

Horizon Music®

Ofek Harel .

Summer 2020.

(COVID-19).



Table of Contents

The Project	3
User Guide	4
Networking	6
The Code	8

The Project

Horizon Music היא שילוב של עקרונות מתוכנות הזרמת המוזיקה הידועים והטובים שיש, התוכנה בנויה באופן אסטטי, מודולרי וקוד קריא ומובן השומר על עקרונות הכתיבה והשימוש החוזר.

התוכנה מספקת:

- ספריית שירים רחבה המציעה מגוון של סגנונות וסוגים.
- שמיעת מוזיקה דרך האינטרנט ללא צורך להוריד את השיר.
- הורדת שירים ושמיעה מקומית.

האפליקציה עוצבה באופן עצמאי בעזרת תוכנת Adobe After Effect ומימוש ה – UI בעזרת שפת עיצוב הנקראת "Kivy". עיצוב ה UI נתן דגש על עיצוב פשוט ומודרני תוך לקיחת רעיונות מאפליקציות המוזיקה המוכרות (Apple Music, Spotify, YouTube Music). ראשית עוצבו מסכים שלמים כדי להוציא ולהביא על "הנייר" את הנראות ואחר כך עיצוב כל כפתור ("ווידג'ט") בנפרד ומיקום ואינטראקציה בעזרת – Kivy.

הקוד שמאחורי האפליקציה נכתב בעזרת שפת Python. קוד קריא פשוט ומודולרי המממש את עקרונות התכנות שלמדתי בפרויקט ה-FRC.

User Guide

דרישות מערכת:

Client	Server
1. פייטון 3.7.7	1. פייטון 3.7.7
2. ספריות KIVY 1.11.1	2. ספריית השירים
3. קבצי ה-GUI	3. PyCharm
4. PyCharm	

הרצת האפליקציה היא תהליך מאוד פשוט :

1. מריצים את קובץ השרת (Server.py).
2. הולכים לקובץ הפרמטרים (Constants.py) ומשנים פרמטר הנמצא תחת Network – את ה-IP של השרת (על ברירת מחדל הוא מקומי).
3. מריצים את קובץ הלקוח (HorizonMusic.py).

מריצים דרך תוכנת PyCharm שמספקת תמיכה ב-Modules.

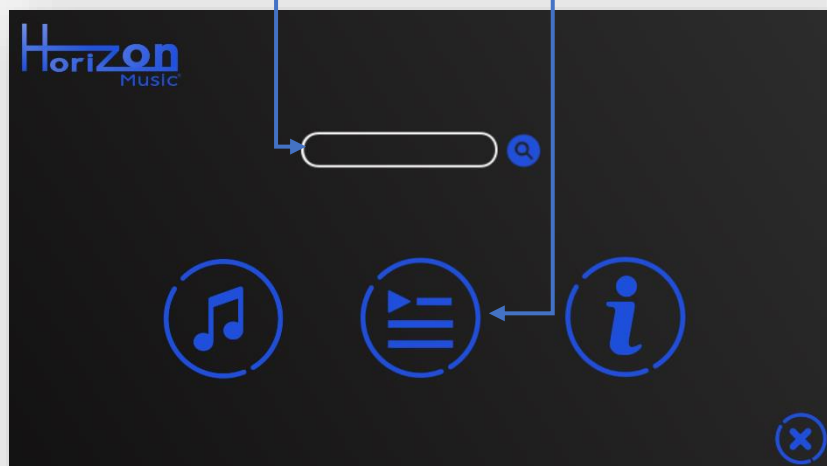
האפליקציה:

האפליקציה מספקת (כרגע) רק את האפשרות לנגן שירים Online –

- דרך שורת החיפוש .

להקליד שם שיר (אם השיר לא קיים זה לא יקבל) – אם השיר קיים, הוא יתנגן

- דרך הפלייליסט הראשי מהשרת. תצוגת כל השירים (148) הנמצאים על השרת.



:Media Control

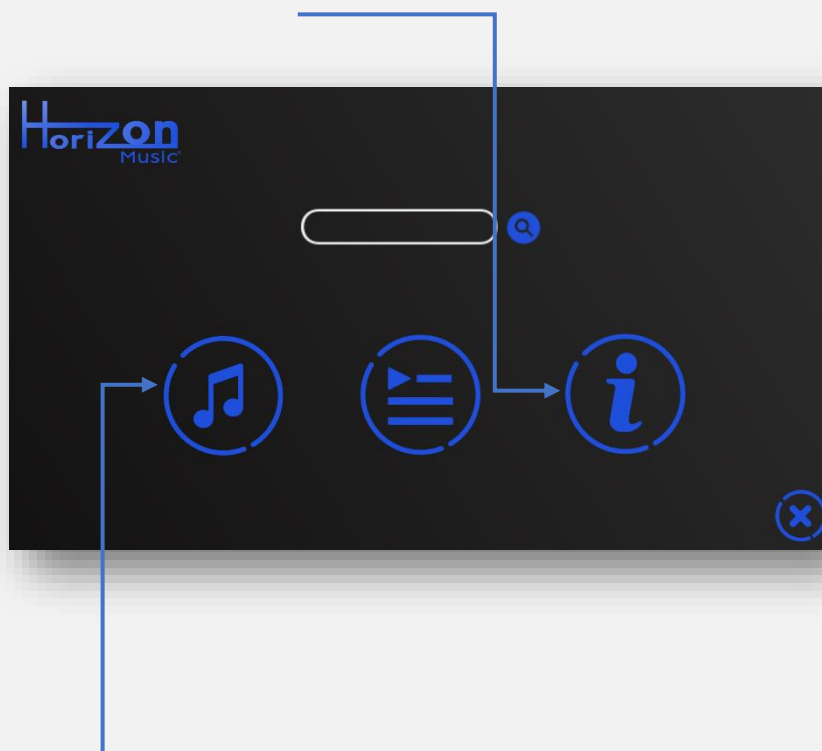
ניתן לשלוט בשירים – עצור / המשך (toggle), דלג, חזור.



ניתן לשלוט בכל שיר המתנגן – לא משנה באיזה דרך הוא נוגן (בין עם דרך החיפוש או דרך הפלייליסט הראשי).

השרת ישלח אוטומטית את השיר הבא בסדר ברגע שהשיר הנוכחי נגמר.

מידע נוסף הודות הפרויקט ניתן למצוא במסך ה-INFO במסך הבית של האפליקציה.



Work in progress (Will be published with the next version).

Project example video

Networking

חלקי ההודעה:

כל הודעת רשת של האפליקציה מורכבת מ-2 חלקים:

1. אורך ההודעה.
 2. ההודעה עצמה.
- a. פעולות, מידע ונתונים.

הפרוטוקול:

הקושי ביצירת פרוטוקול תקשורת המעביר קבצי mp3. הוא מציאת תו מפריד נטרלי (כלומר שלא מופיע בפורמט) לכן נאלצתי לעבוד עם תו מפריד: |||

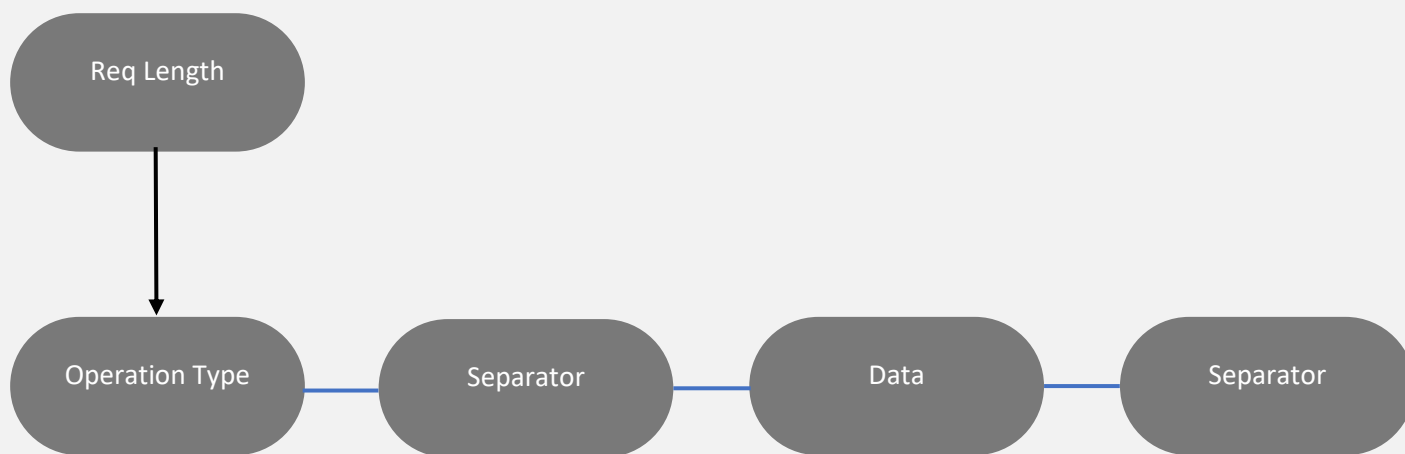
לכל הודעה יש סוג פעולה (operation type):

1. ALL_SONGS
 - a. בקשה ששולח הלקוח והשרת עונה עם מידע
 - b. המידע – רשימת השירים שיש ברגע זה אצל השרת [אמן, שם השיר]
2. REQ_SONGS
 - a. בקשה ששולח הלקוח עם מספר השיר במערך השירים של השרת
 - b. השרת עונה ושולח את הבייטים של השיר המבוקש
3. DISCONNECT
 - a. בקשה ששולח הלקוח
 - b. השרת לא שולח חזרה כלום, רק סוגר את ה-Socket

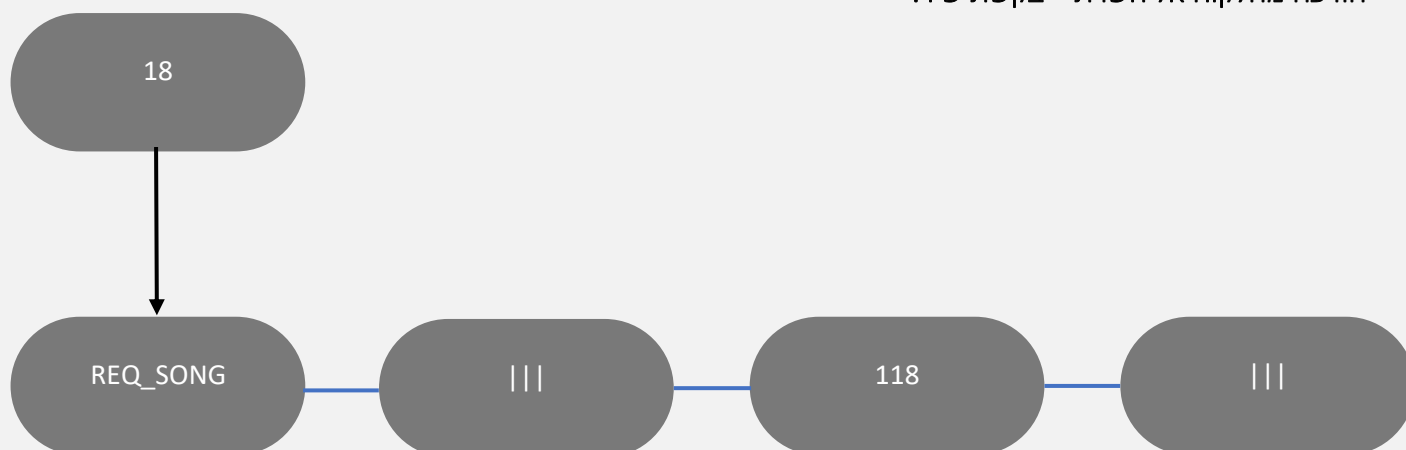
כאשר לקוח חדש מתחבר:

4. הוא מבקש מהשרת את רשימת השירים שיש לו - ALL_SONGS
 - a. אמן ושם השיר
5. הלקוח יכול לבקש ולחפש איזה שיר שהוא רוצה - REQ_SONG
 - b. כך שהבדיקה אם השיר שהוא רוצה קיים נבדקת על הלקוח
2. הלקוח שולח בקשת התנתקות - DISCONNECT
 - a. נסגרת האפליקציה וה-Socket

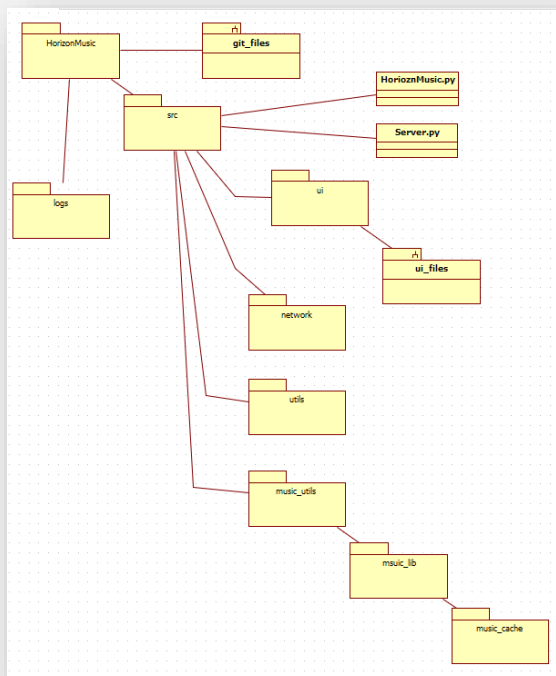
דוגמת הודעה גנרית:



הודעה מהלקוח אל השרת – בקשת שיר:



The Code



מבנה הפרויקט

הפרויקט בנוי בצורה קריאה ונוחה כך שאפשר לכתוב בקלות את מה שרוצים מבלי להתעסק יותר מידי מאחורי הקלעים, דוגמה טובה לכך – מחלקת `NetworkCommunication` :

מחלקה זו מספקת גישה מאוד נוחה ליצירת הודעות ושליחתן כך שלא צריך להתעסק במאחורי הקלעים אלא ישר לגשת לעניין.

המחלקה מכילה מגוון של פעולות שעוזרות בתחום התקשורת – הרכבת הודעה והכנסה לתוך פרוטוקול, שליחה, קבלה של הוגעה ופיצול ההודעה לחלקיה.

Horizon Music®