

FINAL PROJECT – DESIGN PAPER

יובל קרליבך 318850971 תומר כלתי 318366168

מטרת הפרויקט:

הפרויקט שלנו הוא Multi-Room Chat System המאפשרת למשתמשים לתקשר זה עם זה דרך הרשת בזמן אמת. המערכת תומכת ביצירת מספר חדרי צ'אט במקביל, כאשר כל חדר מאפשר לקבוצת משתמשים לשוחח ביניהם. המשתמשים יכולים להצטרף לחדרים קיימים, ליצור חדשים, לשלוח הודעות קבוצתיות לכל חברי החדר, ולשלוח הודעות פרטיות למשתמש ספציפי.

השרת מנהל את כל חדרי הצ'אט ומתאם בין הקליינטים השונים, כאשר כל חדר צ'אט משתמש ב-multicast group נפרד לשידור הודעות. הקליינטים מתחברים לשרת דרך TCP לפקודות בקרה ומקבלים הודעות דרך multicast. המערכת כוללת תמיכה בכמה עשרות משתמשים בו-זמנית עם יכולות ניהול חדרים כמו הרשאות, רשימת משתמשים מחוברים, והתראות על כניסה ויציאה של משתמשים.

המטרה היא ליצור פלטפורמת תקשורת יעילה ויציבה המדמה מערכות צ'אט מודרניות כמו Discord או Slack, תוך מימוש עקרונות תכנות רשתות מתקדמים.

ניהול מקביל של משתמשים:

השרת ישתמש בגישה היברידית המשלבת select() כגישה ראשית עם threads נוספים לפעולות מיוחדות. השרת מנהל main thread אחד עם select() שעוקב אחר כל הסוקטים של המשתמשים המחוברים (welcome socket + כל המתמששים הפעילים) ומגיב מיד לאירועים. כאשר מתקבלת הודעה רגילה, השרת מעבד אותה מיד ושולח multicast לחדר הרלוונטי. לפעולות כבדות כמו file transfer או room management, השרת יוצר threads נפרדים כדי לא לחסום את הטיפול בקליינטים אחרים.

בצד הקליינט, כל משתמש ירוץ עם שני threads נפרדים: sender thread שחוסם על user input ושולח הודעות לשרת דרך TCP, ו-receiver thread שמקשיב להודעות multicast מהחדר הנוכחי ולהודעות מהשרת. כל חדר צ'אט משתמש ב-multicast group נפרד (כתובות שונות), כך שהודעות מגיעות רק למשתמשים בחדר הרלוונטי. הגישה הזו מבטיחה יעילות בשרת (select מנהל עשרות קליינטים ביחד) ותגובה מהירה אצל המשתמשים (שליחה וקבלה לא חוסמות זו את זו).

סוגי סוקטים:

המערכת תשתמש בשלושה סוגי סוקטים עיקריים. השרת יחזיק TCP Welcome Socket לקבלת חיבורים חדשים מקליינטים, וכאשר קליינט מתחבר ייוצר חיבור TCP ייעודי לתקשורת עם אותו קליינט. החיבורים האלה ישמשו לפקודות בקרה כמו join room, authentication, הודעות פרטיות וקבלת confirmations מהשרת. בנוסף השרת יחזיק UDP Multicast Sender Socket לשידור הודעות לחדרי הצ'אט, כאשר כל חדר ישתמש בכתובת multicast שונה כך שהודעות מגיעות רק לחברי החדר הרלוונטי.

כל קליינט יחזיק את קצה החיבור TCP שלו עם השרת לשליחת פקודות וקבלת תגובות, ובנוסף UDP Multicast Receiver Socket לקבלת הודעות מהחדר הנוכחי. הקליינט יצטרך לבצע bind ולהצטרף לקבוצת המולטיקאסט של החדר כדי לקבל הודעות. החלוקה הזו מבטיחה אמינות לפקודות חשובות דרך TCP ומהירות ויעילות להודעות הצ'אט דרך multicast. כאשר משתמש שולח הודעה היא עוברת דרך TCP לשרת, והשרת משדר אותה דרך multicast לכל הקליינטים בחדר.

סוגי פרוטוקלים:

המערכת תשתמש בשלושה פרוטוקולי רשת עיקריים. TCP Protocol יישמש לכל התקשורת הקריטית הדורשת אמינות כמו confirmations authentication, room management, private messages כדי להבטיח שפקודות חשובות מגיעות בסדר הנכון וללא אובדן. UDP Protocol יהווה את הבסיס ל Multicast עבור הודעות הצ'אט הרגילות בחדרים, כאשר הרשת תשתמש ב-IGMP לניהול החברויות בקבוצות מולטיקאסט ו-PIM routing בין נתבים.

ברמת האפליקציה, הפרוטוקול יכלול סוגי הודעות מוגדרים כמו LOGIN, CREATE_ROOM, JOIN_ROOM, SEND_MESSAGE ו-PRIVATE_MESSAGE עם מבנה אחיד של type, length ו-data fields. הפרוטוקול יכלול מספר timeouts קריטיים: keep-alive interval של 30 שניות למניעת ניתוק, response timeout של 10 שניות לפקודות TCP, connection timeout של 5 שניות להתחברות ראשונית, ו-session timeout של שעה לניתוק אוטומטי של משתמשים לא פעילים. המערכת תטפל בשגיאות תקשורת כמו הודעות פגומות או אובדן קישור באמצעות error handling ו-automatic reconnection.

בדיקות QA של הפרוייקט:

- 1. בדיקת עומס** - חיבור 15-20 קליינטים בו-זמנית עם שליחת הודעות מרובות מכל הקליינטים למדידת זמני תגובה ויציבות המערכת תחת לחץ ובדיקת יכולת הטיפול במספר חדרים פעילים במקביל.
- 2. בדיקת תקלות** - בחינת התאוששות מניתוק פתאומי של קליינטים ושרת, כולל ניתוק כפוי של חיבורים, הפסקת רשת באמצע פעילות ובדיקת יכולת התחברות מחדש אוטומטית.
- 3. בדיקת הודעות פגומות** - שליחת packets עם מבנה שגוי, אורכים לא תקינים, סוגי הודעות לא חוקיים ונתונים חלקיים כדי לוודא שהמערכת מטפלת בשגיאות ללא קריסות ומחזירה error messages מתאימים.
- 4. בדיקת timeouts** - בחינת מנגנוני ה-keep-alive, response timeout ו-session timeout על ידי הפסקת תקשורת מקליינטים, אי-מענה להודעות שרת ובדיקת ניתוק אוטומטי לאחר תום זמני ההמתנה המוגדרים.
- 5. בדיקה פונקציונלית ומקרי קצה** - בדיקת כל התכונות כמו authentication, יצירת חדרים, הצטרפות, שליחת הודעות רגילות ופרטיות, ובחינת מקרי קצה כמו הצטרפות לחדר לא קיים, שני משתמשים עם שם זהה ושליחת הודעות לאחר יציאה מחדר.

