**MushApp**

שם בית ספר - בית ספר ויצ"ו הדסים

שם התלמיד - אריאל שרגא שיפר

תעודת זהות – 329090195

שם המנחה – ניר דוויק

תאריך ההגשה – 31.5.2023

Table of Contents

[**1: מבוא** 3](#_Toc136484665)

[**1.1 תכולת הספר** 3](#_Toc136484666)

[**1.2 הרקע לפרויקט** 3](#_Toc136484667)

[**1.3 הסיבות לבחירת הנושא** 3](#_Toc136484668)

[**1.4 תהליך מחקר הפרויקט** 3](#_Toc136484669)

[**1.5 המצב הקיים בשוק** 4](#_Toc136484670)

[**1.6 אתגרים מרכזיים והפתרונות שלי** 4](#_Toc136484671)

[2: מבנה הפרויקט 5](#_Toc136484672)

[**2:1 הלקוח – הממשק החיצוני** 6](#_Toc136484673)

[**2:1:1 Home Page** 7](#_Toc136484674)

[**2:1:2 Log In Page and Sign Up page** 7](#_Toc136484675)

[**2:1:3 Join Party Page** 8](#_Toc136484676)

[**2:1:4 After Enter Page** 9](#_Toc136484677)

[**2:1:5 Party For Client** 10](#_Toc136484678)

[**2:1:6 Party For Leader** 11](#_Toc136484679)

[**2:2 השרת** 12](#_Toc136484680)

[**2:2:1 main()** 13](#_Toc136484681)

[**2:2:2 create\_http\_respond()** 13](#_Toc136484682)

[**2:2:3 process\_client()** 13](#_Toc136484683)

[**2:2:4 validate\_http\_request()** 13](#_Toc136484684)

[**2:2:5 check\_in\_db()** 13](#_Toc136484685)

[**2:2:6 add\_to\_db()** 13](#_Toc136484686)

[**2:2:7 hash\_string()** 14](#_Toc136484687)

[**2:2:8 generate\_new\_party()** 14](#_Toc136484688)

[**2:2:9 party\_thread()** 14](#_Toc136484689)

[**2:2:10 get\_song\_data()** 14](#_Toc136484690)

[**2:2:11 sender()** 14](#_Toc136484691)

[**3: מדריך למשתמש** 15](#_Toc136484692)

[**3.1 סביבת עבודה** 15](#_Toc136484693)

[**3:2 הפעלת פרויקט** 15](#_Toc136484694)

[**4: בסיס נתונים** 19](#_Toc136484695)

[5: מדריך למפתח 20](#_Toc136484696)

[**GuiApp** 20](#_Toc136484697)

[**5:1 ClientCommunication()** 20](#_Toc136484698)

[**5:1:1 innit** 20](#_Toc136484699)

[**5:1:2 connect** 20](#_Toc136484700)

[**5:1:3 send\_data** 21](#_Toc136484701)

[**5:1:3 receiving data** 21](#_Toc136484702)

[**5:1:4 close** 21](#_Toc136484703)

[**5:2 create\_http\_request** 21](#_Toc136484704)

[**5:3 handle\_http\_respond** 21](#_Toc136484705)

[**5:4 HomePage** 21](#_Toc136484706)

[**5:4:1 open\_join\_party\_func** 21](#_Toc136484707)

[**5:4:2 open\_sign\_in\_func** 21](#_Toc136484708)

[**5:4:3 open\_log\_in\_func** 21](#_Toc136484709)

[**5:5 JoinPartyPage** 21](#_Toc136484710)

[**5:5:1 party\_enter** 21](#_Toc136484711)

[**5:6 LogClientInfo** 22](#_Toc136484712)

[**5:6:1 log\_enter** 22](#_Toc136484713)

[**5:7 SignClientInfo** 22](#_Toc136484714)

[**5:7:1sign\_enter** 22](#_Toc136484715)

[**5:8 AfterEnterPage** 22](#_Toc136484716)

[**5:8:1 start** 22](#_Toc136484717)

[**5:9 PartyPage1** 22](#_Toc136484718)

[**5:9:1 play\_song** 22](#_Toc136484719)

[**5:10 PartyPage2** 22](#_Toc136484720)

[**ServerMushApp** 23](#_Toc136484721)

[**5:11 main()** 23](#_Toc136484722)

[**5:12 create\_http\_respond()** 23](#_Toc136484723)

[**5:13 process\_client()** 23](#_Toc136484724)

[**5:14 validate\_http\_request()** 23](#_Toc136484725)

[**5:15 check\_in\_db()** 23](#_Toc136484726)

[**5:16 add\_to\_db()** 24](#_Toc136484727)

[**5:17 hash\_string()** 24](#_Toc136484728)

[**5:18 generate\_new\_party()** 24](#_Toc136484729)

[**5:19 party\_thread()** 24](#_Toc136484730)

[**5:20 get\_song\_data()** 24](#_Toc136484731)

[**5:21 sender()** 24](#_Toc136484732)

[**Party** 24](#_Toc136484733)

[**5:22 Party** 24](#_Toc136484734)

[**5:22:1 innit** 25](#_Toc136484735)

[**5:22:2 add\_guest** 25](#_Toc136484736)

[**5:22:3 get\_guests** 25](#_Toc136484737)

[**5:22:4 get\_code** 25](#_Toc136484738)

[**5:22:5 set\_mode** 25](#_Toc136484739)

[**5:22:6 get\_mode** 25](#_Toc136484740)

[**5:22:7 get\_leader** 25](#_Toc136484741)

[**5:22:8 set\_started** 25](#_Toc136484742)

[**5:22:9 get\_started** 25](#_Toc136484743)

[**6: רפלקציה** 26](#_Toc136484744)

[**7 - הקוד** 27](#_Toc136484745)

[**7:1 ServerMushApp** 27](#_Toc136484746)

[**7:2 guiApp** 41](#_Toc136484747)

[**7:3 Party** 57](#_Toc136484748)

[**7:4 database** 58](#_Toc136484749)

# **1: מבוא**

## **1.1 תכולת הספר**

ספר פרויקט זה יכיל בו כל מה שצריך לדעת על הפרויקט השנתי שלי, איך הוא עובד, מהן הסיבות שבחרתי להכין דווקא אותו, מידע למשתמש ולמתכנת.

## **1.2 הרקע לפרויקט**

במהלך השנים שלי במגמת המחשבים ביצעתי כמה פרויקטים(אחד באסמבלר ואחד בפיתון) ושניהם היו משחקים יחסית פשוטים והחלטתי שהשנה אני אמשיך בכיוון הזה של פרויקט, שאפשר לייצר אינטראקציות איתו בקלות ושתמיד אפשר לראות או לשמוע אם יש בעיות בפרויקט בקלות יחסית. כשקיבלנו את המשימה ואת ההוראות לפרויקט השנתי השנה העליתי לעצמי כל מיני רעיונות שונים לפרויקט השנתי, אם זה היה מרשימת רעיונות שהמורים הביאו לנו, התייעצות עם תלמידים שסיימו את המגמה והתייעצות עם מורי המחשבים השונים בביתי ספר. לאחר כל הרעיונות שקיבלתי הייתה לי תחושה והרעיון שאני מאוד ארצה לעשות פרויקט שקשור למשהו שאני מאוד אוהב וכך אוהב לעבוד עליו ולחקור עליו עם יותר רצון והנאה. ואז יום אחד יצאתי לריצה עם אחד החברים שלי ותוך כדי הריצה רצינו להקשיב לאותו השיר בלי להשתמש ברמקול או בווליום גבוהה, שכל אחד יוכל להקשיב לשיר מהטלפון שלו או מהאזניות שלו לאחר חיפוש קצר אחרי הריצה בחנות האפליקציות בטלפון הבנו שאין באמת אפליקציה חינמית שמאפשרת לך לשמוע את אותה המוזיקה כמו החבר שלך בוא זמנית וישר צץ לי הרעיון של למה שאני לא אעשה את זה ואפתח משהו דומה בשביל הפרויקט השנתי שלי בסייבר כדי לתת לאחרים שרוצים לחוות זאת כמוני את האפשרות.

## **1.3 הסיבות לבחירת הנושא**

בחרתי את נושא השמע מהסיבה העיקרית שזהו נושא שמאוד מעניין אותי, אני שומע מוזיקה רוב היום ואפילו קשה לי לעשות דברים כמו ללמוד ולהתאמן בלי השימוש במוזיקה אז אמרתי לעצמי שאם אני אעשה פרויקט הנקשר למוזיקה בצורה מסוימת יהיה לי נורא קל להביא את עצמי לעבוד עליו ולעסוק בו לזמן רב. בנוסף ראיתי שיש מחסור בשוק באפליקציות חינמיות שעושות את מה שאני עושה בפרויקט זה, משמיע בו זמנית בכמה מכשירים שיר שהמשתמש בוחר (מרשימת שירים קבועה מראש).

## **1.4 תהליך מחקר הפרויקט**

בתחילת הפרויקט רציתי מאוד שלא תהיה לי הגבלה של מספר שירים או סוג שירים שאני יכול לשמוע אז עשיתי מחקר על איזה אתרים, ספריות, שיטות שאוכל להשתמש בהן כדי להשיג את הקובץ של כל שיר שארצה, ואז ראיתי שלאפליקציית ספוטיפי יש אפשרות להשיג מידע על כל שיר שקיים להם במאגר, אז התחלתי להתנסות עם זה ואז הסתבר שמדובר רק במידע של הכותב, הזמר והמילים ולא קובץ הסופי של השיר.

אז המשכתי לחפש והסתבר לי שהדרך הכי פשוטה שלי לעשות את זה בה זה בדרך ידנית שבה אני בוחר שירים ופשוט מוריד אותם כקובץ. בנוסף בתחילת הפרויקט רציתי לפתח את הפרויקט שלי לאפליקציה אז חיפשתי אפשרויות שונות המאפשרות להפוך קוד פייטון לאפליקציות אנדרואיד וקיבלתי המלצה מחבר לכיתה להשתמש בספרייה שנקראת טוגה, כשהתחלתי לעבוד היה לי נורא קשה לעבוד ולהבין את ספרייה זו אז בחרתי להחליף לספריית הקיווי.

## **1.5 המצב הקיים בשוק**

כיום בשוק ישנן המון אפליקציות המתעסקות בהשמעת מוזיקה ולכל אחת מהן יש את הגרסה שלהן להשמעה משותפת בו זמנית של מוזיקה, חלק יוצרות רשימות שירים משותפות שהמשתמשים יוכלו להיכנס אליהן ולשמוע את השירים האהובים עליהם ועל חבריהם לבד, חלקן עושות לך הגבלה על כמות המשתתפים שיכולים להקשיב לך בו זמנית וחלקן כמובן לא חינמיות. החידוש בפרויקט שלי הוא שאין לי הגבלה של כמה אנשים יכולים להשתמש בזה בו זמנית, זה חינמי ותמיד אני יכול להוסיף שירים ללא בעיה לפרויקט כדי שיהיה יותר גיוון בבחירת השירים.

## **1.6 אתגרים מרכזיים והפתרונות שלי**

האתגרים המרכזיים שלי בפרויקט היו אתגרים שקשורים להעברת מידע בין השרת ללקוח, איך אני עושה זאת בצורה הטובה ביותר ואיך אני מוודא שהלקוח שולח רק מה שהוא אמור לשלוח ושרת יודע להתמודד עם כל מה לקוח שולח. וכמובן איך להעביר ללקוח את השיר ולהפעיל את השיר אצל הלקוח. בנוסף הייתה לי בעיה של איך לפתוח חיבור חדש עם השרת בשילוב עם הממשק החיצוני של הפרויקט. את הפתרונות פתרתי בעזרת שילוב של מחקר אינטרנטי וקטעי קוד שמצאתי באינטרנט.

# 2: מבנה הפרויקט

המבנה מחולק לשלושה חלקים חלק ראשון הוא השרת חלק שני הוא הלקוח(שאצלו קיים הממשק החיצוני) ובסיס נתונים.

מאשר או דוחה תלוי אם קיים או לא

מעביר שם + סיסמא או שם של שיר כדי לבדוק אם קיים בבסיס נתונים

שולח פקודות (שם + סיסמא) או שם של שיר

אם יש אישור מחזיר את תוכן השיר אם לא הודעת דחייה

**Database**

**בסיס נתונים**

**ServerMushApp**

**שרת**

**guiApp**

**לקוח**

הלקוח זהו החלק שהמשתמש רואה, זהו ממשק חיצוני ששולח לשרת הודעות לפי מעשיו של הלקוח בו.

השרת זהו החלק שמקבל את ההודעות של הלקוחות, ומתמודד עם כל הודעה לפי מה שהיא אומרת לעשות וממי שהיא מגיעה השרת בודק מידע שמקבל מהלקוח עם בסיס הנתונים ולפי הבדיקה הוא מחזיר תגובות ללקוח.

## **2:1 הלקוח – הממשק החיצוני**

הלקוח והממשק החיצוני פועלים בצורה של מסלול קבוע, תחילה הוא מבצע קישור עם השרת ולאחר מכן למשתמש יש אפשרות לבחור בין להירשם מחדש לאפליקציה, להירשם למשתמש קיים או רק להיכנס ל"מסיבה". ולפי מה שבוחר המשתמש תמשך פעולתו באפליקציה, אם המשתמש בוחר להירשם מחדש או להירשם למשתמש קיים, הממשק החיצוני יבקש ממנו להכניס שם וסיסמא ואם הם נכנסו כמו שצריך לאפליקציה הם יקבלו את קוד ה"מסיבה" שלהם(שאותו יוכלו לשתף לחבריהם) ומתי שהמשתמשים רוצים הם יוכלו להתחיל את המסיבה ואת בחירת השירים שהם רוצים לשמוע.

**Home Page**

Join A Party

Sign Up

Log In

Log In Page

Sign Up Page

Join Party Page

The details entered are ok.

Pressed start button

The details entered are ok.

In Case Entered Wrong Name or Password

In Case Entered Wrong Party Code

Party for client

Party for leader

After enter page

### **2:1:1 Home Page**

דף ממשק הנתונים הזה ראשית מוודא שהחיבור עם השרת בסדר על ידי שליחת בקשה, כאשר הוא מקבל תשובה חיובית הוא בונה את הממשק החיצוני עם 3 כפתורים, כפתור אחד להירשם למשתמש קיים השני זהו כפתור של לפתוח משתמש חדש והכפתור השלישי הוא להצטרף ל"מסיבה" קיימת המשתמש לוח. על אחד הכפתורים וזה שולח אותו לעמוד הבא.

A screenshot of a computer

Description automatically generated with medium confidence

### **2:1:2 Log In Page and Sign Up page**

שני מחלקות אלו של הממשק החיצוני מאוד דומות, בשתיהן המשתמש צריך להכניס את השם שלו והסיסמא שלו והוא מכניס את זה כשהוא לוחץ על הכפתור. שולחים לשרת את השם והסיסמא ואם השרת מאשר התחברות, הוא מחזיר את הקוד של ה"מסיבה" של הלקוח והלקוח מתקדם לשלב הבא. אם השרת לא מאשר התחברות ניתנת ללקוח לבצע התחברות שוב ולשנות את השם והסיסמא שלו.

A screenshot of a computer

Description automatically generated

### **2:1:3 Join Party Page**

במחלקה זו של הממשק החיצוני מבקשים מהלקוח להכניס את הקוד של ה"מסיבה" שהוא רוצה להיכנס אליה, לוקחים את קוד זה ושולחים אותו לשרת ואם מקבלים מהשרת אישור התחברות ממשיכים למחלקה הבאה, אם לא ניתנת האפשרות לחזור ולהכניס קוד שונה.

A screenshot of a computer

Description automatically generated

### **2:1:4 After Enter Page**

במחלקה זו אנחנו פשוט מחקים עד שזה שפתח את ה"מסיבה" יתחיל עם המסיבה, בזמן זה לקוחות אחרים יכולים להיכנס למסיבה זו אבל ברגע שהמשתמש לוחץ על כפתור ההתחלה ה"מסיבה" מתחילה ואי אפשר להיכנס אליה יותר.

A screenshot of a computer

Description automatically generated with medium confidence

### **2:1:5 Party For Client**

במחלקה זו ישנה לולאה שמקבלת כל פעם את התוכן של הקובץ של השיר שבחר זה שפתח את ה"מסיבה", היא מקבלת את המידע ומכניסה אותו לקובץ חדש וברגע שהיא מכניסה את כל התוכן לקובץ היא מפעילה את השיר שבקובץ.

### A screenshot of a computer Description automatically generated with medium confidence**2:1:6 Party For Leader**

במחלקה זו אנחנו מחקים עד שהלקוח מכניס שם של השיר שהוא רוצה לשמוע ולוחץ על כפתור ההשמעה, כאשר הוא לוחץ על הכפתור אנחנו שולחים את השם של השיר שהלקוח רוצה לשמוע לשרת והשרת מחזיר לנו את התוכן של הקובץ של השיר, כאשר אנחנו מקבלים את התוכן של השיר אנחנו כותבים אותו לקובץ חדש ומשמיעים אותו.

A screenshot of a computer

Description automatically generated with medium confidence

# **2:2 השרת**

בפרויקט זה אני משתמש בפרוטוקול

כדי לבצע העברת מידע בין השרת ללקוח ובין הלקוח לשרת.HTTP

השרת מכיל המון פונקציות אבל יש בו 3 פונקציות עיקריות שהמידע עובר בעיקר ביניהן.

A picture containing text, screenshot, diagram, design

Description automatically generatedMain(), process\_client(), thread\_of\_party()

### **2:2:1 main()**

הפונקציה הזאת אחראית על כל מה שקשור ל לקבל את המידע מהלקוח ולוודא שהוא לא עושה בעיות. בפונקציה זאת יש לולאה אין סופית שבתוכה יש מנגנון

שלפיו כל לקוח שמתחבר לשרת צריך לשלוח לו קודם הודעת התחברות ואז Select

שהשרת יאשר אותו. במקרה ומדובר בלקוח שכבר התחבר השרת קורא את המידע שהלקוח שלח לו ושולח את מידע זה לפונקציה הבאה שתתמודד עם המידע ותחזיר לשרת מה לעשות ומה לשלוח ללקוח חזרה.

### **2:2:2 create\_http\_respond()**

הפונקציה הזאת מקבלת מה שהשרת רוצה להחזיר ללקוח ומסדרת את מידע זה בצורה של בקשה סטנדרטית של פרוטוקול

.HTTP

### **2:2:3 process\_client()**

פונקציה זו מקבלת את הבקשה של הלקוח ולפי הבקשה היא פועלת, עם הבקשה היא ליצור חשבון חדש הפונקציה לוקחת את השם והסיסמא שלו ושולחת לפונקציה אחרת שתוסיף את הלקוח לבסיס נתונים. אם הלקוח רוצה להיכנס לחשבון שקיים כבר, הפונקציה לוקחת את השם וסיסמתו ושולחת לפונקציה שבדוק האם הוא באמת קיים בבסיס נתונים. אם הלקוח רוצה להצטרף למסיבה, הפונקציה תבדוק האם המסיבה לה הוא רוצה להצטרף קיימת ואם היא קיימת הוא מוסיף אותו אליה, אם הלקוח רוצה להתחיל את המסיבה שלו, הפונקציה תחזיר לפונקציה המרכזית שהוא רוצה לעשות כך, הפונקציה מחזירה לפונקציה הראשית את מה שהלקוח רצה לעשות והאם יש אישור שזה אכן קרה וכמובן אם הלקוח יצר חשבון חדש או התחבר לחשבון קיים אז הפונקציה גם מחזירה לפונקציה המרכזית את הקוד של המסיבה של הלקוח.

### **2:2:4 validate\_http\_request()**

פונקציה זו מקבלת את הבקשה של הלקוח ומחזירה האם בקשה זאת עומדת בהגדרות של בקשה סטנדרטית של פרוטוקול

HTTP.

### **2:2:5 check\_in\_db()**

פונקציה זו מקבלת את השם משתמש ואת הסיסמא של המשתמש, מהצפנת את הסיסמא ובודקת האם אכן קיים משתמש כזה בבסיס הנתונים היא מחזירה האם קיים או לא קיים משתמש כזה.

### **2:2:6 add\_to\_db()**

הפונקציה הזאת מקבל את שם המשתמש והסיסמא של המשתמש ומצפינה את הסיסמא ואז מוסיפה אותו לבסיס הנתונים, פונקציה זו מחזירה אם המשתמש הוסף בהצלחה לבסיס הנתונים.

### **2:2:7 hash\_string()**

פונקציה זו מקבלת את הסיסמא של המשתמש ומחזירה את ערכה המוצפן על ידי

של הסיסמא.Hashlib

### **2:2:8 generate\_new\_party()**

פונקציה זו יוצרת אובייקט חדש מסוג מסיבה ומגרילה לו רנדומאלית קוד בעל 5 ספרות ובנוסף מוסיפה לו את ה

של זה שפתח את המסיבה.socket

### **2:2:9 party\_thread()**

בפונקציה זו אנחנו מקבלים את הבקשות שירים של הלקוח, בודקים אם השיר הזה קיים בבסיס נתונים, אם כן אנחנו קוראים את הקובץ של השיר ושולחים את התוכן של הקובץ לפונקציה אחרת שמטרתה היא לשלוח את התוכן לכל האנשים באובייקט המסיבה של הלקוח .אם אין אנחנו שולחים למנהיג של המסיבה ששיר זה לא קיים לנו במאגר שירים.

### **2:2:10 get\_song\_data()**

פונקציה זו מקבלת את השם של הקובץ שהיא צריכה לקרוא והיא פותחת רשימה וכל פעם מוסיפה לרשימה עוד חלק מהשיר עד שכל התוכן של השיר ברשימה. בסוף הפונקציה מחזירה את הרשימה.

### **2:2:11 sender()**

פונקציה זו נפתחת עבור כל חבר במסיבה כ

ועבור כל אחד מהחברים במסיבה היא שולחת את כל התוכן של השיר Thread והודעה סופית שמודיעה כי כל השיר נשלח ואת השם של השיר.

# **3: מדריך למשתמש**

## **3.1 סביבת עבודה**

המערכת צריכה לפעול על מערכת ההפעלה

יש לוודא שמריצים את השרת לפני שמריצים את הלקוחות ושאכן כלwindows השירים שרשומים בבסיס הנתונים אכן מורדים למחשב ולוודא שבקובץ

אכן יש את הכתובת הנכונה של השרת.guiApp.py

(IP)

## **3:2 הפעלת פרויקט**

לאחר וידוא שכל הדברים בסביבת העבודה הם כמו שצריך, אנחנו יכולים להפעיל את הפרויקט. כדי להתחיל זאת נריץ את הקובץ את

ולאחר מכן נריץ את הקובץ ServerMushApp.py

לאחר ההרצה של שניהם נקבל את המסך הבא:guiApp.py

A screenshot of a computer

Description automatically generated with medium confidence

במסך זה יש לך אפשרות או לפתוח משתמש חדש או להיכנס למשתמש קיים או פשוט להיכנס למסיבה. תבחר את מה שאתה רוצה לעשות, במקרה ובחרת לפתוח משתמש חדש או להיכנס למשתמש קיים, המסך יראה כך.

A screenshot of a computer

Description automatically generated

ובמקרה זה אתה צריך להכניס את השם והסיסמא שאתה רוצה או את השם והסיסמא של המשתמש הקיים כשסיימת לחץ על הכפתור. אם הכנסת נתונים נכונים המסך שלך יראה כך:

A screenshot of a computer

Description automatically generated with medium confidence

במסך זה אתה מקבל את המספר מסיבה שלך, אתה מוזמן לשתף אותו עם חבריך שיכנסו למסיבה. מתי שכולם נכנסו אתה מוזמן ללחוץ על כפתור ההתחלה. לאחר לחיצה על הכפתור המסך שלך יראה כך:

A screenshot of a computer

Description automatically generated with medium confidence

במסך זה אתה מוזמן להכניס את השמות של השירים שאתה רוצה שאתה וחבריך תשמעו (שים לב לרשום לפי ההוראות שרשומות על המסך – לרשום באותיות קטנות וגם במקום רווחים לשים מקף. לאחר בחירת השיר ורשימה כמו שצריך תלחץ על כפתור ההפעלה וכעבור קצת זמן השיר שלך ישמע.

**במקרה ובהתחלה בחרת באפשרות של להצטרף למסיבה**

אתה תקבל את המסך הבא:

A screenshot of a computer

Description automatically generated

במסך זה אתה אמור להכניס את הקוד של המסיבה אליה אתה רוצה להצטרף. אם תכניס קוד שאכן קיים תקבל את המסך הבא:

A screenshot of a computer

Description automatically generated with medium confidence

פה אתה רק צריך לחקות עד שמנהיג המסיבה יבחר בשירים שהוא רוצה לשמוע ואתה תשמע אותם גם.

# **4: בסיס נתונים**

בבסיס הנתונים יש לי שתי טבלאות, אחת מהן היא טבלה ששומרת נתונים על הלקוחות שהתחברו כבר, היא שומרת אצלה את השם של הלקוח, סיסמא שמוצפנת בהאש, את התעודות זהות שהשרת מעניק לו, ואת ה

שלו. והטבלה השנייה היא מחזיקה לי את השירים כך שהיא מחזיקה את Socket השם של

השיר שרשום בצורה שאני רוצה שהלקוח ירשום אותו ובעמודה השנייה את השם קובץ של השיר הזה.

**דוגמא לשירים שעל בסיס הנתונים :**

**A screenshot of a computer

Description automatically generated with medium confidence**

# 5: מדריך למפתח

הפרויקט מחולק ל3 חלקים עיקריים- השרת, הלקוח ושרת הנתונים. אך חוץ מכך יש לנו עוד 15 קבצי שירים, קובץ פייטון שבו יש את המחלקה "מסיבה" וקובץ לוגים אחד.

בפרק זה יש הסבר על הפעולות, מה הן עושות ומה הן מחזירות (חלקן).

פירוט נוסף נמצא בקוד.

# **GuiApp**

## **5:1 ClientCommunication()**

במחלקה זו אנחנו פותחים את ה

של הלקוח ויוצרים את כל הפעולות אשר זה יכול לעשות.Socket

### **5:1:1 innit**

בפונקציה זו אנחנו בעצם פותחים את המשתנה

Client\_socket

### **5:1:2 connect**

בפונקציה זו אנחנו מנסים להתחבר לשרת שלנו אם לא מצליחים סוגרים את הסוקט.

### **5:1:3 send\_data**

פונקציה זו מקבלת הודעה לשלוח לשרת ובנוסף מקבלים גם משתנה שאומר לנו אם להצפין את הודעה זו או לא לפני השליחה לשרת.

### **5:1:3 receiving data**

פונקציה זו מקבל משתנה שמראה כמה היא צריכה לקרוא מהודעת השרת ועוד משתנה שאומר אם אנחנו צריכים לפצח את זה אחרי הקריאה או לא.

### **5:1:4 close**

פונקציה זו סוגרת את הסוקט של הלקוח.

## **5:2 create\_http\_request**

פונקציה זו מקבלת את המידע שהלקוח רוצה להעביר לשרת ובעצם מסדרת את המידע בצורה של בקשת פרטוקול

סטנדרטית.http

## **5:3 handle\_http\_respond**

פונקציה זו בעצם היא זו שמקבלת אצלה את התגובה של השרת לבקשה שלנו ולפי התגובה של השרת היא מחזירה את הדברים העיקריים בה.

## **5:4 HomePage**

מחלקה זו עוסקת בדף הבית של המשתמש, מתחילה קישור עם השרת ובעלת 3 כפתורים אשר הלקוח יכול לבחור מה הוא רוצה לעשות באפליקציה.

### **5:4:1 open\_join\_party\_func**

במקרה והמשתמש לחץ על כפתור להצטרף למסיבה זה שולח אותו לעמוד של הל היצטרף למסיבה.

### **5:4:2 open\_sign\_in\_func**

במקרה והמשתמש לחץ על כפתור ליצור חשבון חדש זה שולח אותו לעמוד של הליצור חשבון חדש.

### **5:4:3 open\_log\_in\_func**

במקרה והמשתמש לחץ על כפתור להיכנס למשתמש קיים זה שולח אותו לעמוד של הל היכנס למשתמש קיים.

## **5:5 JoinPartyPage**

במחלקה זו אנחנו רוצים שהלקוח ירשום לנו את המספר של המסיבה אליה הוא רוצה להצטרף ולאחר מכן שילחץ על הכפתור

### **5:5:1 party\_enter**

בפונקציה זו אנחנו שולחים לשרת את המספר של המסיבה אליה הלקוח רוצה להצטרף ומחקים לתגובה מהשרת של אישור אם השרת אישר אז אנחנו עוברים לדף של המסיבה.

## **5:6 LogClientInfo**

במחלקה זו אנחנו רוצים שהלקוח ייתן לנו את השם והסיסמא שלנו כדי שנוכל לשלוח אותם ולאמת אותם עם השרת.

### **5:6:1 log\_enter**

בפונקציה זו אנחנו לוקחים את התוכן שנתן לנו הלקוח ושולחים אותו לשרת ומחקים שיאשר לנו את ההתחברות עם הבסיס נתונים ושיחזיר לנו את הקוד למסיבה, אם הכל בסדר עם השרת והבקשה של הלקוח נעבור לדף הבא ונעביר אליו גם את הקוד של המסיבה.

## **5:7 SignClientInfo**

במחלקה זו אנחנו רוצים שהלקוח ייתן לנו את השם והסיסמא שלנו כדי שנוכל לשלוח אותם ולאמת אותם עם השרת.

### **5:7:1sign\_enter**

בפונקציה זו אנחנו לוקחים את התוכן שנתן לנו הלקוח ושולחים אותו לשרת ומחקים שיאשר לנו את ההתחברות עם הבסיס נתונים ושיחזיר לנו את הקוד למסיבה, אם הכל בסדר עם השרת והבקשה של הלקוח נעבור לדף הבא ונעביר אליו גם את הקוד של המסיבה.

## **5:8 AfterEnterPage**

במחלקה זו אנחנו משתפים עם הלקוח את קוד המסיבה שלו וגם מחקים עד שילחץ על כפתור ההתחלת מסיבה.

### **5:8:1 start**

בפונקציה זו אנחנו שולחים לשרת שאנחנו רוצים להתחיל עם המסיבה ומחקים לאישור של השרת שהתחלנו את המסיבה ואז עוברים לדף הבא.

## **5:9 PartyPage1**

מחלקה זו היא המחלקה של המנהיג של המסיבה, במחלקה זו אנחנו רוצים שהלקוח יכניס שם של שיר לפי ההוראות שקיבל וילחץ על כפתור ההפעלה.

### **5:9:1 play\_song**

פונקציה לוקחת את מה שרשם המשתמש בבחירת השיר ושולחת לשרת, היא מחכה עד שהשרת ישלח חזרה את האישור ואת תוכן הקובץ שיר שרצה המשתמש בפונקציה זו אנחנו גם שומרים את תוכן הקובץ בקובץ חדש ומפעילים את קובץ שמע זה.

## **5:10 PartyPage2**

מחלקה זו היא המחלקה של הלקוח שלא שולט במסיבה, במחלקה זו אנחנו תמיד מחכים למידע מהשרת וברגע שאנחנו מקבלים אותו אנחנו אכן מוודאים שמדובר בשיר ומורידים אותו לקובץ חדש ומפעילים אותו.

# **ServerMushApp**

## **5:11 main()**

הפונקציה הזאת אחראית על כל מה שקשור ל לקבל את המידע מהלקוח ולוודא שהוא לא עושה בעיות. בפונקציה זאת יש לולאה אין סופית שבתוכה יש מנגנון

שלפיו כל לקוח שמתחבר לשרת צריך לשלוח לו קודם הודעת התחברות ואז Select

שהשרת יאשר אותו. במקרה ומדובר בלקוח שכבר התחבר השרת קורא את המידע שהלקוח שלח לו ושולח את מידע זה לפונקציה הבאה שתתמודד עם המידע ותחזיר לשרת מה לעשות ומה לשלוח ללקוח חזרה.

## **5:12 create\_http\_respond()**

הפונקציה הזאת מקבלת מה שהשרת רוצה להחזיר ללקוח ומסדרת את מידע זה בצורה של בקשה סטנדרטית של פרוטוקול

.HTTP

## **5:13 process\_client()**

פונקציה זו מקבלת את הבקשה של הלקוח ולפי הבקשה היא פועלת, עם הבקשה היא ליצור חשבון חדש הפונקציה לוקחת את השם והסיסמא שלו ושולחת לפונקציה אחרת שתוסיף את הלקוח לבסיס נתונים. אם הלקוח רוצה להיכנס לחשבון שקיים כבר, הפונקציה לוקחת את השם וסיסמתו ושולחת לפונקציה שבדוק האם הוא באמת קיים בבסיס נתונים. אם הלקוח רוצה להצטרף למסיבה, הפונקציה תבדוק האם המסיבה לה הוא רוצה להצטרף קיימת ואם היא קיימת הוא מוסיף אותו אליה, אם הלקוח רוצה להתחיל את המסיבה שלו, הפונקציה תחזיר לפונקציה המרכזית שהוא רוצה לעשות כך, הפונקציה מחזירה לפונקציה הראשית את מה שהלקוח רצה לעשות והאם יש אישור שזה אכן קרה וכמובן אם הלקוח יצר חשבון חדש או התחבר לחשבון קיים אז הפונקציה גם מחזירה לפונקציה המרכזית את הקוד של המסיבה של הלקוח.

## **5:14 validate\_http\_request()**

פונקציה זו מקבלת את הבקשה של הלקוח ומחזירה האם בקשה זאת עומדת בהגדרות של בקשה סטנדרטית של פרוטוקול

HTTP.

## **5:15 check\_in\_db()**

פונקציה זו מקבלת את השם משתמש ואת הסיסמא של המשתמש, מהצפנת את הסיסמא ובודקת האם אכן קיים משתמש כזה בבסיס הנתונים היא מחזירה האם קיים או לא קיים משתמש כזה.

## **5:16 add\_to\_db()**

הפונקציה הזאת מקבל את שם המשתמש והסיסמא של המשתמש ומצפינה את הסיסמא ואז מוסיפה אותו לבסיס הנתונים, פונקציה זו מחזירה אם המשתמש הוסף בהצלחה לבסיס הנתונים.

## **5:17 hash\_string()**

פונקציה זו מקבלת את הסיסמא של המשתמש ומחזירה את ערכה המוצפן על ידי

של הסיסמא.Hashlib

## **5:18 generate\_new\_party()**

פונקציה זו יוצרת אובייקט חדש מסוג מסיבה ומגרילה לו רנדומאלית קוד בעל 5 ספרות ובנוסף מוסיפה לו את ה

של זה שפתח את המסיבה.socket

## **5:19 party\_thread()**

בפונקציה זו אנחנו מקבלים את הבקשות שירים של הלקוח, בודקים אם השיר הזה קיים בבסיס נתונים, אם כן אנחנו קוראים את הקובץ של השיר ושולחים את התוכן של הקובץ לפונקציה אחרת שמטרתה היא לשלוח את התוכן לכל האנשים באובייקט המסיבה של הלקוח .אם אין אנחנו שולחים למנהיג של המסיבה ששיר זה לא קיים לנו במאגר שירים.

## **5:20 get\_song\_data()**

פונקציה זו מקבלת את השם של הקובץ שהיא צריכה לקרוא והיא פותחת רשימה וכל פעם מוסיפה לרשימה עוד חלק מהשיר עד שכל התוכן של השיר ברשימה. בסוף הפונקציה מחזירה את הרשימה.

## **5:21 sender()**

פונקציה זו נפתחת עבור כל חבר במסיבה כ

ועבור כל אחד מהחברים במסיבה היא שולחת את כל התוכן של השיר Thread והודעה סופית שמודיעה כי כל השיר נשלח ואת השם של השיר.

# **Party**

## **5:22 Party**

במחלקה זו אנחנו יוצרים אובייקט בשם מסיבה ומוסיפים פונקציות שמאפשרות לנו לקבל ולשנות את כל הנתונים בו.

### **5:22:1 innit**

בפונקציה זו אנחנו בעצם בונים את האובייקט על ידי הסוקט של המנהל של המסיבה וקוד המסיבה.

### **5:22:2 add\_guest**

פונקציה זו בעצם מוסיפה לרשימת המשתתפים את הסוקט של הלקוח שרוצה להצתרף.

### **5:22:3 get\_guests**

פונקציה זו מחזירה לנו את רשימת המשתתפים במסיבה.

### **5:22:4 get\_code**

פונקציה זו מחזירה לנו את הקוד של המסיבה.

### **5:22:5 set\_mode**

פונקציה זו מקבלת את הערך החדש של המצב של המסיבה ומשנה את המצב של המסיבה לערך זה.

## **5:22:6 get\_mode**

פונקציה זו מחזירה לנו את המצב של המסיבה.

### **5:22:7 get\_leader**

פונקציה זו מחזירה לנו את הסוקט של מנהיג המסיבה.

### **5:22:8 set\_started**

פונקציה זו מקבלת ערך חדש לערך האם המסיבה התחילה ומשנה את הערך הזה לערך החדש.

### **5:22:9 get\_started**

פונקציה זו מחזירה לנו את הערך של האם היא התחילה או לא.

# **6: רפלקציה**

העבודה על הפרויקט הייתה מאוד מאתגרת ולחוצה בזמן אבל גם מהנה בוא זמנית, בזמן העבודה על הפרויקט קיבלתי המון תכונות חדשות לאישיות שלי כגון האפשרות לעשות הבחנה נפשית ולראות מתי אני עובד קשה ומתי אני מוותר לעצמי וגם קיבלתי את היכולת להיעזר יותר בחברים, מורים ומידע אינטרנטי שלא נעזרתי בהם יותר מדי בפרויקטים האחרונים שעשיתי, במיוחד בהקשבה לחברים והתייעצות איתם על קשיים שהיה להם במהלך הפרויקט ואיך הם פתרו אותם, מה שמאוד עזר לי לתכנן להמשך ולבנות את הקוד שלי בצורה האידיאלית ביותר ובמיוחד עוד משהו שאקח איתי להמשך זה שלא להרגיש מאוים מפרויקטים גדולים או לפחד מהם אלא להפך לנסות לפרק אותם לחלקים קטנים ואז לא רק שהם מרגישים קלים יותר ולא מאיימים אלא גם יש לך יותר מוטיבציה לשבת ולעבוד עליהם וכל פעם למחוק לך מטלה קטנה שהיא חלק מפרויקט גדול. הקשיים שהיו לי בפרויקט היו שבשילוב עם שאר הדברים שיש לי בחיים לא התעסקתי עם הפרויקט הרבה זמן במהלך השנה אלא הייתה לי עבודה אינטנסיבית מאוד בחודש האחרון של העבודה וזה גרם לי ללחץ רב עם מקצועות אחרים אבל עם עבודה קשה ועבודה תמידית על הפרויקט ועל מחקר לפרויקט הצלחתי בסופו של דבר להביא פרויקט קיים ועובד בזמן שלפני חודש וחצי מהיום חשבתי שלא אגיע למצב שאני בו היום והיו בי מחשבות של לפרוש מההרחבה.

המסקנות שלי מהפרויקט הן שאסור לי לפחד מלקחת דברים ופרויקטים גדולים על עצמי ואסור לי להתעכב איתם יותר מדי אלא אני צריך להתחיל לעבוד עליהם ברגע שאני יכול מה שיגרום ליותר נוחות וסדר בעתיד.

אם היום היה היום שבו קיבלתי את ההנחיות של הפרויקט, לא הייתי מתעכב יותר מדי וישר מתחיל עבודה עליו ולא הייתי מפחד ומרגיש מאוים ממנו כמו שקרה.

בסופו של דבר אני מאוד שמח שאני בסיטואציה היום של להגיש את הפרויקט ואני גאה בעצמי מאוד שלא ויתרתי לעצמי והחלטתי ללכת על זה עד הסוף ולסיים את הדרך שהתחלתי בהרחבה הזאת.

# **7 - הקוד**

## **7:1 ServerMushApp**

**A screenshot of a computer program

Description automatically generated with medium confidence**

**A screenshot of a computer program

Description automatically generated with medium confidence**

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A picture containing text, screenshot, menu

Description automatically generated

A picture containing text, screenshot, font

Description automatically generated

## **7:2 guiApp**

A screen shot of a computer program

Description automatically generated with low confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screen shot of a computer program

Description automatically generated with low confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screen shot of a computer program

Description automatically generated with medium confidence

A screen shot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

## **7:3 Party**

A screen shot of a computer program

Description automatically generated with medium confidence

## **7:4 database**

A screenshot of a computer program

Description automatically generated with medium confidence