גיא גור-אריה - Repositories

"איש שלום" כפר-יונה

תיק פרויקט – Repositories

שם התלמיד: גיא גור-אריה

ת.ז.: 212499347

שם המנחה: אופיר שביט

גיא גור-אריה - Repositories

הפרויקט יהיה תוכנה שבה למשתמש תהיה גישה למאגר או מאגרי קבצים, ושם יוכל להעלות קבצים, להוריד קבצים למחשב שלו או למחוק קבצים, ועוד אפשרויות.
משתמשים יצטרכו להירשם באמצעות שם וסיסמא עם אפשרות להכניס גם את שמם המלא, אך להתחבר רק באמצעות שם וסיסמא.

כיום קיימות כמה תוכנות מארי קבצים וירטואליים, או "ענן", המוכרת מבניהן היא "Google Drive" של גוגל.

בפרויקט עצמו אין הרבה חידושים – העלאה והורדה של קבצים זה הבסיס, וישנן עוד כמה אפשרויות קטנות כמו שינוי שם ומחיקה. המספר הקטן של האפשרויות עוזר בהפיכת התוכנה ליותר ידידותית למשתמש.

אתגרים מרכזיים:

האתגר המרכזי הוא שליחת הקבצים והצגתם באופן יעיל ונוח לעין. את השרת ואת הלקוח התחלתי לכתוב ב-python, אך מפני שחיפשתי שפה שבה ה-GUI הוא נוח ואסתטי, עברתי לכתוב את הלקוח ב-C# wpf.
לאחר מכן הבעיה שנוצרה היא איך אני שולח קבצים ב-C# ומקבל אותם ב-python, ולהיפך.
הבעיה הזאת רק עוררה עוד בעיות – איך לשלוח קבצים ולקבל קבצים במקביל, איך להתמודד עם כמות גדולה של פקודות מהלקוח לשרת, ואיך לבטל שליחה או קבלה של קבצים.

מכך נוצרה הבעיה לתקשר בין הלקח הכתוב בC# לבין השרת הכתוב בpython, ולעשות זאת בצורה נוחה ומסודרת.

פתרון הבעיה – סידור העבודה לתתי משימות ועשייתה בפרוטוקול מסוים, כך שנוח להבין הכל ולחזור ולתקן דברים בהמשך.

בחרתי לעבוד על הנושא מפני שזהו משהו שאני משתמש בו ביום יום, ורציתי לקחת על עצמי את האתגר הזה, לתקן מה שאני חושב שטעון בשיפור באפליקציות ענן אחרות, כגון נוחות ופשטות.

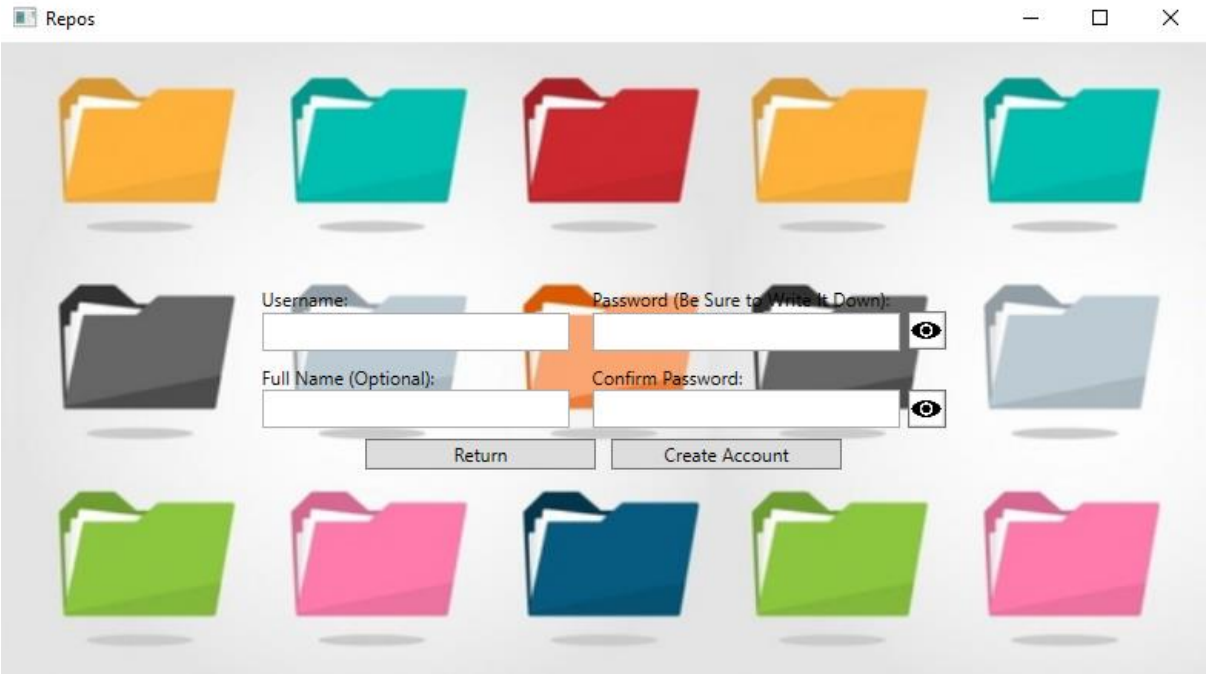
הפרויקט עונה על הצורך לאכסן קבצים חשובים במקום בטוח. אם קבצים נשמרים על המחשב ומשהו קורה למחשב, הקבצים האלה אבודים לתמיד. הפרויקט שלי מציע סביבת עבודה נוחה ובטוחה לשמירת עבודות או פרויקטים אחרים, במקום להשתמש ב-USB drive או לשמור אותן על המחשב עצמו.

גיא גור-אריה - Repositories

תיאור פונקציונלי של הפרויקט:

בעת כניסה לאפליקציה, יוצג מסך כניסה אשר יהיו בו האפשרויות להתחבר באמצעות שם וסיסמא.
אם המשתמש ירצה משום מה לצאת מהאפליקציה, זה יעשה באמצעות לחיצה על כפתור הX בפינת החלון עצמו.

בנוסף, אם אין למשתמש חשבון קיים בתוכנה, יש לו את האפשרות להירשם באמצעות לחיצה על כפתור הREGISTER, אשר יביא אותו למסך ההרשמה, שם יצטרך להכניס שם משתמש (חייב להיות אחד שלא קיים כבר, התוכנה תתריע לו אם קיים כבר שם כזה), סיסמא ואימות סיסמא (התוכנה תבקש שהסיסמאות תהינה זהות), ושם מלא, אשר השרת יקרא לו בשם הזה (שם זה לא חייב להיות unique, ואם השדה יישאר ריק השם יהיה שם המשתמש).



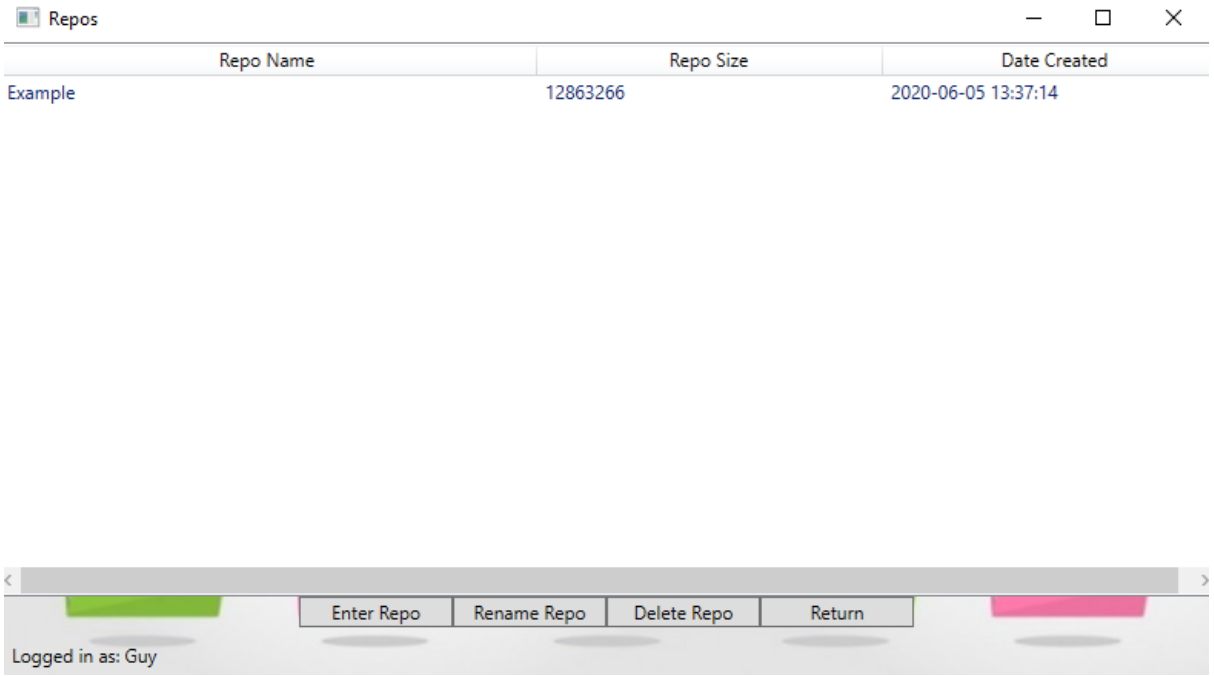
גיא גור-אריה - Repositories

לאחר שהמשתמש נרשם והתוכנה אימתה שאין משתמש קיים כזה, או אם למשתמש היה כבר חשבון קיים והוא נכנס אליו, הוא יתחבר לאפליקציה.

לאחר הכניסה לאפליקציה, יוצג עוד חלון (אשר בתחתיתו יופיע שמו המלא של המשתמש) ובו האפשרויות:

1.לראות את רשימת המאגרים (אקרא להם מעכשיו REPOS או REPO ביחיד) – My Repos list.

בעת לחיצה על הכפתור הזה, יפתח עוד חלון ובוא רשימת הREPOS שהמשתמש יצר:



בחלון הזה, יוצגו כל שמות הREPOS שהמשתמש עצמו יצר, ובנוסף הגודל שלו והתאריך שבו הוא נוצר. לחיצה כפולה על אחד או לחיצה עליהם ואז על כפתור ה-Enter Repo, יפתח את המאגר. ישנם גם האפשרויות לשנות את שם המאגר וגם למחוק את המאגר כולו, אשר מוגן באמצעות סיסמת המשתמש.

2. יצירת REPO.

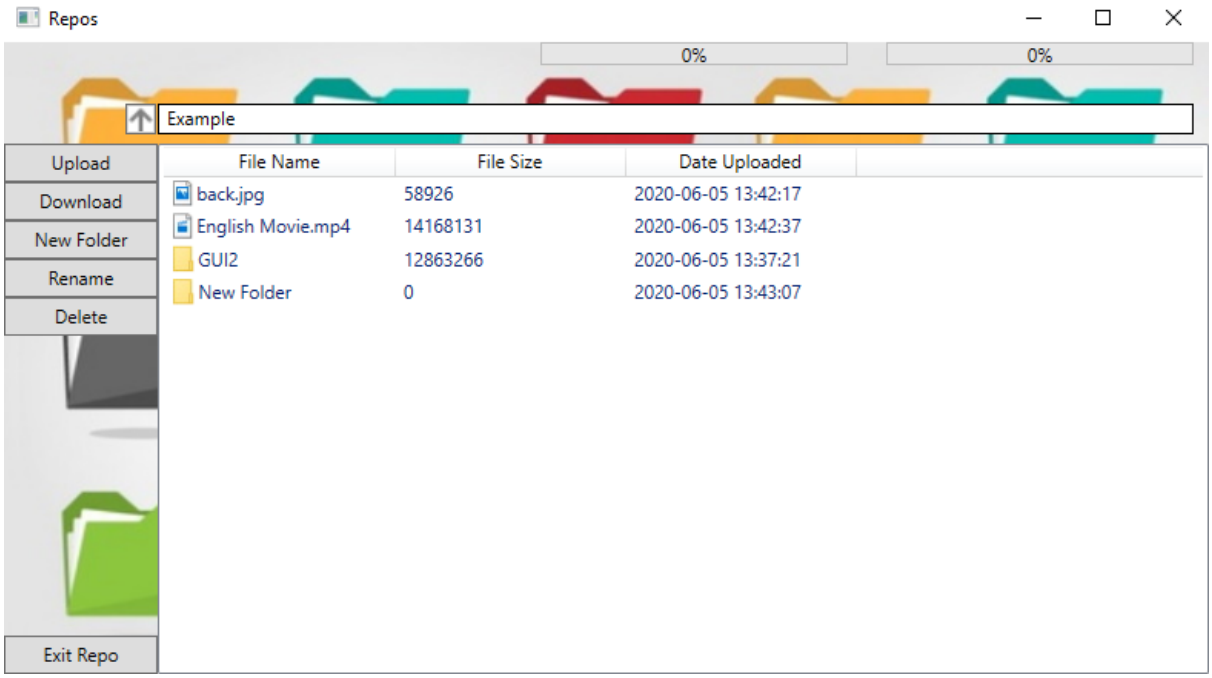
בעת יצירת REPO, כל אשר יתבקש מהמשתמש הוא לתת שם לREPO, אשר ישר יתווסף לרשימה.

3. יציאה מהמשתמש.

כפתור זה מתנתק מהמשתמש הנוכחי מהאפליקציה, לא סוגר אותה. לאחר לחיצה על כפתור זה, יפתח שוב המסך הראשון.

גיא גור-אריה - Repositories

בעת כניסה לREPO:



תחילה יוצגו כל התיקיות וקבצים שנמצאים בREPO. מעליהם יוצג הPATH שהמשתמש נמצא בא כרגע, עם כפתור "חזרה" כדי לצאת מSUBFOLDER שהמשתמש נכנס אליה. מימין למעלה נמצאים ה-Progress Bars, האחד מימין הוא של ההעלאות, והשמאלי הוא של ההורדות.

לחיצה כפולה על תיקייה תפתח אותה (ותעדכן את הPATH למעלה). בחירת קבצים תיתן את האפשרות להוריד אותם או למחוק אותם, ולשנות שם של קובץ. הורדת הקבצים תוריד אותם לתיקיה שנקראת "DOWNLOADS". מחיקת קבצים תפתח חלון השואל אם המשתמש בטוח שהוא אכן רוצה למחוק אותם.

העלאת קבצים תעשה באמצעות לחיצה על הכפתור Upload ובחירת הקובץ או תיקייה, או באמצעות DRAG & DROP של קובץ או תיקייה לPATH המתאים. אם המשתמש בחר להעלות קובץ אשר שמו הוא שם של קובץ של קיים, התוכנה תיצור קובץ חדש ששמו [FILENAME](1). כמובן, אם גם קובץ זה קיים, ייווצר הקובץ (2).

אם המשתמש רוצה ליצור תיקייה חדשה, הוא ילחץ על כפתור הNEW FOLDER, וייתן שם לתיקייה (שם ברירת המחדל הוא NEW FOLDER).

גיא גור-אריה - Repositories

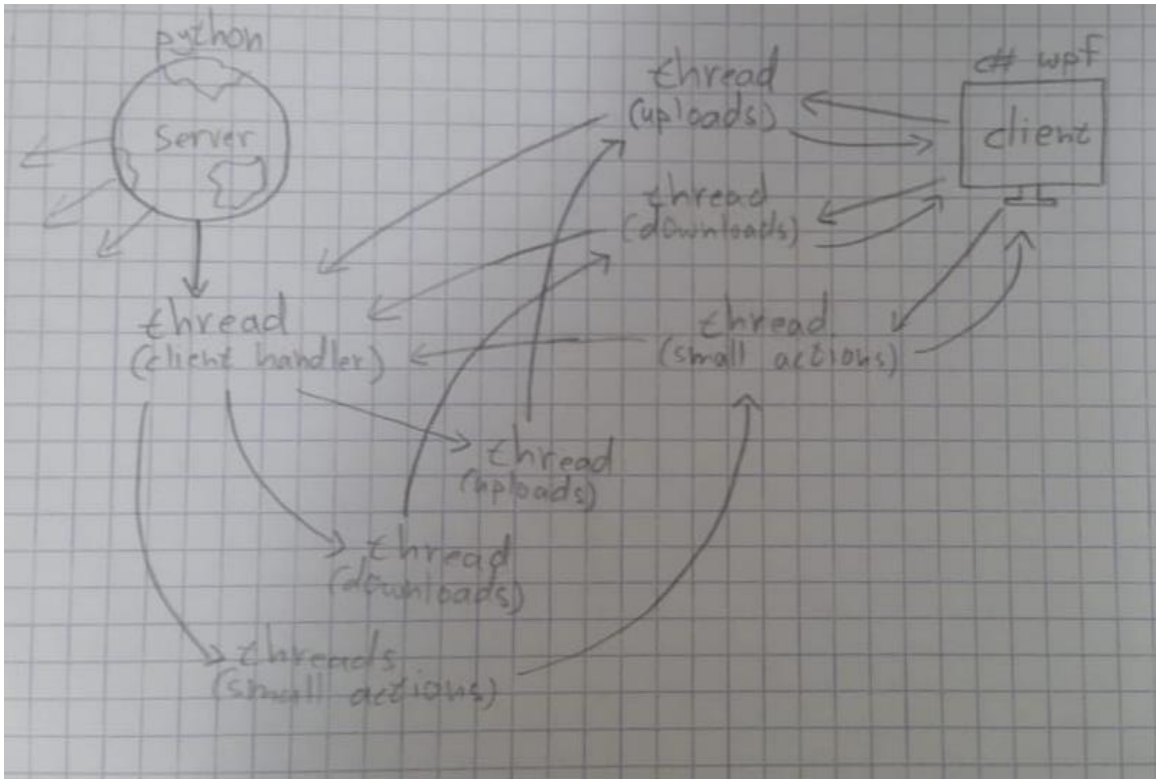
מבנה הפרויקט:

הפרויקט מתחלק לשני חלקים – הצד של השרת והצד של הלקוח.
הלקוח מקבל INPUT מהמשתמש והשרת מקבל INPUT מהלקוח.

כל חלק אחראי על 3 SUBPROCESSES (או threads):

1. העלאות,
2. הורדות,
3. כל השאר.

תרשים:



בצד של הלקוח כאשר המשתמש שולח פקודה כלשהיא – העלאת קובץ, הורדת קובץ, או משהו אחר (כגון הוספת תיקיה ריקה או שינוי שם קובץ), תתווסף הפקודה לתור המתאים – תור העלאות, תור הורדות, או תור "כל השאר".
כך התוכנה תוכל לטפל בהורדות או העלאות לפי תורן, במקביל לפקודות הפשוטות יותר ומבלי להפריע להן.

אותו דבר קורה גם בצד של השרת – כשהוא מקבל פקודה מהשרת, היא מתווספת לתור המתאים ואז הSUBPROCESS המתאים מטפל בה.

לפני כל שליחת "הודעה", יעשה תהליך "Encoding" מסוים שעובד לפי הפרוטוקול שכתבתי, אשר יגיד לשרת\לקוח, האם המסר שישלח הוא פקודה, מידע (למשל בהורדות או העלאות נדרשות כמה הודעות אחת אחרי השנייה), לאיזה תודה הוא מתאים, ולבסוף המידע עצמו.

הפרוטוקול בקיצור:

Write([command/data], queue, [msg1, msg2, msg3....])

כאשר השרת\לקוח מקבל פקודה כזאת, הוא עושה לה Encoding ומחלק את המסרים לתורים המתאימים להם.

גיא גור-אריה - Repositories

אבטחה:

הסיסמאות הנשמרות בDATABASE הן מוצפנות באמצעות מודול הMD5 בC#, קח שהסיסמא האמיתית לא נמצאת בשרת.

מחיקת REPO תיעשה רק באמצעות הכנסת הסיסמא.

קבצים:

Server – תיקיה שבה יש את כל הדברים הקשורים לשרת – האלגוריתם הראשי, ועוד כל מיני קבצים של פונקציות או Classes הקשורים לתקשורת בין השרת:

- Server.py – האלגוריתם הראשי: מקבל משתמש, ויוצר thread שיטפל בו. בקובץ זה גם נוצרת הטבלה של הDATABASE אם היא לא קיימת כבר.
- LogInHandler.py – מטפל בכל מה שקשור להרשמה של המשתמש ופותח את הדף הראשי. לאחר מכן הוא פותח חלון חדש כאשר המשתמש בחר את הREPO שבו הוא יעבוד.
- SERVER – הקובץ שבו יש את האלגוריתם הראשי של הסרבר, רק החלק שבו הוא מקבל משתמש חדש. כל קבצי השרת כתובים בPYTHON.
- RepoHandler.py – הקובץ שבו הדבר הטיפול בREPO קורה – העלאת קבצים, הורדת קבצים, וכו'.
- Function.py – קובץ שבו נמצאות פונקציות אשר השרת משתמש בהן, כגון חישוב גודל של תיקיה, כיווץ קבצים לZIP ועוד.
- Repository – תיקיה נוספת שבה שמורים כל הREPOS של המשתמשים, מסודרים לפי שמות משתמשים (בתיקיה יש עוד תיקיות ששמן הן שמות המשתמשים, ובכל תיקיה מצויים הREPOS של אותו משתמש).

Client – תיקיה שבה יש את כל הדברים הקשורים ללקוח:

- Downloads – תיקיה שבה כל הקבצים המורדים ללקוח ימצאו בה.
- Temp – תיקיה המשומשת רק בתור מקום זמני, בעיקר לכיווץ והוצאה של קבצי ZIP.
- Client – התיקיה שבה יש את האלגוריתם הראשי של הלקוח, אשר כתוב בC# (WPF):
  - Client.sln – הקובץ אשר מריצים כדי להתחיל את הלקוח.
  - Client (כן, תיקיה שקוראים לה CLIENT בתוך תיקיה שרואים לה CLIENT בתוך תיקיה שקוראים לה CLIENT) – תיקיה של כל הclasses והעמודים של השרת:
    - buttons – תיקיה עם כל התמונות של הכפתורים
    - icon – תיקיה עם כל התמונות של האייקונים של הקבצים (תיקיה, קובץ PNG וכו').
    - MainWindow.xaml – הקובץ הראשי שנפתח, ובו נמצא מסך ההרשמה והכניסה למשתמש (הקבצים משתמשים בחלונות שונים שאת שמם לא אפרט כי הם אינם קריטיים).
    - MainMenu.xaml – הקובץ שנפתח לאחר הכניסה למשתמש ובו אפשר לגשת לREPOS.
    - Repository.xaml – הקובץ שבו הלקוח משתמש בREPO.
    - Socket.cs – הCLASS של הלקוח הכתוב לפי הפרוטוקול שכתבתי של התקשורת בין הלקוח לשרת.