

# המחלקה להנדסת תוכנה

פרויקט גמר – תשע"ח

דמיון שירים על פי תגים

MUSIC SIMILARITY

מאת

אבי כהן

שגיא מרסיאנו

תאריך:

אישור:

מנחה אקדמי: דר' מרים אללוף

תאריך:

אישור:

רכז הפרויקטים: דר' אסף שפיינר

## מערכות ניהול הפרויקט:

#	מערכת	מיקום
1	מאגר קוד	<a href="https://github.com/avicohen89/Music-Similarity">https://github.com/avicohen89/Music-Similarity</a>
2	יומן	<a href="https://calendar.google.com/calendar/embed?src=pnu92pav3s91nku9u35gks3m0c%40group.calendar.google.com&amp;ctz=Asia%2FJerusalem">https://calendar.google.com/calendar/embed?src=pnu92pav3s91nku9u35gks3m0c%40group.calendar.google.com&amp;ctz=Asia%2FJerusalem</a>
3	סרטון גירסת אלפא	<a href="https://youtu.be/Kuv0vKDEa68">https://youtu.be/Kuv0vKDEa68</a>

## תקציר

פרויקט גמר זה נעשה בשיתוף עם מיזם תמרינגה. תמרינגה הינה פלטפורמה היוצרת גירוי ושמע מותאמים אישית לחולי אלצהיימר ובכך משפרת את מצבם. במהלך הפרויקט, תכננו ובנינו מערכת המלצת שירים על פי קריטריונים ודמיון בין שירים עבור תמרינגה. מערכת ההמלצה תתבסס על מאגרי מידע של מוזיקה שיתופיים ברשת ותציג עבור המשתמש את השירים המתאימים לו וקישורי השמעה בתוך המערכת, ללא יציאה לאתר חיצוני. המערכת תלמד את הקריטריונים ותגים המאפיינים את השירים שאהב ובכך תמליץ עבורו על שירים בעתיד.

בחיבור זה, נציג את הפתרון המוצע, את תכנונו ואת אופן מימושו, זאת לאחר מחקרים רבים שהיו על נושאים שונים.

## מילון מונחים, סימנים וקיצורים

- YouTube הוא אתר אינטרנט לאחסון ושיתוף של סרטוני וידאו.
- MusicBrainz הוא מאגר מוזיקלי אינטרנטי שמבוסס על קוד פתוח. המאגר מכיל מידע על מוזיקה, ומאפשר להשתמש במידע זה באופן ציבורי. מופיע גם בראשי תיבות MB.
- Last.fm הוא אתר שמספר רדיו אינטרנטי ורשת חברתית שעיקרה הוא מוזיקה. האתר מכיל מידע רב על מוזיקה ומאפשר להשתמש במידע זה באופן ציבורי.
- Discogs הוא אתר ומסד נתונים המכיל מידע על הקלטות אודיו. האתר מתמחה במוזיקה אלקטרונית. האתר מאפשר קניה ומכירה של מוצרי מוזיקה שונים (דיסקים, תקליטים וכדומה).

## תוכן עניינים

	<b>1. מבוא</b>	שגיאה! הסימניה אינה מוגדרת.
<b>4</b>	<b>2. תיאור הבעיה</b>	
4	2.1 דרישות ואפיון הבעיה	
6	2.2 הבעיה מבחינת הנדסת תוכנה	
<b>7</b>	<b>3. תיאור הפתרון</b>	
7	3.1 אפיון המערכת	
8	3.2 ארכיטקטורה	
8	3.2.1 גישה למאגרים חיצוניים	
8	3.2.2 מבנה ה-DATABASE המקומי של המערכת	
9	3.2.3 תיאור הפיתרון המוצע	
9	3.3 תהליכים ונתוני המערכת	
10	3.4 תיאור הכלים המשמשים לפתרון	
<b>11</b>	<b>4. בדיקות</b>	
<b>14</b>	<b>5. סקירת עבודות דומות בספרות והשוואה</b>	
<b>14</b>	<b>6. סיכום/מסקנות</b>	
<b>14</b>	<b>5. נספחים</b>	
14	5.1 רשימת ספרות \ ביבליוגרפיה	
15	5.2 תרשימים וטבלאות	
16	5.3 תכנון הפרויקט	
16	5.4 טבלת סיכונים	
17	5.5 רשימת טבלת דרישות	

## 1. מבוא

מחלת האלצהיימר היא מחלה קשה וחשוכת מרפא, המתבטאת בניוון איטי ומתמשך עד מוות של תאי מוח ומשפיעה על מערכת העצבים המרכזית. המחלה נפוצה בייחוד בקרב קשישים. המחלה מתאפיינת בירידה קוגניטיבית. תחילה בולטת בייחוד הפגיעה בזיכרון. עם הזמן נפגעים גם תפקודים עליונים אחרים של המוח, כמו ההתמצאות, יכולת החשיבה והסקת המסקנות ועוד. עם המשך הניוון המוחי נפגעת גם הפעילות המוטורית של החולה (כגון הליכה, אכילה וכו'). בשלבים המתקדמים, החולה הוא חסר ישע וזקוק לעזרה בכל פעולותיו.

למוזיקה יש כוח מיוחד המשפיע על אנשים החולים באלצהיימר ודמנטיות קשורות. השפעת המוזיקה מראה תוצאות משכנעות גם בשלבים מאוחרים ביותר של המחלה. כשמבוצע בטיפול זה שימוש ראוי, מוזיקה יכולה לשפר את מצב הרוח, לטפל בתסיסת מתח מושרה, לעורר מגעים חברתיים חיוביים, להקל על התפקוד הקוגניטיבי ולשפר תנועות מוטוריות.

טיפול בקשישים הלוקים באלצהיימר, הביאה את לאה כהן סבן וסטלה מלינצר לפתח את תמרינגה. תמרינגה הינה מערכת חכמה המציגה שירים וסרטים שיותאמו אינדיבידואלית לכל קשיש על סמך אלגוריתם סינון שמתבסס על גילו ושפתו של הקשיש. וכן קטגוריות כלליות נוספות שנמצאו אטרקטיביות בקרב הדיירים בבית האבות. ישנו דף 'מועדפים' אישי לכל משתמש עם לינקים פרטיים שמועדפים עליו, הן אם הוא סימן אותם כמועדפים במערכת הסינון והן אם הוא הוסיף אותם בעצמו, כך שיהיה לו גישה נוחה לתכנים הקרובים לליבו.

מטרת הפרויקט היא לבנות מערכת המלצה לומדת עבור תמרינגה, שתתאים עבור כל משתמש את התכנים המתאימים על פי מאפייניו. המערכת תרכיז, מתוך מאגר אינטרנטי שיתופי, שירים על פי פרמטרים כגון שם אמן, שם השיר, שנת הוצאה, איזור גיאוגרפי, פופולאריות ותגים המאפיינים את השיר וסגנונו, ותציג את השירים המתאימים ביותר לכל משתמש על פי מאפיינים אלו. בנוסף המערכת תדע לקשר בין שירים דומים על פי מאפיינים ומשתמשים דומים שאהבו את אותם שירים.

ישנם הרבה מאגרים המכילים מטה-דטה של שירים ומידע עליהם, אך אין מאגר שמרכז את כל המידע שנחוץ לפרויקט כגון שם אמן, שם השיר, שנת הוצאה, איזור גיאוגרפי, פופולאריות ותגים המאפיינים את השיר וסגנונו ובצורה מסודרת ומקוטלגת. כדי לאסוף מידע על שיר ספציפי יש לפנות להרבה מאגרי מידע שונים בעלי ממשקים ותוכנה שונים.

## 2. תיאור הבעיה

### 2.1 דרישות ואפיון הבעיה

כיום, חיפוש מוזיקה במערכות מוכרות כגון YouTube מתבצע על ידי זיהוי והזנה של שם המבצע, שם השיר, מילים מתוך השיר או זיהוי על פי קטע ממנו (Shazam). YouTube הוא אתר אינטרנט המאפשר למשתמשים רשומים לשתף (העלאה וצפייה) תכני מדיה במגוון רחב של תחומים, לרבות מוזיקה. האתר מאפשר לדרג ולהגיב על סרטונים אחרים, ליצור רשימות השמעה פרטיות ולשתף אותן. בנוסף, בהתבסס על צפיות קודמות ועל ידי אלגוריתם חכם, מציע האתר המלצות על תכנים שעשויים להתאים למשתמש.

נרצה ליצור מערכת המלצה נוחה וידידותית למשתמש, כזאת המיועדת להשמעת תכני מוזיקה לאוכלוסייה ספציפית. עבור אנשים שלא מתמצאים בYouTube, חווית השימוש יכולה להיות לא פשוטה, ניתן בקלות לגלוש לתכנים לא רלוונטיים ולהיתקל בפרסומות. החיסרון המרכזי בYouTube הוא שאלגוריתם ההמלצה של YouTube דורש שימוש ממושך באתר על מנת להגיע לתוצאות רלוונטיות למשתמש, אנו נרצה מערכת שתדע להמליץ על תכני מדיה ללא שימוש קודם במערכת, בעזרת פרמטרים קבועים מראש המאפיינים את המשתמש.

על מנת לבנות data base במערכת המלצה ולהשתמש בו ביעילות כבסיס לה, הבנו כי לא ניתן להסתפק רק בלינקים של תכנים, אלא גם במידע אודות התוכן עצמו.

תחילת העבודה על הפרויקט החלה במחקר מעמיק על מאגרים המכילים מידע על תכני מוזיקה. הצורך שלנו במידע מגוון, המכיל מידע מפורט כגון שם המבצע, שם השיר, שנת הוצאה, תגים המאפיינים את השיר, דירוג פופולאריות, החל מאמצע שנות ה-50 התגלה כבעייתי ונמצאו סתירות וחוסרים במידע רלוונטי. לצערינו אין תיעוד מספק מהו המידע הניתן ע"י המאגר ולכן יש ראשית ללמוד את ממשק הגישה התכנית (API) הייחודי לכל מאגר.

מאגרי המידע הקיימים שאותם חקרנו הם :

- YouTube
- MusicBrainz
- Last.fm
- Discogs
- Spotify בהמשך

פירוט למחקר מתואר בפסקאות הבאות :

מחקר ראשוני החל בפלטפורמה המוכרת והשימושית ביותר כיום, YouTube [\[https://www.youtube.com\]](https://www.youtube.com).

מחקר מעמיק ב-Youtube API מראה כי הוא לא מספק שדות מטאדאטה אשר יכולים לסייע באפיון תוכן המדיה ובניית מערכת ההמלצה. ראינו כי התכנים אינם מאופיינים בסגנון מוזיקלי, או שנת הוצאה ומכילים בעיקר שם מבצע ושם השיר ותוכן מילולי שהכניס המשתמש שהעלה את השיר לאתר. התכנים בעיקר מסווגים על פי סוג התוכן ברמת סרט או שיר. הבנו כי YouTube לא נותן מענה לדרישתנו והרחבנו את המחקר והחיפוש לפלטפורמות פחות מוכרות וחדשות לנו.

דרישה נוספת שלנו בפרויקט היא היכולת להציג ולנגן, תוך כדי הישארות במערכת שלנו ולא לצאת למערכת חיצונית, כל שיר שעלה כתוצאה מהרצת המערכת, וברור כי YouTube היא הפלטפורמה הראשית לכך, מכוון שהיא המערכת הגדולה ביותר ומכילה תכנים רבים. בתחילת המחקר לא נמצא הדרך לקבלת לינק לשיר לפי שם השיר, מאחר והשירים מאופיינים ב-id ייחודי ל-Youtube. במהלך חודש אוקטובר יצא לאוויר YouTube API 3 שאחד מחידושיהו הינו קבלת רשימת id של שירים על פי חיפוש מסוים. חידוש נוסף הוא קבלת לינק להצגת השיר לפי id מסוים. ובכך בעצם התגברנו על הקושי והתאפשרה לנו היכולת לנגן כל שיר שיעלה בחכתנו.

התחנה הבאה הייתה מאגר המוזיקה והמידע MusicBrainz (MB) [\[https://musicbrainz.org\]](https://musicbrainz.org). מחקר מעמיק ב-MB מראה כי הוא מכיל מידע רב אודות תוכן מוזיקה, אך לאחר חקירה ממושכת ב-MB API התגלו קשיים רבים בחיפוש מידע ספציפי (חיפוש לפני שנה או איזור גיאוגרפי) ונמצא מידע כפול וסותר כגון שנת הוצאה ומיקום. חקירה אינטנסיבית הביאה אותנו למסקנות הבאות. ראשית כל, תוכן מדיה (שיר) מכיל שנת הוצאה ומיקום גיאוגרפי על פי שנת הוצאת האלבום המכיל את השיר והמיקום הגיאוגרפי.

הסיבה העיקרית לכפילות שנת הוצאה ומיקום היא הוצאות אלבומי להיטים או אלבומים המכילים שירים מתקופות שונות, ולכן לכל שיר קיימים מספר רב של שנות הוצאה ומיקום. ולכן, על פי בדיקות רבות ואימות הנתונים בגוגל על כל שירים רבים, הבנו כי הדרך הטובה ביותר להתמודד עם הבעיה ולקבל את המידע המדויק ביותר היא לקחת את שנת ההוצאה הראשון שמופיע ברשימה. בנוסף גלינו כי MB אינו מכיל דירוג פופולאריות אודות תוכן מדיה, מידע שחשוב מאוד לפרויקט. הבנו כי לא ניתן להסתפק אך ורק ב-MB במקור ראשי למידע ונצטרך לדעת לשלב בין מאגרים שונים.

בעיה נוספת שנתקלנו איתה במהלך העבודה עם MusicBrainz, הייתה שלאחר 16 בקשות של השרת לקבלת מידע, החיבור היה מסתיים והעברת המידע נפסקה. בכל בקשה היה ניתן לקבל 100 שירים, ולאחר 16 בקשות היה ניתן לקבל 1600 שירים. המטרה שלנו הייתה לקבל מספר גבוה של שירים שיכול להגיע גם ל-300,000 שירים בכל פעם שאנו נגשים ל-MusicBrainz (שירים שיצאו שנה מסוימת/מיקום גיאוגרפי). על מנת להתמודד עם הבעיה הורדנו ושמרנו בקבצים את כל המידע על השירים מהשנים 1880 עד 2018. הנתונים נשמרו בקבצי json המכילים 100 שירים בכל קובץ, לאחר מכן העברנו את כל המידע לתוך שרת המונגול עם המידע שהיה רלוונטי לנו. עדיין היה חסר מידע מסוים (כמו דירוג השיר או מספר צפיות, והיכולת להשמיע אותו) והשלמנו את המידע מ-Youtube שהוא מס מזהה של הוידאו על מנת לנגן אותו במערכת, מספר צפיות ותגים.

מחקר נוסף ביצענו במאגר המוזיקה והמידע Last.fm [\[https://www.last.fm\]](https://www.last.fm). בדומה ל-MB, מציג Last.fm מידע על שירים מתקופות שונות.

לשלב הזה במחקר ידענו כי עלינו לחפש מידע כגון שם מבצע, שם שיר, שנה, תגים אותו השיר, איזור גיאוגרפי ודירוג פופולאריות. בשונה מMB, מאגר המידע Last.fm מציג מצעד שירים באמצעות API לפי תקופה זמן מסוימת, אך בדיקה ידנית עבור מצעד השירים שקיבלנו התברר כי השירים לא עדכניים ולא מציגים את הלהיטים עבור אותה תקופה. בדיקה מעמיקה בפורומים שונים הביאה לידיעתנו כי מאגר המידע לא מעודכן מאז 2015. הוחלט לזנוח אותו ולהמשיך במחקר.

מחקר נוסף ביצענו במאגר המידע והמוזיקה Discogs [<https://www.discogs.com/>]. בדומה לקודמיו, ה-Discogs מספק מידע רב על שירים, כגון שנות הוצאה ומידע על חברות תקליטים שונות. האתר מתחב בעיקר במוזיקה אלקטרונית, ואנו נרצה מערכת המלצה שתשלוט במגוון רחב של מוזיקה. בנוסף, ראינו כי המידע מקוטלג על פי אלבומים שלמים, וכי אנו נרצה לקבל מידע אודות שיר ספציפי. דבר נוסף שפסל את השימוש בפלטפורמה זאת במערכת שלנו היה כי המידע אינו מקוטלג על פי מיקום גיאוגרפי, דבר שמאוד קריטי למטרת הפרויקט, היות והקריטריון הראשי הוא מיקום גיאוגרפי, בנוסף לשנת הוצאת השיר.

## 2.2 הבעיה מבחינת הנדסת תוכנה

הפרויקט מספק אתגרים בתחום מערכת המלצה, המבוססות על קשר בין תכנים דומים ושליפת תכנים רלוונטיים על פי פרמטרים. על מנת לבנות מערכת המלצה שתעבוד בצורה מדויקת ויעילה, קיים צורך להבין איך לקשר בין תכנים דומים, ועולות השאלות הבאות:

- כיצד נשלוף מידע רלוונטי ומדויק?
- איך מזהים קווי שוני/דמיון בין תכנים?
- על פי אילו מאפיינים ניתן יהיה לבסס קשר בין תוכן אחד לאחר?
- כיצד לזהות קשר בין אישיות שונות על פי תכנים דומים?

לאחר חקירה מעמיקה בארבעת מאגרי מידע, הבנו כי ה-data source הקיים באינטרנט אינו מוצג בצורה ברורה ומ"מוכנה לשימוש", המלווה במידע כפול וסותר. הבנו כי אין ברירה, ועל מנת להכניס מידע מדויק ככל האפשר עבור תוכן מדיה database שלנו, יש צורך לברור מתוך כל המידע שקיבלנו ולהבין: האם יש תבנית קבועה ששם נמצא שנת ההוצאה והמיקום הראשוני של השיר? איך להתעלם מהוצאות שונות בשנים מאוחרות יותר?

הבדיקה נעשתה לעיתים על ידי בדיקה וחיפוש באינטרנט, חיפוש תקופות ואזורים שנכונים לתקופתנו, ואף נעזרנו באנשים קרובים בגילאים שונים, ממוצאים שונים על מנת לאמת את המידע.

לאחר מחקר מעמיק ב-API של הפלטפורמות ה"ל, הוחלט לשלב ולהשתמש ב-MusicBrainz ו-YouTube על מנת לבנות data-base שמכיל מידע רלוונטי עבור מערכת ההמלצה.

כעת, נותר לנו לאפיין את מבנה ה-database ולקבוע את ה-data model כך שנאכסן ונשלוף את המידע מתוך data basen בצורה מהירה ויעילה.

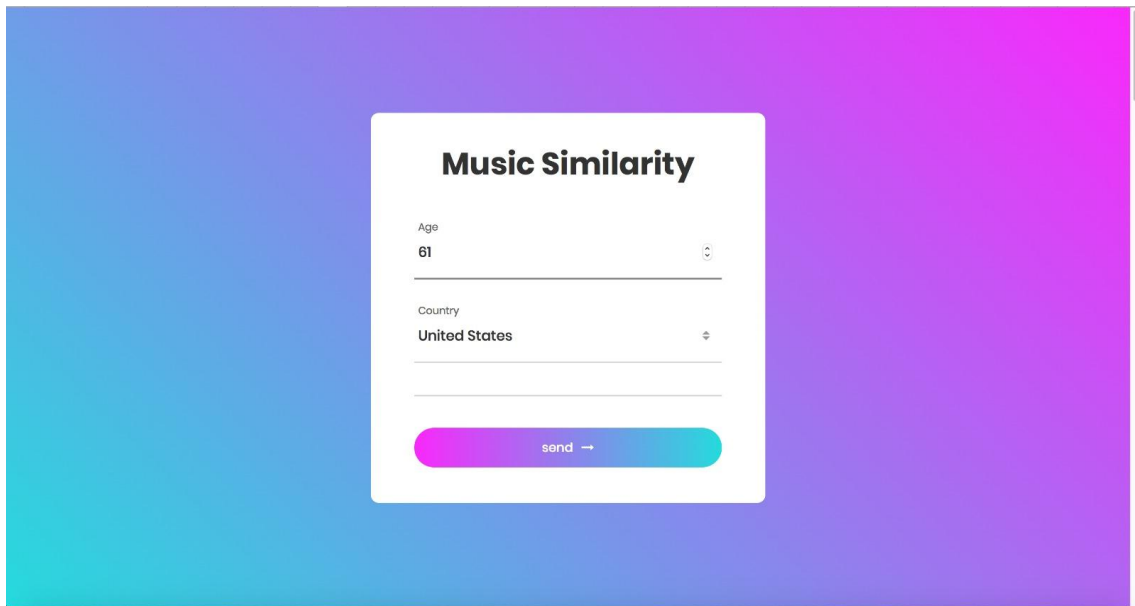
כמו כן, נרצה לדעת באיזה אופן נקשר בין תכנים המזהים כדומים על פי תגים מסוימים, איך לשמור את המידע ה"ל ב-data base ולנצל את הדמיון בניהם כדי לקבל המלצות טובות עבור המשתמשים.

חשיבה רבה על data model, והתייעצות עם אנשים מנוסים הוחלט על מבנה databasen ומבנו מוצג בהמשך. נשתמש ב-database של mongoDB.

### 3. תיאור הפתרון

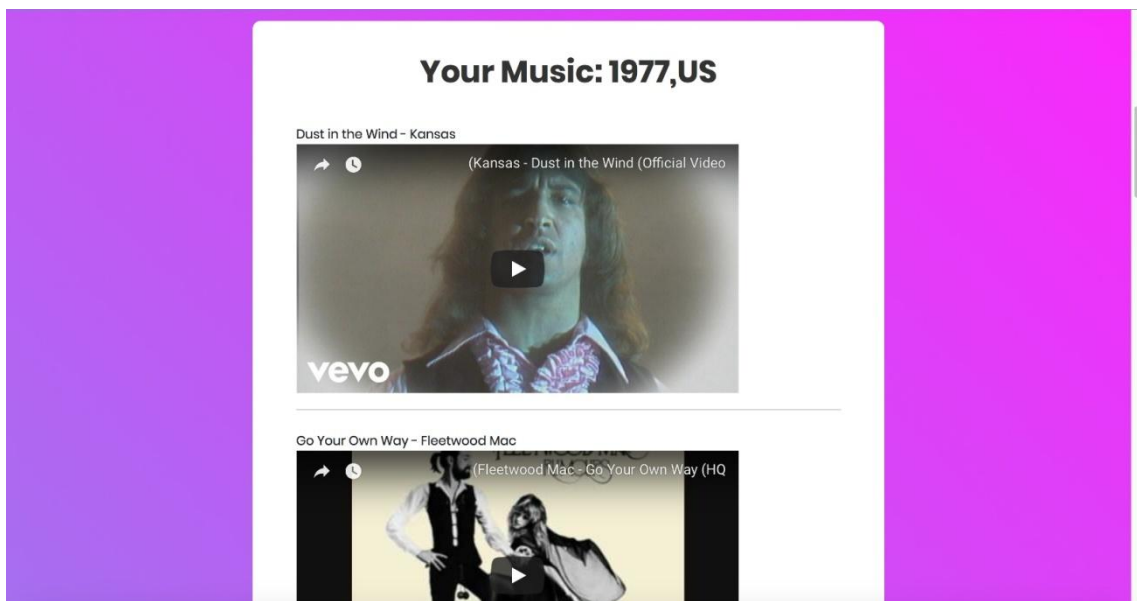
#### 3.1 אפיון המערכת

מערכת בעלת ממשק web. המערכת תאפשר קבלת נתונים מהמשתמש ותציג עבורו רשימת שירים מומלצת בעקבות אלגוריתם המלצה המתבסס על פי נתונים אלה (גיל וארץ מוצא). המשתמש יוכל להאזין לתכנים, ובהמשך, יוכל לקבוע אם אהב או לא ועל פי מידע זה, המערכת תתאים למשתמש שירים נוספים בעתיד.



מסך ראשון של המערכת, המשתמש מתבקש להזין גיל וארץ מוצא:

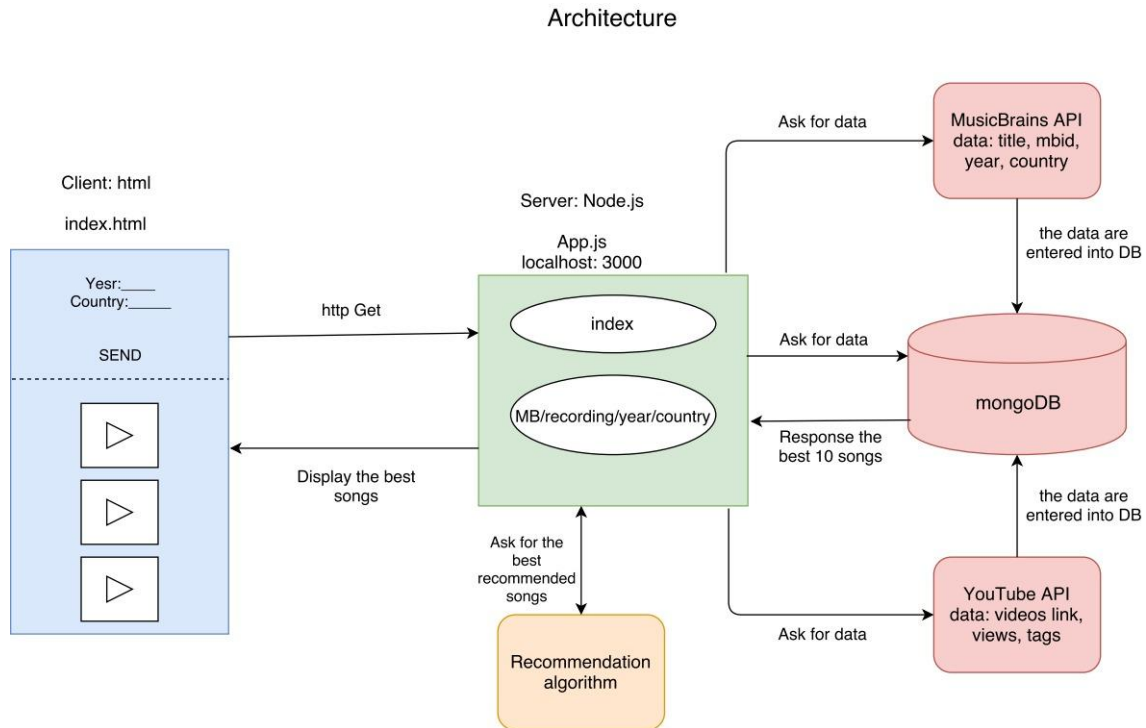
לאחר לחיצה על כפתור Send, אלגוריתם ההמלצה ישלוח מהדאטאבזן רשימה של 10 שירים, והמערכת תציג אותם למשתמש בצירוף קישור לצפייה/השמעה YouTube, בלי לצאת מהמערכת. בנוסף, יוצג שם השיר ושם האמן. בראש העמוד ניתן לראות את השנה בה המשתמש היה בן 20 ובשנה זאת התבצע חיפוש של שירים, כמו כן מוצג גם ארץ המוצא:





### 3.2 ארכיטקטורה

להלן תרשים הארכיטקטורה של המערכת:



#### 3.2.1 גישה למאגרים חיצוניים

על מנת לבסס את Databases המקומי של המערכת, ניגשנו למאגרים חיצוניים על מנת לקבל מהם מידע. לאחר מחקר מקיף, לראשונה ניגש MusicBrainz לקבל שירים ומידע עליהם כגון שם השיר, שם אמן, שנה וארץ הוצאה. בקשת המידע התבצעה לפני שימוש המערכת, וזאת מאחר וה-MB לא מתקשר לאחר 16 בקשות. ולכן ביצענו מראש הורדה גדולה של מידע. לאחר מכן, נשללים את המידע באמצעות YouTube, ממנו נבקש מידע כגון לינק לצפייה עבור כל שיר, מספר צפיות ותגים שמאפיינים את השיר.

הפניה ל-Youtube התבצעה על ידי ספריית Node YouTube. שלחנו בקשת http request וקיבלנו key שבאמצעותו נוכל לתקשר עם YouTube API. אנו בונים את השאלות הרצויות ושולחים לשרת של YouTube, הוא מצידו מחזיר לנו את המידע המבוקש, כגון ID של שיר/וידאו. באמצעות ID הנתון, יכולנו לקבל מידע נוסף על שיר (כגון מספר צפיות ותגים) על ידי שאילתה שכתבנו בעצמנו ומכילה בקשה עבור כל המידע הנתון על הוידאו הספציפי, ותוכו בררנו את המידע הרצוי.

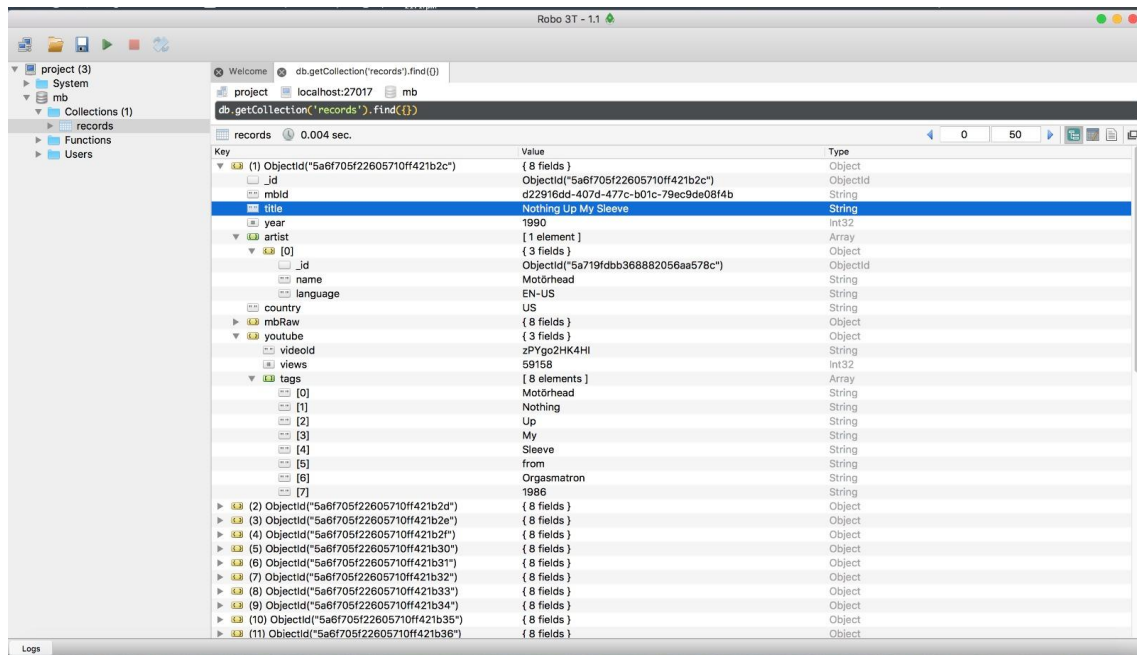
פניה ל-MB התבצעה באופן דומה. הפניה של MB API בבקשה לקבלת מידע, הייתה מוגבלת, מאחר והשרת עצמו מגביל אותנו ל-16 פניות. את המידע קיבלנו בקבצי json, ואותם הכנסנו לdb המקומי בסדר ובאופן הרצוי למערכת. MB מבוסס RDF שהוא מודל החלפת נתונים ברשת המאפשר מיזוג נתונים גם אם מדובר בסכמות שונות.

#### 3.2.2 מבנה Databases המקומי של המערכת.

את המידע שקיבלנו מה-MB ו-Youtube, הכנסנו ל-Databases מקומי, המבוסס MongoDB. dbn מסודר בצורה כזאת שתאמים למבנה השאלתה. מאחר והשאלתה הראשית של המערכת היא קבלת 10 שירים, המדורגים גבוהה ביותר, על פי ארץ מוצא ושנת הוצאה, כך גם מסודר הcollection (טבלה) db. כל השירים



נמצאים בcollection אחד, המאופיין באינדקסים של שנת הוצאה, ארץ מוצא, ודירוג. Indexing בmongo הוא כלי המאפשר לסדר את האובייקטים בdb כך שהשליפה שלהם תהיה יעילה ומהירה. הוא מסדר אותם בסדר יורד (או עולה, תלוי בהגדרה) ובכך במקום לסרוק את כל הcollection לחפש תוצאות מתאימות, הוא ניגש בדיוק למקום בו הם נמצאים. להלן צילום מסך של Database, ניתן לראות את השדות השונים, המכילים מידע על שיר ספציפי (בסדר יורד): ID שניתן לו ע"י מונגו, ID שניתן לו על ידי MB, שם השיר, שנת הוצאה, אמן (שם, ID, שפה), ארץ מוצא, מידע מ-Youtube (מספר צפיות, לינק לצפייה, תגים).



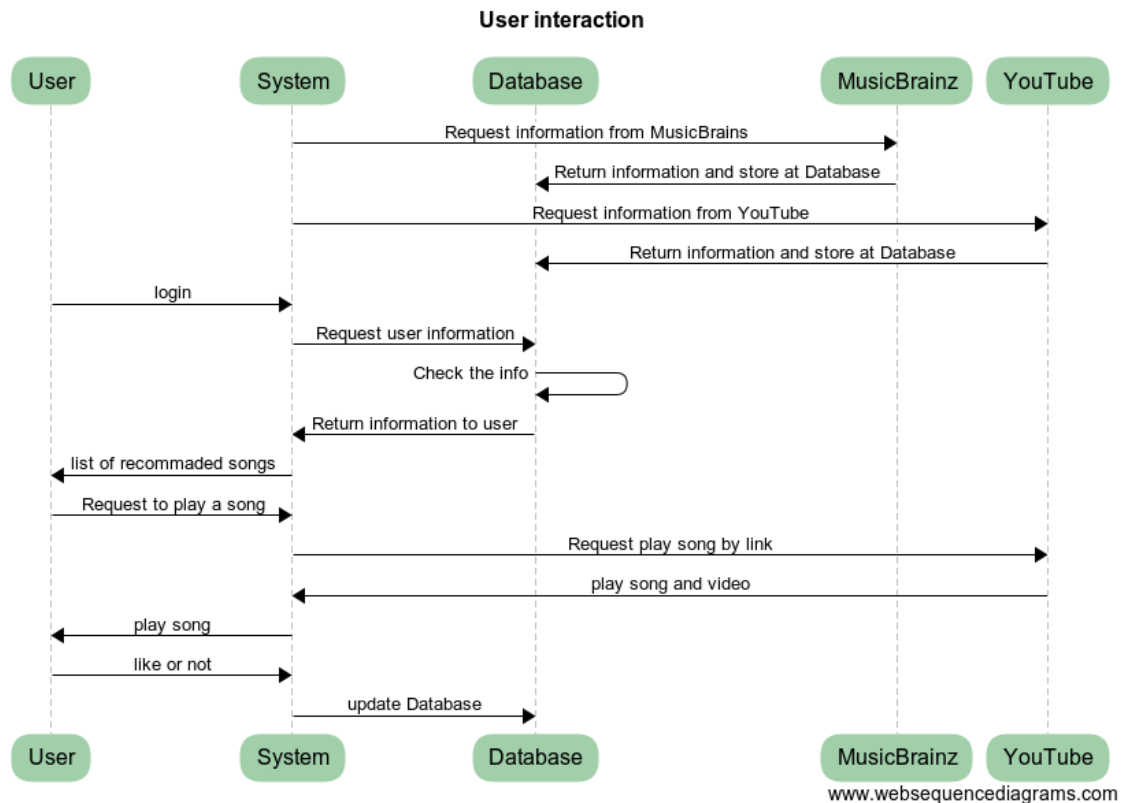
Key	Value	Type
(1) ObjectId("5a6f705f22605710ff421b2c")	{ 8 fields }	Object
_id	ObjectId("5a6f705f22605710ff421b2c")	ObjectId
mbid	d22916dd-407d-477c-b01c-79ec9de08f4b	String
title	Nothing Up My Sleeve	String
year	1990	Int32
artist	[ 1 element ]	Array
[0]	{ 3 fields }	Object
_id	ObjectId("5a719fdbb36882056aa578c")	ObjectId
name	Motörhead	String
language	EN-US	String
country	US	String
mbRaw	{ 8 fields }	Object
youtube	{ 3 fields }	Object
videoid	zPYgo2HK4HI	String
views	59158	Int32
tags	[ 8 elements ]	Array
[0]	Motörhead	String
[1]	Nothing	String
[2]	Up	String
[3]	My	String
[4]	Sleeve	String
[5]	from	String
[6]	Ogasmatron	String
[7]	1990	String
(2) ObjectId("5a6f705f22605710ff421b2d")	{ 8 fields }	Object
(3) ObjectId("5a6f705f22605710ff421b2e")	{ 8 fields }	Object
(4) ObjectId("5a6f705f22605710ff421b2f")	{ 8 fields }	Object
(5) ObjectId("5a6f705f22605710ff421b30")	{ 8 fields }	Object
(6) ObjectId("5a6f705f22605710ff421b31")	{ 8 fields }	Object
(7) ObjectId("5a6f705f22605710ff421b32")	{ 8 fields }	Object
(8) ObjectId("5a6f705f22605710ff421b33")	{ 8 fields }	Object
(9) ObjectId("5a6f705f22605710ff421b34")	{ 8 fields }	Object
(10) ObjectId("5a6f705f22605710ff421b35")	{ 8 fields }	Object
(11) ObjectId("5a6f705f22605710ff421b36")	{ 8 fields }	Object

### 3.2.3 תיאור הפיתרון המוצע

צד לקוח המבוסס על html, javascript הוא זה שפונה אל השרת בבקשה לקבל רשימת שירים מומלצת על פי פרמטרים. הבקשה והפרמטרים נשלחים אל שרת בבקשה המבוססת javascript ו jquery. שרת ה Node.js מקבל את הפניה, ושולף את המידע הרלוונטי מתוך databasen מבוסס mongoDB באמצעות אלגוריתם המלצה. השרת מחזיר את הבקשה ללקוח, ומציג עבורו את רשימת השירים בצירוף לינק השמעה YouTube.

### 3.3 תהליכים ונתוני המערכת

התרחיש הבא מתאר את תהליכי המערכת ומסביר איך היא נבנתה ועתידה להיבנות. כמו כן ניתן לראות מהן הפעולות העיקריות במערכת ומה תפקיד כל שחקן המשתמש בה:



ראשית כל, אנו שואבים את המידע הנדרש ממאגרי מידע חיצוניים, ומאחסנים אותם בDatabase מקומי. db המקומי מסודר כך ששליפת המידע תהיה יעילה ומהירה ובהתאם לשאלות המרכזיות של המערכת. לאחר מכן המשתמש יוכל להיכנס למערכת ולהזין את פרטיו. על פי פרטים אלה יקבל רשימת שירים מומלצת עבורו על פי אלגוריתם המלצה. רשימת השירים מוצגת עבורו בצירוף לינקים להשמעה. בלחיצה על לינק, נשלחת בקשת השמעה מYouTube. בהמשך, יוכל המשתמש לקבוע אם אהב או לא, מידע זה ישמש את אלגוריתם ההמלצה וימליץ עבורו על שירים נוספים בעתיד.

### 3.4 תיאור הכלים המשמשים לפתרון

בבניית המערכת השתמשנו בטכנולוגיות הבאות:

- צד הלקוח: css, html, javascript - צד הלקוח בנוי משלד של html ועליו עיצוב בעזרת css. מידע הגיל והארץ נשלח בעזרת javascript וjquery לשרת וכך גם המידע המתקבל מצד השרת.
- צד השרת: צד השרת המבוסס node.js, ומשתמש בספריות כדוגמת - youtube-node, mongoose, על מנת לשאוב מידע. צד השרת מתווך בין המידע שמאוחסן בשרתי mongo ובין צד הלקוח.
- גישה מהשרת לYouTube: על ידי ספריית Node YouTube, אנו שולחים בקשת http request ומקבלים key שבאמצעותו נוכל לתקשר עם YouTube API. אנו בונים את השאלות הרצויות ושולחים לשרת של YouTube, הוא מצידו מחזיר לנו את המידע המבוקש.
- גישה מהשרת לMusicBrainz: אנו שולחים בקשת http GET לMB API בבקשה לקבלת מידע. MB מבוסס RDF שהוא מודל החלפת נתונים ברשת.
- אחסון המידע- המידע על השירים מאוחסן בשרת mongodb בתוך collection אחד המאופיין באינקסים המאפשרים שליפה מהירה ויעילה של המידע. כל שיר מאוחסן ב document עם הפרטים הרלוונטיים למערכת.

#### 4 תכנית בדיקות

בסיום כל שלב בפיתוח האפליקציה, יבוצעו מספר בדיקות תקינות שימולאו בהתאם בטבלה -

##### בדיקות מערכת

- ווידוא שהמערכת עונה לדרישות.
- בדיקות תקינות קלט מהמשתמש – האם המשתמש אכן יכול להכניס קלט לאפליקציה.
- בדיקות תקינות של בסיס הנתונים – היכולת לשלוף ולהכניס נתונים (C.R.U.D).
- בדיקת ממשק משתמש (GUI) עבור הפקדים והרכיבים במסכי האפליקציה(האם מגיבים ועובדים כנדרש).

מספר בדיקה	מודול נבדק	פעולה (תיאור הבדיקה)	תוצאה צפויה	תוצאה בפועל
1	App Launcher	הפעלת המערכת	המערכת עולה ומוצג מסך הבית שכולל טופס מילוי נתונים וכפתורי שליחה	✓
2	מסך ראשי	בחירת שנה	ניתן להכניס גיל במספרים שלמים, גדולים מ20 וקטנים מ120	✓
3	מסך ראשי	רשימת בחירה- מדינה	נפתחת רשימה של כל המדינות לבחירה	✓
3	מסך ראשי	כפתורי החיפוש במסך הבית- טופס חיפוש	כפתורים עובדים כנדרש כאשר בוחרים בהם	✓
4	מסך הבית	לחיצה על כפתור send	נשלחים הפרטים ומוצגת הודעה בה מצוין מדינה ושנת החיפוש	✓
5	מסך הבית	לחיצה על כפתור by Track	גורמת לאחד משני המיקרים הבאים: 1. מקבלים רשימת שירים במיקרה וקיימים dba 2. הודעה על כך שהנתונים מובאים לבאז db ויש ללחוץ שוב על הכפתור	✓
6	מסך הבית	קלט משתמש	המשתמש מצליח להכניס קלט לחיפוש	✓

##### בדיקות אינטגרציה

- בדיקה האם האפליקציה מתקשרת כנדרש עם השרת.
- בדיקה האם הנתונים שנשלפים מבסיס הנתונים הם נכונים ותקינים.
- האם השרת מתקשר כנדרש עם MusicBrainz
- האם השרת מתקשר כנדרש עם YouTube
- בדיקת תקשורת עם mongo
- אינטגרציה בין לקוח לשרת

מספר בדיקה	מודול נבדק	פעולה (תיאור הבדיקה)	תוצאה צפויה	תוצאה בפועל
1	התחברות לשרת	המערכת פונה לשרת node.js מקומי	המערכת מצליחה להתחבר לשרת	✓
2	שליפת נתונים	האפליקציה פונה לשרת בבקשה לקבלת מידע מבסיס הנתונים	המערכת מצליחה לקבל נתונים מבסיס הנתונים	✓
3	התחברות בין השרת ל-MusicBrainz	השרת פונה לmusicbrainz עם שאילתה לקבל מידע	המערכת מצליחה לתקשר עם musicbrainz	✓
4	התחברות בין השרת ל-Youtube	השרת פונה ל-Youtube עם שאילתה לקבל מידע	המערכת מצליחה לתקשר עם Youtube	✓
5	הוספה ועדכון נתונים מ-musicbrainz, youtube	השרת פונה לmusicbrainz ו-youtube על מנת לקבל נתונים ולהכניס או לעדכן נתונים אלה למסד הנתונים	השרת מצליח לעדכן ולהוסיף נתונים בבסיס הנתונים	✓
6	בדיקת תקשורת עם mongo	המערכת פונה לשירות mongo	נפתח שירות mongo המכיל database	✓
7	אינטגרציה בין לקוח לשרת	צד לקוח מבקש קבלת נתונים מהשרת	מקבל רשימת שירים או תגובה מתאימה	✓
8	אינטגרציה בין שרת ללקוח	צד שרת מחזיר ללקוח נתונים	מחזיר רשימת שירים או תגובה מתאימה	✓

### בדיקות פונקציונאליות

בדיקות המוודאות שהמערכת מגיבה כנדרש לדרישות.

מספר בדיקה	מודול נבדק	פעולה (תיאור הבדיקה)	תוצאה צפויה	תוצאה בפועל
1	מסך ראשי	בדיקת תקינות לנתוני הקלט	הודעת שגיאה במקרה של נתונים לא תקינים (גיל קטן מ-20 או גדול מ-120)	✓
2	העברת נתונים לשרת	האם הנתונים עוברים לשרת ומתקבלים אצלו	הודעת שגיאה במקרה והנתונים לא הגיעו לשרת	✓
3	התקשורת בין השרת ל-db	בדיקה האם השרת ניגש ל-database לקבלת נתונים	במקרה ויש נתונים, הם יחזרו לשרת, במקרה ואין השרת ייגש לmusicbrainz ול-youtube לקבל הנתונים	✓

4	התקשרות בין השרת לבד	בדיקה האם השרת מחזיר נתונים מהdatabases על פי הקלט	המידע יעבור לתצוגה במסך הראשי	✓
5	שרת	בדיקה השירים מתאימים לשנה המתבקשת	השירים שהתקבלו אכן מתאימים לשנה המתבקשת בחיפוש בעזרת Index	✓
6	שרת	בדיקה האם השירים מתאימים למדינה המתבקשת	השירים שהתקבלו אכן מתאימים למדינה המתבקשת בחיפוש בעזרת Index	✓
7	מסך ראשי	בדיקה האם מוצג קישור מתאים לכל שיר	מוצג קישור מתאים להשמעה לכל שיר	✓
8	מסך ראשי	בדיקת תאימות בין כותרת לקישור שיר	הקישור תואם למידע המוצג בכותרת	✓

#### בדיקת תאימות

בדיקה שמטרתה לוודא שהמערכת מוצגת כרגיל בסוגי דפדפנים שונים בעלי רזולוציות שונות ובעלי גרסאות שונות.

מספר בדיקה	מודול נבדק	פעולה (תיאור הבדיקה)	תוצאה צפויה	תוצאה בפועל
1	סוגי דפדפנים שונים : chrom, explorer, safari, firefox	בדיקה האם המערכת מגיבה אותו דבר בדפדפנים שונים בעלי רזולוציות שונות וגרסאות שונות ומוצגת באופן זהה	המערכת מגיבה אותו דבר בכל דפדפן, הטופס ורשימת השירים מוצגת כנדרש והפונקציונאליות נשארת	✓

## 5 סקירת עבודות דומות בספרות והשוואה

Follow the algorithm: An exploratory investigation of music on YouTube

מאמרם של Massimo Airolidi, Davide Beraldi, Alessandro Gandini משנת 2016 החוקר את אלגוריתם ההמלצה של YouTube, המציע באופן אוטומטי רשימה של סרטונים קשורים למשתמש כתגובה לסרטון שצפה כעת. במהלך המחקר נבדקו רשת האסוציאציות בין 22,141 סרטוני מוזיקה של YouTube.  
<https://www.sciencedirect.com/science/article/pii/S0304422X16300973>

Music therapy in moderate and severe dementia of alzheimer's type: a case-control study.  
מאמרם של H. B. Svendsdottir, J. Snaedal משנת 2006 העוסק בקשר בין דצמנציה לבין טיפול באמצעות מוזיקה.  
במהלך כתיבת המאמר, התבצע ניסוי על 38 אנשים הלוקים במחלת האלצהיימר והדמנציה ברמות שונות של חומרה, בשני בתי אבות, שבו השמיעו למטופלים מוזיקה.  
תוך 6 שבועות ניתן היה להצביע על ירידה בהפרעה הנמדדת בסולם מחלת האלצהיימר וכמו כן ירידה בתוקפנות והחרדה.  
מסקנות המאמר הם שקיים קשר ישיר, בטוח ויעיל לטיפול בחרדה ובחומרת מחלת האלצהיימר והדמנציה.  
<https://www.cambridge.org/core/journals/international-psychogeriatrics/article/music-therapy-in-moderate-and-severe-dementia-of-alzheimers-type-a-case-control-study/D51CCDFF17656F27C927D1BA322BA74C>

## 6 סיכום \ מסקנות

מה בוצע עד כה ומסקנות מעניינות שעלו מביצוע הפרויקט.

## 7 נספחים

ספרות, תרשימים נוספים, תכנון הפרויקט, טבלת ניהול סיכונים, טבלת דרישות (URD),

### א. רשימת ספרות \ ביבליוגרפיה

מחשבון המחשב כמה עולה כל שאילת ל API YouTube. YouTube מגביל אותך במספר הבקשות שניתן לבצע בכל ששן בקשות וכך תדע כמה מידע תוכל לבצע בכל ששן כזה.  
[https://developers.google.com/youtube/v3/determine\\_quota\\_cost](https://developers.google.com/youtube/v3/determine_quota_cost)

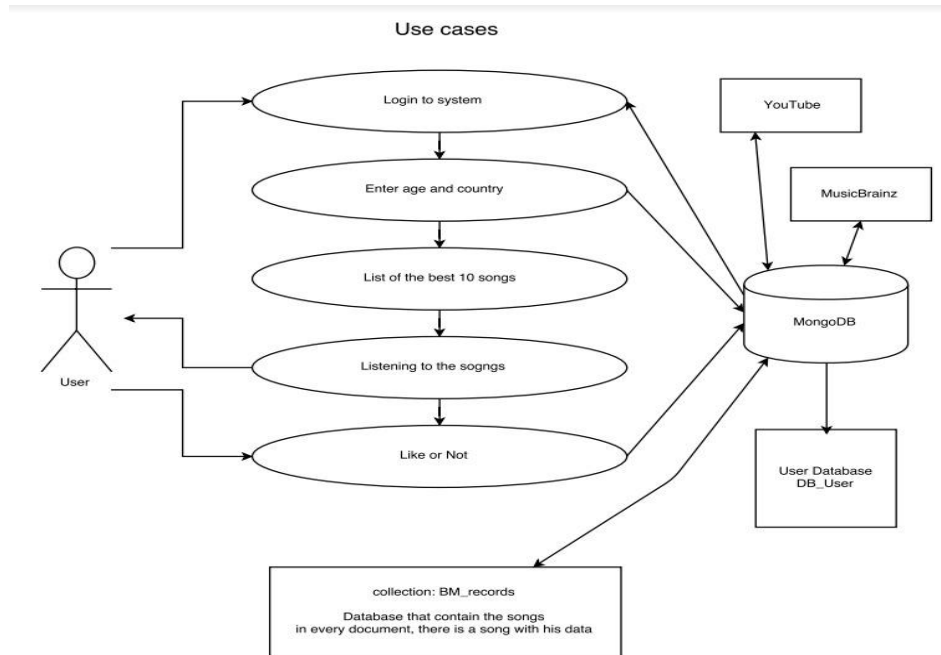
ה MusicBrainz גם כן מגביל את כמות הבקשות שניתן לבצע אל השרת שלו. בכל ששן ניתן לבצע עד 16 בקשות, מה שלא הספיק. המטרה שלנו הייתה לקבל בכל פעם 150 אלף שירים, ובעקבות ההגבלה יכלנו לקבל 100 שירים בכל בקשה, סך הכל 1600, וזאת הסיבה שהיינו צריכים לבנות את Databases של המערכת מראש.  
[https://musicbrainz.org/doc/XML\\_Web\\_Service/Rate\\_Limiting](https://musicbrainz.org/doc/XML_Web_Service/Rate_Limiting)

מאמר העוסק בהגבלה של מספר הבקשות שניתן לבצע ל API של תוכנות של גוגל, הכוללות את YouTube ב"ן היתר  
<https://books.google.co.il/books?id=LUAlDwAAQBAJ&pg=PA46&lpg=PA46&dq=youtube+api+request+limit&source=bl&ots=zV6IJUGs-&sig=SIUoHTfgV33vtPWBJCXHn4bsg&hl=iw&sa=X&ved=2ahUKEwilz9CugoXZAhUCVywKHZAJAYg4FBD0ATAHegQIBhAB#v=onepage&q=youtube%20api%20request%20limit&f=false>

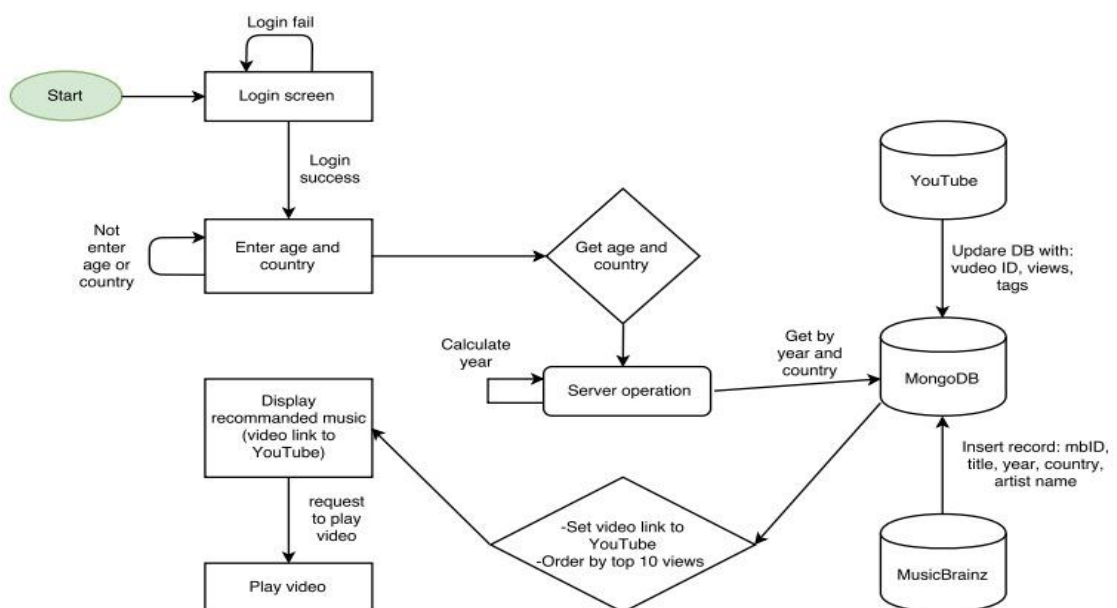
github מגביל את משקל הקבצים שניתן להעלות אליו עד 100 מגה, ולכן השתמשנו ב dump כדי להעלות את Databases שלנו  
<https://help.github.com/articles/working-with-large-files>

## ב. תרשימים וטבלאות

:Use cases



: State chart





### ג. תכנון הפרויקט

<<פגישת הכרות עם המנחה >>	21.7.2017
פגישת הכרות עם הלקוחות , לאה וסטלה.	3.8.2017
פגישה עם המנחה ועם הלקוחות	20.8.2017
פגישה עם המנחה וקביעת משימות	18.9.2017
הגשת שלב ההצעה הכולל סקר שוק	19.11.2017
הצגת פונקציות לשליפת המידע הרלוונטי	20.12.2017
הגשת דוח אלפא	01.02.2018
בדיקות מקרי קצה	
סיום שלב בניית המערכת- דו"ח בנייה	
מסירת הפרויקט	

### ד. טבלת סיכונים

דרגת החומרה : 1 : לא חמור בכלל, 5 : חמור ביותר

#	הסיכון	חומרה	מענה אפשרי
1	אי התאמה של youtube api לצרכי הפרויקט	5	שימוש בפלטפורמה אחרת
2	אי התאמה של musicbrainz api לצרכי הפרויקט	5	שימוש בפלטפורמה אחרת
3	אי התאמה של last.fm api לצרכי הפרויקט	1	שימוש בyoutube
4	אי התאמה של discogs api לצרכי הפרויקט	1	שימוש בmusicbrainz
5	בחירת קטגוריות שאלון לא נכונה	3	שינוי שאילתות ומידע נשלף
6	אי אפשרות של ניגון youtube באתר נפרד	4	בדיקת זכויות יוצאים
7	שמירת מסד הנתונים על ידי שרת לא איכותי ולא מהיר	5	החלפת השימוש במנוגנו לשרת מהיר יותר
8	ממשק לא ידידותי מספיק למשתמש	3	שינוי תצוגה והתאמה
9	אי מציאת מדיה להשמעה המתאימה לנתונים	5	אי הכנסה של השיר למסד הנתונים
10	אלגוריתם שליפה איטי	4	החלפה לאלגוריתם מהיר יותר
11	למידת הטכנולוגיות api ושליפת המידע	2	מחקר רציני בנושא
12	עדכון גרסה לאחד מסוגי api	1	התאמת השאילתות לגרסה החדשה
13	אינדקס שירים לא מעודכן בlast.fm	2	שימוש בyoutube
14	אי מציאת מאגר המתאים לנתוני המשתמש	4	ניסיון חיפוש לפי מדינה קרובה או גיל קרוב אחר
15	שינוי אופי שימוש התוכנה ומטרתה	1	התאמתו מחדש למטרתו החדשה
16	נפילה של מיוזיקברינו, יוטיוב	5	שימוש בפלטפורמה אחרת
17	שרת node.js לא מצליח לעלות(פורט)	3	בדיקה שהחיבור אכן הצליח, אחרת נצא בהודעת שגיאה
18	קבלת הנתונים מMusicBrainz ארוכה מדי	4	נחשוב לבצע את קבלת הנתונים מראש, או דרך אחרת יעילה יותר

19	שהלינקים מYouTube לא מתאימים למה שביקשנו	3	נבצע בדיקת תאימות בין הלינק לשמו על מנת לוודא. אחרת לא נציג את הלינק
----	--	---	--

#### ה. רשימת טבלת דרישות

פורמט טבלת הדרישות יהיה לפי המקובל בארגון. להלן דוגמא:

#### טבלת דרישות (User Requirement Document)

מס' דרישה	תיאור
1	מערכת קלה ונוחה לתפעול.
2	המערכת תתבסס על ממשק web.
3	המערכת תתבסס על השפה ג'אווה סקריפט.
4	תשתמש ב node.js לצד שרת
5	תשתמש בmongo להחזקת מסד נתונים
6	המערכת תאפשר הזנת נתונים של המשתמש.
7	המערכת תלמד את גילו, ארץ מוצאו של המשתמש ובהתאם תציג רשימת השירים המתאימים עבורו.
8	המערכת תציג את תוכן המדיה באופן ברור וקל להשמעה.
9	המערכת תנגן את תוכן המדיה מבלי לצאת לאתר אחר או תוכנה אחרת.
10	המערכת תציג את תוכן המדיה בזמן קצר שלא יעלה על מספר שניות
11	במקרה ונתוני הלקוח לא נמצאים, המערכת תבקש מהלקוח להמתין לקבלת כל הנתונים ורק לאחר מכן תציג אותם
12	המערכת תאפשר למשתמש לסמן האם תוכן המדיה ששמע מתאים או לא מתאים (לייק או דיס-לייק) בשבילו.