**נושאים מתקדמים בתכנות מונחה עצמים ושפת JAVA**

**תרגיל בית 2**

**מגישים: מישל שומילוב – 206415747, שון פורמן - 211420476**

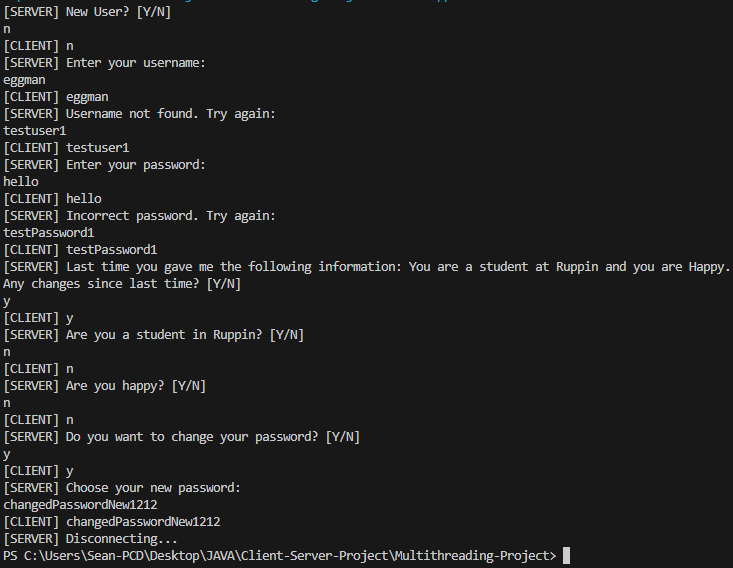
שיחה מצד השרת והלקוח:

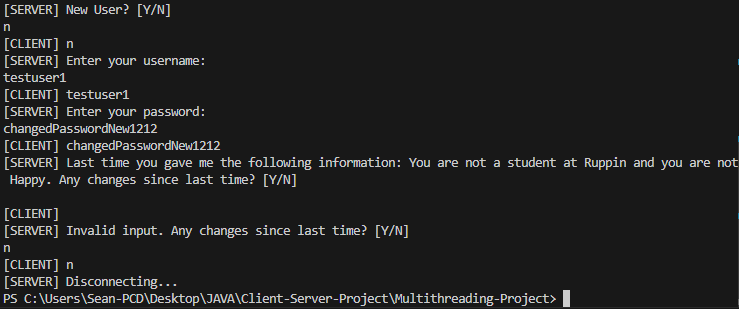
A screen shot of a computer

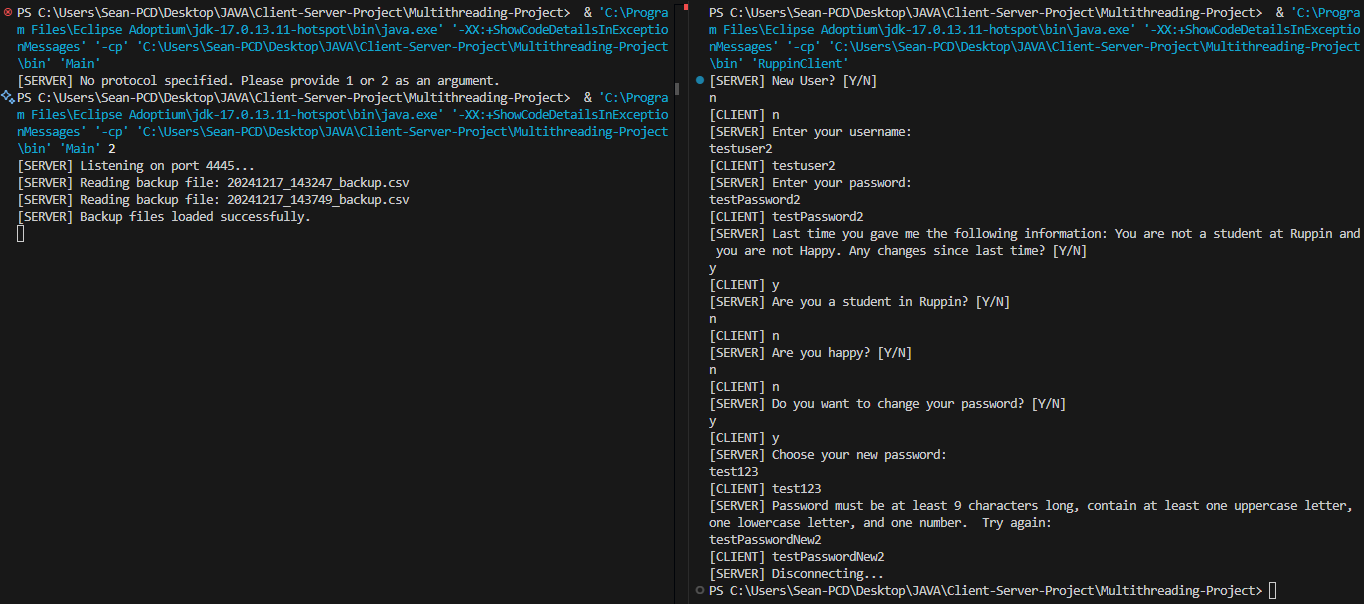
Description automatically generated

A computer screen shot of a black screen

Description automatically generated



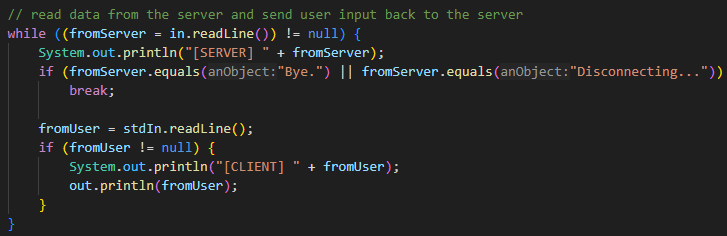




**הסברים:**

בשלב זה השרת מנתק את הלקוח – חשבו איך לעשות את זה?

על מנת לנתק את הלקוח מהשרת, נשתמש בעיבוד (קריאה) של הנתונים מהצד של השרת שמתרחש ב RuppinClient. במקרה וה input מהשרת שווה ל-"Bye." או, במקרה של מה שהוספנו – "Disconnecting…" אז השרת יבין שהוא צריך לנתק את הלקוח וינתק אותו.



יש ליצור מחלקת Client המייצגת לקוח, יש להסביר מה יהיה בה?

ניצור מחלקת Client שיהיו לה התכונות – username, password, isStudent, isHappy

ניצור 2 בנאים, אחד עם כל ארבעת התכונות שמשתמשים בסטים שיצרנו, ובנאי שמקבל רק שם משתמש וסיסמה שאנחנו משתמשים בו ליצירת לקוח זמני ומנצל את מתודת equals בשביל לבדוק אם השם משתמש והסיסמה של הלקוח הזמני שווי ערך לפרטים של אחד מכל שאר הלקוחות שנמצאים ברשימה clientState.

ניצור 4 מתודות גט בשביל לגשת לכל אחת מהתכונות שהן private, ו-4 מתודות סט על מנת לשנות את הערכים תוך כדי זריקת חריגות במקרה הצורך. לא יזרקו חריגות עבור שינוי מצב הלקוח בין אם הוא סטודנט ברופין או לא, ובין אם הוא שמח או לא, מכיוון שלא ניתן בתוכנית בפרוטוקול עצמו להכניס ערך שהוא לא TRUE או FALSE לטרמינל.

לבסוף, יש לנו מתודה שבודקת אם הסיסמה שהמשתמש הכניס עומדת בדרישות שבוקשו במטלה, במקרה ולא תזרק חריגה מהסט של אותה תכונה של הסיסמה;

ויש לנו 2 מתודות בנויות ששינינו שהן מתודת toString להדפסת המידע על הסטודנט, ומתודת equals מכיוון שרצינו לערוך בדיקה אם השם משתמש והסיסמה שהמשתמש הכניס בהתחברות למערכת הם תקינים, וניצלנו יצירת לקוח זמני על מנת לעשות את הבדיקה הזאת עם מתודת ה equals.

מתודות נוספות לפי הצורך?

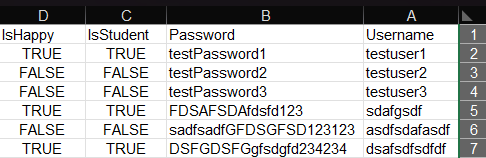
חוץ ממתודות checkUser, checkPassword, addUser, changePassword הוספנו את מתודות findClient שמחפשת לקוח לפי השם משתמש שלו ברשימה clientState, ו-saveUsersToCSV שבוקשה במטלה בשלב מאוחר יותר והיא עובדת ביחד עם addUser ושומרת את הלקוחות לקובץ backup כאשר מספר היוזרים מתחלק ב-3.

**צירוף צילום מסך של התיקייה כאשר יש בה לפחות 2 קבצים (זאת אומרת יצרנו לפחות 6 לקוחות):**

**A screenshot of a computer

Description automatically generated**

**צילום מסך של תוכן מאחד מהקבצים שנוצרו:**

****

**הערה:** בתרגיל התבקש לשמור מידע על הלקוחות רק כאשר הכמות בליסט מתחלקת ב-3. במקרה ואנחנו עובדים באופן פעולה הזה, במקרה ויש לנו לקוח שמור במערכת, והוא משנה את הסיסמה שלו לדוגמה, השינוי לא ישמר בקבצי ה-backup, הוספנו פשוט 2 שורות של שמירה, גם כאשר הלקוח משנה את הסטטוס שלו כסטודנט ובין אם הוא שמח או לא, וכאשר הוא משנה את הסיסמה שלו, **אחרת נאבד מידע קריטי**.

התווספה בנוסף פונקציה של קריאת כל הקבצי backup בשרת של רופין (RuppinServer), לא היה בדיוק ברור מהתרגיל אם צריך לעשות זאת או לא, אבל הפונקציה פשוט עוברת על כל הקבצי backup ומכניסה את הלקוחות לתוך הרשימה של clientState כשהשרת מופעל.

התייחסנו למקרה שהמידע הנכון של לקוח מסוים תמיד ימצא בקובץ הכי אחרון מבחינת התאריך והשעה, ככה שכל פעם שנכנס לקוח לרשימה, במקרה והוא יופיע עוד פעם, הרשימה הקודמת שלו תמחק ותוכנס אחת חדשה במקום.

**הבעיה בשיטה הזאת** היא שאנחנו מסתמכים על כך שהקובץ האחרון שיקרא הוא הקובץ הכי מעודכן ולא בהכרח הקובץ עם הנתונים הכי חדשים. אנחנו לא בטוחים אם ניתן להבטיח זאת רק לפי שמות הקבצים:

