**המחלקה להנדסת תוכנה**

**פרויקט גמר**

**תמרינגה – מערכת המלצה למוסיקה**

**חיבור זה מהווה חלק מהדרישות לקבלת**

**תואר ראשון בהנדסה**

**מאת**

**דוד מילשטיין**

**יונה מילשטיין**

**סיוון, תשע"ט יוני, 2019**

**המחלקה להנדסת תוכנה**

**תמרינגה**

**חיבור זה מהווה חלק מהדרישות לקבלת**

**תואר ראשון בהנדסה**

**מאת**

**דוד מילשטיין**

**יונה מילשטיין**

**מנחה אקדמי: דר' מרים אללוף אישור: תאריך:**

**אחראי תעשייתי: גב' סטלה מלניצר ולאה כהן סבן אישור: תאריך:**

**רכז הפרויקטים: דר' אסף שפיינר אישור: תאריך:**

**תקציר**

פרויקט גמר זה הוא פרויקט עבור הסטארט-אפ תמרינגה.

הפרויקט הוא פרויקט המשך של לאה כהן סבן, אבי כהן ושגיא מרציאנו ומתבסס על מחקריהם.

תמרינגה הינה פלטפורמה היוצרת גירוי ושמע מותאמים אישית לחולי דמנציה ובכך משפרת את מצבם.

במהלך הפרויקט, בנינו מערכת המלצת מוזיקה הפועלת לפי קריטריוני המשתמש, דמיון בין משתמשים ודירוג המשתמשים כך שהמערכת תציע מוזיקה המותאמת באופן אישי.

המערכת מכילה פלטפורמה למחקר, כך שחוקר המשתמש במערכת יוכל ליצור מחקר ולעקוב אחר משתמשים.

בדוח זה נציג את המערכת, את המימוש ואת אופן השימוש בה, וכן את המסקנות.

קישורים למערכות ניהול הפרויקט ובקרת תצורה:

|  |  |  |
| --- | --- | --- |
| # | מערכת | מיקום |
| 1 | מאגר קוד | <https://github.com/DavidMilstein/TamaringaProject> |
| 2 | יומן | <https://trello.com/b/fnRKWKDH/meetings> |
| 3 | ניהול משימות | [https://trello.com/tamaringa](https://trello.com/tamaringa/home) |
| 4 | סרטון גרסת אלפא | <https://bit.ly/2SjMHJQ> |
| 5 | סרטון פרויקט גמר | <https://www.youtube.com/watch?v=Uo8IMCd1DOM> |

הצהרה:

הפרויקט נעשה בהנחיית ד"ר מרים אללוף במחלקה להנדסת תוכנה, עזריאלי – המכללה להנדסה ירושלים.

החיבור מציג את עבודתנו האישית ומהווה חלק מהדרישות לקבלת תואר ראשון בהנדסה.

היקף הפרויקט גדול ומיועד לשני סטודנטים באישור המנחה האקדמי.

**תודות**

נבקש להודות לאנשים שעזרו לנו ותמכו בנו לאורך ביצוע פרויקט הגמר:

**לד"ר מרים אללוף** - המנחה האקדמית לפרויקט. תודה על העזרה, התמיכה, הסבלנות והדחיפה לאורך הדרך**.**

**לאה כהן סבן וסטלה מליצר** - מייסדות מיזם תמרינגה.

תודה על הזכות שנתתן לנו להיות חלק מהפרויקט המדהים והחשוב הזה, תודה על העזרה, התמיכה, ההסברים, הפגישות והזמינות.

**לשגיא מרצינאו** – מחבר עבודת המחקר שעלייה התבססנו. תודה על הזמן שהשקעת, על הישיבות והסבלנות.

**לד"ר אסף שפיינר** - רכז הפרויקטים. תודה על הסדר והארגון, על הסברים מפורטים וזמין לכל שאלה.

תודה לכם, שהפכתם את חוויית הפרויקט גמר לנעימה יותר. למדנו הרבה מהפרויקט הזה, ושמחנו להיות חלק מפרויקט חשוב שנגע לליבנו.

תודה רבה

**תוכן עניינים**

[1. מבוא 8](#_Toc12392782)

[2. תיאור הבעיה 8](#_Toc12392783)

[2.1 אפיון המערכת ומסכים 8](#_Toc12392787)

[3. תיאור הפתרון 12](#_Toc12392788)

[3.1 תיאור המערכת 12](#_Toc12392789)

[3.2 תהליכים ונתוני המערכת 13](#_Toc12392790)

[3.3 מבנה ה-Data Base של MongoDB 13](#_Toc12392791)

[3.4 אלגוריתם ההמלצה 14](#_Toc12392792)

[3.5 תיאור הכלים המשמשים לפתרון 16](#_Toc12392793)

[4. תכנית בדיקות 16](#_Toc12392794)

[4.1 בדיקות פנימיות על ידי המפתחים 16](#_Toc12392795)

[4.2 בדיקות חיצוניות על ידי משתמשים 18](#_Toc12392796)

[5.סקירת עבודות דומות בספרות והשוואה 19](#_Toc12392797)

[6.סיכום \ מסקנות 19](#_Toc12392798)

[7. נספחים 21](#_Toc12392799)

[7.1 תרשימים וטבלאות 22](#_Toc12392800)

[7.2 רשימת ספרות \ ביבליוגרפיה 23](#_Toc12392802)

**מילון מונחים, סימנים וקיצורים**

* **Cosine Similarity** הינו מדד של דמיון בין שני וקטורים. חישוב זה משמש למדידת דמיון בין משתמשים על סמך הדירוג שלהם במערכות המלצה. בפרויקט זה נעשה שימוש בחישוב זה על מנת לחשב דמיון בין משתמשים אשר דירגו את אותם השירים.
* **YouTube** הוא אתר אינטרנט לאחסון ושיתוף של סרטוני וידאו.
* **MusicBrainz** הוא מאגר מוזיקלי אינטרנטי שמבוסס על קוד פתוח. המאגר מכיל מידע רב על שירים, ומאפשר להשתמש במידע זה באופן ציבורי. מופיע גם בראשי תיבות MB.
* **MongoDB** בסיס נתונים מבוסס NoSQL הנשען על מבנה של מסמך בניגוד לבסיסי נתונים טבלאיים העובדים מעל טבלאות מקושרות. מבנה המסמכים עובד מעל מימוש של JSON  .
  + **Collection** טבלה בבסיס הנתונים.
  + **Document** שדה בתוך הטבלה בבסיס הנתונים.
* **JSON Web Tokens** הוא תקן פתוח מבוסס JSON ליצירת מפתח גישה (Access Token) המשמש לולידציה של "טענות" (לדוגמה שם משתמש, הרשאות, סיסמה). לדוגמה, שרת יכול ליצור מפתח הטוען "בוצעה כניסה כמשתמש במערכת" ולספק  את המפתח ללקוח. הלקוח יכול לאחר מכן להשתמש במפתח כחתימה המאשרת שהוא אכן משתמש במערכת. המפתחות חתומים על ידי מפתח ייחודי של השרת, כך שהלקוח והשרת מסוגלים כל אחד בנפרד לוודא שהמפתח לגיטימי.
* **AWS** טכנולוגיה המאפשרת לשכור מיקום אחסון ממוחשב, אשר זמין בכל עת באמצעות האינטרנט
  + **S3 AWS** השרתהמאחסן את צד הלקוח של הפרויקט – Angular App
  + **Elastic Beanstalk** השרת המאחסן את השרת של הפרויקט Node JS –
  + **EC2 – Elastic Compute Cloud** שרת המאפשר להריץ מכונות וירטואליות בגדלים וסוגים שוניםלצורך חישובים כבדים ולאחסון מידע בצורה מאובטח

# **מבוא**

# 

דמנציה (שיטיון) היא מחלה בה נפגעים התפקודים הקוגניטיביים והמנטליים.

מחלת הדמנציה פוגעת באופן הדרגתי בזיכרון, ביכולת החשיבה, בהתמצאות בזמן ובמרחב, וביכולת לזהות אנשים וחפצים.  
לפי מחקרים רבים בנושא, מצאו קשר ישיר בין העלאת זיכרונות ילדות לחולי הדמנציה מעכבת את התפתחותה של מחלת הדמנציה.

נמצא קשר ישיר בין העלאת זיכרונות עבר להשמעת מוזיקה מהעבר, ומכאן נולד המיזם תמרינגה.  
המערכת אותה נפתח, נועדה להאט את התקדמותו של מחלות הדמנציה וביניהם המחלה המוכרת אלצהיימר, מאחר וזיכרון מוזיקלי וכן גם שירים הם בין הזיכרונות האחרונים אשר נפגעים אצל החולים.

מכאן, המערכת מציעה פלייליסט מוזיקה המותאם לכל משתמש באופן אישי על פי אלגוריתם המלצה להפקת התוצאות המקסימליות.

# **תיאור הבעיה**

## הבעיה המרכזית שצריך להתמודד איתה היא העובדה שבשביל להעלות זיכרונות מהעבר לחולה צריך שמישהו ישקיע מזמנו האישי ויהיה בנוכחותו של החולה שעות רבות במהלך היום, פתרון זה דורש משאבים יקרים – כסף, זמן ואף סבלנות רבה.

## כיום לא קיימת שום מערכת המלצה בשוק אשר מתאימה מוזיקה לגיל השלישי לפי הפרמטרים הרלוונטיים לחולי הדמנציה אלא רק מערכות המלצה המתאימות שירים על פי סגנון המוזיקה, פופולריות ועוד. על אף שמערכות אלו, לדוגמה YouTube, מעולות ומתקדמות מאוד הן פשוט לא מתאימות לקהל היעד שלנו. בנוסף מערכות מתקדמות כמו YouTube נחשבות לא נוחות עבור קהל היעד אליו אנו מכוונים, אנו רוצים מערכת שתהייה נוחה וידידותית למשתמש. כיום, בכדי לבחור פלייליסט מתאים לחולה דמנציה, נדרש בנאדם נוסף אשר יבחר לו את השירים אחד אחד וגם אז לא בטוח יקלע בשירים הנכונים עבור האפקט אליו אנו מכוונים.

## נרצה לפתח מערכת נוחה לשימוש, שתדע להמליץ בצורה הכי נכונה שירים המותאמים באופן אישי לכל משתמש. כמובן שהמערכת צריכה להיות פשוטה לשימוש ככל האפשר, ומותאמת לגיל השלישי.

דרישה נוספת שלנו בפרויקט היא היכולת להציג ולנגן, תוך כדי הישארות במערכת שלנו ולא לצאת למערכת חיצונית. כל שיר שעלה כתוצאה מהרצת המערכת יוצג כסרטון בפורמט של YouTube.

## 

## **אפיון המערכת ומסכים**

משתמשי המערכת הם:

* קשישים
* חוקרים
* מנהלי המערכת

**מטרת הפרויקט**

מערכת המלצה של מוזיקה, לפי קריטריוני המשתמש, שתיעזר בדמיון בין משתמשים ודירוג המשתמשים כך שהמערכת תציע מוזיקה המותאמת באופן אישי.

המערכת צריכה להיות נוחה לשימוש עבור קשישים ולהתאים לצרכיהם – הסברים ברורים ובעברית, כתב גדול.

מתן אתר ניהולי נוח לחוקר על מנת לאפשר ניהול של מחקר.

המערכת תהיה זמינה מכל מחשב ומכשיר עם חיבור לאינטרנט.

**הרשאות לפי בעלי תפקידים במערכת**

אדמין/מנהל המערכת – יכול ליצור משתמשים חדשים של קשישים וחוקרים.

חוקר – יצירת מחקר חדש וצפיה במחקרים קיימים. המחקר נועד לעקוב אחר השימוש במערכת, ולבצע ניסויים עם הקשישים שמשתמשים במערכת.

קשיש/חולה הדמנציה – לאחר קבלת שם משתמש וסיסמה מהמטפל, הקשיש יוכל להתחבר בכל עת למערכת ולאחר ההתחברות יוצג בפניו פלייליסט חכם של שירים (וסרטונים) אשר הותאם עבורו אישית, באפשרותו לדרג כל שיר ושיר עד כמה נהנה ממנו.

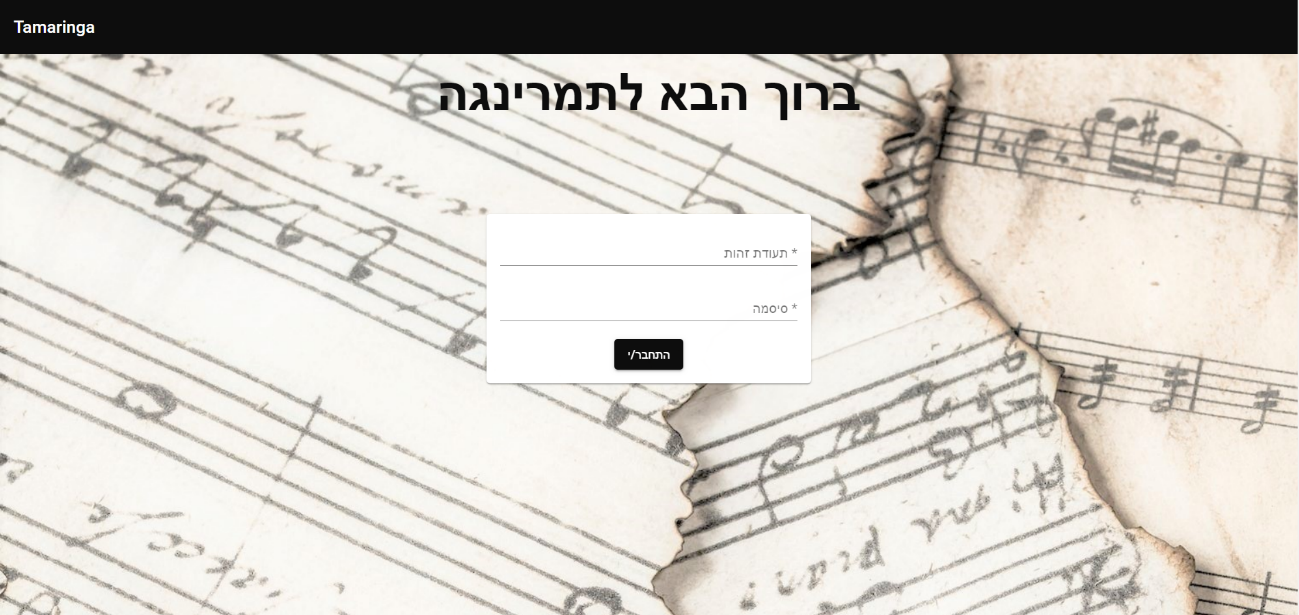
**מסכים:**

**מסך כניסה למערכת** – אין צורך בבחירה במשתמש/חוקר/אדמין, המערכת מזהה לבד איזה משתמש נכנס למערכת ומביאה אותו למסך המתאים.

כאשר אדמין נכנס למערכת, הוא יוכל ליצור משתמשים עבור חוקרים וקשישים.

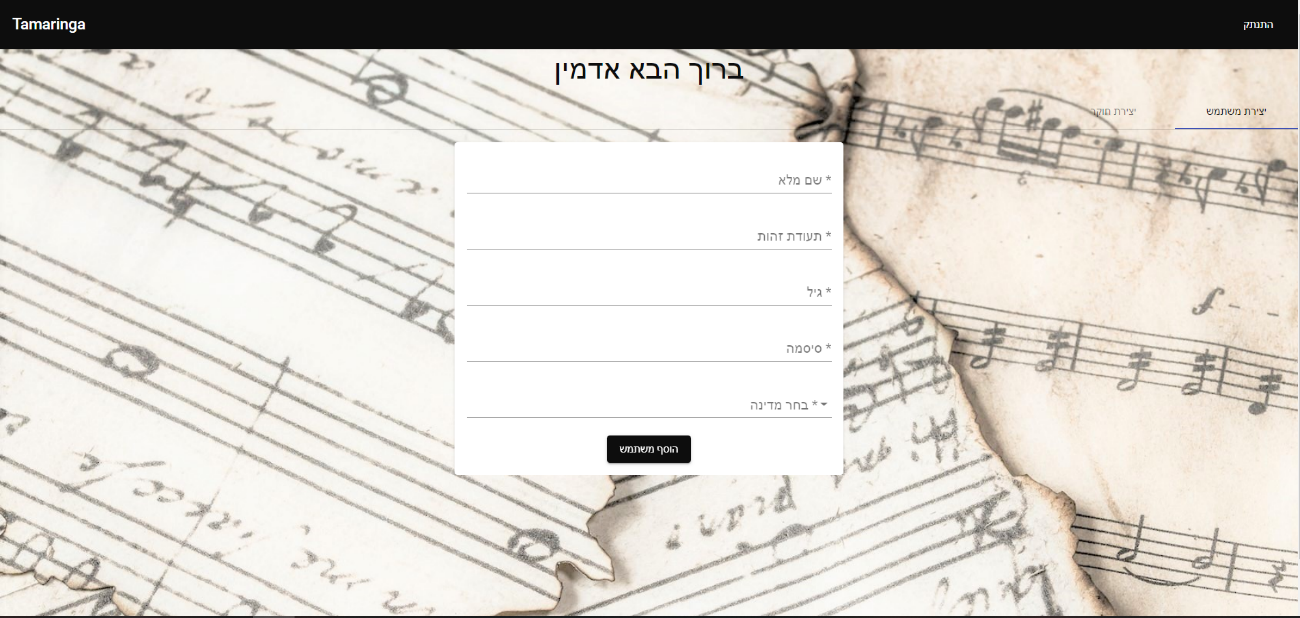
כאשר קשיש נכנס למערכת, הוא יקבל פלייליסט מותאם באופן אישי.

כאשר החוקר יכנס למערכת הוא יוכל ליצור, לצפות ולערוך מחקרים.

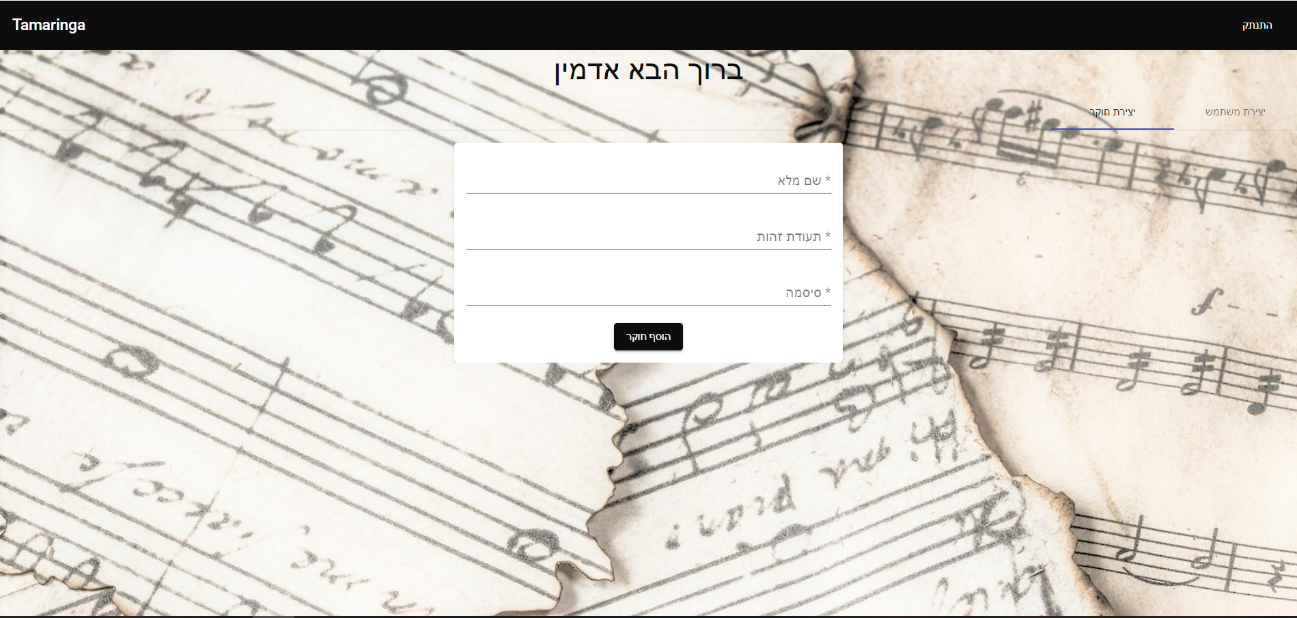


**מסך אדמין – יצירת משתמש**

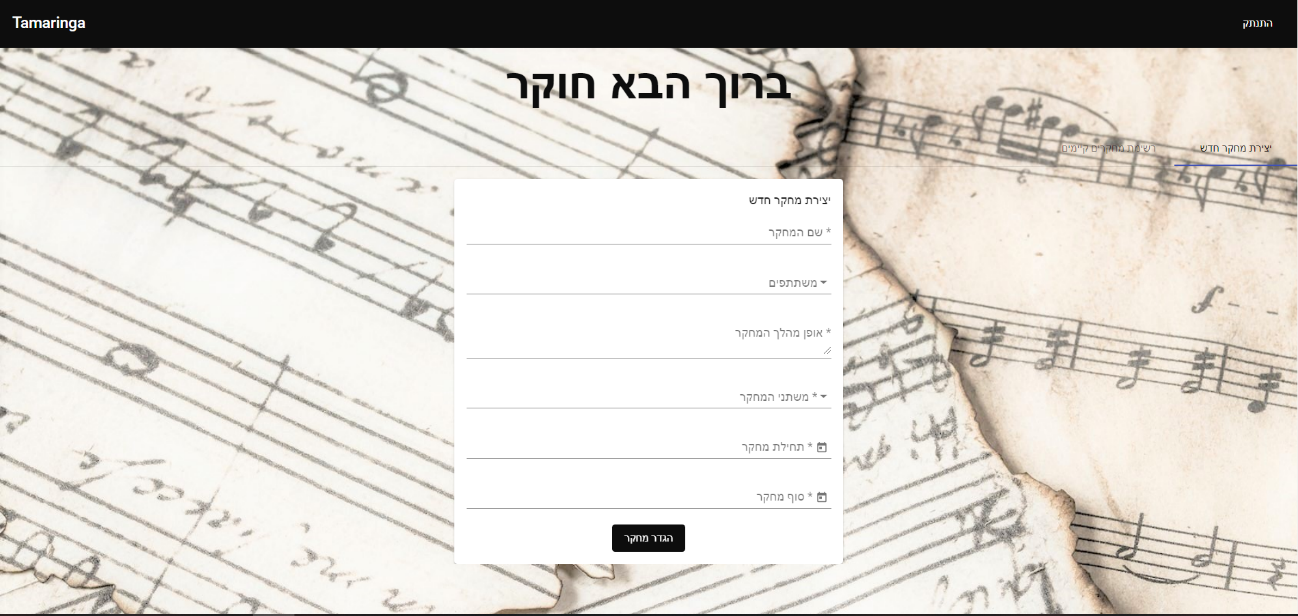
כאשר יוצרים משתמש, המערכת בונה עבורו פלייליסט שיישמר בDB, כך שבפעם הבאה שהוא יתחבר, הפלייליסט כבר יהיה קיים והשליפה תהיה מהירה.

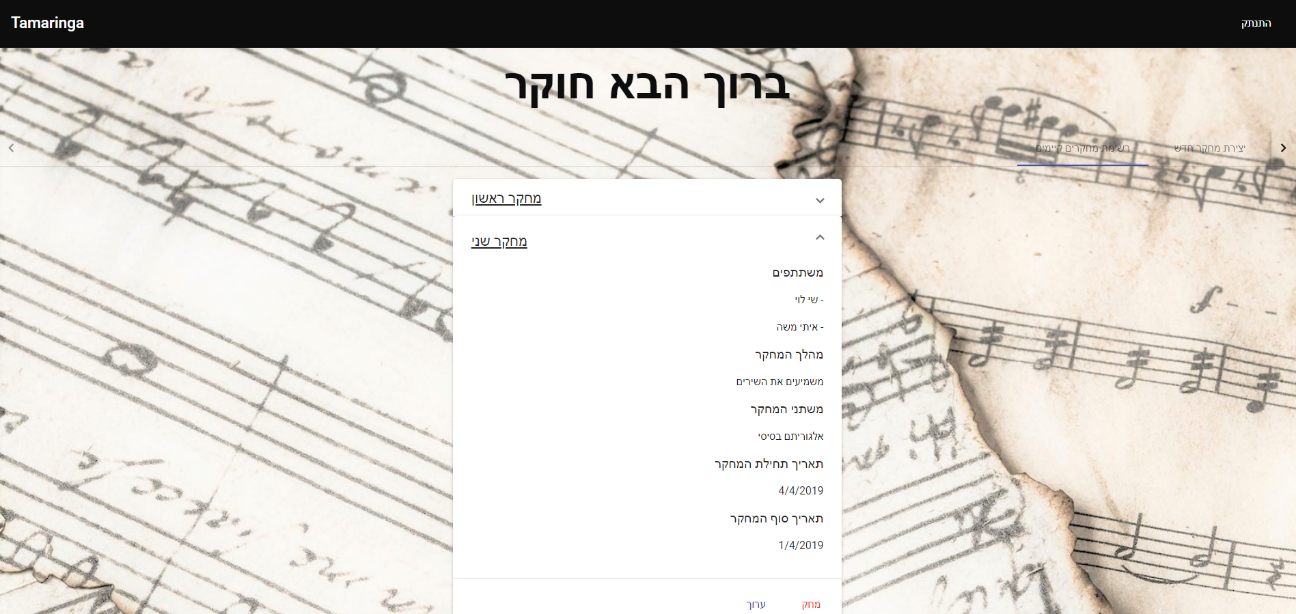


**מסך אדמין – יצירת חוקר**



**מסך חוקר – יצירת מחקר חדש**



**חוקר – רשימת מחקרים.**

**מסך משתמש** – בעת כניסתו הראשונה למערכת, יוצג פלייליסט המורכב מ30 שירים, שימצאו מתאימים על פי קריטריונים שונים: המדינה ממנה המשתמש הגיע, גילו ושירים פופולריים לאותה תקופה. השירים יהיו משנות ה20 של הקשיש 3 ± שנים.

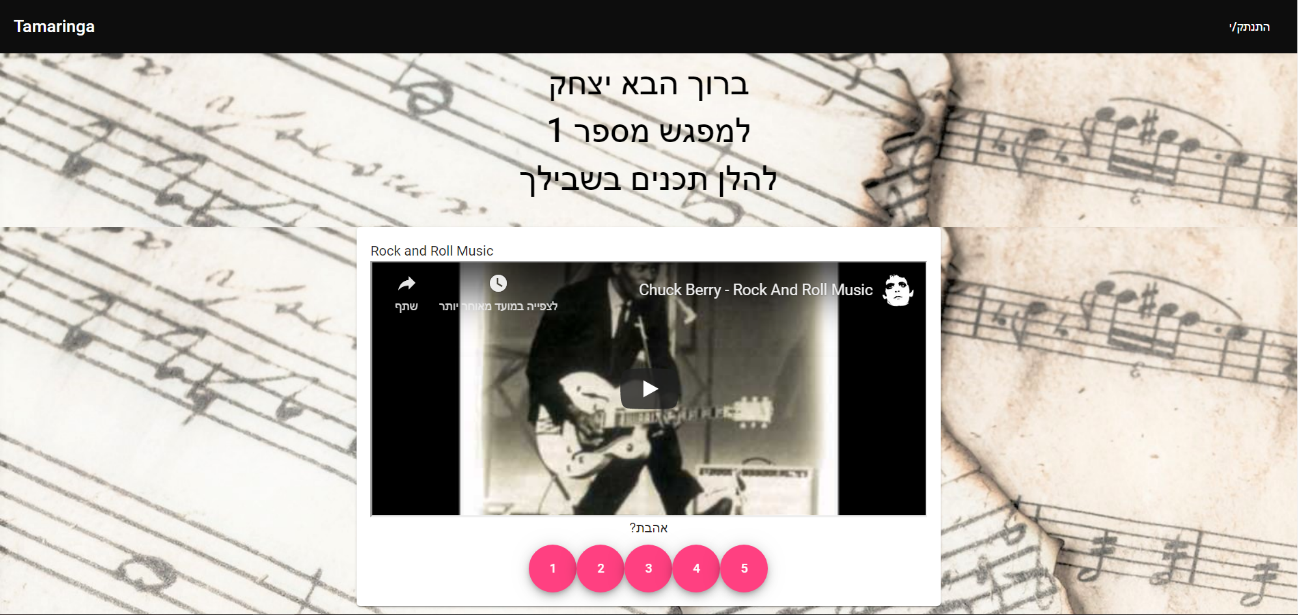
למשתמש ישנה אפשרות לדרג כל שיר מ1 עד 5, כאשר המשתמש מדרג שיר מסוים, אנו שולחים את הדירוג לDB ובמצעים חישוב cosine-similarity שמטרתו היא לחשב את הדמיון בינו לבין שאר המשתמשים בקבוצה.

בפעמים הבאות שהוא יכנס למערכת, הפלייליסט יהיה בנוי כך:

10 שירים שהוא אהב (דירוג 5 ו4)

10 שירים שדומים שמשתמשים דומים לו אהבו.

10 שירים אקראיים המתאימים לפרופיל שלו שהוא עוד לא דירג.



* 1. **הבעיה מבחינת הנדסת תוכנה**

המערכת כוללת צד לקוח (Angular) צד שרת (Node.js & Express.js ) ומסד נתונים (MongoDB & mongoose) שגם הוא חלק מצד השרת, בנוסף המערכת תומכת בניהול משתמשים עם הרשאות שונות לחלוטין.

הפרויקט מספק אתגרים בתחום מערכות ואלגוריתמי המלצה, המבוססות על קשר בין שני תכנים דומים. על מנת לבנות מערכת המלצה שתעבוד בצורה מדויקת ויעילה, קיים צורך להבין איך לקשר בין תכנים דומים.   
על מנת לבנות Data Base במערכת המלצה ולהשתמש בה ביעילות כבסיס לה, לא ניתן להסתפק רק בלינקים של התכנים (מתוך שמות השירים נמצא את הלינקים המתאימים להם בYouTube), אלא גם במידע אודות התוכן עצמו. מחקר מעמיק ב- API YouTube מראה כי הוא לא מספק מידע רחב אשר יכול לסייע באפיון תוכן המדיה ובניית מערכת ההמלצה.

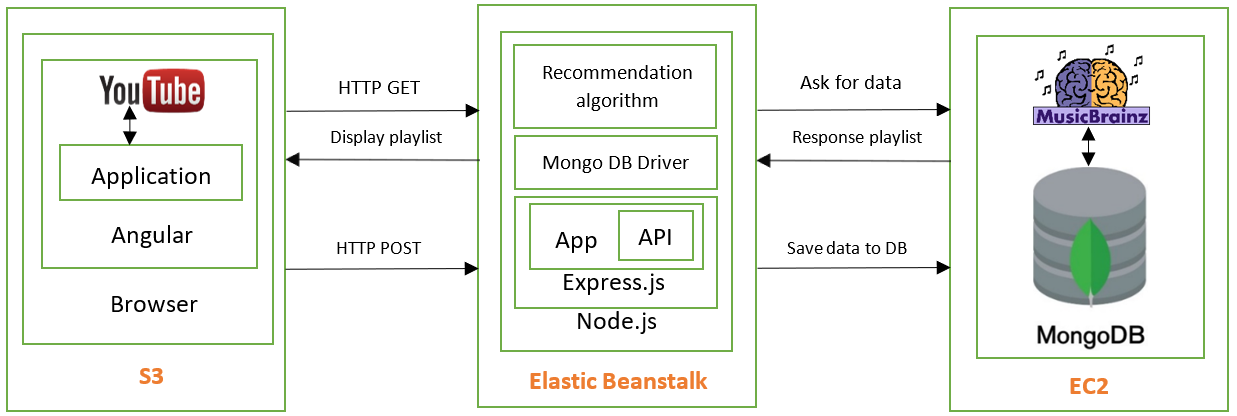
מערכת לניהול משתמשים, קיימים היום כלים מובנים רבים לניהול משתמשים כדוגמת Firebase, הבעיה שרובם מבוססים על חשבונות של חברות ענק כמו Google, Apple, Facebook ועוד רבים, אנחנו יוצאים מהנחה שאוכלוסיית היעד שלנו אינה צורכת מידע דרך החשבונות הללו ולכן אנחנו חייבים לפתח מערכת לניהול משתמשים מלאה מבוססת JSON Web Tokens.

שימוש נרחב בAWS: אחסון צד לקוח בS3, צד שרת בElastic Beanstalk וDB בEC2.   
כמו כן, קיים הצורך לאפיין את מבנה ה Data Base עבור המערכת. באופן כללי, הפרויקט כולל לא מעט אינטגרציות בין כלים ומערכות שונות וחדשות עבורנו וכאן יהיה עיקר המאמץ.

# **תיאור הפתרון**

## **תיאור המערכת**

**ארכיטקטורת המערכת:**



הארכיטקטורה מתארת את מבנה המערכת אשר מבוססת Web.   
אינטראקציות במערכת:  
   - בין Angular App (צד לקוח) לבין הREST API  המבוסס Node.js (צד שרת)

   - בין Node.js לMongoDB

   - בין Angular App לYouTube

המערכת מאוחסנת בשלושה שרתים שונים של AWS:

S3 -    עבור Angular App

   - Elastic Beanstalk עבור השרת

   - EC2 עבור הData Base

## **3.2 תהליכים ונתוני המערכת**

התרשים לעיל מתאר את מצבי המערכת.

## **3.3 מבנה ה-Data Base של MongoDB**

כאמור, אנו ממשיכים את הפרויקט של אבי ושגיא, בזמנו הם עשו מחקר מקיף ונודע להם שהשימוש במאגר השירים MusicBrainz, שלפו כמויות מידע גדולות מהמאגר ואחסנו אותם בDB מקומי כ Documents בפורמט של JSON כך שיתאימו לשימוש ב.MongoDB

כל שיר במערכת הינו Document נפרד בתוך Collection של שירים.  
כיום ישנם בDB כ130,000 שירים שונים מרחבי העולם משנים בין השנים 1900-2010.

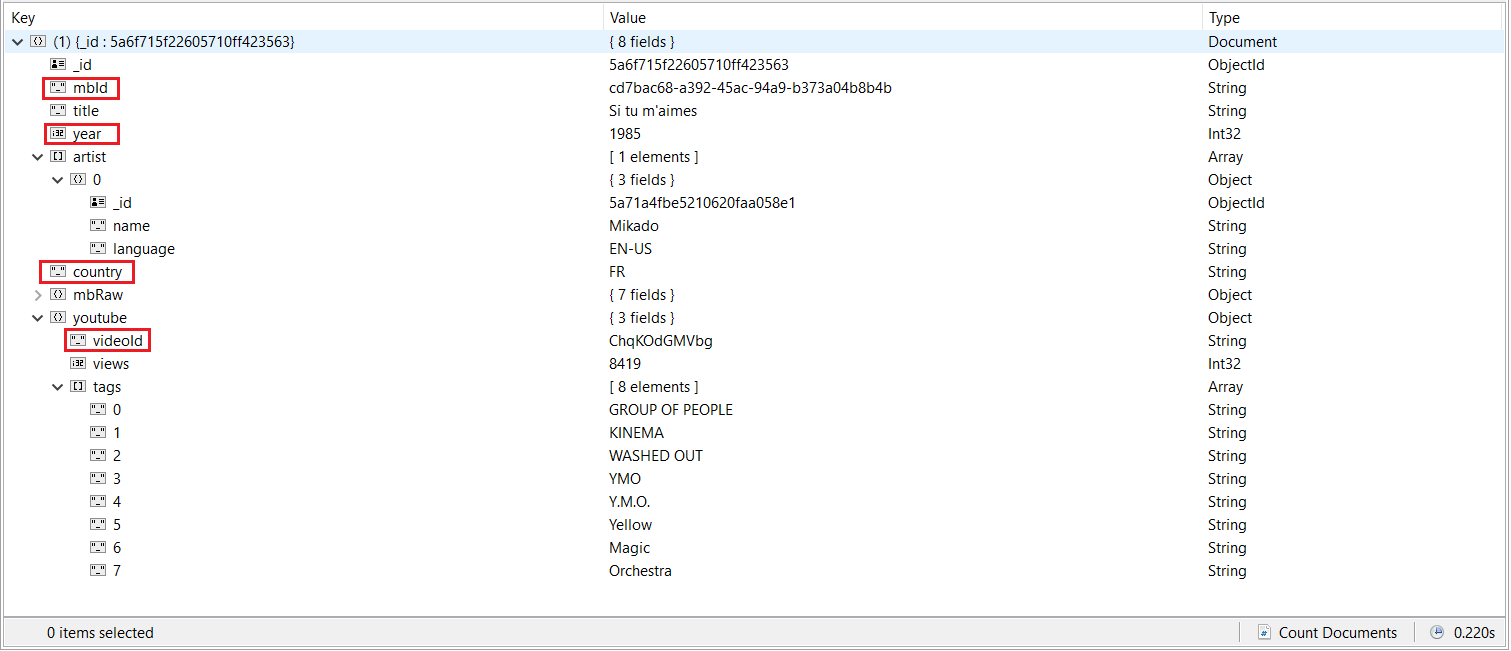
בDB קיימים 4 Collections שונים: users, records, playlists, researches.

נסקור שניים מהם:

להלן צילום מסך המציג מבנה של שיר (record) מסוים בDB תחת Collection של records:

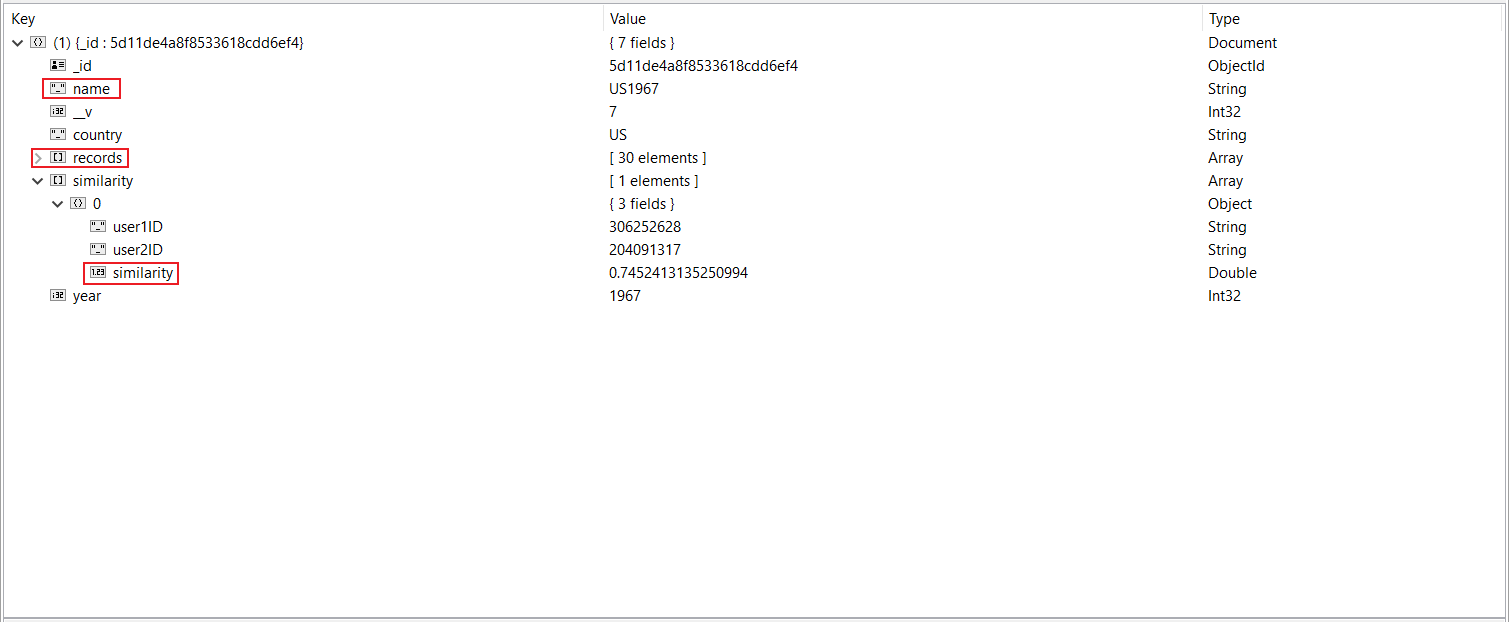
השדות בהם נעשה השימוש העיקרי הם mbId – מזהה השיר מטעם MusicBrainz,

year – שנת ההוצאה, country – ארץ מוצא, videoId – מזהה השיר מטעם YouTube



להלן צילום מסך המציג מבנה של פלייליסט מסוים בDB תחת Collection של playlists:

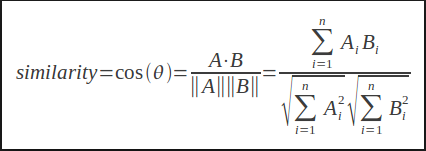
השדות בהם נעשה השימוש העיקרי הם name – אשר מורכב מקוד המדינה והשנה בהם השירים יצאו, records - הינו Collection בפני עצמו של שירים (records) וsimilarity – תוצאת חישוב הדמיון בין שני משתמשים אשר דירגו את אותו השיר.



## **3.4 אלגוריתם ההמלצה**

אלגוריתם ההמלצה מבצע שליפה של תכני מדיה מתוך הDB על פי ארץ מוצא והשנה בה המשתמש היה בשנות ה20 לחייו 3 ±שנים וכן שירים בעלי דירוג פופולאריות גבוה ביותר (על פי צפיות בYouTube).   
בנוסף, האלגוריתם מחשב דמיון בין משתמשים אשר חולקים את אותו הפרופיל (ארץ מוצא ושנת לידה) בכל פעם שמשתמש מסוים דירג את אחד השירים שהוצעו עבורו מופעל האלגוריתם Cosine Similarity אשר מחשב את הדמיון בין שני משתמשים בעל פרופיל דומה, כך שבפעם הבאה שהמשתמש יתחבר, המערכת תמליץ עבורו שירים נוספים שהוא עשוי לאהוב.

Cosine Similarity מתבסס על הנוסחה המתמטית הבאה:



תוצאת חישוב האלגוריתם תהייה תמיד בין 0 ל1, ככל שהתוצאה קרובה יותר ל1 כך גם הדמיון קרוב יותר.

**נכונות האלגוריתם**

נבצע ניסוי המתאר את הפונקציונאליות של המערכת ונראה את נכונותה.

במערכת רשומים 3 משתמשים בעלי פרופיל זהה, כלומר נולדו באותה השנה והגיעו מאותה המדינה ולכן בעת ביצוע הclustering , המערכת שייכה אותם לאותה קבוצה קבוצה, עבור שלושתם נשלפה רשימה של 25 שירים בהתאם לקבוצה שלהם, במקרה שלנו, המשתמשים אליס, בוב וצ'ארלי בני 75 מארה"ב וקבוצתם.

להלן רשימת 25 השירים השייכים לplaylist המשותף של הקבוצה US1964 וכן דירוג של חלק מהשירים על ידי שלושת המשתמשים:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **צ'ארלי** | **אליס** | **בוב** | **שם השיר** |  |
| 4 |  |  | California Dreamin’ | **1** |
|  | 4 | 5 | Oh, Pretty Woman | **2** |
| 4 |  |  | (I Can’t Get No) Satisfaction | **3** |
|  |  |  | Bring It On Home to Me | **4** |
|  | 3 |  | Blowin’ in the Wind | **5** |
|  |  | 5 | The Times They Are A-Changin’ | **6** |
|  | 3 |  | Mr. Tambourine Man | **7** |
|  |  |  | A Change Is Gonna Come | **8** |
|  | 4 |  | I’ve Been Loving You Too Long | **9** |
| 5 |  |  | Boom Boom | **10** |
|  |  |  | Moon River | **11** |
|  |  |  | Scarborough Fair / Canticle | **12** |
| 3 |  |  | Puff, the Magic Dragon | **13** |
|  | 4 | 4 | You’ll Lose a Good Thing | **14** |
|  | 5 |  | You Never Can Tell | **15** |
|  | 5 |  | Be My Baby | **16** |
|  | 3 | 3 | Twistin’ the Night Away | **17** |
|  |  | 4 | These Arms of Mine | **18** |
| 5 |  | 5 | You Really Got Me | **19** |
| 3 |  | 3 | The Loco‐Motion | **20** |
|  |  | 4 | Baby, I Love You | **21** |
| 5 | 2 |  | Bus Stop | **22** |
| 2 |  | 2 | Wipe Out | **23** |
| 5 |  | 5 | Early Mornin' Rain | **24** |
| 3 | 2 |  | Baby I Need Your Loving | **25** |

לאחר שבוב, אליס וצ'ארלי דירגו את שירהם מ1 עד 5, המערכת חישבה בעזרת הCosine Similarity את ציוני ההתאמה בינהם:

|  |  |
| --- | --- |
| **תוצאת חישוב Cosine Similarity** | **משתמשים עליהם בוצע החישוב** |
| 0.29926934464747973 | אליס ובוב |
| 0.10866767118696098 | בוב וצ'ארלי |
| 0.3784620056424481 | אליס וצ'ארלי |

נשים לב כי לפי תוצאות חישוב ההתאמה בין שלושת המשתמשים, התוצאה בין אליס וצ'ארלי היא הגבוהה ביותר, שזה אכן מה שציפינו לקבל מאחר והדירוגים של אליס וצ'ארלי הם הקרובים ביותר ביחס לשאר המשתמשים.   
לכן בפעם הבאה שבוב או צ'ארלי יתחברו, המערכת תמליץ להם על שירים שהאחר אהב.

נראה זאת על ידי התחברות של אליס ונראה איזה שירים היא קיבלה, בנוסף לשירים שצ'ארלי אהב אנו מצפים לראות גם שירים שאליס עצמה דירגה ואהבה :

|  |  |
| --- | --- |
| **סיבת ההמלצה** | **שם השיר** |
| שניהם אהבו | Early Mornin' Rain |
| המלצה מצ'ארלי | Bus Stop |
| המלצה מצ'ארלי | California Dreamin’ |
| המלצה מצ'ארלי | Boom Boom |
| אליס אהבה | The Times They Are A-Changin’ |
| שניהם אהבו | You Really Got Me |
| אליס אהבה | Oh, Pretty Woman |
| צ'ארלי אהב | (I Can’t Get No) Satisfaction |
| שיר חדש שאליס עוד לא שמעה בעבר | Be My Baby |
| שיר חדש שאליס עוד לא שמעה בעבר | Bring It On Home to Me |

## **3.5 תיאור הכלים המשמשים לפתרון**

**בבניית המערכת השתמשנו בטכנולוגיות הבאות:**

**WebStorm** - סביבת פיתוח משולבת (IDE).

**Angular** - פלטפורמת העבודה לתכנות באינטרנט מבוסס TypeScript, תשמש אותנו לצד לקוח.

**Node.js and Express.js** - עבור שרת המערכת.

**MongoDB and mongoos**e - עבוד מסד הנתונים של המערכת.

**Trello** - כלי לניהול פרויקטים, הן למשימות והן לניהול יומן הפגישות לאורך הפרויקט.

**GitHub** - כלי לניהול גרסאות ואחסון קוד הפרויקט.

**Studio** **3T** - כלי מקומי עבור מסד הנתונים כך שיהיה אפשר לראות ולערוך אותו באופן ויזואלי.

**AWS** - אספקת שירותי ענן לאחסון ממוחשב עבור מערכות הפרויקט השונות.

# **תכנית בדיקות**

## **4.1 בדיקות פנימיות על ידי המפתחים**

בסיום פיתוח כל פיצ'ר במערכת או מיזוג קוד חדש עם קוד ישן, יבוצעו מספר בדיקות תקינות שימולאו בהתאם בטבלאות השונות.

**בדיקות צד לקוח**

* תקינות קלט תקין ואינו תקין שהוכנס על ידי המשתמש.
* ממשק משתמש ויזואלי, מסכי המערכת.
* הודעות שגיאה בהתאם לתרחיש.
* שליחת הנתונים לשרת באופן תקין וכן קבלתם.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר הבדיקה | מודול | תיאור הבדיקה | תוצאה צפויה | תקינות הבדיקה |
| 1 | Login | הכנסת פרטי אדמין | מעבר למסך האדמין | ✔ |
| 2 | Login – התחברות למערכת | הכנסת פרטי משתמש | מעבר למסך המשתמש + הופעת פרטי המשתמש הנכונים + הצגת פלייליסט מותאם | ✔ |
| 3 | Login – התחברות למערכת | הכנסת פרטי חוקר | מעבר למסך החוקר | ✔ |
| 4 | Login – התחברות למערכת | הכנסת קלט לא תקין\משתמש שאינו קיים במערכת | תקפוץ הודעת שגיאה על פרטי משתמש לא נכונים | ✔ |
| 5 | מסך המשתמש | היכולת לנגן ולעצור שירים | השיר מתחיל לפעול\עוצר | ✔ |
| 6 | מסך המשתמש | היכולת לדרג שיר | שיר דורג בהצלחה (Debugger) | ✔ |
| 7 | מסך אדמין | הוספת משתמש\חוקר עם קלט תקין | הודעה של משתמש\חוקר נוסף בהצלחה + (Debugger) | ✔ |
| 8 | מסך אדמין | הכנסת קלט לא תקין ביצירת משתמש\חוקר | הודעת שגיאה מתאימה +  (Debugger) | ✔ |
| 9 | מסך חוקר | יצירת מחקר ומחיקתו | הודעה של מחקר נוסף\נמחק בהצלחה  +  (Debugger) | ✔ |
| 10 | Logout – התנתקות מהמערכת | יציאה מהמערכת על ידי לחיצה על "התנתק/י" מכל מקום במערכת | יציאה מהמערכת  +  (Debugger) | ✔ |

**בדיקות צד שרת + אינטגרציה**

* תקשורת תקינה עם הDB – התחברות, שליפה והכנסת נתונים (פעולות C.R.U.D).
* בדיקות פונקציונליות של שלל המטודות בצד השרת.
* קבלת נתונים תקינה מצד לקוח וכן שליחתם ללקוח.
* החזרת הודעות או הודעות שגיאה לצד לקוח.
* כל שרתי הAWS מתקשרים זה עם זה באופן מלא (הבדיקה מתבצעת רק לאחר עדכון/שינוי באחד משלושת השרתים).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מספר הבדיקה | Endpoint  /Action | תיאור הבדיקה | תוצאה צפויה | תקינות הבדיקה |
| 1 | signup | יצירת משתמש | וידוא יצירת משתמש בDB, יצירת פלייליסט בעת הצורך | ✔ |
| 2 | signup | יצירת חוקר | וידוא יצירת חוקר בDB | ✔ |
| 3 | login | כניסת משתמש | וידוא משתמש קיים וסיסמה תקינה + החזרת פלייליסט מתאים ללקוח | ✔ |
| 4 | login | כניסת אדמין\חוקר | וידאו משתמש קיים + סיסמה תקינה | ✔ |
| 5 | get user | בקשה לקבלת נתוני משתמש מהDB | וידוא שהפרטים אכן נשלחים למבקש | ✔ |
| 6 | Create research | יצירת מחקר | נוצר מחקר חדש בDB | ✔ |
| 7 | Delete research | מחיקת מחקר | מחקר נמחק מהDB | ✔ |
| 8 | vote | משתמש מדרג שיר במערכת | ביצוע פעולת cosine-similarity על משתמשים שדירגו שירים מתוך הפלייליסט בעבר ושמירת כל הנתונים בDB | ✔ |
| 9 | העלאה ועדכון צד שרת בBeanstalk AWS | כל הבדיקות שנעשו עד כה [1-8] | תקשורת תקינה בין השרת לבין הלקוח והDB | ✔ |
| 10 | העלאה ועדכון צד לקוח בS3 AWS | כל הבדיקות שנעשו עד כה [1-8] | תקשורת תקינה בין הלקוח לבין השרת וYouTube | ✔ |

## **4.2 בדיקות חיצוניות על ידי משתמשים**

מאחר ופרויקט זה הינו תחת ההגדרה של "פרויקט יישומי" נדרשנו לספק את המערכת למשתמשים אמיתיים על מנת לסקור את חוויתם במערכת.

סיפקנו את המערכת (URL link) לארבעה משתמשים בני הגיל השלישי וזאת לאחר שאנו כמנהלי המערכת יצרנו עבורם משתמשים במערכת בהתאם לפרטיהם.  
  
משתמשת א' - אלה ברק בת 79 מבריה"מ (רוסיה).

משתמש ב' - בובי בירנבאום בן 76 מאנגליה.

משתמש ג' - יצחק הלימי בן 83 ממרוקו.

משתמשת ד' - קלאריס בת 73 מברזיל.

על מנת לבחון את מערכת ההמלצה כראוי, יצרנו ארבעה משתמשי דמה עם פרופילים זהים.

ביקשנו מהמשתמשים להתחבר ולצאת מהמערכת מספר פעמים, לדרג את השירים ולהנות מהמוזיקה.

**להלן הממצאים**

* עוד לפני שביקשנו מהמשתמשת הראשונה להשתמש במערכת גילינו בעיה טכנית.

למרות מאגר השירים היחסית גדול, לא נמצאו שירים מתאימים למשתמשת א' מבריה"מ, לאחר שבדקנו את הנושא מצאנו שMusicBrainz סיפק לנו שירים מבריה"מ בעיקר מ1986 בזמן שהיינו צריכים שירים מהשנים 1957-1963.

ציון: נכשל.

* משתמש ב' התחבר והתנתק מהמערכת בהצלחה, דירג שירים ושיתף פעולה באופן מלא, בזמן שהיינו איתו, שמנו לב שהוא כמשתמש יכול להיכנס למסך החוקר ואף ליצור מחקרים. התקלה טופלה מיד לאחר מכן.  
  ציון: טוב.
* ממשתמש ג' נודע לנו בעיה נוספת, אמנם הוא נולד במרוקו אך בגיל צעיר הוא עלה לצרפת ולכן הפלייליסט שקיבל לא היה רלוונטי כלל, למדנו מזה שמוטב ומנהל המערכת אשר יוצר את המשתמשים יוודא תחילה עם המשתמש באיזו מדינה הוא העביר את שנות ה20.  
  לאחר מכן יצרנו עבורו משתמש נוסף אך הפעם הכנסו שהוא הגיע מצרפת.   
  ציון: טוב.
* משתמשת ד' התלהבה נורא, נהנתה מהשירים ואף אמרה שהחזרנו אותה 60 שנים אחורה!   
  ציון: מצוין.

# **סקירת עבודות דומות בספרות והשוואה**

הפרויקט הוא פרויקט המשך של שנה שעברה שאבי לוי ושגיא מרציאנו עשו.

עבודות דומות:

**YouTube** - היא חברה אמריקאית המפעילה [אתר אינטרנט](https://he.wikipedia.org/wiki/%D7%90%D7%AA%D7%A8_%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98) ל[אחסון ושיתוף סרטוני וידאו](https://he.wikipedia.org/wiki/%D7%A9%D7%99%D7%AA%D7%95%D7%A3_%D7%95%D7%99%D7%93%D7%90%D7%95_%D7%91%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98). היא גם בונה פלייליסט עם שירים שאתה רגיל לשמוע – החיסרון הוא שחייב לשמוע שירים כדי שימליץ לך על פלייליסט התחלתי מתאים.

**Spotify** - הוא שירות מוזיקה דיגיטלי המאפשר הזרמה של שירים ופודקאסטים. הוא ממלית שירים בהתאמה אישית עם אלגוריתם חכם ולומד. החיסרון הוא שהשימוש בתוכנה לא טריוויאלי, לכן לקהל שלנו יהיה קשה להשתמש בספוטיפיי.

# **סיכום \ מסקנות**

במסגרת פרויקט הגמר התבקשנו לפתח מערכת המלצת שירים עבור מיזם תמרינגה.

תמרינגה הינה פלטפורמה היוצרת גירוי ושמע מותאמים אישית לחולי דמנציה ובכך משפרת את מצבם.

על מנת לבנות מערכת המלצה נדרש מאיתנו להבין מספר דברים:

·        מה מטרת מערכת ההמלצה ומה המשתמש מצפה ממנה.

·        מיהו קהל היעד שלנו.

·        מהו מסד הנתונים עליו תתבסס מערכת ההמלצה.

·        מהן השאילתות היעילות ביותר עבור המערכת בשביל חווית משתמש מקסימלית.

·        איך עובדת מערכת המלצה, מהן הסוגים השונים ומה מתאים יותר למערכת שלנו.

·        'להיכנס' לראש של קשיש ולהבין איך ליצור עבורו ממשק שתפיק את שימוש וחווית המשתמש המקסימלית והנוחה ביותר.

·        עבור אילו מכשירים המערכת תתאים? אפליקציה/אתר אינטרנטי.

·        באילו טכנולוגיות נשתמש בשביל להשיג את המטרה.

· כיצד נממש וננהל את מערכת המשתמשים.

על מנת לענות על השאלה הראשונה, ביצענו פגישות עם מנחת הפרויקט ועם מיזם תמרינגה על מנת להבין מהי מטרת מערכת ההמלצה, מהו סוג התוכן שהמשתמש מעוניין לקבל ומה מאפיין את המשתמש.

הבנו כי המשתמש הוא בדרך כלל קשיש ולאו דווקא חולה דמנציה והמידע שנוכל לקבל ממנו עצמו לגבי העדפות מוזיקליות שלו הוא מועט ולא יעיל, על אחת כמה וכמה שאנחנו מכוונים לשירים משנות ה20 של הקשיש, ההמלצות שאנו מעוניינים להציע למשתמש הן שירים שאפיינו את נעוריו וחייו הבוגרים ונמצאים עמוק בתת המודע שלו. לשלב זה של המערכת המידע שנמצא בידנו על אותו קשיש הוא לכל היותר שנת לידה וארץ מוצא. 

לאחר שהבאנו את מטרת מערכת ההמלצה, נדרשנו להבין מה Database שיבסס אותה. ידענו כי המידע שיש לנו על המשתמש שישמש להמלצה הוא לא רב ומכיל בעיקר שנת לידה וארץ ומוצא, ולכן היה חשוב לנו למצוא מידע על מוזיקה שיכלול בין היתר שנת וארץ הוצאה.

למזלנו, השתמשנו הרבה במחקר שאבי ושגיא ביצעו בפרויקט שלהם ולכן הבנו מהר מאוד שאנחנו הולכים להשתמש במאגר השירים שהם השיגו מMusicBrainz ועכשיו תפקידנו היה לבנות את המערכת כך שתתאים לDB שבראשותנו ולא להיפך.

מכאן הדרך לבחור את MongoDB שתשמש כפלטפורמה לDB שלנו הייתה קצרה מאחר ומבנה המידע שהיה לנו מMusicBrainz התאים מאוד לשימוש בMongoDB.

מתחילת הפרויקט, הבנו לראשונה כי למרות העידן בו אנו נמצאים, הקושי להשיג את המידע ולארגן אותו בDatabase היה מורכב וכלל אינטגרציה בין פלטפורמות שונות והיווה חלק גדול מהפרויקט שלנו.

ראינו כי קיים קושי לחשב את מדד הפופולאריות של שיר בתקופה בה הוא השתחרר והיינו צריכים להסתמך על מדד פופולאריות עכשווי עבור שיר שהתנגן לפני 40-60 שנה. ברור כי הקושי נובע מטכנולוגיה שלא הייתה קיימת באותה תקופה.

מסקנה נוספת שעלתה בזמן ביצוע הפרויקט הוא שיהיה עלינו להתגבר על אלמנט חשוב במערכות המלצה, שהוא המידע המערכת יודעת על המשתמש. מערכת המלצה קלאסית תאסוף מידע רב על המשתמש והעדפותיו ובכך תדע לתת המלצה על תכנים שמתאימים לו. בפרויקט נתקלנו בקושי שהמידע על המשתמש במערכת מוגבל והשתדלנו לתת המלצה טובה ביותר בהתבסס על נתונים אלה (גיל וארץ מוצא) ועליהם ביססנו את המערכת כולה.

לגבי ניהול המשתמשים במערכת, לאחר שהבנו שלרוב המשתמשים שלנו אין חשבונות Google   
ו Facebook פעילים, החלטנו שלא להשתמש בכלים מובנים לניהול המשתמשים במערכת כדוגמת firebase ולכן פיתחנו מערכת מאובטחת לניהול משתמשים מבוססת טכנולוגיית JSON Web Tokens.

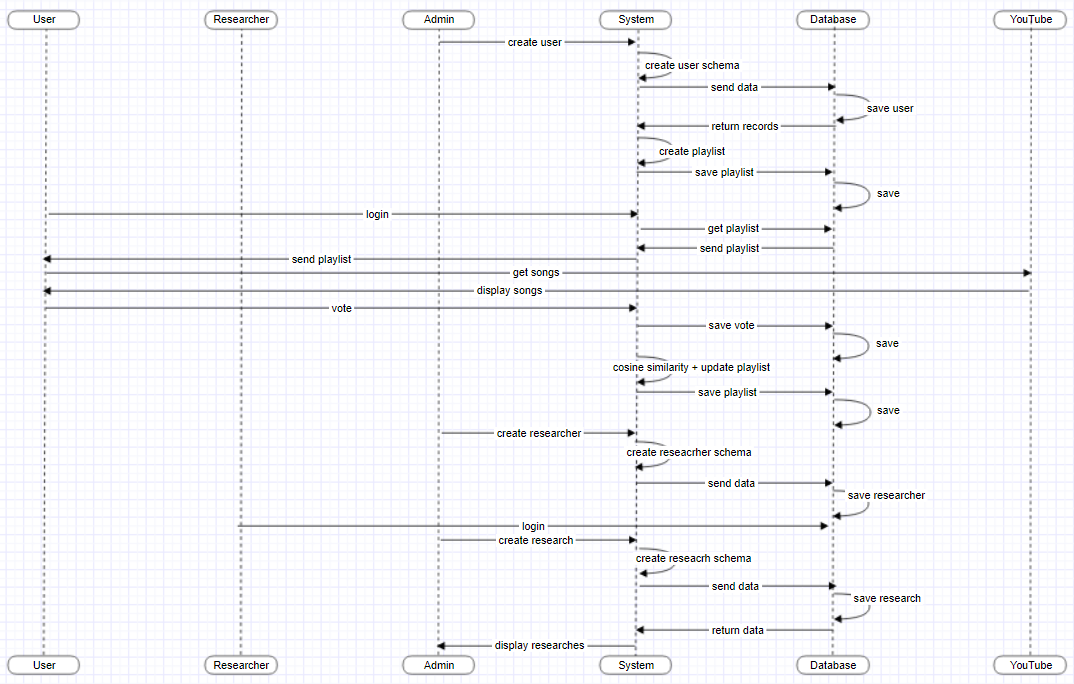
לאורך כל הדרך היינו צריכים לתכנן את המערכת שתהייה כמה שיותר פשוטה ולא מסורבלת מאחר והמשתמשים שלנו במקרה הטוב קשישים ובמקרה הרע קשישים חולי דמנציה ומצד שני לשמור על חווית משתמש ברמה גבוהה.

# **נספחים**

## **תרשימים וטבלאות**

## 

## Sequence diagram:



## **רשימת ספרות \ ביבליוגרפיה**

[1] mongoDB הוא בסיס נתונים מקטגוריית NoSQL, נשען על מבנה מסמך מעל מימוש של BSON (Binary JSON file). <https://www.sciencedirect.com/science/article/pii/S1742287616300317>

[2] Studio 3T הינו כלי GUI ניהולי וכלי לכתיבת שאילתות שפותח עבור mongoDB

<https://studio3t.com/>

[3] Cosine similarity הינו מדד של דמיון בין שני וקטורים שונים מאפס ומודד את גודל הזוית בניהם. חישוב זה משמש למדידת דמיון בין משתמשים על סמך הדירוג שלהם במערכות המלצה שונות. בפרויקט נעשה שימוש בחישוב זה על מנת לחשב דמיון בין משתשמים שהצביעו עבור אותם שירים. פרק 9 Recommendation System בספר Mining of Massive Datasets מציג וממחיש את יעילות החישוב. <http://infolab.stanford.edu/~ullman/mmds/book.pdf>

Just-for-Me: An Adaptive Personalization System for Location-Aware Social Music [4] Recommendation

מאמרם של Zhiyong Cheng Jialie Shen משנת 2014 המפתח מערכת המלצה של שירים המשלב מיקום גיאוגרפי. במאמר מתוארת מערכת המלצת שירים, שממליצה על שירים על פי מדד פופלריות, תגים, הקשר מילולי ומיקום גיאוגרפי. במאמר מתואר כיצד נבנתה המערכת ומתוארים תוצאות המעידות שההמלצה מדויקת יותר ומשביעה רצון כאשר נלקחים הפרמטרים הנ"ל כאשר ממליצים שירים עבור משתמשים.

<https://dl.acm.org/citation.cfm?id=2578751>

# Follow the algorithm: An exploratory investigation of music on YouTube [5]

מאמרם של Alessandro Gandini, Davide Berald, Massimo Airoldi 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>

# Spotify: How Does Spotify know you So Well? [6]

הוא שירות מוזיקה דיגיטלי המאפשר [הזרמה](https://he.wikipedia.org/wiki/%D7%94%D7%96%D7%A8%D7%9E%D7%AA_%D7%9E%D7%93%D7%99%D7%94) של [שירים](https://he.wikipedia.org/wiki/%D7%A9%D7%99%D7%A8) ו[פודקאסטים](https://he.wikipedia.org/wiki/%D7%A4%D7%95%D7%93%D7%A7%D7%90%D7%A1%D7%98). זהו שירות  [freemium](https://he.wikipedia.org/wiki/Freemium), כלומר ניתן לקבל את השירות בחינם עם תכונות בסיסיות, פרסומות והגבלות, אך תכונות נוספות כגון איכות האזנה גבוהה, או הורדה של התכנים, ניתנות למנויים תמורת תשלום. במאמר הבא של Sophia Ciocca משנת 2017 מתוארת איך המערכת לומדת את המשתמש בשביל להמליץ לו על שירים, בין היתר על ידי דירוג המשתמש עצמו.

<https://medium.com/s/story/spotifys-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe>

Music therapy in moderate and several dementia of Alzheimer’s type: a case-control study. [7]

מאמרם של [H. B. Svansdottir](https://www.cambridge.org/core/search?filters%5BauthorTerms%5D=H.%20B.%20Svansdottir&eventCode=SE-AU) , [J. Snaedal](https://www.cambridge.org/core/search?filters%5BauthorTerms%5D=J.%20Snaedal&eventCode=SE-AU) 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 מחלקת האלצהיימר והדמנציה.

<https://www.cambridge.org/core/journals/international-psychogeriatrics/article/music-therapy-in-moderate-and-severe-dementia-of-alzheimers-type-a-casecontrol-study/D51CCDFF17656F27C927D1BA322BA74C>

**Abstract**

The project has been created for Tamaringa Start-Up.

This project is continuation of Lea Cohen Saban and Avi Cohen and Sagi Marciano projects.

Tamaringa is a platform that creates personalized stimulation and audio for Dementia's patient to improve their condition.

During the project, we built a music recommendation system that works according to user criteria, similarities between users and ranking users so that the system will offer personalize music.

The system contains a research platform so that the user of the system can create research and track users.

This report displays the system implementation and how it is used, as well as the conclusion.

**Software Engineering Department**

**Final Project – 2019**

**Tamaringa**

**By**

**David Milstein**

**Yona Milstein**

**Academic Supervisor:**

**Dr. Miriam Allalouf**

**Software Engineering Department**

**Tamaringa**

**By**

**David Milstein**

**Yona Milstein**

June 2019