תיק פרויקט

המורה: יהודה אבני

מגיש: אופק כנרי

ת.ז:212140008

בית ספר: שמעון בן צבי

כיתה: י"ב 2

תאריך:19/04/2019

תוכן עניינים

רקע לפרויקט

הפרויקט שלי הוא תוכנה המאפשרת יצירת קשר בין שני מחשבים ברשת מקומית כך שלכל משתמש קיימת אופציה לשלוח בקשה לשחק עם משתמש שני, במידה והמשתמש השני הסכים לבקשה לשני המשתמשים יפתח חלון משותף ובו לוח המשחק וצ'אט. אותה התוכנה עולה בשני המחשבים על ידי המשתמש והסנכרון בין שני המחשבים נעשה על ידי הודעות טקסט בלבד, האבטחה של התוכנה באה לידי ביטוי בכך שכל התקשורת בין המשתמשים מוצפנת בעזרת מפתחות הצפנה.

התוכנה בנויה משני צדדים: צד שרת וצד לקוח. התקשורת נעשית על ידי התחברות של כל אחד מהלקוחות לשרת אחד קבוע שרץ ללא הפסקה.

בחרתי לעשות את הפרויקט על נושא זה מכמה סיבות:

נושא זה של תקשורת בין שני משתמשים במשחק אינטרנטי עניין אותי רבות מכיוון שאני משחק בעצמי במשחקי מחשב הדורשים חיבור לאינטרנט.

משתמשים פוטנציאליים:

- צעירים
- חובבי המשחק צוללות

התוכנה שלי מורכבת מלקוחות שכוללים בתוכם מספר טפסים ושרת:

מספר משתמשים וסרבר אחד, המשתמשים מתקשרים עם השרת והוא אחראי לשליחת ההודעות אל המשתמש השני. המשתמשים מתקשרים אחד עם השני באמצעות הודעות טקסט בלבד, כל שינוי בחלון האפליקציה גם הוא עובר באמצעות הודעת טקסט.

תחומים

אבטחת מידע

תקשורת בין מחשבים

בסיס נתונים

מטרות הפרויקט

המוצר המוגמר אמור לבצע את הדברים הבאים:

המוצר המוגמר אמור לתפקד בתור תוכנה המאפשרת משחק צוללות וצ'אט מאובטח ונוח בין שני משתמשים.

מטרות מרכזיות:

לאפשר יצירת קשר בין משתמשים לשרת ואז לזהותם (בעזרת הרשמה והתחברות) ולאחר מכן יצירת מרחב למשחק זהה בין שני המשתמשים, כל משתמש ישחק בתורו . כל שינוי שייעשה יעשה באחריות המשתמשים וירשם בקובץ טקסט שישמר במחשב השרת, קצב העברת הנתונים יהיה מהיר, ללא עיכובים, אחרת עלול להשתבש עיבוד ההודעות.
בנוסף לכך, הקשר בין המשתמשים לכל אורך הדרך יהיה מוצפן ומאובטח כך שגם אם צד שלישי מאזין לקשר, הוא לא יידע לפענח את ההודעות העוברות.

אבטחת החשבונות תתבצע באמצעות דואר אלקטרוני אשר מקושר לחשבונו של המשתמש ובעזרתו הם יאשרו את חשבונם

שפות התכנות וסביבות העבודה

בפרויקט שלי עשיתי שימוש ב #C וב-SQL, כאשר ב- #C רשמתי את מרבית הפרויקט והוא אחראי על ההצפנה והפענוח, על שליחת המידע דרך הסוקטים, על ממשק המשתמש, ועל פעילות הסרבר שכולל את קבלת ההודעות פענוח שלהם עיבודם ושליחת הודעת תגובה למשתמש/ים הנדרשים. המידע על המשתמשים הרשומים נשמר בשרת SQL, השרת מעביר את המידע לסרבר והסרבר שולף ממנו נתונים בעזרת שילוב השפות SQL ו #C.

הפרויקט נכתב בתוכנת הפיתוח Visual Studio 2015.

הפרויקט נרשם ונבדק אך רק על מערכת ההפעלה windows 10 64 bit, אולם הוא אמור לעבוד בכל שאר מערכות ההפעלה המודרניות של ווינדוס.

פירוט הבעיה האלגוריתמית וניתוחה

1. ביצוע תקשורת עם מספר לקוחות בו-זמנית.

השרת רשאי לתקשר אך ורק עם לקוח אחד על סוקט מסוים, לכן הוא איננו יכול לתפקד כאשר מחוברים אליו 2 משתמשים או יותר אם הוא עובד רק עם Socket אחד. במערכת שלנו יש צורך שהשרת ידע לתקשר בו זמנית עם מספר רב של משתמשים.

2. הצפנת המידע נגד צד שלישי:

על מנת שצד שלישי אשר "יושב" על החיבור בין הלקוח לשרת לא יוכל להבין את ההודעות אשר עוברות בין שני הצדדים, ועל מנת שלא נצטרך להעביר מידע רגיש כלשהו כדי להתחיל את התקשורת, יש להשתמש בהצפנה א-סימטרית. עם הצפנת המידע צצה עוד בעיה, להמיר את המידע לבייטים, להצפין אותו, לשלוח ולהמיר בחזרה בלי שהמידע ייפגם בנוסף על הצפנה א-סימטרית קיימת הגבלה על גודל המחרוזת שנשלחת.

לכן נשתמש בהצפנה א-סימטרית על המפתחות של הצפנה סימטרית ולאחר מכן נשתמש בהצפנה סימטרית לכל אורך הפעולה.

3. יצירת CAPTCHA - בדיקת אנושיות:

על מנת שתכנות רובוטיות וזדוניות למיניהן לא יוכלו להציף את השרת או לנסות לפרוץ לחשבונות הלקוחות (BRUTE FORCE), יש לבצע בדיקת CAPTCHA בכל התחברות של לקוח חדש. על בדיקת הCAPTCHA להיווצר בצד השרת כדי שללקוח זדוני לא תהיה את האפשרות לדוג את הטקסט הנכון מהקוד של הלקוח ועליה להיות מועברת בצורה מוצפנת . כמו כן על הבדיקה של נכונות הפתרון לCAPTCHA הנשלח מהלקוח לשרת להיות בצד שרת, כדי שהטקסט הנכון לא יועבר בתקשורת בין השרת ללקוח.

יש ליצור בכל פעם שנוצרת CAPTCHA טקסט אקראי בעל רווחים משתנים בין התווים וקשקושים אקראיים בתמונה כדי שרק בן אנוש יוכל לפתור אותה.

4. <u>זיהוי הבקשות של הלקוח ותגובות השרת</u>

השרת מאזין כל הזמן ללקוח ולהפך, הבעיה העיקרית של תקשורת זו היא ההבנה ביניהם, לדוגמה כאשר הלקוח רוצה לבקש מלקוח אחר לשחק איתו כיצד השרת יודע שהלקוח רוצה לבקש משחק ולא לכתוב בצ'אט. השרת צריך לדעת איזו פעולה הלקוח רוצה לעשות בכל פעם שהוא מקבל ממנו הודעה, גם כן הלקוח צריך לדעת מהי המשמעות של כל הודעה מהשרת.

5. <u>שליחת דואר אלקטרוני</u>

הוספת רובד של אבטחה ואימות מתרחשת בעת קישור דואר אלקטרוני לכל משתמש. לכן, בכל הרשמה של משתמש חדש יש על השרת לשלוח הודעת דואר אלקטרוני ללקוח, אשר מכילה קוד שעל המשתמש לשלוח בחזרה לשרת. שליחת הדואר האלקטרוני מתבצעת באמצעות חשבון מייל קבוע מראש כאשר ישנו שימוש ב SMTP - Simple Mail Transfer Protocol לכתיבת הודעה מהדואר האלקטרוני במקרה שלי - Gmail. התוכנה מייצרת קוד - רצף של אותיות באופן אקראי ושולח אותם בגוף ההודעה, לאחר מכן ללקוח נפתח חלון להכנסת הקוד (ישנה אפשרות לקבלת קוד חדש מהשרת). במידה והקוד נכון הלקוח יתחבר לתוכנה, במידה ולא תופיע הודעת שגיאה שהקוד שהוזן לא נכון.

6. <u>ייעול זמן תגובה של התקשורת</u>

בכדי שהיישום יהיה אפשרי ונוח לתפעול צריך לקצר את זמן התגובה של השרת לבקשות הלקוח. בתחילה השרת מתחבר אל בסיס הנתונים והדבר גורם לעיקוב משמעותי בזמן התגובה. בנוסף כאשר השרת מכניס משתמש למערכת עליו קודם כל לבדוק האם שם המשתמש תקין (האם הוא קיים כבר במערכת). לאחר מכן בודק האם ניתן לשלוח דואר אלקטרוני לכתובת הדואר האלקטרונית שהתקבלה ובמידה וניתן שולח לו הודעת דואר אלקטרוני שמכילה קוד סודי ובסוף התהליך מכניס את הלקוח אל בסיס הנתונים, פעולה זו לוקחת זמן רב ויש למצוא דרך לייעל אותה.

פתרונות - יתרונות וחסרונות

1. ביצוע תקשורת עם מספר לקוחות בו-זמנית

   a. שימוש בחיבור בפרוטוקול TCP-

      i. יתרונות-

         1. חיבור אמין ועקבי בין הלקוח לשרת, כל מידע שיישלח מהלקוח יתקבל בשרת ולהפך.

      ii. חסרונות-

         1. חיבור איטי לעומת אופציות אחרות, דורש מעבר של יותר הודעות.

   b. שימוש ב Socket נפרד לכל לקוח

      i. יתרונות-

         1. עבודה עם כמה לקוחות בו זמנית ולמנוע ו"דריסת" Socket.

      ii. חסרונות-

         1. יוצר עומס, חוסר יעילות בניצול מירבי של כל Socket.

2. הצפנת המידע נגד צד שלישי

   a. שימוש בRSA-

      i. יתרונות-

         1. כל התקשורת תהיה מאובטחת. פחות סכנות של האזנות מצד שלישי.
         2. לא נדרשת הבנה מעמיקה כיצד מתבצעת ההצפנה

      ii. חסרונות-

1. לא ניתן לדעת מה השרת מעביר ללקוח בכדי לפקח על תגובות השרת.
2. הגבלה על גודל המחרוזת הנשלחת.

b. שימוש בהצפנה סימטרית
   i. יתרונות-
      1. פחות הגבלה על גודל המחרוזת הנשלחת.
   ii. חסרונות-
      1. פחות מאובטח מהצפנה א-סימטרית

3. <u>יצירת CAPTCHA - בדיקת אנושיות - בצד שרת ולהעביר אותה ללקוח</u>

a. בדיקה של קוד-

i. יתרונות-

1. מונע מעבר של לקוחות ללא בדיקה. במידה והבדיקה בצד לקוח, הלקוח יכול לאשר את הבדיקה בעצמו.

ii. חסרונות-

1. מכיוון שהקוד נוצר בצד הלקוח קיימת אפשרות להסתכל על דרך יצירת הקוד.
2. אין עבודה מול השרת – יכול להוביל לבעיית אבטחה.

b. יצירת ה CAPTCHA-

i. יתרונות-

1. יצירת צירוף אותיות ומספרים אקראי כל פעם מחדש, כמו כן קשקוש אקראי על התמונה.

ii. חסרונות-

1. יצירת צירוף אותיות בעל משמעות גסה. דרך פתרון: שמירת מאגר מילים גסות ובדיקה כי אף אחת מהמילים לא מופיעה.

4. <u>זיהוי הבקשות של הלקוח ותגובות השרת</u>

a. יצירת הודעה בתבנית מוסכמת על ידי הלקוח-

i. יתרונות-

1. ייעול הקוד

ii. חסרונות-

1. שימוש בתבנית קבועה מגביל את המידע שניתן להעביר בהודעה

b. זיהוי הפקודות על ידי השרת בעזרת מילות מפתח-

i. יתרונות-

1. הבנה טובה של השרת והלקוח.

ii. חסרונות-

1. לא מוגן מפני צד שלישי. מאוד קל לזהות מהן הפקודות המוסכמות.

5. <u>שליחת דואר אלקטרוני</u>

a. שימוש בחשבון מייל

i. יתרונות-

1. ניהול חשבון בצורה נוחה.

ii. חסרונות-

1. נתון לפרצה.

b. שימוש בפרוטוקול Smtp

i. יתרונות-

1. אוטומציה פשוטה ונוחה לשליחת מייל
2. תמיכה בפרוטוקול SSL

ii. חסרונות-

1. פרטי ההתחברות למייל מופיעים בקוד

6. <u>ייעול זמן תגובה של התקשורת</u>

a. שליחה מוקדמת ללקוח

i. יתרונות-

1. ייעול זמן התגובה של התקשורת

2. דילוג על זמן הכנסת הפרטים למערכת

ii. חסרונות-

1. השרת אינו בטוח כי הכנסת הנתונים הצליחה ולמרות חוסר הוודאות שולח השרת אישור להתחברות הלקוח

הפתרון שנבחר

כעת נפרט יותר על הפתרונות שממשנו, למה ממשנו אותם, איך ממשנו אותם בקוד, ונביא דוגמאות.

1. ביצוע תקשורת עם מספר לקוחות בו-זמנית

תחילה השרת פותח את האזנה ומחכה ללקוחות שיתחברו. ברגע החיבור השרת מייצר עצם של המחלקה Client.

אחר יצירת האובקייט Client מתחיל תהליך התקשורת בין אותו לקוח לשרת.

לכל לקוח השרת מייצר TCPClient משלו ועל ידי כך פותח Socket לכל לקוח.

2. הצפנה - RSA

בתחילת התקשורת שני הצדדים מעבירים אחד לשני את המפתח הציבורי.

קוד העברת המפתח הציבורי וקבלת המפתח הציבורי של הצד השני:

לאחר מכן, כל מידע שעובר בין השרת והלקוח עובר קודם כל הצפנה ולאחר מכן פענוח:

פעולות הצפנה ופענוח:

חלק מההמרות נעשות בפורמט מחרוזת מבסיס 64 על מנת למנוע פגימה במידע המועבר

3.יצירת CAPTCHA - בדיקת אנושיות - בצד שרת והעברתה ללקוח

התוכנה מייצרת את הCAPTCHA לאחר ניסיון ההתחברות / ההרשמה.

-המחלקה Captcha:

-לאחר יצירת התמונה התוכנה תציג חלון שבה מופיעה התמונה.

-כאשר הלקוח מכניס את תשובתו, מתקיים אימות ולאחר האימות האם היא נכונה או לא.

4.זיהוי הבקשות של הלקוח ותגובות השרת-

-כאשר השרת מקבל הודעה מהלקוח מתבצע ניתוח ההודעה לפי מילות מפתח בתחילת המחרוזת.

5.שליחת דואר אלקטרוני-

-לצורך שליחת הדואר האלקטרוני קיימת מחלקה ובה נוצרת ההודעה ומתרחשת ההתחברות אל חשבון הGmail ושולחת את הודעה אל המייל הדרוש

6.ייעול זמן תגובה של התקשורת-

-בכדי לייעל את מירב הזמן השרת מבצע התחברות אל בסיס הנתונים כבר בתחילת ריצת הפרויקט.

דרישות ומגבלות המערכת

דרישות

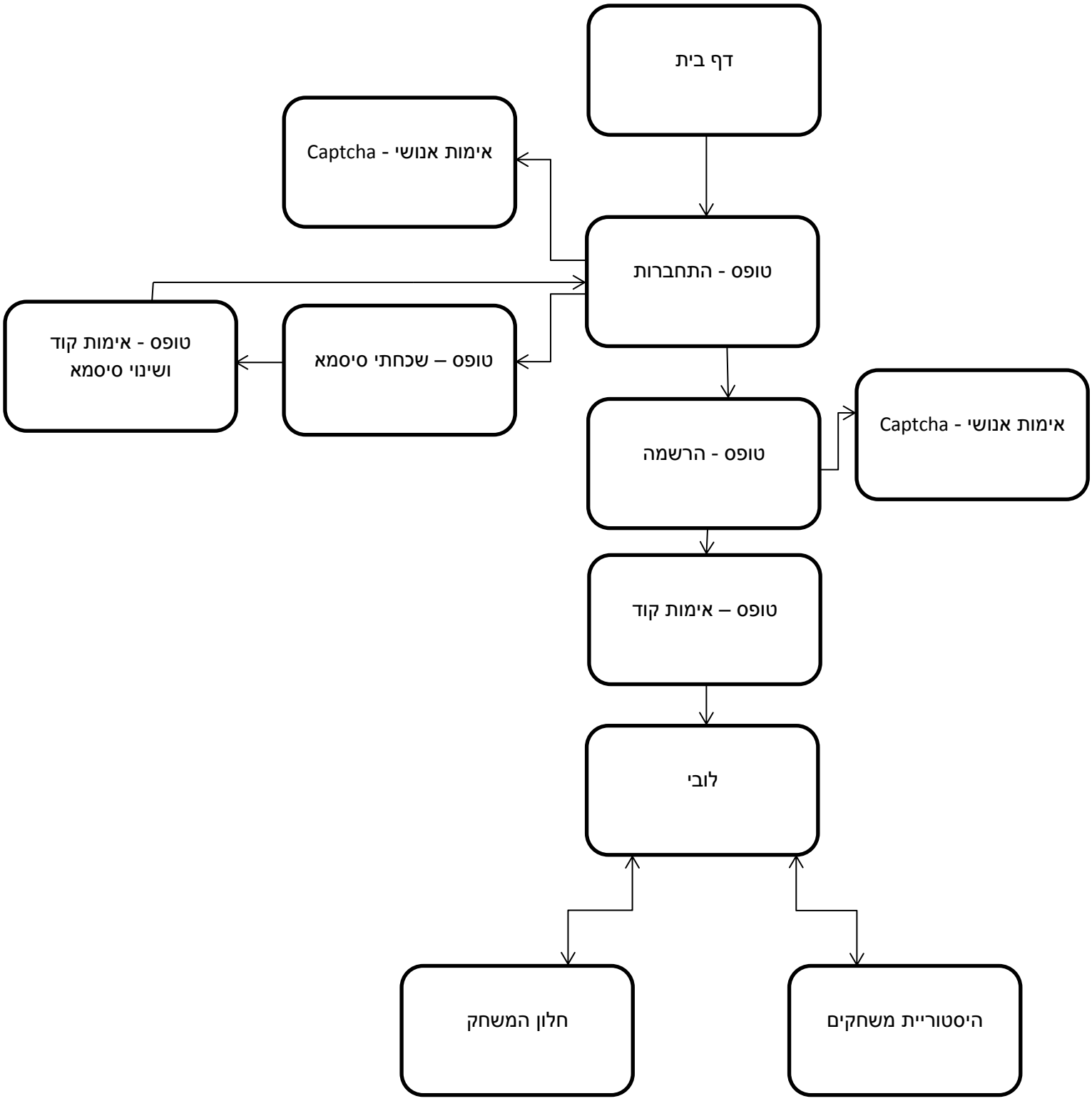
על מנת שהתכנית תפעל ישנן כמה דרישות שצריכות להיענות קודם לכן:

- מערכת הפעלה Windows XP ומעלה על המחשב.
- התוכנה Visual Studio 2015 מותקנת על המחשב.

מוגבלויות

- תמיכה בשפה האנגלית בלבד.
- הנחה בסיסית שהלקוח יודע את הIP והפורט שעליהם מאזין השרת.
- הגבלה של 256 בייטים בסטרים בין השרת ללקוח.
- יכול להיווצר עיכוב בתקשורת בין הלקוח לשרת בתחילת התקשורת עקב החלפת המפתחות הציבוריים.

ממשק משתמש


דף בית
אימות אנושי - Captcha
טופס - התחברות
טופס - אימות קוד
ושינוי סיסמא
טופס – שכחתי סיסמא
אימות אנושי - Captcha
טופס - הרשמה
טופס – אימות קוד
לובי
חלון המשחק
היסטוריית משחקים