:



מחלקה שמגדירה את המסך בו מתבצעת ההרשמה למערכת על ידי המשתמש:



מחלקה שמגדירה את מסך הצ'אט בו ניתן לראות את ההודעות הנשלחות ולשלוח הודעות:

****

**רפלקציה:**

הפרויקט שלי עסק בכמה נושאים שונים, גם ברשתות ותקשורת בין מחשבים אך גם בגיבוב ושימוש בdatabase בסיסי. גיוון זה נתן לי טעם ראשוני מכמה סוגים של נושאים מעניינים במיוחד ולמדתי הרבה נושאים חדשים בעזרת הפרויקט. כמו כן, למדתי ושיפרתי את השליטה שלי בחומרים ישנים יותר כמו תכנון פרויקט, כתיבת מחלקות וכתיבת קוד בצורה כללית יותר.

השתפרתי מאוד גם בכתיבת מסמכים (ייזום, אפיון ועיצוב) דבר שבהחלט יעזור לי בעתיד.

למדתי והשתפרתי מאוד בלמידה עצמית של חומרים וחיפוש ברשת האינטרנט פתרונות לסוגיות שונות.

נהניתי מאוד מכתיבת העבודה בכלל אבל בכתיבת הקוד בפרט, החלק של כתיבת המסמכים, למרות שהוא מלמד וחשוב אינו הצד החזק שלי והינו, לדעתי, החלק הפחות מעניין בכתיבת העבודה.

אני מרוצה מהתוצר הסופי של הפרויקט אך גם חושב שיש מה לשפר ואפילו חשבתי על כמה רעיונות לשיפור, העשרת והרחבת הפרויקט.

**ביבליוגרפיה**

שליחת וידאו בזמן אמת בפייטון :

<https://stackoverflow.com/questions/30988033/sending-live-video-frame-over-network-in-python-opencv>

ValueError: insecure string pickle :

<https://stackoverflow.com/questions/1746825/valueerror-insecure-string-pickle>

שליחת קבצים דרך פייטון:

<https://stackoverflow.com/questions/9382045/send-a-file-through-sockets-in-python>

גיבוב 256 בפייטון:

<https://www.pythoncentral.io/hashing-strings-with-python/>

תהליכונים בפייטון:

<https://www.tutorialspoint.com/python/python_multithreading.htm>

שמירה של מילון בקובץ pickle :

<https://stackoverflow.com/questions/11218477/how-can-i-use-pickle-to-save-a-dict>

ספרי הדרכה רשתות ופייטון:

<http://cyber.org.il>

יצירת תיקייה:

<https://stackoverflow.com/questions/273192/how-can-i-create-a-directory-if-it-does-not-exist>

מדריך ליצירת ui בצורה מונחת עצמים ב tkinter:

<https://pythonprogramming.net/object-oriented-programming-crash-course-tkinter/>

מד גלילה בפייטון :

<https://stackoverflow.com/questions/31762698/dynamic-button-with-scrollbar-in-tkinter-python>

הוספת תמונות לכפתורים:

<https://stackoverflow.com/questions/37515847/how-to-add-an-image-to-a-widget-why-is-the-image-not-displayed>

פתיחת חלון גלישת קבצים:

<https://stackoverflow.com/questions/10993089/opening-and-reading-a-file-with-askopenfilename>

צ'אט בנוי:

<https://github.com/xysun/pychat>

**נספחים**

**נספח א':**

1. תיאור הבעיה למחקר: שידור וידאו בזמן אמת.

מקור מידע רלוונטי: <https://stackoverflow.com/questions/30988033/sending-live-video-frame-over-network-in-python-opencv>

מסקנות שלב המחקר : לאחר שלב המחקר הגעתי למסקנה שהשימוש ב cv2 חיוני ושבכדי להגיע לתוצאות טובות בקלות רבה יש להשתמש בספריית struct ולפענח את ה string של ה numpy בעזרת ספריית pickle משום שהוא פשוט ומהיר.

היישום בפרויקט: שידור הוידאו והצגתו.

1. תיאור הבעיה למחקר: ניהול שרת בעל כמה חדרים המורכבים ממשתמש רגיל וממשתמש שהוא עובד.

מקור מידע רלוונטי: <https://github.com/xysun/pychat>

מסקנות שלב המחקר: כתיבת שרת כזה אינה קשה, ניתן לבנות רשימה של רשימות, במקרה שלי כל רשימה היא בגודל 2 בה socket של שני המשתמשים.

היישום בפרויקט: כתיבת שרת שעושה זאת בעזרת רשימה של רשימות.

1. תיאור הבעיה למחקר: שימוש בקבצי pickle.

מקור מידע רלוונטי: <https://stackoverflow.com/questions/11218477/how-can-i-use-pickle-to-save-a-dict>

מסקנות שלב המחקר: שמירה של מידע בעזרת קבצי pickle היא דרך קלה ביותר לשמור מידע, בעיקר מבני נתונים מסוימים, קל לגשת אליהם. המחקר שלי הראה לי איך לשמור על מידע בעזרתם, כתיבה אליהם ושימוש כללי בהם.

היישום בפרוייקט: שמירה של מידע כמו סיסמאות ושמות משתמש בקבצי pickle .

**נספח ב' – תהליך הבדיקות שבוצעו על המערכת:**

1. בדיקת מסכים:

הוקלדו לשדות השונים טסטים שונים בכדי לבדוק בעיות בקלטים שונים.

1. בדיקת אורך השידור – בדיקה אם אורך השידור משפיע על שידור הוידאו.
2. בדיקת כמות חדרים – נבדקו אם כאשר כמה חדרים קיימים ופעילים באותו זמן הם לא מקבלים את הקלט אחד של השני.
3. בדיקת פלט של שליחת נתונים מסוימים בעזרת הדפסות של הנתונים השונים.
4. בדיקת שמירה של הקובץ הנכון בתיקייה הנכונה.
5. בדיקת שמירה של מידע בקבצי ה pickle בצורה הנכונה והנוחה ביותר לגישה.
6. בדיקה שמטרתה לבדוק מה קורה אם המשתמש לוחץ על כפתורים שונים שאינם רצויים כמו לדוגמה כפתור היציאה (איקס) האדום במקרים מסוימים.
7. בדיקה של שליחה של מידע לא רגיל או נצפה אל ומהסרבר בשלבים מסוימים בהרצת התוכנה.
8. בדיקה של שליחה לא רצויה של מידע שמפריע לפרוטוקולים בשרת.
9. בדיקת תקינות קלט במסכים השונים. התוכנה בודקת אם הקלטים שהוכנסו מתאימים.

נספח ג':

|  |  |  |
| --- | --- | --- |
| # | שם מסמך | סוג השינוי |
| 1 | מסמך ייזום | טיוטה ראשונה |
| 2 | מסמך ייזום | טיוטה שנייה |
| 3 | מסמך ייזום | סיום |
| 4 | מסמך איפיון | טיוטה ראשונה |
| 5 | מסמך איפיון | סיום |
| 6 | מסמך עיצוב | טיוטה ראשונה |
| 7 | מסמך עיצוב | הגשה לבדיקה ראשונה |
| 8 | מסמך עיצוב | תיקונים |
| 9 | מסמך עיצוב | שינויים והגשה |

.