גיבוי בענן פרויקט גמר בסייבר

שם התלמיד:דן שמש

ת"ז: 206484107

מנחים: מיכאל צ'רנובילסקי וגד רוזנטל

בית הספר: אורט גוטמן נתניה

# תקציר כולל ורציונל הפרויקט

לאנשים רבים בעולם יש צורך במערכת לניהול קבצים השמורים במקום מרוחק על רת מרוחק על מנת לשמור עותקים נוספים של הקבצים שלהם.

כדי לענות על צרכים אלו נוצרו ממשקים שונים של חברות שונות המספקים שירותי שמירת קבצים על שרתים מרוחקים כגון(דרייב של גוגל ודרופבוקס).

המטרה היא ליצור עותק מאובטח אליו רק למשתמש שיצר אותו יש גישה כדי שיוכל לשחזר קובץ

שנמחק, לחסוך מקום שמירה על המחשב בעזרת שמירת הקבצים על שרת מרוחק ועוד.

התוכנה שלי אמורה לספק שירות שמירת קבצים על ענן (כלומר שרת מרוחק ומאובטח) הנמצא במקום שונה לגמרי ממיקום המשתמש. אשר יאפשר למשתמשים הרשומים עליו לשמור,למחוק,לערוך ולשתף משתמשים אחרים הרשומים למערכת.



# מבוא ורקע כללי

הפרויקט שלי מבוסס על תקשורת בין מחשב מרוחק למספר מחשבים שונים המחוברים אליו בו זמנית וניהול הבקשות והקבצים של המשתמשים הללו ע"י השרת המדובר. בין המחשבים ברשת נוצר חיבור עם השרת המרכזי, וחיבור זה מאפשר שליחת בקשות של המשתמשים עליהן השרת עונה במידה והן תקינות.

הנושא המרכזי של הפרויקט הוא **ניהול קבצים של משתמשים שונים ע"י מחשב שרת מרוחק(ענן)**.על השרת המרוחק תיקייה שבה שמורות תיקיות בעלות שמות כל המשתמשים הרשומים. השרת מעניק למשתמש גישה לתיקיה שלו ולקבצים שלו רק לאחר שאותו משתמש ביצע הרשמה והתחברות שאושרו על ידי השרת.

.

לפרויקט יש מספר מרכיבים עיקריים ולהלן סקירת רקע לגביהם:

* מרכיב עיקרי בפרויקט שלי הוא **תקשורת**. תקשורת בעולם המחשבים מבוססת על Sockets, שהם נקודות קצה עבור זרם נתונים בתקשורת בין תהליכים על גבי רשת מחשבים. רוב הסוקטים כיום מבוססים על ה- Internet Protocol (IP).

בפרויקט שלי התקשורת בין השרת למנהל הפעולות של הלקוחות מתבצעת באמצעות Sockets, ובין ממשק המשתמש למנהל הפעולות יש שימוש בסוקט נוסף בו מנהל הפעולות מאזין כמו שרת לבקשות המשתמש בממשק.

* בנוסף, הפרויקט משלב עבודה עם **מסד נתונים**. הנתונים שהתוכנה מקבלת לגבי כל משתמש כמו שם משתמש,סיסמא,קבצים נשמרים במסד הנתונים, שהעבודה איתו היא בטכנולוגיית ה- SQLite, שמממשת את רוב הסטנדרט של SQL (שהיא השפה המקובלת לטיפול ועיבוד מידע בבסיסי נתונים). להבדיל מרוב המערכות לניהול בסיסי נתונים, SQLite אינה תהליך עצמאי נפרד, אלא ספריה בתוך התהליך באופן אינטגרלי. ‏ זוהי בחירה פופולרית כאשר מפתחים בסיס נתונים מוטמע עבור צד לקוח כשירות ניהול מידע מקומי. מסד הנתונים שאותו מנהלת התוכנה שלי שומר מידע לשימוש מקומי בלבד, ולכן בחרתי להשתמש ב- SQLite שלא דורש הרצה של SQL Server.

על מנת לצפות בנתונים שנשמרו במסד הנתונים השתמשתי בsqlite browser מכיוון ש sqlite לא מאפשר צפייה בנתונים שנשמרו במסד.



שם המוצר: dan's cloud.

המוצר הוא יצירת שירות בעל יכולת גיבוי מסמכים בענן שבו אפשרות לשתף משתמשים מורשים (כלומר תיקייה שיתופית הנמצאת במחשב מרוחק).

מערכת לניהול גיבויים, איסוף וגיבוי קבצים ששונו, בצורה יעילה ממספר מחשבים ברשת, שרת בעל יכולת לנהל מספר משתמשים בו זמנית ולענות לכולם יתפעל את המערכת ממחשב מרוחק אשר עליו התוכנה תהיה מותקנת ותפעל תמיד.

כניסה למסמכים השמורים ב"ענן" ואפשרות לשמור דברים ב"ענן" תהיה למורשים בלבד. כלומר רק אדם אשר נרשם למערכת ומתחבר לחשבונו האישי והפרטי יוכל לראות את קבציו ואת הקבצים אשר שיתפו איתו.

כאשר משתמש ירצה להיכנס לתיקייה הוא יתבקש להיכנס במערכת login (GUI) על ידי הכנסת שם משתמש וסיסמא, המערכת תבדוק אם המשתמש המנסה להיכנס למערכת אכן רשום (כל הרשומים יהיו ב-DATABASE) ואמורה להיות לו גישה למערכת, אם לא המערכת תחסום את הכניסה שלו אליה. אם המשתמש אכן רשום הוא יועבר למערכת שבה הוא יוכל לראות את כל התיקיות והקבצים שלו בצורה נוחה.

הקבצים במערכת יהיו מוצפנים על מנת להבטיח שמירתם, הכניסה למערכת תתאפשר על ידי השרת רק לאחר זיהוי המשתמש כמשתמש רשום וזאת באמצעות המפתח הקיים בשרת.

כמו כן, כדי לשמור על סודיות המשתמשים הרשומים הנתונים שלהם ב-DATABASE יהיו מוצפנים אף הם ולא חשופים לכל אדם.

# שפת התכנות וסביבת העבודה

הפרויקט נכתב ב- Python וב- C#. כל הפרויקט נכתב בשפת התכנות python וממשק המשתמש נכתב בc# ויש קשר ביניהם באמצעות ציונורות(pipes).

סביבת העבודה עם Python הייתה בPyCharm שהיא תוכנה מקצועית ונוחה לעבודה על פרויקטים מרובי מודולים וקבצים. היא שימשה גם לפיתוח הקוד ועבודה במקביל עם כל המודולים השונים.

סביבת העבודה עם C# הייתה Microsoft Visual Studio 2010 ששימשה לכתיבת ה- GUI באמצעות יצירות טו]ס שאותו עיצבתי.

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. ממשק המשתמש הויזואלי בו השתמשתי כדי לראות את טבלאות המסד היא sqlite browser



מודול לניהול הרשאות: קביעת קבוצות הרשאה של משתמשים או הכרת משתמשים יחידים לספריות השונות. כלומר החלטה למי יש גישה לאיזה תיקייה או קובץ.

מודול רישום: מודול זה יהיה אחראי על רישום של משתמשים חדשים והכנסתם לDATABASE.

מודול הזדהות: טופס בו אני מבקש מהמשתמש להכניס שם משתמש וסיסמה על מנת לוודא כי המשתמש אכן משתמש רשום וחוקי שמגיעה לו גישה לקבצים או תקיות במערכת.

מודול הצפנה ופענוח: מודול זה יהיה אחראי להצפנת הקבצים ופענוח הקבצים עבור משתמשים.

מודול ניהול קבצים: מודול זה יתעסק בהעלאת הקבצים לתקיית המשתמש ושינויים במידת הצורךץ בנוסף לכך המודול יתעסק בשיתוף הספריות כלומר קביעת מי ישותף וכו.

מודול שמירת המידע לDATABASE: מודול זה לוקח את המידע ומשתמש במודול ההצפנה והפענוח על מנת להציג את המידע לבעל ההרשאה המתאימה וכמו כן ינהל את האינטראקציה מול מסד הנתונים.

# ניסוח וניתוח הבעיה האלגוריתמית

במהלך פיתוח המוצר נתקלתי בבעיות אלגוריתמיות רבות:

חיבור הממשק משתמש למנהל הפעולות של המשתמש ולשרת של הענן:

פתרון: על מנת לחבר את הממשק משתמש לשרת בסופו של דבר ועל מנת שכל המידע אותו השרת צריך לקבל יגיע מפעולות המשתמש בממשק ולכן החלטתי להפוך את מנהל הפעולות של המשתמש לשרת רק עבור אותו ממשק ואת הפעולות והמידע אותו השרת במנהל הפעולות מקבל מנהל הפעולות שולח לשרת המטפל בכל המשתמשים.

העלאת קבצים הנמצאים על מחשב המשתמש לתיקיות על השרת המרוחק:

פתרון: על מנת להעלות את הקבצים הנמצאים על המחשב של המשתמש אל תיקיה עם שמו על שרת מרוחק החלטתי שברגע שהמשתמש בוחר איזה קובץ הוא רוצה להעלות לתיקייה שלו הממשק ישלח את שם המשתמש של המשתמש הנוכחי, יפרק את הקובץ וישלח אותו בחלקים לשרת, השרת יחבר את החלקים ויכתוב את הקובץ מחדש לתיקייה עם שמו של המשתמש.

מחיקת קבצים מהתיקייה:

כאשר המשתמש יבחר את הקובץ אותו הוא רוצה למחוק הממשק ישלח את שם הקובץ לשרת ואת שם המשתמש, השרת יחפש בתיקיית המשתמש את הקובץ המדובר ויסיר אותו לאחר מכן ישלח לממש שהקובץ נמחק ויבקש מהמשתמש יעדכן את המסך שלו.

# מבט אישי על העבודה ותהליך הפיתוח

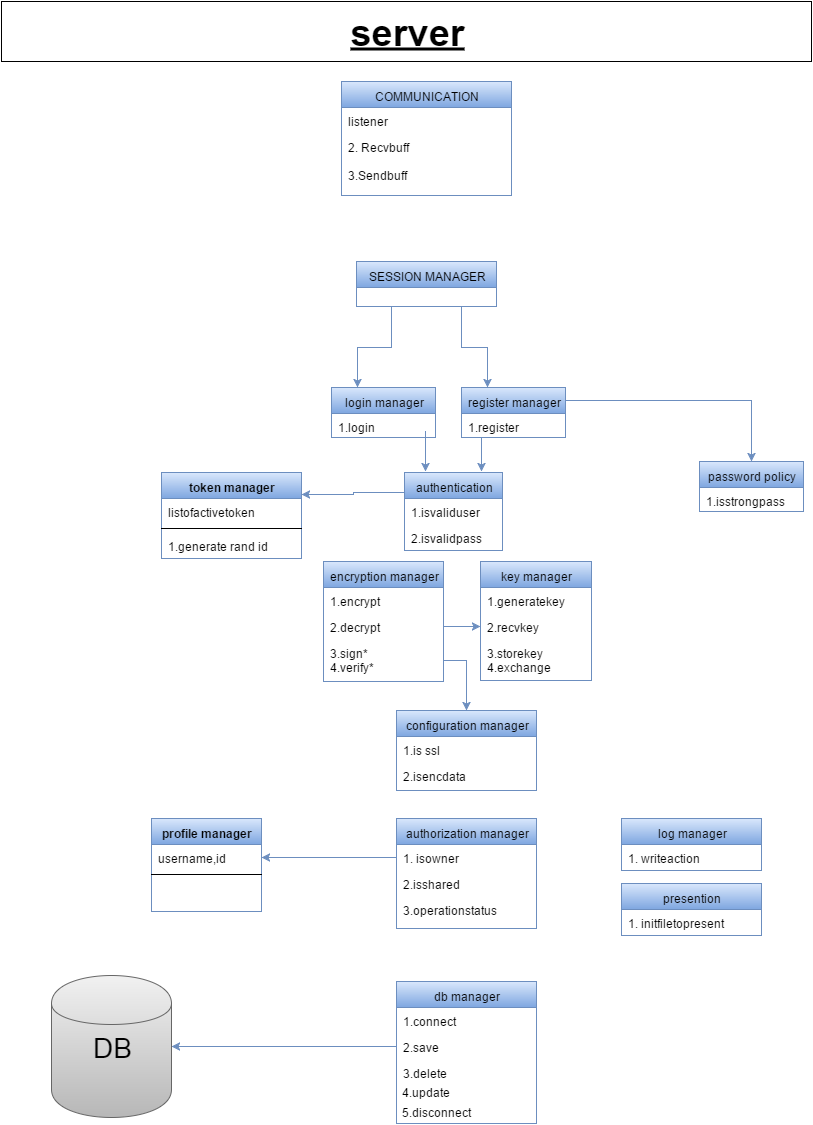
במהלך העבודה על הפרויקט למדתי לראשונה על דברים שונים שניתן לעשות כמו העברת קבצים בין מחשבים שונים, שימוש בממשק משתמש על מנת להקל על המשתמש בשימוש באפליקציה אותה יצרתי, כיצד להשתמש בספריית PIL על מנת לעצב את הטקסטים ולהפוך קבצי טקסט לתמונות ועד.

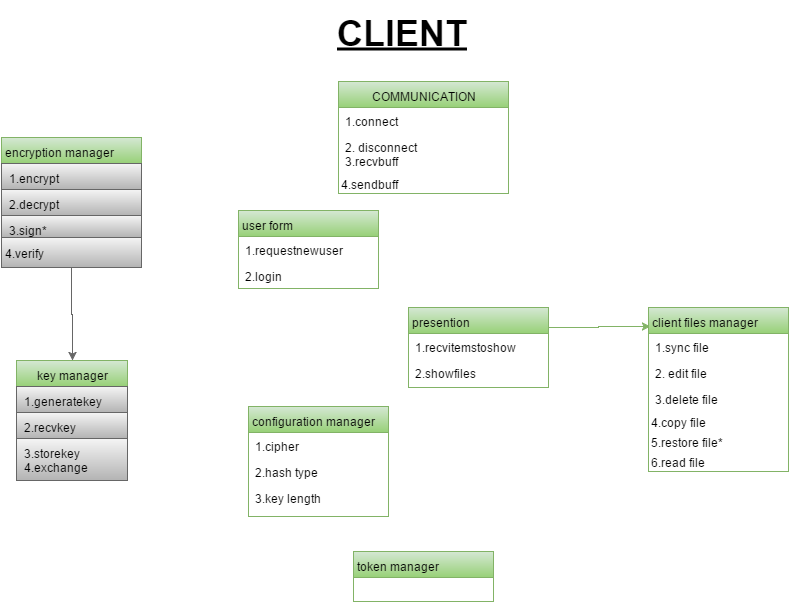
פיתחתי הרגלי תכנות טובים במהלך העבודה על הפרויקט, הכוללים חלוקה למודולים ותכנון לוגי מראש, תיעוד מלא, שימוש בקבועים, במחלקות ובפונקציות, נתינת שמות משמעותיים למשתנים.

האתגרים והקשיים שנתקלתי בהם במהלך פיתוח הפרויקט הם:

* כאשר עבדתי על הקישור בין ממשק המשתמש הכתוב בC# לקבצי ההרצה של השרת והמשתמש הכתובים בPYTHON נתקלתי בקשיים מכיוון שמעולם לא קישרתי בין תוכנות שונות בהן כתובים בשפות תכנות שונות לגמרי.

הערכת הפתרון לעומת התכנון:





# ביבליוגרפיה

1. אתר הדרכה לסייבר שיוסד על ידי המנחה שלי מיכאל:  
   [https://sites.google.com/site/coursecyber](https://sites.google.com/site/coursecyber/)
2. הפורום הגדול ביותר לשאלות ותשובות בנושאי תכנות:  
   [http://stackoverflow.com](http://stackoverflow.com/)
3. MSDN, הרשת הרשמית של Microsoft למפתחים:  
   <http://msdn.microsoft.com/library>
4. התיעוד הרשמי של Python:  
   [https://docs.python.org/2.7](https://docs.python.org/2.7/)
5. הוויקי הרשמית של Python:  
   [https://wiki.python.org/moin](https://wiki.python.org/moin/)
6. להורדת Python 2.7 (32-bit):

<https://www.python.org/ftp/python/2.7/python-2.7.msi>

1. להתקנת ספריית PyWin32 בשביל Python 2.7 (32-bit):

[http://sourceforge.net/projects/pywin32/files/pywin32/Build 219/pywin32-219.win32-py2.7.exe/download](http://sourceforge.net/projects/pywin32/files/pywin32/Build%20219/pywin32-219.win32-py2.7.exe/download)

1. להתקנת ספריית PyCrypto בשביל Python 2.7 (32-bit):

<http://www.voidspace.org.uk/downloads/pycrypto26/pycrypto-2.6.win32-py2.7.exe>