Name: Omer Gerhard ID: 204156384 Email: omergerhard@mail.tau.ac.il

Name: Saar Bergerbest ID: 204540710 Email: bergerbest@mail.tau.ac.il

תיאור הפרוטוקול:

* שליחת הודעה מצד א' לצד ב' ביצענו בעזרת הפונקציה send\_all. הדבר הראשון שמתבצע בפונקציה הוא שליחת 4 בתים המייצגים את אורך ההודעה שעתידה להישלח. לאחר מכן, תישלח ההודעה עצמה.
* קבלת הודעה שנשלחה מצד א' לצד ב' ביצענו בעזרת הפונקציה receive\_all. הדבר הראשון שמתבצע בפונקציה הוא קריאת 4 בתים המייצגים את אורך ההודעה שעתידה להיקלט. לאחר מכן, תיקרא ההודעה עצמה בהתאם.
* השרת שומר את הביקורות השונות שהתקבלו בקובץ בשם "rates.txt" הנשמר בנתיב אותו השרת קיבל כפרמטר.
* מיפוי בין הפקודות לערך שמייצג אותן, אותו הלקוח שולח אל השרת על פי הפירוט למטה:

list\_of\_courses 🡪 "1"

add\_course 🡪 "2"

rate\_course 🡪 "3"

get\_rate 🡪 "4"

quit 🡪 "5"

1. **פקודת list\_of\_courses**
   1. צד לקוח: שולח לשרת את המחרוזת '1' אותה השרת מפרש על פי המיפוי הנ"ל.
   2. צד שרת: מקבל מהלקוח את המחרוזת "1" ומזהה שמדובר בפקודת list\_of\_courses. מתחיל לשלוח אל הלקוח את הקורסים על פי הפורמט 'course\_number course\_name'. כאשר מסיים לשלוח את כל הקורסים שקיימים שולח אל הלקוח את ההודעה "end" שמציינת סיום שליחת רשימת הקורסים מהשרת.
   3. צד לקוח: כל עוד לא קיבל את ההודעה "end" מהשרת, מדפיס כל פעם קורס שקיבל מהשרת על פי הפורמט 'course\_number:<TAB> course\_name' (מקבל כל פעם שורה אחת המכילה את מספר הקורס ושמו מהשרת).
2. **פקודת add\_course**
   1. צד לקוח: שולח לשרת את המחרוזת '2 course\_number course\_name'.
   2. צד שרת: מקבל את המחרוזת הנ"ל מהלקוח, ומזהה לפי מספר הפקודה שמדובר בפקודת add\_course. בודק האם הקורס קיים ברשימת הקורסים (מבנה נתונים שהשרת מחזיק). אם הקורס כבר קיים, השרת שולח ללקוח את ההודעה 'fail' ואחרת מוסיף את הקורס למבנה הנתונים ושולח את ההודעה 'ok' אל הלקוח ששלח את הפקודה. במידה והשרת שלח 'ok' אל הלקוח ששלח את הפקודה, הוא בנוסף שולח לכל **שאר** הלקוחות הפעילים את ההודעה "A new course was just added!".
   3. צד לקוח: מקבל את ההודעה 'fail' או 'ok' מהשרת ומדפיס הודעה בהתאם להנחיות התרגיל.
3. **פקודת rate\_course**
   1. צד לקוח: שולח לשרת את המחרוזת '3 course\_number rate\_grade rate\_text'.
   2. צד שרת: מקבל את המחרוזת הנ"ל מהלקוח, ומזהה לפי מספר הפקודה שמדובר בפקודת rate\_course. בודק האם הקורס קיים ברשימת הקורסים. אם הקורס לא קיים, השרת לא מבצע פעולה (פירוט נוסף מופיע בחלק על מקרי הקצה), אחרת, אם הקורס קיים, השרת כותב לקובץ הביקורות שורה חדשה על פי הפורמט 'course\_number username rate\_grade rate\_text'.
4. **פקודת get\_rate**
   1. צד לקוח: שולח לשרת את המחרוזת '4 course\_number'.
   2. צד שרת: מקבל את המחרוזת הנ"ל מהלקוח, ומזהה לפי מספר הפקודה שמדובר בפקודת get\_rate. השרת פותח את קובץ הביקורות, עובר שורה שורה בקובץ ובודק האם המספר שמופיע בתחילת כל שורה, המייצג את מספר הקורס עליו נתונה הביקורת, תואם למספר הקורס שקיבל מהלקוח. כל שורה מתאימה שמוצא, שולח אל הלקוח כמו שהיא. אחרי שמסיים לעבור על קובץ הביקורות כולו, שולח אל הלקוח את ההודעה 'end' שמטרתה לסמן שאין עוד ביקורות על הקורס המבוקש.
   3. צד לקוח: כל עוד לא קיבל את ההודעה "end" מהשרת, מדפיס כל שורת ביקורת שקיבל מהשרת על פי הפורמט 'username:<TAB>rate\_grade:<TAB>rate\_text'.
5. **פקודת broadcast**
   1. צד לקוח: שולח לשרת את המחרוזת '6 broadcast\_message'.
   2. צד שרת: מקבל את המחרוזת הנ"ל מהלקוח, ומזהה לפי מספר הפקודה שמדובר בפקודת broadcast. השרת עובר על מבנה הנתונים שמחזיק את מספרי ה-socket-ים של הלקוחות הפעילים (כלומר לקוחות שהתחברו בהצלחה) ולכל socket שכזה (שאינו ה-socket של הקשר עם הלקוח אשר שלח את הודעת ה-(broadcast, שולח את הודעת ה-broadcast שהתקבלה.
6. **פקודת quit**
   1. צד לקוח: שולח לשרת את המחרוזת '5'.
   2. צד שרת: מקבל את המחרוזת הנ"ל מהלקוח, ומזהה לפי מספר הפקודה שמדובר בפקודת quit. מסיים את התקשורת עם אותו הלקוח.

מבנה התכניות:

ישנם 3 מודולים – server, client, utils. ה-server וה-client מכילים את הקוד הרלוונטי לכל אחד מהצדדים בהתאם. Utils מכיל פונקציות וערכים המשמשים את שני המודולים האחרים.

הערות כלליות:

1. בעת ניסיון התחברות בעזרת שם משתמש וסיסמה בצד הלקוח. למשל עם שם המשתמש Bob והסיסמא simplepass. הפורמט הנכון לבצע את ההתחברות הוא:

א. יש לכתוב בצד הלקוח: User:<space>Bob

ב. להקיש על Enter.

ג. לאחר מכן לכתוב: Password:<space>simplepass

ד. לבסוף להקיש שוב על Enter.

2. כל לקוח זמין לקבל הודעת push מהשרת בכל רגע נתון ומדפיס את ההודעה במידה וקיבל כזאת.

מקרי קצה:

1. אם הלקוח שולח פקודת rate\_course על מנת לדרג קורס **שאינו קיים ברשימת הקורסים**, השרת לא כותב את הביקורת לקובץ הביקורות כיוון שמזהה שהקורס אינו קיים ברשימת הקורסים. כמו כן, הלקוח לא יקבל אינדיקציה על כך שהביקורת שלו לא נקלטה, כיוון שהחלטנו לא לזהות מצב כזה כשגיאה.
2. אם הלקוח שולח פקודת get\_rate על מנת לקבל ביקורות על **קורס שאינו קיים ברשימת הקורסים,** השרת יזהה שהקורס המבוקש לא קיים ברשימת הקורסים ופשוט יישלח הודעת "end" אל הלקוח כאילו סיים להעביר לו את הביקורות הרלוונטיות (הלקוח מצפה להודעת "end" בעת סיום שליחת הביקורות מהשרת. לכן, למעשה לא יודפס כלום בצד לקוח, והלקוח לא יקבל אינדיקציה על כך שהקורס עליו ביקש דירוגים לא קיים כלל ברשימת הקורסים.
3. אם הלקוח שולח פקודת add\_course עם שם הקורס ריק, למשל add\_course 0 "", החלטנו במקרה זה לא להדפיס שגיאה אל הלקוח אלא להחשיב פקודה זו כפקודה חוקית, והקורס יתווסף לרשימת הקורסים עם שם ריק.
4. אם הלקוח שולח פקודת rate\_course עם ביקורת "מילולית" ריקה,

למשל rate\_course 0 50 "", החלטנו במקרה זה לא להדפיס שגיאה אל הלקוח אלא להחשיב פקודה זו כפקודה חוקית, והביקורת תתווסף לקובץ הביקורות עם תיאור מילולי ריק.

1. בכל פקודת לקוח שאחד הפרמטרים שלה הוא טקסט שנמצא בין גרשיים, אנו לא מאפשרים מצב בו הטקסט מכיל את התו ' " '. במקרה ופקודה שכזו נשלחת, ההתנהגות היא לא צפויה.