

**Chat app – שחר הקר**

תיכון הדרים הוד השרון

ת"ז: 212726673

שם המנחה: ניר סליקטר

תאריך הגשה: 9.5.2020

**מבוא**

הפרויקט שלי הוא chat app, אפליקציה שמאפשרת לך להתכתב עם חבריך, ולשמור את היסטוריית השיחות שלכם באמצעות בסיס נתונים. בנוסף לכך, האפליקציה מאפשרת לך לבחור ערוץ, שבו אתה יכול לדבר רק עם מי שנמצא בערוץ זה, וההסיטוריה הספציפית של הערוץ נשמרת. כמו כן האפליקציה מאפשרת לך למחוק את ההודעה האחרונה ששלחת בערוץ, ולהחליף ערוץ באמצע.

**בספר זה מספר פרקים:**

**ארכיטקטורת הפרויקט:** פרק זה מכיל הסבר על הפרויקט, על המצבים של השרת והלקוחות, כמו גם api של הפעולות והclasses שהשתמשתי בהם. בנוסף לכך, פרק זה מכיל הסבר קצר על בסיס הנתונים שהשתמשתי בו, שיפורט יותר בהמשך.

**מדרך למשתמש:** פרק זה מכיל הסבר למשתמש, מה הפקודות השונות שניתן להכניס לאפליקציה, ומה הדרישות הבסיסיות בשביל שיהיה אפשר להשתמש בה. בנוסף לכך, מוצגים מסכים שונים הממחישים את הפקודות השונות.

**בסיס נתונים:** פרק זה מכיל הסבר מפורט יותר על בסיס הנתונים שהשתמשתי בו, וכיצד ההודעות נשמרות, וכיצד מבדילים בין הערוצים השונים.

**מדריך למפתח:** פרק זה נועד להסביר למפתח הבא של הקוד כיצד להמשיך. הוא מכיל הסבר מפורט יותר על הפונקציות השונות כמו גם צילומים של הקוד.

**רקע לפרויקט:** הרקע לפרויקט שלי הווא העובדה שיש מקרים רבים בהם אני מתכתב עם חברים שלי, ותמיד תהיתי כיצד בסיסי הנתונים שבהם ההודעות נשמרות עובדים, ולכן רציתי להתחיל לעבוד בעצמי עם בסיסי נתונים בשביל לראות כיצד זזה עובד.

**תהליך המחקר:**

חיפוש באינטרנט ובgit hub

המצב הקיים בשוק: ישנן אפליקציות כמו whatsapp ןskype, שעושות אותו דבר שהפרויקט שלי עושה ויותר

חידושים: חידושים עם עבודה עם בסיס נתונים כמו גם עבודה עם אודיו.

**אתגרים מרכזיים:**

הבעיה איתה התמודדתי בתהליך העבודה הוא הלימוד של העבודה עם בסיס הנתונים. נושא זה היה חדש לי יחסית, ולכן הייתי צריך ללמוד הכל כמעט מאפס. בחרתי את הנושא הזה בגלל שהוא נראה לי מעניין, ותמיד התעניינתי בעבודה עם בסיס נתונים. בנוסף לכך, אני יודע שעבודה עם בסיסי נתונים מאוד חשובה לטכנולוגיה של ימינו, וללמוד משהו כזה נראה לי חשוב. הפרויקט עונה על הצורך לדבר עם אנשים, ולהיות מסוגל לראות על מה דיברתם קודם לכן, למקרה שכתבתם אחד לשני דברים חשובים, או שסתם אתם רוצים להמשיך את השיחה מאיפה שעצרתם

**פתרונות לבעיה:** פתרתי את הבעיה על ידי כך ששאלתי חברים שלי שגם היו צריכים לעבוד עם בסיסי נתונים, כמו גם את אבא שלי, שיש לו ניסיון רב בעבודה עם בסיסי נתונים.

**ארכיטקטורת הפרויקט**

**תיאור הפרויקט:** פרויקט chat app מאפשר לאנשים לדבר אחד עם השני, בערוצים ספציפיים שהם רוצים. בנוסף לכך, הפרויקט מאפשר לשמור את ההיסטוריה של השיחה בערוץ מסויים, וכל פעם שמישהו מתחבר לערוץ האפליקציה שולחת לו את ההיסטוריה של ההשיחה. כמו כן, האפליקציה מאפשרת למשתמש למחוק את ההודעה האחרונה שלו בערוץ, ולהחליף ערוצים באמצע השיחה.

הפתרון המוצע בפרויקט זה הוא כתיבת שרת ולקוח באמצעות פייתון, ושמירת ההודעות באמצעות בסיס הנתונים SQLite. פתרון זה נבחר מאחר שלפייתון יש ספרייה מובנית של SQLite.

**קבצי הפרויקט, והפעולות שבהם:**

## Applications

### Client

uiCallback(message)

chatCommunicationCallbcack(line)

#### Main

#### Class UILayer

send(event = none)

on\_closing(event = none)

setChannel()

setMessage()

start()

#### Class chatCommunication

send(message)

receive()

### Server

accept\_incoming\_connections()

handle\_client(client)

broadcast(message, channel, prefix = "")

quitUser(client, channel, name)

deleteMessage(channel, user)

changeChannel(oldChannel, newChannel, client, name)

sendHistory(channel, client)

formatMessage(name, message)

#### Main

#### Class HistoryStore

storeChannelMessage(channel, message, user)

readChannelHistory(channel)

connect()

deleteLastMessage(channel, user)

#### Class ChatManager

addToChannel(channel, name, client)

removeFromChannel(channel, client)

getChannel(channel)

## Data Store

### SQLLite

#### ChannelHistory

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Comment** |
| Channel | String | Channel name |
| InsertDate | DateTine | The date and time the message was added. |
| Message | String | A specific message |
| User | String | The name of the user |
|  |  |  |
|  |  |  |
|  |  |  |

יש שתי תיקיות בתוך הפרויקט. בתוך תיקיית הclient נמצאים הקבצים client.py, uilayer.py, וchatcommunication.py uilayer אחראי על מה שהלקוח רואה, בעוד chatcommunication אחראי על התקשורת בין בלקוח לבין השרת, והclient.py אחראי על החיבור בין שני הclasses. בתיקיית הserver נמצאים הקבצים server.py, historystore.py, וchatmanager.py. historystore אחראי על החיבור בין השרת לבסיס הנתונים, וchatmanager אחראי על רשימת המשתמשים שבתוך הערוץ, ועל להוציא ולהכניס משתמשים מהערוץ. Server.py אחראי על התקשורת בין השרת ללקוח, והוא משתמש בשני הclasses.

**ארכיטקטורת הרשת:**

Message2

Message1

Message1

Message2

messages

database

server

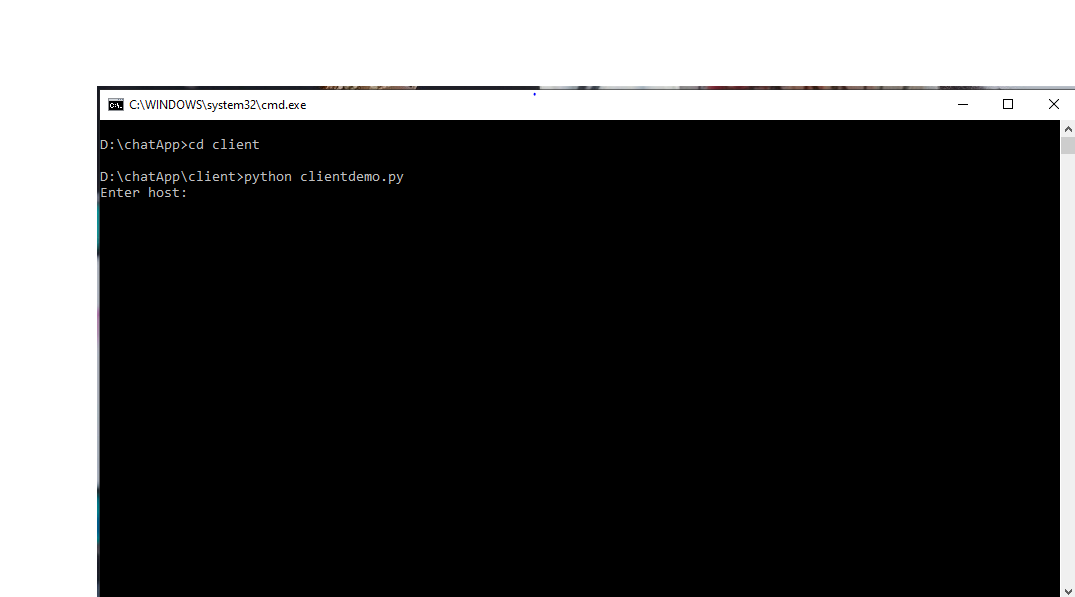
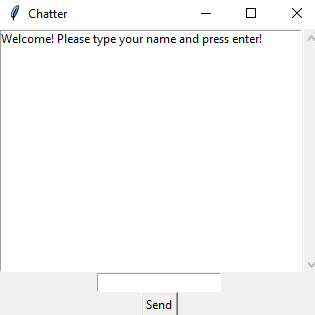
Client2

Client1

ארכיטקטורת הרשת מבוססת על חיבור בין שרת ללקוחות באמצעות sockets, וחיבור בין השרת לבסיס הנתונים באמצעות הספרייה הטבעית בפייתון לעבודה עם SQLite3.

**מדריך למשתמש**

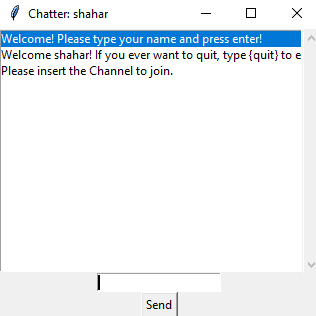
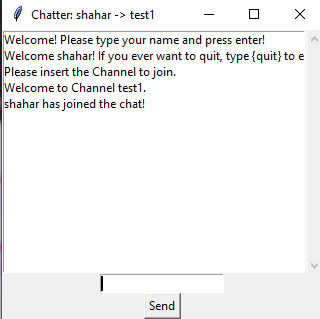
**קבצים שצריך להתקין:** [sqlite-tools-win32-x86-3310100.zip](https://www.sqlite.org/2020/sqlite-tools-win32-x86-3310100.zip) בנוסף לכך צריך את הdllים של sqlite 3.

**תרשים מסכים:** 

האפליקציה מבקשת ממך לכתוב את השם שלך.

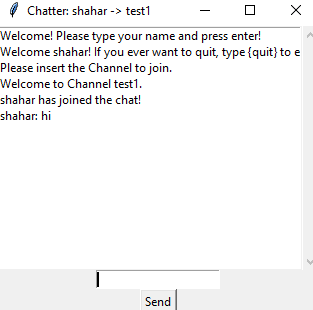
פותחים את הלקוח והוא שואל אותך לאיזה שרת אתה רוצה להתחבר.

אתה רושם את הip של השרת שאתה רוצה להתחבר אליו.

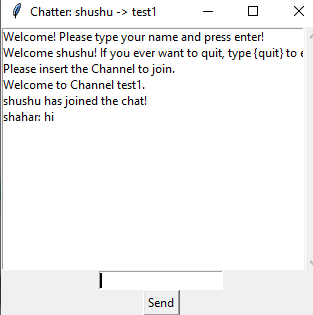


כאשר אתה כותב את הערוץ אליו אתה רוצה להכנס האפליקציה משנה את הכותרת למעלה ומוסיפה גם את שם הערוץ, ורושמת את ההודעה "welcome to channel" ומוסיפה את שם הערוץ. כמו כן האפליקציה שולחת הודעה לכל המשתתפים בערוץ על הבן אדם שהצטרף

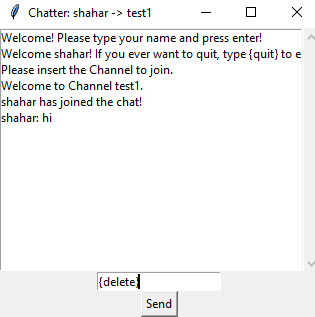
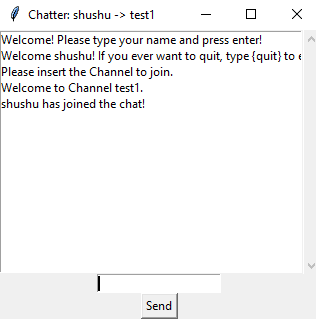
לאחר שאתה מכניס את השם שלך האפליקציה משנה את הכותרת למעלה ומוסיפה את השם שלך, ומבקשת ממך להכניס את הערוץ אליו אתה רוצה להתחבר



כאשר אתה כותב הודעה ההודעה נשלחת לכל האנשים המחוברים לערוץ באותו הרגע

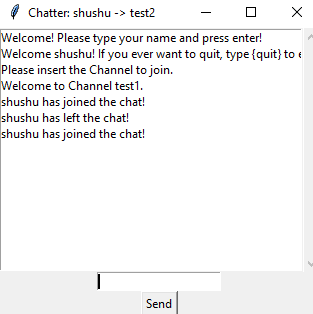


כאשר מישהו מצטרף לערוץ ובערוץ נשלחו כבר הודעות, האפליקציה שולחת לו את ההיסטוריה של הערוץ



ופעם הבאה זה לא יראה אותה בהיסטוריה.

כאשר אתה שולח {delete}, האפליקציה תמחק את ההודעה האחרונה ששלחת בערוץ הזה...



כאשר אתה רושם {channel} ומוסיף אחר כך את שם הערוץ אליו אתה רוצה לעבור, האפליקציה מוציאה אותך מהערוץ הקודם ומעבירה אותך לאחד חדש, ומעדכנת את בכותרת למעלה בהתאם.

**מגבלות הפרויקט:**

מספר האנשים שיכולים להיות מחוברים בו זמנית הוא 5 אנשים.

אין דרך למחוק הודעה ספציפית.

ההסיסטוריה של הערוץ נשלחת לכולם לא משנה אם זה פעם ראשונה שלהם בערוץ או לא.

אין דרך לחסום משתמש.

שם של ערות לא יכול להיות עם רווח.

**בסיס הנתונים**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Comment** |
| Channel | String | Channel name |
| InsertDate | DateTine | The date and time the message was added. |
| Message | String | A specific message |
| User | String | The name of the user |
|  |  |  |
|  |  |  |
|  |  |  |

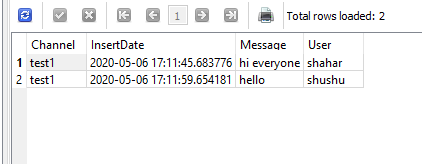
בסיס הנתונים שהשתמשתי בו הו SQLite3.

בטבלה ארבע שדות. השדה הראשון הוא הערוץ אליו נשלחה ההודעה, בשביל להבדיל לשם כתיבת ההיסטוירה ולאיזה משתמשים ההודעה תשלח.

השדה שני הוא התאריך והשעה שבו ההודעה נשלחה.

השדה השלישי הוא ההודעה עצמה.

השדה האחרון הוא המשתמש ששלח את ההודעה.



**מדריך למפתח**

Message2

Message1

Message1

Message2

messages

database

server

Client2

Client1

**פעולות**

**server.py**

**נמצא בתיקיית server שבפרויקט.**

בקובץ זה נמצא השרת, שאחראי על העברת ההודעות בין המשתמשים בערוץ, הפקודות שהמשתמשים שולחים, והעברת ההודעות לבסיס הנתונים

**def accept\_incoming\_connections():**

מטרת הפונקציה לקבל לקוחות שמנסים להתחבר

**def handle\_client(client):**

מטרת הפונקציה לטפל בשליחת ההודעות והפקודות שהלקוח שולח.

**def broadcast(msg, channel):**

מטרת הפונקציה לשלוח לכלל האנשים הנמצאים בערוץ את ההודעה שנלחה

**def quitUser(client,channel,name)**

מטרת הפונקציה להוציא את המשתמש מרשימת המשתמשים ושלוח הודע הלכולם על כך שהוא עזב

**def deleteMessage(channel, user)**

מטרת הפונקציה למחוק את ההודעה האחרונה של המשתמש בערוץ המסוים.

**def changeChannel(oldChannel, newChannel, client, name)**

מטרת הפונקציה לשנות את הערוץ בו המשתמש נמצא מהערוץ הישן לחדש.

**def sendHistory(channel, client)**

מטרת הפונקציה לשלוח את ההיסטוריה של הערוץ לכל המשתתפים בו.

**def fomatMessage(name, msg)**

מטרת הפונקציה לעצב את ההודעה בצורה בה אנו רוצים שהיא תישלח.

**historystore.py**

**נמצא בתיקיית server שבפרויקט.**

בקובץ זה נמצא הclass שאחראי על הטיפול בבסיס הנתונים וההיסטוריה.

**def \_\_connect()**

מחזיר חיבור לSQlite database

**def readChannelHistory(channel)**

מחזיר את ההיסטוריה של הערוץ אותו הוא מקבל.

**def storeChannelMessage(channel, message, user)**

מעביר את ההודעה מהמשתמש להיסטוריה בצירוץ הערוץ בו היא נשלחה.

**def deleteLastMessage(channel, user)**

מוחק את ההודעה האחרונה של המשתמש בערוץ.

**chatmanager**

**נמצא בתיקיית server שבפרויקט.**

בקובץ הזה נמצא הclass שאחראי על ניהול הערוצים והמשתמשים שבהם

**def addToChannel(channel, name, client)**

מטרת הפונקציה להוסיף משתמש לערוץ מסוים.

**def removeFromChannel(channel, name)**

מטרת הפונקציה להסיר משתמש מערוץ מסוים.

**uilayer.py**

**נמצא בתיקיית client שבפרויקט**

בקובץ זה נמצא הclass שאחראי על הui

**def start()**

מתחיל את הgui

**def \_\_send()**

מטרת הפונקציה לשלוח הודעה ולהציג אותה

**def \_\_on\_closing()**

שולח quit כשהמשתמש רוצה לצאת.

**def \_\_setChannel()**

משנה את הכותרת בהתאם לערוץ

**def setMessage(msg)**

משנה את ההודעה להודעה הנוכחית

**chatcommunication.py**

**נמצא בתיקיית client שבפרויקט**

בקובץ זה נמצא הclass שאחראי לטיפול בתקשורת בין השרת ללקוח מהצד של הלקוח

**def \_\_receive()**

הפונקציה משמשת לקבלת הודעות.

**def send(msg)**

הפונקציה משמשת לשליחת הודעות.

**client.py**

**נמצא בתיקיית client שבפרויקט**

בקובץ נמצא החיבור בין הuilayer לבין הchatcommunication, כמו גם תחילת הפעולה של הלקוח

**def uiCallback(msg)**

מטרת הפונקציה לחבר בין שכבת הui לבין שכבת הchatcommunication

**Def chatCommunicationCallback(line)**

מטרת הפונקציה לחבר בין שכבת הchatcommunication לבין שכבת הui

**משתנים**

**server**

**Server.py**

Name השם של המשתמש

PORTהפורט בו משתמשים להעביר הודעות

BUFSIZ גודל ההודעה שניתן להעביר

ADDR הכתובת של השרת

SERVER הsocket שהשרת משתמש בה

\_chatManager הclass האחראי על ניהול המשתמשים

\_historyStore הclass האחראי על החיבור לבסיס הנתונים

Client הsocket של הלקוח

Channel הערוץ שהלקוח רוצה להצטרף אליו

Msg ההודעה שהלקוח שלח

**historystore.py**

\_fileName השם של הקובץ של בסיס הנתונים

\_createTable פקודת הSQlite3המשמשת ליצירת טבלה חדשה

\_insertMessage פקודת הSQlite3 המשמשת להכנסת הודעה לטבלה

\_readHistory משמש לקבל את ההיסטוריה של הערוץ מבסיס הנתונים

\_readLastMessage פקודת הSQlite3 המשמשת לקריאת ההודעה האחרונה של המשתמש בערוץ

\_deleteMessage פקודת הSQLite3 המשמשתלמחיקת הודעה

**chatmanager.py**

\_channels{} רשימה שמכילה את הערוצים הקיימים

**Client**

**Client.py**

Msg הודעה שנשלחת על ידי הלקוח

uilayer המשתנה של שכבת הuiשל הלקוח

chatcommunication המשתנה שאחראי על החיבור בין הלקוח לשרת